



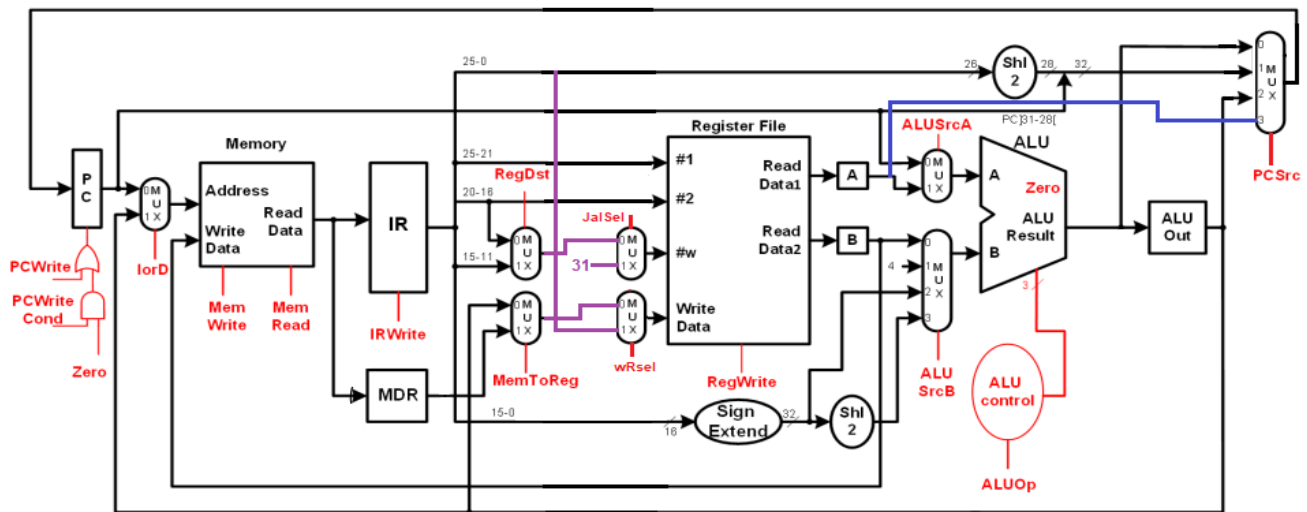
گزارش پروژه ۳ درس سیستم های دیجیتال ۲

اعضای گروه :

سینا طبسی (۸۱۰۱۹۹۵۵۴)

سیدحامد میرامیرخانی (۸۱۰۱۹۹۵۰۰)

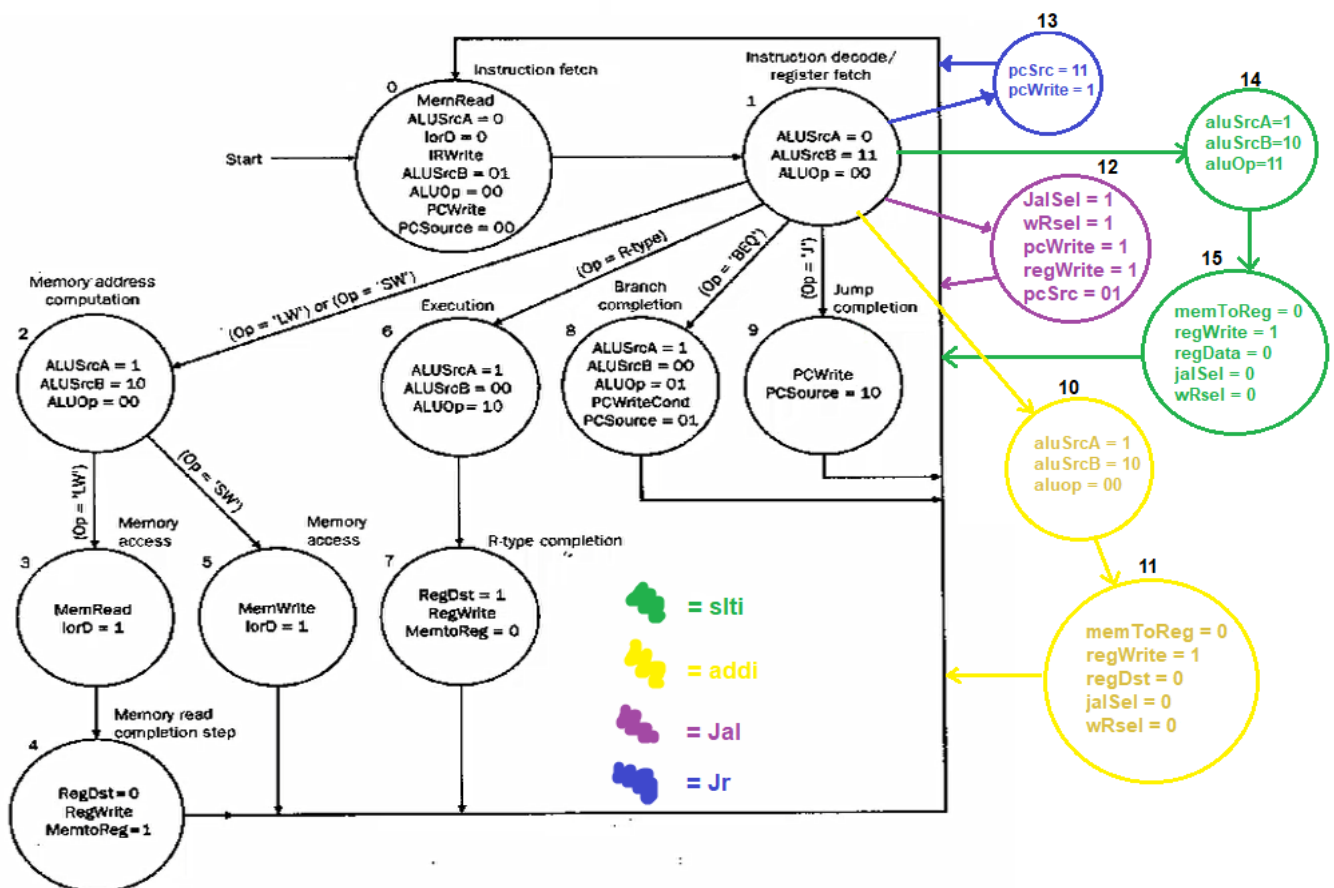
Data Path :



= Jr

= Jal

Controler :



دستورات مختلف به همراه آپکودهایشان (برای دستورات Rtype بیت های func هم مشخص است)

```
// RType instructions
6'b000000 : ns = 6;

// Load Word (lw) instruction
6'b100011 : ns = 2;
// Store Word (sw) instruction
6'b101011 : ns = 2;

// Branch on equal (beq) instruction
6'b000100 : ns = 8;

//Jump
6'b000010: ns = 9 ;
//Jump And Link
6'b000011: ns = 12 ;
//Jump Register
6'b000110: ns = 13 ;

// Add immediate (addi) instruction
6'b001001: ns = 10;
// Set on less than immediate
6'b001010: ns = 14;
```

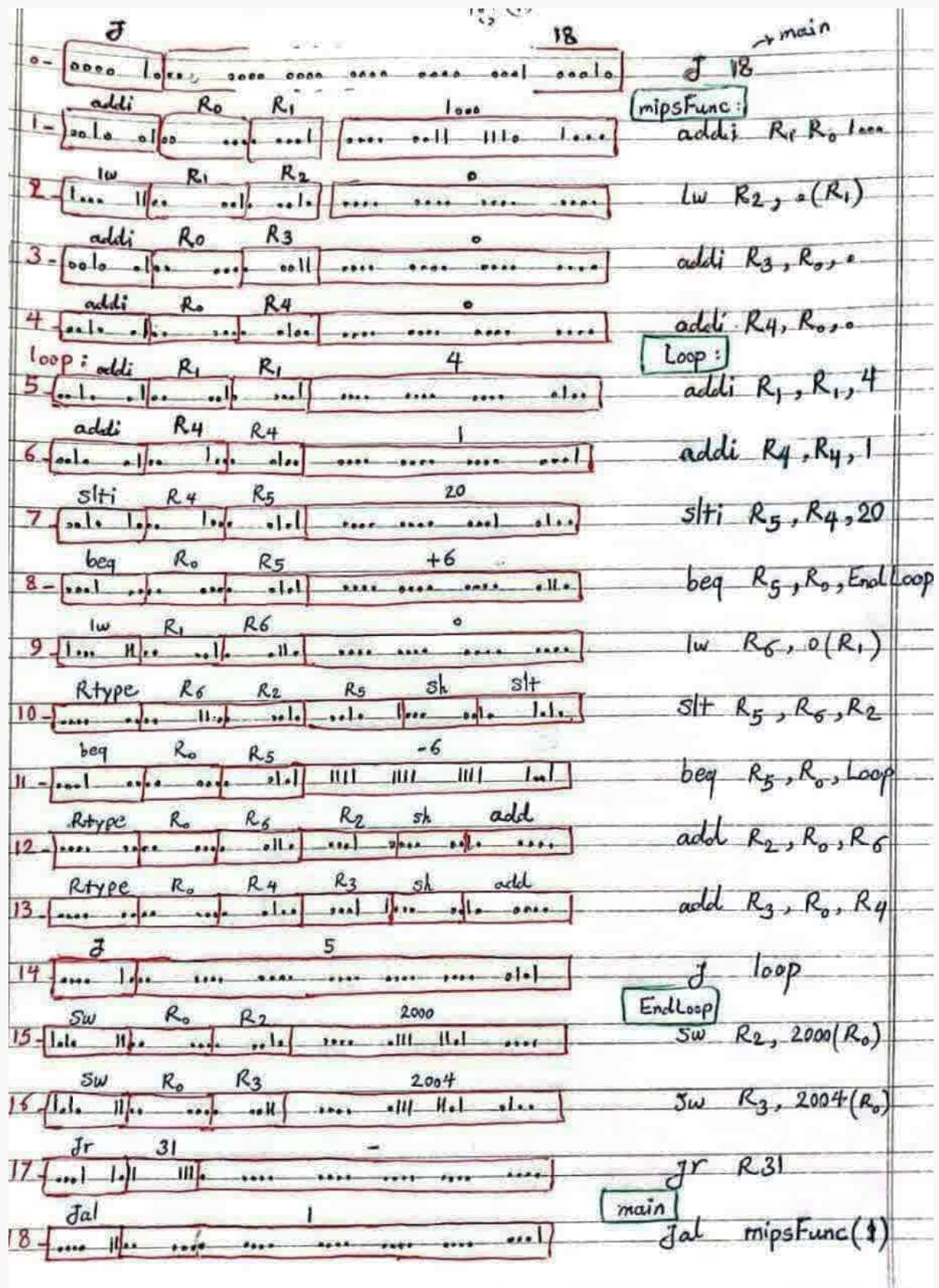
```
case (func)
  6'b100000: operation = 3'b010; // add
  6'b100011: operation = 3'b110; // sub
  6'b100100: operation = 3'b000; // and
  6'b100101: operation = 3'b001; // or
  6'b101010: operation = 3'b111; // slt
```

```
1  j main
2
3  mipsFunc:
4      addi R1, R0, 1000
5      lw R2, 0(R1)
6      addi R3, R0, 0
7      addi R4, R0, 0
8
9      Loop:
10         addi R1, R1, 4
11         addi R4, R4, 1
12         slti R5, R4, 20
13         beq R5, R0, EndLoop
14         lw R6, 0(R1)
15         slt R5, R6, R2
16         beq R5, R0, Loop
17         add R2, R0, R6
18         add R3, R0, R4
19         add R3, R0, R4
20         j Loop
21
22     EndLoop:
23         sw R2, 2000(R0)
24         sw R3, 2004(R0)
25         jr R31
26
27 main:
28     jal mipsFunc
```

برای بررسی ، برنامه روبه رو اجرا میشود که هدف آن این است که یک آرایه ۲۰ تایی از عناصر را بگیرد (مکان این عناصر از خانه ۱۰۰۰ حافظه شروع میشود) و در نهایت کوچکترین عنصر آنها را بازگرداند.

کد اسمبلی

پس در واقع لیست دستورات ما (برای فهم بهتر سمت راست هر خط دستور، شکل اسمبلی آن نیز نوشته شده) به این صورت خواهد بود:



حالا به حافظه دقت کنید و دیتاهایی که در آن وجود دارد، همانطور که گفته شد از خانه ۲۰۰۰ شروع میکنیم (3e8 معادل هگزادسیمال ۲۰۰۰ است).

```

1  @3e8
2  01000101
3  00000000
4  00000000
5  00000000
6  @3ec
7  01100101
8  00000000
9  00000000
10 00000000
11 @3f0
12 10101100
13 11111111
14 11111111
15 11111111
16 @3f4
17 01100110
18 00000000
19 00000000
20 00000000
21 @3f8
22 00000011
23 00000000
24 00000000
25 00000000
26 @3fc
27 00000000
28 00000000
29 00000000
30 00000000

```

```

31 @400
32 11111100
33 11111111
34 11111111
35 11111111
36 @404
37 00001111
38 00000000
39 00000000
40 00000000
41 @408
42 01010101
43 00000000
44 00000000
45 00000000
46 @40c
47 11100010
48 11111111
49 11111111
50 11111111
51 @410
52 00001101
53 00000000
54 00000000
55 00000000
56 @414
57 01111011
58 00000000
59 00000000
60 00000000

```

```

61 @418
62 10100001
63 11111111
64 11111111
65 11111111
66 @41c
67 10001111
68 11111111
69 11111111
70 11111111
71 @420
72 11110001
73 11111111
74 11111111
75 11111111
76 @424
77 00010111
78 00000000
79 00000000
80 00000000

```

```

81 @428
82 01001100
83 00000000
84 00000000
85 00000000
86 @42c
87 01001100
88 00000000
89 00000000
90 00000000
91 @430
92 10111011
93 11111111
94 11111111
95 11111111
96 @434
97 00010101
98 00000000
99 00000000
100 00000000

```

در فایل
dataMem.mem
این موارد نوشته شده
است.

```

1  69
2  101
3  -84
4  102
5  3
6  0
7  -4
8  15
9  85
10 -30

```

```

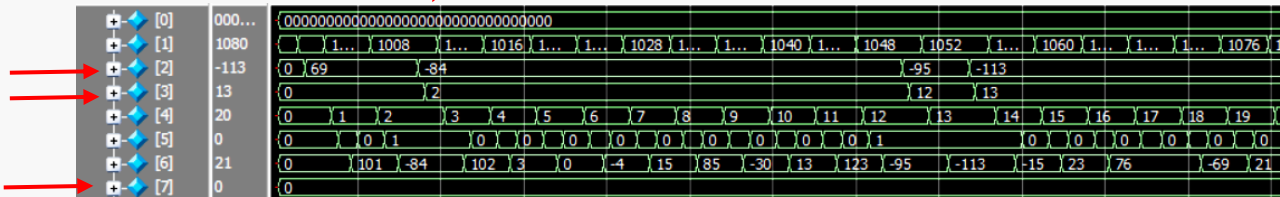
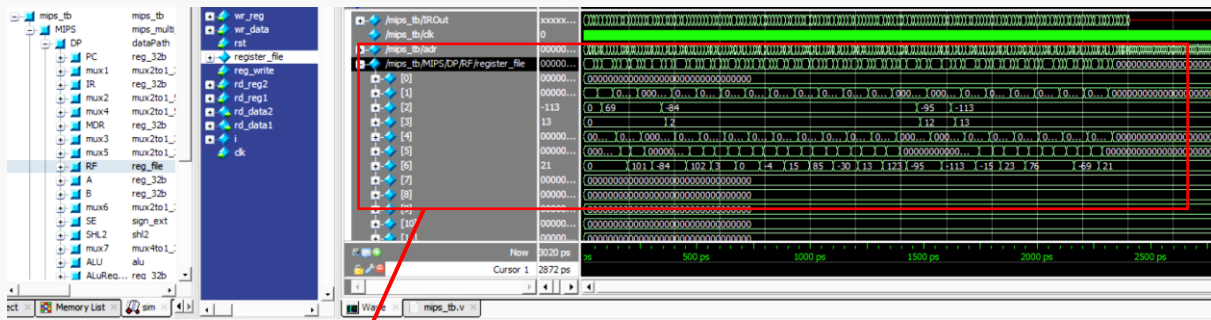
11  13
12  123
13  -95
14  -113
15  -15
16  23
17  76
18  76
19  -69
20  21

```

محتویات این ۲۰ عنصر
حافظه هم مطابق رو به
روست.

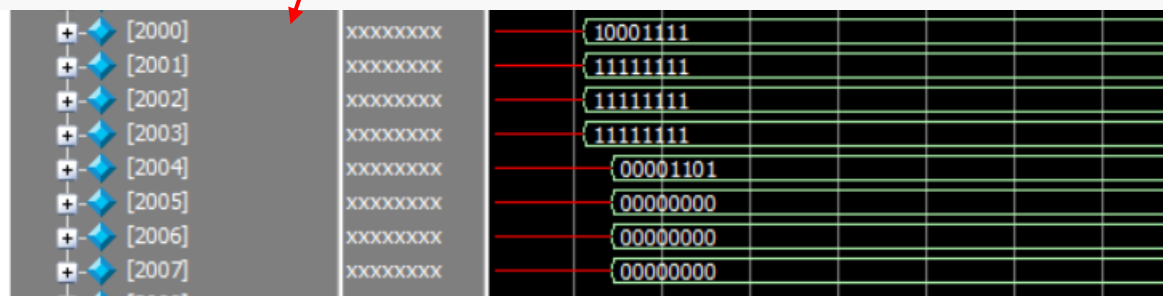
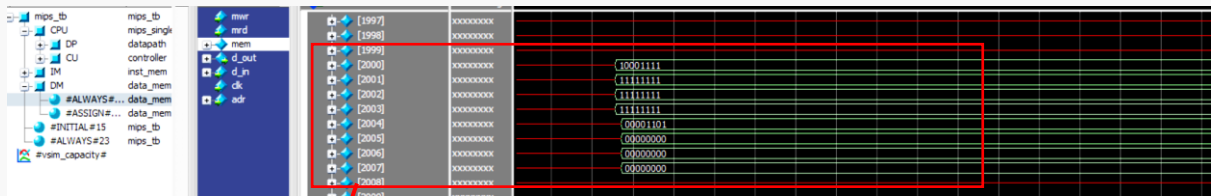
در فایل arrayData.txt
نوشته شده است.

اتفاقاتی که در رجیستر فایل می افتد :

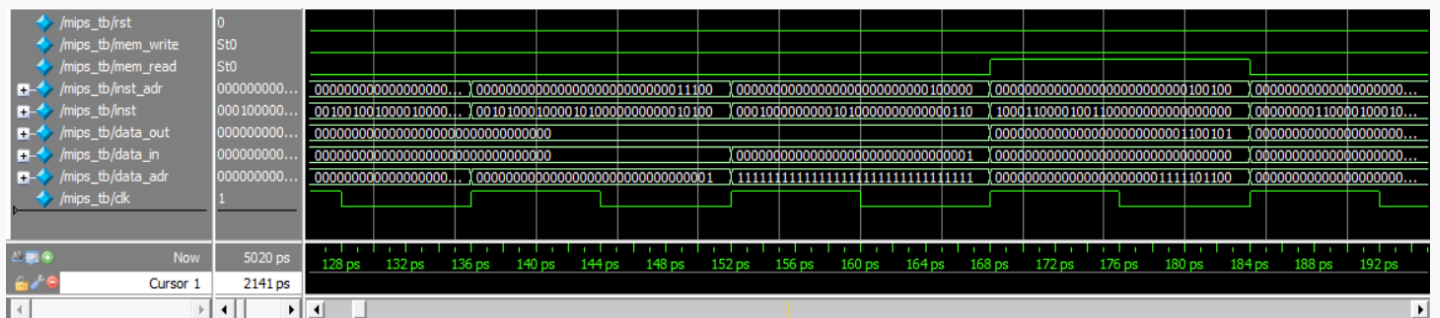
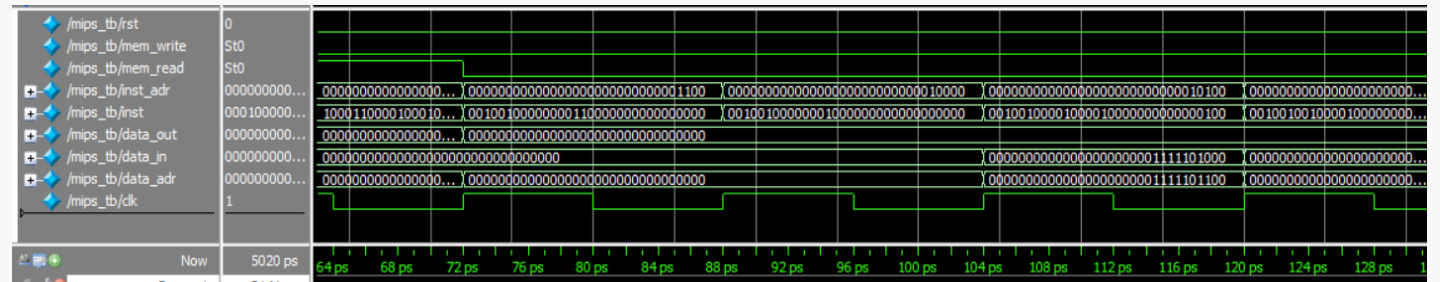
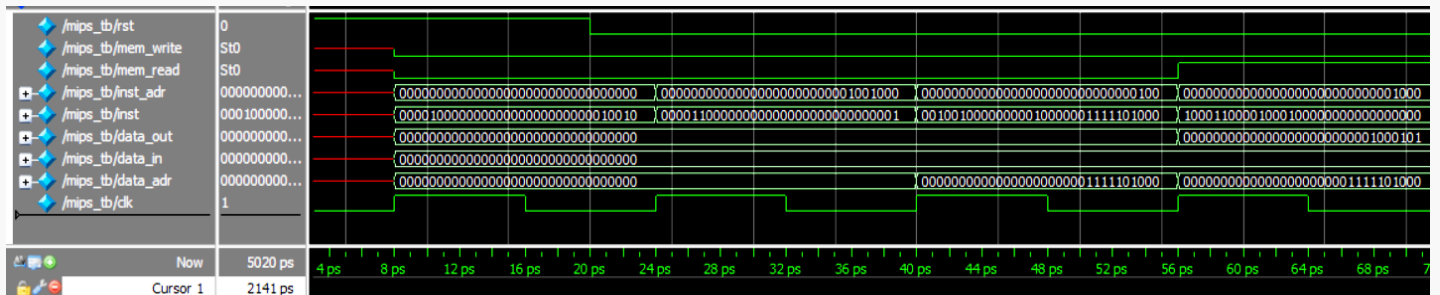


همانطور که مشاهده میکنید عناصر آرایه پیمایش میشوند و هربار چک میشود آیا مقدار R6 از R2 کوچکتر است یا خیر، اگر بود مقدار R2 آپدیت میشود و اندیس متناظر با مقدار جدید در R3 ثبت میشود.

در نهایت کوچکترین عنصر آرایه در خانه ۲۰۰ حافظه ذخیره میشود و مقدار اندیس آن هم در خانه ۲۰۴ حافظه. یعنی در این مثال عدد ۱۱۳- در خانه های ۲۰۰-۲۰۳ حافظه و اندیس آن یعنی ۱۳ در خانه های ۲۰۴-۲۰۷ قرار میگیرند.



خروجی تست بنج بصورت کلی :



بهمین صورت تا انتها دستورات مختلف (برای درک بهتر به کد اسمبلی که پیشتر گفته شد دقت کنید) اجرا میشوند.

نمای کلی خروجی مدل‌سیم بصورت زیر خواهد بود :

