Stack Calculator

User Guide

Version/Revision	
Date	10.08.2016
From	CDG-SMT/ESM2
Our Reference	Markus Becker
Telephone	+49(711)811-43339

BOSCH

Invented for life

StackCalculator

User Guide

Contents

1 Introduction

The StackCalculator provides a means to statically calculate the worst case stack usage of a given elf file. This tool works for programs that were compiled with the Green Hills compiler for Renesas processors. This user guide describes how the stack tool can be used.

1.1 Product version

This User Guide is relevant for StackCalculator, Version 1.3.0.

1.2 Conventions

command	A command that is entered on command line
<parameter>	Compulsory parameter
[parameter]	Optional parameter
File or folder name	The name of a file or folder is made bold and Italic

1.3 Prerequisites

There are several prerequisites that have to be met in order for the stack tool to be able to calculate a correct worst case stack prediction.

- Elf-File must be built with the Green Hills compiler for Renesas processors
- Elf-File, and all the linked libraries, must be built with debug information. So --no_debug may not be used
- Listfiles from compiling the sources must be present

2 Using stack tool

The tool is packed as a .jar-file. Therefore it must be called using the java-program. This is the command line that starts the stack tool:

```
java -jar stacktool.jar <parameters> [parameter(s)]
```

The behaviour of the stack tool is controlled through

1. Command line parameters
2. Calculation description file

2.1 Command line parameters

The stack tool accepts a number of command line parameters. Each of these parameters is prefixed with a hyphen (-). Some parameters are standalone while others shall be followed by a value.

The general syntax of command line parameters is

```
java -jar stacktool.jar -paramter1 -parameter2 <value>
```

Available command line parameters and their descriptions are as follows:

-additionalOptionsFile <file_name>

Optional. An additional file that contains command line parameters for gstack.

-calculationRules <file_name>

Mandatory. The file that describes how the worst case stack of the different entry points shall be added up.

-detailedLogfile <file_name>

Optional. A file to which additional info about the worst case stack shall be written. That includes info such as a call tree. If the parameter is not present the output is written to a file named "additionalStackInfo.txt".

-elfFile <file_name>

Mandatory. The elf-file for which the worst case stack usage shall be calculated

-ghsToolsPath <folder_name>

Mandatory. The folder that contains the binary files of the Greenhills tools.

-help / --help

Optional. Displays a help text that describes what the program does, and what parameters it reads

-lisPath <folder_name>

Mandatory. The folder that contains the list-files that were created when compiling the sources of the elf-file.

-logfile <file_name>

Optional. The file to which the output of the program shall be written. If the parameter is not present the output is written to a file named "stackCalculationResult.txt".

-schedulingFile <file_name>

Mandatory. The rbsys_dyb_mp.c-file that contains information about the structure of the elf-file.

-temporaryOptionsFile <file_name>

Optional. Use this parameter to specify the file used to store temporary options for gstack.

-D

Optional. This debug option writes the following information to the log file specified in -logfile:

- which function pointers were found in which function
- which function pointers point to which functions

2.2 Calculation description file

The calculation description file contains information about the entry points of the program and how their respective worst case stack usages shall be added up.

The calculation description file is an xml-file. Inside of the xml-file there can be several entry points defined. There are two different types of entry points that share a common tag. This tag is the <ENTRY_POINT>-tag. The different types of entry points are differentiated by their names.

If an entry point starts with the prefix "c_" then the stack tool processes it as an entry vector. This means that the entry point is not a function, but a whole array of functions. The name of the <ENTRY_POINT>-tag stands for the name of the array. The arrays are looked up in the rbsys_dyn_mp.c-file. All the functions that may be reached through an entry vector are processed, but only the one that produces the highest worst case stack prediction is assumed as the value of the entry vector. All the other entry points must be functions in the code for the .elf-file.

The exact structure that the calculation description file must follow is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://wwww.bosch.de/
    stackcalc" xmlns:tns="http://wwww.bosch.de/stackcalc" elementFormDefault="qualified"
    >

    <complexType name="STACK_CALCULATION">
        <all>
            <element
                name="OS_CONFIG"
                type="tns:OS_CONFIG"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="CORES"
                type="tns:CORES"
                maxOccurs="1"
                minOccurs="1">
            </element>
            <element
                name="DEFINES"
                type="tns:DEFINES"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SHOWONLY"
                type="tns:SHOWONLY" maxOccurs="1" minOccurs="0">
            </element>
        </all>
    </complexType>

    <complexType name="OS_CONFIG">
        <sequence>
            <element
                name="COMPONENT_NAME"
                type="tns:COMPONENT_NAME"
                maxOccurs="1"
                minOccurs="1">
            </element>
            <element
                name="COMPONENT_VERSION"
                type="tns:COMPONENT_VERSION"
                maxOccurs="1"
                minOccurs="1">
            </element>
        </sequence>
    </complexType>

    <complexType name="SUM">
        <choice
            maxOccurs="unbounded"
            minOccurs="1">
            <element
                name="ENTRY_POINT"
                type="tns:ENTRY_POINT"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SUM"
                type="tns:SUM"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="MAX"
                type="tns:MAX"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SAFETY_BUFFER"
                type="tns:SAFETY_BUFFER"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SUB"
                type="tns:SUB"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="DEFINE_REF"
                type="tns:DEFINE_REF"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="MUL"
                type="tns:MUL" maxOccurs="1" minOccurs="0">
            </element>
        </choice>
        <attribute
            name="name"
            type="string" use="optional">
        </attribute>
    </complexType>

    <complexType name="MAX">
        <choice
            maxOccurs="unbounded"
            minOccurs="1">
            <element
                name="SUM"
                type="tns:SUM"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SAFETY_BUFFER"
                type="tns:SAFETY_BUFFER"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="MAX"
                type="tns:MAX"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="ENTRY_POINT"
                type="tns:ENTRY_POINT"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="DEFINE_REF"
                type="tns:DEFINE_REF"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="MUL"
                type="tns:MUL" maxOccurs="1" minOccurs="0">
            </element>
        </choice>
        <attribute
            name="name"
            type="string" use="optional">
        </attribute>
    </complexType>

    <simpleType name="ENTRY_POINT">
        <restriction base="string">
            <pattern value="[a-zA-Z_][a-zA-Z0-9_]*"/>
        </restriction>
    </simpleType>

    <simpleType name="SAFETY_BUFFER">
        <restriction base="integer"></restriction>
    </simpleType>

    <simpleType name="COMPONENT_NAME">
        <restriction base="string"></restriction>
    </simpleType>

    <simpleType name="COMPONENT_VERSION">
        <restriction base="string">
            <pattern value="(\d+\.)*\d+"/>
        </restriction>
    </simpleType>

    <element
        name="STACK_CALCULATION"
        type="tns:STACK_CALCULATION">
    </element>

    <complexType name="CORES">
        <sequence maxOccurs="unbounded" minOccurs="1">
            <element
                name="CORE"
                type="tns:CORE">
            </element>
        </sequence>
    </complexType>

    <complexType name="CORE">
        <choice
```

```
            maxOccurs="unbounded"
            minOccurs="0">
            <element
                name="SUM"
                type="tns:SUM">
            </element>

        </choice>
        <attribute
            name="name"
            type="string" use="optional">
        </attribute>
    </complexType>


    <complexType name="SUB">
        <choice
            maxOccurs="unbounded"
            minOccurs="1">
            <element
                name="ENTRY_POINT"
                type="tns:ENTRY_POINT"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="MAX"
                type="tns:MAX"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SUM"
                type="tns:SUM"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="DEFINE_REF"
                type="tns:DEFINE_REF"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="MUL"
                type="tns:MUL" maxOccurs="1" minOccurs="0">
            </element>
        </choice>
        <attribute
            name="name"
            type="string">
        </attribute>
    </complexType>
```

```
<complexType name="DEFINES">
    <sequence>
        <element
            name="DEFINE"
            type="tns:DEFINE"
            maxOccurs="unbounded"
            minOccurs="0">
        </element>
    </sequence>
</complexType>

<complexType name="DEFINE">
    <choice
        maxOccurs="unbounded"
        minOccurs="1">
        <element
            name="SUM"
            type="tns:SUM">
        </element>
        <element
            name="SUB"
            type="tns:SUB">
        </element>
        <element
            name="MAX"
            type="tns:MAX">
        </element>
        <element
            name="ENTRY_POINT"
            type="tns:ENTRY_POINT">
        </element>
        <element
            name="MUL"
            type="tns:MUL">
        </element>
    </choice>
    <attribute
        name="name"
        type="string" use="required">
    </attribute>
</complexType>

<simpleType name="DEFINE_REF">
    <restriction base="string"></restriction>
</simpleType>

<complexType name="SHOWONLY">
    <sequence
        maxOccurs="unbounded"
        minOccurs="1">
        <element
```

```
                name="ENTRY_POINT"
                type="tns:ENTRY_POINT">
            </element>
        </sequence>
        <attribute
            name="name"
            type="string">
        </attribute>
    </complexType>

    <complexType name="MUL">
        <sequence maxOccurs="1" minOccurs="1">
            <element
                name="FACTOR"
                type="tns:FACTOR"
                maxOccurs="1"
                minOccurs="1">
            </element>
            <element
                name="FACTOR"
                type="tns:FACTOR"
                maxOccurs="1"
                minOccurs="1">
            </element>
        </sequence>
        <attribute
            name="name"
            type="string" use="optional">
        </attribute>
    </complexType>

    <complexType
        name="FACTOR"
        mixed="true">
        <choice maxOccurs="1" minOccurs="0">
            <element
                name="DEFINE_REF"
                type="tns:DEFINE_REF">
            </element>
            <element
                name="MUL"
                type="tns:MUL">
            </element>
            <element
                name="MAX"
                type="tns:MAX">
            </element>
            <element
                name="SUM"
                type="tns:SUM">
            </element>
            <element
<?xml version="1.0" encoding="UTF-8"?>
```

```
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.bosch.de/
    stackcalc" xmlns:tns="http://www.bosch.de/stackcalc" elementFormDefault="qualified"
    >

    <complexType name="STACK_CALCULATION">
        <all>
            <element
                name="OS_CONFIG"
                type="tns:OS_CONFIG"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="CORES"
                type="tns:CORES"
                maxOccurs="1"
                minOccurs="1">
            </element>
            <element
                name="DEFINES"
                type="tns:DEFINES"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SHOWONLY"
                type="tns:SHOWONLY"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="UNKNOWN_POINTERS"
                type="nonNegativeInteger"
                maxOccurs="1"
                minOccurs="0">
            </element>
        </all>
    </complexType>

    <complexType name="OS_CONFIG">
        <sequence>
            <element
                name="COMPONENT_NAME"
                type="tns:COMPONENT_NAME"
                maxOccurs="1"
                minOccurs="1">
            </element>
            <element
                name="COMPONENT_VERSION"
                type="tns:COMPONENT_VERSION"
                maxOccurs="1"
                minOccurs="1">
            </element>
        </sequence>
    </complexType>

    <complexType name="SUM">
        <choice
            maxOccurs="unbounded"
            minOccurs="1">
            <element
                name="ENTRY_POINT"
                type="tns:ENTRY_POINT"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SUM"
                type="tns:SUM"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="MAX"
                type="tns:MAX"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SAFETY_BUFFER"
                type="tns:SAFETY_BUFFER"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SUB"
                type="tns:SUB"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="DEFINE_REF"
                type="tns:DEFINE_REF"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="MUL"
                type="tns:MUL" maxOccurs="1" minOccurs="0">
            </element>
        </choice>
        <attribute
            name="name"
            type="string" use="optional">
        </attribute>
    </complexType>
```

```
<complexType name="MAX">
    <choice
        maxOccurs="unbounded"
        minOccurs="1">
        <element
            name="SUM"
            type="tns:SUM"
            maxOccurs="1"
            minOccurs="0">
        </element>
        <element
            name="SAFETY_BUFFER"
            type="tns:SAFETY_BUFFER"
            maxOccurs="1"
            minOccurs="0">
        </element>
        <element
            name="MAX"
            type="tns:MAX"
            maxOccurs="1"
            minOccurs="0">
        </element>
        <element
            name="ENTRY_POINT"
            type="tns:ENTRY_POINT"
            maxOccurs="1"
            minOccurs="0">
        </element>
        <element
            name="DEFINE_REF"
            type="tns:DEFINE_REF"
            maxOccurs="1"
            minOccurs="0">
        </element>
        <element
            name="MUL"
            type="tns:MUL" maxOccurs="1" minOccurs="0">
        </element>
    </choice>
    <attribute
        name="name"
        type="string" use="optional">
    </attribute>
</complexType>

<simpleType name="ENTRY_POINT">
    <restriction base="string">
        <pattern value="[a-zA-Z_][a-zA-Z0-9_]*"/>
    </restriction>
</simpleType>

<simpleType name="SAFETY_BUFFER">
```

```
        <restriction base="integer"></restriction>
    </simpleType>

    <simpleType name="COMPONENT_NAME">
        <restriction base="string"></restriction>
    </simpleType>

    <simpleType name="COMPONENT_VERSION">
        <restriction base="string">
            <pattern value="(\d+\.)*\d+"/>
        </restriction>
    </simpleType>

    <element
        name="STACK_CALCULATION"
        type="tns:STACK_CALCULATION">
    </element>

    <complexType name="CORES">
        <sequence maxOccurs="unbounded" minOccurs="1">
            <element
                name="CORE"
                type="tns:CORE">
            </element>
        </sequence>
    </complexType>

    <complexType name="CORE">
        <choice
            maxOccurs="unbounded"
            minOccurs="0">
            <element
                name="SUM"
                type="tns:SUM">
            </element>

        </choice>
        <attribute
            name="name"
            type="string" use="optional">
        </attribute>
    </complexType>


    <complexType name="SUB">
        <choice
            maxOccurs="unbounded"
            minOccurs="1">
            <element
                name="ENTRY_POINT"
                type="tns:ENTRY_POINT"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="MAX"
                type="tns:MAX"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="SUM"
                type="tns:SUM"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="DEFINE_REF"
                type="tns:DEFINE_REF"
                maxOccurs="1"
                minOccurs="0">
            </element>
            <element
                name="MUL"
                type="tns:MUL" maxOccurs="1" minOccurs="0">
            </element>
        </choice>
        <attribute
            name="name"
            type="string">
        </attribute>
    </complexType>


    <complexType name="DEFINES">
        <sequence>
            <element
                name="DEFINE"
                type="tns:DEFINE"
                maxOccurs="unbounded"
                minOccurs="0">
            </element>
        </sequence>
    </complexType>

    <complexType name="DEFINE">
        <choice
            maxOccurs="unbounded"
            minOccurs="1">
            <element
                name="SUM"
                type="tns:SUM">
            </element>
            <element
                name="SUB"
                type="tns:SUB">
            </element>
            <element
                name="MAX"
                type="tns:MAX">
            </element>
            <element
                name="ENTRY_POINT"
                type="tns:ENTRY_POINT">
            </element>
            <element
                name="MUL"
                type="tns:MUL">
            </element>
        </choice>
        <attribute
            name="name"
            type="string" use="required">
        </attribute>
    </complexType>

    <simpleType name="DEFINE_REF">
        <restriction base="string"></restriction>
    </simpleType>

    <complexType name="SHOWONLY">
        <sequence
            maxOccurs="unbounded"
            minOccurs="1">
            <element
                name="ENTRY_POINT"
                type="tns:ENTRY_POINT">
            </element>
        </sequence>
        <attribute
            name="name"
            type="string">
        </attribute>
    </complexType>

    <complexType name="MUL">
        <sequence maxOccurs="1" minOccurs="1">
            <element
                name="FACTOR"
                type="tns:FACTOR"
                maxOccurs="1"
                minOccurs="1">
            </element>
            <element
                name="FACTOR"
                type="tns:FACTOR"
                maxOccurs="1"
                minOccurs="1">
            </element>
        </sequence>
        <attribute
            name="name"
            type="string" use="optional">
        </attribute>
    </complexType>

    <complexType
        name="FACTOR"
        mixed="true">
        <choice maxOccurs="1" minOccurs="0">
            <element
                name="DEFINE_REF"
                type="tns:DEFINE_REF">
            </element>
            <element
                name="MUL"
                type="tns:MUL">
            </element>
            <element
                name="MAX"
                type="tns:MAX">
            </element>
            <element
                name="SUM"
                type="tns:SUM">
            </element>
            <element
                name="SUB"
                type="tns:SUB">
            </element>
        </choice>
    </complexType>
</schema>
```

2.3 Scheduling file

The scheduling file is an input for the stack calculator. The stack calculator needs this file as an input because of the format of the calculation description file. The calculation description file contains all the relevant entry points to calculate the worst case stack usage. Some of these entry points start with the prefix "c_". This prefix indicates that this entry point is not a function, but an array of function pointers. The stack calculator has to find this array and add all the elements of this array as entry points for gstack.

The stack calculator searches for these entry points inside of the scheduling file. For each entry vector (entry point with "c_"-prefix) there must exist exactly one array of the type PRC_PTR in the scheduling file.

Example:

- Content of stackCalcRules.xml

  ```
  <SUM name="wcs␣idle␣task">
  <ENTRY_POINT>c_TaskIdle</ENTRY_POINT>
  </SUM>
  ```

- Content of Scheduling File

```
const PRC_PTR c_TaskIdle[] =
{
 PRC_CUBAS_MemStack_Cyclic,
 PRC_CUBAS_Mcal_Cyclic,
 (PRC_PTR)0
};
```

Result: Each entry in c_TaskIdle will be added as entry point.

2.4 Additional Options file

This is an additional input file that can be used to give gstack additional information that cannot be extracted automatically. Technically this file may include any parameters that gstack processes. Here are some suggestions for which parameters may be of interest:

-a caller:callee1[,callee2,...]

Adds connections to the call graph.
Example: -a caller:callee1,callee2 adds calls from caller to callee1 and callee2 to the call graph.

-d callee[:caller,...]

If you specify one function without caller, this option deletes all calls to that function. If you also specify a list of callers, this option deletes all calls from those callers to the specified callee. Instead of a function name, you can also use «address_taken».
Examples:

- *-d callee* deletes all calls to callee.
- *-d callee:caller1,caller2* deletes all calls from caller1 and caller2 to callee.
- *-d «address_taken»:callee* deletes all unknown, indirect calls from callee

-max_instances func=n

Specifies the maximum number of instances n of the function func that can exist on the stack at any given time. This option helps gstack compute the stack use of recursive clusters.

-x func=n

Specifies an additional amount n to add to the known maximum stack value for function func. This option is the equivalent of #pragma ghs extra_stack. A full list of command line parameters can be found in the documentation of gstack[1].

2.5 Output files

The stack tool produces two output files. One output file specified by the command line option -logfile gives an overview about the calculation process. It also shows the worst case stack usage for each core that was defined in the xml-file.

The other output file specified by the command line option -detailedLogfile is produced by gstack. It contains the worst case stack information about each function of the program, as well as a call tree and information about missing information (e.g. unannotated assembler functions).

3 Calculation process

The tool works in three steps: first it calls gstack, then it reads the gstack output and finds all the pointer information that it can find, at the end it calls gstack again, but provides the pointer information.

The main procedure in the tool finds the global, constant function pointers. It checks where these pointers point to. Then it checks for the functions that use the global, constant function pointers and adds a link from the function that uses a function pointer to the target (the value) of the function pointer.

If the program detects that a function uses a function pointer, but could not locate to which functions it points it assumes that this function pointer has no targets (i.e. nothing is added to the worst-case stack).

4 Deprecated features

This section lists deprecated features in the StackCalculator.

- Since tool version 1.3.0 the StackCalculator assumes that no target exists if either the StackCalculator or gstack could not find a target for a function pointer. In prior releases, a user-defined value could be specified for that scenario. This feature is deprecated.

References

[1] Green Hills Inc, *MULTI: Building Applications for Embedded V850 and RH850*, Version 2015.1.7