```
 1  library IEEE;
 2  use IEEE.STD_LOGIC_1164.ALL;
 3  use ieee.std_logic_arith.all;
 4  use ieee.std_logic_unsigned.all;
 5
 6  entity CPU is
 7      Port ( DATA : inout  STD_LOGIC_VECTOR (3 downto 0);
 8             CMD  : inout  STD_LOGIC_VECTOR (3 downto 0);
 9             RAMdataRD : in STD_LOGIC;
10             ADDR : out    STD_LOGIC_VECTOR (7 downto 0);
11             RSTM : in     STD_LOGIC;
12             CLKM : in     STD_LOGIC);
13  end CPU;
14
15  architecture bhv of CPU is
16
17     --Register Bank--
18     component register_bank
19        port( internal_data_bus : inout STD_LOGIC_VECTOR(3  downto 0);
20              rbcl              : in    STD_LOGIC_VECTOR(31 downto 0);
21              DRBO              : out   STD_LOGIC_VECTOR(3  downto 0);
22              DROO              : out   STD_LOGIC_VECTOR(3  downto 0);
23              AR0O              : out   STD_LOGIC_VECTOR(7  downto 0);
24              rstIN             : in    STD_LOGIC);
25        end component;
26
27     --Address path--
28     component addr_path
29        port( AR0     : in  STD_LOGIC_VECTOR (7 downto 0);
30              DRO     : in  STD_LOGIC_VECTOR (3 downto 0);
31              DRB     : in  STD_LOGIC_VECTOR (3 downto 0);
32              strDC   : in  STD_LOGIC;
33              strPC   : in  STD_LOGIC;
34              strADDR : in  STD_LOGIC;
35              jmpA    : in  STD_LOGIC;
36              jmpB    : in  STD_LOGIC;
37              jmpO    : in  STD_LOGIC;
38              SEL     : in  STD_LOGIC_VECTOR (1 downto 0);
39              ADDR    : out STD_LOGIC_VECTOR (7 downto 0);
40              rst     : in  STD_LOGIC);
41        end component;
42
43     --Registro semplice N bit--
44     component reg_N
45        port( I     : in  STD_LOGIC_VECTOR (3 downto 0);
```

```
46              O     : out STD_LOGIC_VECTOR (3 downto 0);
47              str   : in  STD_LOGIC;
48              rst   : in  STD_LOGIC);
49      end component;
50
51      --Registrio D_I/O:
52      component D_IO_reg
53         port( DATABUS           : inout  STD_LOGIC_VECTOR (3 downto 0
    );
54              internalDataBus    : inout  STD_LOGIC_VECTOR (3 downto 0
    );
55              strIO              : in     STD_LOGIC;
56              invDir             : in     STD_LOGIC;
57              rst                : in     STD_LOGIC);
58         end component;
59
60      --Unità di controllo CU--
61      component CU
62         port(RSTM      : in STD_LOGIC;
63              CLKM      : in STD_LOGIC;
64              intDataBus: in STD_LOGIC_VECTOR(3 downto 0);
65              cmdBUS    : inout STD_LOGIC_VECTOR(3 downto 0);
66              RAMdataRD : in STD_LOGIC;
67              rstout    : out STD_LOGIC;
68              strPC     : out STD_LOGIC;
69              strDC     : out STD_LOGIC;
70              strADDR   : out STD_LOGIC;
71              strIO     : out STD_LOGIC;
72              strACC    : out STD_LOGIC;
73              strOP     : out STD_LOGIC;
74              jmpO      : out STD_LOGIC;
75              jmpB      : out STD_LOGIC;
76              jmpA      : out STD_LOGIC;
77              SEL       : out STD_LOGIC_VECTOR (1 downto 0);
78              invDir    : out STD_LOGIC;
79              rbcl      : out STD_LOGIC_VECTOR (31 downto 0));
80      end component;
81
82      --Segnali interni--
83      signal intDataBus : STD_LOGIC_VECTOR (3 downto 0);
84      signal rst,strDC,strPC,strADDR,strIO,strACC,strOP : STD_LOGIC;
85      signal jmpO,jmpB,jmpA,invDIR : STD_LOGIC;
86      signal SEL : STD_LOGIC_VECTOR (1 downto 0);
87      signal AR0 : STD_LOGIC_VECTOR (7 downto 0);
88      signal DRB, DRO : STD_LOGIC_VECTOR (3 downto 0);
```

```
 89      signal aluIN1,aluIN2 : STD_LOGIC_VECTOR(3 downto 0);
 90      signal rbcl : STD_LOGIC_VECTOR(31 downto 0);
 91      signal CMDbus : STD_LOGIC_VECTOR(3 downto 0);
 92
 93
 94  begin
 95
 96      --CMD(0) <= RAMdataRD;
 97      --strobeRAM <= CMD(1);
 98      CMDbus(0) <= RAMdataRD;
 99      CMD(1) <= CMDbus(1);
100      CMD(2) <= CMDbus(2);
101
102      --Registro D_I/O: DATABUS,internalDataBus,strIO,invDir,rst
103      DIO: D_IO_reg PORT MAP (DATA,intDataBus,strIO,invDIR,rst);
104
105      -- Registro semplice N bit: PORT MAP I,O,str,rst
106      ACC: reg_N PORT MAP (intDataBus,aluIN1,strACC,rst);
107      REGOP: reg_N PORT MAP (intDataBus,aluIN2,strOP,rst);
108
109      --Address path: PORT MAP:
     AR0,DRO,DRB,strDC,strPC,strADDR,jmpA,jmpB,jmpO,SEL,ADDR,rst
110      AP: addr_path PORT MAP(AR0,DRO,DRB,strDC,strPC,strADDR,jmpA,jmpB
     ,jmpO,SEL,ADDR,rst);
111
112      --Register bank: PORTMAP: internalDataBus,rbcl,DRBO,DROO,AR0O
113      RB: register_bank PORT MAP (intDataBus,rbcl,DRB,DRO,AR0,rst);
114
115      --CU: PORT MAP:
     RSTM,CLKM,intDataBus,cmdBUS,rst,strPC,strDC,strADDR,strIO,strACC,str
     OP
116                   -- jmpO,jmpB,jmpA,SEL,invDir,rbcl
117      CUnit: CU PORT MAP (RSTM,CLKM,intDataBus,CMDbus,RAMdataRD,rst,
     strPC,strDC,strADDR,strIO,strACC,strOP,jmpO,jmpB,jmpA,SEL,invDIR,
     rbcl);
118
119  end bhv;
120
121
```