# Software Assignment 03

Assignment 3-01

a) Following the coded processors technique introduced in the lecture (Lecture 2.3 "Dependable Development Processes", Slides 36 ff.) select suitable values for A, Bx, By, Bz and D so you can be confident that all but 1/5000 of all bit errors are detectable, the current sequence cycle is 6. You shall additionally to the simple addition (z = x + y with x = 5 and y = 7) also calculate the value of z with the introduced alternate approach and perform the described check to confirm the correctness of the result.  
(4 Points)  
b) What do you have to keep in mind regarding the usage of data types for the alternate approach in contrast to the simple calculation and why?  
(1 Point)

Solution:

### b)

We have to watch out for the data types because the alternate approach involves encoded values which can be significantly larger than the original values. Using inappropriate data types that cannot handle these larger values might lead to overflow errors or incorrect calculations, thereby compromising the error detection mechanism. It is essential to use data types that can accommodate the maximum possible values during the coding process to ensure reliability and correctness of the results.

a)

variables:

𝑥=5  
𝑦=7  
𝐴=3  
𝐵𝑥=2  
𝐵𝑦=4  
𝐵𝑧=6  
𝐷=7

calculations: First, we compute the simple addition:

𝑧=𝑥+𝑦 =5+7=12

Now, we apply the coded processor technique:

𝑧𝐴=𝐴⋅𝑧=3⋅12=36

Calculate the coded values of 𝑥and 𝑦:

𝑥𝐴=𝐴⋅𝑥=3⋅5=15

𝑦𝐴=𝐴⋅𝑦=3⋅7=21

Sum these coded values:

𝑧𝐴, alt=𝑥𝐴+𝑦𝐴=15+21=36

Next, encode 𝑧 using 𝐵𝑧*:*

𝑧𝐵𝑧=𝐵𝑧⋅𝑧=6⋅12=72

To perform the check, we need the following encodings of 𝑥*x* and 𝑦*y*:

𝑥𝐵𝑥=𝐵𝑥⋅𝑥=2⋅5=10

𝑦𝐵𝑦=𝐵𝑦⋅𝑦=4⋅7=28

Sum these encoded values:

𝑧𝐵𝑧, alt=𝑥𝐵𝑥+𝑦𝐵𝑦=10+28=38

Now, perform the check:

𝑧𝐵𝑧mod 𝐷=? 𝑧𝐵𝑧, alt mod D

Substitute the values:

72 mod 7=?38 mod 7

2=?3

The check fails, indicating an error. Since we are detecting bit errors with high confidence (all but 1/5000), this mismatch would suggest such an error if it were to occur.

**Decoding:** If the check passes, the original result 𝑧can be decoded from the encoded value 𝑧𝐴​ or 𝑧𝐵𝑧

### b)

Assignment 2 -03

### Updated arith.adb

with Ada.Text\_IO; use Ada.Text\_IO;

package body Arith is

function Sqrt (Arg : Integer) return Integer is

-- Start with an initial guess

Guess : Integer := Arg / 2;

Previous : Integer := 0;

begin

Put\_Line("Calculating Sqrt of " & Arg'Image);

-- Special case for Arg = 1

if Arg = 1 then

Put\_Line("The result is 1");

return 1;

end if;

-- Iteratively improve the guess

while Guess /= Previous loop

Previous := Guess;

Guess := (Guess + Arg / Guess) / 2;

end loop;

Put\_Line("The result is " & Guess'Image);

-- Return the integer part of the square root

return Guess;

end Sqrt;

end Arith;

### **arith.ads**

package Arith is

-- Sqrt calculates the square root of the provided integer argument

-- Preconditions:

-- - Argument must be positive

-- Postconditions:

-- - Result is non-negative

-- - The square of the result is less than or equal to the argument

-- - The square of the result + 1 is greater than the argument

function Sqrt (Arg : Integer) return Integer

with Pre => Arg > 0,

Post => (Sqrt'Result >= 0) and then

(Sqrt'Result \* Sqrt'Result <= Arg) and then

((Sqrt'Result + 1) \* (Sqrt'Result + 1) > Arg);

end Arith;

**test\_sqrt.adb**

-- test\_sqrt.adb

with Ada.Text\_IO; use Ada.Text\_IO;

with Arith;

procedure Test\_Sqrt is

Arg : Integer := 25; -- different input values

Result : Integer;

begin

Result := Arith.Sqrt(Arg);

Put\_Line("The square root of " & Integer'Image(Arg) & " is " & Integer'Image(Result));

end Test\_Sqrt.

**Question 1 AGAIN**

variables:

𝑥=5

𝑦=7

𝐴=11

𝐵𝑥=3

𝐵𝑦=4

𝐵𝑧=5

𝐷=30

Now, performing the calculations:

**1.Calculating 𝑧 using the simple addition method**:

𝑧=𝑥+𝑦=5+7=12

**2.Calculating 𝑧 using the coded processors technique:**

𝑧alt=𝐴×𝑥+𝐵𝑥×𝑦+𝐵𝑦×𝑧+𝐵𝑧×(𝑥+𝑦) mod 𝐷

Substituting the values

𝑧alt=11×5+3×7+4×12+5× (5+7) mod 30

𝑧alt=55+21+48+60mod 30

𝑧alt=184mod 30

𝑧alt=4

**3.Performing the check to confirm the correctness of the result:**

𝐴×𝑧+𝐵𝑧×(𝑥+𝑦) mod 𝐷

Substitute the values:

11×12+5× (5+7) mod 30

11×12+5×12mod 30

132+60mod 30

192mod 30

192÷30=6 with a remainder of 12

192 mod 30=12

As 𝑧=𝑧alt, the result is correct.