

Examen langage V H D L

Durée 2H00

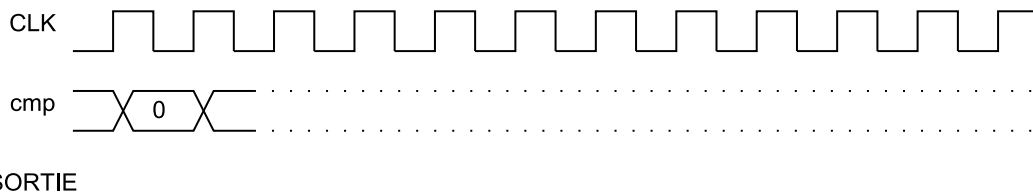
- 1- Dans quels cas on privilégie l'utilisation d'un ASIC par rapport à un FPGA et pourquoi ?
- 2- Pour le programme ci-dessous, complétez le chronogramme ci-dessous.

```

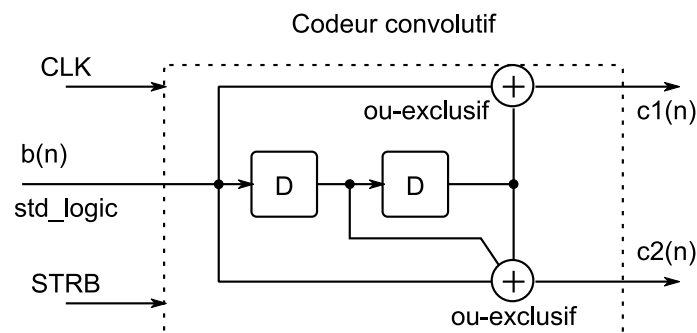
signal cmp : integer range 0 to 5;

process (CLK)
begin
    if CLK'event and CLK='1' then
        cmp <= cmp + 1;
        if cmp = 3 then SORTIE <= '1';
        else SORTIE <= '0';
        end if;
    end if;
end process;

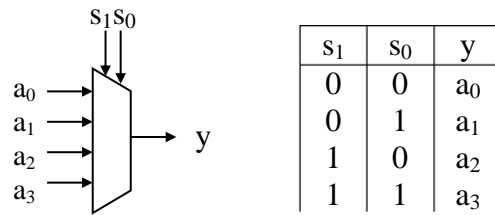
```



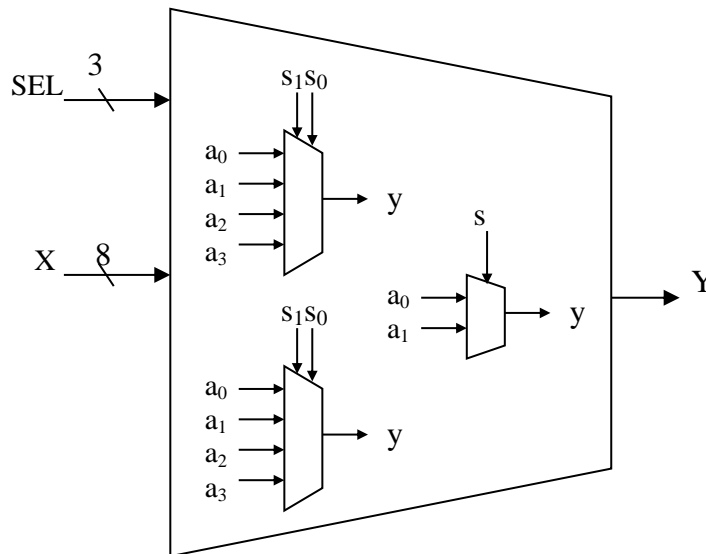
- 3- La figure ci-dessous présente la structure d'un codeur convolutif (7,5). Donner un programme VHDL pour réaliser ce composant. Les bits en entrée sont à prendre sur le front montant de CLK quand le signal STRB='1'. C'est à dire que la fréquence de STRB fixe la fréquence d'échantillonnage. Les additionneurs sont modulo 2 (ou-exclusif). Les "D" sont des retards.



- 4- Nous avons à notre disposition une entité intitulée "mux4to1" qui est un multiplexeur 4 voies vers une, schématisé ci-dessous.



Le but est d'utiliser cette entité pour fabriquer un multiplexeur 8 voies vers une. Des indications sur le schématique du circuit résultant sont donnée ci-dessous.



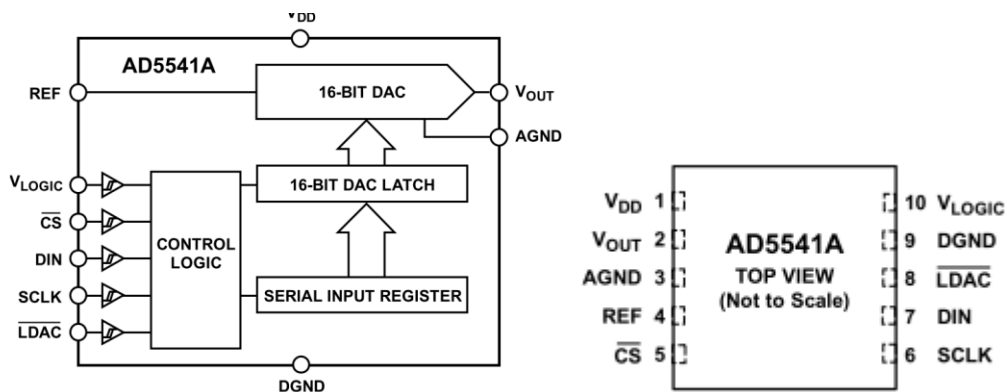
Donner la table de vérité du multiplexeur 8 -> 1.
 Faites des interconnexions nécessaires sur le schéma ci-dessus pour obtenir le mux8to1.
 Compléter le programme ci-dessous pour générer ce multiplexeur.

```

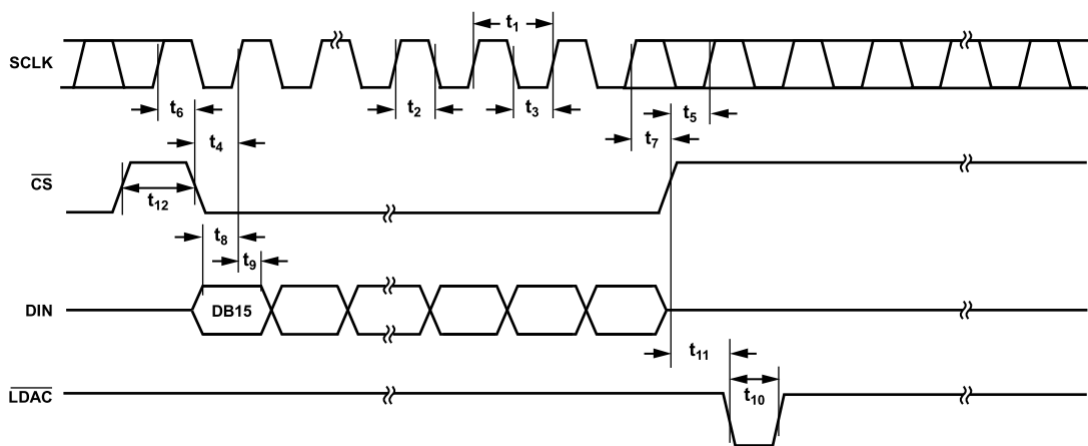
entity MUX8TO1 is
    port ( ...,
          ... ,
          );
end MUX8TO1;

architecture RTL of MUX8TO1 is
    component MUX4TO1
        port(entree : in std_logic_vector(3 downto 0);
              SEL   : in std_logic_vector(1 downto 0);
              Y     : out std_logic);
    end component;
    signal . . .
begin
    . . .
end RTL;
    
```

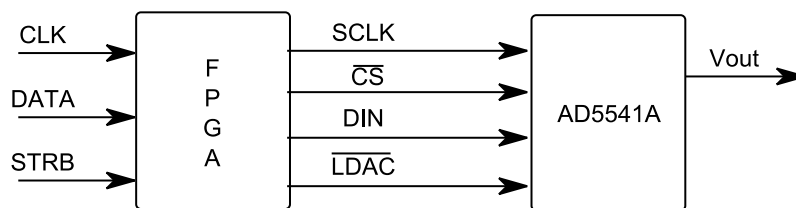
5- Nous disposons d'un convertisseur numérique analogique Analog Device AD5541A présenté ci-dessous:



Pour que le CNA fonctionne correctement, il faudra que nous envoyions à AD5541A une donnée numérique sur 16 bits en respectant le protocole ci-dessous.



Le circuit à mettre en œuvre est le suivant:



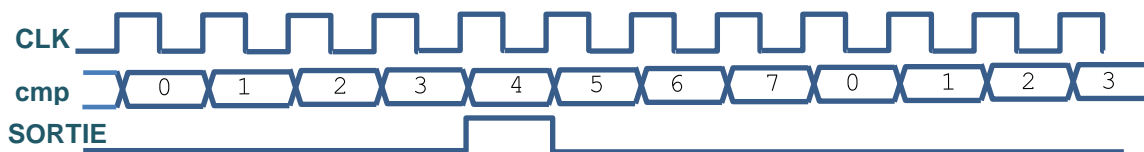
On suppose que votre circuit reçoit en son entrée une horloge qui a la même fréquence que SCLK et que la data : `std_logic_vector(15 downto 0)` est à mémoriser quand STRB est à '1'. On suppose que $T_{10} = T_{11} = T_{12}$ est égal à une période d'horloge. On suppose aussi que le signal SCLK est sortie continuellement.

Corrigé

Question 1

ASIC est privilégié pour de gros tirage parce que le prix revient trop cher pour une faible quantité de production, alors que pour de gros tirage (supérieur à quelques centaines de milliers d'exemplaires) le prix par pièce devient très bon marché.

Question 2



Question 3:

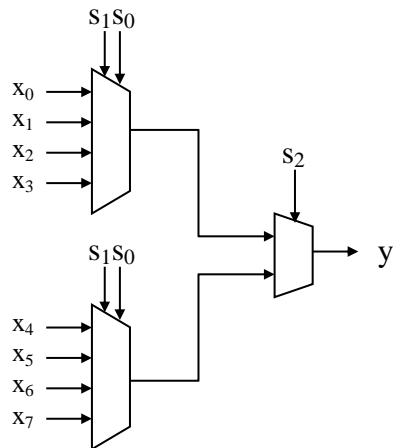
```
entity code_conv is
  Port ( CLK : in  STD_LOGIC;
        STRB : in  STD_LOGIC;
        BN  : in  STD_LOGIC;
        C1N : out  STD_LOGIC;
        C2N : out  STD_LOGIC);
end code_conv;

architecture Behavioral of code_conv is
  signal bn_1, bn_2 : std_logic;
begin
  process (CLK)
  begin
    if CLK'event and CLK='1' then
      if STRB = '1' then
        bn_1 <= BN;
        bn_2 <= bn_1;
      end if;
    end if;
  end process;

  C1N <= BN xor bn_2;
  C2N <= BN xor bn_1 xor bn_2;

end Behavioral;
```

Question 4 :



S ₂	S ₁	S ₀	y
0	0	0	X ₀
0	0	1	X ₁
0	1	0	X ₂
0	1	1	X ₃
1	0	0	X ₄
1	0	1	X ₅
1	1	0	X ₆
1	1	1	X ₇

```

entity MUX8TO1 is
    port( X: std_logic_vector (7 downto 0);
          SEL : std_logic_vector (2 downto 0);
          Y : out std_logic
        );
end MUX8TO1;

architecture RTL of MUX8TO1 is

    component MUX4TO1
        port(entree : in std_logic_vector(3 downto 0);
              SEL    : in std_logic_vector(1 downto 0);
              Y      : out std_logic);
    end component;

    signal IN1, IN2: std_logic;

begin
    U1:  MUX4TO1 port map(entree => X(3 downto 0),
                          SEL => SEL(1 downto 0),
                          Y=>IN1);

    U2:  MUX4TO1 port map(entree => X(7 downto 4),
                          SEL => SEL (1 downto 0),
                          Y=>IN2);

    Y <= IN1 when SEL(2)='0' else IN2;
end RTL;

```

Question 5

```
entity exam is
  Port ( CLK : in  STD_LOGIC;
        DATA : in  STD_LOGIC_VECTOR (15 downto 0);
        STRB : in  STD_LOGIC;
        SCLK : out STD_LOGIC;
        CS : out  STD_LOGIC;
        DIN : out  STD_LOGIC;
        LDAC : out STD_LOGIC);
end exam;

architecture Behavioral of exam is
  signal cmp : integer range 0 to 15;
  signal reg : std_logic_vector (15 downto 0);
  type T_etat is (repos, envoi, tempo, fin);
  signal etat : T_etat := repos;
begin
  SCLK <= not CLK;

  process (CLK)
  begin
    if CLK'event and CLK='1' then
      case etat is
        when repos =>
          CS <= '1';
          LDAC <= '0';
          cmp <= 15;
          DIN <= 'Z';
          if STRB = '1' then
            reg <= DATA;
            etat <= envoi;
          end if;

        when envoi =>
          CS <= '0';
          DIN <= reg(cmp);
          cmp <= cmp - 1;
          if cmp = 0 then
            etat <= tempo;
          end if;

        when tempo =>
          DIN <= 'Z';
          CS <= '1';
          etat <= fin;

        when others => -- etat = fin
          LDAC <= '1';
          cmp <= 15;
          etat <= repos;
      end case;
    end if;
  end process;

end Behavioral;
```