

# 3. vaja: Uvod v visokonivojsko sintezo

Visokonivojska sinteza je avtomatiziran proces načrtovanja vezij, kjer se abstrakten model vezja v jeziku C pretvori v opis vezja na ravni registrov (VHDL ali Verilog). Abstrakten model vsebuje opis funkcionalnosti, časovne parametre pa določi sintetizator.

## 3.1 Algoritem množenja kompleksnih števil

Naredimo vezje, ki sprejme dve kompleksni številici in izračuna njun produkt. Algoritem je opisan s funkcijo `op()`, v datoteki `op.cpp`. Model vezja vsebuje še zaglavno datoteko `op.h` in testno datoteko z glavno funkcijo, ki vsebuje kodo za preizkus algoritma (Test Bench).

```
#include "op.h"

void op(DAT_T a, DAT_T b, DAT_T c,
        DAT_T d, REZ_T &x, REZ_T &y)
{
    x = a * c - b * d;
    y = a * d + b * c;
}

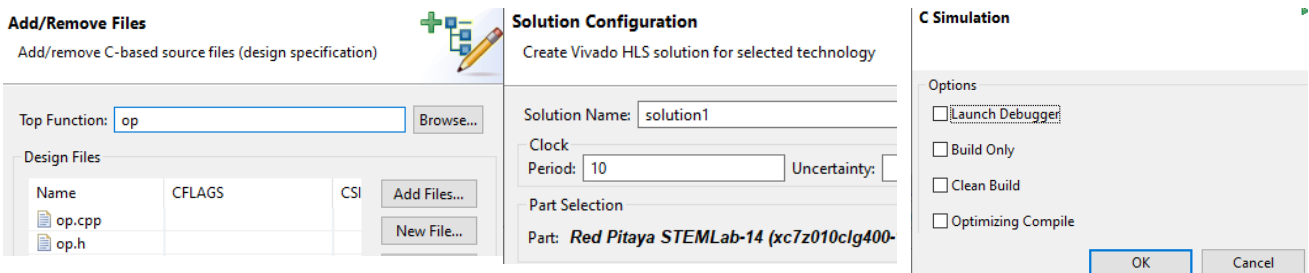
#ifdef OP_H
#define OP_H

#include <ap_int.h>
#define DAT_T ap_int<16>
#define REZ_T ap_int<32>

void op(DAT_T a, DAT_T b, DAT_T c,
        DAT_T d, REZ_T &x, REZ_T &y);

#endif
```

V knjižnici `ap_int.h` so definirani celoštevilski podatkovni tipi poljubne natančnosti. Vhode smo definirali kot 16 bitne celoštevilске vrednosti, izhode pa kot 32 bitne (zaradi množenja).



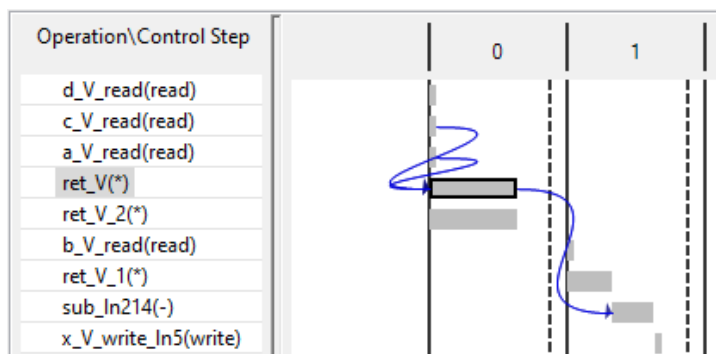
1. Na računalniku naredi projektno mapo z datotekami v jeziku C. Odpri orodje **Vivado HLS** in naredi nov projekt. Dodaj datoteko z opisom funkcije in zaglavno datoteko, vnesi ime funkcije (Top Function), v naslednjem oknu pa dodaj datoteko s testno strukturo. Nazadnje določi parametre sinteze (privzeto) in ciljno vezje oz. razvojno ploščo.
2. Preizkusi delovanje algoritma s simulacijo. Dopolni testno datoteko s kodo, ki nastavi vrednosti vhodov, pokliče funkcijo in izpiše izhode.
3. Izvedi sintezo vezja in pregled rezultate: zakasnitve, porabo FPGA in časovni potek (okno Analysis).

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
1	1	10.000 ns	10.000 ns	1	1	none

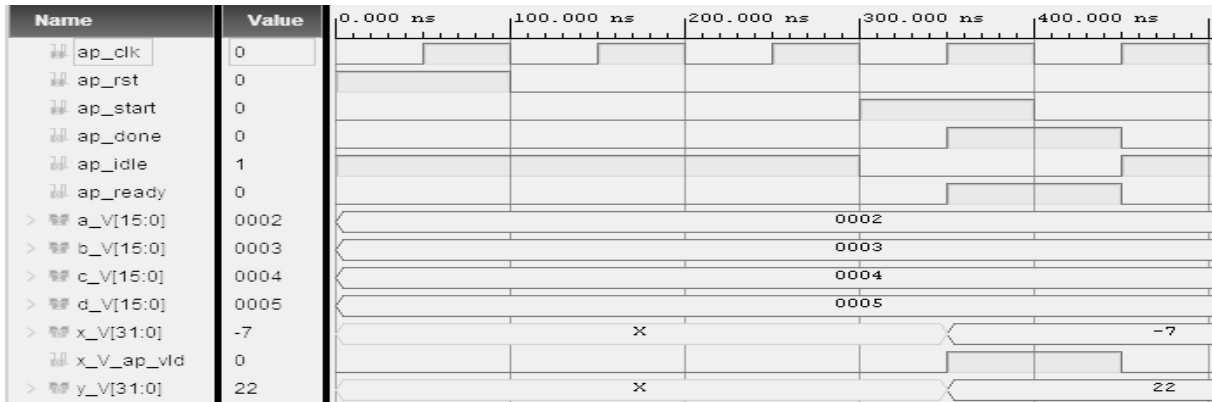
**Utilization Estimates**

Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	4	-	-	-
Multiplexer	-	-	-	15	-
Register	-	-	130	-	-
<b>Total</b>	<b>0</b>	<b>4</b>	<b>130</b>	<b>15</b>	<b>0</b>
Available	120	80	35200	17600	0
Utilization (%)	0	5	~0	~0	0



- Preglej sintetiziran RTL model (VHDL, Verilog). Preveri delovanje z izvedbo kosimulacije v orodju Vivado HLS in s simulacijo v Vivadu ali drugem RTL simulatorju (naredi nov projekt).



- Spremeni podatkovni tip vhodnih in izhodnih signalov in preveri, kako vpliva na rezultat sinteze vezja. Namig: preizkusi različne velikosti celoštevilskih vektorjev, realna števila in števila s fiksno vejico, ki so definirana v knjižnici `ap_fixed.h` (npr. `ap_fixed<16,12>` je 16 bitno število z 12 biti celoštevilskega dela).

podatkovni tip	latenca / interval	DSP48E	FF	LUT

Kateri podatkovni tip je najbolj primeren za operacije z realnimi števili na vezju FPGA?

### 3.2 Deljenje kompleksnih števil in direktive

$$\frac{a + bi}{c + di} = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2} i$$

- Spremeni algoritem, da bo izračunal rezultat deljenja dveh kompleksnih števil in preglej rezultate sinteze. Za računanje uporabi števila v fiksni decimalki z 8 biti za celi del in 4 biti za decimalke. Iz poročila analize vezja ugotovi katera operacija porabi največ ciklov ure.
- Dodaj direktivo, da bo sintetizirano vezje v cevovodni obliki. Kaj se zgodi s časovnimi parametri po sintezi in število katerih gradnikov vezja se bo bistveno povečalo?
- Naredi model vezja, ki izvaja množenje in deljenje kompleksnih števil. Definiraj nov vhodni signal tipa bool za izbiro operacije. Izvedi simulacijo in sintezo vezja in preglej rezultate.

vezje	latenca / interval	DSP48E	FF	LUT