

```

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.NUMERIC_STD.all;
4
5  ENTITY biu_dma2fifo IS
6      PORT (
7          INPUT  : IN  STD_LOGIC_VECTOR(3 DOWNTO 0);
8          OUTPUT : OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
9          STATE  : OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
10         RSTATE : IN  STD_LOGIC_VECTOR(3 DOWNTO 0);
11         RESET  : IN  STD_LOGIC;
12         IN_DELAY : IN  STD_LOGIC;
13         OUT_DELAY : OUT STD_LOGIC
14     );
15 END ENTITY biu_dma2fifo;
16
17 ARCHITECTURE ALC_XMS OF biu_dma2fifo IS
18
19     COMPONENT FGC_Block IS
20         PORT (
21             INPUT  : IN  STD_LOGIC_VECTOR(7 DOWNTO 0);
22             OUTPUT : OUT STD_LOGIC
23         );
24     END COMPONENT;
25
26     COMPONENT NSTATE_Block IS
27         PORT (
28             INPUT  : IN  STD_LOGIC_VECTOR(7 DOWNTO 0);
29             OUTPUT : OUT STD_LOGIC_VECTOR(3 DOWNTO 0)
30         );
31     END COMPONENT;
32
33     COMPONENT OUT_Block IS
34         PORT (
35             INPUT  : IN  STD_LOGIC_VECTOR(7 DOWNTO 0);
36             OUTPUT : OUT STD_LOGIC_VECTOR(1 DOWNTO 0)
37         );
38     END COMPONENT;
39
40     COMPONENT D_Latch IS
41         Port (
42             EN : in  STD_LOGIC;
43             D  : in  STD_LOGIC;
44             Q  : out STD_LOGIC
45         );
46     END COMPONENT;
47
48     SIGNAL SSTATE : STD_LOGIC_VECTOR(3 DOWNTO 0);
49     SIGNAL SNSTATE: STD_LOGIC_VECTOR(3 DOWNTO 0);
50     SIGNAL SLSTATE: STD_LOGIC_VECTOR(3 DOWNTO 0);
51     SIGNAL SSOUT  : STD_LOGIC_VECTOR(1 DOWNTO 0);
52     SIGNAL SOUT   : STD_LOGIC_VECTOR(1 DOWNTO 0);
53
54 BEGIN
55     B1: FGC_Block    PORT MAP(INPUT & SSTATE, OUT_DELAY);
56     B2: NSTATE_Block PORT MAP(INPUT & SSTATE, SNSTATE);
57     B3: OUT_Block    PORT MAP(INPUT & SSTATE, SOUT);
58     STT0: D_Latch    PORT MAP(IN_DELAY OR RESET, SLSTATE(0), SSTATE(0));
59     STT1: D_Latch    PORT MAP(IN_DELAY OR RESET, SLSTATE(1), SSTATE(1));
60     STT2: D_Latch    PORT MAP(IN_DELAY OR RESET, SLSTATE(2), SSTATE(2));
61     STT3: D_Latch    PORT MAP(IN_DELAY OR RESET, SLSTATE(3), SSTATE(3));
62     OUT0: D_Latch    PORT MAP(SSOUT(0) XOR SOUT(0), SOUT(0), OUTPUT(0));
63     OUT1: D_Latch    PORT MAP(SSOUT(1) XOR SOUT(1), SOUT(1), OUTPUT(1));
64
65     PROCESS(INPUT, RSTATE, RESET, IN_DELAY)
66     BEGIN
67         OUTPUT <= SSOUT;
68         IF (RESET = '1') THEN
69             SLSTATE <= RSTATE;
70         ELSE
71             SLSTATE <= SNSTATE;
72         END IF;
73         STATE <= SSTATE;
74     END PROCESS;
75 END ALC_XMS;

```