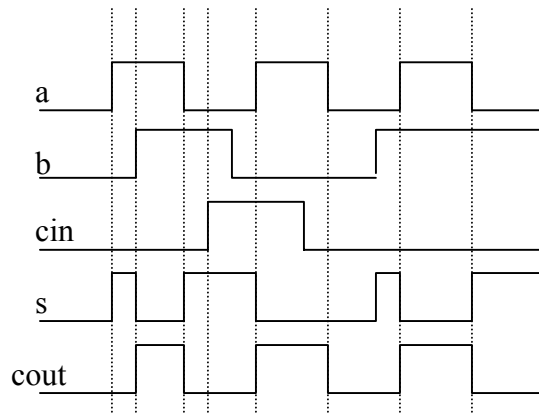


U2.1

a)



b) Insignalen cin saknas i sensitivitetslistan – lägg till den och det fungerar

U2.2

a)  $z = x_0 + x_1'$

b)  $z = x_0' \cdot x_1' \cdot x_2 + x_0' \cdot x_1 \cdot x_2' + x_0 \cdot x_1' \cdot x_2' + x_0 \cdot x_1 \cdot x_2$

c)

| Ma | Mc | Mb | Md | A1 | A0 |
|----|----|----|----|----|----|
| 0  | 0  | 0  | 0  | -  | -  |
| 0  | 0  | 0  | 1  | 0  | 1  |
| 0  | 0  | 1  | 0  | 0  | 1  |
| 0  | 0  | 1  | 1  | 0  | 1  |
| 0  | 1  | 0  | 0  | 1  | 0  |
| 0  | 1  | 0  | 1  | 1  | 0  |
| 0  | 1  | 1  | 0  | 1  | 0  |
| 0  | 1  | 1  | 1  | 1  | 0  |
| 1  | 0  | 0  | 0  | 1  | 1  |
| 1  | 0  | 0  | 1  | 1  | 1  |
| 1  | 0  | 1  | 0  | 1  | 1  |
| 1  | 0  | 1  | 1  | 1  | 1  |
| 1  | 1  | 0  | 0  | 1  | 1  |
| 1  | 1  | 0  | 1  | 1  | 1  |
| 1  | 1  | 1  | 0  | 1  | 1  |
| 1  | 1  | 1  | 1  | 1  | 1  |

Koden är ofullständigt skriven där det inte anges vad A1, A0 ska vara då samtliga ingångar är noll. Den här kretsen har funktionen av en binär prioritetsavkodare så är den ingångskombinationen otillåten. Bortsett från det blir de logiska uttrycken:

De logiska uttrycken för A1 och A0 kan tas fram på t.ex SP normalform från sanningstabellen.

## U2.3

```
Library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity D2_3 is
  port (
    A, B : in  std_logic_vector(7 downto 0);
    Z     : out std_logic_vector(7 downto 0));
end D2_3;
architecture rtl of D2_3 is
begin
  process (A, B)
  begin
    if (A > B) then
      Z <= A;
    else
      Z <= B;
    end if;
  end process;
end rtl;
```

## U2.4

```
Library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity ED_2 is
    port (
        A: in std_logic_vector(7 downto 0);
        Z: out std_logic);
end ED_2;
architecture rtl of ED_2 is
begin
    p1 : process (A)
        variable count : integer;
        variable flag : boolean;
    begin -- process p1
        count := 0;
        flag := false;
        for i in 7 downto 0 loop
            if A(i) = '1' then
                count := count + 1;
            else
                count := 0;
            end if;
            if count > 2 then
                flag := true;
            end if;
        end loop; -- i
        if flag = true then
            Z <= '1';
        else
            Z <= '0';
        end if;
    end process;
end rtl;
```