

Application Note: AN10114

How to specify loop iterations in nested loops

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to specify loop iterations in nested loops.

Required tools and libraries

This application note is based on the following components:

- xTIMEcomposer Tools - Version 14.0.0

Required hardware

Programming how-tos are generally not specific to any particular hardware and can usually run on all XMOS devices. See the contents of the note for full details.

1 How to specify loop iterations in nested loops

In many cases the XTA will know the number of iterations for a given loop. However, in some cases, this information cannot be deduced by the tools so you must supply it.

This example describes some options for setting loop iterations for nested loops, and introduces the concept of the loop scope.

For example, compile the following code:

```
int f() {
    int i, j, k = 0;
    for (i = 0; i < 5; ++i) {
        #pragma xta label "outer_loop"
        for (j = 0; j < i; ++j) {
            #pragma xta label "inner_loop"
            k += i + j;
        }
    }
    return k;
}

int main() {
    f();
    return 0;
}
```

By default, the tool assumes that the iterations for loops are *relative*, meaning that the iterations for an inner loop will be multiplied by the iterations of enclosing loops. However this is not sufficient to describe all loop structures. If this assumption is not correct (as is the case in the above code) a loop count can be set to *absolute*. The iteration count set on an absolute loop is not multiplied up by the iterations set on enclosing loops. For example, in this case, if you set the inner loop to have an absolute scope, and a loop count of 10, then this will correctly capture the behavior of the code.

2 Setting a global loop scope

Load the resulting executable into the XTA then click the 'Add define' button in the toolbar. In the dialog, switch to the *Loop Scope* tab, enter 'inner_loop' in the reference section and check the *make scope absolute* box. Now, any iterations set on 'inner_loop' will be treated as absolute.

To add a global loop scope using the command line XTA, or from an XTA script/embedded source command, the following can be used:

```
add loopscope inner_loop a
```

3 Setting a local loop scope

Load the resulting executable into the XTA then time the function 'f'. This will create the route. Right-click in the left hand side border of the editor on the source line inside the loop ($k += j + i$) and select 'Set loop scope'. In the resulting dialog check the *make scope absolute* box.

To add a local loop scope using the command line XTA, or from an XTA script/embedded source command, the following can be used:

```
set loopscope 0 inner_loop a
```

This will set a the inner loop for the route with an id of 0 to have a scope of *absolute*.