

گزارش تمرین کامپیوتری اول درس معماری کامپیوتر پیشرفته

امیررضا غلامی

شماره دانشجویی : ۸۱۰۱۹۸۴۴۶

پارسا منفرد

شماره دانشجویی :

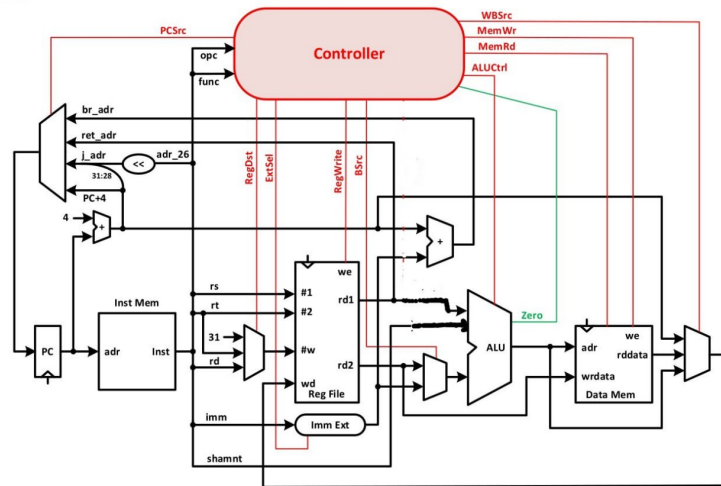
۲۷ مهر ۱۴۰۳

فهرست مطالب

۳	۱ پیاده سازی single cycle
۳	۱.۱ برنامه تست
۶	۲.۱ نتایج شبیه سازی
۷	۲ پیاده سازی پایپ لاین
۷	۱.۲ برنامه تست پایپ لاین
۱۰	۲.۲ نتایج شبیه سازی

۱ پیاده سازی single cycle

شماتیک پیاده سازی single cycle به صورت زیر است.



شکل ۱: شماتیک پیاده سازی single cycle

۱.۱ برنامه تست

برنامه پیدا کردن بزرگترین عنصر بین ۱۰ عنصر داخل حافظه داده در شکل (؟؟) به تصویر کشیده شده است.

```

1  .data
2  array: .word 5, -3, 12, 7, -8, 19, 0, -15, 22, 10 # 10-element array
3  length: .word 10
4  maxValue: .word 0
5
6  .text
7  .globl main
8
9  main:
10 # Initialize pointers and load the first element of the array
11 la $t0, array # $t0 points to the start of the array
12 lw $t1, ($t0) # Load the first element into $t1 (current max)
13 li $t2, 1 # $t2 is the index, starting from 1
14
15 lw $t3, length # Load the length of the array into $t3
16 sub $t3, $t3, 1 # We already considered the first element, so $t3 = length - 1
17
18 loop:
19 beq $t3, $zero, done # If $t3 is 0, we are done
20 lw $t4, 0($t0) # Load current array element into $t4
21 addi $t0, $t0, 4 # Move to the next array element
22
23 # Compare $t4 with current max in $t1
24 ble $t4, $t1, skip # If $t4 <= $t1, skip updating max
25 move $t1, $t4 # Update max to $t4
26
27 skip:
28 sub $t3, $t3, 1 # Decrease $t3 (remaining elements to check)
29 j loop # Repeat loop
30
31 done:
32 sw $t1, maxValue # Store the maximum value in memory
33
34 infinite_loop:
35 j infinite_loop # Jump to itself to create an infinite loop

```

شکل ۲: برنامه اسمبلی تست

برنامه اسمبلی را به زبان ماشین (باینری) تبدیل می کنیم

```

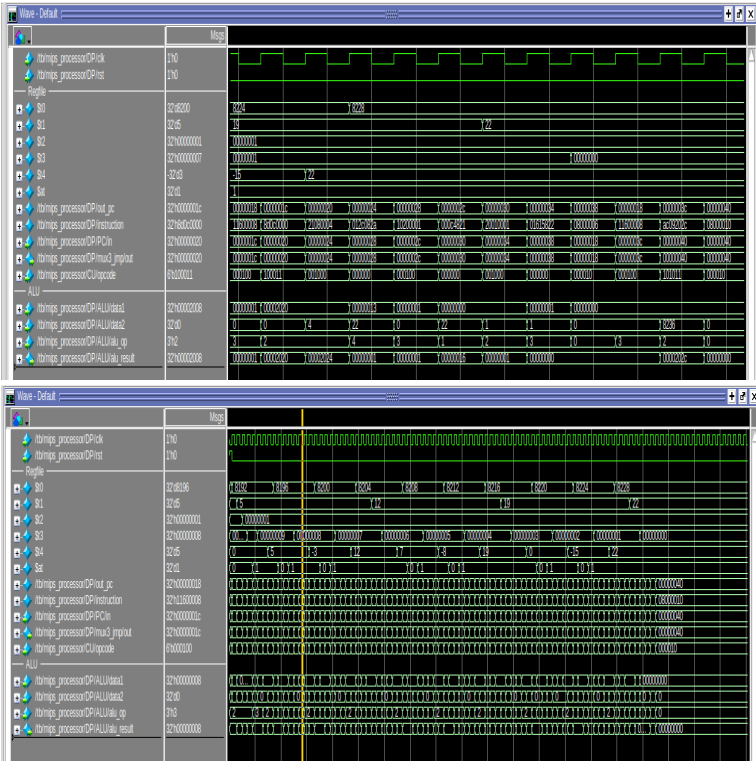
1  00100000000010000010000000000000
2  10001101000010010000000000000000
3  00100100000010100000000000000001
4  10001100000010110010000000101000
5  00100000000000010000000000000001
6  00000001011000010101100000100010
7  00010001011000000000000000001000
8  10001101000011000000000000000000
9  001000010000100000000000000000100
10 00000001001011000000100000101010
11 00010000001000000000000000000001
12 00000000000011000100100000100001
13 00100000000000010000000000000001
14 00000001011000010101100000100010
15 00001000000000000000000000000110
16 10101100000010010010000000101100
17 00001000000000000000000000001000

```

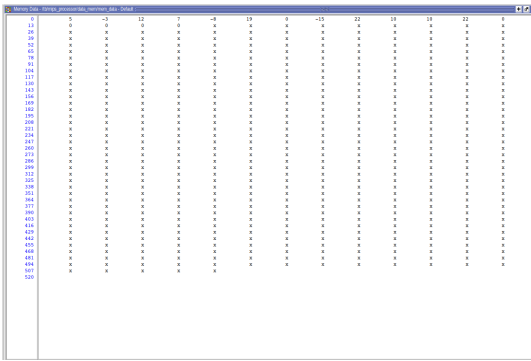
شکل ۳: دستورات برنامه تست به صورت باینری

شکل ۴: داده های برنامه تست به صورت باینری در حافظه داده

۲.۱ نتایج شبیه سازی

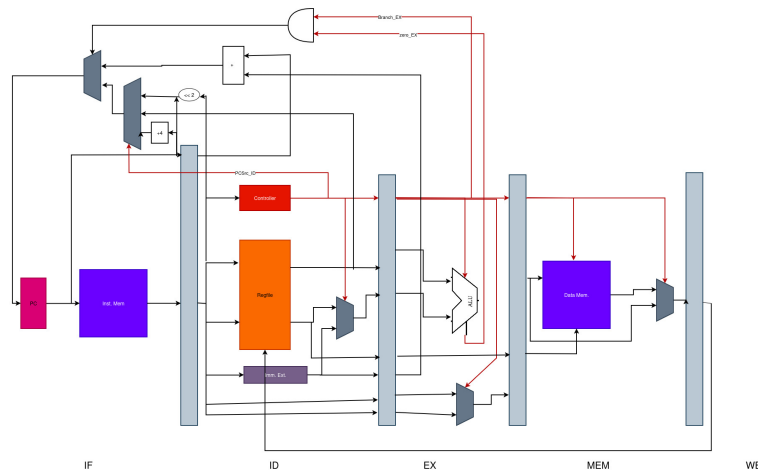


شکل ۵: شبیه سازی single cycle



شکل ۶: حافظه داده در انتهای شبیه سازی single cycle

۲ پیاده سازی پایپ لاین



شکل ۷: شماتیک مسیر داده پایپ لاین

۱.۲ برنامه تست پایپ لاین

برنامه تست پایپ لاین همان برنامه تست single cycle است با این تفاوت که بین تمام دو دستور، سه دستور NOP قرار داده شده است که از مخاطرات جلوگیری شود که در نتیجه این تغییر باید دستورات Branch و Jump تغییر کنند و درست شوند.

```

1  00100000000010000010000000000000
2  000000000000000000000000000000
3  000000000000000000000000000000
4  000000000000000000000000000000
5  100011010000100100000000000000
6  000000000000000000000000000000
7  000000000000000000000000000000
8  000000000000000000000000000000
9  0010010000001010000000000000001
10 000000000000000000000000000000
11 000000000000000000000000000000
12 000000000000000000000000000000
13 10001100000010110010000000101000
14 000000000000000000000000000000
15 000000000000000000000000000000
16 000000000000000000000000000000
17 0010000000000001000000000000001
18 000000000000000000000000000000
19 000000000000000000000000000000
20 000000000000000000000000000000
21 00000001011000010101100000100010
22 000000000000000000000000000000
23 000000000000000000000000000000
24 000000000000000000000000000000
25 00010001011000000000000000100011
26 000000000000000000000000000000
27 000000000000000000000000000000
28 000000000000000000000000000000
29 100011010000110000000000000000
30 000000000000000000000000000000
31 000000000000000000000000000000
32 000000000000000000000000000000

```

شکل ۸: حافظه دستورات در پایپ لاین

علاوه بر تست فوق، تست دیگری مبنی بر بررسی تمام دستورات زده شده انجام شده است.

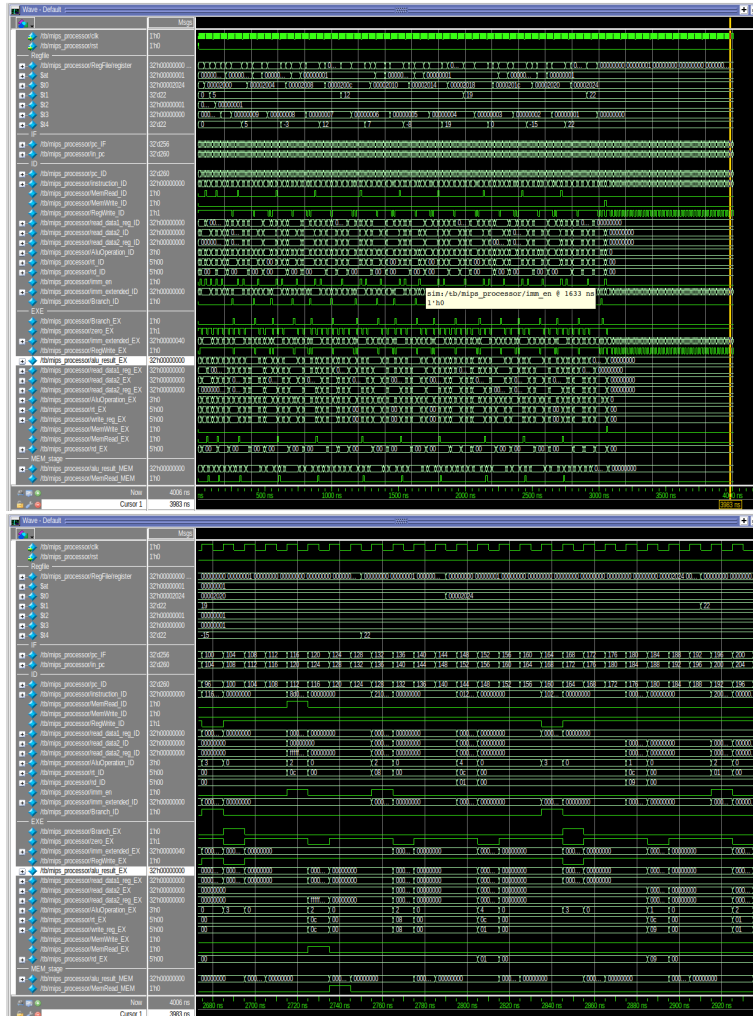

```

1 .data
2 array: .word 10, 20, 30, 40, 50 # An array of integers
3 result: .word 0 # Variable to store the final result
4 .text
5 .globl main
6 main:
7     # Arithmetic operations
8     li $t0, 10 # Load immediate value into $t0
9     nop
10    nop
11    nop
12    li $t1, 20 # Load immediate value into $t1
13    nop
14    nop
15    nop
16    add $t2, $t0, $t1 # $t2 = $t0 + $t1
17    nop
18    nop
19    nop
20    addu $t3, $t0, $t1 # $t3 = $t0 + $t1 (unsigned)
21    nop
22    nop
23    nop
24    sub $t4, $t1, $t0 # $t4 = $t1 - $t0
25    nop
26    nop
27    nop
28    subu $t5, $t1, $t0 # $t5 = $t1 - $t0 (unsigned)
29    nop
30    nop
31    nop
32
33    # Comparison operations
34    slt $t6, $t0, $t1 # $t6 = 1 if $t0 < $t1, else 0
35    nop
36    nop
37    nop
38    sltu $t7, $t0, $t1 # $t7 = 1 if $t0 < $t1 (unsigned), else 0
39    nop
40    nop
41    nop
42
43    # Logical operations
44    and $t8, $t0, $t1 # $t8 = $t0 & $t1
45    nop
46    nop
47    nop
48    or $t9, $t0, $t1 # $t9 = $t0 | $t1
49    nop
50    nop
51    nop
52    xor $t0, $t0, $t1 # $t0 = $t0 ^ $t1
53    nop
54    nop
55    nop
56    nor $t1, $t0, $t1 # $t1 = ~( $t0 | $t1 )
57    nop
58    nop
59    nop
60
61    # Shift operations
62    sll $t2, $t2, 2 # $t2 = $t2 << 2
63    nop
64    nop
65    nop
66    srl $t3, $t3, 1 # $t3 = $t3 >> 1 (logical)
67    nop
68    nop
69    nop
70    sllv $t4, $t2, $t1 # $t4 = $t2 << $t1 (variable shift)
71    nop
72    nop
73    nop
74
75    # Arithmetic operations with immediates
76    addi $t5, $t5, 10 # $t5 = $t5 + 10
77    nop
78    nop
79    nop
80    addiu $t6, $t6, 5 # $t6 = $t6 + 5 (unsigned)
81    nop
82    nop
83    nop
84    slti $t7, $t0, 30 # $t7 = 1 if $t0 < 30, else 0
85    nop
86    nop
87    nop
88    # ... (more instructions) ...

```

شکل ۹: تست دوم پایپ لاین

۲.۲ نتایج شبیه سازی

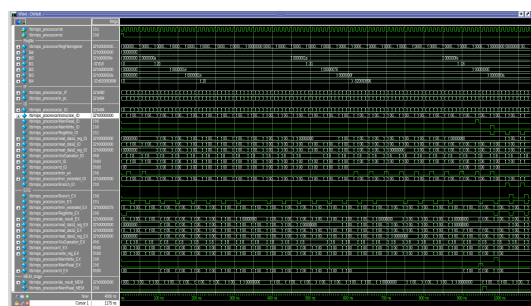


شکل ۱۰: شبیه سازی پایپ لاین

Menu Data - Contas pessoais - Nominis das Contas													
0	5	+5	12	7	+6	13	9	+10	22	18	19	22	9
13	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0	0	0	0	0	0
130	0	0	0	0	0	0	0	0	0	0	0	0	0
140	0	0	0	0	0	0	0	0	0	0	0	0	0
150	0	0	0	0	0	0	0	0	0	0	0	0	0
160	0	0	0	0	0	0	0	0	0	0	0	0	0
170	0	0	0	0	0	0	0	0	0	0	0	0	0
180	0	0	0	0	0	0	0	0	0	0	0	0	0
190	0	0	0	0	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0	0	0	0	0
210	0	0	0	0	0	0	0	0	0	0	0	0	0
220	0	0	0	0	0	0	0	0	0	0	0	0	0
230	0	0	0	0	0	0	0	0	0	0	0	0	0
240	0	0	0	0	0	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	0	0	0	0	0	0	0
260	0	0	0	0	0	0	0	0	0	0	0	0	0
270	0	0	0	0	0	0	0	0	0	0	0	0	0
280	0	0	0	0	0	0	0	0	0	0	0	0	0
290	0	0	0	0	0	0	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0	0	0	0	0	0	0
310	0	0	0	0	0	0	0	0	0	0	0	0	0
320	0	0	0	0	0	0	0	0	0	0	0	0	0
330	0	0	0	0	0	0	0	0	0	0	0	0	0
340	0	0	0	0	0	0	0	0	0	0	0	0	0
350	0	0	0	0	0	0	0	0	0	0	0	0	0
360	0	0	0	0	0	0	0	0	0	0	0	0	0
370	0	0	0	0	0	0	0	0	0	0	0	0	0
380	0	0	0	0	0	0	0	0	0	0	0	0	0
390	0	0	0	0	0	0	0	0	0	0	0	0	0
400	0	0	0	0	0	0	0	0	0	0	0	0	0
410	0	0	0	0	0	0	0	0	0	0	0	0	0
420	0	0	0	0	0	0	0	0	0	0	0	0	0
430	0	0	0	0	0	0	0	0	0	0	0	0	0
440	0	0	0	0	0	0	0	0	0	0	0	0	0
450	0	0	0	0	0	0	0	0	0	0	0	0	0
460	0	0	0	0	0	0	0	0	0	0	0	0	0
470	0	0	0	0	0	0	0	0	0	0	0	0	0
480	0	0	0	0	0	0	0	0	0	0	0	0	0
490	0	0	0	0	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0	0	0	0	0

شکل ۱۱: حافظه داده در انتهای شبیه سازی پایپ لاین

نتایج شبیه سازی تست دوم که تست تمام دستورات است به شرح زیر است:



شکل ۱۲: نتایج شبیه سازی تست دوم پایپ لاین