

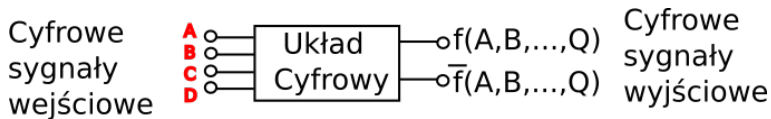
PODSTAWY UKŁADÓW PROGRAMOWALNYCH FPGA AUTOMAT STANÓW W VHDL. REJESTR PRZESUWNY

Dariusz Tefelski
dariusz.tefelski@pw.edu.pl

Politechnika Warszawska,
Wydział Fizyki

PODSTAWOWE UKŁADY CYFROWE

- W technice cyfrowej nośnikami informacji zakodowanej w postaci liczb są sygnały cyfrowe.
- Schemat funkcjonalny układu cyfrowego:



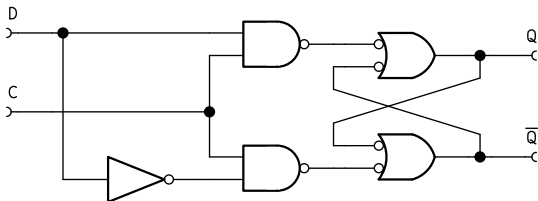
- Sygnały cyfrowe doprowadzone do wejścia układu cyfrowego powinny wymusić na jego wyjściu odpowiedź również w postaci sygnału cyfrowego zgodnie z założonym równaniem logicznym.
- Układy cyfrowe dzielimy na:
 - kombinacyjne
 - sekwencyjne

UKŁADY KOMBINACYJNE VS SEKWENCYJNE

- Układy kombinacyjne to układy logiczne, których wielkości wyjściowe w dowolnej chwili zależą wyłącznie od wielkości wejściowych w tejże chwili np. bramki.
- Układy sekwencyjne charakteryzuje zależność wielkości wyjściowych od bieżących wielkości wejściowych i stanu wewnętrznego tych układów. Z tego wynika, że określonej kombinacji sygnałów wejściowych może odpowiadać wiele kombinacji sygnałów wyjściowych.
- Do układów sekwencyjnych zalicza się: przerzutniki, liczniki, rejestry, dzielniki częstotliwości oraz sekwencyjne układy impulsowe.
- Przerzutniki są elementarnymi komórkami pamięci. To takie układy, dla których możliwe są tylko dwie, wzajemnie wykluczające się, kombinacje sygnałów wyjściowych a zmiana ich stanu może nastąpić wyłącznie pod wpływem sygnałów ustawiających. Tego typu układy nie przetwarzają informacji, lecz służą do ich przechowywania.
- Wśród przerzutników wyróżnia się następujące podstawowe typy: R-S, J-K, D i T.

PRZERZUTNIK D

- Najprostszym typem przerzutnika jest przerzutnik typu D (ang. DATA), który posiada jedno wejście, nazywane często zatrzaskiem (ang. LATCH).
- Aby można było rozróżnić wartość wielkości wejściowej jest stosowany zegar. Przerzutnik zapamiętuje taką wartość danej, jaka występowała w momencie pojawienia się impulsu zegarowego



D	C	Q	$\bar{Q}$
H	H	H	L
L	H	L	H

AUTOMAT STANÓW = MASZYNA STANÓW

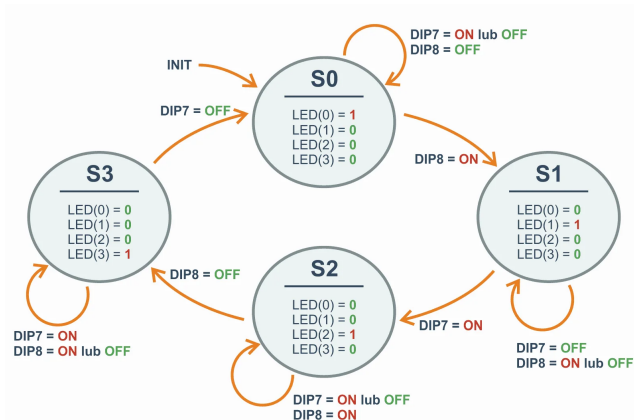
- Automaty skończone stanów składają się ze skończonego zbioru stanów i zbioru przejść pomiędzy stanami.
- Przykładowy automat stanów obrazujący zasadę działania alarmu domowego:



Źródło obrazka: <https://forbot.pl/>

PRZYKŁAD IMPLEMENTACJI AUTOMATU STANÓW

- Założmy, że chcemy stworzyć układ zgodnie z poniższym automatem stanów:



Źródło obrazka: <https://forbot.pl/>

PRZYKŁAD IMPLEMENTACJI AUTOMATU STANÓW

- Do jej realizacji potrzebujemy 4 diod LED oraz 2 przycisków:

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity maszyna_stanow is
```

```
Port (
```

```
    Clk : in  STD_LOGIC;
```

```
    DPSSwitch : in  STD_LOGIC_VECTOR(1 downto 0);
```

```
    LED : out  STD_LOGIC_VECTOR(3 downto 0));
```

```
end maszyna_stanow;
```

```
architecture Behavioral of maszyna_stanow is
```

```
-- Definiowanie stanów
```

```
type typ_stanu is (S0, S1, S2, S3);
```

```
signal Stan : typ_stanu := S0;
```

- W języku VHDL możemy tworzyć własny typ danych, który będzie przyjmować wartości z listy w następujący sposób:

type *własna_nazwa_typu* **is** (*wartości jakie może przyjmować*).

BLOK CASE... WHEN

- Blok *case... when* jest stosowana wtedy gdy chcemy wykonać inne komendy w zależności od wartości jakiejś zmiennej.
- Implementacja bloku *case... when* wygląda następująco:

```
case zmienna is:
```

```
    when wartosc1 => instrukcje;
```

```
    when wartosc2 => instrukcje;
```

```
    when others => instrukcje;
```

```
np:
```

```
case state is
```

```
    when S0 => a <= '0';
```

```
    when S1 => a <= '1';
```

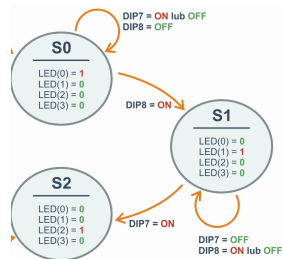
```
    when others => a <= '0';
```

PRZYKŁAD IMPLEMENTACJI AUTOMATU STANÓW

```

begin
process (Clk)
  begin
    if rising_edge(Clk) then
      case Stan is
        when S0 =>
          if (not DPSwitch(0)) = '1' then
            Stan <= S1;
          else
            Stan <= S0;
          end if;
        when S1 =>
          if (not DPSwitch(1)) = '1' then
            Stan <= S2;
          else
            Stan <= S1;
          end if;
        --Analogicznie dla stanów S2 i S3
        -- (pominiemy ten fragment)
        when others => Stan <= S0;
      end case;
    end if;
  end process;
end process;

```



Źródło obrazka: <https://forbot.pl/>

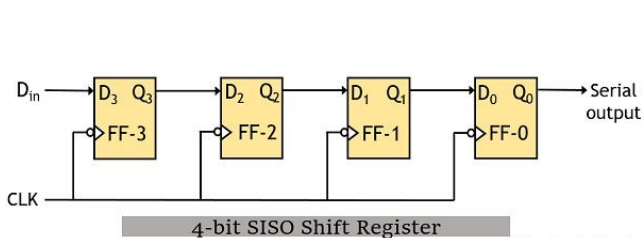
PRZYKŁAD IMPLEMENTACJI AUTOMATU STANÓW

```
-- Ponizsze instrukcje przenosza wartosc stanu
-- na diody LED
LED(0) <= '1' WHEN Stan=S0 ELSE '0';
LED(1) <= '1' WHEN Stan=S1 ELSE '0';
LED(2) <= '1' WHEN Stan=S2 ELSE '0';
LED(3) <= '1' WHEN Stan=S3 ELSE '0';

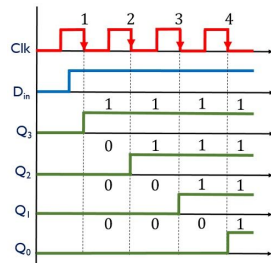
end Behavioral;
```

REJESTRY PRZESUWNE

- Rejestry - są to układy cyfrowe, które służą do przechowywania informacji cyfrowej. Rejestry składają się z przerzutników (najczęściej typu D), przy czym N-bitowy rejestr zawiera N przerzutników.



Electronics Coach



Waveform Representation of SISO Shift Register

Electronics Coach

Dziękuję za uwagę!

Prezentacja powstała w oparciu o materiały z:

- W. Tłaczała, Wirtualne laboratorium podstaw techniki cyfrowej, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2008
- <https://forbot.pl/>