

Test und Verlässlichkeit, Übungsblatt 1 (20P)

Prof. G. Kemnitz, TU Clausthal, Institut für Informatik

26. März 2015

Aufgabe 1.1

Zählen Sie (je eine) Ihnen bekannte Maßnahmen zur Sicherung der Verlässlichkeit auf

- die während des Entwurf oder der Fertigung
- nach Abschluss des Entwurfs oder der Fertigung
- während des Betriebs

eines informationsverarbeitenden Systems zum Einsatz kommt.

3P

Aufgabe 1.2

Was sind Fehler und was sind Fehlfunktionen? Welcher kausaler Zusammenhang besteht zwischen beiden?

3P

Aufgabe 1.3

Eine Service-Leistung sei definiert durch

- das Eingabeformat: a, b: 16-Bit Zweierkomplement
- das Ausgabeformat: Rückgabewert 16-Bit Zweierkomplement
- die Sollfunktion: Rückgabe des Wertes des Ausdrucks $a-b+25$

Die Implementierung als C-Funktion ist nicht ganz korrekt:

```
int16_t fkt(int16_t a, int16_t b){  
    }    return a-b+25;  
}
```

- Wie groß sind die kleinsten und größten darstellbaren Ein- und Ausgabewerte? 2P
- Für welche Bedeutungen der Eingabe unterscheidet sich die Ist-Ausgabe von der Soll-Ausgabe¹? 2P
- Ändern Sie das vorgegebene Programm so, dass bei einem Bereichsüber- bzw. -unterlauf des Ergebnisses der größte bzw. kleinste darstellbare Wert zurückgegeben wird? 2P

¹Beischreibung des Bedutungsbereichs durch einen logischen Ausdruck unter Nutzung von arithmetischen und Vergleichsoperatoren.

Aufgabe 1.4

Eine Service-Leistung sei definiert durch

- das Eingabeformat: uint16_t (16 Bit, vorzeichenfrei)
- das Ausgabeformat: uint8_t (8 Bit, vorzeichenfrei)
- die Sollfunktion: Rückgabe der ganzzahligen Anteils der Wurzel

Die nachfolgende Implementierung als C-Funktion ist fehlerhaft:

```
uint8_t wurzel(uint16_t x){
    uint8_t w=0;
    uint16_t sum=0;
    while (sum<x){sum += (w<<1)+1;
    w++;}
    return w;
}
```

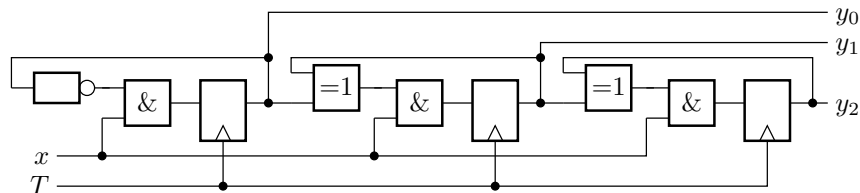
- Für welchen Eingabewertebereich liefert das Programm ein (korrektes) Ergebnis?² 2P
- Ändern Sie das Programm so, dass es für den gesamten Eingabewertebereich ein sinnvolles Ergebnis liefert. 2P

Aufgabe 1.5

Eine Schaltung soll bei $x = 0$ in den Zustand $\mathbf{y} = y_2y_1y_0 = 000$ übergehen und sonst bei jeder aktiven Taktflank seinen Wert um eins erhöhen (Binärzähler).

Takt	x	\mathbf{y}_{soll} (Folgeschritt)	\mathbf{y}_{ist} (Folgeschritt)
1	0	000	
2	1	001	
3	1	010	
4	1	011	
5	1	100	
6	1	101	
7	1	110	
8	1	111	
9	1	000	

Das nachfolgende Bild zeigt eine fehlerhafte Implementierung:



- Bestimmen Sie die tatsächlichen Ausgabesignalverläufe \mathbf{y}_{ist} (rechte Tabellenspalte ausfüllen). 2P
- Korrigieren Sie die Schaltungsrealisierung so, dass sie das Testbeispiel richtig abarbeitet. 2P

²Notfalls in einer Schleife für alle Eingabewerte ausprobieren. Das Programm stürzt bei unzulässigen Eingaben ab.