MAY 2020

TEXAS
The University of Texas at Austin

Verification of 64 Point FFT/IFFT Processor

EE 382M-11 Verification of Digital Systems

Bagus Hanindhito (bh29293), Tosin Jemilehin (tej474), Ashen Ekanayake (ese396)
The University of Texas at Austin

Agenda

- Introduction
- Formal Verification
- Universal Verification Methodology
- Logical Equivalence Checking
- Conclusion

Introduction

Basic Operation

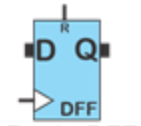
- The circuit decomposes 64-point FFT/IFFT into two-dimensional 8-point FFT/IFFT.
  - $A(s+8t) = \sum_{l=0}^{7}\left[W_{64}^{sl}\sum_{m=0}^{7} B(l+8m)W_8^{sm}\right]W_8^{lt}$
  - W_8^{sm} and W_8^{lt} are 8-point twiddle-factors (complex constant)
  - W_{64}^{sl} is 64-point twiddle-factors (complex constant)
- No true multiplier; Multiplication is implemented as shift-and-add.

Data Format

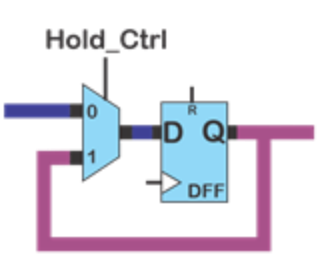
- The circuit uses Q4.12 Signed Fixed-Point format for both real data and imaginary data to form 32-bit complex data.
  - Largest value is 7.9997558593750 (0b0111111111111111)
  - Smallest value is -8.0000000000000 (0b1000000000000000)
  - Resolution is 0.0002441406250 (0b0000000000000001)

Complex Data Set (32-bit)	
31...16 Real Data (16-bit)	15...0 Imaginary Data (16-bit)

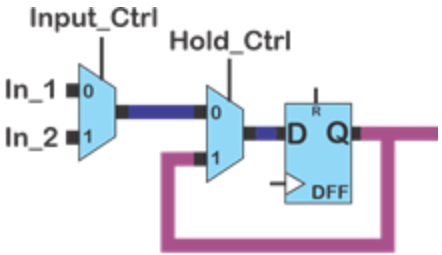
Block Diagram



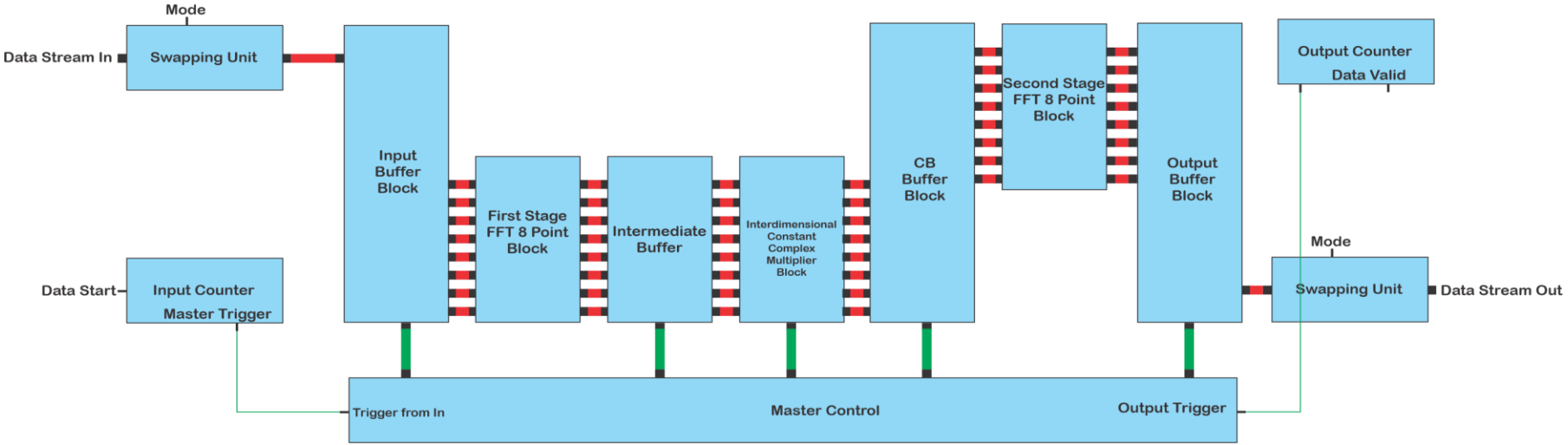
Basic DFF



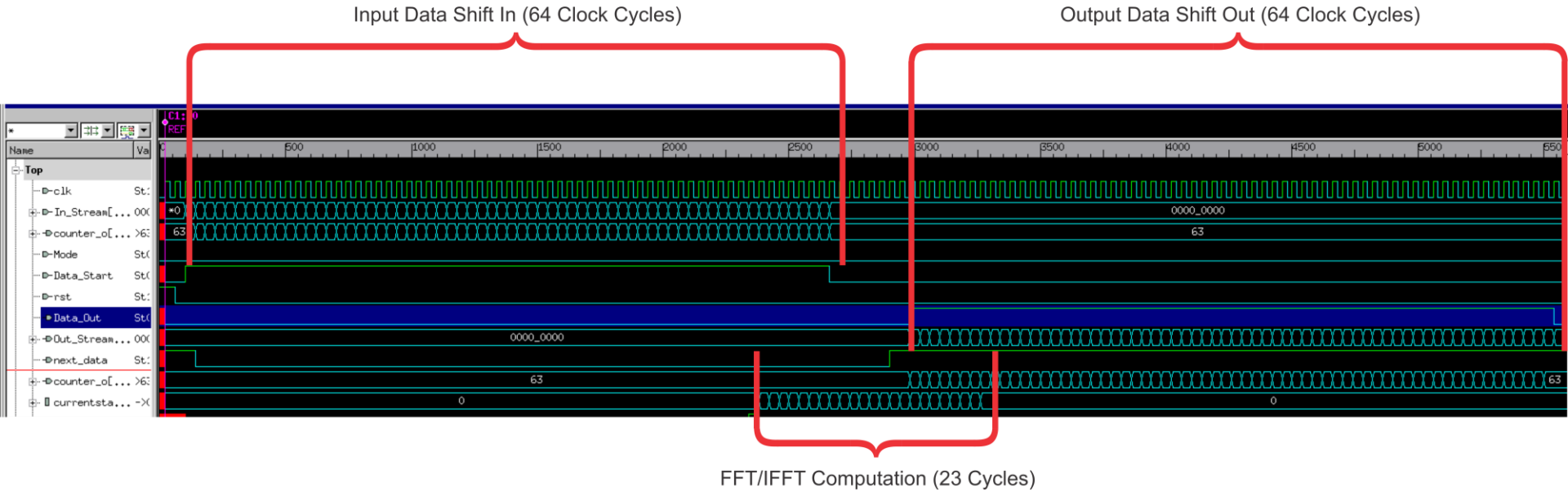
DFF with Hold



DFF with Hold and Input Ctrl



Circuit Behavior



- Single FFT/IFFT operation can be completed in 135 Cycles

Verification Plan

- Formal Verification
  - To verify the Master Control block (RTL-Level)
- Universal Verification Methodology (UVM)
  - To verify full-circuit functionality (RTL-Level)
- Logical Equivalence Checking (LEC)
  - To verify synthesized netlist against RTL model
  - To verify implementation against the generated netlist

Formal Verification

Overview

- Formal Verification using Cadence JasperGold.
- Verification is done for Master Control block.
  - The heart of the circuit.
  - Control the dataflow and operation throughout the circuit.
  - Need to be verified extensively.
- Properties are written based on the specification of the circuits.

Assumption

- Master control is triggered by Input Counter when the 56th input data enters the Input Buffer.
  - Mastertrig signal goes high at least one clock cycle.
- Master control consists of 23 states
  - 23 clock cycles to perform FFT/IFFT computation
  - No interruption is possible after triggered.
  - Full cycle starts at IDLE state, end at IDLE state.
  - Cannot be triggered before the full cycle is ended.

Input Data Set

- Input buffer groups the input data into eight sets.
- Each set enters the 8-point FFT block per clock cycle.

Set	Input Data
0	$B(0), B(8), B(16), B(24), B(32), B(40), B(48), B(56)$
1	$B(1), B(9), B(17), B(25), B(33), B(41), B(49), B(57)$
2	$B(2), B(10), B(18), B(26), B(34), B(42), B(50), B(58)$
3	$B(3), B(11), B(19), B(27), B(35), B(43), B(51), B(59)$
4	$B(4), B(12), B(20), B(28), B(36), B(44), B(52), B(60)$
5	$B(5), B(13), B(21), B(29), B(37), B(45), B(53), B(61)$
6	$B(6), B(14), B(22), B(30), B(38), B(46), B(54), B(62)$
7	$B(7), B(15), B(23), B(31), B(39), B(47), B(55), B(63)$

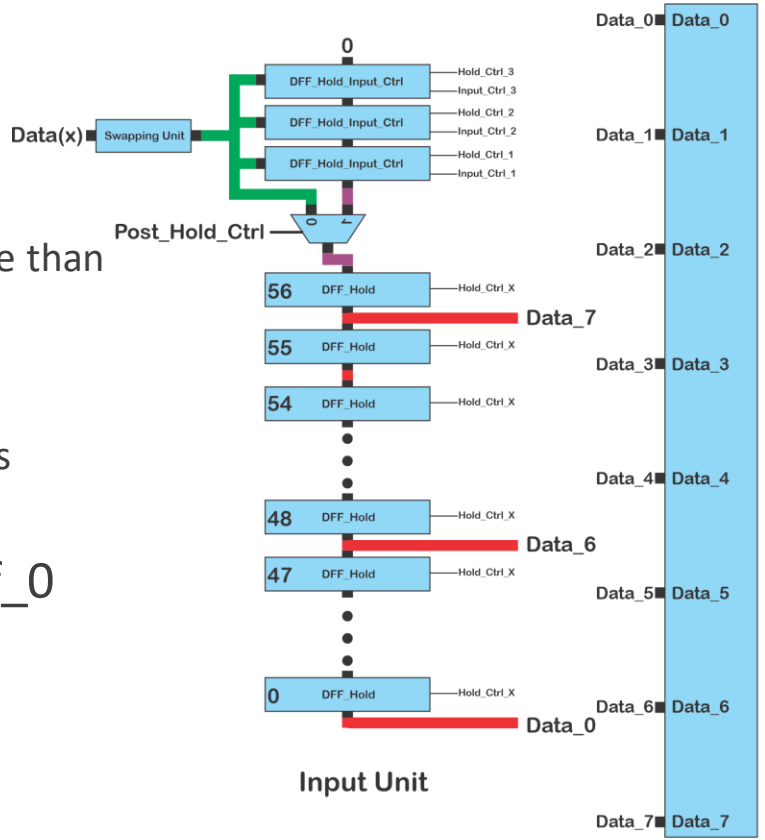
Complex Constant Multiplier Scheduling

- This specification governs how the master control block behaves. (Red = Multi-Cycle Multiplication)

SET	DATA0		DATA1		DATA2		DATA3		DATA4		DATA5		DATA6		DATA7	
	CONST	TYPE	CONST	TYPE	CONST	TYPE	CONST	TYPE	CONST	TYPE	CONST	TYPE	CONST	TYPE	CONST	TYPE
SET 0	BYPASS	1	BYPASS	1	BYPASS	1	BYPASS	1	BYPASS	1	BYPASS	1	BYPASS	1	BYPASS	1
SET 1	BYPASS	1	1	1	2	1	3	1	4	1	5	1	6	1	7	1
SET 2	BYPASS	1	2	1	4	1	6	1	8	1	HOLD	HL	HOLD		HOLD	HL
	HOLD	1	HOLD	HL	HOLD	HL	HOLD	HL	HOLD	HL	6	5	4	5	2	5
SET 3	BYPASS	1	3	1	6	1	7	5	4	5	1	5	2	7	5	7
SET 4	BYPASS	1	4	1	8	1	HOLD	HL	BYPASS	5	HOLD	HL	HOLD	HL	HOLD	HL
	HOLD	HL	HOLD	HL	HOLD	HL	4	5	HOLD	HL	HOLD	HL	HOLD	HL	HOLD	HL
	HOLD	HL	HOLD	HL	HOLD	HL	HOLD	HL	HOLD	HL	4	7	8	7	HOLD	HL
	HOLD	HL	HOLD	HL	HOLD	HL	HOLD	HL	HOLD	HL	HOLD	HL	HOLD	HL	4	3
SET 5	BYPASS	1	5	1	6	5	1	5	4	7	7	3	2	3	3	2
SET 6	BYPASS	1	6	1	4	5	2	7	8	7	HOLD	HL	HOLD	HL	HOLD	HL
	HOLD	HL	HOLD	HL	HOLD	HL	HOLD	HL	HOLD	HL	2	3	4	2	6	6
SET 7	BYPASS	1	7	1	2	5	5	7	4	3	3	2	6	6	1	4

Input Buffer Control Signal

- hold_all_in
  - Stall input circuit when multiplier needs more than 1 cycle
- hold_buf_2, hold_buf_1, hold_buf_0
  - Stall additional buffers when multiplier needs more than 1 cycle.
- in_ctrl_buf_2, in_ctrl_buf_1, in_ctrl_buf_0
  - Select the input of additional buffers.
- pos_hold_ctrl
  - Select the input of main buffer segment.



Input Buffer Control Signal Assertion

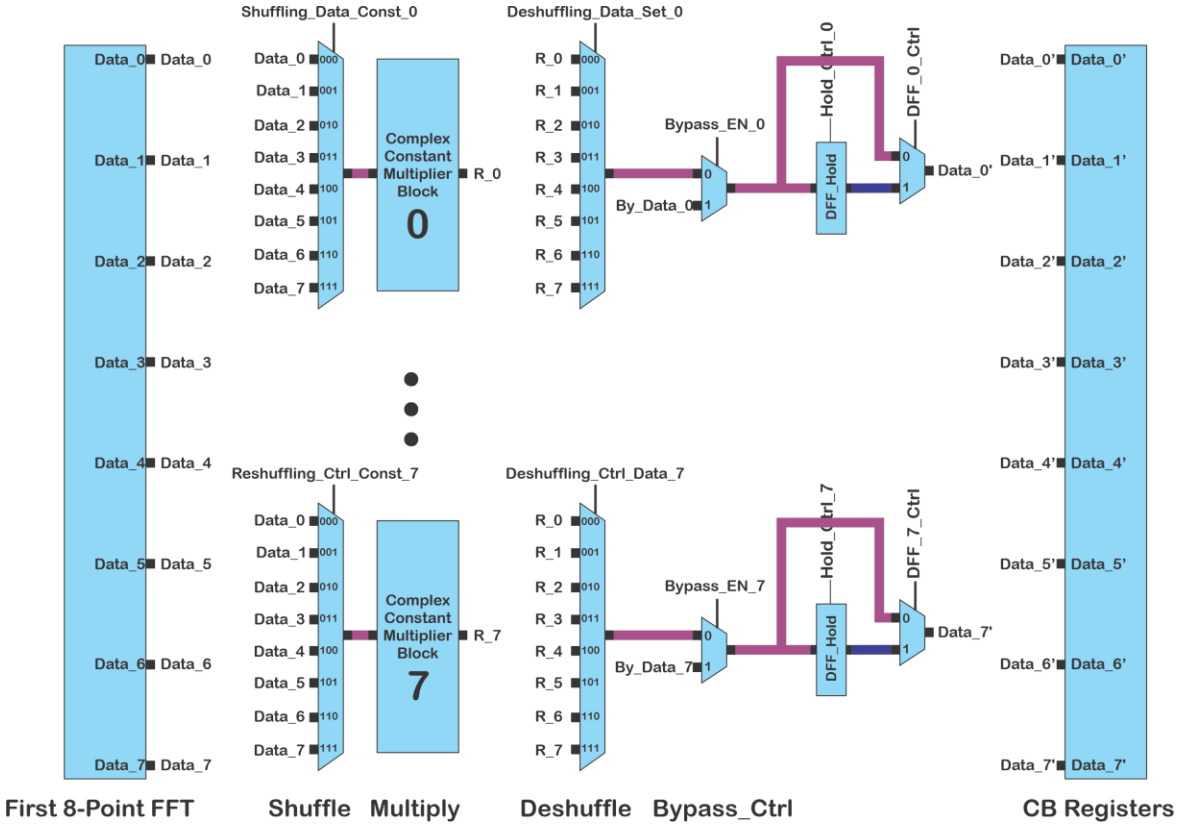
✔	Assert	master_control.v_master_control_inst.assert_hold_all_in	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_all_in:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_hold_buf_2	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_buf_2:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_hold_buf_1	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_buf_1:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_hold_buf_0	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_buf_0:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_in_ctrl_buf_2	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_in_ctrl_buf_2:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_in_ctrl_buf_1	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_in_ctrl_buf_1:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_in_ctrl_buf_0	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_in_ctrl_buf_0:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_pos_hold_ctrl	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_pos_hold_ctrl:precondition1	B	1	0.0	<embedded>	Analysis Session

Complex Constant Multiplier Unit

- Eight complex constant multiplier blocks, one for each type.

Type	Input Operand	Pre-Processing	Constant Operand	Result	Post Processing
1	$A + jB$	NONE $A + jB$	$C + jD$	$AC - BD + j(AD + BC)$	NONE $AC - BD + j(AD + BC)$
2	$A + jB$	IM_INVERT $A - jB$	$C + jD$	$AC + BD + j(AD - BC)$	IM_INVERT $AC + BD + j(-AD + BC)$
3	$A + jB$	RE_INVERT $-A + jB$	$C + jD$	$-AC - BD + j(-AD + BC)$	IM_INVERT $-AC - BD + j(AD - BC)$
4	$A + jB$	RE_INVERT IM_INVERT $-A - jB$	$C + jD$	$-AC + BD + j(-AD - BC)$	NONE $-AC + BD + j(-AD - BC)$
5	$A + jB$	RE-IM SWAP $B + jA$	$C + jD$	$-AD + BC + j(AC + BD)$	RE_INVERT $AD - BC + j(AC + BD)$
6	$A + jB$	RE-IM SWAP IM_INVERT $B - jA$	$C + jD$	$AD + BC + j(-AC + BD)$	NONE $AD + BC + j(-AC + BD)$
7	$A + jB$	RE-IM SWAP RE_INVERT $-B + jA$	$C + jD$	$-AD - BC + j(AC - BD)$	NONE $-AD - BC + j(AC - BD)$
8	$A + jB$	RE-IM SWAP RE_INVERT IM_INVERT $-B - jA$	$C + jD$	$AD - BC + j(-AC - BD)$	RE_INVERT $-AD + BC + j(-AC - BD)$

Complex Constant Multiplier Unit (cont'd)



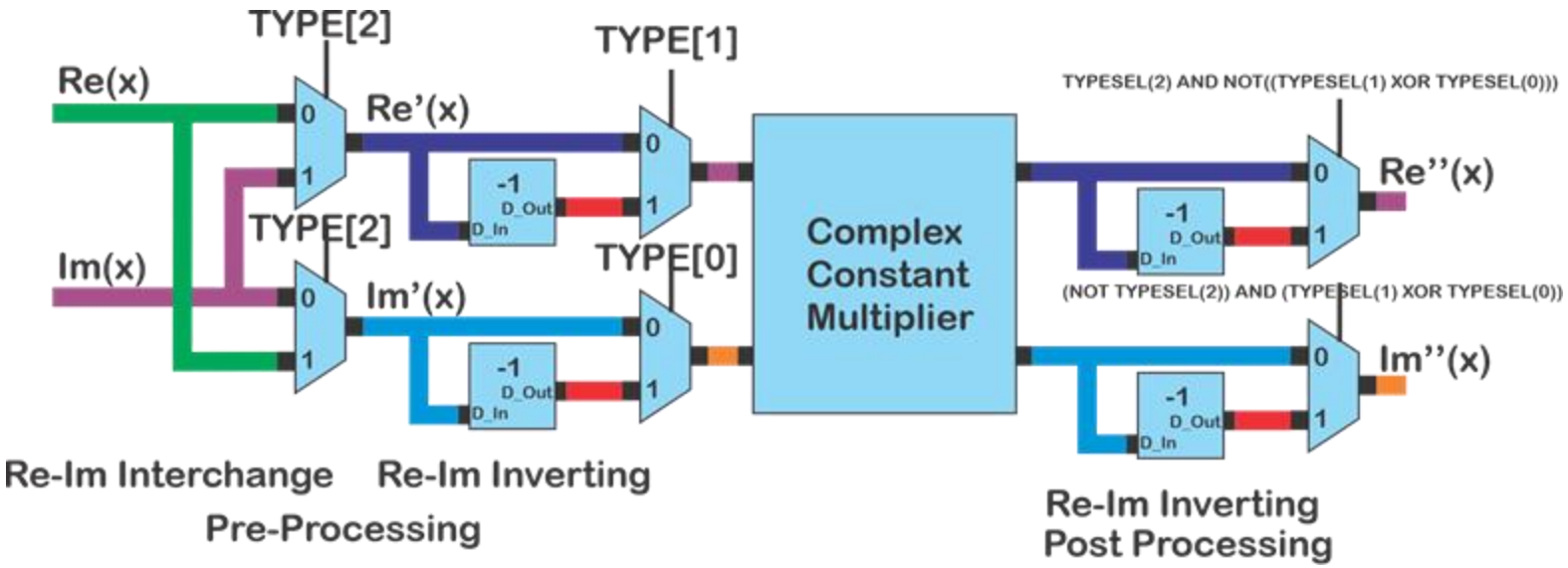
Complex Const. Mult.: Shuffle Unit

- Shuf_Ctrl_0 through Shuf_Ctrl_7
  - Select which set of data goes which constant multiplier.

✔	Assert	master_control.v_master_control_inst.assert_Shuf_Ctrl_0	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Shuf_Ctrl_0:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Shuf_Ctrl_1	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Shuf_Ctrl_1:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Shuf_Ctrl_2	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Shuf_Ctrl_2:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Shuf_Ctrl_3	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Shuf_Ctrl_3:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Shuf_Ctrl_4	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Shuf_Ctrl_4:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Shuf_Ctrl_5	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Shuf_Ctrl_5:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Shuf_Ctrl_6	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Shuf_Ctrl_6:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Shuf_Ctrl_7	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Shuf_Ctrl_7:precondition1	B	1	0.0	<embedded>	Analysis Session

Complex Const. Mult.: Multiplier Unit

- Type_Sel_0 through Type_Sel_7
  - Select the type of operation that each multiplier does.

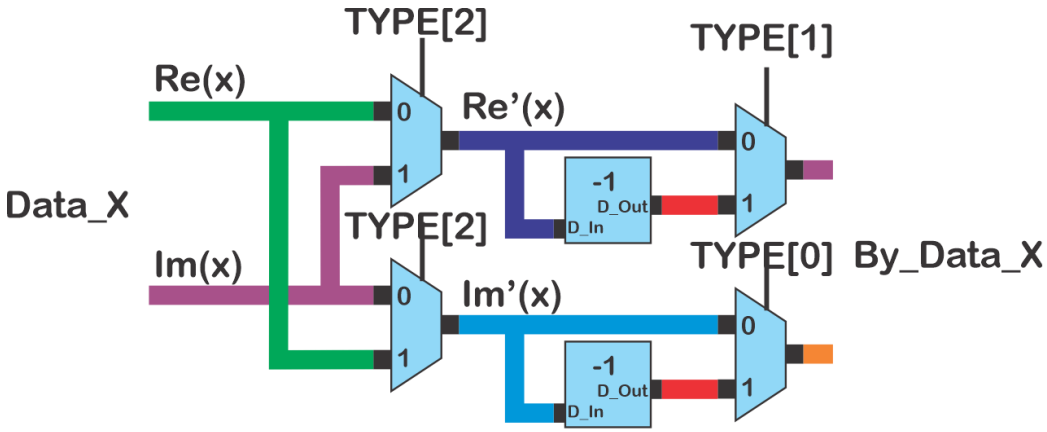


Complex Const. Mult.: Multiplier Unit Assertion

✔	Assert	master_control.v_master_control_inst.assert_Type_Sel_0	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Type_Sel_0:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Type_Sel_1	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Type_Sel_1:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Type_Sel_2	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Type_Sel_2:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Type_Sel_3	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Type_Sel_3:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Type_Sel_4	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Type_Sel_4:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Type_Sel_5	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Type_Sel_5:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Type_Sel_6	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Type_Sel_6:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Type_Sel_7	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Type_Sel_7:precondition1	B	1	0.0	<embedded>	Analysis Session

Complex Const. Mult.: Bypass Unit

- Bypass_Sel_0 through Bypass_Sel_7
  - Select whether a multiplication is performed or not for each multiplication block. Bypass means no multiplication (only swapping or sign inverting the real and imaginary parts).



Bypass Circuit

Complex Const. Mult.: Bypass Unit Assertion

✔	Assert	master_control.v_master_control_inst.assert_Bypass_Sel_0	PRE	Infinite	0.0	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_Sel_0:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_Sel_1	PRE	Infinite	0.0	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_Sel_1:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_Sel_2	PRE	Infinite	0.0	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_Sel_2:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_Sel_3	PRE	Infinite	0.0	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_Sel_3:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_Sel_4	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_Sel_4:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_Sel_5	PRE	Infinite	0.0	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_Sel_5:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_Sel_6	PRE	Infinite	0.0	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_Sel_6:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_Sel_7	PRE	Infinite	0.0	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_Sel_7:precondition1	B	1	0.0	<embedded>	Analysis Session

Complex Const. Mult.: Deshuffle Unit

- DeShuf_Ctrl_0 through DeShuf_Ctrl_7
  - Select return path of the dataset after being multiplied.

✔	Assert	master_control.v_master_control_inst.assert_DeShuf_Ctrl_0	PRE	Infinite	0.0	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DeShuf_Ctrl_0:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DeShuf_Ctrl_1	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DeShuf_Ctrl_1:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DeShuf_Ctrl_2	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DeShuf_Ctrl_2:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DeShuf_Ctrl_3	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DeShuf_Ctrl_3:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DeShuf_Ctrl_4	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DeShuf_Ctrl_4:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DeShuf_Ctrl_5	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DeShuf_Ctrl_5:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DeShuf_Ctrl_6	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DeShuf_Ctrl_6:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DeShuf_Ctrl_7	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DeShuf_Ctrl_7:precondition1	B	1	0.0	<embedded>	Analysis Session

Complex Const. Mult.: Datapath Unit

- Bypass_EN_0 through Bypass_EN_7
  - Select whether to use dataset from multiplier unit or from bypass unit.
- Hold_Ctrl_0 through Hold_Ctrl_7
  - Select whether to stall the datapath due to multicycle multiplication.
- DFF_Ctrl_0 through DFF_Ctrl_7
  - Select whether to use next data or delayed data.

Complex Const. Mult.: Datapath Unit (Assertion)

✔	Assert	master_control.v_master_control_inst.assert_Bypass_EN_0	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_EN_0:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_EN_1	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_EN_1:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_EN_2	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_EN_2:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_EN_3	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_EN_3:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_EN_4	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_EN_4:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_EN_5	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_EN_5:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_EN_6	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_EN_6:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Bypass_EN_7	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Bypass_EN_7:precondition1	B	1	0.0	<embedded>	Analysis Session

Complex Const. Mult.: Datapath Unit (Assertion)

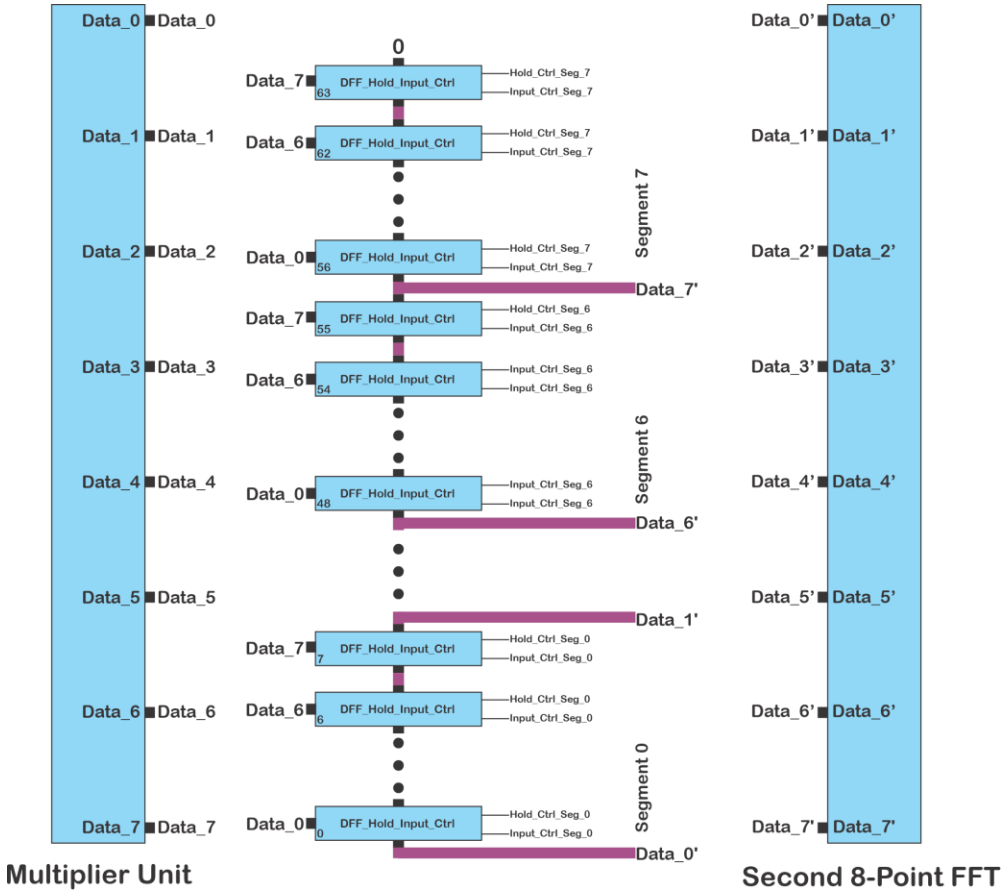
✔	Assert	master_control.v_master_control_inst.assert_Hold_Ctrl_0	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Hold_Ctrl_0:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Hold_Ctrl_1	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Hold_Ctrl_1:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Hold_Ctrl_2	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Hold_Ctrl_2:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Hold_Ctrl_3	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Hold_Ctrl_3:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Hold_Ctrl_4	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Hold_Ctrl_4:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Hold_Ctrl_5	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Hold_Ctrl_5:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Hold_Ctrl_6	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Hold_Ctrl_6:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_Hold_Ctrl_7	PRE	Infinite	0.0	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_Hold_Ctrl_7:precondition1	B	1	0.0	<embedded>	Analysis Session

Complex Const. Mult.: Datapath Unit (Assertion)

✔	Assert	master_control.v_master_control_inst.assert_DFF_Ctrl_0	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DFF_Ctrl_0:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DFF_Ctrl_1	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DFF_Ctrl_1:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DFF_Ctrl_2	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DFF_Ctrl_2:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DFF_Ctrl_3	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DFF_Ctrl_3:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DFF_Ctrl_4	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DFF_Ctrl_4:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DFF_Ctrl_5	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DFF_Ctrl_5:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DFF_Ctrl_6	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DFF_Ctrl_6:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_DFF_Ctrl_7	PRE	Infinite	0.0	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_DFF_Ctrl_7:precondition1	B	1	0.0	<embedded>	Analysis Session

Compute Buffer (CB)

- Rearrange Dataset for second stage of 8-point FFT.
- hold_seg_0 through hold_seg_7
  - To stall each segment of the buffer
- input_ctrl_all_cb
  - To select which data to buffer.

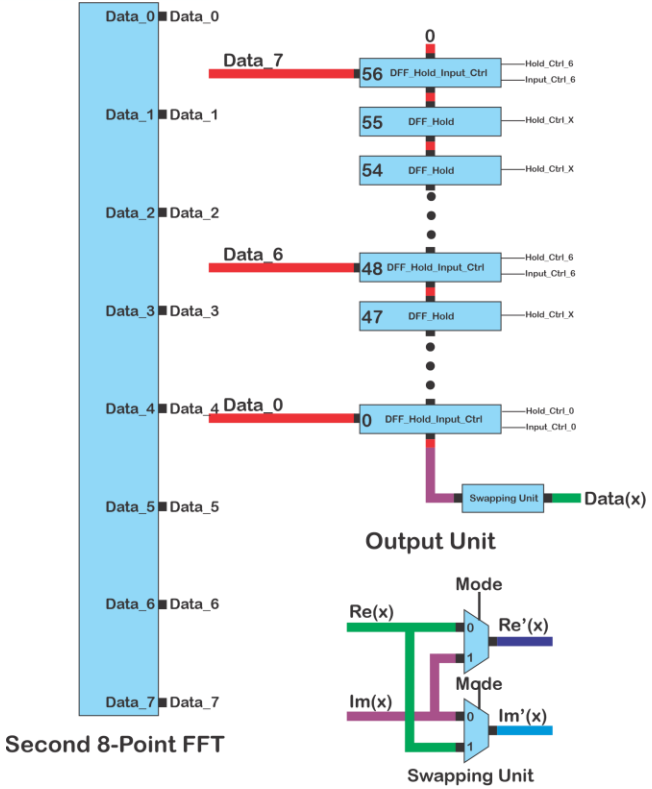


Compute Buffer (CB) (Assertion)

✔	Assert	master_control.v_master_control_inst.assert_hold_seg_0	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_seg_0:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_hold_seg_1	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_seg_1:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_hold_seg_2	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_seg_2:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_hold_seg_3	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_seg_3:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_hold_seg_4	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_seg_4:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_hold_seg_5	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_seg_5:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_hold_seg_6	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_seg_6:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_hold_seg_7	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_hold_seg_7:precondition1	B	1	0.0	<embedded>	Analysis Session
✔	Assert	master_control.v_master_control_inst.assert_in_ctrl_all_cb	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✔	Cover (related)	master_control.v_master_control_inst.assert_in_ctrl_all_cb:precondition1	B	1	0.0	<embedded>	Analysis Session

Output Buffer

- Serialize the Output Data
- counter_en
  - to indicate valid output data
- next_data (currently unused)
  - to indicate the next operation can be started



✓	Assert	master_control.v_master_control_inst.assert_counter_en	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✓	Cover (related)	master_control.v_master_control_inst.assert_counter_en:precondition1	B	1	0.0	<embedded>	Analysis Session
✓	Assert	master_control.v_master_control_inst.assert_next_data	Hp (26)	Infinite	45.5	<embedded>	Analysis Session
✓	Cover (related)	master_control.v_master_control_inst.assert_next_data:precondition1	B	1	0.0	<embedded>	Analysis Session

Universal Verification Methodology (UVM)

UVM Verification Steps

- Step 1: Understanding the Interface

```
interface dut_in;
    logic           clk, rst;
    logic           Mode;
    logic           In_Valid;
    logic   [31:0]  In_Stream;
endinterface: dut_in
```

```
interface dut_out;
    logic           clk;
    logic   [31:0] Out_Stream;
    logic          Out_Valid;
    logic          next_data;
endinterface: dut_out
```

- Step 2: Create a dummy (DUT) module to setup the UVM environment

  Mimic the basic behavior without worrying about the exact functionality

  (Using the same input output interfaces)

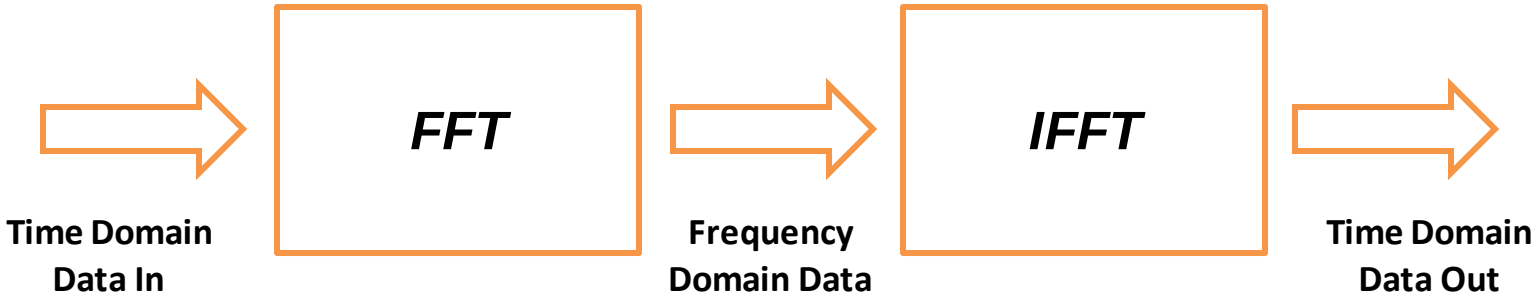
UVM Verification Steps (cont'd)

- Step 3: Modify the Scoreboard

```
task run();
    forever begin
        fifo_in.get(tx_in);
        fifo_out.get(tx_out);
        if(tx_in.In_Valid && (tx_in.rst == 0) && (index1 < 64))begin
            in_data[index1] =  tx_in.In_Stream;
            index1 = index1 + 1;
            if (index1==64) begin
                generate_results();
            end
        end
        if(tx_out.Out_Valid)begin
            out_data[index2] =  tx_out.Out_Stream;
            index2 = index2 + 1;
            if (index2==64) begin
                compare();
            end
        end
    end
endtask: run
```

- Step 4: Write a Perfect FFT
  - ➢ Difficult
    - – 16-bit Fixed Point
    - – No SV Support
  - ➢ Solution
    - – Connect FFT and IFFT together as the DUT
  - ➢ Alternative Solution
    - – DPI-C

UVM Verification Steps (cont'd)



Theoretically,

Time Domain Data In = Time Domain Data Out

However there might be small errors due to rounding off in fixed point arithmetic

UVM Verification Steps (cont'd)

- Step 5: Generating the sequences

  - Reset sequences (created a new sequence class)
  - Usual input data sequence with necessary constraints

(- 0.10009765625 < Data < 0.10009765625)

```
constraint c_In_Stream_imag { (In_Stream[15 :0] >= 16'hFE66) || (In_Stream[15: 0] <= 16'h019A);}
constraint c_In_Stream_real { (In_Stream[31:16] >= 16'hFE66) || (In_Stream[31:16] <= 16'h019A);}
```

  - Generate enough sequences to get an output

UVM Verification Steps (cont'd)

- Step 6: Simulating Using "ModelSim"

```
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 00c8fe8a, FFT Out: 00c8fa8a
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 0140ff1f, FFT Out: 0140fb1e
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 0115fe97, FFT Out: 0115fa96
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 0011ffaa, FFT Out: 0011fbaa
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: ff6fffde, FFT Out: fb6ffbde
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 0104fe81, FFT Out: 0103fa80
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 0076000b, FFT Out: 0075000b
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: ffee0061, FFT Out: fbee0060
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 0198fffd, FFT Out: 0198fbfc
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: ff4e00cf, FFT Out: fb4e00ce
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: feb1fefb, FFT Out: fab1fafb
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: ff0fff2b, FFT Out: fb0ffb2a
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 00f7fe76, FFT Out: 00f7fa76
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: ff990170, FFT Out: fb99016f
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 00b3ffe9, FFT Out: 00b3fbe8
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 00ea007c, FFT Out: 00ea007c
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: ff92fe8c, FFT Out: fb92fa8c
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: fea1febe, FFT Out: faa1fabd
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 00bb0036, FFT Out: 00ba0035
# UVM_INFO ../tb/scoreboard.sv(72) ... [fft_scoreboard] Gold Out: 0143ff5e, FFT Out: 0143fb5e
```

Logical Equivalence Checking

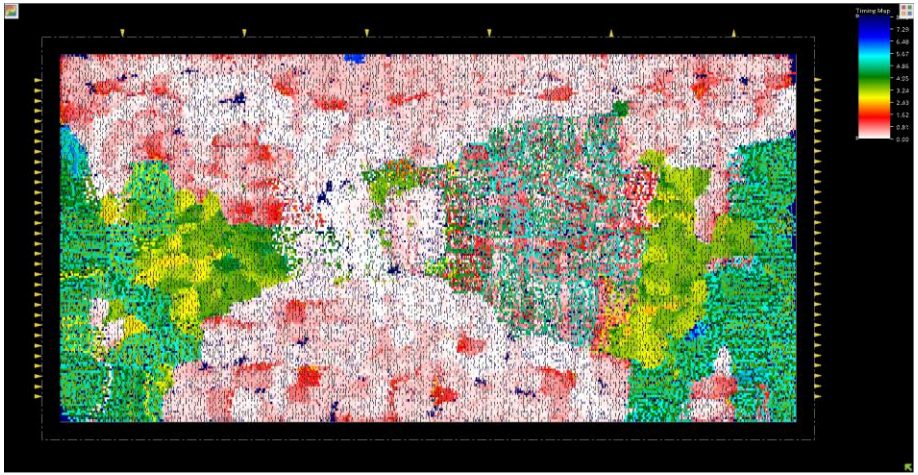
Logical Equivalence Checking

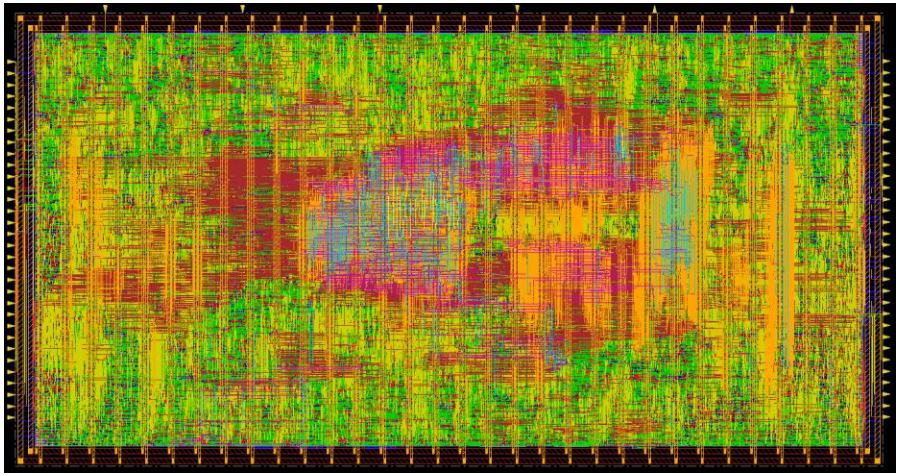
Motivation

- To ensure logical equivalence doesn't break along chip design process
- To ensure design intent is retained as design move into real life implementation

Stages considered

- RTL to Synthesized Netlist (using Synopsys DC)
- Synthesized Netlist to Implementation (using Cadence Innovus)





LEC Workflow

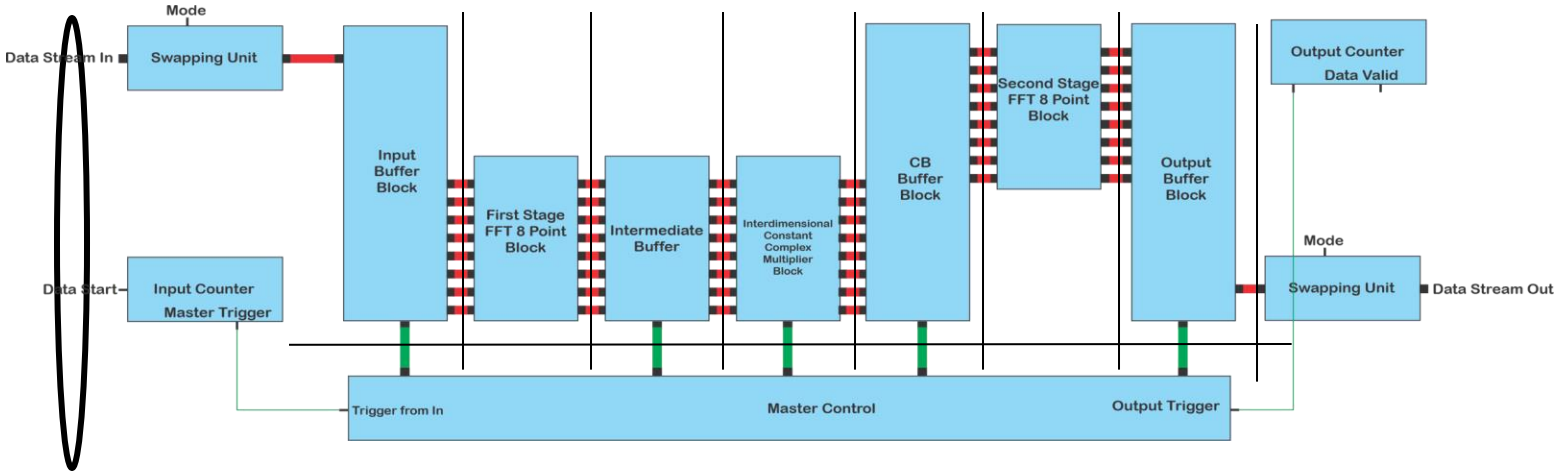
- Setup
  - Identifying keypoints/cuts
- Mapping
  - Mapping keypoints
- Comparison

First stage

- RTL ---> Synthesized Netlist

RTL to Synthesized Netlist

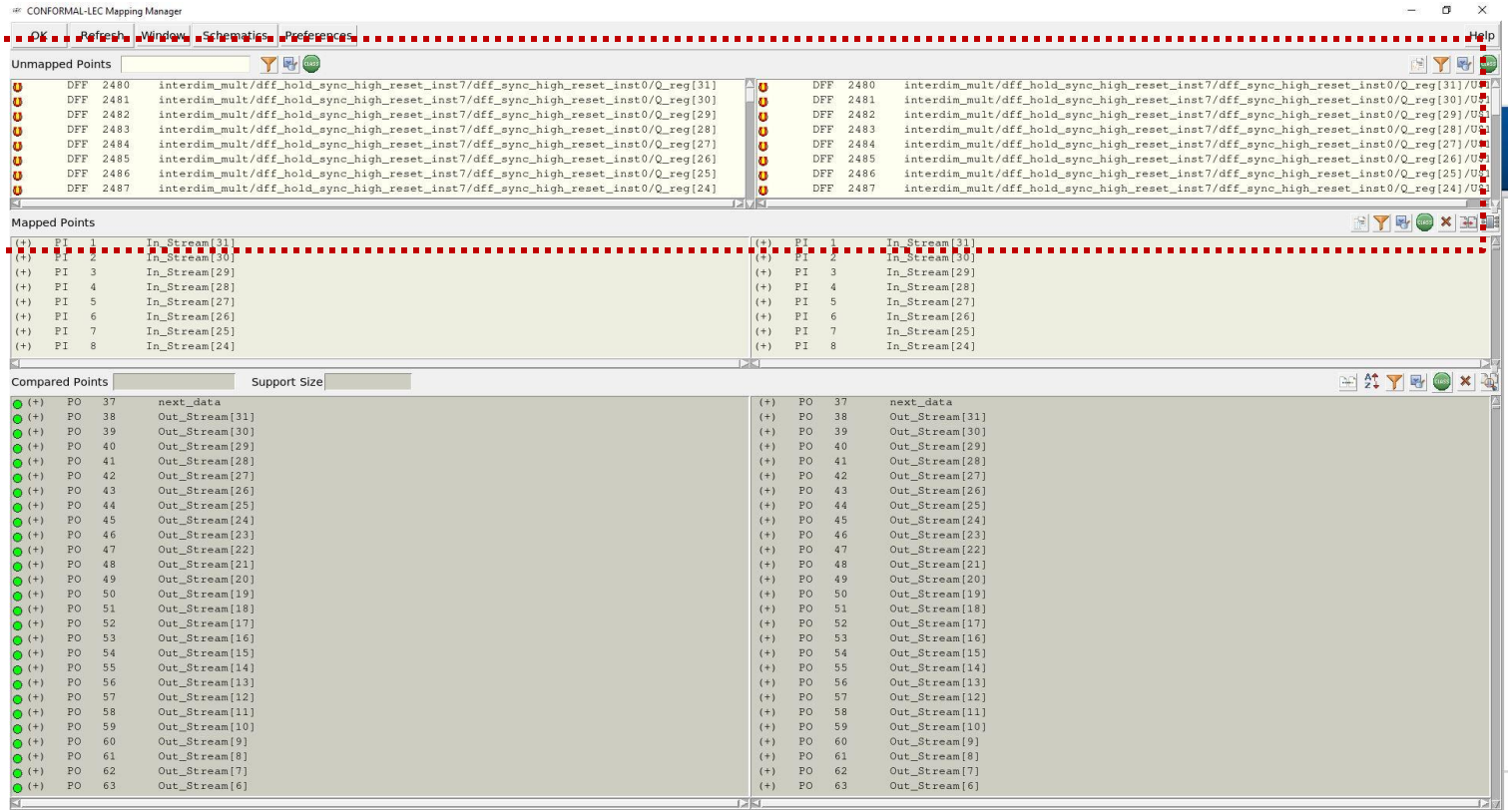
- Setup
  - Golden -> RTL
  - Revised -> Generated Netlist
- Assumption
  - Sequential constant set to 0
  - Sequential merge to reduce unreachable states



RTL to Synthesized Netlist(Mapping)

```
// Golden  key points = 6538
// Revised key points = 6508
// Mapping key points ...
================================================================================
Mapped points: SYSTEM class
--------------------------------------------------------------------------------
Mapped points     PI     PO     DFF       Total
--------------------------------------------------------------------------------
Golden            36     34     6406      6476
--------------------------------------------------------------------------------
Revised           36     34     6406      6476
================================================================================
Unmapped points:
================================================================================
Golden:
--------------------------------------------------------------------------------
Unmapped points   DFF       Total
--------------------------------------------------------------------------------
Unreachable       62        62
================================================================================
Revised:
--------------------------------------------------------------------------------
Unmapped points   DFF       Total
--------------------------------------------------------------------------------
Unreachable       32        32
================================================================================
```

RTL to Synthesized Netlist(Mapping)



Mapping manager

RTL to Synthesized Netlist(Comparison)

```
--------------------------------------------------------------------------------
Unmapped points    DFF        Total
--------------------------------------------------------------------------------
Unreachable        32         32
================================================================================
// Running automatic setup...
// Automatic setup finished.
0
// Command: add_compared_points -all
// 6440 compared points added to compare list
0
// Command: compare
================================================================================
Compared points       PO      DFF        Total
--------------------------------------------------------------------------------
Equivalent            34      6406       6440
================================================================================
0
// Command: puts "Num of compare points = [get_compare_points -count]"
Num of compare points = 6440
// Command: puts "Num of diff points    = [get_compare_points -NONequivalent -count]"
Num of diff points    = 0
// Command: puts "Num of abort points   = [get_compare_points -abort -count]"
Num of abort points   = 0
// Command: puts "Num of unknown points = [get_compare_points -unknown -count]"
Num of unknown points = 0
```

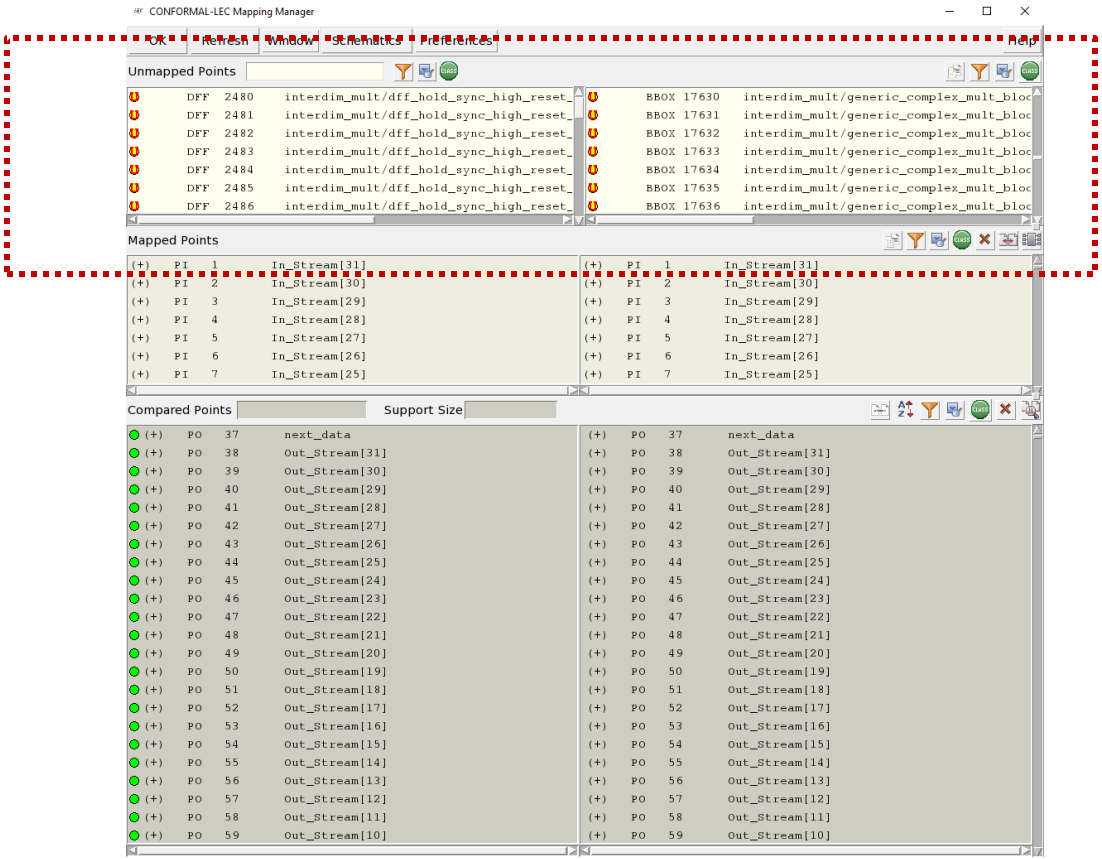
Netlist to Implementation Stage

- **Setup**
  - Golden -> Verified Netlist
  - Revised -> Implementation Design
- **Assumption**
  - Sequential constant set to 0
  - Sequential merge to reduce unreachable states
  - Black boxes were defined due to insufficient library definitions

```
================================================================================
Mapped points: SYSTEM class
--------------------------------------------------------------------------------
Mapped points      PI      PO      DFF        Total
--------------------------------------------------------------------------------
Golden             36      34      6406       6476
--------------------------------------------------------------------------------
Revised            36      34      6406       6476
================================================================================
Unmapped points:
================================================================================
Golden:
--------------------------------------------------------------------------------
Unmapped points    DFF         Total
--------------------------------------------------------------------------------
Unreachable        32          32
================================================================================
```

Mapping

Netlist to Implementation(Mapping)



RTL to Synthesized Netlist(Comparison)

```
================================================================================
// Running automatic setup...
// Automatic setup finished.
0
// Command: add_compared_points -all
// 6440 compared points added to compare list
0
// Command: compare
================================================================================
Compared points       PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent            34     6406      6440
================================================================================
0
// Command: puts "Num of compare points = [get_compare_points -count]"
Num of compare points = 6440
// Command: puts "Num of diff points    = [get_compare_points -NONequivalent -co
unt]"
Num of diff points    = 0
// Command: puts "Num of abort points   = [get_compare_points -abort -count]"
Num of abort points   = 0
// Command: puts "Num of unknown points = [get_compare_points -unknown -count]"
Num of unknown points = 0
```

Conclusion

Conclusion

- We were able to complete a comprehensive verification on the FFT/IFFT Processor
- Formal Verification was carried out on the “Master Control” unit
- RTL design was tested using UVM
- Gate level and implementation level netlists were verified using Logical Equivalence Checking
- Our project is available on GitHub to anyone for further experiments

https://github.com/ashanuka/FFT-IFFT-Verification

Reference

- Maharatna, Koushik, Grass, Eckhard and Jagdhold, Ulrich (2004) A 64-point Fourier transform chip for high-speed wireless LAN application using OFDM. IEEE Journal of Solid-State Circuit, 39 (3), 484-493.