

## 14.2 Realisierung Boolescher Funktionen durch PLAs, PROMs und PALS

Bernd Becker – Technische Informatik II

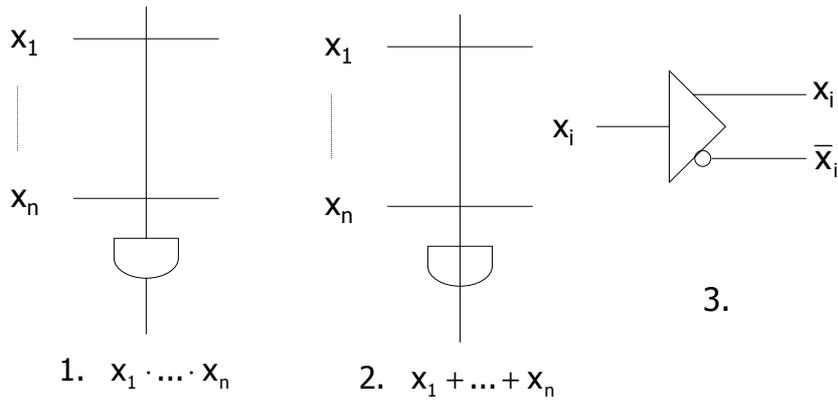
### PLAs

PLA = Programmable Logic Array

PLAs realisieren Boolesche Polynome (vgl. TI I):

1.  $x_1 \cdot \dots \cdot x_n$
2.  $x_1 + \dots + x_n$
3. 2 Treiber (nicht invertierend, invertierend)  
mit gemeinsamem Eingang

## Symbole

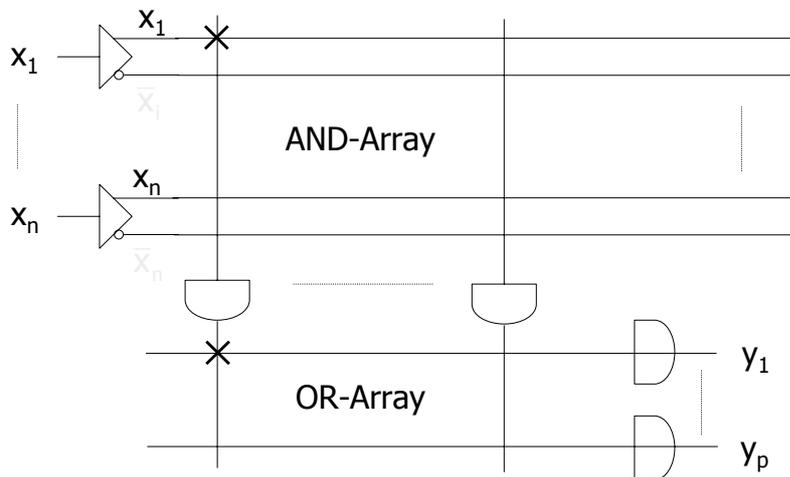


## Aufbau eines PLAs

Ein PLA besteht aus AND-Array und OR-Array.  
Das AND-Array berechnet Monome,  
das OR-Array dann daraus Polynome.

(siehe nächste Abbildung)

## AND- und OR-Array (Schaltbild)



## Erklärung der Symbole

Unterbrechungen der Verbindung eines Signals  $x_i$

zu einem n-fachen AND-Gatter

(d.h. Ersetzen von  $x_i$  durch 1)

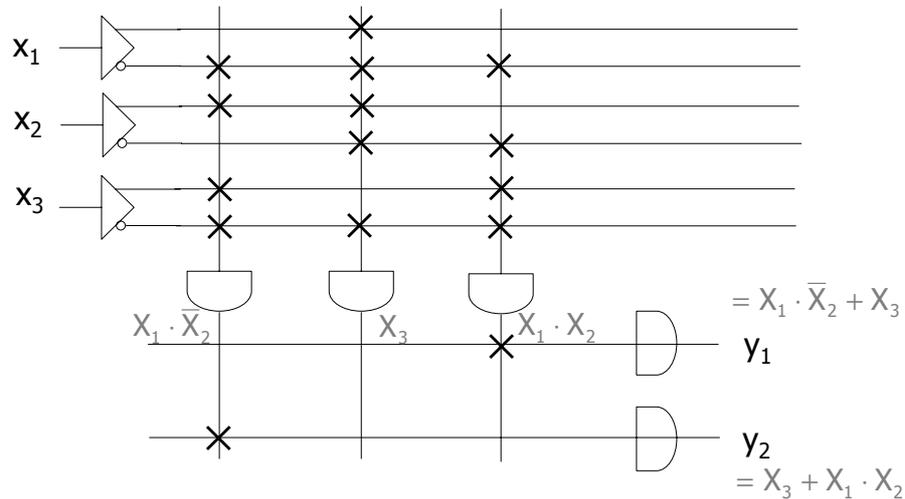
werden durch Kreuze kenntlich gemacht.

Analog für n-faches OR-Gatter

(Ersetzen von  $x_i$  durch 0).

→ Realisierung beliebiger Monome und Polynome

## Beispiel:



## Satz:

Ist  $f : \mathbf{B}^n \rightarrow \mathbf{B}^p$  (für jede Komponente  $f_i$ )

durch ein Boolesches Polynom

$$\sum_{j=1}^{k_i} m_j^i$$

darstellbar und gilt

$$|\{m_j^i \mid 1 \leq i \leq p, 1 \leq j \leq k_i\}| \leq q$$

so läßt sich  $f$  durch ein PLA mit

$q$  Spalten und  $2n+p$  Zeilen realisieren.

## PROMs

PROM = Programmable Read Only Memory

Ähnlich wie bei PLA, aber AND-Array ist fest:

Bei  $n$  Eingängen  $x_1, \dots, x_n$  werden alle  $q = 2^n$  vollständigen Monome  $x_1^{\varepsilon_1} \cdot \dots \cdot x_n^{\varepsilon_n}$  realisiert.

BB - TI II 14.2/9

## Folgen daraus:

- Bei  $p$  Zielen im OR-Array sind alle Booleschen Funktionen  $f : \mathbf{B}^n \rightarrow \mathbf{B}^p$  realisierbar
- Nach Spezialisierung des OR-Arrays für festes  $f$ : Verhalten wie ein  $2^n \cdot p$  – RAM  
Schreiben erfolgt nur einmal bei der Spezialisierung

BB - TI II 14.2/10

## **EPROMs**

Variante, bei der man die Spezialisierung des OR-Arrays rückgängig machen kann:

erasable PROM = EPROM

EPROMs können durch EPROM-Brenner neu programmiert werden.

Beispiel:

2<sup>15</sup>x8 EPROM von Hitachi (HN27C256AG-12)

Lesezugriffszeit  $t_{acc} = ( 5.0, 120.0 )$

BB - TI II 14.2/11

## **PALs**

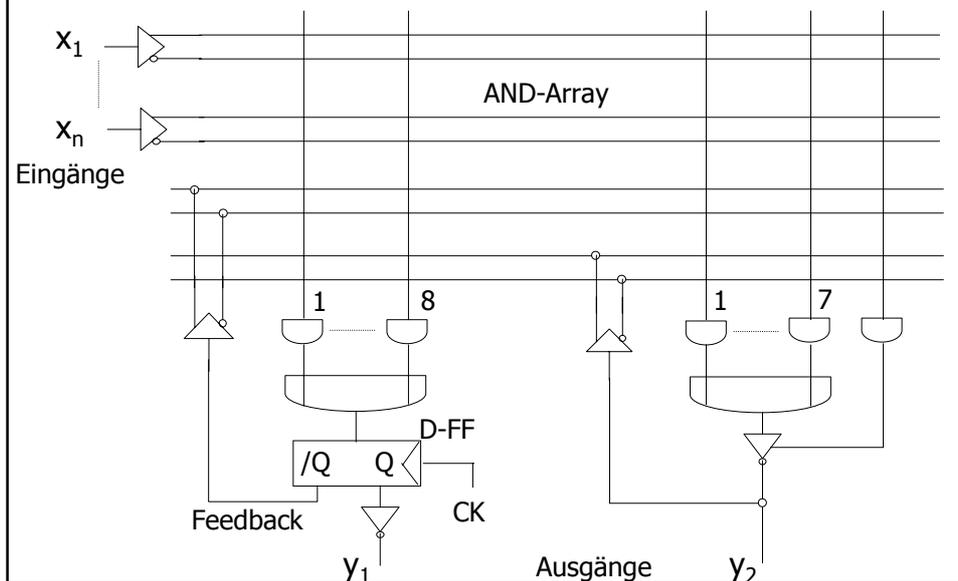
PAL = Programmable Array Logic

Wie PLA, nur diesmal ist das OR-Array von vornherein spezialisiert, das AND-Array programmierbar.

Eingänge der OR-Gatter sind mit festen AND-Gattern verbunden.

BB - TI II 14.2/12

## Schaltbild eines PAL



## Zusätzliche Besonderheiten:

- Evtl. *Registerausgänge*, d.h. Ausgänge der OR-Gatter sind mit D-FFs verbunden ( Register – PALs = PALs mit mind. 1 Registerausgang, sonst: kombinatorische PALs )
- Treiber oder invertierende Treiber vor Ausgängen; bei kombinatorischen Ausgängen häufig invertierende Tristate-Treiber, Output-enable-Signal durch AND-Gatter berechnet

## Zusätzliche Besonderheiten (ff)

- Rückkopplungen = Rückführung von Ausgängen/  
Registerausgängen ins AND-Array
  - bei kombinatorischen Ausgängen:
    - Realisierung von Funktionen in mehrstufiger Form  
(nicht nur Polynome!)
    - Wenn Ausgangstreiber disabled: Ausgangs-Pin als  
zusätzlicher Eingang in AND-Array nutzbar
  - bei Registerausgängen → Realisierung endlicher Automaten

BB - TI II 14.2/15

## Hier benutzt:

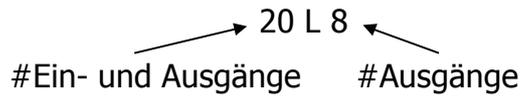
PALs von AMD mit 12 Eingängen, 8 Ausgängen und Rückkopplungen.

- Ausgänge mit invertierenden Treibern
- bei kombinatorischen Ausgängen mit Enable-Signal.

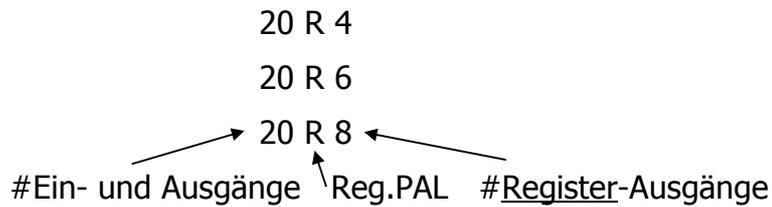
BB - TI II 14.2/16

## kombinatorisches vs. Register – PAL

- kombinatorisches PAL:



- Register – PAL:



BB - TI II 14.2/17

## Umgang mit PALs

PALs werden programmiert bzw. spezialisiert durch PAL-Brenner, z.B. gesteuert durch PCs.

Spezifikation der Booleschen Polynome durch einfache Programmiersprachen, z.B. PALASM.

BB - TI II 14.2/18

## Wesentliche Eigenschaften von PALASM:

- Signalnamen bezeichnen am PAL außen anliegende Signale,  
z.B. /x für active low – Signal,  
x für active high – Signal
- PAL-Gleichungen werden geschrieben für Ausgänge der OR-Gatter  
(Invertierung muss berücksichtigt werden)

BB - TI II 14.2/19

## Wesentliche Eigenschaften von PALASM (ff)

- Beispiel:
  - Für active low – Signal mit Name /x:  
x = P, wobei P Polynom
  - Für active high – Signal mit Name x:  
/x = P, wobei P Polynom

BB - TI II 14.2/20

## Wesentliche Eigenschaften von PALASM (ff)

- Wenn  $/x$  bzw.  $x$  kombinatorischer Ausgang mit Output enable – Signal gesteuert durch

Monom  $m$ :

IF ( $m$ )  $x = P$  bzw.

IF ( $m$ )  $/x = P$

Wenn Output immer enabled:

IF (VCC) statt IF (1)