

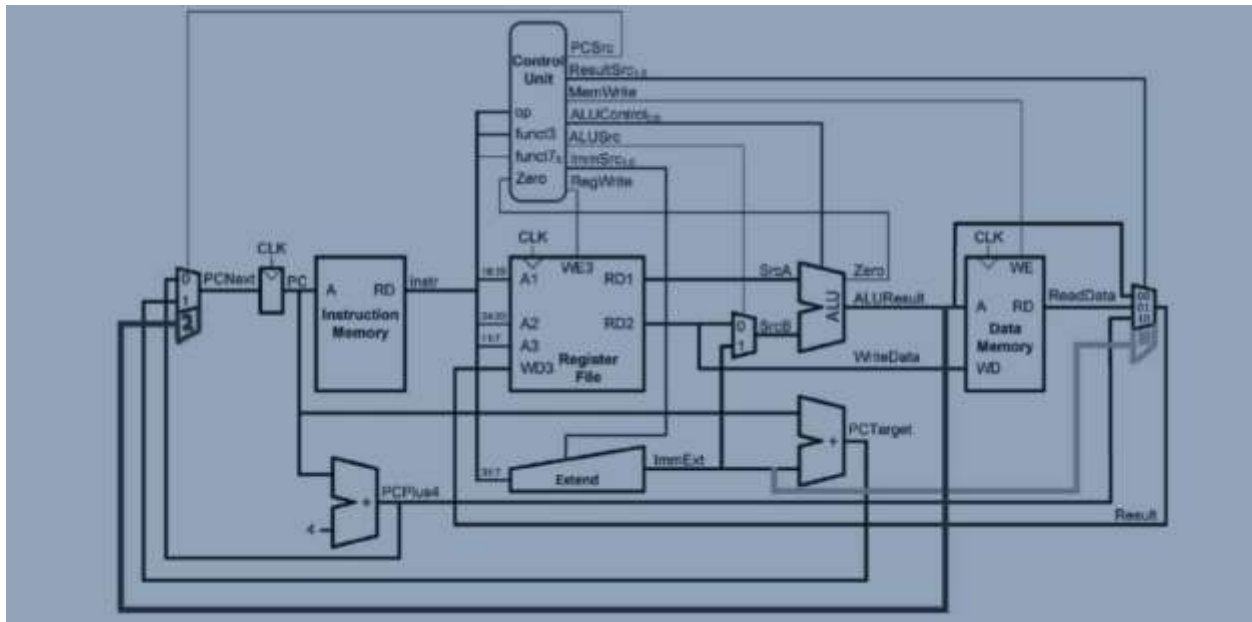
C code for finding minimum:

```
1  int A[10] = {1, 2, 42, 32, 12, 25, 43, 7, -45, 8};
2  int min = A[0];
3  for (int i = 1; i < 10; i++)
4  {
5      if (A[i] < min)
6      {
7          min = A[i];
8      }
9  }
```

Assembly code of This function:

```
1  addi x8, x8, 0x000      -> 000000000000001000100000010010011
2  addi x5, x0, 13        -> 00000000011000000000001010010011
3  sw x5, 0(x8)           -> 000000001010100000000010000100011
4  addi x5, x0, -7        -> 11111111100100000000001010010011
5  sw x5, 4(x8)           -> 000000001010100000001010000100011
6  addi x5, x0, 20        -> 00000000101000000000001010010011
7  sw x5, 8(x8)           -> 000000001010100000010010000100011
8  addi x5, x0, -20       -> 11111111101000000000001010010011
9  sw x5, 12(x8)          -> 000000001010100000011010000100011
10 addi x5, x0, 33         -> 00000000100000000000001010010011
11 sw x5, 16(x8)          -> 000000001010100000100010000100011
12
13 addi x5, x0, 0          -> 00000000000000000000001010010011
14 sw x5, 20(x8)          -> 000000001010100000101010000100011
15 addi x5, x0, -11       -> 11111111110100000000001010010011
16 sw x5, 24(x8)          -> 000000001010100000110010000100011
17 addi x5, x0, 92        -> 00000001011100000000001010010011
18 sw x5, 28(x8)          -> 000000001010100000111010000100011
19 addi x5, x0, -58       -> 11111111011100000000001010010011
20 sw x5, 32(x8)          -> 000000001010100001000010000100011
21 addi x5, x0, 19        -> 00000000100110000000001010010011
22 sw x5, 36(x8)          -> 000000001010100001001010000100011
23 addi x6, x0, 1         -> 000000000000100000000001100010011
24 lw x9, 0(x8)           -> 000000000000001000010010010000011
25 addi x18, x18, 0       -> 000000000000010010000010000010011
26
27 loop:
28 beq x18, x5, end_loop   -> 000000000010110010000000001100011
29 add x19, x18, x8        -> 000000000010010001001100110110011
30 lw x20, 0(x19)          -> 000000000000010011010010100000011
31 slt x21, x20, x9        -> 000000010010010010101010110110011
32 beq x21, x6, end_if     -> 000000000011010101000000101100011
33
34 end_if:
35 addi x18, x18, 4         -> 000000000010010010000010000010011
36 jal loop                -> 111111111111111111111111111111111|
37 end_loop:
```

Datapath:



Controller:

Type	Instruction	Pc src	Result src	Mem write	Alu cnt	Alu src	Imm src	Reg write	Alu op	Branch	Jump
R-type	Add	00	00	0	000	0	x	1	0	0	00
	Sub	00	00	0	001	0	x	1	10	0	00
	And	00	00	0	010	0	X	1	10	0	00
	Or	00	00	0	011	0	X	1	10	0	00
	Slt	00	00	0	100	0	X	1	10	0	00
I-type	Lw	00	01	0	000	1	000	1	00	0	00
	Addi	00	00	0	001	1	000	1	11	0	00
	Xori	00	00	0	110	1	000	1	11	0	00
	Ori	00	00	0	011	1	000	1	11	0	00
	Slti	00	00	0	100	1	000	1	11	0	00
	Jalr	10	11	0	000	1	000	1	00	0	01
B-type	Beq	01:00	X	0	001	0	011	0	01	1	00
	Bne	00:01	X	0	001	0	011	0	01	1	00
U-type	Lui	00	01	0	X	X	100	1	X	0	00
J-type	Jal	00	01	0	X	X	100	1	X	0	10
S-type	Sw	00	X	1	000	1	001	0	00	0	00

Output:

