

# Standard usage

When using a VCI-compliant component, you must define explicitly VCI parameters:

```
Uses('vci_multi_ram',
    cell_size = 4,
    plen_size = 1,
    addr_size = 32,
    rerror_size = 1,
    clen_size = 1,
    rflag_size = 1,
    srcid_size = 8,
    pktid_size = 1,
    trdid_size = 1,
    wrplen_size = 1 )
```

When repeated for all components, this can be redundant and error prone (not even speaking of readability):

```
todo = Platform(
    Uses('vci_multi_ram', cell_size = 4, plen_size = 1, addr_size = 32, rerror_size = 1,
        clen_size = 1, rflag_size = 1, srcid_size = 8, pktid_size = 1, trdid_size = 1,
        wrplen_size = 1 ),
    Uses('vci_vgmn', cell_size = 4, plen_size = 1, addr_size = 32, rerror_size = 1,
        clen_size = 1, rflag_size = 1, srcid_size = 8, pktid_size = 1, trdid_size = 1,
        wrplen_size = 1 ),
    Uses('vci_xcache', cell_size = 4, plen_size = 1, addr_size = 32, rerror_size = 1,
        clen_size = 1, rflag_size = 1, srcid_size = 8, pktid_size = 1, trdid_size = 1,
        wrplen_size = 1 ),
    Uses('vci_timer', cell_size = 4, plen_size = 1, addr_size = 32, rerror_size = 1,
        clen_size = 1, rflag_size = 1, srcid_size = 8, pktid_size = 1, trdid_size = 1,
        wrplen_size = 1 ),
    Uses('mips'),
)
```

So you may factor out common parameters:

```
vci_params = dict(
    cell_size = 4,
    plen_size = 1,
    addr_size = 32,
    rerror_size = 1,
    clen_size = 1,
    rflag_size = 1,
    srcid_size = 8,
    pktid_size = 1,
    trdid_size = 1,
    wrplen_size = 1
)

todo = Platform(
    Uses('mips'),
    Uses('vci_locks', **vci_params),
    Uses('vci_multi_ram', **vci_params),
    Uses('vci_multi_tty', **vci_params),
    Uses('vci_timer', **vci_params),
    Uses('vci_vgmn', **vci_params),
    Uses('vci_xcache', **vci_params),
    Source('top.cc'),
)
```

The `**variable_name` will be expanded with all the things defined in the variable, for every line, source:trunk/soclib/platforms/timer\_4mips/platform\_desc is a live example.

If you have some exceptionnal template parameter to add in for one component, you can specify it *before* the expanded one:

```
vci_params = dict(  
    cell_size = 4,  
    [...]  
)  
  
[...]  
    Uses('vci_vgmn', write_buffer_depth = 16, **vci_params),  
[...]
```