Rechnerstrukturen

2. Grundlagen

Inhalt

- Elektronische Schalter
- Elementare Gatterfunktionen
- Schaltnetze
- Schaltwerke

2.2

Motivation

- Unterscheidung von zwei Zuständen
  - Strom / kein Strom
  - Spannung / keine Spannung
  - Positiv / Negativ geladen
  - Reflektierend / nicht reflektierend
  - ...
- Technische Umsetzung
  - Mechanisch
  - Elektromechanisch
  - Elektronisch
  - Licht
- Abbildung auf
  - binäre Zahlen
  - Wahrheitswerte
  - Zeichen

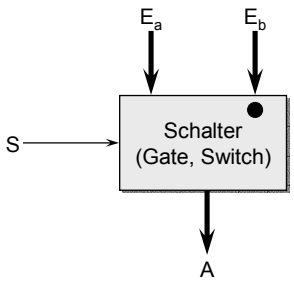
2.3

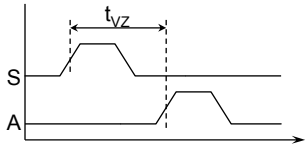
Elektronische Schalter

- Elementar Wechselschalter

```
if S then
   A := Ea
else
   A := Eb
```

- Vereinfachungen
  - Ein Eingang konstant 0 oder 1
- Verzögerungszeit t_{VZ}
  - wird durch die eingesetzte Technologie bestimmt
  - wenige Nanosekunden typisch

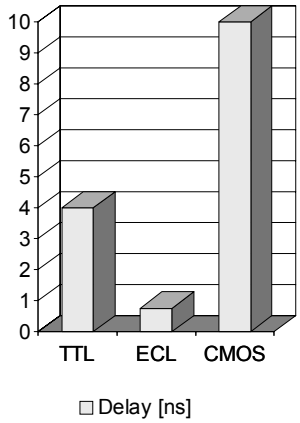




2.4

Realisierungsvarianten

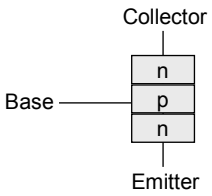
- Transistor-Transistor-Logik (TTL)
- Emitter-Coupled-Logik (ECL)
- Metal Oxid Semiconductor (MOS)
- Unterschiede
  - Verzögerungszeit
  - Versorgungsspannung
    - 5 V (TTL)
    - 2.8 - 3.3 V (MOS)
  - Integrationsdichte
    - 0.13μm - 0.18μm (MOS)
    - 0.08μm demnächst (MOS) (vgl. Haar = 100μm)
- Materialien
  - Silicium, Aluminium
  - in Zukunft ev. GaAs, Kupfer

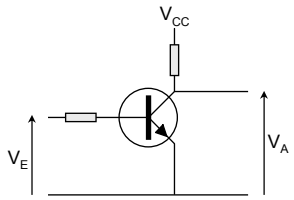


2.5

Bipolare Transistoren

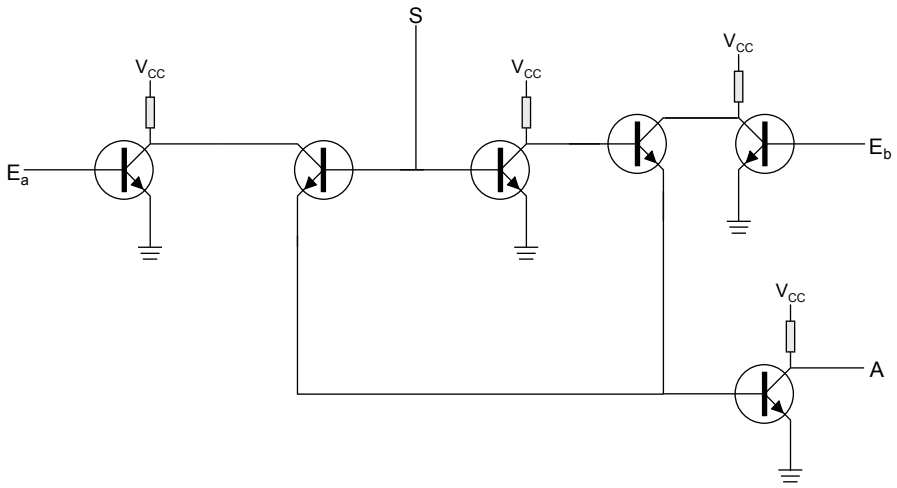
- npn- und pnp-Transistor
  - Dotierung von Silicium
- Schaltverhalten
  - V_E = 0 V, V_A = 3 V
  - V_E = 3 V, V_A = 0.2 V
- Hohe Ströme
  - p-Kanal muß mit Elektronen gefüllt werden
  - Wärmeentwicklung
  - Dicke der Leiterbahnen
- Geringe Kapazitäten
- Hohe Schaltgeschwindigkeit





2.6

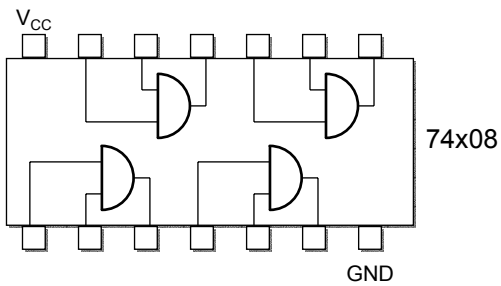
Wechselschalter: TTL-Realisierung



2.7

TTL-Familien

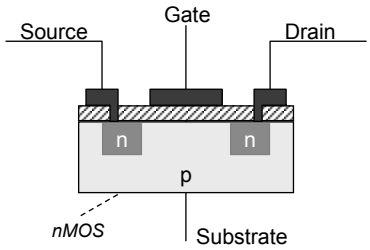
74xx	Standard TTL	30 mW/Gate	9 ns t_{VZ}
74Lxx	Low-power TTL	1	33
74Hxx	High-speed TTL	22	6
74Sxx	Schottky TTL	20	3
74LSxx	Low-power Schottky TTL	2	9
74ASxx	Advanced Schottky TTL	20	1.6
74ALSxx	Advanced Low-power Schottky TTL	1.3	5
74Fxx	Fast TTL	5	3

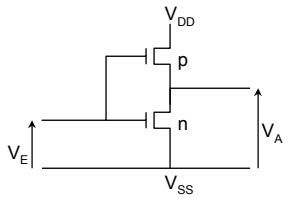


2.8

Feldeffekttransistoren

- FETs
  - NMOS (n-Kanal)
  - PMOS (p-Kanal)
  - CMOS (complementary MOS)
    - NMOS und PMOS paarweise
- Schaltverhalten
  - V_E = 0V, VA = V_{DD}
  - V_E = V_{DD} , VA = 0V
- Längere Verzögerungszeit
  - Kapazitives Element
- Geringe Spannungswerte
- Extrem geringe Ströme





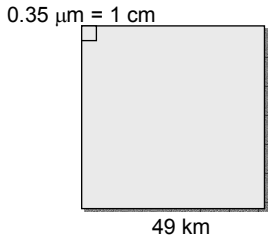
2.9

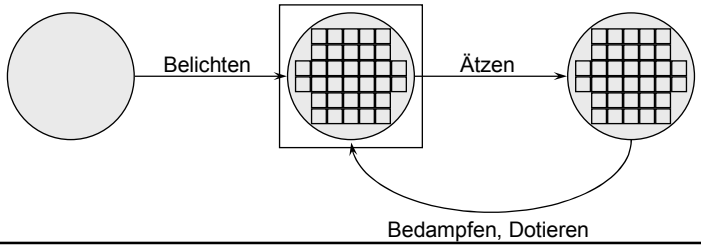
Wechselschalter: CMOS-Realisierung

2.10

State of the Art

- z.B. DEC Alpha 21264
  - 0.35μm CMOS-Technik
  - 3.02 cm^2
  - 15200000 Transistoren
  - 600 MHz Taktfrequenz
  - 72 W Stromverbrauch

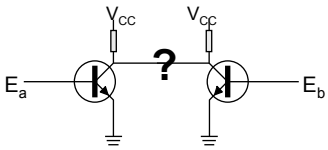


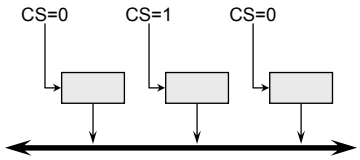


2.11

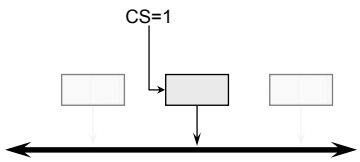
Tri-State

- Direkte Verschaltung von Ausgängen kritisch





- Zugang mehrerer Elemente zu einem gemeinsamen Bus
- Dritter, hochohmiger Zustand der Ausgänge
  - Steuerung durch zusätzlichen Eingang (CS = Chip Select)

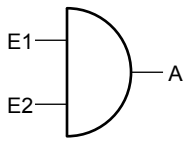


2.12

Die Grundgatter

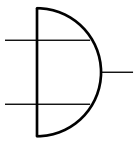
- UND

E1	E2	A
0	0	0
0	1	0
1	0	0
1	1	1



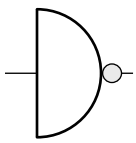
- ODER

E1	E2	A
0	0	0
0	1	1
1	0	1
1	1	1



- NICHT

E1	A
0	1
1	0

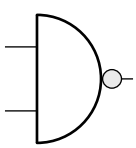


2.13

NAND, NOR, XOR

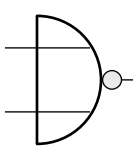
- NAND

E1	E2	A
0	0	1
0	1	1
1	0	1
1	1	0



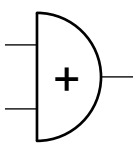
- NOR

E1	E2	A
0	0	1
0	1	0
1	0	0
1	1	0



- XOR

E1	E2	A
0	0	0
0	1	1
1	0	1
1	1	0



2.14

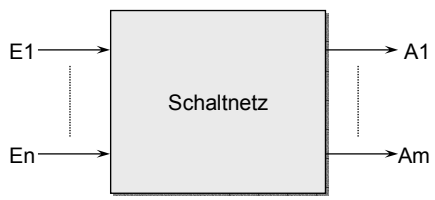
Äquivalenz

- Ist XOR ein Grundgatter?
- Wieviele Grundgatter braucht man minimal, um beliebige boolesche Ausdrücke zu beschreiben?

2.15

Schaltnetze

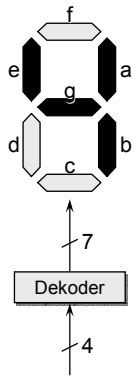
- Schaltung aus logischen Grundgattern mit
  - n Eingängen
  - m Ausgängen
  - Rückkopplungsfrei
- 2n Eingangskombinationen
- m boolesche Ausdrücke über E1 ... En
- Wahrheitstabelle



2.17

Wahrheitstabelle 7-Segment-Dekoder

- 4 Eingänge
  - Binärzahl $e_3\ e_2\ e_1\ e_0$
- 7 Ausgänge
  - Austeuerung der Segment a bis g



e_3	e_2	e_1	e_0	a	b	c	d	e	f	g
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0							
1	1	1	1							

2.18

7-Segment-Dekoder (cont.)

- Wie erhält man boolesche Ausdrücke für a bis g?

$$a(e_3, e_2, e_1, e_0) =$$
$$b(e_3, e_2, e_1, e_0) =$$
$$\vdots$$
$$g(e_3, e_2, e_1, e_0) =$$

2.20

7-Segment-Dekoder (cont.)

- Gatterschaltung:

2.22

Disjunktive und konjunktive Normalformen

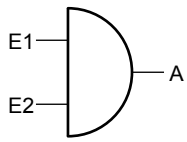
- Disjunktive Normalform
  - Summe von Produkten (minterme)
- Minterm
  - 1 Zeile in Wahrheitstabelle
  - Eingänge mit 1: e
  - Eingänge mit 0: e negiert
- Konjunktive Normalform
  - Produkt von Summen (maxterme)
- Maxterm
  - 0 Zeile in Wahrheitstabelle
  - Eingänge mit 1: e negiert
  - Eingänge mit 0: e

e2	e1	e0	a
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

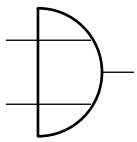
2.24

Boolesche Algebra

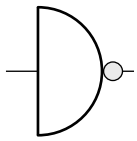
- Dualität zwischen
  - Gatterschaltungen
  - Booleschen Ausdrücken
- Positive Logik
  - 0 = False
  - 1 = True
- Gesetze



$$A = E1 \cdot E2$$



$$A = E1 + E2$$



$$A = \overline{E}$$

2.26

Gesetze

- Operationen mit 0 und 1

  $X + 0 = X, X + 1 = 1$

  $X \cdot 0 = 0, X \cdot 1 = X$

- Idempotenz

  $X + X = X$

  $X \cdot X = X$

- Komplementärgesetz

  $X + \overline{X} = 1$

  $X \cdot \overline{X} = 0$

2.27

Gesetze (cont.)

- Kommutativitätsgesetz

$$X + Y = Y + X$$
$$X \cdot Y = Y \cdot X$$

- Assoziativitätsgesetz

$$(X + Y) + Z = X + (Y + Z)$$
$$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$$

- Distributivgesetz

$$X \cdot (Y + Z) = X \cdot Y + X \cdot Z$$
$$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$$

2.28

Gesetze (cont.)

- Vereinfachungsgesetze

$$X \cdot Y + X \cdot \overline{Y} = X, (X + Y) \cdot (X + \overline{Y}) = X$$
$$X + X \cdot Y = X, X \cdot (X + Y) = X$$
$$(X + \overline{Y}) \cdot Y = X \cdot Y, (X \cdot \overline{Y}) + Y = X + Y$$

- DeMorgan's Gesetz

$$\overline{X + Y + \ldots + Z} = \overline{X} \cdot \overline{Y} \cdot \ldots \cdot \overline{Z}$$
$$\overline{X \cdot Y \cdot \ldots \cdot Z} = \overline{X} + \overline{Y} + \ldots + \overline{Z}$$

2.29

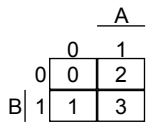
Umformung und Minimierung

- Gründe
  - Begrenzungen bei der Schaltungstiefe
  - Minimaler Materialeinsatz
  - Bestimmter Gattervorrat
  - Elektrische Eigenschaften
  - Platzbeschränkungen
  - ...
- Algebraische Umformungen
- Algorithmische Verfahren
  - Karnaugh-Diagramme (1-4 Eingangsvariablen, 1 Ausgang)
  - Quine-McCluskey-Verfahren (n Eingänge, 1 Ausgang)
  - Bündelminimierung (n Eingänge, m Ausgänge)

2.30

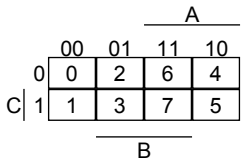
Karnaugh-Diagramme

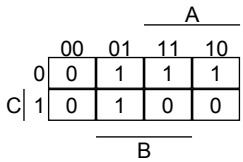
- Graphische Methode
- max. 4 Eingänge ABCD
- Übertragung der Wahrheitstabelle
  - Position im Diagramm entspricht ABCD als Binärzahl



A	B	C	a
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

= (2,3,4,6) =

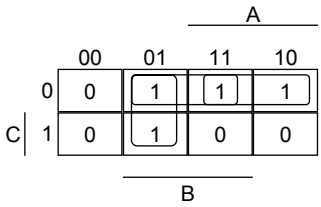




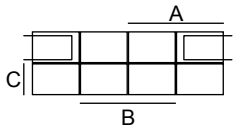
2.31

Minimierung im Karnaugh-Diagramm

- Legende = Gray-Code
  - benachbarte Zeilen und Spalten ändern sich nur in einem Bit
- Zusammenfassen von Gruppen zu 1, 2, 4 oder 8 Einsen
  - 1 = keine Minimierung
  - 2 = Term mit 2 aus 3 Eingängen
  - 4 = Term mit 1 aus 3 Eingängen
  - 8 = Funktion konstant 1
- Diagramm als Torus auffassen!



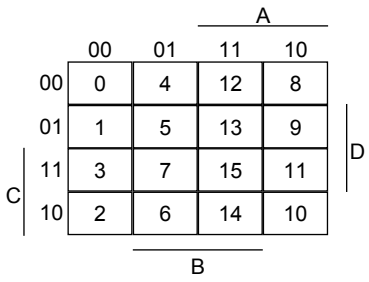
$$= \overline{A}B + B\overline{C} + A\overline{C}$$



2.32

4 Eingänge

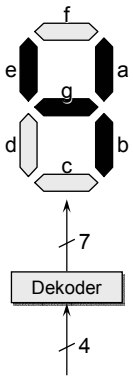
- Zusammenfassen von Gruppen zu 1, 2, 4, 8 oder 16 Einsen
  - 1 = keine Minimierung
  - 2 = Term mit 3 aus 4 Eingängen
  - 4 = Term mit 2 aus 4 Eingängen
  - 8 = Term mit 1 aus 4 Eingängen
  - 16 = Funktion konstant 1
- Torus



2.33

Minimierung 7-Segment-Dekoder

- 4 Eingänge
  - Binärzahl e3 e2 e1 e0
- 7 Ausgänge
  - Austeuerung der Segment a bis g

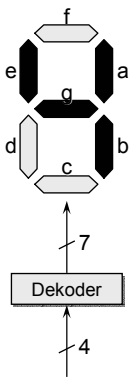


e3	e2	e1	e0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	0	1
0	1	0	1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	0	1	1	1	1
1	0	1	1	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1	1	1	0
1	1	0	1	1	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1

2.34

Minimierung 7-Segment-Dekoder (Don´t Care)

- 4 Eingänge
  - Binärzahl e3 e2 e1 e0
- 7 Ausgänge
  - Austeuerung der Segment a bis g



e3	e2	e1	e0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	0	1
0	1	0	1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

2.39

Bemerkungen

- Bei bis zu vier Eingangsvariablen ideale Minimierungstechnik
- 5 und 6 Eingänge theoretisch auch möglich
  - 5: zwei übereinander liegende 4er-Diagramme
  - 6: vier übereinander liegende 4er-Diagramme
  - insgesamt aber praktisch nicht handhabbar
- Konjunktive Minimalform ebenfalls möglich
  - Zusammenfassen der 0-Gruppen

2.44

Quine-McCluskey-Methode

- Algorithmisches Verfahren
  - beliebig viele Eingänge
  - 1 Ausgang
- 3 Schritte
  - Initialisierung
  - Ermittlung der Primimplikanten
  - Ermittlung der minimalen Überdeckung
- Beispielfunktion
  - Segment a

e3	e2	e1	e0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	0	1
0	1	0	1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	0	1	1	1	1
1	0	1	1	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1	1	1	0
1	1	0	1	1	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1

2.45

1. Initialisierung

- Sortierung der Minterme aufsteigend nach Anzahl 1

```
0000  (0)

0001  (1)
0010  (2)
0100  (4)
1000  (8)

0011  (3)
1001  (9)
1010  (10)

0111  (7)
1101  (13)
```

2.46

2. Primimplikanten

- Vergleich jedes Element einer Gruppe mit allen Elementen der nächsten Gruppe
- Übernahme in die nächste Spalte, wenn nur in einer Position verschieden
- Gewählte Zeilen markieren (ok)

```
0000  (0) ok      000-  (0,1)
                  00-0  (0,2)
0001  (1) ok      0-00  (0,4)
0010  (2) ok      -000  (0,8)
0100  (4) ok
1000  (8) ok      00-1  (1,3)
                  -001  (1,9)
0011  (3) ok      001-  (2,3)
1001  (9) ok      -010  (2,10)
1010  (10) ok     100-  (8,9)
                  10-0  (8,10)
0111  (7) ok
1101  (13) ok     0-11  (3,7)
                  1-01  (9,13)
```

2.47

Abbruchkriterium

- Markierung mit der nächsten Spalte wiederholen
- Bis auf eine Position gleich (- = -)
- Abbruch, wenn keine weitere Spalte entsteht

```
000-   (0,1) ok      00--   (0,1,2,3) *
00-0   (0,2) ok      -00-   (0,1,8,9) *
0-00   (0,4) *       -0-0   (0,2,8,10) *
-000   (0,8) ok

00-1   (1,3) ok
-001   (1,9) ok
001-   (2,3) ok
-010   (2,10) ok
100-   (8,9) ok
10-0   (8,10) ok

0-11   (3,7) *
1-01   (9,13) *
```

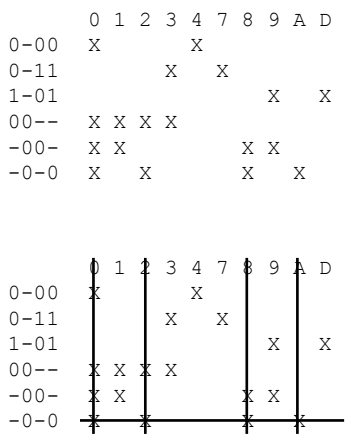
2.48

3. Minimale Überdeckung

- Sammeln der mit * markierten Primimplikanten:

```
0-00   (0,4)
0-11   (3,7)
1-01   (9,13)
00--   (0,1,2,3)
-00-   (0,1,8,9)
-0-0   (0,2,8,10)
```

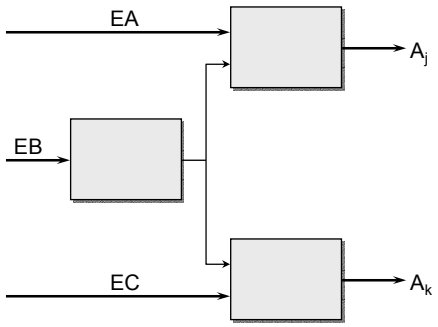
- Nur einmal markierte Spalten suchen
  - essentielle Primimplikanten
  - Zusätzliche Spalten streichen
- Schritt auf unmarkierten Spalten wiederholen



2.49

Bündelminimierung

- Anwendbar auf n Eingänge und m Ausgänge
- Nutzung gemeinsamer Teilsummen in verschiedenen Ausgangsfunktionen
- Multilevel-Logik
- Ansatz
  - Erweiterung der Zeileninformation
  - ansonsten Quine-McCluskey




```
00-01-0      (2,3,7,9)    110...0
```

A_m A_1

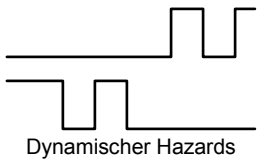
2.50

Hazards

- Ungewollte Wechsel an einem Ausgang
  - Unterschiedliche Verzögerungszeiten der Gatter
  - Unterschiedliche Gatteranzahl zwischen Eingängen und Ausgang
- Statischer Hazard
  - Einmaliger Wechsel
- Dynamischer Hazard
  - Mehrfacher Wechsel


Statischer 1-Hazard


Statischer 0-Hazard


Dynamischer Hazards

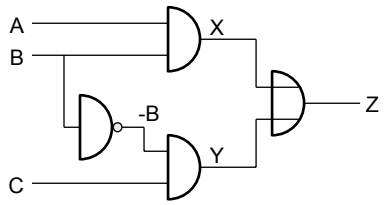
2.51

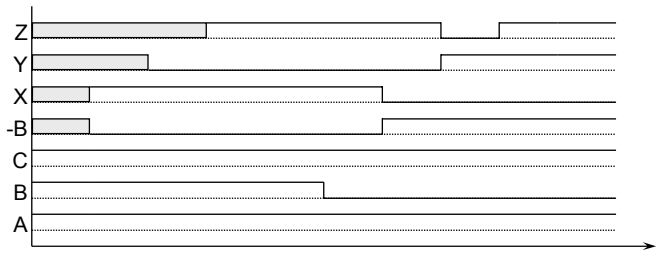
Statischer Hazard

- Nur 1-Bit-Wechsel
- Beispiel

$$Z = AB + \overline{B}C$$

  - Wechsel 111 nach 011 (ABC)



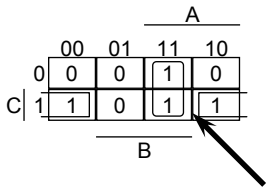


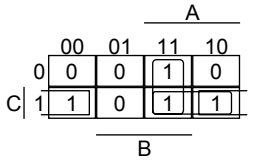
2.52

Eliminierung statischer Hazards

- Grundlage Karnaugh-Diagramm
- Merkmal für 1-Hazard
  - Wechsel des Primimplikanten bei 1-Bitwechsel der Eingabe
  - Lösung: Redundante Implikanten
- Merkmal für 0-Hazard
  - Karnaugh-Diagramm für konjunktive Normalform
  - Analog 1-Hazard
- Eliminierung statischer Hazards auch in mehrstufigen Schaltungen möglich

$$Z = AB + \overline{B}C$$





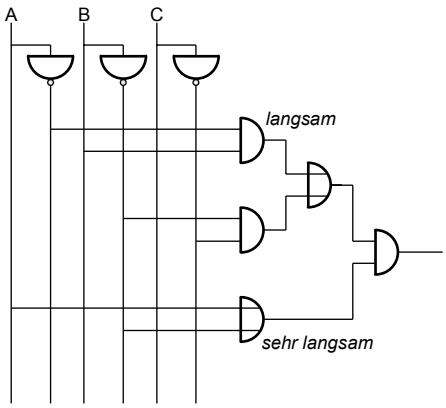
2.53

Dynamischer Hazard

- Beispiel

$$(\overline{A}B + \overline{B}\overline{C})(A + \overline{B})$$

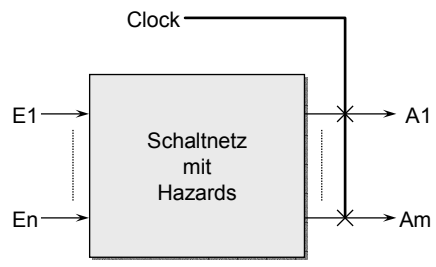
  - Wechsel 000 nach 010 (ABC)
- Grund
  - Unterschiedlich lange Wege von einem Eingang zu einem Ausgang
- Eliminierung schwierig

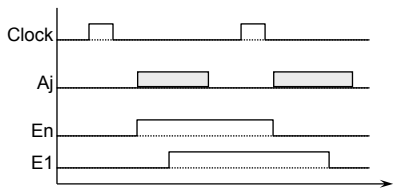


2.54

Bemerkungen

- Beschränkte Möglichkeiten
  - Nur 1-Bit-Wechsel
  - Multilevel-Logik schwierig
- Schaltungen frei von statischen und dynamischen Hazards
  - 2-stufige Logik
  - Fan-In/Fan-Out-Problem
- Gängiger Ansatz
  - Auferzwungene Taktung
  - Periode länger als maximaler Hazard

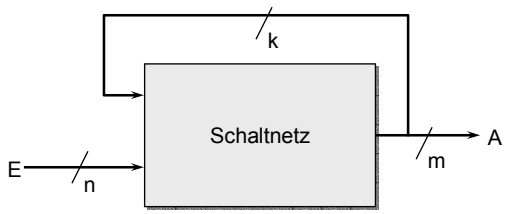




2.55

Schaltwerke

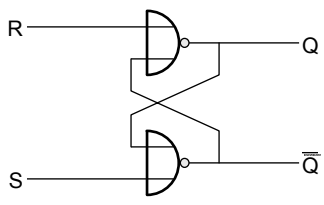
- Kombinatorische Schaltnetze
  - Ausgang hängt nur von den Eingängen ab
  - Unterschiedliche Laufzeiten (Hazard-Problematik)
- Schaltwerk
  - Ausgang hängt von den Eingängen und den vorherigen Ausgaben ab
- Aspekte
  - Speichern möglich
  - Synchron / Asynchron
  - Schwingungen
  - Metastabilität
  - Selbsttaktung
- Elementarbausteine
  - Latch
  - Flip-Flop



2.56

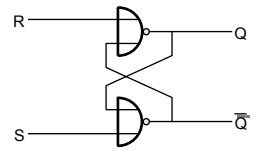
Elementarspeicher: R-S-Latch

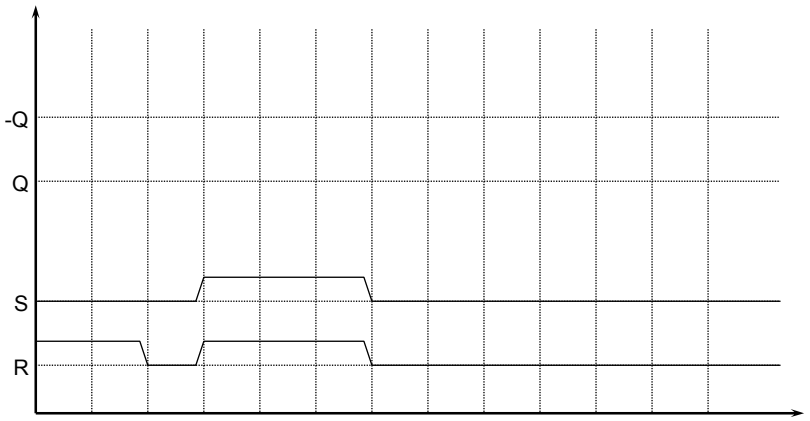
- Kreuzverschaltete NOR-Gatter
  - R: Reset
  - S: Set
  - Ausgang Q mit Komplement
- Ausgang zurücksetzen
  - R=1, S=0
- Ausgang setzen
  - R=0, S=1
- Wert speichern
  - R=0, S=0



2.57

R-S-Latch: Zeitdiagramm

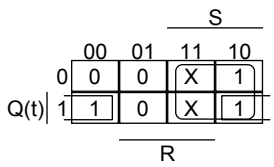




2.58

R-S-Latch: Charakeristische Gleichung

S(t)	R(t)	Q(t)	Q(t+Δ)	
0	0	0	0	Hold
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	X	Invalid
1	1	1	X	



$$Q(t+\Delta) = S(t) + \overline{R}(t) \cdot Q(t)$$

2.60

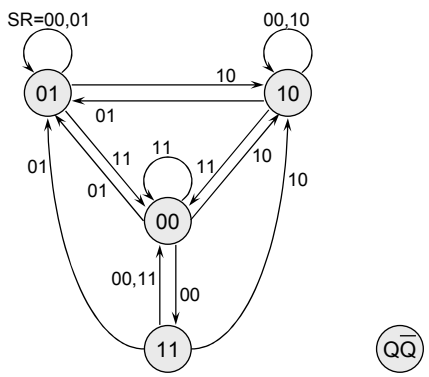
Probleme

- Verbotene Eingangskombination
  - R=S=1
  - Beide Ausgänge sind 0
- Verbotener Übergang
  - R und S gleichzeitig von 1 nach 0
  - Oszillation der Ausgänge
- Wie lang kann die Oszillation dauern?
  - Theoretisch?
  - Praktisch?
- Race-Condition

S	R	Q
0	0	Speichern
0	1	0
1	0	1
1	1	Instabil

2.61

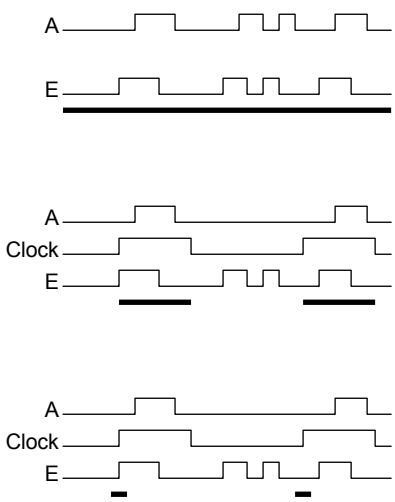
Theoretische und beobachtbare Zustände



2.62

Steuerungsarten

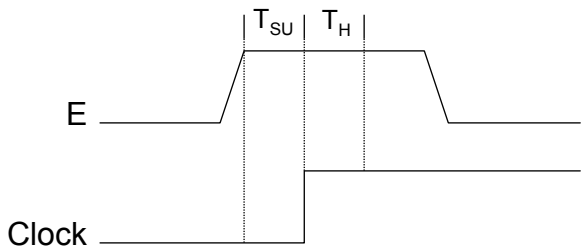
- Wann wirken die Eingänge auf die Ausgänge
- Drei Varianten
  - Ungesteuert (R-S-Latch)
  - Pegelgesteuert
  - Flankengesteuert
    - Positiv (0 nach 1)
    - Negativ (1 nach 0)
- Zusätzliche Steuerungsleitung
  - Enable
  - Clock



2.63

Flankensteuerung

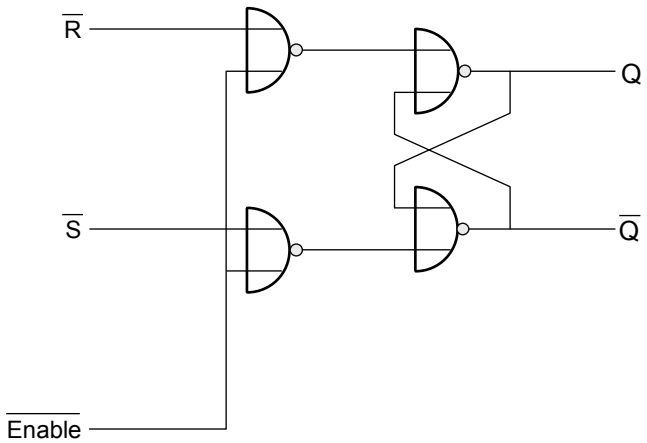
- Stabiler Eingang in einem Zeitfenster vor dem Flankenwechsel
  - Setup-Zeit: T_{SU}
  - Hold-Zeit: T_H
- Verhalten ansonst undefiniert
- Typische Werte (TTL)
  - T_{SU} = 20ns
  - T_H = 5ns



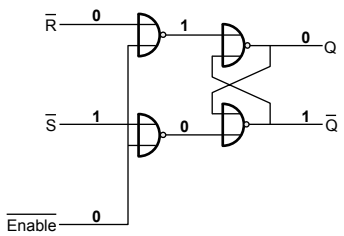
2.64

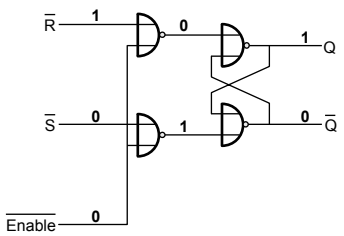
Pegelgesteuertes R-S-Latch

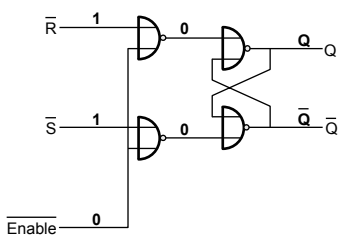
- Steuereingang Enable

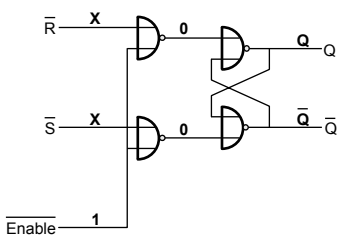


2.65







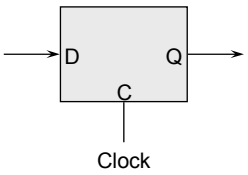


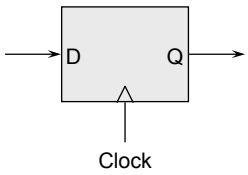
2.66

Latch vs. Flip-Flop

- Latch
  - Ungesteuert
  - Pegelgesteuert
- Änderung der Ausgänge bei Änderung der Eingänge

- Flip-Flop
  - Positiv flankengesteuert
  - Negativ flankengesteuert
  - Master/Slave
- Änderung der Ausgänge wird durch den Steuerungseingang getriggert

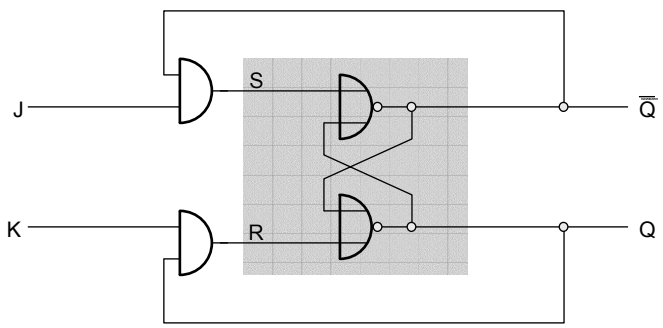




2.67

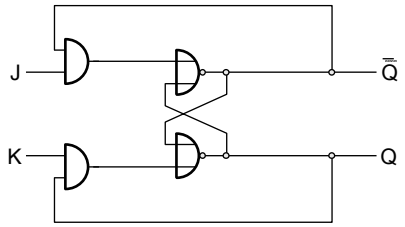
JK-Latch

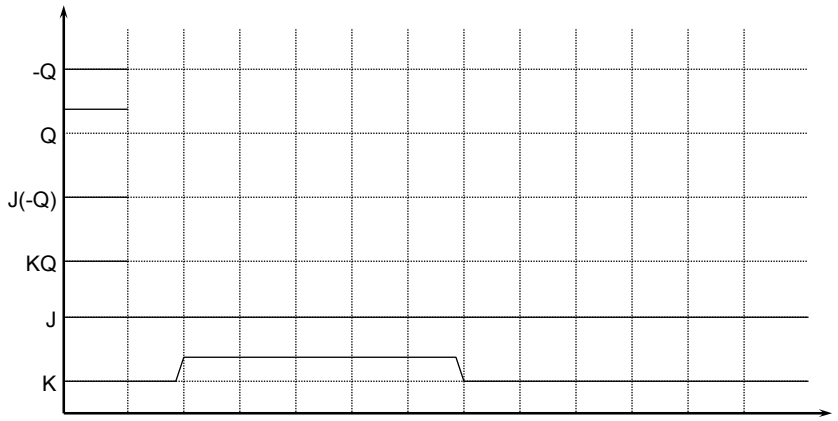
- Erweiterung eines R-S-Latch
  - Ungültige Eingabe R=S=1 verhindern
- Was passiert bei J=K=1?



2.68

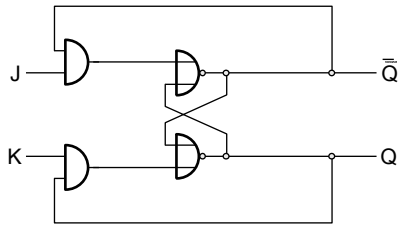
Zeitdiagramm: K=1, Q=1

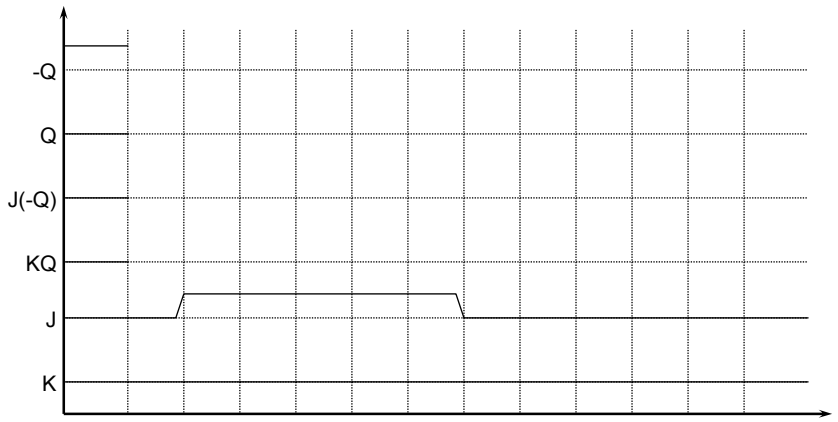




2.69

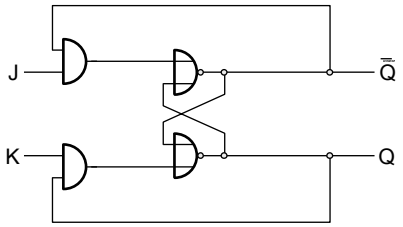
Zeitdiagramm: J=1, Q=0

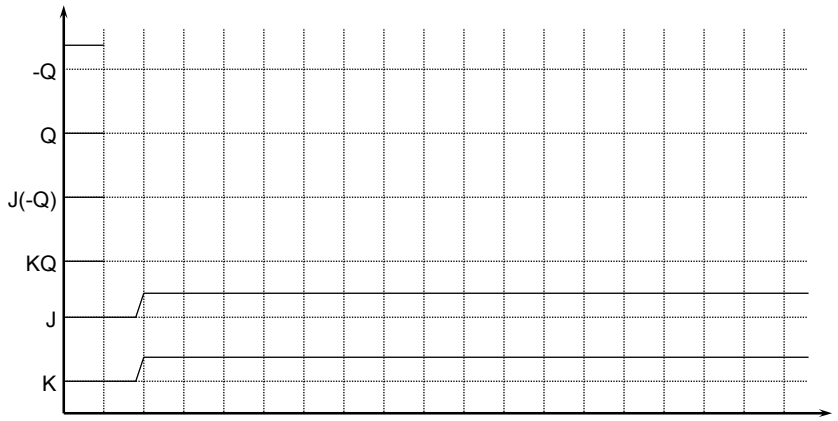




2.71

Zeitdiagramm: J=K=1, Q=0

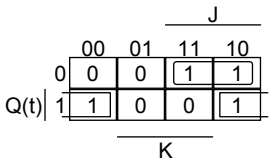




2.73

J-K-Latch: Charakeristische Gleichung

J(t)	K(t)	Q(t)	Q(t+Δ)	
0	0	0	0	Hold
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	1	Toggle
1	1	1	0	

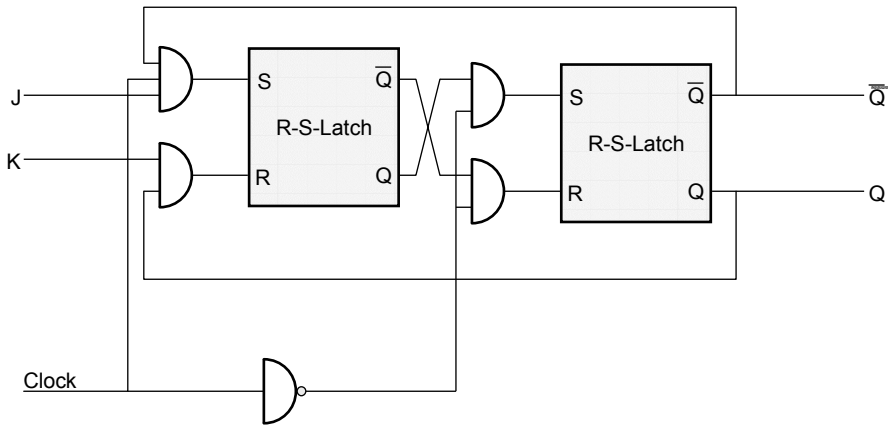


$$Q(t+\Delta) = J(t) \cdot \overline{Q}(t) + K(t) \cdot Q(t)$$

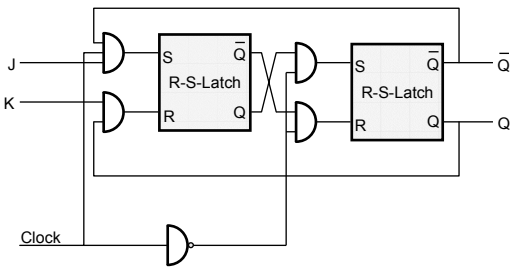
2.75

JK-Master/Slave-Flip-Flop

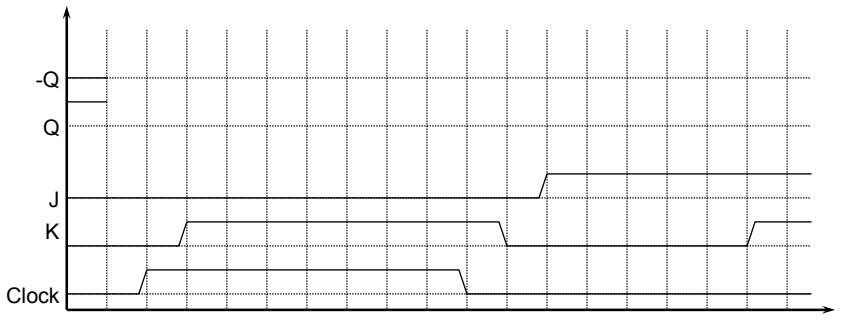
- Dauer-Toggle verhindern



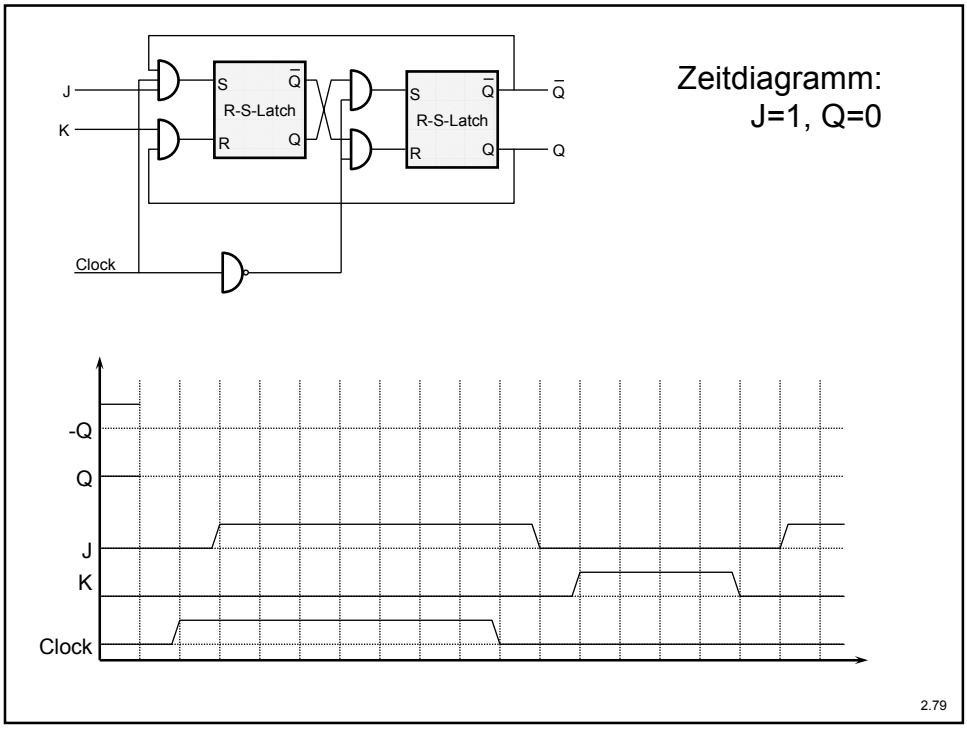
2.76



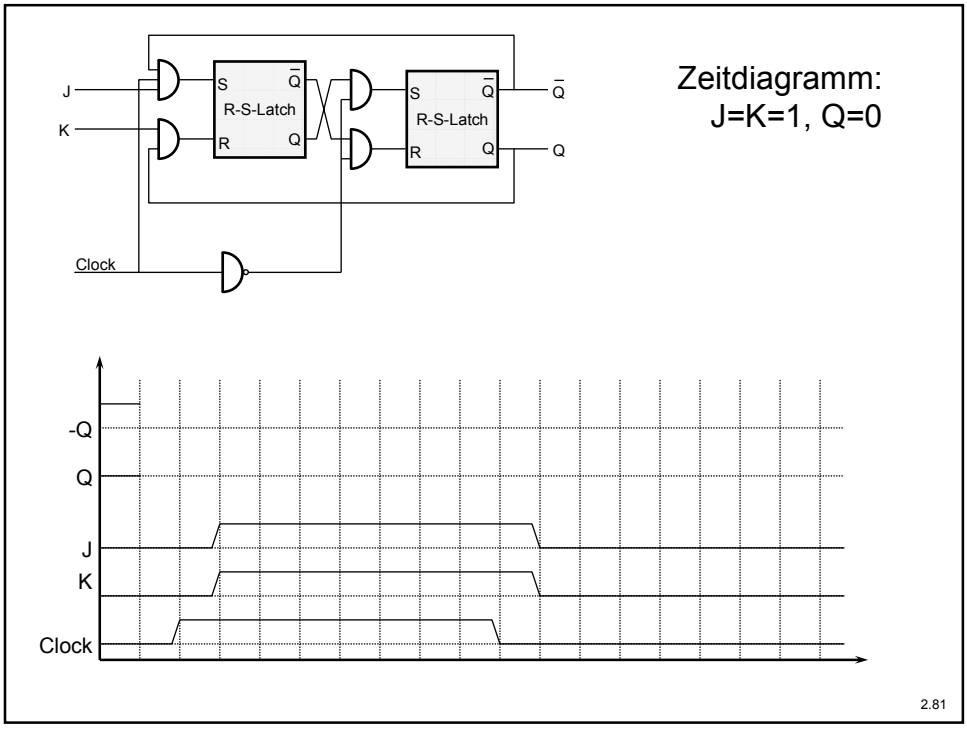
Zeitdiagramm:
K=1, Q=1



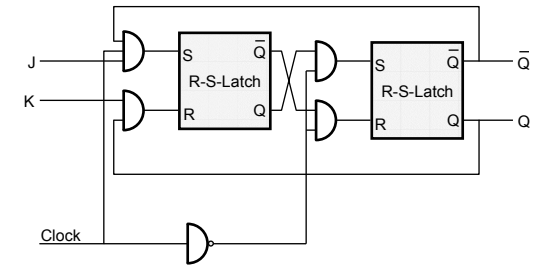
2.77



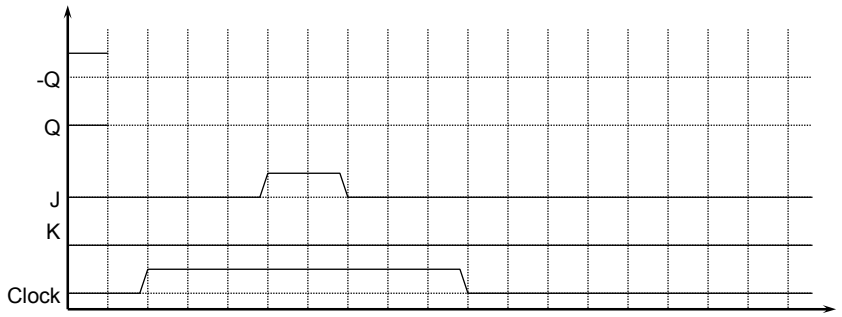
Zeitdiagramm:
J=1, Q=0



Zeitdiagramm:
J=K=1, Q=0



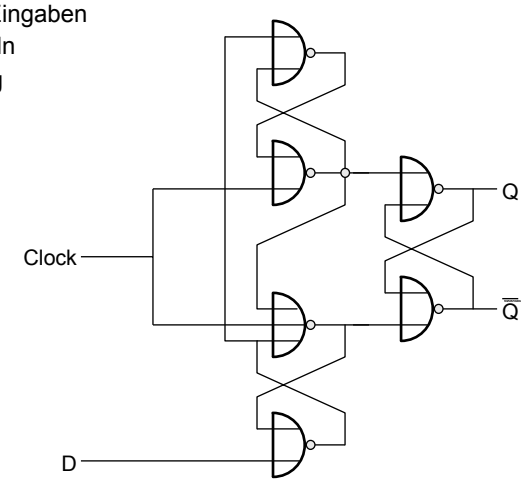
Zeitdiagramm: One Catching



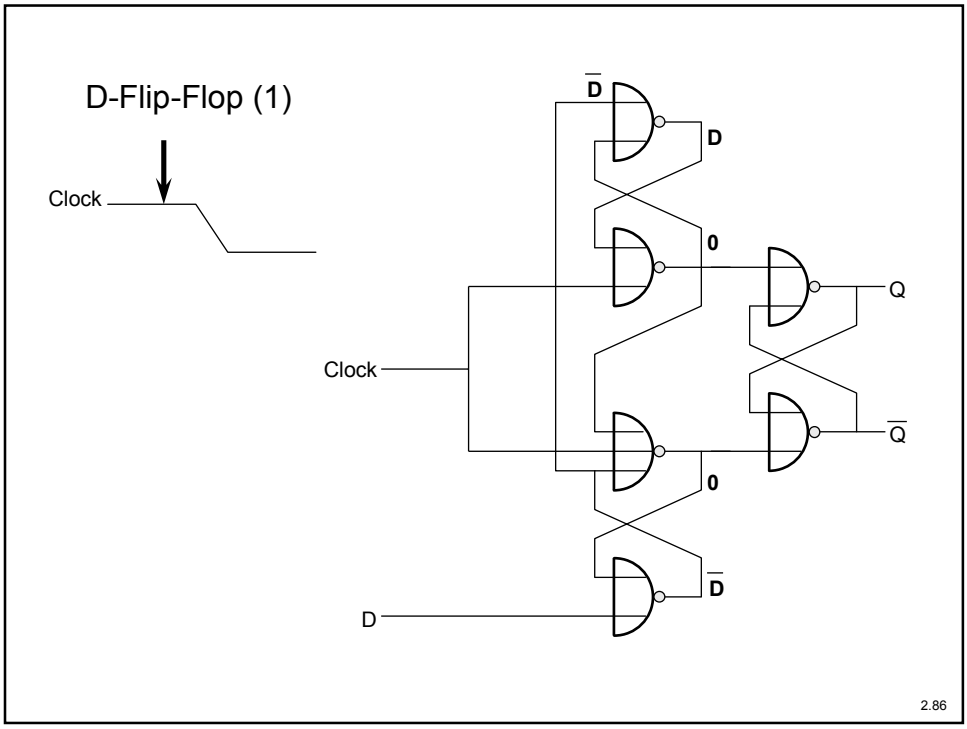
2.83

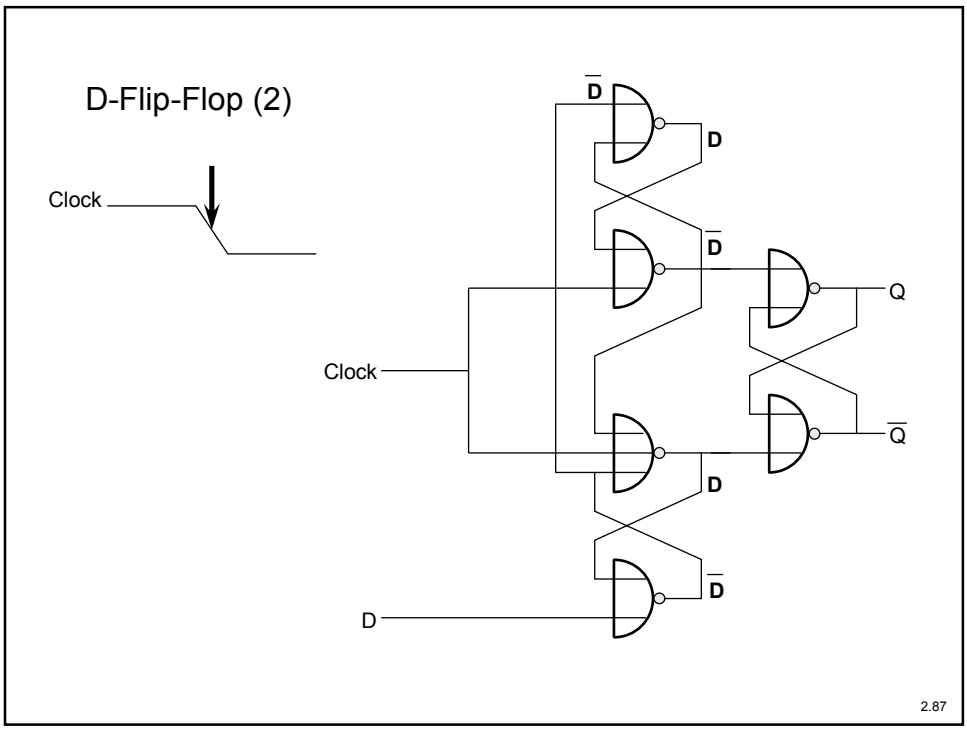
D-Flip-Flop

- Reine Flankensteuerung
  - Keine ungültigen Eingaben
  - Kein Dauer-Toggeln
  - Kein One-Catching
- Beispiel D-Flip-Flop
  - Negativ Flankengesteuert

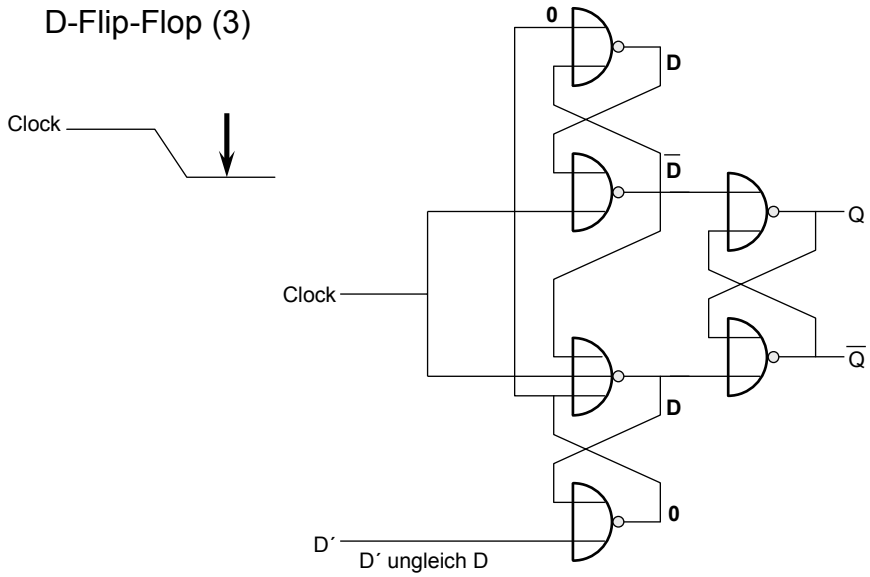


2.85


D-Flip-Flop (1)
Clock
$\overline{D}$
D
0
Q
Clock
$\overline{Q}$
0
$\overline{D}$
D
2.86

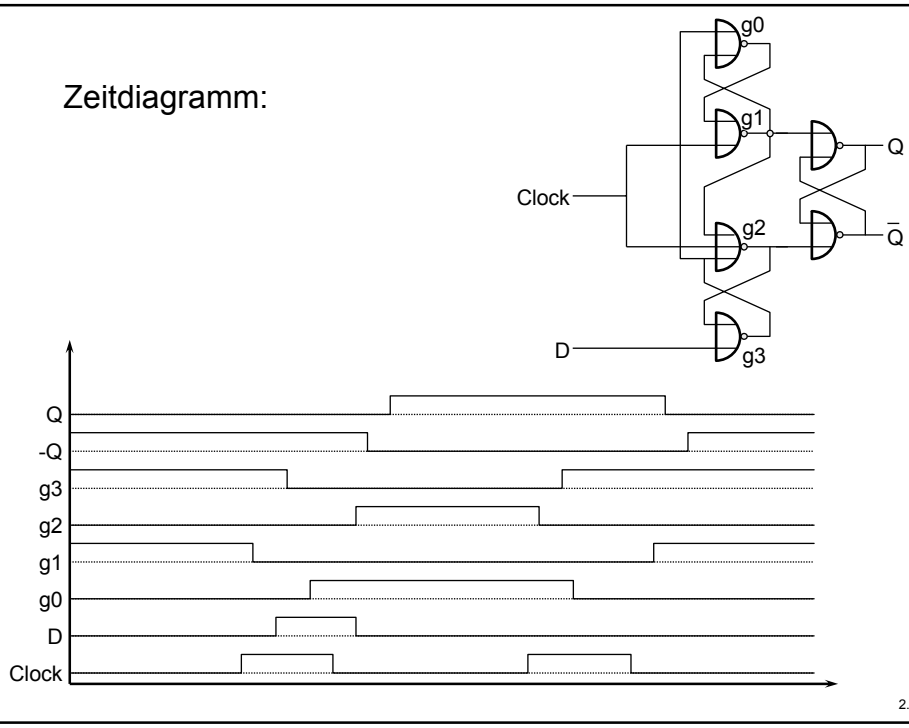

D-Flip-Flop (2)
Clock
$\overline{D}$
D
$\overline{D}$
Q
Clock
$\overline{Q}$
D
$\overline{D}$
D
2.87

D-Flip-Flop (3)



2.88

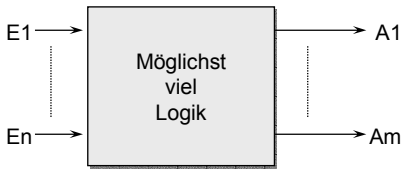
Zeitdiagramm:



2.89

Programmierbare Logik

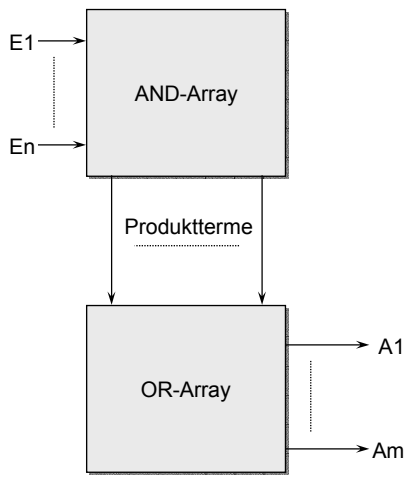
- Realisierung von Schaltnetzen und Schaltwerken
- Aufbau mit Hilfe von TTL- und CMOS-ICs aufwendig
  - Große Anzahl an Bausteinen
  - Hoher Platz- und Stromverbrauch
  - Geringe Integrationsdichte
- IC mit 1000 AND mit jeweils 2 Eingängen = 3002 Pins
- "Programmierbare Bausteine"
  - Genügend Eingängen
  - Genügend Ausgängen
  - Ausreichende Programmierbarkeit
- Zweistufige Normalformen



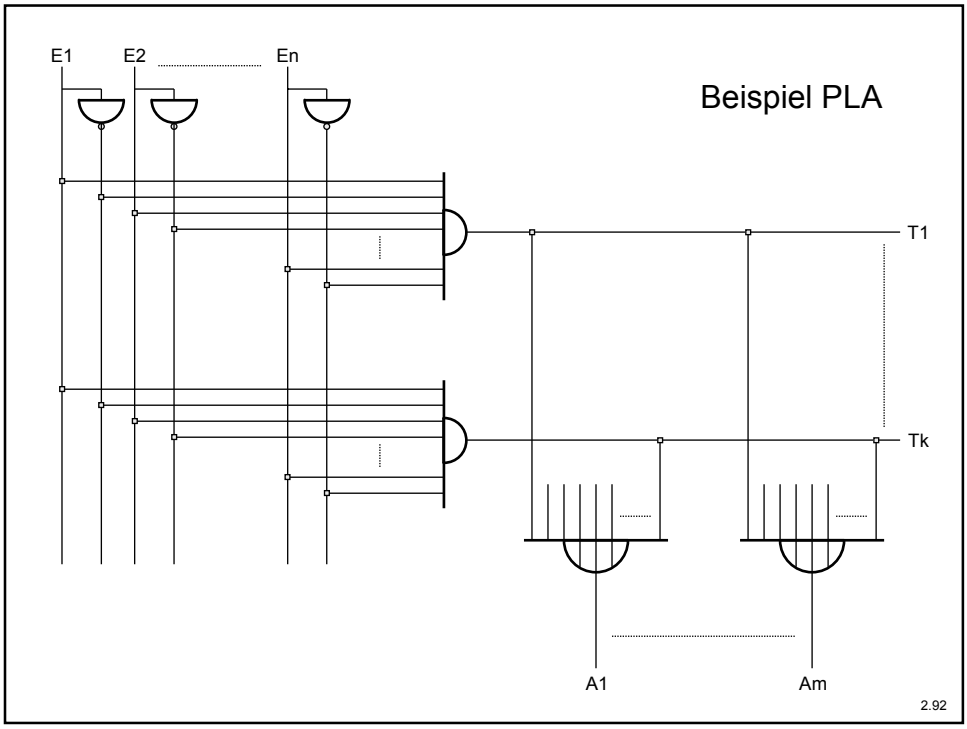
2.90

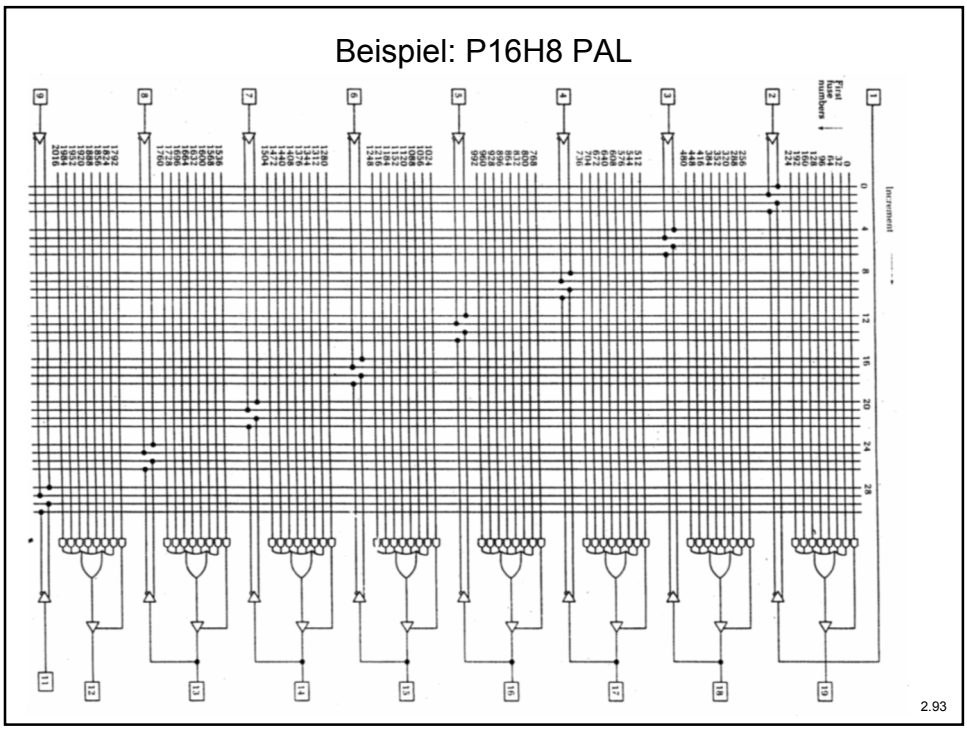
PLA und PAL

- Programmierbare disjunktive Normalform
  - n Eingängen
  - m Ausgängen
  - k Terme
- n,m und k vom jeweiligen Baustein abhängig
- PLA = Programmable Logic Array
  - UND- und ODER-Array programmierbar
- PAL = Programmable Array Logic
  - Nur UND-Array programmierbar
- Programmieren
  - Höhere Programmierspannung
  - "Durchbrennen" einer Verbindung

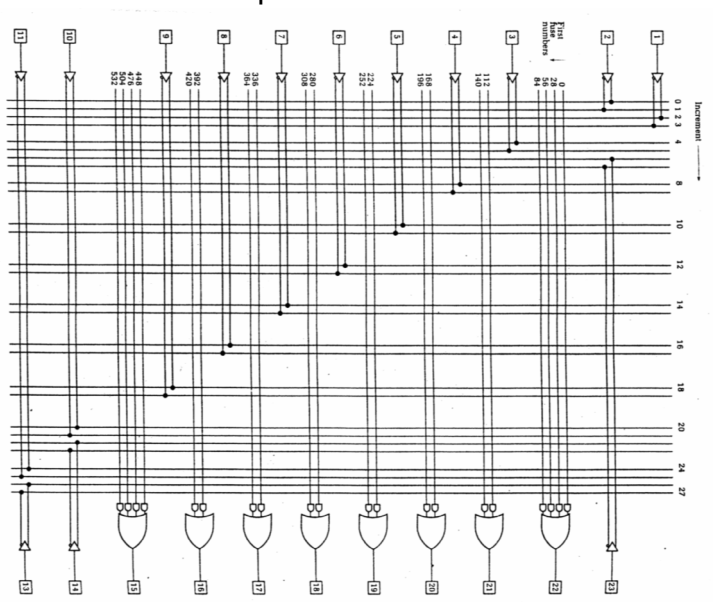


2.91


Beispiel PLA
E1
E2
En
T1
Tk
A1
Am
2.92

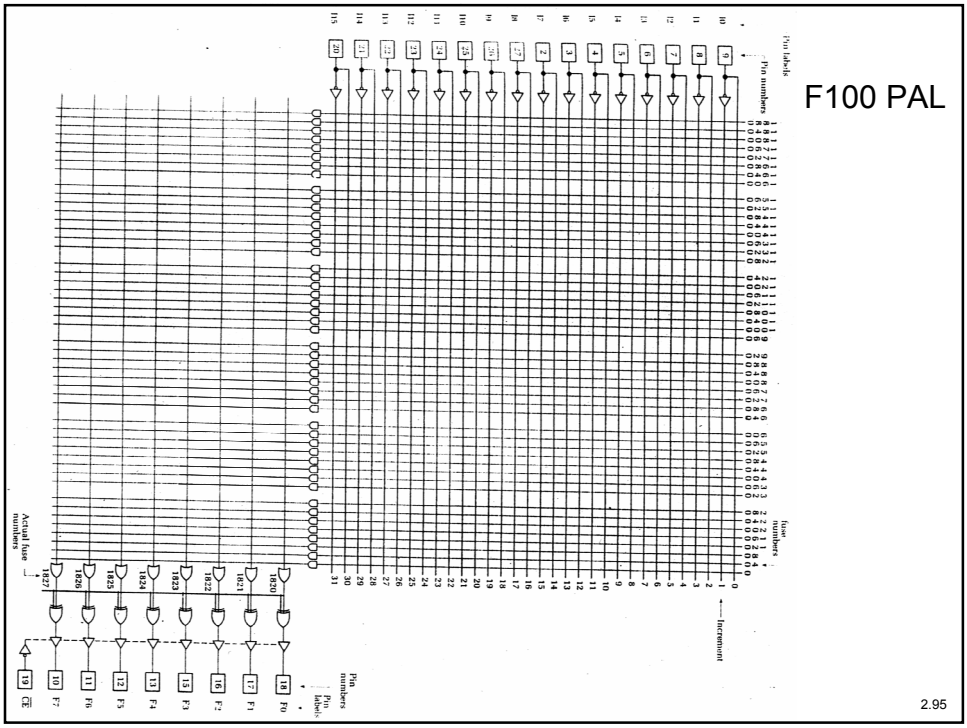

Beispiel: P16H8 PAL
2.93

Beispiel: P14H8 PAL



2.94

F100 PAL



2.95