Univerzitet u Tuzli
Fakultet elektrotehnike

Dizajn logičkih sklopova na bazi osnovnih/složenih logičkih kola

Dr. Sc. Asmir Gogić, vanr. prof.

Tuzla, 2021.

Cilj i ishod predavanja

- ***Dizajn logičkih sklopova*** na bazi osnovnih i složenih logičkih kola.
- Električne AC i DC karakteristike logičkih sklopova
- Arhitektura ***FPGA i CPLD*** čipova.
- Jezici za opis hardvera ***HDL jezici***.
- Sintaksa ***VHDL jezika***.
- Sinteza i simulacija HDL koda.
- ***Programiranje i dizajn*** logičkih sklopova na bazi FPGA i CPLD čipova.

- Predznanje: Boolova algebra, logička kola, FF, registri, brojači, multiplekseri...

- Predispitne aktivnosti: ***Zadaće 2x po 30 bodova.***
- Završni ispit: projekat ***40 bodova***

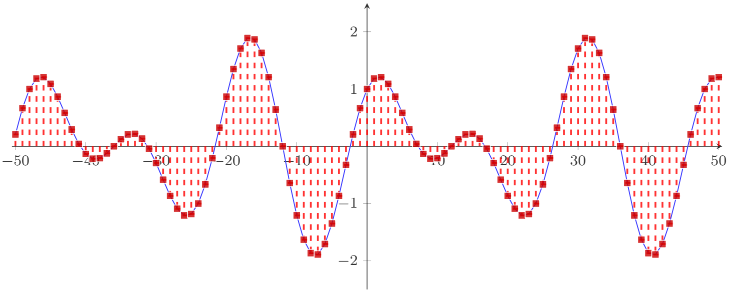
Struktura kursa

Predavanja		
Stelekt	Ponedjeljak	08:00 - 11:00
AV - Vježbe		
Stelekt	Petak	08:00 - 09:00
LV - Vježbe		
Stelekt Lab	Petak	09:00 - 11:00

- ***Prisustvo*** na predavanjima je ***obavezno***!
- Prije ulaska na predavanje ***mobilne aparate isključiti***.
- ***Studenti koji zakasne*** na tekući čas predavanja moraju ***sačekati početak narednog časa*** kako bi mogli prisustvovati.
- Maksimalno ***tri izostanka*** sa predavanja se mogu ***tolerirati***.
- Studenti koji nisu u mogućnosti prisustvovati ispitu obavezni su u roku od 7 dana obavijestiti nastavnika ili asistenta, o razlogu izostanka sa ispita (*putem email-a ili direktno*).

Fizički procesi

- Fizički procesi imaju za posljedicu ***transformacije materije ili energije***.
- Svaki fizički proces karakterizira jedan ili viši parametara.
- ***Mjerljive parametre*** nazivamo veličinama koje mogu biti ***diskretne ili kontinualne*** prirode.
- ... s ciljem analize fizičkih procesa uvodimo pojam ***signala i sistema***.
- ***Signal*** je fizički ***nosilac informacije*** koji obično mapira prostor ili vrijeme u neku mjerljivu veličinu.
- ***Informacija*** je zapisana u obliku promjene nekog od parametara: ***amplituda, faza ili frekvencija***.



Signali i sistemi

- ***Sistem*** je ***matematički model fizičkog procesa*** koji definira pravilo prema kojem se vrši transformacija ulaznog signala (pobude) u izlazni signal (odziv).
- Signali a i sistemi po svojoj prirodi mogu biti ***kontinualni ili diskretni***.
- ***Kontinualni signali*** su signali kod kojih je nezavisno promjenljiva kontinualna veličina tj. iz skupa realnih brojeva.
- ***Diskretni signali*** se dobiju iz kontinualnih u procesu uzorkovanja a njihova nezavisno promjenljiva je diskretna tj. iz skupa cijelih brojeva.
- Uzorkovanje predstavlja prvi korak u procesu ***analogno-digitalne konverzije ADC*** signala.
- ***Analogni signali*** su ustvari kontinualni signali kod kojih je amplituda ograničena tj. pripada nekom intervalu (a, b).
- ... digitalni signali su diskretni signali koji mogu poprimiti samo konačan broj vrijednosti.
- Proces konverzije analognih signal u digitalne uključuje sljedeće korake:
  - ***Uzorkovanje*** - koje predstavlja diskretizaciju signala po vremenu,
  - ***Kvantizacija*** - koja predstavlja diskretizaciju signala po amplitudi i
  - ***Kodiranje*** - predstavljanje signala u odgovarajućem brojnom sistemu

AD konverzija

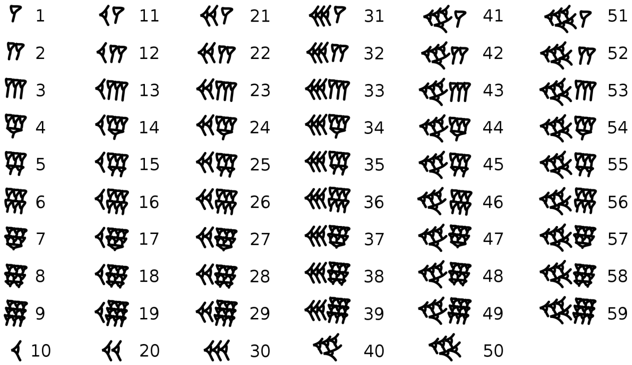
- ***U procesu AD konverzije*** jedini gubitak informacije je evidentan prilikom kvantizacije.
- U procesu uzorkovanja ***neće doći do gubitka informacije*** ukoliko je zadovoljena *Nyquistova* teorema.

Neka kontinualni signal $x(t)$ ima ograničen spektar W. Signal $x(t)$ se može rekonstruirati na osnovu svojih uzoraka $x(nTs)$ ukoliko je period uzorkovanja Ts signala odabran tako da važi $\frac{1}{Ts} \geq 2W$

- U procesu kodiranja diskretizovane vrijednosti ***amplitude signala*** se zapisuju u nekom od brojnih sistema.
- ...Najčešće je to binarni brojni sistem gdje se logička stanja 0 i 1 ***predstavljaju sa naponskim nivoima*** koji mogu biti npr (0V i 5V); (0V i 3.3V); (-5V i 5V) itd.
- U procesu detekcije naponskih nivoa, ***detektori imaju određeni prag tolerancije*** prema kojem mogu i niže odnosno više naponske nivo tretirati kao logičku jedinicu odnosno nulu tvz tvrda logika ***hard decision***
- Naprednije verzije detektora koriste meku logiku ***soft decision*** na bazi vjerovatnoća.

Brojni sistemi

- ... ***prvi brojni sistem*** potječe iz Babilona 2000 BC i imao je bazu 60.



Brojni sistemi

- ***Dekadski brojni sistem*** koji danas koristimo razvijen je u Indiji u petom stoljeću nove ere
- ...oko 10-tog stoljeća "došao" je u Europu preko Arabije, gdje je dekadski brojni sistem proširen sa cifrom nula.
- ***Svaki brojni sistem*** sastoji se ***baze i cifara***.
- Brojni sistemi koji danas koristimo su ***pozicioni brojni sistemi***.

Cijeli broj N u pozicionom brojnom sistemu sa bazom B i ciframa c_i se može predstaviti na sljedeći način:

$$N = \sum_{i=0}^{n-1} c_i B^i$$

Brojni sistemi

- Neki od najčešće korištenih pozicionih brojnih sistema su:

Baza	Brojni sistem	Cifre
2	Binarni	0,1
3	Ternarni	0,1,2
5	Kvinarni	0,1,...,4
8	Oktalni	0,1,...,7
10	Dekadski	0,1,...,9
12	Duodekani	0,1,...,9,α,β
16	Heksadekadski	0,1,...,9,A,...,F

Brojni sistemi

- ***Konverzija broja*** X iz dekadskog brojnog sistema u neki drugi pozicioni brojni sistem sa bazom B izvodi se sukcesivnim dijeljenjem broja X sa bazom B.

- Izvršiti konverziju broja 23_{10} zapisanog u dekadskom brojnom sistemu sistema u broj zapisan u binarni brojnom sistemu.

$$
\begin{array}{rl}
23 : 2 = 11 & \text{ostatak } 1 \\
11 : 2 = 5 & \text{ostatak } 1 \\
5 : 2 = 2 & \text{ostatak } 1 \\
2 : 2 = 1 & \text{ostatak } 0 \\
1 : 2 = 0 & \text{ostatak } 1
\end{array}
$$

zapis cifara

Brojni sistemi

Brojni sistemi

- Sa ***tehničkog aspekta*** gledano ako bi smo željeli predstaviti n brojnih mjesta u brojnom sistemu sa bazom B, neophodno nam je n sklopova sa B stanja. Ukupan broj različitih diskretnih stanja je

$$v = B \times n \tag{1}$$

sa druge strane ukupan broj kombinacija koje možemo imati je

$$N = B^n \tag{2}$$

Na osnovu prethodne dvije relacije možemo odrediti ***ukupan broj različitih diskretnih stanja*** u funkciji od N

$$v = \ln N \frac{B}{\ln B} \tag{3}$$

Deriviranjem izraza 3 i izjednačavanjem sa nula, možemo odrediti vrijednost baze B za koju izraz 3 najmanji, što ujedno predstavlja i ***optimalnu bazu sa stanovišta zapisa***, računske kompleksnosti i broja različitih stanja.

$$\frac{dv}{dB} = \ln N \cdot \frac{\ln B - 1}{(\ln B)^2} = 0$$

$$\ln B - 1 = 0 \quad \Rightarrow \quad B = e = 2.7178$$

Brojni sistemi

- ***Sekvence binarnih cifara*** se mogu koristiti za predstavljanje karaktera i proizvoljnih simbola.

$$\begin{array}{cccc} 6 & 9 & 3 & \mathrm{B} \\ \overbrace{0\ 1\ 1\ 0} & \overbrace{1\ 0\ 0\ 1} & \overbrace{0\ 0\ 1\ 1} & \overbrace{1\ 0\ 1\ 1} \end{array}$$

$$\begin{array}{cccccc} 0 & \underbrace{1\ 1\ 0}_{6} & \underbrace{1\ 0\ 0}_{4} & \underbrace{1\ 0\ 0}_{4} & \underbrace{1\ 1\ 1}_{7} & \underbrace{0\ 1\ 1}_{3} \end{array}$$

- Ukupan ***broj različitih simbola*** koji se može prikazati sa binarnom sekvencom od **n** bita je

$$N = 2^n$$

- Generalno, za predstavljanje ***M različitih simbola*** potrebno je koristi $k = \lceil \log_2 M \rceil$ bita.
- Najčešće korišteni ***binarni kodovi*** za prijenosa informacija su:
  - Kod 8421
  - Kod 2421
  - Kod XS-3 (kod više 3)
  - Bikvinarni (5043210) kod
  - Gray-ov kod
  - ASCII (7 bitni) znakovni kod
  - EBCDIC (8 bitni) znakovni kod
  - UTF8 (*Universal Character Set Transformation Format*) 16 bitni kod

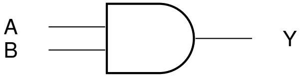
Logičke funkcije

- Sposobnost obavljanja kompleksnih aritmetičkih i logičkih operacija digitalnih sklopova zasniva se na izvođenju jednostavnih binarnih operacija.
- 1847 godine ***George Boole*** definirao je matematičke ***osnove logičke algebre***.
- Opšti interes za ovom granom matematike pojavio se 1938 godine kada je ***Claude Elwood Shannon*** pokazao da se ista može koristiti za analizu elektro-mehaničkih relejnih sklopova.
- ***Bool-ova algebra*** se zasniva na ***tvrdnji ili iskazu***, koja je tačna ili netačna.
- Tvrdnja se može zapisati u obliku matematičke funkcije koja je izražena preko logičkih varijabli.
- ***Logičke varijable*** mogu imati jednu od dvije moguće vrijednosti **tačno** (logička jedinica) ili **netačno** (logička nula).
- Funkcija logičkih promjenljivih naziva se ***logička funkcija***.
- Svaka logička funkcija može se opisati pomoću tri osnovne ***logičke operacije AND, OR i NOT***.
- ***Ponašanje logičkih sklopova*** opisujemo logičkom funkcijom koju možemo zapisati analitički ili preko tablice istinitosti.

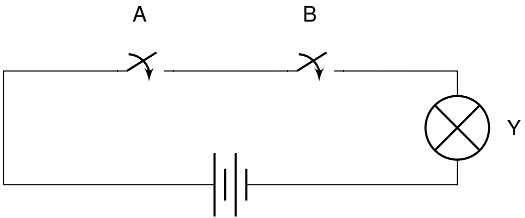
Logička kola

- Logičko kolo ***AND*** ili "I" ili konjukcija ***obavlja logičku operaciju množenja*** i za slučaj dvije ulazne logičke varijable analitički izraz je

$$Y = A \cdot B \tag{4}$$

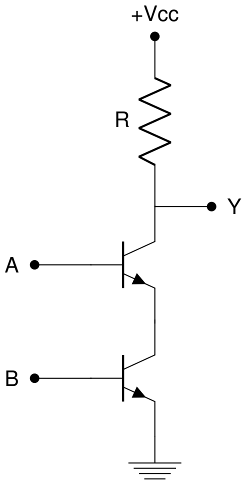


- Model logičkog kola AND sastoji se od serijske veze dva prekidača, jednog izvora i svjetlosnog indikatora.



Logička kola

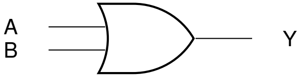
- Logičko kolo možemo implementirati na jednostavan način korištenjem dva BJT



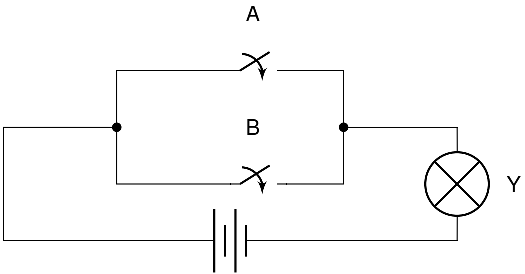
Logička kola

- Logičko kolo ***OR*** ("ILI" ili disjunkcija) ***obavlja logičku operaciju sabiranja*** i za slučaj dvije ulazne logičke varijable analitički izraz je

$$Y = A + B \tag{5}$$

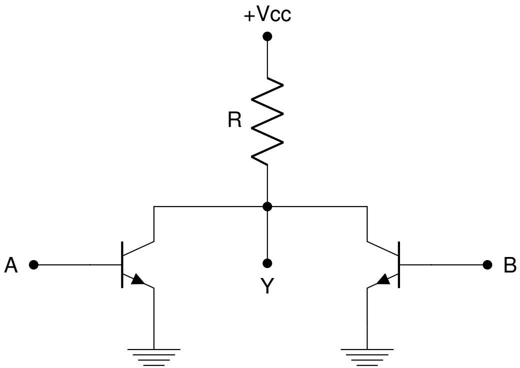


- Model logičkog kola OR sastoji se od paralelne veze dva prekidača, jednog izvora i svjetlosnog indikatora.



Logička kola

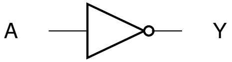
- Logičko kolo možemo implementirati na jednostavan način korištenjem dva BJT



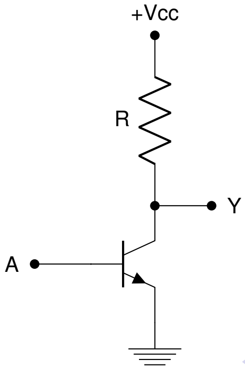
Logička kola

- Logičko kolo ***NOT*** ("NE" ili negacija, invertor) ***obavlja logičku operaciju negiranja*** jedne logičke varijable

$$Y = \overline{A} \tag{6}$$



- Logičko kolo možemo implementirati na jednostavan način jednim BJT



Boolova algebra

- ***Ponašanje logičkih kola*** se može opisati ***Boolovom algebrom*** koja se bazira na konzistentnom matematičkom sistemu.
- U osnovi, Boolova algebra ***uključuje skup aksioma*** (postulata) nad kojima su definirani operatori.
  - Skupu dva ili više članova (elemenata)
    $s = a, b, c, \ldots$
  - Operatorima "+" i "·" čija primjena nad elementima skupa s generira nove članove koji su također iz skupa s
  - Skupu aksioma.
- ... 1904 godine ***Edward Vermilye Huntington*** je definirao skup od šest aksioma od kojih su dva sadržana u prethodnim stavovima

Boolova algebra

- ***AKSIOM A1:*** Postoji neutralan element "0" i "1" u odnosu na operacije "+" i "·" takav da vrijedi
  - $A + 0 = A$
  - $A \cdot 1 = A$
- ***AKSIOM A2:*** Za svaki element "A" skupa s ($A \in s$) postoji element $\overline{A}$ koji je također is s takav da vrijedi
  - $A + \overline{A} = 1$
  - $A \cdot \overline{A} = 0$
- ***AKSIOM A3:*** Operatori "+" i "·" su komutativni
  - $A + B = B + A$
  - $A \cdot B = B \cdot A$
- ***AKSIOM A4:*** Operatori "+" i "·" su distributivni jedan preko drugog
  - $A \cdot (B + C) = A \cdot B + A \cdot C$
  - $A + B \cdot C = (A + B)(A + C)$
- Iz aksioma se može izvesti niz ***teorema*** od kojih su najznačajnije:
- ***TH1:*** Identiteta
  - $A + 1 = A$
  - $A \cdot 0 = 0$
- ***TH2:*** Idempotencije
  - $A + A = A$
  - $A \cdot A = A$

Boolova algebra

- ***TH3:*** Involucije
  - $\overline{\overline{A}} = A$
- ***TH4:***
  - $A + \overline{A} \cdot B = A + B$
  - $A \cdot (\overline{A} + B) = A \cdot B$
- ***TH5:*** Asocijacije
  - $(A + B) + C = A + (B + C)$
  - $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
- ***TH6:*** DeMorganov Zakon
  - $\overline{A + B} = \overline{A} \cdot \overline{B}$
  - $\overline{A \cdot B} = \overline{A} + \overline{B}$
- ***TH7:*** Zakon apsorbcije
  - $A + A \cdot B = A$
  - $A \cdot (A + B) = A$
- ***TH8:*** Zakon simplifikacije
  - $AB + A \cdot \overline{B} = A$
  - $(A + B) \cdot (A + \overline{B}) = A$
- ***TH9:*** Generalizirani DeMorganov Zakon
  - $\overline{A + B + C + \ldots} = \overline{A} \cdot \overline{B} \cdot \overline{C} \ldots$
  - $\overline{A \cdot B \cdot C \cdot \ldots} = \overline{A} + \overline{B} + \overline{C} + \ldots$

Boolove funkcije

- Dvije ili više ***logičkih funkcija*** mogu imati različite analitičke izraze ali iste vrijednosti (tablice istinitosti).
- Kreiranja logičke funkcije možemo izvesti na dva načina i to preko:
  - sume potpunih proizvoda,
  - proizvoda potpunih suma.

A	B	$A\overline{B}$	$\overline{A}B$	f	P_i	S_i
0	0	0	0	0 = v_0	$P_0 = \overline{A} \cdot \overline{B}$	$S_0 = A + B$
0	1	0	1	1 = v_1	$P_1 = \overline{A} \cdot B$	$S_1 = A + \overline{B}$
1	0	1	0	1 = v_2	$P_2 = A \cdot \overline{B}$	$S_2 = \overline{A} + B$
1	1	0	0	0 = v_3	$P_3 = A \cdot B$	$S_3 = \overline{A} + \overline{B}$

- ***Tablični zapis*** vrijednosti funkcije postaje ***nepraktičan*** sa porastom broja logičkih varijabli i kompleksnošću logičkog izraza. S Nekanonski oblik logičke funkcije se svodi na standardni zapis logičke funkcije, upotrebom aksioma i pravila na sljedeći način:
  - Ako formiramo DNF, nepotpunim članovima dodajemo varijable koje nedostaju u obliku $(X + \overline{X})$.
  - Ako formiramo KNF, nepotpunim članovima dodajemo varijable koje nedostaju u obliku $X\overline{X}$.

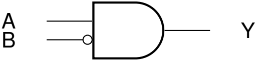
Boolove funkcije

- ***Dualna funkcija*** ili ***komplement funkcije*** je logička funkcija koja ima komplementarne vrijednosti zadate funkcije.
- Za dvije logičke varijable A i B možemo kreirati 2^{2^2} različitih logičkih funkcija.

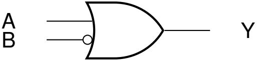
A B	0 0	0 1	1 0	1 1	Naziv	Funkcija
f_0	0	0	0	0	NULA	0
f_1	0	0	0	1	AND	$A \cdot B$
f_2	0	0	1	0	INHIBICIJA	$A \cdot \overline{B}$
f_3	0	0	1	1	IDENTITET	A
f_4	0	1	0	0	INHIBICIJA	$\overline{A} \cdot B$
f_5	0	1	0	1	IDENTITET	B
f_6	0	1	1	0	EXOR	$A \cdot \overline{B} + \overline{A} \cdot B$
f_7	0	1	1	1	OR	$A + B$
f_8	1	0	0	0	NOR	$\overline{A + B}$
f_9	1	0	0	1	EKVIVALENCIJA	$A \cdot B + \overline{A} \cdot \overline{B}$
f_{10}	1	0	1	0	KOMPLEMENT	$\overline{B}$
f_{11}	1	0	1	1	IMPLIKACIJA	$A + \overline{B}$
f_{12}	1	1	0	0	KOMPLEMENT	$\overline{A}$
f_{13}	1	1	0	1	IMPLIKACIJA	$\overline{A} + B$
f_{14}	1	1	1	0	NAND	$\overline{A \cdot B}$
f_{15}	1	1	1	1	JEDAN	1

Ostala logička kola

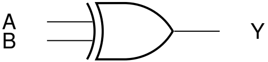
- ***INHIBICIJA*** $Y = A \cdot \overline{B}$



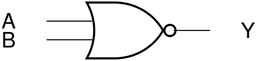
- ***IMPLIKACIJA*** $Y = A + \overline{B}$



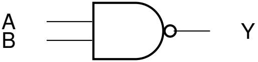
- ***XOR*** $Y = A\overline{B} + \overline{A}B$



- ***NOR*** $Y = \overline{A + B}$



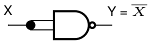
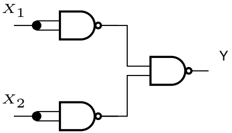
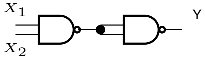
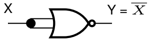
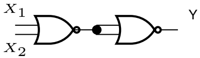
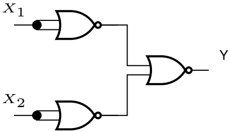
- ***NAND*** $Y = \overline{A \cdot B}$



Konverzija logičkih funkcija u NOR i NAND oblik

- ***NOR i NAND*** logičke funkcije su ***univerzalne*** jer omogućuju realizaciju NOT, OR i AND logičkih funkcija.
- Koriste se kod ***realizacije*** odnosno implementacije logičkih funkcija.
- ***Transformacija logičke*** funkcije u logičku funkciju pogodnu za implementaciju preko NOR i NAND kola izvodi se ***dvostrukom negacijom***.
- Drugi način implementacije logičke funkcije f sa NOR i NAND logičkim kolima svodi se na direktnu zamjenu osnovnih logičkih kola
- Procedura implementacije logičke funkcije f sa NAND kolima je sljedeća:
  - Logičku funkciju f zapisati u DNF
  - Dva puta negirati funkciju $\overline{\overline{f}}$
  - Primijeniti DeMorganov teorem u obliku $\overline{A+B} = \overline{A} \cdot \overline{B}$ na izraz pod unutrašnjom negacijom.
- Procedura implementacije logičke funkcije f sa NOR kolima je sljedeća:
  - Logičku funkciju f zapisati u KNF
  - Dva puta negirati funkciju $\overline{\overline{f}}$
  - Primijeniti DeMorganov teorem u obliku $\overline{A \cdot B} = \overline{A} + \overline{B}$ na izraz pod unutrašnjom negacijom.

Konverzija logičkih funkcija u NOR i NAND oblik

Kolo	NOT	OR	AND
NAND		 Y = $X_1 + X_2$	 Y = $X_1 \cdot X_2$
NOR		 Y = $X_1 + X_2$	 Y = $X_1 \cdot X_2$

Minimizacija logičkih funkcija

- ***Svođenje zapisa logičke funkcije*** na minimalni broj logičkih operacija naziva se ***minimizacija***.
- ... Da li postoji pravilo koje će uvijek osigurati svođenje logičke funkcije na minimalni zapis?
- Kod ***algebarskog pristupa***, svođenje se sastoji u primjeni aksioma i teorema Boolove algebre.
- Za algebarski pristup ***ne postoji definiran slijed operacija*** čija primjena osigurava dobijanje minimalnog zapisa logičke funkcije.
- ***Karnaugh-ove mape*** predstavljaju grafički metod minimizacije logičke funkcije a bazira se na tablicama kombinacija logičke funkcije.
- Karnaugh-ova mapa ***se sastoji*** od onoliko polja koliko data funkcija ima standardnih produkt-članova.
- Ako logička funkcija ima N logičkih promjenljivih tada će Karnaugh-ova mapa ima 2^N polja.
- Jedan član iz tablice kombinacija zauzima jedno polje i naziva se *minterm*.

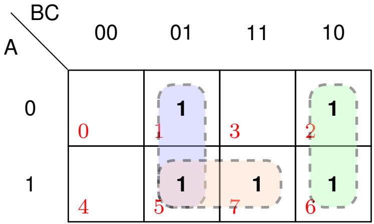
Minimizacija logičkih funkcija

- ***Postupak optimizacije*** logičke funkcije grafičkim metodom sastoji se iz sljedećih koraka:
  - ***Grupisati*** 2^k ***susjednih članova*** u Karnaugh-ovoj mapi tako da čine geometrijsko oblik kvadrat ili pravougaonik.
  - Susjedni članovi su oni koje se ***međusobno dodiruju*** kao i članovi tipa $m_o \longleftrightarrow m_8, m_o \longleftrightarrow m_2, m_o \longleftrightarrow m_{10}$ itd. Drugačije rečeno članovi koji imaju istu vrijednost promjenljive.
  - Grupe od 2^k susjednih članova ***svode se na grupe od*** $N-k$ ***varijabli*** (N je ukupan broj varijabli) na način da se eliminiraju varijable čije se vrijednosti u grupama mijenjaju.
- Za nepotpuno specificirane funkcije (nestandardne zapise logičke funkcije) potrebno je ***kreirati tablicu stanja*** pa onda na osnovu iste popuniti i minimizirati Karnaugh-ovu mapu.

Minimizacija logičkih funkcija

- Optimizirati logičku funkciju $F = AB + A\overline{B}C + B\overline{C} + \overline{AB}C + AB\overline{C}$ koristeći grafički metod

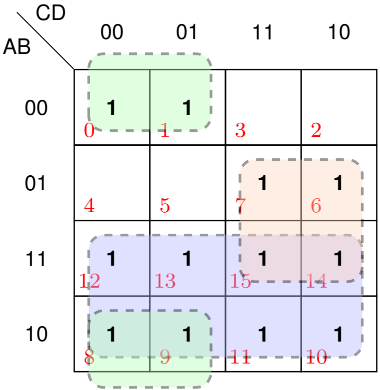
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



$$F_{min} = \overline{B}C + AC + B\overline{C}$$

Minimizacija logičkih funkcija

- Optimizirati logičku funkciju $F = \sum(0, 1, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$ koristeći grafički metod.



$F_{min} = A + BC + \overline{BC}$

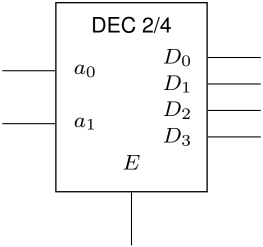
Kombinacijski sklopovi i modularne mreže

- ***Svaka logička funkcija*** se može implementirati sa tri osnovna logička kola AND, OR i NOT ili pomoću NAND i NOR kola.
- ***Za velike i kompleksne sisteme*** ovakav pristup postaje ne efikasan pa stoga koristimo standardne ili specijalne module.
- Rješenje: izdijeliti sistema na manje cjeline pri čemu je ***glavni kriterij*** jasno definirana logička funkcija i broj ulaznih/izlaznih signala.
- Procedura realizacije logičke funkcije pomoću standardnih modula sa stanovišta cijene je ***opravdana i vremenski kraća***.
- ***Brzina rada sklopa*** baziranog na standardnim modulima je nešto manja u odnosu na gotovo rješenje koje u potpunosti integrirano.
- U osnovi postoje dvije grupe standardnih modula
  - sklopovi koji obavljaju aritmetičko operacije,
  - sklopovi koji obavljaju univerzalne funkcije:
    - Dekoder,
    - Multiplekser,
    - Permanentna memorija,
    - Programabilno logičko polje (FPGA).
- S obzirom na funkciju, standardni moduli se mogu klasificirati na
  - Moduli sa fiksnom funkcijom,
  - Moduli sa programabilnom funkcijom.

Dekoder

- Predstavlja ***kombinacijski logički sklop*** koji za svaku binarnu kombinaciju na ulazu ima jedan izlaz koji je u stanju logičke jedinice.

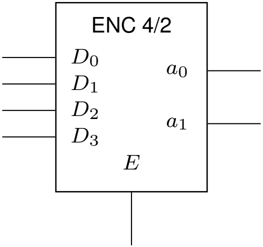
E	A	B	D_0	D_1	D_2	D_3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



Koder

- Predstavlja ***kombinacijski logički sklop*** koji za svaki ulaz generira jednu binarnu kombinaciju na izlazu. Ukoliko koder ima M ulaza broj izlaznih linija N mora zadovoljavati relaciju $2^N >= M$

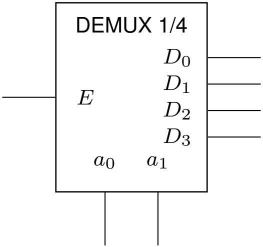
E	D_0	D_1	D_2	D_3	A	B
0	x	x	x	x	Z	Z
1	1	0	0	0	0	0
1	0	1	0	0	0	1
1	0	0	1	0	1	0
1	0	0	0	1	1	1



Demultiplekser

- Predstavlja ***kombinacijski logički sklop*** koji prosljeđuje signal sa ulaza na 2^N izlaza (N predstavlja broj kontrolnih linija). Može se realizovati sa dekoderom pri čemu se za ulaz uzme *ENABLE*.

$$D_i = \begin{cases} 1, & ako\ je\ a_i = 1\ \ i\ \ E = 1 \\ Z, & ostalo \end{cases}$$

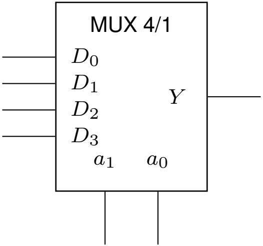


Demultiplekser

- Predstavlja ***kombinacijski logički sklop*** koji informaciju na jednom od 2^N ulaza prosljeđuje na jedan izlaz (N predstavlja broj kontrolnih linija). Naziva se još i selektor podataka a najčešće se koristi za implementaciju složenih logičkih kola te čini osnovnu komponentu FPGA ćelije.

$$Y = \begin{cases} D_i, & E = 1 \\ Z, & ostalo \end{cases}$$

E	A	B	Y
0	x	x	Z
1	0	0	D_0
1	0	1	D_1
1	1	0	D_2
1	1	1	D_3



Literatura

- RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability, 1st Editon by Pong P. Chu, 2006.
- Digital Systems Design Using VHDL 2 nd Edition, by Charles H. Roth, Jr. and Lizy Hurian John, Thomson, 2007.
- The Designer's Guide to VHDL, Third Edition, Peter J. Ashenden, 2008.
- U. Peruško, *Digitalna elektronika*, ŠK Zagreb 1995.
- S. Tešić, *Integrisana digitalna elektronika*, NK Beograd 1990.
- N. Nosović, *Uvod u digitalne računare*, ETF Sarajevo 2003.