教育教学成果奖申报支撑材料

成果名称：依托“一课三室”科教产融合贯通式培养计算机系统与处理器芯片拔尖创新人才

成果完成人：张科、常轶松、王亚洲、张思、刘士祺、郭晓康、王元卓

2025 年 10 月

支撑材料目录

1 教育教学相关证书

1.1 国家级一流本科课程，2020

1.2 北京高校优质本科课程，2020

1.3 北京高校优质本科教材课件，2021

1.4 北京高校课程思政示范课程、课程思政教学名师，2022

2 教育教学相关奖励

2.1 作为团队成员，入围第三批“全国高校黄大年式教师团队”创建示范活动，2023

2.2 入选教育部第二批虚拟教研室建设试点，2022

2.3 教育部典型虚拟教研室、典型教研成果，2023

2.4 基础学科拔尖学生培养计划 2.0 创新案例奖，2021

2.5 第三届全国高等学校计算机教育教学青年教师优秀论文“一等奖”，2019

2.6 曙光奖教金，2019

3 科技进步、技术发明、科普传播、产学研创新等相关证书与奖励

3.1 国家科技进步二等奖，2012

3.2 中国电子学会科技进步二等奖，2024

3.3 中国计算机学会科技进步一等奖，2023

3.4 北京市科技进步二等奖，2023

3.5 中国人工智能学会吴文俊科技进步奖，2023

3.6 河南省科技进步二等奖，2022

3.7 北京市技术发明二等奖，2015

3.8 中国电子学会技术发明二等奖，2014

3.9 中国产学研创新奖，2021

3.10中国十大科学传播人物，2019

3.11第二十四届北京优秀青年工程师荣誉称号，2020

4 教学改革项目：2019 年北京高等教育“本科教学改革创新项目”立项名单节选

5 教育教学研究论文

5.1 依托实践教学云平台培养计算机系统与处理器芯片拔尖人才，计算机教育，2025

5.2 SIGCSE 2019 长文

5.3 TALE 2021 长文

5.4 处理器芯片设计人才培养探索：一生一芯计划实践经验与教训，计算机学会通讯，2023

5.5 计算机教育研究浅析，计算机学会通讯，2019

5.6 SIGCSE 2018 Poster

5.7 CompEd 2019 Poster

5.8 CompEd 2019 BoF

5.9 ITiCSE 2021 Poster

5.10TURC 2023 Poster

6 其他

6.1 教学云平台已授权 7 件专利，其中美国发明专利 1 件、中国发明专利 5 件、中国实用新型专利 1 件

6.2 教学云平台软件著作权登记证书，2021

6.3 教学平台用户证明,郑州大学计算机与人工智能学院、软件学院，2023

6.4 师资培训活动报道，郑州日报，2023 年 8 月 31 日第 6 版

6.5 教指委“计算机专业系统能力培养实施方案工作组成员”聘书，2019

6.6 学生反馈:国科大 2022-2023 学年春季学期《计算机组成原理》课程学生匿名评估问卷，2018-2019 学年春季学期学生评教结果等材料

国家级一流本科课程

证

书

★ ★ ★ ★ ★

课 程 类 别：线下一流课程

课 程 名 称：计算机组成原理

课 程 负 责 人：张科

主 要 建 设 单 位：中国科学院大学

中华人民共和国教育部

教 育 部

2020年11月

证 书 编 号：2020130219

荣誉证书

中国科学院大学张 科老师：

在2020年北京高校“优质本科课程”项目评选中，您讲授的课程“计算机组成原理与配套实验课程”被评为“北京高等学校优质本科课程”，您被评为“北京高等学校优秀专业课（公共课）主讲教师”。

特发此证！

北京市教育委员会

二〇二〇年十月

荣誉证书

中国科学院大学张 科、常轶松老师：

在2021年北京高校“优质本科教材课件”项目评选中，您主编的课件“计算机组成原理实验课件”被评为“北京高等学校优质本科教材课件”。

特发此证！

北京市教育委员会

二〇二一年九月

荣誉证书

中国科学院大学：

你校张 科、常轶松、赵 然、王亚洲、张红霞、齐 乐负责的“计算机组成原理”入选北京市课程思政示范课程，授课教师入选课程思政教学名师和教学团队。

特发此证！

中共北京市委教育工作委员会　　北京市教育委员会

二〇二二年一月

信息名称： 教育部关于公布第三批“全国高校黄大年式教师团队”创建示范活动入围名单的通知
信息索引： 360A10-04-2023-0017-1 生成日期： 2023-08-31 发文机构： 教育部
发文字号： 教师函〔2023〕8号 信息类别： 教育综合管理
内容概述： 教育部关于公布第三批“全国高校黄大年式教师团队”创建示范活动入围名单的通知

教育部关于公布第三批“全国高校黄大年式教师团队”创建示范活动入围名单的通知

教师函〔2023〕8号

各省、自治区、直辖市教育厅（教委），新疆生产建设兵团教育局，有关部门（单位）教育司（局），部属各高等学校、部省合建各高等学校：

为学习贯彻党的二十大精神，贯彻落实习近平总书记对黄大年同志先进事迹的重要指示和给“全国高校黄大年式教师团队”代表的重要回信精神，根据《教育部关于开展第三批“全国高校黄大年式教师团队”创建活动的通知》（教师函〔2023〕2号）要求，各地各高校扎实推进有关工作，择优推荐成绩突出的团队参与申报第三批“全国高校黄大年式教师团队”创建。经专家评审，北京大学天然药物学教师团队等199个团队入围（见附件），现将名单予以公布，并就深入推进创建活动有关工作通知如下。

各地各高校要以习近平新时代中国特色社会主义思想为指导，组织引导广大高校教师和科研工作者以黄大年同志为榜样，心有大我、至诚报国，教书育人、敢为人先，淡泊名利、甘于奉献，把爱国之情、报国之志融入祖国改革发展的伟大事业之中、融入人民创造历史的伟大奋斗之中，为实施科教兴国战略，强化现代化建设人才支撑奠定坚实基础，为全面建设社会主义现代化国家、全面推进中华民族伟大复兴贡献智慧和力量。各地各高校要按照第三批“全国高校黄大年式教师团队”的创建指标，通过制定创建计划、实施措施、奖励办法、考评方式等，因地制宜地开展创建活动。各地各高校要坚持精神奖励、典型宣传与发展支持相结合，结合现有科研计划和人才计划，积极搭建干事创业平台，在拓展发展通道、承担科研任务、提供保障条件、加大激励力度等方面对入围的团队给予专门支持。

教育部将在重大教育改革试点、重大工程项目建设中，把第三批“全国高校黄大年式教师团队”创建情况作为一个重要观测指标，会同有关部门统筹加大团队建设支持力度，组织开展跨领域的学术交流、联合攻关、研修培训以及与产业技术需求对接等活动，加强团队突出业绩和典型事迹的宣传。

请各省级教育行政部门、有关部门（单位）、部属各高等学校、部省合建各高等学校加强对“全国高校黄大年式教师团队”创建的常态化管理，优化考评方法，形成长效机制，并于每年12月31日前向教育部报送年度团队建设的典型经验、重要进展和下一年度工作计划。教育部将委托相关机构和专家，采取材料审核和实地调研相结合的方式，根据结果认定第三批“全国高校黄大年式教师团队”。

附件：第三批“全国高校黄大年式教师团队”创建示范活动入围名单

教 育 部

2023年8月30日

扫一扫分享本页

发布日期：2023-09-01 来源：教育部 下载 收藏 责任编辑：刘洋

相关阅读

- 教育部公布第三批“全国高校黄大年式教师团队”创建示范活动入围名单

党政机关 政府网站找错 网站声明 网站地图 联系我们

版权所有：中华人民共和国教育部 中文域名：教育部.政务

京ICP备10028400号-1 京公网安备11010202007625号 网站标识码：bm05000001

附件

第三批“全国高校黄大年式教师团队”创建示范活动入围名单

（所在高校按学校标识码排序）

序号	所在高校	团队名称	团队负责人
1	北京大学	天然药物学教师团队	屠鹏飞
2	中国人民大学	习近平新时代中国特色社会主义民商法学教师团队	王利明
3	清华大学	大气复合污染治理教师团队	贺克斌
4	北京交通大学	交通系统科学与工程教师团队	高自友
5	北京工业大学	材料生命周期工程教师团队	崔素萍
6	北京航空航天大学	惯性技术创新教师团队	房建成
7	北京理工大学	飞行器动力学与控制教师团队	胡海岩
8	北京化工大学	生命健康材料教师团队	徐福建
9	北京工商大学	数字商科教师团队	黄先开
10	北京邮电大学	多频微波通信技术教师团队	刘元安
11	北京石油化工学院	低碳节能环保教师团队	宇波
12	中国农业大学	养分资源利用与农业绿色发展教师团队	张福锁
13	北京农学院	现代园艺学教师团队	段留生
14	北京协和医学院	协和医学免疫学教师团队	黄波
15	北京师范大学	理论及计算光化学教师团队	方维海
16	北京语言大学	区域国别研究协同创新教师团队	罗林
17	中央财经大学	财政与税务教师团队	樊勇
18	对外经济贸易大学	国际贸易教师团队	赵忠秀
19	中国人民公安大学	社会安全风险防控教师团队	霍宏涛
20	中央音乐学院	音乐人工智能与脑科学教师团队	李小兵
21	中国政法大学	行政法与法治人才培养教师团队	罗智敏
22	华北电力大学	“新能源电力系统保护控制”教师团队	毕天姝
23	北京工业职业技术学院	智能设备技术教师团队	张春芝
24	北京电子科技职业学院	药品生物技术专业群教师团队	辛秀兰
25	中国矿业大学（北京）	安全科学与工程教师团队	王凯
26	中国石油大学（北京）	重油高效清洁转化教师团队	徐春明
27	中国地质大学（北京）	国土空间治理与美丽中国建设教师团队	白中科
28	中国科学院大学	计算机系统与处理器芯片教师团队	孙凝晖
29	中国社会科学院大学	“媒体融合与国家治理”教师团队	胡正荣
30	南开大学	智能科技教师团队	方勇纯
31	天津大学	激光及光电测试技术教师团队	邾继贵

一、团队基本信息

团　队　名　称	计算机系统与处理器芯片教师团队		
团　队　人　数	32	团队所属一级学科	计算机科学与技术
主要负责人姓名	孙凝晖	职　　称	院士、教授
职　　务	计算机科学与技术学院院长	办　公　电　话	[redacted]
移　动　电　话	[redacted]	电　子　邮　件	[redacted]
单位地址及邮编	北京市石景山区玉泉路19号（甲） 100049		
联　系　人　姓　名	包云岗	职　　务	副院长、教授
移　动　电　话	[redacted]	电　子　邮　件	[redacted]

二、团队成员

姓　名	职称/职务	身份证号	出生年月	政治面貌	学历学位	专业方向
孙凝晖	院士/教授/计算机科学技术学院院长/杰青	[redacted]	1968年3月	中共党员	研究生/博士	高性能计算机系统
包云岗	教授/副院长/副所长	[redacted]	1980年9月	中共党员	研究生/博士	计算机体系结构
陈熙霖	教授/教研室主任/所长/杰青	[redacted]	1965年10月	中共党员	研究生/博士	多模式人机交互
陈云霁	教授/副所长/杰青	[redacted]	1983年2月	中共党员	研究生/博士	计算机体系结构
冯晓兵	教授/教研室主任	[redacted]	1969年11月	中共党员中共党员	研究生/博士	编译系统
张健	教授/教研室主任/杰青	[redacted]	1969年6月	无党派人士	研究生/博士	软件工程

姓　名	职称/职务	身份证号	出生年月	政治面貌	学历学位	专业方向
徐志伟	教授/ 杰青	[redacted]	1956年9月	中共党员	研究生/博士	分布式计算
范东睿	教授	[redacted]	1979年4月	中共党员	研究生/博士	处理器结构设计
韩银和	教授/杰青	[redacted]	1980年11月	中共党员	研究生/博士	集成电路与体系结构
李华伟	教授	[redacted]	1974年6月	中共党员	研究生/博士	系统芯片测试
李晓维	教授	[redacted]	1964年9月	中共党员	研究生/博士	系统芯片测试
陈明宇	教授	[redacted]	1972年10月	中共党员	研究生/博士	计算机体系结构
谭光明	教授/杰青	[redacted]	1980年5月	中共党员	研究生/博士	高性能计算机系统
崔慧敏	教授	[redacted]	1979年9月	群众	研究生/博士	编译系统
王元卓	教授	[redacted]	1978年9月	中共党员	研究生/博士	数据智能与知识计算
王剑	教授	[redacted]	1971年6月	中共党员	研究生/博士	微处理器体系结构
詹剑锋	教授	[redacted]	1976年1月	群众	研究生/博士	操作系统
张福新	教授	[redacted]	1976年6月	群众	研究生/博士	微处理器体系结构
张玉军	教授	[redacted]	1976年6月	中共党员	研究生/博士	计算机网络
武成岗	教授	[redacted]	1969年7月	中共党员	研究生/博士	程序设计语言与环境
张科	教授	[redacted]	1982年10月	群众	研究生/博士	计算机体系结构
沈海华	教授	[redacted]	1971年9月	群众	研究生/博士	计算机体系结构
齐洪钢	教授	[redacted]	1975年9月	中共党员	研究生/博士	计算机体系结构
詹乃军	教授/杰青	[redacted]	1971年5月	中共党员	研究生/博士	理论计算机科学
蔡少伟	教授	[redacted]	1986年11月	中共党员	研究生/博士	理论计算机科学
金蓓弘	教授	[redacted]	1967年6月	中共党员	研究生/博士	系统软件
王伟	教授	[redacted]	1982年7月	中共党员	研究生/博	系统软件

教 育 部 办 公 厅

教高厅函〔2022〕13号

教育部办公厅关于公布第二批虚拟教研室建设试点名单的通知

各省、自治区、直辖市教育厅(教委),新疆生产建设兵团教育局,有关部门(单位)教育司(局),部属各高等学校、部省合建各高等学校,2018—2022年教育部高等学校教学指导委员会:

为贯彻落实"十四五"教育发展规划有关部署,加快虚拟教研室建设,经各地各高校和教育部高等学校教学指导委员会推荐、专家综合评议,我部按工作程序确定了第二批虚拟教研室建设试点名单,现予以公布(名单见附件),并就有关事宜通知如下。

一、加快推进试点建设工作。请第二批虚拟教研室建设试点根据《教育部办公厅关于公布首批虚拟教研室建设试点名单的通知》(教高厅函〔2022〕2号)有关要求,围绕创新教研形态、加强教学研究、共建优质资源、开展教师培训等重点任务,充分借鉴首批试点的实践探索经验,做好虚拟教研室试点建设工作。

二、广泛开展研究交流活动。我部将推动开展虚拟教研室建设课题研究工作,从理念、技术、方法、评价等方面开展新型基层教

学组织研究。通过虚拟教研室微信公众号、《高校智慧教研》(内刊)等平台,促进虚拟教研室建设研究成果和实践经验的交流共享。

三、加强建设质量监测和评价。我部将结合虚拟教研室成员队伍建设情况、教研活动组织频次、教研资源建设数量与质量等监测指标,基于常态化质量监测与评价情况,对试点名单进行动态调整,并适时推出一批示范性虚拟教研室,打造教师教学发展共同体和质量文化,引导教师回归教学、热爱教学、研究教学,提升教育教学能力,为高等教育高质量发展提供有力支撑。

附件:第二批虚拟教研室建设试点名单

教育部办公厅

2022年5月19日

附件

第二批虚拟教研室建设试点名单

序号	教研室名称	学校名称	带头人
1	新结构经济学虚拟教研室	北京大学	林毅夫
2	“101 计划”计算机专业核心课程虚拟教研室	北京大学	郭　耀
3	“101 计划”算法设计与分析课程虚拟教研室	北京大学	汪小林
4	“101 计划”离散数学课程虚拟教研室	北京大学	王捍贫
5	“101 计划”操作系统课程虚拟教研室	北京大学	陈向群
6	面向人文社科学生的大学数学课程群虚拟教研室	中国人民大学	龙永红
7	“101 计划”数据库系统课程虚拟教研室	中国人民大学	杜小勇
8	大学体育课程虚拟教研室	清华大学	刘　波
9	面向沿边地区人居环境改善的城乡规划大数据理论与方法课程虚拟教研室	清华大学	龙　瀛
10	核辐射物理及探测学课程虚拟教研室	清华大学	杨祎罡
11	“101 计划”计算机网络课程虚拟教研室	清华大学	吴建平
12	“101 计划”计算机组成与系统结构课程虚拟教研室	清华大学	刘卫东
13	轨道交通信号与控制专业虚拟教研室	北京交通大学	戴胜华
14	车辆工程专业（轨道车辆智能运维方向）虚拟教研室	北京交通大学	刘志明
15	“101 计划”编译原理课程虚拟教研室	北京航空航天大学	张　莉
16	自然语言处理课程群虚拟教研室	北京理工大学	黄河燕
17	思想政治理论课程群虚拟教研室	北京理工大学	李林英
18	信息对抗技术专业虚拟教研室	北京理工大学	罗森林
19	大学数学课程群虚拟教研室	北京科技大学	郑连存

序号	教研室名称	学校名称	带头人
43	材料力学课程虚拟教研室	中国矿业大学（北京）	周宏伟
44	油矿地质学课程虚拟教研室	中国石油大学（北京）	吴胜和
45	劳动教育与劳动实践课程虚拟教研室	中国劳动关系学院	刘玉芳
46	计算机系统与处理器芯片课程虚拟教研室	中国科学院大学	张　科
47	日本社会与历史文化课程虚拟教研室	南开大学	韩立红
48	信号与系统课程虚拟教研室	南开大学	孙桂玲
49	保密管理专业虚拟教研室	天津大学	马寿峰
50	建筑环境与能源应用工程课程群虚拟教研室	天津大学	张　欢
51	北部地区高校纺织工程专业虚拟教研室	天津工业大学	马崇启
52	信号类课程群虚拟教研室	中国民航大学	韩　萍
53	电子信息工程专业（航空电子电气方向）虚拟教研室	中国民航大学	孙淑光
54	航空外语课程虚拟教研室	中国民航大学	张艳玲
55	机械电子工程专业虚拟教研室	天津理工大学	郑清春
56	职业教育学课程虚拟教研室	天津职业技术师范大学	赵文平
57	思想政治理论课社会实践虚拟教研室	河北大学	柴素芳
58	新闻学专业（中国共产党新闻史教学）虚拟教研室	河北大学	韩立新
59	医学史课程虚拟教研室	河北医科大学	翟海魂
60	生理课程虚拟教研室	河北医科大学	武宇明
61	智能制造新工科建设改革虚拟教研室	燕山大学	赵永生
62	消防工程专业虚拟教研室	中国人民警察大学	李思成
63	边境安全课程虚拟教研室	中国人民警察大学	马长泉
64	国际政治专业虚拟教研室	山西大学	王　毅

序号	教研室名称	学校名称	带头人
207	简明新疆地方史课程虚拟教研室	新疆大学	李　乐
208	面向信创产业的边疆各族数据库人才培养模式研究虚拟教研室	新疆大学	钱育蓉
209	文化与边疆课程思政虚拟教研室	塔里木大学	屈玉丽
210	化工实验课程虚拟教研室	塔里木大学	田维亮
211	西北地区生物科学专业虚拟教研室	塔里木大学	王彦芹
212	西部地区化学工程与工艺专业虚拟教研室	石河子大学	代　斌
213	新疆生产建设兵团农学专业虚拟教研室	石河子大学	张旺锋
214	生物化学与分子生物学课程虚拟教研室	新疆医科大学	关亚群
215	“101 计划”软件工程课程虚拟教研室	国防科技大学	毛新军
216	数据结构课程虚拟教研室	陆军工程大学	陈卫卫
217	运筹学课程虚拟教研室	陆军工程大学	刘华丽
218	航空发动机原理课程虚拟教研室	中国人民解放军空军工程大学	于锦禄

（此件主动公开）

部内发送：有关部领导，办公厅

教育部办公厅　　2022 年 5 月 30 日印发

教育部虚拟教研室建设试点

典型虚拟教研室

计算机系统与处理器芯片课程虚拟教研室

负责人：张科

教育部虚拟教研室建设专家组

2023年12月12日

虚拟教研室建设专家组

关于公布虚拟教研室试点建设典型名单的通知

各虚拟教研室：

为深入贯彻落实党的二十大报告中关于“推进教育数字化”“培养高素质教师队伍”的重要部署，深化虚拟教研室建设，加大典型示范力度，经研究，虚拟教研室建设专家组开展了典型虚拟教研室、典型教研方法、典型教研成果选树工作。

基于各虚拟教研室年报数据，经学科协作组推荐、虚拟教研室建设专家组论证等工作程序，形成虚拟教研室试点建设典型名单，其中典型虚拟教研室50个、典型教研方法100项、典型教研成果50项。现将名单予以公布（见附件）。

请各虚拟教研室借鉴典型经验和做法，开展广泛学习和交流，进一步创新基层教学组织建设，营造智慧教研良好氛围，为打造高等教育一流的教师团队，推动高等教育数字化的高质量发展做出积极贡献。

附件：虚拟教研室试点建设典型名单

虚拟教研室建设专家组

（武汉理工大学代章）

2023年12月12日

附件

典型虚拟教研室名单

序号	虚拟教研室名称	负责人	负责人所在高校
1	“101 计划”计算机专业核心课程虚拟教研室	郭耀	北京大学
2	数据库课程虚拟教研室	杜小勇	中国人民大学
3	鲲鹏计算机系统能力培养课程群虚拟教研室	赵欢	湖南大学
4	计算思维导论课程虚拟教研室	战德臣	哈尔滨工业大学
5	大学计算机课程改革虚拟教研室	郑庆华	同济大学
6	计算机系统与处理器芯片课程虚拟教研室	张科	中国科学院大学
7	北部地区高校电工电子基础课程群虚拟教研室	王淑娟	哈尔滨工业大学
8	轨道交通信号与控制专业虚拟教研室	戴胜华	北京交通大学
9	电气工程及其自动化专业虚拟教研室	胡敏强	南京师范大学
10	电磁场与无线技术专业虚拟教研室	苏东林	北京航空航天大学
11	面向沿边地区人居环境改善的城乡规划大数据理论与方法课程虚拟教研室	龙瀛	清华大学
12	工程材料与机械制造基础（金工）课程群虚拟教研室	孙康宁	山东大学
13	机械设计课程群虚拟教研室	邓宗全	哈尔滨工业大学

典型教研成果名单

序号	典型教研成果	负责人	负责人所在高校
1	思想道德与法治课程知识图谱	燕连福	西安交通大学
2	微生物学知识图谱	李振兴	中国海洋大学
3	现场勘查知识图谱	李伟	铁道警察学院
4	通风工程学知识图谱	程卫民	山东科技大学
5	物流信息管理知识图谱	夏春玉	东北财经大学
6	木材保护与改性课程知识图谱	曹金珍	北京林业大学
7	交通运输导论知识图谱	陈峻	东南大学
8	模拟电路基础知识图谱	何松柏	电子科技大学
9	电工电子课程群知识图谱	王建新	南京理工大学
10	简明新疆地方史知识图谱	李乐	新疆大学
11	中国近现代史纲要知识图谱	李林英	北京理工大学
12	生态旅游知识图谱	田卫民	云南大学
13	冶金物理化学知识图谱	沈峰满	东北大学
14	临床血液检验知识图谱	尚红	中国医科大学

序号	典型教研成果	负责人	负责人所在高校
45	统计学导论知识图谱	金蛟	北京师范大学
46	计算机系统与处理器芯片课程知识图谱	张科	中国科学院大学
47	心理学（公共课）知识图谱	姚本先	合肥师范学院
48	线性代数（2023）知识图谱	刘春凤	华北理工大学
49	计算机科学与技术——软件工程知识图谱	毛新军	中国人民解放军国防科技大学
50	基于 KT-SQEP 方法的“101 计划”计算概论核心课程知识图谱	战德臣	哈尔滨工业大学

荣誉证书

中国科学院大学：

在“基础学科拔尖学生培养计划2.0”工作中探索新理念、新模式、新方法、新标准，“拓展计算机专业核心课程群，实施‘一生一芯’计划，培养国家处理器芯片急需人才”实践成效显著。特授予2021年度

创新案例奖

“基础学科拔尖学生培养计划2.0”秘书组

（清华大学教务处代章）

二〇二一年十一月

第三届全国高等学校计算机教育教学青年教师优秀论文奖

中国科学院大学 张科老师 在《SIGCSE2019》发表的论文

《Computer Organization and Design Course with FPGA Cloud》

荣获第三届青年教师优秀论文一等奖，特发此证，以兹鼓励！

教育部高等学校计算机类专业教学指导委员会

教育部高等学校大学计算机课程教学指导委员会

全国高等学校计算机教育研究会

2019 年 12 月 08 日

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

荣誉证书

张科 同志：

在教育工作中做出突出贡献，荣获2018年度**曙光奖教金**，特发此证，以资鼓励。

中国科学院计算技术研究所

2019年1月25日

国家科学技术进步奖

证　书

为表彰国家科学技术进步奖获得者，特颁发此证书。

项目名称：大规模网络信息监测与服务系统关键技术及应用

奖励等级：二等

获 奖 者：王元卓

中华人民共和国国务院

2012年12月19日

证书号：2012-J-220-2-02-R08

中国电子学会科学技术奖

获奖证书

为表彰中国电子学会科学技术奖获得者，特颁发此证书。

证书号：KJ2023-J2-25-R04

项目名称：中国联通超大规模湖仓一体大数据平台关键技术及应用

奖　　种：科技进步

奖励等级：二等奖

获 奖 人：王元卓

中国电子学会

2024 年 3 月

2023年CCF科技成果奖

科技进步一等奖

授 予

中国科学院计算技术研究所、
中国科学技术出版社有限公司、南方科技大学
“《科幻电影中的科学：
科学家奶爸的＜流浪地球 2＞手绘》”项目

主要完成人 王元卓、崔原豪、陆源、
郑洪炜、沈英汉、江旭晖

CCF理事长　　CCF奖励委员会主席

中国计算机学会
2023年10月27日

编号：CCF-AWRD-AWD-2023-00030

学术荣誉称号

北京市科学技术奖

科学技术进步奖

证书

为表彰北京市科学技术奖获得者，特颁发此证书。

项目名称：科幻电影中的科学普及与科教融合

奖励等级：二等奖

获奖者：王元卓

北京市人民政府

2023年10月31日

No.2022-J13-2-01-R01

吴文俊人工智能科学技术奖

获奖证书

【国科奖社证字第0218号】

获奖项目：《科幻电影中的科学：科学家奶爸的AI手绘》

授奖等级：科技进步奖(科普项目)

获 奖 者：王元卓

中国人工智能学会

2023年03月12日

证书编号：KJJBKP-2022-B-R01

NO.K 006937

河南省科学技术进步奖
证　书

为表彰河南省科学技术进步奖获得者，特颁发此证书。

项目名称：知识计算引擎驱动的科技大数据分析系统及应用

奖励等级：贰等奖

获 奖 者：王元卓

河南省人民政府

2022年3月22日

证书号：　2021-J-060-R01/10

北京市科学技术奖

№ 2015 计-2-002-03

为表彰在推动科学技术进步、对首都经济建设和社会发展作出贡献者，特颁此证，以资鼓励。

获奖项目：**社会化媒体数据的感知与处理关键技术及应用**

获 奖 者：**王元卓**

获奖等级：**贰等奖**

北京市人民政府

二〇一五年十二月

中国电子学会科学技术奖

获奖证书

为表彰中国电子学会科学技术奖获得者，特颁发此证书。

证书号：KJ2014-F2-02-R03

获奖项目：大规模社会化媒体数据的感知与分析处理系统

奖励类别：技术发明

获奖等级：二 等 奖

获 奖 者：王元卓（第三完成人）

二〇一四年十二月

国科奖社证字第0191号

2021年中国产学研合作创新与促进奖
产学研合作创新奖
获奖证书

王元卓 同志

为表彰在产学研协同创新中作出突出贡献的单位和个人，在科技部和国家科技奖励办支持下，设立了中国产学研合作创新与促进奖。经评审，授予你2021年中国产学研合作创新奖。

特颁此证。

证书号：20213164

中国产学研合作促进会

2022年1月

科普中国
CHINA SCIENCE COMMUNICATION

荣誉证书

王元卓：

在“典赞·2019科普中国”活动中被评为：

十大科学传播人物

特发此证，以兹鼓励！

中国科学技术协会
2019年12月

荣誉证书

授予 张 科 同志：

第二十四届北京优秀青年工程师荣誉称号

北京市科学技术协会　　北京市人力资源和社会保障局

二零二零年十一月

关于公布2019年北京高等教育 “本科教学改革创新项目”的通知

发布时间：2019-11-06

京教函〔2019〕552号

各普通本科高等学校：

为贯彻《中共北京市委 北京市人民政府印发〈关于统筹推进北京高等教育改革发展的若干意见〉的通知》（京发〔2018〕12号）精神，进一步强化教育教学工作，形成教学奖励项目与科研奖励项目、教学人才项目与科研人才项目、教育教学成果与科学研究成果同等重要、相互促进、融合发展的新局面，我委组织了2019年北京高等教育“本科教学改革创新项目”申报及评审工作，现将市教委立项项目予以公布（见附件）。

各高校要加强本科教学改革创新项目管理，做好经费支持及政策保障工作，把握好项目建设进度，确保项目建设取得预期目标。在此基础上，进一步完善校内本科教学改革支持体系，积极开展系统性、前瞻性、持续性研究及探索，加快构建体现北京高等教育优势与特色的一流人才培养体系。

附件：2019年北京高等教育“本科教学改革创新项目”立项项目名单

北京市教育委员会

2019年11月6日

附件

2019年北京高等教育“本科教学改革创新项目”立项项目名单

序号	项目名称	项目负责人	项目主持学校	类型备注
1	清华大学本科通识课程体系改革探索与实践	彭刚	清华大学	重大项目
2	ICT领域“四融合”新工程教育体系设计与实践	乔建永	北京邮电大学	重大项目
3	新时代思政课质量和水平提升路径研究	王洪元	北京林业大学	重大项目
4	中医学领军人才培养模式改革创新项目	徐安龙	北京中医药大学	重大项目
5	本科生学业发展的动因系统模型研究	涂清云、张生	北京师范大学	重大项目
6	基于“课程思政”的卓越新闻传播人才培养创新	高晓虹	中国传媒大学	重大项目
7	新时代高校劳动教育实施体系建构	刘向兵	中国劳动关系学院	重大项目
8	促进医学教育协同发展的区域协作机制研究	管仲军	首都医科大学	重大项目
9	回望党办大学初心，挖掘红色资源——基于延河联盟的高校思政教育研究与实践	张军	北京理工大学	重大委托项目
10	“五位一体”的课程思政体系建设	苏海佳	北京化工大学	重大委托项目
11	北京市属高校创新型人才培养管理机制研究与改革实践	黄先开	北京工商大学	重大委托项目

序号	项目名称	项目负责人	项目主持学校	类型备注
155	计算机组成原理敏捷教学模式探索与实践	张科	中国科学院大学	
156	混合式教学模式在英语课程中的应用研究	王蕙	中国社会科学院大学	
157	一流大学科教融合与个性化人才培养研究——以中国社会科学院大学为例	刘文瑞	中国社会科学院大学	
158	以科研育人为抓手的马克思主义理论教育模式创新与质量提升	秦国伟	中国社会科学院大学	
159	基于大工程的土木工程专业复合型创新人才培养体系构建与实践探索	张建伟	北京工业大学	
160	跨专业大类招生的材料本科人才培养研究	崔素萍	北京工业大学	
161	瞄准时代需求，以服务智慧北京为目标的计算机专业改革建设	杜金莲	北京工业大学	
162	依托优质国际办学资源，提升创新人才培养能力--以北工大都柏林学院为例	张优良	北京工业大学	委托项目
163	精准资助育人模式的创新与实践探索	高国伟	北方工业大学	
164	面向新工科创新人才培养的 MSP 教学模式	王力	北方工业大学	
165	理实联合科教融合虚实结合智能建造专业人才培养模式探索与实践	宋志飞	北方工业大学	
166	基于“以学生为中心”的线上线下混合式教学模式构建——以新闻学为例	王擎	北京工商大学	
167	基于化妆品工程实际的创新性教学实践	赵华	北京工商大学	

序号	项目名称	项目负责人	项目主持学校	类型备注
220	基于设计思维教学的艺术设计课程教学模式改革与实践应用	郭丽	北京吉利学院	
221	民办应用型本科人工智能专业课程实践体系构建-以北京吉利学院为例	施一飞	北京吉利学院	
222	从专业竞赛切入 全面推动专业建设——影视专业人才培养模式的新探索	王锟	首都师范大学科德学院	
223	新时代金融学专业国际化人才培养研究与实践	孙雪梅	北京工商大学嘉华学院	
224	基于 CDIO 工程教育理念的专业建设研究与实践	夏素霞	北京邮电大学世纪学院	
225	“四位一体”目标下的学校体育改革与实践研究	戴雄	北京工业大学耿丹学院	
226	基于“数字建筑”产业需求的工程管理人才培养模式创新实践	张淑红	北京工业大学耿丹学院	
227	旅游酒店教师 5H 认证培训	曾丽婷	北京第二外国语学院中瑞酒店管理学院	

COMPUTER EDUCATION

CC 计算机教育

CCF《计算领域高质量科技期刊分级目录》T2类期刊
中国知网《学术精要数据库》高影响力论文期刊
中国最具国际影响力学术期刊
BIBF2023中国精品期刊
《中国学术期刊影响因子年报》统计源期刊

CN 11－5006/TP

05
2025
总第365期

www.jsjjy.com

专题策划："101计划"实践平台建设（一）

产教融合驱动的"101计划"核心课程实践平台创新与应用

依托实践教学云平台的计算机系统与处理器芯片拔尖人才培养

基于学习成效的混合式程序设计课程教学改革

面向网络空间安全专业的"本硕博"贯通培养机制

ISSN 1672-5913
05 >
9 771672 591257

中华人民共和国教育部主管　清华大学主办

文章编号：1672-5913(2025)05-0009-06　　中图分类号：G642

依托实践教学云平台的计算机系统与处理器芯片拔尖人才培养

张　科[1,2]

（1. 中国科学院 计算技术研究所，北京 100190；
2. 中国科学院大学 计算机科学与技术学院，北京 100049）

摘　要：针对新时代对计算机系统与处理器芯片方向拔尖人才的迫切需求，在“101 计划”的引领与指导下，提出以培养学生计算机系统能力为出发点，统筹理论课教室、研讨课实验室与虚拟教研室教学资源，依托科研团队自研面向计算机系统教学的“思沃”实践云平台，介绍使用云上敏捷开发流程服务计算机系统与处理器芯片的实践教学，说明教学效果，为加快我国计算机领域拔尖人才的培养提供一条可能路径。

关键词：计算机系统能力；芯片人才；实践平台；虚拟教研；FPGA；101 计划
DOI:10.16512/j.cnki.jsjjy.2025.05.009

1　背景与相关工作

当今世界，互联网、云计算、人工智能、大语言模型等新兴数字化信息技术深刻改变着人类的生产、生活、学习乃至思维方式，开辟了巨大的发展前景。日新月异的科技进步带来了对新型创新人才的迫切需求，而人才的成长又离不开长期的教育培养。党的二十大报告提出：“教育、科技、人才是全面建设社会主义现代化国家的基础性、战略性支撑”“全面提高人才自主培养质量，着力造就拔尖创新人才”[1]。作为数字化信息技术的基石，计算机系统与处理器芯片的重要性不言而喻。

近年来，为培养计算机系统与芯片方向人才，国内各高校以提升学生计算机系统能力[2-6]为抓手、以设计简易处理器（CPU）与计算机系统为目标，开展了一系列计算机系统能力培养实验与实践教学活动[7-10]。在计算机组成原理、计算机系统结构、计算机系统导论等专业课程教学中，一线教师通过多种多样的技术手段开展此类实践实验活动，例如软件模拟器、分离元器件电路板、FPGA（Field Programmable Gate Array，可编程门阵列芯片）板卡等。自 2016 年开始，中国科学院大学（以下简称国科大）计算机组成原理课程教学团队联合中国科学院计算技术研究所科研团队，利用异构 FPGA 与云计算等数字化新兴技术手段构建实验实践教学云环境，将实验环境中的 FPGA 硬件设施云化，支撑学生在线开展 CPU 的微架构设计、逻辑代码综合与 FPGA 部署评测，并在教育部“101 计划”的引领与指导下，统筹“计算机系统与处理器芯片课程虚拟教研室”教学资源，探索建立系统与芯片拔尖人才培养机制。

2　学情分析

国科大计算机组成原理课程每学期约有 100 名选课学生。在开课之初面向学生发放匿名问

基金项目：中国科学院战略性先导科技专项“RISC-V 敏捷设计与跨层优化”（XDA0320300，XDA0320000）；教育部虚拟教研室建设试点项目“计算机系统与处理器芯片课程虚拟教研室”；中国科学院大学“拔尖创新人才培养工程—计算机系统与处理器芯片课程虚拟教研室建设”项目。
作者简介：张科，男，正高级工程师，研究方向为计算机系统结构、异构加速计算、FPGA 云，zhangke@ict.ac.cn。

中国知网　https://www.cnki.net

卷，调查学生对哪些计算机相关领域感兴趣，例如计算理论、算法、编译与编程、体系结构、人工智能（AI）、深度学习、芯片设计、操作系统、人机交互、机器人、生物计算、量子、云计算、元宇宙等。对 2021—2023 年连续 3 年的调查结果进行分析与比对（如图 1 所示），选择“人工智能 / 深度学习”和“算法、应用软件”的学生比例绝对值最高，这体现了学生对软件比较熟悉，也侧面展示出近年的“AI 热”对学生兴趣的影响；对“芯片设计”“体系结构”“编译编程”“操作系统”这 4 组计算机基础方向感兴趣的学生比例呈现较为明显的上升趋势，体现了学生对处理器芯片等“卡脖子”关键技术的关注度；最右侧一列“我对所有技术和方向都感兴趣”呈现出近年来学生的好奇心有所下降，值得关注。

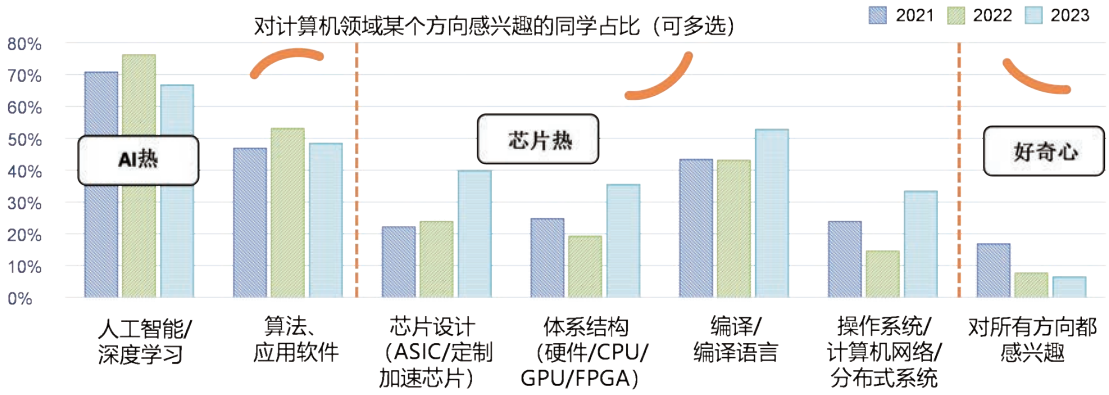
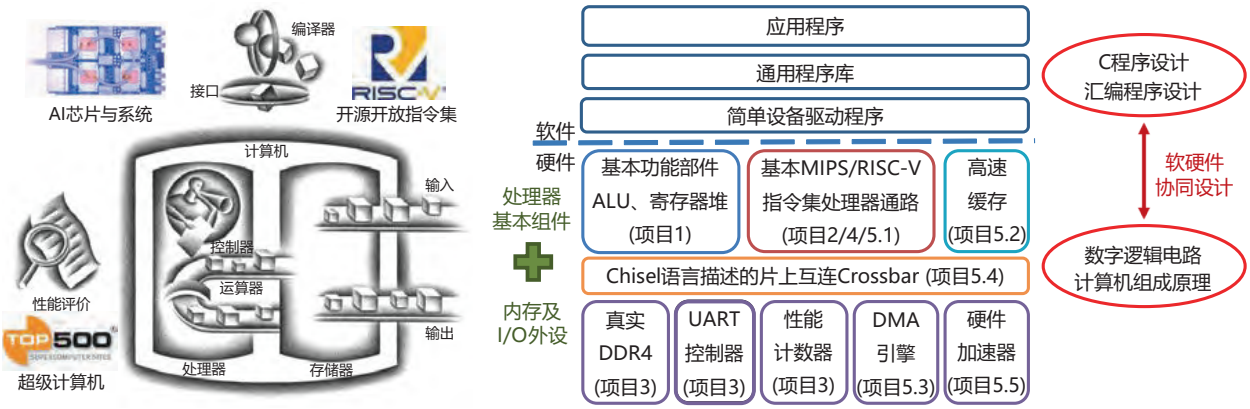


图 1　教学团队针对选课学生的专业兴趣进行课前调查结果统计（3 条短曲线示意了 3 类的变化趋势）

3　计算机组成原理课程教学内容体系

作为计算机专业核心课程之一的计算机组成原理分成理论和实验研讨两个部分。其中，理论部分如图 2（a）所示，主要是结合最新的发展趋势来讲解组成原理的传统基础知识以建立系统概念。新技术与新趋势包括 AI 芯片与系统、超级计算机、RISC-V 开源开放指令集等。实验研讨部分则主要培养学生计算机软硬件设计与调试能力，强化计算思维，加深对理论知识的理解。实践项目体系如图 2（b）所示，不仅包括基础功能部件的算术逻辑运算单元（ALU）与寄存器堆、单周期处理器、高速缓存、硬件加速器等，还会涉及真实 DDR4 内存、简易外设等片上系统芯片相关内容，进一步加深学生对计算机系统的理解，提升设计、调试和评测能力，因此，实验内容与理论知识相互配合，可以帮助学生更好地理解吸收这些知识点并在实操中进行应用。此外，



（a）组成原理的理论知识体系　　（b）组成原理的实践项目体系

图 2　国科大计算机组成原理课程的理论知识体系与实验项目体系

中国知网 https://www.cnki.net

教学团队也会针对性地带领基础略微薄弱的学生复习若干先修课的知识点，包括程序设计、汇编语言、数字逻辑电路等，有效地帮助这类学生顺利开展实践项目。

4　统筹“三室”资源，开展云上教学新形态

统筹“三室”（理论课教室、研讨课实验室、计算机系统与处理器芯片课程虚拟教研室）教学资源，自研面向计算机系统教学的“思沃”实践教学云平台，开展云上教学新形态（如图 3 所示）。

教学团队所讲授的计算机组成原理课程，在常规的理论课课堂教学互动之外，还设计有独具特色的实验课。内容上，实验课与理论课的知识体系交叉融合；形式上，实验课依托自研“思沃”云平台开展实验与实践项目。学生根据实验要求向云平台提交实验代码，平台的计算资源完成代码评测之后再将实验结果反馈回实验室中的学生。同时，教师深度参与教研活动，在虚拟教研室里既开展关于课程内容的研讨、开展跨地域和跨学校的师资交流，也参与调研实验室内外学生的实验项目开展情况，夯实教学成果。教研室内的教师积极对外交流，普及推广教研成果。

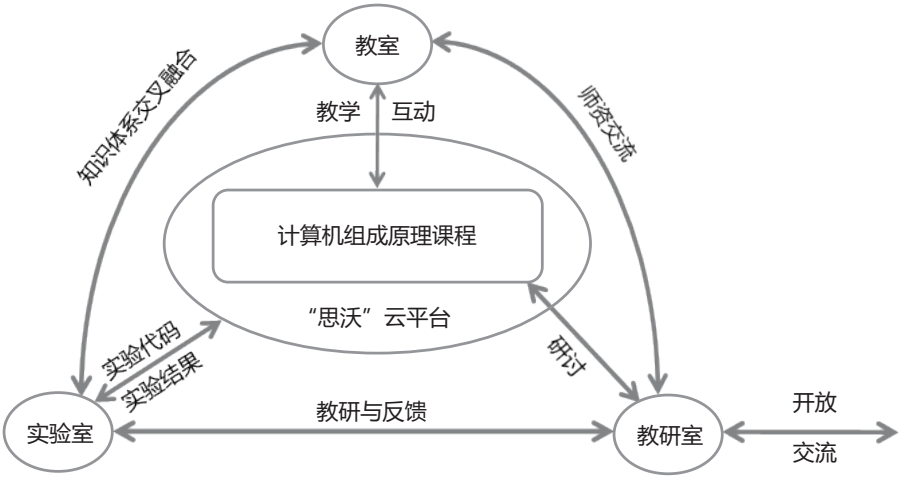


图 3　统筹“三室”教学资源，培养计算机拔尖人才

5　面向计算机系统类课程的“思沃”教学云平台

“思沃（SERVE）”实践教学云平台系统结构如图 4 所示。学生用户通过互联网访问云平台，由若干 x86 服务器构成的容器集群对学生代码进行保存托管、版本管理，以及硬件 Verilog/Chisel 代码的逻辑综合与 FPGA 评测用的比特流文件生成等操作。集群中部署了 Git、容器管理等开源软件和 FPGA 厂商的 EDA 工具软件。与 x86 集群通过局域网连接的云平台后端为自研的异构 FPGA 服务器，内含定制 SoC FPGA 板卡。基于云计算容器化技术，x86 商用服务器、FPGA 服务器以及 FPGA 板卡的数量均可根据云平台服务的活跃学生用户数量进行动态调整，以实现节能并降低运行成本。另外，x86 服务器集群中的软件亦可更换为公有云服务，从而进一步降低部署成本。值得一提的是，这套云平台在新冠疫情期间很好地满足了硬件实验在线教学的需求。

通过近年来持续改进云平台，从 2023 年开始，学生只需一台笔记本电脑并启动可以连通互联网的浏览器软件即可完成课程所要求的实验项目（如图 2（b）所示）。学生通过浏览器登录云平台，在云平台的环境中编辑软硬件代码、提交代码、利用云平台 x86 与 FPGA 异构算力评测代码，最后查看评测结果，十分便捷。具体来讲，学生通过互联网连接“思沃”云平台中的 x86 通用计算服务器集群，x86 服务器再通过局域网与

中国知网　https://www.cnki.net

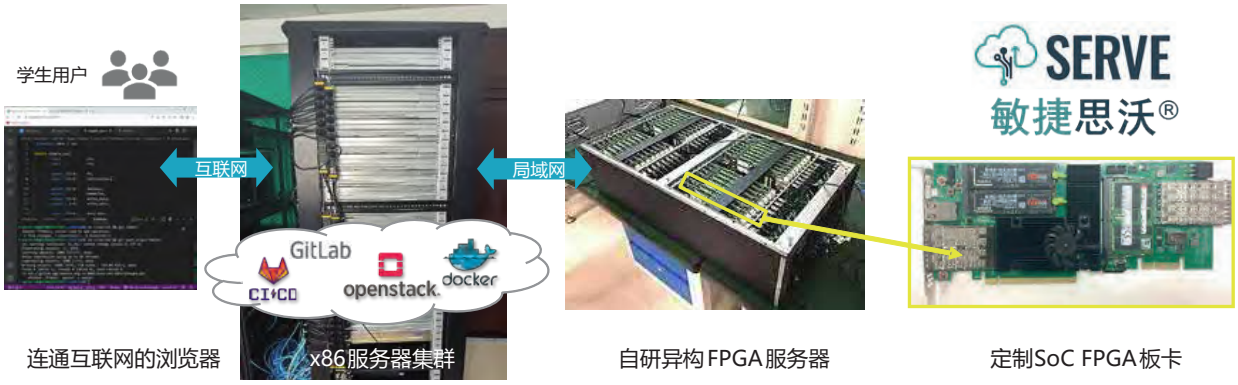


图 4　面向计算机系统实验教学的“思沃”实践云平台硬件基础设施

云平台自研的异构 FPGA 算力服务器连接，将具体的实验待测任务分配到定制的 SoC FPGA 板卡上。学生开展实验的过程中，云平台主要提供两个功能：一是利用 x86 CPU 算力进行待测任务的设计与仿真，二是利用 FPGA 算力进行待测实验任务的系统级原型验证。前者的关键问题是 CPU 云化，这一点可以利用较为成熟的处理器虚拟化等解决方案，而后者的关键在于 FPGA 云化，这里存在诸多技术难点，例如如何增强 FPGA 算力密度、优化虚拟粒度等问题。教学团队与科研团队一同从 FPGA 逻辑自治化、FPGA 算力集群化、异构资源虚拟化和工具流程敏捷化 4 个方向进行了技术探索，实现了处理器芯片的系统级敏捷验证。为支持更多高校教师和学生的使用，“思沃”云平台已在北京、郑州、南京、上海多地部署。相比国外当时主流的 HAPS 平台，“思沃” FPGA 算力密度提高 22%，成本降低 40%，同时“思沃”云平台支持第三方算力接入，并可定制设计实验验证流程。

5.1　FPGA 异构资源虚拟化

为实现 FPGA 云平台的共享使用，FPGA 算力资源虚拟化是其中的关键技术。如图 5 所示，科研团队采取 FPGA 逻辑区域划分与动态部分可重构技术，配合 ARM 通用处理器协同实现虚拟化 [11-12]。通过图 5 中 1—6 个步骤，不同学生用户编写的 CPU 设计会被部署在一颗 FPGA 芯片中的不同逻辑区域。“思沃”云平台的虚拟化性将单节点 FPGA 逻辑资源利用率提升 3 倍，虚拟化可重构时间降低至 30ms。在虚拟化粒度层面，“思沃”云平台可以实现 FPGA 单节点并发部署不同学生的多个独立评测任务，从而降低部署成本和运行成本。

5.2　基于云开发环境的全实验流程覆盖

“思沃”实践云平台借鉴互联网领域较为成

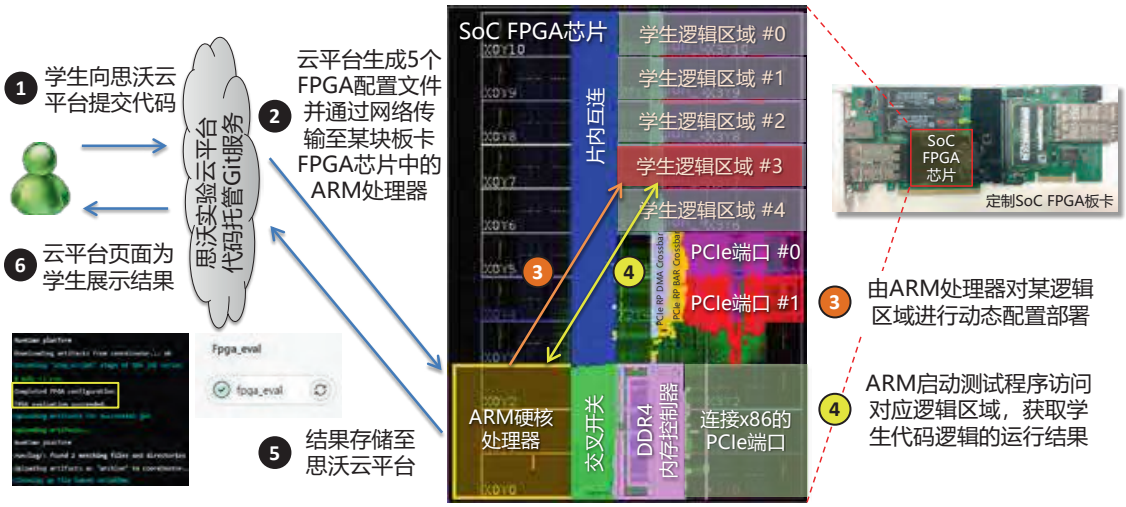


图 5　FPGA 逻辑区域划分与动态部分可重构技术

中国知网 https://www.cnki.net

熟的云原生开发理念，提出了基于持续集成 / 持续部署的自动化敏捷软硬件系统开发流程（如图 6 右侧所示）。从 2018 年开始逐步上云，至 2023 年实现了实验过程中编码、开发、部署和调试的全面上云，实现教学实验项目的快速迭代，支持学生 7 × 24h 随时随地开展云上实验。通过期末调查可知，相比传统的本地手动模式（如图 6 左侧所示），“一键式”全自动模式的敏捷实验教学有效提升了学生的使用效率与课程满意度。这一流程可以显著降低学生开发环境构建时间，配置环境时间降低 92%，开发时间降低 48%，平台支撑的并发学生数量提高一倍。

图 6　基于云开发环境的全实验流程覆盖

6　教学效果

基于自研“思沃”实践云平台，国科大教学团队近 9 年累计培养计算机系统与芯片方向学生 900 余人，服务国科大计算机组成原理、操作系统、计算机体系结构 3 门本科生及研究生课程，2018—2023 年的 6 年间计算机组成原理累计选课 947 人次，持续推进学生“低开销全周期上云”开展软硬件实践项目。计算机组成原理课程期末考试结束后的学生匿名调查问卷显示，2021 及 2022 年云平台使用满意度分别为 67.3%（122 人反馈，统计方法为选择“非常满意”与“比较满意”的学生比例之和）和 52.7%（131 人反馈，统计方法同上）；而在 2023 年，因为首次采用了“一键式”全自动模式的敏捷实验教学方法，云平台满意度提升至 88.9%（108 人反馈，统计方法同上），体现了便捷的云上实践环境对于学生的吸引力。调查也显示出，部分学生认为现有云平台的响应速度不够及时，尤其 x86 容器集群生成比特流文件过慢，另外 FPGA 在云上的可调试手段不足，制约了实验评测速度，这些将作为科研团队下一步的技术探索方向。

经过实践云平台训练培养的学生多人次参加“龙芯杯”全国大学生计算机系统能力培养大赛（其中 2018 年获特等奖）与“一生一芯”计划，部分学生已成为“香山”开源处理器研发团队的核心骨干。此外，“思沃”实践云平台服务了 2021 年共青团中央组织的全国 76 所高校、116 支队伍参加的“逐梦杯”中国青年芯片科技创新大赛开源芯片设计赛道，并提供统一的线上测试平台。

7　结　语

在“101 计划”的引领及指导下，计算机组成原理教学团队以课程教学改革为抓手，在国科大及中国科学院计算技术研究所的支持下，组织教学团队与科研团队联合攻关，面向计算机系统类课程教学目标，探索了线上新形态教学模式，自研了“思沃”教学实践云平台，取得了较好的教学效果。教学团队与自研“思沃”云平台助力培养计算机系统及芯片方向拔尖人才，训练学生的专业本领，锤炼学生的品德修为。下一步，将继续提升优化实践云平台的响应速度和可调试

中国知网　https://www.cnki.net

性，丰富实验项目内容，满足“计算机系统与处理器芯片课程虚拟教研室”内更多的实践教学需求，使所培养的学生未来能够成为回馈社会、服务国家的有用之才。

参考文献：

[1] 习近平. 高举中国特色社会主义伟大旗帜 为全面建设社会主义现代化国家而团结奋斗: 在中国共产党第二十次全国代表大会上的报告[EB/OL]. (2022-10-16)[2023-10-17]. http://www.qstheory.cn/yaowen/2022-10/25/c_1129079926.htm.
[2] 郑纬民. 计算机专业大学生的系统能力培养[J]. 中国大学教学, 2021(5): 19-23.
[3] 蒋宗礼. 计算机类专业人才专业能力构成与培养[J]. 中国大学教学, 2011(10): 11-14.
[4] 袁春风, 王帅. 大学计算机专业教育应重视“系统观”培养[J]. 中国大学教学, 2013(12): 41-46.
[5] 高小鹏. 计算机专业系统能力培养的技术途径[J]. 中国大学教学, 2014(8): 53-57.
[6] 刘卫东, 张悠慧, 向勇, 等. 面向系统能力培养的计算机专业课程体系建设实践[J]. 中国大学教学, 2014(8): 48-52.
[7] 谭志虎, 秦磊华, 胡迪青. 面向系统能力培养的计算机专业实践教学模式[J]. 中国大学教学, 2017(9): 80-84.
[8] 张进. 计算机系统类课程改革实践[J]. 计算机教育, 2025(1): 14-16.
[9] 陆游游, 陈康, 李山山, 等. 计算机组成原理课程建设[J]. 计算机教育, 2024(5): 20-24.
[10] 袁春风, 余子濠, 朱光辉, 等. 计算机系统导论课程教学思路及课程资源建设[J]. 计算机教育, 2023(11): 12-17.
[11] Ruan J, Chang Y, Zhang K, et al. Increasing flexibility of cloud FPGA virtualization[C]//John McAllister, Roger Woods. 2022 32nd International Conference on Field-Programmable Logic and Applications (FPL). Belfast: IEEE, 2022: 350-357.
[12] Zhang K, Chang Y, Chen M, et al. Computer organization and design course with FPGA cloud[C]//Elizabeth Hawthorne, Manuel Pérez-Quiñones. Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE TS). New York: Association for Computing Machinery, 2019: 927-933.

（编辑：赵　原）

（上接第8页）

理、进程上下文、系统调用、操作系统的运行时环境、文件系统等，最终在 NEMU 上实现一个批处理系统。

分时多任务部分从多道程序的概念开始，通过相应的实验过程逐步引入上下文切换、进程的存储器映射、虚拟地址空间划分、页式虚拟存储管理、TLB、中断机制、内核栈、用户栈等，最终在 NEMU 中实现分时多任务系统。

参考文献：

[1] 袁春风, 朱光辉, 余子濠. 计算机系统: 基于x86+Linux平台[M]. 北京: 机械工业出版社, 2024.
[2] 袁春风, 余子濠, 陈璐. 计算机系统: 基于RISC-V+Linux平台[M]. 北京: 高等教育出版社, 2024.
[3] 袁春风, 余子濠. 计算机系统导论[M]. 北京: 机械工业出版社, 2023.
[4] 中国大学MOOC. 计算机系统基础(四): 编程与调试实践[EB/OL]. [2025-02-16]. http://www.icourse163.org/course/NJU-1449521162.
[5] 中国大学MOOC. 计算机系统基础(五): x86模拟器编程实践[EB/OL]. [2025-02-16]. https://www.icourse163.org/course/NJU-1464941173.
[6] 头歌实践教学平台. 数字逻辑与计算机组成实验[EB/OL]. [2025-02-16]. https://www.educoder.net/paths/wtg5fn6p.
[7] 头歌实践教学平台. 竞赛种子课程: 计算机系统导论[EB/OL]. [2025-02-16]. https://www.educoder.net/paths/fzjwo7kn.
[8] 计算机领域本科教育教学改革试点工作计划工作组. 高等学校计算机类专业人才培养战略研究报告暨核心课程体系[M]. 北京: 高等教育出版社, 2023.
[9] Introduction Gitbook. 南京大学计算机学院“计算机系统基础”课程PA实验讲义[EB/OL]. [2025-02-16]. https://nju-projectn.github.io/ics-pa-gitbook/ics2024/index.html.

（编辑：赵　原）

中国知网 https://www.cnki.net

Computer Organization and Design Course with FPGA Cloud

Ke Zhang, Yisong Chang, Mingyu Chen, Yungang Bao, Zhiwei Xu
State Key Laboratory of Computer Architecture, ICT, CAS
University of Chinese Academy of Sciences (UCAS)
Beijing, China
{zhangke,changyisong,cmy,baoyg,zxu}@ict.ac.cn

ABSTRACT

Computer Organization and Design (COD) is a fundamentally required early-stage undergraduate course in most computer science and engineering curricula. During the two sessions (lecture and project part) of one COD course, educational platforms play an important role in cultivating students' computational thinking, especially the ability of viewing the hardware and software in a computer system as a whole (computer system thinking ability for short in this paper). In order to improve teaching quality, in this paper, we discuss the deployment of an inexpensive in-house Field Programmable Gate Array (FPGA) cloud platform, which can provide students with hardware-software co-design methodology and practice. The platform includes 32 FPGA nodes and the scale can be dynamically changed. Each cloud node is heterogeneously composed of an ARM processor and a tightly-coupled reconfigurable fabric to provide students with hands-on hardware and software programming experiences. We illustrate our efforts to make the FPGA cloud as an easy-to-use resource pool to elastically support a class with 92 undergrads via Internet access and to monitor students' experimental behaviors. We also present key insights in our teaching activities that indicate such appliance is feasible to provide practice of both basic principles and emerging co-design techniques for students. We believe that our cost-effective FPGA cloud is of significant interests to educators looking forward to improving computer system-related courses.

CCS CONCEPTS

• **Applied computing → Education**; • **Computer systems organization**; • **Hardware → Reconfigurable logic and FPGAs**; **Hardware-software codesign**; • **Networks → Cloud computing**;

KEYWORDS

FPGA Cloud, Computer Organization and Design, Computer System Education, Hardware-Software Co-design

ACM Reference Format:
Ke Zhang, Yisong Chang, Mingyu Chen, Yungang Bao, Zhiwei Xu. 2019. Computer Organization and Design Course with FPGA Cloud. In *SIGCSE'19: 50th ACM Technical Symposium on Computer Science Education, February 27–March 2, 2019, Minneapolis, MN, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3287324.3287475

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGCSE'19, February 27–March 2, 2019, Minneapolis, MN, USA
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-5890-3/19/2…$15.00
https://doi.org/10.1145/3287324.3287475

1 INTRODUCTION

Computer Organization and Design (COD) is an early-taught undergraduate course that typically covers fundamental CPU design principles, instruction set architecture (ISA), assembly language and input/output interfacing in both lecture and project sessions. As a key component in ACM computer engineering curriculum [1], COD is essentially required to teach students the concepts of hardware and software as well as the synergy of both parts in a computer system. Moreover, COD gradually attracts attention in computer science undergraduate programs (e.g., ACM's 2013 curriculum guidelines [2]) as either a required course or an elective to show an abstraction of hardware from a programmer's perspective.

In the COD lab course, students are required to fully understand and carefully apply abstract concepts they learnt in lectures to practicing ordinary low-level software programming in assembly and C language or even advanced hardware logic design. Therefore, compared with other traditional lecture-centric courses, the COD course largely depends on educational platforms to provide real and interactive engineering environments for students to conduct hands-on experiments. As a result, it is crucial for educators and instructors to elaborately design and massively use educational platforms in both lecture and project sessions of a COD course in order to cultivate students' computational thinking [34], especially ability of viewing the hardware and software in a computer system as a whole (*computer system thinking ability* for short).

A) Using new hardware in COD platforms

The evolution of modern computer systems has been widely inspired by emerging heterogeneous acceleration and domain-specific architecture [14]. This trend emphasizes the key importance of *hardware-software co-design* to all participants in the field of computer system [3]. Therefore, besides basic software programming, students are strongly recommended to study hardware design principles and the co-design concepts as early as possible in computer system-related courses and especially in the early-stage COD class, improving their *computer system thinking ability*. Given the state-of-the-art heterogeneous computing systems, we argue that the key challenge to further improve teaching effects by manifesting system-level *hardware-software co-design* in COD courses lies in the exploitation of an appropriate educational platform that provides heterogeneous co-design capability for students.

In traditional COD courses, software ISA simulators [7, 17, 22, 32] are widely used as experimental environments for students' writing

[1]First authors Zhang and Chang have made equal contributions to this work.

and executing low-level programs. However, such virtual platform not only inherently restricts execution time and flexibility, but also exhibits an incomplete system view to students due to the shortage of real hardware components. The avant-grade single-board computers based on System-on-Chip (SoC) [28, 31] provide software execution environment on real hardware platforms but lack capability to integrate additional hardware accelerators. Similarly, despite suitability for hardware-related experiments, Field Programmable Gate Array (FPGA)-based development boards [9, 26] fail to support complicated software development.

Fortunately, a heterogeneous FPGA (e.g., Xilinx's Zynq UltraScale+ MPSoC [36]) which has been just commercially available can overcome all the above-mentioned shortcomings. This hybrid device tightly couples a high-performance, software-programmable hard core SoC (commonly based on ARM processor) with a reconfigurable logic fabric on the same chip die. Such configuration has been confirmed in setting up a feasible experimental platform for embedded system-related courses [25, 33]. We believe that heterogeneous FPGAs are more appropriate and cost-effective as educational platforms to demonstrate the concept of hardware-software co-design and deliver related experimental environment in COD courses.

B) Introducing cloud computing to COD platforms

Instructors of COD courses used to hand out each student one hardware experimental kit that would be connected to his or her laptop during projects. Due to different progresses of students' experimental projects, 1) status of dozens of kits are diverse, and 2) the number of simultaneously in-use kits is frequently changed. These two factors lead to a low utilization rate of the experimental kits. In addition, each student conducts projects privately in the laptop with an attached hardware kit, making his or her experimental behaviors entirely invisible to instructors. Motivated by emerging cloud computing technology, we believe that these situations would be improved if we could make experimental kits as a pool of remotely accessed resources that can be unilaterally provisioned and automatically monitored. Some preliminary cloud-based experimental platforms [10, 24, 37, 38] were established to provide remotely accessed development kits for students.

C) Why not commercial public FPGA cloud?

Cloud vendors have provided FPGAs as public acceleration resources for users to rent (e.g., Amazon's AWS F1 instance [4]). From our perspective, such public FPGA cloud exhibits several drawbacks in COD teaching:

1) Expensive economic costs to deploy the same number of FPGAs as in our in-house FPGA cloud within a semester. Note that the purpose of public FPGA cloud is mostly for accelerating web services, so the FPGA devices used in public cloud are always high-end and costly products,

2) Complicated FPGA development and configuration flow in public cloud that requires students to learn additional skills non-related to the COD course,

3) The execution environment of FPGA cloud is mainly based on virtual machines (VMs) running on the remote server, introducing unpredictable penalties and inter-VM contentions for various users.

D) Putting it all together

In this paper, we propose an in-house heterogeneous FPGA-enabled cloud platform with multiple nodes (32 for our current implementation) connected via Gigabit Ethernet for teaching COD courses in both lecture and project. Each node is an inexpensive custom circuit board based on Xilinx Zynq UltraScale+ MPSoC which provides both hardware and software programmability in a single board for students with the cost of only $950. We make each physical node with a fully-fledged Linux kernel and related environment atop the ARM SoC as a general cloud resource in which students can configure FPGA fabric, interact with logic modules in FPGA and launch software applications. All available resources are managed and scheduled via Ethernet by a commercial-grade cloud computing framework on a front-end x86 server. A pre-configured user-end VM running on each student's laptop provides standard toolchains of both hardware and software development. Unlike the server-end VMs for execution in public FPGA cloud, in such user-end VM, students use a simple script-based design flow to compile their FPGA designs and software programs offline, request an experimental cloud resource, and transparently upload the generated FPGA configuration files as well as executable binaries to the requested resource for execution via campus network. In order to fully achieve teaching objectives in both computer science and engineering curricula and comprehensively demonstrate the concepts of *hardware-software co-design* on our FPGA cloud, we propose an exploratory assignment on ISAs and a live demo about details of network packets processing in our COD lectures, as well as experimental projects that cover both fundamental principles and emerging techniques. Some key advantages our FPGA cloud provides for teaching COD course include:

Ease of use and maintenances. The cloud platform allows students to conduct projects as using local experimental kits without resource contention between each other, making it possible to carry out performance-related studies on remote hardware. Moreover, our FPGA cloud platform can be simply managed by teaching assistants of the COD course, without additional engineering efforts.

Elasticity and low cost. Resource pooling dynamically adjusts the number of available resources as per students' demands to afford a class of 92 students to conduct projects within the scope of only 32 nodes, significantly saving the infrastructural budget.

Teaching Assistance. Detailed execution behaviors of each student are monitored and logged in the cloud. Such information indicates learning outcomes of one student, helping instructors make individual teaching policies for different students.

2 BACKGROUND

Conventional COD courses mainly concentrate on how to teach students hardware abstraction from a software programmer's point of view, thus emphasizing the importance of ISAs and relevant assembly programming in experimental projects. With the minimum economic cost and educators' efforts, the software ISA simulator is a good choice for students to use, which is even leveraged today in some representative COD-like courses [29, 30]. As the rise of inexpensive hardware single-board computers such as Raspberry Pi, some educators have asked their students to conduct experiments on such platforms [21, 27].

On the other hand, emphasizing hardware design in COD course offers a new bottom-up perspective to students to fully understand computer systems instead of pure ISAs. In experimental projects of such class, students are required to implement their preliminary

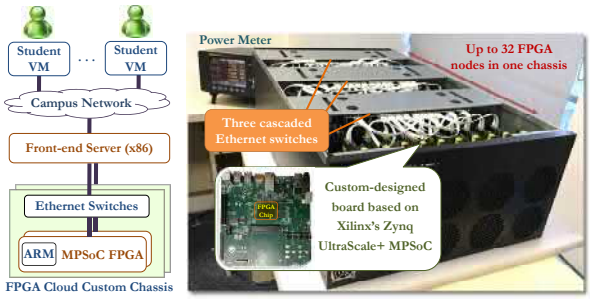


Figure 1: Our proposed FPGA cloud architecture and one prototype chassis with 32 individual physical FPGA nodes that is large enough to support our COD class with 92 students.

CPU cores using a hardware description language (HDL) such as Verilog HDL. However, these CPU cores are always verified just in HDL behavioral simulation and logic synthesis rather than implemented to run on real hardware [18, 29, 30]. Although real-world FPGAs are involved in some classes [5, 6, 8, 11, 13, 15, 16, 23], it is still extremely difficult for early undergrads to independently afford the task of organizing a minimum runnable system that wraps their CPU cores to various I/O components (e.g., memory, UART, etc.) with standard interfaces. This task always requires a significant amount of FPGA-specific knowledge and sufficient hardware design experiences that are out of the range of COD course.

In summary, existing experimental platforms in COD-related courses rarely offer students a real-world executable system-level environment with an easy-to-use interface to help them fully understand computer systems.

3 FPGA CLOUD IN COD COURSE

In this section, we introduce our proposed FPGA cloud architecture as well as one prototype chassis, and related methods to provide an easy-to-use interface for students. Based on the FPGA cloud, we discuss improvements in both lectures and projects of our COD course to emphasize *hardware-software co-design*.

3.1 Architectural and Technical Overview

Figure. 1 depicts our proposed FPGA cloud architecture and one in-house chassis in which at maximum 32 nodes based on custom circuit boards using Xilinx's Zynq UltraScale+ MPSoC FPGAs are interconnected via three cascaded Ethernet switches. More chassis can be cascaded for larger class. The design objective of our FPGA cloud is to effectively abstract platform details and agilely help students to leverage the FPGA cloud in campus via an user-friendly Ethernet-accessible interface, avoiding unnecessary efforts that negate students' concentration on the COD course.

First, we leverage and modify a commercial-grade cloud resource management framework—OpenStack [20] to provision each physical node in the custom chassis as a general cloud resource. As shown in the left part of Figure. 2, the modified OpenStack controller runs on the front-end x86 server to communicate with hardware control planes in FPGA chassis and provides APIs to allow students to request and release resources on-demand using either webpage or command line via Ethernet. OpenStack maintains a database of

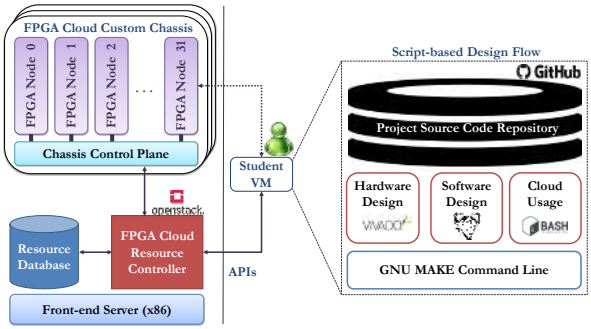


Figure 2: Efforts to provide an easy-to-use FPGA cloud service. An OpenStack-enabled resource manager (left part) allows students' remote access to the allocated FPGA node (arrowed dash line). A scripted design flow on each student's VM (right part) simplifies FPGA development and usage.

information of all physical nodes in the FPGA cloud, both in use and available for elastic allocation and release.

Second, in order to remove numerous FPGA design complexities for students, we implement a bunch of pre-built scripts that runs inside the commercial FPGA design tool (i.e., Xilinx's Vivado [35] in our current COD course) in the VM to automatically setup an FPGA project with static logic of complex I/O components surrounding students' synthesizable HDL designs and to strictly execute key design flow steps, making students purely concentrate on their HDL design that are strongly related to the contents of COD course. We further spread such script-based design flow via standard GNU `MAKE` command line to uniformly cover FPGA hardware development, software compilation and FPGA cloud usage (right part of Figure. 2). By this means, complicated experimental procedures are abstracted into one single and simple `MAKE` command, saving a large amount of efforts for students to carry out projects via the FPGA cloud. Moreover, all relevant source files and scripts in an experimental project are organized into a repository hosted on GitHub Classroom [12]. Therefore, all students' behaviors are committed in the repository to trace variations of one student's ability in different phase of COD course and make just-in-time guidance.

3.2 Usage in Lectures

Rigid oral explanations in conventional COD lectures leave students far away from real systems, making it difficult to understand some knowledge units that requires hardware-software collaboration. Although hands-on experiments relieve such situation to some extent, they are still unable to cover all relevant key points mentioned in the lecture class. In order to fill such gap, we attempt to propose an exploratory assignment as well as a live demo based on our FPGA cloud as supplementaries in lectures of our COD course.

Exploratory Assignment. As a fundamental interface between hardware and software, ISA is a strongly required knowledge unit in COD courses. This point is especially important to computer science students to understand hardware and computer system, as they have rare opportunities to engage in hardware-related courses.

Consequently, we push students to make a comprehensive comparison among ISAs of x86, ARM, MIPS and the emerging RISC-V

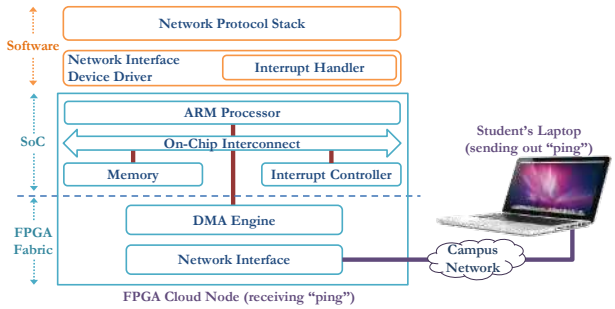


Figure 3: Full system stack on an FPGA cloud node to demonstrate system-level details when processing a "ping" network packet as a live demo in lectures of our COD course.

in basic assembly blocks, segmentations in binary object and executable files, endianness, application binary interface (ABI) and stack management based on practical compilation as well as analysis of binary files. Students use the standard `GCC` compilers and related `binutils` for x86 and ARM ISAs in the operating systems on each student's VM and the requested cloud resource respectively, as well as `GCC` cross-compiling environments for MIPS and RISC-V pre-deployed in the VMs. We provide a series of standard micro-benchmarks written in C language and encourage students to custom-design some other programs for comparison.

Live Demo. Advanced input/output interfacing, mainly including direct memory access (DMA) and interrupt, is an essential system concept that deeply relies on hardware-software collaboration. Due to shortage in real application scenario, it is very difficult for students to understand this key concept just according to descriptions in COD textbook. Moreover, such knowledge unit is rarely involved in a COD experimental project since students are required to learn a huge amount of additional knowledge out of the range of COD course. As a result, in our lecture in COD course, we build a live demo about details of network packet processing on real-world hardware-software full system stack inside our FPGA cloud platform, which is interactively visible to students for observation of system-level I/O processing behaviors. To make the demo easily to understand, we select `ping`, a common networking utility students are familiar to use in daily life as an illustrative example. We also abstractly introduce some easy-to-understand concepts of network protocol stack and device driver in our COD course in advance.

Figure. 3 briefly describes the full system stack in one FPGA cloud node and the hardware network processing path in a student's laptop with the VM. For students' hardware visibility, we implement a network interface and its related DMA engine in the FPGA fabric of each physical cloud node. System-level information of this newly-involved network interface is configured during resource allocation. Meanwhile, students are enabled to observe operations in ARM-end operating system via traces logged by pre-modifications in kernel-level network device driver and protocol stack.

After obtaining the IP address of an allocated FPGA cloud resource, one student executes a GNU `MAKE` command in the VM to launch the FPGA development tool GUI that automatically prepares the environment for remote signal probing of key hardware ports from network interface and DMA engine. The student then sends

Table 1: FPGA cloud-inspired COD lab projects

#	Project name and brief description
1	**Basis of Assembly Language:** Function call routine and stack management
2	**Basic CPU Design:** Register file, ALU and single-cycle CPU
3	**Advanced CPU Design:** Multi-cycle CPU with interface to DDR memory and UART
4	**Performance Evaluation:** Hardware performance counter design
5	**Memory Hierarchy (Optional):** Instruction and data cache
6	**Domain-Specific Architecture (Optional):** Custom deep convolutional neural network accelerator with DMA interface

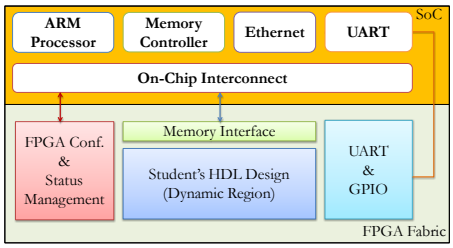


Figure 4: A template architecture in FPGA cloud node to facilitate easy integration of students' HDL design into a minimum runnable system.

out a "ping" packet to the obtained IP address. When the "ping" packet is injected into the network interface of the cloud node, variations of hardware signals are displayed in the GUI in time, allowing students to observe packet movements in hardware data path. Traces in the operating system allow students to clearly understand life cycles of network packets and motivate students to obtain insights in the approaches of hardware-software interactions.

3.3 Usage in Labs

Based on our FPGA cloud, we redesign the projects in our COD labs as shown in Table. 1. The main target is to push students to make their outcomes in these projects executable in an actual hardware-software environment, instead of a pure simulation.

In the first experimental project, students are disciplined in software programming abilities within hybrid assembly and C language on the ARM processors of cloud nodes, which emphasizes the most fundamental concepts of function call routine and runtime stack management. In this manner, students are driven to deeply understand the hardware interface from a programmer's perspective. If educators would like to extend such project arrangement with more software-oriented experiments, existing projects based on conventional platforms like Raspberry Pi or x86 are also inherently supported in our FPGA cloud without modifications to run atop the high-performance ARM processor. Moreover, exclusive occupancy of one physical cloud node guarantees students to conduct further software performance studies.

In the following projects, students are guided to implement synthesizable hardware components in the FPGA fabrics and make it executable with software collaborations. In order to simplify students' design efforts in such hardware-centric projects, we propose a template architecture in our FPGA cloud node as shown in Figure. 4. Based on such template, students are encouraged to

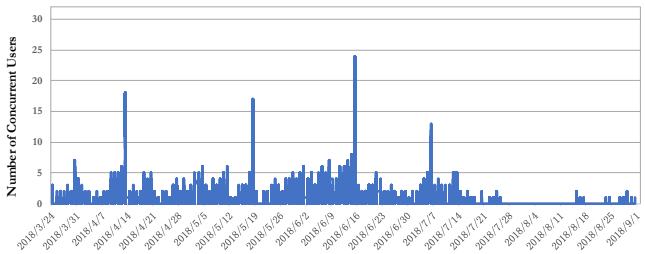


Figure 5: The number of in-use nodes in the FPGA cloud measured every 20 seconds during the 2018 Spring semester in which our COD course carried on, and short Summer semester to conduct students' research projects, respectively. The X-axis is cataloged into each day during both semester.

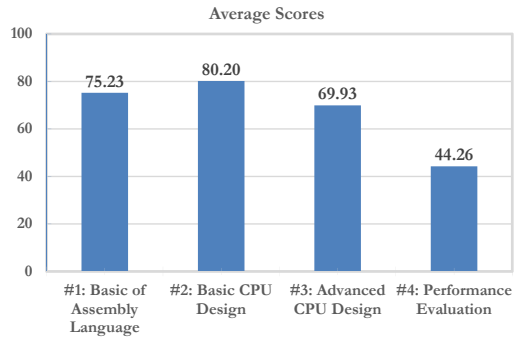


Figure 6: Average score (100-point grading system) of each required project in our class as declared in Table. 1.

implement custom CPU cores in the dynamic region based on either conventional MIPS or emerging RISC-V ISA to practice fundamental CPU design principles. Custom cores share the same address space of DDR4 memory populated on the FPGA circuit board with the ARM processor. In this manner, ARM processor is leveraged as an auxiliary engine that is easily driven by students in their VM-based development environment, taking charge of FPGA configuration, data and program loading, UART communication, status management and hardware debugging of students' cores. Undergrads are also required to implement a C language-based software framework running atop custom CPU cores to allow applications access memory-mapped hardware UART and GPIO interfaces.

Students are cultivated with the quantitative approach via a project towards design of hardware performance counters. Based on performance evaluation results, students optionally optimize the memory hierarchy of their CPU cores via instruction and data cache. Students also electively involve in design of deep convolutional neural network accelerators in the dynamic region of template architecture as well as implement an ARM-end full software stack incorporating with accelerators. By this means, students acquire preliminary skills in heterogeneous acceleration and domain-specific architecture design.

After disciplined with these experimental projects in a full semester, outstanding students in our COD course afford to further conduct research projects independently on our FPGA cloud in the short terms during summer vacations, some of which were highlighted in one tutorial [19] we held in ISCA 2018 (one of the flagship annual academia conferences in computer architecture).

4 TEACHING PRACTICE AND DISCUSSION

In this section, we mainly discuss some key observations we acquire from teaching practice in our COD course with the FPGA cloud in the Spring and Summer semester of 2018 to illustrate the efficiency and feasibility of our FPGA cloud as an easy-to-use infrastructure in motivating students to study computer system-related courses.

The number of concurrent users (Y-axis) monitored in our cloud appliance is chronologically plotted in Figure. 5. We used Linux shell scripts persistently running on the front-end server with our OpenStack-based resource controller to acquire how many students were using this platform every 20 seconds and log the data for the whole 2018 Spring and Summer semester (160 days, 690K log entries in total). In this figure, several spikes can be recognized around 2018/3/31, 4/14, 5/19, 6/16 and 7/7, which are due dates for each project. The maximum number in this plot is 24, which means 24 FPGAs in the cloud were simultaneously occupied and utilized by 24 students. Therefore, in the case of a COD class with 92 students, one 32-node FPGA cloud chassis can meet our course requirement. One more thing to be noticed is that several single-digit numbers were recorded from 8/11 to 8/31 in the Summer semester. The reason is that after the COD course, three outstanding students volunteered to join our research group and were actively involved in one of our research projects. Therefore, utilizing the FPGA cloud to teach in our computer system course can successfully help some early undergrads engage in computer system-related research.

Figure. 6 shows the average score of each required experimental project in the class of 2018 Spring semester. With the help of our easy-to-use design flow and user interface of the FPGA cloud, all experimental projects are feasibly carried out. In the beginning of this course, students started to be acquainted with this cloud platform in Project #1 and #2 that target programming in assembly as well as basic CPU design respectively and got good scores. In Project #3, students' average grade decreased as the integration of their custom CPUs into real-world environment increases difficulties in CPU design and HDL implementation compared with the previous projects. This situation was even worse in Project #4 since more sophisticated benchmarks were involved for performance evaluation exposing more design defects in students' CPU design and implementations. Another reason causes the lowest average grade in Project #4 lies in that existing debugging environment in our FPGA cloud is not efficient enough for students to fix complex bugs of their CPU cores in a four-week period. Moreover, grades of optional projects are ignored as there were few students to engage in such projects as important dates of these two projects were much close to final exams of other courses.

We asked our students to complete two surveys respectively in the middle and end of spring semester. The survey results from Table. 2 show that the students' satisfaction with the usage of remote FPGA cloud platform (72/82=87.80%) was much higher than using FPGA cards located in our computer lab (22/82=26.83%). As demonstrated in Figure. 7, most of students preferred to use cloud rather than local hardware kits due to the ease-of-use of our FPGA cloud

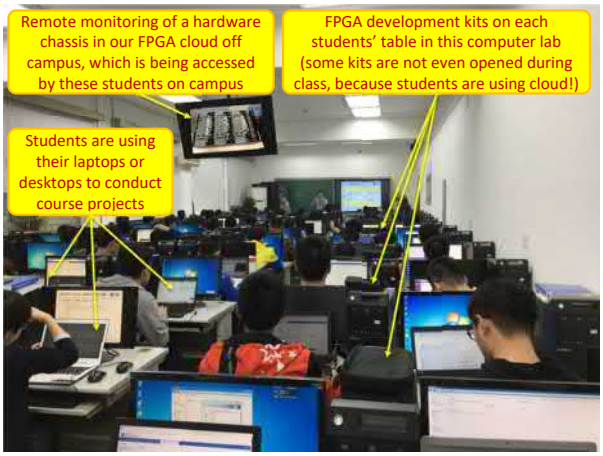


Figure 7: An instructor and a teaching assistant are leading the 92 students in our COD course to conduct experimental projects in a computer lab.

Table 2: Survey results of satisfaction to the FPGA cloud in our COD course. Note that 10 students who are minor in computer science are not required to conduct lab projects.

# of students	Remote FPGA Cloud Platform	Local FPGA Development Kits
who were fond of using	72	22
who never used	2	19
who completed lab projects	82	82
in the class (lectures + labs)	92	92

Table 3: Student engagement in three academic fields represented by student counts, which shows that how the degree of learners' interest in these areas changes during the semester ("+": favor, "++": strong interest, "−": dislike).

	Computer Science Major	Computer Architecture and System	Research in Computer Science
+ → ++	21	24	16
++ → ++	51	18	36
− → +	8	32	16
+ → −	10	10	11
− → −	2	8	13

platform. In terms of teaching effectiveness for both lecture and lab part of our COD course, as shown in Table. 3 we can notice that 55.43% (51/92) of the students have been sustainably interested in the major of computer science and 45.66% ((24+18)/92) for computer architecture & system throughout this course. More importantly, by leveraging the FPGA cloud platform in both lectures and labs, our teaching approach gradually changes additional 34.78% (32/92) students' attitude towards computer system-related knowledge from dislike to favor. Therefore, over 80% sophomore undergrads would like to take more computer system-related courses in the following semesters. Last but not least, in the third column of this table, an increasing number (16+16) of these early undergraduates are willing to conduct computer system-related research in the future and 3 people joined our research lab during this summer. As instructors, we should also pay attention to the 10 or 11 students who became unfriendly to CS in the end of our course. One student replied "Coding in HDL was terrible and hardware debugging was also tedious". Although this phenomenon exists for certain learners, we need to mitigate this side effect in following terms in order to improve the teaching effectiveness for all students. There are several possible solutions, such as adding HDL introduction and practice in the beginning or before the class, improving our FPGA cloud with an advanced debugging framework, and so on.

5 LIMITATIONS AND FUTURE WORK

Debugging and Simulation. As discussed in Section 4, debugging FPGA design in a remote manner currently exhibits inefficiency, especially for hardware-software co-debugging to find design defects of a custom CPU core. We plan to further boost our simulation framework which would be nearly the same as the hardware-software execution environment in our cloud FPGA node to allow students conduct local co-debugging in their VMs.

Project Schedule. We would like to rearrange the time slot of each project and coordinate with other courses to meet learning diversity of various students, making each of them possible to participate in all experimental projects.

Grading Method. Currently, the FPGA cloud platform can only return a result log file to each student after he or she completes a test case online. Therefore, the student could not see his or her score for this project instantly. Moreover, this also increases instructors' and teaching assistants' workloads due to the tedious and manual updates of project scores for each student. To tackle this problem, we would like to integrate an advanced auto-grading feature in our next-version FPGA cloud.

6 CONCLUSIONS

In this paper, we describe an in-house FPGA cloud platform for a COD course which can be effectively used to introduce the essential concept of hardware-software co-design for next-generation computer system design to early undergrads. Our proposed FPGA cloud provides an easy-to-use interface as well as a simple script-based design flow to cultivate students' *computer system thinking ability* with the minimum additional cost. We believe that our FPGA cloud and relevant ecosystem would be also appropriate to service students and instructors in many other computer system-related courses such as operating system, computer architecture, embedded system design, and so on.

7 ACKNOWLEDGMENTS

This work is supported by the National Key Technologies Research and Development Program of China under Grant No. 2017YFB1001602, by the National Natural Science Foundation of China under Grant No. 61702485, 61420106013, 61521092 and 61702480, and by the Youth Innovation Promotion Association of CAS under Grant No. 2017143 and 2013073. We appreciate Prof. Xiufeng Sui for his teaching in one of the two lecture classes of COD course at UCAS. We thank Zihao Yu, Bowen Huang, Sha Liu, Huizhe Wang, Xu Zhang, Lei Yu, Ran Zhao and Zhiwei Lai for their teaching and engineering assistance in our COD lab projects. We also thank anonymous reviewers' valuable feedbacks.

REFERENCES

[1] ACM. 2004. Computer Engineering Curricula 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/ce-final-report.pdf.

[2] ACM. 2013. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf.

[3] ACM. 2016. Computer Engineering Curricula 2016: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. https://www.acm.org/binaries/content/assets/education/ce2016_web_final.pdf.

[4] Amazon Web Services. 2017. Amazon EC2 F1 Instances. https://aws.amazon.com/ec2/instance-types/f1/.

[5] Arizona State University. 2018. Progressive Learning Platform. https://plp.asu.edu/.

[6] D. Baldwin, P. Sanderson, R. McCartney, S. Ludi, N. T. Ramachandran, and C. Taylor. 2011. SIGCSE Special Project Showcase. In *Proc. 42nd SIGCSE Technical Symposium on Computer Science Education*. 5–6.

[7] Bochs. 2018. The Cross Platform IA-32 Emulator. https://bochs.sourceforge.net/.

[8] E. Brunvand. 2011. Games as Motivation in Computer Design Courses: I/O is the Key. In *Proc. 42nd SIGCSE Technical Symposium on Computer Science Education*. 33–38.

[9] P. Bulic, V. Gustin, D. Sonc, and A. Strancar. 2013. An FPGA-based Integrated Environment for Computer Architecture. *Computer Applications in Engineering Education* 21, 1 (March 2013), 26–35.

[10] Z. Chai, Z. Wang, W. Yang, S. Ding, and Y. Zhang. 2014. OpenHEC: A Framework for Application Programmers to Design FPGA-based Systems. In *Proc. 1st International Workshop on FPGAs for Software Programmers*. 59–64.

[11] M. L. Corliss and M. Melara. 2011. VIREOS: An Integrated, Bottom-up, Educational Operating Systems Project with FPGA Support. In *Proc. 42nd SIGCSE Technical Symposium on Computer Science Education*. 39–44.

[12] Github. 2018. Github Classroom: Your Course Assignments on GitHub. https://classroom.github.com/.

[13] M. Gschwind. 1994. Reprogrammable Hardware for Educational Purpose. In *Proc. 25th SIGCSE Technical Symposium on Computer Science Education*. 183–187.

[14] J. Hennessy and D. Patterson. 2018. A New Golden Age for Computer Architecture: Domain-Specific Hardware/Software Co-Design, Enanced Security, Open Instruction Sets, and Agile Chip Development. In *Turing Lecture of ACM/IEEE International Symposium on Computer Architecture (ISCA)*.

[15] M. J. Jipping, S. Henry, K. Ludewig, and L. Tableman. 2006. How To Integrate FPGAs Into a Computer Organization Course. In *Proc. 37th SIGCSE Technical Symposium on Computer Science Education*. 234–238.

[16] D. B. Larkins, W. M. Jones, and H. E. Rickard. 2013. Using FPGAs as a Reconfigurable Teaching Tool Throughout CS Systems Curriculum. In *Proc. 44th SIGCSE Technical Symposium on Computer Science Education*. 397–402.

[17] LC3Help.com. 2018. LC-3 Simulator. http://www.lc3help.com/.

[18] MIT Computation Structure Group. 2017. 6.175: Constructive Computer Architecture. https://csg.csail.mit.edu/6.175/.

[19] ICT of CAS. 2018. The Case for Labeled Von Noeumann Architecture (LvNA). https://sdc.ict.ac.cn/isca2018-tutorial/.

[20] OpenStack. 2018. OpenStack. http://www.openstack.com/.

[21] R. Ord. 2018. CSE 30: Computer Organization and System Programming. https://cseweb.ucsd.edu/~ricko/CSE30/.

[22] D. Patti, A. Spadaccini, M. Palesi, F. Fazzino, and V. Catania. 2012. Supporting Undergraduate Computer Architecture Students Using a Visual MIPS64 CPU Simulator. *IEEE Transactions on Education* 55, 3 (Aug. 2012), 406–411.

[23] Penn Engineering. 2016. Computer Organization (Course: ESE534). https://www.seas.upenn.edu/~ese534/.

[24] C. Quan, Y. Chen, S. Li, and Y. Zhao. 2016. Exploration of the Computer Hardware Experiment Teaching Method based on the Cloud Platform. In *Proc. IEEE Frontiers in Education Conference (FIE)*. 1–5.

[25] D. Roggow, P. Uhing, P. Jones, and J. Zambreno. 2015. A Project-based Embedded Systems Design Course Using a Reconfigurable SoC Platform. In *Proc. IEEE International Conference on Microelectronics Systems Education (MSE)*. 9–12.

[26] S. Sohoni, S. D. Craig, and S. Lu. 2017. Impact of Prior Exposure to the PLP Instruction Set Architecture in a Computer Architecture Course. In *Proc. 48th SIGCSE Technical Symposium on Computer Science Education*. 555–560.

[27] Stanford University. 2017. CS107E Computer Systems From the Ground Up. https://web.stanford.edu/class/cs107e/.

[28] D. Tarnoff. 2015. Integrating the ARM-based Raspberry Pi into an Architectural Course. *Journal of Computing Sciences in Colleges* 30, 5 (May 2015), 67–73.

[29] UC Berkeley EECS. 2018. CS61C: Machine Structures. https://inst.eecs.berkeley.edu/~cs617.

[30] UIUC ECE. 2018. ECE 411: Computer Organization and Design. https://courses.engr.illinois.edu/ece411/fa2018/.

[31] E. Upton, J. Duntemann, R. Roters, T. Mamtora, and B. Everard. 2016. *Learning Computer Architecture with Raspberry Pi* (1st ed.). Wiley Publishing.

[32] K. Vollmar and P. Sanderson. 2006. MARS: An Education-Oriented MIPS Assembly Language Simulator. In *Proc. 37th SIGCSE Technical Symposium on Computer Science Education*. 239–243.

[33] K. L. Wang, C. S. Cole, T. Wang, and J. Harris. 2013. An Effective Project-based Embedded System Design Teaching Method. In *the 120th American Society for Engineering Education (ASEE) Annual Conference and Exposition*. 6849:1–6849:9.

[34] J. M. Wing. 2006. Computational Thinking. *Communication of the ACM* 49, 3 (March 2006), 33–35.

[35] Xilinx. 2018. Vivado Design Suite — HLx Editions. https://www.xilinx.com/products/design-tools/vivado.html/.

[36] Xilinx. 2018. Zynq UltraScale+ MPSoC. https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html/.

[37] K. Zhang, M. Chen, and Y Bao. 2018. ZyForce: An FPGA-based Cloud Platform for Experimental Curriculum of Computer System in University of Chinese Academy of Sciences (Abstract Only). In *Proc. 49th SIGCSE Technical Symposium on Computer Science Education*. 1091.

[38] Y. Zhang, Y. Chen, X. Ma, Y. Tang, Y. Niu, S. Li, and W. Liu. 2017. Remote FPGA Lab Platform for Computer System Curriculum. In *Proc. ACM Turing 50th Celebration Conference - China (TUR-C)*. 3:1–3:6.

Teaching Computer System Courses with an Online Large-Scale Method

Ke Zhang[1,2,4], Congwu Zhang[1,2], Yisong Chang[1,2,4], Yungang Bao[1,2,4], Mingyu Chen[1,2,4], Longbing Zhang[1,2], Zhiwei Xu[1,2], Jonathan Gratch[3], Jian Zhang[3], Ninghui Sun[1,2]
[1] SKLCA, Institute of Computing Technology of Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Texas Woman's University, Denton, USA
[4] Peng Cheng Laboratory, Shenzhen, China
{zhangke,zhangcongwu,changyisong,baoyg,cmy,lbzhang,zxu,snh}@ict.ac.cn, {jgratch,jzhang}@twu.edu

2021 IEEE International Conference on Engineering, Technology & Education (TALE) | 978-1-6654-3687-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/TALE52509.2021.9678730

***Abstract*—Domain-Specific Architectures (DSAs) and hardware-software co-design are greatly emphasized in the CS community, which demands a significant number of participants with Computer System (CSys) capabilities and skills. Conventional CSys courses in a lecture-lab format are limited in physical resources and inherently difficult to cultivate talents at a large scale. Online teaching is a potential alternative to instantly enlarge the face-to-face class size. Unfortunately, simply putting the lecture contents in CSys courses online lacks 1) personal attention, 2) learner-instructor interactions, and 3) real-hardware experimental environments.**

To tackle the above challenges, we introduce a four phase online CSys course program and the related teaching methods for a cloud-based teaching platform. The four-phase course program included two basic/required stages and two advanced/optional stages to promote students' knowledge and skill level with appropriate personal attention. We studied if online interaction methods, such as in-class chat and one-on-one online grading interview, can strengthen the connections between teachers and students in both lectures and labs. We created a heterogeneous cloud platform to enable students nationwide to reliably conduct labs or projects on remote programmable hardware. We believe that our proposed course design methodology is beneficial to other CS courses in the post-COVID-19-era.

***Index Terms*—FPGA Cloud, Computer Education, Computer Organization Design Course**

I. Introduction

The end of Moore's Law and Dennard Scaling implies that traditional computer systems with general-purpose processors are unable to keep up with the rapid demands of emerging and prevalent application scenarios, such as Artificial Intelligence (AI) acceleration, big data processing, etc. Fortunately, an effective approach named domain-specific architectures (DSAs) [1] or specialized computing [2] was recently proposed by computer scientists and adopted by engineers to coordinate software algorithms and hardware components inside computing systems aiming for different purposes and performance improvements. This trend, therefore, requires a large amount of CS talent [3] with the experience in software-hardware co-design and computer system thinking ability, which should become new teaching objectives in future computing education, especially in the Computer System (CSys) courses.

However, a traditional offline CSys course with interwoven lectures and lab sessions is physically limited in terms of both classroom space and experimental resources, and thus is not suitable for large-scale education. For example, a physical classroom can only accommodate a limited number of students. *Parallel* class mode could expand learners but yet demands more instructors and teaching assistants (TAs). Also, students need a laboratory to conduct hands-on projects using computers, hardware development kits, or other hardware debugging equipment.

Compared to physical classrooms, virtual cyberspace can accommodate almost unlimited students in terms of numbers, geography, Internet access and sufficient network bandwidth. Moreover, the scalable cloud computing technology and infrastructure can simultaneously support large-scale users at an affordable cost. More urgently, taking the COVID-19 situation in consideration, migration of conventional face-to-face CSys curricula to a fully online mode becomes critical for universities and colleges to continue CSys education uninterrupted. Although remote learning is currently an emergency measure, we believe that a successful online education in every discipline including CSys, AI, and other CS fields would be of vital importance even when the pandemic is over.

Unfortunately, moving face-to-face CSys classes to the online format without a careful course design process would bring the following challenges in its delivery:

Inappropriate selection of online teaching format. Online courses are usually divided into synchronous and asynchronous formats. In the asynchronous format (e.g., MOOC course), teachers record course videos that are flexible and easy to watch online for students. However, it may be difficult to monitor learning progress of various students. Moreover, there are insufficient online real-time interaction methods between teachers and students in asynchronous courses [4]. Meanwhile, in the synchronized online courses, each student is required to log in and participate in class at a fixed time, making their learning status visible to teachers. Real-time discussions with instructors and classmates in synchronous courses are made possible when network quality could be

978-1-6654-3687-8/21/$31.00 ©2021 IEEE

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on January 26,2022 at 01:36:06 UTC from IEEE Xplore. Restrictions apply.

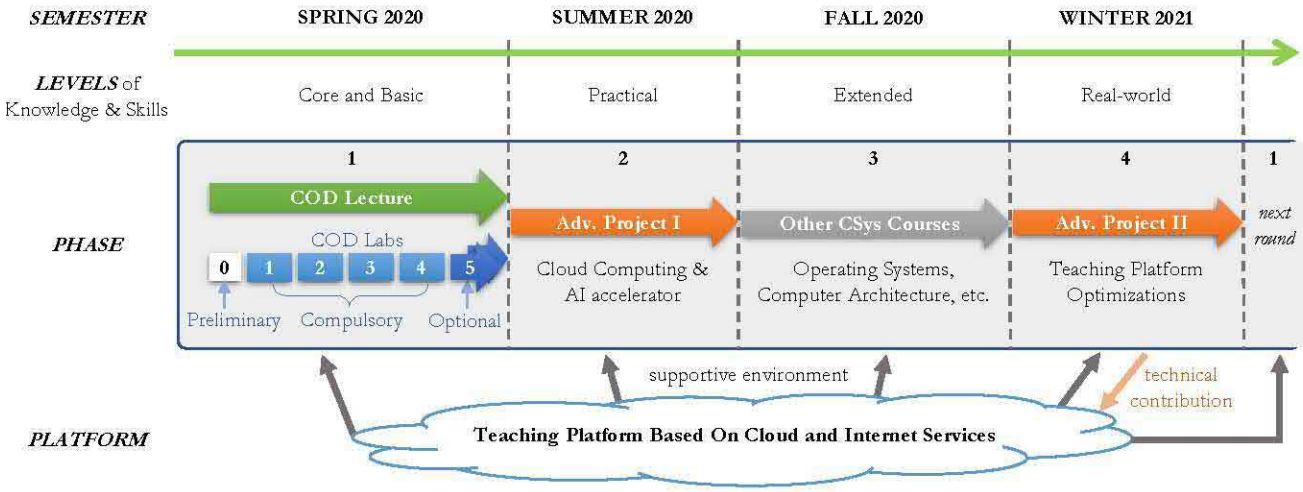


Fig. 1. Four-phase course design based on a customized teaching platform.

guaranteed. It is challenging for educators to select an appropriate format to conduct online CSys courses because of system thinking ability is cultivated through hands-on experiments, timely guidance, and feedback from the instructors and TAs.

Lack of personal attention. Online courses are commonly aimed to cover students at almost all levels and consequently lack of personalized contents [5]. Educators may ignore specific characteristics of a small group of students, negating such students' personalities and concentrations on CSys courses.

Lack of instructor-to-student interactions. Sufficient interactions with educators are important for students to feel connected and motivated [6]. In particular, one-on-one interaction and guidance from instructors significantly improve students' practice quality in the lab session. In conventional offline classrooms and laboratories, teachers and students can conveniently communicate with each other. However, in online courses, teachers and students are geographically distributed, so feasible interaction approaches in both lecture and lab session are strongly recommended.

Lack of real-hardware experimental environments. In CSys courses, hardware practice is essential to cultivate students' system knowledge and skills. A simple but efficient experimental platform plays an important role in motivating students' interest in practical engineering. Compared with offline courses, it is extremely expensive to distribute hardware kits to students nationwide or worldwide in online courses. Software simulator is a cost-effective choice for online CSys lab courses. However, simulators-based as virtual platforms can only support software experiments and cannot provide students with real hardware experiences.

During COVID-19, we experimented with a four phase course program design for teaching CSys courses to address the above challenges, in which the first phase was transforming the traditional face-to-face Computer Organization and Design (COD) course in University of Chinese Academy of Sciences (UCAS) to a synchronized FPGA cloud based online experience with supportive interaction methods in lecture and lab. In this report, we first describe the four-phase course program design, the transformed COD course design as an online synchronized lecture-lab format, and the course delivering platform. Then we present our finding on how various interaction methods for lecture and lab are perceived by the students. We conclude with lessons learned and look toward future work.

II. Curricular Design

A. 4-Phase Course Program Design

The proposed online CSys course program is divided into four phases (Fig. 1) and was delivered in four consecutive terms in UCAS to inspire students' abilities in hardware-software co-design via both large-scale training and small-group cultivation.

Phase 1 and 3 are required and large-scale teaching in formal semesters, while Phase 2 and 4 are designed as optional experiential projects for small-group cultivation in summer and winter breaks. Phase 1 is a Computer Organization and Design (COD) course for all students to learn core knowledge and skills, aiming to fundamentally establish students' hardware-software co-design thinking. Phase 2 is optional and targets groups of motivated students to conduct advanced projects, in which they can learn more practical knowledge and skills under faculty guidance. Phase 3 is implemented in another CSys courses (e.g., Operating Systems and Computer Architecture) arranged in the formal semester, in which all students from Phase 1 are required to attend and broader CSys core knowledge and skills are introduced. In Phase 4, volunteer students participated in solving complicated problems of real-world computer systems with CSys capabilities obtained in the previous phases. In the current design, students participated in Phase 4 were encouraged to contribute technical and feasible

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on January 26,2022 at 01:36:06 UTC from IEEE Xplore. Restrictions apply.

optimizations to the supportive teaching platform for next-year's online teaching offerings.

B. Transformed COD course Design

During February 2020 - July 2020, we transferred the conventional COD course to an online synchronized lecture-lab format to answer the COVID-19 crisis. Unlike other algorithm and software design courses, CSys courses present data movements between different hardware components triggered by software and signal variations occurring in digital circuits. As an introductive CSys course, COD lectures usually adopt an interactive way to guide students in viewing the hardware and software in a computer system as a whole. When migrating the COD course to online, an appropriate online teaching platform is important for the success of the course. In our reformed COD course, the synchronous meeting format was selected to enable live explanation of hardware behaviors in a virtual classroom.

For the lab component during which students apply the knowledge they have learned to conduct experiments on real hardware, there are three types of lab platforms available to choose from: software simulation, hardware development board, and remote hardware cloud. Software simulation is a low-cost method, students locally implement their hardware design within a simulator. However, software simulation cannot comprehensively describe real hardware, and does not provide student with real hardware experiences. A hardware development board is a way for students to experience real hardware. An FPGA board, for an example, is programmable hardware for students to turn hardware design into "real" hardware, and it is reusable due to its re-programmable feature. However, distributing hardware boards across an entire nation is economically and practically infeasible. The hardware cloud such as an FPGA cloud, is a method combining software and hardware. Students complete their hardware design using software and then implement the design with real hardware located in the cloud. Our reformed COD course used the hardware cloud as the lab platform.

C. COD Course Delivery Platform

The course delivery platform for the transformed COD course consists of several elements: Zoom for large-scale online synchronous teaching, UCAS's Learning Management System C Education Platform (CEP) for distributing course materials and discussion, WeChat for announcements and private communications between students and the instructors.

Our lab platform was based on [7] with technical modification and abstractions to hide actual hardware development flows to reduce the cost to students. To work on the labs, students write and synthesize hardware description language (HDL) codes in a local Linux virtual machine, then send the compiled FPGA configuration bitstream file to the remote lab platform to verify its correctness. The number of successful connections to the platform made by students and the students' geographical location according to their IP addresses are visualized in Fig. 2. With a total of 19447 connections

Fig. 2. Students' access records to the FPGA cloud platform depicted on a China provincial map (The redder the color of a province is, the more access indicated; certain provinces are in gray, meaning no access from those areas)

from 128 students averages 152 connections per students, and demonstrates that our lab platform is able to support students' remote lab experience across the country.

III. Interaction Methods

Without effective and meaningful ways to interact, students may feel disconnected from other students and instructors. We modified interaction methods frequently used in face-to-face courses for use in the transformed COD courses.

A. One-on-one Online Grading Interview

One-on-one grading interview [8] is widely used to help an instructor to check students' project outcomes in an offline CSys course with labs. During a face-to-face grading interview, students need to elaborate on their assignments, for example, source code of a software application or a hardware component, and receive timely feedback. This process requires the learner to deeply understand what the assignment is about, and thus effectively prevents plagiarisms. Having been adopted successful experience in previous face-to-face COD courses, this approach was adopted and adapted in the transformed online COD course.

Each TA sets up an individual Zoom meeting with waiting room, which allows only one student to join the meeting at a time. This setup helps those who are not confident at expressing themselves in front of other students to communicate freely. Students are randomly assigned to a TA's meeting room by a customized tool and they are allowed to record the interview process. After online discussion, the TA can assign an appropriate grade to the student. The instructor can also audit the whole class' interview process simultaneously which enables the instructor to easily switch among the virtual rooms to interact with the students.

B. Online Lecture Interaction

To improve peer interactions, students are provided the opportunity to engage in a survey of research topics related

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on January 26,2022 at 01:36:06 UTC from IEEE Xplore. Restrictions apply.

TABLE I
STUDENTS' SATISFACTION RATES WITH OUR COD COURSE IN THE MIDDLE AND THE END OF SPRING SEMESTER.

	Midterm	Final-term
Lecture	83.76%	93.34%
Lab	85.44%	84.78%
Aver. Rate	84.60%	89.06%

to the course to foster a greater understanding of CSys topics and present their learning to faculty and students.

IV. EVALUATION

Since this was the first fully online COD course we had taught, we collected data from survey results and platform logs to get a deeper understanding of the students' perception on the course design, and use the results to guide the next round's course design. The collected data is divided into two categories. One is the results from the questionnaires which are students' perception of the transformed online COD course. The second is the students' grades int lecture and lab components which reflect students' mastery of the class contents.

A. Online Course & Platforms

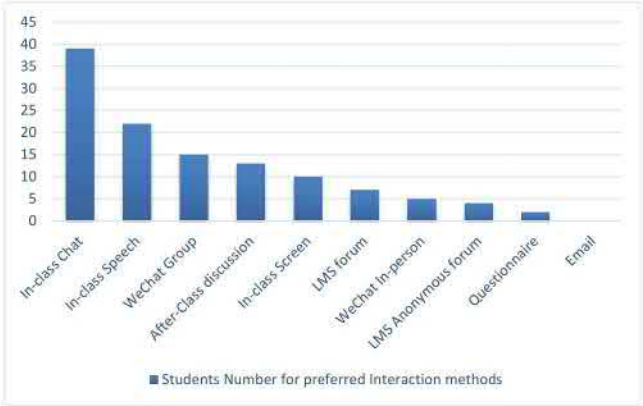
Course satisfaction rating. We asked the students "how satisfied are you with the online COD course?" after the midterm and the final exams. Table I shows that the satisfaction rates of lecture and lab components achieve 93.34% and 84.78% respectively, with an average of 89.06% at the end of Spring semester, indicating that students were largely satisfied with the online course.

For the lecture component, students were more satisfied at the end of term than at the midterm. In our opinion, both students and teachers were not familiar enough with the online format at the beginning of the term which suggested that students became more comfortable and familiar with the format as the semester progressed.

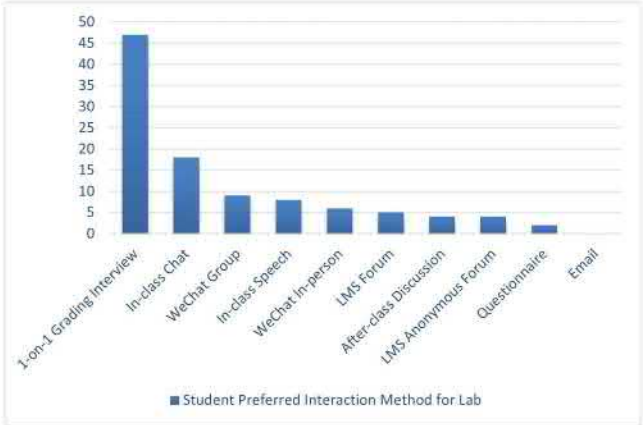
As for the lab component, the satisfaction rate dropped slightly (less than 1%) by the end of term. The drop in satisfaction may be attributed to the progressive difficulty of the labs, as well as the particularly challenging difficulty level of the optional Lab5.

Selection of video conferencing platforms. We noted several features of conferencing tools that are desired in our course and conducted several tests on three different platforms (Zoom, T Platform, and C Platform)' performance during a trial class. Based on student feedback, Zoom was selected primarily on the connection stability, teachers' speech clarity and slide quality. In our survey, an average of 98.39% students thought Zoom was better than the other two (74.01% and 41.38% for T Platform and C Platform respectively). What should be noticed is that Zoom was the best choice at the time according to our survey results, and those results can be very different at different locations and times.

Student preference of different interaction methods. We conducted a survey for the students to rank the ten interaction approaches adopted in lecture and lab parts respectively, and the results are shown in Fig. 3(a) and Fig. 3(b). We found that students preferred real-time interactions such as in-class chat and one-on-one grading interview. As shown in Fig. 3(b), 74.76% (77 out of 103) valid responses indicate that 1-on-1 grading interview(47), in-class chat(18), in-class speech(8) and after-class discussion(4) are the most favorite interaction methods among students.



(a) Ranking of interaction methods for lecture session



(b) Ranking of interaction methods for lab session

Fig. 3. Ranking different interaction methods used in COD course with lecture and lab sessions.

Student expectation of interaction time. To create a balanced learning experience between lecturing and interaction with the students, we were interested in finding how much interaction time students expect for during the lecture and lab. Table III indicates the proportion of real interaction time (Q&A, peer interaction etc.) recorded in our course vs. student expectation. From Table III, we found that students' expectation of interaction time was 21.02% and 35.65% for lecture and lab respectively in the midterm survey, while in reality the interaction time was 17.78% and 30.51%. After midterm, we increased the proportion of lecture's interaction time to 20.09% and kept the proportion of lab's at 29.33%. However, students' expectation for real-time interaction also increased to 25.03% and 35.26% after midterm. Taking this observation into consideration, we should plan for more interaction time

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on January 26,2022 at 01:36:06 UTC from IEEE Xplore. Restrictions apply.

TABLE II
GRADE DISTRIBUTIONS OF LECTURE AND LAB PART. (A: 90-100, B: 80-89, C: 70-79, D: 60-69, F: 0-59, W: WITHDRAW)

	A	B	C	D	F	W	Aver. Score
Lecture in 2020	11.57%	32.23%	34.71%	13.22%	4.96%	3.31%	76.40
Lab in 2020	14.81%	46.30%	15.74%	8.33%	12.04%	2.78%	77.00
Lecture in 2019	11.57%	38.02%	19.83%	22.31%	6.61%	1.65%	75.75
Lab in 2019	25.23%	49.53%	7.48%	13.08%	3.74%	0.93%	80.25

TABLE III
ACTUAL INTERACTION TIME VS. STUDENTS EXPECTED INTERACTION TIME.

	Before Midterm		After Midterm	
	Actual	Students expected	Actual	Students expected
Lecture	17.78%	21.02%	20.09%	25.30%
Lab	30.51%	35.65%	29.33%	35.26%

in the lab part. In our opinion, the increase of expectation interaction time indicates that students believe more interaction time is useful to their learning, and course designers should pay attention to the time allocated for instructor-student interaction.

Grades in COD course with lecture and lab. To study if online format would influence student grades, we compare the grades from the online COD course to the previous face-to-face offering in 2019. Table II shows the grade distributions and average scores of two COD course. We found that the students' average lecture score in 2020 was 0.65 points higher than in 2019, which indicates that transferring lecture online has little influence in students' performance. The students' average lab scores decreased from 80.25 in 2019 to 77.00 points in 2020, which may be resulted from the less than student expected real-time interaction in the labs.

V. LIMITATIONS & POTENTIAL IMPROVEMENTS

An Integrated platform. The onset of COVID-19 forced us to quickly build an online teaching platform consists of many heterogeneous elements to support the 2020 CSys courses as discussed above. In the future, we plan to design and build a platform with all the communication functions we need, on which students could attend lecture and conduct their labs.

An Integrated Lab Platform based on DevOps Concept. From to our experience, the completion of the labs still requires a lot of software development time on students' own computer before they submit the compiled file to the remote hardware platform. To better educate students about the software-hardware co-design, we plan to upgrade the current lab platform to contain the cloud-based software development based on the DevOps concept.

As mentioned in the evaluation section, students attending the online COD course did not perform as well as in the labs compared to the students in the previous year's face-to-face course. The unprecedented disruptive nature of COVID on the day to day lives of the students, the initial unfamiliarity of working within an on-line educational environment, and the progressively increasing difficulty of the labs may have impacted student performance. We intend to continue to evaluate and revise the labs as necessary in order to increase student performance.

One-on-one Online Grading Interview in Mixed Learning Environment. The one-on-one online grading interview was favored by the students. However, the effectiveness of this method will be put to test if courses are to be conducted in a hybrid format. Some students will attend the course offline and the others will take this course remotely. A possible solution is to divide all TAs into two groups, each is assigned to a different learning environment.

VI. CONCLUSION

In this paper, we present a 4-phase online CSys course program design which was first implemented in year 2020 at the onset of COVID-19. As the Phase 1 of the 4-phase program, a previously face-to-face COD course was transformed to online learning format. We focus the discussion on the process undertaken and the lessons learned in this transformation. We report that real-time interaction is critical in online learning and one-on-one online grading interview is widely approved by the students as a way to strengthen connections between teachers and students during labs. According to the results from a survey taken by 128 students in Spring 2020, 89.06% were pleased with the transformed online COD course, and 74.76% recognized the teaching effectiveness of the proposed interaction method. The final grades show that most students successfully acquired the basic hardware-software co-design ability. In addition, the 4-phase program design demonstrates its potential in increasing personal attention in the education. 7 and 2 students attended the optional advanced Phases 2 and 4.

VII. ACKNOWLEDGEMENTS

This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDC05030400, by the Proving Ground for Information Superbahn at CAS Innovation Academy for Network and Computing Technology, by the National Natural Science Foundation of China under Grant No. 61702485, and by the Youth Innovation Promotion Association of CAS under Grant No. 2017143.

We sincerely thank the anonymous reviewers of TALE 2021 for their feedback. We also thank Prof. Xilin Chen from ICT for guidance for our online course. We thank several teaching assistants for their contributions to make this course a success.

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on January 26,2022 at 01:36:06 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] John L. Hennessy and David A. Patterson, "A new golden age for computer architecture," *CACM*, vol. *62*, no. *2*, pp. *48-60*, 2019.

[2] Neil C. Thompson and Svenja Spanuth, "The decline of computers as a general purpose technology," *CACM*, vol. *64*, no. *3*, pp. *64-72*, 2021.

[3] Autumn Toney and Melissa Flagg. (2020) *U.S. demand for AI-Related talent* [Online]. Available: https://cset.georgetown.edu/publication/u-s-demand-for-ai-related-talent/

[4] David A. Joyner et al., "The synchronicity paradox in online education," in *Proc. 7th ACM Conference on Learning @ Scale.*, NY, USA, 2020, pp. *015-024*.

[5] Sean Kross and Philip J. Guo, "Students, systems, and interactions: synthesizing the first four years of learning @ scale and charting the future," in *Proc. 5th Annu. ACM Conference on Learning @ Scale.*, NY, USA, 2018, pp. *001-010*.

[6] Azad Ali and D. T. Smith, "Comparing social isolation effects on students attrition in online versus face-to-face courses in computer literacy," in *IISIT*, vol. *12*, no. *1*, pp. *011-020*, 2015.

[7] Ke Zhang, Yisong Chang, Mingyu Chen, Yungang Bao, and Zhiwei Xu, "Computer Organization and Design Course with FPGA Cloud." in *Proc. 50th ACM Technical Symposium on Computer Science Education.*, NY, USA, 2019, pp. *927–933*.

[8] Dirk Grunwald, Elizabeth Boese, Rhonda Hoenigman, Andy Sayler, and Judith Stafford, "Personalized attention @ scale: talk isn't cheap, but it's effective," in *Proc. 46th ACM Technical Symposium on Computer Science Education.*, NY, USA, 2015, pp. *610-615*.

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on January 26,2022 at 01:36:06 UTC from IEEE Xplore. Restrictions apply.

人才培养

处理器芯片设计人才培养探索：“一生一芯”计划实践经验与教训

余子濠 解壁伟 张 科 等
中国科学院计算技术研究所

关键词：“一生一芯” 芯片设计 人才培养

引言

中国科学院大学发起的“一生一芯”计划于2019年8月正式启动，目前已进行到第四期。在推进“一生一芯”计划的过程中，我们认识到该计划探索出的芯片人才培养新模式的意义与价值，同时也认识到这个模式仍然存在很大的改进空间。本文将介绍“一生一芯”计划的背景与目标，总结四期“一生一芯”计划实施过程中的经验与教训，并对未来我国芯片人才培养如何实现质量与规模的共同提升提出若干建议。

“一生一芯”的起源

据统计，2008~2017年十年间的计算机体系结构国际研讨会（ISCA）论文的第一作者中，有20%为中国国籍（美国本土为25%）[1]。但这些作者中有85%的人选择在美国就业，仅有4%的人在中国就业。大量中国学生到国外学习芯片设计，毕业后留在美国工作。中美两国在高水平芯片人才方面的差距，固然与过去多年中国芯片的产业环境有关，但同时也与国内高校的芯片人才培养方式密切相关。

近年来，国内对芯片人才的需求快速增长，人才缺口不断扩大。芯片企业和各科研院所对芯片人才的质量和数量都提出了较高的要求。高校的芯片人才培养体系面临着提升质量和扩大培养规模的双重挑战。过去十年，国内高校推行计算机系统能力培养的教学改革，部分高校开设了“计算机组成原理”课程并设置了处理器设计相关的实验课，但更多高校在处理器设计相关实验方面还处于刚刚起步的阶段。完整的处理器芯片设计能力，涉及计算机（CS）和电子（EE）等多个专业的知识，而目前国内高校计算机与电子专业的课程体系仍相对独立，缺少融合。同时，培养芯片设计（如流片等）人才的成本颇高，也让诸多高校望而却步。

事实上，美国也曾经历过类似的人才危机。20世纪80年代，全美上千所大学中只有不到100位教授和学生从事半导体相关的研究[2]。为此，美国国防部高级研究计划署（DARPA）在1981年启动MOSIS项目，资助大学开展芯片研究，并提供流片服务，通过多项目晶元（Multi Project Wafer，MPW）模式大幅降低芯片设计门槛。40余年来，MOSIS为美国的大学和研究机构流了60000多款芯片，培养了数十万名学生。由此可见，降低芯片设计门槛，可大幅促进芯片技术发展，同时提高人才培养效率。

2019年8月，中国科学院大学发起“一生一芯”计划，指导学生设计并实现一款可流片的开源处理器芯片，为我国培养芯片设计人才探索新的途径。

图1　四期“一生一芯”实施情况

“一生一芯”坚持“人才培养”和“公益”两个重要目标，培养方案对所有学生（包括在校生和已毕业的学生）和高校开放，并为在校学生提供免费流片机会，得到了广泛认可。

第四期“一生一芯”于 2022 年 2 月 20 日启动，吸引了来自全国 272 所高校的 1513 位学生报名参加（截至 2022 年 6 月 21 日），见图 1。经历了四期的“一生一芯”项目组（下文简称项目组）积累了很多经验和教训，也加深了对芯片人才培养目标和方法的认识。

培养独立解决问题的意识和能力——重中之重

在第二期“一生一芯”中，为了锻炼学生独立解决问题的能力，项目组只提供了必要的资料（包括 RISC-V 指令集手册、Chisel 教程、开源仿真器 Verilator 和测试方法的简单介绍等），其余资料要求学生自行查阅和探索。

但是学生并不适应这种以探索为主的学习方式。他们在项目启动会上跃跃欲试，一周后就反映难度较大，两周后仍然不知道如何下手，希望老师对每一步给予详细指导。到了第三周，一部分学生已经开始出现消极情绪，老师的鼓励并未起到明显效果。

第一期中也曾出现类似问题：学生的基础知识掌握得很好，但学习习惯和方法并未从课程实验的模式中调整过来。在第一期中，项目组通过细致地观察学生的调试过程，发现问题在于学生缺乏独立解决问题的意识和能力，具体表现为：(1) 学生习惯了课程实验的填鸭式训练，只关心自己负责的模块而忽略其他模块，难以形成对项目的整体认识。(2) 学生缺乏全局的系统观，无法站在全系统角度将出错现象关联到学过的知识点。例如学生在函数中定义一个局部数组，发现数组较大时程序出现错误，但未想到因局部数组分配在栈上而导致栈溢出引发的未定义行为。(3) 学生缺乏查阅手册和相关资料的意识，没有养成主动学习新知识和新工具的习惯。(4) 学生使用工具的意识薄弱，很少思考如何通过合适的工具提升工作效率，导致无法按时完成任务。

上述问题反映出高校课程实验教学与真实项目之间存在鸿沟和脱节现象。“一生一芯”应该将学生的学习习惯和实训手段从课程实验教学模式引导到真实项目实践模式上来，而不应该让“一生一芯”成为课程实验教学的重复和延伸。项目组工程师刘彤对此也深有体会，他本科修读商学院信息专业，后因兴趣转到芯片方向。在他看来，在工作中独立解决问题的能力是非常重要的，真实的项目开发过程中，没有人会像学校的老师那样把所有的细节告诉学生，如果学生总是依赖老师解决问题，到了工作岗位就无法独立承担项目。

为此，项目组成员余子濠博士作了一次题为《如何在没有指导书的情况下生存》的报告，指出学生当前存在的问题和应对的方法。报告中提到：(1) 学生需要从近一个月的尝试中判断自己是否缺乏解决新问题的能力，同时也需要明确成长是自己的责任；(2) 对问题进行分解是解决复杂问题的唯一方法，而分解问题的一个重要原则是 KISS 法则（Keep It Simple and Stupid）[3]，先完成，后完美，将一个大问题分解成若干个简单、明确、可操作的子任务；(3) 学生需要主动了解项目中的每一个模块，以及各个模块之间的联系，不能因为无须在某个模块中编写代码，就认为该模块与自己无关；(4) 遇到新知识和新工具时，首先要尝试通过搜索互联网和查阅相关手册独立解决问题。

上述报告成功扭转了学生的消极情绪。学生明确了自己的责任之后更愿意去动手尝试解决问题，

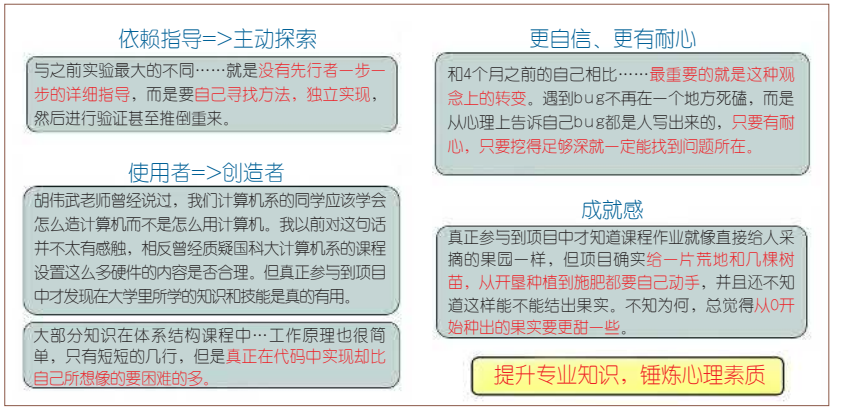


图2　第一期“一生一芯”五位学生的心得分享

同时也逐步从自己的付出中得到了正反馈：成功运行第一个测试，发现手册中有用的参数，成功接入测试框架，等等。虽然第二期大部分学生的成果都比第一期更简单，但在分享心得时都有与第一期“一生一芯”五位学生类似的感悟（见图2）：认识到独立解决问题的重要性，发现自己可以解决很多新问题，也因此收获了宝贵的成就感和自信心。

开展了两期“一生一芯”后，项目组把“一生一芯”的首要目标明确定位为培养学生独立解决问题的意识和能力，而并非要求学生冲刺性能指标。后来在第三期和第四期的实施过程中，项目组也发现不少非计算机和电子专业、低年级、零基础的学生，通过独立解决问题锻炼出了优秀的专业素质。在访谈过程中，他们积极分享自己的成长经历和实训经验。真实的项目（不仅仅是芯片设计）是一个复杂的系统工程，很多大公司的面试题表面上是考查学生能否将多门课程的知识融会贯通，但其背后是考查学生能否解决真实系统中的复杂问题。因此，项目组建议：高校课堂不仅要传授知识，更要改变传统的填鸭式教学模式，着重培养学生独立解决问题的意识和能力。

融合计算机和电子的软硬件协同全链条培养——坚持高标准

前两期“一生一芯”中，参加的学生都是修读计算机专业。自第三期开放报名后，项目组观察到，很多电子类专业的学生也对“一生一芯”感兴趣，甚至有不少非芯片相关专业的零基础学生报名参加。为放缓学生的学习坡度，项目组弱化了教学方案中系统软件和工具部分，将仿真环境和测试方法作为框架代码直接提供给学生，同时将学习任务周期分解到以“周”为单位。为了培养学生独立解决问题的意识和能力，项目组只提供相关资料，并安排助教在关键节点上提供指引性的讲座报告，其余细节需要学生自行探索。

随着时间的推移，有不少学生选择退出学习。针对这一情况，项目组多次讨论是否应当给予学生更多指导，也有成员提出“把进度较慢的学生搀扶到流片阶段”的建议。然而，当助教真正尝试“搀扶”的时候，却发现比想象中困难得多：芯片设计是一个复杂性较高的项目，学生只有掌握了独立解决问题的能力，对芯片的每一处细节都有清晰的认识，才有信心和能力解决遇到的复杂问题。“搀扶式指导”虽然可以帮助学生解决眼前的问题，却无法培养学生独立解决问题的能力，也并未真正帮助学生建立信心来完成后续学习。

上述结论在答辩环节得到了印证。项目组发现，不同层次高校、不同专业、不同年级中都有学生进入第三期流片名单（985、211和普通高校学生的占比分别为52.4%、21.4%和26.2%）。这些学生在答辩过程中都展现出优秀的独立解决问题的能力，他们在前期独立解决小问题的过程中积累了对处理器的整体认识，又在后期复杂的调试中帮助其他同学定位和排查问题。学生们都感觉到，三个月的高强度训练让自己的能力和自信心得到了明显的提升。

在第三期的实施过程中，项目组还发现如下问题：(1) 部分学生9月开学后课业压力较大或忙于导师项目，没有足够时间继续参加“一生一芯”。(2) 零基础学生直接参与“一生一芯”的难度较大，且需要投入更多时间学习前置知识。(3)“一生一芯”缺

乏完善的讲义，学生不易理解学习任务之间的联系，项目组也不易向学生传达正确的学习方法。（4）组队学习的效果普遍不如单人学习。在不少队伍中，一个队员通常只负责部分模块，不了解多模块之间的细节和联系，难以形成对处理器整体的认识，甚至出现多位高年级学生组队的设计成果与一位低年级学生单人参加的设计成果相当的情况。（5）学生普遍不重视软硬件之间的关联和工具的使用。项目组在答辩过程中考查关于软硬件协同工作的问题（例如，若对 PC 寄存器的复位值进行改动，该如何相应地修改程序的入口地址）时，发现 80% 的学生未思考过相关问题，还有学生表示其修读的是电子专业，不必了解软件知识，希望将考查内容聚焦到硬件设计上。对于项目组已经提供的框架代码和工具，大部分学生认为并不重要，处理器的编码开发才是第一位。

关于“一生一芯”的培养是否应该覆盖软件能力，项目组与“香山”团队[1]和部分企业进行了交流。“香山”团队提到，虽然“香山”是一个芯片项目，但开发者经常需要与软件和工具交互，而不仅仅是编写 RTL。例如在系统测试的过程中出错时，既需要分析硬件层次的信号，也需要阅读软件代码，以理解程序的预期行为；当仿真环境和工具的功能不满足需求时，需要有能力对它们进行增强，从而提升设计和验证的效率。企业则认为目前已有较多成熟的集成电路（IC）培训班可以培养 RTL 工程师，希望“一生一芯”能够培养更优秀的全栈复合型人才，帮助企业解决更有挑战的问题。

针对上述问题和反馈，项目组在第四期中进行了以下尝试：(1) 将“培养和流片”分离，学习流程不设置截止日期。项目组在一年时间中设置多趟流片班车，学生完成学习指标后，可就近申请流片。这使得学生不必追赶特定的截止日期，可以按照自己的节奏安排学习进度。只要学生愿意持续学习，项目组愿意在流程上给予最大的支持。(2)新增一个月的“预学习”阶段，供学生学习数字电路、C 语言、Linux 使用等

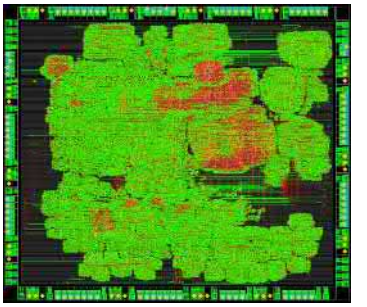


图3　使用开源EDA设计的“一生一芯”处理器芯片

基础知识，掌握后再进入正式学习。（3）取消组队，只允许单人参加学习，同时鼓励学生完成学习后组队攻关更有挑战的项目。（4）为了培养学生的软硬件协同能力，项目组引入了南京大学“计算机系统基础”课程的 PA（Programming Assignment）实验[4]，重新设计了教学方案，将系统软件和工具开发融合到芯片设计的过程中。（5）提出打通芯片设计前后端的全链条培养方案。目前该方案正在线下进行小规模试点，因受限于商业电子设计自动化（Electronic Design Automation，EDA）工具而暂未进行大规模推广。中科院计算所团队已经联合国内多家优势单位，布局了开源 EDA 项目（使用开源 EDA 设计的“一生一芯”处理器芯片如图 3 所示）。待开源 EDA 工具的可用性提升后，“一生一芯”将把全链条培养纳入正式的培养流程中。

总的来说，项目组提出了“高标准低坡度”的培养方法。一方面，即使是零基础的非计算机 / 电子专业的学生，也需要完成相同标准的学习目标；另一方面，“一生一芯”取消了截止时间的限制，鼓励学生按照自己的节奏持续学习。经过四期的迭代改进，“一生一芯”的培养方案更像一门融合计算机和电子专业知识的新型实践开放课程。

敬畏芯片——项目管理要跟上

第一期“一生一芯”的成功给项目组来带了很强的信心，但也使项目组忽视了“一生一芯”作为

[1]“香山”是中科院计算所的一个开源高性能 RISC-V 处理器项目。

芯片项目的工程属性，没有建立完善的项目流程来控制流片的风险，导致第二期生产的芯片无法点亮。项目组总结了两点教训：

其一，项目组的老师和学生并未充分重视片上系统（SoC）全系统测试验证的重要性。一部分学生在完成处理器的开发工作后，开始出现懈怠心理，认为处理器只需要通过自己设计的仿真测试即可，把 SoC 全系统仿真测试看作是冗余的任务。项目组的老师也未及时督促学生配合工程师完成测试验证工作。随着投片截止日期的到来，一部分学生设计的处理器只能带着较高的风险进行投片生产。

其二，项目组没有建立完善的项目流程，导致投片工作出现了疏漏：与工厂关于锁相环（Phase Locked Loop，PLL）模块的沟通对接出现了问题，使得芯片回片后无法产生时钟信号，整个芯片无法工作。这对项目组来说是一个沉重的教训。

吸取教训后，在思想方面，项目组提出“敬畏芯片”的理念。项目组资深工程师何伟在分享经验时对此作出了很好的诠释：“芯片设计，做到 99 分和 0 分没有区别”。项目组希望通过这次教训让大家将“敬畏芯片”的理念铭记于心，同时也提醒后续参加项目的老师和学生认真做好“一生一芯”中的每一个细节。

在行动方面，项目组解壁伟博士提出，要将第二期的经验教训融入进第三期的项目流程并实施，通过项目管理来保障芯片质量，避免再次出现同样的失误。项目组在第三期中明确规定，只有通过指定测试的处理器才能获得流片机会，同时为后端流程的实施建立双重保障：一方面小组成员需要制定检查表（check list）并严格执行，另一方面也会在投片前邀请资深工程师对芯片进行检视。在新的项目流程管理下，项目组重新设计了第二期的芯片，并与第三期的芯片一起投片。

大规模人才培养流程——学生才是主角

项目组原计划在第三期招收 100 名学生，但报名通知发布后，一周内报名参加的学生超过 500 人。这体现了大家对“一生一芯”的认可，但同时也对项目组的支撑工作提出了新的挑战：即使按每 50 人配备一名指导老师来计算，也需要超过 10 名指导老师，SoC 集成和后端物理设计工作也需要至少 5 名后端工程师来支撑。若报名人数继续增长，则需要更多的指导老师和工程师。

为应对大规模人才培养的挑战，解壁伟提出了“让学生指导学生”的方案：(1) 将第三期“一生一芯”划分成前端、SoC、后端三个小组，分别负责前端处理器核设计、SoC 集成验证和后端物理设计，并分别邀请有经验的工程师担任小组的技术指导。(2) 在报名的学生中筛选一部分能力较强且学习热情较高的学生进入三个小组担任助教，由工程师（或老师）指导他们完成相应的学习内容。(3) 由助教来支撑“一生一芯”的流程，担任指导老师的角色指导其他学生，或者负责 SoC 集成和后端物理设计的工作。

在上述流程的支撑下，第三期的讲座、答疑、进度检查、SoC 集成验证和后端物理设计工作，皆由学生完成，工程师仅在最后阶段参与芯片后端的检视工作并提出修改建议。这一流程不仅大大降低了大规模人才培养对指导老师和工程师的依赖，同时也给予学生更多的学习机会，充分发挥学生的积极性，让“一生一芯”的重点从“芯”转移到“生”。图 4 为第三期“一生一芯”项目组部分成员合影。

助教是开展大规模人才培养的中流砥柱。但是，如何培养好助教，是项目组面临的一项新的挑战。与工程师相比，助教是学生，项目组需要多方面充

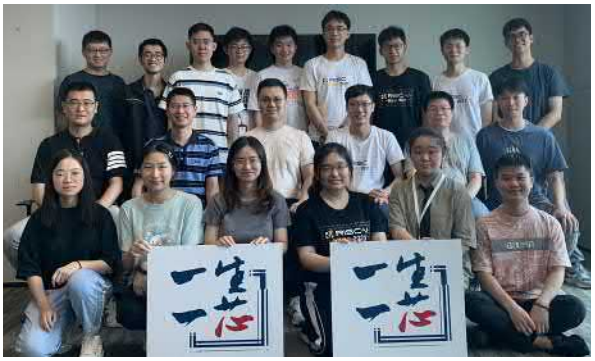


图 4　第三期“一生一芯”项目组部分成员合影

分考虑：(1) 助教在初次承担项目任务时相对缺乏经验，项目有一定的延期风险，项目组需要安排足够的时间进行容错。(2)助教除了需要学习技术知识，还需要学习如何指导其他学生。项目组不仅需要建立一套学习流程供学生开展学习，还需要建立一套“教会助教如何指导学生”的流程。(3) 项目与课程实验不同，如果出现失误，将会影响整个“一生一芯”的投片进度。项目组需要引导助教理解项目的严肃性，同时也需要帮助助教建立对“一生一芯”的归属感，提升助教的工作热情和责任心。(4) 助教的时间存在不确定性，可能会因学校课程或导师任务等突发情况随时离开项目组。项目组需要在助教的人数上进行备份，在部分助教离开的情况下也可以保证项目进度不受明显影响。(5) 部分助教对毕业课题有一定诉求，可能会带着毕业课题进入项目组，甚至是在项目组中寻找毕业课题。项目组需要分析每位助教的情况，尽量兼顾毕业课题、项目需求、个人兴趣三方面诉求，让助教在不影响毕业课题进度的情况下，积极地推进项目任务和助教任务。

总结

截至2022年7月，“一生一芯”已经历四期，项目组从芯片设计流程到教学方案，再到大规模人才培养流程的探索，逐步加深了对芯片人才培养体系的理解，总结建议和思考如下：

1.“一生一芯”专注培养高水平芯片人才，希望联合各高校共建芯片人才培养体系，加强对学生“独立解决问题的意识和能力”的培养，帮助学生从课堂向真实项目过渡。

2. 项目和企业对贯通电子和计算机专业知识的全栈复合型人才有很大需求，“一生一芯”需要坚持高标准严要求，但也需要通过减缓坡度来降低学生的学习压力。

3.“一生一芯”具有教学和工程的双重属性。流片是一项严肃的工程任务，项目组以及参加“一生一芯”的师生都要做到“敬畏芯片”，端正心态，认真做好每一处细节，建立完善的项目管理流程。

4. 助教是开展大规模人才培养的中流砥柱。围绕助教完善培养和组织流程，也是未来“一生一芯”的工作重心之一。

目前，“一生一芯”仍然有不足之处，项目组将持续改进“一生一芯”的教学指导方案，与各界一同推动我国芯片人才培养的进程。 ■

致谢：感谢中科院计算所孙凝晖院士对“一生一芯”计划的总体指导，感谢项目组唐丹、刘彤、何伟、李兴权等老师给予的指导和支持。感谢南京大学袁春风教授对本文提出的修改建议。感谢支撑“一生一芯”计划的助教，感谢所有参加“一生一芯”的师生！

余子濠

CCF专业会员。中国科学院计算技术研究所工程师。负责中国科学院大学“一生一芯”计划的方案制定和学生指导工作。主要研究方向为数据中心云计算架构QoS保障及二进制翻译。yuzihao@ict.ac.cn

解壁伟

CCF专业会员。中国科学院计算技术研究所助理研究员。主要研究方向为开源EDA、体系结构、高性能计算、性能测试和分析。xiebiwei@ict.ac.cn

张　科

CCF高级会员，CCF集成电路设计专委会执行委员、计算机工程与工艺专委会执行委员。中国科学院计算技术研究所高级工程师，中国科学院大学岗位教师。主要研究方向为芯片敏捷开发、异构加速计算与FPGA云。zhangke@ict.ac.cn

其他作者：包云岗

参考文献

[1] 包云岗，张科，孙凝晖. 处理器芯片开源设计与敏捷开发方法思考与实践[J].中国计算机学会通讯, 2019, 15(10): 42-48.

[2] Ralph C. SRC: Celebrating 30 Years[EB/OL]. https://www.src.org/src/story/src-celebrating-30-years/, 2012.

[3] 包云岗. 系统设计黄金法则：简单之美[J]. 中国计算机学会通讯. 2012, 8(5): 45-48.

（本文责任编委：李　涛　谭晓生）

计算机教育研究浅析
——从 ACM 计算机科学教育大会看国内外计算机教育科研

张 科[1,4] 张 铭[2] 陈 娟[3] 等
[1] 中国科学院计算技术研究所
[2] 北京大学
[3] 国防科技大学
[4] 中国科学院大学

关键词：计算机教育 计算学科 课程体系 计算思维

随着信息产业的蓬勃发展，各国劳动力市场对计算机人才的需求越来越大，未来计算机人才将成为经济增长的重要推动力。计算机教育乃至更广泛意义上的计算教育，关系到国家未来科技竞争力，已成为全民的基础教育学科，被世界各国广泛重视。如同先进的计算机理论和技术大多首先发表于相关学术会议一样，计算机教育领域的会议对教学也起到了非常重要的引领作用。本文通过概要介绍 ACM SIGCSE 组织的几个重要的计算机教育会议，回顾国际计算机教育与计算机教育研究领域的若干里程碑事件，展示近年来相关领域的国际最新动态和重要成果，梳理自 1956 年以来我国计算机教育及计算机教育科研领域的发展脉络，展现我国计算机教育工作者的贡献。

SIGCSE 及相关学术会议

美国计算机学会计算机科学教育特殊兴趣组 (ACM SIGCSE) 成立于 1968 年，注册会员近 3000 人，遍及 70 多个国家，规模及影响力位居 ACM 全部 36 个特殊兴趣小组的第三位，对全球各阶段、各层次计算机科学教育产生着重要影响。目前 SIGCSE 每年举办三个颇具影响力的年度国际旗舰学术会议：SIGCSE 年度技术大会 (SIGCSE Technical Symposium)、创新技术大会 (ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE)、计算教育研究会议 (International Computing Education Research Conference, ICER)；并将于 2019 年 5 月在中国成都首次举办全球计算教育大会 (ACM Global Computing Education Conference, CompEd)。

SIGCSE 年度技术大会是 SIGCSE 组织主办的历史最为悠久、规模最为盛大的旗舰会议。自 1970 年至今，已成功举办了 50 届。该会议通常于每年 2 月或 3 月在美国举行，为计算机教育研究人员提供了展示创新教学计划、课程体系和授课成果的平台，也为全球教师提供了各抒己见的机会。会议投稿涉及计算机教学及多样性、K-12 与初学者教学、教学法改进扩展、教学活动创新和计算机教学科研等诸多领域，涵盖从幼儿教育至高等教育各阶段的计算机教学相关方向。

回顾第一届 SIGCSE 年度技术大会，可以看到 50 年前计算机科学教育工作者关心的领域及本源问题，时下仍然被广泛关注 [1]：

- 什么是计算？我们是否兼容并蓄地将计算视为一个跨学科领域的广泛主题？
- 计算机课程应该教授什么内容？汇编语言适

用于哪些教学场景？如何教授人工智能[2, 3]、大规模信息处理（即当前的“大数据”）等学科[3, 4]？

● 如何设计并实施第一门计算机课程？

创新技术大会 (ITiCSE) 创办于 1996 年，通常安排在每年 6 月或 7 月于欧洲举行，旨在关注教学方法和教学技术的实际应用。工作组机制 (Working Group) 是 ITiCSE 的一大亮点。工作组成员共同开展和推进计算机教育相关课题，并在会议结束几个月后发布经同行评议后的研究报告。截至 2018 年，ITiCSE 已先后成立了百余个全球性工作组，涉及数百名计算机教育研究人员，所产生的开创性报告对指明计算机教育研究方向具有深远的影响，并为构建计算机教育研究知识体系作出了贡献。

计算教育研究会议 (ICER) 创办于 2005 年，通常于在每年 8 月或 9 月在北美、欧洲和大洋洲轮流举办，吸引了全世界从事计算机教育研究并采用跨学科（如心理学、教育学、人类学和统计学等）研究方法的大量研究人员参会。

全球计算教育大会 (CompEd) 是 SIGCSE 最新发起并主办的高水平计算机教育国际会议，将在北美和欧洲以外的国家或地区每两年举办一次，通过会议名称可以看出，CompEd 强调意义概念更为广泛的计算教育 (Computing Education)，而非传统的计算机科学教育 (Computer Science Education)。会议力图通过关注计算和学习科学的交叉领域，促进全球计算教育的发展。首届 CompEd 将于 2019 年 5 月 17~19 日在中国成都召开。我国计算机教育工作者和从事计算机教育科研的相关学者将充分利用这次机会，向国际专家学者全方位展示近年来中国在相关领域取得的成果。CompEd 2019 借鉴了 ITiCSE 的成功模式，将组织 3 个工作组，分别就在线判题系统 (Online Judge)、结对教学法 (Peer Instruction) 及计算伦理 (Computing Ethics) 教学等内容展开研讨和跨国协同工作。

从 SIGCSE 2019 会议看计算机教育热点趋势

工业界深度参与

有 51 家公司及机构在 SIGCSE 2019 会议上展出了自己的产品。如谷歌、微软和亚马逊的数字化教学平台，可为学生提供基于虚拟机的云基础设施，支撑云计算、机器学习、并行编程等实践课程内容；GitHub 的 GitHub Classroom，为教师和学生提供便利且安全稳定的课程作业发布和代码托管平台；zyBooks 的系列线上交互式电子教材，等等。

新课程体系与学生胜任力

制定符合 21 世纪下一个十年发展趋势的崭新课程体系是 SIGCSE 2019 的重要议程之一。由北京大学张铭教授担任指导委员会成员的“计算课程 2020”(Computing Curricula 2020, CC2020) 项目 (www.cc2020.net)，在本次大会中成功举办了兴趣小组讨论。该项目由 ACM 和 IEEE-CS 联合组织全球 36 位相关领域专家，计划通过对著名的 CC2005 课程体系进行版本更新，来应对计算机教育的未来挑战。CC2020 项目采用“计算”(computing) 一词作为计算机工程、计算机科学和信息技术等所有计算机领域的统一术语；采用“胜任力”(competency) 一词来代表所有计算机教育项目的基本主导思想。其目标是从知识、技能和性情三部分，使学生未来能胜任计算相关工作。讨论会重点介绍了项目的最新进展，听取会议现场计算机教育专家学者的建议和新观点，丰富相应的计算学科课程体系内涵，进一步完善学生胜任力培养模型。这也延续了 SIGCSE 2017 会议关于 CC2020 项目的深入讨论[5, 6]。ACM 教育委员会推动的数据科学工作小组 (ACM Data Science Task Force) 在其组织的专题讨论中，也表达了构建数据科学中计算机学科课程体系、着力培养学生在数据科学领域胜任力的设计愿景。

计算思维

如何培养学生的计算思维 (Computing Thinking and Computational Thinking) 是全世界计算机教育工作者共同关心的话题。密歇根大学的马克·古兹迪

奥 (Mark Guzdial) 教授在题为“Computing Education as a Foundation for 21st Century Literacy”的主旨报告中认为，编程教学的重要目标是为了让每一个人具备运用程序语言来表达思想并与他人交流的能力，最终正确理解和认知世界。古兹迪奥指出计算是各种思维方式的重要中间媒介，学习者通过计算思维所独有的自动化和模型化方法，可更为有效地掌握其他科学的本质和内在特征。

正是因为美国十多年来持续关注计算机教育与计算思维的全民培养，一项最新的美国、中国、印度、俄罗斯四国联合调查结果表明，美国计算机专业大四学生的计算机科学技能超过其他三国的同龄人[7]。即便如此，调查报告的作者还认为美国不应该安于现状，否则由于其他国家的持续投入，美国的优势将逐渐消失。可见，各国计算机教育工作者都应该对计算思维的培养加以重视，以避免未来国家竞争力落后。

我国的计算机教育界也开始广泛认同计算思维这一重要理念。中科院计算所的徐志伟研究员根据多年为中国科学院大学（简称“国科大”）计算机专业本科生开设并讲授“计算机导论”课程的实际教学经验出发，编写出版了《计算机科学导论》[8]，指出应在学生的入门学习期间着重培养学生的计算逻辑思维、算法思维、网络思维和计算系统思维，进而形成完整的计算思维。

新型硬件平台及计算机系统能力培养

如何为学生提供高性能的新型硬件计算平台，供学生开展实用的软硬件编程设计实验，也是SIGCSE 2019大会关注的重点内容。Intel公司展示其发布的面向下一代并行计算的多核处理器平台和硬件现场可编程门阵列 (Field Programmable Gate Arrays, FPGA) 教学平台及课程案例，支持并行计算、数据科学和人工智能等课程。美国高校（如加利福尼亚大学圣地亚哥分校团队和范德比尔特大学团队）也推出自研的新型硬件平台及软件开发环境，支持学生开展无人机、机器人、网络安全等实践课程。

中科院计算所张科、常轶松、陈明宇、包云岗及徐志伟在会上发表了题为“Computer Organization and Design Course with FPGA Cloud（使用FPGA云平台开展计算机组成原理课程教学活动）”的长文论文。这五位作者均是担任计算所科研与国科大本科一线教学工作的教师。论文介绍了教学团队在国科大本科生“计算机组成原理”理论和实验课程中的教学思路和教学方法，并围绕计算所自研的新型FPGA云平台在相关教学活动中起到的独特作用和教学效果展开论述。作为一种硬件可编程芯片，FPGA可以用于硬件逻辑模块（如组成原理实验课中的单周期处理器）开发与调试。为提升实验项目开发的敏捷性、降低学生在硬件环境搭建和调试的复杂度，教学团队结合学术及工业界主流技术发展趋势，采用了基于面向计算机系统能力培养需求定制的FPGA云平台。相比传统教学模式，FPGA云平台可以更好地达到预期教学活动目标，不仅方便学生随时随地开展实验项目，教师亦可对学生代码进行快速审阅和线上辅导。更为重要的是，云平台的可伸缩性及可扩展性可以更好地适应不断攀升的计算机专业课程选课人数，也能避免硬件资源的过度浪费与闲置。经过授课教师与学生在云平台使用过程中的闭环反馈与调试优化，基于FPGA云平台的敏捷开发流程得到了学生的广泛认可，部分优秀学生已在此FPGA云平台上开展基于RISC-V指令集的开源芯片原型设计与验证：不仅提升了计算机系统能力和软硬件编程能力，还极大增加了对于计算机系统知识及相关科研方向的浓厚兴趣。

值得一提的是，这篇论文是ACM SIGCSE年度技术大会50年以来，国内高校和科研机构作为第一作者单位发表的第四篇全文论文。在人工智能等多个计算机科研领域，我国科研学者的学术研究成果已在国际学术界与美国平分秋色。例如，人工智能领域旗舰会议AAAI 2019共录用1150篇长文论文，其中中国大陆一作长文共382篇，排名全球第一，比排名第二的美国高出118篇。然而在计算机教育领域，中国的相关研究还未能达到这样的成绩，亟待国内高校对计算机教育及计算机教育研究领域给予更多重视并加大投入。

国内计算机教育研究人员很难获得国家自然科学基金等高规格、高额度的国家级独立研究资助，是产生上述问题的重要原因之一。在科研项目考核导向的国内高校，如果继续“重科研、轻教学”，我们在未来信息领域的国际竞争中将面临人才缺失的困境[9]。

国际计算机教育发展的里程碑事件

在 SIGCSE 过去 50 年的努力推动下，国际计算机教育领域发生了若干里程碑式事件，并通过 SIGCSE 各学术会议的研讨推广，极大促进了相关产业的进步和发展。

计算学科的提出

ACM 前主席彼得·丹宁 (Peter J. Denning) 等人在 1989 年《美国计算机学会通讯》(*CACM*) 杂志上发表了一份著名的报告《计算作为一门学科》(Computing As A Discipline)[10]，定义了计算学科的内涵和外延，用“计算学科 (discipline of computing)”一词涵盖计算机科学与工程，使计算科学作为一门学科被广泛承认。2005 年，丹宁在 CACM 上发表题为“Is Computer Science Science?”的文章[11]，再次激发计算机科学如何持久地作为一门学科的深入讨论。

计算思维概念的提出与复兴

计算思维的概念在 20 世纪 70 年代末被提出，并在 80 年代成文，然后开始了一定范围内的探讨（如西蒙·派珀特博士 (Seymour Papert) 在 1980 年的 *Mindstorms: Children, Computers, and Powerful Ideas* 一书和 1996 年的“An Exploration in the Space of Mathematics Educations”论文）。然而，直到 2006 年 3 月，卡耐基梅隆大学的周以真 (Jeannette Wing) 教授在 *CACM* 上发表文章[12]，才将计算思维的概念重新带入了大家的视线并引起了各界关注。她认为计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。她同时提出了计算思维的特征及方法，使计算思维在世界范围内得到推广。周教授的这篇文章以及她后续作为美国国家科学基金会 (NSF) 计算机信息科学与工程学部 (Computer and Information Science and Engineering, CISE) 助理主任客观上推动了计算思维的 NSF 基金立项，并且将计算学科的概念、方法、工具和技术融入到中小学和大学本科教育中。

计算机学科规范和课程体系的发展历程

IEEE-CS/ACM 发布的“Computing Curricular 1991”(CC1991) 是第一个较为系统的学科规范。然而 CC1991 及之后相继发布的 CC2001 和 CC2004，并没有对计算学科进行细致划分。自 CC2005 开始，计算学科规范被划分为计算机科学、计算机工程等 5 个各有侧重的分支，其目的是更好地适应社会发展的需求。ACM/IEEE 于 2008 年发布了信息技术学科规范“Information Technology 2008”(IT2008)，突出强调集成能力培养，使 IT 应用教育得以松绑，学生可专注学习网络维护、数据库调优等专项技能，避免学习计算机基础、数学等复杂理论，适应了工业界对信息技术领域人才培养的客观需求。

之后发布的“Computing Science Curricula 2013”(CS2013) 进一步对计算机科学完整知识体系进行了重新定义并新增计算思维、信息安全、并行计算等知识模块，特别强化系统基础 (system fundamentals)。2016 年 12 月发布的“Computer Engineering Curricula 2016”（CE2016，计算机工程课程体系指南）为保持教学内容先进性，对计算机工程专业的知识体系和知识领域进行了重新梳理，并期待能引领未来 10 年的计算机工程教育方向。IT2017 引入了胜任力 (competency) 模型，旨在为全世界的教育机构制定未来十年的信息技术课程体系提供指南，对信息技术学科的内涵以及融合知识、技能、性情的信息技术能力进行了定义，具体包括 10 个基本领域、9 个补充领域和其他选修领域组成的信息技术能力域。正在开展的 CC2020 项目采纳了

IT2017 的胜任力模型，强调元学科规范 (Meta Curricula)，融合知识、技能、性情三个方面的综合能力培养，加强了对职业素养、团队精神等方面要求，鼓励各教学机构根据自己的定位而设计具体培养方案。以上这些新的学科规范和课程体系都对实现我国“新工科”建设所强调的“以能力培养促进工程科技创新和产业创新”的目标有着重要的促进作用。

大规模在线教育 (MOOC)

伴随着 Udacity、Coursera 和 edX 的正式上线，2012 年成为名副其实的 MOOC 元年。Udacity、Coursera、edX 三大标志性 MOOC 平台容纳的大学数量、提供的课程规模、注册学生人数早已超过千万。围绕 MOOC 的相关教育技术、教育理念、教学方式等引发的巨大变革也在深刻影响着中国的计算机教育。各种新兴的网络和多媒体技术也极大地推动着 MOOC 的发展。同时一种更加灵活的针对小众人群的 SPOC 教学方式也应运而生。无论是 MOOC 还是 SPOC，中国应用在线教育达到教育资源最大化的前景十分广阔。

中国计算机教育及计算机教育科研的历史与现状

我国计算机教育可追溯至 1956 年清华大学和哈尔滨工业大学两所高校率先设立计算机专业 [13]。其中，哈尔滨工业大学计算机专业在 1956 年筹建本科生班时就同时举办了研究生班。回顾往昔，我国计算机教育事业发展曾有过四次高潮，并正在经历第五次高潮。

第一次高潮（1956 – 1960 年）

1956 年，国务院制定了《十二年科学技术发展规划》(即《1956 – 1967 年科技发展规划》)，将建立与发展我国的计算机技术纳入规划并列为五项紧急措施之一。该规划推动了我国计算机专业的创建。到 20 世纪 50 年代末，在教育部的部分重点大学和国防工业院校建立了 14 个以“计算装置与仪器”“计算数学”为主要分支的计算机专业。

该阶段的计算机教育特点是：(1) 张效祥、杨芙清、夏培肃、金兰等计算机专业创始当年陆续从国外学习归来（前三人 1991 年同时被评为中国科学院院士），带回计算机新技术，成为我国第一批计算机专业教师培训的启蒙老师；(2) 大多采取“以任务带学科，以科研带队伍”的专业发展模式，组织教师和学生在研制计算机的同时学习计算机设计技术，把培养高素质的人才与发展新技术和支持新技术产业结合起来。例如，中国科学技术大学于 1960 年研制成功中国第一台自主设计的通用电子计算机——107(KD-1) 机，哈尔滨军事工程学院于 1965 年研制成功中国第一台晶体管数字计算机——441-B 计算机。这种发展模式一直持续到 20 世纪 70 年代。

第二次高潮（1978 – 1986 年）

十年动乱结束，国家将工作重点转移到四个现代化建设上。1978 年 3 月全国科学大会的召开、《1978 – 1985 年全国科学技术发展规划纲要》将电子计算机列为 8 个影响全局的综合性课题之一等事件，大大促进了计算机教育事业的快速发展。这段时期共开办了 74 个以“计算机及应用”和“计算机软件”为主要分支的计算机专业。

该阶段的计算机教育特点是：(1) 改革开放促进了计算机新技术、新课程的引进。例如邀请美国一流计算机专业的教授在北京讲授编译技术、操作系统、系统结构等课程；向西方先进国家大量派出进修教师、访问学者，并在他们回国后组织开设相应课程并编写教材。(2) 软件得到普遍重视，计算机应用技术教育开始普及。到 1982 年，我国的软件专业由原先的几个迅速扩大到三十多个。(3) 高层次人才培养开始起步。1978 年恢复计算机软件和计算机应用两个本科专业招生；同时按 5 个二级学科设点招收培养新一批计算机专业研究生。1981 年经国务院学位委员会审定，首批计算机学科硕士点和博士点在部分重点大学批准建立。之后，在 1987 年我国首次设立重点学科，其中计算机科学与技术包括计算机组织与系统结构、计算机软件、以及计算机

应用等三个二级学科。

第三次高潮（1994 – 2003 年）

1994 年，全国计算机专业统一为“计算机科学与技术”，并在之后受到各项国家科教战略的广泛影响。1995 年我国开始实施“科教兴国”的发展战略，启动“211 工程”重点建设一批高校和一批学科。在 1998 年亚洲金融危机之际，教育部出台《面向 21 世纪教育振兴行动计划》，并于 1999 年启动以“拉动内需、刺激消费、促进经济增长、缓解就业压力”为目标的高校扩招，同时启动以“建设具有世界先进水平的一流大学”为目标的“985 工程”。另一方面，网络技术与应用促使计算机技术发生革命性变革，计算机应用更加普及，也引发了我国计算机专业内在的发展变化。若干相关课程被引入到计算机专业的教学计划中，还有一些学校于 2001 年开始办起了“网络工程”“ 软件工程”“ 电子商务”“信息安全”等新专业。2001 至 2002 年，中国计算机学会教育委员会、全国高等学校计算机教育研究会和清华大学出版社联合组建研究组，成员由清华大学等八所国内高校的计算机教育专家组成，研究组参考 IEEE-CS/ACM 的 CC2001，制订并出版了《中国计算机科学与技术学科教程 2002》（简称 CCC2002）。到 2003 年末，全国具有计算机专业的学校达 505 个，在校生近 30 万。

该阶段的计算机教育特点是：(1) 计算机专业的内涵和外延发生较大变化。计算机专业的教育内容已不再局限于传统的计算机组织与体系结构、计算机理论与软件和计算机应用技术，计算机网络及其应用技术、多媒体及其应用技术、网络与信息安全等教育内容得以强化。(2) 办学单位和在校生人数迅速增加。(3) 教材内容逐步与国际接轨，2000 年前后，我国几大出版社从国外著名出版公司引进了成套的计算机专业教材，其中包括不少经典的著名教材。

第四次高潮（2003 – 2016 年）

在这期间，我国高等教育面临一系列新情况、新问题：(1) 高校继续扩招 [14]，带来了高校教育质量问题；(2) 计算技术飞速发展，进入以移动互联、物联网、云计算和大数据计算为主要特征的新型网络时代，对计算机专业人才的知识结构与创新能力提出更高要求；(3) 我国开始实施人才强国战略，引入各类人才计划等，很多高校片面地将论文、专利、资金数量作为人才评价标准，导致潜心将科研融入到教学、开展教学研究的高校教师越来越少。为应对高校扩招带来的教育质量问题，教育部于 2001 年重组各类学科教学指导委员会，将其工作职责由以教材编审为主转向为对学科的教学工作进行全面的研究、咨询和指导；2003 年启动“高等学校教学质量与教学改革工程”（简称“质量工程”），通过建设品牌专业、工程认证、精品课程等来改善高校教育质量。除上述面向各类专业的改革措施外，我国计算机教育工作者也结合本专业情况采取一系列新举措。2006 年在教育部指导下启动了具有开创性意义的计算机专业工程教育认证。同年计算机科学与技术专业教学指导分委员会（2013 年变更为计算机类专业教学指导委员会，简称“教指委”）先后发布《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求（试行）》与《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》等刊物；2007 年和 2010 年教指委又分别发布了《高等学校计算机科学与技术专业公共核心知识体系与课程》和《高等学校计算机科学与技术专业人才专业能力构成与培养》，进一步规范国内计算机专业教育。2010 年 11 月，陈国良院士在第六届大学计算机课程报告论坛中发表题为《计算思维与大学计算机基础教育》的报告并倡议：以计算思维能力培养为切入点、推动大学计算机基础课程教学改革；随后，2012 年首届“计算思维与大学计算机课程教学改革研讨会”召开；2016 年《大学计算机基础课程教学基本要求》发布。这些措施在一定程度上有积极作用，但是由于人才评价体制等因素，整体上对教学研究的深度和投入不足。

该阶段的计算机教育特点是：(1) 计算机专业的内涵和外延进一步扩张。2012 年计算机专业教育发展为计算机类专业，主要包括计算机科学与技术、

软件工程、网络工程、信息安全、物联网工程、数字媒体技术等6个基本专业，逐步增设智能科学与技术、空间信息与数字技术、电子与计算机工程等3个特设专业。2013年全国计算机类专业的招生专业点突破2000，达到2082个。(2) 教指委积极开展计算机专业本科教学的研究、咨询、指导、评估、服务工作，组织研究小组研究高等学校计算机科学与技术专业发展战略并出版一系列专业规范。(3) 在编译原理和操作系统等计算机专业核心课被很多学校取消或减少学时的同时，计算技术与信息产业发展对学生提出系统能力培养的新需求。2013 – 2017年，教指委启动“计算机类专业系统能力培养研究”项目，分析系统能力的主要内涵及培养需求[15]。(4) 引入若干与具体编程语言、应用技术相关的课程，各种细分专业的教学计划区分度不明显。(5)2006年周以真教授关于计算思维文章的发表、2012年美国顶尖大学陆续设立MOOC对国内计算机教育界产生影响，掀起国内一系列的相关研究与建设。中国高校计算机教育MOOC联盟于2014年12月在京成立。(6) 虽然出版了很多教材，但是大量为应用技术类，许多核心专业课的教材内容滞后。(7) 大多数高校教师的科研与教学不相匹配，科研有很大发展，但是将科研运用到教学上的教学研究极其缺乏，导致高校教育质量提升内力不足。(8) 虽然国内科研经费显著增长，但是教研经费微薄，甚至轻视教研。

第五次高潮（2017年至今）

2016年“新工科”概念被提出，其出发点是适应引领新经济发展的战略视角，对国际工程教育改革发展做出中国的本土化回应。2017年以来，为主动应对新一轮科技革命和产业变革，支撑服务创新驱动发展、“新一代人工智能发展”等一系列国家战略，教育部积极推进“新工科”建设，先后形成复旦共识、天大行动和北京指南，发布关于新工科研究与实践的开展、项目推荐和首批项目公布以及《高等学校人工智能创新行动计划》等通知。2017年11月，在教育部高教司、工信部信息化和软件服务业司及人事教育司共同指导下，信息技术新工科产学研联盟在京成立。“新工科”建设旨在针对新兴产业，以互联网和工业智能为核心，建立“存量更新”的新型工科专业以及“增量补充”的新生工科专业和新兴工科专业，全力探索形成领跑全球工程教育的中国模式和中国经验。为顺应“新工科”建设理念在计算机科学领域的发展需要，2016年新设数据科学与大数据技术、网络空间安全等战略性新兴产业相关专业。教育部也在2018年正式发布计算机类专业教学质量国家标准。截至2018年，我国计算机类专业涵盖计算机科学技术、软件工程、网络工程、物联网、信息安全、数字媒体技术、智能科学与技术等14个专业，专业布点数达3367个，占工科门类的18%，在校生约150万人。计算机学科近年来一直是我国各大高校的热门学科，在招聘需求和毕业生专业薪酬上均排名第一。根据目前的发展趋势，预计2020年全国计算机类专业招生专业点将突破4000个。2019年3月，教育部公布2018年度普通高等学校本科专业备案和审批结果，其中35家院校新设人工智能专业。这将势必大幅提升国内人工智能领域相关人才储备的数量和质量。

在“新工科”建设背景下，清华大学出版社于2018年7月整理出版《培养计算机类专业学生解决复杂工程问题的能力》，正式明确计算机类专业本科教育基本定位；同年12月，在2018 – 2022教育部计算机类专业教指委的第一次工作会议中审议通过计算机类教指委1个联盟（计算机教育联盟）、2个工作组（教学质量、系统能力）、3个专业组（物联网、大数据、人工智能）的成立；其中系统能力工作组成立智能时代计算机专业教育研究组，启动系统能力培养项目第二阶段的研究。

历数各次高潮，可以看到教学与研究深入融合是提升我国高校教育质量的关键因素之一。国家需要重视对教学研究的投入，改革人才评价制度。从2018年开始，国家自然科学基金委(NSFC)在科学基金申请代码中增设教育科学研究相关申请代码F0701，支持用自然科学的手段、范式来研究教育科学问题，推动教育的科学发展，支撑新时代的教育。2018年度国家自然科学基金F0701类共收到项

目申请737项，涵盖个性化教学、教育大数据、机器学习、增强现实、教育机器人、学习评测、交互学习、数字资源、协同学习、资源配置等十个主题聚类，充分反映出广大教师对教育研究参与的积极性。其中60项申请获得资助，直接经费总额为2346万元，批准率为8.1%。通过对资助项目研究内容的分析可以看出，目前的教育信息科学与技术研究仍处于技术迁移期，主要以信息领域向教育领域渗透的研究工作为主，但对教育领域的重大关键科学问题缺乏深刻凝练，深度交叉融合不足[16]。F0701下设10个三级申请代码（见表1）。在2018年度F0701的申请和资助中，F070108的申请占比最高，为23%，批准率为8.9%；F070102批准率最高，为13.4%，申请占比为13%；F070105批准率最低，为0，申请占比为5%。F0701代码的设立，开启了教育跨学科交叉融合研究新元年，标志着教育研究进入了新时代，将极大促进中国教育科学研究水平的提高。希望我国能在不同层次提供高校教师开展教学研究的资金、政策等的支持。

表1　F0701下设三级申请代码

代　码	方　向
F070101	教育信息科学基础理论与方法
F070102	在线与移动交互学习环境构建
F070103	虚拟与增强现实学习环境
F070104	教学知识可视化
F070105	教育认知工具
F070106	教育机器人
F070107	教育智能体
F070108	教育大数据分析与应用
F070109	学习分析与评测
F070110	自适应个性化辅助学习

未来发展

2017年，北京大学教授张铭团队采访了ACM SIGCSE主席安布尔·塞特尔(Amber Settle)教授[17]。塞特尔指出，计算机科学技术飞速更新的特性意味着计算机专业教育必须紧跟业界发展节奏。这要求计算机教育者不仅能对计算机领域的现状做出准确判断，而且能敏锐地捕捉到计算机技术未来的发展动向。学科自身和现代社会的发展都要求开展面向非计算机专业人士的计算机教育。如何提供令全社会满意的教育服务，并适应编程学习者不断低龄化的趋势，需要多领域专家进行广泛探讨。计算机教育不仅是任课教师的任务，更与整个行业的未来发展息息相关。它需要足够的支持以产出优秀的研究成果并应用于实践，也需要广大计算机爱好者的共同努力。

在2018年12月的*ACM Inroads*中，美国波莫纳学院(Pomona College)的金·布鲁斯(Kim B. Bruce)教授根据自身教学经验和分析结果，列举了他认为的计算教育领域存在的问题[18]。这些问题虽不能完全代表计算教育领域，但也凸显了一线教师对计算教育的关切程度：

- 如何应对持续上升的计算机专业本科生课程选课人数；
- 如何使计算机专业课程更具包容性；
- 如何支持并发展大学预科阶段的计算机教育；
- 如何让学生深刻地理解计算的道德含义；
- 如何缓解计算机导论课程中的高退课率。

此外，计算思维作为热点，引发最近几期*CACM*文章的观点讨论。恩里克·纳德力(Enrico Nardelli)强调在计算思维中自动化与建模的重要性[19]。这与SIGCSE 2019技术大会上马克·古兹迪奥教授在主旨演讲中的部分观点是一致的。伊利诺伊大学厄巴纳-香槟分校与麻省理工学院的学者提出，我们在教学中除了需要培养学生抽象的计算思维，更需要“计算行动”(computational action)，鼓励学生尤其是低龄学生避免机械地完成大量过时的重复习题，更多参与到有现实意义的实践项目之中[20]。

随着近年来我国信息产业的飞速发展，全社会对计算机专业人才的需求不断提高。这些需求也激发中国的计算机教育工作者开拓国际视野，不断寻求与世界先进计算机教育前沿接轨并深度融合；同时结合自身国情，提出一些独具特色的观点见解和未来规划纲要，为全球计算机教育及计算机教育科

研事业的未来发展贡献“中国力量”。

2019年1月，高等教育出版社出版了由“中国计算机教育20人论坛”发布的《计算机教育与可持续竞争力》[21]一书。相比上述国际最新的计算机教育领域观点，我国的计算机教育专家们站在了更为全面且更加符合中国国情的角度，阐述了可持续竞争力（或称胜任力）的概念[1]，以及如何培养可持续竞争力。中国学者还创新地提出了敏捷教学的概念与内涵，并再次重申了计算思维的重要性和特征。

专家普遍认同要把计算机科学为主导的信息技术作为基础，服务其他学科在新时期的发展需要。随着国内广大教育者的持续投入，“新工科”发展已凸显成效。2018年3月底，教育部公布了首批612项“新工科”研究与实践项目。从项目一年来整体实施的效果来看，不少专业都在基于信息技术重塑本专业的工科教学体系。

可以预期，随着我国对高等教育、尤其是计算机教育重视程度的不断加强，特别是基金委从2018年开始确立了F0701教育信息科学与技术二级代码，全面启动支持教育科学基础研究的相关工作，国内将会有更多的有识之士参与到计算机教育事业，助力发展全民计算教育，推广先进计算机教育理念的全球化传播，携手共建计算领域的人类命运共同体。 ■

致谢：

此专题内容得到了北京市科委北京市重点研发计划新一代人工智能技术培育项目(Z181100008918005)的支持。此外，本文作者感谢以下老师及专家对本文提出的宝贵意见与建议：John Impagliazzo、Alison Clear、Jian Zhang、Yanxia Jia、Haiyan Cheng、Meng Han、Liling Li、李晓明、徐晓飞、蒋宗礼、张龙、何炎祥、王志英、陈熙霖、许卓群、李文新、杨晓春、韩飞、杨波、傅育熙、李晓鸿、陈卫卫、贾斌、孙艳春、谢昆青、刘卫东、战德臣、王漫、严勇、毛晓飞等。

张　科

CCF专业会员，CCF计算机工程与工艺专委委员。中科院计算所高级工程师，中国科学院大学岗位教师。主要研究方向为计算机系统结构与异构加速计算。zhangke@ict.ac.cn

张　铭

CCF高级会员，CCF计算机教育工委副主任。北京大学教授。教育部计算机课程教指委委员，ACM SIGCSE China主席。主要研究方向为机器学习、大数据分析、计算机教育。mzhang@net.pku.edu.cn

陈　娟

CCF高级会员，CCF高性能计算、理论计算机专委委员。国防科技大学副教授。ACM SIGCSE China副秘书长。主要研究方向为高性能计算、低功耗软件优化技术、计算机教育。juanchen@nudt.edu.cn

其他作者：张　昱　常轶松

参考文献

[1] Beck R E, Walker H M. The SIGCSE Symposium: A Brief History[J]. *ACM Inroads*, 2018, 9(4): 31-39.

[2] Hammer P C. Undergraduate Computer Science Education[C]//*Proceedings of the 1st ACM SIGCSE Technical Symposium on Computer Science Education* (SIGCSE'70). ACM, 1970: 1-5.

[3] Atchison W F, Conte S D, T E, et al. Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science[J]. *Communication of the ACM*, 1968, 11(3): 151-197.

[4] Guzdial M. What We Care About Now, What We' ll Care About in the Future[J]. *ACM Inroads*, 2018, 9(4): 63-64.

[5] Clear A, Parrish A, Zhang M, et al. CC2020: A Vision on Computing Curricula[C]//*Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (SIGCSE'17). ACM, 2017: 647-648.

[6] 张昱，陈娟，肖胜刚，张铭．由第48届ACM计算机

[1] 可持续竞争力指面对未来社会变化和竞争的适应能力、基于使命和技术的创新能力、推动社会发展与科技进步的行动能力。

科学教育大会看国内计算机教育科研 [J]. 计算机教育 , 2017, 9:176-179.

[7] Loyalka P, Liu O L, Li G, et al. Computer science skills across China, India, Russia, and the United States[R/OL].Proceedings of the National Academy of Sciences (2019). DOI: 10.1073/pnas.1814646116.

[8] 徐志伟 , 孙晓明 . 计算机科学导论 [M]. 清华大学出版社 . 2018.

[9] 张铭 . 计算机教育的科学研究和展望 [J]. 计算机教育 , 2017(12):5-10.

[10]Denning P J, Comer D E, Gries D, et al. Computing As A Discipline[J]. Communications of the ACM. 1989, 32(1).

[11]Denning P J. Is Computer Science Science? [J]. Communications of the ACM. 2005, 48(4): 27-31.

[12]Wing J M. Computational Thinking[J]. Communications of the ACM. 2006, 49(3): 33-35.

[13]教育部高等学校计算机科学与技术专业教学指导分委员会 . 中国计算机本科专业发展战略研究报告暨专业规范 [R]. 高等教育出版社 . 2006.12.

[14]教育部 . 中国教育概况——2016 年全国教育事业发展情况 [OL].(2017-11-10). http://www.moe.gov.cn/jyb_sjzl/s5990/201711/t20171110_318862.html

[15]王志英 , 周兴社 , 袁春风 , 等 . 计算机专业学生系统能力培养和系统课程体系设置研究 [J]. 计算机教育 , 2013(9):1-6.

[16]郑永和 . 科学基金资助教育信息科学与技术的科学计量分析—基于 2018 年度 F0701 科学基金申请与资助项目数据 . 科学基金教育信息科学与技术（申请代码 F0701）资助管理工作研讨会 . 北京 . 2019.1.22.

[17]陆俊林 , 陈沫 , 张铭 . 现在，我们需要怎样的计算机教育？——专访 ACM SIGCSE 主席安布尔·塞特尔博士 [J]. 中国计算机学会通讯 , 2017, 13(10): 72-75.

[18]Bruce K B. Five Big Open Questions in Computing Education[J]. ACM Inroads, 2018, 9(4): 77-80.

[19]Nardelli E. Do We Really Need Computational Thinking?[J].Communications of the ACM, 2019, 62(2): 32-35.

[20]Tissenbaum M, Sheldon J, Abelson H. From Computational Thinking to Computational Action, Communications of the ACM, 2019, 62(3): 34-36.

[21]“计算机教育 20 人论坛”报告编写组 . 计算机教育与可持续竞争力 [M]. 高等教育出版社 . 2019.

[22]Computer Science. Wikipedia. https://en.wikipedia.org/wiki/Computer_science

[23]History of the Department. Department of Computer Science, Purdue University. https://www.cs.purdue.edu/history

[24]FACT SHEET: President Obama Announces Computer Science For All Initiative. The White House Office of the Press Secretary. https://obamawhitehouse.archives.gov/the-press-office/2016/01/30/fact-sheet-president-obama-announces-computer-science-all-initiative-0.

[25]计算机技术与产业（中国信息化进程蓝皮书 . 李国杰文选 . http://www.ict.cas.cn/liguojiewenxuan/wzlj/lgjxsbg/201201/P020120106404772326722.pdf 2009.07.28

[26]邓志鸿，谢昆青 . 机器学习课程的教学实践——以北京大学“智能科学与技术”本科专业为例 . 计算机教育，第 10 期，2016.10.

[27]国务院学位委员会 教育部关于增设网络空间安全一级学科的通知 . 学位〔2015〕11 号 . 中华人民共和国教育部 . 2015 年 6 月 11 日 .

[28]一流本科教育宣言（“成都宣言”）. 中华人民共和国教育部 . http://www.moe.gov.cn/jyb_xwfb/xw_fbh/moe_2069/xwfbh_2018n/xwfb_20180622/sfcl/201806/t20180622_340649.html. 2018 年 6 月 22 日

[29]教育部关于加快建设高水平本科教育全面提高人才培养能力的意见 . 教高〔2018〕2 号 . 中华人民共和国教育部 . 2018 年 10 月 8 日 .

ZyForce: An FPGA-based Cloud Platform for Experimental Curriculum of Computer System in University of Chinese Academy of Sciences

Ke Zhang, *Institute of Computing Technology of Chinese Academy of Sciences (ICT of CAS) & University of Chinese Academy of Sciences (UCAS)*
Mingyu Chen, *Institute of Computing Technology of Chinese Academy of Sciences (ICT of CAS) & University of Chinese Academy of Sciences (UCAS)*
Yungang Bao, *Institute of Computing Technology of Chinese Academy of Sciences (ICT of CAS) & University of Chinese Academy of Sciences (UCAS)*
Contact: zhangke@ict.ac.cn

To cultivate students' capability of computer system thinking and software/hardware programming, experimental curriculum of computer system is regarded as one of the most effective methods. Some universities have set up hardware labs equipped with several or dozens of FPGA (Field Programmable Gate Array) boards for these courses. However, these lab kits are always in a relatively low utilization rate and how the students' capability is improved by these assets is not easy to be evaluated. Inspired by the merits of FPGA public cloud (e.g. Amazon AWS F1 instance), an in-house-designed FPGA-based online cloud platform (named ZyForce) is proposed to deploy in UCAS. This platform is equipped with 40 custom designed boards using Xilinx Zynq UltraScale+ MPSoC FPGAs and the utilization rate of these education resources is boosted by means of advanced cloud computing technology. With ZyForce, students remotely carry out lab assignments (e.g. MIPS, RISC-V or domain-specific architecture processor design with cache/memory, DMA, accelerator and performance counter) as using local FPGA boards; instructors can analyze the downloaded operation log file for each student and know how these kits are being used. It's believed that this kind of online hardware lab appliances provides a novel pay-as-you-go service model for those universities in remote regions who cannot afford to set up their own hardware laboratories, and also facilitates our students, the future scientists and engineers, with this promising cloud development approach.

Keywords: Experimental Curriculum; Hardware Platform; FPGA Cloud; Online Learning; Virtual Lab

DOI: https://doi.org/10.1145/3159450.3162241

Compiler Practice System Integrated with Real Open Source Compiler

Yu Zhang, *University of Science and Technology of China*
Contact: yuzhang@ustc.edu.cn

The rapidly growing scale of modern computer systems has been increasing the skill gaps between graduating students and industry expectations. The Compiler Course, as one of the core CS courses, is not only a course introducing the theory and practice of programming language translation, but also a comprehensive course cultivating students' multidimensional competencies, such as programming, language design, software engineering, communication and collaboration, etc. Writing a compiler for a toy language is a common assignment in many compiler courses. Yet our compiler practice system differs from most of its peers in several aspects: integrated with real open source LLVM compiler, practice of some modern compilation mechanisms, process control and version management (using git), team work, etc. We designed two kinds of projects to integrate the LLVM compiler, one is coding class, e.g. developing an LLVM IR generator and a JIT-compiler on LLVM IR; the other is source code understanding class, e.g. providing guidance and issues for understanding the mechanisms of Clang parser or static analyzer. We also designed some team projects to let students investigate some modern language features and their implementation mechanisms, to discuss within and among teams, and finally to give team presentation. The poster describes the components of our compiler practice system, related practice support package and guidance. Some tradeoffs among difficulty, complexity, time and knowledge points are discussed. So far we have been practiced and improved the practice system for 4 years, and lessons as well as experiences are shared on the poster as well.

This work is supported by the 2016 Anhui provincial major teaching and research projects (No.2016jyxm1134, 2016jyxm1135).

Keywords: computer science education; compiler projects; LLVM; open source

DOI: http://dx.doi.org/10.1145/3159450.3162242

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.
SIGCSE'18, February 21-24, 2018, Baltimore, MD, USA.
© 2018 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-5103-4/18/02.

ZyCube: An In-House Mini-Cluster for Agilely Developing and Conducting Computer Systems Course Projects

Ke Zhang, Yisong Chang, Mingyu Chen, Yungang Bao, Zhiwei Xu
SKLCA, ICT, Chinese Academy of Sciences; University of Chinese Academy of Sciences (UCAS)
{zhangke,changyisong,cmy,baoyg,zxu}@ict.ac.cn

ABSTRACT

In the new era of AI and IoT, it's crucial to guide students to learn principles of computer systems design with both hardware (HW) and software (SW), or at least equip them with a HW-SW co-design concept. To overcome the lack of hardware reconfigurability in conventionally leveraged single-board computers (SBCs, e.g., Raspberry Pi) in class, this poster delivers ZyCube, a heterogeneous 4-node SoC-FPGA mini-cluster. These four nodes are orchestrated in a manner of on-demand network service to support a dozen of students to agilely and simultaneously conduct computer systems projects. Three capacity-building course projects are shown with ZyCube, which are (1) HW design of a simplified CPU core with performance profiling using C-language microbenchmarks, (2) rapid co-design of a deep neural network inference accelerator with the CPU core, and (3) datapath optimization with a co-designed DMA engine. These projects are believed to help students to obtain a comprehensive perspective of HW-SW coordination and teachers to leverage a similar platform and pedagogy in their courses.

KEYWORDS

FPGA cluster, Computer system, HW-SW co-design

ACM Reference Format:
Ke Zhang, Yisong Chang, Mingyu Chen, Yungang Bao, Zhiwei Xu. 2019. ZyCube: An In-House Mini-Cluster for Agilely Developing and Conducting Computer Systems Course Projects. In *ACM Global Computing Education Conference 2019 (CompEd'19), May 17–19, 2019, Chengdu,Sichuan, China.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3300115.3312515

1 INTRODUCTION

It is an important mission for computer science (CS) educators and instructors to teach students' thinking and practicing capabilities in computer systems and HW-SW co-design. Computer systems experimental curricula and educational tools are of significance in this procedure. Low-cost SBCs are on trial in CS courses. However, this kind of SBCs lacks of hardware reconfigurability and agility, which are of essential necessity for emerging domain-specific acceleration in AI and IoT. Applying FPGA-based platforms [1, 2] in computer systems courses is an alternative approach to provide undergrads with hands-on HW development environments.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CompEd'19, May 17–19, 2019, Chengdu,Sichuan, China
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6259-7/19/05.
https://doi.org/10.1145/3300115.3312515

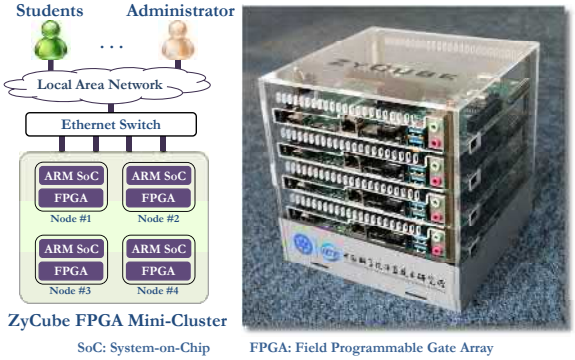


Figure 1: A four-node ZyCube that consist of high-end SoCs, FPGA fabrics and plentiful I/O resources.

2 ZYCUBE: AT A GLANCE

This poster shows an in-house experimental mini-cluster, ZyCube, based on Zynq MPSoC boards used in [3] , which could be useful for students to agilely conduct HW-SW co-design experiments, and easy for other educators to re-design. ZyCube that is composed of four heterogeneous FPGA nodes provides network accessibility for students to simultaneously conduct their experimental projects in a computer lab (Fig. 1). An easy-to-use interface is also built upon the SoC for FPGA configuration and program execution. This in-house appliance will be greatly effective in cultivating students' capabilities in several existing CS courses such as computer organization, computer architecture and operating system.

3 POTENTIAL USERS OF ZYCUBE

Potential users of ZyCube mainly include: (1) educators and instructors; (2) developers interested in creating an agile educaitonal online HW-SW co-design environment; (3) students aiming to improve capabilities in computer systems design and HW-SW programming.

4 ACKNOWLEDGEMENT

This work is partly supported by the National Key Technologies R&D Program of China (2016YFB1000401) and NSF China (61702485).

REFERENCES

[1] M. J. Jipping, S. Henry, K. Ludewig, and L. Tableman. 2006. How To Integrate FPGAs Into a Computer Organization Course. In *Proc. 37th SIGCSE Technical Symposium on Computer Science Education.* 234–238.
[2] D. B. Larkins, W. M. Jones, and H. E. Rickard. 2013. Using FPGAs as a Reconfigurable Teaching Tool Throughout CS Systems Curriculum. In *Proc. 44th SIGCSE Technical Symposium on Computer Science Education.* 397–402.
[3] K. Zhang, Y. Chang, M. Chen, Y. Bao, and Z. Xu. 2019. Computer Organization and Design Course with FPGA Cloud. In *Proc. 50th SIGCSE Technical Symposium on Computer Science Education.* 927–933.

Birds of a Feather

Introductory Concurrency and Parallelism Education

Nasser Giacaman, *University of Auckland*
Joel Adams, *Calvin College*
Contact: n.giacaman@auckland.ac.nz

Undergraduate or novice programmers are often challenged by higher-level and abstract concepts in programming courses. Compared to constructing a sequential program, parallel and concurrent programming requires a different and more complex mental model of control flow. Now that multi-core processors have become the norm for computers and mobile devices, the responsibility of developing software to take advantage of this extra computing power now rests with the modern software developer. In recognition of this new era, curricula guidelines have been proposed specifically targeting the complex world of parallel and distributed computing. CS2013 also recognizes this with a dedicated Parallel and Distributed Computing knowledge area with core hours, as well as dispersing parallelism concepts across other fundamental knowledge areas. Parallel programming was once considered an advanced area of computing, and only taught to students by experts in graduate-level elective courses. However, *it is now expected that all undergraduate computing students will become familiar with the fundamentals of parallelism.* Concurrency and parallelism concepts are undoubtedly difficult for students to learn. This can even be daunting for teachers that are inexperienced with all elements of the underlying parallelism concepts, but even more daunting is devising pedagogically-sound materials that will allow undergraduate students to grasp the concepts. This is especially challenging for early undergraduate courses where students are often novice programmers, barely confident in sequential programming let alone parallel programming. This session will provide an opportunity for instructors to discuss and share ideas and experiences in this area, as well as explore potential collaboration opportunities.

Keywords: Concurrency; Parallel Programming; Multi-Core; Multi-Threading

DOI: https://doi.org/10.1145/3300115.3312505

Evolution and Revolution of Computer Systems Courses with the Open RISC-V ISA

Ke Zhang,
Institute of Computing Technology of Chinese Academy of Sciences; University of Chinese Academy of Sciences
Contact: zhangke@ict.ac.cn

Computer systems courses, including computer organization, architecture, operating system, compiler, and other relevant courses, play an important role in cultivating students' computational thinking, especially the ability of viewing hardware and software in a computer system as a whole. In order to help learners to better understand principles and concepts taught in lectures of aforementioned courses, some visionary instructors have started to design lab projects for their courses with real-world instruction set architectures (ISAs), such as MIPS, ARM and x86, or even request students to conduct projects on these ISAs-based hardware platforms. Meanwhile, with the overwhelming trend towards AI, IoT and heterogeneous computing, RISC-V, an open and free ISA, has emerged and recently attracted a lot of attention from both academia and industry. Compared with proprietary ISAs, RISC-V is more simple, affordable, extensible and compatible. Given this context, several arising questions can be examined and discussed in this BoF session. For example, 1) whether the computer systems courses should integrate RISC-V, 2) do we need minor (evolutionary) or major (revolutionary) curriculum changes with the involvement of RISC-V for each course or the entire CS curriculum, 3) how to help the majority of students accommodate to these changes, 4) what kind of new course materials should be provided to instructors, TAs and students, 5) whether it's possible for some senior students to tape-out their own RISC-V chip designs and finally produce a cornucopia of workable computing systems with this revolutionary and open curriculum.

Keywords: Computer System; Lab Project; Open Source; Hardware-software co-design; RISC-V

DOI: https://doi.org/10.1145/3300115.3312506

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CompEd '19, May 17–19, 2019, Chengdu, Sichuan, China.
© 2019 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-6259-7/19/05.

EdUCAS: An In-house CI/CD Platform with Cloud FPGAs for Agilely Conducting Computer Systems Course Projects

Ke Zhang, Yisong Chang, Ran Zhao, Zelin Wang, Mingyu Chen, Yungang Bao, Zhiwei Xu
SKLCA, ICT, Chinese Academy of Sciences (CAS) and University of Chinese Academy of Sciences (UCAS)
Beijing, China
{zhangke,changyisong,zhaoran,wangzelin20s,cmy,baoyg,zxu}@ict.ac.cn

ABSTRACT

In recent years, there has been a rapidly growing recognition of the importance of conducting hands-on HW-SW co-design labs with real hardware (e.g., programmable logic chips named FPGAs) while studying computing curricula [1], especially the computer systems (CSys) courses. However, using FPGA is quite a time-consuming and error-prone process for students, and manipulating FPGA development tools and boards also distracts students and instructors. To overcome these obstacles and improve agility, we introduce an in-house platform named EdUCAS, in combination with the previously designed cloud FPGA servers for students to automatically conduct computer systems course projects. EdUCAS aims at enabling students to concentrate on their logic designs using hardware description language (e.g., Verilog HDL), without wasting useless time in FPGA tools and experimental environment.

KEYWORDS

Computer systems course; FPGA development; CI/CD; Course labs

1 ARCHITECHTURE OF EDUCAS

An FPGA cloud for computer organization course was previously designed [2], in which FPGA resources were remotely accessed by students and instructors. However, students were required to download, install and use virtual machines (VMs) with complicated FPGA development tools on their local laptops. In order to release students from such heavy VMs, EdUCAS is proposed to leverage an x86 cluster as a continuous integration (CI) platform for automatic remote FPGA development and as a continuous deployment (CD) manager for automatically dispatching FPGA evaluation workloads among multiple FPGAs as shown in Figure 1.

Based on Git and our refactored FPGA-compatible CI flow in EduCAS, students agilely conduct CSys projects using lightweight laptops with just a Git push operation. After one student pushed his or her latest logic design files into a project repository, an instance

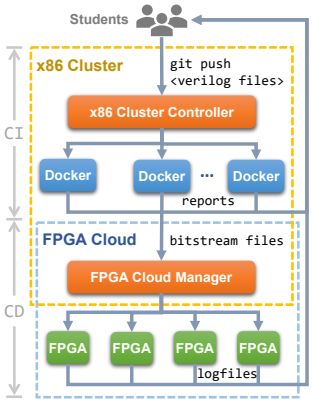


Figure 1: Architecture and flow chart of EdUCAS.

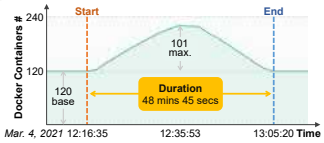


Figure 2: The number of active containers during test.

(e.g., Docker) of our pre-defined container image would be automatically launched in the cloud by a custom controller to execute each stage of the conventional FPGA design flow. Meanwhile, we design an FPGA cloud manager in the x86 cluster to continuously deploy CI-generated configuration files as well as evaluation tasks to cloud FPGAs via intranet. In addition, sufficient results and reports are collected in each CI/CD step and delivered to students for facilitating agile and flexible optimizations of their project designs.

2 PRELIMINARY RESULT

We have conducted a series of pressure tests in an EdUCAS prototype that contains four cloud FPGAs and nine x86 virtual machines (16 vCPUs and 64GiB memory for each). Figure 2 depicts a representative test during 12:16:35 - 13:05:20, Mar. 4th, 2021. In such a short period, up to 100 workloads packed in separated containers were launched sequentially. For fault tolerance, one of the workloads was executed twice. There were also 120 supportive containers running continuously in background. The result indicates that our experimental EdUCAS platform is able to complete project workloads submitted simultaneously by more than 100 students within 48'45".

[1]This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDC05030400, by the Proving Ground for Information Superbahn at CAS Innovation Academy for Network and Computing Technology, by the National Natural Science Foundation of China under Grant No. 61702485, and by the Youth Innovation Promotion Association of CAS under Grant No. 2017143.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
ITiCSE'21, June 21–22, 2021, Paderborn, Germany
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8397-4/21/06.
https://doi.org/10.1145/3456565.3460060

REFERENCES

[1] João M. P. Cardoso. 2005. New Challenges in Computer Science Education. In *Proc. 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Educationi (ITiCSE)*. 203–207.
[2] K. Zhang, Y. Chang, M. Chen, Y. Bao, and Z. Xu. 2019. Computer Organization and Design Course with FPGA Cloud. In *Proc. 50th ACM Technical Symposium on Computer Science Education*. 927–933.

Preliminary exploration of a competency model for open-source processor chip design talent

Xiaochun Yang
Shanghai AchieveFun Info Tech Co., Ltd.,
Shanghai, China
jane@janeyoungtech.com

Ke Zhang
Institute of Computing Technology of Chinese Academy of Sciences,
University of Chinese Academy of Sciences,
Beijing, China
zhangke@ict.ac.cn

Zihao Yu
Institute of Computing Technology of Chinese Academy of Sciences,
University of Chinese Academy of Sciences,
Beijing, China
yuzihao@ict.ac.cn

ABSTRACT

To support the cultivation of open source processor chip design talents and to provide a high-quality talent development and assessment model for universities to assess students of the "Computing System and Processor" course and for enterprises to recruit graduates, professionals from enterprises and teachers from universities have collaborated to design an open source processor chip design talent competency model based on the CC2020 competency model, science and engineering practice theory and curriculum theory, in order to provide relatively reasonable, comprehensive and accurate reference information for talent training and education teaching. This poster designed a competency model that includes statement, computing and professional knowledge-skills pairs and dispositions. The summary results of the visual assessment tool designed on the basis of the competency model are presented, including an initial description of student characteristics, a comprehensive task-based assessment conclusion, as well as a detailed analysis of the dimensions of computing and professional knowledge-skill pairs and dispositions and corresponding suggestions. The authors will continue to refine the quantification of student learning behavior markers to provide easy-to-operate reference data for education and teaching and to enhance the usability of the model.

CCS CONCEPTS

• **Social and professional topics**; • **Professional topics**; • **Computing education**; • **Student assessment**;

KEYWORDS

Open-Source Processor Chip Design, Competency Model, Engineering Practice, Behavioral Marker

ACM Reference Format:
Xiaochun Yang, Ke Zhang, and Zihao Yu. 2023. Preliminary exploration of a competency model for open-source processor chip design talent. In *ACM Turing Award Celebration Conference 2023 (ACM TURC '23), July 28–30, 2023, Wuhan, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3603165.3607456

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
ACM TURC '23, July 28–30, 2023, Wuhan, China
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0233-4/23/07.
https://doi.org/10.1145/3603165.3607456

1 PROBLEM STATEMENT

To meet the requirements of the current chip industry trends and talent pool, the development of open-source processor chip design talent has become an urgent endeavor. A competency model needs to be adopted on how to develop the knowledge, skills, and dispositions that students will need in their future jobs through the process of engineering practice during the teaching of university courses. A similar research work is detailed in [1].

2 BACKGROUND AND RELATED WORK

The competency model developed based on the CC2020 competency model [2], engineering practice theory [3] and curriculum theory [4] is described below.

1. Competency Statement: Ability to apply chip agile design methodologies, use EDA tool procedure, design open source processor chips, evaluate processor chips using open source EDA tools, enhance chip designs through computer architecture optimization techniques, and complete chip manufacturing through tape-out.
2. Knowledge-skills pairs

The knowledge component of the model includes both computing knowledge and professional knowledge as shown in Table 1 and Table 2.

3. Dispositions are shown in Table 3

3 METHODS AND RESULTS

Based on the definition of the model, visual and easy-to-use assessment tools are designed in conjunction with course theory, and Frequently Asked Questions(FAQs) are provided on general knowledge of the model and how to use it. Teachers can use the assessment tools to output initial conclusions about student characteristics(such as being good at research, engineering, or maintenance), comprehensive assessments for completing each practicing task and detailed analyses and suggestions of various dimensions of computer knowledge and professional knowledge-skill pairs and disposition.

4 CONTRIBUTIONS AND FUTURE WORK

The competency model proposed in this paper features a detailed behavioral marker [5] based on engineering practice theory to provide an informative assessment of talent development. The current model can give some initial suggestions. The authors will continue to refine the quantification of student learning behaviors based on

Table 1: Computing Knowledge-Skills pairs

Knowledge	Skills	Student activities
Processor Chip Design and Agile Development	Applying	Engineering Practice
Basic design and implementation of processor chips	Creating	Engineering Practice and Assessment
Processor performance optimization techniques	Analyzing	Engineering Practice
Physical design and sign-off of chips	Applying	Engineering Practice

Table 2: Professional Knowledge-Skills pairs

Knowledge	Skills	Student activities
Analytical and Critical Thinking	Applying	Engineering Practice, Coding Assessment and Defense
Problem-Solving and Trouble Shooting	Applying	Engineering Practice and Assessment
Research and self-Starter/Learner	Analyzing	Engineering Practice
Quality Assurance/Control	Applying	Engineering Practice
Collaboration and Teamwork	Applying	Engineering Practice

Table 3: Elements of Dispositions and Behavioral Marker [5]

Element	Elaboration	Behavioral Marker
Adaptable	Flexible; agile, adjust in response to change	Actively learn and apply new knowledge
Inventive	Exploratory. Look beyond simple solutions	Seeking solutions based on unknown problems and existing knowledge
Meticulous	Attentive to detail; thoroughness, accurate	Focuses on completeness in analysis and reflection, and accuracy in presentation and description
Self-directed	Self-motivated, determination, independent	Active thinking and practice, Not relying on the help of others
Passionate	Conviction, strong commitment, compelling	Consistently completes difficult tasks with convincing behavior and results

curriculum theory and behavioral research to provide easily actionable reference data for education research and teaching process, and to improve the usability of the model.

ACKNOWLEDGMENTS

The work is supported by the Virtual Teaching and Research Section of the Ministry of Education of P. R. China for the Computer System and Processor Courses.

REFERENCES

[1] Ana Hol, Joan Richardson, James McGovern, and Margaret Hamilton. 2023. A New Sustainable Model for Aligning Industry Requirements and University Programs. *ACM Inroads* 14, 1 (March 2023), 30–39. DOI: https://doi.org/10.1145/3583086

[2] ACM and IEEE-CS. 2021. ACM and IEEE-CS Release Computing Curricula 2020. Retrieved from https://www.acm.org/media-center/2021/march/computing-curricula-2020.

[3] National Research Council Report. 2012.*A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. National Academies Press, Washington, D. C..

[4] ScienceDirect. Curriculum Theory. Retrieved from https://www.sciencedirect.com/topics/social-sciences/curriculum-theory.

[5] Lacey Schmidt (NASA Mission Operations Directorate, ITCB HBP Training Working Group). 2013.International Space Station Human Behavior and Performance Competency Model: Volume I. Retrieved from https://ntrs.nasa.gov/citations/20080018552.

To Promote the Progress *of Science and Useful Arts*

The Director

of the United States Patent and Trademark Office has received an application for a patent for a new and useful invention. The title and description of the invention are enclosed. The requirements of law have been complied with, and it has been determined that a patent on the invention shall be granted under the law.

Therefore, this United States

Patent

grants to the person(s) having title to this patent the right to exclude others from making, using, offering for sale, or selling the invention throughout the United States of America or importing the invention into the United States of America, and if the invention is a process, of the right to exclude others from using, offering for sale or selling throughout the United States of America, products made by that process, for the term set forth in 35 U.S.C. 154(a)(2) or (c)(1), subject to the payment of maintenance fees as provided by 35 U.S.C. 41(b). See the Maintenance Fee Notice on the inside of the cover.

Katherine Kelly Vidal

DIRECTOR OF THE UNITED STATES PATENT AND TRADEMARK OFFICE

Maintenance Fee Notice

If the application for this patent was filed on or after December 12, 1980, maintenance fees are due three years and six months, seven years and six months, and eleven years and six months after the date of this grant, or within a grace period of six months thereafter upon payment of a surcharge as provided by law. The amount, number and timing of the maintenance fees required may be changed by law or regulation. Unless payment of the applicable maintenance fee is received in the United States Patent and Trademark Office on or before the date the fee is due or within a grace period of six months thereafter, the patent will expire as of the end of such grace period.

Patent Term Notice

If the application for this patent was filed on or after June 8, 1995, the term of this patent begins on the date on which this patent issues and ends twenty years from the filing date of the application or, if the application contains a specific reference to an earlier filed application or applications under 35 U.S.C. 120, 121, 365(c), or 386(c), twenty years from the filing date of the earliest such application ("the twenty-year term"), subject to the payment of maintenance fees as provided by 35 U.S.C. 41(b), and any extension as provided by 35 U.S.C. 154(b) or 156 or any disclaimer under 35 U.S.C. 253.

If this application was filed prior to June 8, 1995, the term of this patent begins on the date on which this patent issues and ends on the later of seventeen years from the date of the grant of this patent or the twenty-year term set forth above for patents resulting from applications filed on or after June 8, 1995, subject to the payment of maintenance fees as provided by 35 U.S.C. 41(b) and any extension as provided by 35 U.S.C. 156 or any disclaimer under 35 U.S.C. 253.

US011841733B2

(12) **United States Patent**
Zhang et al.

(10) **Patent No.: US 11,841,733 B2**
(45) **Date of Patent: Dec. 12, 2023**

(54) **METHOD AND SYSTEM FOR REALIZING FPGA SERVER**

(71) Applicant: **Institute of Computing Technology, Chinese Academy of Sciences**, Beijing (CN)

(72) Inventors: **Ke Zhang**, Beijing (CN); **Yazhou Wang**, Beijing (CN); **Mingyu Chen**, Beijing (CN); **Yisong Chang**, Beijing (CN); **Ran Zhao**, Beijing (CN); **Yungang Bao**, Beijing (CN)

(73) Assignee: **Institute of Computing Technology, Chinese Academy of Sciences**, Beijing (CN)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **17/791,511**

(22) PCT Filed: **Jan. 8, 2020**

(86) PCT No.: **PCT/CN2020/070974**
§ 371 (c)(1),
(2) Date: **Jul. 7, 2022**

(87) PCT Pub. No.: **WO2021/138846**
PCT Pub. Date: **Jul. 15, 2021**

(65) **Prior Publication Data**

US 2023/0101208 A1 Mar. 30, 2023

(51) **Int. Cl.**
G06F 13/42 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 13/4282*** (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,760,898 B1 * 7/2004 Sanchez G01R 31/318519 716/111
7,814,255 B1 * 10/2010 Deva G06F 13/4081 710/10

(Continued)

FOREIGN PATENT DOCUMENTS

CN 102693141 A 9/2012
CN 103678248 A 3/2014

(Continued)

OTHER PUBLICATIONS

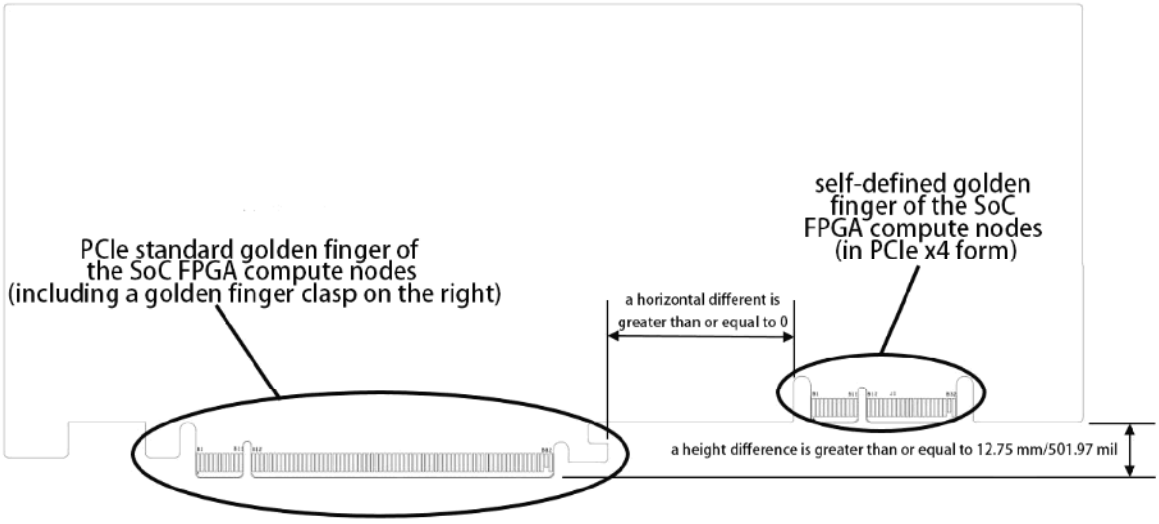
International Search Report for International Application No. PCT/CN2020/070974 dated Sep. 25, 2020.

Primary Examiner — Henry Tsai
Assistant Examiner — Christopher A Bartels
(74) *Attorney, Agent, or Firm* — MASCHOFF BRENNAN

(57) **ABSTRACT**

A method and system for realizing a FPGA server, wherein centralized monitoring and managing all SoC FPGA compute nodes within the server by a motherboard, the motherboard comprising: a plurality of self-defined management interfaces for connecting the SoC FPGA compute nodes to supply power and data switch to the SoC FPGA compute nodes; a management network switch module for interconnecting the SoC FPGA compute nodes and supplying management; and a core control unit for managing the SoC FPGA compute nodes through the self-defined management interfaces and a self-defined management interface protocol, and acquiring operating parameters of the SoC FPGA compute nodes to manage and monitor the SoC FPGA compute nodes based on the management interface protocol.

8 Claims, 3 Drawing Sheets



证书号第4761050号

发明专利证书

发明名称：一种实现FPGA服务器的方法和系统

发明人：张科;于磊;王亚洲;陈明宇;常轶松;赵然;包云岗

专利号：ZL 2020 1 0019013.9

专利申请日：2020年01月08日

专利权人：中国科学院计算技术研究所

地址：100080 北京市海淀区中关村科学院南路6号

授权公告日：2021年10月29日　　授权公告号：CN 111258951 B

国家知识产权局依照中华人民共和国专利法进行审查，决定授予专利权，颁发发明专利证书并在专利登记簿上予以登记。专利权自授权公告之日起生效。专利权期限为二十年，自申请日起算。

专利证书记载专利权登记时的法律状况。专利权的转移、质押、无效、终止、恢复和专利权人的姓名或名称、国籍、地址变更等事项记载在专利登记簿上。

局长
申长雨

国家知识产权局
2021年10月29日

其他事项参见续页

证 书 号 第 4761050 号

国家税务总局北京市海淀区税务局
印 花 税
代扣专用章
国家知识产权局

专利权人应当依照专利法及其实施细则规定缴纳年费。本专利的年费应当在每年 01 月 08 日前缴纳。未按照规定缴纳年费的，专利权自应当缴纳年费期满之日起终止。

申请日时本专利记载的申请人、发明人信息如下:

申请人:

中国科学院计算技术研究所

发明人:

张科; 于磊; 王亚洲; 陈明宇; 常轶松; 赵然; 包云岗

证书号第7710519号

专利公告信息

发明专利证书

发 明 名 称：一种面向SSI器件的FPGA持续集成开发方法与系统

专 利 权 人：中国科学院计算技术研究所

地　　址：100190 北京市海淀区中关村科学院南路6号

发　明　人：张科;齐乐;王泽霖;赵然

专　利　号：ZL 2021 1 1156001.1　　授权公告号：CN 113887164 B

专利申请日：2021年09月30日　　授权公告日：2025年02月07日

申请日时申请人：中国科学院计算技术研究所

申请日时发明人：张科;齐乐;王泽霖;赵然

国家知识产权局依照中华人民共和国专利法进行审查，决定授予专利权，并予以公告。

专利权自授权公告之日起生效。专利权有效性及专利权人变更等法律信息以专利登记簿记载为准。

局长
申长雨

国家知识产权局

2025年02月07日

证书号第7855728号

专利公告信息

发明专利证书

发 明 名 称：一种面向处理器系统级设计的测试方法及系统

专 利 权 人：中国科学院计算技术研究所

地　　址：100190 北京市海淀区中关村科学院南路6号

发　明　人：常轶松;陈欲晓;张科;张旭;齐乐;陈明宇;包云岗

专　利　号：ZL 2023 1 0555661.X　　授权公告号：CN 117149536 B

专利申请日：2023年05月17日　　授权公告日：2025年04月04日

申请日时申请人：中国科学院计算技术研究所

申请日时发明人：常轶松;陈欲晓;张科;张旭;齐乐;陈明宇;包云岗

国家知识产权局依照中华人民共和国专利法进行审查，决定授予专利权，并予以公告。

专利权自授权公告之日起生效。专利权有效性及专利权人变更等法律信息以专利登记簿记载为准。

局长
申长雨

国家知识产权局
2025年04月04日

证书号第7851984号

专利公告信息

发明专利证书

发 明 名 称：一种云原生的硬件逻辑仿真FPGA加速方法及系统

专 利 权 人：中国科学院计算技术研究所

地　　址：100080 北京市海淀区中关村科学院南路6号

发　明　人：常轶松;陈欲晓;张科;陈明宇;包云岗

专　利　号：ZL 2022 1 1667431.4　　授权公告号：CN 116029242 B

专利申请日：2022年12月23日　　授权公告日：2025年04月04日

申请日时申请人：中国科学院计算技术研究所

申请日时发明人：常轶松;陈欲晓;张科;陈明宇;包云岗

国家知识产权局依照中华人民共和国专利法进行审查，决定授予专利权，并予以公告。

专利权自授权公告之日起生效。专利权有效性及专利权人变更等法律信息以专利登记簿记载为准。

局长

申长雨

国家知识产权局

2025年04月04日

证书号第7686262号

专利公告信息

发明专利证书

发 明 名 称：一种在虚拟机中进行FPGA动态部分可重构配置方法和系统、存储介质、电子设备

专 利 权 人：中国科学院计算技术研究所

地　　址：100080 北京市海淀区中关村科学院南路6号

发　明　人：常轶松;阮金杰;张科;齐乐

专　利　号：ZL 2022 1 0569933.7　　授权公告号：CN 114968487 B

专利申请日：2022年05月24日　　授权公告日：2025年01月24日

申请日时申请人：中国科学院计算技术研究所

申请日时发明人：常轶松;阮金杰;张科;齐乐

国家知识产权局依照中华人民共和国专利法进行审查，决定授予专利权，并予以公告。

专利权自授权公告之日起生效。专利权有效性及专利权人变更等法律信息以专利登记簿记载为准。

局长
申长雨

国家知识产权局

2025年01月24日

证书号第14506049号

实用新型专利证书

实用新型名称：一种抽屉式高密度FPGA云平台机箱

发　明　人：张科;于磊;王亚洲;常轶松;赵然;陈明宇

专　利　号：ZL 2020 2 3019916.1

专利申请日：2020年12月15日

专 利 权 人：中国科学院计算技术研究所

地　　址：100080 北京市海淀区中关村科学院南路6号

授权公告日：2021年10月29日　　授权公告号：CN 214545185 U

国家知识产权局依照中华人民共和国专利法经过初步审查，决定授予专利权，颁发实用新型专利证书并在专利登记簿上予以登记。专利权自授权公告之日起生效。专利权期限为十年，自申请日起算。

专利证书记载专利权登记时的法律状况。专利权的转移、质押、无效、终止、恢复和专利权人的姓名或名称、国籍、地址变更等事项记载在专利登记簿上。

局长
申长雨

国家知识产权局
2021年10月29日

其他事项参见续页

证书号第14506049号

国家税务总局北京市海淀区税务局
印花税
代扣专用章
国家知识产权局

专利权人应当依照专利法及其实施细则规定缴纳年费。本专利的年费应当在每年12月15日前缴纳。未按照规定缴纳年费的，专利权自应当缴纳年费期满之日起终止。

申请日时本专利记载的申请人、发明人信息如下:

申请人:

中国科学院计算技术研究所

发明人:

张科; 于磊; 王亚洲; 常轶松; 赵然; 陈明宇

中华人民共和国国家版权局

计算机软件著作权登记证书

证书号： 软著登字第8207426号

软 件 名 称： 处理器芯片敏捷设计开发云服务平台
[简称：SERVE]
V1.0

著 作 权 人： 中国科学院计算技术研究所

开发完成日期： 2021年03月11日

首次发表日期： 2021年06月26日

权利取得方式： 原始取得

权 利 范 围： 全部权利

登 记 号： 2021SR1484800

根据《计算机软件保护条例》和《计算机软件著作权登记办法》的规定，经中国版权保护中心审核，对以上事项予以登记。

中华人民共和国国家版权局
计算机软件著作权
登记专用章

No. 09022676

2021年10月11日

中国科学院大学《计算机组成原理（研讨课）》及处理器芯片与计算机系统虚拟仿真实验服务平台评价意见

2023 年 8 月 28 日-29 日，郑州大学计算机与人工智能学院、软件学院和郑州大学国家超级计算郑州中心的相关老师和研究生们参加了由中国科学院大学教学团队组织的《计算机组成原理（研讨课）》师资培训，并在培训后使用课程研发的“处理器芯片与计算机系统”虚拟仿真实验服务平台进行实践学习。

本课程教学内容和虚拟仿真实验服务平台得到了参与课程培训和平台试用的师生的一致认可。基于课程研发的虚拟仿真实验平台，通过较为完整的实验项目，可使学生较为全面地掌握计算机系统与处理器芯片设计、开发与调试的完整流程与知识技能，有助于塑造和构建学生的计算机系统思维。

我校师生十分感谢中国科学院大学《计算机组成原理（研讨课）》教学团队介绍的实验教学内容与实验方法，并提供虚拟仿真实验平台供师生进行处理器芯片与计算机系统教学实践。使用中国科学院大学的上述教研成果，加速助力了我校计算机系统及处理器芯片方向拔尖人才培养，极大扩展了我校在相关领域教学的思路和手段。

郑州大学计算机与人工智能学院、软件学院

2023 年 10 月 31 日

财经观察

拓展智能工厂、推广工业互联网平台

数字技术加力赋能实体产业

加快研制一批建材重点细分领域智能装备、智能工厂、智能服务等关键标准；开展钢铁行业数字化转型三年行动；发布实施石化化工行业智能制造标准体系建设指南……近日，工业和信息化部联合相关部门发布钢铁、有色金属、建材、石化化工等行业稳增长方案，加快数字化升级是其中重点。

工业和信息化部有关负责人表示，下一步，将全面推动制造业数字化普及，激发实体产业发展的新动能。

大屏上，生产画面、各类数据实时显示；车间里，系统智能推出优化配料方案……在山东九羊集团，通过浪潮云洲工业互联网平台智能炼钢应用，加料、调温等工序可以自动精准完成。企业有关负责人介绍，借助数字技术，企业实现智能炼钢全流程能力与效益的提升。通过智能化改造，进一步提高炼钢操作标准化水平，降低冶炼时间，实现节能减排。

我国持续加大力度推动数字技术在实体经济领域的融合应用。今年以来，面对需求收缩等多重压力，大量企业特别是传统行业制造业企业通过数字化应用提质增效。

这是智慧矿山的实践：工作人员坐在智能车间中控室里，双手熟练地操作着控制台，用摇杆和按钮完成放矿作业，监视屏里的矿车抵达指定的位置。金川集团有关负责人介绍，企业已实现在5G高速通信模式下，电机车的无人驾驶和智能调度，人员操作环境得到明显改善。

这是智能工厂的延展：在中国一汽的红旗繁荣工厂，集成机器人集群有条不紊，超声波设备精准质检，通过数据传感，生产线上的每个细节都会“搬”到虚拟“孪生工厂”，让制造提质增效，为创新提供支撑。

汽车、大飞机、工程机械等装备制造业探索协同设计、虚拟验证、远程运维等模式；家电、服装等消费品行业创新大规模定制、用户直连制造、共享制造等模式……工业和信息化部数据显示，当前，5G应用加速向重点领域拓展深化，应用案例数累计超5万个，具有一定影响力的工业互联网平台超过240家。各地建设数字化车间和智能工厂近8000个。智能制造新场景、新方案、新模式不断涌现。

近期，一系列支持数字技术与实体产业融合的举措接连推出——

聚焦石化化工领域，建设智能制造示范工厂，培育重点行业特色型工业互联网平台，支持地方创建智慧化工园区；

针对钢铁行业，探索打造超级能效工厂，鼓励有条件的地方建立数字化转型产业联盟，建设钢铁工业互联网平台和大数据中心；

在装备制造领域，以试点示范和先行区建设为引领，系统深入推进智能制造，加快推进装备数字化发展……

分业施策推进数字化、智能化应用的同时，着眼于夯实数字根基的措施出台。

工业和信息化部新闻发言人赵志国说，将坚持适度超前，积极推进5G网络建设，开展“信号升格”专项行动，全面提升重点场景5G网络质量。打造“5G+工业互联网”发展升级版，今年推动不少于3000家企业建设5G工厂。大力推动5G轻量化技术研发，促进5G应用持续降成本、上规模。

记者从工业和信息化部了解到，工业和信息化部将着眼于推动企业上云，进一步降低数字化门槛。

工业和信息化部有关负责人表示，将以智能制造为主攻方向，持续深入推进场景模式的推广、解决方案攻关等重点工作，全面提升制造业数字化水平，不断增强实体经济发展的新动能。

新华社北京8月30日电

“深港一卡通”正式发行

持卡可畅行香港及内地327个城市

据新华社深圳8月30日电（记者 毛思倩）记者从深圳通公司了解到，该公司与八达通卡公司联合发行的深港互通行卡30日正式上线，乘客持此卡可实现香港及内地327个城市公共交通的“一卡通行”。

据介绍，深港互通行卡又称“深港一卡通”，是一张“一卡一芯片双钱包”实体交通卡，港币钱包采用八达通技术标准，人民币钱包采用交通运输部技术标准，可在香港以及内地327个城市公共交通领域刷卡乘车。“深港一卡通”用户在深圳向人民币钱包充值后，可刷卡乘坐深圳公交地铁，乘车费用从深圳通人民币钱包中扣除；在香港向港币钱包充值后，可刷卡乘坐香港公交地铁，乘车费用从八达通港币钱包中扣除，也可以用卡在便利店消费。

目前，深港互通行卡仅在深圳发售，乘客可在深圳通APP、深圳通自营客服网点、深圳通智能微客服终端、深圳邮政支局、部分口岸地铁站客服网点及口岸自助售卡终端等渠道进行购买。

据悉，在深港互通的基础上，深圳通还将辐射至粤港澳大湾区其他城市，为湾区居民提供公交、地铁、跨境巴士、城际铁路无缝对接的一体化MaaS（出行即服务）出行联程联运服务，促进区域深度融合发展，计划在2023年底前与澳门开展互联互通工作。

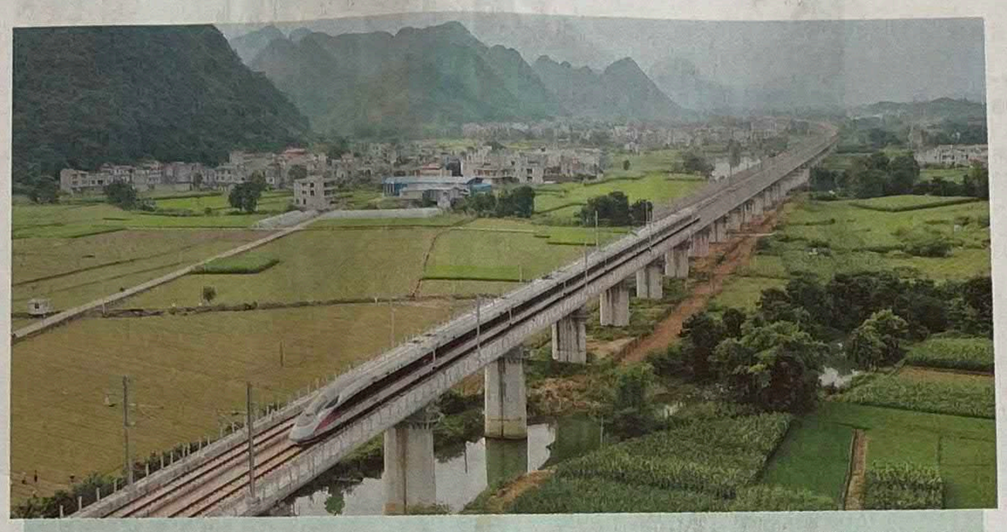
跨境电商出口退运商品税收政策延续实施

新华社北京8月30日电（记者 王雨萧 申铖）记者30日获悉，为支持跨境电商等新业态新模式加快发展，财政部、海关总署、税务总局近日联合发布公告，延续实施跨境电子商务出口退运商品税收政策。

公告规定，对2023年1月30日至2025年12月31日期间在跨境电子商务海关监管代码（1210、9610、9710、9810）项下申报出口，且自出口之日起6个月内因滞销、退货原因原状退运进境的商品（不含食品），免征进口关税和进口环节增值税、消费税；出口时已征收的出口关税准予退还；出口时已征收的增值税、消费税，参照内销货物发生退货有关税收规定执行；其他规定仍按照《财政部 海关总署 税务总局关于跨境电子商务出口退运商品税收政策的公告》（财政部 海关总署 税务总局公告2023年第4号）相关规定执行。

财政部关税司相关负责人介绍，此次发布的公告将企业在跨境电子商务海关监管代码项下申报出口的期限，由原来的2024年1月29日延长至2025年12月31日。上述期限的延长，有利于充分发挥政策效应，进一步稳定企业预期，推动外贸新业态加快发展。

此外，为支持办好中国国际服务贸易交易会，财政部会同海关总署、税务总局30日对外发布通知，明确延续执行服贸会进口展品税收政策，对2024年至2025年期间举办的服贸会，在展期内销售的规定数量或金额以内的进口展品免征进口关税、进口环节增值税和消费税。享受税收政策的展品不包括烟、酒、汽车、列入《进口不予免税的重大技术装备和产品目录》的商品、濒危动植物及其产品，以及国家禁止进口商品。



贵阳至南宁高速铁路今日全线贯通运营

8月30日，新型复兴号动车组列车在贵南高铁广西河池境内试跑（无人机照片）。继新建贵阳至南宁高速铁路贵阳至荔波段8月8日开通运营后，南宁至荔波段将于8月31日开通运营。届时，贵南高铁实现全线贯通运营，我国西南地区新增一条交通大动脉，黔桂两省区间交通出行更加便捷、西南与华南地区联系将更加紧密。新华社发（高东风 摄）

我国绿色贸易规模基本保持增长态势

据新华社济南8月30日电（记者 张武岳 袁敏）29日发布的《中国绿色贸易发展报告2023》显示，近年来我国绿色贸易规模基本保持增长态势，2013年至2022年年均增长3.18%，在全球占比提升2.3个百分点。

《报告》是商务部国际贸易经济合作研究院的最新研究成果之一，发布于在山东烟台举行的2023绿色低碳高质量发展大会期间。《报告》显示，2022年，受全球经济复苏乏力、外需减弱等多重因素影响，中国绿色贸易规模有所下滑，但市场结构呈现出明显多元化趋势，绿色贸易方式持续优化。

《报告》同时指出，当前，经济全球化遭遇逆流，绿色贸易发展面临多重压力。2013年至2022年，世界环境产品进出口规模小幅波动增长，年均增速为0.85%。中国、美国、欧盟一直保持前三的领先地位，且市场份额呈上升趋势，2022年占比达40.4%。

据《报告》介绍，我国绿色贸易发展区域集中度较高。东部沿海地区，绿色贸易活跃。中西部地区绿色贸易占比较低，但增速快，有较大发展潜力。

2023年绿色低碳高质量发展大会期间，除了发布《中国绿色贸易发展报告2023》等一批创新成果外，还安排了2023碳达峰碳中和会议、新旧动能转换项目推介及商务洽谈会、2023清洁能源发展会议，以及绿色低碳高质量发展决策咨询会、绿色低碳高质量发展成果展等多场活动。

郑东新区·“周报”

郑东新区多措并举确保开学平稳有序

本报讯（记者 覃岩峰 通讯员 李盼 冯婕）全市中小学迎来开学季。连日来，郑东新区细化部署、扎实推进、精准落实，全面做好开学各项准备工作，确保秋季学期平稳、有序、顺利开学。

郑东新区教文体局审慎稳妥做好阳光招生，对全区适龄儿童实施“免试入学、就近划片”全覆盖，周密做好教育教学准备，全面提升教育教学质量，深入开展心理健康教育，为学生健康成长保驾护航。同时抓细抓实校园安全管理，切实加强开学工作督导检查，确保师生安全和校园稳定，开学各项工作规范有序、安全顺利。

郑东新区宣传部（文明办）牵头，联合郑州市文化市场综合行政执法支队六大队和祭城路街道办事处开展“护苗”开学季专项检查行动。专项行动重点对聚源路小学、郑州市第七十八中学等学校周边文化用品店、书店、复印店等场所进行检查，重点查看经营店铺是否证照齐全，是否存在侵权盗版等非法出版物、是否有涉黄书册画册，出版物进货渠道是否正规等情况。同时，向各经营店铺负责人宣传“扫黄打非”“护苗·绿书签”等相关知识，提高其自觉守法经营意识，坚决防止违禁、低俗、暴力、侵权盗版等非法出版物危害中小学生身心健康。

郑东新区纪检监察工委深入部分中小学，对秋季开学安全措施落实情况开展监督检查。监督检查过程中，郑东新区纪检监察工委采取实地检查、现场询问、个别访谈、查阅资料等方式，对郑东新区部分中小学校、幼儿园开学安全工作开展随机监督检查，精准查找学校开学准备工作中存在的短板和漏洞。督促校方严格落实安全责任，做到隐患排查+整改双到位。发现个别学校存在学校餐厅地面积水过多、学生宿舍铁架床有安全隐患、厕所卫生不整洁等问题，督查人员现场向学校反馈意见，督促立行立改，及时堵塞短板漏洞。

豫兴路办事处卫健办联合消防安全服务中心、应急办、教文体办深入辖区中小学、幼儿园开展安全检查。检查人员重点对学校的消防安全责任制度是否严格落实、安全疏散通道是否畅通、消防安全标志和应急照明设施是否完好、灭火器和消防栓等设备是否完好、教室和食堂用电用气用火是否安全合理、突发公共卫生事件管理、传染病预防控制管理生活饮用水卫生管理、生活环境卫生管理等各项制度落实情况进行检查。并现场抽查了学校相关负责人及部分教师对消防知识、灭火器使用的掌握情况。针对检查中发现的隐患，检查人员要求学校负责人抓紧落实整改，做到不留隐患、不留死角。

全力纾困解难 助企发展提质

本报讯（记者 覃岩峰 通讯员 李根 杨蕊 陈晓宁 鲁畅畅）郑东新区持续开展“万人助万企”活动，以精准服务、精准施策，全方位帮助企业纾困解难，推动企业发展提质提速。

郑东新区科技人才局和龙源路办事处联合推进“万人助万企”活动，双方加大对创新企业的政策支持与宣传，对于符合专精特新“小巨人”标准的企业可享受一次性50万元奖励，鼓励企业进行技术创新和产品研发，切实以创新促进企业提高核心竞争力。为促进企业的发展，强化企业人才队伍建设，定期开展培训班、讲座、招聘会等活动，为企业的发展提供有力支持，同时推动企业与高校、科研机构之间的合作，建立产学研一体化机制，加速创新企业的成长和发展。同时，加强企业融资支持，办事处切实把做好科技创新金融服务摆在突出位置，指导金融机构建立科技金融工作站，根据不同发展阶段的科技型企业不同需求，为科技型企业提供全生命周期的多元化接力式金融服务，有效降低科技型中小微企业融资成本，为企业提供更为精准、优质、高效的金融服务。

龙子湖街道办事处召开银企对接会，邀请中国银行、光大银行等金融机构及辖区创意岛孵化器、大豫众创空间、数字育才、启迪之星（河南）、大象科技孵化器5家孵化器，双方就深化合作、共同推动在孵企业发展进行了深入探讨。对接会上，银行方面向孵化器企业详细介绍了相关金融服务政策、信贷产品，并以答疑解惑和政策探讨的形式进行了现场解读。办事处相关负责人表示，将持续深入开展“四项对接”活动，建立“政银企”三方联络机制，打通金融活水流向实体经济“最后一公里”。

商都路办事处以“万人助万企”活动为契机，聚焦数字马力（郑州）信息技术有限公司等重点企业发展需求，主动履职、细化措施，全力打造“企业全生命周期”服务链，筑牢企业全周期服务根基。办事处按照“重点企业、重点帮扶”原则，针对辖区新入驻“蜜雪冰城”“牧原集团”“蚂蚁集团”等重点企业，商都路办事处充分发挥属地帮扶职责，抽调由书记、主任在内的10个分包领导、12个社区书记等组建“蜜工圈”“牧工圈”“蚁工圈”三大工圈助企服务专班，实行“1+1”辅导联系服务机制。聚焦企业入驻难题，高效提供跨部门咨询服务，切实帮助企业解决前期入驻、logo安装、税务手续办理等程序问题，围绕企业资源整合难题，积极借助办事处“商事办”便民服务小程序增设“万人助万企”——“三大工圈”服务专栏，助力企业盘活自身招商资源、链通周边产业链资源，进一步帮助企业提升综合竞争力和发展潜力。

郑州中原科技城综合服务中心持续开展企业走访调研活动，助推企业转型升级。综合服务中心科企服务专班走访了河南海义集团、河南金盾信安检测评估中心有限公司、河南南秀建筑科技有限公司等东区重点企业，了解企业在传统产业、商业地产开发与运营管理等业务之外，在金融投资、医疗健康、智能制造及生物科技等新兴产业板块布局，针对企业提出的引进外资奖补、高新技术人才缺乏等制约企业发展的问题，科企服务专班表示，将积极对接各相关职能部门，落实惠企政策、提供人才招引渠道、协调推动各项问题解决，切实帮助企业纾困解难。

聚焦开源芯片领域 培育创新型人才

本报讯（记者 覃岩峰 通讯员 孔倩倩）8月28日至29日，“计算机系统与开源芯片敏捷开发”首期师资培训班在国家超级计算郑州中心举办，来自郑州大学计算机与人工智能学院、软件学院及郑大超算共32位中青年骨干教师参加了本次师资培训。

本次由中国科学院计算技术研究所先进计算机系统研究中心副主任、中国科学院大学教授、教育部“计算机系统与处理器芯片课程虚拟教研室”（简称“CSP虚拟教研室”）秘书长张科及中国科学院计算所副高工、中国科学院大学副教授常轶松领衔的专家教学团队以中国科学院大学计算机组成原理、一生一芯计划、操作系统等理论和实验课为基础，与思沃/SERVE云平台紧密融合，为骨干教师提供了一体化、贯通式、多层次的师资培训课程。

培训由中科大数据研究院智能系统与芯片实验室牵头，得到了中国科学院计算所、中国科学院大学、中国开放指令生态（RISC-V）联盟（简称“CRVA联盟”）、CSP虚拟教研室及郑州大学的大力支持。中国科学院计算所副所长、CRVA联盟秘书长、CSP虚拟教研室主任包云岗，中国科学院计算所教育处处长、中科大数据研究院院长、CSP虚拟教研室副秘书长王元卓，郑大计智学院党委书记、郑大超算主任商建东，以线上线下结合的方式列席参会，分别为培训致辞并带来精彩报告。

据悉，中科大数据研究院将通过与河南高校开展更深入的合作，逐步探索出一套标准化的、成体系的、可复制的、更适应地方高校发展的培训认证体系，结合思沃/SERVE云平台，实现“以拔尖师资培养拔尖人才”，为河南省培养更多优秀的创新型科技人才贡献力量。

传承红色精神 汲取奋进力量

本报讯（记者 覃岩峰 通讯员 李瑞洁 袁兴 文/图）日前，郑东新区党政办组织党员干部前往郑州二七纪念馆，开展主题党日活动，进一步激发党员干部爱国情怀。

党员干部们参观二七纪念堂，在铁路修建、星星之火、革命烈焰、永恒丰碑等板块，大家驻足聆听讲解，通过一件件文物，近距离感受革命先烈英勇不屈的抗争意志和强烈的为民情怀（如图）。随后登上二七纪念塔，重温革命工人血染大地的英雄史诗，仰视他们用热血和生命铸就的伟大丰碑。今年以来，郑东新区党政办创新开展“三课堂一活动”，即政治理论课堂、党办业务课堂、读书分享课堂和主题党日活动，累计开展法治宣传、支部共学、志愿服务、红色教育等党建活动30余场，着力打造“三强一好”的党员干部队伍，汇聚郑东新区“三高地、两中心、一新城”建设蓬勃力量。

提升管理水平 营造整洁环境

本报讯（记者 覃岩峰 通讯员 冯凯 刘媛媛）连日来，郑东新区全面提高城市精细化管理水平，最大限度发挥城市管理工作效能，营造整洁有序城市环境。

按照郑东新区综合行政执法局（城市管理局）、爱卫办及社会事业局安排部署，河南东越市政工程有限公司有序开展城市家具、环卫设施病媒防制消杀专项工作。东越市政依托自有病媒防制专业队伍和服务能力，持续开展病媒生物密度监测，全力做好城市家具、环卫设施秋季病媒防制工作。同时根据垃圾中转站、爱心驿站等环卫设施病媒生物的特性、繁殖状况和种群变化，采取集中喷洒消杀、物理防制和化学防制等综合防制手段，设置灭蝇灯、悬挂捕蝇笼，放置毒鼠盒、粘蝇板，科学用药、安全施药，有效控制蚊蝇密度达到控制水平标准要求。

如意湖办事处城管办积极行动确保中元节文明祭奠。办事处积极发动各相关部门及社区工作人员向辖区居民发放文明祭祀倡议书，积极宣传禁燃禁放的有关规定，引导居民不违规燃放烟花爆竹，不随地无序焚烧香蜡、纸钱、冥物等祭祀用品，提倡文明祭祀。城管队员对辖区重点区域进行夜巡，对焚烧冥币等不文明行为进行劝阻、制止，进行安全文明劝导，增强居民防火意识，倡导文明祭祀，引导居民用生态环境保护的方式开展祭祀活动。

聘　书

兹聘请 **张科** 为教育部高等学校计算机类专业教学指导委员会“计算机专业系统能力培养实施方案工作组”成员。聘期为即日起至 2022 年 12 月 31 日止。

教育部高等学校计算机类专业教学指导委员会

（主任单位代章）

2019 年 9 月

2022-2023学年春季学期《计算机组成原理》课程评估问卷

1. 老师在开学时明确告知了所要学习的知识和掌握的技能（即教学目标）

非常同意（47）
同意（4）
一般（0）
不同意（0）
非常不同意（0）

2. 课程要求、考核方式和评分标准在一开始就相当明确,对学生的义务也有很好的说明

非常同意（48）
同意（3）
一般（0）
不同意（0）
非常不同意（0）

3. 上课内容紧扣教学目标

非常同意（48）
同意（3）
一般（0）
不同意（0）
非常不同意（0）

4. 老师根据教学内容，有针对性地选用教材和参考资料

非常同意（48）
同意（3）
一般（0）
不同意（0）
非常不同意（0）

5. 教学中融入了老师本人或领域中的最新专业研究成果

非常同意（48）
同意（3）
一般（0）
不同意（0）
非常不同意（0）

6. 每节课前老师对本节课的教学目标和计划做出清晰的说明

非常同意（48）
同意（3）
一般（0）
不同意（0）
非常不同意（0）

7. 老师能使我保持长时间的听课注意力

非常同意（45）
同意（6）

一般（0）
不同意（0）
非常不同意（0）
8. 老师提出的要求，对我有挑战，但我通过努力可以达到
非常同意（45）
同意（6）
一般（0）
不同意（0）
非常不同意（0）
9. 为了鼓励同学参与，老师采用多种教学方法（提问、小组学习等）
非常同意（45）
同意（6）
一般（0）
不同意（0）
非常不同意（0）
10. 老师对概念、原理等的讲述清晰易懂
非常同意（46）
同意（5）
一般（0）
不同意（0）
非常不同意（0）
11. 多媒体资源运用得当
非常同意（47）
同意（4）
一般（0）
不同意（0）
非常不同意（0）
12. 课后作业有助于我了解课程内容
非常同意（47）
同意（4）
一般（0）
不同意（0）
非常不同意（0）
13. 课后作业的批改和反馈及时不拖延
非常同意（45）
同意（4）
一般（2）
不同意（0）
非常不同意（0）
14. 老师对作业、考试的作弊采取了合理的预防

非常同意（46）
同意（5）
一般（0）
不同意（0）
非常不同意（0）

15. 本课程很有意思，我喜欢学习

非常同意（46）
同意（5）
一般（0）
不同意（0）
非常不同意（0）

16. 平均每周课后投入本课程学习（含完成工作）的时间

超过4小时（19）
3（不含）- 4小时（9）
2（不含）- 3小时（15）
1（不含）- 2小时（7）
1小时及以下（1）

17. 本课程让你在以下哪些方面受益

学术知识（49）
专业技能（36）
学术伦理（15）
科学研究思维（24）
批判性思维（13）
阅读能力（11）
写作能力（7）
沟通能力（11）
合作能力（10）
演讲水平（8）
学术视野（13）
人文底蕴（3）
社会责任（6）
都没有（0）
科学精神（7）
家国情怀（2）
创新能力（5）
其他（请填空）（0）

18. 你会向其他同学推荐这门课/这位（组）老师吗

很推荐（47）
推荐（4）
不推荐（0）

19. 你对本课程的助教老师工作的评价？

好

没啥感觉
好
工作认真
好好好，非常好，非常棒，很不错
助教老师提醒上课十分及时
张科yyds!
答疑认真
助教很负责！！！

助教负责认真
助教认真负责，考勤及时
好
助教很负责

助教老师作业批改及时
很好!
1
讲的非常好，受益匪浅
好
好
态度认真
好，作业反馈及时
非常好，工作负责用心
好
助教老师工作很认真
暂无

非常好
助教老师工作认真，作业反馈及时
课程内容丰富，教师上课认真负责

我觉得没有问题需要改进
助教工作认真负责，作业反馈不是很及时

助教老师工作非常认真
很负责任 改作业改的很好很及时

助教老师认真负责，唯一的建议就是希望习题课能讲的更仔细具体一点。

很好

总体不错，除了作业批改拖延较多
非常认真负责！

20. 请对该门课程助教老师的整体工作进行打分，满分为10分。
10（31）
9（8）
8（4）
7（0）
6（0）
5（0）
4（0）
3（0）
2（0）
1（0）
21. 你对本课程的总体感受或给老师的建议
好
赞
老师讲的很好，没有什么需要改进的
从各方面来说都是计系讲得最好的课了，不过或许可以换本新一点的、不要这么理论化的教材……
很负责，很高级，好的好的
老师授课风格诙谐，使得课堂十分生动
好好好，非常好，非常棒，很不错
希望老师在布置作业后可以同时开放作业提交入口
求求张科来教我所有的计系专业课！
张科老师神中神，课程准备认真，考虑学生感受，没有建议！
张老师的课很生动，课件上的内容也很充实，没有缺点
课程很好，老师很认真
老师讲课很有趣，认真负责
总体感受很好,张老师非常负责,讲的很好
老师课讲的很好，备课认真，课堂生动有趣
很好的课，没啥意见。
老师可以上课时把进度加快一些
没啥建议，凑字数1111111
五星好课，老师讲课简单易懂，课程体系成熟

老师讲得很好，但最后明明课都快上不完了还是在举行的科研调查报告和汇报带给人很大焦虑
老师讲课很有激情，收获良多!
1111111111111111
讲的非常好，受益匪浅
非常好，非常推荐，谢谢老师
老师讲解课程知识点清晰，让我受益匪浅
课程非常不错，课程非常不错
张科老师对计组的讲解细腻，还能活跃课堂氛围，鼓励学生自己思考
张老师的计算机组成原理课程的教学是本科计算机教学的一大亮点，内容丰富，讲解细致，考虑周全，关心学生，ppt制作生动幽默，让我们自主调研计算机领域，自己探索现阶段计算机领域的前进方向，在让我们获得专业知识的同时培养了我们对硬件的兴趣。大爱COD!
真正站在了学生的角度来授课，好!
张科老师讲的非常好, 既有意思, 也有深度!
讲的不错，以后多多讲讲
部分ppt上文字解释较少，对于复习存在一些困难，可以进行优化
非常好，老师很认真，非常好
这门课程还是挺不错的 值得推荐给后面的同学
课程内容丰富，教师上课认真负责
不要再加大力度了，真受不了
我觉得没有问题需要改进
老师上课内容讲述清楚，课程模式多样，善于玩梗，寓教于乐
张老师的课果然名不虚传，生动活泼，能把计组这门知识细碎的学科讲得很有条理真的非常棒，同学调研环节也是非常好的一个设定，学到了很多
张老师的课程节奏适中又不枯燥，课程讲解也非常细致
好的要命 特别好 非常好 讲得好 很负责任
张老师认真细致，真心为同学；给人的感觉是计组原理这门课内容太少，张老师可以游刃有余地肆意发挥。确实厉害。
好好好好好好好好好好
我的感受非常非常良好!
总体感受很好，难得的好课
课程整体感觉比较良好
好！先生之风，山高水长，循循善诱，娓娓道来。迄今为止上过的最好的专业课!
张老师的课堂总是很有趣、很精彩，能够将很多枯燥、复杂的内容通过更新颖、更有意思的方式呈现出来，真的花了不少的心思为我们准备，非常感谢!
张科老师真的太好了，很幸运能有这样一位老师，是我不够好。

老师认真，讲解认真。

当前评估课程为：计算机组成原理 主讲教师为：张科 所属院系为：计算机科学与技术学院

选课人数为：64 人 参评人数为：62 人 评估结果：优

序号	评选项	平均分	评估结果
1	教学态度：1）治学严谨，备课充分，讲课认真	99.68	优
2	2）教师对本课程有足够的掌握和认识	100.00	优
3	3）停课、调课未影响教学效果	100.00	优
4	教学内容：1）符合大纲要求，目前的教学进度与教学安排一致	99.19	优
5	2）讲课内容重点突出，详略得当	99.19	优
6	3）教材、讲义及参考书符合教学内容	99.68	优
7	教学方式：1）富于启发性，因材施教，语言生动，能调动学生学习积极性	100.00	优
8	2）注重传授知识与培养能力相结合	99.68	优
9	3）合理配置作业或课外阅读	100.00	优
10	教学效果：1）达到教学目的和要求，提高学生知识能力	100.00	优
11	2）通过本课程的学习，提升了学生对该课程及该领域的兴趣	98.87	优

序号	对这门课程和教学的总体看法、意见与建议	如本课程有助课教师，请对助教给出简要评价	请自我评价通过学习这门课程的收获以及学习过程中遇到的问题
1	教师授课专业，学风严谨，讲课非常有耐心。	助教老师认真负责，总结到位，回答问题十分耐心。	基础知识掌握良好，在此基础上对前沿科技认知有所提高。
2	课程安排合理，老师特别负责，讲课也不错	助教老师很好	学习了计算机组成原理

3	老师上课很认真, 讲课有趣且有条理	助教老师主要负责改作业和讲作业, 都挺负责的	学到了很多计算机的硬件知识, 感觉对计算机整个学科都有了更深的了解
4	老师非常认真负责，教学效果很好。	认真负责	收获很大
5	课程内容丰富，老师认真负责。学到很多知识。	助课教师认真负责。	学到很多知识，收获很大。
6	老师对待教学认真负责，语言生动，条理清晰，举例充分恰当，对待学生严格要求，能够鼓励学生踊跃发言，使课堂气氛比较积极热烈。课堂内容充实，简单明了，使学生能够轻轻松松掌握知识。	老师对待教学认真负责，语言生动，条理清晰，举例充分恰当，对待学生严格要求，能够鼓励学生踊跃发言，使课堂气氛比较积极热烈。课堂内容充实，简单明了，使学生能够轻轻松松掌握知识。	内容需要课下消化
7	老师认真负责，我学到了很多知识	助教认真负责，我学到了很多知识	无
8	张老师治学严谨，备课充分，讲课认真，对待学生非常友善，必须点赞！！	我对助教给出简要评价：好！	更加深入地了解了计算机的底层设计，感受到计算机底层的魅力。
9	让我了解了计算机的组成，对软硬件协同有了更深理解。	助教老师认真批改同学问题，积极答疑。	在这门课中学到了很多计算机相关知识，内容丰富。
10	老师讲解介绍的知识较为新颖前沿，对我们有很大的启发作用和引导作用，谢谢老师	助教老师很认真负责，很认真的带领我们复习备考，也很认真的给我们讲题，感激	老师讲的有些东西很难理解，还要加强自主学习的能力
11	这门课程教学很好，老师讲课清晰，生动活泼	助教老师认真负责，批改作业以及习题讲解得很好	学习过程中更加深入的了解了这门课程
12	好	好	对计算机的组成有了一些了解，平常对计算机硬件相关的内容也有了兴趣
13	老师讲课生动有趣，助教认真负责。	老师讲课生动有趣，助教认真负责。	没有
14	老师讲课非常认真，准备充分，课堂内容充实	助教老师工作很认真，非常尽心尽力	收获了很多方面的知识

15	张科老师很是认真负责了，理论课上课的形式和内容都很不错	助教老师们改作业都很认真，讲题也很清楚	实验课与理论课相互填充，非常有趣，收获很多
16	体验很好，学到很多东西，有收获。	助教挺好的	因为一些原因没有上实验课，还是有点遗憾，觉得自己少了运用所学知识的机会，理解的也不够深入。
17	老师讲得很好老师讲得很好老师讲得很好	老师讲得很好老师讲得很好老师讲得很好	老师讲得很好老师讲得很好老师讲得很好
18	计算机组成原理是了解计算机硬件方面的基础课程。	助教挺好的。	对计算机硬件有了基本的了解。
19	教师讲课认真细致，并且注重为学生补充课外知识。个人感觉计算机组成和体系结构是计算机相关课程中比较难和枯燥的，自己看教材和课件时容易走神，但是经过教师的讲解，感觉接受起来会容易很多。目前没有需要提出的意见和建议。	助教：李作骏，张聪武，阮金杰 助教工作认真负责，批改、讲解作业都很细致。	对于计算机的组成有了基本的了解。在学习这门课之前，虽然略微知道计算机内部的结构很复杂，但是没有清晰的认识，现在逐渐了解了计算机的组成部分以及它们如何实现功能、彼此沟通，感觉很有收获。学习过程中的问题主要是，由于之前没有学过数字电路，对一部分内容（比如处理器）的理解有一定的困难。
20	挺好	很好	实验课与理论课相互填充，非常有趣，收获很多
21	计算机组成原理理论课介绍了很多新知识，调研作业设计也很合理，可以启发兴趣。	助教老师批改作业认真。	这门课巩固了专业基础并且提升了能力。
22	这门课教学充分咨询了学生的意见，也有非常多的教学参与过程，非常好！	助教老师非常耐心地解答同学们的问题，并且认真批改作业。	学习到了计算机组成原理相关知识
23	老师讲的很细致，由浅入深，循序渐进。	助教每次都耐心的讲解习题，十分认真	系统地学习了计算机组成原理的知识

24	老师每次都会很早来教室准备好 ppt，内容也很详细，很幽默，同时会提供计算机硬件方面的最新消息，也会邀请学生对计算机硬件方面的知识进行课堂展示，课下也会耐心解答疑问	李作骏、阮金杰、张聪武助教对作业会进行认真的批改，然后对其中大部分同学存在的问题在课堂上进行讲解，课下也会对同学的疑问进行耐心解答	学到了关于计算机硬件设计方面的知识，为之后的进一步的计算机体系结构的设计打基础
25	这门课程介绍了计算机组成方面的基础知识	助教阮金杰工作认真负责	通过学习这门课程我掌握了很多计算机组成方面的知识
26	老师上课方式较为新颖，十分有趣	助教认真负责	无
27	老师讲课很吸引人，准备的课件也很详细。老师还会时常请一些人来给我们作报告，拓宽我们的知识面。学期末同学们的调研报告也很有意思	助教老师很负责	我对这门课得了解仍然不是很深，在许多地方理解的时候都不太好。不过好在实验课能帮助理解
28	老师认真负责，上课生动有趣，且讲课结构清晰	助教老师认真负责	对计算机如何工作有更深的认识
29	张老师讲课很好，而且善于与学生沟通	助教好	硬件方面有了更深的理解
30	张科老师讲课内容丰富，教学水平非常高，并且对教学十分重视，经常为我们介绍前沿发展消息，鼓励学生自行调研，撰写报告，课程教授的形式多样，加深了对计算机组成的理解。	助教认真负责，但是希望讲解习题时能更加详细，可以讲慢一点。	加深了对计算机组成的理解
31	老师上课风趣幽默、准备充分，重点突出，非常值得点赞	助教认真负责，习题讲解细致	对计算机组成有了更为深入的理解
32	老师很负责，讲的很好	认真负责	投入时间不够
33	张科老师非常负责，是督促同学们交作业最为勤快的老师，上课也非常易懂，对任何有难点的内容都会详细解释，并加以扩展。	助教批改作业认真及时，每次课前也都会用几页 ppt 讲解之前作业我们遇到的问题，很不错。	这门课程我的最大收获就是结合汇编语言，对计算机的基本组成和指令集有了较为全面的认识。

34	讲课速度可以加快一些，尤其是第六章的计算部分。	希望助教讲解作业时可以更清楚一些。	教材上的内容掌握起来没有太大难度，但是拓展的部分，尤其是现代计算机所用技术 理解得不够清晰。
35	张老师备课充分，有着极高的工作热情，把一门工程性的课程上出了一点点意思。	助教老师准备充分，繁忙之余能每次都来听课，很不容易。	学习到了计算机的基本组成，为后续硬件课打下了一些基础。
36	张科老师上课很幽默风趣，课堂效果很好，我们能学到很多知识。	助教老师批改作业负责。	对一些知识还是没有太多了解。
37	很好，通过这门课程的学习，学到了很多有用的知识	助教很认真	通过这门课程的学习，学到了很多有用的知识
38	老师非常认真负责，我学到了很多	助教老师非常认真负责地为大家批改作业并评讲	我自认为还行
39	课程非常引人入胜，通过许多额外的内容加深了我们对课程的兴趣和掌握程度，能接触到前沿的知识报告，还能和高级工程师交流。	认真。	在理论上掌握了计算机组成原理的初步知识。
40	教学形式多样，教学内容灵活丰富	助教专业负责	对计算机组成及其发展有了初步了解
41	老师很不错，教学方式很好，有助于知识的理解	助教认真负责	无
42	COS 作为一门承前启后的课程，却依然是国科大最有技术含量的课之一。教师的许多措施大大提高了学生对课程的兴趣，对下一届同学还可以继续增大难度，让他们从基础开始就领先于同龄人。	挺好的，助教老师认真负责。	对计算机如何工作有更深的认识
43	很好很好，可以加快细节的讲解，可以通过教材自学	作业认真	学习前沿

44	老师上课讲课生动有趣，认真负责，课堂效果很好，也讲了很多前沿的知识。	助教老师能够很好地完成助教工作。	在本门课程学习中有较大收获，无明显问题。
45	整体是很好的，老师讲课很认真，讲的东西也很好	助教存在感一般，但是批改作业还是很认真的	还挺有收获的
46	这门课教的非常好，体系完整，教学思路清晰	助教老师比较认真负责	基本没有什么问题
47	张老师讲得很好，备课也很充分，总体上很优秀	助教老师非常认真负责	通过学习这门课，我对计算机的底层硬件组成有了更深的了解，收获非常大
48	老师准备充分，内容充实，讲的也很有意思。	老师准备充分，内容充实，讲的也很有意思。	老师准备充分，内容充实，讲的也很有意思。
49	教师教学严谨，讲解深入透彻，能够及时解答同学疑问	教师教学严谨，讲解深入透彻，能够及时解答同学疑问	对计算机的运行有了新的理解
50	张科老师的课将课本的知识讲的十分透彻却又不拘泥于课本，收获很多	希望助教老师讲解习题时能给出标准答案和思路	了解了计算机的基本组成，为实验课提供了知识基础
51	老师治学严谨，要求严格，能深入了解学生的学习和生活状况，循循善诱，平易近人;注意启发和调动学生的积极性，课堂气氛较为活跃;上课例题丰富，不厌其烦，细心讲解，使学生有所收获;	助教还算负责认真	收获颇多
52	这门课讲了计算机组成的原理知识，对于计算机领域的入门者（我）而言，非常适合。	这门课的助教认真批改作业，清晰地讲解题目。	我学到了一些计算机的组成方面的基础知识。
53	这门课程的内容很好，但在学习的过程中总感觉太过抽象，但也不知道如何能让这些知识变的形象。	助教老师很负责。	通过学习这门课程，我又向“会修电脑”的目标进了一步。
54	治学严谨，备课充分，讲课认真。	治学严谨，备课充分，讲课认真	通过本课程的学习，提升了学生对该课程及该领域的兴趣

55	张老师非常认真负责，不仅讲解课程范围内的知识，而且设法引起我们对计算机的兴趣	助教老师批改讲解作业都很认真	理论课和实验课结合效果更好
56	老师讲授了很多计算机领域的新进展，让我在课上看到了很多未曾了解、不够了解的东西，对计算机的发展历史和各部件的工作原理等有了深刻的认识，非常感谢老师本学期的授课。	助教老师讲授都很清晰。	本课程中的内容比较丰富，需要理解的内容也比较多，有些内容我有时候难以完全掌握。
57	老师认真负责，课堂内容深入浅出，使我受益匪浅。	助教老师批改作业与讲解习题认真负责。	通过这门课程的学习，我对于计算机的组成原理有了更深层次的了解。
58	老师认真负责，必须给好评，希望学校有更多这样的好老师。 非常满意，很好的课程，没有意见和建议。	比较满意，挺好的课程，没有意见和建议。	比较满意，挺好的课程，没有意见和建议。
59	计算机组成原理这门课比较难，也非常重要，但是张科老师讲的还是非常的清楚，在教学过程中还增加了许多最新的前沿相关进展，受益良多，就是讲课速度稍微慢一些就好了。	助教老师还是非常认真负责的，能够按时批改作业，就是讲作业的时候声音大一点，不要太羞涩。	这门课还是让我对整体的计算机的结构有了基本的认识，也为实验课打下了理论基础。就是内容比较多吧，学起来比较吃力，但是学下来还是收获满满。
60	个人看来本学期最棒的授课了。 内容覆盖面广，教学风趣，很棒。	认真	对硬件结构有了更好的了解
61	老师上课很棒，很幽默，讲课也很清晰。	助教负责	理论与实验结合，很不错
62	计组课难度适中，覆盖面广，知识点多，与实际贴合地很好，同时与实验配合学习效果很好，强化同学们的理解。	助教老师很辛苦地帮助同学批改作业，debug，很良心。	作业量很大，但是学到了很多的计算机组成原理知识，同时结合实验课也强化了动手写硬件代码的能力。