Protokol

1) zadání

2) teor. rozbor Spartan a VHDL

3) definice stavů a jejich kódování

4) popis automatu Moore nebo Mealy

5) orientovaný graf

6) tabulky přechodů mezi vnitřními stavy
v závislosti na vstupních stavech
tabulky výstupů

7) VHDL moduly - programy

8) simulace

9) celkové schéma

10) výpis pinů

11) zhodnocení

1. Garáž pro pět aut + semafor když se garáž zaplní
2. **Spartan**

   - Spartan je lehká vývojová deska FPGA, je založena na čipu Xilinx Spartan-7

   - Můžete jej použít jako s Arduinem k ovládání LCD a fotoaparátu nebo jako samostatnou vývojovou desku FPGA.

   - místo makrobuněk obsahují logické bloky

   - log. Bloky jsou navzájem propojeny globální propojovací maticí

   - obsahuje SRAM

   VHDL

   - Používá se pro návrh a simulaci digitálních integrovaných obvodů, například programovatelných hradlových polí (CPLD, FPGA) nebo různých zákaznických obvodů (ASIC).

   - Jazyk VHDL může být použit i jako paralelní programovací jazyk.

3.

vstupní stavy		
	příjezd = p	odjezd = o
x0	0	0
x1	1	0
x2	0	1
x3	1	1

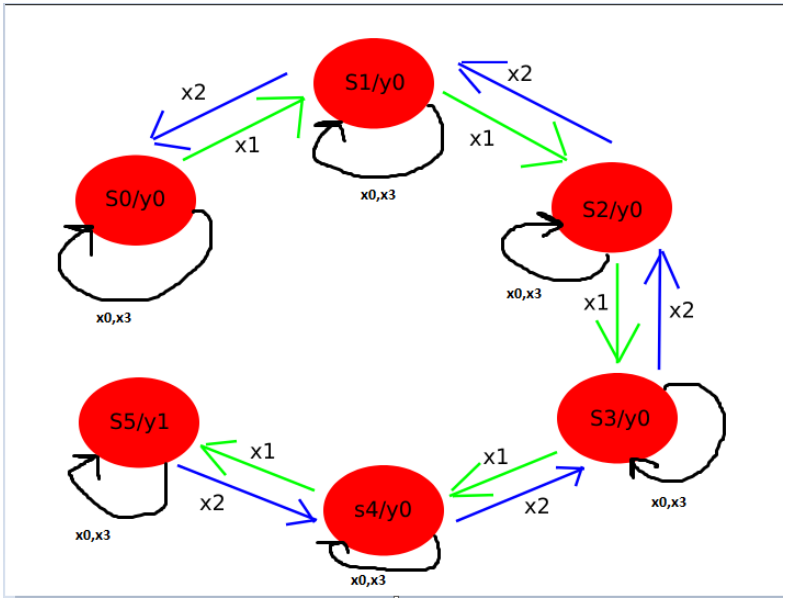
výstupní stavy	
	led = L
y0	0
y1	1

vnitřní stavy				
s0	0	0	0	....0 aut
s1	0	0	1	....1 auto
s2	0	1	0	....2 auta
s3	0	1	1	....3 auta
s4	1	0	0	....4 auta
s5	1	0	1	....5 aut

4. y= f (S) Moore... závislý na vnitřním stavu
   y= f (X,S) Mealy... závislý na vnitřním stavu a vstupu

5.

6.



	x0	x1	x2	x3	y
s0	s0	s1	s0	s0	y0
s1	s1	s2	s0	s1	y0
s2	s2	s3	s1	s2	y0
s3	s3	s4	s2	s3	y0
s4	s4	s5	s3	s4	y0
s5	s5	s5	s4	s5	y1

7.

DEKODER

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

entity dekoder is
    Port ( hex : in  STD_LOGIC_VECTOR (2 downto 0);
           led : out  STD_LOGIC_VECTOR (6 downto 0));
end dekoder;

architecture Behavioral of dekoder is
```

```
begin

        with HEX select
  LED<= "1111001" when "001",          --1
                        "0100100" when "010",--2
                        "0110000" when "011",--3
                        "0011001" when "100",--4
                        "0010010" when "101",--5
                        "1000000" when others;          --0

end Behavioral;
```

DELICKA

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

entity delicka is
    Port ( clk_in : in  STD_LOGIC;
```

```
        clk_out : out  STD_LOGIC);
end delicka;

architecture Behavioral of delicka is

begin
                process (clk_in)
                        variable i : integer range 0 to 15000000;
begin
                if rising_edge(clk_in) then
                                if i=0 then clk_out <= '1' ;
                                        i := 9843000 ;
                                else
                                        clk_out <='0' ;
                                        i := i-1 ;
                                end if ;
                        end if;
                end process;
end Behavioral;
```

GARAZ

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

entity garaz is
   Port ( clk : in  STD_LOGIC;
          reset : in  STD_LOGIC;
          p : in  STD_LOGIC;
          stav : inout  STD_LOGIC_VECTOR (2 downto 0);
```

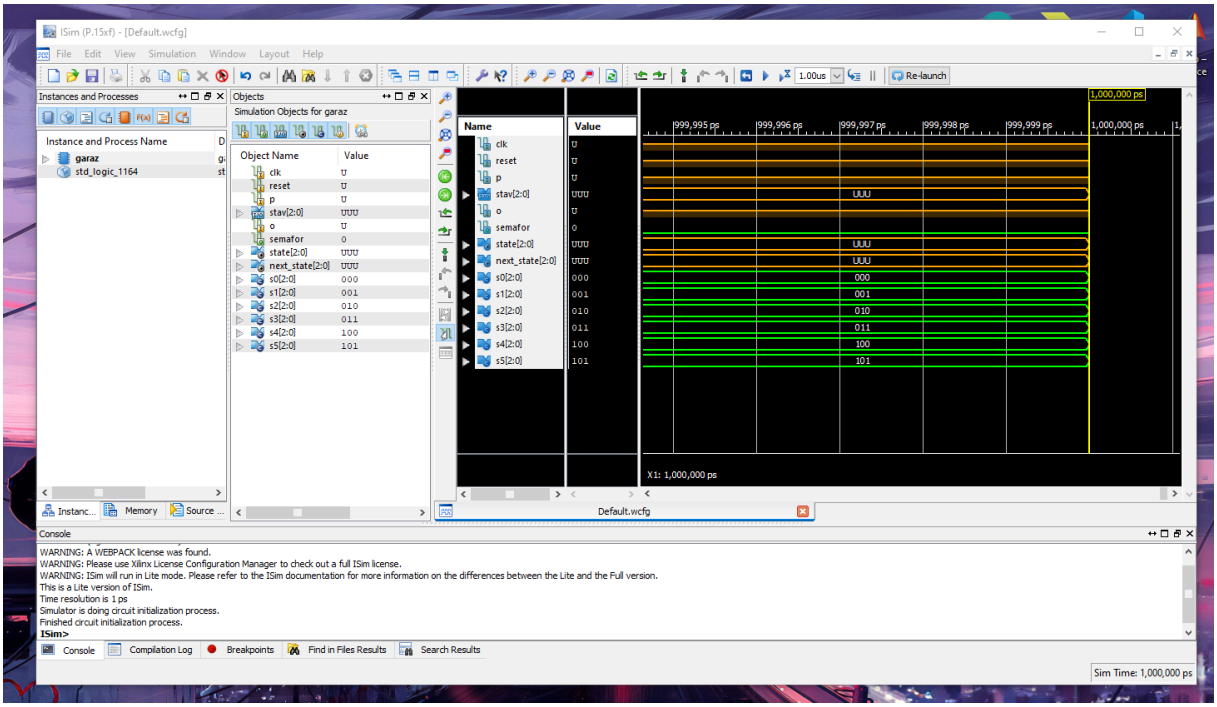
```
        o : in  STD_LOGIC;
        semafor : out  STD_LOGIC);
end garaz;

architecture Behavioral of garaz is

        signal state, next_state : std_logic_vector(2 downto 0);

        constant s0 : std_logic_vector(2 downto 0)  := "000";
        constant s1 : std_logic_vector(2 downto 0)  := "001";
        constant s2 : std_logic_vector(2 downto 0)  := "010";
        constant s3 : std_logic_vector(2 downto 0)  := "011";
        constant s4 : std_logic_vector(2 downto 0)  := "100";
        constant s5 : std_logic_vector(2 downto 0)  := "101";

begin

        SYNCHRONIZACNI_PROCES: process (clk, reset)
        begin
                        if (reset = '1') then
                                state <= s0;
                        elsif rising_edge(clk) then
                                state <= next_state;
                        end if;
        end process SYNCHRONIZACNI_PROCES;


        ZAKODOVANI_VYSTUPU: process (state)
        begin
                case (state) is
                when s5 =>
```

```
                semafor <='1';

        when others =>
                semafor <='0';
        end case;
end process ZAKODOVANI_VYSTUPU;


ZAKODOVANI_STAVU: process (p, o, state)
begin
        case(state) is

                when s0 =>
                        if (p ='1' AND o = '0') then
                  next_state <= s1;
                        else
                  next_state <= s0;
                        end if;

                when s1 =>
                        if (p ='1' AND o = '0') then
                                next_state <= s2;
                        elsif (p = '0' AND o = '1') then
                                next_state <= s0;
                        else
                  next_state <= s1;
                        end if;

                when s2 =>
                        if (p ='1' AND o = '0') then
                                        next_state <= s3;
```

```
        elsif (p = '0' AND o = '1') then
  next_state <= s1;
        else
  next_state <= s2;
        end if;

when s3 =>
        if (p ='1' AND o = '0') then
  next_state <= s4;
        elsif (p = '0' AND o = '1') then
  next_state <= s2;
        else
            next_state <= s3;
        end if;

when s4 =>
        if (p ='1' AND o = '0') then
  next_state <= s5;
        elsif (p = '0' AND o = '1') then
  next_state <= s3;
        else
  next_state <= s4;
        end if;

when s5 =>
        if (p = '0' AND o = '1') then
  next_state <= s4;
        else
  next_state <= s5;
        end if;
```

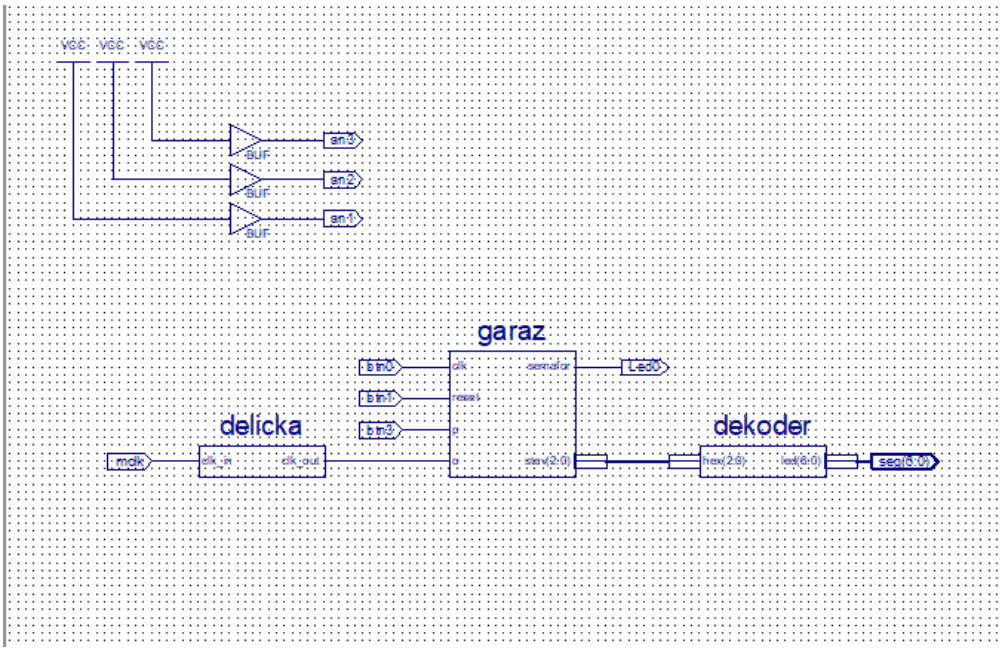
```
                when others => NULL;
            end case;

        stav <= state;

    end process ZAKODOVANI_STAVU;

end Behavioral;
```

8.



V simulaci se mi to nepodařilo zprovoznit.

9.



10.

clock pins for Basys2 Board

NET "mclk" LOC = "B8"; # Bank = 0, Signal name = MCLK

Pin assignment for DispCtl

Connected to Basys2 onBoard 7seg display

NET "seg<0>" LOC = "L14"; # Bank = 1, Signal name = CA

NET "seg<1>" LOC = "H12"; # Bank = 1, Signal name = CB

NET "seg<2>" LOC = "N14"; # Bank = 1, Signal name = CC

NET "seg<3>" LOC = "N11"; # Bank = 2, Signal name = CD

NET "seg<4>" LOC = "P12"; # Bank = 2, Signal name = CE

NET "seg<5>" LOC = "L13"; # Bank = 1, Signal name = CF

NET "seg<6>" LOC = "M12"; # Bank = 1, Signal name = CG

#NET "dp" LOC = "N13"; # Bank = 1, Signal name = DP

NET "an3" LOC = "K14"; # Bank = 1, Signal name = AN3

NET "an2" LOC = "M13"; # Bank = 1, Signal name = AN2

NET "an1" LOC = "J12"; # Bank = 1, Signal name = AN1

#NET "an0" LOC = "F12"; # Bank = 1, Signal name = AN0

```
# Pin assignment for LEDs
#NET "Led<7>" LOC = "G1" ; # Bank = 3, Signal name = LD7
#NET "Led<6>" LOC = "P4" ; # Bank = 2, Signal name = LD6
#NET "Led<5>" LOC = "N4" ;  # Bank = 2, Signal name = LD5
#NET "Led<4>" LOC = "N5" ;  # Bank = 2, Signal name = LD4
#NET "Led<3>" LOC = "P6" ; # Bank = 2, Signal name = LD3
#NET "Led<2>" LOC = "P7" ; # Bank = 3, Signal name = LD2
#NET "Led<1>" LOC = "M11" ; # Bank = 2, Signal name = LD1
NET "Led0" LOC = "M5" ;  # Bank = 2, Signal name = LD0

# Pin assignment for SWs
#NET "sw7" LOC = "N3";  # Bank = 2, Signal name = SW7
#NET "sw6" LOC = "E2";  # Bank = 3, Signal name = SW6
#NET "sw5" LOC = "F3";  # Bank = 3, Signal name = SW5
#NET "sw4" LOC = "G3";  # Bank = 3, Signal name = SW4
#NET "sw3" LOC = "B4";  # Bank = 3, Signal name = SW3
#NET "sw2" LOC = "K3";  # Bank = 3, Signal name = SW2
#NET "sw1" LOC = "L3";  # Bank = 3, Signal name = SW1
#NET "sw0" LOC = "P11";  # Bank = 2, Signal name = SW0

NET "btn3" LOC = "A7";  # Bank = 1, Signal name = BTN3
#NET "btn2" LOC = "M4";  # Bank = 0, Signal name = BTN2
NET "btn1" LOC = "C11"; # Bank = 2, Signal name = BTN1
NET "btn0" LOC = "G12"; # Bank = 0, Signal name = BTN0

## Pin assignment for PS2
#NET "ps2c"    LOC = "B1"   | DRIVE = 2  | PULLUP ; # Bank = 3, Signal name = PS2C
#NET "ps2d"    LOC = "C3"   | DRIVE = 2  | PULLUP ; # Bank = 3, Signal name = PS2D
```

11.

Po dvou dnech jsem se dostal až k simulaci. Při prací z ISE project jsem narazil na pár problémů. Ze začátku mi nešel vytvořit nový projekt s tím mi poradil pan učitel Mádecky ještě v hodině. Dále mi Nešlo vytvořit nový source pokaždé mi software crashnul ale našel jsem řešení tohoto problému na

youtube a fungovalo. Jako další problém byl že mi nešlo otevřít ani garaz, delicka a dekoder a proto jsem to udělal celé od začátku. Snažil jsem se najít chybu v simulaci ale nenašel jsem ji.