

# Digitaltechnik & Rechnersysteme

## Timing & Speicher

Martin Kumm

**Hochschule Fulda**  
University of Applied Sciences



Angewandte Informatik

WiSe 2022/2023

# Was bisher geschah...

---



- Automatenentwurf
  - Schritte von der Spezifikation zur Schaltung
  - Erstellung des Zustandsdiagramms: Kreativer Prozess
  - Erstellung der (minimierten) Schaltung: Geradliniger Prozess (»Schema F«)
  - Unterschiede Moore/Mealy

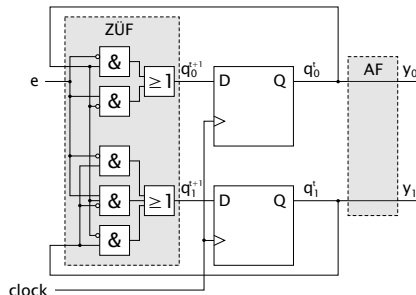
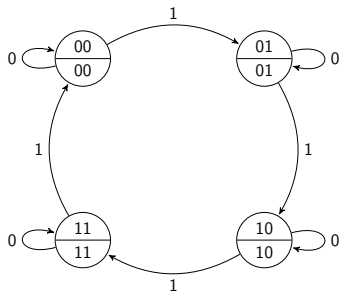
# Automatensynthese (wdh.)



Ablauf:

- 1 Festlegung der Ein- und Ausgangsvariablen
- 2 Festlegung ob Moore/Mealy-Automat
- 3 Bestimmung der Anzahl der Zustände
- 4 Ermittlung des Zustandsgraphen
- 5 Zustandskodierung
- 6 Bestimmung Zustandsübergangs- und Ausgangstabelle
- 7 Ermittlung (minimaler) Übergangs- und Ausgangsfunktionen
- 8 Erstellung des Schaltbildes aus Gattern und FFs

# Mod-4 Zähler mit enable (wdh.)



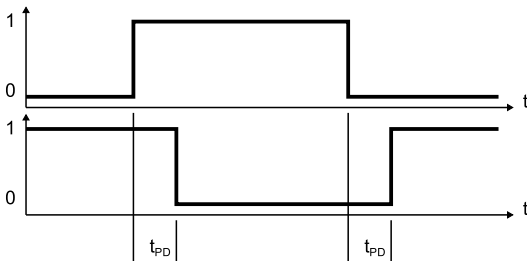
# Zeitverhalten: Gatterlaufzeit



Schaltglieder weisen unabhängig vom realisierten Operator ein charakteristisches zeitliches Verhalten auf.

Dabei erscheint die Reaktion des Ausgangssignals, bedingt durch Umladungsvorgänge innerhalb des Halbleiterbauelements und endlich schnelle Spannungsänderungen, mit einer **zeitlichen Verzögerung**  $t_{PD}$  (PD: *propagation delay*).

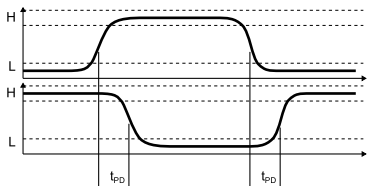
Beispiel Inverter:



# Zeitverhalten



Genauer: Realisierung durch Spannungen, z.B.:



Anstiegszeit und Abfallzeit klein(!) gegen Gatterlaufzeit.

Die Durchlaufzeiten der Spannungsänderungen werden über die 50%-Punkte auf den Flanken der Signale bestimmt

# Zeitverhalten eines Inverter (74HC04)



## 10. Dynamic characteristics

**Table 7. Dynamic characteristics**

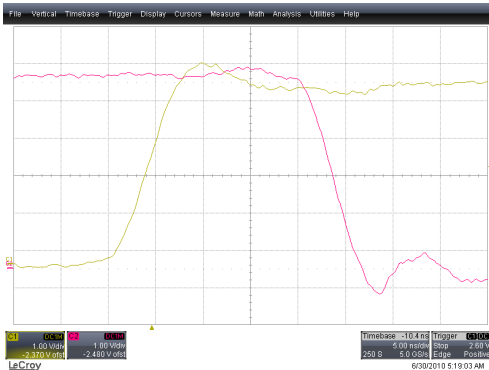
$GND = 0\text{ V}$ ;  $C_L = 50\text{ pF}$ ; for test circuit see [Figure 7](#).

Symbol	Parameter	Conditions	25 °C			−40 °C to +125 °C		Unit
			Min	Typ	Max	Max (85 °C)	Max (125 °C)	
74HC04								
t <sub>pd</sub>	propagation delay	nA to nY; see <a href="#">Figure 6</a> <a href="#">[1]</a>						
		V <sub>CC</sub> = 2.0 V	-	25	85	105	130	ns
		V <sub>CC</sub> = 4.5 V	-	9	17	21	26	ns
		V <sub>CC</sub> = 5.0 V; C <sub>L</sub> = 15 pF	-	7	-	-	-	ns
		V <sub>CC</sub> = 6.0 V	-	7	14	18	22	ns
t <sub>L</sub>	transition time	see <a href="#">Figure 6</a> <a href="#">[2]</a>						

Auszug aus »74HC04; 74HCT04 Hex inverter, Product data sheet, Rev. 5 — 27 November 2015«

# Messung der Gatterverzögerung

## Messung des Zeitverhaltens an einem Inverters (74HC04):





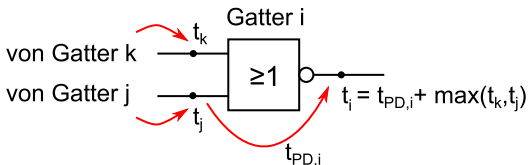
# Berechnung des kritischen Pfades



Der Zeitpunkt  $t_i$ , zu dem die Ausgänge einer (Teil-)Schaltung stabil anliegen (*data ready time*), lässt sich berechnen:

$$t_i = t_{PD,i} + \max_j t_j$$

wobei  $d_i$  die Gatterverzögerung des aktuellen Gatters  $i$  und  $t_j$  die Pfadverzögerungen bis zu dem jeweiligen Eingang  $j$  des Gatters  $i$  bezeichnen.



Die primären Eingänge werden mit  $t_i = 0$  angenommen.

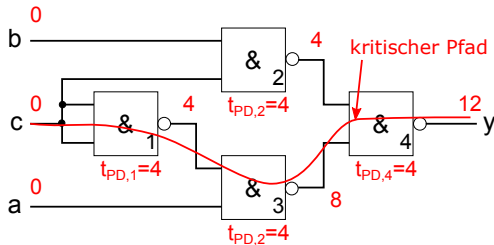
# Berechnung des kritischen Pfades



Durch Vorwärtspropagierung der Verzögerungszeiten lassen sich die max. Pfadverzögerungen zu allen Ausgängen berechnen.

Durch anschließende Rückwärtspropagierung entlang des längsten Pfades lässt sich der **kritische Pfad** bestimmen.

**Beispiel:** Für die verwendeten NAND-Gatter sei  $t_{PD} = 4$  ns.



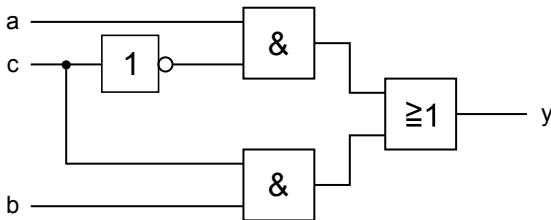
Der kritische Pfad hat eine Verzögerung von  $t_{py} = 12$  ns.

# Kritischer Pfad: Beispiel



## Vorlesungsaufgabe: Bestimmen Sie den kritischen Pfad!

Die Durchlaufzeiten aller Gatter betragen  $t_{pD} = 1$  ns.



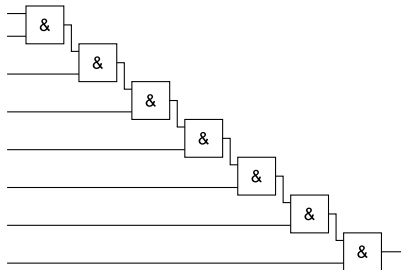
# Optimierung des Zeitverhaltens



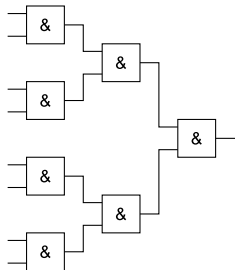
Beispiel 1: Zweistufige UND/ODER-Realisierung gegenüber mehrstufiger Logik

⇒ nur zwei Gatterstufen im kritischen Pfad

Beispiel 2: Balancierung



7 Gatterverzögerungen



3 Gatterverzögerungen

# Zeitverhalten sync. Schaltungen



Werden Schaltnetze und Flipflops zu komplexen Schaltungen zusammengefügt, muss neben der logischen Funktion auch das Zusammenspiel des Zeitverhaltens berücksichtigt werden.

Die meisten digitalen Schaltungen (z.B. Prozessoren) sind **synchrone** Schaltungen, d. h. die Zustandsänderung der Speicherelemente wird durch ein **Taktsignal** kontrolliert.

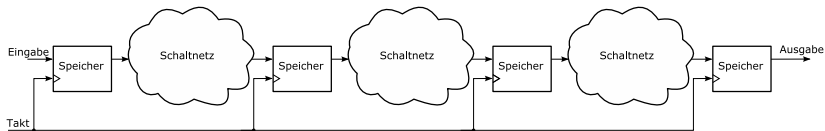
Die **maximale Taktrate** eines synchronen Systems wird wesentlich durch die Verzögerungszeiten in den Gatternetzen und den Verbindungen dazwischen bestimmt. Ausschlaggebend ist der **kritische Pfad**.

Zusätzlich müssen aber **auch** die Zeitbedingungen der Flipflops berücksichtigt werden.

# Zeitverhalten sync. Schaltungen



Eine synchrone Schaltung hat typischerweise den folgenden Aufbau:



Das Schaltnetz mit der größten Laufzeit bestimmt die maximale Taktfrequenz.

Der (zeitlich) längste Pfad in dem Schaltnetz mit der größten Laufzeit wird als **kritischer Pfad** bezeichnet.

Die **maximale** Durchlaufzeit des Schaltnetzes bezeichnet man als **propagation delay**,  $t_{PD}$ .

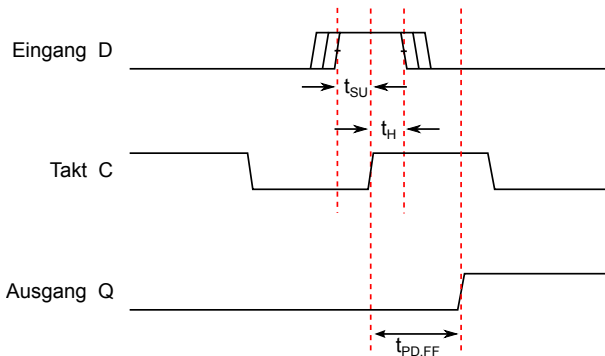
# Verhalten von Flipflops



Da Flipflops selbst als (rückgekoppelte) Schaltnetze aufgebaut sind, ist leicht einzusehen, dass auch für diese Schaltungen Zeitbedingungen existieren müssen:

- **Setup-Zeit**  $t_{SU}$ : Zeitspanne **vor** der aktiven Taktflanke, zu der eine Eingangsänderung bereits stabil vorliegen muss, um noch korrekt erkannt zu werden
- **Hold-Zeit**  $t_H$ : Zeitspanne **nach** der aktiven Taktflanke, in der ein Eingangssignal noch stabil anliegen muss, um die sichere Übernahme in den Flipflop-Speicher zu ermöglichen
- **Propagation-Zeit**  $t_{PD,FF}$  (auch Clock-to-Q delay genannt, von C→Q): Zeitspanne **nach** der aktiven Taktflanke, während der der geänderte Speicherinhalt noch nicht stabil am Ausgang anliegt

# Verhalten von Flipflops



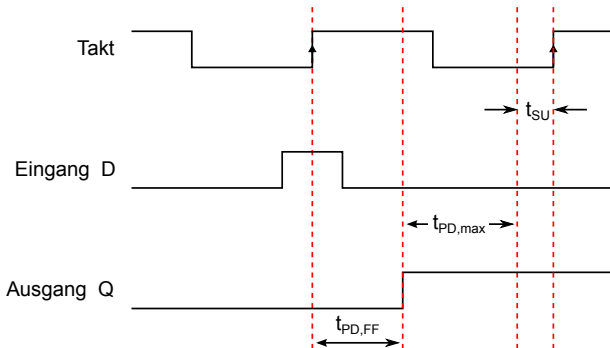
- $t_{SU}$ : Setup-Zeit (muss mindestens eingehalten werden)
- $t_H$ : Hold-Zeit (muss mindestens eingehalten werden)
- $t_{PD,FF}$ : Propagation-Zeit des Flipflops (Höchstdauer)



# Minimale Taktperiode



Für die minimal mögliche Taktperiode spielen folgende Zeiten eine Rolle:



$t_{PD,max}$  ist dabei die verfügbare Zeit für Logikfunktionen.

# Min. Taktperiode/Max. Taktfrequenz



Die **minimale Taktperiode**  $T_{\min}$  ergibt sich daraus zu

$$T_{\min} = t_{PD,FF} + t_{PD,max} + t_{SU}$$

Die **maximale Taktfrequenz**  $f_{\max}$  ergibt sich aus dem Kehrwert der minimalen Taktperiode:

$$f_{\max} = \frac{1}{T_{\min}}$$

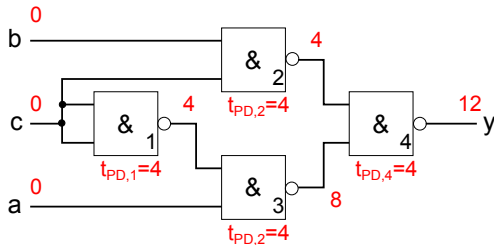
$t_{SU}$  und  $t_{PD,FF}$  sind konstant für ein Flipflop in einer gegebenen Technologie.

Soll ein schnellerer Takt verwendet werden, muss immer der kritische Pfad  $t_{PD,max}$  optimiert werden, da dieser die minimalen Taktperiode bestimmt.

# Min. Taktperiode/Max. Taktfrequenz



**Vorlesungsaufgabe:** Mit welcher Taktfrequenz kann die Schaltung maximal betrieben werden wenn sie in einer synchronen Schaltung betrieben wird?



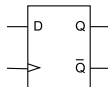
Die Zeiten der Flipflops lauten  $t_{PD,FF} = 3\text{ ns}$  und  $t_{SU} = 5\text{ ns}$ .

$$T_{\min} = t_{PD,FF} + t_{PD,\max} + t_{SU}$$

# D-Flipflops



Bisher haben wir ausschließlich 1-Bit Speicher wie z.B. das D-Flipflop betrachtet

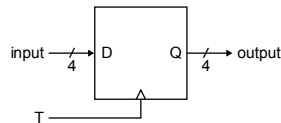
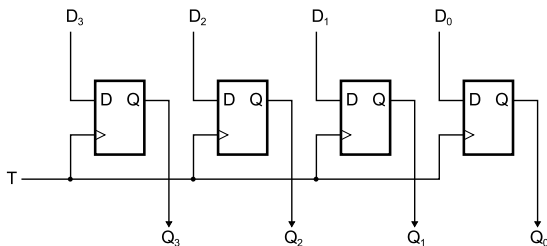


Wie können ganze Worte gespeichert werden?

# Register



Mehrere D-Flipflops bilden ein **Register**:



Parallele Anordnung von D-Flipflops, gemeinsamer Arbeitstakt

Speichert eine neue Eingabe mit **jeder** steigenden Taktflanke

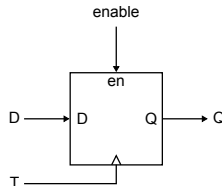
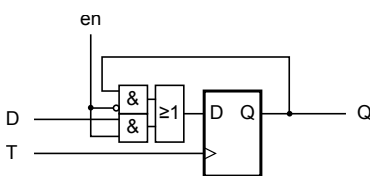
# Register mit Enable



Oft ist es sinnvoll, ein Datum **über mehrere Taktzyklen** zu halten, bevor es durch einen neuen Wert ersetzt wird.

Dazu wird das Schreiben durch ein Kontrollsignal (**Load** oder **Enable**) gesteuert → paralleles Laden oder Wert behalten.

Der Aufbau erfolgt aus D-Flipflops mit Enable:



$$\text{en} = \begin{cases} 0 & \text{aktuelle Ausgabe } Q \text{ bleibt gespeichert} \\ 1 & \text{Eingang } D \text{ wird gespeichert} \end{cases}$$

# Adressierbarer Speicher



Problem: Parallele Speicherung in Registern führt schnell zu einem I/O-Problem, da zu viele Leitungen benötigt werden

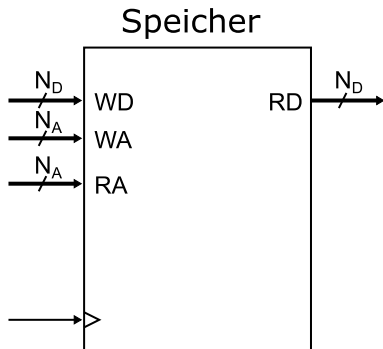
⇒ Lösung: Aufteilung in Daten- und Adressbits

Die **Datenbits** repräsentieren nur einen kleinen Teil des Speichers der zu einem Zeitpunkt gelesen oder geschrieben werden kann

Die **Adressbits** repräsentieren den Ort an dem die Datenbits geschrieben werden.

Speicher dieser Art lässt sich beliebig ansprechen, er wird daher auch **Random Access Memory (RAM)** bezeichnet.

# Funktionsweise Speicher



Adresse ( $N_A$ Bits)	Daten ( $N_D$ Bits)
000...000	Speicherzelle 1
000...001	Speicherzelle 2
000...010	Speicherzelle 3
⋮	
111...111	Speicherzelle $2^{N_A}$

**WD** und **RD**: Write- und Read-Data (Schreib-/Lese-Daten)

**WA** und **RA**: Write- und Read-Address (Schreib-/Lese-Adresse)

Mit  $N_A$  Adress-Bits lassen sich  $2^{N_A}$  Datenworte adressieren.



# Aufbau eines Speichers



Im RAM werden Speicherzellen als zweidimensionales **Array** mit  $2^{N_A}$  Zeilen zu je  $N_D$  Speicherzellen angeordnet.

Es entsteht ein  $2^{N_A} \times N_D$  Bit Speicher.

Jede Zeile speichert ein Datenwort mit  $N_D$  Bit

Werden Flipflops als Speicherstellen verwendet spricht man von einem **Registerfile**.