

Tutoriumssitzung BIT 1 2020

Beatrice Inschakov, Dominik Sekular

Stellenwertsysteme

Stellenwert- systeme

Aufgabe:

Füllen Sie die nachfolgende Tabelle aus, indem Sie die Zahlen zwischen den Stellenwertsystemen umrechnen.

Binär	Oktal	Dezimal	Hexadezimal
100			
	57		
		511	
			10B

Stellenwert- systeme

Aufgabe:

Füllen Sie die nachfolgende Tabelle aus, indem Sie die Zahlen zwischen den Stellenwertsystemen umrechnen.

Binär	Oktal	Dezimal	Hexadezimal
100	4	4	4
0010 1111	57	47	2F
1 1111 1111	777	511	1FF
1 0000 1011	413	267	10B

Stellenwert- Systeme: typische Fehler

$$7_{10} = 111_2$$

tief gestellte Zahl, die das Stellenwertsystem angibt

Zweierkomplementdarstellung

Zweier- komplement

Aufgabe:

Wir betrachten Binärzahlen mit der Größe von genau 8 Bit.

Berechnen Sie die folgenden Zahlen im Zweierkomplement

a) 10_{10}

b) -10_{10}

c) 0_{10}

d) -0_{10}

e) -65_{10}

f) -129_{10}

Zweier- komplement

Aufgabe:

Wir betrachten Binärzahlen mit der Größe von genau 8 Bit.

Berechnen Sie die folgenden Zahlen im Zweikomplement

a) $10_{10} = (0000\ 1010)_Z$

b) $-10_{10} = (1111\ 0110)_Z$

c) $0_{10} = (0000\ 0000)_Z$

d) $-0_{10} = (0000\ 0000)_Z$

e) $-65_{10} = (1011\ 1111)_Z$

f) $-129_{10} = \text{nicht lösbar}$

Zweier- komplement: typische Fehler

Auffüllen auf die angegebene Bit-Zahl bedenken!

Abbildung der Zweierkomplement bei negativen Zahlen
nur wenn ausdrücklich gefordert!

tiefgestelltes „Z“ nicht vergessen!



Logikgatter

Logikgatter

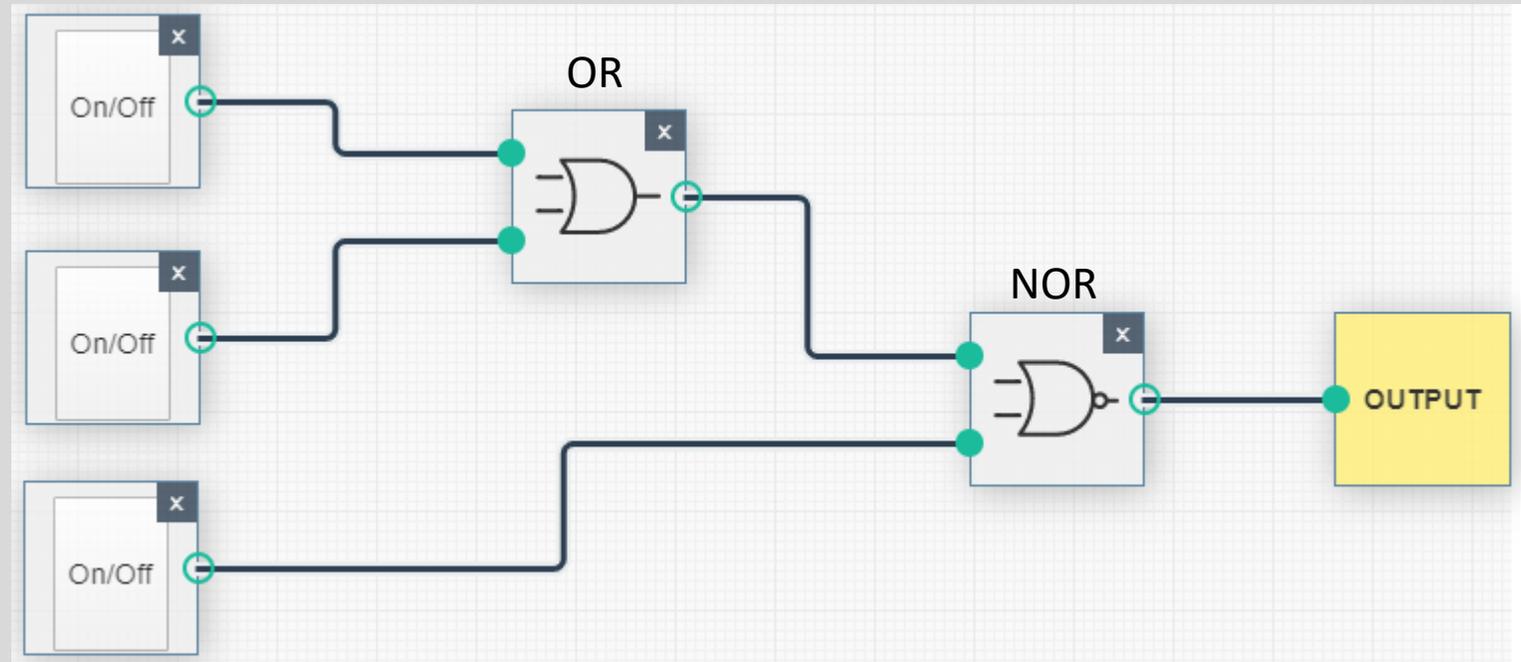
Aufgabe:

Skizzieren Sie eine Schaltung, bei der drei Inputs durch zwei Logikgatter so miteinander verbunden werden, dass „die Leuchte“ nur eingeschaltet ist, wenn alle Inputs deaktiviert sind.

alle am PC können auch dieses praktische Applet verwenden:

academo.org/demos/logic-gate-simulator/

Logikgatter



Logikgatter: typische Fehler

Verwechselln von Logikgattern miteinander

Assemblersprache

Assembler- sprache

Aufgabe:

Schreiben Sie ein Programm im AssemblerCode für den Sehr Einfachen Computer, welches eine beliebige Zahl (x) als Input annimmt und den Wert

$$(x^2) - 1$$

in den Output schreibt. Schreiben Sie Ihr Programm so, dass es nach Ausgabe des Ergebnisses für eine weitere Eingabe zur Verfügung steht.

Assembler- sprache

1		Jmp	Start	//Überspringe den Datenbereich
2	m:	1		//Wert, welcher am Ende subtrahiert werden soll
3	n:	0		//Wert, welcher dazu dient „x“ zu speichern
4	Start:	Read		//Input x nach A lesen
5		store	n	//Wert A auf n speichern
6		mul	n	//Wert A mit n(=A) multiplizieren = $A = n * n$
7		sub	m	//m von A subtrahieren
8		write		//A nach OUT schreiben
9		jmp	Start	//zum Anfang springen

Assembler- sprache: typische Fehler

Groß- Kleinschreibung bei Variablen berücksichtigen!

Dopplungen weglassen!

- load und read laden Daten in A
- müssen also nicht zwischengespeichert werden



Danke