CoreSight System Configuration Manager

Author: Mike Leach <mike.leach@linaro.org>
Date: October 2020

Introduction

The CoreSight System Configuration manager is an API that allows the programming of the CoreSight system with pre-defined configurations that can then be easily enabled from sysfs or perf.

Many CoreSight components can be programmed in complex ways - especially ETMs. In addition, components can interact across the CoreSight system, often via the cross trigger components such as CTI and CTM. These system settings can be defined and enabled as named configurations.

Basic Concepts

This section introduces the basic concepts of a CoreSight system configuration.

Features

A feature is a named set of programming for a CoreSight device. The programming is device dependent, and can be defined in terms of absolute register values, resource usage and parameter values.

The feature is defined using a descriptor. This descriptor is used to load onto a matching device, either when the feature is loaded into the system, or when the CoreSight device is registered with the configuration manager.

The load process involves interpreting the descriptor into a set of register accesses in the driver - the resource usage and parameter descriptions translated into appropriate register accesses. This interpretation makes it easy and efficient for the feature to be programmed onto the device when required.

The feature will not be active on the device until the feature is enabled, and the device itself is enabled. When the device is enabled then enabled features will be programmed into the device hardware.

A feature is enabled as part of a configuration being enabled on the system.

Parameter Value

A parameter value is a named value that may be set by the user prior to the feature being enabled that can adjust the behaviour of the operation programmed by the feature.

For example, this could be a count value in a programmed operation that repeats at a given rate. When the feature is enabled then the current value of the parameter is used in programming the device.

The feature descriptor defines a default value for a parameter, which is used if the user does not supply a new value.

Users can update parameter values using the configfs API for the CoreSight system - which is described below.

The current value of the parameter is loaded into the device when the feature is enabled on that device.

Configurations

A configuration defines a set of features that are to be used in a trace session where the configuration is selected. For any trace session only one configuration may be selected.

The features defined may be on any type of device that is registered to support system configuration. A configuration may select features to be enabled on a class of devices - i.e. any ETMv4, or specific devices, e.g. a specific CTI on the system.

As with the feature, a descriptor is used to define the configuration. This will define the features that must be enabled as part of the configuration as well as any preset values that can be used to override default parameter values.

Preset Values

Preset values are easily selectable sets of parameter values for the features that the configuration uses. The number of values in a single preset set, equals the sum of parameter values in the features used by the configuration.

e.g. a configuration consists of 3 features, one has 2 parameters, one has a single parameter, and another has no parameters. A single preset set will therefore have 3 values.

Presets are optionally defined by the configuration, up to 15 can be defined. If no preset is selected, then the parameter values defined in the feature are used as normal.

Operation

The following steps take place in the operation of a configuration.

1. In this example, the configuration is 'autofdo', which has an associated feature 'strobing' that works on ETMv4 CoreSight Devices.
2. The configuration is enabled. For example 'perf' may select the configuration as part of its command line:

   ```
   perf record -e cs_etm/autofdo/ myapp
   ```

   which will enable the 'autofdo' configuration.
3. perf starts tracing on the system. As each ETMv4 that perf uses for trace is enabled, the configuration manager will check if the ETMv4 has a feature that relates to the currently active configuration. In this case 'strobing' is enabled & programmed into the ETMv4.
4. When the ETMv4 is disabled, any registers marked as needing to be saved will be read back.
5. At the end of the perf session, the configuration will be disabled.

Viewing Configurations and Features

The set of configurations and features that are currently loaded into the system can be viewed using the configfs API.

Mount configfs as normal and the 'cs-syscfg' subsystem will appear:

```
$ ls /config
cs-syscfg  stp-policy
```

This has two sub-directories:

```
$ cd cs-syscfg/
$ ls
configurations  features
```

The system has the configuration 'autofdo' built in. It may be examined as follows:

```
$ cd configurations/
$ ls
autofdo
$ cd autofdo/
$ ls
description  feature_refs  preset1  preset3  preset5  preset7  preset9
enable       preset        preset2  preset4  preset6  preset8
$ cat description
Setup ETMs with strobing for autofdo
$ cat feature_refs
strobing
```

Each preset declared has a 'preset<n>' subdirectory declared. The values for the preset can be examined:

```
$ cat preset1/values
strobing.window = 0x1388 strobing.period = 0x2
$ cat preset2/values
strobing.window = 0x1388 strobing.period = 0x4
```

The 'enable' and 'preset' files allow the control of a configuration when using CoreSight with sysfs.

The features referenced by the configuration can be examined in the features directory:

```
$ cd ../../features/strobing/
$ ls
description  matches  nr_params  params
$ cat description
Generate periodic trace capture windows.
parameter 'window': a number of CPU cycles (W)
parameter 'period': trace enabled for W cycles every period x W cycles
$ cat matches
SRC_ETMV4
$ cat nr_params
2
```

Move to the params directory to examine and adjust parameters:

```
cd params
$ ls
period  window
$ cd period
$ ls
value
$ cat value
0x2710
# echo 15000 > value
```

```
# cat value
0x3a98
```

Parameters adjusted in this way are reflected in all device instances that have loaded the feature.

Using Configurations in perf

The configurations loaded into the CoreSight configuration management are also declared in the perf 'cs_etm' event infrastructure so that they can be selected when running trace under perf:

```
$ ls /sys/devices/cs_etm
cpu0  cpu2  events  nr_addr_filters         power  subsystem  uevent
cpu1  cpu3  format  perf_event_mux_interval_ms      sinks  type
```

The key directory here is 'events' - a generic perf directory which allows selection on the perf command line. As with the sinks entries, this provides a hash of the configuration name.

The entry in the 'events' directory uses perfs built in syntax generator to substitute the syntax for the name when evaluating the command:

```
$ ls events/
autofdo
$ cat events/autofdo
configid=0xa7c3dddd
```

The 'autofdo' configuration may be selected on the perf command line:

```
$ perf record -e cs_etm/autofdo/u --per-thread <application>
```

A preset to override the current parameter values can also be selected:

```
$ perf record -e cs_etm/autofdo,preset=1/u --per-thread <application>
```

When configurations are selected in this way, then the trace sink used is automatically selected.

Using Configurations in sysfs

Coresight can be controlled using sysfs. When this is in use then a configuration can be made active for the devices that are used in the sysfs session.

In a configuration there are 'enable' and 'preset' files.

To enable a configuration for use with sysfs:

```
$ cd configurations/autofdo
$ echo 1 > enable
```

This will then use any default parameter values in the features - which can be adjusted as described above.

To use a preset<n> set of parameter values:

```
$ echo 3 > preset
```

This will select preset3 for the configuration. The valid values for preset are 0 - to deselect presets, and any value of <n> where a preset<n> sub-directory is present.

Note that the active sysfs configuration is a global parameter, therefore only a single configuration can be active for sysfs at any one time. Attempting to enable a second configuration will result in an error. Additionally, attempting to disable the configuration while in use will also result in an error.

The use of the active configuration by sysfs is independent of the configuration used in perf.

Creating and Loading Custom Configurations

Custom configurations and / or features can be dynamically loaded into the system by using a loadable module.

An example of a custom configuration is found in ./samples/coresight.

This creates a new configuration that uses the existing built in strobing feature, but provides a different set of presets.

When the module is loaded, then the configuration appears in the configfs file system and is selectable in the same way as the built in configuration described above.

Configurations can use previously loaded features. The system will ensure that it is not possible to unload a feature that is currently in use, by enforcing the unload order as the strict reverse of the load order.