Diagram

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Data | Present State | Next State |
| 0 | S0 | S0 |
| 0 | S1 | S1 |
| 0 | S2 | S2 |
| 0 | S3 | S3 |
| 0 | S4 | S4 |
| 0 | S5 | S5 |
| 0 | S6 | S6 |
| 0 | S7 | S7 |
| 0 | S8 | S8 |
| 1 | S0 | S4 |
| 1 | S1 | S8 |
| 1 | S2 | S7 |
| 1 | S3 | S6 |
| 1 | S4 | S5 |
| 1 | S5 | S3 |
| 1 | S6 | S2 |
| 1 | S7 | S1 |
| 1 | S8 | S0 |

|  |  |  |
| --- | --- | --- |
| Data | Present State | Next State |
| 0 | 0000 | 0000 |
| 0 | 0001 | 0001 |
| 0 | 0010 | 0010 |
| 0 | 0011 | 0011 |
| 0 | 0100 | 0100 |
| 0 | 0101 | 0101 |
| 0 | 0110 | 0110 |
| 0 | 0111 | 0111 |
| 0 | 1000 | 1000 |
| 1 | 0000 | 0100 |
| 1 | 0001 | 1000 |
| 1 | 0010 | 0111 |
| 1 | 0011 | 0110 |
| 1 | 0100 | 0101 |
| 1 | 0101 | 0011 |
| 1 | 0110 | 0010 |
| 1 | 0111 | 0001 |
| 1 | 1000 | 0000 |

library ieee;

use ieee.std\_logic\_1164.all;

entity machine is

port

(

clk : in std\_logic;

data\_in : in std\_logic;

reset : in std\_logic;

student\_id : out std\_logic\_vector(3 downto 0);

current\_state: out std\_logic\_vector(3 DOWNTO 0);

);

end entity;

architecture fsm of machine is

type state\_type is (s0, s1, s2, s3, s4, s5, s6, s7, s8);

signal yfsm : state\_type;

begin

process (clk, reset)

begin

if reset = '1' then

yfsm <= s0:

elsif (clk 'EVENT AND clk = '1')

case yfsm is

when s0 =>

if data\_in = '1' then

yfsm <= s4;

else yfsm <= s0;

end if;

when s1 =>

if data\_in = '1' then

yfsm <= s8;

else yfsm <= s1;

end if;

when s2 =>

if data\_in = '1' then

yfsm <= s7;

else yfsm <= s2;

end if;

when s3 =>

if data\_in = '1' then

yfsm <= s6;

else yfsm <= s3;

end if;

when s4 =>

if data\_in = '1' then

yfsm <= s5;

else yfsm <= s4;

end if;

when s5 =>

if data\_in = '1' then

yfsm <= s3;

else yfsm <= s5;

end if;

when s6 =>

if data\_in = '1' then

yfsm <= s2;

else yfsm <= s6;

end if;

when s7 =>

if data\_in = '1' then

yfsm <= s1;

else yfsm <= s7;

end if;

when s8 =>

if data\_in = '1' then

yfsm <= s0;

else yfsm <= s8;

end if;

end case;

end if;

end process

process(yfsm, data\_in)

begin

case yfsm is

when s0=> "0101";

when s1 => "0000";

when s2 => "0001";

when s3 => "0000";

when s4 => "0001";

when s5 => "0001";

when s6 => "1000";

when s7 => "0101";

when s8 => "0010";

end case;

end process;