Pedagogical case: ASIC Design

Authors: Vassiliy Tchoumatchenko, Tania Vasileva
Affiliation: Technical University of Sofia, Bulgaria

The educational problem

To increase motivation of students to learn, to enhance their professional competences and soft skills:

- Abilities to resolve complex problems;
- Better practical training (experience in using professional software and CAD tools);
- Working on multidisciplinary tasks, utilizing multiple knowledge sources;
- Abilities to work in group;
- Abilities to use modern computer technologies and environments;
- Networking.

Key experiences

Working in small teams, the students are required to design a digital Application Specific Integrated Circuit (ASIC). The design workflow is based on HDL (Hardware Design Language) modelling, verification and synthesis. The main design artefacts (HDL models and test-benches) are text files; therefore we are able to borrow many tools and workflows from the software development community.

Projects are hosted on GitHub – one repository per project. In parallel with the code development, the teams are required to create and maintain a Google Docs document which is one of the major deliverables.

Key experiences (cont.)

Initially the document contains the technical specifications of the design. Later on, the students have to add description of the implemented algorithms and architectures, argumentation of the tradeoffs made and the results from the simulation, synthesis and physical design.

Most of development takes place outside the regular classes. For their intra-team communication, the students are free to choose whatever tools they prefer (chat, conferencing, email). For student - teacher communications we decided to use the Google tools. Students were encouraged to submit their questions as in-document comment or email.

Infrastructure For Collaborative Learning

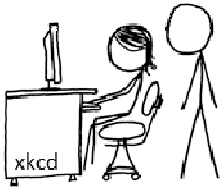
VHDL Code Repository

GitHub

Simulation
and Synthesys

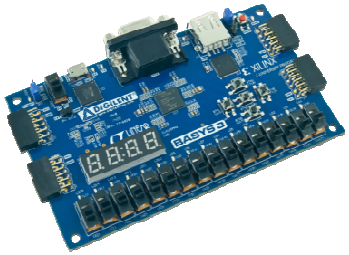
EDA playground

XILINX®



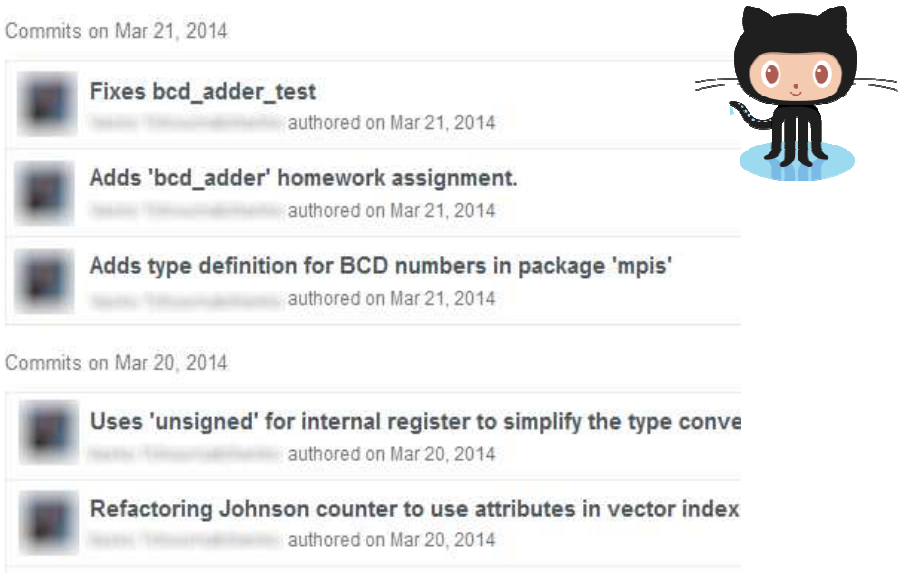
Team

Google Tools



Lab Test

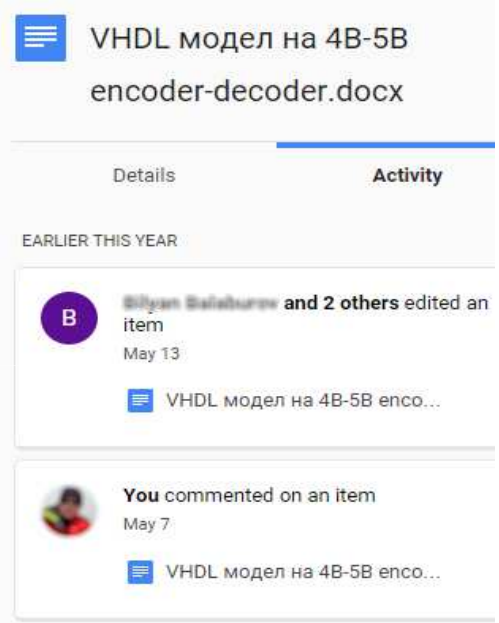
Collaborative Artefacts Evolution - GitHub

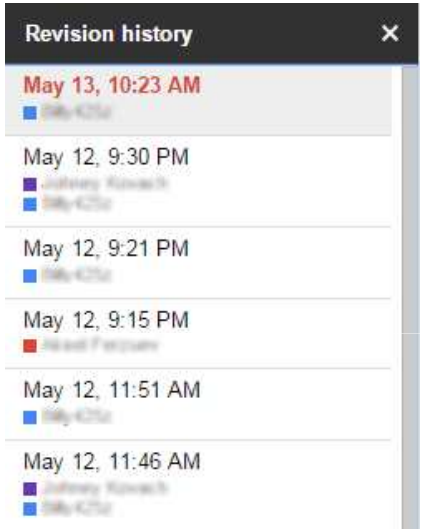


history view – every change is time-stamped and attributed to an author

diff view – each change can be inspected in details

Collaborative Artefacts Evolution - Google Docs / Drive



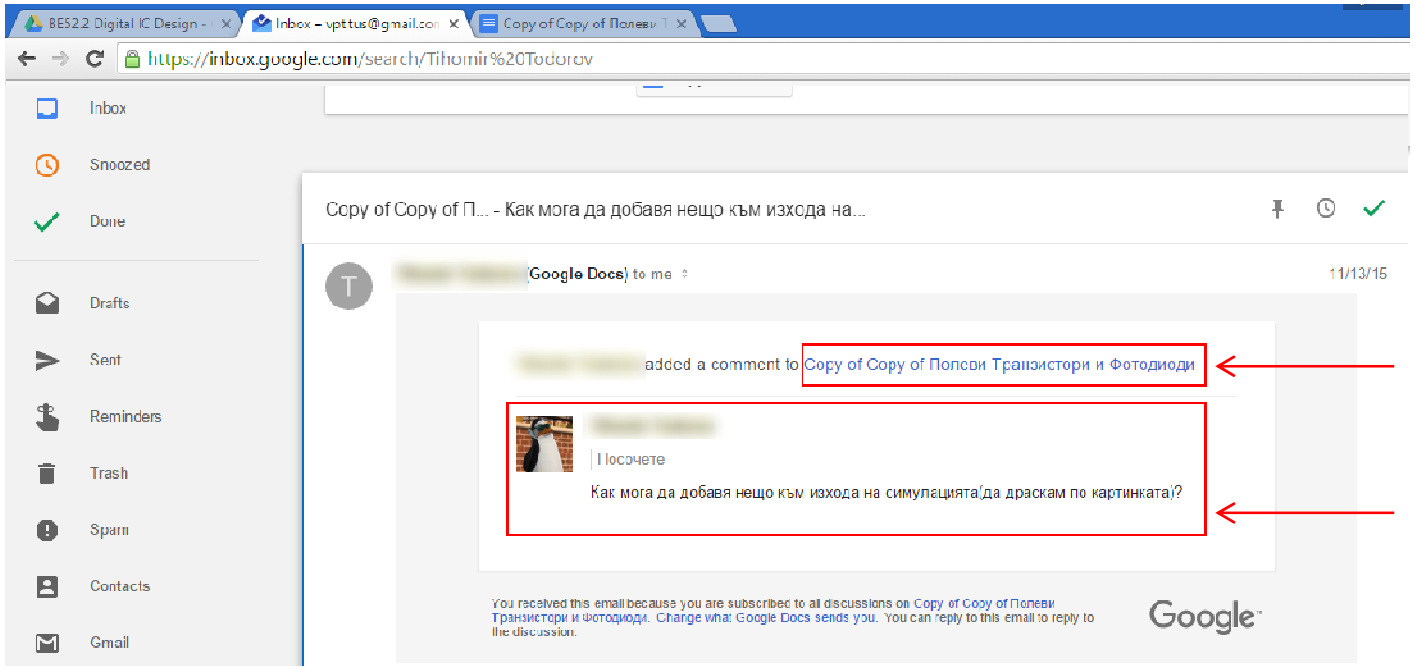


Activity view – overview of who created or modified a particular document

Revision History View - timeline of the changes, but finding the exact change in the document is difficult

Student – Professor Communications

Students submit their questions as in-document comment

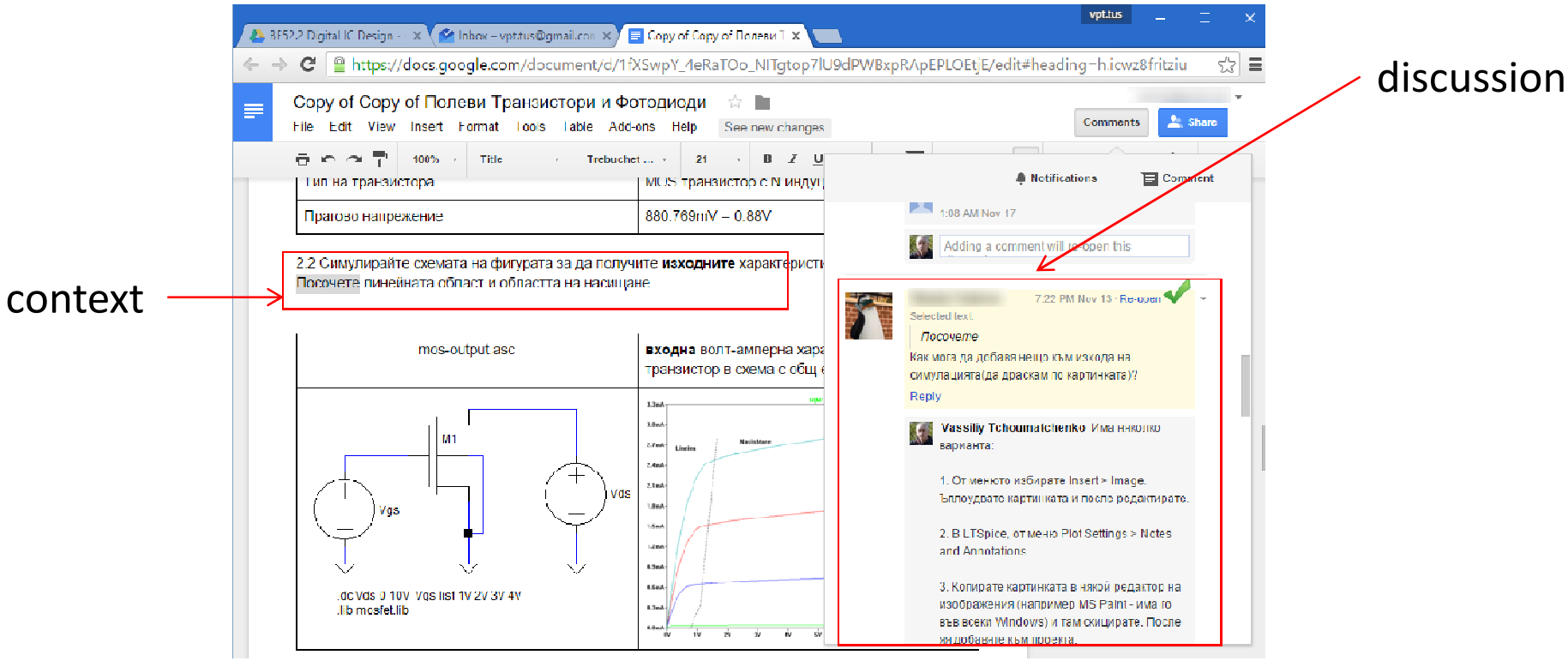


link to the document

question

Student – Professor Communications (cont.)

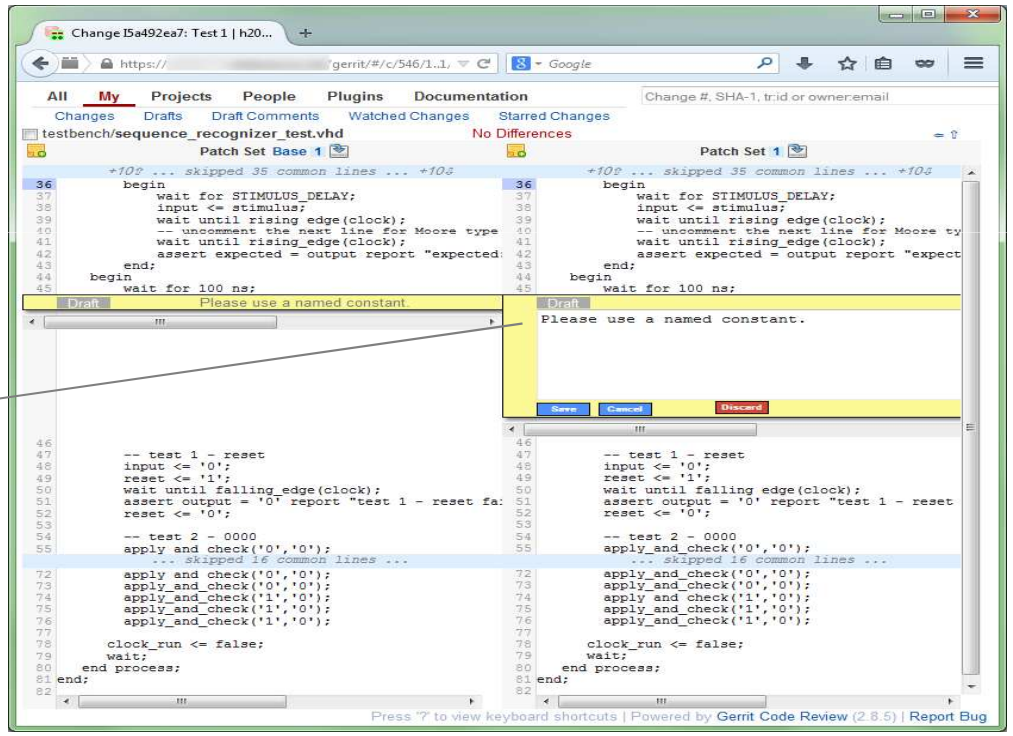
Professor answers in the document. The discussion is **part of the artefact**.



Peer reviews
(experimental)

Another team member shall perform a code review and either approve the change or return it to the submitter for rework.

Gerrit allows the reviewer to attach comments to a source code file or a particular line.



Implementing the trialogical design principles

DP1: Organising activities around shared objects

- Collaborative development of a 10 weeks long project, preparation of a shared report.
- Activities: face to face meeting for the distribution of project tasks, defining and preliminary review of the tools used; getting students acquainted with the phases of the design cycle.
- Team organization: students are free to choose team partners.

Implementing the trialogical design principles (cont.)

DP2: Supporting integration of personal and collective agency and work

- Coordinate the participants' interests – team members to choose an appropriate project they want to develop and are interested in.
- Motivate students to manage tasks distribution between team members, having respect to deadlines by giving each student the responsibility of being a team leader in turn. There are two strictly defined deadlines during the project cycle and performing tasks within given deadlines is one of the criteria for project evaluation.
- Combining participants' own interests and shared project through assessment process – the quality of the shared project and the responsibilities concerning deadlines are evaluated with a higher grade.

Implementing the trialogical design principles (cont.)

DP3: Emphasizing development and creativity through knowledge transformations and reflection

- Support versatile use of various kinds of knowledge: theoretical or literary sources; practical examples and cases; pictures, models. Discuss problems student faced during collective work on the shared object.
- Reflect on collective practices and knowledge.
- Make students comment on each other's work through the semester.
- Encourage students to independently and creatively initiate, lead and manage a development process.
- Require student to apply already obtained knowledge and skills in using professional CAD tools to resolve complex practical tasks.

Implementing the trialogical design principles (cont.)

DP4: Fostering long-term processes of knowledge advancement

- Continuous working process, iterative design process to improve circuit performance in order to fulfill technical specifications. Team members have collaborator rights for the respective repositories, but they are asked not to commit directly. Each change had to be peer reviewed before it can be committed to the project repository.
- Planning and start writing the documents, sharing the drafts, getting feedback from the teacher and other students, improving the project and project report, submitting relevant documents.
- The best final team projects and their shared reports are used during the course as good examples for other teams, as well as after the course by the students themselves or by others.

Implementing the trialogical design principles (cont.)

DP5: Promoting cross-fertilization of knowledge practices and artifacts across communities

- Students use professional tools for projects development. They are provided with professional work models and design flow cycle used in the software and electronic industry.
- Expert practices are modeled for students, via templates and tools.
- Students and teachers collaborate on solving a shared problem.
- Students use up-to-date cloud computing and communication tools for planning, organizing and writing shared reports.

Implementing the trialogical design principles (cont.)

DP6: Providing flexible tools for developing artifacts and practices

- Google Drive for collaborative authoring of the project reports, reviewing and commenting.
- Google calendar – to set deadlines and to monitor progress – assignments, intermediate stages reporting, deadline for submission of the project.
- For inter-team communications students can choose their preferred tools (chat, conferences, e-mail, forums).
- For student – teacher communications Google applications are used : Gmail, Calendar, Drive.
- Projects are hosted on GitHub – one repository per project.

Conclusions

Introducing new technologies in established engineering courses is always challenging. In addition to the core subject matter, students had to learn new tools and development workflows as well as knowledge work practices:

- Information processing
- Analysis and presentation
- Sharing
- Versioning
- Commenting
- Longitudinal work
- Using digital tools and group work

Conclusions (cont.)

In a whole, it has been a rewarding experience for both students and teachers. The trialogical approach was well accepted and considered as an appropriate path for transforming students' individual work into more collaborative activities.

The students appreciated the visibility of their contributions to the project – GitHub commit history and Google doc revision history.

Playing (and learning) with new technologies is fun. Although the students had no previous experience with version control and code review tools most of them enjoyed playing with the new toys and learning "cool" new skills .

Conclusions (cont.)

The immediacy of the help provided via email and in-document, contextual comments, compared to the scheduled face to face meeting, was cited as a major plus in the post-course surveys.

"The motivational divide" - The introduction of relatively complex, “real world” design workflows and tools highlighted even more the difference between the motivated teams and the students that just wanted to “get over it”. This observation was confirmed by the scores distribution – most were clustered in the top and bottom of the scale with very few in between.