An early physical synthesis prototyping flow with RTL Architect

Marc Galceran & Alfredo Taddei
Marvell Technology

Outline

- Introduction and Motivation
- Methodology and Flow
- Results on Orion Coherent DSP Marvell Chip
- Conclusions

Introduction

- Target: Set up a system to obtain
  - Fast physical synthesis results, review both area and timing
  - Fast and accurate vector-based RTL power estimations
  - Enable engineers to understand power-impact of incremental RTL modification
  - Enable managers to forecast area and block consumption for project planning purposes

- New flows we built for this effort
  - RTL-Architect (RTLA) for fast synthesis and Clock tree synthesis
  - PrimePower-RTL for power analysis using RTL vectors
  - Comparison point: Fusion Compiler Physical Design flow (FC flow) + PrimePower vector-based power analysis

- Joint effort between CAD, Design team, PD team and Synopsys AEs.
  - Acknowledgements: Marcos Macchi, Alex Vidal

Flow definition

Methodology

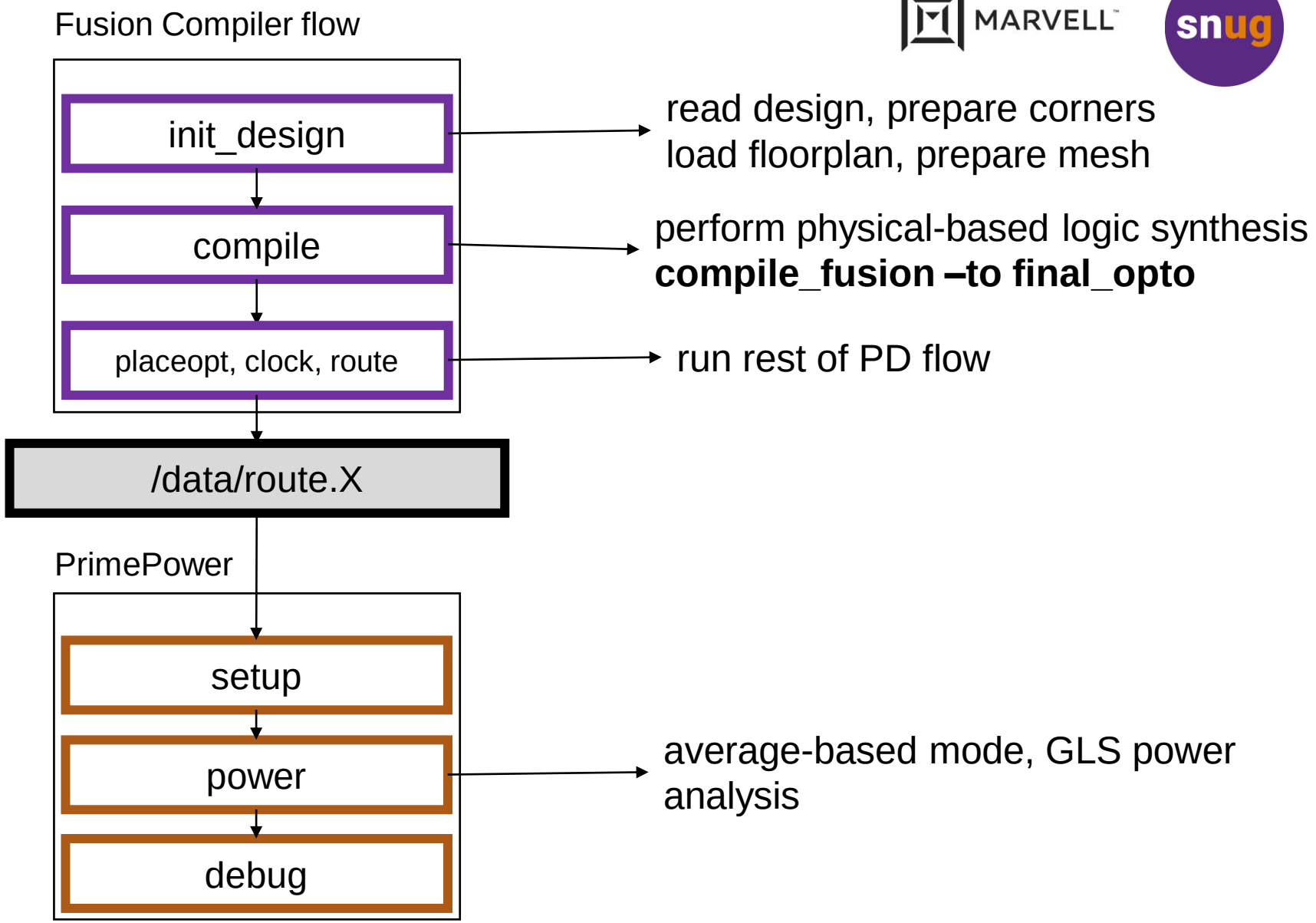
- Flow input is an RTL release directory
  - Filelist
  - SDC
  - Floorplan
  - FSDB

- RTL-Architect (RTLA) is integrated with pre-existing Fusion Compiler (FC) flow
  - Use same scripts as FC flow for maximum correlation between fast RTLA synthesis and FC synthesis
  - Target: RTLA synthesis QoR within 10% wrt. FC synthesis in various tested blocks (after iterating with **Synopsys**), while runtime down by 33-66%
  - Output of RTLA flow is the release of netlist/power data for PP-RTL
  - Includes timing and area results that were correlated to full FC results

Methodology

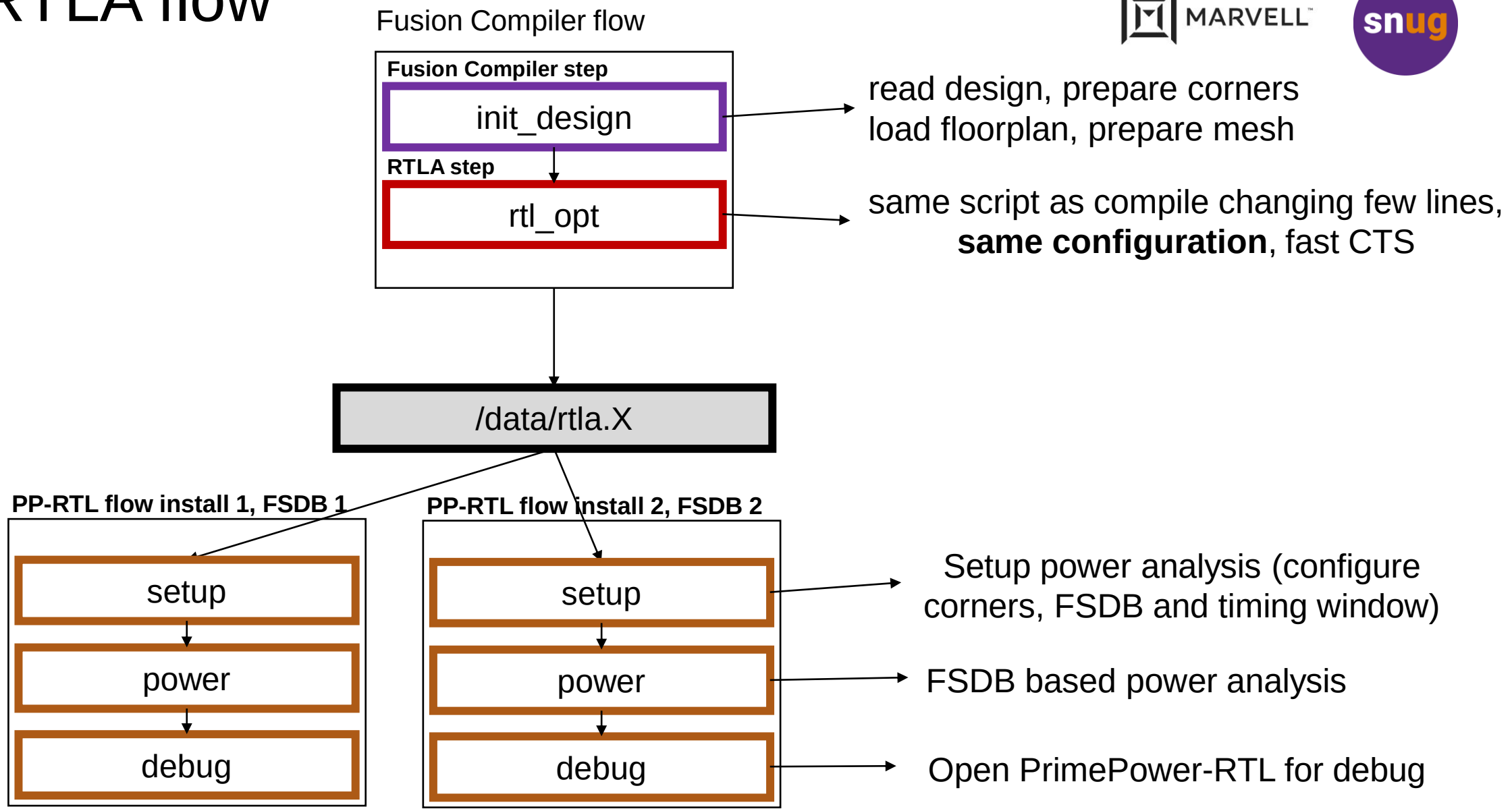
- Stand-alone PrimePower-RTL flow
  - Maps, applies and propagates vectors from RTL simulation (ports and flops) to synthesis models.
  - Glitch power available by annotating delays to cells
  - Can be installed multiple times on same data to test different FSDB
  - Generates annotation and consumption reports
  - User can open GUI for further session debugging

Fusion flow



Turnaround time: ~1 week

PP-RTLA flow



Turnaround time: ~1 day

Results in Orion Coherent DSP Chip

FC / RTLA correlation (at compile)

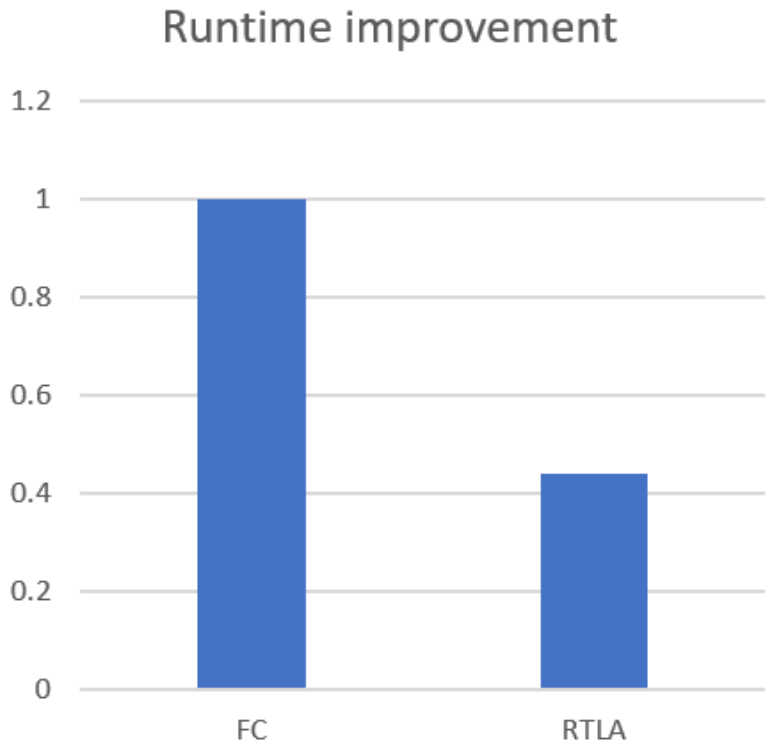
Runtime	FC Runtime	RTLA Runtime	Speedup
dsp_tx	10.46h	4.81h	2.17x
dsp_rx	17.80h	7.17h	2.48x
sfec	27.43h	9.40h	2.91x
host	20.60h	13.35h	1.54x

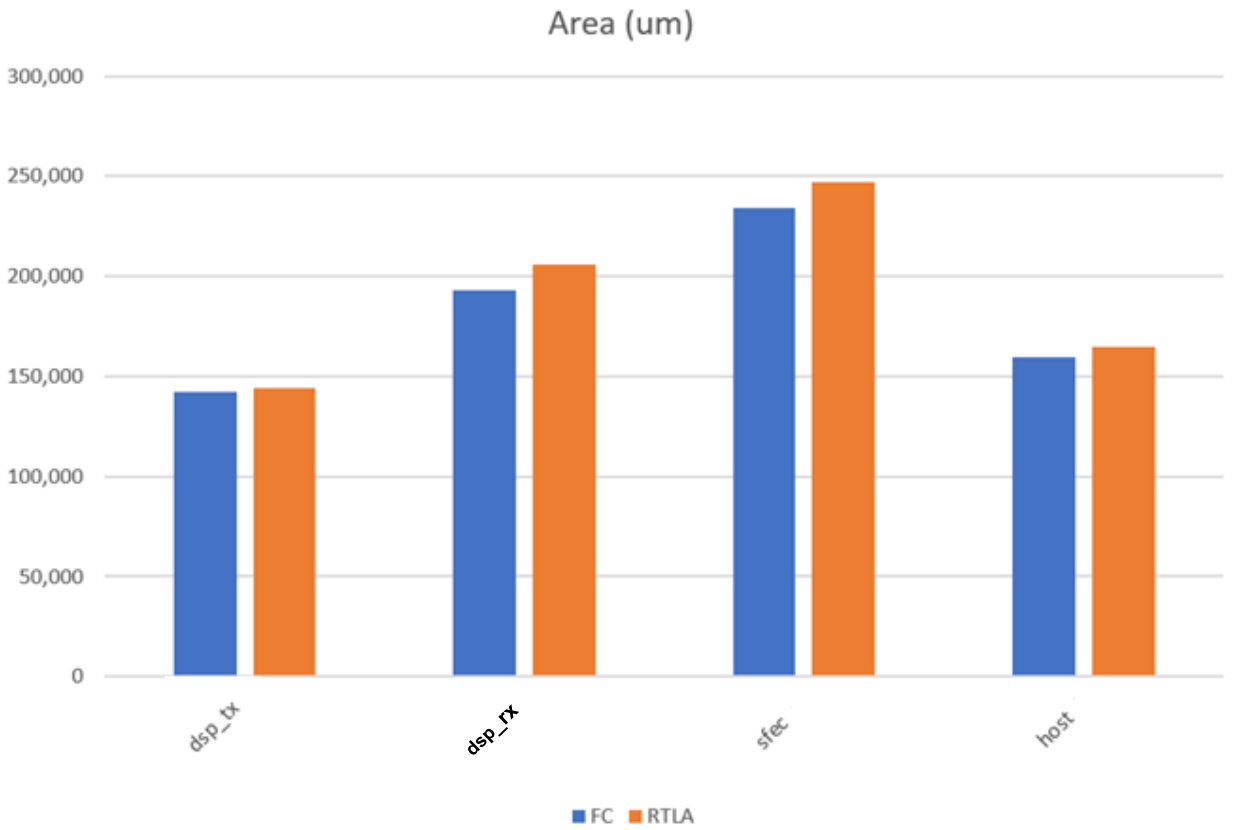
1.5-3x speedup

<5% count/area difference

Combinational Cells	Count FC	Count RTLA	Diff %	Area FC um2	Area RTLA um2	Diff %
dsp_tx	1,927,284	1,988,375	+3,17%	142,014	144,394	+1,68%
dsp_rx	2,527,708	2,608,124	+3,18%	193,080	205,783	+6,58%
sfec	3,754,579	3,752,281	-0,06%	234,290	246,940	+5,41%
host	1,857,575	1,871,080	+0,01%	159,404	164,727	+3,34%
Sequential Cells	**Count FC**	**Count RTLA**	**Diff %**	**Area FC um2**	**Area RTLA um2**	**Diff %**
dsp_tx	38,742	38,279	-1,20%	38,755	38,776	+0,05%
dsp_rx	58,232	58,068	-0,28%	180,799	186,229	+3,00%
sfec	306,195	286,437	-6,45%	345,093	345,964	+0,25%
host	177,160	74,392	-58,0% (Different retiming setup)	80,568	76,899	-4,55%
Average first 3	-	-	2,64%	-	-	1,1%

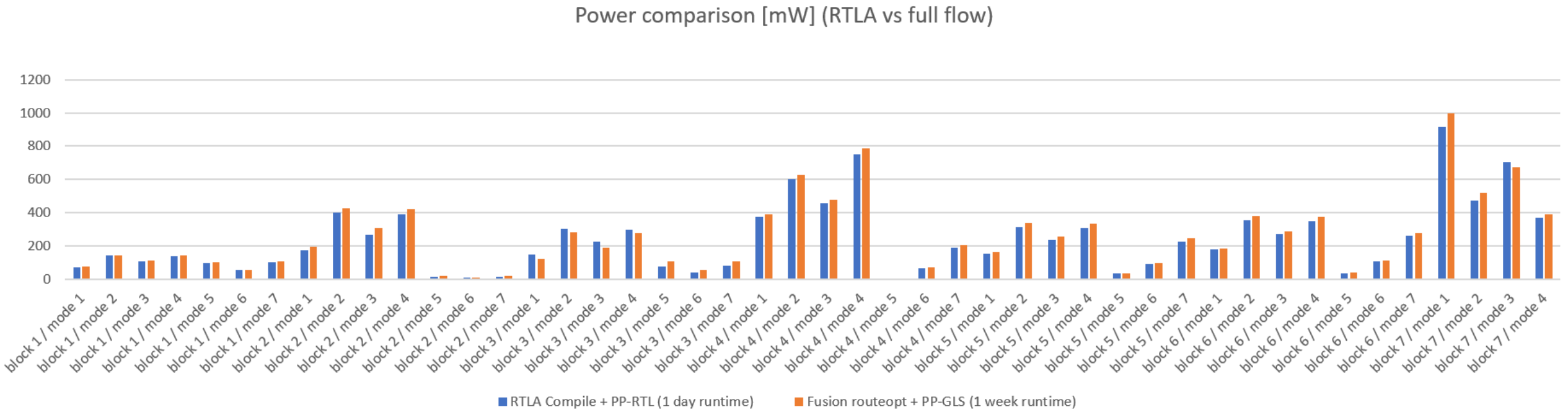
FC / RTLA correlation (at compile)





Timing was closed in all cases

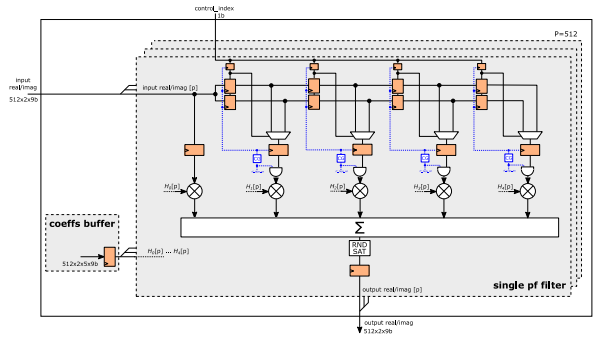
Power correlation (full flow)

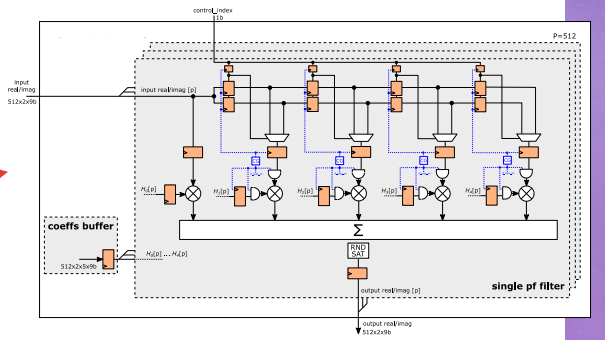


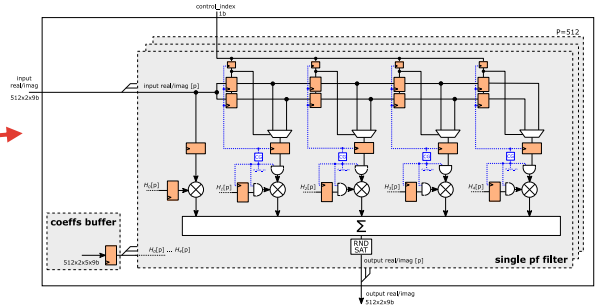
Quick turnaround time allows for faster exploration using reliable data

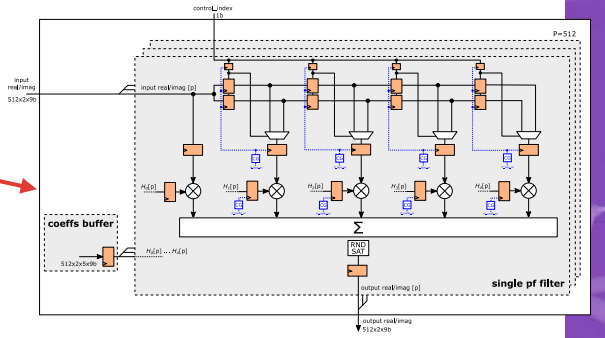
Power reduction on a single block

	Cells count (M)	PPRTLA power with BE floorplan (mW)						
		Total power	Leakage power	Switching power	Internal power	Sequential power	Combinational power	Glitch power
dsp_rx_filter_netlist (with pipes after mults) STABLE RELEASE	**3.13**	**882.0**	8.9	527.0	346.0	120.0	762.0	**338.0**
|---u_dsp_rx_filter		747.0	8.1	410.0	329.0	104.0	643.0	
|---u_dsp_rx_filter_pf		672.0	6.7	356.0	310.0	77.6	595.0	
|---u_coeffs_buffer		74.5	1.4	54.4	18.7	26.4	48.1	
dsp_rx_filter_netlist (no pipes after mults, pipe output)	**2.51**	**876.0**	11.3	560.0	304.0	55.8	820.0	**387.4**
|---u_dsp_rx_filter		755.0	10.6	451.0	294.0	44.1	711.0	
|---u_dsp_rx_filter_pf		678.0	9.2	391.0	278.0	27.8	650.0	
|---u_coeffs_buffer		76.9	1.4	59.5	16.0	16.3	60.6	
dsp_rx_filter_netlist (pipes after mux, pipe output)	**2.49**	**733.0**	10.9	456.0	266.0	88.7	645.0	**207.0**
|---u_dsp_rx_filter		649.0	10.4	381.0	257.0	75.5	573.0	
|---u_dsp_rx_filter_pf		564.0	9.0	317.0	238.0	55.1	508.0	
|---u_coeffs_buffer		84.8	1.4	64.5	18.9	20.2	64.6	
dsp_rx_filter_netlist (pipes after mux, pipes after coeffs buff, pipe output)	**2.57**	**711.0**	11.5	437.0	263.0	115.0	596.0	**163.8**
|---u_dsp_rx_filter		625.0	11.0	361.0	253.0	102.0	523.0	
|---u_dsp_rx_filter_pf		553.0	9.6	309.0	235.0	78.5	475.0	
|---u_coeffs_buffer		70.8	1.4	51.8	17.6	22.6	48.2	
dsp_rx_filter_netlist (pipes after mux, pipes after coeffs buff, NO pipe output, no ANDs in coeffs buffer)	**2.53**	**707.0**	11.8	432.0	264.0	109.0	599.0	**171.6**
|---u_dsp_rx_filter		610.0	11.1	350.0	249.0	92.6	517.0	
|---u_dsp_rx_filter_pf		572.0	9.8	328.0	234.0	70.4	502.0	
|---u_coeffs_buffer		37.7	1.3	21.6	14.8	22.2	15.5	
dsp_rx_filter_netlist (pipes after mux, pipes after coeffs buff, pipe output, remove all muting ANDs)	**2.20**	**718.0**	11.4	450.0	257.0	166.0	553.0	**144.9**
|---u_dsp_rx_filter		633.0	10.9	374.0	248.0	152.0	480.0	
|---u_dsp_rx_filter_pf		588.0	9.7	346.0	232.0	125.0	463.0	
|---u_coeffs_buffer		44.4	1.2	28.0	15.1	27.3	17.1	



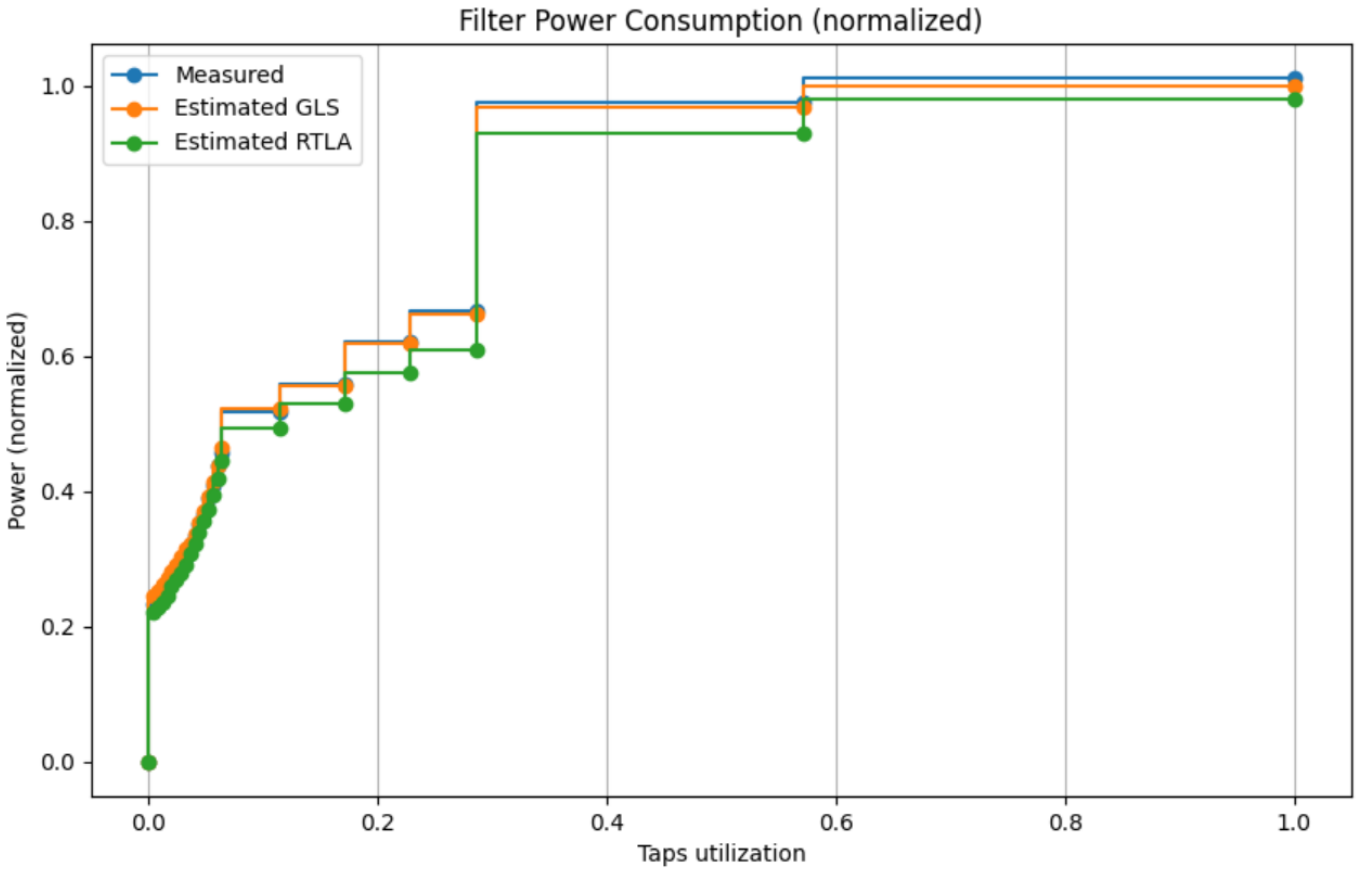






Example driving 20% power reduction via RTL optimization

Lab data correlation



- The power consumption in this figure is the aggregate behavior of multiple netlists operating in a specific mode within a certain operational range.
- For each data point, these netlists can be configured at specific frequencies or settings to cover that particular point.
- The estimated values are derived from RTLA estimations using FSDB data and GLS with annotated activities.
- These estimations are performed across various partitions, with frequency and voltage scaling applied to predict the real consumption

Conclusions

Conclusions

Summary

- Early prototyping flow for Front-end team to estimate PD results
- Co-development between CAD, Front-end and Back-end teams
- The flow is used in production projects
- Provided data correlating early exploration vs final results

Value provided

- We helped bring PD data earlier in the project life cycle
- Enabled Front-End team make better decisions and optimize RTL better (helped achieve 20% total power reduction on dsp_rx)

Improvements

- Better integration between RTLA and PPRTL would help increase engineer productivity
- Keeping FC and RTLA results aligned is key for having high confidence on the results.
- We plan to work on correlation when using custom clock networks.


THANK YOU
YOUR
INNOVATION
YOUR
COMMUNITY