What's new

New features in Part 2

Group 9 - October 8, 2014

Part 2 has 5 distinctive features compared with Project 1, the new features will be listed as following:

1. Instructions

All the required instructions in the *Project Description* can be automatically compiled into binary code.

The instructions implemented include:

a) Load/Store

b) Transfer

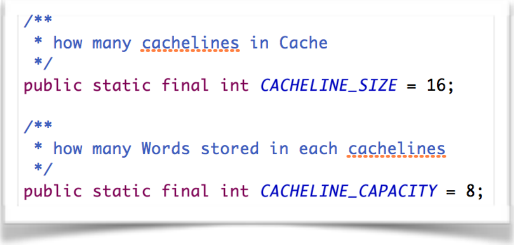
c) Arithmetic and Logical

d) I/O

2. Cache

a) Cache Design

The Cache is designed with 16 lines (same meaning as Block), each line is consisted of 8 words. Besides, the amount of the lines and the words that the line contains are not fixed, they are flexible and can be set up in the source code(Setting).

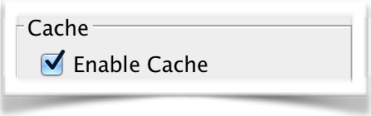


```
/**
 * how many cachelines in Cache
 */
public static final int CACHELINE_SIZE = 16;

/**
 * how many Words stored in each cachelines
 */
public static final int CACHELINE_CAPACITY = 8;
```

b) Feature

The Cache has a selective *Enable Cache* button, when it is enabled, the CPU will run with Cache, otherwise, the CPU only accesses Memory. When the mouse stays on a word, the value of the word will be showed on the Cache Section.

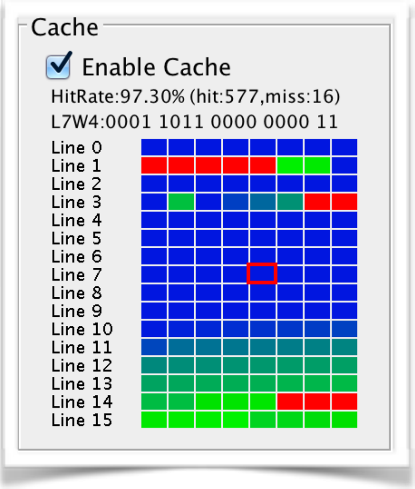


Cache

☑ Enable Cache

c) Visualization

When running the program, we set different colors to show the Cache *HitRate*, the more recently the WORD in Cache hit, the green its visual block would be. Red means these words are loaded into cache but never been visited. The color blue represents that the word recently has not been visited.



Cache

☑ Enable Cache

HitRate:97.30% (hit:577,miss:16)

L7W4:0001 1011 0000 0000 11

Line 0
Line 1
Line 2
Line 3
Line 4
Line 5
Line 6
Line 7
Line 8
Line 9
Line 10
Line 11
Line 12
Line 13
Line 14
Line 15

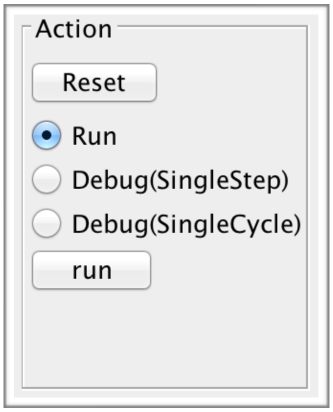
3. Running Mode

There are 3 different types of run model: After load the program from the Memory,

a) Run: The instructions will be executed from beginning to the end.

b) SingleStep: it will run only one instruction once.

c) SingleCycle: One instruction contains several cycles, after clicking the SingleCycle button, it will only run one cycle once.



Action

Reset

● Run

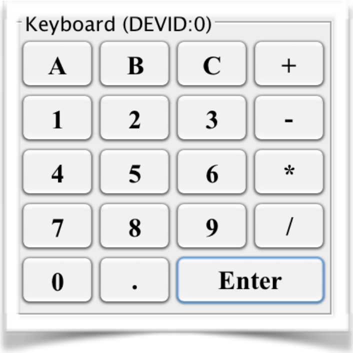
○ Debug(SingleStep)

○ Debug(SingleCycle)

run

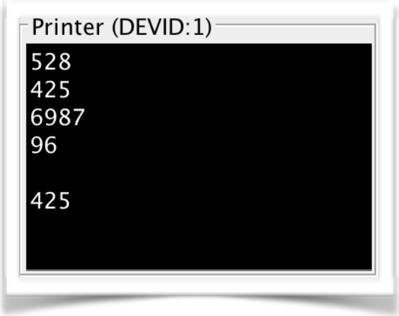
4. Keyboard

The Keyboard is an Input device. We add Keyboard function into our CPU Simulator, so when we run the program 1 which is written with assembly language, we can test the required number.



5. Printer

This is the Output device we add, when the *out* instruction is executed, the content will be showed on this section.



```
528
425
6987
96

425
```