

6. Übung: More Power!

Name(n):

Punkte:

1 Architekturelle Verbesserungen

In der letzten Übung haben Sie eine einfache Möglichkeit kennen gelernt, mit der die Leistungsaufnahme eines sequentiellen Entwurfs verringert werden kann. Ziel dieser Übung ist nun, durch *architekturelle* Veränderungen eines Entwurfs eine weitere Verbesserung zu erreichen.

Sie wissen, dass die Leistungsaufnahme eines Hardwareentwurfs von mehreren Komponenten abhängt:

$$P \sim \alpha \cdot f \cdot C \cdot V_{dd}^2.$$

Während, solange der Fertigungsprozess nicht geändert wird, die Lastkapazität C und die Spannung V_{dd} schwer zu beeinflussen sind, liegen die Taktfrequenz f und die Schaltaktivität α mehr oder weniger im Einflussbereich des Entwicklers. In dieser Übung sollen Sie anhand eines einfachen Entwurfs eine Möglichkeit zur Beeinflussung dieser beiden Größen kennen lernen.

Gegeben sei dazu ein Entwurf, der für zwei Eingabekanäle X und Y , die aus je zwei 32-Bit-Werten (X_a, X_b bzw. Y_a, Y_b) bestehen, jeweils das Produkt der beiden Werte (also $X_a \cdot X_b$ bzw. $Y_a \cdot Y_b$) berechnet. Um einen möglichst platzsparenden Entwurf realisieren zu können, wird in diesem Entwurf nur ein Multiplizierer verwendet, und die Kanäle werden abwechselnd berechnet (siehe Abbildung 1). Das Ein-/Ausgabeverhalten ist in Abbildung 2 dargestellt.

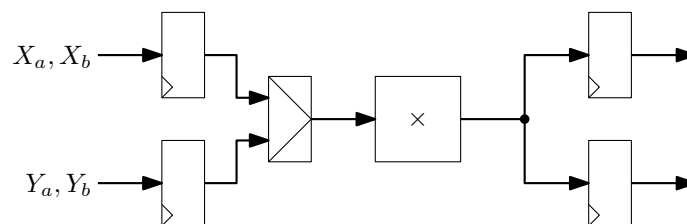


Abbildung 1: Serielle Implementierung des Entwurfs.

Gesucht ist eine architekturelle Verbesserung, um die Stromaufnahme des gegebenen Entwurfs verbessern zu können: das *Resource Sharing* soll vermieden werden (siehe Abbildung 3).

Gegeben seien nun vier verschiedene Klassen von Eingabedaten, die anhand der Intra- und Interkanalkorrelation eingeteilt sind:

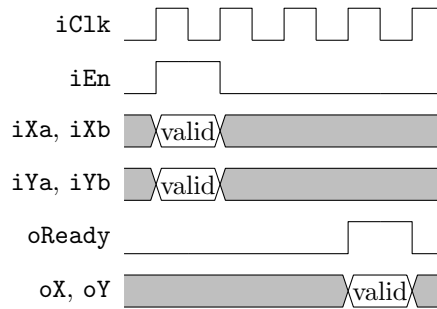


Abbildung 2: Ein-/Ausgabeverhalten des Entwurfs.

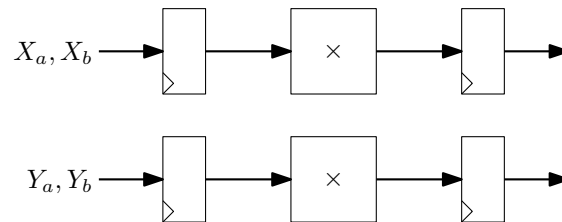


Abbildung 3: Parallele Implementierung des Entwurfs.

		Interkanalkorrelation	
		ja	nein
Intrakanalkorrelation	ja	$x = y = \sin(t)$	$x = 1, y = -1$
	nein	$x = y = \text{rand}()$	$x = \text{rand}(), y = \text{rand}()$

Überlegen Sie, welche Architektur für welche dieser Klassen von Eingabedaten vorteilhaft ist, und wie wahrscheinlich eine entsprechende Verteilung der Eingabedaten ist. Dokumentieren Sie diese

□ Überlegungen!

Im Moodle steht Ihnen eine Datei zur Verfügung, die folgendes beinhaltet:

- Sourcen: Testbench, Multipliers-e.vhd und Multipliers-RtlSer-a.vhd
- Scripts: Synthese, RTL- und GL-Simulation und Power-Analyse

Implementieren Sie nun eine parallele Version, die sich nach außen gleich verhält wie der serielle Entwurf: das Timing und die Verwendung der Steuersignale (wie in Abbildung 2 gegeben) darf also nicht verändert werden! Sinnvollerweise können Sie auch den gegebenen Register-Satz übernehmen.

Führen Sie für beide Varianten der Implementierung nun eine Power-Simulation durch. Um dies zu automatisieren stehen Ihnen zusätzliche Skripts zur Verfügung (all_syn.sh, all_sim.sh und all_pt_sim.sh). Interpretieren Sie die Ergebnisse und stellen Sie sie grafisch dar!

□

2 PROL16: Even less power!

Wenden Sie nun Ihre Erkenntnisse aus dem obigen Beispiel auf Ihren PROL16 an: implementieren Sie mindestens eine architekturelle Veränderung, durch die der Entwurf stromsparender wird. Erklären

- ☐ Sie, wie und warum Sie die Verbesserung ausgewählt haben!