

Prof. Dr.-Ing. Dr. h. c. J. Becker

becker@kit.edu

Karlsruher Institut für Technologie (KIT)

Digitaltechnik

Bausteine der Digitaltechnik - Schaltnetze (Teil 2) -

Schaltnetze

Allgemein: Hardwaretechnische Realisierung von Schaltnetzen

- Anwendung **schaltalgebraischer Regeln** auf **logische Ausdrücke** erlaubt **vielfältige Formen** für **Schaltnetze** → **“krause Logik“**
- **Komplexe Schaltnetze**: unübersichtliche und teilweise schwierige Verhältnisse

Benötigt: **übersichtliche Methode** zur schnellen und kostengünstigen **Realisierung** von **Schaltnetzen** in Form integrierter Schaltungen (ICs)

Grundprinzip: **Weglassen / Hinzufügen** von **Signalverbindungen** (Schalter!) in **universell** nutzbaren (“personalisierbaren”) **Hardwarestrukturen**

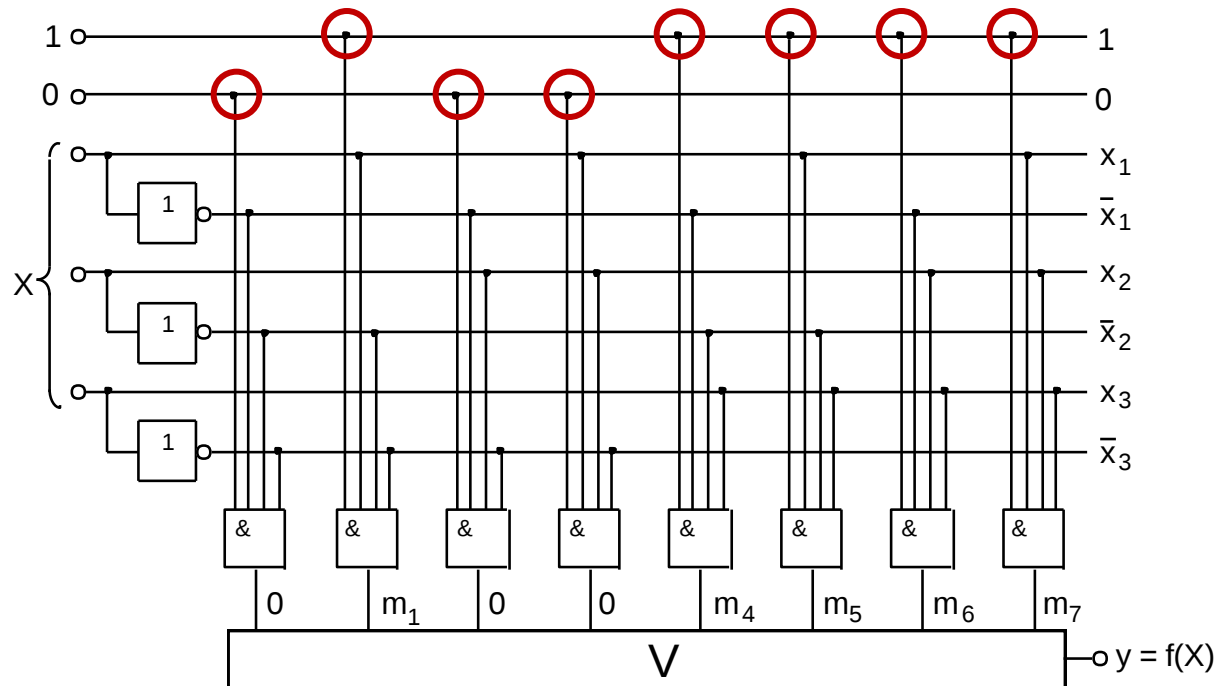
- **Prinzipielle Unterscheidung** in folgende beiden zweistufigen Klassen:
 - **Minterm-** bzw. **Maxterm-orientiert** (basierend auf **DNF** bzw. **KNF**)
 - **blockorientiert** (basierend auf **DMF** bzw. **KMF**)
- **Universelle Realisierungsmöglichkeit**:
 - > durch eine **Matrixstruktur**

Normalformorientierte Strukturen: DNF

Beispiel: ULA (Universal Logic Array)

- besteht aus 2^n UND-Schaltgliedern, zur Realisierung **jedes Minterms** in der **1. Stufe**
- einem oder mehreren **ODER-Schaltgliedern** in der **2. Stufe**
- **Personalisierung:** in 1. Stufe durch konjunktive Verknüpfung der Minterme mit 1 oder 0

Beispiel: dargestellte Funktion $y = m_1 \vee m_4 \vee m_5 \vee m_6 \vee m_7$

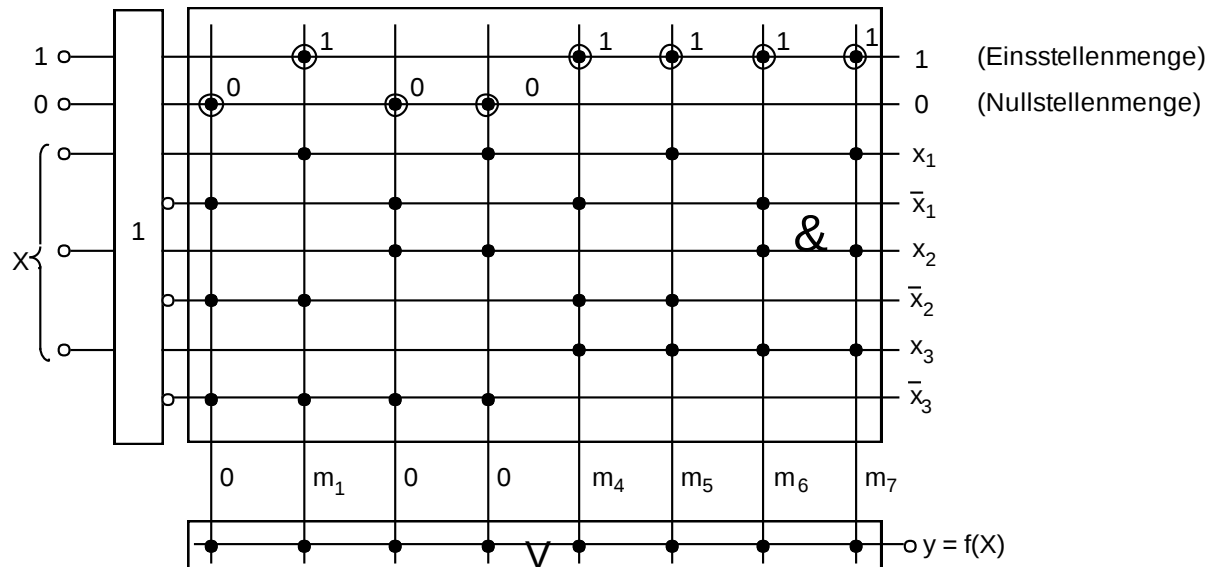


Normalformorientierte Strukturen: DNF

Vereinfachte Darstellung der Kontaktierung eines **ULA**:

- **einfache Punkte**: repräsentieren **festе Verbindungen**
- **Punkte mit einem Kreis**: **programmierbare (personalisierte) Verbindungen**

Beispiel: dargestellte Funktion $y = m_1 \vee m_4 \vee m_5 \vee m_6 \vee m_7$

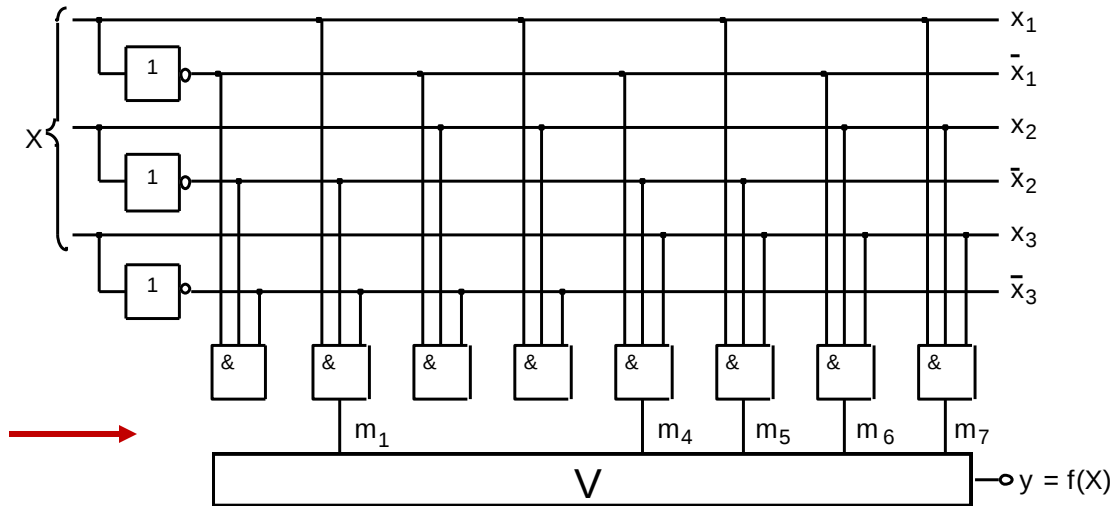


Normalformorientierte Strukturen: DNF

Beispiel: ROM (Read Only Memory)

- besteht aus **2ⁿ UND-Schaltgliedern** in der **1. Stufe**
- einem oder mehreren **ODER-Schaltgliedern** in der **2. Stufe**
- **Personalisierung**: in **2. Stufe** durch **Weglassen / Hinzufügen** von **Mintermen**

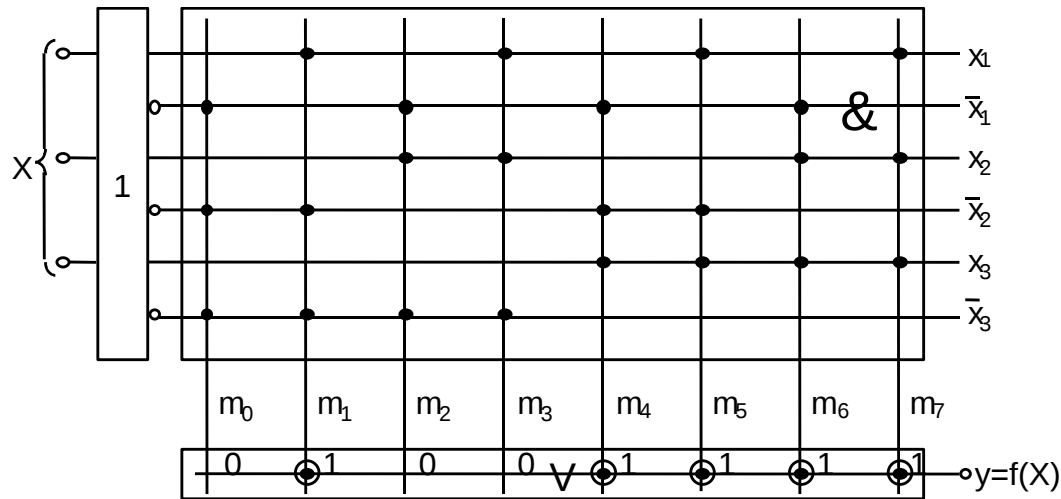
Beispiel: dargestellte Funktion $y = m_1 \vee m_4 \vee m_5 \vee m_6 \vee m_7$



Normalformorientierte Strukturen: DNF

Vereinfachte Darstellung für die Kontaktierung eines ROM:

Beispiel: dargestellte Funktion $y = m_1 \vee m_4 \vee m_5 \vee m_6 \vee m_7$

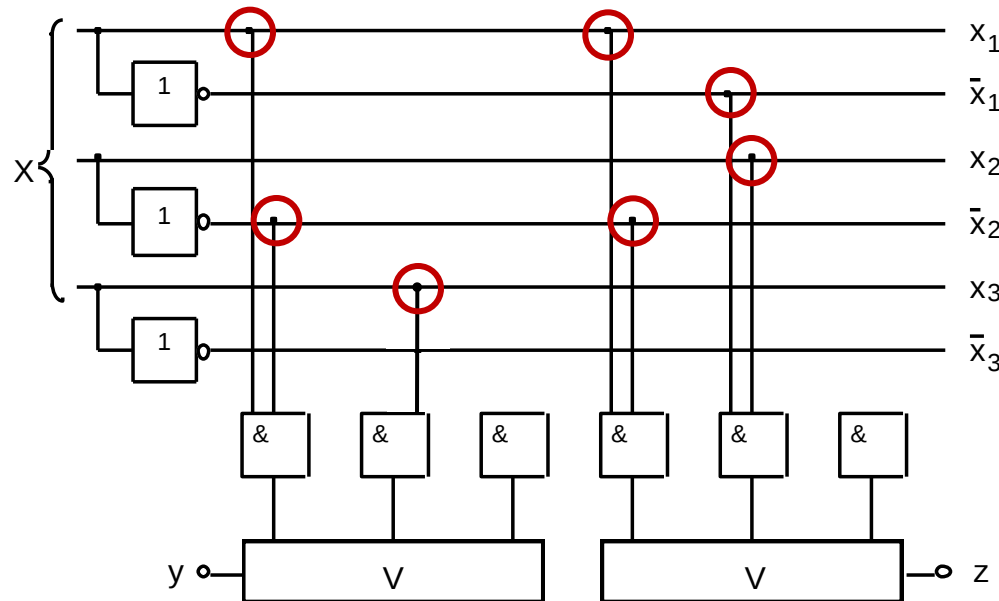


Blockorientierte Strukturen: DMF

Beispiel: PAL (Programmable Array Logic):

- besteht i.a. aus weniger als 2^n UND-Schaltgliedern in der 1. Stufe
- einem oder mehreren ODER-Schaltgliedern in der 2. Stufe
- die Personalisierung erfolgt in der 1. Stufe durch Festlegung der Primterme

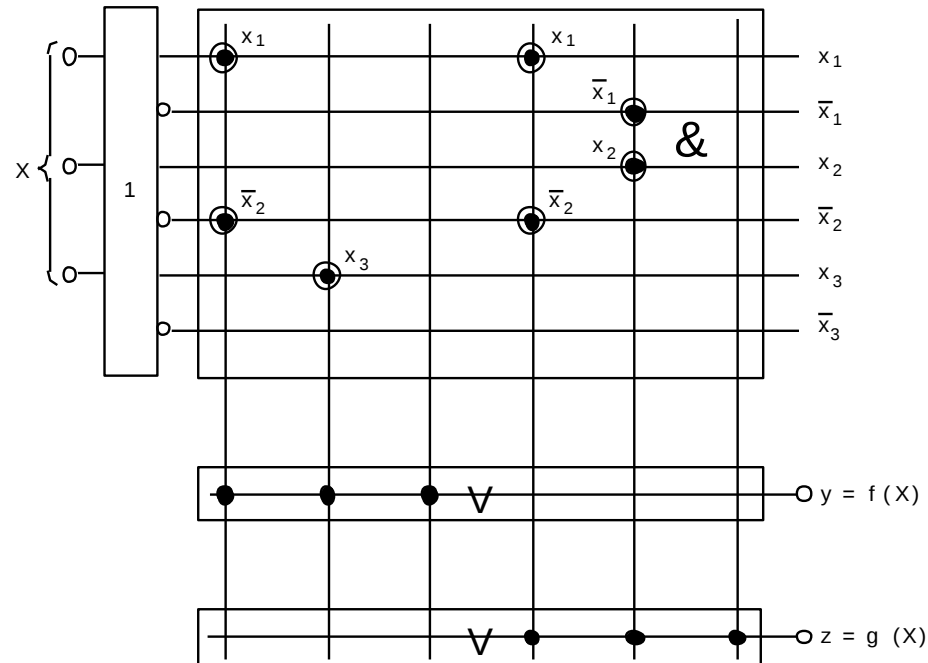
Beispiel: dargestellte Funktion $y = (\bar{x}_2 \& x_1) \vee (x_3)$, $z = (\bar{x}_2 \& x_1) \vee (x_2 \& \bar{x}_1)$



Blockorientierte Strukturen: DMF

Vereinfachte Darstellung eines PALs:

Beispiel: dargestellte Funktion $y = (\bar{x}_2 \& x_1) \vee (x_3)$, $z = (\bar{x}_2 \& x_1) \vee (x_2 \& \bar{x}_1)$



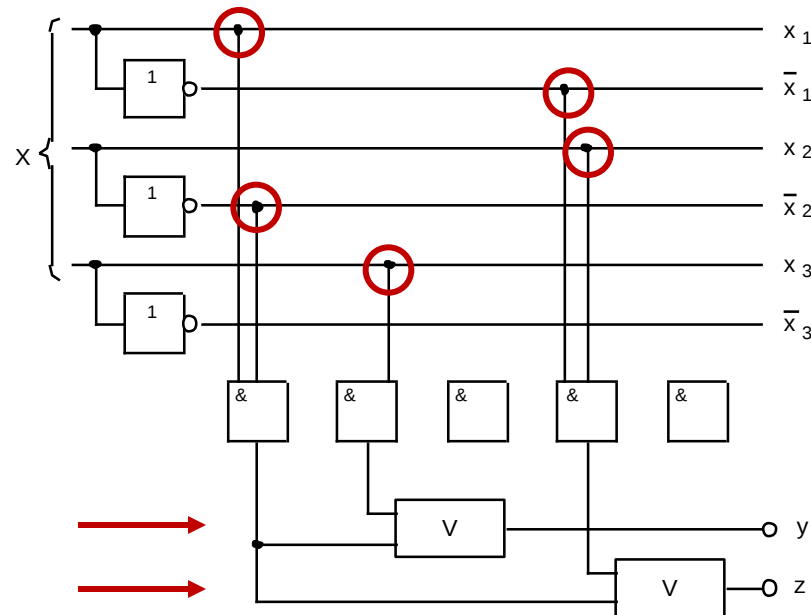
Blockorientierte Strukturen: DMF

Beispiel: PLA (Programmable Logic Array)

- Aufbau **ähnlich** zu **PALs**, jedoch **flexibler** in der **2. Stufe**
- **beide Matrizen** sind **programmierbar** (*personalisierbar*)

Vorteil: **mehrfache Ausnutzung** von **Primtermen**

Beispiel: dargestellte Funktion $y = (\bar{x}_2 \& x_1) \vee (x_3)$, $z = (\bar{x}_2 \& x_1) \vee (x_2 \& \bar{x}_1)$



Blockorientierte Strukturen: DMF

Vereinfachte Darstellung eines PLAs:

Beispiel: dargestellte Funktion $y = (\bar{x}_2 \& x_1) \vee (x_3)$, $z = (\bar{x}_2 \& x_1) \vee (x_2 \& \bar{x}_1)$

