```
 1  library ieee;
 2  use ieee.std_logic_1164.all;
 3
 4  entity memorySubsystem is
 5     port(
 6        BusMuxOut      : in std_logic_vector(31 downto 0);
 7        BusMuxInMDR : inout std_logic_vector(31 downto 0);
 8        MDRin, MARin, clock, clear: in std_logic;
 9        readSig, writeSig, mdrReadSig: in std_logic
10     );
11  end entity;
12
13  architecture behaviour of memorySubsystem is
14
15  component multiplexerMDR is
16     port(
17        BusMuxOut, Mdatain: IN STD_LOGIC_VECTOR(31 DOWNTO 0);
18        ReadChannel : IN STD_LOGIC;
19        MDRMuxOut : OUT STD_LOGIC_VECTOR(31 DOWNTO 0)
20     );
21  end component;
22
23  component reg_32 is
24     port(
25        clk, clr, Rin : IN STD_LOGIC;
26        BusMuxOut : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
27        BusMuxIn : OUT STD_LOGIC_VECTOR(31 DOWNTO 0)
28     );
29  end component;
30
31  component regMAR IS
32     PORT(
33        clk, clr, Rin : IN STD_LOGIC;
34        BusMuxOut : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
35        BusMuxIn : OUT STD_LOGIC_VECTOR(8 DOWNTO 0)
36        );
37  END component;
38
39
40  component ram IS
41     PORT(
42        address     : IN STD_LOGIC_VECTOR (8 DOWNTO 0);
43        clock    : IN STD_LOGIC  := '1';
44        data     : IN STD_LOGIC_VECTOR (31 DOWNTO 0);
45        rden     : IN STD_LOGIC  := '1';
46        wren     : IN STD_LOGIC ;
47        q     : OUT STD_LOGIC_VECTOR (31 DOWNTO 0)
48     );
49  END component;
50
51  signal mdMuxToMDR, mdrToRam: std_logic_vector(31 downto 0);
52  signal address : std_logic_vector(8 downto 0);
53  begin
54  --MDMUX
55  U0: multiplexerMDR port map(
56        BusMuxOut => BusMuxOut,
57        Mdatain => BusMuxInMDR,
58        ReadChannel => mdrReadSig,
59        MDRMuxOut => mdMuxToMDR
60  );
61  --MDR Reg
62  U1: reg_32 port map(
63     clk => clock,
64     clr => clear,
65     Rin => MDRin,
66     BusMuxOut => mdMuxToMDR,
67     BusMuxIn => mdrToRam
68  );
69  --MAR
70  U2: regMAR port map(
71     clk => clock,
```

```
72      clr => clear,
73      Rin => MARin,
74      BusMuxOut => BusMuxOut,
75      BusMuxIn => address
76  );
77  --RAM
78  U3: ram port map(
79        address => address,
80        clock => clock,
81        data => mdrToRam,
82        rden => readSig,
83        wren => writeSig,
84        q => BusMuxInMDR
85  );
86  end architecture;
87
```