

Übungen “Telematik” Wintersemester 2002/2003 Blatt 2

Aufgabe 1:

Es soll ein einfacher Parity-Generator entworfen werden:

- a) Male auf Papier einen endlichen Automaten (Moore Machine) auf, der anhand des von ihm angenommenen Zustands die Parity für unendliche Bitfolgen angibt. Der Automat soll keine Endzustände oder Ausgaben haben, vielmehr soll gerade bzw. ungerade Parität für die bisherige Bitfolge aus dem Zustand, in dem sich der Automat jeweils befindet, ersichtlich sein.
- b) Beschreibe den Automaten aus a) als SDL-Prozess *ParityGenerator* und biete ihn in ein vollständiges SDL-System mit den Eingabesignalen *Null* und *Eins* ein.
- c) Erweitere das SDL-System, aus b) derart, daß nach 8 Bit der Prozess das jeweils passende Signal *Even* oder *Odd* ausgibt und sich selbst (per *Stop*-Symbol) beendet.
- d) Überprüfe, ob die MSC-Szenarien, die sich im Verzeichnis zu dieser Übung befinden, in dem erstelltem SDL-System erfolgreich (bzw. erfolglos für die **_error* MSCs) validiert werden können. (Dies ist natürlich nur möglich, wenn im SDL-Modell auch die unter b) und c) verlangten Namen verwendet wurden!)

Aufgabe 2:

Erweitere das SDL-Modell des Cola-Automaten aus dem Kurz-Tutorial¹ um einen Timer, der dafür sorgt, daß der eingeworfene Geldbetrag verfällt, falls 100 Zeiteinheiten lang keine Eingaben in der *Getraenkeausgabe* erfolgen.

Hinweis zu Timern im Tau/SDT Simulator und Validator: Bei Tau/SDT läuft nicht unaufhaltsam eine Uhr im Hintergrund. Es ist vielmehr so, daß Simulator und Validator die Zeit nur in's Spiel bringen, wenn auch tatsächlich Timer gestartet wurden. In diesem Fall muß dann nicht tatsächlich z.B. 100 Sekunden gewartet werden, sondern ein Timeout steht sofort zur Verfügung und kann konsumiert werden. (Falls im Simulator der textuelle Trace-Level ausführlich genug gesetzt wurde, wird im Trace der jeweils aktuelle Wert von NOW angezeigt.)

¹Wer die SDL-Dateien hierfür nicht mehr besitzt, findet diese ebenfalls im Verzeichnis zu dieser Übung.