1. Übung, WS 2012/13
Zuverlässige Systeme/ Dependable Systems
Ausgegeben am 25.10.2011
Vorstellung 8.11.2012

Katinka Wolter
Inst. für Informatik
Freie Universität Berlin

Assignment 1. Failures of the drinks distributor

Imagine several failures of the drinks distributor. What is

1. A static failure?
2. A dynamic failure?
3. A temporary failure?
4. A static and persistent failure?

Assignment 2. Faults of the drinks distributor

Look at the global behavioural graph of the drinks distributor.

- Imagine several types of funktional and hardware faults and show how they transform the graph. What failures do they lead to?
- Which type of faults (and on which part of the graph) affect the user's satisfaction by altering the functioning of the money management (accepting and giving back coins)?
- Find a functional transformation which allows this distributor to serve several rinks with the same initial amount of money or to give the change back by pressing on *Cancel* (you can add a *Return-Money* button).

Assignment 3. Software faults

We consider a program which declares two global variables, A and B, of Integer type and the two following functions:

```
function F1 return integer is
begin
   A := A + 1;
   return A;
end F1;
function F2 return integer is
begin
   A := 2*A;
   return A;
end F2;
```

Determine the value of B after the execution of the following statements assuming the initial value $A = 1$.

```
B := F1 + F2;
```

Assignment 4. Latency of an asynchronous counter

Let us consider a 16-bit asynchronous counter which counts pulses coming on the input I and gives the result on the output O with a natural 4-bit binary code (a, b, c, d). A stuck-at '0' of the variable a occurs while the counter is in the intitial state (0, 0, 0, 0). Hence, the counter will now evolve with constrained configurations $(0, b, c, d)$.

Determine the average latency of this fault for a 2 ms average-time between input pulses.

Assignment 4. Fault-Error-Failure in a Program

The following procedure extract aims at counting the number of sheets in a book. At first, it computes the number of the last right page.

```
procedure Count_Number_of_Sheets (Sheets_Number: out positive) is

Last_Right_Page: positive;

begin
   Last_Right_Page := <determine_number>;
   Sheets_Number := Ceil(Last_Right_Page / 2);
end Count_Number_of_Sheets;
```

We assume that the expression which calculates the `Last_Right_Page` (not further specified in the program) contains a fault. We use this procedure in a book editing processor to analyse a book whose actual last right page number is 325.

1. Is there a failure if the faulty expression gives 326 as a result? Answer the same question with 327 and 328.
2. Define the failure(s), if any.
3. Is there an error in the three cases?
4. Based on the outcome of the experiment can you identify all errors?