

Lernzettel

Speicher- und Flusslogik: Flip-Flops, Latches,
Register und Registerdateien

Universität:	Technische Universität Berlin
Kurs/Modul:	Rechnerorganisation
Erstellungsdatum:	September 20, 2025



Zielorientierte Lerninhalte, kostenlos!
Entdecke zugeschnittene Materialien für deine Kurse:

<https://study.AllWeCanLearn.com>

Rechnerorganisation

Lernzettel: Speicher- und Flusslogik: Flip-Flops, Latches, Register und Registerdateien

(1) Grundbegriffe. In digitalen Systemen speichern und verschieben sich Daten über Speicherzellen. Zwei zentrale Speicherelemente sind Latches (level-gefüttert) und Flip-Flops (taktsignalabhängig). In Rechnerarchitekturen werden diese Elemente zu Registern und Registerdateien verknüpft, um Daten zuverlässig zu speichern und zu übertragen.

(2) Latches. Latches speichern einen Bitwert basierend auf einem oder mehreren Eingängen. Ein häufiges Beispiel ist der SR-Latch mit NOR-Gates.

Ein typischer SR-Latch (NOR-basiert) mit S (Set) und R (Reset) hat die folgenden Zustände für die nächste Stabilität Q^+ (ausgedrückt als logische Folge):

$$S = 0, R = 0 \Rightarrow Q^+ = Q$$

$$S = 1, R = 0 \Rightarrow Q^+ = 1$$

$$S = 0, R = 1 \Rightarrow Q^+ = 0$$

$$S = 1, R = 1 \Rightarrow Q^+ \text{ undefiniert (ungültig)}$$

(3) Flip-Flops. Flip-Flops sind synchronisierte Speicherelemente, deren Zustand durch eine Taktsignalfanke aktualisiert wird. Wichtige Typen:

- D-Flipflop (edge-triggered):

$$Q^+ = D \quad \text{bei einer aktiven Clock-Flanke; sonst } Q^+ = Q$$

- JK-Flipflop:

$$Q^+ = J\bar{Q} + \bar{K}Q$$

Typischerweise gilt: - J=0, K=0: kein Wechsel - J=0, K=1: Reset - J=1, K=0: Set - J=1, K=1: Toggle

- T-Flipflop (Toggle-Flipflop):

$$Q^+ = Q \oplus T$$

(Q ändert sich nur, wenn T=1)

(4) Register und Registerdateien. - Register: Eine Gruppe von D-Flip-Flops, die zusammen ein W-bit Wort speichern. Breite W bedeutet W Bits pro Register.

- Registerdatei: Mehrere Register (R Register), typischerweise mit: - zwei Lesekanälen (read ports) und einem Schreibkanal (write port) - Adressierung über Adressbusse - Adressdecode-Einheit, die das eine Register auswählt, das beschrieben wird - Leseausgänge, die durch Multiplexer zu zwei unabhängigen Read Outputs zusammengeführt werden

Funktionsweise: - Schreibvorgang: Der Adressdecoder wählt das Zielregister aus und das Write Enable-Signal ermöglicht das Schreiben der Eingangsdaten in dieses Register. - Lesevorgang: Die aktuellen Inhalte der gewählten Register werden über zwei separate Read-Ports ausgegeben (via Multiplexer) an die nachfolgenden Recheneinheiten.

(5) Timing- und Synchronisationsaspekte. - Setup- und Hold-Time: Daten müssen vor und nach der Clock-Flanke stabil vorliegen, damit der Flip-Flop korrekt schreibt. - Synchronisation: Alle Register arbeiten idealerweise synchron zur Uhr; asynchrone Reset- oder Clear-Signale sind separat zu beachten. - Propagation Delay: Die Zeit, die von der Clock-Flanke bis zur Änderung

von Q vergeht, beeinflusst Pipeline- und Speicherverhalten. - Fan-out: Anzahl der Folgeschaltungen, die von einem Ausgangssignal getrieben werden kann, ohne Signale zu schwächen.

(6) Anwendung in Rechnerarchitektur. - Programme und Daten werden über Registerdateien verschoben, gelesen und geschrieben. - Wichtige Beispiele: - Program Counter (PC): ein spezielles Register, das die aktuelle Adress- oder Befehlsposition hält. - Instruction Register (IR): hält die aktuell geladene Instruktion während der Decodierung. - Allgemeine Register in der Datenpfad-Einheit: temporäres Speichern von Operanden, Zwischenergebnissen. - In der Speicherhierarchie dienen Registerdateien als sehr schneller Puffer zwischen Rechenwerk und Arbeitsspeicher; sie unterstützen Adressierung, Synchronisation und direkte Speicherzugriffe (Direct Memory Access, DMA) in größeren Systemen.

(7) Beispielstruktur einer Registerdatei. - Annahme: Registerdatei mit R Registern, Breite W Bits. - Schreibpfad: Write-Enable (WE) aktiv; Decoder wählt das Zielregister aus; Eingangsdaten D_W werden in das ausgewählte Register geschrieben. - Lesepfad 1/2: Zwei unabhängige Read-Ports liefern Wortausgänge $RD1$ und $RD2$ durch Multiplexer, basierend auf Read-Adressen. - Aufbauverständnis: Jedes Register besteht aus W D-Flip-Flops; der Decoder entscheidet, welches Register beschrieben wird; Multiplexer entscheiden, welches Register gelesen wird.

(8) Typische Aufgabenstellungen. - Skizziere eine 8-Bit-Registerdatei mit 16 Registern, zwei Lesepfaden und einem Schreibpfad. Erläutere, wie Daten von zwei Registern gleichzeitig gelesen werden können. - Beschreibe den Aufbau eines D-Flipflops, eines 8-Bit-Registers und einer 8×16 -Registerdatei mit zwei Read Ports. Skizziere die wichtigsten Signale (CLK, WE, Adressen, Data In/Out). - Erkläre den Unterschied zwischen Latches und Flip-Flops im Kontext synchroner Rechenwerke.