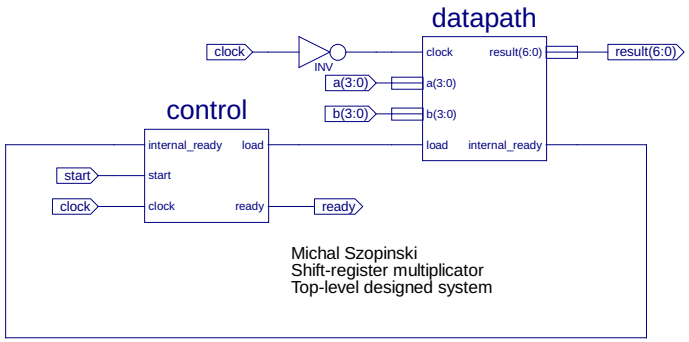
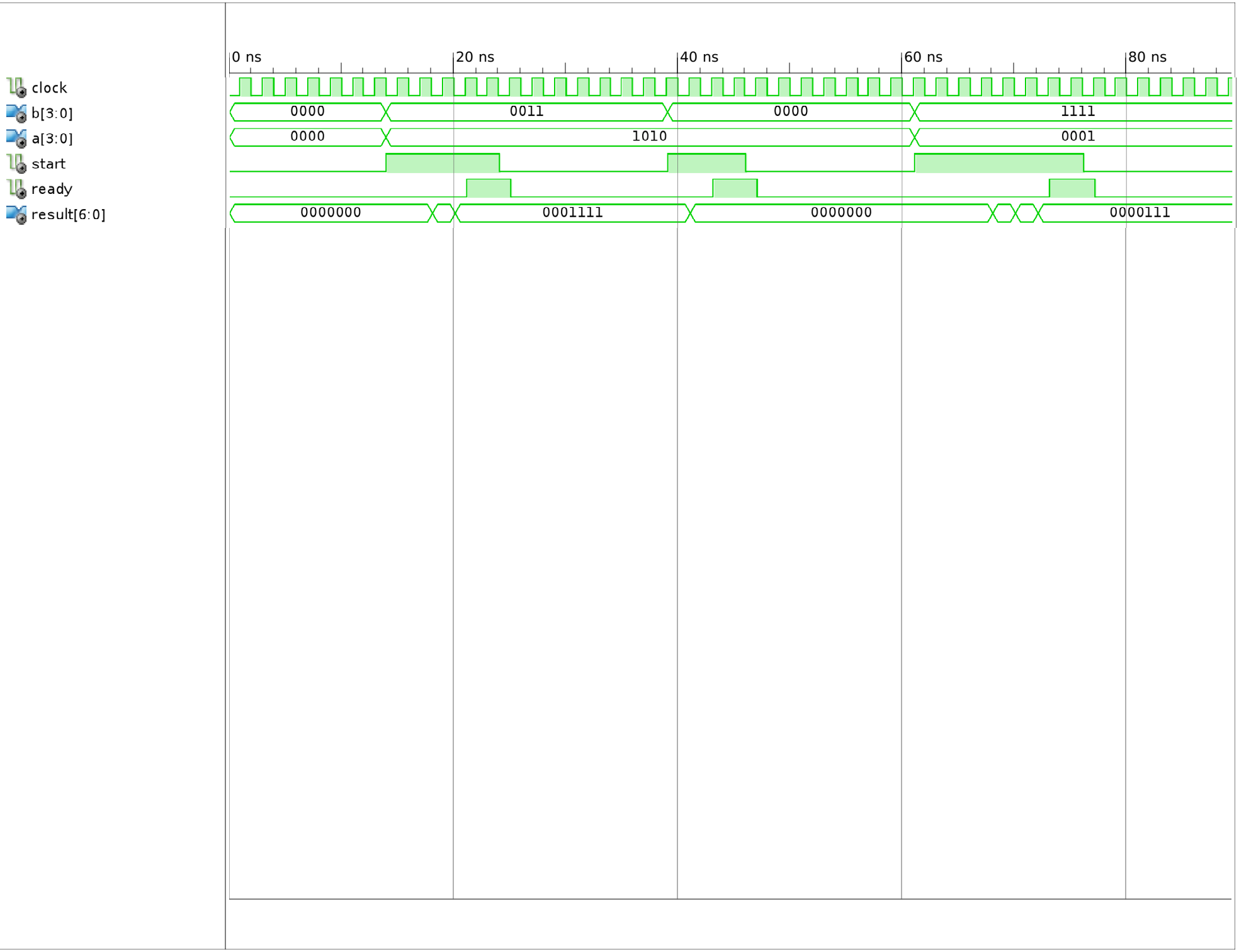
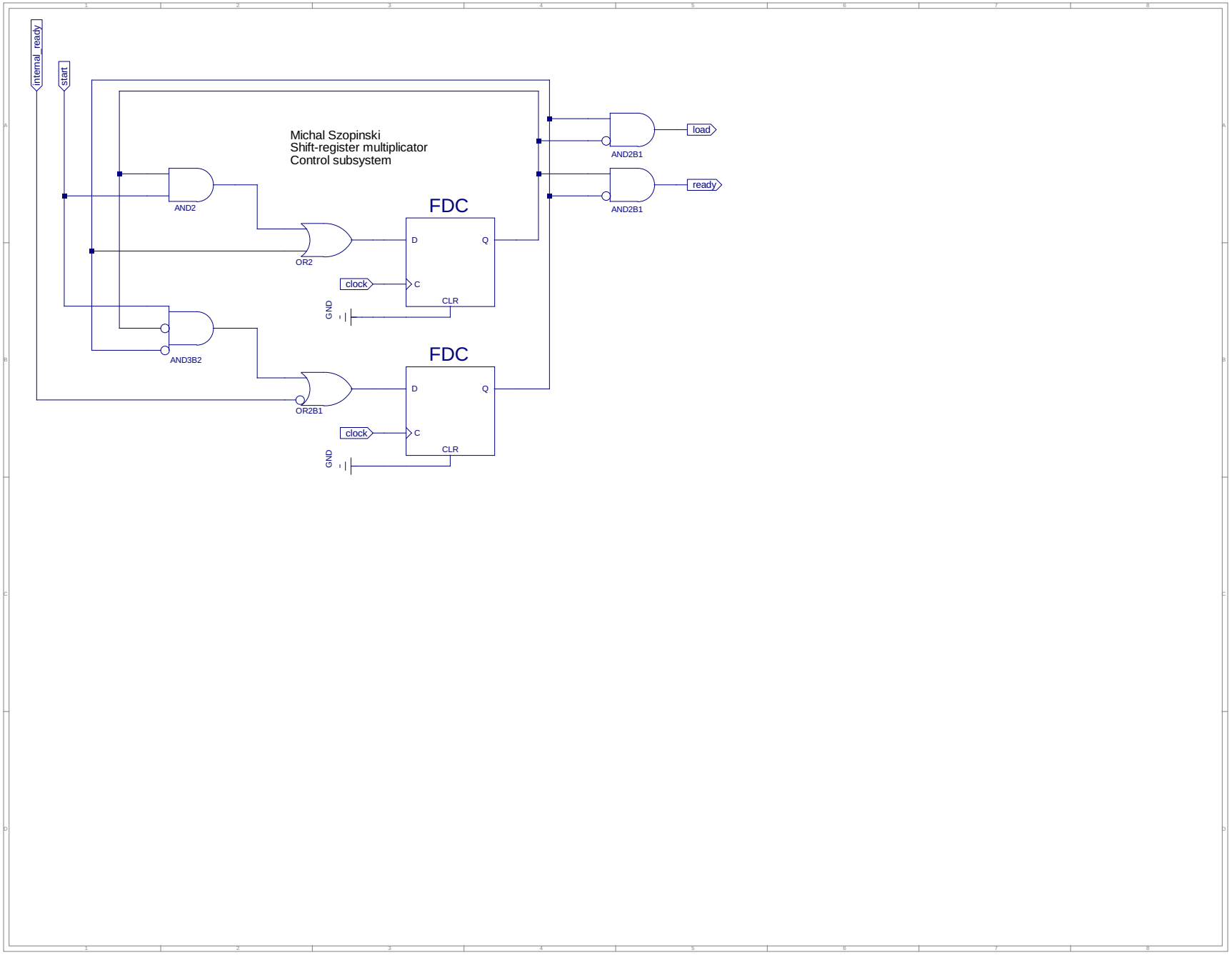
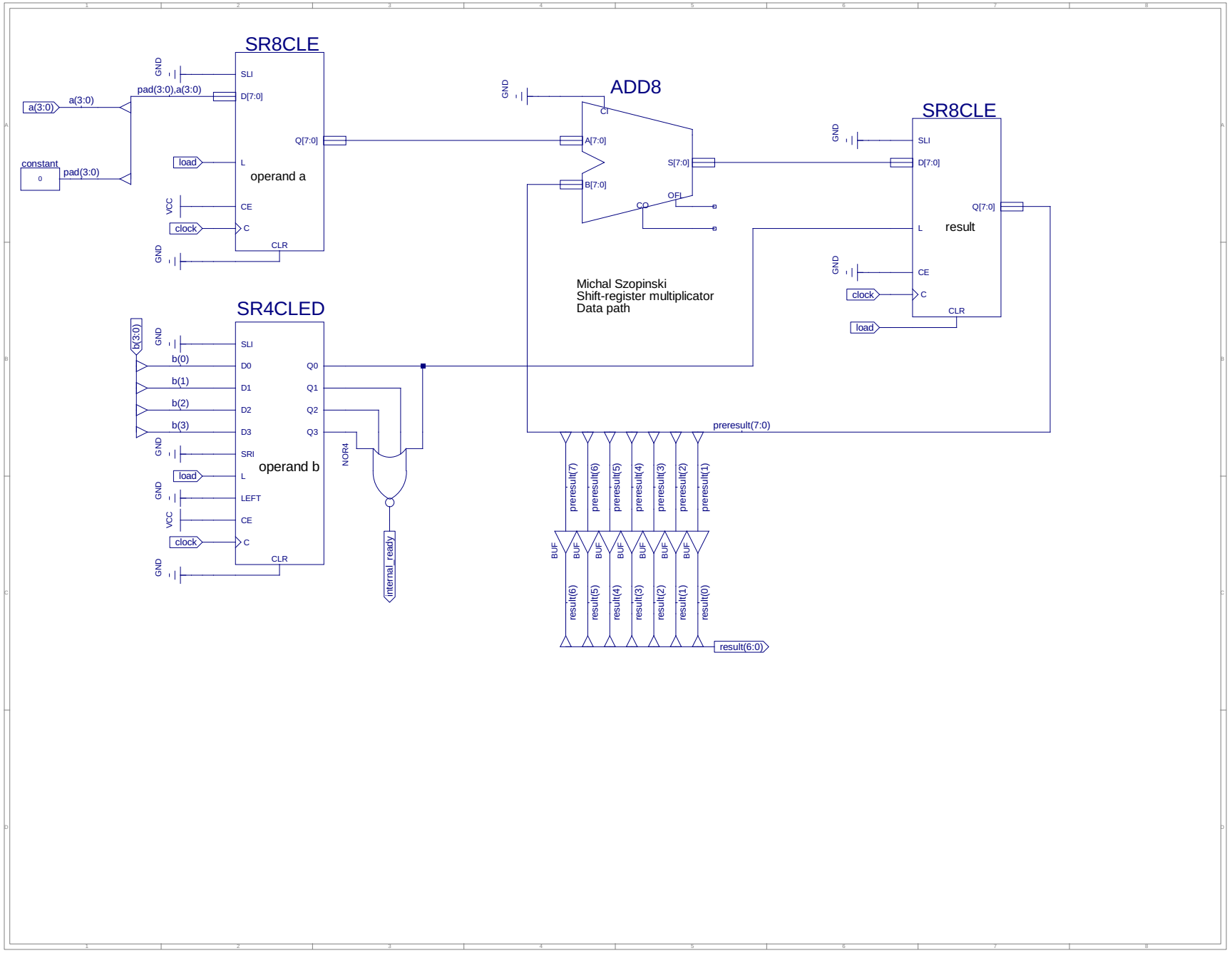

datapath
clock
INV
clock
result(6:0)
result(6:0)
a(3:0)
a(3:0)
b(3:0)
b(3:0)
control
internal_ready
load
load
internal_ready
start
start
clock
clock
ready
ready
Michal Szopinski
Shift-register multiplicator
Top-level designed system

```vhdl
-- Vhdl test bench created from schematic /home/mszopinski/Desktop/labko4/designedsystem.sch - Fri Dec 13 17:45:32 2019
--
-- Notes:
-- 1) This testbench template 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 guarantee that the testbench will bind
-- correctly to the timing (post-route) simulation model.
-- 2) To use this template as your testbench, change the filename to any
-- name of your choice with the extension .vhd, and use the "Source->Add"
-- menu in Project Navigator to import the testbench. Then
-- edit the user defined section below, adding code to generate the
-- stimulus for your design.
--
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;
ENTITY designedsystem_designedsystem_sch_tb IS
END designedsystem_designedsystem_sch_tb;
ARCHITECTURE behavioral OF designedsystem_designedsystem_sch_tb IS

   COMPONENT designedsystem
   PORT( clock  :   IN  STD_LOGIC;
         b :   IN  STD_LOGIC_VECTOR (3 DOWNTO 0);
         a :   IN  STD_LOGIC_VECTOR (3 DOWNTO 0);
         start :   IN  STD_LOGIC;
         ready :   OUT STD_LOGIC;
         result    :   OUT STD_LOGIC_VECTOR (6 DOWNTO 0));
   END COMPONENT;

   SIGNAL clock :   STD_LOGIC := '0'; -- SET DEFAULT VALUES
   SIGNAL b :   STD_LOGIC_VECTOR (3 DOWNTO 0) := "0000";
   SIGNAL a :   STD_LOGIC_VECTOR (3 DOWNTO 0) := "0000";
   SIGNAL start :   STD_LOGIC := '0';
   SIGNAL ready :   STD_LOGIC;
   SIGNAL result    :   STD_LOGIC_VECTOR (6 DOWNTO 0);

BEGIN

   UUT: designedsystem PORT MAP(
        clock => clock,
        b => b,
        a => a,
        start => start,
        ready => ready,
        result => result
   );

    clock <= not clock after 1 ns; -- SET CLOCK

-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN
        wait for 14 ns;

        -- multiply 10 by 3 and divide by 2
        -- expected result: 15 (0001111)
        a <= "1010";
        b <= "0011";
        start <= '1';
        wait until ready = '1';
        wait for 3 ns;
        start <= '0';

        wait for 15 ns;

        -- multiply 10 by 0 and divide by 2
        -- expected result: 0 (0, instantaneous result)
        a <= "1010";
        b <= "0000";
        start <= '1';
        wait until ready = '1';
        wait for 3 ns;
        start <= '0';

        wait for 15 ns;

        -- multiply 1 by 15 and divide by 2
        -- expected result: 7 (0111, maximum calculation length)
        a <= "0001";
        b <= "1111";
        start <= '1';
        wait until ready = '1';
        wait for 3 ns;
        start <= '0';

        WAIT; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section ***

END;
```


0 ns
20 ns
40 ns
60 ns
80 ns
clock
b[3:0]
a[3:0]
start
ready
result[6:0]
0000
0011
0000
1111
0000
1010
0001
0000000
0001111
0000000
0000111


internal_ready
start
Michal Szopinski
Shift-register multiplicator
Control subsystem
AND2
OR2
FDC
D
Q
clock
C
CLR
GND
AND3B2
OR2B1
FDC
D
Q
clock
C
CLR
GND
AND2B1
load
AND2B1
ready

```vhdl
-- Vhdl test bench created from schematic /home/mszopinski/Desktop/labko4/control.sch - Fri Dec 13 21:03:33 2019
--
-- Notes:
-- 1) This testbench template 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 guarantee that the testbench will bind
-- correctly to the timing (post-route) simulation model.
-- 2) To use this template as your testbench, change the filename to any
-- name of your choice with the extension .vhd, and use the "Source->Add"
-- menu in Project Navigator to import the testbench. Then
-- edit the user defined section below, adding code to generate the
-- stimulus for your design.
--
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;
ENTITY control_control_sch_tb IS
END control_control_sch_tb;
ARCHITECTURE behavioral OF control_control_sch_tb IS

   COMPONENT control
   PORT( internal_ready :   IN  STD_LOGIC;
         start :   IN  STD_LOGIC;
         clock :   IN  STD_LOGIC;
         load  :   OUT STD_LOGIC;
         ready :   OUT STD_LOGIC);
   END COMPONENT;

   SIGNAL internal_ready    :   STD_LOGIC := '1'; -- SET DEFAULT VALUES
   SIGNAL start :   STD_LOGIC := '0';
   SIGNAL clock :   STD_LOGIC := '0';
   SIGNAL load  :   STD_LOGIC;
   SIGNAL ready :   STD_LOGIC;

BEGIN

   UUT: control PORT MAP(
        internal_ready => internal_ready,
        start => start,
        clock => clock,
        load => load,
        ready => ready
   );

    clock <= not clock after 1 ns; -- SET CLOCK

-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN
      wait for 20 ns;

        -- multiplication by non 0
        start <= '1';
        wait for 1 ns;
        internal_ready <= '0';
        wait for 5 ns;
        internal_ready <= '1';
        wait for 5 ns;
        start <= '0';
        wait for 20 ns;

        -- multiplication by 0
        start <= '1';
        wait for 11 ns;
        start <= '0';
        wait for 20 ns;

        WAIT; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section ***

END;
```


SR8CLE
GND
SLI
pad(3:0),a(3:0)
D[7:0]
a(3:0)
a(3:0)
Q[7:0]
load
L
operand a
constant
0
pad(3:0)
VCC
CE
clock
C
CLR
GND
ADD8
GND
CI
A[7:0]
S[7:0]
B[7:0]
OFL
CO
SR8CLE
GND
SLI
D[7:0]
Q[7:0]
L
result
GND
CE
clock
C
CLR
load
Michal Szopinski
Shift-register multiplicator
Data path
SR4CLED
b(3:0)
GND
SLI
b(0)
D0
Q0
b(1)
D1
Q1
b(2)
D2
Q2
b(3)
D3
Q3
GND
SRI
load
L
operand b
GND
LEFT
VCC
CE
clock
C
CLR
GND
NOR4
internal_ready
preresult(7:0)
preresult(7)
preresult(6)
preresult(5)
preresult(4)
preresult(3)
preresult(2)
preresult(1)
BUF
result(6)
result(5)
result(4)
result(3)
result(2)
result(1)
result(0)
result(6:0)

```vhdl
-- Vhdl test bench created from schematic /home/mszopinski/Desktop/labko4/datapath.sch - Fri Dec 13 16:20:58 2019
--
-- Notes:
-- 1) This testbench template 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 guarantee that the testbench will bind
-- correctly to the timing (post-route) simulation model.
-- 2) To use this template as your testbench, change the filename to any
-- name of your choice with the extension .vhd, and use the "Source->Add"
-- menu in Project Navigator to import the testbench. Then
-- edit the user defined section below, adding code to generate the
-- stimulus for your design.
--
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;
ENTITY datapath_datapath_sch_tb IS
END datapath_datapath_sch_tb;
ARCHITECTURE behavioral OF datapath_datapath_sch_tb IS

   COMPONENT datapath
   PORT( clock  :   IN  STD_LOGIC;
         a :   IN  STD_LOGIC_VECTOR (3 DOWNTO 0);
         b :   IN  STD_LOGIC_VECTOR (3 DOWNTO 0);
         internal_ready    :   OUT STD_LOGIC;
         load  :   IN  STD_LOGIC;
         result    :   OUT STD_LOGIC_VECTOR (6 DOWNTO 0));
   END COMPONENT;

   SIGNAL clock :   STD_LOGIC := '0'; -- SET DEFAULT VALUES
   SIGNAL a :   STD_LOGIC_VECTOR (3 DOWNTO 0) := "0000";
   SIGNAL b :   STD_LOGIC_VECTOR (3 DOWNTO 0) := "0000";
   SIGNAL internal_ready    :   STD_LOGIC;
   SIGNAL load  :   STD_LOGIC := '0';
   SIGNAL result    :   STD_LOGIC_VECTOR (6 DOWNTO 0);

BEGIN

   UUT: datapath PORT MAP(
        clock => clock,
        a => a,
        b => b,
        internal_ready => internal_ready,
        load => load,
        result => result
   );

    clock <= not clock after 1 ns; -- SET CLOCK

-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN
        wait for 14 ns;

        -- multiply 10 by 3 and divide by 2
        -- expected result: 15 (0001111)
        a <= "1010";
        b <= "0011";
        load <= '1';

        -- wait one clock cycle to load registers
        wait for 1 ns;
        load <= '0';

      WAIT; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section ***

END;
```