ssbagi /
Workflow

Code Issues Pull requests Actions Projects Security Insights

main Workflow / README.md

ssbagi Update README.md 44cd866 · now

251 lines (198 loc) · 8.55 KB

Preview Code Blame Raw

Workflow

The Workflow for better and easy use of access which transforms the things.

NOTE : I am not interested in doing this. I maybe not the best person to dop this work. We can ask S/W Developers from Microsoft, Google or any S/W Tech Giants. Hire floks(200-500) to setup the flow. This can take 6months to 1 year. A new way of doing things for MT/VT/LT chip execution.

GIT Based Workflow :

- I maybe wrong and there might be few things which may seem a lot. Pepole are there to correct and provide feedback.
- I maybe not the best person we can ask S/W Developers from Microsoft, Google or any Tech Giants. Hire 200's of them to setup the flow. They can instruct max it can take 6months to 1 year. A new way of doing things for MT/VT/LT chips.
- This inclusion of workflow in Chip building will be paving path for Next Genenartion AI flow. You know we can query the work out of it. Make simpler seeing.
- A lot of repetative work like you know launching the runs, collecting the simulation results and checking sims terminated or many more stuff. We can use AI/AI Agents so in order to use it. We might need this Methodology workflow which can be GIT Based. Not enforcing just a scenario for VLSI flow for execution Engineers.
- I am newbie and faced a lot of time you know in these minor task which used to be hindrance to work. Go check the sims whehter they got launched or license issue or sims started wait of the results or probe the logs and get the values. Once ednde launch few more scripts to extract the KPI or Values or intended results.

- We can use GIT based, NBF or CI/CD Pipeline Chain.
- This might seem a lot but paving this for future VLSI flow for Execution Engineers. So, the Execution enginners can work in Chip design and Chip cycle. Upgrade the engineers into this flow.

Assume there are 100's Teams. So, from every team a Staff Engineer, a Senior/Lead Engineer given they have to be in team for moe than 2/3 years. They know a lot of things. A Program Manager for that paticular Sub block, block, Subsystem or Core level. I maybe not the best person we can ask S/W Developers from Microsoft, Google or any Tech Giants. Hire few of them to set this flow. They can instruct max it can take 6months to 1 year. A new way of doing things.

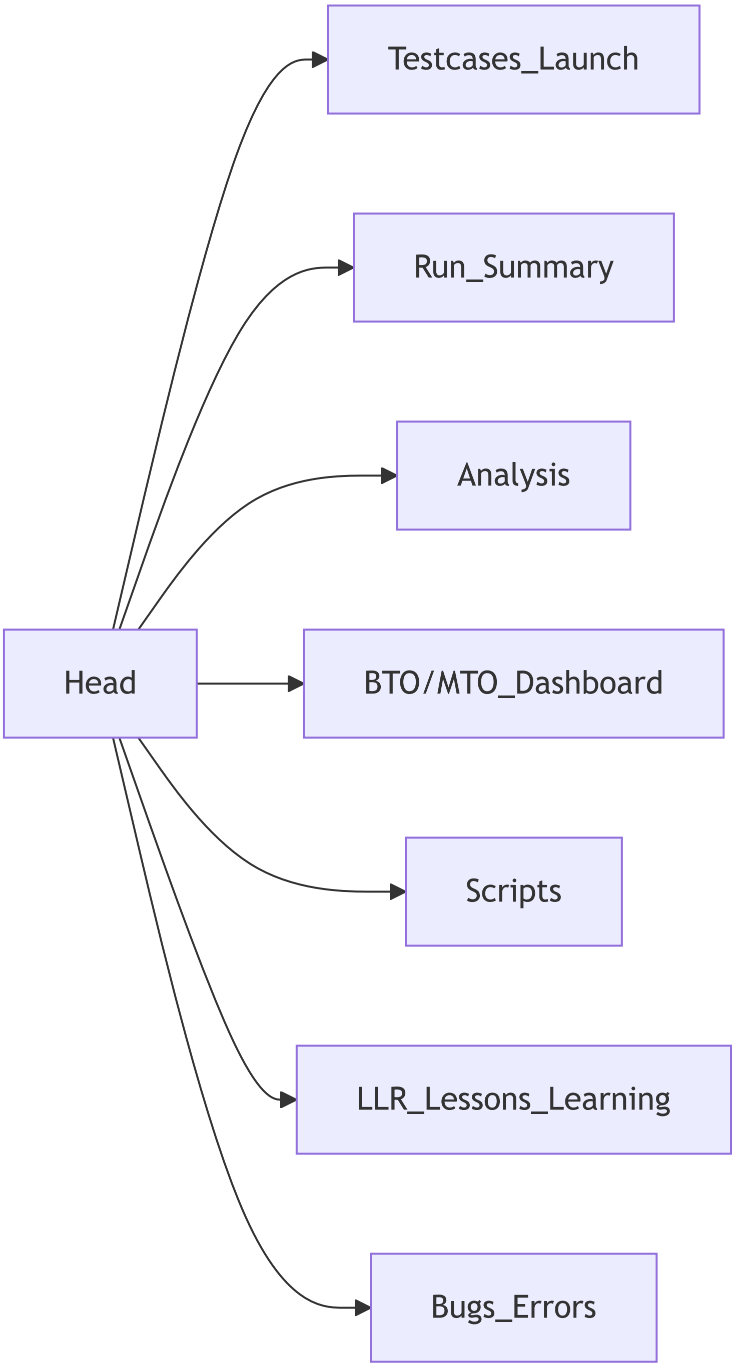
A team of Four :

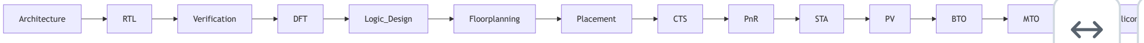
- Staff or Senior Staff
- Senior/Lead Engineer
- Program Manager
- Director/Prinicipal Engineer

As part of CCI we can restrict the access to only Staff and above people. Or limited access can be granted.

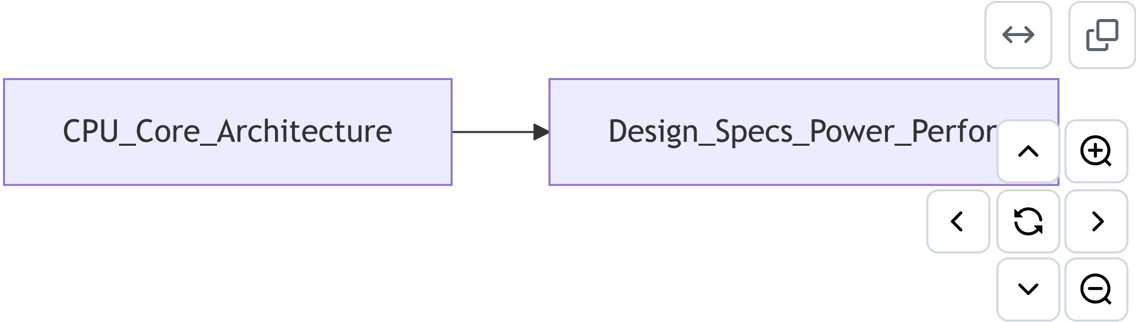
Create a setup chip or mimic some chip using these tools. Create a GIT based flow. Each team having their own git and integrate it to one Flow at End. SO, viewing from top to bottom becomes easy.

General Directiory Git Structure like this : A separate Repo for each Chip. No chip gating used create a new Repo everytime for each chip. This makes flow simpler and easier.





For example leat use assume CPU Core : Each team having different repo and merging all the repos.



A separate Git Repo capturing the detials of RTL codes of all the blocks integrating. For example when we see ARM Cortex CPU core : Reference : https://developer.arm.com/documentation/102530/0002/Technical-overview/Core-components

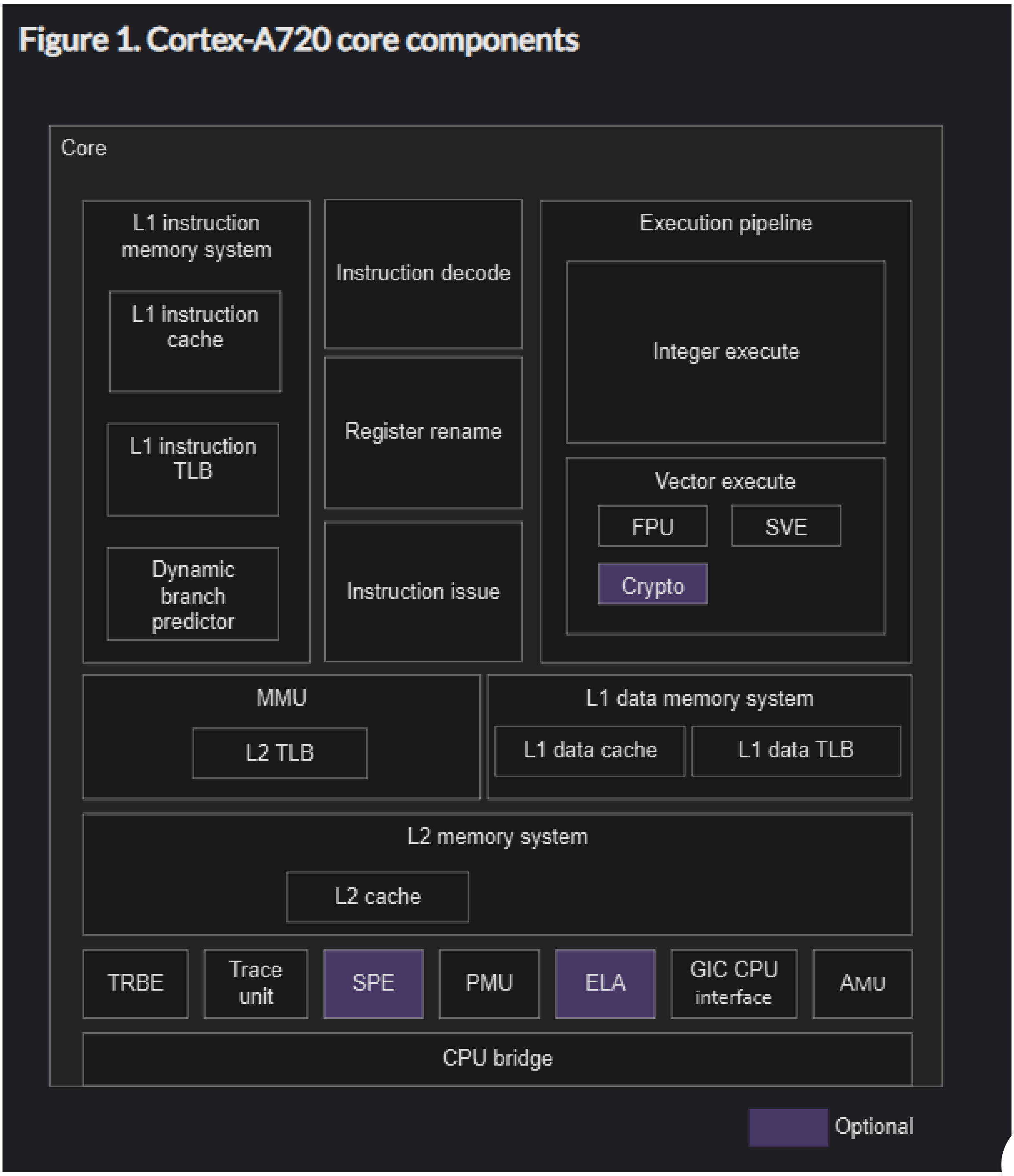


Figure 1. Cortex-A720 core components

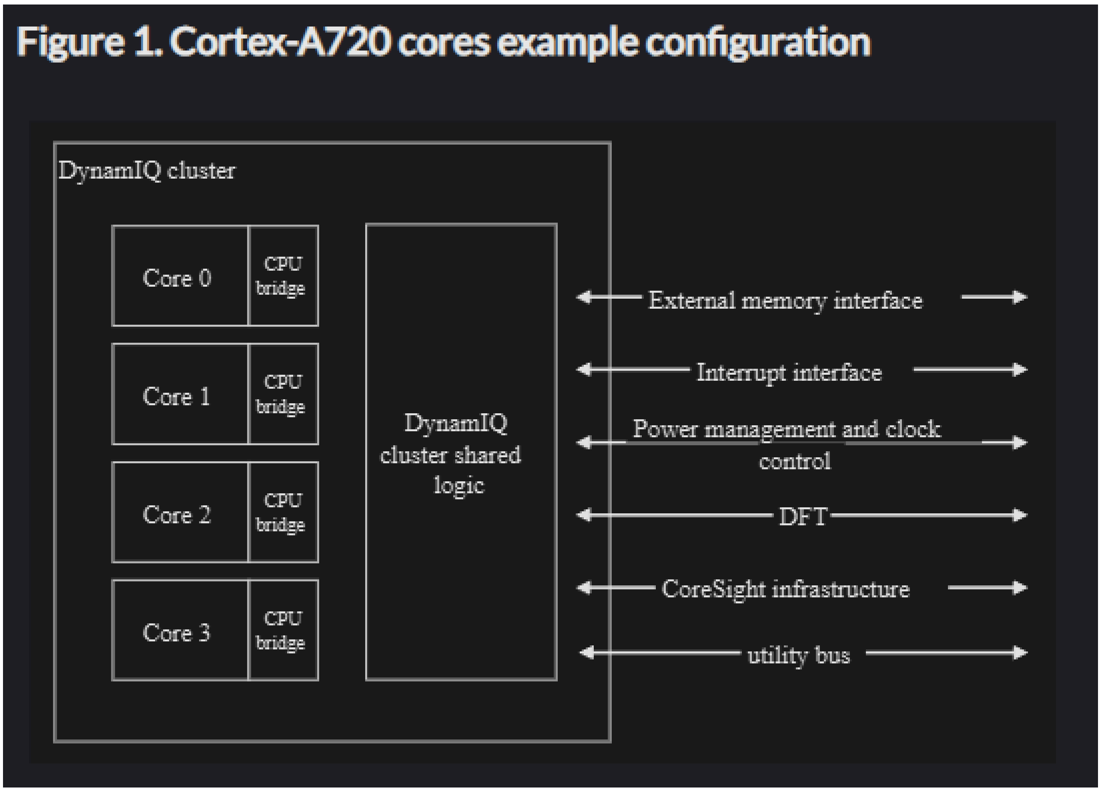


Figure 1. Cortex-A720 cores example configuration

DL1 :

- L1 Instruction memory system
- Instruction Decode
- Register Rename
- Instruction Issue
- Execution Pipeline
- MMU
- L1 data memory system
- L2 memory system
- CPU bridge
- TRBE
- Trace Unit
- SPE
- PMU
- ELA
- GIC CPU
- AMU

DL2 :

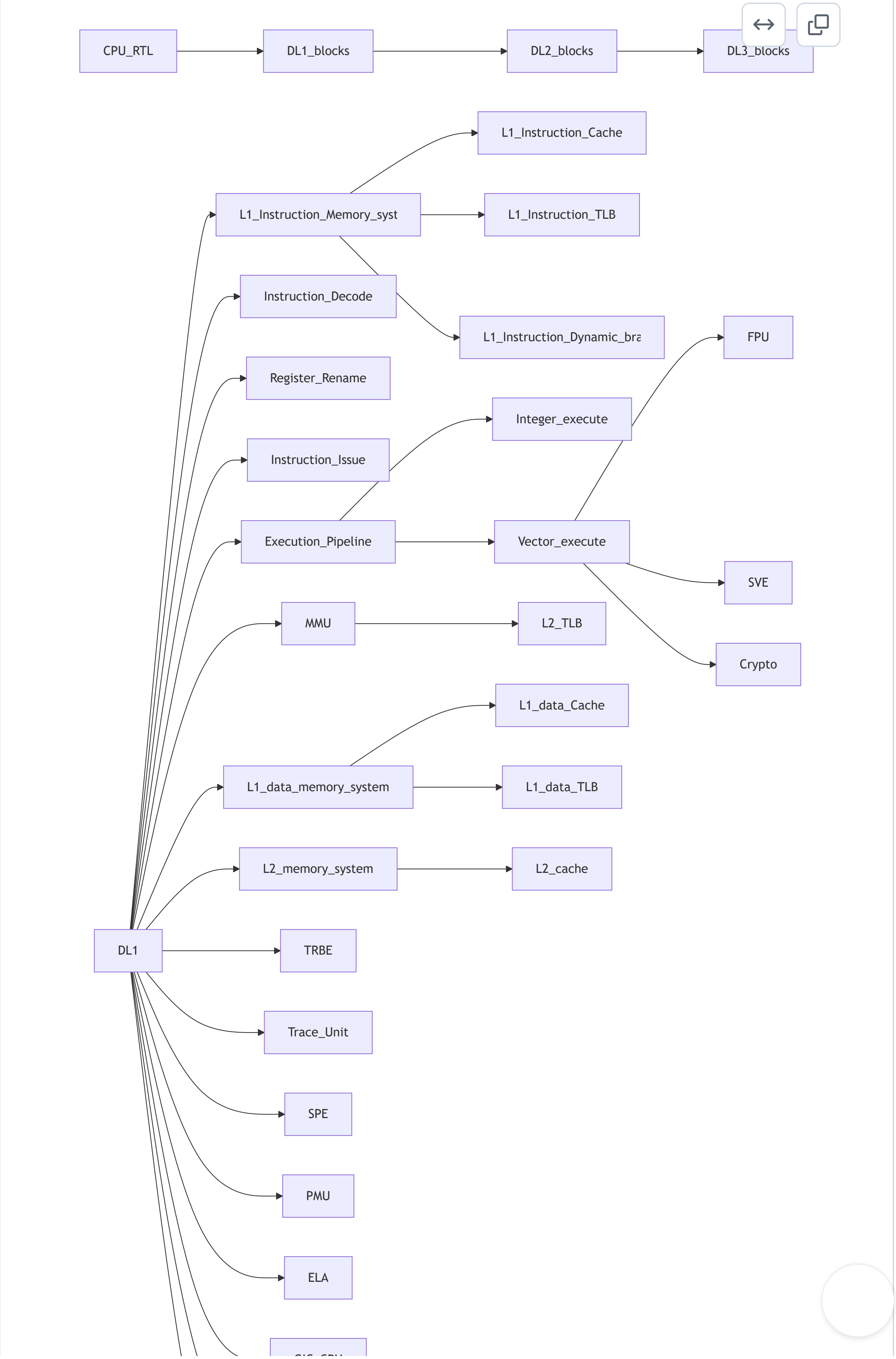
- L1 I cache
- L1 I TLB
- Dynamic Branch Predictor

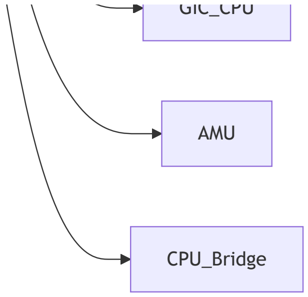
- Integer Execution Unit
- Vector Execute
- L2 TLB
- L1 d cache
- L1 d TLB
- L2 cache

DL3 :

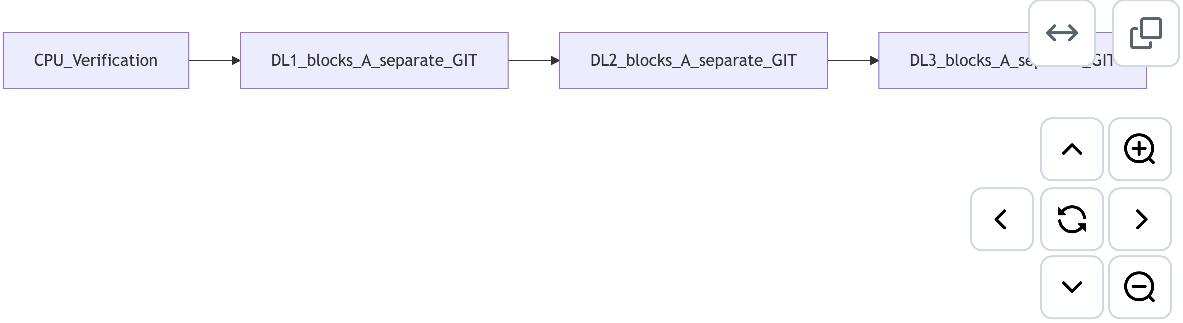
- Subblocks present inside DL2

We have RTL Integration team, RTL Core team, RTL Blocks team many more of them. To follow the order. To create a working system.

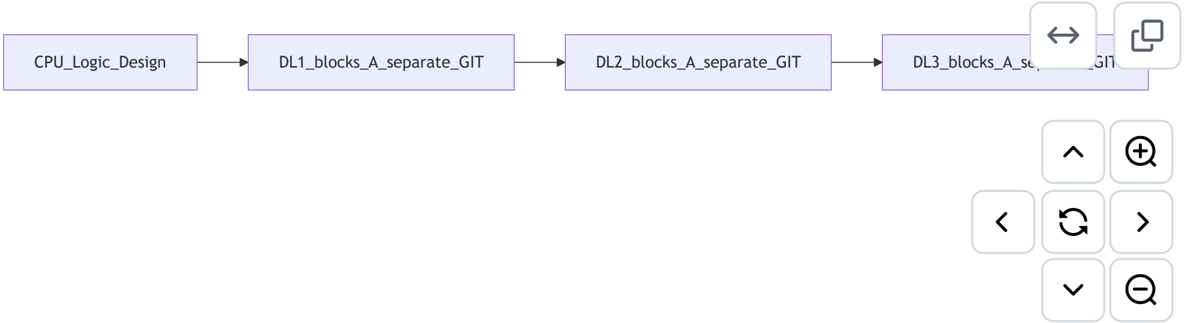

CPU_RTL
DL1_blocks
DL2_blocks
DL3_blocks
L1_Instruction_Cache
L1_Instruction_Memory_syst
L1_Instruction_TLB
Instruction_Decode
L1_Instruction_Dynamic_bra
FPU
Register_Rename
Integer_execute
Instruction_Issue
Execution_Pipeline
Vector_execute
SVE
MMU
L2_TLB
Crypto
L1_data_Cache
L1_data_memory_system
L1_data_TLB
L2_memory_system
L2_cache
DL1
TRBE
Trace_Unit
SPE
PMU
ELA

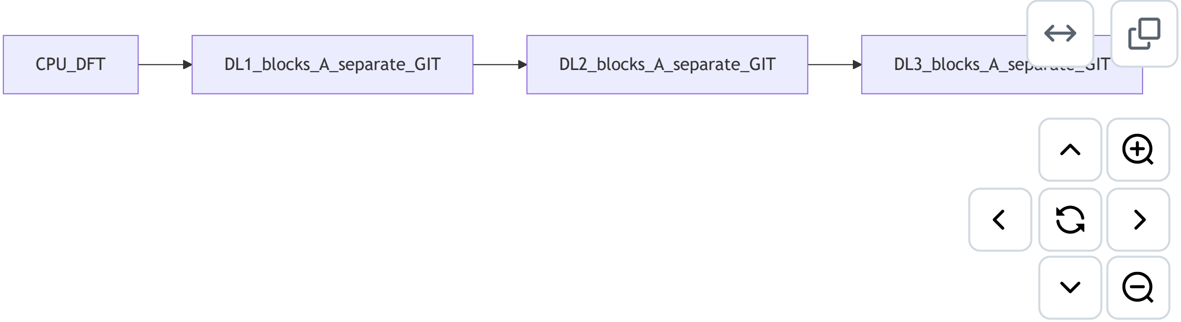


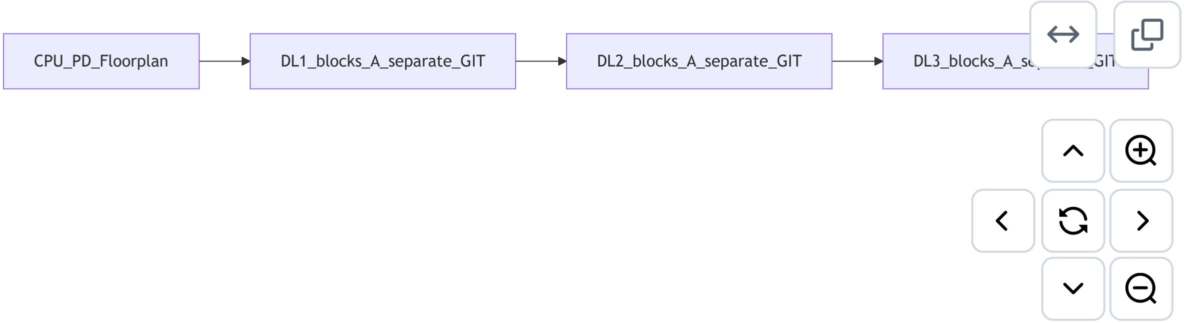
We can a Verification Git repo similarly as of RTL and create Testcases scenarios and all the stuff.

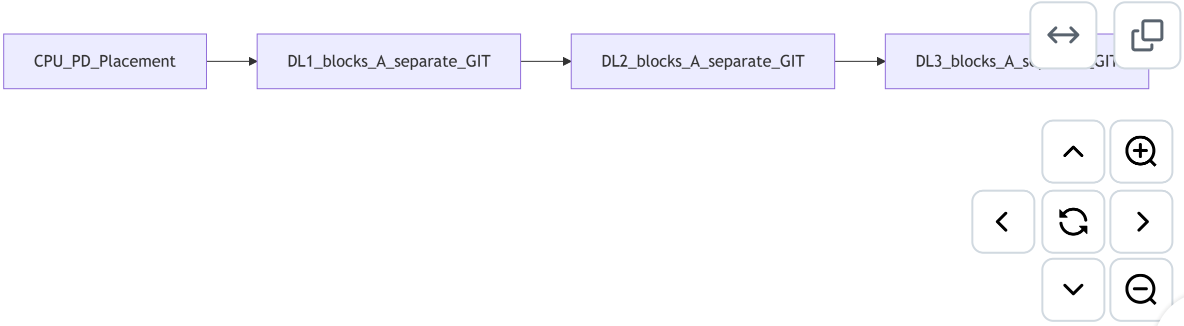


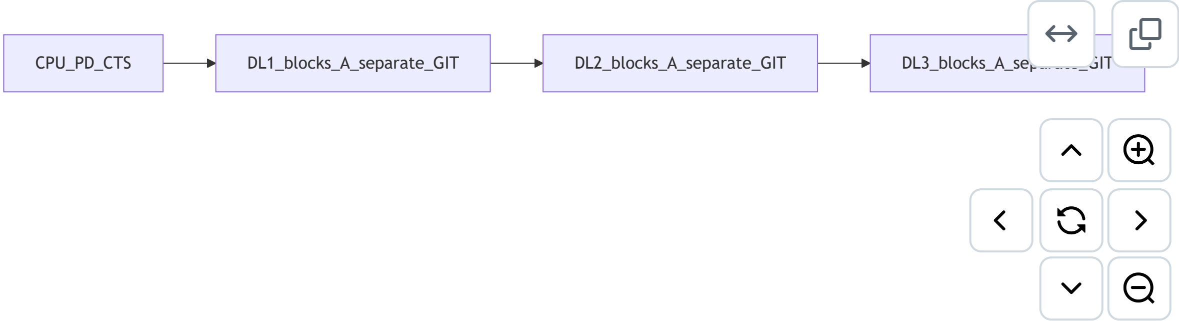
In a way Similar by all others in VLSI flow

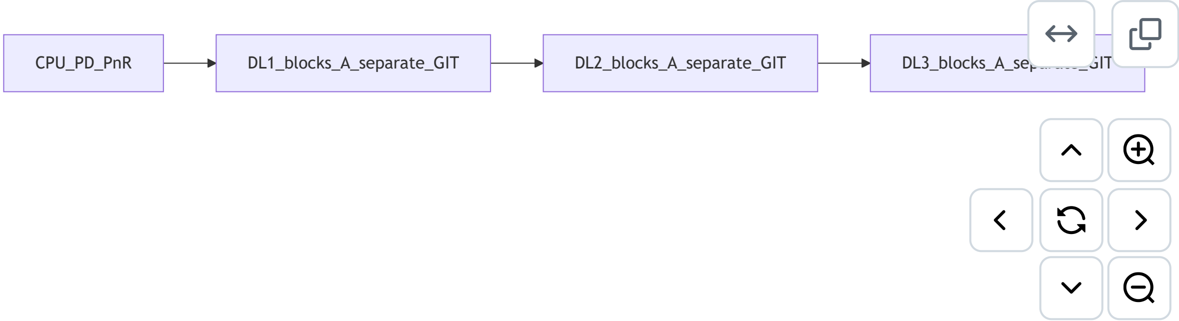


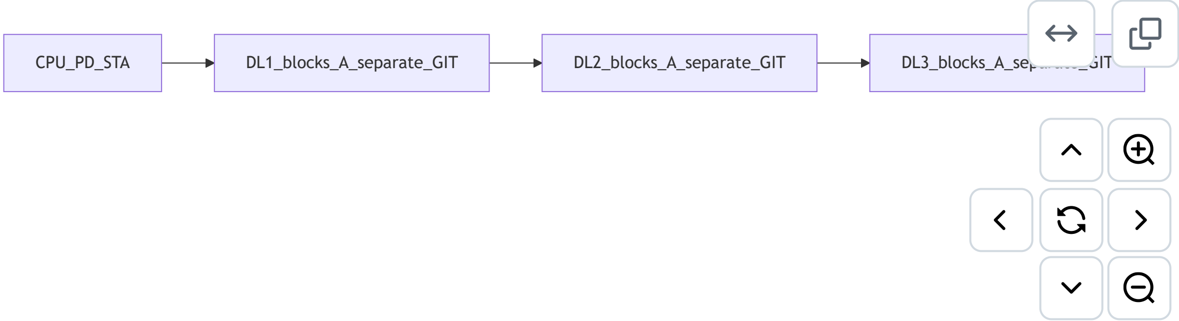

CPU_DFT
DL1_blocks_A_separate_GIT
DL2_blocks_A_separate_GIT
DL3_blocks_A_separate_GIT

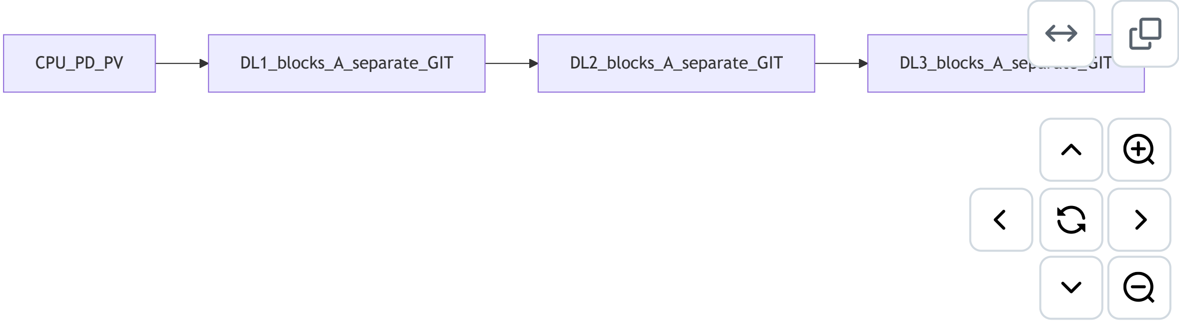


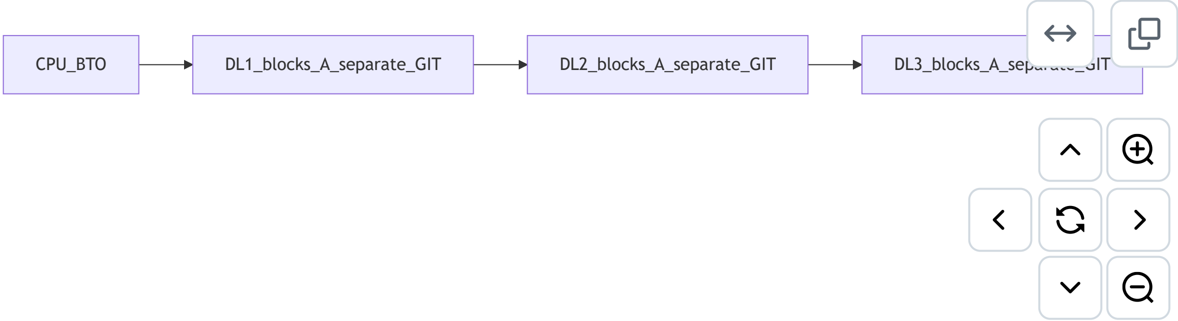


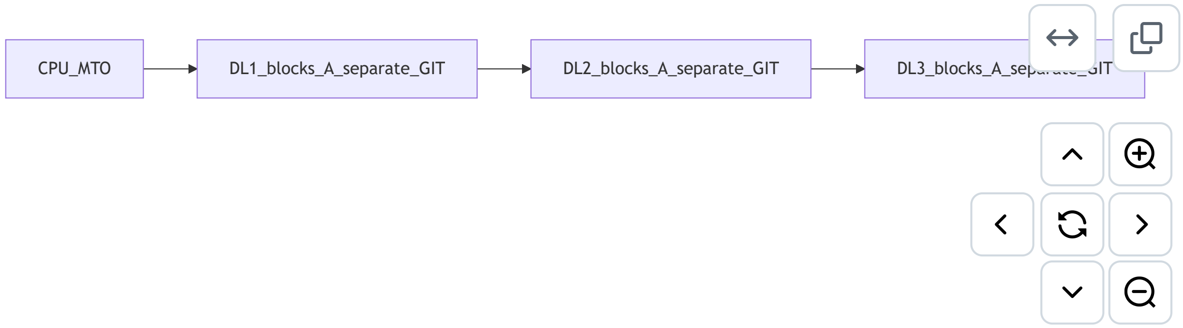

CPU_PD_CTS
DL1_blocks_A_separate_GIT
DL2_blocks_A_separate_GIT
DL3_blocks_A_separate_GIT

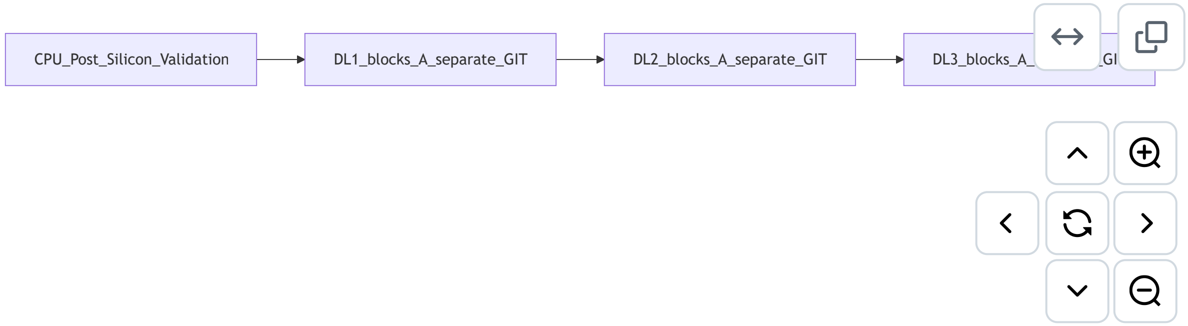

CPU_PD_PnR
DL1_blocks_A_separate_GIT
DL2_blocks_A_separate_GIT
DL3_blocks_A_separate_GIT


CPU_PD_STA
DL1_blocks_A_separate_GIT
DL2_blocks_A_separate_GIT
DL3_blocks_A_separate_GIT


CPU_PD_PV
DL1_blocks_A_separate_GIT
DL2_blocks_A_separate_GIT
DL3_blocks_A_separate_GIT


CPU_BTO
DL1_blocks_A_separate_GIT
DL2_blocks_A_separate_GIT
DL3_blocks_A_separate_GIT


CPU_MTO
DL1_blocks_A_separate_GIT
DL2_blocks_A_separate_GIT
DL3_blocks_A_separate_GIT


CPU_Post_Silicon_Validation
DL1_blocks_A_separate_GIT
DL2_blocks_A_separate_GIT
DL3_blocks_A_ GI

The above one is for CPU similarly we can make for GPU, NPU, Camera, Video, Audio, NoC's and all the blocks present in the SoC.

At the Integarte all to One Git Repo with multiple things to one. In one go we can see a lot of results and view the whole Chip only for restricted Access uage people to see. Now Map the things from Pre Silicon to Post Silicon and all the stuff. Any Violation or Bugs or Lessons Learning everything can be seen.