

Aufgabe 2.10: FPGA

Gegeben ist eine einfache Version eines Xilinx FPGAs bestehend aus CLBs und Switch-Matrixs. Ein CLB enthält eine 8 elementige LUT mit 3 Eingängen.

a) Realisieren Sie einen Volladdierer mittels zwei CLBs.

Ein Volladdierer besitzt drei Eingänge (x , y , c_{in}) und zwei Ausgänge (c_{out} , s). Es werden sämtliche Eingänge addiert und das Ergebnis als 2-Bit Zahl auf die Ausgänge gelegt. So ist $s=1$ wenn die Summe aller Eingänge ungerade ist und $c_{out}=1$ wenn die Summe aller Eingänge ≥ 2 ist.

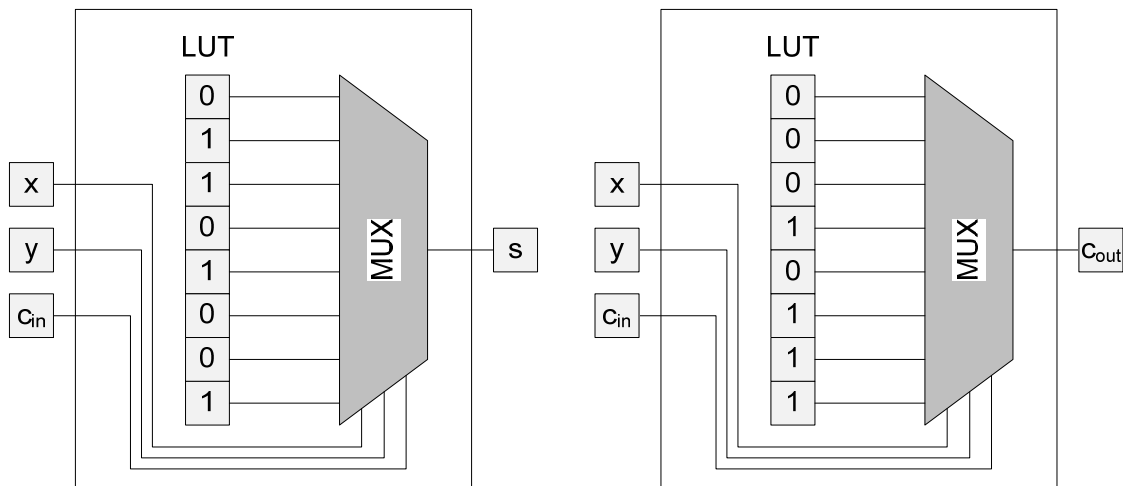


Figure 2.1: LUT Realisierung für einen Volladdierer

Hardware/Software Co-Design

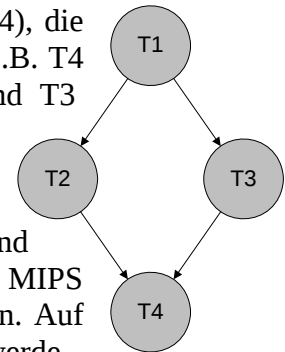
Übungsskript

Institut für Technik der Informationsverarbeitung, Karlsruher Institut für Technologie

Kapitel 3: Schätzung der Entwurfsqualität

Aufgabe 3.01: Design Space, Pareto Punkte

Die rechte Abbildung zeigt einen Task-Graphen mit vier Tasks (T1 ... T4), die auf verschiedenen Komponenten ausgeführt werden können. Dabei kann z.B. T4 erst aufgeführt werden, wenn T2 und T3 abgearbeitet wurden. T2 und T3 hingegen können parallel ausgeführt werden.



Die folgende Tabelle zeigt alle verfügbaren Komponenten (MIPS, DSP, FPGA, ASIC), die maximale Anzahl einer Komponente, deren Kosten und die Ausführungsgeschwindigkeit der vier Tasks. Zum Beispiel kostet der MIPS Prozessor 200 Einheiten und kann Task T1 in 5 und T4 in 2ms ausführen. Auf einer Komponente kann jeweils nur ein Task zur gleichen Zeit ausgeführt werden.

Komponente	Anzahl	Kosten [€]	Ausführungszeit [ms]			
			T1	T2	T3	T4
MIPS	1	200	5	-	-	2
DSP	1	100	-	20	18	5
FPGA	1	250	-	12	10	-
ASIC	1	400	-	-	0,8	-

Tabelle 3.1: Eigenschaften der verfügbaren Komponenten

- a) Vervollständigen Sie die Ausführungszeit und Kosten der folgenden Tabelle. Sie zeigt sämtliche Realisierungsmöglichkeiten, welcher Task auf welchem Prozessor ausgeführt werden kann.

#	Tasks				Ausführungszeit [ms]	Kosten
	T1	T2	T3	T4		
1	MIPS	DSP	DSP	MIPS	$5 + 20 + 18 + 2 = 45$	$200+100= 300$
2	MIPS	DSP	DSP	DSP	$5 + 20 + 18 + 5 = 48$	$200+100= 300$
3	MIPS	DSP	FPGA	MIPS	$5 + \max(20, 18) + 2 = 27$	$200+100+250= 550$
4	MIPS	DSP	FPGA	DSP	$5 + \max(20, 18) + 5 = 30$	$200+100+250= 550$
5	MIPS	DSP	ASIC	MIPS	$5 + \max(20, 0.8) + 2 = 27$	$200+100+400= 700$
6	MIPS	DSP	ASIC	DSP	$5 + \max(20, 0.8) + 5 = 30$	$200+100+400= 700$
7	MIPS	FPGA	DSP	MIPS	$5 + \max(12, 18) + 2 = 25$	$200+250+100= 550$
8	MIPS	FPGA	DSP	DSP	$5 + \max(12, 18) + 5 = 28$	$200+250+100= 550$
9	MIPS	FPGA	FPGA	MIPS	$5 + 12 + 10 + 2 = 29$	$200+250= 450$
10	MIPS	FPGA	FPGA	DSP	$5 + 12 + 10 + 5 = 32$	$200+250+100= 550$
11	MIPS	FPGA	ASIC	MIPS	$5 + \max(12, 0.8) + 2 = 19$	$200+250+400= 850$
12	MIPS	FPGA	ASIC	DSP	$5 + \max(12, 0.8) + 5 = 22$	$200+250+400+100=950$

Tabelle 3.2: Ausführungszeit und Kosten für sämtliche Realisierungsmöglichkeiten

- b) Tragen Sie die Lösungen aus Aufgabe a) in folgendes Zeit-Kostendiagramm ein und markieren Sie die Pareto-Punkte.

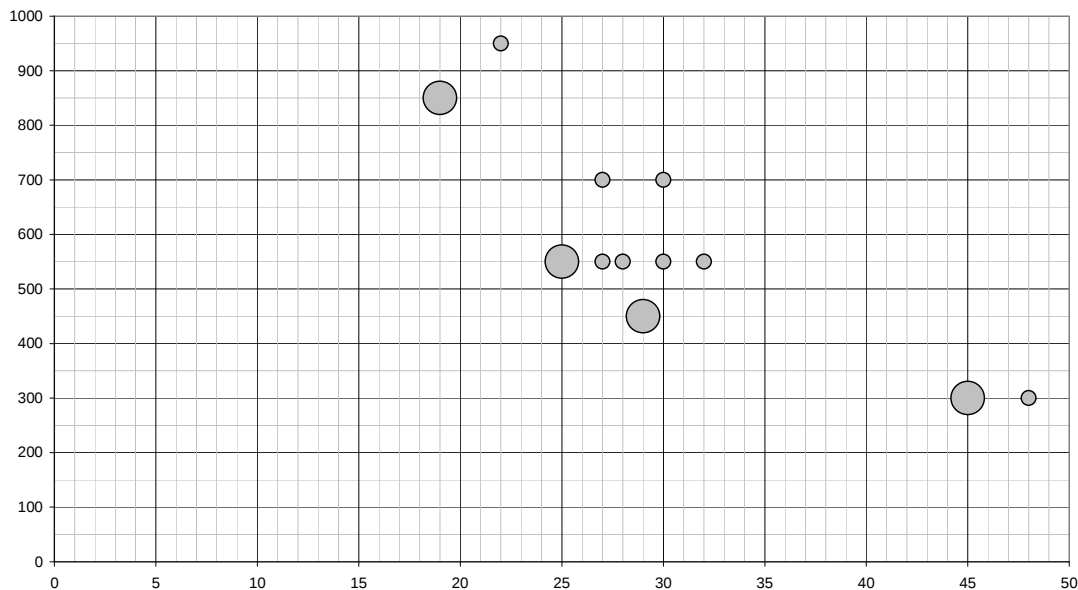


Figure 3.1: Kosten - Zeit Diagramm

- c) Was passiert, wenn die Anzahl der Komponenten nicht auf 1 beschränkt ist? Welche zusätzlichen Design-Möglichkeiten ergeben sich? Verändert sich die Menge der Pareto Punkte?

#	Tasks				Ausführungszeit [ms]	Kosten
	T1	T2	T3	T4		
13	MIPS	DSP	DSP	MIPS	$5 + \max(20, 18) + 2 = 27$	$200+100+100= 400$
14	MIPS	DSP	DSP	DSP	$5 + \max(20, 18) + 5 = 30$	$200+100+100= 400$
15	MIPS	FPGA	FPGA	MIPS	$5 + \max(12, 10) + 2 = 19$	$200+250+250= 700$
16	MIPS	FPGA	FPGA	DSP	$5 + \max(12, 10) + 5 = 22$	$200+250+250+100=800$

Tabelle 3.3: Ausführungszeit und Kosten für zusätzliche Realisierungsmöglichkeiten

Dadurch ändert sich die Menge der Pareto Punkte

- Lösung #15 ist genauso schnell wie Lösung #11 kostet aber weniger.
- Lösung #13 überdeckt Lösung #9 in Kosten und Ausführungszeit.

Aufgabe 3.02: Exaktheit & Treue

In folgender Tabelle sind für vier Entwurfspunkte Metriken und Entwurfsqualität dargestellt, und zwar geschätzt Werte $E(D)$ sowie die gemessenen Werte $M(D)$.

Entwurfspunkt	$E(D)$	$M(D)$
W	112	100
X	128	137
Y	139	121
Z	205	132

Tabelle 3.4: Geschätzte $E(D)$ und gemessene $M(D)$ Werte

- a) Bestimmen Sie die Exaktheit (A) des Entwurfspunktes W.

$$A_W = 1 - \frac{|E(D_W) - M(D_W)|}{|M(D_W)|} = 1 - \frac{|112 - 100|}{|100|} = 1 - \frac{12}{100} = \frac{88}{100} = 0,88$$

- b) Bestimmen Sie die Treue (F) des Schätzverfahrens.

$$\begin{aligned} F &= 100\% \cdot \frac{2}{n(n-1)} \cdot \sum_{i=1}^n \sum_{j=i+1}^n \mu_{i,j} \\ &= 100\% \cdot \frac{2}{4 \cdot 3} \cdot (\mu_{W,X} + \mu_{W,Y} + \mu_{W,Z} + \mu_{X,Y} + \mu_{X,Z} + \mu_{Y,Z}) \\ &= 100\% \cdot \frac{1}{6} \cdot (1 + 1 + 1 + 0 + 0 + 1) = 100\% \cdot \frac{4}{6} = 67\% \end{aligned}$$

Aufgabe 3.03: Taktschlupf

Gegeben ist eine Menge von funktionale Einheiten vk. Die mögliche Taktperiode Ihrer Zieltechnologie liegt zwischen 20 und 50 ns.

Funktionale Einheit	k	delay(v _k) [ns]	occ(v _k)
MUL	1	135	9
ADD	2	45	10
SUB	3	55	1

Tabelle 3.5: Eigenschaften der funktionalen Einheiten

- a) Was ist der Taktschlupf?

Der Taktschlupf bezeichnet den Anteil einer Taktperiode, der von einer funktionalen Einheit nicht ausgenutzt wird.

- b) Was bringt die Taktschlupfminimierung?

Bei der Taktschlupfminimierung wird versucht den durchschnittlichen Taktschlupf pro Operation zu minimieren. Dadurch steigt die Taktauslastung der Hardware und somit der Performanz.

- c) Berechnen Sie den „slack“ für alle funktionalen Einheiten bei einer Taktperiode von 20ns.

$$\begin{aligned} \text{slack}(20\text{ns}, v_{\text{MUL}}) &= (\lceil \text{delay}(v_{\text{MUL}}) / 20 \rceil * 20 - \text{delay}(v_{\text{MUL}})) \\ &= (\lceil 135 / 20 \rceil * 20 - 135) = 5 \\ \text{slack}(20\text{ns}, v_{\text{ADD}}) &= (\lceil \text{delay}(v_{\text{ADD}}) / 20 \rceil * 20 - \text{delay}(v_{\text{ADD}})) \\ &= (\lceil 45 / 20 \rceil * 20 - 45) = 15 \\ \text{slack}(20\text{ns}, v_{\text{SUB}}) &= (\lceil \text{delay}(v_{\text{SUB}}) / 20 \rceil * 20 - \text{delay}(v_{\text{SUB}})) \\ &= (\lceil 55 / 20 \rceil * 20 - 55) = 5 \end{aligned}$$

- d) Berechnen Sie den mittleren Schlupf (avgslack) für eine Taktperiode von 20ns.

$$\begin{aligned} \text{avgslack}(T) &= \frac{\sum_{k=1}^{|V_T|} (\text{occ}(v_k) * \text{slack}(20, v_k))}{\sum_{k=1}^{|V_T|} \text{occ}(v_k)} \\ \text{avgslack}(20\text{ns}) &= \frac{\text{occ}(v_{\text{MUL}}) * \text{slack}(20\text{ns}, v_{\text{MUL}}) + \text{occ}(v_{\text{ADD}}) * \text{slack}(20\text{ns}, v_{\text{ADD}}) + \text{occ}(v_{\text{SUB}}) * \text{slack}(20\text{ns}, v_{\text{SUB}})}{\text{occ}(v_{\text{MUL}}) + \text{occ}(v_{\text{ADD}}) + \text{occ}(v_{\text{SUB}})} \\ &= \frac{9 * 5\text{ns} + 10 * 15\text{ns} + 1 * 5\text{ns}}{9 + 10 + 1} = \frac{45\text{ns} + 150\text{ns} + 5\text{ns}}{20} = 10\text{ns} \end{aligned}$$

e) Raten/Überlegen Sie, welche Taktperiode den niedrigsten mittleren Schlupf hat?

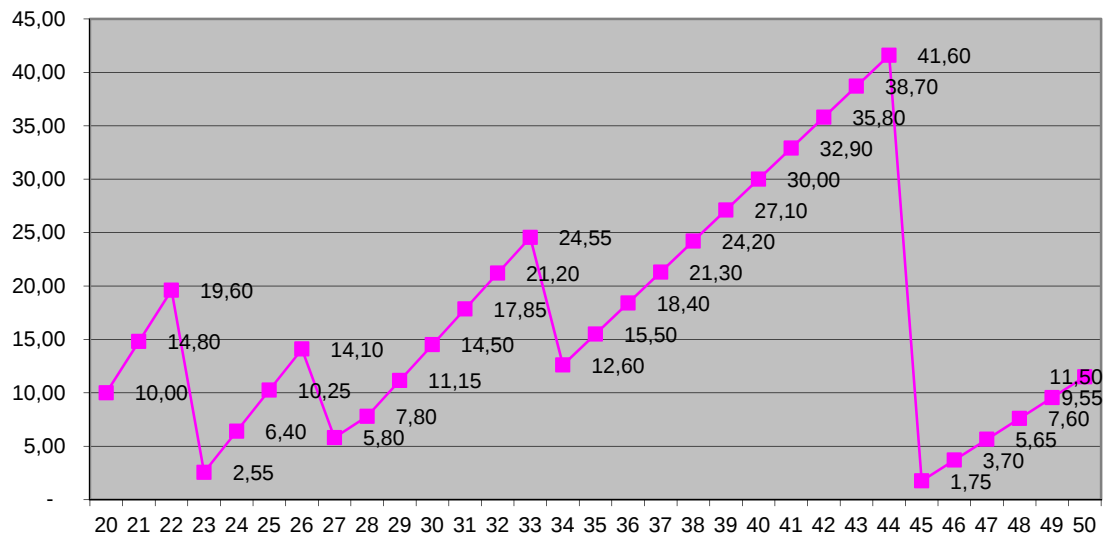


Abbildung 3.1: Niedrigster mittlerer Taktschlupf bei verschiedenen Taktperioden