How to use overlays to reduce application memory requirements

version	1.1.1
scope	Example. This code is provided as example code for a user to base their code on.
description	How to use overlays to reduce application memory requirements
boards	Unless otherwise specified, this example runs on the SliceKIT Core Board, but can easily be run on any XMOS device by using a different XN file.

An overlay is a block of code and data that is loaded on demand at runtime. Each overlay has a predetermined region of memory that it is copied to called an overlay region. Several overlays may be associated with the same overlay region, but only one of these overlay can be loaded at any one time.

Overlays reduce the amount of memory needed to run your application since it is no longer necessary to reserve space for all your code and data - instead the tools only need to reserve space for the largest overlay that can be loaded into each overlay region.

Create an overlay by marking a function definition as an overlay root:

```
[[overlay]]
void print_hello()
{
  printstr("Hello");
}
```

This creates an overlay containing the function `print_hello`. The tools will automatically place code and read only data that is only referenced from `print_hello` in this overlay. The overlay is loaded on demand when `print_hello` is called. Calls to `print_hello` may be slow (since the overlay must be loaded), but once inside the overlay code will execute at full speed.

Create a second overlay by marking a second function as an overlay root:

```
[[overlay]]
void print_world()
{
  printstrln(" World!");
}
```

Call the functions in sequence:

XMOS®

```
int main()
{
  print_hello();
  print_world();
  return 0;
}
```

The two overlays will be automatically placed in the same overlay region since they are never in use at the same time. At the start of `main` no overlay is loaded. When `print_hello` is called the overlay runtime will load the overlay containing `print_hello` into the overlay region. When `print_world` is called the overlay runtime will load the overlay containing `print_world` into the same overlay region, replacing `print_hello` in memory.

Finally specify the option `-foverlay=syscall` you build the application. This links against an overlay runtime that loads overlays from a host machine using a system call. Applications that use this overlay runtime must either be run via JTAG using a debug adapter or be run in XSIM.

XMOS®

REV A