# axislv2noc's interface & usage in practice

This file provides a brief introduction to the ports of `axislv2noc`, as well as how `axislv2noc` is used within ESP. Although we are referring to the original implementation of `axislv2noc` in this document, the same naming conventions were used throughout this entire project's codebase. Thus, familiarising yourself with this document will make it easier for you to understand all the code in this project. Note also, the code snippets shown here are typically an annotated copy of what is found in ESP. You can find the ESP repository here.

## Generic Map

- `tech` (unused) target technology, typically set by `CFG_FABTECH`. Used for target-specific code, which isn't required here. Can take the following values:

```
constant inferred    : integer := 0;
constant virtex7     : integer := 1;
constant virtexup    : integer := 2;
constant virtexu     : integer := 3;
constant gf12        : integer := 4;
```

- `nmst` number of masters.
- `retarget_for_dma` make all transactions DMA-based (i.e. `REQ_DMA_WRITE` / `DMA_TO_DEV`) regardless of tile type but does not change which NoC Planes are used. Note, DMA accesses are non-coherent.
- `mem_axi_port` if `-1` then all ports are memory, otherwise only the slave at the given index is memory. If a slave is memory, then the transaction reads/writes shared (possibly cached) data, otherwise the data is assumed to be uncached (e.g. accessing memory-mapped registers or I/O).
- `mem_num` number of memory tiles. It is the sum of a combination of the following,

```
constant CFG_NMEM_TILE : integer := /* autogen */; -- number of generic memory tiles (typical tile capacity is 1Gb)
constant CFG_NSLM_TILE : integer := /* autogen */; -- number of shared local memory tiles (typical tile capacity is 64Mb)
constant CFG_NSLMDDR_TILE : integer := /* autogen */; -- number of shared local LPDDR memory (typical tile capacity is 1Gb)
```

  Note, `SLM(DDR)` is high-speed scratchpad memory.

- `mem_info` identification information for each memory tile. Each memory tile has the following information,

```
constant tile_mem_info_none : tile_mem_info := (
    x => (others => '0'), -- tile grid x-pos
    y => (others => '0'), -- tile grid y-pos
    haddr => 16#000#,     -- 12-bit base address of memory tile
    hmask => 16#fff#      -- address mask, if non-zero then haddr is invalid
    );
```

- (`slv_y`, `slv_x`) is the miscellaneous tile grid position. This is used when accessing the miscellaneous tile (i.e. I/O tile). In this case we use the `remote_ahbs_***` ports to transfer data over NoC Plane #5. Clearly, there can be several connections (master <-> slave) to the miscellaneous tile for different operations.

## Port Map

`rst` asynchronous reset signal.

`clk` clock signal. `axislv2noc` is triggered on rising edge of clock.

`local_y` & `local_x` this tile grid position.

`mosi` Master Out Slave In. Set of AXI signals from this tile (masters) to remote tiles (slaves).

`somi` Slave Out Master In. Set of AXI signals from remote tiles (slaves) to this tile (masters).

### Coherence Ports

CPU tile (see `tile_cpu.vhd`) instantiates `axislv2noc` twice:

- The first instance uses the `coherence_***` ports if the CPU does not have the L2 cache enabled. This is because when the write-back L2 cache is enabled, coherent memory transactions are performed by the L2 cache itself (so not in the `tile_cpu`). If, instead, there isn't an L2 cache, then we must handle communication with the LLC found in SLM. To this end, the `coherence_***` ports are used to access the SLM. In this case, `coherence_req_***` is Tile to NoC Plane #1 (requests), and `coherence_rsp_***` is NoC Plane #3 to Tile (responses).
- Now, the second instance of `axislv2noc` always uses the `coherence_rsp_***` ports for NoC Plane #4 (coherent-DMA responses), `coherence_req_***` for NoC Plane #6 (coherent-DMA requests), and disables the `remote_ahbs_***` ports as these are already driven by the first instance. However, we don't enable `retarget_for_dma` because we want DMA accesses to be coherent/cached.

### Remote AHBS Ports

NoC Plane #5 is used to access memory mapped registers, perform I/O read and write operations, and other miscellaneous operations. That is, all operations that access the miscellaneous tile are done via NoC Plane #5.

`remote_ahbs_snd_***` is transaction from Tile to NoC Plane #5.

`remote_ahbs_rcv_***` is transaction from NoC Plane #5 to Tile.