Annex I-

Building Blocks of the Layout design

David Alvarez Outerelo, Francisco Javier Diaz Otero
EI Telecomunicacion Campus Universitario s/n 36310 Vigo Spain

Following is included the library code of the main blocks of the layout design.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
Library of components for HHI-14
"""
from IPython import get_ipython
get_ipython().magic('reset -sf')
import nazca as nd
import hhi as hhi
import nazca.interconnects as ic
# =============================================================================
# Re-define interconnects
# You can in principle change the deafult values of interconnects,
# but be aware and careful not to degradade the performance
# =============================================================================
e17=ic.Interconnect(xs='E1700', radius=150, width=2)
met=ic.Interconnect(xs='MetalDC', radius=10)

# =============================================================================
# Elements definitions
# =============================================================================
mmi     = hhi.MMI1x2E1700()
mmi2x2  = hhi.MMI2x2E1700()
Tr17to2 = hhi.WGTE1700E200()
Tr2to17 = hhi.WGTE200E1700()
láser   = hhi.DFBsection(WL_L=1550.0)

# =============================================================================
# Parameters
# =============================================================================
lvoa=150
hvoa=150

# =============================================================================
# VOA function
# =============================================================================
def voa(pmci_length=250):
    """This function defines VOA.
    Parameters:
        pmci_length (float): length of PMCI
    Returns:
        Cell with voa BB."""
    with nd.Cell(name='VOA_len_{}'.format(pmci_length)) as voaBB:
        eopm_bb = hhi.PMCIsection(pmci_length)
        #part 1: BBs
        #N1=hhi.WGMETXE1700().put(0,0)
        left_mmi = mmi.put()
        e17.sbend(Ltot=lvoa,offset=hvoa ).put(left_mmi.pin['b0'])
        TrL1 = hhi.WGTE1700E200().put()
        hhi.BJl().put()
        hhi.ISOsection().put()
        top_eopm = eopm_bb.put()
        hhi.ISOsection().put()
        hhi.BJr().put()
        TrR1 = hhi.WGTE200E1700().put()
        xsuperior=hhi.WGMETXE1700().put()
        S1=e17.sbend(Ltot=lvoa,offset=-hvoa ).put()
        right_mmi = mmi.put('b1',S1.pin['b0'])
        #N2=hhi.WGMETXE1700().put(right_mmi.pin['a0'])
        e17.sbend(Ltot=lvoa,offset=-hvoa ).put(left_mmi.pin['b1'])
        hhi.WGTE1700E200().put()
        bot_eopm=PMsection(pmci_length).put()
        hhi.WGTE200E1700().put()
        xinferior=hhi.WGMETXE1700().put()
        S2=e17.sbend(Ltot=lvoa,offset=+hvoa ).put(right_mmi.pin['b0'])
        t1=top_eopm.pin['n0'].xya()
        t2=bot_eopm.pin['n0'].xya()
        t3=[t1[0],(t1[1]+t2[1])/2,t1[2]]
        pin1=(t3[0],t3[1],t3[2])
        t1=xsuperior.pin['d0'].xya()
        t2=xinferior.pin['c0'].xya()
        t3=[t1[0],(t1[1]+t2[1])/2,t1[2]]
        pin2=(t3[0],t3[1],t3[2])
        #Metal contacts
        met.strt_p2p(top_eopm.pin['n0'],bot_eopm.pin['n0']).put()
        met.strt_p2p(pin1,pin2).put()
        met.strt_p2p(xsuperior.pin['d0'],xinferior.pin['c0']).put()
        # part 4: add pins
        nd.Pin('a0', pin=left_mmi.pin['a0']).put()
        nd.Pin('b0', pin=right_mmi.pin['a0']).put()
        nd.Pin('s0', pin=top_eopm.pin['p0']).put()
        nd.Pin('s1', pin=bot_eopm.pin['p0']).put()
        nd.Pin('n0', pin=top_eopm.pin['n0']).put()
        nd.Pin('n1', pin=bot_eopm.pin['n0']).put()
        nd.Pin('m1', pin=xinferior.pin['d0']).put()
    return voaBB # <<< make sure to have correct indentation

# ==============================================================================
# Splitter
# ==============================================================================
def spliter(pmci_length=250):
    """This function defines spliter.
    Parameters:
        pmci_length (float): length of PMCI
    Returns:
        Cell with spliter BB."""
    with nd.Cell(name='Spliter_len_{}'.format(pmci_length)) as spliter:
#part 1: BBs
        eopm_bb = hhi.PMCIsection(pmci_length)
        #N1=hhi.WGMETXE1700().put(0,0)
        left_mmi = mmi.put()
        e17.sbend(Ltot=lvoa,offset=hvoa ).put(left_mmi.pin['b0'])
        TrL1 = hhi.WGTE1700E200().put()
        hhi.BJl().put()
        hhi.ISOsection().put()
        top_eopm = eopm_bb.put()
        hhi.ISOsection().put()
        hhi.BJr().put()
        TrR1 = hhi.WGTE200E1700().put()
        xsuperior=hhi.WGMETXE1700().put()
        S1=e17.sbend(Ltot=lvoa,offset=-hvoa-0.6 ).put()
        right_mmi = mmi2x2.put('b1',S1.pin['b0'])
        e17.sbend(Ltot=lvoa,offset=-hvoa ).put(left_mmi.pin['b1'])
        hhi.WGTE1700E200().put()
        bot_eopm=PMsection(pmci_length).put()
        hhi.WGTE200E1700().put()
        xinferior=hhi.WGMETXE1700().put()
        S2=e17.sbend(Ltot=lvoa,offset=+hvoa+0.6).put(xinferior.pin['b0'],right_mmi.pin['b0'])
        t1=top_eopm.pin['n0'].xya()
        t2=bot_eopm.pin['n0'].xya()
        t3=[t1[0],(t1[1]+t2[1])/2,t1[2]]
        pin1=(t3[0],t3[1],t3[2])
        t1=xsuperior.pin['d0'].xya()
        t2=xinferior.pin['c0'].xya()
        t3=[t1[0],(t1[1]+t2[1])/2,t1[2]]
        pin2=(t3[0],t3[1],t3[2])
        #Metal contacts
        met.strt_p2p(top_eopm.pin['n0'],bot_eopm.pin['n0']).put()
        met.strt_p2p(pin1,pin2).put()
        met.strt_p2p(xsuperior.pin['d0'],xinferior.pin['c0']).put()
        #part 4: add pins
        nd.Pin('a0', pin=left_mmi.pin['a0']).put()
        nd.Pin('b0', pin=right_mmi.pin['a1']).put()
        nd.Pin('b1', pin=right_mmi.pin['a0']).put()
        nd.Pin('s0', pin=top_eopm.pin['p0']).put()
        nd.Pin('s1', pin=bot_eopm.pin['p0']).put()
        nd.Pin('n0', pin=top_eopm.pin['n0']).put()
        nd.Pin('n1', pin=bot_eopm.pin['n0']).put()
        nd.Pin('m1', pin=xinferior.pin['d0']).put()
        nd.Pin('m0', pin=xsuperior.pin['c0']).put()
    return spliter

# ==============================================================================
# Spliter3
# ==============================================================================
def spliterv2(pmci_length=250):
    """This function defines spliter.
    Parameters:
        pmci_length (float): length of PMCI
    Returns:
        Cell with spliter BB."""
    with nd.Cell(name='Spliterv2_len_{}'.format(pmci_length)) as spliterv2:
#part 1: BBs
        eopm_bb = hhi.PMCIsection(pmci_length)
        #N1=hhi.WGMETXE1700().put(0,0)
        left_mmi = mmi.put()
        e17.sbend(Ltot=lvoa,offset=hvoa ).put(left_mmi.pin['b0'])
        xsuperior=hhi.WGMETXE1700().put()
        TrL1 = hhi.WGTE1700E200().put()
        hhi.BJl().put()
        hhi.ISOsection().put()
        top_eopm = eopm_bb.put()
        hhi.ISOsection().put()
        hhi.BJr().put()
        TrR1 = hhi.WGTE200E1700().put()
        S1=e17.sbend(Ltot=lvoa,offset=-hvoa-0.6 ).put()
        right_mmi = mmi2x2.put('b1',S1.pin['b0'])
        e17.sbend(Ltot=lvoa,offset=-hvoa ).put(left_mmi.pin['b1'])
        xinferior=hhi.WGMETXE1700().put()
        TrL2 = hhi.WGTE1700E200().put()
        bot_eopm=PMsection(pmci_length).put()
        hhi.WGTE200E1700().put()
        S2=e17.sbend(Ltot=lvoa,offset=+hvoa+0.6 ).put(right_mmi.pin['b0'])
        t1=top_eopm.pin['n0'].xya()
        t2=bot_eopm.pin['n0'].xya()
        t3=[t1[0],(t1[1]+t2[1])/2,t1[2]]
        pin1=(t3[0],t3[1],t3[2])
        t1=xsuperior.pin['d0'].xya()
        t2=xinferior.pin['c0'].xya()
        t3=[t1[0],(t1[1]+t2[1])/2,t1[2]]
        pin2=(t3[0],t3[1],t3[2])
        #Metal contacts
        met.strt_p2p(top_eopm.pin['n0'],bot_eopm.pin['n0']).put()
        met.strt_p2p(pin1,pin2).put()
        met.strt_p2p(xsuperior.pin['d0'],xinferior.pin['c0']).put()
        #part 4: add pins
        nd.Pin('a0', pin=left_mmi.pin['a0']).put()
        nd.Pin('b0', pin=right_mmi.pin['a1']).put()
        nd.Pin('b1', pin=right_mmi.pin['a0']).put()
        nd.Pin('s0', pin=top_eopm.pin['p0']).put()
        nd.Pin('s1', pin=bot_eopm.pin['p0']).put()
        nd.Pin('n0', pin=top_eopm.pin['n0']).put()
        nd.Pin('n1', pin=bot_eopm.pin['n0']).put()
        nd.Pin('m1', pin=xinferior.pin['d0']).put()
    return spliterv2

# ==============================================================================
# PM
# ==============================================================================
def PM(pmci_length=200):
    """Give me a description......
    PM function is now parameterized and the length parameter is transferred
    to the Cell name"""
    with nd.Cell(name='PM_len_{}'.format(pmci_length)) as PMBB:
        T2 = hhi.WGTE1700E200().put()
        hhi.BJl().put()
        hhi.ISOsection().put()
        PM = hhi.PMCIsection(pmci_length).put()
        hhi.ISOsection().put()
        hhi.BJr().put()
        T3 = hhi.WGTE200E1700().put()
        nd.Pin('a0', pin=T2.pin['a0']).put()
        nd.Pin('b0', pin=T3.pin['b0']).put()
        nd.Pin('s0', pin=PM.pin['p0']).put()
        nd.Pin('n0', pin=PM.pin['n0']).put()
    return PMBB  # <<< make sure to have the correct indentation
def PMsection(pmci_length=200):
    """Give me a description......
    PM function is now parameterized and the length parameter is transferred
    to the Cell name"""
    with nd.Cell(name='PM_len_{}'.format(pmci_length)) as PMBB:
        T2 = hhi.BJl().put()
        hhi.ISOsection().put()
        PM = hhi.PMCIsection(pmci_length).put()
        hhi.ISOsection().put()
        T3 =  hhi.BJr().put()


        nd.Pin('b0', pin=T2.pin['a0']).put()
        nd.Pin('a0', pin=T3.pin['b0']).put()
        nd.Pin('p0', pin=PM.pin['p0']).put()
        nd.Pin('n0', pin=PM.pin['n0']).put()
    return PMBB

# ==============================================================================
# PHD
# ==============================================================================
def phd():
    with nd.Cell(name='PHD') as phd:
        T4=hhi.WGTE1700E200().put()
        phd1=hhi.PDRF().put()
        nd.Pin('a0', pin=T4.pin['a0']).put()
        #    nd.Pin('b0', pin=phd1.pin['b0']).put()
        nd.Pin('s1', pin=phd1.pin['c1']).put()
        nd.Pin('n0', pin=phd1.pin['c2']).put()
        nd.Pin('n1', pin=phd1.pin['c0']).put()
    return phd

# ==============================================================================
# Test library components
# ==============================================================================
if __name__ == '__main__':
    voa().put(0)
    spliter().put(0, 800)
    spliterv2().put(2000, 800)
    nd.export_gds(filename='library.gds')
```

6.1.2 Codigo fuente principal principales.

```
# -*- coding: utf-8 -*-
"""
@author: Velocier
"""
from IPython import get_ipython
get_ipython().magic('reset -sf')
import library as lib # import library with your BBs
import nazca as nd
import hhi as hhi
import nazca.interconnects as ic
import nazca.demopackager.packages as demopackages

# ==============================================================================
# Verify versions of Nazca and PDK
# ==============================================================================
print('Nazca version: ', nd.__version__)
print('HHI PDK version: ', hhi.__version__)
# =============================================================================
# Re-define interconnects
# You can in principle change the deafult values of interconnects,
# but be aware and careful not to degradade the performance
# =============================================================================
e17=ic.Interconnect(xs='E1700', radius=150, width=2)
met=ic.Interconnect(xs='MetalDC', radius=10)

# =============================================================================
# DIE dimensions
# =============================================================================
Hcell = 2000
Lcell = 12000

#==============================================================================
# Create and initialize die and package
#==============================================================================
with nd.Cell(name='HHI_MPWcell') as HHI_MPWcell:
    # foundrt MPW template
    DIE = hhi.project(die_height=Hcell, die_length=Lcell, doublecell=False)
    die = DIE.cell().put(0)
    die.raise_pins()
    #packaging/pad information:
    DC1 = {
        'edge': 'top',
        'type': 'DC',
        'count': 5,
        'pitch': 250,
        'center': True,
        'edge_sep_side': 150,
        'edge_sep_front': 100}
    DC2 = {
        'edge': 'bottom',
        'type': 'DC',
        'count': 11,
        'pitch': 250,
        'center': True,
        'edge_sep_side': 100,
        'edge_sep_front': 100}
    pads = [DC1, DC2]
    PACKAGE = demopackages.Package2(pads=pads)
    PACKAGE.die_size(die_length=DIE.die_length, die_height=DIE.die_height)
    # Add all pads to the cell
    package = PACKAGE.cell().put(0)
    for name, pin in package.pin.items():
        if 'dc' in name:
            DCpad = hhi.pad_dc().put('rc', pin)
            nd.Pin(name=name, pin=DCpad.pin['a0']).put()

#==============================================================================
# DESIGN
#==============================================================================
posx=1000
posy=1000
with nd.Cell(name='HHI14_03_UVIGO') as david_design1:
    pack = HHI_MPWcell.put(0)
    pack.raise_pins(nd.bbox_pinnames)
    # define cells that will be reused
    voa_bb = lib.voa(100)
    phd_bb = lib.phd()
    pm_len=770
    pm_bb = lib.PM(pm_len)
    splitter_bb = lib.spliter(100)
    splitter_bbv2 = lib.spliterv2(100)
#    slitter2_bb = lib.spliter2input(100)
    gsg_pad=hhi.gsg_pad()
    # put all components
    L1= lib.láser.put(posx,posy)
    temp1=hhi.BJl().put('b0',L1.pin['a0'])
    phd1 = hhi.PDRF().put(temp1.pin['a0'])
    hhi.ISOsection().put(L1.pin['b0'])
    hhi.BJr().put()
    T2 = hhi.WGTE200E1700().put()
    R1=e17.strt(length=150).put()
    m1=hhi.MMI2x2E1700().put('a0',R1.pin['b0'])
    e17.strt(length=20).put(m1.pin['b1'])
    e17.bend(angle=-90).put()
    R3=e17.strt(length=100).put()
    e17.bend(angle=90).put()
    e17.strt(length=300).put()
    Tx6=hhi.WGMETXE1700().put()
    e17.strt(length=400).put()
    Tx7=hhi.WGMETXE1700().put()
    e17.strt(length=1900).put()
    Tx8=hhi.WGMETXE1700().put()
    e17.strt(length=450).put()
    Tx9=hhi.WGMETXE1700().put()
    e17.strt(length=1650).put()
    Tx10=hhi.WGMETXE1700().put()
    e17.strt(length=300).put()
    Tx11=hhi.WGMETXE1700().put()
    e17.strt(length=20).put(m1.pin['a1'])
    e17.bend(angle=90).put()
    Tx1=hhi.WGMETXE1700().put()
    e17.strt(length=120).put()
    Tx2=hhi.WGMETXE1700().put()
    S1=e17.bend(angle=-90).put()
    SSC1 = hhi.SSCLAT7E1700().put('b0', 0, posy-484-142)
    e17.sbend_p2p(S1.pin['b0'], SSC1.pin['a0']).put()
    e17.strt(length=20).put(m1.pin['b0'])
    e17.bend(angle=180).put()
    e17.strt(length=770).put()
    e17.bend(angle=-180).put()
    e17.strt(length=20).put()
    voa1 = voa_bb.put()
    pm = pm_bb.put()
    e17.bend(angle=-180).put()
    e17.strt(length=pm_len+960).put()
    e17.bend(angle=90).put()
    e17.strt(length=25).put()
    e17.bend(angle=90).put()
    Spliter2 = splitter_bb.put()
    e17.strt(length=50).put()
    e17.bend(angle=90).put()
    Tx3=hhi.WGMETXE1700().put()
    e17.strt(length=130).put()
    e17.bend(angle=-90).put()
    e17.strt(length=50).put()
    phd2 = phd_bb.put()
    e17.bend(angle=-30 ).put( Spliter2.pin['b1'])
    e17.bend(angle=+30 ).put()
    Spliter3 = splitter_bb.put()
    SSC2 = hhi.SSCLAT7E1700().put('b0', DIE.die_length,posy+690,180)
    e17.bend(angle=83).put(SSC2.pin['a0'])
    temp1= e17.strt(length=0).put()
    e17.sbend_p2p(Spliter3.pin['b0'], temp1.pin['b0']).put()

    e17.bend(angle=-30).put( Spliter3.pin['b1'])
    R2=e17.bend(angle=30).put()
    m2=hhi.MMI1x2E1700().put('b1',R2.pin['b0'])
    e17.sbend_p2p(m2.pin['b0'],Tx11.pin['b0']).put()
    Tx5=hhi.WGMETXE1700().put(m2.pin['a0'])
    Spliter4 = splitter_bbv2.put()
    e17.strt(length=20).put(Spliter4.pin['b0']  )
    e17.bend(angle=+90).put()
    e17.strt(length=50).put()
    e17.bend(angle=-90).put()
    phd3 = phd_bb.put()
    e17.strt(length=20).put(Spliter4.pin['b1']  )
    e17.bend(angle=-90).put()
    e17.strt(length=50).put()
    e17.bend(angle=-90).put()
    phd4 = phd_bb.put()
##### test circuit
    L2= hhi.DFB(WL_L=1550.0).put(7900,1430)
    phd5 = hhi.PDRF().put(L2.pin['a0'])
    hhi.WGTE200E1700().put(L2.pin['b0'])
    e17.strt(length=0).put()
    tx13=hhi.WGMETXE1700().put()
    e17.strt(length=20).put()
    m3=hhi.MMI1x2E1700().put()
    e17.bend(angle=+90).put()
    e17.bend(angle=-90).put()
    e17.bend(angle=-90).put()
    e17.bend(angle=+90).put()
    e17.bend(angle=+90).put()
    temp1=e17.bend(angle=-90).put()
    ttemp=hhi.WGTE200E1700().put('b0',temp1.pin['b0'])
    phd6=hhi.PDDC().put('a0',ttemp.pin['a0'])
    temp1=e17.strt(length=900).put(m3.pin['b1'])
    ttemp=hhi.WGTE200E1700().put('b0',temp1.pin['b0'])
    phd7=hhi.PDDC().put('a0',ttemp.pin['a0'])
   # connect to DC pads
    met.strt_bend_strt_p2p(L1.pin['d0'],Tx2.pin['d0']).put()
    met.strt(20).put(voa1.pin['s0'])
    met.bend(angle=-90).put()
    modp1=met.strt(2700).put()
    met.sbend_p2p(modp1.pin['b0'],pack.pin['dcT000']).put()
    met.strt(20).put(pm.pin['s0'])
    met.bend(angle=-90).put()
    modp2=met.strt(800).put()
    metpads1=met.strt(150).put(pack.pin['dcT001'])
    met.sbend_p2p(modp2.pin['b0'],metpads1.pin['b0']).put()
    met.strt(10).put(Spliter2.pin['s0'])
    met.bend(angle=-90).put()
    modp3=met.strt(900).put()
    metpads2=met.strt(200).put(pack.pin['dcT002'])
    met.sbend_p2p(modp3.pin['b0'],metpads2.pin['b0']).put()
    met.strt(25).put(Spliter3.pin['s0'])
    met.bend(angle=90).put()
    met.strt(300).put()
    met.bend(angle=90).put()
    inter1=met.strt(20).put()
    inter2=met.strt(50).put(Tx3.pin['d0'])
    met.sbend_p2p(inter1.pin['b0'],inter2.pin['b0']).put()
    met.strt(80).put(Tx3.pin['c0'])
    met.bend(angle=-90).put()
    modp4=met.strt(380).put()
    metpads3=met.strt(200).put(pack.pin['dcT003'])
    met.sbend_p2p(modp4.pin['b0'],metpads3.pin['b0']).put()
    # middle metal connections
    met.sbend_p2p(voa1.pin['s1'],Tx6.pin['c0'],Lstart=100).put()
    met.sbend_p2p(voa1.pin['m1'],Tx7.pin['c0'],Lstart=100).put()
    met.sbend_p2p(pm.pin['n0'],Tx7.pin['c0'],Lstart=100).put()
    met.sbend_p2p(Spliter2.pin['s1'],Tx8.pin['c0'],Lstart=30).put()
    met.sbend_p2p(Spliter2.pin['m1'],Tx9.pin['c0'],Lstart=30).put()
    met.sbend_p2p(Spliter3.pin['s1'],Tx10.pin['c0'],Lstart=20).put()
    met.sbend_p2p(Spliter3.pin['m1'],Tx11.pin['c0'],Lstart=30).put()
    met.sbend_p2p(L2.pin['d0'],Spliter3.pin['m0'],Lstart=110).put()
    temp1=met.strt(10).put(L2.pin['d1'])
    temp2=met.strt(95).put(tx13.pin['d0'])
    met.bend_strt_bend_p2p(temp2.pin['b0'],temp1.pin['b0']).put()
    met.strt(20).put(Spliter4.pin['s0'])
    inter1=met.bend(angle=90).put()
    met.strt_bend_strt_p2p(inter1.pin['b0'],Tx5.pin['c0']).put()
    ##bottom metal connections
    met.strt(100).put(pack.pin['dcB010'])
    inter1=met.bend(angle=90).put()
    met.sbend_p2p(Tx2.pin['c0'],inter1.pin['b0'],Lstart=30).put()


    met.strt(150).put(pack.pin['dcB009'])
    inter1=met.bend(angle=90).put()
    met.sbend_p2p(Tx1.pin['c0'],inter1.pin['b0'],Lstart=200).put()
    met.sbend_p2p(pack.pin['dcB008'],Tx6.pin['d0'],Lstart=200).put()
    met.sbend_p2p(pack.pin['dcB007'],Tx7.pin['d0'],Lstart=250).put()
    met.sbend_p2p(pack.pin['dcB006'],Tx8.pin['d0'],Lstart=300).put()
    met.sbend_p2p(pack.pin['dcB005'],Tx9.pin['d0'],Lstart=350).put()
    met.sbend_p2p(pack.pin['dcB004'],Tx10.pin['d0'],Lstart=350).put()
    met.sbend_p2p(pack.pin['dcB003'],Tx11.pin['d0'],Lstart=300).put()
    met.sbend_p2p(pack.pin['dcB002'],Tx5.pin['d0'],Lstart=250).put()
    met.sbend_p2p(pack.pin['dcB001'],Spliter4.pin['m1'],Lstart=200).put()
    met.sbend_p2p(pack.pin['dcB000'],Spliter4.pin['s1'],Lstart=150).put()
    ##GSG metal connections
    M1=hhi.gsg_bend(angle=-90).put(phd1.pin['c1'])
    hhi.gsg_strt(length=495).put()
    gsg_pad1=gsg_pad.put()
    M4=hhi.gsg_strt(length=700).put(L1.pin['c1'])
    gsg_pad1=gsg_pad.put('a0',M4.pin['b0'])
    M2=hhi.gsg_bend(angle=90).put(phd2.pin['s1'])
    gsg_pad1=gsg_pad.put('a0',M2.pin['b0'])
    M3=hhi.gsg_bend(angle=-90).put(phd3.pin['s1'])
    hhi.gsg_strt(length=745).put()
    gsg_pad1=gsg_pad.put()
    M5=hhi.gsg_bend(angle=90).put(phd4.pin['s1'])
    gsg_pad1=gsg_pad.put('a0',M5.pin['b0'])
    M6=hhi.gsg_bend(angle=-90).put(phd5.pin['c1'])
    hhi.gsg_strt(length=50).put()
    gsg_pad1=gsg_pad.put()
    hhi.gsg_strt(length=300).put(L2.pin['c1'])
    gsg_pad1=gsg_pad.put()
    # add logos
    hhi.nazca_logo.put(700, 1400)
    hhi.nazca_logo.put(11200, 700)
              attlantic = nd.image('logo_atlanTTic_antefirma_horizontal2.jpg', threshold=0.92, size=1900,   pixelsize=1,layer=24)
    attlantic.put(1200, 200)
    attlantic.put(8280, 165)
    vigo = nd.image('16185_universidad-de-vigo1.jpg', threshold=0.92, size=1900, pixelsize=1,layer=24)
    vigo.put(2500, 500)
    vigo.put(10800, 300)
###metal connections extra circuit
    L1d=hhi.pad_dc().put(8800,1850,90)
    met.sbend_p2p(L1d.pin['a0'],tx13.pin['c0'],Lstart=100).put()
    L2d=hhi.pad_dc().put(9750,1850,90)
    met.sbend_p2p(phd6.pin['p0'],L2d.pin['a0'],Lstart=10).put()
    L3d=hhi.pad_dc().put(10200,1850,90)
    L4d=hhi.pad_dc().put(10400,1850,90)
    L5d=hhi.pad_dc().put(10600,1850,90)
    met.strt(90).put(phd6.pin['n0'])
    temp1=met.bend(angle=90).put()
    met.strt_bend_strt_p2p(L3d.pin['a0'],temp1.pin['b0']).put()
    met.strt(90).put(phd7.pin['p0'])
    temp1=met.bend(angle=-90).put()
    met.strt_bend_strt_p2p(L4d.pin['a0'],temp1.pin['b0']).put()
    met.strt(90).put(phd7.pin['n0'])
    temp1=met.bend(angle=90).put()
    met.strt_bend_strt_p2p(L5d.pin['a0'],temp1.pin['b0']).put()
    #====Text
    nd.text(text="Phd láser1", height=70, layer=24).put(500,1800)
    nd.text(text="CTRL láser1", height=70, layer=24).put(1450,1800)
    nd.text(text="láser external input", height=70, layer=24).put(70,640)
    nd.text(text="Mod0", height=40, layer=24).put(5290,1750)
    nd.text(text="PMp0", height=40, layer=24).put(5620,1700)
    nd.text(text="Sp1 0", height=40, layer=24).put(5830,1700)
    nd.text(text="Sp2 0", height=40, layer=24).put(6100,1700)
    nd.text(text="Power reference", height=40, layer=24).put(6330,1700)
    nd.text(text="<=Phd láser2", height=40, layer=24).put(7600,1700)
    nd.text(text="CTRL láser2=>", height=40, layer=24).put(7600,1850)
    nd.text(text="CTRL Temp láser2", height=40, layer=24).put(8350,1800)
    nd.text(text="phd bends p0", height=40, layer=24).put(9400,1800)
    nd.text(text="phd bends n0", height=40, layer=24).put(9870,1850)
    nd.text(text="phd p0", height=40, layer=24).put(10240,1700)
    nd.text(text="phd n0", height=40, layer=24).put(10430,1700)
    nd.text(text="Quantum output", height=40, layer=24).put(11500,1850)
    nd.text(text="gsg1 mix", height=70, layer=24).put(11100,150)
    nd.text(text="gsg2 mix", height=70, layer=24).put(9214,150)
    nd.text(text="CTRL Temp láser1 d0", height=40, layer=24).put(4200,155)
    nd.text(text="T L1 d1", height=40, layer=24).put(4790,265)
    nd.text(text="Mod 1", height=40, layer=24).put(5080,265)
    nd.text(text="M0", height=40, layer=24).put(5400,265)
    nd.text(text="SP1 1", height=40, layer=24).put(5600,265)
    nd.text(text="SP1 m0", height=40, layer=24).put(5800,265)
    nd.text(text="SP2 1", height=40, layer=24).put(6070,265)
    nd.text(text="SP2 m0", height=40, layer=24).put(6320,265)
    nd.text(text="SP3 0", height=40, layer=24).put(6570,265)
    nd.text(text="SP3 m0", height=40, layer=24).put(6820,265)
    nd.text(text="SP3 1", height=40, layer=24).put(7100,265)
               nd.text(text="Designed by David Alvarez Outerelo and Fran Diaz Otero", height=60, layer=24).put(2200,150)
           nd.text(text="acknowledgments to Ronald, Kasia and Valentina of Bright Photonics", height=60, layer=24).put(2200,80)

#===============================================================================
# Export to GDS
#===============================================================================
nd.export_gds(topcells=david_design1,filename='HHI14_03_UVIGO.gds')
```