

```

1    #include <iostream>
2    using namespace std;
3
4    ✓ int main(void) {
5        int arr[10] = {9, 6, 3, 8, 5, 2, 7, 4, 1, 0};
6        int n = 10;
7
8        for(int i = 0; i < n; i++) {
9            for(int j = 0; j < n-i-1; j++) {
10                if(arr[j] > arr[j+1]) {
11                    swap(arr[j], arr[j+1]);
12                }
13            }
14        }
15    }

```

Figure 1: Bubble Sort in C++

```

1  .data
2  array: .word 9, 6, 3, 2, 5, 8, 7, 4, 1, 0
3  size:  .word 10
4  .text
5  main:  la t0, array
6         lw t1, size
7
8  out:   addi t2, x0, 1
9         addi t3, x0, 0
10
11  inner: slli t4, t3, 2
12         add t4, t4, t0
13
14         lw t5, 0(t4)
15         lw t6, 4(t4)
16
17         ble t5, t6, no_swap
18
19         sw t6, 0(t4)
20         sw t5, 4(t4)
21         addi t2, x0, 0
22
23  no_swap: addi t3, t3, 1
24
25         bne t3, t1, inner
26         beq t2, x0, out
27
28         addi a0, x0, 22
29         add x0, x0, x0
30
31
32
33

```

Line: 33 Column: 2 ☒ Show Line Numbers

Figure 2: Bubble Sort in RISC-V



Figure 3: Expected Result

0	00	00	00	00
4	01	00	00	00
8	02	00	00	00
12	03	00	00	00
16	04	00	00	00
20	05	00	00	00
24	06	00	00	00
28	07	00	00	00
32	08	00	00	00
36	09	00	00	00

Figure 4: Actual Result

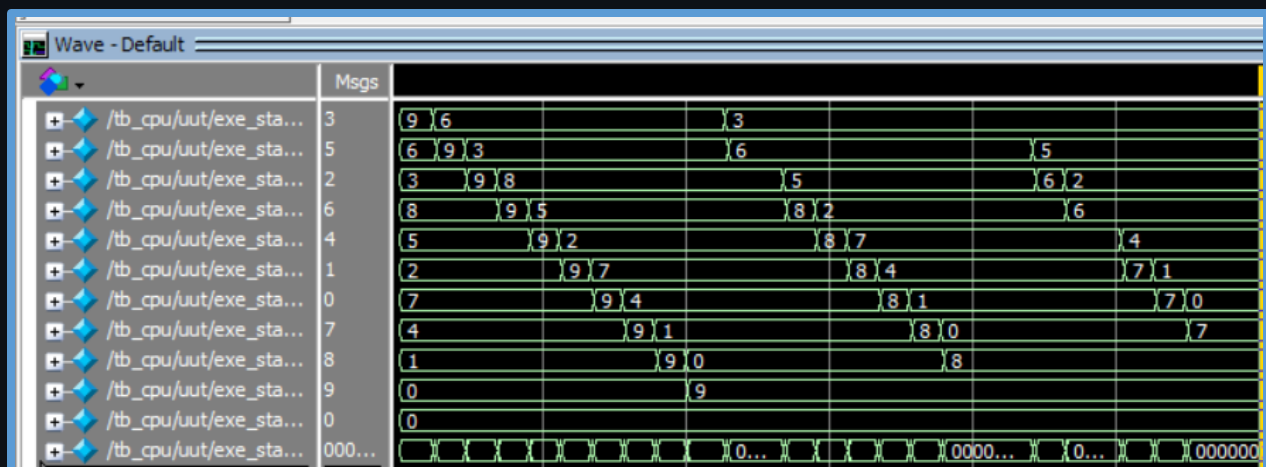


Figure 5: Data Memory [0ns, 3ns]

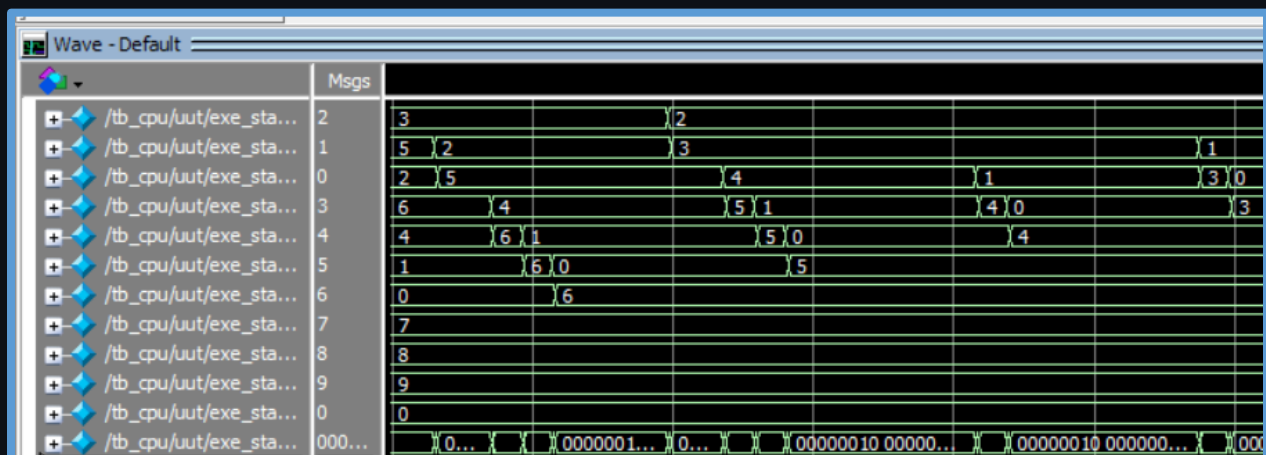


Figure 6: Data Memory [3ns, 6ns]

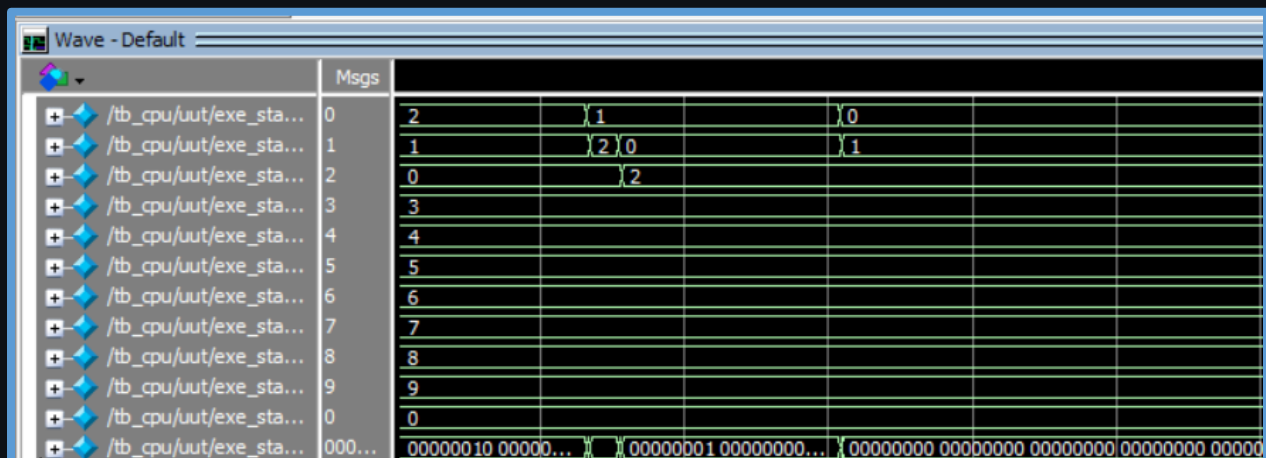


Figure 7: Data Memory [6ns, 9ns]