

Vorlesung 13

Das Thema dieser Vorlesung sind die Analog-Digital Wandler (ADC). Das Thema gliedert sich auf:

Charakterisierung eines A/D Konverters

INL, DNL, SNR, ENOB

Flash ADCs

Zeitbasierte ADCs

- Pseudodifferentieller Komparator
- TDC
- Vernier TDC
- time interleaving

Algorithmischer ADC

Zyklischer- und Pipeline ADC

Analog/digital-Wandler

(Analog/Digital Converter)

In modernen Schaltungen findet die Signal-Verarbeitung am meistens digital statt. Die Eingangs- und Ausgangsgröße ist normalerweise Analog. Bei der Umwandlung von analogen in digitale Signale wird Zeit und Amplitude diskretisiert. Ein Analog/Digital Wandler (ADC) diskretisiert die Amplitude und gibt für jedes Analog-Sample eine Zahl D als Ergebnis.

Man kann die Eingangsspannung V_{in} des ADCs als Funktion von D zeichnen und die Mittelwerte von V_{in} für jede Zahl D bestimmen:

$$\mu(V_{in}(D)) \equiv D_{in} \quad (1)$$

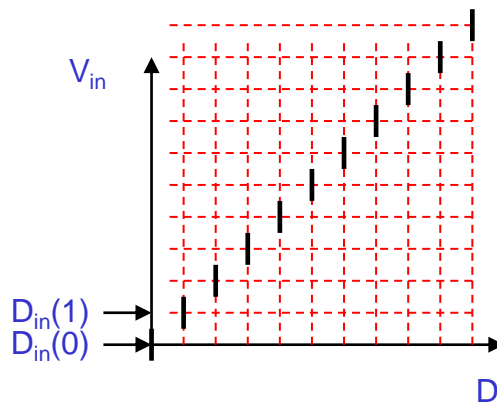


Abbildung 1: Bestimmung von ADC Kennlinie $D_{in}(D)$

Falls D Ergebnis der AD-Konversion ist, ist $D_{in}(D)$ eine gute Approximation des analogen Signals.

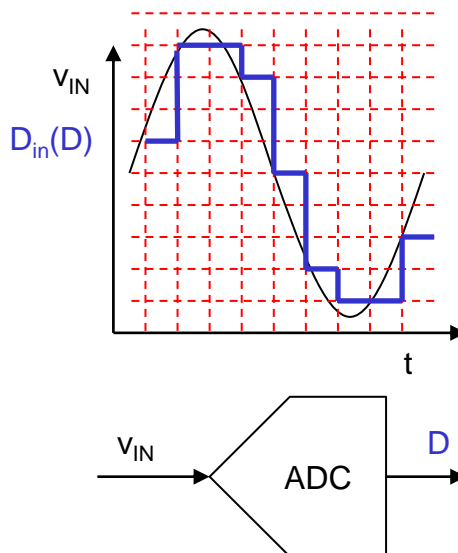


Abbildung 2: Funktion $D_{in}(D)$, wobei D Ergebnis von AD Konversion ist

Viele ADC-Arten benötigen nur die folgenden analogen Schaltungen: MOS-Schalter, Kondensatoren und Komparatoren. Analog/digital-Wandler (ADC) werden oft mit getakteten Schaltungen realisiert.

ADCs enthalten ebenfalls auch digitale Schaltungen. Die für uns am wichtigsten digitalen Schaltungen sind Inverter, Gated Inverter, Multiplexer, Latch, Flipflop, und Zähler. Alle diese Schaltungen können auf einem CMOS Chip realisiert werden. Die digitalen Schaltungen sind das Thema vom Kurs Design Digitaler Schaltkreise (DDS). Wir werden in dieser Vorlesung einige Vorkenntnisse von DDS voraussetzen. Eine kurze Zusammenfassung von wichtigsten digitalen Schaltungen befindet sich im Anhang (Abbildung 32 - Abbildung 35).

Die Schaltungen mit analogen und digitalen Komponenten nennen wir Mischsignalschaltungen (mixed signal circuit).

Charakterisierung von ADCs (optional)

ADCs können basierend auf folgenden Eigenschaften verglichen werden:

Auflösung	Resolution	Zahl von Bits
Abtastrate	Sampling rate	
Leistungsverbrauch	Power consumption	
INL	Integral nonlinearity	Integrale Nichtlinearität
DNL	Differential nonlinearity	Differenzielle Nichtlinearität
ENOB	Effective number of bits	Effektive Bit-Zahl

Genauigkeit hängt von Parametern wie integrale Nichtlinearität INL, differentielle Nichtlinearität DNL und Rauschen ab.

Die ADC-Eigenschaften können auf verschiedene Weise gemessen werden. Wir zeigen hier eine Methode.

Messung

Man kann eine präzise Eingangsquelle verwenden. Die Eingangsspannung V_{in} wird im gesamten ADC Signalbereich in feinen Schritten erhöht und für jede Eingangsspannung ADC Ausgang mehrfach ausgelesen und gespeichert.

Aus diesen Daten kann man für jeden ADC Ausgangscode D eine Menge von Eingangsspannungen $V_{in}(D)$ bekommen, die diesen Code als Ergebnis hatten. Aus diesen Mengen kann Mittelwert und Standardabweichung und von V_{in} für jeden Code bestimmt werden: $\mu(V_{in}(D))$ und $\sigma(V_{in}(D))$.

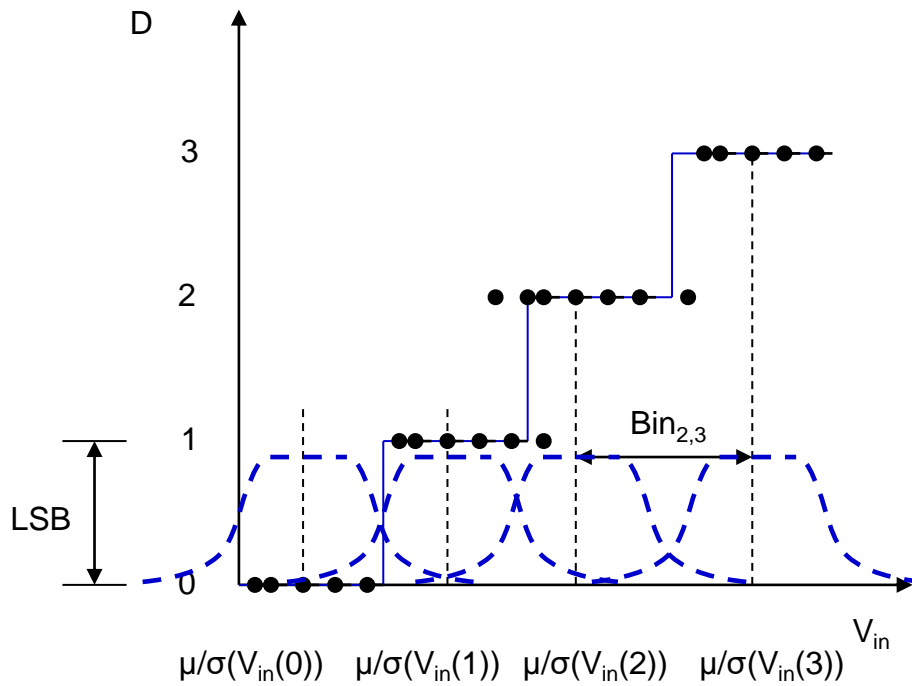


Abbildung 3: Darstellung von Mittelwerten und Standardabweichungen von Eingangsspannungen, die ein ADC Digitalausgang erzeugen.

Die Binbreite (bin width) kann als Abstand von Mittelwerten für zwei nachfolgende ADC Ausgänge definiert werden (Abbildung 3):

$$\text{Bin}_{i,i+1} = \mu(V_{in}(i + 1)) - \mu(V_{in}(i)) \quad (2)$$

Die durchschnittliche Binbreite ist:

$$\text{Bin} = \mu(\text{Bin}_{i,i+1}) \quad (3)$$

Die Standardabweichung hängt vom Rauschen ab. Im Falle eines idealen ADCs wäre die Funktion $V_{in}(D) - \mu(V_{in}(D))$ gleichverteilt. Die Breite der Verteilung wäre gleich wie die Binbreite.

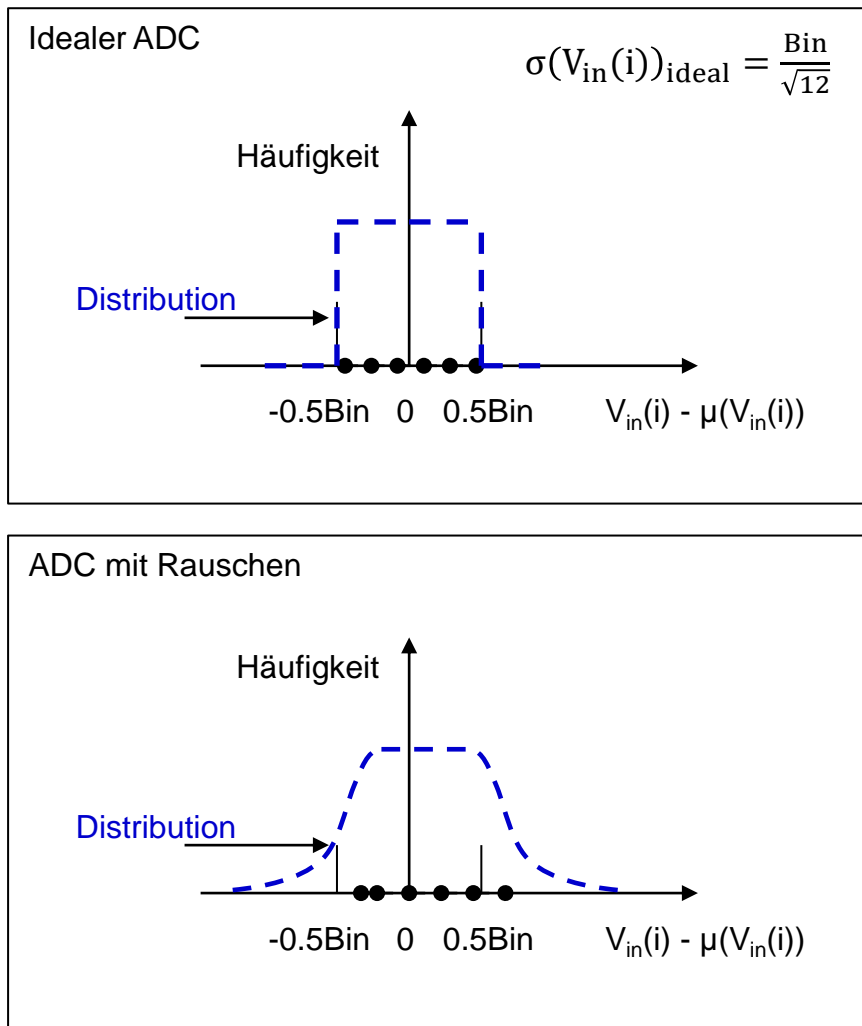


Abbildung 4: Verteilung von Messwerten im Falle vom idealen und rauschbehafteten ADC

Die Standardabweichung dieser Verteilung ist:

$$\sigma(V_{in}(D))_{ideal} = \frac{\text{Bin}}{\sqrt{12}} \quad (4)$$

Ein realistischer ADC hat größere mittlere Standardabweichung.

Definieren wir SNR als

$$\text{SNR}_{\text{dB}} = 20 \log_{10} \left(\frac{V_{in,\text{max}}}{\sigma(V_{in})} \right) \quad (5)$$

mit dem maximalen Signal:

$$V_{in,\text{max}} \equiv \mu(V_{in}(D_{\text{max}})) - \mu(V_{in}(D_{\text{min}})) \quad (6)$$

und dem Rauschen (dem Mittelwert von Standardabweichungen für alle Codes).

$$\sigma(V_{in}) \equiv \mu\sigma(V_{in}(i)) \quad (7)$$

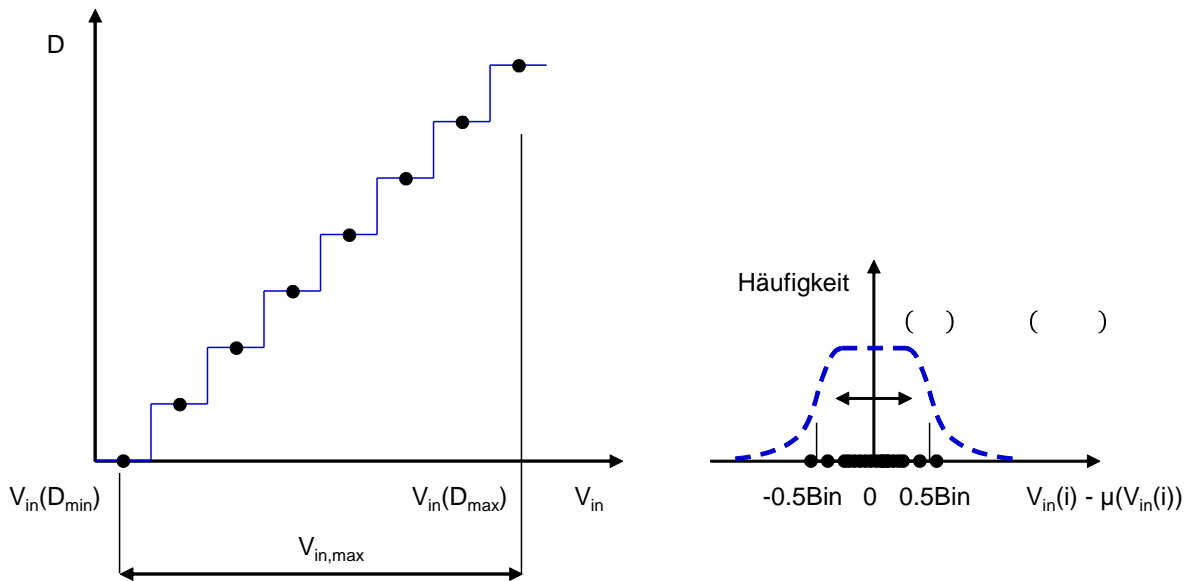


Abbildung 5: Maximales Signal und das Rauschen

Die effektive Zahl von Bits wird wie folgend definiert:

$$ENOB = \frac{SNR_{dB} - 20 \log_{10} \sqrt{12}}{20 \log_{10} 2} = \frac{SNR_{dB} - 10.79}{6.021} \quad (8)$$

Beispiel: Im Fall vom idealen 8-Bit ADC gilt:

$$\mu(V_{in}(D_{max})) - \mu(V_{in}(D_{min})) = 256 \times \text{Bin}$$

Daraus folgt:

$$SNR_{dB} = 20 \log_{10}(256 \times \sqrt{12}) = 59.0 \text{ dB}$$

Die effektive Zahl von Bits ist dann:

$$ENOB = \frac{59.0 - 20 \log_{10} \sqrt{12}}{20 \log_{10} 2} = 8$$

Manchmal wird statt SNR das Verhältnis zwischen Signalleistung P_{signal} (Angenommen wird ein Sinussignal mit Peak to Peak Amplitude $V_{in,max}$) und Rauschleistung ($P_{\text{noise}} = \sigma^2(V_{in})$) gemessen. Es gilt

$$SNR_{p,dB} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) = 10 \log_{10} \left(\frac{\frac{1}{2} \left(\frac{V_{in,max}}{2} \right)^2}{\sigma^2(V_{in})} \right) = 20 \log_{10} \left(\frac{V_{in,max}}{\sigma(V_{in})} \right) - 20 \log_{10}(2\sqrt{2})$$

oder

$$SNR_{p,dB} = SNR_{dB} - 20 \log_{10}(2\sqrt{2})$$

Wenn wir dieses Ergebnis in (8) einsetzen bekommen wir:

$$\text{ENOB} = \frac{\text{SNR}_{p,\text{dB}} - 20 \log_{10} \left(\frac{\sqrt{6}}{2} \right)}{20 \log_{10} 2} \sim \frac{\text{SNR}_{p,\text{dB}} - 1.76}{6.02} \quad (9)$$

Vorteil der Formel (9) ist es, dass SNR_p mittels der FFT von den ADC Samples bestimmt werden kann, falls der Analogeingang eine Sinus-Spannung ist.

Die Messungen von INL und DNL setzen voraus, dass wir für jeden Ausgangscode, den mittleren Eingangssignal kennen, das zu diesem Code führt: $D_{\text{in}} = \mu(V_{\text{in}})$. Beispiel: für 8 Bit ADC hätten wir 256 Punkte ($\mu(V_{\text{in}}(D)), D$). D ist Ausgangscode 0 – 255.

Im Falle einer echten Messung wären die Messpunkte für $D = 0$ und $D = 255$ wahrscheinlich nicht all richtig, da die der ADC Eingang außerhalb des Signalbereichs sein könnte und ADC Ausgang die Werte 0 oder 255 nimmt.

Wir könnten eine Gerade auf ($\mu(V_{\text{in}}(D)), D$) Punkte anpassen (für $0 < D < 255$).

Der Kehrwert von der Steigung der Gerade ist die durchschnittliche Binbreite Bin (3).

INL in LSBs (least significant bits) ist die maximale vertikale Abweichung von Punkt ($\mu(V_{\text{in}}(i)), i$) und der Gerade (Abbildung 6).

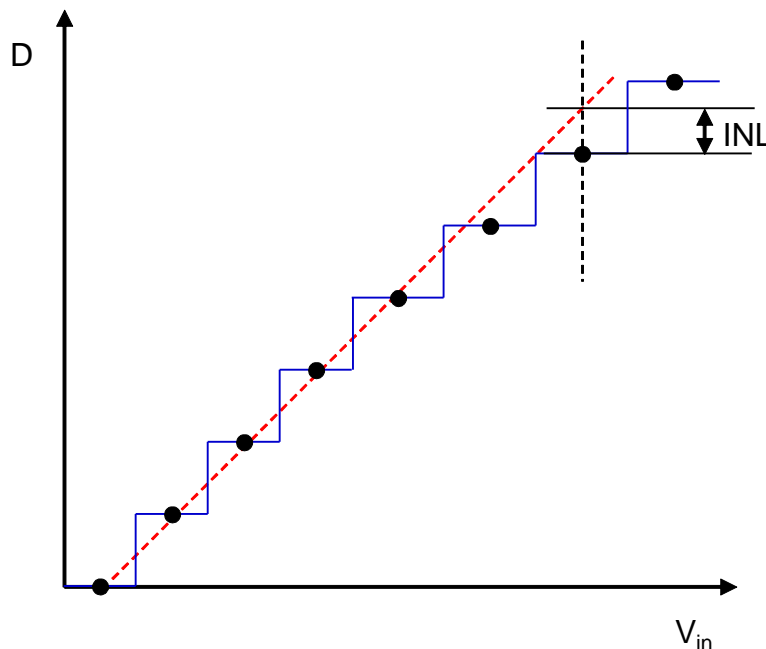


Abbildung 6: *Intergrale Nichtlinearität*

DNL in LSBs ist die Maximale Differenz $\mu(V_{\text{in}}(i)) - \mu(V_{\text{in}}(i+1))$ dividiert durch die durchschnittliche Binbreite minus 1 (Abbildung 7).

$$DNL = \frac{\mu(V_{in(i+1)}) - \mu(V_{in(i)})}{Bin} - 1 \quad (10)$$

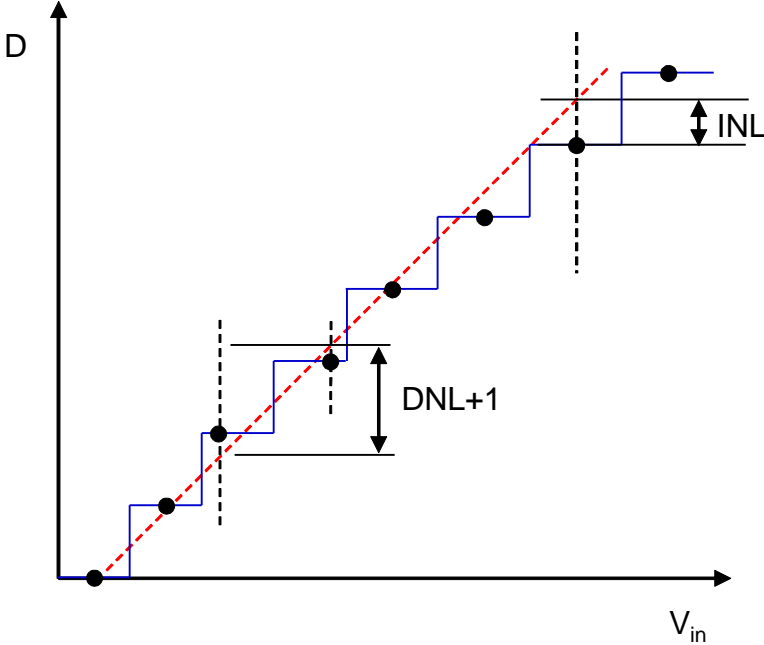


Abbildung 7: Differentielle Nichtlinearität

Wenn V_{in} Werte für manche D Werte Fehlen, kann DNL nicht bestimmt werden.

Flash ADC

Abbildung 8 zeigt Blockschaltplan eines 3-bit flash ADCs. Die Komponenten sind:

$2^n - 1$ Komparatoren (n ist die Zahl von Bits)

Eine kombinatorische Digitalschaltung, Kodierer, mit $2^n - 1$ Digital-Eingängen und n Digital-Ausgängen. Der n-Bit Ausgang ist die Binäre Zahl vom höchsten Eingang = 1. Wenn alle Eingänge 0 sind, ist auch Ausgang 0.

Eine Widerstandskette (resistor ladder), welche $2^n - 1$ Schwellen Th_i erzeugt.

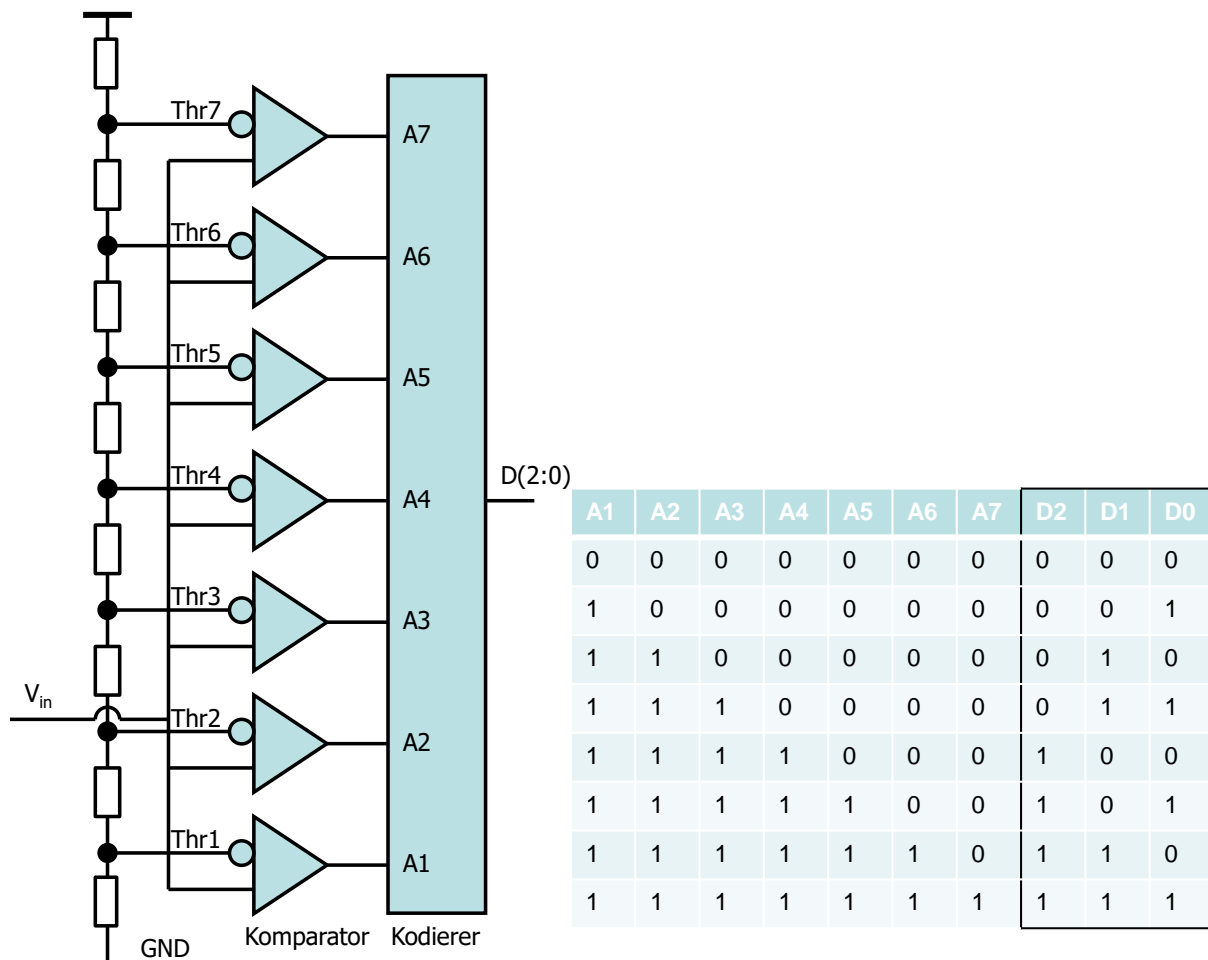


Abbildung 8: Blockschaltplan eines 3-bit flash ADCs

Die Funktionsweise ist relativ einfach. Für $Th_j > V_{in} > Th_{j+1}$, erzeugen alle Komparatoren $Comp_i$ ($i \leq j$), logisch 1 am Ausgang. Die restlichen Komparatoren ($i > j$) erzeugen 0. Die Komparator-Ausgänge sind an Kodierer angeschlossen. Der Kodierer generiert Binärkode $D(2:0)$ am Ausgang.

Die Komparatoren können entweder als getaktete oder als statische Schaltung implementiert werden.

Flash ADC hat normalerweise eine hohe Abtastrate. Der Nachteil ist die große Zahl von Komparatoren, was zu großem Stromverbrauch führt. Typische Auflösungen sind 6 – 8 Bits.

Zeitbasierte-ADCs

(Time-Based ADCs)

Diese ADC-Art hat sich ursprünglich durch einen niedrigen Stromverbrauch und sehr gute Auflösung ausgezeichnet.

Diese zeitbasierten ADCs galten als langsam.

In der letzten Zeit gab es interessante und schnelle Entwicklung bei diesen ADCs, da sie von Technologieskalierung besonders profitieren. Es wurden sehr hohe Abtastraten erreicht.

Die zeitbasierten ADCs basieren auf folgendem Prinzip (Abbildung 9):

Ein Spannung-Zeit Konverter (voltage to time converter - VTC) wandelt das Eingangssignal (Spannung) in ein Zeitsignal um. Das Zeitsignal ist ein Impuls mit der Amplitude auf dem logisch 1 Niveau. Die Impulsdauer (Impulslänge) ist zur Eingangsspannung proportional.

$$T_i = av_i + b$$

Manchmal werden zwei Signale start und stop generiert, wobei der der Abstand zwischen deren steigenden Flanken zur Eingangsspannung proportional ist.

Die Impulsdauer (Impulslänge) wird entweder mit einem Zähler oder mit einem Zeit-Digital Konverter (time to digital convertor - TDC) gemessen.

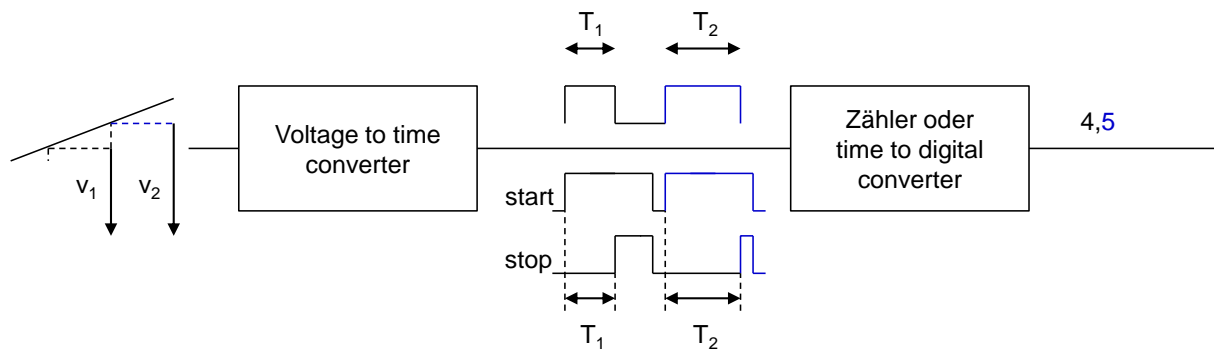


Abbildung 9: Zeitbasierter ADC

Beschreiben wir zuerst den VTC.

Spannung-Zeit Konverter

(Voltage to Time Converter - VTC)

Die einfachste Schaltung für Spannung-Zeit Konversion (Abbildung 10) besteht aus einer Sampling Schaltung (Schalter Sw und Kondensator C), einer Stromquelle I_{ramp} und einem Komparator Comp.

Die Schaltung befindet sich entweder in track Zustand ($\text{track} = 1$) oder in Konversionszustand ($\text{track} = 0$).

Eine State-Maschine erzeugt die Signale für die Ansteuerung von Komponenten.

Für $\text{track} = 1$, wird die Sampling-Kapazität auf V_{in} aufgeladen: $V_{\text{cap}} = V_{\text{in}}$. Die Schwellenspannung des Komparators wird relativ niedrig gewählt: $V_{\text{in}} > V_{\text{thr}}$.

Nachdem track logisch null wird, aktiviert man die NMOS Stromquelle I_{ramp} mit dem Signal en . Die Kapazität C wird entladen und die Spannung V_{cap} sinkt. Im Moment wenn V_{cap} die Schwelle unterschreitet, wechselt der Komparator Ausgang comp von 0 auf 1. Die UND Funktion zwischen dem negierten Komparator-Ausgang und dem Signal en ist das Ausgangssignal des Konverters.

Das Signal en kann als Start-Signal und der Komparator-Ausgang als Stop-Signal verwendet werden.

Die Impulslänge vom Signal out T ist durch die folgende Formel beschrieben:

$$I_{\text{ramp}} = \frac{C(\text{Th}-V_{\text{in}})}{T}; T = \frac{C(\text{Th}-V_{\text{in}})}{I_{\text{ramp}}} \quad (11)$$

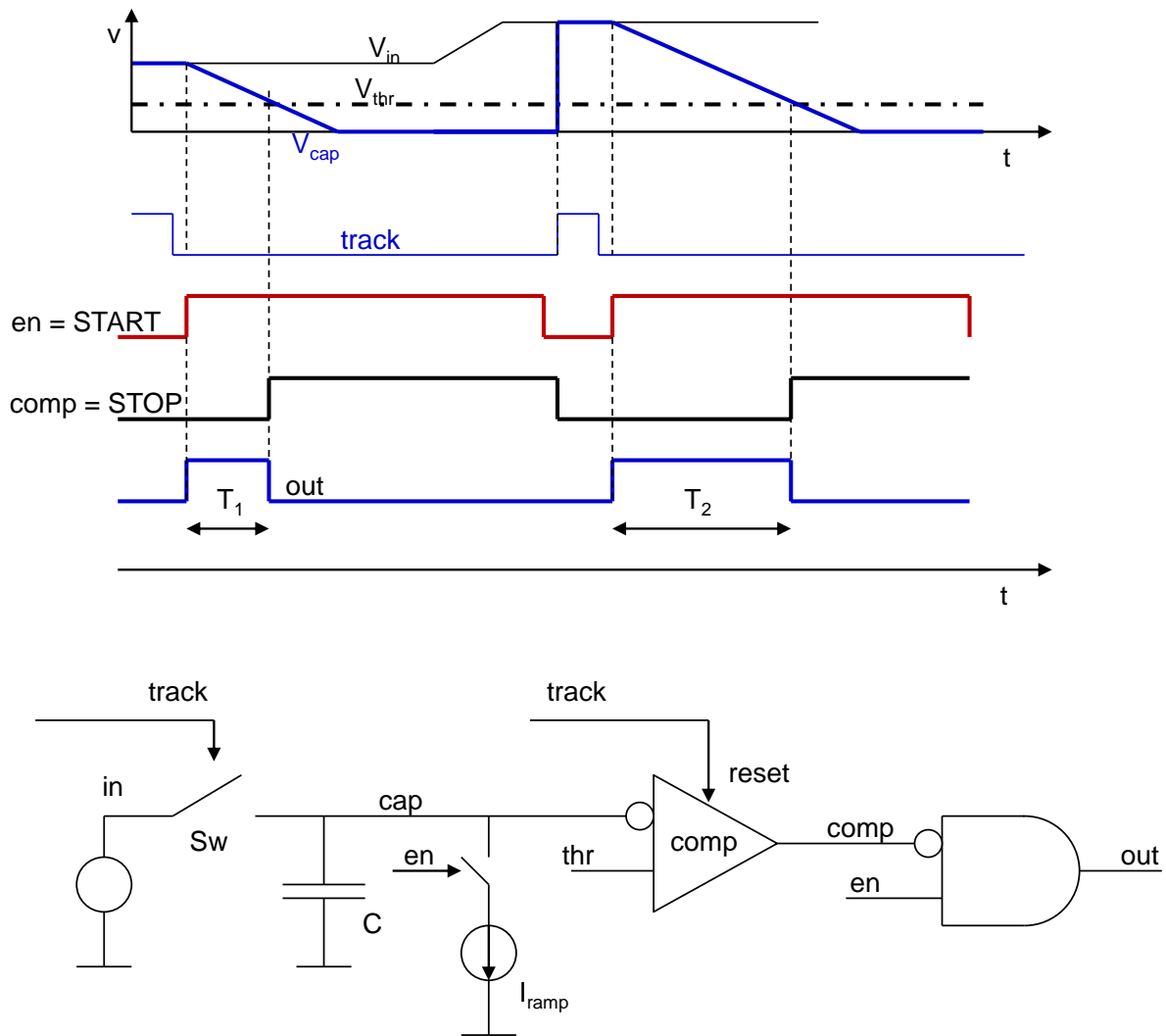


Abbildung 10: Spannung-Zeit Konverter

Designanforderungen

Der Signalbereich liegt über der Schwellenspannung V_{thr} . Wenn wir einen großen Signalbereich möchten, sollte V_{thr} niedrig sein. V_{in} kann bis V_{DD} hoch sein. Der Schalter Sw kann mit einem PMOS realisiert werden. Ein bootstrapped NMOS Schalter kann ebenfalls benutzt werden.

Der Komparator soll möglichst schnell sein. Ein zweistufiger Operationsverstärker kann verwendet werden (Abbildung 11).

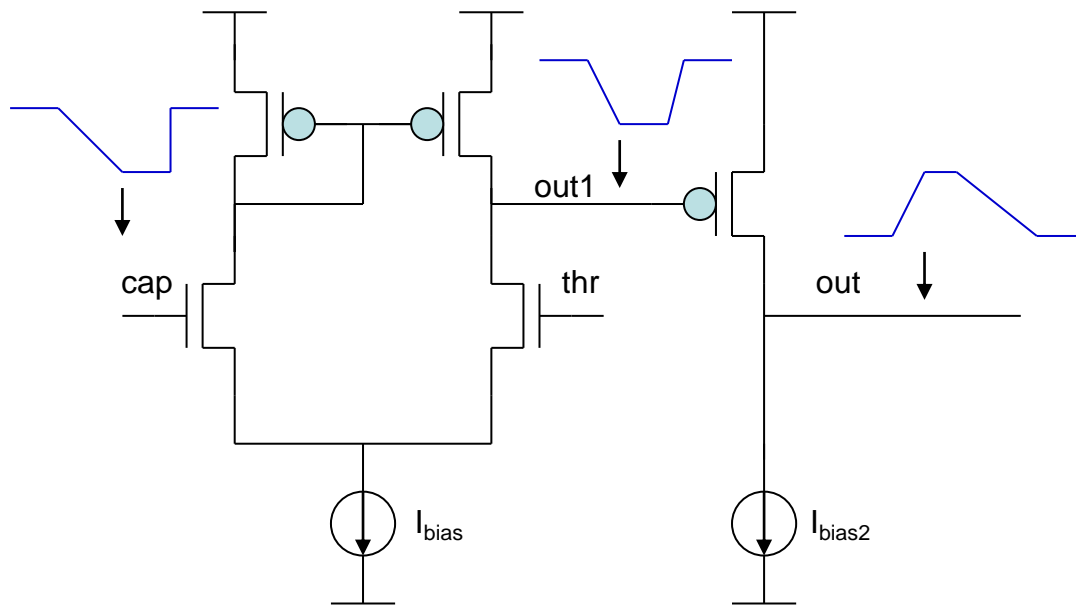


Abbildung 11: Standardkomparator

Diese Implementierung hat einige Nachteile: V_{thr} darf nicht zu niedrig sein, sonst ist die Stromquelle I_{bias} nicht in Sättigung, die Quelle I_{bias2} wird gegen das Ausgangssignal und macht den Komparator langsamer, es dauert relativ lange bis der Ausgang im track Zustand wieder 0 wird.

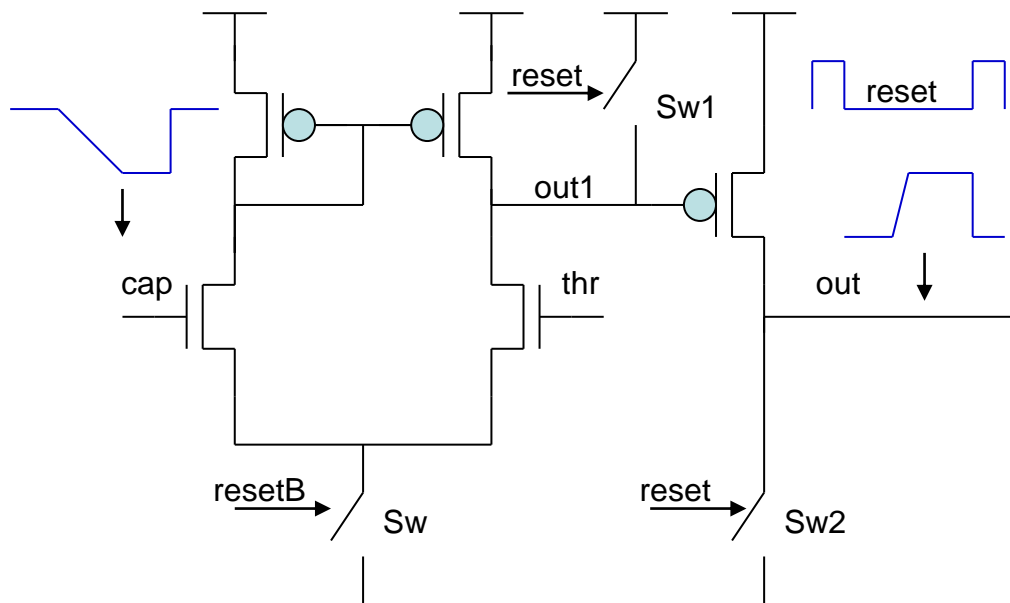


Abbildung 12: Pseudodifferentieller Komparator

Ein Komparator mit Reset (Abbildung 12) wird oft benutzt. Die Stromquellen I_{bias} und I_{bias2} werden durch Schalter Sw und Sw2 ersetzt. Die Schalter Sw1 und Sw2 sind im track Zustand geschlossen und sie stellen die Anfangswerte für $V_{out1} = V_{DD}$ und $V_{out} = 0$ her. Schalter Sw wird in der Konversionsphase geschlossen. Da das Differenzpaar keine Stromquelle sondern Schalter verwendet, kann V_{thr} niedriger sein. Die Differenzverstärker mit den Eingangstransistoren deren Sourcen gerdet sind nennt man pseudodifferenzielle Verstärker.

Der pseudodifferentielle Komparator ist schneller (Bias-Strom im Differenzpaar ist größer und I_{bias2} ist 0) und sein Ausgang kann schnell auf 0 gesetzt werden.

Eine gute Stromquelle I_{ramp} (Abbildung 10) ist für die Qualität der AD Konversion entscheidend. I_{ramp} soll möglichst unabhängig von V_{cap} sein. Das bedeutet, dass der Ausgangswiderstand der Stromquelle groß sein soll. Eine Quelle mit Kaskode ist manchmal eine gute Wahl. Auch das Rauschen der Stromquelle soll minimiert werden. Das Stromrauschen (Rauschleistung) ist zur Transkonduktanz proportional. Stromquelle mit niedriger Transkonduktanz ist in dem Sinne besser. Niedrige Transkonduktanz bedeutet hohe $V_{dssat,ramp}$. Wir sehen dass sich die Anforderung für gute Linearität im großen Signalbereich und kleine Transkonduktanz widersprechen.

Genauigkeit (optional)

Die Größte Fehlerquelle ist das Rauschen im Strom I_{ramp} . Das Rauschen wird mit einer Größe – Spektrale Leistungsdichte beschreiben. Für eine MOSFET Stromquelle gilt:

$$S_I = \frac{2}{3} 4KTg_m = \frac{\frac{2}{3} 4KT2I_{ramp}}{V_{dssat}} = \frac{\frac{16}{3} eU_T I_{ramp}}{V_{dssat}} \quad (12)$$

Man kann zeigen, dass wenn eine Kapazität C für die Dauer T mit dem rauschbehafteten Strom I_{ramp} aufgeladen wird (Abbildung 13, links), hat die Kondensator-Spannung nach der Zeit T eine Standardabweichung von:

$$\sigma^2(v_{cap}) = \frac{1}{C^2} \sigma^2(Q) = \frac{1}{C^2} \sigma^2(TI) = \frac{T^2}{C^2} \sigma^2(I) = \frac{T^2}{C^2} \frac{1}{2} \frac{S_I}{T} = \frac{1}{2} \frac{S_I T}{C^2} \quad (13)$$

Bei dem time based ADC ist die Situation etwas anderes (Abbildung 13, rechts). Es wird die Zeit T gemessen in der der Kondensator von V_{in} auf V_{thr} entladen wird. V_{thr} ist bekannt (Rauschfrei). Die Frage könne wie folgend lauten: Wie groß ist die Messunsicherheit $\sigma(v_{in})$ wenn T bekannt ist?

Da wir nach der Aufladung in der Zeit T eine Unsicherheit in der Kondensatorspannung von $\sigma(v_{cap})$ haben (13), hat auch die gemessene Größe v_{in} die gleiche Unsicherheit:

$$\sigma^2(v_{in}) = \frac{1}{2} \frac{S_I T}{C^2} \quad (14)$$

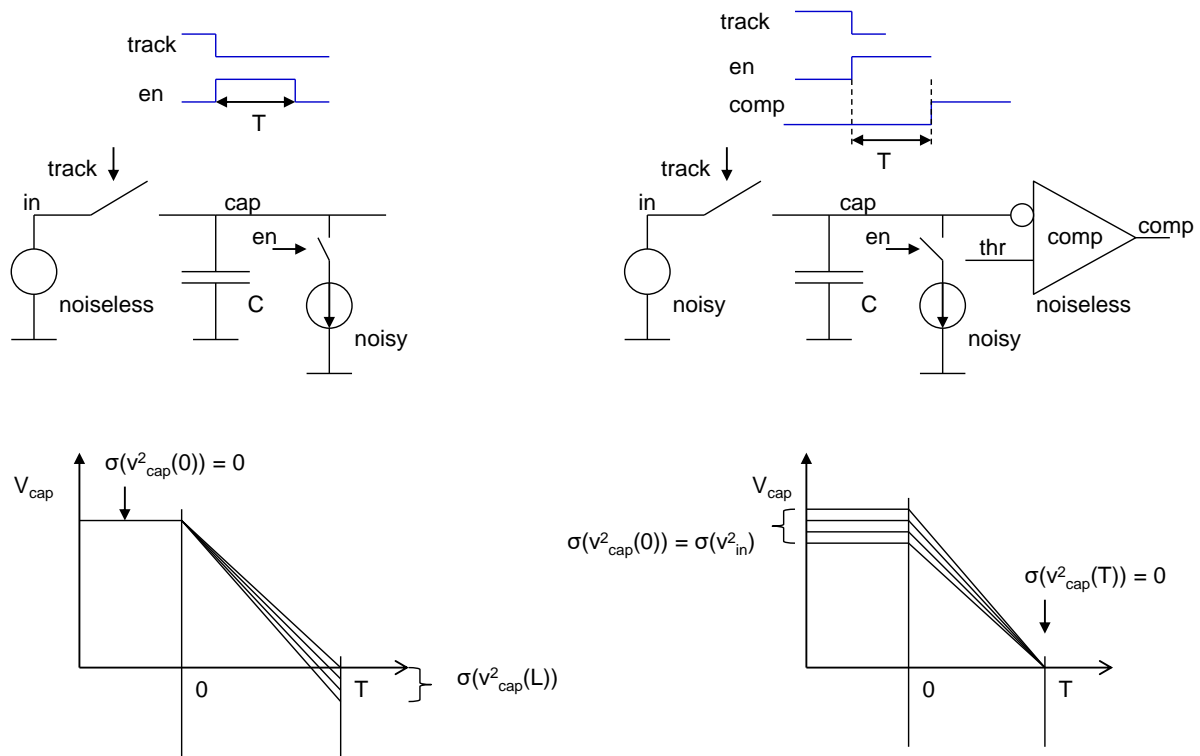


Abbildung 13: Links: Kapazität C wird für die Dauer T mit dem rauschbehafteten Strom I_{ramp} entladen. Rechts: Kapazität C wird auf eine Spannung V_{in} aufgeladen und mit I_{ramp} entladen bis ihre Spannung gleich V_{thr} wird. Die Zeit T wird gemessen. Wenn eine bestimmte Zeit gemessen wird, wie groß ist dann die Unsicherheit in V_{in} Messung?

Berechnen wir nun Signal zu Rauschen Verhältnis definiert als

$$SNR = \frac{V_{in,max}}{\sigma(v_{in})}$$

Es gilt

$$\frac{T_{max} I_{ramp}}{C} = \frac{Q_{max}}{C} = V_{IN,max} - V_{thr} = v_{in,max}$$

Es folgt:

$$SNR = \frac{v_{in,max}}{\sigma(v_{in})} = \frac{\frac{Q_{max}}{C}}{\sqrt{\frac{1}{2} \frac{S_I T_{max}}{C^2}}} = I_{ramp} \sqrt{\frac{2 T_{max}}{S_I}} = \sqrt{\frac{Q_{max} V_{dssat}}{\frac{8}{3} e U_T}} \quad (15)$$

Beispiel: Für die Parameter $C = 100 \text{ fF}$, $v_{in,max} = 0.8 \text{ V}$, $V_{dssat} = 200 \text{ mV}$, bekommen wir mithilfe von (15) $SNR = 1200$ und mit (8) $ENOB = 8.436$.

Rampen-ADCs sind können ziemlich genau sein. Weitere gute Eigenschaften sind ein niedriger Stromverbrauch und eine kleine Layout-Fläche.

Beschreiben wir nun die Schaltung für Zeit-Digital Konversion.

Zeitmessung mit Zähler

Die einfachste Schaltung für Zeitmessung basiert auf einem Zähler.

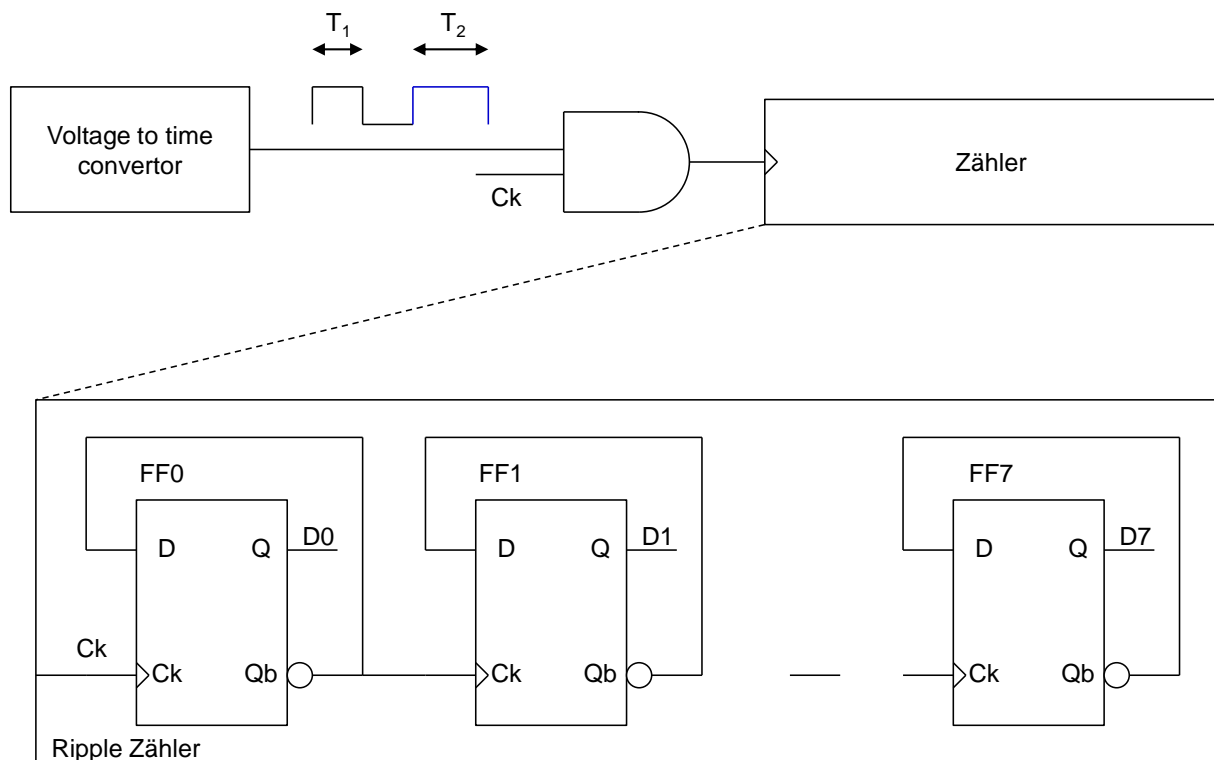


Abbildung 14: Zeitmessung mit einem Ripple-Zähler

Der Zähler zählt solange der Ausgang vom VTC 1 ist. Der Zustand des Zählers n entspricht der Zeit T .

$$T = nT_0$$

T_0 ist die Taktperiode.

Ein Problem bei dieser Schaltung ist die Konversionsgeschwindigkeit. Ein 8-bit ADC hat T_{\max} von:

$$T_{\max} = 2^8 T_0$$

Es ist nicht leicht T_0 kleiner als 250 ps zu erreichen. Deshalb wäre die maximale Konversionsgeschwindigkeit für einen 8-bit ADC etwa $256 \times 250 \text{ ps} = 64 \text{ ns}$.

Warum ist die Taktgeschwindigkeit bei einem Zähler begrenzt?

Ein Zähler besteht aus Speicherzellen – Flipflops – und einer kombinatorischen Logik welche den nächsten digitalen Zustand rechnet.

Struktur von Flipflops und Zähler wurde in DDS erklärt.

Die minimale Taktperiode ist ungefähr die Summe von Signalverzögerung im Flipflop und in der kombinatorischen Logik.

Versuchen wir diese Verzögerungen (Delays) genauer zu bestimmen. Die kombinatorische Logik eines binären 10-Bit Zähler kann kompliziert sein. Die einfachste Implementierung ist ein Ripple-Zähler (Abbildung 14). In diesem Zähler wird der negierte Datenausgang Qb vom LSB Flipflop FF0 an seinen Dateneingang D angeschlossen.

Minimale Taktperiode ist ungefähr die Summe von allen Verzögerungen im Signalpfad vom Takteingang bis zum Flipflop-Ausgang und durch die Rückkopplung wieder zum Flipflop-Eingang.

Die Struktur des Flipflops FF0 und der Signalpfad (blau) ist in Abbildung 15 gezeigt.

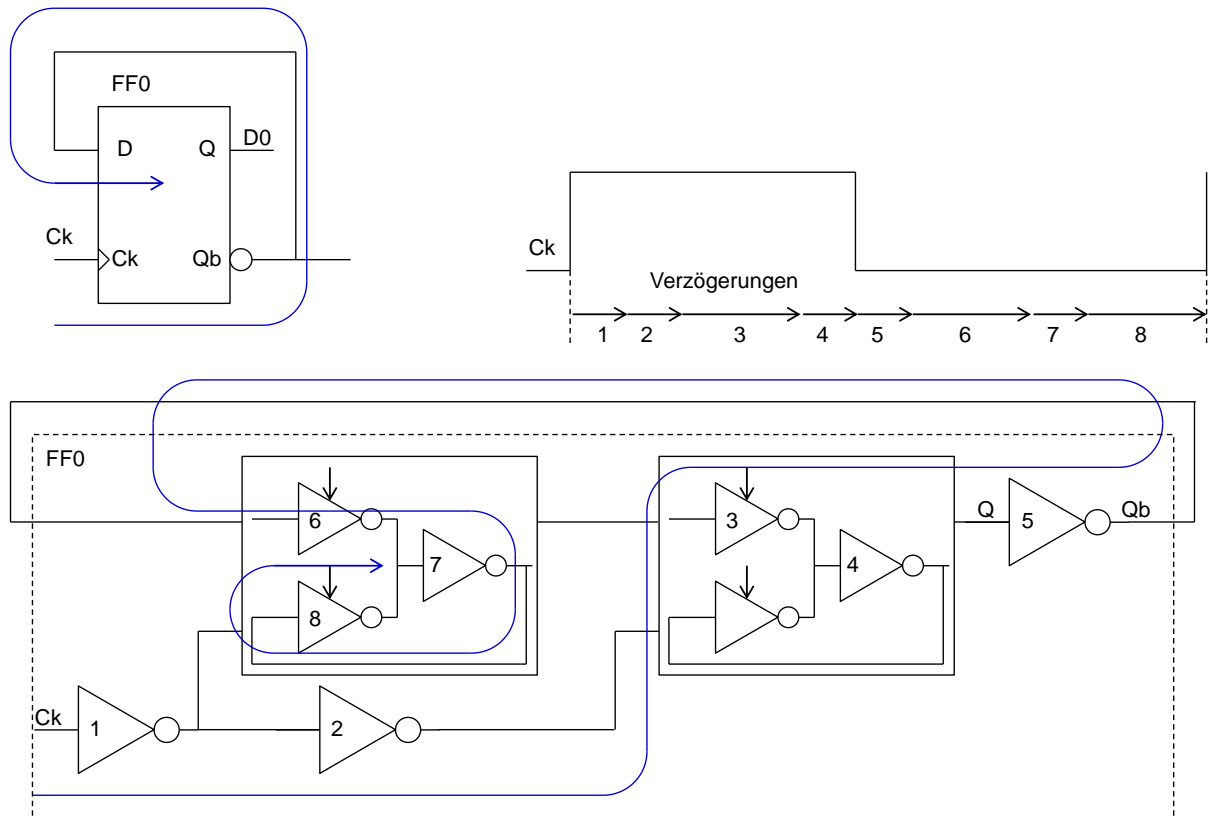


Abbildung 15: Die Struktur des Flipflops FF0 und der Signalpfad (blau)

Im Signalpfad befinden sich 8 Inverter und Gated-Inverter. Es müssen insgesamt 8 Konten mit ihren Kapazitäten zwischen GNN und VDD geladen werden.

Die kleinste Delay hat ein Inverter, angeschlossen an ein weiteres minimum size Inverter, der eine kapazitive Last C_{load} darstellt.

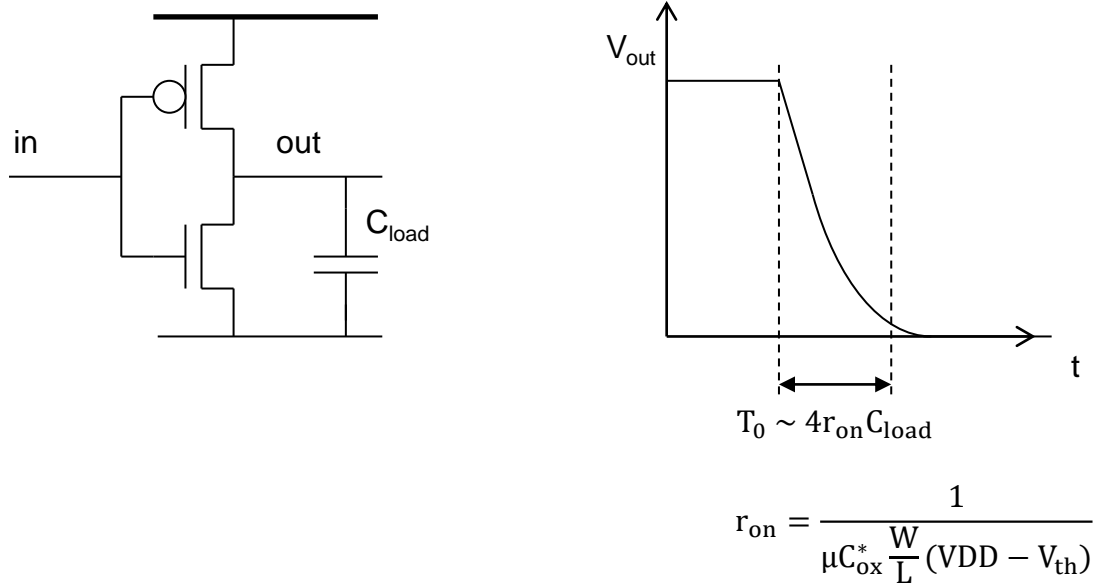


Abbildung 16: Entladezeit am Ausgang eines Inverters

Nennen wir diese Zeit T_0 .

Die Signalverzögerung in einem Inverter beträgt etwa $4 C_{load} R_{on}$.

R_{on} ist verschieden für verschiedene Komponenten:

Inverter: R_{mosfet}

Gated Inverter 2 R_{mosfet}

Auch die Kapazitäten C_{load} sind verschieden, z.B. Inverter 4 sieht muss zwei Inverter treiben.

Die Summe von Verzögerungen im Datenpfad des Ripple Zählers ist etwa $10 T_0$. Typischerweise ist diese Verzögerung > 250 ps. T_0 ist in einer modernen CMOS Technologie ($L < 65$ nm) etwa 25 ps oder kleiner.

Zeit-Digital Konverter

(Time to Digital Converter - TDC)

Eine Verbesserung der Konversionsgeschwindigkeit erreicht man durch die Verwendung vom Zeit-Digital Konverters.

Ein TDC kann auf viele Arten realisiert werden. Eine Implementierung als analoge Schaltung, basierend aus einer Rampe, ist üblich. Eine Kapazität wird für die Dauer des Pulses geladen und die Spannung mit einem ADC gemessen.

Für uns interessanter sind die digitalen TDCs. Eine Idee ist es die schnelle Verzögerung eines Inverters oder Buffers (zwei Inverter in Reihe) für die Zeitmessung zu benutzen. Eine Buffer-Kette nennen wir Delay Line. Abbildung 17 zeigt die Prinzip-Schaltung.

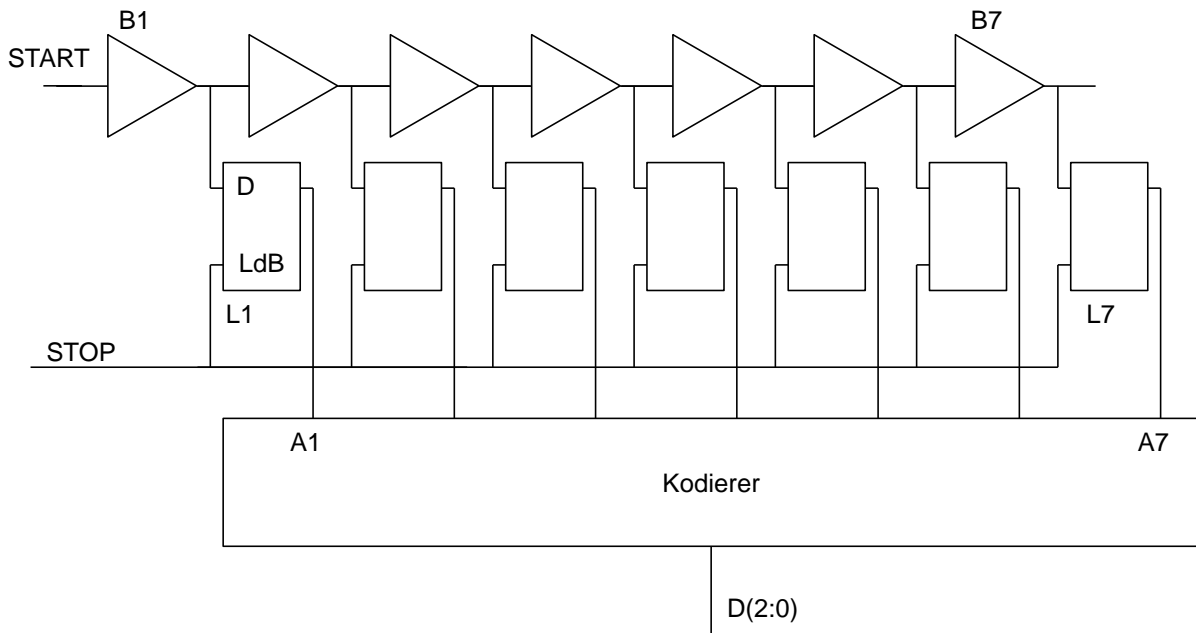


Abbildung 17: Zeit-Digital Konverter (time to digital converter)

Eingang des ersten Buffers in der Kette (B1 - B7) ist an das Startsignal START angeschlossen. Nach der steigenden Start-Flanke wechseln die Buffer nacheinander den Zustand, auf ähnliche Weise wie bei Dominos wenn sie umfallen. Die Steigende Flanke des STOP-Signals speichert den momentanen Zustand der Buffer-Ausgänge. Dafür verwendet man schnelle Speicherzellen – Latches (L1 – L7). Auf diese Weise kann die Verzögerung zwischen START und STOP mit einer Binbreite von T_0 gemessen werden. Eine etwa $10\times$ bessere Auflösung als mit einem Zähler kann erreicht werden.

Damit die Schaltung in unserem ADC verwendet werden kann, muss der VTC Start- und Stop-Signale erzeugen.

Beachten wir, dass die Kette lang sein muss. Für eine 8-bit Auflösung benötigt man $2^8 - 1 = 255$ Buffer. Der gespeicherte Code hat 255 bits. Es handelt sich um ein so genannter Thermometercode (wie bei dem flash-ADC). Diesen Code kann man mit einem Kodierer in 8-bit Binärcode umwandeln. Die Schaltung in Abbildung 17 zeigt einen 3-bit TDC.

Abbildung 18 zeigt Verlauf von Signalen.

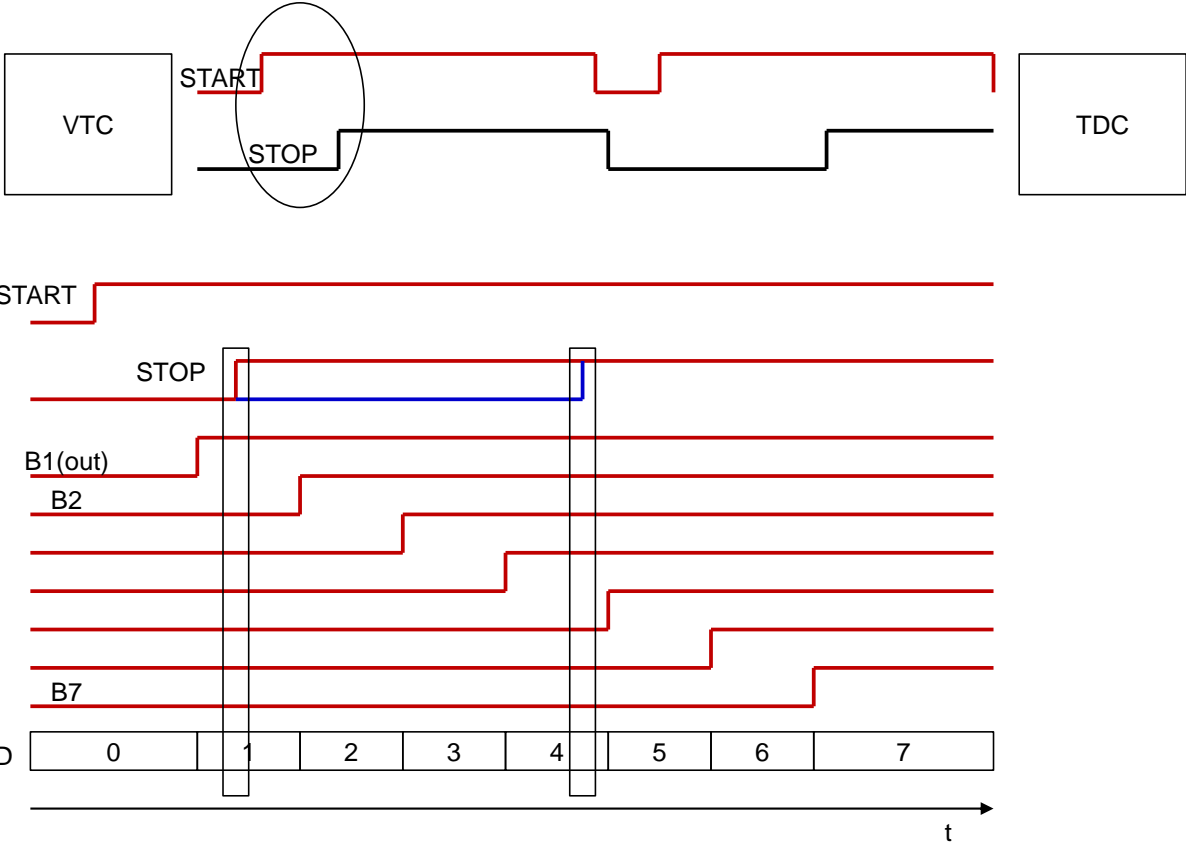


Abbildung 18: Signalverlauf bei einem TDC. Das rote Stop-Signal speichert Code 1, das blaue Stop-Signal speichert Code 4.

Abbildung 19 zeigt die Struktur der Latch Schaltung.

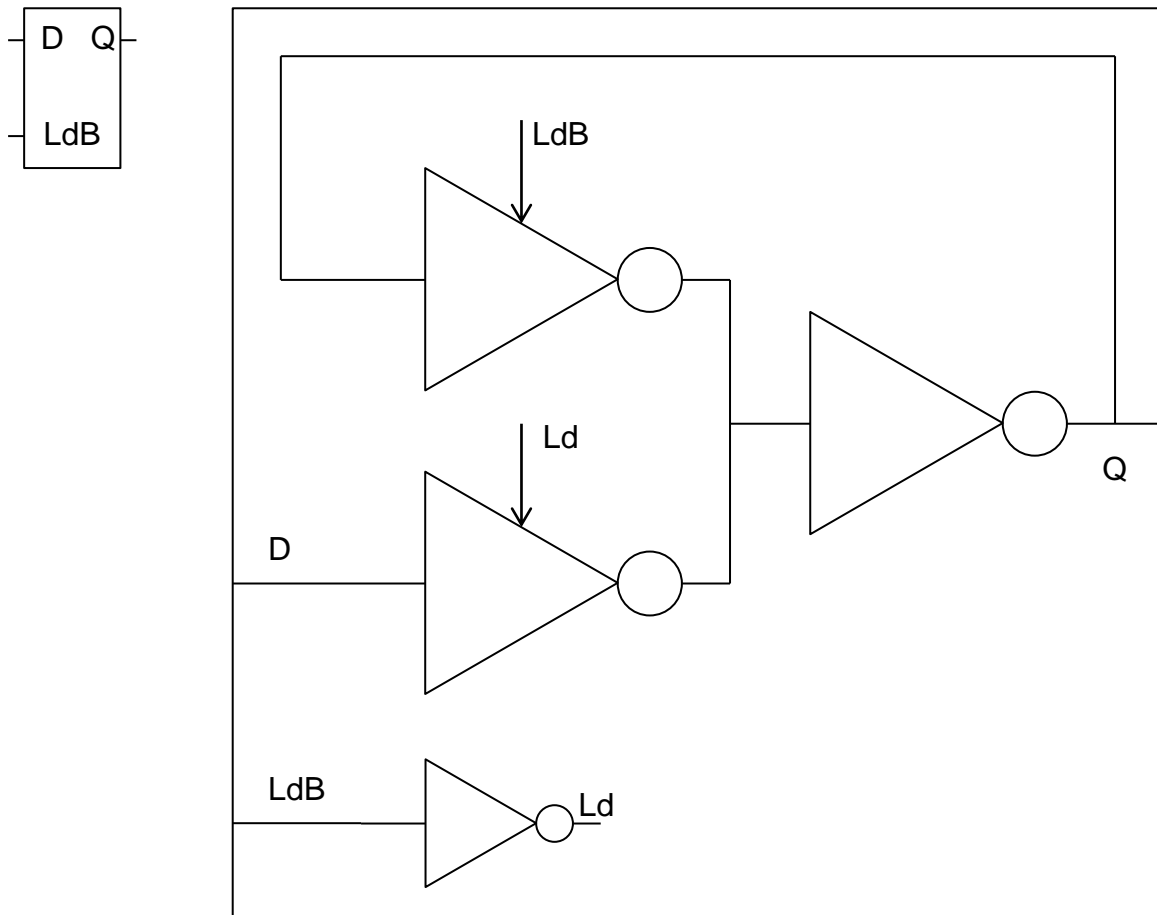


Abbildung 19: Struktur der Latch-Schaltung

Hochauflösende TDCs

Wie kann man die Binbreite einer TDC weiter verbessern?

Eine interessante Möglichkeit ist die Vernier-TDC.

Pierre Vernier (19. August 1580 in Ornans in der Franche-Comté; † 14. September 1637 ebenda) war ein französischer Mathematiker, Münzdirektor von Burgund, Kommandant des Schlosses Ornans und Rat des Königs von Spanien. (Wikipedia)*

Vernier-TDC benutzt zwei Buffer-Ketten. Die Buffers in der Kette, die STARTV als Eingang hat (SB1 – SB7) sind etwas langsamer. Sie haben die Delay von $T_0 + \Delta T$. Die Kette, die STOPV als Eingang hat, benutzt die Standard-Buffes (Delay von T_0).

Die Speicherzelle L_i speichert 1 nur wenn Ausgang i der langsameren Kette (S_{Bi}) vor dem Ausgang i der schnelleren Kette (S_i) 1 wird.

Das Prinzip ist gleich wie Nonius einer Schieblehre (Abbildung 21).

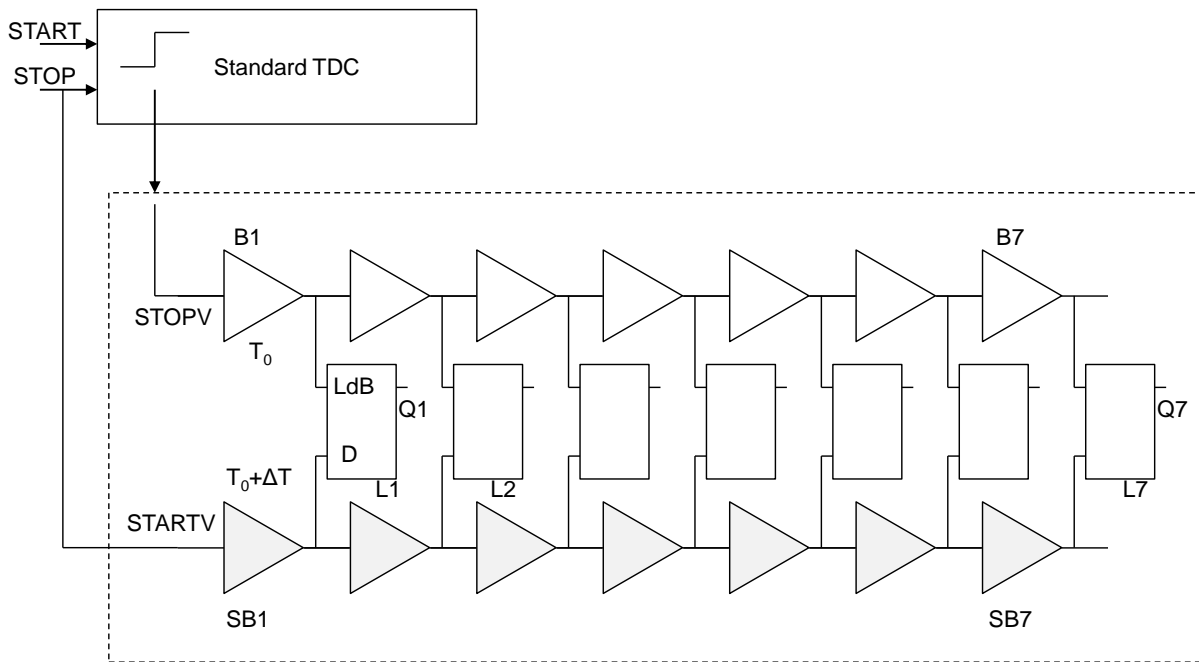


Abbildung 20: Vernier-TDC



Abbildung 21: Nonius

Vernier-TDC wird immer in Kombination mit dem Standard-TDC (genannt auch Flash-TDC) verwendet. Abbildung 22 zeigt die Signale.

Das Startsignal vom Vernier-TDC (STARTV) ist das Stop-Signal des Standard-TDCs.

$$\text{STARTV} = \text{STOP}$$

Vernier-TDC wird gestartet nachdem der Standard-TDC seine Messung beendet hat.

Stop-Signal vom Vernier-TDC (STOPV) ist die erste Signalflanke in der Kette des Standard-TDCs nach seinem Stop-Signal (Abbildung 22).

Vernier-TDC macht die feine Messung und Standard-TDC die grobe Messung. Im Beispiel von Abbildung 22 ist $T_0 = 8 \times \Delta T_0$ und die Buffer-Ketten haben 8 Buffer.

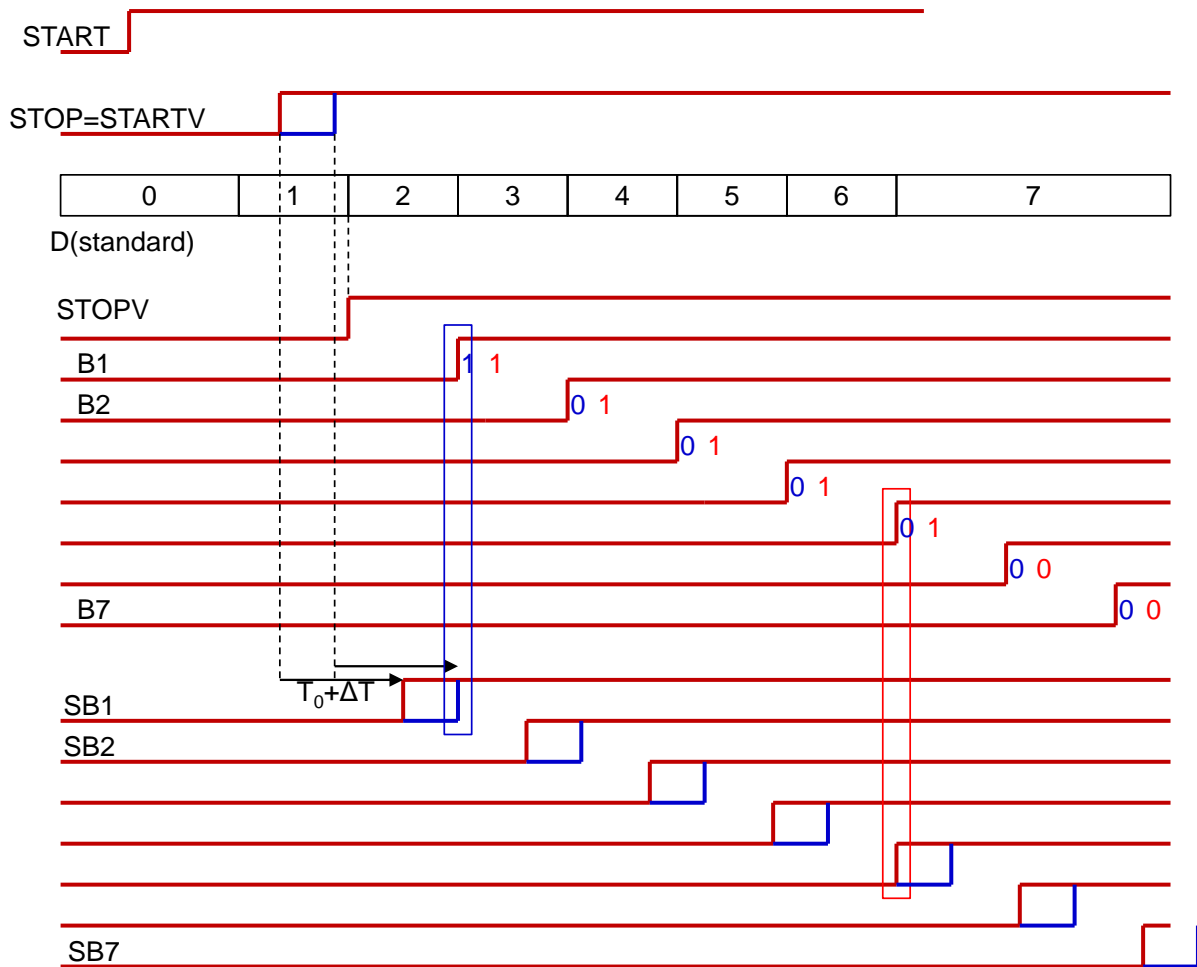


Abbildung 22: Signale im Vernier-TDC

Abbildung 22 zeigt zwei STOP Signale. Die Zeitdifferenz ist kleiner als T_0 und beide Signale führen zum gleichen Ergebnis im Standard-TDC (1). Im Fall vom roten STOP Signal ist das gespeicherte Ergebnis in Vernier-TDC $Q(1:7) = 1111100$. (Die langsamere Kette wird früher gestartet und fünf 1-en werden gespeichert bevor die schnellere Kette die langsamere überholt.) Im Fall vom blauen STOP Signal ist das gespeicherte Ergebnis in Vernier-TDC $Q(1:7) = 1000000$. (Die langsamere Kette wird später gestartet und wird von schnelleren Kette überholt nach dem ersten Buffer.)

Die Binbreite von Vernier-TDC ist ΔT_0 .

Abbildung 23 zeigt die vollständige Schaltung. Schaltung für Erzeugung von STOPV ist auch gezeigt. Sie benutzt die Flip-Flops DFF1 – DFF7. Die Ck Signale von diesen Flip-Flops sind an die Ausgänge von den Buffers des Standard-TDCs angeschlossen. Der D-Eingang von den Flip-Flops ist mit dem STOP Signal verbunden. STOPV ist die Oder-Funktion von allen Flip-Flop Ausgängen. STOPV ist synchron mit dem Buffer Ausgang, der als erster nach dem STOP-Signal von 0 auf 1 wechselt.

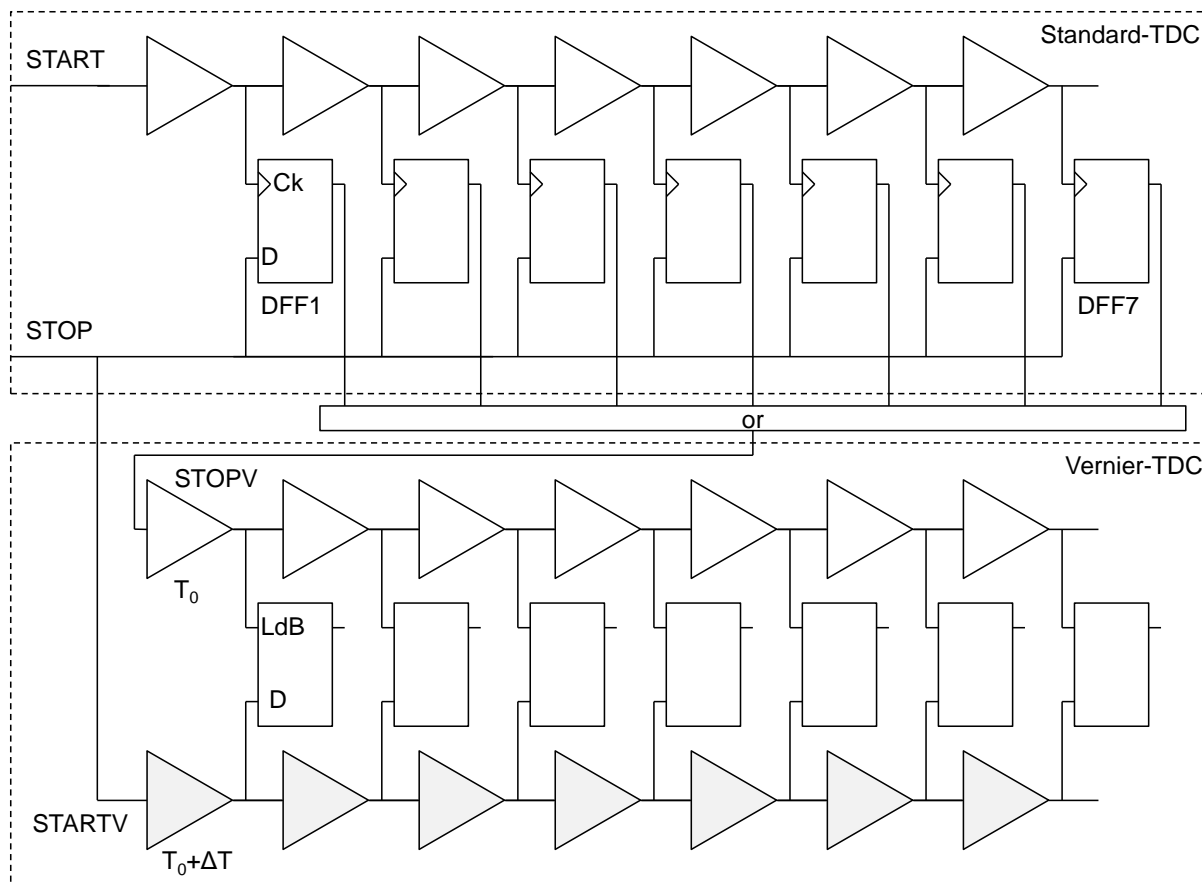


Abbildung 23: Vollständige Schaltung mit der Standard- und Vernier-TDC. Schaltung für Erzeugung von STOPV ist gezeigt.

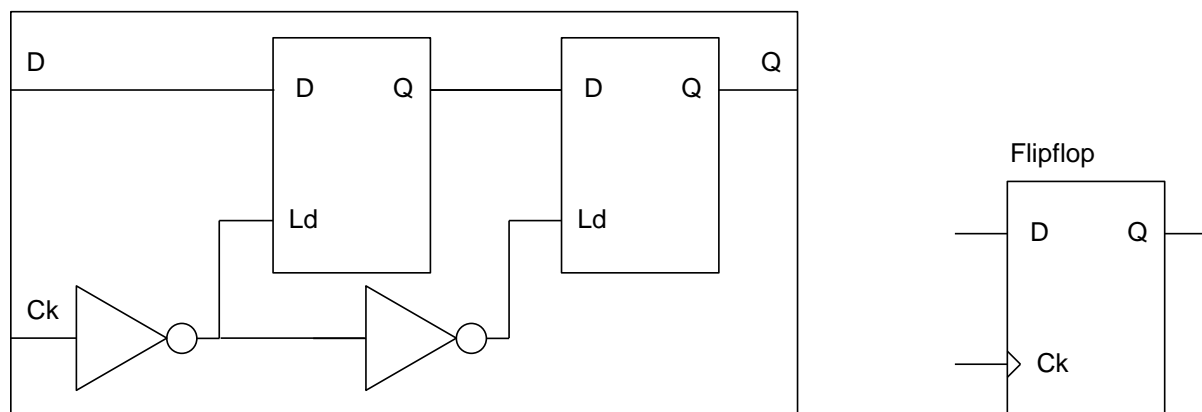


Abbildung 24: Struktur des Flip-Flops

Abbildung 24 zeigt die Struktur des Flip-Flops. Es besteht aus zwei Latch-Schaltungen.

Die Binbreite von Vernier-TDC kann typischerweise $16\times$ kleiner sein als T_0 . Auf diese Weise können die Konversionsgeschwindigkeiten deutlich erhöht werden.

Beispiel:

Vernier- und standard-TDC haben jeweils eine Kette von 16 Buffers. Beide TDC geben eine 4-Bit Zahl als Ergebnis, die kombinierte Auflösung ist 8 Bits. Die Binbreite vom standard-TDC

ist $T_0 = 25 \text{ ps}$ und die Binbreite vom Vernier-TDC ist $25\text{ps}/16 = 1.5625 \text{ ps}$. Die A/D Konversionszeit beträgt ungefähr $16 \times 25 \text{ ps} = 400\text{ps}$. (2.5 G samples /s).

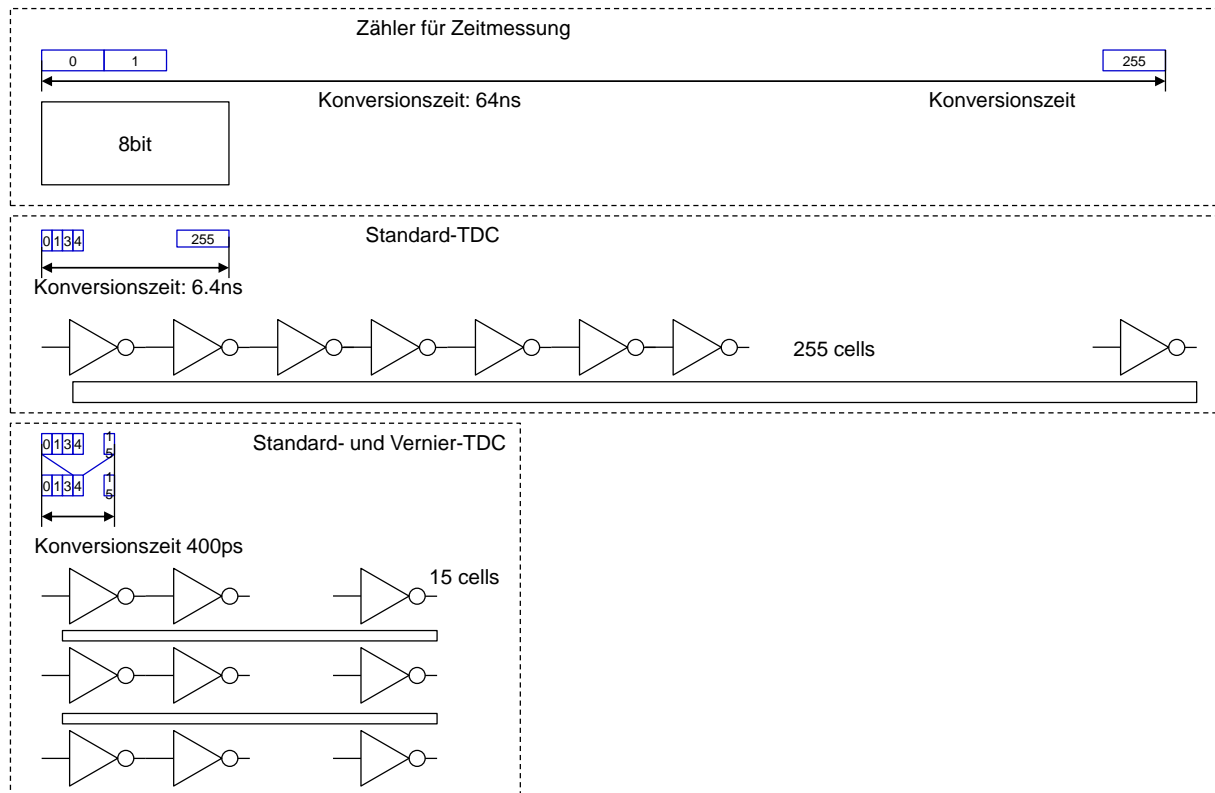


Abbildung 25: Die A/D Konversionszeit von verschiedenen Schaltungen für Zeitmessung

Man kann die Abtastrate weiter erhöhen, indem mehrere ADCs in Parallel geschaltet und ihre track Signale Zeitlich versetzt werden (Abbildung 26). Die zeitbasierten ADCs sind dafür besonders geeignet da sie im Layout klein sind und einen kleinen Stromverbrauch haben. Solche Technik nennt man time interleaving.

Durch time interleaving von zeitbasierten ADCs sind die Abtastraten von 10 G Samples/s bei 8-Bit Auflösung machbar.

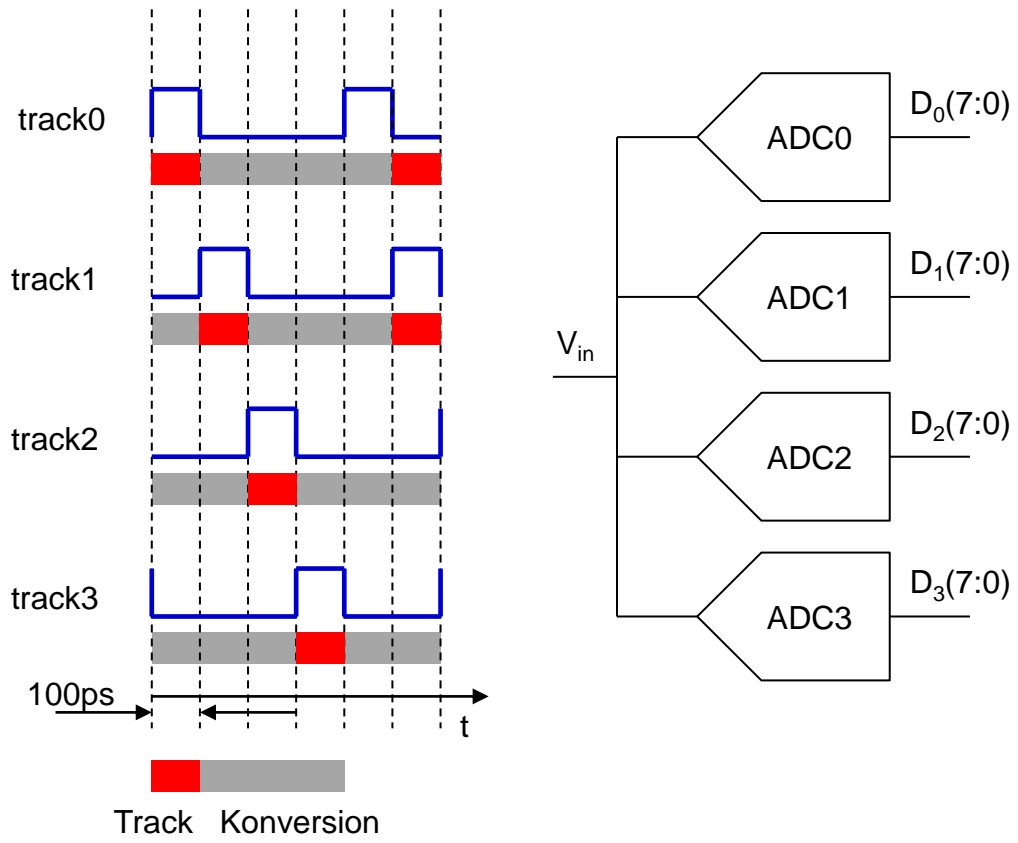


Abbildung 26: Durch time interleaving kann eine hohe Abtastrate erreicht werden

Pipeline-ADC

Eine Übungsaufgabe ist das Design eines Pipeline-ADCs.

Die Grundidee ist der folgende Konversions-Algorithmus:

Das Eingangssignal v_{in} (v_{in} ist Kleinsignal) befindet sich im Bereich $\mp V_{max}$

- 1) Das Eingangssignal wird mit 0 verglichen (mit einem Komparator).
- 2) Wenn das Signal kleiner als 0 ist: wird der digitale Ausgang a auf 0 gesetzt und eine Referenzspannung mit dem Wert $V_{ref} = V_{max}/2$ an den Eingang addiert.
- 3) Wenn das Signal größer als 0 ist wird der digitale Ausgang a auf 1 gesetzt. Die Referenzspannung $V_{ref} = V_{min}/2$ wird vom Eingang abgezogen.
- 4) Das Ergebnis wird mit 2 multipliziert.

Das ergibt die Kennlinie wie in Abbildung 27.

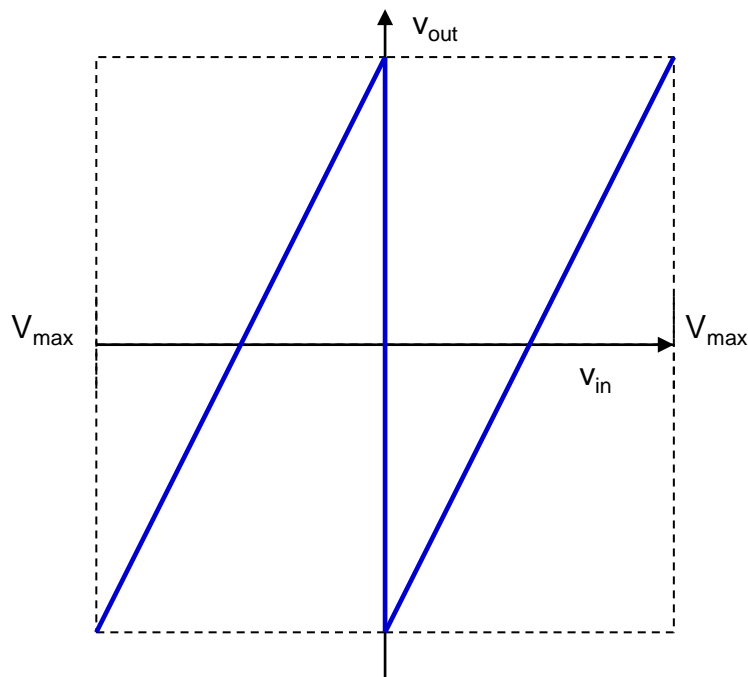


Abbildung 27: Kennlinie nach einem Konversionsschritt eines algorithmischen ADCs

Die Kennlinie hat eine Steigung von 2. Durch die „Faltung“ in der Kennlinie erreichen wir, dass v_{out} im Bereich $(\mp V_{max})$ ist.

Wenn wir den Algorithmus zyklisch n mal wiederholen (in Iterationen 0, 1, 2, $n-1$), bekommen wir das folgende Ergebnis für v_{out} .

$$v_{out} = 2 \left(\dots 2 \left(2 \left(v_{in} - \left(a_{n-1} - \frac{1}{2} \right) V_{max} \right) - \left(a_{n-2} - \frac{1}{2} \right) V_{max} \right) \dots - \left(a_0 - \frac{1}{2} \right) V_{max} \right)$$

Daraus folgt:

$$2^n v_{in} = v_{out} + \sum_{i=0}^{n-1} 2^{i+1} \left(a_i - \frac{1}{2} \right) V_{max}$$

$$2^n v_{in} = v_{out} + \sum_{i=0}^{n-1} 2^{i+1} a_i V_{max} - (2^n - 1) V_{max}$$

$$2^{n-1} \left(\frac{v_{in} + V_{max}}{V_{max}} \right) \sim \frac{1}{2} \frac{v_{out}}{V_{max}} + \sum_{i=0}^{n-1} 2^i a_i \equiv \Delta + D_{out} \quad (16)$$

Wir haben $2^n - 1 \sim 2^n$ angenommen.

Die Zahl:

$$D_{out} = \sum_{i=0}^{n-1} 2^i a_i \in (0, 2^n - 1)$$

ist eine digitale Repräsentation von

$$2^{n-1} \left(\frac{v_{in} + V_{max}}{V_{max}} \right)$$

und dadurch auch von v_{in} .

Faktor

$$\Delta = \frac{1}{2} \frac{v_{out}}{V_{max}}$$

ist Konversionsfehler.

Da v_{out} im Bereich $\mp V_{max}$ liegt, ist der Konversionsfehler im Bereich ∓ 0.5 . Der Konversionsfehler entspricht also einem Bit.

Der Algorithmus kann relativ leicht mit getakteten Schaltungen realisiert werden. Es gibt dabei zwei Möglichkeiten, die in Abbildung 28 gezeigt sind.

Man kann entweder den Analogausgang der Schaltung an eigenen Eingang rückkoppeln und den Algorithmus zyklisch n-mal durchlaufen. Die Konversionszeit ist dann n-Mal länger als Bearbeitungszeit in einer Zelle. Solche ADC Variante nennt man zyklischen ADC (Abbildung 28 oben).

Man kann auch n identische Zellen nacheinander in eine Art Pipeline schalten. Die erste Zelle bekommt als Eingang v_{in} und die nachfolgenden Zellen bekommen als Eingang den Ausgang der vorherigen Zelle (Abbildung 28 unten).

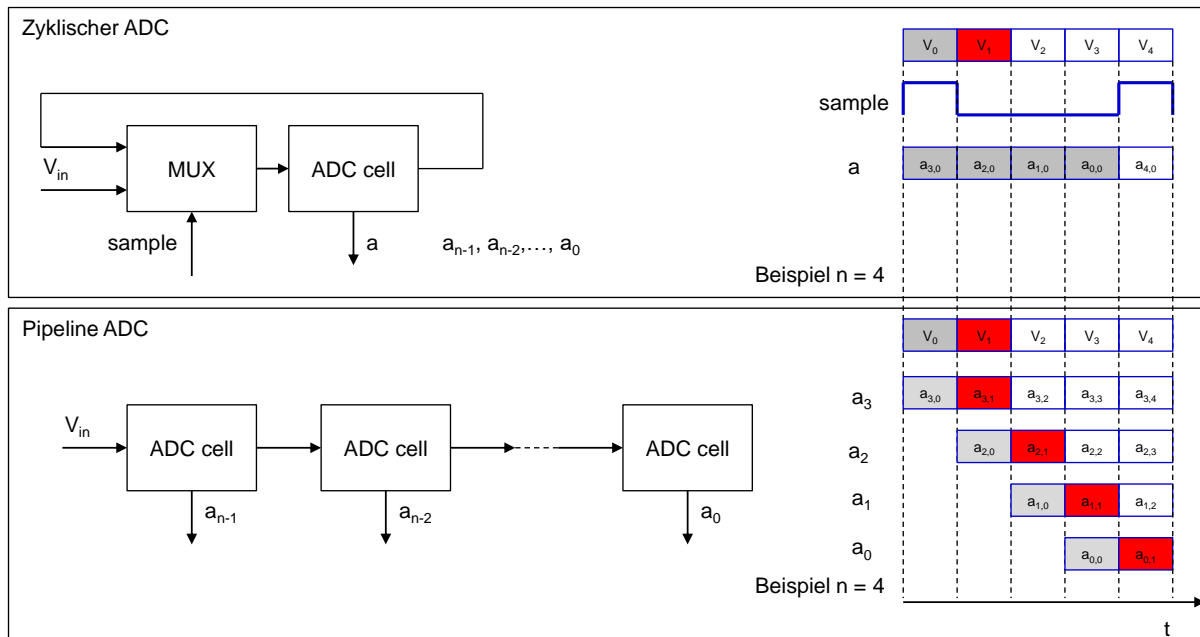


Abbildung 28: Prinzip eines zyklischen und eines Pipeline-ADCs.

Im ersten Takt bearbeitet die erste Zelle das erste Sample V_0 und erzeugt das Bit $a_{n-1,0}$. Im zweiten Takt erzeugt die zweite Zelle das Bit $a_{n-2,0}$. (Die erste Zelle bekommt nun das zweite Sample V_1) Im n -ten Takt erzeugt die n -te Zelle das bit $a_{0,0}$. (Die erste Zelle bekommt nun das n -te Sample). Abbildung 28 rechts zeigt die Signale für den Fall $n = 4$.

Abbildung 29 zeigt einen detaillierteren Blockschaltplan. Die ADC-Zellen werden mit einem Takt Ck angesteuert. Eine ADC-Zelle befindet sich entweder im reset-Zustand ($reset = 1$) oder in Verstärkungszustand ($reset = 0$). Am Ende vom Verstärkungszustand ist das Ergebnis (Digitalausgang a_i) bereit. Beachten wir, dass zwei nachfolgende Zellen in verschiedenen Phasen arbeiten. Eine Zelle befindet sich im reset-Zustand, die andere im Verstärkungszustand.

Mithilfe von Schieberegistern erreichen wir, dass das Konversionsergebnis von einem Sample i ($a_{n-1,i}, \dots, a_{0,i}$) an den Digitalausgang ($D_{n-1}-D_0$) in Parallel ankommt. (Wir nennen es Phasenausgleich.) Es gibt dabei eine Verzögerung von n -Taktzyklen. Wenn das Sample V_0 an den Eingang der ersten ADC-Zelle im Takt 1 ankommt, ist das Konversionsergebnis erst im Takt n bereit. Im Takt $n+1$ ist das Konversionsergebnis vom nächsten Sample V_1 am Digitalausgang bereit. Das heißt, obwohl die Konversionszeit n -Taktperioden ist, entspricht die Sampling-Periode einer Taktperiode. Pipeline ADCs sind dementsprechend relativ schnell. Durch Hinzufügen von zusätzlichen Zellen kann die Auflösung erhöht werden ohne die Sampling-Rate zu ändern. Die Verzögerung von n -Taktperioden ist für die meisten Anwendungen kein Problem.

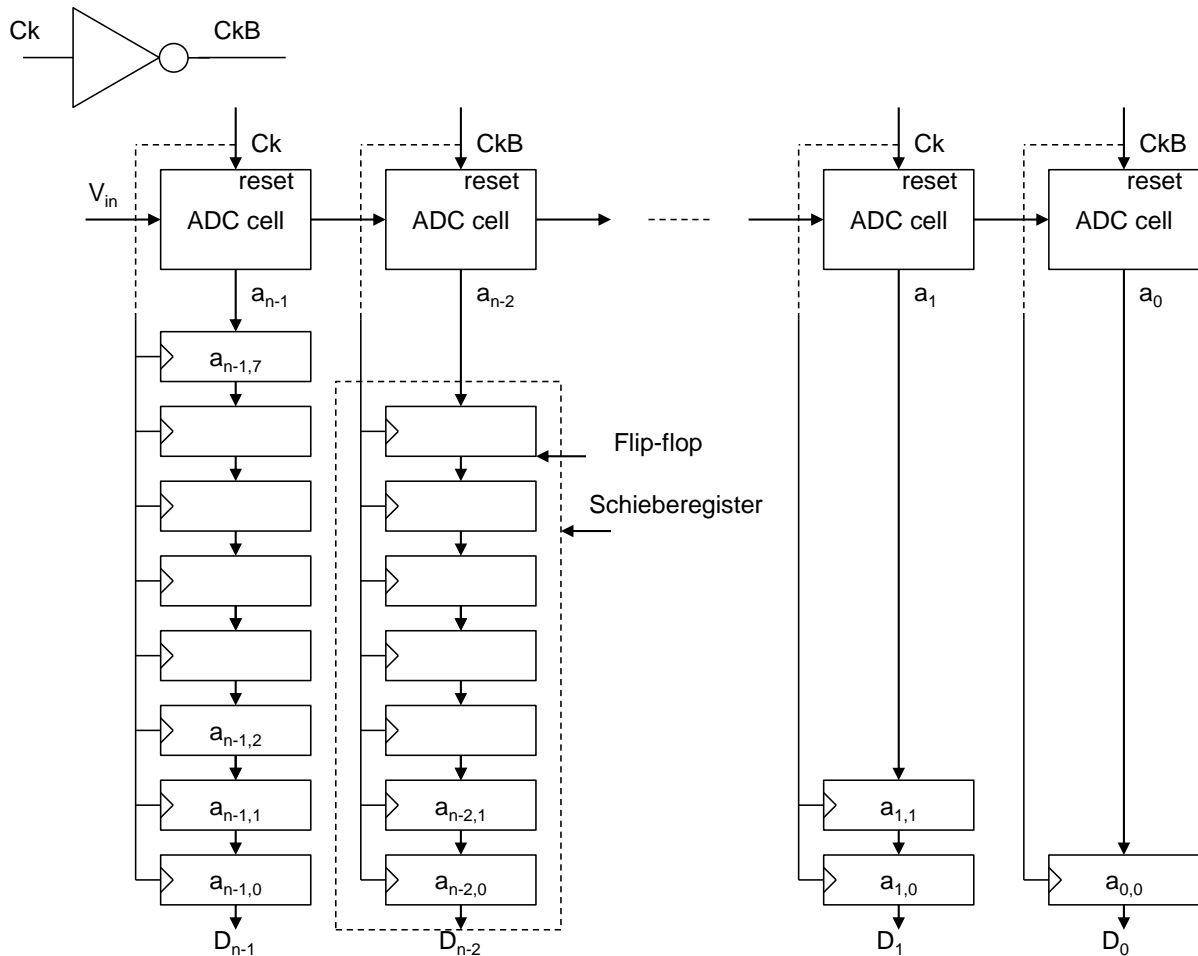


Abbildung 29: Pipeline-ADC mit den Schieberegistern für den Phasenausgleich.

Pipeline ADC mit getakteten Verstärkern (optional)

Beschreiben wir nun den ADC aus der Übung.

Realisierung der ADC Zelle

Die ADC Zelle hat die folgenden Eingänge:

- Reset - Digitaleingang
- In - Analogeingang, hier wird entweder der Ausgang vom getakteten Verstärker oder der Ausgang der vorherigen ADC Zelle angeschlossen
- Out - Analogausgang für die nächste ADC Zelle
- TooHi (= a) - Digitalausgang, generiert durch den Komparator

Die ADC Zelle enthält die folgenden Teile:

- Multiplizierer $\times 2$ mit der Möglichkeit der Addition und Subtraktion der Referenzspannung

– Komparator

Wie funktioniert die ADC Zelle?

Multiplizierer $\times 2$ wird als ein getakteter Verstärker realisiert. Dieser Verstärker hat keinen Multiplexer am Eingang (der Verstärker in Vorlesung 12 hatte den Multiplexer), da die Eingangsspannung bereits die Form einer Spannungsstufe hat. Die Feedbackkapazität C_{fb} ist zweimal kleiner als C_{in} , dadurch wird Verstärkung von zwei erreicht.

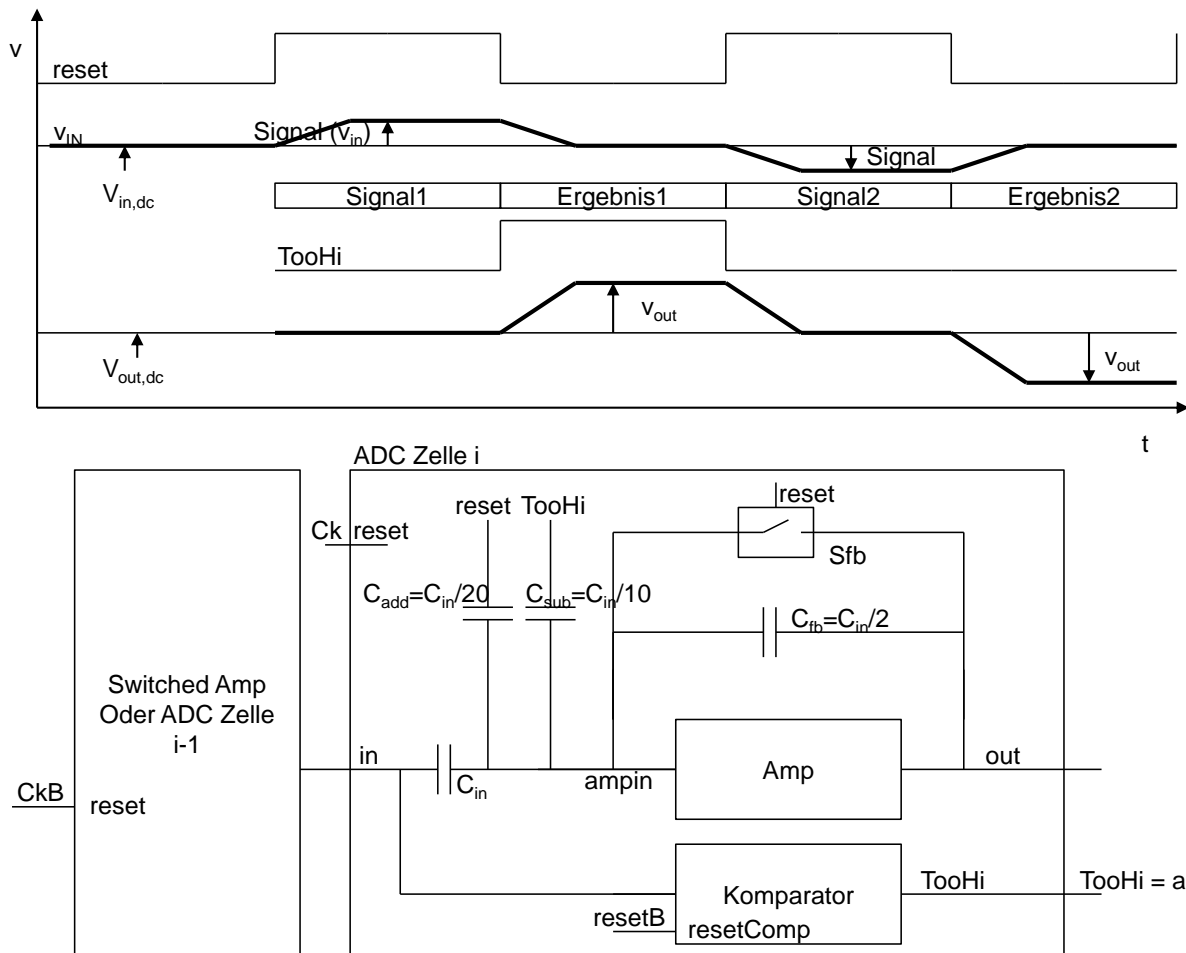


Abbildung 30: Realisierung einer ADC-Zelle

Funktionsweise (Abbildung 30): In der Periode „Signal 1“, wird das Eingangssignal generiert (v_{in}). Der Komparator verstärkt das Signal und bereitet das Ergebnis vor (nicht gezeigt in Abbildung 30). Der Multiplizierer ist im Reset-Zustand. Es gilt:

$$V_{OUT} = V_{out,dc}$$

In der Periode „Ergebnis 1“, erzeugt der Komparator das Ergebnis. V_{in} sinkt vom Wert $v_{in} + V_{in,dc}$ auf $V_{in,dc}$. (Symbol v_{in} ist das Kleinsignal.) Das Signal wird vom Verstärker verstärkt. Falls $v_{in} > 0$ (wie beim Signal 1 in Abbildung 30 der Fall ist) erzeugt der Komparator $TooHi = 1$.

Beachten wir, dass reset und TooHigh an Kondensatoren C_{add} and C_{sub} angeschlossen sind. Am Anfang von Ergebnis 1 Periode ändert sich reset von 1.8 auf 0 und TooHigh von 0 auf 1.8. Die Gesamtladung Q_{in} , die durch C_{in} , C_{add} und C_{sub} fließt ist:

$$Q_{in} = C_{in} \times v_{in} + C_{add} 1.8V - C_{sub} 1.8V = C_{in} \times (v_{in} + 0.05 \times 1.8V - 0.1 \times 1.8V)$$

Daraus folgt:

$$v_{out} = \frac{Q_{in}}{C_{fb}} = 2(v_{in} - 0.09) \quad (17)$$

Falls $v_{out} > 0$ (s. „Signal 2“), TooHi wird nicht generiert. Es gilt dann:

$$Q_{in} = C_{in} \times v_{in} - C_{add} 1.8V = C_{in} \times (v_{in} + 0.05 \times 1.8V)$$

und

$$v_{out} = \frac{Q_{in}}{C_{fb}} = 2(v_{in} + 0.09) \quad (18)$$

Gleichungen (17) und (18) entsprechen dem Konversionsalgorithmus und erzeugen die Kennlinie von Abbildung 27.

Das Eingangssignal sollte im Bereich $V_{in,dc} \mp 2 \times 90 \text{ mV}$ ($\mp 180 \text{ mV}$) sein.

Es gilt in unserem Fall $V_{in,dc} = V_{out,dc}$.

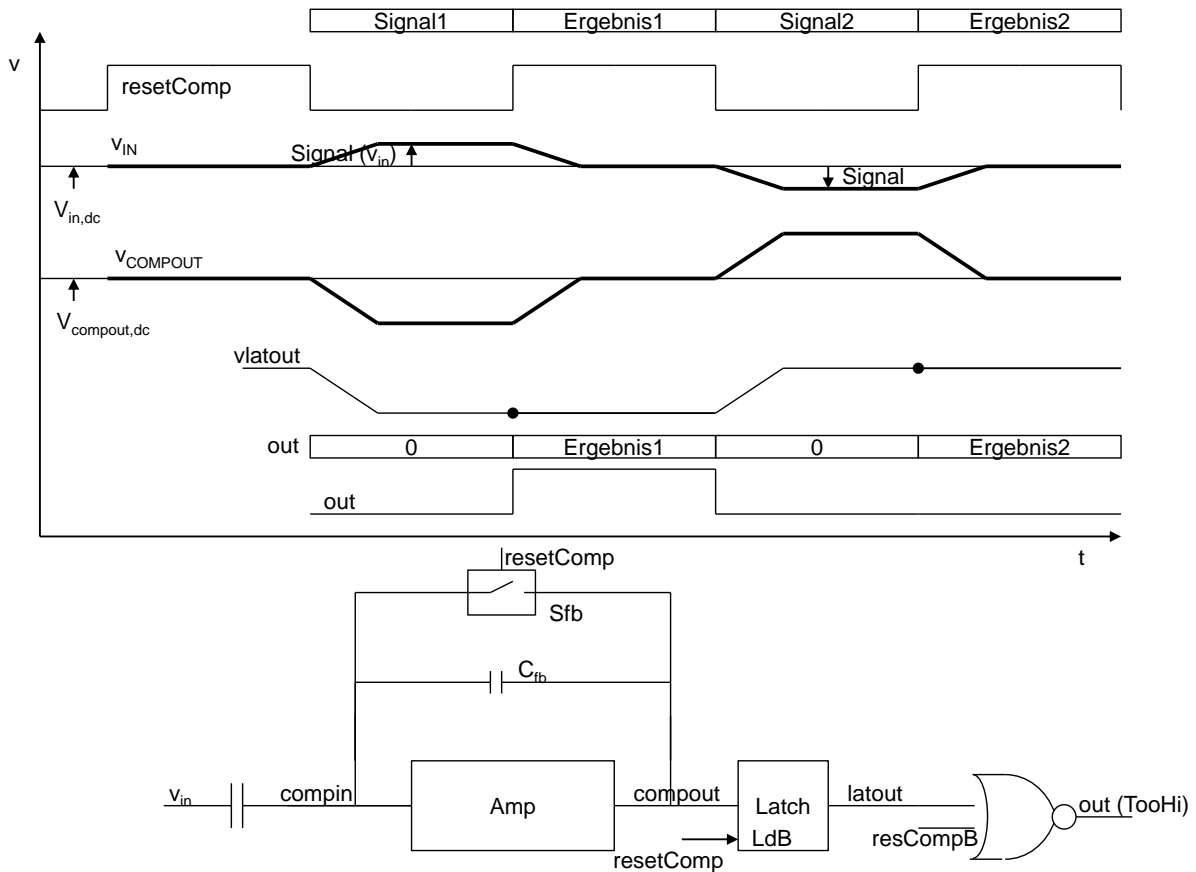


Abbildung 31: Realisierung des Komparators

Realisierung des Komparators

Abbildung 31 zeigt den Schaltplan des Komparators.

Eingang des Komparators ist an v_{in} der ADC-Zelle angeschlossen.

Komparator ist auch als ein getakteter Verstärker realisiert, nur mit viel kleinerer C_{fb} und größerer Verstärkung.

Beachten wir, dass das Reset-Signal des Komparators ($resetComp$) an das negierte Reset-Signal des Verstärkers ($resetB$) angeschlossen ist (Abbildung 31).

Die Latch-Schaltung (Latch) ist transparent (im Schreibzustand) für $resetComp = 0\text{ V}$ und im Speicherzustand für $resetComp = 1.8\text{ V}$. Dadurch ist das Ergebnis des Komparators auch in der Phase $resetComp = 1.8\text{ V}$ am Ausgang vorhanden.

Wenn das Eingangssignal (v_{in}) positiv ist (Periode Signal 1), wird $v_{COMPOUT}$ niedrig.

Wenn das Eingangssignal (v_{in}) negativ ist (Periode Signal 2), wird $v_{COMPOUT}$ hoch.

$V_{compout}$ is also negiert.

Komparator Ergebnis ist in der Ergebnis-Periode ($resetComp = 1.8$) nach dem jeweiligen Signal am Latch-Ausgang $latout$ vorhanden.

Eine Nor-Schaltung gewährleistet, dass der Ausgang out null ist für $\text{resetComp} = 0 \text{ V}$ (Periode Signal). Am Anfang der Signal-Periode, ist der logische Pegel am Ausgang vom Latch nicht definiert. Nor soll verhindern, dass die nachfolgenden digitalen Schaltungen einen undefinierten logischen Pegel empfangen. Die Nor-Schaltung macht auch eine Negation des Signals, was notwendig ist, da V_{compout} negiert ist.

Anhang

Grundlegende CMOS Digitalschaltungen

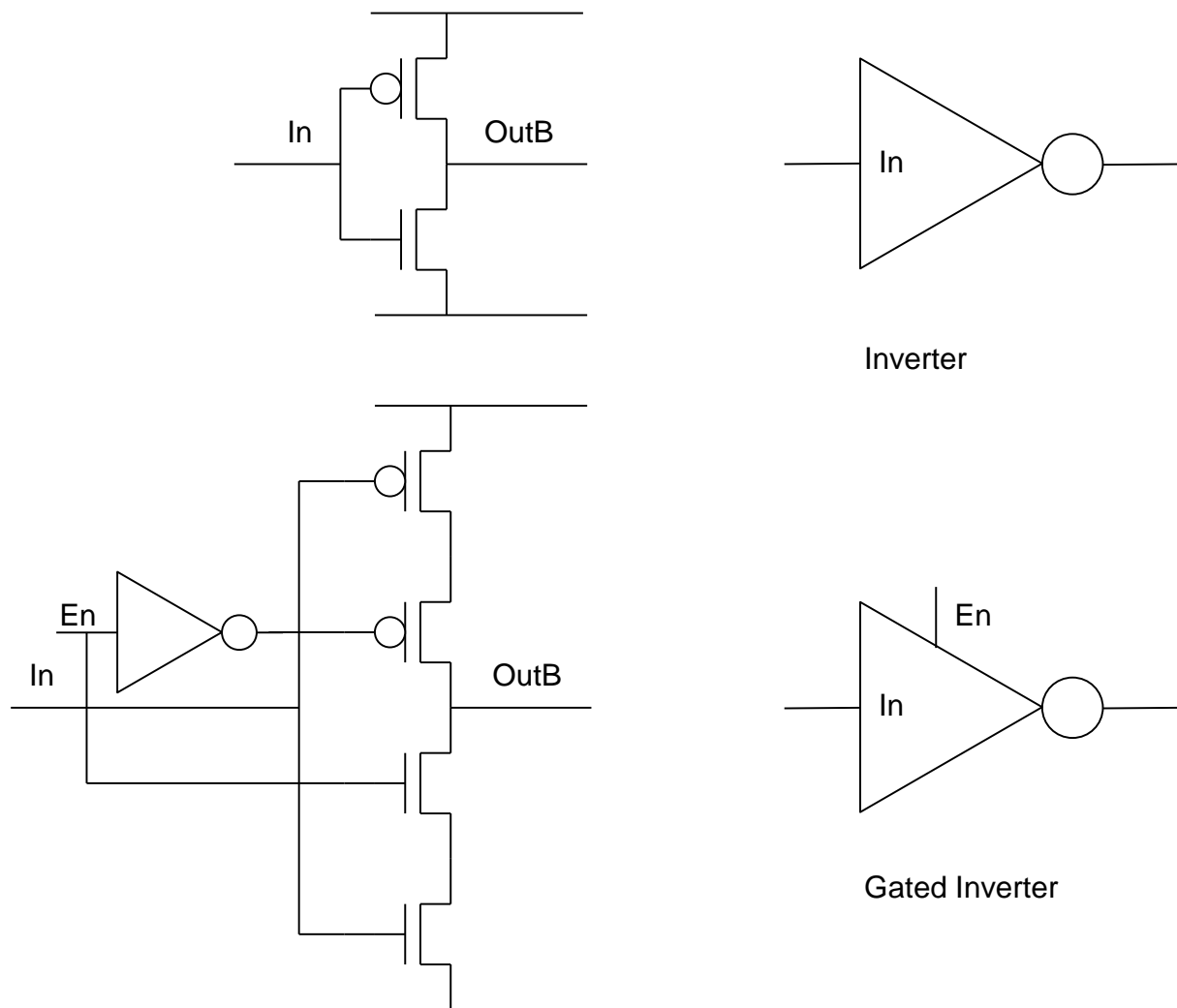


Abbildung 32: Inverter und Gated-Inverter

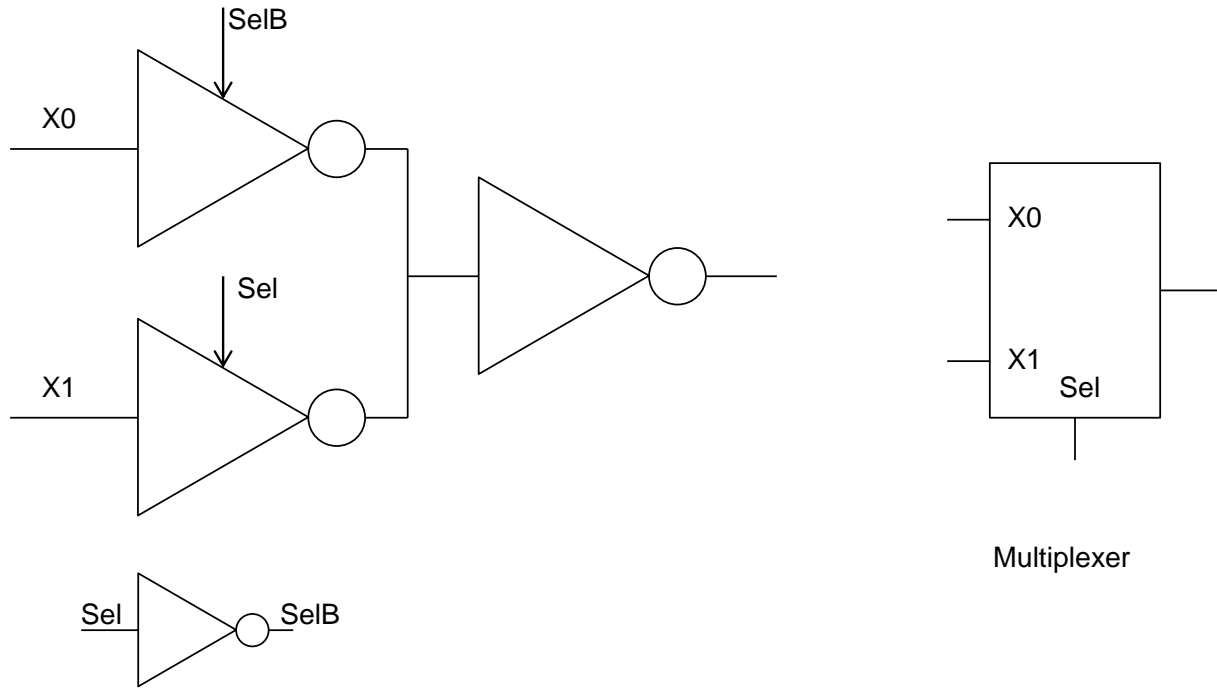


Abbildung 33: Multiplexer

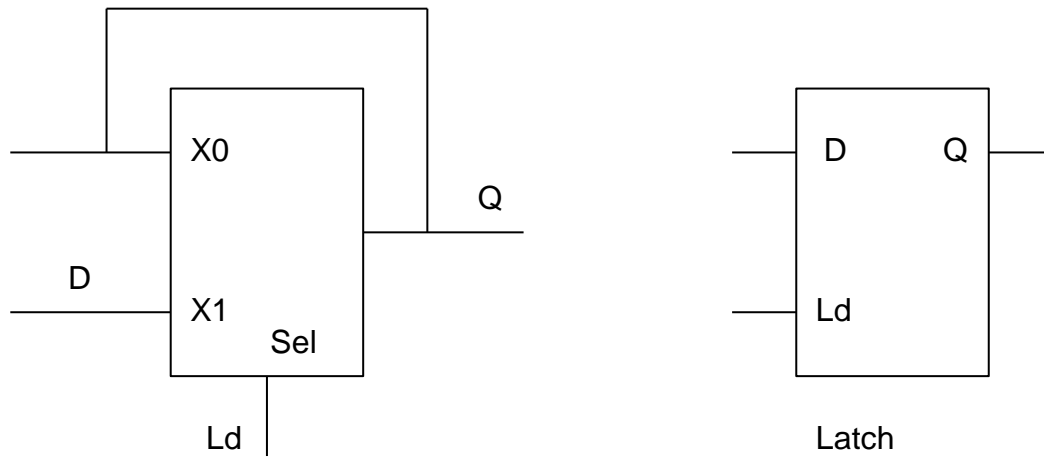


Abbildung 34: Latch

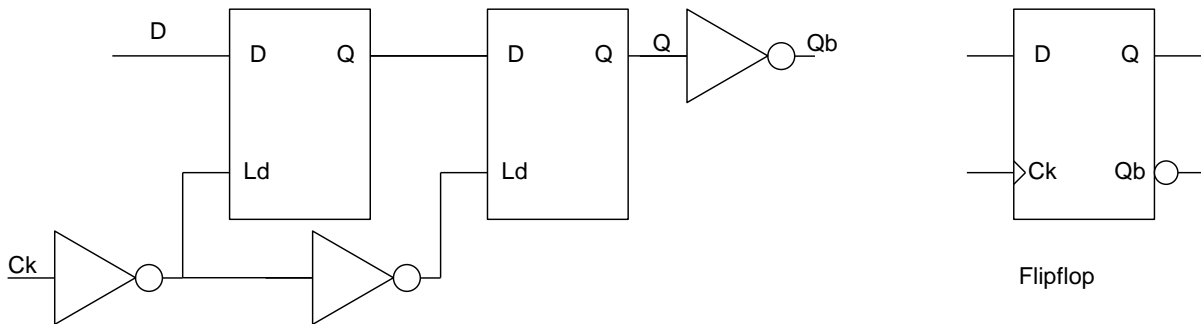


Abbildung 35: Flip-flop