

1. Interconnects

- 1. Vgmn
 - 2. LocalCrossbar
- 2. VCI Initiators
 - 1. Xcache
- 3. Processors
 - 1. Mips
- 4. VCI Targets
 - 1. MultiRam
 - 2. MultiTty
 - 3. Locks

Interconnects

Vgmn

- functionality : a generic VCI compliant micro-network
- Mandatory arguments:
 - ◆ instance name
- Optional arguments:
 - ◆ min_latency (one-way)
- Example:

```
my_vgmn = Vgmn("my_vgmn", 10)
```

- Defined ports:
 - ◆ getBoth(), getInit() and getTarget() : local ports, allocated on demand

LocalCrossbar

- functionality : a VCI compliant crossbar interconnect
- Mandatory arguments:
 - ◆ instance name
- Example:

```
my_lc = LocalCrossbar("lc0")
```

- Defined ports:
 - ◆ getBoth(), getInit() and getTarget() : local ports, allocated on demand
 - ◆ upstream, bidirectional port to upper-level interconnect

VCI Initiators

Xcache

- functionality: a direct mapping cache controller (separated instruction & data cache)
- Mandatory arguments:
 - ◆ instance name
- Optional arguments:
 - ◆ dcache_lines : number of lines in data cache
 - ◆ dcache_words : number of words per line in data cache
 - ◆ icache_lines : number of lines in instruction cache

- ◆ `icache_words` : number of words per line in instruction cache
- Example:


```
my_cache = Xcache( "my_cache",
                    dcache_lines = 32,
                    dcache_words = 8,
                    icache_lines = 32,
                    icache_words = 8 )
```
- Defined ports:
 - ◆ `cache`: to the CPU cache port
 - ◆ `vci`: to the micro-network

Processors

Mips

- Functionality : a MIPS R3000 micro-processor
- Mandatory arguments:
 - ◆ `name`
- Example:


```
my_proc = Mips("my_proc")
```
- Defined ports:
 - ◆ `cache`: to the cache's cache port
 - ◆ `irq[n]`: interrupt line ($0 \leq n < 6$)

VCI Targets

MultiRam

- !Mandatory arguments:
 - ◆ `name`
- Optional arguments:
 - ◆ a list of segments, allocated with `Segment()`
- Example:


```
my_ram = MultiRam("my_ram", seg1, seg2, seg3)
```
- Defined ports:
 - ◆ `vci`: to the micro-network

MultiTty

- functionality: a TTY controler (up to 256 TTYs)
- Mandatory arguments:
 - ◆ `instance name`
 - ◆ an ordered list of names (one name per emulated terminal)
- Example:


```
my_tty = MultiTty("my_tty_controler", "TTY0", "TTY1", "TTY2")
```
- Defined ports:
 - ◆ `vci`: to the micro-network
 - ◆ `irq[n]`: interrupt line ($0 \leq n < \text{nb of ttys}$)

Locks

- functionality : a locks controler
- Mandatory arguments:
 - ◆ instance name
- Example:

```
my_locks = Locks("my_locks_controler")
```

- Defined ports:
 - ◆ vci: to the micro-network