**Comparator (Dataflow)**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

library UNISIM;

use UNISIM.VComponents.all;

entity comparator\_data is

Port ( a : in STD\_LOGIC\_VECTOR (1 downto 0);

b : in STD\_LOGIC\_VECTOR (1 downto 0);

eq : out STD\_LOGIC);

end comparator\_data;

architecture Dataflow of comparator\_data is

signal s0, s1, s2, s3 : std\_logic;

begin

s0 <= (not a(0) and not b(0)) and (not a(1) and not b(1));

s1 <= (a(0) and b(0)) and (not a(1) and not b(1));

s2 <= (not a(0) and not b(0)) and (a(1) and b(1));

s3 <= (a(0) and b(0)) and (a(1) and b(1));

eq <= s0 or s1 or s2 or s3;

end Dataflow;

**Comparator (Behavioral)**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

library UNISIM;

use UNISIM.VComponents.all;

entity comparator\_behav is

Port ( a : in STD\_LOGIC\_VECTOR (1 downto 0);

b : in STD\_LOGIC\_VECTOR (1 downto 0);

eq : out STD\_LOGIC);

end comparator\_behav;

architecture Behavioral of comparator\_behav is

begin

process(a, b)

begin

if (a(0) = b(0)) and (a(1) = b(1)) then

eq <= '1';

else

eq <= '0';

end if;

end process;

end Behavioral;

**Comparator (Structural)**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

library UNISIM;

use UNISIM.VComponents.all;

use work.all;

entity comparator\_struct is

Port ( a : in STD\_LOGIC\_VECTOR (1 downto 0);

b : in STD\_LOGIC\_VECTOR (1 downto 0);

eq : out STD\_LOGIC);

end comparator\_struct;

architecture Structural of comparator\_struct is

component comparator1Bit

port( x, y : in std\_logic;

eq : out std\_logic );

end component;

for all : comparator1Bit use entity work.comparator1Bit(Dataflow);

signal out1, out2 : std\_logic;

begin

c1b1 : comparator1Bit port map(a(0), b(0), out1);

c1b2 : comparator1Bit port map(a(1), b(1), out2);

eq <= out1 and out2;

end Structural;

**Comparator testbench**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

library UNISIM;

use UNISIM.VComponents.all;

entity comparator\_tb is

end comparator\_tb;

architecture Behavioral of comparator\_tb is

signal at, bt : std\_logic\_vector(1 downto 0);

signal eqt : std\_logic;

begin

--replace filename below with each design file

p0 : entity work.comparator\_data(Dataflow) port map(at, bt, eqt);

process

begin

--sequence below was altered slightly between each run

wait for 100ns;

at <= "00";

bt <= "00";

wait for 100ns;

at <= "10";

bt <= "11";

wait for 100ns;

at <= "11";

bt <= "11";

wait for 100ns;

at <= "01";

bt <= "01";

wait for 100ns;

at <= "10";

bt <= "00";

wait for 100ns;

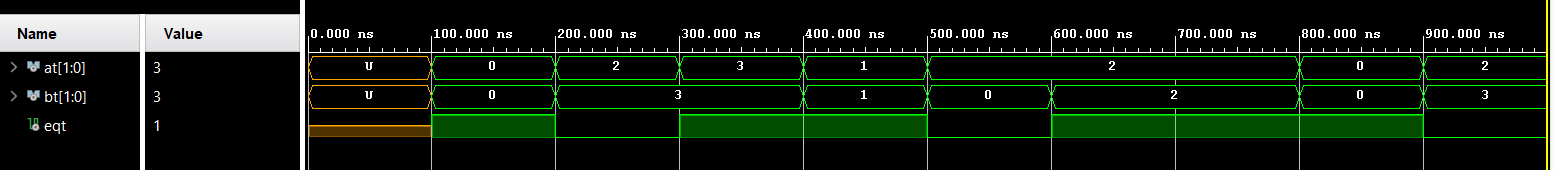
at <= "10";

bt <= "10";

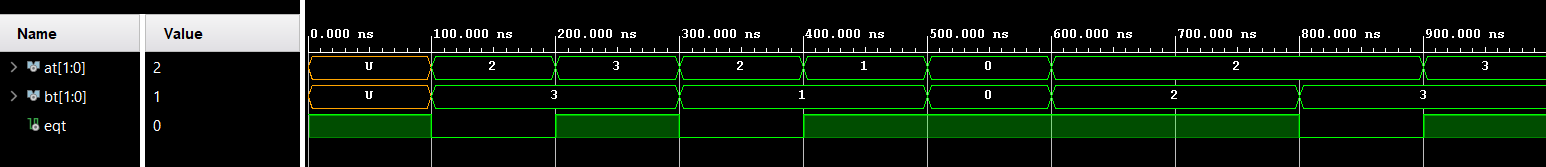
wait for 100ns;

end process;

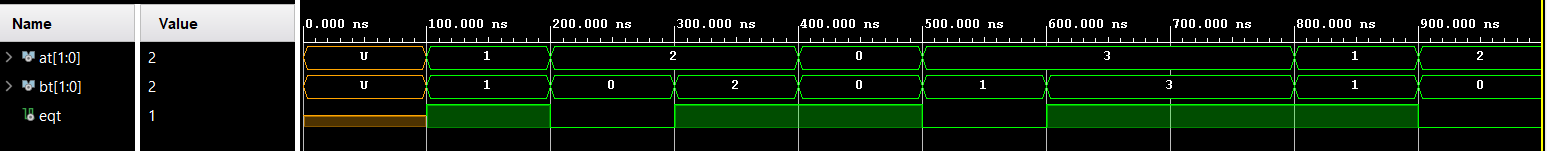
end Behavioral;



**Dataflow testbench**



**Behavioral testbench**



**Structural testbench**