Увод у организацију и архитектуру рачунара 1

Александар Картељ

kartelj@matf.bg.ac.rs

прилагодила: Јована Ковачевић

Напомена: садржај ових слајдова је преузет од проф. Саше Малкова

Основи дигиталне логике

Булова алгебра

Алгебра логике

- Алгебра логике је структура { S; $\wedge$, $\vee$, ¬ }
  - Константе: S={ $\top$, $\bot$ }
  - Унарна операција: ¬
  - Бинарне операције: $\wedge$, $\vee$

Алгебра логике (2)

- Рачунарска нотација:
  - Константе: 0, 1
  - Променљиве: A, B, C, ...
  - Операције: ~A (или A'...), A·B (или AB), A+B

Законитости

$AB = BA$	$A+B = B+A$	Закон комутације
$(AB)C = A(BC)$	$(A+B)+C$	Закон асоцијације
$A(B+C) = AB+AC$	$A+(BC) = (A+B)(A+C)$	Закон дистрибуције
$1 \cdot A = A$	$0+A = A$	Неутрални елемент
$A\ A' = 0$	$A+A' = 1$	Инверзни елемент

Законитости (2)

$A \cdot 0 = 0$	$A + 1 = 1$	Нула
$A \cdot A = A$	$A + A = A$	Идемпотенција
$A\,(A + B) = A$	$A + (AB) = A$	Апсорпција
$A'' = A$		Двострука негација
$(AB)' = A' + B'$	$(A+B)' = A'\,B'$	Де Морганова правила

Логичке функције

- Свака функција са доменом S^n и кодоменом S назива се **логичка функција**:

$$f: S \times S \times \ldots \times S \rightarrow S$$

Логичке функције (2)

- Колико постоји различитих логичких функција реда *n*
  (са *n* аргумената)?

$$2^{2^n}$$

Логичке функције реда 0

- Логичке функције без аргумената:
  - $f_{00}(x) = 0$
  - $f_{01}(x) = 1$

Логичке функције реда 1

Argument	Vrednost		Naziv	Oznaka
A	0	1		
Funkcija				
f_{10}	0	0	nula funkcija	
f_{11}	0	1	identitet	
f_{12}	1	0	negacija	¬A
f_{13}	1	1	jedinična funkcija	

Логичке функције реда 2

Argument	Vrednost			
A	0	0	1	1
B	0	1	0	1
Funkcija				
f_{20}	0	0	0	0
f_{21}	0	0	0	1
f_{22}	0	0	1	0
f_{23}	0	0	1	1
f_{24}	0	1	0	0
f_{25}	0	1	0	1
f_{26}	0	1	1	0
f_{27}	0	1	1	1
f_{28}	1	0	0	0
f_{29}	1	0	0	1
f_{2A}	1	0	1	0
f_{2B}	1	0	1	1
f_{2C}	1	1	0	0
f_{2D}	1	1	0	1
f_{2E}	1	1	1	0
f_{2F}	1	1	1	1

Логичке функције реда 2 (2)

Argument	Vrednost				Naziv	Oznaka
A	0	0	1	1		
B	0	1	0	1		
Funkcija						
f_{20}	0	0	0	0	nula funkcija	
f_{21}	0	0	0	1	konjunkcija	$A \wedge B$
f_{22}	0	0	1	0	negacija implikacije od A ka B	$A \wedge \neg B = \neg(A \Rightarrow B)$
f_{23}	0	0	1	1	prva projekcija	
f_{24}	0	1	0	0	negacija implikacije od B ka A	$\neg A \wedge B = \neg(B \Rightarrow A)$
f_{25}	0	1	0	1	druga projekcija	
f_{26}	0	1	1	0	ekskluzivna disjunkcija	$(A \wedge \neg B) \vee (\neg A \wedge B)$
f_{27}	0	1	1	1	disjunkcija	$A \vee B$
f_{28}	1	0	0	0	Pirsova (Lukašievičeva) funk.	$A \downarrow B$
f_{29}	1	0	0	1	ekvivalencija	$(A \Leftrightarrow B)$
f_{2A}	1	0	1	0	negacija druge projekcije	
f_{2B}	1	0	1	1	implikacija od B na A	$B \Rightarrow A$
f_{2C}	1	1	0	0	negacija prve projekcije	
f_{2D}	1	1	0	1	implikacija od A na B	$A \Rightarrow B$
f_{2E}	1	1	1	0	Šeferova funkcija	$A \uparrow B$
f_{2F}	1	1	1	1	Jedinična funkcija	

Пун систем функција

- Пун систем функција је скуп функција на основу кога се могу извести све остале функције

- Ако се из неког система функција могу извести све функције неког пуног система функција, онда је и тај систем пун систем функција

Неки пуни системи функција

- $\{ \wedge, \vee, \neg \}$
- $\{ \wedge, \neg \}$
- $\{ \vee, \neg \}$
- $\{ \uparrow \}$
- $\{ \downarrow \}$

Нормалне форме функција

- Елементарна конјункција
  - логички израз који не садржи дисјункцију
  - тј. садржи само негацију и конјункцију
  - пример:
    - A B C’ D E’
- Елементарна дисјункција
  - логички израз који не садржи конјункцију
  - тј. садржи само негацију и дисјункцију
  - пример:
    - A + B’ + C + D’ + E

Нормалне форме функција (2)

- Савршена елементарна конјункција
  - елементарна конјункција која садржи све променљиве из скупа променљивих (обично “све аргументе функције”)

- Савршена елементарна дисјункција
  - елементарна дисјункција која садржи све променљиве из скупа променљивих (обично “све аргументе функције”)

Нормалне форме функција (3)

- Дисјунктивна форма
  - логички израз који се састоји од елементарних конјункција међусобно повезаних операцијама дисјункције
- Конјунктивна форма
  - логички израз који се састоји од елементарних дисјункција међусобно повезаних операцијама конјункције

Нормалне форме функција

- Канонска дисјунктивна нормална форма
  - дисјунктивна форма у којој су све конјункције савршене елементарне конјункције
  - КДНФ (енгл. *SOP – sum of products*)

- Канонска конјунктивна нормална форма
  - конјунктивна форма у којој су све дисјункције савршене елементарне дисјункције
  - ККНФ (енгл. *POS – product of sums*)

Улога ККНФ и КДНФ

- Свака логичка функција се може дефинисати у облику ККНФ и КДНФ

Основи дигиталне логике

Логичка кола и логички елементи

Логичка кола

- Логичка кола су апстрактна дигитална кола која имплементирају логичке функције
- Представљају апстракцију електричних (оптичких и других) кола

Логички елементи

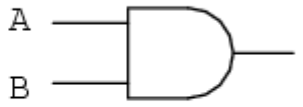
- Логички елементи су елементарна дигитална кола која имплементирају елементарне логичке функције
- Обично логички елементи имплементирају функције које чине неки пун систем функција

Уобичајени логички елементи

Функција	**Назив логичког елемента**	**Назив на енглеском**
Негација	НЕ-елемент	NOT-element
Конјункција	И-елемент	AND-element
Дисјункција	ИЛИ-елемент	OR-element
Шеферова функција	НИ-елемент	NAND-element
Пирсова функција	НИЛИ-елемент	NOR-element

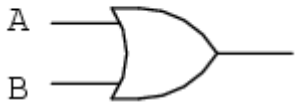
И - елемент

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



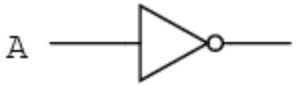
ИЛИ - елемент

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



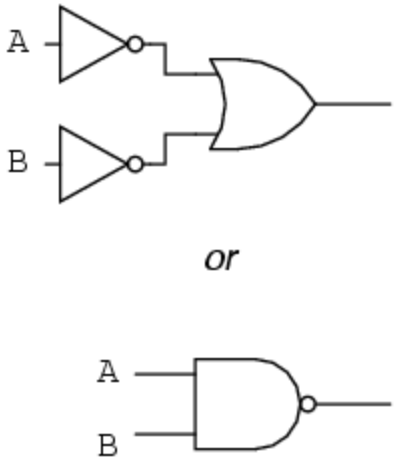
НЕ - елемент

A	Output
0	1
1	0



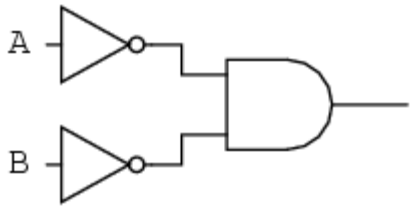
НИ - елемент

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



НИЛИ - елемент

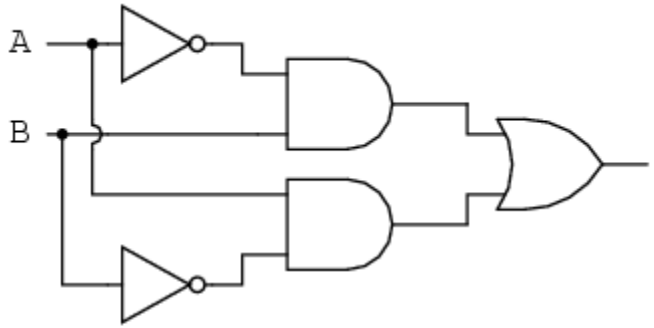
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



or

ЕИЛИ (ексклузивно ИЛИ)- елемент

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

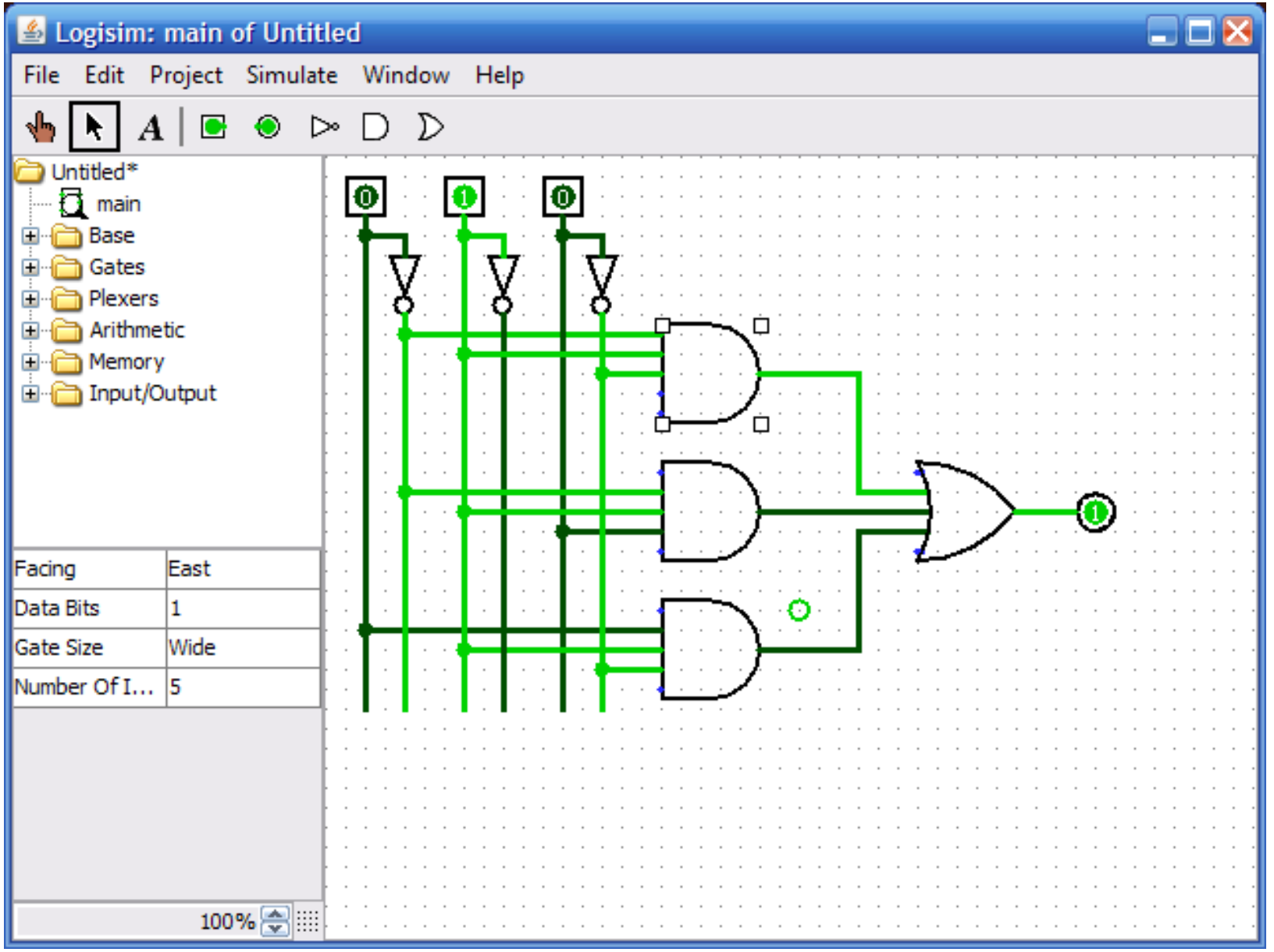


or

Софтвер

- Постоји више програма за вежбање пројектовања логичких кола и симулирање њиховог рада
  - *CEDAR Logic*
    - подржава релативно сложена кола
  - *Logisim*
    - постоји за различите ОС
    - могућност чувања и употребе модула (“кола”)
  - *Logic Circuit Designer*
    - могућност чувања и употребе модула (“кола”)
  - *Logic Gate Simulator*
    - могућност чувања и употребе модула (“кола”)

Logisim


Logisim: main of Untitled
File Edit Project Simulate Window Help
Untitled*
main
Base
Gates
Plexers
Arithmetic
Memory
Input/Output
Facing East
Data Bits 1
Gate Size Wide
Number Of I... 5
100%

Пројектовање логичких кола

- Одређивање проблема
  - често интегрисано са наредним кораком
- Извођење истинитосне таблице
- Извођење логичког израза
  - обично КДНФ, непосредно из таблице
  - некада и без претходног корака
- Упрошћавање логичког израза
  - свођење на минималан облик
- Обликовање логичког кола

Основи дигиталне логике

Минимизација логичких функција

Минимизација логичких функција

- Свака логичка функција се може изразити на више различитих начина
- Свака логичка функција се може записати у облику КДНФ и ККНФ
- Да би се записивање и имплементација функција учинили поједноставили, потребно је минимизовати функције
  - **минимизовање логичке функције је проналажење њеног најједноставнијег записа**

Методи минимизације

- Алгебарске трансформације
- Карноове мапе
- Таблична метода Квин-Мек Класког

Алгебарске трансформације

- Почивају на примени законитости и правила идентитата Булове алгебре

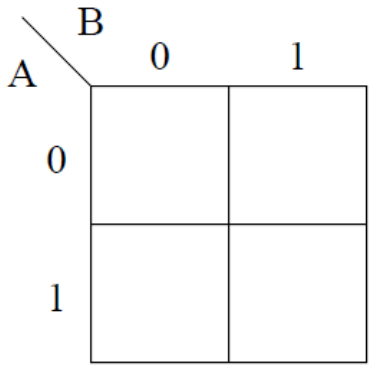
Пример

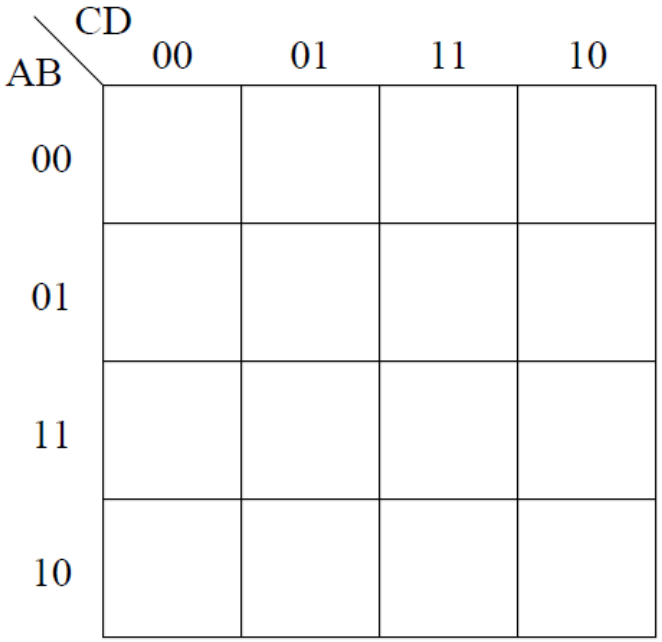
- Пример: пронаћи минималан запис функције
  - F = A’BC’ + A’BC + ABC’
- корак 1: идемпотенција и комутација
  - F = A’BC’ + A’BC + ABC’ + A’BC’
- корак 2: закон дистрибуције
  - F = A’B(C’ + C) + BC’ (A + A’)
- корак 3: закон о инверзним елементима
  - F = A’B + BC’
- корак 4: закон дистрибуције
  - F = B(A’ + C’)

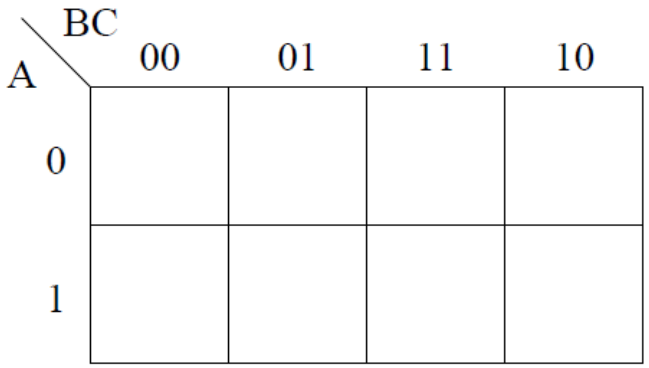
Карноове мапе

- Почивају на представљању табела у облику који се лако може оптимизовати
  - једноставно за аутоматизацију
- Основна идеја:
  - прави се вишедимензиона мрежа
  - на свакој димензији се наводе највише по два аргумента функције
  - вредности аргумената се наводе таквим редом да се мења по тачно један бит (Хамингова дистанца 1):
    - 00, 01, 11, 10

Примери







Карноове мапе (2)

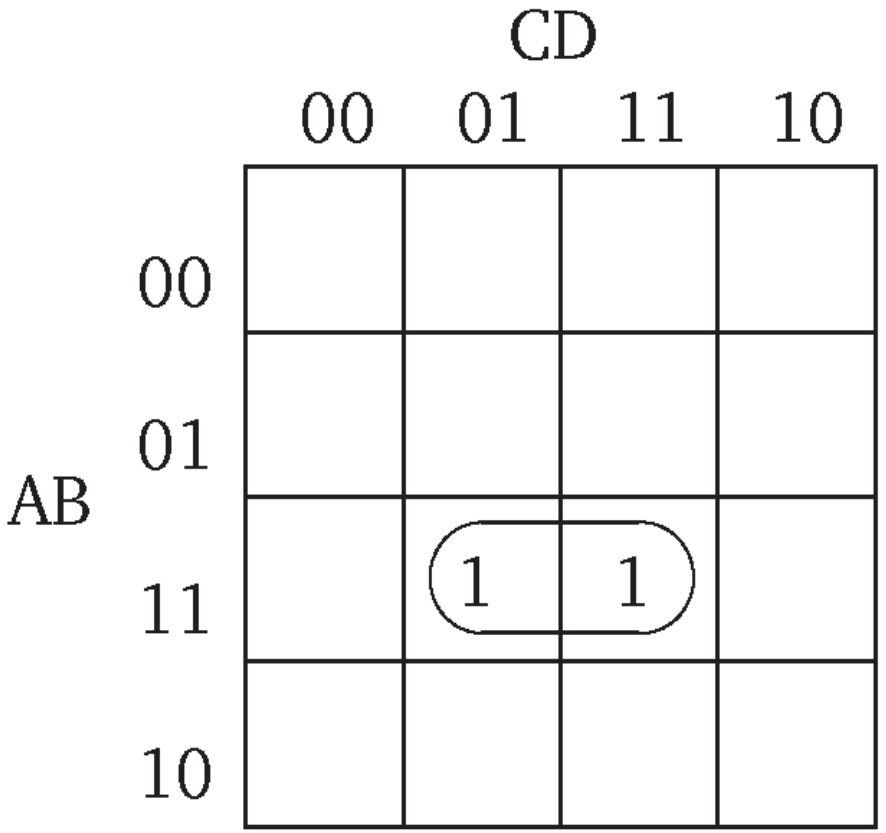
- Основу за упрошћавање представља правило:
  - $A_1A_2...A_k...A_n + A_1A_2...A_k'...A_n = A_1A_2...A_{k-1}A_{k+1}...A_n$

- Како се суседни квадрати у Карноовим мапама увек разликују за по највише један бит, ако одговарају истим вредностима функције онда се одговарајући бит може елиминисати
- И квадрати на крајњим супротним странама се сматрају за суседне

Карноове мапе (3)

- КДНФ има по дисјункт за сваки квадрат у мапи који садржи јединицу
- Ако два суседна квадрата садрже јединицу, два дисјункта којима они одговарају се могу заменити једним, који не садржи аргумент који се разликује
- ...слично и за четири квадрата, 4x1 или 2x2

Пример

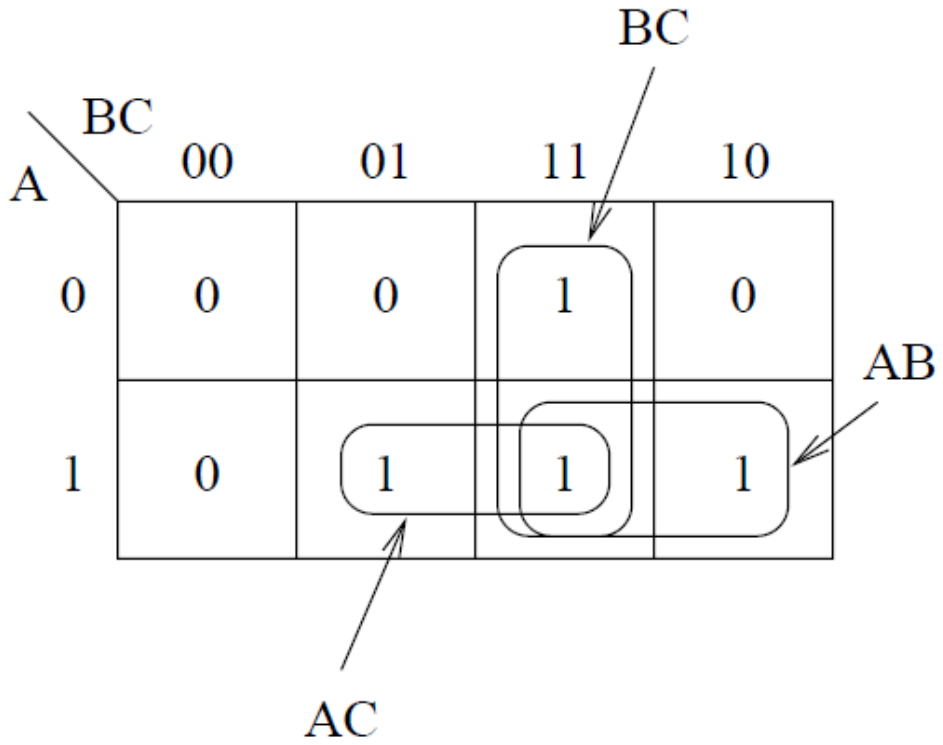


Карноове мапе (4)

- Уопштено:
  - област реда 0 је јединични квадрат
  - област реда *n*+1 је унија две суседне области реда *n*
  - ако две суседне области реда *n* садрже јединице, оне се замењују једном облашћу реда *n*+1
  - претходни корак се примењује докле год је могуће
  - почев од области највишег реда
    - ако област није у потпуности обухваћена другим областима вишег или истог реда, прави се дисјункт
      - дисјункт се састоји само од аргумената који су константни за дату област
  - провери се редундантност
  - добијени израз је минимизована функција

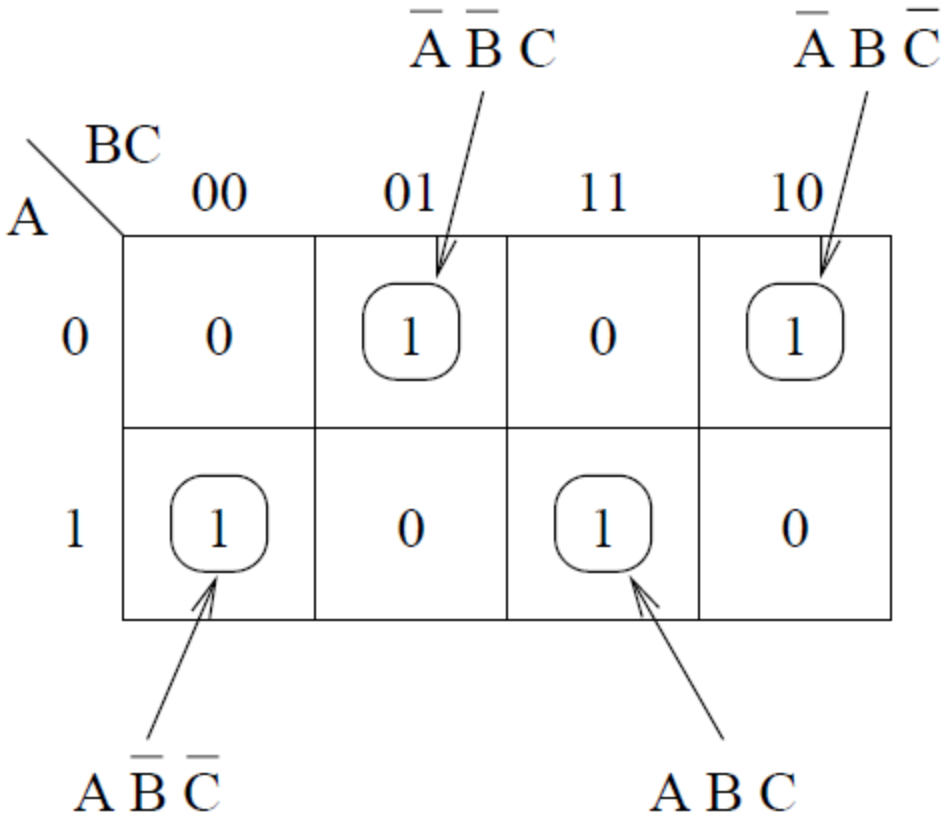
Пример

- Функција три аргумента рачуна ону вредност која је заступљенија у аргументима



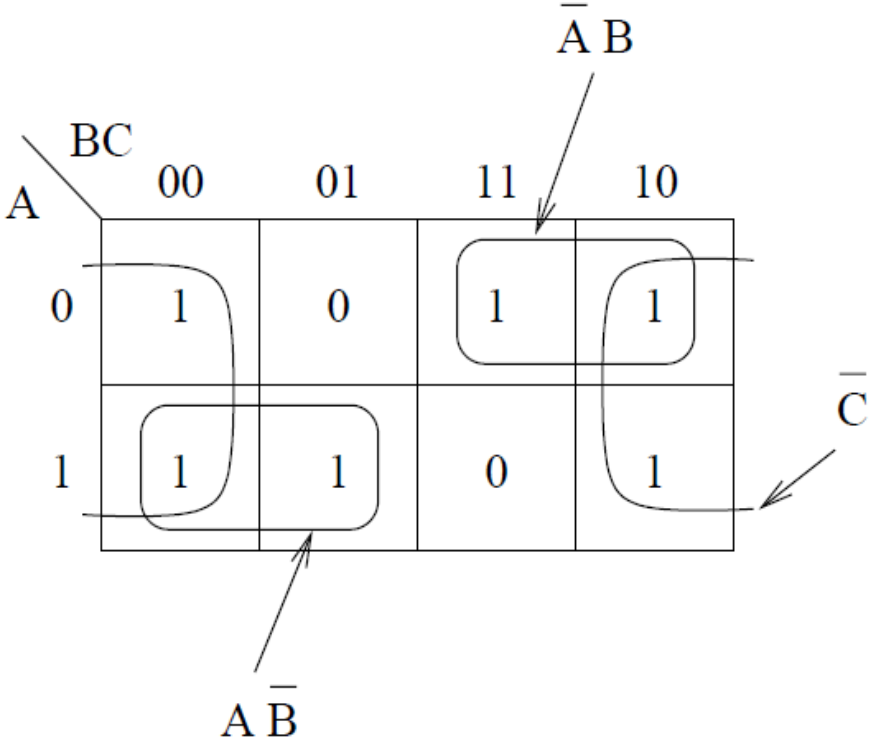
Пример

- Функција три аргумента рачуна парност

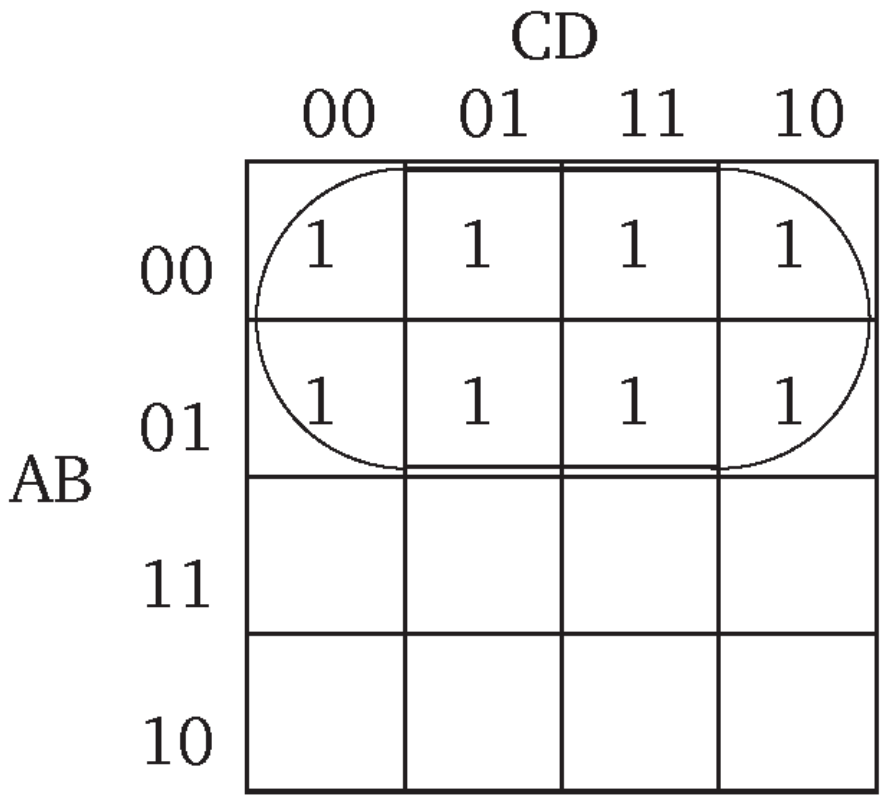


Пример

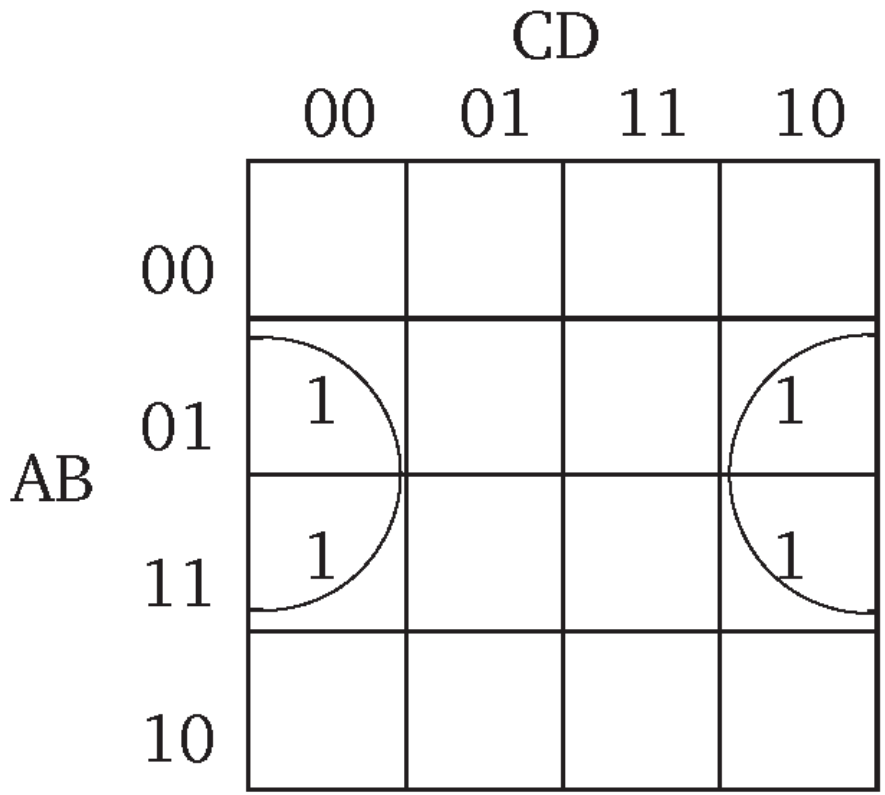
- Пример функције три аргумента код које постоје суседне области на супротним крајевима мапе



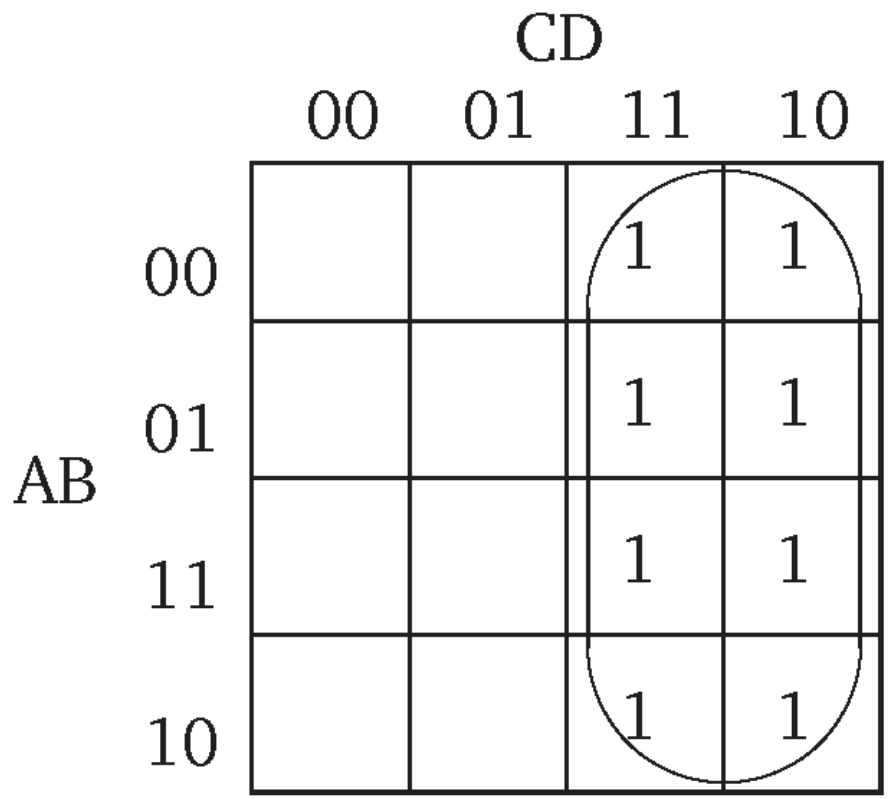
Пример



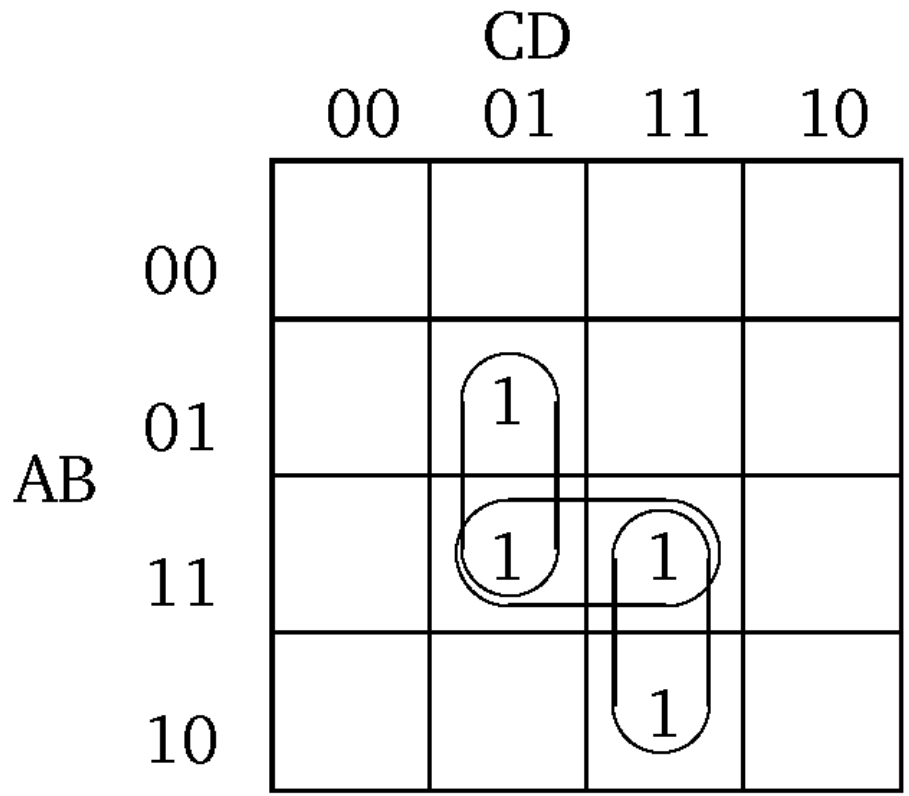
Пример



Пример



Пример



Пример

- Логичка функција која реализује сабирање декадне цифре записане у 8421 коду

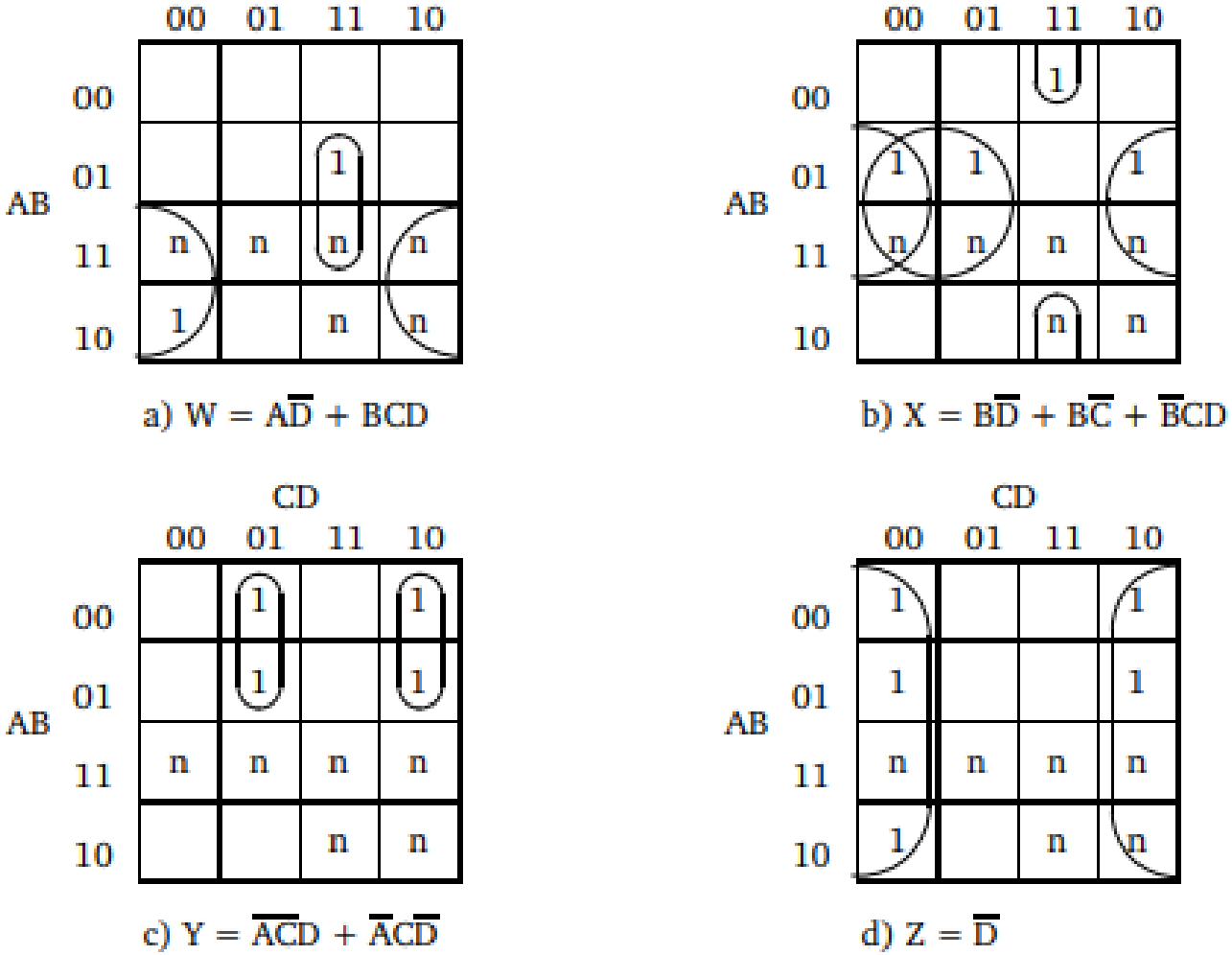
- Комбинације које се не користе у коду 8421 (1010-1111) су означене са „n“ (небитно)

- „n“ можемо користити и као 0 и као 1

Ulaz					Izlaz				
Dekadna cifra	kod 8421				Dekadna cifra	kod 8421			
	A	B	C	D		W	X	Y	Z
0	0	0	0	0	1	0	0	0	1
1	0	0	0	1	2	0	0	1	0
2	0	0	1	0	3	0	0	1	1
3	0	0	1	1	4	0	1	0	0
4	0	1	0	0	5	0	1	0	1
5	0	1	0	1	6	0	1	1	0
6	0	1	1	0	7	0	1	1	1
7	0	1	1	1	8	1	0	0	0
8	1	0	0	0	9	1	0	0	1
9	1	0	0	1	0	0	0	0	0
n	1	0	1	0		n	n	n	n
n	1	0	1	1		n	n	n	n
n	1	1	0	0		n	n	n	n
n	1	1	0	1		n	n	n	n
n	1	1	1	0		n	n	n	n
n	1	1	1	1		n	n	n	n

Tabela 5: Tabela istinitosnih vrednosti za jednocifreni BCD sabirač

Пример



Slika 10: Karnoove mape za BCD jednocifreni sabirač

Метода Квин-МекКласког

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

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

- Дата је функција *F* таблично. Из табеле можемо закључити њену канонску дисјунктивну нормалну форму (КДНФ):

$$F = \overline{A}\overline{B}\overline{C}D + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}CD + AB\overline{C}\overline{D} + AB\overline{C}D + ABCD$$

Метода Квин-МекКласког

- Фаза 1: конструкција табеле у којој сваки ред одговара једном дисјункту
- Редови су груписани по броју комплементарних променљивих

Term	A	B	C	D	
$\overline{A}\overline{B}\overline{C}D$	0	0	0	1	✓
$\overline{A}B\overline{C}D$	0	1	0	1	✓
$\overline{A}BC\overline{D}$	0	1	1	0	✓
$AB\overline{C}\overline{D}$	1	1	0	0	✓
$\overline{A}BCD$	0	1	1	1	✓
$A\overline{B}CD$	1	0	1	1	✓
$AB\overline{C}D$	1	1	0	1	✓
$ABCD$	1	1	1	1	✓

- Следећи корак је проналажење свих парова терма који се међусобно разликују само за вредност једне променљиве
- Кад год се овакви терми упаре, штиклираћемо врсту сваког од њих, комбиновати пронађени пар елиминацијом променљиве која се разликује и добијени терм прикључити новој табели

Term	A	B	C	D	
$\overline{A}\overline{C}D$	0		0	1	
$\overline{A}BC$	0	1	1		
$\overline{A}BD$	0	1		1	✓
$AB\overline{C}$	1	1	0		
$B\overline{C}D$		1	0	1	✓
ABD	1	1		1	✓
ACD	1		1	1	
BCD		1	1	1	✓

- Поступак се наставља све док се почетна табела у потпуности не исцрпи
- У наредном кораку табела се обрађује на исти начин као у претходном кораку
- У општем случају, процес се наставља све док се не добије табела у којој не могу да се пронађу упарени терми

Term	A	B	C	D
BD		1		1

Метода Квин-МекКласког

- Друга фаза:
  - Конструисати табелу чији редови одговарају термима који нису елиминисани а колоне термима из улазне КДНФ функције
  - Ако је ознака врсте садржана у ознаци колоне, на њиховом пресеку уписати X
  - Ако колона садржи само једно X, заокружити га
  - У свим редовима који садрже заокружено X уоквирити остале X
  - Ако свака колона садржи или заокружено или уоквирено X, поступак је завршен и минимални облик чине терми додељени редовима у којима је означено X
  - Ако постоји бар једна колона која не садржи ни заокружено ни уоквирено X, неопходно је спровести додатни корак који се састоји у додавању редова све док не буду обухваћене све колоне

Метода Квин-МекКласког

	$\overline{A}\overline{B}CD$	$\overline{A}B\overline{C}D$	$\overline{A}BC\overline{D}$	$\overline{A}BCD$	$A\overline{B}CD$	$AB\overline{C}D$	$AB\overline{C}\overline{D}$	$ABCD$
BD		X		X			X	X
$\overline{A}CD$	(X)	[X]						
$\overline{A}BC$			(X)	[X]				
$AB\overline{C}$						(X)	[X]	
ACD					(X)			[X]