ECE 197SA: ECE Systems Appreciation Calculator Lab

Designed by Arman Pouraghily and Tilman Wolf

Introduction

Embedded computing is unique because it is a hardware/software (HW/SW) co-design problem, where hardware and software must be designed together to make sure that the implementation not only functions properly but also meets power, area, and performance goals. In this lab assignment, which is following up on the experiences gained in ENGIN 112, you will design and implement a simple 4-operation calculator using HW/SW co-design approach.

The hardware part is divided into two parts. The first one is a minimal NIOS-II-based system implemented on the DE2i-150 board with the necessary interfaces to communicate with external components. The other, external hardware part is a logic circuit implementing an adder, which is what you implement in this lab assignment.

In computer arithmetic, there are various possible implementations for additions (and other arithmetic functions). For the sake of simplicity, we use bit serial adder. In our implementation, we use just a half adder and iteratively shift the inputs in and the results out bit by bit. This implementation is suitable for the cases in which power and area are more important than performance and we can use just a 1-bit adder instead of, for example, a 32-bit adder.

In this assignment, the calculator component is already implemented on the embedded system. The only thing missing is the half-adder, which you will construct on a bread board using CD4011BE and CD74HCT86E which are quad NAND gate and XOR gate chips. After implementing the adder on the board and connecting your module to the embedded system, you should be able to use the calculator. Since all the four operations (add, sub, multiply, and division) are done by iterative additions, the correctness of operations is completely depended on correctness of your adder circuit.

Lab Equipment

To complete the lab assignment, you will need one "lab kit," which contains a DE2i-150 board, an AC adaptor, a breadboard, 2 of each CD4011BE and CD74HCT86E ICs, a flat cable, and wires. The lab kits are available for sign-out outside Prof. Wolf's office (Knowles 211C).

Breadboard

As you recall from ENGIN 112, a breadboard can be used for prototyping electronic circuits.

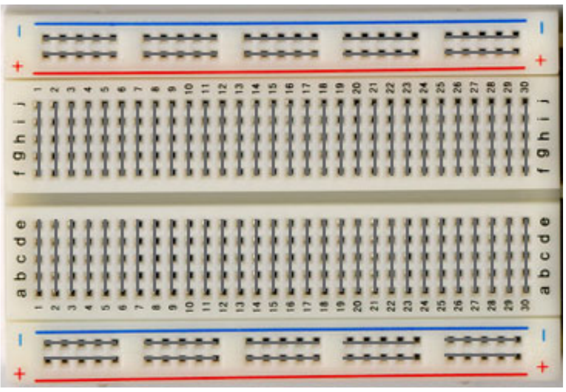

Figure 1- Layout of Breadboard

The layout of a typical breadboard is made up of two different types of connected holes, called strips. First type is the *terminal strip*. The holes in the terminal strips are connected together vertically and are numbered as shown in Figure 1. Each column is broken in the middle where a horizontal gap could be seen. So the holes of each column are connected together from *a* to *e* and *f* to *j*. Terminal strip is where an IC should be placed. As you know, the DIP ICs, which we use in this experiment, mostly have their pins arranged in two parallel lines. So the IC should be placed on the breadboard in such a way that the gap separates these two lines and prevent the short circuit between the pins as shown in Figure 2.

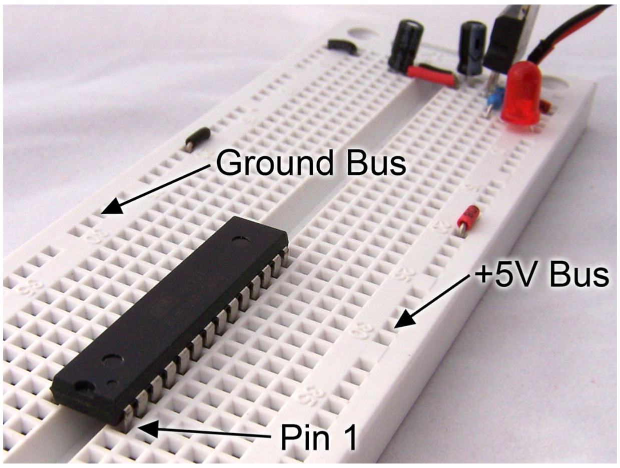

Figure 2- Placing an IC on Breadboard

The other connected part is called *bus strip*. As it can be seen in Figure 3, in these strips which are specified with red and blue lines, all the holes are connected together horizontally; so these strips are useful for connecting Vcc or Gnd to them and then just split a branch from the strip to wherever Vcc or Gnd is needed.

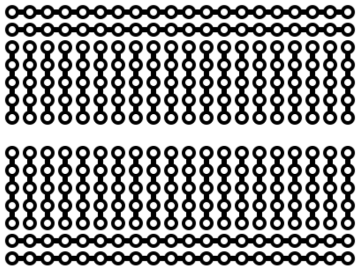

Figure 3- Underlying connection grid

DE2i-150 Board

DE2i-150 is an embedded system with an Intel Atom processor and an Altera FPGA. This board can be used for a variety of projects that involve prototyping and development of HW/SW embedded systems. (The calculator is implemented on the Cyclone IV FPGA.) The layout of this board is shown in Figure 4.
As it can be seen, this board has a 16x2 LCD module, which we use to show the

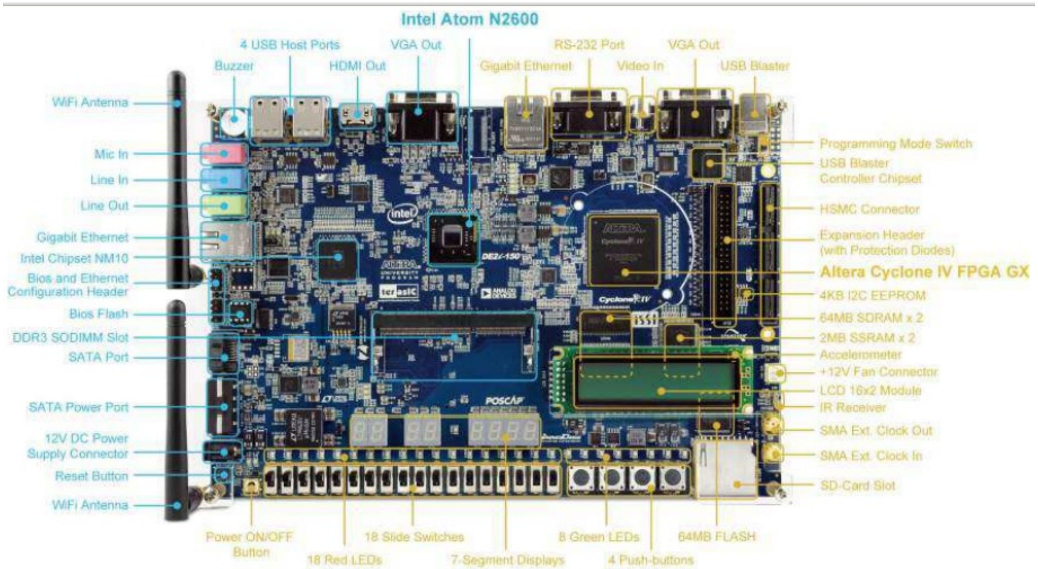

Figure 4- DE2i-150 board layout

input calculation, output results and directions. In the lower right corner, just beside the SD card slot, you can find 4 different push buttons. The rightmost one is used as *reset* in the design which means that by pressing this button, system will restart. Just above the push buttons you could find eight green LEDs used to show the progress of system in the self-checking procedure discussed below. On the left side of green LEDs, a strip of red LEDs starts from which the first one (the rightmost one) is used to indicate error in calculation. In order to connect the system to the external circuit, we use the expansion header, which is located on the right side of the board. To connect the header to breadboard, we use the provided flat cable. To input data (e.g., operands), we used the IR based remote controller provided with the board.

The layout of remote controller is shown in Figure 5. In this experiment, we use the buttons as follows:

- 0-9 Digits: You can enter the desired numbers by pressing these buttons.
- Channel Up/Down: In order to input an operation between two operands, you can use these two buttons.
- Return: The functionality of this button is equivalent to *enter* in a keyboard, i.e., pressing this button will make the system to start the evaluation process.
- Adjust: From the two adjust arrows, only one of them, the left one, is functional in our system. The functionality of this arrow is similar to backspace in keyboard. By pressing this arrow, you can simply undo the last character entered.
- Play: When system restarts and after pressing *enter* to start its operation, you can press play instead of entering the calculation. It will make the system to start its self-test procedure, which is described below.

Figure 5- IR based remote controller

CD74HCT86E IC

As mentioned before, in order to implement the HA functionality on the breadboard, we use two different ICs. One of them is the CD74HCT86E IC. It contains four independent EXCLUSIVE OR gates in one package. They provide the system designer with a means for implementation of the EXCLUSIVE OR function. Logic gates utilize silicon gate CMOS technology to achieve operating speeds similar to LSTTL gates with the low power consumption of standard CMOS integrated circuits. The internal connection of this IC is shown in Figure 6. Please note that in this figure, the circle is on right hand side and the arc is in left hand side. When you are using this schematic, make sure you've placed the IC in the same way.

CD4011BE IC

The other IC needed to implement the HA is *CD4011BE* provides the system designer with direct implementation of the NAND function and supplement the existing family of CMOS gates. The functional diagram of this IC is shown in Figure 7. Please pay attention to the position of circle and the arc when placing the IC on breadboard.

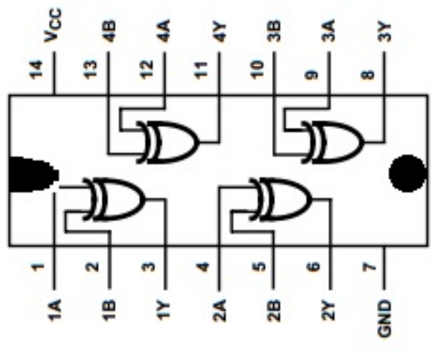

Figure 6- Functional diagram of CD74HCT86E

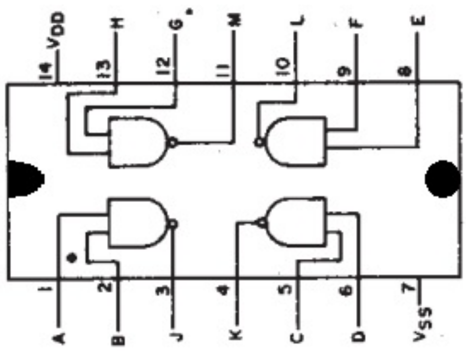

Figure 7- Functional diagram of CD4011BE

Flat Cable

As mentioned before, we use a 2x20 flat cable to connect the expansion header of the DE2i-150 to the bread board. It is important to understand the pin organization of this cable. As shown in Figure 8, the cable has 3 identical headers. The connections between these headers match in one to one fashion between each connection and its counterparts in the other two headers. Each connection in the headers has a number so it can be distinguished from the others. When viewing the header vertically from the front with the small bump at the right hand side, the upper right connection is numbered as 1 and the left one as 2. For the second row, it is 3 for the right one and 4 for the left one, etc.

Figure 8- 2x20 Flat Cable Headers

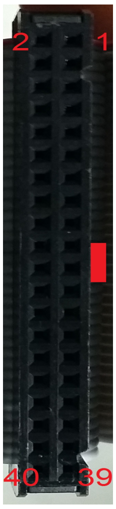

Figure 9- Pin numbering scheme

Lab Instruction

In this section, we will go through the steps that you will need to take to do this assignment.

Building Half Adder

The first step of this assignment is to implement the half adder on the breadboard. In Figure 10, you can find the schematic of a half adder circuit. As you see, it can be realized by just an XOR and an AND gate. But we only have NAND gates available, so you should think about a way to build an AND functionality with NAND gate. (Hint: use an extra NAND gate on the output of the first one)

In order to power the ICs, you need Vcc and Gnd voltages. The DE2i-150 provides these supply voltages, so just reserve a pin for the supply voltages and postpone the connection.

Notice: Always make sure you do not accidentally short VCC and GND supplies! Doing so could damage the board.

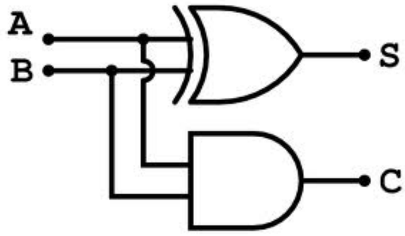

Figure 10- Schematic of a Half Adder circuit

In Figure 11 you can find the final implementation of the HA circuit. As you can see I used a + rail from top of board for Vcc (red wire) and a – rail in middle for Gnd (black wire). And I left them unconnected.

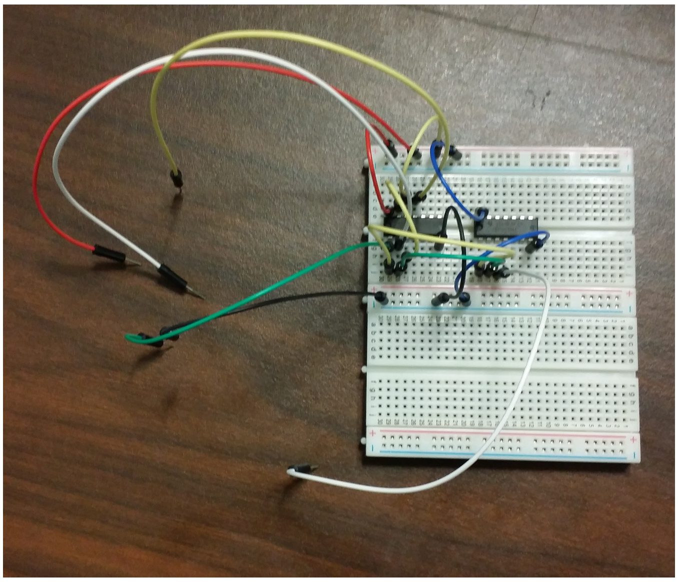

Figure 11- Final implementation of half adder

Connecting the breadboard and DE2i-150 together

In this step, you will connect the implemented half adder to the DE2i-150 board using the flat cable. First, connect the flat cable to the expansion header of DE2i-150. Then, connect your breadboard wires into the cable header's pins. In Table 1, you can find the pin number of each signal in flat cable headers.

Table 1- Header pin number and the associated signals

Header pin number	Associated signal
29	Vcc
30	Gnd
32	C
35	B
36	S
39	A

In Figure 12, you can see the complete connection between the DE2i-150 and the implemented circuit. The next step is to evaluate the whole design.

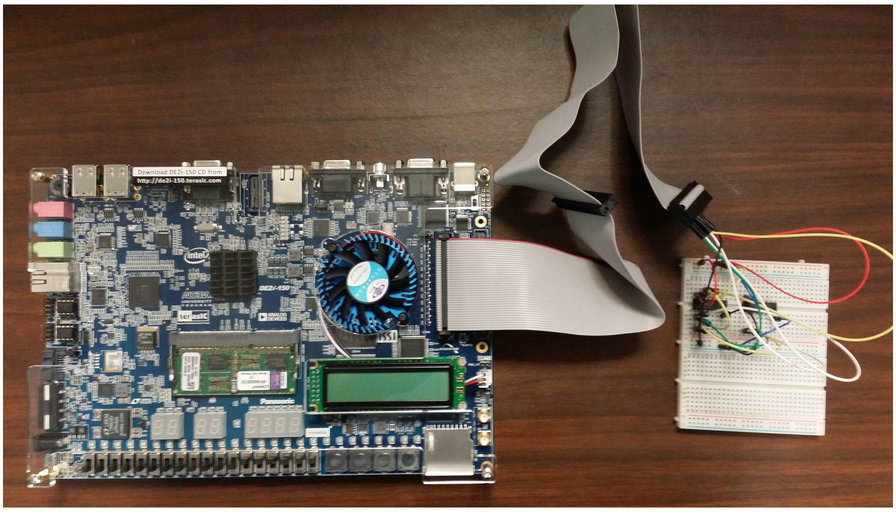

Figure 12- Complete setup

Evaluating the system

Once implemented, you need to check if your system works. In order to do so, you need to first plug the power cable into the power supply connector which is located in left down corner of the board in Figure 4 and then simple push the power button which is located below the power supply connector (not directly below; the one directly below is the reset button). After turning the board on, please wait for couple of seconds until the "ECE 197SA …" message appears on the LCD. Then, you can start evaluation.

Entering Calculation

After powering the system on, you can press enter and it will be ready to accept your calculation. To enter the operands you can use digits on remote controller (ensure that the controller is facing to the receiver sensor on the right edge of the board). For entering an operation you can select your desired operation by channel switches and when you are done, you can start calculation by pressing *return* button. In case of mistyping, you can simply delete the entered digit or operation by pressing *adjust* left arrow.

Checking the result

Upon completing each operation, the system will show the result calculated by your half adder and also the result of a comparison of the calculated value with that of its internal calculation. If the result is not correct, a red light will start to flash. If the result is correct, the system displays "Correct" on the LCD.

Self-Checking Procedure

The system is able to do an automated test of your half adder ("self-test"). This is done by calculating 8000 randomly generated calculations and comparing the results with the results obtained from the internal arithmetic unit. To start the self-test procedure, you should first reset the system by pressing the *reset* push button located below the LCD. After resetting the system and pressing return to start the system, you need just to press play button on the remote controller to take the system to the self-test procedure. During the procedure, the green light will show the progress. You should be patient until the procedure ends successfully and you get 8 green lights on. Figure 13 shows the state of the board during the self-test procedure. Two green lights show that 25% of the process is completed and 75% is ahead.

Figure 13- Test progress

In Figure 14, the system faced a problem during the self-test procedure and the red light starts flashing.

Figure 14- Self-test failed

Finally, in you can see that the system passed the whole procedure successfully and we have 8 green lights on and "Passed" printed on the LCD.

Figure 15- Self-test passed

Submitting Results on Moodle

In order to receive credit for the lab assignment, you need to submit on Moodle the following:

- A photo of you and your lab implementation that shows that the self-test has passed (i.e., green lights). Ideally, the photo should show the embedded system, the external circuit, and your face. If that is not possible, submit multiple pictures that show that you completed the lab.
- Optional: The above picture (i.e., you and the *working* lab setup) in an interesting environment (e.g., scenic background). Best pictures will be presented during the last lecture. Please do not do anything dangerous!

Returning the Kit

Please disassemble your implementation, put all parts back into the kit, and return the kit to Prof. Wolf's office (or pass it on to another student in the class who needs to do the assignment). If any part is broken, please contact us (see below).

Help!

If you need help, please contact:

- Arman Pouraghily (apouraghily@engin.umass.edu) or
- Prof. Wolf (wolf@umass.edu).