

Układ detektora sekwencji bitów: 100101.

Detektor sekwencji bitów jest układem, który w przypadku otrzymania na wejściu zadanej sekwencji bitów (100101), na wyjście poda stan wysoki, potwierdzający wykrycie tej sekwencji. Jest więc połączeniem 6-bitowego rejestru szeregowo-równoległego i uproszczonego komparatora 6-bitowego.

Rejestr szeregowo-równoległy jest układem, na który szeregowo podawane są dane wejściowe (jedna dana w jednym momencie), które „przesuwają się” w takt (zbocza narastającego) zegara *Zegar*. Dane (podawane przez 1-bitowe wejście *Q*) wchodzi na LSB (bit najmniej znaczący – $Y(0)$) i przesuwają się w kierunku MSB (bit najbardziej znaczący – $Y(5)$). Dana z MSB nie ma się już dokąd przesunąć, więc zostaje utracona na rzecz wchodzącej z bitu mniej znaczącego. Równolegle połączone wyjścia reprezentują 6-bitową daną wyjściową *Y*.

```
15  -- Rejestr szeregowo - równoległy -----
16  procedure reg
17      (signal Q, Zegar : in std_logic;
18       signal Y       : inout std_logic_vector (5 downto 0)) is
19  begin
20      if (Zegar'event and Zegar = '1') then
21          for i in 5 downto 1 loop
22              Y(i) <= Y(i-1);
23          end loop;
24          Y(0) <= Q;
25      end if;
26  end reg;
27  -----
```

Uproszczony komparator pełni tylko jedną z trzech funkcji pełnego komparatora – porównuje, czy dwie liczby zapisane w formie wektorów (podane przez 2 magistrale) są równe. Układ ten porównuje ze sobą bity dwóch liczb (*num1* i *num2* = 100101) o tych samych indeksach, tj. $num1(0)$ z $num2(0)$, $num1(1)$ z $num2(1)$ itd. W przypadku, gdy wszystkie są sobie równe, podaje na wyjście stan wysoki. W przeciwnym razie, stan niski.

```
29  -- Komparator -----
30  procedure comp
31      (signal num1 : in std_logic_vector (5 downto 0); -- liczba wejściowa
32       signal eq   : out std_logic) is -- czy równe
33      constant num2 : std_logic_vector (5 downto 0) := "100101";
34  begin
35      if (num1 = num2) then
36          eq <= '1';
37      else
38          eq <= '0';
39      end if;
40  end comp;
41  -----
```

Opis całego układu w języku VHDL:

Układ ma 2 wejścia 1-bitowe: *clk* (zegarowe) i *wej* (wejście danych podawanych do rejestru); wyjście 1-bitowe *det* (wskazujące, czy wykryto sekwencję 100101); 1 6-bitowy port 2-kierunkowy *sekwencja* przechowujący aktualną sekwencję zapisaną w rejestrze (domyślnie ustawioną na 000000).

W architekturze synchronicznie wyzwalany (zbroczem zegara *clk*) jest proces, w którym wywoływana jest procedura rejestru szeregowo-równoległego *reg*. Poza procesem, współbieżnie wywoływana jest procedura komparatora *comp*.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_UNSIGNED.ALL;
4  USE IEEE.NUMERIC_STD.ALL;
5
6  entity Zad1 is
7      port (clk      : in std_logic;
8            wej      : in std_logic; -- dana wejściowa
9            sekwencja : inout std_logic_vector (5 downto 0) := "000000"; -- poprawna sekwencja
10           det      : out std_logic); -- dana wejściowa - czy wykryto sekwencję?
11 end Zad1;
12
13 architecture Behavioral of Zad1 is
14
15     -- Rejestr szeregowo - równoległy -----
16     procedure reg
17         (signal Q, Zegar : in std_logic;
18          signal Y        : inout std_logic_vector (5 downto 0)) is
19     begin
20         if (Zegar'event and Zegar = '1') then
21             for i in 5 downto 1 loop
22                 Y(i) <= Y(i-1);
23             end loop;
24             Y(0) <= Q;
25         end if;
26     end reg;
27     -----
28
29     -- Komparator -----
30     procedure comp
31         (signal num1 : in std_logic_vector (5 downto 0); -- liczba wejściowa
32          signal eq   : out std_logic) is -- czy równe
33         constant num2 : std_logic_vector (5 downto 0) := "100101";
34     begin
35         if (num1 = num2) then
36             eq <= '1';
37         else
38             eq <= '0';
39         end if;
40     end comp;
41     -----
42
43     -- Układ -----
44     begin
45         process (clk)
46         begin
47             reg (wej, clk, sekwencja);
48         end process;
49         comp (sekwencja, det);
50     end Behavioral;
51     -----
52
```

Przebiegi symulacyjne:

Na przebiegach symulacyjnych widać, że w takt (zbocza narastającego) zegara, wpisywana jest do rejestru dana wejściowa (powodując przesunięcie pozostałych bitów, aby „zrobiły jej miejsce”). Gdy sekwencja zapisana w rejestrze wynosi 100101 (a więc jest taka na jaką zaprojektowany jest detektor), stan wyjścia „det” wskazującego fakt detekcji stanu, zmienia się na wysoki. Później sekwencja zmienia się na inną, i wyjście „det” z powrotem jest w stanie niskim.

