

prof. dr.sc. A. Hamzić

OSNOVE POLUVODIČKE DIGITALNE ELEKTRONIKE

Fizički odsjek, Prirodoslovno-matematički fakultet, Zagreb,
kolegij: MIKROELEKTRONIKA
siječanj, 2010.

1 OSNOVE POLUVODIČKE DIGITALNE ELEKTRONIKE

Osnovni principi rada digitalnih elektroničkih sklopova su zapravo jednostavniji od onih u analognoj elektronici. Razlog za to je što se u digitalnoj elektronici sklopovi u velikoj mjeri koriste isključivo kao sklopke, kojima se kontrolirano upravlja. To drugim riječima znači da elektronički elementi, koji se koriste za procesiranje signala (koji predstavljaju numeričke podatke), mogu biti u dva kontrolirana, ali različita stanja. Pri tome se koristi binarni sustav prikaza, sklopovi vrše samo nekoliko jednostavnih operacija, no one se izvode veliki broj puta.

1.1 Binarni sustav

Binarni sustav sastoji se od samo dva stanja, broja ili nivoa signala, koji su međusobno različiti. To može biti: *istinito* – *neistinito*, *visoko* – *nisko*, *1* – *0*, *da* – *ne*. U binarnom sustavu koristi se prikaz s bazom broja 2. Najveći dekadski broj N koji se može prikazati s n bitova je

$$N = 2^n - 1 \quad (1.1)$$

(*bit* = **binary digit** = 0 ili 1; *byte* = grupa od 8 (16, 32...) bita; *word* = grupa bitova koji zajednički imaju neko značenje)

Binarni prikaz npr. broja 73 je:

$$73 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \Leftrightarrow 1001001 \quad (1.2)$$

Binarna znamenka uz najvišu potenciju broja 2 označava se kao MSD (*most-significant digit*), a uz najnižu potenciju kao LSD (*least-significant digit*).

Binarna točka razdvaja pozitivne i negativne potencije broja 2:

$$101.101 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 5 + 0.5 + 0.125 = 5.625 \quad (1.3)$$

Negativni brojevi se prikazuju dodavanjem bita za predznak (0 = pozitivni broj; 1 = negativni broj) ispred MSD u *word*-u, i umetanjem posebnog znaka:

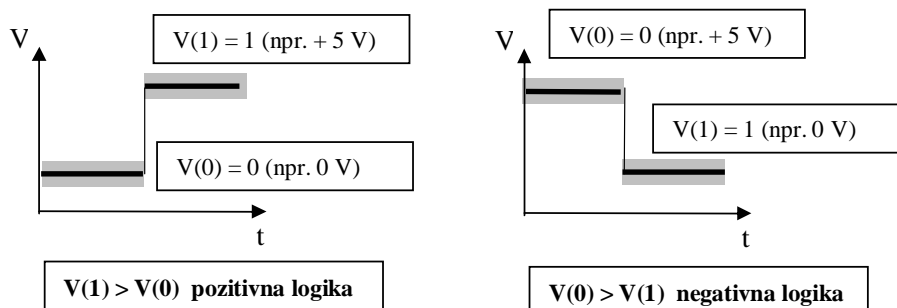
$$+ 73 \Leftrightarrow 0 \wedge 1001001 \quad (1.4)$$

$$- 73 \Leftrightarrow 1 \wedge 1001001 \quad (1.5)$$

Ako se za 2 logička stanja koriste naponski signali, tada se razlikuje:

- (a) **DCLL** (*DC logic level* ili istosmjerna razinska logika) – može biti pozitivna i negativna (slika 1.1)

Pri tome uvijek postoji određena neodređenost u iznosima napona (npr. $5 \text{ V} = (5 \pm 0.2) \text{ V}$), ali to ne predstavlja poteškoću sve dok je razlika između napona koji odgovaraju stanjima $V(1)$ i $V(0)$ veća od suma.

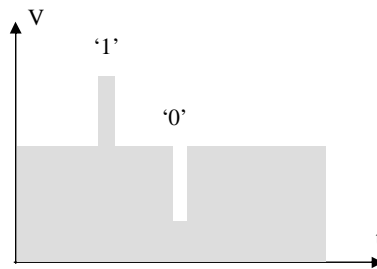


Slika 1.1: Nivoi napona u DCLL logici

- (b) **impulsna** (ili dinamička) logika – bit se manifestira kao postojanje ili odsustvo impulsa (slika 1.2).

1.1.1 Booleova algebra

George Boole je u XIX stoljeću razvio algebru sustava koji se sastoji od samo dva moguća stanja. Ona se praktički uopće nije koristila sve do



Slika 1.2: Nivoi napona u impulsnoj logici

1938. godine, kad je Shannon pokazao da se koncepti Booleove algebre mogu koristiti u teoriji elektroničkog prekidanja.

U Booleovoj algebri simbol $+$ znači ILI (OR), a simbol \cdot znači I (AND):

$A + B = Y$ (Ako je A istinito **ILI** ako je B istinito tada je Y istinito.)

$A \cdot B = Y$ (Ako je A istinito **I** ako je B istinito tada je Y istinito.)

Za potpunu primjenu Booleove algebre potrebne su samo 3 osnovne logičke funkcije (AND, OR i NOT), tj. potrebna su samo troja logička vrata. Kompleksniji izrazi ostvaruju se kombinacijom osnovnih vratiju.

Booleovi identiteti su:

$$A + A = A$$

$$A \cdot A = A \text{ (postulat)}$$

$$A + 1 = A$$

$$A \cdot 1 = A \text{ (postulat)}$$

$$A + 0 = A \text{ (postulat)}$$

$$A \cdot 0 = 0$$

$$A + \bar{A} = 1$$

$$A \cdot \bar{A} = 0$$

$$\bar{\bar{A}} = A$$

$$A + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A$$

$$A \cdot (A + B) = A \cdot A + A \cdot B = A + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A$$

$$A \cdot (\bar{A} + B) = A \cdot \bar{A} + A \cdot B = 0 + A \cdot B = A \cdot B$$

$$A + \bar{A} \cdot B = A \cdot (B + 1) + \bar{A} \cdot B = B \cdot (A + \bar{A}) + A = B + A = A + B$$

$$\bar{A} + A \cdot B = \bar{A} \cdot (B + 1) + A \cdot B = B \cdot (A + \bar{A}) + \bar{A} = \bar{A} + B$$

$$\bar{A} + A \cdot \bar{B} = \bar{A} \cdot (\bar{B} + 1) + A \cdot \bar{B} = \bar{B} \cdot (\bar{A} + A) + \bar{A} = \bar{A} + \bar{B}$$

$$\overline{A + B} = \bar{A} \cdot \bar{B} \text{ (de Morganov zakon)}$$

$$\overline{A \cdot B} = \bar{A} + \bar{B} \text{ (de Morganov zakon)}$$

Dokaz de Morganovih zakona:

Izjava *Izlaz je 1 ako su svi ulazi 1* je logički jednaka izjavi *Ako je (barem) jedan ulaz 0 tada je i izlaz 0*. Tada slijedi: .

$$A \cdot B = \overline{\bar{A} + \bar{B}} \tag{1.6}$$

a nakon komplementiranja

$$\overline{A \cdot B} = \bar{A} + \bar{B} \tag{1.7}$$

Ako se pak "zamijeni" logika, tj. $A \rightarrow \bar{A}$ te $\bar{A} \rightarrow A$, te komplementira, odmah slijedi $\overline{\bar{A} + \bar{B}} = \bar{A} \cdot \bar{B}$.

1.2 Osnovna logička vrata

1.2.1 OR (ILI) vrata

OR vrata imaju 2 (ili više ulaza) i jedan izlaz, te zadovoljavaju definiciju: *izlaz je u stanju 1 ako je jedan ili više ulaza u stanju 1*.

$$Y = A + B + \dots + N \tag{1.8}$$

koja se čita: *Y je A ili B ili ... ili N*.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

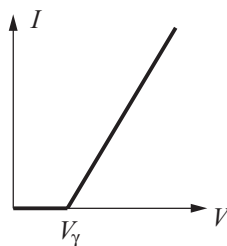
Tablica 1.1: Tablica istine za OR sklop

Najjednostavnija izvedba OR sklopa je u DL (diodnoj) logici, pri čemu se pretpostavlja idealizirana $I - V$ karakteristika diode (za $V < V_\gamma$, dioda

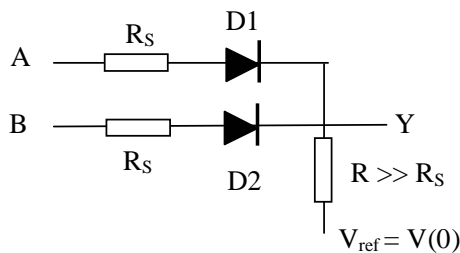


Slika 1.3: Simbol OR vratiju

ne vodi struju; za $V > V_\gamma$, dioda provodi struju, a njezin unutrašnji otpor $R_f \leq 10 \Omega$; pri tome je za Ge diodu $V_\gamma = 0.2 \text{ V}$, odnosno 0.7 V za Si diodu).



Slika 1.4: Idealizirana I-V karakteristika diode



Slika 1.5: Izvedba OR vratiju u DL logici

Izvedba OR vratiju za pozitivnu logiku je prikazana na slici 1.5 (uz $R \simeq 5 \text{ k}\Omega$, te unutrašnji otpor izvora $R_S \simeq 100 \Omega$):

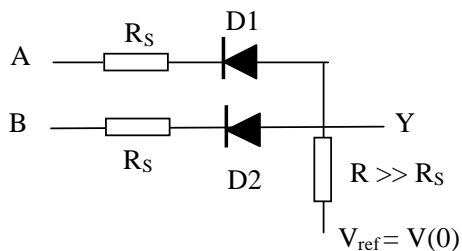
- (a) Ulaz A (ili B , ili oba) je u stanju $V(0)$: dioda ne vodi struju i izlaz Y je u stanju $V(0)$ (tj. na izlazu je referentni napon).
- (b) Jedan od ulaza (npr. A) je u stanju $V(1)$: izlazni napon je (uz $R \gg R_S, R_f$)

$$v_i = V_{ref} + \frac{V_i - V_{ref} - V_\gamma}{R_S + R_f + R} \cdot R \simeq V(0) + V(1) - V_\gamma = V(1) \quad (1.9)$$

- (c) Ako je više ulaza u stanju $V(1)$, izlazni napon je $v_i = V_i(1) - V_\gamma$.

No, ulazni napone $V_i(1)$ se mogu razlikovati (unutar dozvoljene granice šuma), pa će uvijek onaj ulaz koji je najveći po iznosu dati na izlazu napon $v_i = V_i(1) - V_\gamma$ i tako 'blokirati' ostale diode.

Izvedba OR vratiju u negativnoj logici prikazana je na slici 1.6:



Slika 1.6: Izvedba OR vratiju u negativnoj DL logici

Ako je na ulazu $V(0) > V(1)$, dioda ne vodi i $Y = V(0)$. Za $V(1)$ na ulazu, izlazni napon je

$$v_i = V(0) - \frac{V(0) - V(1) - V_\gamma}{R_s + R_f + R} \cdot R \simeq V(1) \quad (1.10)$$

1.2.2 AND (I) vrata

AND vrata imaju 2 (ili više ulaza) i jedan izlaz i zadovoljavaju definiciju: *izlaz je u stanju 1 ako i samo ako su svi ulazi u stanju 1.*

$$Y = A \cdot B \cdot \dots \cdot N \quad (1.11)$$

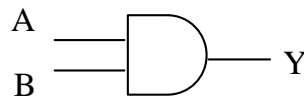
koja se čita: *Y je A i B i ... i N.*

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

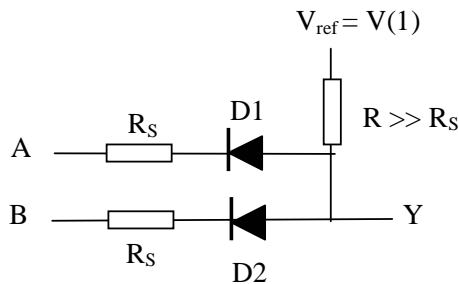
Tablica 1.2: Tablica istine za AND sklop

Na slici 1.8 je DL izvedba AND vratiju u pozitivnoj logici.

(a) Za $V(1)$ na ulazu, diode ne vode i $Y = V(1)$;



Slika 1.7: Simbol AND vratiju



Slika 1.8: Izvedba AND vratiju u pozitivnoj DL logici

(b) Ako je na ulazu $V(0)$, izlazni napon je $V(0)$, jer vrijedi:

$$v_i = V(1) - \frac{V(0) - V(1) - V_\gamma}{R_S + R_f + R} \cdot R \simeq V(0) - V_\gamma = V(0) \quad (1.12)$$

(c) Ako su oba ulaza u stanju $V(0)$ (koji se razlikuju do na iznos šuma) voditi će ona dioda na kojoj je najniži napon $V(0)$ i tako blokirati ostale diode.

Za AND logička vrata u negativnoj logici je obrnut polaritet dioda.

1.2.3 de Morganov teorem

Usporedbom izvedbe i rada krugova AND i OR očito je da vrijedi jednakost:

AND (pozitivna logika) = OR (negativna logika) (i obrnuto)

Ovo svojstvo komplementarnosti je općenito i ne ovisi o tipu korištene logike (DL ili neka druga izvedba) te čini tzv. *de Morganov teorem*: invertiranjem svih ulaza i izlaza logicka operacija AND postaje OR (i obrnuto).

Komplementiranje AND u pozitivnoj logici ($Y = A \cdot B$) daje (uz primjenu de Morganovog zakona) $\overline{Y} = \overline{A \cdot B} = \overline{A} + \overline{B}$. No komplementiranje svih ulaza i izlaza znači da pozitivna logika prelazi u negativnu, pa je dobiveni izraz OR u negativnoj logici. De Morganov teorem se može jednostavno dokazati i odgovarajućim tablicama istine.



Slika 1.9: de Morganov teorem

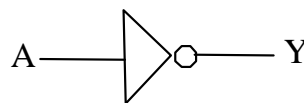
1.2.4 NOT (NE) vrata

NOT vrata su jedina logička vrata sa samo jednim ulazom i izlazom - sklop je inverter: Y je ne-A ili Y je komplement A:

$$Y = \bar{A} \tag{1.13}$$

A	Y
0	1
1	0

Tablica 1.3: Tablica istine za NOT sklop



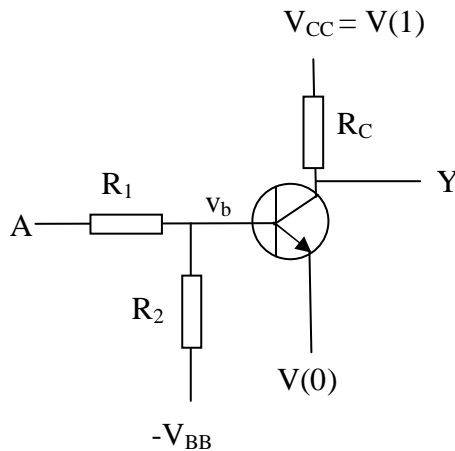
Slika 1.10: Simbol NOT vratiju

Izvedba NOT vratiju zahtijeva postojanje aktivnog elementa (tranzistora). Najjednostavnija izvedba za pozitivnu logiku s BJT tranzistorom je u tzv. RTL (Resistor -Tranzistor) logici – slika 1.11.

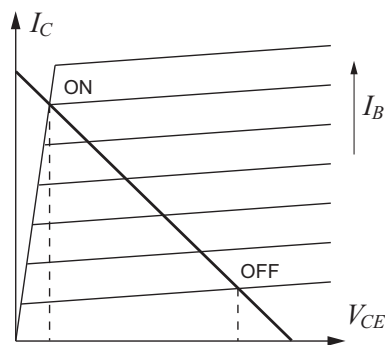
Vrijedi $i = (v_u + V_{BB}) / (R_1 + R_2)$ i $v_b = i \cdot R_2 - V_{BB}$. Koristeći karakteristike BJT tranzistora (slika 1.12)je:

- (a) Za $A = V(0) = 0 \text{ V}$ je $v_b = V_{BB} \cdot (R_2 / (R_1 + R_2) - 1) < 0$, tj. tranzistor je zatvoren (*OFF*), i $V_{CE} \simeq V_{CC} = V(1)$;
- (b) Za $A = V(1) = 5 \text{ V}$ slijedi $v_b > 0$ (treba biti dovoljno velik da izazove struju $I_B = I_{Bsat}$, pa je tranzistor u području saturacije (*ON*), a $V_{CE} = V(0) \simeq 0$.

Ovakva analiza NOT vratiju vrijedi za DCLL logiku. U impulsnoj logici ulazni impuls npr. logičke jedinice (koji je pozitivni pravokutni signal s



Slika 1.11: Izvedba NOT vratiju u pozitivnoj RTL logici



Slika 1.12: I-V karakteristika BJT

određenim vremenom trajanja) dolazi atenuiran na bazu tranzistora (preko djelitelja napona R_1 i R_2). No zbog postojanja međuelektrodnih kapaciteta, realni impuls koji dolazi na bazu nije pravokutan, već eksponencijalno raste ovisno o iznosu R_2C_2 konstante, pri čemu je kapacitet

$$C_2 = C_{BE} + C_{BC} \cdot (1 - A) = C_{\pi} + C_{\mu}(1 - A) \quad (1.14)$$

To znači da bi zbog C_2 porast ulaznog napona mogao biti tako spor da napon na bazi ne dosegne napon $V(1)$, čime bi ulazna informacija mogla biti izgubljena. Veliko vrijeme porasta napona na djelitelju je zbog toga što sve frekvencije sinusoidalnog napona (koje su, po Fourierovoj analizi, komponente skoka ulaznog napona) nisu jednako atenuirane, već su više frekvencije jače atenuirane. Da se postigne jednoliko prigušenje svih frekvencija vrši se tzv. **kompensacija atenuatora** – slika 1.13, kod koje se

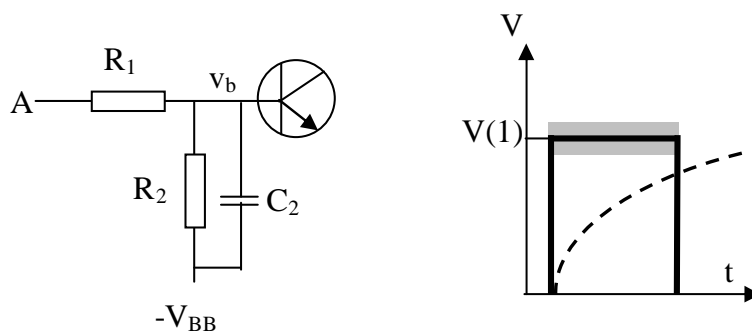
paralelno s otporom R_1 stavlja promjenljivi kapacitet C_1 takav da vrijedi:

$$R_1 : R_2 = \frac{1}{\omega C_1} : \frac{1}{\omega C_2} \quad (1.15)$$

odnosno

$$R_1 C_1 = R_2 C_2 \quad (1.16)$$

U tom slučaju je atenuacija svih frekvencija jednaka atenuaciji istosmjene komponente.



Slika 1.13: Kompenzacija atenuatora za NOT vrata

1.2.5 NOR (NILI) vrata

Logička vrata AND, OR i NOT (te njihove kombinacije) omogućuju realizaciju bilo kakvog logičkog kruga. Kombinacijom OR i NOT dobiva se NOR sklop.

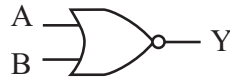
Za NOR vrata vrijedi

$$Y = \overline{A + B} \quad (1.17)$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

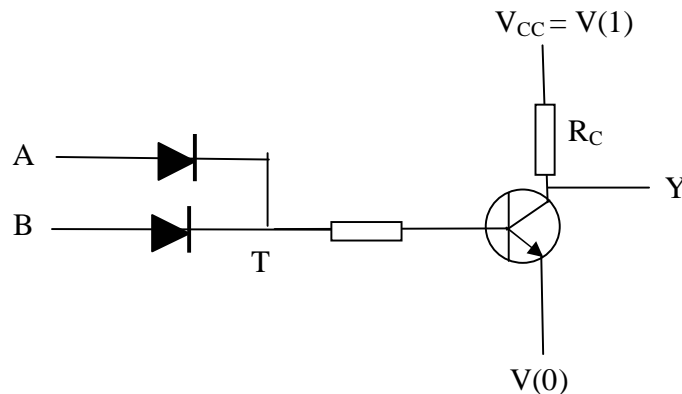
Tablica 1.4: Tablica istine za NOR sklop

Ako su A ili B ili oba na $V(1)$, prednapon baze tranzistora (slika 1.15) je veći od nule, i tranzistor je u zasićenju, tj. $Y = V_{CE} = 0 = V(0)$. Ako je



Slika 1.14: *Simbol NOR vratiju*

na ulazu $V(0)$, struja baze je nula i tranzistor je u prekidnom području, pa je $Y = V_{CE} = 1 = V(1)$.



Slika 1.15: *NOR vrata*

1.2.6 NAND (NI) vrata

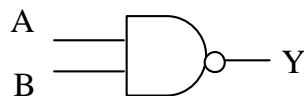
Kombinacijom AND i NOT dobivaju se NAND logička vrata (slika 1.17). Ako su svi ulazi na $V(1)$, diode ne vode, i u točki T je napon $V(1)$. Tranzistor je otvoren i $Y = V(0) = 0$. Ako je pak neki od ulaza na naponu $V(0)$, dioda vodi struju i u točki T je napon $V(0)$, što znači da je tranzistor zatvoren i $Y = V(1)$. Može se, međutim, dogoditi da je iznos napona u točki T veći od napona V_γ , pa tranzistor nije u prekidnom području (već u 'normalnom' režimu rada s radnom točkom između *ON* i *OFF* režima). Da se to izbjegne, između baze i točke T se stavlja jedna (ili čak i 2) dioda, čime se osigurava da je tranzistor zatvoren (slika 1.17).

Za NAND vrijedi

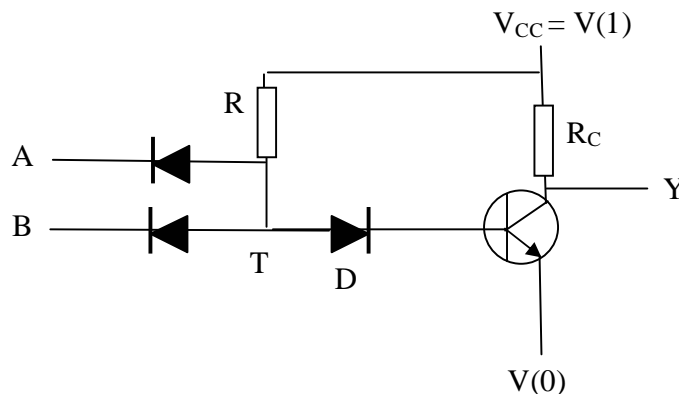
$$Y = \overline{A \cdot B} \quad (1.18)$$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Tablica 1.5: Tablica istine za NAND sklop



Slika 1.16: Simbol NAND vratiju

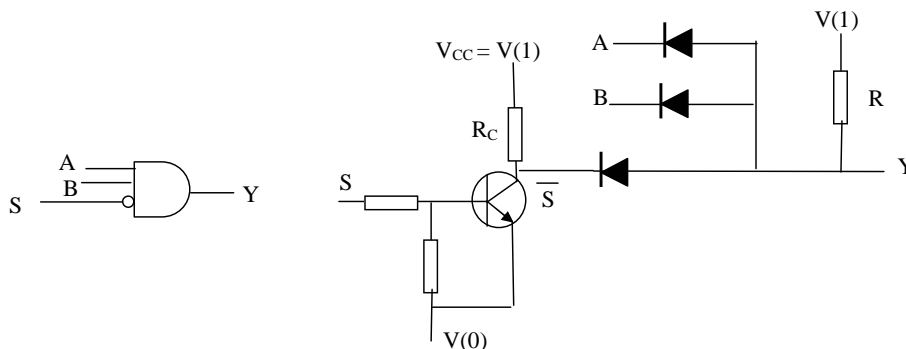


Slika 1.17: NAND vrata

1.2.7 Operacije zabrane (INHIBIT)

Stavljanjem NOT na jedan ulaz AND sklopa ostvaruje se operacija zabrane, tj. antikoincidencije. Sklop AND se, naime, u normalnom režimu rada može smatrati kao sklop za koincidenciju, jer je izlaz 1 kada su svi ulazi 1. U slučaju antikoincidencije je pak izlaz jednak 1 ako su svi ulazi 1 i ulaz $S = 0$; dakle $Y = A \cdot B \cdot \dots \cdot \bar{S}$

Bit S omogućuje da sklop AND izvrši svoju logiku samo ako je $S = 0$, dok $S = 1$ daje na izlazu $Y = 0$ neovisno o o ostalim ulaznim bitovima! Bit S se još naziva i *ENABLE* bit, a shema realizacije sklopa je prikazana na slici 1.18.



Slika 1.18: Vrata za antikoincidenciju: (a) simbol; (b) RTL izvedba

1.2.8 Exclusive – or (EX – OR) sklop

Prvi "viši" sklop koji se realizira pomoću do sada opisanih osnovnih logičkih vratiju je EX - OR. On zadovoljava definiciju: *Y je 1 ako je samo jedan od ulaza u stanju 1.*

$$Y = A \cdot \bar{B} + \bar{A} \cdot B = (A + B) \cdot \overline{A \cdot B} \quad (1.19)$$

$$= (A + B) \cdot (\bar{A} + \bar{B}) = \overline{A \cdot B} + \overline{\bar{A} \cdot \bar{B}} \quad (1.20)$$

$$= A \oplus B \quad (1.21)$$

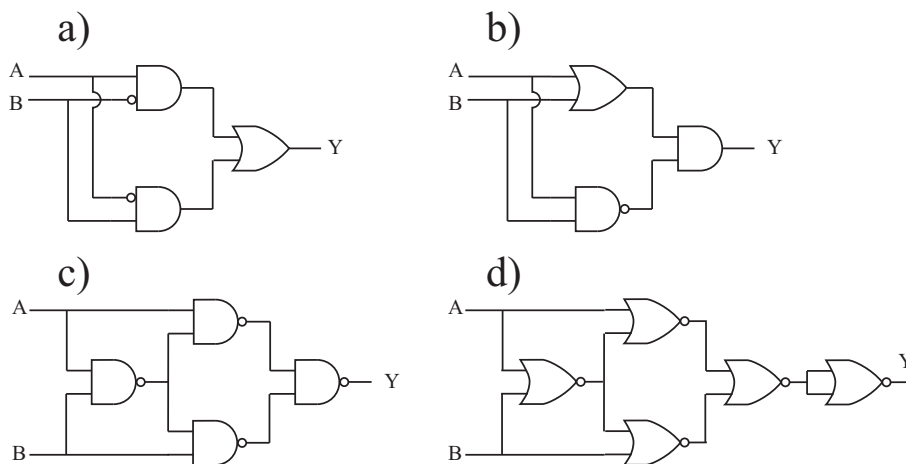
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Tablica 1.6: Tablica istine za EX-OR sklop



Slika 1.19: Simbol EX-OR sklopa

Izvedbe mogu biti različite i zavise o načinu pisanja logičke operacije (slika 1.20).



Slika 1.20: Neke moguće izvedbe EX-OR sklopa: (a) $Y = A \cdot \bar{B} + \bar{A} \cdot B$; (b) $Y = (A + B) \cdot \bar{A} \cdot \bar{B}$; (c) izvedba sa samo NAND vratima; (d) izvedba sa samo NOR vratima

1.2.9 Određivanje logičkog sklopa na temelju tablice istine

U ovom postupku se, radi jednostavnosti, uvijek razmatraju one manji broj onih linija koje imaju isti izlaz, a zatim se dobijeni izraz uredi što je više moguće. Pogledati ćemo u tu svrhu tablicu istine 1.7. Očito je manji broj linija u kojima kombinacije ulaza daju logičku jedinici, pa se zbog toga razmatraju samo te linije (Isti konačni rezultat dobio bi se naravno i za linije koje daju logičku nulu, samo bi tada postupak bio dulji).

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

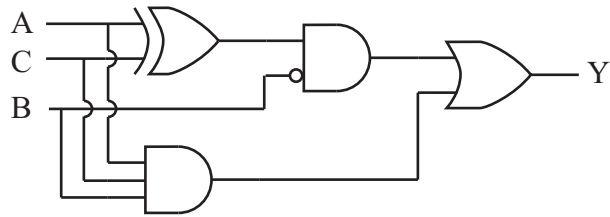
Tablica 1.7: Primjer tablice istine za koju treba realizirati logički sklop

Tablica 1.7 se čita: Y je 1 ako su $ne-A$ i $ne-B$ i C jednaki 1 ili ako su

A i $\text{ne-}B$ i $\text{ne-}C$ jednaki 1 ili ako su A i B i C jednaki 1.

$$Y = \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C = \bar{B} \cdot (\bar{A} \cdot C + A \cdot \bar{C}) + A \cdot B \cdot C = \bar{B} \cdot A \oplus C + A \cdot B \cdot C \quad (1.22)$$

Konačna realizacija sklopa prikazana je na slici 1.21.



Slika 1.21: Izvedba logičkog sklopa za zadanu tablicu istine

2 KOMBINATORNI I SEKVENCIJALNI SKLOPOVI

Postoji mnogo načina na koji se osnovna logička vrata ili sklopovi (koji su razmatrani do sada) mogu kombinirati radi izvođenja raznih korisnih logičkih operacija. Pri tome se razlikuju dvije vrste izlaznih signala:

1. Ako izlazni signali ovise o trenutnim logičkim stanjima ulaznih signala tada se govori o **kombinatornim** logičkim sklopovima.
2. Ako izlazni signali ovise ne samo o trenutnim logičkim stanjima ulaznih signala, već i o prethodnim signalima, pa zapravo ovise o nizu ulaznih signala, tada se govori o **sekvencijalnim** logičkim sklopovima.

2.1 Kombinatorni logički sklopovi

Osnovni zahtjev na svako računalo je da korištenjem logičkih funkcija izvodi aritmetičke operacije. Osnova tih operacija je zbrajanje, jer ako znamo zbrojiti, onda znamo i oduzeti pa zatim i množiti i dijeliti. Kako se, dakle, zbrajaju dva binarna broja?

Uzimamo 2 bita, pri čemu svaki bit može imati dvije moguće vrijednosti 0 ili 1. Rezultat njihovog zbrajanja je:

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 10$$

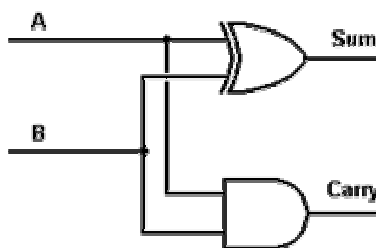
Rezultat pokazuje da se pri zbrajanju dva bita treba za izlaz predvidjeti 2 bita (a ne jedan). Takvi izlazni bitovi nazivaju se **suma** i **prijenos** (*carry*). Istinita tablica je prema tome:

ULAZ		IZLAZ	
A	B	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

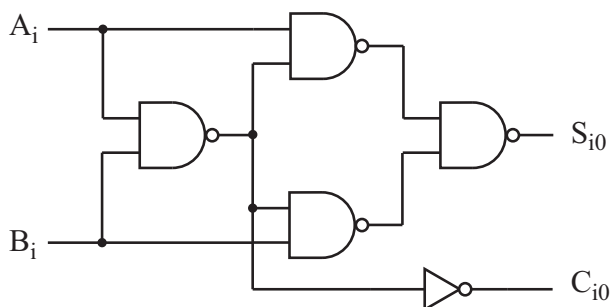
Tablica 2.1: Tablica istine za zbrajanje dva bita

Očito, izlaz *CARRY* je izlaz sklopa AND ($C = A \cdot B$), a izlaz sume je EX-OR ($S = A \oplus B$).

Kombinacijom EX-OR i AND dobiva se (najjednostavniji) sklop za zbrajanje dva bita (slika 2.1 odnosno 2.2):



Slika 2.1: Izvedba sklopa za poluzbrajanje



Slika 2.2: Izvedba sklopa za poluzbrajanje

No, ovo je najjednostavniji sklop za zbrajanje samo 2 bita. U realnosti se zbrajaju brojevi od mnogo više bitova. Isto tako, ako se zbrajanjem



Slika 2.3: *Simbol sklopa za poluzbrajanje*

svakog para bitova dobiva i *carry*, pravi sklop za zbrajanje mora moći "prepoznati" i uključiti u svoj rad vrijednost izlaza *carry* od prethodne operacije. Tako je to uostalom i kod rada s decimalnim brojevima: kad se zbroji jedan stupac tada govorimo: "pišem" ... "pamtim ..." i taj "pamti" se onda dodaje slijedećem stupcu, itd... Zbog toga je sklop koji smo do sada konstruirali samo **sklop za poluzbrajanje** (*half adder*, *HA*) jer se s njime ne dobiva potpuni rezultat!

Ako se, dakle, zbrajaju dva binarna broja A i B , rezultat u i -tom stupcu biti će suma S_i (1 ili 0) i prijenos (*carry*) C_i (isto 1 ili 0), koji se treba dodati $i+1$ stupcu. U binarnom prikazu, gdje su A_i i B_i bitovi i -tog stupca, C_{i-1} prijenos iz prethodnog stupca, a S_i suma i C_i prijenos u slijedeći $i+1$ stupac, vrijedi:

ULAZ			IZLAZ	
C_{i-1}	A_i	B_i	S_i	C_i
carry	znamenka	znamenka	suma	carry
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

Tablica 2.2: *Tablica istine sklopa za potpuno zbrajanje*

Direktna analiza ove istinite tablice je malo kompliciranija (izlaz S je zapravo EX-OR između C_{i-1} i S izlaza half-addera, čiji ulazi su A i B ; također C_i je 1, ako su bilo koja dva od tri ulaza jednaka jedan!), a ovakvu

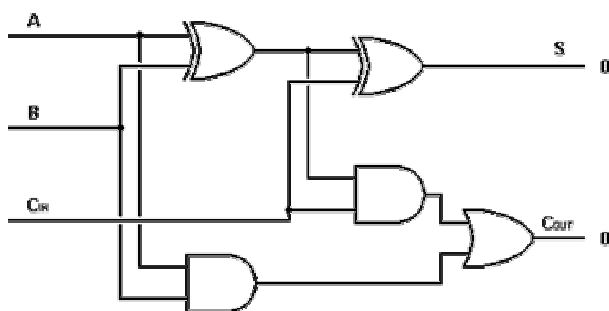
operaciju vrši **sklop za potpuno zbrajanje** (*full adder – FA*). On ima 3 ulaza (A_i , B_i i C_{i-1}) i dva izlaza (S_i i C_i): može se dobiti ako se koriste dva HA (što se intuitivno moglo i pretpostaviti!). Prvi će zbrajati A_i i B_i i davati parcijalnu sumu, a drugi će zbrajati tu parcijalnu sumu sa C_{i-1} i davati konačnu sumu. Kad god će bilo koji od HA sklopova davati *carry*, na izlaznom *carry* biti će jedinica: izlazni *carry* je OR funkcija izlaznih *carry* od dva HA. Vrijedi:

$$S_i = S_{i0} \oplus C_{i-1} \quad (2.1)$$

$$C_i = A_i \cdot B_i + S_{i0} \cdot C_{i-1} = C_{i0} + S_{i0} \cdot C_{i-1} = \overline{\overline{C_{i0} \cdot (S_{i0} \cdot C_{i-1})}} \quad (2.2)$$

gdje su S_{i0} i C_{i0} suma i prijenos ako nema ulaznog prijenosa (tj. suma i prijenos HA).

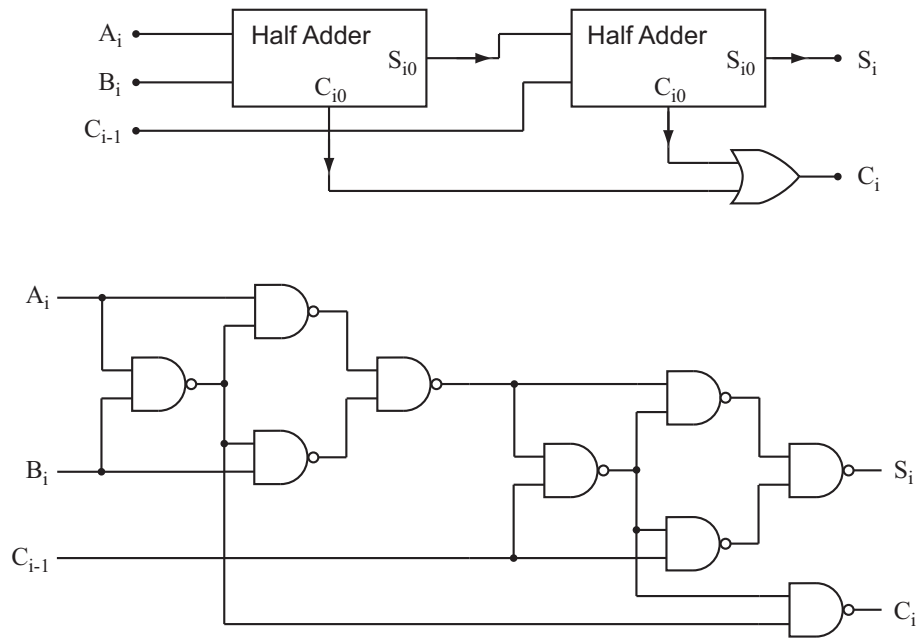
Cjelokupna shema sklopa za potpuno zbrajanje je prikazana na slikama 2.4 i 2.5.



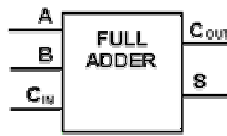
Slika 2.4: Izvedba sklopa za potpuno zbrajanje

Očito je da je ovakav prikaz FA previše kompliciran da bi se koristio u većim i kompleksnijim shemama. Zato se uvodi shema full addera za 1 bit. (To je zato jer nas u osnovi zapravo ne zanima egzaktan način kako je neki logički sklop izveden za izvođenje neku operacije, već samo ta sama logička funkcija odnosno operacija).

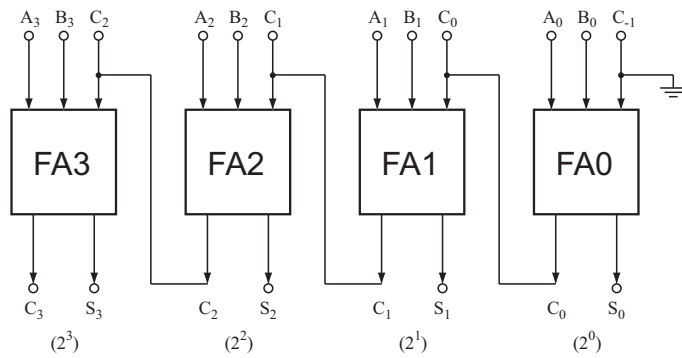
Zbrajanje dva višeznamenakasta broja (N znamenki) može se ostvariti serijski (posebno svaki stupac) ili paralelno (istovremeno zbrajanje svih stupaca) - što je uobičajenije. Kod paralelnog zbrajanja se za svaki par znamenki, koje se žele zbrojiti, koristi 1 FA. Sve se brojke unose istovremeno, a izlazni prijenos jednog FA je ulazni prijenos za slijedeći FA. Za svaku znamenku zbroja (osim LSD, tj. potencije 2^0) potreban je FA. Izlaz za LSD dobiva se pomoću HA (jer nema prethodnog prijenosa).



Slika 2.5: Izvedba sklopa za potpuno zbrajanje



Slika 2.6: Simbol sklopa za potpuno zbrajanje



Slika 2.7: Paralelno zbrajanje 4 bitna broja

Uz oznake A_n , B_n , S_n i C_n (gdje je n potencija broja 2), za $n = 3$, za paralelno zbrajanje 4 bitna broja, koji će dati 4-bitnu sumu i mogući *carry*, treba koristiti 4 FA (slika 2.7).

Za svaku sumu i prijenos vrijedi (na osnovu tablice za zbrajanje 3 binarna bita)

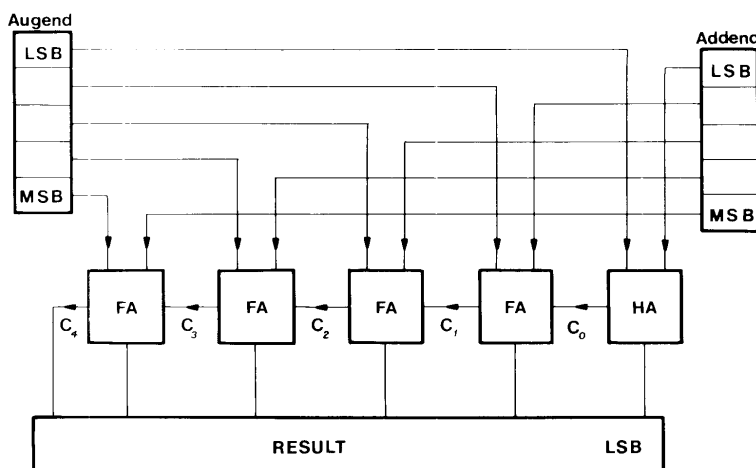
$$S_n = \bar{A}_n \cdot \bar{B}_n \cdot C_{n-1} + \bar{A}_n \cdot B_n \cdot \bar{C}_{n-1} + A_n \cdot \bar{B}_n \cdot \bar{C}_{n-1} + A_n \cdot B_n \cdot C_{n-1} \quad (2.3)$$

$$C_n = \bar{A}_n \cdot B_n \cdot C_{n-1} + A_n \cdot \bar{B}_n \cdot C_{n-1} + A_n \cdot B_n \cdot \bar{C}_{n-1} + A_n \cdot B_n \cdot C_{n-1} \quad (2.4)$$

Izraz za C_n se može bitno pojednostaviti: ako se uvede $Y = A_n \cdot B_n \cdot C_{n-1}$, koristi jednakosti $Y = Y + Y + Y$ te $X + \bar{X} = 1$ (pri čemu je, shodno potrebi, $X = A_n, B_n$ ili C_{n-1}), tada slijedi:

$$C_n = B_n \cdot C_{n-1} + C_{n-1} \cdot A_n + A_n \cdot B_n \quad (2.5)$$

Slično je za dva 5-bitna broja (drugačiji prikaz) – slika 2.8.



Slika 2.8: Paralelno zbrajanje 5 bitna broja

Istim sklopom može se izvesti i binarno oduzimanje – dovoljno je na ulaz B staviti negativni broj.

2.2 Sekvencijalni logički sklopovi

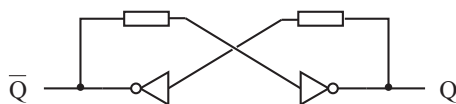
U sekvencijalnoj logici signali moraju slijediti u određenom redu da bi se na izlazu dobio željeni signal. To znači da sekvencijalni logički sklopovi moraju imati i neki oblik memorije, radi "pamćenja" dijela signala koji su već stigli na ulaz. Ovaj memorijski efekt se postiže korištenjem određene povratne veze.

Sekvencijalni logički sklopovi se općenito dijele na:

1. sinkroni sklopovi ili sustavi – u njima se sve promjene dešavaju istovremeno u trenutku određenom nekim upravljačkim signalom zajedničkim za sve ulazne signale.
2. asinkroni sklopovi ili sustavi – nemaju upravljačkog signala, a promjene u jednom dijelu sklopa izazivaju promjenu u drugom dijelu; promjena se širi brzinom koju određuje rad pojedinog dijela sustava.

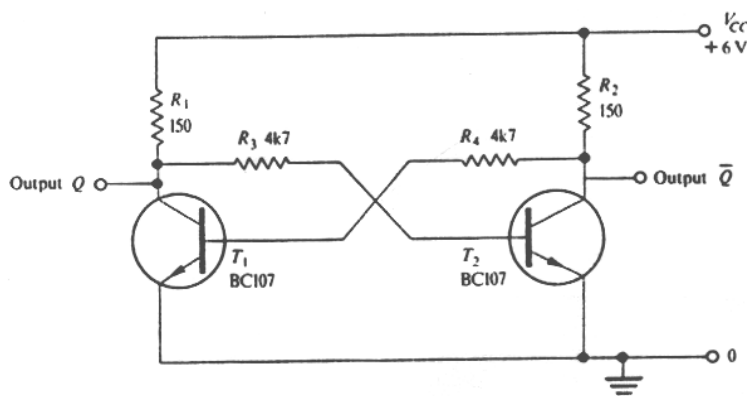
2.2.1 Flip-flop s diskretnim elementima

Većina sekvencijalnih logičkih sklopova zasniva svoj rad na radu **flip-flopa** (drugi nazivi su *bistabil* te *binar*), koji ima 2 različita stanja. Flip-flop je tzv. regenerativni sklop, koji u osnovi čine dvije međusobno povezane sklopke, pri čemu je u svakom trenutku jedna zatvorena, a druga otvorena. Promjena njihovog stanja može se postići sama (nakon određenog vremena) ili pomoću vanjskog impulsa.



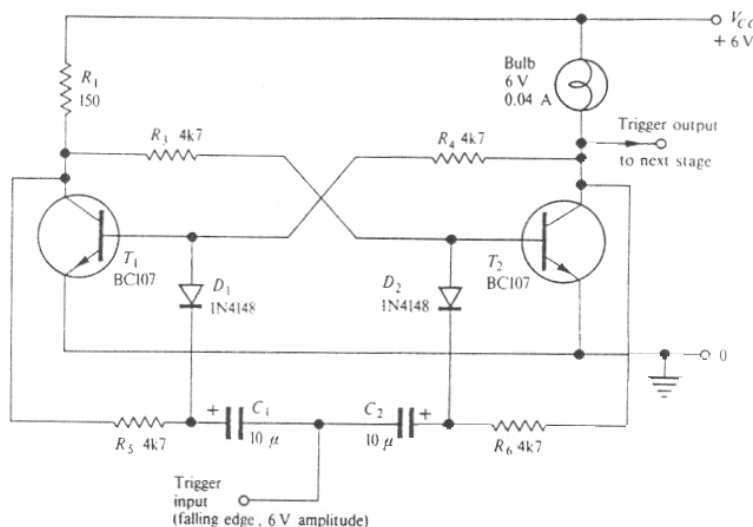
Slika 2.9: Princip flip-flopa

Sheme diskretnih izvedbi flip-flopa su prikazane na slikama 2.10 i 2.11.



Slika 2.10: Izvedbe flip-flopa s diskretnim elementima

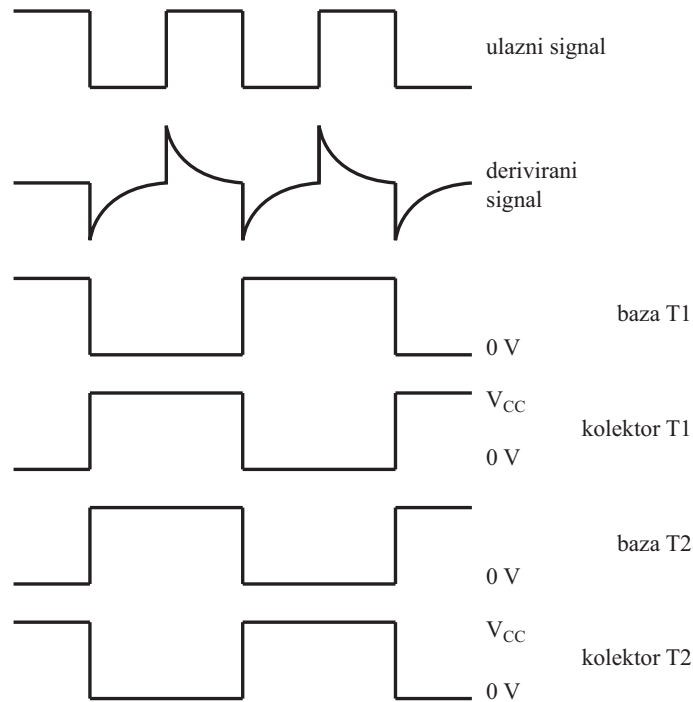
Neka je u trenutku uključivanja napona napajanja V_{CC} , lijevi transistor T_1 otvoren, tj. u stanju ON ($V_{CE1} \simeq 0$). Tada kroz otpornik R_3 ne teče struja i desni tranzistor T_2 je zatvoren ($V_{CE2} \simeq V_{CC}$), a to pak znači da kroz otpornik R_4 teče struja i "drži" otvoren T_1 . Ako se pomoću nekog vanjskog impulsa T_1 zatvori, tada će T_2 prijeći u otvoreno stanje. Na taj način se dobiva drugo stabilno stanje flip-flopa. Bitna karakteristika je da sklop "pamti" posljednje stanje (tj. ostaje u njemu), i kao takav je osnovni element za brojače i memorije.



2.5 Bistable with trigger input suitable for binary counting.

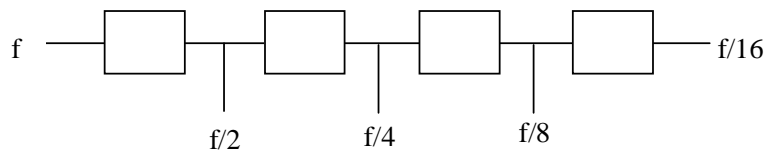
Slika 2.11: Izvedba flip-flopa sa sklopom za okidanje

Realna izvedba (slika 2.11) uključuje i sklop za okidanje s pravokutnim impulsima (*trigger*) pomoću kojeg se flip-flop prebacuje iz jednog u drugo stanje. Elementi R_5C_1 odnosno R_6C_2 čine sklopove za deriviranje, pri čemu negativni dijelovi deriviranih pravokutnih signala prebacuju flip-flop iz jednog u drugo stabilno stanje. Prvi negativni impuls snižava napon baze T_1 , zbog čega napon na kolektoru T_1 raste na V_{CC} . Porast tog napona se prenosi na bazu tranzistora T_2 , što pak izaziva pad napona na kolektoru T_2 . Slijedeći pozitivni impuls ne izaziva nikakve promjene stanja flip-flopa (zbog postojanja dioda). Novi negativni impuls sada snižava napon baze T_2 , što izaziva porast napona na kolektoru T_2 , koji se prenosi na bazu T_1 , i zbog čega pada napon na kolektoru T_1 . Ukupno, to znači da se za 2 ulazna impulsa dobiva 1 izlazni impuls, tj. operacija dijeljenja s 2!



Slika 2.12: Oblici napona na relevantnim točkama sklopa sa slike 2.11

Serijskim spajanjem binara dobivaju se brojači u binarnom sustavu. Uvođenjem određene povratne veze takvi se brojači mogu pretvoriti u brojače u dekadskom sustavu.



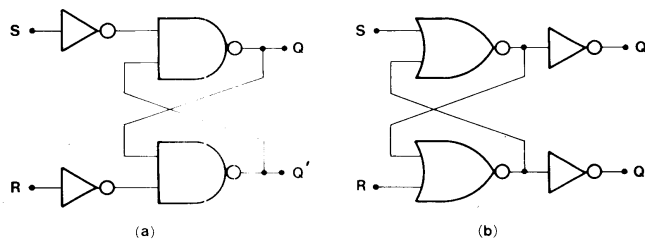
Slika 2.13: Binarni brojač

2.2.2 SR flip-flop

Kod asinkronog okidanja se izlaz bistabila uvijek postavlja u određeno stanje, zavisno da li je okidni impuls došao na jedan ili na drugi ulaz. (Ako bi okidni impuls došao na ulaz koji bistabil postavlja u stanje "1", onda je na njemu odgovarajućem izlazu (izlazu 1) stanje "1", a na drugom (komplementarnom) izlazu (izlazu 0) je stanje "0". Ako se pak sada okidni signal

dovede na drugi ulaz, tada je na izlazu 0 stanje "1", a na izlazu 1 stanje "0").

Najjednostavniji asinkroni sekvencijalni sklop u integriranoj izvedbi je SR flip-flop. On se može ostvariti na više načina, a najčešće je to s NAND logičkim vratima.



Slika 2.14: SR flip-flop

Ulazi su S i R , a izlazi Q i Q' . U odnosu na kombinatorne logičke sklopove, u sekvencijalnim sklopovima postoji povratna veza, zbog koje izlaz ne ovisi samo o ulazu već i izlazu!

Za sklop s NAND vratima (slika 2.14. (a)) može se pisati:

$$Q = \overline{\overline{S} \cdot \overline{Q'}} = S + \overline{Q'} \quad (2.6)$$

$$Q' = \overline{\overline{R} \cdot \overline{Q}} = R + \overline{Q} \quad (2.7)$$

Mogući slučajevi su:

- (a) $S = R = 0$. To je normalno stanje mirovanja sklopa, u kojem su Q i Q' različiti ($Q = \overline{Q'}$), pri čemu je to moguće za dvije kombinacije $Q = 1$ i $Q' = 0$ odnosno $Q = 0$ i $Q' = 1$. No oba ta moguća stanja su jednako stabilna i vjerojatna dok god je $S = R = 0$. To onda u stvari znači da izlazi nisu do kraja definirani, tj. stvarna stanja izlaza ovise o tome što se prethodno dogodilo.
- (b) $S = 1, R = 0$. Ovakva kombinacija ulaznih signala stavlja izlaz Q u stanje logičke jedinice – zato se ulaz S naziva *SET*.
- (c) $S = 0, R = 1$. To je obrnut slučaj; ovakvom kombinacijom izlaz Q prelazi u stanje logičke nule – ulaz R se zato naziva *RESET*.

- (d) $S = R = 1$. Za ovu kombinaciju ulaza su oba izlaza u stanju logičke jedinice ($Q = 1 + \overline{Q'}$, $Q' = 1 + \overline{Q}$). To samo po sebi ne predstavlja nikakvu poteškoću za rad sklopa, sve dok su ulazi S i R na stalnom iznosu logičke jedinice. Problem nastaje kad se S i R simultano mijenjaju s 1 na 0, što onda vodi na jedno od moguća dva početna stanja, pri čemu nije definirano na koje od njih! Ako su $Q = Q' = 1$, a S i R prelaze simultano iz 1 u 0, tada bi i Q i Q' trebali postati 0. No za $Q = Q' = 0$ i $S = R = 0$ bi Q i Q' trebali prijeći u stanje 1! Očito se dobiva neka vrsta oscilacije izlaza između stanja 0 i 1. U realnom slučaju ipak ne dolazi do oscilacija, jer se ulazni signali ne mijenjaju jednako brzo, pa će ulaz koji prvi prijeđe u stanje 0 izgubiti kontrolu, dok će drugi ulaz (koji će ostati na nivou 1) upravljati bistabilom. Zbog toga će sklop ipak doći u jedno od moguća dva stanja (koje nam međutim nije poznato). Općenito može se reći da je izlaz sklopa zapravo neodređen, što se naziva *race condition* ili *hazard* (*race* zato jer konačno stanje izlaza ovisi o relativnom odnosu brzina promjena ulaza). Stanje neodređenosti se treba izbjegavati, tj. kombinacija $S = R = 1$ se ne smije nikada pojaviti, i sklop će uvijek davati različite izlaze, pri čemu će uvijek biti $Q = \overline{Q'}$. Zato se komplementarni izlazi Q i Q' označuju s Q i \overline{Q} .

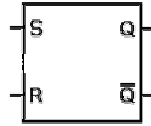
Rad SR flip-flopa može se prema tome sažeti s:

1. S i R su uobičajeno na nivou logičke nule, a izlazi su nepromijenjeni u jednom od dva moguća stanja $Q = \overline{Q}$;
2. Ulaznim nizom $0 \rightarrow 1 \rightarrow 0$ na ulazu S ostvaruje se $Q = 1$ i $Q' = 0$;
3. Ulaznim nizom $0 \rightarrow 1 \rightarrow 0$ na ulazu R ostvaruje se $Q = 0$ i $Q' = 1$;
4. U normalnom režimu rada ne smije postojati slučaj $S = R = 1$;

U nekim primjenama je važno početno stanje sklopa, pa je zbog toga potrebno posebnim sklopom postaviti bistabil u željeno početno stanje.

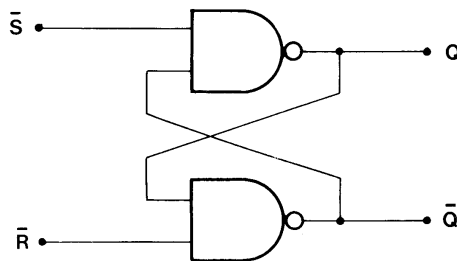
Shematski prikaz SR flip-flopa je na slici 2.15.

SR flip-flopom se upravlja koristeći niske nivoe ulaznih signala, odnosno logičkom nulom. S druge strane, bilo bi "očiglednije" da logička jedinica kontrolira sklop, a da se tokom logičke nule ništa ne dešava. Taj problem



Slika 2.15: *Simbol SR flip-flopa*

se može jednostavno riješiti stavljanjem invertera na ulaze (što se i realizira u TTL izvedbi), a takav sklop se naziva \overline{SR} flip-flop (slika 2.16). Njegov



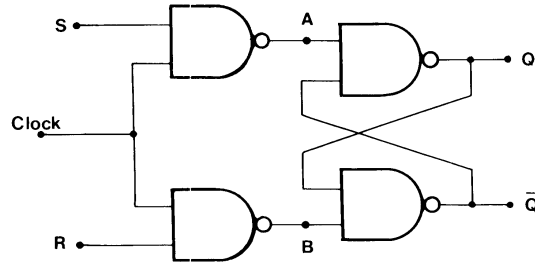
Slika 2.16: *Simbol \overline{SR} flip-flopa*

rad je u potpunosti analogan radu SR flip-flopa, pri čemu je nedozvoljeno stanje ulaza $\overline{S} = \overline{R} = 0$, a *SET* i *RESET* djelovanje se postiže sa nizom $1 \rightarrow 0 \rightarrow 1$.

2.2.3 Taktni SR flip-flop (*clocked SR flip-flop*)

Jedan problem osnovnog SR flip-flopa bio je da se ulazni impulsi trebaju invertirati da se postigne 'očigledan' rad. SR flip-flop naime reagira samo na logičku nulu (a za logičke jedinice na ulazu postiže se neodređenost izlaza), a bilo bi "normalnije" da logička jedinica kontrolira sklop, a da se tokom logičke nule ne dešava ništa. Taj problem se jednostavno riješio sa \overline{SR} flip-flopom. S druge strane, SR flip-flop spada u grupu tzv. asinkronih bistabila, tj. sklopova koji mogu mijenjati stanje u bilo kojem času, tj. čim se pojavi ulazni signal. Ako je pak potrebno kontrolirati upravljanje promjenu izlaznog stanja, (a naročito ako je to potrebno izvršiti istovremeno s velikim brojem istih sklopova), uvodi se dodatni ulaz za taktne (*clock*) impulse. U takvim tzv. sinkronim bistabilima se uvjeti za promjenu izlaznih stanja ispunjavaju ranije, ali do same promjene dolazi tek u trenutku pojave taktnog impulsa. Korištenjem istog taktnog impulsa za veći broj istovrsnih bistabila, osigurava se njihovo sinkrono (istovremeno) djelovanje. Shema

sinhronog SR–bistabila je prikazana na slici 2.17.



Slika 2.17: Taktvi SR flip–flop

Dva dodatna NAND sklopa na ulazu osiguravaju korištenje normalnih (a ne invertiranih) impulsa, te sinkronizaciju. Rad sklopa može se opisati na slijedeći način:

		t_n		t_{n+1}		
S	R	Q_n	\bar{Q}_n	Q_{n+1}	\bar{Q}_{n+1}	napomena
0	0	0	1	0	1	nema
0	0	1	0	1	0	promjene
1	0	0	1	1	0	SET
1	0	1	0	1	0	
0	1	0	1	0	1	RESET
0	1	1	0	0	1	
1	1	0	1	?	?	“race
1	1	1	0	?	?	condition”

Tablica 2.3: Rad taktnog SR flip–flopa

- (a) Ako je $CLK = 0$ tada su izlazi A i B uvijek 1, neovisno o stanju ulaza S i R . Ostatak sklopa se tada ponaša kao flip–flop, s ulazima $A = B = 1$, što znači da izlazi Q i \bar{Q} zadržavaju svoje prethodne vrijednosti. Drugim riječima, za $CLK = 0$ izlazi ne ovise o S i R (pa je moguć i slučaj neodređenosti $S = R = 1$).
- (b) Ako je $CLK = 1$, ulazi S i R utječu na stanje izlaza: pojavom taktnog impulsa dobiva se normalni rad flip–flopa. Rad taktnog flip–flopa može se prikazati u tabličnom obliku (koji je analogan istinitim tablicama u kombinatornoj logici). (t_n označava vrijeme prije taktnog impulsa,

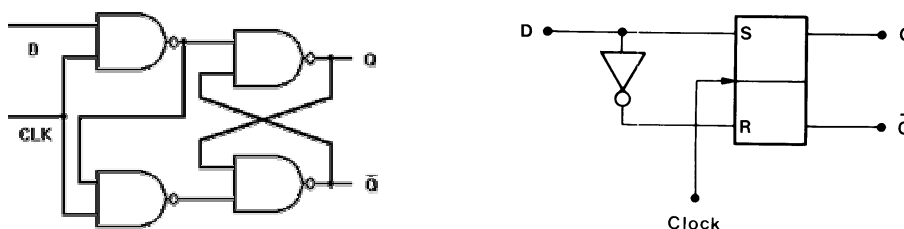
t_{n+1} označava vrijeme nakon taktnog impulsa, Q_n i Q_{n+1} te Q'_n i Q'_{n+1} označuju stanja izlaza koji odgovaraju tim vremenima).

Iako se s taktnim SR flip-flopom osigurava sinkronizacija, taj sklop i dalje ima određene nedostatke:

1. Za $S = R = 1$ ostaje problem neodređenosti rada sklopa;
2. U slučaju da je $CLK = 1$ svaka eventualna i/ili neželjena promjena ulaza S i/ili R će se manifestirati na izlazu, tj. sklop može promijeniti izlazno stanje prije nego se taktni impuls vrati u stanje logičke nule. Čak i ako se CLK ulaz "drži" stalno na logičkoj nuli i dozvoljava samo njegova brza promjena na logičku jedinicu, to ne osigurava da će flip-flop samo jednokratno promijeniti svoje izlazno stanje (a osim toga kratkotrajno trajanje $CLK = 1$ ne mora biti dovoljno za istovremeno okidanje velikog broja spojenih flip-flova).

2.2.4 D flip-flop

Vrlo korisna verzija taktnog SR flip-flopa je D flip-flop (*Data*). On se dobiva tako da se kao R ulaz koristi invertirani S ulaz. Sklop prema tome ima samo jedan ulaz, koji se označava s D . Dva načina prikaza D flip-flopa su (slika 2.18).



Slika 2.18: D flip-flop

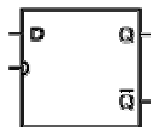
Ako je taktni signal (CLK) logička jedinica, izlaz Q će uvijek dati logički nivo koji postoji na D , bez obzira kako se on mijenja.

Ako je CLK ulaz logička nula, zadnje logičko stanje koje je postojalo na D ulazu je zadržano u bistabilu (te se može – ako je to potrebno – dalje koristiti). Budući je uvijek $R = \bar{S}$, na izlazu se nikada ne može dobiti stanje neodređenosti (*race condition*).

Početno stanje		Ulazni signal		Novo stanje	
Q	\bar{Q}	D	CLK	Q	\bar{Q}
1	0	0	1	0	1
1	0	1	1	1	0
0	1	0	1	0	1
0	1	1	1	1	0

Tablica 2.4: Rad D flip-flopa

Na simboličkom prikazu D flip-flopa (slika 2.19) je CLK ulaz prikazan s malim polukrugom, koji označuje da se rad sklopa regulira s naponskom razinom CLK signala.



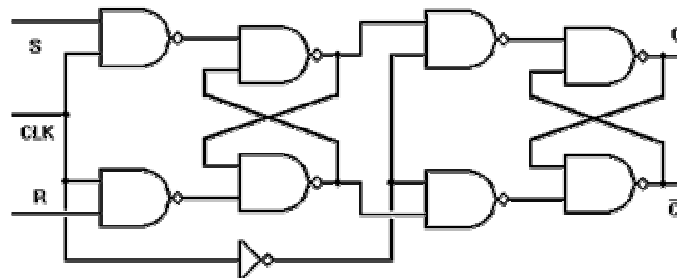
Slika 2.19: Simbol D flip-flopa

2.2.5 Master-slave flip-flop

Ovaj sklop osigurava da se izlazi mijenjaju samo u danom trenutku takt-nog impulsa, i to pri njegovoj nagloj promjeni (*edge triggered*). Sastoji se od dva ista takt-na SR flip-flopa (prvi tzv. *master*, a drugi tzv. *slave*). Inverter između dva CLK ulaza osigurava da su svaki od dva bistabila aktivni tijekom suprotnih dijelova takt-nog impulsa, što je bitna karakteristika rada, te se tako može precizno kontrolirati trenutak kad flip-flop mijenja svoje stanje. Naime, ako je ulazni CLK impuls na nivou logičke nule, to znači da je CLK za *slave* na nivou logičke jedinice (i obratno). To pak znači da je *master* blokiran, tj. da je u njemu spremljeno ono što je bilo na njegovim ulazima kad je CLK bio logička jedinica.

Schema master-slave flip-flopa je na slici 2.20.

- (a) Ako je $CLK = 0$, ulazi S i R nemaju utjecaj na stanje izlaza.
- (b) Kad CLK ulaz prijeđe u stanje logičke jedinice, ulazi S i R upravljaju izlazima ulaznog (*master*) bistabila. Istovremeno invertirani CLK sig-



Slika 2.20: *Master-slave flip-flop*

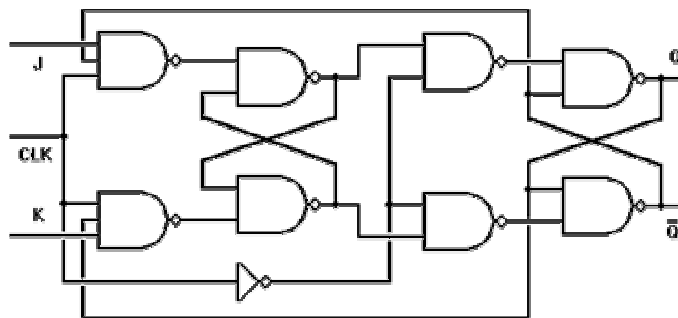
nal na ulazu izlaznog (*slave*) bistabila sprječava utjecaj *master*-a na *slave* bistabil. To znači da se promjene ulaza S i R ne manifestiraju na izlazima Q i Q' .

- (c) Kad CLK ulaz prijeđe natrag u stanje logičke nule, ulazi S i R ponovno ne utječu na ulazni (*master*) bistabil, već je na njegovim izlazima ono stanje koje je bilo kad je CLK bio jedinica. Istovremeno, invertirani CLK signal utječe na izlazni (*slave*) bistabil. To sada znači da se izlazi Q i Q' mijenjaju samo kad CLK signal pada s nivoa logičke jedinice na nivo logičke nule (*edge triggered*).

Preostaje problem moguće neodređenosti u slučaju da su i S i R u stanju logičke jedinice kad CLK signal pada s 1 na 0. Taj se problem rješava dodatnom povratnom vezom sa *slave* na *master* bistabil, a dobiveni sklop se naziva JK flip-flop

2.2.6 JK flip-flop

JK flip-flop je najkompliciraniji, ali vjerojatno i najvažniji od svih bistabila. On se može koristiti kao i svaki od prethodnih bistabila, ali $J = K = 1$ slučaj (jednak slučaju $S = R = 1$) ne samo da nije neodređen, već se i koristi kao prednost. Neodređenost (*race condition*) u radu *edge-triggered* SR flip-flopa se javlja kad su oba ulaza S i R u stanju logičke jedinice, a CLK ulaz pada s logičke jedinice na logičku nulu. To se može izbjeći tako da se spriječi utjecaj ulaza S i R na master sklop, ali istovremeno se mora dozvoliti da se izlazno stanje promijeni pri promjeni CLK signala (ako su se promijenili ulazi).



Slika 2.21: JK flip-flop

Iz sheme (slika 2.21) je vidljivo da se u JK flip-flopu izlazi Q i Q' vraćaju na ulaz (NAND vrata mogu imati neograničeni broj ulaza), a novi ulazi su sada J (umjesto S) i K (umjesto R).

U osnovi se JK flip-flop ponaša poput SR flip-flopa. Izlazi Q i Q' mijenjaju svoja stanja samo na padajućem dijelu CLK signala, a J i K ulazi upravljaju s budućim izlaznim stanjima, kao što je to bilo i u prethodnim slučajevima. Ako se, dakle, ograničimo samo na "radni" dio (kada se CLK impuls mijenja od logičke jedinice na logičku nulu) dobiva se tablica akcije (djelovanja) JK flip-flopa (tablica 2.5).

J	K	izlaz Q
0	0	nema promjene; $Q_{n+1} = Q_n$
1	0	SET ; $Q_{n+1} = 1$
0	1	$RESET$; $Q_{n+1} = 0$
1	1	promjena ($toggle^1$); $Q_{n+1} = \overline{Q_n}$

Tablica 2.5: Tablica istine za JK flip-flop (t_n je trenutak prije promjene CLK ulaza, t_{n+1} trenutak nakon promjene CLK ulaza).

JK flip-flop se može prikazati pomoću još jedne tablice, tzv. *excitation table* – tablica 2.6, u kojoj su navedeni zahtjevi na ulaze J i K (tj. upravljanje ulazima) s kojima se postiže ispunjenje određenih zahtjeva na stanje

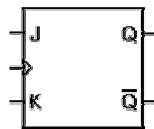
¹toggle - to change a bit from whatever state it is in to the other state; to change from 1 to 0 or from 0 to 1. The word "toggle" actually refers to the mechanism that keeps the switch in the position to which it is flipped rather than to the fact that the switch has two positions. There are four things you can do to a bit: set it (force it to be 1), clear (or zero) it, leave it alone, or toggle it.

izlaza prije (Q_n) i nakon (Q_{n+1}) djelovanja CLK impulsa. (X = nije bitno = 'don't care')

Budući da je ponašanje JK flip-flopa u potpunosti opisano za sve kombinacije ulaza, taj bistabil je najčešće korišteni bistabil u većini logičkih sklopova. SR flip-flop se koristi samo kad je sigurno da se ne pojavljuje slučaj $S = R = 1$.

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Tablica 2.6: 'Excitation table' za JK flip-flop



Slika 2.22: Simbol JK flip-flopa (trokutić označava 'edge triggered')

2.2.7 Zapis binarnih brojeva – pomačni ili *shift* registar

Da bi se mogli zapisati svi brojevi decimalnog brojnog sustava bio bi potreban neki element koji ima deset stabilnih stanja – po jedno za svaku brojku. Za zapisivanje binarnih brojeva dovoljno je pak da odgovarajući element ima samo dva stabilna stanja, a takav element je bistabil. Niz međusobno povezanih bistabila, na koji se mogu zapisivati (registrirati) binarni brojevi s odgovarajućim brojem binarnih mjesta, naziva se registar. Broj binarnih mjesta odgovara broju bistabila, tj. registar može registrirati onoliko bitova binarnog broja koliko ima bistabila.

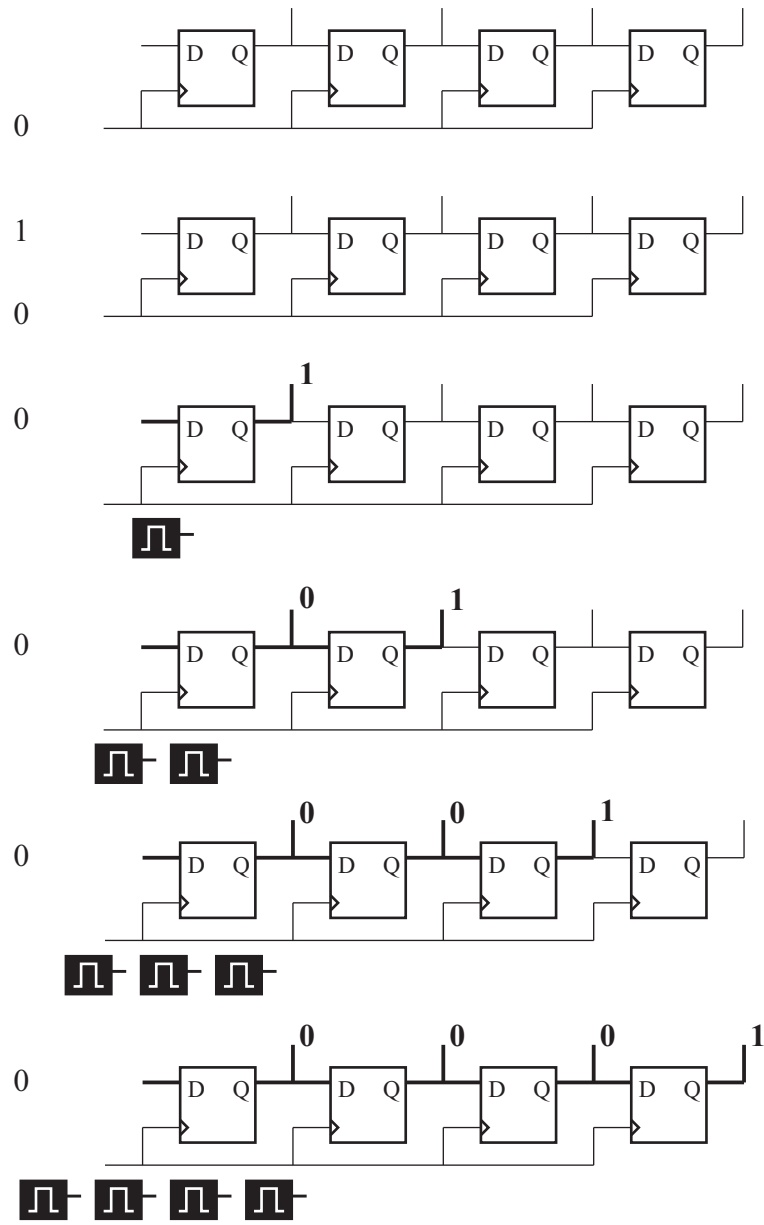
Najjednostavniji pomačni registar može se dobiti pomoću D-bistabila, tako da se izlaz prethodnog bistabila veže na ulaz slijedećeg bistabila. To je tzv. *serial-in, parallel-out shift register* ili *serial-to-parallel shift register*.

Pretpostavimo da su prije početka rada svi ulazi na nivou logičke nule. Pretpostavimo, također, da se nakon toga na ulaz D priključi logička jedi-

nica, a na *CLK* ulaz niz taktnih impulsa, pri čemu se nakon prvog taktnog impulsa, signal na *D* vrati u stanje logičke nule i tamo ostaje. (Po standardnoj konvenciji, najmanje značajni bit (LSB) se pomiče prvi).

Nakon prvog taktnog impulsa izlaz prvog flip-flopa je na nivou logičke jedinice. Drugi taktni impuls prebacuje drugi flip-flop u stanje logičke jedinice, itd.: nakon 4 taktna impulsa je prvi ulazni podatak u potpunosti serijski prenešen na izlaz zadnjeg (četvrtog) flip-flopa, a daljnji taktni impulsi dovode do toga da je početni podatak izgubljen i počinje novi proces prijenosa. No ulazni podatak (koji je pohranjen u registru) je prisutan i u paralelnom obliku, ako se istovremeno promatraju sva 4 izlaza bistabila (to je tzv. *serial-to-parallel shift register*). Dakle, ulazni podatak se unosi u sekvencijalnom (serijskom) obliku, a nakon toga prikazan je u 'paralelnom' obliku – najjednostavniji primjer korištenja takvog registra je u kalkulatorima, kod kojih se ulazni podatak unosi serijskim ukucavanjem znamenaka nekog broja, a zatim se s tim podatkom vrši neka aritmetička operacija (za zbrajanje su potrebna 2 *parallel shift register*).

Obrnuti, *parallel-in, serial-out shift register*, ima nešto kompleksniju izvedbu, ali je njegov rad, općenito, vrlo sličan.



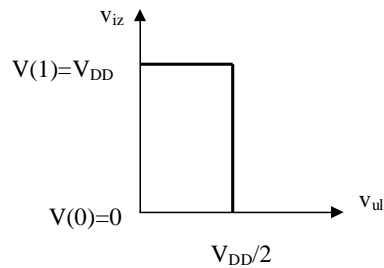
Slika 2.23: Shift registar

3 IDEALNI I REALNI INVERTER

U ovom dijelu usporediti će se osnovne karakteristike idealnog i realnog logičkog invertera.

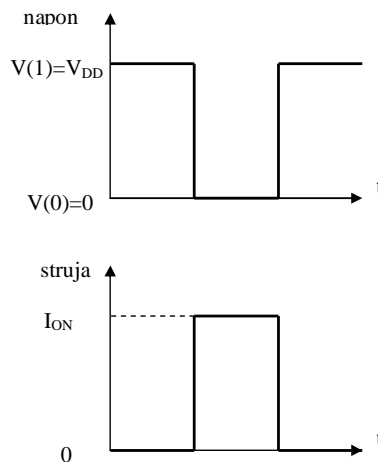
IDEALNI INVERTER

1. Prijelaz iz jednog u drugo logičko stanje dešava se na 50% vrijednosti napona koji odgovara logičkoj jedinici (slika 3.1).



Slika 3.1:

2. Nema ograničenja na iznose napona koji odgovaraju stanjima $V(1)$ i/ili $V(0)$; neodređenost u ulaznim naponima nemaju utjecaja.
3. Vremenska ovisnost napona i struje je takva da je statička disipacija jednaka nuli (slika 3.2): za otvoreni krug je napon jednak naponu $V_{DD} = V(1)$, ali je struja nula; za zatvoreni krug je struja konačna, ali je napon nula.
4. Dinamička disipacija (snaga utrošena tijekom prekidanja) je također nula (prijelaz iz $V(1)$ u $V(0)$ ili obrnuto je trenutačan).
5. Vrijeme prebacivanja (iz $V(1)$ u $V(0)$ ili obrnuto) je nula.



Slika 3.2:

6. Broj ulaznih i/ili izlaznih stupnjeva je neograničen.

REALNI INVERTER

1. Ovisnost izlaznog napona o ulaznom naponu prikazana je na slici 3.3, a nagib te krivulje je zapravo pojačanje!

Za $v_{ul} < V_{inL}$ te $v_{ul} > V_{ulH}$ je nagib krivulje manji od 1, u točkama A i B je nagib krivulje -1, a između A i B je nagib krivulje veći od 1. To znači da je prijelaz između dva logička stanja moguć samo za $V_{inL} \leq v_{ul}$ i $v_{ul} \geq V_{inH}$.

Najviši ulazni napon koji odgovara stanju $V(0)$ je V_{inL} i on dovodi do prijelaza u $V(1)$. To znači da je V_{izH} je najniži izlazni napon koji odgovara stanju $V(1)$. Analogno, V_{inH} je najniži ulazni napon koji odgovara stanju $V(1)$ i koji dovodi do prijelaza u $V(0)$, pa je V_{izL} je najviši izlazni napon koji odgovara stanju $V(0)$.

Spajanjem izlaza prvih logičkih vratiju na ulaz drugih vratiju dobivaju se uvjeti na iznose napona za normalan rad takve kombinacije vratiju:

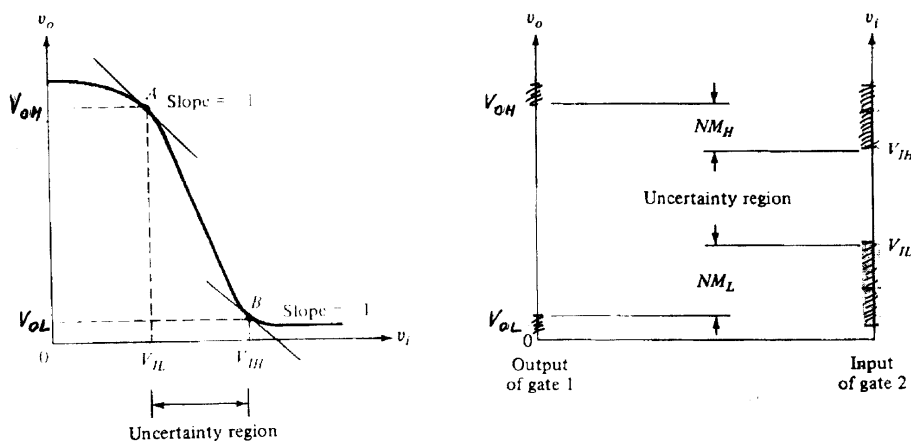
$$\begin{aligned} V_{izH1} &> V_{ulH2} \\ V_{izL1} &< V_{ulL2} \end{aligned} \tag{3.1}$$

Razlike definiraju okvire dozvoljenog šuma (*noise margins – NM*), tj. neodređenosti u naponskim nivoima koji neće utjecati na normalan

rad:

$$\begin{aligned} NM_H &= V_{izH} - V_{ulL} \\ NM_L &= V_{izH} - V_{ulL} \end{aligned} \quad (3.2)$$

a područje između NM je područje neodređenosti napona, za koje sklop neće raditi.

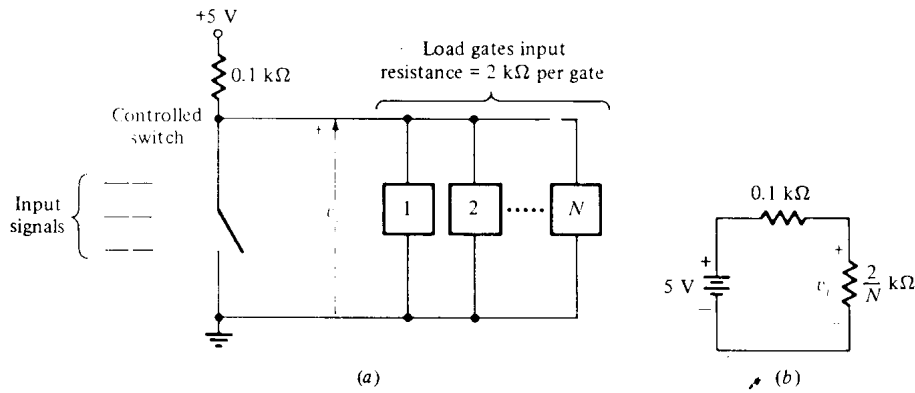


Slika 3.3: Realni inverter: (a) ovisnost izlaznog napona o ulaznom naponu; (b) definicija područja neodređenosti napona. ($i = input = ulaz$; $o = output = izlaz$; $L = low = V(0)$; $H = high = V(1)$)

2. *Fan-in* je broj ulaza koji mogu biti priključeni na neki logički sklop. Ako je broj tih ulaza veći od deklariranog iznosa, izlaz logičkog sklopa će biti ili nedefiniran ili netočan, a može doći i do izobličenja ulaznih impulsa zbog preopterećenja.
3. *Fan-out* je broj krugova koji mogu biti priključeni na izlaz nekog logičkog sklopa (i koje taj sklop može dalje pokretati). Ako se npr. pretpostavi da je opterećenje logičkog sklopa (slika 3.4) sastavljeno od N jednakih sklopova (pri čemu je ulazni otpor svakog od njih $2 \text{ k}\Omega$), tada će logički sklop biti opterećen s $2/N \text{ k}\Omega$. Ako je još potrebno da izlazni napon logičkog sklopa bude najmanje 3.5 V (a napon napajanja 5 V), tada je:

$$v_i = \frac{2/N}{0.1 + 2/N} \cdot 5 \geq 3.5V \quad (3.3)$$

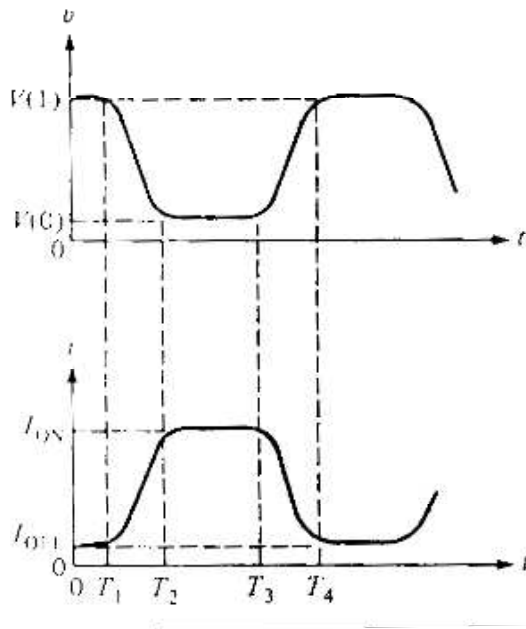
odnosno $N = 8.57$. Budući da broj izlaznih sklopova treba biti cijeli broj, *fan-out* je u ovom slučaju jednak 8.



Slika 3.4: Fan-out: shema sklopa i ekvivalentna shema

4. Disipacija (P_D)

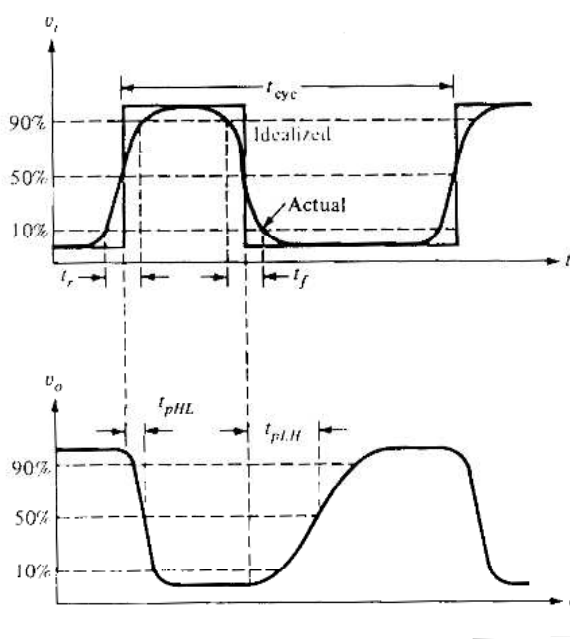
Statička disipacija je uvijek različita od nule: ili je nivo $V(0) \neq 0$ ili je struja prekida (za $V(1)$ nivo) različita od nule. Čak i kad je statička disipacija značajno reducirana (CMOS), postoji konačna dinamička disipacija pri prijelazima između dva logička stanja.



Slika 3.5: Realni oblici napona i struje kod invertera. Disipacija postoji u oba logička stanja (statička disipacija) ali i tokom prijelaza između ta dva stanja (dinamička disipacija)

5. Umnožak kašnjenja i disipacije (*Delay-Power Product DP*)

Dinamička svojstva nekog logičkog sklopa karakterizira vrijeme kašnjenja (*propagation delay* t_p) njegovog osnovnog invertera, tj. vrijeme potrebno za prijelaze između dva logička stanja. U realnom ulaznom signalu razlikuju se vrijeme porasta t_r i pada t_f , a u realnom izlaznom signalu vremena kašnjenja t_p za prijelaz $H \rightarrow L$ (t_{pHL}) i prijelaz $L \rightarrow H$ (t_{pLH}) (pri čemu ona ne moraju nužno biti jednaka) – slika 3.6.



Slika 3.6:

Ukupno vrijeme kašnjenja invertera je definirano kao:

$$t_p = (t_{pLH} + t_{pHL})/2 \tag{3.4}$$

Uobičajeni zahtjev na logičke sklopove je da imaju veliku brzinu (mali t_p) i malu disipaciju, no ta su dva zahtjeva često u suprotnosti. Smanjivanje disipacije (niži naponi napajanja i/ili niže vrijednosti struja) smanjuju vodljiva svojstva vratiju i time povećavaju t_p . Zbog toga se za mogući način usporedbe logičkih sklopova dobijenih različitim tehnologijama koristi umnožak vremena kašnjenja i disipacije:

$$DP = P_D \cdot t_p \tag{3.5}$$

Tipične vrijednosti napona napajanja, naponskih signala, dozvoljenog šuma (NM), dispacije, umnoška kašnjenja i disipacije (DP) te broja mogućih sklopova koji mogu biti na njih priključeni ($fan-out$) logičkih sklopova koji se danas najčešće koriste prikazani su u tablici 3.7.

TABLE 6-4 Comparison of Logic Families

Family series	TTL			CMOS*		ECL	
	74 LS	74 AS	74 ALS	74 C	74 HC	10K	100K
Nominal supply voltage, V	5	5	5	5	5	-5.2	-4.5
Maximum V_{OH} , V	0.5	0.5	0.5	0.4	0.4	-1.7	-1.7
Minimum V_{OH} , V	2.7	2.7	2.7	4.2	4.2	-0.9	-0.9
Maximum V_{OL} , V	0.8	0.8	0.8	1.0	1.0	-1.4	-1.4
Minimum V_{OL} , V	2.0	2.0	2.0	3.5	3.5	-1.2	-1.2
NM_H , V	0.7	0.7	0.7	0.7	0.7	0.3	0.3
NM_L , V	0.3	0.3	0.3	0.6	0.6	0.3	0.3
Logic swing, V	2.0	2.0	2.0	3.8	3.8	0.8	0.8
Power dissipation per gate, mW	2	20	1	≈0	≈0	24	40
Delay-power product, pJ	10	1.5	4	30	10	2	0.75
Fan-out	100	10	100	>100	>100	10	10

Slika 3.7:

4 IZVEDBE DIGITALNIH SKLOPOVA

Načini realizacije logičkih funkcija u integriranoj tehnici zavise o primjenjenoj tehnologiji. Prvi sklopovi sadržavali su diskretne komponente, a do 1960. godine su integrirane izvedbe bile analogne diskretnim sklopovima.

4.1 DTL (*diode - transistor logic*)

DTL je omogućila izvedbu osnovnog logičkog sklopa - invertera. Ova tehnologija je direktni nastavak diskretne **DL** logike (*diode logic*). Njeno poboljšanje predstavljala je **HTL** (*high threshold logic*), koja je također koristila diode, no koje su imale visoki prag šuma (*noise margins*)

4.2 RTL (*resistor - transistor logic*)

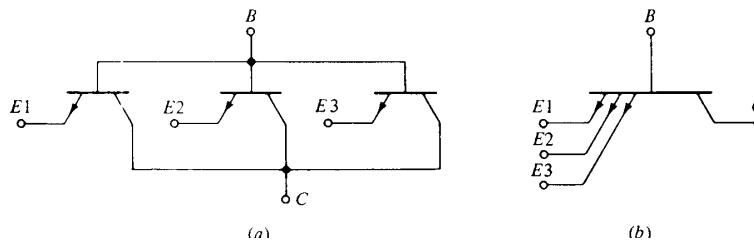
RTL predstavlja jednu od prvih komercijalnih izvedbi, kod koje je dioda zamijenjena s tranzistorom. Karakterizira je izvedba u kojoj je jedan otpornik zajednički kolektorski otpornik za više tranzistora. Iako je brzina rada bila relativno velika, sklopovi su imali veliku disipaciju. "Nadogradnju" ove tehnologije predstavljala je **DCTL** (*direct coupled transistor logic*), čija izvedba je analogna RTL logici, ali nema otpora u krugu baze tranzistora.

4.3 TTL (**T²L**) (*transistor - transistor logic*)

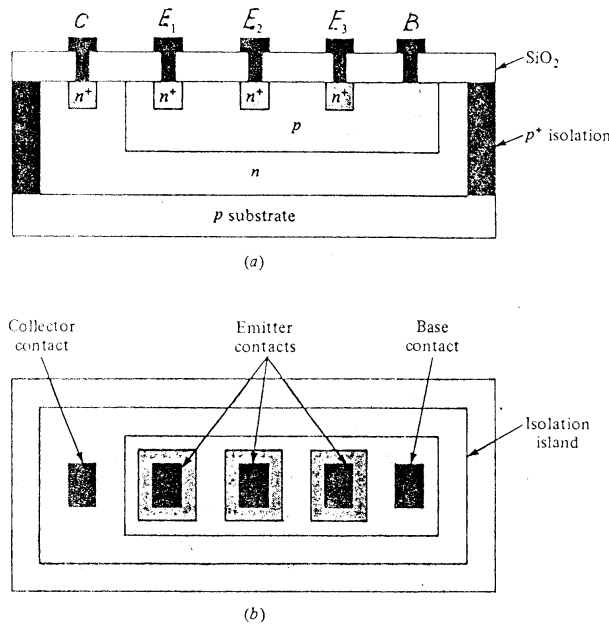
Povećanjem stupnja integracije – od SSI (*small scale integration*), preko MSI (*medium*), LSI (*large*) do današnjih VLSI (*very large*) i VVLSI (*veryvery*)

large) – mijenjale su se i izvedbe, ali i karakteristike danih logičkih sklopova. Tehnološki se pokazalo moguće, a ekonomski prihvatljivo i isplativo izrađivati čipove s velikim brojem istih elemenata (tranzistora).

U razdoblju SSI tehnologije (1966.–1985.) TTL je bila najčešće korištena logika. Njenu osnovu čini multiemitterski transistor (čime se ostvaruje ušteda prostora); pri tome broj emitera može biti i veći od 60 (slike 4.1 i 4.2). U odnosu na DTL logiku je TTL logika dvostruko brža (DTL ~ 25 ns; TTL ~ 10 – 13 ns).



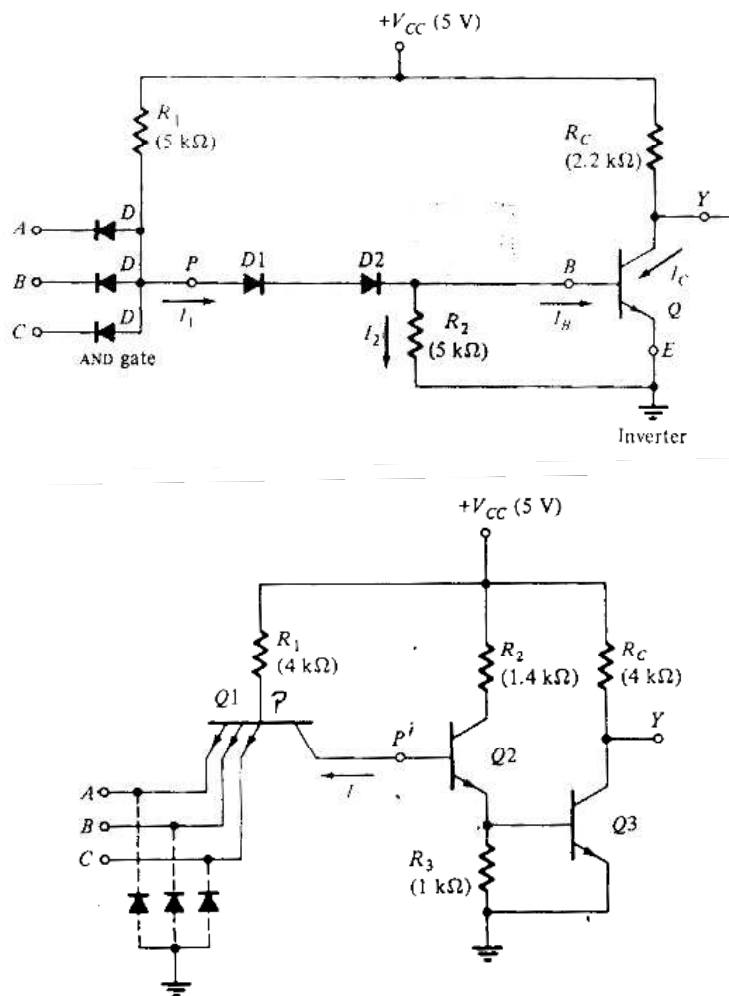
Slika 4.1: Tri tranzistora sa zajedničkim kolektorom i bazom i njihov ekvivalentni multiemitterski tranzistor



Slika 4.2: Presjek multiemitterskog tranzistora

Osnovni element TTL logike je NAND. Na slici 4.3 prikazane su izvedbe NAND-a u DTL i TTL logici: ulazne diode D zamijenjene su B–E spojevima

multiemitskog tranzistora Q_1 , dioda D_1 zamijenjena je C–B spojem istog tranzistora Q_1 , a diodu D_2 zamjenjuje tranzistor Q_2 .



Slika 4.3: NAND u DTL i TTL izvedbi

4.4 ECL (*emitter-coupled logic*)

Kod ECL logike se signali dovode na baze tranzistora, čiji emiteri su međusobno povezani. To su najbrži logički sklopovi, jer se kod njih tranzistori ne dovode u područje zasićenja. Karakterizira ih, međutim, veća potrošnja (disipacija) te veće dimenzije (SSI i MSI izvedbe). Ulazni signali

se uspoređuju s određenim referentnim naponom, a uvijek sadrže 2 komplementarna izlaza.

ECL izvedba OR/NOR logičkog sklopa prikazana je na slici 4.4.

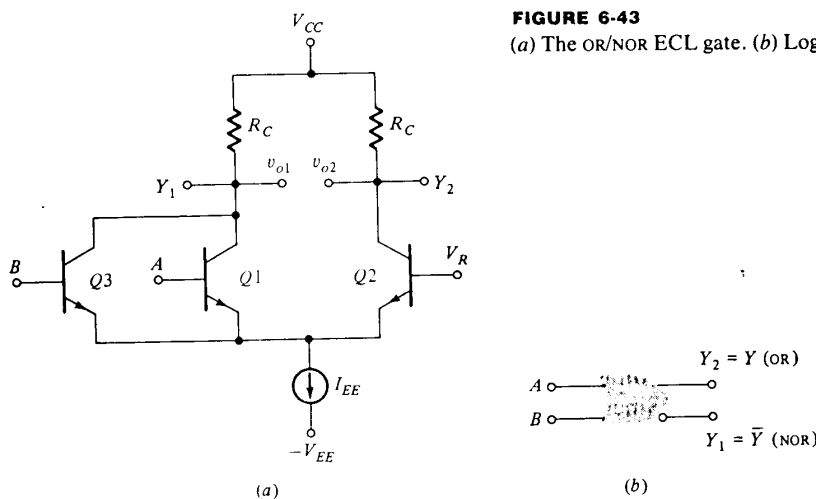


FIGURE 6-43
(a) The OR/NOR ECL gate. (b) Logic symbol.

Slika 4.4: OR/NOR u ECL izvedbi

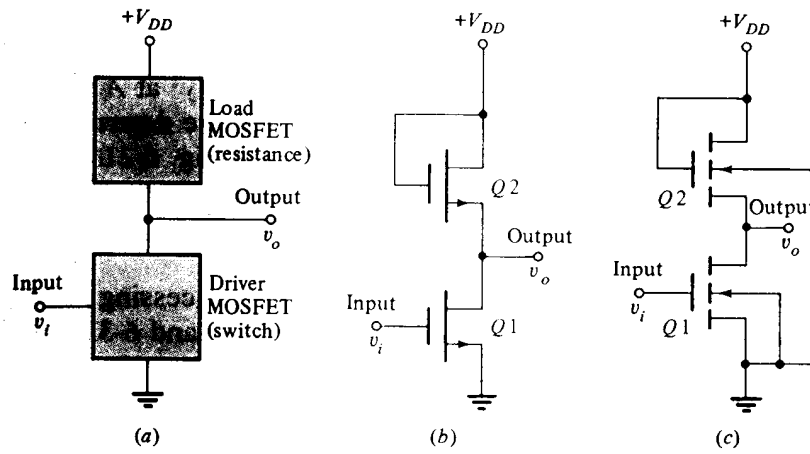
4.5 MOSFET logika

U današnjim LSI, VLSI i VVLSI tehnologijama dominira korištenje MOSFET tranzistora. Prvi logički sklopovi (inverteri) koristili su NMOS tranzistore (*enhancement* ili *depletion* tipa) – pri tome se u slučaju da je opterećenje *depletion*-NMOS dobiva oštiji prijelaz između dva logička stanja. (slike 4.5 i 4.6). Na slikama 4.7 i 4.8 prikazane su realizacije NAND i NOR vrata s MOSFET tranzistorima.

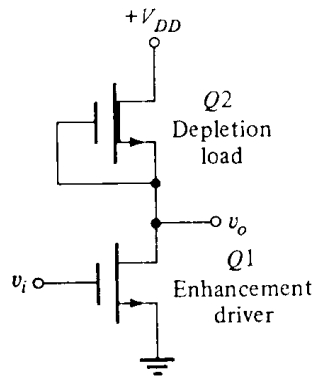
MOSFET logički sklopovi su karakterizirani vrlo malom dispacijom (~ 1 nW/vrata).

4.6 CMOS (*complementary MOS*)

CMOS logika koristi NMOS i PMOS tranzistore, a njena bitna prednost je da praktički nema statičke disipacije. U praksi sklopovi koriste široki opseg vrijednosti napona napajanja. To su, međutim, relativno "spori" sklo-



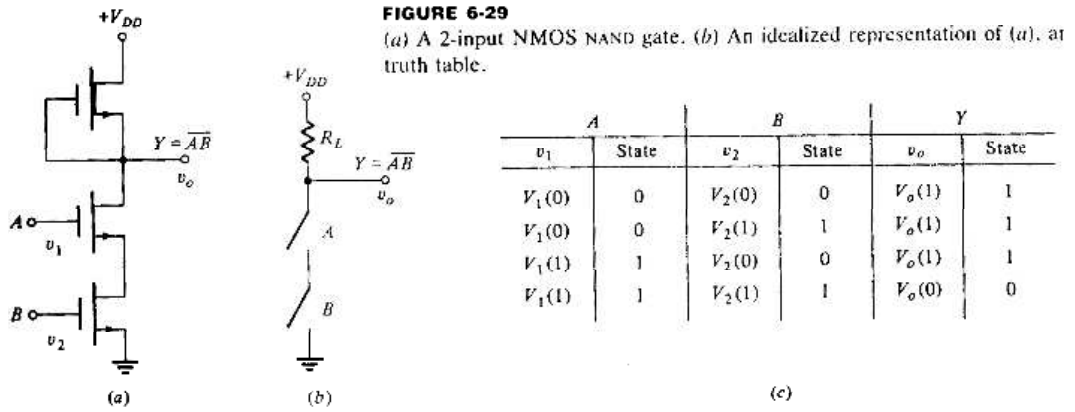
Slika 4.5: MOSFET inverter: (a) osnovna struktura, (b) shema sklopa (c) shema s naznačenim spojevima između substrata



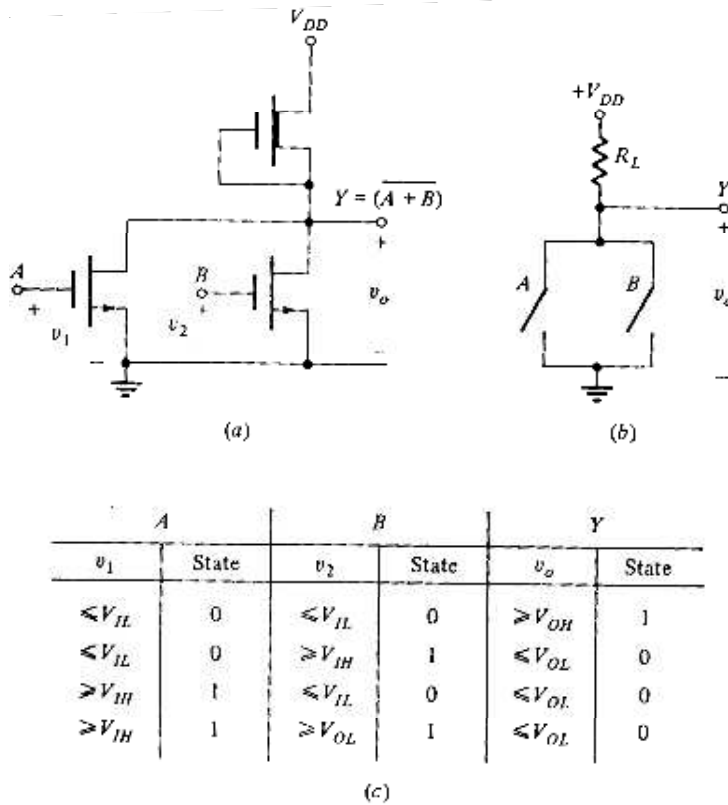
Slika 4.6: MOSFET inverter s depletion–NMOS tranzistorom kao opterećenjem

povi (zbog PMOS tranzistora), a povećana brzina rada postiže se korištenjem **ACL** (*advanced CMOS logic*) tehnologije.

CMOS strukture karakterizira velika gustoća pakovanja, koja se ostvaruje VVLSI (*very very large*) stupnjem integracije. Na taj način je postignuta značajna redukcija duljine kanala, koja je danas $\leq 0.15 \mu\text{m}$.

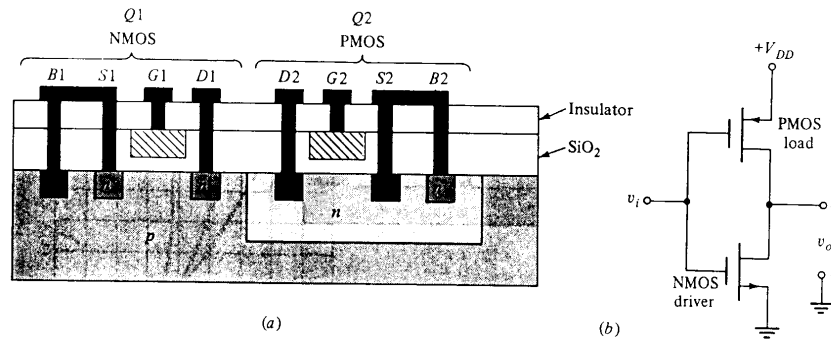


Slika 4.7: NMOS NAND logička vrata

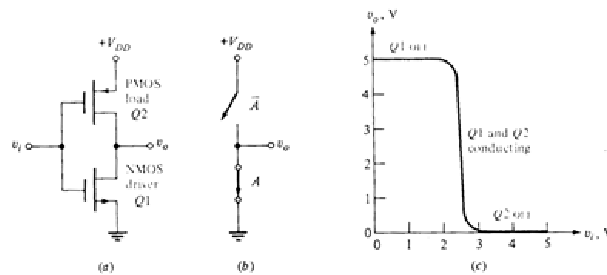


Slika 4.8: NMOS NOR logička vrata

Za razliku od BJT, MOS tranzistori su karakterizirani velikim ulaznim otporima, što omogućuje 'spremanje naboja' (*charge storage*), tj. njihovu primjenu u memorijskim elementima.



Slika 4.9: CMOS struktura



Slika 4.10: CMOS inverter

Današnji CMOS logički sklopovi najčešće koriste 'klasičnu' CMOS tehnologiju. U nekim slučajevima zamjenjuju ih:

- *pseudo-NMOS* (ulazni signal dovodi se samo na vrata NMOS-a, a vrata PMOS-a su spojena na masu; to znači da je za svaka dodatna ulazna vrata potreban samo jedan dodatni NMOS transistor, čime se smanjuje površina sklopa i povećava brzina rada);

- *PTL (pass-transistor logic)*, (korištenjem NMOS tranzistora kao prekidača između ulaza i izlaza upravlja se prolazom (*pass*) signala, pri čemu treba biti ispunjeno u svakom trenutku da su svaki od ulaza te izlaz kratko spojeni na napon napajanja ili masu);

- *dynamic logic*, (logički sklopovi koji rade u impulsnom, a ne statičkom, režimu, a kod kojih se signal pohranjuje u parazitskim kapacitetima – zbog smanjivanja naboja tijekom vremena, takvi krugovi zahtijevaju periodičko

obnavljanje (*refreshment*) korištenjem taktnog (*clock*) signala).

Iako su danas, zbog razvoja tehnika integracije sve do VLSI nivoa, u logičkim krugovima dominantni MOSFET tranzistori, sklopovi s bipolarnim (BJT) tranzistorima su također prisutni u velikoj mjeri. Tako se TTL logika i dalje koristi za SSI i MSI sklopove, dok ECL logički sklopovi imaju najveće brzine rada, pa ta njihova prednost često nadilazi njihove nedostatke (relativno velika površina, značajna disipacija).

Daljnji razvoj integriranih poluvodičkih logičkih sklopova ide prema:

BiCMOS (*bipolar and CMOS*) tehnologiji, koja kombinira velike brzine BJT s malom disipacijom CMOS-a. Trenutno se BiCMOS razvija za posebne primjene, uključujući memorijske čipove, gdje njihove karakteristike opravdavaju kompleksnu i skupu tehnologiju njihove proizvodnje.

GaAs (*Gallium Arsenide*) tehnologiji, u kojoj se umjesto silicija koristi GaAs, kod kojeg su pokretljivosti nosilaca naboja znatno veće (a time veće i brzine rada). Za sklopove s GaAs se tek očekuje jače komercijalno korištenje.