

# The Memory Map in XML Format

Sameer D. Sahasrabuddhe

November 2006

## 1 Introduction

The AHIR tools use an XML file to describe the contents of memory as seen by the generated AHIR specification. A template map is first created by `irlink` using the locations assigned to various functions during the linking process. The map provides a descriptive label for each location, along with the type and numerical address. The user is expected to fill-in relevant values, using the label.

```
<scalar addr="8" descr="start_formal_0" type="int"></scalar>
```

This instance of the tag “`<scalar>`” declares a scalar variable described as the first formal argument of the function `start`, of type `int`. The address assigned to this variable is 8. The user may initialise this variable by inserting the initial value (say, 10) as contents of the tag:

```
<scalar addr="8" descr="start_formal_0" type="int">10</scalar>
```

**NOTE:** The XML file for the memory map that `irlink` generates, is given a name derived from the name of the module. For a module `foo`, for example, the memory map is called `foo_map.xml`. This file is created only when `irgen` runs for the first time. If a file of this name already exists, `irgen` will instead create a file with the name `foo_new_map.xml`. This file is over-written on subsequent runs, with no warnings.

## 2 Structure of the Memory Map

The memory map is an XML document, which must start with the tag “`<map>`”, that has two compulsory attributes - “`id`” and “`size`”. The memory map is divided into two regions - the initial values and the final values present in memory.

```
<map id="func" size="42">
<init>
<!-- Initial contents of memory -->

</init>

<!-- first free address: 42
addresses from this value onwards can be used for external data. -->
```

```

<fin>
<!-- Final contents of memory -->
</fin>
</map>

```

## 2.1 Populating the memory map

The memory map has to be populated only when simulating the generated AHIR description of a complete system. For example, `irsim` uses the values to create an `std::map<unsigned, bit_vector>` look-up table which is used to model the external memory. In `irsyn`, the generated test-bench contains an array of values, that are written to the system memory through suitable access ports before the test bench begins execution.

In case of synthesisable VHDL code, the test bench is entirely ignored, so that the values serve no purpose either. Only the size of the memory map is important in order to generate a suitable memory component.

### The size attribute

The initial memory map is generated by `irlink`, and contains all the locations that are declared in the input C program. These include locations for arguments and return value of each function, as well as statically declared variables within functions. These locations are assigned addresses linearly, starting from 0.

The generated XML file also contains a comment that mentions the first free location in the map. The user is free to use addresses from there onwards for additional data. When such additional data is introduced, the `size` attribute of the map must be set to a number that accomodates all the locations declared. This attribute is used as a hint by further tools, when creating a memory component in the system.

### The init and fin sections

The `init` and `fin` sections of the memory map serve two purposes:

1. as data entry points for a user during simulation
2. as information for driver modules, that need to know the location of postboxes when sending data to the system during actual runs