

Aufgabe: Bit 6 auf Port 0x0300 überprüfen, ob es gesetzt ist.

Lösungsschritte:

1. Einlesen des Bytes von Port 0x0300 mit `inportb(0x0300)`;
2. Maske für 6. Bit erstellen: `0100 0000` \Rightarrow `0x40`
3. Eingelesenen Byte und Maske bitweise UND-Verknüpfen
4. Wenn Ergebnis > 0 dann ist das Bit gesetzt, sonst nicht.

Aufgabe: Bit 3 und 7 auf Port 0x0300 überprüfen, ob sie gesetzt sind.

Lösungsschritte:

1. Einlesen des Bytes von Port 0x0300 mit `inportb(0x0300)`;
2. Maske für 3. und 7. Bit erstellen: `1000 1000` \Rightarrow `0x88`
3. Eingelesenes Byte und Maske bitweise UND-Verknüpfen
4. Wenn Ergebnis = 0 \Rightarrow kein Bit gesetzt
 Wenn Ergebnis = `0x08` \Rightarrow Bit 3 gesetzt
 Wenn Ergebnis = `0x80` \Rightarrow Bit 7 gesetzt
 Wenn Ergebnis = `0x88` \Rightarrow Bit 3 und 7 gesetzt

Aufgabe: Bit 5 in Port 0x0300 setzen. Dabei die anderen Bits unverändert lassen.

Lösungsschritte:

1. Einlesen des Bytes von Port 0x0300 mit `inportb(0x0300)`;
2. Maske für 5. Bit erstellen: `0010 0000` \Rightarrow `0x20`
3. Eingelesenen Byte und Maske bitweise ODER-Verknüpfen
4. Byte wieder auf Port 0x0300 mit `outportb(0x0300)` schreiben.

Aufgabe: Bit 2 in Port 0x0300 löschen. Dabei die anderen Bits unverändert lassen.

Lösungsschritte:

1. Einlesen des Bytes von Port 0x0300 mit `inportb(0x0300)`;
2. Maske für 2. Bit löschen erstellen: `1111 1011` \Rightarrow `0xFB`
3. Eingelesenen Byte und Maske bitweise UND-Verknüpfen
4. Byte wieder auf Port 0x0300 mit `outportb(0x0300)` schreiben

Bitmuster für gezieltes Löschen und Setzen von Bits

| Bit | Bit löschen mit bitweisem UND |
|-----|--|
| 0 | <code>1111 1110</code> \Rightarrow <code>0xFE</code> |
| 1 | <code>1111 1101</code> \Rightarrow <code>0xFD</code> |
| 2 | <code>1111 1011</code> \Rightarrow <code>0xFB</code> |
| 3 | <code>1111 0111</code> \Rightarrow <code>0xF7</code> |
| 4 | <code>1110 1111</code> \Rightarrow <code>0xEF</code> |
| 5 | <code>1101 1111</code> \Rightarrow <code>0xDF</code> |
| 6 | <code>1011 1111</code> \Rightarrow <code>0xBF</code> |
| 7 | <code>0111 1111</code> \Rightarrow <code>0x7F</code> |

| Bit | Bit setzen mit bitweisem ODER |
|-----|--|
| 0 | <code>0000 0001</code> \Rightarrow <code>0x01</code> |
| 1 | <code>0000 0010</code> \Rightarrow <code>0x02</code> |
| 2 | <code>0000 0100</code> \Rightarrow <code>0x04</code> |
| 3 | <code>0000 1000</code> \Rightarrow <code>0x08</code> |
| 4 | <code>0001 0000</code> \Rightarrow <code>0x10</code> |
| 5 | <code>0010 0000</code> \Rightarrow <code>0x20</code> |
| 6 | <code>0100 0000</code> \Rightarrow <code>0x40</code> |
| 7 | <code>1000 0000</code> \Rightarrow <code>0x80</code> |