Digitaltechnik

1 Moore'sches Gesetz

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

2 Einheiten

Potenz	Vorsatz
10^{12}	T
10^{9}	G
10^{6}	M
10^{3}	k
10^{2}	h
10^{1}	da

Potenz	Vorsatz
10^{-1}	d
10^{-2}	c
10^{-3}	m
10^{-6}	μ
10^{-9}	n
10^{-12}	p
10^{-15}	f

Hz	s^{-1}
N	$kgms^{-2}$
J	$Nm = VAs$
W	$VA = Js^{-1}$
C	As
V	JC^{-1}
F	CV^{-1}
Ω	VA^{-1}
H	VsA^{-1}

$$Bit \xrightarrow{\cdot 8} Byte \xrightarrow{\cdot 1000} kByte \xrightarrow{\cdot 1000} MByte$$
$$Bit \xrightarrow{\cdot 8} Byte \xrightarrow{\cdot 1024} KiB \xrightarrow{\cdot 1024} MiB$$

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

3.3 Boolesche Funktionen

$$f : \{0,1\}^n \to \{0,1\} \qquad f(\underline{\boldsymbol{x}}) = f(x_1, x_2, \ldots, x_n)$$

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

Kofaktor bezüglich

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

Substitutionsregel

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

Boolsche Expansion

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

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

- tautologisch $\Leftrightarrow f(\underline{\boldsymbol{x}}) = 1 \qquad \forall \underline{\boldsymbol{x}} \in \{0,1\}^n$
- kontradiktorisch $\Leftrightarrow f(\underline{\boldsymbol{x}}) = 0 \qquad \forall \underline{\boldsymbol{x}} \in \{0,1\}^n$
- 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 $\overline{x_i}$	$i \in I_0 = \{1, \ldots, n\}$
Literallänge	#Teilterme $+ \sum_{i=1}^{\#\text{Teilterme}}$ #Literale v. Teilterm	Summe aller Eingänge
Minterme,0-Kuben	M0C $\ni m_j = \prod_{i \in I_0} l_i$	$\lvert$M0C$\rvert = 2^n$
d-Kuben	MC $\ni c_j = \prod_{i \in I_j \subseteq I_0} l_i$	$\lvert$MC$\rvert = 3^n$
Distanz	$\delta(c_i, c_j) = \lvert\{l \mid l \in c_i \wedge \overline{l} \in c_j\}\rvert$	$\delta_{ij} = \delta(c_i, c_j)$
Implikanten	$MI = \{c \in MC \mid c \subseteq f\}$	
Primimplikanten	$MPI = \{p \in MI \mid p \not\subset c \; \forall c \in MI\}$	$MPI \subseteq MI \subseteq MC$

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 Implikaten **UND** verknüpfen:
$Z = (\overline{A} + \overline{C}) \cdot (\overline{A} + \overline{D}) \cdot (\overline{B} + \overline{C}) \cdot (\overline{B} + D)$

4.3 Umwandlung in jeweils andere Form

1. Doppeltes Negieren der Funktion: $Z = \overline{\overline{\overline{A} \cdot \overline{B} + \overline{C} \cdot D}}$
2. Umformung "untere" Negation (DeMorgan) : $Z = \overline{\overline{\overline{A} \cdot \overline{B}} \cdot \overline{\overline{C} \cdot D}} = \overline{(A + B) \cdot (C + \overline{D})}$
3. Ausmultiplizieren: $Z = (A + B) \cdot (C + \overline{D}) = A \cdot C + A \cdot \overline{D} + B \cdot C + B \cdot \overline{D}$
4. Umformung "obere" Negation (DeMorgan) :

$Z = \overline{AC} \cdot \overline{A\overline{D}} \cdot \overline{BC} \cdot \overline{B\overline{D}} = (\overline{A} + \overline{C}) \cdot (\overline{A} + D) \cdot (\overline{B} + \overline{C}) \cdot (\overline{B} + D)$

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
- K_{pi} Kernprimimplikant: Überdeckt einen Minterm als **einziger** c_i
- K_{Pi} Kernprimimplikat: Überdeckt einen Maxterm als **einziger** C_i

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

$z \diagdown xy$	00	01	11	10
0	1	0	0	0
1	X	1	1	0

Gleiche Zellen zusammenfassen: z.B. $\overline{xy} + yz$
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 Resolutionsgesetz: $x \cdot a + \overline{x} \cdot a = a$
Absorptionsgesetz: $a + a \cdot b = a$

1. kanonische Form (KKNF/KDNF) bestimmen (z.B. $f(x,y,z) = xy = xyz + xy\overline{z}$)
2. Alle Min-/Maxterme in Tabelle eintragen (Index von m ist (binär)Wert des Min-/Maxterms), sortieren nach der Anzahl der positiven Literale (=Klasse)
3. 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
4. Der 1-Kubus muss zusammenhängend sein! (d.h. alle 1-Kubus Min-/Maxterme müssen zusammenhängen)
5. Wenn möglich 2-Kubus bilden.
6. Wenn keine Kubenbildung mehr möglich → Nicht abgehakte Kuben sind Primimplikanten

Beispiel (Quine Methode):

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

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

5.4 Resolventenmethode

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

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

Anwendung mit Schichtenalgorithmus

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

$f(x_1,\ldots,x_n)$	Schicht
$x \cdot w + \overline{x} \cdot w + x \cdot y \cdot w \cdot \overline{z} + \overline{x} \cdot y \cdot w \cdot \overline{z} + \overline{y} \cdot w \cdot \overline{z}$	0
$+w + y \cdot w \cdot \overline{z}$	1
$+w \cdot \overline{z}$	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:

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

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:

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

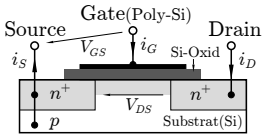
Analog auch für Bestimmung der konjunktiven Minimalform (KMF)

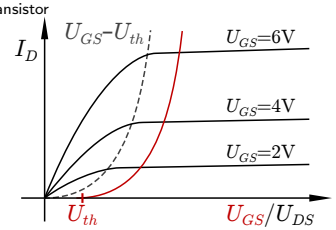
6 Halbleiter

	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 **O**xide **S**emiconductor **F**ield **E**ffekt **T**ransistor





Bereiche NMOS:

Sperrbereich: $U_{GS} < U_{th}$
Linearer Bereich: $U_{DS} < U_{GS} - U_{th}$
Pinch-Off: $U_{DS} = U_{GS} - U_{th}$
Sättigung: $U_{DS} > U_{GS} - U_{th}$

7.1 Bauteilparameter

Verstärkung: $\beta = K' \frac{W}{L}$ mit $K' = \frac{\mu \varepsilon_{ox} \varepsilon_0}{t_{0x}}$ $[\beta] = \frac{A}{V^2}$

Kanalweite	W
Kanallänge	L
Elektronenbeweglichkeit	μ $\mu_n \approx 250 \cdot 10^{-4} \frac{m^2}{Vs}$, $\mu_p \approx 100 \cdot 10^{-4} \frac{m^2}{Vs}$
rel. Dielektrizität des Gateoxyds	$\varepsilon_{ox} \approx 3,9$
Dielektrizitätskonstante	$\varepsilon_0 = 8.8541878 \cdot 10^{-12} \frac{\mathrm{A\,s}}{\mathrm{V\,m}}$
Gateoxyddicke	t_{ox}
Verstärkung	$\beta = \frac{\mu_n \varepsilon_{ox} \varepsilon_0}{t_{ox}} \cdot \frac{W}{L} = K' \frac{W}{L} = \frac{\mu_n C_G}{L^2}$
Kapazität	$C_G = \varepsilon_{ox} \varepsilon_0 \frac{WL}{t_{ox}}$
Verzögerungszeit	$t_{pHL} \propto \frac{C_L t_{ox} L_p}{W_p \mu_p \varepsilon_{ox} (V_{DD} - \lvert V_{th} \rvert)}$

- große Kanalweite $\Rightarrow$ große Drain-Störme
  $\Rightarrow$ 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

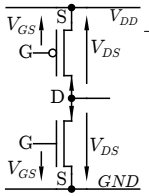
nMos (p-dotiertes Substrat, n-dotierte Drain/Source), schlechter pull up (Pegeldegenerierung)

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

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

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

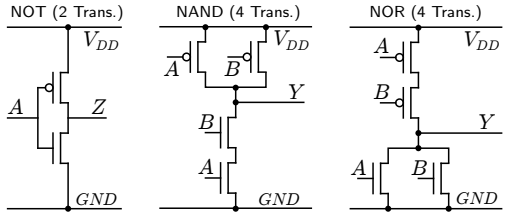
7.3 pMos und nMos



Transistor	Source liegt immer am	V_{GS}, V_{DS}, I_D	Substrat
pMos normally on	höheren Potential	$< \mathbf{0}$	$+(V_{DD})$
nMos normally off	niedrigeren Potential	$> \mathbf{0}$	$-(GND)$

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.

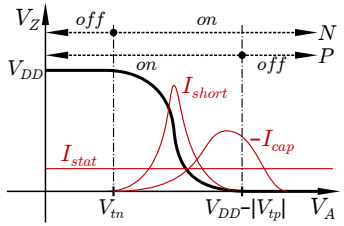
8.1 Gatterdesign

Netzwerk	Pull-Down	Pull-Up
Transistoren	**n**Mos	**p**Mos
NAND	Serienschaltung	Parallelschaltung
NOR	Parallelschaltung	Serienschaltung

1. Möglichkeit: Direkt; ggf. Inverter vor die Eingänge und Ausgänge schalten.
2. 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 $P_{short} = \alpha_{01} f \beta_n \tau (V_{DD} - 2V_{tn})^3$

Schalthäufigkeit $\alpha_{0\to1} = \frac{\text{Schaltvorgänge(pos. Flanke)}}{\text{\# Betrachtete Takte}}$

Schalthäufigkeit (periodisch) $\alpha = \frac{f_{\text{switch}}}{f_{\text{clk}}}$

Abhängig von den Signalflanken, mit Schaltfunktionen verknüpft

$\approx V_{DD}1/ \propto$ Schaltzeit: $\frac{V_{DD2}}{V_{DD1}} = \frac{t_{D1}}{t_{D2}}$ (bei Schaltnetzen t_{log})

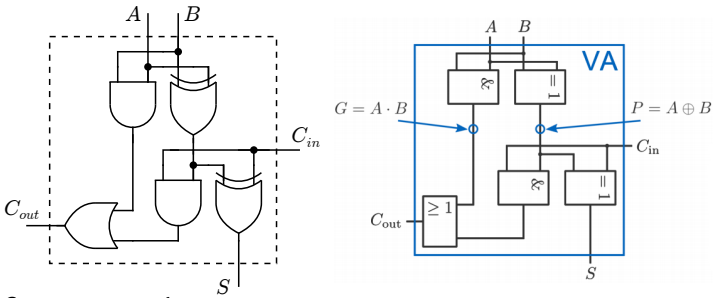
Verzögerungszeit $\propto \frac{C_L t_{ox} L_p}{W_p \mu_p \varepsilon (V_{DD} - V_{th})}$

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 $S_n = c_n \oplus p_n = a_n \oplus b_n \oplus c_n$

$S_n = \underbrace{a_n \overline{b_n} \overline{c_n} + \overline{a_n} b_n \overline{c_n} + \overline{a_n} \overline{b_n} c_n}_{\text{genau ein Eingang high}} + \underbrace{a_n b_n c_n}_{\text{alle Eingänge high}}$ (Ungerade Anzahl von Eingängen 1)

Carry-out $c_{n+1} = c_n \cdot p_n + g_n$

$c_{n+1} = \underbrace{a_n b_n \overline{c_n} + a_n \overline{b_n} c_n + \overline{a_n} b_n c_n}_{\text{zwei Eingänge 1}} + \underbrace{a_n b_n c_n}_{\text{drei Eingänge 1}}$ (Mindesten zwei Eingänge 1)

Laufzeiten

$$t_{sn} = \begin{cases} t_{cn} + t_{xor} & t_{cn} > t_{xor} \\ 2t_{xor} & sonst \end{cases}$$

$$t_{cn+1} = \begin{cases} t_{and} + t_{or} & a_n = b_n = 1 \quad (g_n = 1) \\ t_{xor} + t_{and} + t_{or} & a_n = b_n = 0 \quad (p_n = 0, g_n = 0) \\ t_{cn} + t_{and} + t_{or} & a_n \neq b_n \quad (p_n = 1) \end{cases}$$

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	$t_{clk} > t_{1,c2q} + t_{logic,max} + t_{2,setup}$
Max. Taktfrequenz	$f_{max} = \lfloor \frac{1}{t_{clk}} \rfloor$ (Nicht aufrunden)
Holdzeitbedingung	$t_{2,hold} < t_{1,c2q} + t_{logic,min} \rightarrow$ Dummy Gatter einbauen
Durchsatz	$\frac{\text{1Sample}}{t_{clk,pipe}} = f$ (Sample: Anzahl der Eingänge ins Register)
Latenz	$t_{clk} \cdot$ \#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

Durchsatz $= \frac{\#\text{Modul}}{t_{clk,Modul}} = f$ 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 $clk_{0\to1}$ 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.

Speicherkapazität = 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	$\overline{Q}$
$0 \to 1$	D	$\overline{D}$
sonst	Q	$\overline{Q}$

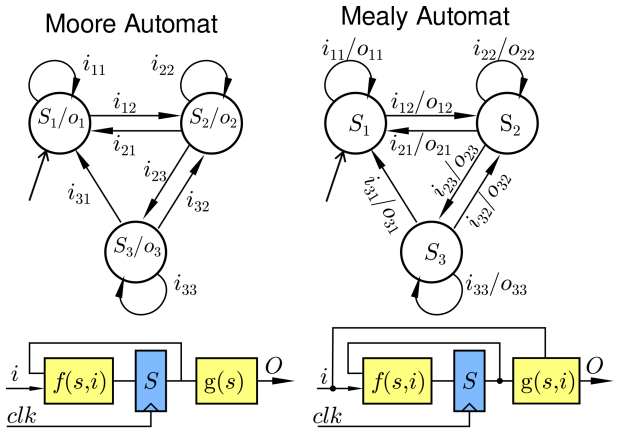
12 Automaten (FSM)

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
$f : S \times I \to S$	Übergangsrelation
g	Ausgaberelation

12.2 Moore und Mealy FSMs



Moore	Mealy
Ouput hängt nur vom Zustand ab	Output hängt von Zustand und Eingabe ab
$s' = f(s,i),\ o = g(s)$	$s' = f(s,i),\ o = g(s,i)$
$g : S \to O$	$g : S \times I \to O$

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.

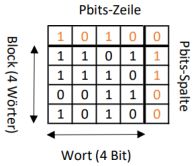
13 Block Parity Schaltungen

13.1 Paritätsprüfsumme

Konzept zur Absicherung von digitalen Daten durch Anhängen eines Paritätsbits p.
- Gerade Anzahl an Einsen im Datenwort: $p = 0$.
- Ungerade Anzahl an Einsen im Datenwort: $p = 1$.
- Realisierung durch XOR-Gatter

13.2 Block Parity

„Zweidimensionale“ Erweiterung der Paritätsprüfsumme
Beispielblock:



Vorteile:
- Erkennung und Korrektur von einzelnen Bitfehlern.
- Erkennung von zusammenhängenden Mehrfachfehlern (Bursts).

13.3 Realisierung der Block Parity Schaltung

Datenpfad:
Das n-Bit Datenwort wird durch das Paritybit zu (n+1)-Bit erweitert und übertragen
Kontrollpfad:
- Die Datenworte (Zeilen im obigen Beispielblock) aus dem Datenpfad werden bitweise verxodert und es entsteht die Pbits-Zeile
- Nach einem Block wird die Pbit-Zeile übertragen