

Quartus



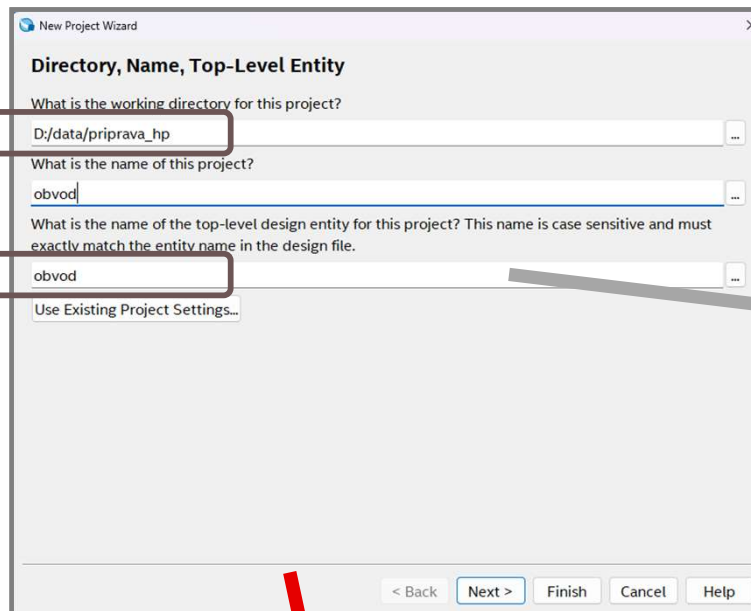
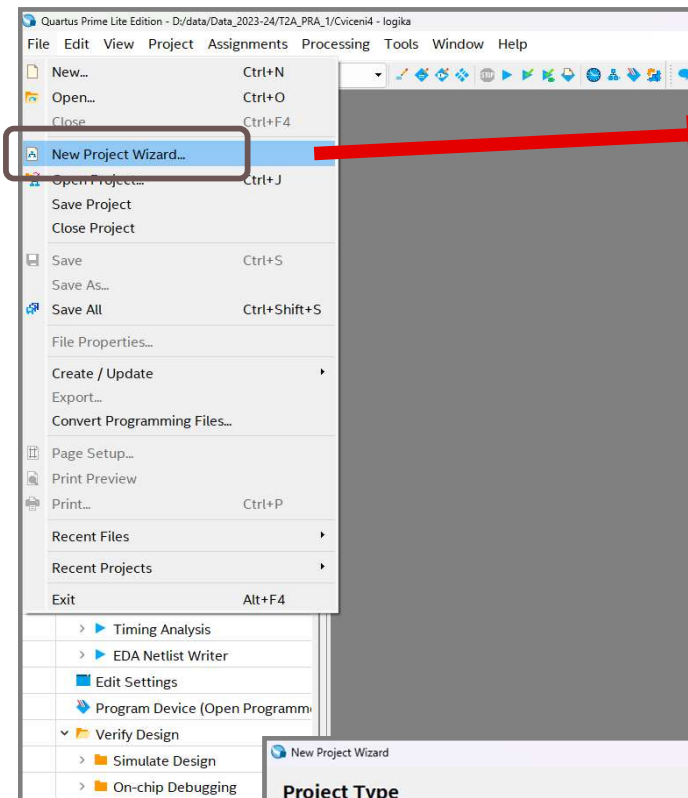
Vytvoření aplikace



Postup vytvoření aplikace

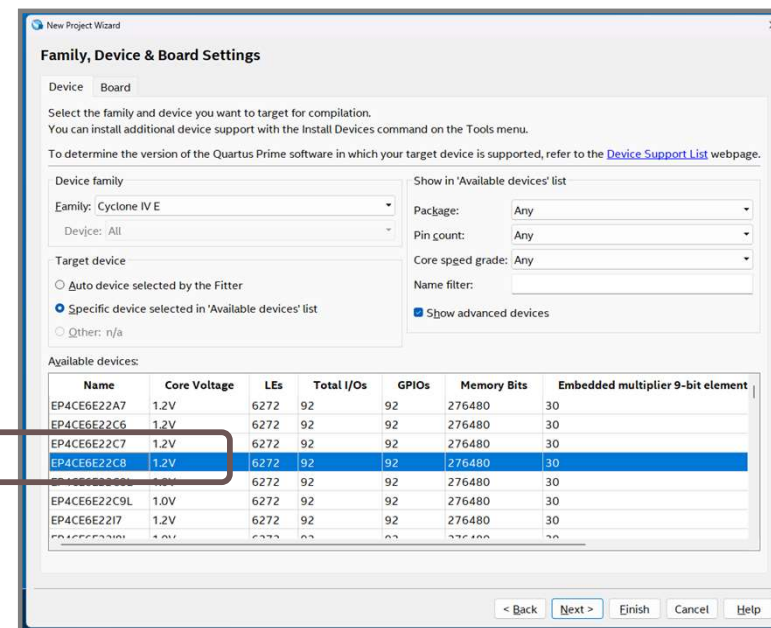
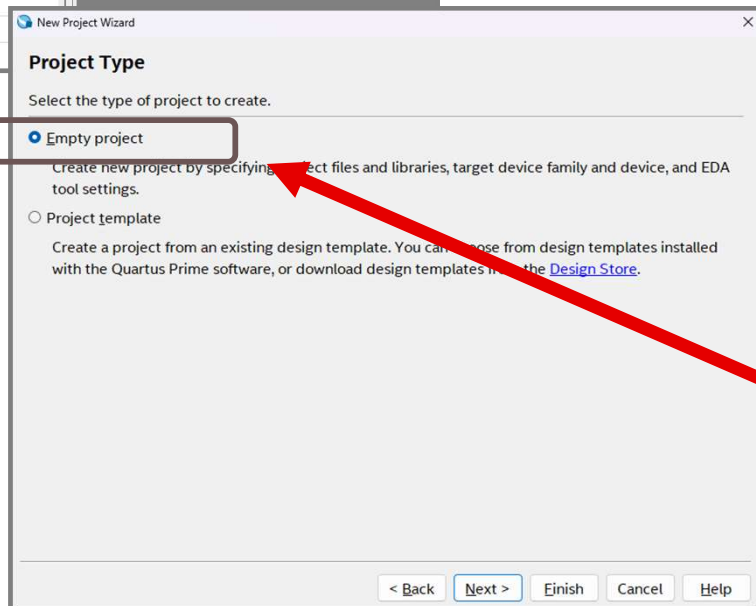
1. Umístění projektu, volba typu HW
2. Volba typu aplikace (schéma nebo VHDL)
3. Kontrola syntaxe
4. Připojení markerů k vývodům FPGA
5. Generování programovacího souboru
6. Nahrání do přípravku

1. Umístění projektu a volba HW

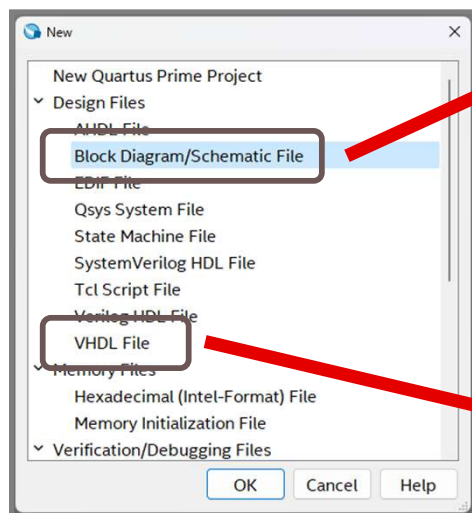
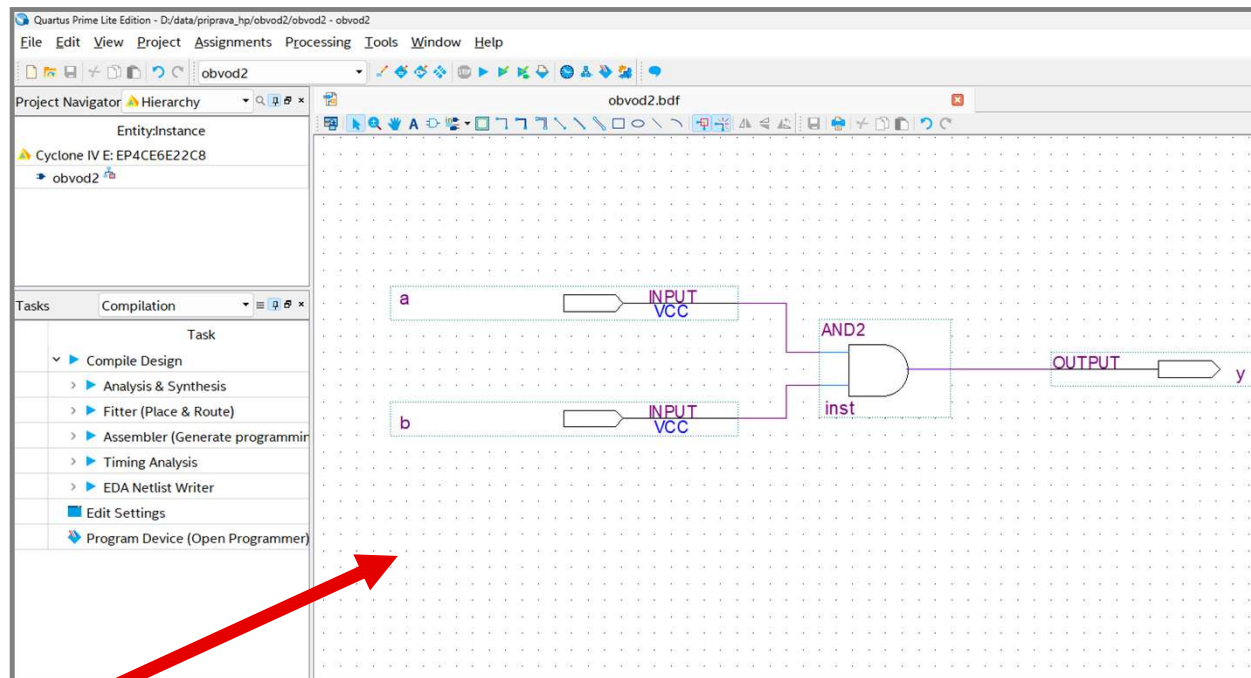
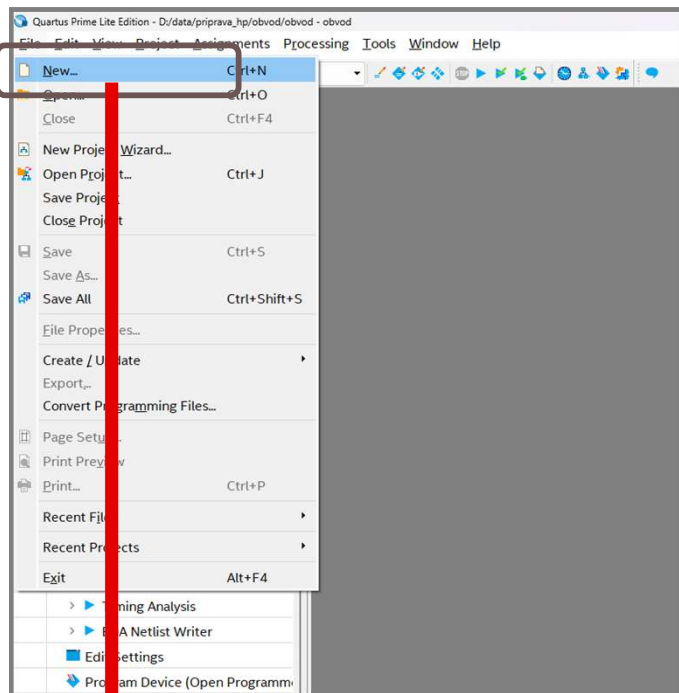


Top level
design entity
Nastavení
v souboru VHDL

Cyclone IV E:
EP4CE6E22C8



2. Volba typu aplikace



This screenshot shows the VHDL editor in Quartus Prime Lite Edition. It displays the VHDL code for the 'obvod2' entity. The code is as follows:

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_ARITH.ALL;
4 use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6 entity obvod is
7     Port (
8         a : in std_logic;
9         b : in std_logic;
10        y : out std_logic
11    );
12 end obvod;
13
14 architecture Behavioral of obvod is
15     -- signals
16 begin
17     y <= a and b;
18
19 end Behavioral;
```

3. Kontrola syntaxe

Quartus Prime Lite Edition - D:/data/priprava_hp/obvod2/obvod2 - obvod2

File Edit View Project Assignments Processing Tools Window Help

obvod2

Project Navigator Hierarchy

Entity:Instance

Cyclone IV E: EP4CE6E22C8

obvod2

Tasks

Compilation

Task

Compile Design

47% > Analysis & Synthesis

> Fitter (Place & Route)

> Assembler (Generate programming file)

> Timing Analysis

> EDA Netlist Writer

Edit Settings

Program Device (Open Programmer)

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
 - Flow Messages
 - Flow Suppressed Messages

Flow Summary

<<Filter>>

Flow Status	In progress - Sun Oct 15 14:48:36 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	obvod2
Top-level Entity Name	obvod2
Family	Cyclone IV E

Messages

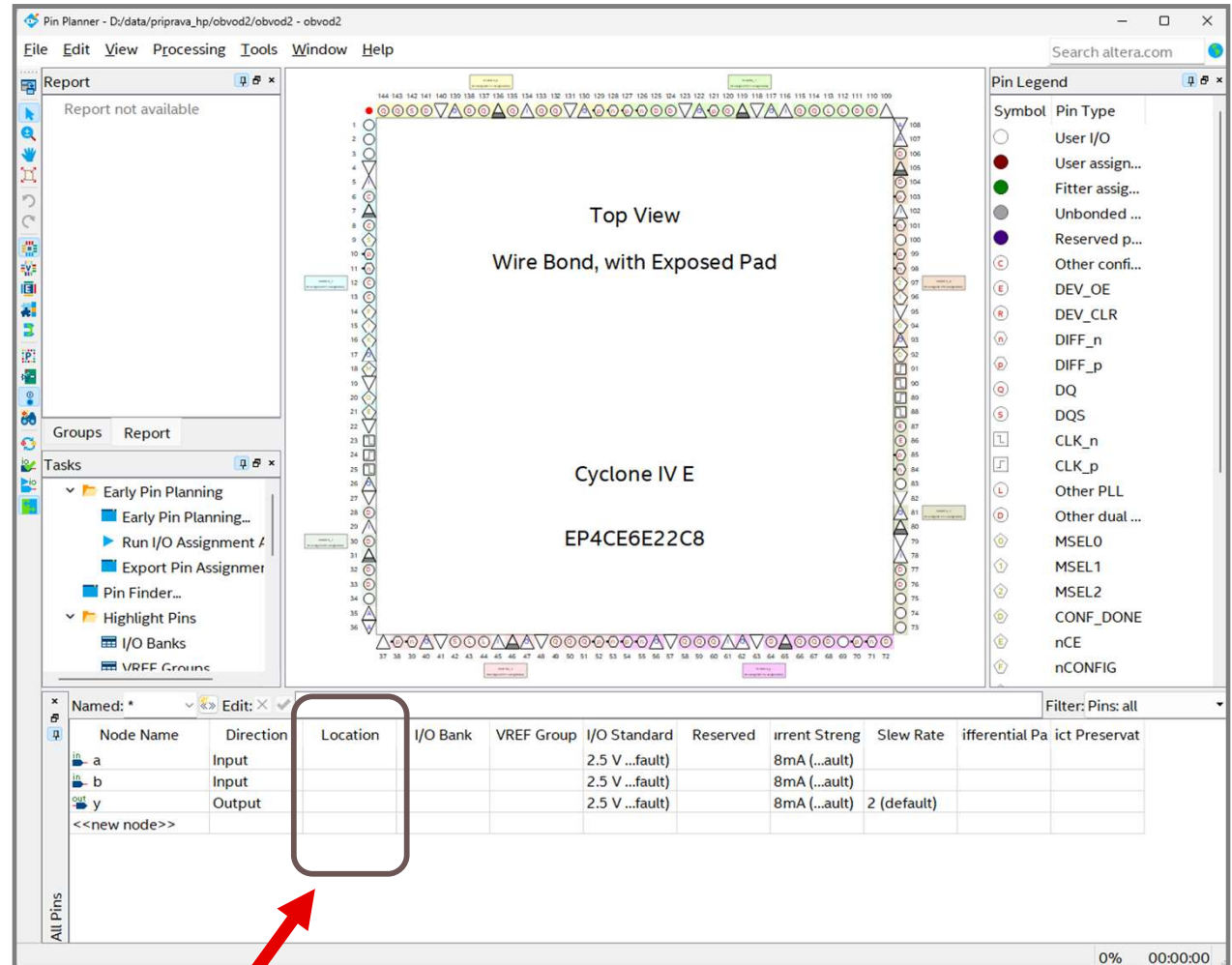
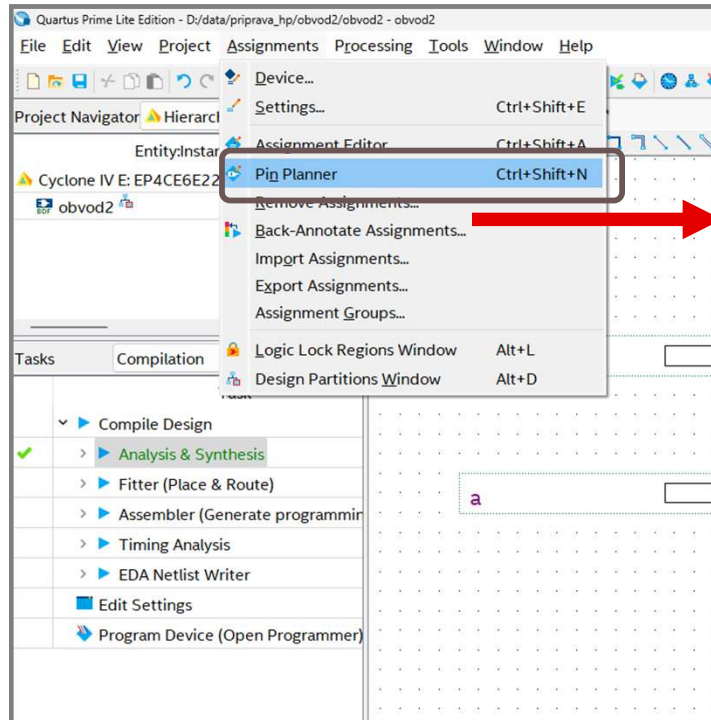
All Find... Find Next

Type ID Message

- 20030 Parallel compilation is enabled and will use 4 of the 4 processors detected
- > 12021 Found 1 design units, including 1 entities, in source file obvod2.bdf
- 12127 Elaborating entity "obvod2" for the top level hierarchy

System Processing (7)

4. Připojení markerů k vývodům FPGA



a - 88
b - 89
y - 87

5. Generování programovacího souboru

Quartus Prime Lite Edition - D:/data/priprava_hp/obvod2/obvod2 - obvod2

File Edit View Project Assignments Processing Tools Window Help

obvod2

Project Navigator Hierarchy

Entity: Instance

Cyclone IV E: EP4CE6E22C8

obvod2

Tasks Full Design

Task

- > Start Project
- > Create Design
- > Assign Constraints
- > Compile Design
 - > Analysis & Synthesis
 - > Fitter (Place & Route)
 - > Assembler (Generate programn)
 - > Timing Analysis
- > EDA Netlist Writer
- Edit Settings
- Program Device (Open Programm
- > Verify Design
 - > Simulate Design

4%

0%

0%

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Sett
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
 - Flow Messages
 - Flow Suppressed Messages

Flow Summary

<<Filter>>

Flow Status	In progress - Sun Oct 15 17:18:47 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	obvod2
Top-level Entity Name	obvod2
Family	Cyclone IV E

Messages

All

<<Filter>>

Find... Find Next

Type	ID	Message
Information		Command: quartus_map --read_settings_files=on --write_settings_files=off obvod2 -c obvod2
Warning	18236	Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment
Information	20030	Parallel compilation is enabled and will use 4 of the 4 processors detected

System Processing (5)

6. Nahrání do přípravku

Pokud není zvolen žádný HW, je třeba vybrat z nabídky
Hardware Setup

The screenshot displays the Quartus Prime Lite Edition interface. The main window shows the 'Program Device (Open Programmer)' button highlighted. A 'Hardware Setup' dialog box is open, showing the 'USB-Blaster [USB-0]' hardware item selected. The 'Flow Status' window is also visible in the background.

Hardware Setup Dialog Box:

- Hardware Settings | JTAG Settings
- Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.
- Currently selected hardware: USB-Blaster [USB-0]
- Available hardware items:

Hardware	Server	Port
USB-Blaster	Local	USB-0

Buttons: Add Hardware..., Remove Hardware, Close

Flow Status Window:

- Flow Status: Successful - Sun Oct 15 17:46:44 2023
- Programmer: D:/data/priprava_hp/obvod2/obvod2 - obvod2.cdf
- Mode: JTAG
- Progress: []

Task List:

- Start Project
- Create Design
- Assign Constraints
- Compile Design
 - Analysis & Synthesis
 - Fitter (Place & Route)
 - Assembler (Generate program files)
 - Timing Analysis
 - EDA Netlist Writer
- Program Device (Open Programmer)
- Verify Design
- Simulate Design

Aplikace VHDL



Základní pojmy

Zdrojový soubor

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

Knihovny

```
entity logika is
```

```
    Port (
```

Vstupy a
výstupy

```
    );  
end logika;
```

```
architecture Behavioral of logika is
```

Signály

```
begin
```

Definice
struktury HW

```
end Behavioral;
```

Základní pravidla

Jednoduchý vstup a výstup

a : **in** std_logic;

Název: a
Směr: in

y : **out** std_logic;

Název: y
Směr: out

Sběrnice vstup a výstup

adresa : **in** std_logic_vector(7 downto 0);

Název: **adresa**
Směr: in
Šířka: 8 bitů

data : **out** std_logic_vector(3 downto 0);

Název: **adresa**
Směr: in
Šířka: 8 bitů

Signály

citac : std_logic;

Název: **citac**
Šířka: 5

citac : std_logic:= '0';

citac : std_logic_vector(4 downto 0)="10010";

citac : std_logic_vector(x downto 0)=(others =>'0');

Název: **citac**
Šířka: libovolná

Přiřazení <=

```
y <= a and b;
```

Základní hradla: and, or, xor, not

Sloučení

```
b <= "0" & a & c(7 downto 1);
```

C šířka 6 bitů (bity 1-7)

A šířka 1 bit

B šířka 8 bitů

konstanta 2 bity

Rozdělení

```
b <= c(7);
```

C šířka 1 bit (7.bit)

B šířka 1 bit

Číselné soustavy

```
"0";
```

```
x"9";
```




Asynchronní podmínky

Select /When Statement

Select /When

```
with vstup select
    výstup <= hodnota1 when hodnota_vstupu1,
    hodnota2 when hodnota_vstupu2,
    hodnota3 when others;
```

Dekodér

```
with vstup select
    y <= "0001" when "00",
    "0010" when "01",
    "0100" when "10",
    "1000";
```

x: vstup
y: výstup

When /Else Statement

When /Else

```
výstup <= hodnota1 when vstup=hodnota_vstupu1 else  
          hodnota2 when vstup=hodnota_vstupu2 else  
          hodnota3;
```

Dekodér

```
y <= "0001" when x="00" else  
     "0010" when x="01" else  
     "0100" when x="10" else  
     "1000";
```

x: vstup
y: výstup



Synchronní podmínky

Case

Case

```
process(clk)  
begin  
    if ( clk'event and clk ='1') then  
        case (<2-bit select>) is  
            when "00" =>  
                <statement>;  
            when "01" =>  
                <statement>;  
            when "10" =>  
                <statement>;  
            when "11" =>  
                <statement>;  
            when others =>  
                <statement>;  
        end case;  
    end if;  
end process;
```

Case - dekodér

Case

x: vstup
y: výstup

```
process(x)
begin
    if ( clk'event and clk = '1') then
        case (x) is
            when "00" =>
                y <= "1000";
            when "01" =>
                y <= "1000";
            when "10" =>
                y <= "1000";
            when others =>
                y <= "1000";
        end case;
    end if;
end process;
```


If/Elself/Else

If/Elself/Else

```
process(clk)  
begin  
    if ( clk'event and clk ='1') then  
        if <condition> then  
            <statement>  
        elsif <condition> then  
            <statement>  
        else  
            <statement>  
        end if;  
    end if;  
end process;
```

If/Elsif/Else - dekodér

If/Elsif/Else

x: vstup
y: výstup

```
process(clk)  
begin  
    if ( clk'event and clk ='1') then  
        if x="00" then  
            y <= "1000"  
        elsif x="01" then  
            y <= "0100"  
        elsif x="10" then  
            y <= "0010"  
        else  
            y <= "0001"  
        end if;  
    end if;  
end process;
```



Kombinační obvody

Process

```
process(<citlivostní seznam>)  
begin  
    --tělo procesu  
end process;
```

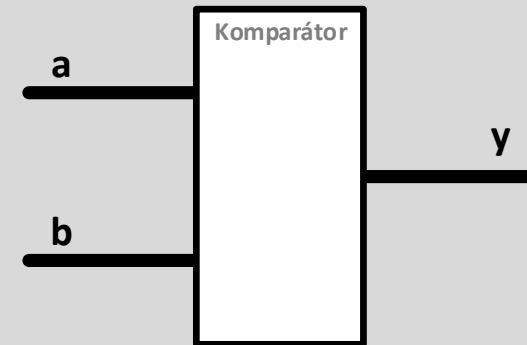
Proces – synchronní struktura

Citlivostní seznam – Proces se spustí, když dojde ke změně úrovně veličiny v citlivostním seznamu

Tělo procesu – V těle jsou příkazy popisující zapojení obvodu

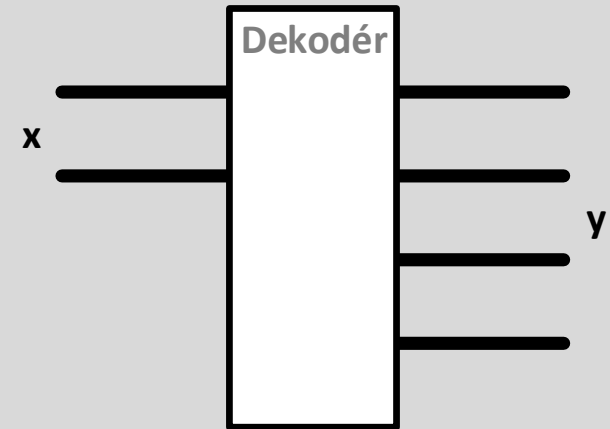
Komparátor - If/Elseif/Else

```
process(a,b,clk)
begin
    if ( clk'event and clk = '1' ) then
        if (a=b) then
            b <= '1';
        else
            b <= '0';
        end if;
    end if;
end process;
```



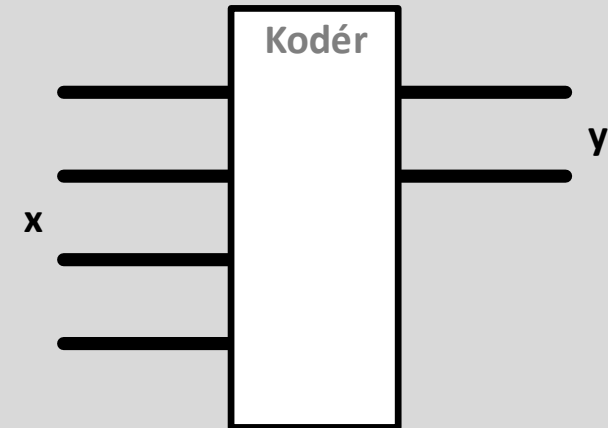
Dekodér - Case

```
process(x,clk)
begin
    if (clk'event and clk = '1') then
        case (x) is
            when "00" =>
                y <= "0001";
            when "01" =>
                y <= "0010";
            when "10" =>
                y <= "0100";
            when others =>
                y <= "1000";
        end case;
    end case;
end process;
```



Kodér - Case

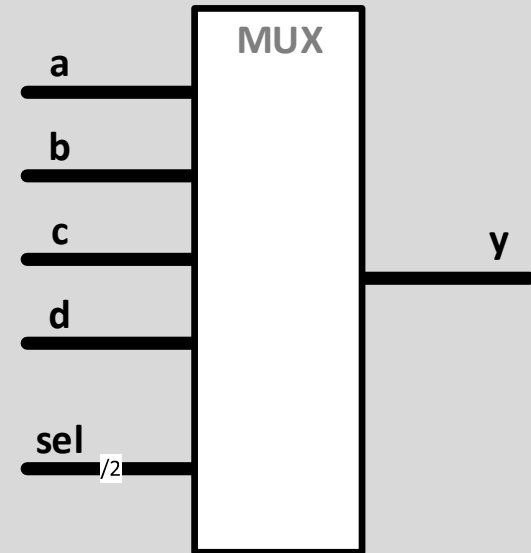
```
process(x,clk)
begin
    if (clk'event and clk = '1') then
        case (x) is
            when "0001" =>
                y <= "00";
            when "0010" =>
                y <= "01";
            when "0100" =>
                y <= "10";
            when "1000" =>
                y <= "11";
            when others =>
                y <= "00";
        end case;
    end case;
end process;
```



Multiplexer – různé konstrukce

When/Else

```
y <= a when sel="00" else  
     b when sel="01" else  
     b when sel="01" else  
     d;
```

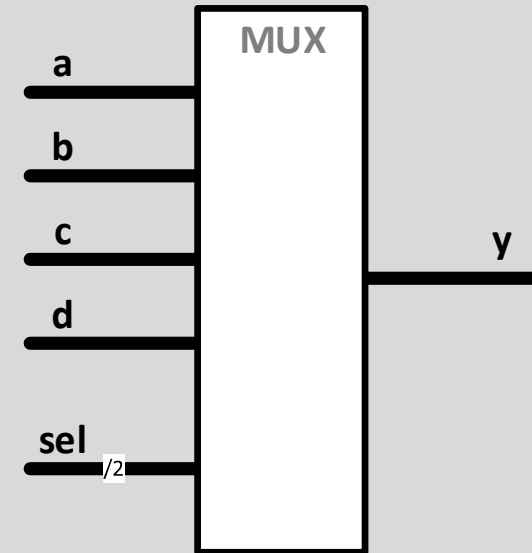


Select /When

```
with sel select
```

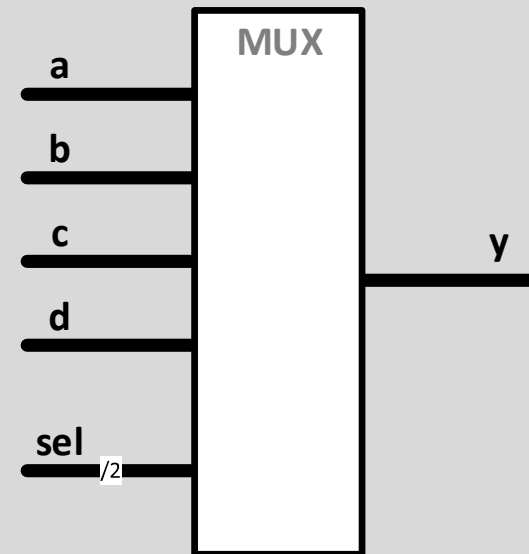
```
  y <=
```

```
    a when "00" else  
    b when "01" else  
    b when "01" else  
    d;
```



Synchronní/Case

```
process(a,b,c,d,sel)
begin
    case sel is
        when "00" =>
            y <=a;
        when "01" =>
            y <=b;
        when "10" =>
            y <=c;
        when others
            y <=d;
    end case
end process;
```



Synchronní/If

```
process(a,b,c,d,sel)
begin
    if sel="00" then
        y <=a;
    elsif sel="01" then
        y <=b;
    elsif sel="10" then>
        y <=c;
    else
        y <=d;
    end case
end process;
```

