technische universität dortmund   fakultät für informatik   cs 12 computer science 12

lea.schoenberger [☺] tu-dortmund.de
benjamin.glaeser [☺] tu-dortmund.de
niklas.ueter [☺] tu-dortmund.de
mikail.yayla [☺] tu-dortmund.de

Exercise for the lecture
Embedded Systems
Wintersemester 17/18

Exercise Sheet 11/12

(20 Points)

Lab exercises for the period from Wednesday, 24th January 2018

For this exercise sheet, we use the virtual machine **RTEMS-de**.

Since our license for one of the required tools is restricted to certain computers, you have to establish a SSH connection from this VM to the respective computer by executing the command **ssh youraccount@ls12pc5.cs.tu-dortmund.de** in a terminal window using your account and the related password.

Preparation

Whenever you start a new session, you have to re-register some components. For this reason, enter the following command into a terminal window:

- **cd wcet**
- **./env.sh**
- **m+ tricore-gcc**
- **m+ ait**

The files required for this exercise are already located in your home directory on the target server, more precisely, in the directory `wcet`. Please note that since it is not possible to use the VM's file browser due to the SSH connection, the command line must be used.

11/12.1 Step 1: Simple Analysis (5 Points)

Execute the command `cd ~/wcet/step1` to change in the directory used for this assignment. Compile the example program with the command **tricore-gcc -g -T ../tc1796.lds test.c** and inspect the source code, e.g., via `kate test.c`, `gedit test.c`, `gvim test.c`, or `cat test.c`.

What does the program do?

Then, execute the command `a3tricore` to start the Tricore analyzer. Under Configuration→Files, choose the file `a.out` as executable and the file `a.ais` as "AIS file". Click on the pen symbol in the "AIS file" line to start the AIS editor. **How are the loop bounds annotated by means of this file?**

Select the entry "Analyses→Create" in the navigation box on the left hand side and click on "aiT" (Safe WCET Analysis). In the newly opened screen displaying the analysis parameters, `main` must be chosen as symbol for the start of the analysis, before it can be started with a click on the "Play" button in the toolbar (the right one showing a triangle without a dot).

What is the WCET of the program? You can display additional details by clicking on the entry "WCET contributions" in the navigation box (left hand side).

Contemplate the control flow graph (Analysis→Control-Flow Graph).

How does the control flow graph represent the program? Consider carefully, how loops are represented in the CFG.

Repeat the analysis by means of the interactive analysis (play symbol with a dot). Examine the additional information shown in the boxes and open some of these via a double click. **Which information do they contain?**

11/12.2 Step 2: Scratchpad Allocation (5 Points)

A scratchpad (SPM, scratchpad memory) is a very fast, internal memory which is used as temporary storage for calculation results, data or program parts. In contrast to a cache, a scratchpad memory is maintained by the programmer and not automatically by the hardware.

Which parts of a program should be swapped out to an SPM?

Which are the differences between a "normal" system and a real-time system with respect to the usage of SPM?

Change to the directory `step2` (`cd ~/wcet/step2`), where a second version of the program considered in the previous assignment can be found. Here, two functions are swapped out to the SPM. Compile the program analogously to the `step1` program and perform an analysis with aiT.

Which impact does the usage of SPM have on the program's execution time?

If you could choose another function to be swapped out to the SPM, which one would you choose? Why?

11/12.3 Step 3: Scratchpad Allocation and Function Outlining (5 Points)

Execute the command `cd ~/wcet/step3` to open the directory used for this assignment. In the file `test.c`, some preparations have been made to transform the inner loops of the functions `Initialize` and `Sum` into separate functions via function outlining. In this manner, it is possible to swap out the inner functions to the scratchpad memory (SPM) individually.

Perform the function outlining of the inner loops of the aforementioned both functions.

Thereon, compile the program by executing the command analogously to the `step1` program and load it into the `a3tricore` analyzer as explained previously. If you start an aiT analysis, some errors will be reported.

Correct the errors in the file `a.ais`, so that the analysis can be performed properly.

How did the WCET change compared to the previous version (complete functions in the SPM)?

11/12.4 Step 4: Integer Linear Programming (5 Points)

aiT performs the WCET analysis by analyzing each basic block of a program separately, constructing an ILP (integer linear program) based on these results and solving it.

In the directory `step4`, more precisely, in the file `example.lp`, an exemplary formulation of an ILP can be found. Open this file with a text editor of your choice and solve it with an ILP solver via `lp_solve example.lp`.

Another example is given in the file `example2.lp`, where the control flow graph of a simple program is modeled. Solve this problem via `lp_solve example2.lp`.

Explain the result.

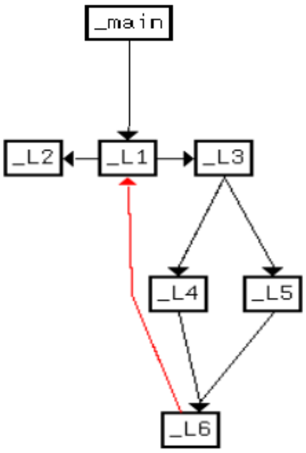
How can we model a certain basic block with execution time of more than one cycle in the ILP?

Here, the following program containing a loop is given:

```
int main()
{
  int i, j = 0;

  _Pragma( "loopbound min
                  100 max 100" );
  for ( i = 0; i < 100; i++ ) {
    if ( i < 50 )
      j += i;
    else
      j += ( i * 13 ) % 42;
  }

  return j;
}
```



Block	Cycles
main	21
L1	27
L2	20
L3	2
L4	2
L5	20
L6	13

Figure 1: Example program with loop.

The basic structure of this program is already modeled in the file `ipet.lp`.

Add the number of cycles for each basic block.

If you try to solve the ILP via `lp_solve ipet.lp`, the solver will abort with the error message *This problem is unbounded*.

Why is it impossible for the ILP solver to find a solution?

Add the missing constraint for edge h.

If you try again to solve the corrected file with lp_solve, you should obtain a result.
Why does the solver compute a solution in which L4 is never executed?

General information: An overview about the exercise sessions as well as further information can be found on `https://ls12-www.cs.tu-dortmund.de/daes/de/lehre/lehrveranstaltungen/wintersemester-20172018/es-1718.html`. The exercise sheets will usually be published on the course website on Mondays and will be solved during the respective exercise sessions. The exercises are divided into two parts, in each of which at least 50% of the points must be achieved in order to receive the exam admission.