

Rechnertechnik

Prof. Dr. rer. nat. Peter Gerwinski

27. April 2021

Rechnertechnik

1 Einführung

2 Vom Schaltkreis zum Computer

2.1 Logik-Schaltkreise

2.2 Binärdarstellung von Zahlen

2.3 Vom Logik-Schaltkreis zum Addierer

2.4 Negative Zahlen

2.5 Vom Addierer zum Computer

...

3 Architekturmerkmale von Prozessoren

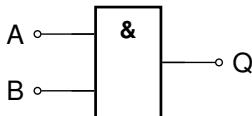
4 Der CPU-Stack

...

2.5 Vom Addierer zum Computer

1-Bit-Multiplikierer = Und-Verknüpfung

| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



n-Bit-Multiplikierer: „schriftlich“ multiplizieren

Beispiel: $13 \cdot 5$

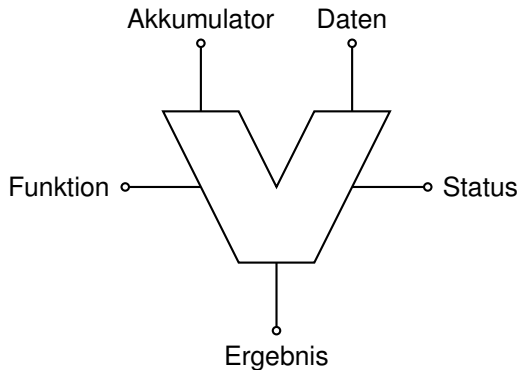
$$\begin{array}{r} 1101 \cdot 101 \\ \hline 1101 \\ 0 \\ 1101 \\ \hline 1111 \\ \hline 100001 \end{array}$$

Multiplikier-Schaltkreis
für zwei 2-Bit-Zahlen:

(gestern)

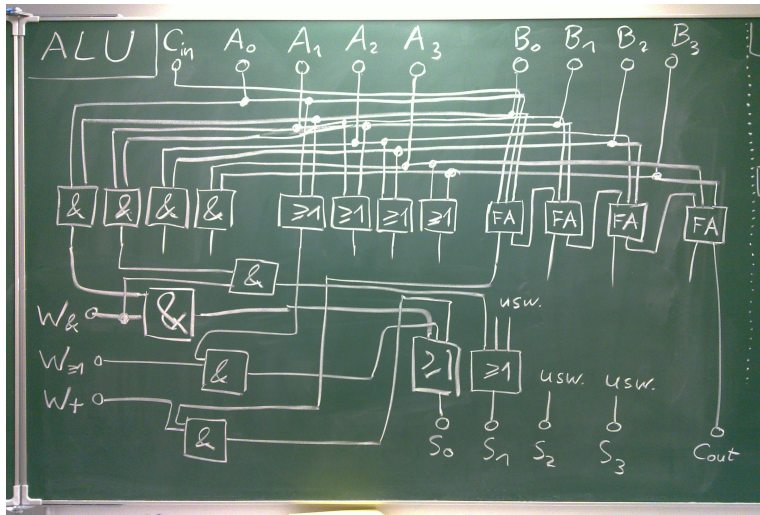
2.5 Vom Addierer zum Computer

Schaltkreis, der wahlweise eine von mehreren Verknüpfungen durchführt:
arithmetisch-logische Einheit – arithmetic logic unit (ALU)



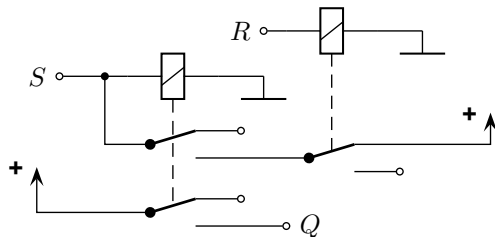
2.5 Vom Addierer zum Computer

Schaltkreis, der wahlweise eine von mehreren Verknüpfungen durchführt:
arithmetisch-logische Einheit – arithmetic logic unit (ALU)



2.5 Vom Addierer zum Computer

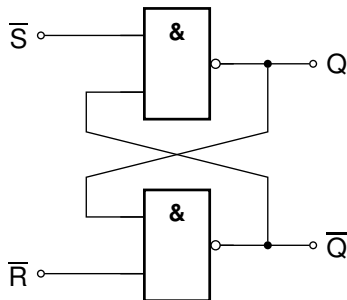
Information speichern



Selbsthalteschaltung
1-Bit-Speicherzelle

2.5 Vom Addierer zum Computer

Information speichern



Bistabile Kippstufe – Bistabiler Multivibrator – Flip-Flop
1-Bit-Speicherzelle

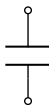
2.5 Vom Addierer zum Computer

Information speichern

Kondensator

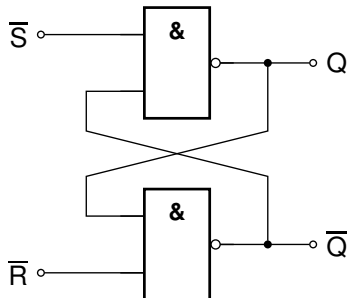
dynamische 1-Bit-Speicherzelle

→ benötigt *Refresh*-Schaltung



Flip-Flop

statische 1-Bit-Speicherzelle



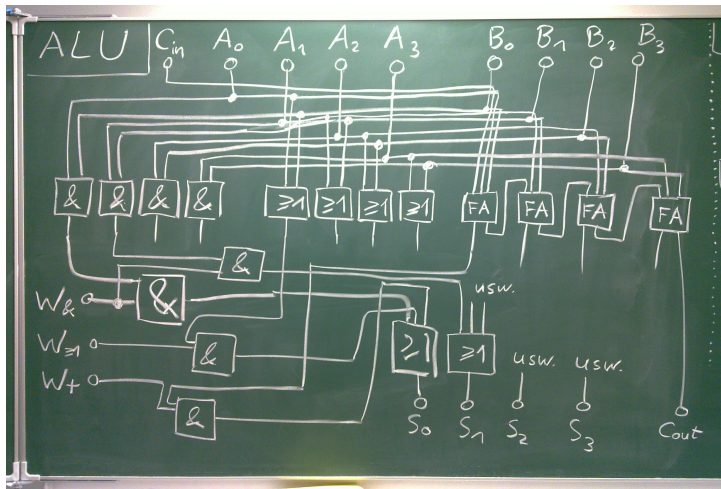
2.5 Vom Addierer zum Computer

Bau eines Turing-vollständigen Computers

2.5 Vom Addierer zum Computer

Bau eines Turing-vollständigen Computers

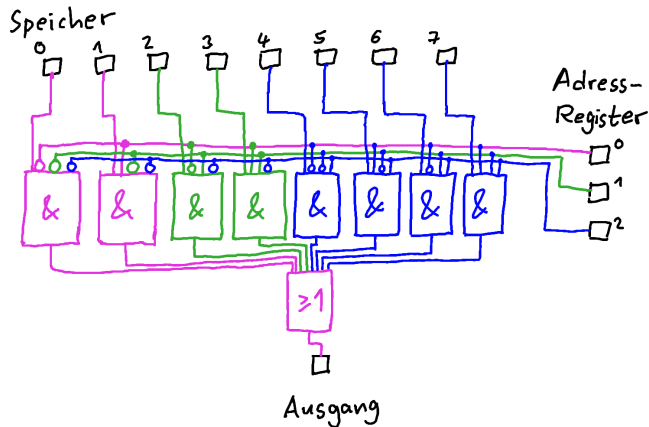
- Rechenwerk: Arithmetisch-logische Einheit (ALU)



2.5 Vom Addierer zum Computer

Bau eines Turing-vollständigen Computers

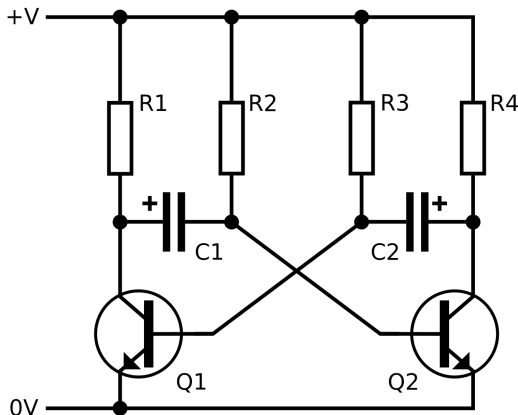
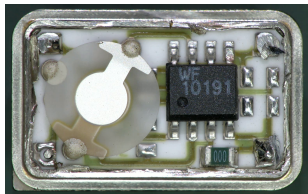
- Rechenwerk: Arithmetisch-logische Einheit (ALU)
- Speicher: Register, adressierbarer Hauptspeicher



2.5 Vom Addierer zum Computer

Bau eines Turing-vollständigen Computers

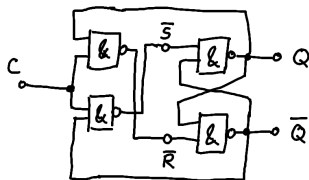
- Rechenwerk: Arithmetisch-logische Einheit (ALU)
- Speicher: Register, adressierbarer Hauptspeicher
- Takt:



2.5 Vom Addierer zum Computer

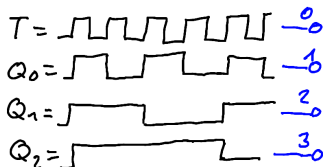
Bau eines Turing-vollständigen Computers

- Rechenwerk: Arithmetisch-logische Einheit (ALU)
- Speicher: Register, adressierbarer Hauptspeicher
- Takt: Speicher durchgehen



„Stromstoßschalter“

Funktions-Eingänge der ALU



Adress-Register
Speicher Bit für Bit auslesen



Zähler-Schaltung

2.5 Vom Addierer zum Computer

Bau eines Turing-vollständigen Computers

- Rechenwerk: Arithmetisch-logische Einheit (ALU)
- Speicher: Register, adressierbarer Hauptspeicher
- Takt: Speicher durchgehen und Befehle abarbeiten
 - Register-Ladebefehl:
„Lade Speicherzelle 42 an den A-Eingang der ALU.“
 - Rechenbefehl:
„Berechne die Summe der beiden Register an der ALU.“
 - Register-Speicherbefehl:
„Speichere den Status-Ausgang der ALU in Speicherzelle 137.“

2.5 Vom Addierer zum Computer

Bau eines Turing-vollständigen Computers

- Rechenwerk: Arithmetisch-logische Einheit (ALU)
- Speicher: Register, adressierbarer Hauptspeicher
- Takt: Speicher durchgehen und Befehle abarbeiten
 - Register-Ladebefehl:
„Lade Speicherzelle 42 an den A-Eingang der ALU.“
 - Rechenbefehl:
„Berechne die Summe der beiden Register an der ALU.“
 - Register-Speicherbefehl:
„Speichere den Status-Ausgang der ALU in Speicherzelle 137.“
 - Sprungbefehl:
„Lade den Inhalt der Speicherzelle 1117
in das Adressregister für den nächsten Befehl.“

2.5 Vom Addierer zum Computer

Bau eines Turing-vollständigen Computers

- Rechenwerk: Arithmetisch-logische Einheit (ALU)
- Speicher: Register, adressierbarer Hauptspeicher
- Takt: Speicher durchgehen und Befehle abarbeiten
 - Register-Ladebefehl:
„Lade Speicherzelle 42 an den A-Eingang der ALU.“
 - Rechenbefehl:
„Berechne die Summe der beiden Register an der ALU.“
 - Register-Speicherbefehl:
„Speichere den Status-Ausgang der ALU in Speicherzelle 137.“
 - Sprungbefehl:
„Lade den Inhalt der Speicherzelle 1117
in das Adressregister für den nächsten Befehl.“
 - Bedingter Sprungbefehl:
„Wenn das C_{out} -Bit der ALU den Wert 1 hat,
springe nach Speicherzelle 23.“

2.5 Vom Addierer zum Computer

Bau eines Turing-vollständigen Computers

- Rechenwerk: Arithmetisch-logische Einheit (ALU)
- Speicher: Register, adressierbarer Hauptspeicher
- Takt: Speicher durchgehen und Befehle abarbeiten
- Peripherie: Kommunikation mit der Außenwelt
 - Output-Port:
Ausgang einer Speicherzelle nach draußen führen
 - Input-Port:
Anstelle einer Speicherzelle einen eingehenden Draht abfragen

2.5 Vom Addierer zum Computer

Bau eines Turing-vollständigen Computers

- Rechenwerk: Arithmetisch-logische Einheit (ALU)
- Speicher: Register, adressierbarer Hauptspeicher
- Takt: Speicher durchgehen und Befehle abarbeiten
- Peripherie: Kommunikation mit der Außenwelt

→ in Maschinensprache programmierbar

Rechnertechnik

1 Einführung

2 Vom Schaltkreis zum Computer

2.1 Logik-Schaltkreise

2.2 Binärdarstellung von Zahlen

2.3 Vom Logik-Schaltkreis zum Addierer

2.4 Negative Zahlen

2.5 Vom Addierer zum Computer

2.6 Computer-Sprachen

2.7 Struktur von Assembler-Programmen

3 Architekturmerkmale von Prozessoren

4 Der CPU-Stack

...