

Aufgaben 3.2

Bau einer 4-Bit ALU

Es ist eine 4-Bit ALU (Arithmetic and Logic Unit) zu erstellen, die nachfolgende Berechnungen – via die Selektoreingänge S auswählbar – ausführen kann:

- A or B, bitweise (für S = 0), stets mit Overflow Flag OF=0
- A and B, bitweise (für S = 1), stets mit Overflow Flag OF=0
- not A, bitweise (für S = 2), stets mit Overflow Flag OF=0
- not B, bitweise (S für = 3), stets mit Overflow Flag OF=0
- A plus B (A, B: 4-bit unsigned integer), mit Overflow Flag OF (für S = 4)
- A minus B (A, B: 4-bit unsigned integer), mit Underflow Flag OF (für S = 5)
- 2 mal A, mit Overflow Flag OF (für S = 6)
- 4 mal A, mit Overflow Flag OF (für S = 7)

A ist der erste 4-Bit Operand, B der zweite 4-Bit Operand. R ist das 4-Bit Resultat. OF repräsentiert das 1-Bit Flag Overflow für die Addition und Underflow für die Subtraktion, S ist der Operations-Selektor. Die 7-Segment-Anzeigen zeigen Operanden und Resultat in hexadezimal an.

Hinweise zur Abgabe

Sie erhalten die Datei *Teamname 4-Bit ALU.circ*, die zwingend zu verwenden ist. Insbesondere ist für die Abgabe zu beachten:

- Als Hauptschaltung muss das Schaltungsblatt *4-Bit ALU* gewählt sein. Dort ist ein zu wählender Teamname einzutragen und die Mailadressen aller Team-Mitglieder anzugeben.
- Nehmen Sie eine Umbenennung von *Teamname 4-Bit ALU.circ* vor. Teamname ersetzen Sie mit dem gewählten Namen.
- In der Interface-Ansicht (Symbol mit Maus-Tip-Anzeige "*Aussehen der angezeigten Schaltung als Teilschaltung bearbeiten*") des Schaltungsblatts *4-Bit ALU* ist "Team Name" im Symbol mit Ihrem gewählten Teamnamen zu ersetzen. Ansonsten sind hier keine weiteren Änderungen zugelassen.
- Im Schaltungsblatt *Interface* und in Unterschaltungen realisieren Sie ihre 4-Bit ALU.

Weitere Hinweise

- Die Aufgabe wird (i.d.R.) in Vierergruppen realisiert. Drei Personen realisieren die 8 Subsaltungen, die vierte Person ist für die Integration der Subsaltungen in eine ALU verantwortlich.
- Die Schaltung soll auch getestet werden.
- Die Subsaltungen können unabhängig realisiert werden und müssen untereinander nicht optimiert sein.
- Subsaltungen können als *Logisim Libraries* eingesetzt werden (vgl. Help). Es ist dann darauf zu achten, dass alle Files in entsprechenden Ordnern abgegeben werden (Ganzes als Zip, Rar o.ä. abgeben).
- Die im Unterricht abgegebene ALU (Skript) darf nicht verwendet werden. Auch Schaltungen aus den Bibliotheken *Arithmetik* und *Speicher* von Logisim sind hier nicht zugelassen, alle andern Schaltungen hingegen schon.

Schliesslich

- Bezeichnen Sie Anschlüsse und Bausteine.
- Stellen Sie Ihre Schaltungen übersichtlich dar. I.A. gilt: Je weniger Leitungsüberkreuzungen umso übersichtlicher.

→

Testing

Logisim bietet die Möglichkeit, die Korrektheit einer Schaltung mit Black Box Tests zu prüfen. Für diese Aufgabe wurde durch den Dozenten eine Testbench-Schaltung erstellt. Mittels eines Zählers werden alle möglichen Eingangsmuster erzeugt, welche an eine Referenzschaltung (Master) und Ihre Schaltung angelegt werden. Darauf werden die Resultate verglichen. Bei Ungleichheit der Resultate oder bei korrekt durchgelaufenem Vergleich wird abgebrochen (alles ok: Zählerstand 7FF und LED im Zustand grün).

Die Vergleichsschaltung sieht wie folgt aus:

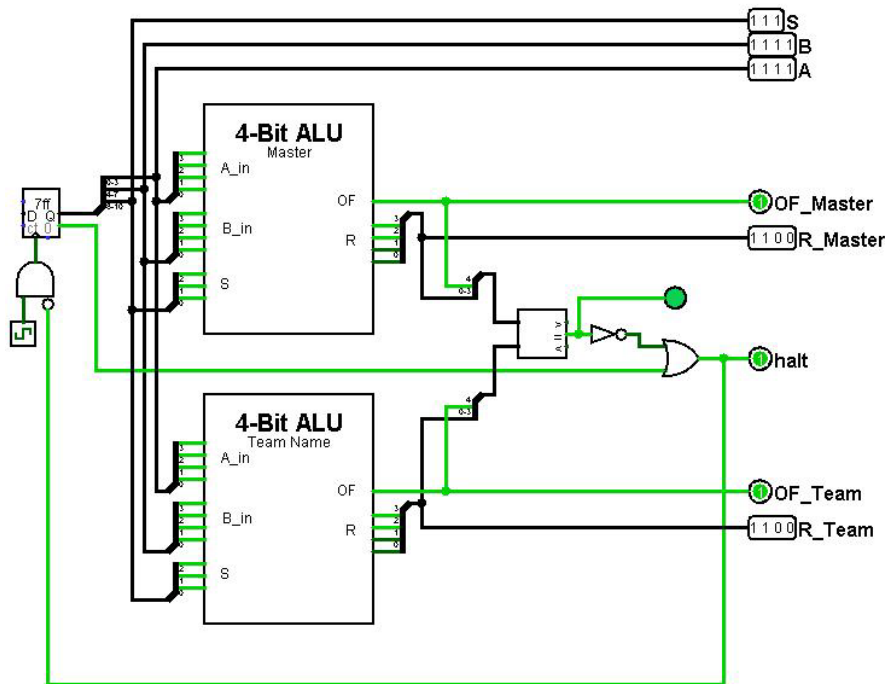


Fig. 1: Erfolgreich abgeschlossener Test

Aus obiger Figur wird nun auch ersichtlich, weshalb es wichtig ist, dass Sie Ihre Datei gemäss den Punkten unter *Hinweise zur Abgabe* abliefern.

Eigentlich wäre es sinnvoll die Testbench abzugeben, so dass Sie den Test gleich selbst vornehmen könnten. Dagegen spricht, dass es in Logisim nicht möglich ist, die Masterschaltung (eine mögliche Lösung) kompiliert, transmogrifiziert o.ä. abzugeben.