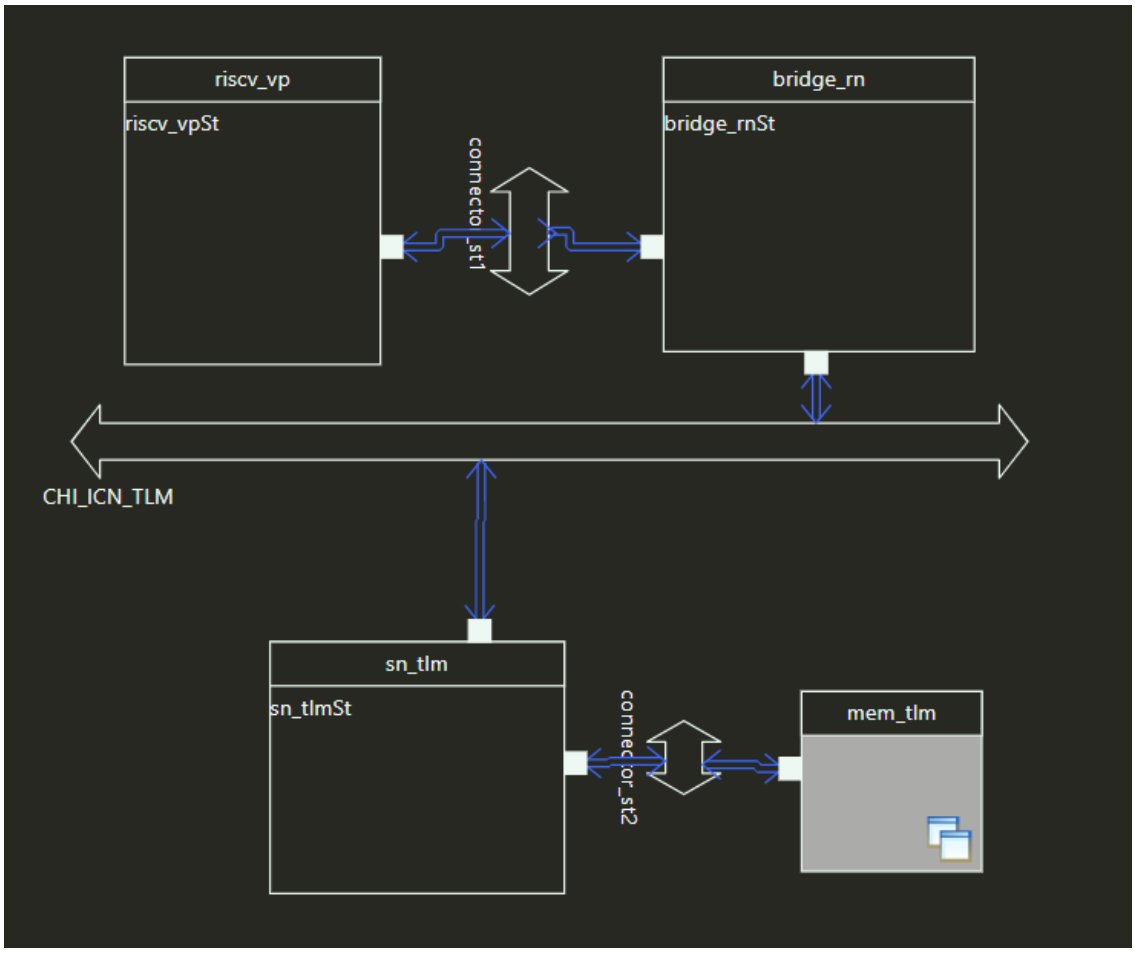
Consolidate RISCV-VP with CHI bus

model information

- **This model consolidates RISCV-VP(a simulated riscv cpu core) with a bus and a memory to a SoC system under CHI protocol.**
- **All components are connected by systemC and TLM2.0 interface.**
- **components**
  - RISCV-VP : A simulated cpu core of riscv architecture. It fetches instructions from inner instruction memory, decoding and executing them.
  - bridge-rn : A CHI cache. It transforms vp's transactions to CHI transactions, which is supported by CHI bus.
  - CHI_ICN_TLM : A CHI bus. It receives transactions from bridge-rn and initiates CHI transactions to sn_tlm. All CHI transactions keep cache coherency.
  - sn_tlm(slave node) : A CHI node. It receives CHI transactions and transforms them to non-chi form. Then it initiates transformed requests to downstream memory.
  - mem_tlm : A simulated memory module. It receives non-CHI TLM generic payload and serves as a common memory.
- **the full model looks like this**



environment dependency

- os : ubuntu 20.04 or higher
- compiler : gcc >= 7.5
- boost library

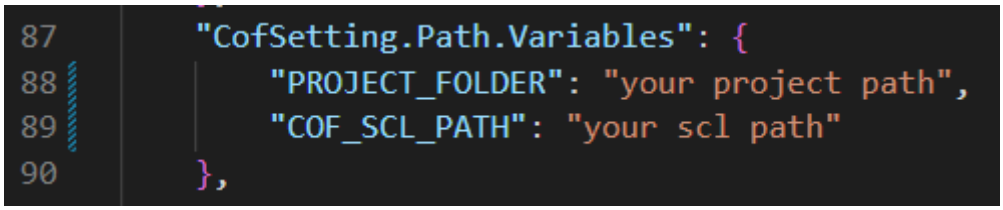
get model

```
sudo apt-get install libboost-all-dev
git clone https://github.com/CharlesChenGitHub/cofluent_soc_plf_tlm.git
git branch -b riscv-vp origin/riscv-vp
```

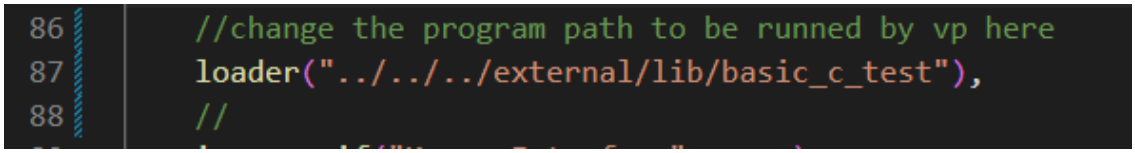
the model is under chi_icn_Sys/

run model

- first, open model folder(chi_icn_Sys/) in your workspace
- modify cofluent path varibles in cofbuild.json. input your project path and cof_scl path. for example: `~/cof-workspace/cofluent_soc_plf_tlm/chi_icn_Sys` and `~/cof-pack/cof-scl-isim-9.1.0-221024_nolic`

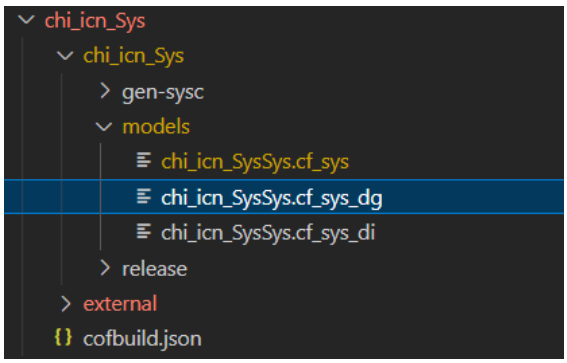


- riscv-vp reads program from `external/lib`, you can change the path of test program at `external/src/riscv_vp_adapter.cpp`,line 87

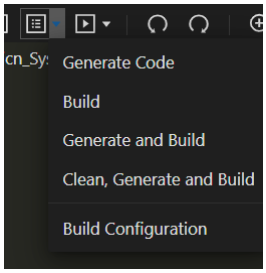


#the test program must be compiled by `riscv32-unknown-elf-gcc`, which is provided by https://github.com/riscv/riscv-gnu-toolchain

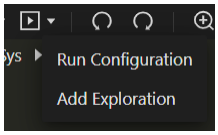
- open `chi_icn_Sys/models/chi_icn_SysSys.cf_sys_dg`

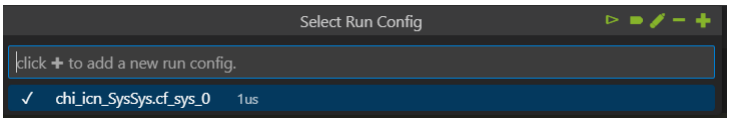


- directly generate and build



- create Run Configuration and run





- **NOTE**
  - Depending on the program you are running, the specified simulation time in cofrun.json may not be long enough. For example, the program takes 5690ns simulation time , but the "simulationDuration" option in cofrun.json may be shorter than it. Then the test program cannot get the expected result. So you need to modify "simulationDuration" in cofrun.json to adapt it to the model.

```
5670 ns    RN[0 ] -> HN[20]: 000100c0 Evict
5670 ns    HN[20] -> XN[0 ]: 00000000 RetryAck
5670 ns    HN[20] -> XN[0 ]: 00000000 PCrdGrant
5670 ns    RN[0 ] -> HN[20]: 000100c0 Evict
5670 ns    HN[20] -> XN[0 ]: 00000000 Comp
5670 ns    RN[0 ] -> HN[20]: 01ffffc0 ReadShared
5670 ns    HN[20] -> XN[10]: 01ffffc0 ReadNoSnp
5670 ns    SN[10] -> HN[20]: 00000000 RetryAck
5670 ns    SN[10] -> HN[20]: 00000000 PCrdGrant
5670 ns    HN[20] -> XN[10]: 01ffffc0 ReadNoSnp
5670 ns    SN[10] -> HN[20]: 00000000 CompData
5670 ns    HN[20] -> XN[0 ]: 00000000 CompData
5670 ns    RN[0 ] -> HN[20]: 01ffffc0 CompAck
5690 ns    RN[0 ] -> HN[20]: 01ffffc0 Evict
5690 ns    HN[20] -> XN[0 ]: 00000000 RetryAck
5690 ns    HN[20] -> XN[0 ]: 00000000 PCrdGrant
5690 ns    RN[0 ] -> HN[20]: 01ffffc0 Evict
5690 ns    HN[20] -> XN[0 ]: 00000000 Comp
5690 ns    RN[0 ] -> HN[20]: 000100c0 ReadShared
5690 ns    HN[20] -> XN[10]: 000100c0 ReadNoSnp
```

chi_icn_Sys > release > chi_icn_SysSys.cf_sysBuild_0 > {} cofrun.json > {} 0 > simulationDuration

```
[
    {
        "runName": "chi_icn_SysSys.cf_sys_0",
        "modelPath": "chi_icn_Sys/chi_icn_Sys/models/chi_icn_SysSys.cf_sys",
        "executionFilePath": "chi_icn_Sys/chi_icn_Sys/release/chi_icn_SysSys.cf_sysBuild_0/chi_icn_SysSys.exe",
        "generateTraceFile": false,
        "simulationDuration": "100us",
        "enableAlgorithmProfiling": false,
        "enableArchitectureProfiling": false,
        "verbosityLevel": "info",
        "args": "--cf-gui-connect=yes --cf-gui-ip-address=127.0.0.1 --cf-gui-socket=39000 --cf-gui-time-scale=ns --cf-mon-config=file",
        "active": true,
        "environment": "",
        "workingDirectory": "",
        "isim": {
            "enableISIM": false,
            "instanceName": "",
            "preElaboration": [],
            "postElaboration": [],
            "postSimulation": []
        }
    }
]
```

  - The version of your boost header files must be the same as the boost library file in `/external/lib`. In default, the boost lib files in this repo is version 1.71. You can check your boost version by `dpkg -S /usr/include/boost/version.hpp`.

If it is not 1.71, you need to substitute `external/lib/libboost_iostreams.a` & `external/lib/libboost_iostreams.so` by your boost libs. They can be found by `ls /usr/lib/x86_64-linux-gnu/ | grep iostream`

```
sunjiaqi@jiaqisu1-mob1:/usr/lib/x86_64-linux-gnu$ ls /usr/lib/x86_64-linux-gnu/ | grep iostream
libboost_iostreams.a
libboost_iostreams.so
```

expected output

When running, the console shows the CHI transactions by CHI bus, the risc-v assemble instructions by riscv-vp, like:

```
0 s        RN[0 ] -> HN[20]: 00010100 ReadShared
0 s        HN[20] -> XN[0 ]: 00000000 RetryAck
0 s        HN[20] -> XN[0 ]: 00000000 PCrdGrant
0 s        RN[0 ] -> HN[20]: 00010100 ReadShared
0 s        HN[20] -> XN[10]: 00010100 ReadNoSnp
0 s        SN[10] -> HN[20]: 00000000 RetryAck
0 s        SN[10] -> HN[20]: 00000000 PCrdGrant
0 s        HN[20] -> XN[10]: 00010100 ReadNoSnp
0 s        SN[10] -> HN[20]: 00000000 CompData
0 s        HN[20] -> XN[0 ]: 00000000 CompData
0 s        RN[0 ] -> HN[20]: 00010100 CompAck
core  0: prv 3: pc    1012c: JAL ra   (x1), 0xffffff48
50 ns      RN[0 ] -> HN[20]: 00010040 ReadShared
50 ns      HN[20] -> XN[10]: 00010040 ReadNoSnp
50 ns      SN[10] -> HN[20]: 00000000 CompData
50 ns      HN[20] -> XN[0 ]: 00000000 CompData
50 ns      RN[0 ] -> HN[20]: 00010040 CompAck
core  0: prv 3: pc    10074: ADDI sp   (x2), sp   (x2), 0xfffffe0
core  0: prv 3: pc    10078: SW sp   (x2), ra   (x1), 0x1c
80 ns      RN[0 ] -> HN[20]: 01ffffc0 ReadUnique
80 ns      HN[20] -> XN[0 ]: 00000000 RetryAck
80 ns      HN[20] -> XN[0 ]: 00000000 PCrdGrant
80 ns      RN[0 ] -> HN[20]: 01ffffc0 ReadUnique
80 ns      HN[20] -> XN[10]: 01ffffc0 ReadNoSnp
80 ns      SN[10] -> HN[20]: 00000000 RetryAck
80 ns      SN[10] -> HN[20]: 00000000 PCrdGrant
80 ns      HN[20] -> XN[10]: 01ffffc0 ReadNoSnp
80 ns      SN[10] -> HN[20]: 00000000 CompData
80 ns      HN[20] -> XN[0 ]: 00000000 CompData
80 ns      RN[0 ] -> HN[20]: 01ffffc0 CompAck
core  0: prv 3: pc    1007c: SW sp   (x2), s0/fp(x8), 0x18
130 ns     RN[0 ] -> HN[20]: 00010080 ReadShared
130 ns     HN[20] -> XN[10]: 00010080 ReadNoSnp
```

Finally, it prints vp state at end state of the program:

```
=[ core : 0 ]=========================
simulation time: 5810 ns
zero (x0) =        0
ra   (x1) =    10130
sp   (x2) =  1ffffc
gp   (x3) =        0
tp   (x4) =        0
t0   (x5) =        0
t1   (x6) =        0
t2   (x7) =        0
s0/fp(x8) =        0
s1   (x9) =        0
a0  (x10) =        0
a1  (x11) =       5b
a2  (x12) =        0
a3  (x13) =        0
a4  (x14) =        d
a5  (x15) =       5b
a6  (x16) =        0
a7  (x17) =       5d
s2  (x18) =        0
s3  (x19) =        0
s4  (x20) =        0
s5  (x21) =        0
s6  (x22) =        0
s7  (x23) =        0
s8  (x24) =        0
s9  (x25) =        0
s10 (x26) =        0
s11 (x27) =        0
t3  (x28) =        0
t4  (x29) =        0
t5  (x30) =        0
t6  (x31) =        0
pc = 10140
num-instr = 164
num-cycles = 312
[info]  Shutdown OK
```