Workload Description File

This workload description file describes the structure of XML files that describe both the input stimuli and output vectors for the GroundHog 2009 benchmark suite. Note that an implemented design+workload represent a benchmark in the GroundHog 2009 benchmark suite. Below we will describe some of the details of the workload description file.

Each individual design specification within the GroundHog benchmark specifies the unique resources that are used in their workloads.

File Details

The workload description file can be outputted in Extensible Markup Language (XML). The tags for the file are described below.

XML Tags

Workload

The root of the workload description file is:
<workload> … </workload>
The body of workload can contain Conditions and Events tags.

Conditions

Conditions (a direct child of the root) contains a series of conditions that define when something happens what the environment response will be.
<conditions> … </conditions>
The conditions tags will wrap condition tags

Events

Events (a direct child of the root) contains a series of events that define when input stimuli will be passed to the system under test (SUT).
<events> … </events>
The event tags will wrap event tags

Condition

A condition tag describes a specific condition on the environment that provides the stimuli. This is a condition on an output event and is used to allow the environment to respond to output actions of the SUT.
<condition> … </condition>
Within the condition tag there is an if tag, an associated resource tag, and a time tage. The if tag describes a condition based on an output event happening. The resource describes the action to take when the condition is true, and this action corresponds to an event that will be added to the workload stimuli. The time tag in this case describes based on current time, when in the future the action will be taken. This time must be > 0.

Event

An event tag describes a specific event. This can be an event on resource.
<event> … </event>
Within the event tag there is a time, assertion (optional), and resource tag where time describes when it happens (assuming we start at time = 0). The assertion tag describes what should happen at the output because of the event (normally a time constraint). The resource describes what stimuli is made to a resource.

If

The if tag describes a condition to evaluate.
<if> … </if>
Within the if tag a resource tag is specified. If the resource specified is in a current state that is true then the condition is true.

Resource

The resource tag describes a resource based on a name and can assign a resource some value.
<resource> … </resource>
Resources wrap the name and value tags

Name

The name tag contains a unique string name
<name>{string}</name>

Value

The value tag contains a value string, where the [0-9*] denotes a number for a value tag. The numbers progress from 0 to X, where X < 2^32.
<value[0-9*]>{string}</value[0-9*]>

Time

The time tag describes time of something happening from the indicated starting time.
<time>{integer}{m, s, ms, us, ps}</time>
m = minute
s = second
ms = millisecond
us = microsecond
ns = nanosecond

Workload number

This holds an integer representing the number of this workload. This is mainly used within the tool set to associate workloads with parameters for event waveforms.
<workload_number>{integer}</workload_number>

Design name

This holds a character string of the associated design.
<design_name> … </design_name>

Additional details

This holds additional details that may help the parsing of a workload
<additional_details> … </ additional_details >

Minimum arrival rate

Describes the minimum arrival rate that the solution is expected to handle (see the groundhog_09_meta_document.pdf for more details).
< minimum_arrival_rate>{integer}{m, s, ms, us, ps}</ minimum_arrival_rate>
m = minute
s = second
ms = millisecond
us = microsecond
ns = nanosecond

Minimum operating speed

Describes the heart beat of the system (see the groundhog_09_meta_document.pdf for more details).
<minimum_operating_speed>{integer}{m, s, ms, us, ps}</minimum_operating_speed>
m = minute
s = second
ms = millisecond
us = microsecond
ns = nanosecond

Minimum sampling speed

Describes the heart beat of the system (see the groundhog_09_meta_document.pdf for more details).
<minimum_sampling_speed>{integer}{m, s, ms, us, ps}</minimum_sampling_speed>
m = minute
s = second
ms = millisecond
us = microsecond
ns = nanosecond

Minimum arrival rate on serial interfaces

Describes the operation rate of any serial interfaces (see the groundhog_09_meta_document.pdf for more details).
< minimum_arrival_rate_on_serial_interfaces>{integer}{m, s, ms, us, ps}</ minimum_arrival_rate_on_serial_interfaces>
m = minute
s = second
ms = millisecond
us = microsecond
ns = nanosecond

Minimum arrival rate on parallel interfaces

Describes the operation rate of any parallel interfaces (see the groundhog_09_meta_document.pdf for more details).
< minimum_arrival_rate_on_parallel_interfaces>{integer}{m, s, ms, us, ps}</ minimum_arrival_rate_on_parallel_interfaces>
m = minute
s = second
ms = millisecond
us = microsecond
ns = nanosecond

Example structure of an XML workload

Here is a sample structure of the workload file for an input workload.

```
<workload>
        <design_name>portkey</design_name>
        <workload_number>10</workload_number>
        <minimum_arrival_rate>25000000ns</minimum_arrival_rate>
        <minimum_operating_speed>32ns</minimum_operating_speed> <!-- can be related to
        clock speed of system-->
        <minimum_sampling_speed>31250ns</minimum_sampling_speed> <!-- heart beat of the
        system -->
        <minimum_arrival_rate_on_serial_interfaces>32000ns</minimum_arrival_rate_on_serial_in
        terfaces> <!-- can realte to the serial clock -->
        <minimum_arrival_rate_on_parallel_interfaces>32000ns</minimum_arrival_rate_on_parall
        el_interfaces> <!-- can relate to the parallel clock -->
        <additional_details>keypad: 8x7</additional_details>

        <conditions>
                <condition>
                        <if>
                        <!--the resource to search for to activate the condition -->
                                <resource>
                                        <name>interrupt</name>
                                </resource>
                        </if>
                        <time>Y</time>
                        <!--the resource to trigger at Y time units from current time -->
                        <resource>
                                …
                        <resource>
                </condition>
        </conditions>
        <events>
                <event>
                        <time>Y</time>
                        <!--the resource happening during this event at time Y -->
                        <resource>
                                <name>X</name>
                                <value0>?</value0>
                                …
                        <resource>
                        <resource>
                                …
                        </resource>
                </event>
        </events>
</workload>
```