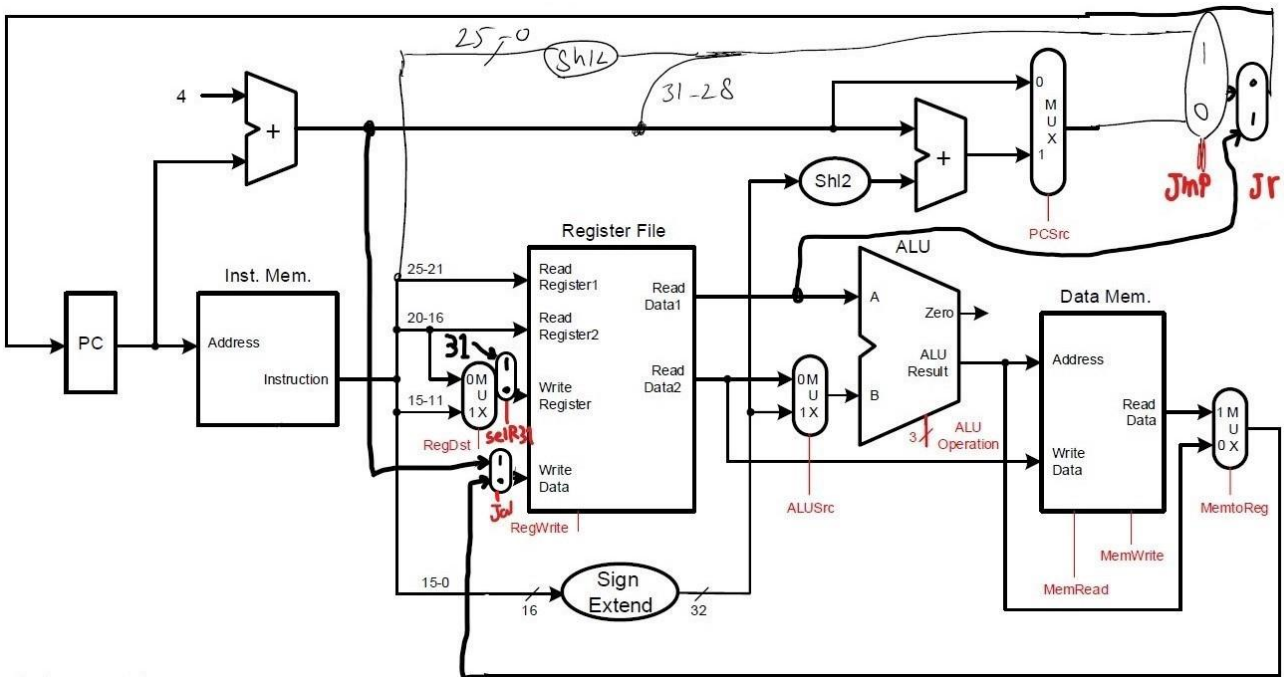
CA2

علی هدائی 810199513

علی عطاللهی 810199461

Data path:



Controller:

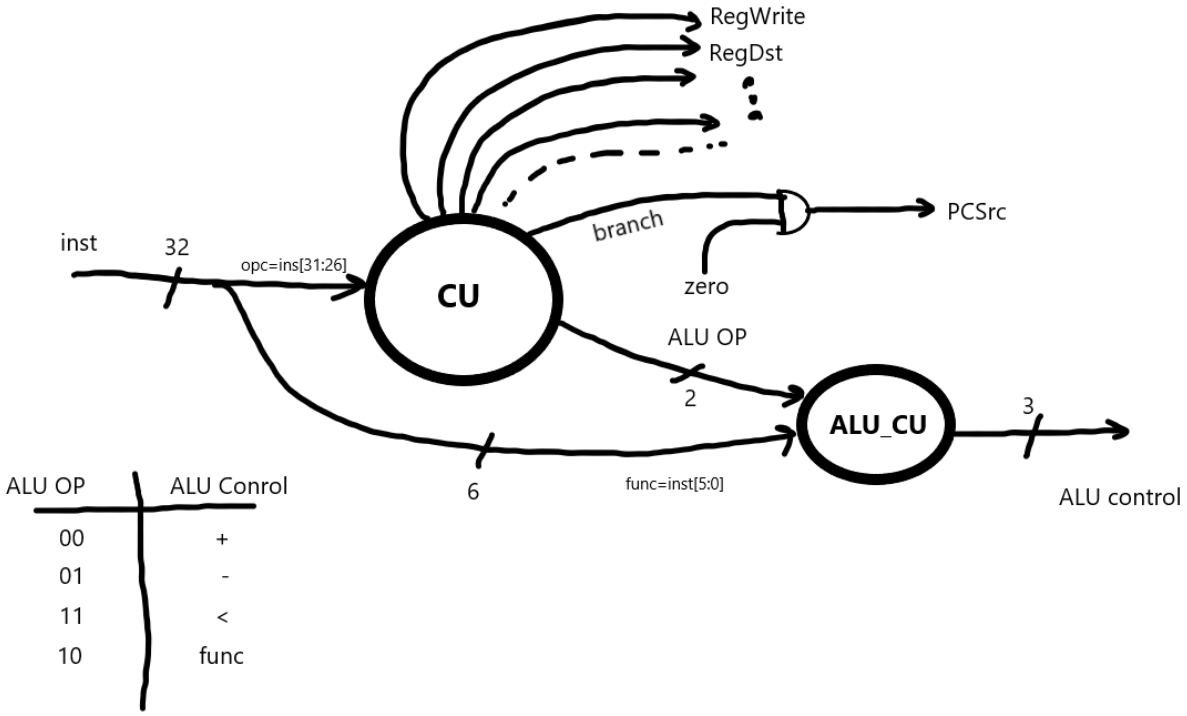
Opcode:

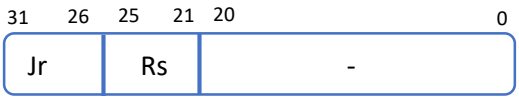
```
`define R_type 6'b000000
`define Addi 6'b001000
`define Slti 6'b001010
`define Lw 6'b100011
`define Sw 6'b101011
`define J 6'b000010
`define Jal 6'b000011
`define Jr 6'b000111
`define Beq 6'b000100
```

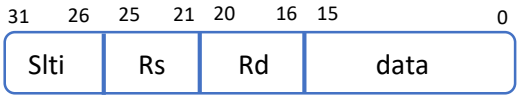
Function:

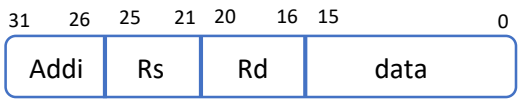
```
`define Add 6'b100000
`define Sub 6'b100010
`define And 6'b100100
`define Or 6'b100101
`define Slt 6'b101010
```

	RegDst	RegWrite	ALUSrc	mem_read	mem_write	MemToReg	branch	Jmp	PCSrc	Jr	selR31	Jal	ALU op
RT	1	1	0	0	0	0	0	0	0	0	0	0	10
Addi	0	1	1	0	0	0	0	0	0	0	0	0	00
Slti	0	1	1	0	0	0	0	0	0	0	0	0	11
Lw	0	1	1	1	0	1	0	0	0	0	0	0	00
Sw	-	0	1	0	1	-	0	0	0	0	-	-	00
J	-	0	-	0	0	-	0	1	0	0	-	-	-
Jal	-	1	-	0	0	-	0	1	0	0	1	1	-
Jr	-	0	-	0	0	-	0	-	0	1	-	-	-
Beq	-	0	0	0	0	-	1	0	Zero & branch	0	-	-	01









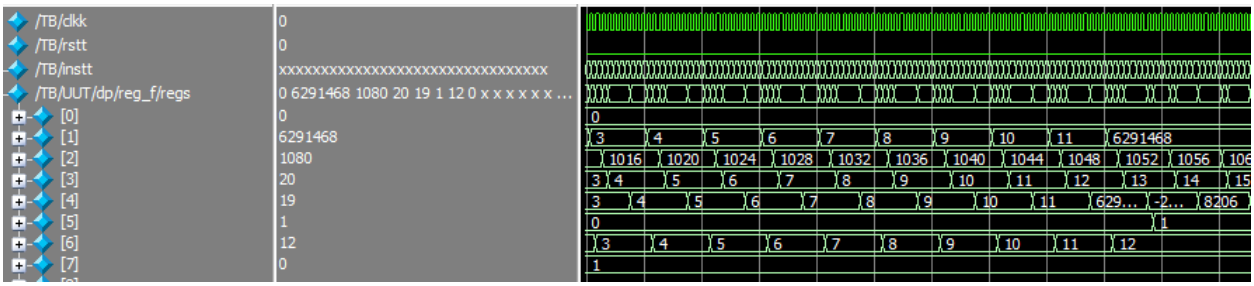
Assembly code for testing:

```
int A[20] = {0,1,2,3,4,5,6,7,8,9,10,11,6291468,-2147483635,8206,15,16,17,18,19};
int Max_index = 0; // R6
int Max_value = 0; // R1
int data_address = 1000; // R2
for (int i = 0; i < 20; i++) { // i = R3          i < 20 = R7
    int temp = A[i]; // temp = R4
    if (temp >= Max_value) { // R5
        Max_value = temp;
        Max_index = i;
    }
    data_address += 4;
}
```

```
 1   000000_00000_00000_00110_00000100000 // add R6,R0,R0
 2   000000_00000_00000_00001_00000100000 // add R1,R0,R0
 3   001000_00000_00010__0000001111101000 // addi R2,R0,1000
 4   000000_00000_00000_00011_00000100000 // add R3,R0,R0
 5   001010_00011_00111__0000000000010100 // Slti R7,R3,20           LOOP
 6   000100_00000_00111__0000000000001001 // beq R7,R0,END_LOOP
 7   100011_00010_00100__0000000000000000 // lw R4,0(R2)
 8   000000_00100_00001_00101_00000101010 // slt R5,R4,R1
 9   000100_00000_00101__0000000000000010 // beq R5,R0,IF
10   000010____00000000000000000000001100 // J END_IF
11   000000_00000_00101_00001_00000100000 // add R1,R0,R4            IF
12   000000_00000_00011_00110_00000100000 // add R6,R0,R3
13   001000_00010_00010__0000000000000100 // addi R2,R2,4            END_IF
14   001000_00011_00011__0000000000000001 // addi R3,R3,+1
15   000010____00000000000000000000000100 // J LOOP
16   101011_00000_00001__0000011111010000 // Sw R1,2000(R0)          END_LOOP
17   101011_00000_00110__0000011111010100 // Sw R6,2004(R0)
```

Wave forms:

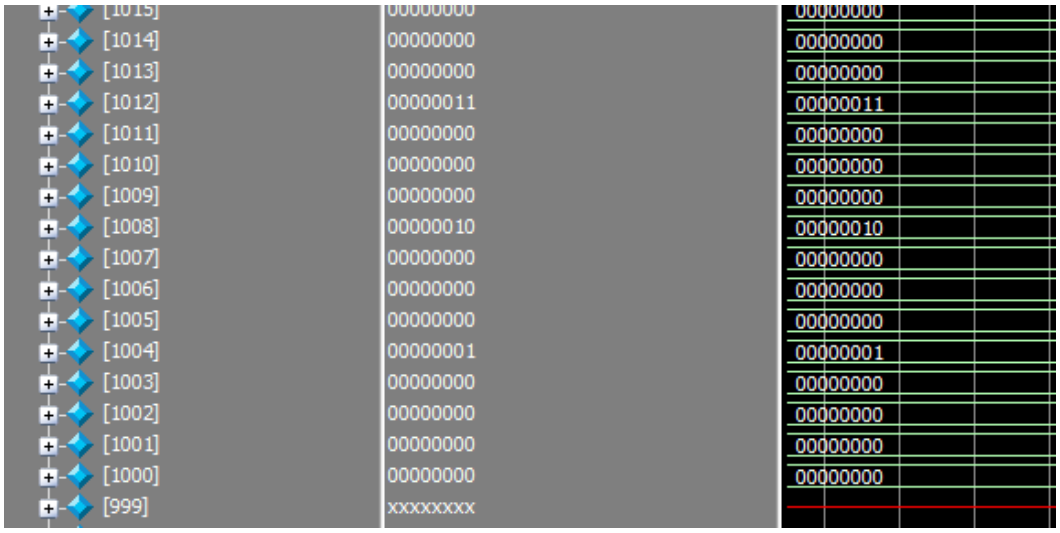
Registers:



همانطور که میبینیم مقادیر رجیسترها هر بار به درستی آپدیت میشوند.

Memory:

آدرس ها در مموری پردازنده میپس به صورت مضرب ۴ است. پس برای درست کردن این مسئله و همچنین دسترسی به بایت ها مموری را ۸بیت در نظر گرفتیم به طوری که هر ۴ تا ردیف با هم یک ورد ۳۲ بیتی را میسازند.



همانطور که مشاهده میشود پس از پایان دستورات، در آدرس ۲۰۰۰ و ۲۰۰۴ مموری مقادیر ماکسیمم عدد و ایندکسش به درستی ذخیره شده است. و همچنین مقادیر آرایه از خانه ۱۰۰۰ به بعد به درستی نوشته شده اند.