Termin zajęć Wtorek NP 7:30 – 11:00	Układy cyfrowe i systemy wbudowane	
Osoby wykonujące ćwiczenie: Jakub Suski 264028, Adam Czekalski 264488		**Grupa:** **D**
Tytuł ćwiczenia: Układy kombinacyjne i sekwencyjne w VHDL-u		**Laboratorium nr:** **5**
Data wykonania ćwiczenia	21.11.2023	**Ocena:**
Data oddania sprawozdania	5.12.2023	

Na zajęciach laboratoryjnych wykorzystano syntezę układów z poprzednich zajęć. Tym razem jednak układy te skonstruowano w środowisku Xilinx nie za pomocą „układania” poszczególnych elementów na schemacie w pliku z rozszerzeniem .sch, a bezpośrednio kodując w języku VHDL poprzez utworzenie nowego modułu.

1. Implementacja funkcji logicznej

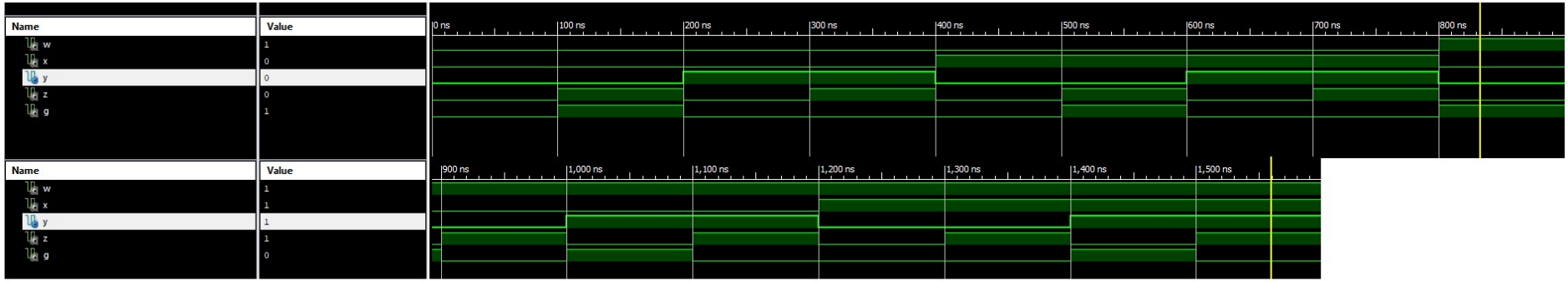
$$G(w,x,y,z) = \Pi(0, 2, 3, 4, 6, 7, 9, 11, 12, 13, 15)$$

$$G(w,x,y,z) = \sum(1,5,8,10,14)$$

Implementacja za pomocą funkcji boolowskich:

```
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22
23  -- Uncomment the following library declaration if using
24  -- arithmetic functions with Signed or Unsigned values
25  --use IEEE.NUMERIC_STD.ALL;
26
27  -- Uncomment the following library declaration if instantiating
28  -- any Xilinx primitives in this code.
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity sch1 is
33      Port ( w : in  STD_LOGIC;
34             x : in  STD_LOGIC;
35             y : in  STD_LOGIC;
36             z : in  STD_LOGIC;
37             G : out  STD_LOGIC);
38  end sch1;
39
40  architecture DATAFLOW of sch1 is
41
42  begin
43
44  G <= (not w and not y and z) or (w and not x and not z) or (w and y and not z);
45
46  end DATAFLOW;
47
```

Symulacja:


Name
Value
w
x
y
z
g
1
0
0
0
1
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
Name
Value
w
x
y
z
g
1
1
1
1
0
900 ns
1,000 ns
1,100 ns
1,200 ns
1,300 ns
1,400 ns
1,500 ns

Plik z pobudzeniami testowymi:

```
ARCHITECTURE behavior OF testl IS

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

    COMPONENT schl
    PORT(
         w : IN  std_logic;
         x : IN  std_logic;
         y : IN  std_logic;
         z : IN  std_logic;
         G : OUT  std_logic
        );
    END COMPONENT;


    --Inputs
    signal w : std_logic := '0';
    signal x : std_logic := '0';
    signal y : std_logic := '0';
    signal z : std_logic := '0';

     --Outputs
    signal G : std_logic;
    -- No clocks detected in port list. Replace <clock> below with
    -- appropriate port name

BEGIN

     -- Instantiate the Unit Under Test (UUT)
    uut: schl PORT MAP (
          w => w,
          x => x,
          y => y,
          z => z,
          G => G
        );

    -- Clock process definitions


    -- Stimulus process
    stim_proc: process
    begin
       w <= '0', '1' after 800 ns;
         x <= '0', '1' after 400 ns, '0' after 800 ns, '1' after 1200 ns;
         y <= '0', '1' after 200 ns, '0' after 400 ns, '1' after 600 ns, '0' after 800 ns, '1' after 1000 ns, '0' after 1200 ns, '1' after 1400 ns;
         z <= '0', '1' after 100 ns, '0' after 200 ns, '1' after 300 ns, '0' after 400 ns,
         '1' after 500 ns, '0' after 600 ns, '1' after 700 ns, '0' after 800 ns,
         '1' after 900 ns, '0' after 1000 ns, '1' after 1100 ns, '0' after 1200 ns,
         '1' after 1300 ns, '0' after 1400 ns, '1' after 1500 ns;
       wait;
    end process;
```

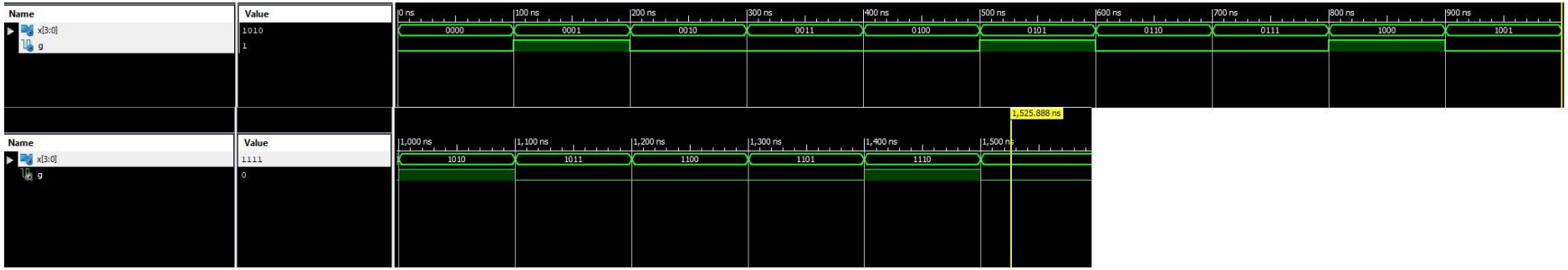
Plik .ucf:

```
# Keys
NET "z" LOC = "P42";
NET "y" LOC = "P40";
NET "x" LOC = "P43";
NET "w" LOC = "P38";
#NET "Key<4>" LOC = "P37";
#NET "Key<5>" LOC = "P36";   # shared with ROT_A
#NET "Key<6>" LOC = "P24";   # shared with ROT_B
#NET "Key<7>" LOC = "P39";   # GSR

# LEDS
NET "G"  LOC = "P35";
```

Implementacja za pomocą zapisu tablicowego:

```
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;

entity sch1tabP is
    Port ( X : in  STD_LOGIC_VECTOR (3 downto 0);
           G : out  STD_LOGIC);
end sch1tabP;

architecture DATAFLOW of sch1tabP is

begin
with X select
    G <= '1' when "0001" | "0101" | "1000" | "1010" | "1110", '0' when others;


end DATAFLOW;
```

Symulacja:


Name
Value
x[3:0]
g
1010
1
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1,525.888 ns
Name
Value
x[3:0]
g
1111
0
1,000 ns
1,100 ns
1,200 ns
1,300 ns
1,400 ns
1,500 ns
1010
1011
1100
1101
1110

Plik z pobudzeniami testowymi:

```
28  LIBRARY ieee;
29  USE ieee.std_logic_1164.ALL;
30
31  -- Uncomment the following library declaration if using
32  -- arithmetic functions with Signed or Unsigned values
33  --USE ieee.numeric_std.ALL;
34
35  ENTITY testltab IS
36  END testltab;
37
38  ARCHITECTURE behavior OF testltab IS
39
40      -- Component Declaration for the Unit Under Test (UUT)
41
42      COMPONENT schltabP
43      PORT(
44           X : IN  std_logic_vector(3 downto 0);
45           G : OUT  std_logic
46          );
47      END COMPONENT;
48
49
50     --Inputs
51     signal X : std_logic_vector(3 downto 0) := (others => '0');
52
53      --Outputs
54     signal G : std_logic;
55     -- No clocks detected in port list. Replace <clock> below with
56     -- appropriate port name
57
58  BEGIN
59
60      -- Instantiate the Unit Under Test (UUT)
61     uut: schltabP PORT MAP (
62           X => X,
63           G => G
64          );
65
66     -- Clock process definitions
67
68     -- Stimulus process
69     stim_proc: process
70     begin
71          X <= "0000", "0001" after 100 ns, "0010" after 200 ns, "0011" after 300 ns, "0100" after 400 ns,
72          "0101" after 500 ns, "0110" after 600 ns, "0111" after 700 ns, "1000" after 800 ns,
73          "1001" after 900 ns, "1010" after 1000 ns, "1011" after 1100 ns, "1100" after 1200 ns,
74          "1101" after 1300 ns, "1110" after 1400 ns, "1111" after 1500 ns;
75        wait;
76     end process;
77
78  END;
```

Plik .ucf:

```
 5  # Keys
 6  NET "X(0)" LOC = "P42";
 7  NET "X(1)" LOC = "P40";
 8  NET "X(2)" LOC = "P43";
 9  NET "X(3)" LOC = "P38";
10  #NET "Key<4>" LOC = "P37";
11  #NET "Key<5>" LOC = "P36";   # shared with ROT_A
12  #NET "Key<6>" LOC = "P24";   # shared with ROT_B
13  #NET "Key<7>" LOC = "P39";   # GSR
14
15  # LEDS
16  NET "G"  LOC = "P35";
```

Na płytce CPLD lampka LED paliła się gdy funkcja przyjmowała wartość „0”, zaś gasła gdy miała wartość „1”. Wartość funkcji logicznej wprowadzano za pomocą przycisków K3-K0.

2. Implementacja układu translatora kodu

Implementacja za pomocą funkcji boolowskich:

```
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22
23  -- Uncomment the following library declaration if using
24  -- arithmetic functions with Signed or Unsigned values
25  --use IEEE.NUMERIC_STD.ALL;
26
27  -- Uncomment the following library declaration if instantiating
28  -- any Xilinx primitives in this code.
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity sch2bool is
33      Port ( Q : in  STD_LOGIC_VECTOR (3 downto 0);
34             Y : out  STD_LOGIC_VECTOR (3 downto 0));
35  end sch2bool;
36
37  architecture Behavioral of sch2bool is
38
39  begin
40  Y(3) <= (Q(2) and not Q(1)) or (Q(2) and Q(0));
41  Y(2) <= (Q(2) and not Q(1)) or (Q(2) and not Q(0));
42  Y(1) <= Q(3) or (not Q(2) and Q(1)) or (Q(1) and Q(0));
43  Y(0) <= (not Q(3) and Q(2) and not Q(1) and not Q(0)) or (not Q(2) and not Q(1) and Q(0))
44   or (not Q(2) and Q(1) and not Q(0)) or (Q(2) and Q(1) and Q(0)) or (Q(3) and Q(0));
45
46
47  end Behavioral;
```

Symulacja:


Name
Value
q[3:0]
y[3:0]
1010
0011
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
0000
0001
0011
0010
1101
1100
0100
1011
0010
0011
1,526.350 ns
Name
Value
q[3:0]
y[3:0]
1111
1011
1,000 ns
1,100 ns
1,200 ns
1,300 ns
1,400 ns
1,500 ns
1010
1011
1100
1101
1110
0011
1110
1111
0110

Plik testowy VHDL:

```
END test2bool;

ARCHITECTURE behavior OF test2bool IS

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

    COMPONENT sch2bool
    PORT(
         Q : IN  std_logic_vector(3 downto 0);
         Y : OUT  std_logic_vector(3 downto 0)
        );
    END COMPONENT;


   --Inputs
   signal Q : std_logic_vector(3 downto 0) := (others => '0');

   --Outputs
   signal Y : std_logic_vector(3 downto 0);
   -- No clocks detected in port list. Replace <clock> below with
   -- appropriate port name

BEGIN

   -- Instantiate the Unit Under Test (UUT)
   uut: sch2bool PORT MAP (
          Q => Q,
          Y => Y
        );


   -- Stimulus process
   stim_proc: process
   begin
      Q(0) <= '0', '1' after 100 ns, '0' after 200 ns, '1' after 300 ns, '0' after 400 ns,
      '1' after 500 ns, '0' after 600 ns  , '1' after 700 ns, '0' after 800 ns  ,
      '1' after 900 ns, '0' after 1000 ns  , '1' after 1100 ns, '0' after 1200 ns,
      '1' after 1300 ns, '0' after 1400 ns, '1' after 1500 ns, '0' after 1600 ns;
      Q(1) <= '0', '1' after 200 ns, '0' after 400 ns, '1' after 600 ns, '0' after 800 ns, '1' after 1000 ns, '0' after 1200 ns, '1' after 1400 ns, '0' after 1600 ns;
      Q(2) <= '0', '1' after 400 ns, '0' after 800 ns, '1' after 1200 ns, '0' after 1600 ns;
      Q(3) <= '0', '1' after 800 ns, '0' after 1600 ns;
   wait;
   end process;

END;
```

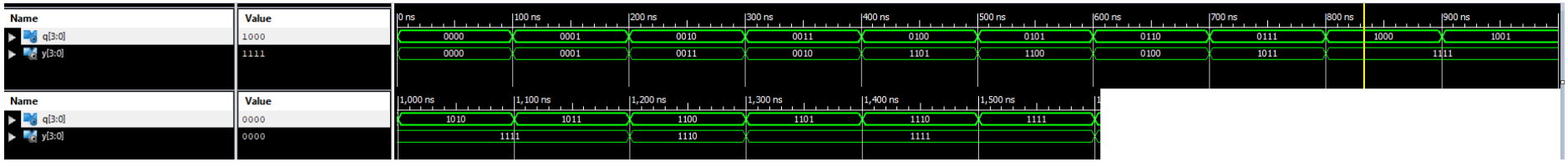
Plik .ucf:

```
# Keys
NET "Q(0)" LOC = "P42";
NET "Q(1)" LOC = "P40";
NET "Q(2)" LOC = "P43";
NET "Q(3)" LOC = "P38";
#NET "Key<4>" LOC = "P37";
#NET "Key<5>" LOC = "P36";  # shared with ROT_A
#NET "Key<6>" LOC = "P24";  # shared with ROT_B
#NET "Key<7>" LOC = "P39";  # GSR

# LEDS
NET "Y(0)"  LOC = "P35";
NET "Y(1)"  LOC = "P29";
NET "Y(2)"  LOC = "P33";
NET "Y(3)"  LOC = "P34";
```

Implementacja za pomocą zapisu tablicowego:

```
-- Module Name:    sch2tab - 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;

entity sch2tab is
    Port ( Q : in  STD_LOGIC_VECTOR (3 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end sch2tab;

architecture Behavioral of sch2tab is

begin
with Q select
Y(3) <= '0' when "0000" | "0001" | "0010" | "0011" | "0110", '1' when others;
with Q select
Y(2) <= '0' when "0000" | "0001" | "0010" | "0011" | "0111", '1' when others;
with Q select
Y(1) <= '0' when "0000" | "0001" | "0100" | "0101" | "0110", '1' when others;
with Q select
Y(0) <= '0' when "0000" | "0011" | "0101" | "0110" | "1100", '1' when others;

end Behavioral;
```

Symulacja:


Name
Value
q[3:0]
y[3:0]
1000
1111
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
0000
0001
0011
0010
1101
1100
0100
1011
1111
Name
Value
q[3:0]
y[3:0]
0000
0000
1,000 ns
1,100 ns
1,200 ns
1,300 ns
1,400 ns
1,500 ns
1010
1011
1100
1101
1110
1111
1111
1110
1111

Plik testowy VHDL:

```
36  END test2bool;
37
38  ARCHITECTURE behavior OF test2bool IS
39
40      -- Component Declaration for the Unit Under Test (UUT)
41
42      COMPONENT sch2bool
43      PORT(
44           Q : IN  std_logic_vector(3 downto 0);
45           Y : OUT  std_logic_vector(3 downto 0)
46          );
47      END COMPONENT;
48
49
50     --Inputs
51     signal Q : std_logic_vector(3 downto 0) := (others => '0');
52
53     --Outputs
54     signal Y : std_logic_vector(3 downto 0);
55     -- No clocks detected in port list. Replace <clock> below with
56     -- appropriate port name
57
58  BEGIN
59
60     -- Instantiate the Unit Under Test (UUT)
61     uut: sch2bool PORT MAP (
62            Q => Q,
63            Y => Y
64          );
65
66
67     -- Stimulus process
68     stim_proc: process
69     begin
70        Q(0) <= '0', '1' after 100 ns, '0' after 200 ns, '1' after 300 ns, '0' after 400 ns,
71        '1' after 500 ns, '0' after 600 ns  , '1' after 700 ns, '0' after 800 ns  ,
72        '1' after 900 ns, '0' after 1000 ns  , '1' after 1100 ns, '0' after 1200 ns,
73        '1' after 1300 ns, '0' after 1400 ns, '1' after 1500 ns, '0' after 1600 ns;
74        Q(1) <= '0', '1' after 200 ns, '0' after 400 ns, '1' after 600 ns, '0' after 800 ns, '1' after 1000 ns, '0' after 1200 ns, '1' after 1400 ns, '0' after 1600 ns;
75        Q(2) <= '0', '1' after 400 ns, '0' after 800 ns, '1' after 1200 ns, '0' after 1600 ns;
76        Q(3) <= '0', '1' after 800 ns, '0' after 1600 ns;
77     wait;
78     end process;
79
80  END;
```

Plik .ucf:

```
 5  # Keys
 6  NET "Q(0)" LOC = "P42";
 7  NET "Q(1)" LOC = "P40";
 8  NET "Q(2)" LOC = "P43";
 9  NET "Q(3)" LOC = "P38";
10  #NET "Key<4>" LOC = "P37";
11  #NET "Key<5>" LOC = "P36";   # shared with ROT_A
12  #NET "Key<6>" LOC = "P24";   # shared with ROT_B
13  #NET "Key<7>" LOC = "P39";   # GSR
14
15  # LEDS
16  NET "Y(0)"  LOC = "P35";
17  NET "Y(1)"  LOC = "P29";
18  NET "Y(2)"  LOC = "P33";
19  NET "Y(3)"  LOC = "P34";
```

Za pomocą guzików K3-K0 wprowadzamy liczbę w NKB, jej przekodowanie widoczne jest na diodach.

3. Detektor sekwencji: 11100 w postaci automatu Mealy'ego

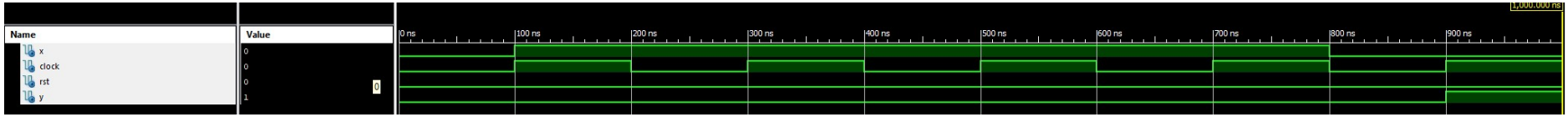
Plik VHDL:

```
entity sch3 is
    Port ( X : in  STD_LOGIC;
           Y : out  STD_LOGIC;
           clock : in  STD_LOGIC;
           RST : in  STD_LOGIC);
end sch3;

architecture Behavioral of sch3 is

type state_type is (q0, q1, q2, q3, q4, q5);
signal state, next_state : state_type;

begin
    process1 : process(clock)
    begin
        if rising_edge(clock) then
            if RST = '1' then
                state <= q0;
            else
                state <= next_state;
            end if;
        end if;
    end process process1;

    process2 : process(clock)
    begin
        next_state <= state;

        case state is

            when q0 =>
                if X = '1' then
                    next_state <= q1;
```

```
65                  else
66                      next_state <= q0;
67                  end if;
68
69              when q1 =>
70                  if X = '1' then
71                      next_state <= q2;
72                  else
73                      next_state <= q0;
74                  end if;
75
76              when q2 =>
77                  if X = '1' then
78                      next_state <= q3;
79                  else
80                      next_state <= q0;
81                  end if;
82
83              when q3 =>
84                  if X = '1' then
85                      next_state <= q3;
86                  else
87                      next_state <= q4;
88                  end if;
89
90              when q4 =>
91                  if X = '1' then
92                      next_state <= q1;
93                  else
94                      next_state <= q5;
95                  end if;
96
97              when q5 =>
98                  if X = '1' then
99                      next state <= q1;
```

```
100                 else
101                     next_state <= q0;
102                 end if;
103
104             end case;
105         end process process2;
106
107         Y <= '1' when state = q4 and X = '0'
108             else '0';
109
110
111
112     end Behavioral;
```

Symulacja:


Name
x
clock
rst
y
Value
0
0
0
1
0
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns

Plik testowy VHDL:

```
34
35  ENTITY test33 IS
36  END test33;
37
38  ARCHITECTURE behavior OF test33 IS
39
40      -- Component Declaration for the Unit Under Test (UUT)
41
42      COMPONENT sch3
43      PORT(
44           X : IN  std_logic;
45           Y : OUT  std_logic;
46           clock : IN  std_logic;
47           RST : IN  std_logic
48          );
49      END COMPONENT;
50
51
52     --Inputs
53     signal X : std_logic := '0';
54     signal clock : std_logic := '0';
55     signal RST : std_logic := '0';
56
57     --Outputs
58     signal Y : std_logic;
59
60
61  BEGIN
62
63     -- Instantiate the Unit Under Test (UUT)
64     uut: sch3 PORT MAP (
65            X => X,
66            Y => Y,
67            clock => clock,
68            RST => RST
69          );
70
71        clock <= not clock after 100 ns;
72
73        -- insert stimulus here
74        X <= '0', '1' after 100 ns, '0' after 800 ns;
75
76  END;
77
```

Plik .ucf:

```
# Clocks
NET "clock" LOC = "P7" | BUFG = CLK | PERIOD =   5ms HIGH 50%;
#NET "Clk_XT" LOC = "P5" | BUFG = CLK | PERIOD = 500ns HIGH 50%;

# Keys
NET "X" LOC = "P42";
NET "RST" LOC = "P40";
#NET "Key<2>" LOC = "P43";
#NET "Key<3>" LOC = "P38";
#NET "Key<4>" LOC = "P37";
#NET "Key<5>" LOC = "P36";  # shared with ROT_A
#NET "Key<6>" LOC = "P24";  # shared with ROT_B
#NET "Key<7>" LOC = "P39";  # GSR

# LEDS
NET "Y"  LOC = "P35";
```

Układ gdy wykryje sekwencję „11100”, którą wprowadza się przez przycisk K0, wprowadzając kolejne bity w „rytm” zegara, zgasza diodę P35, która symbolizuje jej detekcję.

4. Wnioski

Wszystkie układy udało się skonstruować na zajęciach z wyjątkiem układu licznika synchronicznego modulo 9 negatywnego w kodzie +3. Tak jak dla układów kombinacyjnych kod nie był skomplikowany, tak dla sekwencyjnych wymagał dobrej znajomości podstaw języka VHDL i precyzji.