Digitale Schaltungen
Sebastian Pointner, Robert Wille
sebastian.pointner@jku.at

21. November 2018
Institut für Integrierte Schaltungen
Johannes Kepler Universität

Übung 4

Abgabe bis **Mittwoch, 28. November 15:30** via EPIIC: `http://ep.iic.jku.at`.

In der letzten Vorlesung haben wir die arithmetischen Grundschaltungen kennen gelernt. In dieser Übung wollen wir nun diese Grundschaltungen verwenden, um eine sogenannte Arithmetische Logische Einheit (kurz ALU) zu erstellen. Eine ALU bildet einen Grundbestandteil heutiger Rechner und realisiert wie der Name schon sagt, arithmetische (z.B. eine Addition) oder logische Operationen (z.B. ein AND). Neben den arithmetischen- und logischen Operationen verfügt eine ALU auch über Logik, die dazu verwendet werden kann, die beiden Eingänge der ALU direkt miteinander zu vergleichen. Die zu entwerfende ALU arbeitet im Zweierkomplement – kann also mit positiven, als auch mit negativen Zahlen umgehen.

Die ALU soll nun in den einzelnen Teilen der Übung Schritt für Schritt erstellt werden. Dabei werden in den ersten Teilen die Komponenten der ALU erstellt und diese dann, im letzten Schritt zu einer Gesamtschaltung zusammengefügt.

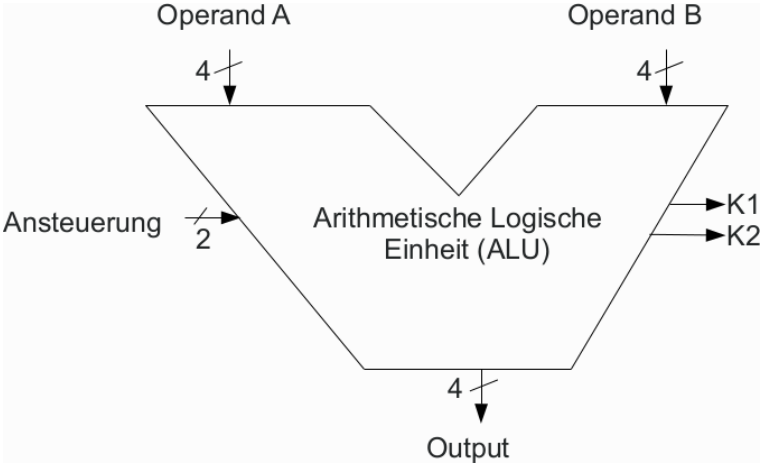

Abbildung 1: Blockschaltbild einer ALU

Das Blockschaltbild in Abbildung 1 zeigt eine ALU, die mit einer Bit-Breite von 4-Bit arbeitet. Neben den beiden Operanden A und B verfügt die ALU über eine 2-Bit breite *Ansteuerung*, welche die Operation auswählt. Die Ausgänge der ALU bilden der 4-Bit Ausgang *Output*, welcher das Ergebnis der Berechnungen ausgibt, sowie die beiden Komparator-Ausgänge K_1 und K_2, welche die beiden Eingänge in Relation zu einander setzten.

1. Ansteuerung der Operationen (2)
Zur Realisierung der *Ansteuerung* wollen wir einen MUX-2 Multiplexer verwenden, wie er in Abbildung 2 dargestellt wird.

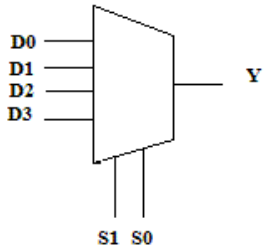

Abbildung 2: Multiplexer mit 2-Bit breiter Ansteuerung.

(a) Zeichne das Schaltbild eines MUX-2 Multiplexers und verwende dazu ausschließlich MUX-1 als Basisblöcke.

2. Addition und Subtraktion (2 + 2 + 2)
Die ALU soll vier verschiedene Operationen realisieren. Um die Addition und die Subtraktion abzudecken, verwenden wir ein kombiniertes Additions-Subtraktionswerk auf Basis des Carry-Ripple Adders. Mittels dieses Additions-Subtraktionswerks können wir sowohl die Addition, als auch die Subtraktion im Zweikomplement mit einer Komponente abdecken.

(a) Führe die Subtraktion mit folgenden Binärzahlen im Zweierkomplement durch:

$$\begin{array}{rcccc} & 1 & 0 & 1 & 0 \\ - & 0 & 1 & 0 & 1 \\ \hline \end{array}$$

$$\begin{array}{rccccc} & 1 & 1 & 0 & 0 & 0 \\ - & 0 & 1 & 1 & 1 & 0 \\ \hline \end{array}$$

(b) Leite die Wahrheitstabelle eines 1-Bit Vollsubtrahierers her. Ein Vollsubtrahierer besitzt, wie ein Volladdierer, drei 1-Bit Eingänge und zwei 1-Bit Ausgänge. Der Ausgang D (Difference) stellt dabei die Differenz zwischen den Eingängen dar und der Ausgang B_o (Borrow out) zeigt an, ob ein Borrow notwendig ist. Neben dem Eingängen A und B berücksichtigt der Subtrahierer auch ein Borrow-in (B_i) am Eingang.

(c) Nun wollen wir aber versuchen die Realisierung der Subtraktion auf eine Addition zurückzuführen. Um die Addition und die Subtraktion von zwei 4-Bit Zahlen im Zweierkomplement durchzuführen verwenden wir das Additions-Subtraktionswerk, welches wir in der Vorlesung kennengelernt haben. Erstelle ein 4-Bit Additions-Subtraktionswerk unter Verwendung von Volladdierern und beliebigen anderen Gattern.

Tipp: Verwende dabei den 4-Bit Carry-Ripple Adder der letzten Übung als Ausgangspunkt,

Verwende das Additions-Subtraktionswerk nun dazu um die beiden Zweierkomplement-Zahlen $1001_{2k} - 1101_{2k}$ zu subtrahieren. Markiere jeweils den Pegel an den Ein- und Ausgängen der Addierer und Gatter.

3. Multiplikation im Zweierkomplement (4 + 1)
Neben der Addition und der Subtraktion soll unsere ALU auch eine Multiplikation im Zweierkomplement durchführen können. Im Gegensatz zur Addition und zur Subtraktion soll die Multiplikation nur für 2-Bit Operanden durchgeführt werden, da das Ergebnis ansonsten nicht in 4-Bit darstellbar ist.

(a) Entwerfe einen parallelen Multiplizierer, der zwei 2-Bit Zahlen im Zweierkomplement am Eingang akzeptiert und das Ergebnis als 4-Bit Zahl berechnet. Verwende dazu Halb- und Volladdierer und beliebige andere Logikgatter.

(b) Verwende den Multiplizierer und berechne das Produkt von 01 und 11. Markiere jeweils den Pegel an den Ein- und Ausgängen der Addierer und Gatter,

4. Komparatoren (2 + 2)
Neben dem 4-Bit Ausgang *Output* verfügt die ALU über zwei 1-Bit Komparator-Ausgänge ($K1$ und $K2$), welche die Eingänge A und B miteinander vergleichen. Ist der Wert des Eingangs $A < B$ wird der Ausgang $K1$ auf den logischen Wert 1 gesetzt, ansonsten wird der Ausgang auf den Wert einer logischen 0 gesetzt. Das Gleiche gilt für den Ausgang $K2$, ist $A \geq B$ wird der Ausgang auf eine logische 1 gesetzt, ansonsten auf eine 0. Verwende dafür Subtrahierer beliebiger Bitbreite und beliebige andere Logikgatter.

Realisiere folgende Komparatoren:

(a) Kleiner Operator ($K1 = A < B$)

(b) Größer Gleich Operator ($K2 = A \geq B$)

5. Zusammensetzung und Testen der ALU (3 + 4)
Im letzten Schritt soll nun die ALU zusammengesetzt werden. Für diesen Schritt sollen die bis jetzt entworfenen Komponenten in Form von Blockschaltbildern verwendet werden und so zusammengeschalten werden, dass eine ALU mit folgendem Verhalten entsteht:

- Ist die Ansteuerung 00, so sollen Operanden A und B addiert werden und am Ausgang Output angelegt werden.
- Ist die Ansteuerung 01, so sollen Operanden A und B subtrahiert (A - B) werden und am Ausgang Output angelegt werden.
- Ist die Ansteuerung 10, so sollen die beiden niederwertigsten Bits der Operanden A und B multipliziert werden und das Ergebnis in Form einer 4-Bit Zahl am Ausgang Output angelegt werden.
- Ist die Ansteuerung 11, so sollen am Ausgang Output die Ver-Oderung der beiden Operatoren A und B angelegt werden.
- Verwende vier Multiplexer MUX-2 um die Werte je nach Ansteuerung an den Ausgang *Output* zu legen.

- Unabhängig vom Wert der Ansteuerung sollen die beiden Komparatorausgänge die Relation der beiden Eingänge zueinander anzeigen. Verdrahte diese direkt mit den Ausgängen $K1$ und $K2$.

Tipp: Verwende das LSB der Ansteuerung, um die Operation im Additions-Subtraktionsrechenwerk zu konfigurieren.

(a) Zeichne das Schaltbild der ALU und verwende dazu die zuvor entworfenen Komponenten in Form von Blockschaltbildern und beliebige andere Logikgatter.

(b) Teste die ALU mit folgenden Eingaben:

- *Ansteuerung*: 00 Operand A: 1110 Operand B: 0111
- *Ansteuerung*: 01 Operand A: 1111 Operand B: 1110
- *Ansteuerung*: 10 Operand A: 1010 Operand B: 1101
- *Ansteuerung*: 11 Operand A: 1010 Operand B: 0101

Gib jeweils den Wert von *Output*, $K1$ und $K2$ an. Zeichne dafür jeweils die Ein- und Ausgangswerte der jeweiligen Blockschaltbilder in das Schaltbild ein. Verwende dafür verschiedene Farben, um den Gedankengang nachvollziehbar zu gestalten.