Übung 4

Stellen Sie Spezifikation und Implementierung der ALU4 aus Übung 3 fertig.

Verifikation der 4-Bit ALU

In dieser Übungseinheit wird die Funktion der 4-Bit ALU verifiziert. Dazu müssen für alle möglichen Eingangsbelegungen die Ausgabesignale der Zelle mit den Sollwerten verglichen werden. Da dies in unserem Fall bei 15 Eingängen zu 2^{15}=32768 verschiedenen Eingangsbelegungen führt, ist ein „händisches“ Durchtesten nicht praktikabel. Deshalb erstellen wir in dieser Übung eine so genannte „Testbench“. Darunter versteht man ein VHDL-Programm im Strukturstil, das keine externen Anschlüsse besitzt, sondern eine interne Komponente zur Signalerzeugung enthält. Wenn für einen Entwurf sowohl die Spezifikation als auch die Implementierung als VHDL-Code vorliegen, lassen sich Ist- und Soll-Werte wie folgt vergleichen:

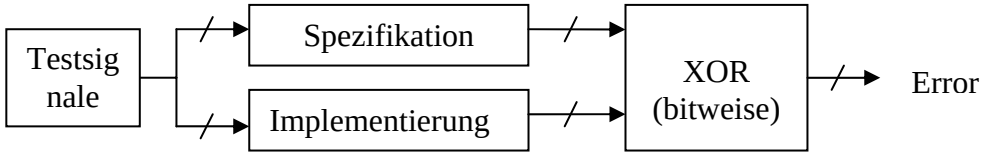

Abbildung 12: Aufbau der Testbench

Um eine solche Testbench für unsere 4-Bit ALU zu erzeugen gehen Sie wie folgt vor:

- Erzeugen Sie ein neues Blockschaltbild für die **entity alu4_tb**
- Platzieren Sie zwei **ALU4** Zellen sowie ein Feld von 8 **XOR2** Zellen auf dem Arbeitsbereich.
- Erzeugen Sie drei globale Bus-Signale. Benennen Sie diese Signale anschließend mit **error(7:0), control(2:0)** bzw. **stimulus(11:0)** und verbinden Sie die Zellen wie folgt:

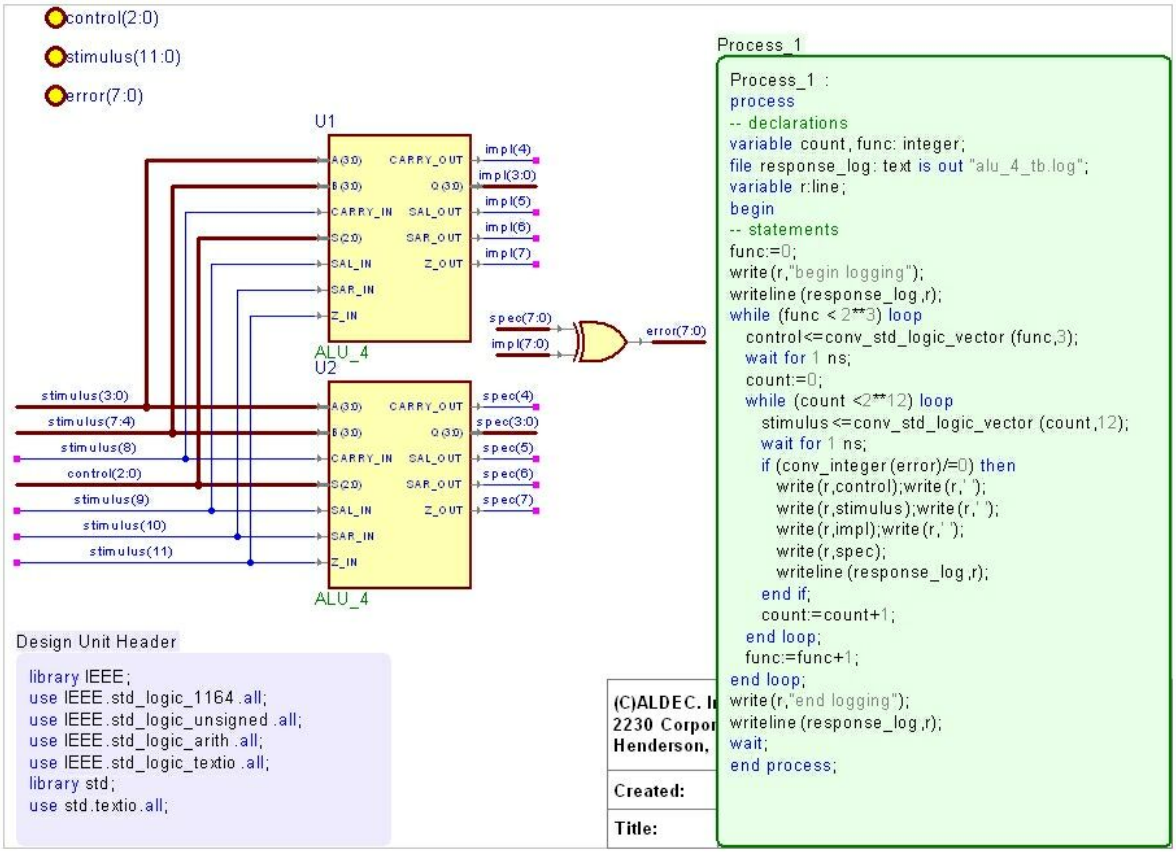

Abbildung 13: Testbench mit Testpattern-Generator

- Wählen Sie nacheinander die beiden **ALU4** Symbole aus und öffnen Sie über die Rechte Maustaste die Einstellungen(*Properties)*. Wählen Sie für **U1** die **architecture impl** während Sie für **U2** die **architecture spec** wählen. Aktivieren Sie jedes Mal die Option *Generate configuration specification*, um die Konfiguration auch im generierten VHDL-Code festzulegen.
- Fügen Sie dem Blockschaltbild noch einen Prozess für die Generierung der Testmuster hinzu (*Diagram->VHDL->Process*). Der Prozess soll zudem bei Auftreten eines Fehlers die aktuellen Signalzustände in eine Log-Datei schreiben.
- Um die Dateizugriffe ausführen zu können benötigen Sie noch einige Bibliotheken. Erzeugen Sie dazu die entsprechenden Einträge im Design Header (*Diagram->VHDL->Design Unit Header*). Verwenden Sie für Dateizugriffe das Paket std.textio
- Benutzen Sie zur Erstellung des Prozesses den Quelltext im Bild als Vorlage.

Auswertung der Testbench

Die nun erzeugte Testbench muss noch simuliert werden:

- Selektieren sie dazu im Design Browser die Datei *alu4_tb.bde*, übersetzen Sie diese und wählen sie die **entity alu4_tb** als **top-level-unit** für die Simulation.
- Initialisieren Sie die Simulation und wählen Sie *Run* im Menü *Simulation*. Nach dem Beenden des Prozesses beenden Sie die Simulation und laden anschließend die Log-Datei in den Editor, um eventuelle Fehler im Design zu entdecken.
- Korrigieren Sie ihren Entwurf bis Sie eine fehlerfrei funktionierende 4-Bit ALU erhalten haben.