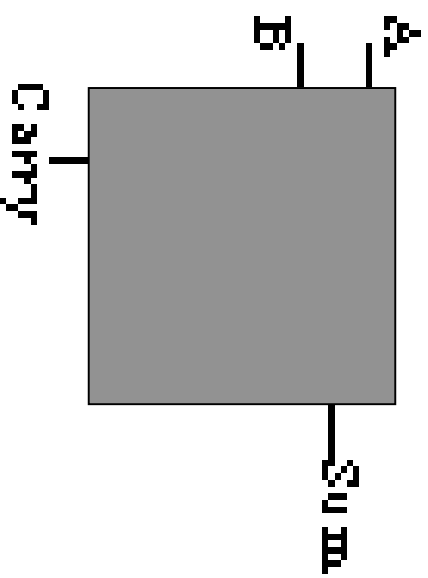


At bygge logiske kredse af komponenter

Test program

Eksempel: En »halfadder«

Lægge to bits A og B sammen:

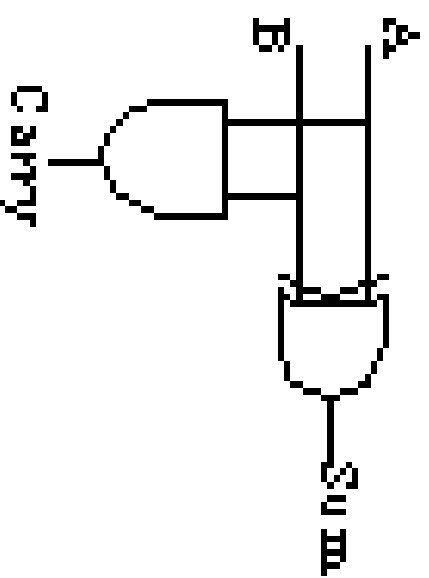


At bygge logiske kredse af komponenter

Test program

Eksempel: En »halfadder«

Lægge to bits A og B sammen:

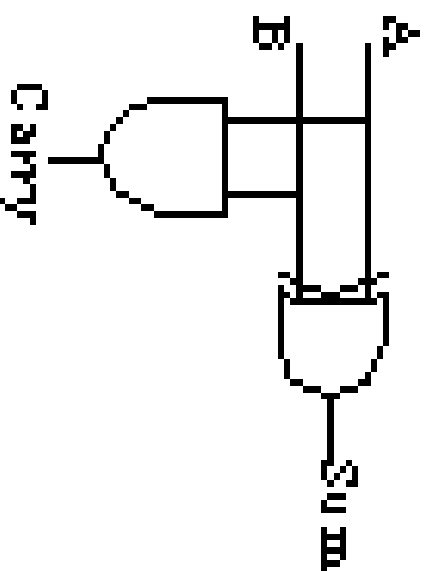


At bygge logiske kredse af komponenter

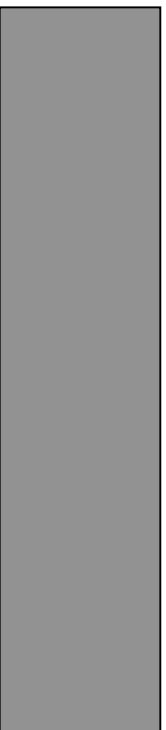
Test program

Eksempel: En »halfadder«

Lægge to bits A og B sammen:



halfadder(A, B, Carry, Sum) :-

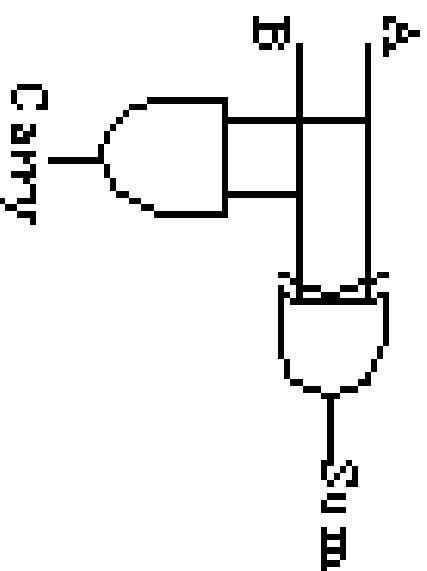


At bygge logiske kredse af komponenter

Test program

Eksempel: En »halfadder«

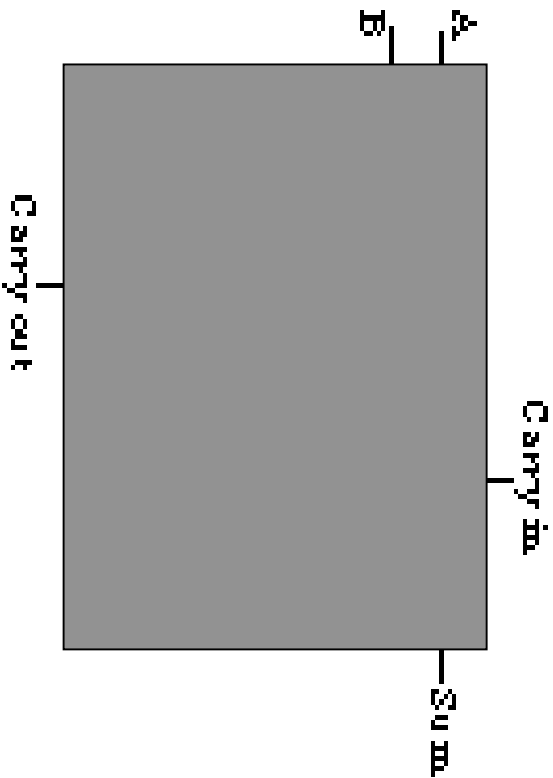
Lægge to bits A og B sammen:



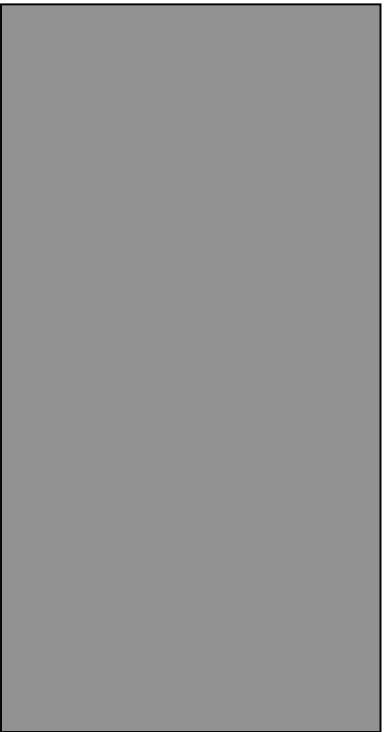
```
halfadder(A, B, Carry, Sum) :-  
    and(A, B, Carry),  
    xor(A, B, Sum).
```

Bemærk: Analogi mellem Prolog-variabel og elektrisk leder

En full-adder, nu med gammel mente

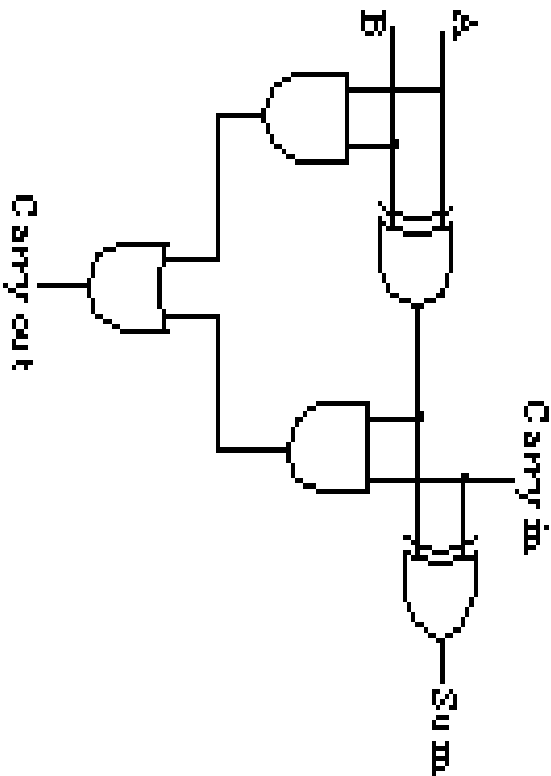


fulladder(A, B, Carryin, Sum, Carryout) :-

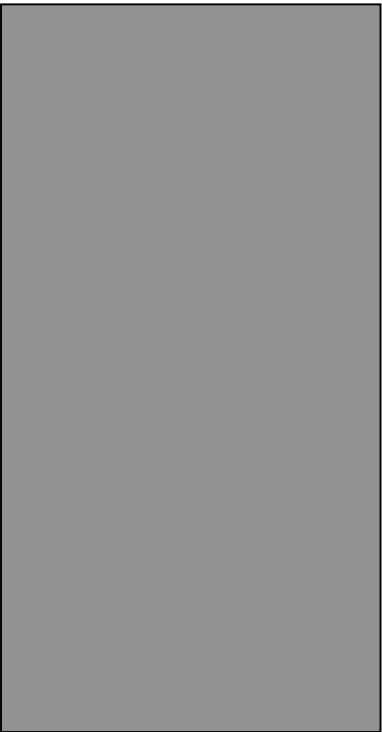


Test program

En full-adder, nu med gammel mente

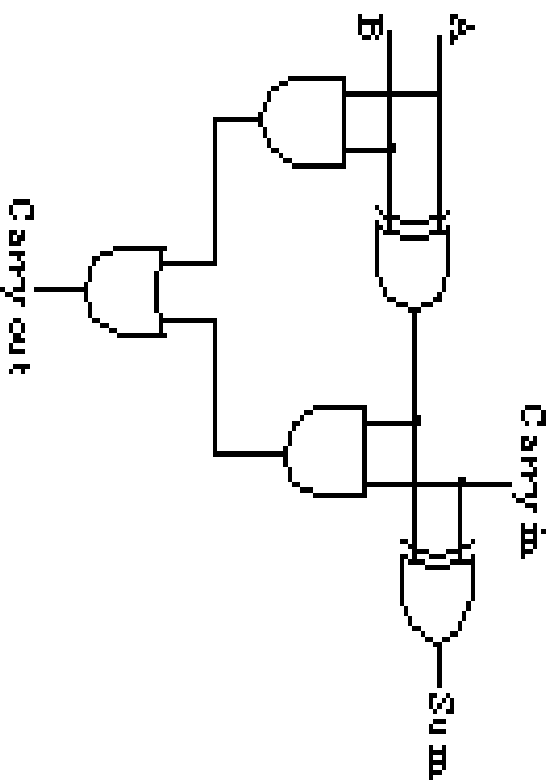


fulladder(A, B, Carryin, Sum, Carryout) :-



Test program

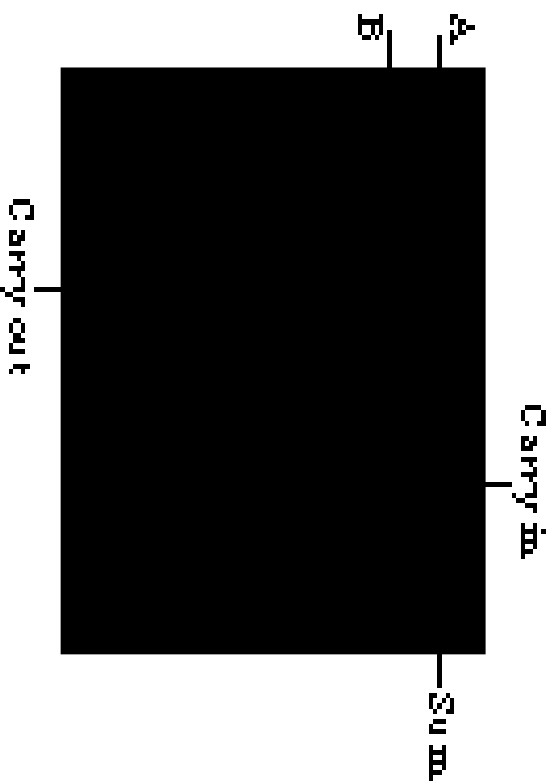
En full-adder, nu med gammel mente



```
fulladder(A, B, Carryin, Sum, Carryout) :-  
    xor(A, B, X),  
    and(A, B, Y),  
    and(X, Carryin, Z),  
    xor(Carryin, X, Sum),  
    or(Y, Z, Carryout).
```

Test program

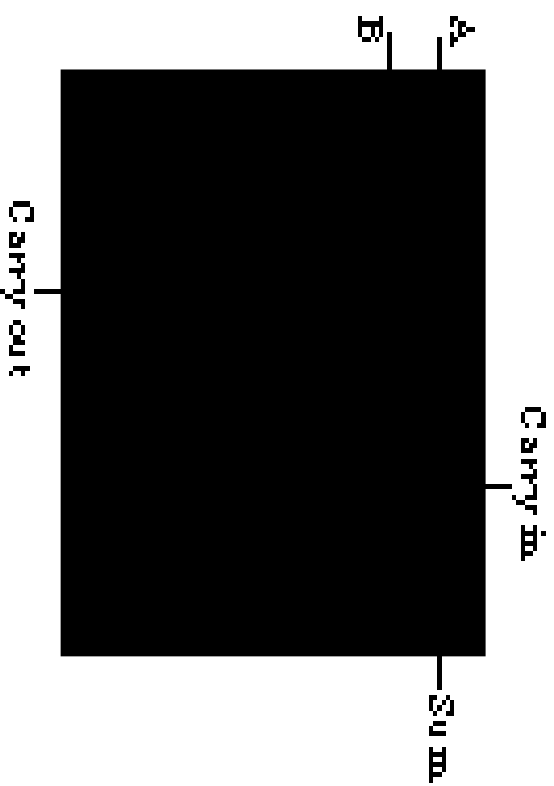
En full-adder, nu med gammel mente



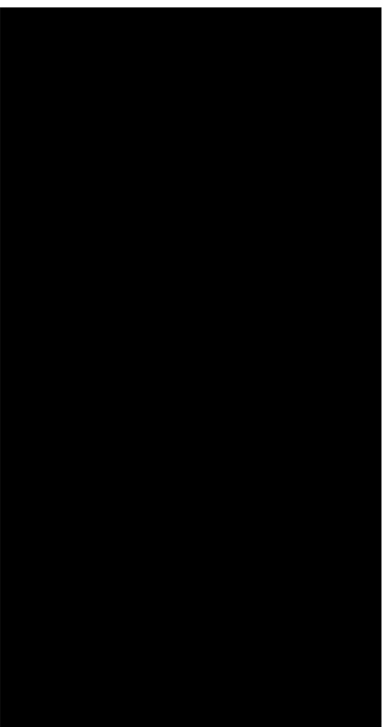
```
fulladder(A, B, Carryin, Sum, Carryout) :-  
    xor(A, B, X),  
    and(A, B, Y),  
    and(X, Carryin, Z),  
    xor(Carryin, X, Sum),  
    or(Y, Z, Carryout).
```

Test program

En full-adder, nu med gammel mente



fulladder(A, B, Carryin, Sum, Carryout) :-



Test program