

Lernzettel

ALU-Grundlagen: Funktionscodes, Operationen, Carry-Propagation und Statusbits

Universität: Technische Universität Berlin
Kurs/Modul: Technische Grundlagen der Informatik (TechGI) - Digitale Systeme
Erstellungsdatum: September 6, 2025



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

<https://study.AllWeCanLearn.com>

Technische Grundlagen der Informatik (TechGI) - Digitale
Systeme

Lernzettel: ALU-Grundlagen: Funktionscodes, Operationen, Carry-Propagation und Statusbits

(1) Überblick. Die Arithmetic Logic Unit (ALU) führt arithmetische und logische Operationen auf Eingabewerten A und B durch. Sie erhält einen Funktionscode, einen optionalen Carry-In und liefert ein Ergebnis sowie ggf. einen Carry-Out. Zentrale Konzepte:

- Operationen durch Funktionscodes auswählen
- Carry-Propagation beim Addieren
- Statusbits (Flags) zur Fehler- und Zustandsinformation

(2) Funktionscodes und Operationen. Die Operationen werden durch einen Funktionscode gesteuert. Typische 4-Bit-Codes (Beispiele):

- 0000 Addieren
- 0001 Subtrahieren
- 0010 UND (bitweises AND)
- 0011 ODER (bitweises OR)
- 0100 XOR
- 0101 NOT A
- 0110 SHL (Shift Left)
- 0111 SHR (Shift Right)
- 1000 INC
- 1001 DEC
- 1010 CMP/Ausgabe-Vergleich (A gegen B)
- 1111 NOP

(3) Carry-Propagation. Beim Addieren entsteht ein Carry-Verhalten, das von Bit zu Bit weitergetragen werden kann. Für bitweise Addition gilt im Normalfall (Full-Adder) mit Carry-In C_{in} :

$$S_i = A_i \oplus B_i \oplus C_i, \quad C_{i+1} = (A_i \& B_i) \mid ((A_i \oplus B_i) \& C_i).$$

Carry-Lookahead (Kurzfassung).

$$G_i = A_i \& B_i, \quad P_i = A_i \oplus B_i, \quad C_{i+1} = G_i \mid (P_i \& C_i).$$

(4) Statusbits (Flags). Wichtige Statusbits, die von der ALU gesetzt werden:

- Z (Zero): Ergebnis ist Null (alle Bits 0).

- C (Carry): Carry-Out des höchsten Bits (bei Addition).
- N (Negative): Vorzeichen des Ergebnisses (MSB des Ergebnisses).
- V (Overflow): Sign-Overflow bei zweistelliger Zweierkomplement-Interpretation.

(5) Beispiel: 4-Bit Addition mit Carry-Propagation. Gegeben seien $A = 0100(4)$, $B = 0100(4)$, $C_{in} = 0$, *Funktionscode*0000(*Addieren*).

Berechnung schrittweise (LSB zuerst):

i	A_i	B_i	C_i	S_i	C_{i+1}
0	0	0	0	0	0
1	0	0	0	0	0
2	1	1	0	0	1
3	0	0	1	1	0

Summe $S = S_3S_2S_1S_0 = 1000_2 = 8$. Carry-Out $C_4 = 0$.

Flags am Ende:

$$Z = 0, \quad N = 1, \quad C = 0, \quad V = 1.$$

Hinweis: Overflow tritt bei signierter Zweierkomplement-Darstellung auf, z. B. positives + positives ergibt negatives Resultat.

(6) Kurze Übungen (Canvas-Formate). - Notiere zwei Operationen, die kein Carry-Out erzeugen, aber sinnvoll sind (z. B. AND, XOR). - Erkläre, wann das Overflow-Flag bei Subtraktion ($A - B$) relevant ist. - Skizziere grob die Carry-Lookahead-Logik in zwei Sichten: G/P und das endgültige C_n .

(7) Fazit. Die ALU bildet das Rechen- und Logikzentrum eines Prozessors. Funktionscodes steuern Operationen, Carry-Propagation bestimmt das Weiterreichen von Zählern, und die Statusbits liefern wichtige Zustandsinfos für weitere Verarbeitungsschritte.