# Mapping Description

In this section we will see how one can describe the mapping of the different soft object in DSX-VM.

## Mapper object

This python object must be initialized first. It has the following arguments :

```
Mapper(hard, tcg1, tcg2 ...)
```

- the first argument is a python description of the hardware (see https://www-asim.lip6.fr/trac/dsx/wiki/DsxvmHardware).
- the rest of arguments is either one or more *tcg* objects of applications.

## The mapping

Once we declared the Mapper object we can use the **'map**' methode to map the different soft object. The arguments of the '**map**' function depend on the type of object to be mapped:

- For soft resources (mwmr channel, barriers, locks):

```
mapper.map(soft_object, pseg = physical_segment_to_map_into)
```

- For task objects :
  - If we map the task on a processor :

```
mapper.map(soft_object, cluster = cluster_number, proc = proc_number, stack = physi
```

  - If we map the task on a coprocessor :

```
m.map(task, coprocessor = coproc)
```

- For the Tcg objects :

```
m.map(tcg, code = physical_segment_to_map_into,  data = physical_segment_to_map_into,  pt
```

- For the Kernel (GIET-VM) objects:

```
m.map('system', boot = physical_segment_to_map_into,  kernel = physical_segment_to_map_in
```

The meaning of the arguments is :

- soft_object : this is the only mandatory argument, it can either be the name of the object or the object itself
- physical_segment_to_map_into : the the physical segment(RAM/ROM objects) on which the mapping is done

## Example

We will see the mapping of the a Hello world application.

- The Hardware description :

```
#!/usr/bin/env python
```

```
from dsx.hard.hard import *


def AlmoArch(nb_proc = 4, nb_tty = 2, timer_pbase = 0x91000000, icu_pbase = 0x9F000000):

    cluster_x = 1
    cluster_y = 1
    nb_clusters = cluster_x * cluster_y
    hd = Hardware(cluster_x, cluster_y, nb_proc)

    ####### peripherals ########
    hd.add(Tty('tty', pbase = 0x90000000, channel_size = 16, nb_channel = nb_tty))

    for cl in range(nb_clusters):
        hd.add(Timer('timer%d'%cl, pbase = timer_pbase + (cl * hd.cluster_span), channel_
        hd.add(Icu  ('icu%d'%cl  , pbase = icu_pbase + (cl * hd.cluster_span)  , channel_


    ######## irqs ########
    for i in xrange(nb_proc):
        hd.add(Irq(proc_id = i , cluster_id = 0, icu_irq_id = i, peri = Timer , channel_i


    ##### mwmr_coproc ######
    hd.add(MwmrCoprocTaskWrapper("tg_coproc"    , pbase = 0x94000000, channel_size = 32,
    hd.add(MwmrCoprocTaskWrapper("ramdac_coproc", pbase = 0x95000000, channel_size = 32,

    ######## Mems ########
    hd.add(ROM("PSEG_ROM", pbase = 0xbfc00000, size = 0x00010000))
    hd.add(RAM("PSEG_RAU", pbase = 0x00000000, size = 0x01000000))
    hd.add(RAM("PSEG_RAK", pbase = 0x80000000, size = 0x00100000))

    return hd
```

- The TCG description :

```
#!/usr/bin/env python

import dsx


tcg = dsx.Tcg("hello", dsx.Task('main', 'hello'), )
```

- The mapping description :

```
#!/usr/bin/env python

import dsx

from almo import AlmoArch
from app import tcg
from dsx.mapper.mapper import Mapper


hd =  AlmoArch(nb_tty = 2)

mpr = Mapper(hd, tcg)


mpr.map('main', cluster = 0, proc = 0, stack = 'PSEG_RAU')
mpr.map('hello', code = 'PSEG_RAU', data = 'PSEG_RAU', ptab = 'PSEG_RAU')
mpr.map('system' , boot = 'PSEG_ROM', kernel = 'PSEG_RAK')
```

```
mpr.generate(dsx.Giet(outdir = 'plateform', debug = True))
```

More examples can be found in the *examples* directory in DSX-VM.