

# Übung 09: Automaten und Multi-Cycle-Prozessor

Einführung in die Rechnerarchitektur

**Niklas Ladurner**

School of Computation, Information and Technology  
Technische Universität München

13. Dezember 2024



*TUM Uhrenturm*

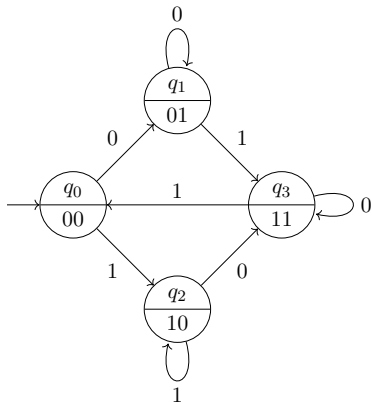
Keine Garantie für die Richtigkeit der Tutorfolien.  
Bei Unklarheiten/Unstimmigkeiten haben VL/ZÜ-Folien recht!

# Endliche Automaten

- Repräsentiert Funktion einer sequentiellen Schaltung (sequentiell: zustandsabhängig)
- als Diagramm: Zustände  $\rightarrow$  Kreise, Übergänge  $\rightarrow$  Kanten, Bedingungen  $\rightarrow$  Kantenbeschriftungen
- als 6-Tupel  $(I, O, S, s_0, \delta, \lambda)$ :
  - $I$ : Menge möglicher Eingaben
  - $O$ : Menge möglicher Ausgaben
  - $S$ : Zustandsmenge
  - $s_0$ : Startzustand
  - $\delta : S \times I \rightarrow S$ : Zustandsübergangsfunktion
  - $\lambda : S \rightarrow O$  (Moore),  $\lambda : S \times I \rightarrow O$  (Mealy): Ausgabefunktion

## Moore-Automat

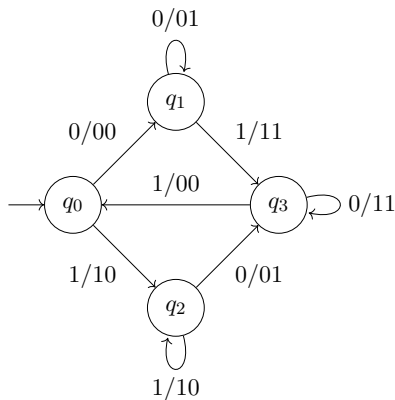
Ausgabe abhängig von aktuellem Zustand



$I = \{0, 1\}$ ,  $O = \{00, 01, 10, 11\}$ ,  $S = \{q_0, q_1, q_2, q_3\}$ ,  $\delta, \lambda$  (abh. vom Typen)

## Mealy-Automat

Ausgabe abhängig von aktuellem Zustand + Eingabe



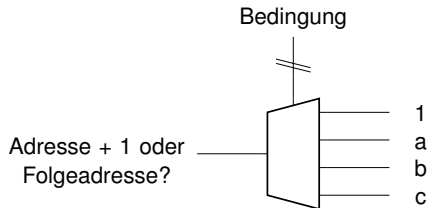
# Endliche Automaten: Realisierung

- One-Hot-Kodierung: Genau 1 FF ist auf 1 (aktueller Zustand), einfach aber verschwenderisch
- Binärkodierung: FFs zusammen bilden Binärzahl des aktuellen Zustands, spart FFs aber komplexer
- Mikroprogrammierte Steuerwerke: Nur ein Speicherbaustein, enthält vollständigen Automaten. Eingaben werden als Adressen interpretiert, sehr flexibel.

Zustand	One-Hot	Binär
$S_0$	0001	00
$S_1$	0010	01
$S_2$	0100	10
$S_3$	1000	11

# Adressmodifizierendes mikroprogrammiertes Steuerwerk

	ROM		
	Folgeadresse	Bedingung	Steuersignale
00:	10	01	0
01:	01	10	0
10:	00	11	0
11:	00	00	0



if a then goto 10 else goto 01

while b

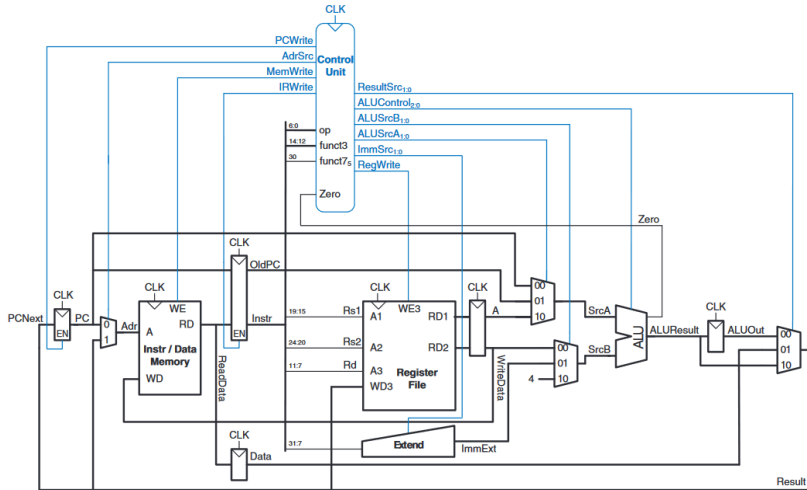
if c then goto 00 else goto 11

goto 00

- Aufteilung einer Instruktion in mehrere Schritte
- kürzere kritische Pfade in den einzelnen Teilschritten → höhere Taktfrequenz möglich
- allerdings benötigt eine Instruktion jetzt auch mehrere Taktzyklen!
- komplexeres Steuerwerk, da Zustandsautomat umgesetzt werden muss

in der Praxis haben sich Multi-Cycle-Prozessoren nicht durchgesetzt!

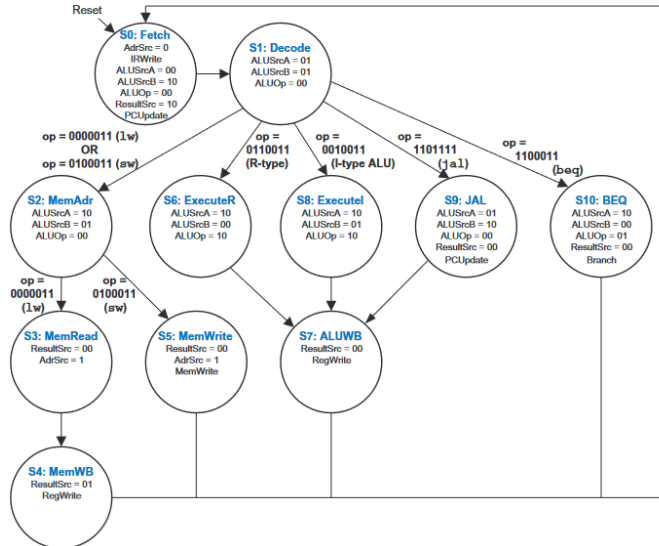
# RISC-V Multi-Cycle-Prozessor: Schaltbild



(Quelle: Vorlesungsmaterialien ERA)



# RISC-V Multi-Cycle-Prozessor: Zustandsautomat



- „H09 — Sequenzielles Steuerwerk“ bis 05.01.2025 23:59 Uhr
- Implementierung des Steuerwerks des Multi-Cycle-Prozessors
- `StateUpdate.dig` kann durchaus umfangreich werden

- Zulip: „ERA Tutorium - Do-1600-1“ bzw. „ERA Tutorium - Fr-1500-2“
- ERA-Moodle-Kurs
- ERA-Artemis-Kurs
- Prozessor-Assets (kein offizielles Material!)

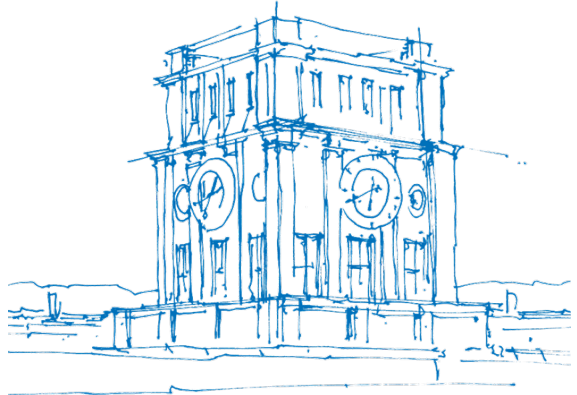
# Übung 09: Automaten und Multi-Cycle-Prozessor

Einführung in die Rechnerarchitektur

**Niklas Ladurner**

School of Computation, Information and Technology  
Technische Universität München

13. Dezember 2024



*TUM Uhrenturm*