

# Computer Engineering Curricula 2016

## 计算机工程课程体系指南 CE2016

Association for Computing Machinery(ACM)

IEEE Computer Society(IEEE CS)

ACM 中国教育委员会

教育部高等学校计算机类专业教学指导委员会

编制

译

# 计算机工程课程体系指南

## CE2016

( Computer Engineering Curricula 2016)

Association for Computing Machinery( ACM)  
IEEE Computer Society( IEEE CS)

编制

ACM 中国教育委员会

教育部高等学校计算机类专业教学指导委员会 译

高等教育出版社·北京



## 译者序

2014年,美国计算机学会(ACM)和国际电子与电气工程师协会计算机学会(IEEE CS)启动了《计算机工程课程体系指南》(Computer Engineering Curricula 2004,以下简称CE2004)的修订工作。两家机构各选派专家组成了课程指导委员会,具体承担此项工作。受ACM邀请,ACM中国教育委员会派出清华大学计算机科学与技术系刘卫东教授和北京大学信息科学技术学院陆俊林博士参加课程指导委员会。

2016年12月,新版《计算机工程课程体系指南》(Computer Engineering Curricula 2016,以下简称CE2016)英文版在ACM网站上正式发布。而在此之前的2016年4月,经ACM授权同意,ACM中国教育委员会和教育部高等学校计算机类专业教学指导委员会(以下简称教指委)成立了联合工作组,启动了中文版的翻译工作。工作组由国内高校承担一线教学任务的教师志愿组成,义务承担翻译任务。具体成员和分工如下表所示:

章节	翻译人	审阅人
前言 第一章 附录 A:CE-SWD 附录 B.7 附录 C	刘卫东(清华大学)	陆俊林(北京大学)
第二章 附录 A.1~A.5 及 CE-CAE、CE-ESY 附录 B.1~B.3	陆俊林(北京大学)	袁春风(南京大学)
第三章 附录 A:CE-CAO 附录 B.4	袁春风(南京大学)	张亮(复旦大学)
第四章 附录 A:CE-SPE、CE-SRM 附录 B.5	张亮(复旦大学)	秦磊华(华中科技大学)

续表

章节	翻译人	审阅人
第五章 附录 A:CE-SGP、CE-DIG 附录 B.6	秦磊华(华中科技大学)	吴承荣(复旦大学)
第六章 附录 A:CE-PPP、CE-SEC 附录 B.8	吴承荣(复旦大学)	边凯归(北京大学)
第七章 附录 A:CE-CAL、CE-NWK 附录 B.9	边凯归(北京大学)	刘卫东(清华大学)
全书编辑	韩飞(高等教育出版社)	

工作组于2016年4月在太湖之滨的无锡召开了任务启动会,确定了具体分工和工作计划,明确了术语和重要词汇的翻译方法。2016年暑假前完成了第一稿翻译,经初步汇总后,经过一个暑期的努力,各位审核人提交了审核意见,形成了校核稿。同时,针对翻译中发现的原稿中存在的问题,给CE2016课程指导委员会提交了上百处的修订建议。2016年10月,课程指导委员会也对英文稿进行了最终审订,高度评价了中文翻译组的工作,采纳了绝大部分修改建议。修订完成的最终英文版经ACM和IEEE CS批准后,于2016年12月正式在网站发布(<http://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf>)。根据英文报告的最终版,工作组进行了相应中文版的修订,并于2017年3月在华中科技大学进行了最终审定,完成了翻译工作。

与CE2004比,CE2016对计算机工程专业的知识体系、知识领域进行了重新梳理,以保持教学内容的先进性,并期待能引领未来10年的计算机工程教育方向。CE2016取消了对知识点的叙述,而强调学习成效,希望各教学单位能更重视学生通过学习后所具备的能力,而不仅仅是流于表面地讲授哪些知识点,这与近几年国内强调系统能力培养的理念不谋而合。另外,更为重要的是,CE2016中用专门的章节,重点描述了对毕业生职业素质和社会实践能力的要求和实现手段,尤其是职业素质的培养,需要在专业教育中有针对性地加强。总之,课程指南的翻译,旨在方便国内计算机相关专业的教师及时了解、借鉴国外计算机专业本科培养方案及课程设置的最新成果,以进一步提高国内计算机专业的教学水平。

在翻译过程中,始终得到ACM中国教育委员会和教指委的有力领导。ACM中

国教育委员会主席张铭教授多次为翻译事宜和 ACM 总部协商。教指委秘书长马殿富教授十分关心翻译工作的进展,并给出了许多具体意见和建议。还要特别感谢 CE2016 课程指导委员会的全体同仁为翻译工作提供的便利,这次中英文版本的合作过程十分愉快。最后,还要对课程指导委员会主席、ACM 杰出教育家获得者、纽约 Hofstra 大学教授 John Impagliazzo 表示由衷的敬意。他不但领导了 CE2016 课程指导委员会团队,完成了英文版报告,在修订过程中两次来到中国征询国内专家的意见,还对中文版的翻译提供了许多便利条件。没有他的支持,不可能在如此短的时间内完成此项艰巨任务。

由于中美两国文化、教学方法和条件上的差异,一些译文可能无法精确表达原文的含义。另外,翻译时间相对比较仓促,工作组水平也有限,译文中难免还存在一些错误,欢迎读者批评指正。

译者

2017年3月19日



## CE2016 课程指导委员会

美国计算机学会 (ACM)	国际电子与电气工程师协会 计算机学会 (IEEE CS)
John Impagliazzo (主席) 美国, 霍夫斯特拉大学	Eric Durant 美国, 密尔沃基工程学院
Susan Conry 美国, 克拉克森大学	Herman Lam 美国, 佛罗里达大学
Joseph L. A. Hughes 美国, 佐治亚理工学院	Robert Reese 美国, 密西西比州立大学
刘卫东 中国, 清华大学	Lorraine Herger 美国, IBM 研究中心
陆俊林 中国, 北京大学	
Andrew McGettrick 英国, 思克莱德大学	
Victor Nelson 美国, 奥本大学	





# 前 言

本报告是大学本科计算机工程(Computer Engineering, CE)专业的课程指南。它总结了2004年出版的计算机工程课程报告,即《计算机工程2004:计算机工程专业本科课程体系指南》(CE2004)的反馈意见,也吸取了近年来美国计算机学会(ACM)、国际电子电气工程师协会计算机学会(IEEE CS)及美国信息系统协会(Association for Information Systems, AIS)等组织开发的计算机学科课程其他专业方向课程指南的经验,包括计算机科学(Computer Science, CS)[ACM/IEEECS, 2013]、信息技术(Information Technology, IT)[ACM/IEEECS 2008]和软件工程(Software Engineering, SE)。信息技术方向课程指南预期将于2017年推出新版。

从学科角度看,计算机工程属于电子电气工程和计算机科学交叉的一个广泛领域。本报告中,我们这样来定义计算机工程:

计算机工程是一门科学与技术学科,包含现代计算系统及计算机控制设备的软件和硬件组件的设计、构造、实现和维护。

因此,计算机工程专业培养的学生将面对从设计到实现计算机系统的职业生涯。计算机系统在当今社会中得到了广泛的应用,例如,汽车发动机中的燃油喷射系统、医疗设备中的X光机、通信设备中的智能手机以及家居中的报警系统和洗衣机等。设计计算机系统或其他产品中的计算部件,为物联网设计网络计算机和设备,开发和测试其原型系统,并将其实现后投入市场是计算机工程专业毕业生的典型工作。

本报告给出了计算机工程领域的一些背景,并回顾了其发展过程。报告还描述了对本学科毕业生的基本要求,以及与计算机其他学科方向毕业生的区别。这些要求包括计算机工程专业毕业生的知识背景、基本能力以及他们的雇主所期待的基本技能。具体来说,就是具备设计计算机系统的能力、具有重视专业技能的意识以及一名胜任实际工作要求的工程师所必需掌握的宽广的知识面。另外,报告还讨论了计算机工程专业如何应对政府或第三方机构的各项检查及认证的方法。

本报告提出了计算机工程专业的基本知识体系(Body of Knowledge, BoK),并以此为基础,为相关教学机构设计或修订培养方案提供指南和帮助。该知识体系包含有多个知识领域(Knowledge Area, KA),可适用于全世界范围的计算机工程专业

---

<sup>1</sup>方括号中引用的参考文献的完整信息可在附录D中获得。

教学。每个知识领域由主题范围描述和一组知识单元(Knowledge Unit, KU)构成。而对每个知识单元,则通过一组学习成效来进行定义。报告中进而将一些知识单元标记为“核心”,希望每个培养方案中都包含这些内容;其余的知识单元为建议选修。核心知识单元表示每个培养方案在该知识领域中应达到的最少的知识点或深度。对于计算机工程的培养方案来说,仅仅包含了核心知识单元是绝对不够的。

以上述计算机工程知识体系为基础,计算机工程专业培养方案应该包含有充足的初级、中级和高级课程,并附加精心选择和设计的选修课程加以完善。同时,具备相当深度和广度的自然科学和数学知识也是这个学科所必需的。设计环节对于培养方案来说也是十分重要的,一般是通过毕业设计或高级生产实习来实现。课程指南中还十分强调职业素养、法律和伦理问题和毕业生实现工程设计时要面临的相关社会环境问题。课程指南还强调,从事计算机工程的学习和研究,解决实际问题能力、批判性思维能力、人格魅力(软技能)、口头和书面沟通能力、团队合作能力和实际动手能力等是必要的基本能力。另外,本报告还提供了一些培养方案的示例,为相关教育机构根据本身的条件、使命和特定的培养目标开发计算机工程专业的培养计划提供方法上的指导。

这些建议可以帮助计算机工程专业培养方案的设计,培养出的毕业生可担任工业界入门级职位、并具备在工业界持续职业成长的能力,或者可以进入研究生阶段深入培养。本报告吸收了来自工业界和教学机构的反馈意见,是全球范围内专业人士合作的结晶,根本目的是为世界各地的教育机构或任何对其感兴趣的单位提供一个建设高水平计算机工程专业的有效途径。我们坚信这个目的已经达到。

CE2016 课程指导委员会

# 目 录

<b>第一章 引言</b> .....	1
1.1 计算机学科课程体系规范总体结构 .....	1
1.2 CE2016 编写过程 .....	2
1.3 指导原则 .....	3
1.4 CE2016 报告结构 .....	4
<b>第二章 计算机工程作为一门学科</b> .....	5
2.1 背景 .....	5
2.2 计算机工程学科的演变 .....	6
2.3 计算机工程专业毕业生的特质 .....	7
2.3.1 区分原则 .....	7
2.3.2 职业素养 .....	8
2.3.3 设计能力 .....	9
2.3.4 知识的广度 .....	9
2.4 组织部门的考虑 .....	10
2.5 专业实践的准备 .....	10
2.6 学科评估与认证 .....	11
<b>第三章 计算机工程知识体系</b> .....	13
3.1 知识体系的组织结构 .....	13
3.1.1 核心知识单元和附加知识单元 .....	13
3.1.2 估算一个知识单元教学所需的时间 .....	14
3.1.3 知识领域和知识单元的标识 .....	15
3.1.4 公共知识单元 .....	15
3.2 学习成效 .....	15
3.3 知识体系概要 .....	16
3.3.1 相关的数学基础 .....	19
3.3.2 相关的科学基础 .....	20
3.3.3 软件的作用 .....	21
3.4 CE2016 知识体系与 CE2004 知识体系比较 .....	21
3.5 计算机工程设定核心学时数的依据 .....	22

3.6	课程体系模型	22
<b>第四章</b>	<b>工程实践和计算机工程课程体系</b>	<b>24</b>
4.1	计算机工程探源	25
4.2	新兴技术策略	25
4.2.1	应用型新兴技术	25
4.2.2	概念型新兴技术	26
4.3	课程体系中的设计	26
4.3.1	贯穿课程体系的设计	26
4.3.2	毕业设计经历	27
4.4	实验经历	27
4.4.1	计算机工程实验	28
4.4.2	软件	29
4.4.3	开放式实验	31
4.4.4	随堂实验	31
4.4.5	技术支持	31
4.4.6	学生自行采购	31
4.5	工程工具的作用	32
4.6	计算机工程原理的应用	32
4.7	专业之外的技能	33
4.7.1	沟通技巧	33
4.7.2	团队合作技能	34
4.7.3	软技能或个人技能	35
4.7.4	经验	35
4.7.5	终身学习	36
4.7.6	商业视角	36
4.8	成为专业人士	37
4.9	工程教育的构成要素	37
4.10	研究生和继续职业教育	38
<b>第五章</b>	<b>专业实践</b>	<b>39</b>
5.1	专业实践综述	39
5.1.1	专业实践与计算机工程课程体系	40
5.1.2	专业人员需求	40
5.2	社会背景下的决策	40
5.3	职业化与教育	41
5.3.1	学生特殊体验	41

5.3.2	行政部门、教师与学生角色 .....	42
5.3.3	将专业实践纳入课程体系 .....	43
5.3.4	职业素养与学生体验 .....	43
5.4	职业素养与工作场所 .....	44
5.4.1	私营与公共部门 .....	44
5.4.2	模拟本地和国际工作环境 .....	45
5.4.3	资格证书 .....	46
5.5	职业精神培养 .....	46
5.5.1	职业道德规范 .....	46
5.5.2	教育与专业实践 .....	47
<b>第六章</b>	<b>课程体系实现问题 .....</b>	<b>48</b>
6.1	总体考虑 .....	48
6.2	课程体系设计原则 .....	49
6.3	计算机工程课程体系的基本成分 .....	50
6.3.1	知识体系核心单元的覆盖 .....	50
6.3.2	课程安排 .....	50
6.3.3	实验室实践 .....	51
6.3.4	毕业设计 .....	51
6.3.5	工程职业素养 .....	51
6.3.6	沟通技能 .....	51
6.4	由其他部门提供的课程 .....	51
6.4.1	数学要求 .....	52
6.4.2	科学要求 .....	53
6.4.3	其他要求 .....	53
6.5	课程样例 .....	54
<b>第七章</b>	<b>教育机构的适应性变化 .....</b>	<b>55</b>
7.1	自我调整的需求 .....	55
7.2	吸引和保留教师 .....	56
7.3	充足实验室资源的需求 .....	56
7.4	学业变更和教育途径 .....	57
7.4.1	四年制的学业变更 .....	57
7.4.2	技术学院(Technical Institute)的学业变更 .....	57
7.4.3	社区学院(Community College)的学业变更 .....	58
<b>附录 A</b>	<b>计算机工程专业知识体系 .....</b>	<b>59</b>
A.1	简介 .....	59

A. 2	知识体系 .....	59
A. 2. 1	核心成分与附加成分 .....	60
A. 2. 2	覆盖一个单元所需的时间 .....	60
A. 2. 3	知识领域和知识单元的标签 .....	61
A. 2. 4	共有的知识单元 .....	61
A. 3	学习成效 .....	61
A. 4	计算机工程知识体系总结 .....	61
A. 5	知识领域与知识单元 .....	65
CE-CAE	电路与电子 .....	65
CE-CAL	算法 .....	71
CE-CAO	计算机体系结构和组成 .....	74
CE-DIG	数字设计 .....	80
CE-ESY	嵌入式系统 .....	85
CE-NWK	计算机网络 .....	90
CE-PPP	职业素养和专业实习 .....	96
CE-SEC	信息安全 .....	101
CE-SGP	信号处理 .....	105
CE-SPE	系统与项目的工程化 .....	109
CE-SRM	系统资源管理 .....	117
CE-SWD	软件设计 .....	120
附录 B	计算机工程课程体系实例 .....	125
B. 1	格式和约定 .....	125
B. 1. 1	课程学时的约定 .....	125
B. 1. 2	计算机工程知识体系到课程体系实例的映射 .....	126
B. 1. 3	课程描述 .....	127
B. 2	开始职业生涯的准备 .....	127
B. 3	课程体系的共性 .....	128
B. 4	课程体系 A: 由电子与计算机工程系开设 .....	128
B. 4. 1	培养目标和特色 .....	128
B. 4. 2	培养要求汇总 .....	129
B. 4. 3	课程体系 A 的四年制模式 .....	129
B. 4. 4	计算机工程 BoK 与课程体系 A 之间的映射 .....	131
B. 4. 5	课程体系 A-课程概述 .....	133
B. 5	课程体系 B: 由计算机科学系开设 .....	137
B. 5. 1	培养目标和特色 .....	137

B. 5.2	培养要求汇总 .....	137
B. 5.3	课程体系 B 的四年制模式 .....	138
B. 5.4	计算机工程 BoK 与课程体系 B 之间的映射 .....	139
B. 5.5	课程体系 B-课程概述 .....	141
B. 6	课程体系 C:由计算机科学系和电子与计算机工程系联合开设 .....	144
B. 6.1	培养目标和特色 .....	144
B. 6.2	培养要求汇总 .....	145
B. 6.3	课程体系 C 的四年制模式 .....	145
B. 6.4	计算机工程 BoK 与课程体系 C 之间的映射 .....	147
B. 6.5	课程体系 C-课程概述 .....	148
B. 7	课程体系 D:由中国开设 .....	151
B. 7.1	培养目标和特色 .....	151
B. 7.2	培养要求汇总 .....	152
B. 7.3	课程体系 D 的四年制模式 .....	152
B. 7.4	计算机工程 BoK 与课程体系 D 之间的映射 .....	155
B. 7.5	课程体系 D-课程概述 .....	157
B. 8	课程体系 E:博洛尼亚-3 模型 .....	165
B. 8.1	培养目标和特色 .....	165
B. 8.2	培养要求汇总 .....	166
B. 8.3	课程体系 E 的三年制模式 .....	167
B. 8.4	计算机工程 BoK 与课程体系 E 之间的映射 .....	168
B. 8.5	课程体系 E-课程概述 .....	169
附录 C	计算机工程的实验 .....	178
C. 1	电路与电子学 .....	178
C. 2	计算机系统结构设计 .....	178
C. 3	数字逻辑设计 .....	179
C. 4	数字信号处理 .....	179
C. 5	数字逻辑与系统设计 .....	179
C. 6	嵌入式系统 .....	180
C. 7	工程学入门 .....	180
C. 8	网络 .....	181
C. 9	软件设计 .....	181
附录 D	致谢 .....	182
参考文献	.....	185





# 第一章 引言

1980年代,美国计算机学会(ACM)和国际电子电气工程师协会计算机学会(IEEE CS)成立了一个联合委员会,来编写计算机本科专业的课程体系规范(Computing Curricula, CC)指南。其工作成果就是诞生了课程体系规范1991,也称为CC1991或CC'91[CC91]。多年以来直至今天,这项工作在持续进行,并产生了一系列的文档。其中之一就是产生自CC'91的计算机工程专业课程体系指南,也就是我们熟知的CE2004[ACM/IEEECS, 2004]。本报告是对CE2004的修订,也专注于计算机工程专业,简称为CE2016。

CE2016努力的目标之一是对CE2004进行修订,使其能容纳过去十年计算机工程专业的发展成果,并满足未来十年的需要。在过去的这段时期,计算技术发展极为迅速,并以不同的方式对课程教学内容和教学方法产生了深远的影响。而我们努力的另外一个目标则是对世界范围内正在从事计算机工程专业培养方案设计或者教学的专业人士提供有力的支持。因此,本报告需要有国际视野,能反映全球计算机工程的发展成就。

## 1.1 计算机学科课程体系规范总体结构

由于计算机专业的范围不断扩展,以及我们收到的对前述课程体系规范的反馈意见,计算机专业课程体系规范(CC)首次被划分成多个组成报告。这些报告分别描述了本学科的重要方向,如计算机工程(CE)、计算机科学(CS)、信息系统(IS)、信息技术(IT)和软件工程(SE)等,它们均具备各自的特性,以及鲜明的教学传统。围绕这5个构成整体计算机专业的重要方向,学术机构分别编写了5个相似的课程规范报告。目前,这些分报告的最新版本包括计算机工程(CE2004)、计算机科学(CS2001, CS2008, CS2013)、信息系统(IS1997, IS2002, IS2006, IS2010)、信息技术(IT2008)和软件工程(SE2004, SE2015)。以上这些报告的最新版本可以参考[ACM/IEEECS, 2004]、[ACM/IEEECS 2013]、[ACM/IEEECS 2010]、[ACM/IEEECS 2008]、[ACM/IEEECS 2015]。

随着各分报告的逐步完成,来自这五个专业方向的专家代表一起撰写了一个总体报告(2005),将它们联系到一起。总体报告内容包含有对这几个专业方向的定义,以及它们之间的共同点和不同点的描述,还给出了计算机专业以后可能产生的新的专业方向的建议。图1.1来自总体报告的图1.1[OVERVIEW],给出了这些系

列报告的结构。

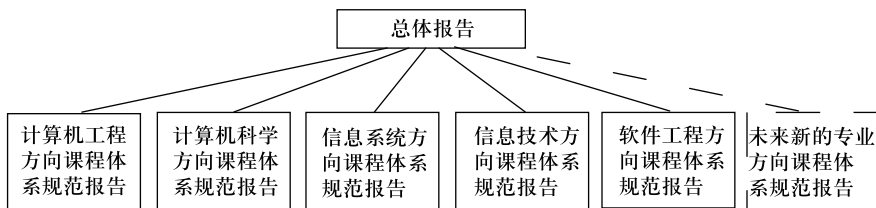


图 1.1 计算机专业课程规范报告

未来新的专业方向的课程体系规范报告尚在讨论中,包括网络空间安全和数据科学。

ACM、IEEE-CS 及其他的专业协会和组织也为其他团体提出能更好反映他们特定学科方向需求的课程报告提供了方便。当然,也确实对这些报告提出了一定的要求,如至少应该解决哪些问题,且这些特定报告中至少应包含哪些组成部分。这些要求包括:

- 本专业方向的知识体系(BoK);
- 通过一种或多种途径覆盖该知识体系的课程集合;
- 适用于本专业方向所有本科生的核心要求;
- 本专业方向培养的毕业生的基本素质。

专业组织提出的上述要求是对课程体系规范的最低要求,目的是不要使其过于教条和严格,也给编写规范的专家团队保留了充分的操作上的独立性,因此,这应该是各个分报告都应该遵循的原则。可以预期的是,在每个方向的报告中,都会以不同的方式满足并超出这个最低的要求。

## 1.2 CE2016 编写过程

为应对前面提及的 CE2016 编写所面临的挑战,ACM 和 IEEE-CS 组成了一个 CE2016 课程指导委员会,来编写计算机工程专业方向的课程体系规范。为更好地完成任务,委员会认为需要有更为广泛的参与度,因此,委员会由来自中国、英国和美国的代表组成。另外,委员会还在世界范围内举行了多次咨询活动,确保在本版本的报告中进一步吸纳来自全球专家的意见。

CE2016 课程指导委员会承担了本报告的主要编写任务,其成员在本报告的开始部分已列出。CE2016 项目规模庞大,涉及面广,课程指导委员会认为有必要让更多的专家参与到工作中,使本报告的内容更能反映本领域专家的观点和意见。因此,编写过程中,我们在全球范围内征集了意见和建议,并在本报告的附录 D 中列出

了提出宝贵建议的专家以示感谢。

## 1.3 指导原则

计算机工程是一个蓬勃发展的重要领域方向,鉴于此,CE2016 课程指导委员会提出了一系列原则来指导课程体系规范的编写工作。这些原则如下(排序不分先后):

1. 计算机工程的快速演进要求对其相应的课程体系进行不断地评估和改进。本学科的专业协会应当建立一个可伴随学科进步而对推荐的教学内容进行部分更新的持续的评估机制。

2. 计算机工程课程的开发和设置必须灵敏反映技术变化、教学方法的发展以及终身学习的重要性。而且,在计算机工程这样一样快速发展的领域,各教学部门必须采用切实可行的办法来适应变化。计算机工程教育也需要找到激励学生终身学习的办法,让他们能超越今天的技术,迎接未来的挑战。

3. 总结提炼出所有计算机工程专业方向本科毕业生都必须掌握的基础技能和知识十分重要。计算机工程是一个被广泛依赖的基础学科。本报告最终应当总结提炼出本学科方向的重要知识内容和技能。

4. 必须掌握的核心知识体系应该在合理范围内尽可能的小。为方便教学机构提出一个具有自身特点的教学计划,将核心知识体系的规模保持到最小范围十分重要,这样可以使基于此产生的不同培养方案具有灵活性、可定制性,并可兼容其他课程规范。

5. 计算机工程专业方向培养方案中必须要包含有适当和必要的设计和实验,培养的学生应当有设计、制造和测试硬、软件系统的“亲身经历”。

6. 计算机工程专业方向课程教学经常要受到认证、授权许可或行政管理的约束。本报告要充分认识到这些外部约束的存在,并为课程建设提供相应的指导。

7. 计算机工程专业方向课程中,必须将专业素养培育作为其不可或缺的组成部分。这些专业素养包括管理能力、社会伦理和价值观、书面和口头交流能力、团队工作能力和保持在快速发展的学科中不落伍的能力。

8. 本计算机工程专业方向的课程体系规范在提出高水平教学内容的同时,还要讨论这些教学内容的实现策略和方法。尽管计算机课程体系规范中应清晰描述出计算机专业教学的宽广视野,但任何课程规范的成功均严重依赖于实现的细节。为达到这个目的,报告应提供一些培养方案的示例。

9. 本报告最终应该建立在广泛共识的基础上。为此,报告的编写过程应该通过不同的方式,得到来自产业界、政府管理部门以及与计算机工程教育有关的全部高等教育机构等各方面力量的参与和支持。

10. 本报告应该可以服务于全球范围的计算机工程教学。尽管每个国家的课程要求都不相同,我们希望本报告可以对全世界的计算机教育工作者提供帮助。虽然美国教育实践对课程规范有较大影响,但我们尽力确保报告中推荐的教学内容对不同国家和文化的差异保持敏感,使之能在世界范围内得到广泛应用。

11. 相关工具、标准及/或工程约束应该在整个知识体系的各知识领域中出现,因为它们是所有工程师完成实际工作的基础。

12. 学习成效是本科专业教育中不可缺少的组成部分。由于学习成效可直接对应于知识点和概念,为避免重复,本报告中将不再包含有知识点。

13. 硬件和软件的集成和整合是计算机工程的关键工作。由于计算机系统由硬件和软件组件集成而来,本报告应该在实验部分强调“完整计算机系统”的开发,即能完成一定应用功能的硬件、操作系统、软件系统的设计和实现。

14. 设计这个概念必须是贯穿整个报告的主线。

## 1.4 CE2016 报告结构

CE2016 是计算机工程专业方向的本科课程体系规范报告。报告正文部分包括本章共有 6 章。第二章说明计算机工程成为本科专业的原因,并强调了计算机工程专业本科毕业生预期的基本特征,包括他们服务公众的能力、设计能力以及掌握知识的宽广程度。该章还对可能的组织结构、职业素养的责任以及培养方案的评估给出了建议。第三章和第四章给出了计算机工程专业方向知识体系的概览,并描述了达成这些知识体系的教学内容。在这两章中,还详细论述了学习成效的概念、核心知识单元和选修知识单元的不同、培养方案中的核心学时数、设计和实验经历的重要性以及成为合格计算机工程师所应具备的各种技能。第五章强调职业素养在计算机工程实际工作中的重要性。第六章讨论了影响计算机工程课程体系规范实现的问题,包括学生学习计划的安排、专业课程和其他教学环节课程的关系,以及其他实现方面的问题。第七章对创建或发展一个计算机工程培养方案的挑战性提出了一些建议。本报告还提供了我们引用的参考文献的列表。

本报告的主体部分是它的两个附录。附录 A 详细说明了计算机工程专业方向的本科教育知识体系。包括全部的知识领域、每个知识领域相关的知识单元以及学生学习完这些知识单元后的成效。附录 B 给出了不同教育机构的一些教学方案和典型课程的示例。CE2016 课程指导委员会希望通过提供知识体系、示例方案和课程描述来帮助各部门提出有效的培养方案,或者帮助改进他们已有的教学方案。附录 C 提供了计算机工程专业方向相关实验开发课程的一些指导原则。附录 D 列出了为本报告的改进提出了宝贵意见的世界各地专家,并向他们致谢。

## 第二章 计算机工程作为一门学科

本章介绍计算机工程作为一门学科区别于其他计算学科的特点。本章提供了本领域的背景,并展示了计算机工程是如何演变而成的。本章强调了学习计算机工程专业相关课程应有的基础、该专业毕业生应有的素质,以及对学生学习成效的考察和评价。本章还强调了毕业生具备正确职业素养的重要性,这将确保他们在计算机工程实践中有正确的世界观指导。

### 2.1 背景

计算机工程是一门关于设计、构建、实现和维护现代计算机系统、计算机控制设备和网络智能设备的软硬件组件的学科。从传统上看,计算机工程是电子工程和计算机科学二者在一定程度上的结合。经过四十年的演变,它已经成为一门独立但仍和电子工程、计算机科学保持密切关联的学科。计算机工程扎根于计算机、数学、科学、工程等学科的理论 and 基本原理,运用这些理论和原理,通过计算机硬件、软件、网络和过程的设计来解决技术问题。

从历史角度看,计算机工程领域被广泛认为是“设计计算机”。在现实中,计算机本身的设计一直是相对较少的高度熟练的工程师的职责范围,其目标是推动计算机和微电子技术的前进。随着硅器件的小型化,以及系统构建模块和完整片上系统可靠性的提高,计算机已经越来越普遍地取代了传统的电子设备。其应用主要体现在移动智能手机、平板电脑、多媒体和定位设备、无线网络以及类似产品。同时还体现在了大量的嵌入式系统中,如汽车、控制系统、大型家电和物联网这类设备。

越来越多的计算机工程师参与到基于计算机系统的设计中,以满足高度专业化和特定性的应用需求。在大多数行业中,都有计算机工程师在工作,包括计算机、汽车、航空、电信、电力生产、制造、国防、电子等。他们设计的高科技设备涵盖了从微电子集成电路芯片到采用了这些芯片的强大系统,以及连接这些系统的高效通信系统,还包括分布式计算环境(局域网、广域网、无线网络、互联网、内部网)和嵌入式计算机系统(如:飞机、飞船、汽车控制系统中执行各种功能的嵌入式计算机)。像发电和配电系统和现代化的加工和制造厂这样复杂的技术系统,都在由计算机工程师通过计算机系统的设计 and 开发。

技术进步和创新继续推动计算机工程的发展。现在通过若干技术的融合(如多媒体、计算机和网络技术),就可以广泛传播和随时访问海量的信息。技术融合与创

新已经成为经济发展的核心和众多组织的未来,这给计算机工程师带来了许多机会和挑战。这预示着计算机工程会带来成功的职业生涯。

## 2.2 计算机工程学科的演变

如前所述,计算机工程是从电子工程和计算机科学的学科发展而来的。计算机工程的最初课程往往来自于电子工程专业内部,从数字逻辑设计扩展到小型数字系统的创建,并最终完成微处理器和计算机系统的设计。后来计算机工程的课程越来越多地开始引入计算机科学相关的知识,并最终发展到与之融合。如今,这一趋势正在逐渐减弱,计算机工程专业已经形成了自身的知识领域。本报告就体现了这一过程。

在中国,大学的计算机教育已经发展了六十多年。第一个快速发展阶段是在 20 世纪 50 年代末,十五所大学开设了计算机专业。当时这些专业的名称多为“计算设备”,强调从基本组件开始设计计算机。这些与计算机工程专业的要求非常相似。第二次快速发展阶段是 20 世纪 70 年代末至 80 年代中期,有 74 个专业成立于这一时期,名称包括“计算机应用”和“计算机软件”。有一些大学在“计算机应用”专业上强调硬件,这也反映了计算机工程的需求。从 20 世纪 90 年代中期开始,中国计算机教育进入了第三个快速发展阶段,高校开设了五百多个相关专业。在这一阶段,教育部指定“计算机科学与技术”作为一级学科,以“计算机软件与理论”,“计算机系统结构”和“计算机应用技术”作为二级学科。计算机工程相关的知识出现在一级学科“计算机科学与技术”中。2010 年之后,教育部增加了“软件工程”和“网络信息安全”作为新的一级学科,与“计算机科学与技术”并列。

在英国,计算机工程专业往往与曼彻斯特大学和剑桥大学这些从事早期计算机开发的大学相关。曼彻斯特大学曾在 1980 年推出了计算机工程学位,但没有延续下去。当时的一些大学里,确实已经出现了反映计算机工程的知识点。如今,“计算机系统工程”、“计算与电子系统”这样的学位课程更为常见,体现了包括计算机工程在内但更为广泛的工程方面知识点。英国几乎所有的大学都在计算机系统工程领域提供学位。很大程度上得益于市场的力量,有很多学位要求一段时间的工业界经历(例如,计算机系统工程与工业实习)或包含一种现代语言(例如,计算机系统工程与现代语言)。

在英国,工程委员会[EngC]全面负责工程学位的认证,并维护认证的学位专业清单。通过其搜索工具显示,目前(2015 年 8 月)工程委员会已经认证了 44 个计算机系统工程学位专业和 11 个计算机及电子系统工程学位专业。由于英国的学位头衔种类繁多,上述的数量应该是被低估的。

在美国,第一个计算机工程专业是由 ABET(原名工程与技术认证委员会)的工

程认证委员会(EAC)于1971年在凯斯西储大学所认证。截至2016年10月,工程认证委员会(EAC of ABET)认证了超过301个计算机工程或类似名称的专业,包括美国以外的49个专业。表2.1展现了专业的发展情况,包括专业名称和ABET首次认证(或专业名称变化)的年份。可与之对比的是,大约有370个认证的电气学或电子学的工程专业。

表 2.1 总结 ABET 认证的计算机工程专业(截至 2016 年 10 月)<sup>i</sup>  
(括号中为国际专业数量)

专业名称	首次认证的年份					合计
	1980 之前	1980-1989	1990-1999	2000-2009	2010-2016	
计算机工程	10	39	56	96(16)	40(21)	241(37)
计算机系统工程	2	1	0	3(1)	2(1)	8(2)
电子与计算机工程 (包括之前专业的名称为 EE)	8	5	3	12(3)	4(1)	32(4)
计算机科学与工程	2	6	2	3(1)	3(1)	16(2)
其他名称	0	0	0	1(1)	3(3)	4(4)
合计	22	51	61	115(22)	52(27)	301(49)

可以预期的是,随着计算技术和电子技术的日益复杂化,计算机工程专业将呈现增长趋势。其演进过程可能会有多种形式,包括但不限于:

- 与计算机专业集成整合,并从中扩展的内容;
- 与软件工程专业和嵌入式系统专业协作,前者着重在面向应用的工程项目,后者则更强调用设计与分析工具来进行复杂度管理;
- 与电子工程专业再次整合,因为基于计算机的系统已在控制系统和通讯等领域中占据了主导地位。

## 2.3 计算机工程专业毕业生的特质

随着计算机或基于计算机的系统和网络在当今世界的普及,计算机工程师必须掌握全面的知识,除了计算机科学和电子工程中的标准知识点,还有数学和科学的基础。计算领域的迅速变化,要求计算机工程师必须是终身学习者,以在所选择的这个学科中保持必备的知识 and 技能。

### 2.3.1 区分原则

如何区分计算机工程师、电子工程师和其他计算机专业人员以及工程技术人员



是非常重要的。计算机工程师一般应该具备以下 3 个特点,尽管这样的区分原则有时也是模糊的。

- 具备设计计算机或基于计算机的系统和网络的软硬件的能力,以及集成它们以解决新的工程问题的能力,并能够在一组相互制约的目标和约束条件之间进行权衡。在这种情况下,“设计”指的是超越“装配”或“配置”系统的水平能力。

- 在数学和工程学科方面有着广泛的知识,能联系到更广泛的工程领域,并且不仅限于该领域的需求。

- 获得并保持计算机工程领域进行专业实践的能力。

其他相关学科的特点可以概述如下。

- 电子工程涵盖了广泛的领域,包括生物工程、电力工程、电子、通信和数字系统。在与计算机工程有关的领域中,电气工程师主要关注的是电子电路、信号分析和微电子设备的物理方面问题。

- 计算机科学家主要关注的是有关计算理论和算法方面的计算理论基础问题。

- 软件工程师专注于开发(通常是大型的)软件和维护其在生命周期中保持正确,信息系统专家关注组织过程中获取、部署和管理信息资源。

- 信息技术专家专注于通过计算技术的选择、创建、应用、集成和管理来满足用户在组织和社会环境中的需求。

- 计算机工程技术人员则是安装、操作并维护基于计算机的产品以向工程师提供支持。

### 2.3.2 职业素养

因为计算机工程师设计的系统(比如 X 射线机、空中交通管制系统或核电站)都直接或间接影响到公众,所以工程师是承担了公共责任的。对于计算机工程师来说,以最高的责任心设计和实现计算系统是义不容辞的。因此,本专业的毕业生应该深入理解工程实践相关的责任,包括所从事工作专业、社会和道德等方面。这种责任往往涉及财经和社会方面的复杂权衡,包括一系列的法律和经济问题,如知识产权、安全和隐私、债务、技术接入和技术运用全球化的影响。

职业素养和道德规范是至关重要的因素。因为工程专业强调设计与开发,这使得在该领域研究社会环境变得非常重要。计算机工程专业学生必须学会在职业生涯中结合理论、职业实践和社会构造。所有的计算机工程师都有义不容辞的责任去坚持自己的专业原则和职业行为法规。公众期望工程师遵守行规,不从事会损害其形象或同行的活动。由于本项内容的重要性,本报告的第五章专门进行关于职业行为和责任的扩展讨论。

### 2.3.3 设计能力

工程非常倚重设计能力。国际技术与工程教育协会(ITEEA)定义工程设计为“科学和数学原理在实践中的系统性和创造性应用,如设计、制造和操作高效的和经济的建筑、机械、流程和系统。”[ITEEA]可能也有其他的定义,比如开发更好的设备、系统、流程和新产品所需的创新能力。许多原因促使新的设计,如利用相关技术探索新发展,或在现有产品上改进(例如,使产品更便宜、更安全、更灵活或更轻便)。明确现有产品的缺陷或弱点,是工程设计的另一个动力。当然,新奇的想法尤其重要。

设计是所有工程的基础。对于计算机工程师而言,设计一词涉及到创建和集成现代计算机系统和计算机控制设备的软件和硬件组件。计算机工程师运用科学和数学的理论和原理,设计和集成硬件、软件、网络和流程,并解决技术问题。计算机和数字系统的持续进步,让专业人士有机会将这些进步广泛地应用于工程中。从根本上说,这是在诸多限定条件下深思熟虑后做出选择或进行权衡。这些涉及结构和组织、技巧、技术、方法、接口以及组件的选择等问题。成果需要表现出良好的性能,而且往往趋向于简单和简洁。第四章更详细地讨论设计以及相关的实验室经验。

### 2.3.4 知识的广度

由于计算机工程领域的广泛性,在不同的专业中课程内容会有很大的不同,甚至是同一个专业中的学生所学课程也是如此。计算机方面的课程通常由计算机组织与体系结构、算法、编程、数据库、网络、软件工程和通信等组成。电子工程方面的课程通常由电路、数字逻辑、微电子、信号处理、电磁学、控制系统和集成电路设计等组成。典型的知识点通常包括科学基础、离散及连续域数学、概率论和数理统计的应用。

对于计算机工程专业的课程设置,一个可能的极端做法是:培养学生对整个领域的广泛知识点的掌握;另一个可能的极端做法是:让学生专注于计算机工程一个特定方面的专业,以使学生在相关方面达到一定的深度。这些专业的毕业生通常会倾向于在他们学习过的特定领域寻求发展机会,无论是多媒体系统开发、计算机设计、网络设计、关键安全系统、普适计算,或其他新兴的并变得重要的方向。有一种区分计算机工程专业不同设置的通用方法,就是考量其知识点在电子工程或计算机科学课程上的分布情况。

尽管计算机工程专业的重点和内容设置有一些差异,但也有共同的元素。如第3章中描述的知识主体,指出了在所有的计算机工程专业中应该包含的知识点,而不是那些仅出现在少数专业或选修的知识点。然而,从一个更高层次的角度来看,人们可以认为所有计算机工程专业毕业生应具备几个共同特点,包括:

- 系统层面的视角——毕业生应该理解计算机系统的概念、系统的硬件和软件设计,以及关于系统的构建、分析和维护的过程,应该比从外部观察系统工作情况或系统使用方式更为深入的理解。
- 深度和广度——毕业生应广泛熟悉本学科的知识,并在一个或多个领域掌握深入的知识。
- 设计经验——毕业生应完成一系列的设计实践,包括设计硬件和软件组件,以及在已有工作的基础上集成这些组件,并至少包括一个主修的项目。
- 工具的运用——毕业生应能够使用各种基于计算机的实验室工具进行计算机系统的分析和设计,计算机系统中包括硬件和软件组件。
- 专业实践——毕业生应了解工程实践所处的社会环境,以及工程项目对社会的影响。
- 交流技能——毕业生应能够以适当的形式(书面、口头、绘图)与其他人交流自己的工作,并对他人的工作进行评估。

## 2.4 组织部门的考虑

计算机工程专业的管理属于各种组织机构。目前,计算机工程专业很少组织成独立的学术部门。这类专业经常出现在工程学院,要么是电子工程系,要么是电子和计算机工程系,抑或是一个综合工程系。在这种情况下,电路和电子元件自然成为强调的重点。计算机工程专业也出现在诸如计算机科学系、艺术与科学学院、信息技术学院或分部,或由多个实体共同资助的机构。在这些情况下,专业设置往往从理论抽象或组织机构的角度出发,而不是更多从应用角度来考虑。最后,计算机工程专业可以由两个这样的系(例如,电子工程和计算机科学)共同管理。在这种情况下下的专业设置会尝试在课程中融合硬件和软件组件以求平衡。

如表 2.1 所示,这些专业最常见的学位名称是“计算机工程”。其余名称也可以反映出各自的专业方向、组织结构、历史局限或其他因素。无论其组织结构或正式的学位名称如何,本报告提出的原则实际上适用于所有的计算机工程专业。

## 2.5 专业实践的准备

与法律、医药等行业不同,在工程领域就业一般不需要有高级学位。因此,计算机工程专业的本科课程,不但必须包括该领域的基本知识,而且必须将其应用到实际专业的解决方案中。这个包括几个方面的准备工作。

第 2.3.2 节定义了职业素养和职业道德是计算机工程专业毕业生的基本特质。因此,毕业生为专业实践所作的准备包括对工程实践相关职责的理解,以及运用原

则应对特定情况的能力。

职业化应该是一个永恒的主题,贯穿于整个课程。尤其是工程所处的社会环境,应该作为工程设计的教学内容,包括技术、财政和社会需求之间进行最优的实践和权衡。

除了职业素养,还有技术(包括工程设计工具运用能力以及实验室经验)和非技术(团队合作、沟通)方面的适当准备。本报告的第六章对此进行了详细的讨论,并将这些问题纳入课程体系当中。

## 2.6 学科评估与认证

学科评估程序必须适应计算机工程专业的变化。这样的评估是至关重要的,可以确保毕业生已经做好充分的准备,并且确保专业在不断发展以满足本领域的新需求。通常,专业协会和政府都会考虑采用外部评估,以确保毕业生达到职业机构所期望的最低要求。

在澳大利亚,澳大利亚工程师协会承担工程专业的入门认证,通常为五年周期。该认证确保学术机构符合国家和国际标准,确保被认证的工程专业毕业生获得澳大利亚工程师协会相应职业等级的会员资格,并享有海外同等专业机构的对等权益。澳大利亚工程师协会所使用的“澳大利亚学历资格框架”(AQF)于2015年生效。

在中国,教育部先后于2004年、2008年和2012年组织了三次学科评估。大学可以选择是否加入评估。2012年进行的学科评估,由覆盖了4235个学科的391所大学和机构参与。除了两所大学之外,几乎所有的“211工程”大学和“985工程”大学都参加了本次评估。参加评估的国家级重点学科的比例达到93%。计算机科学与技术,其中包含了计算机工程在内,是被评估的一级学科之一。评估指标主要包括教师和资源、科研质量、培训质量和学科声誉。评估持续一年以上,于2013年1月发布了评估报告。目前,教育部正在为第四个评估周期做准备。

在英国,学位基准是作为英国政府质量保障工作的一部分出现的。每一个机构必须证明自己的学位符合该学科的基准。这些基准的一个例子是主题基准陈述[SBS]。基准测试通常定义阈值(最小)和模态(平均)的数学期望,以着重体现学生的知识、技能和判断能力。

英国工程师委员会(EngC)承担英国以及之外的工程学位认证责任。其基本职责包括为工程学位的认证制定标准(关于能力和承诺),并批准提名机构代表以进行详细认证[ENGC]。一般来说,英国计算机协会(BCS)进行计算学位专业的认证。无论是英国计算机协会还是工程技术学院(IET)都可以认证计算机工程学位专业。由两个机构联合认证的情况是很常见的。

在美国,ABET认证是被广泛认可和接受的。此外,ABET目前还在另外二十八

个国家进行专业认证。用于工程专业认证的 ABET EAC 标准[ ABET,2016 ],目的是确保所有经过 ABET EAC 标准认证的专业满足一个标准最小集。该最小集中包含适用于所有工程学科的通用标准和针对每个学科的专用标准。该程序的关键因素是要求每一个专业都处在持续地自我评估和持续地改进的过程中。所有毕业生都应该达到该专业教学目标要求的学习成果。ABET 标准是比较宽泛的。它把学科知识结构的解释权,留给了该学科的专业机构。我们希望本报告可以在计算机工程专业的技术内容方面为认证机构提供指导意见。

许多国家已经通过政府或专业协会建立了自己的评估和(或)认证程序。在华盛顿协议[ Washington ]、首尔协议[ Seoul ]、悉尼协议[ Sydney ]、都柏林协议[ Dublin ]、欧洲国家工程协会联合会[ FEANI ]和国际注册专业工程师[ IRPE ]之间,都存在评审和(或)认证程序的互认机制。

一般情况下,机构倾向于使用认证作为媒介,证明他们有资质进行营销;大多数提供工程学位的机构都通过了某种形式的认证。从获认证的工程专业毕业,通常是取得专业注册或执照的先决步骤。目前,有些工作岗位要求认证学位状况或者专业执照,不过这种情况在计算相关领域不像其他工程领域那么普遍。

虽然认证和基准测试标准通常指向毕业生的最低标准或平均水平,但是也期望借此让计算机工程专业为最优秀的学生提供实现他们潜力的机会。这些学生通过运用课程中学到的原理,将会更具创造力和创新性;他们将能够在复杂系统的分析、设计和开发中做出自己的贡献;他们也能够对自己的和他人的工作进行评估和审查。

## 第三章 计算机工程知识体系

计算机工程本科课程体系规范应反映计算机工程专业学生的现实需求,也要满足未来进入研究生阶段学习或工业界工作的要求。该课程体系规范还必须反映目前现有的教育实际情况,并提出必要的改进建议。以下讨论将以此为目标,提出一套能满足培养计算机工程专业合格毕业生的知识体系。

该报告定义了一组知识领域(Knowledge Area, KA),每个知识领域包含一系列相关核心知识单元和可选知识单元(Knowledge Unit, KU),每个知识单元都给出了相应的学习成效。所有知识领域集合形成了课程规范的基础,但是,仅给出知识领域还不够。该报告列出的知识领域大约仅代表一个典型四年制培养方案中所包含的学科内容的50%。此外,对于计算机工程方面的培养方案,应确保其学科内容至少包含1.5年工程方面的训练。本报告后续的章节将进一步阐述为满足特定专业目标而设计一个完整的计算机工程专业课程体系所需考虑的其他方面。

### 3.1 知识体系的组织结构

从结构上说,计算机工程专业的知识体系可以划分为三个层次。最高一层为知识领域,表示一个特定的学科分支,而不是一门课程。知识领域还包含一个“领域界限”,它界定了特定知识领域涵义。知识领域再被分解为更小的组成部分,称为知识单元,它表示在一个知识领域中单独的主题。每个知识单元将对应一组学习成效,属于最低层次。

#### 3.1.1 核心知识单元和附加知识单元

修订 CE2004 报告的目标之一是保持知识体系的必要部分尽可能小。这样可以使不同教学机构尽可能灵活地设置培养方案,来保持其教学目标或目的多样性。为了贯彻这个原则,我们将知识单元分成核心(或基础)和附加(或额外)两类。核心知识单元包含了该学科四年制本科毕业生应该具备并广泛认可的知识和技能。附加知识单元包含一个特定的培养方案所要求的能够从事更高层次工作所需的知识和技能。

在讨论 CE 2016 报告的过程中,课程指导委员会发现强调以下几点非常有帮助。

- 核心知识单元是指计算机工程的所有培养方案的每一个本科毕业生应该具

备的知识和技能。有些学习成效虽然对许多学生的专业教育来说都是非常重要的,但因为不是对所有学生都需要,因而也不包含在核心知识单元部分,而只能出现在附加知识单元部分。这些学习成效没有包含在核心知识单元部分,并不意味着其价值、重要性以及与其他知识点的关联等方面具有负面性,而只是简单意味着它们并不是所有 CE 培养方案的学生都必须具备的知识和技能。

- 知识领域不是课程,核心知识部分也不能组成一个完整的课程体系。每个培养方案可以以不同的方式选择如何涵盖核心知识单元。

- 其他的技术领域,以及起支撑作用的数学、科学和通识课程,对于塑造一个有竞争力的计算机工程师也是必要的。

- 一个四年制培养方案不应该只在早期入门课程中实现对核心知识单元的教学,许多核心知识单元确实是入门性的,不过,有些核心知识单元可以学生已经建立了必要的知识背景的前提下再进行教学。

### 3.1.2 估算一个知识单元教学所需的时间

为了给读者一个完成特定知识单元教学的时间上的感觉,本报告采用其他课程规范报告中相同的模式。CE2016 课程指导委员会采用小时来计量时间,具体来说,特指核心学时。这对应的是采用传统的课堂讲授教学方式,完成知识单元教学内容的课内学时。因此,我们定义一个“核心学时”为一个讲课学时,通常为 50 分钟。

为消除所有可能的混淆,我们在此着重强调将讲课学时作为衡量标准的几点想法:

- 本报告并没有打算认同某一种授课方式。即使我们曾经采用以传统的课堂讲授形式为根本来作为学时的测算单位,我们相信这对其他授课形式起码也是有效的,尤其是最近这些在教育技术方面取得进步的新形式,如翻转课堂( Flipped Classroom)、大规模在线开放课程( Massive Open Online Courses, MOOC)、混合式学习( Blended Learning)、预录课程( Prerecorded Lectures)和研讨课程( Seminar)。对其中一些新的教学形式,准确计算学时可能比较困难。即使这样,需要用到的时间至少可以作为一个可比较的参数。从某种意义上说,不管采用哪种教学形式,学习一个五学时的核心知识单元可以推断是学习一个学时核心知识单元时间的 5 倍。

- 对于实验部分的学时估算,通常,3 学时的师生交互的实验课程( Contact Laboratory)相当于一学时的讲座课程。也就是说,150 分钟(3 个 50 分钟的实验学时)相当于一个讲座学时。不过,不同学校可能有不同的计算方法。尽管如此,对于计算机工程来说实验是必需的,因此在计算每个知识单元学时时,必须包含实验学时。

- 在进行学时估算时没有包含课外时间。每个知识单元所分配的学时也不包含教师的备课时间或是学生预习或复习时间。作为一种惯例,课外时间应该大约是课内时间的三倍。因此,对于一个 3 学时的知识单元,其总学时大约是 12 个学时

(课内 3 学时,课外 9 学时)。

- 每个知识单元所列出的学时数是一个最小值,大家可以把它理解为学生要达成一个知识单元的学习成效所必需的最少学时。许多老师将会发现,如果要向学生讲授更深的内容可能需要更多的学时,因此,花费比最少建议学时更多的时间是很正常的。

### 3.1.3 知识领域和知识单元的标识

我们用像 CE-NWK 这样的标识表示知识领域,CE-NWK 表示计算机工程中的计算机网络(Computer Networks)。在标识后面增加的一个数字表示知识单元,例如,CE-NWK-2 表示计算机网络中的第 2 个知识单元。附加的知识单元仅仅只有可选的学习成效,而没有给出任何建议学时数。

### 3.1.4 公共知识单元

在每个知识领域中,第一个知识单元都是“历史沿革与综述(History and overview)”,第二个知识单元都是“相关工具、标准和(或)工程约束(Relevant tools, standards and/or engineering constraints)”。这两种知识单元提供了相关知识领域中其他知识单元的背景信息。简短的历史回顾和综述为学习成效提供了背景知识,包括该领域中重要的贡献者、发展沿革等。工程实践必须使用现代工具和当代标准,这些工具和标准可能随时间而变化。对于不同的知识领域和不同培养方案目标,这两个知识单元的扩展内容有极大的不同。

## 3.2 学习成效

对于每个知识单元,学生应该学习什么内容呢?为了帮助理解这个问题,本报告使用学习成效(Learning Outcomes)来描述每个知识单元。这里,对学习(Learning)的强调是非常重要的,并使用“定义(define)”、“评估(evaluate)”等动词用来描述所期望的学习程度。学习的程度包括从基本能力(如复述定义)到高级能力(如综合运用和评估)各个层次。知识单元中用来描述学习成效的动词受到了 bloom 分类法的影响[Bloom 1956]。因此,学习成效提供了一种机制,这种机制不但描述了相关知识和实践技能本身,还描述了个人掌握和转换技能的程度。学习成效描述了我们期望一个学生在毕业时必须了解的知识和具备的能力。我们可以从表述学习成效的语句中推断相应知识单元所对应的最低学习要求。学习成效并非限定在某个知识单元,也可以是和一个课程活动、一门课甚至是一个培养方案相关联。

在这个报告中,CE2016 课程指导委员会试图限制学习成效的数量,以强调必备的技能 and 知识。这样,制定培养方案时可以根据学习成效来构造课程体系,使学



们通过各种不同方式或途径来表明他们已经获取了相关知识和技能。而通过对一组精心构思的对学生掌握技能的程度的特定描述,我们也可以得到评估学习成效的有效方法。

### 3.3 知识体系概要

表 3.1 给出了知识体系所包含的 12 个知识领域。表 3.2 出示了本报告给出的 12 个知识领域以及各自所包含的知识单元。这两个表就是 CE 的知识体系描述。表 3.2 中还给出了每个知识领域和知识单元的核心学时(核心课程学时),例如,

表 3.1 CE2016 知识领域

CE-CAE 电路与电子	CE-PPP 职业素养和专业实习
CE-CAL 算法	CE-SEC 信息安全
CE-CAO 计算机体系结构和组成	CE-SGP 信号处理
CE-DIG 数字设计	CE-SPE 系统与项目的工程化
CE-ESY 嵌入式系统	CE-SRM 系统资源管理
CE-NWK 计算机网络	CE-SWD 软件设计

表 3.2 CE2016 知识体系  
(核心学时: 420)

计算机工程知识领域和单元	
CE-CAE 电路与电子 [50 核心学时]	CE-CAL 算法 [30 核心学时]
CE-CAE-1 历史沿革和综述 [1]	CE-CAL-1 历史沿革和综述 [1]
CE-CAE-2 相关工具、标准和(或)工程限制 [3]	CE-CAL-2 相关工具、标准和(或)工程限制 [1]
CE-CAE-3 电气量和基本元件 [4]	CE-CAL-3 基本算法分析[4]
CE-CAE-4 电力电路 [11]	CE-CAL-4 算法策略[6]
CE-CAE-5 电子材料,二极管和双极型晶体管 [7]	CE-CAL-5 常见任务的经典算法[3]
CE-CAE-6 MOS 晶体管电路,时序和功率 [12]	CE-CAL-6 分析和设计面向特定应用的算法 [6]
CE-CAE-7 存储单元结构 [3]	CE-CAL-7 并行算法和多线程[6]
CE-CAE-8 接口逻辑 [3]	CE-CAL-8 算法复杂性[3]
CE-CAE-9 运算放大器 [3]	CE-CAL-9 调度算法
CE-CAE-10 混合信号电路设计 [3]	CE-CAL-10 基本的可计算性理论
CE-CAE-11 设计参数和问题	
CE-CAE-12 电路建模与仿真方法	

续表

CE-CAO 计算机体系结构和组成 [60 核心学时] CE-CAO-1 历史沿革和综述 [1] CE-CAO-2 相关工具、标准和(或)工程限制[1] CE-CAO-3 指令集体系结构[10] CE-CAO-4 性能评测[3] CE-CAO-5 计算机算术[3] CE-CAO-6 处理器组成[10] CE-CAO-7 存储器系统的组织和体系结构[9] CE-CAO-8 输入/输出接口和通信[7] CE-CAO-9 外围子系统[7] CE-CAO-10 多核/众核体系结构[5] CE-CAO-11 分布式系统体系结构[4]	CE-DIG 数字设计 [50 核心学时] CE-DIG-1 历史沿革和综述 [1] CE-DIG-2 相关工具、标准和(或)工程限制 [2] CE-DIG-3 数字系统和数据编码[3] CE-DIG-4 布尔代数应用[3] CE-DIG-5 基本逻辑电路[6] CE-DIG-6 组合电路的模块化设计[8] CE-DIG-7 时序电路的模块化设计[9] CE-DIG-8 控制和数据通路设计[9] CE-DIG-9 可编程逻辑设计[4] CE-DIG-10 系统设计和约束[5] CE-DIG-11 故障模式、测试和可测性设计
CE-ESY 嵌入式系统 [40 核心学时] CE-ESY-1 历史沿革和综述 [1] CE-ESY-2 相关工具、标准和(或)工程限制 [2] CE-ESY-3 嵌入式系统的特点[2] CE-ESY-4 嵌入式应用的基本软件技术[3] CE-ESY-5 并行输入输出[3] CE-ESY-6 异步串行通信[6] CE-ESY-7 周期性中断,波形产生,时间测量[3] CE-ESY-8 数据采集,控制,传感器,执行器[4] CE-ESY-9 复杂嵌入式系统的实现策略[7] CE-ESY-10 低功耗操作技术[3] CE-ESY-11 移动和网络化嵌入式系统[3] CE-ESY-12 高级输入/输出主题[3] CE-ESY-13 嵌入式系统计算平台	CE-NWK 计算机网络 [20 核心学时] CE-NWK-1 历史沿革和综述 [1] CE-NWK-2 相关工具、标准和(或)工程限制 [1] CE-NWK-3 网络体系结构[4] CE-NWK-4 局域网和广域网[4] CE-NWK-5 无线和移动网络[2] CE-NWK-6 网络协议[3] CE-NWK-7 网络应用[2] CE-NWK-8 网络管理[3] CE-NWK-9 数据通信 CE-NWK-10 性能评估 CE-NWK-11 无线传感器网络

续表

<p>CE-PPP 职业素养和专业实习 [20 核心学时]</p> <p>CE-PPP-1 历史沿革和综述 [1]</p> <p>CE-PPP-2 相关工具、标准和(或)工程限制 [1]</p> <p>CE-PPP-3 有效沟通策略 [2]</p> <p>CE-PPP-4 跨学科团队方法 [1]</p> <p>CE-PPP-5 哲学框架与文化问题 [2]</p> <p>CE-PPP-6 工程解决方案和社会影响 [2]</p> <p>CE-PPP-7 职业与道德责任 [3]</p> <p>CE-PPP-8 知识产权与法律问题 [3]</p> <p>CE-PPP-9 当代问题 [2]</p> <p>CE-PPP-10 经营和管理问题 [3]</p> <p>CE-PPP-11 职业素养中的折衷</p>	<p>CE-SEC 信息安全 [20 核心学时]</p> <p>CE-SEC-1 历史沿革和综述 [2]</p> <p>CE-SEC-2 相关工具、标准和(或)工程限制 [2]</p> <p>CE-SEC-3 数据安全性和完整性 [1]</p> <p>CE-SEC-4 脆弱性:技术和人为因素 [4]</p> <p>CE-SEC-5 资源保护模型 [1]</p> <p>CE-SEC-6 私密密钥和公开密钥加密 [3]</p> <p>CE-SEC-7 消息认证码 [1]</p> <p>CE-SEC-8 网络与 Web 安全 [3]</p> <p>CE-SEC-9 认证 [1]</p> <p>CE-SEC-10 可信计算 [1]</p> <p>CE-SEC-11 侧信道攻击 [1]</p>
<p>CE-SGP 信号处理 [30 核心学时]</p> <p>CE-SGP-1 历史沿革和综述 [1]</p> <p>CE-SGP-2 相关工具、标准和(或)工程限制 [3]</p> <p>CE-SGP-3 卷积 [3]</p> <p>CE-SGP-4 变换分析 [5]</p> <p>CE-SGP-5 频率响应 [5]</p> <p>CE-SGP-6 采样与混叠 [3]</p> <p>CE-SGP-7 数字光谱和离散变换 [6]</p> <p>CE-SGP-8 有限和无限脉冲响应滤波器设计 [4]</p> <p>CE-SGP-9 窗函数</p> <p>CE-SGP-10 多媒体处理</p> <p>CE-SGP-11 控制系统理论和应用</p>	<p>CE-SPE 系统与项目的工程化 [35 核心学时]</p> <p>CE-SPE-1 历史沿革和综述 [1]</p> <p>CE-SPE-2 相关工具、标准和(或)工程限制 [3]</p> <p>CE-SPE-3 项目管理 [3]</p> <p>CE-SPE-4 用户体验 [6]</p> <p>CE-SPE-5 风险,可靠性,安全性和容错性 [3]</p> <p>CE-SPE-6 硬件和软件过程 [3]</p> <p>CE-SPE-7 需求分析及其获取 [2]</p> <p>CE-SPE-8 系统规约 [2]</p> <p>CE-SPE-9 系统架构设计与评价 [4]</p> <p>CE-SPE-10 并发硬件和软件设计 [3]</p> <p>CE-SPE-11 系统集成,测试和验证 [3]</p> <p>CE-SPE-12 可维护性、可持续性、可制造性 [2]</p>

续表

CE-SRM 系统资源管理 [20 核心学时]	CE-SWD 软件设计 [45 核心学时]
CE-SRM-1 历史沿革和综述 [1]	CE-SWD-1 历史沿革和综述 [1]
CE-SRM-2 相关工具、标准和(或)工程限制 [1]	CE-SWD-2 相关工具、标准和(或)工程限制 [3]
CE-SRM-3 管理系统资源[8]	CE-SWD-3 编程结构和范式[12]
CE-SRM-4 实时操作系统设计[4]	CE-SWD-4 问题求解策略[5]
CE-SRM-5 移动设备的操作系统[3]	CE-SWD-5 数据结构[5]
CE-SRM-6 并发处理的支持[3]	CE-SWD-6 递归[3]
CE-SRM-7 系统性能评估	CE-SWD-7 面向对象程序设计[4]
CE-SRM-8 虚拟化支持	CE-SWD-8 软件测试、验证和确认 [5]
	CE-SWD-9 数据模型[2]
	CE-SWD-10 数据库系统[3]
	CE-SWD-11 事件驱动和并发程序设计[2]
	CE-SWD-12 使用应用程序接口(API)
	CE-SWD-13 数据挖掘
	CE-SWD-14 数据可视化

#### CE-ESY-5 并行输入和输出[3]

表示“并行输入和输出(parallel input and output)”知识单元的相对重要性可以用3个核心学时来衡量,它属于“嵌入式系统(embedded systems, ESY)”知识领域中的第5个知识单元,是计算机工程专业培养方案中的核心单元。如果在条目中没有最后的数字(如[2])则说明这个知识单元不属于核心部分,而是附加的可选部分。需要注意的是,CE2016课程指导委员会采用了整五整十的原则近似地设置了这些知识领域的学时数,以便为灵活设置计算机工程方向的不同培养方案。附录A给出了每个知识领域以及所包含的知识单元。

### 3.3.1 相关的数学基础

表3.3描述了计算机工程知识体系的数学部分。CE2016课程指导委员会建议,要培养一个四年以后(2020年代)具有职业竞争力的计算机工程师,那么计算机工程培养方案中,至少应该包含4个数学方面的知识领域,并且至少需要120个学时。显然,典型的培养方案中,大多都具有更多数学知识,以达到其培养目标。这四个领域是:连续函数分析(微积分)、离散结构、线性代数以及概率论和数理统计。CE2016课程指导委员会认为,这4个领域对于计算机工程来说是最必要的数学基础。

表 3.3 计算机工程相关的数学基础  
(核心学时:120)

数学知识领域和知识单元	数学知识领域和知识单元
<b>CE-ACF 连续函数分析</b> [30 核心学时] CE-ACF-1 历史沿革和综述 [1] CE-ACF-2 相关工具、标准和工程约束 [1] CE-ACF-3 微分法 [4] CE-ACF-4 积分法 [6] CE-ACF-5 线性微分方程 [8] CE-ACF-6 非线性微分方程 [3] CE-ACF-7 偏线性微分方程 [5] CE-ACF-8 函数序列 [2]	<b>CE-DSC 离散结构</b> [30 核心学时] CE-DSC-1 历史沿革和综述 [1] CE-DSC-2 相关工具、标准和工程约束 [1] CE-DSC-3 函数、关系和集合 [6] CE-DSC-4 布尔代数原理 [4] CE-DSC-5 一阶逻辑 [6] CE-DSC-6 证明技术 [6] CE-DSC-7 计数基础 [2] CE-DSC-8 图和树的表示及属性 [2] CE-DSC-9 迭代和递归 [2]
<b>CE-LAL 线性代数</b> [30 核心学时] CE-LAL-1 历史沿革和综述 [1] CE-LAL-2 相关工具、标准和工程约束 [2] CE-LAL-3 基, 向量空间, 正交性 [4] CE-LAL-4 线性系统的矩阵表示 [4] CE-LAL-5 矩阵求逆 [2] CE-LAL-6 线性变换 [3] CE-LAL-7 线性系统解 [3] CE-LAL-8 非线性系统的数值解 [4] CE-LAL-9 系统转换 [3] CE-LAL-10 特性系统 [4]	<b>CE-PRS 概率论与数理统计</b> [30 核心学时] CE-PRS-1 历史沿革和综述 [1] CE-PRS-2 相关工具、标准和工程约束 [2] CE-PRS-3 离散概率 [5] CE-PRS-4 连续概率 [4] CE-PRS-5 期望与偏差 [2] CE-PRS-6 随机过程 [4] CE-PRS-7 抽样分布 [4] CE-PRS-8 评估 [4] CE-PRS-9 假设检验 [2] CE-PRS-10 相关与回归分析 [2]

### 3.3.2 相关的科学基础

CE2016 课程指导委员会决定不推荐特定的科学领域,也不对培养方案中应有多少学时来学习自然科学提出建议。不过,我们强烈建议计算机工程专业的学生学习尽可能多的自然科学(如生物、化学和物理等)知识,来保证有一个坚实的科学基础。推荐科学基础的理由是,在工程领域的学生应该具有很强的分析问题能力,以及从实验和经验中学会知识的能力。有关计算机工程学生的科学素养问题的详细讨论见第六章。

### 3.3.3 软件的作用

尽管所有计算机工程师都需要熟悉软件开发和实现,但本知识体系中并没有指定特定的编程语言和操作系统。知识体系中的学习成效强调了更高级别的设计概念以及软、硬件之间的相互作用。一些计算机工程师可能要开发操作系统、编译器和其他软件工具,不过,本专业最关注的还是在系统设计中运用这些工具来满足特定的需求。根据其培养目标以及学生基础的不同,一个偏学术方面的培养方案可能需要包含一些在本知识体系之外的额外软件知识和技能(如操作系统设计),以培养出具有竞争力的软件工程师。

## 3.4 CE2016 知识体系与 CE2004 知识体系比较

表 3.4 知识体系:CE2016 知识领域与 CE2004 知识领域对比

CE2016 知识领域	学时	CE2004 知识领域	学时
CE-CAE 电路与电子	50	CE-CSG 电路与信号	43
CE-CAL 算法	30	CE-ALG 算法	30
CE-CAO 计算机体系结构与组成	60	CE-CAO 计算机体系结构与组成	63
CE-SPE 系统与项目的工程化	35	CE-CSE 计算机系统工程	18
CE-DIG 数字设计	50	CE-DIG 数字逻辑	57
CE-ESY 嵌入式系统	40	CE-ESY 嵌入式系统	20
CE-NWK 计算机网络	20	CE-NWK 计算机网络	21
CE-PPP 职业素养/专业实习	20	CE-SPR 社会与职业素养	16
CE-SEC 信息安全	20		
CE-SGP 信号处理	30	CE-DSP 数字信号处理	17
CE-SRM 系统资源管理	20	CE-OPS 操作系统	20
CE-SWD 软件设计	45	CE-SWS 软件工程	13
		CE-DBS 数据库系统	5
		CE-ELE 电子	40
		CE-HCI 人机交互	8
		CE-PRF 程序设计基础	39
		CE-VLS VLSI 设计和光纤技术	10
总计	420	总计	420

CE2004 报告中推荐的知识体系包含有 420 个计算机工程的核心学时,以及 66 个数学方面的学时,总共 486 个核心学时。CE2016 报告提出的知识体系中包含有

420 个计算机工程的核心学时,加上 120 个学时的数学,共 540 个核心学时。表 3.4 列出了 CE2016 和 CE2004 在知识领域及对应的学时数的比较。

如表 3.4 所示,CE2016 知识体系有 12 个知识领域,而 CE2004 则有 16 个知识领域。直观的印象是 CE2016 的知识体系在内容上进行了缩减。其实不是这样的。CE2016 课程指导委员会将 CE2004 知识体系中的 5 个知识领域进行了合并融合,因而减少了 CE2016 知识体系中的知识领域。例如,VLSI 设计的趋势是强调片上系统(SOC)技术,因而被合并到了嵌入式系统中。表 3.2 所示的 CE2016 知识单元和附录 A 都显示了这样一种融合方式。

### 3.5 计算机工程设定核心学时数的依据

正如前面所述,CE2016 知识体系包含 420 个核心学时。此外,CE2016 的数学知识包含 120 个核心学时。在 3.1.2 节中,我们给出了核心学时的定义,1 个核心学时代表 1 个 50 分钟的授课课时。

在典型的大学教学环境下,一个四年期的本科培养方案至少有 120 个学期学时或学分。一学年由两个学期组成,一共包含 30 个学期学时,因而每个学期有 15 个学期学时(学分)。我们把一个学分(或学期学时)定义为等价于 14 周 $\times$ 1 学时/周(一学时指 50 分钟的一门课程)的课程教学加上一个考试周。所以,一年 30 个学分的学习相当于  $30 \times 14 = 420$  个学时。

CE2016 课程指导委员会相信 420 个核心学时反映了一个有竞争力的计算机工程师在本科阶段所必须的最低学时数要求,通常,大多数培养方案将会选择包含更多的计算机工程方面的教学内容。另外,一个计算机工程专业的学生在数学方面应该至少有 120 个学时的学习经历。我们认为在数学方面经历 120 个核心学时的学习对于培养一个具有竞争力的计算机工程师是不够的,因此,列入数学基础的知识单元仅仅是那些在计算机工程领域中涉及的最基本的数学知识及技能。培养方案毕业生应该掌握数学基础知识的宽度和广度可以由具体的培养单位和培养方案来决定,对这个问题的详细讨论参见第六章。

### 3.6 课程体系模型

在前面章节的讨论中为计算机工程培养方案提供了相关知识体系的一个最低要求框架。为了培养将来在计算机工业界或继续深造中更有能力和更有竞争力的计算机工程师,一个更高要求的培养方案可能会包含更多的数学、科学和计算机工程方面的内容。尽管我们提出了计算机工程专业毕业生应该具备的 420 个核心学时知识体系以及应该达到的学习成效,但是,并不是所有计算机工程的培养方案都

能包含所有的内容,这主要取决于培养方案的目标以及所在学校的定位和所拥有的资源。当然,我们期望的是好的计算机工程培养方案应该要达成所有的学习成效。

一个三年的课程体系包含半年的数学和科学、一年的计算机工程核心课,一年的计算机工程选修课,以及半年的通识课和其他工程方面的学习。这个模型对于那些大学前就选修过通识课、数学和科学的情况比较有用。一个四年的课程体系包含一年的数学和科学,一年的计算机工程核心课,一年的计算机工程选修课,一年的通识课和其他工程方面的学习。这个模型适合许多世界范围内的大学学制。有关课程体系的讨论和具有代表性的样板课程体系将在第 6 章和附录 B 中出现。



## 第四章 工程实践和计算机工程课程体系

从本质上说,计算机工程课程体系应该反映出一种工程精神,并一以贯之地渗透到课程的历年教学中。这种方法有助于引领学生进入工程(这里是计算机工程)领域,教导他们以工程师的立场去思考、行动和规划未来。实践的准备是必不可少的,因为当今世界上大多数经历四年制培养方案的毕业生直接开始了他们的专业生涯。

计算机工程(CE)领域已经发展成熟,遍及整个工业界。在广泛且多样化的领域,如商业、工业、政府、服务业、社会团体以及那些利用计算机有效自动化或驱动其产品和服务的各类领域中,计算机工程专业人员都在应用其技能。

计算机工程的从业者具有巨大的成功潜力。美国劳工统计局(BLS)最近的一项研究表明,到2022年,计算机职位将增加17.7%,信息安全将以36.5%居首位[BLS]。不幸的是,虽然目前和将来存在这么多职位,但找到合格的人来填补它们则往往很困难。获得工程(如计算机工程)学位的毕业生并非具备行业需要的所有素质。他们或许具备从学习中获得的技术技能,但缺乏可“契合”工业界或政府环境的其他技能。

从四年制大学毕业的学生认为其学士学位是得到一份职位足够的资格。这种理解在某些领域可能是对的,但在计算机工程领域却行不通。这种虚幻的信念给世界各地的求职者都造成了许多麻烦。学位证书可能是必要条件,但不是充分条件。一个普遍的共识存在于工程和其他领域,即一个成功的专业人士必须是一个具备良好沟通能力的人,一个强大的团队合作者和一个胸怀成功激情的人。因此,具备学位并不一定能够确保就业。

有些人甚至认为,一个具有较高平均绩点(GPA)的计算机工程专业的毕业生会比一个具有较低GPA的毕业生更有可能获得职位。这是另一个虚幻的认识。具有很高GPA毕业生是难能可贵的。但是,如果没有激情和动力,不能与他人合作,或缺乏良好的沟通能力,又没有必要的个人技能,该人将可能难以通过第一轮面试。

这一章的作用是超越第三章介绍的知识体系,考证计算机工程专业毕业生在真实场景下运用这一知识体系解决实际问题所必需的基本技能。第5章将讨论职业素养一重要问题,第六章将考虑整体的课程设计,并介绍附录B中给出的示例课程实施情况。

## 4.1 计算机工程探源

工程精神重要也是最本质的方面就是获取必要的背景知识来理解和推导工程概念及作品。这种背景知识源于计算、电子、数学和物理等领域的基本思想。计算机工程知识体系的重要作用之一是揭示和发展这些基本概念。知识体系的核心通过多种方式反映出我们是如何精心选择材料来达到这一目的的。

这些基本材料将成为其他材料的基础,最终的表现是建立更好或新的基于计算机的系统。理论联系实际,并以理论指导实践,似乎是这一学科的最佳途径。课程体系应该为这种融合补充一组专业的、道德的和法律的关注点,以此引导这些受过良好教育的计算机工程师的行为和态度。课程体系也应促进他们熟悉广泛而多样化的应用。

## 4.2 新兴技术策略

近来,计算和工程领域发生了迅速的变化。而且,大家也广泛认为,这些领域的变化将在未来大幅加快。因此,计算机工程师必须具备必要的的能力,以敏捷的方式适应崭新的和新兴的技术。他们应该能够识别新技术的贡献者,并能鉴别那些因不能适应不断变化的领域而失败的公司。所以,计算机工程师至少应该意识到新兴技术的发展所带来的积极和消极的后果。

那么,计算机工程师应该如何与一个新兴技术的时代联系呢?一种方法是确定与这些技术有关的利益相关者,并辨识一些与这些新领域的发展和用相关的战略性前提和社会价值。通常情况下,工业界会打破学术的障碍来规划这样的战略前提;有时政府会制订政策来扩展或限制这些战略,甚至制订标准来规范新兴技术。这些战略可能包含应用技术,有些也可能停留在概念上。

### 4.2.1 应用型新兴技术

计算机工程师应该关注应用型新兴技术。这些技术已经存在于市场,但由于太新以至于它们对社会的影响还不完全清楚。学生应该能够识别一些应用的新兴技术,并指出他们对计算机工程的影响。对于培养在一个不断变化的世界中仍可做出专业贡献的称职工程师来说,辨识和理解应用型新兴技术对计算机工程的影响是有用的。

计算机工程课程体系应允许探索应用型新兴技术。例如,教师可鼓励学生去探究3D打印机是否会制作出危害社会的物品,或描述在设计 and 生产集成电路时面临的挑战。作为另一个例子,学生应该能够解释纳米技术或物联网(IoT)将改变技术

工作环境的各种方式。计算机工程师已经或即将与光学、生物或量子计算机打交道,他们或将为制造业设计一种新型的机器人系统。这些新兴且现代的技术给计算机工程专业的学生和从业者提出了挑战,可能涉及金融和道德的权衡,直接影响着一个不断变化的世界中的专业实践。

### 4.2.2 概念型新兴技术

计算机工程师也应该关注到概念型的新兴技术。这些技术还处于发展中状态,刚刚找到或即将出现进入市场的机会。学生应该能够识别一些概念上的新兴技术,并指出它们对计算机工程的影响。

计算机工程课程体系应该允许探索那些尚未成为可行技术的新发明。例如,教师应鼓励探索各种新的途径,譬如计算机工程师设计带有增强现实和虚拟世界的环境,或是可能影响计算机工程师工作的大数据和数据分析。这些行为将影响像计算生物学和生物信息学这些新兴领域中的设计约束。此外,让学生探索计算机工程师在机器学习和智能系统时代的作用,或讨论在开发绿色计算和可持续发展的文化方面所需要的工程策略是有益的。新技术甚至可能暴露影响计算机工程领域的安全问题。对这些和其他一些问题的认识对培养一个全面和具有社会意识的计算机工程师来说是十分重要的。

## 4.3 课程体系中的设计

本报告的第二章简要讨论了一个计算机工程师的特点,如设计能力,并提供了一个工程设计的定义。以下部分给出一个参考,指导如何将设计融入计算机工程课程体系之中。

### 4.3.1 贯穿课程体系的设计

工程设计的原则必须渗透到整个计算机工程课程体系,以造就合格的毕业生。贯穿整个的教育过程,计算机工程专业的学生应该遇到不同的设计方法,使他们对这些方法的长处和短处了然于胸。通常情况下,设计时遇到的各种情况就提供了一个框架,决定了应该做出什么样的选择。根据特定应用的需求,设计时可能会强调技术因素、可靠性、安全性、成本、用户界面,也可能关注其他问题。必要的设计技能的培养不可能通过一个单一的课程来实现,而必须在整个课程体系范围里整合,建立在学生对技术知识的积累和先前设计经验的基础上。

对计算机工程师来说,特别值得关注的一点是软件/硬件接口,其困难的权衡抉择往往成为工程性挑战。在这个边界上的考虑将导致对计算机体系结构以及对机器代码重要性的评价和洞察。在这个接口或边界上,会产生硬件/软件权衡的困难

抉择,这又自然地引发了专用计算机和系统的设计。例如,在设计一个安全攸关的系统时,重要的是要确保系统不会对用户或公众造成伤害。计算机工程师必须彻底地测试,哪怕是对那些不太可能的参数、硬件、软件直至系统本身,以确保系统正确可靠地运行。

在不同的层次上充满着软件设计中的难点问题,包括人机界面。全面解决这些问题可能会涉及多媒体、图形学、动画和一系列技术的考虑。同样地,在硬件设计上也会有同样的说法。总之,设计是计算机工程的核心。

### 4.3.2 毕业设计经历

毕业设计项目的概念得到了广泛的重视,被认为是学生完成课程体系过程中的一个重要经历。学生思考本学科的重要问题,在解决这个问题的过程中,他们有机会展示其系统化地运用工程原理来产生解决问题的能力。对于计算机工程,解决方案通常涉及一个系统或子系统,包含硬件和软件组件,并考虑各种相互作用和权衡的设计和实施。设计经历往往以团队为基础,这最能体现行业实际。理想情况下,设计经历应包括工程标准和现实的约束,以代表一个真实的环境中的各种可能情况。

毕业设计经历应该为学生提供丰富的学习效益。来自于这种经历的好处包括:

- 参与整个系统的结构和功能,以及对相关的道德和社会影响因素的考虑;
- 展示整合若干不同学科的概念到完整的系统,形成一个解决问题的能力;
- 展示运用计算机工程相关工具和原理的能力;
- 能编写表达清晰规范的书面文档(及其他材料),详细介绍设计和设计过程;
- 展示创意和创新;
- 提高时间管理和计划方面的能力;
- 通过对成果的评估带来的自我成就感。

一个培养方案可以有多种途径发现学生已获得了某项专门的知识技能的证据。这可能包括一个演示、一个展示报告、一个口试、一个网站或者行业评价。虽然没有罗列在核心知识体系之中,毕业设计经历必须是本科经历中的一个组成部分。在毕业设计之前已经获得的许多核心学习成效,有些会在毕业设计期间得到很好的运用,这也是终身学习必要性的有力证明。

## 4.4 实验经历

实验经历是计算机工程课程体系的重要组成部分,它们具有多种功能。正如其它任何工程课程一样,计算机工程专业的学生有许多机会去观察、探究和控制实际设备、系统和程序的特性与行为。这包括对硬件和软件进行设计、实现、测试和编制

文档,设计实验来获取数据,分析和解释这些数据,并使用这些数据来纠正或改进设计,验证设计符合需求。

初级实验(例如,编程实验)通常是有针对性的,旨在强化课堂和课后作业中的概念。这类实验展示特定的现象或行为,为实验者维持某种特定的属性,以供他们观察或者测量。中级和高级实验应包括更多的开放式的问题,要求学生设计和实现解决方案,或要求他们设计实验,获取完成设计或测量各种特性所需的数据。

本报告的目的不在于推荐什么是吸引学生参与实验的最好方法。每个培养方案都应依据可用的场所、课程目标和可用的资源寻找最适合他们需求的配置方案。

#### 4.4.1 计算机工程实验

计算机工程领域的许多课程都应该包含一些实验经历。通常情况下,一个实验宜持续 2 ~3 个学时,可以在一个配备有专门设备的房间或场地进行。这些实验的深度和广度在不同学校将有所不同。通常由于课程体系分配给实验的学时数、物理空间和资源的不同而得到不同的效果。表 4.1 给出了 CE2016 课程指导委员会的建议,即计算机工程类的学生应该完成的实验。一些实验所有的计算机工程专业的学生必须或应该去完成。其中的标记为“••••”的实验指的是每一个计算机工程师必须经历的,而标记为“••”的实验则是建议每一个计算机工程师应该完成的。附加实验的实施取决于各个学校培养方案的要求、目标和资源。表 4.1 列出了一些常见实验的类型。附录 C 给出了除高级课程设计实验之外的这些实验典型的描述、配置和实施方案,高级课程设计实验需要反映每个培养方案的具体需求和使命。

表 4.1 计算机工程实验的类型

实验类型	必要实验	建议实验	附加实验
电子学和电路	••••		
计算机体系结构设计			•
数字信号处理			•
数字信号和系统设计	••••		
嵌入式系统	••••		
工程学导论			•
网络		••	
软件设计		••	
高级课程设计	••••		

表 4.2 给出了培养方案中可以考虑的一些额外的相关实验。

表 4.2 建议性的附加计算机工程实验

音频工程	微波测量
制造业中的计算机应用	操作系统
电力能源系统	机器人
图形学	专业的电子学实验
机电一体化	教学强化训练
	通信

实验应包括一些设计的真实实现,如电子和数字电路、面包板、FPGA/CPLD、基于微控制器的系统、原型机以及固件实现。实验还应包括数字设计和计算机系统的应用程序和仿真软件。仿真工具的使用体现了它们作为专业的计算机工程实践一部分的内在外在价值。它们对建模和研究实际系统非常有用,而为了让学生研究那些在给定的时间和资源条件下难以设计和实现的系统,这些仿真工具往往是可取和必需的。此外,配置管理和版本控制软件也可在实验中用于硬件和软件的管理。

学生应学会记录实验活动,书面记录和跟踪所有的设计活动、完成的实验步骤、测量或观察到的结果,无论这些结果是好是坏。这也提供了记录并探讨设计上采用的对实验参数进行的权衡以及这些权衡所造成的影响的机会。实验经验也应该有助于学生学会处理一些实际问题,如:

- 所有实验,尤其是电子设备和电器设备发生危险的场合的安全问题;
- 正确使用计算机和测试设备;
- 搭建电气或电子线路及系统;
- 理解与产品开发和制造相关的过程和问题;
- 把握实验条件和参数取舍的时机(如:硬件、软件、现成的 IP 核、时间/空间、电源、模块化、可修改性、安全性、效率)并能够在这些方面做出明智的决定。

在他们教育的形成阶段,学生往往受到工程动手实践特性的吸引。实验恰恰充分利用了这一兴趣为实践活动的其他重要要素奠定了基础。从根本上来讲,实验中精心策划的实验步骤和任务能够帮助学生培养对他们技术能力的信心。实验还应该帮助学生提高在制造新产品等方面所需的专业知识以及对实验技术人员、现场协助人员以及其他学科专业人员的重要作用表示尊重和感谢。

#### 4.4.2 软件

与计算机工程相关的软件包将根据每一个培养方案的理念和需求而有所不同。表 4.3 列举了一些软件,可以出现在特定的实验场景的所有机器上。表中所列出的

产品仅仅是用于示例的目的,而且仅代表本报告写作时的选择,并没有隐含推荐的含义。此外,也并不建议任何一个培养方案都需要包括所有这些软件应用。每一个培养方案的组织单位应该确定自身的需要,纳入这些应用的最新版本。

表 4.3 推荐的软件应用

设计模型和仿真	软件开发	通用的计算软件/产品
<ul style="list-style-type: none"> <li>• 电路级(如:SPICE)</li> <li>• 门级(原理图输入)</li> <li>• 数字系统级(如:VHDL、Verilog)</li> <li>• 模拟/混合信号电路(如:VHDL-AMS、Verilog-AMS)</li> <li>• 系统级设计(如: System Verilog、System C)</li> </ul>	<ul style="list-style-type: none"> <li>• 集成开发环境(IDE)               <ul style="list-style-type: none"> <li>■ 设计入口/管理</li> <li>■ 库函数支持</li> </ul> </li> <li>• 编译器(如: C、C++、C#、Java、Python)</li> <li>• 操作系统支持</li> <li>• 源代码级排错</li> <li>• 平台支持(如:智能手机、平板电脑)</li> </ul>	<ul style="list-style-type: none"> <li>• Web 浏览器</li> <li>• Email</li> <li>• 办公软件</li> <li>• PDF 阅读器/编辑器</li> <li>• 图表/照片 浏览器/编辑器</li> <li>• 多媒体播放器/编辑器</li> <li>• 文件压缩/解压缩</li> <li>• 文件传输(FTP、SCP、SFTP)</li> <li>• 终端模拟器、远程登录、安全 shell、X Window 客户端</li> </ul>
<p><b>数字硬件原型设计</b></p> <ul style="list-style-type: none"> <li>• FPGA/CPLD 开发工具</li> <li>• 设计文件输入和管理</li> <li>• 组件/IP 库函数支持</li> <li>• 设备的编程实现</li> <li>• 交互式排错</li> </ul>	<p><b>集成电路/ASIC 设计</b></p> <ul style="list-style-type: none"> <li>• 设计验证和仿真</li> <li>• 综合</li> <li>• 物理布图               <ul style="list-style-type: none"> <li>■ 物理布图编辑器</li> <li>■ 自动布图</li> </ul> </li> <li>• 设计确认(设计规则、布图 vs. 原理图、参数提取)</li> <li>• 可测试性设计,测试用例自动生成</li> </ul>	<p><b>系统工程工具</b></p> <ul style="list-style-type: none"> <li>• 项目管理(如: GANTT 或 PERT 图)</li> <li>• 需求规格管理(如: UML 工具)</li> </ul>
<p><b>微控制器系统设计</b></p> <ul style="list-style-type: none"> <li>• 集成开发环境(IDE)               <ul style="list-style-type: none"> <li>■ 设计入口/管理</li> <li>■ 库函数支持</li> </ul> </li> <li>• 编译器、汇编器、链接器</li> <li>• 处理器模拟/仿真</li> <li>• 设备的编程实现</li> <li>• 电路级测试/排错</li> </ul>	<p><b>印刷板电路(PCB)设计</b></p> <p><b>计算机辅助设计和建模(CAD 工具)</b></p>	<p><b>其他工具</b></p> <ul style="list-style-type: none"> <li>• 机器人软件开发</li> <li>• 半导体设备和处理模型(如: TCAD)</li> <li>• 微波、RF 及其他高速/高频设计</li> <li>• 电磁场仿真</li> <li>• MEMS 设计</li> </ul>
<p><b>数学工具包</b></p> <ul style="list-style-type: none"> <li>• 问题求解</li> <li>• 数据分析</li> <li>• 建模和仿真</li> </ul>	<p><b>实验室自动化与仪器(如: IntuiLink 或 LabVIEW software)</b></p>	

### 4.4.3 开放式实验

计算机工程课程体系往往包含开放式的实验,进行真正的研究和开发。人们可以把这看作“终极的实验”。毕业设计或毕业实践通常具有这种开放式的特点。在这种场合,一位教师和一个学生团队共同选定一个研究领域。一旦确定,学生团队便启动研究和设计过程。培养方案通常提供了一个现代化的场所用于团队会面和在工作。这些场所或者高级设计实验室,一般都包含现代化设施,也为电子设备(如机器人)和正在进行的项目所需的其他器材提供足够的空间。

### 4.4.4 随堂实验

传统的实验通常在一个与授课不同的地点和时间下完成。例如,图形学课程可能在星期二和星期四上课,而另外在星期三安排3个学时的实验。尽管如此,在现代教育环境中,将实验嵌入常规授课是很常见的。一种做法是在实验的过程中完成讲授。另一种做法,是按照某种方式将教室划分出几个区域,在一个区域中授课,而把实验和其相关的设备放到另一个区域中。还有一种新的趋势就是所谓的“翻转课堂”,学生事先看完讲课录像,教学课时用来让学生进行主动学习,可以是学生把个人的面包板和仪器带到教室/实验室,完成实验练习。

### 4.4.5 技术支持

一个培养方案需要可以随时访问技术支持,以开发“可工作的实验”环境。例如,要为实验设计制作一个新的天线,就可能需要在某个带有机床和焊接工具的技术工作区,使用现有的设施来完成。技术支持或许配备3D打印机、各种标准的电路元件和芯片以及通用的微控制器或者FPGA开发板等工具。

### 4.4.6 学生自行采购

另一个趋势是让学生自行采购实验室设备,或取得廉价甚至是免费的专供软件工具。一些设备的成本已非常低廉,特殊工具包的尺寸也减少到一个简单的工具箱,学生已可以购买自己的软件和实验装置带到实验课上(系里应该保存一定数量的备件,以备学生不时之需)。这种做法的好处是让学生超越正式的实验室环境进行实验。学生可自行采购的东西包括:

- 可连接一台装有对应操作台软件的笔记本电脑的个人仪器设备(如:示波器、万用表、逻辑分析仪、波形发生器等);
- 用于实现电路的带有电源的面包板;
- 廉价的微控制器和FPGA开发板;
- 学生版的建模、仿真和软件开发工具包。



## 4.5 工程工具的作用

工具的使用是工程的基础,可有效地组织信息和管理设计的复杂性。熟悉常用的工具,在适当的情况下部署它们,并有效地使用它们的能力都是重要的技能。在快速变化的计算机工程世界中,总会存在新工具及其应用发挥作用的机会。开发和利用高质量的工具是计算机工程师工作的一部分。

对于计算机工程师来说,相关的工具范围跨越了整个硬件和软件领域。硬件设计和分析工具,包括有测量和分析硬件行为的仪器、用于建模和仿真电路及系统的集成电路专用模拟程序、硬件描述语言;自动完成各种设计步骤的工具,包括集成、测试模式生成及物理实现(例如,FPGA 映射-布局-布线),模拟器和调试工具。其他硬件设计工具还包括支持电路设计、印刷电路设计布局、电路行为分析、方框图创建和编辑、通信系统建模、混合模拟和数字仿真建模、设计规则检查和虚拟仪器等。

软件设计和分析工具包括操作系统、版本控制系统、集成开发环境 (IDE), GUI 设计器、语言处理程序、交互式调试器以及计算机辅助软件工程 (CASE) 工具。一般支持工具包括算法开发和数学分析的环境(如 MATLAB 和 MathCad),办公软件(文字处理、电子表格、浏览器和搜索引擎),数据库软件,通信软件和项目管理工具。

并不是说每一个计算机工程培养方案都必须采纳所有这些工具。各个培养方案应根据其培养目标和目的在方案的实施过程中选择适当的工具。了解工具的局限性,并在总体上确定工具和组件的开发范围,是计算机工程师的另一项本领。随之而来的接续工作是从事这些项目的设计和开发。该活动需要受到诸如安全性、可用性、可靠性、机密性等各种形式的质量属性的约束。

## 4.6 计算机工程原理的应用

对于计算机工程的特质以及学生学习这些课程的期望,应用都将发挥基础性作用。教师可以将应用作为手段,达到下面的目的:

- 激发学生的学习积极性;
- 引导学生的思考和志向;
- 为某些材料的内在含义或突出特点提供合理性解释;
- 给出理论的实际应用。

培养方案可以通过各种可能的途径来达成这些特性。这些措施包括使用最新的和专题性的案例研究、导读、现场参观、邀请工业界人士演讲等多重方式。也可以在不同层面上来进行,包括芯片设计,软件工具开发,直至整个系统的开发。适当的应用也可以为团队活动创造一种环境,比如一个跨学科性质的论坛。为此,计算机

工程专业的所有学生都应该投身于一些深入的研究中,开展以计算机工程为实质性内容的活动。

计算机工程专业的学生会有广泛的兴趣和专业目标。对于许多学生而言,深入学习计算机工程的某些方面将是非常有用的。学生们可用几种方法完成这类工作。一些方法可能包括扩展的实习经历或相当于修读某个专业一个完整学期的工作。一些机构也提供合作教育计划,让学生交替地完成学习学期和工业界的工程实践。这类活动具有跨学科的性质,并提供了一些机会,特别有益于团队活动。因此,计算机工程师应该能与其他学科的专业人员共事,如计算机科学家、电气工程师、机械工程师、企业家、财务专家、营销人员以及产品设计师等。

## 4.7 专业之外的技能

当今的社会,学校面临着压力,要确保毕业生有能力满足雇主的需求。一个更积极的观点是学校要作为变化的弄潮儿,培养的毕业生进入就业时就具备可造福于雇主的技能,满足雇主的期望。

达到这个目标的一种方法是确保学生拥有一些“可迁移能力”,以及诸如沟通能力、团队合作能力和表达能力等的个人技能。可迁移能力指的是个人可在任何职业中运用的技能,可以从一个工作带到另一个工作而无需再次培训。此外,个人也要具备对图书馆利用及研究的技能,以及诸如时间管理、项目管理、信息素养、信息管理、职业发展、认识自我、在专业领域中能跟上创新的步伐等专业技能。教师应该激励学生发展这些能力,在计算机工程的教学给学生过程性的反馈和评价,强调它们与这一学科的相关性和重要性。

经常存在这样的一种危险,即在专业外的技能上花费了大量的时间和精力,淹没或取代了传统的教学内容,进而减弱了专业知识的掌握。这里需要做一个平衡,通常情况下,需要有一个巧妙的教学和测试的方法,来确保课程中不存在偏废。

### 4.7.1 沟通技巧

计算机工程师必须能够有效地与不同的听众沟通。因为在几乎所有的职业生涯中良好的沟通技能是如此地重要,学生必须提高他们在各种情况下的口头和书面表达技能,无论是在计算机工程课程之内或之外。

计算机工程师工作的一个重要方面是向生产或技术支持人员传达和协商项目需求与实现细节,在工业界,这种场合可能是在本地或远程。提供清晰简洁的文档,能正确的认识支持人员的作用和目的,会直接影响工作环境的效率和性质。这是一种最基本的沟通能力。因此,学生应该学会:

- 以书面形式有效沟通观点,包括技术文档的编写(例如,规约、需求、安全案

例、文档)和报告的写作,应能正确并恰当地运用图、表,并给出适当的参考文献;

- 制作有说服力的解说材料,并进行有效的口头报告,包括正式和非正式两种形式;

- 理解别人的报告并提出建设性意见;

- 在答辩中能进行(礼貌而有效的)辩论,使用专业和系统的方法通过仔细而深入的询问来提炼客户方的需求;

- 对产品的能力进行有效展示。

虽然学校可能会采取不同的策略来达成这些目标,每个计算机工程专业的学生培养方案还必须在多种场合来提高学生的写作、口头表达和积极聆听的技能。

为了提高或强调所有学生所必需的沟通技巧,计算机工程课程体系至少应该要求:

- 强调写作的机制和过程的课程内容;

- 强调表达的机制和过程的课程内容;

- 一份或多份正式的书面报告;

- 评论书面报告的机会;

- 一个或多个正式的小组内口头报告;

- 评论口头报告的机会。

而且,计算机工程课程体系应切实地整合书面形式和口头讨论。学校不应该把沟通技能视为分离的实体,相反,教师应该将这样的技能完整地融入计算机工程课程体系及其要求之中。

基于电子媒介的沟通已经成为一项重要的补充手段。它们在工程师的生涯中起着核心作用。除了电子邮件、电子日历、代码和文档库这些显然需要的手段之外,学生还应该在一定程度上参与有效的协同工作和团队学习活动。

## 4.7.2 团队合作技能

计算机工程专业人员很少会长期单独工作。通常的情况是一群人作为一个团队实施重要的计算机工程项目。很多时候,这个团队都是跨学科的。因此,在本科学习阶段,计算机工程专业的学生需要了解高效团队的组成方法以及有效参与团队活动的机制和动力。此外,由于团队工作的价值(以及产生的困难)在小规模的项目中不明显,学生需要参与一个相对长期的团队项目,比如完整的一个学期甚至更长时间。

团队合作的很多问题涉及沟通技巧。当涉及跨学科的团队时,个人往往会根据其技术专长而被分配角色。然而,在团队活动中,会出现由于团队工作造成的其他的一些重要问题,如:团队本身与团队组成、团队中的角色、组织团队会议、达成共识的处理方法及记录重大决策、接口的重要性、截止期和计划、以及质量控制机制的重

要性等。计算机工程培养方案应包括一些活动,以确保学生在本科阶段有机会获得这些技能。例如:

- 在培养方案的早期尽快提供团队合作的机会;
- 由小规模的学生团队完成一个涉及复杂设计及实现的产品或原型项目。

### 4.7.3 软技能或个人技能

工业界的经理们几乎一致同意,软技能(或个人技能)是雇用一个与计算相关职位的毕业生时的主要判断标准。他们的传统宝典认为软技能和技术技能具有相同或相似的价值。

那么,到底什么是软技能呢?一种定义是:软技能是“不依赖于所学知识的某些宝贵品质;包括常识,与人相处的能力,以及积极灵活的态度”[DICT]。另一个定义表明,软技能是“性格特征和人际交往能力,刻画了一个人与其他人的关系”[INVEST]。而且,“相对于我们知道什么,软技能更多地关注我们是谁。因此,软技能包括那些决定一个人能与他人良好互动的性格特征,通常体现了一个人的个性”[INVEST]。

在计算机工程和其他领域,软技能往往与硬技能互补,这是需要特别养成的能力。通常我们将这些软技能称为社会智慧的一部分或“以深切而直接的方式与他人联系、并进行良好的互动的能力”[CHIN]。这种以一种令人信服的方式与同事联系的能力在未来是非常重要的。事实上,它有可能成为区分职业生涯成败与否的显著因素。

### 4.7.4 经验

即使已经有必要的个人、沟通和团队合作能力,在特定的行业环境中,没有事先的行业经验,仅凭技术知识可能是不够的。这种先有鸡还是先有蛋的情况困扰了大学毕业生几十年甚至几个世纪。

好在近年来,工程界一直提供临时性的兼职或全职工作机会。对学生来说,这样的机会会有多种形式。参与实习、合作教育或勤工俭学计划不仅让学生获得实际经验,也可以让他们获得学分,在许多情况下,还可以获得工作报酬。通常,这种经历不要求学生选修课程,所以他们关注的重点是实习,而不是通过考试。学生可获得的任何形式的建设性经验,对那些在毕业后寻求长久职业的人是一个肯定的加分。

在大学里,学生经常在业余时间做兼职工作。这种融合现实世界经验和学术努力的方式为他们提供了一个必要的经验,帮助他们确定未来的职业目标。学术界需要包容这方面的经历,因为这有助于提高学生之间的区分度。经验往往成为毕业生获得职位的一个重要组成部分。尽管合格计算机专业人士短缺的局面预计会持续到2020年代,为争取一个理想的职位而不得不参加复杂的面试并竞争上岗的现象

依然如故[CHIN]。

本科阶段的计算机工程培养方案应为弥合学术界和工业界之间的经验差距探讨所有的可能性。与工业界建立一个强大的联系应是首要的任务。例如,建立一个强大的专业咨询委员会不失为一种向工业界打开大门的方法,因为该委员会的成员将与培养方案建立起联系。因此,学校的计算机工程培养方案应寻求业界的所有机会,使他们的毕业生有更大的就业和受雇机会。

#### 4.7.5 终身学习

快速的技术变化一直是计算机工程的一个特点,并将继续保持一段时间。毕业生必须跟上时代的变化,本科教育的一个关键要求是让学生具备这种能力。

以下一些基本策略是适用的。课程体系本身必须是当前的,设备必须是最新的,教师必须从事相关的学术活动。经鉴别为没过时的资源或有趣的资料,诸如教科书、软件、网站、MOOC、案例研究和演示实验等参考资料都可以成为学习经验中的一部分。此外,学生应熟悉来自 ACM 学习中心、IEEE 讲座角等专业团体所提供的广泛的相关资源。

终身学习本质上是一种态度,包括认识到需要新的知识,寻找和识别相关信息的能力,以及评估和应用这些知识的能力。学校可以通过新的教学和学习方法来培养这种态度,不断质疑和挑战现状,强调进步的机会。教师可以通过探索新途径的练习来挑战学生。在课程体系中都将学习当作值得注意的一个方面是十分必要的。也可以规划一个学习体验活动,挑战学生的思维过程。

#### 4.7.6 商业视角

计算机工程师需要对新产品开发中相关的各种非技术过程有一定的理解,以补足和完善他们的经历中的技术部分。从根本上说,计算机工程师需要能够评估创造力和创新能力,并关注可盈利的商业企业,包括已经成立的公司和尚在孵化的企业。学生会以多种形式从以下知识中获得教益,包括:

- 了解与新产品和新组织机构相关的金融与经济要素的重要性;
- 从市场角度评估相关性;
- 了解在产品设计和产品接受度方面的相关要素;
- 了解知识产权保护的各种途径;
- 评估团队合作,通常是跨学科团队合作的好处。

此外,学生需要评估他们对雇主的财政责任。将时间转化为金钱,按时完成工作是非常重要的。商业世界中会呈现企业制度和道德之间的冲突。学生应该知晓在政府或企业服务中即将面临的专业挑战。在计算机工程课程体系中,这些内容可能会出现于独立的课程(例如,经济学、工程经济学、市场营销或会计学),也可能作

为毕业设计的一部分,或者被整合到整个培养方案的其他课程中。

## 4.8 成为专业人士

当学生为未来的职业做准备时,一个重要的考虑因素是他们从学术环境向公司、组织、学术机构甚至是一个创业环境的职业生涯过渡的能力。如果一个人在他或她的学术生涯中没有接触过技术和专业外的技能训练,这种过渡可能会很困难。

适应性是一种个性特质,在计算行业中尤为重要,对未来职业生涯的成功将也是非常重要的。Gartner 集团预测,一场新的“数字工业革命”将迫使我们作出重大变化,如 3D 打印、不可改写的智能系统(non-overridable smart systems)和可穿戴计算机[ALIBAB]。

除了上大学期间要关注工业和获得宝贵的工作经验之外,还有一项重要的任务是,即将毕业的学生要准备好重要的面试,重新组织自己的简历,突出他们的技术背景。一份技术简历和普通简历的区别体现在对个人特点的强调上,如特定的技术、个人技能以及行业认证。能够应对一个成功的面试是一种职业技能,是学生在他们的学术研究的过程中必须实践和掌握的。如果学生无法应付面试的磨练,他们的学术成绩和各种学术成就也可能无法达成计算机工程教育预期目标。

## 4.9 工程教育的构成要素

总之,对专业实践的适当准备将使得毕业生具备以下能力:

- 明白他们的学科建立于合理的原则和健全的基础之上,认识到这些是什么,并能够运用它们;
- 理解理论与实践之间的重要关系;
- 重视设计,并能根据实际情况选择适当的方法;
- 认识到理解相关专业能力、伦理和法律问题的重要性;
- 认识到工具的重要性,能够应对构建工具的挑战,并认识正确而有效使用工具的必要性;
- 认识到他们的工作的适用范围;
- 重视创新和创造力,并了解相关的商业视角和机会;
- 认识到团队活动的重要性以及可以由此派生的长处;
- 了解产品设计的原则,包括健康、安全以及营销问题;
- 认识到专业化方法是重要的;
- 了解工程师所处的社会背景;
- 能够解决计算机工程中的一个重要的问题,并展现部署合适工具与技术、采

用专业化方法达成解决方案的能力。

除了这些特点,本章也试图勾勒学校必备并精心整合到计算机工程培养方案基本要素的范围,以确保毕业生知晓工程实践的最佳传统。

## 4.10 研究生和继续职业教育

由于计算领域的快速变化,CE2016 课程指导委员会强烈鼓励计算机工程培养方案的毕业生进入研究生院进一步学习。由于本报告侧重于学士学位(本科)课程,有兴趣者可探讨什么最符合自己的兴趣和需要。根据各自本科教育的结构,研究生培养可以有多种可能方案。在这个意义上,深度是必要的,也是一个专业化元素。

从事研究生学习的学生有机会获得关于硬件问题、软件问题、模型以及这些问题相互作用等方面更深入的理解。理论、实践、应用和态度(在这个层面上,甚至创新)的结合将继续指导每个模块或课程的建设。作为硕士学位的一项作用,强调(通常是单独的)项目工作,特别是在最后一年是非常重要的。对于这个水平上的学生,项目通常会论及新型研发前沿的创新概念。

## 第五章 专业实践

正如第二章所指出的,人们通常认为计算机工程(以及其他工程学科)是一种专业(如医生、律师等),而不仅仅是一个职业领域,认为它在专业实践的社会背景方面对其从业人员施加了一定的义务。因此,职业素养和专业实践是计算机工程师以及所有从事该领域的工程师生活的重心。

在可以影响公众生活的计算机系统(包括硬件和软件)的设计与实现方面,计算机工程师比其他计算机专业人员责任更大。因此,他们必须在从事工作时,考虑其工作的专业、社会以及道德背景。这包括版权和专利所体现的知识产权问题、法律问题(包括商务合同和就职协议)、从事网络和数据库相关工作而产生的安全和隐私问题、进行硬件和软件开发产生的责任问题以及因权衡产品质量和利润而造成的经济问题等。

职业素养还包括与设计决策相关的道德问题,它需要关注平等问题,因为这些决定会运用到每个用户的技术访问中。计算机工程师必须意识到其行为的社会背景,并对其行为在全球不同地方的含义保持敏感。这需要一种与人交往的技能(软技能),理解文化和风俗,与团队合作,使用多种媒体与广泛的受众进行有效沟通,在企业组织内部发挥作用,以及个人技能(如创造力、创新和领导力)。因此,随着计算领域的不断变化发展,职业互动变得更加全球化,一个前所未有的机会出现了。这个机会使专业实践成为计算机工程课程体系中一个不可或缺的部分。

我们现在给出专业实践成为计算机工程的一个组成部分的方法,并且探讨将专业实践纳入计算机工程课程体系的多种方式。下面分节讨论本质原因、教育界的现行实践、私营及公共部门对专业实践的支持、将专业实践纳入课程体系的技术方法以及评估这些技术有效性的策略。

### 5.1 专业实践综述

社会环境应该是工程设计和开发的一个组成部分。公众不会期望一座建筑、桥梁或隧道的设计和建造违背社会规则。同样地,人们也不会期望在 X 射线机中所使用的计算机系统的设计和制造会违背社会规则。计算机工程师应将他们的工作完成得最好。他们也应该遵循已有的专业实践规定,不从事会玷污自己形象、雇主形象或同事形象的活动。



### 5.1.1 专业实践与计算机工程课程体系

职业素养和道德应该是所有计算机工程课程的基石。在设计与开发上的关注点应该将社会规则置于个人研究领域中的最高位置。职业素养应该是贯穿整个课程体系的永恒主题。计算机工程专业的学生必须学会将理论、专业实践以及社会建构融入其工程职业。计算机职业素养应该是课程体系的一大重点。

把专业实践纳入课程体系十分重要,因为计算机工程专业的毕业生将在工作中面临现实世界中的各种问题,比如公共和私营部门的需求、公众对高质量产品的需求、越来越多的计算责任事件以及终身学习的需要。在大多数情况下,学生进校时对这些问题并没有一个完整的了解和评估。这种知识上的缺失对他们的老师和雇主来说都是一个挫败之源。事实上,随着学生学习到越来越多的专业实践及其基本问题后,他们会在学习以及如何与他人更好合作越来越感兴趣。因此,将专业实践纳入课程体系可以作为催化剂来刺激和提高学生对计算机的兴趣。

### 5.1.2 专业人员需求

私营和公共部门都能从学习过专业实践的学生那里受益。他们寻找这样的学生:具有在专业岗位上从事实际工作的经历;知道在和团队同事及客户合作时人际关系的价值;能够专注于进行高质量的工作;拥有强烈的道德信念;愿意为有价值的外部事业投入时间及才能;坚持终身学习;并且能够参与促进部门的完善。

用人单位、毕业生和学生之间的关系变得越来越重要。日益增长的对质量更好、缺陷更少的产品的需求也增加了将专业实践纳入课程体系的压力。随意的网络系统设计技术被公认为造成高缺陷网站系统的一个重要因素。因此,在与供应商签署合同前,客户会索要可靠系统流程的“证据”。学生需要了解与客户建立面对面联系的重要作用,约定可实现的要求,提交尽可能高质量的计算机系统。

## 5.2 社会背景下的决策

计算机工程师在其职业生涯中将会面临许多决定。虽然这些决定大部分是技术性的,但是其他一些也会涉及重要的社会背景。计算机工程师应了解合同法、企业组织和管理以及公司法的法律后果。

尤为重要的是涉及知识产权方面的问题。免费的开源硬件和软件可能有法律和道德上的隐含条件,存在着相当的利益。了解专利法是很重要的,特别是当他们工作的公司可能有一个积极的专利计划或当企业家自主开发产品的时候。了解版权法也是必要的,因为许多企业家和用人单位拥有他们所开发软件的版权。保护知识产权的另一种方法是商业秘密的使用。不同国家的政府对专利、版权和商业秘密

设有不同的法律。由于计算机工程师会在全球范围内工作,对专利、版权以及商业秘密的理解和应用就显得十分重要。

隐私和保密是计算机运作的基础。电脑可以储存大量的个人、企业、行业和政府的信息。人们可以利用这些信息来创建它们的说明文件。参与这类信息存储系统设计的计算机工程师必须仔细审计所开发系统的多种用途。计算机工程专业的学生应该研究一些案例,知道在他们所处的社会环境下,应该如何正确使用计算机系统。

计算机工程师必然要处理权衡问题。有时候,这些都是技术决策,比如计算机系统的时间与空间的权衡。然而,有时这些决策会涉及社会、经济或道德的权衡。这样的决策可能会关于风险水平、产品可靠性以及专业责任这类。计算机工程师必须意识到承担风险的各种后果和社会影响,要对他们开发的设计负责,并注意所采取的行动或将面临的潜在后果。这些决定甚至可能涉及安全性要求极高的系统,甚至面临生死选择。优秀的工程师不仅应该认真评估这类决定的社会影响,还应该采取专业措施以保护公众,培育公众信任。

最佳实践开始于教学实验室。教育机构应该鼓励能反映最佳实践的实验室行为模式。这种模式确定了一个水平或行为标杆,提高了学生的职业期望。他们还创建了一个支持专业信念的学习环境,而这正是计算机工程师所渴望的。例如,学校应建立正确使用机器和设备的安全指引,还应该为学生之间、学生小组之间以及学生与技术人员在实验室中的互动提供人际关系技能的指导准则。学校应该给所有计算机工程专业的学生灌输专业意识和最佳实践意识。

道德是在社会环境中做出决策的另一个方面。计算机工程师应该意识到存在许多道德体系。案例学习可以帮助学生理解他们将要工作的环境。

## 5.3 职业化与教育

目前已有许多措施将专业实践纳入课程体系,其中最普遍的是开设能够帮助学生提高沟通能力、解决问题的能力以及技术技能的课程。计算机工程培养方案可以在计算机课程中加强学生这些方面技能的培养。另外一个选择是在计算机工程系以外开设相关课程,比如传播系的演讲课或者英语系的技术写作课。然而,认证机构通常不仅要求学生通过通识教育课程或者计算机工程专业课程来获得这些技能,他们还要求学生在日后的学术经历中应用这些技能。

专业实践投入的覆盖面因机构投入、部门资源和教师兴趣而异。随着专业认证中逐步重视职业技能,学校可能会加强对专业实践教学的投入。

### 5.3.1 学生特殊体验

下面列出了几种为专业实践吸纳额外材料的潜在机制。

(1) 毕业设计课程。这些课程通常在学生的最后一年开设 1 个学期或 2 个学期。通常,学生必须团队协作来完成课题的设计和实现。这些课题通常包含对实际问题的考虑,包括成本、安全性、效率以及预期用户的适用性。学生可以开发仅仅为教学设计的项目,也可以完成来自校内外客户的真实课题。虽然课程的重点是项目工作和学生陈述,但是一些有关知识产权、著作权、专利、法律以及伦理的材料也可以包括在内。

(2) 职业道德与法律课程。这些课程要学一个学期。学生可以接触专业实践、道德行为、计算机法律以及不同国家法院管辖的地域界限等问题。研究领域包括计算历史、计算机对社会的影响、计算职业、法律与道德责任、国际计算机法律以及计算专业职业素养。

(3) 实习课/实习/合作项目。这些项目由学校或系资助,以使学生们有机会在毕业前在工业界进行全职或兼职工作。对此类项目提供充分的行政支持是其成功的关键。学生通常在暑假以及/或他们 4 年学习期间的 1~3 个学期来进行这些项目。选择合作项目或实习的学生一般在校外工作,这可能会中断他们一个夏天或一个学期的学业。学生通常会得到工作报酬,还会得到课程学分。

(4) 基于团队实施的课程。这些课程强调计算机工程开发的过程,通常是一个团队合作项目。课程包括开发过程、项目管理、经济、风险管理、需求工程、设计、实现、维护、软件和硬件的更替、系统质量保证、职业道德以及团队合作。覆盖范围通常广泛但不深入。

(5) 企业创新课程。计算机行业需要颠覆性的创新,也需要大公司提供新的技术和更多的就业机会。本课程讨论的是每一个经理在创业公司或已经成熟的公司中成功组织技术驱动创新所需要的基本知识。这将整合工程、管理、沟通和商务功能中的创新和设计思维。学生将利用在创业过程中所学的知识来评价、研究、编写和表述商业计划。

计算机工程系以外的许多课程也能帮助学生做出更好的专业实践。这些课程包括但不限于哲学伦理学、心理学、商业管理、经济学、技术沟通以及工程设计。

### 5.3.2 行政部门、教师与学生角色

在最高的制度层面,管理层必须支持教师进行专业实践以及部门发展活动。这些活动可能包括咨询工作、专业协会和社区服务、夏季奖学金、获得证书和专业执照、获得认证、成立拥有适当章程的工业咨询委员会、建立合作培养/实习/实践项目以获得课程学分,并与公共和私营部门建立更多的联系。这样的活动可能是非常耗时的。然而,它们对个人和学校都非常有价值,在决定晋升职称和终身职位时必须考虑这些活动。

通过共同采纳、促进和执行由专业协会制定的道德和专业行为准则,教师和学生

生可以协作工作。教师应加入专业协会,帮助建立这些协会的学生分会。通过学生分会,教师可以为课程工作、社区服务或者相关的专业活动中的重大成就颁发奖励。此外,学生分会还提供了一个场所,让学生与潜在的雇主一起工作,并有助于获得来自机构外的捐款、演讲者和导师。

### 5.3.3 将专业实践纳入课程体系

专业实践的纳入必须是一次清晰且积极的行动,因为大部分内容混合到了现有的课程结构中。例如,专业的导论课程包括计算机、互联网对社会的影响以及专业实践的重要性之类的讨论和作业。当学生进入二年级的课程时,他们可以像专业人士那样,以需求、设计和测试文档的格式来记录工作。

额外的材料,如计算机历史、数字图书馆、搜索技术、解决不明确问题的方法、带有个人责任的团队合作、现实生活中的伦理问题、标准和指导方针、法律约束和要求以及道德论证的哲学基础等,可能出现在专门课程中,或贯穿整个课程体系。贯穿于不同课程中的优点是在实际应用领域的背景下呈现这些材料。但是,它带来的问题也是显而易见的,即教师会因为学时的原因而尽量减少专业实践内容的讲授,而给专业技术知识更多的时间。当然,项目也可为这些内容提供一个更自然的利用方式,特别是老师能找到那些需要非关键系统的外部客户。当学生在社区从事服务学习项目或与外部客户一起工作时,他们开始从完全不同的角度认识到道德行为的必要性。因此,他们可以学到更多的如何解决客户不确定问题的方法。然而,无论老师如何把专业实践纳入课程体系,加强对这方面知识的练习、项目、考试都是至关重要的。

对于学时非常灵活或者有足够数量的教职人员和充足资源的系别来说,开设针对专业实践的课程是合适的。这些课程包括专业实践、道德、计算机相关的法律知识以及毕业设计和其他合适的课程。此外,网络环境下的经济、质量、安全和保密方面的高级课程可能是实践的一部分。这些课程可以来自计算机工程以外的学科,且将对学生的专业发展产生深远的影响。

计算工程课程体系中的职业道德是学科的基础。作为计算机工程知识体系的一部分(见附录 A),其对应的学习成效列表在专业实践(CE-PPP)知识领域。

### 5.3.4 职业素养与学生体验

教师可以通过建立一套规章制度来提高学生的职业素养。规章制度是学生工作的共同标准,鼓励其完成职业素养的养成。规章制度中应包括以下内容:

- 经常回忆含有适当专业实践内容的作业、课题以及考试;
- 建立健全的评价体系(如基于学习成效),对学生工作进行严格评估,以展示学生的进步和提高;

- 让学生参与评价和评估过程,使其获得更好的过程体验;
- 从私人 and 公共部门聘请专业人士来帮助评估学生的项目;
- 使用标准化的测试来衡量学生总体的进步;
- 对校友进行毕业后的调查,看看校友们认为在校教育对其职业生涯有何帮助;
- 获得项目认证,以证明符合专业实践的相关教育标准;
- 根据用人单位的需求同步课程实验,确保学生学到所需要的工作技能。

评价和评估程序应激励学生去做更好的技术实践,并高标准地完成。而不要鼓励学生试图在自身时间不充裕的情况下或者以一种随意的方式(比如在上交作业的前一晚才开始赶做)完成工作。即使学生是在一个团队中共同完成任务,考核过程也应充分考虑学生个人的责任心。该过程应该有一套一致的评估标准,让学生能够习惯使用,并知道如何将其与自身进步联系起来。

## 5.4 职业素养与工作场所

大多数学生大学毕业后继续在私营或公共部门就业。作为毕业生的主要接纳者,行业和政府帮助教育机构提高专业实践方面发挥着重要的作用。举个例子,参加过合作办学或在政府实习的学生在解决问题方面可能会成熟得更快一些,并能更加认真地学习。这样的实习也可以帮助那些提供实习机会的机构,因为学生毕业后可能选择继续留在实习过的公司工作。有了私营/公共部门的支持,专业实践的规模在课堂内外均得到了扩展。

### 5.4.1 私营与公共部门

私营和公共部门支持教育过程的最重要的方式之一就是鼓励他们的员工在帮助培养学生方面发挥更大的作用。这些员工可以在以下一些方面提供支持,他们可以:

- 以导师的身份帮助学生完成项目工作;
- 在课堂上进行特别陈述,向学生和老师介绍其公司、工作内容以及发展过程;
- 在大学兼职,以助教的身份延伸学校开设的课程;
- 在某些特定的研究、处理或软件工具领域为师生提供内部培训资料或课程;
- 服务于工业咨询委员会向校、系两级提供关于学生优点和不足的有价值的反馈意见。

通过上述任何方式,私营和公共部门都可以与向他们提供未来员工的教育机构建立重要的沟通。

除了校园里的各种机会,行业和政府也通过将教师和学生带到学术圈以外的环境中来促进发展强大的专业实践。学生和老师可以参观当地的公司,并开始建立更好的关系。长期来看,合作办学、实践课和实习机会让学生们更好地了解工作中的生活是什么样子。此外,学生可能对学习更感兴趣,并且利用这种新的兴趣增加自身的市场潜力。学生还可以与特定的用人单位形成联系,这样就更有可能在毕业后返回公司工作。对教师来说,咨询机会为其与公司之间建立了一个更高层次的信任。由于这些举措,用人单位、学生和教师对彼此了解更多,并且更愿意提升彼此的价值。

作为最重要的支持方式之一,私营和公共部门也可以捐赠或授权的方式,将硬件、软件、产品折扣、资金以及时间等向教育机构和专业协会提供帮助。通常,这些捐赠和授权对于学校是十分重要的,可以供实验室更新硬件和软件资源,资助学生奖学金/奖励以及教师教学/研究奖,也可以赞助学生编程、设计和教育竞赛。资助可以使更多的研究和项目实现。在这个层面上,私营/公共部门有助于确保未来教育的生存/进步,并推进计算机领域的进步。

通过耐心、长期的合作,相互理解对方受到的约束,并相互了解对方的价值观,私营/公共部门和教育机构可以共同努力,培养出具有良好专业技能和行为的学生。为培养具有高道德水准的学生,也为了使用这些学生开发的产品的用户的安全,企业、政府和教育机构之间的合作是十分必要的。

#### 5.4.2 模拟本地和国际工作环境

正如业界代表越来越多地寻求“立即上岗”的毕业生,大多数学生希望一毕业就在工作地从事计算机方面的工作,而不需要有大量额外培训。虽然教育经历不同于工作场所,但是教育工作者可以通过以下几点来缓解从学术界到工商界的过渡:

- 模拟工作环境的计算和网络资源;
- 教学生如何在团队中工作;
- 提供重要的项目经验。

把这些要点引入课程体系使得模拟本地和国际工作环境中的重大问题成为可能。教师可以讨论,并且让学生在设定的计算机资源、团队和项目的背景下去解决跨国家、跨文化以及工作场所的问题。

由于计算和网络环境的迅速变化并且有多种不同的环境并存,预测学生毕业后准确的工作环境是不可能的。因此,在课程体系上将注意力集中在特定的工具环境上是不可取的。接触多种计算平台和 Web 系统工具为扎实的专业工作提供了坚实的基础,培养出灵活多样的而不是幼稚地坚持他们熟悉环境的学生。当然,对一些工具和平台投入更大的精力进行深度掌握也是有益的。

正如后面第六章将要讨论的,对于许多学生来说,学习如何在团队中工作不是

一个自然的过程,但这仍然是非常重要的。学生应该学会在小型和大型的团队中工作,这样他们就可以获得计划、预算、组织协调和人际关系技能。应该有充足的课程教学内容来支持学生的团队合作能力培养,课堂讲授的内容可以包括项目计划、沟通技能、运作良好的团队和运作不良的团队的特点比较以及团队环境中的压力源。评估可以基于一个团队的工作结果、成员的个人工作或类似的一些组合。团队成员的行为也可以是评估的一个因素。

通过解决那些定义不明确或没有直接解决方案的问题所获得的重要项目经验可以增强学生解决问题的能力。这样的项目可以是一个可控的课堂体验,或者是由于校外客户带来一定程度的不可预测的情况。项目应服务于拓展学生超越仅仅完成在一个知识领域内的个人作业的目的。除此之外,项目还应跨越多个知识领域,从而帮助学生把所有的基本技能整合在一起。

### 5.4.3 资格证书

我们承认供应商和行业认证的价值,也鼓励学生去追求他们认为必要的东西。但是,我们并不认为应该为完成这些资格证书或此类认证的额外培训而提供学分,除非它也涵盖了本报告中定义的相关学习成效。我们站在这一立场,是因为许多提供认证培训的机构不具备高等学校的资格。大多数的资格证书是以实践为导向的,而不专注于基础理论和概念。此外,许多证书只针对某个给定的供应商,受用范围很狭窄。因此,它们通常不符合计算机工程学位所需要的学习成效。

## 5.5 职业精神培养

在前几节强调的问题已经导致许多专业协会组建委员会、开发职业计划、编写道德规范、进行专业实践以及有利于其支持者的其他工作。这些项目有助于从业者了解专业协会倡导的预期标准和从业者之间的期望。

### 5.5.1 职业道德规范

这些规范还提供了被看作是职业中心的有关戒律的公开信息。同时为专业人员提供了一个公平的竞技场,使其尽可能避免道德困境,并帮助他们在专业实践过程中面对伦理决策时“做正确的事情”。在计算机行业中,这些规范往往约束协会的会员,为从业人员作出影响其行动的决定提供指导和帮助。

美国计算机学会(ACM)、电气和电子工程师协会(IEEE)、信息技术专业人员协会(AITP)以及其他组织通过多种方式促进专业责任的发展。

- 他们发展和促进成员必须遵循的道德规范。在一般情况下,这些规范倡导诚实、正直、追求高标准质量,领导力,维护公众利益以及致力于终身学习的工作态

度。[ACM, IEEE, AITP, SEEPP]

- 他们赞助建立专业小组,比如直接关注道德和职业素养问题的计算机和社会专业小组(SIGCAS)和技术社会影响协会(SSIT)[SIGCAS, SSIT]。

- 他们制定和完善课程规范,如本报告及以前的版本。
- 他们参与制定认证准则,确保课程中包含专业实践。
- 他们支持学生分会的形成,鼓励学生树立一种成熟的专业实践态度。
- 他们通过技术出版物、会议和专题报告提供终身专业发展的机会。

以下列出部分职业规范。

- 国家专业工程师协会(NSPE):国家专业工程师协会工程师职业道德规范[NSPE, 2003]

- 电气与电子工程师协会(IEEE):电气与电子工程师协会道德规范[IEEE, 1990]

- 美国计算机协会(ACM):美国计算机协会道德规范和职业素养[ACM, 1992]

- 美国计算机学会/电气和电子工程师协会:计算机协会:软件工程道德规范与专业实践[ACM/IEEECS, 1999]

- 国际信息处理联盟(IFIP):职业标准的统一及计算伦理[IFIP, 1998]

- 信息技术专业人员协会(AITP):信息技术专业人员协会道德规范及行为标准[AITP, 2002]

计算机工程师可以利用这些协会规范来指导他们在工程事业中做出决策。

### 5.5.2 教育与专业实践

计算机工程培养方案应告知学生和社会他们能或者可以从在计算学科受过专业训练的人身上期望些什么。例如,学生需要明白专业行为对工作的重要性和粗心大意的后果。他们也应该认识到,专业协会通过建立工作组强调专业实践和道德规范,可以提供一个支持网络,使他们能够坚持道德上和职业行为的正确选择。在四年的培养方案中,就让学生充分了解这个支持网络,学生可以避免年轻专业人士经常感到的孤独感,并且准备好以一种更加成熟和道德的方式实践他们的职业。

尽管专业实践的原则专注于某协会的特殊目的,但都有普遍的共识。所有专业实践的根本都强调计算责任以及工程人员对公众及公共利益的专业态度。此外,专业实践和相关的道德规范解决了利益冲突、能力范围、客观性和真实性、欺诈以及职业素养等问题。

专业实践中划定的戒律应该是所有在职计算机工程师的特征。计算机工程师应在所有的工作中贯彻专业实践的原则。这对任何培养方案来说是义不容辞的——教导计算机工程师为了自己事业的利益和计算机行业、工程师职业以及全人类的利益接受这些原则。



## 第六章 课程体系实现问题

创建一个完整的培养方案远不是那么简单的。第三章介绍的、并在附录 A 中详细描述的知识体系可以看成是一个起点,但有许多其他的因素影响课程体系的创建。本章的目的是探讨在设计和创建一个完整的计算机工程培养方案时所存在的问题。这些问题包括一些细节,诸如将知识体系和附加素材包装成为必修课和选修课,确定所需的数学和科学课程,为特定计算机工程培养方案创建整体风格或精神的总体考虑等。

### 6.1 总体考虑

一个计算机工程培养方案需要在单一框架下关注大量的知识、实践技能、可迁移能力以及态度。一个培养方案应该表现出明显的、一致的精神,渗透到完整的学习计划中。在开始阶段就喜欢培养方案的一些特点,并对其有正反馈的学生可以确信他们能继续喜欢这些特点并可在更高水平上获得成功。

一个关键的问题是如何在不同学年的学习过程中分配相对成熟的素材(如电路和支撑性的数学课程)和近期发展的素材。当前,计算机工程学科正处于快速变化和发展中,并且这个特点很可能会延续下去。传统的课程设计方法建议基本和核心的素材应该出现在培养方案的开始阶段。它本质上的逻辑是:这些素材应该表现出一定程度的持久性和耐久性,应该在整个培养方案的生命周期内是变化不大的。然后,学生可以在打好这些基础后,进入到培养方案的后续部分,并继续进行终身学习。

这种观点需要考虑从学生的角度进行调整。选择学习计算机工程的学生往往是出于工程的动手实践特性以及他们以往使用计算机的经历。在最初的几个学期,如果学生只学习数学和科学,没有明显的计算机工程应用,可能产生挫折和失望的情况。

新工程领域通常处在研究和发展的前沿,在学习它们以后,学生才可以真正声称他们与学科领域的发展同步。这是很重要的,因为他们将以技术变更和转化的代言人角色进入工业界就业。其他因素也会影响特定培养方案的特点。这些因素包括:

- 本地需要(机构或地区);
- 日益多样化的学生群体的需求;

- 教师的兴趣和背景。

本报告通过不限定如何将核心材料组织成课程,同时给选修课程和创新课程足够空间来促进计算机工程培养方案设计的灵活性。核心知识体系提供涵盖学科的广度知识,并在关键领域提供更深度的知识。这为学生提供专精的机会,同时确保毕业生在变化和快速发展的领域为专业实践做好准备。

有些情况下,一个机构可能要设计一个聚焦于特定计算机工程领域的计算机工程培养方案,或者可能允许学生在多个领域中选择一个学习。在总体框架下可以很好衍生出多样化的培养方案设计。例如,可以是面向特定领域方向的学位,包括计算机通信、嵌入式计算机系统、系统层集成、移动计算系统、计算机系统设计、计算机设备、数字信号处理、多媒体系统、计算和广播、普适计算、高集成计算系统以及实时系统。也可以是面向一些专业领域进行开发,如机器人、生物医学工程、生物信息学、数字/模拟控制系统、网络空间安全等。

另一个考虑是有多少模块可以专门地为计算机工程学生所设计,以及有多少可以和计算机科学或电子工程(专业)共享。例如,机构可以采用以下几个可选项构建计算机工程课程体系。

- 在计算机工程(专业)中可能有足够的学生,有理由为计算机工程专业学生专门设计专业课程。
- 计算机工程师可以参加计算机科学和电子工程专业的课程体系中的课程,加上一些额外挑选的专门为计算机工程专业学生准备的课程。
- 根据当地的具体安排和情况,也存在其他可能性。

## 6.2 课程体系设计原则

尽管事实上课程体系设计需要在很大程度上适应当地情况,课程设计人员可以借鉴几个关键原则,以帮助决策过程。这些原则包括以下几个方面。

- 课程体系必须反映计算机工程专业作为一门独立学科的完整性和特性。计算机工程本身是一门学科。理论、实践、知识和技能的结合是这门学科的特点。任何计算机工程课程体系都应同时确保理论和职业素养指导实践的精神。
- 课程体系必须应对快速的技术变革,并鼓励学生也同样(应对技术变革)。计算机工程是一个充满活力、快速变化的学科。巨大的变化步伐意味着计算机工程培养方案必须定期更新它们的课程。同样重要的是,课程体系必须教导学生应对变化。计算机工程专业的毕业生应激发对工程的好奇心,从而跟上时代的发展。计算机工程培养方案的一个最重要目标应该是培养能成为终身学习者的学生。
- 用培养方案所期望达到的培养目标来指导课程体系设计。在定义计算机工程课程体系的过程中,很有必要考虑培养目标和学生培养方案结束时必须

具备的具体能力。这些目标和确定培养方案是否满足这些目标的相关技术提供了整个课程体系的基础。在世界各地,认证机构都越来越重视目标的定义和评估策略。教育者若试图捍卫其有效性,必须表明他们的课程达成了目标。

- 整体上,课程体系应该保持一致的精神,促进创新、创造力和职业素养。当学生理解到对他们的期望时,他们会作出最好的反应。若是在早期课程中鼓励某些特定的行为模式,却在后期课程中反对同样的行为,这对学生是不公平的。应鼓励学生在整个课程体系中运用他们的主动性和想象力,以超越最低要求。同时,应该鼓励学生从一开始就保持一个对自己工作的专业、负责的态度,相信那些将影响他们专业实践的伦理和法律问题。

## 6.3 计算机工程课程体系的基本成分

在组合课程体系时,机构必须将素材打包成模块,通常为单节课或课程。尽管有许多方式将这些课组织成为一个完整的课程体系,还是有一些共性可以作为一般准则在这里论述。

### 6.3.1 知识体系核心单元的覆盖

将知识体系本身看成一个课程的列表,为知识体系的每个领域指定一门课或课程序列,以核对其对知识体系的覆盖情况,这是有诱惑力的(且容易的)。事实上,大多数课程体系具有与知识体系的(各个)知识领域完美匹配的课程和课程序列。然而,覆盖一个知识体系的知识领域也可以采用将知识单元散布到几门课程中的方法。实际上,计算机科学课程体系 2013(Computer Science Curricula 2013)就建议,将信息安全分散整合到整个课程体系中。职业素养和专业实习知识领域的知识单元可以类似地被整合进贯穿课程体系的几门课程中。将知识单元散布到几门课程中,需要对负责这些课程的教师进行更多的协调和规划,这已得到公认。

### 6.3.2 课程安排

课程序列通常用于对知识点的覆盖,让学生随着课程体系进展,沿着入门、中级、高级知识点的顺序前进。入门的计算机工程课程的例子是基本程序设计课程,数字逻辑,基本计算机组成,电子线路,为进一步学习提供了基础。中级课程一般对这些知识点增加深度,向学生展现这些领域中的技术发展水平。中级课程也加强了来自入门课程中的知识点,例如,需要使用一个入门课程中讲授的编程语言来完成中级课程中的工作。高级课程是那些需要来自几门课程的广阔知识范围的课程,如计算机网络课程或操作系统课程。计算机工程课程体系也应该至少有一个或两个专业选修课程,允许学生在塑造他们的教育经历过程中有所选择。为了强调个人选

择,对专业选修课程的选择不应该被过分限制,例如,允许计算机科学或电子和计算机工程的高层次课程作为专业选修课程。

### 6.3.3 实验室实践

实验室实践对于提高实践计算机工程技术所需的技能是至关重要的。好的实验室实践让学生通过成功地完成一个满足一定要求的程序或硬件系统,来展示他们理解了基本的知识点。好的实验室实践,也向学生揭示了最先进的工具和方法,这些将为他们毕业后从事真正的工程工作做好准备。CE2016 课程指导委员会理解到实验室实践的开发需要教师投入相当多的时间,还需要部门资金支持,但依然强调,为了培养高质量的计算机工程专业毕业生,这些投入是必需的。有时,增加实验室实践需要创新的方法,也许可以把一个传统的 3 学时的课堂教学转化为“2 学时的讲课 + 1 学时的实验室实践”。

### 6.3.4 毕业设计

计算机工程学习的终点应该包括一个最后一年的项目,要求学生在解决一个实质性问题的过程中展示其对整个知识体系、实践和技术的掌握与应用。这个毕业设计可以(让学生)综合运用本科所学的广泛的知识,可以在合作伙伴间培养团队精神和专业实践精神。毕业设计是每一个计算机工程培养方案所必需的。

### 6.3.5 工程职业素养

职业素养对于培养全面发展的计算机工程专业毕业生是重要的。第五章给出了将职业素养元素结合入计算机工程课程体系的几种方法。

### 6.3.6 沟通技能

计算机工程专业的学生必须能够有效地采用书面以及正式、非正式口头演讲的方法沟通思想,如第五章所述。因此,计算机工程培养方案必须培养它们的学生用理性和合理的论据,向广泛的听众讲述技术和非技术素材的能力。讲述的方式包括所有的工科培养方案中都需要的口头、电子和书面方法。虽然计算机工程以外的课程可能有助于(学生)获取这些技能,但是在计算机工程课程中包含适当的沟通要求也是必要的。这是确保学生有能力沟通与学科相关的内容所必要的,也有助于学生学习技术内容。

## 6.4 由其他部门提供的课程

除了那些计算机工程专门的技术课程,还有一些其他课程反映了需要包含在课

程体系中的素材。例如,计算机工程专业的学生必须学习数学和科学,它们构成了工程的基础。在这一节中,我们论述学生必须学习,但通常不由计算机工程所在系自身开设的课程。在某些情况下,学生可能在进入计算机工程培养方案之前已经学习了这些内容。

### 6.4.1 数学要求

数学技巧和形式化数学推理是计算机工程中大多数领域不可或缺的部分。这个学科的许多基本支撑基础都依赖于数学。此外,数学提供了一种语言用以处理与计算机工程相关的思想,提供了分析和验证的具体工具,提供了理解重要思想的理论框架。

鉴于数学在计算机工程中的普适角色,课程体系需要尽早地、并经常地包含数学概念。基本的数学概念应该较早在学生的课程中出现,并且在后续课程应该经常使用这些概念。尽管不同的学院和大学需要调整自己的先修课结构以反映当地的需求和机会,对高层计算机工程课程来说,能利用早期课程中的数学内容仍十分重要。一个正式的先修课结构应该反映这种依赖性。

数学中的一些内容,从本质上说处于计算机科学与工程边界区域,实际上计算机工程系的教师就可以讲授它们。其他内容如微积分基础就很可能需要计算机工程系之外的教师来授课。例如,一门或多门离散结构课程对所有计算机工程的学生都是非常重要的。CE2016 课程指导委员会认为,它是计算机工程的一个必需的成分。无论怎样实施,计算机工程培养方案必须负起责任确保学生获得他们所需要的适用的数学知识。

CE2004 报告中包含离散结构和概率与统计的数学知识领域。对此,CE2016 知识体系添加了两个额外的数学知识领域(连续函数分析,线性代数)。应该注意的是,大多数的计算机工程培养方案已经包括了对这些知识领域的覆盖,这只是对此事实的确认。包含这 4 个数学知识领域原因如下。

- 离散结构。所有学生都需要了解离散结构的数学原理并接触相关工具。所有培养方案都应该包括对此领域的足够的接触,以覆盖在计算机工程知识体系中所规定的核心学习成效。

- 连续函数分析。微积分和微分方程是必需的,以支持诸如通信理论、信号和系统、模拟电路等计算机工程素材,它也是所有工科培养方案的基础。

- 概率论与数理统计。这个数学领域支撑了计算机工程师所关注的可靠性、安全性、性能分析、依赖性和许多其他概念。许多培养方案让学生学习公共的概率与统计课程,另外一些可能会让一些学生在这个专业下学习不到一个学期的课程。不管如何实现,所有的学生都应该至少简单接触一些离散和连续概率,随机过程,抽样分布,估计,假设检验,以及相关与回归,正如计算机工程知识体系所述。

- 线性代数。线性代数需要用来求解基本电路中描述电压/电流关系的方程式网络,并运用于诸如计算机图形学和机器人技术等计算机工程应用领域。

学生可能需要学习额外的数学,以提高其在这个领域的知识素养,并支持对诸如通信理论、安全、信号和系统、模拟电路、人工智能等课程的课程学习。这些数学可能由各个领域的课程组成,包括进阶微积分、变换理论、数值方法、复变函数、几何、数论、优化方法以及符号逻辑。对课程的选择应基于培养方案的目标、学校的要求和学生个人的需求。

### 6.4.2 科学要求

抽象化是在计算机工程领域中逻辑思维的一个重要成分。科学方法(形成假设,实验和数据收集,分析)代表了对计算机工程学科大部分知识领域都有用的一个基本方法,学生应该扎实地掌握这种方法。

计算机工程专业学生需要自然科学知识,如物理和化学。电磁学的基本物理概念构成了知识体系中电子工程知识点的基础。其他科学课程,如生物学,与计算机工程师可能专攻的特定应用领域有关。对自然科学的确切要求可能因学校和培养方案的需求以及资源的不同而不同。

要形成一个对科学方法坚实的理解,学生必须对假设构想、实验设计、假设检验和数据分析直接动手实践。而一个课程体系可以将这种实践作为自然科学课程作业的一部分,另一个方式是通过计算机工程中适当的课程本身来实践。例如,对用户界面的考虑就提供了丰富的实验情境。

计算机工程专业学生是“做科学”,而不仅仅是“读科学”,这在对他们的教育中是至关重要的。课程体系这一要素的总体目标包括以下内容。

- 学生应该掌握支撑计算机工程及相关应用领域的自然科学知识。
- 学生必须形成对科学方法的理解,并且在一些包含实验室工作(包括数据收集和分析)的课程中经历这种探究模式。
- 根据培养方案的目标和他们感兴趣的领域,学生可以获得任何他们需要的科学认识。

### 6.4.3 其他要求

许多机构都有些适用于所有学生的其他附加要求,如通识教育要求。这种要求的规模和内容有很大的不同,取决于所在的国家、学校的使命,法律的要求以及其他因素。用来满足这些要求的课程通常来自人文科学、社会科学、语言和文学艺术。在设计一个计算机工程培养方案时,应注意利用这些课程要求,帮助学生理解工程的社会背景以及工程解决方案在全球环境中的潜在影响。

## 6.5 课程样例

附录 B 提供了完整的计算机工程培养方案的实现样例。为了提供一个能展现本报告所提出的观点的框架,前三个样例做出如下假定。

- 每年包含两个学期,学生每个学期学习四到五个模块(课程)。每个模块约 42 学时的课堂教学。
- 学生应当在第一年学习至少两个计算机工程模块,第二年至少学习四个模块,第三年和第四年至少学习五个模块。

美国的学校使用上述的模式,在世界上许多其他地区也是同样的。课程体系实现的第四个样例是中国的一个典型的四年培养方案。第五个样例是反映了欧洲博洛尼亚宣言的可能的课程体系。

## 第七章 教育机构的适应性变化

本报告旨在为那些正在尝试发展和提高计算机工程专业本科生培养方案的大学或学院提供具有借鉴意义的内容。报告的附录广泛地分析了计算机工程专业知识的范围和结构,以及开展本科生课程可行的方法。然而,成功地开展课程教学,要求每一个教育机构全面考虑战略战术问题,涵盖但不局限于这些细节。本章枚举了这些问题,并提供了一些解决方案。对于已经存在计算机工程培养方案的学校,下文所述的内容可能已经在这些学校部署到位,或者广为人知。

### 7.1 自我调整的需求

设计计算机工程课程体系是一项艰巨的任务,一部分原因是因为这项任务与教育机构自身的特性以及教师的研究领域及科研水平是密不可分的。即使所有教育机构可以遵循同一套关于本科所需讲授的知识和技能的标准,仍然有很多额外的因素会影响课程的设计。

- 教育机构的类型及毕业预期。不同的教育机构在使命、结构和毕业要求上具有巨大的差异。例如,一套在一所美国小型学院适用的课程体系,放在其他国家的研究性大学中可能完全不适用。

- 计算机工程专业的研究范围。每一所教育机构需要选择研究重心,并且保证学科的研究风气在深度和广度上具备一个良性的平衡。

- 潜在的研究领域。有些教育机构的首要任务是培养计算机工程领域技术型人才,而另一些教育机构则培养研究生和研究型人才,两类机构在课程目标是截然不同的。每一所学校均需要保证其培养方案为学生最终的目标提供必要的准备,无论这个目标是学业上的还是职业规划上的,以及是否属于计算机工程学科范畴。

- 新生背景和入学准备。不同教育机构的学生(通常一个学生只就读于一所教育机构)在入学准备的程度上是十分不同的。因此,计算机工程系通常需要调整引导型课程,以满足学生的实际需求。

- 机构内培养方案的定位。计算机工程的培养方案可能由计算机学院、工学院、文理学院等主持。任何情况下,这些培养方案需要环境支持,以保证其具有持续的健康和活力。

- 教师资源。支撑一个培养方案所需的教师数目的差异可能是很大的,在一所小型学院中,这个数字可能小到不足五人,而在一所大型研究型大学中,这个数字



可能超过一百。培养方案规模会严重影响其灵活性和选择余地。除去规模,在师资队伍中需要建立优先次序以更好地使用有限的教育资源。

- 教师的研究方向及科研水平。单个课程通常因特定的研究领域及院系的知识库而异,尤其在那些具有特定研究领域的较小的教育机构。

想要创造切实可行的课程体系,就必须在这些因素之间找到平衡,这使得不同的教育机构会做出不同的选择。没有任何一种课程体系适用于所有教育机构。所有大学或学院都需要考虑本文中提到的各种模型,然后设计一种方案以应对实际环境。

## 7.2 吸引和保留教师

计算机工程系面对的一个最令人沮丧的问题是招募并留住高质量的教职员工。在计算机工程领域,广告上登出的职位通常要比高质量的申请求职者多。申请成为教师的人稀缺,并且在学术界外的计算机工程师通常可以获得较高的薪水,使得吸引和留住教师变得很困难。教育机构需要一种积极进取的方案以招募和保留教师;激励机制诸如入职礼包及调整教学责任可能会产生有益的影响。积极关注职业机构可以提供与点对点项目负责人的社交机会,进而有机会见到及接触潜在的教师候选人。与其他项目或教育机构的合作或跨学科的交流也是一种潜在的策略。

虽然计算机工程专业可以从相关领域吸收教师,但作为一个专业领域,一个专业中必须有一个经过专业训练并具备深厚经验的核心教师。另外,教职员工必须时刻关注领域的发展,保证自己能跟上业界的发展脚步。教育机构必须为教师提供恰当的便利条件,以鼓励他们取得职业进步,无论这些进步是通过研究、参加会议、学术休假(也许是在工业界)、访问交流或其他活动获得的。

## 7.3 充足实验室资源的需求

教育机构必须认识到对计算机工程培养方案的资金支持是十分重要的。对于软件,获取和维护占据了计算总成本中一个相当大的部分,尤其是包括课程软件的开发成本的情况。对于硬件,采购及维护用于实验或提供给学生让他们自行开发系统的仪器设备是很昂贵的。另外,聘用支持人员管理并维护实验室基础设施也会是一笔开销。因此,想要取得成功,计算机工程培养方案必须取得足够多的资金支持以满足教师和学生的需求,进而提供一个适合学习的环境。

技术在飞速革新,然而计算机硬件却不能跟上技术的步伐,它们通常直到已经远远跟不上时代了,才会被淘汰。计算机系统,尤其是那些服务于高级实验室和安装了最新的软件工具的系统,适用年限可能短到只有两三年。所以制订定期更新或

替换系统的计划并将其编入预算是十分必要的。

计算机工程专业通常会把实验部分引入到课程体系当中,而且实验部分越来越多地要求实验室工作人员加强实验室设备建设和实验教学环节。这种发展会增加建设高质量计算机工程培养方案的开销,但经常与相关企业交流联系可以为学生提供有趣的案例研究,更可以为学生提供实习的机会。定期更新实验室资源有助于不断激励新学生。

最后,教育机构需要为学生提供访问在线参考文献的权限,诸如 IEEE Xplore Digital Library, ACM Digital Library, Webinars, 电子书, 在线课程, MOOCs 以及其他相关可用的资源。这些资源可以在 ACM Learning Center 上查询。

## 7.4 学业变更和教育途径

许多因素,如录取标准、学生学术准备、学生群体的多样性、生活经验、不同教育途径的设定等,会影响到不同本科计算机工程学术培养方案的入学、研习和获得学位。高中到大学这一过渡点,对于成功求职或深造更高学位会产生影响,然而影响的程度在全世界范围内是不同的。这一小节以全球性观点审视了进入和度过本科计算机工程学位培养方案的教育途径。

### 7.4.1 四年制的学业变更

理解进入计算机工程本科培养方案的切入点和教育途径可以帮助组织这些教育途径以更好地帮助学生、企业甚至整个计算机工业界。特别地,理解世界各地教育途径的差异和彼此的兼容性有助于培养世界型计算机人才。

CE2016 课程指导委员会考虑了攻读计算机工程本科学位的教育途径和一些计算机工程培养方案的严格程度,进而定义那些要求学生比公共基础教育学习更多科研课程的培养方案为“严格的”培养方案。这类培养方案至少要求学生学习五门数学类课程,并且学习的所有课程中至少有一半的课程是与计算机工程相关的。可以推测到,变更到严格培养方案的学生会少于变更到非严格培养方案的学生。如果一个培养方案过于严格,那么学生很可能无法完成。转学只可能发生在同等水平的大学之间。当然,也有一些例外,尽管人们可以预期到,在严格培养方案之间或非严格培养方案之间的学业变更,要比从非严格培养方案到严格培养方案的学业变更更可能发生。

### 7.4.2 技术学院(Technical Institute)的学业变更

技术学院培养方案中的学生不太可能成功地转学到四年制大学的计算机工程专业中,两类教育机构的培养方案的内容是截然不同的。技术学院的学生通常不会

学习大学水平的数学和科学课程。另外他们也很可能没有学习过大学要求的公共基础必修课。所以学业变更不太可能发生。

在世界上的许多地方,从技术学院转学到大学计算机工程培养方案是几乎不存在的。虽然会有少许例外,但学生将在技术学院的课程学习经历迁移到大学中是基本不可能的。

### 7.4.3 社区学院(Community College)的学业变更

在许多盛行社区学院或两年制大学培养方案的国家,从社区学院进行学业变更至大学的计算机工程专业是很普遍的。事实上,这样一种变更模式是受鼓励的,尤其是在美国和加拿大。比如说,在美国,一些州的学生可以将他们在社区学院中修得的学分冲抵到大学中的相同的课程上,这是他们的合法权利。这两个课程甚至可能拥有完全相同的名称和课程编码,例如“CHEM 101: Chemistry 1”。并且,许多州是区分社区学院迁移培养方案和纯社区学院培养方案的。他们为迁移培养方案设立了“理科类副学士”(Associate in Science),区别于职业导向的“应用科学副学士”(Associate in Applied Science)。

更进一步,在美国,社区学院和大学之间的“衔接协议”(Articulation Agreements)是非常普遍的。这类协议具有合法的地位,规定学生拥有将在社区学院中完成的课程冲抵到大学中相应的课程的权利。由于社区学院的学生数量和大学中学生数量近似相同,这种学习方式在整个国家内是非常受欢迎的。通常,一所社区学院会和若干所大学达成协议,允许学生进行学业变更。

# 附录 A 计算机工程专业知识体系

本附录定义的知识领域,很可能出现在计算机工程专业的本科课程体系。本报告正文的第三章已包含了该知识体系分类方案的基本原理及其历史、结构和应用方面的更多细节。本附录总结了这些基本概念以助于理解。为了阅读方便,我们会重复一些已经出现在报告中的材料。详细信息请参考第三章。

## A.1 简介

CE2016 课程指导委员会在制定这套课程模型时,首先定义了计算机工程专业的知识体系,所包含的知识领域如下。

- CE-CAE 电路与电子
- CE-CAL 算法
- CE-CAO 计算机体系结构和组成
- CE-DIG 数字设计
- CE-ESY 嵌入式系统
- CE-NWK 计算机网络
- CE-PPP 职业素养和专业实习
- CE-SEC 信息安全
- CE-SGP 信号处理
- CE-SPE 系统与项目的工程化
- CE-SRM 系统资源管理
- CE-SWD 软件设计

## A.2 知识体系

从结构上说,计算机工程专业的知识体系可以划分为三个层次。最高一层为知识领域,表示一个特定的学科分支,而不是一门课程。知识领域还包含一个“领域界限”,它界定了特定知识领域涵义。知识领域再被分解为更小的组成部分,称为知识单元,它表示在一个知识领域中单独的主题。每个知识单元将对应一组学习成效,属于最低层次。

### A.2.1 核心成分与附加成分

CE2016 报告的目标是尽可能少地保留知识本体所需的内容。这样做可以让计算机工程专业能够灵活地课程设置。为了贯彻这一原则,我们在知识单元中进行了区分,分为核心知识单元和附加知识单元。核心知识单元包含四年期学位所必须的知识和技能。附加知识单元则包括更高层次的工作所需的知识和技能。

CE2016 课程指导委员会发现有必要强调以下几点:

- 核心知识单元是指所有的学生在计算机工程学位课程中必须掌握的知识和技能。虽然有些内容也很重要,但是没有列在核心知识单元中,而是作为附加知识单元出现。没有在核心知识单元中出现的知识点并非不重要,只是意味着这个知识点不是计算机工程专业每个学生所必须掌握的。
- 核心知识单元并不能构成完整的课程体系。要培养出一名称职的计算机工程师,还需要额外的技术知识点,包括提供支撑的数学、科学和通识教育。
- 这并不是说,在四年期的课程体系中,要把核心知识单元都放到入门课程中。许多核心知识单元的确是入门型的。但是,有些核心知识单元则必须要求学生已经具备了有效的学习背景。

### A.2.2 覆盖一个单元所需的时间

为了让读者对覆盖一个单元所需的时间有感性认识,本报告采用了与其他课程报告相同的模式。CE2016 课程指导委员会选择“学时”作为时间单位,最重要的是“核心学时”。这个时间对应于传统课堂讲授所需的时间。因此,我们定义一个“核心学时”为 50 分钟的课堂讲授时间。

为了避免混淆,关于如何使用授课时间进行衡量,必须要强调以下几点。

- 本报告不规定教学的形式。虽然我们以传统课堂讲授形式为标准,但是我们认为对于其他形式也是有效的。尤其是随着教育技术的最新变化,出现了很多新的形式,包括翻转课堂、大规模开放在线课程(MOOCs)、混合式学习、预先录制的讲座和研讨会等。对于这些形式,概念上的学时数恐怕难以适用,但即使如此,时间至少可以作为比较的指标。在这个意义上,5 个核心学时比 1 个核心学时大概需要约 5 倍的时间或精力,这与教学形式无关。
- 报告中规定的时间不包括在课堂外的时间。分配给知识单元的时间不包括教师的准备时间或学生在课堂以外的时间。通常来说,课堂外的时间大约是课堂上时间的 3 倍。因此,一个要求 3 学时的单元通常需要实际花费 12 个学时(即在课堂上的 3 个学时和在课堂外的 9 个学时)。
- 报告中分配给一个知识单元的时间是其所需的最少时间。我们认为,让一个学生实现每个知识单元相关的学习成果,至少需要这个时间。使用者需要注意如

何解读这个时间测量手段。许多教师会发现,要学生达到他们希望的深度,往往需要比建议时间更长的时间。

### A.2.3 知识领域和知识单元的标签

每个知识领域有一个确定的标签,如 CE-NWK 代表计算机工程专业的“计算机网络”这个知识领域。通过在标签后加一个数字后缀,以标示对应的知识单元。例如,CE-NWK-2 是计算机网络知识领域的第二知识单元。此外,附加知识单元只有选修的学习成效,不包含建议的核心学时数。

### A.2.4 共有的知识单元

在每个知识领域内,第一个知识单元都是“历史沿革和综述”,第二个知识单元都是“相关工具、标准和(或)工程约束”。这两个单元提供了所在知识领域的相关背景知识。简短的历史沿革和综述介绍了本领域重要的贡献者和发展情况。而工程实践要求运用的现代工具和同时期的标准,都将随时间而变化。因此,根据知识领域和专业目标的不同,这些知识单元涉及的范围会差别很大。

## A.3 学习成效

为了便于理解学生应该从每个知识单元学到什么,本报告运用“学习成效”来描述每个知识单元。其重点在于“学习”。我们使用诸如“定义”、“评价”这样不同类别的动词来描述预期的学习深度。学习层次的范围从基本能力到高级能力,前者如背诵定义,后者如综合运用或评论。因此,“学习成效”提供了一种机制,不仅用于描述知识和相关的实际技能,也用于描述个人的可迁移能力。它描述了我们期望一个学生在毕业时知道什么或者能做什么。通过学习成效的描述语句,我们可以推定每个知识单元所需达到的最小深度。学习成效可以与知识单元、课堂活动、课程甚至培养方案相联系。

本报告中,CE2016 课程指导委员会已经尝试限制学习成效的数量,以强调基本知识和技能。设置培养方案时,可以通过构建课程体系,使学生以多样的方式表现出他们的知识和技能。如果能够用富于想象力的方法来评价学习成果,可以引导出一系列技能的独特表达方式。

## A.4 计算机工程知识体系总结

在本报告中提出的计算机工程知识体系的十二个知识领域,连同其相关的知识单元,都在表 A.1 中列出。这就是计算机工程知识体系。该表还显示了与每个领域

和每个单元相关联的核心学时数。例如,

CE-ESY-5 并行输入输出[3]

表示“并行输入输出”属于“嵌入式系统”知识领域的第 5 个知识单元,需要 3 个核心学时的讲授。这 3 个核心学时的讲授时间要求,表明它是计算机工程专业学位课程的核心知识单元。如果没有类似[3]的编号,意味着这个知识单元不是核心知识单元,而是附加知识单元。

**表 A.1 CE2016 知识体系**  
(核心学时:420)

计算机工程知识领域和单元	
CE-CAE 电路与电子 [50 核心学时]	CE-CAL 算法 [30 核心学时]
CE-CAE-1 历史沿革和综述 [1]	CE-CAL-1 历史沿革和综述 [1]
CE-CAE-2 相关工具、标准和(或)工程限制 [3]	CE-CAL-2 相关工具、标准和(或)工程限制[1]
CE-CAE-3 电气量和基本组件 [4]	CE-CAL-3 基本算法分析[4]
CE-CAE-4 电力电路 [11]	CE-CAL-4 算法策略[6]
CE-CAE-5 电子材料,二极管和双极型晶体管 [7]	CE-CAL-5 常见任务的经典算法[3]
CE-CAE-6 MOS 晶体管电路,时序和功率 [12]	CE-CAL-6 分析和设计面向特定应用的算法[6]
CE-CAE-7 存储单元结构 [3]	CE-CAL-7 并行算法和多线程[6]
CE-CAE-8 接口逻辑 [3]	CE-CAL-8 算法复杂性[3]
CE-CAE-9 运算放大器 [3]	CE-CAL-9 调度算法
CE-CAE-10 混合信号电路设计 [3]	CE-CAL-10 基本的可计算性理论
CE-CAE-11 设计参数和问题	
CE-CAE-12 电路建模与仿真方法	
CE-CAO 计算机体系结构和组成 [60 核心学时]	CE-DIG 数字设计 [50 核心学时]
CE-CAO-1 历史沿革和综述 [1]	CE-DIG-1 历史沿革和综述 [1]
CE-CAO-2 相关工具、标准和(或)工程限制[1]	CE-DIG-2 相关工具、标准和(或)工程限制 [2]
CE-CAO-3 指令集体系结构[10]	CE-DIG-3 数字系统和数据编码[3]
CE-CAO-4 性能评测[3]	CE-DIG-4 布尔代数应用[3]
CE-CAO-5 计算机算术[3]	CE-DIG-5 基本逻辑电路[6]
CE-CAO-6 处理器组成 [10]	CE-DIG-6 组合电路的模块化设计[8]
CE-CAO-7 存储器系统的组织和系统结构[9]	CE-DIG-7 时序电路的模块化设计[9]
CE-CAO-8 输入/输出接口和通信[7]	CE-DIG-8 控制和数据通路设计[9]
CE-CAO-9 外围子系统[7]	CE-DIG-9 可编程逻辑设计[4]
CE-CAO-10 多核/众核体系结构[5]	CE-DIG-10 系统设计和约束[5]
CE-CAO-11 分布式系统体系结构[4]	CE-DIG-11 故障模式、测试和可测性设计

续表

CE-ESY 嵌入式系统 [40 核心学时] CE-ESY-1 历史沿革和综述 [1] CE-ESY-2 相关工具、标准和(或)工程限制 [2] CE-ESY-3 嵌入式系统的特点[2] CE-ESY-4 嵌入式应用的基本软件技术[3] CE-ESY-5 并行输入输出[3] CE-ESY-6 异步串行通信[6] CE-ESY-7 周期性中断,波形产生,时间测量 [3] CE-ESY-8 数据采集,控制,传感器,执行器[4] CE-ESY-9 复杂嵌入式系统的实现策略[7] CE-ESY-10 低功耗操作技术[3] CE-ESY-11 移动和网络化嵌入式系统[3] CE-ESY-12 高级输入/输出主题[3] CE-ESY-13 嵌入式系统计算平台	CE-NWK 计算机网络 [20 核心学时] CE-NWK-1 历史沿革和综述 [1] CE-NWK-2 相关工具、标准和(或)工程限制 [1] CE-NWK-3 网络体系结构[4] CE-NWK-4 局域网和广域网[4] CE-NWK-5 无线和移动网络[2] CE-NWK-6 网络协议[3] CE-NWK-7 网络应用[2] CE-NWK-8 网络管理[3] CE-NWK-9 数据通信 CE-NWK-10 性能评估 CE-NWK-11 无线传感器网络
CE-PPP 职业素养和专业实习 [20 核心学时] CE-PPP-1 历史沿革和综述 [1] CE-PPP-2 相关工具、标准和(或)工程限制 [1] CE-PPP-3 有效沟通策略[2] CE-PPP-4 跨学科团队方法[1] CE-PPP-5 哲学框架与文化问题[2] CE-PPP-6 工程解决方案和社会影响[2] CE-PPP-7 职业与道德责任[3] CE-PPP-8 知识产权与法律问题[3] CE-PPP-9 当代问题[2] CE-PPP-10 经营和管理问题[3] CE-PPP-11 职业素养中的折衷	CE-SEC 信息安全 [20 核心学时] CE-SEC-1 历史沿革和综述 [2] CE-SEC-2 相关工具、标准和(或)工程限制 [2] CE-SEC-3 数据安全性和完整性[1] CE-SEC-4 脆弱性;技术和人为因素[4] CE-SEC-5 资源保护模型[1] CE-SEC-6 私密密钥和公开密钥加密[3] CE-SEC-7 消息认证码[1] CE-SEC-8 网络与 Web 安全[3] CE-SEC-9 认证[1] CE-SEC-10 可信计算[1] CE-SEC-11 侧信道攻击[1]



## 计算机工程知识领域和单元

CE-SGP 信号处理 [30 核心学时]	CE-SPE 系统与项目的工程化 [35 核心学时]
CE-SGP-1 历史沿革和综述 [1]	CE-SPE-1 历史沿革和综述 [1]
CE-SGP-2 相关工具、标准和(或)工程限制 [3]	CE-SPE-2 相关工具、标准和(或)工程限制 [3]
CE-SGP-3 卷积[3]	CE-SPE-3 项目管理[3]
CE-SGP-4 变换分析[5]	CE-SPE-4 用户体验 [6]
CE-SGP-5 频率响应[5]	CE-SPE-5 风险,可靠性,安全性和容错性[3]
CE-SGP-6 采样与混叠[3]	CE-SPE-6 硬件和软件过程[3]
CE-SGP-7 数字光谱和离散变换[6]	CE-SPE-7 需求分析及其获取[2]
CE-SGP-8 有限和无限脉冲响应滤波器设计[4]	CE-SPE-8 系统规约[2]
CE-SGP-9 窗函数	CE-SPE-9 系统架构设计与评价[4]
CE-SGP-10 多媒体处理	CE-SPE-10 并发硬件和软件设计[3]
CE-SGP-11 控制系统理论和应用	CE-SPE-11 系统集成,测试和验证[3]
	CE-SPE-12 可维护性、可持续性、可制造性[2]
CE-SRM 系统资源管理 [20 核心学时]	CE-SWD 软件设计 [45 核心学时]
CE-SRM-1 历史沿革和综述 [1]	CE-SWD-1 历史沿革和综述 [1]
CE-SRM-2 相关工具、标准和(或)工程限制 [1]	CE-SWD-2 相关工具、标准和(或)工程限制 [3]
CE-SRM-3 管理系统资源[8]	CE-SWD-3 编程结构和范式[12]
CE-SRM-4 实时操作系统设计[4]	CE-SWD-4 问题求解策略[5]
CE-SRM-5 移动设备的操作系统[3]	CE-SWD-5 数据结构[5]
CE-SRM-6 并发处理的支持[3]	CE-SWD-6 递归[3]
CE-SRM-7 系统性能评估	CE-SWD-7 面向对象程序设计[4]
CE-SRM-8 虚拟化支持	CE-SWD-8 软件测试、验证和确认 [5]
	CE-SWD-9 数据模型[2]
	CE-SWD-10 数据库系统[3]
	CE-SWD-11 事件驱动和并发程序设计[2]
	CE-SWD-12 使用应用程序接口(API)
	CE-SWD-13 数据挖掘
	CE-SWD-14 数据可视化

## A.5 知识领域与知识单元

### CE-CAE 电路与电子

[核心学时数:50]

#### 领域界定

本知识领域的知识单元组成如下。

1. 电路和电子在计算机工程中的作用和目的,包括模拟和数字电路的主要区别,它们的实现,以及用模拟系统近似数字行为的方法。
2. 基本电量和元素的定义和表示,以及它们之间的关系。
3. 简单的电子电路的分析和设计,运用包括软件工具在内的适当技术,并结合适当的约束和权衡。
4. 用于制造电子设备的材料性质。
5. 半导体器件的性能,其作为放大器和开关的运用,以及它们在一系列基本的模拟电路和逻辑电路中的应用。
6. 器件参数和各种设计方式对电路特性的影响,如定时、功率、性能。
7. 考虑和权衡大规模电路中的信号分布和不同逻辑系列及外部环境之间的接口。

#### CE-CAE 核心知识单元

##### CE-CAE-1 历史沿革和综述

达成核心学习成效所需最小学时数: 1

核心学习成效:

- 描述计算机工程如何应用电子设备和电路并从中获益。
- 明确电路和电子的贡献者,并说明他们与本知识领域的联系。
- 解释模拟和数字系统的主要区别以及它们的具体实现,并说明用模拟系统近似数字行为的方法。
- 总结基本电量和元件,用以展现电流和电压之间的关系。
- 描述晶体管的运用,包括用做放大器和用做开关。
- 解释自分立器件经集成电路一直到当前最先进电子设备的历史演变进程。

### CE-CAE-2 相关工具、标准和(或)工程限制

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 描述至少 2 种常见的电路模拟器,并对比它们的优点和应用。
- 解释数字计算机系统与模拟世界相联的问题,包括运用标准数据转换电路。
- 总结兼容性、互联性和安全性方面的标准的作用。
- 明确总线和其他互连和通信网络的目的。
- 说明在电子电路设计中的约束、参数和权衡的作用。

### CE-CAE-3 电量和基本元件

达成核心学习成效所需最小学时数: 4

核心学习成效:

- 陈述基本电量(电荷、电流、电压、能量、功率)的定义和表示,以及它们之间的关系。
- 定义并表示基本的电路元件(电阻、电感、电容)。
- 用欧姆定律解决问题,包括其功率表示。
- 用欧姆定律分析基本电路。
- 解释电阻和感抗的区别、相位的含义,以及频率对电容和电感的影响。

选修学习成效:

- 解读电容器和电感作为基本存储元件的作用。
- 对比相关的电气和概念,包括频率响应、正弦函数、卷积、二极管和晶体管,以及其他存储元件。
- 提供运用电路模拟器对简单电路建模和分析的实例。

### CE-CAE-4 电路

达成核心学习成效所需最小学时数: 11

核心学习成效:

- 比较电路模型中的各组成元件,包括独立电源和受控电源、串联和并联元件。
- 运用网路与节点分析方法、基尔霍夫定律、叠加、戴维南定理和诺顿定理等分析基本电路。

- 应用不同特性组合的电路,包括电阻(R)、电感(L)和电容(C),以及瞬态和稳态响应和衰减。

- 分析和设计包含电阻(R)、电感(L)和电容(C)的简单电路。
- 说明电路的频域特性。
- 对比电阻和反馈电路。
- 分析电路中电压和电流的相量表示。
- 解释正弦信号激励的电路响应。
- 定义和运用阻抗和导纳以及源转换。
- 运用电路仿真以建模和分析简单的电阻和RLC电路。

选修学习成效:

- 明确变压器的特性和用途。
- 解释电量和一些概念之间的关系,如传递函数、双端口电路、并联与串联谐振、最大功率传输和互感。

- 描述电压源的特性,如理想电压源、参考电压、射极跟随器以及采用放大器的电压源。

- 表述电流源的特性:理想电流源;晶体管电流源;共射,共源共栅,以及规管共源共栅电路;采用放大器的电流源。

### **CE-CAE-5 电子材料,二极管和双极型晶体管**

达成核心学习成效所需最小学时数:7

核心学习成效:

- 解释电子材料的特征和性能,包括电子和空穴;掺杂、受体和供体;p型和n型材料;电导率和电阻率;漂移和扩散电流,迁移率和扩散系数。

- 说明二极管的操作和性能,包括I-V特性、工作区、等效电路模型及其局限性。

- 说明NPN和PNP晶体管的操作和性能,包括I-V特性、工作区、等效电路模型及其局限性,以及与负载电阻相连的转换特性。

- 对比NPN和PNP晶体管逻辑和放大的偏置。
- 解释双极晶体管用做放大器和开关时的性能。
- 创建数学模型来表示电子器件的材料特性。
- 提供在电路模拟器中运用数学模型的例子。

选修学习成效:

- 对比肖特基、齐纳和可变电容二极管的差异。

- 设计一个单二极管电路,描述负载线的意义。
- 说明多二极管电路,例如,整流器和包含 DC-DC 电压电平转换器的直流(DC)电路。
  - 设计一个包含整流器的多二极管电路。
  - 设计一个包含 DC-DC 电压电平转换器的多二极管电路。
  - 只运用与、或功能实现二极管逻辑。
  - 提供用于构建常见电路的双极型晶体管的例子。

### CE-CAE-6MOS 晶体管电路,时序和功率

达成核心学习成效所需最小学时数: 12

核心学习成效:

- 说明 nMOS(N 型金属氧化物半导体)和 pMOS 场效应晶体管的操作和性能,包括 I-V 特性、工作区、等效电路模型和局限性、增强型和耗尽型器件,以及与负载电阻相连的转换特性。
  - 采用 nMOS 和 pMOS 晶体管的逻辑和放大器的偏置。
  - 比较 nMOS 和 pMOS 晶体管作为开关的性能。
  - 运用 nMOS、pMOS 和互补型金属-氧化物半导体(CMOS)逻辑实现基本的逻辑功能。
    - 运用通路晶体管和传输门实现逻辑函数。
    - 分析开关网络与逻辑门实现逻辑功能的影响。
    - 定义传播延迟、上升时间、下降时间。
    - 说明电路时序的简化单元延迟和 T 模型。
    - 分析逻辑门扇入和扇出对电路时序和功率的影响及其相关的权衡。
    - 对比时序和功率对晶体管尺寸的影响,包括 nMOS 和 CMOS 功率/延迟的缩放比例。
  - 计算不同设计风格对电路特性的影响,例如,静态逻辑、动态逻辑、多时钟方案。

### CE-CAE-7 存储单元结构

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 对比各种存储元件(例如,锁存器、触发器、时钟寄存器)的电路特性。
- 对比不同的存储器单元(例如,静态存储器、动态存储器、只读存储器)和相

关电路(例如,检测放大器)的电路特性。

- 对比存储元件和存储单元,强调对特定用途进行适当权衡。

选修学习成效:

- 对比不同类型的非易失性存储元件(例如,闪存,只读存储器)的电路特性。
- 通过时序图来展示和说明不同存储设备的输入、输出和时钟信号之间的关系。

### **CE-CAE-8 接口逻辑**

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 解释将系统内部接口信号传输到外部世界的实际困难。
- 运用各种逻辑系列和标准接口的终端特性。
- 写出不同的逻辑系列,如晶体管-晶体管逻辑(TTL)和CMOS之间信号转换的一般要求。
- 说明连接不同逻辑系列的常见困难的克服方法。
- 解释从单端到差分和从差分到单端转换的实际困难。
- 对比传输线特性、反射和总线终止选项,包括被动、主动、直流和交流(AC)特性。

### **CE-CAE-9 运算放大器**

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 解释理想运算放大器的性能(Op-Amp)。
- 分析和设计包含理想运算放大器电路的电路,包括反相和非反相放大器、求和差分放大器、集成电路和低通滤波器。

选修学习成效:

- 对比非理想运算放大器的性能,包括直流误差、共模抑制比(CMRR)、输入电阻和输出电阻、频率响应、输出电压和电流限制。
- 分析和设计含有非理想运算放大器的简单电路。
- 对比与设计多级运算放大器电路。

### CE-CAE-10 混合信号电路设计

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 讨论常见类型的混合信号电路和应用,包括数字-模拟(D/A)与模拟-数字(A/D)转换器和采样保持电路。
- 描述 D/A 和 A/D 转换器的关键特性,如最低有效位(LSB)、线性、偏移和增益误差。
- 对比 D/A 和 A/D 转换器的特性区别,以满足系统的设计要求。
- 分析在单个集成电路或封装中同时集成数字电路和模拟电路的问题,包括收益和挑战。
- 提供商业化的混合信号设备的例子。

选修学习成效:

- 描述基于具体实现的 D/A 转换器特性。例如,加权电阻、R/2R 梯形电阻/加权电流源变换器和 delta-sigma(三角积分)转换器。
- 描述基于具体实现的 A/D 转换器特性。例如,逐次逼近转换器、单和双斜率转换器、闪存转换器和 delta-sigma(三角积分)转换器。
- 设计一个 A/D 数模和 D/A 模数转换器,运用指定的实现方式满足给定的标准。

### CE-CAE 附加知识单元

#### CE-CAE-11 设计参数和问题

附加

选修学习成效:

- 计算设计参数对于开关能量、功率延迟乘积、功耗和噪声裕度的影响。
- 指出电源分配相关的问题。
- 描述信号耦合和退化的来源及其对电路特性的影响。
- 对比传输线效应,尤其是被动、主动、直流和交流终端。
- 运用适当的设计策略和软件工具,进行功率分布和传输线及元件的容差和权衡。
- 运用适当的设计策略和软件工具,减少噪声和其他信号退化的设计。
- 开发最坏情况下电路分析的方法。
- 用蒙特卡洛方法进行分析,并在电路设计中应用蒙特卡洛分析方法的工具。

- 检查六西格玛(six-sigma)设计方法在电路中的应用。

## CE-CAE-12 电路建模与仿真方法

附加

选修学习成效:

- 将仿真作为电路分析方法的优点和缺点。
- 将仿真方法应用到直流分析、交流分析、暂态分析和稳态分析。
- 明确哪些电路是不适合仿真的。
- 对比控制仿真的方法和参数,包括内建设备模型、设备参数控制、设备和电路库。

## CE-CAL 算法

[核心学时数:30]

### 领域界定

本知识领域的知识单元组成如下。

1. 基本的算法设计原则。
2. 算法特性的分析,包括算法之间的比较。
3. 用于诸如搜索和分类等常见任务的经典算法。
4. 设计和分析面向特定应用的算法。
5. 并行算法的特性。

## CE-CAL 核心知识单元

### CE-CAL-1 历史沿革和综述

达成核心学习成效所需最小学时数:1

核心学习成效:

- 阐述算法在软/硬件系统中的作用。
- 给出范例,说明算法的选择在程序中的作用。
- 讨论这个领域先驱们的贡献。
- 阐述为什么理论与算法是非常重要的。

### CE-CAL-2 相关工具、标准和(或)工程约束

达成核心学习成效所需最小学时数:1



核心学习成效:

- 使用函数库中的类和程序代码中可用的算法。
- 阐述如何找到函数库来支持感兴趣的程序。

### CE-CAL-3 基本算法分析

达成核心学习成效所需最小学时数:4

核心学习成效:

- 使用  $O$ 、 $\Omega$  (omega)、 $\Theta$  (theta) 标记来描述算法时间和空间复杂度的渐进上界、渐进下界和渐进确界。
- 确定简单算法的时间复杂度和空间复杂度。
- 通过实验方法度量算法的性能。
- 阐述为什么时间/空间的权衡在计算系统中很重要。

### CE-CAL-4 算法策略

达成核心学习成效所需最小学时数:6

核心学习成效:

- 设计和实现暴力算法 (brute-force algorithms)。
- 设计和实现贪心算法 (greedy algorithms)。
- 设计和实现使用分治策略 (divide and conquer strategy) 的算法。
- 阐述递归算法 (recursive algorithms) 的工作方式。
- 阐述启发式方法 (heuristic) 的作用并给出运用实例。

### CE-CAL-5 解决常见问题的经典算法

达成核心学习成效所需最小学时数:3

核心学习成效:

- 描述历史上用于搜索和排序的算法。
- 使用有效的排序算法解决问题。
- 说明在常见问题中选择合适算法所做出的权衡。
- 在涉及搜索的程序中使用抽象的数据类型 (比如哈希表和二叉搜索树)。

### CE-CAL-6 分析和设计面向特定应用的算法

达成核心学习成效所需最小学时数:6

核心学习成效:

- 指明影响算法选择的程序的特征。
- 说明用于应用程序领域比如控制程序、移动或位置感知的应用程序、离散事件仿真应用程序或者加密/解密程序中算法的特点。
- 指明影响特定程序算法性能的因素。

### CE-CAL-7 并行算法和多线程

达成核心学习成效所需最小学时数:6

核心学习成效:

- 分析一个简单顺序算法中包含的固有的并行性。
- 解释为什么通信和协调对于保障正确性至关重要。
- 计算理论上可达的加速比,并解释那些影响实际达到的因素。
- 解释可扩展性上的限制。
- 讨论并行算法的结构并给出示例。
- 展现在多线程下管理算法执行的方法。
- 选择适当的方法测量独立多线程算法的性能。

### CE-CAL-8 算法复杂度

达成核心学习成效所需最小学时数:3

核心学习成效:

- 解释一个问题是 NP 完全问题的含义。
- 给出 NP 完全问题的例子并解释其重要性。
- 说明为什么启发式方法在解决棘手问题时很重要。

## CE-CAL 附加知识单元

### CE-CAL-9 调度算法

附加

选修学习成效:

- 说明基于任务的优先级、任务的长度、到达时间等调度算法的基本策略,及

其对满足实时性限制的影响。

- 说明在应用中影响调度算法选择的因素。
- 分析调度算法对系统性能的影响。
- 给定一个任务集,展示调度算法的性能。

### CE-CAL-10 可计算性理论初步

附加

选修学习成效:

- 运用有限状态机(finite state machine)展示并分析系统的行为。
- 说明正则表达式和有限状态机的关联关系,及其重要性。
- 设计一个确定的有限状态机来接受一个简单的语言。
- 生成一个表达一种特定语言的正则表达式。
- 说明什么是上下文无关语法以及为什么有限状态机无法识别所有的上下文无关语法。
  - 解释什么是不可判定的问题(undecidable problem)。
  - 讨论什么是停机问题(halting problem)及其重要性。

### CE-CAO 计算机体系结构和组成

[核心学时数:60]

#### 领域界定

本知识领域的知识单元组成如下。

1. 计算机系统结构和组成的发展历史,以及它们在计算机工程中的作用。
2. 计算机系统结构和组成方面的标准和设计工具。
3. 指令集体系结构,包含机器级和汇编级表示以及汇编语言程序设计。
4. 计算机性能评价,包含性能度量 and 基准程序以及各自的优缺点。
5. 不同数制下数据的算术运算算法。
6. 计算机中处理器的组成及设计权衡,包括数据通路、控制单元和性能增强技术。
7. 存储器技术和存储器系统设计,包括主存、高速缓存和虚拟存储器。
8. 输入/输出系统相关的技术、系统接口、编程方法和性能。
9. 多核和众核体系结构,包括互连和控制策略,编程技术和性能。
10. 分布式系统体系结构,各级并行技术,各种体系结构的分布式算法。

## CE-CAO 核心知识单元

### CE-CAO-1 历史沿革和综述

达成核心学习成效所需最小学时数: 1

核心学习成效:

- 了解在计算机系统结构和组成方面历史上的一些贡献者以及他们的主要成就。
- 详细说明计算机系统结构和计算机组成之间的差别。
- 用框图说明一个简单计算机的各主要组成部分。
- 解释不同计算机系统结构的原因及策略,并指出各自的优缺点。
- 了解现代高性能计算技术,例如多核/众核和分布式体系结构。

### CE-CAO-2 相关工具、标准和(或)工程约束

达成核心学习成效所需最小学时数:1

核心学习成效:

- 辨别计算机系统不同层次的设计抽象:系统层、ISP 层、寄存器传送层(Register Transfer Level, RTL)和逻辑门层次。
- 论述部件之间互连的标准所包含的信息类型。
- 论述不同体系结构设计的选择和权衡如何影响性能和功耗。

选修学习成效:

- 比较两种不同的硬件描述语言,如 VHDL 和 Verilog。

### CE-CAO-3 指令集体系结构

达成核心学习成效所需最小学时数:10

核心学习成效:

- 解释说明冯诺依曼机器的基本组成及其主要功能部件。
- 说明计算机从存储器中取指令、对指令进行译码和执行的过程。
- 比较冯诺依曼结构和哈佛结构或其他结构的优缺点。
- 描述计算机的主要指令类型、操作数和寻址方式。
- 解释说明机器级操作的二进制表示以及对应的汇编级符号表示之间的关系。

- 解释说明在选择不同指令格式时需要考虑哪些方面,如每条指令所包含的地址个数,变长还是定长指令格式等。
- 描述精简指令集计算机(RISC)和复杂指令集计算机(CISC)两种架构的特点。
  - 使用汇编语言编写一些小程序,以体现对机器级操作的理解。
  - 用汇编语言实现一些基本的高级程序结构,包括像子程序和过程调用等控制流结构。
    - 使用程序直接控制方式和中断驱动方式编写一些汇编语言小程序,以实现简单输入输出设备的访问。

选修学习成效:

- 描述向量指令集(如:流式扩展、Altivec)的特征和应用,以及计算机系统结构和多媒体应用之间的关系。

#### **CE-CAO-4 性能评测**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 列出影响计算机性能的因素。
- 详细说明常用计算机性能指标(如:时钟频率、MIPS、CPI、吞吐率、带宽)的定义及其局限性。
  - 描述基准程序的基本原理和局限性。
  - 列举并描述两种常用的评价计算机性能的基准程序,并且使用公开的基准程序测试结果对两种不同的计算机系统进行比较。
    - 选择最合适的性能指标和(或)基准程序对一个给定的计算机系统上的一个目标应用程序进行评价。
      - 解释阿姆代尔(Amdahl)定律在计算机性能评测中的作用以及控制器和数据通路设计影响计算机性能的途径。

#### **CE-CAO-5 计算机算术**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 明确数制系统常用特征,包括表示范围、表示精度、准确性和导致算术溢出及下溢的条件,以及不同数制系统在特征之间的权衡。

- 描述计算机算术的局限性,以及由此所带来的计算错误的影响。
- 描述二进制整数的加法、减法、乘法和除法的基本运算算法。
- 使用标准浮点数 IEEE754 表示法,对浮点数及其二进制表示之间进行相互转换。

- 描述浮点数的加法、减法、乘法和除法的基本运算算法。
- 描述如何在计算机系统中进行多精度算术运算。
- 讨论处理器中运算部件会如何影响总体性能。

选修学习成效:

- 描述更复杂功能的实现算法,如平方根、超越函数。
- 给出饱和运算的概念,并讨论饱和运算所适用的应用场景。

### CE-CAO-6 处理器组成

达成核心学习成效所需最小学时数:10

核心学习成效:

- 讨论指令集体系结构和处理器组成之间的关系。
- 对比和讨论冯·诺依曼机数据通路的不同实现方式之间的权衡。
- 为给定的简单指令集体系结构实现数据通路和硬连线控制器。
- 设计乘法器、除法器 and 浮点数运算器。
- 解释基于流水线的指令级并行基本概念、流水线技术对计算机性能的影响、主要的流水线冒险,以及由于冒险而带来的性能损失。
- 描述缓解分支指令引起的流水线冒险可能带来的错误结果的几个步骤。
- 描述计算机系统中处理异常和中断所使用的通用机制。
- 描述超标量体系结构的特征,包括多发射、按序执行和乱序执行机制。
- 描述计算机系统每个功能部分如何影响总体性能。

选修学习成效:

- 讨论指令集提高计算机性能的方式,例如,预测/推测执行和支持 SIMD。
- 讨论处理器设计中的频率、功耗以及如何对其进行权衡。
- 讨论各种加速器(如:GPU、DSP、FPGA)如何用于提升计算机性能。
- 讨论如何运用并行处理方式设计标量和超标量处理器。
- 讨论如何运用向量处理技术增强多媒体和信号处理的指令集。

### CE-CAO-7 存储器系统组织和体系结构

达成核心学习成效所需最小学时数:9

核心学习成效:

- 识别目前正在使用的各种主要存储器技术类型。
- 使用给定的存储器元件设计一个特定参数的主存储器。
- 讨论如何通过存储器性能指标,如时延、周期时间、带宽、交叉编址等,来评价存储器对计算机整体性能的影响。
  - 解释采用存储器层次结构能够降低系统访存延时的原因。
  - 说明 cache 存储器的一般组成,解释为何使用 cache 存储器可以提升性能,并讨论不同 cache 组织方式下的性能/成本权衡策略。
  - 描述存储管理和虚拟存储器系统的基本原理。
  - 描述目前主流的外部存储器的技术特征,如磁盘、光盘和固态硬盘。

选修学习成效:

- 理解存储系统中的错误如何发生的,并说明用于解决这些错误的方法和机制,如检错和纠错系统、RAID 结构。

### CE-CAO-8 输入/输出接口和通信

达成核心学习成效所需最小学时数:7

核心学习成效:

- 画框图说明处理器如何与输入/输出设备交互,包括 I/O 地址空间的编址(独立、存储器映射)、握手协议和缓冲技术。
- 解释如何使用中断方式实现 I/O 控制和数据传送,包括向量中断方式、中断优先级,并讨论影响中断额外开销和时延的因素。
  - 使用汇编语言写出一个小的中断服务程序和 I/O 驱动程序。
  - 说明如何使用 DMA 方式与 I/O 设备进行交互。
  - 针对特定的应用,对程序控制 I/O、中断驱动 I/O 和 DMA 三种方式进行比较权衡。
  - 描述并行总线的特性,包括其数据传输协议。
  - 描述异步和同步串行通信协议的特征。
  - 对设备间并行传输和串行传输两种方式进行权衡比较。

### CE-CAO-9 外围子系统

达成核心学习成效所需最小学时数:7

核心学习成效:

- 对比一个或多个计算机系统扩展总线的特征。
- 为给定的部件/子系统与计算机系统之间的连接选定合适的总线。
- 描述从外存储器,如磁盘或固态硬盘,访问数据的过程。
- 解释外存储器子系统接口、存储控制器是如何工作的。
- 解释显示子系统及显示控制器的工作原理。
- 描述其他的输入设备子系统(如键盘、鼠标、音频)。
- 描述通信子系统:网络控制器、串行和并行通信功能。

### CE-CAO-10 多核/众核体系结构

达成核心学习成效所需最小学时数:5

核心学习成效:

- 讨论由于时钟频率和功耗墙问题而带来的单核处理器的性能局限。
- 描述多核/众核、共享存储多处理器系统的基本组成。
- 对同构系统和多核/众核异构系统的优缺点进行比较权衡。
- 讨论片上互连网络和存储控制等问题。
- 描述在多核/众核处理器系统中如何将程序划分执行。
- 详细说明在多核/众核处理器系统中目前所用的编程技术、编程模型、编程框架和编程语言。

### CE-CAO-11 分布式系统体系结构

达成核心学习成效所需最小学时数:4

核心学习成效:

- 解释不同分布式系统模型的差异。
- 解释分布式系统的并行粒度和并行层次的影响,包括线程、线程级并行和多线程
- 描述目前流行的多处理器系统和多计算机系统的拓扑结构、耦合度以及其他特征。
- 描述客户端-服务器模式如何以一种非集中式的方式工作。
- 解释代理是如何工作的,以及它们如何完成简单任务。



- 详细说明在分布式并行系统中目前所用的编程技术、编程模型、编程框架和编程语言。

选修学习成效:

- 描述客户端-服务器模式的现代实现机制,如基于云的计算。
- 描述逻辑时钟相对于物理时钟的概念,并说明它们是如何影响分布式系统的实现的。
- 比较简单选举算法和互斥算法的差别及其应用。
- 描述并行、同步、线程安全、并发数据结构的设计方式。
- 讨论分布式事务模型、分类和并发控制等概念。

## CE-DIG 数字设计

[核心学时数:50]

### 领域界定

本知识领域的知识单元组成如下。

1. 数字逻辑基础:数字表示法、算术运算、布尔代数及其基础逻辑电路实现。
2. 构件:组合电路、时序电路、存储器和算术运算电路。
3. 硬件描述语言(HDL)、数字电路建模、设计工具和工具流程。
4. 为实现数字系统而构建可编程逻辑平台(如:FPGA)。
5. 由组合和时序电路构件组成的数据通路和控制单元。
6. 分析和设计数字系统,包括设计空间探索和基于诸如性能、功率和成本等约束的权衡。

### CE-DIG 核心知识单元

#### CE-DIG-1 历史沿革和综述

达成核心学习成效所需最小学时数:1

核心学习成效:

- 指出数字逻辑设计领域的一些早期的贡献者并说明他们在该领域的成就。
- 论述计算机工程中受益于数字逻辑设计的应用。
- 描述布尔逻辑与数字逻辑设计的联系。
- 列举数字逻辑设计的基础电路,如组合逻辑门、存储元件和算术运算部件。

### CE-DIG-2 相关工具、标准和(或)工程约束

达成核心学习成效所需最小学时数:2

核心学习成效:

- 描述用于创建和模拟数字电路和系统的设计工具和工具链(如:设计输入、编译、模拟和分析)。
- 论述标准的必要性并列举数字逻辑设计领域中重要的标准,如浮点数标准(IEEE 754)和字符编码标准(ASCII, Unicode)。
- 运用以下标准硬件描述语言(HDL)之一模拟简单的数字电路:如 IEEE 1364/Verilog、IEEE 1076/VHDL。
- 给出一些重要工程约束条件的定义,如定时、性能、功耗、大小、重量、成本及其在数字系统设计环境中的权衡。

### CE-DIG-3 数字系统和数据编码

达成核心学习成效所需最小学时数:3

核心学习成效:

- 将带符号/无符号、整数/定点的十进制数与其对应的二/十六进制表示法进行相互转化。
- 运用二/十六进制表示法进行整数/定点数的加/减运算。
- 定义整数/定点数、带符号数/无符号数、加/减运算的精度和溢出。
- 运用 ASCII 和 Unicode 标准对字符串进行编码/解码。

### CE-DIG-4 布尔代数应用

达成核心学习成效所需最小学时数:3

核心学习成效:

- 给出基础(与、或、非)和派生(如:与非、或非、异或)布尔运算的定义。
- 列举布尔代数运算规律和定理。
- 运用基础的和派生的布尔运算评估布尔表达式。
- 运用恰当的代数定律和定理及其他技术(如卡诺图)编写并化简布尔表达式。

### CE-DIG-5 基础逻辑电路

达成核心学习成效所需最小学时数:6

核心学习成效:

- 描述二值逻辑中真/假的电子表示。
- 描述基础(与、或、非)和派生(如:与非、或非、异或)布尔运算的物理逻辑门实现。
- 描述高阻抗状态和诸如三态缓冲的逻辑门实现。
- 运用与-或、或-与、与非-与非、或非-或非和正/负/混合逻辑约定的两级门形式实现布尔表达式。
- 运用多个逻辑门级和正/负/混合逻辑约定实现布尔表达式。
- 论述逻辑门的物理特性如扇入、扇出、传播延迟、功耗,逻辑电压水平和噪声容限及其对设计约束和权衡的影响。
- 解释硬件描述语言(HDL)在数字系统设计中的必要性。
- 描述从硬件描述语言转化为物理实现的逻辑合成过程。
- 运用硬件描述语言实现组合逻辑电路,并用适当的设计工具来生成/验证。

#### **CE-DIG-6 组合电路的模块化设计**

达成核心学习成效所需最小学时数:8

核心学习成效:

- 描述并设计组合构件的单比特/多比特结构/操作,如多路选择器、多路输出选择器、译码器和编码器。
- 描述并设计运算构件的结构/操作,如加法器(行波进位)、减法器、移位器和比较器。
- 描述和设计提高加法器性能的结构,如先行进位和进位选择。
- 运用标准和定制的组合构件,以层次化、模块化的方式,分析和设计组合电路(例如,算术逻辑单元、ALU)。
- 运用硬件描述语言实现组合构件和模块化电路,并用适当的设计工具进行生成/验证。

#### **CE-DIG-7 时序电路的模块化设计**

达成核心学习成效所需最小学时数:9

核心学习成效:

- 运用周期、频率和占空比等参数定义一个时钟信号。
- 解释基础的锁存器(D、SR)和触发器(D、JK、T)的结构/操作。

- 描述基础锁存器和触发器的传输延迟、建立时间和保持时间。
- 描述并设计时序构件的结构/操作,如寄存器、计数器和移位寄存器的结构/操作。
- 分析并创建时序模块操作的时序图。
- 运用不同类型的时序构件实现基础存储元件时,列举设计权衡。
- 运用硬件描述语言实现时序构件,并运用适当的设计工具生成/验证。
- 描述不同类型的静态存储器特征,例如,静态随机存储器(SRAM)、只读存储器(ROM)和电可擦只读存储器(EEPROM)的特征。
- 描述动态存储器的特征。

选修学习成效:

- 描述异步设计技术(如握手),并论述其优点(例如,某些情况下的性能/功耗)和设计问题(如竞争冒险和缺少支持工具)。
- 描述高级存储器技术,如多端口存储器、双倍数据速率(DDR)存储器和混合存储器的特征(如混合存储立方体、HMC)。

### **CE-DIG-8 控制和数据通路设计**

达成核心学习成效所需最小学时数:9

核心学习成效:

- 描述被划分为控制+数据通路的数字系统并解释数据通路序列操作的必要性。
- 对比不同类型的有限状态机(FSM):如,米利型状态机、摩尔型状态机以及算法状态机(ASM)。
- 运用状态图描绘有限状态机(FSM)操作:如,米利型状态图、摩尔型状态图或算法状态(ASM)图。
- 分析有限状态机(FSM)的运行状态图并为其创建时序图。
- 为控制+数据通路设计的数字系统计算时序参数,如最高工作频率、同步输入的建立/保持时间、时钟到输出的传输延迟和引脚到引脚的传输延迟。
- 用硬件描述语言为控制+数据通路设计一个 RTL 模型,并用适当的设计工具进行综合/验证。

选修学习成效:

- 论述与控制+数据通路设计相关的时钟产生、时钟分布和时钟偏移。
- 运用流水线技术提高控制+数据通路设计的性能。

- 论述需要比特流序列化/反序列化的应用,并实现执行序列化/反序列化的设计。

### **CE-DIG-9 可编程逻辑设计**

达成核心学习成效所需最小学时数:4

核心学习成效:

- 描述可编程逻辑的基本要素,如查找表(LUT),与/或平面可编程逻辑、可编程复用器逻辑和可编程路由。
- 论述可编程逻辑架构,如现场可编程门阵列(FPGA)和复杂可编程逻辑器件(CPLD)。
- 描述可编程逻辑架构的常见功能,如硬宏(如:加法器、乘法器、SRAM)、时钟产生支持(如:PLL、多时钟网络),和不同逻辑标准支持。
- 在FPGA或CPLD中实现数字系统,描述和评估实现诸如可编程逻辑资源、最大时钟频、外部输入建立/保持时间和时钟输出传输延迟特征的权衡。

选修学习成效:

- 描述硬宏形式的可编程逻辑架构的高级功能,如CPU、高速串行收发器,以及其他收发器标准支持(如:PCI Express和Ethernet PCS)。

### **CE-DIG-10 系统设计和约束**

达成核心学习成效所需最小学时数:5

核心学习成效:

- 对比系统设计的自上而下和自下而上两种设计方法。
- 描述如何用逻辑综合时序约束和适当的设计工具,以影响控制+数据路径实现的逻辑生成。
- 运用时钟周期延迟和时钟周期吞吐量约束,创建数字系统的备选设计。
- 运用其他适当设计工具(如功率估计器)进行设计空间探索和基于诸如性能、功率和成本等约束的权衡。
- 描述系统设计约束可测试性的作用,以及提高可测试性的不同方法和工具。
- 描述联合测试行动组(JTAG)标准的功能/架构及其在数字系统测试中的作用。
- 为数字系统设计创建基于硬件描述语言的自检行为测试基准。

## CE-DIG 附加知识单元

### CE-DIG-11 故障模式、测试和可测试性设计

附加

选修学习成效:

- 解释系统测试方法在数字逻辑设计中的必要性。
- 定义故障模式,如固定故障、桥接故障和延迟故障。
- 在数字系统中设计测试方法时,定义术语可控性、可观察性、测试覆盖率和测试生成。
  - 描述可测试性方法设计,如随机测试、全扫描/部分扫描和内置自测试(BIST)。
  - 描述数字系统测试的计算机辅助测试工具的作用。

## CE-ESY 嵌入式系统

[核心学时数:40]

### 领域界定

本知识领域的知识单元组成如下。

1. 嵌入式系统在计算机工程中的用途和作用,以及在功率、性能和成本等方面的重要权衡
2. 采用汇编语言或高级语言设计嵌入式系统软件,运用现代工具和方法开发和调试典型嵌入式系统应用
3. 运用并行和异步/同步串行技术的数字接口,连接典型的片上模块,如通用输入输出、定时器、串行通信模块(如:UART、SPI、I<sup>2</sup>C 和 CAN 等)
4. 采用模拟-数字转换器的模拟接口,将通用传感器元件和数字模拟转换器连接到典型的执行元件
5. 采用短距离(如:蓝牙,802.15.4)和远程(如:蜂窝、以太网)等不同的互连架构连接的移动和无线嵌入式系统。

## CE-ESY 核心知识单元

### CE-ESY-1 历史沿革和综述

达成核心学习成效所需最小学时数:1

核心学习成效:

- 明确嵌入式系统的贡献者,并将其成果与知识领域相关联。
- 描述嵌入式系统的特性及其在示例应用中的作用。
- 解释嵌入式系统之所以重要的原因。
- 描述编程语言和嵌入式系统之间的关系。
- 描述计算机工程如何运用或得益于嵌入式系统。

### **CE-ESY-2 相关工具、标准和(或)工程限制**

达成核心学习成效所需最小学时数: 2

核心学习成效:

- 运用集成开发环境(IDE),面向目标嵌入式系统,编写、编译、组装和调试程序(高级语言或汇编语言)。
  - 比较和对比用于诊断/理解嵌入式系统硬件行为的各种仪器的特点。
  - 列举适用于嵌入式标准,如信号电平和串行通信协议。

### **CE-ESY-3 嵌入式系统的特点**

达成核心学习成效所需最小学时数: 2

核心学习成效:

- 对比用于嵌入式系统和通用计算的 CPU。
- 对不同的嵌入式系统应用的成本、功率和性能进行评估和分等级权衡。
- 描述目标嵌入式系统的体系结构特点(寄存器结构、内存结构、处理器特性、外围子系统)。
  - 对比不同类型处理器的嵌入式系统:CPU 微控制器,DSP 处理器,图形处理器(GPU),异构 SoC(CPU/加速器),基于 FPGA 的处理器。

### **CE-ESY-4 嵌入式应用的基本软件技术**

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 手动将简单的高级语言语句转换为等价的汇编语言。
- 描述程序翻译过程中编译、汇编、连接的操作过程。
- 描述编译器生成的代码在系统复位后、用户应用执行之前的操作。
- 描述编译器为全局变量、局部变量、子程序参数和动态分配存储所作的内存分配。

- 解释嵌入式程序基本的永久循环结构。
- 为嵌入式系统应用设计简单的程序,包括子程序和函数等模块化/层次化的编程技术。
- 演示简单的嵌入式应用程序调试技术。

### **CE-ESY-5 并行输入输出**

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 描述给定目标应用中的通用输入/输出(GPIO)中可能有的不同输入/输出配置(输入、强驱动、弱上拉/下拉、开漏级、三态)的合理性。
- 用轮询方式编写程序,实现针对多个GPIO的一系列输入/输出操作。
- 描述目标嵌入式系统是如何支持中断的。
- 用中断驱动的方式编写程序,实现针对多个GPIO的一系列输入/输出操作。
- 论述硬件和软件实现的先进先出(FIFO)数据缓冲机制。

选修的学习成效:

- 论述直接存储器访问(DMA)方式,并描述在目标嵌入式系统上如何实现。
- 用DMA方式编写程序,实现一系列输入/输出操作。

### **CE-ESY-6 异步串行通信**

达成核心学习成效所需最小学时数: 6

核心学习成效:

- 论述全双工和半双工通信的概念。
- 对比并行与串行输入/输出在吞吐量、布线成本和应用等方面的权衡。
- 描述异步串行接口中使用的数据格式、时序图和信号电平。
- 为采用异步串行接口的外部设备或系统编写程序。
- 描述用于同步串行接口的数据格式、时序图和信号电平,如SPI或者I<sup>2</sup>C。
- 编写程序,用SPI或I<sup>2</sup>C等同步串行接口连接外部设备或系统进行输入输出操作。

### **CE-ESY-7 周期性中断,波形产生,时间测量**

达成核心学习成效所需最小学时数: 3



核心学习成效：

- 描述嵌入式系统中典型硬件定时器的基本特征和操作。
- 编写程序,执行硬件定时器中断触发的周期性输入输出。
- 编写程序,用硬件定时器测量脉冲宽度和频率等波形特性。
- 描述脉冲宽度调制的应用。
- 编写程序,用于外部设备控制的脉冲宽度调制。

### **CE-ESY-8 数据采集,控制,传感器,执行器**

达成核心学习成效所需最小学时数: 4

核心学习成效：

- 描述模数转换(ADC)数模转换(DAC)的条件和特性,如采样率、参考电压、转换时间、精度、范围、编码方法等。
  - 给定范围、编码方式和参考电压参数,完成电压到二进制和二进制到电压的数值转换。
  - 描述 DAC 和 ADC 的架构方法,如电阻梯、逐次逼近、 $\Delta-\Sigma$  等,并给出转换时间和电路复杂度的权衡。
  - 给定一个样例传感器及其特征方程或图,展示从压力、温度、加速度等物理量到电压或电流的数值转换。
  - 编写程序,运用一个或多个外部传感器监测物理特征。
  - 给定一个样例执行器的特征方程或查找表,展示从电压或电流到线性/角运动、声音、光等物理量的数值转换。
  - 编写程序,采用一个或多个执行器,影响嵌入式系统的物理控制。
  - 设计电路,将电压电平/电流从外部传感器或执行器转换到目标处理器,或者反之。

### **CE-ESY-9 复杂嵌入式系统的实现策略**

达成核心学习成效所需最小学时数: 7

核心学习成效：

- 描述编写复杂嵌入式应用的结构化方法需求
- 描述事件驱动状态机框架中所运用的技术,例如,事件、事件队列、活动对象、事件处理、优先级队列、分层状态机等。
  - 描述用于实时操作系统的技术,例如,消息传递、抢占式与协同调度、信号量、队列、任务、协同例程、互斥器等。

- 运用状态机框架或实时操作系统(或两者兼具),为样例嵌入式系统应用编写程序。

### **CE-ESY-10 低功耗操作技术**

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 描述能源消耗的来源,如翻转、泄漏,以及用于功率最小化的引脚配置。
- 描述嵌入式系统设计中采用的节能方法,以及相应的性能/功耗的权衡,例如,睡眠/休眠模式、外围系统启用/禁用、时钟频率和睡眠/休眠期间适当的通用输入输出接口配置管理等。
- 描述唤醒机制,如看门狗定时器、实时时钟、外部中断等。
- 编写程序,演示睡眠或休眠模式下执行输入输出任务的最小能耗。
- 给定电池容量、电流、唤醒时间、睡眠时间、时钟频率等参数,计算嵌入式系统平台的系统电池寿命。

### **CE-ESY-11 移动和网络化嵌入式系统**

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 描述嵌入式系统在“物联网”中的作用。
- 论述如何在嵌入式系统中增加短距离无线连接,如蓝牙和 802.15.4,以及有关成本、功率、吞吐量和连通性的权衡。
- 论述如何在嵌入式系统中增加远程无线连接,如蜂窝和以太网,以及有关成本、功率、吞吐量和连通性的权衡。
- 对比在嵌入式系统中增加无线连接的硬件方案和软件方案,前者例如外部智能模块,后者例如软件栈加无线集成电路。
- 对比连通性结构,如点对点、星形和网格。
- 论述保护无线通信链路的安全选项。

### **CE-ESY-12 高级输入/输出问题**

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 论述 I/O 总线中的概念,如主/从设备、仲裁、交易、优先级和数据包等。

- 对比单端信号和用于高速串行总线的差分信号,以及测量差分信号质量的方法,如眼图。

- 描述一个或多个先进串行总线协议的特性,如控制器现场网络(CAN)、通用串行总线(USB)和 IEEE 火线的拓扑、特征信号电平、仲裁、速度、包结构、数据传输等。

- 论述嵌入式系统的体系结构和持久性存储应用,如闪存驱动器、SD 卡和 FRAM 等。

## CE-ESY 附加知识单元

### CE-ESY-13 嵌入式系统计算平台

附加

选修学习成效:

- 描述在先进嵌入式片上系统中出现的多媒体外设,如视频编码、音频处理、显示处理。

- 描述可用于 SoC 的互连和网络方式,如片上网络。

- 对比现场可编程门阵列中的硬核和软件 CPU 的性能、功耗、灵活性权衡。

- 描述嵌入式应用受益于多核的方法。

- 描述嵌入式应用受益的其他类型嵌入式系统处理器: DSP 处理器、图形处理器、异构 SoC(CPU/加速器)、基于 FPGA 的处理器。

## CE-NWK 计算机网络

[核心学时数:20]

### 领域界定

本知识领域的知识单元组成如下。

1. 计算机网络的发展历史,网络的层次结构及其在计算机行业所产生的巨大作用。

2. 在计算机网络研究中用到的相关标准以及常用的工具(如用于性能评估和网络拓扑的工具)。

3. 计算机网络体系结构,OSI 模型,TCP/IP 模型等。

4. 物理层和数据链路层数据通信和传输协议的基本原理和技术。

5. LAN 网络,MAC 层协议及 WAN 的概念和特性。

6. 网络层,传输层,应用层以及典型的网络应用如 e-mail、www 和 ftp。

7. 各种网络架构及协议之间的权衡取舍。
8. 网络管理的基本概念,目的及常见协议。
9. 无线传感器网络的特性和网络技术。

## CE-NWK 核心知识单元

### CE-NWK-1 历史沿革和综述

达成核心学习成效所需最小学时数:1

核心学习成效:

- 描述计算机网络的起源、演变历史及其应用。
- 解释计算机网络的重要应用。
- 指出哪些人在该领域做出了巨大贡献并且能列举他们的贡献。
- 解释说明计算机网络的基本组成和分层结构。
- 讨论分层结构在计算机网络构建中所起到的作用。
- 解释计算机网络的主要协议以及关键技术。

### CE-NWK-2 相关工具、标准和(或)工程约束

达成核心学习成效所需最小学时数:1

核心学习成效:

- 描述广义上的分类以及无线标准,如蜂窝网络和 802 系列标准。
- 对 802 系列标准有一个概述性的了解,包括 IEEE802.3、802.11、802.15、802.16 等。
- 对蜂窝网络标准有一个概述性的了解,包括 2G、3G、3.5G、4G、5G 和 LTE。
- 解释无线蓝牙技术标准。
- 对比现代网络不同模拟器的功能和基本用法。
- 对计算机网络发展的一些限制进行讨论,如传输介质、网络安全和网络管理。

### CE-NWK-3 网络体系结构

达成核心学习成效所需最小学时数:4

核心学习成效:

- 陈述网络的基本概念及它们的拓扑结构。
- 对比网络体系结构和网络的硬件组成。

- 对比网络协议的组成成分和分层的概念。
- 解释说明网络标准的重要性以及它们的规范委员会。
- 对 OSI 模型的七个层级进行描述。
- 定义网络的作用以及网络互联设备,如中继器、网桥、交换机、路由器和网关。
- 解释说明各类网络拓扑结构的优缺点,如网状、星形、树形、总线型、环形和 3-D torus。
- 描述 TCP/IP 模型。
- 对比 TCP/IP 模型和 OSI 模型。

#### **CE-NWK-4 局域网和广域网**

达成核心学习成效所需最小学时数: 4

核心学习成效:

- 解释 LAN、MAN 和 WAN 技术、拓扑以及相关的权衡取舍。
- 描述网络技术在片上互连网络的应用,如,Network-on-Chip (NoC) 架构。
- 对比网络协议不同的组成及需求以及它们的权衡取舍。
- 解释说明物理层的定义和功能,并描述不同传输介质和技术的特性。
- 阐述数据链路层差错检测和纠正的基本概念。
- 对比电路交换与分组交换。
- 解释共享介质的过程和控制方法。
- 对比以太网和千兆以太网的关键创新。
- 解释说明载波侦听多路访问(CSMA)的概念。
- 解释如何运用运行在物理层和数据链路层上的网络协议来构建简单的网络。

选修学习成效:

- 描述用于寻址及拥塞控制的协议。
- 描述用于虚电路及服务质量的协议。

#### **CE-NWK-5 无线和移动网络**

达成核心学习成效所需最小学时数: 2

核心学习成效:

- 从新型服务模式的角度解释说明无线和移动产业根本性的改变,如移动生

态系统(Apple 和 Android 生态系统)。

- 描述那些倾向于不变的基本组成成份,如移动 IP、Wi-Fi 和 cellular。
- 解释那些无线媒体传输过程中的潜在问题,如隐藏终端问题和暴露终端问题。
- 解释 Wi-Fi 网络的基础知识,如协议栈和帧结构以及它的进一步发展如 IEEE802.11 a/b/g/n 系列标准。
- 对比蜂窝网络的基本概念,如网络体系结构,框架和 LTE 等。
- 描述移动 IP 的主要特性,从移动解释它与标准 IP 的移动管理和位置管理上的区别,阐释移动 IP 的流量是如何选择路径的。
- 描述典型无线 MAC 协议的特性。

选修学习成效:

- 解释说明无线 CSMA/CA 和 RTS/CTS 放大机制。

### **CE-NWK-6 网络协议**

达成核心学习成效所需最小学时数: 3

核心学习成效:

- 对比面向连接和无连接服务。
- 从语法、语义以及时序等几个维度来对比网络协议。
- 从软件栈角度定义关键层的作用,包括物理层网络概念、数据链路层概念、网络互连和路由。
- 解释一些常见的协议组以及它们提供的服务(例如 IPv4、IPv6 和 TCP/UDP)。
- 描述网络层功能和组网技术。
- 对比不同的网络体系结构。
- 描述路由器中应用的重要技术。
- 对比至少两种重要的路由算法。
- 解释拥塞控制的概念,并比较相关算法。
- 描述 IP、TCP 和 UDP 的主要内容。
- 解释域名系统(DNS)的作用及其分布式设计的好处。

### **CE-NWK-7 网络应用**

达成核心学习成效所需最小学时数: 2

核心学习成效：

- 使用 LAMP (Linux, Apache HTTP Server, MySQL, PHP/Perl/Python) 或其他类似的例子来描述 Web 解决方案的关键组成部分。
- 解释说明客户端和服务端在一系列可能的应用中所发挥的不同作用。
- 选择一组可确保实现主从式配置的工具。
- 设计和构建一个简单的基于 Web 的交互式应用(例如一个简单的 Web 表格,从客户端收集信息并将其存储在服务器)。
- 讨论 Web 软件栈技术,如 LAMP 解决方案。
- 解释 Web 服务器的特性,如权限操作、文件管理和常见的服务器架构性能。
- 描述 Web 站点建立和 Web 管理的支持工具。
- 从高层次描述提供世界范围 e-mail 服务的各类客户端和服务器进行交互的方法。

选修学习成效：

- 给出使用动态 HTML、客户端和服务端模型来构建 Web 应用的解决方案。
- 陈述 P2P 模型的优缺点并给出例子。
- 解释云计算的原理、优点以及面临的挑战。
- 给出云计算 API 或其商业服务的例子,并总结它的关键能力。
- 描述一个要求计算机网络和其他许多领域,如机器学习、数据挖掘、HCI 和运输系统相结合的现代网络应用的主要组成及权衡策略。

### **CE-NWK-8 网络管理**

达成核心学习成效所需最小学时数：3

核心学习成效：

- 讨论网络管理可能的目的和主要的工具。
- 描述一个域名服务器在分布式网络管理中所发挥的作用。
- 描述常见的网络管理协议,如 ICMP、SNMP 等。
- 对比与网络管理有关的 3 个主要问题。
- 讨论网络管理的 4 种典型结构,包括管理控制台、聚合器和设备代理。
- 展示如何通过管理控制台管理网络设备,如企业级交换机。
- 对比多种网络管理技术在有线或无线网络中的应用,包括它们如何管理设备、用户、服务质量、部署和配置。
- 讨论将 IP 地址和 MAC 地址联系起来的 ARP 协议。
- 解释说明两个服务质量的问题,如性能和故障恢复。

- 描述自组织网络。
- 解释与网络相关的故障发现原则和技术。
- 描述与网络有关的管理功能域。

## CE-NWK 附加知识单元

### CE-NWK-9 数据通信

附加

选修学习成效：

- 定义数据通信的基本概念。
- 将信号和信号编码方法应用于通信服务方法以及数据传输模式。
- 解释说明调制在数据通信中的作用。
- 对比数据通信中 A/D 和 D/A 转换的相关问题。
- 对比通信硬件接口,例如调制解调器。
- 解释说明各类多路技术方法。
- 举例说明差错检测和纠错码的基本原理。

### CE-NWK-10 性能评估

附加

选修学习成效：

- 描述性能度量。
- 对比不同性能度量对特定网络和/或服务范式的影响。
- 对比服务范式,例如面向连接服务和无连接服务。
- 对比网络性能特性,包括延迟和吞吐量。
- 讨论网络错误的原因,例如丢包和数据损坏。
- 定义一个“体验质量”的度量标准(QoE、QoX 或简单的 QX),它用来度量网络服务(Web 浏览、打电话、电视广播、打电话给呼叫中心)的用户体验。
- 应用基本的模型理论(如:M/M/1 队列)分析网络性能。

### CE-NWK-11 无线传感器网络

附加

选修学习成效：

- 描述无线传感器网络(WSN)系统的特性。



- 描述无线传感器网络的 MAC 和路由协议。
- 讨论无线传感器网络数据融合的需求和策略。
- 给出一个无线传感器网络实际应用的例子。
- 对比电路交换和包交换:虚电路交换(MPLS)。

## CE-PPP 职业素养和专业实习

[核心学时数:20]

### 领域界定

本知识领域的知识单元组成如下。

1. 在专业人士和其他不同受众间进行有效沟通的重要性。
2. 领导力和专业互动能力在包含跨学科团队运作中的意义。
3. 执业计算机工程师的职业道德和责任及其对社会的影响。
4. 理解当代问题,终身学习策略,法律和知识产权问题的重要性。
5. 商业敏锐性的重要性和计算机工程领域中项目管理技巧。

### CE-PPP 核心知识单元

#### CE-PPP-1 历史沿革和综述

达成核心学习成效所需最小学时数:1

核心学习成效:

- 描述职业素养的本质,以及在计算机工程领域中的地位。
- 识别一些贡献者,并将他们的成就与社会和职业问题联系起来。
- 对与计算机工程相关的伦理问题和法律问题进行比较。
- 指出一些研究社会和职业问题的原因。
- 明确在某个确定问题中的利益相关者,以及作为工程师对他们的义务。
- 解释与执业计算机工程师相关的职业素养和执业许可。
- 描述计算机工程如何运用或受益于社会和职业的问题。

#### CE-PPP-2 相关工具、标准和(或)工程约束

达成核心学习成效所需最小学时数:1

核心学习成效:

- 解读某个特定应用的社会背景。
- 识别工程师可关联到计算机部件设计的非技术假设和价值观。

- 解释为什么网络空间中的“言论自由”对计算机工程是重要的。
- 描述计算机工程改变人与人之间的互动模式的积极和消极途径。
- 解释为什么在一些国家计算机/网络访问被限制。
- 举例说明在伦理辩论中,举例、类比和反类比的运用。
- 对比什么是法律什么是伦理。
- 解释道德诚信在计算机工程实践中的重要性。

### CE-PPP-3 有效沟通策略

达成核心学习成效所需最小学时数:2

核心学习成效:

- 在技术和非技术环境下都能用心倾听。
- 运用正确的拼写和语法撰写技术报告。
- 在撰写技术报告和做口头演讲时,运用令人信服的论据。
- 成为书面和口头的自信沟通者。
- 与听众建立融洽的关系。
- 培养书面和口头的有效沟通策略。
- 描述用身体语言影响沟通的方法。
- 使用适当的视频工具进行有效沟通。
- 在展示技术论文或报告时使用可视化技巧。
- 遵循指定的规则撰写技术报告。
- 运用适合于广泛受众群体的展现方式。
- 运用适合于广泛受众群体的写作风格。
- 积极回应听众的问题。
- 依据公认标准撰写组织良好、结构清晰的技术报告。

### CE-PPP-4 跨学科团队方法

达成核心学习成效所需最小学时数:1

核心学习成效:

- 描述跨学科团队在两种不同语境中的意义。
- 开发有效地运作一个跨学科团队可能需要的能力集合。
- 描述一些计算机工程项目,在这些项目中跨学科(融合)方法是很重要的。
- 探索工业界为达到一个共同目标的团队协作方法。
- 通过为每个团队成员分配角色和责任,为一个特定的项目创建一个跨学科

团队。

- 识别会破坏跨学科团队成员间交互的情况。
- 探索可以评估跨学科团队绩效的方法。
- 描述用于监督跨学科团队的可能的评估方法。

### **CE-PPP-5 哲学框架与文化问题**

达成核心学习成效所需最小学时数:2

核心学习成效:

- 总结相对主义、功利主义和义务论的基本概念。
- 描述与道德相对主义相关的一些工程问题。
- 描述对计算机工程困境而言科学和哲学方法之间的差异。
- 对比伦理理论和职业道德的不同。
- 识别“雇佣代理”方法、严格律法主义、天真利己主义、天真相对主义作为伦理框架的弱点。
- 对比适用于计算机工程领域的非西方哲学方法与思路。

### **CE-PPP-6 工程解决方案和社会影响**

达成核心学习成效所需最小学时数:2

核心学习成效:

- 阐述设计计算机系统时产品安全的重要性。
- 描述正确性、可靠性和安全性之间的差异。
- 解释用测试确保正确性的局限性。
- 描述风险、安全和可靠之外的其他社会的影响。
- 明确对错误的统计独立性的无根据假设。
- 论述重用现成组件的潜在的隐藏问题。
- 解释计算机工程师评估和管理风险的手段,以及如何向公众告知风险。
- 阐述公众认知的风险通常与实际风险不同的原因,以及这种不同可能产生的影响。
- 解释为何产品安全和公共消费应该作为计算机工程的一个品质证明。

### **CE-PPP-7 职业与道德责任**

达成核心学习成效所需最小学时数:3

核心学习成效：

- 识别一个举报事件的推进步骤。
- 详述用以表述职业素养的相关职业准则的优缺点。
- 明确职业准则如何成为决策的指南。
- 探索一些历史上软件风险的例子,如 Therac-25 案例。
- 提供支持和反对计算机工程执照的论据。
- 提供支持和反对非工程职业执照的论据。
- 识别软件开发中出现的伦理问题,确定如何在技术上和道德上解决它们。
- 开发一个使用强制措施的计算机使用政策。

### **CE-PPP-8 知识产权与法律问题**

达成核心学习成效所需最小学时数:3

核心学习成效：

- 描述知识产权的基础。
- 区分专利、版权和商业秘密保护。
- 区分专利和著作权。
- 列出一些有关知识产权的跨国问题。
- 论述著作权在国家与国际法中的法律背景。
- 解释不同国家的专利和版权法可能不同的原因。
- 列出软件专利的历史发展,并与其他形式的软件知识产权保护进行比较。
- 区分员工、承包商、顾问及各身份的影响。
- 了解一个计算机工程相关的专利并提供其内容概要。
- 解释产品和职业责任,并阐述其在计算机工程领域中的适用性。
- 分析免费开源的硬件和软件的道德后果。

### **CE-PPP-9 当代问题**

达成核心学习成效所需最小学时数:2

核心学习成效：

- 总结自己国家的隐私权和言论自由的法律基础。
- 讨论隐私如何因国而异。
- 描述当前由于计算机造成的对隐私的威胁。
- 对比病毒、蠕虫和特洛伊木马之间的差异。
- 解释互联网如何可能会在保护言论自由方面改变历史上的平衡。

- 阐述与海量数据库系统相关的一些对隐私的影响问题。
- 列出病毒以及拒绝服务攻击的技术基础。
- 提供计算机犯罪的例子,阐明一些犯罪预防策略。
- 在计算机工程领域中定义“侵入”/“骇”(cracking),并与“破解”/“黑”(hacking)对比。
  - 枚举用来对抗“骇客”攻击的技术。
  - 论述几个“骇客”的方法和动机。

### **CE-PPP-10 经营和管理问题**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 评估一个项目的总体工作成本。
- 运用工程经济原则考虑财务安排。
- 总结反垄断努力的基础理论。
- 描述由于劳动力供给短缺而影响信息技术产业的几种情形。
- 解释计算机工程师在考虑制造、硬件、软件和工程影响的情况下,估算出工作成本的几种方法。
  - 对比创业中的一些前景与陷阱。
  - 描述技术熟练工的供给和需求对计算产品质量的影响。
  - 总结反垄断影响的基本原理。
  - 对比可以用在计算领域的两种定价策略。

### **CE-PPP 附加知识单元**

#### **CE-PPP-11 职业素养中的折衷**

附加

选修学习成效:

- 指出一个计算机工程师在进行职业实践时可能要做的一些重要的折衷。
- 阐明一些做技术决定时道德的折衷。
- 描述一些在职业实践中可能发生的不道德的折衷。
- 评估进入自己所从事的行业的风险。
- 识别专业人员在安全方面的作用,以及所涉及的折衷。
- 描述安全和隐私之间的折衷,尤其像在后 911 时代所反映出的。
- 论证对访问计算资源进行限制的解决方法

## CE-SEC 信息安全

[核心学时数:20]

### 领域界定

本知识领域的知识单元组成如下。

1. 了解安全就是风险管理,并且内生地包括折衷。
2. 熟悉敌对用户,包括社会工程攻击和误用情况造成的影响。
3. 用于理解增强安全性的算法和其他技术措施的框架。
4. 信息安全的战略与战术设计问题。

### CE-SEC 核心知识单元

#### CE-SEC-1 历史沿革和综述

达成核心学习成效所需最小学时数:2

核心学习成效:

- 陈述著名的破坏安全和拒绝服务的例子。
- 论述常见计算机犯罪成本的估算方法和估算它们的困难性。
- 定义道德黑客。
- 对比主动和被动攻击。
- 论述计算机安全和隐私权相关的问题。
- 枚举攻击者的各种动机。
- 识别计算机犯罪的类型和攻击目标。
- 总结网络罪犯发起的攻击的主要类型。
- 论述安全方面的职业角色和所涉及的折衷。
- 给出历史和现代的密码算法的例子。
- 论证各类不同安全原则的应用,例如,纵深防御,功能和保证要求的矛盾,通过隐匿实现的安全是有缺陷的,安全就是风险管理,复杂是安全的敌人,以及负责任的公开漏洞披露的益处。
- 解释和论证各种安全机制的使用,如最小特权、失效-安全默认值、完全仲裁、特权分离和心理接受能力。

#### CE-SEC-2 相关工具、标准和(或)工程约束

达成核心学习成效所需最小学时数:2

核心学习成效:

- 论述相关的法律的主要规定,如 HIPAA 或欧盟数据保护令。
- 总结知识产权和出口控制法律对安全的影响,尤其是加密。
- 描述渗透式测试中常用的方法和工具。
- 阐明计算机取证的一些挑战。

### **CE-SEC-3 数据安全和完整性**

达成核心学习成效所需最小学时数:1

核心学习成效:

- 定义机密性和完整性。
- 给出只要完整性就足够的系统的例子。
- 定义“完美前向保密”(“perfect forward secrecy”),并解释为什么它是可取的。

### **CE-SEC-4 脆弱性:技术和人为因素**

达成核心学习成效所需最小学时数:4

核心学习成效:

- 定义误用情况(misuse cases),并解释其在信息安全中的作用。
- 描述人类行为在安全系统设计中的作用,包括社会工程攻击的例子。
- 完成一个简单的故障树分析。
- 解释 fuzz 测试可以揭示的错误类型。
- 论述更新已部署系统的困难的相关问题。
- 解释在系统安全中代码审查的作用。
- 定义不安全默认值的问题。
- 解释负责的漏洞披露中固有的折衷。
- 论述为什么在许多情况下优势总在攻击者一方,以及如何必须在系统设计中解决这个问题。
  - 解释如何执行栈溢出攻击和这个攻击所要求的知识。
  - 论述近期被利用的内存访问漏洞的例子,以及导致它们被部署和利用的错误。
    - 解释安全库和参数验证在防御缓冲区溢出方面的作用。
    - 举例说明“堆栈金丝雀”(stack canary)是如何工作的。
    - 解释地址空间随机化和非执行内存解决的问题。

- 定义几种类型的恶意软件,如病毒、蠕虫、木马、键盘记录器和勒索软件。
- 论述常见类型恶意软件的对抗措施。
- 解释当前恶意软件作者和防御系统作者之间“军备竞赛”的问题。

### CE-SEC-5 资源保护模型

达成核心学习成效所需最小学时数:1

核心学习成效:

- 解释各种自主和强制资源保护模式的利弊。
- 阐述访问控制矩阵模型。
- 定义 Bell-LaPadula 模型。

### CE-SEC-6 私密密钥和公开密钥加密

达成核心学习成效所需最小学时数:3

核心学习成效:

- 陈述将所有加密算法的变异性放在密钥上的动机。
- 论述处理能力对加密有效性的影响。
- 解释三种基本类型的密码攻击的含义以及它们之间的关系:唯密文,已知明文,选择明文。
  - 论述三种基本类型的密码函数之间的相似性和差异性:哈希,秘密密钥和公开密钥(零钥,一钥和二钥)。
  - 论述与秘密密钥加密算法相关的分组长度和密钥长度问题。
  - 描述和评估的一个对称加密算法,如高级数据加密标准(AES),重点说明设计和实现问题。
    - 解释一次一密加密机制的一些运用。
    - 进行取模运算(加法,乘法,和幂)。
    - 应用取模运算的基本理论(欧拉函数和欧拉定理)。
    - 执行和应用 RSA 算法进行加密和数字签名。
    - 执行和应用 Diffie-Hellman 算法建立一个共享秘密。
    - 展示和论述和应用秘密密钥(分组)加密算法对消息流进行加密的各种方法的动机和弱点,如密码分组链接(CBC)、密码反馈模式(CFB)和计数器模式(CTR)。

### CE-SEC-7 消息认证码

达成核心学习成效所需最小学时数:1



核心学习成效:

- 运用生日问题解释为什么哈希的长度大约需要两倍密钥长度。
- 论述哈希在指纹和签名中的运用。
- 论述加密哈希函数对比一般哈希函数的关键特性。
- 解释在加密哈希中使用的关键操作的目的,如置换和替换。
- 解释在消息认证码(MAC)中如何运用哈希。
- 陈述安全散列算法和诸如 SHA-2,SHA-3 系列或它们继承者的关键属性。
- 解释 HMAC 标准解决的问题。

### **CE-SEC-8 网络和 Web 安全**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 描述传输层安全(TLS)的目标和它们是如何使用私密密钥及公开密钥的方法以及证书达到这些目标的。
  - 论述使用防火墙的原因,以及各种拓扑结构和防火墙的局限性。
  - 图示和解释虚拟专用网络(VPN)的运用。
  - 描述拒绝服务攻击的常见方法,包括分布式和放大攻击以及在计算系统、协议设计和骨干网供应商层面的对抗措施。
    - 定义包过滤(防火墙)。
    - 论述 HTTP / HTTPS Web 平台的“无状态”设计带来的影响。
    - 描述与安全相关的 URL、HTTP 请求和 HTTP 摘要认证的基本结构。
    - 解释 HTTP cookies 的运用,包括会话 cookie、使用期限和对关键操作重新认证。
      - 定义跨站脚本。
      - 解释 SQL 注入攻击和各种补救方法。
      - 熟悉诸如开放式 Web 应用程序安全项目(OWASP)和 OWASP Top10 列表的标准。

### **CE-SEC-9 认证**

达成核心学习成效所需最小学时数:1

核心学习成效:

- 解释授权和认证的区别。

- 评论使用口令和(或)基于地址的认证方法。
- 论述搭线窃听和服务数据库读取,解释各种认证方法是如何对付它们的。
- 解释私密密钥和私钥系统对可信中介的一般运用。
- 论述对人进行鉴定的具体问题,包括三个主要方法。
- 描述包括生物识别技术在内的多因子身份验证方法解决的问题。

### **CE-SEC-10 可信计算**

达成核心学习成效所需最小学时数:1

核心学习成效:

- 描述当前可信计算的方法,如可信硬件、安全存储和生物识别技术。
- 评估对一个可信计算系统的规避方法,论述实施成本、信息的价值和规避难度之间的折衷。

### **CE-SEC-11 侧信道攻击**

达成核心学习成效所需最小学时数:1

核心学习成效:

- 论述各种侧信道以及在其上进行信息编码的方法。
- 论述侧信道保护和系统可用性之间的折衷。

## **CE-SGP 信号处理**

[核心学时数:30]

### **领域界定**

本知识领域的知识单元组成如下。

1. 信号采样和量化时所做出的权衡及其必要性。
2. 线性定常系统特性。
3. 与时域互补的频域。
4. 滤波器设计和实现。
5. 控制系统性质和应用。

## **CE-SGP 核心知识单元**

### **CE-SGP-1 历史沿革和综述**

达成核心学习成效所需最小学时数:1

核心学习成效:

- 解释计算机工程中数字信号处理和多媒体的目的与作用。
- 解释一些重要信号处理领域,如数字音频、多媒体、图像处理、视频、信号压缩、信号检测和数字滤波器。
- 运用采样和量化概念比较模拟和数字信号。
- 绘制数字信号处理框图并定义其主要组成部分:抗混叠滤波器、模拟/数字变换器、数字信号处理、数字/模拟变换滤波器和重建滤波器。
- 解释使用变换的必要性及对于模拟和离散信号的不同之处。
- 对比变换中运用的一些技巧,如拉普拉斯变换、傅立叶变换和小波变换。
- 指出低通和高通滤波器的设计标准。

### **CE-SGP-2 相关工具、标准和(或)工程约束**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 描述与提高采样率相关的权衡。
- 指出与包括采样周期的采样周期信号相关的关键问题。
- 指出与包括采样周期的采样非周期信号相关的关键问题。
- 鉴于输入到输出的映射,证明系统是线性的、不变时的、临时和(或)稳定的。
- 鉴于对线性、不变时系统输入输出特性的描述(视情况而定),推导非递归和递归差分方程。

### **CE-SGP-3 卷积**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 解释如何从线性和时不变性组合中产生脉冲响应的概念。
- 从脉冲响应和线性定义中导出线性卷积和。
- 以卷积的交换性为基础,为系统输入如何依赖于输入和系统脉冲响应提供两种解释。

### **CE-SGP-4 变换分析**

达成核心学习成效所需最小学时数:5

核心学习成效：

- 陈述、证明并应用  $z$  变换及其逆变换特性。
- 陈述、证明并应用离散时间傅里叶变换 (DTFT) 及其逆变换特性。
- 解释如何将离散时间傅里叶变换 (DTFT) 诠释为光谱。
- 解释原始和转换域之间的关系 (如混叠)。
- 陈述、证明并应用离散傅里叶变换 (DFT) 及其逆变换特性。
- 证明并陈述实信号傅里叶变换的对称性。
- 陈述傅里叶变换的频移特性。
- 证明并陈述傅立叶变换中, 帕塞瓦尔定理是如何与功率或能量, 视情况而定相联系的。
- 解释  $z$  变换、离散时间傅里叶变换 (DTFT)、离散傅里叶变换 (DFT) 和快速傅里叶变换 (FFT) 之间的关系。
- 定义并计算连续信号的拉普拉斯变换。定义并计算逆拉普拉斯变换。

### CE-SGP-5 频率响应

达成核心学习成效所需最小学时数:5

核心学习成效：

- 理解线性时不变系统的频率响应是单位冲激响应的另一种表示。
- 通过 DTFT 和 DFT 分析系统的频率响应。
- 给定系统的微分方程, 在  $z$  平面确定系统函数零极点的位置。
- 对系统进行  $z$  变换分析滤波器的频率选择特性。
- 理解频域采样与时间序列重复性的关联。

### CE-SGP-6 采样和混叠

达成核心学习成效所需最小学时数:3

核心学习成效：

- 陈述采样定理和相关奈奎斯特频率的混叠概念。
- 展现一个在正弦波上采样的混叠。
- 陈述与采样相关的时间和频域之间的关系。
- 解释光谱何时为离散的 vs 连续的。
- 计算采样和量化所产生的误差和噪音。

### CE-SGP-7 数字光谱和离散变换

达成核心学习成效所需最小学时数:6

核心学习成效:

- 概述周期信号光谱。
- 对比脉冲和方波光谱。
- 计算周期和非周期信号光谱。
- 解释区块大小是如何控制光谱分辨率和密度之间的权衡的。
- 计算光谱图并解释其关键参数。
- 解释在频域中以对数刻度过滤添加光谱。
- 应用正弦函数设计插值和重建滤波。

### CE-SGP-8 有限和无限脉冲响应滤波器设计

达成核心学习成效所需最小学时数:4

核心学习成效:

- 设计具有特定频率特性,包括幅度和相位响应的有限和无限脉冲响应(有限脉冲响应(FIR)和无限脉冲响应(IIR))滤波器。
  - 解释有限脉冲响应(FIR)和无限脉冲响应(IIR)滤波器之间的普通权衡。
  - 证明并非所有的递归滤波器都是无限脉冲响应(IIR)滤波器,如运用移动平均。
  - 运用离散傅里叶变换(DFT)完成(循环)卷积过滤。
  - 陈述有限脉冲响应(FIR)滤波器的线性相位。
  - 解释有限脉冲响应(FIR)滤波器的光谱分辨率、长度和延迟之间的权衡。
  - 解释为什么一种或多种有限脉冲响应(FIR)滤波器设计方法可以发挥作用。
  - 解释为什么一种或多种无限脉冲响应(IIR)滤波器设计方法可以发挥作用,包括运用零极点对的陷波滤波器。
  - 运用模拟技术(如双线性转换)设计数字滤波器并解释其关键参数。
  - 解释与滤波器设计相关的物理可实现系统,包括物理可实现性、时间变化和响应截断。

## CE-SGP 附加知识单元

### CE-SGP-9 窗函数

附加

选修学习成效:

- 解释窗函数是如何提高变换特性的。
- 解释光谱分析中的周期性假设。
- 解释窗函数目的及其对光谱的影响。
- 论述常见窗函数如矩形窗、布莱克曼窗、汉宁窗和哈明窗之间的权衡。
- 选择合适的窗函数来给出有关检测和识别之间权衡的问题描述。

### CE-SGP-10 多媒体处理

附加

选修学习成效:

- 定义在时间和(或)空间上变化的信号,并说明两个域中的频率。
- 解释采样是如何影响图像完整性的。
- 解释低通滤波是如何趋于平稳图像的。
- 比较重建和增强滤波器。
- 描述减少图像噪音的方法。
- 描述数字技术是如何在可感知或其他方面增强语音和音频信号的。
- 解释音频处理中的降噪(如维纳滤波或中值滤波)和除噪(如 LMS 滤波)技术。
- 解释音频编码的动机并描述 MPEG 编码的关键要素或相关算法,包括感知因素。

### CE-SGP-11 控制系统理论和应用

附加

选修学习成效:

- 定义基本的控制系统概念(例如,零状态响应,零输入响应,稳定性)。
- 比较控制系统的设计方法(根轨迹、频率响应、状态空间)。
- 解释与数字控制系统微控制器实现相关的局限性和折衷。
- 描述机电系统(包括机器人)等数字控制系统的潜在应用。
- 实现一个简单的、基于微控制器的、带有传感器和执行器的运动控制系统。

### CE-SPE 系统与项目的工程化

[核心学时数:35]

## 领域界定

本知识领域的知识单元组成如下。

1. 系统工程化原则在整个计算机系统的生命周期中的作用,包括在能耗力、性能和成本等方面的重要权衡。
2. 项目管理,包括团队管理、进度计划、项目配置、信息管理和项目计划的设计。
3. 人机交互风格和可用性要求、用户界面设计、输入/输出技术。
4. 基于计算机的系统中的关于期望的风险水平、可靠性、安全性和容错性分析以及设计方法。
5. 系统需求以及采集和分析基于计算机的系统需求的方法。
6. 系统规约、规约与需求和系统设计之间的关系,以及计算机系统开发和评估质量规约的方法。
7. 系统架构设计和评价,包括建模、仿真以及在架构层面上评估系统设计的工具和方法。
8. 用于并发硬件和软件设计、系统集成、测试和验证的方法和工具,包括单元和系统级测试计划。
9. 设计的可制造性、可持续性,以及在整个产品生命周期中的可维护性的设计。

## CE-SPE 核心知识单元

### CE-SPE-1 历史沿革和综述

达成核心学习成效所需最小学时数:1

核心学习成效:

- 阐明软件工程、硬件工程及计算机系统工程三者之间的差别。
- 简要解释系统、子系统的概念,讨论人、相关的不同学科的作用,以及跨学科方法对于拓展计算机系统应用范围时的必要性。
  - 指出计算机系统工程的一些重要的因素,如设计过程、需求、规约、设计、测试、验证、演化、项目管理、硬件软件接口和人机界面。
  - 定义和解释产品生命周期、系统工程在整个产品生命周期中的作用,以及许多基于计算机的系统设计成为持续演进系统的原因。
    - 指出计算机系统中测试、验证和维护之所以重要的原因。
    - 解释系统层面上设计决策和权衡的重要性,包括均衡成本、性能、功耗、可靠性和市场因素。

### **CE-SPE-2 相关工具、标准和(或)工程约束**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 选择并论证一组适当的工具,支持广泛的基于计算机的系统的开发,包括支持项目管理、需求和规约的定义与分析、配置管理、权衡分析的工具,以及关于软件、硬件和涉及建模、仿真、评估和测试等系统设计的计算机辅助工具。
- 在给定的计算机系统开发领域(例如,管理、建模或测试),分析和评估一组工具。
- 展现使用一系列工具开发中等规模的基于计算机的系统的的能力(可通过课内项目或大作业的形式来完成)。
- 解释标准、准则、法律、法规和职业素养等问题在基于计算机的系统开发过程中的重要性和影响。
- 描述遵循监管标准和法规情况下发生的折衷。

### **CE-SPE-3 项目管理**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 描述适用于各种应用,包括跨学科领域中基于计算机系统开发时项目管理的基本要素。
- 描述系统的生命周期的不同阶段并确定支持这些阶段的工具,包括项目管理工具,例如,用于项目规划、进度管理、成本分析、资源分配和团队合作的甘特图。
- 通过参与团队项目,展示团队建设和团队管理的核心要素,包括团队组成和组织、设计团队中的角色和责任、决策过程、项目跟踪和团队协作问题的解决等。
- 描述项目配置管理和项目信息管理的方法和工具,确保符合规范和及时交付。
- 为计算机系统设计项目准备一份项目开发计划,包括预期规模和成果、进度安排、资源保障、配置控制、变更管理和项目风险说明及管理(可通过课内项目或大作业的形式来完成)。

选修学习成效:

- 确定和描述在项目管理中使用的度量指标。
- 描述设计项目中顾问和分包商的作用,包括他们的使用和管理。
- 讨论标准和法律要求如何影响设计项目的管理。



### CS-SPE-4 用户体验<sup>1</sup>

达成核心学习成效所需最小学时数:6

核心学习成效:

- 定义用户体验(UX)。
- 描述从人的因素到用户体验设计(UXD)的演变过程。
- 对比 UXD 的物理和非物理两个方面的因素。
- 概括一些常见的人机交互风格,讨论人们是如何分析人与计算机的系统的交互。
  - 针对不同能力(如年龄、身体残障)的用户,说明对开发以人为中心的计算机系统有利的几个功能性和可用性需求。
    - 确定计算机工程中适用于不同应用和不同系统平台下高效图形用户界面(GUI)设计中的若干基本原则。
      - 讨论在用户体验系统(UX)环境开发中的折衷。
      - 确定适合于实现高质量多媒体接口的系统组件。
      - 以恰当的以人为中心的标准和可用性准则,评估一个已经存在的交互式系统,给出选择标准和技术的理由。
      - 讨论可视化技术在人机交互中的作用。
      - 描述诸如头脑风暴之类的创新技术是如何造就最佳用户界面的。
      - 解释在用户界面设计中社会心理学的重要作用。
      - 描述通用设计的两个主要原则。
      - 列举基于生物特征的访问控制方法的优缺点。
      - 描述允许严重身体残障人士使用网站的一个界面。
      - 使用给定的图形用户界面(GUI)生成器,对一个简单的二维图形用户界面,设计、原型化、并实施一个可使用性测试,并在这个过程中,产生一个合适的可使用性测试计划。

选修学习成效

- 讨论其他交互技术,如命令行接口和 shell 脚本。
- 说明在计算机的应用程序中广泛使用智能系统的可能性,描述在什么情况下智能系统能够或不足以提供可靠的响应。

---

<sup>1</sup>用户体验(UX)即以前的人机交互(HCI)。

### CE-SPE-5 风险、可靠性、安全性和容错性

达成核心学习成效所需最小学时数:3

核心学习成效:

- 认识适用于众多计算机系统的风险、可靠性和安全性需求,讨论将这些需求与系统性能和低功耗操作等其他需求权衡的可能性。
- 解释可靠性、可用性和可维护性作为衡量系统可信性的指标时的定义,以及它们与故障的关系。
- 完成一个中型的计算机系统的风险分析。
- 给出至少两项与开发安全攸关系统相关的权衡要素。
- 说明为什么知道如何从不可靠的组件建立可信系统的方法是十分重要的。
- 针对一个简单的计算机系统,展示可靠性、可用性和可维护性建模的能力。
- 描述实现预期可靠性、安全性和保密性的若干策略。
- 讨论硬件和软件故障的性质,以及用冗余实现容错的方法。
- 描述不同应用(如数据库、航空航天、电信、工业控制和事务处理)的容错性和可信性需求。

选修学习成效:

- 描述一种或多种降低风险和控制风险的策略,包括他们对系统实现的影响。
- 讨论与风险、安全和可信性相关的国际标准、法律要求和法规对设计计算机系统的影响。
- 讨论在高完整性系统设计中使用失效模式及影响分析(FEMA)和故障树分析的方法。
- 确定容错系统设计的一些硬件冗余方法,包括使用检错纠错码。
- 讨论一项或多项容忍硬件故障的软件方法。
- 描述一项或多项容忍软件故障方法,如N-版本程序设计、恢复块、回滚和恢复。

### CE-SPE-6 硬件和软件过程

达成核心学习成效所需最小学时数:3

核心学习成效:

- 解释需要一个专业化方法来开发系统的必要性,并说明在特定情况下,这类方法的诸要素。
- 描述生命周期的性质、生命周期模型的作用、与生命周期有关的质量、系统

性质的影响,以及选择生命周期模型的余地。

- 描述一些常见的软件和硬件开发模型,展示在开发一个计算机系统时如何应用这些模型。
- 解释如何收集数据以告知、评估和改进系统设计过程。
- 描述敏捷方法给硬件和软件设计带来的好处。
- 讨论模块化设计过程的重要性,并在一个计算机系统开发中应用模块化设计和重用模块。
- 选择用于各种不同类型计算机系统开发和维护时的最合适的系统开发模型,并陈述理由。

选修学习成效:

- 解释过程成熟度模型、标准和准则的作用。
- 确定软件、硬件和系统流程的若干度量。

### **CE-SPE-7 需求分析及其获取**

达成核心学习成效所需最小学时数:2

核心学习成效:

- 对一个给定的计算机系统设计项目,完成其分析,包括需求识别、信息收集、问题定义,可行性因素和经济因素。
- 针对适用于广泛应用的计算机系统的设计,阐明一系列功能和非功能性的要求,并讨论需求是如何随着系统设计项目的进展而改变的。
- 对于给定的计算机系统的设计,讨论对不同的系统需求进行权衡的必要性。
- 描述不同的方法来引导和获取需求的长处和短处。
- 应用一种或多种引导和分析技术,为中等规模的计算机系统完成需求。
- 描述用于衡量系统设计符合需求的程度的若干质量因素。
- 使用最佳实践确定文档的质量,完成一个计算机系统的需求文档的审查。

选修学习成效:

- 使用一个共同的、非形式化方法(如结构化分析或面向对象分析),以需求分析规范化文档的形式建模和描述一个中等规模的计算机系统的需求。

### **CE-SPE-8 系统规约**

达成核心学习成效所需最小学时数:2

核心学习成效:

- 讨论系统规约和需求之间的关系和差异。
- 阐明设计基于计算机的系统时一些典型的功能和非功能规约,并说明规约在设计过程中的重要性。
  - 讨论从需求文档讨论导出系统规约的一种或多种方法。
  - 讨论为满足系统的需求而在不同的系统规约之间权衡的必要性。
  - 考虑完整性、一致性、简单性、可验证性、设计依据、失效规约、降级运行模式等因素,评估一份给定规约的质量。
  - 对于给定的一组需求,为中等复杂程度的基于计算机的系统创建一个高质量的规约。
    - 根据规约,并考虑测试、安全测试用例和测试局限性之间的独立作用,创建一个测试计划。

选修学习成效:

- 描述和展示一个或多个形式化规约语言和技术的运用。
- 将按照常用的形式化规约语言描述的系统规约翻译成自然语言。

### **CE-SPE-9 系统架构设计与评价**

达成核心学习成效所需最小学时数:4

核心学习成效:

- 描述系统架构设计的概念和原则,如自顶向下的设计、细分为系统和子系统、模块化和重用、硬件/软件接口,以及各种设计方案之间的权衡。
  - 描述各种系统级架构设计方法的长处和短处,包括过程化和函数化方法。
  - 描述为满足系统规约并实现包括可信性和安全性等性能指标的设计方法。
  - 对于给定的系统规约,选择合适的设计方法(例如,结构化设计或模块化设计),为一个中等规模的基于计算机的系统创建一个架构设计。
    - 展现对一些基于计算机的系统架构进行建模、仿真和原型化的能力。
    - 使用适当的指导方针,完成一个或多个基于计算机的系统设计的审查,以根据关键设计原则和概念来评估设计质量。

选修学习成效:

- 在架构层面,讨论可能的失效模式、常见的故障原因和故障处理,包括故障的诊断以及实现容错设计的方法。
  - 讨论与实现可靠性相关的设计问题、冗余的作用、设计独立性、关注点分离

以及子系统规约。

### **CE-SPE-10 并发硬件和软件设计**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 认识到合适环境下硬件-软件协同设计的潜力。
- 讨论为何在特定的设计约束条件下,如设计低功耗系统、实时系统、有很高性能要求的系统的时候,硬件和软件的协调开发会变得十分重要。
  - 在适度复杂的情况下应用硬件软件协同设计原则。
  - 讨论有效的硬件-软件协同设计面临的挑战,例如,当存在硬实时的需求时。
  - 展示为实现低功耗、实时操作、高性能等特定的技术目标时的协同设计能力。
- 选择和应用计算机辅助工具来支持硬件和软件协同设计。

### **CE-SPE-11 系统集成、测试和验证**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 了解系统生命周期各阶段适用的测试与验证方法,例如,硬件模型审查、软件代码审查,白盒、黑盒和回归测试,压力测试,接口测试等。
  - 描述各种系统验证工具的作用,并展示工具是如何高效地支持开发的。
  - 讨论在单元级、集成级和系统级的测试与验证方法。
  - 为中等规模复杂度的基于计算机的系统创建一个测试计划,并生成测试用例,选择适当的测试组合确保系统的质量。
    - 在中等规模的基于计算机的系统上演示不同类型和级别的测试(单元测试、集成测试、系统测试和验收测试)。
    - 作为团队活动的一部分,观察一个中等规模的基于计算机的系统的设计。
    - 讨论用于制造测试与检验以及验收测试的方法。

选修学习成效:

- 讨论针对安全性、可靠性/容错性、可用性的专门测试方法。

### **CE-SPE-12 可维护性、可持续性、可制造性**

达成核心学习成效所需最小学时数:2

核心学习成效:

- 描述易维护软件、硬件和系统设计的必要性及其性质。
- 讨论在某些系统中的维护的必然性,例如,诊断、清除缺陷,硬件和(或)软件升级,以及增强。
  - 描述在适度复杂的基于计算机的系统上应用可维护性设计原则的方法。
  - 确定与系统演化相关的问题,并解释他们对系统生命周期的影响。
  - 解释在工程系统中的配置管理和版本控制:必要性、相关问题、必须保存的信息的性质、法律要求,以及应对可能灾难的规划。
    - 针对一个中等规模产品变化请求,开发一个再造工程计划。
    - 在各种上下文中发现和利用组件重用的机会。
    - 讨论设计决策如何影响后代,包括对环境和能源资源的影响,以及在使命结束时系统和组件的处置。
      - 讨论面向制造的设计、部件选择与标准化、制造成本和产品交付时间。

## CE-SRM 系统资源管理

[核心学时数:20]

### 领域界定

本知识领域的知识单元组成如下。

1. 具有多种组件的计算系统中的资源管理。
2. 实时操作约束及其对系统资源管理的影响。
3. 移动环境中的资源管理。
4. 在不同的操作环境中与资源管理有相关的权衡。

## CE-SRM 核心知识单元

### CS-SRM-1 操作系统的历史与综述

达成核心学习成效所需最小学时数:1

核心学习成效:

- 解释操作系统的目的及其提供的服务。
- 描述移动、网络、客户机-服务器、分布式等操作系统以及单用户系统中的功能差异。
  - 定义关键的设计标准,包括效率、健壮性和安全性。
  - 解释操作系统面临的主要威胁,以及如何防范这些威胁。

### CE-SRM-2 相关工具、标准和(或)工程约束

达成核心学习成效所需最小学时数:1

核心学习成效:

- 举例说明实时性能监控工具和基于日志的性能监控工具。
- 解释实时性能监测工具可提供些什么信息,这类工具何时有用。
- 解释基于日志的性能监测工具提供些什么信息,这类工具何时有用。
- 列举 IEEE POSIX (可移植操作系统接口)的关键部件。
- 定义一些关键的系统资源管理 API 的作用,如 WinAPI 和各种 Java APIs。

### CE-SRM-3 管理系统资源

达成核心学习成效所需最小学时数:8

核心学习成效:

- 描述操作系统在管理系统资源和软硬件之间接口中的作用。
- 解释什么是并发并说明为什么在系统资源管理中必须得到它的支持。
- 举例说明多任务或多组件系统中由于并发导致的运行时问题,如死锁和竞争条件。
  - 描述中断的基本类型,以及处理中断时必须做些什么。
  - 举例说明为什么在系统资源管理中需要任务调度和分派。
  - 解释抢占式调度和非抢占式调度之间的区别,展示对于常用调度算法的了解。
  - 描述如何使用中断、任务调度和上下文切换来支持并发。
  - 解释内存层次结构(从 Cache 直至虚存)以及设计中的性价比权衡。
  - 描述文件系统设计中的选择,以及这些选择对系统资源管理的影响。

### CE-SRM-4 实时操作系统设计

达成核心学习成效所需最小学时数:4

核心学习成效:

- 解释硬实时、软实时、安全攸关实时环境的特点。
- 讨论内存层次结构设计和磁盘或其他固定存储设计中出现的不确定性问题,并讨论解决这些问题的方法。
  - 解释延迟及其在 RTOS 设计中的作用。
  - 解释事件驱动的调度程序如何运行,并能够给出常用算法的例子。

- 解释轮转调度与事件驱动策略的差异。
- 解释为什么内存分配在一个实时系统中是至关重要的。
- 解释适用于 RTOS 的失效模式和恢复策略。
- 讨论各种操作系统选项之间的权衡。

### **CE-SRM-5 移动设备的操作系统**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 描述在移动设备(如 Wi-Fi、蓝牙)中必须管理的系统通信。
- 讨论在移动环境中固有的约束。
- 解释由用户提出的移动操作系统需求。
- 讨论实施跨移动平台的挑战。
- 讨论安全威胁的来源及其管理。

### **CE-SRM-6 并发处理的支持**

达成核心学习成效所需最小学时数:3

核心学习成效:

- 解释并举例说明并发处理的基本概念,如多处理器、多核、SIMD、MIMD、共享内存以及分布式存储。
- 解释支持多线程调度需要些什么。
- 描述同步多线程(SMT)是如何执行的。

## **CE-SRM 附加知识单元**

### **CE-SRM-7 系统性能评估**

附加

选修学习成效:

- 描述为什么在重要的应用程序(如关键任务系统)中性能是重要的
- 解释为什么诸如响应时间、吞吐率、延迟、可用性、可靠性和功耗等指标在性能评估中非常重要。
- 解释重要性能评估指标必须考量什么内容。
- 描述常用的基准测试工具集的优势及其局限性。
- 展示对常用的(确定的、随机的、模拟的)评估模型的了解并分别说明其适



用环境。

- 解释如何收集分析和跟踪数据,并说明怎么用以评估系统的性能。

### **CE-SRM-8 虚拟化支持**

附加

选修学习成效:

- 定义虚拟机管理程序或虚拟机监控程序的作用。
- 描述宿主机的作用以及它与虚拟机的关系。
- 描述什么是宿主机操作系统,以及它与虚拟机操作系统的关系。
- 解释什么是原生的虚拟机管理程序以及它与托管虚拟机管理程序间的区别。
- 举例说明虚拟化环境所导致的隔离和安全问题。

### **CE-SWD 软件设计**

[核心学时数:45]

#### **领域界定**

本知识领域的知识单元组成如下。

1. 编程范式和结构。
2. 数据结构以及操作它们的标准库函数的运用。
3. 面向对象设计及建模语言的运用。
4. 软件测试和软件质量概念。
5. 不同软件设计方法的权衡。

#### **CE-SWD 核心知识单元**

##### **CE-SWD-1 历史沿革和综述**

达成核心学习成效所需最少学时数:3

核心学习成效:

- 解释早期的软件都是用机器语言或汇编语言编写的原因。
- 给出一些早期程序设计语言的名称并列出它们的主要特点。
- 给出一些交互式用户界面的里程碑的例子。
- 描述软件开发环境随时间变化的发展过程。
- 解释高级语言对提高程序产出的重要性。

- 列出软件生命周期各步骤的名称,并给出具体的定义。

### **CE-SWD-2 相关工具、标准和(或)工程约束**

达成核心学习成效所需最少学时数:3

核心学习成效:

- 明确软件开发工具,如编译器、汇编器、链接器和调试器的作用。
- 明确典型的现代集成开发环境(IDE)的预期功能。
- 使用 IDE 开发一个简单的应用。
- 有效运用调试器跟踪代码执行过程并发现其中的错误。

### **CE-SWD-3 编程结构和范式**

达成核心学习成效所需最少学时数:12

核心学习成效:

- 解释一个简单程序的执行过程。
- 编写简单可靠的程序完成预期的任务。
- 编写简单的函数,解释其中形参和实参的作用。
- 设计、实现、测试并调试一个通过不平常的方式使用基本编程结构的程序。
- 选择合适的循环和条件结构来完成一个确定的程序任务。
- 对比命令式(如过程)、说明式(如函数)、结构化(如面向对象)等软件设计范型。
- 定义并解释好程序的要素(包括避免安全漏洞)。

### **CE-SWD-4 问题求解策略**

达成核心学习成效所需最少学时数:5

核心学习成效:

- 区分现实中使用不同问题求解策略的实际例子。
- 运用分而治之的策略来求解问题。
- 运用贪心方法解决一个合适的问题,并判断该方法是否得到了最优解。
- 解释启发式方法在问题求解中的作用。
- 讨论如何权衡不同的问题求解策略。
- 针对给定的问题,判断求解过程中使用的求解策略是否合适。

选修学习成效:

- 运用动态规划解决合适的问题。

### **CE-SWD-5 数据结构**

达成核心学习成效所需最少学时数:5

核心学习成效:

- 说明基本数据结构的相关概念、用途以及它们之间的权衡。
- 说明如何运用高质量的程序库来创建和查找数据结构。
- 解释什么是哈希表,它的用途是什么?说明冲突避免以及冲突解决的作用。
- 解释什么是二叉查找树并说明维持平衡对算法性能的影响。
- 运用库函数解决标准数据结构(链表、已排序数组、树和哈希表)的插入、删除、查找和排序问题(而不是简略实现其算法)。

### **CE-SWD-6 递归**

达成核心学习成效所需最少学时数:3

核心学习成效:

- 描述递归的概念并举例说明其用途。
- 解释循环和递归的关系。
- 明确用递归定义的问题中的基准情形和普遍情形。
- 列举出递归程序运行时可能产生的问题。

### **CE-SWD-7 面向对象程序设计**

达成核心学习成效所需最少学时数:4

核心学习成效:

- 将一个问题分解为对象的类来描述,类中应包含相关的状态(数据成员)和行为(方法)。
  - 对比方法重载和方法覆盖,并举例说明。
  - 论述编译时方法绑定和运行时方法绑定的优缺点。
  - 使用模型语言(如 UML)来展示一个简单的类继承,该类中有允许代码在不同子类重用的子类结构。
    - 解释当类中方法名被重载或覆盖的时候,对其进行函数调用时消除歧义的机制。

- 在面向对象程序设计的情景下,解释信息隐藏、耦合和聚合、数据抽象等概念的含义。

### **CE-SWD-8 软件测试、验证和确认**

达成核心学习成效所需最少学时数:5

核心学习成效:

- 解释测试、验证和确认的不同含义。
- 展示对于单元测试策略(白盒、黑盒、灰盒等)及其取舍标准的理解。
- 展示对于不同的验证和确认策略的理解。
- 在模块的单元测试过程中,创建一个测试数据集,使用该数据集进行测试并给出测试报告。
- 解释单元测试和集成测试的不同点。
- 解释被广泛应用的软件质量指标。
- 描述至少一种自动测试工具和(或)测试模式生成器。

### **CE-SWD-9 数据模型**

达成核心学习成效所需最少学时数:2

核心学习成效:

- 解释并举例说明数据模型及其应用。
- 采用标准的模型符号(如UML)表示并记录计算机工程问题中的一个合适的数据库模型。

### **CE-SWD-10 数据库系统**

达成核心学习成效所需最少学时数:3

核心学习成效:

- 解释从直接对一些数据文件编程到使用数据库系统的发展过程。
- 描述现代数据库系统的主要组成。
- 描述SQL之类的语言所提供的功能。
- 举例说明计算机工程领域中数据库系统的运用。

### **CE-SWD-11 事件驱动和并发程序设计**

达成核心学习成效所需最少学时数:2

核心学习成效：

- 解释与移动系统和位置敏感系统有关的一些特定的问题,包括安全问题。
- 解释在多核处理器上编出高效程序的难点。
- 展示在多核处理器上编程的能力。

## CE-SWD 附加知识单元

### CE-SWD-12 使用应用程序接口(API)

附加

选修学习成效：

- 设计和实现可重用的函数。
- 给出向后兼容和向前兼容的定义,并提出解决相关问题的方案。
- 利用常见的应用程序接口(API)为系统服务、数据结构、网络通信等写一些代码。

### CE-SWD-13 数据挖掘

附加

选修学习成效：

- 解释数据挖掘在计算机工程中的作用。
- 说明计算机工程中机器学习的作用。

### CE-SWD-14 数据可视化

附加

选修学习成效：

- 对数据进行可视化操作,以提高对数据的理解,方便相关信息的交流,帮助进行决策。

## 附录 B 计算机工程课程体系实例

本附录给出了若干真实的课程体系实例,说明如何按照本报告提供的知识体系详细要求来实现专业培养方案。这些课程体系的学习时限不同,有三年的和四年的。虽然四年的课程是最常见的,但是这些不同的例子表明,在不同的机构要求和资源约束下,不同特点的本科专业课程都可以落实本报告中的这些建议。这些例子并不是指定的标准,他们只是服务于不同的教育目标和学生需求。

下表概括了本附录中的课程体系实例。表中提供了课程体系的来源作为参考,以便用于特点和需求相似的机构。

实例	实施机构
A	由电子与计算机工程系开设
B	由计算机科学系开设
C	由计算机科学系和电子工程系联合开设
D	来自中华人民共和国
E	来自博洛尼亚-3

### B.1 格式和约定

本附录中的所有课程体系实例采用共同的格式,都包括 5 个部分。

1. 本专业的培养目标,开设该专业的机构或院系的特色,以及资源约束。
2. 获得学位的培养要求汇总,以表格的形式表示,指明对应的课程内容。
3. 通常情况下一个学生的课程安排计划表的示例。
4. 计算机工程知识体系与课程体系之间的映射关系。
5. 课程体系中各课程的具体说明。

#### B.1.1 课程学时的约定

这里采用课程编号来表示课程在本专业中所处的层次。例如,课程编号  $MTH_x$  100 表示这是 X 课程体系的一门课,通常在第一年讲授(大学一年级水平)。同样,  $PHY_x$  200 表示通常在第二年讲授的课程(大学二年级水平);  $ECE_x$  300 表示通常在

第三年讲授(大学三年级水平);课程 ECE<sub>x</sub>400 则一般是第四年讲授的课程(大学四年级水平)。

为了便于比较,所有的课程实施方案都按照美国教育系统的模式呈现,即一个学期包含 14 周的课堂教学和实验。通常,有一个星期用于考试、放假和阅读时间。为简单起见,用“学时”来描述课堂和实验的时间。讲座或实验的一个“学时”通常是一段持续 50 分钟的时间。

根据给定的一周内正式活动的数量和类型,我们为每门课程分配了一学期的学分,安排如下。

- 课堂教学时间:在教室环境中的教学材料的讲解。

1 学分=每周 1 次 1 学时的讲座

- 实验时间:在实验室环境中的正式实验。

1 学分=每周 1 次 3 学时的实验室

下面是一个计算课堂教学和实验学分的示例,其 1 学时是指持续 50 分钟的时间段。

- 3 学分的课程:

每周 3 次课堂教学,14 周=42 课堂学时(加考试 1 周)。

- 1 学分实验室课程:

每周一次 3 学时的实验室活动,14 周=42 实验室学时。

- 3 学分的课程,含每周两次课堂教学和一次实验室活动:

每周 2 个课堂教学学时,14 周=28 课堂学时(加考试 1 周)。

每周 1 次 3 学时实验室活动,14 周=42 实验室学时。

- 3 学分的工程设计课程:

每周 1 次课堂会议,14 周=14 学时(加考试 1 周)。

实验室 2 学分=每周 6 学时的实验室活动,14 周=84 实验室学时。

## B.1.2 计算机工程知识体系到课程体系实例的映射

每个课程体系实例都有各一个映射表,详细描述了计算机工程知识体系到课程体系实例的映射。表格中的行标题是课程编号,列标题是知识体系中的知识领域。如果表格中的某个空格中有数字,那就表明这一行对应的课程覆盖了这一列对应的知识领域,数字代表所覆盖的知识单元的编号。例如,知识领域 DIG 对应的数字“1-3,5”表明这个课程涵盖了来自 CE-DIG 的第 1、2、3 和 5 知识单元。需要注意的是:

- 一门课程可能包含同一个知识领域的知识单元,也可能包含多个知识领域的知识单元。

- 相同的知识单元可能出现在多个课程。例如,数字设计的两个不同课程都可以包含 DIG-1/2 知识单元,因为两门课程都要包括“历史和相关工具/标准/约束”这些内容,但是从不同的角度来讲解。

表格的第二行列出了每个知识领域的最低核心学时,以作为参考。表格的最后两行列出了该课程体系所覆盖的核心知识体系单元和附加知识体系单元的总体情况。因为所有的课程体系实例都完整覆盖了核心知识体系单元,所以“核心知识体系单元”那一行包含了所有核心知识单元的知识领域。而“附加知识体系单元”那一行有可能有空白的格,有数字的格则是列出了被覆盖的附加知识体系的知识单元。课程体系实例不会覆盖所有附加知识体系单元,而且覆盖情况也不代表优先级或者推荐程度。

### B.1.3 课程描述

课程描述通常会出现在课程目录中。受限于篇幅,这些简短的描述不可能详细列出课程中讲授的知识点。因此,需要增加一个列表,用以描述这些课程所覆盖的知识体系中的知识领域以及知识单元。对于包含实验的课程,课程描述中不会包括实验过程的细节。第 4.4 节描述实验过程的整体情况,包括团队合作、数据收集和分析以及其他技能。

## B.2 开始职业生涯的准备

本附录中的课程体系实例的主要目标是让毕业生为开始计算机工程的职业生涯做好准备工作。有很多方法来建设本科课程体系,以培养受过良好教育的计算机工程师。为了强调这一点,本节描述的学习方式有很多不同,包括强调的重点和制度约束等,但这不代表所有的专业都有这样的结构设置。例如,许多专业在一年级课程中向学生介绍本学科,并提供实际的工程经验,以此激励学生。如果在课程体系实例中没有这类课程,也不会影响对这类课程的价值判断。

CE2016 课程指导委员会设计的这些课程,是为了确保本报告定义的计算机工程知识体系能够得到恰当地覆盖,以获得对应的学习成效。然而,正如在报告正文中讨论的那样,创建一个有利于毕业生做好实践准备的计算机工程专业,还有很多其他要素,如设计经验和实验室经验、口头和书面交流能力,以及现代工程工具运用能力。因此,职业认证不仅仅是关注课程。对此有兴趣的读者应该从认证机构中参考相关的标准。

此外,每个计算机工程专业都可能各自有不同的教育目标,没有直接反映在计算机工程知识体系和在本报告中提出的课程模式中。每个专业都有责任确保学生获得该专业教育目标所规定的学习结果。



## B.3 课程体系的共性

对于有志于学习计算机和数字系统应用的学生,计算机工程专业的学习会是非常有价值的经历。学生期望能够精通计算机工程领域的硬件、软件和系统方面的技能,包括计算机和数字系统的设计、分析、开发和应用。该专业的学习内容密集,很有挑战性。学生将会掌握实用的技能,涵盖计算机系统、体系结构、网络 and 进程的分析、设计和测试。

每个课程体系实例都是以获得计算机工程学士学位为目标的,并以遵循硬件和软件的平衡处理原则,建立计算机科学和电气电子工程相结合的理论、分析和设计的宽广基础。此外,前三个课程体系实例(A、B和C)强调建立宽广的基础,包括科学、离散数学和连续数学,以及美国通识教育中常见的其他方面。其余的课程体系实例(D和E)说明了在中国、英国等世界其他地区典型的计算机工程专业。实例E是与博洛尼亚宣言一致的,即通过本科三年获得工程专业的学士学位。

通常要求课程允许重复的螺旋式学习。每个课程体系都应有足够的灵活性,以支持不同领域的专门课程。每个专业结构应有宽广的基础课程,并提供大量的专业选修课。当学生已经深入学习了核心领域的知识后,应该进行最终的设计体验。每门课程的建设都是理论、实践、应用、态度相结合。

每个专业的目标都是通过打好终身学习和发展的基础,让学生为计算机工程职业生涯做好准备。每个专业还提供了进一步发展的平台,包括计算机工程专业以及商业、法律、医学、管理和其他领域的研究生学习。学生从第一门课程开始,从程序设计、电路分析、数字电路、计算机体系结构、网络等逐步提高设计技能,并将其积累的知识应用到课程中的实际问题。最终设计课程将补全课程体系的分析部分。

任何专业的毕业生都应为就业或深造做好充分的准备。他们应该了解计算机工程的各个领域,如应用电子设备、数字设备和系统、软件设计、计算机体系结构、系统和网络。毕业生应能够将所学到的知识和技能应用到计算机工程领域和其他领域。他们也应该拥有设计的技能,并能深刻理解硬件问题、软件问题、模型,以及这些问题之间的相互作用,还有与之相关的应用。计算机工程课程体系所提供的广泛教育的准备,有助于在全球化社会背景下了解工程解决方案的影响。

## B.4 课程体系 A:由电子与计算机工程系开设

(一个由电子与计算机工程系制定的计算机工程专业培养方案)

### B.4.1 培养目标和特色

该培养方案的目标是获得计算机工程学士学位,这个学位一般情况下是由一个

传统的电子与计算机工程(ECE)系提供的。计算机科学系有时会为该培养方案提供计算机科学方面的基础课程,如程序设计。而电子和计算机工程系则提供其他课程的教学。像大多数工程专业一样,本培养方案具有比课程体系 B(后续介绍)更小规模的通识教育,而是把更多的学时用到计算机工程教学中。本培养方案通常在电路/电子、数字设计、嵌入式系统和计算机程序设计方面提供一个多课程的系列,通过在计算机体系结构、操作系统、网络 and 计算机安全方面的课程来拓宽知识面。它更加面向硬件和系统设计,但是,在软件方面也包含了足够的内容,以使毕业生能够成为一个具有实力的计算机工程师。

### B.4.2 培养要求汇总

该培养方案的学习内容包含计算机科学的 5 门必修课(17 个学分)和电子与计算机工程的 14 门必修课(42 个学分)。有两门计算机工程选修课(6 个学分)可以灵活设置。这两门课程可以从计算机科学系或电子与计算机工程系提供的课程中选择。有 10 门课程提供了实验,为学生提供了大量的使用现代工具和设计技术的实际操作经历。大四年级会有两门以上课程提供毕业设计类实验体验,要求学生完成规模较大的、完整的设计实验。必修的技术写作课程还可以增强口头表达和书面交流能力,通过实验和毕业设计类课程项目也可以增强这些能力。该课程体系的学分数如下。

学分数	领域
21	数学
10	自然科学(物理、化学)
24	英语写作、人文和社会科学
38	电子与计算机工程必修课
17	计算机科学必修课
6	专业选修课(来自 ECE 或 CSC)
4	工程设计
0	自由选修
120	计算机工程培养方案的总学分

### B.4.3 课程体系 A 的四年制模式

CE: 可以由计算机工程系开设

ECE:由电子与计算机工程系开设

CSC:由计算机科学系开设

CE 专业选修:由 ECE 和 CSC 开设的选修课程

课程	描述	学分	课程	描述	学分
<b>第 1 学期</b>			<b>第 2 学期</b>		
MTH 101	微积分 I	3	MTH 102	微积分 II	3
CHM 101	化学 I 和实验	4	PHY 101	物理 I	3
CSC <sub>A</sub> 101	计算机程序设计入门	4	CSC <sub>A</sub> 102	中级计算机程序设计	4
	英语写作 I	3	ECE <sub>A</sub> 101	ECE 介绍	2
	人文学选修	3		英语写作 II	3
	总学分	17		总学分	15
<b>第 3 学期</b>			<b>第 4 学期</b>		
MTH 201	微积分 III	3	MTH 203	微分方程	3
PHY 201	物理 II	3	MTH <sub>A</sub> 204	离散结构	3
ECE <sub>A</sub> 201	数字设备	4	ECE <sub>A</sub> 202	微处理器	4
CSC <sub>A</sub> 201	数据结构和算法分析	3	ECE <sub>A</sub> 203	电子电路入门	3
MTH 202	线性代数	3	MTH 205	概率和统计	3
	总学分	16		总学分	16
<b>第 5 学期</b>			<b>第 6 学期</b>		
CSC <sub>A</sub> 301	算法入门	3	CSC <sub>A</sub> 302	分布式客户端/服务器程序设计	3
ECE <sub>A</sub> 301	中级电子电路	4	ECE <sub>A</sub> 303	信号与系统	3
ECE <sub>A</sub> 302	数字系统设计	3	ECE <sub>A</sub> 304	数据通信网络	3
	人文学选修	3	ECE <sub>A</sub> 305	计算机体系结构	3
				社会科学选修	3
	总学分	13		总学分	15
<b>第 7 学期</b>			<b>第 8 学期</b>		
ECE <sub>A</sub> 401	CE 设计 I	2	ECE <sub>A</sub> 402	CE 设计 II	2



续表

BoK 领域 课程	C A E	C A L	C A O	D I G	E S Y	N W K	P P P	S E C	S G P	S P E	S R M	S W D	A C F	D S C	L A L	P R S
ECE <sub>A</sub> 302				1, 2, 6-11												
ECE <sub>A</sub> 303									1-7							
ECE <sub>A</sub> 304						1-11										
ECE <sub>A</sub> 305			1-11													
ECE <sub>A</sub> 401							1-6, 11			7-10						
ECE <sub>A</sub> 402							7-11			10- 12						
ECE <sub>A</sub> 403					9-13					10	4,6	11				
ECE <sub>A</sub> 404								1-11								
ECE <sub>A</sub> 405		9									1-8					
MTH202															1-10	
MTH203													1-7			
MTH <sub>A</sub> 204														1-9		
MTH205																1-10
涵盖的核心知 识体系单元	1-10	1-8	1-11	1-11	1-13	1-11	1-11	1-11	1-7	1-12	1-6	1-10	1-7	1-9	1-10	1-10
附加知识体系 单元		9									7-8	11,12				

### B. 4.5 课程体系 A-课程概述

#### CSC<sub>A</sub>101: 计算机程序设计入门

问题求解和使用面向对象技术进行计算机程序设计的概述性介绍;程序设计和问题求解方面的理论和实践。

先修要求: 大学代数或相当基础。

学分: 4;授课学时: 42;实验学时: 42。

覆盖 BoK: CE-SWD 1-4。

#### CSC<sub>A</sub>102: 中级计算机程序设计

面向对象的问题求解、设计和编程;数据结构、算法设计和复杂性简介。

先修要求: CSC<sub>A</sub>101。

学分: 4;授课学时: 42;实验学时: 42。

覆盖 BoK: CE-SWD 4-8。

#### CSC<sub>A</sub>201: 数据结构和算法分析

非线性数据结构及其相关算法;树、图、哈希表、关系数据模型、文件组织;高级软件设计与开发

先修要求: CSC<sub>A</sub>102 以及大学代数或相当基础。

学分: 3;授课学时: 42;实验学时: 0。

覆盖 BoK: CE-SWD 5-9。

#### CSC<sub>A</sub>301: 算法入门

算法的复杂性研究以及算法设计;有效性分析工具;算法设计,包括递归、分治、动态规划和贪心算法。

先修要求: CSC<sub>A</sub>201、MTH 201 和 MTH<sub>A</sub>204。

学分: 3;授课学时: 42;实验学时: 0。

覆盖 BoK: CE-CAL 1-8。

#### CSC<sub>A</sub>302: 分布式客户端/服务器程序设计

分布式环境中的软件系统设计;客户端/服务器模型、多线程编程、服务器端 Web 编程、图形用户界面、客户端/服务器系统中的小组实验项目。

先修要求: CSC<sub>A</sub>201。

学分: 3;授课学时: 28;实验学时: 42。

覆盖 BoK: CE-SWD 10, CE-SPE 1-12, CE-SWD 12。

**ECE<sub>A</sub>101: ECE 介绍**

一名工程师意味着什么,工程师行为准则,工程建模,设计过程,ECE 领域,沟通能力。

先修要求:修完或正在修 MTH 101。

学分:2;授课学时:15;实验学时:42。

覆盖 BoK:CE-PPP 1-3,5。

**ECE<sub>A</sub>201: 数字设备**

二进制编码,布尔值,代数,组合逻辑设计,触发器,计数器,同步时序逻辑,可编程逻辑设备,MSI 逻辑设备,加法器电路。

先修要求:修完或正在修 CSC<sub>A</sub>101。

学分:4;授课学时:42;实验学时:42。

覆盖 BoK:CE-DIG 1-9。

**ECE<sub>A</sub>202: 微处理器**

微处理器系统体系结构;微处理器操作、汇编语言、算术运算、接口。

先修要求:ECE<sub>A</sub>201, CSC<sub>A</sub>201。

学分:4;授课学时:42;实验学时:42。

覆盖 BoK:CE-ESY 1-8,CE-CAO 3,5,CE-DIG 3。

**ECE<sub>A</sub>203: 电子电路入门**

电子电路基础和网络分析;瞬态分析和网络频率响应;运算放大器介绍;AC 电源。

先修要求:PHY 201, MTH 202, 修完或正修 MTH 203。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-CAE 1-4。

**ECE<sub>A</sub>301: 中级电子电路**

运算电路模型和二极管的应用、场效应和双极型晶体管;电子仪器;电子通信系统基础。

先修要求:ECE<sub>A</sub>203。

学分:4;授课学时:42;实验学时:42。

覆盖 BoK:CE-CAE 5-10。

**ECE<sub>A</sub>302: 数字系统设计**

基于逻辑综合的现代数字设计技术, 硬件描述语言; 现场可编程门阵列 FPGA, 模块化构建技术。

先修要求: ECEA202。

学分: 3; 授课学时: 28; 实验学时: 42。

覆盖 BoK : CE-DIG 1, 2, 6-11。

**ECE<sub>A</sub>303: 信号与系统**

模拟和离散信号及系统的建模, 时域分析; 傅里叶级数, 连续和离散时间傅里叶变化及其应用, 采样, Z-变换, 状态变量。

先修要求: ECE<sub>A</sub>301。

学分: 3; 授课学时: 4; 实验学时: 0。

覆盖 BoK: CE-SGP 1-7。

**ECE<sub>A</sub>304: 数据通信网络**

数据通信和网络的概念和实践, 使学生能够理解数据通信中所使用的硬件和软件技术。

先修要求: ECE<sub>A</sub>202。

学分: 3 授课学时: 42 实验学时: 0。

覆盖 BoK : CE-NWK 1-11, CE-SET 3-5。

**ECE<sub>A</sub>305: 计算机体系结构**

“存储程序”数字计算机系统的详细设计和实现; CPU、I/O 子系统和存储器组织的设计; ALU 设计和计算机算术。

先修要求: ECE<sub>A</sub>202。

学分: 3; 授课学时: 42; 实验学时: 0。

覆盖 BoK: CE-CAO 1-11。

**ECE<sub>A</sub>401: CE 设计 I**

按小组讲课, 项目管理, 工程标准, 经济学和职业道德和行为准则; 学生选择教师作为导师, 实施项目设计和口头报告。

先修要求: ECEA302, CSCA302。

学分: 2; 授课学时: 14; 实验学时: 45。

覆盖 BoK: CE-SPE 7-10, CE-PPP 1-6, 11



**ECE<sub>A</sub>402: CE 设计 II**

按小组讲课,项目管理,工程标准,经济学和职业道德和行为准则;学生选择教师作为导师,实施项目设计和口头报告。

先修要求: ECEA401。

学分: 3;授课学时: 14;实验学时: 42。

覆盖 BoK: CE-SPE 10-12, CE-PPP 7-11。

**ECE<sub>A</sub>403: 嵌入式系统**

使用当代技术进行嵌入式系统设计的高级挑战;中断驱动,响应式,实时,面向对象,分布式客户端/服务器嵌入式系统。

先修要求: ECE<sub>A</sub>302, ECE<sub>A</sub>301。

学分: 3;授课学时: 28;实验学时: 42。

覆盖 BoK: CE-ESY 9-13, CE-SPE 10, CE-SRM 4, 6; 附加 CE-SWD 11。

**ECE<sub>A</sub>404: 信息安全**

加密和网络安全的基本和高级概念:对称和非对称加密,密钥管理,有线和无线网络网络安全协议,网络系统安全。

先修要求: ECE<sub>A</sub>304。

学分: 3;授课学时: 42;实验学时: 0。

覆盖 BoK: CE-SEC 1-11。

**ECE<sub>A</sub>405: 操作系统**

控制复杂计算机系统的操作系统的发展历史;进程管理,通信,调度技术,文件系统概念和操作;数据通信,分布式进程管理。

先修要求: ECE<sub>A</sub>202。

学分: 3;授课学时: 42;实验学时: 0。

覆盖 BoK: CE-SET 3-5, CE-SRM 1-6; 附加 CE-CAL 9, CE-SRM 7-8。

**MTH<sub>A</sub>204: 离散结构**

算法的概念,归纳,递归,证明,逻辑概念,集合理论,组合,计算机科学研究中的图理论基础。

先修要求: CSC<sub>A</sub>101 以及大学代数或相当基础。

学分: 3;授课学时: 42;实验学时: 0。

覆盖 BoK: CE-DSC 1-7。

## B.5 课程体系 B: 由计算机科学系开设

(一个由计算机科学系制定的计算机工程培养方案)

### B.5.1 培养目标和特色

在授予理学学士学位的计算机科学系,制定计算机工程专业培养方案时,往往会从计算机专业培养方案演化而来。有些学校设置了计算机科学系,或者有工程类培养方案,或者由物理系讲授工程导论课程。这类学校可能会对这个模式有兴趣。由于带有很多计算机科学培养方案的特征,这种模式会比本附录中的其他课程体系具有更大的通识教育模块,因此专门留给计算机工程的时间会更少一些。为此,CE2016 课程指导委员会专门设计了几门课程,以涵盖计算机工程知识体系的核心学习成效。需要指出的是,委员会特意设计的这些课程仅覆盖了传统电气工程的课程中计算机工程知识体系所要求的核心材料,而没有对核心学习成效之外的内容在深度和广度上添加更多的要求。与之形成对比的是,一些计算机科学课程则有部分超越了核心材料。不管如何,课程体系对电气工程、硬件和系统有了足够的覆盖,以使毕业生成为合格的计算机工程师。

### B.5.2 培养要求汇总

这项培养方案包括计算机科学 11 门必修课(包括毕业设计项目)和 3 门电气工程必修课。该培养方案可以灵活设置 3 门专业选修课和毕业设计。在前两个计算机科学引论课程电子电路和数字逻辑课程中设置有实验课经历。用于实验经历的总学时数小于本附录中提出的其他课程模型。此外,由于只有较少的课程包括工程设计,所以将毕业设计项目扩展到两个学期,以确保所有的毕业生都具有显著的设计经验、团队合作经验和利用现代工程工具的经验。CE2016 课程指导委员会认为,口头和书面的沟通能力会在通识教育和本课程的计算机工程部分体现。

本课程采用了一个相对传统的课程结构和内容,学分分配如下。

学分	领域
20	数学与统计学
11	自然科学(物理学、化学)
33	人文、社会科学、作文、文学
25	计算机必修课程(不包括设计项目)
11	必修的电气工程

续表

学分	领域
9	专业选修课(从计算机科学或工程培养方案中选)
5	毕业设计项目(从计算机科学培养方案中选)
6	任选课
120	计算机工程培养方案的总学分

### B.5.3 课程体系 B 的四年制模式

ELE:由工程或物理系开设

CSC:由计算机科学系开设

CE 专业选修:由工程或计算机科学系开设

课号	课名	学分	课号	课名	学分
<b>第 1 学期</b>			<b>第 2 学期</b>		
MTH 101	微积分 I	4	MTH 102	微积分 II	4
CHM 101	化学 I	4	PHY 101	物理 I	4
CSC <sub>B</sub> 101	计算机科学 I	3	CSC <sub>B</sub> 102	计算机科学 II	3
	英语写作	3		人文/社会科学	3
	总学分	14		总学分	14
<b>第 3 学期</b>			<b>第 4 学期</b>		
MTH 201	微分方程	3	MTH 203	线性代数	3
PHY 201	物理 II	3	CSC <sub>B</sub> 201	算法设计	3
MTH <sub>B</sub> 202	离散结构	3	ELE <sub>B</sub> 202	电路与电子	4
ELE <sub>B</sub> 201	数字设计引论	4	CSC <sub>B</sub> 202	计算机组成	3
	人文/社会科学	3		人文/社会科学	3
	总学分	16		总学分	16

续表

课号	课名	学分	课号	课名	学分
第 5 学期			第 6 学期		
MTH 301	概率和统计	3	CSC <sub>B</sub> 302	嵌入式系统	3
CSC <sub>B</sub> 301	计算机体系结构	3	CSC <sub>B</sub> 303	计算机网络和信息安全	3
ELE <sub>B</sub> 301	系统与信号处理	3	CSC <sub>B</sub> 304	操作系统	3
	科技写作	3		人文/社会科学	3
	人文/社会科学	3		人文/社会科学	3
	总学分	15		总学分	15
第 7 学期			第 8 学期		
CSC <sub>B</sub> 401	高级项目 I	2	CSC <sub>B</sub> 402	高级项目 II	3
CSC <sub>B</sub> 403	职业道德和职业素养	1		专业选修	3
	专业选修	3		专业选修	3
	人文/社会科学	3		人文/社会科学	3
	人文/社会科学	3		自由选修	3
	自由选修	3			
	总学分	15		总学分	15

#### B.5.4 计算机工程 BoK 与课程体系 B 之间的映射

对于下表的解释参见 B.1.2

BoK 领域 课程	C	C	C	D	E	N	P	S	S	S	S	S	A	D	L	P
	A	A	A	I	S	W	P	E	G	P	R	W	C	S	A	R
	E	L	O	G	Y	K	P	C	P	E	M	D	F	C	L	S
核心知识体系的最少学时	50	30	60	50	40	20	20	20	30	35	20	45	30	30	20	30

续表

课程	BoK 领域															
	C A E	C A L	C A O	D I G	E S Y	N W K	P P	S E C	S G P	S P E	S R M	S W D	A C F	D S C	L A L	P R S
CSC <sub>B</sub> 101												1-5				
CSC <sub>B</sub> 102												1, 6-10, 11-12				
CSC <sub>B</sub> 201		1-8, 10														
CSC <sub>B</sub> 202			1-4, 6-8													
CSC <sub>B</sub> 301			6-11	8, 10												
CSC <sub>B</sub> 302					1-12											
CSC <sub>B</sub> 303						1-8		1-11								
CSC <sub>B</sub> 304											1-6, 7-8					
CSC <sub>B</sub> 401							2-4, 6-9			1-12						
CSC <sub>B</sub> 402							2-4, 6-9			1-12						
CSC <sub>B</sub> 403							1, 5, 6-10									
ELE <sub>B</sub> 201			5	1-7, 9, 11												
ELE <sub>B</sub> 202	1-10															
ELE <sub>B</sub> 301									1-8							

续表

BoK 领域 课程	C A E	C A L	C A O	D I G	E S Y	N W K	P P P	S E C	S G P	S P E	S R M	S W D	A C F	D S C	L A L	P R S
MTH 101													1-3			
MTH 102													3-4			
MTH 201													5-7			
MTH <sub>B</sub> 202														1-9		
MTH203															1-10	
MTH301																1-9
涵盖的核心知 识体系单元	1-10	1-8	1-11	1-10	1-12	1-8	1-9	1-11	1-8	1-12	1-6	1-10	1-7	1-9	1-10	1-9
附加知识体系 单元		10		11							7-8	11-12				

### B.5.5 课程体系 B-课程概述

#### CSC<sub>B</sub>101: 计算机科学 I

计算引论;现代程序设计语言及其相关开发环境下的算法思维、问题求解。

先修要求:微积分入门或等同课程。

学分:3;授课学时:28;实验学时:42。

覆盖 BoK:CE-SWD 1-5。

#### CSC<sub>B</sub>102: 计算机科学 II

程序设计语言与系统的第二门课程;面向对象设计、数据结构、递归、数据建模、软件工程中的基本概念。

先修要求:CSC<sub>B</sub>101。

学分:3;授课学时:28;实验学时:42。

覆盖 BoK:CE-SWD 1, 6-10; 附加 CE-SWD 11-12。

**CSC<sub>B</sub>201: 算法设计**

算法分析与设计,算法设计策略,搜索与排序算法,并行算法,算法性能的权衡,算法复杂度。

先修要求:CSC<sub>B</sub>102。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-CAL 1-8; 附加 CE-CAL 10。

**CSC<sub>B</sub>202: 计算机组成**

计算机组成与体系结构导论课程;处理器组织、指令集体系结构、存储系统组织、性能、接口基础。

先修要求:CSC<sub>B</sub>101。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-CAO 1-4, 6-8。

**CSC<sub>B</sub>301: 计算机体系结构**

计算机总线结构、存储器组成与结构、中断结构、运算单元、输入输出结构、中央处理器组织、控制功能实现、流水线、性能测量、分布式系统模型。

先修要求:CSC<sub>B</sub>202。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-CAO 6-11; CE-DIG 8, 10。

**CSC<sub>B</sub>302: 嵌入式系统**

嵌入式系统的特点,嵌入式应用技术,并行输入和输出,同步和异步串行通信,中断处理,涉及数据采集,控制,传感器和执行器的应用程序,复杂的嵌入式系统的实现策略。

先修要求:CSC<sub>B</sub>202, CSC<sub>B</sub>301。

学分:3;授课学时:28;实验学时:42。

覆盖 BoK:CE-ESY 1-12。

**CSC<sub>B</sub>303: 计算机网络和信息安全**

局域网的设计和性能分析介绍;计算机网络的体系结构、协议、标准和技术;信息安全原理,认证,脆弱性的来源,恶意软件,攻防,网络安全。

先修要求:CSC<sub>B</sub>202, CSC<sub>B</sub>301, 并修要求:CSC<sub>B</sub>304。

学分:3;授课学时:28;实验学时:0。

覆盖 BoK:CE-NWK 1-8; CE-SEC 1-11。

**CSC<sub>B</sub>304: 操作系统**

基本操作系统及其组件;并发,调度和分派,内存及设备管理,文件系统及性能评估,实时操作系统,移动设备的操作系统。

先修要求:CSC<sub>B</sub>201, CSC<sub>B</sub>202。

学分:3;授课学时:28;实验学时:42。

覆盖 BoK:CE-SRM 1-6; 附加 CE-SRM 7-8。

**CSC<sub>B</sub>401: 高级项目 I**

面向独立设计的项目,以提供以下各个方面的经验:项目管理、风险管理、系统需求规约及体系结构、系统设计、集成,测试和部署;开放式问题的解决方案、在这些项目中的 新兴技术。

先修要求:CSE<sub>B</sub>302, CSE<sub>B</sub>312, CSE<sub>B</sub>332。

学分:2;授课学时:14;实验学时:42。

覆盖 BoK:CE-PPP 2-4, 6-9, CE-SPE 1-12。

**CSC<sub>B</sub>402 高级项目 II**

延续高级项目 I,关注项目设计的实现。

先修要求:CSC<sub>B</sub>401。

学分:3;授课学时:14;实验学时:84。

覆盖 BoK:CE-PPP 2-4, 6-9, CE-SPE 1-12。

**CSC<sub>B</sub>403: 职业道德和职业素养**

严格审查与计算机科学和与工程相关的道德问题,也要考虑法律和准法律(即政策和法规)问题,道德决策过程,隐私和保密性,计算机犯罪,职业准则和责任,软件盗版,计算机对社会的影响。

先修要求:达到大三学分标准。

学分:1;授课学时:0;实验学时:42。

覆盖 BoK:CE-PPP 1, 5, 6-10。

**ELE<sub>B</sub>201: 数字设计引论**

数制系统和信息表示;计算机算术;组合逻辑和时序逻辑电路的分析与综合;现代硬件描述语言的运用;计算系统的组成和结构。

先修要求:CSC<sub>B</sub>101。

学分:4;授课学时:42;实验学时:42。

覆盖 BoK:CE-DIG 1-7, 9, CE-CAO 5; 附加 CE-DIG 11。



**ELE<sub>B</sub>202: 电路与电子**

电路及网络分析的基本原理;瞬态分析、频率响应、Laplace 变换,傅里叶级数,电子材料和器件引论,二极管、双极型晶体管和逻辑元件,MOS 技术。

先修要求:MTH 201, PHY 102。

学分:4;授课学时:42;实验学时:42。

覆盖 BoK:CE-CAE 1-10。

**ELE<sub>B</sub>301: 系统与信号处理**

正弦和瞬态分析,卷积,变换分析,频率响应,信号的数字处理,差分方程,采样和混叠,离散时间变换,数字滤波器设计。

先修要求:ELE<sub>B</sub>202。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-SGP 1-8。

**MTH<sub>B</sub>202: 离散结构**

集合,函数,关系,布尔代数,一阶逻辑,证明技术,计数参数,迭代和递归,图和树。

先修要求:微积分入门或等同课程。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-DSC 1-9。

## B. 6 课程体系 C:由计算机科学系和电子与计算机工程系联合开设

(一个由计算机科学系和电子与计算机工程系联合制定的计算机工程培养方案)

### B. 6.1 培养目标和特色

计算机科学(CSC)系和电子与计算机工程(ECE)系共同开办该培养方案,完成该培养方案后即可获得计算机工程(CE)学士学位。本课程体系在计算机软件和硬件综合知识方面,为学生带来核心竞争力和独特价值,并提供计算机系统、硬件和软件三者之间以及理论和应用之间的平衡。计算机工程专业同时融合了计算机科学和电子工程领域内容。计算机工程专业的学生接受的是灵活的课程体系学习,作为公共核心的必修课程在计算机科学和电子工程两方面取得均衡。通过从两科系中选择专业选修课实现专业化,使得培养方案既满足学生兴趣,又能在不断变化的就

业市场满足人才需求。计算机工程课程体系为学生提供了全面的教育,这是他们理解全球和社会背景下工程解决方案的影响必不可少的。毕业生完成培养方案后,可继续在计算机工程领域深造,也可以选择众多与计算机及相关高科技环境下的行业就业。

### B.6.2 培养要求汇总

本课程的通识教育组成包括数学(21 学分)、自然科学(16 学分)以及人文社会科学课程(18 学分)。计算机工程课程的必修核心课程组成如下:传统计算机科学(CSC)课程(如程序设计基础),传统电子与计算机工程(ECE)课程(如电路与电子)以及可由两个系的计算机工程专业教师授课的计算机工程(CE)课程(如计算机组织与结构)。通过从两个系中选择涉及各个知识领域的专业选修课程实现专业化。最后,毕业设计课程开设于第四学年,横跨两个学期(计算机工程设计 I 和计算机工程设计 II)。学分分配如下。

学分	领域
21	数学
16	自然科学(物理、化学)
18	人文科学、社会科学、写作、文学
12	必修计算机工程(CE)
19	必修电子与计算机工程(ECE)
16	必修计算机科学(CSC)
12	专业选修
6	设计项目
0	自由选修
120	计算机工程培养方案的总学分

### B.6.3 课程体系 C 的四年制模式

CE:由计算机工程系开设。

ECE:由电子与计算机工程系开设;

CSC:由计算机科学系开设

CE 专业选修课:各系均可开设选修课

课程	描述	学分	课程	描述	学分
<b>第 1 学期</b>			<b>第 2 学期</b>		
MTH 101	微积分 I	4	MTH 102	微积分 II	4
CHM 101	化学 I +实验	4	PHY 101	物理 I +实验	4
	人文/社会科学	3	CSC <sub>c</sub> 101	程序设计基础 I	3
	人文/社会科学	3	CHM 102	化学 II 或生物科学	4
	总学分	14		总学分	15
<b>第 3 学期</b>			<b>第 4 学期</b>		
MTH 201	微积分 III	4	MTH 202	微分方程	3
PHY 201	物理 II +实验	4	ECE <sub>c</sub> 202	电路与电子	4
CSC <sub>c</sub> 201	程序设计基础 II	3	CE <sub>c</sub> 201	计算机组织与结构概论	3
ECE <sub>c</sub> 201	数字与逻辑设计+实验	4	CSC <sub>c</sub> 202	离散结构概论	3
				人文/社会科学	3
	总学分	15		总学分	16
<b>第 5 学期</b>			<b>第 6 学期</b>		
MTH 301	计算机线性代数	3	CE <sub>c</sub> 301	计算机系统工程概论	3
CSC <sub>c</sub> 301	数据结构与算法	4	MTH 302	工程统计学	3
ECE <sub>c</sub> 301	嵌入式与微处理器系统 +实验	4	CE <sub>c</sub> 302	计算机网络	3
ECE <sub>c</sub> 302	数字系统设计+实验	4		专业选修	3
				人文/社会科学	3
	总学分	15		总学分	15
<b>第 7 学期</b>			<b>第 8 学期</b>		
CSC <sub>c</sub> 401	操作系统	3	CE <sub>c</sub> 401	计算机网络安全	3
ECE <sub>c</sub> 401	信号与系统	3		专业选修	3
CE <sub>c</sub> 402	计算机工程设计 I	3		专业选修	3
	专业选修	3	CE <sub>c</sub> 403	计算机工程设计 II	3
	人文/社会科学	3		工程师专业沟通与伦理	3
	总学分	15		总学分	15



续表

BoK 领域 课程	C A E	C A L	C A O	D I G	E S Y	N W K	P P P	S E C	S G P	S P E	S R M	S W D	A C F	D S C	L A L	P R S
MTH 202													1-7			
MTH 301															1-10	
MTH 302																1-9
涵盖的核心知 识体系单元	1-10	1-8	1-11	1-10	1-8	1-8	1-10	1-11	1-8	1-12	1-6	1-10	1-7	1-9	1-10	1-9
附加知识体系 单元											7	12				

### B. 6.5 课程体系 C-课程概述

#### CSC<sub>c</sub>101: 程序设计基础 I

初级问题求解和计算机程序设计概念,包括面向对象的程序设计、程序和数据抽象和程序模块化。

先修要求:无。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-SWD 1-5。

#### CSC<sub>c</sub>201: 程序设计基础 II

中级问题求解和计算机程序设计概念,包括算法策略、递归及数据结构和应用程序编程接口(API)的有效设计和运用。

先修要求:CSC<sub>c</sub>101。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-CAL 4、CE-SWD 3-7;附加 CE-SWD 12。

#### CSC<sub>c</sub>202: 离散结构概论

离散结构概念,包括布尔代数、一阶代数、证明技术、集合论和图论。

先修要求:CSC<sub>c</sub>201、MTH 101。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-DSC 1-9。

### **CSC<sub>c</sub>301: 数据结构与算法**

基础数据结构设计(如:栈、队列、堆、链接结构、树、图)及其操作;常见任务经典算法的设计和分析(如:排序、查找、图算法)。

先修要求:CSC<sub>c</sub>201。

学分:4;授课学时:56;实验学时:0。

覆盖 BoK:CE-CAL 1-8、CE-SWD 4-5。

### **CSC<sub>c</sub>401: 操作系统**

操作系统基础及其组件:调度、资源管理、进程管理、中断处理、并发处理和系统性能评测。

先修要求:CSC<sub>c</sub>301, CE<sub>c</sub>201。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-SRM 1-6;附加 CE-SRM 7。

### **ECE<sub>c</sub>201: 数字与逻辑设计+实验**

数字逻辑设计要素(如布尔代数、基础逻辑电路),组合和时序逻辑电路的分析和综合,有限状态机、硬件描述语言(HDL)以及可编程逻辑器件(如FPGA)的介绍。

先修要求:CSC<sub>c</sub>101。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-DIG 1-9。

### **ECE<sub>c</sub>202: 电路与电子**

电路分析基础:基本概念(如:电压、电流、欧姆定律);瞬态分析和网络的频率响应;运算放大器、电子材料与器件、双极晶体管与逻辑系列以及金属氧化物半导体(MOS)技术。

先修要求:MTH 201、PHY 201。

学分:4;授课学时:56;实验学时:0。

覆盖 BoK:CE-CAE 1-10。

### **ECE<sub>c</sub>301: 嵌入式与微处理器系统+实验**

基于微处理器的嵌入式系统、同步和异步串行通信、接口、中断处理、数据采集、实时处理。

先修要求:CE<sub>c</sub>201、ECE<sub>c</sub>201。

学分:4;授课学时:42;实验学时:14。

覆盖 BoK:CE-CAO 6-8、CE-ESY 1-8。

### **ECE<sub>c</sub>302: 数字系统设计+实验**

高级数字设计:组合和时序逻辑构件和控制-数据路径的模块化设计;硬件描述语言(HDL)和可编程逻辑器件(如 FPGA)的广泛运用;系统架构设计和评估。

先修要求:ECE<sub>c</sub>201, CE<sub>c</sub> 201。

学分:4;授课学时:42;实验学时:14。

覆盖 BoK:CE-DIG 2, CE-DIG 6-10、CE-SPE 8-9。

### **ECE<sub>c</sub>401: 信号与系统**

连续时间和离散时间信号分析,包括傅立叶级数和傅立叶变换;采样;有限和无限脉冲响应(FIR 和 IIR)滤波器设计;频率响应和系统功能。

先修要求:MTH 201、MTH 202、MTH 301。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-SGP 1-8。

### **CE<sub>c</sub>201: 计算机组织与结构概论**

计算机组织与系统结构介绍,包括指令集体系结构、处理器组成、总线结构、存储器子系统,输入/输出接口和通信,流水线 and 性能测量。

先修要求:CSC<sub>c</sub>201, ECE<sub>c</sub>201。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-CAO 1-9、CE-CAL 6。

### **CE<sub>c</sub>301: 计算机系统工程概论**

计算机系统工程基础,包括项目管理、系统结构设计、并发硬件和软件设计;系统集成、测试和验证;以及如可维护性、可持续性、可制造性等概念。

先修要求:CE<sub>c</sub>201、CSC<sub>c</sub>301。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-SPE 1-12,CE-PPP 2,CE-PPP 6,CE-SRM 3,CE-SWD 8-10。

### **CE<sub>c</sub>302: 计算机网络**

计算机网络体系结构、协议、标准和技术,包括局域和广域网、无线和移动网络、网络应用程序、网络管理和安全。

先修要求:CE<sub>c</sub>201、ECE<sub>c</sub>301。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-CAL 6、CE-NWK 1-8、CE-SEC 8。

#### **CE<sub>c</sub>401: 计算机网络安全**

计算机网络安全概念,包括数据安全性和完整性、漏洞和利用、社会工程、加密、身份认证、网络和 Web 安全、可信计算。

先修要求:CE<sub>c</sub>201、CE<sub>c</sub>302。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-NWK 8、CE-SEC 1-11。

#### **CE<sub>c</sub>402: 计算机工程设计 I**

横跨两学期的计算机工程毕业设计的第一学期课程,该课程强调团队协作、项目管理和并行硬件和软件设计;系统集成、测试和验证。

先修要求:CE<sub>c</sub>301、ECE<sub>c</sub>301。

学分:3;授课学时:14;实验学时:84。

覆盖 BoK:CE-PPP 1-4。

#### **CE<sub>c</sub>403: 计算机工程设计 II**

横跨两学期的计算机工程毕业设计的第二学期课程,该课程强调团队协作、项目管理和并行硬件和软件设计;系统集成、测试和验证。

先修课程:CE<sub>c</sub>402。

学分:3;授课学时:14;实验学时:84。

覆盖 BoK:CE-PPP 2-10。

## **B.7 课程体系 D: 由中国开设**

(由中国的计算机科学与技术系制定的计算机工程专业培养方案)

### **B.7.1 培养目标和特色**

本培养方案的目标是获取计算机科学与技术(CST, Computer Science and Technology)专业的学士学位。在中国,计算机科学与技术系通常是计算机工程和计算机科学的结合,本培养方案是为计算机工程方向的学生设计的。课程体系中包含了计算机理论、计算机体系结构、计算机网络和计算机程序设计等方面的课程,覆盖了CST学位的核心要求,同时强调以系统、全面的观点来理解硬件和软件。该计划培养学生计算机工程方向宽广的知识背景和扎实的实践能力,为今后研究生阶段的



学习或者在企事业单位就业打下坚实的基础。

### B.7.2 培养要求汇总

本培养方案的课程学习部分包括 22 学分必修的计算机科学与技术课程。为提高培养方案的灵活性,允许从电子工程系或计算机科学与技术系中选修 10 学时的课程。四门课程中包含课内的实验学时,可以为学生提供必要的运用现代工具和设计技术的实践能力。而最后一个学期的毕业设计(论文训练)则为所有毕业生应具备的设计能力提供了保证。在这些课程实验和毕业设计中,还训练了学生的口头和书面沟通能力。本课程培养方案的学分分配如下表所示。

学分	领域
25	数学
10	自然科学(物理、化学)
2	选修数学和自然科学
41	人文、社科、写作、文学、外语和体育
48	必修计算机科学与工程(CE)
12	实验或计算机科学与工程(CE)
10	专业选修——基础和高级
15	毕业设计(CE)
0	自由选修
163	计算机工程专业课程的总学分

### B.7.3 课程体系 D 的四年制模式

课号	课名	学分	课号	课名	学分	课号	课名	学分
第 1 学期			第 2 学期			第 3 学期		
	体育 I	1		体育 II	1		英语实践	2
	人文和社科 I	3		人文和社科 II	3	CST <sub>p</sub> 301	程序设计实践(实验课)	2

续表

课号	课名	学分	课号	课名	学分	课号	课名	学分
	英语 I	2		英语 II	2			
MTH101	微积分 I	5	MTH201	微积分 II	5			
MTH102	线性代数 I	4	MTH202	线性代数 II	2			
MTH103	离散数学 I	3	CST <sub>D</sub> 201	面向对象程序设计基础	2			
CST <sub>D</sub> 101	程序设计基础	3	MTH203	离散数学 II	3			
CST <sub>D</sub> 102	信息科学导论	1	PHY201	大学物理 I	4			
	人文选修	1						
学分小计		23	学分小计		22	学分小计		4
第 4 学期			第 5 学期			第 6 学期		
PHY401	大学物理 II	4	CST <sub>D</sub> 501	数字电路及实验	4	CST <sub>D</sub> 601	汇编语言程序设计与训练(实验课)	3
PHY402	大学物理实验 I	1	PHY501	大学物理实验 II	1	CST <sub>D</sub> 602	Java 语言程序设计与训练(实验课)	2
CST <sub>D</sub> 401	数据结构	4	MTH501	概率论与数理统计	3			
	人文与社科 III	5	CST <sub>D</sub> 502	形式语言与自动机	2			
	大学英语 III	2	CST <sub>D</sub> 503	人工智能的导论	2			
	体育 III	1		体育 IV	1			
MTH401	复变函数引论(数学/科学选修)	2		大学英语 IV	2			

续表

课号	课名	学分	课号	课名	学分	课号	课名	学分
CST <sub>D</sub> 402	电路原理及实验	4		人文与社科选修	1			
			CST <sub>D</sub> 504	高性能计算导论	2			
学分小计		23	学分小计		18	学分小计		5
第 7 学期			第 8 学期			第 9 学期		
CST <sub>D</sub> 701	计算机组成原理	4	CST <sub>D</sub> 801	操作系统	3	CST <sub>D</sub> 901	计算机系统综合设计与实现(实验课)	5
CST <sub>D</sub> 702	计算机网络	3	CST <sub>D</sub> 802	计算机系统结构	3			
CST <sub>D</sub> 703	编译系统原理	2	CST <sub>D</sub> 803	数值分析	3			
CST <sub>D</sub> 704	软件工程	3		政治理论	4			
CST <sub>D</sub> 705	信号处理原理	3		人文与社科 IV	3			
	人文与社科选修	2	CST <sub>D</sub> 804	前沿技术导论(专业选修)	2			
CST <sub>D</sub> 706	数据库系统概论	2						
学分小计		19	学分小计		18	学分小计		5
第 10 学期			第 11 学期					
	人文与社科 V	5	CST <sub>D</sub> 1101	毕业设计	15			
CST <sub>D</sub> 1001	网络安全与网络管理(专业选修)	3						



续表

课程	BoK 领域															
	C A E	C A L	C A O	D I G	E S Y	N W K	P P P	S E C	S G P	S P E	S R M	S W D	A C F	D S C	L A L	P R S
CST <sub>D</sub> 504			1-2, 10-11													
CST <sub>D</sub> 601			3,8,9													
CST <sub>D</sub> 602												4,8				
CST <sub>D</sub> 701			1-9	8												
CST <sub>D</sub> 702						1-10	6,7	8,9								
CST <sub>D</sub> 703		10														
CST <sub>D</sub> 704							3-5, 10			1-5, 7-9						
CST <sub>D</sub> 705									1-8							
CST <sub>D</sub> 706												1-2, 9-10				
CST <sub>D</sub> 801												1-7				
CST <sub>D</sub> 802			1-2, 10-11													
CST <sub>D</sub> 803#																
CST <sub>D</sub> 804																
CST <sub>D</sub> 901												1-12				
CST <sub>D</sub> 1001						8		1-11								
CST <sub>D</sub> 1002					1-13											
CST <sub>D</sub> 1101								1-11								
MTH101													1-4			

续表

BoK 领域 课程	C A E	C A L	C A O	D I G	E S Y	N W K	P P P	S E C	S G P	S P E	S R M	S W D	A C F	D S C	L A L	P R S
MTH102															1-5	
MTH103														1-6		
MTH201													1-2, 5-7			
MTH202															1-2, 6-10	
MTH203														1-2, 7-9		
MTH401 <sup>#</sup>																
MTH501																1-9
涵盖的核心知识体系单元	1-10	1-8	1-11	1-10	1-12	1-8	1-10	1-11	1-8	1-12	1-6	1-10	1-7	1-9	1-10	1-9
附加知识体系单元	12	10			13	9-10	11				7					

<sup>#</sup> 本课程没有覆盖计算机工程知识体系中的任何知识单元。

### B.7.5 课程体系 D-课程概述

#### CST<sub>D</sub>101 程序设计基础

计算机程序设计的基本概念介绍,包括问题求解、算法思维、简单的数据结构和一些基本的算法。

先修要求:无。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:CE-SWD 1-6。

**CST<sub>d</sub>102 信息科学导论**

信息科学的引导类课程,介绍信息科学中的基础问题和解决方法,包括一些前沿技术的讲座。

先修要求:无。

学分:1;授课学时:16;实验学时:0。

覆盖 BoK:CE-SET 1,3-4。

**CST<sub>d</sub>201 面向对象程序设计基础**

本课程包括 C++程序设计、面向对象程序设计的基础知识,并介绍运用面向对象技术进行问题求解和程序设计的方法。

先修要求:CST<sub>d</sub>201。

学分:2;授课学时:32;实验学时:0。

覆盖 BoK:CE-SWD 1-2,6-8。

**CST<sub>d</sub>301 程序设计实践**

本课程是一门面向实践的课程,目的培养学生的创新理念和实际动手的技能。通过课程教学和实验,使学生掌握 C 语言的基本知识和基本的程序设计算法。

先修要求:CST<sub>d</sub>101。

学分:2;授课学时:16;实验学时:32。

覆盖 BoK:CE-SWD 3-8。

**CST<sub>d</sub>401 数据结构**

本课程包括数据抽象和表示,数据结构(如:向量、数组、列表、栈、队列、树、优先级队列、哈希表和图等)设计和实现,经典算法的设计(如:查找、排序和选择、遍历、模式匹配、拓扑查找、哈希)以及算法评价和分析。

先修要求:CST<sub>d</sub>101。

学分:4;授课学时:64;实验学时:32。

覆盖 BoK:CE-CAL 1-8, 10。

**CST<sub>d</sub>402 电路原理及实验**

电路的基础知识,包括电压和电流、功率等的基本概念,电路基本元件(如:电阻、电感、电容、MOS 管、运算放大器),基本定律(如:欧姆定律、基尔霍夫定律、电阻等效变化)以及电路分析的基本方法。

先修要求:无。

学分:4;授课学时:48;实验学时:16。

覆盖 BoK:CE-CAE 1-10, 12。

### **CST<sub>D</sub>501 数字逻辑电路及实验**

本课程重点讲解数字逻辑电路的分析、设计和实现方法,包括布尔逻辑和逻辑化简、组合逻辑电路与时序逻辑电路的设计和分析、逻辑设计中竞争和冒险的管理等。

先修要求: CST<sub>D</sub>402。

学分:4;授课学时:48;实验学时:16。

覆盖 BoK:CE-DIG 1-10。

### **CST<sub>D</sub>502 形式语言与自动机**

课程内容以正规语言、上下文无关语言及其计算模型为主,图灵机和计算理论初步为辅。

先修要求: MTH103, MTH203。

学分:2;授课学时:32;实验学时:0。

覆盖 BoK:无。

### **CST<sub>D</sub>503 人工智能导论**

人工智能的先导性课程,主要内容有人工智能的基本问题、产生式系统、搜索问题、与或图搜索、谓词演算及其应用、高级搜索和专家系统等。

先修要求: CST<sub>D</sub>101, MTH103, MTH203。

学分:2;授课学时:32;实验学时:0。

覆盖 BoK:无。

### **CST<sub>D</sub>504 高性能计算导论**

了解高性能计算机的基本组成和一般概念,并行程序设计的方法,程序的性能模型,性能评价和优化技术,使用 MPI 和 OpenMP 编写并行算法和程序。

先修要求: CST<sub>D</sub>101。

学分:2;授课学时:32;实验学时:0。

覆盖 BoK:CE-CAO 1-2, 10-11。

### **CST<sub>D</sub>601 汇编语言程序设计与训练**

关于汇编语言基本理论、编程工具、程序设计方法和应用的实践性课程。

先修要求: CST<sub>D</sub>101。

学分:3;授课学时:32;实验学时:32。



覆盖 BoK:CE-CAO 3, 8-9。

### **CST<sub>D</sub>602 Java 语言程序设计与训练**

关于 Java 语言程序设计,Java 的历史和演化、开发工具、面向对象程序设计方法和网络编程技术的实践性课程。

先修要求: CST<sub>D</sub>101。

学分:2;授课学时:32;实验学时:32。

覆盖 BoK:CE-SWD 4, 8。

### **CST<sub>D</sub>701 计算机组成原理**

单 CPU 计算机组成和体系结构,指令系统,数据通路和控制器设计,指令流水和性能评价,层次存储器组织,总线结构,输入/输出接口,外部设备。

先修要求: CST<sub>D</sub>101, CST<sub>D</sub>501。

学分:4;授课学时:48;实验学时:16。

覆盖 BoK:CE-CAO 1-9, DIG 8。

### **CST<sub>D</sub>702 计算机网络原理**

计算机网络体系结构,局域网和广域网的典型协议和标准,无线和移动网络,传统网络应用,网络管理和安全,以及网络的社会影响,网络工程师的职业素养和责任。

先修要求: CST<sub>D</sub>101, CST<sub>D</sub>401。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:CE-NWK 1-10, CE-PPP 6-7, CE-SEC 8-9。

### **CST<sub>D</sub>703 编译原理**

编译系统概述;形式语言、文法和自动机的基础知识;词法分析;语法分析;语法制导的语义处理基础;语义分析和中间代码生成;符号表组织;运行时存储组织;代码优化和目标代码生成。

先修要求: CST<sub>D</sub>401, CST<sub>D</sub>502, CST<sub>D</sub>602。

学分:2;授课学时:32;实验学时:0。

覆盖 BoK:无。

### **CST<sub>D</sub>704 软件工程**

主要内容包括:软件开发方法,软件生命周期,分析和设计方法,软件测试和性能评价,软件开发管理。

先修要求: CST<sub>D</sub>101, CST<sub>D</sub>401。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:CE-SPE 1-5, 7-9 CE-PPP 3-5, 10。

### **CST<sub>D</sub>705 信号处理原理**

主要内容包括:信号和信号处理的基本概念、连续时间傅里叶变换、离散时间傅里叶变换、Z 变换与离散系统分析、滤波器的设计与实现。

先修要求: MTH101, MTH201。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:CE-SGP 1-8。

### **CST<sub>D</sub>706 数据库系统概论**

实体-联系模型和关系模型,数据库系统的概念设计、逻辑设计和物理设计,数据库管理系统的核心技术(如:查询处理、事务管理、并发控制与故障恢复),数据库管理系统体系结构的扩展,数据库技术的研究现状和新的应用领域。

先修要求: CST<sub>D</sub>101, CST<sub>D</sub>401。

学分:2;授课学时:32;实验学时:0。

覆盖 BoK:CE-SWD 1-2, 9-10。

### **CST<sub>D</sub>801 操作系统**

计算机操作系统概论,进程管理,进程调度,存储管理,文件管理以及设备管理。

先修要求: CST<sub>D</sub>101, CST<sub>D</sub>401, CST<sub>D</sub>701。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:CE-SRM 1-7。

### **CST<sub>D</sub>802 计算机系统结构**

本课程主要介绍计算机系统的结构、设计原理等关键技术。包括:计算机系统的基本设计原理和性能评价;时间与空间并行技术;多核处理器与多处理器系统;高性能计算与互连网络;单核与多核系统层次存储结构。

先修要求: CST<sub>D</sub>501, CST<sub>D</sub>701, CST<sub>D</sub>703。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:CE-CAO 1-2, 10-11, CE-SET 4。

### **CST<sub>D</sub>803 数值分析**

数值计算与误差分析;线性方程组的解法:直接法与迭代法,以及矩阵特征值与

特征向量的计算;函数逼近与插值求值法;拉格朗日插值、牛顿插值、分段低次插值和三次样条插值;数值积分与数值微分的近似计算;方程求根的近似解法

先修要求: MTH101, MTH102, MTH201, MTH202。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:无。

### **CST<sub>D</sub>804 软件前沿技术**

本课程主要介绍计算机软件技术的前沿研究,包括:系统软件,数据和知识工程,软件工程;计算机辅助设计技术。

先修要求:无。

学分:2;授课学时:32;实验学时:0。

覆盖 BoK:CE-SET 1-5。

### **CST<sub>D</sub>901 计算机系统综合设计与实现**

设计和实现完整计算机系统的实践类课程。要求学生 2 或 3 人一组,实现一台具备基本硬件系统、操作系统和编译系统的完整的计算机系统。

先修要求: CST<sub>D</sub>701, CST<sub>D</sub>702, CST<sub>D</sub>703, CST<sub>D</sub>801。

学分:5;授课学时:20;实验学时:140<sup>1</sup>

覆盖 BoK:CE-SPE 1-12。

### **CST<sub>D</sub>1001 信息安全与网络管理**

密码学基础、对称密码和公钥密码原理、哈希算法、数字签名、用户认证技术、网络安全技术、网络管理和运营的体系结构和协议标准。

先修要求: CST<sub>D</sub>702。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:CE-SEC 1-11, CE-NWK 8。

### **CST<sub>D</sub>1002 嵌入式系统**

嵌入式系统的组成部件(如:嵌入式处理器、Flash 等嵌入式存储器、嵌入式系统的总线和通信方式、传感器和驱动器),电源设计,启动过程,实时操作系统,驱动程序和中间件软件等嵌入式软件的设计,嵌入式系统开发和测试方法。

先修要求: CST<sub>D</sub>701, CST<sub>D</sub>801。

学分:3;授课学时:48;实验学时:0。

<sup>1</sup>暑期课。学生要在 5 周内设计和实现一个完整的计算机;在这个暑期课程时段,学生没有其他任务。

覆盖 BoK:CE-ESY 1-13。

### **CST<sub>d</sub>1002 毕业设计**

毕业设计检验和确保学生有足够且良好的设计经历,尤其是团队工作经历和使用现代工程设计工具的能力。

先修要求:无。

学分:15;授课学时:32;实验学时:400<sup>1</sup>。

覆盖 BoK:CE-PPP 1-11。

### **MTH101 微积分 I**

本课程介绍极限的基本理论,包括函数极限、函数的连续性以及极限的计算。课程还介绍了导数和微分、积分理论、常微分方程,以及广义积分、数项级数、收敛域和一致性收敛、逐项极限、逐项求和、逐项积分。

先修要求:无。

学分:5;授课学时:80;;实验学时:0。

覆盖 BoK:CE-ACF 1-4。

### **MTH102 几何与代数 I**

本课程是高等教育中的基础数学课程之一,主要介绍几何、代数以及它们之间的关系。教学内容包括:映射,几何的序,群、环和域的基本概念,几何空间中的向量,线性空间和内积空间;代数相关内容包括线性映射、矩阵、行列式的定义和性质、线性方程组、正交矩阵和相似矩阵。

先修要求:无。

学分:4;授课学时:64;实验学时:0。

覆盖 BoK:CE-LAL 1-5。

### **MTH103 离散数学 I**

命题逻辑和命题演算,一阶谓词逻辑与谓词演算,命题逻辑与谓词逻辑的自然推理形式系统,以及集合和二元关系的基本运算和性质,任意集合中的函数及其性质。

先修要求:无。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:CE-DSC 1-6。

<sup>1</sup>在中国,本科生必须在最后一个学期完成一项毕业设计,至少持续16周,包含400个实验学时。

### **MTH201 微积分 II**

多元函数及其微分学,第一类曲线积分与第一类曲面积分,第二类曲线曲面积分,平面向量场和 Green 公式,空间向量场和 Gauss 公式,Stokes 公式,积分与路径无关;二阶线性常微分方程,一阶线性常微分方程组。

先修要求: MTH101。

学分:5;授课学时:80;实验学时:0。

覆盖 BoK:CE-ACF 1-2, 6-10。

### **MTH202 几何与代数 II**

二次型基本概念,二次曲线及其分类,常见曲面,空间曲线的方程,二次曲面及其分类,平面正交变换与平面仿射映射,射影平面与齐次坐标,射影变换和射影映射,响亮函数及其微积分,曲线的弧长和 Frenet 标架。。

先修要求: MTH102。

学分:2;授课学时:32;实验学时:0。

覆盖 BoK:CE-LAL 1-2, 6-10。

### **MTH203 离散数学 II**

本课程讲授图论和代数结构的基本知识,包括图的基本概念及图的代数表示,道路、回路及图的连通性;树的基本概念,树的表述与性质;代数结构基础概念,群论环和域,格与布尔代数等基本理论。

先修要求: MTH103。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:CE-DSC 1-2, 7-9。

### **MTH501 概率论与数理统计**

本课程讲授随机现象的数学描述,以及概率论与数理统计中的核心概念。包括:概率空间,确定随机事件概率的方法,条件概率及其在概率计算中的应用,随机事件的统计独立性,随机变量及其分布,随机现象的定量统计分析,利用随机模型进行统计分析和统计推断的方法。

先修要求:无。

学分:3;授课学时:48;实验学时:0。

覆盖 BoK:CE-PRS 1-9。

## B. 8 课程体系 E: 博洛尼亚-3 模型

(与博洛尼亚宣言第一阶段培养方案兼容的计算机工程培养方案)

### B. 8.1 培养目标 and 特色

本课程体系模型展示了一个典型的计算机工程培养方案。该方案可以在按博洛尼亚宣言运作的欧洲高等教育区内的第一阶段培养方案中找到。

博洛尼亚宣言是 1999 年 6 月 19 日在意大利博洛尼亚,由来自 29 个国家的主管高等教育的部长签署的。其目的是要建立一个欧洲高等教育区(EHEA),能够便于学生在各国之间流动和提高教育的质量。实质上,他们的学位结构可分为两个阶段,第一阶段的学士学位为三年,第二阶段的硕士学位再有二年,或者是五年制的本硕连读。要实现上述提议,要求规范学位内的学分数量和确保跨机构的学生学习负担的可比性。截至 2015 年 5 月,47 个国家的 4000 多所高等院校都采纳了博洛尼亚宣言。

博洛尼亚可比性的基础是欧洲学分转换系统(ECTS)和 ECTS 学分。简而言之,一个学年算为 60 个 ECTS 学分,相当于在所有国家的 1500-1800 学时的学习。通常,一个学士学位要求 180-240 个 ECTS 学分,需要 3-4 年,硕士学位要求 90-120 个 ECTS 学分需要 1-2 年。相关资格确认体系由学习成效决定;这已经成为博洛尼亚教学过程的本质。

博洛尼亚框架尝试将不同的教育系统连接起来,而不是协调它们。博洛尼亚约定中相当大的灵活性在于它促进广泛的兼容性。例如英国,采纳了博洛尼亚约定,但其荣誉学位(不在博洛尼亚约定中)的思路是自主的,经常使用不同的(但兼容的)学分系统和资格确认体系,硕士培养方案通常需要(完整的)12 个月。

博洛尼亚约定适应各种各样的培养方案,许多包含诸如文化问题和语言学等元素,常常有助于在国家之间的流动。本课程体系的实例提供了计算机工程学位第一阶段的指南,第一阶段基本上提供计算机工程的基础。第二阶段通常安排更高级的内容,往往专注于专业化,预期获得硕士学位。

希望深入学习计算机工程的学生将发现这个培养方案是具有挑战性的,并可获得有益的经历。该课程体系为计算机和数字系统的科学和工程提供了一个广阔的基础,将理论、分析和设计作为重点。该课程体系还将促进学生分析能力、计算机水平和应用技术,使学生能够分析、设计和测试数字和计算机系统、架构、网络和流程。该培养方案的毕业生将能够应用和评价各种领域的计算机工程,如应用电子、数字设备和系统、电磁场和波、计算机体系结构、系统和网络。他们将具备进入到相关的第二阶段或获得硕士学位的能力。毕业生必须具备设计技能,他们将有能力将他们积累知识应用到计算机中。这个计算机工程课程体系所提供的完全的准备,包括了

解工程解决方案在全球和社会背景中产生的影响所需的广泛的教育。

理论、实践、应用和态度的组合,伴随着每一个模块或课程的建设。其目的是传达一种计算机工程特定的精神气质。尤其是在早期的课程中,这是一个重要的考虑因素。任何模范课程体系都应该包含总体目标(或目的)和具体学习目的,也应该体现毕业生的预期特征。

### B.8.2 培养要求汇总

这个三年期的培养方案要求学生预先学习了两个学期的微积分和两个学期的物理学。本培养方案中的很多课程都以此作为先修课。

对计算机工程概念的介绍设置在培养过程的早期。这么做的理由是,学生们确实应该对他们所要学的学科进行尝试,这对提高他们学习本专业的动机是非常重要的。第三年,课程体系包括一些选修课。

在本培养方案的学习中,实验室实践的元素将构成计算机工程中每门课程的组成部分,这种整合的目的是加强和例证相关的课堂教学工作。在一些课程中,实验室工作的数量通常要比其他部分的多。相应地,我们采用了以下的规矩。

- 对于预期有密集的实验活动的课程,一门 3 学分课通常是包括 28 学时的课堂教学, 28 学时的实验室工作加上相关的复习时间。
- 对于预期需要较少实验活动的课程,则是 14 学时的实验室工作, 42 学时的课堂教学和相关复习学时。

一个三年期的计算机工程培养方案包含 30 门课程和 90 学分的学习,学分分布如下。

学分	领域
15*	数学
0*	自然科学(物理、化学)
0	人文,社会科学,写作,文学
21	必修的计算机工程(CE)
21	必修的电子与计算机工程(ECE)
9	必修的软件工程
12	必修的计算机科学(CSC)
6	专业选修
6	设计项目(CE)
0	自由选修
90*	博洛尼亚-3 计算机工程培养方案的总学分

\*注:不包含先修课程的学分,如微积分、物理学及其他大学水平的课程

在这个示例中,在第三年的学习中允许有选修课(可选)。学生可以从 4 门计算机工程课程集合中选取 2 门作为第三年的选修课。第三年的 4 门选修课包括:计算机图形学、智能系统和机器人、设备开发和多媒体系统。

下面是一个三年期培养方案可能的课程体系。其后是这个培养方案的课程和计算机工程知识体系的映射表。在课程概览中出现的是这个计算机工程课程体系例子中的课程(或模块)说明。

### B. 8.3 课程体系 E 的三年制模式

课程	描述	学分	课程	描述	学分
<b>第 1 学期</b>			<b>第 2 学期</b>		
MTH <sub>E</sub> 101	计算的离散结构	3	MTH <sub>E</sub> 103	微积分与几何	3
MTH <sub>E</sub> 102	应用概率统计	3	MTH <sub>E</sub> 104	线性代数	3
CSC <sub>E</sub> 101	计算机和信息系统	3	CE <sub>E</sub> 101	计算机工程的概念	3
ELE <sub>E</sub> 101	电子技术基础	3	ELE <sub>E</sub> 102	数字电路 I	3
SWE <sub>E</sub> 101	编程基础	3	SWE <sub>E</sub> 102	程序设计基础	3
	总学分	15		总学分	15

<b>第 3 学期</b>			<b>第 4 学期</b>		
MTH <sub>E</sub> 201	工程师数学	3	CSC <sub>E</sub> 202	操作系统和以网络为中心的计算机	3
CE <sub>E</sub> 201	网络与通信	3	CE <sub>E</sub> 203	计算机工程职业问题	3
CSC <sub>E</sub> 201	算法分析与设计	3	ELE <sub>E</sub> 202	模拟电路	3
CE <sub>E</sub> 202	计算机组成	3	CE <sub>E</sub> 204	计算机系统工程	3
ELE <sub>E</sub> 201	数字电路 II	3	CE <sub>E</sub> 205	嵌入式计算机系统	3
	总学分	15		总学分	15

<b>第 5 学期</b>			<b>第 6 学期</b>		
CE <sub>E</sub> 301	单项项目 I	3	CE <sub>E</sub> 302	单项项目 II	3
CSC <sub>E</sub> 301	编程语言和语法指导工具	3	CE <sub>E</sub> 303	计算机体系结构	3
ELE <sub>E</sub> 301	信号与系统	3	ELE <sub>E</sub> 302	系统控制	3



续表

课程	描述	学分	课程	描述	学分
SWE <sub>E</sub> 301	软件工程	3	ELE <sub>E</sub> 303	数字信号处理	3
	专业选修 A	3		专业选修 B	3
	总学分	15		总学分	15

### B.8.4 计算机工程 BoK 与课程体系 E 之间的映射

对于下表的解释请参见 B.1.2。

BoK 领域 课程	C	C	C	D	E	N	P	S	S	S	S	S	A	D	L	P
	A	A	A	I	S	W	P	E	G	P	R	W	C	S	A	R
	E	L	O	G	Y	K	P	C	P	E	M	D	F	C	L	S
最少核心知识体系学时	50	30	60	50	40	20	20	20	30	35	20	45	30	30	20	30
CE <sub>E</sub> 101								1-5		1-5						
CE <sub>E</sub> 201						1-8										
CE <sub>E</sub> 202			1-7													
CE <sub>E</sub> 203							1-10									
CE <sub>E</sub> 204								6-11		6-10						
CE <sub>E</sub> 205					1-12											
CE <sub>E</sub> 303			8-11													
CSC <sub>E</sub> 101		1-3				1-3				1-4	1-3					
CSC <sub>E</sub> 201		1-10														
CSC <sub>E</sub> 202						1-3		1-4			1-6					
CSC <sub>E</sub> 301			2-5									4-10, 12				

续表

BoK 领域 课程	C A E	C A L	C A O	D I G	E S Y	N W K	P P P	S E C	S G P	S P E	S R M	S W D	A C F	D S C	L A L	P R S
ELE <sub>E</sub> 101	1-6															
ELE <sub>E</sub> 102				1-7												
ELE <sub>E</sub> 201	7-10			8-10												
ELE <sub>E</sub> 202	4-8															
ELE <sub>E</sub> 301								1-8								
ELE <sub>E</sub> 302				7-10												
ELE <sub>E</sub> 303								7-8								
MTH <sub>E</sub> 101														1-9		
MTH <sub>E</sub> 102																1-9
MTH <sub>E</sub> 103													1-7			
MTH <sub>E</sub> 104															1-10	
MTH <sub>E</sub> 201																
SWE <sub>E</sub> 101												1-7				
SWE <sub>E</sub> 102		1-8														
SWE <sub>E</sub> 301												8-12				
核心知识体系 单元覆盖	1-10	1-8	1-11	1-10	1-12	1-8	1-10	1-11	1-8	1-12	1-6	1-10	1-7	1-9	1-10	1-9
附加知识体系 单元	11-12	10		11	13	9-11	11		9-10		7-8	11-12				10

### B.8.5 课程体系 E-课程概述

#### CE<sub>E</sub>101: 计算机工程的概念

通过各种不同应用背景中计算机工程发展的例证, 展现其中的硬件系统和软件

系统的运用包括简单的设备、嵌入式系统、重要人机界面系统、涉及计算机通信的系统,以及具备敏感特性的系统,如安全关键系统;涉及电子、软件、人机界面、使用的工具、系统和工程维度的问题。

先修要求:2 门微积分课程和 2 门物理学课程(在进入此培养方案之前完成)。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-SEC 1-5, CE-SPE 1-5。

### **CE<sub>E</sub>201: 网络与通信**

计算机和计算机通信;安全性、可靠性问题;速度、容量度量、可靠性度量;物理现实和局限性;无线可能性;通信网络体系结构;计算机网络协议;基本拓扑结构演变;局域和广域网络;客户-服务器计算;数据完整性和数据安全性、问题和解决方案;性能问题;网络管理;移动计算的特性和特殊问题。

先修要求:2 门微积分课程和 2 门物理学课程(在进入此培养方案之前完成)。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-NWK 1-8。

### **CE<sub>E</sub>202: 计算机组成**

数字逻辑的基本元素及其在计算机构建中的应用;计算机执行情况的寄存器层次描述和计算机的功能性组织;计算机体系结构的必备组件;现代计算机系统的主要功能部件;机器码的特性;指令格式和寻址方式。机器码和汇编编程要素;存储器层次结构和组织;处理器和外围设备之间的接口和通信;提供实验室实践的软硬件实验,以连接存储器、外设部件和计算机系统。

先修要求:CE<sub>E</sub>101。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-CAO 1-7。

### **CE<sub>E</sub>203: 计算机工程职业问题**

与计算机工程相关的伦理问题的检视,在经典哲学的伦理理论的框架内对这些问题进行论述;法律和准法律(即政策和规定)问题;在道德决策过程、隐私和保密、计算机犯罪、职业准则和责任、职业实践、系统安全性和一般的网络空间安全、计算机对社会的影响等方面的话题。

先修要求:进入第 2 年。

学分:3;授课学时:42;实验学时:0。

覆盖 BoK:CE-PPP 1-10。

**CE<sub>E</sub>204: 计算机系统工程**

计算机工程系统的开发方法;特殊问题和要点;生命周期的概念,生命周期模型的特性,典型的生命周期的阶段,质量问题;过程和过程改进;团队,团队选择,团队角色,团队工作的要素等问题;支持工具,标准和技术的选择;与不同阶段相关的技术和方法;设计的特殊问题和与折衷有关的问题,硬件/软件折衷的特殊问题;测试;维护;项目管理。

先修要求:CE<sub>E</sub>101。

学分:3;授课学时:28;实验学时:28。

覆盖 BoK:CE-SEC 6-10, CE-SPE 6-10。

**CE<sub>E</sub>205: 嵌入式计算机系统**

嵌入式系统的特性,特定问题,特殊问题;在计算机工程中的角色;嵌入式微控制器,嵌入式软件;实时系统,定时和调度问题;测试和性能问题,可靠性;低功耗计算,能源,泄漏;设计方法,支持此类系统开发的软件工具;维护和升级问题;网络化嵌入式系统。

先修要求:2 门微积分课程和 2 门物理课程。

学分:3;授课学时:28;实验学时:28。

覆盖 BoK: CE-ESY 1-12。

**CE<sub>E</sub>301: 单项项目 I**

跨越两个学期的综合项目,在导师的指导下解决某个重要技术问题;学生要展示其应用课程中训练的方法解决问题的能力,学生需要完成一篇关于此项工作的最终论文,加上一个可运行的系统,两部分一起来获得最终评价。

先修要求:进入第 3 年。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:取决于项目。

**CE<sub>E</sub>302: 单项项目 II**

单项项目 I 的延续。

先修要求:进入第 3 年。

学分:3;授课学时:0;实验学时:42。

覆盖 BoK:取决于项目。

**CE<sub>E</sub>303: 计算机体系结构**

与现代计算机体系结构相关的设计原则;性能和成本考虑;受到诸如操作系统

和窗口系统、高级语言、网络、安全考虑等因素的特性影响的体系结构特点;处理器实现策略,微程序设计,流水线,CISC 和 RISC,向量处理器;存储器层次结构,缓存,高性能主机的虚拟内存组织;专用部件和设备;提供不同类型的计算机体系结构的设计和经验的简单演示,如内存体系结构,I/O 总线子系统,专用体系结构、并行处理、分布式系统;探索在可运行计算机系统的设计、实现、仿真中的硬件软件问题以及折衷方法。

先修要求:CE<sub>E</sub>202。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-CAO 8-11。

### CSCE101:计算机和信息系统

不同种类的数据表示;机器码和汇编语言代码编写的元素;操作系统的角色和作用(包括网络,电子邮件和分布式系统)以及相关的功能;编程语言的层次、工具和库;各类应用,包括有关软件(文字处理、数据库、电子表格)的功能描述和它们的运用;接口软件的人机交互性、重要性和相关性;计算机交互的元素,包括屏幕设计和界面的期望特性;WEB 基础;信息检索中的浏览器和搜索引擎运用;简单网页构建;信息服务器的描述;搜索策略;信息存储和检索;版权和知识产权的法律问题。

先修要求:进入第 1 年。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-NWK 1-3, CE-CAL 1-3, CE-SPE 1-4, CE-SRM 1-3。

### CSC<sub>E</sub>201:算法分析与设计

离散概率的基本思想和结果;支持复杂性和性能度量所需的数学基础;来自计数的基本概念;图和树的概念;支撑算法设计的基本策略;计数、搜索、排序的基本算法,哈希表、符号表,队列,树和图的操作;一些简单任务的分布式算法;可计算性理论基础;与安全的相关性;算法设计与分析和软件设计与实现的相关性。

先修要求:MTH<sub>E</sub>101, SWE<sub>E</sub>101。

学分:3;授课学时:28;实验学时:28。

覆盖 BoK:CE-CAL 1-10。

### CSC<sub>E</sub>202:操作系统和以网络为中心的计算

操作系统的功能和角色;主要组件;设计考虑;操作系统分层设计方法,包括对设计的主要影响;高级语言,实时问题,网络,安全,多媒体;文件系统,层次设计;进程管理,调度策略;资源分配策略;并发性,同步原则,死锁避免;设备驱动程序和接口;以网络为中心的计算;对不同平台和移动性的考虑。

先修要求: CSC<sub>E</sub>101。

学分: 3; 授课学时: 42; 实验学时: 14。

覆盖 BoK: CE-NWK 1-3, CE-SEC 1-4, CE-SRM 1-6。

### **CSC<sub>E</sub>301: 编程语言和语法指导工具**

语言展的历史; 语言, 编程, 脚本, 标记, 规范的不同韵味; (不同) 语言的角色, 特点, 比较; 不同编程的范式, 意义, 主要应用领域, 必要性, 功能, 逻辑, 面向对象的语言; 并发性; 语言设计的目的和目标; 语言设计的原则, 包括局限性; 语言设计和翻译处理的相互作用; 翻译的基本方法; 翻译的目的和目标; 翻译的主要组成部分和它们的实现; 库设计, 独立编译, 设计考虑和实现。

先修要求: 2 门微积分课程和 2 门物理学课程(在进入培养方案前完成)。

学分: 3; 授课学时: 28; 实验学时: 28。

覆盖 BoK: CE-SWD 4-10, 12。

### **ELE<sub>E</sub>101: 电子技术基础**

对基本电气量的介绍, 如电荷、电流、电压, 能量和功率; 对经典动力学, 静电学和磁学的介绍; 基本定律, 如基尔霍夫定律、欧姆定律、戴维南定理, 诺顿定理; 电阻电路和网络, 无功电路和网络; 电容、电感、衰减、变压器; 材料的电学性质; 二极管和二极管电路; MOS 晶体管和偏置, MOS 逻辑系列。

先修要求: 2 门微积分课程和 2 门物理学课程(在进入培养方案前完成)。

学分: 3; 授课学时: 42; 实验学时: 14。

覆盖 BoK: CE-CAE 1-6。

### **ELEE102: 数字电路 I**

基本开关理论, 组合逻辑电路; 组合电路的模块化设计; 存储单元; 时序逻辑电路; 数字系统设计; 对电子、通信和电力应用中使用的的基本类型的电路和电子网络的理解和分析。

先修要求: 2 门微积分课程和 2 门物理学课程(在进入培养方案前完成)。

学分: 3; 授课学时: 42; 实验学时: 14。

覆盖 BoK: CE-DIG 1-7。

### **ELE<sub>E</sub>201: 数字电路 II**

对 MOS 系列和电路的回顾; 双极晶体管和逻辑系列; 数字参数和问题; 外部存储单元; 接口逻辑系列和标准总线; 数字系统设计基础, 包括状态图; 建模与仿真, 相关工具运用; CAD 工具运用; 为可测试性以及其它特性开展的设计; 验证和确认的

问题;形式化验证。

先修要求:ELE<sub>E</sub>102。

学分:3;授课学时:28;实验学时:28。

覆盖 BoK:CE-DIG 8-10, CE-CAE 7-10。

### **ELE<sub>E</sub>202:模拟电路**

数据转换问题,A/D 和 D/A 电路;电压和电流源;高、低通滤波器、切比雪夫和巴特沃斯逼近,Sallen-Key 滤波器;负反馈;运算放大器电路;对双极型晶体管的介绍。

先修要求:ELE<sub>E</sub>201。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-CAE 4-8。

### **ELE<sub>E</sub>301:信号与系统**

信号与系统的概念,包括连续时间的和离散时间的;信号处理;信号的对称性和正交性;系统的线性度和时间不变量;系统的脉冲响应和阶跃响应;频率响应、正弦分析、卷积和相关;时间采样和幅度量化;拉普拉斯变换;傅里叶分析,滤波器;运用 Z 变换的离散时间信号和系统分析;反变换过程。

先修要求:ELE<sub>E</sub>201, ELE<sub>E</sub>202。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-SGP 1-6。

### **ELE<sub>E</sub>302:系统控制**

复数,叠加,复合系统的回顾;频域表示;拉普拉斯变换表示;时域的系统表示;一阶和二阶系统;衰减。

先修要求:ELE<sub>E</sub>301。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-DIG 7-10。

### **ELE<sub>E</sub>303:数字信号处理**

数字信号处理(DSP)的目的、理论和概念,DSP 在计算机工程的背景下的角色;数字光谱分析;离散傅立叶变换的应用,卷积类型;滤波,数字滤波;变换;离散时间信号;采样问题;包括图像处理,音频处理的应用;相关软件工具的运用。

先修要求:ELE<sub>E</sub>301。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-SGP 7-8。

### **MTH<sub>E</sub>101: 计算的离散结构**

集合、关系和函数的基本数学概念, 以及涉及它们的操作; 逻辑及其角色, 命题逻辑, 真值表, 等价性问题; 局限性; 谓词逻辑, 它的能力和它的局限性, 在计算机工程背景下的相关性; 证明技术; 常见的数学概念, 如图, 树; 具有代表性的问题; 这些与计算机工程的关联; 递归; 计数; 组合; 这些思想与计算机工程的关联。

先修要求: 1 门微积分课程(在进入本培养方案之前完成)。

学分: 3; 授课学时: 42; 实验学时: 0。

覆盖 BoK:CE-DSC 1-9。

### **MTH<sub>E</sub>102: 应用概率与统计**

随机性, 有限概率空间, 概率测度, 事件; 条件概率, 独立性, 贝叶斯定理; 离散随机变量; 二项分布和泊松分布; 均值和方差的概念; 连续随机变量; 指数和正态分布; 概率密度函数; 均值和方差的计算; 中心极限定理和正态分布的含义; 采样的目的和特性; 估计的特性, 点估计, 区间估计; 最大似然原理方法; 最小二乘法; 置信区间; 对一到两个样本的估计; 模型和相关假设的发展; 假设公式的特性, 空和替代假设, 假设检验; 对假设检验中的 t-检验和卡方检验的接受准则; 相关与回归; 马尔可夫的过程, 离散时间系统和连续时间系统; 支持数据分析的软件包。

先修要求: 2 门微积分课程(在进入本培养方案之前完成)。

学分: 3; 授课学时: 42; 实验学时: 14。

覆盖 BoK:CE-PRS 1-9。

### **MTH<sub>E</sub>103: 微积分与几何**

基本微分和积分的回顾; 技术和方法; 一元积分和二重积分; 简单的微分方程和求解; 复数; 向量微积分; 适当的图形软件包支持的图形概念

先修要求: 2 门微积分课程(在进入本培养方案之前完成)。

学分: 3; 授课学时: 42; 实验学时: 14。

覆盖 BoK:CE-ACF 1-7。

### **MTH<sub>E</sub>104: 线性代数**

基、向量空间和正交性; 线性系统的矩阵表示; 矩阵变换; 线性变换; 线性系统的求解; 非线性系统的求解; 行列式; 特征向量和特征值; 适用线性代数的软件包的运用。

先修要求: MTH<sub>E</sub>101。



学分:3;授课学时:42;实验学时:14。

覆盖 BoK:CE-LAL 1-10。

### **MTH<sub>E</sub>201: 工程师数学**

数值方法及其在工程中的运用介绍;仿真和建模:基本原理和技术;工程中的应用;支持工程仿真和建模的工具运用;排队论,包括系统仿真和建模,排队方法;适当的统计软件包运用;复数;傅立叶变换。

先修要求:MTH<sub>E</sub>102, MTH<sub>E</sub>103, MTH<sub>E</sub>104。

学分:3;授课学时:42;实验学时:14。

覆盖 BoK:无。

### **SWE<sub>E</sub>101: 编程基础**

需求和规范的概念介绍;与编程语言和翻译相关的基本概念;基本编程,原始数据类型,操作,简单语言构成;简单算法和涉及计数问题的解决方案,包括计数,扫描元素,选择元素(如最大值和最小值),迭代;数组,字符串和简单预定义类的运用;作为基本抽象机制的流程或方法;与这些流程或方法相关的设计和构建的原则;简单库,类的运用;软件质量的几个方面;软件测试和验证的相关活动

先修要求:进入第1年。

学分:3;授课学时:28;实验学时:28。

覆盖 BoK:CE-SWD 1-7。

### **SWE<sub>E</sub>102: 程序设计基础**

谓词逻辑的概念;面向对象的编程,方法,类,信息隐藏和继承的思想;基本算法,分类和搜索;基本数据结构,链接的数据结构,用户定义类;递归的概念,效益和问题;异常处理;API 运用;简单图形编程;软件设计的概念。

先修要求:SWE<sub>E</sub>101。

学分:3;授课学时:28;实验学时:28。

覆盖 BoK:CE-CAL 1-8。

### **SWE<sub>E</sub>301: 软件工程**

软件工程,软件工程师的角色;软件评价及其原则,软件生命周期模型;需求,规范,设计实现的概念;主要技术;维护的重要性;在软件开发过程各个阶段的质量事宜;过程的概念;软件过程成熟度模型;软件过程改进;软件工程各方面,软件重用的好的实践及其重要的益处;验证和确认;度量的运用;工具的选择和运用;团队的性质和结构;作为软件工程活动的人机界面;相关的生命周期;标准;相关库的运用;实

践活动的重要性;作为工程师一项重要技能的团体活动。

先修要求: SWE<sub>E</sub>101, SWE<sub>E</sub>102。

学分:3;授课学时:28;实验学时:28。

覆盖 BoK: CE-SWD 8-12。

## 附录 C 计算机工程的实验

本报告的这个附录描述教学中可能用到的实验的配置,这将有助于计算机工程培养方案中的教学实验的设计。本附录对应报告中 4.4 节、6.3.3 节的讨论。CE2016 课程指导委员会不会为任何产品或产商背书,这里列举的项目仅仅作为一个指南,以帮助在实验室的环境中为计算机工程专业方向的学生开发实验。

### C.1 电路与电子学

概述:实验室仪器的实验作用;电压、电流、电阻、频率与波形的测量;电路元素的建模与设计;滤波器设计、搭建、模拟;周期信号的组成。

典型配置:

每个工作台(1~2 名学生使用)应当包含:

- 构建电路的平台/面包板;
- 三端输出的直流电源;
- 双通道混合信号示波器;
- 万用表;
- 函数波形/任意波形发生器;
- 带有电路级建模与仿真工具和仪器控制的计算机。

测试仪器可以是独立的,也可以集成到平台或面包板中,或是通过个人计算机使用属于院系或学生个人的仪器。

在 2016 年,这些仪器的供应商包括安捷伦(Agilent Technologies)、国家半导体(National Instruments)、泰克(Tektronix)、福禄克(Fluke)、惠普(Hewlett-Packard)等。

典型安排:低难度实验;可每周安排一次 3 学时的实验。

### C.2 计算机系统结构设计

概述:使用硬件描述语言(如 VHDL 或 Verilog),实现一个简单的数据通路与控制过程的设计、仿真、验证;在 32 位或 64 位仿真微处理器上进行汇编语言编程;在 FPGA 上实现一个指令集体系结构的 RTL 模型。

典型配置:

- 可以进行 Verilog 和(或)VHDL 建模和仿真的计算机;

- FPGA 开发板;
  - 支持所选 FPGA 开发板的 FPGA 开发套件;
  - 用于检测 FPGA 输出的示波器和逻辑分析仪。
- 典型安排:低难度实验;可每周安排一次 2 学时的实验。

### C.3 数字逻辑设计

概述:包括了复杂程度逐渐增长的一系列数字电路的实验;小规模集成(SSI)与中规模集成(MSI)的组合逻辑电路;算术电路与时序电路;状态机的分析与综合。

典型配置:

- 用于构建 SSI 和 MSI 模块的数字电路的面板;
- 示波器和逻辑分析仪(独立仪器或集成在面包板中);
- FPGA 开发板;
- 带有 VHDL/Verilog 建模和仿真工具、FPGA 开发套件的计算机。

典型安排:低难度实验;可每周安排一次 3 学时的实验。

### C.4 数字信号处理

概述:可以体现数字信号处理原理、技术的硬件和软件实验;数字信号处理器(DSP)芯片上的编程;实时信号处理算法。

典型配置:

- 数字信号处理器(DSP)开发板/工具套件;
- 带有 DSP 软件开发工具的计算机;
- 混合信号示波器;
- 逻辑分析仪;
- 向量信号发生器;
- 光谱分析仪。

在 2016 年,这些仪器的供应商包括德州仪器(Texas Instruments)、安谋(ARM)、泰克(Tektronix)、安捷伦(Agilent)、罗德与施瓦茨(Rohde & Schwarz)等。

典型安排:较高难度实验;可每周安排一次 3 学时的实验。

### C.5 数字逻辑与系统设计

概述:复杂程度逐渐增长的数字系统的层次化、模块化设计;状态机的设计、分析和综合;计算机辅助数字系统建模、仿真、分析、综合;使用可编程逻辑器件和(或)

FPGA 实现设计。

典型配置:

- 用作主机设计工具的计算机;
- VHDL 和(或)Verilog 建模和仿真工具;
- FPGA 开发板;
- FPGA 与外围部件接口的原型板;
- 支持所选 FPGA 开发板的 FPGA 开发套件;
- 嵌入式处理器软核,或带有嵌入式处理器硬核的 FPGA;
- 示波器;
- 逻辑分析仪;
- 作为 FPGA 输入的测试模式生成器。

在 2016 年,这些仪器的供应商包括德致伦(Digilent)、赛灵思(Xilinx)、阿尔戴(Aldec)、阿尔特拉(Altera)、泰克(Tektronix)等。

典型安排:较高难度实验;可每周安排一次 3 学时的实验。

## C.6 嵌入式系统

概述:包括存储器接口与外围设备直到微处理器的实验;控制外围设备的软件的设计;用于系统控制的软硬件集成。

典型配置:

- 微控制器开发板/工具套件;
- 用于安装软件工具的计算机;
- 所选微控制器的集成开发环境;
- 将外围设备连接到微控制器的带电源的面板;
- 示波器;
- 逻辑分析仪;
- 万用表;
- 三输出直流电源。

在 2016 年,这些仪器的供应商包括意法半导体(ST Microelectronics)、德致伦(Digilent)、Keil、安捷伦(Agilent)等。

典型安排:较低难度实验;可每周安排一次 2 学时的实验。

## C.7 工程学入门

概述:基本的工程技术与实践;让学生学习电路、直流发动机以及无线通信的基

础;一次包括计算机工程课程的一些方面的设计项目,比如一辆使用无线电控制的车;重点在于工程设计、团队协作、沟通技巧以及其他相关活动。通常作为跨学科的课程。

典型配置:多样化的配置,基于设计项目以及课程的倾向而定。

常见安排:较低难度实验;可每周一次 2 学时的实验。

## C.8 网络

概述:根据需求以及设备(比如路由器与交换机)情况设计和实现信息网络;用于数据、音频与视频通信的信息网络上的应用;传输媒介,调制,错误控制,流量控制,局域网,以及以太网的协议;数据通信信号与错误控制的实验;计算中常见的网络上的数据传输与软件特点;使用各种协议的服务器、客户端的实现。

典型配置:

- 连在同一网络中的多台个人计算机;
- 与外部隔离的独立网络;
- 可配置的路由器和(或)交换机;
- 测量网络通信量和网络状态的网络分析仪和(或)软件工具。

在 2016 年,这些仪器的供应商包括 Emona、思科(Cisco)等。

典型安排:较高难度实验;可每周安排一次 3 学时的实验。

## C.9 软件设计

概述:软件建设实验;测试、调试、相关工具等主题;配置管理;底层的文件、设备输入输出接口;系统以及事件驱动编程;编程语言,包括 C、C++、Python、Ruby、Java 以及其他可用于计算机工程项目的适当语言。

典型配置:

- 运行流行操作系统的现代计算平台;
- 管理工程文件和库的集成开发环境;
- 课程中所用编程语言的编译器和链接器;
- 源代码级调试器;
- 用于记录、讲演、文件传输等用途的支持工具;
- 用于分析、仿真和建模的数学软件包。

典型安排:较低难度实验;可每周安排一次 2 学时的实验。

## 附录 D 致谢

在本报告的形成过程中,CE2016 课程指导委员会向全球的计算机工程团体征集了意见和建议。此报告是通过两个主要方法完成的。

其一,CE2016 课程指导委员会成员和其他参与报告编制工作的人员在众多会议上通过论文、座谈和专题讨论会等形式宣讲本报告。专题研讨会的参与者对本报告提供了许多细致的反馈。会议其他议程的一些参与者也对本报告提供了现场反馈,或者联系 CE2016 课程指导委员会成员提供反馈。估计总共有超过 500 人参加了各种此类讨论会(见表 D.1),有很多人对此报告提供了有益的想法和思路。

其二,我们在网上发布了此报告的两个草案版本,以通过网络形式来收集来自个人的建议。同时,我们也通过邮件向相关专业协会的成员发送邀请,请他们审阅此报告。这些专业协会包括 ACM 计算机教育专业组(SIGCSE)、IEEE 教育协会和一些其他组织。

在报告完善过程中收到了众多有益的评价和建议,我们深感荣幸。对具名提供反馈意见的下列人士表示感谢:

Jose L. Aguilar C., Universidad de Los Andes, 委内瑞拉和厄瓜多尔;  
白晓颖,清华大学,北京,中国;

Olga I. Bogoiavlenskaia, Petrozavodsk State University, 卡累利阿,俄罗斯;

Iurii A. Bogoiavlenskii, Petrozavodsk State University, 卡累利阿,俄罗斯;

Tarek El-Bawab, IEEE Communications Society, 美国;

Manuel Gericota, School of Engineering-Polytechnic of Porto, 葡萄牙;

Jorge Guerra, Universidad Nacional Mayor de San Marcos (UNMSM), 秘鲁;

Wilfrido Inchausti, Universidad Nacional Mayor de San Marcos (UNMSM), 巴拉圭;

秦磊华,华中科技大学,武汉,中国;

张亮,复旦大学,上海,中国;

Doug Lyon, Fairfield University, 费尔菲尔德,康涅狄格州,美国;

Clive Maynard, Curtin University, 珀斯,澳大利亚;

Doug Myers, Curtin University, 珀斯,澳大利亚;

Richard Perry, Villanova University, 维拉诺瓦,宾夕法尼亚州,美国;

Carlos Ribeiro, Instituto Superior Técnico, Universidade de Lisboa, 葡萄牙;

Cristian Rusu, Pontificia Universidad Católica de Valparaíso, 智利;

Mitch Thornton, Southern Methodist University, 达拉斯, 德克萨斯州, 美国;  
 Murali Varanasi, University of North Texas, 丹顿, 德克萨斯州, 美国;  
 Timothy Wilson, Embry-Riddle Aeronautical University, 戴托纳海滩, 佛罗里达州, 美国;

唐玉华, 国防科技大学, 长沙, 中国;

王志英, 国防科技大学, 长沙, 中国。

CE2016 课程指导委员会向上述为本报告提供了意见和建议的人士表示衷心的感谢。

表 D.1 CE2016 宣讲活动

年份	日期	活动	地点
2012	2. 29-3. 3	第四十三届 ACM 计算机科学教育技术研讨会	罗利, 北卡罗莱纳州, 美国
2012	10. 3-10. 6	2012 教育前沿会议(FIE)	西雅图, 华盛顿州, 美国
2013	6. 23-6. 26	2013 ASEE 年会及博览会	亚特兰大, 佐治亚州, 美国
2013	10. 23-10. 26	2013 教育前沿会议(FIE)	俄克拉何马城, 俄克拉何马州, 美国
2014	6. 15-6. 18	2014 ASEE 年会及博览会	印第安纳波利斯, 印第安纳州, 美国
2014	10. 22-10. 25	2014 教育前沿会议(FIE)	马德里, 西班牙
2014	11. 3-11. 7	第四届巴西计算机系统工程研讨会	玛瑙斯, 亚马逊州, 巴西
2014	12. 8-12. 10	IEEE 教学、评估、工程学习国际会议(TALE 2014)	惠灵顿, 新西兰
2015	3. 18-3. 20	IEEE 全球工程教育会议(EduCon 2015)	塔林, 爱沙尼亚
2015	4. 23-4. 24	第十三届中国国际软件合作洽谈会(CHINASOFT 2015)	成都, 中国
2015	4. 25	四川大学	成都, 中国
2015	4. 25-4. 26	北京计算机教育研究会	北京, 中国
2015	6. 14-6. 17	2015 ASEE 年会及博览会	西雅图, 华盛顿州, 美国
2015	10. 21-10. 24	2015 教育前沿会议(FIE)	埃尔帕索, 德克萨斯州, 美国
2015	11. 27-11. 29	高校计算机课程教学系列报告会(2015)(UFCE)	天津, 中国



续表

年份	日期	活动	地点
2015	12. 1-12. 2	西安交通大学	西安, 中国
2016	3. 18-3. 22	2016 电气和计算机工程部门主管协会年度会议( ECEDHA)	拉由拉市, 加利福尼亚州, 美国
2016	10. 12-10. 15	2016 教育前沿会议( FIE)	伊利, 宾夕法尼亚州, 美国

## 参考文献

- [ ABET, 2016 ] ABET Evaluation Criteria; <http://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2016-2017/>.
- [ ACM, 1992 ] ACM Code of Ethics and Professional Conduct, adopted 16 October 1992; <https://www.acm.org/about-acm/acmcode-of-ethics-and-professional-conduct>.
- [ ACM/IEEECS, 1999 ] ACM and IEEE Computer Society, Software Engineering Code of Ethics and Professional Practice, 1999; <https://www.computer.org/cms/Publications/code-of-ethics.pdf>.
- [ ACM/IEEECS, 2008a ] Computer Science Curriculum 2008: An Interim Revision of CS2001; <http://www.acm.org/education/CS2008.pdf>.
- [ ACM/IEEECS, 2008b ] Information Technology 2008: Curriculum Guidelines for Undergraduate Degree Programs in Information Technology, November 2008; <http://www.acm.org/education/curricula/IT2008%20Curriculum.pdf>.
- [ ACM/IEEECS, 2013 ] Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, December 20, 2013; <http://www.acm.org/education/CS2013-final-report.pdf>.
- [ ACM/IEEECS, 2015 ] Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, February 23, 2015; <http://www.acm.org/binaries/content/assets/education/se2014.pdf>.
- [ ACM/AIS, 2010 ] IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems, 2010; <http://www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf>.
- [ ACM/IEEECS, 2004 ] Software Engineering 2004, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, IEEE Computer Society Press and ACM Press, August 23, 2004.
- [ AITP ] Association of Information Technology Professionals (AITP), Code of Ethics and Standards of Conduct; [http://c.yimcdn.com/sites/www.aitp.org/resource/resmgr/forms/code\\_of\\_ethics.pdf](http://c.yimcdn.com/sites/www.aitp.org/resource/resmgr/forms/code_of_ethics.pdf).
- [ AITP, 2002 ] Association of Information Technology Professionals, Code of Ethics, 2002; <http://www.aitp.org/?page=EthicsConduct>.

- [ ALIBAB ] Alibabaoglan; <http://www.alibabaoglan.com/blog/gartners-top-predictions-till-2020/>.
- [ Bloom 1956 ] B. S. Bloom, ed., *Taxonomy of Educational Objectives: The Classification of Educational Goals: Handbook I, Cognitive Domain*, Longmans, 1956.
- [ BLS ] Bureau of Labor Statistics; [http://www.bls.gov/emp/ep\\_table\\_102.htm](http://www.bls.gov/emp/ep_table_102.htm).
- [ CHIN ] China Gorman; <http://chinagorman.com/2013/04/16/you-think-we-have-skills-shortages-now-lets-talk-in-2020/>.
- [ CC91 ] Computing Curricula 1991; <http://dl.acm.org/citation.cfm?doid=103701.103710>.
- [ DICT ] Dictionary.com; <http://dictionary.reference.com/browse/soft+skills>.
- [ Dublin ] Dublin Accord; <http://www.ieagrements.org/dublin/>.
- [ EHEA ] European Commission/EACEA/Eurydice, *The European Higher Education Area in 2015: Bologna Process Implementation Report*, Luxembourg: Publications Office of the European Union.
- [ EngC ] Engineering Council; [www.engc.org.uk](http://www.engc.org.uk).
- [ FEANI ] FEANI-European Federation of National Engineering Associations; <http://www.feani.org>.
- [ IEEE ] IEEE, Code of Ethics; <http://www.ieee.org/about/corporate/governance/p7-8.html>.
- [ IEEE, 1990 ] IEEE Code of Ethics, August 1990; [http://ewh.ieee.org/cmte/substations/posted\\_documents/ieee\\_codeofethics.pdf](http://ewh.ieee.org/cmte/substations/posted_documents/ieee_codeofethics.pdf).
- [ IFIP, 1998 ] Harmonization of Professional Standards (Draft Version), October 1998; [http://www.ifip.org/minutes/C99/C99\\_harmonization.htm](http://www.ifip.org/minutes/C99/C99_harmonization.htm).
- [ IRPE ] International Register of Professional Engineers, <http://ncees.org/records/international-registry/>.
- [ ITEEA ] International Technology and Engineering Educators Association; <http://www.iteea.org/>.
- [ INVEST ] Investopedia; <http://www.investopedia.com/terms/s/soft-skills.asp>.
- [ NSPE, 2003 ] National Society of Professional Engineers, NSPE Code of Ethics for Engineers, 2003; <https://www.nspe.org/resources/ethics/code-ethics>.
- [ OVERVIEW ] Computing Curricula 2005, The Overview Report; [http://www.acm.org/education/education/curric\\_vols/CC2005-March06Final.pdf](http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf)
- [ SBS ] Subject benchmark statement: Masters degrees in Computing 2011, The Quality Assurance Agency for Higher Education.
- [ SEEPP ] Software Engineering Ethics and Professional Practices (SEEPP), Code of

Ethics; <https://www.acm.org/about/se-code>.

[Seoul] Seoul Accord; <http://www.seoulaccord.org/>.

[SIGCAS] Special Interest Group on Computers and Society (SIGCAS); (ACM)  
<http://www.sigcas.org/>.

[SSIT] Society on Social Implications of Technology (SSIT) of IEEE; <http://ieeessit.org/>.

[Sydney] Sydney Accord; <http://www.ieagrements.org/sydney/>.

[Washington] Washington Accord; <http://www.ieagrements.org/Washington-Accord>.

