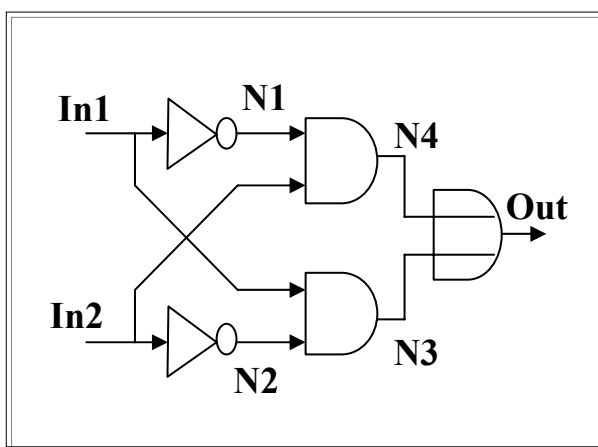


## 8. СИМУЛИРАНЕ С ПРОЛОГ

**Въведение.** Симулирането е често използван подход при изследване поведението на сложни системи. При симулирането се построява модел на системата и нейното поведение се изследва, като на входовете на модела се задават входни въздействия и след съответни изчисления се получават стойностите на сигналите на изходите на модела. Връзката между входовете и изходите в модела може да се опише с математически зависимости или да се получи от зададената структура на системата.

Най-просто е симулирането, когато изходите на системата са функция само на нейните входове. В такъв случай поведението на системата може да се изследва, като за всяка комбинация от стойности на входните сигнали се получи съответната комбинация от стойности за сигналите на изходите. Удобно е пълното изследване на поведението на такива системи да се оформи като таблица, в която в последователни редове се изреждат входните въздействия (подредени по подходящ начин) и за всеки ред се записва съответната изходна реакция на системата. Когато всички величини приемат само две стойности (0 и 1), тази таблица се нарича таблица на истинност.

**Постановка.** Да се симулира логическата схема XOR. На следващата фигура е показан един от възможните варианти за реализирането на тази схема.



При реализацията са използвани елементите НЕ(not), И(and) – N3 и N4, ИЛИ(or) – N5. Означенията на фигурата са за елемента и за неговия изход. Двата входа са означени с In1 и In2, двата НЕ елемента и съответните им изходи – с N1 и N2, двата И-елемента и изходите им – с N3 и N4, и накрая ИЛИ и изходът на схемата – с Out.

**Упътване.** За построяването на модела са необходими предикатите not\_, or\_ и and\_. Всеки от тях е от вида **предикат(вход1, вход2,..., изход)**.

Най-напред се описват като факти тези предикати. За всеки предикат описанието може да изглежда като последователност от редовете на неговата таблица на истинност. Например за предиката **and(вход1, вход2, изход)** фактите са четири: and\_(0,0,0).

and\_(0,1,0).

and\_(1,0,0).

and\_(1,1,1).

След това самата схема се описва като правило. За нашия случай това правило изглежда по следния начин:

```
xor_(Input1, Input2, Output) :-  
    not_(Input1,N1),  
    not_(Input2,N2),  
    and_(Input1, N2, N3),  
    and_(Input2, N1, N4),  
    or_(N3, N4, Output).
```

Цялата програма за симулиране на схема XOR изглежда така:

Програма:

```
not_(1,0).  
not_(0,1).  
and_(0,0,0).  
and_(0,1,0).  
and_(1,0,0).  
and_(1,1,1).  
or_(0,0,0).  
or_(0,1,1).  
or_(1,0,1).  
or_(1,1,1).  
xor_(Input1, Input2, Output) :-  
    not_(Input1,N1),  
    not_(Input2,N2),  
    and_(Input1, N2, N3),  
    and_(Input2, N1, N4),  
    or_(N3, N4, Output).
```

Цел:

Reconsulted from: C:\My Documents\Visual Prolog 6  
Examples\pie\EXE\log\_xor.pro

```
xor_(Input1, Input2, Output)  
INPUT1= 1, INPUT2= 1, OUTPUT= 0  
INPUT1= 1, INPUT2= 0, OUTPUT= 1  
INPUT1= 0, INPUT2= 1, OUTPUT= 1  
INPUT1= 0, INPUT2= 0, OUTPUT= 0  
4 Solutions
```

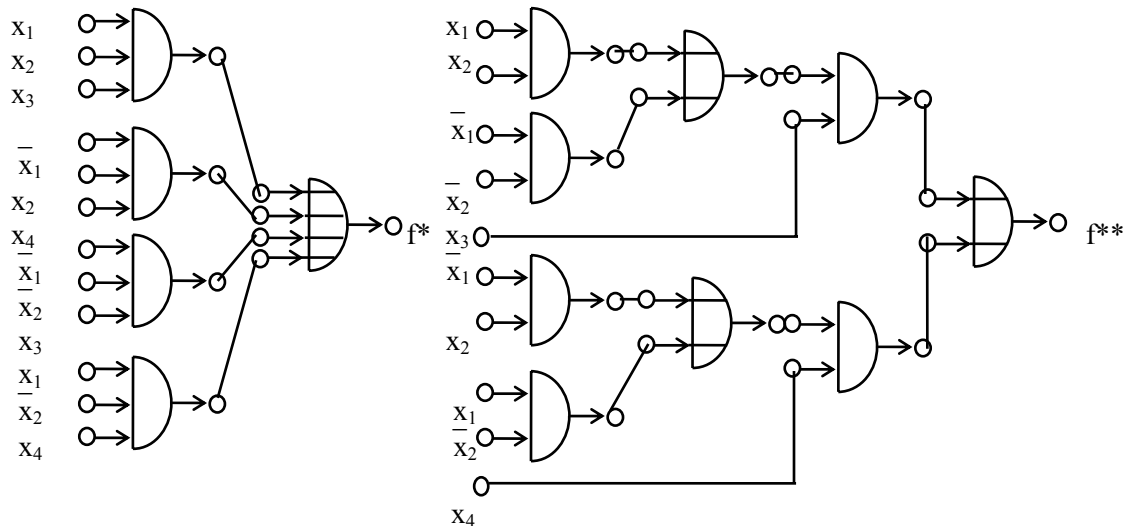
```
xor_(1,Input2, Output)  
INPUT2= 1, OUTFUT= 0  
INPUT2= 0, OUTFUT= 1  
2 Solutions
```

```
xor_(1,0, Output)  
OUTPUT= 1  
1 Solution
```

### Задание

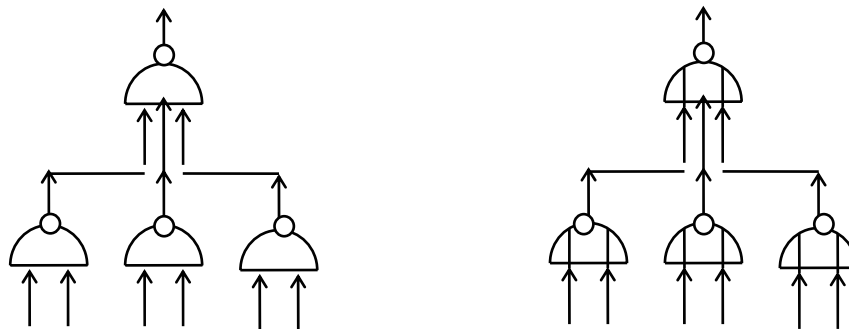
1. Разучете програмата и я трасирайте. Проследете работата на машината за извод.

2. Съставете програма за симулиране на следната схема с два изхода.



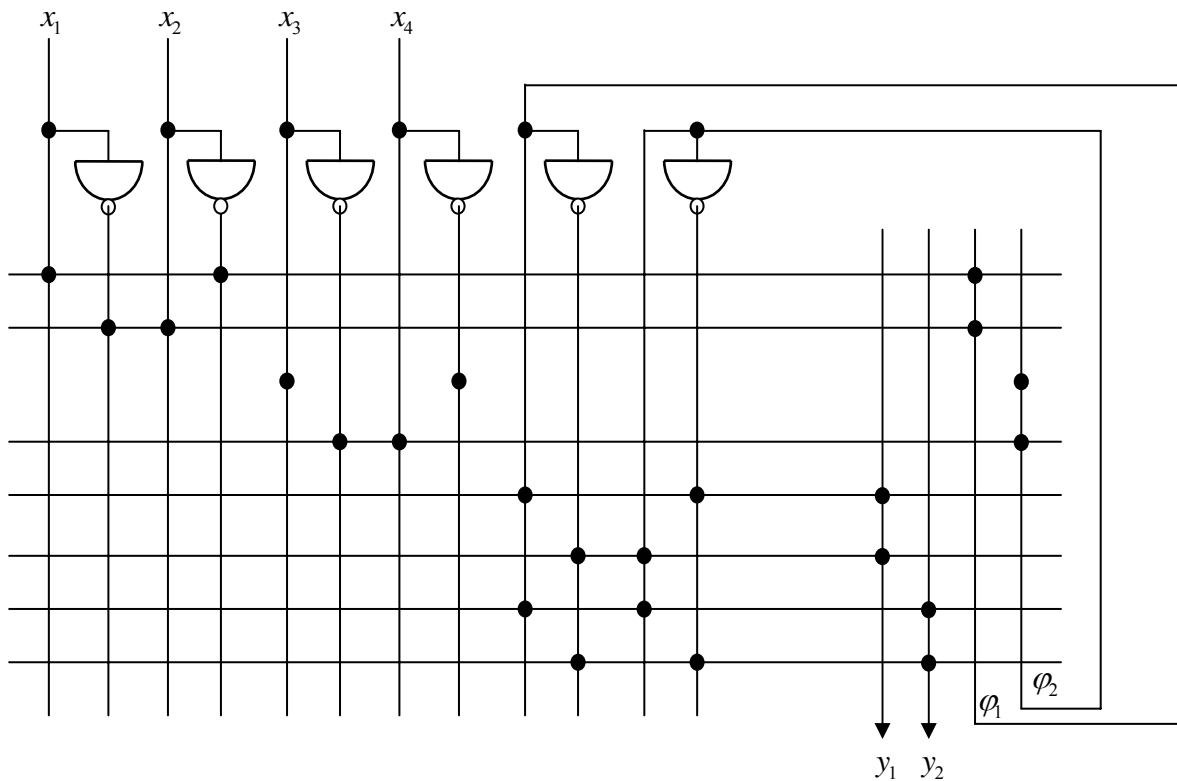
**Забележка.** Предварително означете всички връзки между елементите по подходящ начин. Покажете, че двете схеми реализират еднакви функции.

3. Съставете програма за симулиране на следната схема с два изхода.



**Забележка.** Елементите в лявата схема са НЕ-И (NAND), а елементите в дясната – НЕ-ИЛИ (NOR). Елементът НЕ-И (НЕ-ИЛИ) може да се замени от последователно свързани елемент И (ИЛИ) и елемент НЕ. На входовете се подават последователно отляво надясно величините  $x_1, x_2, x_3, \bar{x}_1, \bar{x}_2, \bar{x}_3$ .

4. Съставете програма за симулиране на следната схема с четири входа и два изхода.



Забележка. Схемата има ясно изразени две полета – ляво и дясно. Малките кръгчета означават наличие на връзка. Първите осем вертикални проводника носят входните сигнали от  $x_1$  до  $x_4$  (нечетни) и техните инверсни стойности (четни). Сигналът на хоризонтален проводник е логическо И от онези сигнали, чийто носещ проводник е свързан с хоризонталния проводник в лявото поле. Самият хоризонтален проводник може да се интерпретира като изход на И-елемент.

Вертикалните проводници в дясното поле представляват изходи на ИЛИ-елементи. Сигналът на вертикален проводник от дясното поле е логическо ИЛИ от онези сигнали, чийто носещ хоризонтален проводник е свързан с вертикалния проводник в дясното поле.

Обърнете внимание, че два изхода от дясното поле са и входове за лявото поле на схемата. За да сведете моделирането до предишните случаи, необходимо е да прекъснете тези връзки и да ги разглеждате като още два входа и още два изхода на схемата.