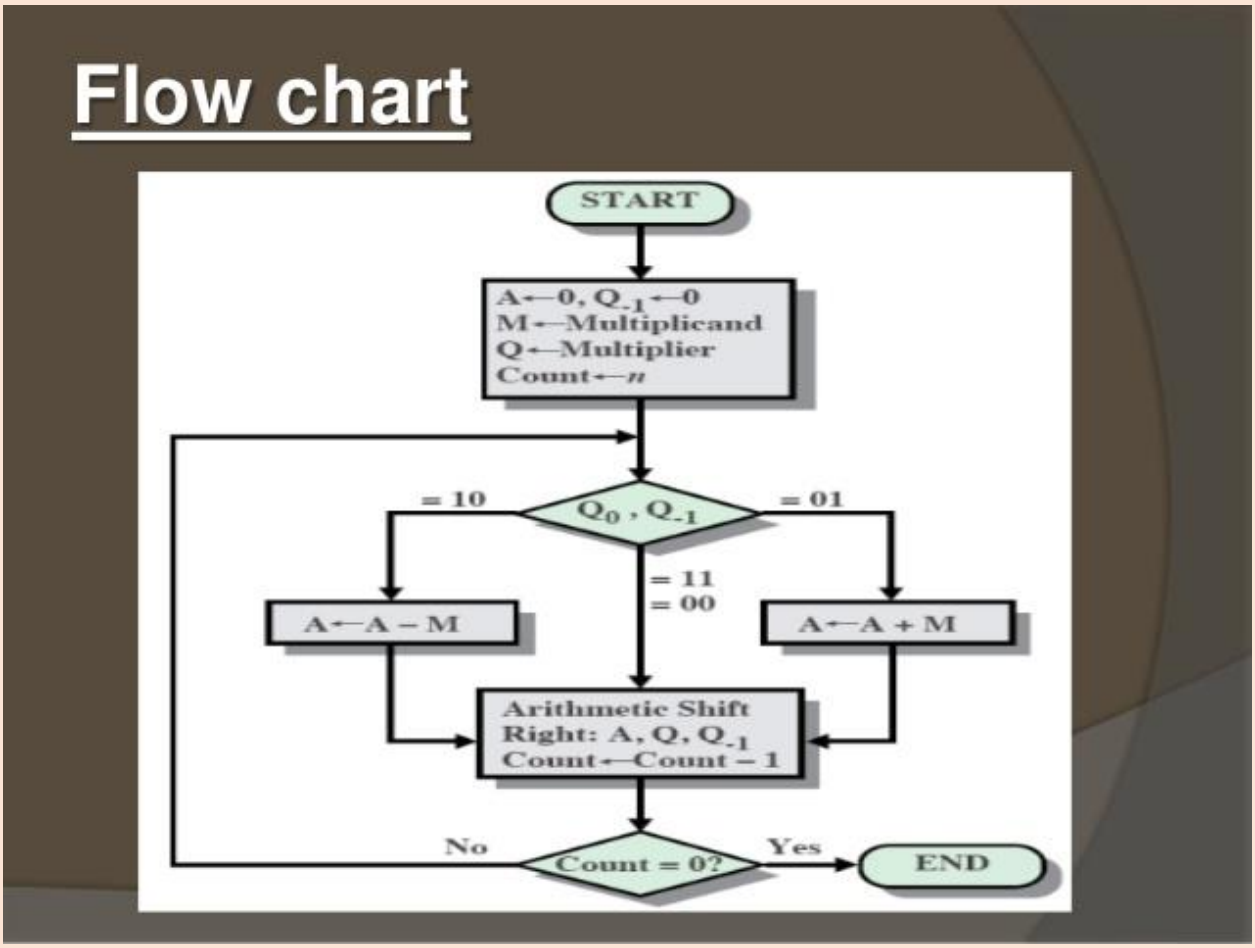
به نام خدا

با توجه به الگوریتم زیر:



کد این الگوریتم (بوث) را می نویسیم:

```
----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date:    17:11:40 08/22/2020

-- Design Name:
```

```
-- Module Name:    booth - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
```

```
USE ieee.std_logic_signed.ALL;

entity booth is

        GENERIC ( n : integer := 64
                                );

        port ( m : in STD_LOGIC_VECTOR(n - 1 downto 0);
             r : in STD_LOGIC_VECTOR(n - 1 downto 0);
                        result : out STD_LOGIC_VECTOR(n + n - 1 downto 0)
                       );

end booth;


architecture behavior of booth is

begin

        process(m, r)

                constant NUMBER : std_logic_vector(n-1 downto 0) := (others => '0');
```

```
        variable a, s, p : std_logic_vector(n+n+1 downto 0);
        variable mn      : std_logic_vector(n-1 DOWNTO 0);

begin

        a := (others => '0');
        s := (others => '0');
        p := (others => '0');

        if (m /= NUMBER and r /= NUMBER) then

                a(n+n downto n+1) := m;
                a(n+n+1) := m(n-1);

                mn := (not m) + 1;

                s(n+n downto n+1) := mn;
                s(n+n+1) := not(m(n-1));

                p(n downto 1) := r;

                for i in 1 to n loop

                        if (p(1 downto 0) = "01") then
```

```
                p := p + a;
            elsif (p(1 downto 0) = "10") then
                p := p + s;
            end if;

            -- Shift Right Arithmetic
            p(n+n downto 0) := p(n+n+1 downto 1);

        end loop;

    end if;

    result <= p(n+n downto 1);

end process;

end behavior;
```

حال برای تست کردن این کد، test bench ای برای آن تعریف می کنیم و داریم:

```
----------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date:   19:54:23 08/22/2020
```

```
-- Design Name:
-- Module Name:   F:/Xilinx_ISE/Code/booth/t_booth.vhd
-- Project Name:  booth
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: booth
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-- Notes:
-- This testbench has been automatically generated using types std_logic and
-- std_logic_vector for the ports of the unit under test.  Xilinx recommends
-- that these types always be used for the top-level I/O of a design in order
-- to guaranatee that the testbench will bind correctly to the post-implementation
-- simulation model.
--------------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
```

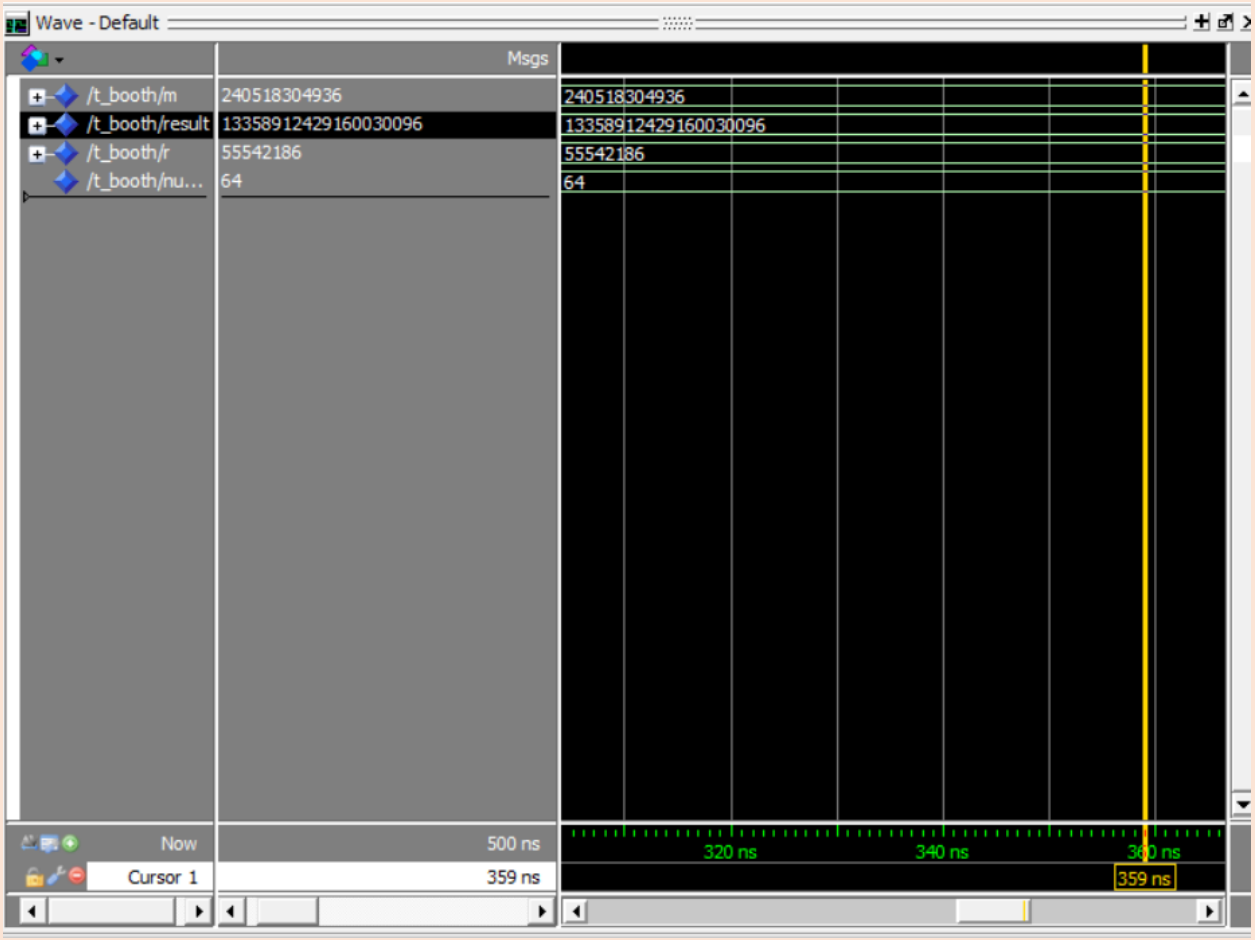
```
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY t_booth IS
END t_booth;

ARCHITECTURE behavior OF t_booth IS

        constant number: integer:= 64;

   -- Component Declaration for the Unit Under Test (UUT)

   COMPONENT booth
   PORT(
        m : IN  std_logic_vector(number-1 downto 0);
        r : IN  std_logic_vector(number-1 downto 0);
        result : OUT  std_logic_vector(number+number-1 downto 0)
       );
   END COMPONENT;


  --Inputs
```

```
  signal m : std_logic_vector(number-1 downto 0) := (others => '0');
  signal r : std_logic_vector(number-1 downto 0) := (others => '0');

        --Outputs
  signal result : std_logic_vector(number+number-1 downto 0);

BEGIN

        -- Instantiate the Unit Under Test (UUT)
  uut: booth PORT MAP (
       m => m,
       r => r,
       result => result
      );

        process

        begin

                m <=
"00000000000000000000000000011100000000000000000100001010010101000"
;
                r <=
"00000000000000000000000000000000000000011010011111000000110101010"
;
                wait for 100 ns;
```

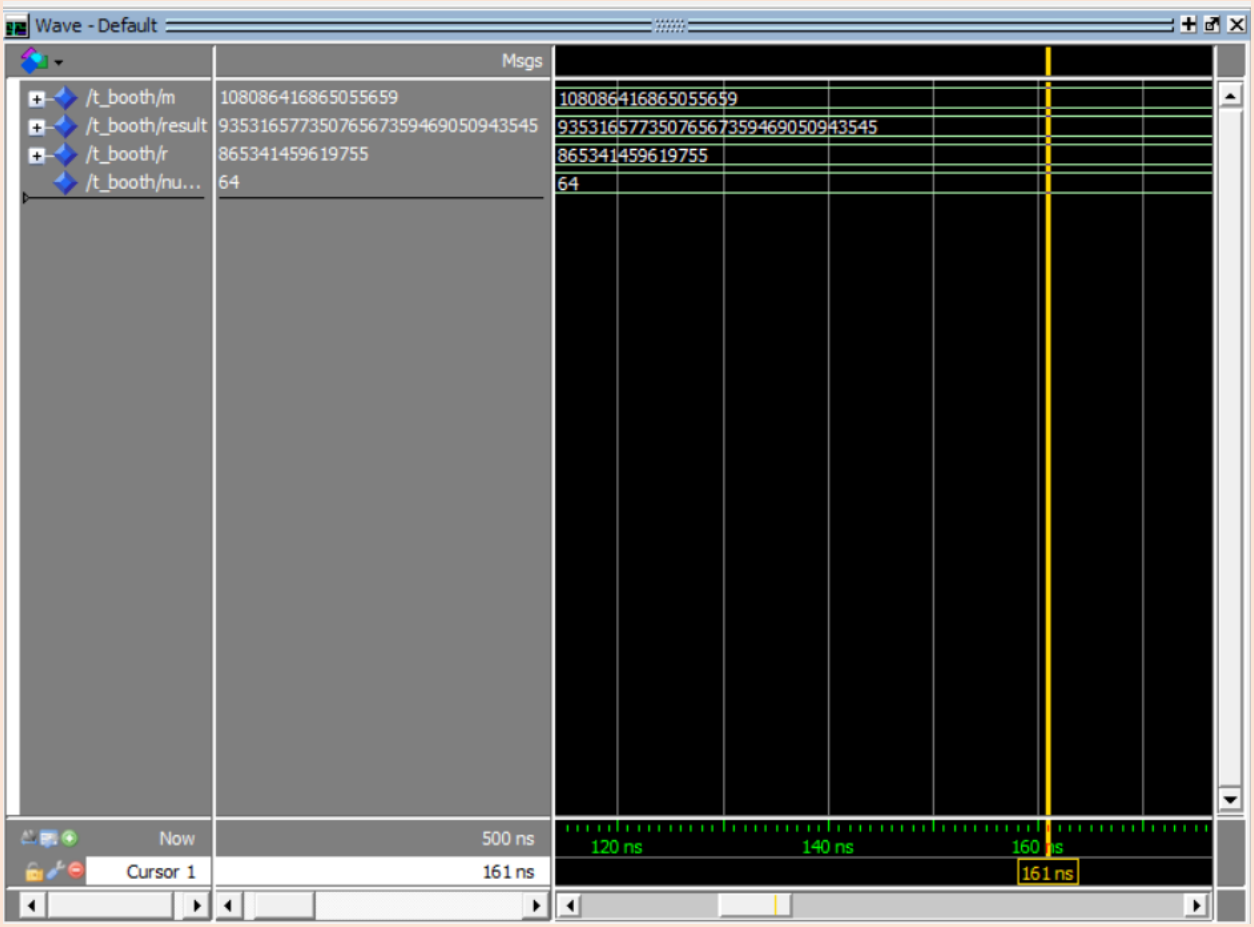
```
		m <=
"00000001100000000000000000000011000000010010010010101001110101011"
;
		r <=
"00000000000000011000100110000011000000010010011110101111110101011"
;
		wait for 100 ns;


		m <=
"00000000000000011000011000000110000000100100111101011111110101011"
;
		r <=
"00000000000000000000000000000011010010010010010010101111110101011"
;
		wait for 100 ns;


	end process;


END ARCHITECTURE;
```

پس از simulate کردن این testbench داریم:

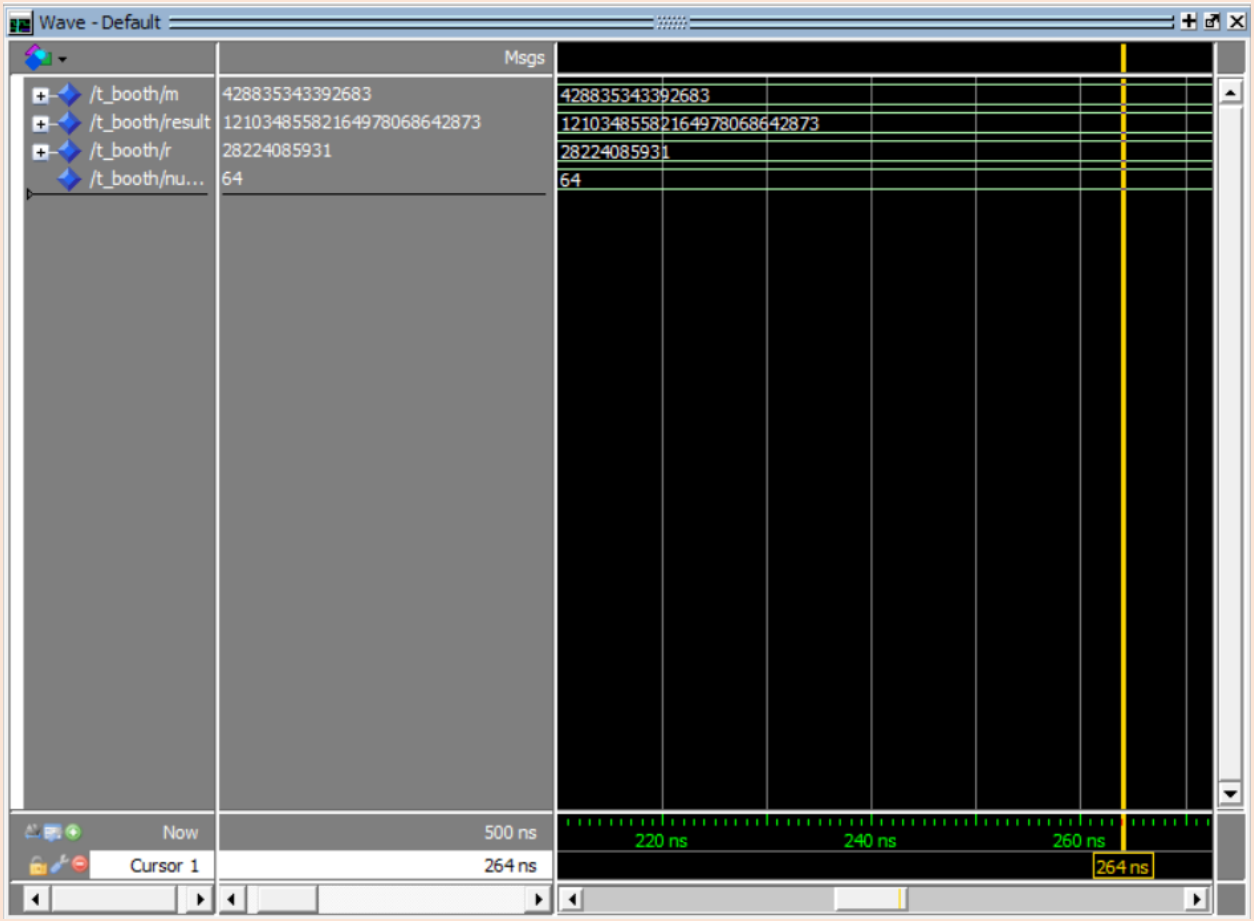
برای نمونه ی اول:



برای نمونه ی دوم:



برای نمونه ی سوم:



توجه کنید که در مثال های بالا اعداد را در حالت decimal یا همان ده دهی قرار داده ایم وگرنه اصل اعداد به صورت دو-دویی بوده اند.