



```
from dsx.hardware import *
from soclib import *

class VgmnNoirqMono(SoclibGenericArchitecture):
    def architecture(self):

# Définition des segments

        cram0 = Segment("cram0", Cached)
        uram0 = Segment("uram0", Uncached)
        cram1 = Segment("cram1", __remplir__)
        uram1 = Segment("uram1", __remplir__)
        reset = Segment("reset", Cached, addr = 0xbfc00000)
        excep = Segment("excep", Cached, addr = __remplir__)

# Instanciation des composants matériels

        vgmn = Vgmn("vgmn", __remplir__)
        xcache = Xcache("xcache", __remplir__)
        processor = Mips("processor")
        tty = MultiTty("tty", __remplir__)
        ram0 = MultiRam("ram0", cram0, uram0, excep)
        ram1 = MultiRam("ram0", __remplir__)
        locks = Locks("locks")

# Définition de la net-list (connexions)

        processor.cache // xcache.cache
        xcache.vci // vgmn.getTarget()
        tty.vci // vgmn.getInit()
        ram0.vci // __remplir__
        ram1.vci // __remplir__
        locks.vci // __remplir__
        tty.irq[0] // processor.irq[0]

# Construction architecture et creation mapping table

        self.setBase(vgmn)

# Exportation des variables locales

        self.cram0 = cram0
        self.uram0 = uram0
        self.cram1 = cram1
        self.uram1 = uram1
        self.reset = reset
        self.excep = excep

        self.vgmn = vgmn
        self.xcache = xcache
        self.processor = processor
        self.tty = tty
        self.locks = locks
        self.ram0 = ram0
        self.ram1 = ram1
```