LC\_Curs3.

Echivalențe logice în logica propozițională

* Legile lui **DeMorgan**:
* Legile de **absorbție**:
* Legile de **comutativitate**:
* Legile de **asociativitate**:
* Legile **distributivității**:
* Legile de **idempotență**:

O formulă se numește **consistentă** dacă și numai dacă are cel puțin un model, deci poate fi evaluată ca adevărată.

O formulă se numește **validă** dacă și numai dacă formula este evaluată ca adevărată în orice interpretare (toate interpretările sunt modele ale formulei).

O formulă se numește **inconsistentă** dacă și numai dacă formula este falsă în orice interpretare (nu are niciun model).

O formulă se numește **contingentă** dacă și numai dacă este consistentă, dar nu este validă.

Forme normale în logica propozițiilor

1. Un **literal** este o variabilă propozițională sau negația sa.
2. O **clauză** este disjuncția unui număr finit de literali.
3. Un **cub** este conjuncția unui număr finit de literali.
4. **Clauza vidă** (□) este clauza fără literali, fiind clauza vidă inconsistentă.
5. O formulă este în **formă normală disjunctivă (FND)**, dacă aceasta este scrisă ca o disjuncție de cuburi.

1. O formulă este în **formă normală conjunctivă (FNC)**, dacă aceasta este scrisă ca o conjuncție de clauze.

Algoritmul de normalizare:

Pas 1: Înlocuire .

Pas 2: Legile lui DeMorgan

Pas 3: Legile distributivității

Pas 4: Simplificarea folosind legile de simplificare, legile absorbției, legile de idempotență.

LC\_Curs4.

Axiome și reguli de inferență

* Modus Ponens:

O formulă astfel încât se numește **teoremă**.

Consecințele teoremei de deducție

* – legea silogismului
* – legea permutării premizelor
* – legea reuniunii premizelor
* – legea separării premizelor

Teorema de **corectitudine**:

Dacă atunci .

(Validitatea sintactică implică validitatea semantică)

Teorema de **completitudine**:

Dacă atunci .

(Validitatea semantică implică validitatea sintactică)

LC\_Curs5.

Metoda tabelelor semantice

clasa α – formule de clasa β – formule de

tip conjunctiv tip disjunctiv

* O ramură se numește **închisă**(⊗) dacă ea conține o formulă și negație ei, în caz contrar, se numește **deschisă**.
* O ramură se numește **completă** dacă ea fie este închisă, fie toate formulele de pe acea ramură au fost descompuse.
* O tabelă se numește **închisă** dacă toate ramurile sale sunt închise.
* O tabelă se numește **deschisă** dacă are cel puțin o ramură deschisă.
* O tabelă se numește **completă** dacă toate ramurile ei sunt complete.

Teorema de **corectitudine și completitudine** a metodei tabelelor semantice:

O formulă este **teoremă** **(tautologie)** dacă și numai dacă există o tabelă semantică închisă pentru formula .

LC\_Curs6.

Metoda rezoluției (*sintactică, prin respingere*)

Teorema de **corectitudine și completitudine**:

Mulțimea este inconsistentă dacă și numai dacă .

este **tautologie** dacă și numai dacă .

Strategii:

* Strategia eliminării

Pas 1: Eliminarea clauzelor tautologice

Pas 2: Eliminarea clauzelor subsumate de alte clauze din S

Pas 3: Eliminarea clauzelor care conțin literali puri în S

* Strategia saturării pe nivele

1. *Res: C1-C2; C1-C3; C1-C4; C2-C3; C2-C4; C3-C4 ->*
2. *Res: C5-C1; C5-C2; C5-C3; C5-C4; C5-C6; C6-C1; C6-C2; C6-C3; C6-C4; C6-C5 -> una dintre acestea va ieși clauza vidă, altfel se continuă!*

* Strategia mulțimii suport

*, , ,*  – consistentă

Res: Luăm clauza din și o clauză din . Încercăm să rezolvăm. Dacă nu iese cu nicio clauză din , atunci este consistentă.

LC\_Curs7.

Rafinările rezoluției

* Rezoluția blocării

Teorema de **completitudine și corectitudine**:

Fie o mulțime de clauze în care fiecare literal este indexat în mod arbitrar cu un întreg.

este inconsistentă dacă și numai dacă din se deduce prin rezoluția blocării clauza vidă.

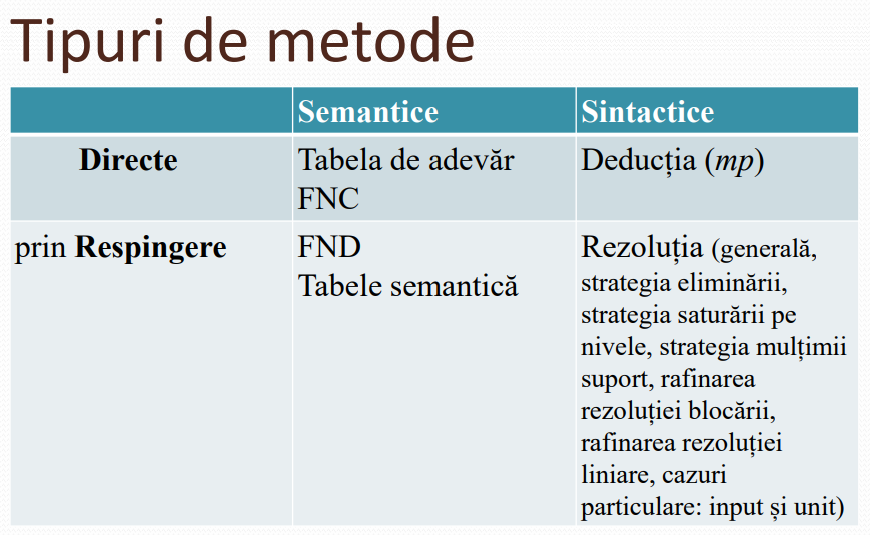
* Rezoluția liniară

Teorema de **completitudine și corectitudine**:

Mulțimea de clauze este inconsistentă dacă și numai dacă .

Cazuri particulare: Rezoluția unitară (*unit*) – clauzele centrale au *cel puțin o clauză părinte unitară* (conține un singur literal)

Rezoluția de intrare (*input*) – clauzele laterale sunt clauze *inițiale* (de intrare)

Teorema de echivalență input - unit:

dacă și numai dacă .

LC\_Curs8.

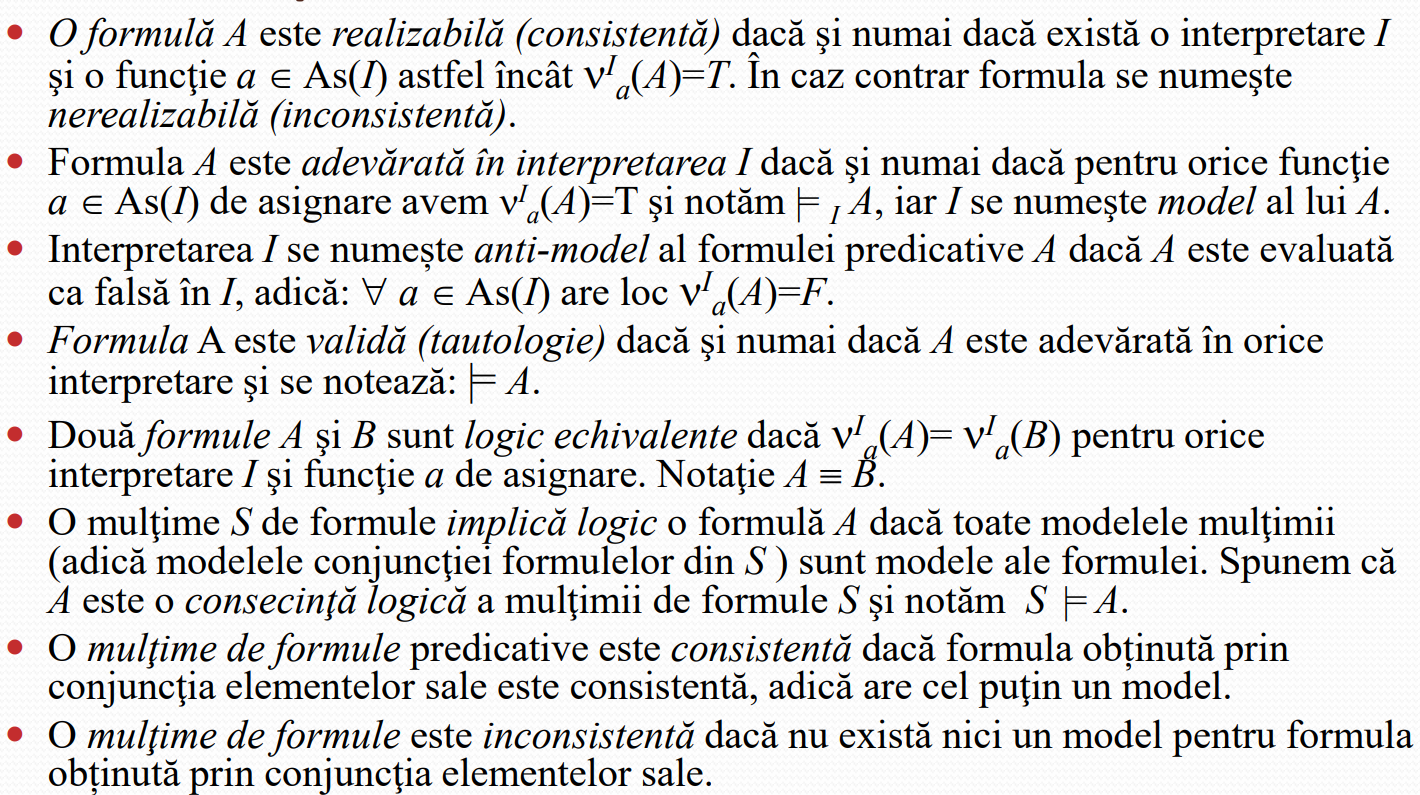
Logica predicatelor de ordinul I

* unde *t* este un termen arbitrar
* unde este o variabilă liberă în care nu apare în , iar nu este variabilă liberă nici în nici în .
* Modus Ponens:
* Regula generalizării:

( era o variabilă liberă în )

Variabilele din formulele predicative care se află sub incidența unui cuantificator se numesc ***variabile legate***, în caz contrar, ele se numesc ***variabile libere***.

O formulă predicativă se numește ***închisă***, dacă toate variabilele sale sunt legate, iar, în caz contrar, se numește***deschisă***.

 O formulă astfel încât se numește **teoremă**.

Legile distributivității:

* față de :
* față de :

Legile semidistributivității:

* față de :
* față de :

Algoritmul de aducere la forma normală clauzală:

Pas 1: Se înlocuiesc .

Pas 2: Se aplică legile lui DeMorgan.

Pas 3: Se redenumesc variabilele legate.

Pas 4: Se extrag cuantificatorii în fața formulei. (forma prenexă)

Pas 5: Eliminarea cuantificatorilor existențiali. (forma Skolem)

Pas 6: Eliminarea cuantificatorilor universali. (forma Skolem)

Pas 7: Distributivitatea lui față de . (forma normală clauzală)

Teorema de **completitudine și corectitudine**:

Fie o mulțime de formule predicative, iar o formulă predicativă.

dacă și numai dacă .

LC\_Curs9.

Metoda tabelelor semantice în calculul predicatelor

clasa – formule cuantificate universal:

| |

| |

| |

... ...

clasa – formule cuantificate existențial:

| |

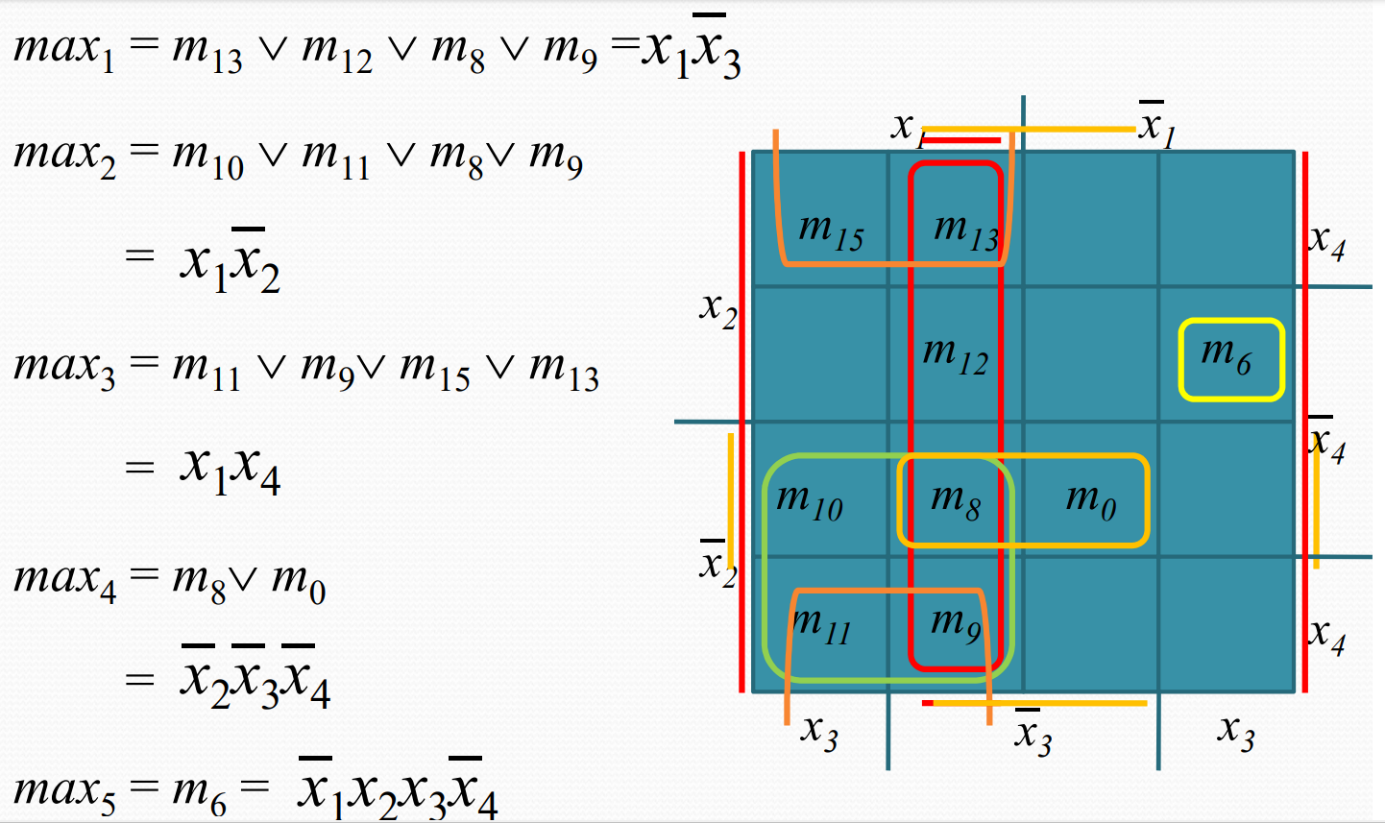
LC\_Curs11.

Algebre Booleene

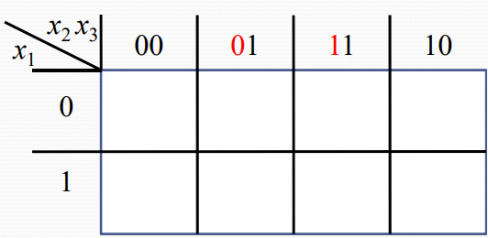
O conjuncție de variabile se numește **monom**.

Un monom care conține toate cele variabile se numește **monom canonic** sau **minterm** de variabile.

Disjuncția care conține toate cele variabile se numește **maxterm** de variabile.

****Metoda diagramelor **Veitch**

Forma simplificată:

Metoda diagramelor **Karnaugh**

Metoda analitică a lui **Quine-Mc’Clusky**

Pas 1: Ordonarea mulțimii suport a funcției cu variabile, după numărul de valori 1 conținut de fiecare -uplu.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grupul |  |  |  |  |  |
| I | 0 | 1 | 1 | 1 |  |
|  | 1 | 0 | 1 | 1 |  |
| II | 0 | 1 | 1 | 0 |  |
|  | 1 | 0 | 0 | 1 |  |
|  | 1 | 0 | 1 | 0 |  |
|  | 1 | 1 | 0 | 0 |  |
| III | 0 | 1 | 0 | 0 |  |
|  | 1 | 0 | 0 | 0 |  |
| IV=I+II | 0 | 1 | 1 | - |  |
| Factorizare | 1 | 0 | - | 1 |  |
| simplă | 1 | 0 | 1 | - |  |
| V=II+III | 0 | 1 | - | 0 |  |
|  | 1 | 0 | 0 | - |  |
|  | 1 | 0 | - | 0 |  |
|  | - | 1 | 0 | 0 |  |
|  | 1 | - | 0 | 0 |  |
| VI=IV+V |  |  |  |  |  |
| Factorizare | 1 | 0 | - | - |  |
| dublă |  |  |  |  |  |

Pas 2: Pas 3: Factorizare

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grupul |  |  |  |  |  |
| I | 0 | 1 | 1 | 1 |  |
|  | 1 | 0 | 1 | 1 |  |
| II | 0 | 1 | 1 | 0 |  |
|  | 1 | 0 | 0 | 1 |  |
|  | 1 | 0 | 1 | 0 |  |
|  | 1 | 1 | 0 | 0 |  |
| III | 0 | 1 | 0 | 0 |  |
|  | 1 | 0 | 0 | 0 |  |

**Mulțimea monoamelor maximale** este formată din toate monoamele corespunzătoare liniilor nebifate din tabel.

=

=

=

=

=

Pas 4: Identificarea monoamelor centrale

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | ⊛ |  |  |  |  |
|  |  |  |  |  | ⊛ |
|  | \* | \* |  |  |  |
|  |  |  |  |  | ⊛ |
|  |  |  |  |  | ⊛ |
|  |  |  | \* | \* |  |
|  |  | \* | \* |  |  |
|  |  |  |  | \* | \* |

Suntem în cazul II.

*Cazuri:*

* Cazul I: =>
* Cazul II: =>

Cazul III: =>

Pas 5: Indentificarea formelor simplificate

În tabelul de mai sus:

Se hașurează coloanele care conțin ⊛.

Se hașurează liniile ce conțin \* hașurate anterior.

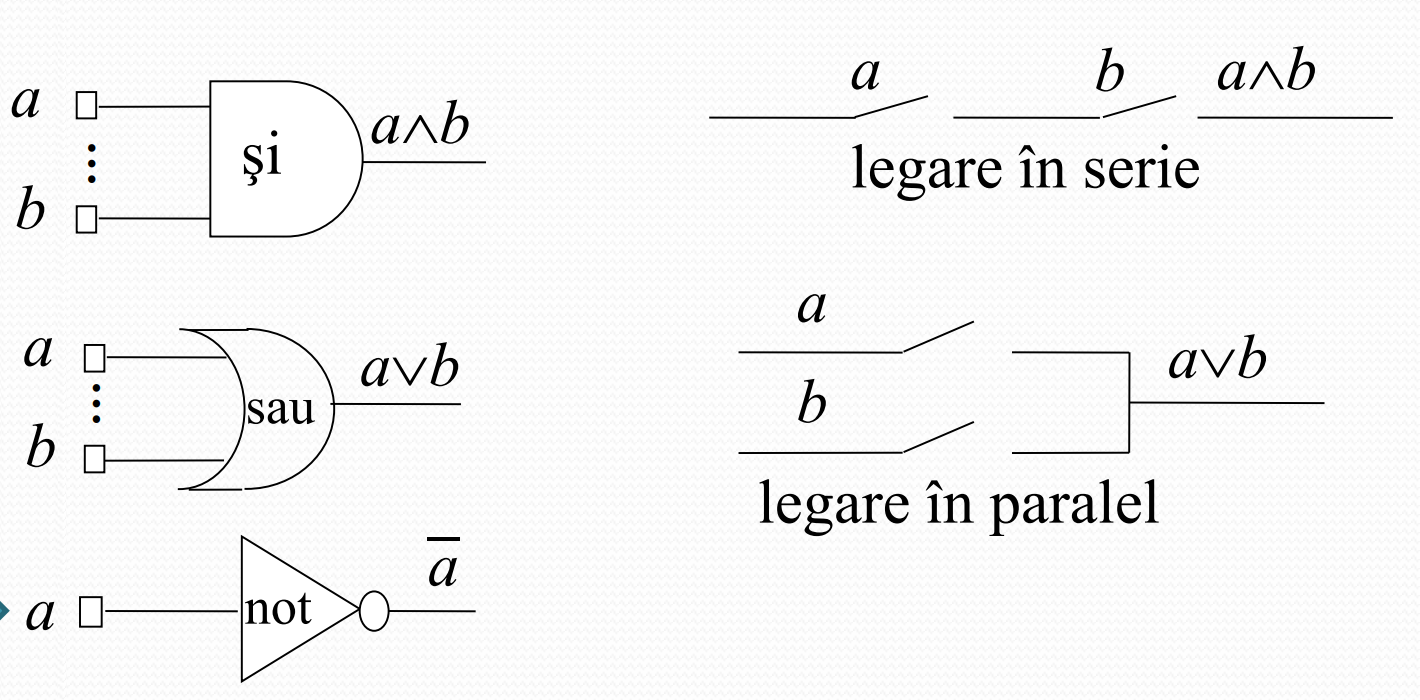
Se observă că cel mai simplu mod de a acoperi tot tabelul este:

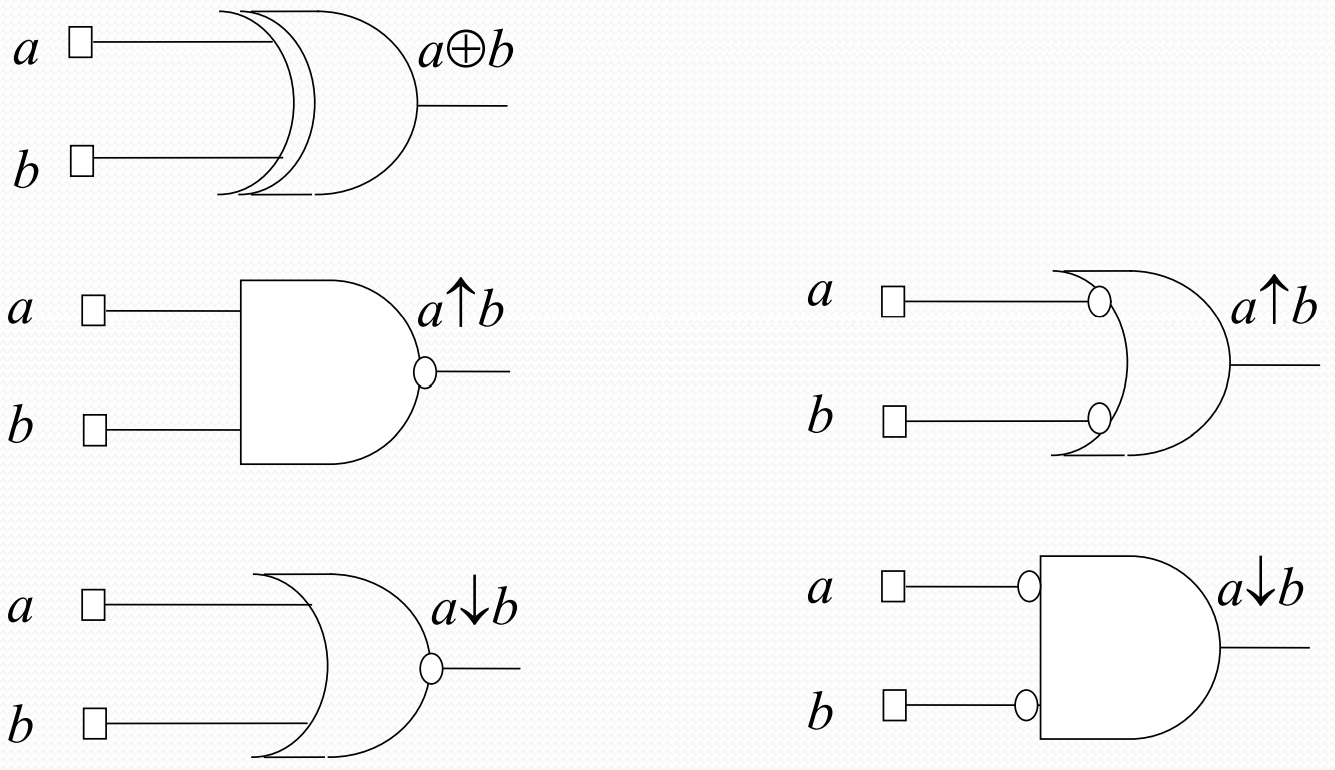
Rezultă că forma simplificată este:

LC\_Curs13.

Circuite logice

O **poartă** este un minicircuit logic care realizează una dintre operațiile logice de bază: , , ̅.

Porți logice

Porți derivate

Pasul 1: Identificarea intrărilor (variabilelor) și ieșirilor (funcțiilor)

* intrare: 4 cifre binare: .
* ieșire: pentru

Pasul 2: Construirea tabelei de valori asociate

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | FCD (cu un singur element) |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |  |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |  |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |  |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |  |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |  |

Pasul 3: Obținerea expresiilor funcțiilor (FCD de mai sus)

Pasul 4: Simplificarea funcțiilor

Pasul 5: Desenarea circuitului

...