COE ABET COMMITTEE

Activity Report

Term T111

COMPUTER ENGINEERING

Program

at

King Fahd University of Petroleum & Minerals
DHAHRAN, SAUDI ARABIA

January 2012~~October 2011~~

CONFIDENTIAL

The information supplied in this Self-Study Report is for the confidential use of ABET and its authorized agents, and will not be disclosed without authorization of the institution concerned, except for summary data not identifiable to a specific institution.

Table of Contents

Formatted: Heading 1

PLANNING AND ASSESSMENT FOR NEXT TWO YEARS

The ABET Program Assessment Committee approved the following Action Plan for the following two academic years.

Action Plan

First Year 2010-2011:

1. Addressing the comments raised by ABET in the Final Statement as well as those that were verbally formulated:
    1. Proposing some revisions,
    2. Seeking approval by COE faculty,
    3. Revising the COE the assessment system (rubrics and surveys) based on potential revisions.
2. Develop a policy for the effective implementation of Continuous improvements:
    1. Proposing a policy,
    2. Approval by COE faculty, and
    3. Implementation.
3. Continuous improvement of the COE program, identify programs outcomes that have weak performance indicator using available assessment data, selection of some outcome for improvement, carry out improvement, and documenting.

Second Year 2011-2012:

1. Continuous improvement of the COE program, identify programs outcomes that have weak performance indicator using available assessment data, selection of some outcome for improvement, carry out improvement, and documenting.
2. Develop logistic to address all program outcomes. Provide supporting material.

The above plan can be summarized using the following Table:

COE PLAN	T111	T112	T121	T122
Continuous Improvement	Two Selected POs		Two Selected POs	
Program Assessment		Assessing all Program		Assessing all Program

		outcomes (a)-(k)		**outcomes (a)-(k)**

Formatted: Body Text

CONTINUOUS IMPROVEMENT in T111

Continuous improvement of the COE program is a continuous task that is carried out by the curriculum and the ABET Committees. Curriculum revisions or corrective actions proposed by either of the above committees are presented to all COE faculty members in council meetings for discussion, review, and approval.

Action to Improve the Program

The process of direct and indirect assessment started in 2007, it has taken a few years to mature into an acceptable level. Using the collected direct and indirect assessment data, the COE ABET Committee started analyzing the level of achievement of each program outcome based on passed assessment data. The committee has made use of the POs achievement data as the basis for improving some POs in each semester in accordance with the COE Assessment Plan. In T081, the COE ABET committee started implementing the process adopted for continuous improvement in the COE Program.

The POs assessment process uses the following assessment channels: (1) the Exit Exam, (2) the average score in supporting courses, (3) the Exit Survey, (4) the COOP Supervisor and Employer Surveys, and (4) the Rubrics score for a set of representative core junior and senior courses

In Criterion 3, the status of the POs were grouped together in one single assessment table which includes for each PO the results of various assessment channels, the performance criteria, the score representing the level of achievement, and a summary of the committee comments. Following the analysis of the assessment data, the committee adopted a 2-year continuous improvement plan. The committee decided to improve Outcomes (c) and (g-W) in the academic year 2008-2009. These outcomes were selected first because of their importance to the program and due to their low score. These improvements were originally based on developing some guidelines for the students. Over time the issue of Continuous improvement matured into the need to provide teaching material in order to directly improve the student learning. For this reason, the committee carried out Continuous Improvement (CI) for (g-W) and decided to develop appropriate teaching material for COE 390 on three soft outcomes which are the “Engineering Ethics”, “Awarness of Contemporary Issues”, and “Awarness of the Impact of Engineering Solution on Society”. These can be found in our term report:

http://faculty.kfupm.edu.sa/coe/mayez/ABET/COE-ABET-Committee-2011-2012/COE-ABET-Activity-Report-T102-Oct-2011.doc

In Term 111, the committee analyzed (1) the previously collected direct and indirect assessment data and (2) current status of the program outcomes. The above issues were largely debated and the committee and the committee decided to carry out CI cycle in T111 for improving the (1)

Engineering Design, and (2) the Integration of Hardware and software, an extra outcome added by the COE faculty to the standard ABET a-k outcomes. On 1st October 2011, the committee assigned following faculty:

a. Dr. Mohamad Sqalli, assistant professor at the COE, was assigned as Faculty in Charge to carry out the CI for the Integration of Hardware and software,
b. Dr. Yahia Osais, assistant professor at the COE, was assigned as Faculty in Charge to carry out the CI for the Integration of Hardware and software. On 18 December 2011 Dr. Yahia Osais decided to retire from the committee. Dr. Wassim Raad was appointed to carry out the same task.

Improving Engineering Design (Dr. W. Raad and Dr. M. Al-Mouhamed)

Formatted: No underline, Complex Script Font: Not Bold

Formatted: Heading 2

In T111, following the analysis and examination of the previous Rubric Assessment Data and the available indirect assessment data, the committee decided to carry out Continuous Improvement measures for the Program Outcome on Engineering Design (c) and Dr. Waseem Raad (will be referred to as faculty in charge) has been nominated to conduct the above process. Outcome (c) had always received somehow marginal scores of about 2.6/4, and accordingly was selected for improvement because of its importance to the COE program.

For Program Outcome on Engineering Design (c), the ratings of learning using direct measures with student-by-student. The rubric assessment data is presented as part of the display material. In addition, the display material presents the performance criteria used for the rating of each PO. Outcome "C" has five performance criteria: (1) translate general requirements, (2) identify and formulate any problem, (3) list different design alternatives, (4) choose the appropriate design, (5) fine tune the chosen solution. Average student performance was considered inadequate (score below 2.5 / 4) score for the performance criteria (2), (3), and (4). This indicates that the concept of Engineering Design needs to be better defined, practiced wherever possible, and be integrated as part of all the COE courses with a culminating engineering design experience in the Capstone course.

To close the loop, the faculty proposed the following action plan:

1. In the light of the new COE BSc Program, there is need to spread the education of Engineering Design over all the Program courses. Specifically, at the 203 "Logic Design Lab" and COE 306 "Embedded Systems" are basically developed based on a set of lab experiments. Therefore, the Engineering Design package would improve if Mini-projects are adopted for the above courses.
2. The Committee reviewed the course syllabi of above courses and contacted the course instructors to discuss the idea of Mini-projects which received acceptance from both the instructor and committee.
3. Progressively introducing Engineering Design Education in the COE program Labs. Specifically developing or adapting educational material on Engineering Design for faculty lecturing, students reading, and provide web resource and references. Also, coordinating with the concerned course instructors to emphasize outcome (c) and to provide supporting material. Engineering education must be addressed in all relevant

courses and labs of the COE program to better prepare the student to the practice of engineering design.

4. The committee decided to develop a set of Mini-projects and work out the project formulation so that the concept of Engineering Design will be progressively introduced in the above courses. Each Min-Project is to be run for no less than 6 weeks. The students will team out for the project and finally submit a report. Having a set of Mini-projects provides the instructor different alternatives to choose from as well as to avoid centering the course for one specific project.
5. Also instructors, may deliver presentations to students aimed at improving outcome (c) "Engineering Design…" to improve their awareness about this issue.
6. Develop templates and guidelines for the students on Engineering Design.
7. Explore how Engineering Design Education can be enhanced in the COE Program.

The following action steps were carried out:

1. **Developing the following set of Mini-Projects (See the Appendix A):**

    a. XO-Game (COE 203),
    b. Reaction Timer (COE 203),
    c. RFID Authentication (COE 203),
    d. Traffic Light (COE 203),
    e. Elevator (COE 306),
    f. Edge Detection (COE 306).

2. To improve awareness of the students, a detailed description on **Engineering Design Processes** is provided to the students to reflect on its steps and their meaning on their project:

    http://faculty.kfupm.edu.sa/COE/mayez/ABET/COE-ABET-Committee-2011-2012/Engienering-Design-Process.pdf

3. **Extending the Engineering Education to COE Labs:** To develop a culture of Engineering Education at the COE, the engineering design concepts must be progressively implemented at different program levels. The committee reiterates its earlier recommendation to introduce some engineering design concepts at all levels in the Programs. Following Table provides a Plan for the Progressive Introduction of Engineering Design (Design a System, Component, or Process) in the COE program. It is recommended to adopt the following behavioral approach in all the COE Labs.

Plan for Progressive Introduction of Engineering Design

Elements of Engineering Design	**200-level Courses**	**300-level Courses**	**400-level Courses**

Formatted: Font: Bold, Complex Script Font: Bold
Formatted: Indent: Before: 1.27 cm, No bullets or numbering
Formatted: Font: Not Bold, Complex Script Font: Not Bold
Formatted: Font: Not Bold, Complex Script Font: Not Bold
Formatted: Font: Not Bold, Complex Script Font: Not Bold
Formatted: Font: Not Bold, Complex Script Font: Not Bold
Formatted: Font: Not Bold, Complex Script Font: Not Bold
Formatted: List Paragraph, Numbered + Level: 2 + Numbering Style: a, b, c, … + Start at: 1 + Alignment: Left + Aligned at: 1.9 cm + Indent at: 2.54 cm
Formatted: List Paragraph, Indent: Before: 1.9 cm
Formatted: Complex Script Font: 12 pt
Formatted: List Paragraph, Indent: Before: 2.54 cm
Formatted: Font: Bold, Complex Script Font: Bold
Formatted: Indent: Before: 1.27 cm, No bullets or numbering
Formatted: Indent: Before: 1.27 cm, Don't add space between paragraphs of the same style, No bullets or numbering
Field Code Changed
Formatted: Don't add space between paragraphs of the same style, Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.63 cm + Indent at: 1.27 cm
Formatted: Font: Not Bold, No underline, Complex Script Font: 12 pt, Not Bold

1. **Identifying a need** (There is a need for solving this problem!, what is this need?)			**Identify the need for solving the problem and domain of application**
2. **Defining the problem (**The design task will be completed if I solve a specific problem, What is this problem**)**	**Word definition of the problem.**	**Word definition of the problem. Enforcing use of Notebooks in all activities.**	**Word definition of the problem. Enforcing the use of Notebooks in all activities.**
3. **Conducting research** (How and where to search the library, ebooks, datasheets, manufacturer web etc to find relevant material to solve the problem)			**Searching similar problems (wording) and proposed solutions**
4. Narrowing the research (knowing the problem to be solved I need to filter Software/Hardware components that may help in building a solution)			**Knowing relevant examples be able to identify a few similar systems**
5. **Analyzing set criteria** (knowing the problem, its specifications, and its relevant components found, I need to find out whether some components meet the problem specifications or not)		**Analysis of problem specifications and identify components meeting some of the above**	**Analysis of problem specifications and identify components meeting some of the above**
6. **Finding alternative solutions** (determine a few possible solutions using found components for the above problem, I need to identify each of these possible solutions),	**Using different approaches or components. Use of Notebooks.**	**Combining components to find different solutions.**	**Combining components to find different solutions**
7. **Analyzing possible solutions** (knowing the problem (specifications) and possible solutions I need to find out whether some solutions meets the problem specifications or not)		**Pruning possible solutions which do not meet the specifications**	**Pruning possible solutions which do not meet the specifications**
8. **Making a decision** (given two or more possible solutions I need to select one feasible and economical solution)	**Selecting an economical solution**	**Selecting the most economical feasible solution**	**Selecting the most economical feasible solution (ethics, environmental)**
9. **Presenting the product** (I need to describe my solution in writing using diagrams, graphics, drawings, etc.)			**Technical description of the design**
10. **Communicating** (prepare a written report in which each task is broken down into: Identifying a need, Defining the problem, Conducting research, Narrowing down the research, Analyzing set criteria, Finding alternative solutions, Analyzing possible solutions, and Making a decision.)	**A simple report describing the above steps.**	**A report defining the problem, specification, analysis of components vs specification, solutions, solution analysis, and decision**	**A report defining the problem, specification, analysis of components vs specification, solutions, solution analysis, and decision**

Improving the Integration of Hardware and Software (Dr. M. Sqalli)

Continuous improvement (CI) is the process of devising and implementing effective corrective actions (CAs) on COE courses and labs to improve on the fulfillment of program outcomes in response to shortcomings detected through ABET assessments.In 2010-2011, the COE ABET Committee developed a framework for integratingContinuous Improvement into the teaching process.

In T111, following the analysis and examination of the previous Rubric Assessment Data and the available indirect assessment data, the committee decided to carry out Continuous Improvement measures for the Program Outcome on Integration of Hardware/Software, i.e., PO (l) (Note that in the SSR, this PO is referred to as PO (n), however, since two POs have been removed, we had to rename this one).This section outlines the proposed continuous improvement for the Hardware and Software Integration in the COE Department

Based on the rubrics assessment through COE 400, COE 485, and COE 351, PO (l) received a score of 2.77 in T062, 071, 072 and a score of 2.68 in T081, which are marginal scores, and accordingly was selected for improvement because of its importance to the COE program.As stated in the program outcomes performance indicators document, Table 1 shows the rubrics that have been used for the Program Outcome on Integration of Hardware/Software, i.e., PO (l).

Table 1.Hardware/Software Integration Rubrics

Outcome	**1** **Exemplary**	**2** **Proficient**	**3** **Apprentice**	**4** **Novice**
Selection of hardware equipment and software	Selection of hardware equipment and software follows a thorough approach where many criterions are used: performance, compatibility, standard compliance, protocol support, interoperability, manufacturer strength.	Selection of hardware equipment and software follows a thorough approach where few criterions are used: performance, compatibility, standard compliance. One or more relevant important criterions are ignored.	Selection of hardware equipment and software is based on the selection of a single manufacturer already integrated solution among several candidates.	Selection of hardware equipment and software is based on the suggestions of the marketing team of one single vendor
Integration Methodology	The integration methodology is well described and followed. Interfaces are well defined and their compatibility discussed. Use of an integration plan featuring integration phases and a test plan for each	The integration methodology is well described and followed. Interfaces are mentioned but their compatibility is not considered. No use of an integration plan. Some mention of a test plan.	The integration methodology is not described properly and not always followed. Interfaces are not mentioned. No use of an integration plan.	An ad-hoc integration (No) methodology is followed but not described. No use of an integration plan. Tests are carried out without

Formatted: No underline, Complex Script Font: Not Bold

Formatted: No underline, Complex Script Font: Not Bold

Formatted: Heading 2

Formatted: No underline, Complex Script Font: Not Bold

Formatted: Justified

		phase.		Tests are carried out without a plan.	a plan.

Although, it is stated in the SSR that three courses are used for assessment of PO (l), i.e., COE 400, COE 485, and COE 351, it has been agreed to change this in the new balancing of the COE Rubric Assessment Process approved by the COE council meeting. In the new balancing, only COE 400 will be used for assessing PO (l) on Hardware/Software integration through student course projects. However, PO (l) will not be assessed in COE 351 and COE 485 because it has been found that this outcome is presently not addressed in these courses. In the newly approved COE B.Sc. curriculum, a newly proposed course on Introduction to Embedded Systems, i.e., COE 305, will cover the aspect of Hardware/Software Integration; and therefore, we propose to incorporate our proposed changed related to this continuous improvement in this course. One major proposed change is the adoption of a mini-project in COE 305.

Assessment Plan

The following action plan has been proposed for CI of PO (l):

- Review the current status of PO (l) including delivery material (Lab, course project, etc.), and point to strengths, and weaknesses.
- Discuss with relevant COE faculty members how to improve the students' knowledge and skills with respect to PO (l).
- Benchmarking and review of similar experiences in other universities.
- Set a target for what to improve in terms of specific curricular actions. We can make use of the approved guidelines for CI, and whether there is a need to develop/adapt teaching material, presentations, guidelines, template, as well as behavioral aspect like students work organization in the lab and method of student working.
- Discuss the proposed curricular actions with relevant COE faculty members.
- Update the proposed curricular actions based on feedback obtained.
- Present the proposed curricular actions to the ABET committee for further discussion and refinements.
- Present the proposed curricular actions to the COE Council for discussion and approval.
- Report the approved curricular actions onto the COE Web-Syllabi.

Scope

The CI process aims at improving the fulfillment of Program Outcome (l) on Hardware/Software Integration of the COE program, which needs improvement and its scope covers the COE-305 course and lab that contribute to this outcome.

Procedure

For the purpose of improving the PO (l), the committee decided to develop a set of Mini-projects so that the concept of integration of hardware and software, i.e., PO (l) is introduced in the COE 305 course. Each Min-Project is to be run for no less than 6 weeks. The students will team out for the project and finally submit a report. Having a set of Mini-projects provides the instructor with different alternatives to choose from. In addition, the committee recommends the implementation of the agreed corrective action plan by the concerned faculty members teaching the COE 305 course.

Proposed Mini Projects for COE-305

Table 2 presents a list of the proposed mini projects for COE-305 (developed with assistance of Dr. Wassim Raad). More details are being worked out for this list of projects.

Table 2. Proposed Mini Projects for COE-305

Projects	**Interdisciplinary Nature**	**Level of Difficulty**	**Novelty**	**Student Interest**	**Hardware Availability**	**Information Available about Implementation**
Automated Voice based Home Navigation System for the Elderly and the Physically Challenged (Arafeh)	High	Medium	High	High	Not readily	Medium
Design and Implementation of Pyroelectric Infrared Sensor Based Security System Using Microcontroller	Low	Low-Medium	Low	Medium	Yes	Medium
Development of an Electronic Stick with Audio Sensory Perception for the Visually Impaired	Medium	Medium	Medium	High	Yes	High
A Novel Light-Sensor-Based Information Transmission System for Indoor Positioning and Navigation (Arafeh)	Low	High	High	High	Not readily	Low

Formatted Table

Electronic Speaking Glove for Speechless Patients - A Tongue to a Dumb	High	High	High	High	Not readily	Low
Automobile Anti-theft System Design based on GSM (Al-Suwaiyan)	Low	Medium	Medium	High	Not readily	Medium
Design of Intelligent Fire Alarm System Based on GSM Network (Al-Suwaiyan)	Low	Low-Medium	Low	Low	Yes, mostly	Low

Time Scope and Follow-up

The CI process initiated for improving performance on theProgram Outcome (l) should be implemented and remain active until the next scheduled program outcome assessment by the ABET Committee. During this time, the ABET committee will continue to provide assistance and support to faculty implementing the approved CAs plan.

Evaluation

At the next scheduled program outcome evaluation, improvements on the Program Outcome (l) targeted by thiscontinuous improvement process will be evaluated. The ABET Committee will study the results of the continuous improvement process and decide if further action is still required and whether the changes introduced by the CAs will be integrated permanently in the courses/labs involved.

Formatted: Body Text

COE ABET ACTIVITIES FOR TERM 102

As per the above action plan for Term 102, the COE ABET committee carried following tasks:

1. **Balancing the COE Rubric Assessment Process: developed a proposal for the balancing the rubric assessment and mapping, presented the proposal in COE council, faculty discussion, and COE council resolution to approve the proposal. See sub-section below describing the above proposal and resolution.**
2. **Continuous Improvements: The ABET committee appointed a faculty to serve as "Faculty in Charge" for improving program outcomes:**
    a. **g(w) "English Writing Skills",**
    b. **Outcome (f) on Engineering Ethics,**
    c. **Outcome (h) on the Awareness of the impact of engineering solutions on Society, and**
    d. **Outcome (j) on knowledge of contemporary issues,**

Note that that outcomes (f), (h), and (j) are not injected in the COE program. They have been assessed in the past in an Ad-Hoc manner. The ABET PEV commented of these issues. In other words , there is a pressing need to inject the above in the program (COE 390), prepare material and teaching method. See sub-section below describing the above proposal and resolution.

Balancing the COE Rubric Assessment Process

Formatted: Font: 16 pt, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

~~**One of the most important ABET-PEVfeedback is related to the mapping between the courses and the Rubrics. The PEV found the COE Rubric assignment to courses as is unbalanced as there are few 400-level courses where almost all outcomes are evaluated.**~~

~~**To address the above feedback, the COE ABET committee has used the following three criteria to come up with a new mapping that will take into account the PEV's feedback:**~~

1. ~~**Balanced rubric assessment: In the new mapping, the maximum number of outcomes per course is 8 (for 400) compared to 11 (for COE 400) in the old mapping. In addition, each outcome is assessed in at most 3 courses (different channels).**~~

2. ~~**Early assessment: Some outcomes could be assessed earlier than 400-level courses such as POs (a) and (b). The student is expected achieved them in the third year of the program. Therefore, we could assess them at the 300-level courses.**~~

3. ~~**Close to graduation assessment for important outcomes: Assess the most important outcomes when the student is close to graduation, i.e. using high level courses. For example engineering design is assessed in COE 351, COE 400, and COE 485.**~~

4. ~~**The three Soft Outcomes ((f): Engineering Ethics, (h): Awareness of the Impact of Engineering Solutions, and (j): Awareness of the Contemporary Issues) must be injected in the program prior to their assessment.**~~

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

5. **Some POs might be assessed by faculty members other than the instructor, e.g., for COE 400.**

In summary, the COE ABET Committee proposes the following revision:

"Mapping for COE PO Assessment"

Math (a) + Experiments (b) → COE 300 level courses

Ethics (f) + Eng. Sol (h) + Contemporary (j) → COE 390 (COE ABET committee will restructure the course after consultation with the instructors, and will provide all necessary material).

Main POs (c)(d)(e)(g)(i)(k) → COE 400 + COE 485 + COE 351

HW/SW (n) → COE 400

Outcomes (l)(m) → Remove

Specifically, the COE ABET Committee recommends the following changes:

1. **PO (a) and PO (b) will be assessed using 300-level courses: COE 308, COE 344, and COE 360**
2. **COE 390 will be used for injecting and assessing the Engineering Ethics (f), Awareness of the Impact of**

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Engineering Solutions (h), and Awareness of the Contemporary Issues (j). The ABET committee will restructure COE 390 and provides all necessary material. This will be part of the current Continuous Improvement Process (2010-2011) which is conducted by Dr. Zubair Baig, a member of the COE ABET Committee.

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

3. **The main program outcomes, i.e. POs (c) (d) (e) (g) (i) (k) will be assessed using COE 400, COE 485, and COE 351.**

4. **For PO (n) on Hardware/Software integration, COE 400 will be assessing this PO through student course projects. However, PO(n) will not be assessed in COE 351 and COE 485 because this outcome is presently not addressed in these courses.**
5. **The two extra POs on (1) Probability and Statistics (n), and (2) the Boolean Algebra (m) will be eliminated from the program outcomes because these outcomes are only indirectly addressed in some COE courses.**

6. **Some POs might be assessed by faculty members other than the instructor, e.g., for COE 400.**

Following the above proposal of the COE ABET committee, the COE Faculty were invited to discuss the above proposal. Dr. El-Maleh said that as per his experience with PEV, before removing any POs, we have to give strong justification. So, for removing (l) & (m) POs, the committee will have to justify in this regard. The chairman inquired whether the POs (l) and (m) are removed from the list of POs or not? Dr. Mayez Al-Mouhamed, chairman of the ABET committee, responded by saying that creating specific assessment rubric will be somehow problematic because

~~these outcomes are only indirectly addressed in the program. Also removing them will not affect the program as all ABET basic outcomes (a-k) are present in the current proposal. The Committee is proposing to remove the extra outcomes l and m because they cannot be assessed in the current implementation of the COE program.~~

~~One of the council members presented the motion to remove l & m POs. Also, the council unanimously agreed on the above as well as improve COE 390 course to address the soft outcomes.~~

~~After some discussions and deliberations, the COE council unanimously agreed to pass the following resolution:~~

Formatted: Font: 16 pt, Complex Script Font: 16 pt

~~**Resolution No. COE/4/25/1431-1432H (2010-2011):**~~

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

~~The COE council unanimously recommends approval for the following changes~~

1. ~~PO (a) and PO (b) will be assessed using 300-level courses: COE 308, COE 344, and COE 360.~~
2. ~~COE 390 will be used for injecting and assessing the Engineering Ethics (f), Awareness of the Impact of Engineering Solutions (h), and Awareness of the Contemporary Issues (j). The ABET committee will restructure COE 390 and provides all necessary material.~~
3. ~~The main program outcomes, i.e. POs (c) (d) (e) (g) (i) (k) will be assessed using COE 400, COE 485, and COE 351.~~
4. ~~For PO (n) on Hardware/Software integration, COE 400 will be assessing this PO through student course projects. However, PO(n) will not be assessed in COE 351 and COE 485 because this outcome is presently not addressed in these courses.~~

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

5. ~~The two extra POs on (1) Probability and Statistics (n), and (2) the Boolean Algebra (m) will be eliminated from the program outcomes because these outcomes are only indirectly addressed in some COE courses.~~
6. ~~Some POs might be assessed by faculty members other than the instructor, e.g., for COE 400.~~

Formatted: Font: 16 pt, Complex Script Font: 16 pt, Not Bold

~~The following Table shows the approved Rubric Assessment and Mapping to the COE Courses:~~

~~**Table Approved Rubric Assessment Mapping to the COE Courses**~~

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Outcome / Course	(a) apply knowledge of mathematics, science, and engineering	(b) design and conduct experiments, analyze and interpret data	(c) design a system, component, or process to meet desired needs	(d) function on multi-disciplinary teams	(e) identify, formulate, and solve engineering problems	(f) understanding of professional and ethical responsibility	(g) effective communications	(h) understanding the impact of engineering solutions	(i) life-long learning	(j) knowledge of contemporary issues	(k) use of techniques, skills, and modern engineering tools in design	(l) Knowledge of probability and statistics and their applications	(m) knowledge of discrete Mathematics	(n) design a system through integration of hardware and software
COE 202 - Digital Logic Design														
COE 203 - Dig. Design Lab														
COE 205 - Comp. Org. & Ass. Lang.														
~~COE 305 - Microcomputer System Design~~		~~A~~												
COE 308 - Computer Architecture	A													
COE 341 - Data & Computer Comm.														
COE 344 - Computer Networks	A	A												
COE 351 - Coop			A	A	A		A		A	A	A			
COE 360 - Principles of VLSI Design.	A													
COE 390 - Seminars						A		A		A				
COE 399 - Summer Training														
COE 400 - System Design Lab			A	A	A		A	A	A		A			A
COE 485 - Senior Design Project			A	A	A		A		A	A	A			
STAT 319														
ICS 252														
IAS 211														
ENGL 214														

Continuous Improvement for the Program Outcome g-W (Writing Skills)

Continuous improvement (CI) is the process of devising and implementing effective corrective actions (CAs) on COE courses and labs to improve on the fulfillment of program outcomes in response to shortcomings identified through ABET assessments.

According to the COE ABET Committee Action Plan for 2010-2011, the ABET Committee should develop a framework for integrating Continuous Improvement into the teaching process. This document outlines the proposed framework for implementing CAs for the g-W (Writing Skills) outcome, which was found to fall short of standards set by the engineering discipline at large, on all students.

Current Situation and Analysis

A thorough investigation was carried out to evidence reasons behind the low quality of technical reports submitted by students as part of course projects and laboratory reports. The following points summarize the findings of the committee:

a. Students did not follow a strict format for their technical reports.
b. Grammar and vocabulary fell short of expectations for a standard technical report.
c. Plagiarism was commonly noticed, especially instances of direct copy-pasting of articles from the Internet.
d. A standard writing template and a mechanism for rating of student writing was not available.

Plan of Action

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt
Formatted: Font: 16 pt, Complex Script Font: 16 pt
Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt
Formatted: Font: 16 pt, Complex Script Font: 16 pt
Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt
Formatted: Font: 16 pt, Complex Script Font: 16 pt

~~**In light of the above shortcomings, the ABET committee agreed that the department needs to take effective measures to address this important requirement that must be possessed by all its students. The following corrective actions were proposed by the committee based on deliberations and analysis.**~~

a. ~~**Provide students with a copy of the attached writing guide, to ascertain that all students adhere to the formatting and template requirement for any writing assignment that they carry out.**~~
b. ~~**Assign (if not done yet) a 20% weight of the entire course to laboratory reports submitted by the students.**~~
c. ~~**A 20% or higher weight of each course must be assigned for the course term project (if a course project is part of the syllabus)** ***p.s. this may already be the case in most courses.***~~
d. ~~**It is suggested that advance-level courses be assigned a higher weight for writing components, based on the discretion of the instructor.**~~
e. ~~**A writing guide, attached herewith, prepared by myself, is to be adhered to by all course instructors for student report evaluation, as well as by all students writing project or laboratory reports. The guide (template) will consist of the following sections, the contents of which will be duly explained within the template, to be followed for each course:**~~
   1. ~~***Abstract***~~
   2. ~~***Introduction & Background***~~
   3. ~~***Technical Content***~~
      a. ~~***Problem Statement***~~
      b. ~~***Design of the Solution***~~
   4. ~~***Experiments/Simulations and Results (including snapshots of simulations)***~~
   5. ~~***Conclusions***~~

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt

6. References

f. **Encourage students to use built-in tools such as Microsoft spell-checker to ensure that spelling mistakes are avoided.**

g. **Provide students with regular reading assignments, to help improve their awareness of the quality of internationally published articles. Such assignments can be evaluated through a review or a summary of the article to be provided by the student. In addition, if such assignments are given on sections/chapters of the textbook, it will also make students aware of the importance of a textbook, and the support that it can provide to them to aid the learning process.**

h. **It is suggested to have a short writing assignment for *COE 3xx* and above courses, to be blind assessed by peer students, wherein the students will gain a strong feel of their level of competence for technical writing within a given class. Such an assignment will also provide the instructors with a general opinion on how a student rates himself, as well as to make students aware of their level of competence in writing, as compared with their peers.**

i. **It is suggested to complement writing skills with enhancement of oral communication skills through video recordings of project demos by peer students, and subsequent presentation of the video in class for critical review by all students.**

The above continuous improvement has been implemented for courses: COE 203, 390, 360, 351, 400, and 485. The Web-Syllabi where the curricular actions were injected is available at:

http://www.ccse.kfupm.edu.sa/coe/?page_id=680&_login=1285a3dd66/

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

Formatted: Font: 16 pt, Bold, Complex Script Font: 16 pt

~~See Appendix A: Components of a high-quality written report for more information.~~

Appendix A: –Mini-Projects to Address the Engineering Design ~~Components of a High-Quality Written Report~~

Formatted: Font: 16 pt, Complex Script Font: 16 pt

COE 203 PROJECT

MINI-PROJECT TITLE

AN XO GAME USING XILINX FPGA

Formatted: Centered

Formatted: Font: 16 pt, Complex Script Font: 16 pt

The project is based on designing & building an XO game for 2 players using a XILINX FPGA and external decoders & colourful LEDs.

An array of 3X3 LEDs will be built on a separate board to mimic the XO game, such that it will take 1 player against the machine player. Each of the players will have one LED with different colour merged into same location on the XO matrix. For each location there will be a unique 4-bit binary number. The computer has to start the game using randomness(Pseudo random numbers) i.e choosing the next hit in a smart way. The winning player will be shown on the LCD and there will also be a timeout if somebody is too slow(out of the game). The decoder will be used externally to cut down the number of outputs going out of the FPGA board. Explore all design techniques to generate the random delay and the computer strategy to play the game.

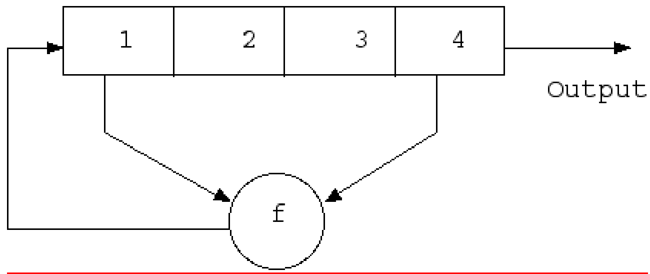
Hint: Use the LFSR (linear feedback shift registers) to generate the random numbers. Tutorial documentation is shown below.

Note: you can find on webct a verilog code for displaying characters on LCD.

Attachment: Linear Feedback Shift Registers

For those unfamiliar with Linear Feedback Shift Registers, see the Primer following Figure 1. For quick information on any of the LFSR parts (shift register, feedback function, output stream, tap sequences) click on the corresponding part in Figure 1.

Figure 1) 4-Bit LFSR, Tap Sequence; [4,1]



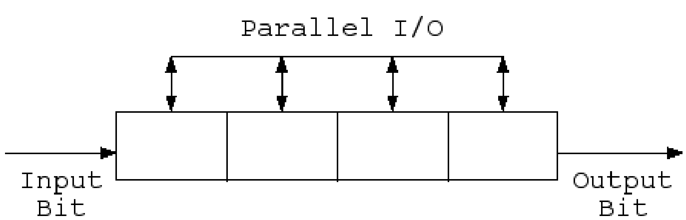
SHIFT REGISTER

One of the two main parts of an LFSR is the shift register (the other being the feedback function). A shift register is a device whose identifying function is to shift its contents into adjacent positions within the register or, in the case of the position on the end, out of the register. The position on the other end is left empty unless some new content is shifted into the register.

The contents of a shift register are usually thought of as being binary, that is, ones and zeroes. If a shift register contains the bit pattern 1101, a shift (to the right in this case) would result in the contents being 0110; another shift yields 0011. After two more shifts, things tend to get boring since the shift register will never contain anything other than zeroes.

Two uses for a shift register are 1) convert between parallel and serial data and 2) delay a serial bit stream. The conversion function can go either way -- fill the shift register positions all at once (parallel) and then shift them out (serial) or shift the contents into the register bit by bit (serial) and then read the contents after the register is full (parallel). The delay function simply shifts the bits from one end of the shift register to the other, providing a delay equal to the length of the shift register.

Figure 2) Shift Register



Some nomenclature:

Clocking) One of the inputs to a shift register is the clock; a shift occurs in the register when this clock input changes state from one to zero (or from zero to one, depending on the implementation). From this, the term "clocking" has arisen to mean activating a shift of the register. Sometimes the register is said to be "strobed" to cause the shift.

Shift direction) A shift register can shift its contents in either direction depending on how the device is designed. (Some registers have extra inputs that dictate the direction of the shift.) For the purposes of this discussion, the shift direction will always be from left to right.

Output) During a shift, the bit on the far right end of the shift register is moved out of the register. This end bit position is often referred to as the output bit. To confuse matters a bit, the bits that are shifted out of the register are also often referred to as output bits. To really muddy the waters, every bit in the shift register is considered to be output during a serial to parallel conversion. Happily, the context in which the term "output" is used generally clears things up.

Input) After a shift, the bit on the left end of the shift register is left empty unless a new bit (one not contained in the original contents) is put into it. This bit is sometimes referred to as the input bit. As with the output bit, there are several different references to input that are clarified by context.

FEEDBACK FUNCTION

In an LFSR, the bits contained in selected positions in the shift register are combined in some sort of function and the result is fed back into the register's input bit. By definition, the selected bit values are collected before the register is clocked and the result of the feedback function is inserted into the shift register during the shift, filling the position that is emptied as a result of the shift.

The feedback function in an LFSR has several names: XOR, odd parity, sum modulo 2. Whatever the name, the function is simple: 1) Add the selected bit values, 2) If the sum is odd, the output of the function is one; otherwise the output is zero. Table 1 shows the output for a 3 input XOR function.

Table 1) XOR Function

Input A	Input B	Input C	XOR Output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

The bit positions selected for use in the feedback function are called "taps". The list of the taps is known as the "tap sequence". By convention, the output bit of an LFSR that is n bits long is the nth bit; the input bit of an LFSR is bit 1.

TAP SEQUENCES

An LFSR is one of a class of devices known as state machines. The contents of the register, the bits tapped for the feedback function, and the output of the feedback function together describe the state of the LFSR. With each shift, the LFSR moves to a new state. (There is one exception to this -- when the contents of the register are all zeroes, the LFSR will never change state.) For any given state, there can be only one succeeding state. The reverse is also true: any given state can have only one preceding state. For the rest of this discussion, only the contents of the register will be used to describe the state of the LFSR.

A state space of an LFSR is the list of all the states the LFSR can be in for a particular tap sequence and a particular starting value. Any tap sequence will yield at least two state spaces for an LFSR. (One of these spaces will be the one that contains only one state -- the all zero one.) Tap sequences that yield only two state spaces are referred to as maximal length tap sequences.

The state of an LFSR that is n bits long can be any one of 2^n different values. The largest state space possible for such an LFSR will be 2^n - 1 (all possible values minus the zero state). Because each state can have only once succeeding state, an LFSR with a maximal length tap sequence will pass through every non-zero state once and only once before repeating a state.

One corollary to this behavior is the output bit stream. The period of an LFSR is defined as the length of the stream before it repeats. The period, like the state space, is tied to the tap sequence and the starting value. As a matter of fact, the period is equal to the size of the state space. The longest period possible corresponds to the largest possible state space, which is produced by a maximal length tap sequence. (Hence "maximal length")

Table 2 is a listing of the internal states and the output bit stream of a 4-bit LFSR with tap sequence [4, 1]. (This is the LFSR shown in Figure 1.)

Table 2) 4-Bit LFSR [4, 1] States and Output

Register States				
Bit 1 (Tap)	Bit 2	Bit 3	Bit 4 (Tap)	**Output Stream**
1	1	0	1	
0	1	1	0	1
0	0	1	1	0
1	0	0	1	1
0	1	0	0	1
0	0	1	0	0
0	0	0	1	0
1	0	0	0	1

1	1	0	0	0
1	1	1	0	0
1	1	1	1	0
0	1	1	1	1
1	0	1	1	1
0	1	0	1	1
1	0	1	0	1
1	1	0	1	0

MAXIMAL LENGTH TAP SEQUENCES

LFSR's can have multiple maximal length tap sequences. A maximal length tap sequence also describes the exponents in what is known as a primitive polynomial mod 2. For example, a tap sequence of 4, 1 describes the primitive polynomial x^4 + x^1 + 1. Finding a primitive polynomial mod 2 of degree n (the largest exponent in the polynomial) will yield a maximal length tap sequence for an LFSR that is n bits long.

There is no quick way to determine if a tap sequence is maximal length. However, there are some ways to tell if one is not maximal length:

1) Maximal length tap sequences always have an even number of taps.
2) The tap values in a maximal length tap sequence are all relatively prime. A tap sequence like 12, 9, 6, 3 will not be maximal length because the tap values are all divisible by 3.

Discovering one maximal length tap sequence leads automatically to another. If a maximal length tap sequence is described by [n, A, B, C], another maximal length tap sequence will be described by [n, n-C, n-B, n-A]. Thus, if [32, 3, 2, 1] is a maximal length tap sequence, [32, 31, 30, 29] will also be a maximal length tap sequence. An interesting behavior of two such tap sequences is that the output bit streams are mirror images in time.

CHARACTERISTICS OF OUTPUT STREAM

By definition, the period of an LFSR is the length of the output stream before it repeats. Besides being non-repetitive, a period of a maximal length stream has other features that are characteristic of random streams.

1) Sums of ones and zeroes. In one period of a maximal length stream, the sum of all ones will be one greater than the sum of all zeroes. In a random stream, the difference between the two sums will tend to grow progressively smaller in proportion to the length of the stream as the stream gets longer. In an infinite random stream, the sums will be equal.

2) Runs of ones and zeroes. A run is a pattern of equal values in the bit stream. A bit stream like 10110100 has six runs of the following lengths in order: 1, 1, 2, 1, 1, 2. One period of an n-bit LFSR with a maximal length tap sequence will have 2^(n-1) runs (e.g., a 5 bit device yields 16 runs in one period). 1/2 the runs will be one bit long, 1/4 the runs will be 2 bits long, 1/8 the runs will be 3 bits long, etc., up to a single run of zeroes that is n-1 bits long and a single run of ones that is n bits long. A random stream of sufficient length shows similar behavior statistically.

3) Shifted stream. Take the stream of bits in one period of an LFSR with a maximal length tap sequence and circularly shift it any number of bits less than the total length. Do a bitwise XOR with the original stream. The resulting pattern will exhibit the behaviors discussed in items 1 and 2. A random stream also shows this behavior.

One characteristic of the LFSR output not shared with a random stream is that the LFSR stream is deterministic. Given knowledge of the present state of the LFSR, the next state can always be predicted.

COE 203 PROJECT

MINI-PROJECT TITLE

Design & implementation of an FPGA based hack-proof authentication protocol for the RFID car immobilizer key

The project aims at following the authentication protocols distributed in class between the RFID key & reader. The first stage is to implement the simple protocol between tag & reader, then go to more secure one adding the manufacturer level.

Generate pseudo random numbers using the LFSR (linear feedback shift register) mechanism. Recommended to use statecad for implementing finite state machine for control unit. Each tag & reader share a 32 bit key or password.

At the end of project each group should successfully demonstrate their project documented with a short report. The student should explore all engineering tradeoffs between performance & security, and explore all various alternative solutions for generating random numbers.

Formatted: Centered

Formatted: Font: 16 pt, Complex Script Font: 16 pt

Writing as a skill is built upon by an individual over a period of time, based on reading, assessment, and analysis skills acquired through team work and effort placed in reading articles on contemporary engineering issues and findings, during their respective undergraduate programs.

Motivation for improving writing skills:

1. It is very convenient to convey your ideas and findings in clear language – understandable by the larger audience.
2. Writing a large collection of jargon will not help you in delivering useful and at times earth-breaking ideas to stakeholders (which include your project partners, instructors, bosses, and the global engineering family).
3. When you write well, you gain more respect from your bosses and colleagues – most high ranked professionals in the corporate world possess strong writing skills, since writing skills are a key factor in promotion decisions.
4. An engineer's inability to write quality reports will invariably affect outcomes of crucial projects, which require clear and easy reporting of findings, diagnoses, investigation results, solution scenarios, test cases, and final results.

Scenario of a poorly written document and its effect:

A person working as a Network Engineer with a local firm writes a report with incomplete details as to the network device deployment locations, types of devices to be used, types of cabling to be used, and human resources that need to be involved in the project. The project manager will not be able to provide the necessary management support for timely completion of such a project. Such delays will not only the affect the company profile and reputation, but will also hinder any progress an engineer can make in his career.

A neat and well-written report will also portray a company as being competitive, and possessing employees who have worked hard through their careers in all aspects (including writing skills), and not just technical abilities.

Every person begins and ends writing a report differently. Some people prefer to clearly write all details related to the problem, solution, and results, and subsequently wish to write the introduction and conclusion. Others work on the introduction first, and subsequently complete the remaining sections in sequential order, based on the table of content.

A standard technical report written by an engineering student consists of the following components:

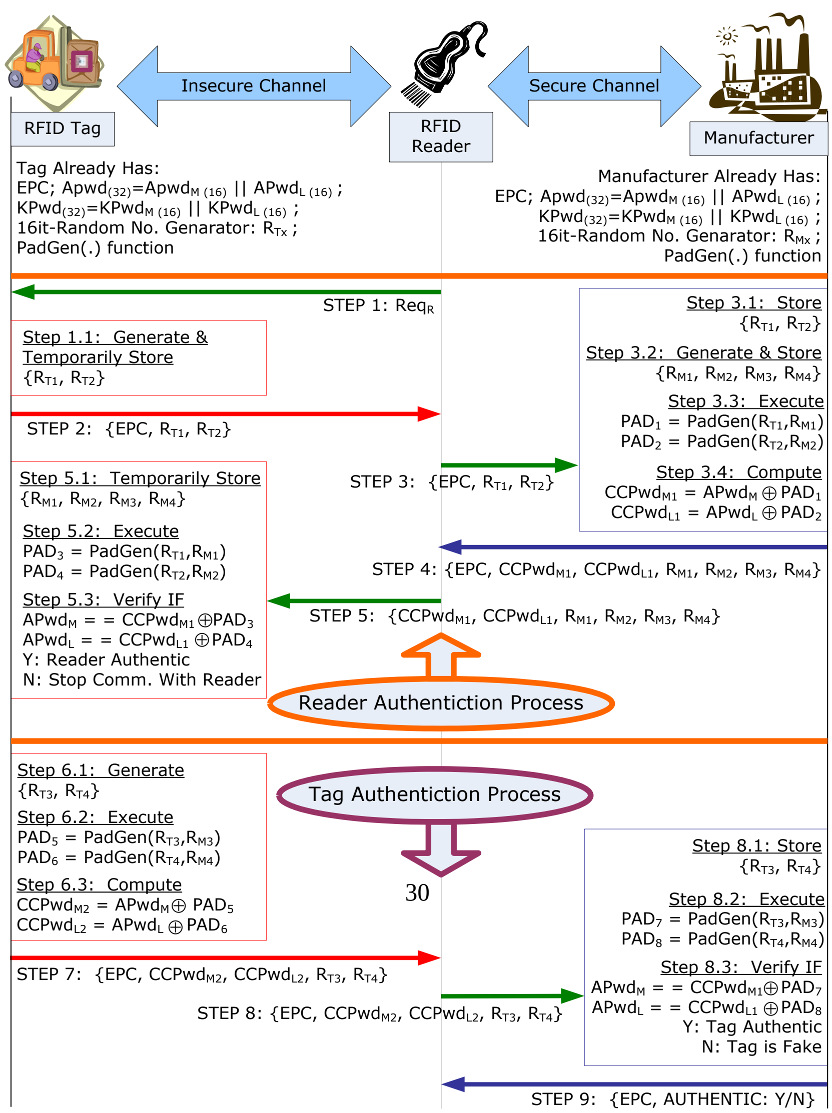
1. **Abstract – around 200 words**
2. **Introduction and Background**

3. ~~Technical Content (this section may have varying titles)~~
4. ~~Experiments and Results~~
5. ~~Conclusions~~
6. ~~References~~

~~Template for Grading a Report~~

		~~Suggested Grade Distribution (out of 100)~~
~~Report Quality & Writing Skills~~	~~Spelling and Grammar~~	~~3%~~
	~~Punctuation~~	~~2%~~
	~~Structure and Organization~~	~~3%~~
	~~Figures and Tables~~	~~3%~~
	~~Formulae & Equations (Proper Use)~~	~~2%~~
	~~Proper Use of References~~	~~2%~~
	~~Proper Use of Appendices (If Applicable)~~	~~-~~
~~Technical Content~~	~~The Abstract~~	~~2%~~
	~~Problem description and motivation (Introduction)~~	~~10%~~
	~~Objectives & Deliverables (Introduction)~~	~~8%~~
	~~Project Management Plan (Introduction)~~	~~10%~~
	~~Technical Content~~	~~20%~~
	~~Experiments/Simulations~~	~~15%~~
	~~Results and Discussion~~	~~15%~~
	~~Conclusions~~	~~2%~~
	~~Overall Quality of Engineering Documentation~~	~~3%~~

Formatted: Body Text


Insecure Channel
Secure Channel
RFID Tag
RFID Reader
Manufacturer
Tag Already Has:
EPC; $Apwd_{(32)}=Apwd_{M\,(16)}$ || $APwd_{L\,(16)}$;
$KPwd_{(32)}=KPwd_{M\,(16)}$ || $KPwd_{L\,(16)}$;
16it-Random No. Genarator: R_{Tx} ;
PadGen(.) function
Manufacturer Already Has:
EPC; $Apwd_{(32)}=Apwd_{M\,(16)}$ || $APwd_{L\,(16)}$;
$KPwd_{(32)}=KPwd_{M\,(16)}$ || $KPwd_{L\,(16)}$;
16it-Random No. Genarator: R_{Mx} ;
PadGen(.) function
STEP 1: Req_R
Step 1.1: Generate & Temporarily Store
$\{R_{T1}, R_{T2}\}$
STEP 2: $\{EPC, R_{T1}, R_{T2}\}$
Step 3.1: Store
$\{R_{T1}, R_{T2}\}$
Step 3.2: Generate & Store
$\{R_{M1}, R_{M2}, R_{M3}, R_{M4}\}$
Step 3.3: Execute
$PAD_1 = PadGen(R_{T1},R_{M1})$
$PAD_2 = PadGen(R_{T2},R_{M2})$
Step 3.4: Compute
$CCPwd_{M1} = APwd_M \oplus PAD_1$
$CCPwd_{L1} = APwd_L \oplus PAD_2$
STEP 3: $\{EPC, R_{T1}, R_{T2}\}$
Step 5.1: Temporarily Store
$\{R_{M1}, R_{M2}, R_{M3}, R_{M4}\}$
Step 5.2: Execute
$PAD_3 = PadGen(R_{T1},R_{M1})$
$PAD_4 = PadGen(R_{T2},R_{M2})$
Step 5.3: Verify IF
$APwd_M == CCPwd_{M1} \oplus PAD_3$
$APwd_L == CCPwd_{L1} \oplus PAD_4$
Y: Reader Authentic
N: Stop Comm. With Reader
STEP 4: $\{EPC, CCPwd_{M1}, CCPwd_{L1}, R_{M1}, R_{M2}, R_{M3}, R_{M4}\}$
STEP 5: $\{CCPwd_{M1}, CCPwd_{L1}, R_{M1}, R_{M2}, R_{M3}, R_{M4}\}$
Reader Authentiction Process
Tag Authentiction Process
Step 6.1: Generate
$\{R_{T3}, R_{T4}\}$
Step 6.2: Execute
$PAD_5 = PadGen(R_{T3},R_{M3})$
$PAD_6 = PadGen(R_{T4},R_{M4})$
Step 6.3: Compute
$CCPwd_{M2} = APwd_M \oplus PAD_5$
$CCPwd_{L2} = APwd_L \oplus PAD_6$
Step 8.1: Store
$\{R_{T3}, R_{T4}\}$
Step 8.2: Execute
$PAD_7 = PadGen(R_{T3},R_{M3})$
$PAD_8 = PadGen(R_{T4},R_{M4})$
Step 8.3: Verify IF
$APwd_M == CCPwd_{M1} \oplus PAD_7$
$APwd_L == CCPwd_{L1} \oplus PAD_8$
Y: Tag Authentic
N: Tag is Fake
STEP 7: $\{EPC, CCPwd_{M2}, CCPwd_{L2}, R_{T3}, R_{T4}\}$
STEP 8: $\{EPC, CCPwd_{M2}, CCPwd_{L2}, R_{T3}, R_{T4}\}$
STEP 9: {EPC, AUTHENTIC: Y/N}

COE 203 PROJECT

MINI-PROJECT TITLE

DESIGN OF A REACTION TIMER USING FPGA

Eye-hand coordination is the ability of the eyes and hands to work together to perform a task. A reaction timer circuit measures how fast a human hand can respond after a person sees visual stimulus.

1-The circuit has three input pushbuttons or micro switches, correspond to clear, start & stop signals. It uses a single discrete LED & displays relevant information on a seven segment LED display.

2- A user pushes clear button to force the circuit to return to the initial state and display welcome message the goes off.

3- When ready, a user pushes start button or micro switch.

4- After a random interval from 2-15 seconds, the LED goes on & timer starts counting upward and increases every millisecond and 0.000 displayed on seven segment display.

5- After LED on, user should try to push stop button or micro switch as soon as possible then timer pauses and seven segment display shows reaction time. For most people it should be between 0.15 to 0.3 sec.

6. if stop is not pushed timer stops after 1 second and displays 1.000.

7. if stop button is pushed before LED goes on, it displays 9.999 and stops.

Implement the following:

Derive control & data units and implement design on FPGA.

Hint: use LFSR (Linear feedback shift registers) to generate random number.

Formatted: Font: 12 pt, Complex Script Font: 12 pt

Formatted: Normal

COE 203 PROJECT

MINI-PROJECT TITLE

TRAFFIC DIRECTING LIGHT SYSTEM

The traffic light (TL) is composed of four faces A, A', B and B' which are intended for use at a four way intersection, where A and A' are two bidirectional roads where A is opposing A'. The same is true for B and B'. Each face is composed of two sets of lights: an 'X' for stop and two connected arrows pointing in the forward and right directions (no left turns for this intersection). The lighting is a set of LEDs, red for the X and green for the arrows.

- The main option for this traffic light is denoted by "Go forward or turn right" which is described below:
  - Enable at time only one forward direction such as A at a time which implicitly means that both the forward AF and turn-right AR directions are enabled. Note that here we may also enable the turn-right of the opposite direction BR because there is no turn left for the above TL.
  - Therefore, the time is split into four parts and each part is allocated to enabling one of four possible states: (AF, AR, and BR), (BF, BR, and AR), (A'F, A'R, and B'R), (B'F, B'R, and A'R) and repeating the above pattern in a round robin fashion.
- The above option can be improved by finding some opportunities to maximize the possibility of non-conflicting turn-rights. Determine this opportunity and rewrite the possible states.
- Now suppose that turn left is allowed for traffic when the forward direction is enabled. In this case, we need to set up the new rule for this traffic light.

Implement the above traffic light of the FPGA board and experience its operation by assuming that a light can be enabled for at most 1 minute. For this you need to use an embedded timer. A normal transition from the green light to the red light must be preceded by a yellow light (warning) for 3 seconds. Include your own analysis for the detection of erroneous functioning and propose a strategy to be used in this case.

COE 305 PROJECT

MINI-PROJECT TITLE

Real-Time EDGE Detection

Formatted: Font: 18 pt, Complex Script Font: 18 pt

Introduction

The project aims at implementing a real-time Sobel Edge Detection (SED to motivate the students about computer vision and digital image processing. Edge detection is extensively used in image segmentation to divide an image into areas corresponding to different objects. Edges occur in parts of the image with strong intensity contrast, which often represent object boundaries. Edges can be detected by applying a high pass frequency filter in the Fourier domain or by convolving the image with an appropriate kernel in the spatial domain. Edge detection is commonly performed in the spatial domain, because it is computationally less expensive and often yields better results. Since edges correspond to strong illumination gradients, the derivatives of the image are used for calculating the edges.

The basic edge-detection operator is a matrix area gradient operation that determines the level of variance between different pixels. The edge-detection operator is calculated by forming a matrix centered on a pixel chosen as the center of the matrix area. If the gradient value of this matrix area is above a given threshold, the middle pixel is classified as an edge. Figure 1 shows the setup of the project.

High Level Design

The camera module captures the image and the raw image data is converted into an RGB color space. The image is then converted to grayscale to obtain the image intensity for edge detection. The grayscale image is mirrored to be displayed properly, and stored in the SDRAM FIFO. Three rows of 640-wide pixels are continuously stored as blocks, and edge detection is initiated once the data is ready.

In digital image processing, each image is quantized into pixels. The first step in edge detection is to convert the raw data to a grayscale image, where each pixel indicates the level of brightness of the image: from 0 representing black to 1023 representing white, with a 10-bit wide pixel. The image is then threshold-ed to create a clear gradient. Edge information for a particular pixel is obtained by exploring the brightness of pixels in its neighborhood. If all of the pixels in the neighborhood have the same brightness, it indicates that there is no edge in the area. However, if some of the neighbors are much brighter than the others, it indicates that there is an edge present.

Measuring the relative brightness of pixels in a neighborhood is mathematically analogous to calculating the derivative of brightness. The SED algorithm uses a 3x3 convolution table to store a pixel and its neighbors to calculate the derivatives. The table is moved across the image, pixel by pixel. For a 640x480 image, the convolution table will move through 302964 (638x478) different locations because we cannot calculate the derivative for pixels on the perimeter of the image.

The SED algorithm identifies both the presence of an edge and the direction of the edge. There are eight possible directions: north, northeast, east, southeast, south, southwest, west, and northwest.

For a convolution table, calculating the presence and direction of an edge and is done in three major steps:

1. Calculate the derivative along each of the four orientations. The equations for the derivatives are written in terms of elements of a 3*x*3 table.

```
DerivNE_SW =  (table[0,1] + 2* table[0,2] + table[1,2])  (table[1,0] + 2* table[2,0] + table[2,1] )
DerivN_S     =  (table[0,0] + 2* table[0,1] + table[0,2]) - (table[2,0] + 2* table[2,1] + table[2,2] )
DerivE_W     =  (table[0,2] + 2* table[1,2] + table[2,2]) - (table[0,0] + 2* table[1,0] + table[2,0] )
DerivNW_SE =  (table[1,0] + 2* table[0,0] + table[0,1]) - (table[2,1] + 2* table[2,2] + table[1,2] )
```

2. Find the value and direction of the maximum derivative, and the absolute value of the derivative that is perpendicular to the maximum derivative.

EdgeMax = Maximum of absolute values of four derivatives
DirMax = Direction of EdgeMax
EdgePerp = Absolute value of derivative of direction perpendicular to DirMax

3. Check if the maximum derivative is above the threshold. When comparing the maximum derivative to the threshold, the Sobel algorithm takes into account both the maximum derivative and the derivative in the perpendicular direction.

```
if EdgeMax + EdgePerp/8 >= threshold then
    Edge = true
    Dir = DirMax
else
    Edge = false
    Dir = 000
```

Formatted: Left, Indent: Before: 1.27 cm

Project Description

The students are required to implement a real-time Sobel Edge Detection (SED) on an FPGA module. Traditionally, SED is done on still images. This project may use a Camera connected to a PC to continuously capture images, transfer an image or part of an image to the FPGA, SED is run in real-time on an FPGA, the resulting image is uploaded back to the PC, and displayed the monitor. This motivates student about computer vision and digital image processing.

Formatted: Justified, Indent: Before: 0 cm

Appendix:

Conversion of a RGB Image to Grayscale

In order to perform edge detection, we would need to convert the RGB image from the camera to a grayscale image. In a grayscale image, the green component is required to be higher in intensity. We first tried to perform the conversion by scaling down the RGB image:

Grayscale = 0.3*red + 0.6*green + 0.1*blue

The closest that represents this to implement on hardware efficiently is as follows:

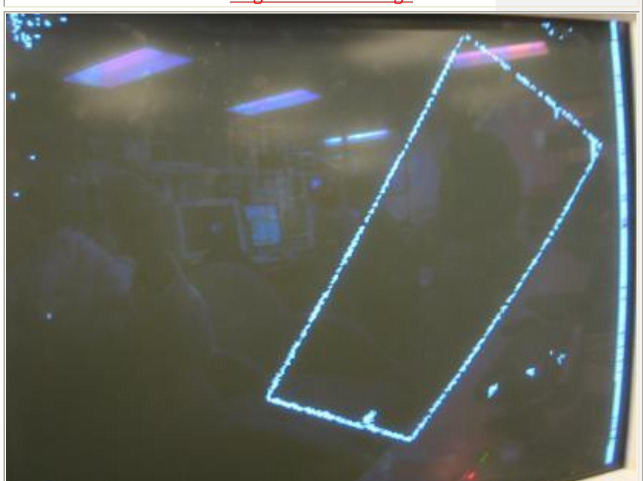
Grayscale = red>>2 + green>>1 + blue>>2

However, the resulting image did not achieve enough contrast for edge detection. After several refinements, it is discovered that scaling up the green give the best result for the DE2 camera.

grayscale = red + green<<1 + blue

Results

The following images show the result of the threshold grayscale image before edge detection, and the result after edge detection. The following image shows the camera pointing towards to the ceiling to edge detect the light. We can observe that the clear outline of the light is detected. It was unable to detect the details inside the light. This is due to the fact that edge detection is performed on a grayscale scale. In grayscale, it is difficult to threshold the difference between areas with high brightness.

Actual Image	Edge Detection Image
	

Formatted: Body Text

COE 305 PROJECT
MINI-PROJECT TITLE
AUTOMATIC ELEVATOR

The project is based on designing & building an FPGA Module to Control the operation of an 8-floor elevator which we denote by E.

Elevator State Vector E

The state of the elevator E is defined by an 8-bit state vector denoted as S[i], where S[i]=1 if the elevator is currently at the ith floor of the building. We always must have some component of s[] being 1 and no more than one component can be 1 at any given time.

Elevator Desired Button Vector D

We also have inside the elevator room 8 binary entries denoted by D[i] to allow the users to select the desired floor. Entries D[i]=1 if one user presses the button D[i] at any time, D[0] is the ground floor and d[7] is the seventh floor. For example D[]=00001110 if there are users to go to floor 4, 5, and 6. Note the following:

- If D[i]=1 and now S[i]=1, then we can clear D[i]=0 because E is now at the ith floor. In other words when E state changes to a new state (like S[i-1]=0 and s[i]=1) then we must check D[i] and clear it. We assume the service of the ith floor is satisfied when E passes by the ith floor.
- It clear that D[] must be considered as input to decide how to move E up or down. Another important input is the E state which is S[i]. Therefore the next state of E^+ must be function of E^+ (D[],E[]). For example D[]=00001110 and E[0]=1 (meaning E[]=10000000), then E must move up to serve floors 4, 5, and 6. Therefore $E^+(1)=1$ to indicate the E moves to 1st floor and it is no more at ground floor E[0]=0.

Simple Solution

Note that the next state E^+ must be decided based upon the ith position satisfying E[i]=1 and vector D[] . A generic approach would be to move up if "the number of 1s to the right of the ith position in D[]" is greater than "the number of 1s to the left of the ith position in D[]". Lets' use the following notation:

- Variable Left is "the number of 1s to the left of the ith position in D[]", Left can be 0 if S[0]=1 where E is at the ground Floor.

- Variable Right is "the number of 1s to the right of the ith position in D[]", Right can be 0 if S[7]=1 where E is at the 7th Floor.

- The above suggest that to the change the state of E we need to compute variables Left and Right. And compare them. If Left =< Right then E must move up (more cabin users in upper floors) to serve the largest number of cabin users. On the other hand, If Left > Right then E must move down to serve the largest number of cabin users in lower floors.

Engineered Solutions

A more elaborated and optimized Service Approach is decide (move up or down) the next state E^+ based upon (Left , Right, and R[i]), where R[] is a 8-component request vector. R[i]=1 corresponds to the elevator request button which is available (outside the elevator) at the ith floor. Each component R[i] can be in one of 3 states:

- 0 (no request),

- +1 if there is a request to move to a floor which is above the ith floor, and

- -1 if there is a request to move to a floor which is below the ith floor.

Therefore, the next state E^+ based be function of (E, Left , Right, R[]), i.e E^+ should be function of current state E, the number of cabin users asking to go to the lower floors (Left) and those asking to go to upper floors (Right), and the request vector E for the waiting users at the various floors.

Miniproject Objective Functions

Design the automatic elevator controller by deciding how to update the state of E. In other words the students are requested to determine the relationship that allow what should be E^+ based on function of (E, Left , Right, R[]) to satisfy some optimization criteria like:

- Have a very simpler criteria for evaluating E^+ = (E, Left , Right, R[])

- An optimized criteria like evaluating E^+ = (E, Left , Right, R[]) to serve the largest number of users.

- An optimized criteria like evaluating E^+ = (E, Left , Right, R[]) to make the least number of travels to serve all the request from inside the cabin and from the various floors.

- An optimized criteria of your choice.

Appendix B: Mini-Projects: Integration of Hardware/Software

Formatted: Font: 16 pt, Complex Script Font: 16 pt

Formatted: Body Text

Structure and Organization

1. Consistent use of fonts for titles, sub-titles, chapters, sub-chapters, sections, sub-sections, to enhance the readability and understanding of the report.
2. Consistent and correct display of information in the Table of Content, List of Figures and the List of Tables.

The Abstract

1. Should not be longer than a single paragraph (2 paragraphs for a longer report).
2. Outlines the importance of what you have accomplished through the project or assignment.
3. Contains a summary of all work done.
4. Contains a summary of results obtained through experiments or simulation.
5. Try to write the abstract after the entire report is written.
6. Is a strong symbol of what will follow in the rest of the report (so work hard on writing a good one!).

The Introduction

1. Provides all the background information required for understanding your project work as well as the results.
2. Contains several statements to describe the motivation behind the project.
3. Contains references to past work done in this area. For example, for a project on Computer Network Routing, the references must include recent work on computer networking as well as routing. A reference on digital system design is not relevant to this project.
4. Clearly write down your hypothesis in the Introduction. The hypothesis is a statement that can either be proven correct or incorrect through the work accomplished by your self

through the project. An example of a hypothesis is: '***If a router placed at the fourth floor of a 9 floor high building is connected to two other routers on the same floor, with the same data transfer rate, then the overall computer network of the building will have a 4 times higher throughput***'.

5. Clearly mention the significance of the problem that is addressed through this project, the number of team members involved, and how this team can help achieve the goals of the project.

Technical Content

This section contains a detailed set of information associated with the project. For instance, a project on computer network routing for a university campus will contain the following sections or subsections: Campus floor plan, Type and Quantity of the networking devices used, Configuration of all switches and routers, the network topology, statistics on usage of the campus network, and miscellaneous technical details associated with such a project.

Experiments and Simulations

1. Provides detailed text to describe all procedures, techniques, and resources used for running the experiments or simulations for the project.
2. Contains complete referencing to known techniques for performing experiments and simulations. E.g. ***The experiment was conducted on a Xilinx DS09 Spartan03 board, with two additional block RAMs. A total of 109 logic gates were connected to form a circuit associated with a traffic light control system. 10 of these gates were logical AND gates, ……***
3. Contains schematic diagrams of all experiments (such as that of a logic circuit design), and other diagrams.

Results

1. Contains all experimental or simulation results either tabulated or in the form of a graph.
2. Contains all the necessary information that a reader of the report may look for.
3. Clear figures and tables will help the reader appreciate your work better.
4. Contains both usual (expected) as well as unusual (anomalous) results. Do not hesitate to report negative results of an experiment or simulation.

An example of a decent snapshot of a network topology and its accompanying simulation results*:

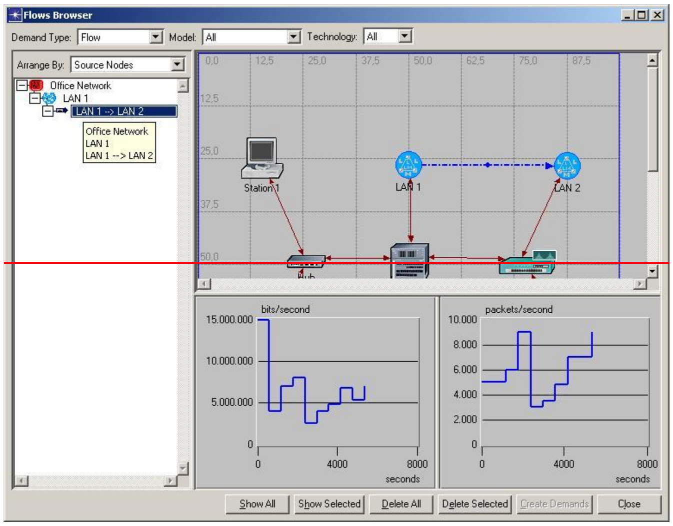


Figure 1. A network topology for a single Local Area Network deployed inside a 3-storey hotel, accompanied with results on the data rate experienced by this network for two different scenarios (Src: http://homes.esat.kuleuven.be)

Summary or Conclusions

1. Contain a brief summary of technical aspects associated with the project.
2. Contains a summary of achievements, problems faced, and solutions proposed.
3. Contain a brief outline of the results obtained from experiments or simulations.
4. May contain future directions of work to extend the work already done through this project.

References

Use standard templates for listing all resources that were referred to in the project report. A commonly used template is the IEEE template, found at:

http://www.ieee.org/documents/ieeecitationref.pdf

Appendices

1. Contain miscellaneous content that may help the user better understand your accomplishments.

Examples of such content include: Source code, Large set of snapshots for experiments or simulation, maps, large scale tabulated data, etc.

Using the newly developed PEOs, the ABET committee developed surveys for (1) exposing the new PEOs to all program constituents, and (2) offer them the opportunity for reformulating the PEOs. The results of these surveys are presented next.

~~Appendix B. Writing Guide for Project and Term Reports~~

~~A standard technical report written by an engineering student must include the following components:~~

~~1. Abstract – around 200 words~~

~~2. Introduction and Background~~

~~3. Technical Content (this section may have varying titles)~~

~~4. Experiments and Results~~

~~5. Conclusions~~

~~6. References~~

~~Template for Grading a Report~~

		Suggested Grade Distribution (out of 100)
Report Quality & Writing Skills	Spelling and Grammar	3%
	Punctuation	2%
	Structure and Organization	3%
	Figures and Tables	3%
	Formulae & Equations (Proper Use)	2%
	Proper Use of References	2%
	Proper Use of Appendices (If Applicable)	-
Technical Content	~~The Abstract~~	~~2%~~
	Problem description and motivation (Introduction)	10%
	Objectives & Deliverables (Introduction)	8%
	Project Management Plan (Introduction)	10%
	Technical Content	20%
	Experiments/Simulations	15%
	Results and Discussion	15%
	Conclusions	2%
	Overall Quality of Engineering Documentation	3%

Formatted: Heading 2, Left, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Indent: Before: 0 cm, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2

Structure and Organization

1. Consistent use of fonts for titles, sub-titles, chapters, sub-chapters, sections, sub-sections, to enhance the readability and understanding of the report.

2. Consistent and correct display of information in the Table of Content, List of Figures and the List of Tables.

Formatted: Heading 2, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Indent: Before: 0 cm, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

The Abstract

1. Should not be longer than a single paragraph (2 paragraphs for a longer report).

2. Outlines the importance of what you have accomplished through the project or assignment.

3. Contains a summary of all work done.

4. Contains a summary of results obtained through experiments or simulation.

5. Try to write the abstract after the entire report is written.

6. Is a strong symbol of what will follow in the rest of the report (so work hard on writing a good one!).

Formatted: Heading 2, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Indent: Before: 0 cm, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

The Introduction

1. Provides all the background information required for understanding your project work as well as the results.

2. Contains several statements to describe the motivation behind the project.

3. Contains references to past work done in this area. For example, for a project on Computer Network Routing, the references must include recent work on computer networking as well as routing. A reference on digital system design is not relevant to this project.

4. Clearly write down your hypothesis in the Introduction. The hypothesis is a statement that can either be proven correct or incorrect through the work accomplished by your self through the project. An example of a hypothesis is: '*If a router placed at the fourth floor of a 9 floor high building is connected to two other routers on the same floor, with the same data transfer rate, then the overall computer network of the building will have a 4 times higher throughput***'.**

5. Clearly mention the significance of the problem that is addressed through this project, the number of team members involved, and how this team can help achieve the goals of the project.

Formatted: Heading 2, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Indent: Before: 0 cm, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Technical Content

This section contains a detailed set of information associated with the project. For instance, a project on computer network routing for a university campus will contain the following sections or subsections: Campus floor plan, Type and Quantity of the networking devices used, Configuration of all switches and routers, the network topology, statistics on usage of the campus network, and miscellaneous technical details associated with such a project.

Experiments and Simulations

1. Provides detailed text to describe all procedures, techniques, and resources used for running the experiments or simulations for the project.

2. Contains complete referencing to known techniques for performing experiments and simulations. E.g. *The experiment was conducted on a Xilinx DS09 Spartan03 board, with two additional block RAMs. A total of 109 logic gates were connected to form a circuit associated with a traffic light control system. 10 of these gates were logical AND gates, ……*

3. Contains schematic diagrams of all experiments (such as that of a logic circuit design), and other diagrams.

Results

1. Contains all experimental or simulation results either tabulated or in the form of a graph.

2. Contains all the necessary information that a reader of the report may look for.

3. Clear figures and tables will help the reader appreciate your work better.

4. Contains both usual (expected) as well as unusual (anomalous) results. Do not hesitate to report negative results of an experiment or simulation.

Summary or Conclusions

1. Contain a brief summary of technical aspects associated with the project.

2. Contains a summary of achievements, problems faced, and solutions proposed.

3. Contain a brief outline of the results obtained from experiments or simulations.

4. May contain future directions of work to extend the work already done through this project.

Formatted: Heading 2, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Indent: Before: 0 cm, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Indent: Before: 0 cm, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Indent: Before: 0 cm, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

References

Use standard templates for listing all resources that were referred to in the project report. A commonly used template is the IEEE template, found at:

http://www.ieee.org/documents/ieeecitationref.pdf

Appendices

Contain miscellaneous content that may help the user better understand your accomplishments. Examples of such content include: Source code, Large set of snapshots for experiments or simulation, maps, large scale tabulated data, etc

Appendix C: Sample Web Syllabi for COE 390

King Fahd University of Petroleum and Minerals

College of Computer Sciences and Engineering

Department of Computer Engineering

Standard Syllabus for COE 390: Seminar (1-0-1)

General:

Course Code: **COE 390**

Formatted: Heading 2, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Heading 2, Left

Formatted: Heading 2

Title: **Seminar**

Credit Code: **1-0-1**

Pre-requisite(s): **Junior Standing**

Catalog Description:

The purpose of this course is to help improve students' ability for presenting their technical work. It also teaches students about the nature of engineering as a profession, codes of professional conducts, ethics & responsibility, and the role of engineering societies and organizations world-wide. Case studies of conflict between engineering professional ethical values and external demands. The course features students' participation in discussion held by COE faculty members and invited guests. Each student is required to deliver a short talk toward the end of the semester.

Textbook:

There is no assigned text book for this course. However, the following references are highly recommended:

1. Bowyer, Kevin W. *Ethics and Computing*. IEEE Computer Society Press, 1996.

2. Johnson, D. G. *Computer Ethics.* Prentice Hall, Englewood Cliffs, NJ, 1994.

3. Kizza, Joseph M. *Ethical and Social Issues in the Information Age*. Springer, 1997.

4. Gary Kroehnert *Basic Presentation Skills*, McGraw-Hill, 1999.

5. Nido R. Qubein *How to Be a Great Communicator: In Person, on Paper, and on the Podium*, John Wiley & Sons, 1996.

6. http://www.businessballs.com/presentation.htm

7. http://ethics.csc.ncsu.edu/

Grading Policy:

- **First Presentation** 10%
- **Second Presentation** 10%
- **Short Essay on Computing Ethics (Rubrics)** 20%
- **Short Essay on Impact of Eng. Solutions (Rubrics)** 20%
- **Short Essay on Contemporary Issues (Rubrics)** 20%

Formatted: Heading 2, Space Before: 0 pt, After: 0 pt

Formatted: Heading 2

Formatted: Heading 2, Widow/Orphan control

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Widow/Orphan control

Formatted: Heading 2

- Attendance & Critique of Two Seminars 10%
- Attendance of One Invited Talk by a Guest Speaker 10%

Course Objectives:

After successfully completing the course, students will be able to:

1. **Make effective presentation,**
2. **Engage in long-life learning,**
3. **Understand the impact of engineering solutions in a global and social context,**
4. **Appreciate the importance of ethics in engineering, and**
5. **Identify some contemporary issues related to computing.**

Course Learning Outcomes and Indicators:

Course Learning Outcomes	Outcome Indicators and Details	Assessment Methods and Metrics	Min. Weight
O1. Knowledge of professional and ethical responsibility	• **Selecting a topic related to computing ethics** • **Writing a Short essay and making a presentation of the selected topic**	• **First presentation** • **Short essay**	**25%**
O2. Ability to make effective presentation	• **Use of professional tools like PowerPoint to prepare presentations** • **Learning about presentation skills by reading and attending seminars organized by the different departments inside KFUPM**	• **Two presentations**	**55%**
O3. Knowledge of contemporary issues	• **Selection of a recent technical topic in the field of computer engineering**	• **Approval of instructor**	**N/A**

Formatted: Heading 2, Left, Space Before: 0 pt, Tab stops: Not at 3.17 cm

Formatted: Heading 2

Formatted: Heading 2, No bullets or numbering

Formatted: Heading 2

Formatted: Heading 2, Left

Formatted: Heading 2

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm

Formatted: Heading 2, Left

Formatted: Heading 2, Tab stops: Not at 3.17 cm

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Tab stops: Not at 0.63 cm

Formatted: Heading 2, Left

Formatted: Heading 2

Formatted: Heading 2, Tab stops: Not at 3.17 cm

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Tab stops: Not at 0.6 cm + 1.83 cm + 6.36 cm

Formatted: Heading 2, Left

O4. Understanding the impact of engineering solutions in a global and social context	• Presentation and discussion • Attendance of two seminars and criticizing two seminars delivered by professionals	• Second presentation • Critique of Two Seminars	50%
O5. Ability to engage in self-learning	• Demonstrate the ability to identify and retrieve facts • Demonstrate the ability to apply creative thinking • Demonstrate the ability to criticize and recommend alternatives	• Second presentation	40%

Weekly breakdown of Lecture Course Material

Weeks	Lecture	Activity
1	*Lecture* on Communication Skills for Engineers: Students will be introduced to the use of professional presentation and document writing skills necessary for the engineering discipline including oral, written and body language *Based on Instructor Discretion one presentation is to be selected out of three provided below as links- Talk will last for roughly 50 minutes* 1. Writing Skills Presentation 1 2. Writing Skills Presentation 2 3. Writing Skills Presentation 3 The course instructor is kindly requested to carry out the following actions: 1. Inform the students that their English Writing Skills will be given 10% of overall submitted	Guidelines on Communication Skills is to presented and discussed with the students

Formatted: Heading 2, Tab stops: Not at 3.17 cm
Formatted: Heading 2, Left, Indent: Before: 0 cm, First line: 0 cm, Tab stops: Not at 0.67 cm + 3.17 cm
Formatted: Heading 2, Left
Formatted: Heading 2
Formatted: Heading 2, Left, Tab stops: Not at 3.17 cm
Formatted: Heading 2, Left, Tab stops: Not at 3.17 cm
Formatted: Heading 2, Left, Indent: Before: 0 cm, First line: 0 cm, Tab stops: Not at 0.71 cm + 3.17 cm
Formatted: Heading 2
Formatted: Heading 2, Left
Formatted: Heading 2
Formatted: Heading 2, Left
Formatted: Heading 2, Left
Formatted: Heading 2
Formatted: Heading 2
Formatted: Heading 2, No bullets or numbering
Formatted: Heading 2
Formatted: Heading 2
Formatted: Heading 2, No bullets or numbering

	assignment grades based on returned reports. 2. Request the students to follow the below attached template. 1. Template for Project Report 3. Grade the students writing according to the following guidelines: ABET Committee's Writing Guide 4. Evidence of graded student writing skill report/assignment is to be kept with instructors for use in future ABET visits.	
2	*Lecture* on Introduction to Engineering (Computing) Ethics: Students are introduced to the significance of ethically performing engineering practice. An assignment such as the benefits and effects of the use of 'Software Piracy', Copyright issues, Intellectual Property, Reeferencing Other's Work (quote, copy of drawings and figures), etc. on both the engineer as well as other stakeholders *Based on Instructor Discretion one presentation is to be selected out of three provided below as links. Talk will last for roughly 50 minutes* 1. Ethics Presentation 1 2. Ethics Presentation 2 3. Ethics Presentation 3	• The class may be divided into two groups, with the students required to prepare a 3-4 page report on the implication of engineering ethics on a given problem, such as the use of unlawful means to change grades in a grading server. Secondly, the students prepare a 2-minute presentation
3	*Lecture* on Engineering (Computing) Ethics – continued Eng. Ethics is to be injected and assessesd in COE 390: there must be one rubrics (stu assignemtn to be graded and get rubric score)	• Students present a debate-style lecture based on the previous week's preparation and submit their reports • Give a 4-page writing assignment to every student on the above issue to serve as a rubrics on Engineering Ethics
4	*Lecture* on Introduction to Contemporary Issues in Engineering: Students are introduced to the opportunities and areas of application of technical material covered in the Computer Engineering	Students carry out research on a contemporary issue of choice and prepare a 3-4

Formatted: Heading 2, Indent: Before: 0 cm

Formatted: Heading 2, No bullets or numbering

Formatted: Heading 2, Left

Formatted: Heading 2

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Space After: 0 pt, Add space between paragraphs of the same style, Line spacing: single, Tab stops: Not at 1.27 cm

Formatted: Heading 2, Left

Formatted: Heading 2

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Space After: 0 pt, Add space between paragraphs of the same style, Line spacing: single

Formatted: Heading 2, Left

Formatted: Heading 2

Formatted: Heading 2

	discipline. They are introduced to areas of concern in the COE discipline, faced by the local industry, as well as by the global community of computerengineers. *Based on Instructor Discretion one presentation is to be selected out of three provided below as links- Talk will last for roughly 50 minutes* 1. Contemporary Issues Presentation 1 2. Contemporary Issues Presentation 2 3. Contemporary Issues Presentation 3	page report on a topic of their choice, relevant to addressing contemporary issues faced by Computer Engineers, as well as a brief 2-minute presentation
5	*Lecture* on Contemporary Issues in Engineering continued Contemporary Issues in Engineering is to be injected and assessesd in COE 390: there must be one rubrics (stu assignment to be graded and get rubric score)	• 2-minute presentation is delivered by each student on contemporary issues faced by Computer Engineers in KSA (support using data/statistics), and the report is submitted. • Give a 4-page writing assignment to every student on the above issue to serve as a rubrics on **contemporary issues**
6	*Lecture* on the Impact of Engineering Education on Society: Students are introduced to the effect that the solutions proposed by computer engineers, can have on the society at large. For instance, the use of wireless sensor nodes for monitoring the health of a patient, if deployed irresponsibly, may cause the death of a subject. Or the use of inefficient Programmable Logic Arrays (PLAs), incorrectly programmed, may cause production disruptions and incur financial losses to a company. *Based on Instructor Discretion one presentation is to be selected out of three provided below as links- Talk will last for roughly 50 minutes* 1. Impact of Engineering Presentation 1 2. Impact of Engineering Presentation 2	Students are expected to write a 3-4 page report on an issue that they feel will impact the society at large, and can be effectively addressed by a computer engineer

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Space After: 0 pt, Add space between paragraphs of the same style, Line spacing: single, Tab stops: Not at 1.27 cm

Formatted: Heading 2, Left

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Space After: 0 pt, Add space between paragraphs of the same style, Line spacing: single

Formatted: Heading 2

Formatted: Heading 2, Left

Formatted: Heading 2

Formatted: Heading 2

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Space After: 0 pt, Add space between paragraphs of the same style, Line spacing: single, Tab stops: Not at 1.27 cm

	3. Impact of Engineering Presentation 3	
7	*Lecture* on the Impact of Engineering Education on Society (Cont) Impact of Engineering Education on Society is to be injected and assessed in COE 390: there must be one rubrics (student assignment to be graded and get rubric score)	• Students prepare a 2-page paper style proposal on how to effectively write a research paper, as well as how to prepare and deliver a 5-minute presentation • Give a 4 page writing assignment to every student on the above issue to serve as a rubrics on contemporary issues
8	*Lecture* on how to write a proposal or a research paper	Students prepare a 2-page paper style proposal on how to effectively write a research paper, as well as how to prepare and deliver a 5-minute presentation
9-10 – 9	*Student Activity* – class activity to enhance previous concepts and to provide feedback for the student writing assignments.	Students submit their business propositions and deliver their 5-minute presentations
10	*Lecture* on Professional Ethics	Students prepare their final proposal and presentations based on received feedback
11	*Lecture* on Introduction to Engineering Design	
12	*Lecture* on Product Development Cycle/Project Management	
13-14	*Invited Lectures* delivered by Faculty from either within or from outside the Department	Students Prepare for their Final Presentations
15	*No Lecture*	Students deliver their final presentations (3-hour long class)

Formatted: Heading 2, Left
Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Space After: 0 pt, Add space between paragraphs of the same style, Line spacing: single
Formatted: Heading 2
Formatted: Heading 2, Left
Formatted: Heading 2
Formatted: Heading 2, Left
Formatted: Heading 2
Formatted: Heading 2, Left
Formatted: Heading 2
Formatted: Heading 2, Left
Formatted: Heading 2
Formatted: Heading 2, Left
Formatted: Heading 2
Formatted: Heading 2, Left
Formatted: Heading 2
Formatted: Heading 2, Left
Formatted: Heading 2
Formatted: Heading 2

Description of assignments other than HWs, e.g. term papers, projects, case studies, etc. (if Applicable)

Assignment	Description	% of Grade
Selecting a topic for the first presentation	**Each student is required to select a topic related to computing ethics and write a summary paper and make a presentation of the selected topic. The summary paper should follow the given paper template. Each student will be given 15 minutes to present his paper. Computing ethics topics that need to be covered include:** - Intellectual Property**: copyright laws, patenting laws, software piracy, and related topics.** - Privacy and Anonymity**: email privacy, privacy on the web, encryption, and related topics.** - Computer Abuse and Crime**: hacking, worms, viruses, trojan horses, spamming, and related topics.** - Commerce**: anticompetitive practices, antitrust law, online auctions, fraud, trade, cybersquatting, payment, web ads, and related topics.** - Speech issues**: freedom, misinformation, netiquette, blogs, chain letters, and related topics.** - Social Justice issues**: environmental, equity, noise, workplace, depersonalization, and related topics.** - Rules of practice for Engineers: **competency, objectivity, truthfulness, faithfulness, protection of the public health, safety, and welfare, and related topics.** - Professional obligations for Engineers: **highest standards of honesty and integrity, respect of confidentiality, service to the public interest, and related topics.**	**20%**
Second Presentation	**Every student is to select a recent technical topic in the field of computer engineering and prepare a professional presentation applying effective presentation techniques learned in class. Each presentation will be given 15 minutes. The student is required to select three articles, among which the instructor will choose one for**	**40%**

Formatted: Heading 2

Formatted: Heading 2, Left

Formatted: Heading 2, Left, Indent: Before: 0 cm, First line: 0 cm, Tab stops: Not at 0.64 cm

Formatted: Heading 2, Widow/Orphan control

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Widow/Orphan control

Formatted: Heading 2, Space Before: 0 pt

Formatted: Heading 2, Left, Indent: Before: 0 cm, Tab stops: Not at 0.64 cm

Formatted: Heading 2, Indent: Before: 0 cm, Tab stops: Not at 0.64 cm

Formatted: Heading 2, Widow/Orphan control

Formatted: Heading 2, Space Before: 0 pt

Formatted: Heading 2, Left, Indent: Before: 0 cm, Tab stops: Not at 0.64 cm

	presentation in the class. Articles should be related to computer engineering and should be 4 pages or more. Recent (within the last three years) issues of the following publications may be used. Other sources may not be used except with the explicit approval of the instructor. - IEEE Spectrum - IEEE Computer Magazine - Communications of the ACM - IEEE Network Magazine - Scientific American ACM, IEEE, SIAM, AT&T, BT, Intel, or IBM journal articles.	

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Widow/Orphan control

Formatted: Heading 2, Widow/Orphan control

Formatted: Heading 2

Note: The Course Instructor is required to assign and subsequently grade each of the following 3 rubrics (as per requirements established by the ABET committee):

Formatted: Heading 2

1. R1: Engineering Ethics
2. R2: Contemporary Issues
3. R3: Impact of Engineering Solutions on Society

Formatted: Heading 2, Indent: Before: 0 cm, First line: 0 cm, Space After: 0 pt, Add space between paragraphs of the same style, Line spacing: single, Tab stops: Not at 2.54 cm

Formatted: Heading 2

ABET Continuous Improvement Record

Semester	Date	Action	Faculty-In-Charge
T102 (Spring 2011)	April 16, 2011	Prepared standard syllabus	Dr Yahya Osais
T102 (Spring 2011)	June 13, 2011	Updated the Syllabus and Injected a new Weekly Breakdown of Activity, as well as Introduced Rubric Assessment for three Soft Program Outcomes	Dr. Zubair Baig

Formatted: Heading 2, Left

Formatted: Heading 2, Left, Indent: Before: 0 cm, First line: 0 cm, Tab stops: Not at 0.64 cm

Formatted: Heading 2, Indent: Before: 0 cm, Tab stops: Not at 0.64 cm

Formatted: Heading 2, Space Before: 0 pt

Formatted: Heading 2, Indent: Before: 0 cm, Tab stops: Not at 0.64 cm

Formatted: Heading 2, Space Before: 0 pt