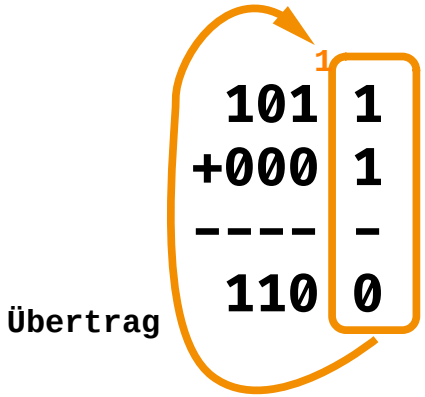
Adder

Halb- und Volladdierer

Beispiel einer bitweisen Addition:



Bei der kleinsten denkbaren Rechnung (hier 1+1=0 mit Übertrag 1) werden nur zwei Bits addiert. Dabei entsteht ein Bit für die Summe und eines für den Übertrag. Dies nennt man auch einen Halbaddierer.

a) Stelle Sie eine Wahrheitstabelle mit allen Eingabemöglichkeiten und allen zu erwartenden Ergebnissen dar. Bestimmen Sie den logischen Ausdruck für das Ergebnis-Bit und das Übertragsbit.

Für die Addition der ersten beiden Bits der obigen Beispielrechnung reicht der Halbaddierer noch. Für die nächsten beiden Bits schon nicht mehr. Warum?

Adder

Halb- und Volladdierer

b) Stellen Sie die Wahrheitstabelle eines Volladdierers auf.

c) Angeblich kann man einen Volladdierer aus zwei Halbaddierern bauen, wenn man noch ein logisches OR zusätzlich verwendet. Überlegen Sie, wie das gehen könnte.

d) Implementieren Sie den Volladdierer als Verilog Code und testen ihn.

© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Adder

2 Bytes addieren

e) Lagern Sie den getesteten Volladdierer als eigenes Modul aus. Beispiel:

```
module mein_modul(
     input  my_in_1, my_in_2
     output my_out
     );

     assign my_out = my_in_1 ^ my_in_2; // irgend ein Beispiel
endmodule
```

Irgendwo z.B. in Ihrem Top-Modul

```
...
mein_modul(HW_switch_0, 1, HW_led_0);
...
```

Im Prinzip wird der Code aus mein_modul lediglich textuell an der Stelle des Aufrufs eingesetzt. Daraus entsteht dadurch jedes mal eine neue (Teil-)Schaltung durch die auf dem FPGA Wahrheitstabellen verbraucht werden.

© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Adder

2 Bytes addieren

e) (fortgesetzt) Rufen Sie Ihr Modul zum Test in Ihrem Top-Modul auf. Wenn das funktioniert, wollen wir das Volladdierermodul durch wiederholten Aufruf dazu verwenden zwei 8-Bit Zahlen zu addieren. Mit den rechten 8 Schaltern definiert der Benutzer das 1. Byte, mit den linken 8 das 2. Byte. Die ersten 9 LEDs von rechts sollen das Ergebnis darstellen (siehe Schema d. nächsten Seite). Warum 9 LEDs?

Hinweis: Das Übertragsbit des letzten Volladdierers muss irgendwie als Eingabe in den nächsten. Für diese Verbindung brauchen Sie einen Draht, den sog. Wire.

```
module beispiel(
  input  my_in_1, my_in_2,
  output my_out
  );

  wire mein_draht;
  assign mein_draht = my_in_1 & my_in_2;
  assign my_out = mein_draht;
endmodule
```

Ihren **`wire mein_draht`** können Sie auch in eigene Module als Eingabe- oder Ausgabeport einsetzen oder ihn wie z.B. einen Schalter in logischen Ausdrücken verwenden. Obwohl der wire 0 oder 1 sein kann, speichert er nichts. Er leitet lediglich die Information von etwas anderem weiter.

Adder

2 Bytes addieren

e)

