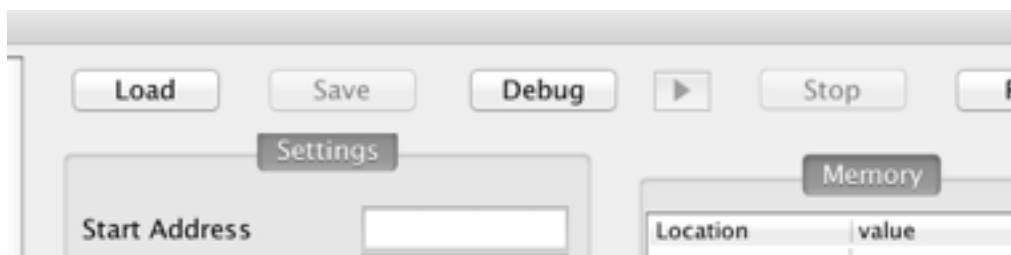


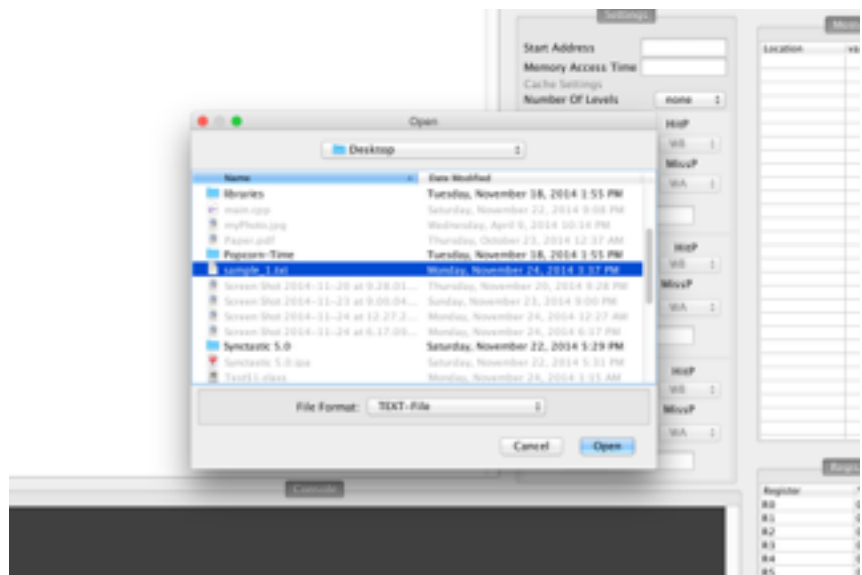


After Opening the program we will be promoted with figure(1). We have 2 ways to input a code, either by loading file or writing the code then save it.

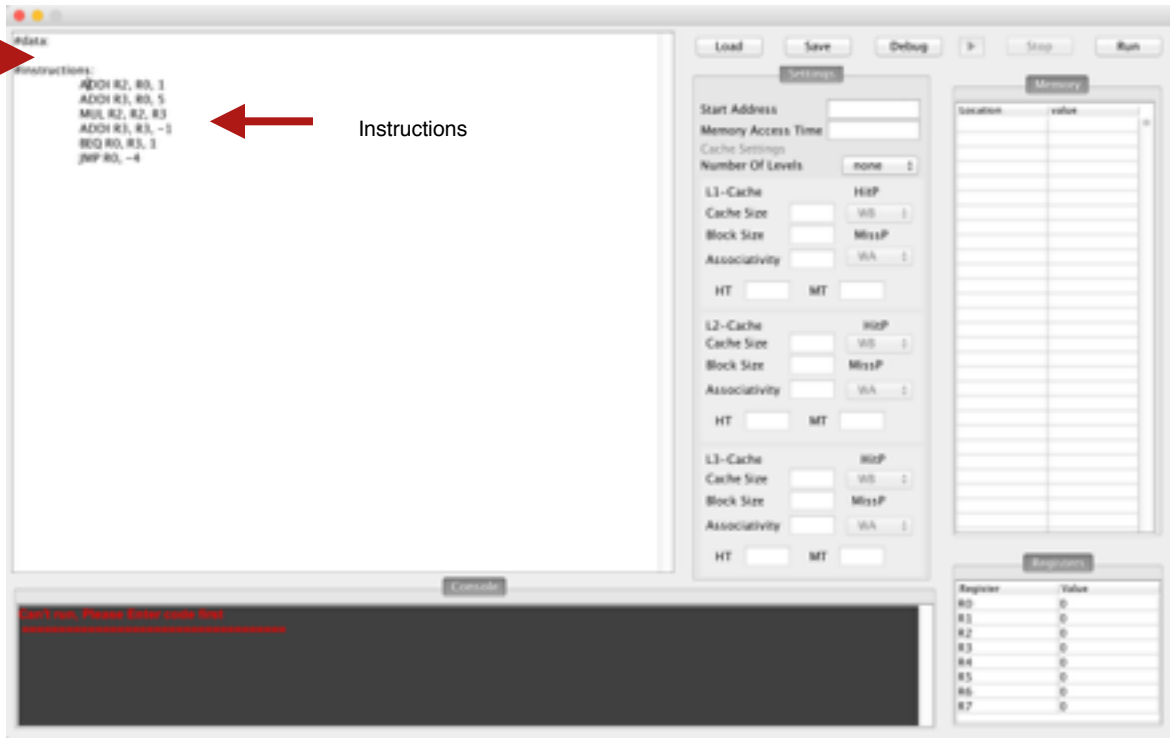


1- We are going to load < sample\_1.txt > by selecting load button as figure(2)

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

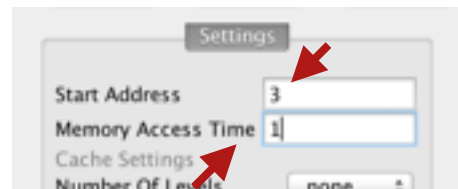


Memory  
Data: Ex.  
15 223



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

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



### Testing level 1-cache (L1).

5- For choosing On Level Cache.. From the dropDown List of the Number of levels we 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.

#### Note that:

Hit Policies as following

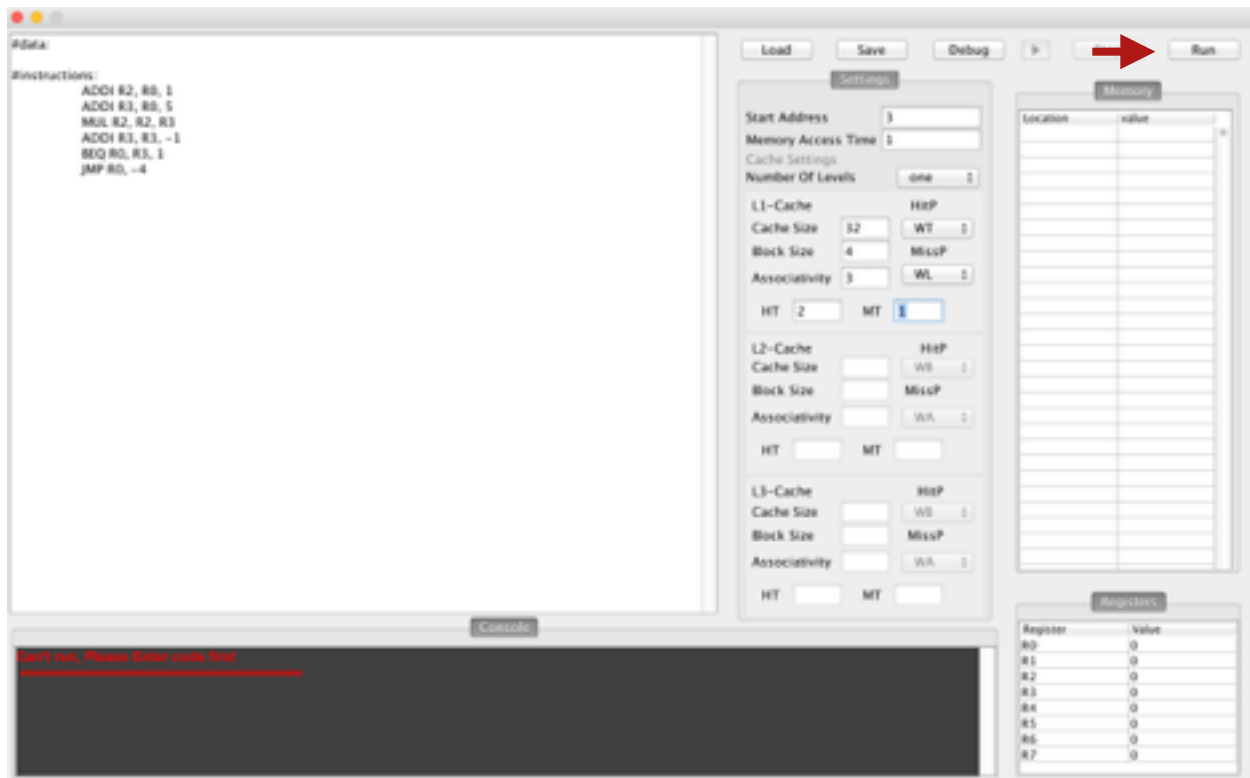
{ WB: Write Back, WT: Write Through}

Miss Policies as following

{ WA: Write Around, WL: Write Allocate}

Associativity : ranges from 1...4

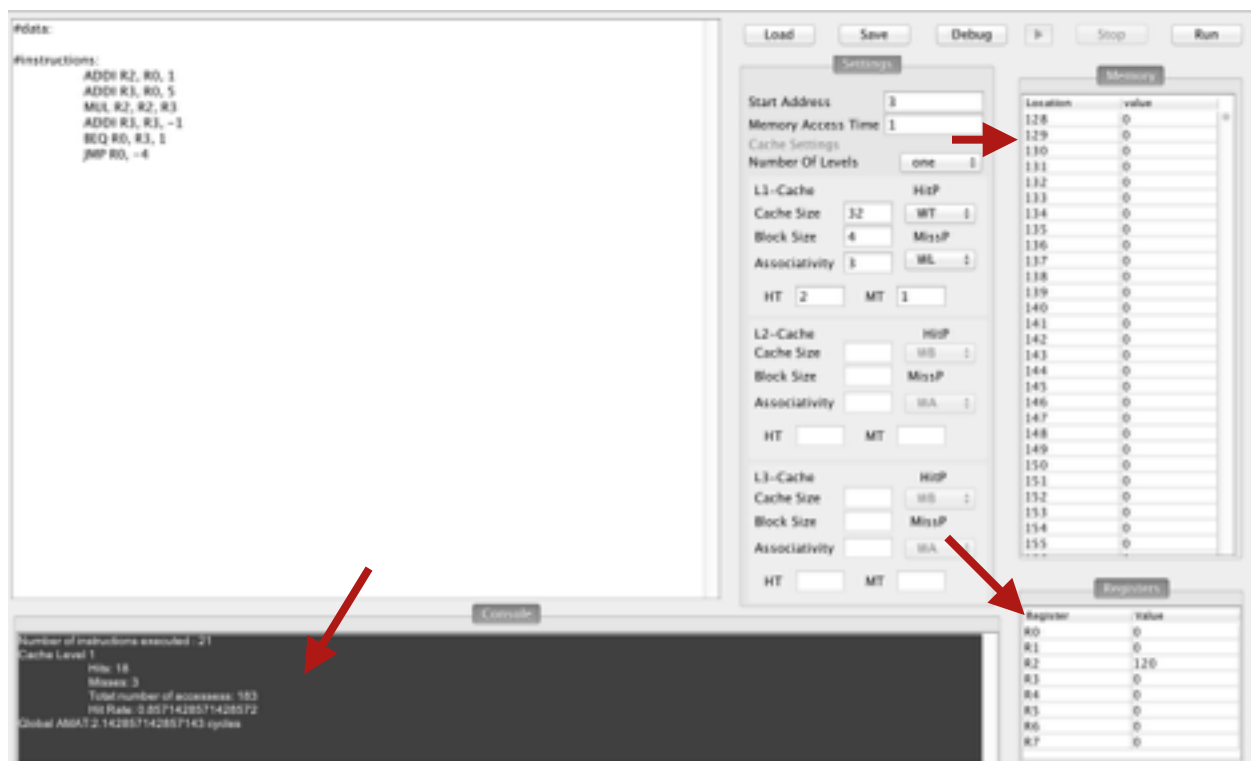




7- After all the setting has been set. now we can run the program.

### Results of The level 1-cache (L1).

8- If every thing ran successfully u will promoted with figure <3> which contains the result in the console and the registers/Memory data , but if you forgot anything you will be given error/s of what is missing.



## 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
    Total number of accesses: 183
    Hit Rate: 0.8571428571428572
Global AMAT:2.142857142857143 cycles
```

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

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

Memory	
Location	value
128	0
129	0
130	0
131	0
132	0
133	0
134	0
135	0
136	0
137	0
138	0
139	0
140	0
141	0
142	0
143	0
144	0
145	0
146	0
147	0
148	0
149	0
150	0
151	0
152	0
153	0
154	0
155	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 Caches.. From the dropDown 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		HitP	
Cache Size	32	WT	÷
Block Size	4	MissP	
Associativity	3	WL	÷
HT	2	MT	1

L2-Cache		HitP	
Cache Size	64	WB	÷
Block Size	2	MissP	
Associativity	1	WA	÷
HT	3	MT	4

7- After all the setting has been set. now we can 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: #
```

```
Cache Level 2
Hits: 0
Misses: 4
Total number of accessess: 04
Hit Rate: 0.0
Global AMAT:2.380952380952381 cycles
```

```
Number of instructions executed : 42
Cache Level 1
Hits: 38
Misses: 4
Total number of accessess: 384
Hit Rate: 0.9047619047619048
```

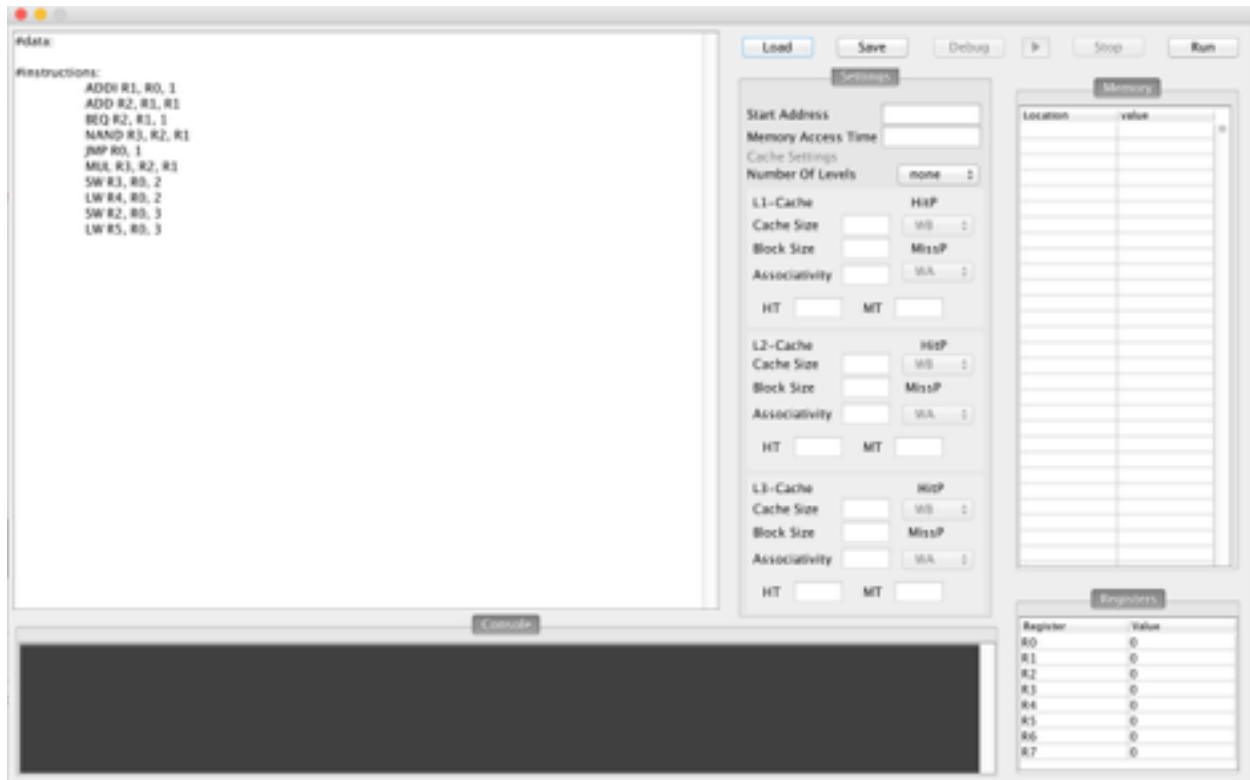
10- The registers Data will contain all the data from R0...R7 that is been used in the program.

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

Memory	
Location	value
128	0
129	0
130	0
131	0
132	0
133	0
134	0
135	0
136	0
137	0
138	0
139	0
140	0
141	0
142	0
143	0
144	0
145	0
146	0
147	0
148	0
149	0
150	0
151	0
152	0
153	0
154	0
155	0

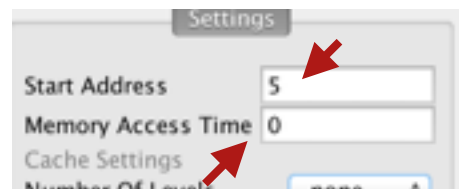
11- The Memory Will contain all the data that is stored in the memory.

## Loading Sample\_2.txt



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

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



## 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.

L2-Cache		HitP	
Cache Size	<input type="text" value="44"/>		<input type="button" value="WT"/>
Block Size	<input type="text" value="1"/>	MissP	
Associativity	<input type="text" value="3"/>		<input type="button" value="WL"/>
HT	<input type="text" value="2"/>	MT	<input type="text" value="2"/>

L1-Cache		HitP	
Cache Size	<input type="text" value="64"/>		<input type="button" value="WB"/>
Block Size	<input type="text" value="4"/>	MissP	
Associativity	<input type="text" value="1"/>		<input type="button" value="WA"/>
HT	<input type="text" value="6"/>	MT	<input type="text" value="1"/>

7- After all the setting has been set. now we can 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: 2
  Misses: 11
  Total number of accesses: 211
  Hit Rate: 0.15384615384615385
```

```
Cache Level 2
  Hits: 1
  Misses: 10
  Total number of accesses: 110
  Hit Rate: 0.09090909090909094
Global AMAT:7.6923076923076925 cycles
```

10- The registers Data will contain all the data from R0...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 dropDown List of the Number of levels we select Three .



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

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

L2-Cache		HitP	
Cache Size	64	WB	:
Block Size	2	MissP	:
Associativity	1	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. 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 : 18
Cache Level 1
  Hits: 6
  Misses: 20
  Total number of accessess: 620
  Hit Rate: 0.23076923076923073
```

```
Cache Level 2
  Hits: 0
  Misses: 20
  Total number of accessess: 020
  Hit Rate: 0.0
```

```
Cache Level 3
  Hits: 14
  Misses: 6
  Total number of accessess: 146
  Hit Rate: 0.7
Global AMAT:9.076923076923077 cycles
```

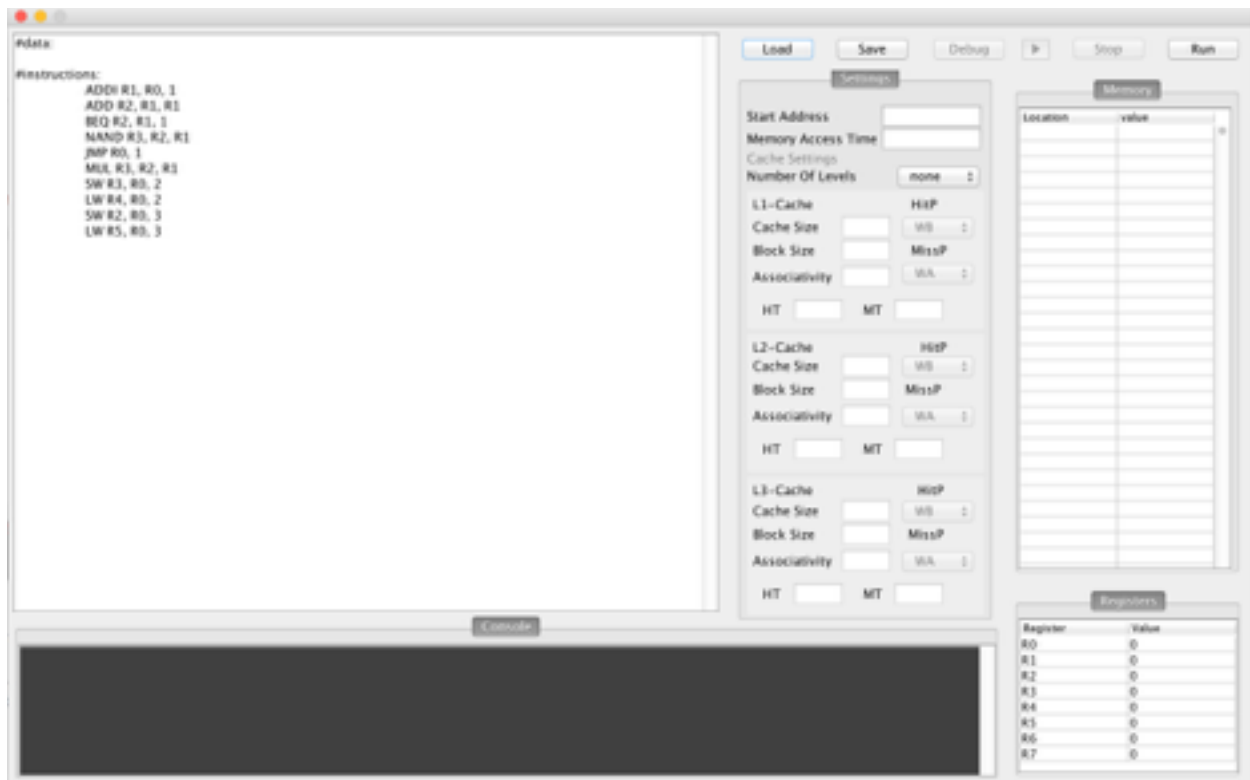
10- The registers Data will contain all the data from R0...R7 that is been used in the program.

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

Memory	
Location	value
100	0
101	0
102	-1
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
...	...

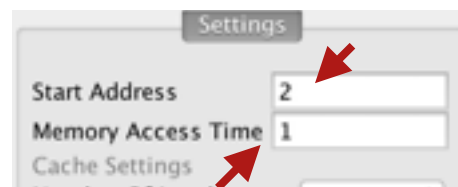
11- The Memory Will contain all the data that is stored in the memory.

## Loading Sample\_3.txt



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

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



## 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.

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

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

7- After all the setting has been set. now we can 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: 0
  Misses: 46
  Total number of accesses: 046
  Hit Rate: 0.0
```

```
Cache Level 2
  Hits: 39
  Misses: 7
  Total number of accesses: 397
  Hit Rate: 0.8478260869565217
Global AMAT:4.1521739130434785 cycles
```

10- The registers Data will contain all the data from R0...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 same configuration for the L1-Cache, we configure the L2-cache with similar settings.

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

L2-Cache		HitP	
Cache Size	64	WB	↓↑
Block Size	2	MissP	
Associativity	2	WA	↓↑
HT	2	MT	2

L3-Cache		HitP	
Cache Size	16	WT	↓↑
Block Size	4	MissP	
Associativity	1	WA	↓↑
HT	1	MT	1

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 : 88
Cache Level 1
  Hits: 2
  Misses: 90
  Total number of accessess: 290
  Hit Rate: 0.021739130434782594
```

```
Cache Level 2
  Hits: 79
  Misses: 11
  Total number of accessess: 7911
  Hit Rate: 0.8777777777777778
```

```
Cache Level 3
  Hits: 0
  Misses: 11
  Total number of accessess: 011
  Hit Rate: 0.0
Global AMAT:4.195652173913043 cycles
```

10- The registers Data will contain all the data from R0...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

Memory	
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

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