

User manual

CSCI 6461



2018-3-4

Index

**1.Summary2**

**2.User Interface3**

2.1 Front panel3

2.2 Magic panel5

2.3 Virtual Machine6

**3.** **Precautions** 7

3.1 LDX & STX7

1. Summary

This project works as a simulator of computer architecture.

We divide the CPU’s component into two categories.

For one part, we call it Chip. Chip is the general term for all components that have operational capabilities. A chip can take one or more inputs and react in one or more outputs.

For the other one, we call it Cable. A cable is used to transmit data between chips.

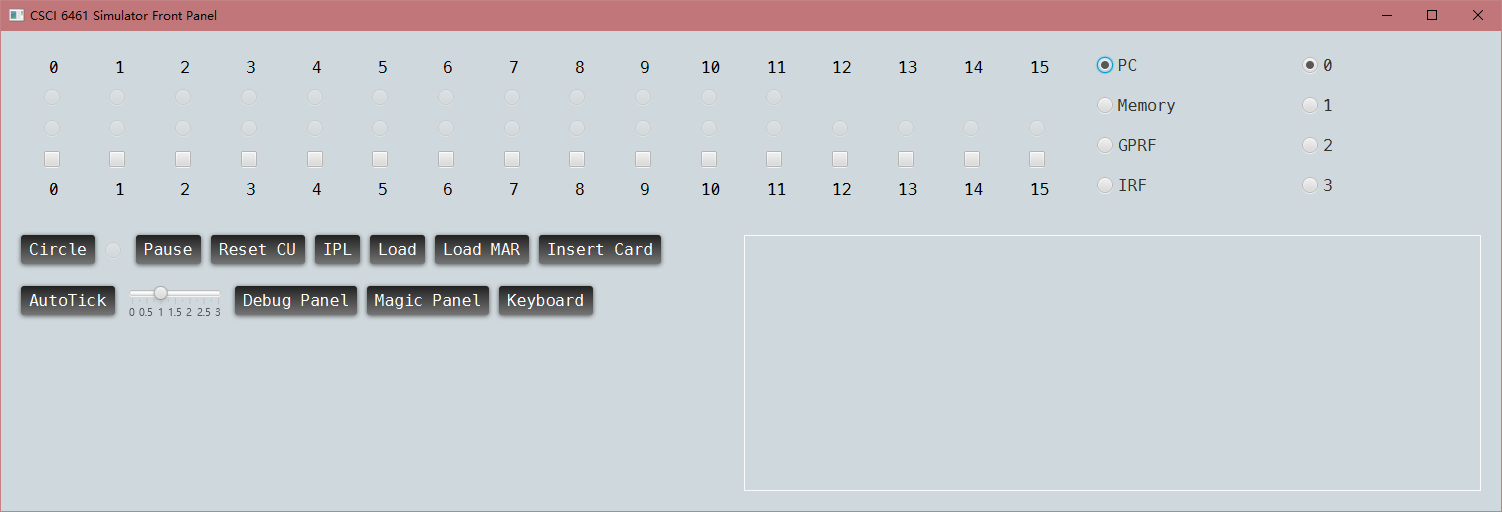
We use these two parts to simulate the processor from the hardware level.

For the clock cycle, we designed two parts, ”tick” and ”evaluate.” “Tick” is the clock rising edge, registers load the value when “tick” is coming. “Evaluate” is the time for “chip” operating. Chips change their outputs when evaluate.

In the second stage, we

1. User Interface

2.1 Front Panel



The front panel is used to put value into “chip” manually. It is what you see if you are using the computer system.

In this section, you can open keyboard in this panel. You can see the console printer to the right of the panel.

Front panel itself interacts the machine with some chips. We use some cables to connect it to our simulated machine.

The front panel has a total of 28 lights, divided into 2 lines, 12 and 16 lights a group respectively.

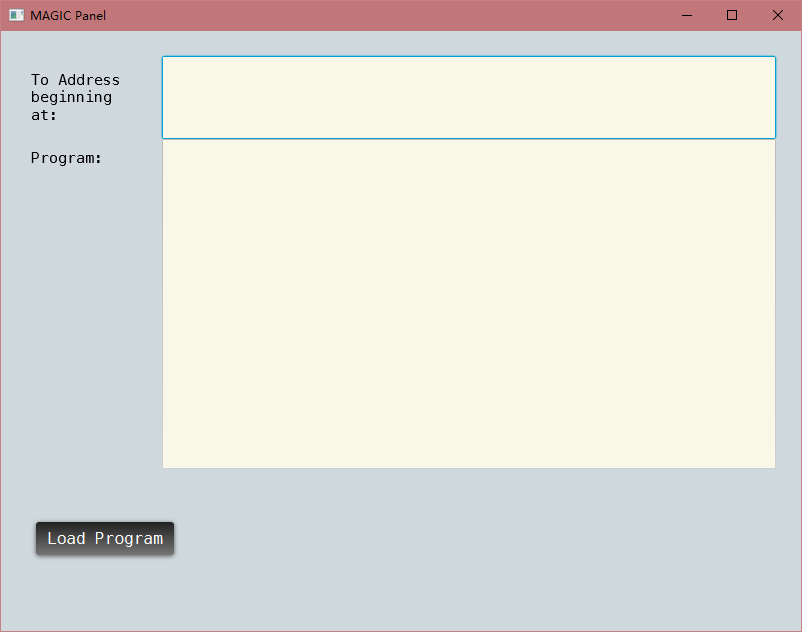
1. Above line (12 bits): the line with 12 lights is used to indicate the memory address register’s value.
2. Lower line (16 bits): the line with 16 lights shows the value of user’s choice.

The Front panel has a total of 16 switches. All the switches are used to enter a value.

The front panel has several functions.

1. Circle: Clicking the Circle button makes the simulator run a complete period of a clock, which includes a “tick” and an “evaluate” process.
2. Pause: The Pause button temporarily stops the simulator. When clicking the Pause button, the control unit pauses, and in the meantime, the pause light (bulb to the left of Pause button) goes on, and the Control Unit’s Current Status in the debug panel turns to PAUSED. Tick and Circle have no functions at Pause status. Remember, loading a value by the front panel is possible only if the simulator is paused.
3. Load: we can load the value to any of the registers or a certain point in the memory by the front panel. First set the value with the switches, then choose a destination with the switches to the right. Then click the load button to load the value to the destination. Again, you can only do this when the machine is paused.
4. Load MAR: to load a value into the memory, you should set the MAR first. Clicking Load MAR button sets the MAR to the value decided by the switches.
5. Reset CU: Clicking Reset CU Status button resets the CU Status to Initialized. This function changes no value in the simulator.
6. IPL: Clicking the IPL button, which stands for Initialize Program Load, loads a program that loads a program in the card and sets PC to its entry point, by magic!
7. Insert Card: This button is used to insert a card into the card reader. By this point, it’s used to load program from a binary file, which simulates the card.
8. Auto tick, and the slider: Used to auto tick the program. In this testing, we recommend using the fastest (left most) speed, because loading includes a lot of instructions and will take a lot of time even with the fastest speed.
9. Keyboard: toggles the keyboard panel.

2.2 Magic Panel



Currently the magic panel is still used to load the document into the simulator, same as previous stage.

The magic panel has 2 textboxes:

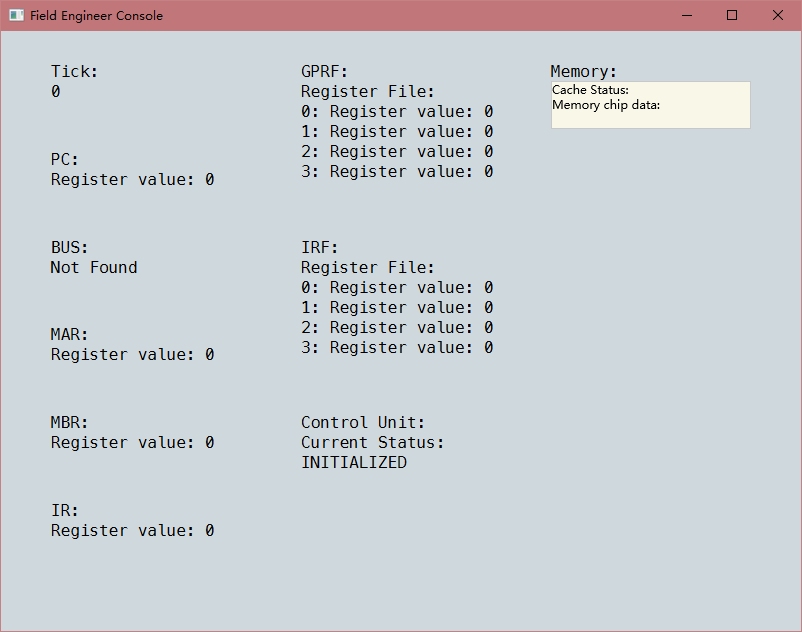
1) The text box - To Address beginning at, is used to set the beginning memory address.

2) The text box – Program, is used to put instructions.

Control panel has 1 button:

1. Load Program: Clicking the Load Program button loads the compiled program into the simulator’s memory by magic. So you do not have to flip the switches on the front panel for hours.

2.3 Debug panel (A.K.A. the Field Engineer Panel)



We use the panel to show the internal state of the simulator.

You can find the cache in the memory pane.

The bus is removed from this design, so there is nothing to show there.

1. Precautions

3.1 LDX & STX

There are some ambiguous in the handling of LDX and STX instructions. When running LDX or STX, the IX part of the instruction indicates both destination and a part of EA. Our team chooses to handle it this way. The other possible explanation of these instructions is ignore IX when calculating EA.

1. How to use this simulator

4.1 Initialize

You can initialize the simulator by clicking the IPL button on the front panel.

This will load a load program into memory address 1025, who will then load a program to 0 from the card reader. So you need to insert a card to continue.

4.2 Insert card

Click insert card button on the front panel. This will open a file select box for you to select a bin file. Please use the bin file included – It has a starting point followed by the compiled program – in machine code.

Inserting a card itself won’t do anything. You must use the program card together with the load program in the IPL to load the program.

4.3 Run the machine

At this point you can run the machine. We recommend use the fastest (left most) speed. Slide the slider to far left, then hit the auto tick button. This will start the simulator. If you are following the instructions so far, this will load the program in the program card.

1. How to use the program

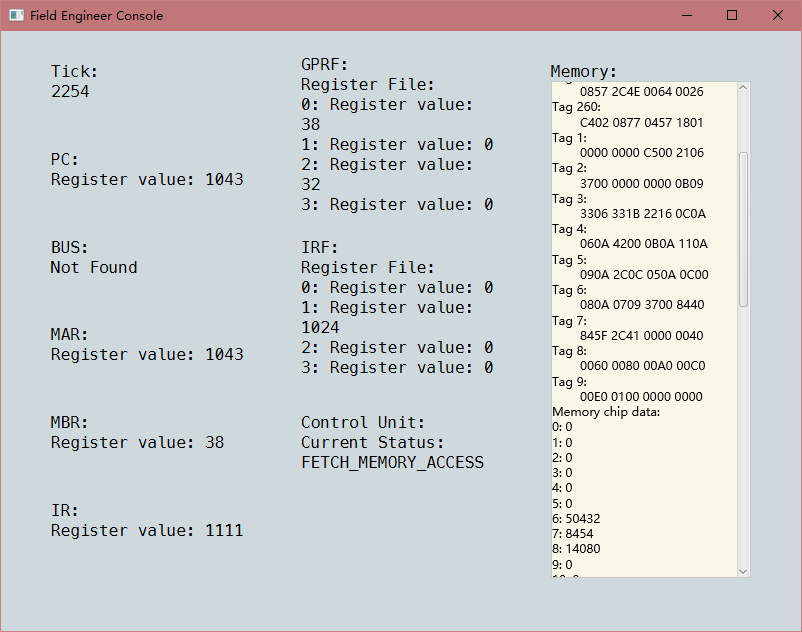
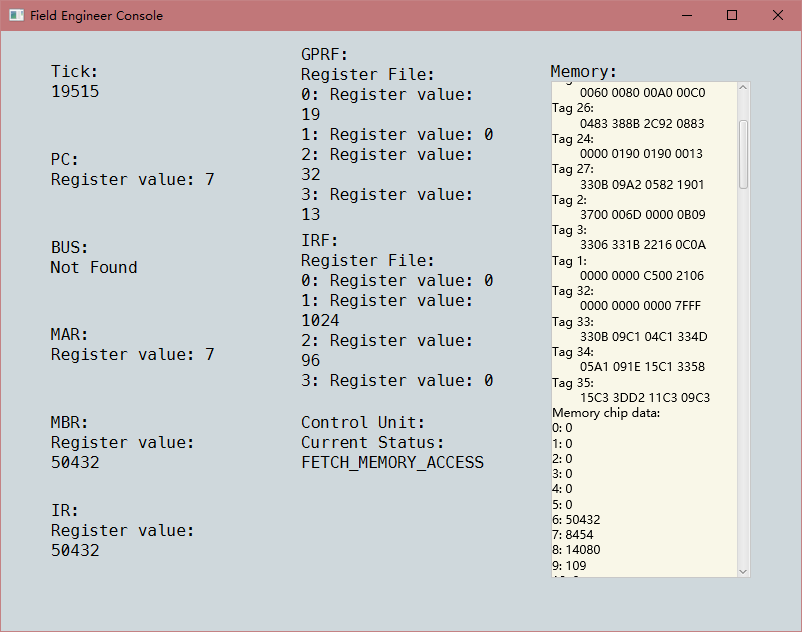
5.1 Program 1 description

This program is a program who reads 20 numbers from 0 to 65535 and prints them out, then takes another number and finds the closest one among them, and then output them both.

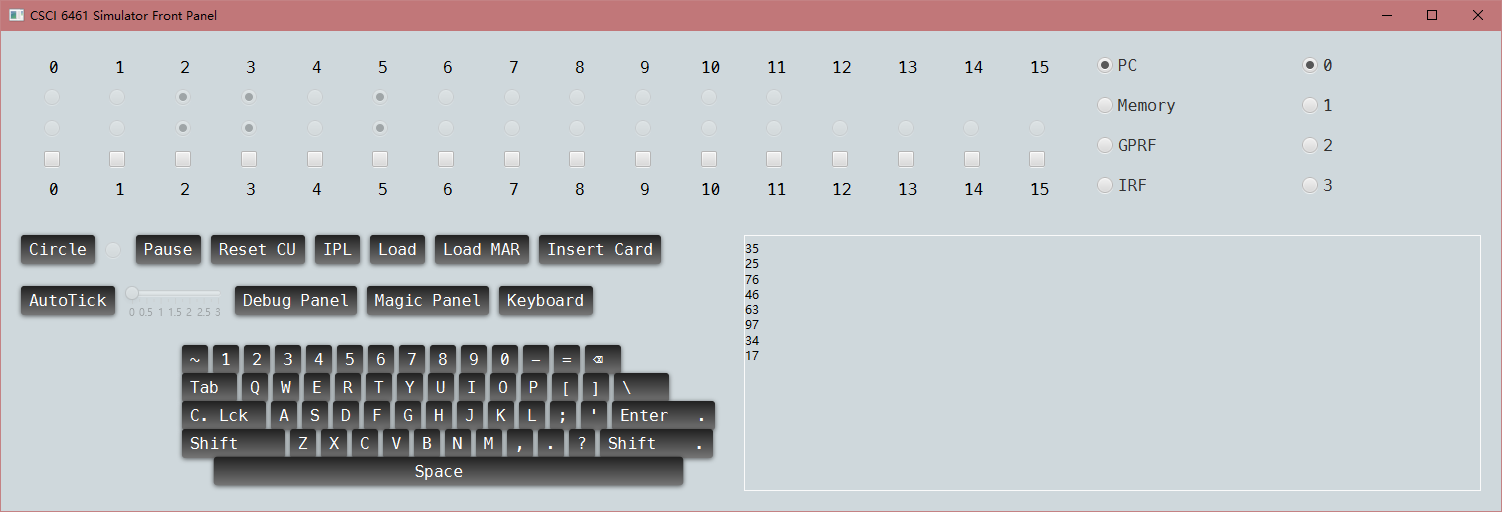
5.2 I/O

You can input number by the keyboard on the front panel. Anything pressed other than number keys will be taken as an end of an inputting number. Input 20 numbers first, then the program will output them, one line by one line. Then program will take another input number, which indicates the number, then the program will search the 20 numbers for a closest match then output them both, in two separate lines. Then the machine will halt.

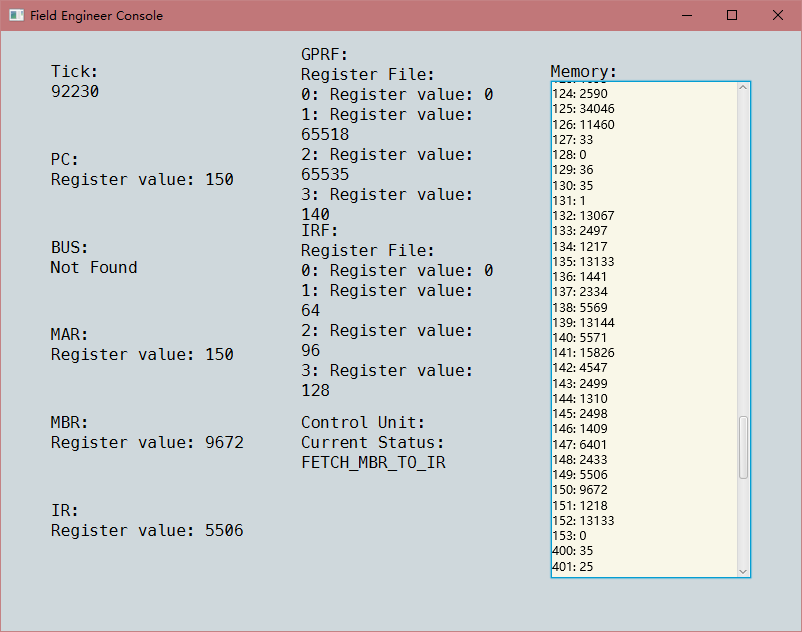
5.3 screenshots:

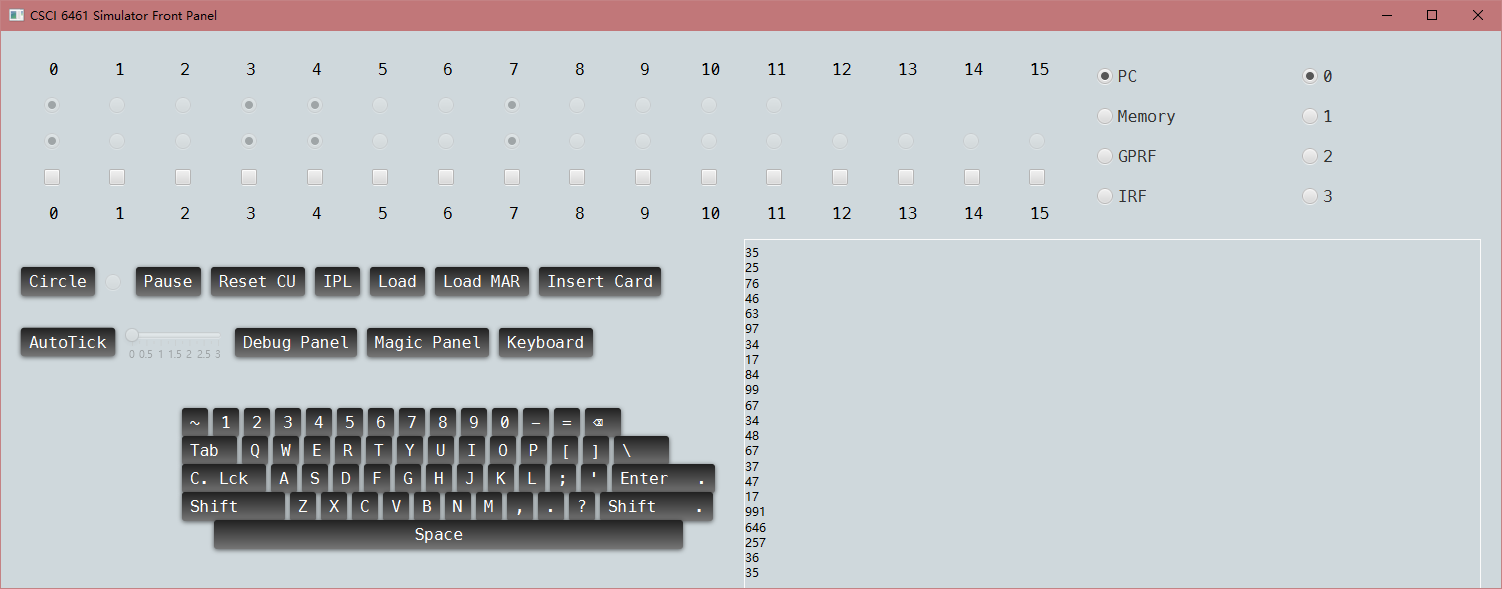
Program being load Waiting input



20 numbers being output



Finding a closest match



Answer (35) found, program halt.

5.4 Program 2 description

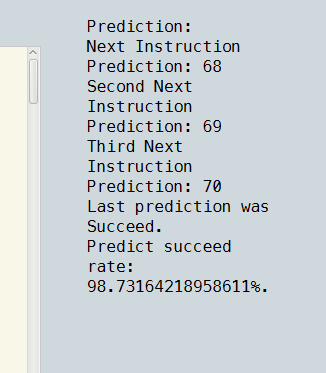
This program is a program who reads a paragraph of 6 sentences from a “card” and prints them out, then takes a word and finds where the word occurs, then output the word, the sentence number, and word number in that sentence. For further specification, please refer to the text in project directory.

5.5 I/O

You can input the word by the keyboard on the front panel. You may confirm your word by either space and enter. Load the paragraph first by inserting the card containing the paragraph, then the program will print the paragraph, one sentence in a line. Then program will take ask you for the input word. Then the program will search for the occurrence in the text, output any occurrence in a line. Then the machine will halt.

5.6 Predictor

We make predictions based on current machine status for each instruction, so the next instruction can be prefetched. This information can be found in the field engineer panel, aka debug panel. There’s also a succeed rate so you can see how awesome this prediction strategy is.

 A screenshot of a cell phone

Description generated with very high confidence

5.7 Demonstration

Please refer to the video attached. The video shows how the predictor works.