

DsxvmHardware

This page explain how to describe the hardware in Dsx-vm.

General description

We first start by creating a Hardware object:

```
Hardware(cluster_x, cluster_y, nb_proc)
```

- `cluster_x` : number of cluster in the abscisse axe (int)
- `cluster_y` : number of cluster in the ordinate axe (int)
- `nb_proc` : number of proc by cluster

Both `cluster_x` and `cluster_y` parameter represent the number of cluster in the platform. If you have a non-clustered platform, you should then set each value to 1.

Peripehrals

Once we have described the platform, we could attach different peripheral to it:

- **TIMER** : a timer peripheral, useful for the scheduling of the processors `Timer(name, pbase, channel_size, nb_channel)`
- **ICU** : a concentrator of interrupt line, mandatory if we got irqs on the platform `ICU(name, pbase, channel_size, nb_channel)`
- **XICU** : a concentrator of interrupt line and a timer, this component and the couple Timer/Icu can't be in the same platform `XICU(name, pbase, channel_size, nb_channel)`
- **Dma** : a dma component `Dma(name, pbase, channel_size, nb_channel)`
- **Tty** : a multi-tty terminal `Tty(name, pbase, channel_size, nb_channel)`
- **Fbf** : a frame-buffer `Fbf(name, pbase, channel_size, nb_channel)`
- **RAM** : a RAM memory `RAM(name, pbase, size)`
- **ROM** : a ROM memory, we must have one at 0xbfc00000 address with the a minimal size of 0x1000 `ROM(name, pbase, size)`
- **MwmrCoproTaskWrapper** : declare a coproc which is able to simulate the a C task by wrapping it in a SystemC coproc. The coproc (a SystemC module) will only be generated if a task have been mapped `MwmrCoproTaskWrapper(name, pbase, channel_size, nb_channel, sc_name)`

The parameters are the following:

- `name` : name of the component
- `pbase` : physical base address of the component
- `channel_size` : the size of one channel, or the size of the component if it doesn't support multiple channel
- `nb_channel` : number of channel, set to '1' if no channel
- `size` : size of the memory
- `sc_name` : the SystemC name of the coproc

Irqs

This is a special component able who is able to describe the the routing of the interrupt line.

```
Irq(proc_id, cluster_id, icu_irq_id, peri, channel_id)
```

- `proc_id` : the `proc_id` at which the irq is attached
- `icu_id` : the `icu_id` line at which the irq is attached
- `cluster_id` : the `cluster_id` which contain the `proc`, `icu` and the `peri`
- `peri` : the class of the peripheral from which the request is sent
- `channel_id` : the `channel_id` of the peripheral

Here's an example for a one cluster platform:

Another example for a multi-cluster platform: