

## Aufgabenblatt 3: VHDL 2

### Aufgabe 1: Timing

Nachfolgend sehen Sie zwei VHDL-Beschreibungen mit gemischten Signal- und Variablen-Anweisungen. Tragen Sie die Werte von b, c, d für alle angegebene Zeitpunkte in die entsprechenden Tabellen ein. Die geplanten Zuweisungen an das *Signal a* sind in die Tabellen bereits eingetragen worden. *Hinweis: Beachten Sie den Unterschied bei einer Signal- bzw. Variablenzuweisung!*

a)

```
architecture A of Aufgabe1 is
  signal a, b: integer := 0;
begin
  a <= 2 after 1 ns, 4 after 2 ns,
      5 after 3 ns;
  process (a, b)
    variable c, d: integer := 0;
  begin
    b <= a + 1;
    c := b + 2;
    d := a + b + c;
  end process;
end A;
```

	0	1	2	3	t
a	0	2	4	5	
b					
c					
d					

b)

```
architecture B of Aufgabe1 is
  signal a, b : integer := 0;
begin
  a <= 13 after 1 ns, 7 after 2 ns;
  process (a, b)
    variable c, d : integer := 0;
  begin
    c := 1 + a ;
    b <= a - c + 3 ;
    d := b - 3 ;
    b <= a + c ;
    c := a - b - d ;
  end process;
end B;
```

	0	1	2	t
a	0	13	7	
b				
c				
d				

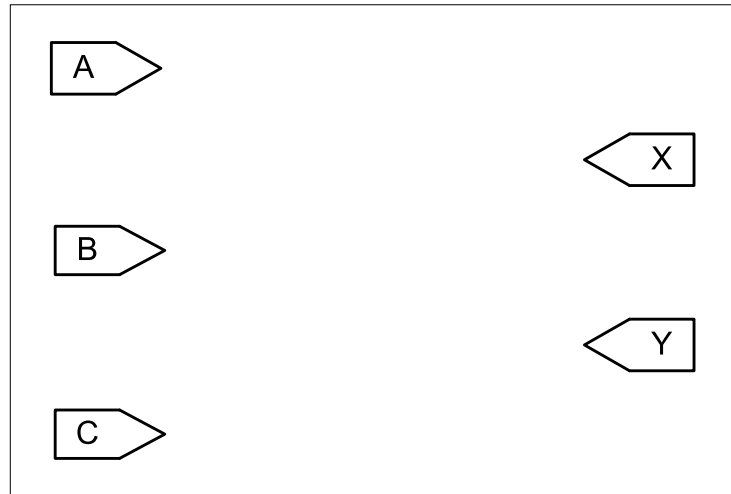
### Aufgabe 2: Gatterschaltung

Geben Sie für die unten angegebenen VHDL Programme jeweils eine **vereinfachte** Gatterschaltung an, die das gleiche Ergebnis liefert. Dabei stehen Ihnen folgende Gattertypen zur Verfügung: AND, OR, Inverter, NAND, NOR, XOR,  $U_{DD}$  (= '1'), GND (= '0').

a)

```
entity C_ent is
  port( A, B, C : in  std_logic;
        X, Y : out std_logic );
end C_ent;

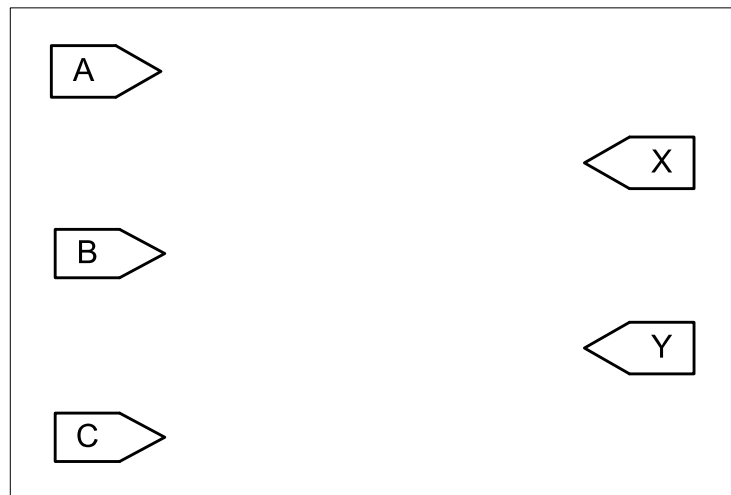
architecture C of C_ent is
  signal D: std_logic;
begin
  P : process (A, B, C, D)
  begin
    D <= A and B and C;
    X <= (not B) and D;
    D <= C xor 1;
    Y <= C or D;
  end process;
end C;
```



b)

```
entity D_ent is
  port( A, B, C : in  std_logic;
        X, Y : out std_logic );
end D_ent;

architecture D of D_ent is
begin
  VAR:process (A, B, C)
  variable D: std_logic;
  begin
    D := A and B and C;
    X <= (not B) and D;
    D := C xor 1;
    Y <= C or D;
  end process;
end D;
```



### Aufgabe 3: Automat

Aufgabe dieser Übung ist es ein VHDL-Programm zu entwickeln, das den in der Abbildung 1 dargestellten Zustandsübergangsgraphen eines **Moore-Automaten** beschreibt.

- Beschreiben Sie die *entity* mit allen benötigten In- und Out-Ports.
- Geben Sie die *architecture* zur der - in Teil a) beschriebenen - *entity* an.

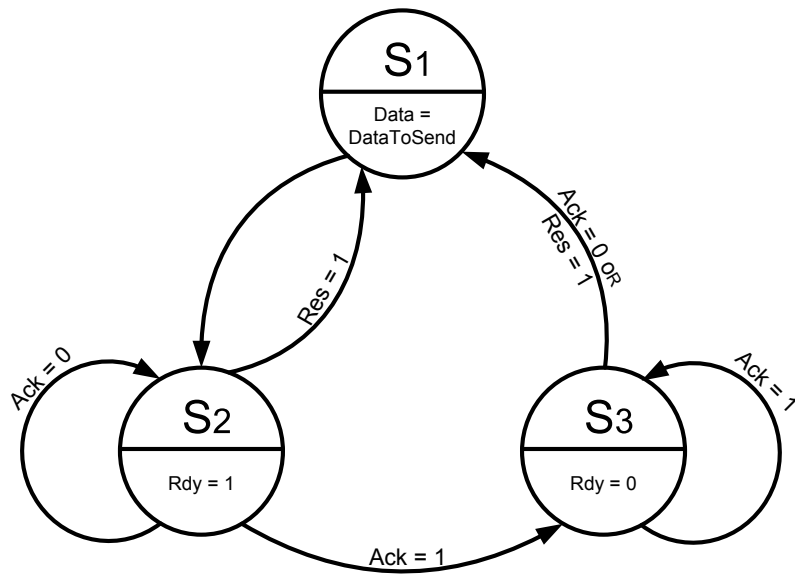


Abbildung 1: Moore-Automat