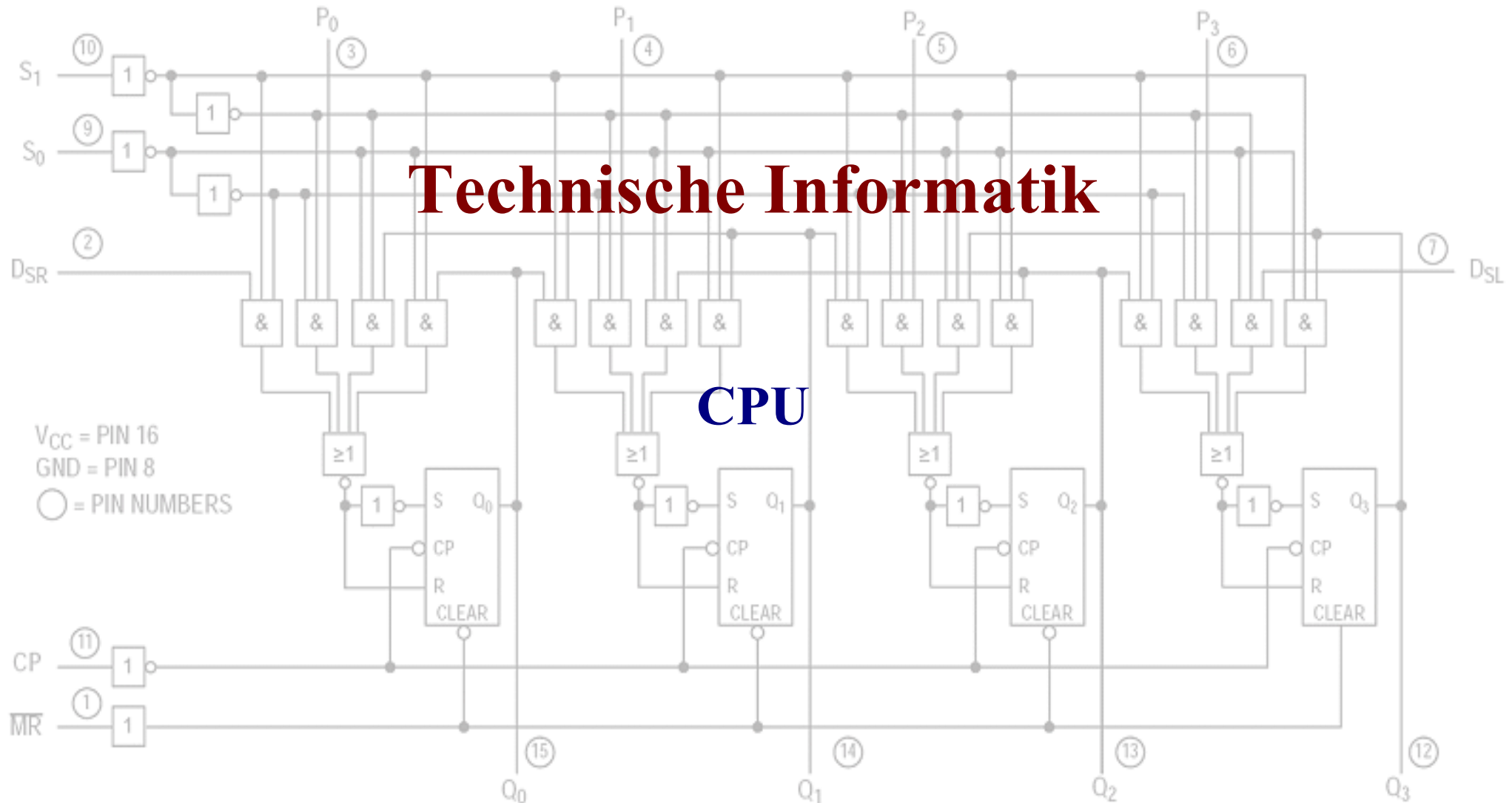


# Technische Informatik

## CPU



# Was ist eine „Architektur“ ?

---

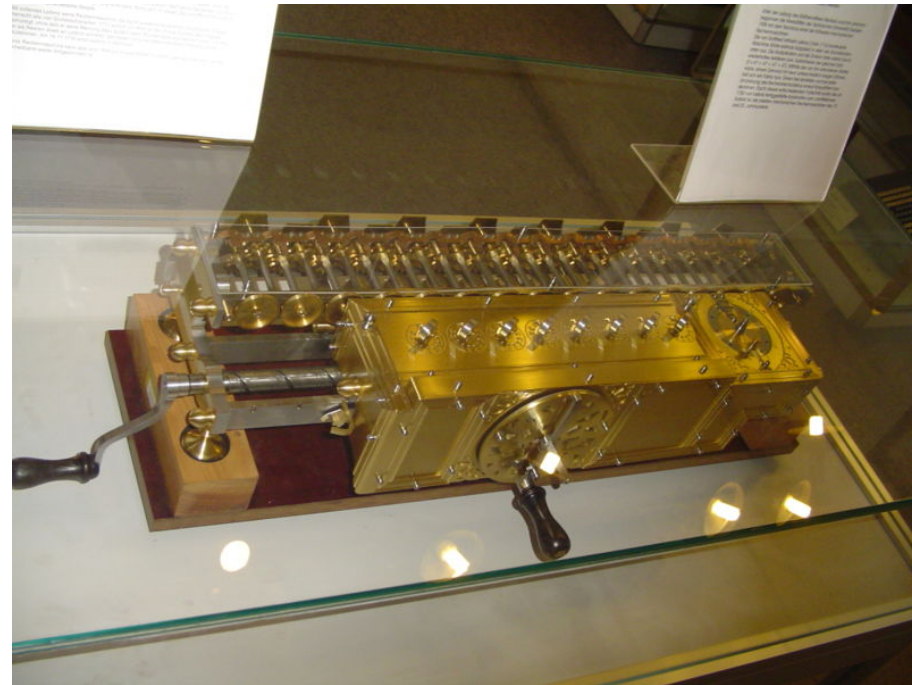
- *Architekt* von Architectus (lat.) oder Architékton (gr.) = Baumeister
- Wortbestandteile *archós* = Anführer und *technikós* = kunstvoll, fachmännisch
- abgeleiteter Begriff *Architektur*: „Baustil“, „Baukunst“
- unterschieden vom Handwerker = Implementator  
(vergl. Maurer - Bauingenieur - Architekt)

# Was ist eine „Architektur“ ?

- The **overall design on system level**, the kinds of instructions available, the kind of data used, the mechanisms available for altering the flow of control, the memory organization and addressing, .. (*Tanenbaum, 1976*)
- The „**blueprint**“ used to built a machine; it is the instruction set and the I/O connection capabilities (*Lipovski, 1978*)
- Study of the **structure**, the **behavior** and **design** of computers (*Hayes, 1978*)
- **Art, science and engineering** of computer structure, organisation, implementation and performance evaluation (*Baer, 1980*)

# Historisches

- **Wilhelm Schickart** (1592 – 1635)
  - Baute 1623 in Tübingen eine Maschine für die vier Grundrechenarten (Vortriebsmaschine )
- **Blase Pascal** (1623 – 1662)
  - Konstruierte eine mechanische Rechenmaschine für Berechnung zur Steuereintreibungen
- **Gottfried Wilhelm Leibniz** (1646 – 1716)
  - Baute 1672 eine Rechenmaschine zum Multiplizieren und Dividieren inkl. Stellenverschiebungen und schlug das Rechnen im Dualsystem vor
- **Joseph Marie Jacquard** (1805)
  - Konstruierte eine Webstuhlsteuerung für vorgegebene Muster mittels Lochkarten



Quelle: de.wikipedia.org

# Historisches

- *Charles Babbage* (1792-1871)
  - Baute um 1840 die "Analytical Engine" = erster frei programmierbarer Computer ?
- *Augusta Ada Countess of Lovelace* (1815-1852)
  - Die erste Informatikerin?
  - Assistentin von Babbage und Tochter des Dracula-Erfinders Lord Byron definierte erstmals, was eine Maschinenoperation ist und dass sie nicht auf die Manipulation von Zahlen beschränkt ist
  - Sie entwarf vielleicht zum ersten Mal in der Geschichte ein Maschinenprogramm: den Plan, mit dem die "Analytic Engine" Bernoulli-Zahlen berechnen konnte
  - ... hätte können, wenn "Analytical Engine" jemals komplett funktionstüchtig gebaut worden wäre ...



- **Konrad Zuse** (1910-1995)

- Baute mit der „Z3“ den ersten Computer der Welt
- Entwickelte die computergerechte Darstellung von Fließkommazahlen ...
- ... und die erste universelle Programmiersprache „Plankalkül“



- **William B. Shockley, John Bardeen** und **Walter Brattain**

- Erfinden 1947 den Transistor und legen damit die Grundlage für die heutige Computertechnik
- 1955 wird der erste Computer in Transistortechnik gebaut
- Erste integrierte Schaltkreise ab 1964
- Intel baut 1971 den ersten Mikroprozessor (4004) in Serienfertigung

# Historisches

- John von Neumann (1903-1957)
  - Mathematiker & Philosoph
  - Beiträge zur Logik und Funktonanalysis sowie Quantenmechanik
  - Manhattan-Projekt (Los Alamos, USA)
- Informatik:
  - Grundlagen zur Rechnerarchitektur
  - Verwendung von Binärcodes
  - Programmierung mit Unterprogrammen



Von LANL -  
<http://www.lanl.gov/history/atomicbomb/images/NeumannL.GIF> (Archivkopie in derWayback Machine)

# Struktur-Prinzip

- Von-Neumann-Prinzipien
  - Ein Rechner besteht aus vier Werken:
    - Hauptspeicher für Programme und Daten
    - Steuerwerk zur Interpretation der Programme
    - Rechenwerk zur Ausführung arithmetischer/logischer Operationen
    - Ein- und Ausgabewerk, eventuell verbunden mit einem Sekundärspeicher
  - Die Rechnerstruktur ist unabhängig von der Aufgabe
  - Programme und Daten stehen im selben Speicher
  - Der Hauptspeicher ist in fortlaufend adressierten Zellen eingeteilt
  - Ein Programm ist eine Folge von Befehlen, die in der gegebenen Reihenfolge ausgeführt werden (Prinzip der Sequenz)
  - Das Prinzip der Sequenz kann durch bedingte oder unbedingte Sprungbefehle außer Kraft gesetzt werden. Bedingte Sprünge sind von gespeicherten Werten abhängig.
  - Befehle und Daten werden binär dargestellt.

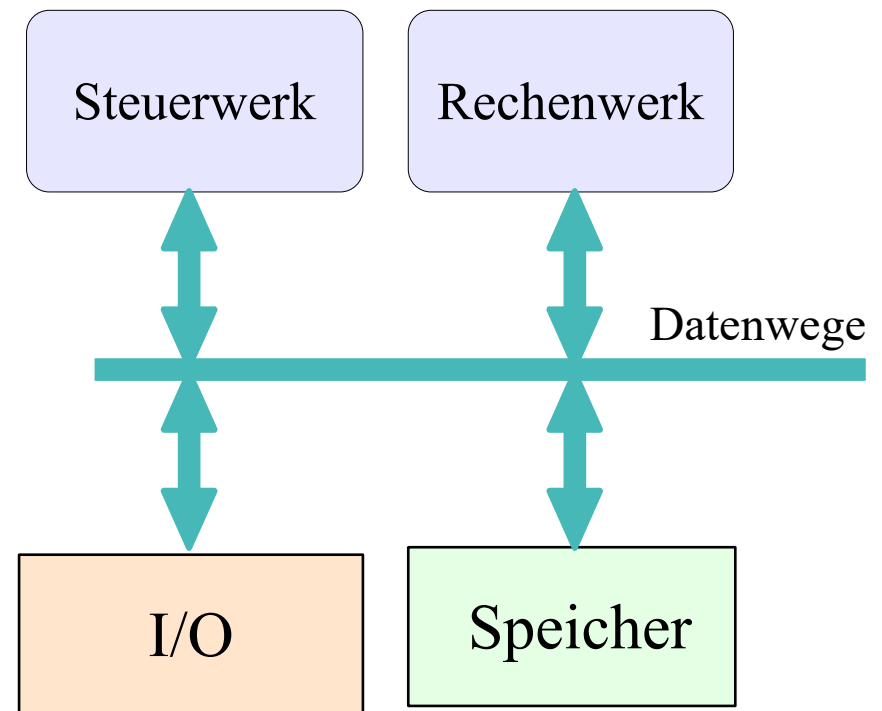


# Grundlegende Strukturen

- Ausgangspunkt:
  - Ein Rechner ist ein programmierbarer, endlicher Automat (State Machine)
  - Programmierbarkeit: Speicher enthält Daten und Anweisungen

- Grundkomponenten:

- Zentraleinheit (CPU)
  - Steuerwerk
  - Rechenwerk
- Speicher
- I/O
- Datenwege



# Daten - Verarbeitung

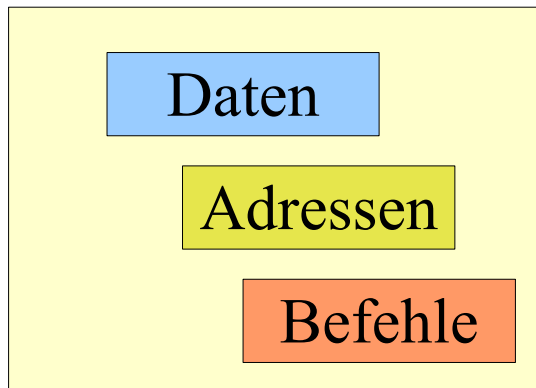
- Daten

- Zahlen, Zeichen, etc.
- Daten werden binär dargestellt und in Speichern oder Registern abgelegt
- Der Zugriff auf Daten erfordert eine Adressierung
- Auch Adressen können wie Daten behandelt werden

- Verarbeitung

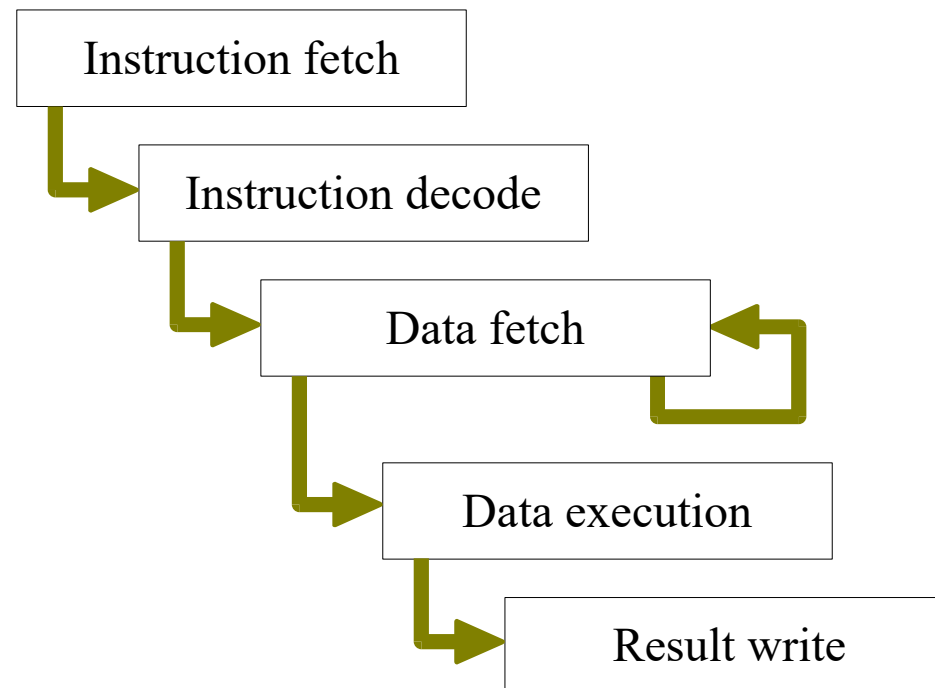
- Algorithmus = Vorschrift, wie Daten miteinander verknüpft werden
- Ein Programm ist eine Folge von Befehlen, z.B. mit arithmetischen und logischen Operationen
- Die Befehlsausführung kann von den Daten selbst abhängig sein (Verzweigung) oder Sprünge enthalten

Speicher enthält:



# Ablauf der Befehlsausführung

- Strategie:
  - Programm setzt sich aus Befehlen zusammen
  - Jeder Befehl wird in einer festen Abfolge abgearbeitet
  - Unterschiedliche Anzahl von Stufen möglich

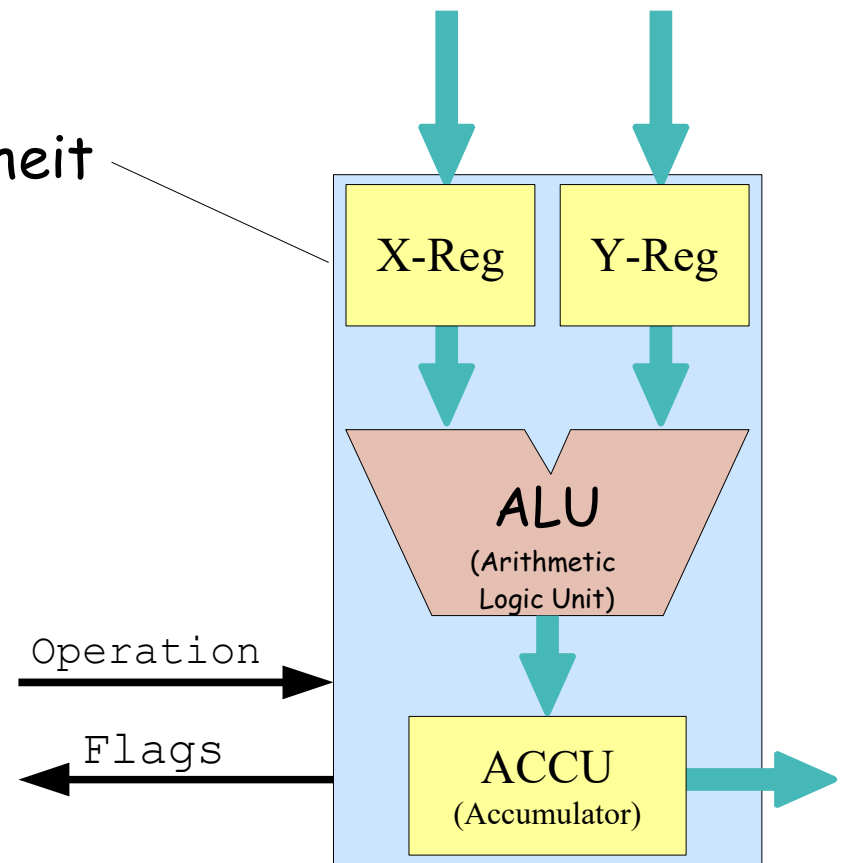


- Die Funktionsweise und Architektur von Rechnersystemen wird oft sehr abstrakt dargestellt, z.B. durch Blockschaltbilder
  - Häufig sind die Funktionseinheiten durch mehr oder weniger komplexe Schaltnetze bzw. Schaltwerke implementiert
  - Die Methodiken zur Beschreibung solcher Funktionseinheiten sind aus der Digitaltechnik bekannt
- Im Folgenden wird das prinzipielle Zusammenwirken verschiedener Funktionseinheiten anhand einer Modell-CPU demonstriert. Dies bedeutet aber nicht, dass alle CPU's so aufgebaut sind!

# Modell-CPU (1/7)

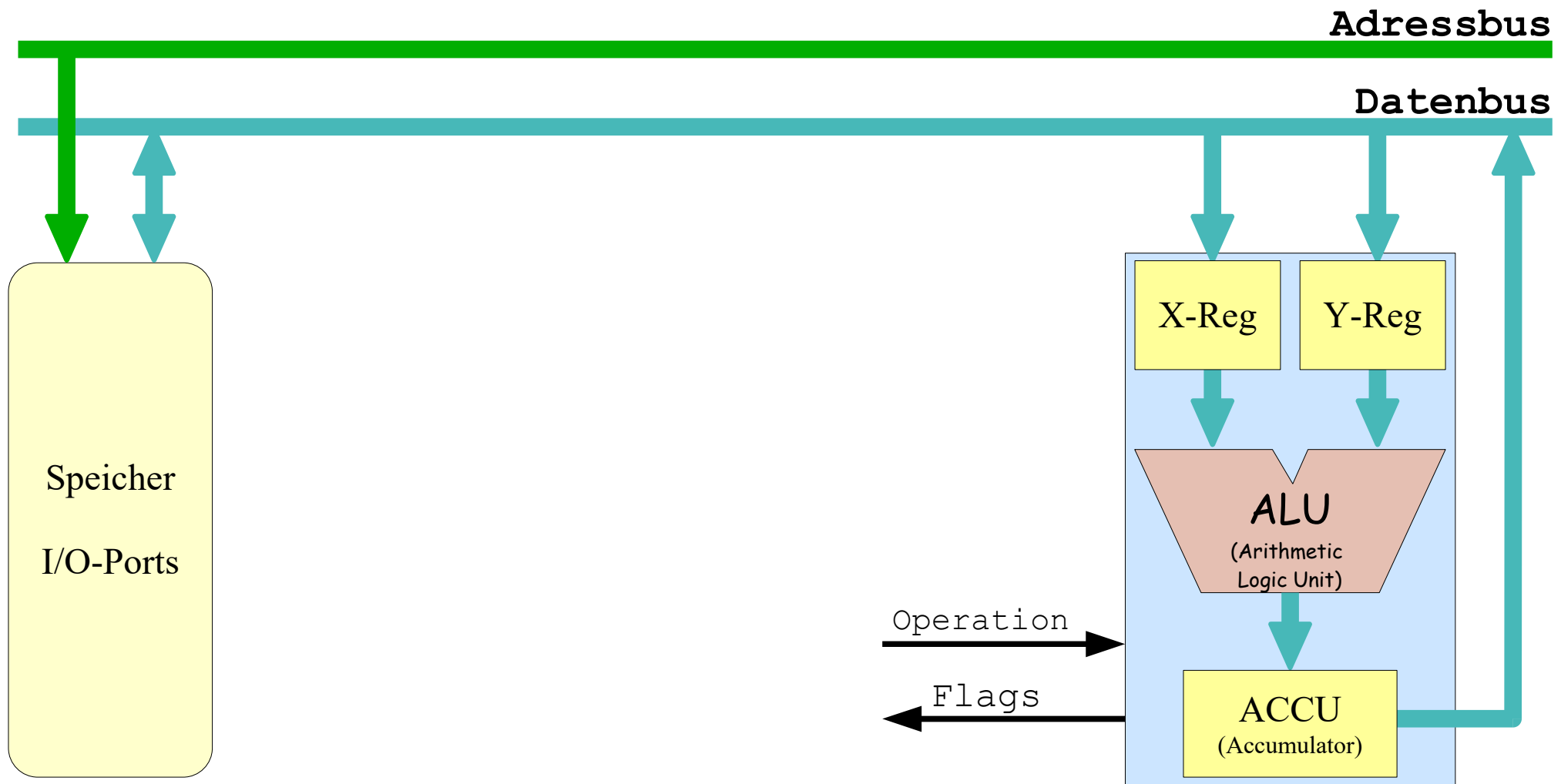
Ausgangspunkt: Die in den Register vorliegenden Daten werden arithmetisch oder logisch verknüpft. Die Verknüpfung (ALU) kann durch Steuersignale (Operation) eingestellt werden. Das Ergebnis wird im Akkumulator abgelegt (Ergebnis-Register). Ggf. werden Flags (Ergebnis = 0 oder negativ, Carry-Overflow) gesetzt. Realisierung der Verarbeitungseinheit als Schaltwerk.

Verarbeitungseinheit



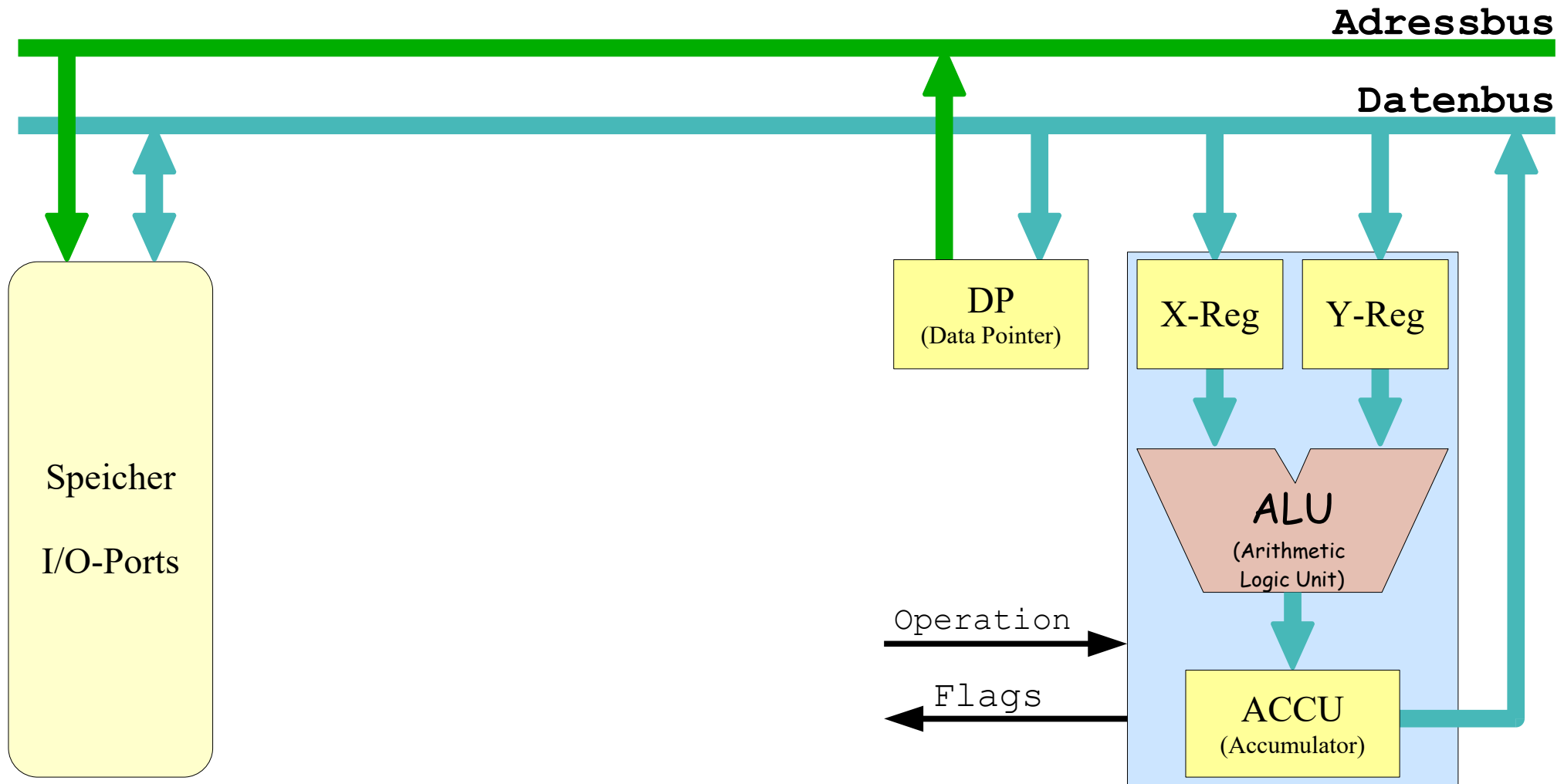
## Modell-CPU (2/7)

Die Operanden werden aus einem Speicher geholt und das Ergebnis wird dorthin zurückgeschrieben.  
Realisierung durch einen Daten- und Adressbus. Probleme: Die Speicheradresse muss gebildet und Schreib-/Lesezugriffe müssen koordiniert werden.



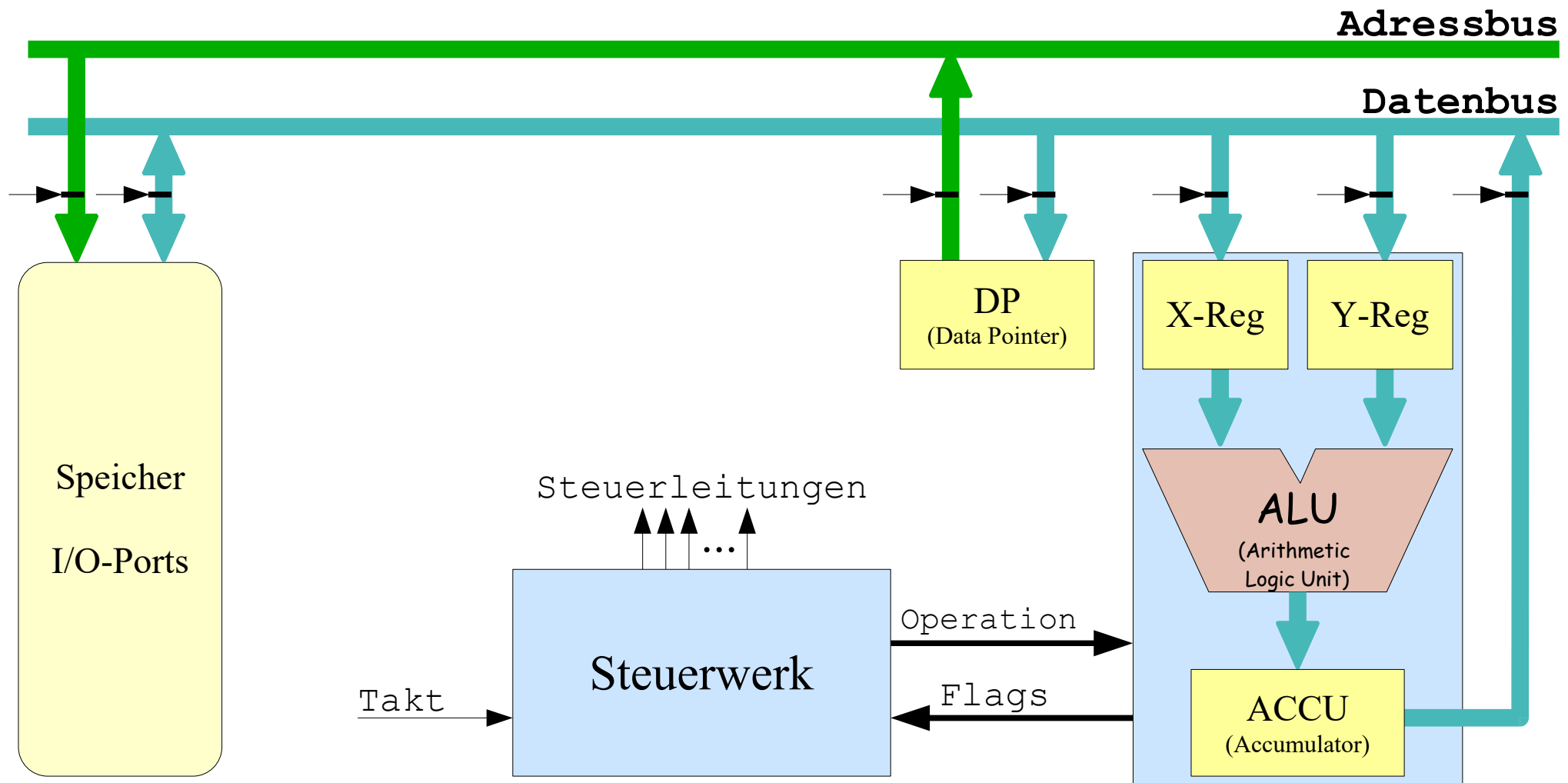
# Modell-CPU (3/7)

Die jeweilige Speicheradresse wird von einem Datenadress-Register (DP = Data Pointer) bereitgestellt, der seinen Inhalt wiederum vom Datenbus bezieht.



# Modell-CPU (4/7)

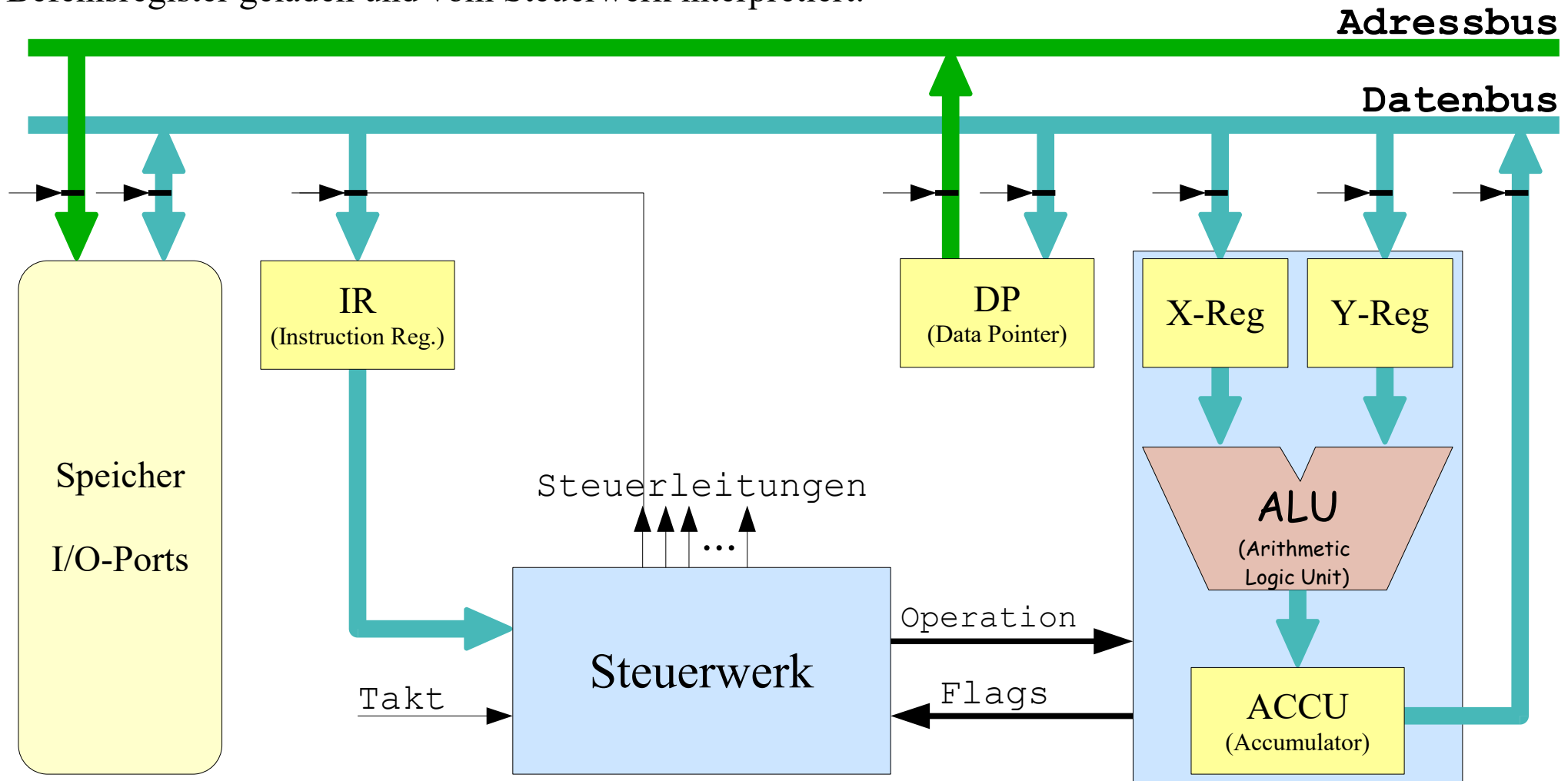
Die Schreib-/Lesezugriffe auf den Bus werden von einem Steuerwerk koordiniert. Dieses kann auch auf die Flags der Verarbeitungseinheit reagieren und dessen Operation steuern. Realisierung durch einen endlichen Automaten. Die Ablaufsteuerung ist im Mikrocode festgelegt.





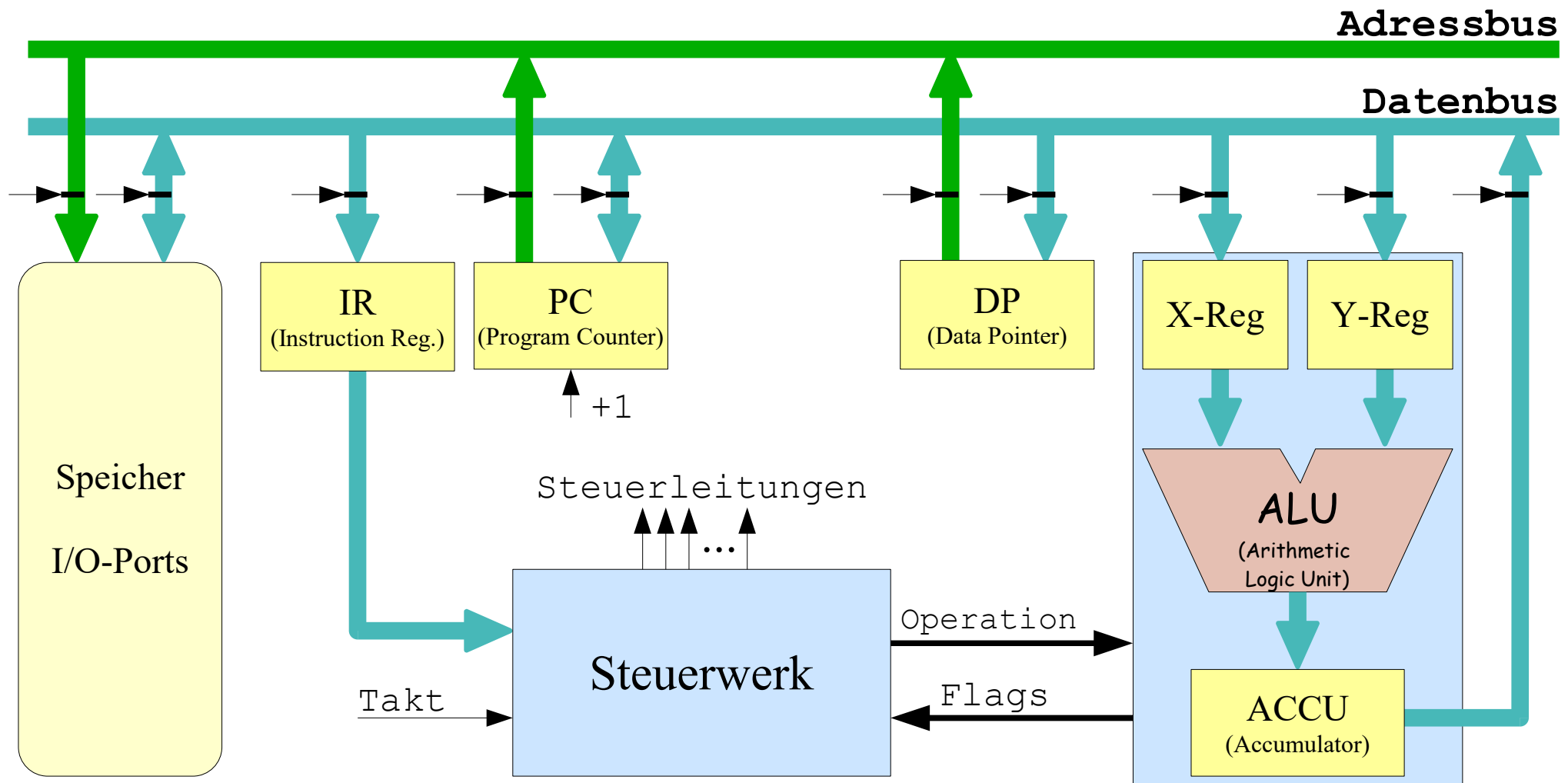
# Modell-CPU (5/7)

Das Steuerwerk kann verschiedene Maschinenbefehle ausführen, die Auswahl erfolgt durch den Eintrag im Befehlsregister (IR = Instruction Register). Die auszuführenden Maschinenbefehle werden sequentiell aus dem Speicher, der neben den Daten auch den Programmcode enthält, in das Befehlsregister geladen und vom Steuerwerk interpretiert.



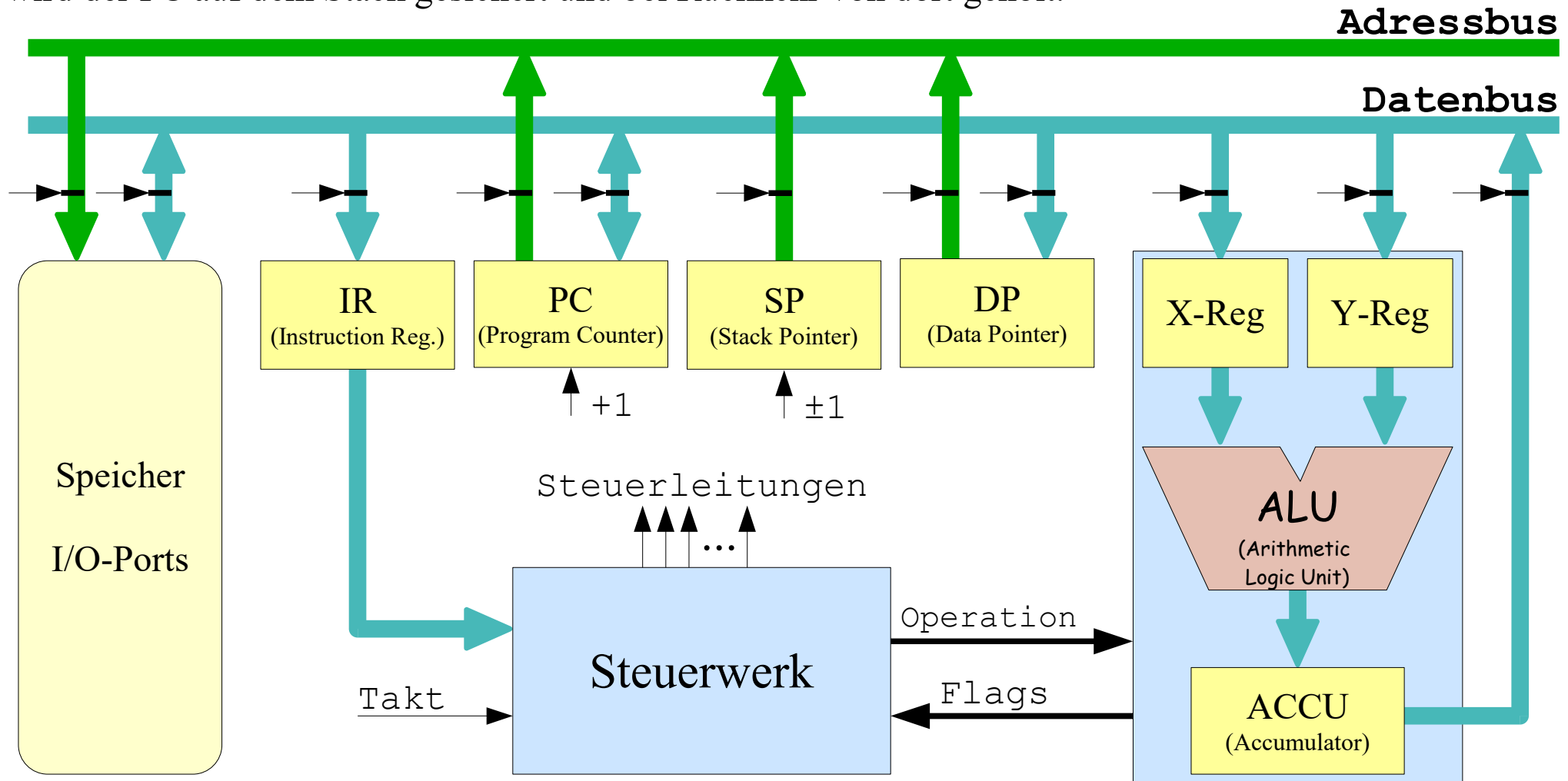
# Modell-CPU (6/7)

Die Adresse, von der der nächste Maschinenbefehl geholt wird, ist im Programmzähler (PC = Program Counter) gespeichert. Der PC kann inkrementiert (nächsten Befehl ausführen) oder mit einem Wert vom Datenbus (Programmsprung) geladen werden.



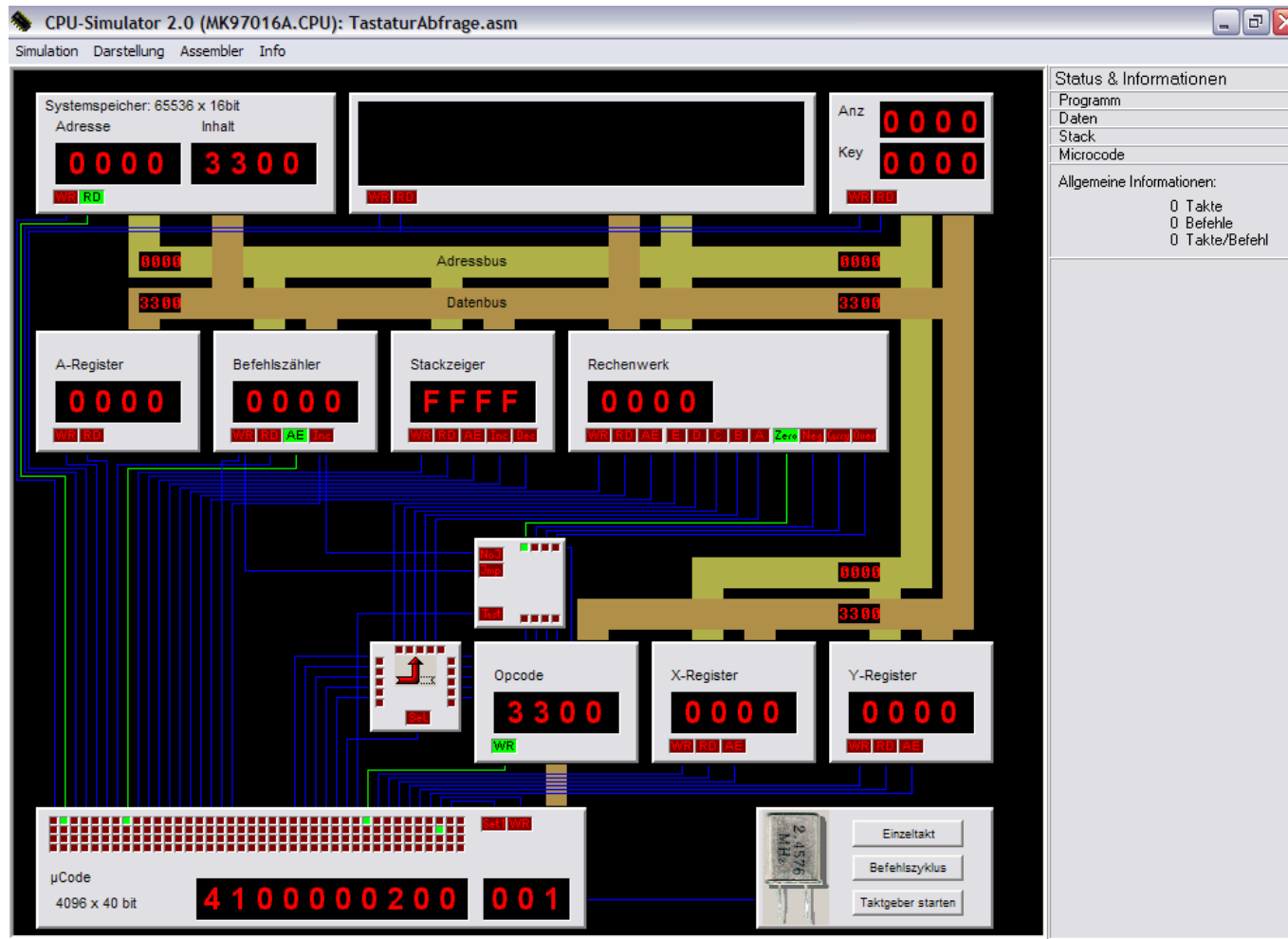
# Modell-CPU (7/7)

Unterprogramme können mit dem Stapelspeicher realisiert werden. Die jeweilige Speicheradresse befindet sich im Register SP (= Stack Pointer), der inkrementiert (nach Schreiben in den Stack) oder dekrementiert (nach Lesen aus dem Stack) werden kann. Vor der Verzweigung in ein Unterprogramm wird der PC auf dem Stack gesichert und bei Rückkehr von dort geholt.



# CPU-Simulator

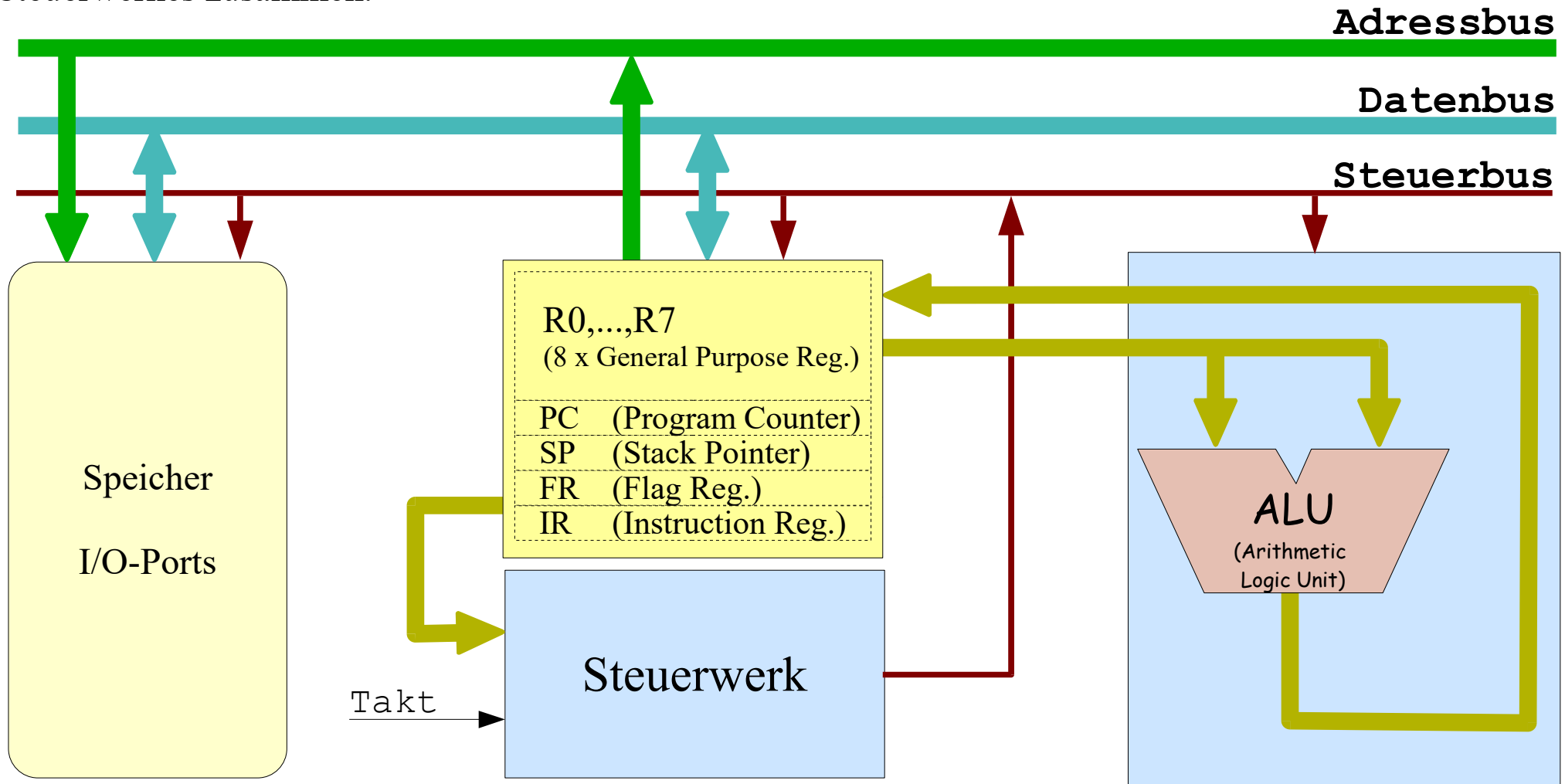
- Die internen Abläufe lassen sich gut an einem Simulator nachvollziehen
- Beispiel:



Download: [LEA, Technische Informatik](#)

# Modell-CPU mit Registerbank

Mit zusätzlichen General-Purpose-Registern lässt sich die Programmausführung beschleunigen. Diese Register können Daten oder Adressen enthalten und damit den Data Pointer und weitere Register ersetzen. Das Flag Register (FR) fasst Flags der ALU und weitere Zustandsinformationen des Steuerwerkes zusammen.



# Verarbeitungseinheit (Rechenwerk)

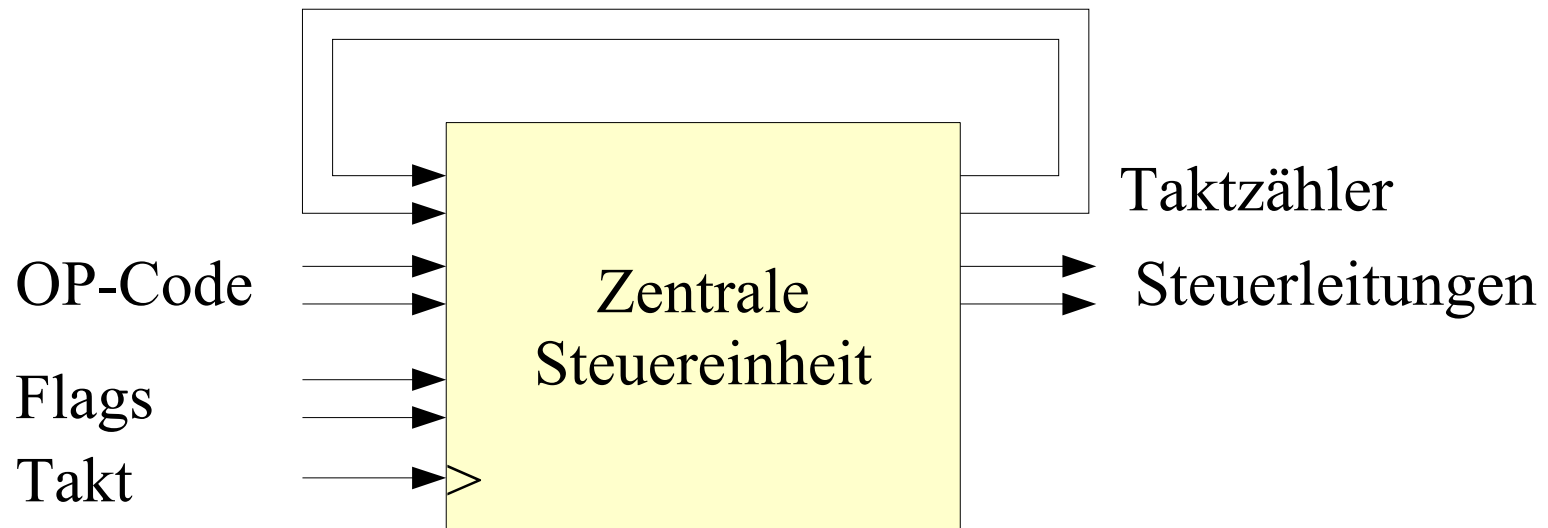
- Die **Verarbeitungseinheit** verknüpft die Daten und wertet sie aus
- Kenngrößen:
  - Art der Arithmetik (Integer, Festkomma, Fließkomma, ...)
  - Datenwortbreite
  - Komplexität der Operationen (ADD, MULT, DIV, ...)
  - Anzahl und Art der Einheiten (SISD, SIMD, MIMD), ..
  - Ausführungsgeschwindigkeit (Durchlaufzeit & Datendurchsatz)

# Steuereinheit (Steuerwerk)

- Die **Steuereinheit** koordiniert alle Abläufe innerhalb des Prozessors
- Ein Maschinenbefehl (Makroprogramm) wird in mehreren Schritten (Mikroprogramm) bearbeitet
- Kenngrößen:
  - Art und Anzahl der Befehle
  - Anzahl der gleichzeitig decodierten Befehle
  - Statische oder dynamische Instruktionszuweisung } → Befehlswortbreite
- Programmsprungausführung und Sprungvorhersage
- Anzahl und Art der koordinierten Rechenwerke
- Anzahl der koordinierten internen Kommunikationspfade
- Unterstützung der Kontextumschaltung (für Unterprogramme und Nebenläufigkeit von Programmen)

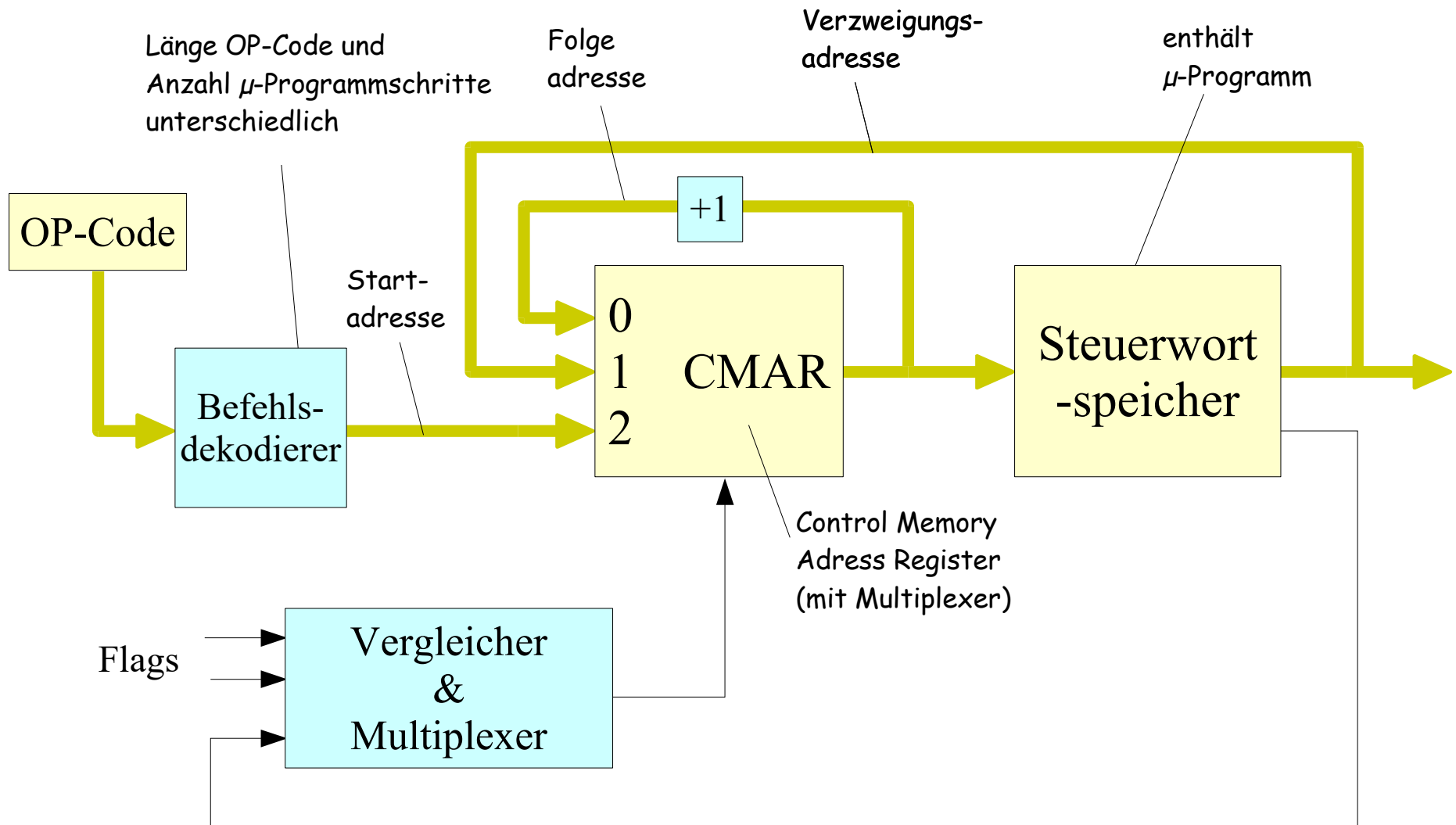
# Steuereinheit (Steuerwerk)

- Steuereinheit → endlicher Automat
- Implementierung der Mikroprogramme
- Realisierung durch:
  - Schaltwerk (aufwändig im Design)
  - EPROM (viele Dummy-Speicher, da nicht alle Flags in allen Operationen relevant)





# Steuereinheit (Mikroprogramm)



# I/O-Einheiten

- Die "Geräte" innerhalb eines Rechnersystems werden wie Speicherzellen über die Pfade des Speichersystems angesprochen
  - Register der Geräte werden als globale Adressen in den Speicherbereich eingeblendet
    - Verschiedene Zugriffsweisen möglich (Nur-Lese-, Schreib-Lese-, Nur-Schreib-Zugriff)
  - Geräte werden über interne Register und Signale gesteuert
    - Auslösung von Signalen im Gerät: z.B. durch Schreib-/Lesezugriff
    - Rückmeldung von Signalen an die CPU: z.B. über Interrupt

