

Automat do wydawania napojów (przyjmuje: 10gr, 20gr, 50gr, 1zł, 2zł, 5zł. Koszt napoju 3,30 zł, lub inny ustalony przez prowadzącego). Automat wydaje resztę (ustawia sygnał reszty).

## 1. Kod źródłowy (zał. automat.vhd).

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
use ieee.numeric_std.all;

entity automat is
  Port ( cz10gr : in STD_LOGIC;
        cz20gr : in STD_LOGIC;
        cz50gr : in STD_LOGIC;
        cz1zl : in STD_LOGIC;
        cz2zl : in STD_LOGIC;
        cz5zl : in STD_LOGIC;
        reset : in STD_LOGIC;

        clk : in STD_LOGIC;
        wydnapi : out STD_LOGIC := '0';

        reszta5 : inout STD_LOGIC_VECTOR(1 downto 0) := (others=>'0');
        reszta2 : inout STD_LOGIC_VECTOR(1 downto 0) := (others=>'0');
        reszta1 : inout STD_LOGIC_VECTOR(1 downto 0) := (others=>'0');
        reszta50 : inout STD_LOGIC_VECTOR(1 downto 0) := (others=>'0');
        reszta20 : inout STD_LOGIC_VECTOR(1 downto 0) := (others=>'0');
        reszta10 : inout STD_LOGIC_VECTOR(1 downto 0) := (others=>'0')
    );
end automat;

architecture Behavioral of automat is
  type stan is (S1, S2, S3, S4);
  signal S : stan := S1;
  signal por : STD_LOGIC := '0';

begin
  process (clk, cz10gr, cz20gr, cz50gr, cz1zl, cz2zl, cz5zl, reset)
    variable sum : integer := 0;

    begin
      if rising_edge(clk) then
        case S is
          when S1 =>
            wydnapi <= '0';

            if cz10gr='1' then sum:=sum+10;
            elsif cz20gr='1' then sum:=sum+20;
            elsif cz50gr='1' then sum:=sum+50;
            elsif cz1zl='1' then sum:=sum+100;
            elsif cz2zl='1' then sum:=sum+200;
            elsif cz5zl='1' then sum:=sum+500;
            else null;
            end if;

            if sum>=330 then por<='1';
            else por<='0';
            end if;

            if (por = '0' and reset = '0') then S<=S1;
            elsif (por = '0' and reset = '1') then S<=S2;
            elsif (por = '1') then S<=S3;
            else null;
            end if;

          when S2 =>
            wydnapi <= '0';

            if sum>=500 then reszta5<=reszta5+'1'; sum:=sum-500;
            elsif sum>=200 then reszta2<=reszta2+'1'; sum:=sum-200;
            elsif sum>=100 then reszta1<=reszta1+'1'; sum:=sum-100;
            elsif sum>=50 then reszta50<=reszta50+'1'; sum:=sum-50;
            elsif sum>=20 then reszta20<=reszta20+'1'; sum:=sum-20;
```

```

        elsif sum>=10 then reszta10<=reszta10+'1'; sum:=sum-10;
        else null;
    end if;

    if sum=0 then S<=S4;
    else null;
    end if;

when S3 =>
    wydnap<='1';

    if sum>=830 then reszta5<=reszta5+'1'; sum:=sum-500;
    elsif sum>=530 then reszta2<=reszta2+'1'; sum:=sum-200;
    elsif sum>=430 then reszta1<=reszta1+'1'; sum:=sum-100;
    elsif sum>=380 then reszta50<=reszta50+'1'; sum:=sum-50;
    elsif sum>=350 then reszta20<=reszta20+'1'; sum:=sum-20;
    elsif sum>=340 then reszta10<=reszta10+'1'; sum:=sum-10;
    else null;
    end if;

    if sum=330 then S<=S4;
    else null;
    end if;

when S4 =>
    wydnap<='0';

    reszta5 <=(others=>'0');
    reszta2 <=(others=>'0');
    reszta1 <=(others=>'0');
    reszta50 <=(others=>'0');
    reszta20 <=(others=>'0');
    reszta10 <=(others=>'0');
    sum:=0;
    por<='0';

    S<=S1;

when others => null;
end case;
end if;
end process;
end Behavioral;

```

## 2. Test bench (zał. t10\_tb.vhd).

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY t10 IS
END t10;

ARCHITECTURE behavior OF t10 IS

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

    COMPONENT automat
    PORT(
        cz10gr : IN  std_logic;
        cz20gr : IN  std_logic;
        cz50gr : IN  std_logic;
        cz1zl  : IN  std_logic;
        cz2zl  : IN  std_logic;
        cz5zl  : IN  std_logic;
        reset  : IN  std_logic;
        clk    : IN  std_logic;
        wydnap : OUT std_logic;
        reszta5 : INOUT std_logic_vector(1 downto 0);
        reszta2 : INOUT std_logic_vector(1 downto 0);
        reszta1 : INOUT std_logic_vector(1 downto 0);
        reszta50 : INOUT std_logic_vector(1 downto 0);
        reszta20 : INOUT std_logic_vector(1 downto 0);
    );

```

```

        reszta10 : INOUT std_logic_vector(1 downto 0)
    );
END COMPONENT;

```

```

--Inputs

```

```

signal cz10gr : std_logic := '0';
signal cz20gr : std_logic := '0';
signal cz50gr : std_logic := '0';
signal cz1zl : std_logic := '0';
signal cz2zl : std_logic := '0';
signal cz5zl : std_logic := '0';
signal reset : std_logic := '0';
signal clk : std_logic := '0';

```

```

--BiDirs

```

```

signal reszta5 : std_logic_vector(1 downto 0);
signal reszta2 : std_logic_vector(1 downto 0);
signal reszta1 : std_logic_vector(1 downto 0);
signal reszta50 : std_logic_vector(1 downto 0);
signal reszta20 : std_logic_vector(1 downto 0);
signal reszta10 : std_logic_vector(1 downto 0);

```

```

--Outputs

```

```

signal wydnapi : std_logic;

```

```

-- Clock period definitions

```

```

constant clk_period : time := 10 ns;

```

```

BEGIN

```

```

    -- Instantiate the Unit Under Test (UUT)

```

```

    uut: automat PORT MAP (

```

```

        cz10gr => cz10gr,
        cz20gr => cz20gr,
        cz50gr => cz50gr,
        cz1zl => cz1zl,
        cz2zl => cz2zl,
        cz5zl => cz5zl,
        reset => reset,
        clk => clk,
        wydnapi => wydnapi,
        reszta5 => reszta5,
        reszta2 => reszta2,
        reszta1 => reszta1,
        reszta50 => reszta50,
        reszta20 => reszta20,
        reszta10 => reszta10
    );

```

```

-- Clock process definitions

```

```

clk_process : process

```

```

begin

```

```

        clk <= '0';
        wait for clk_period/2;
        clk <= '1';
        wait for clk_period/2;

```

```

end process;

```

```

-- Stimulus process

```

```

stim_proc: process

```

```

begin

```

```

    wait for 10 ns;

```

```

        cz5zl<='1';
        wait for 10 ns;
        cz5zl<='0';
        wait for 100 ns;

```

```

        cz10gr<='1';
        wait for 10 ns;
        cz10gr<='0';

```

```

    cz20gr<='1';

```

```

        wait for 10 ns;

```

```

        cz20gr<='0';
        wait for 10 ns;
        reset<='1';
        wait for 10 ns;
        reset<='0';
        wait for 100 ns;

        cz50gr<='1';
        wait for 10 ns;
        cz50gr<='0';

cz20gr<='1';
        wait for 10 ns;
        cz20gr<='0';
        cz2zl<='1';
        wait for 10 ns;
        cz2zl<='0';

cz1zl<='1';
        wait for 10 ns;
        cz1zl<='0';
        wait for 100 ns;

wait;
end process;

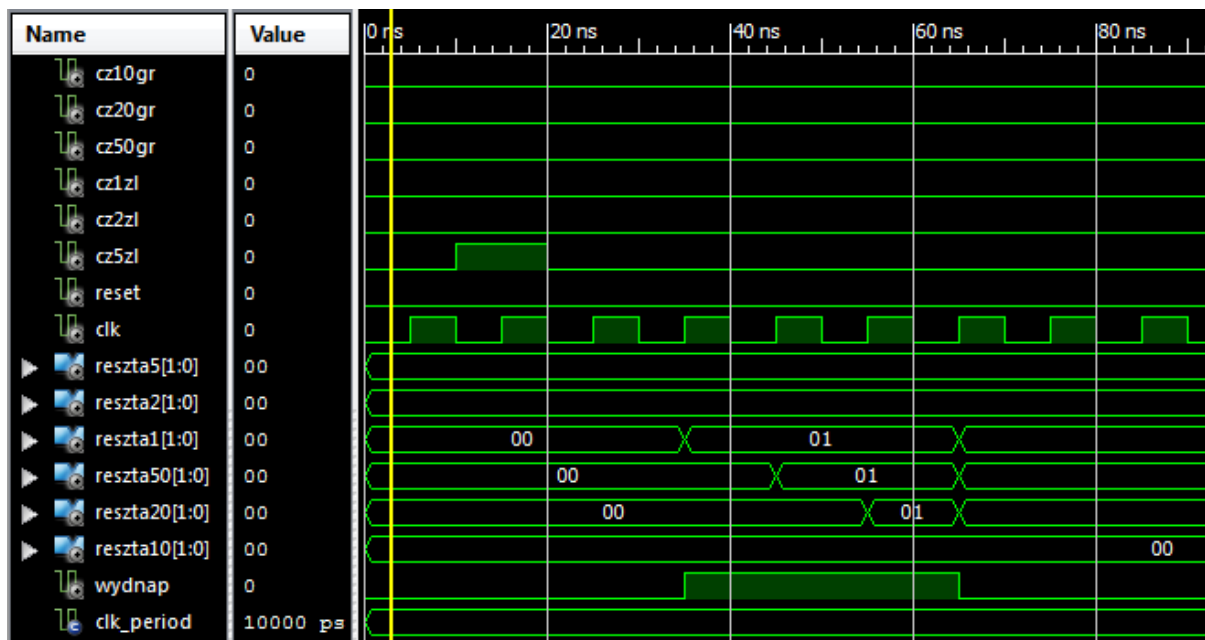
END;
```

### 3. Opis projektu.

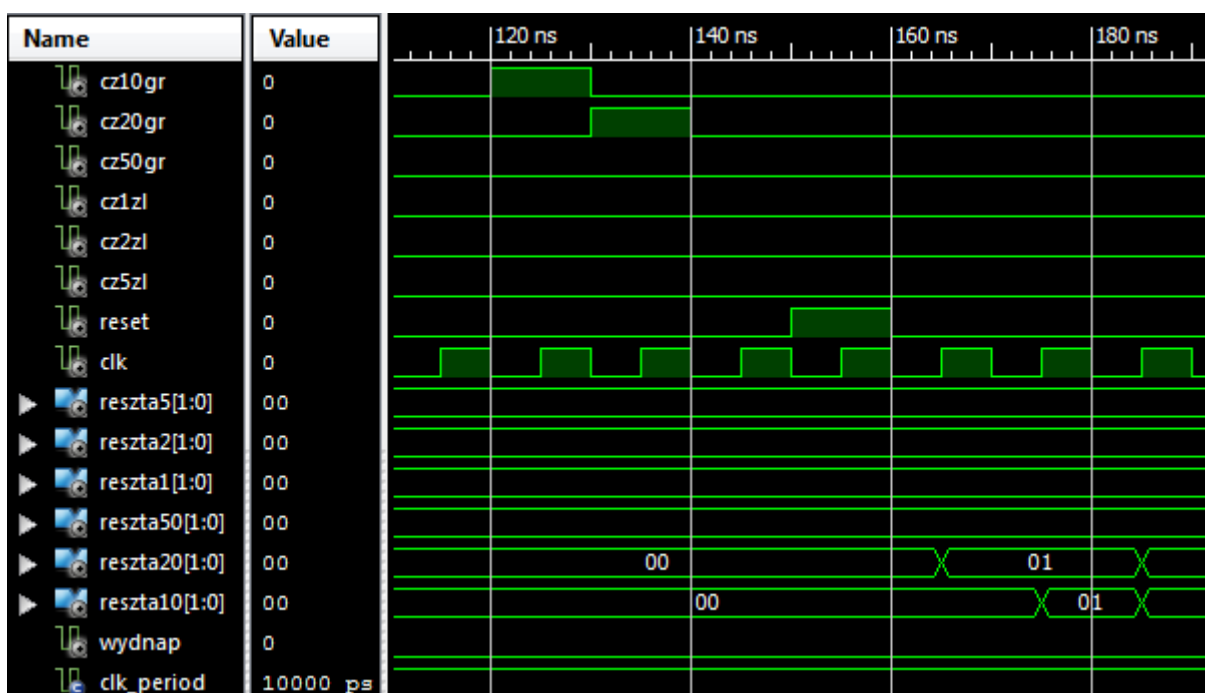
Projekt posiada:

- 6 jednobitowych wejść, na których ma pojawiać się stan wysoki w zależności od wykrytego nominału wrzuconej monety:
  - cz10gr,
  - cz20gr,
  - cz50gr,
  - cz1zl,
  - cz2zl,
  - cz5zl.
- jednobitowe wejście reset, które pozwala zrezygnować z kupna napoju i odzyskać wrzuconą do automatu kwotę.
- jednobitowe wejście zegara clk.
- jednobitowe wyjście wydnap, na którym pojawia się stan wysoki mający spowodować wydanie napoju.
- 5 dwubitowych wejść/wyjść, na których pojawia się w kodzie binarnym liczba monet danego nominału do wydania:
  - reszta10,
  - reszta20,
  - reszta50,
  - reszta1,
  - reszta2,
  - reszta5.

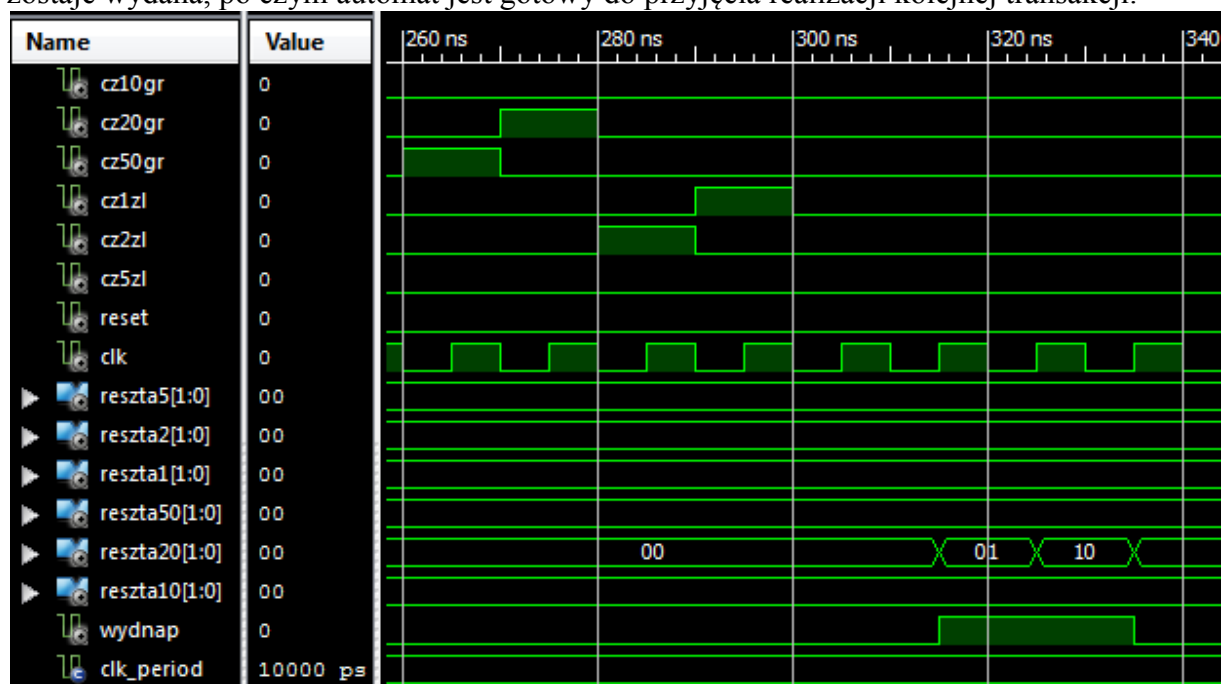
#### 4. Symulacja.



Na początku, po 10ns wejście cz5zl przyjmuje wartość 1 na 10ns. Powoduje to pojawienie się stanu wysokiego na wyjściu wydnap i wartości 1 na wyjściach reszta1, reszta50 i reszta20, po czym wszystkie wyjścia, przy kolejnym zboczach narastającym zegara się zerują. Więc po wrzuceniu 5zł zostaje wydany napój i reszta 1,70zł, po czym automat jest gotowy do przyjęcia realizacji kolejnej transakcji.



Następnie na 10ns wartości 1 przyjmują kolejno wejścia cz10gr i cz20gr, a po kolejnych 10ns również wejście reset. Powoduje to pojawienie się wartości 1 na wyjściach reszta20 i reszta10, po czym wszystkie wyjścia, przy kolejnym zboczu narastającym zegara się zerują. Więc po wrzuceniu 30gr i rezygnacji z zakupu za pomocą klawisza reset, kwota wrzucona zostaje wydana, po czym automat jest gotowy do przyjęcia realizacji kolejnej transakcji.



Następnie na 10ns wartości 1 przyjmują kolejno wejścia cz50gr, cz20gr, cz2zl i cz1zl. Powoduje to pojawienie się stanu wysokiego na wyjściu wydnap i wartości 2 na wyjściu reszta20, po czym wszystkie wyjścia, przy kolejnym zboczu narastającym zegara się zerują. Więc po wrzuceniu 3,70zł zostaje wydany napój i reszta 40gr, po czym automat jest gotowy do przyjęcia realizacji kolejnej transakcji.