Fostering an Entrepreneurial Mindset in "Digital Systems" Class through a Jigsaw-Puzzle Model

Nozar Tabrizi
Department of Electrical and Computer Engineering
Kettering University
Flint, USA
ntabrizi@kettering.edu

***Abstract*—In this paper we introduce a jigsaw-puzzle model to instill an entrepreneurial mindset in students. We then use our model to craft and add innovative lab assignments to "Digital Systems", which is a core course taken by Electrical Engineering, Computer Engineering and Computer Science students worldwide. In each lab assignment, students are provided with some components or *puzzle pieces* as well as the user guide of a digital system, and possibly some reading materials to better understand the operation theory of the system. One selling point of our model is its significant flexibility, so that the *game* may be played in different ways. Here is one example: students go over the user guide and the reading materials to get a solid understating of the underlying digital system. To comprehend what they have available, students also study the puzzle pieces. Then students challenge themselves to complete the puzzle and eventually build the digital system *physically* and test it. The resulting system is supposed to function in accordance with its user guide. Our work is an attempt to fulfill the Kern Entrepreneurial Engineering Network (KEEN) Students Outcomes. The anonymous survey from the students is encouraging and shows that our work is able to excite students' curiosity, provide them with an opportunity to apply creative thinking, and instill in students a feeling of value creation among other things.**

Keywords—Digital systems; entrepreneurial mindset; FPGA; innovation in education; jigsaw-puzzle model; KEEN; VHDL.

I. INTRODUCTION

"Digital Systems" is a core course taken by Electrical Engineering (EE), Computer Engineering (CE) and Computer Science (CS) students worldwide. This class begins with the basic building blocks of digital circuits and ends with complex digital systems. Students additionally learn how to write a code to describe digital circuits using an HDL, or Hardware Description Language. They then use cutting-edge technology to compile and map their code into a pre-made integrated circuit called FPGA, or Field Programmable Gate Array, in just a couple of seconds thus physically building digital circuits.

"Digital Systems" is a must to grasp a basic understanding of the hardware of revolutionizing microprocessors, which are increasingly and inevitably entering our daily lives as we approach the era of IoT, the Internet of Things [1]. This course is also a prerequisite for upper level classes such as "Computer Architecture and Organization" and "VLSI Design." "Digital Systems" is offered every term in our ECE department and similar departments worldwide.

In this paper we introduce a jigsaw-puzzle model to fulfill the KEEN Student Outcomes, hence instilling an entrepreneurial mindset in students. We then use this novel model to add innovative lab assignments to "Digital Systems." We will explain our jigsaw-puzzle model shortly.

KEEN, The Kern Entrepreneurship Education Network, is a collaboration of faculty from over twenty U.S. institutions [2]. The goal is not only to teach engineering skills to undergrads, but also to help students develop a *mindset* that motivates them to find *innovative ways* to *create value*.

KEEN is sponsored by The Kern Family Foundation, a grant-making organization [3]. Our work in this paper was supported by a Kettering Internal KEEN Grant.

The rest of the paper is organized as follows: Our Jigsaw-Puzzle Model is introduced in Section II. We look at the previous work in Section III. Section IV introduces the products. To present a better picture of the type of the work students perform, you will see an example in Section V. Our assessment techniques are covered in Section VI. Section VII is the conclusion.

II. JIGSAW-PUZZLE MODEL

Let us start this section with three basic definitions:

Definition: A *component* is a digital circuit that has already been described in HDL and tested successfully.

Definition: A *product* is a functional digital system comprised of some properly *interconnected* components.

Definition: A *library* is a set of *individual* components.

Here is what happens in the real world: to build a product such as a microprocessor, you need the right *components* as well as the right *interconnects* to properly put the components together.

The big picture of our jigsaw-puzzle model is as follows: Students are provided with the user guide as well as a library of components or *puzzle pieces* for a product. The user guide explains what the function of the product is and how to use it. Students will first go over the user guide and then properly put

This work was supported by a Kettering Internal KEEN Grant.

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

the right puzzle-pieces, or components, together to eventually complete the puzzle and get the functional product.

Our model is significantly *flexible*: The deliverables of this work can be used in different ways with different levels of creativity. *Some* possible scenarios are briefly explained here:

Scenarios 1 and 2: Students on each team are provided with the user guide of the same product. The whole team helps each other understand the product. The instructor then provides the team members with non-identical but overlapping libraries of puzzle pieces (components) with some redundancies. None of the libraries contains all the necessary puzzle pieces. Here the game may continue differently:

Scenario 1 (Cont'd): Each team member uses his/her own library to partially design the product. He/she then communicates with the other team members to get the missing component(s) and complete the puzzle.

Scenario 2 (Cont'd): Each team member uses his/her own library as much as possible to design the product partially, and then passes it to the next team member who will continue to complete the puzzle as much as he/she can by adding zero or more new components to the design using his/her own library. Call the first student the *owner* of this design. This process will continue until the complete design reaches its owner or the current attempt fails.

Scenario 3: All the team members are provided with the *same* library of components. The library contains all the necessary puzzle pieces with some *possible* redundancy. The whole team help each other understand the underlying product, identify the right pieces, put them together properly, and complete the puzzle.

This work is also *flexible* in that the instructor may *thoughtfully* make *smart* changes to different libraries to alter the level of creativity necessary to get the job done. For example, he/she may hide a couple of puzzle pieces and let the students figure out what the best choices are to complete the puzzle.

A so-called *bidirectional* approach inherently exists in the jigsaw-puzzle model. And this is another selling point of our model that distinguishes it from other works. While students read through the user guide, they will think of a design (for the product) with no restrictions on the available components. This is the traditional top-down design direction, which is not only normal but recommended as well. This way they will construct an initial image of the final product. Students will then go over the available components, and most probably see what they have available is very different from what they think they need. Now they have to move bottom up: start with the available components and try to reach a design. This second direction of thought will naturally excite students' more curiosity and therefore will direct them towards creative thinking.

Our jigsaw-puzzle model, which facilitates different types of interactions between the team members, is able to help students develop an innovative mindset and move towards the following KEEN Student Outcomes:

- Stimulate students' curiosity,
- Direct students towards creative thinking and an innovative mentality,
- Instill a feeling of value creation in students,
- Apply system thinking to complex problems,
- Help students identify personal passions and a plan for professional development, and
- Encourage teamwork, collaboration, and connection.

Although this paper is targeted at "Digital Systems," our jigsaw-puzzle model can be utilized and implemented in several other CE/EE/CS courses as well.

III. Previous Work

Academia has recently given more attention to innovative teaching/learning techniques. In [4] Thoroughman et al develop new Entrepreneurial KEEN Modules and integrate them into their previously developed online course motivating students to investigate market and society driven problems, and then seek help from on-campus and local experts to explore solutions. Liu et al introduce and develop some mechanical engineering exercises in [5] based on the problem-based learning pedagogy to instill entrepreneurial skills in students. In [6] Faust et al develop a hands-on module where students use an FPGA-based system to create analog signals. The authors then incorporate this module in the Digital Logic and Design course. The module helps students understand the breath of their knowledge and motivates them to resolve all the issues that they may have in understanding the technical aspects involved in this process. In [7] Riofrio et al improve two courses, namely Introduction to Engineering, and Data Acquisition and Processing, using a combination of elements such as Entrepreneurially Minded Learning, Problem Based Learning, Active Collaborative Learning, and a design framework based on the "living with the lab" program. In [8] and to stimulate students' thoughts in Digital Circuit Design class, Pang proposes an active learning environment through an integration of online tools. Rayess designs a multidisciplinary project-based freshman-level course called Fundamentals of Engineering Design where students analyze a modern commercial technology and propose a plan of action using that technology [9]. In [10] Robert Pech et al design a junior-level course to expose students to four main phases of innovation and entrepreneurship, namely: "identification of a new need or want; invention of a technological solution; testing and implementing that solution; and the mock-start of an entrepreneurial venture." In [11] Bell-Huff et al produce a sophomore-level course to cover the gap that exists in design opportunities and what many students need to develop an entrepreneurial mindset between the freshman and senior year of their engineering education. The course provides students with a team-based multidisciplinary design studio experience. Mallory et al develop modules for Statics and also Probability and Statistics undergraduate courses focusing on entrepreneurially minded learning [12]. The first course focuses on a historical engineering failure case study to let students understand why the failure happened and how to possibly improve the design. The second one is based on a market research analysis to make a data driven decision on where to locate a new facility for a company.

IV. PRODUTCS

We have developed five libraries of components for five products, and successfully tested and incorporated them into our "Digital Systems" lab assignments. Here is the major challenge that we have faced in this process: the products must be sufficiently complex but still digestible to and manageable by sophomores in the middle of a 10-week academic term as specified in our grant proposal. This challenge is further highlighted if you notice that at this time of a term students have not been exposed to sequential logic yet, and therefore, students' core work has to be limited to combinational logic. Moreover, being complex is not sufficient; the application of each product must make enough sense to students, or they will not be sufficiently motivated to challenge themselves to tackle the problems.

For these lab assignments, students need a computer and an FPGA board, which are generally available in EE/CE/CS departments. The software is downloadable at no cost. This means that every university can use our deliverables; i.e., the work is *transferable*, which is another selling point of our work. Now Let us take a *quick look* at the products:

A. *15-bit Unsigned Floating-Point Adder: Model No Fadd15*

Fadd15 is a 15-bit floating-point adder. Fig. 1 shows the adder's inputs and outputs. Once the puzzle is completed, use SW14:0 to produce operand A, and then hit Key3. Use the same switches to generate operand B. You will immediately see the fraction and the exponent of the sum on displays 0 through 2 and display 3, respectively.

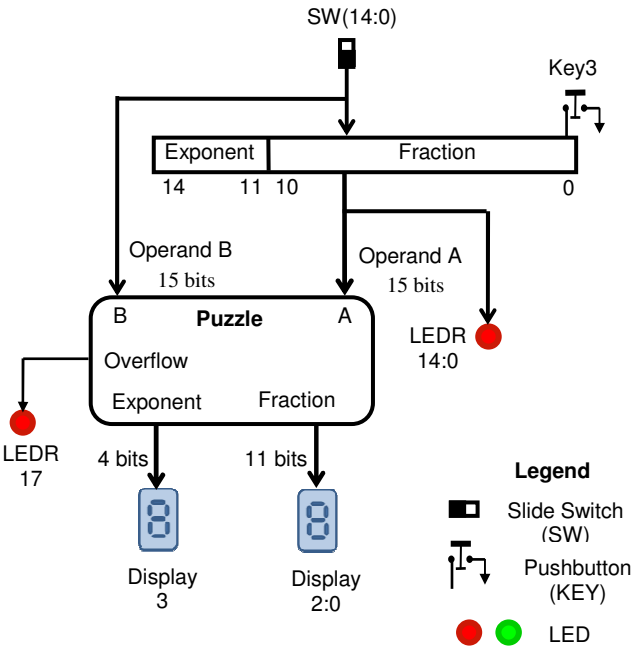

Fig. 1. Floating-point adder's inputs/ouputs

This product is a 15-bit adder and therefore the 18 slides switches available on the FPGA board are not sufficient to generate the two 15-bit operands at the same time. This is why a 15-bit register is used to store the first operand while the second operand is generated by the same 15 slide switches. This part of the deisgn is given as studetns are not expected to know sequentiolnal logic when they start working on this lab assignment.

This lab assignment comes with a 2-page handout on unsigned binary addition. This helps students understand floating-point adders covered in the 4-page user guide of this product.

The major components for this product are as follows:

- 4-bit absolute value calculator
- 12-bit adder
- 4-bit conditional incrementor: adds 1 or 0 to a 4-bit number
- 12-bit shifter
- 4-bit subtractor
- 2-input 4-bit multiplexer and 2-input 11-bit multiplexer

B. *Five-Bit 2-Mapping Scrambler: Model No Scram5x2*

Scram5x2 receives 5-bit words and produces *different* 5-bit words. The input words are generated by slide switches 4 through 0 and displayed on two 7-segment displays. The output words are display on two 7- segment displays as illustrated in in Fig. 2. The scrambler has two mappings, which can be selected using a control input, slide switch 17 as shown in this figure. The user guide provides students with the details of the two mappings

As an extra work, interested students may wish to design a descrambler to reproduce the original code.

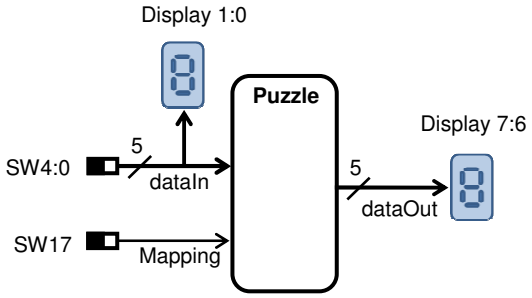

Fig. 2. Scrambller's inputs/ouputs

The major components for this product are as follows:

- 4:16 active-high decoder with an active-high enable input
- 4:16 active-high decoder with an active-low enable input
- 16:4 active-high binary encoder with an active-high enable input
- 2- and 5-input OR gates
- 2-input single-bit multiplexer

C. *Sorting Tree: Model No Srt648*

Srt648 is a 64-key 8-bit descending sorter. Once the jigsaw-puzzle is completed, manually place keys in KeyROM. See Fig. 3. To do so, you need to edit the associated HDL file, and

then compile it and map the whole system into the FPGA chip. Hit the Start button; the largest number will be displayed on displays 7:6 straight away. From now on, the next largest number will appear immediately should you hit the nextOne button.

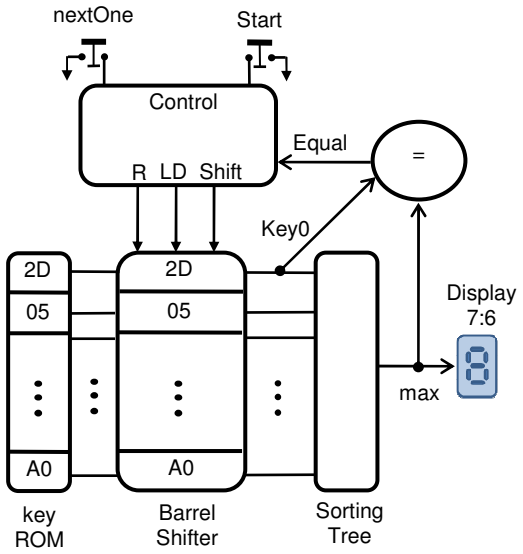

Fig. 3. Sorting tree

The sorting tree consists of the following blocks as shown in Fig. 3:

keyROM, Control, Barrel Shifter, sorting Tree and a Partial Comparator.

Since students are not exposed to sequential logic when they start working on this assignment, the first three blocks are already interconnected and given. It is further explained that when the Start button is hit, the contents of keyROM are written into the Barrel Shifter, which immediately starts rotating its contents. The Barrel Shifter is followed by the Sorting Tree, which is able to sort its 64 input words each 8 bits wide. When the largest number reaches the top of the Barrel Shifter, it stops rotating provided that the partial comparator is placed properly as shown in Fig. 3. This way the largest number will be displayed unless the nextOne button is pushed. This event will trigger the control unit to reset the top word (the current max) to zero eliminating the current max. As a result, the Barrel Shifter will restart rotation until the second largest number reaches the top word. This will end rotation hence making the new max visible on the display. The above sycle may continue until the smallest number is displayed.

The major components for this product are as follows:

- Barrel shifter
- Control
- keyROM
- 8-bit partial comparator
- 16-input sorter (made-up of 2-input sorter, which in turn consists of one full comparator and one multiplexer).

D. Service Request Controller: Model No SRC1603

SRC1603 is able to receive up to 16 requests from 16 devices with different priority levels, and determine the 3 requesting devices with the highest priorities. The numbers of the selected devices are shown on 3 displays as illustrated in Fig. 4 in which the requests are generated by 16 slide twitches. In case of fewer than three requests, the irrelevant displays will turn off.

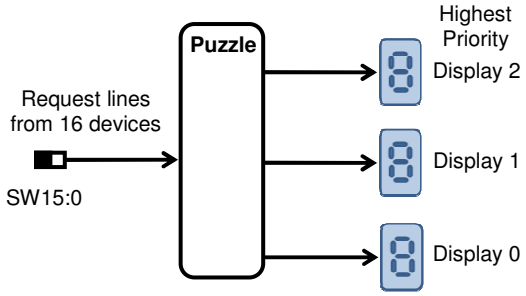

Fig. 4. Inputs and outputs of Service Request Controller

The major components for this product are as follows:

- 4:16 active-low binary decoder with an active-low enable input
- 2-input 3-bit multiplexer
- 2-input OR gate
- 8-to-3 active-high priority encoder with an active-low NO_REQUEST output
- Bank of 16x2-input AND gates.

E. 8-bit Bidirectional Barrel Shifter: Model No BrSh8

BrSh8 is an 8-bit combinational barrel shifter. It receives an 8-bit input data as well as the number of rotations (3 bits) and one extra bit to specify the direction of rotation. The rotated number will be displayed on 8 LEDs as shown in Fig. 5 where the input data, the number of rotations and the direction are generated by slides switches 7:0, 17:15 and 10, respectively.

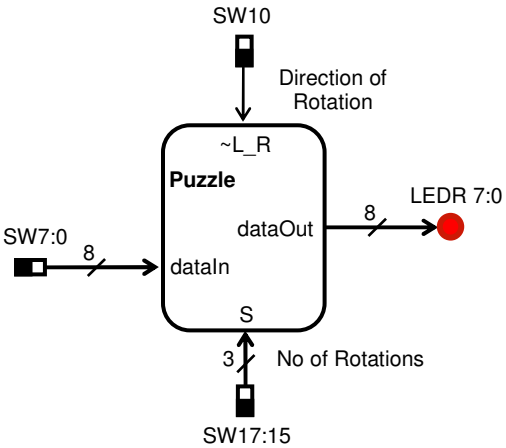

Fig. 5. Barrel shifter's inputs and ouputs

The major components for this product are as follows:

- Bank of 8x2-input OR gates
- 2-input 3-bit multiplexer
- 8-bit shifter to right
- 8-bit shifter to left
- 3-bit 2's complementor: output = 8 – input.

V. EXAMPLE

In this section, a product is presented and analyzed to better illustrate how our model excites curiosity, and therefore motivates students to think critically in order to resolve the issue and eventually instills a feeling of value creation in them. This product is not among the deliverables of our work in order not to disclose the solution of the lab assignment to our prospective students.

Active-high Priority Detector: The product has 16 input bits and 16 output bits. It takes an arbitrary 16-bit input value (with at least one logic 1), and pulls up the output bit associated with the logic 1 that has the highest priority among all the logic 1s in the input vector. This way the logic 1 with the highest priority among all the asserted inputs will be identified. (An all-zero input vector will generate an all-zero output vector.) For example, input 0010 1100 0000 1000 will generate 0010 0000 0000 0000. Let us assume that the most significant bit has the highest priority. Of course the other scenario is possible as well.

The first though for the design of this product should be similar to the circuit illustrated in Fig. 6:

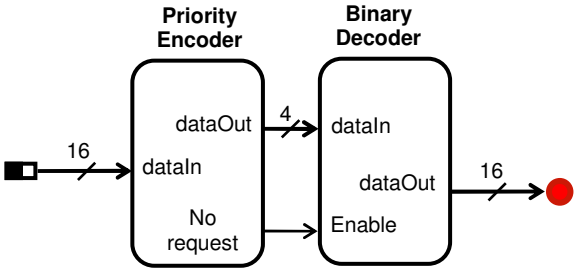

Fig. 6. Logic diagram for priority detector: first thought

In this design, a 16-bit priority encoder determines the 4-bit position (index) of the logic 1 with the highest priority among all the logic 1 input bits, and then a binary decoder converts it to a 16-bit number with exactly one 1, which is associated with the input logic 1 that has the highest priority among all the input logic 1s.

The No-request output of the priority encoder is asserted when all the input bits are at logic 0. This output is tied to the Enable input of the binary decoder as illustrated in Fig. 6, so that in case of no logic 1 at the input, all the output bits will be pulled down.

Example: Let us assume that the input number is

N = 00**1**0 1100 0000 1000

We still assume that the most significant bit has the highest priority. The priority encoder will then generate 1101, which is the position of the logic 1 with the highest priority in the input vector. (This logic 1 is highlighted in the binary pattern of N shown above.) When 1101 is applied to the binary decoder, the expected end result, 00**1**0 0000 0000 0000, will be generated correctly.

Students, however, will be surprised and their curiosity will be excited when they see that the only available component to build this product is a *half adder*! The theory of the half-adder-based priority detector is as follows:

The following operation receives N, an arbitrary number, and produces a same size output called P:

Operation I: If input N is 0, then output P will be 0 as well; otherwise, start with the least significant bit of N, copy all the bits up to and including the first logic 1, and then bit-wise complement the rest of the bits.

Example: Apply Operation I to N = 1010 0100

Starting with the least significant bit of N, the first 1 is highlighted in N = 1010 0**1**00. Therefore, Operation I will generate P = 0101 1**1**00.

It is easily seen that the bits in N and P can be classified in three groups: The first logic 1 (from the right) of N has a same-value counterpart (logic 1) in P. All the bits, if any, on the right side of this logic 1 in both vectors are zeros. For the rest of bits, each bit of P is the complement of the associated bit in N. Therefore, if the two vectors N and P are bit-wise ANDed, the resulting vector will contain exactly one logic 1, which is associated with the first logic 1 starting from the right side of N.

If we assume that the right-most bit of N has the highest priority, then **Operation I is able to detect the logic 1 (in a vector) that has the highest priory among all the logic 1s in that vector.**

It can be proved that Operation I (explained above) and Operation II (see below) generate the same result:

Operation II: Add 1 to the bit-wise complement of N. If a carry-out generated, simply ignore it.

In the half-adder-based design, we will use Operation II to detect the first logic 1 from the right of N, an arbitrary number. This is exactly what we are looking for, provided that the right-most bit of N has the highest priority. A big picture for this design is shown in Fig. 7; where the first two stages (bit-wise complement and +1) implement Operation II. The result however, is what Operation I produces as well. Therefore, N and the output of the incrementor each will have exactly one logic 1 in the *same position*, which is the first logic 1 from the right side of N. Both vectors may only have zeros on the right side of this logic 1. On the left side of the logic 1, the two vectors are bit-wise complement of each other. In the last stage of the diagram in Fig. 7, when these two vectors (N and the output of the incrementor) are bit-wise ANDed, the output pattern (Y) will be all zeros with only one logic 1, which is the first logic 1 from the right side of N. And this is exactly what we need.

An all-zero input vector is supposed to generate an all-zero same-size output vector. In the following example we show that this special case is also handled properly by the design illustrated in Fig. 7:

Example: In Fig. 7, determine output Y if N = 0000 0000.

Bit-wise complement of N = 1111 1111

Bit-wise complement of N + 1 = 1 0000 0000

Note that the carry-out generated above is ignored in the diagram of Fig. 7, and therefore the bit-wise AND block in this figure will receive two 8-bit vectors of zeros, and generate an all-zero output.

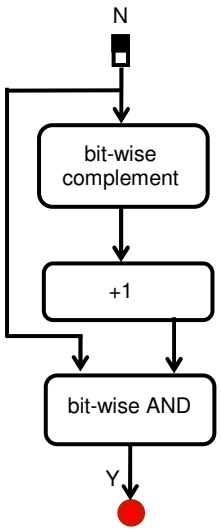

Fig. 7. Different logic diagram for priority detector

In Fig. 8 half adders are used to implement all the three blocks of Fig. 7. Note that the input number has been turned around (reversed), so that now the bit with the highest priority (still bit 15) is located on the right most position.

The AND gates in the half adders located in the top row of of Fig. 8 are not used in this design. On the other hand, one input of each XOR gate in each cell of this row is tied to logic 1, and the other input receives one of the input bits of the reversed input number; this turns each XOR gate into an inverter. Therefore, the first row of half adders is used as a bank of inverters: the output of this stage is always the bit-wise complement of the reversed input number.

The half adders in the next row are configured as an incrementor. This stage takes the bit-wise complement of reversed N and adds 1 to that. In other words, the first two stages in Fig. 8 calculate the two's complement of the reversed N and sends the result to the next stage.

The last row of half adders is used as a bank of AND gates. Each AND gate receives one bit from the reversed N as well as the associated bit from the two's complement of reversed N, and generates the bit-wise AND of these two vectors.

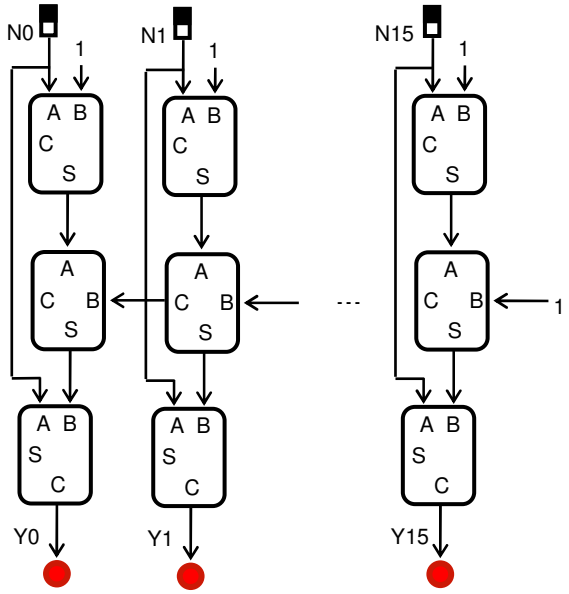

Fig. 8. Half-adder-based priority detector

For example let us consider N = 0010 1100 0000 1000. As illustrated in Fig. 8, N is reversed, inverted, incremented and then ANDED with reversed N. The result is reversed to get the final output:

N = 0010 1100 0000 1000

Reversed N = 0001 0000 0011 0100

Inverted reversed N = 1110 1111 1100 1011

Let M = Inverted reversed N + 1 = 1110 1111 1100 1100

Let K = Reversed N = 0001 0000 0011 0100

Let L = M and K = 0000 0000 0000 0100

Reversed L (Final output) = 0010 0000 0000 0000

VI. Assessment

We introduced the novel jigsaw-puzzle model to help students develop a growth mindset. We then successfully used this model to expose sophomores to *complex* digital systems in the middle of a 10-week core course. The proposed systems are based on real-life scenarios. Although the products are sufficiently complex, they are still manageable by sophomores; otherwise it would not be possible to motivate students to tackle the projects, think creatively and create values.

We have utilized different methods to evaluate our work:

Scheduled meeting with students on a one on one basis and talking about different aspects of the project and students' achievements,

Students' oral presentations: We have noticed students' significant self-confidence during these presentations,

Firsthand information from students' group meetings whenever possible: learning by teaching, an irreplaceable way of learning!

Meeting with students when they stop by and ask questions about their work: We have realized a great deal of enthusiasm and motivation,

We also provided students with 5 choices of Strongly Agree, Agree, Neutral, Disagree and Strongly Disagree for each of the following statements. Note that the last three are the *most challenging goals* of this work.

1. You consider yourself an active participant.
2. You are happy with the level of your contribution.
3. The project stimulated your curiosity.
4. The project helped you apply creative thinking.
5. Your success instilled a feeling of value creation in you …

Total number of students participated in this survey was 17. As shown in Table 1, their *anonymous* answers are very encouraging:

TABLE I. STUDENTS' SURVEY RESULTS

Statement No →	1	2	3	4	5
Strongly Agree	9	7	6	2	8
Agree	8	9	10	14	8
Neutral	0	1	1	0	0
Disagree	0	0	0	0	0
Strongly Disagree	0	0	0	0	0

1 student did not answer the last 2 questions on the back of survey.

The following are some anonymous quotes from students:

... We had to think critically in order to figure out the puzzle.

I spent many nights awake way beyond I should have to figure out the solution. So I am so very incredibly happy with the functioning result.

It made me approach the problem in a different way that I usually do. ...

I am awesome. I created a piece that was used by almost every other group. This gave me a feeling of awesomeness.

... and made me ask a lot of questions.

It was super cool to see the code work and then be able to apply it.

It brought curiosity to my mind because ....

We had to think outside of the box ...

Being able to complete a project like this really helps to show your love for the subject matter.

The project definitely made me think in new ways.

I believe lab06.CCC was good learning exercise in building a project with components much like a real world situation.

I was very curious to find out how our final product would function.

Being able to work with my group members and being able to complete a project like this made me feel accomplished.

... I went along a method of trying to figure out a solution based on methods I was used to, but found a completely different method much more rewarding in the end.

VII. CONCLUSION

We first introduced the novel Jigsaw-Puzzle model, in which students are provided with puzzle pieces with some possible redundancy, and are expected to put pieces properly together to complete the puzzle. This model may be used in different EE, CE and CS courses to instill a growth mindset in engineering students. We then used this model to develop and incorporate five innovative lab assignment in our Digital Systems class which is a core courser taken by EE, CE and CS students worldwide. The selling points of our work that differentiate it from similar works may be summarized as follows:

In order to use our work (presented in this paper) instructors do not need to make significant changes to their existing course outline. What they need to do is simply replace one or more of their current lab assignments with the ones introduced in this paper.

Our Jigsaw-puzzle model can be generalized and used in other classes such as electronics, microcomputers and programming languages.

The game presented in this paper is pretty much flexible, i.e., it may be played in a variety of ways with different levels of creativity that the instructor may choose from.

These lab assignments only need a computer and an FPGA board. The software is available online at no cost. Therefore, our work is transferable.

The products introduced in this paper are complex and based on real-world scenarios, while still digestible by sophomores in the middle of a 10-week academic term.

To handle these lab assignments students need to follow a bidirectional approach. The first direction is the regular top-down design approach in which students make an attempt to translate the word description of a problem into a digital system. The second dimension, which is bottom to top, stems from the fact that students are limited to the components that are available to them. This aspect of our work excites students' and therefore motivates them for thinking more creatively compared to regular term projects.

We also presented the quantitative and qualitative feedback provided by our students. According to this survey, almost every student agrees or strongly agrees that these assignments excited their curiosity, helped them think creatively and eventually instilled a feeling of value creation in them.

REFERENCES

[1] Internet of Things (IoT) Last Access Date 7/5/2017
http://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html

[2] The Kern Entrepreneurship Education Network
http://www.keennetwork.org/ Last Access Date 7/5/2017

[3] The Kern Family Foundation Last Access Date 7/5/2017
http://www.kffdn.org/entrepreneurial-mindset/

[4] Kurt A. Thoroughman et al, "Engineering Virtual Studio: KEEN Modules to Foster Entrepreneurial Mindset in an Integrative, First/Second Year Online Course, "*ASEE's 121st Annual Conference & Exposition, Indianapolis,* June 15-18, 2014.

[5] Liping Liu et al, 2015. “Fostering the entrepreneurial mindset in the junior and senior mechanical engineering curriculum with a multi-course problem-based learning experience,” *Proceddings of 2015 Frontiers in Education Conference* (FIE), pp 1 – 5, 21-24 Oct. 2015, El Paso, TX.

[6] Oliver Faust et al, 2015. “Getting practical with digital: An integrated electronic design project for undergraduates,” *IEEE International Conference on Teaching, Assessment, and Learning for Engineering* (TALE2015), Zhuhai, China, 10-12 Dec. 2015

[7] Jose Antonio Riofrio et al, "Innovation to Entrepreneurship in the First Year Engineering Experience," *ASEE's 122nd Annual Conference & Exposition, Seattle,* June 14-17, 2015.

[8] J. Pang 2015. “Active Learning in the Introduction to Digital Logic Design Laboratory Course,” *Proceddings of 2015 American Society for Engineering Education*, (Zone III). Last Access Date 7/5/2017

https://www.asee.org/documents/zones/zone3/2015/Active-Learning-in-the-Introduction-to-Digital-Logic-Design-Laboratory-Course.pdf

[9] Nassif E. Rayess, "Instilling an Entrepreneurial Engineering Mindset through a Freshman Design Course," *ASEE's 123rd Annual Conference & Exposition, New Orleans,* June 26-26, 2016.

[10] Robert Pech et al, 2016. “Innovation, design and entrepreneurship for engineering students: Development and integration of innovation and entrepreneurship curriculum in an engineering degree,” IEEE Global Engineering Education Conference (EDUCON2016), Abu Dhabi, UAE, 10-13 April 2016.

[11] Cristi L Bell-Huff, Donald D. Carpenter P.E. and Andrew L. Gerhart, "Fostering an Entrepreneurial Mindset through a Sophomore-Level, Multi-Disciplinary, Engineering Design Studio Experience," *ASEE's 123rd Annual Conference & Exposition,* New Orleans, pp. June 26-26, 2016.

[12] Jennifer A. Mallory et al, "Inclusion of Entrepreneurially Minded Learning (EML) Modules in 2nd-Year Core Engineering Courses," *ASEE's 123rd Annual Conference & Exposition,* New Orleans, June 26-26, 2016.