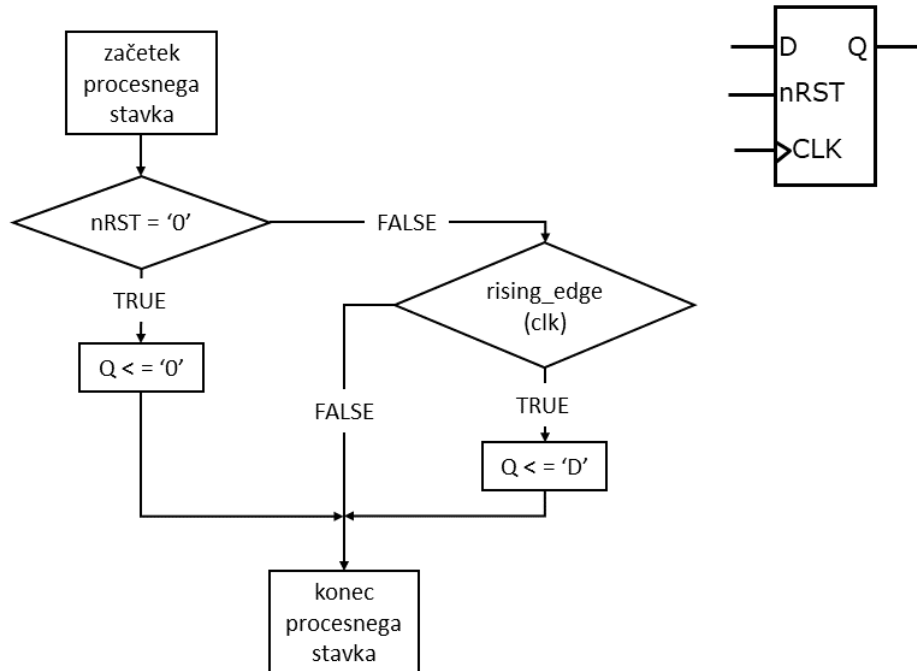


RAZVOJ DIGITALNIH SISTEMOV

VAJA 3: PS2 TIPKOVNICA

3.1 Ustvarite nov projekt (Vaja3_1) in realizirajte sinhrono D spominsko celico (Dflipflop.vhd). Celica ima vhod za ponastavitev (nRST), ki v aktivnem stanju '0' postavi izhod celice (Q) na '0'. S proženjem celice na prednji rob signala ure (CLK) izvajamo prepis vhoda (D) na izhod (Q).



Ta naloga nima svoje UCF datoteke, zato uporabite pripravljeno testno VHDL datoteko (Dflipflop_tb.vhd) in preverite pravilnost realizacije v simulatorju.

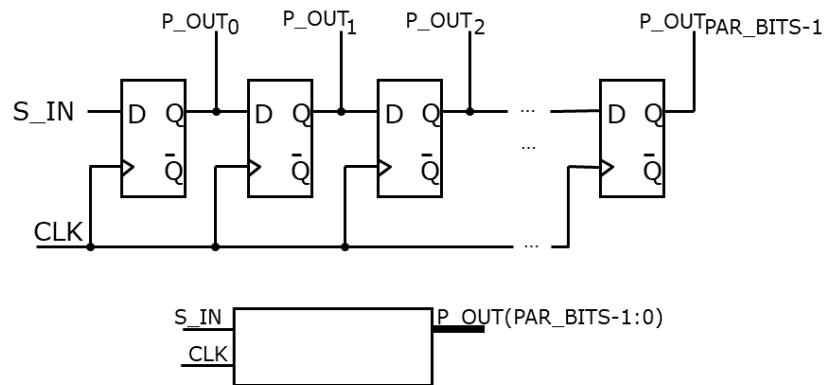
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
USE IEEE.STD_LOGIC_UNSIGNED.all;

entity Dflipflop is
    Port ( nRST : in STD_LOGIC;
          clk  : in STD_LOGIC;
          D    : in STD_LOGIC;
          Q    : out STD_LOGIC);
end Dflipflop;

architecture Behavioral of Dflipflop is
begin

end Behavioral;
```

- 3.2 Ustvarite nov projekt (Vaja3_2) in realizirajte N-bitni zaporedno-vzporedni pomikalni register (serial in – parallel out) v novi datoteki (shiftReg.vhd). Register ima serijski vhod (S_IN) in poljubno velik vzporedni izhod (P_OUT), ki ga določimo z vrednostjo spremenljivke PAR_BITS. (prednastavljena vrednost spremenljivke je 8, kar predstavlja 8-biten paralelni izhod). Zaporedno-vzporedni register deluje tako, da se ob vsakem prednjem signalu ure (CLK) v MSB mesto vzporednega izhoda (P_OUT) naloži vrednost zaporednega vhoda (S_IN).



```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

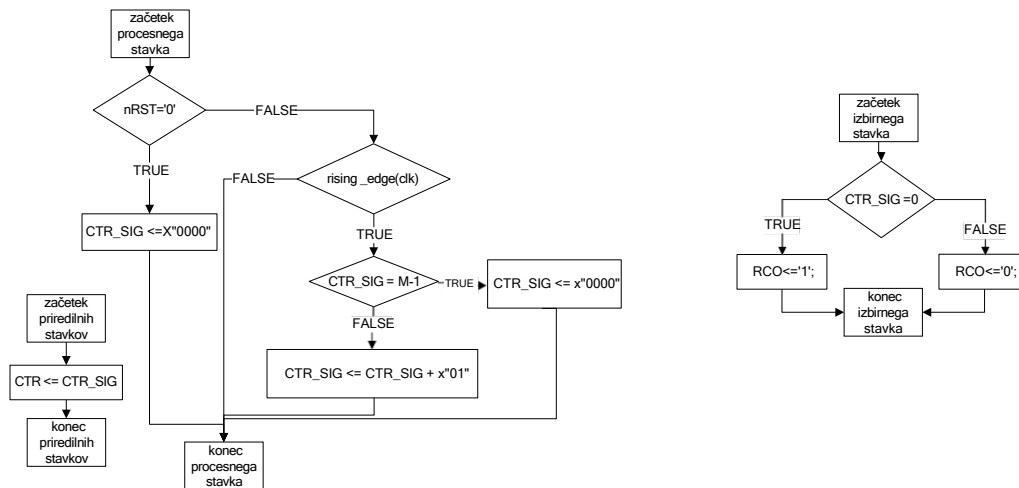
entity shiftReg is
    generic(
        PAR_BITS : integer := 8; --število bitov izhoda N bitnega
        zaporedno-vzporednega pomikalnega registra
    )
    port(
        CLK      : in  STD_LOGIC; --system clock input
        S_IN     : in  STD_LOGIC;  --clock signal from PS2 keyboard
        P_OUT    : in  STD_LOGIC_VECTOR ((PARBITS- 1) downto 0)
    ); --data signal from PS2 keyboard
end shiftReg;

architecture arch of shiftReg is
begin
end arch;

```

Delovanje serijsko-paralelnega registra preverite na plošči Basys2, tako da spreminjate serijski vhod (S_IN), ki se nahaja na stikalu SW0. Vhod signala ure pomikalnega registra (CLK) je vezan na tipko BTN3, medtem ko je 8-biten vzporedni izhod (P_OUT) vezan na LED diode LD0 do LD7 plošče Basys 2.

- 3.3 Ustvarite nov projekt (Vaja3_3) in realizirajte sinhroni 16 bitni števec (module_m_counter.vhd) po modulu M, ki šteje od 0 do pred nastavljene vrednosti M. Števec ima interni register (COUNTER) v katerem hranimo stanje štetja, vhod za ponastavitev (nRST), ki v aktivnem stanju '0' postavi vsebino štetja na 0 (CTR_SIG <= X"0000"). Z vhodom za proženje štetja (CLK) izvajamo štetje navzgor na prednji rob za eno mesto levo kot kaže spodnja slika.



```

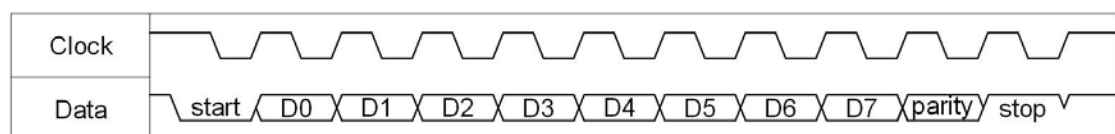
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity module_m_counter is
    generic(
        M: natural := 60
    );
    Port ( nRST, clk : in STD_LOGIC;
          CTR : out STD_LOGIC_VECTOR (15 downto 0);
          RCO : out STD_LOGIC
        );
end module_m_counter;

```

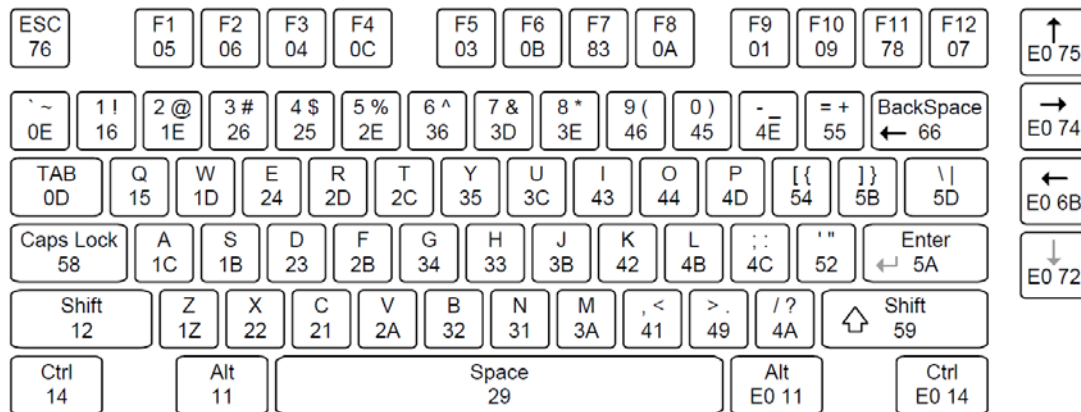
Vezje ima UCF datoteko, v kateri je vhod signala ure števca (CLK) vezan na tipko BTN3, vhod za ponastavitev (nRST) je vezan na stikalo SW0, medtem ko je vrednost števca prikazana na LED diodah ploščice Basys 2.

3.4 Ustvarite nov projekt (Vaja3_4), uvozite datoteko PS2_Keyboard.vhd in realizirajte branje tipk iz tipkovnice PS/2.

PS/2 (IBM Personal System/2) je vmesnik osebnih računalnikov za tipkovnice in miške. Priključek za PS/2 naprave je 6-pinski Mini-DIN. Računalnik mora tipkovnici zagotoviti 5 V napajanje ter ozemljitev. Osnovna komunikacija poteka serijsko po dveh linijah, in sicer podatkovni liniji (ps2_data) ter uri (ps2_clk). Tako ura, kot podatkovna linija sta v neaktivni fazi na visokem nivoju. Tipkovnica zagotovi uro in serijski podatek. Frekvenca ure tipkovnice je med 10 kHz in 16,7 kHz (perioda med 60 do 100 μ s). Podatek se prične z start bitom (nizek nivo), nadaljuje z osmimi biti podatkov, paritetnim bitom ter stop bitom (visokim nivojem). Podatek posreduje najprej manj pomembni bit (LSB), podatke pa beremo ob spremembi ure iz visokega v nizki nivo. Ko se prenos zaključi se tako podatkovna linija kot ura postavita na visoki nivo.



Podatek iz tipkovnice je drugačen za pritisk tipke (make code) ali njeno sprostitve (release code). Trije različni načini kodiranja tipkovnice obstajajo. V naši nalogi se bomo osredotočili le na pritisk tipke. Podatek s tipkovnice je običajno osem biten. Dodatnih osem bitov je namenjenih za funkcijske tipke (običajno je to x"E0" ali x"F0"). Tipki PAUSE in PRNT SCRNL sta izjemi in jih v tem sklopu vaj ne bomo obravnavali.



Arhitektura PS/2 tipkovnice je prikazana na spodnji shemi. Ura (ps2_clk) in podatkovna linija (ps2_data) sta najprej sinhronizirani s procesorsko uro (clk). Podatke iz podatkovne linije naložimo v premikalni register (shiftReg). Za preverjanje ali je preteklo 55 μ s in bi moral biti podatek iz tipkovnice že prenesen uporabimo sinhroni 16 bitni števec (module_m_counter). S kombinacijskim vezjem preverjamo ali je pri prenosu podatka prišlo do napake. Iz deset bitnega registra, poimenovanega PS_WORD preverimo ali se paritetni bit ujema z poslanimi podatki po enačbi:

```
ERROR <= not (not PS2_WORD(0) and PS2_WORD(10) and (ps2_word(9) xor
PS2_WORD(8) xor PS2_WORD(7) xor PS2_WORD(6) xor PS2_WORD(5) xor
PS2_WORD(4) xor PS2_WORD(3) xor PS2_WORD(2) xor PS2_WORD(1)));
```

Ko preverimo, da je podatek pravilen lahko izhodni pin PS2_code_new postavimo na '1'. V registru ostane podatek toliko časa, dokler tipkovnica ne pošlje novega podatka, s katerim izvršimo prepis registra. Hitrost sistemske ure vpliva na čas prenosa podatka v register. Spremenljivka CLK_FREQ mora biti vezana na sistemsko uro 50 MHz.

Vezje ima UCF datoteko, v kateri sta signala tipkovnice (PS2_CLK in PS2_DATA) vezana na PS/2 vhod ploščice Basys 2, medtem ko se vrednost podatka tipkovnice (PS2_code) prikazuje na LED diodah LD0 do LD7 in vrednost signala PS2_code_new na led diodi N13.


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

entity PS2_KEYBOARD2ASCII is
    port(
        clk          : in  STD_LOGIC;  --system clock input
        PS2_CLK       : in  STD_LOGIC;  --clock signal from PS2 keyboard
        PS2_DATA       : in  STD_LOGIC;  --data signal from PS2 keyboard
        ASCII_NEW      : out STD_LOGIC;  --output flag indicating new ASCII value
        ASCII_CODE     : out STD_LOGIC_VECTOR(6 downto 0)); --ASCII value
end PS2_KEYBOARD2ASCII;

architecture arch of PS2_KEYBOARD2ASCII is
begin
end arch;
```