

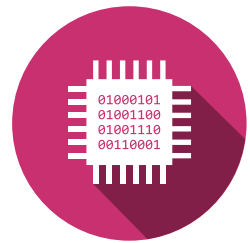
Binäre Addierer

Inhaltsverzeichnis

1 Ziel	1
2 Addierer mit Übertragsfortpflanzung	2
2.1 Schaltung	2
2.2 Ausführung	2
3 Subtrahierer	3
3.1 Schaltung	3
Akronyme	3

1 Ziel

Dieses Labor dient, den Entwurf von iterativen arithmetischen Schaltungen zu üben. Die erstellte Iterativschaltung ist ein Addierer. Das Labor zeigt ausserdem noch wie man die erstellten Addierer benutzt um einen Subtrahierer zu erstellen.



2 Addierer mit Übertragsfortpflanzung

2.1 Schaltung

Abbildung 1 zeigt die Schaltung eines **carry ripple-through adder** (Addierers mit Übertragsfortpflanzung). Er ist aufgeteilt in iterativen Blöcken, welche 2 Bits von gleichem Gewicht zusammen mit einem einkommenden Übertrag addieren, und welche einen Ausgangsübertrag und ein Bit der Summe erstellen.

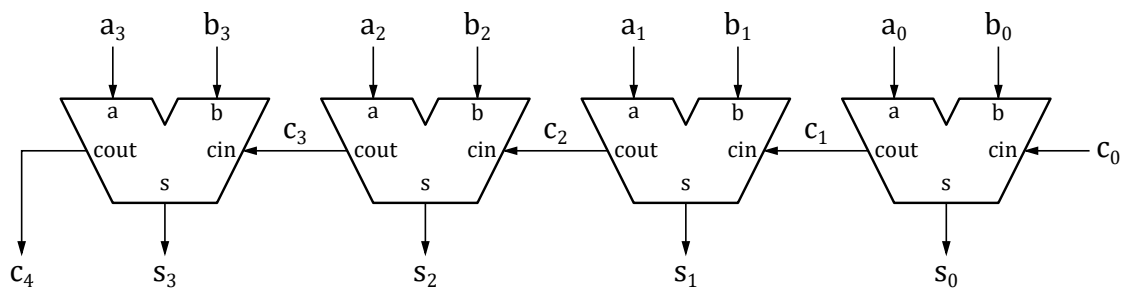
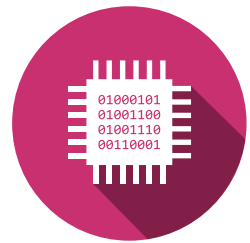


Abbildung 1: Addierer mit Übertragsfortpflanzung

2.2 Ausführung

Mit Hilfe von INV-, UND-, ODER- und XOR-Gattern, zeichnen Sie das Schema des iterativen Blocks.

Zeichnen Sie dann das iterative Schema eines 4-Bit Addierers mit Übertragsfortpflanzung und überprüfen Sie dessen richtigen Betrieb.



3 Subtrahierer

3.1 Schaltung

Die Schaltung eines Subtrahierers kann auf der Grundlage des zuvor entwickelten Addierers erfolgen. Um diese Subtraktion durchzuführen, kann man das Gegenzeichen der zu subtrahierenden Zahl addieren:

$$a - b = a + (-b) \quad (1)$$

Im Zweierkomplement erhält man das Gegenzeichen einer Zahl, indem man alle Bits der Zahl invertiert und zu diesem Zwischenergebnis 1 addiert. Die Inversion aller Bits erfolgt mit einem Inverter für jedes Bit. Die Addition von 1 kann erfolgen, indem man auf den allerersten Übertrag des Addierers einwirkt.

Im Block *sub8_tb* befinden sich 2 Addierer mit jeweils 4 Bits, die so verkettet sind, dass sie einen Addierer mit 8 Bits ergeben. Es gibt auch einen Block, um die Bits der Zahl *b* 4 bis 4 zu invertieren.

Erstellung Zeichne das Schema des Blocks *forSubtraction*, der die Bits der Zahl *b* invertiert. Überprüfe, ob der komplette Subtrahierer richtig funktioniert.

Akronyme

Addierers mit Übertragsfortpflanzung carry ripple-through adder. [2](#)