

```

1  --34567890123456789012345678901234567890123456789012345678901234567890
2  -- $Id: control_registers.vhd,v 1.3 2004/12/09 22:30:21 tofp Exp $
3  --*****CONTROL_REGISTERS.VHD*****
4  --* CONTROL_REGISTERS.VHD
5  --*
6  --*
7  --* REVISION HISTORY:
8  --* 11-Oct-2001 CS Original coding
9  --* 17-May-2002 CS XX registers have been added
10 --
11 --*****CONTROL_REGISTERS.VHD*****
12
13 LIBRARY ieee;
14 USE ieee.std_logic_1164.ALL;
15
16 ENTITY control_registers IS
17 PORT (
18     clock      : IN std_logic;
19     arstn      : IN std_logic;
20     reg_data   : IN std_logic_vector ( 7 DOWNTO 0);
21     reg_addr   : IN std_logic_vector ( 5 DOWNTO 0);
22     reg_load   : IN std_logic;
23     reg_lock   : IN std_logic;
24     ps_reg     : OUT std_logic_vector ( 7 DOWNTO 0);
25     bl_reg     : OUT std_logic_vector ( 7 DOWNTO 0);
26     dt_reg     : OUT std_logic_vector ( 7 DOWNTO 0);
27     fc_reg     : OUT std_logic_vector ( 7 DOWNTO 0);
28     te_reg     : OUT std_logic_vector ( 7 DOWNTO 0);
29     xx_reg     : OUT std_logic_vector ( 7 DOWNTO 0)
30 );
31 END control_registers;
32
33 ARCHITECTURE SYN OF control_registers IS
34
35 BEGIN
36
37 main : PROCESS (clock, arstn)
38     VARIABLE ps_reg_int      : std_logic_vector (7 DOWNTO 0);
39     VARIABLE bl_reg_int      : std_logic_vector (7 DOWNTO 0);
40     VARIABLE dt_reg_int      : std_logic_vector (7 DOWNTO 0);
41     VARIABLE fc_reg_int      : std_logic_vector (7 DOWNTO 0);
42     VARIABLE te_reg_int      : std_logic_vector (7 DOWNTO 0);
43     VARIABLE xx_reg_int      : std_logic_vector (7 DOWNTO 0);
44     VARIABLE psreg_enable    : boolean;
45     VARIABLE blreg_enable    : boolean;
46     VARIABLE dtreg_enable    : boolean;
47     VARIABLE fcreg_enable    : boolean;
48     VARIABLE tereg_enable    : boolean;
49     VARIABLE xxreg_enable    : boolean;
50
51 BEGIN
52
53     IF (arstn = '0') THEN
54         ps_reg_int      := (OTHERS => '0');
55         bl_reg_int      := (OTHERS => '0');
56         dt_reg_int      := (OTHERS => '0');
57         fc_reg_int      := (OTHERS => '0');
58         te_reg_int      := (OTHERS => '0');
59         xx_reg_int      := (OTHERS => '0');
60         psreg_enable    := false;
61         blreg_enable    := false;
62         dtreg_enable    := false;
63         fcreg_enable    := false;
64         tereg_enable    := false;
65         xxreg_enable    := false;
66         ps_reg          <= (OTHERS => '0');
67         bl_reg          <= (OTHERS => '0');

```

```

67      dt_reg      <= (OTHERS => '0');
68      fc_reg      <= (OTHERS => '0');
69      te_reg      <= (OTHERS => '0');
70      xx_reg      <= (OTHERS => '0');
71      ELSIF (clock'event AND clock = '1') THEN
72
73          ps_reg_int := reg_data;
74          IF (psreg_enable) THEN
75              ps_reg <= ps_reg_int;
76          END IF;
77          psreg_enable := ((reg_load = '1') AND (reg_addr = "000000"));
78
79          bl_reg_int := reg_data;
80          IF (blreg_enable) THEN
81              bl_reg <= bl_reg_int;
82          END IF;
83          blreg_enable := ((reg_load = '1') AND (reg_addr = "000001"));
84
85          dt_reg_int := reg_data;
86          IF (dtreg_enable) THEN
87              dt_reg <= dt_reg_int;
88          END IF;
89          dtreg_enable := ((reg_load = '1') AND (reg_addr = "000010"));
90
91          fc_reg_int := reg_data;
92          IF (fcreg_enable) THEN
93              fc_reg <= fc_reg_int;
94          END IF;
95          fcreg_enable := ((reg_load = '1') AND (reg_addr = "000011"));
96
97          te_reg_int := reg_data;
98          IF (tereg_enable) THEN
99              te_reg <= te_reg_int;
100         END IF;
101         tereg_enable := ((reg_load = '1') AND (reg_addr = "000100"));
102
103        xx_reg_int := reg_data;
104        IF (xxreg_enable) THEN
105            xx_reg <= xx_reg_int;
106        END IF;
107        xxreg_enable := ((reg_load = '1') AND (reg_addr = "000101"));
108
109    END IF;
110
111 END PROCESS;
112
113 END SYN;
114

```