```
module SC_SM_TB();

parameter N = 32;
parameter M = 8;

wire [(N-1):0] output_arguments_tb;
wire start_target_current_state_machine_tb;

wire  finish_a_tb;
wire  finish_b_tb;
wire  reset_start_request_a_tb;
wire  reset_start_request_b_tb;
wire  [(M-1):0] received_data_a_tb;
wire  [(M-1):0] received_data_b_tb;

reg sm_clk_tb;

reg reset_tb;
reg target_current_state_machine_finished_tb;
reg  start_request_a_tb;
reg  start_request_b_tb;

reg [(N-1):0] input_argumnets_a_tb;
reg [(N-1):0] input_argumnets_b_tb;
reg [M-1:0] in_received_data_tb;

assign input_argumnets_a_tb = 5;
assign input_argumnets_b_tb = 3;
assign in_received_data_tb = 7;


shared_access_to_one_current_state_machine  smtb(

.output_arguments(output_arguments_tb),
.start_target_current_state_machine(start_target_current_state_machine_tb),
.target_current_state_machine_finished(target_current_state_machine_finished_tb),
.sm_clk(sm_clk_tb),
.start_request_a(start_request_a_tb),
.start_request_b(start_request_b_tb),
.finish_a(finish_a_tb),
.finish_b(finish_b_tb),
.reset_start_request_a(reset_start_request_a_tb),
.reset_start_request_b(received_data_b_tb),
.input_argumnets_a(input_argumnets_a_tb),
.input_argumnets_b(input_argumnets_b_tb),
.received_data_a(received_data_a_tb),
.received_data_b(received_data_b_tb),
.reset(reset_tb),
.in_received_data(in_received_data_tb)
);

initial forever begin //clock

sm_clk_tb = 0;
#1;
sm_clk_tb = 1;
#1;


end


initial begin
reset_tb = 0;
target_current_state_machine_finished_tb= 0;
start_request_a_tb = 1;
start_request_b_tb = 0;
#1;

```

```
70    reset_tb = 1;
71    target_current_state_machine_finished_tb= 0;
72    start_request_a_tb = 1;
73    start_request_b_tb = 0;
74    #1;
75
76    reset_tb = 0;
77    target_current_state_machine_finished_tb= 0;
78    start_request_a_tb = 1;
79    start_request_b_tb = 0;
80    #1;
81
82    reset_tb = 0;
83    target_current_state_machine_finished_tb= 0;
84    start_request_a_tb = 1;
85    start_request_b_tb = 0;
86    #1;
87
88    reset_tb = 0;
89    target_current_state_machine_finished_tb= 0;
90    start_request_a_tb = 1;
91    start_request_b_tb = 0;
92    #2;
93
94
95    reset_tb = 0;
96    target_current_state_machine_finished_tb= 1;
97    start_request_a_tb = 1;
98    start_request_b_tb = 0;
99    #1;
100
101   reset_tb = 0;
102   target_current_state_machine_finished_tb= 1;
103   start_request_a_tb = 0;
104   start_request_b_tb = 0;
105   #2;
106
107   ////////////
108   reset_tb = 0;
109   target_current_state_machine_finished_tb= 0;
110   start_request_a_tb = 0;
111   start_request_b_tb = 1;
112   #2;
113
114   reset_tb = 0;
115   target_current_state_machine_finished_tb= 0;
116   start_request_a_tb = 0;
117   start_request_b_tb = 1;
118   #1;
119
120   reset_tb = 0;
121   target_current_state_machine_finished_tb= 0;
122   start_request_a_tb = 0;
123   start_request_b_tb = 1;
124   #1;
125
126   reset_tb = 0;
127   target_current_state_machine_finished_tb= 0;
128   start_request_a_tb = 0;
129   start_request_b_tb = 1;
130   #2;
131
132
133   reset_tb = 0;
134   target_current_state_machine_finished_tb= 1;
135   start_request_a_tb = 0;
136   start_request_b_tb = 1;
137   #1;
138
```

```
139   reset_tb = 0;
140   target_current_state_machine_finished_tb= 1;
141   start_request_a_tb = 0;
142   start_request_b_tb = 0;
143   #2;
144
145
146   end
147
148   endmodule
```