

Entwurf und Implementierung digitaler Schaltungen mit VHDL

4. praktische Übung im SoSe 2016

„Projekt“

Lösungspräsentation am 18. bzw. 19.7.2016 in der Vorlesungszeit

👉 Präsentation (ppt, odp, pdf) und Code (zip-Datei) bis zum 16.7.2016 (23:59)
im Abgabesystem hochladen (<http://fb05-lehre.hpi.uni-potsdam.de>)

👉 Lösungsabgabe in den bekannten Gruppen wie bisher

Sie haben die *Wahl* zwischen einem *Video-Projekt* und einem *Audio-Projekt*.

1. Video-Projekt

Ihre Aufgabe besteht darin, über PS/2-Maus oder -Tastatur gesteuert auf einem am Nexys-Board angeschlossenen VGA-Display „etwas“ anzuzeigen.

Dabei kann „etwas“ *zum Beispiel* sein

- ASCII-Zeichen,
- geometrische Formen,
- „Farbeffekte/Farbspiele“,
- „echte“ Spiele (wie Pong, Minecraft, ...),
- ...

Wichtig ist, dass die Ausgabe durch das von Ihnen gewählte Eingabemedium (Maus oder Tastatur) *beeinflussbar* ist, also *zum Beispiel*

- über Tastatur eingegebene ASCII-Zeichen fortlaufend angezeigt werden,
- geometrische Formen über die Maus positionier- bzw. veränderbar sind (z.B. in Größe, Farbe, ...)
- Farbspiele beeinflussbar bzw. veränderbar sind,
- „echte“ Spiele im Spielsinn beeinflussbar sind.

Quellenhinweis:

- Nexys4 Reference Manual
 - 9 VGA Port (S. 14)
 - 8.3 Mouse (S. 13)

2. Audio-Projekt

Ihre Aufgabe besteht in diesem Fall darin, mit dem Nexys-Board, einer PS/2-Tastatur und einem Lautsprecher ein kleines „E-Piano“ zu implementieren, welches mindestens folgende Funktionalität bietet:

- „Piano-Modus“:
Über 13 Tasten der Tastatur sollen die Töne c' bis c'' - nebst Halbtönen - abrufbar sein. Die Frequenzen der Töne sind in der Tabelle unten angegeben. Dabei soll ein Ton beim Drücken der zugeordneten Taste jeweils so lange erklingen, bis die Taste wieder losgelassen wird.
- „Player-Modus“:
Über eine weitere Taste soll das *automatische* Abspielen einer *bekannten* Melodie startbar sein.
- Umschaltung zwischen Player- und Piano-Modus.

Weitere Variationen und „Klangspiele“ sind willkommen (aber nicht erforderlich, sozusagen „Kür“). Hier einige Anregungen:

- Den Namen des aktuell gespielten Tons auf der 7-Segmentanzeige ausgeben (ggf. auch noch die drei zuvor gespielten),
- Veränderung der Lautstärke,
- Akkorde (Polyphonie),
- Speichern einer gespielten Ton-Sequenz oder einer selbst gesungenen Melodie („Recorder-Modus“),
- Anschluss SD-Card oder USB-Stick: Einlesen und Abspielen der gespeicherten MP3-Dateien ☺

Quellenhinweis:

Nexys4 Reference Manual

- 15 Microphone (S. 24)

- 16 Mono Audio Output (S. 26)

Ton-name	Frequenzverhältnis zu c'	reine Stimmung
c'	1/1	264 Hz
cis'	25/24	275 Hz
d'	9/8	297 Hz
es'	6/5	317 Hz
e'	5/4	330 Hz
f'	4/3	352 Hz
fis'	25/18	367 Hz
g'	3/2	396 Hz
as'	8/5	422 Hz
a'	5/3	440 Hz
b'	9/5	475 Hz
h'	15/8	495 Hz
c''	2/1	528 Hz

Quelle:

[http://de.wikipedia.org/wiki/Ton_\(Musik\)](http://de.wikipedia.org/wiki/Ton_(Musik))

Hinweise zur Präsentation:

Bei der Präsentation sollen Sie

- die Architektur Ihrer Schaltung kurz an einem Blockschaltbild (RT-Ebene) erläutern (möglichst via ppt), wobei Sie sich
 - auf die „*Highlights*“ („Alleinstellungsmerkmale“) konzentrieren, und
 - NICHT nur den RTL-View von ISE zeigen sollen, sondern ein aussagekräftiges RT-Blockschaltbild.
- falls Sie das Audio-Projekt gewählt haben:
 - die gespeicherte Melodie erklingen lassen, und
 - im Piano-Modus *selbst* eine *bekannte* Melodie vorspielen.
- falls Sie das Video-Projekt gewählt haben:
 - die Beeinflussbarkeit Ihrer Displayausgabe über Tastatur/Maus demonstrieren.

!!!!!!! Die Präsentation sollte insgesamt nicht länger als **7 Minuten** dauern !!!!!!!

Weitere Hinweise:

- Speichern Sie Ihre Lösung *dauerhaft* im (EE)PROM (auch „Flash PROM“, „Platform Flash“, „SPI Flash“ oder „Serial Flash“ genannt) und bringen Sie Ihr so vorbereitetes Board zu Präsentation mit.

Eine Anleitung zur Programmierung des Flash-Speichers finden Sie ab S. 18 im Dokument *Adderzelle_VHDL_ISE_14_2016(2).pdf* unter *R:\VHDL\ISE*.

- Es gibt keine Testgespräche (werden soz. durch Ihre Präsentation ersetzt).



Wir wünschen viel Erfolg in
harmonischer Schwingung
bei bestem Bilde!

