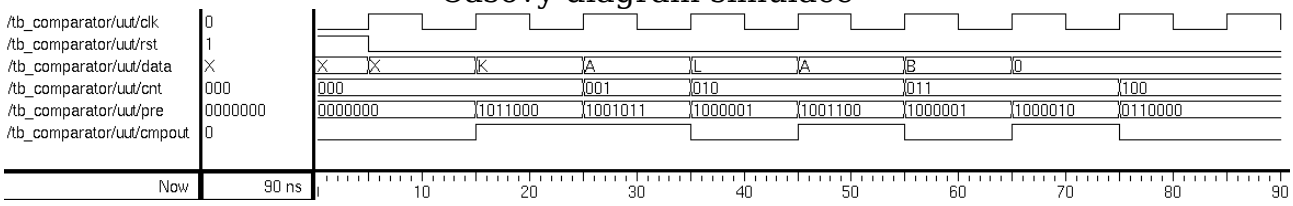
Jan Kaláb, xkalab00

comparator.vhd

```
architecture Comparator_arch of Comparator is

signal pre: std_logic_vector(6 downto 0);
signal cmpout: std_logic; --vystup comparatoru
-----------------------------------------------------------

begin

cmp_proc: process(DATA, pre) --Zde doplnte kod procesu popisujiciho
komparator.
begin
 if (pre > DATA) then --predchozi pismenko je vyssi nez soucasne
  cmpout <= '1'; --ano
 else
  cmpout <= '0'; --ne
 end if;
end process cmp_proc;

reg_proc: process(CLK, RST) --Zde doplnte kod procesu popisujiciho registr.
begin
 if (RST = '1') then --asynchronni reset
  pre <= "0000000"; --ascii ma 7 bitu
 elsif (CLK'event and CLK = '1') then --nabezna hrana
  pre <= DATA; --uloz vstup
 end if;
end process reg_proc;

cnt_proc: process(CLK, RST) --Zde doplnte kod procesu popisujiciho citac.
variable RES: std_logic_vector(2 downto 0); --vysledek (RESult)
begin
 if (RST = '1') then --asynchronni reset
  RES := (others => '0');
 elsif (CLK'event and CLK = '1' and cmpout = '1') then --nabezna hrana a
komparator
  RES := RES + 1; --plus 1
 end if;
CNT <= RES; --ulozeni vysedku
end process cnt_proc;

end architecture Comparator_arch;
```

Časový diagram simulace



comparator_tb.vhd

```
architecture TB_Comparator_arch of TB_Comparator is

constant period: time := 10 ns;

component Comparator is
port(
    CLK: IN std_logic;
    RST: IN std_logic;
    DATA: IN std_logic_vector(6 downto 0);

    CNT: OUT std_logic_vector(2 downto 0)
);
end component Comparator;

-- Vnitrni signaly test benche
signal clk: std_logic := '0';
signal rst: std_logic := '1';
signal data: std_logic_vector(6 downto 0);
signal cnt: std_logic_vector(2 downto 0);

begin

uut: Comparator --Comparator s namapovanymi vstupy
port map (
 CLK => clk,
 RST => rst,
 DATA => data,
 CNT => cnt
);
-------------------------------------------------------
clk <= NOT clk after period / 2;

TB_proc: process
begin

wait until clk'event AND clk = '1';
rst <= '0'; --odresetujeme

data <= "1011000"; --X
wait until clk'event AND clk = '1';

data <= "1001011"; --K
wait until clk'event AND clk = '1';

data <= "1000001"; --A
wait until clk'event AND clk = '1';

data <= "1001100"; --L
wait until clk'event AND clk = '1';

data <= "1000001"; --A
wait until clk'event AND clk = '1';

data <= "1000010"; --B
wait until clk'event AND clk = '1';

data <= "0110000"; --0
wait until clk'event AND clk = '1';

data <= "0110000"; --0
wait until clk'event AND clk = '1';

wait; --stuj
-------------------------------------------------------
end process TB_proc;

end architecture TB_Comparator_arch;
```