

Procesorski sistemi v telekomunikacijah
Od logičnih vezij do mikroprocesorja in nazaj



(c) Arpad Bűrmen, 2010-2012

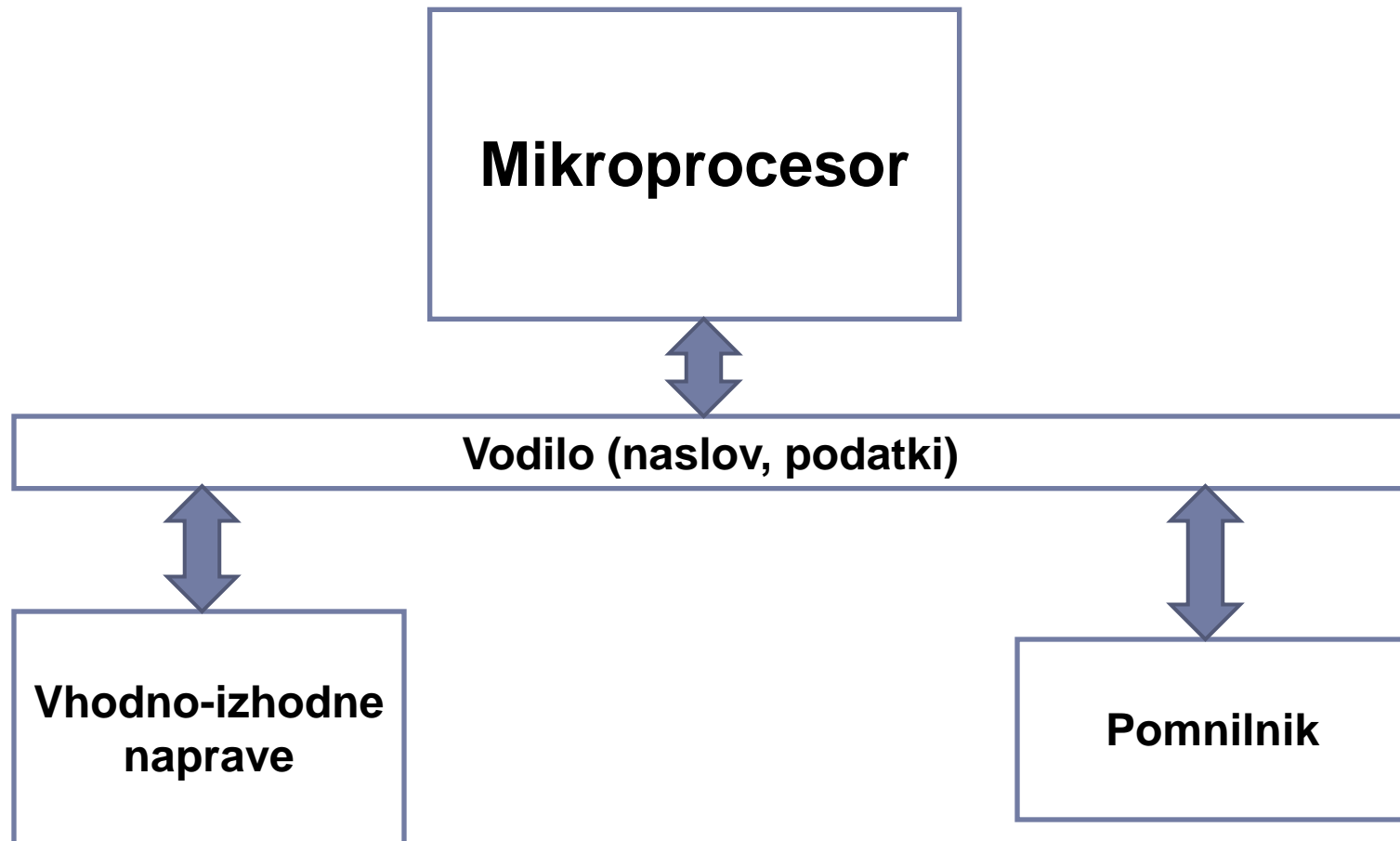
Zapis števil, bit in byte ...

- ▶ Dvojiški zapis (se uporablja v računalniku)
- ▶ Šestnajstiški zapis (uporabljamo ljudje, bolj razumljiv)
- ▶ Enota za količino spomina
1 bit, 8 bitov = 1 byte
1 byte = eno število od -128..127 ali 0..255
- ▶ b... bit, B...byte
- ▶ 1 kilobyte = 1kB = 1000 B
- ▶ Včasih 1kB = 2^{10} B = 1024 B
... ni OK po SI standardu
- ▶ Proizvajalci diskov ... 1kB = 1000 B
Windows Explorer ... 1kB = 1024 B

... zapis števil, bit in byte

- ▶ Predlog (NIST, IEEE): kilobinarni byte
 $1\text{KiB} = 1024\text{ B}$
- ▶ Kilobinarni bit
 $1\text{ kib} = 1024\text{ b}$
- ▶ Kilobit
 $1\text{ kb} = 1000\text{ b}$
... hitrost prenosa
npr. $100\text{Mbit/s} = 100\,000\,000\text{ b/s} = 12.5\text{ MB/s}$
- ▶ LSB – Least significant bit – skrajni desni bit
MSB – Most significant bit – skrajni levi bit
- ▶ LSW/MSW – Least/Most Significant Word

Arhitektura sistema v grobem



Pomnilnik

- ▶ Sestavljen iz 2^m n-bitnih celic
- ▶ Vsaka celica hrani n-bitno celo število
- ▶ Vsaka celica ima (m-biten) naslov
- ▶ Primer: $m=10$, $n=8$
1024x8 bitov ... 1024 bytov spomina
- ▶ Primer: $m=12$, $n=16$
4096x16 bitov = 8192 bytov spomina
- ▶ Fizična organizacija celic – pravokotna shema
npr. 4096x16 bitov = 64x64x16bitov
- ▶ Povezava do pomnilnika: naslovne in podatkovne
linije primer 4096x16 bitov
12 naslovnih linij (m), 16 podatkovnih linij (n)

Program, podatki in pomnilnik ...

- ▶ Program = zaporedje ukazov
- ▶ Vrste ukazov odvisne od vrste mikroprocesorja
- ▶ Program in podatki hranjeni v pomnilniku
- ▶ Ukazi v splošnem različno dolgi (št. bytov)
- ▶ Programski števec (PC, Program Counter)
... naslov naslednjega ukaza
Po branju ukaza se PC poveča.
- ▶ Izvajanje programa
 1. preberi ukaz iz pomnilnika
 2. povečaj PC
 3. dekodiraj ukaz
 4. izvedi ukaz

... program, podatki in pomnilnik

- ▶ Registri = poseben pomnilnik znotraj mikroprocesorja
PC je eden od registrov
- ▶ Ukazi manipulirajo z vsebino registrov in z vsebino pomnilnika preko vodila.
- ▶ Zagon programa ob vklopu (reset) odvisno od tipa mikroprocesorja
 - a) PC = fiksni naslov (0 pri ARM)
 - b) PC = nek naslov, ki se prebere iz pomnilnika (\$FFFE-\$FFFF pri 68HC11)
- ▶ Vhodno-izhodne (IO, Input-Output) naprave.
Registri naprav v naslovnem prostoru mikroprocesorja.

Primer programa ... 68HC11

- ▶ Naslovni prostor ... $2^{16} = 65536$ celic ... po 8 bitov
Naslovi od \$0000 ... \$FFFF
- ▶ FLASH pomnilnik (za program) – 8KiB
\$E000-\$FFFF
- ▶ RAM pomnilnik (za podatke) – 8KiB
\$2000-\$3FFF
- ▶ Von Neumannovo ozko grlo
FLASH in RAM na istem vodilu =
vsaj 2 dostopa za ukaz, ki piše v pomnilnik

```
E000: 86 A3          LDAA    #$A3          #A=$A3
E002: 8B 12          ADDA    #$12          #A=A+$12
E004: B7 20 00       STAA    $2000         #mem($2000)=A
E007: 20 FE          BRA     $E007         #skoci nazaj na sebe
```


Primer programa ... ARM7

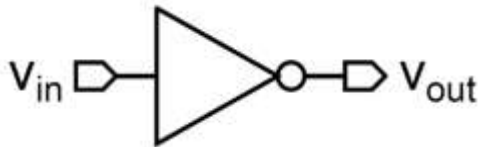
- ▶ Naslovni prostor $2^{32} = 4\,294\,967\,296$ celic po 8 bitov
Naslovi od \$00000000 ... \$FFFFFFFF
- ▶ Š-ARM
- ▶ FLASH pomnilnik (za program) – 256KiB
\$00000000-\$0003E7FF
- ▶ RAM pomnilnik (za podatke) – 16KiB
\$40000000-\$40003FFF

```
00000040: A3 00 A0 E3    MOV R0, #0xA3           R0=0xa3
00000044: 12 00 90 E2    ADDS R0, R0, #0x12      R0=R0+0x12
00000048: 01 11 A0 E3    MOV R1, #0x40000000     R1=0x40000000
0000004C: 00 00 81 E5    STR R0, [R1, #0x0000]   mem(R1+0x0000)=R0
00000050: 50 F0 A0 E3    MOV PC, #0x00000050     skoci nazaj na sebe
```

Logični nivoji

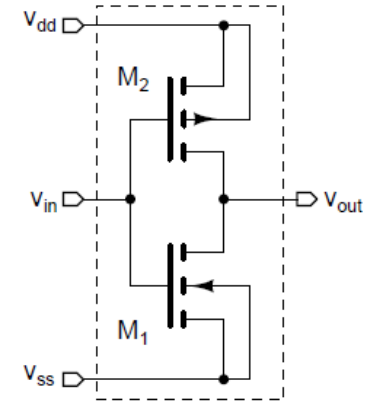
- ▶ Dve osnovni vrednosti – **0** in **1**
- ▶ Preslikava napetosti v 0/1 – logični nivoji
- ▶ Primer: TTL nivoji
 - 0V ... 0.8V = “0”
 - 2.7V ... 5V = “1”
 - 0.8V ... 2.7V ... neveljavno ... napaka v delovanju
- ▶ V splošnem v sodobnih CMOS vezjih
Napajanje Vdd proti referenčni sponki (GND)
Idealno:
 - Vdd = “1”
 - 0V = “0”
- ▶ V praksi – preberi podatkovni list vezja

Logična vrata - gradniki logičnih vezij

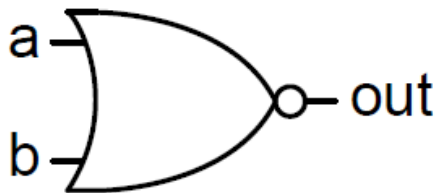


Inverter

in	out
0	1
1	0

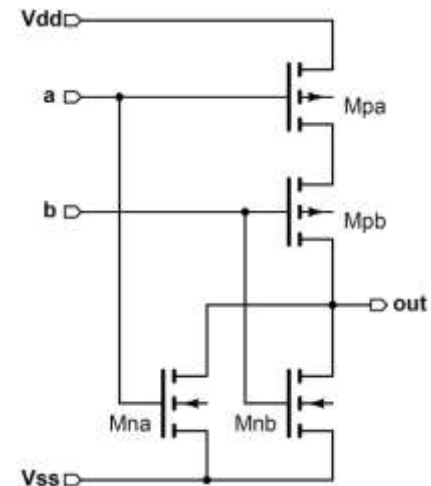


CMOS izvedba



Ne-Ali vrata (NOR)

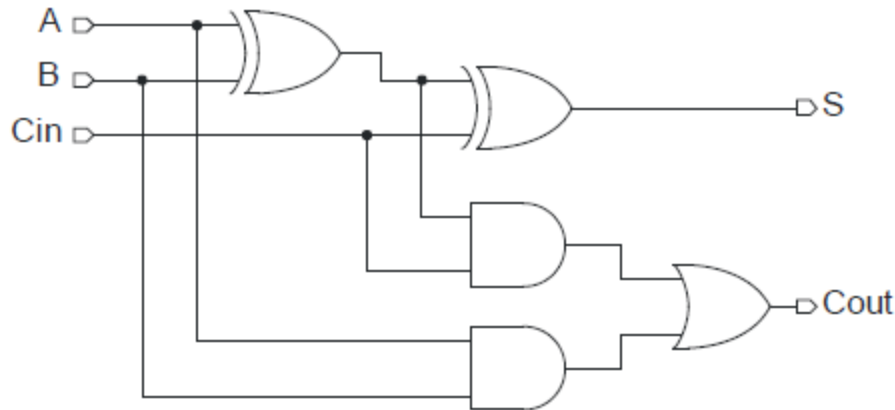
a	b	out
0	0	1
0	1	0
1	0	0
1	1	0



Kombinatorna logična vezja

- ▶ Sestavljena iz večih logičnih vrat
- ▶ Ni povratne vezave
- ▶ Nimajo stanja – odziv odvisen le od vrednosti vhodov
- ▶ Zakasnitev - vrata ne morejo hipoma spremeniti izhoda

Odvisna od kapacitivnosti v vezju.



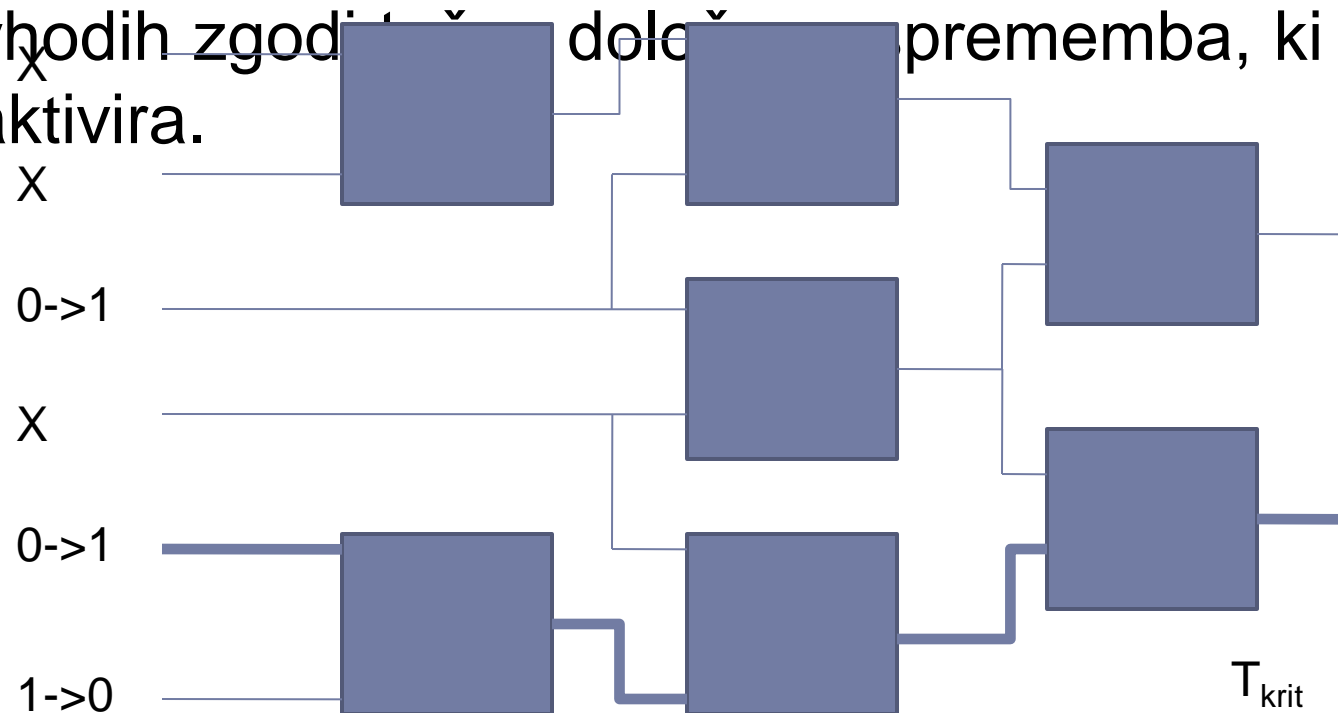
1-bitni seštevalnik

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Pravilnostna tabela

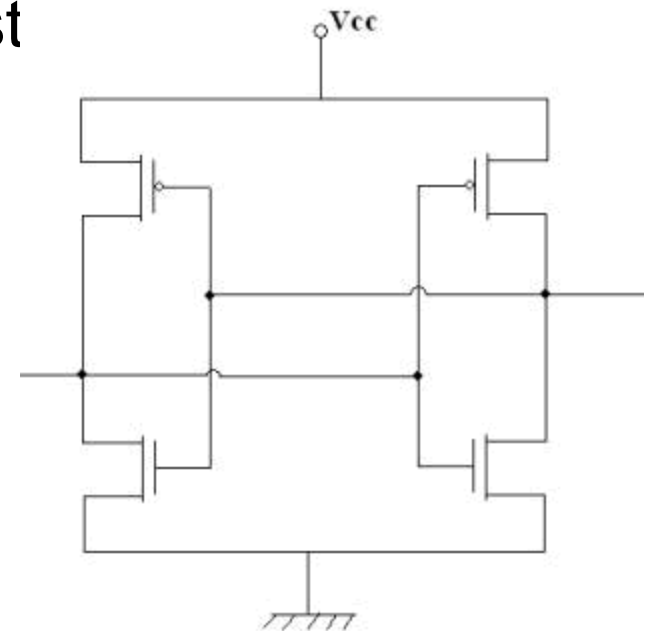
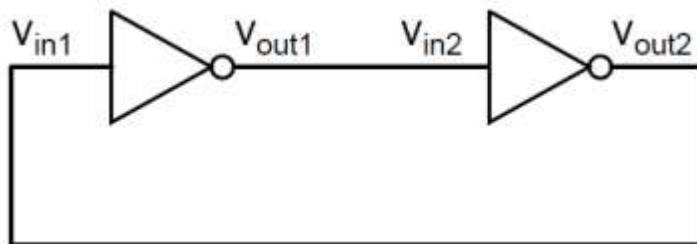
Kritična pot

- ▶ Po njej se širi sprememba vhodov proti izhodu za primer najpočasnejšega odziva kombinatornega vezja
- ▶ Spremembe izhodov potujejo po kritični pot, ko se na vseh vhodih zgodijo dve spremembi, ki pot aktivira.



Sekvenčna logična vezja

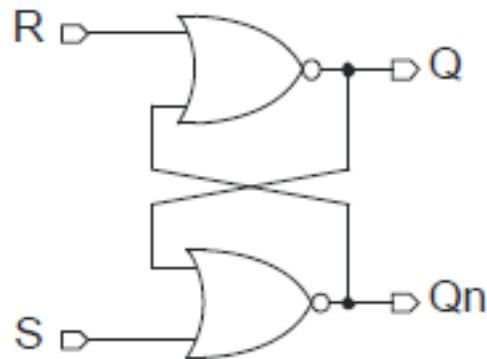
- ▶ Kombinatorno vezje + povratne vezave
- ▶ Izhodi odvisni od preteklosti in od vhodov vezja – **vezje ima spomin**
- ▶ Vezje na sliki ima dve stabilni stanji
 - a) $out1=1$ $out2=0$
 - b) $out1=0$ $out2=1$
- ▶ Sprememba stanja (ni hipna):
vsilimo 1 oz 0 na V_{in2}



Celica statičnega pomnilnika (1 bit)

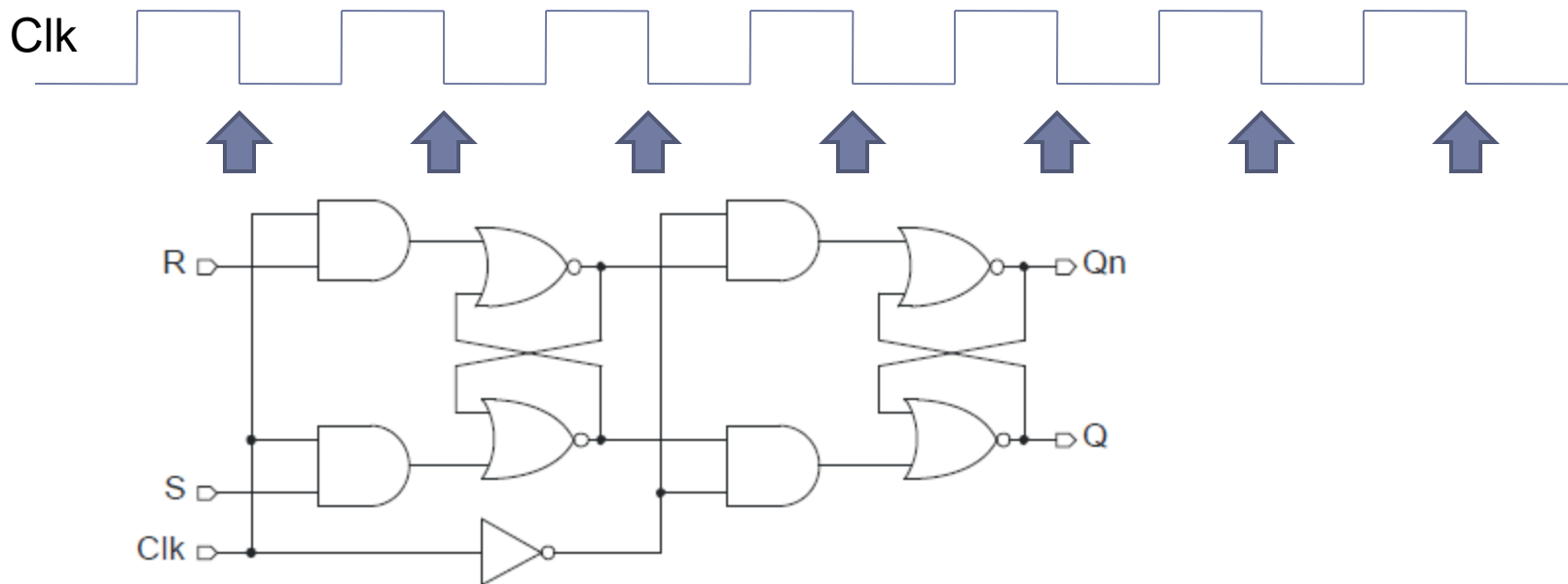
Primer - RS zatič

- ▶ Dve stabilni stanji
 - a) $Q=1$ $Qn=0$
 - b) $Q=0$ $Qn=1$
- ▶ Sprememba stanja:
 - $R=1, S=0$... Q postane 0, Qn postane 1
 - $R=0, S=1$... Q postane 1, Qn postane 0
- ▶ Vzdrževanje stanja ... $R=0, S=0$



Sinhrona in asinhrona logična vezja

- ▶ Asinhrona vezja – sprememba stanja se prične dogajati takoj ob spremembi vhodov
- ▶ Sinhrona vezja – sprememba stanja se zgodi le ob določenih trenutkih – določa jih signal ure (clock)



(Sinhron) master/slave RS flip-flop – sprememba stanja/izhodov ob prehodu Clk iz 1 v 0

Sekvenčno + kombinatorno vezje

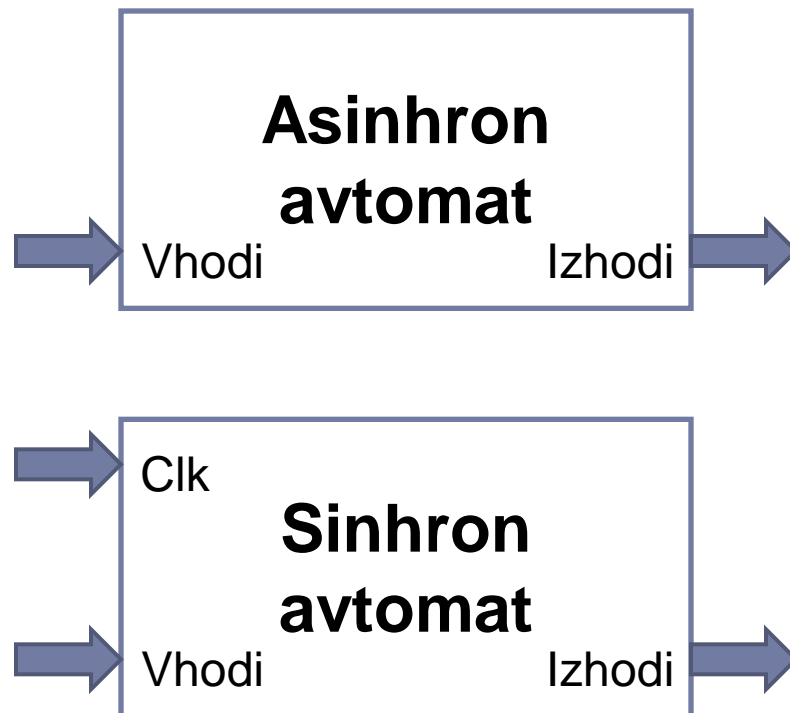
Avtomat

- ▶ Avtomat je sekvenčno vezje
- ▶ Avtomat ima dve ali več stanj
- ▶ Novo stanje = $f(\text{Trenutno stanje}, \text{Vhodi})$
- ▶ Izhodi = $g(\text{Trenutno stanje}, \text{Vhodi})$



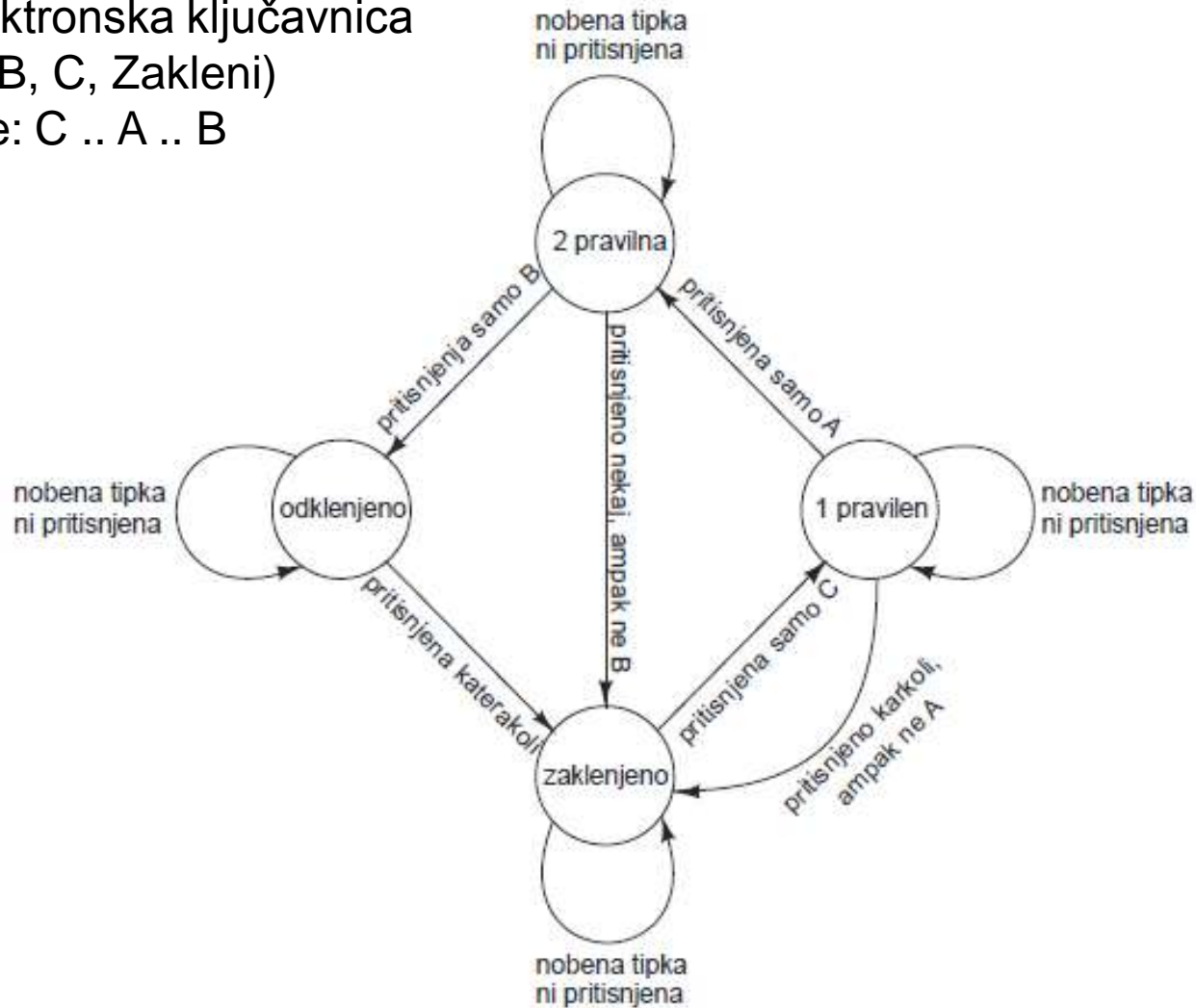
Sinhroni in asinhroni avtomati

- ▶ V splošnem so avtomati asinhroni
- ▶ Sekvenčni del = sinhron ... avtomat = sinhron



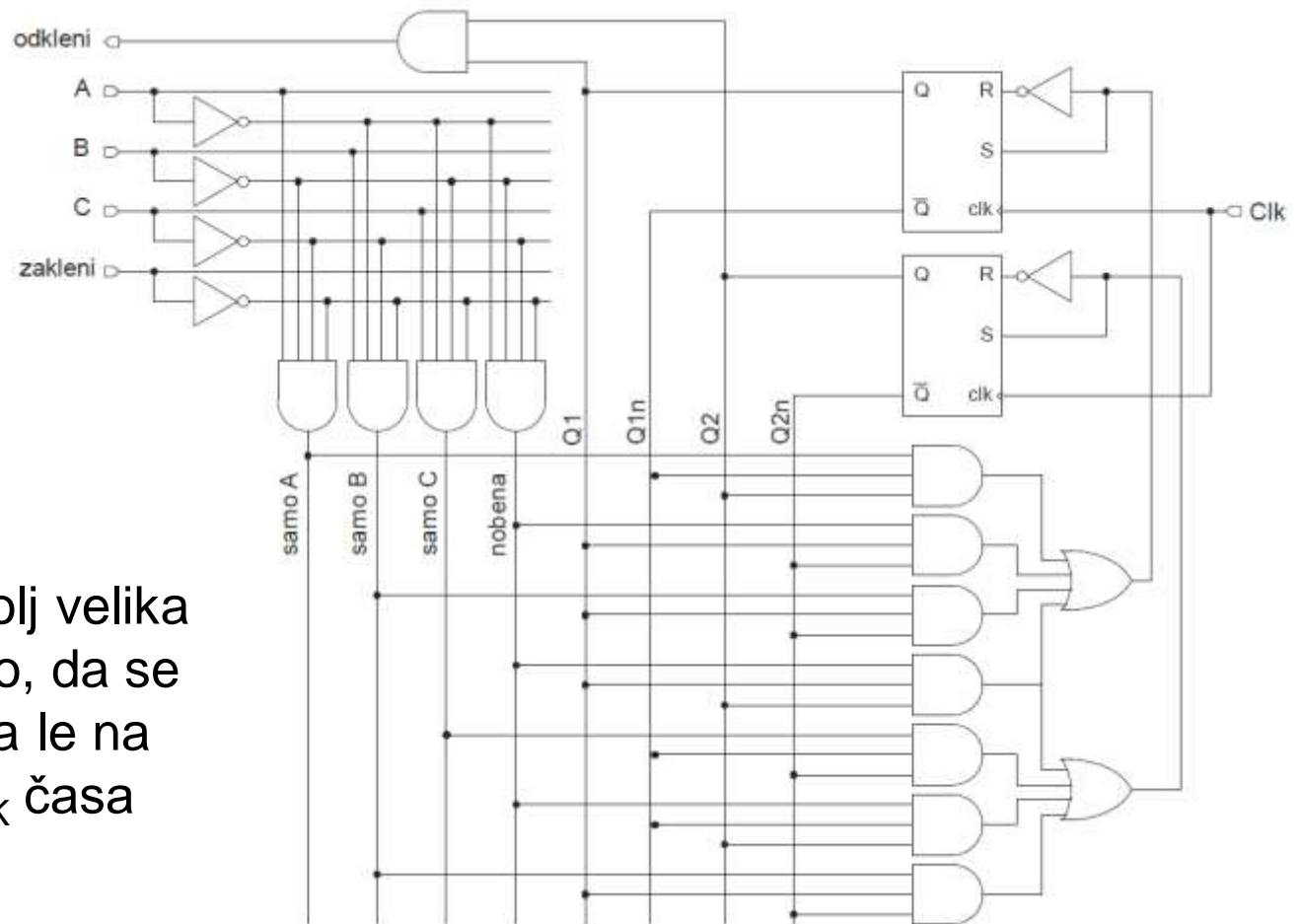
Načrtovanje sinhronega avtomata – Diagram prehajanja stanj

Primer: elektronska ključavnica
4 tipke (A, B, C, Zakleni)
Odklepanje: C .. A .. B



Izvedba sinhronnega avtomata ...

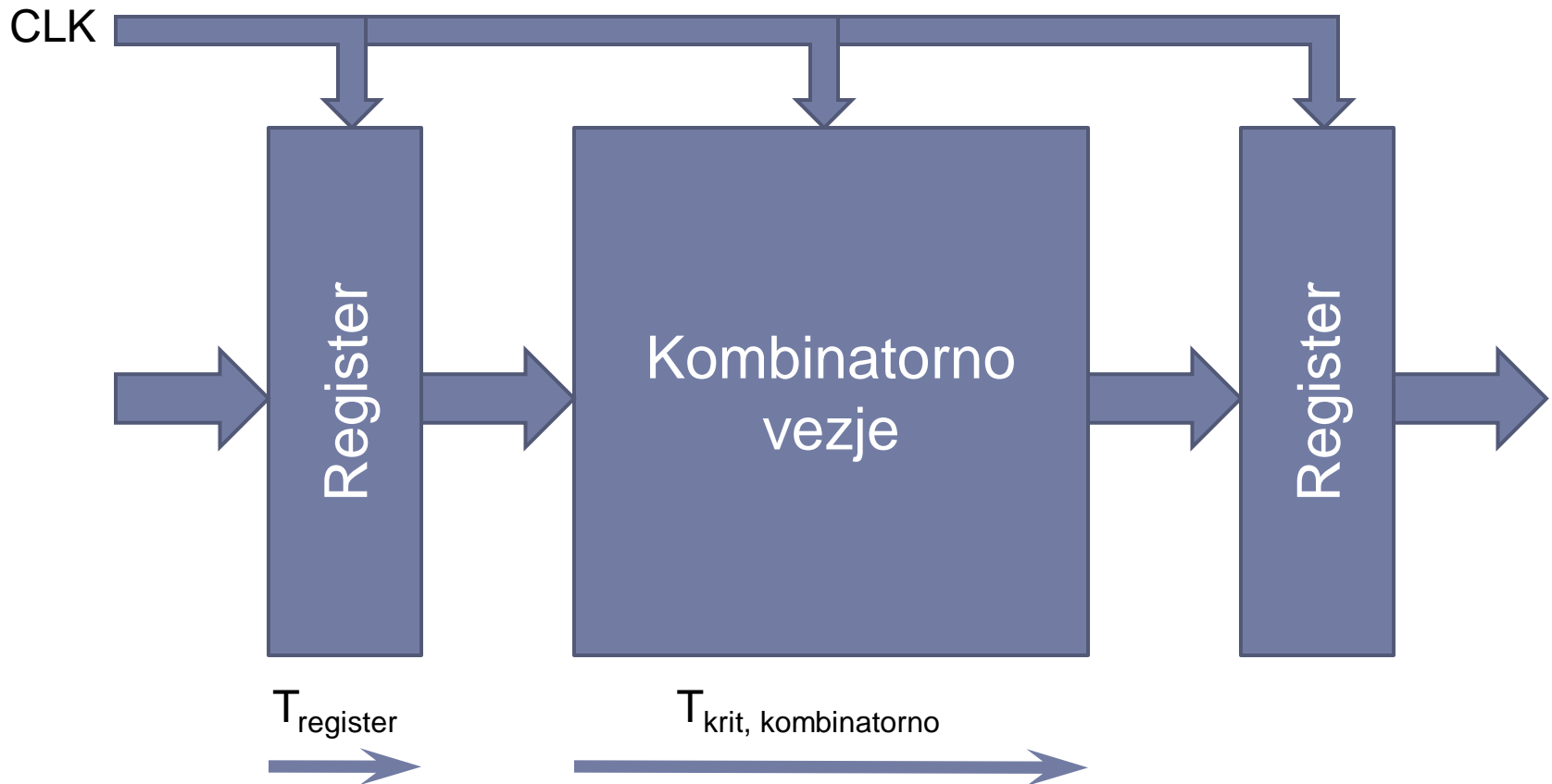
Primer: elektronska ključavnica



- ▶ Če je f_{CLK} dovolj velika niti ne opazimo, da se avtomat odziva le na vsake $T=1/f_{CLK}$ časa

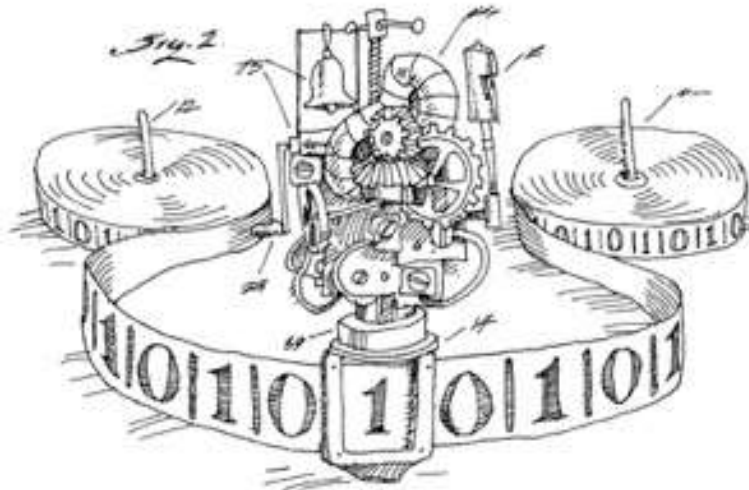
Maksimalna frekvenca signala ure

- ▶ $1/f_{\text{MAX}} = T_{\text{register}} + T_{\text{krit, kombinatorno}}$



Mikroprocesor je avtomat

- ▶ Kaj dela avtomat, povemo z vezjem
- ▶ Kaj bo mikroprocesor delal, povemo s programom
- ▶ Mikroprocesor je torej neke vrste univerzalen avtomat
- ▶ Namenski avtomati so ponavadi hitrejši od rešitev z mikroprocesorjem in imajo manjšo porabo energije



Asinhroni mikroprocesorji

- ▶ Nimajo vhodnega CLK signala
Majhna poraba, ko mikroprocesor nima dela
Majhen EMI (ElectoMagnetic Interference)
- ▶ Caltech CAM (1988)
RISC, 5MIPS, 2V, 5.2mA (približno 10mW)
- ▶ AMULET (Univ. v Manchestru),
ARM arhitektura v asinhroni izvedbi
AMULET1 (1990-1993)
AMULET3 (2000) – enakovreden ARM9TDMI
(>100MIPS)
- ▶ Problem – načrtovanje asinhronnega vezja je težje

Izvedba avtomata ... z mikroprocesorjem

```
#include "io.h"
// Kjučavnica - program za SARM, kot bi ga napisali v 1. letniku
int main(void) {
    char tipka;
    int i=0;
    // Tipke: '0'=A, '1'=B, '2'=C '3'=zakleni. Če lučke gorijo, so vrata odklenjena.
    char kombinacija[]="201";
    int dolzinaKombinacije=3;
    _KeyInit();
    _LEDInit();

    while (1) {
        tipka = getch();
        if (i==dolzinaKombinacije) {
            // Vrata so odklenjena, vsaka tipka jih zaklene nazaj
            i=0;
        } else if (tipka==kombinacija[i]) {
            // Nov pravilen znak
            i++;
        } else {
            // V vseh ostalih primerih zakleni vrata
            i=0;
        }
        // Krmiljenje ključavnice (ponazorjeno z lučkami)
        if (i==dolzinaKombinacije)
            _setleds(0xf);
        else
            _clrleds(0xf);
    }
}
```

Opis ključavnice v jeziku C
(za mikroprocesor)

Načrtovanje sinhronega avtomata s pomočjo opisnega jezika (HDL)

```
library IEEE;
use IEEE.std_logic_1164.all;

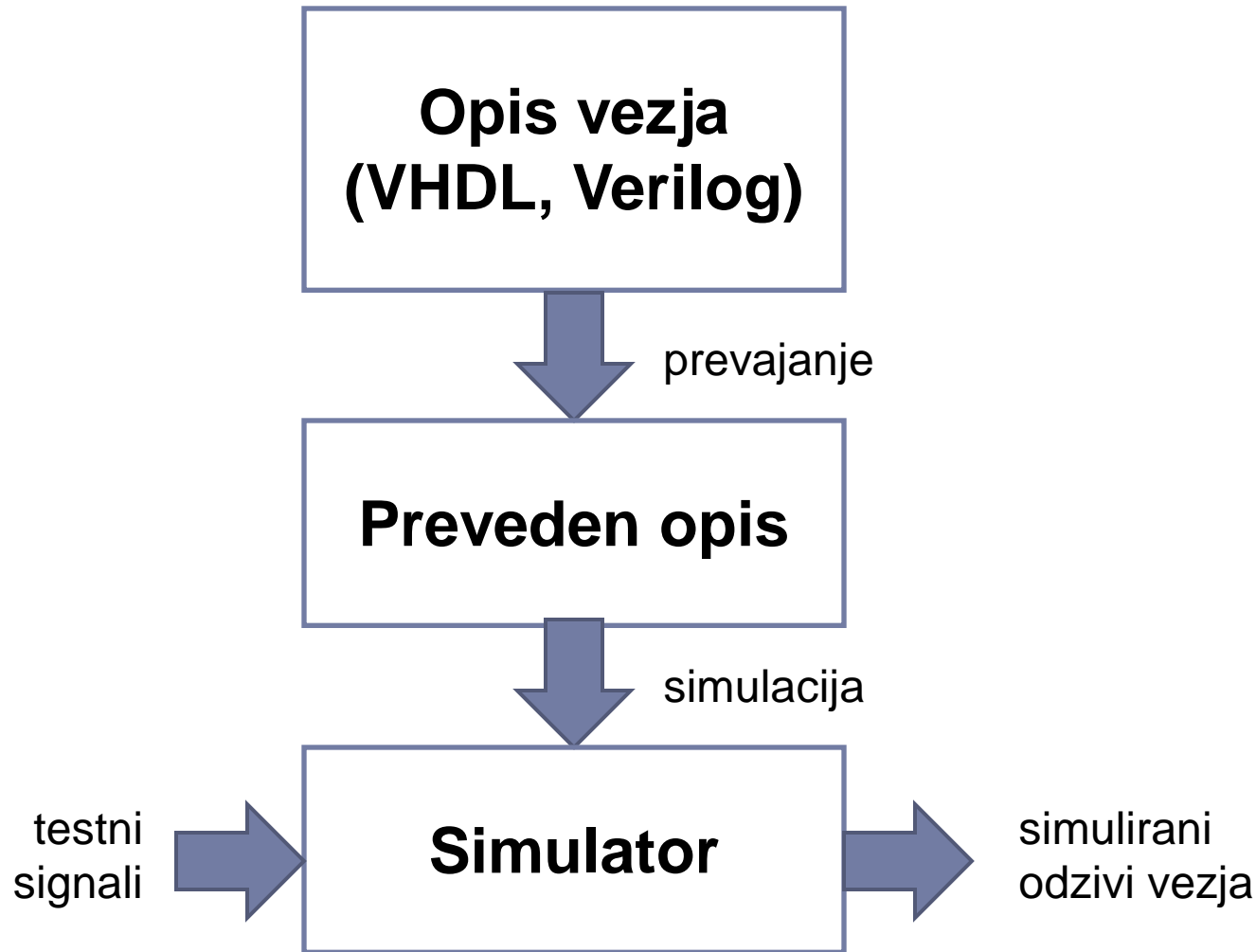
entity kljucavnica is
  port(a, b, c, zakleni, clk: in std_logic; odkleni : out std_logic);
end kljucavnica;

architecture izvedba of kljucavnica is
  -- "00" = zaklenjeno, "01" = en prav, "10" = dva prav, "11" = odklenjeno
  signal stanje: std_logic_vector (1 downto 0);
begin
  process (clk)
  begin
    if (clk'event and clk='1') then
      if (stanje="00" and a='0' and b='0' and c='1' and zakleni='0') then
        stanje <= "01";
        odkleni <= '0';
      elsif (stanje="01" and a='1' and b='0' and c='0' and zakleni='0') then
        stanje <= "10";
        odkleni <= '0';
      elsif (stanje="10" and a='0' and b='1' and c='0' and zakleni='0') then
        stanje <= "11";
        odkleni <= '1';
      elsif (not (a='0' and b='0' and c='0' and zakleni='0')) then
        stanje <= "00";
        odkleni <= '0';
      end if;
    end if;
  end process;
end izvedba;
```

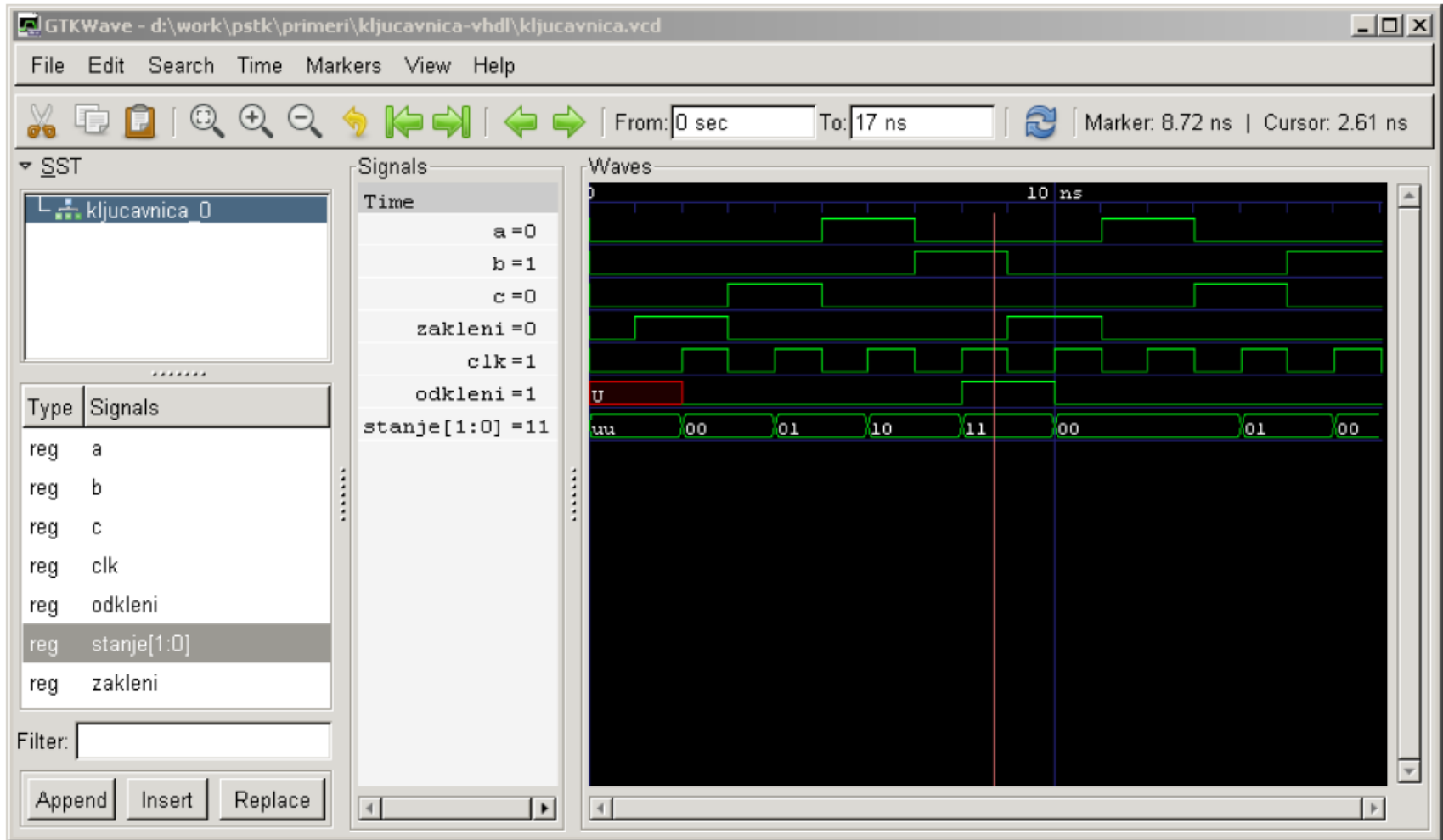
Opis ključavnice v jeziku VHDL (opis namenskega vezja)

Od opisa do delujočega vezja

Simulacija opisa



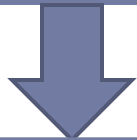
Simulacija opisa v jeziku VHDL



Od opisa do delujočega vezja

Sinteza vezja iz opisa

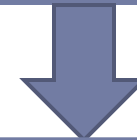
**HDL opis vezja
(VHDL, Verilog)**



sinteza (a)

**Vhodna
datoteka za
programirljivo
logično vezje**

**Opis vezja
(VHDL, Verilog)**



sinteza (b)

**Maske za
postopek
izdelave vezja
(v siliciju)**

Programirljiva logična vezja

- ▶ Vezje programira proizvajalec sistema, v katerem je uporabljen programirljivo vezje PAL, PLA (1978)
GAL (1985) – zbrisljivo,
CPLD – do 100 000 vrat
- ▶ Vezje se programira ob vklopu napajanja – konfiguracija se naloži vedno znova iz bralnega pomnilnika (npr. EEPROM)
FPGA vezja – 1 000 000 vrat v enem vezju (leta 2000)
XILINX, ALTERA (80% tržišča)
XILINX Spartan-3 – 4 000 000 vrat, 70Eur pri 100 kosih



ALTERA MAX 7000
(CPLD z 2500 vrati)

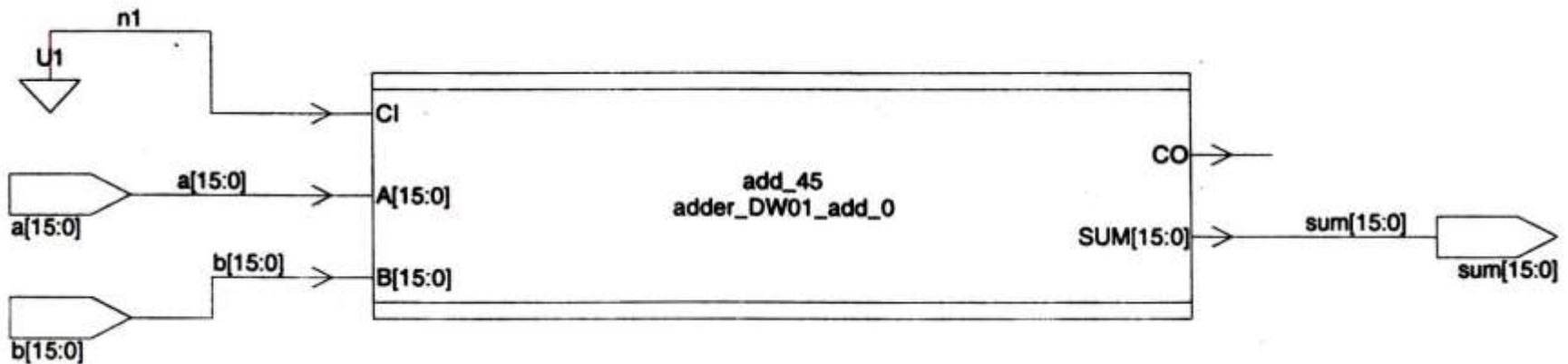
Sinteza logičnih vezij (izvedba v siliciju)



Samo za sinhrona vezja ☹️

Primer

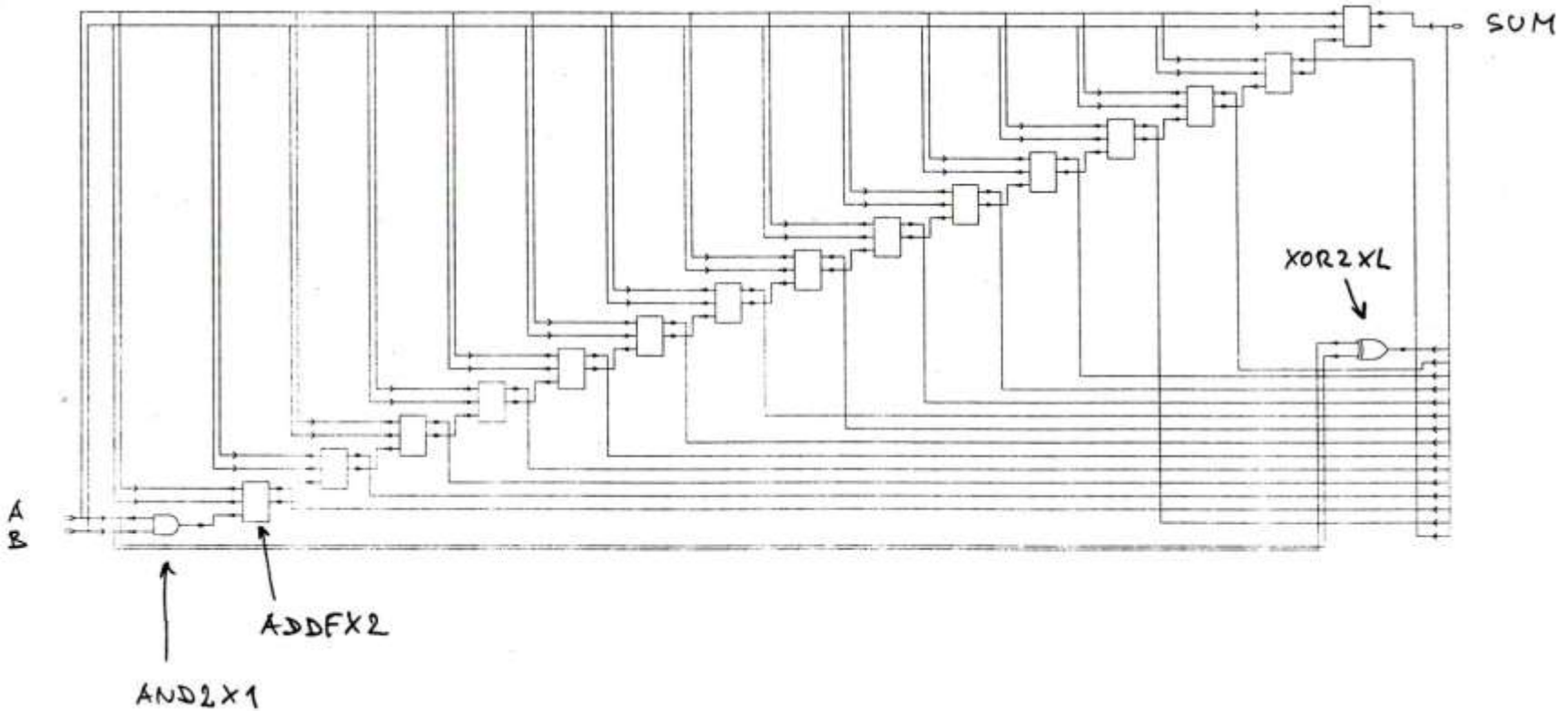
Sinteza 16-bitnega seštevalnika



- ▶ 16 vhodov A, 16 vhodov B, vhod CI za prenos
- ▶ 16 izhodov za rezultat, izhod CO za izhodni prenos
- ▶ 0.25um tehnologija

Sinteza 16-bitnega seštevalnika

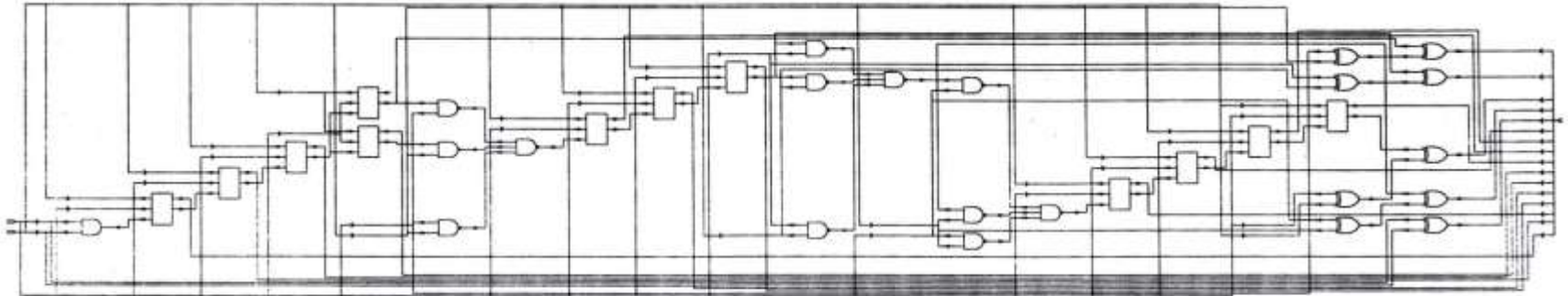
Zahtevana hitrost odziva: 8ns (125MHz)



- ▶ Kritična pot: 7.82ns, 16 nivojev, 17 celic
- ▶ Poraba energije: 5.26
- ▶ Celice: 1-bitni seštevalniki, AND, EXOR

Sinteza 16-bitnega seštevalnika

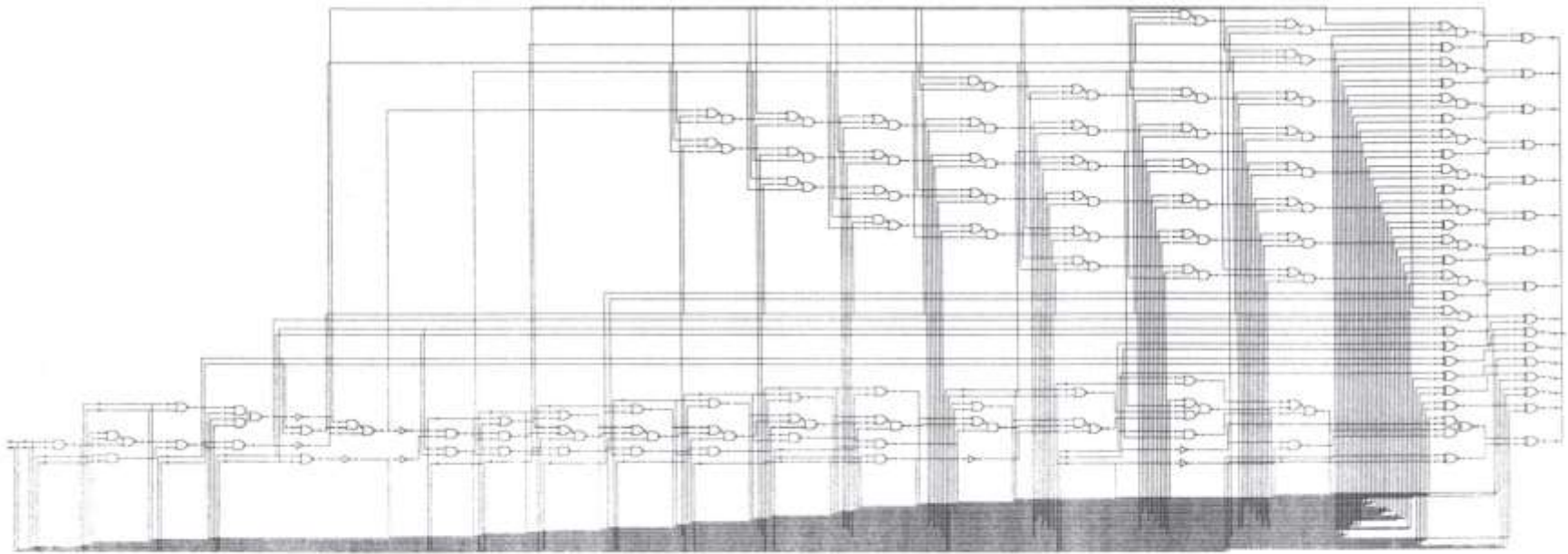
Zahtevana hitrost odziva: 7ns (140MHz)



- ▶ Kritična pot: 6.97ns
- ▶ 19 nivojev, 34 celic
- ▶ Poraba energije: 6.45
- ▶ Celice: 1-bitni seštevalniki, AND, NAND, EXOR

Sinteza 16-bitnega seštevalnika

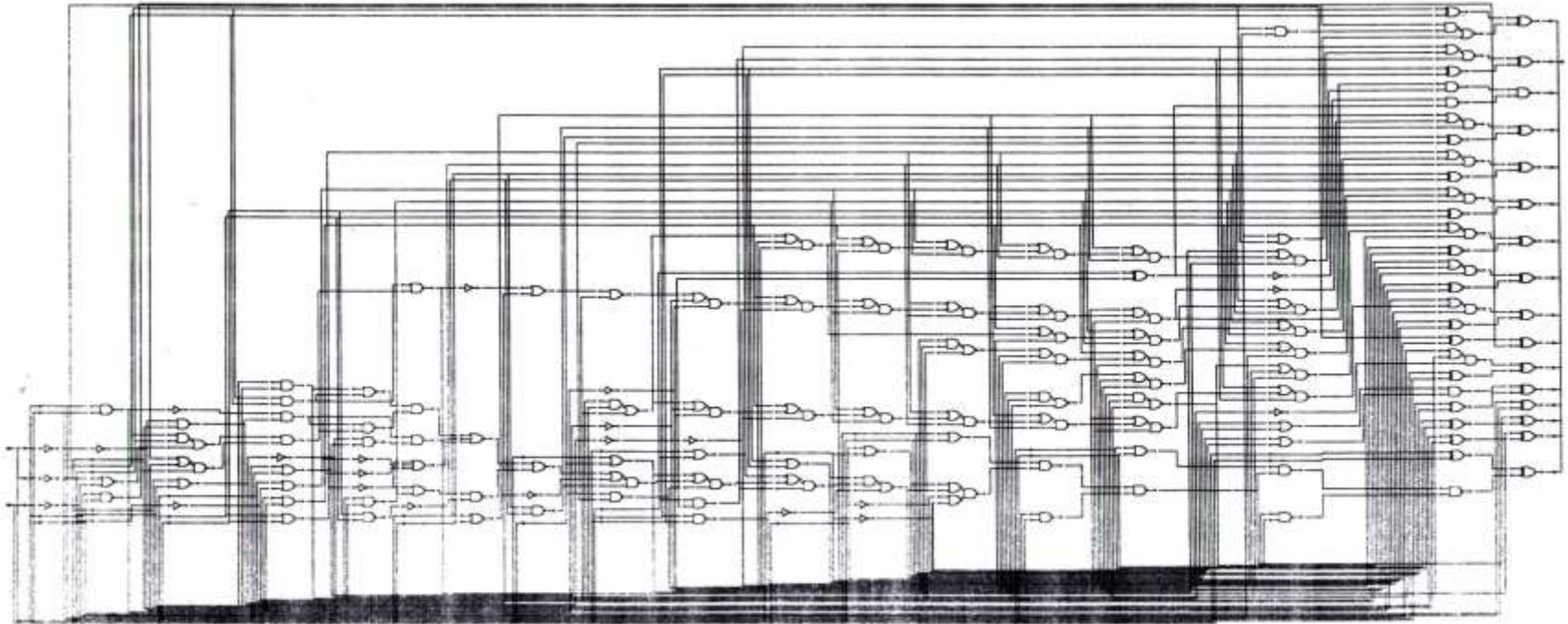
Zahtevana hitrost odziva: 5ns (200MHz)



- ▶ Kritična pot: 4.99ns
- ▶ 18 nivojev, 124 celic
- ▶ Poraba energije: 10.44
- ▶ Celice: negatorji, NAND, NOR, AND, OR, EXOR

Sinteza 16-bitnega seštevalnika

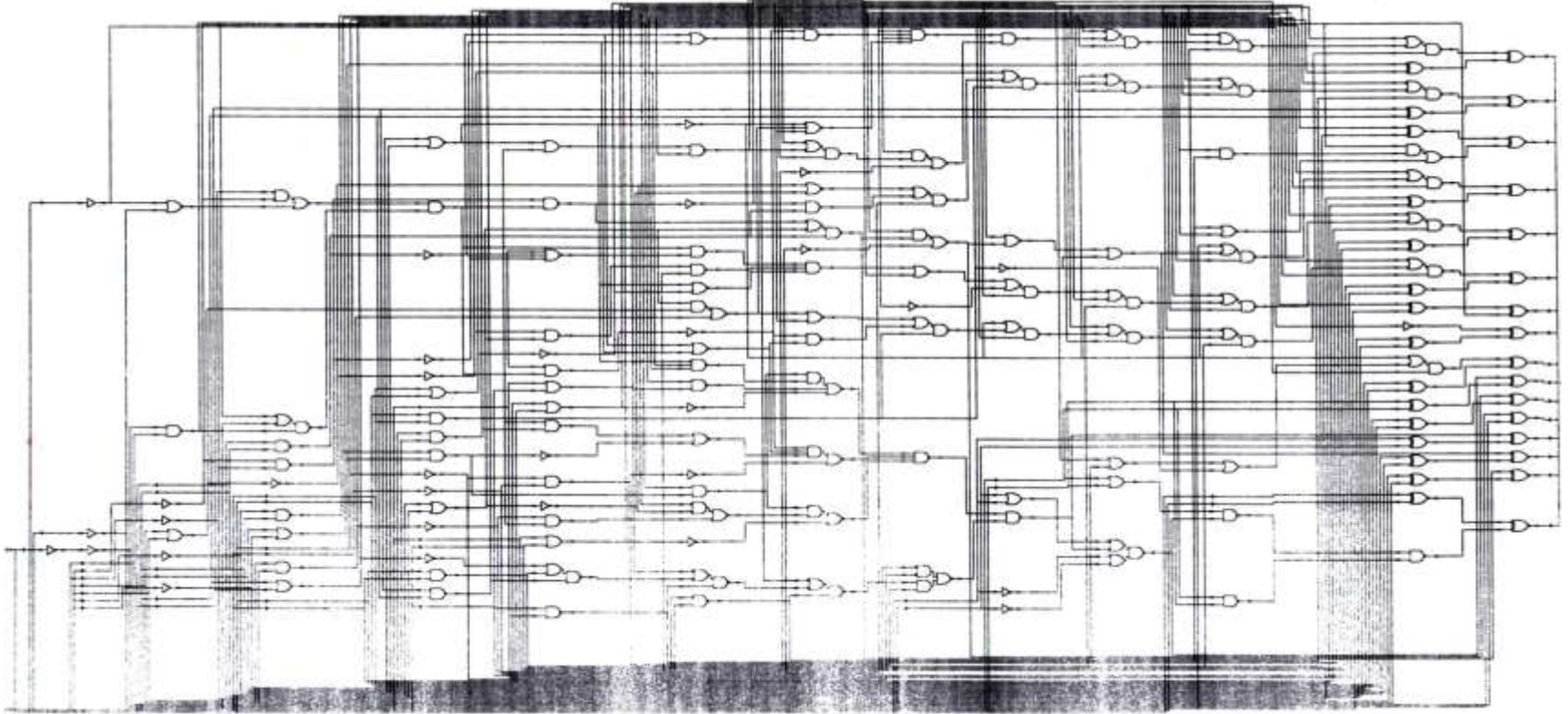
Zahtevana hitrost odziva: 4ns (250MHz)



- ▶ Kritična pot: 3.99ns
- ▶ 14 nivojev, 154 celic
- ▶ Poraba energije: 13.44
- ▶ Celice: negatorji, NAND, NOR, AND, OR, EXOR

Sinteza 16-bitnega seštevalnika

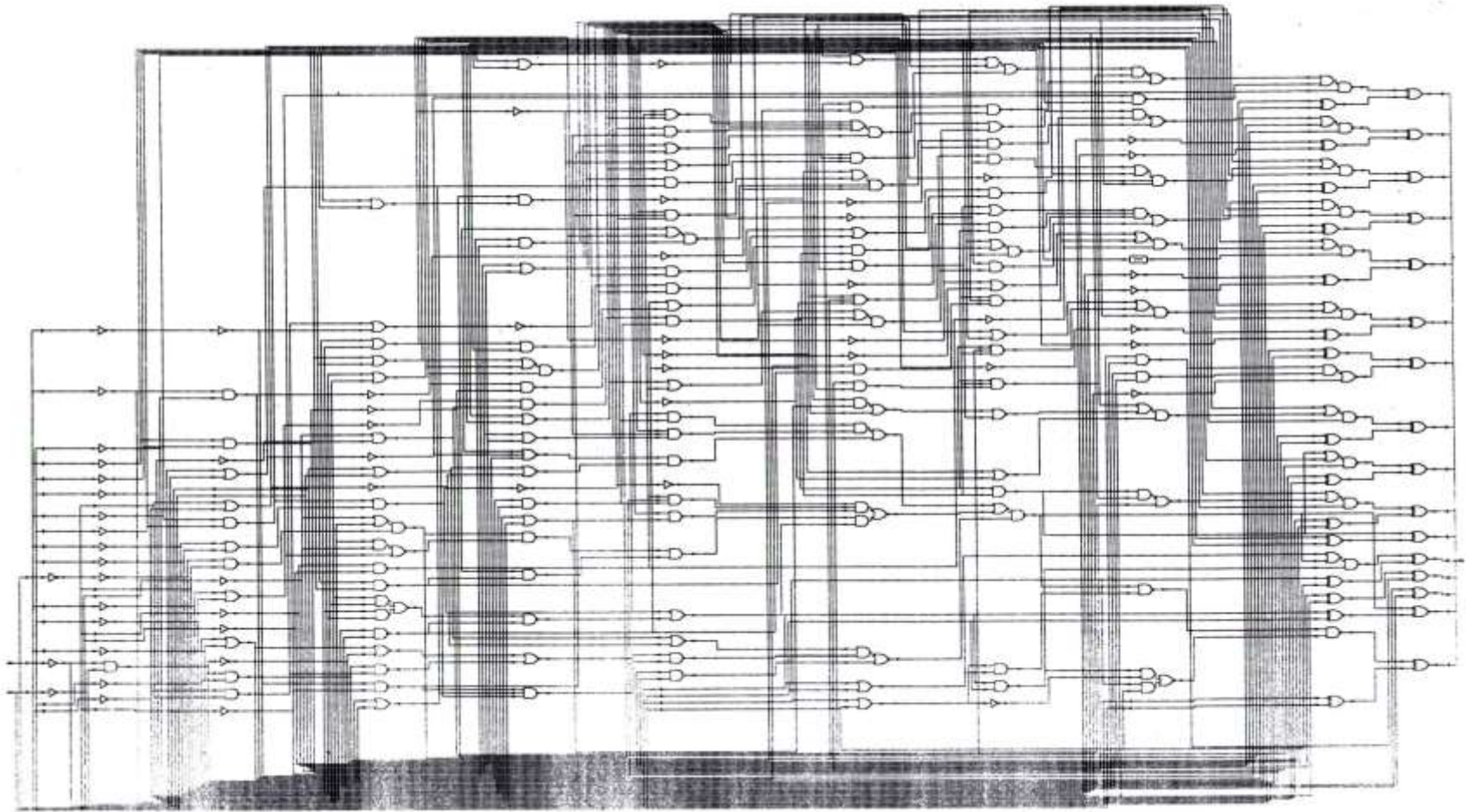
Zahtevana hitrost odziva: 3ns (330MHz)



- ▶ Kritična pot: 3.00ns
- ▶ 12 nivojev, 164 celic, poraba energije: 16.79
- ▶ Celice: negatorji, NAND, NOR, AND, OR, EXOR

Sinteza 16-bitnega seštevalnika

Zahtevana hitrost odziva: 2ns (400MHz)



- ▶ Kritična pot: **2.19ns**, 9 nivojev, 225 celic, poraba 28.19