

# Wiederholung Tutorium 5

- **Minterme:** Haben für genau 1 Belegung den Wert 1 (&)
  - Bilden die **DNF** (Disjunktive Normalform)
- **Maxterme:** Haben für genau 1 Belegung den Wert 0 (V)
  - Bilden die **KNF** (Konjunktive Normalform)
- **Primimplikanten:**
  - größtmöglicher Einser-Block im Symmetriediagramm
  - Oder: bestmöglich vereinfachter/zusammengefasster Minterm
  - Prim:* Implikant ist in keinem anderen enthalten
  - Kerne:* Implikant, der Einsstelle alleinig abdeckt
    - Bilden die **DMF** (Disjunktive Minimalform)
- **Primimplikaten:** Analoges *Primterm* für Nullstellen und Maxterme
  - Bilden die **KMF** (Konjunktive Minimalform)
- Man gelangt von der *DF* zur *KF* und umgekehrt durch ausdistribuiieren

# Nelson - Verfahren

Wird benutzt um die Menge aller *Primimplikanten* zu berechnen

1. Bildung einer vollständigen Nullblocküberdeckung der Schaltfunktion  
(Anhand des Symmetriediagramms, ohne Nutzung der Freistellen)  
 $\tau_0 = \{ \mathbf{B}_1 ; \mathbf{B}_2 ; \dots ; \mathbf{B}_N \} \quad \mathbf{B}_j = (\mathbf{X}_4, \mathbf{X}_3, \mathbf{X}_2, \mathbf{X}_1)$
2. Aufstellen einer Schaltfunktion für die Einsvervollständigung  $f^F$  bestehend aus der Konjunktion der Nullblöcken.  
Diese schreibt man als Maxterme ( $\mathbf{V}$ ), man erhält eine Art KMF
3. Ausdistribuiieren des Ausdrucks aus Punkt  
Vereinfachen/Streichen aller überflüssigen Termanteile
4. Streichen aller Terme, die nur Freistellen abdecken.

Durch Aufstellen der Einsblocküberdeckung und anschließender Disjunktion derselben erhält man am Ende die Menge aller *Primimplikaten*

# Petrack - Verfahren

- Die im Nelson-Verfahren gefunden Primterme bilden keine Minimallösung, diese wird im Petrack-Verfahren gefunden
1. Primimplikanten wird eine Präsenzvariable  $p_k$  zugewiesen.
  2. Aufstellen des **Petrack-Ausdrucks**  
(Kerne und dadurch überdeckte Spalten können in der Rechnung ausgelassen und die Kerne der endgültigen Lösung hinzugefügt werden)
  3. PA ausdistribuierten
  4. Kostengünstigste Überdeckung wählen

P1	j1	j2	j3	Pk
N1	x			p1
N2	x	x		p2
N3		x	x	p3

# Überdeckungstabelle

1. **Kerne** bestimmen, dadurch überdeckte Spalten streichen
2. Spaltendominanzen finden und **dominierende** Spalten streichen
3. Zeilendominanzen finden und **dominierte** Zeilen streichen
4. Rekursiv Schritte 1-3 wiederholen, bis Tabelle nicht mehr reduzierbar ist

$i_1$	$i_2$	
X	X	$M_1$
X		$M_2$
X	X	$M_3$

$i_1$	$i_2$	$i_3$	$i_4$	
X	X	X	X	$M_1$
X	X	X	X	$M_2$

**NEU:** Kosten der Primterme beim Streichen durch Zeilendominanzen beachten!

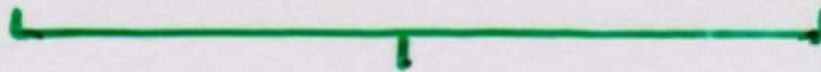
Wird eine Zeile dominiert, ist aber **kostengünstiger** als die Dominierende, werden beide behalten und keine behalten.

Übrig bleibende zyklische Resttabelle mit *Petrick-Verfahren* minimieren

# Automaten

$$A_h^v = f(E_g^v)$$

$$A_h^v = f(E_g^v, E_g^{v-1}, E_g^{v-2}, \dots, E_g^{v-n})$$



$S_k$  (zustand)

$$A_h^v = \lambda(E_g^v, S_k^v)$$

Ausgabefunktion

$$S_k^{v+1} = \delta(E_g^v, S_k^v)$$

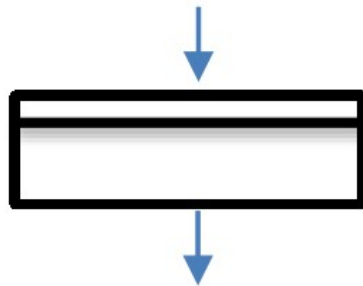
Überföhrungsfunktion

$A_h, E_g$  : beliebige  
Elemente des Ein- und  
Ausgangsalphabets

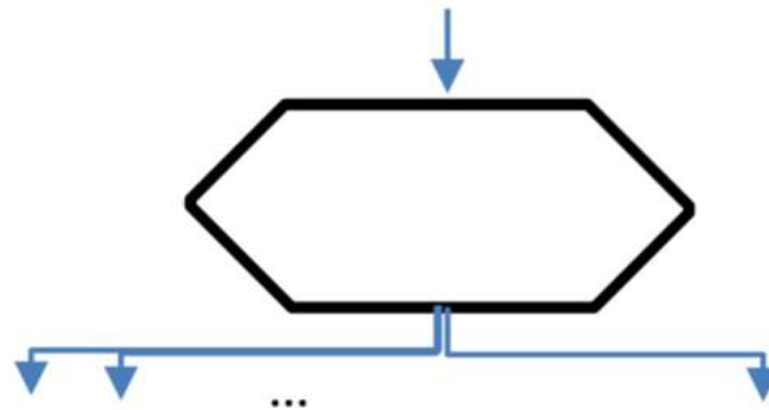
# Automatentypen

- **Mealy:** Ausgabe abhängig von Eingabe und Zustand
- **Moore:** Ausgabe nur abhängig von Zustand
- **Medwedew:** Zustand dient als Ausgabe

Zustandsübergabe



Abfrage



Ausgabe

