TEMU
GRIOMMU Model

Mattias Holm

Version 1.0, 2019-12-12
Table of Contents

1. Introduction .......................................................... 1
2. Loading the Plugin .................................................... 1
3. Configuration .......................................................... 1
4. Attributes ............................................................. 1
   4.1. Properties ....................................................... 1
   4.2. Interfaces ....................................................... 2
   4.3. Ports ............................................................. 2
5. Limitations ........................................................... 3
1. Introduction

The GRIOMMU model is available in the GrIoMmu plugin.

2. Loading the Plugin

    import GrIoMmu

3. Configuration

The model should be attached in two directions:

Firstly, IO-devices need to have their memory access interface references routed through the IOMMU. To do this, connect the memory access iface ref in the device to `IOMMUAccessIface` in the IOMMU.

Secondly, the IOMMU needs to get access to the device’s AMBA PNP info. The info is used to populate the `MasterConfig` registers. To set the PNP info, attach it to the `devicePnp` array.

The `IOMMUAccessIface` and `devicePnp` array assumes that the same device indexes are used. Not connecting devices the correct way is undefined behaviour.

    // Connect command
    connect a=iommu.devicePnp[0] b=device:ApbIface ①
    connect a=device.mem b=iommu:IOMMUAccessIface[0] ②

    // Or with assignment syntax
    iommu.devicePnp[0] = device:ApbIface ①
    device.mem = iommu:IOMMUAccessIface[0] ②

① Index should match index on next line.
② Index should match index on previous line.

4. Attributes

4.1. Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abhFailingAccess</td>
<td>uint32_t</td>
<td>AHB failing access register</td>
</tr>
<tr>
<td>asmpAccessControl</td>
<td>[4 x uint32_t]</td>
<td>ASMP access control register</td>
</tr>
<tr>
<td>capability</td>
<td>[3 x uint32_t]</td>
<td>Capability register</td>
</tr>
</tbody>
</table>
## Name | Type | Description
--- | --- | ---
config.interrupt | uint8_t | Interrupt number
control | uint32_t | Control register
dataRamErrorInjection | uint32_t | Data RAM error injection register
devicePnp | [16 x iref / <unknown>] | Devices under IOMMU control
diagnosticCacheAccess | uint32_t | Diagnostic cache access register
diagnosticCacheAccessData | [8 x uint32_t] | Diagnostic cache access data register
diagnosticCacheAccessTag | uint32_t | Diagnostic cache access tag register
groupConfig | [16 x uint32_t] | Group config register
irqMask | uint32_t | Interrupt mask register
masterConfig | [16 x uint32_t] | Master config register
mem | iref / MemAccessIface | Main memory bus
object.timeSource | object | Time source object (a cpu or machine object)
status | uint32_t | Status register
tagRamErrorInjection | uint32_t | Tag RAM error injection register
tlbCacheFlush | uint32_t | TLB/cache flush register

### 4.2. Interfaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceIface</td>
<td>DeviceIface</td>
<td></td>
</tr>
<tr>
<td>IOMMUAccessIface</td>
<td>MemAccessIface</td>
<td>IOMMU memory access interfaces</td>
</tr>
<tr>
<td>MemAccessIface</td>
<td>MemAccessIface</td>
<td></td>
</tr>
<tr>
<td>ResetIface</td>
<td>ResetIface</td>
<td></td>
</tr>
</tbody>
</table>

### 4.3. Ports

<table>
<thead>
<tr>
<th>Prop</th>
<th>Iface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
5. Limitations

- Non known