

# Digital Systems

## Mid-semester examination. April 20, 2022

Time limit: 110 minutes.

### 1 Short questions (60 %)

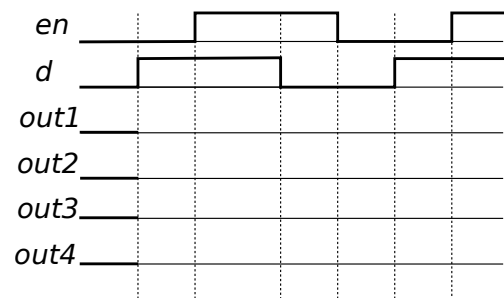
1. Completeu el cronograma de la dreta a partir dels següents processos:

```
-- Process 1
process (en) is
begin
  if en='1' then
    out1 <= not d;
  end if;
end process;

-- Process 2
process (d, en) is
begin
  if en = '1' and d = '0' then
    out2 <= '1';
  else
    out2 <= d;
  end if;
end process;

-- Process 3
process (en) is
begin
  out3 <= '0';
  if en='1' then
    out3 <= d;
  end if;
end process;

-- Process 4
process (d) is
begin
  if en = '1' then
    out4 <= d;
  end if;
end process;
```



2. Quins és el valor en decimal dels següents nombres si usem codificació binaria amb signe (complement a dos)? Indiqueu la seva codificació amb 8 bits.  
101110  
001000
3. Descriviu amb codi VHDL un flip-flop amb enable i reset. El reset ha de ser asíncron i tenir més prioritat que l'enable.

4. Indiqueu si les operacions indicades són vàlides, i en cas de ser vàlides, la codificació en binari i en decimal del resultat.

```

signal a3 : unsigned(2 downto 0) := to_unsigned(4,3);
signal b3 : unsigned(2 downto 0) := to_unsigned(5,3);
signal c3 : unsigned(2 downto 0) := to_unsigned(1,3);
signal u3 : unsigned(2 downto 0);
signal u4 : unsigned(3 downto 0);
signal u5 : unsigned(4 downto 0);
signal u6 : unsigned(5 downto 0);

u3 <= a3+b3;
u4 <= b3+c3;
u5 <= a3*b3;
u6 <= b3*c3;

```

5. És possible escriure un codi VHDL equivalent al següent sense utilitzar el *process*? En cas afirmatiu, feu-ho simplificant-ne al màxim la sintaxi. Dibuixeu també el diagrama de blocs equivalent utilitzant portes lògiques.

```

process (a, b) is
begin
  q <= '1';
  if (a = '0') then
    if (b = '1') then
      q <= '0';
    end if;
  end if;
end process;

```

6. Dibuixeu el cronograma resultant del següent codi VHDL:

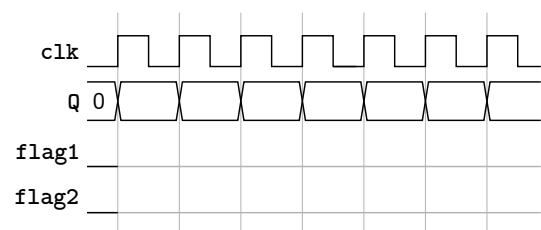
```

signal q: unsigned(1 downto 0) := "00";
signal flag1, flag2: std_logic := '0';

begin
process(clk)
begin
  if rising_edge(clk) then
    q <= "00";
    flag2 <= '0';
    if q < 3 then
      q <= q + 1;
    else
      flag2 <= not flag2;
    end if;
  end if;
end process;

flag1 <= '1' when q < 2 else '0';

```



## 2 Problem: RC5 infrared transmitter (40%)

Al laboratori hem vist com implementar un receptor de senyals infraroigs seguint l'estàndard RC5. L'objectiu d'aquest exercici és dissenyar el sistema que aniria dins un comandament remot per codificar les trames de bits. Recordem que en el protocol RC5 cada bit té una durada d'1.778 ms i el '0' i '1' es codifiquen (codificació de Manchester) enviant senyal infraroig durant la primera o segona meitat del bit, respectivament. Alhora, la informació és agrupada en trames de 14 bits:

- S1 i S2: Bits d'inici que sempre tenen valor '1'
- T: Bit de *toggle* que canvia de valor cada vegada que es prem un botó (inicialitzat a '0').
- Adreça: 5 bits d'identificador del dispositiu que sempre tenen valor 0x00 (TV)
- Comanda: 6 bits d'identificador del canal de TV premut

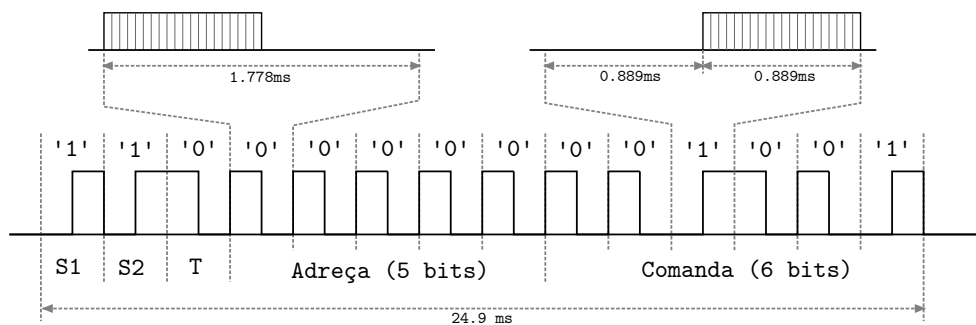


Figure 1: Trama RC5 corresponent al canal 9 en decimal

Dissenyu un mòdul `rc5_transmitter` en VHDL, el qual té les entrades `SEND` (1 bit), `CHANNEL` (6 bits) i `CLK` (1 bit), `clock` de freqüència 1 MHz; i la sortida `RC5_FRAME` (1 bit).

Cada vegada que es rebi un flanc de pujada de `SEND` caldrà transmetre una nova trama RC5 per la sortida `RC5_FRAME`, enviant en els bits de Comanda el valor que hi hagi a `CHANNEL`. El bit de *toggle* s'ha d'invertir en cada nova trama.

Per simplificar el problema podeu considerar que els flancs de pujada de `SEND` estan separats com a mínim 25 ms i que l'entrada `CHANNEL` manté el seu valor entre flancs de pujada de `SEND`.

1. Escriviu l'entity i architecture del sistema. Feu un disseny sintetitzable i síncron. Considereu que el senyal `SEND` és síncron (per fer-lo síncron, proveu de la sortida d'un sincronitzador) i té una durada d'un `CLK`.

```
library ieee; use ieee.std_logic_1164.all, ieee.numeric_std.all;

entity rc5_transmitter is
  port( CLK : in std_logic;
        ...
        RC5_FRAME : out std_logic);
end;

architecture arch of rc5_transmitter is
  signal toggle : std_logic := '0';
  ...
begin ...
```