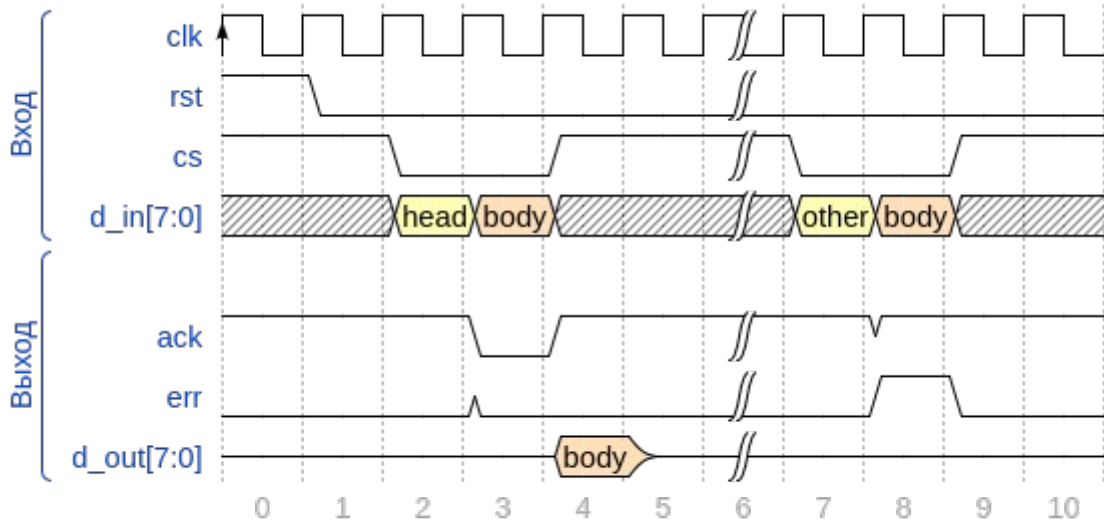


Вариант 1. Разработать устройство в соответствии с диаграммой.



Заголовок пакета: head = "11001010", other <> head.

```

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

entity VAR1 is
  PORT(
    CLK, RST, CS : in std_logic;
    D_IN         : in std_logic_vector(7 downto 0);
    ACK, ERR    : out std_logic;
    D_OUT       : out std_logic_vector(7 downto 0)
  );
end VAR1;

```

```

architecture behav of VAR1 is
  signal CS_fall : std_logic;           --сигнал для определения спада CS
  signal CS_int  : std_logic;           --буферизированный CS
  signal ACK_int : std_logic;           --внутренний сигнал ACK
begin

```

```

  ack_p : process(CLK, RST)
  begin
    if RST = '1' then
      CS_int <= '1';
    elsif CLK'event and CLK = '1' then
      CS_int <= CS;
    end if;
  end process;
  CS_fall <= '1' when CS = '0' and CS_int = '1' else '0';

```

```

  ACK_int_p : process(CLK, RST)
  begin
    if RST = '1' then
      ACK_int <= '1';
      ERR <= '0';
    elsif CLK'event and CLK = '1' then
      if CS_fall = '1' then

```

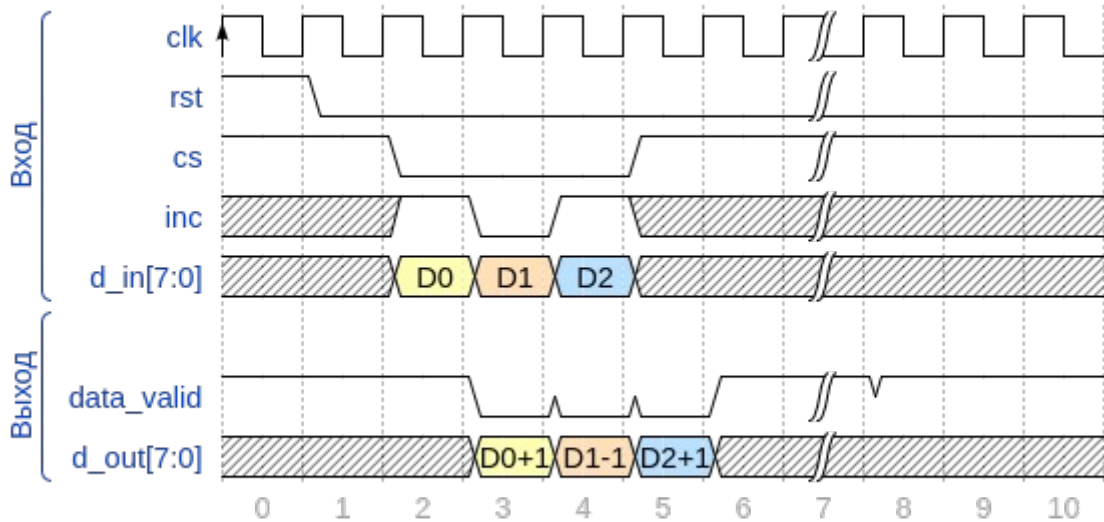
```

        if D_IN = "11001010" then
            ACK_int <= '0';
            ERR    <= '0';
        else
            ACK_int <= '1';
            ERR    <= '1';
        end if;
    else
        ACK_int <= '1';
        ERR    <= '0';
    end if;
end if;
end process;
ACK <= ACK_int;

D_OUT_p : process(CLK, RST)
begin
    if RST = '1' then
        D_OUT <= (others => 'Z');
    elsif CLK'event and CLK = '1' then
        if ACK_int = '0' then
            D_OUT <= D_IN;
        else
            D_OUT <= (others => 'Z');
        end if;
    end if;
end process;
end behav;

```

Вариант 2. Разработать устройство в соответствии с диаграммой.



При inc = "1" - инкремент; при inc = "0" - декремент.

```

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

entity VAR2 is
  PORT(
    CLK, RST, CS, INC : in std_logic;
    D_IN      : in std_logic_vector(7 downto 0);
    DATA_VALID : out std_logic;
    D_OUT      : out std_logic_vector(7 downto 0)
  );
end VAR2;

architecture behav of VAR2 is
begin

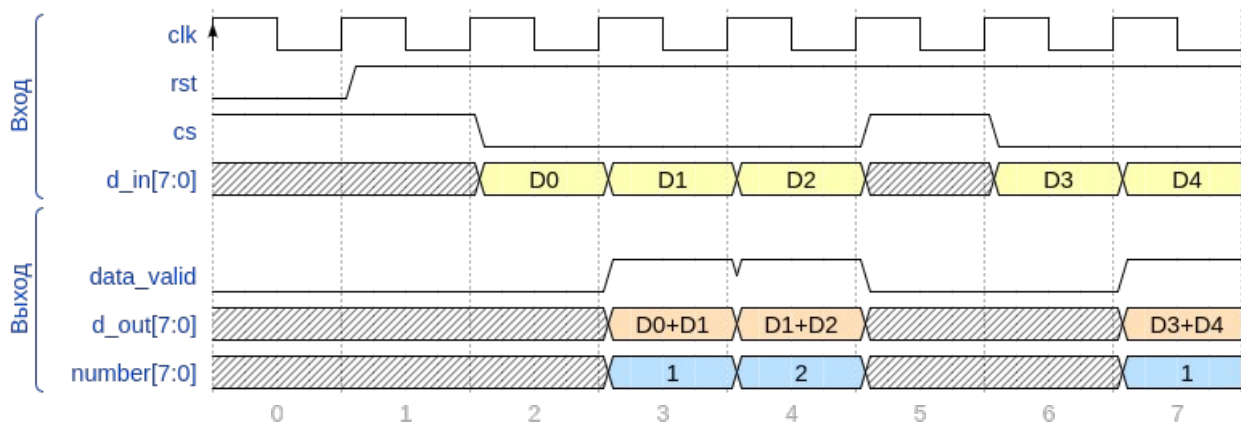
  DATA_VALID_p : process(CLK, RST)
  begin
    if RST = '1' then
      DATA_VALID <= '1';
    elsif CLK'event and CLK = '1' then
      DATA_VALID <= CS;
    end if;
  end process;

  D_OUT_p : process(CLK, RST)
  begin
    if RST = '1' then
      D_OUT <= (others => '0');
    elsif CLK'event and CLK = '1' then
      if INC = '1' then
        D_OUT <= D_IN + 1;
      else
        D_OUT <= D_IN - 1;
      end if;
    end if;
  end process;

```

end behav;

Вариант 3. Разработать устройство в соответствии с диаграммой.



Устройство выполняет сложение текущего с предыдущим и выдает номер слова.

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

```
entity VAR3 is
  PORT(
    CLK, RST, CS : in std_logic;
    D_IN         : in std_logic_vector(7 downto 0);
    DATA_VALID  : out std_logic;
    D_OUT, NUMBER : out std_logic_vector(7 downto 0)
  );
end VAR3;
```

```
architecture behav of VAR3 is
  signal CNT : std_logic_vector(7 downto 0); --счетчик
  signal D_tmp : std_logic_vector(7 downto 0); --буферизированный D_IN
begin
```

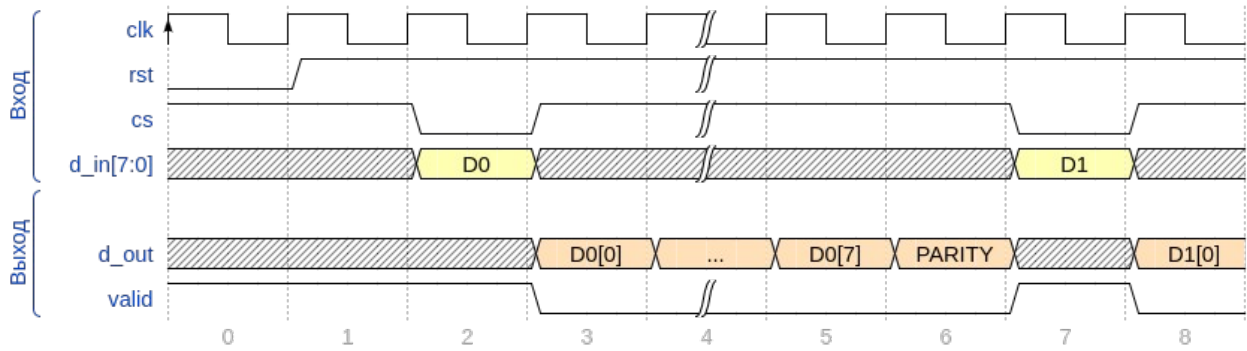
```
  D_TMP_p : process(CLK, RST)
  begin
    if RST = '0' then
      D_tmp <= (others => '0');
    elsif CLK'event and CLK = '1' then
      D_tmp <= D_IN;
    end if;
  end process;
```

```
  CNT_p : process(CLK, RST, CS)
  begin
    if RST = '0' or CS = '1' then
      CNT <= (others => '0');
      DATA_VALID <= '0';
    elsif CLK'event and CLK = '1' then
      CNT <= CNT + 1;
      DATA_VALID <= not CS;
    end if;
  end process;
```

```
  D_OUT <= D_tmp + D_IN;
  NUMBER <= CNT;
```

```
end behav;
```

Вариант 4. Разработать устройство в соответствии с диаграммой.



Устройство выполняет передачу слова и разряда контроля по четности PARITY.

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

```
entity VAR4 is
  PORT(
    CLK, RST, CS : in std_logic;
    D_IN : in std_logic_vector(7 downto 0);
    VALID, D_OUT : out std_logic
  );
end VAR4;
```

```
architecture behav of VAR4 is
  signal CNT : std_logic_vector(3 downto 0); --счетчик
  signal D_tmp : std_logic_vector(8 downto 0); --буферизированный D_IN и PAR
begin
```

```
  D_TMP_p : process(CLK, RST)
  begin
    if RST = '0' then
      D_tmp <= (others => '0');
    elsif CLK'event and CLK = '1' then
      if CS = '0' then
        D_tmp(7 downto 0) <= D_IN;
        D_tmp(8) <= D_IN(7) xor D_IN(6) xor D_IN(5) xor D_IN(4)
xor D_IN(3) xor D_IN(2) xor D_IN(1) xor D_IN(0);
      else
        D_tmp <= '0' & D_tmp(8 downto 1);
      end if;
    end if;
  end process;
```

```
  CNT_p : process(CLK, RST)
  begin
    if RST = '0' then
      CNT <= "1001";
    elsif CLK'event and CLK = '1' then
      if CS = '0' then
        CNT <= (others => '0');
      elsif CNT < "1001" then
        CNT <= CNT + 1;
      else
        end if;
      end if;
    end process;
```

```
  VALID <= '0' when CNT /= "1001" else '1';
```

```
D_OUT <= D_tmp(0);
```

```
end behav;
```