

# Entwurf und Implementierung digitaler Schaltungen mit VHDL

## 1. praktische Übung im SoSe 2016

### Die Lösung

- drei zip-Dateien mit je einem „konfigurations- bzw. simulationsfertigen“ ISE14-Projekt -  
bis zum 21.05.2016 (23:59 Uhr) *einzel*n hochladen auf:

\* wird noch bekanntgegeben \*

☞ Hinweise auf der letzten Seite beachten

☞ Lösungsabgabe in den Gruppen, die jeweils zusammen  
ein Board ausgeliehen haben (also i.w. in Dreiergruppen)

### 1. Aufgabe: Multiplexverbindung (eine einzige Verhaltensspezifikation auf RT-Ebene)

Realisieren Sie mit dem Xilinx-Board eine funktionierende Multiplexverbindung mit vier Ein- und Ausgangssignalen, indem Sie einen MUX und einen DEMUX entsprechender Dimension geeignet verschalten (eine einzige Verhaltensspezifikation auf RT-Ebene, also keine Instanziierung und Verschaltung von Komponenten) .

Reichen Sie als Lösung ein vollständiges Projekt „MUX-Verb“ ein, wobei die Eingangs-, Ausgangs- und Steuersignale genau so benannt sind, wie es in der Schaltung auf Seite 97 des GdS-Skripts vorgegeben ist. Legen Sie dabei (mit PlanAhead oder direkt im ucf-File) folgende Zuordnung fest:

I/O Ports				
	Name	Direction	Neg Diff Pair	Site
[-]	All ports (12)			
[-]	ax (2)	Input		
[-]	ax[2]	Input		R15
[-]	ax[1]	Input		M13
[-]	ay (2)	Input		
[-]	ay[2]	Input		L16
[-]	ay[1]	Input		J15
[-]	Xa (4)	Input		
[-]	Xa[4]	Input		M17
[-]	Xa[3]	Input		P18
[-]	Xa[2]	Input		N17
[-]	Xa[1]	Input		M18
[-]	Ye (4)	Output		
[-]	Ye[1]	Output		N14
[-]	Ye[2]	Output		J13
[-]	Ye[3]	Output		K15
[-]	Ye[4]	Output		H17

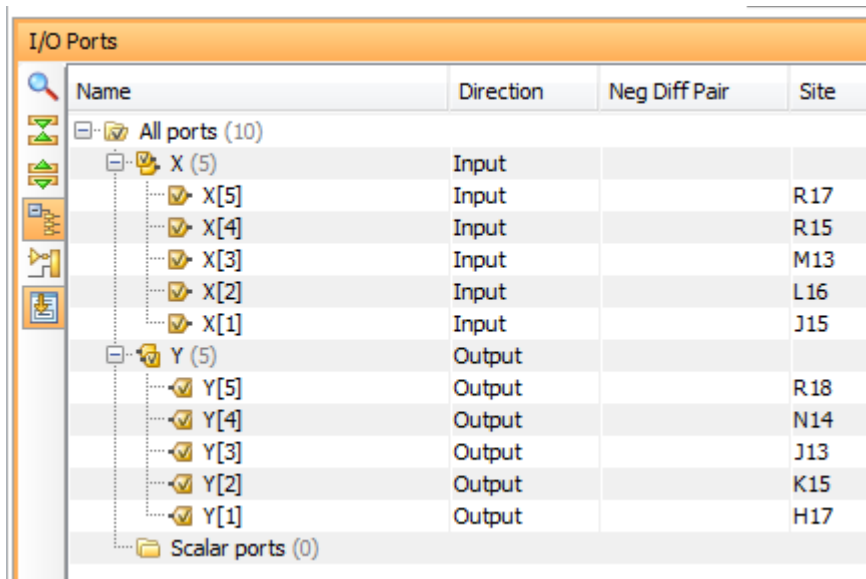
! Das Projekt muss so vorbereitet sein, dass man zum Funktionstest den FPGA-Baustein nur noch (mit „iMPACT“) konfigurieren muss.

## 2. Aufgabe: Iteratives Schaltnetz zur Bildung des Zweierkomplements

Realisieren Sie mit dem Xilinx-Board eine funktionierende Schaltung zur Bildung des Zweierkomplements für eine 5-Bit-Vokabel, und zwar als iteratives Schaltnetz gemäß Übung 1, Aufgabe 3.

Die iterative Struktur soll durch Instanziierung und entsprechende Verschaltung von fünf Zellen erzeugt werden; das Verhalten einer Zelle ist durch eine Funktionstabelle zu spezifizieren.

Reichen Sie als Lösung ein vollständiges Projekt „Komp“ ein. Benennen Sie die Signale so, wie es in der Musterlösung (von A.3 in Ü.1) vorgegeben ist. Legen Sie dabei (mit PlanAhead oder direkt im ucf-File) folgende Zuordnung fest:



Name	Direction	Neg Diff Pair	Site
<b>All ports (10)</b>			
<b>X (5)</b>	Input		
X[5]	Input		R17
X[4]	Input		R15
X[3]	Input		M13
X[2]	Input		L16
X[1]	Input		J15
<b>Y (5)</b>	Output		
Y[5]	Output		R18
Y[4]	Output		N14
Y[3]	Output		J13
Y[2]	Output		K15
Y[1]	Output		H17
<b>Scalar ports (0)</b>			

! Das Projekt muss so vorbereitet sein, dass man zum Funktionstest den FPGA-Baustein nur noch (mit „IMPACT“) konfigurieren muss.

## 3. Aufgabe: Hasardsimulation

Die in der Vorlesung besprochene Gatterschaltung des 2:1-Multiplexers weist bekanntlich für den Übergang  $X_a = (x_3, x_2, x_1) = 101$  nach  $X_e = 010$  einen dynamischen Dreivariablen-Funktionshasard auf.

Erstellen Sie ein VHDL-Modul, welches bei Simulation dieses Übergangs einen Hasardfehler am Ausgang der Schaltung zeigt.

Gehen Sie hierzu wie folgt vor:

- Erstellen Sie zunächst ein VHDL-Modul für die Gatterschaltung, wie sie auf S. 40 im VHDL-Skript angegeben ist, und zwar mit den internen Signalen  $b = x_2$ ,  $c = x_2x_1$ ,  $e = x_3x_1$ ,  $d = x_3b$  (wobei  $y = c \vee e \vee d$  gilt).

Stellen Sie folgende Verzögerungszeiten ein (transport delay):

- UND-Gatter für die Signale  $c$ ,  $d$  und  $e$  sowie das ODER-Gatter für  $y$ : 10ns,
- Inverter: 5ns.

- Erklären Sie die Eingangssignale  $x_3 - x_1$  (zum Zwecke einer einfachen Simulation) zu internen Signalen und weisen Sie diesen jeweils eine Werte-Sequenz („after“) so zu, dass der gewünschten Übergang von  $X_a$  nach  $X_e$  zum Zeitpunkt  $t = 30$  ns stimuliert wird.

b.w.

- Stellen Sie nun auf den Signalpfaden der Signale  $x_3$  bis  $x_1$  (also jeweils vom Schaltungseingang bis zum ersten Gatter, das erreicht wird) die Verzögerungen - unter Einführung weiterer interner Signale - so ein, dass die Simulation einen Hasardfehler am Ausgang  $y$  der Schaltung zeigt. (Wobei man sich vorstellen kann, dass diese Verzögerungen beispielsweise durch reale Leitungsverzögerungen verursacht werden.) Sie müssen natürlich nur dort verzögern, wo es zur Provokation des Hasardfehlers erforderlich ist. Beachten Sie auch, dass „ $x_2$ “ über zwei Pfade in die Schaltung läuft“.

Reichen Sie als Lösung ein „simulationsfertiges“ Projekt „Hasard“ ein, welches bei Simulation mit dem ISim-Simulator den Hasardfehler unmittelbar zeigt.

#### Hinweise:

- Falls Probleme beim Starten von ISim auftreten: manchmal hilft „Project/Cleanup Project Files“ :-)
- Sie dürfen, wenn Sie wollen, auch mit einer „testbench“ arbeiten.

#### Allgemeine Hinweise:

- Lösungsabgabe in den Gruppen, die jeweils zusammen ein Board ausgeliehen haben (also i.w. in **Dreiergruppen**).  
Dazu setzen wir wieder das Moodle-Abgabesystem ein.

*! Hinweise zur Nutzung des Abgabesystems erhalten Sie im Laufe der nächsten Woche per Email.*

- Testatgespräch:

- Jede Gruppe muss ihre Lösungen in Rahmen eines kurzen Testatgesprächs präsentieren (wobei jeder Gruppenteilnehmer zu jeder Aufgabe befragt werden kann). Jede Gruppe muss dazu vollständig erscheinen (also nicht nur ein Stellvertreter).
- Die Testate finden im Zeitraum **23.05. - 27.05.2016** statt.
- Die Testatgespräche finden in der Kommunikationszone A-E statt.
- Zur Präsentation Ihrer Lösungen bitte einen eigenen Laptop mitbringen.

Wir wünschen viel Erfolg und Freude beim Entwerfen !