Application Note: AN10108

How to control the XTA from a python script

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to control the XTA from a python script.

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 control the XTA from a python script

There are occasions when the inbuilt XTA command scripting mechanism does not offer the required level of power and/or flexibility. For example, you may wish to pass arguments to a script, use the return value of one command as input to another or make use of conditional/looping constructs. With this in mind, the XTA provides an interface which allows the tool to be controlled from Python scripts.

For example, compile the following code:

```
#include <stdlib.h>
#include <xs1.h>

port p1 = XS1_PORT_1A;
port p2 = XS1_PORT_1B;

int main() {
  int x;

  #pragma xta endpoint "input"
  p1 :> x;

  // Checks for errors..
  if (x == 1) {
    #pragma xta label "error_case"
    exit(1);
  }

  // do some computation here..

  #pragma xta endpoint "output"
  p2 <: 0;
  return 0;
}
```

As an example, assume that you want to time from the input to the output, set a requirement and exclusion, and this this must be done from a Python script. To achieve this, place the following lines in a file, e.g. script.py:

```
import java

try:
  xta.load("a.xe") ;

except java.lang.Exception, e:
  print e.getMessage()

try:
  xta.addExclusion("error_case")
  ids = xta.analyzeEndpoints("input", "output")
  for id in ids:
    print xta.getRouteDescription(id) + ":",
    print xta.getWorstCase(id, "ns"),

except java.lang.Exception, e:
  print e.getMessage()
```

The above script can then be run using the XTA *source* command:

```
xta source script.py -exit
```

Note: See the XMOS-Timing-Analyzer-Manual for further information on available interface methods.