Laboratorij za načrtovanje integriranih vezij

Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*

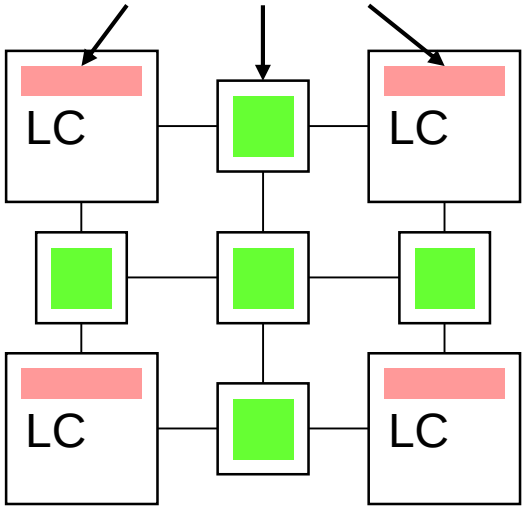
Digitalni Elektronski Sistemi

Programirljiva vezja in mikro-programiranje

programirljiv sekvenčni stroj, mikrosekvenčnik

Programirljiva vezja / mikroprocesorji

- programirljivo vezje
  - programski biti določajo strukturo (konfiguracijo)

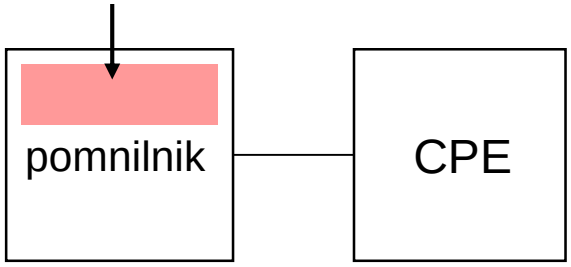


delovanje logičnih celic (LC)

povezave znotraj LC

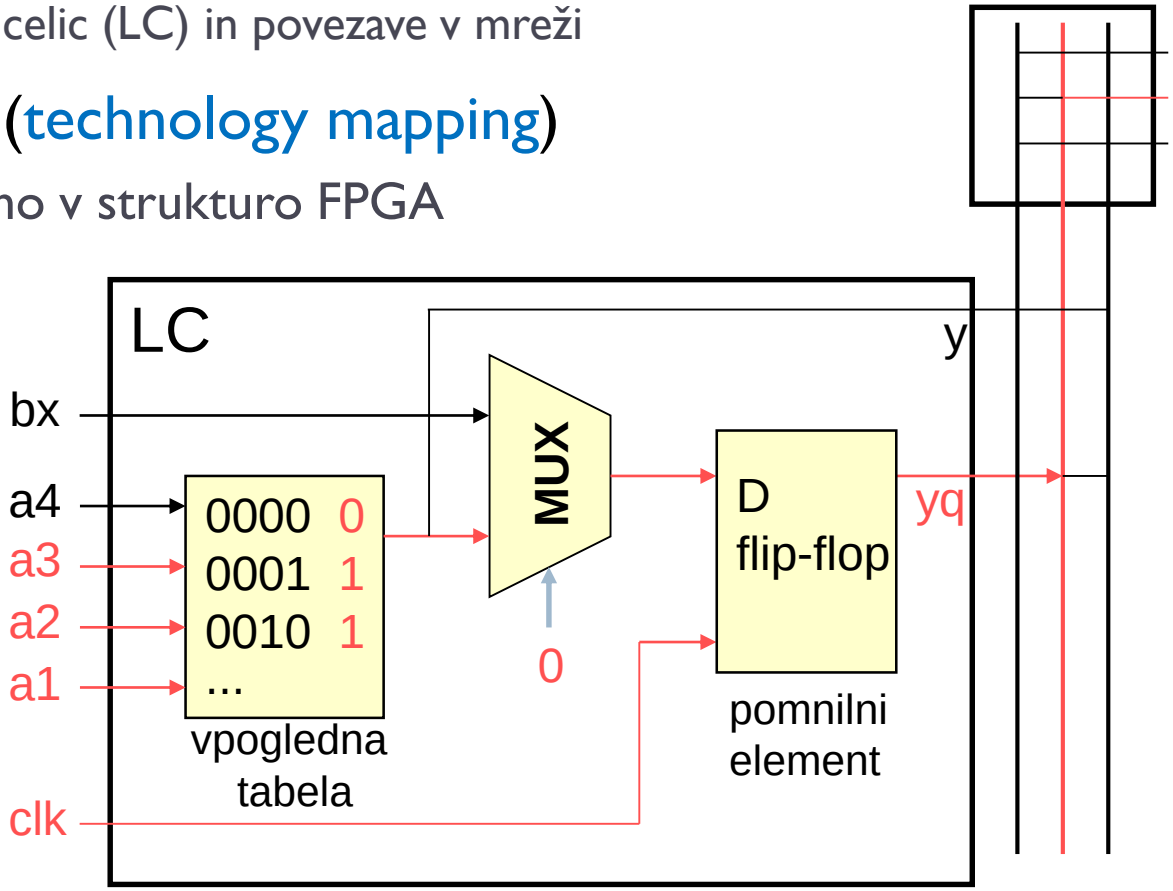
povezave v povezovalni mreži

- mikroprocesor
  - program določa ukaze, ki jih CPE izvaja

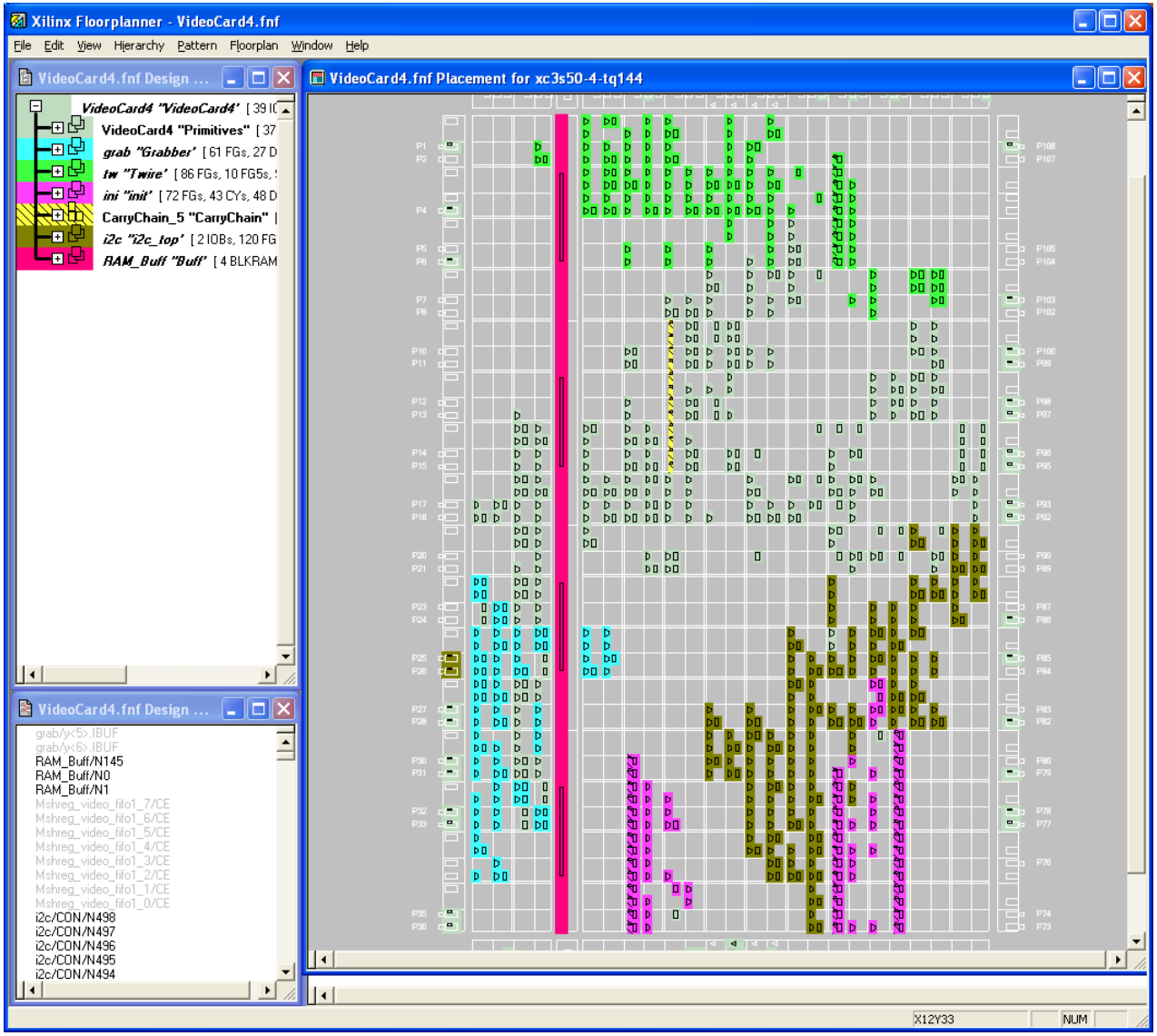


Podrobnejša zgradba FPGA

- pri programiranju (configuration) vpišemo vrednosti v konfiguracijski RAM
  - določa delovanje logičnih celic (LC) in povezave v mreži
- tehnološka preslikava (technology mapping)
  - digitalno vezje preslikamo v strukturo FPGA
- logična celica
  - LUT, MUX, DFF
  - prenosna logika
  - dodatne funkcije



Tehnološko preslikano vezje



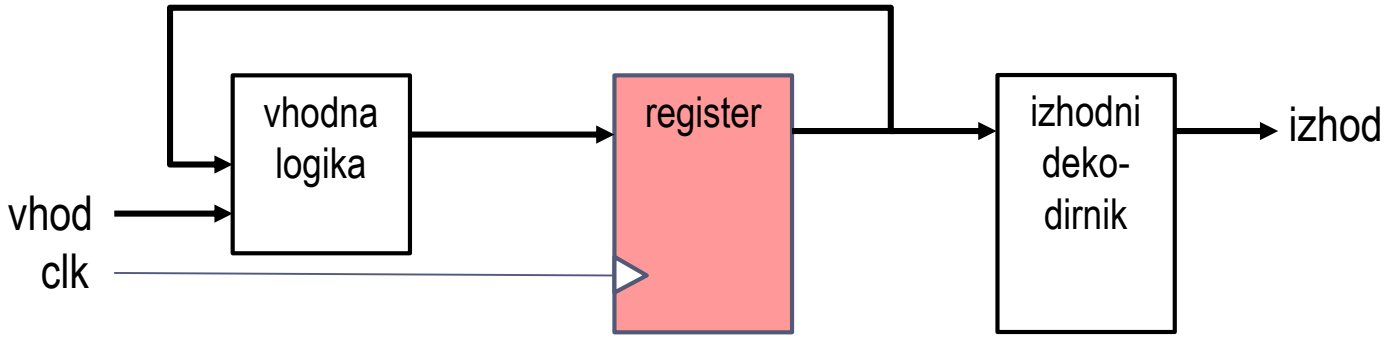
Načrt FPGA (floorplan)

- rezultat avtomatične preslikave
- načrtovanje vezja poteka na višjem nivoju

xc3s50 tq144
500 tabel (LUT)
280 D flip-flopov
8 kB DPRAM
~ 270k vrat ASIC

Sekvenčni stroj v programirljivem vezju

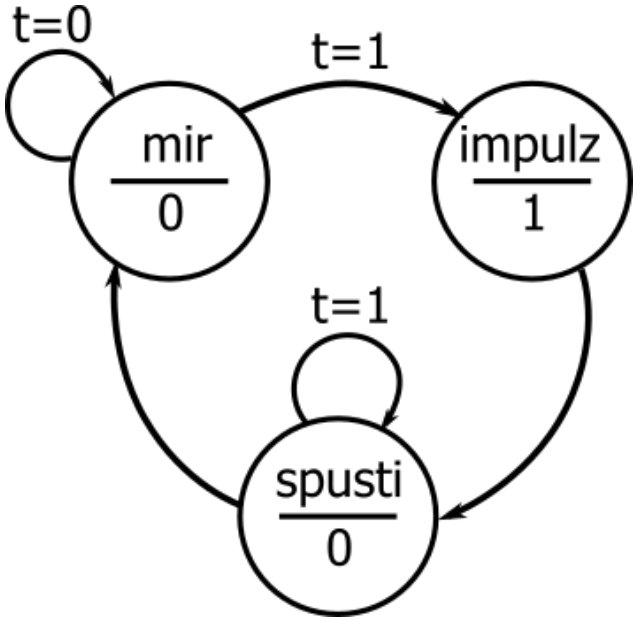
- Sekvenčni stroj vsebuje n-bitni register za 2^n stanj

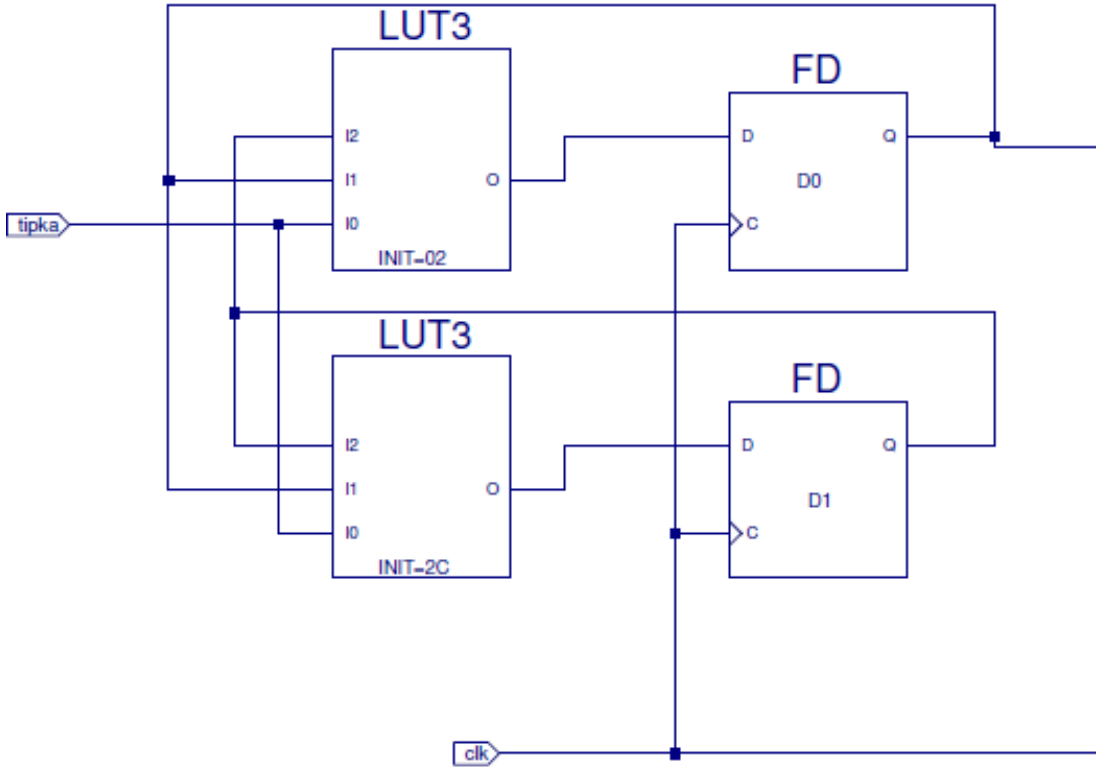


- Kombinacijska logika v povratni vezavi določa prehode stanj
  - logika je v vezju FPGA narejena s tabelami (LUT)
- Če naložimo novo vsebino tabel, spremenimo delovanje vezja !
  - v vezju FPGA imamo "programirljiv sekvenčni stroj", kjer za spremembe prehodov stanj ni potrebno razviti novo vezje

Primer: sekvenčni stroj za detekcijo tipke

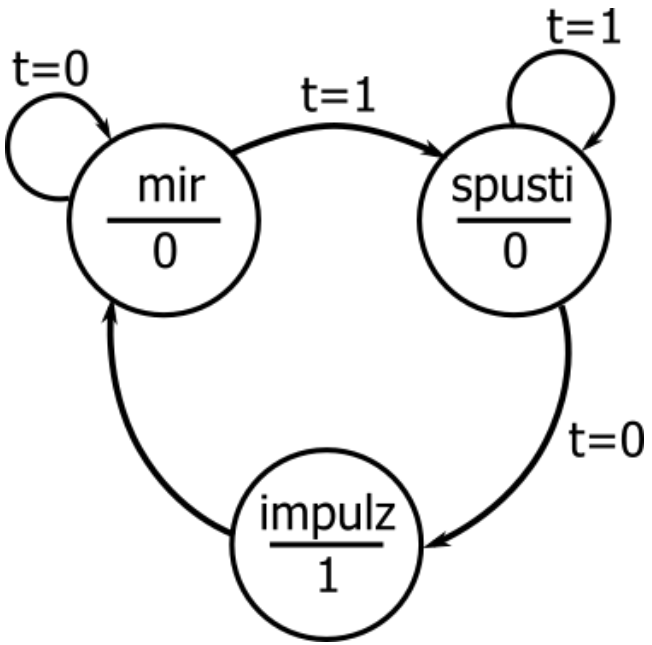
- Vsebina vpoglednih tabel (LUT) določa prehajanje stanj
  - ob pritisku tipke naredi impulz dolžine ene periode ure, nato počaka da tipko spustimo

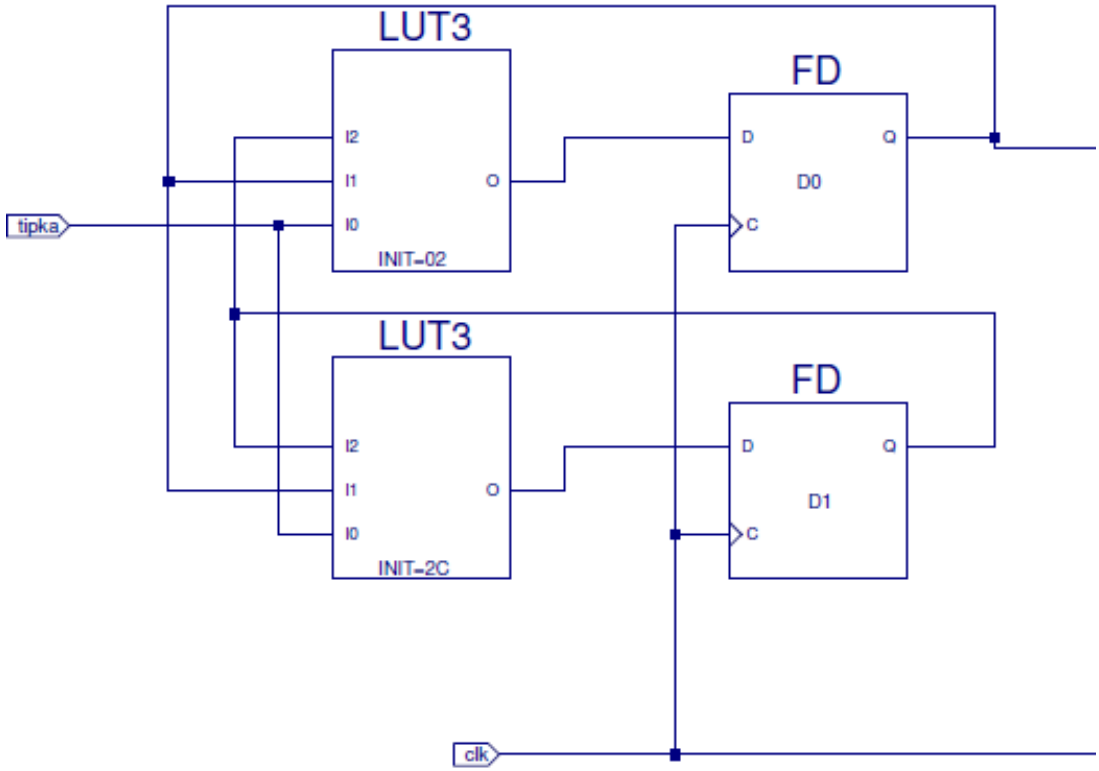




Primer: sekvenčni stroj za detekcijo tipke (2)

- Ob spremembi pogojev za prehod, se ne spremeni zgradba vezja, ampak le vsebina LUT
  - zazna pritisk tipke, počaka, da tipko spustimo in naredi impulz

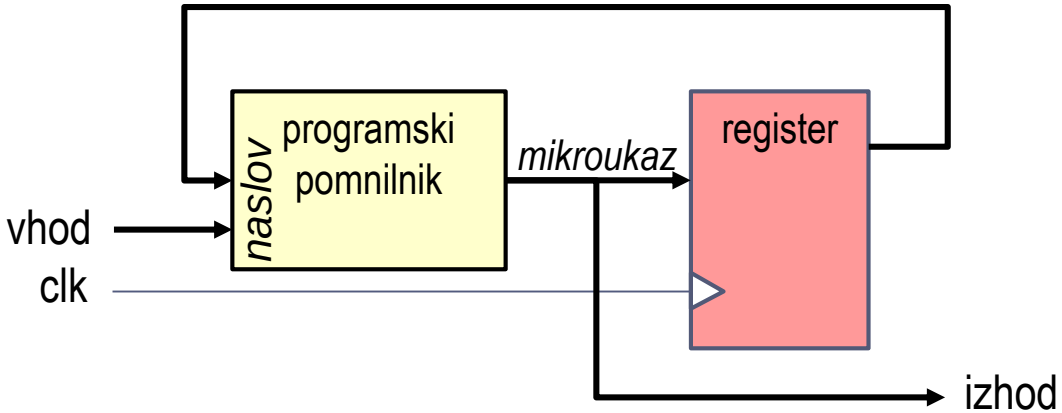




Mikro-programirano sekvenčno vezje

- Sekvenčni stroj narejen s tabelami je mikroprogramiran - mikrosekvenčnik
  - *mikroprogramirano vezje* ima krmilne vrednosti shranjene v programskem pomnilniku

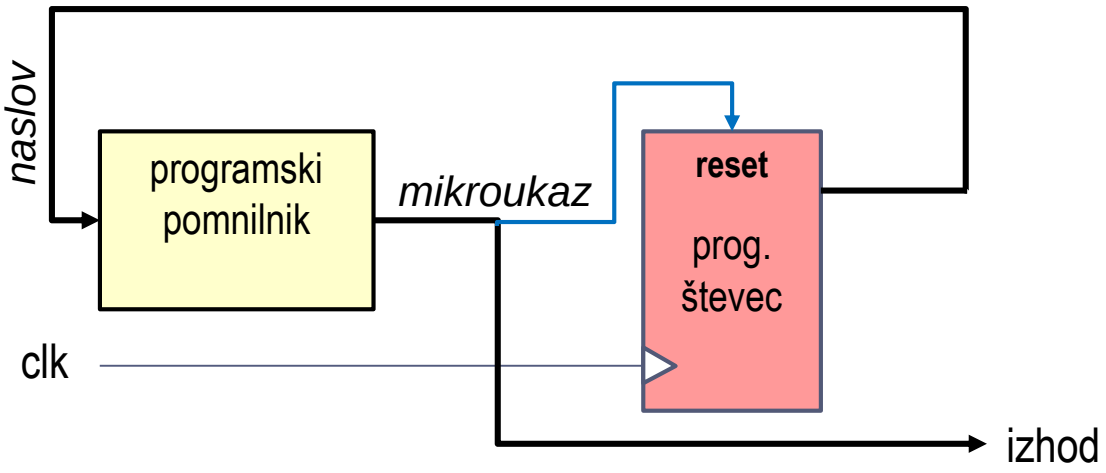
- *mikroukazi* so besede v pomnilniku
- *mikroprogram* je zaporedje *mikroukazov*
- *v programski pomnilnik* vrste RAM lahko vpisujemo *mikroukaze*



Mikrosekvenčnik s programskim števcem

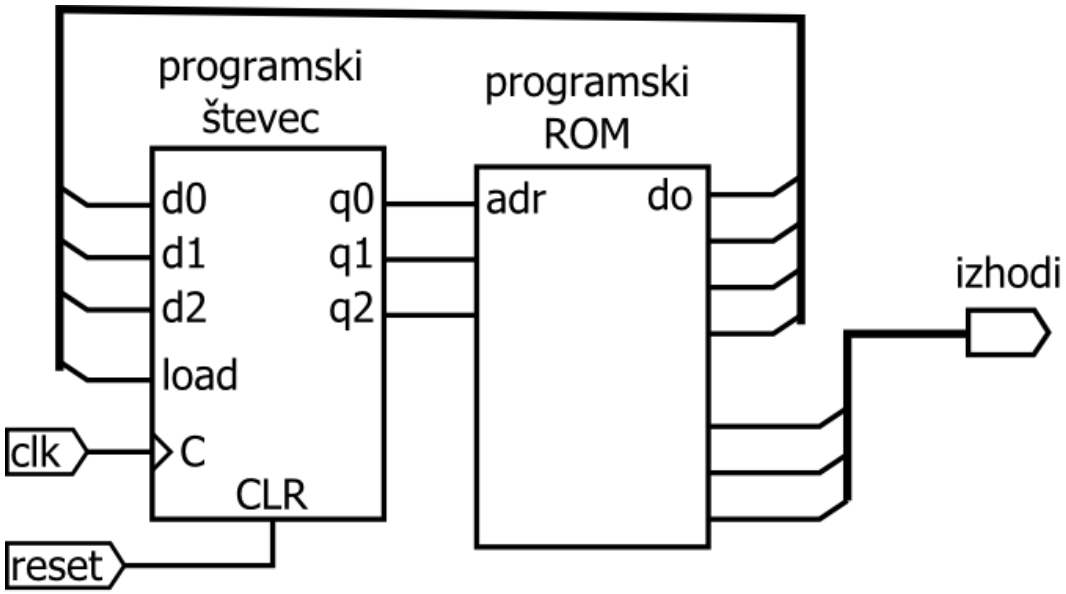
- Register nadomestimo s števcem, ki avtomatsko povečuje naslov
  - izvedba diagramov z enim samim zaporedjem (sekvenco) stanj
  - zaporedje se izvaja ob uri (ni krmilnih vhodov)
- Npr. izhod vezja se spreminja v zaporedju: 100, 110, 001, 001, 010
  - zapredje določa števec
  - en bit mikroukaza je reset števca

naslov	mikroukaz
000	0 100
001	0 110
010	0 001
011	0 001
100	1 010
101	x xxx
111	x xxx



Mikrosekvenčnik s skočnimi ukazi

- Programski števec naloži nov naslov, kar omogoča izvedbo skokov
  - algoritmi, ki nimajo vseh mikroukazov v zaporedju
  - mikroukaz vsebuje naslov, signal za nalaganje (load) in izhod
- Primer: semafor z zaporedjem: 010, 000, 010, 000, 100, 110, 001, 010, 100, 110, 001, 010…

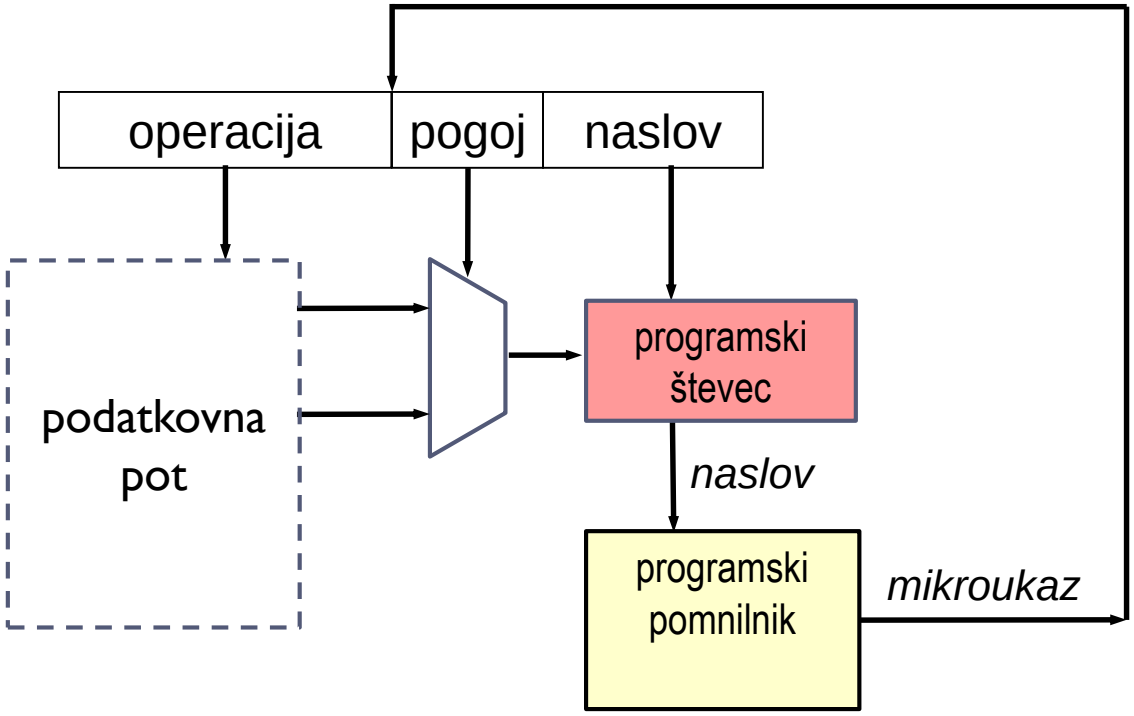


adr	d3:d0	load	izhod
000	0 0 0	0	0 1 0
001	0 0 0	0	0 0 0
010	0 0 0	0	0 1 0
011	0 0 0	0	0 0 0
100	0 0 0	0	1 0 0
101	0 0 0	0	1 1 0
110	0 0 0	0	0 0 1
111	1 0 0	1	0 1 0

skok
rdeča
rumena
zelena

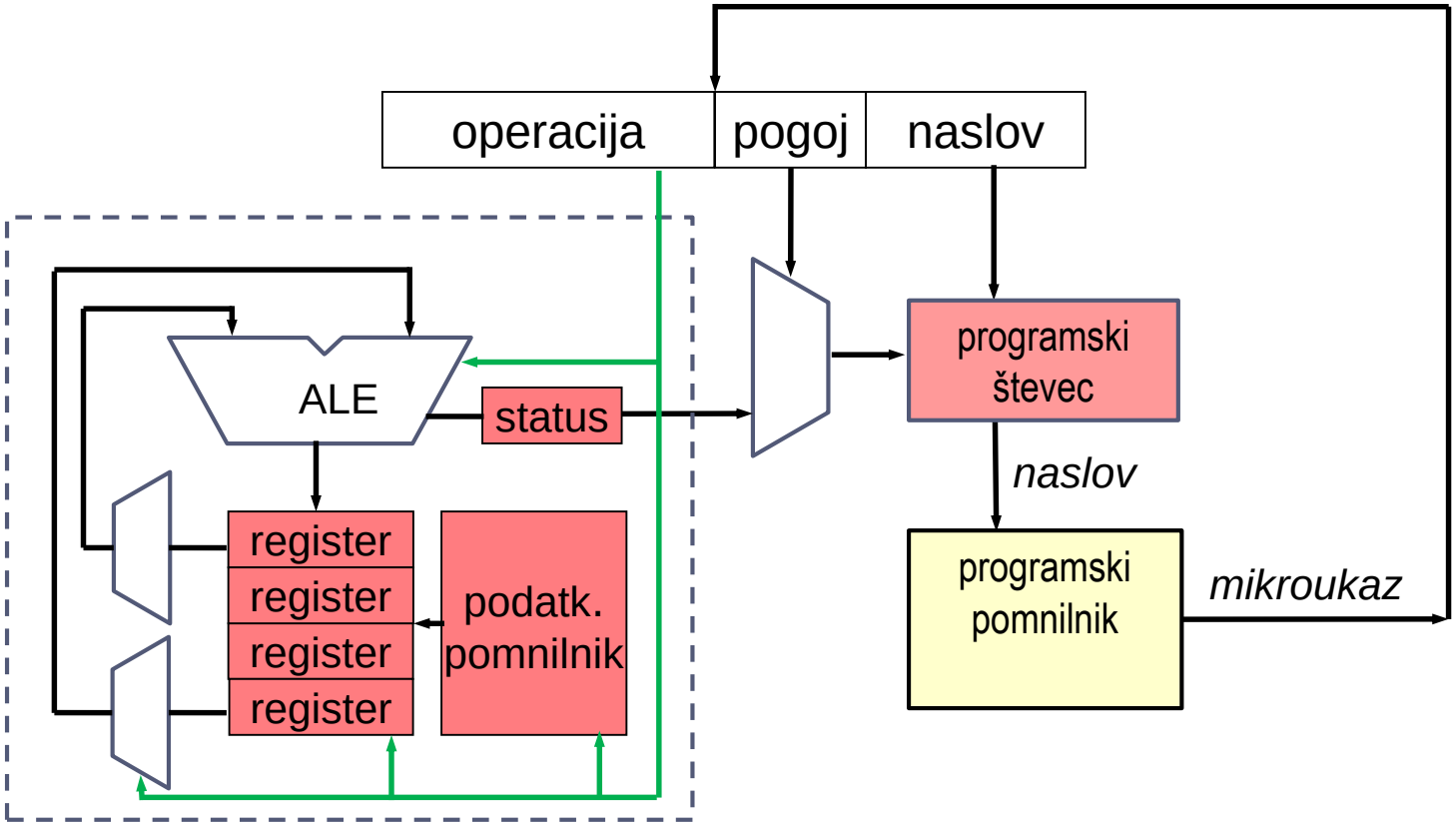
Mikrosekvenčnik in podatkovna pot

- Operacije krmilijo podatkovno pot
  - mikroukaz razdelimo na operacijo, pogoje (za skok) in naslov



Mikroprogramirana krmilna (procesna) enota

- Podatkovna pot izvaja registrske (mikro)operacije
  - vsebuje registre, aritmetično-logično enoto(ALE), podatkovni pomnilnik



Primer: sekvenčno vezje krmilne enote

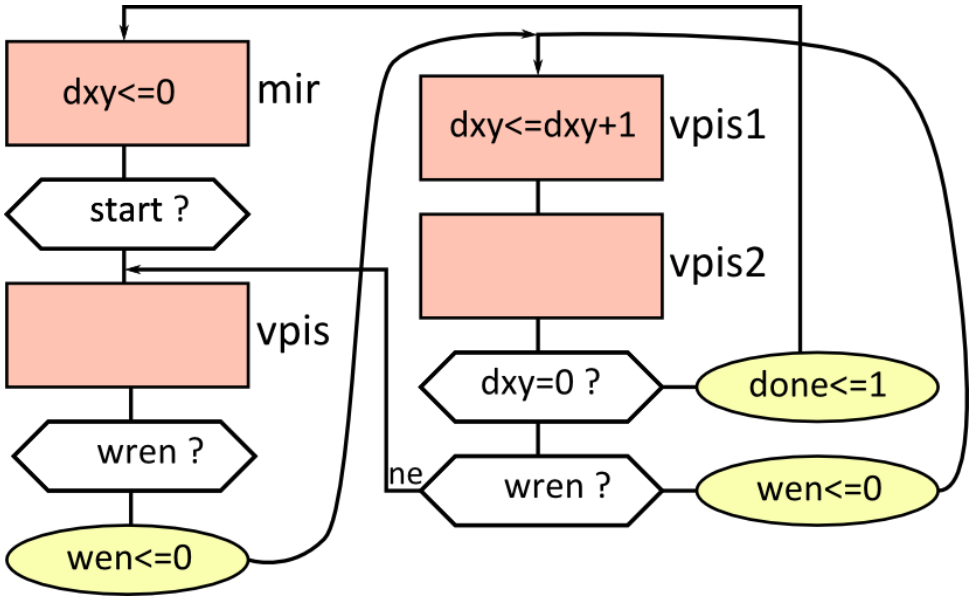
- Algoritem opišem z ASM = Algoritmic State Machine
  - določimo prehajanje stanj in podatkovne operacije
- Npr. krmilna enota za risanje kvadrata 8x8 (Pixel)

  - Simboli ASM:

stanje

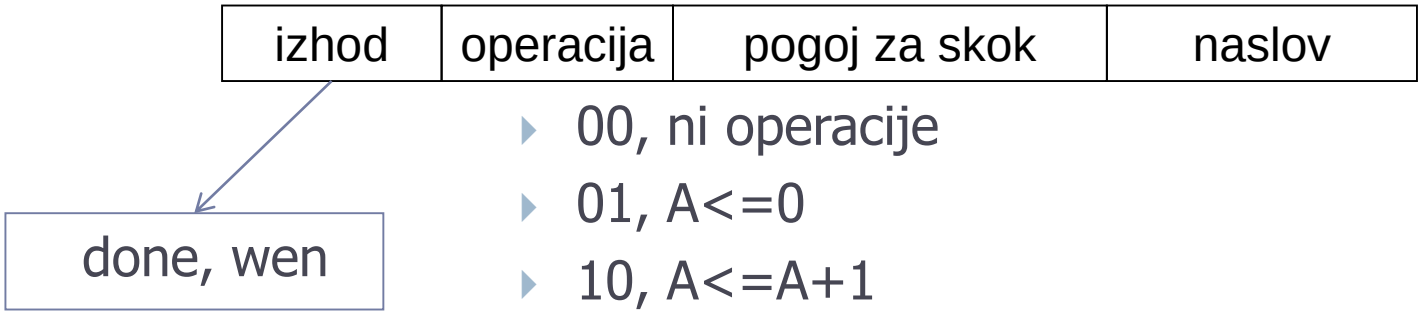
pogoj za prehod

akcija ob prehodu



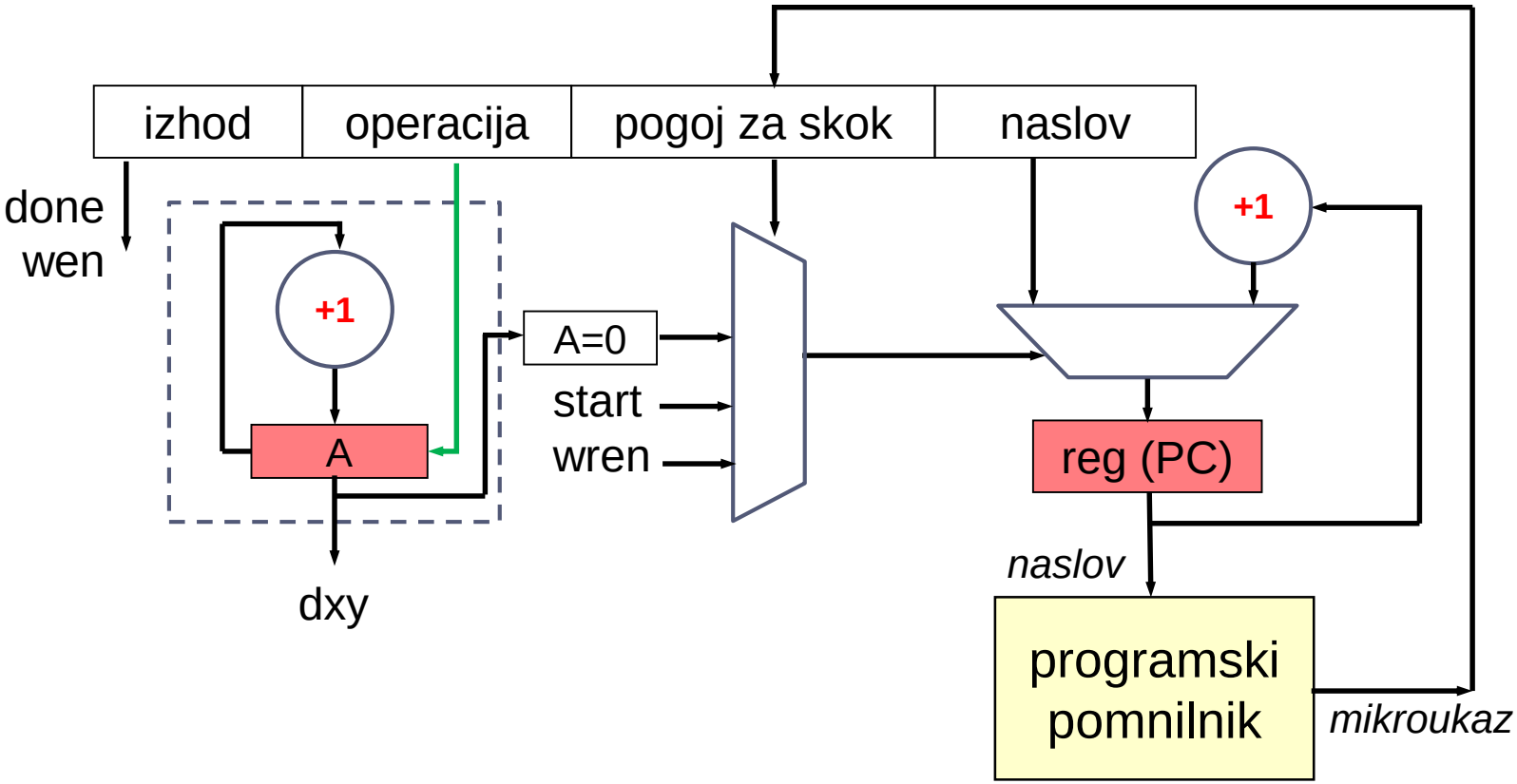
Načrtovanje mikroprogramirane enote

- Stanje določa programski števec
- Števec dxy je v registru A na podatkovni poti
- Določimo obliko mikroukazov:



Shema mikroprogramirane enote

- Podatkovna pot vsebuje register A (dxy)
  - operaciji: A<=0, A<=A+1
- Izhod: done, wen



Kodiranje mikroprogramirane enote

- deklaracija pomnilnika

```
type mem is array(0 to 7) of std_logic_vector(10 downto 0);
signal m: mem := (
    B"01_01_0001_000",  -- (0) a<=0, jmp 0 if start=0
    B"01_00_0010_001",  -- (1) jmp 1 if wren=0
    B"00_10_0000_000",  -- (2) a<=a+1, wen<=0
    B"01_00_0100_110",  -- (3) jmp 6 if a=0 (zero)
    B"01_00_0010_001",  -- (4) jmp 1 if wren=0
    B"01_00_1000_010",  -- (5) jmp 2
    B"10_00_1000_000",  -- (6) jmp 0, done<=1
```

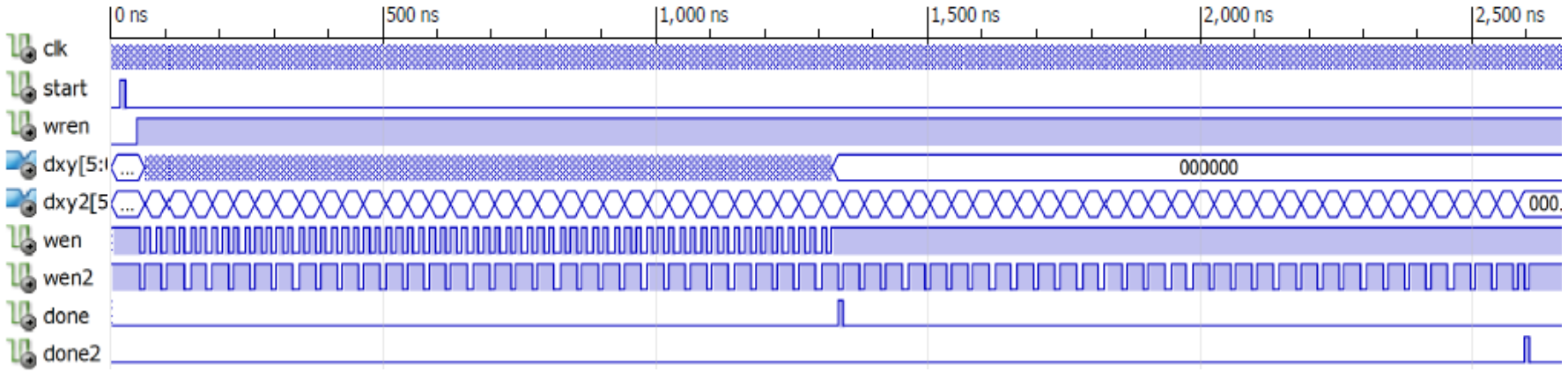
- opis vezja

```
if rising_edge(clk) then
    if inst(8 downto 7)="01" then a <= "000000";
    elsif inst(8 downto 7)="10" then a <= a + 1;
    end if;

    if inst(6 downto 3)="1000" or
      (inst(6 downto 3)="0100" and zero='1') or … then
          pc <= unsigned(inst(2 downto 0));
    else
          pc <= pc + 1;
```

Primerjava obeh izvedb

- Primerjava časovnih potekov na simulaciji
  - ASM cca. 2x hitrejši od procesne enote



- Kodiranje v jeziku VHDL
  - cca. 90 vrstic kode za opis obeh vezij
  - 2 notranja signala (ASM) in 6 signalov za procesor
- Velikost vezja
  - ASM zasede 25 LUT in 12 DFF
  - Procesor zasede 19 LUT in 9 FF

Povzetek

- Kaj se zgodi ko naložimo program v mikroprocesor in ko naložimo program v programirljivo vezje (FPGA) ?
  - kaj predstavljajo programski biti ?
- Opiši izvedbo sekvenčnega stroja v programirljivem vezju.
- Opiši mikroprogramirano sekvenčno vezje (mikrosekvenčnik).
  - Kateri gradniki sestavljajo mikrosekvenčnik.
  - Opiši kakšen primer.