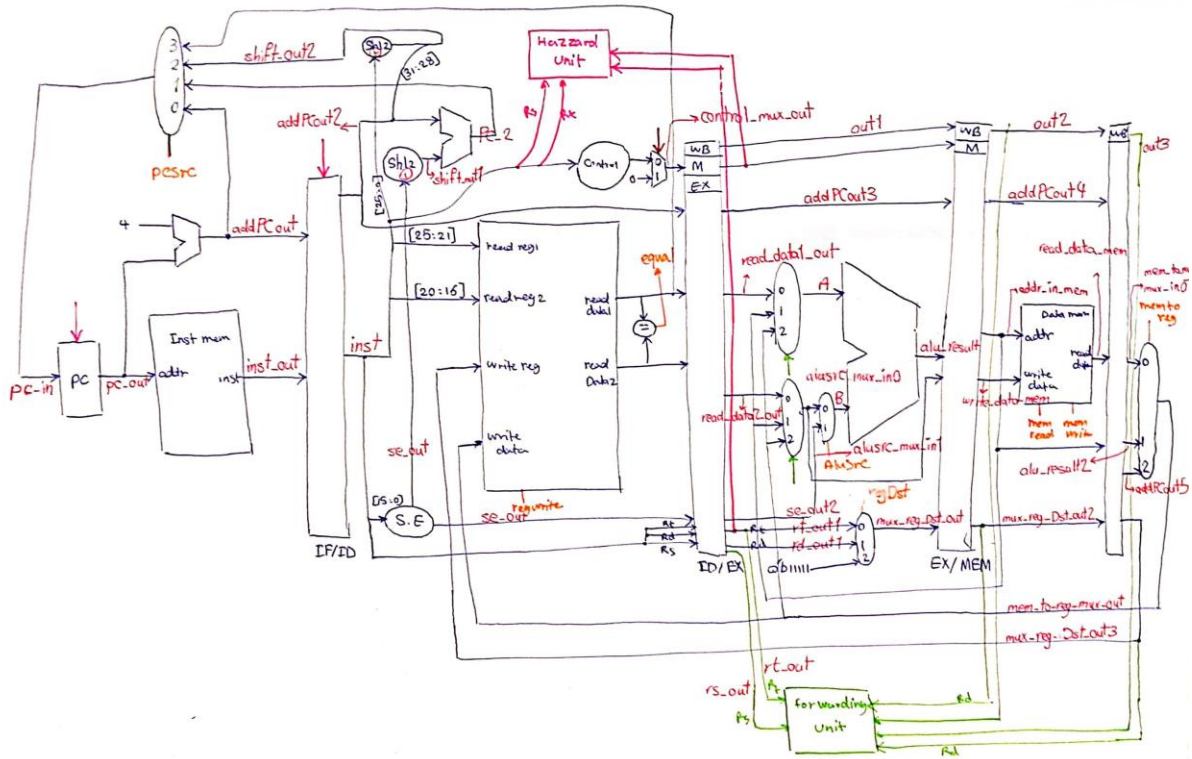


Datapath:



Controller:

سیگنال های کنترلی در controller به شرح زیر می باشند و در هر مرحله مشخص است که کدام سیگنال ها باید چه مقداری داشته باشند.

```
case (opcode)
  6'b000000 : {reg_dst, mem_to_reg, reg_write, alu_src, mem_read, mem_write, psrc, aluop, Flush}
              = {2'b01, 2'b00, 1'b1, 1'b0, 1'b0, 1'b0, 2'b00, 2'b10, 1'b0};
  6'b100011 : {reg_dst, mem_to_reg, reg_write, alu_src, mem_read, mem_write, psrc, aluop, Flush}
              = {2'b00, 2'b01, 1'b1, 1'b1, 1'b1, 1'b0, 2'b00, 2'b00, 1'b0};
  6'b101011 : {reg_dst, mem_to_reg, reg_write, alu_src, mem_read, mem_write, psrc, aluop, Flush}
              = {2'b00, 2'b00, 1'b0, 1'b1, 1'b0, 1'b1, 2'b00, 2'b00, 1'b0};
  6'b000100 : {reg_dst, mem_to_reg, reg_write, alu_src, mem_read, mem_write, psrc, aluop, Flush}
              = {2'b00, 2'b00, 1'b0, 1'b0, 1'b0, 1'b0, {1'b0, Equal}, 2'b00, Equal};
  6'b001001 : {reg_dst, mem_to_reg, reg_write, alu_src, mem_read, mem_write, psrc, aluop, Flush}
              = {2'b00, 2'b00, 1'b1, 1'b1, 1'b0, 1'b0, 2'b00, 2'b00, 1'b0};
  6'b000010 : {reg_dst, mem_to_reg, reg_write, alu_src, mem_read, mem_write, psrc, aluop, Flush}
              = {2'b00, 2'b00, 1'b0, 1'b0, 1'b0, 1'b0, 2'b10, 2'b00, 1'b1};
  6'b000011 : {reg_dst, mem_to_reg, reg_write, alu_src, mem_read, mem_write, psrc, aluop, Flush}
              = {2'b10, 2'b10, 1'b0, 1'b0, 1'b0, 1'b0, 2'b10, 2'b00, 1'b0};
  6'b000110 : {reg_dst, mem_to_reg, reg_write, alu_src, mem_read, mem_write, psrc, aluop, Flush}
              = {2'b00, 2'b00, 1'b0, 1'b0, 1'b0, 1'b0, 2'b11, 2'b00, 1'b0};
  6'b001010 : {reg_dst, mem_to_reg, reg_write, alu_src, mem_read, mem_write, psrc, aluop, Flush}
              = {2'b00, 2'b00, 1'b1, 1'b1, 1'b0, 1'b0, 2'b00, 2'b11, 1'b0};
  6'b111111 : psrc = 2'b00;
endcase
```

add

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				rt				rd				shamt				func											

 add rd,rs,rt

addi

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				rt				value																			

 addi rt,rs,value

sub

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				rt				rd				shamt				func											

 sub rd,rs,rt

slt

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				rt				rd				DC															

 slt rd,rs,rt

slti

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				rt				value																			

 slti rt,rs,value

and

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				rt				rd				shamt				func											

 and rd,rs,rt

or

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				rt				rd				shamt				func											

 or rd,rs,rt

lw

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				rt				adr																			

 lw rt,adr(rs)

sw

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				rt				adr																			

 sw rt,adr(rs)

j

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				adr																											

 j adr

jal

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				adr																											

 jal adr

jr

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				DC																							

 jr rs

beq

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc				rs				rt				adr																			

 beq rs,rt,adr

C++ code:

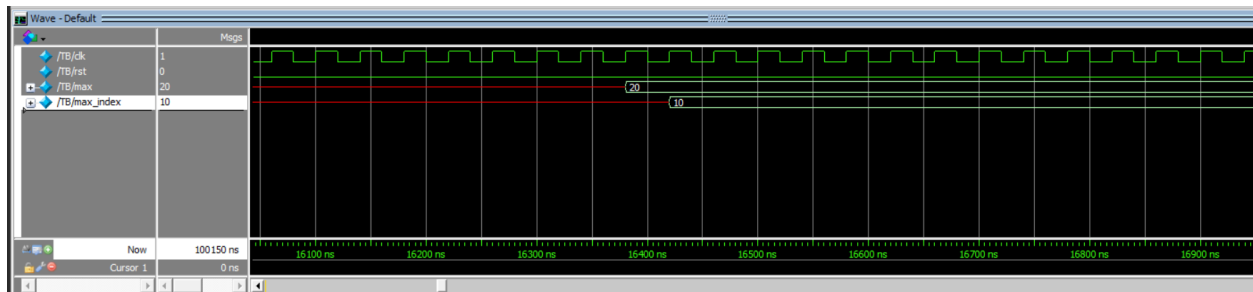
در این کد، 20 داده ورودی در قالب وکتور به برنامه داده شده است و در یک حلقه for ، مقدار ماکسیمم این اعداد و همچنین index آن نیز ذخیره می‌شود.

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

```
19      addi R1,R0,1000      R1 <- 1000 first address
20      lw    R5,1000(R0)    R5 <- numbers[0] , R5 = maxx
21      add   R6,R0,R5       R6 <- maxx
22      addi  R7,R0,0        R7 <- index
23      addi  R2,R0,0        R2 <- i=0
24      addi  R3,R0,20       R3 <- 20
25  Loop:   no operation
26          no operation
27          no operation
28          beq R2,R3,End_Loop
29          lw   R5,0(R1)    R5 <- number[] with address R1
30          slt  R4,R6,R5    if R5>R6 : R4 = 1 (number>max) bayad swap konim
31          no operation
32          no operation
33          no operation
34          beq  R4,R0,After_if
35          add  R6,R5,R0    swaping done
36          add  R7,R2,R0
37  After_if:
38          addi R1,R1,4      run into next address of number
39          addi R2,R2,1      i++
40          j    Loop
41  End_loop:
42          sw   R6,2000(R0)
43          sw   R7,2004(R0)
```

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

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



همانطور که در شکل مشخص است، بزرگترین عدد 20 و index آن نیز 10 می باشد.