

Übungsblatt 7

Bearbeitung ab Montag, 7. September 2020

7.1 Multiplizierwerk in VHDL

Das Multiplizieren ist eine der häufigsten Rechenoperationen die wir in unserem täglichen Leben durchführen. Ist die Grundregel des Multiplizierens einmal verstanden, kann das Produkt von zwei beliebigen Zahlen einfach errechnet werden. Abbildung 1(a) zeigt beispielhaft eine Multiplikation im Dezimalsystem mit Multiplikator 13 und Multiplikand 12 und dem Ergebnis 156.

$$\begin{array}{r}
 & & 1101 \\
 & & \times 1100 \\
 \begin{array}{r} 13 \\ \times 12 \\ \hline 26 \\ + 13 \\ \hline 156 \end{array} & & \begin{array}{r} 0000 \\ 0000 \\ 1101 \\ + 1101 \\ \hline 10011100 \end{array}
 \end{array}$$

(a) Multiplikation im Dezimalsystem.

(b) Multiplikation im Binärsystem.

Abbildung 1: Beispiele der Multiplikation.

Da Computer jedoch im Binärsystem rechnen, müssen diese auch im Binärsystem multiplizieren können. Nur zur Darstellung für den Menschen wird zwischen dem Binär- und Dezimalsystem konvertiert. In Abbildung 1(b) ist die binäre Multiplikation mit den Multiplikator 13 und Multiplikand 12 bzw. 1101 und 1100 dargestellt. Zu erkennen ist auch hier, dass in jeder Zeile eine Stelle des Multiplikanden mit dem vollständigen Multiplikator multipliziert wird. Anschließend werden die, je nach Stelle verschobenen, Teilprodukte aufsummiert um das Ergebnis zu berechnen.

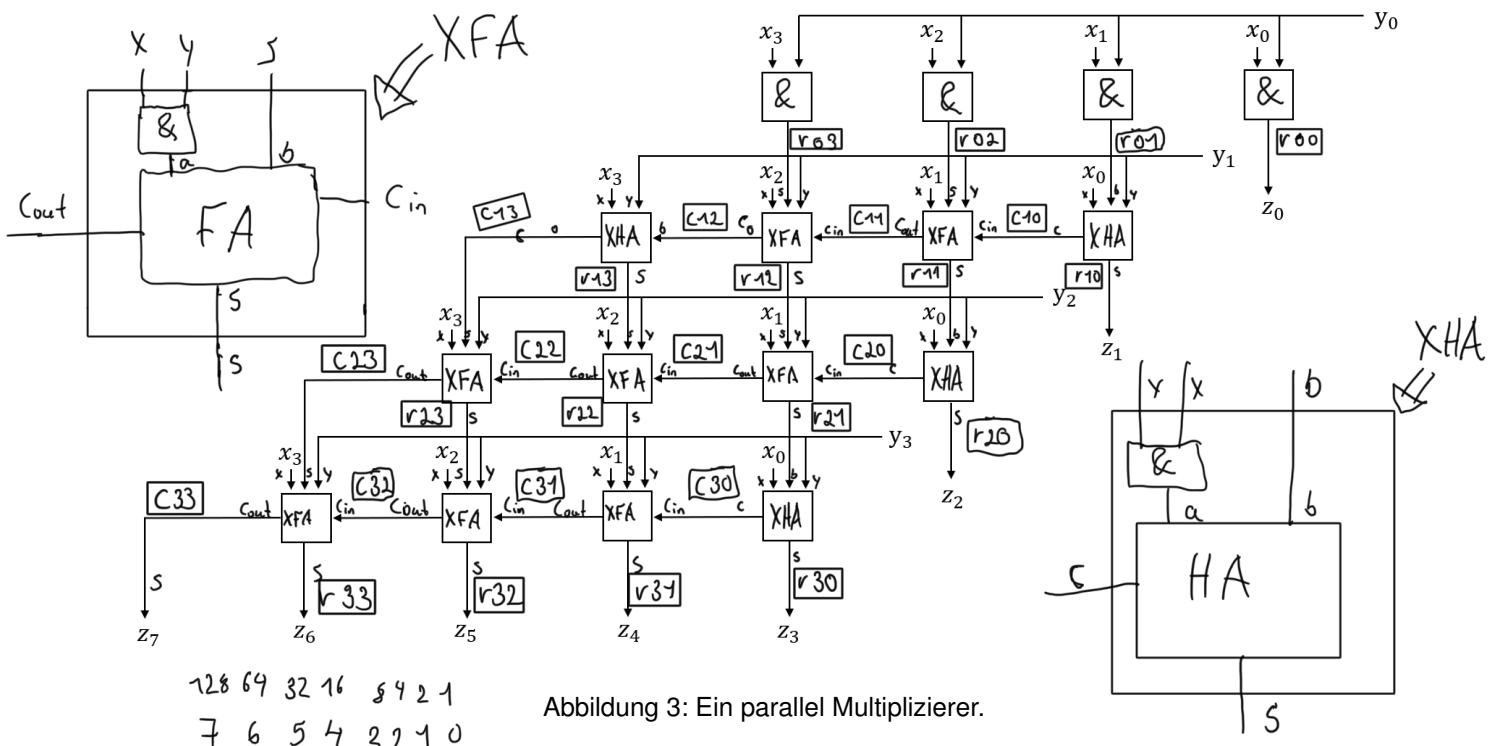
$$\begin{array}{r}
 & & x_3 & x_2 & x_1 & x_0 \\
 & & \times & y_3 & y_2 & y_1 & y_0 \\
 \hline
 & & x_3y_0 & x_2y_0 & x_1y_0 & x_0y_0 \\
 + & & x_3y_1 & x_2y_1 & x_1y_1 & x_0y_1 \\
 + & & x_3y_2 & x_2y_2 & x_1y_2 & x_0y_2 \\
 + & & x_3y_3 & x_2y_3 & x_1y_3 & x_0y_3 \\
 \hline
 z_7 & z_6 & z_5 & z_4 & z_3 & z_2 & z_1 & z_0
 \end{array}$$

Abbildung 2: Beispiel einer 4-Bit Multiplikation.

Aufgaben:

Abbildung 2 zeigt das allgemeine Vorgehen der Multiplikation von zwei 4-Bit Zahlen $x_3x_2x_1x_0$ und $y_3y_2y_1y_0$ mit dem Produkt z_7z_0 .

- a. (4 Punkte) Der parallel Multiplizier ist einer der häufigsten Realisierungen. In Abbildung 3 ist das Blockschaltdiagramm für 4-Bit Multiplikator $x_3x_2x_1x_0$, 4-Bit Multiplikand $y_3y_2y_1y_0$ und 8-Bit Ergebnis z_7z_0 angegeben. Untersuche die Schaltung, ergründe die Funktionsweise der einzelnen Blöcke und vervollständige die Blöcke mit dir bekannten Gattern oder Kombinationen von Gattern.



- b. (8 Punkte) Implementiere den vollständigen 4-Bit parallel Multiplizierer und teste diesen mit den Eingaben $x_3x_2x_1x_0 \in \{1010, 1001\}$ und $y_3y_2y_1y_0 \in \{1101, 0111\}$.

- c. (2 Punkte) Eine andere Realisierung des Multiplizierwerks ist die *add shift* Methode, welches Schieberegister und Addierwerke nutzt. Erkläre wie das Multiplizierwerk in Abbildung 4 funktioniert und zeichne ein Flussdiagramm, welches die Funktionsweise abbildet.

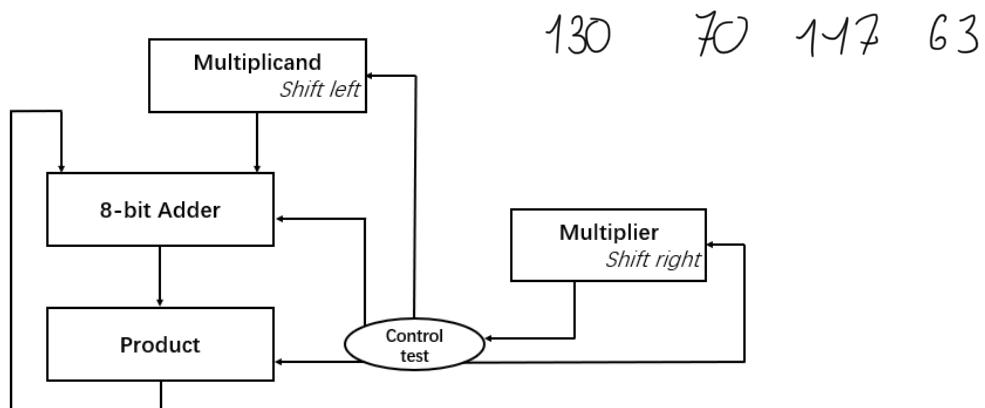


Abbildung 4: Multiplizierwerk nach *add shift* Methode.

- d. (2 Punkte) Vergleiche den *add shift* Multiplizierer aus Abbildung 4 mit dem parallel Multiplizierer aus Abbildung 3. Gebe die Vor- und Nachteile der jeweiligen Realisierung an.

- e. (8 Punkte) Implementiere das Multiplizierwerk in VHDL mithilfe des Schieberegisters und Addierwerks aus dem vorherigen Aufgabenblatt und teste die Schaltung erneut mit den Eingaben $x_3x_2x_1x_0 \in \{1010, 1001\}$ und $y_3y_2y_1y_0 \in \{1101, 0111\}$.