Github: [chaoNTUEE/HLS_Lab-A: This is for HLS Lab-A (github.com)]

此次 Lab 參考步驟 [Vitis-Tutorials/Getting_Started/Vitis_HLS at 2022.1 · Xilinx/Vitis-Tutorials (github.com)]，來了解如何分析資訊和 5 個幫助 optimize 的方式

DCT code structure, hierarchy:

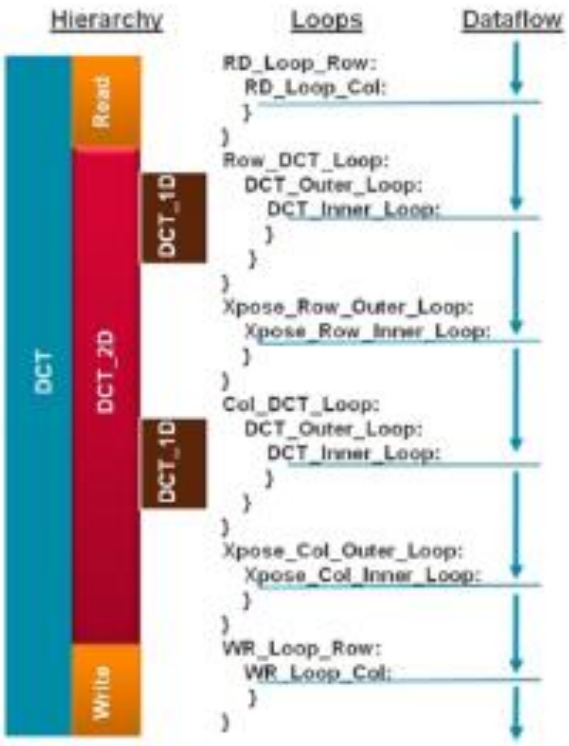
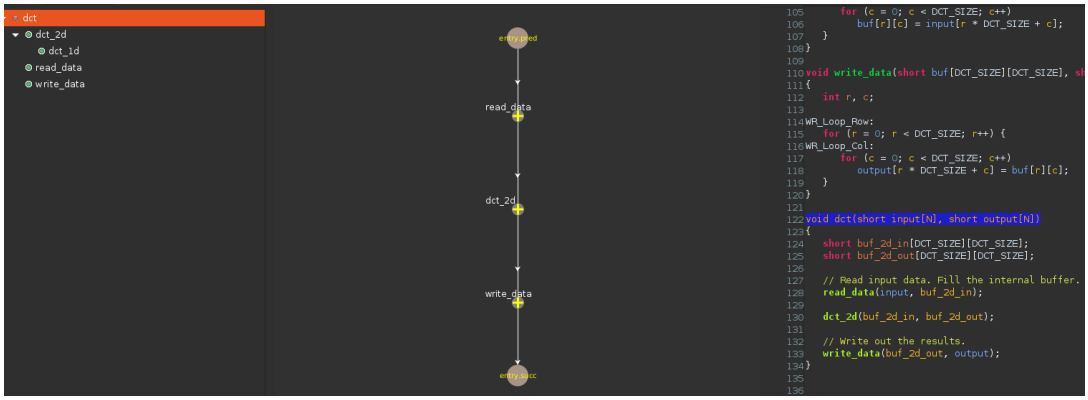
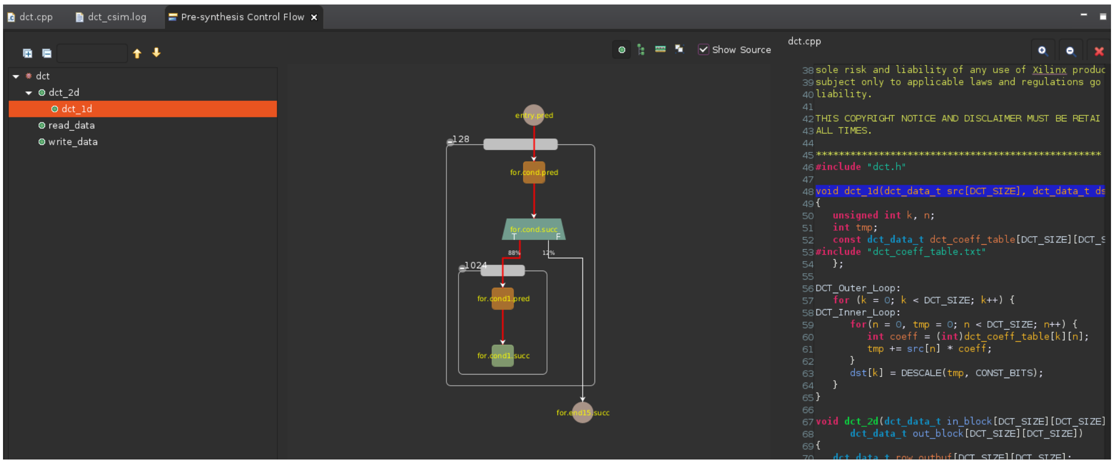


Figure 6-3: **Overview of the DCT Design**

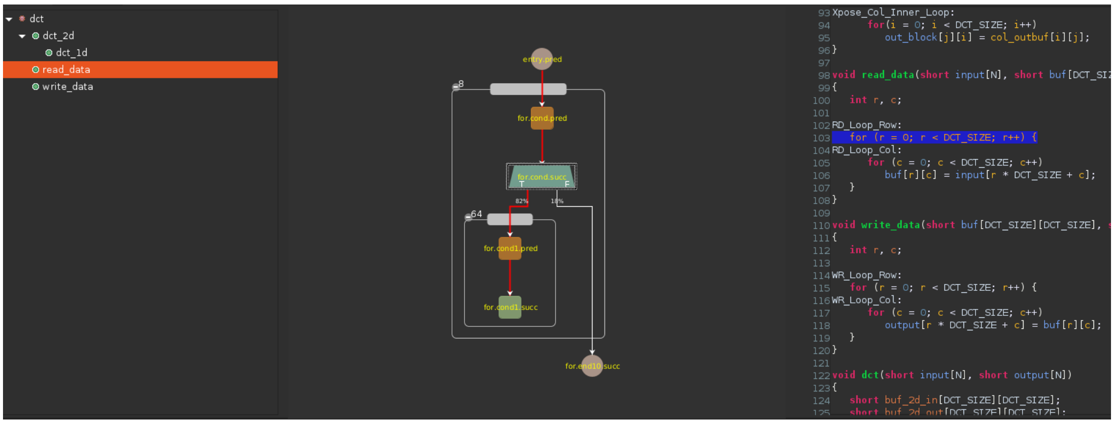
dct



Dct_id



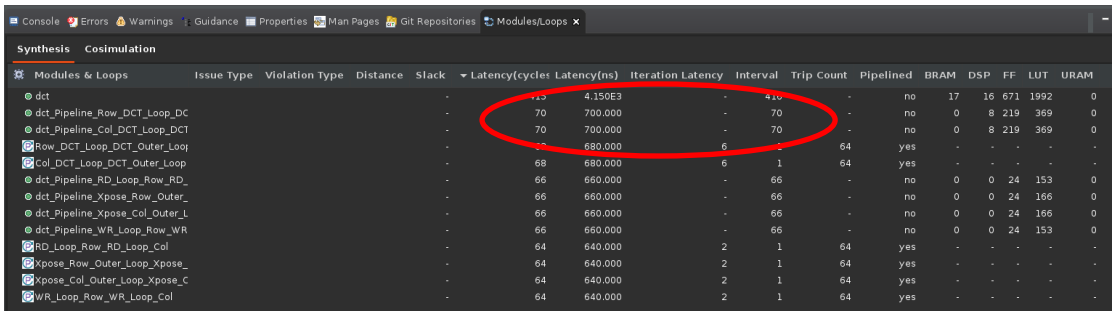
Read/write



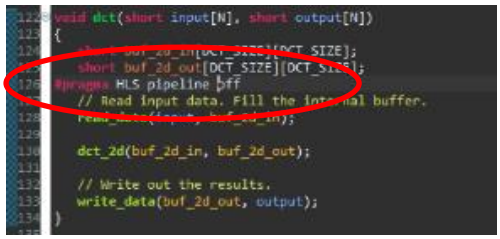
Latency:

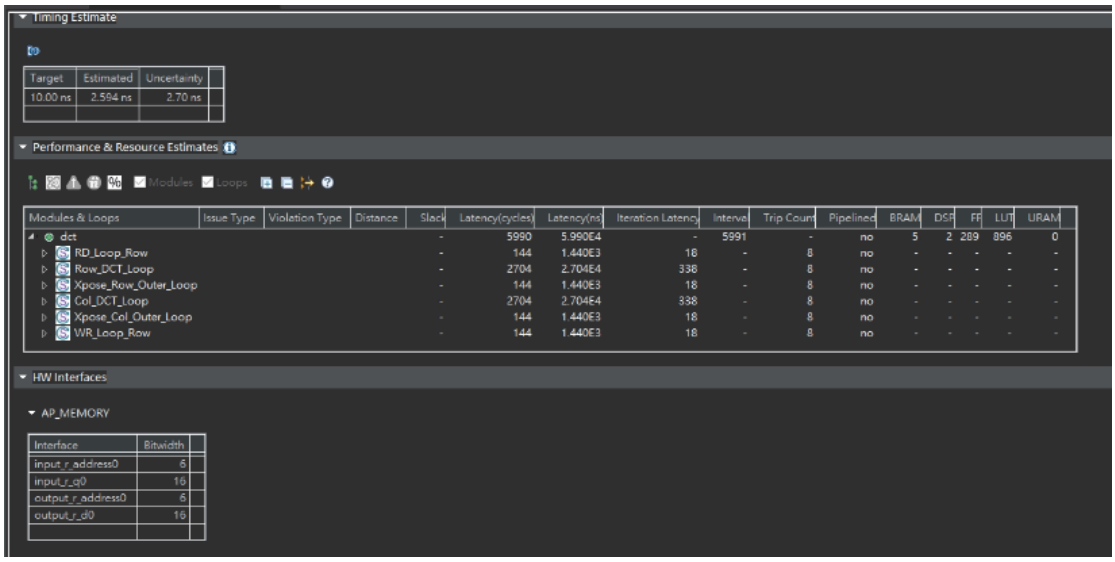
Baseline:

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    | Clock  |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  | 10.00 ns|  4.159 ns|     2.70 ns|
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+---------+
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
    |   min   |   max   |    min   |    max   | min | max |   Type  |
    +---------+---------+----------+----------+-----+-----+---------+
    |      415|      415| 4.150 us | 4.150 us |  416|  416|       no|
    +---------+---------+----------+----------+-----+-----+---------+

    + Detail:
        * Instance:
        +---------------------------------------------------------+------------------------------------------------------+---------+---------+----------+----------+-----+-----+---------+
        |                                                         |                                                      |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
        |                        Instance                         |                        Module                        |   min   |   max   |    min   |    max   | min | max |   Type  |
        +---------------------------------------------------------+------------------------------------------------------+---------+---------+----------+----------+-----+-----+---------+
        |grp_dct_Pipeline_RD_Loop_Row_RD_Loop_Col_fu_70            |dct_Pipeline_RD_Loop_Row_RD_Loop_Col                  |       66|       66| 0.660 us | 0.660 us |   66|   66|       no|
        |grp_dct_Pipeline_Row_DCT_Loop_DCT_Outer_Loop_fu_78        |dct_Pipeline_Row_DCT_Loop_DCT_Outer_Loop              |       70|       70| 0.700 us | 0.700 us |   70|   70|       no|
        |grp_dct_Pipeline_Xpose_Row_Outer_Loop_Xpose_Row_Inner_Loop_fu_100 |dct_Pipeline_Xpose_Row_Outer_Loop_Xpose_Row_Inner_Loop |       66|       66| 0.660 us | 0.660 us |   66|   66|       no|
        |grp_dct_Pipeline_Col_DCT_Loop_DCT_Outer_Loop_fu_106       |dct_Pipeline_Col_DCT_Loop_DCT_Outer_Loop              |       70|       70| 0.700 us | 0.700 us |   70|   70|       no|
        |grp_dct_Pipeline_Xpose_Col_Outer_Loop_Xpose_Col_Inner_Loop_fu_128 |dct_Pipeline_Xpose_Col_Outer_Loop_Xpose_Col_Inner_Loop |       66|       66| 0.660 us | 0.660 us |   66|   66|       no|
        |grp_dct_Pipeline_WR_Loop_Row_WR_Loop_Col_fu_134           |dct_Pipeline_WR_Loop_Row_WR_Loop_Col                  |       66|       66| 0.660 us | 0.660 us |   66|   66|       no|
        +---------------------------------------------------------+------------------------------------------------------+---------+---------+----------+----------+-----+-----+---------+

        * Loop:
        N/A
```



Baseline with “#pragma HLS pipeline off”:





synthesis log to list what steps the tool takes during synthesis:

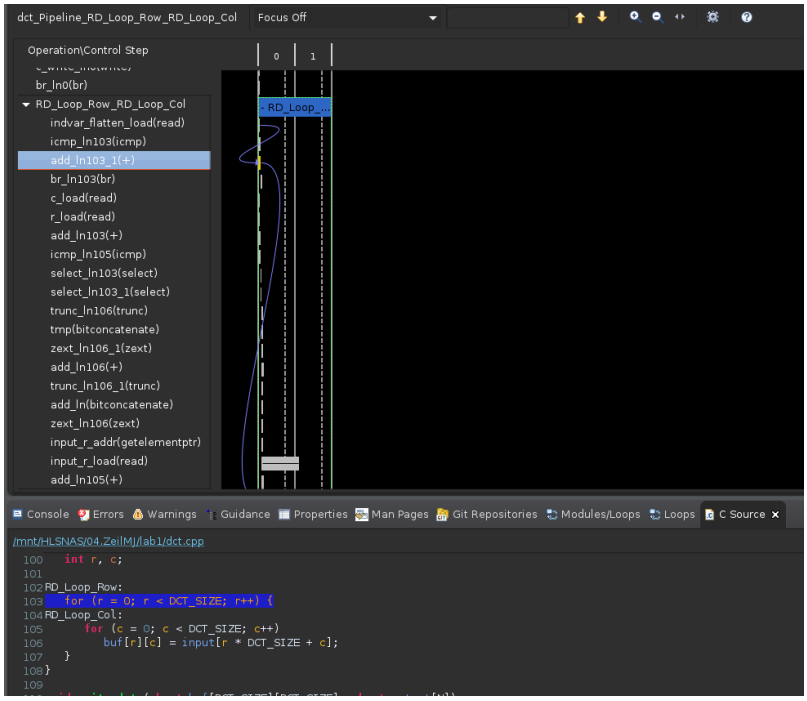
```
INFO: [HLS 200-777] Using interface defaults for 'Vivado' flow target.
INFO: [HLS 214-178] Inlining function 'dct_1d(short*, short*)' into 'dct_2d(short (*) [8], short (*) [8])' (dct.cpp:69:0)
INFO: [HLS 214-178] Inlining function 'read_data(short*, short (*) [8])' into 'dct(short*, short*)' (dct.cpp:123:0)
INFO: [HLS 214-178] Inlining function 'dct_2d(short (*) [8], short (*) [8])' into 'dct(short*, short*)' (dct.cpp:123:0)
INFO: [HLS 214-178] Inlining function 'write_data(short (*) [8], short*)' into 'dct(short*, short*)' (dct.cpp:123:0)
INFO: [HLS 200-111] Finished Compiling Optimization and Transform: CPU user time: 1.98 seconds. CPU system time: 0.16 seconds. Elapsed time: 2.6 seconds; current allocated memory: 212.883 MB.
INFO: [HLS 200-111] Finished Checking Pragmas: CPU user time: 0 seconds. CPU system time: 0 seconds. Elapsed time: 0 seconds; current allocated memory: 212.883 MB.
```

```
Implementing module 'dct_Pipeline_RD_Loop_Row_RD_Loop_Col'
Implementing module 'dct_Pipeline_Row_DCT_Loop_DCT_Outer_Loop'
Implementing module
'dct_Pipeline_Xpose_Row_Outer_Loop_Xpose_Row_Inner_Loop'
Implementing module 'dct_Pipeline_Col_DCT_Loop_DCT_Outer_Loop'
```

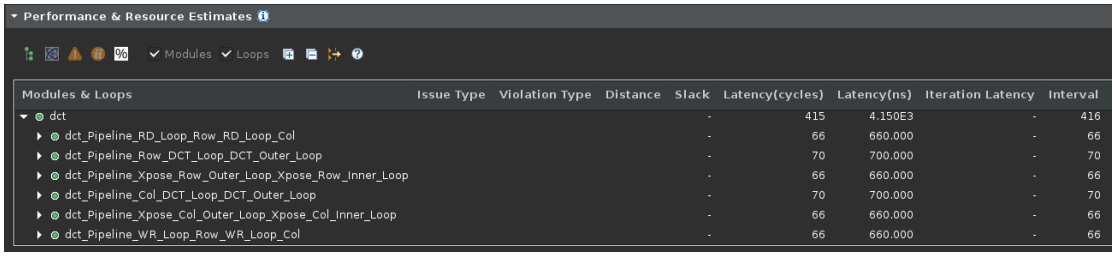
```
Implementing module
'dct_Pipeline_Xpose_Col_Outer_Loop_Xpose_Col_Inner_Loop'
Implementing module 'dct_Pipeline_WR_Loop_Row_WR_Loop_Col'
Implementing module 'dct'
Generating RTL for module 'dct_Pipeline_RD_Loop_Row_RD_Loop_Col'
Generating RTL for module 'dct_Pipeline_Row_DCT_Loop_DCT_Outer_Loop'
```

```
Pipelining result : Target II = NA, Final II = 1, Depth = 2, loop 'RD_Loop_Row_RD_Loop_Col'
Pipelining result : Target II = NA, Final II = 1, Depth = 6, loop 'Row_DCT_Loop_DCT_Outer_Loop'
Pipelining result : Target II = NA, Final II = 1, Depth = 2, loop 'Xpose_Row_Outer_Loop_Xpose_Row_Inner_Loop'
Pipelining result : Target II = NA, Final II = 1, Depth = 6, loop 'Col_DCT_Loop_DCT_Outer_Loop'
Pipelining result : Target II = NA, Final II = 1, Depth = 2, loop 'Xpose_Col_Outer_Loop_Xpose_Col_Inner_Loop'
Pipelining result : Target II = NA, Final II = 1, Depth = 2, loop 'WR_Loop_Row_WR_Loop_Col'
```

Schedule Viewer baseline:



latency, iteration latency



Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval
dct				-	415	4.150E3	-	416
dct_Pipeline_RD_Loop_Row_RD_Loop_Col				-	66	660.000	-	66
dct_Pipeline_Row_DCT_Loop_DCT_Outer_Loop				-	70	700.000	-	70
dct_Pipeline_Xpose_Row_Outer_Loop_Xpose_Row_Inner_Loop				-	66	660.000	-	66
dct_Pipeline_Col_DCT_Loop_DCT_Outer_Loop				-	70	700.000	-	70
dct_Pipeline_Xpose_Col_Outer_Loop_Xpose_Col_Inner_Loop				-	66	660.000	-	66
dct_Pipeline_WR_Loop_Row_WR_Loop_Col				-	66	660.000	-	66

Solution1	Solution2	Solution3	Solution4	Solution5	Solution6
baseline	pipeline_loop	initial_interval	bram	array_partition	dataflow

Performance Estimates

Timing

Clock		solution1	solution2	solution3	solution4	solution5	solution6
ap_clk	Target	10.00 ns	10.00 ns	10.00 ns	10.00 ns	10.00 ns	10.00 ns
	Estimated	4.159 ns	3.133 ns	3.152 ns	4.159 ns	4.159 ns	4.159 ns

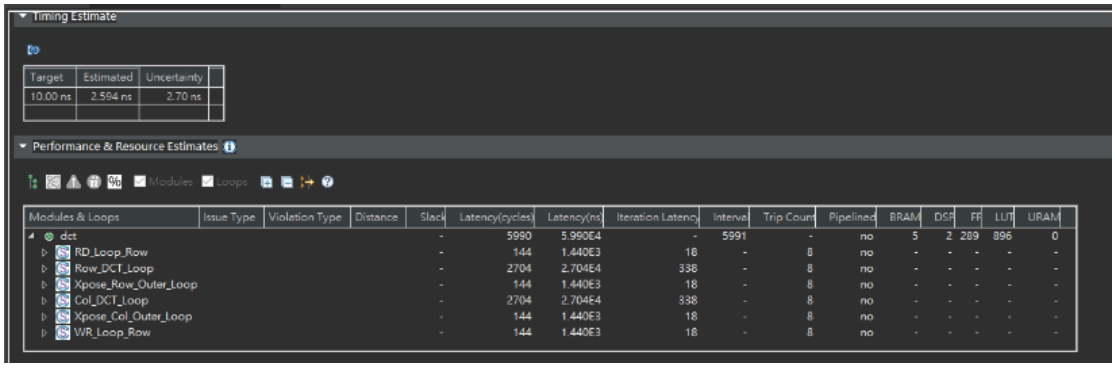
Latency

		solution1	solution2	solution3	solution4	solution5	solution6
Latency (cycles)	min	415	2319	160	608	419	606
	max	415	2319	160	608	419	606
Latency (absolute)	min	4.150 us	23.190 us	1.600 us	6.080 us	4.190 us	6.060 us
	max	4.150 us	23.190 us	1.600 us	6.080 us	4.190 us	6.060 us
Interval (cycles)	min	416	2320	161	609	420	473
	max	416	2320	161	609	420	473

Utilization Estimates

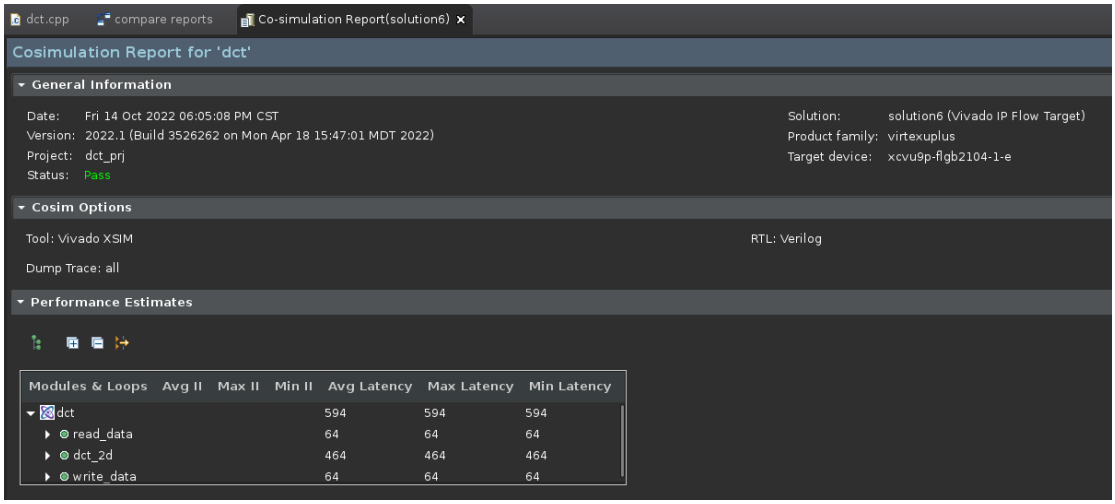
	solution1	solution2	solution3	solution4	solution5	solution6
BRAM_18K	17	5	0	11	9	12
DSP	16	2	112	16	16	16
FF	671	461	5899	875	2211	870
LUT	1992	1467	7226	2093	3088	2025
URAM	0	0	0	0	0	0

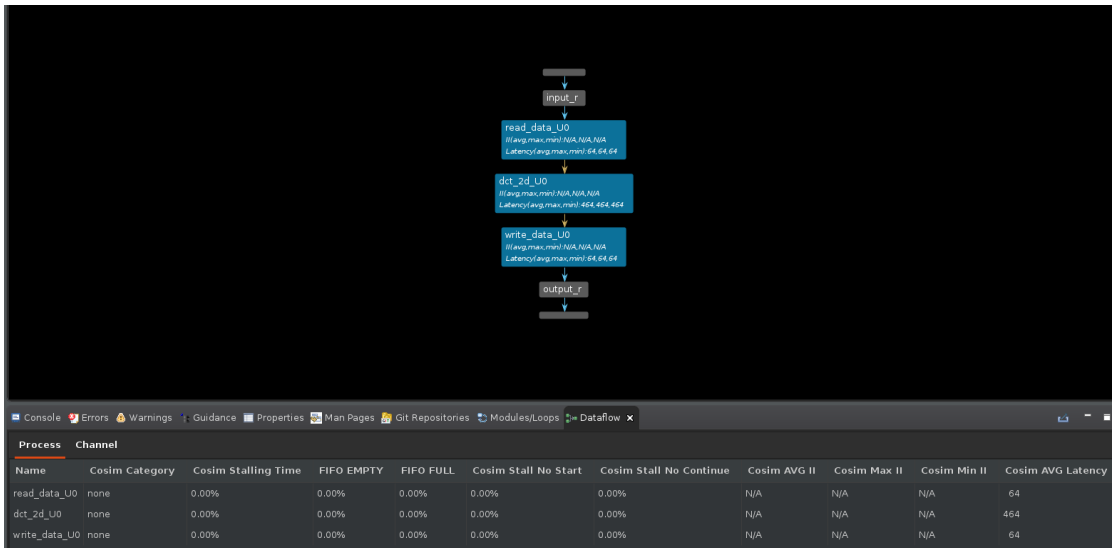
Baseline without pipeline



Dataflow seems not perform as well as we thought, according to cloud tutorial, Dataflow should be directive in sub_loop

Dataflow co_simulation





The biggest problem is that I can’t borrow board since they are rent by others…