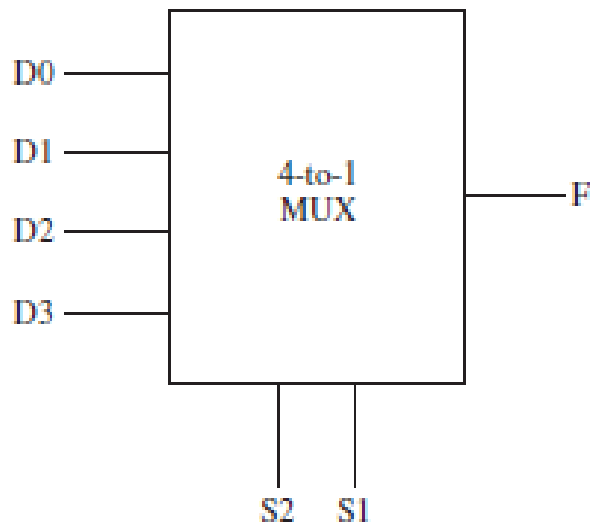


DIGITALNA LOGIKA

Multiplexeri

- Multiplexer povezuje više ulaza sa jednim izlazom



| S2 | S1 | F |
|----|----|----|
| 0 | 0 | D0 |
| 0 | 1 | D1 |
| 1 | 0 | D2 |
| 1 | 1 | D3 |

Implementacija mux koriscenjem I, ILI i NE logickih kola

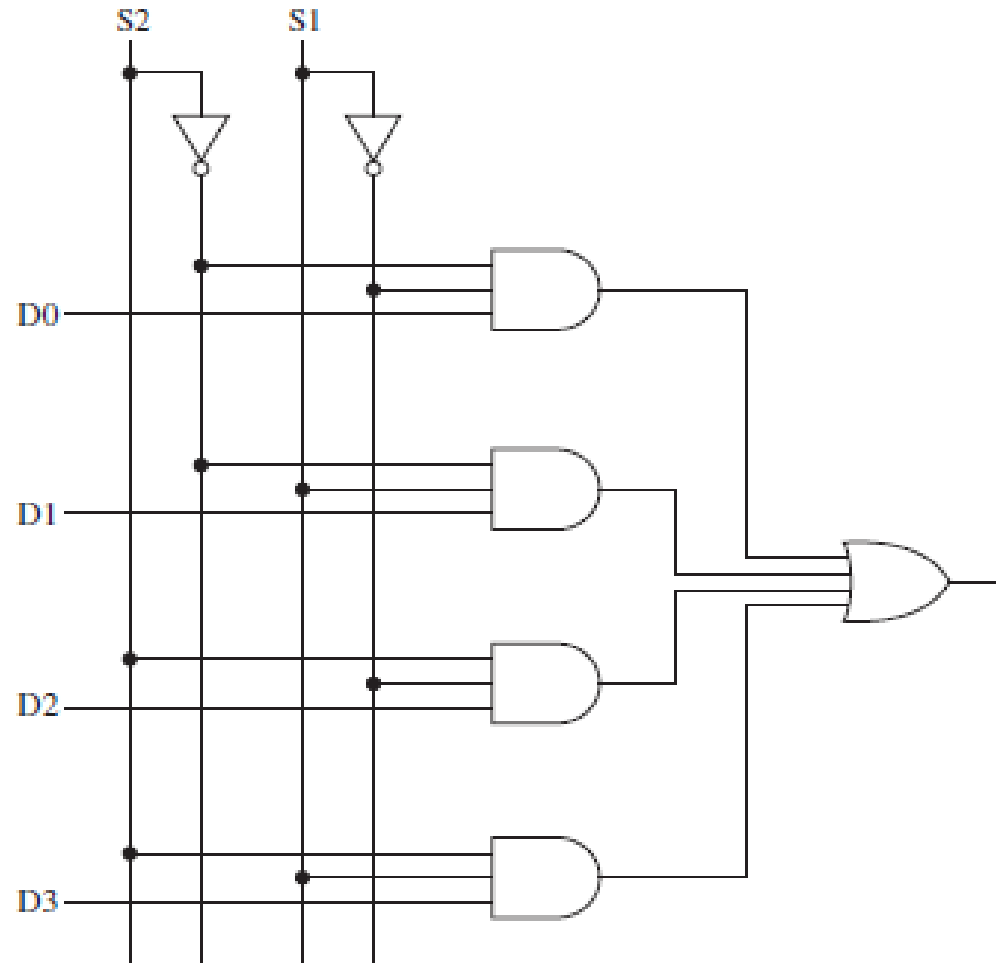


Figure 20.13 Multiplexer Implementation

Implementacija mux koriscenjem I, ILI i NE logickih kola

- Za bilo koju kombinaciju S1 i S2 izlazi iz tri I kola bice 0.
- Cetvrto I kolo imace vrednost izabrane linije.
- Tri ulaza u ILI kolo su 0
- Izlaz ILI kola je vrednost izabrane linije.

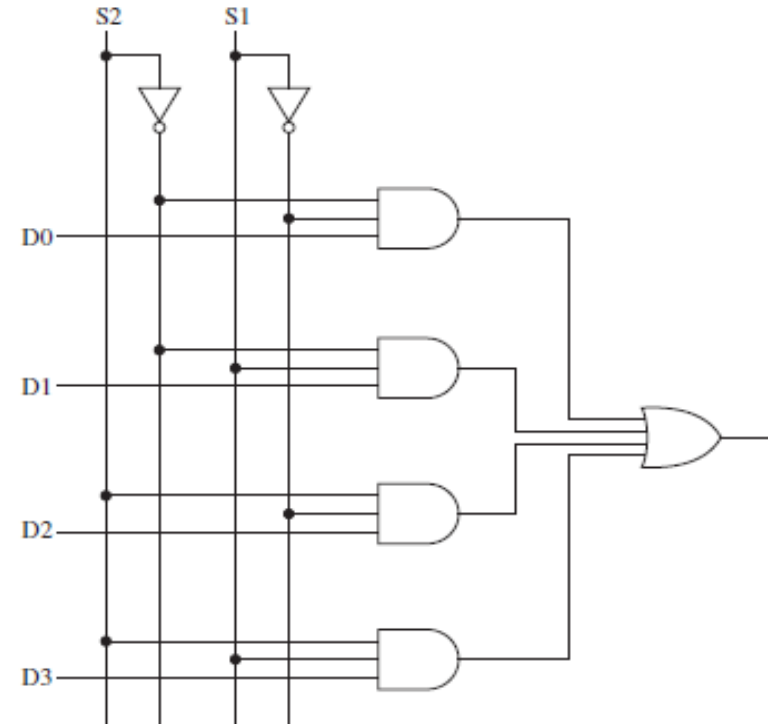


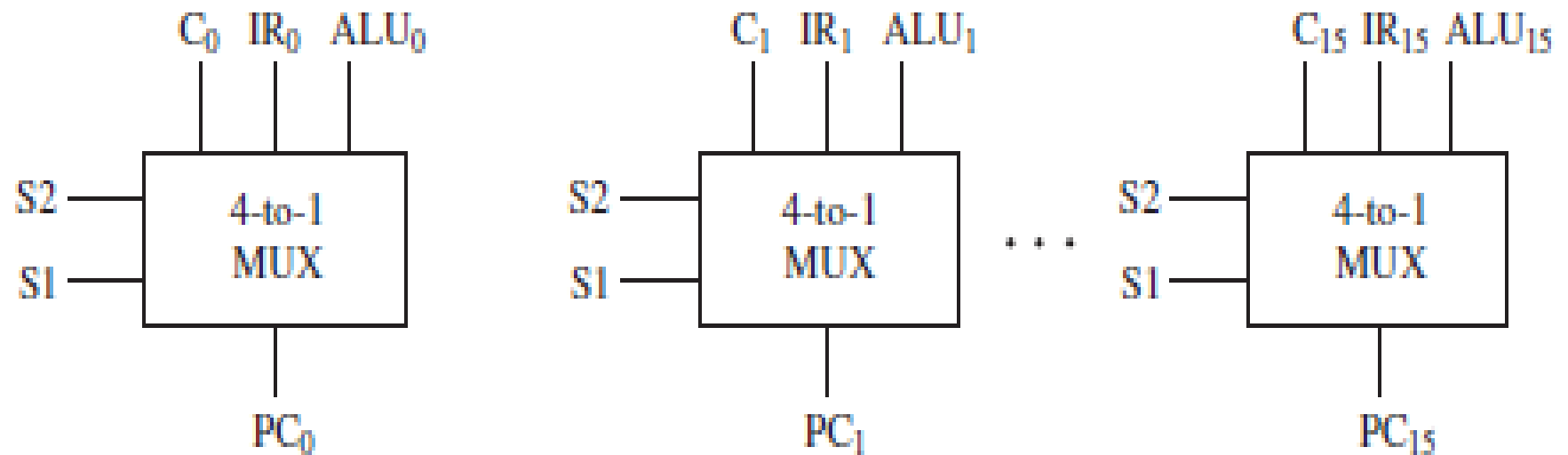
Figure 20.13 Multiplexer Implementation

Ucitavanje programskog brojaca

Programski brojac moze da dobije vrednost iz:

- Binarnog brojaca, uvecanje PC za sledecu instrukciju
- Iz instrukcijskog registra, ako je završena instrukcija grananja sa direktnom adresom
- Izlaz iz ALU, ako se adresa instrukcije grananja odredjuje na osnovu pomeraja

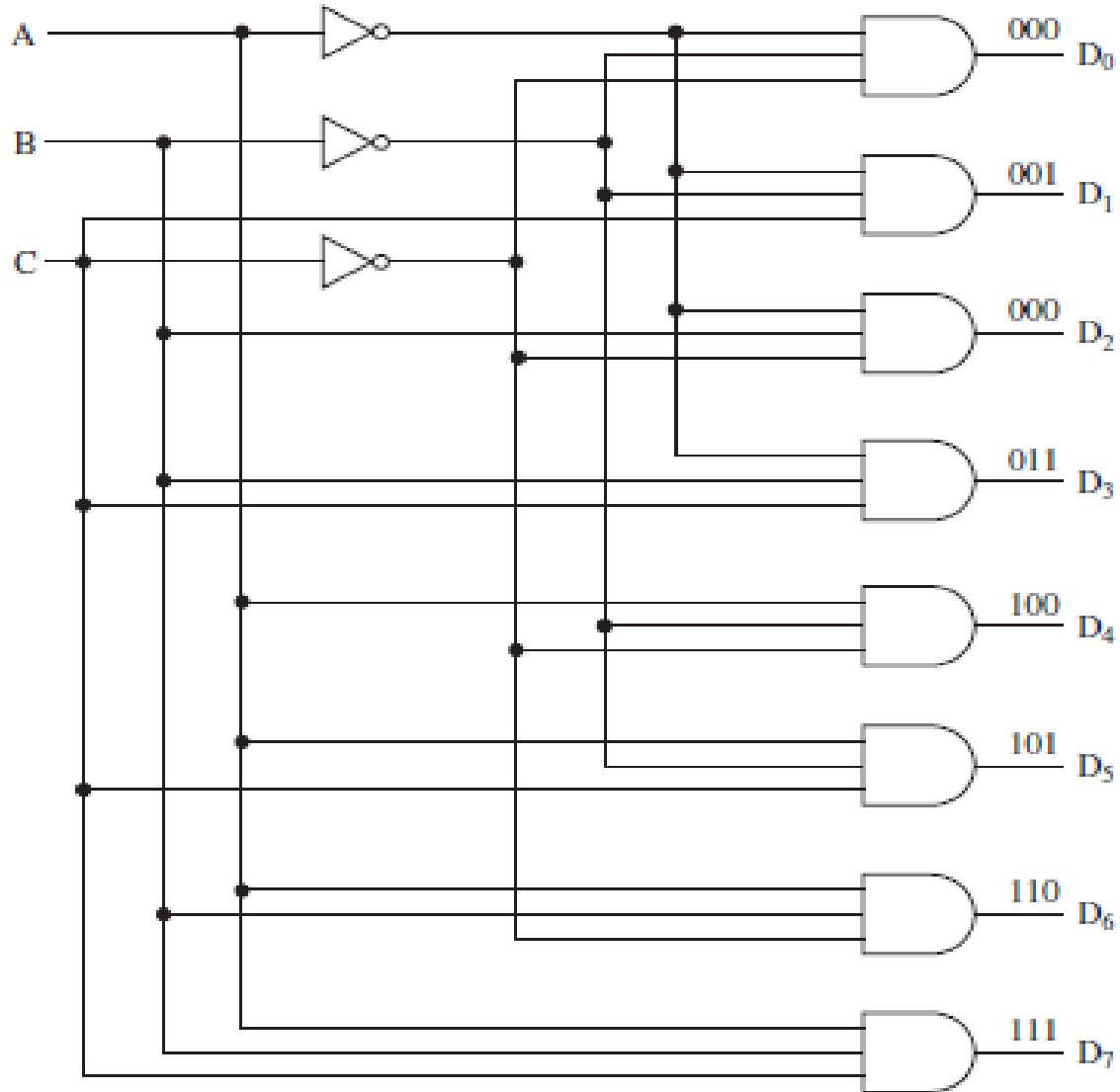
Multiplexerski ulaz u 16-to bitni PC



Dekoderi

- Vise izlaznih linija pri cemu u jednom trenutku postoji vrednost samo na jednoj liniji.
-

Dekoderi



Dekodiranje adresa 1kBajt Memorija

- Sinteza 1kBajt memorije kada na raspolaganju imamo 4 256x8-bitnih RAM cipova.
- Adresni prostor je jedinstven koji u skladu sa cinjenicom da imamo 4 cipa izgleda kao na slici,

Dekodiranje adresa 1kBajt Memorija

| Address | Chip |
|----------------|-------------|
| 0000–00FF | 0 |
| 0100–01FF | 1 |
| 0200–02FF | 2 |
| 0300–03FF | 3 |

2^{10} 1K

2^{20} 1M

2^{30} 1G

Dekodiranje adresa 1kBajt Memorija

- Svaki cip zahteva 8 adresnih linija I to su nizih 8 bitova adrese.
- Dva bita viseg reda 10-to bitne adrese koriste se za selekciju jednog od 4 cipa RAM memorije.
- Koristi se dekođer $2u4$ ciji izlazi ukljucuju pojedine cipove

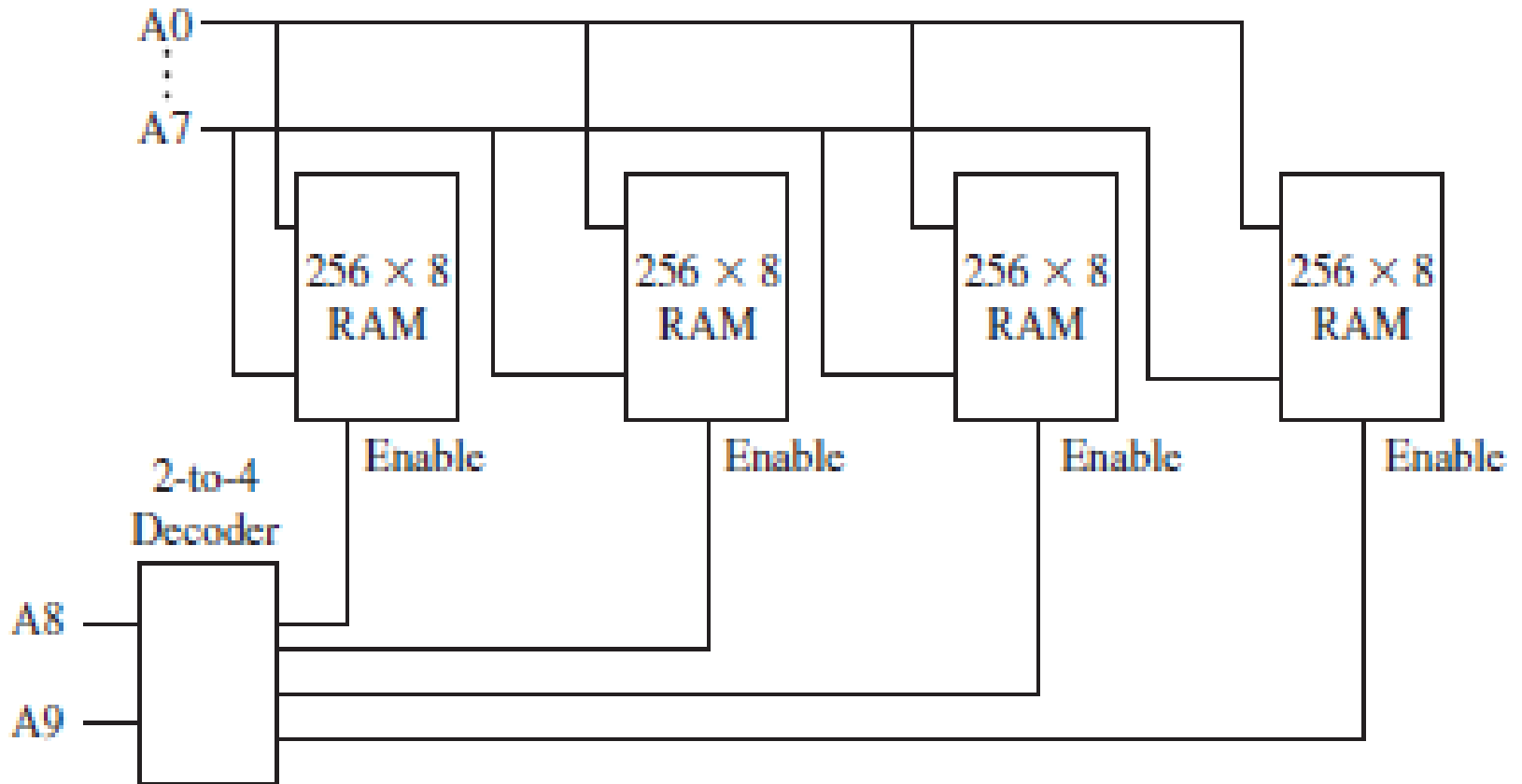


Figure 20.16 Address Decoding

Demultiplekser koriscenjem dekodera

- Dodavanjem jedne ulazne linije dekodera se može koristiti kao demultiplekser.
- Demux ima inverznu funkciju od mux-a, povezuje jedan ulaz sa jednim od više izlaza.

Demultiplekser koriscenjem dekodera

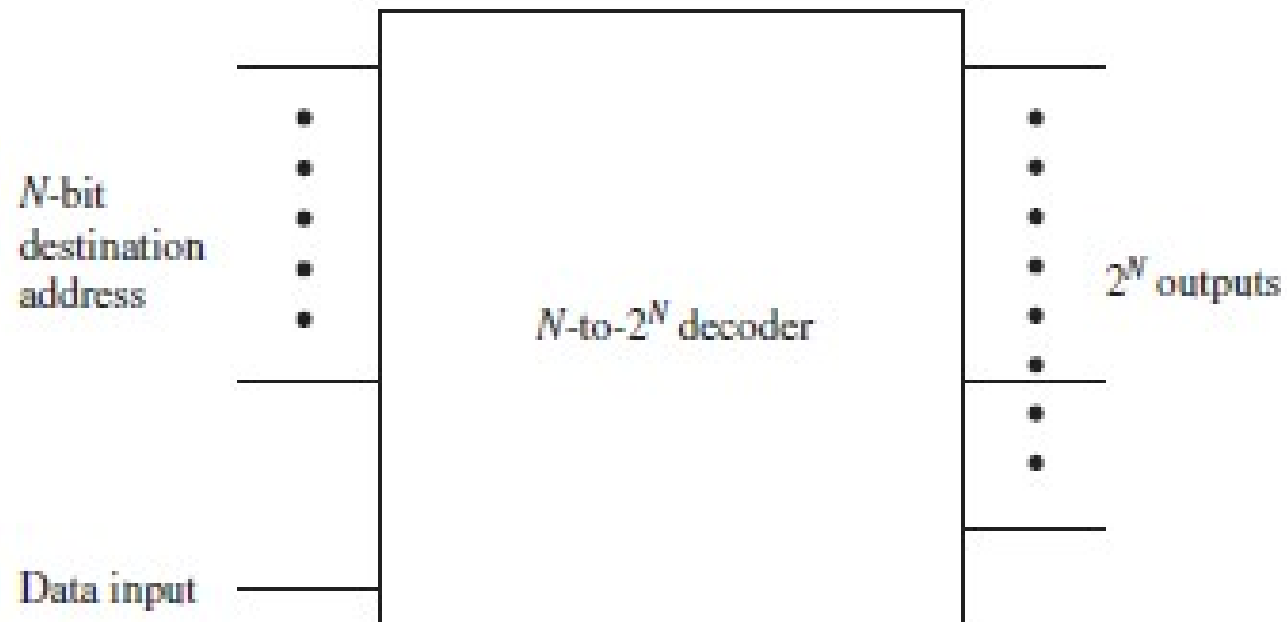


Figure 20.17 Implementation of a Demultiplexer Using a Decoder

ROM memorija

- Kombinaciona kola su kola bez memorije jer njihov izlaz zavisi samo od trenutnog ulaza I ne pamti se podatak o prethodnim ulazima.
- ROM memorija koja se koristi samo za citanje implementira se koriscenjem kombinacionih kola.
- Podaci upisani u ROM memoriju su trajni, jednom zapisani tokom proizvodnje.
- Ulazi u ROM memoriju su adrese, izlazi su su podaci I oni zavise samo od postojanja adrese, a to je u sustini kombinaciono kolo.

ROM memorija

- ROM memorija moze da se sintetise koriscenjem dekodera I ILI kola.
- Posmatrajmo tablicu istinitosti sa cetiri ulaza I cetiri izlaza.

ROM memorija

Table 20.8 Truth Table for a ROM

| Input | | | | | Output | | | | |
|-------|---|---|---|---|--------|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

ROM memorija

- Za svaku od 16 mogućih ulaznih vrednosti, adresa dobija se odgovarajući skup vrednosti na izlazu, podataka.
- Ova tabela predstavlja tabelu istinitosti 64-bitne memorije, 16 memorijskih lokacija na kojima je smesten 4-bitni podatak.
- Četiri ulaza su adresa, a 4 izlaza podatak.

ROM memorija koriscenjem dekodera i ILI kola

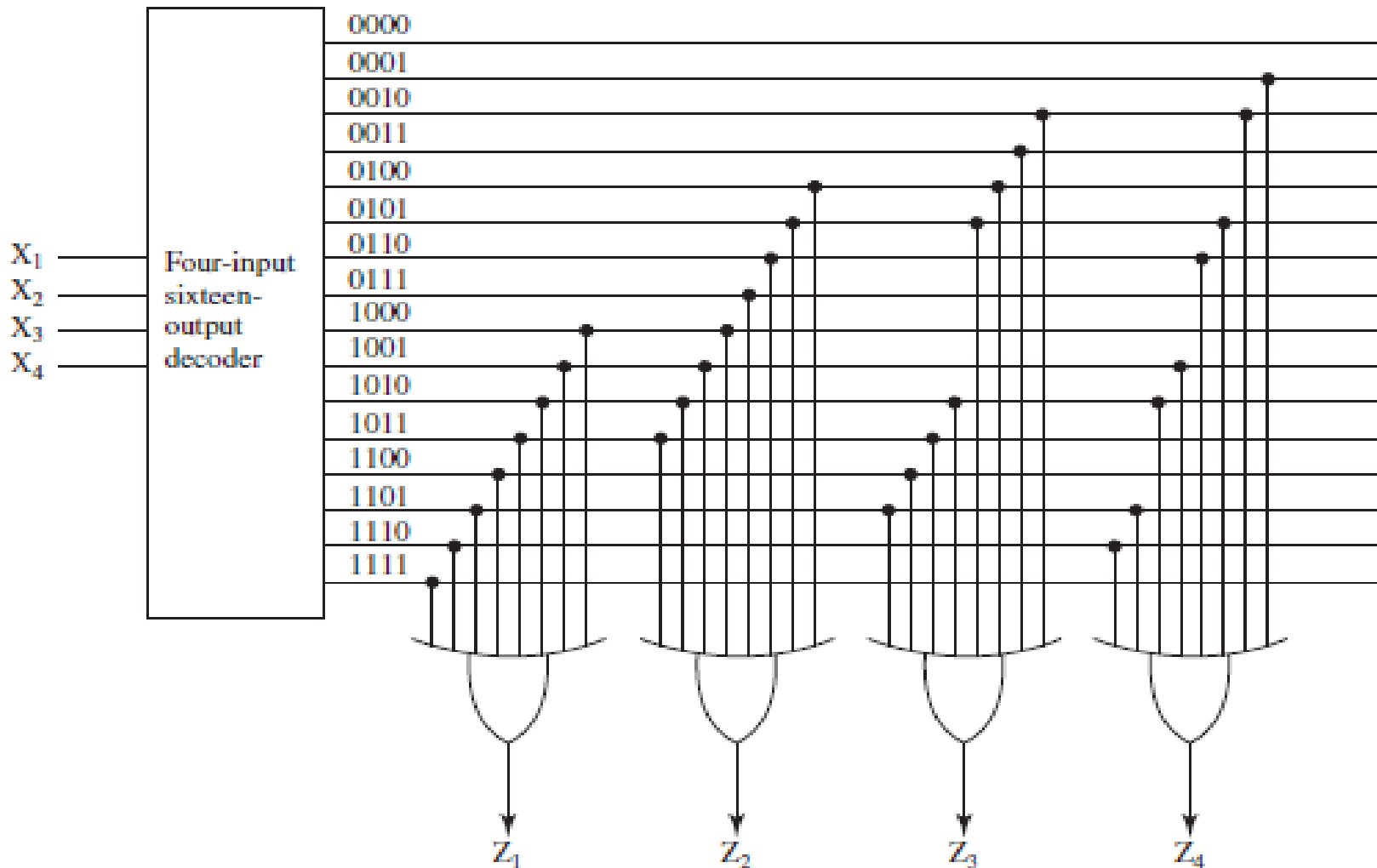


Figure 20.18 A 64-Bit ROM

Sabiraci

Binary addition differs from Boolean algebra in that the result includes a carry term. Thus,

| | | | |
|-----|-----|-----|-----|
| 0 | 0 | 1 | 1 |
| + 0 | + 1 | + 0 | + 1 |
| 0 | 1 | 1 | 10 |

However, addition can still be dealt with in Boolean terms. In Table 20.9a, we show the logic for adding two input bits to produce a 1-bit sum and a carry bit. This truth table

Table 20.9 Binary Addition Truth Tables

| (a) Single-Bit Addition | | | | (b) Addition with Carry Input | | | | |
|-------------------------|---|-----|-------|-------------------------------|---|---|-----|------------------|
| A | B | Sum | Carry | C _{in} | A | B | Sum | C _{out} |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| | | | | 1 | 0 | 0 | 1 | 0 |
| | | | | 1 | 0 | 1 | 0 | 1 |
| | | | | 1 | 1 | 0 | 0 | 1 |
| | | | | 1 | 1 | 1 | 1 | 1 |

Sabirac n-bitnih brojeva

- Bit prenosa jednog jednobitnog sabiraca prestavlja ulaz sledeceg.
- Primer 4-bitog sabiraca

Sabirac n-bitnih brojeva

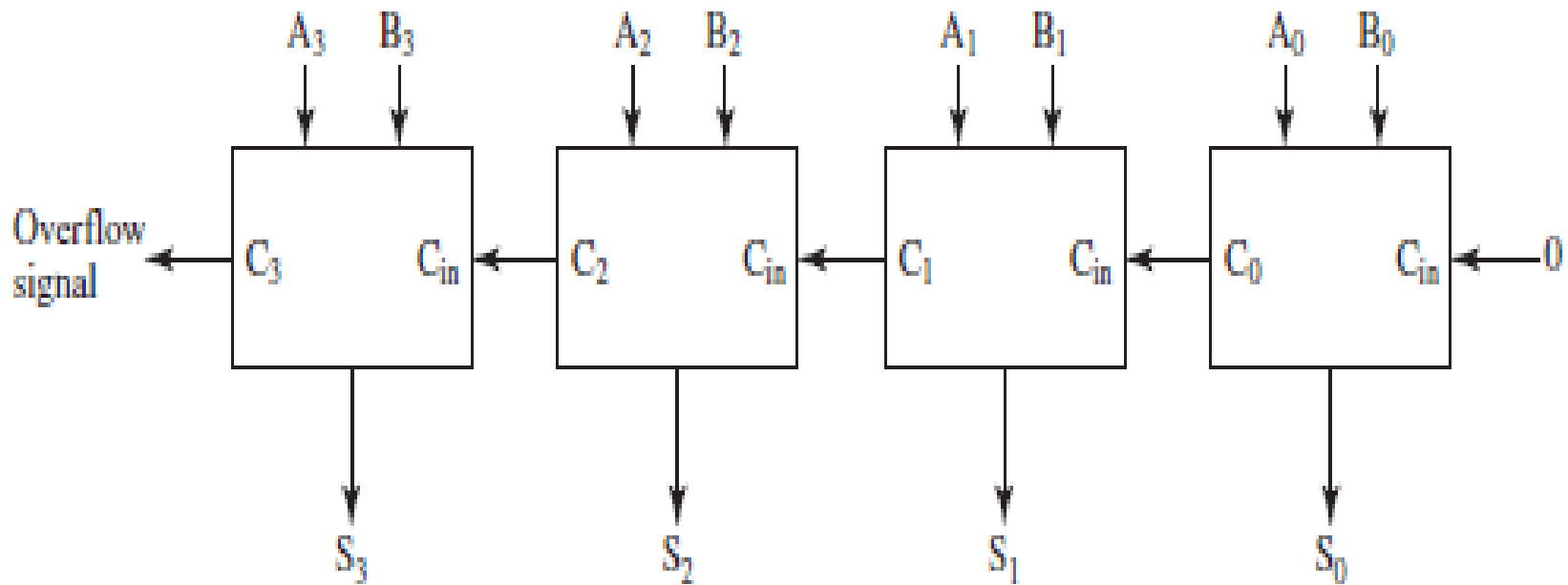


Figure 20.19 4-Bit Adder

Sabirac n-bitnih brojeva

- Da bi visebitni sabirac radio, svi jednobitni sabiraci moraju da imaju tri ulaza, dva za bitove i jedan za prenos iz prethodnog stepena.

Sabirac n-bitnih brojeva

$$\begin{aligned}\text{Sum} &= \overline{A} \overline{B} C + \overline{A} B \overline{C} + A B C + A \overline{B} \overline{C} \\ \text{Carry} &= A B + A C + B C\end{aligned}$$

Implementacija koriscenjem I, ILI i NE kolima

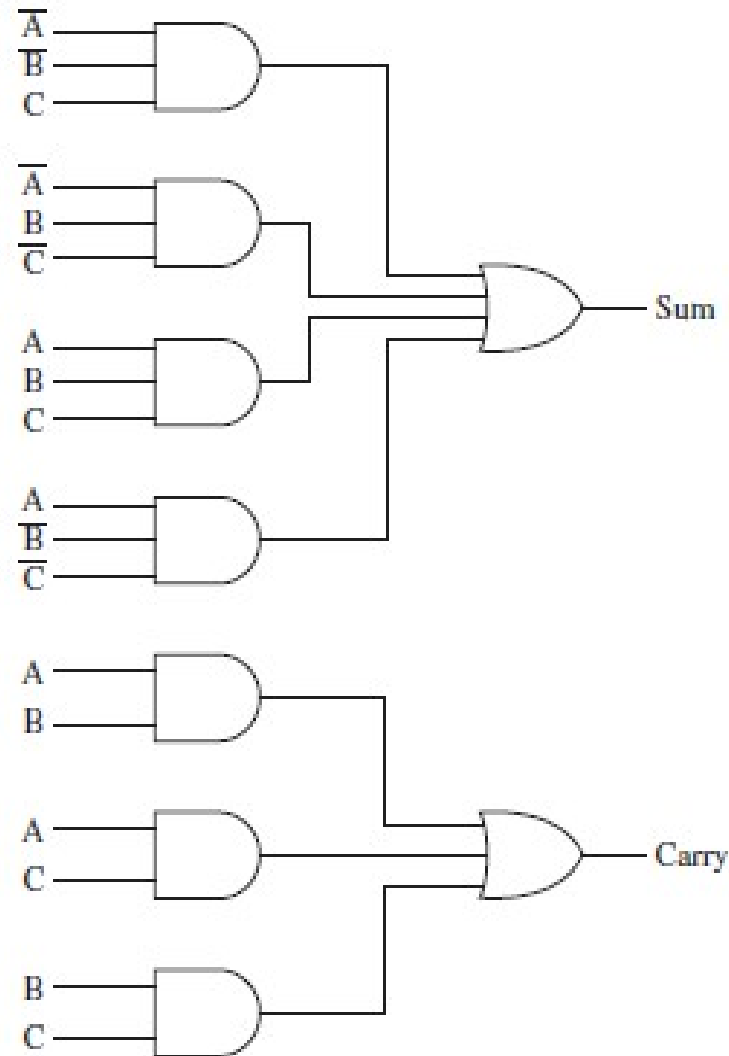


Figure 20.20 Implementation of an Adder

Visebitni sabiraci

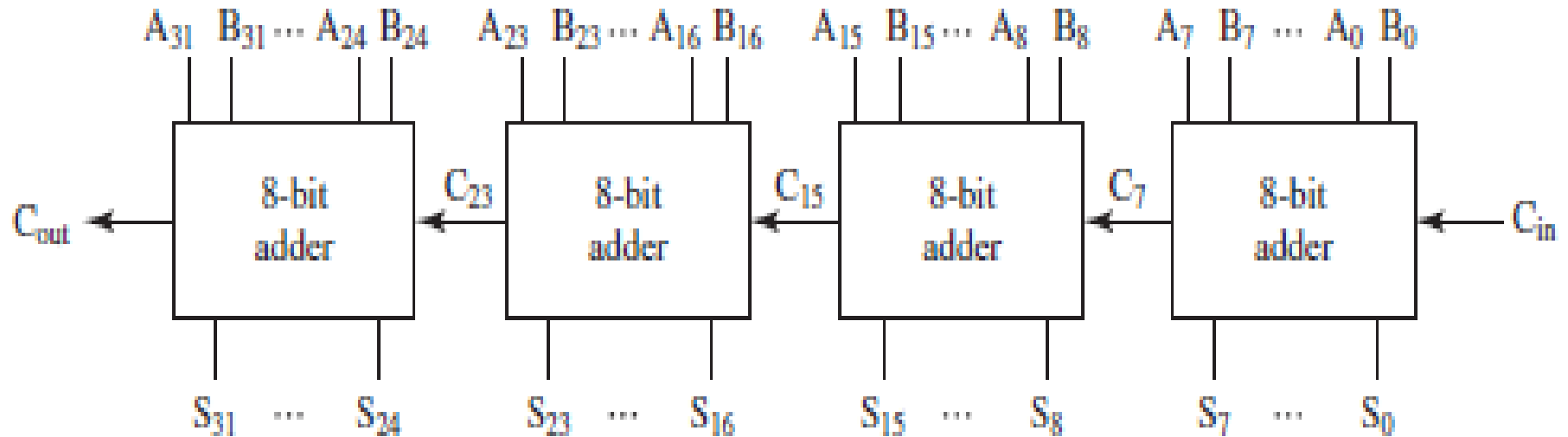


Figure 20.21 Construction of a 32-Bit Adder Using 8-Bit Adders

Visebitni sabirac

- Izlaz svakog sabiraca zavisi od prenosnog bita prethodnog, sto uvodi znacajno kasnjenje od bita najmanje do bita najveće težine.
- Pored toga, postoji i kasnjenje koje imaju sama logicka kola od kojih je dizajniran sabirac, sto moze da dovede ukupno da znacajnog kasnjenja.
- Resenje, sabirac sa predvidjanjem prenosa.

Sekvencijalna kola flip-flopovi

SR elektronski prekidač

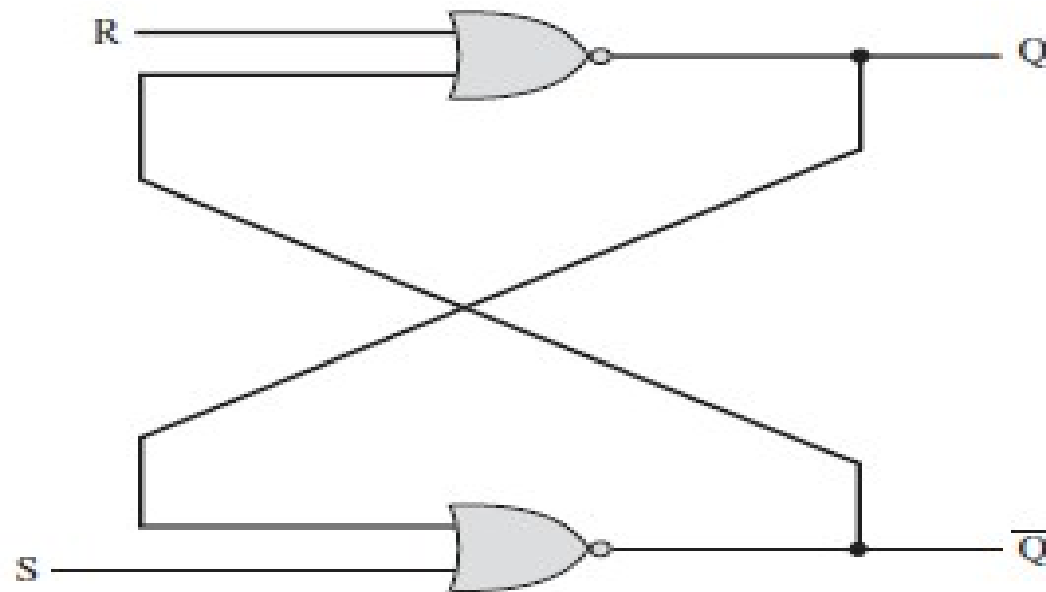


Figure 20.22 The S–R Latch Implemented with NOR Gates

Vremenski dijagram NILI SR elektronskog prekidača

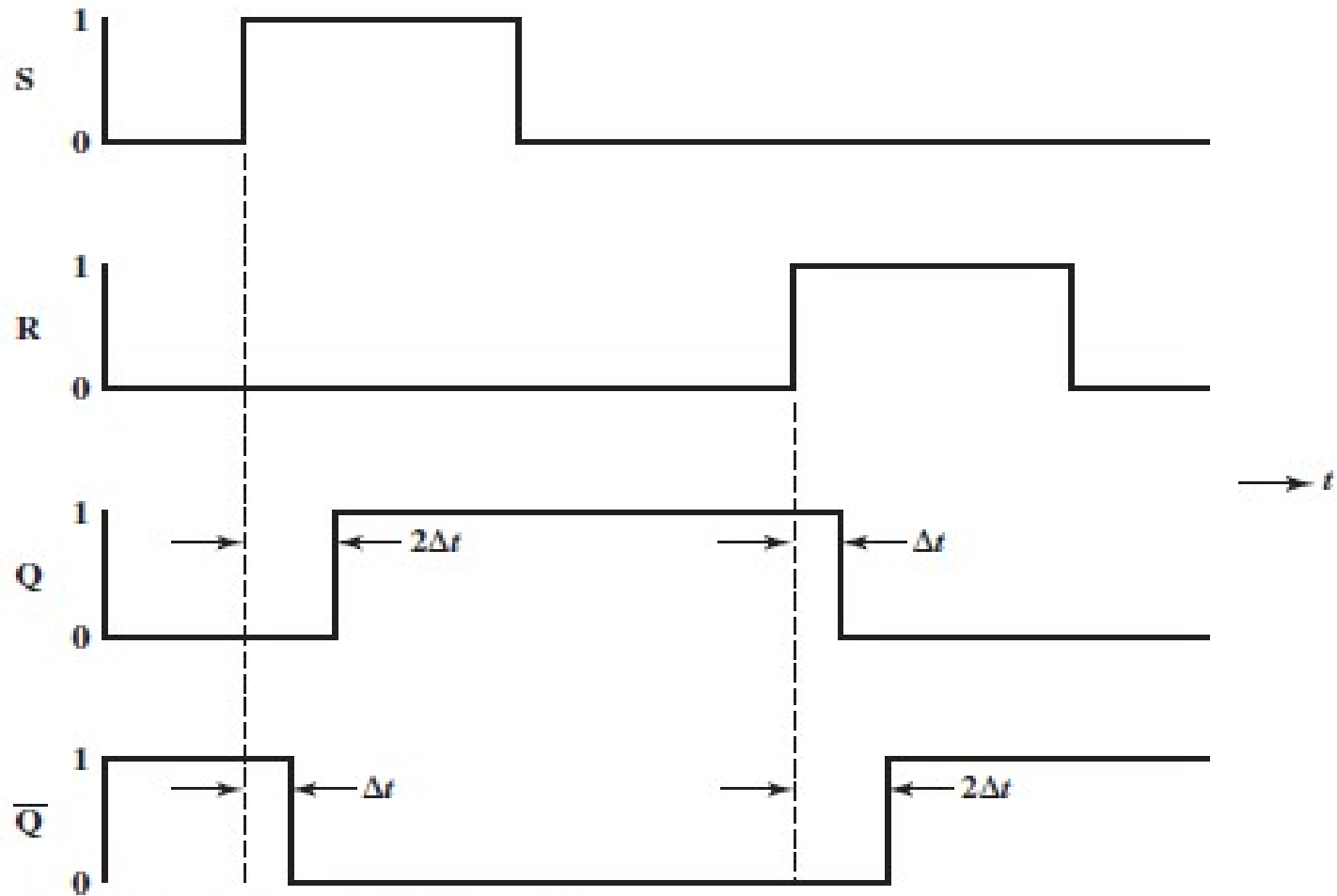


Figure 20.23 NOR S-R Latch timing Diagram

SR elektronski prekidac

Table 20.10 The S–R Latch

| (a) Characteristic Table | | |
|--------------------------|---------------|------------|
| Current Inputs | Current State | Next State |
| SR | Q_n | Q_{n+1} |
| 00 | 0 | 0 |
| 00 | 1 | 1 |
| 01 | 0 | 0 |
| 01 | 1 | 0 |
| 10 | 0 | 1 |
| 10 | 1 | 1 |
| 11 | 0 | — |
| 11 | 1 | — |

| (b) Simplified Characteristic Table | | |
|-------------------------------------|---|-----------|
| S | R | Q_{n+1} |
| 0 | 0 | Q_n |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | — |

| (c) Response to Series of Inputs | | | | | | | | | | |
|----------------------------------|---|---|---|---|---|---|---|---|---|---|
| t | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| S | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| Q_{n+1} | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

SR flip-flop

- Problem kod SR flip flopa je da treba izbegavati stanje $R=1, S=1$.
- Resenje je da se omoguci samo jedan ulaz i to se postize D flip flopom.

SR pobudjen generatorom takta I D flip flop

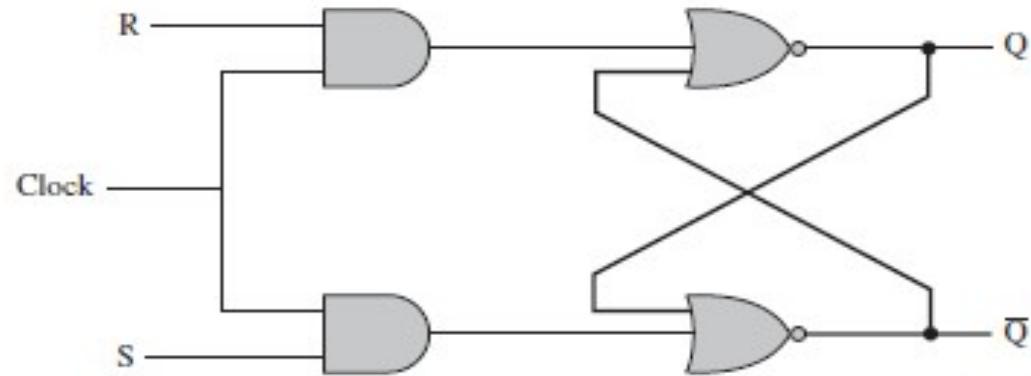


Figure 20.24 Clocked S-R Flip Flop

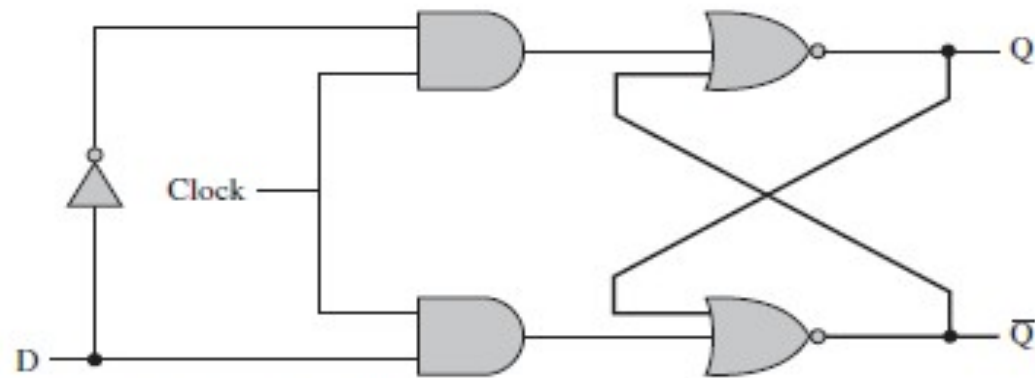


Figure 20.25 D Flip Flop

JK flip flop

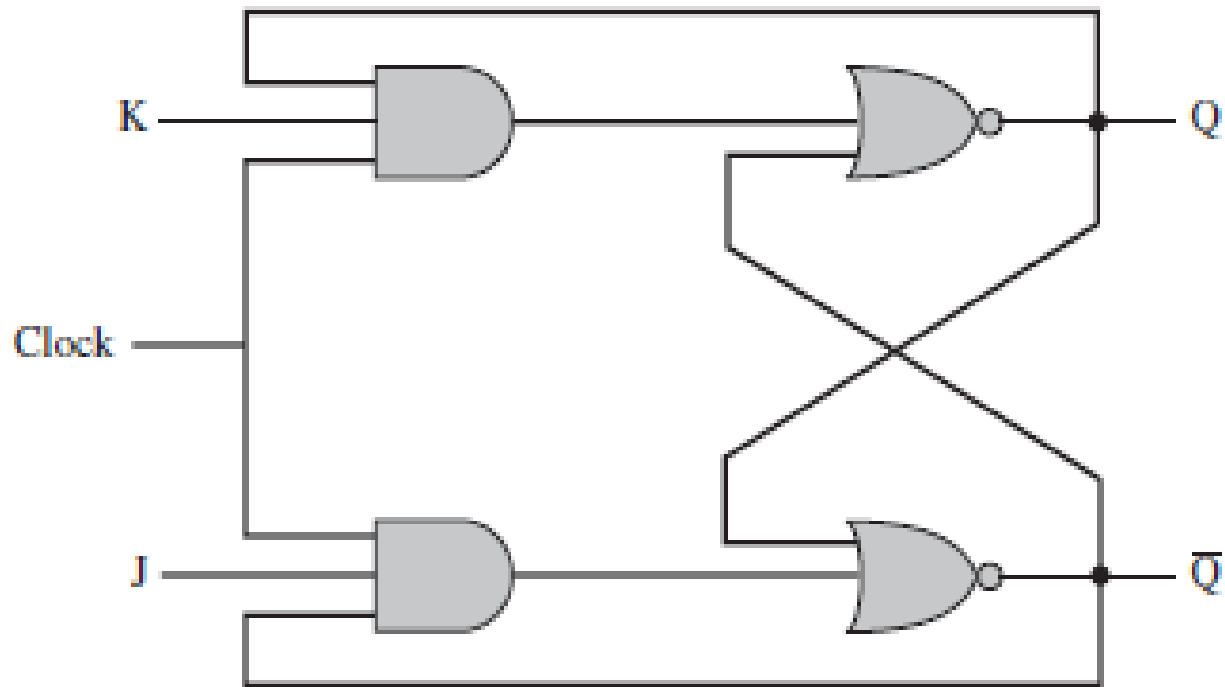


Figure 20.26 J-K Flip Flop

Osnovni tipovi flip flopova

| Name | Graphical Symbol | Truth Table | | | | | | | | | | | | | | | |
|------|------------------|---|---|-----------|-----------|---|---|-------|---|---|---|---|---|---|---|---|------------------|
| S-R | | <table border="1"> <thead> <tr> <th>S</th> <th>R</th> <th>Q_{n+1}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q_n</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>-</td> </tr> </tbody> </table> | S | R | Q_{n+1} | 0 | 0 | Q_n | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | - |
| S | R | Q_{n+1} | | | | | | | | | | | | | | | |
| 0 | 0 | Q_n | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | |
| 1 | 1 | - | | | | | | | | | | | | | | | |
| J-K | | <table border="1"> <thead> <tr> <th>J</th> <th>K</th> <th>Q_{n+1}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q_n</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>$\overline{Q_n}$</td> </tr> </tbody> </table> | J | K | Q_{n+1} | 0 | 0 | Q_n | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | $\overline{Q_n}$ |
| J | K | Q_{n+1} | | | | | | | | | | | | | | | |
| 0 | 0 | Q_n | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | |
| 1 | 1 | $\overline{Q_n}$ | | | | | | | | | | | | | | | |
| D | | <table border="1"> <thead> <tr> <th>D</th> <th>Q_{n+1}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table> | D | Q_{n+1} | 0 | 0 | 1 | 1 | | | | | | | | | |
| D | Q_{n+1} | | | | | | | | | | | | | | | | |
| 0 | 0 | | | | | | | | | | | | | | | | |
| 1 | 1 | | | | | | | | | | | | | | | | |

Figure 20.27 Basic Flip-Flops

Registri

- Registar se koristi unutar procesora da cuva jedan ili vise bitova podataka.
- Dva osnovna tipa su paralelni I pomeracki registri.

Paralelni registri

- Paralelni registri se sastoje od skupa 1-bitnih memorija koje se mogu istovremeno citati i u njih upisavati podaci.
- Koriste se za skladištenje podataka.

Paralelni registri

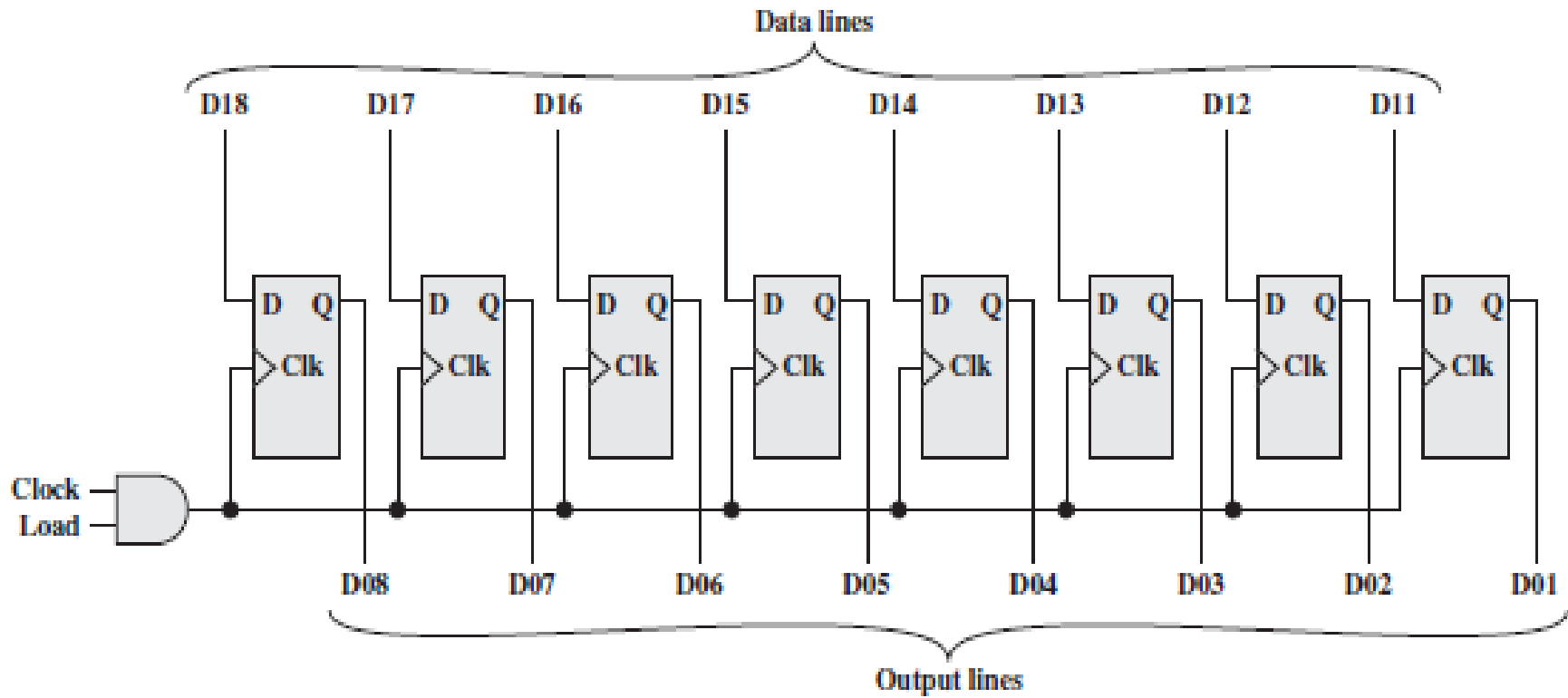


Figure 20.28 8-Bit Parallel Register

Pomeracki registri

- Pomeracki registar prihvata I serijski premesta informaciju.
- Primer 5-bitni pomeracki registar

Pomeracki registri

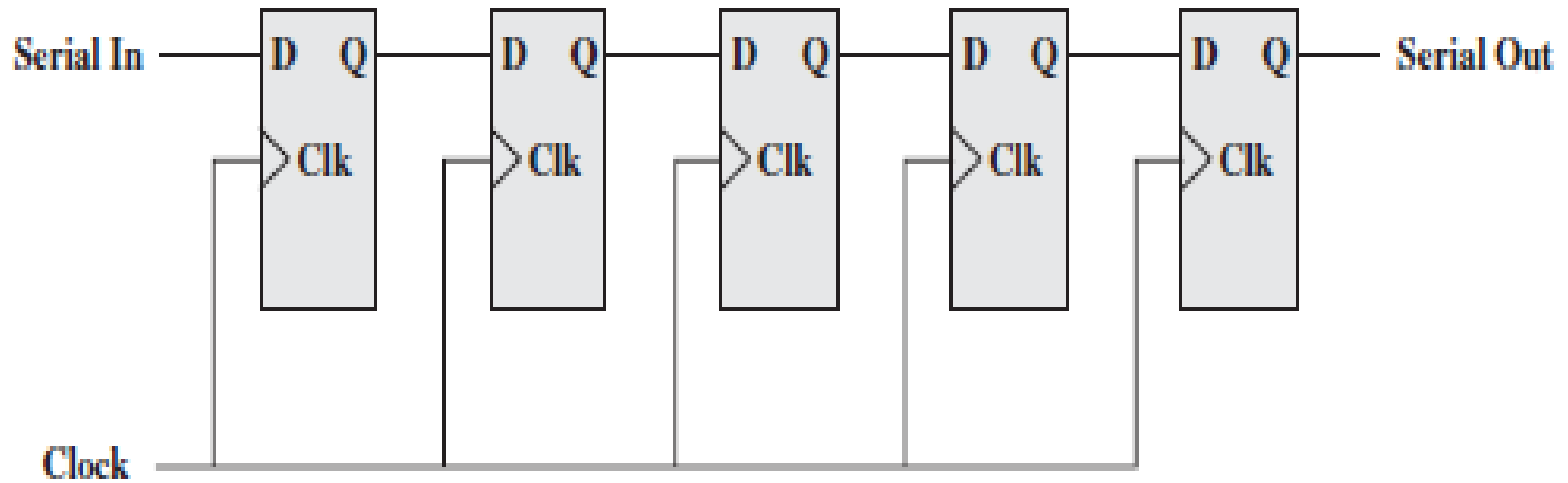


Figure 20.29 5-Bit Shift Register

Pomeracki registri

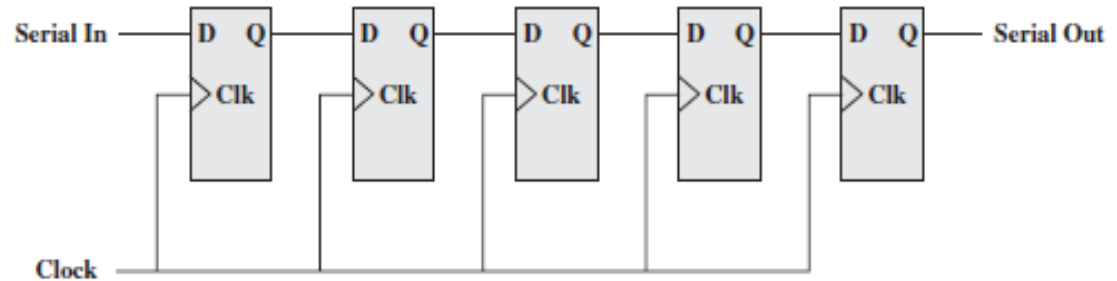


Figure 20.29 5-Bit Shift Register

- Podaci se unose samo u kranji levi flip flop.
- Svakim impulsom generatora takta, bit se pomera za jedno mesto u desno.
- Krajnji desni bit je izlaz.

Brojaci

- Brojaci su registri čija se vrednost lako povećava za jedan.
- Sastoje se od n flip flopova.
- Kada vrednost brojaca dostigne maksimalnu vrednost, vraća se na nulu.
- Brojaci se projektuju kao asinhroni i sinhroni.

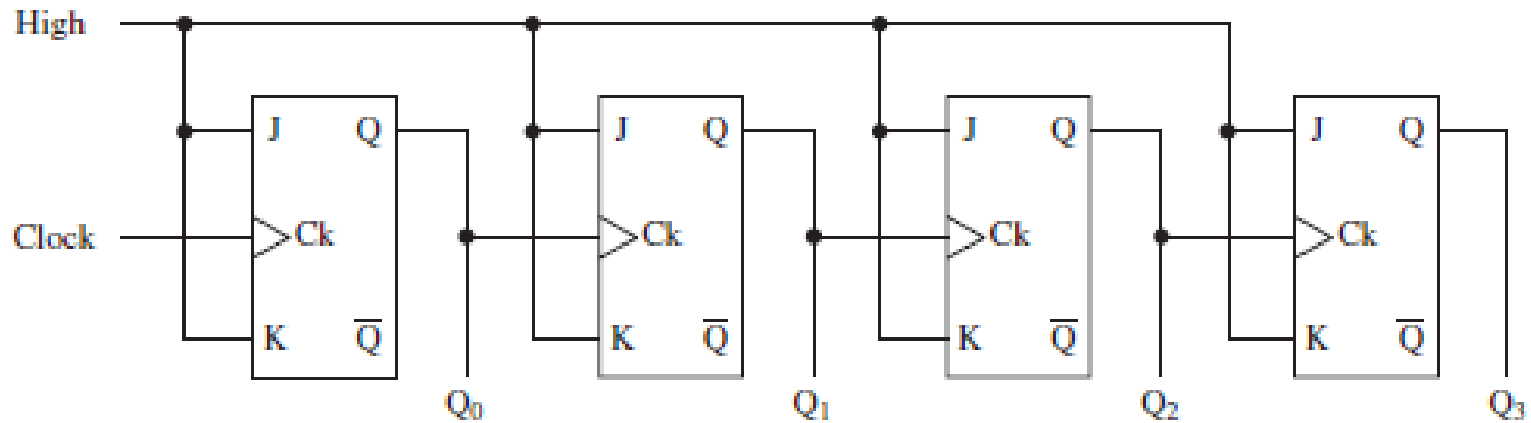
Brojaci

- Asinhroni brojaci su spori, izlaz iz jednog flip flopa inicira promenu stanja drugog flip flopa.
- Kod sinhronog brojaca svi flip flopovi istovremeno menjaju stanja.
- Sinhroni je mnogo brzi I on se koristi kod procesora.

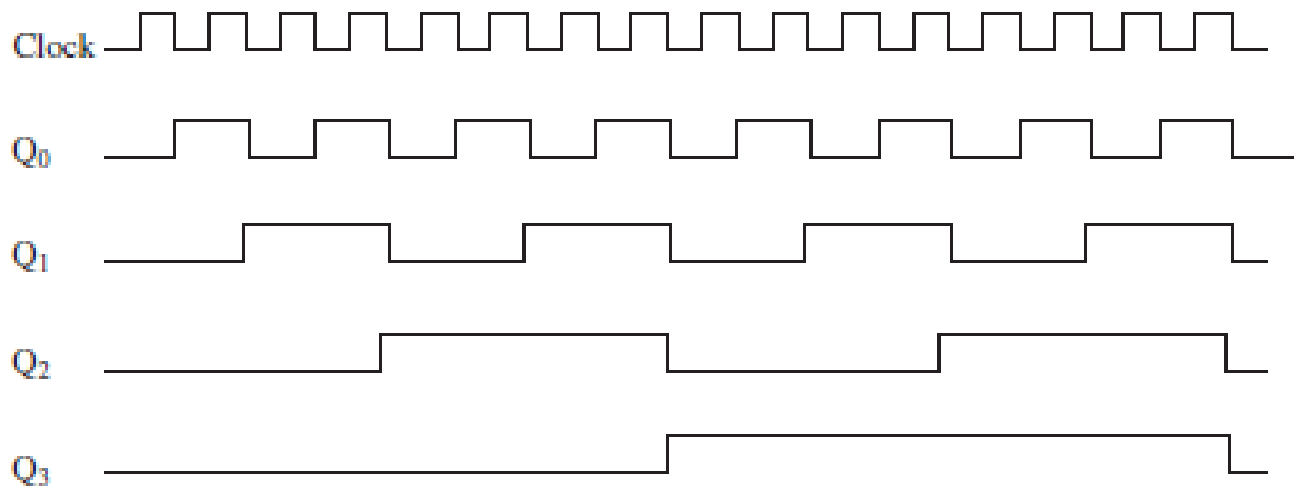
Asinhroni brojaci

- Primer 4-bitnog brojaca sa JK flip flopom
- Vremenski signal je idealan I ne ilustruje kasnjenje koje se javlja tokom prostiranja signala kroz flip flop.

Asinhroni brojaci



(a) Sequential circuit



(b) Timing diagram

Figure 20.30 Ripple Counter

3-bitni sinhroni brojac

- Tri JK flip flopa
- Nekomplementarni izlazi oznaceni sa A, B i C
- C bit najmanjeg znacaja
-

Tablica istinitosti JK flip flopa uticaj JK na izlaze

| J | K | Q_{n+1} |
|----------|----------|-----------------------------|
| 0 | 0 | Q_n |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | $\overline{Q_{n+1}}$ |

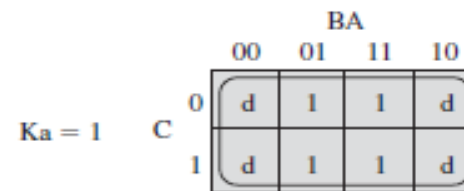
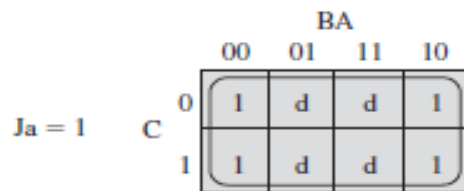
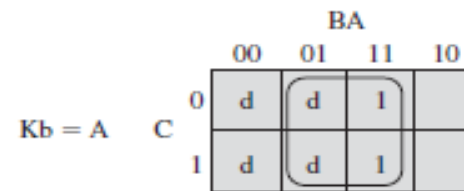
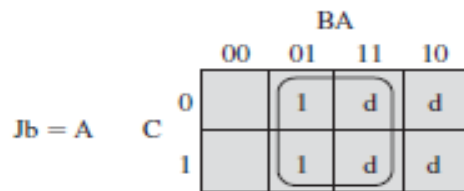
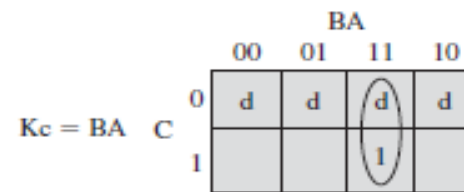
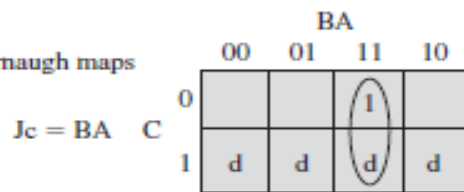
Tablica istinitosti kada su ulazi I sadasnji izlaz poznati, tabela pobude

| Q_n | J | K | Q_{n+1} |
|-------|---|---|-----------|
| 0 | 0 | d | 0 |
| 0 | 1 | d | 1 |
| 1 | d | 1 | 0 |
| 1 | d | 0 | 1 |

| C | B | A | Jc | Kc | Jb | Kb | Ja | Ka |
|---|---|---|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | d | 0 | d | 1 | d |
| 0 | 0 | 1 | 0 | d | 1 | d | d | 1 |
| 0 | 1 | 0 | 0 | d | d | 0 | 1 | d |
| 0 | 1 | 1 | 1 | d | d | 1 | d | 1 |
| 1 | 0 | 0 | d | 0 | 0 | d | 1 | d |
| 1 | 0 | 1 | d | 0 | 1 | d | d | 1 |
| 1 | 1 | 0 | d | 0 | d | 0 | 1 | d |
| 1 | 1 | 1 | d | 1 | d | 1 | d | 1 |

(a) Truth table

(b) Karnaugh maps



(c) Logic diagram

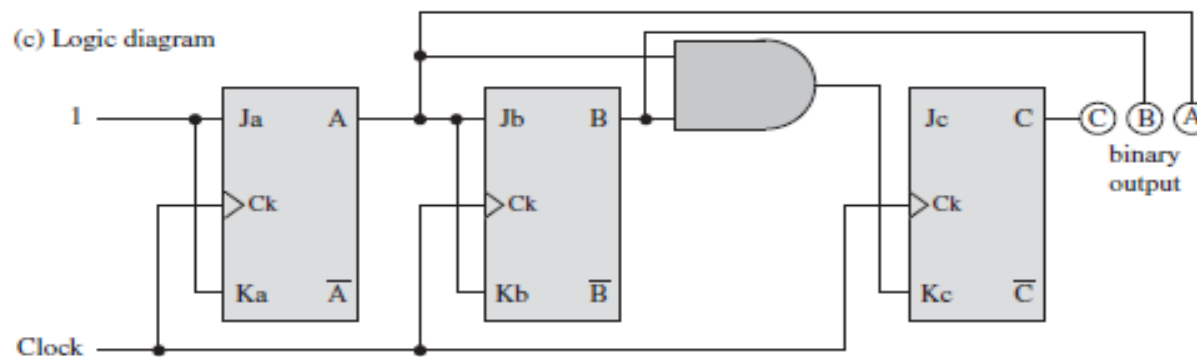


Figure 20.31 Design of a Synchronous Counter

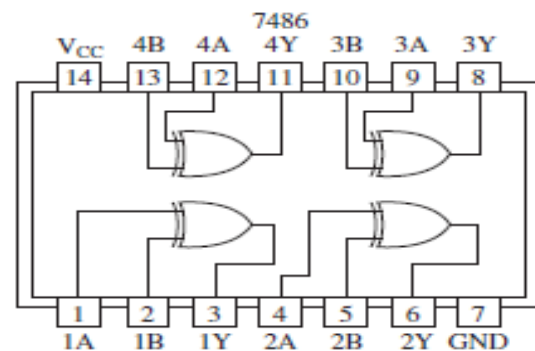
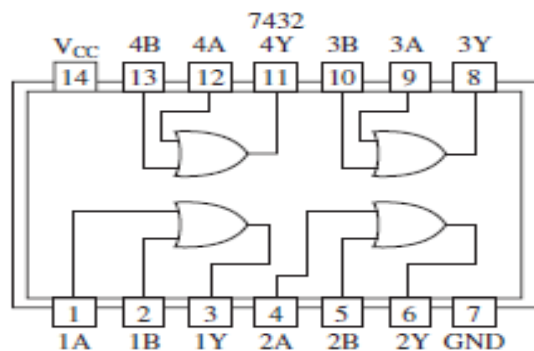
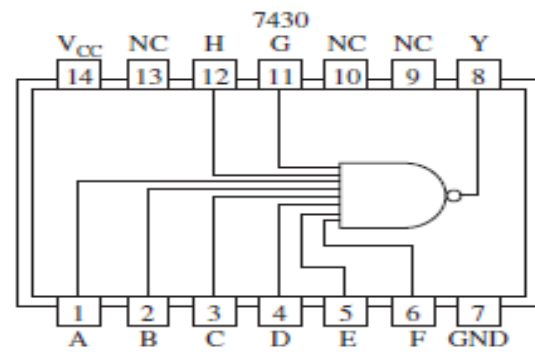
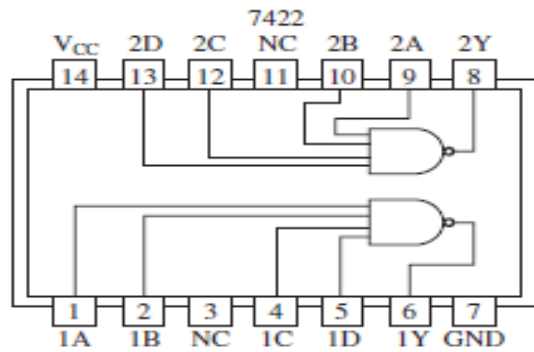
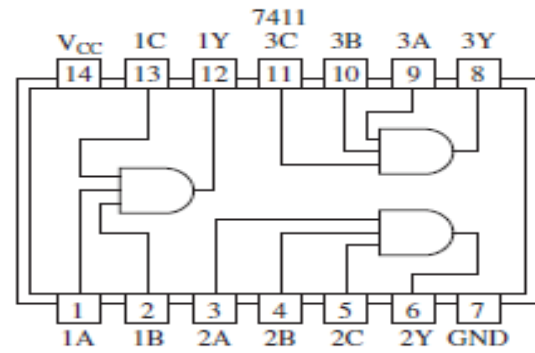
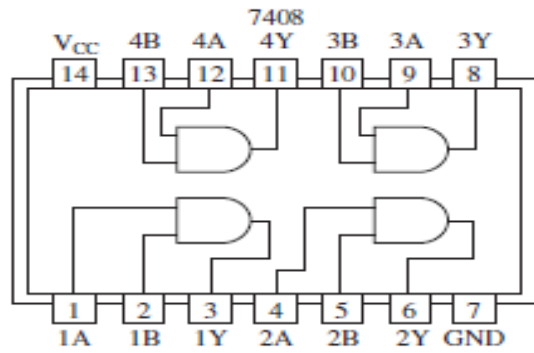
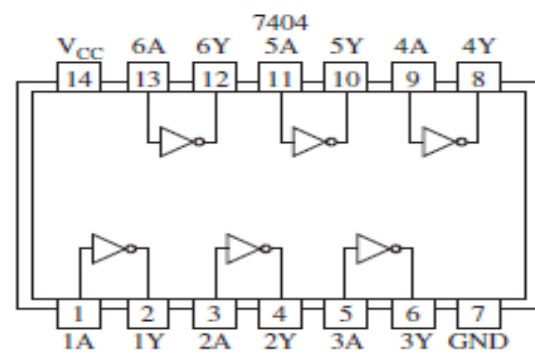
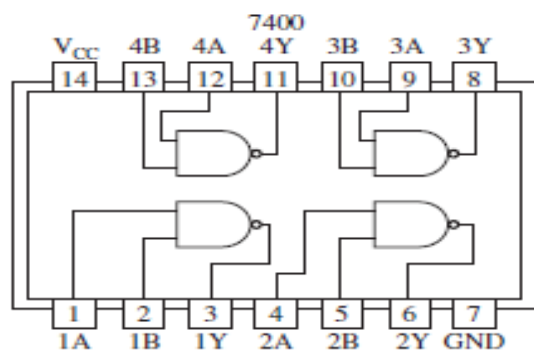


Figure 20.32 Some SSI Chips. Pin layouts from *The TTL Data Book for Design Engineers*, copyright © 1976 Texas Instrument Incorporated.

Programabilni logicki uredjaji

- Prvobitna integrirana kola sa niskim stepenom integracije SSI, deset logickih kola na jednom cipu.
- Povecani nivo integracije omogucio je povecanje broja logickih kola na cipu.
- Smanjena cena cipova.

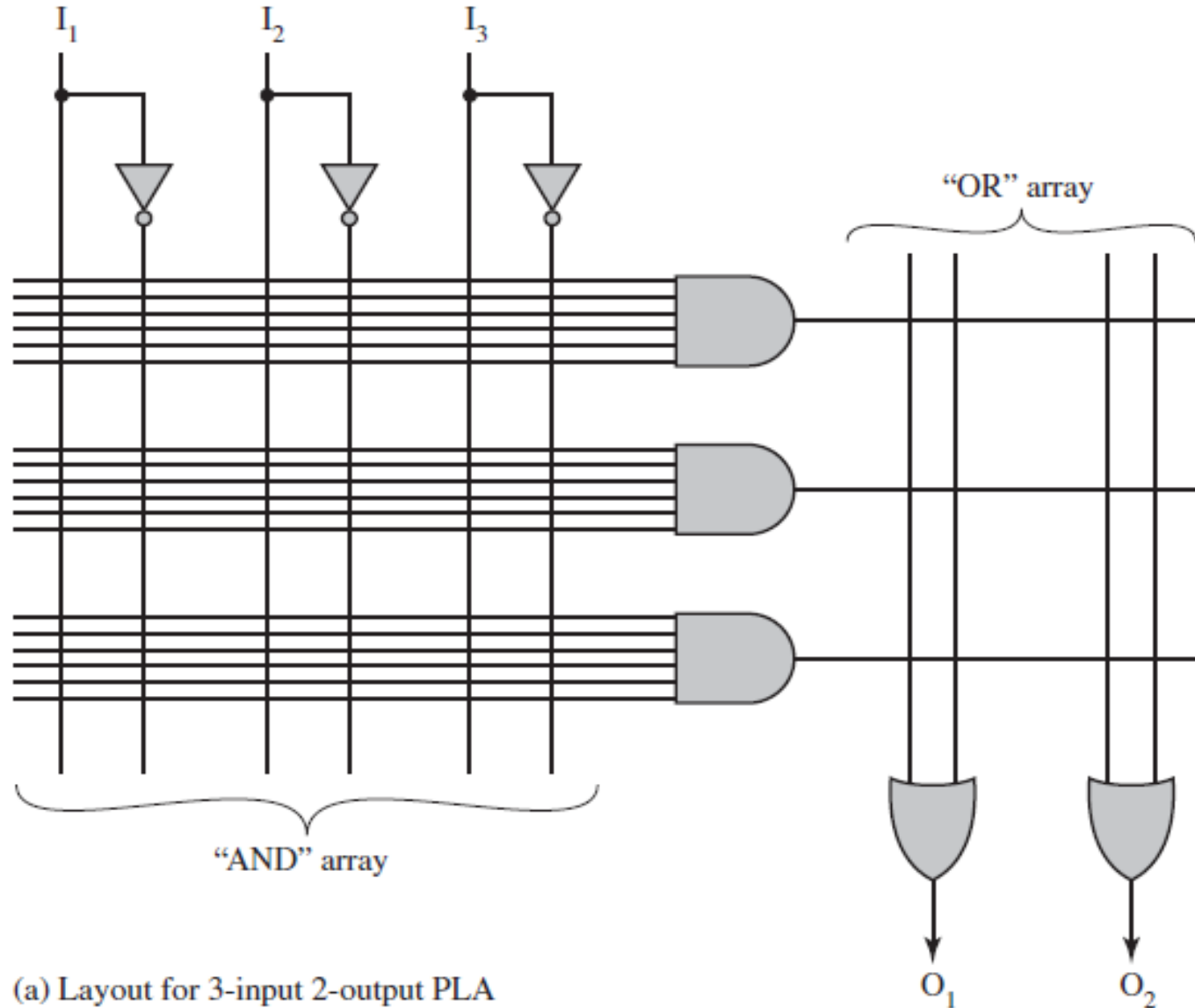
Programabilni logicki uredjaji

- Nedostatak je da za svaku pojedinačnu logicku funkciju ili skup funkcija mora da se projektuje poseban cip sa logickim kolima i vezama izmedju njih.
- Veliki troskovi projektovanja.
- Cipovi opste namene koji se mogu prilagoditi specificnim namenama.
- Programabilni logicki uredjaj(Programmable Logic Device), PLD

Programabilno logicko polje PLA

- PLA zasniva se na cinjenici da se bilo koja tablica istinitosti, Bulova funkcija, moze predstaviti u obliku sume proizvoda.
- PLA se sastoji od NE, I i ILI logickih kola pravilno spakovanih na cipu.
- Svi ulazi cipa se propustaju kroz NE kolo, tako da su svi ulazi I komplementi ulaza dostupni I kolima.
- Izlaz svakog I kola dostupan je svakom ILI kolu, a izlazi ILI kola su izlazi cipa.
- Odgovarajucim vezama moguće je implementirati bilo koju proizvoljnu sumu proizvoda.

PLA sa tri ulaza, 8 logickih kola i 2 izlaza



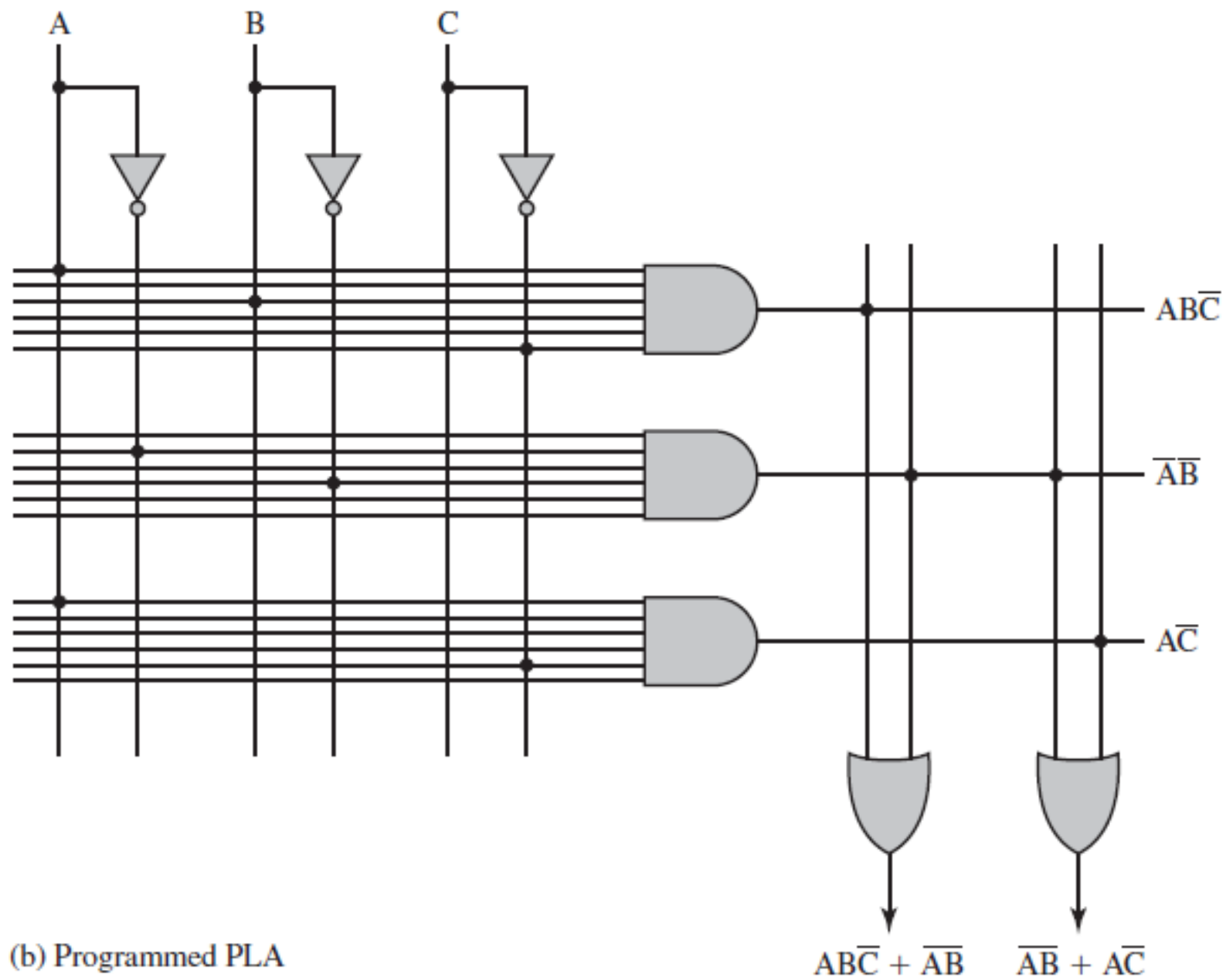
(a) Layout for 3-input 2-output PLA

PLA sa tri ulaza, 8 logickih kola I 2 izlaza

- Na levoj strani je programabilno I podrucje
- Programira se uspostavljanjem veze izmedju bilo kog ulaza PLA ili njegove negacije sa ulazom u I kolo povezivanjem linija u tacki preseka.
- Sa desne strane je programabilno ILI podrucje.
- Povezivanje izlaza I kola sa ulazima ILI kola.

PLA sa tri ulaza, 8 logickih kola I 2 izlaza

- Vecina PLA sastoji se od nekoliko stotina logickih kola, 15-25 ulaza I 5-15 izlaza
- Prvi nacin programiranja je da se veze prave od topljivih spojeva na presecnim tackama.
- Pregorevanjem spojeva, uklanjaju se nezeljene veze.
- Ova vrsta PLA zove se lokalno programabilno logicko polje.
- Drugi nacin je da se veze uspostavljaju u procesu proizvodnje cipova koriscenjem odgovarajuce maske.



(b) Programmed PLA

Figure 20.33 An Example of a Programmable Logic Array

Lokalno programabilna matrica logickih kola FPGA

- PLA je primer jednostavnog PLD, eng. SPLD.
- Uredjaji velikog kapaciteta, integracija vecceg broja SPDL na jednom cipu I da se obezbedi medjupovezivanje blokova.
- Mnogi komercijalni PLD proizvodi danas postoje na trzistu sa takvom osnovnom strukturom I nazivaju se slozeni PLD uredjaji CPLD.
- Najpoznatiji je FPGA.

Lokalno programabilna matrica logickih kola FPGA

- FPGA se sastoji od matrice neangazovanih elemenata kola koji se zovu logicki blokovi i resursa za medjupovezivanje.
- Na sledecem slajdu prikzana je tipicna arhitektura FPGA uredjaja.

Tipicna arhitektura FPGA

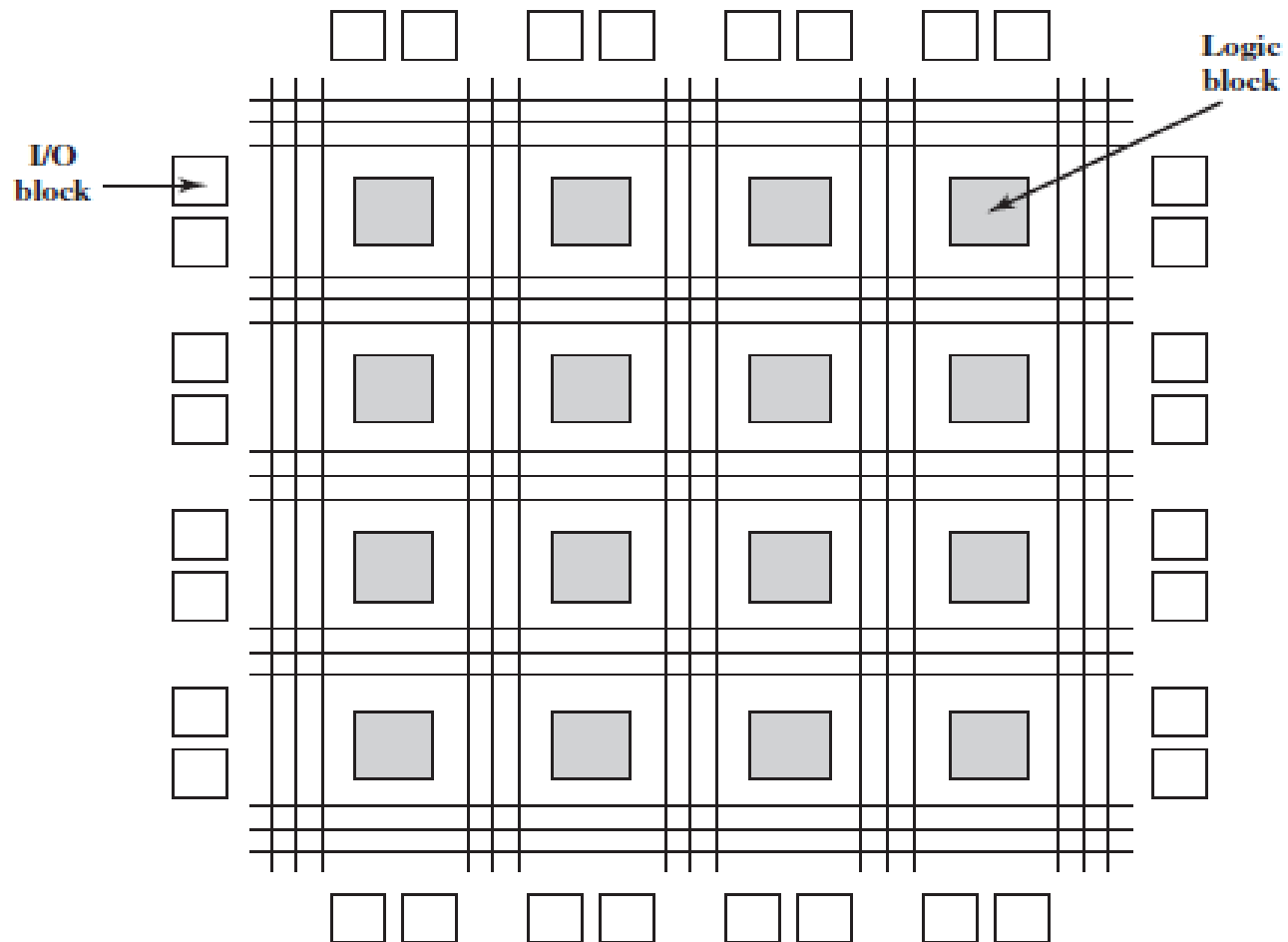
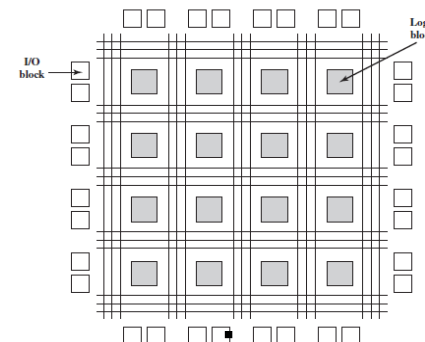


Figure 20.34 Structure of an FPGA

Tipicna arhitektura FPGA



- Logicki blok, mesta na kojima se izvrsavaju izracunavanja korisnikovih kola.
- U/I blok, povezuju U/I pinove na kolima cipa
- Medjupovezivanje, putanje signala za uspostavljanje veza izmedju U/I blokova i logickih blokova.

Tipicna arhitektura FPGA

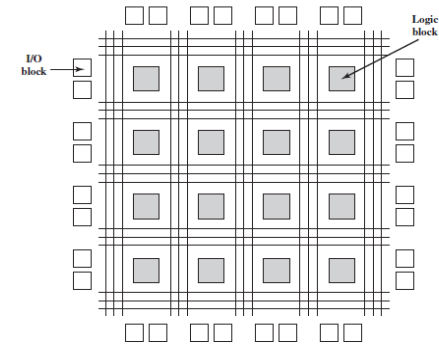


Figure 20.34 Structure of an FPGA

- Logicki blok može biti kombinaciono ili sekvencijalno kolo.
- Programiranje logickog bloka vrši se preuzimanjem sadržaja tablice istinitosti za logicku funkciju.

Jednostavan FPGA logicki blok

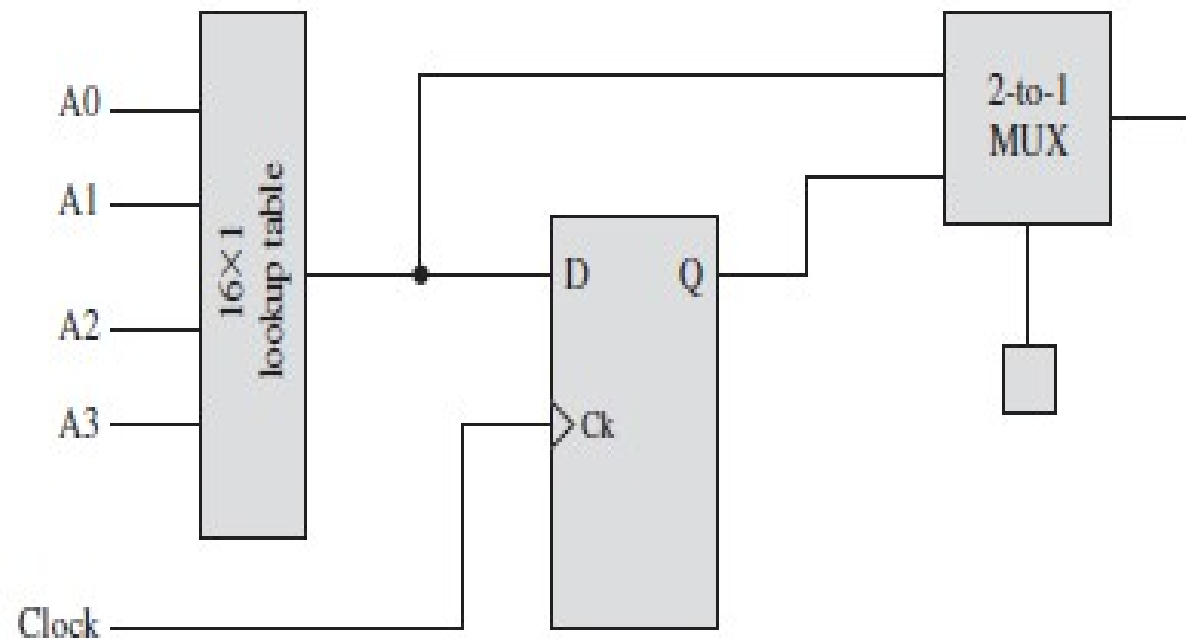


Figure 20.35 A Simple FPGA Logic Block