|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| X1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Y |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Q | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | t |

0 0 ns, 1 10 ns, 0 20 ns, 1 30 ns, 0 40 ns, 1 50 ns,0 60 ns, 1 70 ns, 1 80 ns,0 90 ns, 1 100 ns, 0 110 ns,1 120 ns, 1 130 ns, 0 140 ns, 1 150 ns

0 0 ns, 0 10 ns, 0 20 ns, 0 30 ns, 0 40 ns, 0 50 ns,0 60 ns, 0 70 ns, 1 80 ns,0 90 ns, 0 100 ns, 1 110 ns,1 120 ns, 0 130 ns, 0 140 ns, 0 150 ns х0

Мое

0 0 ns, 1 10 ns, 0 20 ns, 1 30 ns, 0 40 ns, 1 50 ns, 0 60 ns, 1 110 ns, 0 120 ns, 1 130 ns

0 0 ns, 1 10 ns, 0 20 ns, 1 30 ns, 0 40 ns, 1 50 ns, 0 60 ns, 1 70 ns, 0 80 ns, 1 90 ns, 0 100 ns, 1 140 ns, 0 150 ns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Q | Y | X2\X1 | | | |
| 00 | 01 | 10 | 11 |
| 0 | 0 | (0) | - | 1 | - |
| 1 | 0 | 2 | - | (1) | - |
| 2 | 1 | (2) | - | 3 | - |
| 3 | 1 | 4 | - | (3) | - |
| 4 | 0 | (4) | - | 5 | - |
| 5 | 0 | 6 | - | (5) | - |
| 6 | 0 | (6) | - | 7 | - |
| 7 | 0 | - | - | (7) | 8 |
| 8 | 0 | 9 | - | - | (8) |
| 9 | 0 | (9) | - | 10 | - |
| 10 | 0 | - | 11 | (10) | - |
| 11 | 0 | - | (11) | - | 12 |
| 12 | 0 | - | - | 13 | (12) |
| 13 | 1 | 14 | - | (13) | - |
| 14 | 0 | (14) | - | 15 | - |
| 15 | 0 | (!)0 | - | (15) | - |

entity avtom is

generic (Lx: integer :=1;Ly: integer :=0; -- ????? ???????? ??. ? ???. ??????????

K: integer :=15; M: integer :=3; --?????????? ????? ? ???????? ???. ?????/???.

TZ: time :=2ns); -- ??????? ??? ??????????? ?????????????????? ????????????

port(

x: in bit\_vector (Lx downto 0); -- ?????: ? ??????? ?(0)-?, ? ?(1)-D

-- clk: in bit ; ?????????? ( ???????? ??? ??????????? ???????? !!!)

y: out bit\_vector (Ly downto 0) --??????: ? ??????? Q

);

end avtom;

architecture avtom of avtom is

function vecint (vec1: bit\_vector)

return integer is

variable retval:integer:=0;

begin

for i in vec1'length-1 downto 1 loop

if (vec1(i)='1') then retval:=(retval+1)\*2;

else retval:= retval\*2; end if;

end loop;

if vec1(0)='1' then retval:=retval+1;

else null; end if;

return retval;

end vecint;

type stab is array (0 to K, 0 to M) of integer;

constant Ust: integer :=16;--??? ???????????? ?????????

constant tab\_st: stab :=((0,20,20,1),

(2,20,1,20),

(2,0,3,20),(4,20,3,20),(4,20,5,20),(6,20,5,20),(6,20,7,20),(20,20,7,8),

(9,20,20,8),(9,20,10,20),(20,11,10,20),(20,11,20,12),(20,20,13,12),(14,20,13,20),(14,20,15,20),(0,20,15,20)

); --??????? ?????????

type outtab is array (0 to 15) of bit\_vector(0 downto 0);

constant tab\_y : outtab :=("0","0","1","1","0","0","0","0", "0","0","0","0","0","1","0","0"); --??????? ???????

signal st, nexst: integer:=0; --????????? ?????????? ????????? ????????

begin

process

begin

wait on x, st;

if st=Ust then null;

else

nexst<=tab\_st(st,vecint(x));

y<=transport tab\_y(st)after TZ;

end if;

end process;

process

begin

wait on nexst; --( ???????? ??? ???????????? ????????, ?????? ??? ??????????? !!!)

--wait on clk; ( ???????? ??? ??????????? ????????, ?????? ??? ???????????? !!!)

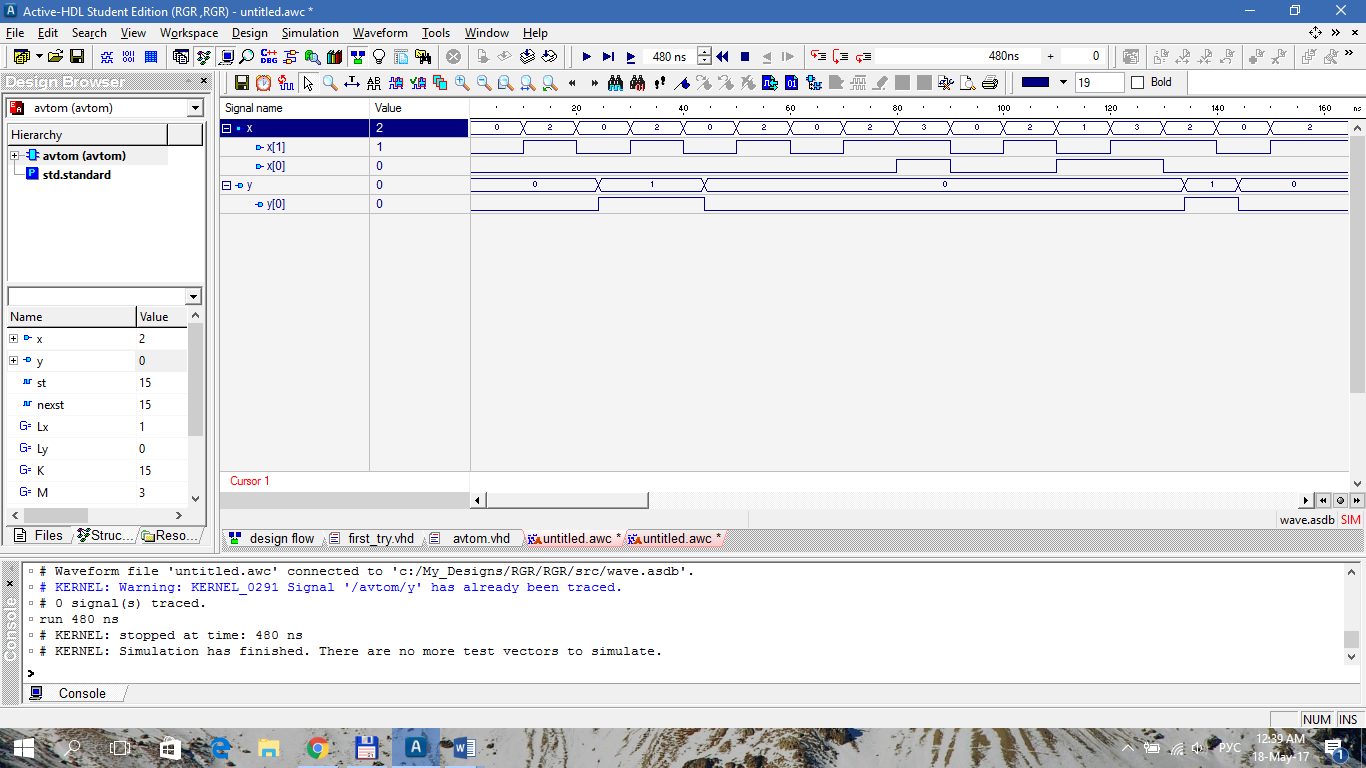
--if pozadge(clk) ='1' then (???????? ??? ??????????? ????????, ?????? ??? ????????. !!!)

st<= transport nexst after TZ;

--end if; (???????? ??? ??????????? ????????, ?????? ??? ????????. !!!)

end process;

end avtom;



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 2 | 11 | 1 | 3 | 8 |
| 4 |  |  | 5 | 13 | 12 |
| 6 |  |  | 7 |  |  |
| 9 |  |  | 10 |  |  |
| 14 |  |  | 15 |  |  |

0-4 -

0-6 1-7 +

0-9 1-10 +

0-14 1-15 -

4-6 5-7 +

4-9 5-10 +

4-14 5-15 +

6-9 7-10 +

6-14 7-15 +

9-14 10-15 +

1-5 -

1-7 +

1-10 +

1-15 -

5-7 +

5.-10 +

5-15 0-6 +

7-10 +

7-15 +

10-15 +

3-13 4-14 5-15 +

8-12 +

Можливий варіант покриття:4-9-14, 5-10-15, 0-6,1-7, 3-13,8-12, 2, 11

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Q | Old | y | X2\X1 | | | |
| 00 | 01 | 10 | 11 |
| 0 | {4,9,14} | 0 | (0) | - | 1 | - |
| 1 | {5,10,15} | 0 | 2 | 7 | (1) | - |
| 2 | {0,6} | 0 | (2) | - | 3 | - |
| 3 | {1,7} | 0 | 6 | - | (3) | 5 |
| 4 | {3,13} | 1 | 0 | - | (4) | - |
| 5 | {8,12} | 0 | 0 | - | 4 | (5) |
| 6 | 2 | 1 | (6) | - | 4 | - |
| 7 | 11 | 0 | - | (7) | - | 5 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| X1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Y |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Q | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | t |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Q | Old | y | X2\X1 | | | |
| 00 | 01 | 10 | 11 |
| 0 | {0,6} | 0 | (0) | - | 1 | - |
| 1 | {1,7} | 0 | 2 | - | (1) | 6 |
| 2 | 2 | 1 | (2) | - | 3 | - |
| 3 | {3,13} | 1 | 4 | - | (3) | - |
| 4 | {4,9,14} | 0 | (4) | - | 5 | - |
| 5 | {5,10,15} | 0 | 0 | 7 | (5) | - |
| 6 | {8,12} | 0 | 4 | - | 3 | (6) |
| 7 | 11 | 0 | - | (7) | - | 6 |

0-4

1-5

2-0

3-1

4-3

5-6

6-2

(0,16,1,16),

(2,16,1,16),

(2,16,3,16),

(4,16,3,16),

(4,16,5,16),

(0,7,5,16),

(4,16,3,6),

(16,7,16,6)

entity avtom1 is

generic (Lx: integer :=1;Ly: integer :=0; -- ????? ???????? ??. ? ???. ??????????

K: integer :=7; M: integer :=3; --?????????? ????? ? ???????? ???. ?????/???.

TZ: time :=2ns); -- ??????? ??? ??????????? ?????????????????? ????????????

port(

x: in bit\_vector (Lx downto 0); -- ?????: ? ??????? ?(0)-?, ? ?(1)-D

-- clk: in bit ; ?????????? ( ???????? ??? ??????????? ???????? !!!)

y: out bit\_vector (Ly downto 0) --??????: ? ??????? Q

);

end avtom1;

architecture avtom1 of avtom1 is

function vecint (vec1: bit\_vector)

return integer is

variable retval:integer:=0;

begin

for i in vec1'length-1 downto 1 loop

if (vec1(i)='1') then retval:=(retval+1)\*2;

else retval:= retval\*2; end if;

end loop;

if vec1(0)='1' then retval:=retval+1;

else null; end if;

return retval;

end vecint;

type stab is array (0 to K, 0 to M) of integer;

constant Ust: integer :=16;--??? ???????????? ?????????

constant tab\_st: stab :=((0,16,1,16),

(2,16,1,6),

(2,16,3,16),

(4,16,3,16),

(4,16,5,16),

(0,7,5,16),

(4,16,3,6),

(16,7,16,6)

); --??????? ?????????

type outtab is array (0 to 7) of bit\_vector(0 downto 0);

constant tab\_y : outtab :=("0","0","1","1","0","0","0","0"); --??????? ???????

signal st, nexst: integer:=0; --????????? ?????????? ????????? ????????

begin

process

begin

wait on x, st;

if st=Ust then null;

else

nexst<=tab\_st(st,vecint(x));

y<=transport tab\_y(st)after TZ;

end if;

end process;

process

begin

wait on nexst; --( ???????? ??? ???????????? ????????, ?????? ??? ??????????? !!!)

--wait on clk; ( ???????? ??? ??????????? ????????, ?????? ??? ???????????? !!!)

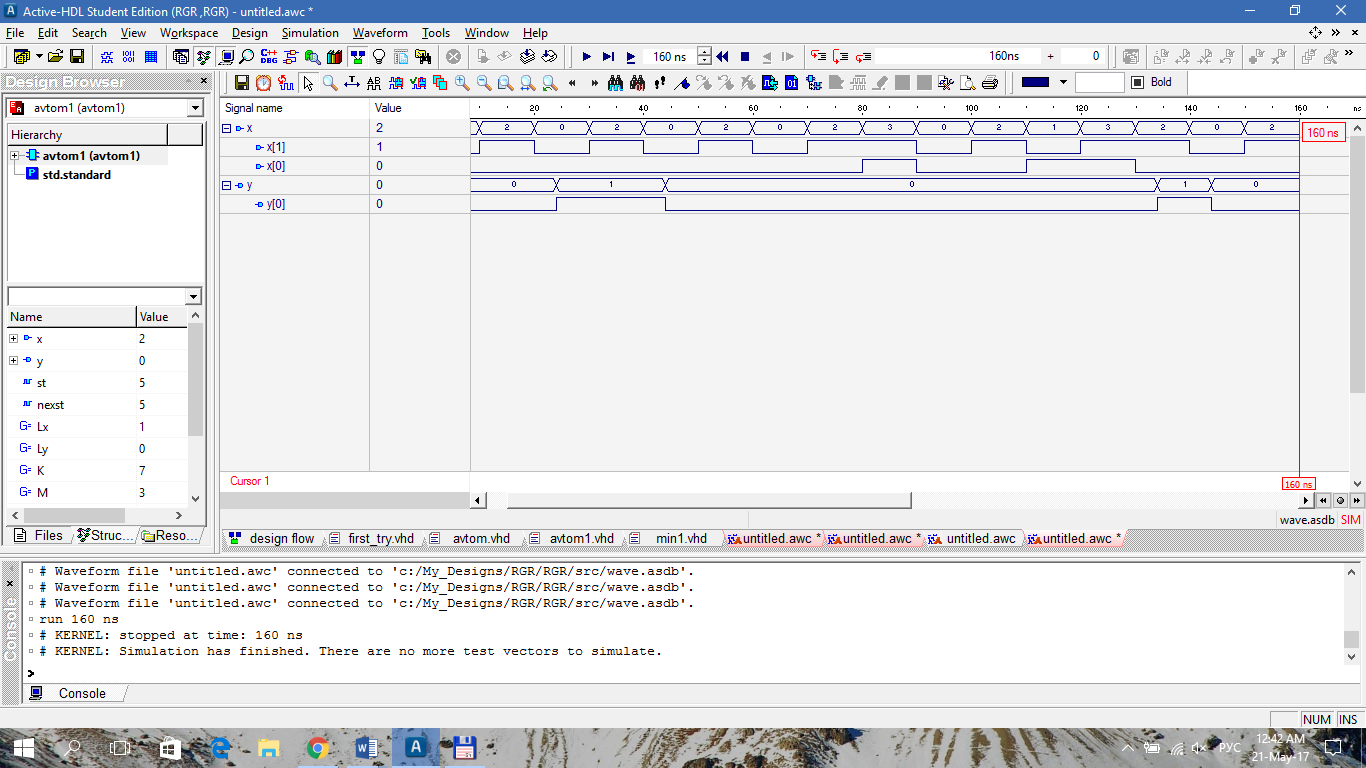
--if pozadge(clk) ='1' then (???????? ??? ??????????? ????????, ?????? ??? ????????. !!!)

st<= transport nexst after TZ;

--end if; (???????? ??? ??????????? ????????, ?????? ??? ????????. !!!)

end process;

end avtom1;



**Об’єднання сумісних внутрішніх станів автомату**

6-7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Q | Old | y | X2\X1 | | | |
| 00 | 01 | 10 | 11 |
| 0 | 0 | 0 | (0) | - | 1 | - |
| 1 | 1 | 0 | 2 | - | (1) | 6 |
| 2 | 2 | 1 | (2) | - | 3 | - |
| 3 | 3 | 1 | 4 | - | (3) | - |
| 4 | 4 | 0 | (4) | - | 5 | - |
| 5 | 5 | 0 | 0 | 6 | (5) | - |
| 6 | {6,7} | 0 | 4 | (6) | 3 | (6) |

entity avtom1 is

generic (Lx: integer :=1;Ly: integer :=0; -- ????? ???????? ??. ? ???. ??????????

K: integer :=6; M: integer :=3; --?????????? ????? ? ???????? ???. ?????/???.

TZ: time :=2ns); -- ??????? ??? ??????????? ?????????????????? ????????????

port(

x: in bit\_vector (Lx downto 0); -- ?????: ? ??????? ?(0)-?, ? ?(1)-D

-- clk: in bit ; ?????????? ( ???????? ??? ??????????? ???????? !!!)

y: out bit\_vector (Ly downto 0) --??????: ? ??????? Q

);

end avtom1;

architecture avtom1 of avtom1 is

function vecint (vec1: bit\_vector)

return integer is

variable retval:integer:=0;

begin

for i in vec1'length-1 downto 1 loop

if (vec1(i)='1') then retval:=(retval+1)\*2;

else retval:= retval\*2; end if;

end loop;

if vec1(0)='1' then retval:=retval+1;

else null; end if;

return retval;

end vecint;

type stab is array (0 to K, 0 to M) of integer;

constant Ust: integer :=16;--??? ???????????? ?????????

constant tab\_st: stab :=((0,16,1,16),

(2,16,1,6),

(2,16,3,16),

(4,16,3,16),

(4,16,5,16),

(0,6,5,16),

(4,6,3,6)

); --??????? ?????????

type outtab is array (0 to 6) of bit\_vector(0 downto 0);

constant tab\_y : outtab :=("0","0","1","1","0","0","0"); --??????? ???????

signal st, nexst: integer:=0; --????????? ?????????? ????????? ????????

begin

process

begin

wait on x, st;

if st=Ust then null;

else

nexst<=tab\_st(st,vecint(x));

y<=transport tab\_y(st)after TZ;

end if;

end process;

process

begin

wait on nexst; --( ???????? ??? ???????????? ????????, ?????? ??? ??????????? !!!)

--wait on clk; ( ???????? ??? ??????????? ????????, ?????? ??? ???????????? !!!)

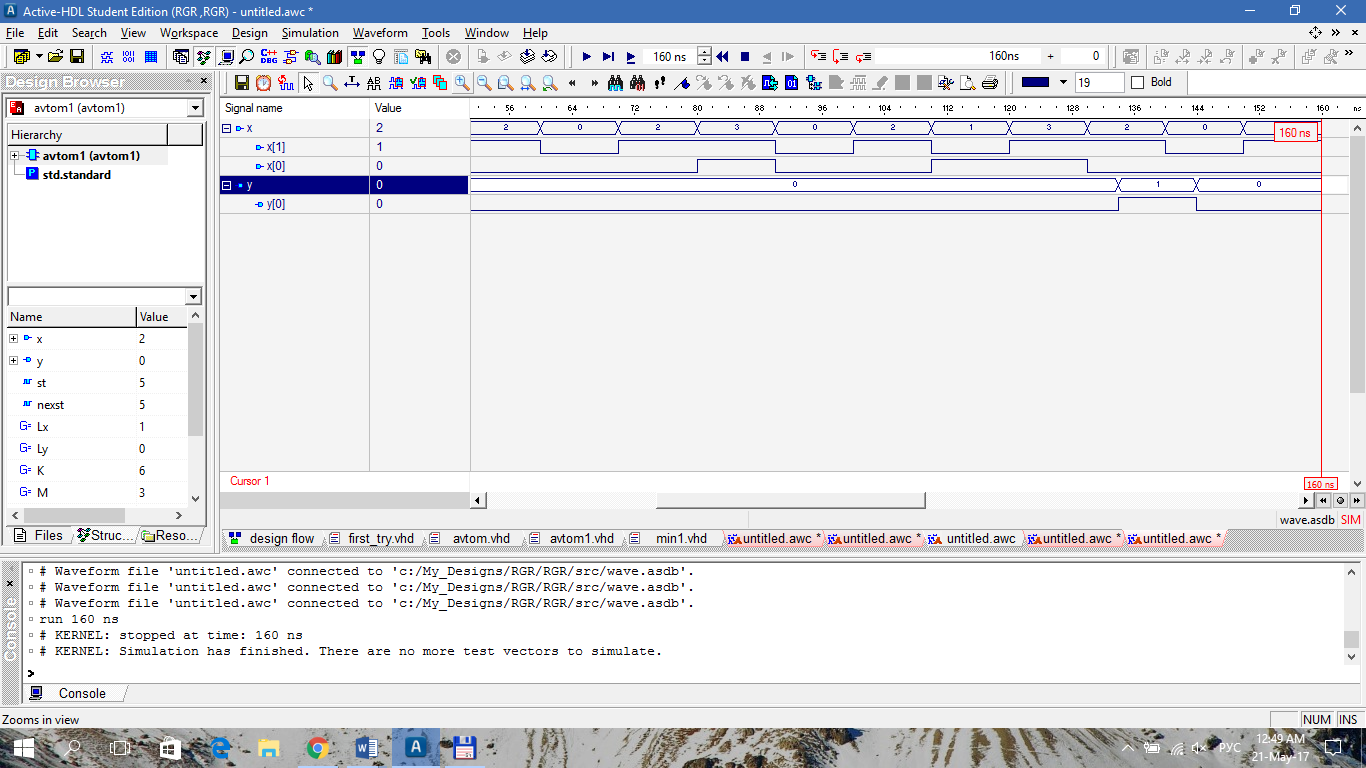
--if pozadge(clk) ='1' then (???????? ??? ??????????? ????????, ?????? ??? ????????. !!!)

st<= transport nexst after TZ;

--end if; (???????? ??? ??????????? ????????, ?????? ??? ????????. !!!)

end process;

end avtom1;



0-4

1-5

2-0

3-1

4-3

5-6

6-2

Замінимо простий перехід (6–>4) складним (6–> 3 –> 4).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Old2 | Q | Y | X2\X1 | | | |
| 00 | 01 | 10 | 11 |
| 0 | 000 | 0 | (000) | - | 110 | - |
| 1 | 110 | 0 | 011 | - | (110) | 100 |
| 2 | 011 | 1 | (011) | - | 010 | - |
| 3 | 010 | 1 | 111 | - | (010) | - |
| 4 | 111 | 0 | (111) | - | 001 | - |
| 5 | 001 | 0 | 000 | 100 | (001) | - |
| 6 | 100 | 0 | 111 | (100) | 010 | (100) |

library IEEE;

use IEEE.STD\_LOGIC\_1164.all;

entity avt\_code is

generic (Lin: integer :=1;

K: integer :=7; M: integer :=3;

TZ: time :=2ns);

port(

x: in bit\_vector (Lin downto 0);

y : inout bit\_vector(0 downto 0)

);

end avt\_code;

architecture avt\_code of avt\_code is

function vecint (vec1: bit\_vector)

return integer is

variable retval:integer:=0;

begin

for i in vec1'length-1 downto 1 loop

if (vec1(i)='1') then retval:=(retval+1)\*2;

else retval:= retval\*2; end if;

end loop;

if vec1(0)='1' then retval:=retval+1;

else null; end if;

return retval;

end vecint;

type stab is array (0 to K, 0 to M) of bit\_vector(2 downto 0);

constant tab\_st: stab :=(

("000","101","110","101"),

("000","100","001","101"),

("111","101","010","101"),

("011","101","010","101"),

("111","100","010","100"),

("101","101","101","101"),

("011","101","110","100"),

("111","101","001","101")

);

type outtab is array (0 to 7) of bit\_vector(0 downto 0);

constant tab\_y : outtab :=("0","0","1","1","0","0","0","0");

signal st,nexst: bit\_vector(2 downto 0):="000";

begin

process

begin

wait on x, st;

if st="101" then null;

else

nexst<=tab\_st(vecint(st),vecint(x));

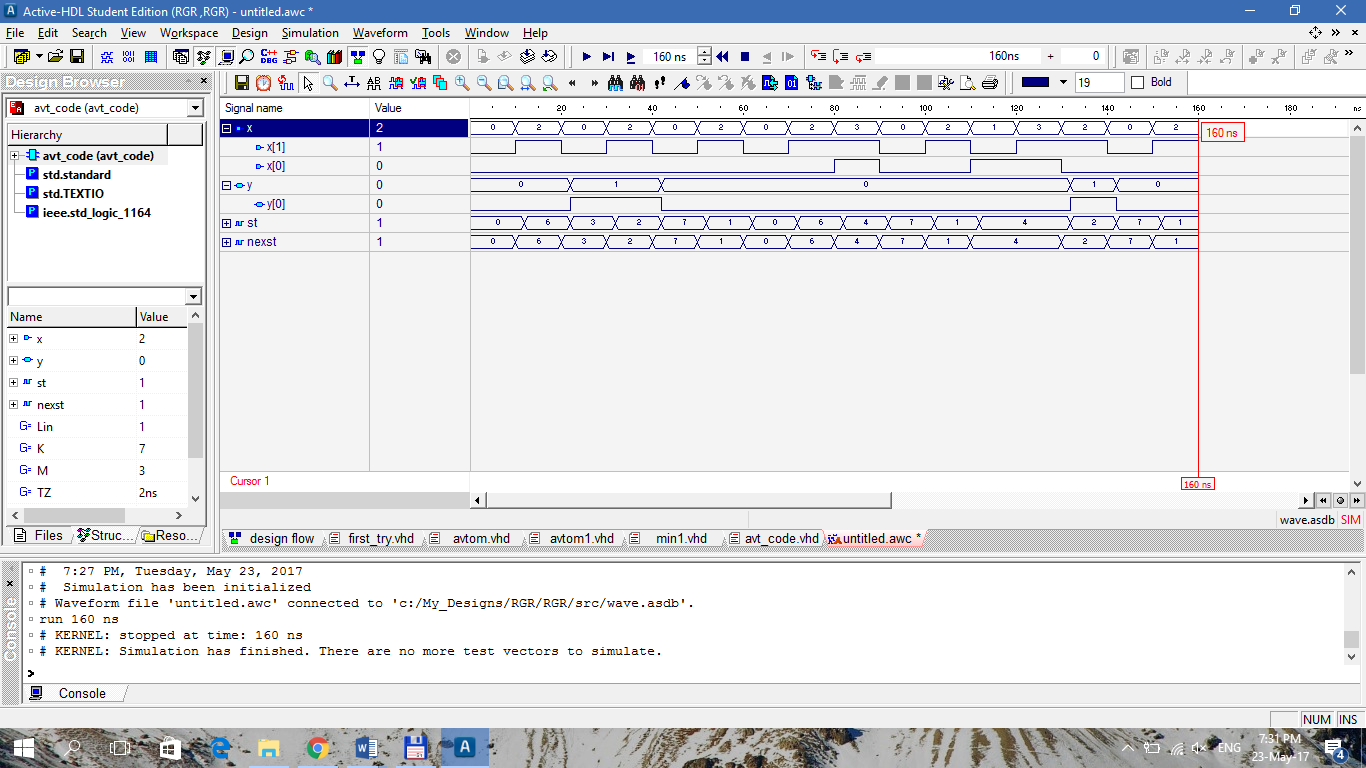
y<= transport tab\_y(vecint(st));

end if;

end process;

st<= transport nexst after TZ;

end avt\_code;



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | X2\X1 | | | | | | | | | | | |
|  |  | 00 | | | 01 | | | 10 | | | 11 | | |
| Q | Y | R2S2 | R1S1 | R0S0 | R2S2 | R1S1 | R0S0 | R2S2 | R1S1 | R0S0 | R2S2 | R1S1 | R0S0 |
| 000 | 0 | \*0 | \*0 | \*0 | - | - | - | 01 | 01 | \*0 | - | - | - |
| 110 | 0 | 10 | 0\* | 01 | - | - | - | 0\* | 0\* | \*0 | 0\* | 10 | \*0 |
| 011 | 1 | \*0 | 0\* | 0\* | - | - | - | \*0 | 0\* | 10 | - | - | - |
| 010 | 1 | 10 | 0\* | 10 | - | - | - | \*0 | 0\* | \*0 | - | - | - |
| 111 | 0 | 0\* | 0\* | 0\* | - | - | - | 10 | 10 | 0\* | - | - | - |
| 001 | 0 | \*0 | \*0 | 10 | 01 | \*0 | 10 | \*0 | \*0 | 0\* | - | - | - |
| 100 | 0 | 0\* | 01 | 01 | 0\* | \*0 | \*0 | 10 | 01 | \*0 | 0\* | \*0 | \*0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **R2** | X2\X1 | | | |
| Q2Q1Q0 | 00 | 01 | 11 | 10 |
| 000 | \* | - | - | 0 |
| 001 | \* | 0 | - | \* |
| 011 | \* | - | - | \* |
| 010 | 1 | - | - | \* |
| 110 | 1 | - | 0 | 0 |
| 111 | 0 | - | - | 1 |
| 101 | - | - | - | - |
| 100 | 0 | 0 | 0 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S2** | X2\X1 | | | |
| Q2Q1Q0 | 00 | 01 | 11 | 10 |
| 000 | 0 | - | - | 1 |
| 001 | 0 | 1 | - | 0 |
| 011 | 0 | - | - | 0 |
| 010 | 0 | - | - | 0 |
| 110 | 0 | - | \* | \* |
| 111 | \* | - | - | 0 |
| 101 | - | - | - | - |
| 100 | \* | \* | \* | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **R1** | X2\X1 | | | |
| Q2Q1Q0 | 00 | 01 | 11 | 10 |
| 000 | \* | - | - | 0 |
| 001 | \* | \* | - | \* |
| 011 | 0 | - | - | 0 |
| 010 | 0 | - | - | 0 |
| 110 | 0 | - | 1 | 0 |
| 111 | 0 | - | - | 1 |
| 101 | - | - | - | - |
| 100 | 0 | \* | \* | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S1** | X2\X1 | | | |
| Q2Q1Q0 | 00 | 01 | 11 | 10 |
| 000 | 0 | - | - | 1 |
| 001 | 0 | 0 | - | 0 |
| 011 | \* | - | - | \* |
| 010 | \* | - | - | \* |
| 110 | \* | - | 0 | \* |
| 111 | \* | - | - | 0 |
| 101 | - | - | - | - |
| 100 | 1 | 0 | 0 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **R0** | X2\X1 | | | |
| Q2Q1Q0 | 00 | 01 | 11 | 10 |
| 000 | \* | - | - | \* |
| 001 | 1 | 1 | - | 0 |
| 011 | 0 | - | - | 1 |
| 010 | 1 | - | - | \* |
| 110 | 0 | - | \* | \* |
| 111 | 0 | - | - | 0 |
| 101 | - | - | - | - |
| 100 | 0 | \* | \* | \* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S0** | X2\X1 | | | |
| Q2Q1Q0 | 00 | 01 | 11 | 10 |
| 000 | 0 | - | - | 0 |
| 001 | 0 | 0 | - | \* |
| 011 | \* | - | - | 0 |
| 010 | 0 | - | - | 0 |
| 110 | 1 | - | 0 | 0 |
| 111 | \* | - | - | \* |
| 101 | - | - | - | - |
| 100 | 1 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| **Y** | Q0 | |
| Q2Q1 | 0 | 1 |
| 00 | 0 | 0 |
| 01 | 1 | 1 |
| 11 | 0 | 0 |
| 10 | 0 | - |