Assignment 3, Embedded Systems II
Mälardalen University

Emil Broberg
School of Innovation, Design and, Engineering
Mälardalens University, Västerås, Sweden
Email: ebg20007@student.mdu.se

I. Question 1

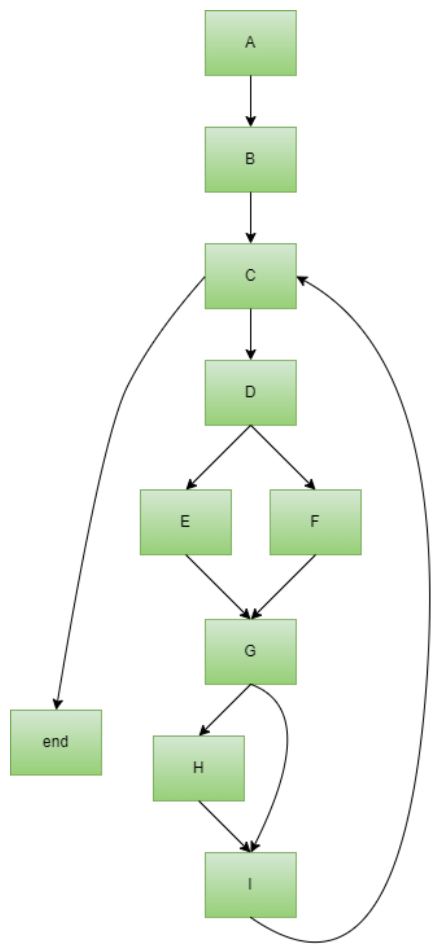

Figure 1. This is the control-flow graph for the code.

II. Question 2

The loop will continue as long as $i < 100$, the intitial value of $i = 0$ and at the end of each iteration i will be increased by 2. So shortly calculated the loop will be run for $\frac{100}{2} = 50$ times. However, this is not really correct. At the end of the iteration where i will go from 98 to 100 the loop will still run one more time since the *while statement* is at the top of the loop and not at the bottom. The statement will then be tested again and count as one more iteration. So, conclusively, the loop will run for 51 times.

III. Question 3

Yes, there is an infeasible path. The infeasible path is $A \rightarrow B \rightarrow C \rightarrow D \rightarrow F \rightarrow G \rightarrow H \rightarrow I$. This path is infeasible because if the statement $a[i] > 0$ on line D is false, the statement $a[i] > 2$ can not be true. Therefore line F and line H can not be executed in the same iteration.

IV. Question 4

The given values are the following: $t_A = t_B = 5, t_C = 7, t_D = 15, t_E = 8, t_F = 10, t_G = 15, t_H = 14, t_I = 5$. Below is a tree-based calulation of WCET bound:

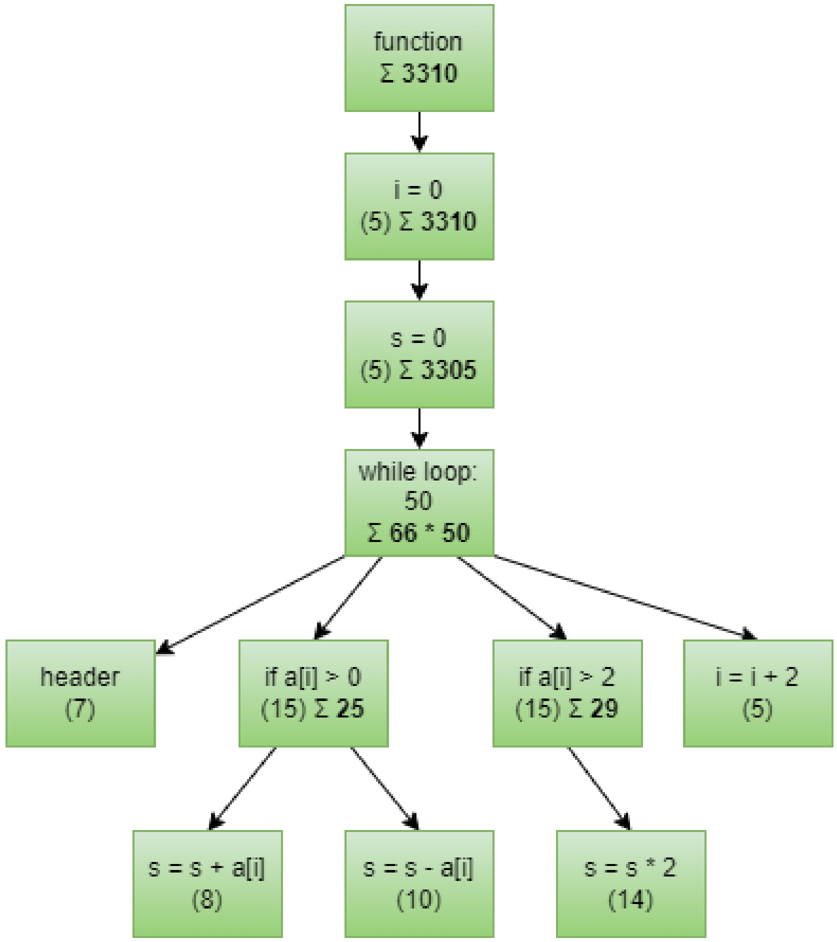

Figure 2. This is the tree-based calculation for WCET bound.

In figure 2 we can see that the WCET bound found using the tree-based calculation is 3310. The sum for the loop was found by first finding the max time for each decision statement. This is done by finding the children of the *if statement* with the longest execution time and sum up that time with the time of the *if statement* itself. After this has been done for all decision statements the sum of the loop is found by summing up all children of the loop and multiplying it by the tight upper bound for the loop. After this is done the execution times for A and B is added to the sum to find the WCET bound. Note, the tight upper bound for the loop is set to 50 and not 51 since the last iteration of will not enter the loop.

V. Question 5

Now we are going to use the path-based method to find the WCET bound. We will use the same execution times for each line of code as before. To do this all feasible paths in the loop must be found and then calculate which one will be the longest. After the longest path has been found it will be multiplied by the tight upper bound for the loop, then we will add the execution times of A and B to find the WCET bound. The following figure shows the flow graph with the execution times for each line of code.

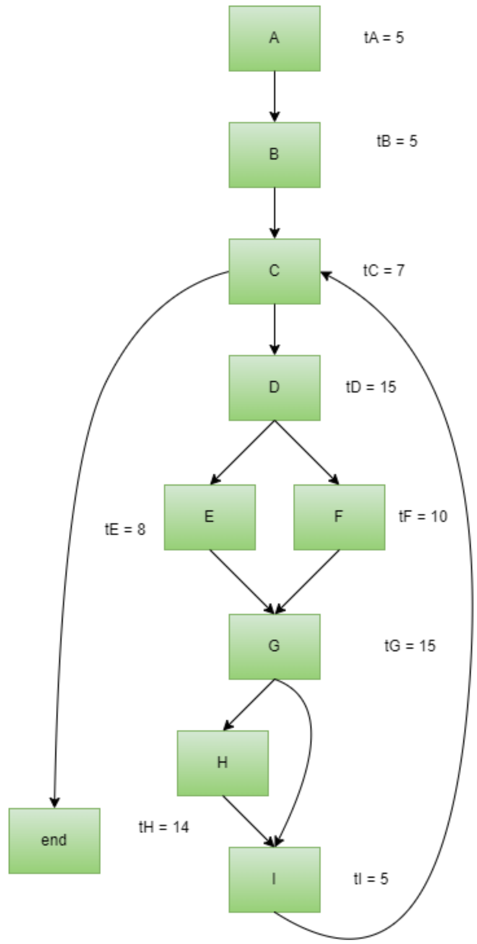

Figure 3. This is the flow-graph with the execution times of each segment used to calclate the WCET bound.

The following paths of the loop are found in the flow graph:

1) $C \to D \to F \to G \to H \to I$
2) $C \to D \to F \to G \to I$
3) $C \to D \to E \to G \to H \to I$
4) $C \to D \to E \to G \to I$

Since the first path is infeasible, no further calculations will be done on it. Calculations of the length of the other paths is as follows:

1) N/A
2) $7 + 15 + 10 + 15 + 5 = 52$
3) $7 + 15 + 8 + 15 + 14 + 5 = 64$
4) $7 + 15 + 8 + 15 + 5 = 50$

The longest path is path 3 with a length of 64. The tight upper bound for the loop is 50. Therefore the WCET bound is $64 * 50 + t_A + t_B = 3210$. This result is smaller than the tree-based calculation by a margin of 100 cycles.