Lösungshinweise zum Aufgabenblatt 2: VHDL

Ziel: VHDL-Programme lesen und verstehen, Automaten-Entwurf, Zeitmodell in VHDL

Zu Aufgabe 1: Entity „A1“

a) Der Automat ist ein **Mealy-Automat**, da das Ausgangssignal an einen Übergang gekoppelt ist und nicht an einen Zustand. Den Zustandsübergangsgraphen für den Automaten sehen Sie in der *Abbildung 1* weiter unten.

b) Die Entity beschreibt einen Automaten, der in den Zuständen den Rest der Division mit 5 angibt. Die Zahl bliebiger Länge wird seriell, mit dem MSB (most significant bit) zuerst, an den Automaten übergeben. Dieser liest bitweise die Zahl ein, und gibt dabei jeweils bis zu dem Zeitpunkt berechneten Rest aus.

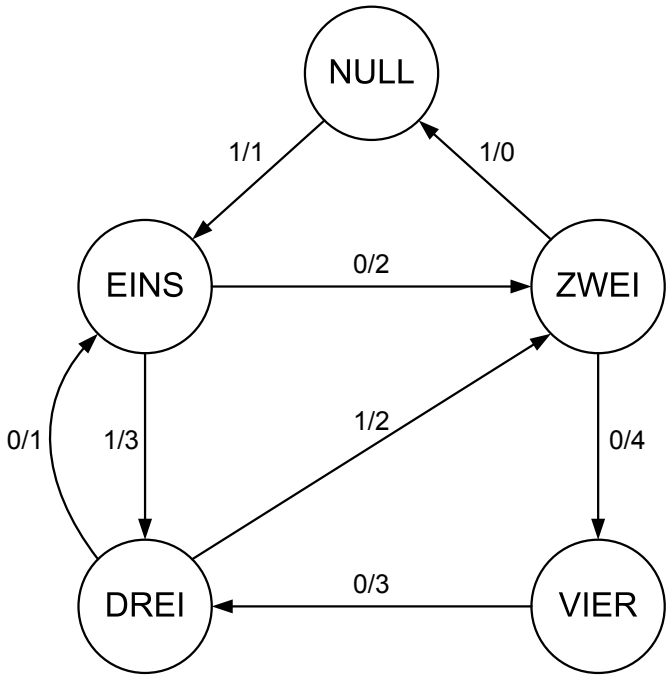

Abbildung 1: Mealy Automat

Zu Aufgabe 2: Package „TSL“ und „System“

a) Die Funktion 'DO_WORD' inkrementiert den Wert um 1. Interessanterweise ohne eine Addition zu verwenden.

b) Die Funktion läuft, angefangen bei dem niederwertigsten Bit, die Bitfolge bis zum ersten Auftreten einer „0“ ab und setzt diese dann auf „1“. Alle bis dahin gefundenen „1“-zen werden dabei auf „0“ gesetzt (siehe *Tabelle 1).*

Vorher	Zahl	$\underbrace{0}_{1.\,null}\ \underbrace{11111}_{5-mal}$
	Wert der Zahl	$31 = 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$
Nachher	Zahl	$\underbrace{1}_{neue1}\ \underbrace{00000}_{5-mal}$
	Wert der Zahl	$32 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$

Tabelle 1: Beispiel

Zu Aufgabe 3: Weitere Funktion

a) Die Funktion 'DO_8' **addiert** binär zwei 8 Bit Zahlen.

b) Dabei wird nacheinander Stellenweise addiert. Die beiden Bit der Zahlen und das Carry werden in einen Bitvektor zusammengefaßt (`A(i) & B(i) & C` ist eine Konkatenation) und addiert. Da Daraus wird das Ergebnis der Stelle und das neue Carry berechnet.

Zu Aufgabe 4: Entity „Puzzel“

a) Hier sind zwei Fallen enthalten: Im Prozess A wird der Wert von 'PipeB' durch einen neuen Treiber überschrieben. Zum anderen wird 'PipeC' nicht wie erwartet durch Prozess C nach außen gespiegelt, da der Prozess nur 'PipeB' in der Sensitivliste enthält. Die Änderungen an 'PipeC' erfolgen im Prozess C eine Deltazeit später.

b) Zur Analyse sollte ein Prozessgraph aufgemalt werden. Dabei sollte vermerkt werden, welcher Prozess welches Signal erzeugt und welcher nur konsumiert. Dabei stellt sich heraus, das Prozess A völlig autonom ist (keine Signalabhängigkeiten). Somit kann dort die Analyse beginnen.
Prozess A hat eine Periode von 70 ns, da er nur von den 'wait'-Anweisungen und nicht von weiteren Signalen abhängt.

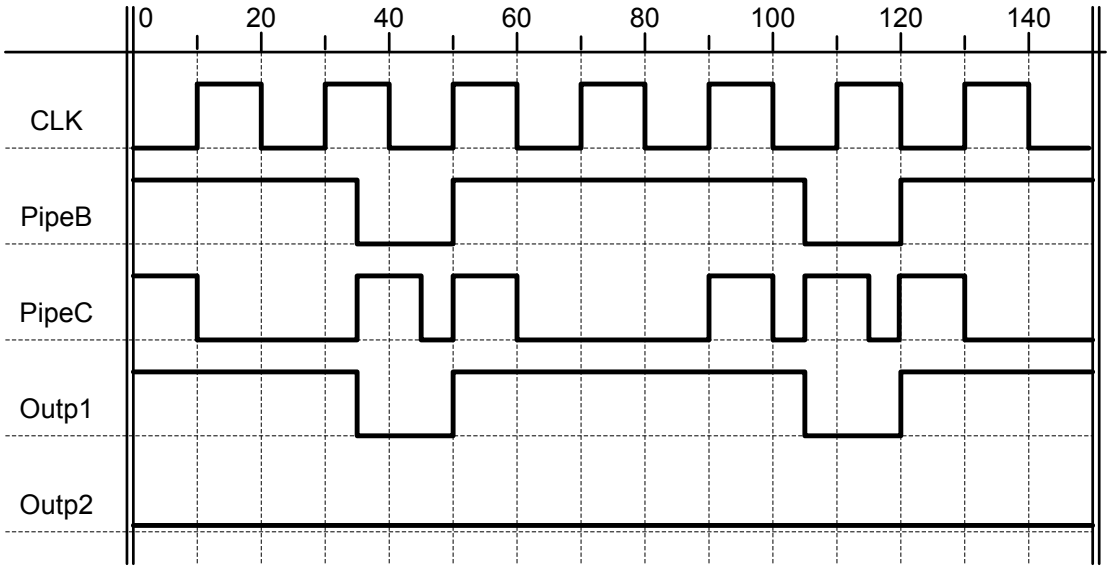

Abbildung 2: Simulationsausgabe