## • IEEE 754 Darstellung

Schritt 1: Normalisieren

$$1,x \ldots x * 2^{y \ldots y}$$

Schritt 2: Biased Exponent berechnen

$$1,x \ldots x * 2^{y \ldots y + Bias}$$

Bias
Single-Precision: 127
Double-Precision: 1023

$$z = \begin{cases} (-1)^S * (1 + Significand) * 2^{E_B}, falls\ E \neq 0 \\ (-1)^S * Significand * 2^{E_B}, falls\ E = 0 \end{cases}$$

Falls $E = 0$ und $Siginficand \neq 0$ => NaN
Falls $E = 255$ und $Siginficand = 0$ => Unendlich
Falls $E = 255$ und $Siginficand \neq 0$ => NaN


1Bit 8 Bit 23 Bit

Schritt 1: Normalisieren

$1,0 * 2^0$

Schritt 2: Biased Exponent berechnen

$1,0 * 2^{0+127}$

$$z = \begin{cases} (-1)^S \times (1 + Significand) * 2^{E_B}, & falls\ E_B \neq 0 \\ (-1)^S \times Significand * 2^{E_B}, & falls\ E_B = 0 \end{cases}$$

| Sign | Exponent | Significand |
|---|---|---|
| 0 | 01111111 | 00000000000000000000000 |

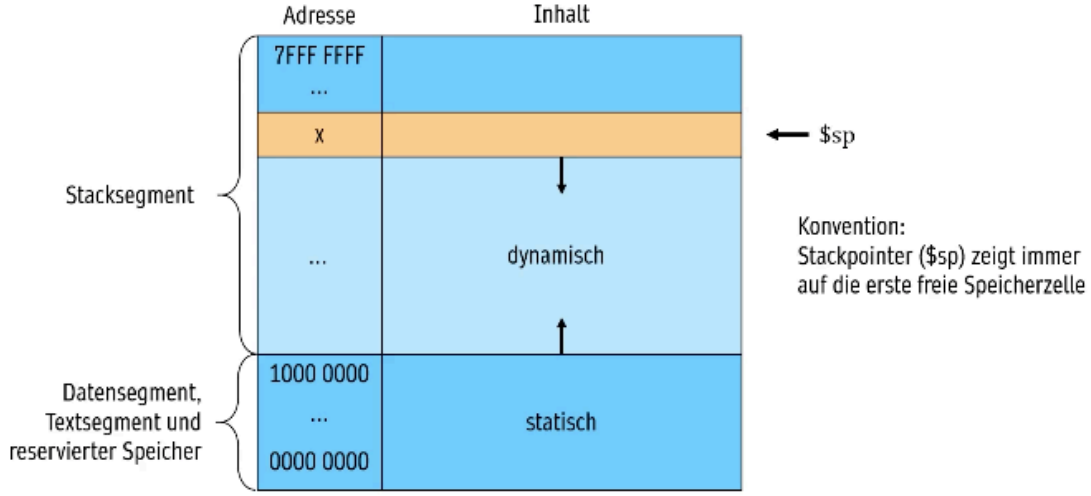
## • Call-by-Value und Call-by-Reference

- call by value: Der Wert einer Variablen wird referenziert
- call by reference: Die Adresse der Variablen wird referenziert

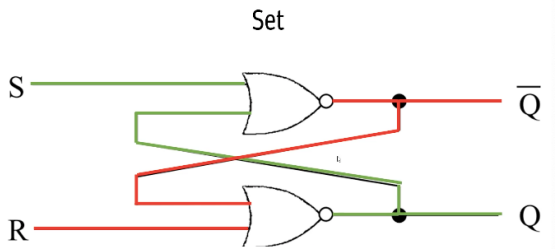
```
int cbv_addition(int y, int z){
    return y + z;
}
```

```
void cbr_addition(int *x, int y*, int *z){
    *x = *y + *z
}
```

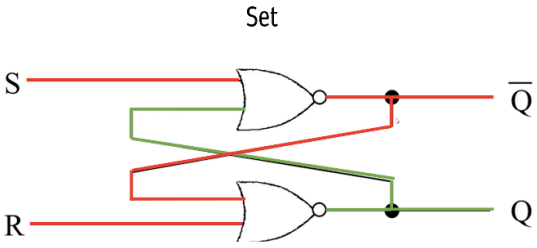
## • Stack



- **SR-Latch**

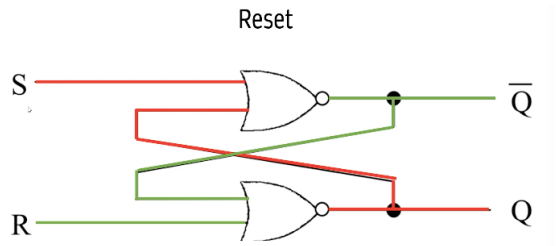


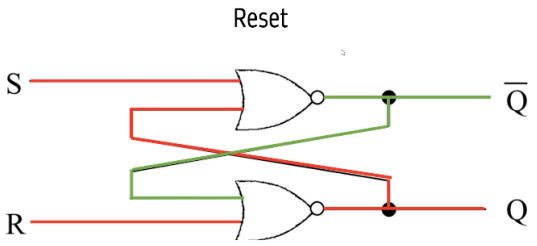

Für S = 1 und R = 0 ergibt sich Q = 1 und $\overline{Q}$ = 0



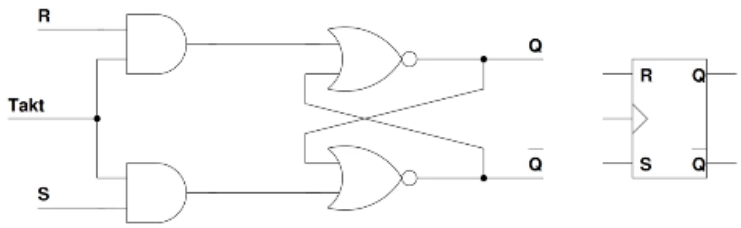

Auch für S = 0 und R = 0 wird der Zustand von Q beibehalten

Genau so wie bei Reset:




Beispiel:





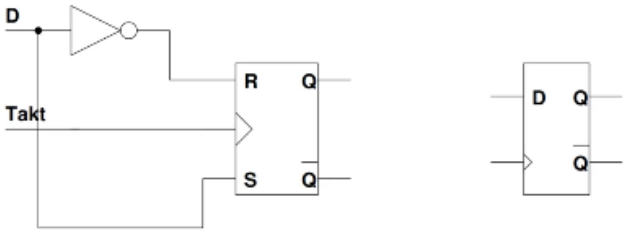


Da das Taktsignal C jeweils mir den anderen Eingängen UND-verknüpft wird, reicht es die Kombinationen der Bitmuster der Eingänge ohne das Taktsignal zu betrachten. Wenn der Takt 0 ist, entspricht das dem Speichern eines Bits.

Die Tabellen weist einen zeitlichen Verlauf von oben nach unten auf!

| S | R | C | Q | $\overline{Q}$ | Kommentar |
|---|---|---|---|---|---|
| 0 | 0 | 1 | Q* | $\overline{Q}$* | Speichern |
| 0 | 1 | 1 | 0 | 1 | Reset |
| 1 | 0 | 1 | 1 | 0 | Set |
| 1 | 1 | 1 | - | - | Unzulässig |
| D | D | 0 | Q* | $\overline{Q}$* | Speichern |

Bei S=1, R=1 und C=1 kommt es zu Race Condition. Je nach dem welches früher Schaltet. Daher ist es „unzulässig". Daher wird D hinzugefügt.



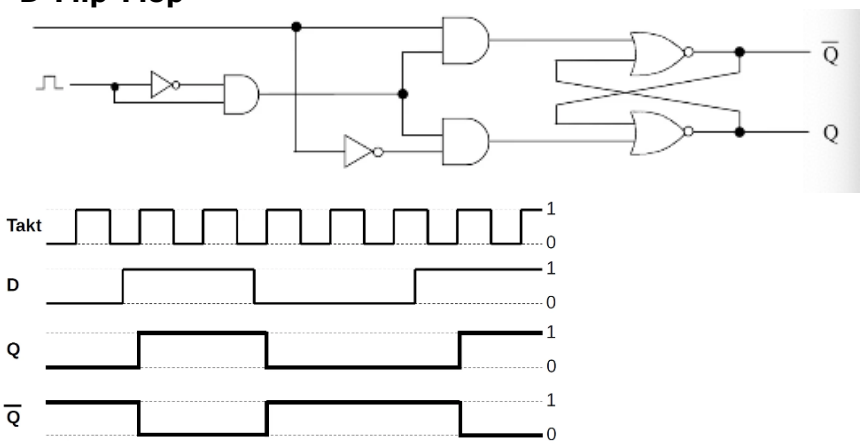




| D | C | Q | $\overline{Q}$ | Kommentar |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | Set |
| 0 | 1 | 0 | 1 | Reset |
| D | 0 | Q* | $\overline{Q}$* | Speichern |

- **Flip-Flops**
  - Eine Variante der Latch-Bauelemente geht davon aus, dass der Zustandsübergang nicht eintritt, wenn der Taktgeber 1 ist, sondern von 0 auf 1 (steigende Flanke) oder von 1 auf 0 (fallende Flanke) übergeht. Solche Elemente heißen Flip-Flops.
  - Eine Flip-Flop-Schaltung wird flankengesteuert genannt (edge-triggered), eine Latch-Schaltung wird pegelgesteuert (level-triggered) genannt.

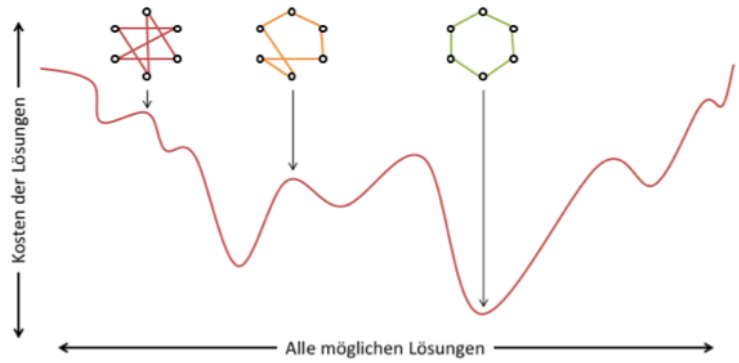
- **D-Flip-Flop**




- **Quantenannealing**

(ausführlich im Übungsblatt 11 erklärt)

- **Annealing als Optimierungsmethode**




- **Hamming Codes**

Übertragung von Daten über physische Kanäle (Kabel etc.) ist fehleranfällig. Indem man ein einzelnes Bit, das Paritätsbit, zu jedem Datenpaket hinzufügt, kann man Ein-Bit-Fehler entdecken. Mit einem einzelnen Paritätsbit ist es allerdings nicht möglich herauszufinden welches Bit fehlerhaft ist. Wenn man also das fehlerhafte Bit erkennen möchte, benötigt man mehr Paritätsbits. Das folgende Verfahren wurde mit dem Ziel Ein-Bit Datenfehler, die durch unzuverlässige Übertragungskanäle verursacht wurden, zu erkennen und zu korrigieren. Der Trick besteht darin zusätzliche Paritätsbits in die Datenpakete einzufügen und so ein Hamming Codewort zu bilden. Das Hamming Codewort besteht aus den eigentlichen Datenbits, die übertragen werden sollen, und einigen Paraitätsbits, die an strategischen Punkten eingefügt wurden. Die Anzahl der benötigten Paritätsbits ist abhängig von der Anzahl der Datenbits:

| Daten Bits | 8 | 16 | 32 | 64 | 128 |
|---|---|---|---|---|---|
| Paritäts Bits | 4 | 5 | 6 | 7 | 8 |
| Codewort | 12 | 21 | 38 | 71 | 136 |

Allgemein gilt: Für Daten den Länge $2^n$ Bits werden $n + 1$ Paritätsbits eingefügt, um das Codewort zu bilden.