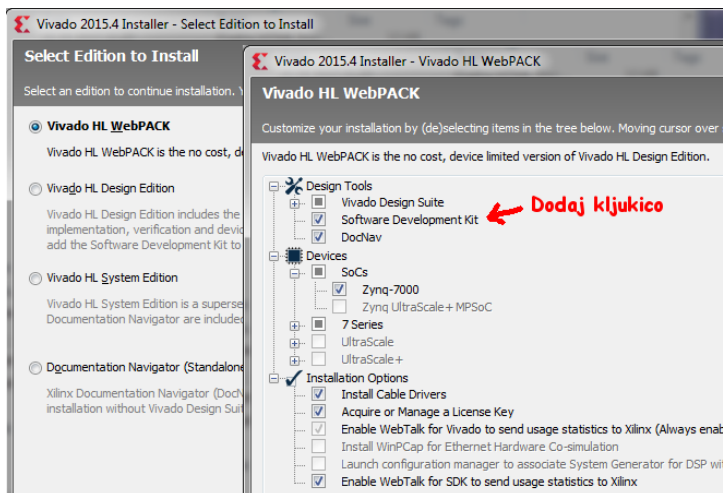


Načrtovanje vezja v programu Vivado

Za načrtovanje digitalnih vezij bomo uporabljali razvojno orodje Xilinx Vivado 2015.4 WebPACK, ki je brezplačno dostopno na <http://www.xilinx.com/support/download.html>

1. Namestitev

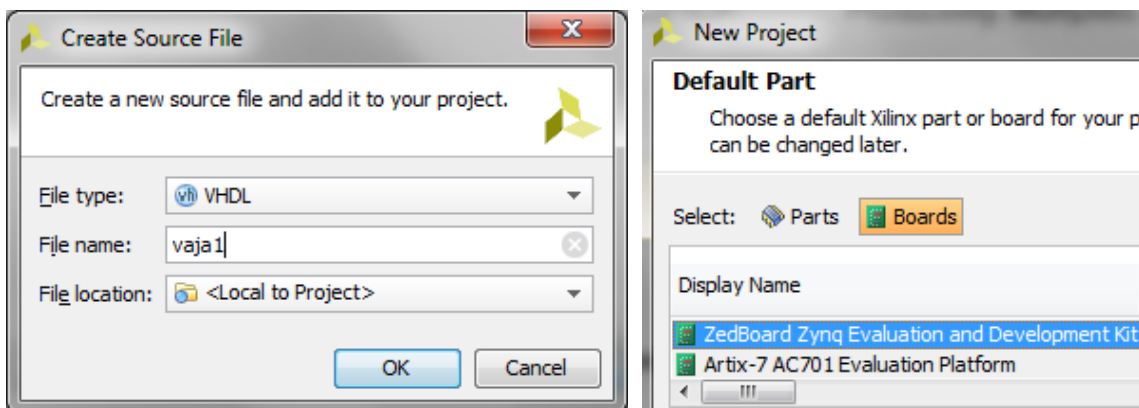
Vivado 2015 podpira Windows 7/8 in različne verzije sistema Linux, za novejšje Windowse pa potrebujete Vivado 2016. Pred prenosom datotek in namestitvijo se je potrebno na strani Xilinx brezplačno registrirati. Za prenos priporočam uporabo klienta (Vivado HLx Web Install Client). Pri namestitvi izberite opcijo HL WebPACK (ta ima brezplačno licenco) in Software Development Kit, če potrebujete tudi SDK za programiranje Zynq:



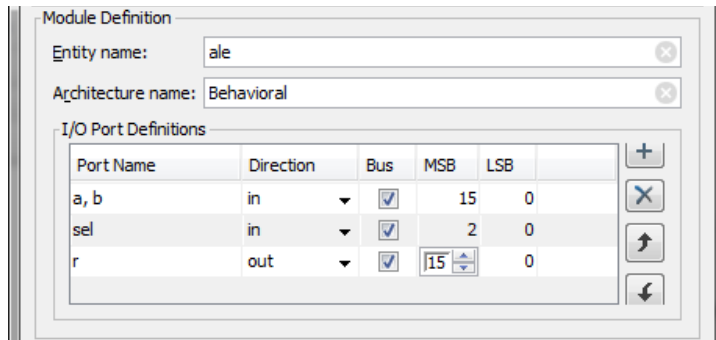
Program Vivado 2015 zasede na disku okoli 7 GB, SDK pa prav tako 7 GB.

2. Izdelava projekta

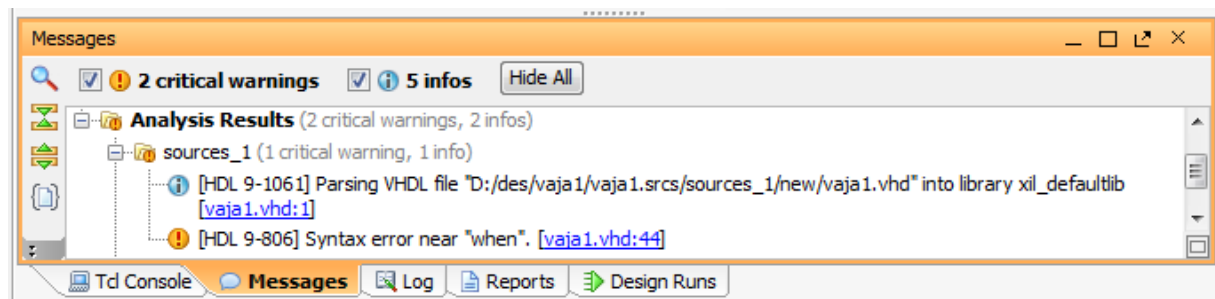
Najprej naredimo nov projekt **Create New Project**, določimo ime (npr. vaja1) in lokacijo, ki naj bo na D:\des\moja mapa. V naslednjem oknu izberemo **RTL Project**, nato pa v naslednjem oknu kliknemo na **Create File** ter določimo vrsto: VHDL in ime datoteke. V zadnjem oknu izberemo FPGA vezje ali ploščo. Izbrali bomo razvojno ploščo *Zedboard*, ki ima podobno vezje kot Red Pitaya.



Pred zaključkom se pojavi še tabela v kateri določimo imena in vrsto vhodnih in izhodnih signalov:



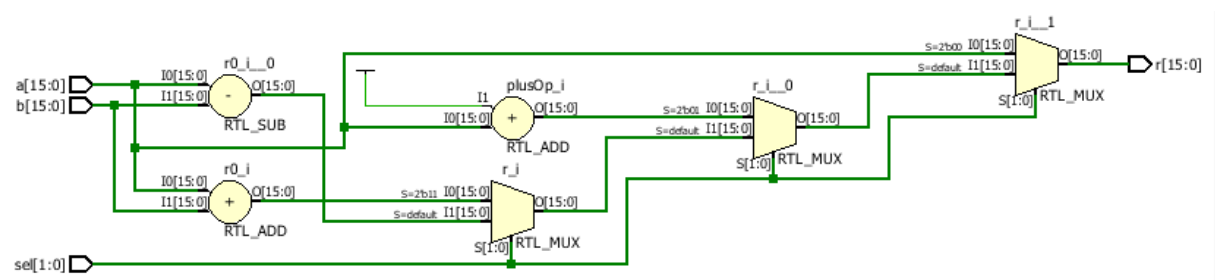
Na podlagi vnešenih podatkov program naredi datoteko z ogrodjem opisa vezja v jeziku VHDL. Opremo jo z dvoklikom na ime datoteke v seznamu Sources in dokončamo opis vezja. Ko shranimo spremembe (ctrl+s) se izvede analiza sintakse in izpiše morebitne napake v zavihku Messages (**Analysis Results**). V primeru napak se datoteka premakne v podmapo Syntax Error Files(1) in napako moramo pred nadaljevanjem prevajanja popraviti.



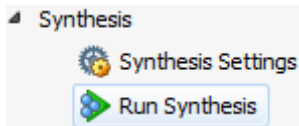
3. Elaboracija in sinteza

Elaboracijo vezja naredimo, če nas zanima shema vezja na nivoju RTL. Iz menija **Flow** izberemo **Open Elaborated Design**. Primer sheme enote ALE, ki je opisana s stavkom:

```
r <= a when sel="00" else a+1 when sel="01" else a+b when sel="10" else a-b;
```



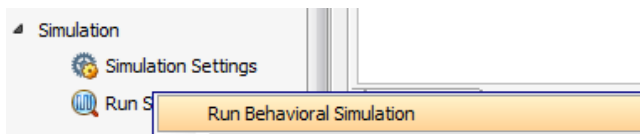
Na shemi vidimo vhodne in izhodne signale ter osnovne gradnike vezja: seštevalnike, odštevalnike, izbiralnike...



S klikom na **Run Synthesis** izvedemo sintezo logičnega vezja. V primeru napak se program ustavi in javi napake, ki jih pregledamo v zavihku Messages, sicer pa odpre okno v katerem izberemo nadaljevanje postopka prevajanja (**Run Implementation**) ali pa pregled poročila (**View Reports**).

V poročilu **Vivado Synthesis Report** so zabeleženi vsi koraki avtomatične sinteze vezja, statistika komponent na nivoju RTL (število in vrsta sintetiziranih komponent) in poročilo o uporabljenih celicah vezja FPGA (npr. vpogledne tabele LUT, vhodni IBUF in izhodni ojačevalniki OBUF). Poročilo **Utilization Report** pa podaja zasedenost zasedenost komponent izbranega vezja FPGA.

4. Simulacija vezja



Simulacijo poženemo z ikono **Run Simulation** in izbiro **Run Behavioral Simulation**. V simulatorju z desnim klikom na vhodni signal nastavimo vrednost signala (**Force Constant** ali **Force Clock**), poženemo simulator za določen čas (**Run For**) in opazujemo izhode.

