Programmerbar logikk

En enkel innføring

Agenda

- Etablere en sammenheng for å lære VHDL
- Kort omkonstruksjon med 74-serien
- Vise hensikten konstruksjoner i programmerbar logikk
- Gå igjennomenkle PLD arkitekturer

Litt om konstruksjon med 74-serien (TTL-logikk)

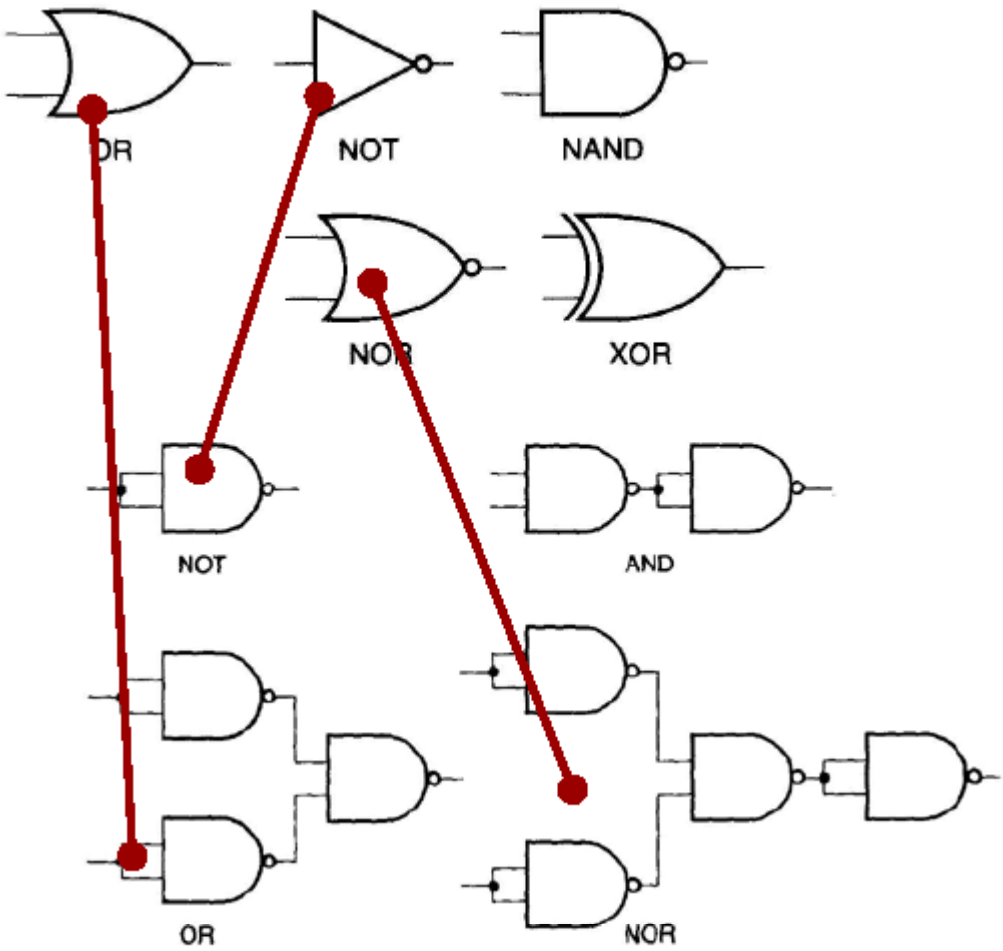
- Texas Instrument
  - Tidligst ute med logikkseriene 54/74
- Hvordan konstruere med TTL
  - Sannhetstabeller
    - Gir sum av produkter/produkt av summer
  - Boolske ligninger optimaliseres ved Karnaughdiagram
  - Vil gjerne ha designet implementert med bare NOR eller NAND
    - Benytter DeMorgan teoremer:

$$X = AB + CD + BD + BC + AD + AC$$

$$\Downarrow$$

$$X = \overline{\overline{AB} \bullet \overline{CD} \bullet \overline{BD} \bullet \overline{BC} \bullet \overline{AD} \bullet \overline{AC}}$$

Logiske porter (Gates)

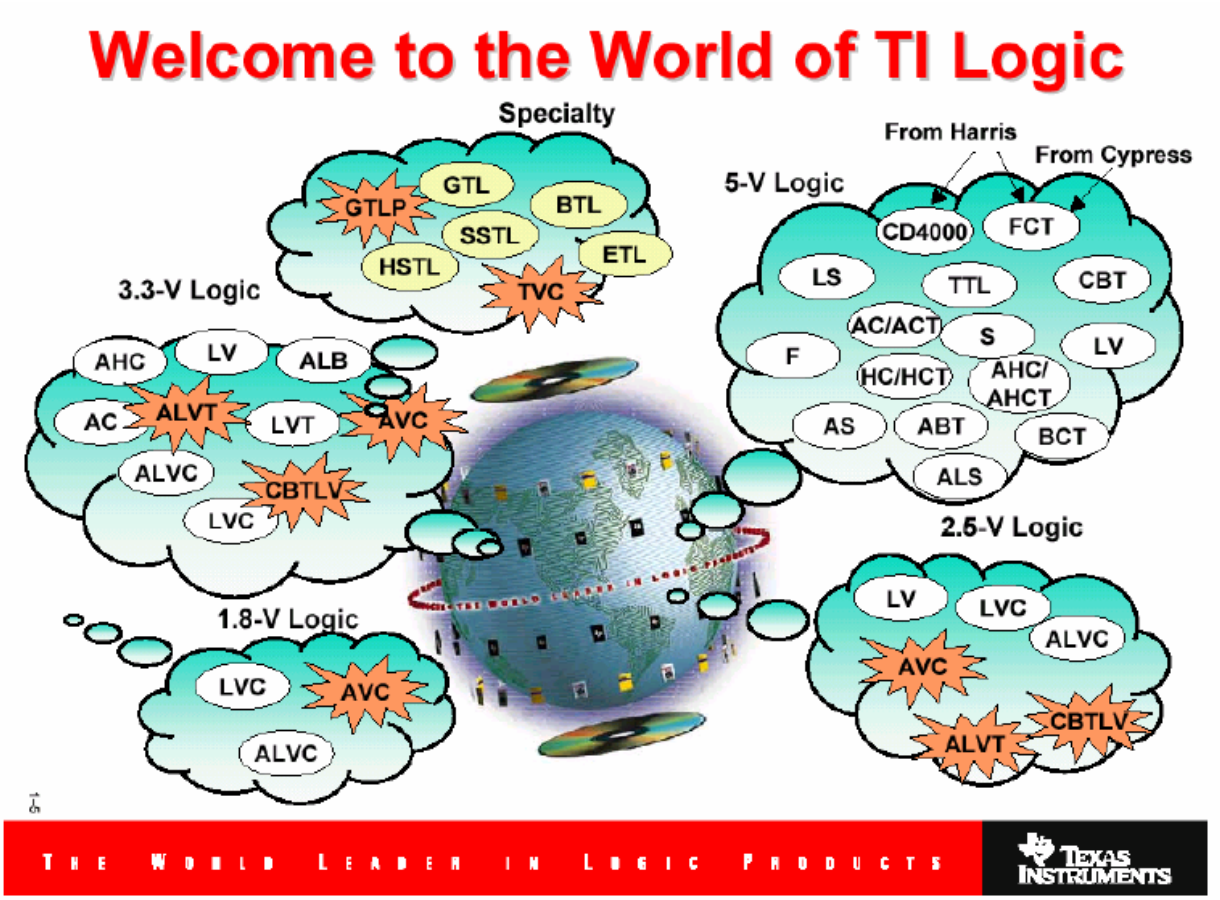


54/74-serien

54/74 Series	Description
7400	Quadruple 2-input positive-NAND gates: $Y = \overline{AB}$
7402	Quadruple 2-input positive-NOR gates: $Y = \overline{A+B}$
7404	Hex inverters: $Y = \overline{A}$
7408	Quadruple 2-input positive-AND gates: $Y = AB$
7430	8-input positive-NAND gates: $Y = \overline{ABCDEFGH}$
7432	Quadruple 2-input positive-OR gates: $Y = A+B$
7451	Dual AND-OR-INVERT gates: $Y = \overline{AB+CD}$
7474	Dual D-type positive-edge-triggered flip-flops with preset and clear
7483	4-bit binary full adder with fast carry
7486	Quadruple 2-input exclusive-OR gates: $Y = A \oplus B$
74109	Dual J-K positive-edge-triggered flip-flops with preset and clear
74157	Quadruple 2-to-1 multiplexers
74163	Synchronous 4-bit counter with synchronous clear
74180	9-bit odd/even parity generator/checker
74374	Octal D-type flip-flops

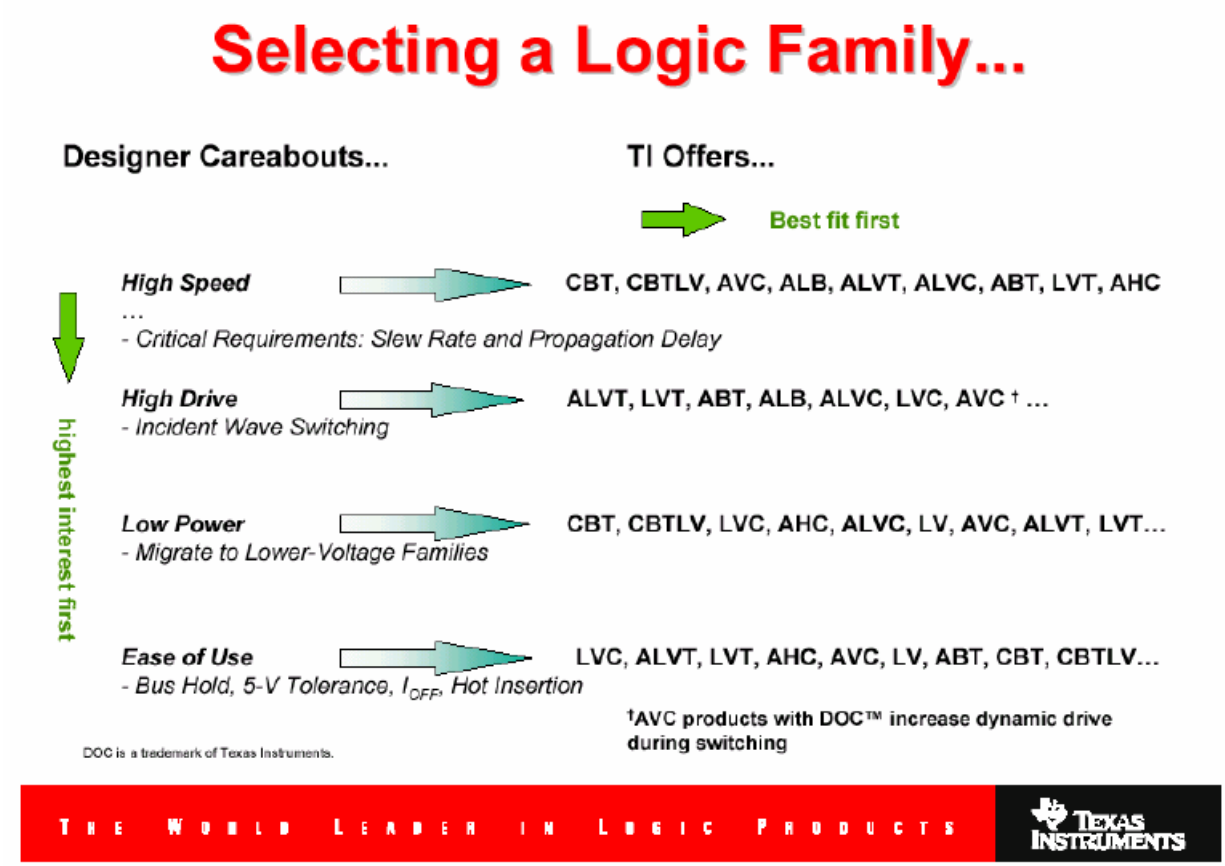
TTL-logikken er ikke død

- Kommer stadig nye serier



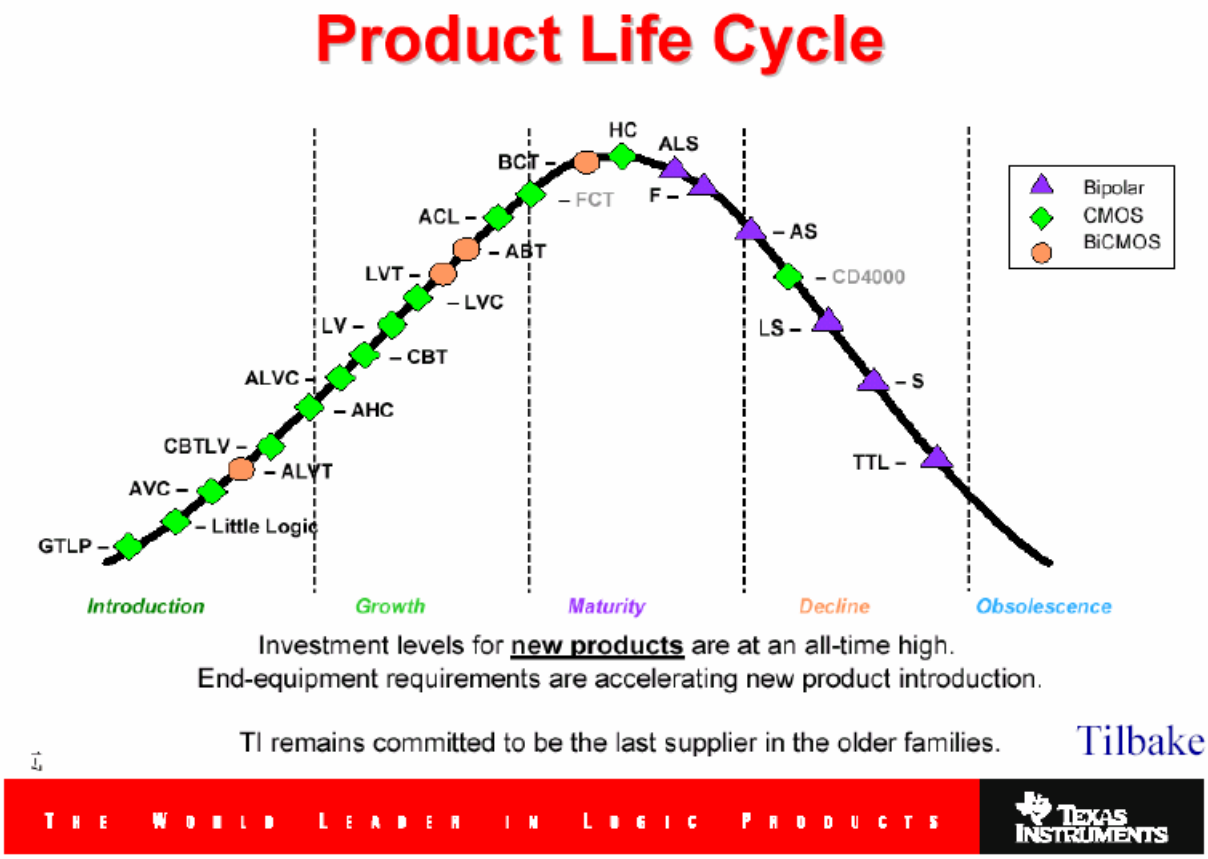
TTL-logikken er ikke død

- TTL familien finner stadig nye anvendelser og det kommer stadig nye serier



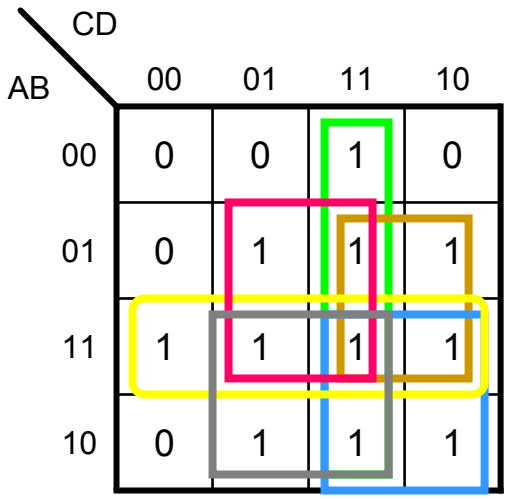
TTL-logikken er ikke død

- De opprinnelige seriene er på vei til å fases ut, mens nye kommer til.



Karnaughdiagram

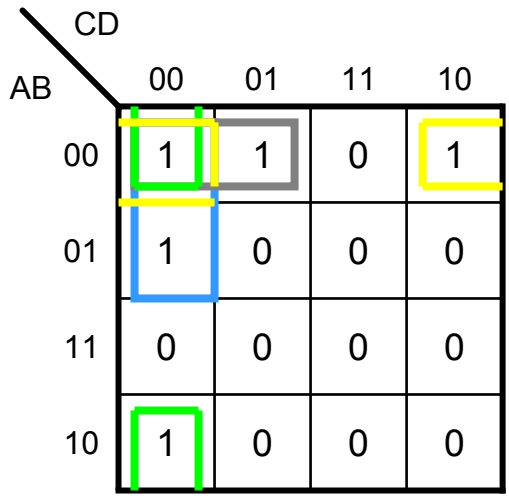
A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



$$X = AB + CD + BD + BC + AD + AC$$

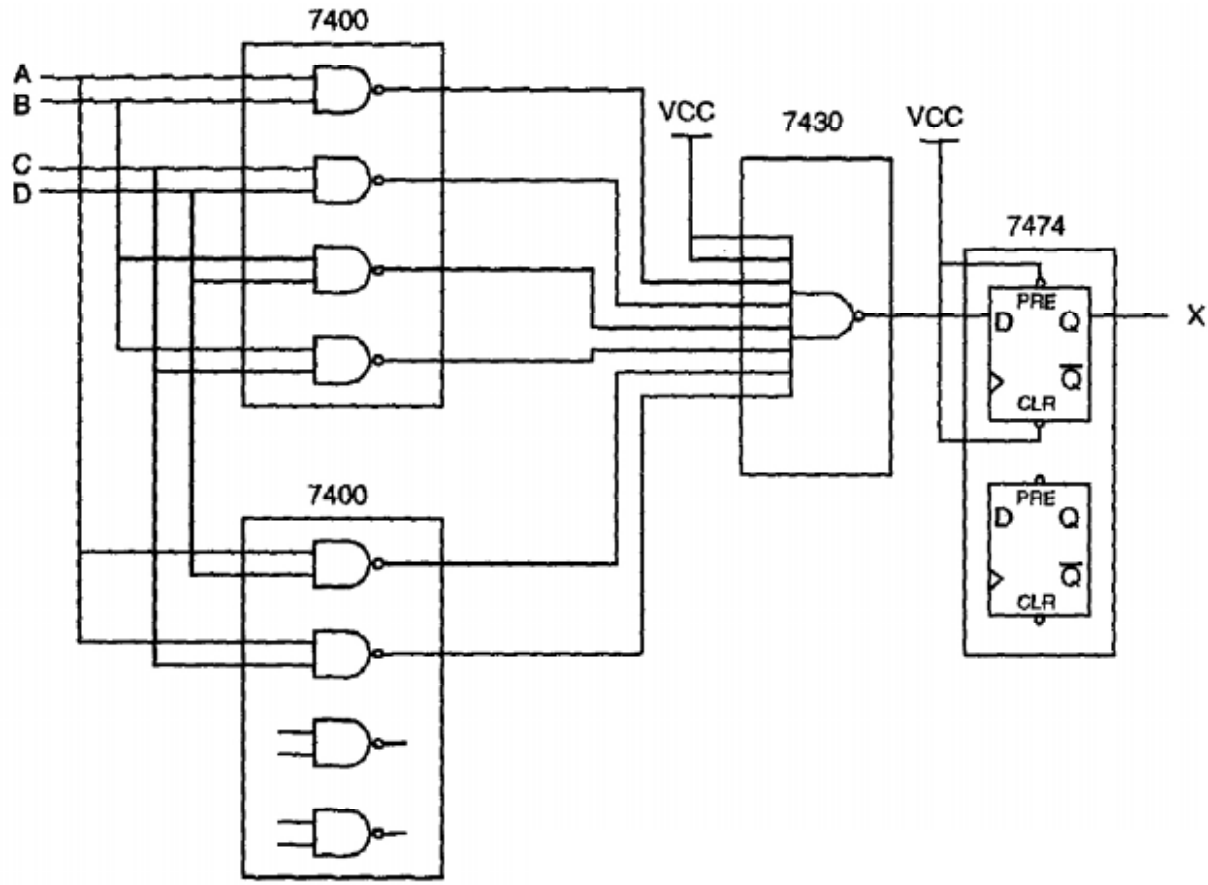
Karnaughdiagram forts.

A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



$$\overline{X} = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}\overline{D} + \overline{A}\overline{C}\overline{D} + \overline{B}\overline{C}\overline{D}$$

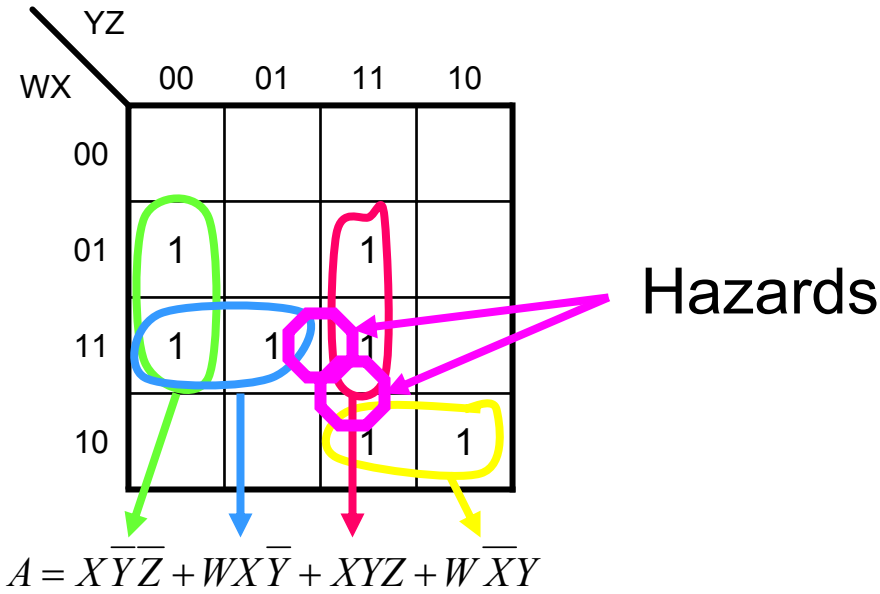
Implementasjon med TTL (nand)

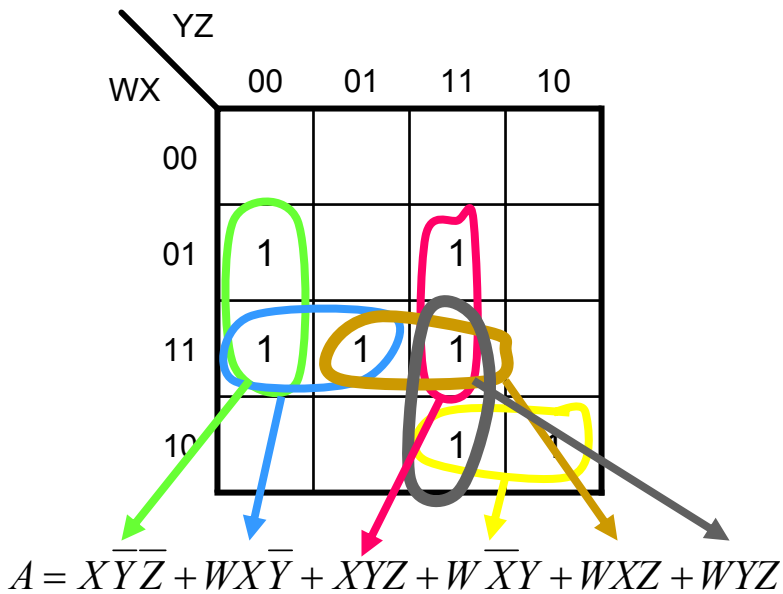


$$X = AB + CD + BD + BC + AD + AC$$

$$X = \overline{\overline{AB} \cdot \overline{CD} \cdot \overline{BD} \cdot \overline{BC} \cdot \overline{AD} \cdot \overline{AC}}$$

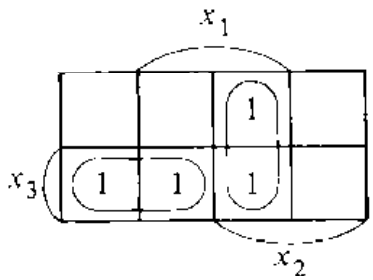
Hazards


YZ
WX
00
01
11
10
Hazards
$A = X\overline{Y}\,\overline{Z} + WX\overline{Y} + XYZ + W\overline{X}Y$

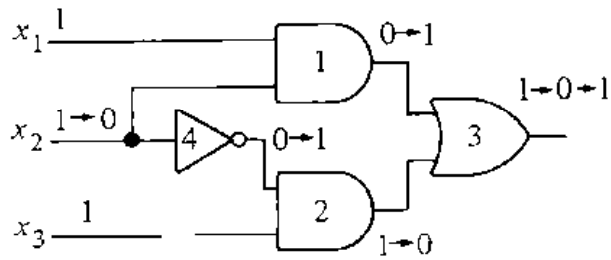

YZ
WX
00
01
11
10
$A = X\overline{Y}\,\overline{Z} + WX\overline{Y} + XYZ + W\overline{X}Y + WXZ + WYZ$

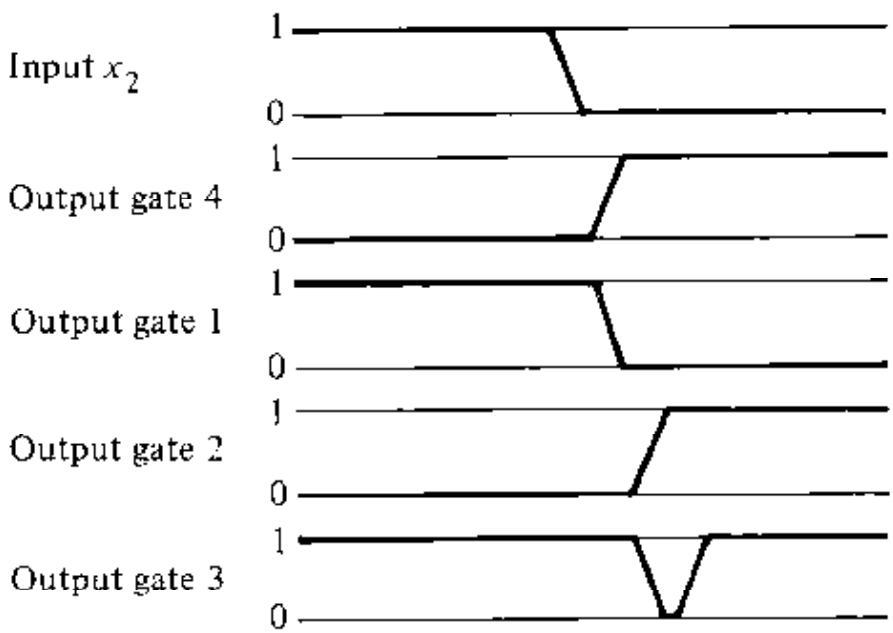
Enkel krets med Hazard

$$f = x_2x_1 + x_3\bar{x}_2$$



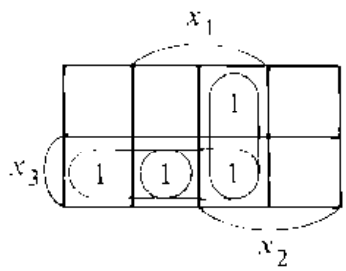
$f = x_3\overline{x_2} + x_2x_1$



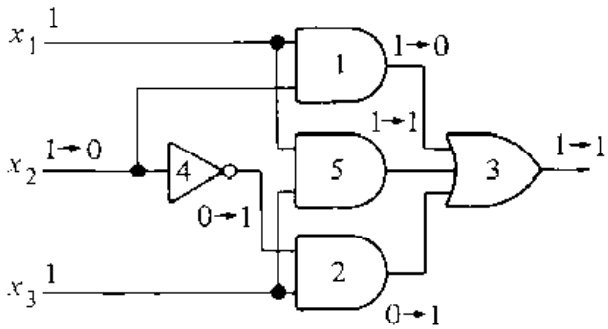


Eliminasjon av Hazard

- Setter inn ekstra gate

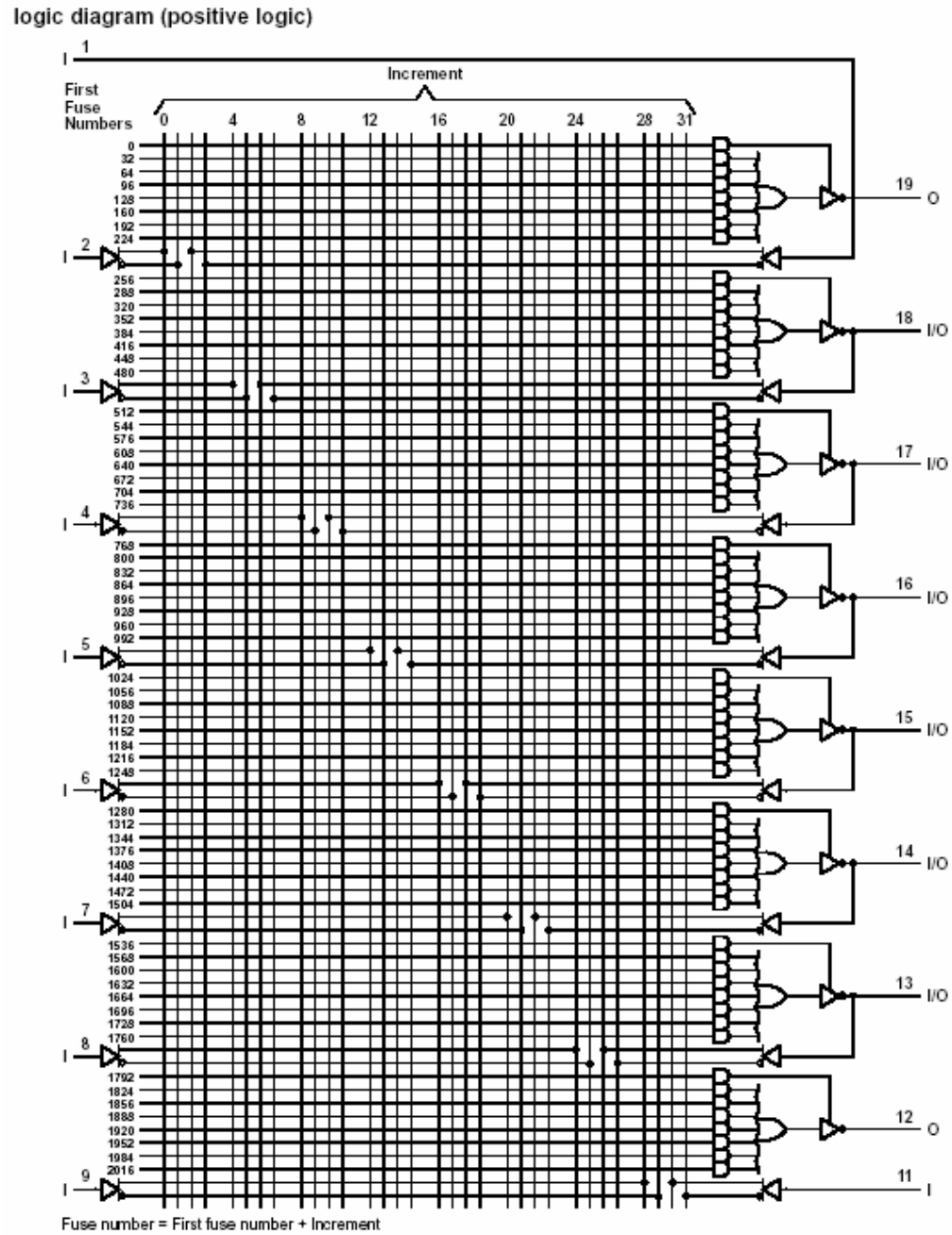


$f = x_3x_2 + x_2x_1 + x_3x_1$

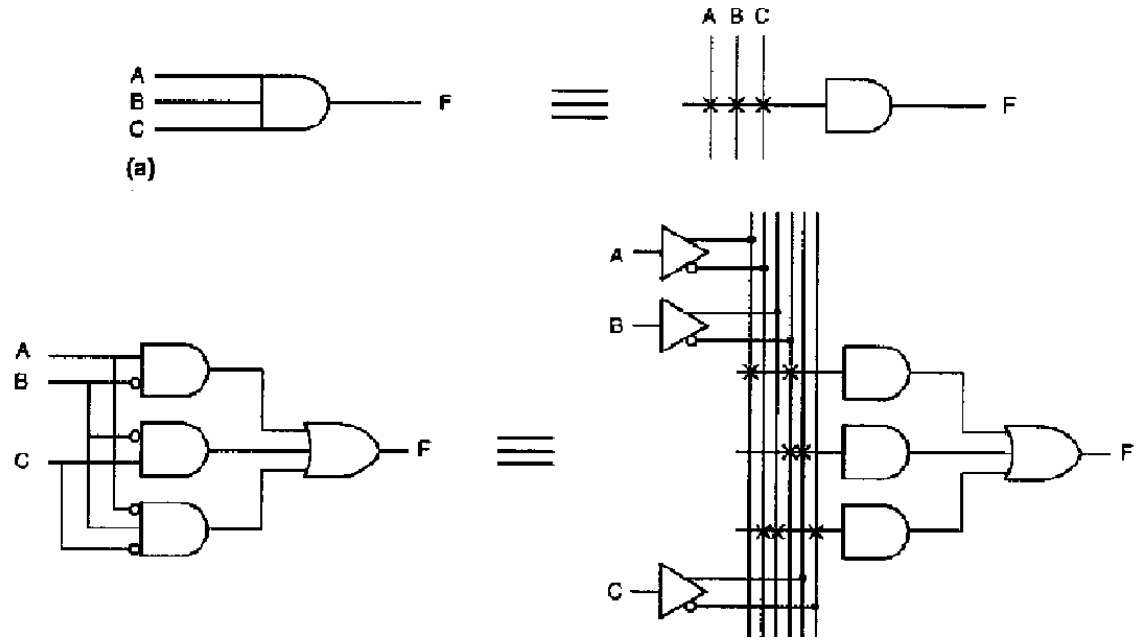


Hva er programmerbare kretser

- PAL (Programable Array Logikk)



PAL basis elementer



PAL programmering

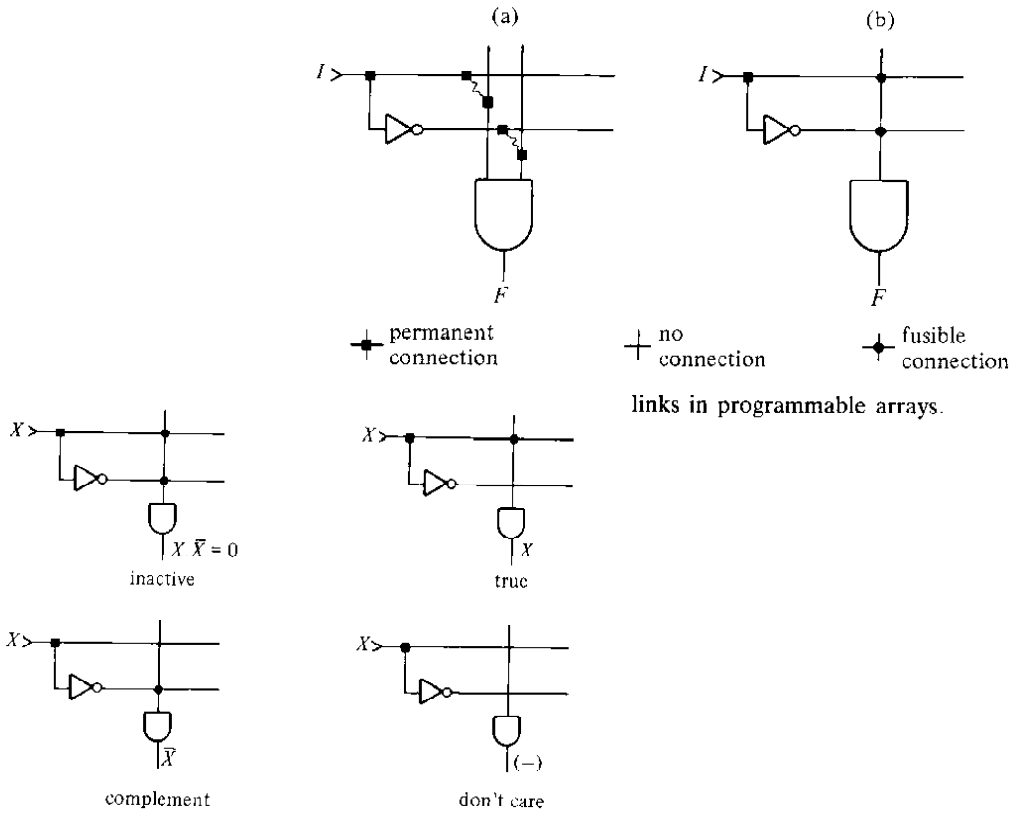
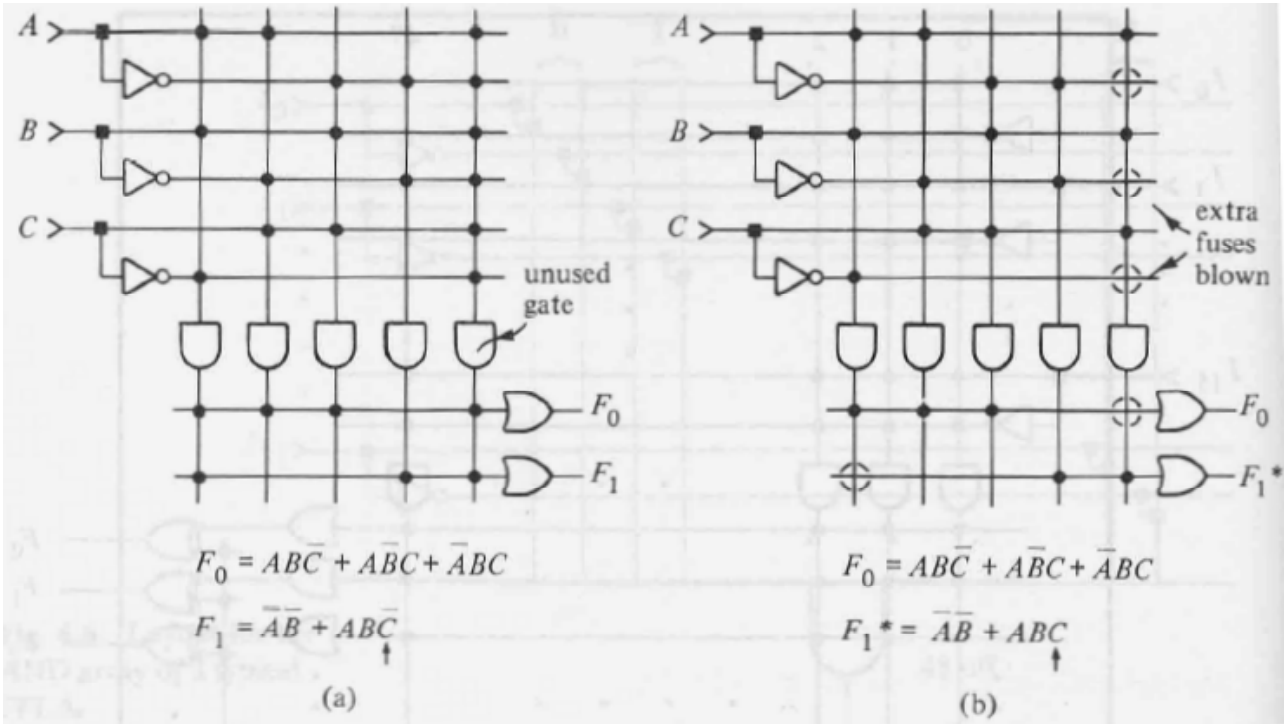


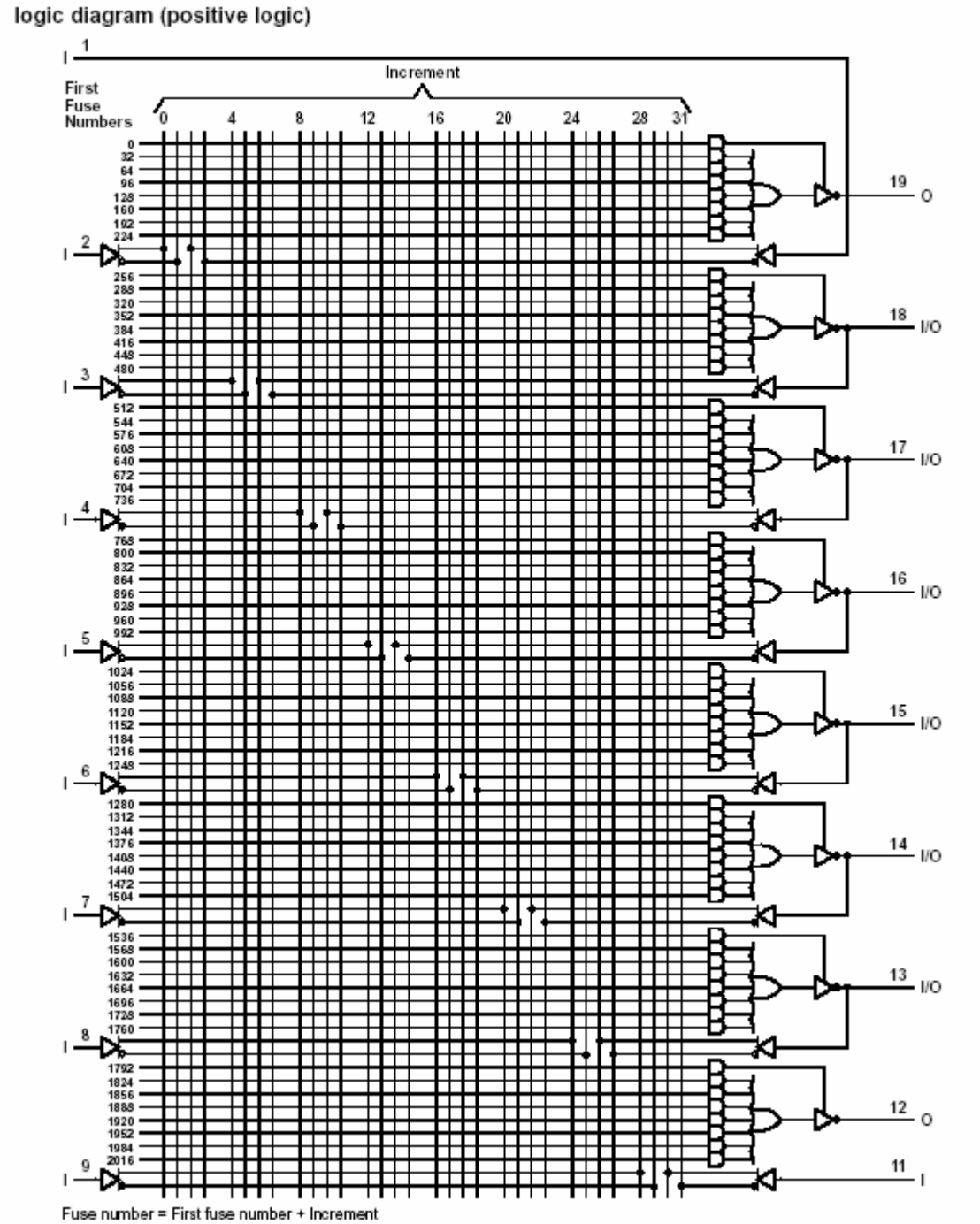
Fig. 4.2 The four possible situations after programming a single node.

PAL

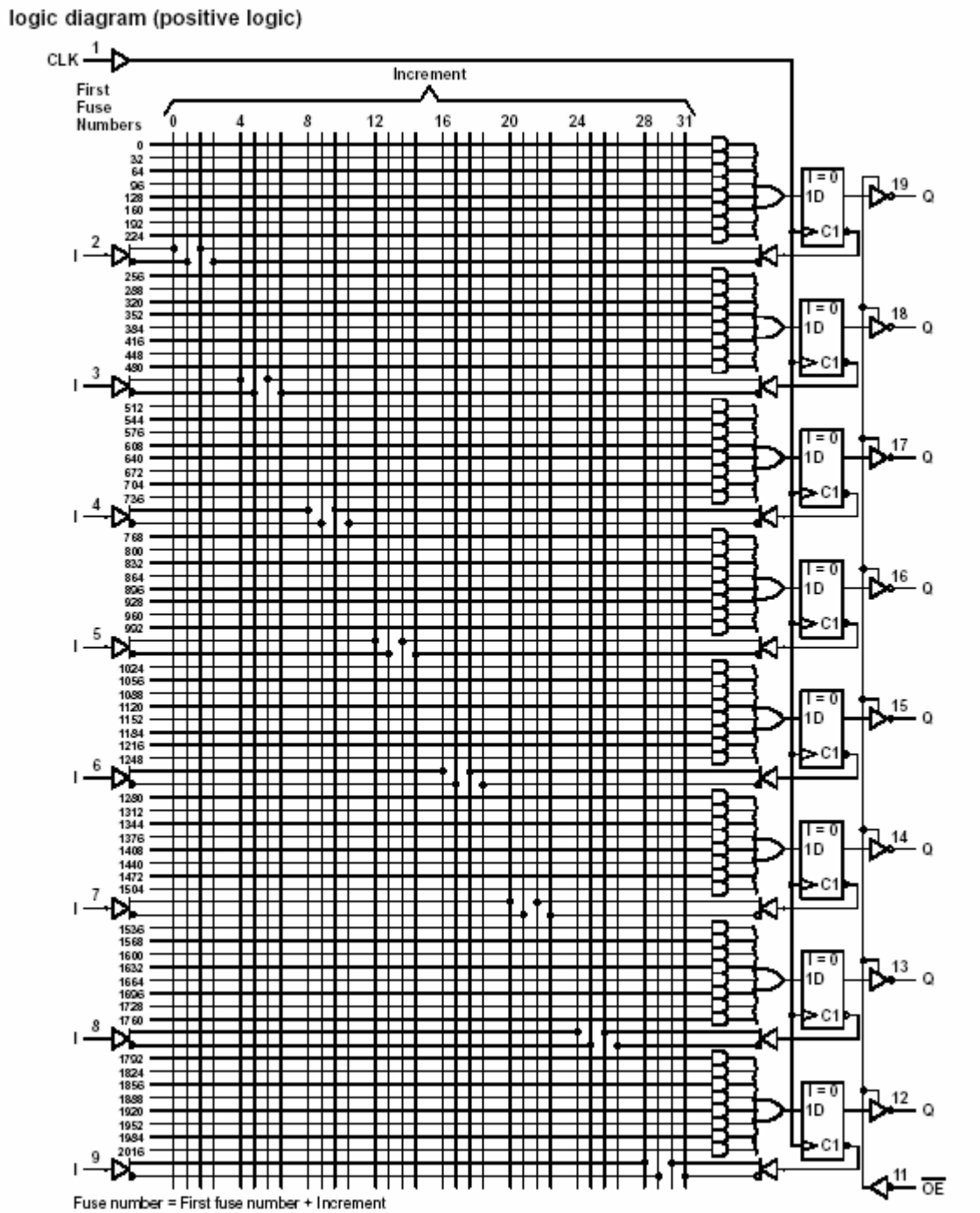
- Bruk av overskudd porter til å gjøre ekstra modifikasjoner



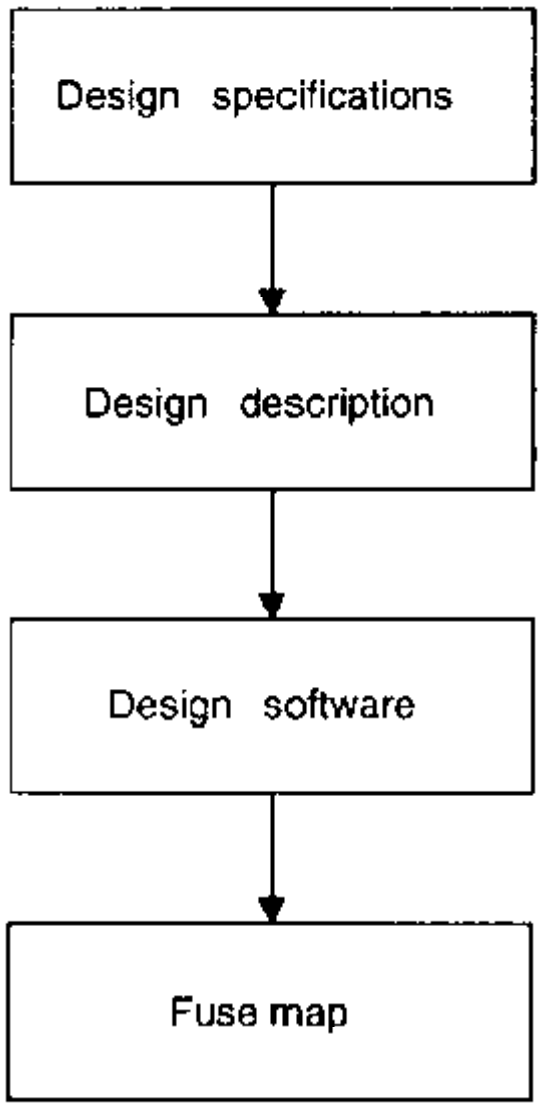
PAL 16L8


logic diagram (positive logic)
Increment
First Fuse Numbers
0 4 8 12 16 20 24 28 31
0 32 64 96 128 160 192 224
256 288 320 352 384 416 448 480
512 544 576 608 640 672 704 736
768 800 832 864 896 928 960 992
1024 1056 1088 1120 1152 1184 1216 1248
1280 1312 1344 1376 1408 1440 1472 1504
1536 1568 1600 1632 1664 1696 1728 1760
1792 1824 1856 1888 1920 1952 1984 2016
1 2 3 4 5 6 7 8 9
19 O
18 I/O
17 I/O
16 I/O
15 I/O
14 I/O
13 I/O
12 O
11 I
Fuse number = First fuse number + Increment

PAL 16R8


logic diagram (positive logic)
CLK
1
Increment
First
Fuse
Numbers
0
4
8
12
16
20
24
28
31
0
32
64
96
128
160
192
224
2
I
256
288
320
352
384
416
448
480
3
I
512
544
576
608
640
672
704
736
4
I
768
800
832
864
896
928
960
992
5
I
1024
1056
1088
1120
1152
1184
1216
1248
6
I
1280
1312
1344
1376
1408
1440
1472
1504
7
I
1536
1568
1600
1632
1664
1696
1728
1760
8
I
1792
1824
1856
1888
1920
1952
1984
2016
9
I
I = 0
1D
C1
19
Q
18
17
16
15
14
13
12
11
OE
Fuse number = First fuse number + Increment

PAL design prosess



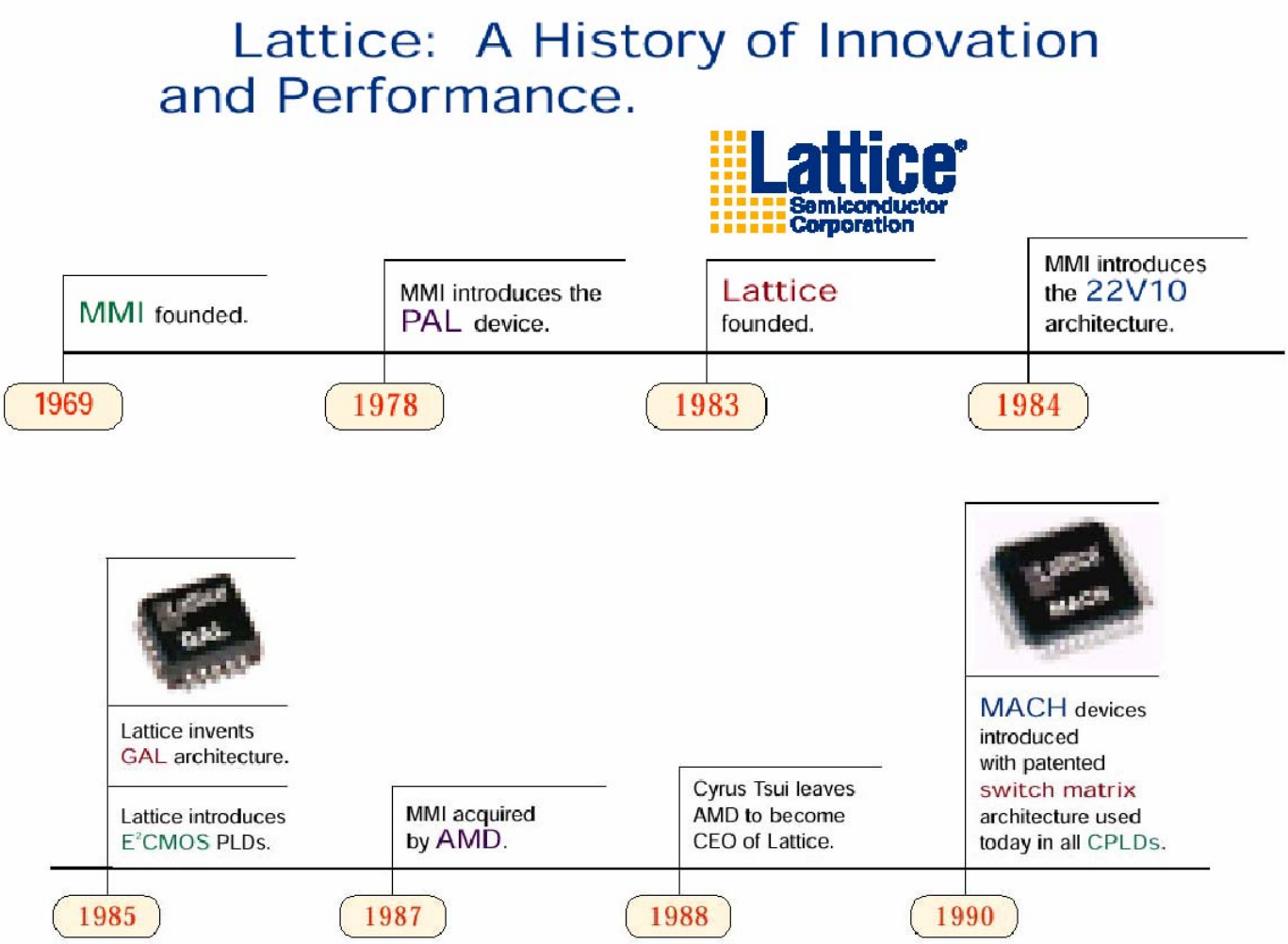
F.eks CUPL

Jedec fil

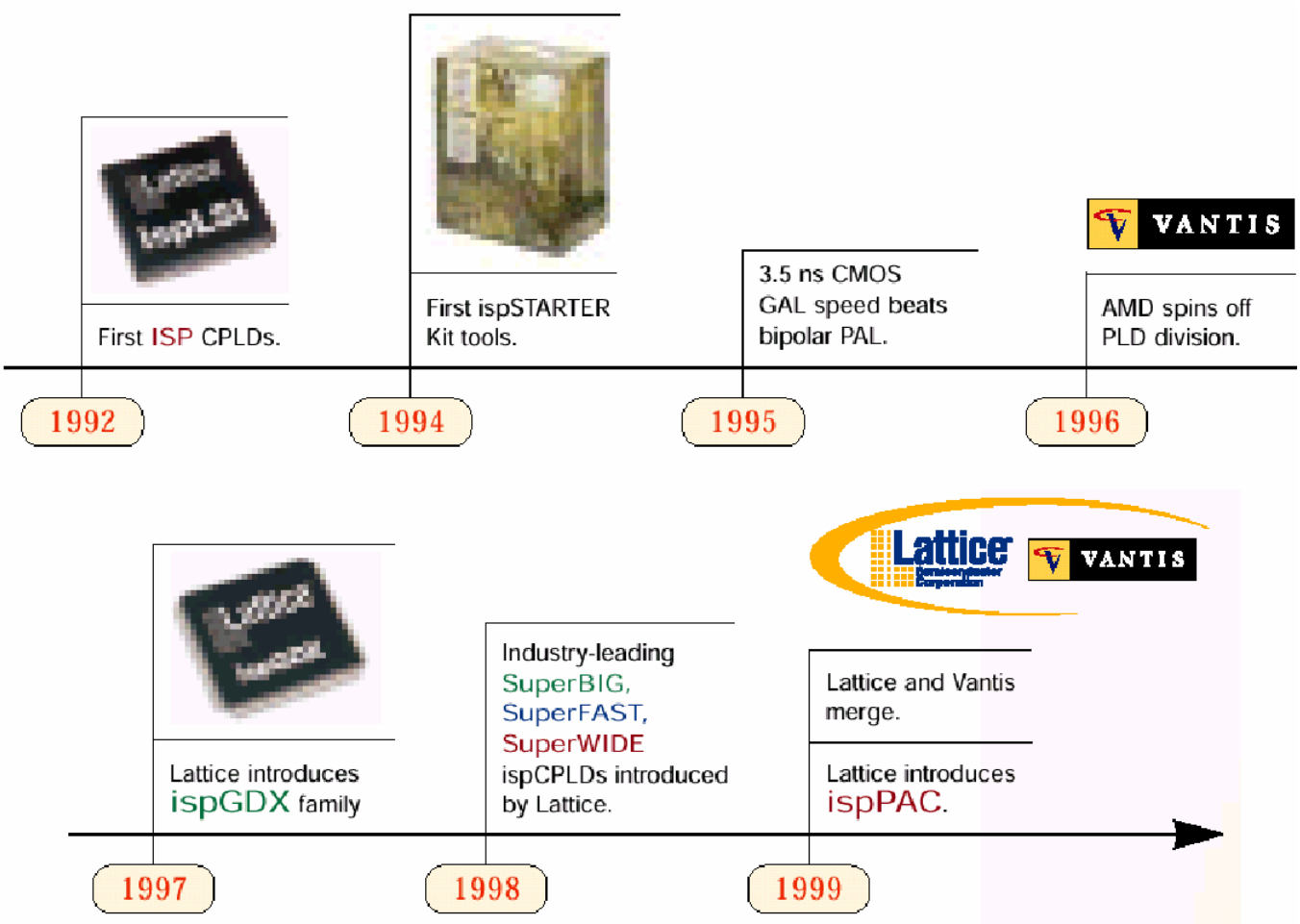
CUPL – lavnivå PLD språk

```
Name 4BIT_COUNTER;
Device V2500B;

/* inputs */
pin 1 = CLK;
pin 3 = LD_;
pin 17 = RST_;
pin [18,19,20,21] = [I0,I1,I2,I3];

/* outputs */
pin [4,5,6,7] = [Q0,Q1,Q2,Q3];

field CNT = [Q3,Q2,Q1,Q0];

/* equations */
Q3.T = (!Q2 & !Q1 & !Q0) & LD_ & RST_ # Q3 & !RST_ # (Q3 $ I3) & !LD_;
Q2.T = (!Q1 & !Q0) & LD_ & RST_ # Q2 & !RST_ # (Q2 $ I2) & !LD_;
Q1.T = !Q0 & LD_ & RST_ # Q1 & !RST_ # (Q1 $ I1) & !LD_;
Q0.T = LD_ & RST_ # Q0 & !RST_ # (Q0 $ I0) & !LD_;

CNT.CK = CLK;
CNT.OE = 'h'F;
CNT.AR = 'h'0;
CNT.SP = 'h'0;
```

Lattice


Lattice: A History of Innovation and Performance.
Lattice Semiconductor Corporation
MMI founded.
1969
MMI introduces the PAL device.
1978
Lattice founded.
1983
MMI introduces the 22V10 architecture.
1984
Lattice invents GAL architecture.
Lattice introduces E²CMOS PLDs.
1985
MMI acquired by AMD.
1987
Cyrus Tsui leaves AMD to become CEO of Lattice.
1988
MACH devices introduced with patented switch matrix architecture used today in all CPLDs.
1990

Lattice forts.



Programmeringsteknlogier

- EPROM
  - Electric Programable ROM
- EEPROM
  - Electric Erasable Programable ROM
- FLASH

”Wired” AND med EPROM celler

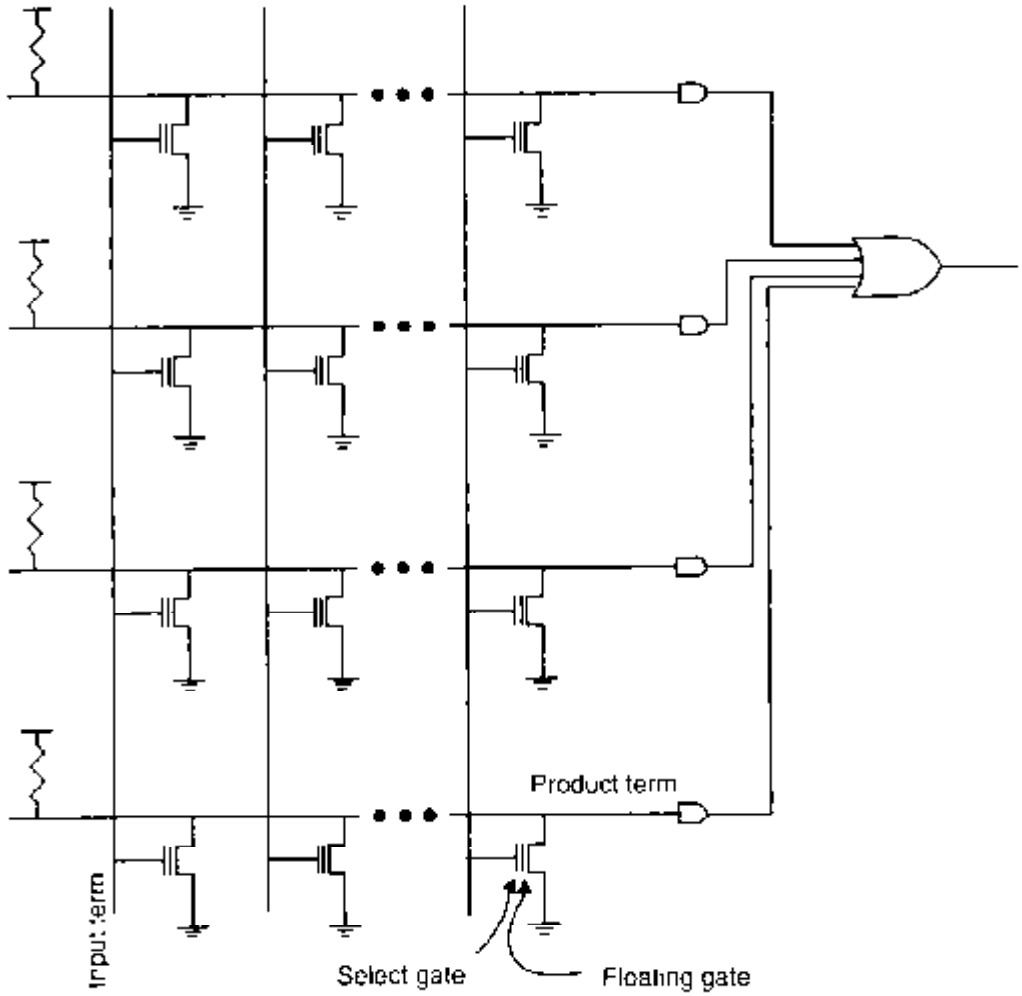


Figure 2-8 Creating a wired-AND with EPROM cells

$$\overline{A+B} = \overline{A} \cdot \overline{C}$$

EPROM celle

- Eprom er basert på FAMOS transistorer (Floating gate Avalanche MOS)

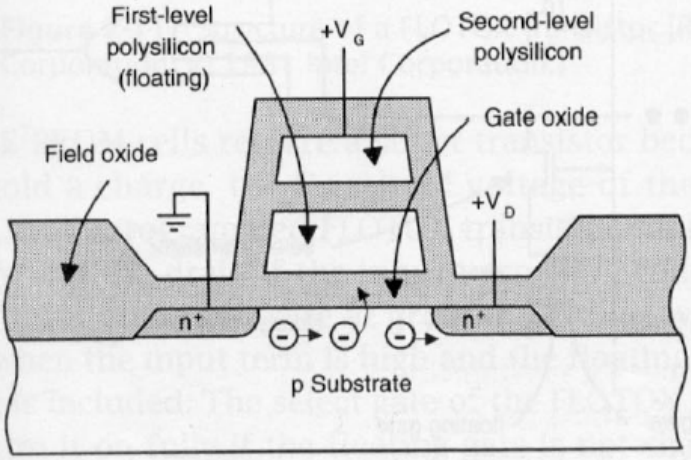
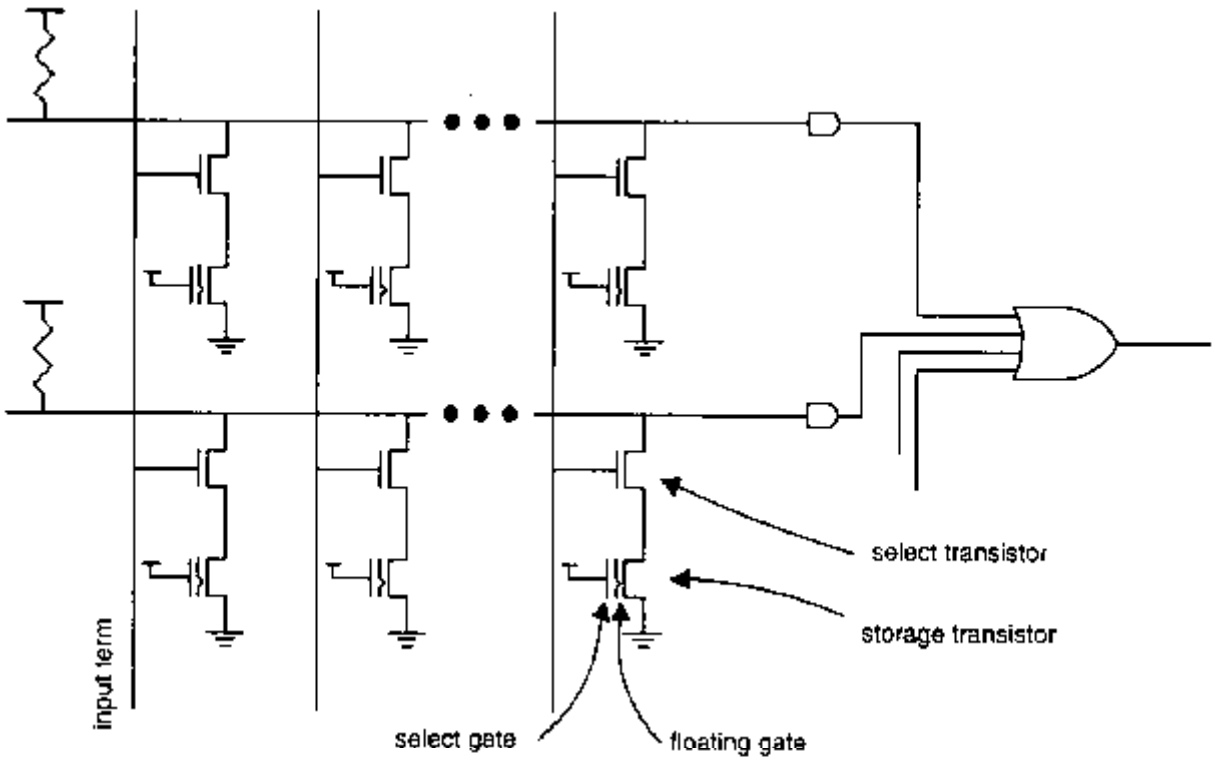


Figure 2-9 Structure of a FAMOS transistor [Reprinted by permission of Intel Corporation. © 1981 Intel Corporation.]

EEPROM celle

- ”Wired” AND med EEPROM celler



EEPROM celle

- Basert på FLOTOX transistorer
  - FLOating gate Tunnel OXide transistors

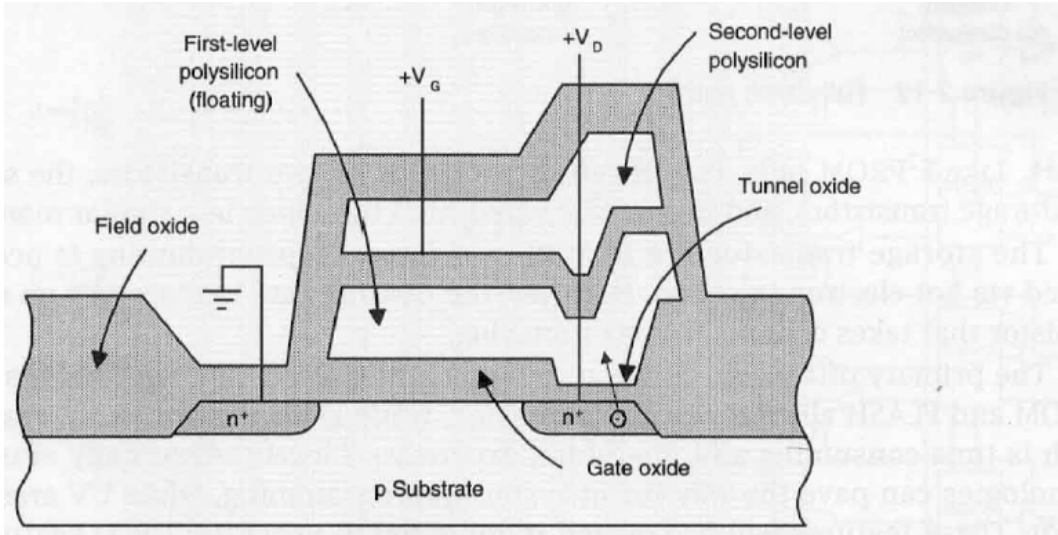
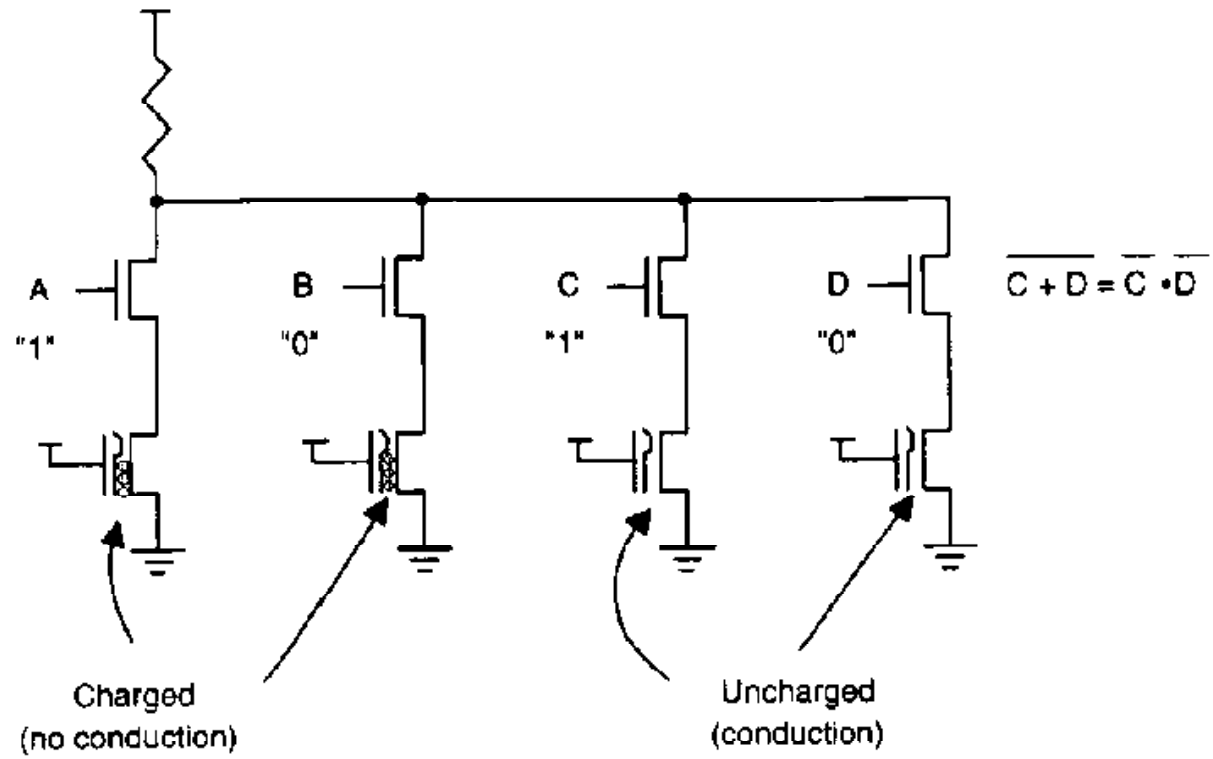


Figure 2-11 Structure of a FLOTOX transistor [Reprinted by permission of Intel Corporation. © 1981 Intel Corporation.]

FLASH

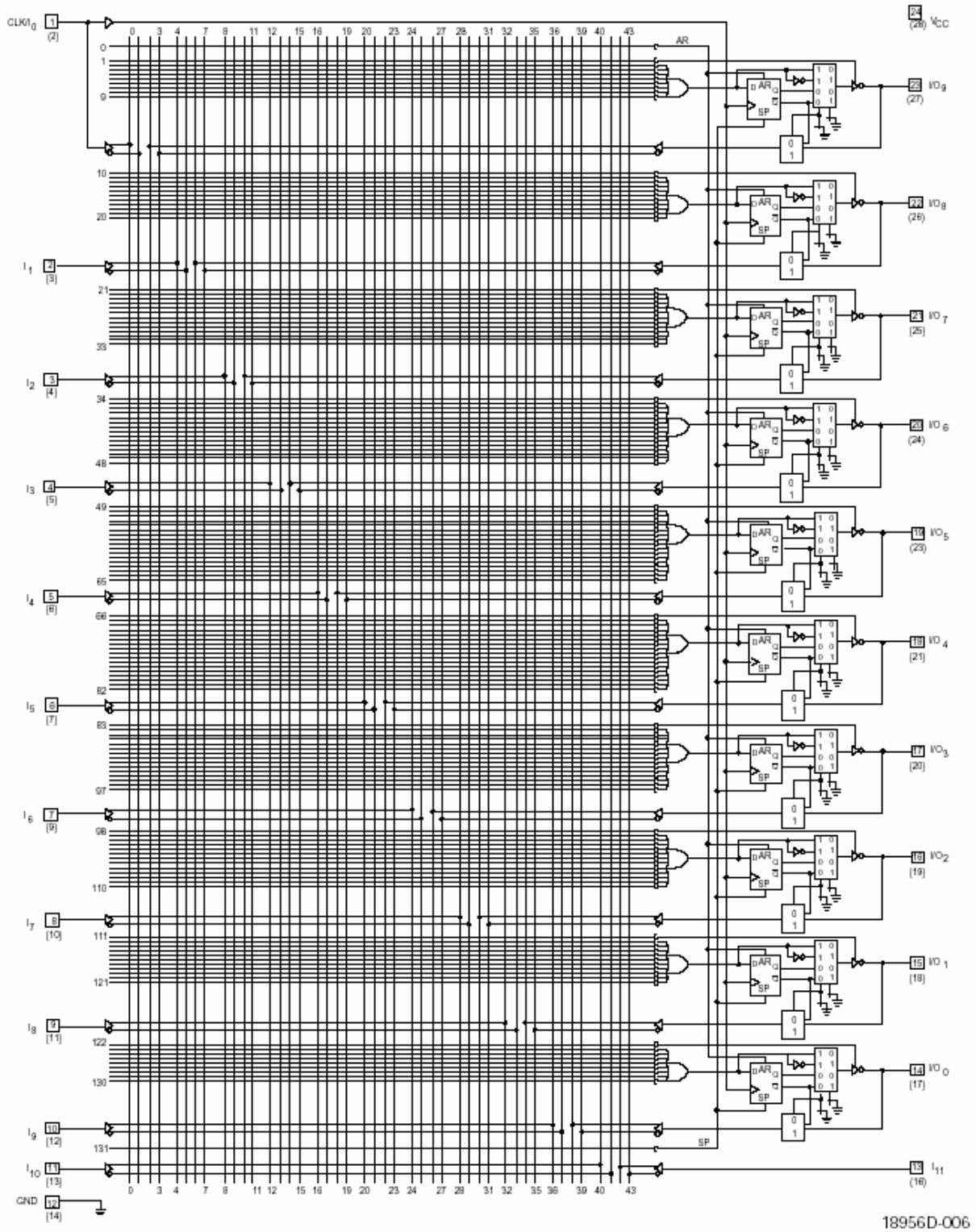
- Basert på FAMOS transistor som EPROM
- Flytende gate delt med en ”erase” transistor som tar ladninger bort ved ”tunneling”



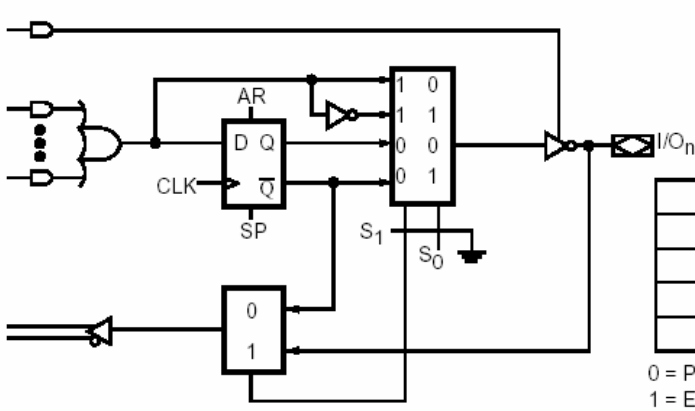
PLD-Programable Logic Device

- PLD er mer komplekse enn PAL'er
- 22V10 var den første PLD og er mye brukt

LOGIC DIAGRAM



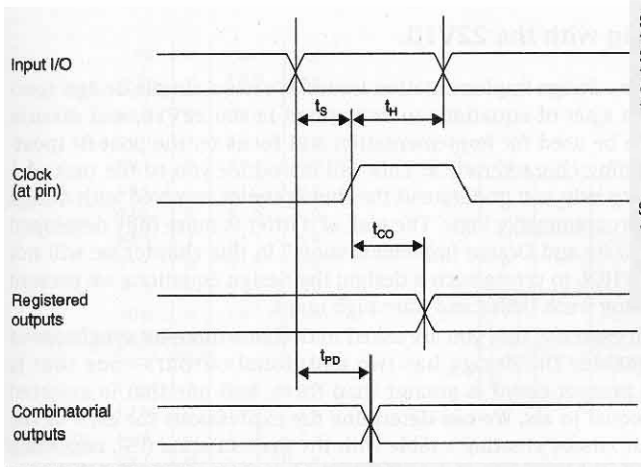
Makrocelle

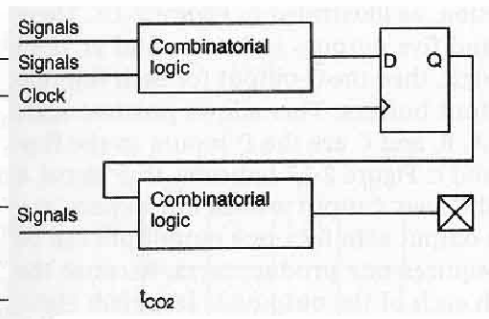


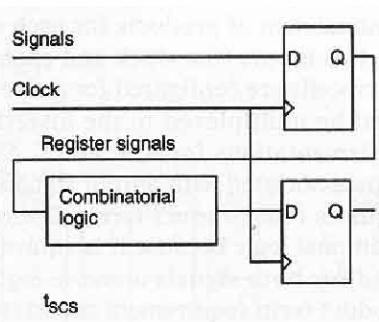
S_1	S_0	Output Configuration
0	0	Registered/Active Low
0	1	Registered/Active High
1	0	Combinatorial/Active Low
1	1	Combinatorial/Active High

0 = Programmed EE bit
1 = Erased (charged) EE bit

Timing modell







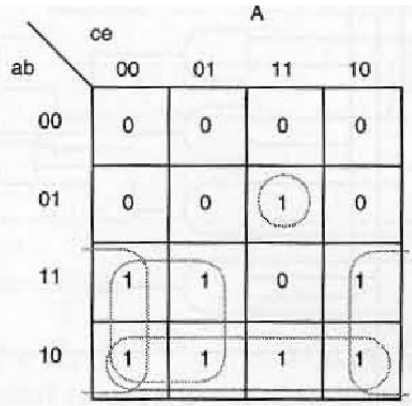
Parameter	Description	Min.	Max.
t_{PD}	Propagation delay		4 ns
t_S	Setup time	2.5 ns	
t_H	Hold time	0	
t_{CO}	Clock-to-output delay		3.5 ns
t_{CO2}	Clock-to-output delay (through logic array)		7 ns
t_{SCS}	System clock to system clock delay		5.5 ns

3-bits teller implemenert i 22V10

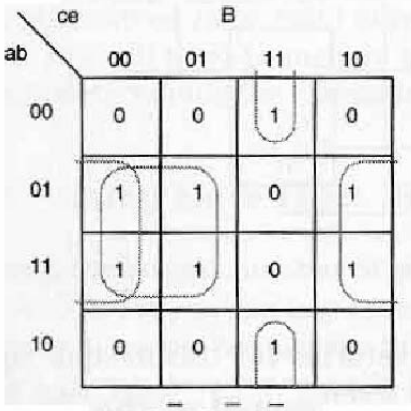
PS a b c	In e	NS A B C	Out x y
0 0 0	0	0 0 0	0 0
0 0 0	1	0 0 1	0 0
0 0 1	0	0 0 1	0 0
0 0 1	1	0 1 0	0 0
0 1 0	0	0 1 0	0 0
0 1 0	1	0 1 1	0 0
0 1 1	0	0 1 1	0 0
0 1 1	1	1 0 0	0 0
1 0 0	0	1 0 0	1 0
1 0 0	1	1 0 1	1 0
1 0 1	0	1 0 1	1 0
1 0 1	1	1 1 0	1 0
1 1 0	0	1 1 0	1 1
1 1 0	1	1 1 1	1 1
1 1 1	0	1 1 1	1 0
1 1 1	1	0 0 0	1 0

$x = a$

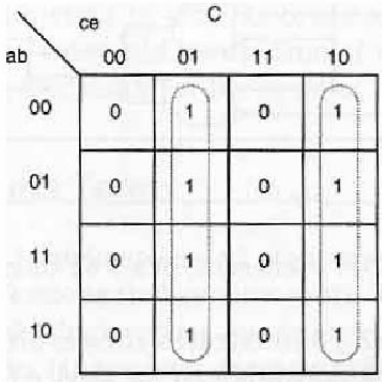
$y = a\bar{b}\bar{c}$



$A = a\bar{b} + a\bar{c} + a\bar{e} + \bar{a}bc\bar{e}$



$B = \bar{b}c + \bar{b}e + \bar{b}ce$



$C = \bar{c}e + c\bar{e}$

3 bits teller

3-bits teller implemenert i 22V10