

1 Moore'sches Gesetz

- alle 18-24 Monate verdoppelt sich die Anzahl der Transistoren auf gleicher Fläche
- Exponentielles Wachstum der Transistorzahl, exponentieller Rückgang des Preises pro Transistor
- Herstellungskosten (Fixkosten, Variable Kosten, Technologiefaktor), Entwicklerproduktivität, Verlustleistungsdichte

2 Einheiten

Potenz	Vorsatz	Potenz	Vorsatz	$H z$	s^{-1}
10^{12}	T	10^{-1}	d	N	$kgms^{-2}$
10^9	G	10^{-2}	c	J	$Nm = VAs$
10^6	M	10^{-3}	m	W	$VA = Js^{-1}$
10^3	k	10^{-6}	μ	C	As
10^2	h	10^{-9}	n	V	$J C^{-1}$
10^1	da	10^{-12}	p	F	$C V^{-1}$
		10^{-15}	f	Ω	$V A^{-1}$
				H	$V s A^{-1}$

$Bit \xrightarrow{\cdot 8} Byte \xrightarrow{\cdot 1024} kByte \xrightarrow{\cdot 1024} MByte$

3 Boolesche Algebra

3.1 Boolesche Operatoren (Wahrheitstabelle WT)

x	y	AND $x \cdot y$	OR $x + y$	XOR $x \oplus y$	NAND $\overline{x \cdot y}$	NOR $\overline{x + y}$	EQV $\overline{x \oplus y}$
0	0	0	0	0	1	1	1
0	1	0	1	1	1	0	0
1	0	0	1	1	1	0	0
1	1	1	1	0	0	0	1

Konfiguration: $f = c_1 + c_2 + c_3 \Rightarrow cov(f) = \{c_1, c_2, c_3\}$

3.2 Gesetze der booleschen Algebra

	Boolesche Algebra ($0, 1; \cdot, +, \overline{}$)	Mengenalgebra ($P(G); \cap, \cup, \overline{}; G, \emptyset$)
Kommutativ	$x \cdot y = y \cdot x$ $x + y = y + x$	$A \cap B = B \cap A$ $A \cup B = B \cup A$
Assoziativ	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$ $x + (y + z) = (x + y) + z$	$(A \cap B) \cap C = A \cap (B \cap C)$ $(A \cup B) \cup C = A \cup (B \cup C)$
Distributiv	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$ $x + (y \cdot z) = (x + y) \cdot (x + z)$	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Äquivalenz	$x \cdot x = x$ $x + x = x$	$A \cap A = A$ $A \cup A = A$
Absorbtion	$x \cdot (x + y) = x$ $x + (x \cdot y) = x$	$A \cap (A \cup B) = A$ $A \cup (A \cap B) = A$
Konstanz	$x \cdot 1 = x$ $x + 0 = x$ $x \cdot 0 = 0$ $x + 1 = 1$	$A \cap G = A$ $A \cup \emptyset = A$ $A \cap \emptyset = \emptyset$ $A \cup G = G$
Komplement	$x \cdot \overline{x} = 0$ $x + \overline{x} = 1$ $\overline{\overline{x}} = x$	$A \cap \overline{A} = \emptyset$ $A \cup \overline{A} = G$ $\overline{\overline{A}} = A$
De Morgan	$\overline{x \cdot y} = \overline{x} + \overline{y}$ $\overline{x + y} = \overline{x} \cdot \overline{y}$	$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$
Resolution (allgemein)	$x \cdot a + \overline{x} \cdot b$ $= x \cdot a + \overline{x} \cdot b + a \cdot b$ $(x + a) \cdot (\overline{x} + b)$ $= (x + a) \cdot (\overline{x} + b) \cdot (a + b)$	$(A \cap Y) \cup (\overline{A} \cap Z)$ $= (A \cap Y) \cup (\overline{A} \cap Z) \cup (Y \cap Z)$ $(A \cup Y) \cap (\overline{A} \cup Z)$ $= (A \cup Y) \cap (\overline{A} \cup Z) \cap (Y \cup Z)$
Resolution (speziell)	$x \cdot a + \overline{x} \cdot a = a$ $(x + a) \cdot (\overline{x} + a) = a$	$(A \cap Y) \cup (\overline{A} \cap Y) = Y$ $(A \cup Y) \cap (\overline{A} \cup Y) = Y$

3.3 Boolesche Funktionen

$f : \{0, 1\}^n \rightarrow \{0, 1\}$ $f(\underline{x}) = f(x_1, x_2, \dots, x_n)$

Einsmenge F von f : $F = \{\underline{x} \in \{0, 1\}^n | f(\underline{x}) = 1\}$
Nullmenge \overline{F} von f : $\overline{F} = \{\underline{x} \in \{0, 1\}^n | f(\underline{x}) = 0\}$

Kofaktor bezüglich

- $x_i : f_{x_i} = f|_{x_i=1} = f(x_1, \dots, 1, \dots, x_n)$
- $\overline{x}_i : f_{\overline{x}_i} = f|_{x_i=0} = f(x_1, \dots, 0, \dots, x_n)$

Substitutionsregel

- $x_i \cdot f(\underline{x}) = x_i \cdot f_{x_i}$
- $\overline{x}_i \cdot f(\underline{x}) = \overline{x}_i \cdot f_{\overline{x}_i}$
- $x_i + f(\underline{x}) = x_i + f_{\overline{x}_i}$
- $\overline{x}_i + f(\underline{x}) = \overline{x}_i + f_{x_i}$

Boolesche Expansion

- $f(\underline{x}) = x_i \cdot f_{x_i} + \overline{x}_i \cdot f_{\overline{x}_i}$
- $f(\underline{x}) = (x_i + f_{\overline{x}_i}) \cdot (\overline{x}_i + f_{x_i})$
- $\overline{f(\underline{x})} = \overline{x}_i \cdot \overline{f_{\overline{x}_i}} + x_i \cdot \overline{f_{x_i}}$
- $\overline{f(\underline{x})} = (\overline{x}_i + \overline{f_{x_i}}) \cdot (x_i + \overline{f_{\overline{x}_i}})$

Eigenschaften von $f(\underline{x})$

- tautologisch $\Leftrightarrow f(\underline{x}) = 1 \quad \forall \underline{x} \in \{0, 1\}^n$
- kontradiktorisch $\Leftrightarrow f(\underline{x}) = 0 \quad \forall \underline{x} \in \{0, 1\}^n$
- unabhängig von $x_i \Leftrightarrow f_{x_i} = f_{\overline{x}_i}$
- abhängig von $x_i \Leftrightarrow f_{x_i} \neq f_{\overline{x}_i}$

3.4 Multiplexer

$f = x \cdot a + \overline{x} \cdot b$ (2 Eingänge a, b und 1 Steuereingang x)
 $f = \overline{x}_1 \overline{x}_2 a + \overline{x}_1 x_2 b + x_1 \overline{x}_2 c + x_1 x_2 d$ (Eingänge: a, b, c, d Steuerung: x_1, x_2)

3.5 Wichtige Begriffe

Wichtige Begriffe:	Definition	Bemerkung
Signalvariable	x	$\hat{x} \in \{0, 1\}$
Literal	$l_i = x_i$ oder $\overline{x_i}$	$i \in I_0 = \{1, \dots, n\}$
Literallänge	$\# \text{Teilterme} + \sum_{i=1}^{\# \text{Teilterme}} \# \text{Literale v. Teilterm}$	Summe aller Eingänge
Minterme, 0-Kuben	$MOC \ni m_j = \prod_{i \in I_0} l_i$	$ MOC = 2^n$
d-Kuben	$MC \ni c_j = \prod_{i \in I_j \subseteq I_0} l_i$	$ MC = 3^n$
Distanz	$\delta(c_i, c_j) = \{l l \in c_i \wedge \overline{l} \in c_j\} $	$\delta_{ij} = \delta(c_i, c_j)$
Implikanten	$MI = \{c \in MC c \subseteq f\}$	
Primimplikanten	$MPI = \{p \in MI p \not\subseteq c \forall c \in MI\}$	$MPI \subseteq MI \subseteq MC$
DNF (SOP)	eine Summe von Produkttermen	Terme sind ODER-verknüpft
KNF (POS)	ein Produkt von Summentermen	Terme sind UND-verknüpft
KDNF (CSOP)	Summe aller Minterme	WT: 1-Zeilen sind Minterme
KKNF (CPOS)	Menge aller Maxterme	WT: 0-Zeilen negiert sind Maxterme
VollSOP (nur 1)	Menge aller Primimplikanten	Bestimmung siehe Quine Methode oder Schichtenalgorithmus
DMF (min. 1)	Minimale Summe v. Primimplikanten	durch Überdeckungstabelle
FPGA: Field Programmable Gate Array		
LUT: Look Up Table		

4 Beschreibungsformen

4.1 Disjunktive Normalform/Sum of products (DNF/SOP)

Eins-Zeilen als Implikanten (UND) schreiben und alle Implikanten mit ODER verknüpfen:
 $Z = \overline{A} \cdot \overline{B} + \overline{C} \cdot D$

4.2 Konjunktive Normalform/Product of sums (KNF/POS)

Null-Zeilen negiert als Implikat (ODER) schreiben und alle Implikanten UND verknüpfen:
 $Z = (\overline{A} + \overline{C}) \cdot (A + \overline{D}) \cdot (\overline{B} + \overline{C}) \cdot (\overline{B} + D)$

4.3 Umwandlung in jeweils andere Form

- Doppeltes Negieren der Funktion: $Z = \overline{\overline{\overline{A} \cdot \overline{B} + \overline{C} \cdot D}}$
- Umformung "untere" Negation (DeMorgan): $Z = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D} = \overline{(A + B) \cdot (C + \overline{D})}$
- Ausmultiplizieren: $Z = \overline{(A + B) \cdot (C + \overline{D})} = \overline{A \cdot C + A \cdot \overline{D} + B \cdot C + B \cdot \overline{D}}$
- Umformung "obere" Negation (DeMorgan):
 $Z = \overline{AC \cdot A\overline{D} \cdot BC \cdot B\overline{D}} = \overline{(A + C) \cdot (A + D) \cdot (B + C) \cdot (B + D)}$

Analog von KNF (POS) nach DNF (SOP).

5 Logikminimierung

5.1 Nomenklatur

- m_i Minterm: UND-Term in dem alle Variablen vorkommen (aus KDNF)
- M_i Maxterm: ODER-Term in dem alle Variablen vorkommen (aus KKNF)
- c_i Implikant: UND-Term in dem freie Variablen vorkommen können
- C_i Implikat: ODER-Term in dem freie Variablen vorkommen können
- p_i Primimplikant: UND-Term mit maximal freien Variablen
- P_i Primimplikat: ODER-Term mit maximal freien Variablen

5.2 Karnaugh-Diagramm

Vorteile: sehr anschaulich

Nachteile: Gray-Kodierung notwendig, nur wenige Inputvariablen

Zyklische Gray-Codierung: 2dim: 00, 01, 11, 10 3dim: 000, 001, 011, 010, 110, 111, 101, 100

$\geq x \cdot y$	00	01	11	10
0	1	0	0	0
1	X	1	1	0

Gleiche Zellen zusammenfassen: z.B. $\overline{x} \overline{y} + y \cdot z$
Don't Care Werte ausnutzen!

5.3 Quine Methode

Vorteile: automatisierbar (DEA/FSM), beliebig viele Inputvariablen
Nachteile: viele paarweise Vergleiche, Erweiterung auf KKNF oder KDNF notwendig, viele Min- und Maxterme
geg.: DNF/KNF oder Wertetabelle von $f(x)$
ges.: alle Primimplikanten/-kate p_i (VollSOP/VollPOS)

Spezielles Resoluionsgesetz: $x \cdot a + \bar{x} \cdot a = a$
Absorptionsgesetz: $a + a \cdot b = a$

- kanonische Form (KKNF/KDNF) bestimmen (z.B. $f(x,y,z) = xy = xyz + xy\bar{z}$)
- Alle Min-/Maxterme in Tabelle eintragen (Index von m ist (binär)Wert des Min-/Maxterms), sortieren nach der Anzahl der positiven Literale (=Klasse)
- 1-Kubus: Min-/Maxterme die sich um eine Negation unterscheiden, zu einem Term verschmelzen (Resolutionsgesetz), dabei notieren aus welchen 0-Kuben er besteht und alle verwendeten 0-Kuben abhaken
- Der 1-Kubus muss zusammenhängend sein! (d.h. alle 1-Kubus Min-/Maxterme müssen zusammenhängen)
- Wenn möglich 2-Kubus bilden.
- Wenn keine Kubenbildung mehr möglich → Nicht abgehackte Kuben sind Primimplikanten

Beispiel (Quine Methode):

	0-Kubus	A	1-Kubus	R	A	2-Kubus	A
m_1	$\bar{x}_1\bar{x}_2x_3$	✓	\bar{x}_2x_3	$m_1\&m_5$	p_1		
m_4	$x_1\bar{x}_2\bar{x}_3$	✓	$x_1\bar{x}_2$	$m_4\&m_5$	✓	x_1	p_2
m_5	$x_1\bar{x}_2x_3$	✓	$x_1\bar{x}_3$	$m_4\&m_6$	✓		
m_6	$x_1x_2\bar{x}_3$	✓	x_1x_3	$m_5\&m_7$	✓		
m_7	$x_1x_2x_3$	✓	x_1x_2	$m_6\&m_7$	✓		

⇒ $f(x_1, x_2, x_3) = p_1 + p_2 = \bar{x}_2x_3 + x_1$

5.4 Resolventenmethode

Vorteile: Keine KDNF Notwendig, skaliert für viele Inputvariablen
Ziel: alle Primimplikanten

Wende folgende Gesetze an:
Absorptionsgesetz: $a + ab = a$
allgemeines Resolutionsgesetz: $x \cdot a + \bar{x} \cdot b = x \cdot a + \bar{x} \cdot b + ab$

Anwendung mit Schichtenalgorithmus

- schreibe die Funktion f in die 0. Schicht
- bilde **alle möglichen** Resolventen aus der 0. Schicht und schreibe sie in die nächste Schicht als ODER Verknüpfungen (Resolventen zu f "hinzufügen")
- überprüfe ob Resolventen aus der 1. Schicht Kuben aus Schicht 0 überdecken(Absorbtion) und streiche diese Kuben aus Schicht 0
- Schicht i besteht aus den möglichen Resolventen von Schicht 0 bis $(i - 1)$. Abgestrichene Kuben aus vorherigen Schichten brauchen **nicht** mehr beachtet werden.
- Sobald in der i-ten Schicht +1 steht oder keine weiteren Resolventen gebildet werden können, ist man fertig. ⇒ alle nicht ausgestrichenen Terme bilden die VollSOP

$f(x_1, \dots, x_n)$	Schicht
$x \cdot w + \bar{x} \cdot w + x \cdot y \cdot w \cdot \bar{z} + \bar{x} \cdot y \cdot w \cdot \bar{z} + \bar{y} \cdot w \cdot \bar{z}$	0
$+w + y \cdot w \cdot \bar{z}$	1
$+w \cdot \bar{z}$	2
$+w$	3

5.5 Überlagerung (Bestimmung der Minimalform)

Geg: CSOP/KDNF ($\sum m_i$) und VollSOP ($\sum p_i$) Ges: DMF (Minimalform)

5.5.1 Überdeckung:

$C = (m_0 \subseteq p_1) \cdot (m_2 \subseteq p_1 + m_2 \subseteq p_2) \stackrel{!}{=} 1$
 $C = \tau_1 \cdot (\tau_1 + \tau_2) = \tau_1 + \tau_1\tau_2 = \tau_1$

5.5.2 Alternativ: Mit Überdeckungstabelle bestimmen

Bsp:

	Minterme				
Primterme	m_1	m_2	...	m_N	$L(p_i)$
p_1	✓				$L(p_1)$
p_2	✓			✓	$L(p_2)$
⋮					⋮
p_K		✓			$L(p_K)$

K : Anzahl der Primterme
 N : Anzahl der Minterme
 $L(p_i)$: Kosten/Länge der Primimplikanten
Vorgehen:

- Kernprimimplikanten auswählen
- Spaltendominanzen prüfen und dominierende Spalten streichen
- Zeilendominanzen prüfen und dominierte Zeilen streichen
- zurück zu 2. falls keine vollständige Überdeckung

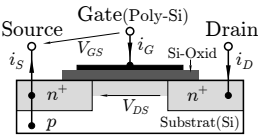
Analog auch für Bestimmung der konjunktiven Minimalform (KMF)

6 Halbleiter

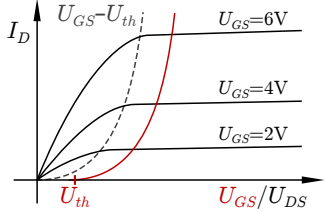
	Isolator	Metall	undotiert	N-Typ	P-Typ
Ladungsträger	Keine	e^-	e^-/e^+	e^-	e^+
Leitfähigkeit	Keine	Sehr hoch	$\propto T$	Hoch	Mittel

7 MOS-FET's

Metal Oxide Semiconductor Field Effekt Transistor



Beispiel: n-Mos
 $V_{Pinch-Off} = V_{GS} - V_{th}$



7.1 Bauteilparameter

Verstärkung: $\beta = K' \frac{W}{L}$ mit $K' = \frac{\mu \epsilon_{ox} \epsilon_0}{t_{ox}}$ $[\beta] = \frac{A}{V^2}$

Kanalweite	W
Kanallänge	L
Elektronenbeweglichkeit	$\mu \quad \mu_n \approx 250 \cdot 10^{-4} \frac{m^2}{Vs}, \mu_p \approx 100 \cdot 10^{-4} \frac{m^2}{Vs}$
rel. Dielektrizität des Gateoxyds	$\epsilon_{ox} \approx 3,9$
Dielektrizitätskonstante	$\epsilon_0 = 8.8541878 \cdot 10^{-12} \frac{As}{Vm}$
Gateoxyddicke	t_{ox}
Verstärkung	$\beta = \frac{\mu_n \epsilon_{ox} \epsilon_0}{t_{ox}} \cdot \frac{W}{L} = K' \frac{W}{L} = \frac{\mu_n C_G}{L^2}$
Kapazität	$C_G = \epsilon_{ox} \epsilon_0 \frac{WL}{t_{ox}}$
Verzögerungszeit	$t_p HL \propto \frac{C_L t_{ox} L p}{W_p \mu_p \epsilon_{ox} (V_{DD} - V_{th})}$

- große Kanalweite ⇒ große Drain-Störme
⇒ schnelle Schaltgeschwindigkeit (da $i_d \propto \beta \propto \frac{W}{L}$)
Aber: große Fläche.
- nMos schaltet schneller als pMOS, da nMOS und pMOS unterschiedliche Majoritätsladungsträger haben. Die Beweglichkeit der Löcher ist im Allgemeinen geringer als die der Elektronen.

7.2 Drainstrom

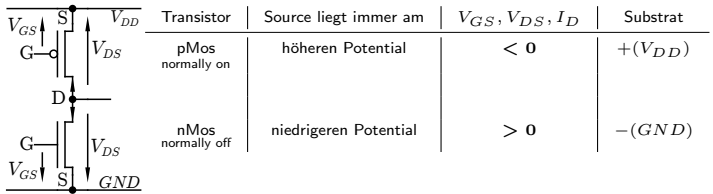
nMos (p-dotiertes Substrat, n-dotierte Drain/Source), schlechter pull up (Pegeldegenerierung)

$$I_d = \begin{cases} 0, & \text{für } U_{gs} - U_{th} \leq 0 \quad (\text{Sperrber.}) \\ \beta[(u_{gs} - U_{th}) \cdot u_{ds} - \frac{1}{2} u_{ds}^2], & \text{für } 0 \leq U_{gs} - U_{th} \leq u_{ds} \quad (\text{linearer Ber.}) \\ \frac{1}{2} \beta \cdot (u_{gs} - U_{th})^2, & \text{für } 0 \leq U_{gs} - U_{th} \leq u_{ds} \quad (\text{Sättigungsber.}) \end{cases}$$

pMos (n-dotiertes Substrat, p-dotierte Drain/Source), schlechter pull down (Pegeldegenerierung)

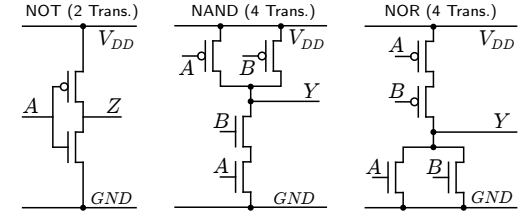
$$I_d = \begin{cases} 0, & \text{für } U_{gs} - U_{th} \geq 0 \quad (\text{Sperrber.}) \\ -\beta[(u_{gs} - U_{th}) \cdot u_{ds} - \frac{1}{2} u_{ds}^2], & \text{für } 0 \geq U_{gs} - U_{th} \leq u_{ds} \quad (\text{linearer Ber.}) \\ -\frac{1}{2} \beta \cdot (u_{gs} - U_{th})^2, & \text{für } 0 \geq U_{gs} - U_{th} \leq u_{ds} \quad (\text{Sättigungsber.}) \end{cases}$$

7.3 pMos und nMos



8 CMOS - Logik

Vorteil: (Fast) nur bei Schaltvorgängen Verlustleistung - wenig statische Verluste
Drei Grundgatter der CMOS-Technologie:



Falls GND und V_{DD} vertauscht würden, dann $NAND \rightarrow AND$ und $NOR \rightarrow OR$
Allerdings schlechte Pegelgenerierung.

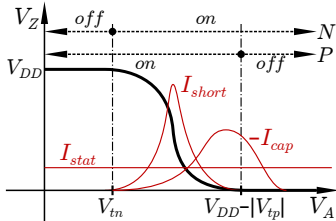
8.1 Gatterdesign

Netzwerk	Pull-Down nMos	Pull-Up pMos
NAND	Serienschaltung	Parallelschaltung
NOR	Parallelschaltung	Serienschaltung

- Möglichkeit: Direkt; ggf. Inverter vor die Eingänge und Ausgänge schalten.
- Möglichkeit: Mit boolescher Algebra die Funktion nur mit NAND und NOR darstellen.

8.2 CMOS Verlustleistung

Inverterschaltvorgang $V_A : 0 \rightarrow 1$:



Dynamische Verlustleistung $P_{dyn} = P_{cap} + P_{short}$
Kapazitive Verluste $P_{cap} = \alpha_{01} f C_L V_{DD}^2$
Kurzschlussstrom $P_{short} = \alpha_{01} f \beta_n \tau (V_{DD} - 2V_{tn})^3$

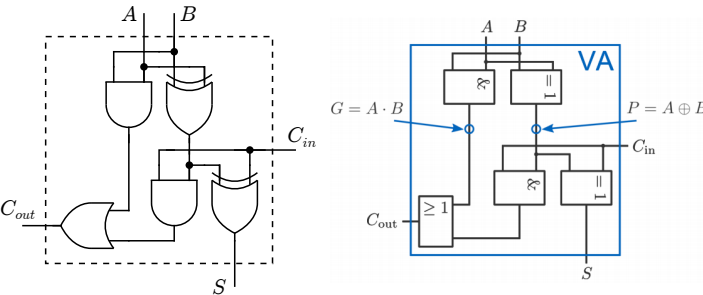
Schalzhäufigkeit $\alpha_{0 \rightarrow 1} = \frac{\text{Schaltvorgänge (pos. Flanke)}}{\text{\# Betrachtete Takte}}$
Schalzhäufigkeit (periodisch) $\alpha = \frac{f_{switch}}{f_{clk}}$

Abhängig von den Signalfanken, mit Schaltfunktionen verknüpft
 $\approx V_{DD} 1 / \propto$ Schaltzeit: $\frac{V_{DD} D2}{C_L D1} = \frac{t_{D1}}{t_{D2}}$ (bei Schaltnetzen t_{log})

Verzögerungszeit $\propto \frac{C_L t_{ox} L_p}{W_p \mu_p \varepsilon (V_{DD} - V_{th})}$
Steigend mit: Kapazitiver Last, Oxiddicke, Kanallänge, Schwellspannung
Sinkend mit: Kanalweite, Ladungsträger Beweglichkeit, Oxyd Dielektrizität, Versorgungsspannung

Statische Verlustleistung P_{stat} : Sub-Schwellströme, Leckströme, Gate-Ströme Abhängigkeit:
 $V_{DD} \uparrow: P_{stat} \uparrow$ $V_{th} \uparrow: P_{stat} \downarrow$ (aber nicht proportional)

9 Volladdierer (VA)/Ripple-C(u)arry-Adder



Generate $g_n = a_n \cdot b_n$
Propagate $p_n = a_n \oplus b_n$
Summenbit $S_n = c_n \oplus p_n = a_n \oplus b_n \oplus c_n$
 $S_n = \underbrace{a_n b_n \bar{c}_n + \bar{a}_n b_n \bar{c}_n + \bar{a}_n b_n c_n}_{\text{genau ein Eingang high}} + \underbrace{a_n b_n c_n}_{\text{alle Eingänge high}}$ (Ungerade Anzahl von Eingängen 1)
Carry-out $c_{n+1} = c_n \cdot p_n + g_n$
 $c_{n+1} = \underbrace{a_n b_n \bar{c}_n + a_n b_n c_n + \bar{a}_n b_n c_n}_{\text{zwei Eingänge 1}} + \underbrace{a_n b_n c_n}_{\text{drei Eingänge 1}}$ (Mindestens zwei Eingänge 1)

Laufzeiten
 $t_{sn} = \begin{cases} t_{cn} + t_{xor} & t_{cn} > t_{xor} \\ 2t_{xor} & \text{sonst} \end{cases}$
 $t_{cn+1} = \begin{cases} t_{and} + t_{or} & a_n = b_n = 1 \\ t_{xor} + t_{and} + t_{or} & a_n = b_n = 0 \\ t_{cn} + t_{and} + t_{or} & a_n \neq b_n \end{cases} \quad \begin{matrix} (g_n = 1) \\ (p_n = 0, g_n = 0) \\ (p_n = 1) \end{matrix}$

10 Sequentielle Logik

Logik mit Gedächtnis (Speicher).

10.1 Begriffe/Bedingungen

t_{Setup}	Stabilitätszeit vor der aktiven Taktflanke
t_{hold}	Stabilitätszeit nach der aktiven Taktflanke
t_{c2q}	Eingang wird spätestens nach t_{c2q} am Ausgang verfügbar
Min. Taktperiode	$t_{clk} \geq t_{1,c2q} + t_{logic,max} + t_{2,setup}$
Max. Taktfrequenz	$f_{max} = \left\lfloor \frac{1}{t_{clk}} \right\rfloor$ (Nicht aufrunden)
Holdzeitbedingung	$t_{hold} \leq t_{c2q} + t_{logic,min} \rightarrow$ Dummy Gatter einbauen
Durchsatz	$\frac{1_{sample}}{t_{clk,pipe}} = f$ (Sample: Anzahl der Eingänge ins Register)
Latenz	$t_{clk} \cdot \text{\#PipelineStufen}$ (Anzahl von Logik+Register-Blöcken)

10.2 Pipelining

Nur bei synchronen(taktgesteuerten) Schaltungen möglich!

- Aufteilen langer kombinatorischer Pfade durch Einfügen zusätzlicher Registerstufen
→ Möglichst Halbierung des längsten Pfades
- Zeitverhalten beachten (evtl. Dummy-Gatter einfügen)
- Durchsatz erhöht sich entsprechend der Steigerung der Taktfrequenz
- Gesamtlatenz wird eher größer
- Taktfrequenz erhöht sich

10.3 Parallel Processing

$$\text{Durchsatz} = \frac{\text{\#Modul}}{t_{clk,Modul}} = f \quad \text{Latenz} = t_{clk}$$

- Paralleles, gleichzeitiges Verwenden mehrere identischer Schaltnetze
- Zusätzliche Kontrolllogik nötig (Multiplexer)
- Taktfrequenz und Latenz bleiben konstant
- Durchsatz steigt mit der Zahl der Verarbeitungseinheiten
ABER: deutlich höherer Ressourcenverbrauch

11 Speicherelemente

Flüchtig Speicherinhalt gehen verloren, wenn Versorgungsspannung V_{DD} wegfällt - Bsp: *RAM
Nicht Flüchtig Speicherinhalt bleibt auch ohne V_{DD} erhalten - Bsp: Flash
Asynchron Daten werden sofort geschrieben/gelesen.
Synchron Daten werden erst mit $clk_{0 \rightarrow 1}$ geschrieben.
Dynamisch Ohne Refreshzyklen gehen auch bei angelegter V_{DD} Daten verloren - Bsp: DRAM
Statisch Behält den Zustand bei solange V_{DD} anliegt (keine Refreshzyklen nötig) - Bsp: SRAM
Bandbreite: Bitanzahl, die gleichzeitig gelesen/geschrieben werden kann. **Latenz**: Zeitverzögerung zwischen Anforderung und Ausgabe von Daten. **Zykluszeit**: Minimale Zeitdifferenz zweier Schreib/Lesezugriffe.

$$\text{Speicherkapazität} = \text{Wortbreite} \cdot 2^{\text{Adressbreite}}$$

11.1 Speicherzelle/Register

Ring aus zwei Invertern.

11.2 Latch (Pegelgesteuert)

Set-Reset Latch:
Zwei gegenseitig rückgekoppelte NAND-Gatter. 0 an R/S schaltet.
Enable-Latch: ändert Speicherzustand auf D nur wenn $e = 1$

e	Q
0	Q
1	D

11.3 Flip-Flop (Flankengesteuert)

Besteht aus zwei enable-Latches
Flip-Flop: Ändert Zustand bei steigender/(fallender) Taktflanke.

clk	Q	\overline{Q}
$0 \rightarrow 1$	D	\overline{D}
sonst	Q	\overline{Q}

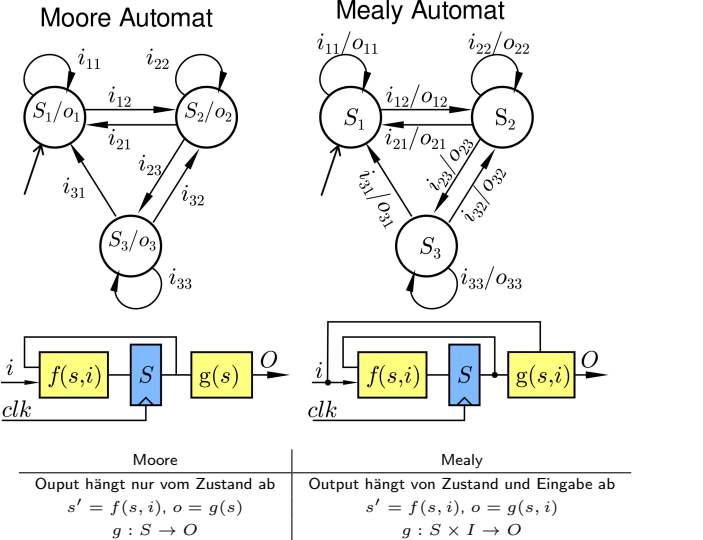
12 Automaten (FSM)

12.1 Deterministic finite state machine (DFA)

DFA 6-Tupel $\{I, O, S, R, f, g\}$

I	Eingabealphabet
O	Ausgabealphabet
S	Menge von Zuständen
$R \subseteq S$	Menge der Anfangszustände
$f: S \times I \rightarrow S$	Übergangsrelation
g	Ausgaberation

12.2 Moore und Mealy FSMs



Vorteile

- Kein kombinatorischer Pfad von Eingängen zu Ausgängen
- Wichtig für Begrenzung der Logiktiefe in sequentiellen Schaltwerken, insb. bei Verkettung

Nachteile

- Hohe Anzahl an Zuständen

Vorteile

- Weniger Zustände
- Übersichtliche Beschreibung
- Allgemeinster Fall einer FSM

Nachteile

- Lange kombinatorische Pfade bei Verkettung
- in der Praxis zu vermeiden

Beim Zeichnen jede Eingabemöglichkeit für jeden Zustand berücksichtigen und Startzustand mit leerem Pfeil kennzeichnen.