



Laboratorij za načrtovanje integriranih vezij

Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*



Digitalni Elektronski Sistemi

Osnove jezika VHDL

Načrtovanje vezja na nivoju RTL

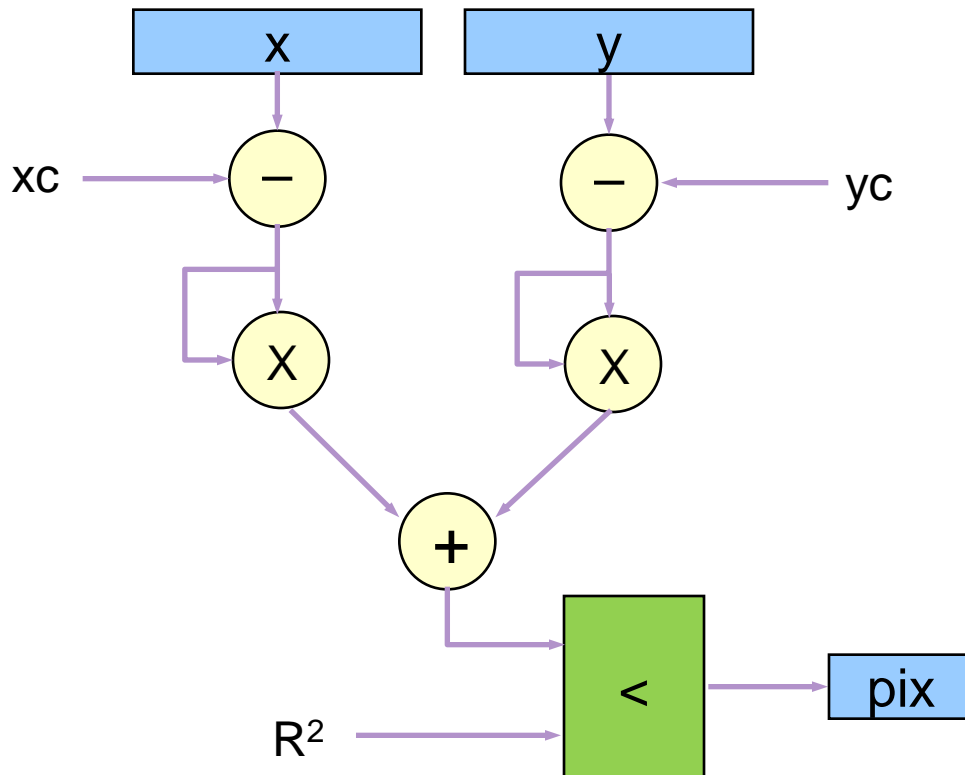
Model vezja na nivoju RTL

- ▶ Vezja za obdelavo podatkov izvajajo nek algoritem
 - ▶ algoritem opišemo kot zaporedje (računskih) korakov
- ▶ Algoritem določa obnašanje vezja
 - ▶ orodja za avtomatično sintezo vezja iz algoritma (C to RTL)
 - ▶ pri algoritmu ni določen časovni potek izvajanja
- ▶ Na nivoju RTL je določeno obnašanje vezja ob urnih ciklih
 - ▶ modeliramo prenos (transfer) in transformacijo podatkov med pomnilnimi elementi (registri): **RTL**
 - ▶ danes se večino digitalnih vezij načrtuje na nivoju RTL
- ▶ Kako pretvoriti algoritem v RTL ?

Primer algoritma – izračun točk v krogu

- ▶ Ali točka (x,y) leži znotraj krožnice (x_c, y_c, R^2) ?

$$(x-x_c)^2 + (y-y_c)^2 < R^2$$



```
dx <= signed(x) - xc;  
dy <= signed(y) - yc;
```

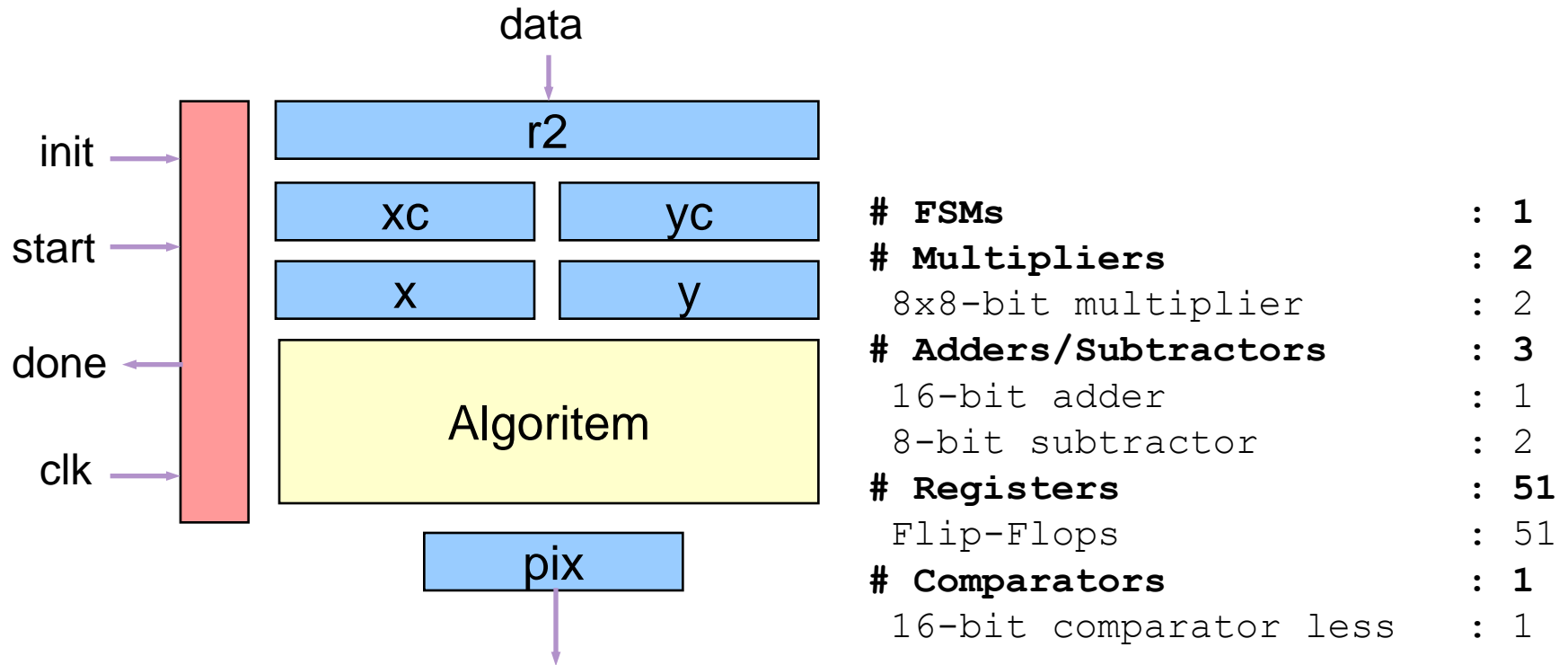
```
p1 <= dx * dx;  
p2 <= dy * dy;
```

```
r <= p1 + p2;
```

```
if rising_edge(clk) then  
  if r < r2 then  
    pix <= '1';  
  else  
    pix <= '0';  
  end if;
```

Vmesnik vezja

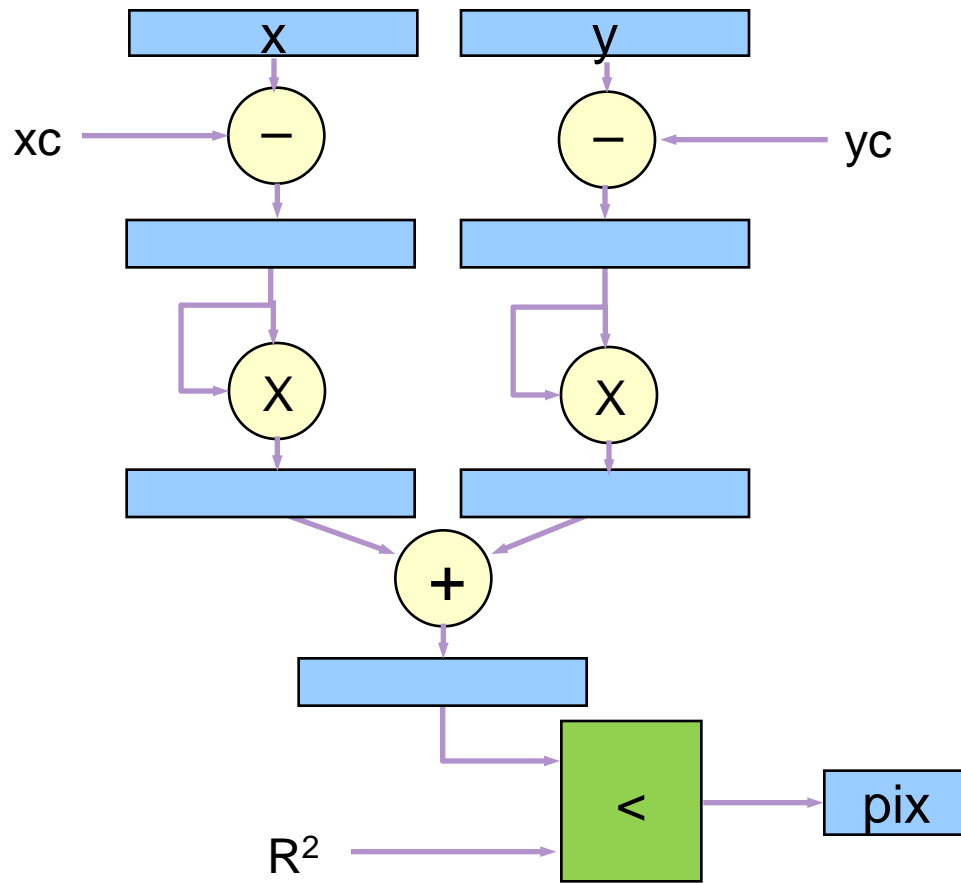
- ▶ Preko vmesnika nastavljamo parametre kroga in prenašamo podatke
- ▶ Potrebujemo registre in krmilne signale



- ▶ Zasedenost vezja XC3S50A: 7% (52), 3% FF, 2/3 MULT
- ▶ Hitrost: max. frekvenca ure 78 MHz, 1 cikel računanja

Optimizacija vezja – računski cevovod

- ▶ Rezultat vsake operacije shranimo v registru
- ▶ manjše zakasnitve v posameznih korakih



```
if rising_edge(clk) then
  dx <= signed(x) - xc;
  dy <= signed(y) - yc;

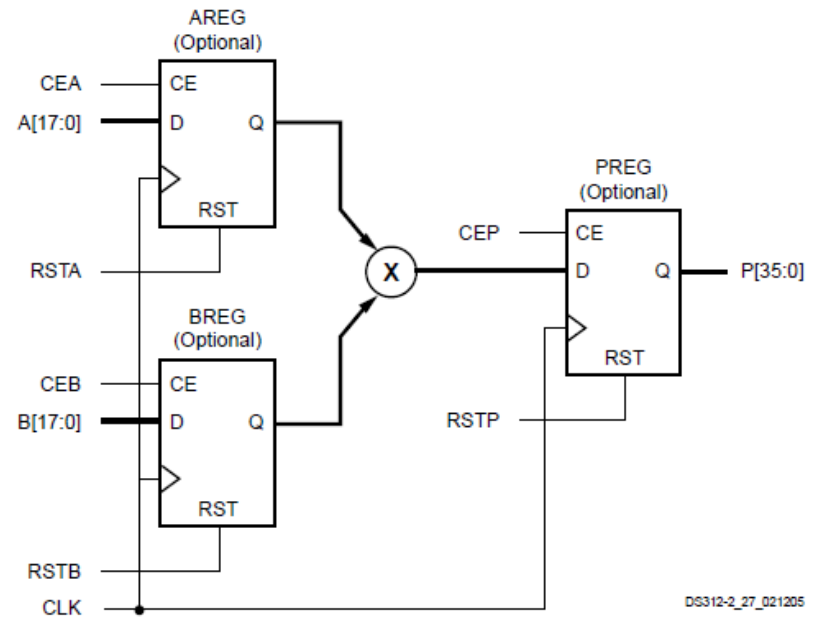
  p1 <= dx * dx;
  p2 <= dy * dy;

  r <= p1 + p2;
```

- ▶ cevovod omogoča, da pride vsak cikel nov podatek
- ▶ začetna zakasnitev (latenca) rezultata je 4 cikle

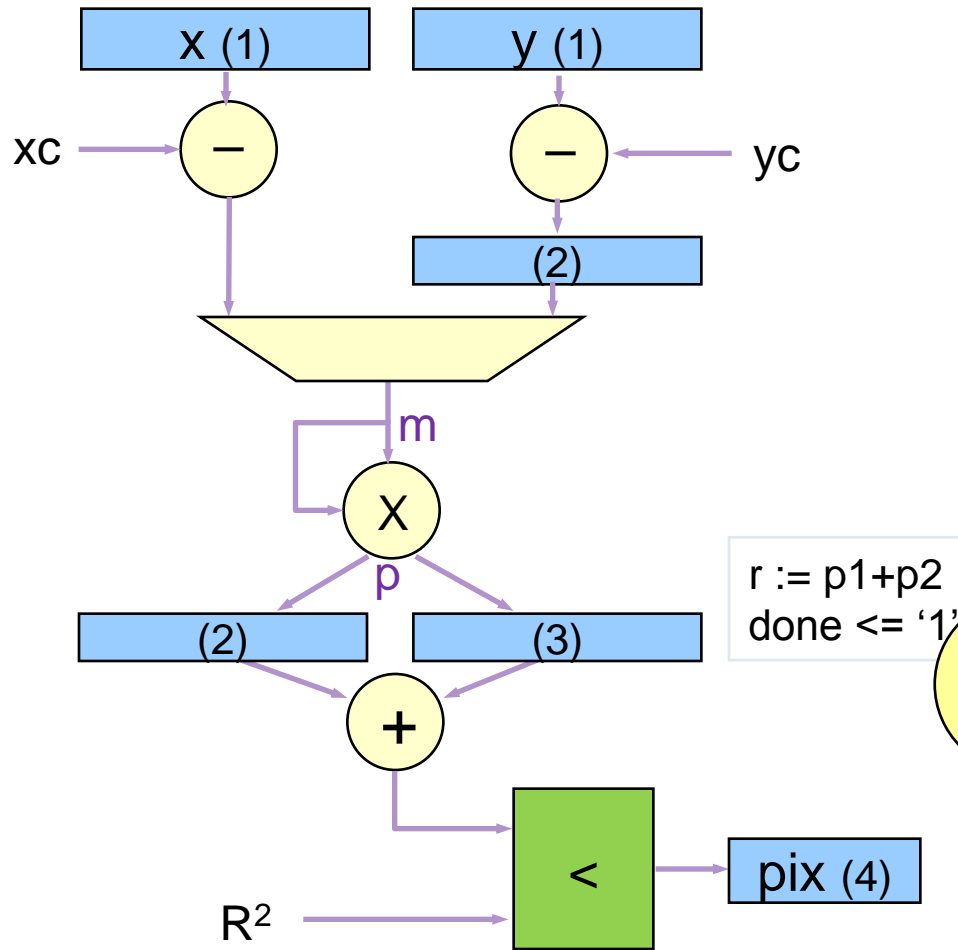
Rezultati sinteze in tehnološke preslikave

# FSMs	:	1
# Multipliers	:	2
8x8-bit multiplier	:	2
# Adders/Subtractors	:	3
16-bit adder	:	1
8-bit subtractor	:	2
# Registers	:	67
Flip-Flops	:	67
# Comparators	:	1
16-bit comparator less	:	1

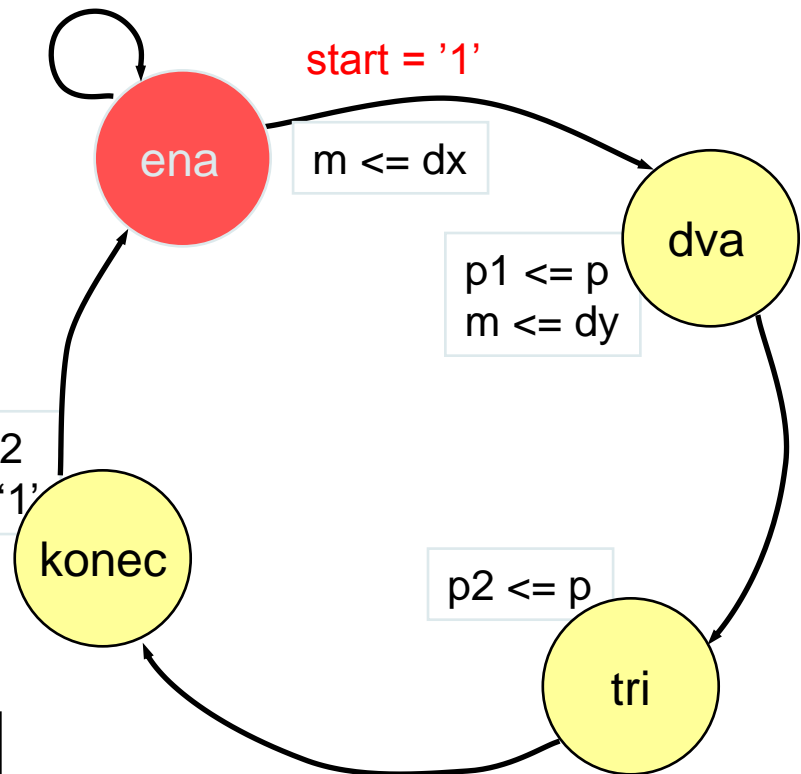


- ▶ Namesto 64 dodatnih FF (dx, dy, p1, p2, r) jih je le 16 !
 - ▶ produkt in registri so v posebnem bloku MULT znotraj FPGA
- ▶ Zasedenost vezja XC3S50A: 7% (53), 4% FF, 2/3 MULT
 - ▶ v celicah vezja FPGA je dovolj FF, zato se zasedenost ne poveča !
- ▶ Hitrost: max. frekvenca ure 200 MHz, 1(4) cikli računanja
- ▶ Rezultat brez uporabe blokov MULT: 20%, 8% FF, 102 MHz

Optimizacija površine – sekvenčno računanje



- ▶ 2x uporabimo en MULT
- ▶ krmilni avtomat



Opis vezja

► Kompakten opis sekvenčnega vezja

```
sekv: process (clk)
  variable dx, dy: signed(7 downto 0);
  variable r: signed(15 downto 0);
begin
  if rising_edge(clk) then
    if stanje=ena and start='1' then
      dx := signed(x) - xc;
      dy := signed(y) - yc;
      m <= dx;  -- vhodi za prvi produkt
      stanje <= dva;
    end if;

    if stanje=dva then
      p1 <= p;  -- shrani produkt in pripravi
      m <= dy;  -- vhode za drugi produkt
      stanje <= tri;
    end if;
```

```
    if stanje=tri then
      p2 <= p;  -- shrani drugi produkt
      stanje <= konec;
    end if;

    if stanje=konec then
      r := p1 + p2;  -- sestaj in primerjaj
      if r < r2 then  pix <= '1';
      else pix <= '0'; end if;
      done <= '1';  -- signal za zaključek
      stanje <= ena;
    end if;
```

► in kombinacijski množilnik:

```
p <= m * m;
```


Rezultati sinteze in tehnološke preslikave

- ▶ Zasedenost vezja XC3S50A: 9% (67), 5% FF, 1/3 MULT
- ▶ Hitrost: max. frekvenca ure 177 MHz, 4(4) cikli računanja

# FSMs	: 2
# Multipliers	: 1
8x8-bit multiplier	: 1
# Adders/Subtractors	: 3
16-bit adder	: 1
8-bit subtractor	: 2
# Registers	: 73
Flip-Flops	: 73
# Comparators	: 1
16-bit comparator less	: 1

- ▶ Rezultat brez uporabe blokov MULT: 15%, 6% FF, 98 MHz
- ▶ Primerjava s cevovodom brez MULT: 20%, 8% FF, 102 MHz

Povzetek

- ▶ Opiši načrtovanje vezij na nivoju registrov (RTL).
 - ▶ Kako pretvorimo algoritem (nalogo) v vezje RTL ?
 - ▶ Naštej nekaj možnih izvedb (optimizacij) v logičnem vezju.