

LABORATORIJSKE VJEŽBE BR. 3

Sinteza logičkih kola. Karnaughove mape

LABORATORIJSKE VJEŽBE BR. 3

Sinteza logičkih funkcija

Prilikom dizajna digitalnih krugova, dizajner najčešće počinje sa tabelom istinitosti koja opisuje šta digitalno kolo treba raditi. Zadatak dizajnera je da odredi koji sklop će obavljati opisanu funkciju. Dvije najčešće korištene metode za ovo su SOP (Sum of Products) i POS (Product of sums).

Nakon toga se izraz korištenjem Booleove algebre ili Karnaughovih mapa (koje će se obrađivati kasnije) implementira sa minimalnim brojem logičkih kola.

SOP implementacija – je izraz koji obuhvata sumu proizvoda svih ulaznih varijabli ili njihovih negacija za koje je izraz u tabeli istinitosti 1 (tzv. mintermi).

SOP je jednostavan za implementaciju korištenjem AND gateova koji dolaze u jedno OR kolo.

Prilikom sinteze logičkih funkcija obično se primjenjuje sljedeća procedura prema redoslijedu:

| | | |
|----|---|--|
| 1. | Specifikacija problema (opis proizvoda ili uređaja) | |
| 2. | Identifikacija potrebnih ulaznih i izlaznih varijabli | |
| 3. | Određivanje veza između ulaznih i izlaznih varijabli | Tabele istinitosti, SOP, POS Minterm/Maxterm izrazi Karnaughove mape |
| 4. | Određivanje logičkih izraza koji karakteriziraju ulazno-izlazne varijable | Booleova algebra |
| 5. | Implementacija logičkih izraza korištenjem logičkih kola | Osnovna logička kola |

ZADATAK 1

| A | B | C | W |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

1

Pronaći W iz tabele istinitosti za SOP oblik. Nacrtati logičko kolo koje to implementira:

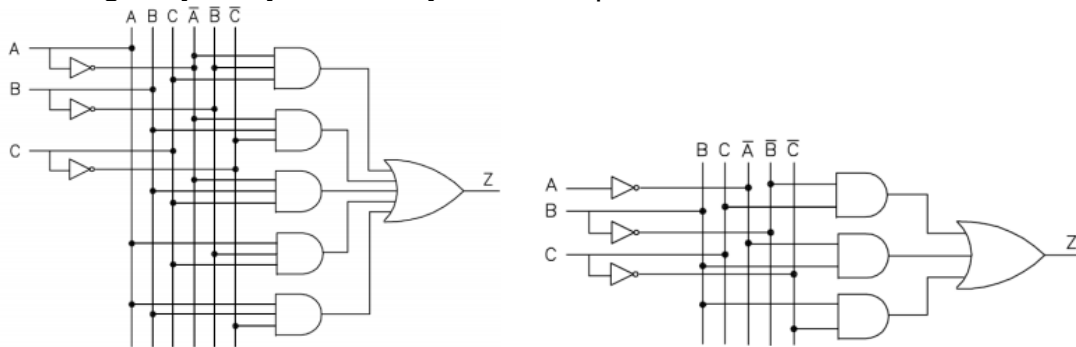
Primijetiti da svaki od proizvoda u SoP obliku sadrži sve varijable

$$W = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C}$$

Funkciju možemo minimizirati korištenjem Booleovih izraza:

$$\begin{aligned} W &= \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} \\ &= \bar{A} \cdot B + \bar{B} \cdot C + A \cdot B \cdot \bar{C} \\ &= \bar{B} \cdot C + B(\bar{A} + A \cdot \bar{C}) = \bar{B} \cdot C + B(\bar{A} + \bar{C}) \\ &= \bar{B} \cdot C + \bar{A} \cdot B + B \cdot \bar{C} \end{aligned}$$

Nakon toga obje varijante funkcije možemo prikazati u Multisimu:

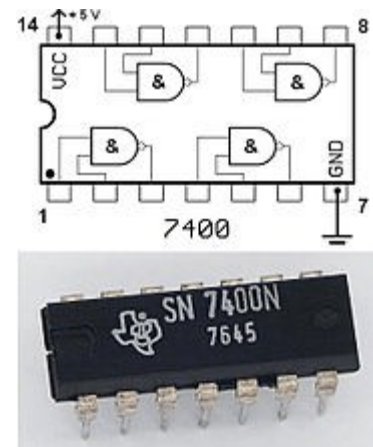


Univerzalna logička kola

Korištenjem samo od NOR elemenata (ili alternativno NAND elemenata) mogu se reproducirati funkcionalnosti svih drugih logičkih kola. Zbog toga, ova se kola nazivaju univerzalnim logičkim kolima.

Iz tog razloga, često se za implementacije složenijih spojeva koristi samo jedan tip kola, npr. NAND kola (u praksi se NAND kola mogu proizvesti s jako malim brojem tranzistora).

U praksi, NAND kola se najčešće izvedu kao čipovi (integrirani krugovi), pri čemu se ugrađuju npr. u DIP kućišta. Primjer integralnog kola koje sadrži više NAND logičkih vrata je IC 7400 (4 dvoulazna NAND kola u 14-pinskom DIP, kao na slici). Dodatno je potrebno dovesti napajanje za ova integralna kola.



NAND gate je invertovani AND gate, čija je tabela istinitosti sljedeća:

$$Q = \text{NOT}(A \text{ AND } B)$$

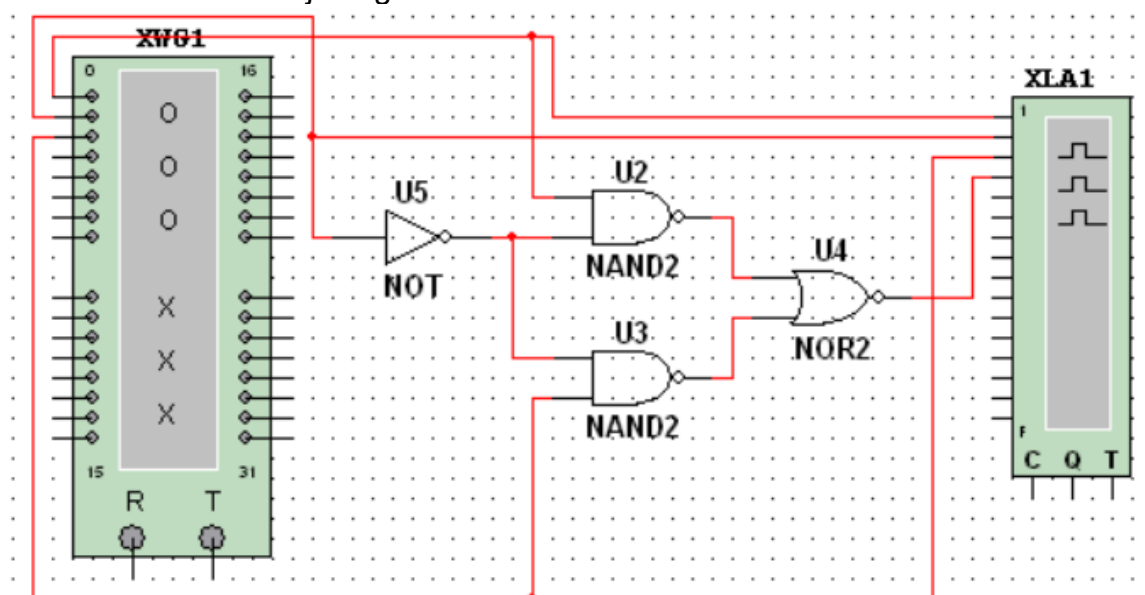
| Input | Input | Output |
|-------|-------|--------|
|-------|-------|--------|

| A | B | Q |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

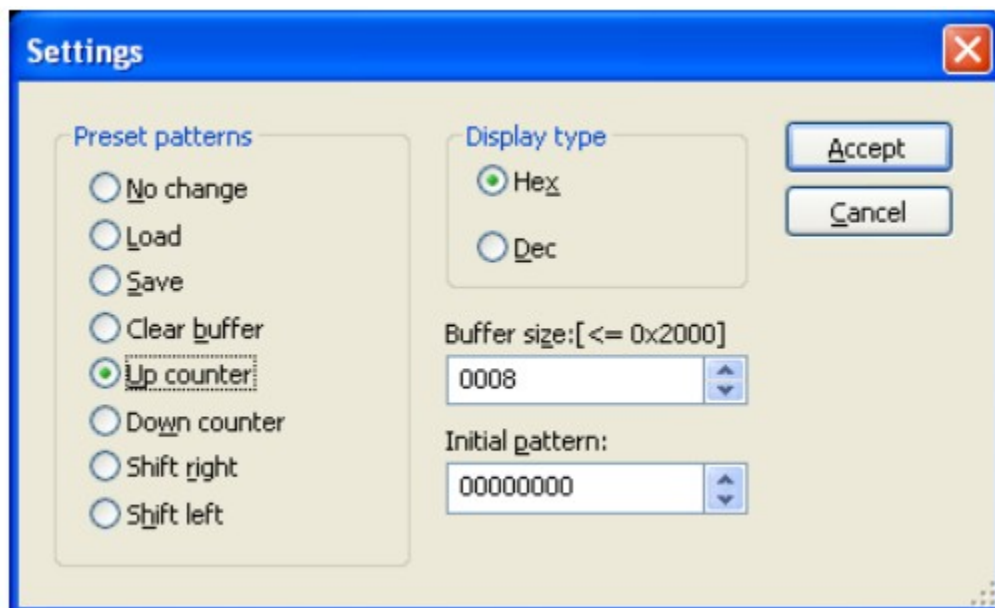
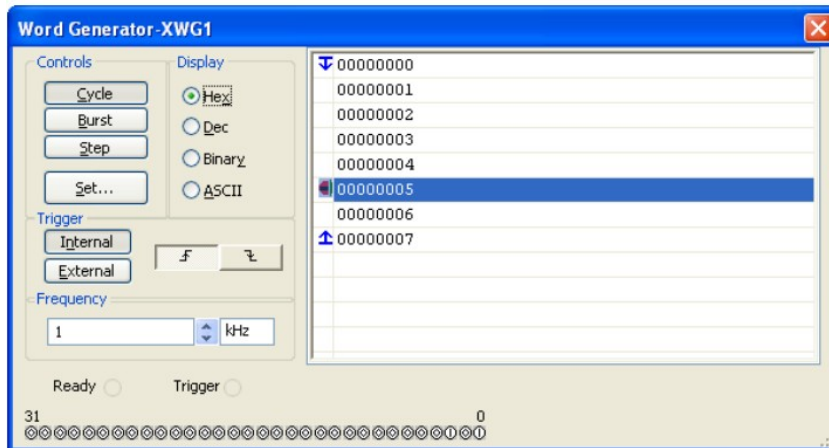


Multisim Word generator

U Multisimu postoji alat za generiranje digitalnih uzoraka - instrument "Word Generator" koji se može dodati kolu preko ikone iz toolbara na desnoj strani ili preko menija **Simulate>Instruments>Word Generator**. Na slici je prikazan Word Generator XWG1, spojen na tri ulaza u kolo. Word generator proizvodi sekvencu uzoraka koji mogu imati 1-32 bita.

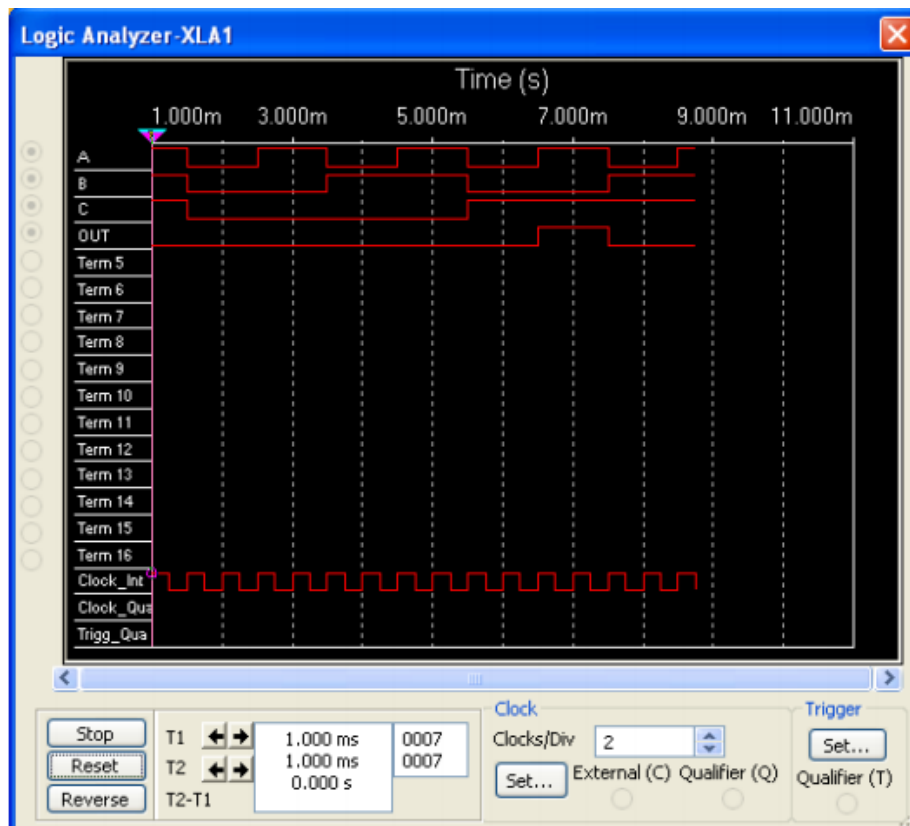


Dvostrukim klikom na simbol Word generatora otvara se prozor za specifikaciju uzoraka i frekvenciju na kojoj se oni ponavljaju. Upravljačka tipka na lijevoj strani prozora dozvoljava uzorcima da se kontinualno ponavljaju "Cycle", da se izvrši sekvenca samo jednom "Burst" ili da se pojavi samo jedan uzorak "Step". Uzorci izlaze prema redoslijedu kojim su ispisani, i mogu se unijeti ručno ili generirati automatski. Za automatski generirane uzorke, pritisnuti tipku Set, i zatim na prozoru Settings odabrati željenu sekvencu (npr. Up counter proizvodi sekvencu binarnih brojeva, "Buffer Size" je podešen na 8 kako bi ograničio sekvencu na 8 brojeva, "Initial Pattern" je podešen na 0 da brojanje kreće od 0).



Primijetiti da su 8 binarnih brojeva koji su dodani pomoću "Up counter" sekvence prikazani u prozoru Word generatora. Frekvenciju na kojoj Word generator izmjenjuje signale podešavamo preko Frequency polja - na slici je ona podešena na 1000kHz, što znači da će 1000 uzoraka biti generirano svake sekunde, tj jedan uzorak svaku 1 ms.

Multisim Logic Analyzer



Drugi koristan alat je instrument Logic Analyzer, prikazan na slici. Logički analizator je instrument koji snima i prikazuje sekvence digitalnih vrijednosti u vremenu, pri čemu su sekvence prikazane kao valni oblici, umjesto kao tabele brojeva.

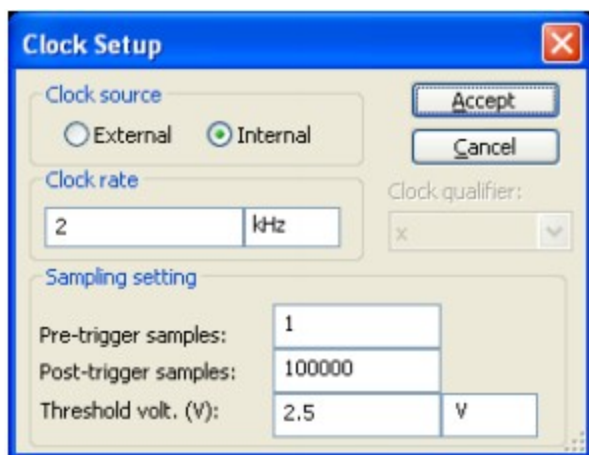
Multisim Logic Analyzer instrument može snimiti do 16 signala.

Uzorci su trigerovani ili preko internog signala takta - N uzoraka po sekundi ili preko eksternog takta (clock).

Na slici je prikazan Logic Analyzer sa snimljena 3 ulaza i izlazom, kao što je nacrtano na šemi od Word Generatora. Prikazan je odziv na "Burst" vrijednosti koje su ranije definirane, plus interni clock.

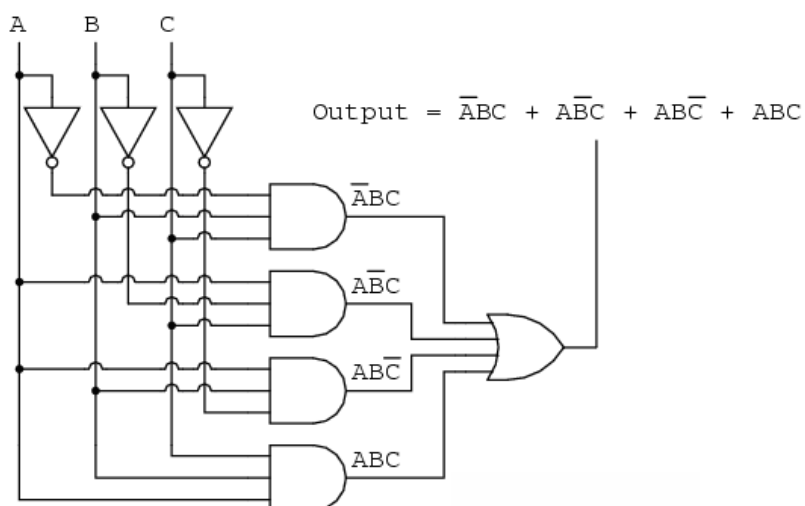
Logički analizator mora biti podešen da hvata vrijednosti u ispravnim vremenskim trenucima. Ovo se izvršava pomoću tipke Set unutar logic analyzera, što otvara prozor "Clock Setup".

S obzirom da je Word Generator konfiguriran da generira uzorke u vremenu 1kHz, odabran je takt od 2kHz kako bismo osigurali da se snime dvije vrijednosti za svaki ulazni pattern.



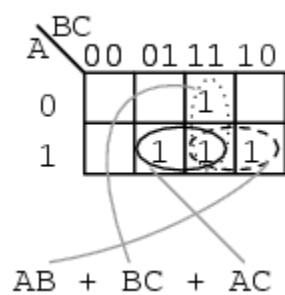
ZADATAK 1

Korištenjem Karnoughovih mapa pokazati da je:
 $ABC + AB'C + ABC' + ABC = AB + BC + AC$



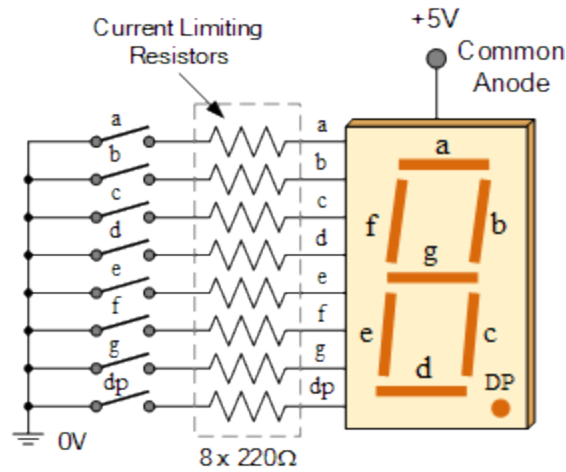
Simulirati rezultate .

Rješenje:



ZADATAK 2

Na slici je zadan 7-segmentni displej, pri čemu svaki od prekidača pali jedan segment od 3 diode – prikazan je broj 3.



Implementirati kolo koje binarne brojeve 0000-1001 pretvara u odgovarajući broj na 7-segmentnom displeju.

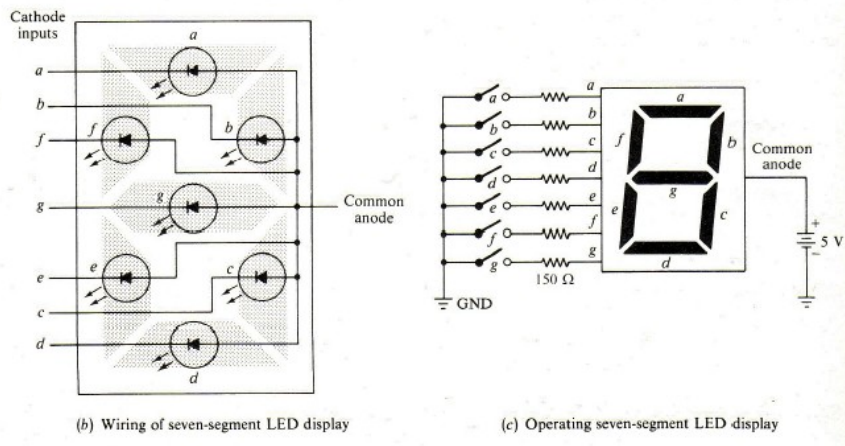


Tabela istinitosti:

| | a | b | c | d | e | f | g | |
|------|---|---|---|---|---|---|---|---|
| 0000 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0001 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0010 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 2 |
| 0011 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 3 |
| 0100 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 4 |
| 0101 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5 |
| 0110 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 6 |
| 0111 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 7 |
| 1000 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| 1001 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 9 |

Karnaughove mape za svaki od segmenata:

Segment a

| | | | | | |
|----|----|----|----|---|----|
| | | !C | !C | C | C |
| | | ID | D | D | ID |
| !A | !B | 1 | 0 | 1 | 1 |
| !A | B | 0 | 1 | 1 | 1 |
| A | B | D | D | D | D |
| A | !B | 1 | 1 | D | D |

$$A + C + B D + !B !D$$

Segment b

| | | | | | |
|----|----|----|----|---|----|
| | | !C | !C | C | C |
| | | ID | D | D | ID |
| !A | !B | 1 | 1 | 1 | 1 |
| !A | B | 1 | 0 | 1 | 0 |
| A | B | D | D | D | D |
| A | !B | 1 | 1 | D | D |

$$!B + !C !D + C D$$

Segment c

| | | | | | |
|----|----|----|----|---|----|
| | | !C | !C | C | C |
| | | ID | D | D | ID |
| !A | !B | 1 | 1 | 1 | 0 |
| !A | B | 1 | 1 | 1 | 1 |
| A | B | D | D | D | D |
| A | !B | 1 | 1 | D | D |

$$B + !C + D$$

Segment d

| | | | | | |
|----|----|----|----|---|----|
| | | !C | !C | C | C |
| | | ID | D | D | ID |
| !A | !B | 1 | 0 | 1 | 1 |
| !A | B | 0 | 1 | 0 | 1 |
| A | B | D | D | D | D |
| A | !B | 1 | 1 | D | D |

$$A + C !D + !B !D + !B C + B !C D$$

Segment e

| | | | | | |
|----|----|----|----|---|----|
| | | !C | !C | C | C |
| | | ID | D | D | ID |
| !A | !B | 1 | 0 | 0 | 1 |
| !A | B | 0 | 0 | 0 | 1 |
| A | B | D | D | D | D |
| A | !B | 1 | 0 | D | D |

$$C !D + !B !D$$

Segment f

| | | | | | |
|----|----|----|----|---|----|
| | | !C | !C | C | C |
| | | ID | D | D | ID |
| !A | !B | 1 | 0 | 0 | 0 |
| !A | B | 1 | 1 | 0 | 1 |
| A | B | D | D | D | D |
| A | !B | 1 | 1 | D | D |

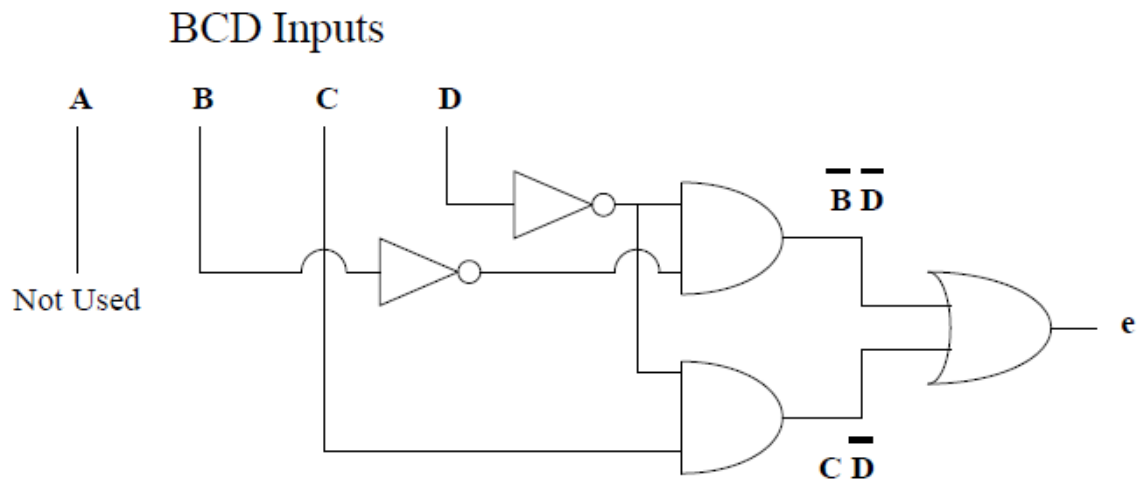
$$A + B !C + !C !D + B !D$$

Segment g

| | | | | | |
|----|----|----|----|---|----|
| | | !C | !C | C | C |
| | | ID | D | D | ID |
| !A | !B | 0 | 0 | 1 | 1 |
| !A | B | 1 | 1 | 0 | 1 |
| A | B | D | D | D | D |
| A | !B | 1 | 1 | D | D |

$$A + B !C + B !D + !B C$$

Primjer: Implementacija za segment e:



Ostale segmente ucrtati samostalno u Quartus simulator i testirati!

| Decimal Digit | Individual Segments Illuminated | | | | | | |
|---------------|---------------------------------|---|---|---|---|---|---|
| | a | b | c | d | e | f | g |
| 0 | x | x | x | x | x | x | |
| 1 | | x | x | | | | |
| 2 | x | x | | x | x | | x |
| 3 | x | x | x | x | | | x |
| 4 | | x | x | | | x | x |
| 5 | x | | x | x | | x | x |
| 6 | x | | x | x | x | x | x |
| 7 | x | x | x | | | | |
| 8 | x | x | x | x | x | x | x |
| 9 | x | x | x | | | x | x |

