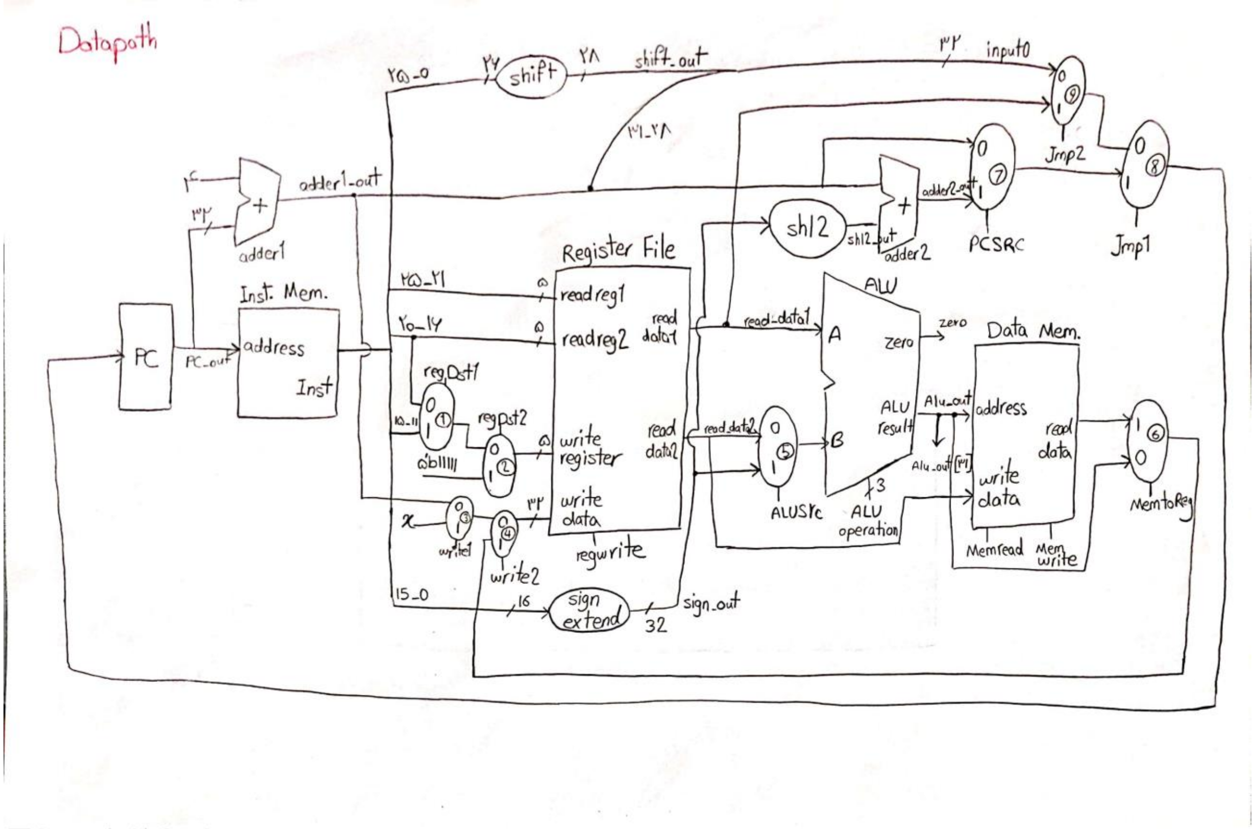
مریم جعفرآبادی آشتیانی 810199549

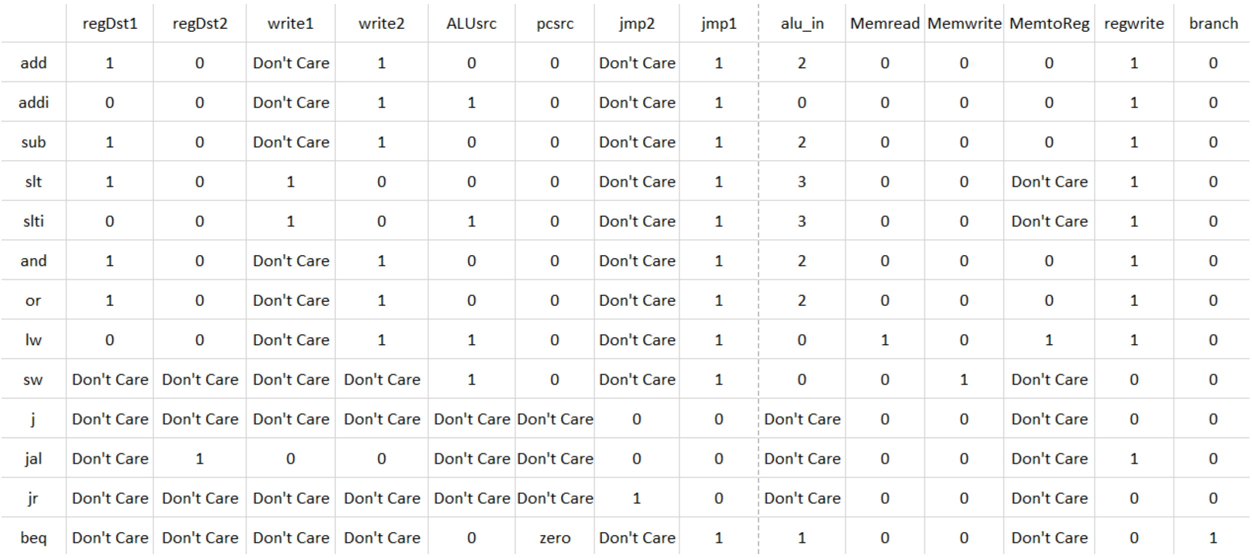
سنا ساری نوایی 810199435

تمرین کامپیوتری دوم معماری کامپیوتر

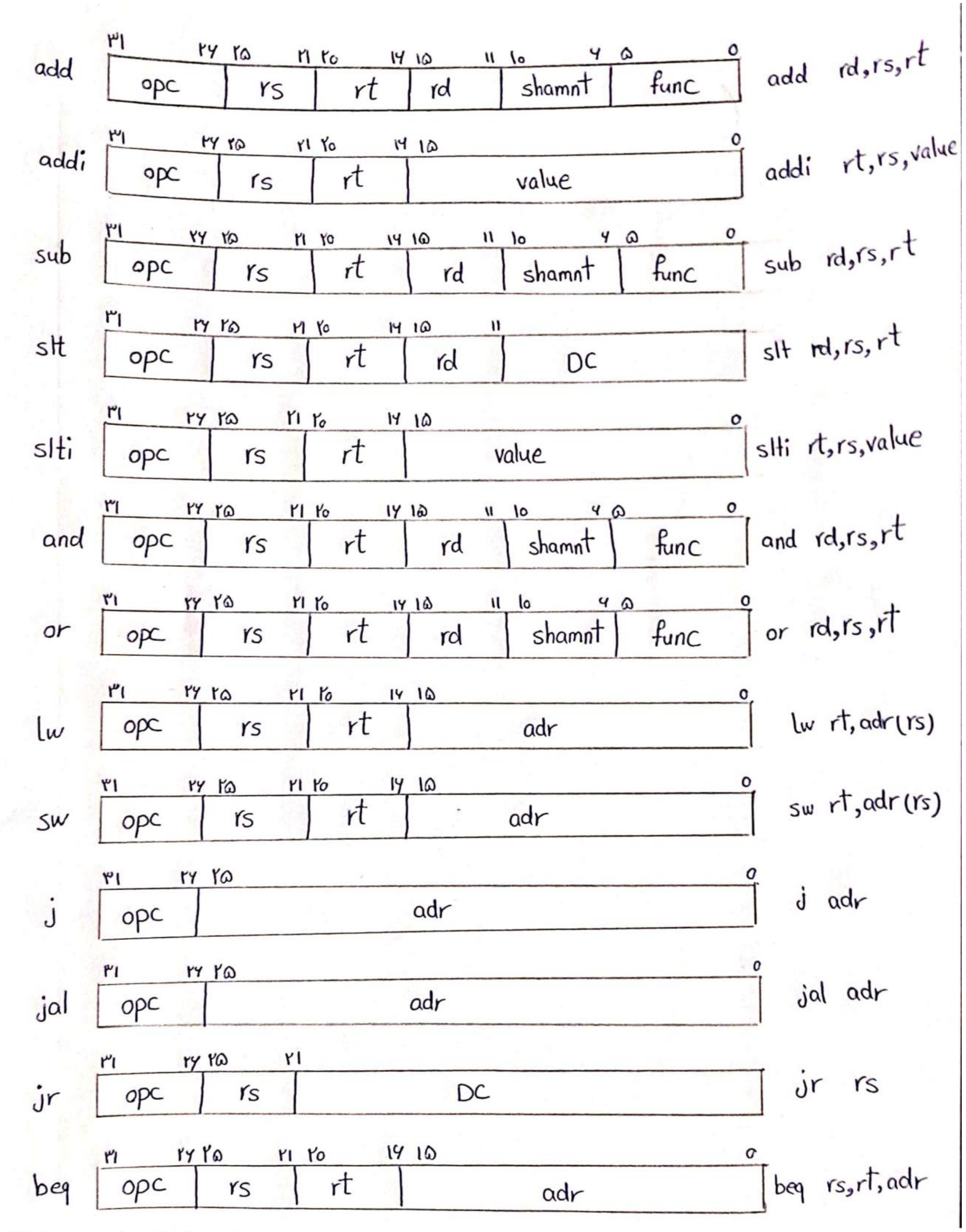
Datapath:



Controller:



	regDst1	regDst2	write1	write2	ALUsrc	pcsrc	jmp2	jmp1	alu_in	Memread	Memwrite	MemtoReg	regwrite	branch
add	1	0	Don't Care	1	0	0	Don't Care	1	2	0	0	0	1	0
addi	0	0	Don't Care	1	1	0	Don't Care	1	0	0	0	0	1	0
sub	1	0	Don't Care	1	0	0	Don't Care	1	2	0	0	0	1	0
slt	1	0	1	0	0	0	Don't Care	1	3	0	0	Don't Care	1	0
slti	0	0	1	0	1	0	Don't Care	1	3	0	0	Don't Care	1	0
and	1	0	Don't Care	1	0	0	Don't Care	1	2	0	0	0	1	0
or	1	0	Don't Care	1	0	0	Don't Care	1	2	0	0	0	1	0
lw	0	0	Don't Care	1	1	0	Don't Care	1	0	1	0	1	1	0
sw	Don't Care	Don't Care	Don't Care	Don't Care	1	0	Don't Care	1	0	0	1	Don't Care	0	0
j	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	0	Don't Care	0	0	Don't Care	0	0
jal	Don't Care	1	0	0	Don't Care	Don't Care	0	0	Don't Care	0	0	Don't Care	1	0
jr	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	1	0	Don't Care	0	0	Don't Care	0	0
beq	Don't Care	Don't Care	Don't Care	Don't Care	0	zero	Don't Care	1	1	0	0	Don't Care	0	1



add

۳۱ ۲۶	۲۵ ۲۱	۲۰ ۱۶	۱۵ ۱۱	۱۰ ۶	۵ ۰
opc	rs	rt	rd	shamnt	func

add rd,rs,rt

addi

۳۱ ۲۶	۲۵ ۲۱	۲۰ ۱۶	۱۵ ۰
opc	rs	rt	value

addi rt,rs,value

sub

۳۱ ۲۶	۲۵ ۲۱	۲۰ ۱۶	۱۵ ۱۱	۱۰ ۶	۵ ۰
opc	rs	rt	rd	shamnt	func

sub rd,rs,rt

slt

۳۱ ۲۶	۲۵ ۲۱	۲۰ ۱۶	۱۵ ۱۱	
opc	rs	rt	rd	DC

slt rd,rs,rt

slti

۳۱ ۲۶	۲۵ ۲۱	۲۰ ۱۶	۱۵ ۰
opc	rs	rt	value

slti rt,rs,value

and

۳۱ ۲۶	۲۵ ۲۱	۲۰ ۱۶	۱۵ ۱۱	۱۰ ۶	۵ ۰
opc	rs	rt	rd	shamnt	func

and rd,rs,rt

or

۳۱ ۲۶	۲۵ ۲۱	۲۰ ۱۶	۱۵ ۱۱	۱۰ ۶	۵ ۰
opc	rs	rt	rd	shamnt	func

or rd,rs,rt

lw

۳۱ ۲۶	۲۵ ۲۱	۲۰ ۱۶	۱۵ ۰
opc	rs	rt	adr

lw rt,adr(rs)

sw

۳۱ ۲۶	۲۵ ۲۱	۲۰ ۱۶	۱۵ ۰
opc	rs	rt	adr

sw rt,adr(rs)

j

۳۱ ۲۶	۲۵ ۰
opc	adr

j adr

jal

۳۱ ۲۶	۲۵ ۰
opc	adr

jal adr

jr

۳۱ ۲۶	۲۵ ۲۱	
opc	rs	DC

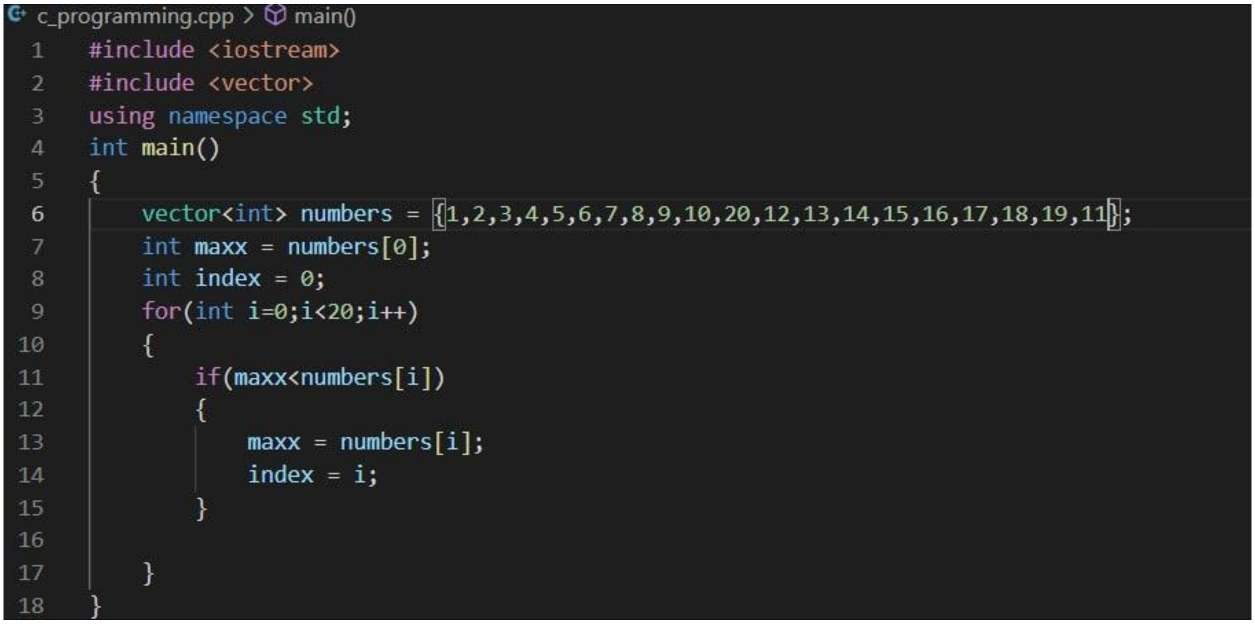
jr rs

beq

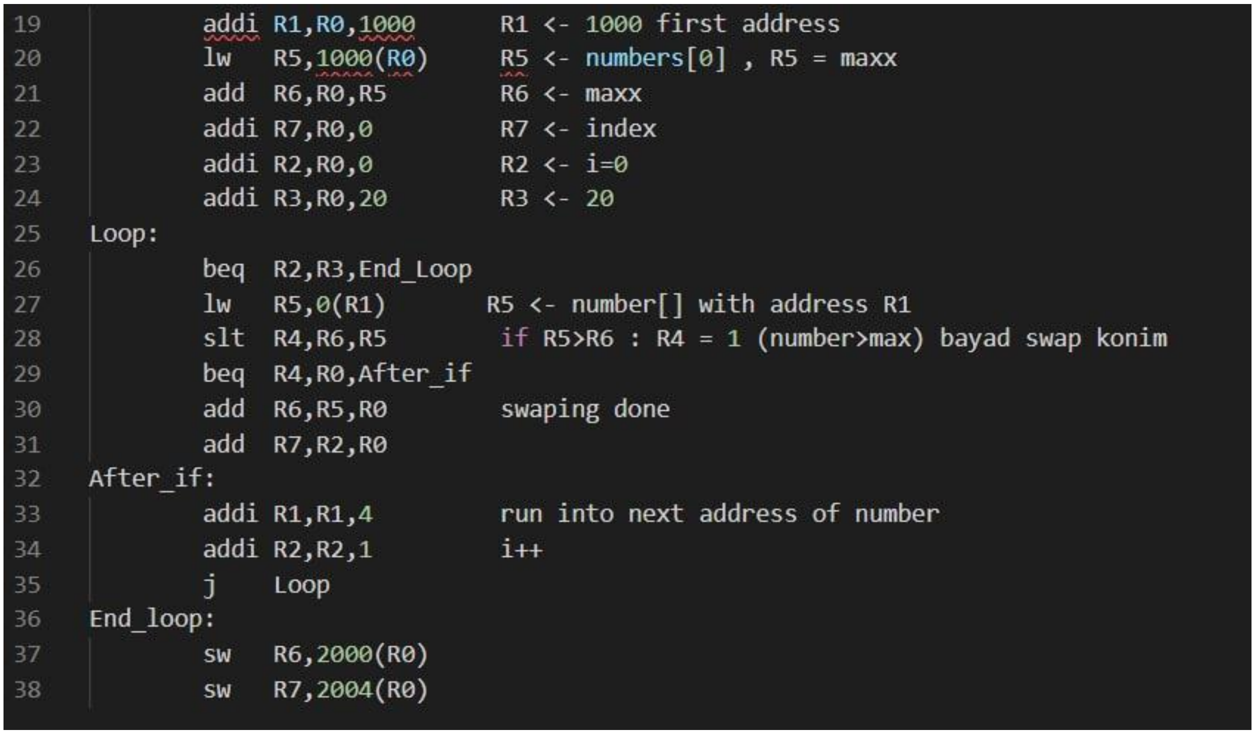
۳۱ ۲۶	۲۵ ۲۱	۲۰ ۱۶	۱۵ ۰
opc	rs	rt	adr

beq rs,rt,adr

C++ code:



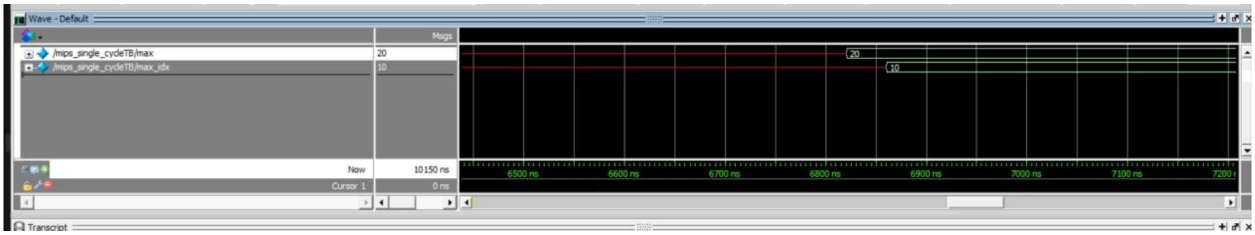
```cpp
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    vector<int> numbers = {1,2,3,4,5,6,7,8,9,10,20,12,13,14,15,16,17,18,19,11};
    int maxx = numbers[0];
    int index = 0;
    for(int i=0;i<20;i++)
    {
        if(maxx<numbers[i])
        {
            maxx = numbers[i];
            index = i;
        }

    }
}
```



```
        addi R1,R0,1000      R1 <- 1000 first address
        lw   R5,1000(R0)     R5 <- numbers[0] , R5 = maxx
        add  R6,R0,R5        R6 <- maxx
        addi R7,R0,0         R7 <- index
        addi R2,R0,0         R2 <- i=0
        addi R3,R0,20        R3 <- 20
Loop:
        beq  R2,R3,End_Loop
        lw   R5,0(R1)       R5 <- number[] with address R1
        slt  R4,R6,R5        if R5>R6 : R4 = 1 (number>max) bayad swap konim
        beq  R4,R0,After_if
        add  R6,R5,R0        swaping done
        add  R7,R2,R0
After_if:
        addi R1,R1,4         run into next address of number
        addi R2,R2,1         i++
        j    Loop
End_loop:
        sw   R6,2000(R0)
        sw   R7,2004(R0)
```

ورودی های داده شده در قالب یک آرایه 20 تایی به شرح زیر می‌باشد:

data	1	2	3	4	5	6	7	8	9	10	20	11	12	13	14	15	16	17	18	19
index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19


/mips_single_cycleTB/max 20
/mips_single_cycleTB/max_idx 10
Now 10150 ns
Cursor 1 0 ns

همانطور که در شکل مشخص است، بزرگترین عدد 20 و index آن نیز مقدار 10 را نمایش می‌دهد.