SPS | LERNEN, ÜBEN, NACHSCHLAGEN

Bits, Hex & Logik

Zahlen lesen, Bitmasken verstehen und logische Entscheidungen prüfen.

Das kannst du danach

- Du rechnest kleine Binär- und Hexzahlen nachvollziehbar um.
- Du unterscheidest Bitmuster, Zahlenwert und vorzeichenbehaftete Deutung.
- Du erklärst AND, OR und XOR an einer Prüfaufgabe.

Dein Weg durch das Heft

01 Ein Bit hat zwei Zustände · S. 2

02 Hex ist eine kurze Schreibweise · S. 3

03 AND, OR und XOR vergleichen · S. 4

04 Bitmasken prüfen einzelne Meldungen · S. 5

05 Gleiche Bits, anderer Zahlenwert · S. 6

06 Schieben, Drehen und Prüfen · S. 7

SO LERNST DU DAMIT

Lies ein Kapitel und rechne das Beispiel nach. Löse danach die Aufgaben zuerst selbst. Auf den Hilfeseiten findest du Denkanstöße; die Lösungen erklären den Weg und typische Stolperstellen.

Die Beispiele dienen dem Lernen. Konkrete Arbeit, Prüfung und Freigabe richten sich nach Aufgabe, Zeichnung, Herstellerangaben und betrieblichen Vorgaben. Dieses Heft ersetzt keine Unterweisung.

KAPITEL 01 | VERSTEHEN

Ein Bit hat zwei Zustände

Ein Bit kann den Wert 0 oder 1 annehmen. Acht Bits bilden ein Byte. Eine Gruppe von Bits ist zunächst nur ein Muster; erst eine vereinbarte Bedeutung macht daraus eine Zahl, ein Zeichen oder mehrere Schaltsignale.

Bei einer Binärzahl verdoppelt sich der Stellenwert von rechts nach links: 1, 2, 4, 8, 16 und so weiter. Addiert werden nur die Stellenwerte der gesetzten Bits. Führende Nullen verändern den positiven Zahlenwert nicht, halten aber die vereinbarte Wortbreite sichtbar.

Stellenwerte addieren, wo eine 1 steht: 32 + 8 + 4 + 1 = 45

BEISPIEL / LERNZIEL

$0010\ 1101_2 = 32 + 8 + 4 + 1 = 45_{10}$. Die tiefste Stelle ist Bit 0. In diesem Byte ist also unter anderem Bit 5 gesetzt.

Praxistipp

Schreibe die Stellenwerte über das Bitmuster. So erkennst du auch einen verrutschten Bitindex.

Typischer Fehler

Bit 4 besitzt den Stellenwert 16, nicht 4. Die Nummerierung beginnt hier bei 0.

Meine Notizen

KAPITEL 02 | VERSTEHEN

Hex ist eine kurze Schreibweise

Im Hexadezimalsystem gibt es 16 Ziffern: 0 bis 9 und A bis F. A bedeutet 10, F bedeutet 15. Eine Hexziffer beschreibt genau vier Bits. Zwei Hexziffern können daher ein Byte darstellen.

Zum Umrechnen von Binär nach Hex gruppierst du von rechts in Vierergruppen. Jede Gruppe wird einzeln übersetzt. Für die Umrechnung in Dezimal gelten die Stellenwerte 1, 16, 256 und so weiter. Oktal arbeitet entsprechend mit Basis 8; eine Oktalziffer entspricht drei Bits.

BEISPIEL / LERNZIEL

$0010\ 1101_2$ wird zu $2D_{16}$. In Dezimal: $2 \times 16 + 13 = 45$. Das Präfix 0x in Programmen kennzeichnet oft eine Hexzahl, etwa 0x2D.

Praxistipp

Schreibe bei Übungsrechnungen die Basis dazu. 10 kann binär zwei, dezimal zehn oder hexadezimal sechzehn bedeuten.

Typischer Fehler

A bis F sind Ziffernwerte. Sie stehen hier nicht für Variablen.

Meine Notizen

KAPITEL 03 | VERSTEHEN

AND, OR und XOR vergleichen

Eine logische Verknüpfung verbindet Wahrheitswerte. AND ist nur dann wahr, wenn beide Eingänge wahr sind. OR ist wahr, wenn mindestens ein Eingang wahr ist. XOR ist wahr, wenn genau einer von zwei Eingängen wahr ist.

Bitweise Operationen prüfen jedes Stellenpaar getrennt. Das ist etwas anderes als die Frage, ob eine ganze Zahl gleich null ist. Eine Wahrheitstabelle zeigt alle Eingangskombinationen und verhindert, dass ein Sonderfall übersehen wird.

A	B	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

BEISPIEL / LERNZIEL

Für A = 1010 und B = 1100 ergeben sich A AND B = 1000, A OR B = 1110 und A XOR B = 0110. Bei A = B = 1 liefert OR eine 1, XOR dagegen 0.

Praxistipp

Teste immer auch den Fall, dass beide Eingänge gleichzeitig 1 sind. Dort unterscheiden sich OR und XOR.

Typischer Fehler

Eine normale Programmlogik ist keine nachgewiesene Sicherheitssteuerung. Die Übung erklärt nur die logische Funktion.

Meine Notizen

KAPITEL 04 | VERSTEHEN

Bitmasken prüfen einzelne Meldungen

Eine Bitmaske wählt bestimmte Stellen aus. Zum Prüfen eines einzelnen Bits setzt du genau diese Stelle in der Maske auf 1 und verknüpfst das Datenwort mit AND. Ein Ergebnis ungleich null bedeutet dann: Dieses Bit war gesetzt.

Mit OR lassen sich ausgewählte Bits setzen. Mit AND und einer passenden invertierten Maske lassen sie sich löschen. XOR kippt ausgewählte Bits um. Beim Invertieren muss die verwendete Wortbreite feststehen, sonst ist nicht klar, wie viele Stellen dazugehören.

BEISPIEL / LERNZIEL

Status = 0010 1101, Maske für Bit 3 = 0000 1000. AND ergibt 0000 1000: Bit 3 ist gesetzt. Für Bit 4 wäre die Maske 0001 0000; das Ergebnis wäre 0.

Praxistipp

Notiere zur Maske auch die Bedeutung des Bits, beispielsweise „Material erkannt“. Hexwerte allein erklären die Anlage nicht.

Typischer Fehler

Prüfen verändert den Status nicht. Eine Löschoperation ist dagegen ein Eingriff in das Datenwort.

Meine Notizen

KAPITEL 05 | VERSTEHEN

Gleiche Bits, anderer Zahlenwert

Ohne Vorzeichen beschreibt ein Byte Werte von 0 bis 255. Im 8-Bit-Zweierkomplement beschreibt es Werte von -128 bis +127. Das oberste Bit hat dabei den Stellenwert -128; die restlichen Stellen behalten ihre positiven Werte.

Ein Überlauf entsteht, wenn das mathematische Ergebnis nicht in den gewählten Wertebereich passt. Manche Systeme behalten nur die unteren Bits, andere melden einen Fehler oder verhalten sich anders. Für jede Aufgabe muss die Regel ausdrücklich feststehen.

BEISPIEL / LERNZIEL

1111 1110 ist ohne Vorzeichen 254. Im 8-Bit-Zweierkomplement ergibt sich -128 + 64 + 32 + 16 + 8 + 4 + 2 = -2. Die Bits sind identisch; die Deutung ist verschieden.

Praxistipp

Schreibe zuerst „8 Bit, mit/ohne Vorzeichen“ auf. Rechne erst danach.

Typischer Fehler

Ein gesetztes höchstes Bit bedeutet nicht bei jeder Datenart eine negative Zahl.

Meine Notizen

KAPITEL 06 | VERSTEHEN

Schieben, Drehen und Prüfen

Beim logischen Schieben werden Bits verschoben und freie Stellen mit Nullen gefüllt. Herausgeschobene Bits gehen bei fester Wortbreite verloren. Bei einer Rotation tauchen sie auf der anderen Seite wieder auf.

Ein Linksschieben entspricht nur dann einer Multiplikation mit zwei, wenn die Deutung dazu passt und kein relevantes Bit verloren geht. Arithmetisches Rechtsschieben mit Vorzeichenerweiterung ist außerdem von logischem Rechtsschieben zu unterscheiden.

BEISPIEL / LERNZIEL

8 Bit, ohne Vorzeichen: 1000 0001 logisch um eine Stelle links geschoben ergibt 0000 0010. Links rotiert ergibt es 0000 0011. Das linke Bit geht beim Schieben verloren, bei der Rotation kommt es rechts wieder an.

Praxistipp

Prüfweg: Wortbreite festlegen, Operation benennen, verlorene Bits markieren, Ergebnis noch einmal dezimal deuten.

Typischer Fehler

Schieben und Rotieren sind trotz ähnlicher Bewegung nicht dieselbe Operation.

Meine Notizen

AUFGABEN 1-4

Jetzt selbst anwenden

Notiere deinen Rechenweg oder begründe deine Entscheidung. Die Kapitelangabe hilft beim gezielten Nachlesen.

Aufgabe 1 | Kapitel 1

Rechne $0010\ 1101_2$ in Dezimal um.

Meine Notizen

Aufgabe 2 | Kapitel 2

Welche Hexzahl entspricht $0010\ 1101_2$? Begründe beide Ziffern.

Meine Notizen

Aufgabe 3 | Kapitel 3

Berechne 1010 AND 1100 sowie 1010 XOR 1100.

Meine Notizen

Aufgabe 4 | Kapitel 4

Welche 8-Bit-Maske prüft Bit 4? Ist es in 0010 1101 gesetzt?

Meine Notizen

AUFGABEN 5-6

Jetzt selbst anwenden

Notiere deinen Rechenweg oder begründe deine Entscheidung. Die Kapitelangabe hilft beim gezielten Nachlesen.

Aufgabe 5 | Kapitel 5

Deute 1111 1110 einmal ohne Vorzeichen und einmal als 8-Bit-Zweierkomplement.

Meine Notizen

Aufgabe 6 | Kapitel 6

Verschiebe und rotiere 1000 0001 um eine Stelle nach links. Es gelten 8 Bit.

Meine Notizen

DENKANSTÖSSE 1-6

Hilfen, bevor du nachsiehst

Zu Aufgabe 1

Addiere nur die Stellenwerte mit einer 1.

Zu Aufgabe 2

Teile in zwei Vierergruppen.

Zu Aufgabe 3

Prüfe jede Stelle einzeln. XOR verlangt unterschiedliche Eingangsbits.

Zu Aufgabe 4

Bit 0 liegt ganz rechts. Die Maske braucht nur eine 1.

Zu Aufgabe 5

Im zweiten Fall hat die höchste Stelle den Wert -128.

Zu Aufgabe 6

Beim Rotieren kehrt das links herauslaufende Bit rechts zurück.

LERNTRICK

Nach einem Hinweis schließt du das Heft kurz. Erkläre dir den Ansatz in eigenen Worten und versuche die Aufgabe erneut. Erst dann mit der Lösung vergleichen.

SELBSTKONTROLLE 1-3

Lösungen mit Begründung

Aufgabe 1

32 + 8 + 4 + 1 = 45. Führende Nullen tragen nichts zur Summe bei.

Aufgabe 2

0010 entspricht 2, 1101 entspricht 13 und damit D. Ergebnis: $2D_{16}$.

Aufgabe 3

AND = 1000. XOR = 0110. Bei zwei Einsen liefert AND eine 1 und XOR eine 0.

MEINE WIEDERHOLUNG

Welche Aufgabe war noch unsicher? Notiere den zugehörigen Begriff oder Rechenschritt. Wiederhole genau diesen Teil im Kapitel und löse die Aufgabe morgen ohne Vorlage erneut.

SELBSTKONTROLLE 4-6

Lösungen mit Begründung

Aufgabe 4

Maske 0001 0000. AND mit 0010 1101 ergibt 0000 0000: Bit 4 ist nicht gesetzt.

Aufgabe 5

Ohne Vorzeichen 254; mit Zweierkomplement -2. Nicht die Bits, sondern die vereinbarte Interpretation ändert sich.

Aufgabe 6

Logisch geschoben: 0000 0010. Rotiert: 0000 0011. Ein verlorenes Bit kann beim Schieben den Zahlenwert stark ändern.

MEINE WIEDERHOLUNG

Welche Aufgabe war noch unsicher? Notiere den zugehörigen Begriff oder Rechenschritt. Wiederhole genau diesen Teil im Kapitel und löse die Aufgabe morgen ohne Vorlage erneut.

NACHSCHLAGEN | DEIN NÄCHSTER SCHRITT

Begriffe & Weiterlernen

Bit

Eine Stelle mit dem Wert 0 oder 1.

Maske

Bitmuster zur Auswahl bestimmter Stellen.

Wortbreite

Anzahl der gemeinsam betrachteten Bits.

Überlauf

Das Ergebnis passt nicht in den festgelegten Wertebereich.

Auf LehrLynx weiterüben

- Zahlensysteme ausprobieren
- Bitoperationen rechnen
- Logik-Labor

Die zugehörigen Online-Seiten findest du über den Downloadbereich oder die Website-Suche. Dort lassen sich Fachbegriffe anklicken und Modelle selbst verändern.

KURZ ERKLÄREN STATT NUR WIEDERERKENNEN

Erkläre einen Begriff mit eigenem Beispiel. Sage anschließend, wo das Beispiel nicht mehr ausreicht. So prüfst du, ob du den Zusammenhang verstanden hast.

Meine Notizen