

## Anleitung bis zu Milestone 2

### Synthesizer Projekt

1. Bestimmen Sie wer welchen Block entwirft und teilen Sie die Gruppe entsprechend ein. Folgende Blöcke müssen entworfen werden:
  - a. Schieberegister p2s und s2p. Die Schieberegister sind separate Entitäts und werden im I2s Master Block je 2 mal instanziiert (links und rechts).
  - b. i2s Master Block
  - c. Path Control
2. Synthetisieren Sie jeden Block zuerst mit Quartus, bevor sie diesen integrieren.
3. Sobald eine Person mit Ihrem Design fertig ist, kann diese mit der Integration der Blöcke in *synthi\_top* beginnen. Die Entity des Blocks *i2s\_master.vhd* ist bereits gegeben.
4. Synthetisieren Sie *synthi\_top* nach der Integration der neuen Blöcke *path\_control* und *i2s\_master*.
5. Auch mit der Testbench kann sofort begonnen werden, sobald eine Person mit Ihren Designaufgaben fertig ist.
6. **Es soll die gleiche Testbench wie für I2C verwendet werden, sie müssen keine neue Testbench erzeugen.** Auf der Testbench müssen lediglich die *i2s\_sim* und *i2c\_chk* Procedures eingebaut und mit den Top-Level Signalen verbunden werden.
 

```

      elsif cmd = string'("i2s_sim") then
        i2s_sim(tv, AUD_ADCLRCK, AUD_BCLK, AUD_ADCDAT);
      elsif cmd = string'("i2s_chk") then
        i2s_chk(tv, AUD_ADCLRCK, AUD_BCLK, AUD_DACDAT, dacdat_check);
      
```
7. Sichern Sie die *testcase.dat* Datei, die Sie für I2c benutzt haben zunächst in eine Datei *testcase\_i2c.dat*, damit diese später wieder benutzt werden kann. Beginnen Sie mit einer neuen *testcase.dat* für die i2s Tests.
8. Fügen Sie im Skript *compile.do* die neuen Dateien für die Testbench hinzu. Lassen Sie die anderen Dateien darin.
9. Erstellen sie eine neue Datei *wave.do*, wo Sie nur die relevanten Signale anzeigen.
10. Kontrollieren Sie vor der Simulation, ob alle Instanziierungen richtig angezeigt werden.

