

PLD-design med udviklingsprogrammet WinPLACE v2.3.2

En PLD er en programmerbar kreds. PLD står for Programmable Logic Device.

Programmet WinPLACE fra firmaet ICT kan bruges til at bestemme, hvad man vil programmere ind i en PLD. Eller rettere, hvad man vil udnytte af de allerede indbyggede muligheder i kredsen. Der er et hav af muligheder, og når en pris på under 20 Kr. tages i betragtning, er det imponerende, hvad mulighederne er.

WinPLACE kan bruges til design af et hav af PLD'er fra ICT. Den PLD, vi bruger, hedder PEEL 18CV8. Det er programmeringen af en sådan, dette kompendium beskriver.

Vha. af programmet kan man mere eller mindre grafisk udvælge de funktioner, der skal udnyttes i kredsen, og fx med boolsk algebra kan man beskrive en funktion for en udgang.

PEEL står for Programmable Electric Erasable Logic. 18 angiver det maksimalt antal indgange, og 8 de maximale udgange. Da det drejer sig om en 20 pin IC, må udgangene altså også kunne fungere som indgange!

Program og dokumentation herfor kan findes på ICT's hjemmeside, <http://ictpld.com/>
I 2005 blev ICTPLD overtaget af Anachips, <http://www.anachip.com/>

For at downloade Winplace, skal man angive en masse oplysninger, derfor har jeg lagt en fil til download på min hjemmeside.

Download / PEEL / Install Winplace v. 2.3.2 Beta

Dette kompendie opdateres løbende. Er der forslag til retning af fejl, eller forslag til forbedringer, kontakt mig venligst.

Dokument-historie:

18/9 2005 Revideret.

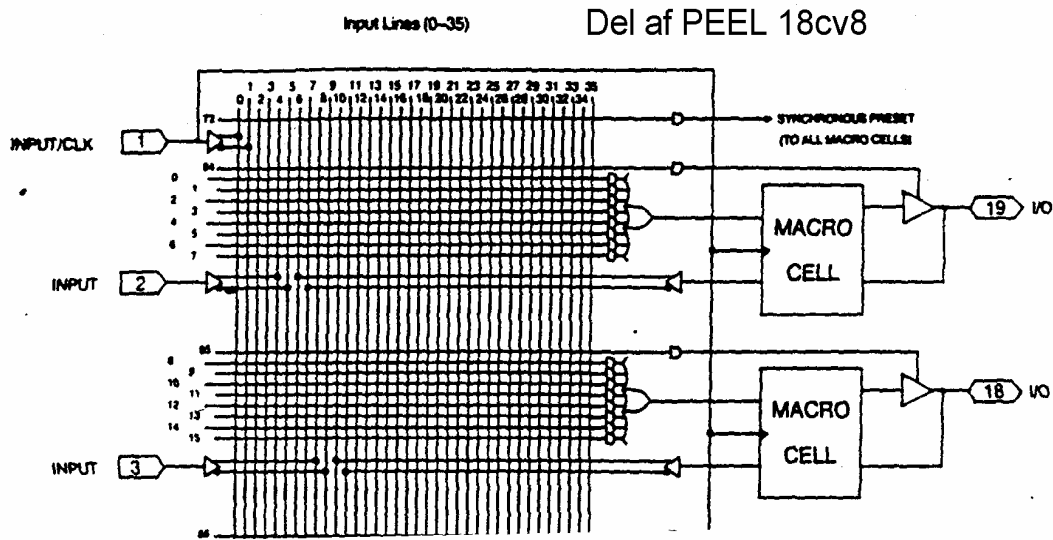
Oktober 06, Sandhedsskema programmering af 18CV8 adderet.

2/10-07, Tilrettet

/ Valle

WINPLACE

Først lidt generelt om Programmerbar Array Logic. PAL.



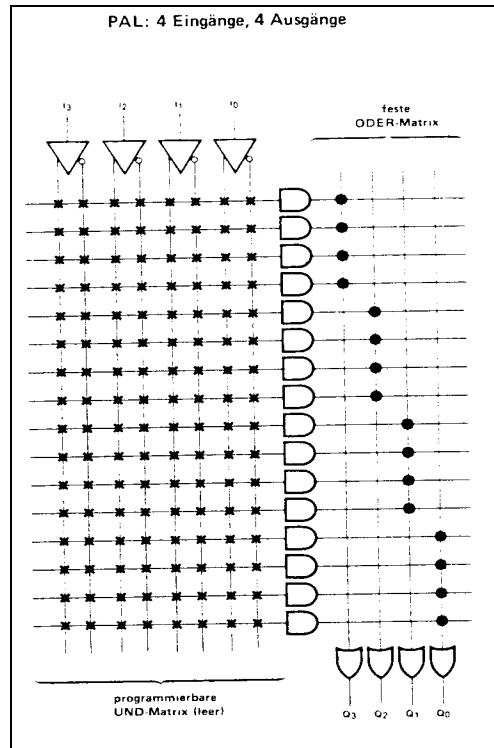
En del af 18CV8 med AND/OR-matrix og MACRO-celle

Udgangene fra hver af de 8 and-gates til højre for matrix-et er ført til en 8-input or-gate. OR-gatens udgang går videre mod udgangen.

AND-gatene har 8 inputs. Hver af disse 8 kan programmeres til at have forbindelse til 1 af kredsens input, eller den inverterede input.

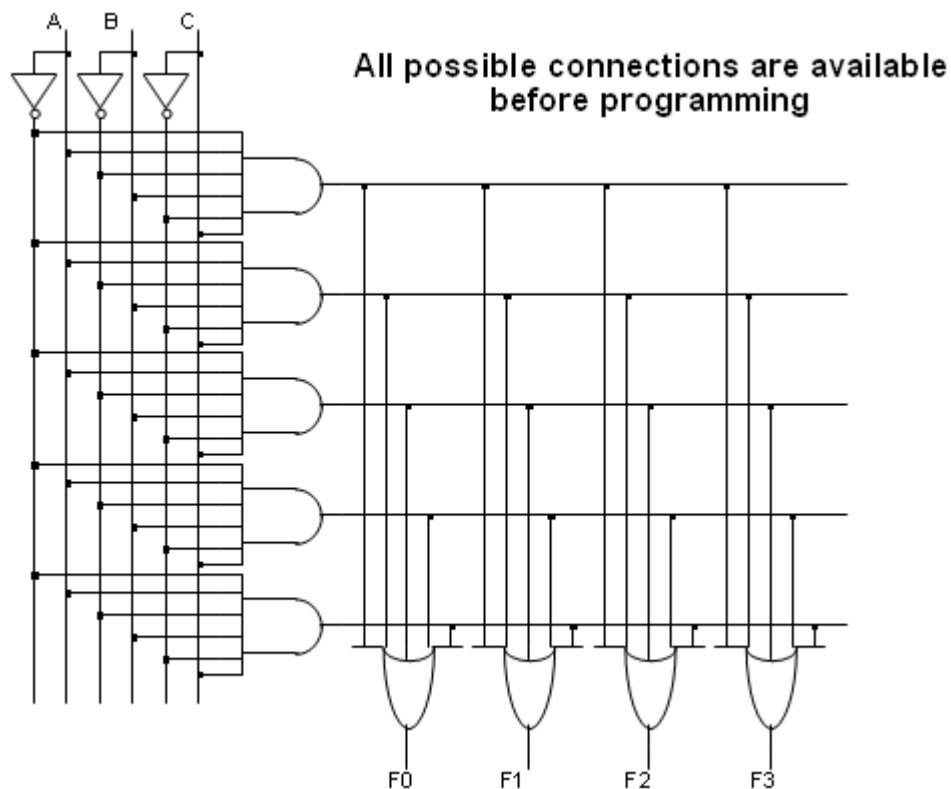
Det er lidt tydeligere på følgende billede:

Figuren til højre viser en PAL. Her ses at OR-matrix-et er fast programmeret, og indgangene til AND-matrix-et er programmerbart.

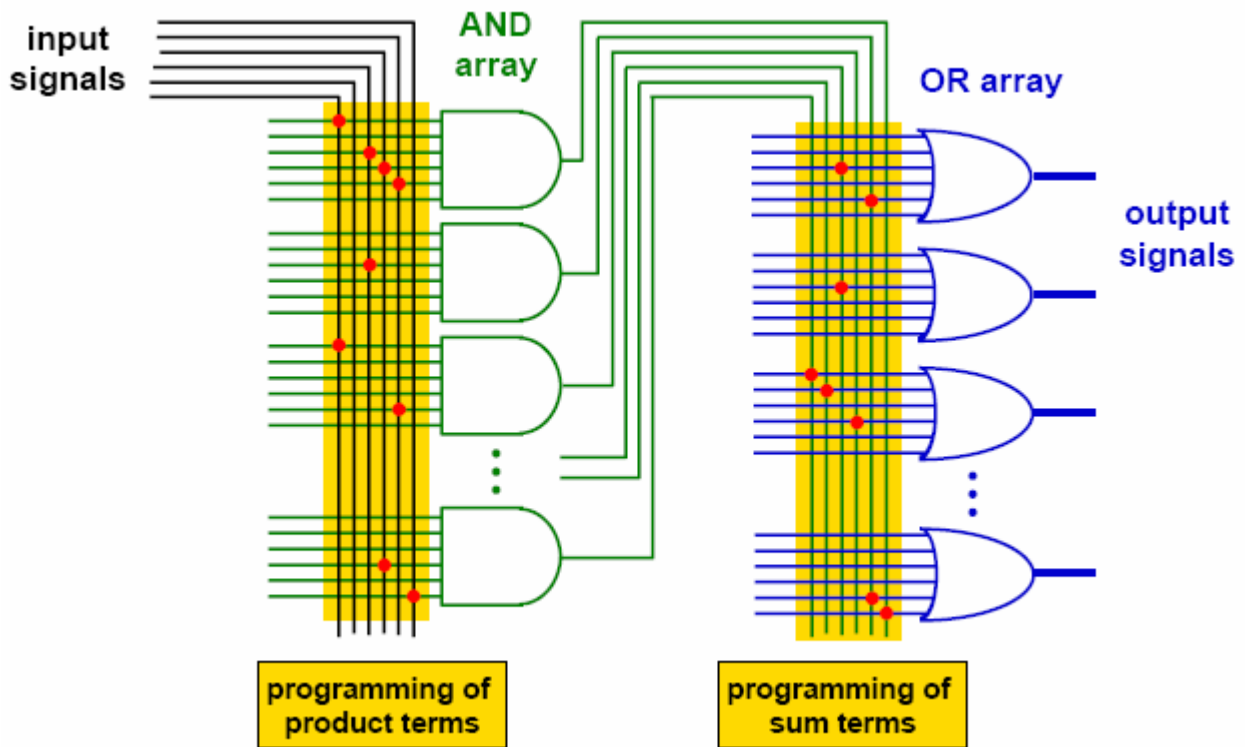


Figuren viser en PAL. Bemærk fast fortrådet OR-matrix. Der er kun vist 1 af de mange indgang i and og or-gatene !

Her er vist alle AND-gates – input, og input til udgangenes OR-gate.

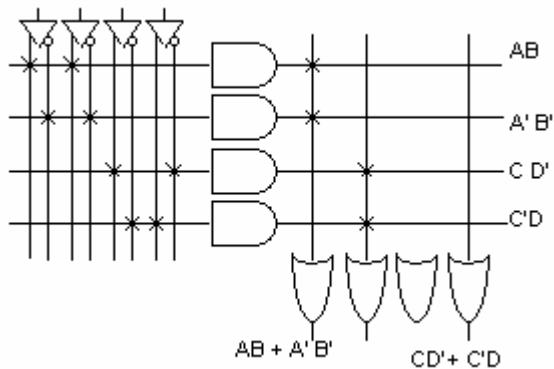
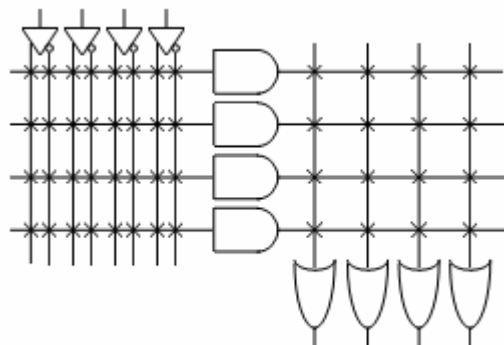


WINPLACE



AND-gatene har 8 input, der dog ikke er tegnet.

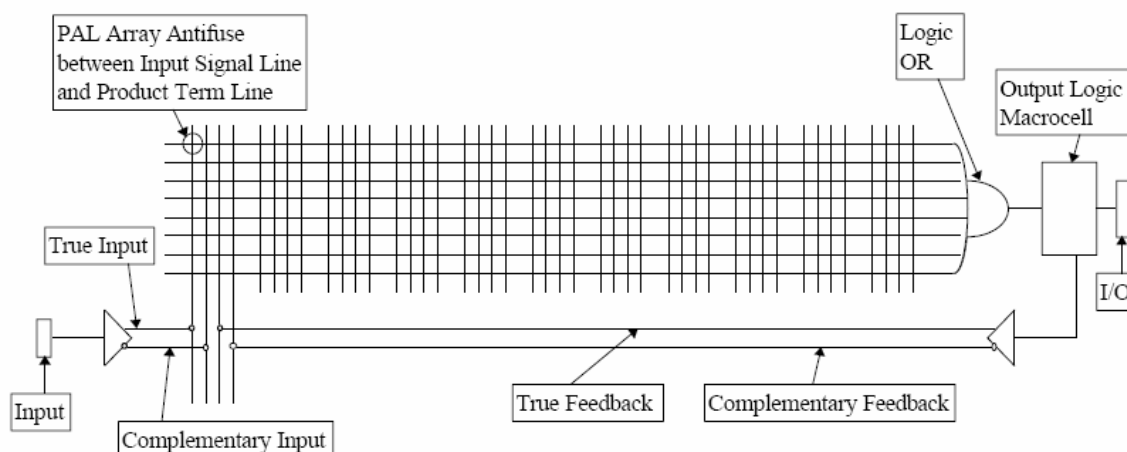
Ligeledes er Output'ene med flere inputs.



Venstre OR-gates udgang har ligningen

$$AB + \bar{A} \bar{B}$$

WINPLACE



WINPLACE

Winplace er beregnet til at udvælge det, i den programmerbare kreds, man ønsker at anvende.

Følgende skitse viser en del af IC'ens indre. Der ses et input-matrix, og en makrocelle foran hver udgang. Vha. Winplace programmeres med boolske udtryk hvilke indgange, der skal føres til hvilken macrocelle. Macrocellerne kan vha simple museklik konfigureres på et utal af måder, fx helt udelades, så resultatet af den boolske ligning føres direkte til udgangen.

WinPLACE er baseret på en gammel DOS-version. Programmet består af 3 del-programmer, **Design**, **Compile**, og **Simulate**. I designdelen bestemmer man kredsens indhold, Compile-delen oversætter til en fil, der vha. programmeringsudstyr kan lægges ind i en PEEL-kreds, og med Simulate kan man tjekke om resultatet af design-delen blev som man ønsker.

Når WinPLACE startes, ses et billede af den aktive type PLD. Programmet har valgt en default kreds.

Vælg File / New, og vælg **18CV8**.

Eller man kan selvfølgelig åbne et gemt projekt.

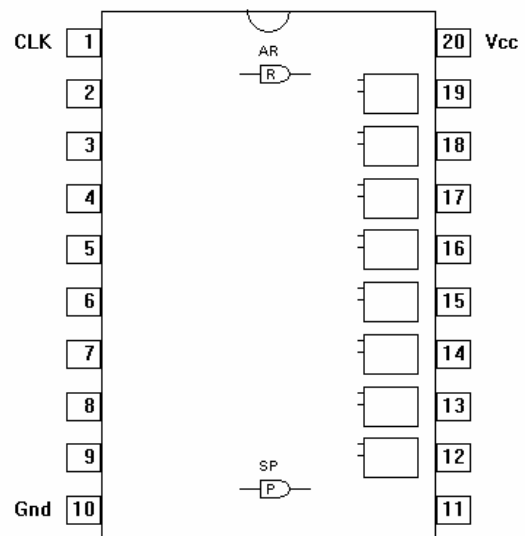
Der fås nu et vindue, hvor der øverst ses et grafisk billede af kredsen, kaldet **DESIGN Window**, og nederst et tekst-vindue kaldet **DESCRIPTION Window**.

WINPLACE

I **designvinduet** kan man grafisk konfigurere et design. Fx hvad man vil udnytte af kredsen, og hvordan outputbenene skal kobles sammen med kredsens indre.

Og man kan tildele inputs og outputs navne, "LABELS".

I **Tekstvinduet** angives fx boolske ligninger for hvilke inputs, der skal føre til et outputben. Ligeledes kan der indskrives kommentarer, så designet senere kan genkendes eller genkaldes, hvis man meget senere skal revidere et design.



```
TITLE ''
DESIGNER ''
DATE ''

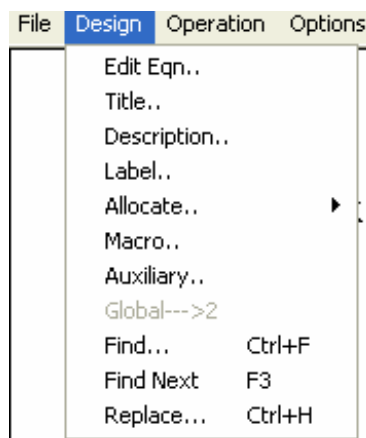
Description
  Enter description here ...
End_Desc;
```

Øverst: Design-Window, nederst Description Window

Skillelinien mellem de to vinduer kan flyttes:

Vælg **Design / Title** Herved hoppes automatisk ned i tekst-vinduet til det rigtige sted, titlen på et design kan indskrives. Eller man kan blot i stedet finde det rigtige sted i description-vinduet, og skrive titlen her.

Indtast en sigende titel for designet mellem de to apostroffer ' '



Vælg derefter

Design / DESCRIPTION og lav en beskrivelse af det design, der skal ind i kredsen.

Det er blot en beskrivelse, der skal bruges til at huske, hvad det er, kredsen skal bruges til.

WINPLACE

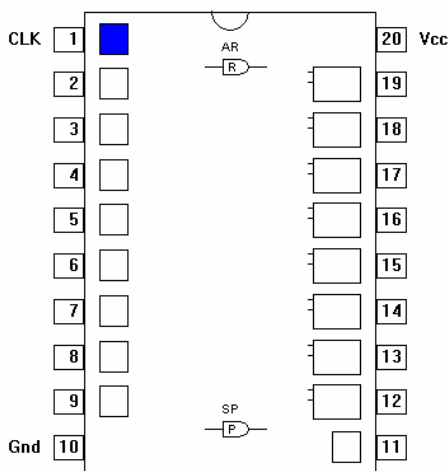
Indskriv tilsvarende også **forfatter** og **dato** mellem de to apostroffer ' ' på de respektive linier.

Man kan enten selv bladere ned i tekstfilen, og vælge det korrekte sted at indskrive data, eller man kan vælge menu-punktet i Design Vinduet.

LABELS

Næste opgave er at tildele de indgange og udgange, der skal bruges, nogle labels. Dvs. nogle navne, der kan refereres til i boolske ligninger.

Input-ben kan fx hedde A, B osv. Og output-ben kan fx døbes O1, O2, --, eller F1, F2 osv.



Vælg øverst i designvinduet **Design / Label** for at starte navngivning af indgange og udgange.

Klik på den pin, der skal tildeles et navn.

Inden for PIN'ene for både indgange i venstre side og indgange, / udgange i højre side ses nogle kasser. De bliver blå hvis musen føres hen over.

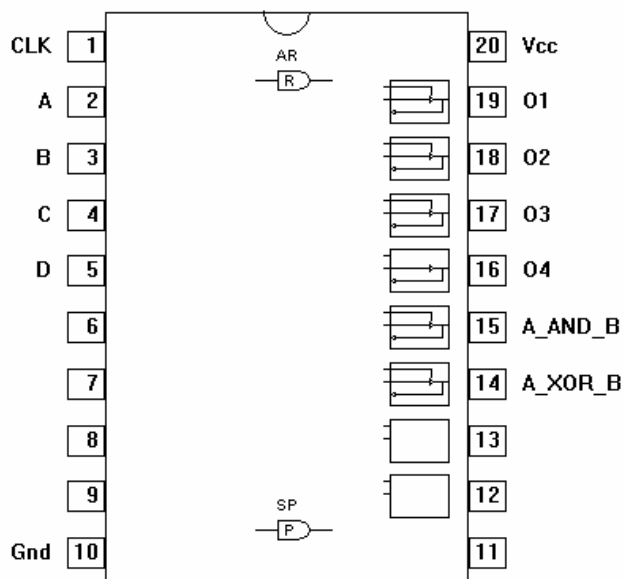
Klikkes på en, får man mulighed for at indtaste en label, dvs. et navn for signalet på pågældende pin i et vindue øverst til højre på skærmen.

I vinduet ses hvilken pin, der er valgt, og der kan indtastes en label, dvs. et navn, for pågældende pin.

Ingen labels må være ens. Det er vigtigt, ikke at bruge danske bogstaver. !!

I/O Pin 16:
03

Eksempel på et design med navngivne pins.



Input-benene til venstre og output-benene til højre på IC'en har fået Labels.

Input og Outputs kaldes IOC's

På skærmen ser det ud som fx dette:

Det ses, at der før hver output sidder en blok. Denne kaldes for en Macrocelle. Denne kan konfigureres grafisk, blot ved at vælge sig frem. Se herunder.

I tekstruden kan nu ses en beskrivelse af pin-numrene med deres tilhørende labels.

Også pin'ene på højre side kan konfigureres som inputs!!

Med Esc forlades Label-mode !

Konfigurering af OUTPUT MACROCELLE.

Næste step er at konfigurere de forskellige outputs.

Imellem GATE-Matrixen og Output'ene er der en Macrocelle, der kan konfigureres på forskellig vis. Det sker ved blot at vælge med museklik.

Klik på en macrocelle. Herved åbnes den, og man ser et billede af hvordan den er konfigureret. Der er vist et antal @'er. Klikkes på en af dem, ændres konfigurationen, eller der fremkommer et blåt valgvindue, hvor man skal vælge konfiguration. Efterfølgende kan resultatet umiddelbart ses.

WINPLACE

Når der klikkes på en macrocelle første gang ses default dette billede:

Det ses, at det signal, der kommer fra B, føres ud til udgangen via en styret buffer. Hvis A er høj, føres signalet gennem!

Signalet fra B er resultatet af en boolsk ligning, hvori indgangs-labels indgår som variable. (se senere)

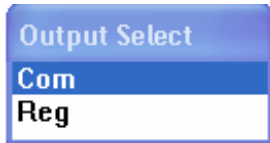
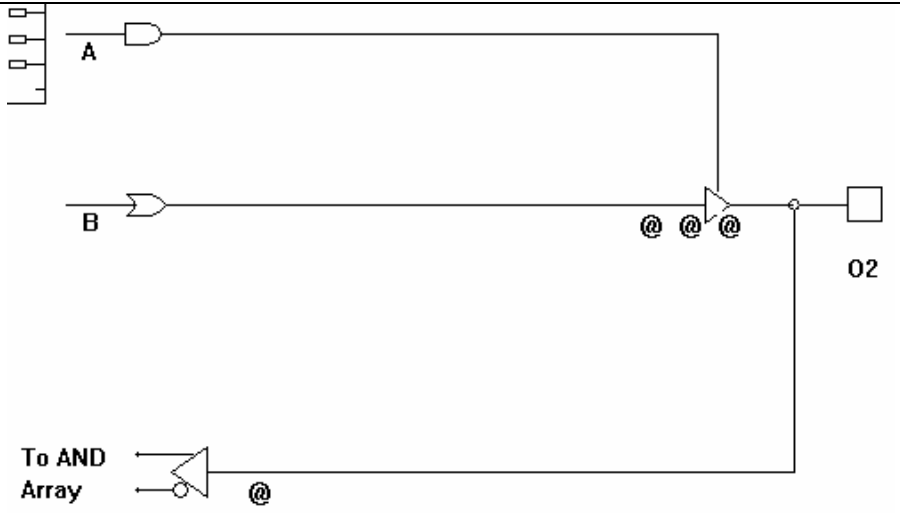
De tre @ ved udgangen hedder fra venstre:

Output Select, Output Enable Select, og Output Polarity

Klikkes på en af dem, kan udgangens konfiguration ændres.

De fleste af mulighederne gennemgås i det følgende:

Output Select

	<p>Valg af Output.</p> <p>Skal output til output-pin'en komme direkte fra en Or-matrix, (COMbinatorisk) eller skal der kobles et REGister, en D-FF ind før outputtet.</p>
	<p>Her er valgt "Com"</p> <p>Der står for kombinatorisk, dvs gate-kredsløb.</p> <p>Output kommer direkte fra gate-matrixen.</p>

WINPLACE

	<p>Vælges ”Reg” indsættes et register, en D-FF i signalets vej til outputtet.</p> <p>Her er det så nødvendigt at definere det signal, der skal føres til D.</p> <p>Clk tages altid fra pin 1. Dette kan ikke ændres.</p> <p>Syncron Preset, og Asyncron clear kan defineres med boolske ligninger. Det sker i teksteditoren.</p>
--	---

Output Enable:

	<p>Her vælges, om der fx skal være et Output Enable signal.</p>
	<p>Vælges I/O kan der defineres en boolsk ligning til at enable output'et. Er outputtet ikke enablet, er den tristatet. Dvs. høj-impedant.</p>

WINPLACE

	<p>Vælges Output Enabled, vil output være enabled hele tiden, hvilket betyder, at signalet fra B føres direkte ud til udgangen.</p>
	<p>Vælges Output disabled, er outputtet frakoblet. Pin'en kan bruges som input !</p>

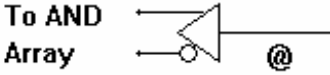
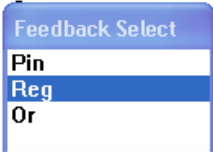
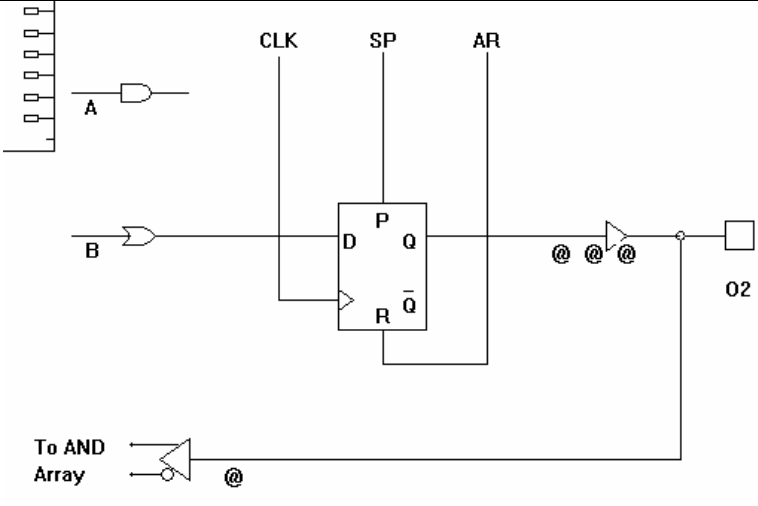
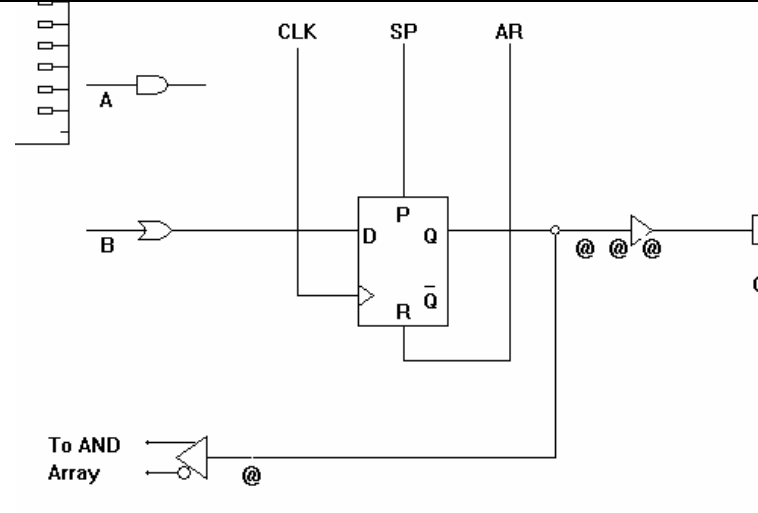
Output Polarity

	<p>Med @ længst til højre kan vælges, om Output til Pin'en skal tages direkte eller inverteres.</p>
	<p>Output direkte</p>
	<p>Output er inverteret.</p>

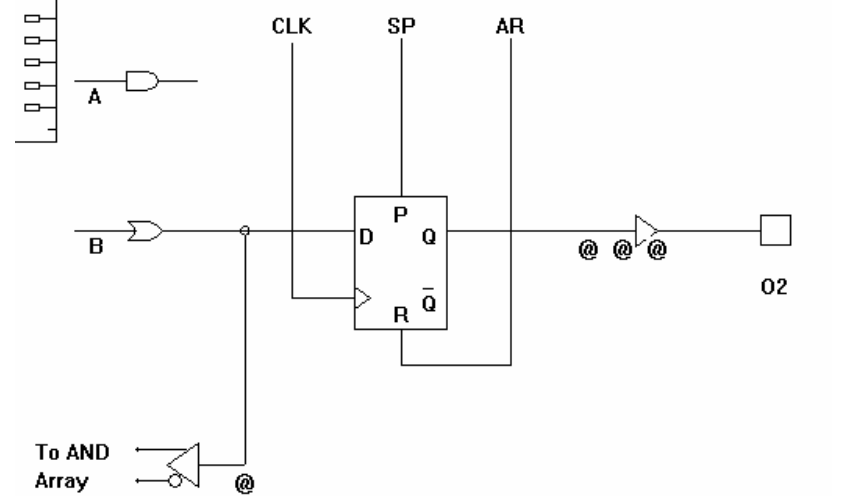
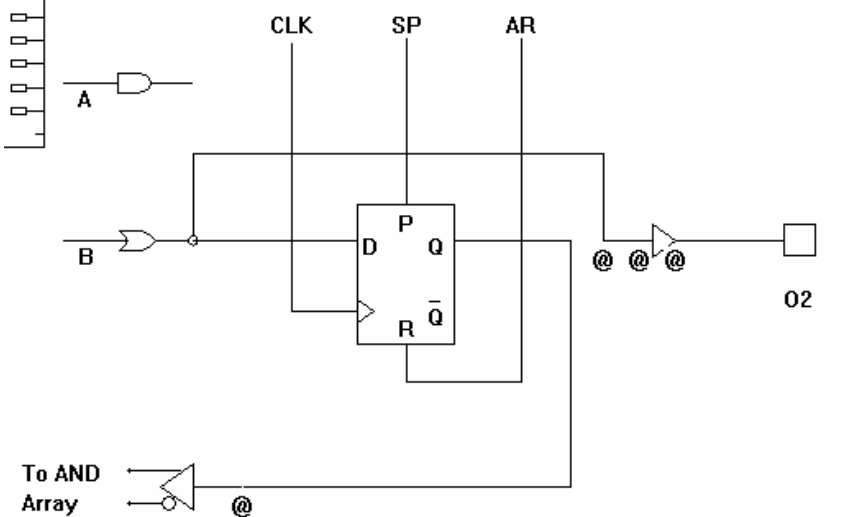
Feed Back Select

WINPLACE

Med feedback menues, at signalet på udgangene (kan) føres tilbage til indgangsmatrixen, og indgå i de boolske ligninger.

 <p>To AND Array</p>	<p>Med den sidste @ vælges feedback signalet. Dvs. - Hvorfra skal udgangssignalet, der føres tilbage til matrix-en komme fra.</p>
	<p>Der er følgende muligheder for at konfigurere feedback.</p>
 <p>CLK SP AR</p> <p>A B</p> <p>D P Q R Q̄</p> <p>To AND Array</p> <p>02</p>	<p><u>PIN</u></p> <p>Her er der valgt, at feed back skal komme direkte fra output-pin'en.</p>
 <p>CLK SP AR</p> <p>A B</p> <p>D P Q R Q̄</p> <p>To AND Array</p> <p>02</p>	<p><u>REG</u></p> <p>Her kommer output fra register-outputtet. Register-outputtet føres her også til output-pin'en.</p>

WINPLACE

 <p>The diagram shows a D flip-flop with inputs D, P, R, and \bar{Q}, and outputs Q and \bar{Q}. It is clocked by CLK. The D input is connected to the output of an OR gate with inputs A and B. The P input is connected to the output of an AND gate with inputs A and B. The R input is connected to the output of an AND gate with inputs A and B. The \bar{Q} output is connected to the D input. The Q output is connected to an OR gate with inputs A and B. The output of this OR gate is connected to a 02 pin. A connection point labeled '@' is shown for the AND array.</p>	<p><u>OR</u></p> <p>Det er muligt at bruge D-signalet fra OR-matrixen og føre det retur til matrixen.</p>
 <p>The diagram shows a D flip-flop with inputs D, P, R, and \bar{Q}, and outputs Q and \bar{Q}. It is clocked by CLK. The D input is connected to the output of an OR gate with inputs A and B. The P input is connected to the output of an AND gate with inputs A and B. The R input is connected to the output of an AND gate with inputs A and B. The \bar{Q} output is connected to the D input. The Q output is connected to an OR gate with inputs A and B. The output of this OR gate is connected to a 02 pin. A connection point labeled '@' is shown for the AND array.</p>	<p><u>REG</u></p> <p>Her er der brugt en D-FF til Feed Back selvom output til pin'en er taget fra D-indgangen til FF'en.</p>

Start design

Start nu med at vælge en output MACRO-Celle, der har fået en label.

Vælg fx viste konfiguration:



Dvs. der nu ikke er indsat et register, D-flip flops eller andet i signalets vej fra or-gatematrixen til udgangen. Output-macrocellen er her konfigureret til at tage inputtet direkte fra en OR-matrix og føre det direkte ud til OUTPUT-Pin'en. Output O1 signalet føres ligeledes tilbage til matrixet, og kan indgå i boolske ligninger for andre output.

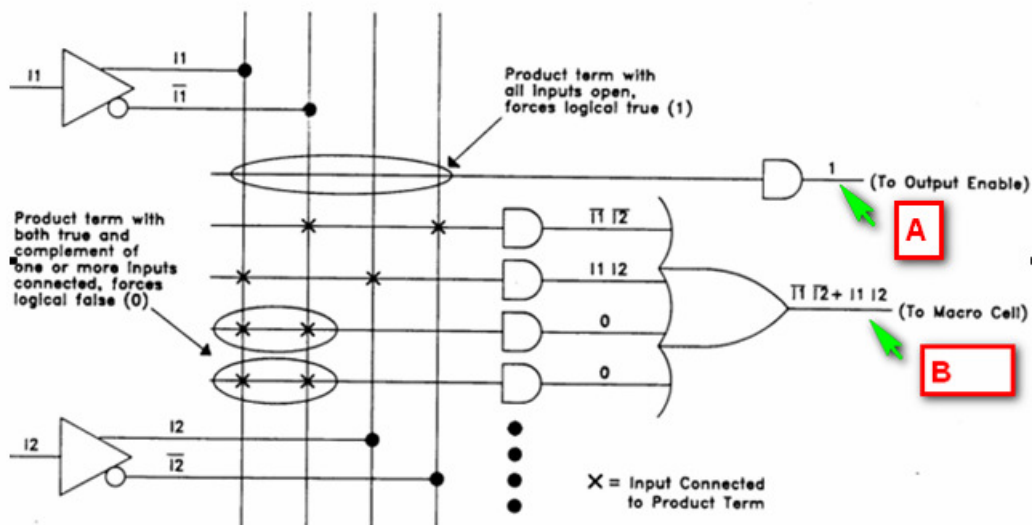
Det signal, der nu skal beskrives og føres til udgangen med labelen O1 er i billedet ovenfor vist som B.

Efter konfiguration, fører Esc eller højre mus retur til startbilledet.

BOOLSKE Ligninger

Vha. af Boolske ligninger skal man nu beskrive det signal, der skal føres til udgangen.

Alle indgange er i kredsen ført til en matrix. Med en boolsk beskrivelse af udgangen kan WinPlace skabe forbindelser på den rigtige måde i matrixen. I dette eksempel ledes resultatet så direkte til udgangen. Men det kunne også have været ført til D på en D-FF og herved videre til udgangen efter et clocksignal.



Skitsen viser to af indgangene, her kaldet I1 og I2. Efter en buffer, føres indgangen og også dens inverterede ind i et netværk, en matrix. Dvs. alle indgange har forbindelse til en "lodret" ledning. De viste krydser skal illustrere, at andgaten til højre "ANDer" alle de lodrette signaler, hvor der er et kryds. AND-gatene føres så videre til en orgate, og videre til udgangen. (Evt til en macrocelle før udgangen)

Forbindelserne i matrixen defineres vha. boolske ligninger for OR-gatens udgang.

Den viste konfiguration har ligningen: $F = \bar{I1} \cdot \bar{I2} + I1 \cdot I2$. Ligningerne skal foreligge som " En sum af produkter ". Dvs. få formen $Fx \quad F = A \cdot B + A \cdot \bar{C} + B \cdot \bar{C} \cdot D$. (Alle led, der And'es, kaldes produkter. I viste ligning er der 3 produkter.!)

Der er flere input til en macro-celle, der kan, eller skal defineres, afhængig af valgt konfiguration.

Clock er altid ben 1

Reset kaldet **AR**, (Asynkron Reset) kan defineres som en boolsk ligning, eller blot som en label, altså en pin. Når Reset produkt-term er sand, dvs. høj, vil outputtet af makrocellen blive reset, altså gå lav, asynkron. (Asynkron vil sige straks !!)

Preset, kaldet SP, (Synchron Preset). Når Preset produkt-term er sand, dvs. høj, vil outputtet af makrocellen gå høj synkront, med stigende clock flanke.

Boolske ligninger

Vælg nu **Design / Edit Eqn**

Dobbelt-Klik på or-gaten til venstre for den macrocelle, der skal laves en ligning for. Orgaten symboliserer or-matrixen !

I tekstvinduet hopper tekst-editorens cursor nu ned til det sted i teksten, hvor ligningen skal skrives. Stedet er angivet med pågældende outputs label.

Der er i teksten kun vist de input, der er relevant for den valgte Macro-celle-konfiguration:

Fx ser det således ud.

```
O3.And = 0;  
O3.SumB = 0;
```

O3 er den label, der er sat på pågældende output pin.

'O3'-cellens kombinatoriske ligning er default = 0. I stedet for '0'-et skrives den ønskede ligning.

WINPLACE

Output O3 skal fx være: $O3 = A * B$:

O3.And = 0;
O3.SumB = A*B;

De tegn, der kan bruges i ligningen er som flg.:

Operator	Alternativ	Funktion	Eks.	Prioritet
()	Ingen	Logisk organisering		1
!	/	Inverteret	/A, !A	2
&	*	AND	A*B;	3
#	+	OR	A+B;	4
\$	Ingen	Eksklusive OR	A\$B; !(A\$B)	5

O1.And = 0;
O1.Com = A*B;

I dette eksempel er indgangen til Macrocellen hørende til O1 defineret til at være en AND-funktion af input A og B.

O2.OE = 0;
O2.Com = 0;

.COM viser, at der er valgt signal direkte fra det kombinatorisk kredsløb.

O3.OE = 0;
O3.Com = 0;

Beskrivelser af evt. andre output med defineret label kommer i alfabetisk orden i tekstvinduet!

O4.And = 0;
O4.Com = 0;

Ligningerne kan være længere, fx som følgende:

$$\begin{aligned} O3.COM = & A * B * C * D + \\ & A * /B * C * D + \\ & A * B * C * /D + \\ & A * B * /C * /D; \end{aligned}$$

Gem Designet.

Først gemmes designet !! Opret et nyt bibliotek til det nye projekt, og vælg et sigende navn til designet. !! **OBS, der må ikke være mellemrum i navnet !!!!**

Kompilering, Se senere.

Definering af indhold vha. sandhedstabel.

I stedet for logiske udtryk kan logikken beskrives med sandhedstabeller.

Start et nyt projekt. Vælg 18CV8

Giv input-benene labels, fx A, B, C, og outputbene fx F1, F2 (Design / Label)
Højreklik returnerer !

Herefter vælges Design / Allocate / Truth Table / Add.
Giv sandhedstabellen et navn, fx Tabel1

Øverst ses nu følgende knapper:	INPUT	OUTPUT
---------------------------------	-------	--------

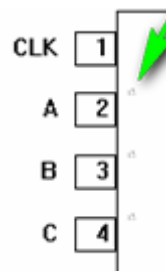
Klik på Input, og vælg de indgange, der skal med i tabellen ved at klikke på felterne i IC'en indenfor benene.

Klik menuen Output, og vælg de output-macroceller, der skal med i tabellen.

Læg mærke til nederste linie, hvor det er markeret, hvad der er med i tabellen som input og output.

(A B C -> F1 F2)

De valgte input og output er i IC'en markeret med et lille mærke, som ses svagt til højre for pin'en:



Højreklik, og svar "yes" til at implement changes

Nu skal selve sandhedstabellen skrives ind. Altså tabellen, der definerer udgangenes funktion i relation til de valgte indgange. I beskrivelsesvinduet for neden rulles ned til følgende tekst:

```
DEFINE  
  
TRUTH_TABLE Tabel1  
(A B C -> F1 F2)  
  
END;
```

Sandhedstabellen for F1 og F2 skrives ind før END. Det kunne se således ud:

```
DEFINE
```

```
TRUTH_TABLE Tabel1
```

```
(A B C -> F1 F2)
```

```
0 0 0    1 1;          ( sæt bare et antal spaces ind !!! )
```

```
0 0 1    0 1;
```

```
0 1 0    1 0;
```

```
0 1 1    1 1;
```

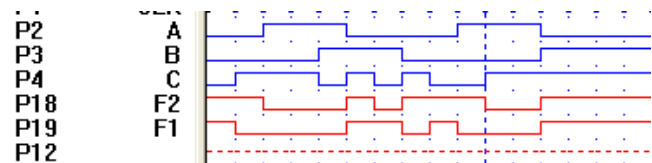
```
1 0 0    0 1;
```

```
END;
```

Bemærk: I stedet for at gennemføre ovenstående procedure, kan man blot indskrive ovenstående i kildeteksten.

Der kan defineres flere sandhedstabeller, blot de har forskellige navne !!

Gem designet, Compiler, og tjek fx output ved en simulering. Fx som følgende:



Definer INPUT og output ved blot at skrive i tekstfilen.

Ved forsøg, er det fundet, at man, - når man er fortrolig med dette, - direkte kan definere et kredsløb ved at indskrive i tekst-delen. Dette gælder både labels, definering af macrocelle osv.

Her følger nogle eksempler på kredsløb defineret med sandhedstabel:

Opbygning af tæller

Her er beskrevet opbygning af en tæller. Output har fået navnene Q3 til Q0. Afhængig af present state på output, skal tælleren skifte, når der kommer en clockpuls.

Sandhedstabellen kan fx defineres som følgende:

```
DEFINE
```

```
PRESENT = [Q3 Q2 Q1 Q0]          ; Binær værdi af nuværende stadie
```

```
NEXT = [Q3 Q2 Q1 Q0]            ; Binær værdi af next stadie.
```

```
TRUTH_TABLE Tabel2
```

```
(PRESENT -> NEXT)
```

```
0    1
```

```
1    2
```

```
2    3
```

WINPLACE

```
3  4
4  5
5  6
6  7
7  8
8  9
9  0
```

END;

3 til 8 Dekoder.

DEFINE

TRUTH_TABLE DECODE "3 til 8 dekoder (Der må skrives kommentarer efter et anførelstegn.)
(C B A -> Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7)

```
0 0 0 -> 1 0 0 0 0 0 0 0
0 0 1 -> 0 1 0 0 0 0 0 0
0 1 0 -> 0 0 1 0 0 0 0 0
0 1 1 -> 0 0 0 1 0 0 0 0
```

o. s. v.

END;

Compilering og overførsel til IC'en.

Når alle ligninger er indskrevet, og designet er gemt, kan der kompiles. Dvs. oversættes til kode, der kan programmeres ind i kredsen.

Vælg **Operation / Compile**

Herved starter et andet program op.

Compile programmet består af 3 vinduer. Øverst til venstre et compile vindue, øverst til højre et error-vindue, og nederst tekst editor vinduet.

I Compile-vinduet vælges **Compile / RUN**

Er der fejl, vises dette i kildeteksten. Ret fejl direkte i tekst-editoren, eller gå retur til Design Mode.

Der vises en sort rude hvor evt. oplysninger om reducere af ligninger gives. Dette er måske et levn fra DOS-tiden. Vinduet lukkes, og compilationen fuldføres.

For at afslutte dette reduktionsvindue automatisk i windows XP, kan man gå ind i Winplace systemmappen. Højre klik på genvejen til dosprogrammet PLREDUCE. Vælg Egenskaber / Program og sæt hak i "Luk ved afslutning", og tryk "anvend".

Efter succesfuld compilation, skabes der en række filer. Den fil, der skal føres over i PEEL 18CV8 – IC'en har efternavnet .JED. Filen indlæses i brænderprogrammet, og brændes over i PEEL-kredsen. Herefter kan kredsen testes.

Men der er også mulighed for først at simulere designet vha WinPLACE's simuleringsprogram.

Simulering:

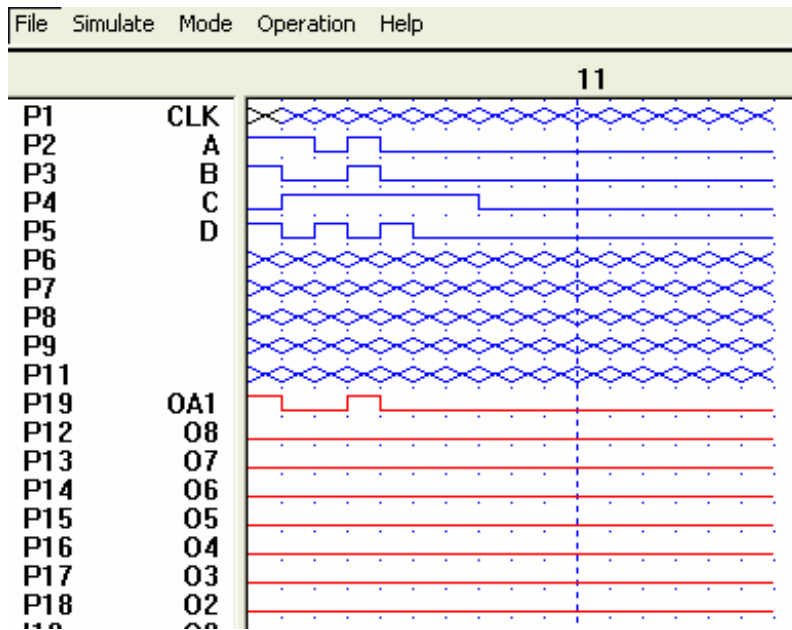
Vælg **Operation / Simulate.**

Efter Compilering kan det netop Compilede design simuleres på baggrund af den '.JED'-fil, der er beregnet til at brænde over i IC'en.

I simuleringsprogrammet kan der sættes signaler, 0'ere eller 1'ere fx på indgangene, og så kan man observere udgangene.

Fx følgende billede:

WINPLACE



Her vises et eksempel på en simulering.

I venstre side ses alle input og output med deres labels.

Signalerne på input A, B, C og D skal først indtastes. Derefter kan signalerne på udgangene observeres

I venstre vindue ses Pinnumre og navne, I højre ses signalerne

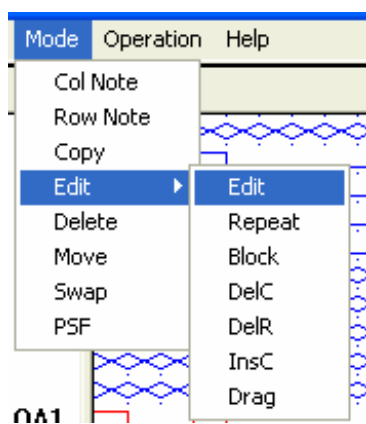
Udgang OA1 er lig $A \cdot B$.

Indtastning af input-signaler.

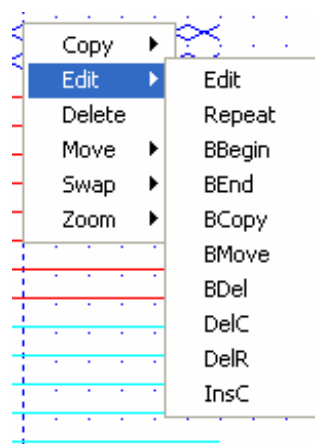
Mode / Edit / Drag

Med **DRAG** kan man hurtigt med musen indtegne nogle input-signaler. Klik i SIMULATE-vinduet i venstre side ud for et input-signal, og træk signalet mod højre. Ved at køre musen lidt opad eller nedad kan vælges 'Høj' eller 'Lav'. Slip musen og fortsæt.

Vælg MODE / Edit / EDIT, eller H.klik I højre vindue



Mode / Edit / Edit












H-klik / Edit / Edit

WINPLACE

Placer den lille ramme, der fremkommer helt til venstre I SIMULATE-Window, ud for den indgang, Fx A, der skal laves input-signal for.

Med gentagne klik med venstre mus vælges det signal, der skal sættes på. Der er 8 forskellige:

Symbol	Color	Function	JEDEC Vector Symbol
	Blue	System Clock	C
	Blue	Input High	1
	Blue	Input Low	0
	Blue	High Voltage Preload	P
	Blue	Input or Output Don't Care	X
	Red	Output High	H
	Red	Output Low	L
	Red	Output High Impedance	Z
	Green	Buried or Internal Signals which cannot be modified	N/A

0 er en blå, lav linie, 1 er en høj lav linie.

Flyt cursoren 1 tak til højre og gentag. Lav signaler for alle indgange ! startende fra Venstre.

Mode / Repeat

Klik på et sted på en signallinie. Indtast antal ens, der skal indsættes herfra.

Mode / Edit / Block

Indram en linie, sekvens eller flere linier. Klick hvor der skal indsættes en kopi af blokken !!

Vis simulerede output

Vælg Simulate / Capture for at få vist output-signalet.

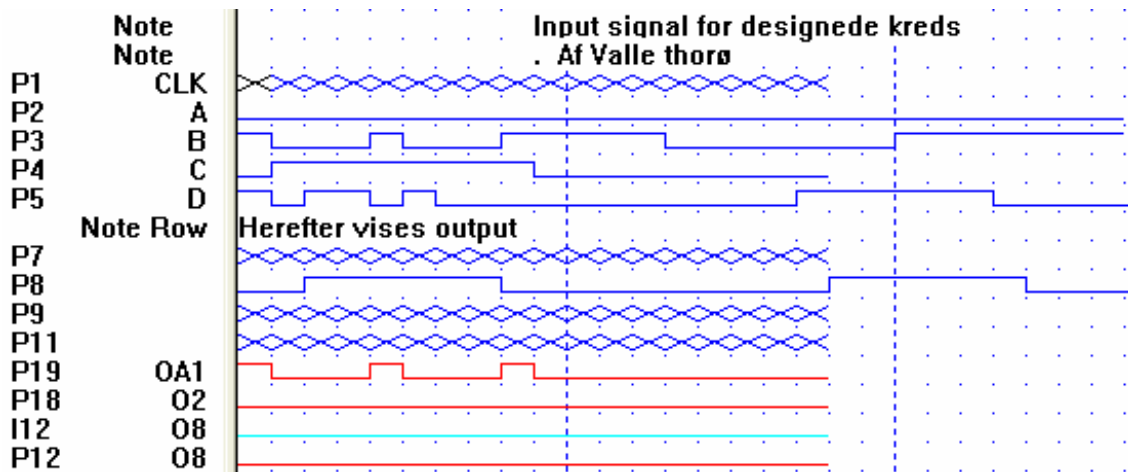
H-klick på linie, Delete sletter pågældende linie

H-klick på linie + Move + From, Derefter H-klick + Move + To for at flytte linie.

Mode / COLm Note Indsætter øverst en Notelinie. Dobbelt klick i linien og indtast fx en overskrift

Mode / Row Note Indsætter en linie hvor der klikkes. Dobbelt klik i linien, og indtast kommentar.

WINPLACE



Symbol	Color	Function	JEDEC Vector Symbol
	Blue	System Clock	C
	Blue	Input High	1
	Blue	Input Low	0
	Blue	High Voltage Preload	P
	Blue	Input or Output Don't Care	X
	Red	Output High	H
	Red	Output Low	L
	Red	Output High Impedance	Z
	Green	Buried or Internal Signals which cannot be modified	N/A

Outputsignalerne er røde, fx Lav, Høj eller Z

Winplace's tekstfil skabelon ser således ud !!

Her er vist en skabelon over den tekstfil, eller kildefil, der kreeres under defineringen af kredsens funktion. De første headere er kun beregnet til at man kan gendanne sig et design, hvis man senere skal fx ændre noget.

```
TITLE ''                               Her kan titlen på designet indskrives.
DESIGNER ''                             Hvem er forfatteren
DATE ''                                 Dato for designet
Description                             Her placers en beskrivelse af designet.
  Enter description here...
End_Desc;
```

WINPLACE

PEEL18CV8	"Device type
"Optional Special Features Identifiers AUTO_SECURE	"Programs Security-bit. If unspecified, "defaults to security-bit OFF
CLK pin 1	"Input or Clock pin declaration "I/O or Macro Cell Configuration
"I/O CONFIGURATION DECLARATION "IOC (PIN_NO 'PIN_NAME' POLARITY OUTPUT_TYPE FEEDBACK_TYPE)	
A Pin 2	
B Pin 3	
D Pin 4	
IOC (12 "Pos Com Feed_Pin)	
IOC (13 " Pos Com Feed_Pin)	
IOC (14 " Pos Com Feed_Pin)	
IOC (15 " Pos Com Feed_Pin)	
IOC (16 "Pos Com Feed_Pin)	
IOC (17 "Pos Com Feed_Pin)	
IOC (18 "Pos Com Feed_Pin)	
IOC (19 'F1' Pos OutCom Feed_Pin)	
AR NODE 21	"Global Asynchronous Reset Node
SP NODE 22	"Global Synchronous Preset Node
DEFINE	
EQUATIONS	"Logic equation syntax
"Equations for the global nodes	
AR = 0;	Hvis brugt, skal AR og SP defineres. I
SP = 0;	stedet for 0 skrives den ligning, der skal
	angive signalet. Er det blot en input pin,
	skrives dens label.
F1.And = 0;	
F1.SumB = A*B;	F1 er den udgang, der her er indskrevet
	ligning for. Udgangen er lig $A \cdot B$
<u>FILNAVNE:</u>	

WINPLACE

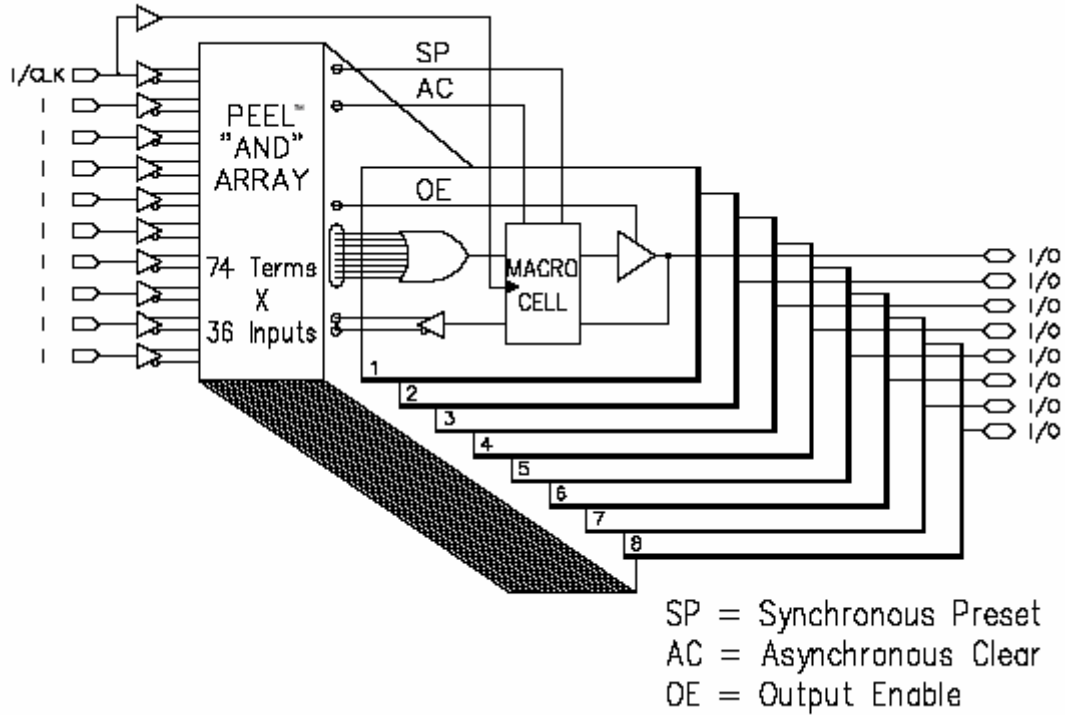
Udviklingsværktøjet genererer forskellige filtyper, alle med det fornavn, man har givet filen. Oversigten viser hvad filerne er beregnet til. Bemærk, at .JED-filen er den eneste, der skal over til brænderen.

Alle filer har samme brugerdefinerede fornavn, mens efternavnene angiver filtypen:

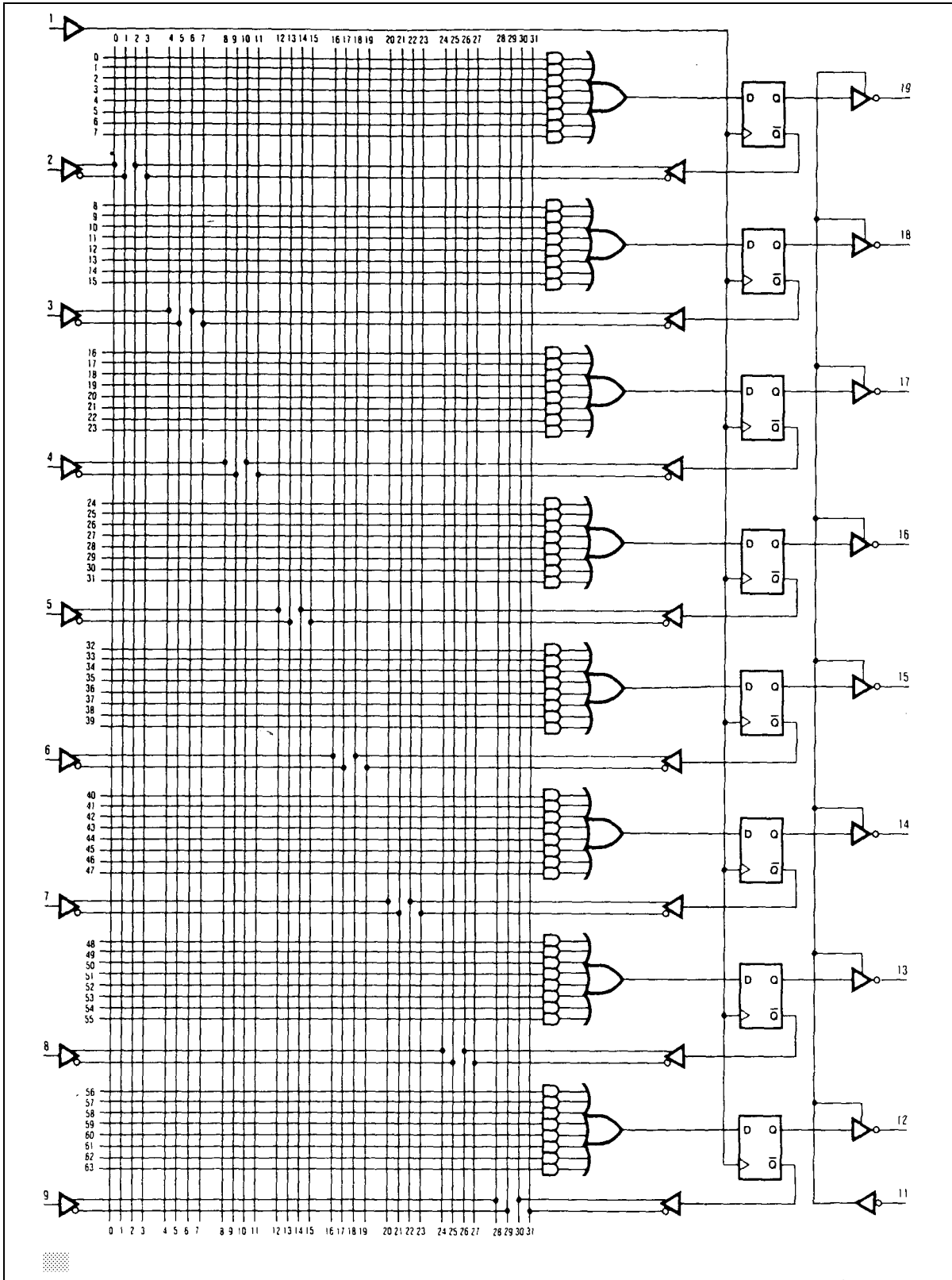
- PSF: Place Source File. Beskrivelsen af dit design. Afløseren for den gamle APL-fil. Du kan køre programmet APL2PSF for at få dine gamle filer oversat til det nye format.
- PS: BAK-fil for PSF
- MAP: Output fra kompilatoren. Ret ligegyldig.
- RED: Output fra optimerings-programmet. Såfremt dine logiske udtryk er blevet optimeret, dannes denne fil. I udskriften kan ses, hvordan Winplace har reduceret ligninger eller sandhedsskema.
- JED: JEDEC-fil til brænding.
- JE: BAK-fil for JED.
- CFG: Output-fil fra simulatoren
- PRT: Opsætningsfil til dokumentations-afsnittet.
- PR: BAK-fil for PRT

WINPLACE

Her gives en række yderligere oplysninger om opbygningen af en PLD-kreds.



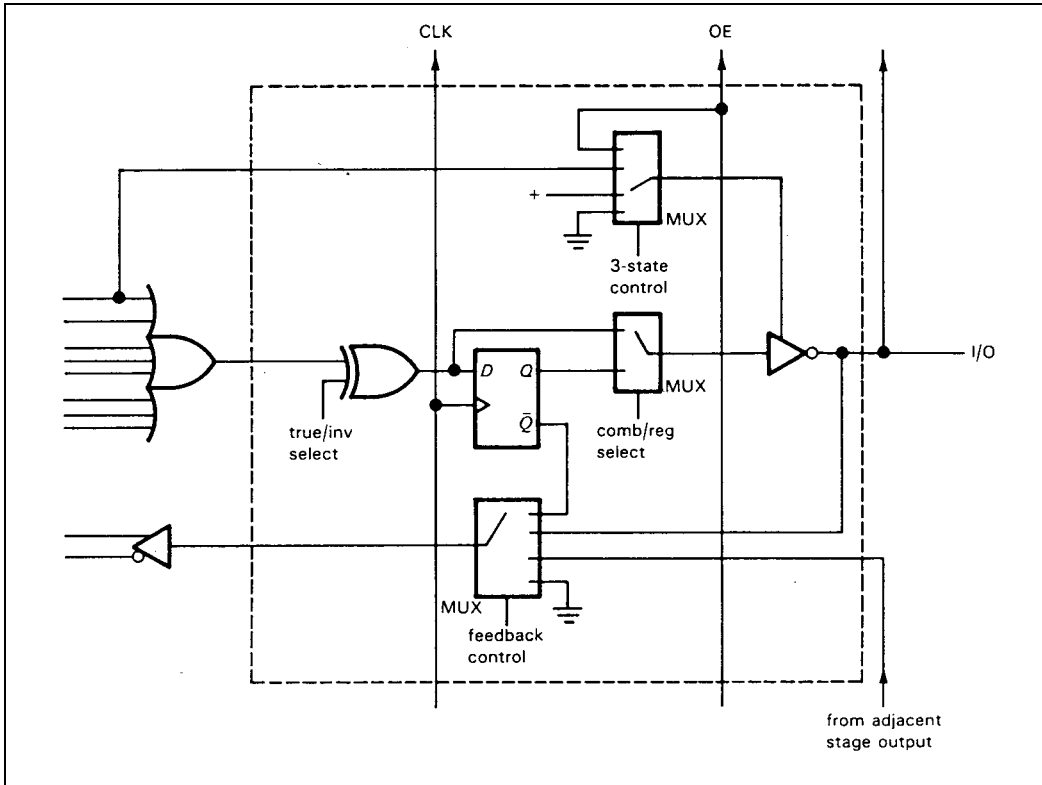
WINPLACE



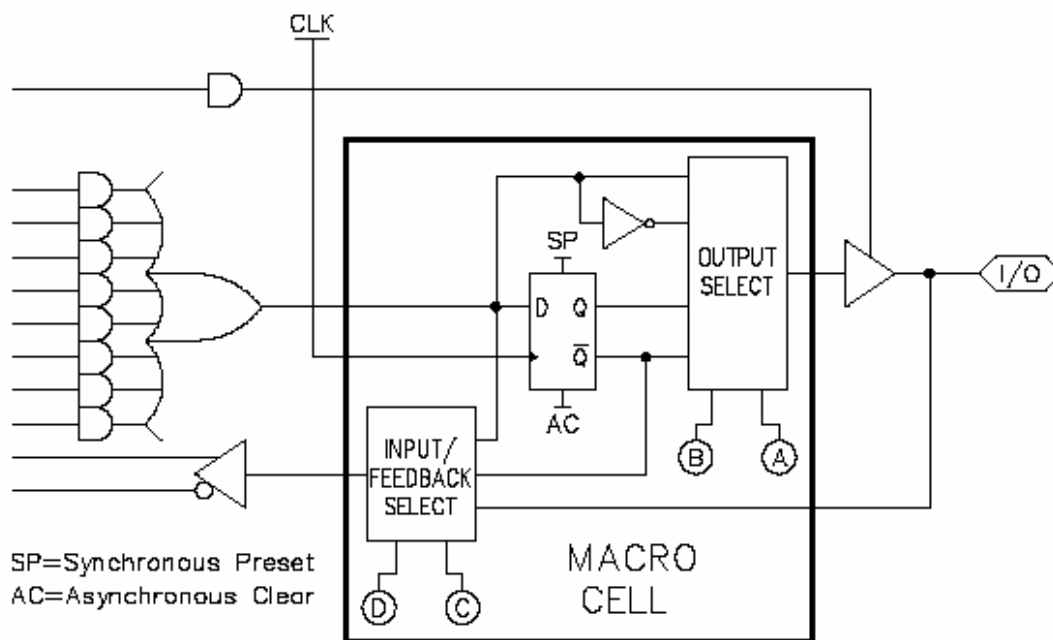
Indgangene er til venstre, og med et programmeringsudstyr kan der skabes forbindelser i matrixen så den rette funktion opnås.

WINPLACE

Forstørres MACRO-cellen, se herunder, ses, at den indeholder en D-FF, - og flere MUX-ere til at vælge, hvorfra og hvortil signaler forbindes.



En Peel 18cv8 – macrocelle eller vist på en anden måde herunder

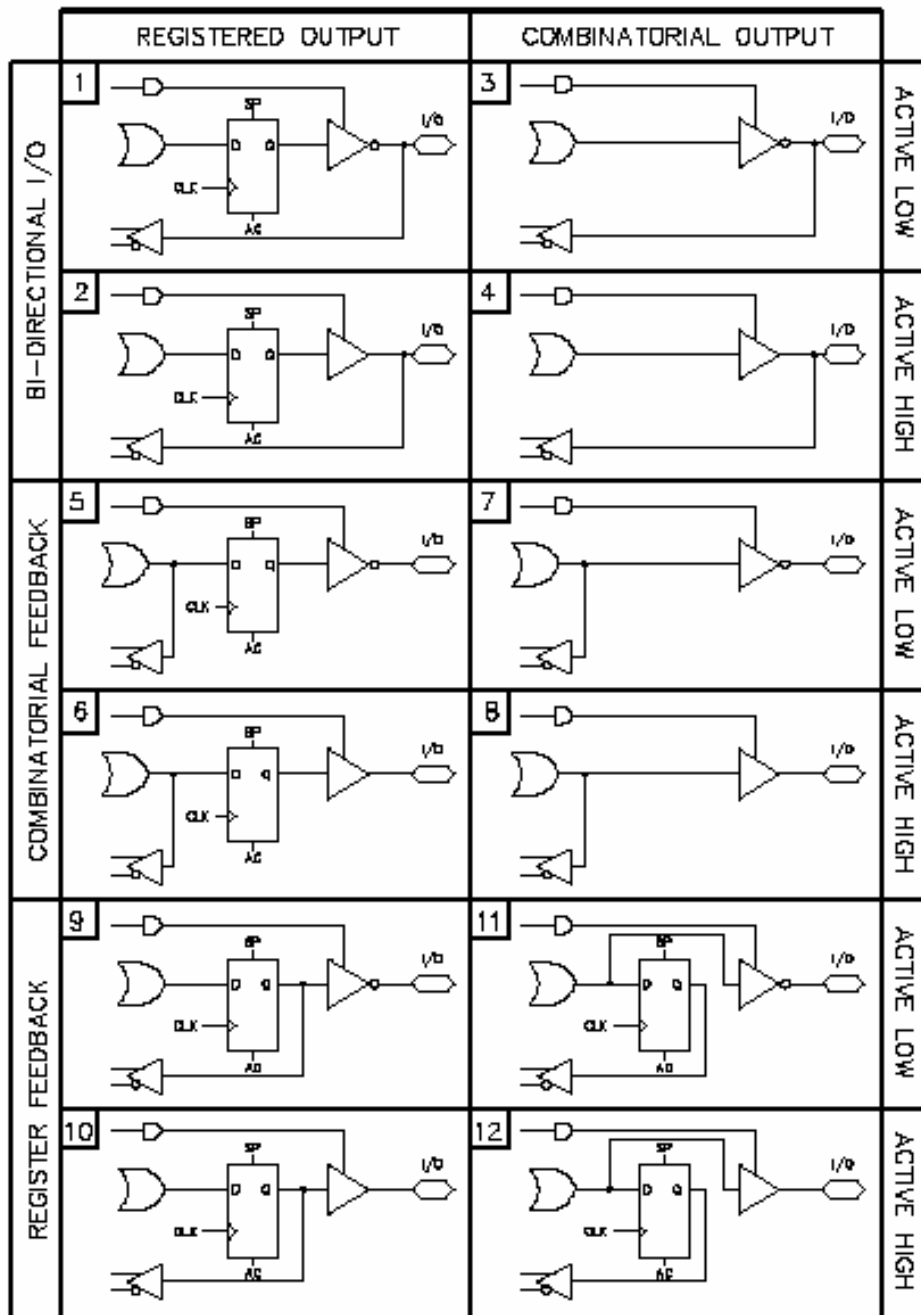


WINPLACE

Configuration					Input/Feedback Select	Output Select	
#	A	B	C	D		Register	Active Low
1	1	1	1	1	Bi-Directional I/O	Register	Active Low
2	0	1	1	1			Active High
3	1	0	1	1		Combinational	Active Low
4	0	0	1	1			Active High
5	1	1	1	0	Combinational Feedback	Register	Active Low
6	0	1	1	0			Active High
7	1	0	1	0		Combinational	Active Low
8	0	0	1	0			Active High
9	1	1	0	0	Register Feedback	Register	Active Low
10	0	1	0	0			Active High
11	1	0	0	0		Combinational	Active Low
12	0	0	0	0			Active High

På følgende figur er de forskellige valgbare muligheder i macro-cellerne vist.

WINPLACE



Forskellige muligheder for konfiguration af en makrocellen i PEEL 18CV8

WINPLACE

Opgaver: Vælg fra nedenstående opgaveliste. Test resultaterne !!

1	<p>Løs flg. opgave med 1 PLD: (Skråstreg betyder "Inverteret" fx. /A er " A Ikke ")</p> $F1 = /A*B*/C*D + A*B*C*/D + /A*/B*D + /D$ $F2 = A*B*C*D + A*/B*/C*D + /A$ <p>Test på fumlebrædt, med lysdioder (med formodstande) på udgangene.</p>																				
2	Lav en PLD, der omsætter en hexadecimal kode til 7-segment. Afprøv på fumlebrædt.																				
3	Lav en PLD med to indgange (A & B), og flere udgange, hhv. med en AND-gate, en NAND-gate, en OR, en NOR og en EXNOR-gate.																				
4	Lav en 3-bit graykodetæller. Afprøv !																				
5	Lav en PLD med en 4-bit parallel ind / parallel out register, en "PIPO" eller "LATCH".																				
6	Konstruer en latched BCD/7segment dekoder.																				
7	Byg en 3-bit binær tæller, der er selvstandsende på "111". Ben 4 er "reset" og ben 3 er "Count enable".																				
8	Byg en "ternings-kode-tæller". Der skal clocksignal på ben 1. Fx Ben 3 er "Count enable". Der skal bruges 7 lysdioder på udgangene.																				
9	Byg en kreds, med 4 indgange, og 4 udgange. Den skal sættes imellem en 4017, og transistorerne, der styrer stepmotoren. Vha. en ekstra ben, kan omdrejningsretningen for motoren bestemmes.																				
10	<p>Lav en 4-bit "Stepmotor-tæller" Der skal være UP/DOWN- og Half / Full Step. Koderne vises i nedenstående skema.</p> <table border="1"><thead><tr><th>Half Step</th><th>Full Step</th></tr></thead><tbody><tr><td>D C B A</td><td>D C B A</td></tr><tr><td>0 0 0 1</td><td>0 1 0 1</td></tr><tr><td>0 0 1 1</td><td>0 1 1 0</td></tr><tr><td>0 0 1 0</td><td>1 0 1 0</td></tr><tr><td>0 1 1 0</td><td>1 0 0 1</td></tr><tr><td>0 1 0 0</td><td></td></tr><tr><td>1 1 0 0</td><td></td></tr><tr><td>1 0 0 0</td><td></td></tr><tr><td>1 0 0 1</td><td></td></tr></tbody></table>	Half Step	Full Step	D C B A	D C B A	0 0 0 1	0 1 0 1	0 0 1 1	0 1 1 0	0 0 1 0	1 0 1 0	0 1 1 0	1 0 0 1	0 1 0 0		1 1 0 0		1 0 0 0		1 0 0 1	
Half Step	Full Step																				
D C B A	D C B A																				
0 0 0 1	0 1 0 1																				
0 0 1 1	0 1 1 0																				
0 0 1 0	1 0 1 0																				
0 1 1 0	1 0 0 1																				
0 1 0 0																					
1 1 0 0																					
1 0 0 0																					
1 0 0 1																					
11																					

WINPLACE

12	
----	--