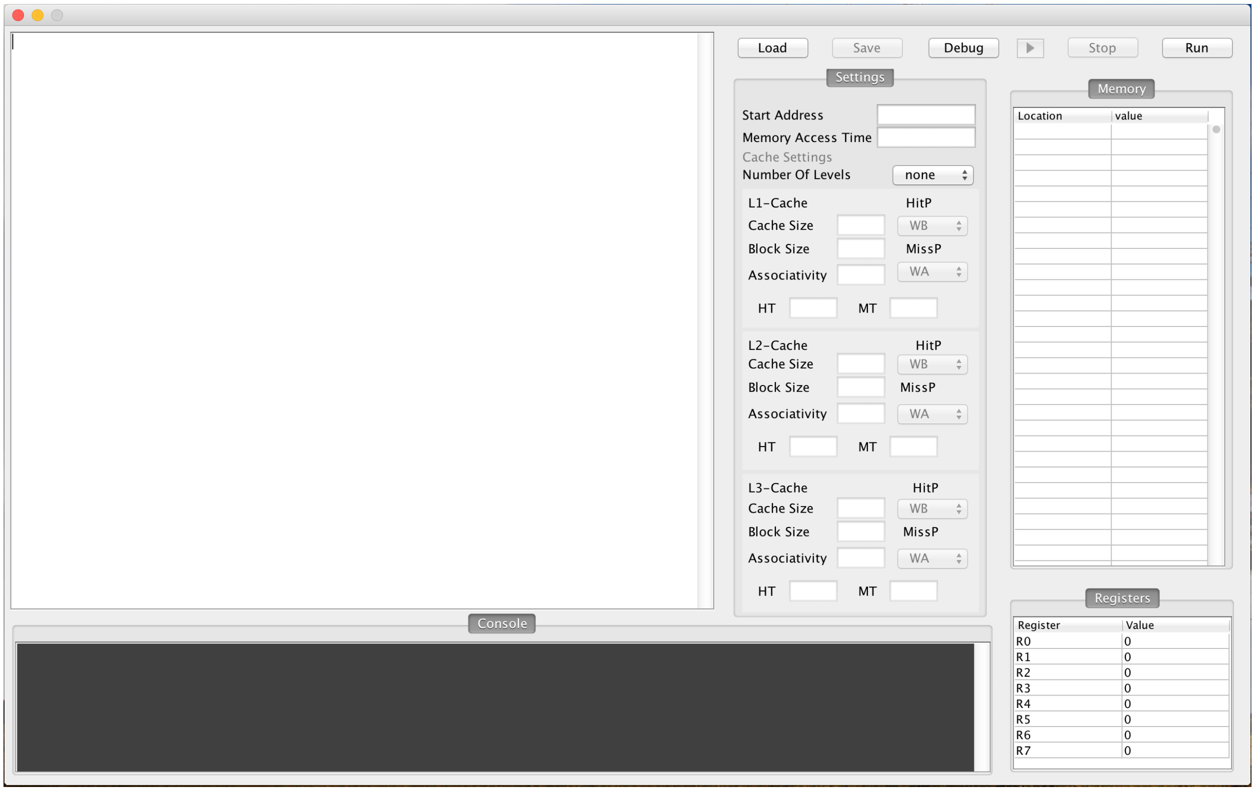
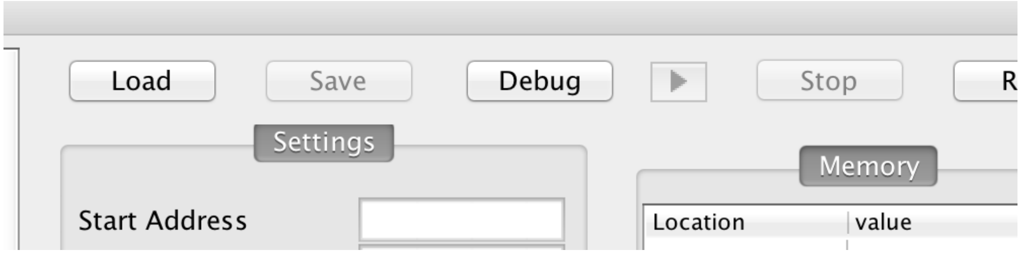


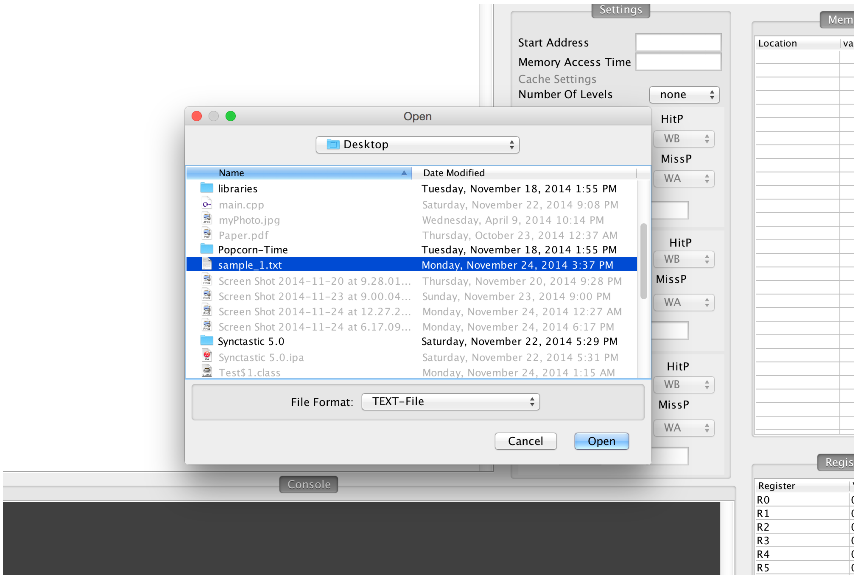
Register	Value
R0	0
R1	0
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0

After Opening the program we will be promoted with this. We have two ways to input a code, either by loading a text file or writing the code then saving it.



1- We are going to load < sample_1.txt > by selecting the load button as shown.

2- You will be promoted with a file browser. Choose the location of the file and select Open.

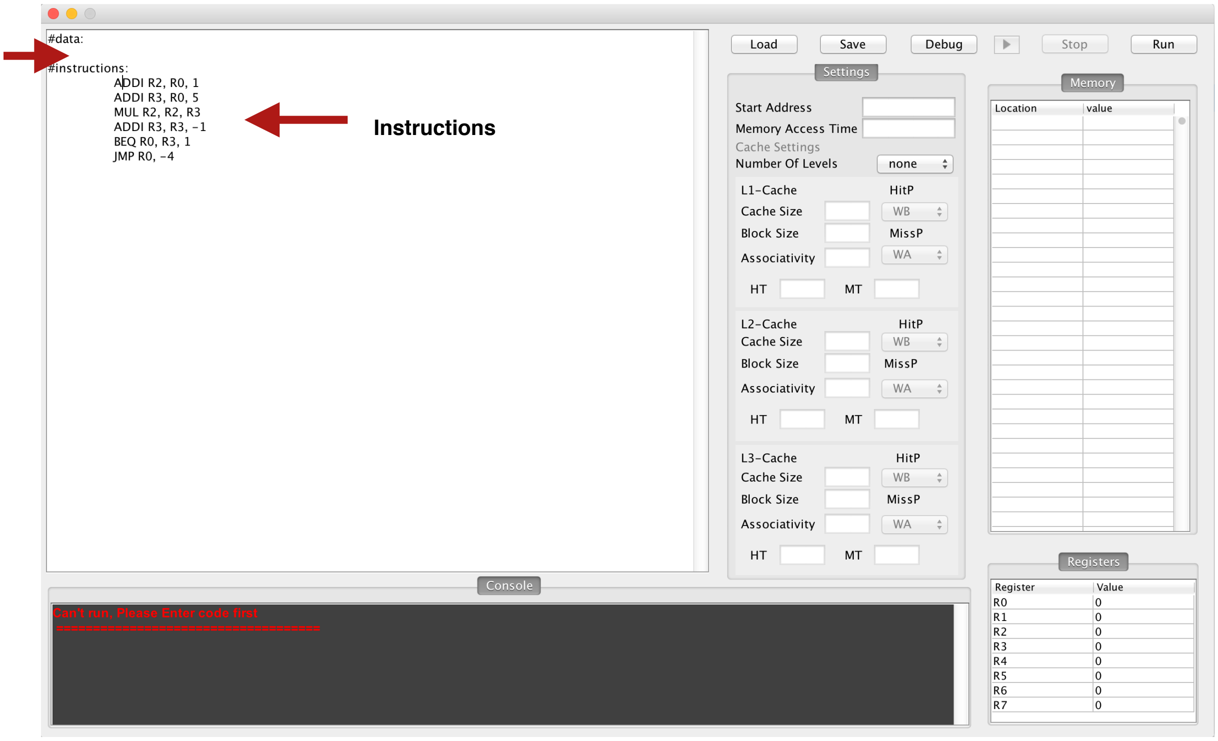


Name	Date Modified
libraries	Tuesday, November 18, 2014 1:55 PM
main.cpp	Saturday, November 22, 2014 9:08 PM
myPhoto.jpg	Wednesday, April 9, 2014 10:14 PM
Paper.pdf	Thursday, October 23, 2014 12:37 AM
Popcorn-Time	Tuesday, November 18, 2014 1:55 PM
sample_1.txt	Monday, November 24, 2014 3:37 PM
Screen Shot 2014-11-20 at 9.28.01...	Thursday, November 20, 2014 9:28 PM
Screen Shot 2014-11-23 at 9.00.04...	Sunday, November 23, 2014 9:00 PM
Screen Shot 2014-11-24 at 12.27.2...	Monday, November 24, 2014 12:27 AM
Screen Shot 2014-11-24 at 6.17.09...	Monday, November 24, 2014 6:17 PM
Synctastic 5.0	Saturday, November 22, 2014 5:29 PM
Synctastic 5.0.ipa	Saturday, November 22, 2014 5:31 PM
Test$1.class	Monday, November 24, 2014 1:15 AM

File Format: TEXT-File

Register
R0
R1
R2
R3
R4
R5

Memory Data: Ex. 15 223



Instructions

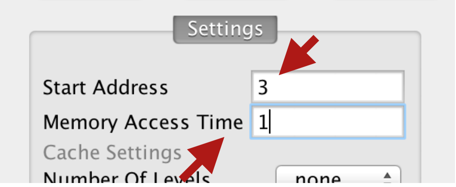
```
#data:

#instructions:
            ADDI R2, R0, 1
            ADDI R3, R0, 5
            MUL R2, R2, R3
            ADDI R3, R3, -1
            BEQ R0, R3, 1
            JMP R0, -4
```

```
Can't run, Please Enter code first
 ====================================
```

Register	Value
R0	0
R1	0
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0

3- If there are any data you want add, in the editor after < #data: > enter the key and the the value with a space separated (ex. 15 223). Any the instructions should go after < #instructions: > .

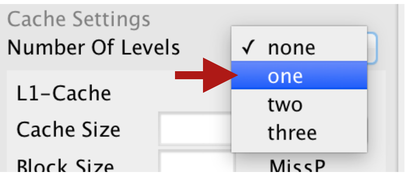
4- Enter the starting address and the Memory Access Time is the Settings panel



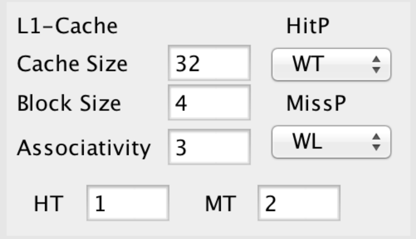
Start Address: 3
Memory Access Time: 1

Testing Level 1-cache (L1)

5- For choosing On Level Cache from the drop-down list of the Number Of Levels and select One.



6- Fill the Settings of the L1-Cache Level: Cache Size, Block Size, Associativity, the Hit/Miss policies and the Hit/Miss times.



Cache Size: 32
HitP: WT
Block Size: 4
MissP: WL
Associativity: 3
HT: 1
MT: 2

Note that:

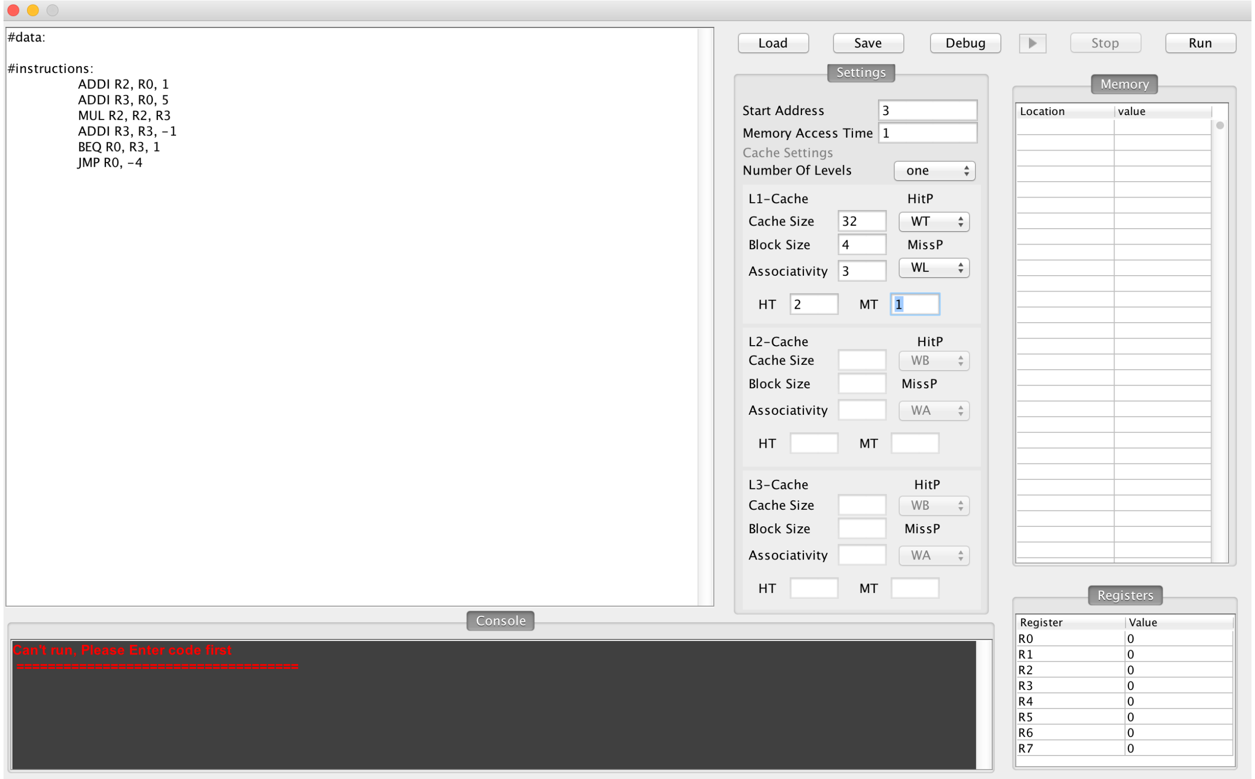
Hit Policies as following:

WB: Write Back,   *WT*: Write Through

Miss Policies as following:

WA: Write Around,   *WL*: Write Allocate

Associativity: ranges from 1 to 4



```
#data:

#instructions:
		ADDI R2, R0, 1
		ADDI R3, R0, 5
		MUL R2, R2, R3
		ADDI R3, R3, -1
		BEQ R0, R3, 1
		JMP R0, -4
```

Start Address 3
Memory Access Time 1
Cache Settings
Number Of Levels one

L1-Cache: Cache Size 32, Block Size 4, Associativity 3, HitP WT, MissP WL, HT 2, MT 1

L2-Cache: Cache Size, Block Size, Associativity, HitP WB, MissP WA, HT, MT

L3-Cache: Cache Size, Block Size, Associativity, HitP WB, MissP WA, HT, MT

Location	value

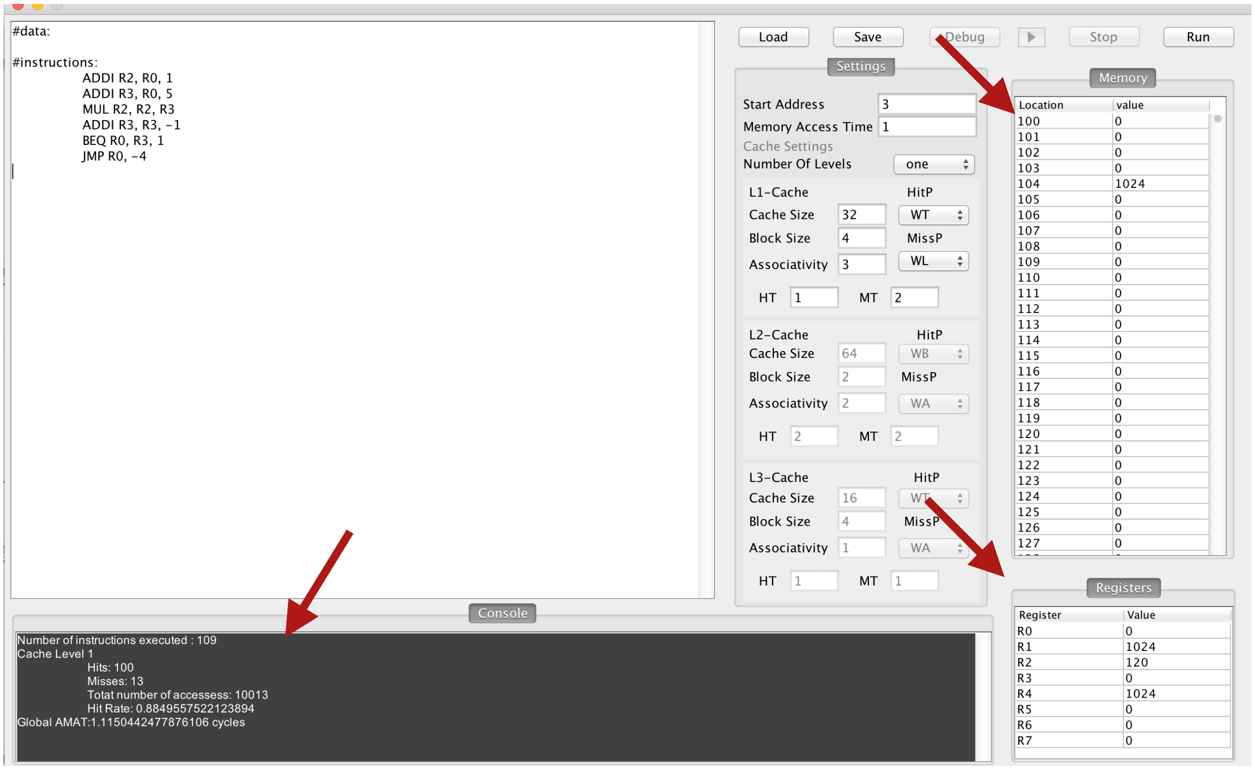
Register	Value
R0	0
R1	0
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0

```
Can't run, Please Enter code first
 ========================================
```

7- After all the settings have been set, we can now run the program.

Results of the Level 1-cache (L1)

8- If every thing ran successfully you will promoted with the following, which contains the results in the console and the Registers/Memory data, but if you forgot anything you will be given errors of what is missing.



```
#data:

#instructions:
		ADDI R2, R0, 1
		ADDI R3, R0, 5
		MUL R2, R2, R3
		ADDI R3, R3, -1
		BEQ R0, R3, 1
		JMP R0, -4
```

Start Address 3
Memory Access Time 1
Cache Settings
Number Of Levels one

L1-Cache: Cache Size 32, Block Size 4, Associativity 3, HitP WT, MissP WL, HT 1, MT 2

L2-Cache: Cache Size 64, Block Size 2, Associativity 2, HitP WB, MissP WA, HT 2, MT 2

L3-Cache: Cache Size 16, Block Size 4, Associativity 1, HitP WT, MissP WA, HT 1, MT 1

Location	value
100	0
101	0
102	0
103	0
104	1024
105	0
106	0
107	0
108	0
109	0
110	0
111	0
112	0
113	0
114	0
115	0
116	0
117	0
118	0
119	0
120	0
121	0
122	0
123	0
124	0
125	0
126	0
127	0

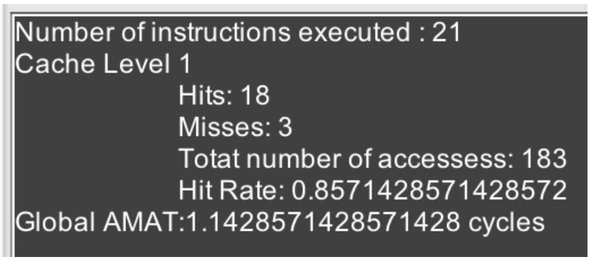
Register	Value
R0	0
R1	1024
R2	120
R3	0
R4	1024
R5	0
R6	0
R7	0

```
Number of instructions executed : 109
Cache Level 1
		Hits: 100
		Misses: 13
		Totat number of accessess: 10013
		Hit Rate: 0.8849557522123894
Global AMAT:1.1150442477876106 cycles
```

Analyzing the Results

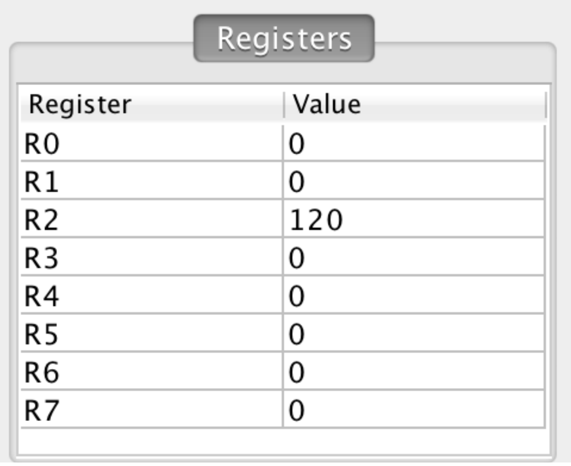
9- The Console results will look like the following :

```
        Cache Level
                Hits: #
                Misses: #
                Total Number Of accesses: #
                Hit Rate: #
                The AMAT: #
```

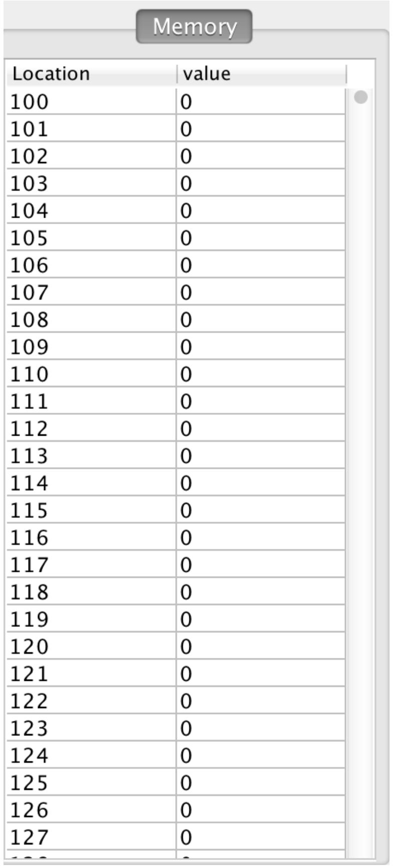


```
Number of instructions executed : 21
Cache Level 1
              Hits: 18
              Misses: 3
              Totat number of accessess: 183
              Hit Rate: 0.8571428571428572
Global AMAT:1.1428571428571428 cycles
```

10- The registers Data will contain all the data from R0 to R7 that has been used in the program.



Register	Value
R0	0
R1	0
R2	120
R3	0
R4	0
R5	0
R6	0
R7	0

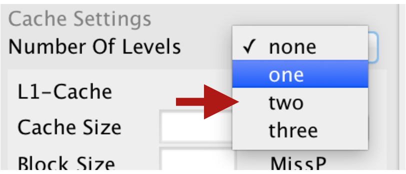


Location	value
100	0
101	0
102	0
103	0
104	0
105	0
106	0
107	0
108	0
109	0
110	0
111	0
112	0
113	0
114	0
115	0
116	0
117	0
118	0
119	0
120	0
121	0
122	0
123	0
124	0
125	0
126	0
127	0

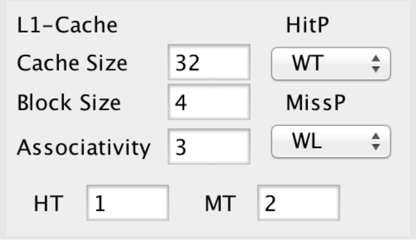
11- The Memory will contain all the data that is stored in the memory.

Testing Level 2-caches (L1- L2)

5- For choosing 2 level laches from the drop-down list of the Number Of Levels we select Two.



6- Assuming the same configuration for the L1-Cache, we configure the L2-cache with smiler settings.



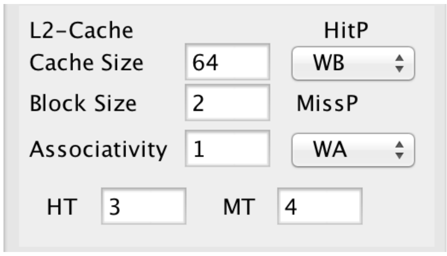
L1-Cache

Cache Size 32 HitP WT

Block Size 4 MissP WL

Associativity 3

HT 1 MT 2



L2-Cache

Cache Size 64 HitP WB

Block Size 2 MissP WA

Associativity 1

HT 3 MT 4

7- After all the settings have been set, we can now run the program.

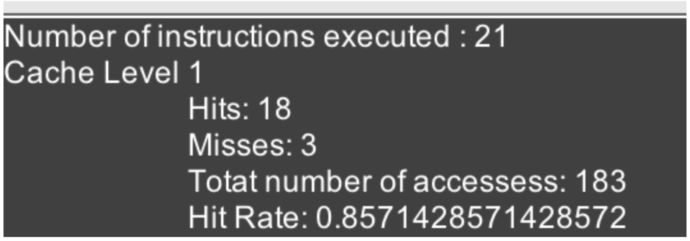
Results of The level 2-cache (L1-L2)

8- Same as the previous one.

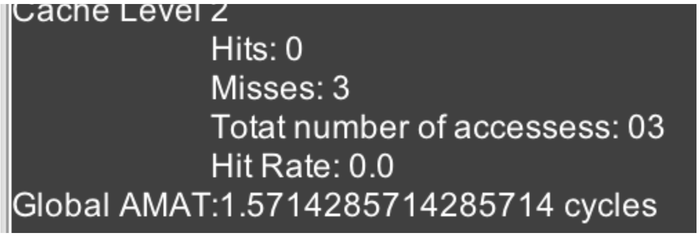
Analyzing the Results

9- The Console results will look like the following :

```
          Cache Level #
                    Hits: #
                    Misses: #
                    Total Number Of accesses: #
                    Hit Rate: #
                    The AMAT: #
```

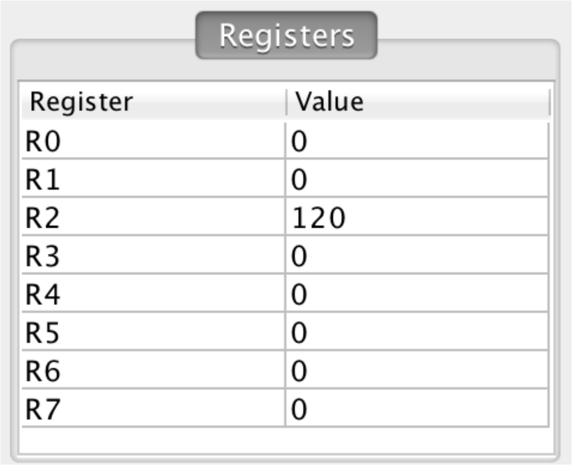


```
Number of instructions executed : 21
Cache Level 1
                    Hits: 18
                    Misses: 3
                    Totat number of accessess: 183
                    Hit Rate: 0.8571428571428572
```

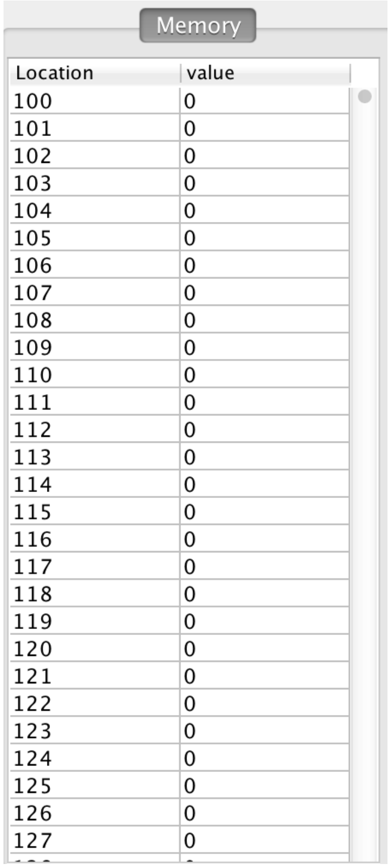


```
Cache Level 2
                    Hits: 0
                    Misses: 3
                    Totat number of accessess: 03
                    Hit Rate: 0.0
Global AMAT:1.5714285714285714 cycles
```

10- The registers Data will contain all the data from R0 to R7 that has been used in the program.



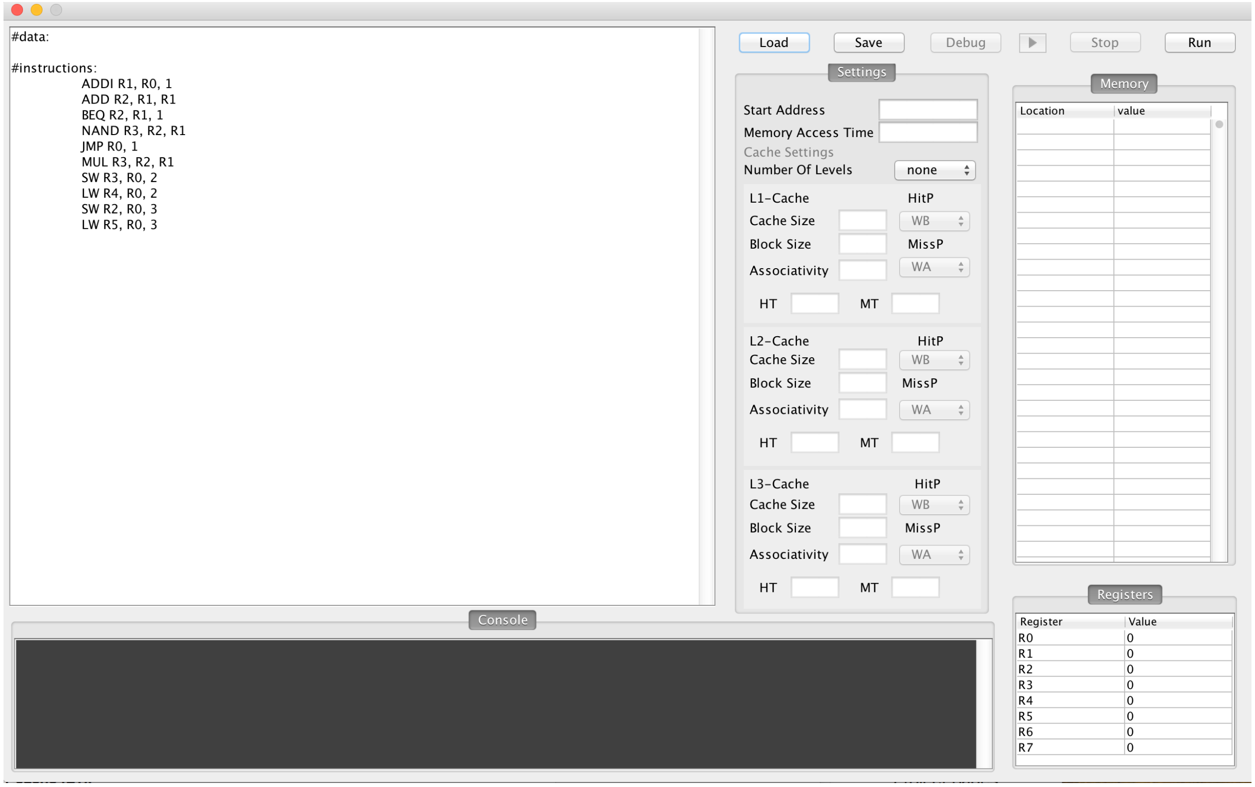
Register	Value
R0	0
R1	0
R2	120
R3	0
R4	0
R5	0
R6	0
R7	0



Location	value
100	0
101	0
102	0
103	0
104	0
105	0
106	0
107	0
108	0
109	0
110	0
111	0
112	0
113	0
114	0
115	0
116	0
117	0
118	0
119	0
120	0
121	0
122	0
123	0
124	0
125	0
126	0
127	0

11- The Memory will contain all the data that is stored in the memory.

Loading Sample_2.txt



```
#data:

#instructions:
	ADDI R1, R0, 1
	ADD R2, R1, R1
	BEQ R2, R1, 1
	NAND R3, R2, R1
	JMP R0, 1
	MUL R3, R2, R1
	SW R3, R0, 2
	LW R4, R0, 2
	SW R2, R0, 3
	LW R5, R0, 3
```

Start Address
Memory Access Time
Cache Settings
Number Of Levels none

L1-Cache HitP
Cache Size WB
Block Size MissP
Associativity WA
HT MT

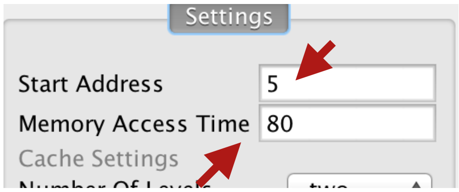
L2-Cache HitP
Cache Size WB
Block Size MissP
Associativity WA
HT MT

L3-Cache HitP
Cache Size WB
Block Size MissP
Associativity WA
HT MT

Register	Value
R0	0
R1	0
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0

Steps 1,2 & 3 are the same as the previous example.

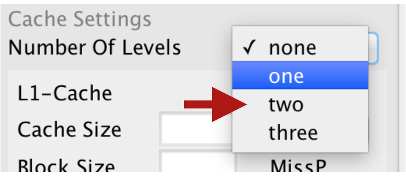
4- Enter the starting address and the Memory Access Time is the Settings panel



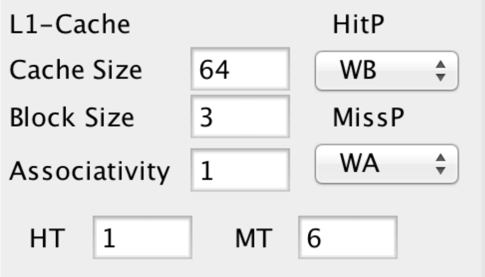
Start Address 5
Memory Access Time 80
Cache Settings
Number Of Levels two

Testing Level 2-cache (L1-L2)

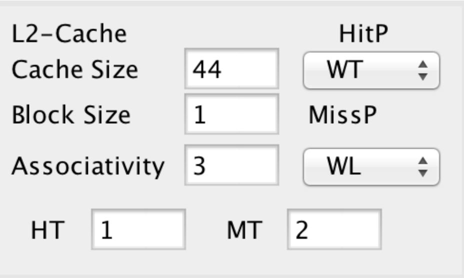
5- For choosing Two Levels of Caches from the dropDown List of the Number Of Levels we select Two .



6- Fill the Settings of the L1-Cache & L2- Cache with the required data.



```
L1-Cache                    HitP
Cache Size     64           WB
Block Size     3            MissP
Associativity  1            WA

  HT  1         MT  6
```



```
L2-Cache                    HitP
Cache Size     44           WT
Block Size     1            MissP
Associativity  3            WL

  HT  1         MT  2
```

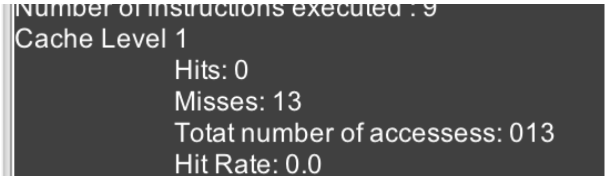
7- After all the setting has been set, we can now run the program.

Results of The 2 levels 2-cache (L1-L2)

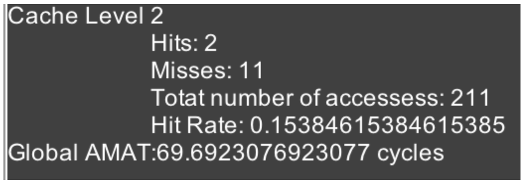
8- Same as the previous example.

Analyzing the Results

9- The Console results will look like the following :

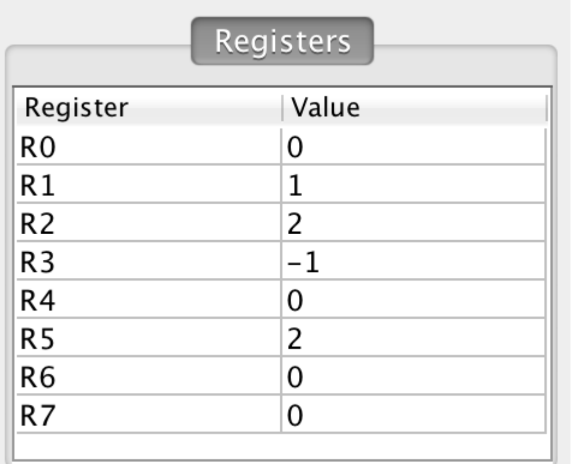


```
Number of instructions executed : 9
Cache Level 1
                Hits: 0
                Misses: 13
                Totat number of accessess: 013
                Hit Rate: 0.0
```



```
Cache Level 2
                Hits: 2
                Misses: 11
                Totat number of accessess: 211
                Hit Rate: 0.15384615384615385
Global AMAT:69.6923076923077 cycles
```

10- The registers Data will contain all the data from R0 to R7 that is been used in the program.

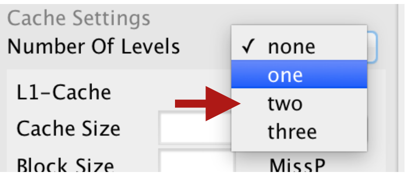


Registers

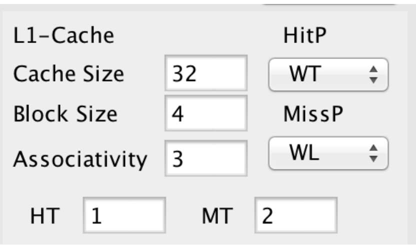
Register	Value
R0	0
R1	1
R2	2
R3	-1
R4	0
R5	2
R6	0
R7	0

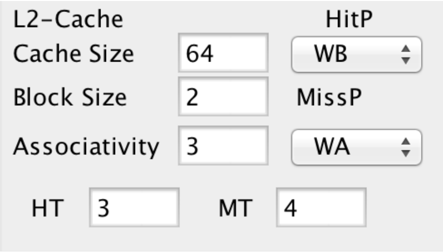
Testing Level 3-caches (L1-L2-L3)

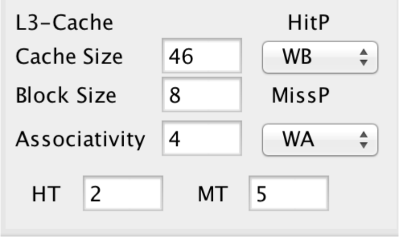
5- For choosing 3 Levels of Caches from the drop-down list of the Number Of Levels we select Three .



6- Assuming the same configuration for the L1-Cache, we configure the L2-cache with smiler settings.


L1-Cache HitP
Cache Size 32 WT
Block Size 4 MissP
Associativity 3 WL
HT 1 MT 2


L2-Cache HitP
Cache Size 64 WB
Block Size 2 MissP
Associativity 3 WA
HT 3 MT 4


L3-Cache HitP
Cache Size 46 WB
Block Size 8 MissP
Associativity 4 WA
HT 2 MT 5

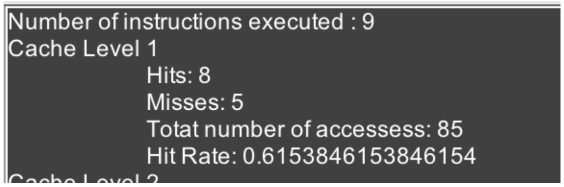
7- After all the setting has been set, we can now run the program.

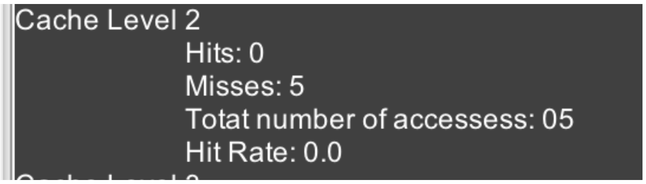
Results of The level 3-caches (L1-L2-L3)

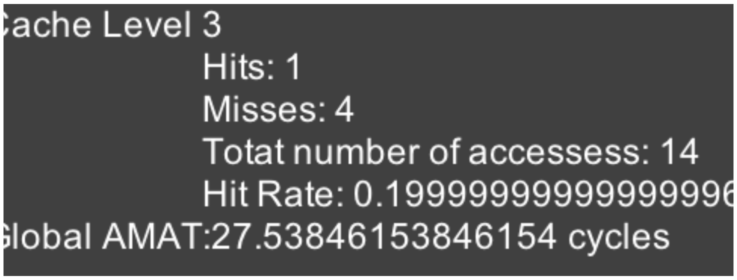
8- Same as the previous one.

Analyzing the Results

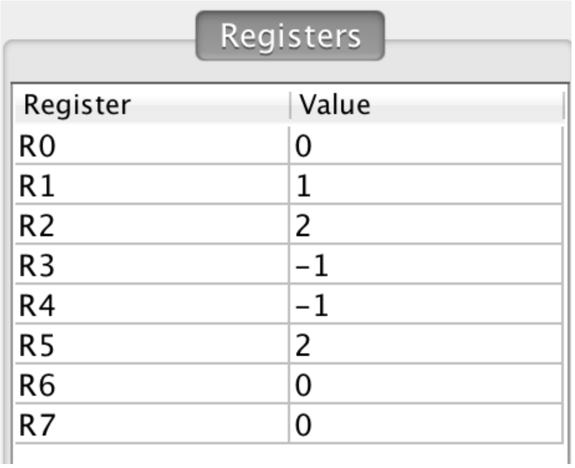
9- The Console results will look like the following :


```
Number of instructions executed : 9
Cache Level 1
                Hits: 8
                Misses: 5
                Totat number of accessess: 85
                Hit Rate: 0.6153846153846154
```

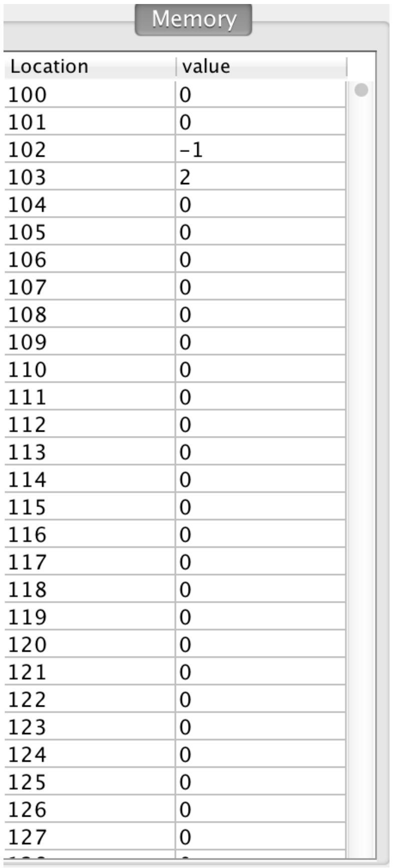

```
Cache Level 2
                Hits: 0
                Misses: 5
                Totat number of accessess: 05
                Hit Rate: 0.0
```


```
ache Level 3
                Hits: 1
                Misses: 4
                Totat number of accessess: 14
                Hit Rate: 0.19999999999999996
lobal AMAT:27.53846153846154 cycles
```

10- The registers Data will contain all the data from R0 to R7 that is been used in the program.



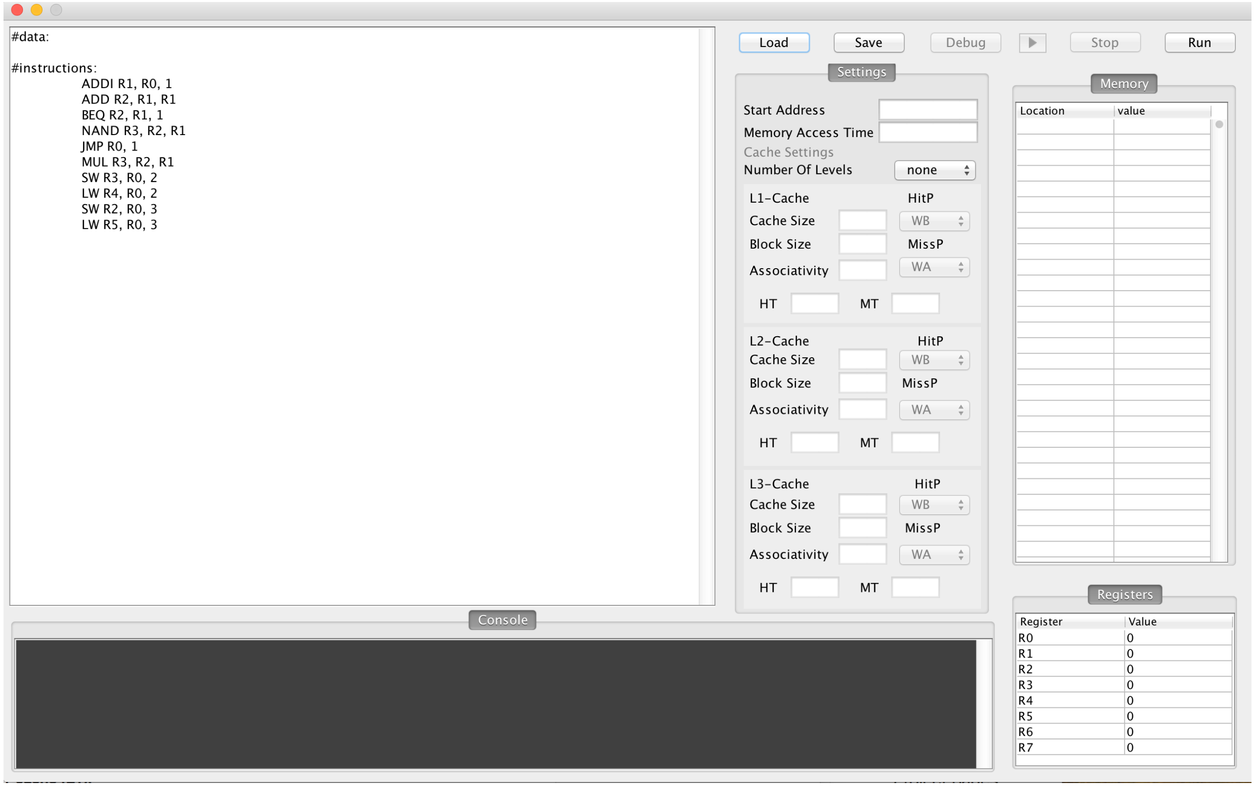
Register	Value
R0	0
R1	1
R2	2
R3	-1
R4	-1
R5	2
R6	0
R7	0



Location	value
100	0
101	0
102	-1
103	2
104	0
105	0
106	0
107	0
108	0
109	0
110	0
111	0
112	0
113	0
114	0
115	0
116	0
117	0
118	0
119	0
120	0
121	0
122	0
123	0
124	0
125	0
126	0
127	0

11- The Memory will contain all the data that is stored in the memory.

Loading Sample_3.txt

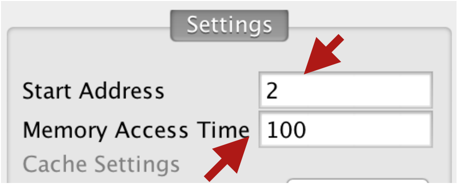


```
#data:

#instructions:
		ADDI R1, R0, 1
		ADD R2, R1, R1
		BEQ R2, R1, 1
		NAND R3, R2, R1
		JMP R0, 1
		MUL R3, R2, R1
		SW R3, R0, 2
		LW R4, R0, 2
		SW R2, R0, 3
		LW R5, R0, 3
```

Register	Value
R0	0
R1	0
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0

Steps 1,2 & 3 are the same as the previous example.

4- Enter the starting address and the Memory Access Time is the Settings panel

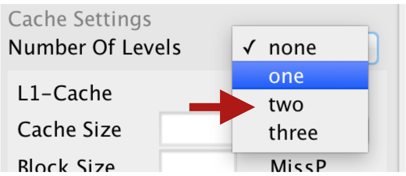


Start Address 2

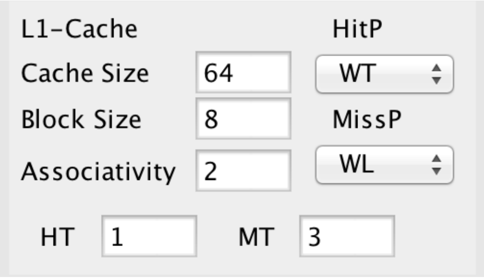
Memory Access Time 100

Testing level 2-cache (L1-L2)

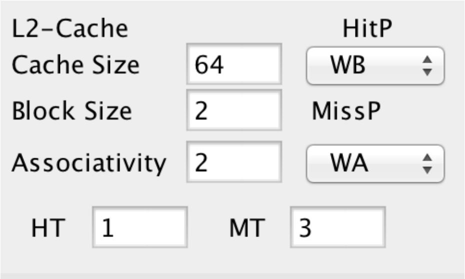
5- For choosing Two Levels of Caches from the dropDown List of the Number of levels we select Two .



6- Fill the Settings of the L1-Cache & L2- Cache with the required data.



```
L1-Cache                 HitP
Cache Size      64       WT
Block Size      8        MissP
Associativity   2        WL

HT   1        MT   3
```



```
L2-Cache                 HitP
Cache Size      64       WB
Block Size      2        MissP
Associativity   2        WA

HT   1        MT   3
```

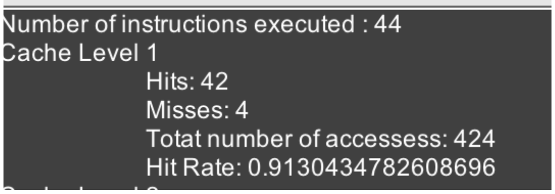
7- After all the setting has been set, we can now run the program.

Results of The 2 levels 2-cache (L1-L2)

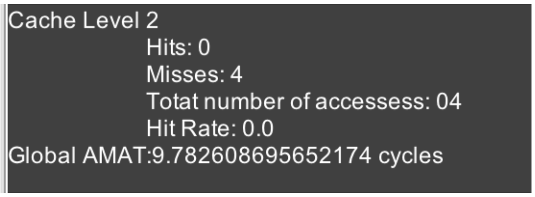
8- Same as the previous example.

Analyzing the Results

9- The Console results will look like the following :

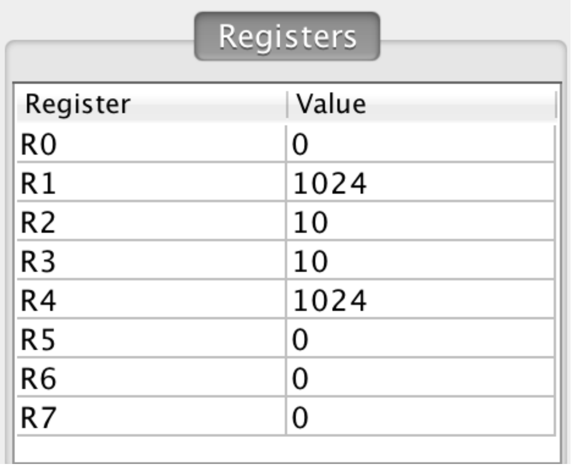


```
Number of instructions executed : 44
Cache Level 1
                Hits: 42
                Misses: 4
                Totat number of accessess: 424
                Hit Rate: 0.9130434782608696
```



```
Cache Level 2
                Hits: 0
                Misses: 4
                Totat number of accessess: 04
                Hit Rate: 0.0
Global AMAT:9.782608695652174 cycles
```

10- The registers Data will contain all the data from R0 to R7 that is been used in the program.

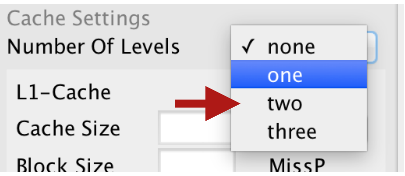


Registers

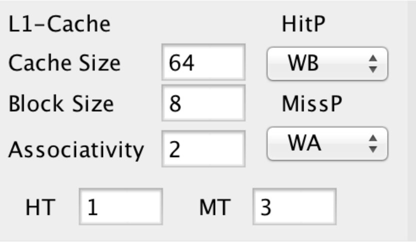
Register	Value
R0	0
R1	1024
R2	10
R3	10
R4	1024
R5	0
R6	0
R7	0

Testing level 3-caches (L1- L2-L3)

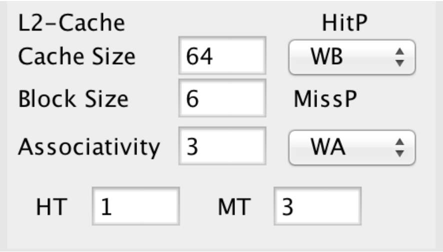
5- For choosing 3 Levels of Caches from the dropDown List of the Number of levels we select Three .



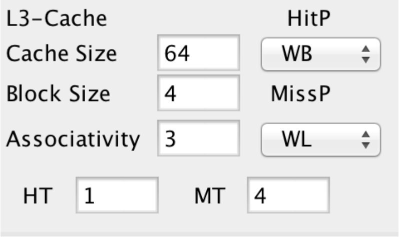
6- Assuming the smiler configuration for the L1-Cache, we configure the L2-cache with smiler settings.



L1-Cache HitP
Cache Size 64 WB
Block Size 8 MissP
Associativity 2 WA
HT 1 MT 3



L2-Cache HitP
Cache Size 64 WB
Block Size 6 MissP
Associativity 3 WA
HT 1 MT 3



L3-Cache HitP
Cache Size 64 WB
Block Size 4 MissP
Associativity 3 WL
HT 1 MT 4

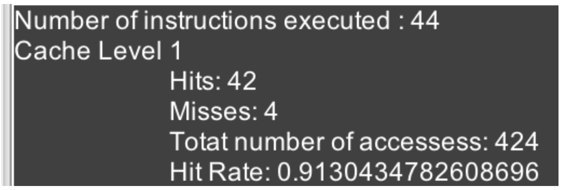
7- After all the setting has been set. now we can run the program.

Results of The level 3-caches (L1-L2-L3)

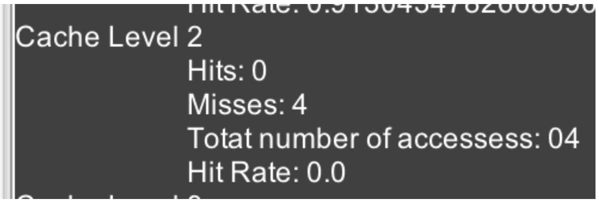
8- Same as the previous one.

Analyzing the Results

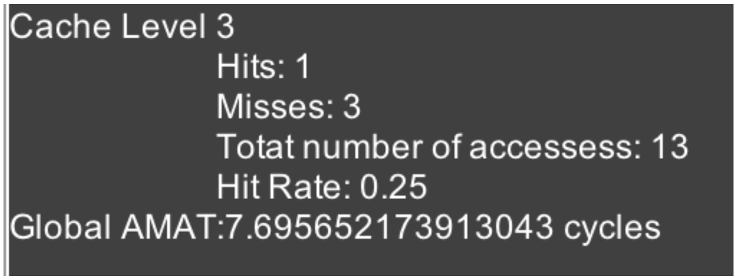
9- The Console results will look like the following :



```
Number of instructions executed : 44
Cache Level 1
                Hits: 42
                Misses: 4
                Totat number of accessess: 424
                Hit Rate: 0.9130434782608696
```

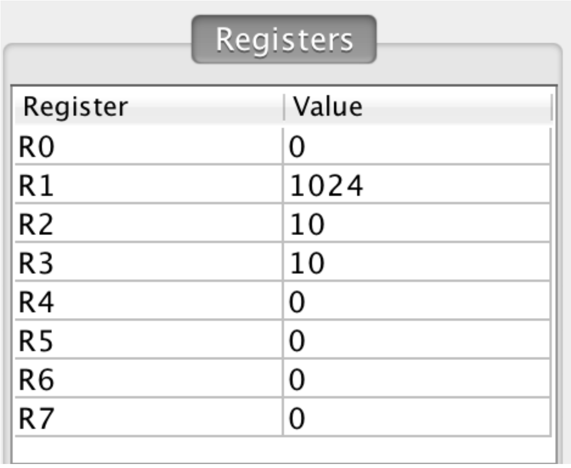


```
Hit Rate: 0.9130434782608696
Cache Level 2
                Hits: 0
                Misses: 4
                Totat number of accessess: 04
                Hit Rate: 0.0
```

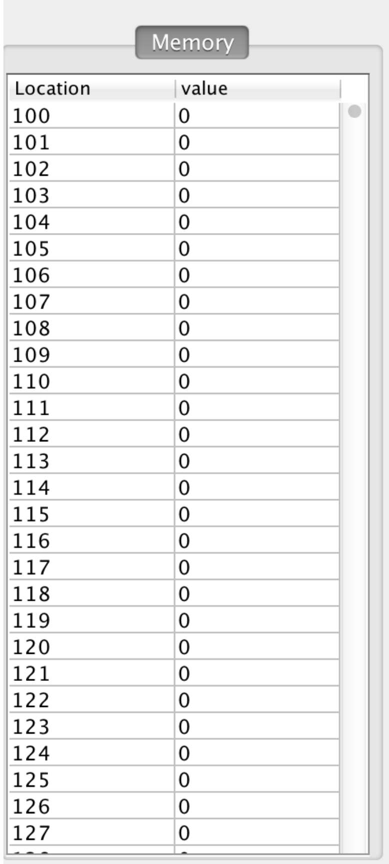


```
Cache Level 3
                Hits: 1
                Misses: 3
                Totat number of accessess: 13
                Hit Rate: 0.25
Global AMAT:7.695652173913043 cycles
```

10- The registers Data will contain all the data from R0 to R7 that is been used in the program.



Register	Value
R0	0
R1	1024
R2	10
R3	10
R4	0
R5	0
R6	0
R7	0



Location	value
100	0
101	0
102	0
103	0
104	0
105	0
106	0
107	0
108	0
109	0
110	0
111	0
112	0
113	0
114	0
115	0
116	0
117	0
118	0
119	0
120	0
121	0
122	0
123	0
124	0
125	0
126	0
127	0

11- The Memory will contain all the data that is stored in the memory.