

IARPA SuperTools Deliverable

ColdFlux Logic Cell Library for MIT-LL SFQ Process

Submitted by

ColdFlux Team
Yokohama National University
Stellenbosch University

Version 2.2

Version History

Version 2.2

This is an updated version of the cell library for Phase 2B after taking into account feedback from the Test & Evaluation teams. The key changes for the AQFP part of the cell library are as follows:

- Replaced the Josephson junction designs of the AQFP cells to that of a grid-snapped design to adhere with MITLL's design grid resolutions.

The key changes for the RSFQ part of the cell library are as follows:

- Minor updates to cell layouts to adhere to latest DRC rules.
- LEF files, extracted using qPALACE, are included in the library.

Version 2.1

This version is the ColdFlux cell library deliverable for SuperTools Phase 2B. The key changes for the AQFP part of the cell library are as follows:

- Completely migrated and re-done schematic/symbol/netlist data from previous Xic formats to the gEDA gschem formats (same format as RSFQ).
- Introduction of new booster cells for driving long distance interconnects.
- Addition of refined MAJ5, AND3, and OR3 logic cells.
- Removal of dummy JJs in schematics/netlists previously used to avoid simulation issues in WRspice/jsim_n. JoSIM is the main simulator now which has no issues with the removal of these dummy JJs.
- LVS-clean cell data verified using InductEx-LVS.

Version 2.1

This version is the ColdFlux cell library deliverable for SuperTools Phase 2B. The key changes for the AQFP part of the cell library are as follows:

- Completely migrated and re-done schematic/symbol/netlist data from previous Xic formats to the gEDA gschem formats (same format as RSFQ).
- Introduction of new booster cells for driving long distance interconnects.

- Addition of refined MAJ5, AND3, and OR3 logic cells.
- Removal of dummy JJs in schematics/netlists previously used to avoid simulation issues in WRspice/jsim_n. JoSIM is the main simulator now which has no issues with the removal of these dummy JJs.
- LVS-clean cell data verified using InductEx-LVS.

The key changes for the RSFQ part of the cell library are as follows:

- RSFQ cells were redesigned from first principles using phase-based equations.
- Cell versions both with and without integrated PTL transmitters and receivers are included in version 2.1.
- Base circuit netlists are included to show how the base cell is designed. An optimized circuit netlist represents the optimized circuit, as extracted from the layout using InductEx.
- A testbench for each cell is included for easy user verification of cell functionality.
- The mesh file is also included in version 2.1 to allow the user to view the 3D layout as generated by TTH/InductEx.
- An XNOR and XNORT cell was developed.
- Additional Always0 (both synchronous and asynchronous) and Always0T cells were developed.
- Parameterized cell layouts are in development, exposing parameters such as inductor widths and track pitch. This allows scaling and regeneration of layouts while maintaining inductance.

Version 2.0

Updates the AQFP and RSFQ logic cell libraries for SuperTools Phase 2A, and combines both libraries in a single document.

Version 1.5

Previous release of AQFP Logic cell library document.

Version 1.1

Previous release of RSFQ Logic cell library document.

Acknowledgment

The research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the U.S. Army Research Office grant W911NF-17-1-0120. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation herein.

Contents

I AQFP Logic	1
1 Introduction and Setup	2
1.1 Introduction	2
1.2 Cell library structure	3
1.3 Conventions	4
1.4 Full List of AQFP Cells	5
1.5 Library setup for schematic capture in gSchem	8
1.5.1 Recommended: Working inside <code>lib_jjmit</code>	8
1.5.2 Using your own workspace	8
1.6 Simulation examples	9
1.7 Useful commands and helper scripts	10
1.7.1 Useful commands	10
1.7.2 Helper scripts	11
1.8 Roadmap	12
2 AQFP Cell Library	13
2.1 Combinational Cells	13
2.1.1 BFR	13
2.1.2 INV	19
2.1.3 AND2	24
2.1.4 AND3	30
2.1.5 OR2	37
2.1.6 OR3	43
2.1.7 MAJ3	49
2.1.8 MAJ5	55
2.1.9 BOOST1	62
2.1.10 BOOST2	67
2.1.11 SPL	73
2.2 Sequential Cells	79
2.2.1 QFPL	79
2.2.2 NDRO_QFPL	85
2.2.3 NDRO_FB	92
2.3 Interconnect slices	99
2.4 Off-chip Interface	103

2.4.1	PRE_BSQUID	103
2.4.2	BFR_SQUID	105
2.4.3	QDC	109
2.5	On-chip Interfaces	114
2.5.1	AQFP2RSFQ	114
2.5.2	RSFQ2AQFP	118
2.6	Sub-Cells	122
2.6.1	Constant	122
2.6.2	Branch	124
2.6.3	Storage gate	126
2.6.4	HDL ac/dc interface	129
2.6.5	HDL dc interface	130
2.7	Standard Delay Format (SDF)	131
II	RSFQ Logic	142
3	Introduction and Setup	143
3.1	Introduction	143
3.2	Setup	145
3.3	License	145
4	RSFQ Cell Library: Standard Connections	146
4.1	Interconnects	146
4.1.1	JTL	146
4.1.2	SPLIT	152
4.1.3	MERGE	158
4.1.4	PTLTX	165
4.1.5	PTLRX	171
4.1.6	Always0 Asynchronous	177
4.1.7	Always0 Synchronous	183
4.2	Logic Cells	189
4.2.1	AND2	189
4.2.2	OR2	198
4.2.3	XOR	206
4.2.4	NOT	215
4.2.5	XNOR	222
4.3	Buffers	231
4.3.1	DFF	231
4.3.2	NDRO	238
4.3.3	BUFF	246
4.4	Interface cells	252
4.4.1	DCSFQ	252
4.4.2	SFQDC	256

5 RSFQ Cell Library: PTL Connections	260
5.1 Interconnects	260
5.1.1 JTLT	260
5.1.2 SPLITT	266
5.1.3 MERGET	272
5.1.4 Always0T Asynchronous	279
5.1.5 Always0T Synchronous	283
5.2 Logic Cells	287
5.2.1 AND2T	287
5.2.2 OR2T	296
5.2.3 XORT	304
5.2.4 NOTT	313
5.2.5 XNORT	321
5.3 Buffers	330
5.3.1 DFFT	330
5.3.2 NDROT	338
5.3.3 BUFFT	346
5.4 Interface cells	352
5.4.1 DCSFQ-PTLTX	352
5.4.2 PTLRX-SFQDC	356
Appendices	360
A RSFQ PCell Script Example	361

List of Figures

1.1	gschem interface with the AQFP library (MIT LL AQFP) loaded.	9
2.1	bfr symbol.	13
2.2	bfr schematic.	14
2.3	bfr layout.	15
2.4	bfr analog waveform.	16
2.5	bfr digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	17
2.6	inv symbol.	19
2.7	inv schematic.	20
2.8	inv layout.	20
2.9	inv analog waveform.	22
2.10	inv digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	23
2.11	and2_pp symbol.	24
2.12	and2_pp schematic.	25
2.13	and2_pp layout.	25
2.14	and2_pp analog waveform.	28
2.15	and2_pp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	29
2.16	and3_ppp symbol.	30
2.17	and3_ppp schematic.	31
2.18	and3_ppp layout.	31
2.19	and3_ppp analog waveform.	34
2.20	and3_ppp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	35
2.21	or2_pp symbol.	37
2.22	or2_pp schematic.	38
2.23	or2_pp layout.	38
2.24	or2_pp analog waveform.	41
2.25	or2_pp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	42
2.26	or3_ppp symbol.	43
2.27	or3_ppp schematic.	44
2.28	or3_ppppp layout.	44
2.29	or3_ppp analog waveform.	47

2.30 or3_ppp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	48
2.31 maj3_ppp symbol.	49
2.32 maj3_ppp schematic.	50
2.33 maj3_ppp layout.	50
2.34 maj3_ppp analog waveform.	52
2.35 maj3_ppp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	53
2.36 maj5_ppppp symbol.	55
2.37 maj5_ppppp schematic.	56
2.38 maj5_ppppp layout.	56
2.39 maj5_ppppp analog waveform.	58
2.40 maj5_ppppp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	60
2.41 boost1 symbol.	62
2.42 boost1 schematic.	63
2.43 boost1 layout.	63
2.44 boost1 analog waveform.	65
2.45 sp12 digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	66
2.46 boost2f2 symbol.	67
2.47 boost2f2 schematic.	68
2.48 boost2f2 layout.	68
2.49 boost2f2 analog waveform.	71
2.50 sp12 digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	72
2.51 sp12 symbol.	73
2.52 sp12 schematic.	74
2.53 sp12 layout.	74
2.54 sp12 analog waveform.	76
2.55 sp12 digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	77
2.56 qfp1 symbol.	79
2.57 qfp1 schematic.	80
2.58 qfp1 layout.	81
2.59 qfp1 analog waveform.	83
2.60 qfp1 digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	84
2.61 ndro_qfp1 symbol.	85
2.62 ndro_qfp1 schematic.	86
2.63 ndro_qfp1 analog waveform.	90
2.64 ndro_qfp1 digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	91
2.65 ndro_fb symbol.	92
2.66 ndro_fb schematic.	93
2.67 ndro_fb analog waveform.	97

2.68	<code>ndro_fb</code> digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.	98
2.68	Cell-to-cell wire interconnect slices made of stripline-type passive transmission lines. Each slice is $10\text{ }\mu\text{m} \times 10\text{ }\mu\text{m}$	102
2.69	<code>pre_bsquid</code> symbol.	103
2.70	<code>pre_bsquid</code> schematic.	104
2.71	<code>pre_bsquid</code> layout.	104
2.72	<code>bfr_squid</code> symbol.	105
2.73	<code>bfr_squid</code> schematic.	106
2.74	<code>bfr_squid</code> layout.	107
2.75	<code>qdc</code> symbol.	109
2.76	<code>qdc</code> schematic.	110
2.77	<code>qdc</code> layout.	110
2.78	<code>qdc</code> analog waveform.	113
2.79	<code>aqfp2rsfq</code> symbol.	114
2.80	<code>aqfp2rsfq</code> schematic.	115
2.81	<code>aqfp2rsfq</code> layout.	115
2.82	<code>aqfp2rsfq</code> analog waveform.	117
2.83	<code>rsfq2aqfp</code> symbol.	118
2.84	<code>rsfq2aqfp</code> schematic.	119
2.85	<code>rsfq2aqfp</code> layout.	119
2.86	<code>rsfq2aqfp</code> analog waveform.	121
2.87	<code>const0</code> symbol.	122
2.88	<code>const0</code> schematic.	123
2.89	<code>const0</code> layout.	123
2.90	<code>branch2</code> symbol.	124
2.91	<code>branch2</code> schematic.	125
2.92	<code>branch2</code> layout.	126
2.93	<code>storage_gate</code> symbol.	127
2.94	<code>storage_gate</code> schematic.	127
2.95	<code>storage_gate</code> layout.	128
4.1	Schematic of RSFQ JTL.	146
4.2	RSFQ JTL Layout	147
4.3	RSFQ JTL analog simulation results.	149
4.4	RSFQ JTL digital simulation results.	151
4.5	RSFQ JTL Mealy finite state machine diagram.	151
4.6	Schematic of RSFQ SPLIT.	152
4.7	RSFQ SPLIT layout.	153
4.8	RSFQ SPLIT analog simulation results.	155
4.9	RSFQ SPLIT digital simulation results.	157
4.10	RSFQ SPLIT Mealy finite state diagram.	157
4.11	Schematic of RSFQ MERGE.	158
4.12	RSFQ MERGE layout.	159
4.13	RSFQ MERGE analog simulation results.	161
4.14	RSFQ MERGE digital simulation results.	164

4.15 RSFQ MERGE Mealy finite state diagram.	164
4.16 Schematic of RSFQ PTLTX.	165
4.17 RSFQ PTLTX Layout.	166
4.18 RSFQ PTLTX analog simulation results.	168
4.19 RSFQ PTLTX digital simulation results.	170
4.20 Schematic of RSFQ PTLRX.	171
4.21 RSFQ PTLRX Layout.	172
4.22 RSFQ PTLRX analog simulation results.	174
4.23 RSFQ PTLRX digital simulation results.	176
4.24 Schematic of RSFQ Always0 Asynchronous.	177
4.25 Schematic of RSFQ Always0 Asynchronous without an a input port.	177
4.26 The physical layouts for (a) the RSFQ Always0 Asynchronous and (b) the RSFQ Always0 Asynchronous without an a input port.	178
4.27 Schematic of RSFQ Always0 Synchronous.	183
4.28 Schematic of RSFQ Always0 Synchronous without an a input port.	183
4.29 The physical layouts for (a) the RSFQ Always0 Synchronous and (b) the RSFQ Always0 Synchronous without an a input port.	184
4.30 Schematic of RSFQ AND2.	189
4.31 RSFQ AND2 Layout.	190
4.32 RSFQ AND2 analog simulation results.	193
4.33 RSFQ AND2 digital simulation results.	197
4.34 RSFQ AND2 Mealy finite state machine diagram.	197
4.35 Schematic of RSFQ OR2.	198
4.36 RSFQ OR2 Layout	199
4.37 RSFQ OR2 analog simulation results.	202
4.38 RSFQ OR2 digital simulation results.	205
4.39 RSFQ OR2 Mealy finite state machine diagram.	205
4.40 Schematic of RSFQ XOR.	206
4.41 RSFQ XOR Layout.	207
4.42 RSFQ XOR analog simulation results.	210
4.43 RSFQ XOR digital simulation results.	214
4.44 RSFQ XOR Mealy finite state machine diagram.	214
4.45 Schematic of RSFQ NOT.	215
4.46 RSFQ NOT Layout.	216
4.47 RSFQ NOT analog simulation results.	218
4.48 RSFQ NOT digital simulation results.	221
4.49 RSFQ NOT Mealy finite state machine diagram.	221
4.50 Schematic of RSFQ XNOR.	222
4.51 RSFQ XNOR Layout.	223
4.52 RSFQ XNOR analog simulation results.	226
4.53 RSFQ XNOR digital simulation results.	230
4.54 RSFQ XNOR Mealy finite state machine diagram.	230
4.55 Schematic of RSFQ DFF.	231
4.56 RSFQ DFF Layout.	232
4.57 RSFQ DFF analog simulation results.	234
4.58 RSFQ DFF digital simulation results.	237

4.59 RSFQ DFF Mealy finite state machine diagram.	237
4.60 Schematic of RSFQ NDRO.	238
4.61 RSFQ NDRO Layout.	239
4.62 RSFQ NDRO analog simulation results.	242
4.63 RSFQ NDRO digital simulation results.	245
4.64 RSFQ NDRO Mealy finite state machine diagram.	245
4.65 Schematic of RSFQ BUFF.	246
4.66 RSFQ BUFF Layout.	247
4.67 RSFQ BUFF analog simulation results.	249
4.68 RSFQ BUFF digital simulation results.	251
4.69 RSFQ BUFF Mealy finite state machine diagram.	251
4.70 Schematic of RSFQ DCSFQ.	252
4.71 RSFQ DCSFQ Layout.	253
4.72 RSFQ DCSFQ analog simulation results.	255
4.73 Schematic of RSFQ SFQDC.	256
4.74 RSFQ SFQDC Layout.	257
4.75 RSFQ SFQDC analog simulation results.	259
 5.1 Schematic of RSFQ JTLT.	260
5.2 RSFQ JTLT Layout.	261
5.3 RSFQ JTLT analog simulation results.	263
5.4 RSFQ JTLT digital simulation results.	265
5.5 RSFQ JTLT Mealy finite state machine diagram.	265
5.6 Schematic of RSFQ SPLITT.	266
5.7 RSFQ SPLITT layout.	267
5.8 RSFQ SPLITT analog simulation results.	269
5.9 RSFQ SPLITT digital simulation results.	271
5.10 RSFQ SPLITT Mealy finite state diagram.	271
5.11 Schematic of RSFQ MERGET.	272
5.12 RSFQ MERGET layout.	273
5.13 RSFQ MERGET analog simulation results.	275
5.14 RSFQ MERGET digital simulation results.	278
5.15 RSFQ MERGET Mealy finite state diagram.	278
5.16 Schematic of RSFQ Always0T Asynchronous.	279
5.17 Schematic of RSFQ Always0T Asynchronous without an a input port.	279
5.18 The physical layouts for (a) the RSFQ Always0T Asynchronous and (b) the RSFQ Always0T Asynchronous without an a input port.	280
5.19 Schematic of RSFQ Always0T Synchronous.	283
5.20 Schematic of RSFQ Always0T Synchronous without an a input port.	283
5.21 The physical layouts for (a) the RSFQ Always0T Synchronous and (b) the RSFQ Always0T Synchronous without an a input port.	284
5.22 Schematic of RSFQ AND2T.	287
5.23 RSFQ AND2T Layout.	288
5.24 RSFQ AND2T analog simulation results.	291
5.25 RSFQ AND2T digital simulation results.	295
5.26 RSFQ AND2T Mealy finite state machine diagram.	295

5.27 Schematic of RSFQ OR2T	296
5.28 RSFQ OR2T Layout.	297
5.29 RSFQ OR2T analog simulation results.	300
5.30 RSFQ OR2T digital simulation results.	303
5.31 RSFQ OR2T Mealy finite state machine diagram.	303
5.32 Schematic of RSFQ XORT.	304
5.33 RSFQ XORT Layout.	305
5.34 RSFQ XORT analog simulation results.	308
5.35 RSFQ XORT digital simulation results.	312
5.36 RSFQ XORT Mealy finite state machine diagram.	312
5.37 Schematic of RSFQ NOTT.	313
5.38 RSFQ NOTT Layout.	314
5.39 RSFQ NOTT analog simulation results.	317
5.40 RSFQ NOTT digital simulation results.	320
5.41 RSFQ NOTT Mealy finite state machine diagram.	320
5.42 Schematic of RSFQ XNORT.	321
5.43 RSFQ XNORT Layout.	322
5.44 RSFQ XNORT analog simulation results.	325
5.45 RSFQ XNORT digital simulation results.	329
5.46 RSFQ XNORT Mealy finite state machine diagram.	329
5.47 Schematic of RSFQ DFPT.	330
5.48 RSFQ DFPT Layout.	331
5.49 RSFQ DFPT analog simulation results.	334
5.50 RSFQ DFPT digital simulation results.	337
5.51 RSFQ DFPT Mealy finite state machine diagram.	337
5.52 Schematic of RSFQ NDROT.	338
5.53 RSFQ NDROT Layout.	339
5.54 RSFQ NDROT analog simulation results.	342
5.55 RSFQ NDROT digital simulation results.	345
5.56 RSFQ NDROT Mealy finite state machine diagram.	345
5.57 Schematic of RSFQ BUFFT.	346
5.58 RSFQ BUFFT Layout.	347
5.59 RSFQ BUFFT analog simulation results.	349
5.60 RSFQ BUFFT digital simulation results.	351
5.61 RSFQ BUFFT Mealy finite state machine diagram.	351
5.62 Schematic of RSFQ DCSFQ-PTLTX.	352
5.63 RSFQ DCSFQ-PTLTX Layout.	353
5.64 RSFQ DCSFQ-PTLTX analog simulation results.	355
5.65 Schematic of RSFQ PTLRX-SFQDC.	356
5.66 RSFQ PTLRX-SFQDC Layout.	356
5.67 RSFQ PTLRX-SFQDC analog simulation results.	359

List of Tables

1.1	Naming convention of AQFP logic cells.	4
1.2	Naming examples and explanations.	5
1.3	Pin naming conventions.	5
1.4	Summary listing of all AQFP cells to date.	5
2.1	bfr pin list.	14
2.2	bfr switching energy table.	18
2.3	inv pin list.	19
2.4	inv switching energy table.	23
2.5	and2_pp pin list.	24
2.6	and2_pp switching energy table.	29
2.7	and3_ppp pin list.	30
2.8	and3_ppp switching energy table.	36
2.9	or2_pp pin list.	37
2.10	or2_pp switching energy table.	42
2.11	or3_ppp pin list.	43
2.12	or3_ppp switching energy table.	48
2.13	maj3_ppp pin list.	49
2.14	maj3_ppp switching energy table.	54
2.15	maj5_ppppp pin list.	55
2.16	maj5_ppppp switching energy table.	61
2.17	boost1 pin list.	62
2.18	sp12 switching energy table.	66
2.19	boost2f2 pin list.	67
2.20	boost2f2 switching energy table.	72
2.21	sp12 pin list.	73
2.22	sp12 switching energy table.	78
2.23	qfp1 pin list.	79
2.24	qfp1 truth table.	80
2.25	qfp1 switching energy table.	84
2.26	ndro_qfp1 pin list.	85
2.27	ndro_qfp1 truth table.	86
2.28	ndro_fb pin list.	92

2.29	Summary of interconnect $10\text{ }\mu\text{m} \times 10\text{ }\mu\text{m}$ slices. Each slice is designated with bidirectional ports a-to-b and/or c-to-d. The signal layers for corresponding to those ports are listed as well as the extracted inductances. The shielding ground layers are also listed.	100
2.30	<code>pre_bsquid</code> pin list.	103
2.31	<code>stack</code> pin list.	106
2.32	<code>qdc</code> pin list.	109
2.33	<code>aqfp2rsfq</code> pin list.	114
2.34	<code>rsfq2aqfp</code> pin list.	118
2.35	<code>const0</code> pin list.	122
2.36	<code>branch2</code> pin list.	125
2.37	<code>storage_gate</code> pin list.	127
4.1	RSFQ JTL pin list.	148
4.2	RSFQ JTL power consumption.	151
4.3	RSFQ SPLIT pin list.	154
4.4	RSFQ SPLIT power consumption.	157
4.5	RSFQ MERGE pin list.	161
4.6	RSFQ MERGE power consumption.	164
4.7	RSFQ PTLTX pin list.	167
4.8	RSFQ PTLTX power consumption.	170
4.9	RSFQ PTLRX pin list.	173
4.10	RSFQ PTLRX power consumption.	176
4.11	RSFQ Always0 Asynchronous pin list.	179
4.12	RSFQ Always0 Asynchronous, without an a input port, pin list.	180
4.13	RSFQ Always0 Asynchronous power consumption.	182
4.14	RSFQ Always0 Asynchronous, without an a input port, power consumption.	182
4.15	RSFQ Always0 Synchronous pin list.	185
4.16	RSFQ Always0 Synchronous, without an a input port, pin list.	186
4.17	RSFQ Always0 Synchronous power consumption.	188
4.18	RSFQ Always0 Synchronous, without an a input port, power consumption.	188
4.19	RSFQ AND2 pin list.	192
4.20	RSFQ AND2 power consumption.	197
4.21	RSFQ OR2 pin list.	201
4.22	RSFQ OR2 power consumption.	205
4.23	RSFQ XOR pin list.	209
4.24	RSFQ XOR power consumption.	214
4.25	RSFQ NOT pin list.	218
4.26	RSFQ NOT power consumption.	221
4.27	RSFQ XNOR pin list.	225
4.28	RSFQ XNOR power consumption.	230
4.29	RSFQ DFF pin list.	234
4.30	RSFQ DFF power consumption.	237
4.31	RSFQ NDRO pin list.	241
4.32	RSFQ NDRO power consumption.	245
4.33	RSFQ BUFF pin list.	248

4.34 RSFQ BUFF power consumption.	251
4.35 RSFQ DCSFQ pin list.	255
4.36 RSFQ SFQDC pin list.	259
5.1 RSFQ JTLT pin list.	262
5.2 RSFQ JTLT power consumption.	265
5.3 RSFQ SPLITT pin list.	268
5.4 RSFQ SPLITT power consumption.	271
5.5 RSFQ MERGET pin list.	275
5.6 RSFQ MERGET power consumption.	278
5.7 RSFQ Always0T Asynchronous pin list.	281
5.8 RSFQ Always0T Asynchronous, without an a input port, pin list.	281
5.9 RSFQ Always0T Synchronous pin list.	285
5.10 RSFQ Always0T Synchronous, without an a input port, pin list.	285
5.11 RSFQ AND2T pin list.	290
5.12 RSFQ AND2T power consumption.	295
5.13 RSFQ OR2T pin list.	299
5.14 RSFQ OR2T power consumption.	303
5.15 RSFQ XORT pin list.	308
5.16 RSFQ XORT power consumption.	312
5.17 RSFQ NOTT pin list.	316
5.18 RSFQ NOTT power consumption.	320
5.19 RSFQ XNORT pin list.	324
5.20 RSFQ XNORT power consumption.	329
5.21 RSFQ DFFT pin list.	333
5.22 RSFQ DFFT power consumption.	337
5.23 RSFQ NDROT pin list.	342
5.24 RSFQ NDROT power consumption.	345
5.25 RSFQ BUFFT pin list.	348
5.26 RSFQ BUFFT power consumption.	351
5.27 RSFQ DCSFQ-PTLTX pin list.	354
5.28 RSFQ PTLRX-SFQDC pin list.	358

Part I

AQFP Logic

1. Introduction and Setup

1.1 Introduction

With our proposed minimalist design methodology, standard cell libraries have been built for MIT 10 kA/cm² Nb/Al-AlO_x/Nb tri-layer process. This cell library contains the basic logic cells, on-chip interfaces and off-chip interfaces as listed below. All cells are driven by 4-phase clock generated by 2 AC sources and a DC source [1].

In the previously released cell libraries (versions 1.4, 1.5, 1.6, and 1.7/2.0), all views of the cell library were developed using Xic from the XicTools suite [2]. From version 2.1, the cell's symbol view and schematic view were developed using gschem [3], whereas the GDS files can be viewed and edited by using Klayout [4]. A team decision amongst the cell library designers in ColdFlux was to drop Xic support due to undesirable user experience and difficulty in integrating ColdFlux tools into the Xic development branch. Further, while Xic had very good support for importing OpenAccess cell libraries, it has extremely limited and buggy capabilities for writing OpenAccess format data. To our knowledge, there are no other open-source tools that can use the OpenAccess API to read/write data, so we will no longer pursue this approach.

Section 1.3 details our naming conventions used for this cell library and the full list of cells is shown in Section 1.4. The library files for circuit design are GDS files that can be opened in Klayout or any other tool that can open standard GDS files.

The components of each cell are listed as follows:

- **Symbol:** These files end with `.sym`. This is the symbol view of the cell. It is designed in gschem and provides a hierarchical way to build digital logic circuits. It is a simple symbol with terminal (I/O) definitions and a file attribute which points to the appropriate sub-circuit definition (`.cir` file).
- **Schematic:** These files end with `.sch`. This is the schematic view of the cell designed in gschem. The circuit parameters have been extracted from the physical layout using InductEx [5]. We define the parameters using the `.param` directive that JoSIM supports. This allows easier manipulation of circuit parameters by other tools such as JoSIM tools.
- **Layout:** These files end with `.gds`. This is the physical layout of the cell designed in KLayout. The GDS files are now standard format unlike in previous versions which also embed schematic and symbol views readable only by Xic. As mentioned above, the symbols and schematics are in separate files now.

- **InductEx layout:** These files end with `.ix.gds`. These are modified layouts to help perform inductance extraction. If an InductEx layout is not available, then extraction can be performed with the normal layout.
- **Standard netlist:** An `.cir` format netlist exported for each logic cell using gschem's utility `gnetlist` utility. Because our cells are built in a hierarchical fashion, the netlister will generate duplicated and nested `.subckt` blocks which JoSIM cannot handle. The library folder has a Python 3 script called `flatcir.py` which takes the original generated netlist, backs it up as `.gnetlist.cir` and overwrites the original `.cir` file as a flattened netlist with comments stripped away for a clean netlist ready to be used in a symbol in the case of a `.subckt` (as it is for all logic cells) or for simulation such as our `test_*.cir` netlists.
- **InductEx netlist:** These files end with `.ix.cir`. This is the netlist used for extracting inductances and is a slightly different format compared to the standard netlist.
- **LVS file :** These files end with `.json`. This the LVS configuration file already prepared to run LVS via `$ inductex-lvs (cellname).json`.
- **Analog waveform:** Using the analog circuit testbenches (prefixed with `test_*`) included in the cell library, functionality was confirmed using JoSIM [6].
- **Verilog model:** Behavior-level model of a cell with built-in timing specification block. It is written in hardware-description-language (HDL) Verilog and can be found as a separate file named 'CellName.v' under the `verilog` sub-directory. Each cell model has an embedded module named 'biasDir_b.v' as an I/O interface to produce a normalized clock based on the relative directions the AC and DC current.
- **SDF:** Standard delay format file containing propagation delays and timing constraints for each cell. Generated by the AQFP timing extraction tool AQFPTX [7]. This file is the `verilog` subdirectory.
- **Digital waveform:** Generated by the open source Verilog simulator Icvarus-Verilog [8] and viewed using the open source wave viewer GTKWave [9].
- **Switching energy:** Switching energies of each logic cell are calculated based on the methodology described in [10]. The energy information is provided as an input-frequency dependent look-up table (.csv) and can be found as a separate file named 'energy_cellName.csv' under the `energy` sub-directory. This lookup table summarizes the energy evolution based on different operating frequency and input Boolean logic patterns, which is compatible with the probabilistic power analysis tool AQFP-QPA.

1.2 Cell library structure

```
coldflux-aqfp-cell-library
|   README.md: readme file
|
```

```

|--- lib_jjmit: cell data (schematic, symbol, layout, test schematics)
|   |--- doc: PDF of current documentation of the library
|   |--- energy: Switching energy tables for each cell
|   |--- figures: Various screenshots for documentation
|   |--- lef: Contains LEF file for the library
|   |--- josim-waveforms: Simulation results of various cell tests
|   |--- lvs: LVS results of the cells produced by inductex-lvs
|   |--- tech: tech and ldf files for inductex, lvs
|   |--- verilog: Verilog models including global SDF
|
|--- legacy: old files and documentation for reference purposes.
|   |   cell-lib-doc-resources: contains .docx files for PDFs.

```

1.3 Conventions

The logic gates are named using the following convention:

`<gate><inputs>_<polarities>`

Table 1.1 describes in detail the different naming components and Table 1.2 shows some examples. Table 1.3 lists the pin naming conventions as well.

Table 1.1: Naming convention of AQFP logic cells.

Name Component	Description
<code><gate></code>	This is the shortened name of the gate. In the case where the gate has one or zero (e.g., constant cells) inputs, we simply use the <code><gate></code> .
<code><inputs></code>	This is the number of logic inputs, i.e., the number of inputs that directly produces the logic function. It doesn't include clocking inputs. In the case of fan-out circuits (<code>sp12</code> , <code>sp13</code>), the number indicates number of outputs/fan-out.
<code><polarities></code>	For each logic input i ($a, b, c, d\dots$), we designate its polarity as positive (<code>p</code>) or negative (<code>n</code>). The polarity mainly applies to Boolean logic gates like MAJ, OR, and AND. It is not used for splitter (<code>sp12</code> or <code>sp13</code>), for example. The ordering explicitly specifies exactly which input is which polarity. For example: <code>ppn</code> designates that input a is positive, input b is positive, and input c is negative.

Table 1.2: Naming examples and explanations.

Name	Explanation
maj3_ppn	A 3-input majority gate. Input a is positive, input b is positive, and input c is negative.
or2_pp	A 2-input OR gate. Both input a and b are positive inputs.
spl3	A 1-input to 3-output splitter/fan-out.

Table 1.3: Pin naming conventions.

Name	Convention
Input	$a, b, c, d\dots$
Output (single)	q
Output (multi)	$q_0, q_1, q_2\dots$
AC excitation (single)	x_{in} / x_{out}
AC excitation (multi)	$x_{in1}, x_{in2}, x_{in3\dots} / x_{out1}, x_{out2}, x_{out3\dots}$
DC offset (single)	d_{cin} / d_{cout}
DC offset (multi)	$d_{cin1}, d_{cin2}, d_{cin3\dots} / d_{cout1}, d_{cout2}, d_{cout3\dots}$

1.4 Full List of AQFP Cells

Table 1.4 shows a full listing of our AQFP cells. The cell name, type of cell, number of Josephson junctions (JJs), height and width in μm , and the Boolean function for each cell is shown. Type is defined as follows: ‘S’ refers to sub-cells which are used only for building logic cells and are not normally used standalone; ‘C’ refers to normal logic cells which are used to build digital circuits; ‘B’ refers to structures that can be both ‘S’ and ‘C’; ‘M’ refers to macro-type cells which are composed of ‘B’ or ‘C’ type cells. ‘M’ type cells do not have a fixed area as its own placement is dependent on how other neighboring cells are also arranged. ‘I’ type cells are interface cells for communicating off-chip or on-chip between AQFP-SFQ logic families. ‘W’ type cells are interconnect slices for both bias and data signal propagation.

Table 1.4: Summary listing of all AQFP cells to date.

Cell Name	Type	JJs	Height (μm)	Width (μm)	Boolean Function
bfr	B	2	43.5	22	$q = a$
inv	B	2	43.5	22	$q = \bar{a}$
const0	B	2	43.5	22	$q = 0$
const1	B	2	43.5	22	$q = 1$

Continued on the next page...

Table 1.4: (Continued from previous page.)

Cell Name	Type	JJs	Height (μm)	Width (μm)	Boolean Function
boost1	C	4	43.5	44	$q = a$
boost2f2	C	6	43.5	88	$q_0, q_1 = a$
boost2f4	C	6	43.5	132	$q_0, q_1, q_2, q_3 = a$
spl2	C	2	63.5	44	$q_0, q_1 = a$
spl3	C	2	63.5	66	$q_0, q_1, q_2 = a$
spl4	C	4	63.5	88	$q_0, q_1, q_2, q_3 = a$
maj3_ppp	C	6	63.5	66	$q = \text{MAJ}(a, b, c) = ab + ac + bc$
maj3_ppn	C	6	63.5	66	$q = \text{MAJ}(a, b, \bar{c}) = ab + a\bar{c} + b\bar{c}$
maj3_pnp	C	6	63.5	66	$q = \text{MAJ}(a, \bar{b}, c) = a\bar{b} + ac + \bar{b}c$
maj3_pnn	C	6	63.5	66	$q = \text{MAJ}(a, \bar{b}, \bar{c}) = a\bar{b} + a\bar{c} + \bar{b}\bar{c}$
maj3_npp	C	6	63.5	66	$q = \text{MAJ}(\bar{a}, b, c) = \bar{a}b + \bar{a}c + bc$
maj3_npn	C	6	63.5	66	$q = \text{MAJ}(\bar{a}, b, \bar{c}) = \bar{a}b + \bar{a}\bar{c} + b\bar{c}$
maj3_nnp	C	6	63.5	66	$q = \text{MAJ}(\bar{a}, \bar{b}, c) = \bar{a}\bar{b} + \bar{a}c + \bar{b}c$
maj3_nnn	C	6	63.5	66	$q = \text{MAJ}(\bar{a}, \bar{b}, \bar{c}) = \bar{a}\bar{b} + \bar{a}\bar{c} + \bar{b}\bar{c} = \text{MIN}(a, b, c)$
maj5_ppppp	C	10	63.5	110	$q = \text{MAJ}(a, b, c, d, e) = abc + abd + abe + acd + ace + ade + bcd + bce + bde + cde$
and2_pp	C	6	63.5	66	$q = ab$
and2_pn	C	6	63.5	66	$q = a\bar{b}$
and2_np	C	6	63.5	66	$q = \bar{a}b$
and2_nn	C	6	63.5	66	$q = \bar{a}\bar{b} = \overline{a + b}$
and3_ppp	C	10	63.5	110	$q = abc$
or2_pp	C	6	63.5	66	$q = a + b$

Continued on the next page...

Table 1.4: (Continued from previous page.)

Cell Name	Type	JJs	Height (μm)	Width (μm)	Boolean Function
or2_pn	C	6	63.5	66	$q = a + \bar{b}$
or2_np	C	6	63.5	66	$q = \bar{a} + b$
or2_nn	C	6	63.5	66	$q = \bar{a} + \bar{b} = \overline{ab}$
or3_ppp	C	10	63.5	110	$q = a + b + c$
qfp1	C	6	93.5	44	$q' = a(\overline{a \oplus b}) + q(a \oplus b)$
ndro_qfp1	M	18	N/A	N/A	$q' = de + q\bar{e}$
ndro_fb	M	26	N/A	N/A	$q' = de + q\bar{e}$
branch2	S	0	20	44	$q_0, q_1 = a$
branch3	S	0	20	66	$q_0, q_1, q_2 = a$
branch5	S	0	20	110	$q_0, q_1, q_2, q_3, q_4 = a$
storage_gate	S	2	30	44	N/A
pre_bsquid	S	2	20	22	N/A
bfr_squid	S	4	52.5	22	N/A
qdc	I	8	116	22	N/A
aqfp2rsfq	I	10	40	30	N/A
rsfq2aqfp	I	4	80	60	N/A
WIRE_V	W	0	10	10	N/A
WIRE_H	W	0	10	10	N/A
WIRE_C_M5_M3	W	0	10	10	N/A
WIRE_C_M5_M1	W	0	10	10	N/A
WIRE_C_M3_M1	W	0	10	10	N/A
VIA_M5_M3	W	0	10	10	N/A
VIA_C_M5_M1	W	0	10	10	N/A
VIA_C_M3_M1	W	0	10	10	N/A
VIA_ac_across	W	0	10	10	N/A
VIA_bias	W	0	10	10	N/A
bias_ac_10um	W	0	10	10	N/A

Continued on the next page...

Table 1.4: (Continued from previous page.)

Cell Name	Type	JJs	Height (μm)	Width (μm)	Boolean Function
bias_ac_corner	W	0	10	10	N/A
bias_ac_cross	W	0	10	10	N/A
bias_dc_10um	W	0	10	10	N/A
bias_pair_10um	W	0	10	10	N/A
bias_pair_11um	W	0	10	11	N/A
bias_pair_separate	W	0	10	10	N/A
bias_pair_separate2	W	0	10	10	N/A

1.5 Library setup for schematic capture in gSchem

The library is built using the gEDA environment. It can be loaded into the schematic capture tool `gschem` so that a circuit designer may instantiate AQFP logic cells into their own schematic design. This can be done through a `gafrc` initialization file in the working/design directory that `gschem` is invoked in.

1.5.1 Recommended: Working inside `lib_jjmit`

The easiest way to make sure all your designs netlist correctly is by working in the same folder as `lib_jjmit`. The folder already includes an appropriate `gafrc` file so the AQFP library symbols will load as seen in Figure 1.1.

1.5.2 Using your own workspace

This approach is not fully tested. We tried alternative ways to prepare the cells so that either the sub-circuit netlists or source schematics can be linked to the symbols regardless of where the files are located, but we ran into netlisting issues. The following is one way to get your schematics to netlist when working in a directory other than `lib_jjmit`.

1. Create a workspace directory for your designs. This can simply be a new folder in any location that you have full access permissions to. For example: `/home/user/gschem-designs`
2. Copy `lib_jjmit` to any accessible location.
3. In your workspace directory, create a file called `gafrc` whose contents are:

```
(component-library "/path/to/lib_jjmit" "MITLL AQFP")
(source-library "/path/to/lib_jjmit")
```

Where `/path/to/lib_jjmit` is the location where you copied `lib_jjmit`.

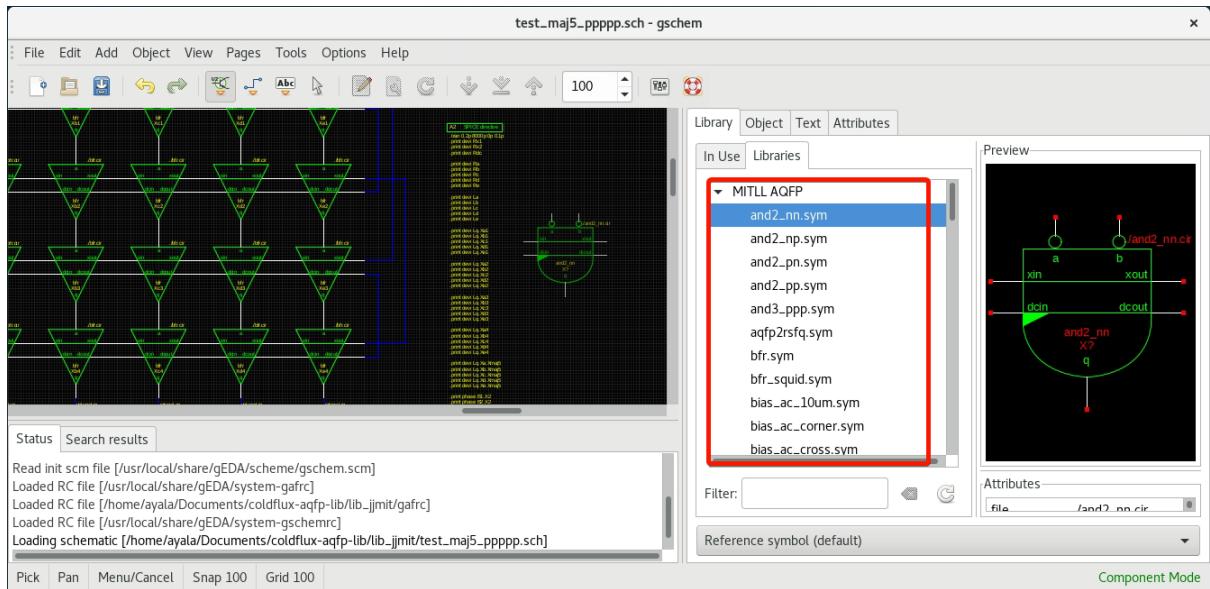


Figure 1.1: gschem interface with the AQFP library (MIT LL AQFP) loaded.

4. When you invoke `gschem` in your workspace directory, the AQFP library should be loaded under MITLL AQFP.
5. When you instantiate an AQFP cell, be sure to prepend the `file` attribute of each instance with your chosen `/path/to/lib_jjmit`. By default, the `file` attribute is simply `./cell-name.cir`. They should be changed to `/path/to/lib_jjmit/cell-name.cir` in order for `gnetlist` to properly find the sub-circuit netlists.

1.6 Simulation examples

Along with the logic cells themselves, we included several simulation examples for analog and digital simulation. For analog simulation, you will find several `test_*.sch` schematics with their corresponding `test_*.cir` netlist file. These netlist files can be simulated with the following command:

```
$ josim-cli -o {path/to/results/results.csv} test_{cell name}.cir
```

The above will run JoSIM on the desired `test_*.cir` file and store the simulation results as a CSV file in your desired location. The CSV file can be plotted using any CSV-compatible plotting tool.

For digital simulation, the `verilog/_celltestbench` contains a complete set of cell testbenches to perform digital simulation. They can be ran as follows:

```
$ iverilog *.v -o {cell name} //compile .v source file
$ ./{cell name} or vvp {cell name} // run simulation
$ gtkwave tb_{cell name}_example.vcd & // open the simulation result in waveform viewer
```

1.7 Useful commands and helper scripts

1.7.1 Useful commands

In this section, we list a few useful commands assuming you have the following environment:

- CentOS 7
- Python 3.8
- gEDA environment, <http://wiki.geda-project.org/geda:download>
- InductEx
- InductEx-LVS
- KLayout, <https://www.klayout.de/build.html>

Opening a schematic or symbol

gEDA's gschem tool is primarily used for schematic and symbol creation. You can use `gschem` to open the schematic files (`.sch`) or symbol files (`.sym`).

```
$ gschem {desired-schematic-or-symbol}.{sch-or-sym}
```

Opening a layout for viewing or editing

KLayout is the recommended tool for opening the GDS files. You can open them just for viewing via:

```
$ klayout {desired-cell-to-view}.gds
```

To open the GDS file for editing, you need to include `-e` as shown below:

```
$ klayout -e {desired-cell-to-view}.gds
```

Netlisting a schematic

`gschem` schematics are netlisted using gEDA's `gnetlist` utility with the `spice-sdb` format:

```
$ gnetlist -g spice-sdb -o {desired-netlist-filename}.cir {schematic-to-netlist}.sch
```

Note that AQFP cells are generally built hierarchically meaning internal sub-circuits may also contain sub-circuits within. At this moment, JoSIM cannot parse multiple levels of sub-circuit definitions in the netlist, but a workaround is to define all sub-circuits at the top-level. See Section 1.7.2 for `lib_jjmit/auto-netlist.sh` that will automatically netlist the desired schematic and then “flatten” the resulting netlist for JoSIM compatibility.

Simulating a netlist

```
$ josim-cli -o {desired-output-filename}.csv {netlist-to-simulate}.cir
```

Note that the netlists of the logic cells by themselves cannot be simulated as they are simply sub-circuits. A fully complete netlist with appropriate input sources, bias sources, output loads, simulation directives, and printing directives are needed. Our included collection of `test_*.sch` files and their corresponding `test_*.cir` netlists are all ready for simulation as explained in Section 1.6.

Running LVS

```
$ inductex-lvs {desired-cell}.json
```

Note that our `.json` files are configured to use `tech/mit_sfq5ee_v1.7.aqfp.tech` technology file and to store results into the `lvs` subdirectory.

Extracting inductances

```
$ inductex {desired-cell}.gds -n {desired-cell}.ix.cir -l tech/mit11_sfqee_set2.ldf
```

Note that some cells may have a special GDS for InductEx. They would be named as `*.ix.gds`. Also the `tech` subdirectory has a number of LDFs for InductEx. Typically `tech/mit11_sfqee_set2.ldf` is used for AQFP. But interface cells such as `rsfq2aqfp`, and `aqfp2rsfq` may need `tech/mit11_sfqee_res_HF1.ldf`. Currently, the netlists do not support the back annotation feature yet.

1.7.2 Helper scripts

Several helper scripts are bundled primarily to help maintain the cell library. But others may find the scripts useful too.

`lib_jjmit/flatcir.py`

Processes `*.cir` netlists so that all sub-circuits including nested ones are moved to the top-level of the netlist. Duplicate sub-circuits are removed and netlist comments are stripped. Original netlist will be overwritten but a backup of it will be saved as `*.gnetlist.cir`. The script is expected to work with netlists generated by `gnetlist -g spice-sdb`.

```
$ ./flatcir.py {desired-netlist-to-flatten}.cir
```

`lib_jjmit/josim-waveforms/simres_formatter.py`

Formats JSIM/JoSIM simulation results into Cadence VCSV (Virutoso Visualization & Analysis, ViVA format) or normal CSV format. By default (no optional arguments), it will convert JSIM results to VCSV format. In the context of the SuperTools project, we primarily use this script to convert the JoSIM results to VCSV format so that we can carefully inspect the waveforms in ViVA. Note: this script is not necessary if you use an alternative plotting tool that can read CSV files.

```
$ ./simres_formatter.py -p josim {josim-result}.csv
```

The above command will parse a JoSIM output result (`.csv` or `.dat`) and generate a VCSV file named `josim-result.csv.vcsv`. This VCSV file can be opened in Cadence Virtuoso Visualization & Analysis (ViVA) tool. More options can be found when running the script with `--help`.

lib_jjmit/auto-netlist.sh

This is a bash script that can accept multiple `*.sch` schematic files (wildcard globbing also possible) for netlisting. It will generate the normal netlist and then flatten it with `flatcir.py`.

```
$ ./auto-netlist.sh {file-1}.sch {file-2}.sch ... {file-n}.sch
```

lib_jjmit/auto-sim.sh

This is a bash script that can accept multiple `*.cir` netlist files (wildcard globbing also possible) for JoSIM simulation. It will run JoSIM sequentially for each netlist, save each result into the `josim-waveforms` subdirectory, and also generate Cadence VCSV format of each result.

```
$ ./auto-sim.sh {file-1}.cir {file-2}.cir ... {file-n}.cir
```

lib_jjmit/auto-netsim.sh

This is a bash script that can accept multiple `*.sch` schematic files (wildcard globbing also possible) for netlisting and immediate JoSIM simulation. It effectively combines the actions of `auto-netlist.sh` and `auto-sim.sh` into one script.

```
$ ./auto-netsim.sh {file-1}.sch {file-2}.sch ... {file-n}.sch
```

lib_jjmit/auto-lvs.sh

This is a bash script that can accept multiple `*.json` LVS files (wildcard globbing also possible) for running LVS via inductex-lvs.

```
$ ./auto-lvs.sh {file-1}.json {file-2}.json ... {file-n}.json
```

1.8 Roadmap

- **Implement new layout architecture (Phase 3):** The current RSFQ cell library has been modified to use a new routing architecture as described in [11]. It is compatible with the general design principles for AQFP logic. The establishment of this track routing architecture is a key step towards converging into a universal library that supports both AQFP and RSFQ logic. This would be a substantial redesign of the AQFP logic cells, but would enable a more systematic and consistent approach towards satisfying various density requirements that the MIT LL SFQ5ee process has. It will also further improve interoperability between AQFP and RSFQ circuits.
- **SPIRA-enabled layouts (Phase 3):** The basic AQFP cell (`bfr`) has been parameterized via SPIRA already but has not been modified to use the new routing architecture. After a re-design of the cells to use the new architecture, we plan to revisit layout parameterization again.

2. AQFP Cell Library

This section presents the schematic, netlist, physical layout, digital model, analog simulation, digital simulation, and switching energy of *representative* AQFP cells. We do not fully describe all variations because it is rather redundant. Instead, we show representative majority, AND, and OR cells. In the respective text, we list the variations of those cells that are also available. Table 1.4 is a complete listing of all of our AQFP cells.

2.1 Combinational Cells

2.1.1 BFR

The buffer (`bfr`) is the basic logic structure of AQFP circuits consisting of a double-Josephson-junction SQUID.

Symbol

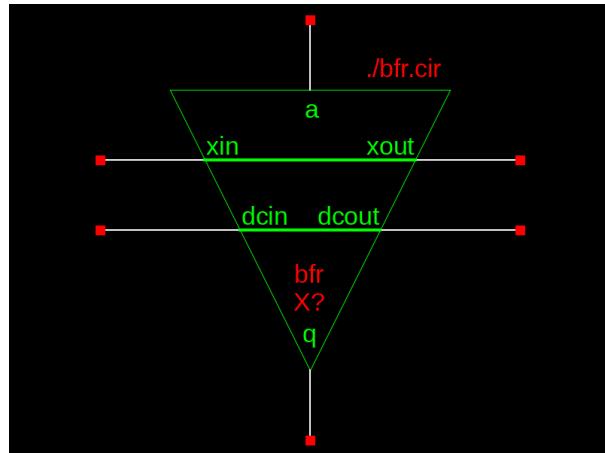
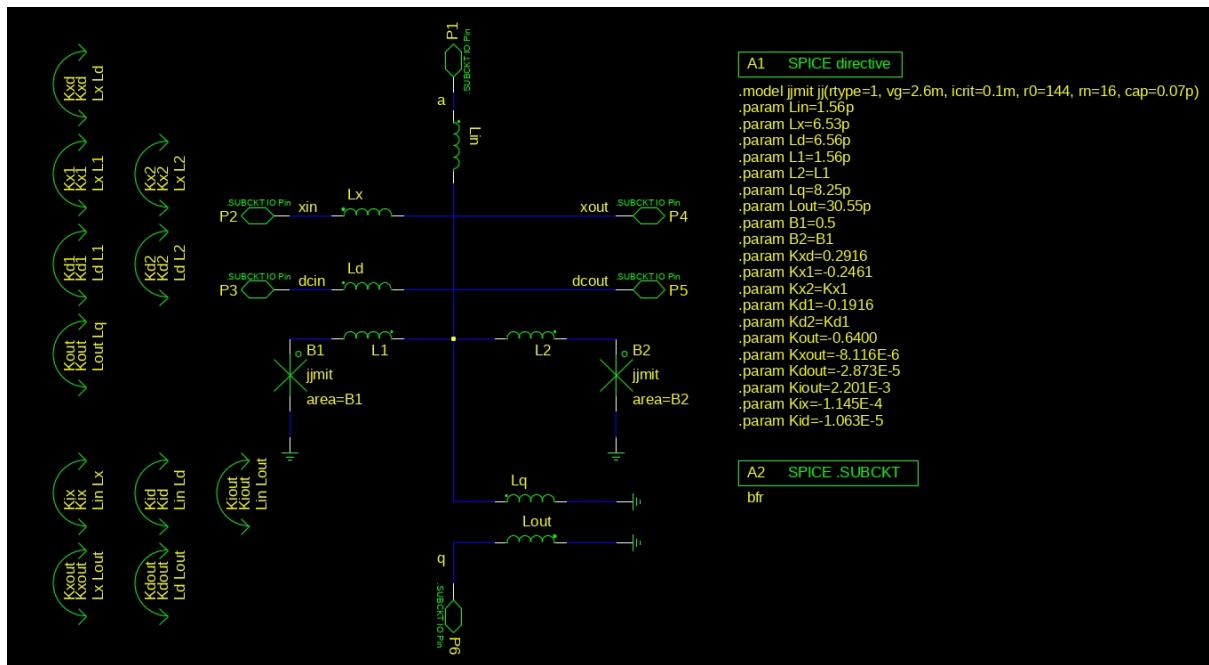


Figure 2.1: `bfr` symbol.

Table 2.1: bfr pin list.

Pin	Description
A	data input
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic**Figure 2.2:** bfr schematic.

Layout

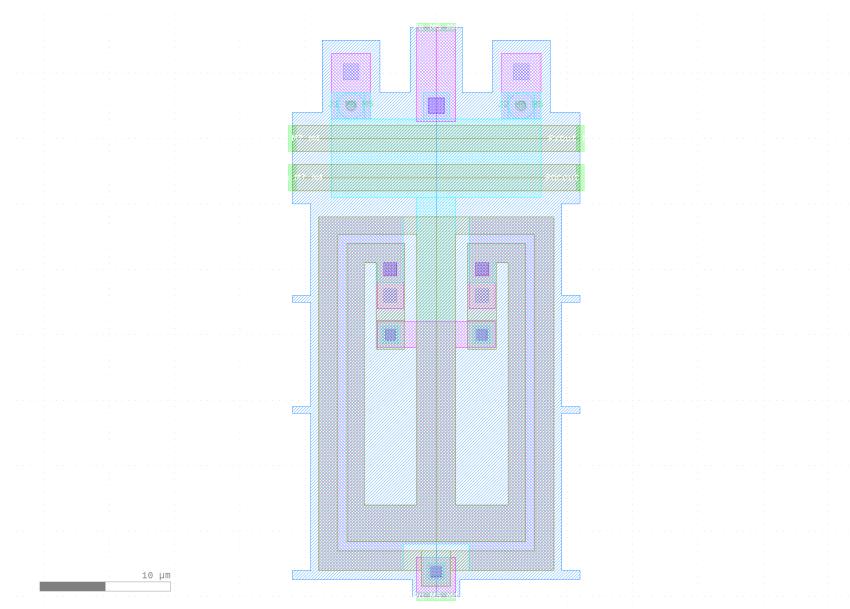


Figure 2.3: bfr layout.

Analog model

```

1  .SUBCKT bfr a xin dcin xout dcout q
2  Kid Lin Ld Kid
3  Kiout Lin Lout Kiout
4  Kix Lin Lx Kix
5  .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
6  .param Lin=1.56p
7  .param Lx=6.53p
8  .param Ld=6.56p
9  .param L1=1.56p
10 .param L2=L1
11 .param Lq=8.25p
12 .param Lout=30.55p
13 .param B1=0.5
14 .param B2=B1
15 .param Kxd=0.2916
16 .param Kx1=-0.2461
17 .param Kx2=Kx1
18 .param Kd1=-0.1916
19 .param Kd2=Kd1
20 .param Kout=-0.6400
21 .param Kxout=-8.116E-6
22 .param Kdout=-2.873E-5
23 .param Kiout=2.201E-3
24 .param Kix=-1.145E-4
25 .param Kid=-1.063E-5
26
27  Kxout Lx Lout Kxout
28  Kdout Ld Lout Kdout
29  Kout Lout Lq Kout
30  Kd2 Ld L2 Kd2
31  Kx2 Lx L2 Kx2
32  Kd1 Ld L1 Kd1
33  Kx1 Lx L1 Kx1
34  Kxd Lx Ld Kxd
35  Lin a 2 Lin

```

```

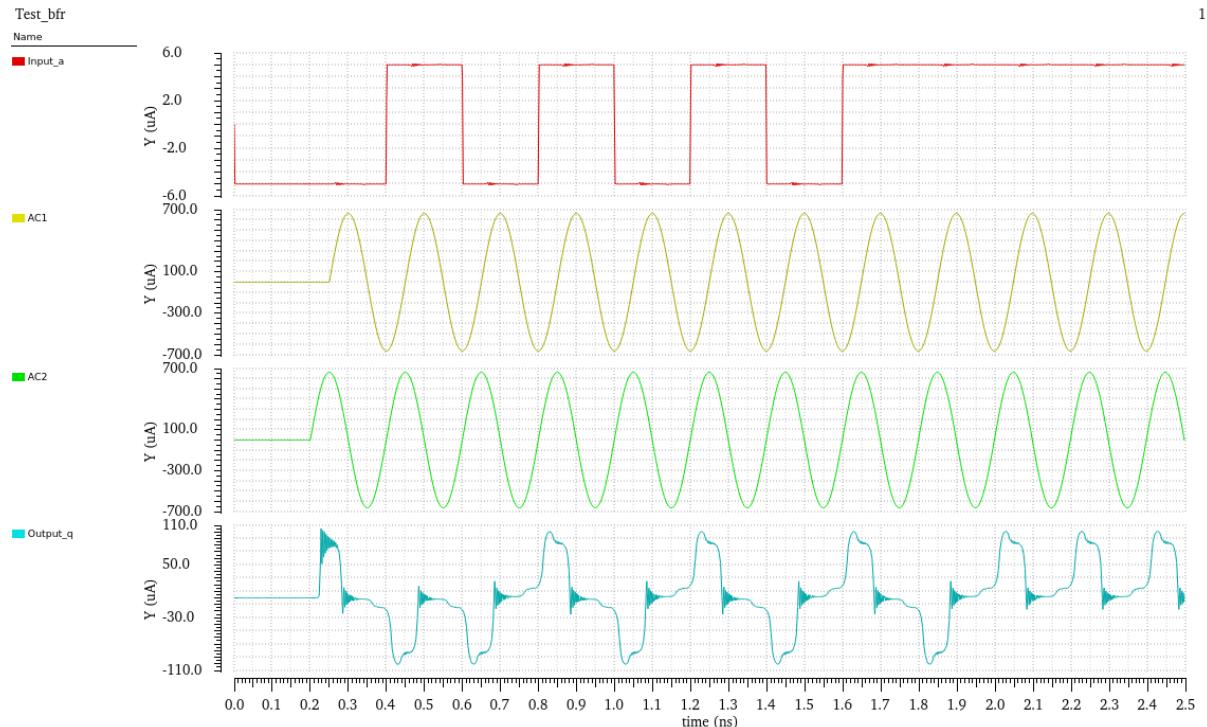
36 | B2 3 0 B2 jjmit area=B2
37 | B1 1 0 B1 jjmit area=B1
38 | Lout 0 q Lout
39 | Lq 2 0 Lq
40 | L2 3 2 L2
41 | L1 2 1 L1
42 | Ld dcin dcout Ld
43 | Lx xin xout Lx
44 | .ends bfr

```

Listing 2.1: bfr netlist (.cir).

Simulation result

Simulation waveform of an BUFFER (bfr) DUT (design under test) with 4-stage buffer before the input (a). Another 3-stage buffer is placed after the inverter's q output. Clock frequency is 5 GHz. Signals from top to bottom: buffer chain input (input of the first buffer) as 010101010111, AC source 1 (generates phase 1 and 3), AC source 2 (generates phase 2 and 4), and the output of the final out of the buffer chain as 10010101010111 with an random initial outputs 10. This is because of the meander structure of AQFP circuits. In an 8-stage AQFP circuit, it takes three clock cycles to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

**Figure 2.4:** bfr analog waveform.

Digital model

```

1 | 'timescale 1ps/10fs
2 | module bfr(a, q, xin, xout, dcin, dcout);
3 |   input a;
4 |   output q;

```

```

5   inout xin, xout, dcin, dcout;
6   reg q;
7   parameter pul_wid = 100;
8   wire not_a;
9
10  assign not_a = !a;
11
12  biasDir_b I0(xin, xout, dcin, dcout, gatex);
13
14  initial begin
15
16    $timeformat(-12, 1, "_ps", 8); // time format
17
18    // output register initialization
19    q = 1'bz;
20  end // initialization
21
22  specify
23    specparam d_clk    = 5;
24    specparam clk_d   = 50;
25
26    $setup(posedge a && a, posedge gatex, d_clk);
27    $setup(negedge a && not_a, posedge gatex, d_clk);
28    $hold(posedge gatex, negedge a && a, clk_d);
29    $hold(posedge gatex, posedge a && not_a, clk_d);
30  endspecify
31
32  always @(posedge gatex)
33  begin
34    if (a == 1 | a == 0)
35      begin
36        q <= a;
37        q <= #pul_wid 1'bz;
38      end
39    else
40      begin
41        q <= 1'bx;
42        q <= #pul_wid 1'bz;
43      end
44    end
45 endmodule

```

Listing 2.2: bfr Verilog model code.**Figure 2.5:** bfr digital waveform. HDL '1': AQFP '1'; HDL '0': AQFP '0'; HDL 'z': inactive; HDL 'x': error.

Switching energy

Different energy consumption results based on different data input patterns ($a = 0$ and $a = 1$) and clock frequencies (from 0.1GHz to 10GHz).

Table 2.2: bfr switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
‘0’ (J)	1.14E-23	2.32E-23	5.86E-23	1.18E-22	2.38E-22	7.18E-22	2.40E-21
‘1’ (J)	1.14E-23	2.32E-23	5.86E-23	1.18E-22	2.38E-22	7.18E-22	2.40E-21

2.1.2 INV

The AQFP inverter (`inv`) is one variation of the buffer cell. The operational principle is to generate the opposite direction of the output current by changing the sign of the coupling coefficient of the output transformer (e.g. K_{out} of the schematic in 2.7).

Symbol

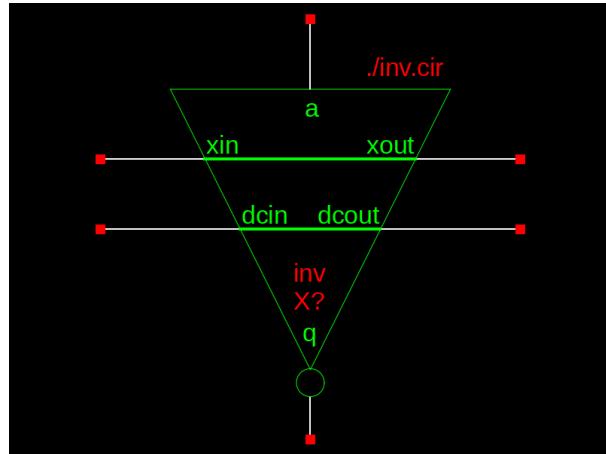


Figure 2.6: `inv` symbol.

Table 2.3: `inv` pin list.

Pin	Description
A	data input
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

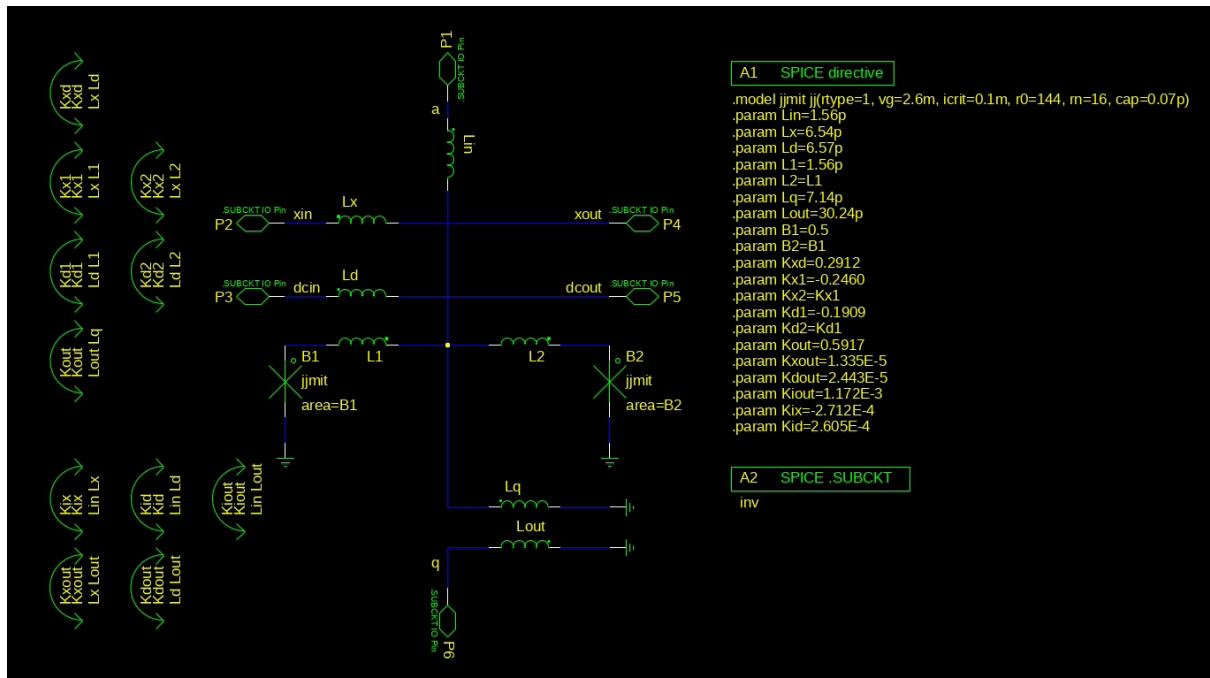


Figure 2.7: inv schematic.

Layout

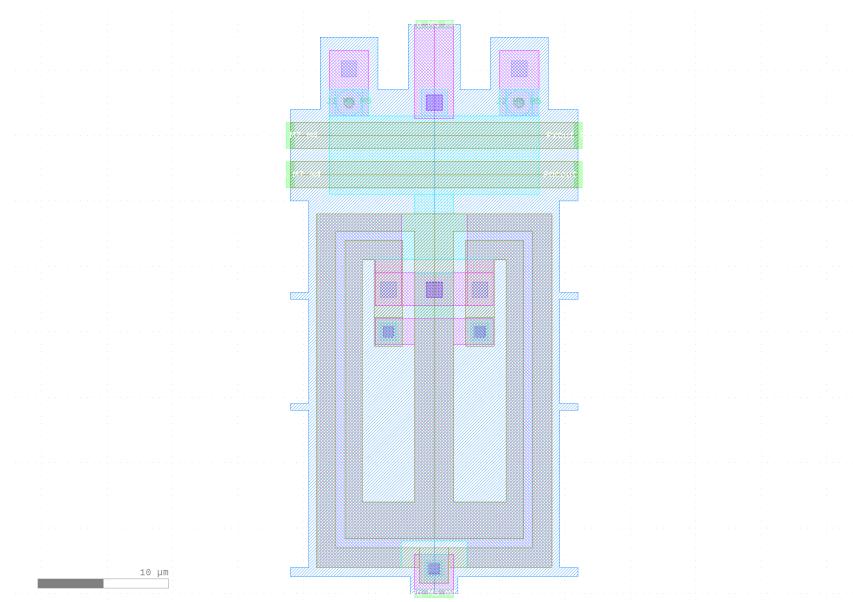


Figure 2.8: inv layout.

Analog model

```

1 .SUBCKT inv a xin dcin xout dcout q
2   Kid Lin Ld Kid
3   Kiout Lin Lout Kiout
4   Kix Lin Lx Kix
5   .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
6   .param Lin=1.56p
7   .param Lx=6.54p
8   .param Ld=6.57p
9   .param L1=1.56p
10  .param L2=L1
11  .param Lq=7.14p
12  .param Lout=30.24p
13  .param B1=0.5
14  .param B2=B1
15  .param Kxd=0.2912
16  .param Kx1=-0.2460
17  .param Kx2=Kx1
18  .param Kd1=-0.1909
19  .param Kd2=Kd1
20  .param Kout=0.5917
21  .param Kxout=1.335E-5
22  .param Kdout=2.443E-5
23  .param Kiout=1.172E-3
24  .param Kix=-2.712E-4
25  .param Kid=2.605E-4
26
27   Kxout Lx Lout Kxout
28   Kdout Ld Lout Kdout
29   Kout Lout Lq Kout
30   Kd2 Ld L2 Kd2
31   Kx2 Lx L2 Kx2
32   Kd1 Ld L1 Kd1
33   Kx1 Lx L1 Kx1
34   Kxd Lx Ld Kxd
35   Lin a 2 Lin
36   B2 3 0 B2 jjmit area=B2
37   B1 1 0 B1 jjmit area=B1
38   Lout 0 q Lout
39   Lq 2 0 Lq
40   L2 3 2 L2
41   L1 2 1 L1
42   Ld dcin dcout Ld
43   Lx xin xout Lx
44 .ends inv

```

Listing 2.3: inv netlist (.cir).

Simulation result

Simulation waveform of an INVERTER (inv) DUT (design under test) with 4-stage buffer before the input (a). Another 3-stage buffer is placed after the inverter's q output. Clock frequency is 5 GHz. Signals from top to bottom: buffer chain input a (input of the first buffer) as 01010101, AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), and the output (q) generated from the final stage of buffer as 1110101010 with two random initial outputs 11. This is because of the meander structure of AQFP circuits. In an 8-stage AQFP circuit, it takes three clock cycles to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

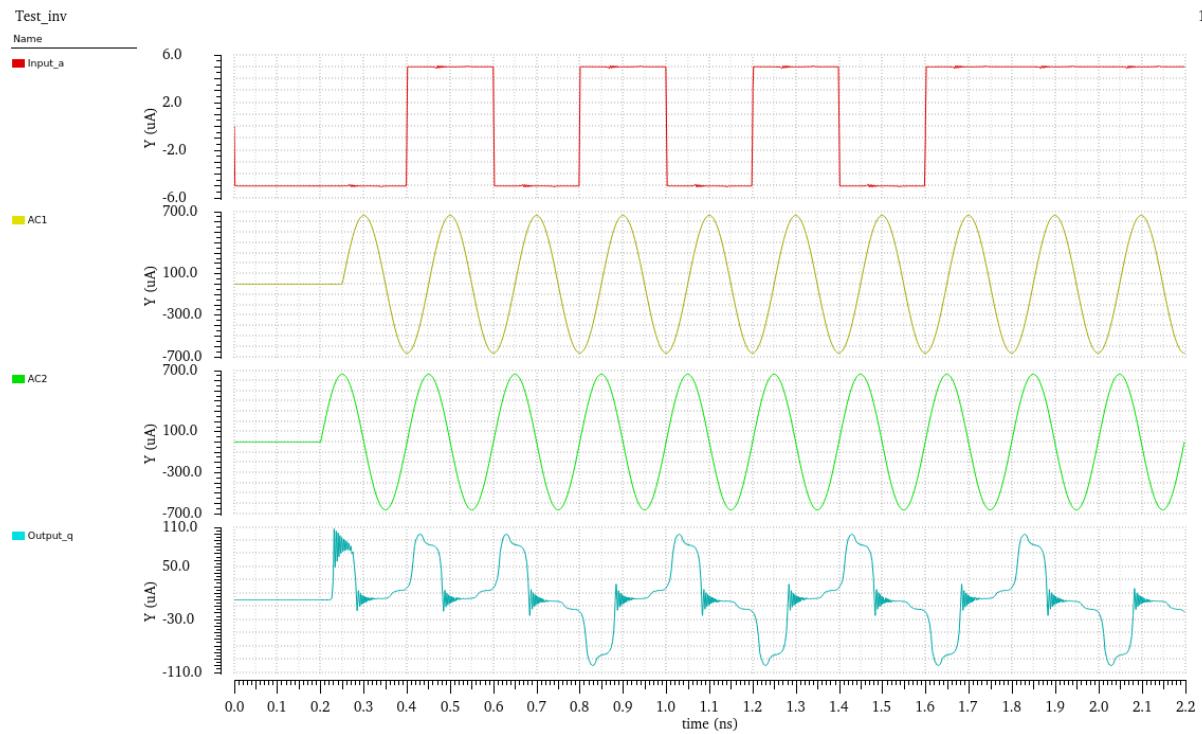


Figure 2.9: inv analog waveform.

Digital model

```

1  `timescale 1ps/10fs
2  module inv(a, q, xin, xout, dcin, dc当地);
3    input a;
4    output q;
5    inout xin, xout, dcin, dc当地;
6    reg q;
7    parameter pul_wid = 100;
8    wire not_a;
9
10   assign not_a = !a;
11
12   biasDir_b I0(xin, xout, dcin, dc当地, gatex);
13
14   initial begin
15
16     $timeformat(-12, 1, "ps", 8); // time format
17
18     // output register initialization
19     q = 1'b0;
20
21   end // initialization
22
23   specify
24     specparam a_xin    = 5;
25     specparam xin_a    = 50;
26
27     $setup(posedge a && a, posedge gatex, a_xin);
28     $setup(negedge a && not_a, posedge gatex, a_xin);
29     $hold(posedge gatex, negedge a && a, xin_a);
30     $hold(posedge gatex, posedge a && not_a, xin_a);
31   endspecify
32
33   always @(posedge gatex)
34     begin

```

```

35      if (a == 1 | a == 0)
36        begin
37          q <= ~a;
38          q <= #pul_wid 1'bz;
39        end
40      else
41        begin
42          q <= 1'bx;
43          q <= #pul_wid 1'bz;
44        end
45    end
46
47 endmodule

```

Listing 2.4: inv Verilog model code.**Figure 2.10:** inv digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.

Switching energy

Different energy consumption results based on different data input patterns ($a = 0$ and $a = 1$) and clock frequencies.

Table 2.4: inv switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
‘0’ (J)	1.78E-23	3.57E-23	8.99E-23	1.85E-22	4.50E-22	1.78E-21	5.18E-21
‘1’ (J)	1.78E-23	3.57E-23	8.97E-23	1.84E-22	4.45E-22	1.75E-21	5.06E-21

2.1.3 AND2

The 2-input AQFP AND cell is built from buffer and constant 0 cells. The operational principle is to merge the output of two buffers and a constant 0 cell through a 3-to-1 branch. There are 4 types of AND gate with different inputs (positive and negative): `and2_pp`, `and2_pn`, `and2_np`, `and2_nn`. Here we present the AND gate with two positive inputs named `and2_pp`. The branch and constant 0 cells are introduced in sub-cells.

Symbol

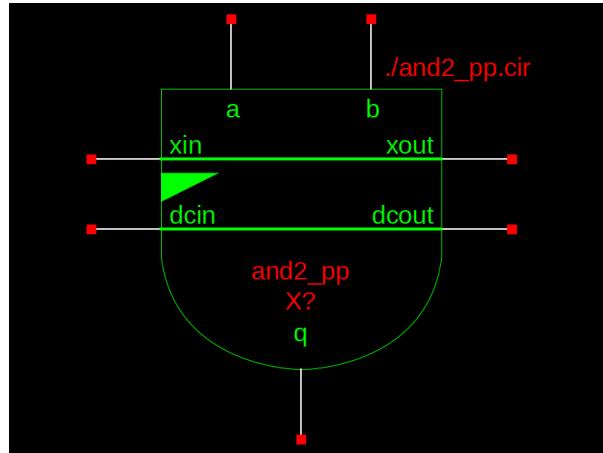


Figure 2.11: and2_pp symbol.

Table 2.5: and2_pp pin list.

Pin	Description
A	data input
B	data input
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

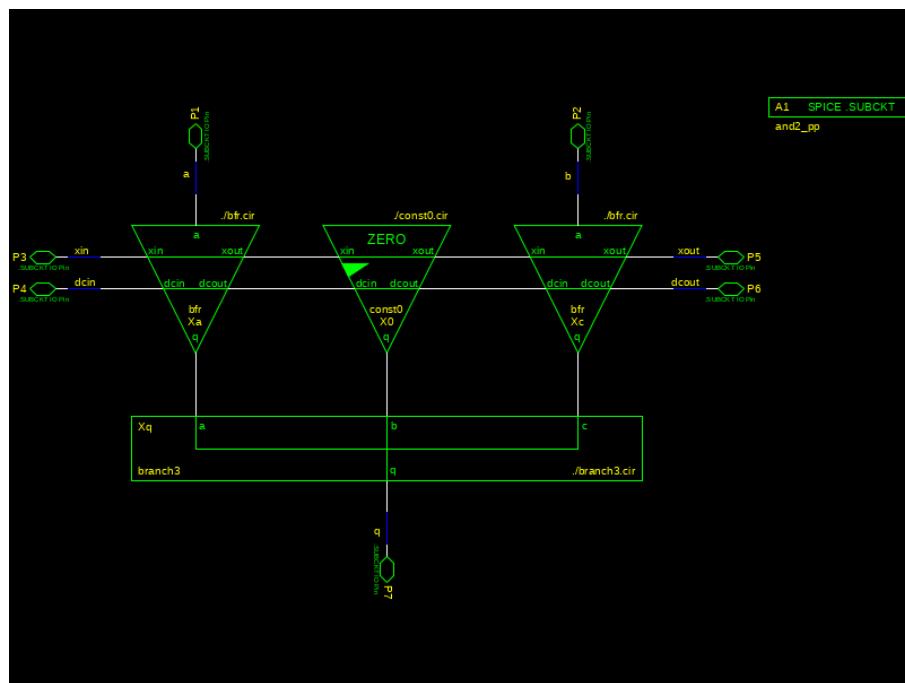


Figure 2.12: `and2_pp` schematic.

Layout

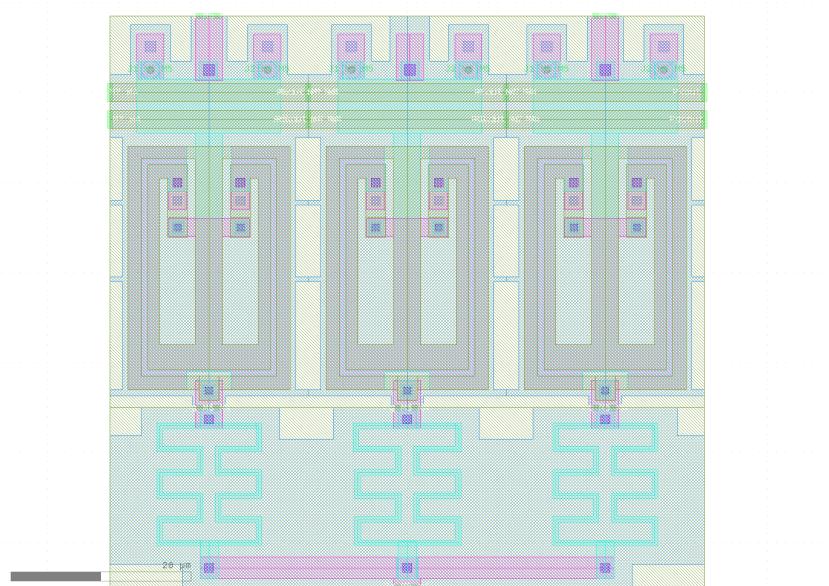


Figure 2.13: `and2_pp` layout.

Analog model

```

1 .SUBCKT bfr a xin dcin xout dcout q
2 Kid Lin Ld Kid
3 Kiout Lin Lout Kiout
4 Kix Lin Lx Kix
5 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
6 .param Lin=1.56p
7 .param Lx=6.53p
8 .param Ld=6.56p
9 .param L1=1.56p
10 .param L2=L1
11 .param Lq=8.25p
12 .param Lout=30.55p
13 .param B1=0.5
14 .param B2=B1
15 .param Kxd=0.2916
16 .param Kx1=-0.2461
17 .param Kx2=Kx1
18 .param Kd1=-0.1916
19 .param Kd2=Kd1
20 .param Kout=-0.6400
21 .param Kxout=-8.116E-6
22 .param Kdout=-2.873E-5
23 .param Kiout=2.201E-3
24 .param Kix=-1.145E-4
25 .param Kid=-1.063E-5
26
27 Kxout Lx Lout Kxout
28 Kdout Ld Lout Kdout
29 Kout Lout Lq Kout
30 Kd2 Ld L2 Kd2
31 Kx2 Lx L2 Kx2
32 Kd1 Ld L1 Kd1
33 Kx1 Lx L1 Kx1
34 Kxd Lx Ld Kxd
35 Lin a 2 Lin
36 B2 3 0 B2 jjmit area=B2
37 B1 1 0 B1 jjmit area=B1
38 Lout 0 q Lout
39 Lq 2 0 Lq
40 L2 3 2 L2
41 L1 2 1 L1
42 Ld dcin dcout Ld
43 Lx xin xout Lx
44 .ends bfr
45 .SUBCKT branch3 a b c q
46 .param La=13.60p
47 .param Lb=10.30p
48 .param Lc=13.60p
49 .param Lq=0.28p
50 Lq 1 q Lq
51 Lc c 1 Lc
52 Lb b 1 Lb
53 La a 1 La
54 .ends branch3
55 .SUBCKT const0 xin dcin xout dcout q
56 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
57 .param Lx=6.62p
58 .param Ld=6.60p
59 .param L1=1.61p
60 .param L2=1.76p
61 .param Lq=8.19p
62 .param Lout=30.65p
63 .param B1=0.5
64 .param B2=B1
65 .param Kxd=0.2972
66 .param Kx1=-0.2265
67 .param Kx2=-0.2602
68 .param Kd1=-0.1688
69 .param Kd2=-0.2088
70 .param Kout=-0.6452

```

```

71 .param Kxout=-2.545E-4
72 .param Kdout=-5.447E-4
73 Kxout Lx Lout Kxout
74 Kdout Ld Lout Kdout
75 Kout Lout Lq Kout
76 Kd2 Ld L2 Kd2
77 Kx2 Lx L2 Kx2
78 Kd1 Ld L1 Kd1
79 Kx1 Lx L1 Kx1
80 Kxd Lx Ld Kxd
81 B2 3 0 B2 jjmit area=B2
82 B1 1 0 B1 jjmit area=B1
83 Lout 0 q Lout
84 Lq 2 0 Lq
85 L2 3 2 L2
86 L1 2 1 L1
87 Ld dcin dcout Ld
88 Lx xin xout Lx
89 .ends const0
90 .SUBCKT and2_pp a b xin dcin xout dcout q
91 X0 2 3 5 6 7 const0
92 Xq 1 7 4 q branch3
93 Xc b 5 6 xout dcout 4 bfr
94 Xa a xin dcin 2 3 1 bfr
95 .ends and2_pp

```

Listing 2.5: and2_pp netlist (.cir).

Simulation result

Simulation waveform of an AND (and2_pp) DUT (design under test) with two 4-stage buffers before the input ‘a’ and ‘b’. Another 3-stage buffer is placed after the AND’s output ‘q’. Clock frequency is 5 GHz. Signals from top to bottom: buffer chain input (a) as 01010101 and (b) as 00110011 (inputs of the first stage buffers), AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), and the output generated from the final stage of buffer (q) as 1100010001 with two random initial outputs 11. This is because of the meander structure of AQFP circuits. In an 8-stage AQFP circuit, it takes two clock cycles to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

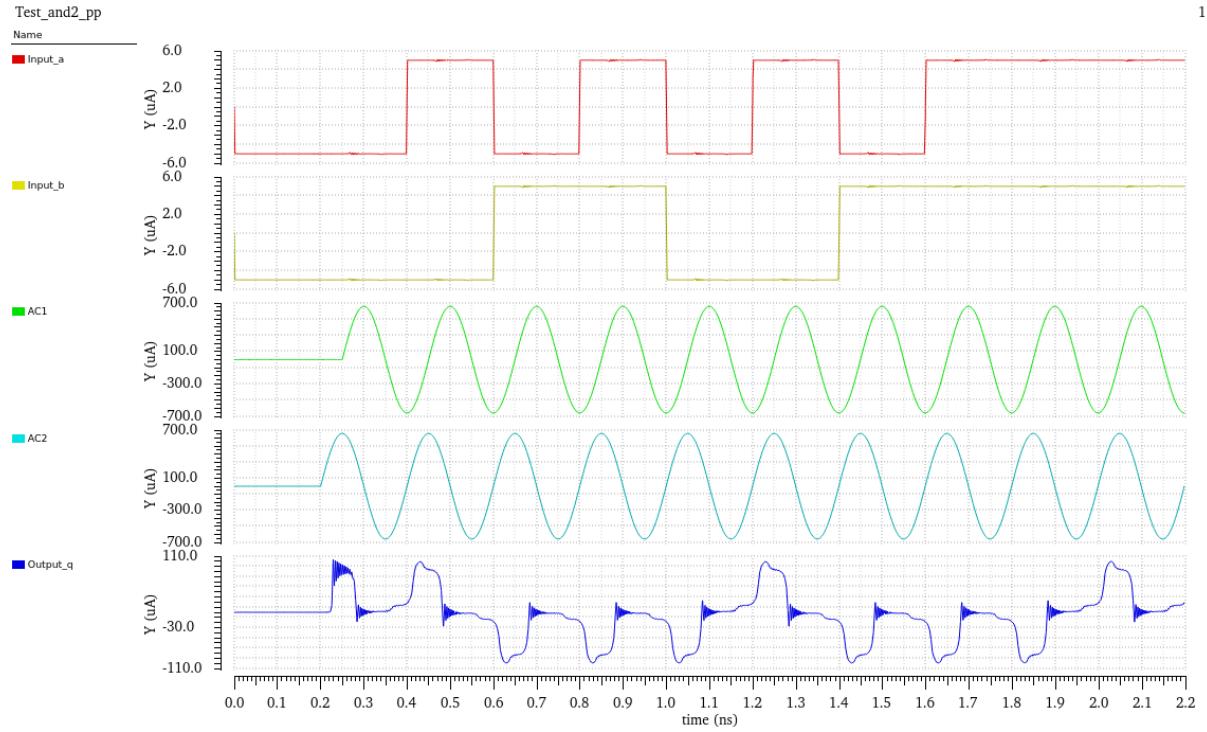


Figure 2.14: `and2_pp` analog waveform.

Digital model

```

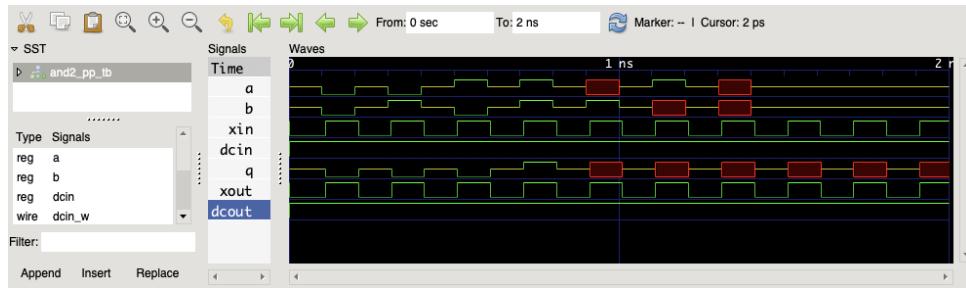
1  `timescale 1ps/10fs
2  module and2_pp(a, b, c, xin, xout, dcin, dc当地);
3    input a, b;
4    output q;
5    inout xin, xout, dcin, dc当地;
6    reg q;
7    parameter pul_wid = 100;
8    wire not_a, not_b;
9
10   assign not_a = !a;
11   assign not_b = !b;
12
13
14   biasDir_b I0(xin, xout, dcin, dc当地, gatex);
15
16   initial begin
17
18     $timeformat(-12, 1, "_ps", 8); // time format
19
20     // output register initialization
21     q = 1'b0;
22   end // initialization
23
24   specify
25     specparam a_xin    = 5;
26     specparam xin_a    = 50;
27
28     $setup(posedge a && a, posedge gatex, a_xin);
29     $setup(negedge a && not_a, posedge gatex, a_xin);
30     $hold(posedge gatex, negedge a && a, xin_a);
31     $hold(posedge gatex, posedge a && not_a, xin_a);
32
33     $setup(posedge b && b, posedge gatex, a_xin);
34     $setup(negedge b && not_b, posedge gatex, a_xin);

```

```

35     $hold(posedge gatex, negedge b && b, xin_a);
36     $hold(posedge gatex, posedge b && not_b, xin_a);
37   endspecify
38
39   always @(posedge gatex)
40   begin
41     if ((a==1'bz) & (b==1'bz) & (a==1'bx) & (b==1'bx))
42       begin
43         q <= a & b;
44         q <= #pul_wid 1'bz;
45       end
46     else
47       begin
48         q <= 1'bx;
49         q <= #pul_wid 1'bz;
50       end
51   end
52
53 endmodule

```

Listing 2.6: and2_pp Verilog model code.**Figure 2.15:** and2_pp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.

Switching energy

Different energy consumption results based on different data input patterns ($ab = 00$, $ab = 01$, $ab = 10$ and $ab = 11$) and clock frequencies.

Table 2.6: and2_pp switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
‘00’ (J)	2.85E-23	6.31E-23	1.69E-22	3.68E-22	9.28E-22	3.93E-21	1.31E-20
‘01’ (J)	5.89E-21	5.76E-21	5.38E-21	4.95E-21	5.45E-21	1.07E-20	1.70E-20
‘10’ (J)	5.92E-21	5.79E-21	5.41E-21	4.98E-21	5.47E-21	1.07E-20	1.71E-20
‘11’ (J)	1.84E-21	1.67E-21	1.65E-21	2.24E-21	3.51E-21	7.10E-21	1.38E-20

2.1.4 AND3

AQFP AND cells are built from buffer cells and constant cells. The operational principle of a 3-input AND gate is to merge the output of three buffers and 2 constant ‘0’ cells through a 5-to-1 branch. There are 8 possible types of 3-input AND gate with different inputs (positive and negative). Here we only present the 3-input AND gate with three positive inputs named `and3_ppp`. Others are under development. The branch cells are introduced in the sub-cells.

Symbol

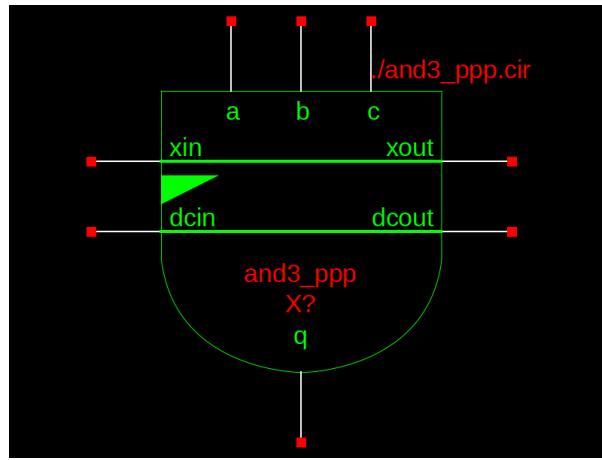


Figure 2.16: `and3_ppp` symbol.

Table 2.7: `and3_ppp` pin list.

Pin	Description
A	data input
B	data input
C	data input
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

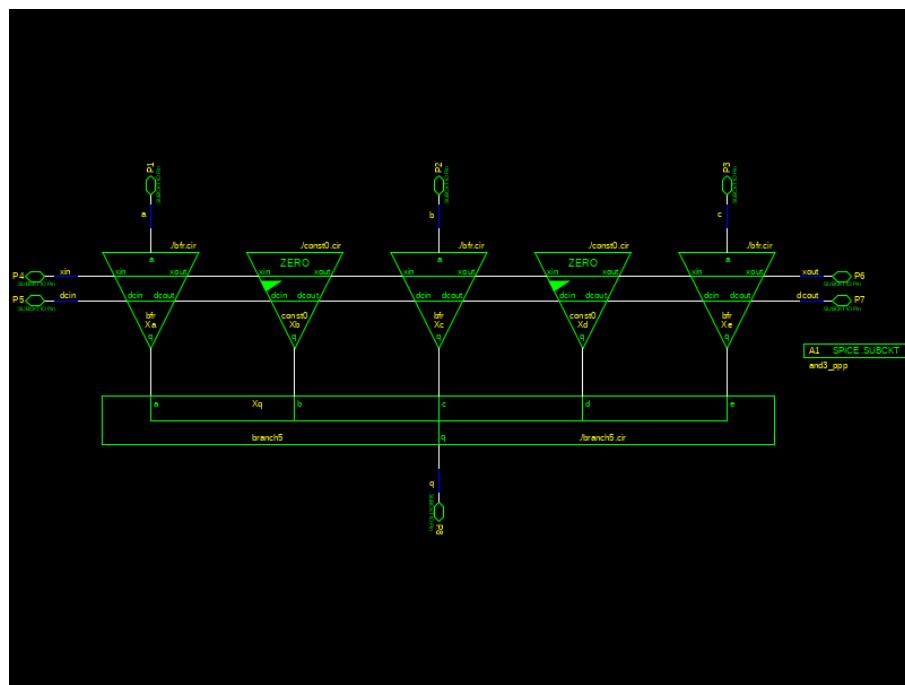


Figure 2.17: and3_ppp schematic.

Layout

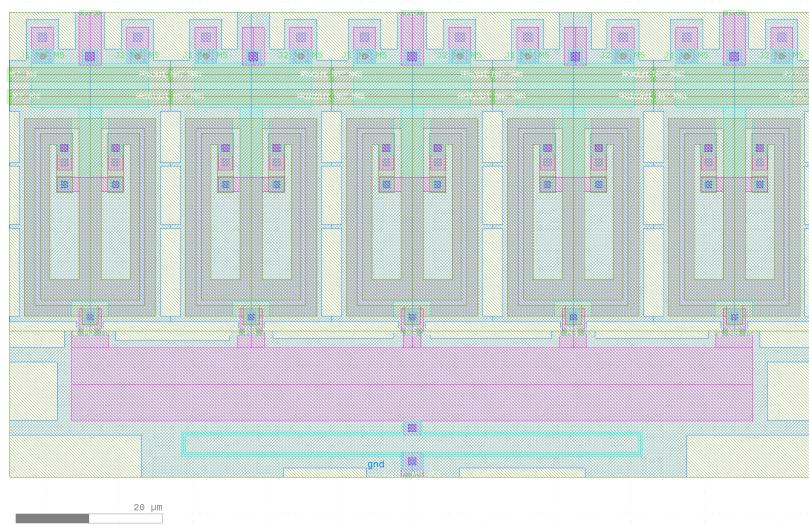


Figure 2.18: and3_ppp layout.

Analog model

```

1 .SUBCKT branch5 a b c d e q
2 L2 2 3 L2
3 L1 1 2 L1
4 Le e 3 Lc
5 Ld d 3 Ld
6 .param La=1.53p
7 .param Lb=0.47p
8 .param Lc=0.67p
9 .param Ld=0.47p
10 .param Le=1.53p
11 .param L1=1.11p
12 .param L2=1.11p
13 .param Lq=18.33p
14 Lq 2 q Lq
15 Lc c 2 Lc
16 Lb b 1 Lb
17 La a 1 La
18 .ends branch5
19 .SUBCKT bfr a xin dcin xout dcout q
20 Kid Lin Ld Kid
21 Kiout Lin Lout Kiout
22 Kix Lin Lx Kix
23 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
24 .param Lin=1.56p
25 .param Lx=6.53p
26 .param Ld=6.56p
27 .param L1=1.56p
28 .param L2=L1
29 .param Lq=8.25p
30 .param Lout=30.55p
31 .param B1=0.5
32 .param B2=B1
33 .param Kxd=0.2916
34 .param Kx1=-0.2461
35 .param Kx2=Kx1
36 .param Kd1=-0.1916
37 .param Kd2=Kd1
38 .param Kout=-0.6400
39 .param Kxout=-8.116E-6
40 .param Kdout=-2.873E-5
41 .param Kiout=2.201E-3
42 .param Kix=-1.145E-4
43 .param Kid=-1.063E-5
44
45 Kxout Lx Lout Kxout
46 Kdout Ld Lout Kdout
47 Kout Lout Lq Kout
48 Kd2 Ld L2 Kd2
49 Kx2 Lx L2 Kx2
50 Kd1 Ld L1 Kd1
51 Kx1 Lx L1 Kx1
52 Kxd Lx Ld Kxd
53 Lin a 2 Lin
54 B2 3 0 B2 jjmit area=B2
55 B1 1 0 B1 jjmit area=B1
56 Lout 0 q Lout
57 Lq 2 0 Lq
58 L2 3 2 L2
59 L1 2 1 L1
60 Ld dcin dcout Ld
61 Lx xin xout Lx
62 .ends bfr
63 .SUBCKT const0 xin dcin xout dcout q
64 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
65 .param Lx=6.62p
66 .param Ld=6.60p
67 .param L1=1.61p
68 .param L2=1.76p
69 .param Lq=8.19p
70 .param Lout=30.65p

```

```

71 .param B1=0.5
72 .param B2=B1
73 .param Kxd=0.2972
74 .param Kx1=-0.2265
75 .param Kx2=-0.2602
76 .param Kd1=-0.1688
77 .param Kd2=-0.2088
78 .param Kout=-0.6452
79 .param Kxout=-2.545E-4
80 .param Kdout=-5.447E-4
81 Kxout Lx Lout Kxout
82 Kdout Ld Lout Kdout
83 Kout Lout Lq Kout
84 Kd2 Ld L2 Kd2
85 Kx2 Lx L2 Kx2
86 Kd1 Ld L1 Kd1
87 Kx1 Lx L1 Kx1
88 Kxd Lx Ld Kxd
89 B2 3 0 B2 jjmit area=B2
90 B1 1 0 B1 jjmit area=B1
91 Lout 0 q Lout
92 Lq 2 0 Lq
93 L2 3 2 L2
94 L1 2 1 L1
95 Ld dcin dcout Ld
96 Lx xin xout Lx
97 .ends const0
98 .SUBCKT and3_ppp a b c xin dcin xout dcout q
99 Xd 10 11 8 9 4 const0
100 Xb 6 7 12 13 2 const0
101 Xc b 12 13 10 11 3 bfr
102 Xe c 8 9 xout dcout 5 bfr
103 Xa a xin dcin 6 7 1 bfr
104 Xq 1 2 3 4 5 q branch5
105 .ends and3_ppp

```

Listing 2.7: and3_ppp netlist (.cir).

Simulation result

Simulation waveform of a 3-input AND (and3_ppp) DUT (design under test) with three 4-stage buffers before the input ‘a’, ‘b’ and ‘c’. Another 3-stage buffer is placed after the AND’s output ‘q’. Clock frequency is 5 GHz. Signals from top to bottom: buffer chain input (a) as 01010101, (b) as 00110011, (c) as 00001111 (inputs of the first stage buffers), AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), and the output generated from the final stage of buffer (q) as 1100000001 with two random initial outputs 11. This is because of the meander structure of AQFP circuits. In an 8-stage AQFP circuit, it takes two clock cycles to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

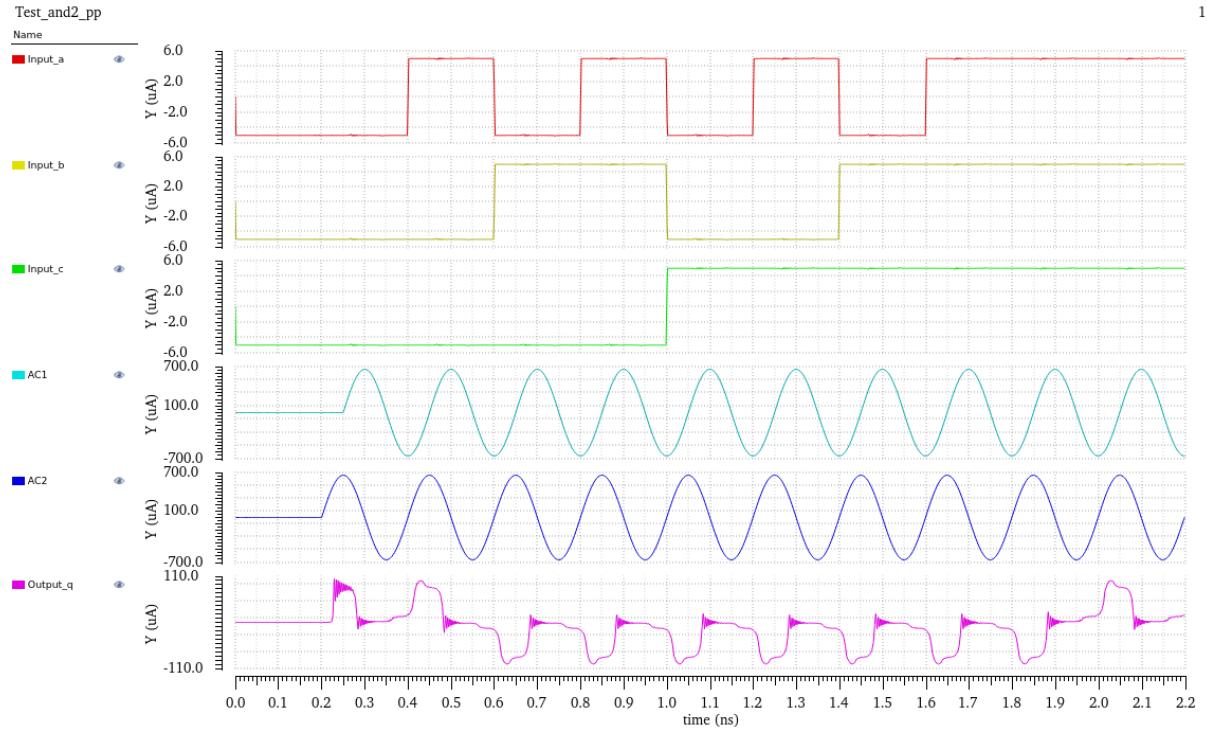


Figure 2.19: and3_ppp analog waveform.

Digital model

```

1  'timescale 1ps/10fs
2  module and3_ppp(a, b, c, q, xin, xout, dcin, dc当地);
3    input a, b, c;
4    output q;
5    inout xin, xout, dcin, dc当地;
6    reg q;
7    parameter pul_wid = 100;
8    wire [2:0] in_and3, neg_in_and3;
9
10   assign in_and3 = {a, b, c};
11  assign neg_in_and3 = ~in_and3;
12
13  biasDir_b I0(xin, xout, dcin, dc当地, gatex);
14
15  initial begin
16
17    $timeformat(-12, 1, "_ps", 8); // time format
18
19    // output register initialization
20    q = 1'b0;
21
22  end // initialization
23
24  specify
25    specparam d_clk      = 5;
26    specparam clk_d     = 50;
27
28    $setup(posedge in_and3[2] && in_and3[2], posedge gatex, d_clk);
29    $setup(negedge in_and3[2] && neg_in_and3[2], posedge gatex, d_clk);
30    $hold(posedge gatex, negedge in_and3[2] && in_and3[2], clk_d);
31    $hold(posedge gatex, posedge in_and3[2] && neg_in_and3[2], clk_d);
32
33    $setup(posedge in_and3[1] && in_and3[1], posedge gatex, d_clk);
34    $setup(negedge in_and3[1] && neg_in_and3[1], posedge gatex, d_clk);

```

```

35   $hold($posedge gatex, $negedge in_and3[1] && in_and3[1], clk_d);
36   $hold($posedge gatex, $posedge in_and3[1] && neg_in_and3[1], clk_d);
37
38   $setup($posedge in_and3[0] && in_and3[0], $posedge gatex, d_clk);
39   $setup($negedge in_and3[0] && neg_in_and3[0], $posedge gatex, d_clk);
40   $hold($posedge gatex, $negedge in_and3[0] && in_and3[0], clk_d);
41   $hold($posedge gatex, $posedge in_and3[0] && neg_in_and3[0], clk_d);
42
43 endspecify
44
45 always @(posedge gatex)
46 begin
47   if ((&in_and3 != 1'bx) & (&in_and3 != 1'bz))
48     begin
49       q <= &in_and3;
50       q <= #pul_wid 1'bz;
51     end
52   else
53     begin
54       q <= 1'bx;
55       q <= #pul_wid 1'bz;
56     end
57   end
58 endmodule

```

Listing 2.8: and3_ppp Verilog model code.

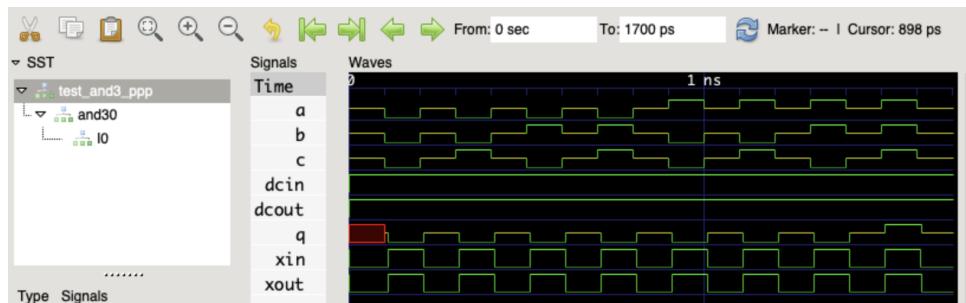


Figure 2.20: and3_ppp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.

Switching energy

Different energy consumption results based on different data input patterns ($abc = 000$, $abc = 001$, $abc = 010$, $abc = 011$, $abc = 100$, $abc = 101$, $abc = 110$ and $abc = 111$) and clock frequencies (0.1GHz to 10GHz).

Table 2.8: and3_ppp switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
‘000’ (J)	4.98E-23	1.05E-22	2.73E-22	5.77E-22	1.46E-21	6.40E-21	1.83E-20
‘001’ (J)	8.15E-21	8.06E-21	7.81E-21	7.39E-21	7.54E-21	1.37E-20	2.84E-20
‘010’ (J)	8.63E-21	8.55E-21	8.31E-21	7.89E-21	7.70E-21	1.36E-20	2.87E-20
‘011’ (J)	1.34E-20	1.31E-20	1.24E-20	1.13E-20	1.10E-20	2.15E-20	3.64E-20
‘100’ (J)	8.15E-21	8.06E-21	7.81E-21	7.39E-21	7.54E-21	1.37E-20	2.84E-20
‘101’ (J)	1.32E-20	1.29E-20	1.22E-20	1.11E-20	1.08E-20	2.15E-20	3.61E-20
‘110’ (J)	1.34E-20	1.31E-20	1.24E-20	1.13E-20	1.10E-20	2.15E-20	3.64E-20
‘111’ (J)	1.34E-20	1.20E-21	1.31E-21	2.35E-21	4.21E-21	1.27E-20	2.88E-20

2.1.5 OR2

The 2-input AQFP OR cell is built from buffer and constant 1 cells. The operational principle is to merge the output of two buffers and a constant 1 cell through a 3-to-1 branch. There are 4 types of OR gate with different inputs (positive and negative): `or_pp`, `or_pn`, `or_np`, `or_nn`. Here we only present the OR gate with two positive inputs named `or_pp` as an example. Others can be found in the deliverables. The branch and constant 1 cells are introduced in sub-cells.

Symbol

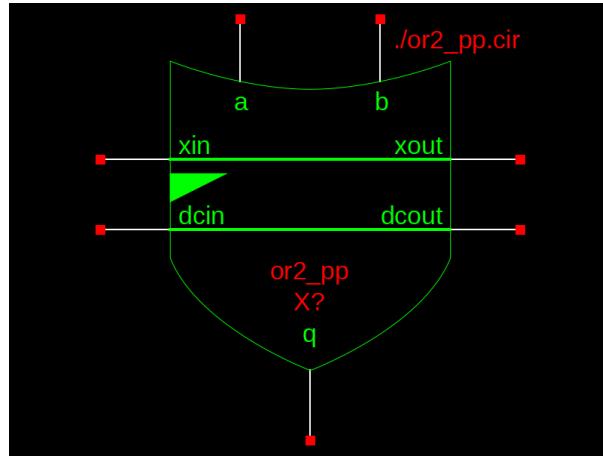


Figure 2.21: `or2_pp` symbol.

Table 2.9: `or2_pp` pin list.

Pin	Description
A	data input
B	data input
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

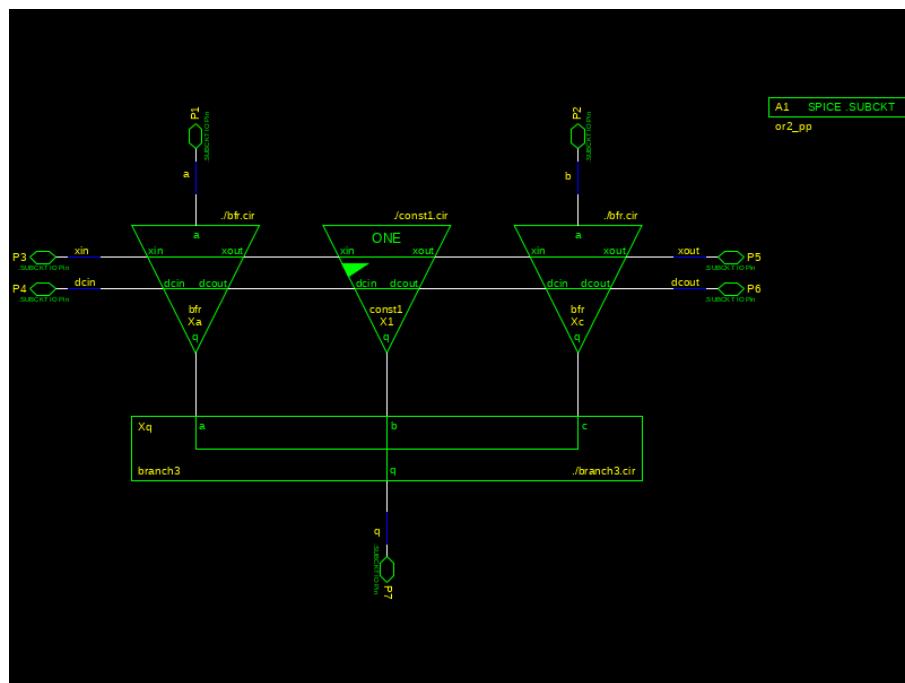


Figure 2.22: `or2_pp` schematic.

Layout

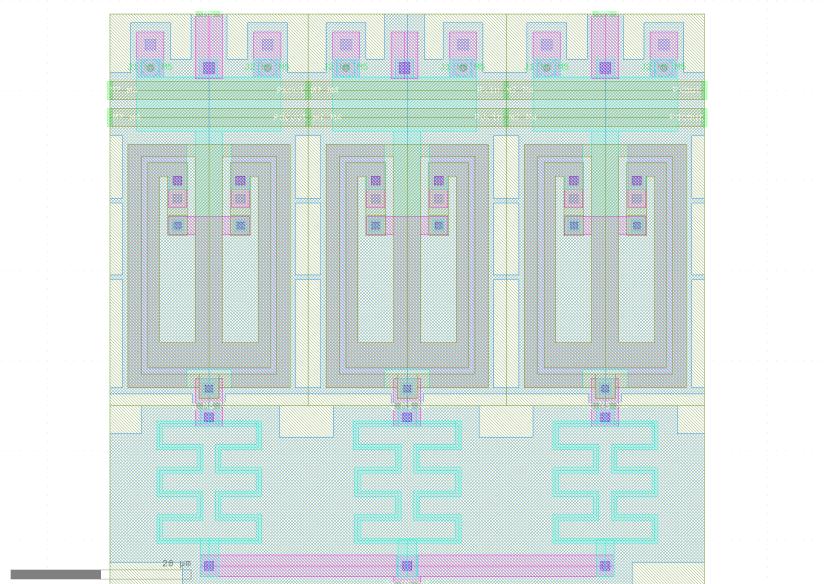


Figure 2.23: `or2_pp` layout.

Analog model

```

1 .SUBCKT bfr a xin dcin xout dcout q
2 Kid Lin Ld Kid
3 Kiout Lin Lout Kiout
4 Kix Lin Lx Kix
5 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
6 .param Lin=1.56p
7 .param Lx=6.53p
8 .param Ld=6.56p
9 .param L1=1.56p
10 .param L2=L1
11 .param Lq=8.25p
12 .param Lout=30.55p
13 .param B1=0.5
14 .param B2=B1
15 .param Kxd=0.2916
16 .param Kx1=-0.2461
17 .param Kx2=Kx1
18 .param Kd1=-0.1916
19 .param Kd2=Kd1
20 .param Kout=-0.6400
21 .param Kxout=-8.116E-6
22 .param Kdout=-2.873E-5
23 .param Kiout=2.201E-3
24 .param Kix=-1.145E-4
25 .param Kid=-1.063E-5
26
27 Kxout Lx Lout Kxout
28 Kdout Ld Lout Kdout
29 Kout Lout Lq Kout
30 Kd2 Ld L2 Kd2
31 Kx2 Lx L2 Kx2
32 Kd1 Ld L1 Kd1
33 Kx1 Lx L1 Kx1
34 Kxd Lx Ld Kxd
35 Lin a 2 Lin
36 B2 3 0 B2 jjmit area=B2
37 B1 1 0 B1 jjmit area=B1
38 Lout 0 q Lout
39 Lq 2 0 Lq
40 L2 3 2 L2
41 L1 2 1 L1
42 Ld dcin dcout Ld
43 Lx xin xout Lx
44 .ends bfr
45 .SUBCKT branch3 a b c q
46 .param La=13.60p
47 .param Lb=10.30p
48 .param Lc=13.60p
49 .param Lq=0.28p
50 Lq 1 q Lq
51 Lc c 1 Lc
52 Lb b 1 Lb
53 La a 1 La
54 .ends branch3
55 .SUBCKT const0 xin dcin xout dcout q
56 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
57 .param Lx=6.62p
58 .param Ld=6.60p
59 .param L1=1.61p
60 .param L2=1.76p
61 .param Lq=8.19p
62 .param Lout=30.65p
63 .param B1=0.5
64 .param B2=B1
65 .param Kxd=0.2972
66 .param Kx1=-0.2265
67 .param Kx2=-0.2602
68 .param Kd1=-0.1688
69 .param Kd2=-0.2088
70 .param Kout=-0.6452

```

```

71  .param Kxout=-2.545E-4
72  .param Kdout=-5.447E-4
73  Kxout Lx Lout Kxout
74  Kdout Ld Lout Kdout
75  Kout Lout Lq Kout
76  Kd2 Ld L2 Kd2
77  Kx2 Lx L2 Kx2
78  Kd1 Ld L1 Kd1
79  Kx1 Lx L1 Kx1
80  Kxd Lx Ld Kxd
81  B2 3 0 B2 jjmit area=B2
82  B1 1 0 B1 jjmit area=B1
83  Lout 0 q Lout
84  Lq 2 0 Lq
85  L2 3 2 L2
86  L1 2 1 L1
87  Ld dcin dcout Ld
88  Lx xin xout Lx
89  .ends const0
90  .SUBCKT const1 xin dcin xout dcout q
91  X0 xout dcout xin dcin q const0
92  .ends const1
93  .SUBCKT or2_pp a b xin dcin xout dcout q
94  X1 2 3 5 6 7 const1
95  Xq 1 7 4 q branch3
96  Xc b 5 6 xout dcout 4 bfr
97  Xa a xin dcin 2 3 1 bfr
98  .ends or2_pp

```

Listing 2.9: or2_pp netlist (.cir).

Simulation result

Simulation waveform of an 2-input OR (or2_pp) DUT (design under test) with two 4-stage buffers before the input ‘a’ and ‘b’. Another 3-stage buffer is placed after the AND’s output ‘q’. Clock frequency is 5 GHz. Signals from top to bottom: buffer chain input (a) as 01010101 and (b) as 00110011 (inputs of the first stage buffers), AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), and the output generated from the final stage of buffer (q) as 0001110111 with two random initial outputs 00. This is because of the meander structure of AQFP circuits. In an 8-stage AQFP circuit, it takes two clock cycles to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

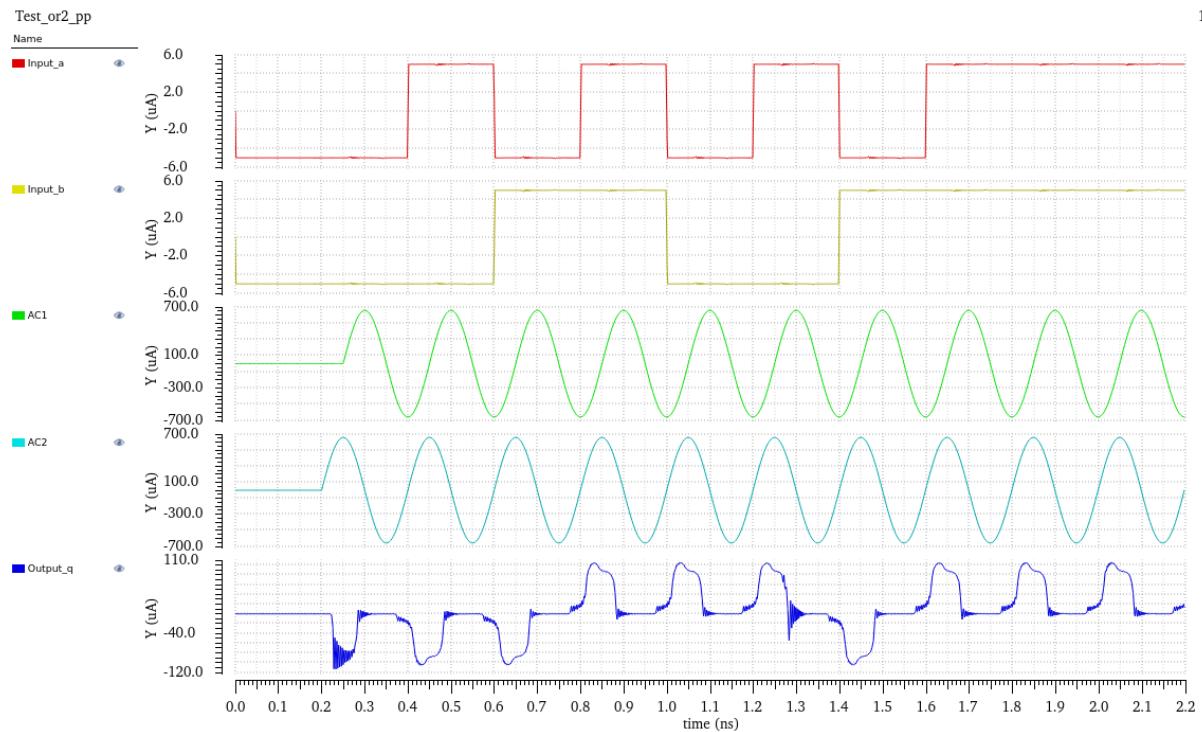


Figure 2.24: or2_pp analog waveform.

Digital model

```

1  `timescale 1ps/10fs
2
3
4  input a, b;
5  output q;
6  inout xin, xout, dcin, dc当地;
7  reg q;
8  parameter pul_wid = 100;
9  wire not_a, not_b;
10
11 assign not_a = ~a;
12 assign not_b = ~b;
13
14 biasDir_b I0(xin, xout, dcin, dc当地, gatex);
15
16 initial begin
17
18 $timeformat(-12, 1, "_ps", 8); // time format
19
20 // output register initialization
21 q = 1'b0;
22 end // initialization
23
24 specify
25   specparam d_clk    = 5;
26   specparam clk_d   = 50;
27
28   $setup(posedge a && a, posedge gatex, d_clk);
29   $setup(negedge a && not_a, posedge gatex, d_clk);
30   $hold(posedge gatex, negedge a && a, clk_d);
31   $hold(posedge gatex, posedge a && not_a, clk_d);
32
33   $setup(posedge b && b, posedge gatex, d_clk);
34   $setup(negedge b && not_b, posedge gatex, d_clk);

```

```

35     $hold(posedge gatex, negedge b && b, clk_d);
36     $hold(posedge gatex, posedge b && not_b, clk_d);
37 endspecify
38
39 always @(posedge gatex)
40 begin
41   if ((a==1'bz) & (b!=1'bz) & (a==1'bx) & (b==1'bx))
42     begin
43       q <= a | b;
44       q <= #pul_wid 1'bz;
45     end
46   else
47     begin
48       q <= 1'bx;
49       q <= #pul_wid 1'bz;
50     end
51 end
52
53 endmodule

```

Listing 2.10: or2_pp Verilog model code.**Figure 2.25:** or2_pp digital waveform. HDL '1': AQFP '1'; HDL '0': AQFP '0'; HDL 'z': inactive; HDL 'x': error.

Switching energy

Different energy consumption results based on different data input patterns ($ab = 00$, $ab = 01$, $ab = 10$ and $ab = 11$) and clock frequencies.

Table 2.10: or2_pp switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
'00' (J)	1.84E-21	1.67E-21	1.65E-21	2.24E-21	3.51E-21	7.10E-21	1.38E-20
'01' (J)	5.92E-21	5.79E-21	5.41E-21	4.98E-21	5.47E-21	1.07E-20	1.71E-20
'10' (J)	5.89E-21	5.76E-21	5.38E-21	4.95E-21	5.45E-21	1.07E-20	1.70E-20
'11' (J)	2.87E-23	6.33E-23	1.69E-22	3.68E-22	9.29E-22	3.93E-21	1.31E-20

2.1.6 OR3

AQFP OR cells are built from buffer cells and constant cells. The operational principle of a 3-input OR gate is to merge the output of three buffers and 2 constant ‘1’ cells through a 5-to-1 branch. There are 8 possible types of 3-input OR gate with different inputs (positive and negative). Here we only present the 3-input OR gate with three positive inputs named `or3_ppp`. Others are under development. The branch cells are introduced in the sub-cells.

Symbol

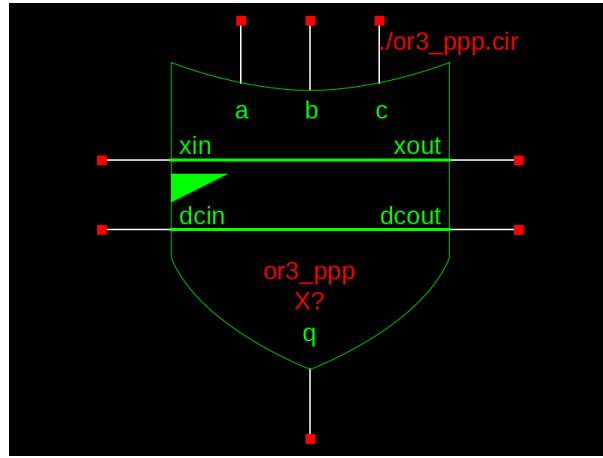


Figure 2.26: `or3_ppp` symbol.

Table 2.11: `or3_ppp` pin list.

Pin	Description
A	data input
B	data input
C	data input
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

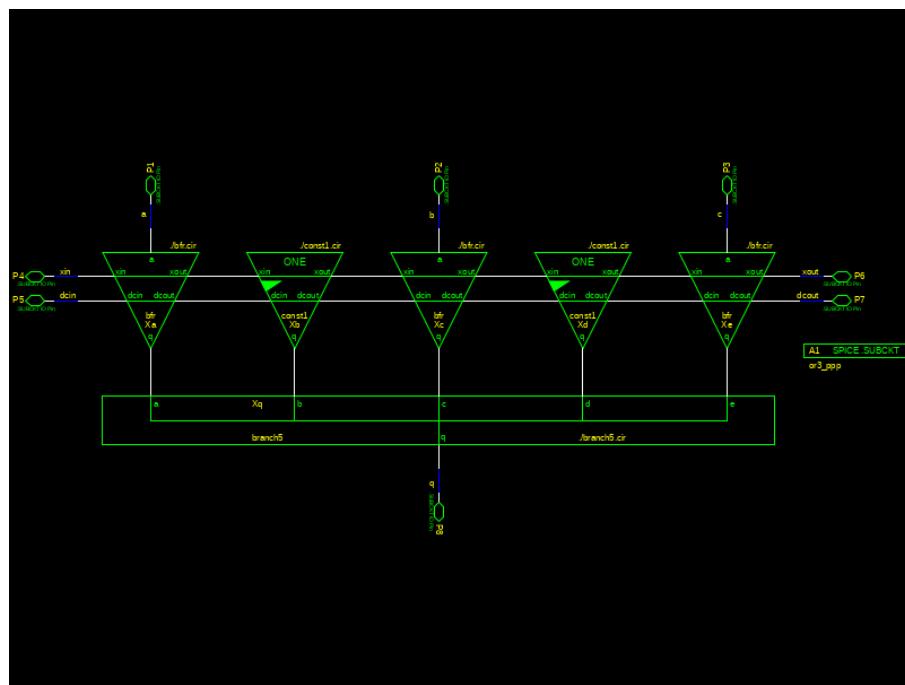


Figure 2.27: or3_ppp schematic.

Layout

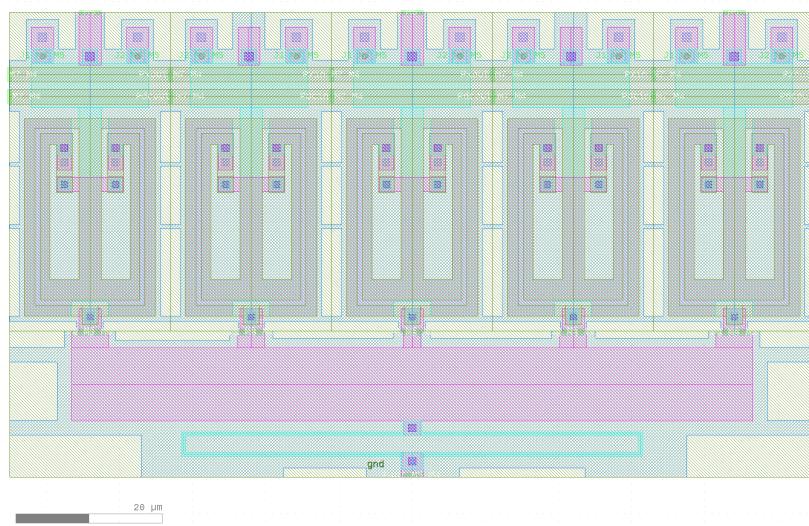


Figure 2.28: or3_ppppp layout.

Analog model

```

1 .SUBCKT branch5 a b c d e q
2 L2 2 3 L2
3 L1 1 2 L1
4 Le e 3 Lc
5 Ld d 3 Ld
6 .param La=1.53p
7 .param Lb=0.47p
8 .param Lc=0.67p
9 .param Ld=0.47p
10 .param Le=1.53p
11 .param L1=1.11p
12 .param L2=1.11p
13 .param Lq=18.33p
14 Lq 2 q Lq
15 Lc c 2 Lc
16 Lb b 1 Lb
17 La a 1 La
18 .ends branch5
19 .SUBCKT bfr a xin dcin xout dcout q
20 Kid Lin Ld Kid
21 Kiout Lin Lout Kiout
22 Kix Lin Lx Kix
23 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
24 .param Lin=1.56p
25 .param Lx=6.53p
26 .param Ld=6.56p
27 .param L1=1.56p
28 .param L2=L1
29 .param Lq=8.25p
30 .param Lout=30.55p
31 .param B1=0.5
32 .param B2=B1
33 .param Kxd=0.2916
34 .param Kx1=-0.2461
35 .param Kx2=Kx1
36 .param Kd1=-0.1916
37 .param Kd2=Kd1
38 .param Kout=-0.6400
39 .param Kxout=-8.116E-6
40 .param Kdout=-2.873E-5
41 .param Kiout=2.201E-3
42 .param Kix=-1.145E-4
43 .param Kid=-1.063E-5
44
45 Kxout Lx Lout Kxout
46 Kdout Ld Lout Kdout
47 Kout Lout Lq Kout
48 Kd2 Ld L2 Kd2
49 Kx2 Lx L2 Kx2
50 Kd1 Ld L1 Kd1
51 Kx1 Lx L1 Kx1
52 Kxd Lx Ld Kxd
53 Lin a 2 Lin
54 B2 3 0 B2 jjmit area=B2
55 B1 1 0 B1 jjmit area=B1
56 Lout 0 q Lout
57 Lq 2 0 Lq
58 L2 3 2 L2
59 L1 2 1 L1
60 Ld dcin dcout Ld
61 Lx xin xout Lx
62 .ends bfr
63 .SUBCKT const0 xin dcin xout dcout q
64 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
65 .param Lx=6.62p
66 .param Ld=6.60p
67 .param L1=1.61p
68 .param L2=1.76p
69 .param Lq=8.19p
70 .param Lout=30.65p

```

```

71 .param B1=0.5
72 .param B2=B1
73 .param Kxd=0.2972
74 .param Kx1=-0.2265
75 .param Kx2=-0.2602
76 .param Kd1=-0.1688
77 .param Kd2=-0.2088
78 .param Kout=-0.6452
79 .param Kxout=-2.545E-4
80 .param Kdout=-5.447E-4
81 Kxout Lx Lout Kxout
82 Kdout Ld Lout Kdout
83 Kout Lout Lq Kout
84 Kd2 Ld L2 Kd2
85 Kx2 Lx L2 Kx2
86 Kd1 Ld L1 Kd1
87 Kx1 Lx L1 Kx1
88 Kxd Lx Ld Kxd
89 B2 3 0 B2 jjmit area=B2
90 B1 1 0 B1 jjmit area=B1
91 Lout 0 q Lout
92 Lq 2 0 Lq
93 L2 3 2 L2
94 L1 2 1 L1
95 Ld dcin dcout Ld
96 Lx xin xout Lx
97 .ends const0
98 .SUBCKT const1 xin dcin xout dcout q
99 X0 xout dcout xin dcin q const0
100 .ends const1
101 .SUBCKT or3_ppp a b c xin dcin xout dcout q
102 Xd 10 11 8 9 4 const1
103 Xb 6 7 12 13 2 const1
104 Xc b 12 13 10 11 3 bfr
105 Xe c 8 9 xout dcout 5 bfr
106 Xa a xin dcin 6 7 1 bfr
107 Xq 1 2 3 4 5 q branch5
108 .ends or3_ppp

```

Listing 2.11: or3_ppp netlist (.cir).

Simulation result

Simulation waveform of a 3-input OR (or3_ppp) DUT (design under test) with three 4-stage buffers before the input ‘a’, ‘b’ and ‘c’. Another 3-stage buffer is placed after the OR’s output ‘q’. Clock frequency is 5 GHz. Signals from top to bottom: buffer chain input (a) as 01010101, (b) as 00110011, (c) as 00001111 (inputs of the first stage buffers), AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), and the output generated from the final stage of buffer (q) as 100111111 with two random initial outputs 10. This is because of the meander structure of AQFP circuits. In an 8-stage AQFP circuit, it takes two clock cycles to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

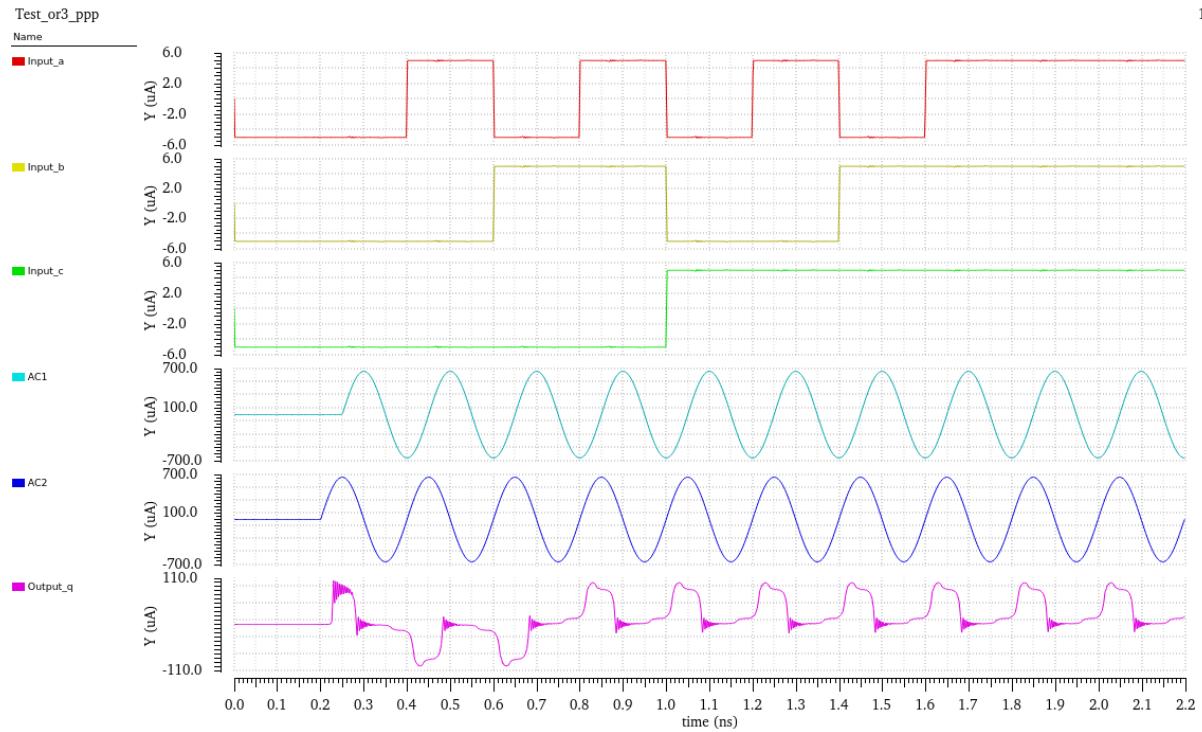


Figure 2.29: or3_ppp analog waveform.

Digital model

```

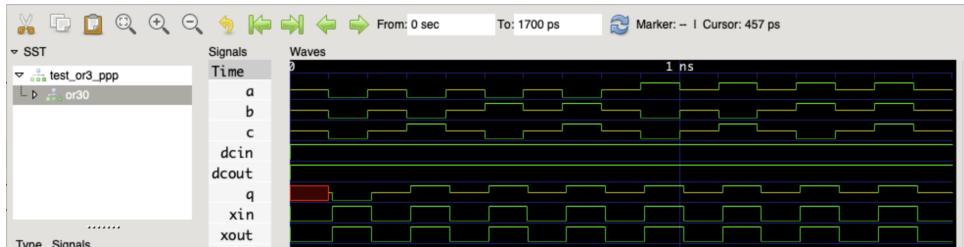
1  `timescale 1ps/10fs
2  module or3_ppp(a, b, c, q, xin, xout, dcin, dcout);
3    input a, b, c;
4    output q;
5    inout xin, xout, dcin, dcout;
6    reg q;
7    parameter pul_wid = 100;
8    wire [2:0] in_or3, neg_in_or3;
10
11   assign in_or3 = {a, b, c};
12   assign neg_in_or3 = ~in_or3;
13
14   biasDir_b I0(xin, xout, dcin, dcout, gatex);
15
16   initial begin
17
18     $timeformat(-12, 1, "_ps", 8); // time format
19
20     // output register initialization
21     q = 1'bz;
22   end // initialization
23
24   specify
25     specparam d_clk    = 5;
26     specparam clk_d   = 50;
27
28     $setup(posedge in_or3[2] && in_or3[2], posedge gatex, d_clk);
29     $setup(negedge in_or3[2] && neg_in_or3[2], posedge gatex, d_clk);
30     $hold(posedge gatex, negedge in_or3[2] && in_or3[2], clk_d);
31     $hold(posedge gatex, posedge in_or3[2] && neg_in_or3[2], clk_d);
32
33     $setup(posedge in_or3[1] && in_or3[1], posedge gatex, d_clk);
34     $setup(negedge in_or3[1] && neg_in_or3[1], posedge gatex, d_clk);

```

```

35     $hold($posedge gatex, $negedge in_or3[1] && in_or3[1], clk_d);
36     $hold($posedge gatex, $posedge in_or3[1] && neg_in_or3[1], clk_d);
37
38     $setup($posedge in_or3[0] && in_or3[0], $posedge gatex, d_clk);
39     $setup($negedge in_or3[0] && neg_in_or3[0], $posedge gatex, d_clk);
40     $hold($posedge gatex, $negedge in_or3[0] && in_or3[0], clk_d);
41     $hold($posedge gatex, $posedge in_or3[0] && neg_in_or3[0], clk_d);
42
43   endspecify
44
45   always @($posedge gatex)
46   begin
47     if ((&in_or3 != 1'bx) & (&in_or3 != 1'bz))
48       begin
49         q <= |in_or3;
50         q <= #pul_wid 1'bz;
51       end
52     else
53       begin
54         q <= 1'bx;
55         q <= #pul_wid 1'bz;
56       end
57   end
58 endmodule

```

Listing 2.12: or3_ppp Verilog model code.**Figure 2.30:** or3_ppp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.

Switching energy

Different energy consumption results based on different data input patterns ($abc = 000, abc = 001, abc = 010, abc = 011, abc = 100, abc = 101, abc = 110$ and $abc = 111$) and clock frequencies (0.1GHz to 10GHz).

Table 2.12: or3_ppp switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
‘000’ (J)	1.34E-20	1.20E-21	1.31E-21	2.35E-21	4.21E-21	1.27E-20	2.88E-20
‘001’ (J)	1.34E-20	1.31E-20	1.24E-20	1.13E-20	1.11E-20	2.15E-20	3.64E-20
‘010’ (J)	1.32E-20	1.29E-20	1.22E-20	1.11E-20	1.08E-20	2.15E-20	3.61E-20
‘011’ (J)	8.15E-21	8.06E-21	7.81E-21	7.39E-21	7.54E-21	1.37E-20	2.84E-20
‘100’ (J)	1.34E-20	1.31E-20	1.24E-20	1.13E-20	1.11E-20	2.15E-20	3.64E-20
‘101’ (J)	9.69E-21	1.22E-20	9.38E-21	8.95E-21	8.77E-21	1.46E-20	2.98E-20
‘110’ (J)	8.15E-21	8.06E-21	7.81E-21	7.39E-21	7.54E-21	1.37E-20	2.84E-20
‘111’ (J)	5.00E-23	1.06E-22	2.73E-22	5.77E-22	1.46E-21	6.40E-21	1.83E-20

2.1.7 MAJ3

AQFP majority cells are built from buffer cells. The operational principle of a 3-input majority gate is to merge the output of three buffers through a 3-to-1 branch. There are 8 types of 3-input majority gate with different inputs (positive and negative): `maj3_ppp`, `maj3_ppn`, `maj3_pnp`, `maj3_pnn`, `maj3_npp`, `maj3_npn`, `maj3_nnp`, `maj3_nnn`. Here we only present the 3-input majority gate with three positive inputs named `maj3_ppp`. Others can be found in the deliverables. The branch cells are introduced in the sub-cells.

Symbol

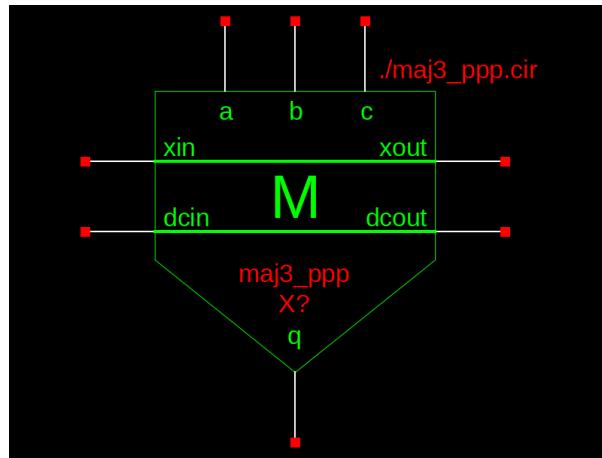


Figure 2.31: `maj3_ppp` symbol.

Table 2.13: `maj3_ppp` pin list.

Pin	Description
A	data input
B	data input
C	data input
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

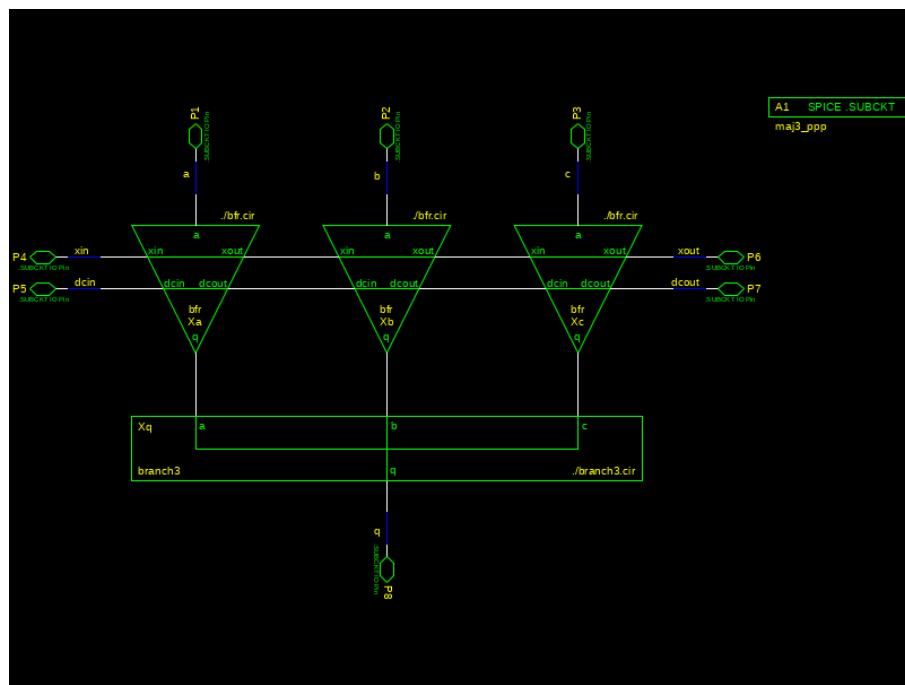


Figure 2.32: `maj3_ppp` schematic.

Layout

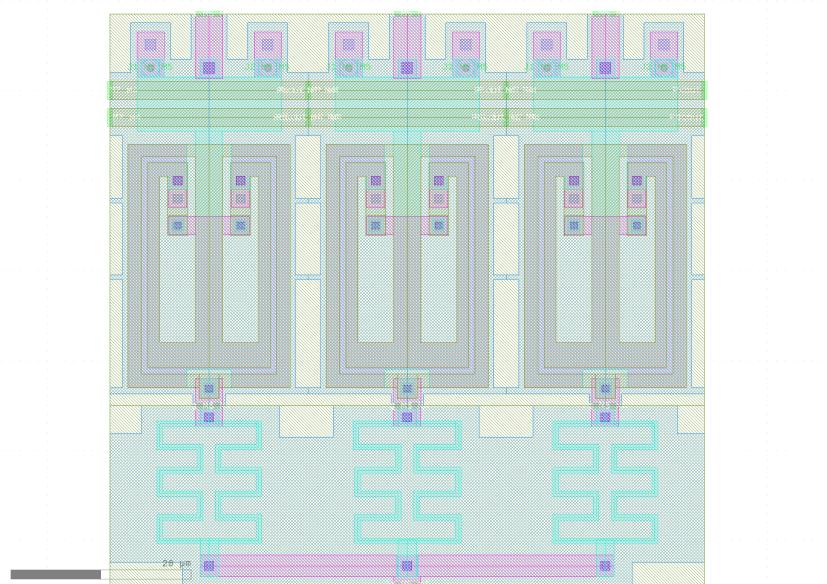


Figure 2.33: `maj3_ppp` layout.

Analog model

```

1 .SUBCKT bfr a xin dcin xout dcout q
2 Kid Lin Ld Kid
3 Kiout Lin Lout Kiout
4 Kix Lin Lx Kix
5 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
6 .param Lin=1.56p
7 .param Lx=6.53p
8 .param Ld=6.56p
9 .param L1=1.56p
10 .param L2=L1
11 .param Lq=8.25p
12 .param Lout=30.55p
13 .param B1=0.5
14 .param B2=B1
15 .param Kxd=0.2916
16 .param Kx1=-0.2461
17 .param Kx2=Kx1
18 .param Kd1=-0.1916
19 .param Kd2=Kd1
20 .param Kout=-0.6400
21 .param Kxout=-8.116E-6
22 .param Kdout=-2.873E-5
23 .param Kiout=2.201E-3
24 .param Kix=-1.145E-4
25 .param Kid=-1.063E-5
26
27 Kxout Lx Lout Kxout
28 Kdout Ld Lout Kdout
29 Kout Lout Lq Kout
30 Kd2 Ld L2 Kd2
31 Kx2 Lx L2 Kx2
32 Kd1 Ld L1 Kd1
33 Kx1 Lx L1 Kx1
34 Kxd Lx Ld Kxd
35 Lin a 2 Lin
36 B2 3 0 B2 jjmit area=B2
37 B1 1 0 B1 jjmit area=B1
38 Lout 0 q Lout
39 Lq 2 0 Lq
40 L2 3 2 L2
41 L1 2 1 L1
42 Ld dcin dcout Ld
43 Lx xin xout Lx
44 .ends bfr
45 .SUBCKT branch3 a b c q
46 .param La=13.60p
47 .param Lb=10.30p
48 .param Lc=13.60p
49 .param Lq=0.28p
50 Lq 1 q Lq
51 Lc c 1 Lc
52 Lb b 1 Lb
53 La a 1 La
54 .ends branch3
55 .SUBCKT maj3_ppp a b c xin dcin xout dcout q
56 Xq 1 4 7 q branch3
57 Xc c 5 6 xout dcout 7 bfr
58 Xb b 2 3 5 6 4 bfr
59 Xa a xin dcin 2 3 1 bfr
60 .ends maj3_ppp

```

Listing 2.13: maj3_ppp netlist (.cir).

Simulation result

Simulation waveform of a 3-input MAJORITY (maj_ppp) DUT (design under test) with three 4-stage buffers before the input ‘a’, ‘b’ and ‘c’. Another 3-stage buffer is placed after the MAJORITY’s output ‘q’. Clock frequency is 5 GHz. Signals from top to bottom:

buffer chain input (a) as 01010101, (b) as 00110011 and (c) as 00001111 (inputs of the first stage buffers), AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), and the output generated from the final stage of buffer (d) as 1000010111 with two random initial outputs 10. This is because of the meander structure of AQFP circuits. In an 8-stage AQFP circuit, it takes two clock cycles to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

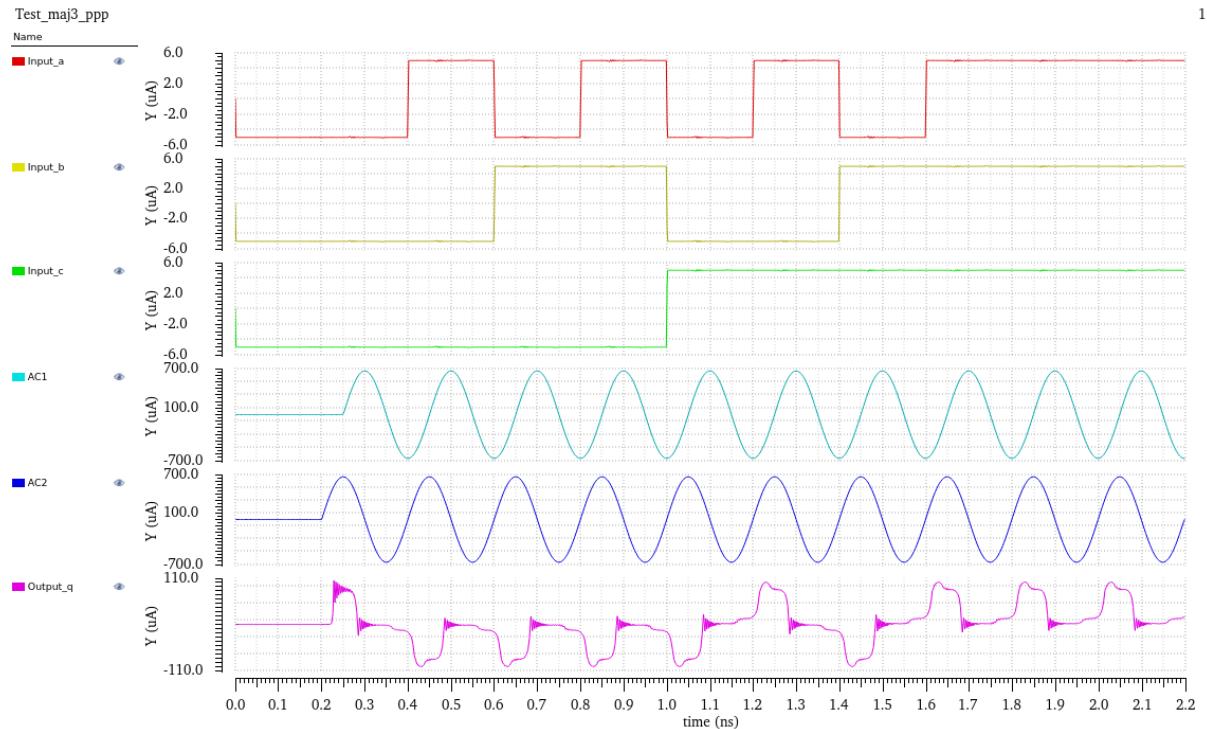


Figure 2.34: maj3_ppp analog waveform.

Digital model

```

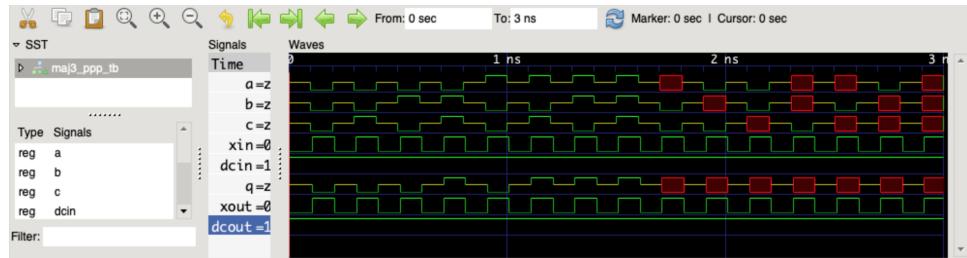
1  `timescale 1ps/10fs
2  module maj3_ppp(a, b, c, q, xin, xout, dcin, dc当地;
3  input a, b, c;
4  output q;
5  inout xin, xout, dcin, dc当地;
6  reg q;
7  parameter pul_wid = 100;
8  wire [2:0] in_maj3, neg_in_maj3;
9
10 assign in_maj3 = {a, b, c};
11 assign neg_in_maj3 = ~in_maj3;
12
13 biasDir_b I0(xin, xout, dcin, dc当地, gate);
14
15 initial begin
16
17 $timeformat(-12, 1, "_ps", 8); // time format
18
19 // output register initialization
20 q = 1'bz;
21 end // initialization
22

```

```

23   specify
24     specparam d_clk    = 5;
25     specparam clk_d   = 50;
26
27     $setup($posedge in_maj3[2] && in_maj3[2], $posedge gatex, d_clk);
28     $setup($negedge in_maj3[2] && neg_in_maj3[2], $posedge gatex, d_clk);
29     $hold($posedge gatex, $negedge in_maj3[2] && in_maj3[2], clk_d);
30     $hold($posedge gatex, $posedge in_maj3[2] && neg_in_maj3[2], clk_d);
31
32     $setup($posedge in_maj3[1] && in_maj3[1], $posedge gatex, d_clk);
33     $setup($negedge in_maj3[1] && neg_in_maj3[1], $posedge gatex, d_clk);
34     $hold($posedge gatex, $negedge in_maj3[1] && in_maj3[1], clk_d);
35     $hold($posedge gatex, $posedge in_maj3[1] && neg_in_maj3[1], clk_d);
36
37     $setup($posedge in_maj3[0] && in_maj3[0], $posedge gatex, d_clk);
38     $setup($negedge in_maj3[0] && neg_in_maj3[0], $posedge gatex, d_clk);
39     $hold($posedge gatex, $negedge in_maj3[0] && in_maj3[0], clk_d);
40     $hold($posedge gatex, $posedge in_maj3[0] && neg_in_maj3[0], clk_d);
41
42   endspecify
43
44   always @($posedge gatex)
45   begin
46     if ((&in_maj3 != 1'bx) & (&in_maj3 != 1'bz))
47       begin
48         q <= (a & b)|(b & c)|(a&&c);
49         q <= #pul_wid 1'bz;
50       end
51     else
52       begin
53         q <= 1'bx;
54         q <= #pul_wid 1'bz;
55       end
56   end
57
58 endmodule

```

Listing 2.14: maj3_ppp Verilog model code.**Figure 2.35:** maj3_ppp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.

Switching energy

Different energy consumption results based on different data input patterns ($abc = 000$, $abc = 001$, $abc = 010$, $abc = 011$, $abc = 100$, $abc = 101$, $abc = 110$ and $abc = 111$) and clock frequencies.

Table 2.14: maj3_ppp switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
'000' (J)	2.36E-23	5.40E-23	1.42E-22	2.92E-22	6.28E-22	2.49E-21	8.02E-21
'001' (J)	5.29E-21	5.15E-21	4.76E-21	4.31E-21	4.63E-21	8.39E-21	1.34E-20
'010' (J)	5.82E-21	5.68E-21	5.29E-21	4.72E-21	5.01E-21	8.73E-21	1.48E-20
'011' (J)	5.30E-21	5.17E-21	4.77E-21	4.33E-21	4.64E-21	8.39E-21	1.35E-20
'100' (J)	5.30E-21	5.17E-21	4.77E-21	4.33E-21	4.64E-21	8.39E-21	1.35E-20
'101' (J)	5.82E-21	5.68E-21	5.29E-21	4.79E-21	5.01E-21	8.73E-21	1.48E-20
'110' (J)	5.29E-21	5.15E-21	4.76E-21	4.31E-21	4.63E-21	8.39E-21	1.34E-20
'111' (J)	2.36E-23	5.40E-23	1.42E-22	2.92E-22	6.28E-22	2.49E-21	8.02E-21

2.1.8 MAJ5

AQFP majority cells are built from buffer cells. The operational principle of a 5-input majority gate is to merge the output of five buffers through a 5-to-1 branch. There are 32 possible types of 5-input majority gate with different inputs (positive and negative). Here we only present the 5-input majority gate with five positive inputs named `maj5_ppppp`. Others are under development. The branch cells are introduced in the sub-cells.

Symbol

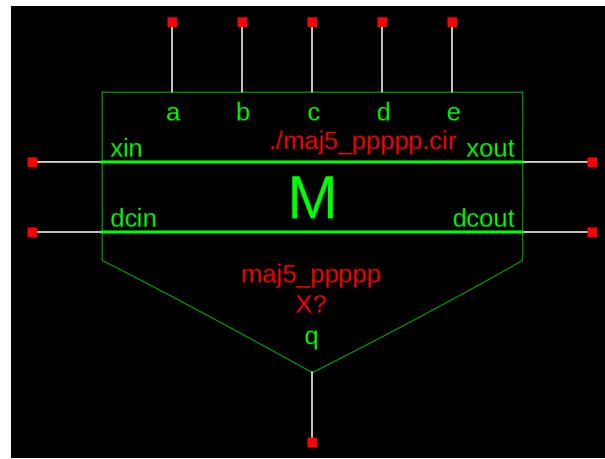


Figure 2.36: `maj5_ppppp` symbol.

Table 2.15: `maj5_ppppp` pin list.

Pin	Description
A	data input
B	data input
C	data input
D	data input
E	data input
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

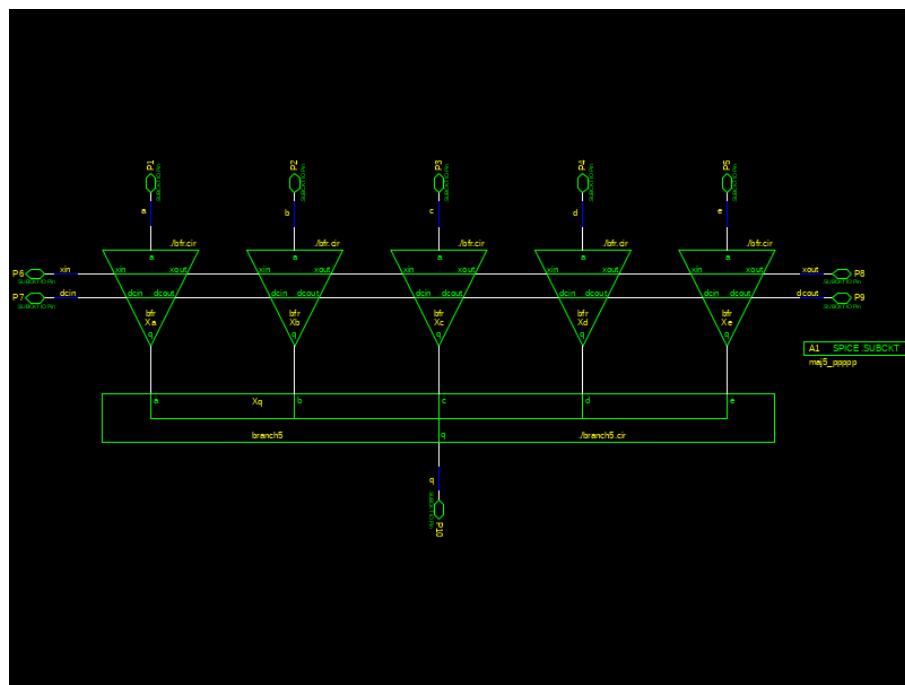


Figure 2.37: maj5_ppppp schematic.

Layout

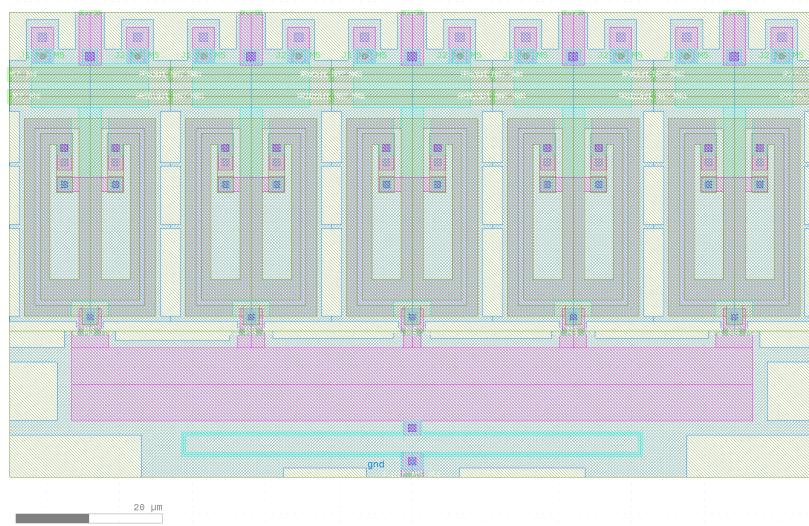


Figure 2.38: maj5_ppppp layout.

Analog model

```

1 .SUBCKT branch5 a b c d e q
2 L2 2 3 L2
3 L1 1 2 L1
4 Le e 3 Lc
5 Ld d 3 Ld
6 .param La=1.53p
7 .param Lb=0.47p
8 .param Lc=0.67p
9 .param Ld=0.47p
10 .param Le=1.53p
11 .param L1=1.11p
12 .param L2=1.11p
13 .param Lq=18.33p
14 Lq 2 q Lq
15 Lc c 2 Lc
16 Lb b 1 Lb
17 La a 1 La
18 .ends branch5
19 .SUBCKT bfr a xin dcin xout dcout q
20 Kid Lin Ld Kid
21 Kiout Lin Lout Kiout
22 Kix Lin Lx Kix
23 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
24 .param Lin=1.56p
25 .param Lx=6.53p
26 .param Ld=6.56p
27 .param L1=1.56p
28 .param L2=L1
29 .param Lq=8.25p
30 .param Lout=30.55p
31 .param B1=0.5
32 .param B2=B1
33 .param Kxd=0.2916
34 .param Kx1=-0.2461
35 .param Kx2=Kx1
36 .param Kd1=-0.1916
37 .param Kd2=Kd1
38 .param Kout=-0.6400
39 .param Kxout=-8.116E-6
40 .param Kdout=-2.873E-5
41 .param Kiout=2.201E-3
42 .param Kix=-1.145E-4
43 .param Kid=-1.063E-5
44
45 Kxout Lx Lout Kxout
46 Kdout Ld Lout Kdout
47 Kout Lout Lq Kout
48 Kd2 Ld L2 Kd2
49 Kx2 Lx L2 Kx2
50 Kd1 Ld L1 Kd1
51 Kx1 Lx L1 Kx1
52 Kxd Lx Ld Kxd
53 Lin a 2 Lin
54 B2 3 0 B2 jjmit area=B2
55 B1 1 0 B1 jjmit area=B1
56 Lout 0 q Lout
57 Lq 2 0 Lq
58 L2 3 2 L2
59 L1 2 1 L1
60 Ld dcin dcout Ld
61 Lx xin xout Lx
62 .ends bfr
63 .SUBCKT maj5_ppppp a b c d e xin dcin xout dcout q
64 Xe e 10 11 xout dcout 13 bfr
65 Xa a xin dcin 4 5 12 bfr
66 Xq 12 1 6 9 13 q branch5
67 Xd d 7 8 10 11 9 bfr
68 Xc c 2 3 7 8 6 bfr
69 Xb b 4 5 2 3 1 bfr

```

```
70 | .ends maj5_ppppp
```

Listing 2.15: maj5_ppppp netlist (.cir).

Simulation result

Simulation waveform of a 5-input MAJORITY (maj5_ppppp) DUT (design under test) with three 4-stage buffers before the input ‘a’, ‘b’, ‘c’, ‘d’ and ‘e’. Another 3-stage buffer is placed after the MAJORITY’s output ‘q’. Clock frequency is 5 GHz. Signals from top to bottom: buffer chain input (a) as 010101001011010101010101, (b) as 001100110011001100110011, (c) as 0000111100011110000111100001111, (d) as 000000011111110000000011111111, and (e) as 000000000000000011111111111111 (inputs of the first stage buffers), AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), and the output generated from the final stage of buffer (q) as 1100000001000101110001011101111111 with two random initial outputs 11. This is because of the meander structure of AQFP circuits. In an 8-stage AQFP circuit, it takes two clock cycles to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

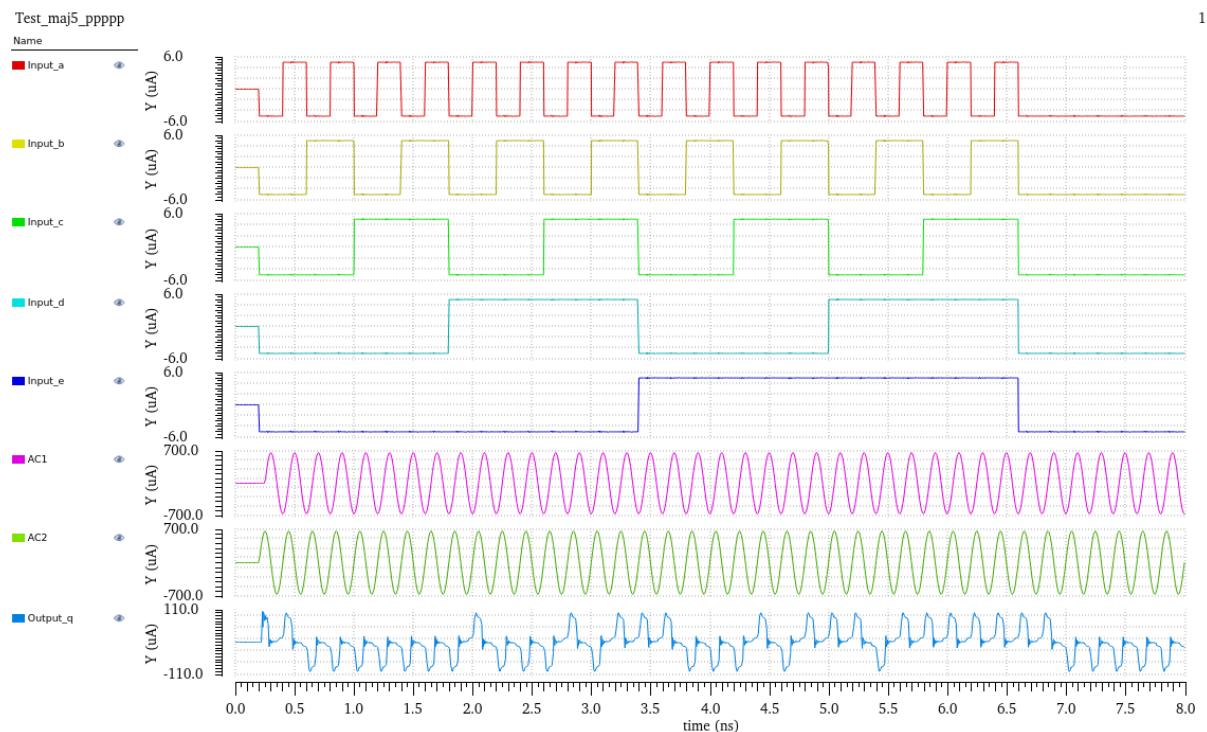


Figure 2.39: maj5_ppppp analog waveform.

Digital model

```
1 |
2 | 'timescale 1ps/10fs
3 | module maj5_ppppp(a, b, c, d, e, q, xin, xout, dcin, dcout);
4 |   input a, b, c, d, e;
5 |   output q;
6 |   inout xin, xout, dcin, dcout;
7 |   wire [4:0] in_maj5;
8 |   reg q;
```

```

9  parameter pul_wid = 100;
10 wire [4:0] neg_in_maj5;
11
12 assign in_maj5 = {a, b, c, d, e};
13 assign neg_in_maj5 = ~in_maj5;
14
15 biasDir_b I0(xin, xout, dcin, dcout, gatex);
16
17 initial begin
18
19 $timeformat(-12, 1, "ps", 8); // time format
20
21 // output register initialization
22 q = 1'bz;
23 end // initialization
24
25 specify
26   specparam d_clk    = 5;
27   specparam clk_d   = 50;
28
29 $setup(posedge in_maj5[4] && in_maj5[4], posedge gatex, d_clk);
30 $setup(negedge in_maj5[4] && neg_in_maj5[4], posedge gatex, d_clk);
31 $hold(posedge gatex, negedge in_maj5[4] && in_maj5[4], clk_d);
32 $hold(posedge gatex, posedge in_maj5[4] && neg_in_maj5[4], clk_d);
33
34 $setup(posedge in_maj5[3] && in_maj5[3], posedge gatex, d_clk);
35 $setup(negedge in_maj5[3] && neg_in_maj5[3], posedge gatex, d_clk);
36 $hold(posedge gatex, negedge in_maj5[3] && in_maj5[3], clk_d);
37 $hold(posedge gatex, posedge in_maj5[3] && neg_in_maj5[3], clk_d);
38
39 $setup(posedge in_maj5[2] && in_maj5[2], posedge gatex, d_clk);
40 $setup(negedge in_maj5[2] && neg_in_maj5[2], posedge gatex, d_clk);
41 $hold(posedge gatex, negedge in_maj5[2] && in_maj5[2], clk_d);
42 $hold(posedge gatex, posedge in_maj5[2] && neg_in_maj5[2], clk_d);
43
44 $setup(posedge in_maj5[1] && in_maj5[1], posedge gatex, d_clk);
45 $setup(negedge in_maj5[1] && neg_in_maj5[1], posedge gatex, d_clk);
46 $hold(posedge gatex, negedge in_maj5[1] && in_maj5[1], clk_d);
47 $hold(posedge gatex, posedge in_maj5[1] && neg_in_maj5[1], clk_d);
48
49 $setup(posedge in_maj5[0] && in_maj5[0], posedge gatex, d_clk);
50 $setup(negedge in_maj5[0] && neg_in_maj5[0], posedge gatex, d_clk);
51 $hold(posedge gatex, negedge in_maj5[0] && in_maj5[0], clk_d);
52 $hold(posedge gatex, posedge in_maj5[0] && neg_in_maj5[0], clk_d);
53 endspecify
54
55 always @(posedge gatex)
56 begin
57   if (&in_maj5!= 1'bx & &in_maj5!= 1'bz)
58     begin
59       q <= (a & b & c)|(a & b &d)|(a &b &e)|(a & c & d)|(a & c & e)|(a & d & e)|(b & c
60       ↛ & d)|(b & c & e)|(b & d & e)|(c & d & e);
61       q <= #pul_wid 1'bz;
62     end
63   else
64     begin
65       q <= 1'bx;
66       q <= #pul_wid 1'bz;
67     end
68   end
69 endmodule

```

Listing 2.16: maj5_ppppp Verilog model code.

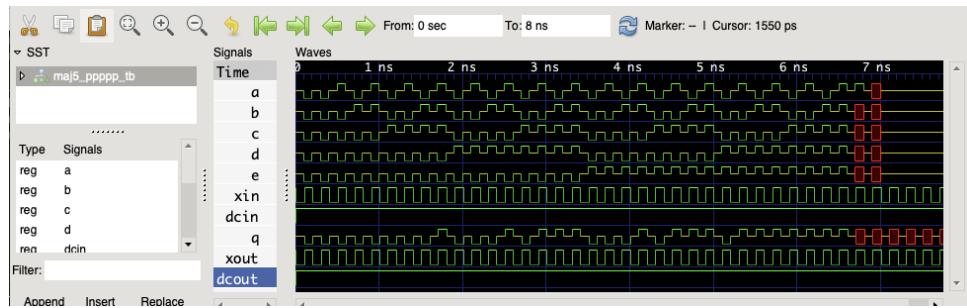


Figure 2.40: maj5_ppppp digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.

Switching energy

Different energy consumption results based on different data input patterns (from $abcde = 00000$ to $abcde = 11111$) and clock frequencies (from 0.1GHz to 10GHz).

Table 2.16: maj5_pppp switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
'00000' (J)	4.55E-23	9.68E-23	2.52E-22	5.38E-22	1.35E-21	5.62E-21	1.54E-20
'00001' (J)	6.07E-21	5.96E-21	5.63E-21	5.07E-21	5.96E-21	1.25E-20	1.78E-20
'00010' (J)	6.82E-21	6.71E-21	6.40E-21	5.95E-21	5.69E-21	1.31E-20	1.87E-20
'00011' (J)	8.60E-21	8.31E-21	7.47E-21	6.72E-21	6.76E-21	1.42E-20	1.59E-20
'00100' (J)	7.29E-21	7.18E-21	6.88E-21	6.32E-21	5.69E-21	1.33E-20	1.99E-20
'00101' (J)	1.03E-20	1.00E-20	9.18E-21	8.14E-21	7.96E-21	1.59E-20	1.96E-20
'00110' (J)	1.05E-20	1.02E-20	9.37E-21	8.26E-21	7.91E-21	1.61E-20	2.07E-20
'00111' (J)	8.60E-21	8.31E-21	7.47E-21	6.72E-21	6.76E-21	1.42E-20	1.59E-20
'01000' (J)	6.82E-21	6.71E-21	6.40E-21	5.95E-21	5.69E-21	1.31E-20	1.87E-20
'01001' (J)	1.03E-20	9.95E-21	9.08E-21	8.05E-21	7.56E-21	1.64E-20	1.89E-20
'01010' (J)	1.05E-20	1.02E-20	9.37E-21	8.09E-21	7.68E-21	1.68E-20	1.97E-20
'01011' (J)	1.03E-20	1.00E-20	9.18E-21	8.14E-21	7.96E-21	1.59E-20	1.96E-20
'01100' (J)	1.05E-20	1.02E-20	9.37E-21	8.26E-21	7.91E-21	1.61E-20	2.07E-20
'01101' (J)	1.03E-20	9.95E-21	9.08E-21	8.05E-21	7.56E-21	1.64E-20	1.89E-20
'01110' (J)	9.64E-21	9.35E-21	8.49E-21	7.39E-21	6.80E-21	1.59E-20	1.69E-20
'01111' (J)	6.07E-21	5.96E-21	5.63E-21	5.07E-21	5.96E-21	1.25E-20	1.78E-20
'10000' (J)	6.07E-21	5.96E-21	5.63E-21	5.07E-21	5.96E-21	1.25E-20	1.78E-20
'10001' (J)	9.64E-21	9.35E-21	8.49E-21	7.39E-21	6.80E-21	1.59E-20	1.69E-20
'10010' (J)	1.03E-20	9.95E-21	9.08E-21	8.05E-21	7.56E-21	1.64E-20	1.89E-20
'10011' (J)	1.05E-20	1.02E-20	9.37E-21	8.26E-21	7.91E-21	1.61E-20	2.07E-20
'10100' (J)	1.03E-20	1.00E-20	9.18E-21	8.14E-21	7.96E-21	1.59E-20	1.96E-20
'10101' (J)	1.05E-20	1.02E-20	9.37E-21	8.09E-21	7.68E-21	1.68E-20	1.97E-20
'10110' (J)	1.03E-20	9.95E-21	9.08E-21	8.05E-21	7.56E-21	1.64E-20	1.89E-20
'10111' (J)	6.82E-21	6.71E-21	6.40E-21	5.95E-21	5.69E-21	1.31E-20	1.87E-20
'11000' (J)	8.60E-21	8.31E-21	7.47E-21	6.72E-21	6.76E-21	1.42E-20	1.59E-20
'11001' (J)	1.05E-20	1.02E-20	9.37E-21	8.26E-21	7.91E-21	1.61E-20	2.07E-20
'11010' (J)	1.03E-20	1.00E-20	9.18E-21	8.14E-21	7.96E-21	1.59E-20	1.96E-20
'11011' (J)	7.29E-21	7.18E-21	6.88E-21	6.32E-21	5.69E-21	1.33E-20	1.99E-20
'11100' (J)	8.60E-21	8.31E-21	7.47E-21	6.72E-21	6.76E-21	1.42E-20	1.59E-20
'11101' (J)	6.82E-21	6.71E-21	6.40E-21	5.95E-21	5.69E-21	1.31E-20	1.87E-20
'11110' (J)	6.07E-21	5.96E-21	5.63E-21	5.07E-21	5.96E-21	1.25E-20	1.78E-20
'11111' (J)	4.55E-23	9.68E-23	2.52E-22	5.38E-22	1.35E-21	5.62E-21	1.54E-20

2.1.9 BOOST1

Boosters are the buffering elements in AQFP logic with large output current, which are used to increase the maximum interconnect wirelength during data propagation with low bit-error rate. They are also used to design splitting elements to increase the fanout during data propagation. There are two types of booster cells: **boost1** and **boost2**.

boost1 is designed to amplify the output current of a single buffer to achieve higher drivability and longer maximum interconnect wirelength. The operational principle is to first split the input of a buffer and send the split signal to an additional buffer, then merge the outputs of two buffers by coupling the two outputs with one single inductor.

Symbol

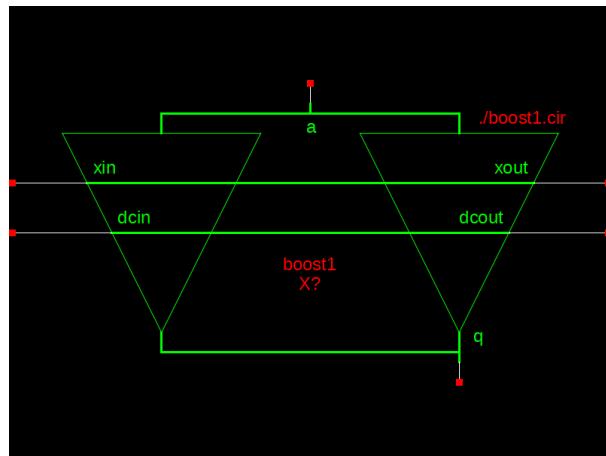


Figure 2.41: boost1 symbol.

Table 2.17: boost1 pin list.

Pin	Description
A	data input
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic



Figure 2.42: boost1 schematic.

Layout

Figure 2.43: boost1 layout.

Analog model

```

1 | .SUBCKT boost1 a xin dcin xout dcout q
2 | Kdinp Ld Linb Kdinp
3 | Kxinp Lx Linb Kxinp
4 | Kdina Ld Lina Kdina
5 | Kxina Lx Lina Kxina
6 | Lin a 4 Lin
7 | Kxqb Lx Lqb Kxqb

```

```

8 | Kdqb Ld Lqb Kdqb
9 | Koutb Lout Lqb Koutb
10 | Kd2b Ld L2b Kd2b
11 | Kx2b Lx L2b Kx2b
12 | Kd1b Ld L1b Kd1b
13 | Kx1b Lx L1b Kx1b
14 | Linb 4 6 Linb
15 | B2b 7 0 B2b jjmit area=B2b
16 | B1b 5 0 B1b jjmit area=B1b
17 | Lqb 6 0 Lqb
18 | L2b 7 6 L2b
19 | L1b 6 5 L1b
20 | .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
21 | .param Lin=0.17p
22 | .param Lx=12.97p
23 | .param Ld=13.11p
24 | .param Lina=4.94p
25 | .param L1a=1.45p
26 | .param L2a=1.41p
27 | .param Lqa=8.44p
28 | .param B1a=0.5
29 | .param B2a=B1a
30 | .param Linb=4.94p
31 | .param L1b=1.41p
32 | .param L2b=1.45p
33 | .param Lqb=8.38p
34 | .param B1b=0.5
35 | .param B2b=B1b
36 | .param Lout=63.86p
37 | .param Kxd=0.28748
38 | .param Kx1a=-0.18202
39 | .param Kx2a=-0.18114
40 | .param Kd1a=-0.14100
41 | .param Kd2a=-0.14178
42 | .param Kouta=-0.42832
43 | .param Kxqa=-1.265E-3
44 | .param Kdqa=-1.553E-3
45 | .param Kx1b=-0.18109
46 | .param Kx2b=-0.18196
47 | .param Kd1b=-0.14164
48 | .param Kd2b=-0.14086
49 | .param Koutb=-0.38308
50 | .param Kxqb=1.546E-3
51 | .param Kdqb=1.873E-3
52 | .param Kxout=7.708E-4
53 | .param Kdout=1.112E-3
54 |
55 | .param Kxina=-8.85E-3
56 | .param Kdina=-5.99E-3
57 | .param Kxinb=8.71E-3
58 | .param Kdinb=5.88E-3
59 | Kxqa Lx Lqa Kxqa
60 | Kdqa Ld Lqa Kdqa
61 | Kxout Lx Lout Kxout
62 | Kdout Ld Lout Kdout
63 | Kouta Lout Lqa Kouta
64 | Kd2a Ld L2a Kd2a
65 | Kx2a Lx L2a Kx2a
66 | Kd1a Ld L1a Kd1a
67 | Kx1a Lx L1a Kx1a
68 | Kxd Lx Ld Kxd
69 | Lina 4 2 Lina
70 | B2a 3 0 B2a jjmit area=B2a
71 | B1a 1 0 B1a jjmit area=B1a
72 | Lout 0 q Lout
73 | Lqa 2 0 Lqa
74 | L2a 3 2 L2a
75 | L1a 2 1 L1a
76 | Ld dcin dcout Ld
77 | Lx xin xout Lx

```

```
78 | .ends boost1
```

Listing 2.17: boost1 netlist (.cir).

Simulation result

Simulation waveform of a buffer type1 (**boost1**) DUT (design under test) with a 4-stage buffer before the input ‘a’. Another two 3-stage buffers are placed after the **boost1**’s output ‘q’. Clock frequency is 5 GHz. Signals from top to bottom: buffer chain input (a) (input of the first stage buffers) as 01010101, AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), and the outputs generated from the output generated from the final stage of buffer (q) as 0001010101 with two random initial outputs 00. This is because of the meander structure of AQFP circuits. In an 8-stage AQFP circuit, it takes two clock cycles to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

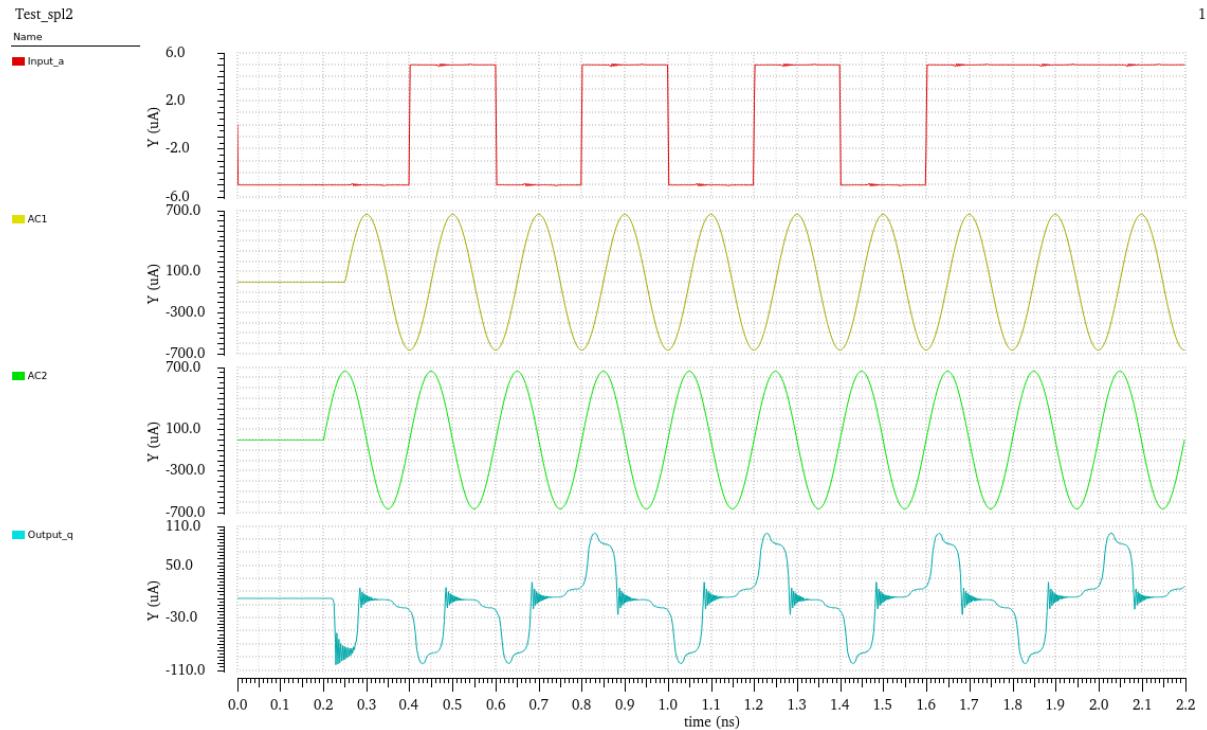


Figure 2.44: boost1 analog waveform.

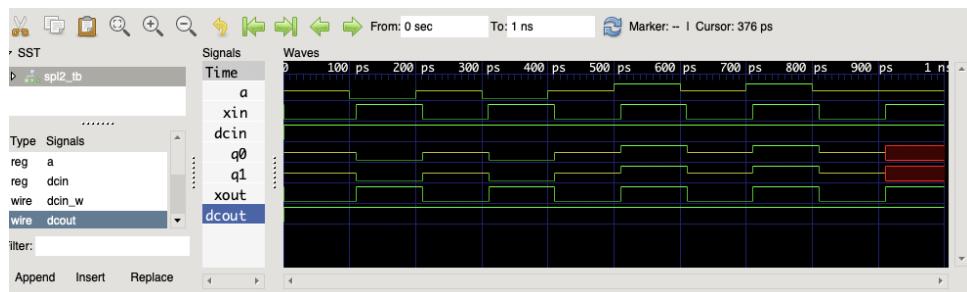
Digital model

```
1  'timescale 1ps/10fs
2  module boost1(a, q, xin, xout, dcin, dc当地);
3  input a;
4  output q;
5  inout xin, xout, dcin, dc当地;
6  reg q;
7  parameter pul_wid = 100;
8  wire not_a;
9
10 assign not_a = !a;
11
12 biasDir_b I0(xin, xout, dcin, dc当地, gatex);
```

```

13
14     initial begin
15
16         $timeformat(-12, 1, "ps", 8); // time format
17
18         // output register initialization
19         q = 1'bz;
20     end // initialization
21
22     specify
23         specparam d_clk      = 5;
24         specparam clk_d     = 50;
25
26         $setup(posedge a && a, posedge gatex, d_clk);
27         $setup(negedge a && not_a, posedge gatex, d_clk);
28         $hold(posedge gatex, negedge a && a, clk_d);
29         $hold(posedge gatex, posedge a && not_a, clk_d);
30     endspecify
31
32     always @(posedge gatex)
33     begin
34         if (a == 1 | a == 0)
35             begin
36                 q <= a;
37                 q <= #pul_wid 1'bz;
38             end
39         else
40             begin
41                 q <= 1'bx;
42                 q <= #pul_wid 1'bz;
43             end
44     end
45 endmodule

```

Listing 2.18: boost1 Verilog model code.**Figure 2.45:** spl2 digital waveform. HDL '1': AQFP '1'; HDL '0': AQFP '0'; HDL 'z': inactive; HDL 'x': error.

Switching energy

Different energy consumption results based on different data input patterns ($a = 0$ and $a = 1$) and clock frequencies.

Table 2.18: spl2 switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
'0' (J)	2.72E-23	7.43E-23	2.16E-22	4.65E-22	1.09E-21	4.12E-21	1.05E-20
'1' (J)	2.71E-23	7.41E-23	2.16E-22	4.64E-22	1.09E-21	4.12E-21	1.05E-20

2.1.10 BOOST2

Boosters are the buffering elements in AQFP logic with large output current, which are used to increase the maximum interconnect wirelength during data propagation with low bit-error rate. They are also used to design splitting elements to increase the fanout during data propagation. There are two types of booster cells: **boost1** and **boost2**.

boost2 uses another approach to first amplify the input current use a single buffer, then split the output and connect them to multiple buffers to obtain multiple fanouts. All the buffers in **boost2** share the same ac (clock) interconnect, therefore there is no extra clock phase consumed in this design. There are 2 types of **boost2** cells with different fanouts: **boost2** with 2 fanouts (**boost2f2**) and with 4 fanouts (**boost2f4**). Here we only present the **boost2f2** cell as an examples. The other one can be found in the deliverables.

Symbol

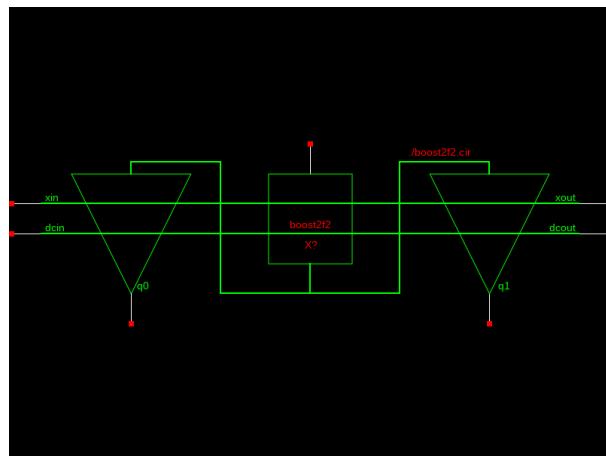


Figure 2.46: **boost2f2** symbol.

Table 2.19: **boost2f2** pin list.

Pin	Description
A	data input
XIN	serial clock input
DCIN	dc offset input
Q0	data output
Q1	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

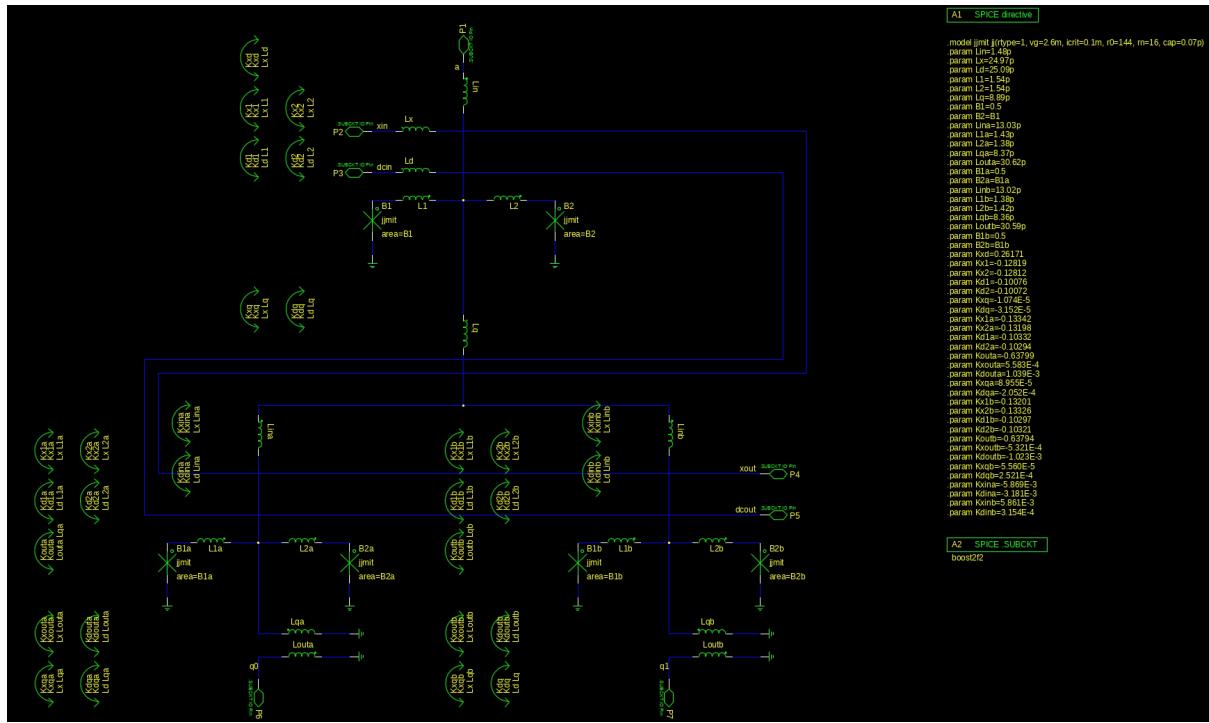


Figure 2.47: boost2f2 schematic.

Layout

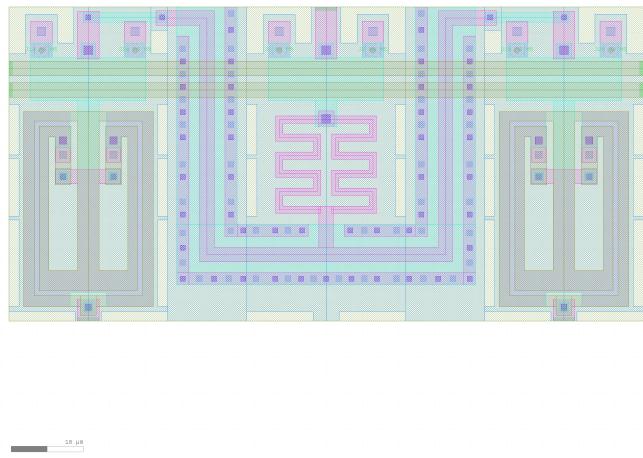


Figure 2.48: boost2f2 layout.

Analog model

```

1 .SUBCKT boost2f2 a xin dcin xout dcout q0 q1
2 Kdinp Ld Linb Kdinp
3 Kxinp Lx Linb Kxinp
4 Kdina Ld Lina Kdina
5 Kxina Lx Lina Kxina
6 Kxqb Lx Lqb Kxqb
7 Kxoutb Lx Louth Kxoutb
8 Kdoutb Ld Louth Kdoutb
9 Koutb Loutb Lqb Koutb
10 Kd2b Ld L2b Kd2b
11 Kx2b Lx L2b Kx2b
12 Kd1b Ld L1b Kd1b
13 Kx1b Lx L1b Kx1b
14 Linb 4 9 Linb
15 B2b 10 0 B2b jjmit area=B2b
16 B1b 8 0 B1b jjmit area=B1b
17 Loutb 0 q1 Loutb
18 Lqb 9 0 Lqb
19 L2b 10 9 L2b
20 L1b 9 8 L1b
21 Kxqa Lx Lqa Kxqa
22 Kdqa Ld Lqa Kdqa
23 Kxouta Lx Louta Kxouta
24 Kdouta Ld Louta Kdouta
25 Kouta Louta Lqa Kouta
26 Kd2a Ld L2a Kd2a
27 Kx2a Lx L2a Kx2a
28 Kd1a Ld L1a Kd1a
29 Kx1a Lx L1a Kx1a
30 Lina 4 6 Lina
31 B2a 7 0 B2a jjmit area=B2a
32 B1a 5 0 B1a jjmit area=B1a
33 Louta 0 q0 Louta
34 Lqa 6 0 Lqa
35 L2a 7 6 L2a
36 L1a 6 5 L1a
37 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
38 .param Lin=1.48p
39 .param Lx=24.97p
40 .param Ld=25.09p
41 .param L1=1.54p
42 .param L2=1.54p
43 .param Lq=8.89p
44 .param B1=0.5
45 .param B2=B1
46 .param Lina=13.03p
47 .param L1a=1.43p
48 .param L2a=1.38p
49 .param Lqa=8.37p
50 .param Louta=30.62p
51 .param B1a=0.5
52 .param B2a=B1a
53 .param Linb=13.02p
54 .param L1b=1.38p
55 .param L2b=1.42p
56 .param Lqb=8.36p
57 .param Loutb=30.59p
58 .param B1b=0.5
59 .param B2b=B1b
60 .param Kxd=0.26171
61 .param Kx1=-0.12819
62 .param Kx2=-0.12812
63 .param Kd1=-0.10076
64 .param Kd2=-0.10072
65 .param Kxq=-1.074E-5
66 .param Kdq=-3.152E-5
67 .param Kx1a=-0.13342
68 .param Kx2a=-0.13198
69 .param Kd1a=-0.10332
70 .param Kd2a=-0.10294

```

```

71 .param Kouta=-0.63799
72 .param Kxouta=5.583E-4
73 .param Kdouta=1.039E-3
74 .param Kxqa=8.955E-5
75 .param Kdqa=-2.052E-4
76 .param Kx1b=-0.13201
77 .param Kx2b=-0.13326
78 .param Kd1b=-0.10297
79 .param Kd2b=-0.10321
80 .param Koutb=-0.63794
81 .param Kxoutb=-5.321E-4
82 .param Kdoutb=-1.023E-3
83 .param Kxqb=-5.560E-5
84 .param Kdqb=2.521E-4
85 .param Kxina=-5.869E-3
86 .param Kdina=-3.181E-3
87 .param Kxinb=5.861E-3
88 .param Kdinb=3.154E-4
89 Kxq Lx Lq Kxq
90 Kdq Ld Lq Kdq
91 Kd2 Ld L2 Kd2
92 Kx2 Lx L2 Kx2
93 Kd1 Ld L1 Kd1
94 Kx1 Lx L1 Kx1
95 Kxd Lx Ld Kxd
96 Lin a 2 Lin
97 B2 3 0 B2 jjmit area=B2
98 B1 1 0 B1 jjmit area=B1
99 Lq 2 4 Lq
100 L2 3 2 L2
101 L1 2 1 L1
102 Ld dcin dcout Ld
103 Lx xin xout Lx
104 .ends boost2f2

```

Listing 2.19: boost2f2 netlist (.cir).

Simulation result

Simulation waveform of a buffer type2 with 2 fan-outs (**boost2f2**) DUT (design under test) with a 4-stage buffer before the input ‘a’. Another two 3-stage buffers are placed after the **boost2f2**’s output ‘q0’ and ‘q1’. Clock frequency is 5 GHz. Signals from top to bottom: buffer chain input (a) (input of the first stage buffers) as 010101011, AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), and the outputs generated from the DUT **boost2f2** as 010101011 with a random initial output 0. This is because of the meander structure of AQFP circuits. Since the DUT **boost2f2** is placed after a 4-stage buffer, it takes one clock cycle to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

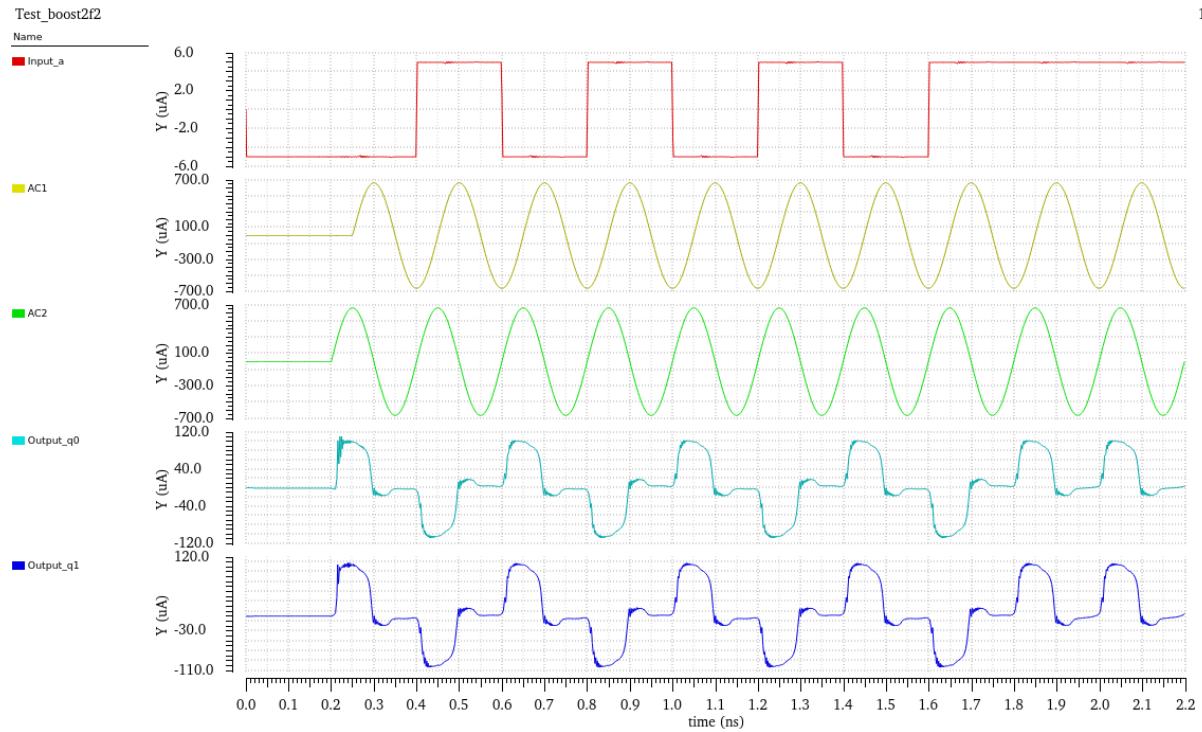


Figure 2.49: boost2f2 analog waveform.

Digital model

```

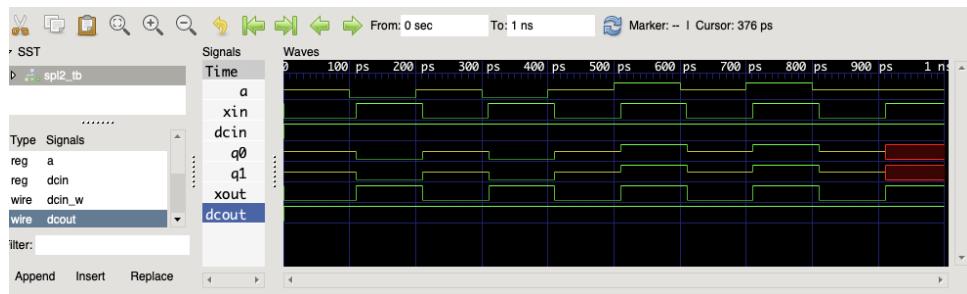
1  `timescale 1ps/10fs
2  module boost2f2(a, q0, q1, xin, xout, dcin, dcout);
3    input a;
4    output q0, q1;
5    inout xin, xout, dcin, dcout;
6    reg q0, q1;
7    parameter pul_wid = 100;
8    wire not_a;
9
10   assign not_a = ~a;
11
12   biasDir_b I0(xin, xout, dcin, dcout, gatex);
13
14   initial begin
15
16     $timeformat(-12, 1, "_ps", 8); // time format
17
18     // output register initialization
19     q0 = 1'b0;
20     q1 = 1'b0;
21   end // initialization
22
23   specify
24     specparam d_clk    = 5;
25     specparam clk_d   = 50;
26
27     $setup(posedge a && a, posedge gatex, d_clk);
28     $setup(negedge a && not_a, posedge gatex, d_clk);
29     $hold(posedge gatex, negedge a && a, clk_d);
30     $hold(posedge gatex, posedge a && not_a, clk_d);
31
32   endspecify
33
34   always @(posedge gatex)

```

```

35   begin
36     if (a == 1 | a == 0)
37       begin
38         q0 <= a;
39         q0 <= #pul_wid 1'bz;
40         q1 <= a;
41         q1 <= #pul_wid 1'bz;
42       end
43     else
44       begin
45         q0 <= 1'bx;
46         q0 <= #pul_wid 1'bz;
47         q1 <= 1'bx;
48         q1 <= #pul_wid 1'bz;
49       end
50   end
51
52 endmodule

```

Listing 2.20: boost2f2 Verilog model code.**Figure 2.50:** spl2 digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.

Switching energy

Different energy consumption results based on different data input patterns ($a = 0$ and $a = 1$) and clock frequencies.

Table 2.20: boost2f2 switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
‘0’ (J)	2.22E-22	4.42E-22	1.10E-21	2.17E-21	4.32E-21	1.13E-20	2.94E-20
‘1’ (J)	2.51E-22	4.66E-22	1.11E-21	2.17E-21	4.38E-21	1.13E-20	2.63E-20

2.1.11 SPL

Splitters are the splitting elements in AQFP logic to increase the fanout during data propagation. An AQFP splitter is built from buffer and 1-to-n branch cells. The operational principle is to split the output of a buffer through a 1-to-n branch. There are 3 types of splitter cells with different fanouts: splitter 1-to-2 (**spl2**), 1-to-3 (**spl3**) and 1-to-4 (**spl4**). One should notice that the (**spl4**) is designed using a different approach, which is more close to (**boost1**). Instead of reading out the outputs of the buffers in (**boost1**), a (**spl4**) splits the output current from the 2 buffers into four to achieve 4 fan-outs. Here we only present the **spl2** cell as an example. The other one can be found in the deliverables. The branch is introduced in sub-cells.

Symbol

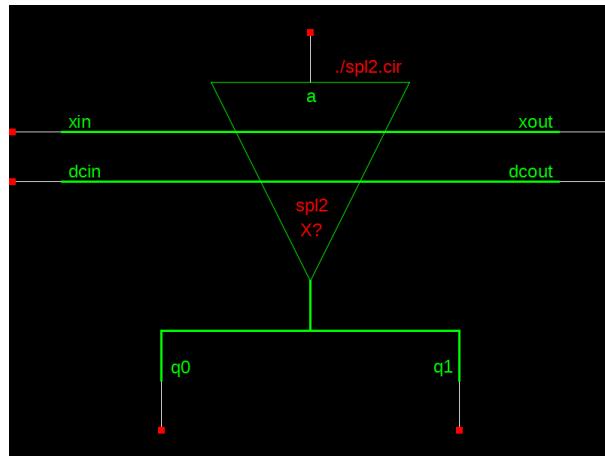


Figure 2.51: *spl2* symbol.

Table 2.21: *spl2* pin list.

Pin	Description
A	data input
XIN	serial clock input
DCIN	dc offset input
Q0	data output
Q1	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

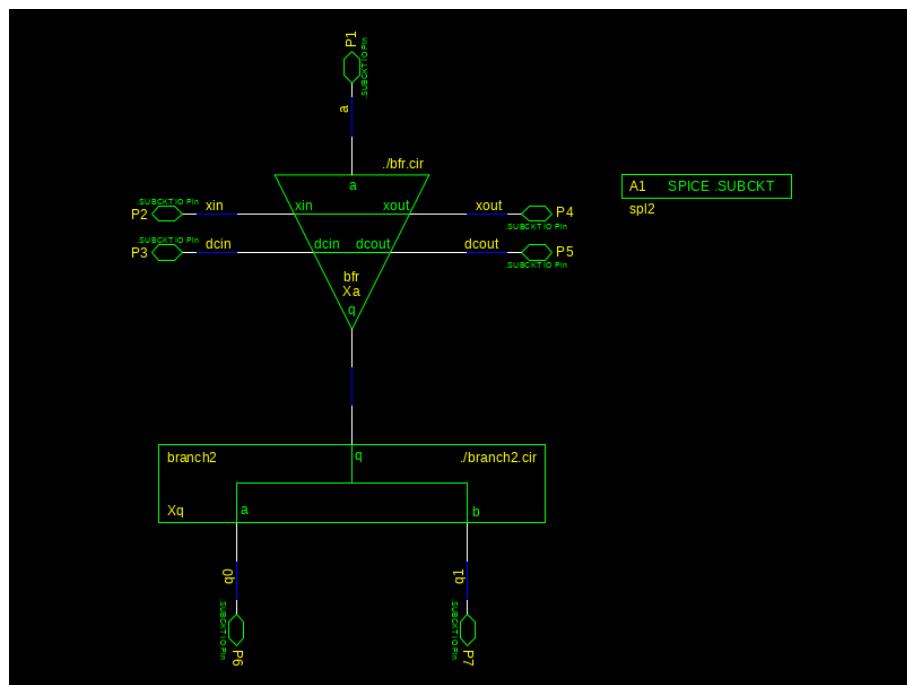


Figure 2.52: spl2 schematic.

Layout

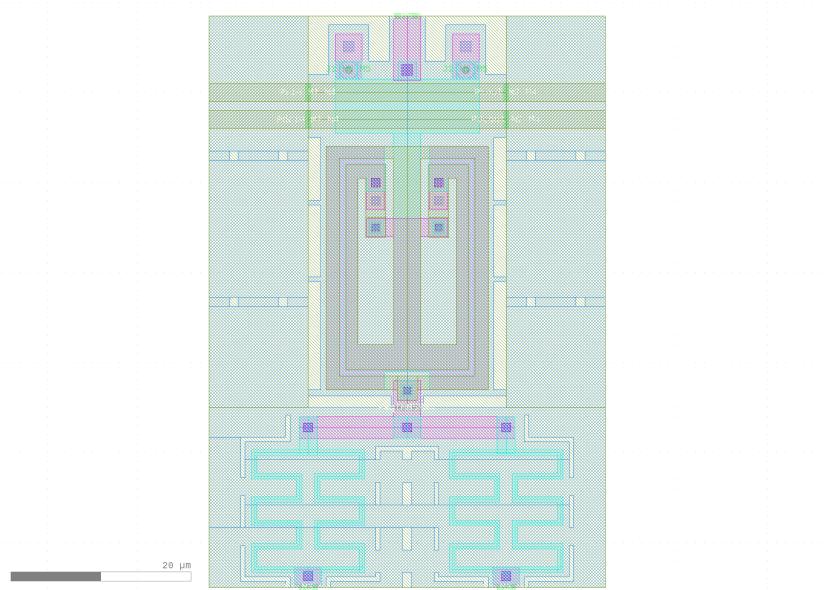


Figure 2.53: spl2 layout.

Analog model

```

1 .SUBCKT bfr a xin dcin xout dcout q
2 Kid Lin Ld Kid
3 Kiout Lin Lout Kiout
4 Kix Lin Lx Kix
5 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
6 .param Lin=1.56p
7 .param Lx=6.53p
8 .param Ld=6.56p
9 .param L1=1.56p
10 .param L2=L1
11 .param Lq=8.25p
12 .param Lout=30.55p
13 .param B1=0.5
14 .param B2=B1
15 .param Kxd=0.2916
16 .param Kx1=-0.2461
17 .param Kx2=Kx1
18 .param Kd1=-0.1916
19 .param Kd2=Kd1
20 .param Kout=-0.6400
21 .param Kxout=-8.116E-6
22 .param Kdout=-2.873E-5
23 .param Kiout=2.201E-3
24 .param Kix=-1.145E-4
25 .param Kid=-1.063E-5
26
27 Kxout Lx Lout Kxout
28 Kdout Ld Lout Kdout
29 Kout Lout Lq Kout
30 Kd2 Ld L2 Kd2
31 Kx2 Lx L2 Kx2
32 Kd1 Ld L1 Kd1
33 Kx1 Lx L1 Kx1
34 Kxd Lx Ld Kxd
35 Lin a 2 Lin
36 B2 3 0 B2 jjmit area=B2
37 B1 1 0 B1 jjmit area=B1
38 Lout 0 q Lout
39 Lq 2 0 Lq
40 L2 3 2 L2
41 L1 2 1 L1
42 Ld dcin dcout Ld
43 Lx xin xout Lx
44 .ends bfr
45 .SUBCKT bias_pair_11um a b c d
46 .param L0=3.60p
47 .param L1=3.60p
48
49 L1 c d L1
50 L0 a b L0
51 .ends bias_pair_11um
52 .SUBCKT branch2 a b q
53 .param La=12.4p
54 .param Lb=12.4p
55 .param Lq=0.17p
56 Lq 1 q Lq
57 Lb b 1 Lb
58 La a 1 La
59 .ends branch2
60 .SUBCKT spl2 a xin dcin xout dcout q0 q1
61 Xq q0 q1 1 branch2
62 X1 xin 4 dcin 5 bias_pair_11um
63 X2 2 xout 3 dcout bias_pair_11um
64 Xa a 4 5 2 3 1 bfr
65 .ends spl2

```

Listing 2.21: spl2 netlist (.cir).

Simulation result

Simulation waveform of a splitter 1-to-2 (spl2) DUT (design under test) with a 4-stage buffer before the input ‘a’. Another two 3-stage buffers are placed after the spl2’s output ‘q0’ and ‘q1’. Clock frequency is 5 GHz. Signals from top to bottom: buffer chain input (a) (input of the first stage buffers) as 010101011, AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), and the outputs generated from the DUT spl2 as 010101011 with a random initial output 0. This is because of the meander structure of AQFP circuits. Since the DUT spl2 is placed after a 4-stage buffer, it takes one clock cycle to propagate the data to the output. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

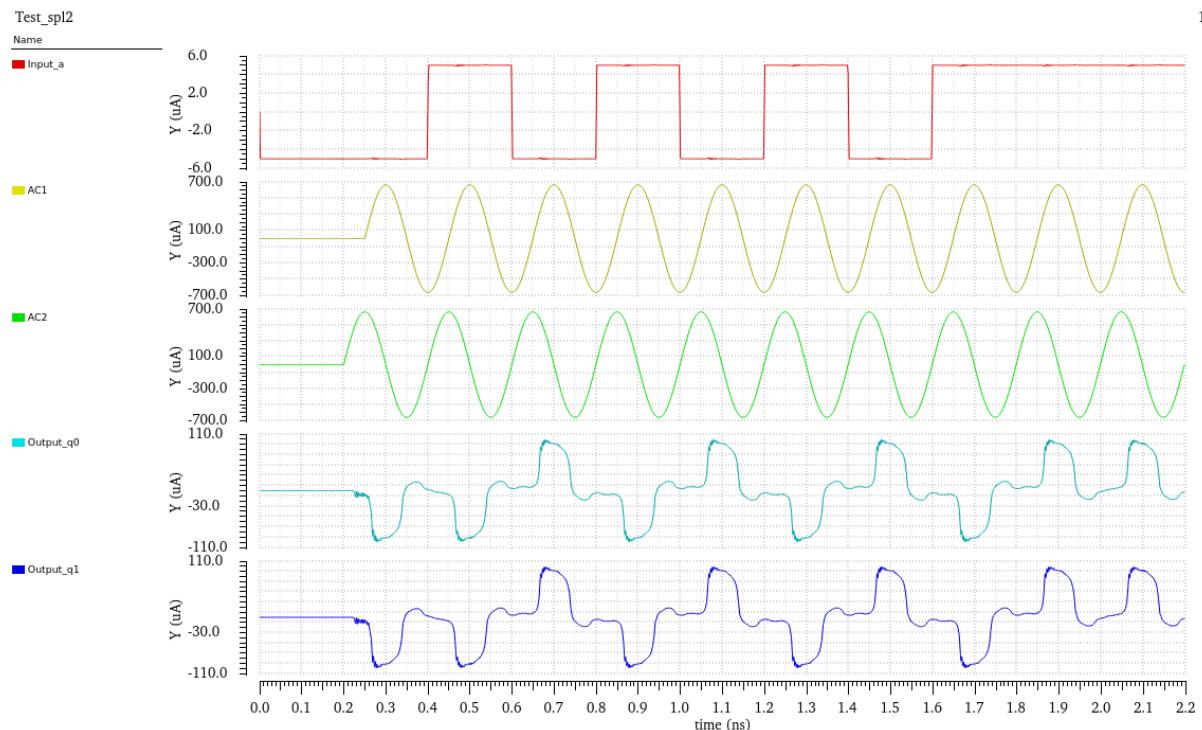


Figure 2.54: spl2 analog waveform.

Digital model

```

1  `timescale 1ps/10fs
2  module spl2(a, q0, q1, xin, xout, dcin, dcout);
3    input a;
4    output q0, q1;
5    inout xin, xout, dcin, dcout;
6    reg q0, q1;
7    parameter pul_wid = 100;
8    wire not_a;
9
10   assign not_a = ~a;
11
12   biasDir_b I0(xin, xout, dcin, dcout, gatex);
13
14   initial begin
15
16     $timeformat(-12, 1, "_ps", 8); // time format

```

```

17 // output register initialization
18 q0 = 1'bz;
19 q1 = 1'bz;
20 end // initialization
21
22
23 specify
24   specparam d_clk    = 5;
25   specparam clk_d    = 50;
26
27 $setup(posedge a && a, posedge gatex, d_clk);
28 $setup(negedge a && not_a, posedge gatex, d_clk);
29 $hold(posedge gatex, negedge a && a, clk_d);
30 $hold(posedge gatex, posedge a && not_a, clk_d);
31
32 endspecify
33
34 always @(posedge gatex)
35 begin
36   if (a == 1 | a == 0)
37     begin
38       q0 <= a;
39       q0 <= #pul_wid 1'bz;
40       q1 <= a;
41       q1 <= #pul_wid 1'bz;
42     end
43   else
44     begin
45       q0 <= 1'bx;
46       q0 <= #pul_wid 1'bz;
47       q1 <= 1'bx;
48       q1 <= #pul_wid 1'bz;
49     end
50   end
51
52 endmodule

```

Listing 2.22: spl2 Verilog model code.**Figure 2.55:** spl2 digital waveform. HDL '1': AQFP '1'; HDL '0': AQFP '0'; HDL 'z': inactive; HDL 'x': error.

Switching energy

Different energy consumption results based on different data input patterns ($a = 0$ and $a = 1$) and clock frequencies.

Table 2.22: spl2 switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
‘0’ (J)	3.37E-24	1.13E-23	3.51E-23	7.48E-23	1.55E-22	4.81E-22	2.06E-21
‘1’ (J)	3.37E-24	1.13E-23	3.03E-23	7.00E-23	1.50E-22	4.76E-22	2.05E-21

2.2 Sequential Cells

2.2.1 QFPL

The quantum-flux-parametron latch (`qfpl`) is a native memory element in the AQFP library. It consists of two `bfr` cells that serve as write gates and a single bi-stable `storage_gate` that is based on the `bfr` core design with no ac excitation line but rather a single dc bias to indefinitely store the state of the `qfpl`. `storage_gate` is introduced in the sub-cells section.

Symbol

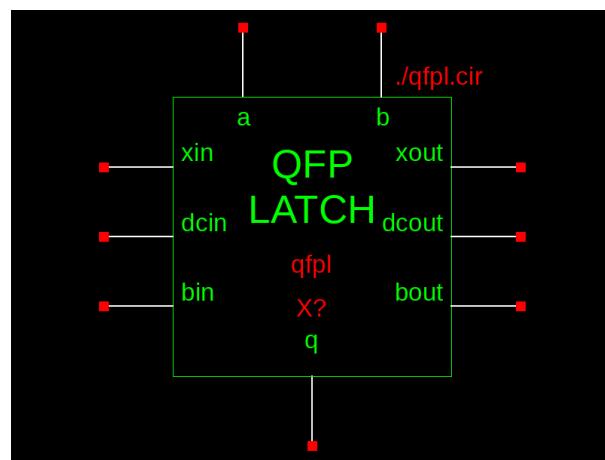


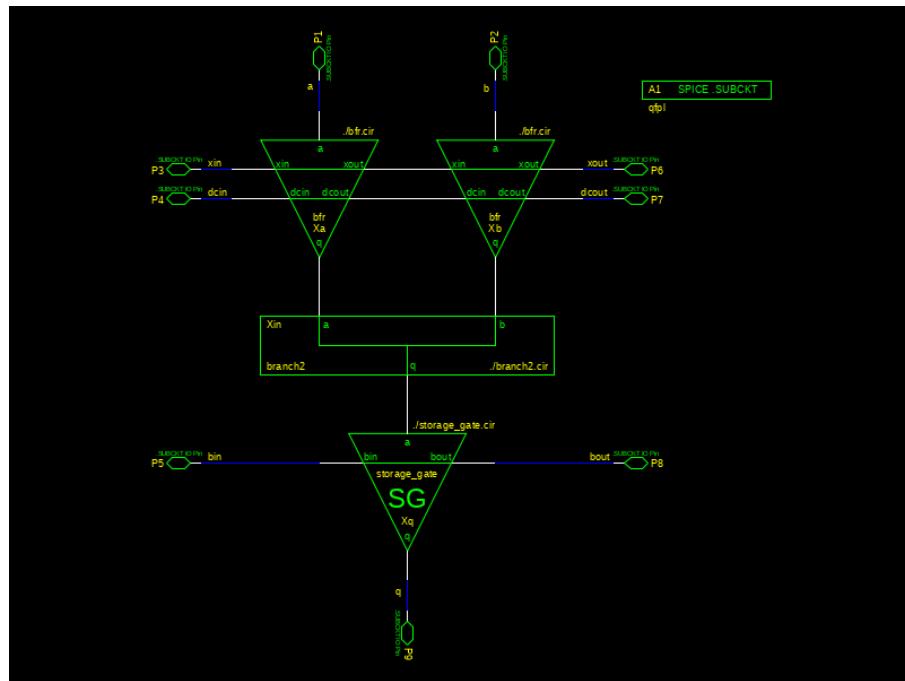
Figure 2.56: `qfpl` symbol.

Table 2.23: `qfpl` pin list.

Pin	Description
A	data input
B	data input
XIN	serial clock input
DCIN	dc offset input
BIN	dc bias input for storage gate
Q	data output
XOUT	serial clock output
DCOUT	dc offset output
BOUT	dc bias input for storage gate

Table 2.24: qfpl truth table.

A	B	Q
0	0	Write '0'
0	1	No change
1	0	No change
1	1	Write '1'

Schematic**Figure 2.57:** qfpl schematic.

Layout

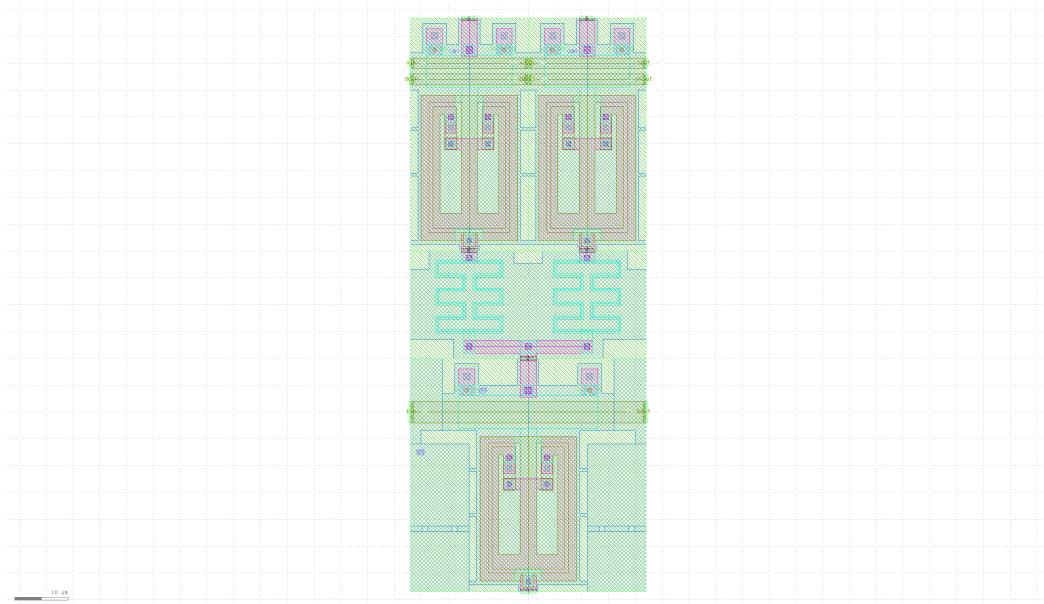


Figure 2.58: qfpl layout.

Analog model

```

1  .SUBCKT storage_gate a bin bout q
2  Kiout Lin Lout Kiout
3  Kbout Lb Lout Kbout
4  .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
5  .param Lin=1.63p
6  .param Lb=9.41p
7  .param L1=2.21p
8  .param L2=L1
9  .param Lq=8.31p
10 .param Lout=30.46p
11 .param B1=0.5
12 .param B2=B1
13 .param Kb1=-0.2842
14 .param Kb2=Kb1
15 .param Kout=-0.6421
16 .param Kbout=-1.861E-6
17 .param Kiout=5.199E-3
18 .param Kib=-1.197E-4
19
20   Kib Lin Lb Kib
21   Kout Lout Lq Kout
22   Kb2 Lb L2 Kb2
23   Kb1 Lb L1 Kb1
24   Lin a 2 Lin
25   B2 3 0 B2 jjmit area=B2
26   B1 1 0 B1 jjmit area=B1
27   Lout 0 q Lout
28   Lq 2 0 Lq
29   L2 3 2 L2
30   L1 2 1 L1
31   Lb bin bout Lb
32   .ends storage_gate
33   .SUBCKT branch2 a b q
34   .param La=12.4p
35   .param Lb=12.4p

```

```

36 .param Lq=0.17p
37 Lq 1 q Lq
38 Lb b 1 Lb
39 La a 1 La
40 .ends branch2
41 .SUBCKT bfr a xin dcin xout dcout q
42 Kid Lin Ld Kid
43 Kiout Lin Lout Kiout
44 Kix Lin Lx Kix
45 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
46 .param Lin=1.56p
47 .param Lx=6.53p
48 .param Ld=6.56p
49 .param L1=1.56p
50 .param L2=L1
51 .param Lq=8.25p
52 .param Lout=30.55p
53 .param B1=0.5
54 .param B2=B1
55 .param Kxd=0.2916
56 .param Kx1=-0.2461
57 .param Kx2=Kx1
58 .param Kd1=-0.1916
59 .param Kd2=Kd1
60 .param Kout=-0.6400
61 .param Kxout=-8.116E-6
62 .param Kdout=-2.873E-5
63 .param Kiout=2.201E-3
64 .param Kix=-1.145E-4
65 .param Kid=-1.063E-5
66
67 Kxout Lx Lout Kxout
68 Kdout Ld Lout Kdout
69 Kout Lout Lq Kout
70 Kd2 Ld L2 Kd2
71 Kx2 Lx L2 Kx2
72 Kd1 Ld L1 Kd1
73 Kx1 Lx L1 Kx1
74 Kxd Lx Ld Kxd
75 Lin a 2 Lin
76 B2 3 0 B2 jjmit area=B2
77 B1 1 0 B1 jjmit area=B1
78 Lout 0 q Lout
79 Lq 2 0 Lq
80 L2 3 2 L2
81 L1 2 1 L1
82 Ld dcin dcout Ld
83 Lx xin xout Lx
84 .ends bfr
85 .SUBCKT qfpl a b xin dcin bin xout dcout bout q
86 Xb b 4 5 xout dcout 3 bfr
87 Xa a xin dcin 4 5 2 bfr
88 Xin 2 3 1 branch2
89 Xq 1 bin bout q storage_gate
90 .ends qfpl

```

Listing 2.23: qfpl netlist (.cir).

Simulation result

Simulation waveform of the qfpl. A 4-stage buffer chain is connected both input A and B. Each input is applied a test pattern with a peak-to-peak amplitude is $\pm 5 \mu\text{A}$. AC amplitude is $664 \mu\text{A}$ at 5GHz , and DC is set to $862 \mu\text{A}$. Input A has a pattern of 010110001100 and B has a pattern of 001101001100. This results in an operation pattern of write ‘0’, hold, hold, write ‘1’, hold, hold, write ‘0’, write ‘0’, write ‘1’, write ‘1’, write ‘0’, write ‘0’. The resulting data output is 000111001100 as shown in the simulation waveform.

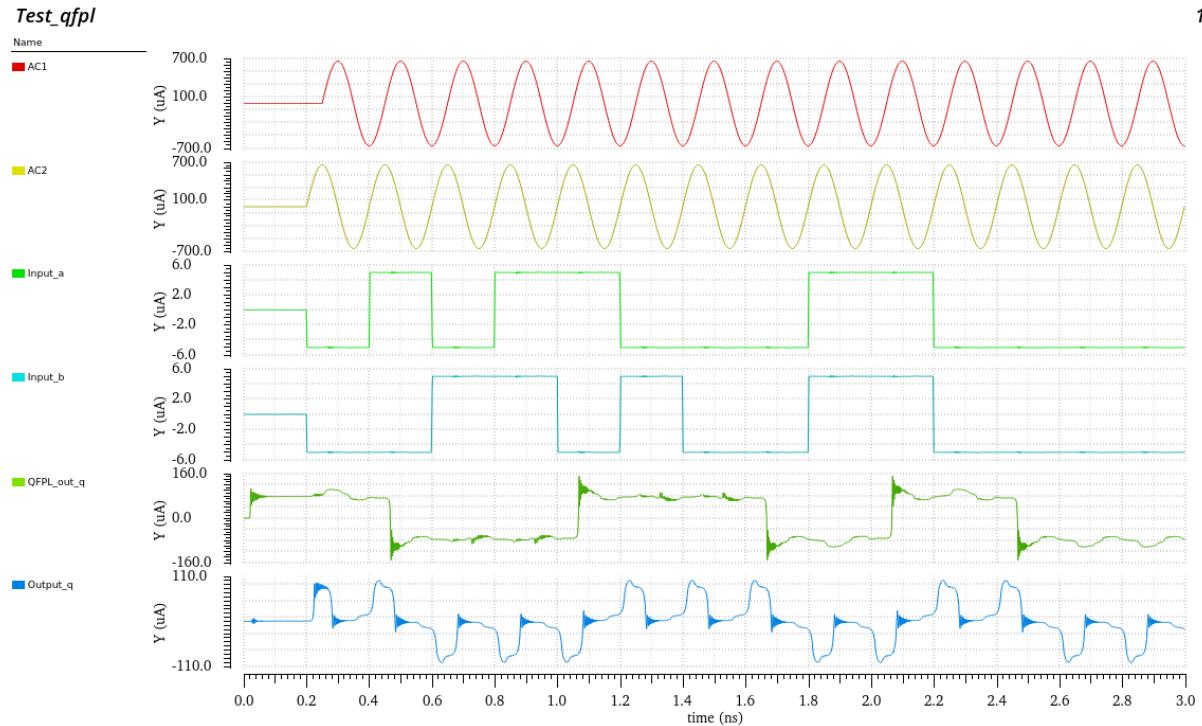


Figure 2.59: qfpl analog waveform.

Digital model

```

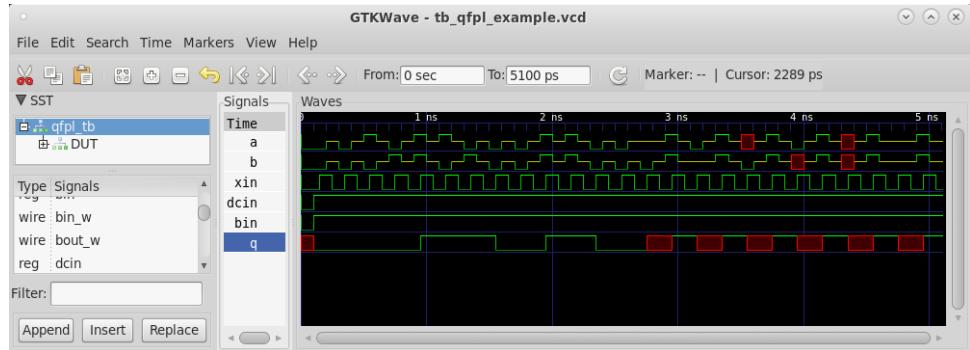
1  `timescale 1ps/10fs
2
3  module qfpl(a, b, q, xin, xout, dcin, dc当地, bin, bout);
4
5      input a, b;
6      inout xin, xout, dcin, dc当地, bin, bout;
7      output q;
8
9      wire not_a, not_b;
10     reg q;
11
12     //use DC bias bidirection model for bin/bout
13     biasDC #(8) I1(bin, bout);
14     biasDir_b I2(xin, xout, dcin, dc当地, gatex);
15
16     assign not_a = !a;
17     assign not_b = !b;
18
19     initial begin
20         $timeformat(-12, 1, "ps", 8); // time format
21         q = 1'bz;
22     end
23
24     specify
25         specparam d_clk    = 5;
26         specparam clk_d   = 50;
27
28         $setup(posedge a && a, posedge gatex, d_clk);
29         $setup(negedge a && not_a, posedge gatex, d_clk);
30         $hold(posedge gatex, negedge a && a, clk_d);
31         $hold(posedge gatex, posedge a && not_a, clk_d);
32
33         $setup(posedge b && b, posedge gatex, d_clk);
34         $setup(negedge b && not_b, posedge gatex, d_clk);

```

```

35     $hold(posedge gatex, negedge b && b, clk_d);
36     $hold(posedge gatex, posedge b && not_b, clk_d);
37   endspecify
38
39   always @(posedge gatex) begin
40     if (a==b && bin==1'b1) begin
41       //write case (using logical compare == so only valid logic values pass)
42       q <= a;
43
44     //no reset in QFPL, state is held indefinitely unlike other AQFP cells
45
46   end else if (a!=b && bin==1'b1) begin
47     //keep state
48   end else begin
49     //Possible errors:
50     //If QFPL bias is not initialized to '1' before operation
51     //then the QFPL should not work. (Detects misconnect of bias).
52
53     //If data are not valid (still Z or X), then a timing error occurred
54     //because of logical equality even 'z=='z' or 'x=='x' will not pass)
55
56     q <= 1'bx;
57   end
58 end
59 endmodule

```

Listing 2.24: qfpl Verilog model code.**Figure 2.60:** qfpl digital waveform. HDL '1': AQFP '1'; HDL '0': AQFP '0'; HDL 'z': inactive; HDL 'x': error.

Switching energy

Different energy consumption results based on different data input patterns ($ab = 00$, $ab = 01$, $ab = 10$ and $ab = 11$) and clock frequencies.

Table 2.25: qfpl switching energy table.

Input Logic	Clock Rate (GHz)						
	0.1	0.2	0.5	1	2	5	10
'00' (J)	1.0023E-20	1.0046E-20	1.0117E-20	1.0236E-20	1.0476E-20	1.1436E-20	1.48E-20
'01' (J)	2.28E-23	4.64E-23	1.172E-22	2.36E-22	4.76E-22	1.436E-21	4.8E-21
'10' (J)	2.28E-23	4.64E-23	1.172E-22	2.36E-22	4.76E-22	1.436E-21	4.8E-21
'11' (J)	1.0023E-20	1.0046E-20	1.0117E-20	1.0236E-20	1.0476E-20	1.1436E-20	1.48E-20

2.2.2 NDRO_QFPL

There are two design approaches to achieve an AQFP non-destructive read-out (ndro): rewriting the boolean logic function of a qfpl by adding proper combinational logic (ndro_qfpl), or using pure combination logic gates with a feedback loop (ndro_bf). ndro_qfpl consists of two

Symbol

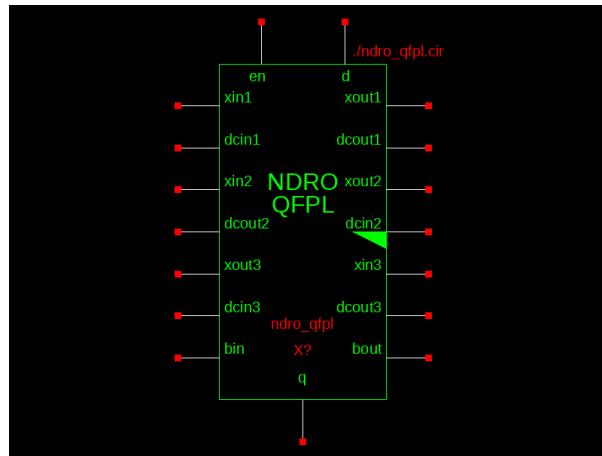


Figure 2.61: ndro_qfpl symbol.

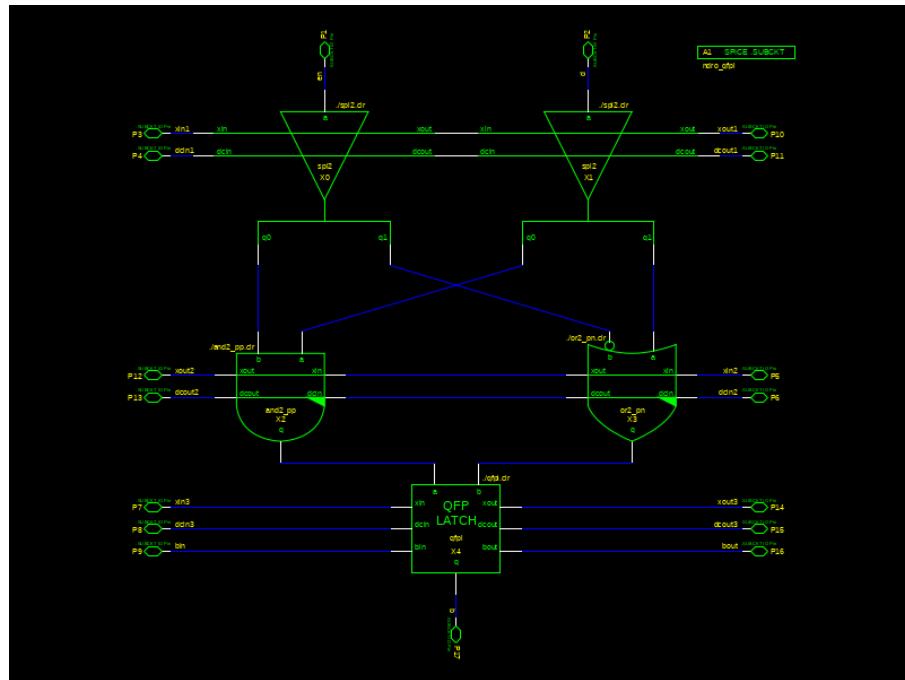
Table 2.26: ndro_qfpl pin list.

Pin	Description
EN	data input
D	data input
XIN1	serial clock input
XIN2	serial clock input
XIN3	serial clock input
DCIN1	dc offset input
DCIN2	dc offset input
DCIN3	dc offset input
BIN	dc bias input for storage gate
Q	data output
XOUT1	serial clock output
XOUT2	serial clock output
XOUT3	serial clock output
DCOUT1	dc offset output
DCOUT2	dc offset output
DCOUT3	dc offset output
BOUT	dc bias input for storage gate

Table 2.27: ndro_qfpl truth table.

E	D	Q
0	0	No Change
0	1	No change
1	0	Write '0'
1	1	Write '1'

Schematic

**Figure 2.62:** ndro_qfpl schematic.

Layout

This is a multi-phase macro cell so we cannot fix the location of the excitation lines as it depends on how the `ndro_qfpl` fits in a larger circuit design. Thus there is no fixed layout, as it should be placed dynamically on a case-by-case basis.

Analog model

```

1 .SUBCKT storage_gate a bin bout q
2   Kiout Lin Lout Kiout
3   Kbout Lb Lout Kbout
4   .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
5   .param Lin=1.63p
6   .param Lb=9.41p
7   .param L1=2.21p
8   .param L2=L1
9   .param Lq=8.31p
10  .param Lout=30.46p
11  .param B1=0.5
12  .param B2=B1

```

```

13 .param Kb1=-0.2842
14 .param Kb2=Kb1
15 .param Kout=-0.6421
16 .param Kbout=-1.861E-6
17 .param Kiout=5.199E-3
18 .param Kib=-1.197E-4
19
20 Kib Lin Lb Kib
21 Kout Lout Lq Kout
22 Kb2 Lb L2 Kb2
23 Kb1 Lb L1 Kb1
24 Lin a 2 Lin
25 B2 3 0 B2 jjmit area=B2
26 B1 1 0 B1 jjmit area=B1
27 Lout 0 q Lout
28 Lq 2 0 Lq
29 L2 3 2 L2
30 L1 2 1 L1
31 Lb bin bout Lb
32 .ends storage_gate
33 .SUBCKT branch2 a b q
34 .param La=12.4p
35 .param Lb=12.4p
36 .param Lq=0.17p
37 Lq 1 q Lq
38 Lb b 1 Lb
39 La a 1 La
40 .ends branch2
41 .SUBCKT bfr a xin dcin xout dcout q
42 Kid Lin Ld Kid
43 Kiout Lin Lout Kiout
44 Kix Lin Lx Kix
45 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
46 .param Lin=1.56p
47 .param Lx=6.53p
48 .param Ld=6.56p
49 .param L1=1.56p
50 .param L2=L1
51 .param Lq=8.25p
52 .param Lout=30.55p
53 .param B1=0.5
54 .param B2=B1
55 .param Kxd=0.2916
56 .param Kx1=-0.2461
57 .param Kx2=Kx1
58 .param Kd1=-0.1916
59 .param Kd2=Kd1
60 .param Kout=-0.6400
61 .param Kxout=-8.116E-6
62 .param Kdout=-2.873E-5
63 .param Kiout=2.201E-3
64 .param Kix=-1.145E-4
65 .param Kid=-1.063E-5
66
67 Kxout Lx Lout Kxout
68 Kdout Ld Lout Kdout
69 Kout Lout Lq Kout
70 Kd2 Ld L2 Kd2
71 Kx2 Lx L2 Kx2
72 Kd1 Ld L1 Kd1
73 Kx1 Lx L1 Kx1
74 Kxd Lx Ld Kxd
75 Lin a 2 Lin
76 B2 3 0 B2 jjmit area=B2
77 B1 1 0 B1 jjmit area=B1
78 Lout 0 q Lout
79 Lq 2 0 Lq
80 L2 3 2 L2
81 L1 2 1 L1
82 Ld dcin dcout Ld

```

```

83  Lx xin xout Lx
84  .ends bfr
85  .SUBCKT qfpl a b xin dcin bin xout dcout bout q
86  Xb b 4 5 xout dcout 3 bfr
87  Xa a xin dcin 4 5 2 bfr
88  Xin 2 3 1 branch2
89  Xq 1 bin bout q storage_gate
90  .ends qfpl
91  .SUBCKT bias_pair_11um a c b d
92  .param L0=3.60p
93  .param L1=3.60p
94
95  L1 c d L1
96  L0 a b L0
97  .ends bias_pair_11um
98  .SUBCKT spl2 a xin dcin xout dcout q0 q1
99  Xq q0 q1 1 branch2
100 X1 xin dcin 4 5 bias_pair_11um
101 X2 2 3 xout dcout bias_pair_11um
102 Xa a 4 5 2 3 1 bfr
103 .ends spl2
104 .SUBCKT branch3 a b c q
105 .param La=13.60p
106 .param Lb=10.30p
107 .param Lc=13.60p
108 .param Lq=0.28p
109 Lq 1 q Lq
110 Lc c 1 Lc
111 Lb b 1 Lb
112 La a 1 La
113 .ends branch3
114 .SUBCKT const0 xin dcin xout dcout q
115 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
116 .param Lx=6.62p
117 .param Ld=6.60p
118 .param L1=1.61p
119 .param L2=1.76p
120 .param Lq=8.19p
121 .param Lout=30.65p
122 .param B1=0.5
123 .param B2=B1
124 .param Kxd=0.2972
125 .param Kx1=-0.2265
126 .param Kx2=-0.2602
127 .param Kd1=-0.1688
128 .param Kd2=-0.2088
129 .param Kout=-0.6452
130 .param Kxout=-2.545E-4
131 .param Kdout=-5.447E-4
132 Kxout Lx Lout Kxout
133 Kdout Ld Lout Kdout
134 Kout Lout Lq Kout
135 Kd2 Ld L2 Kd2
136 Kx2 Lx L2 Kx2
137 Kd1 Ld L1 Kd1
138 Kx1 Lx L1 Kx1
139 Kxd Lx Ld Kxd
140 B2 3 0 B2 jjmit area=B2
141 B1 1 0 B1 jjmit area=B1
142 Lout 0 q Lout
143 Lq 2 0 Lq
144 L2 3 2 L2
145 L1 2 1 L1
146 Ld dcin dcout Ld
147 Lx xin xout Lx
148 .ends const0
149 .SUBCKT and2_pp a b xin dcin xout dcout q
150 X0 2 3 5 6 7 const0
151 Xq 1 7 4 q branch3
152 Xc b 5 6 xout dcout 4 bfr

```

```

153 | Xa a xin dcin 2 3 1 bfr
154 | .ends and2_pp
155 | .SUBCKT const1 xin dcin xout dcout q
156 | X0 xout dcout xin dcin q const0
157 | .ends const1
158 | .SUBCKT inv a xin dcin xout dcout q
159 | Kid Lin Ld Kid
160 | Kiout Lin Lout Kiout
161 | Kix Lin Lx Kix
162 | .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
163 | .param Lin=1.56p
164 | .param Lx=6.54p
165 | .param Ld=6.57p
166 | .param L1=1.56p
167 | .param L2=L1
168 | .param Lq=7.14p
169 | .param Lout=30.24p
170 | .param B1=0.5
171 | .param B2=B1
172 | .param Kxd=0.2912
173 | .param Kx1=-0.2460
174 | .param Kx2=Kx1
175 | .param Kd1=-0.1909
176 | .param Kd2=Kd1
177 | .param Kout=0.5917
178 | .param Kxout=1.335E-5
179 | .param Kdout=2.443E-5
180 | .param Kiout=1.172E-3
181 | .param Kix=-2.712E-4
182 | .param Kid=2.605E-4
183 |
184 | Kxout Lx Lout Kxout
185 | Kdout Ld Lout Kdout
186 | Kout Lout Lq Kout
187 | Kd2 Ld L2 Kd2
188 | Kx2 Lx L2 Kx2
189 | Kd1 Ld L1 Kd1
190 | Kx1 Lx L1 Kx1
191 | Kxd Lx Ld Kxd
192 | Lin a 2 Lin
193 | B2 3 0 B2 jjmit area=B2
194 | B1 1 0 B1 jjmit area=B1
195 | Lout 0 q Lout
196 | Lq 2 0 Lq
197 | L2 3 2 L2
198 | L1 2 1 L1
199 | Ld dcin dcout Ld
200 | Lx xin xout Lx
201 | .ends inv
202 | .SUBCKT or2_pn a b xin dcin xout dcout q
203 | Xa a xin dcin 6 7 1 bfr
204 | Xc b 4 5 xout dcout 3 inv
205 | X1 6 7 4 5 2 const1
206 | Xq 1 2 3 q branch3
207 | .ends or2_pn
208 | .SUBCKT ndro_qfpl en d xin1 dcin1 xin2 dcin2 xin3 dcin3 bin xout1 dcout1 xout2 dcout2
209 | ↛ xout3 dcout3 bout q
210 | X3 8 6 xin2 dcin2 9 10 2 or2_pn
211 | X2 7 5 9 10 xout2 dcout2 1 and2_pp
212 | X1 d 3 4 xout1 dcout1 7 8 spl2
213 | X0 en xin1 dcin1 3 4 5 6 spl2
214 | X4 1 2 xin3 dcin3 bin xout3 dcout3 bout q qfpl
     .ends ndro_qfpl

```

Listing 2.25: ndro_qfpl netlist (.cir).

Simulation result

Simulation waveform of the `ndro_qfp1`. A 4-stage buffer chain is connected both input EN and D. Each input is applied a test pattern with a peak-to-peak amplitude is $\pm 5 \mu\text{A}$. AC amplitude is $664 \mu\text{A}$ at 5 GHz, and DC is set to $862 \mu\text{A}$. Input EN has a pattern of 10001000100 and D has a pattern of 01011010010. This results in an operation pattern of write ‘0’, hold, hold, hold, write ‘1’, hold, hold, hold, write ‘0’, hold, hold. The resulting data output is 00001111000 as shown in the simulation waveform.

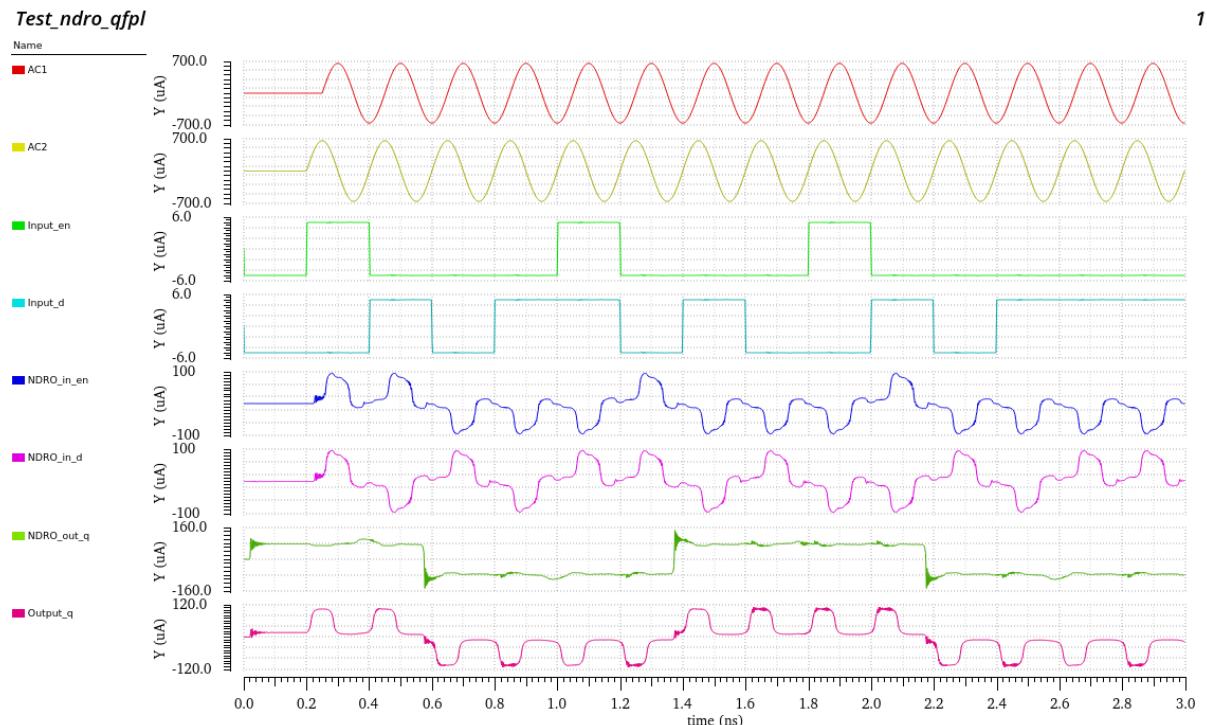


Figure 2.63: `ndro_qfp1` analog waveform.

Digital model

```

1  `timescale 1ps/10fs
2
3  module ndro_qfpl(en, d, xin1, xin2, xin3, dcin1, dcin2, dcin3, bin,
4      q, xout1, xout2, xout3, dout1, dout2, dout3, bout);
5
6  input en, d;
7  inout xin1, xin2, xin3, dcin1, dcin2, dcin3, bin;
8  inout xout1, xout2, xout3, dout1, dout2, dout3, bout;
9  output q;
10 wire s0_q0, s0_q1, s1_q0, s1_q1, and_q, or_q;
11 wire xph1, xph2, dcph1, dcph2;
12
13 //call submodule
14 spl2    s0(.a(en), .xin(xin1), .dcin(dcin1), .q0(s0_q0), .q1(s0_q1), .xout(xph1), .dcout
15     ↪ (dcph1));
16 spl2    s1(.a(d), .xin(xph1), .dcin(dcph1), .q0(s1_q0), .q1(s1_q1), .xout(xout1), .dcout
17     ↪ (dcout1));
18 and2_pp a0(.a(s1_q0), .b(s0_q0), .xin(xph2), .dcin(dcph2), .q(and_q), .xout(xout2), .
19     ↪ dcout(dcout2));
20 or2_pn o0(.a(s1_q1), .b(s0_q1), .xin(xin2), .dcin(dcin2), .q(or_q), .xout(xph2), .dcout
21     ↪ (dcph2));
22 qfpl    q0(.a(and_q), .b(or_q), .xin(xin3), .dcin(dcin3), .bin(bin), .q(q), .xout(xout3)
23     ↪ , .dcout(dcout3), .bout(bout));

```

Listing 2.26: `ndro_qfpl` Verilog model code.

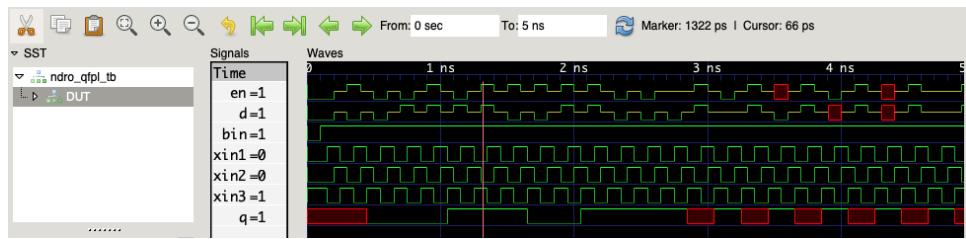


Figure 2.64: `ndro_qfpl` digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.

2.2.3 NDRO_FBF

Non-destructive read-out with feedback (ndro_fb), previously named as ‘D-Flip-Flop (DFF)’, is considered as a sequential logic element built from the developed AQFP combinational logic cells. Here we present the symbol view, block diagram (schematic), netlist and simulation results from a test circuit.

Symbol

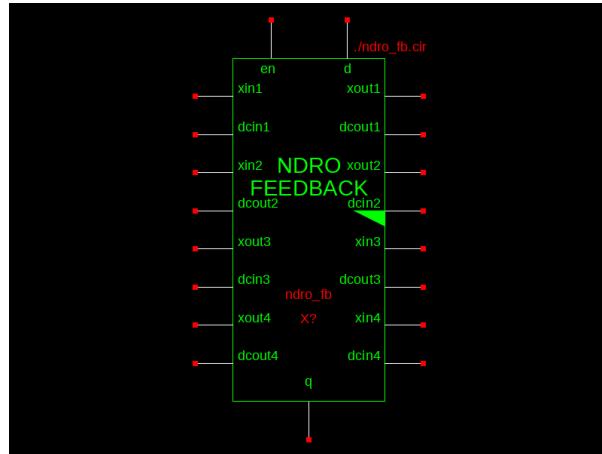


Figure 2.65: ndro_fb symbol.

Table 2.28: ndro_fb pin list.

Pin	Description
E	data input
D	data input
XIN1	serial clock input
XIN2	serial clock input
XIN3	serial clock input
XIN4	serial clock input
DCIN1	dc offset input
DCIN2	dc offset input
DCIN3	dc offset input
DCIN4	dc offset input
Q	data output
XOUT1	serial clock output
XOUT2	serial clock output
XOUT3	serial clock output
XOUT4	serial clock output
DCOUT1	dc offset output
DCOUT2	dc offset output
DCOUT3	dc offset output
DCOUT4	dc offset output

Schematic

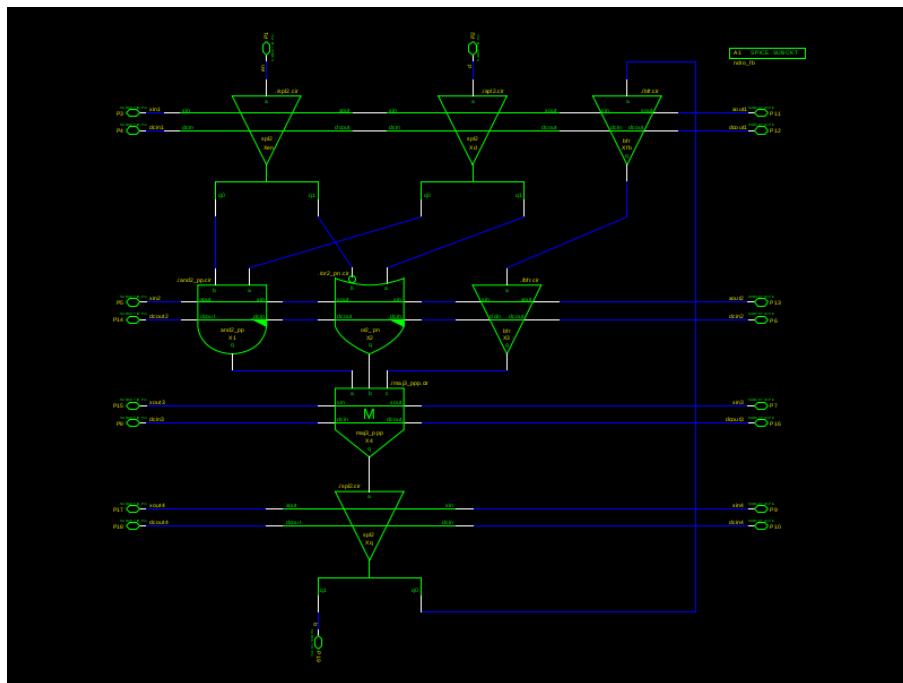


Figure 2.66: ndro_fb schematic.

Layout

This is a multi-phase macro cell so we cannot fix the location of the excitation lines as it depends on how the `ndro_fb` fits in a larger circuit design. Thus there is no fixed layout, as it should be placed dynamically on a case-by-case basis.

Analog model

```

1 | .SUBCKT bfr a xin dcin xout dcout q
2 | Kid Lin Ld Kid
3 | Kiout Lin Lout Kiout
4 | Kix Lin Lx Kix
5 | .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
6 | .param Lin=1.56p
7 | .param Lx=6.53p
8 | .param Ld=6.56p
9 | .param L1=1.56p
10 | .param L2=L1
11 | .param Lq=8.25p
12 | .param Lout=30.55p
13 | .param B1=0.5
14 | .param B2=B1
15 | .param Kxd=0.2916
16 | .param Kx1=-0.2461
17 | .param Kx2=Kx1
18 | .param Kd1=-0.1916
19 | .param Kd2=Kd1
20 | .param Kout=-0.6400
21 | .param Kxout=-8.116E-6
22 | .param Kdout=-2.873E-5
23 | .param Kiout=2.201E-3
24 | .param Kix=-1.145E-4

```

```

25 .param Kid=-1.063E-5
26
27 Kxout Lx Lout Kxout
28 Kdout Ld Lout Kdout
29 Kout Lout Lq Kout
30 Kd2 Ld L2 Kd2
31 Kx2 Lx L2 Kx2
32 Kd1 Ld L1 Kd1
33 Kx1 Lx L1 Kx1
34 Kxd Lx Ld Kxd
35 Lin a 2 Lin
36 B2 3 0 B2 jjmit area=B2
37 B1 1 0 B1 jjmit area=B1
38 Lout 0 q Lout
39 Lq 2 0 Lq
40 L2 3 2 L2
41 L1 2 1 L1
42 Ld dcin dcout Ld
43 Lx xin xout Lx
44 .ends bfr
45 .SUBCKT branch3 a b c q
46 .param La=13.60p
47 .param Lb=10.30p
48 .param Lc=13.60p
49 .param Lq=0.28p
50 Lq 1 q Lq
51 Lc c 1 Lc
52 Lb b 1 Lb
53 La a 1 La
54 .ends branch3
55 .SUBCKT const0 xin dcin xout dcout q
56 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
57 .param Lx=6.62p
58 .param Ld=6.60p
59 .param L1=1.61p
60 .param L2=1.76p
61 .param Lq=8.19p
62 .param Lout=30.65p
63 .param B1=0.5
64 .param B2=B1
65 .param Kxd=0.2972
66 .param Kx1=-0.2265
67 .param Kx2=-0.2602
68 .param Kd1=-0.1688
69 .param Kd2=-0.2088
70 .param Kout=-0.6452
71 .param Kxout=-2.545E-4
72 .param Kdout=-5.447E-4
73 Kxout Lx Lout Kxout
74 Kdout Ld Lout Kdout
75 Kout Lout Lq Kout
76 Kd2 Ld L2 Kd2
77 Kx2 Lx L2 Kx2
78 Kd1 Ld L1 Kd1
79 Kx1 Lx L1 Kx1
80 Kxd Lx Ld Kxd
81 B2 3 0 B2 jjmit area=B2
82 B1 1 0 B1 jjmit area=B1
83 Lout 0 q Lout
84 Lq 2 0 Lq
85 L2 3 2 L2
86 L1 2 1 L1
87 Ld dcin dcout Ld
88 Lx xin xout Lx
89 .ends const0
90 .SUBCKT and2_pp a b xin dcin xout dcout q
91 X0 2 3 5 6 7 const0
92 Xq 1 7 4 q branch3
93 Xc b 5 6 xout dcout 4 bfr
94 Xa a xin dcin 2 3 1 bfr

```

```

95  .ends and2_pp
96  .SUBCKT bias_pair_11um a c b d
97  .param L0=3.60p
98  .param L1=3.60p
99
100 L1 c d L1
101 L0 a b L0
102 .ends bias_pair_11um
103 .SUBCKT branch2 a b q
104 .param La=12.4p
105 .param Lb=12.4p
106 .param Lq=0.17p
107 Lq 1 q Lq
108 Lb b 1 Lb
109 La a 1 La
110 .ends branch2
111 .SUBCKT spl2 a xin dcin xout dcout q0 q1
112 Xq q0 q1 1 branch2
113 X1 xin dcin 4 5 bias_pair_11um
114 X2 2 3 xout dcout bias_pair_11um
115 Xa a 4 5 2 3 1 bfr
116 .ends spl2
117 .SUBCKT maj3_ppp a b c xin dcin xout dcout q
118 Xq 1 4 7 q branch3
119 Xc c 5 6 xout dcout 7 bfr
120 Xb b 2 3 5 6 4 bfr
121 Xa a xin dcin 2 3 1 bfr
122 .ends maj3_ppp
123 .SUBCKT const1 xin dcin xout dcout q
124 X0 xout dcout xin dcin q const0
125 .ends const1
126 .SUBCKT inv a xin dcin xout dcout q
127 Kid Lin Ld Kid
128 Kiout Lin Lout Kiout
129 Kix Lin Lx Kix
130 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
131 .param Lin=1.56p
132 .param Lx=6.54p
133 .param Ld=6.57p
134 .param L1=1.56p
135 .param L2=L1
136 .param Lq=7.14p
137 .param Lout=30.24p
138 .param B1=0.5
139 .param B2=B1
140 .param Kxd=0.2912
141 .param Kx1=-0.2460
142 .param Kx2=Kx1
143 .param Kd1=-0.1909
144 .param Kd2=Kd1
145 .param Kout=0.5917
146 .param Kxout=1.335E-5
147 .param Kdout=2.443E-5
148 .param Kiout=1.172E-3
149 .param Kix=-2.712E-4
150 .param Kid=2.605E-4
151
152 Kxout Lx Lout Kxout
153 Kdout Ld Lout Kdout
154 Kout Lout Lq Kout
155 Kd2 Ld L2 Kd2
156 Kx2 Lx L2 Kx2
157 Kd1 Ld L1 Kd1
158 Kx1 Lx L1 Kx1
159 Kxd Lx Ld Kxd
160 Lin a 2 Lin
161 B2 3 0 B2 jjmit area=B2
162 B1 1 0 B1 jjmit area=B1
163 Lout 0 q Lout
164 Lq 2 0 Lq

```

```

165  L2 3 2 L2
166  L1 2 1 L1
167  Ld dcin dcout Ld
168  Lx xin xout Lx
169  .ends inv
170  .SUBCKT or2_pn a b xin dcin xout dcout q
171  Xa a xin dcin 6 7 1 bfr
172  Xc b 4 5 xout dcout 3 inv
173  X1 6 7 4 5 2 const1
174  Xq 1 2 3 q branch3
175  .ends or2_pn
176  .SUBCKT ndro_fb en d xin1 dcin1 xin2 dcin2 xin3 dcin3 xin4 dcin4 xout1 dcout1 xout2
     ↪ dcout2 xout3 dcout3 xout4 dcout4 q
177  X2 8 4 14 15 9 10 18 or2_pn
178  X4 11 18 13 xout3 dcin3 xin3 dcout3 17 maj3_ppp
179  Xq 17 xin4 dcin4 xout4 dcout4 16 q spl2
180  Xfb 16 5 6 xout1 dcout1 12 bfr
181  X3 12 14 15 xout2 dcin2 13 bfr
182  X1 7 3 9 10 xin2 dcout2 11 and2_pp
183  Xd d 1 2 5 6 7 8 spl2
184  Xen en xin1 dcin1 1 2 3 4 spl2
185  .ends ndro_fb

```

Listing 2.27: ndro_fb netlist (.cir).

Simulation result

Simulation waveform of a **ndro_fb** DUT (design under test) with 4-stage buffer before the D input and before the EN input. Another 4-stage buffer is placed after the **ndro_fb**'s Q output. Clock frequency is 5 GHz. Signals from top to bottom: AC source 1 (generates phase 1 and 3), AC source 2 (generates phase 2 and 4), the buffer chain input D and EN, output EN from buffer just before **ndro_fb**, output D from buffer just before **ndro_fb** and the output Q from buffer just after **ndro_fb**. Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

When **textttndro_fb** is set to 0 (D=1), data is toggled while E=0 to show data remains as 0, whereas when **textttndro_fb** is set to 1 (D=0), data is toggled while E=0 to show data remains as 1, when **textttndro_fb** is set to 0 again, data is toggled while E=0 to show data remains as 0.

As shown in the waveform, where D = NDRO_in_en = (1)10001001000 and E = NDRO_in_d = (0)010110100101, the output Q is (1)0000011110, correspondingly. The numbers marked in brackets represent the initial random outputs given by the meander structure of AQFP circuit, it takes one clock cycle to propagate data from the input buffer to the last-stage buffer.

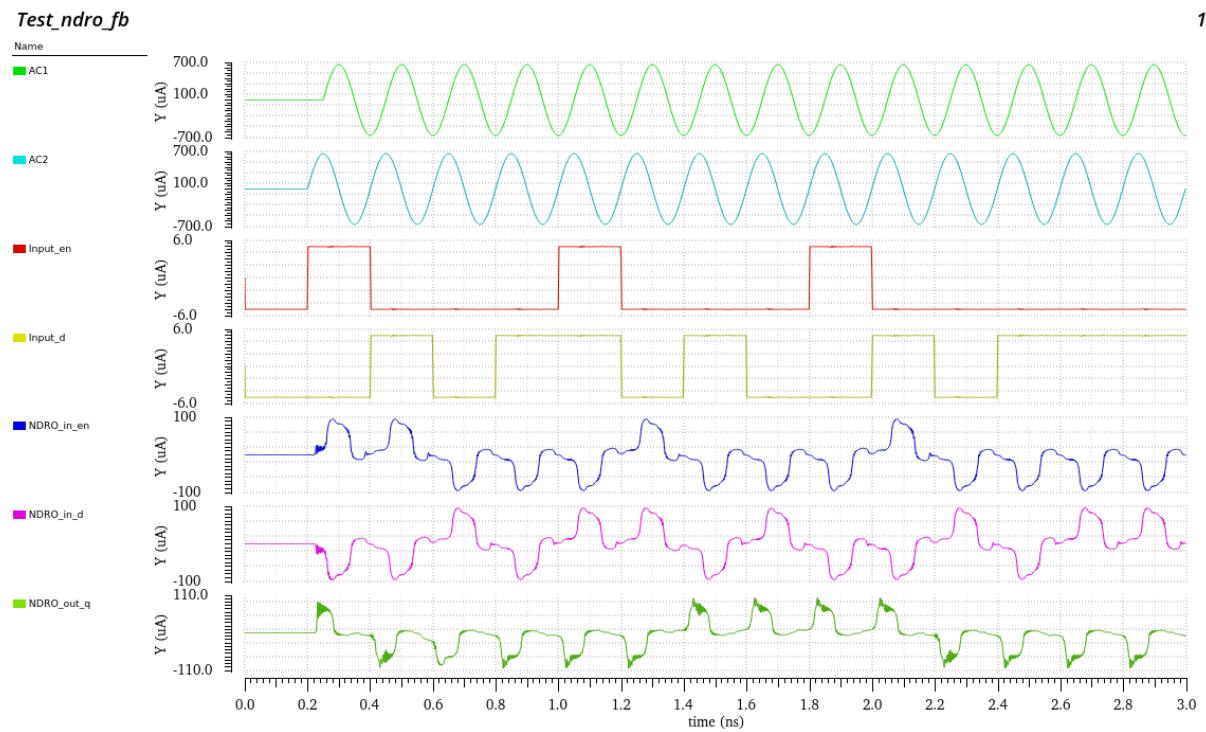


Figure 2.67: ndro_fb analog waveform.

Digital model

```

1  `timescale 1ps/10fs
2
3  module ndro_fb(en, d, xin1, xin2, xin3, xin4, dcin1, dcin2, dcin3, dcin4,
4      q, xout1, xout2, xout3, xout4, dcout1, dcout2, dcout3, dcout4);
5
6  input en, d;
7  inout xin1, xin2, xin3, xin4, dcin1, dcin2, dcin3, dcin4;
8  inout xout1, xout2, xout3, xout4, dcout1, dcout2, dcout3, dcout4;
9  output q;
10 wire s0_q0, s0_q1, s1_q0, s1_q1;
11 wire b0_a, b0_q, and_q, or_q, maj_q;
12 wire xph1_0, xph1_1, xph2_0, xph2_1, dcph1_0, dcph1_1, dcph2_0, dcph2_1;
13
14 //call submodule
15 spl2    s0(.a(en), .xin(xin1), .dcin(dcin1), .q0(s0_q0), .q1(s0_q1), .xout(xph1_0),
16     .dcout(dcph1_0));
17 spl2    s1(.a(d), .xin(xph1_0), .dcin(dcph1_0), .q0(s1_q0), .q1(s1_q1), .xout(xph1_1),
18     .dcout(dcph1_1));
19 bfr    b0(.a(b0_a), .xin(xph1_1), .dcin(dcph1_1), .q(b0_q), .xout(xout1), .dcout(
20     .dcout1));
21 bfr    b1(.a(b0_q), .xin(xph2_1), .dcin(dcph2_1), .q(b1_q), .xout(xout2), .dcout(dcin2
22     ));
23 and2_pp a0(.a(s1_q0), .b(s0_q0), .xin(xph2_0), .dcin(dcph2_0), .q(and_q), .xout(xin2),
24     .dcout(dcout2));
25 or2_pn o0(.a(s1_q1), .b(s0_q1), .xin(xph2_1), .dcin(dcph2_1), .q(or_q), .xout(xph2_0),
26     .dcout(dcph2_0));
27 maj3_ppp m0(.a(and_q), .b(or_q), .c(b1_q), .xin(xin3), .dcin(dcin3), .q(maj_q), .xout(
28     .xout3), .dcout(dcout3));
29 spl2    s2(.a(maj_q), .xin(xin4), .dcin(dcin4), .q0(q), .q1(b0_a), .xout(xout4), .dcout
30     (dcout4));
31
32 endmodule

```

Listing 2.28: ndro_fb Verilog model code.

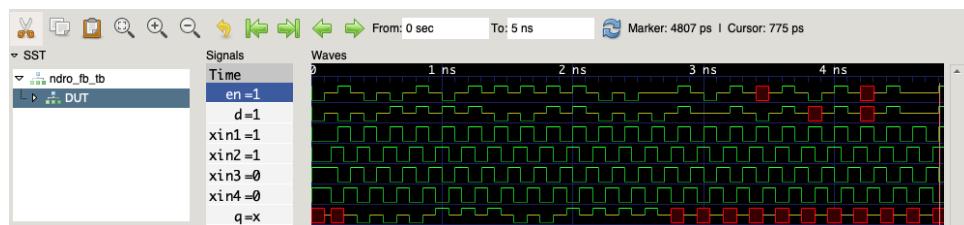


Figure 2.68: ndro_fb digital waveform. HDL ‘1’: AQFP ‘1’; HDL ‘0’: AQFP ‘0’; HDL ‘z’: inactive; HDL ‘x’: error.

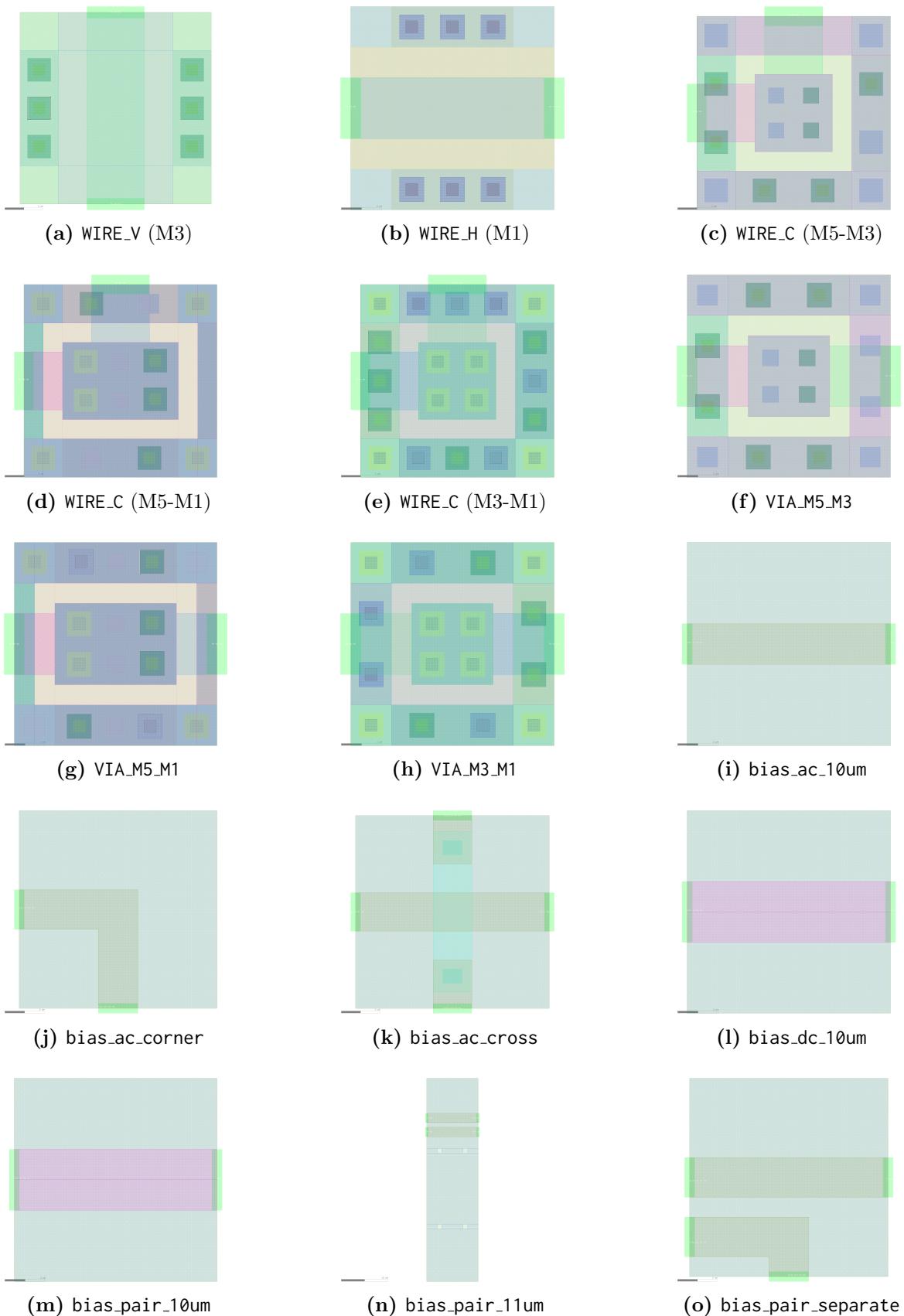
2.3 Interconnect slices

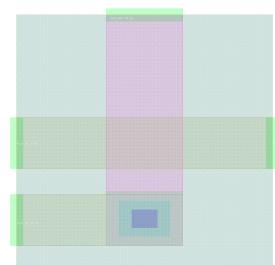
The interconnect of AQFP logic is composed of passive transmission lines (PTLs). PTLs are stripline structures where a signal line has a dedicated ground plane above and below it. RSFQ logic uses striplines to ballistically transport an SFQ pulse from RSFQ cell to another through active transmitter-receiver pairs. AQFP logic uses PTLs to send positive or negative current pulses and the logic cells can connect directly to PTLs without using transmitter-receiver circuits.

The active logic area is dedicated to layers above the M4 ground plane whereas the routing interconnecting exists below the M4 ground plane. The AQFP interconnect slices are $10\text{ }\mu\text{m} \times 10\text{ }\mu\text{m}$ (except for `bias_pair_11um`, which is used in the splitter cells). Table 2.29 summarizes the different types of interconnect cells that are available and Fig. 2.68 shows their corresponding layouts.

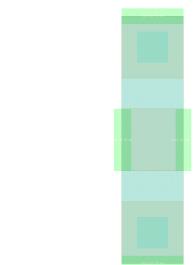
Table 2.29: Summary of interconnect $10\text{ }\mu\text{m} \times 10\text{ }\mu\text{m}$ slices. Each slice is designated with bidirectional ports a-to-b and/or c-to-d. The signal layers for corresponding to those ports are listed as well as the extracted inductances. The shielding ground layers are also listed.

Name	Signal		Ground	pH		Purpose
	a-b	c-d		a-b	c-d	
WIRE_V	M3	N/A	M4 M2	1.07	N/A	Typical north-south interconnect
WIRE_H	M1	N/A	M2 M0	0.98	N/A	Typical east-west interconnect
WIRE_C_M5_M3	M3 M5	N/A	M2	1.86	N/A	Staggered via between M5-M3
WIRE_C_M5_M1	M1 M5	N/A	M0	2.79	N/A	Staggered via between M5-M1
WIRE_C_M3_M1	M3 M1	N/A	M4 M0	1.1	N/A	Staggered via between M3-M1
VIA_M5_M3	M5 M3	N/A	M2	1.58	N/A	Logic cell (M5) to routing (M3)
VIA_M5_M1	M5 M1	N/A	M0	3.43	N/A	Logic cell (M5) to routing (M1)
VIA_M3_M1	M3 M1	N/A	M4 M0	1.21	N/A	Stripline via between M3-M1
bias_ac_10um	M7	N/A	M4	3.17	N/A	AC bias interconnect cell with a length of 10um
bias_ac_corner	M7	N/A	M4	2.7	N/A	AC corner interconnect cell for meandering clock network
bias_ac_cross	M7	M7	M4	3.07	2.76	Large AC crossing interconnect for meandering clock network
bias_dc_10um	M5	N/A	M4	1.48	N/A	DC bias interconnect cell with a length of 10um
bias_pair_10um	M7	M7	M4	2.94	2.99	AC/DC parallel pair with length of 10um
bias_pair_11um	M7	M7	M4	3.6	3.6	AC/DC parallel pair with length of 11um, used in splitter type cells
bias_pair_separate	M7	M7	M4	3.03	1.69	AC/DC parallel pair to perpendicular direction for separation
bias_pair_separate2	M7	M7 M5	M4	3.05	2.6	AC/DC parallel pair to perpendicular direction for separation
VIA_ac_cross	M7	M7	M4	0.34	2.18	Compact crossing via for AC lines
VIA_bias	M7 M5	N/A	M4	1.41	N/A	Compact via for DC lines

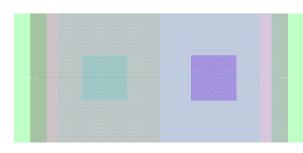




(p) bias_pair_separate2



(q) VIA_ac_cross



(r) VIA_bias

Figure 2.68: Cell-to-cell wire interconnect slices made of stripline-type passive transmission lines. Each slice is 10 $\mu\text{m} \times 10 \mu\text{m}$.

2.4 Off-chip Interface

QDC (QFP-DC-Converter) is an off-chip interface circuit consisting of a dc-SQUID that converts an AQFP signal to an amplified unipolar return-to-zero (RZ) voltage signal for external read out. The qdc test circuit consists of a single buffer before the qdc input and the qdc itself. The qdc has 3 stages: inverter (**inv**), pre-buffer-to-squid (**pre_bsquid**), and buffer-to-squid (**bfr_squid**). **bfr_squid** (previous name: stack) is the dc-SQUID itself coupled to the **pre_bsquid**. The **bfr_squid** creates an output voltage only when it senses negative current at its input. Thus, the qdc includes an inverter so that a positive input results in a high output voltage level. The **pre_bsquid** is an AQFP buffer with the output transformer removed and uses Josephson junctions with large critical current I_c . The **pre_bsquid** is used to amplify the output current level. Here we only introduce the **pre_bsquid** and **bfr_squid**, together with simulation results of a qdc test circuit.

2.4.1 PRE_BSQUID

Symbol

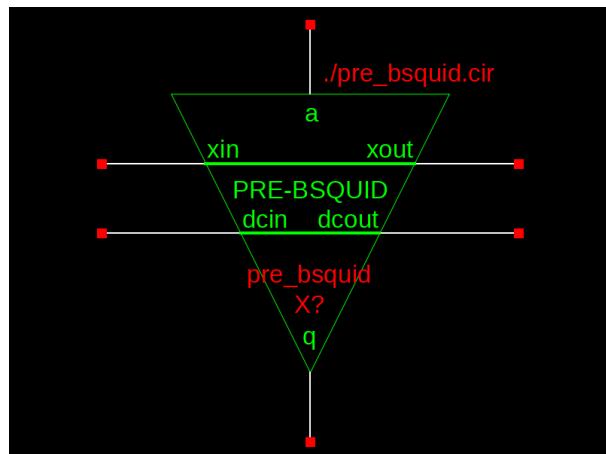


Figure 2.69: pre_bsquid symbol.

Table 2.30: pre_bsquid pin list.

Pin	Description
A	data input
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

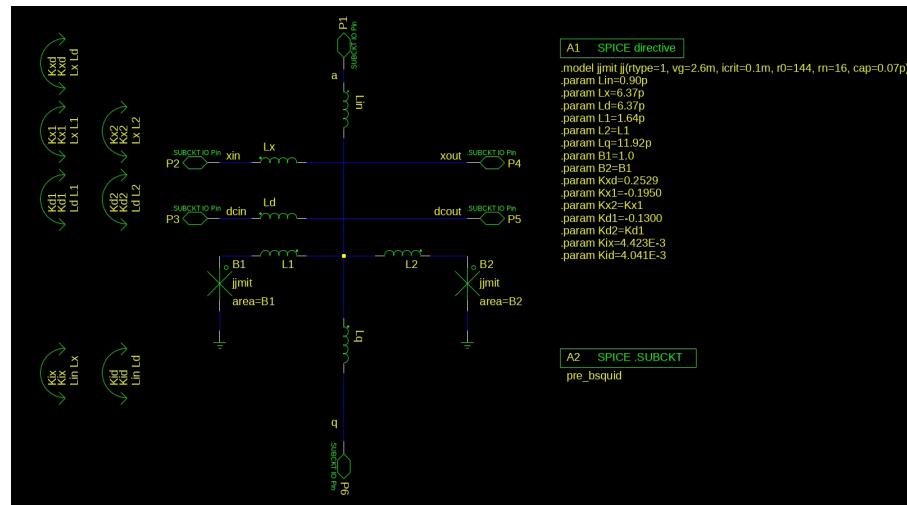


Figure 2.70: pre_bsquid schematic.

Layout

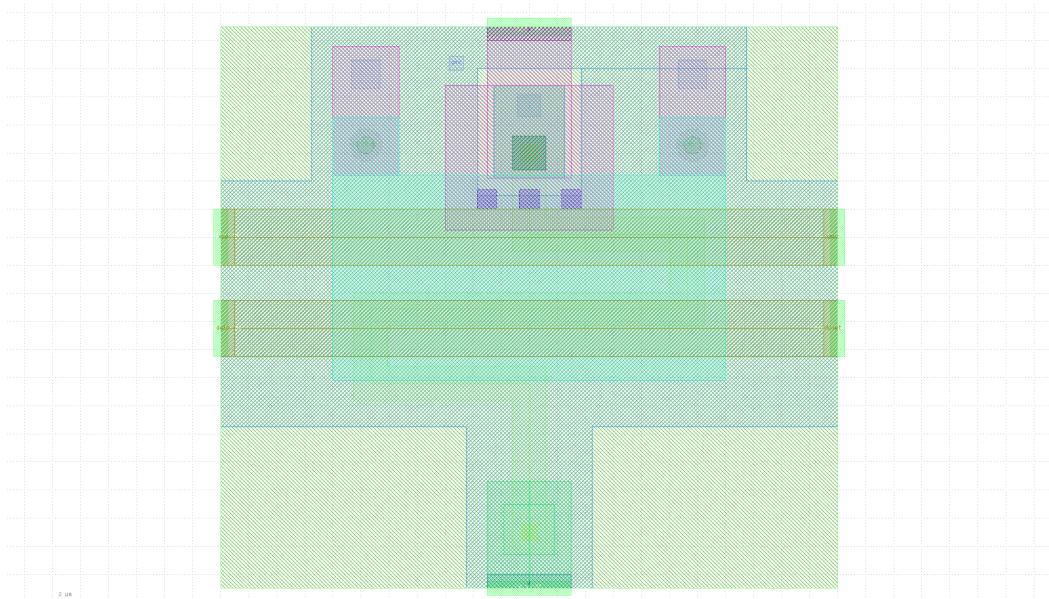


Figure 2.71: pre_bsquid layout.

Analog model

```

1 .SUBCKT pre_bsquid a xin dcin xout dcout q
2   Kid Lin Ld Kid
3   Kix Lin Lx Kix
4   .model jj(jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p))
5   .param Lin=0.90p
6   .param Lx=6.37p
7   .param Ld=6.37p

```

```

8 .param L1=1.64p
9 .param L2=L1
10 .param Lq=11.92p
11 .param B1=1.0
12 .param B2=B1
13 .param Kxd=0.2529
14 .param Kx1=-0.1950
15 .param Kx2=Kx1
16 .param Kd1=-0.1300
17 .param Kd2=Kd1
18 .param Kix=4.423E-3
19 .param Kid=4.041E-3
20
21 Kd2 Ld L2 Kd2
22 Kx2 Lx L2 Kx2
23 Kd1 Ld L1 Kd1
24 Kx1 Lx L1 Kx1
25 Kxd Lx Ld Kxd
26 Lin a 2 Lin
27 B2 3 0 B2 jjmit area=B2
28 B1 1 0 B1 jjmit area=B1
29 Lq 2 q Lq
30 L2 3 2 L2
31 L1 2 1 L1
32 Ld dcin dcout Ld
33 Lx xin xout Lx
34 .ends pre_bsquid

```

Listing 2.29: pre_bsquid netlist (.cir).

2.4.2 BFR-SQUID

Symbol

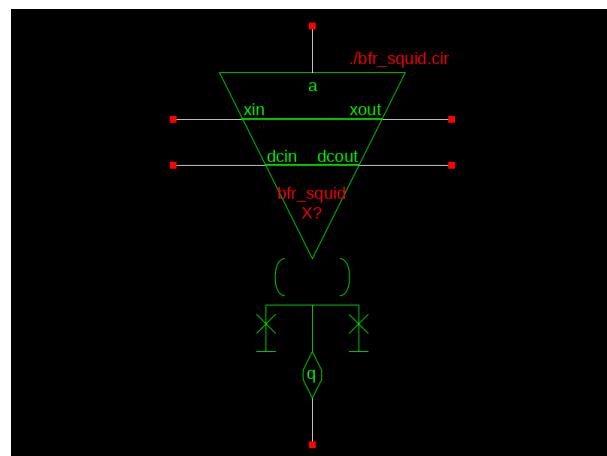
**Figure 2.72:** bfr_squid symbol.

Table 2.31: stack pin list.

Pin	Description
A	data input
XIN	serial clock input
DCIN	dc offset input
BIN	bias input for SQUID
XOUT	serial clock output
DCOUT	dc offset output
BOUT	bias output for SQUID

Schematic**Figure 2.73:** bfr_squid schematic.

Layout

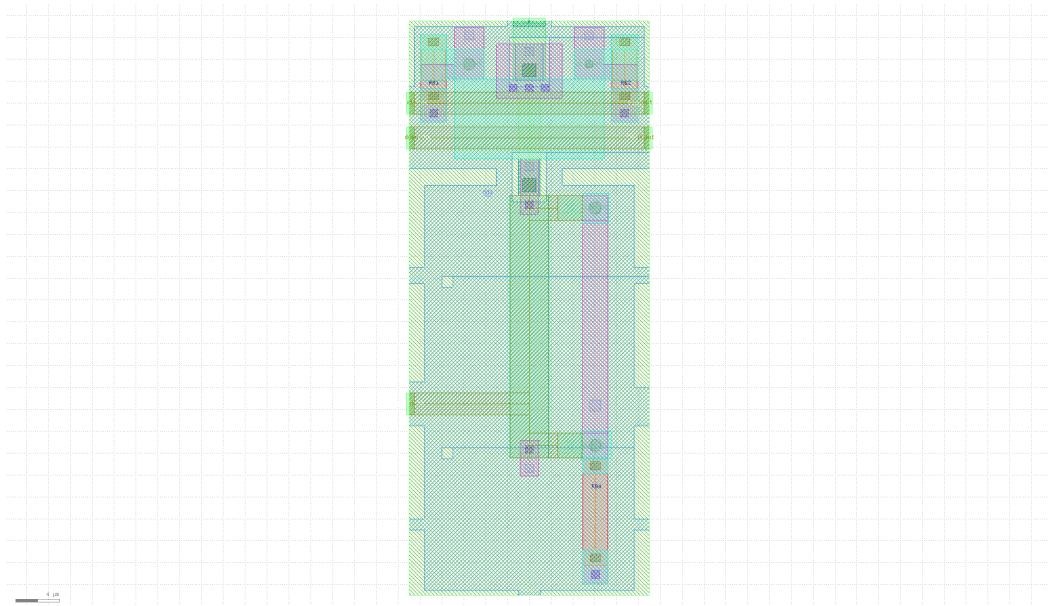


Figure 2.74: bfr_squid layout.

Analog model

```

1 .SUBCKT bfr_squid a xin dcin xout dcout q
2   Kdin Ld Lin Kdin
3   Kxin Lx Lin Kxin
4   Kq4 Lq L4 Kq4
5   Kq3 Lq L3 Kq3
6   Kd4 Ld L4 Kd4
7   Kd3 Ld L3 Kd3
8   Kx4 Lx L4 Kx4
9   Kx3 Lx L3 Kx3
10  KXRB2 Lx LRB2 KXRB2
11  L3 9 8 L3
12  L4 8 10 L4
13  LRB4 12 0 LRB4
14  RB4 10 12 RB4
15  LB4 11 0 LB4
16  B4 10 11 B4 jjmit area=B4
17  B3 9 0 B3 jjmit area=B3
18  Lb 8 q Lb
19  LRB2 7 0 LRB2
20  RB2 3 7 RB2
21  LRB1 6 0 LRB1
22  RB1 1 6 RB1
23  LB2 5 0 LB2
24  LB1 4 0 LB1
25  Lq 0 2 Lq
26  .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
27  .param Lin=1.06p
28  .param Lx=6.11p
29  .param Ld=6.24p
30  .param L1=1.64p
31  .param L2=L1
32  .param Lq=9.56p
33  .param Lb=3.24p
34  .param L3=8.32p
35  .param L4=2.62p

```

```

36
37 .param RB1=3.23
38 .param LRB1=0.680p
39 .param LB1=0.283p
40
41 .param RB2=3.23
42 .param LRB2=0.680p
43 .param LB2=0.283p
44
45 .param RB4=6.36
46 .param LB4=0.538p
47 .param LRB4=0.318p
48
49 .param B1=1.0
50 .param B2=0.5
51 .param B3=1.1
52 .param B4=1.1
53
54 .param Kxd=0.2636
55 .param Kxin=-6.556E-4
56 .param Kx1=-0.1975
57 .param Kx2=Kx1
58 .param Kdin=-9.370E-5
59 .param Kd1=-0.1350
60 .param Kd2=Kd1
61 .param KXRB1=-0.0382
62 .param KXRB2=0.0382
63 .param Kx3=-2.489E-3
64 .param Kx4=1.106E-3
65 .param Kd3=-5.206E-3
66 .param Kd4=3.619E-4
67 .param Kq3=-0.2956
68 .param Kq4=-0.0989
69 KXRB1 Lx LRB1 KXRB1
70 Kd2 Ld L2 Kd2
71 Kx2 Lx L2 Kx2
72 Kd1 Ld L1 Kd1
73 Kx1 Lx L1 Kx1
74 Kxd Lx Ld Kxd
75 Lin a 2 Lin
76 B2 3 5 B2 jjmit area=B2
77 B1 1 4 B1 jjmit area=B1
78 L2 3 2 L2
79 L1 2 1 L1
80 Ld dcin dcout Ld
81 Lx xin xout Lx
82 .ends bfr_squid

```

Listing 2.30: bfr_squid netlist (.cir).

2.4.3 QDC

Symbol

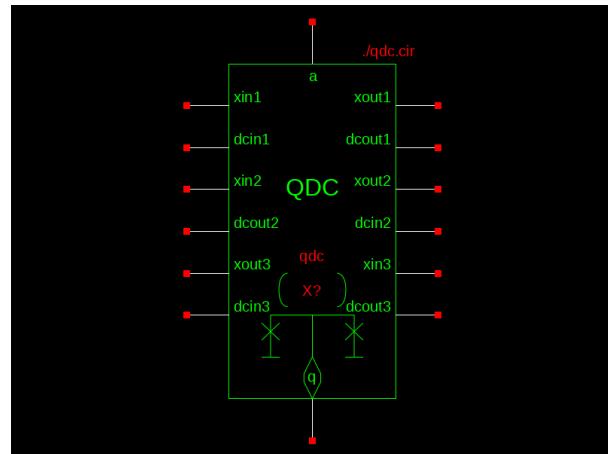


Figure 2.75: qdc symbol.

Table 2.32: qdc pin list.

Pin	Description
A	data input
XIN1	serial AC clock input (inv)
DCIN1	dc offset input (inv)
XIN2	serial AC clock input (pre_bsquid)
DCIN2	dc offset input (pre_bsquid)
XIN3	serial AC clock input (stack)
DCIN3	dc offset input (stack)
XOUT1	serial AC clock output (inv)
DCOUT1	dc offset output (inv)
XOUT2	serial AC clock output (pre_bsquid)
DCOUT2	dc offset output (pre_bsquid)
XOUT3	serial AC clock output (stack)
DCOUT3	dc offset output (stack)
Q	data output (observed through bias input of SQUID)

Schematic

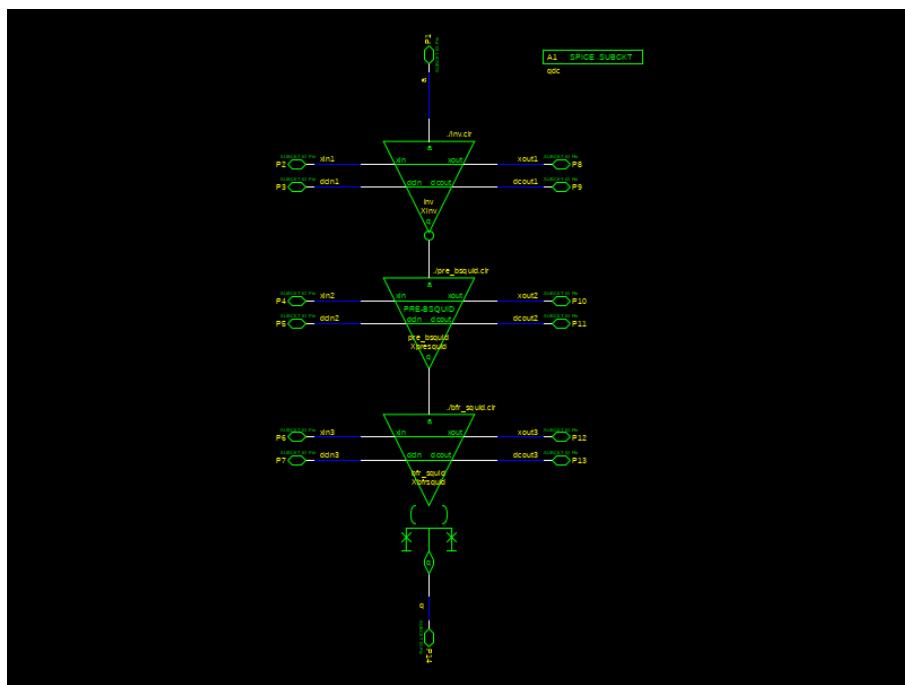


Figure 2.76: qdc schematic.

Layout

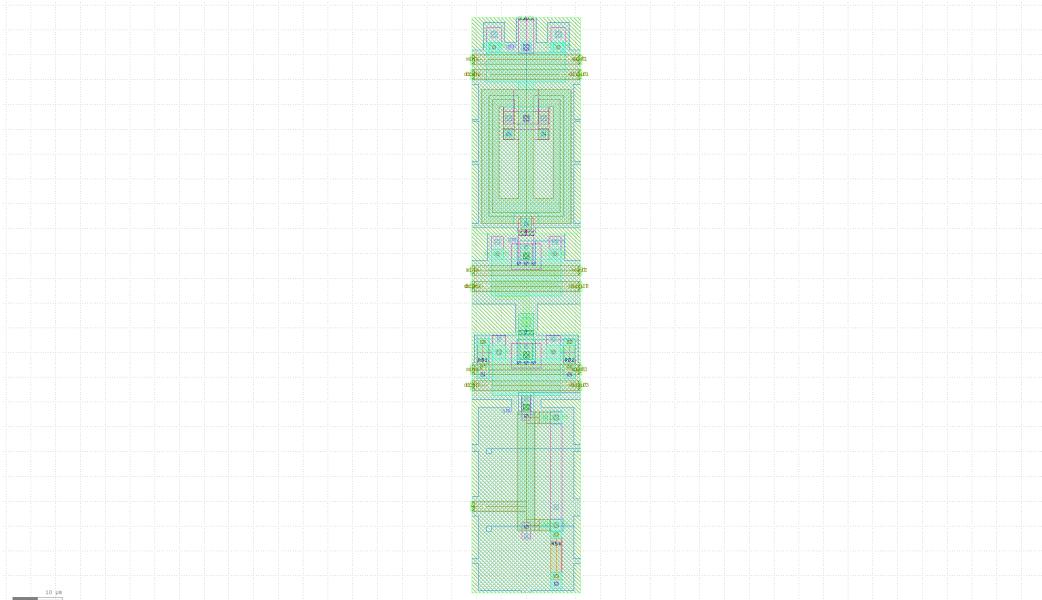


Figure 2.77: qdc layout.

Analog model

```

1 .SUBCKT inv a xin dcin xout dcout q
2 Kid Lin Ld Kid
3 Kiout Lin Lout Kiout
4 Kix Lin Lx Kix
5 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
6 .param Lin=1.56p
7 .param Lx=6.54p
8 .param Ld=6.57p
9 .param L1=1.56p
10 .param L2=L1
11 .param Lq=7.14p
12 .param Lout=30.24p
13 .param B1=0.5
14 .param B2=B1
15 .param Kxd=0.2912
16 .param Kx1=-0.2460
17 .param Kx2=Kx1
18 .param Kd1=-0.1909
19 .param Kd2=Kd1
20 .param Kout=0.5917
21 .param Kxout=1.335E-5
22 .param Kdout=2.443E-5
23 .param Kiout=1.172E-3
24 .param Kix=-2.712E-4
25 .param Kid=2.605E-4
26
27 Kxout Lx Lout Kxout
28 Kdout Ld Lout Kdout
29 Kout Lout Lq Kout
30 Kd2 Ld L2 Kd2
31 Kx2 Lx L2 Kx2
32 Kd1 Ld L1 Kd1
33 Kx1 Lx L1 Kx1
34 Kxd Lx Ld Kxd
35 Lin a 2 Lin
36 B2 3 0 B2 jjmit area=B2
37 B1 1 0 B1 jjmit area=B1
38 Lout 0 q Lout
39 Lq 2 0 Lq
40 L2 3 2 L2
41 L1 2 1 L1
42 Ld dcin dcout Ld
43 Lx xin xout Lx
44 .ends inv
45 .SUBCKT pre_bsquid a xin dcin xout dcout q
46 Kid Lin Ld Kid
47 Kix Lin Lx Kix
48 .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
49 .param Lin=0.90p
50 .param Lx=6.37p
51 .param Ld=6.37p
52 .param L1=1.64p
53 .param L2=L1
54 .param Lq=11.92p
55 .param B1=1.0
56 .param B2=B1
57 .param Kxd=0.2529
58 .param Kx1=-0.1950
59 .param Kx2=Kx1
60 .param Kd1=-0.1300
61 .param Kd2=Kd1
62 .param Kix=4.423E-3
63 .param Kid=4.041E-3
64
65 Kd2 Ld L2 Kd2
66 Kx2 Lx L2 Kx2
67 Kd1 Ld L1 Kd1
68 Kx1 Lx L1 Kx1
69 Kxd Lx Ld Kxd
70 Lin a 2 Lin

```

```

71 | B2 3 0 B2 jjmit area=B2
72 | B1 1 0 B1 jjmit area=B1
73 | Lq 2 q Lq
74 | L2 3 2 L2
75 | L1 2 1 L1
76 | Ld dcin dcout Ld
77 | Lx xin xout Lx
78 | .ends pre_bsquid
79 | .SUBCKT bfr_squid a xin dcin xout dcout q
80 | Kdin Ld Lin Kdin
81 | Kxin Lx Lin Kxin
82 | Kq4 Lq L4 Kq4
83 | Kq3 Lq L3 Kq3
84 | Kd4 Ld L4 Kd4
85 | Kd3 Ld L3 Kd3
86 | Kx4 Lx L4 Kx4
87 | Kx3 Lx L3 Kx3
88 | KXRB2 Lx LRB2 KXRB2
89 | L3 9 8 L3
90 | L4 8 10 L4
91 | LRB4 12 0 LRB4
92 | RB4 10 12 RB4
93 | LB4 11 0 LB4
94 | B4 10 11 B4 jjmit area=B4
95 | B3 9 0 B3 jjmit area=B3
96 | Lb 8 q Lb
97 | LRB2 7 0 LRB2
98 | RB2 3 7 RB2
99 | LRB1 6 0 LRB1
100 | RB1 1 6 RB1
101 | LB2 5 0 LB2
102 | LB1 4 0 LB1
103 | Lq 0 2 Lq
104 | .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
105 | .param Lin=1.06p
106 | .param Lx=6.11p
107 | .param Ld=6.24p
108 | .param L1=1.64p
109 | .param L2=L1
110 | .param Lq=9.56p
111 | .param Lb=3.24p
112 | .param L3=8.32p
113 | .param L4=2.62p
114 |
115 | .param RB1=3.23
116 | .param LRB1=0.680p
117 | .param LB1=0.283p
118 |
119 | .param RB2=3.23
120 | .param LRB2=0.680p
121 | .param LB2=0.283p
122 |
123 | .param RB4=6.36
124 | .param LB4=0.538p
125 | .param LRB4=0.318p
126 |
127 | .param B1=1.0
128 | .param B2=0.5
129 | .param B3=1.1
130 | .param B4=1.1
131 |
132 | .param Kxd=0.2636
133 | .param Kxin=-6.556E-4
134 | .param Kx1=-0.1975
135 | .param Kx2=Kx1
136 | .param Kdin=-9.370E-5
137 | .param Kd1=-0.1350
138 | .param Kd2=Kd1
139 | .param KXRB1=-0.0382
140 | .param KXRB2=0.0382

```

```

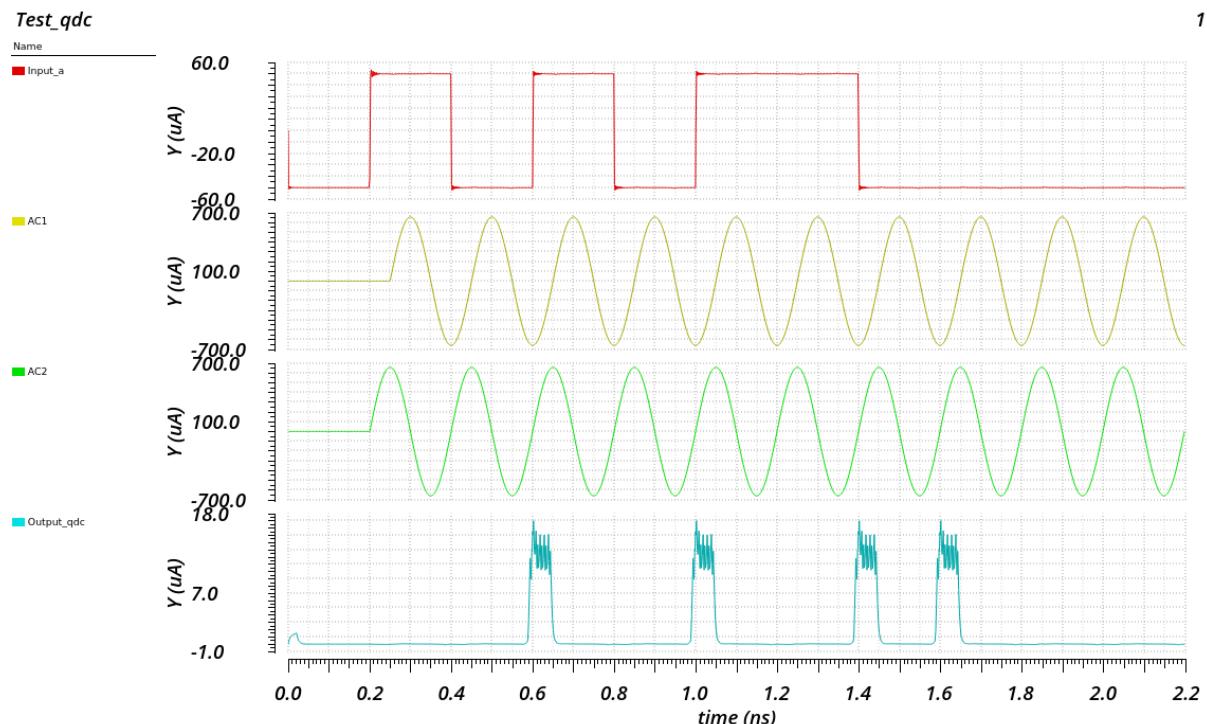
141 .param Kx3=-2.489E-3
142 .param Kx4=1.106E-3
143 .param Kd3=-5.206E-3
144 .param Kd4=3.619E-4
145 .param Kq3=-0.2956
146 .param Kq4=-0.0989
147 KXRB1 Lx LRB1 KXRB1
148 Kd2 Ld L2 Kd2
149 Kx2 Lx L2 Kx2
150 Kd1 Ld L1 Kd1
151 Kx1 Lx L1 Kx1
152 Kxd Lx Ld Kxd
153 Lin a 2 Lin
154 B2 3 5 B2 jjmit area=B2
155 B1 1 4 B1 jjmit area=B1
156 L2 3 2 L2
157 L1 2 1 L1
158 Ld dcin dcout Ld
159 Lx xin xout Lx
160 .ends bfr_squid
161 .SUBCKT qdc a xin1 dcin1 xin2 dcin2 xin3 dcin3 xout1 dcout1 xout2 dcout2 xout3 dcout3 q
162 Xbfrsquid 2 xin3 dcin3 xout3 dcout3 q bfr_squid
163 Xpresquid 1 xin2 dcin2 xout2 dcout2 2 pre_bsquid
164 Xinv a xin1 dcin1 xout1 dcout1 1 inv
165 .ends qdc

```

Listing 2.31: qdc netlist (.cir).

Simulation result

An input of 0 1 0 1 0 1 1 0 is sent to the input of the qdc which corresponds to the output shown on the waveform. Signals from top to bottom: AC source 1 (AC1) (generates phase 1 and 3), AC source 2 (AC2) (generates phase 2 and 4), Input at the buffer before the qdc and the output of qdc (measured across a resistor connected to qdc's output). Input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $664 \mu\text{A}$, and DC is set to $862 \mu\text{A}$.

**Figure 2.78:** qdc analog waveform.

2.5 On-chip Interfaces

2.5.1 AQFP2RSFQ

`aqfp2rsfq` is an interface that allows AQFP logic to transfer its current-based data to SFQ logic which uses SFQ-encoded data. It is based on an AQFP `bfr` cell and a magnetically coupled dc/SFQ converter. Both components are synchronized to the AC excitation current such when it rises, the AQFP buffer sets its output while an SFQ clock is generated through a Josephson junction switching event caused by magnetic flux applied from the AC excitation current via magnetic coupling. The output of AQFP buffer is magnetically coupled to the input branch of the Josephson comparator in the dc/SFQ converter. In the previous design, the polarity of this coupling was negative so the circuit effectively behaves like an inverter since the negative coupling is more natural in physical layout. In this version (ver2.1), we improved the physical layout design and were able to make the coupling positive, which means the circuit is now non-inverting. The comparator will produce an appropriate output when the SFQ clock arrives at the comparator, and the circuit behaves like a buffer. When the AQFP has a logic ‘1’ state during excitation, the dc/SFQ converter will produce an output SFQ pulse. Otherwise, no SFQ pulse is generated.

Symbol

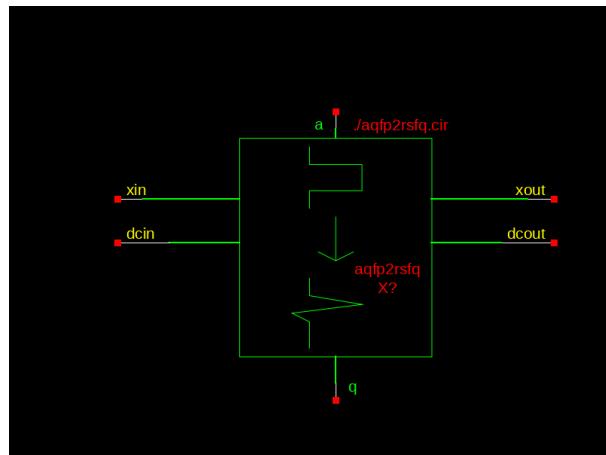


Figure 2.79: `aqfp2rsfq` symbol.

Table 2.33: `aqfp2rsfq` pin list.

Pin	Description
A	data input (AQFP)
XIN	serial clock input
DCIN	dc offset input
Q	data output (RSFQ)
XOUT	serial clock output
DCOUT	dc offset output

Schematic

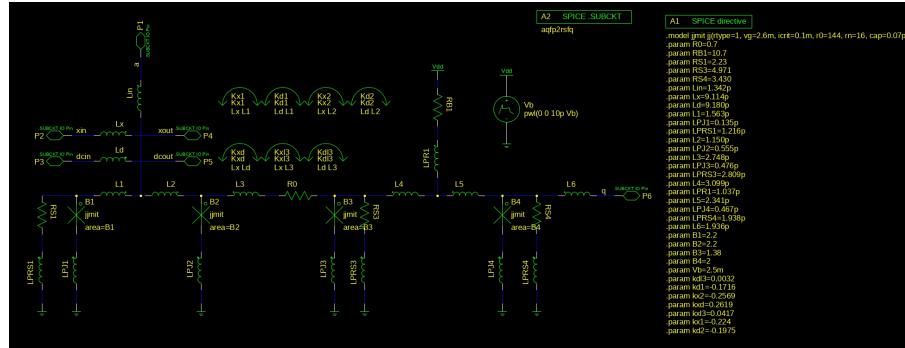


Figure 2.80: aqfp2rsfq schematic.

Layout

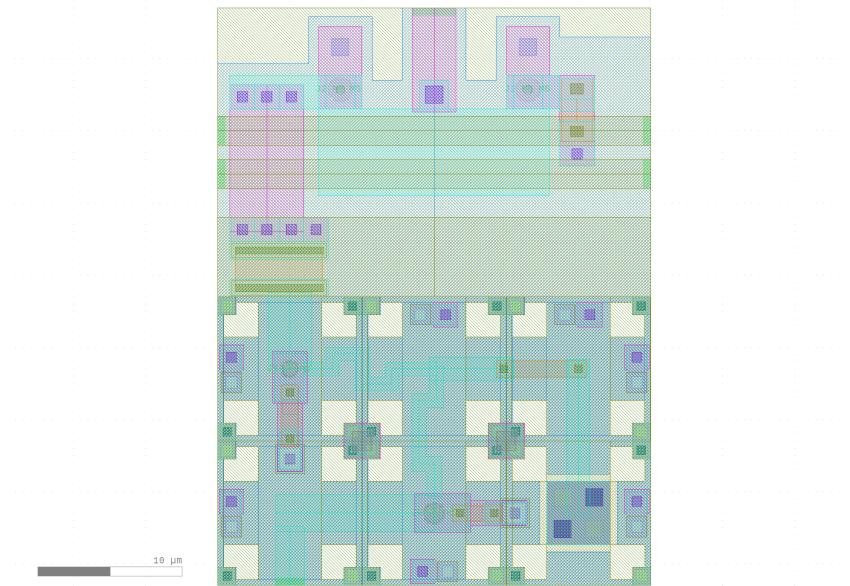


Figure 2.81: aqfp2rsfq layout.

Analog model

```

1 | .SUBCKT aqfp2rsfq a xin dcin xout dcout q
2 | Vb Vdd 0 pwl(0 0 10p Vb)
3 | RS1 1 15 RS1
4 | LPRS1 15 0 LPRS1
5 | LPJ1 2 0 LPJ1
6 | LPRS4 13 0 LPRS4
7 | LPRS3 14 0 LPRS3
8 | RS3 7 14 RS3
9 | RS4 10 13 RS4
10|.model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
11|.param R0=0.7
12|.param RB1=10.7
13|.param RS1=2.23

```

```

14 .param RS3=4.971
15 .param RS4=3.430
16 .param Lin=1.342p
17 .param Lx=9.114p
18 .param Ld=9.180p
19 .param L1=1.563p
20 .param LPJ1=0.135p
21 .param LPRS1=1.216p
22 .param L2=1.150p
23 .param LPJ2=0.555p
24 .param L3=2.748p
25 .param LPJ3=0.476p
26 .param LPRS3=2.809p
27 .param L4=3.099p
28 .param LPR1=1.037p
29 .param L5=2.341p
30 .param LPJ4=0.467p
31 .param LPRS4=1.938p
32 .param L6=1.936p
33 .param B1=2.2
34 .param B2=2.2
35 .param B3=1.38
36 .param B4=2
37 .param Vb=2.5m
38 .param kd13=0.0032
39 .param kd1=-0.1716
40 .param kx2=-0.2569
41 .param kxd=0.2619
42 .param kx13=0.0417
43 .param kx1=-0.224
44 .param kd2=-0.1975
45 B4 10 12 B4 jjmit area=B4
46 Kd13 Ld L3 Kd13
47 Kx13 Lx L3 Kx13
48 Kxd Lx Ld Kxd
49 Kd2 Ld L2 Kd2
50 Kx2 Lx L2 Kx2
51 Kd1 Ld L1 Kd1
52 Kx1 Lx L1 Kx1
53 RB1 Vdd 11 RB1
54 Lin a 5 Lin
55 Ld dcout dcin Ld
56 Lx xout xin Lx
57 LPJ4 12 0 LPJ4
58 LPR1 11 9 LPR1
59 L6 10 q L6
60 L5 9 10 L5
61 L4 7 9 L4
62 LPJ3 8 0 LPJ3
63 B3 7 8 B3 jjmit area=B3
64 R0 6 7 R0
65 L3 3 6 L3
66 LPJ2 4 0 LPJ2
67 L2 3 5 L2
68 L1 5 1 L1
69 B2 3 4 B2 jjmit area=B2
70 B1 1 2 B1 jjmit area=B1
71 .ends aqfp2rsfq

```

Listing 2.32: aqfp2rsfq netlist (.cir).

Simulation result

Simulation waveform of the aqfp2rsfq. The data input peak-to-peak amplitude is $\pm 5 \mu\text{A}$, AC amplitude is $800 \mu\text{A}$, and DC is set to 1.2 mA . The dc/SFQ component uses a bias voltage of 2.5 mV . The AQFP data pattern is 10010110 which resulted in the SFQ comparator producing a same data pattern of 10010110 and the circuit operates as a buffer.

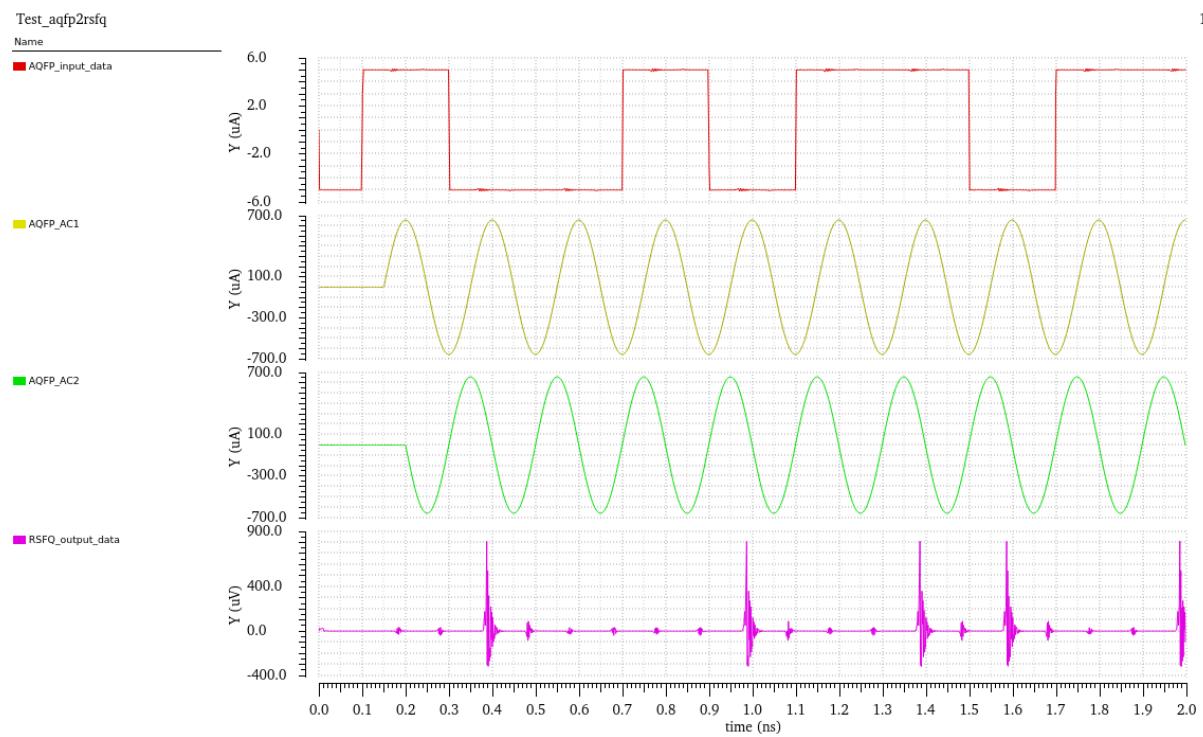


Figure 2.82: aqfp2rsfq analog waveform.

Switching energy

To be investigated in the future.

2.5.2 RSFQ2AQFP

`rsfq2aqfp` is an interface that allows RSFQ logic to transfer its SFQ-based data to AQFP logic which operates on bidirectional current-encoded data. It is based on an RSFQ DFF whose storage loop is magnetically coupled to an output inductor. An inductor with about $-400 \mu\text{A}$ offset current is coupled to the output inductor to shift the current levels induced into the output inductor from unipolar to bipolar. This bipolar current is then suitable as data input for AQFP. When the DFF is in logic state '1', a positive output current is produced. Otherwise, the output current will remain negative. An SFQ clock is used to reset the `rsfq2aqfp` interface back to state '0'.

Symbol

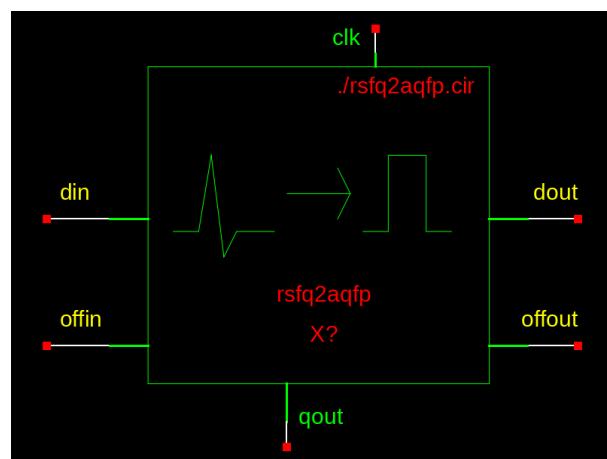


Figure 2.83: `rsfq2aqfp` symbol.

Table 2.34: `rsfq2aqfp` pin list.

Pin	Description
A	data input (RSFQ)
CLK	clock (RSFQ)
OFF	dc offset to shift the level of Q
Q	data output (AQFP)
SQ	data output (SFQ) to eject SFQ pulse

Schematic

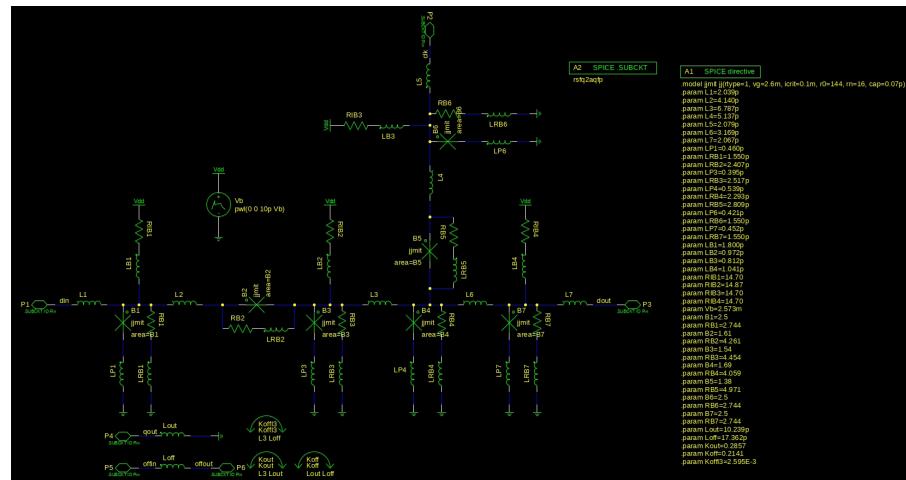


Figure 2.84: rsfq2aqfp schematic.

Layout

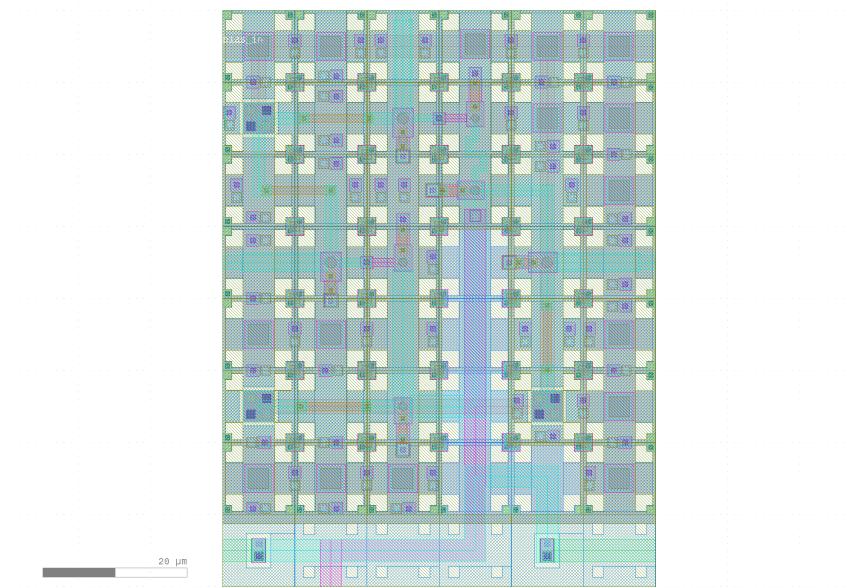


Figure 2.85: rsfq2aqfp layout.

Analog model

```

1 | .SUBCKT rsfq2aqfp din clk dout qout offin offout
2 | Vb Vdd 0 pwl(0 0 10p Vb)
3 | RIB3 Vdd 22 RIB3
4 | RIB4 Vdd 16 RIB4
5 | RIB2 Vdd 9 RIB2
6 | RIB1 Vdd 4 RIB1
7 | .model jjmit jj(rttype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
8 | .param L1=2.039p

```

```

9  .param L2=4.140p
10 .param L3=6.787p
11 .param L4=5.137p
12 .param L5=2.079p
13 .param L6=3.169p
14 .param L7=2.067p
15 .param LP1=0.460p
16 .param LRB1=1.550p
17 .param LRB2=2.407p
18 .param LP3=0.395p
19 .param LRB3=2.517p
20 .param LP4=0.539p
21 .param LRB4=2.293p
22 .param LRB5=2.809p
23 .param LP6=0.421p
24 .param LRB6=1.550p
25 .param LP7=0.452p
26 .param LRB7=1.550p
27 .param LB1=1.800p
28 .param LB2=0.972p
29 .param LB3=0.812p
30 .param LB4=1.041p
31 .param RIB1=14.70
32 .param RIB2=14.87
33 .param RIB3=14.70
34 .param RIB4=14.70
35 .param Vb=2.573m
36 .param B1=2.5
37 .param RB1=2.744
38 .param B2=1.61
39 .param RB2=4.261
40 .param B3=1.54
41 .param RB3=4.454
42 .param B4=1.69
43 .param RB4=4.059
44 .param B5=1.38
45 .param RB5=4.971
46 .param B6=2.5
47 .param RB6=2.744
48 .param B7=2.5
49 .param RB7=2.744
50 .param Lout=10.239p
51 .param Loff=17.362p
52 .param Kout=0.2857
53 .param Koff=0.2141
54 .param Koffl3=2.595E-3
55 Koffl3 L3 Loff Koffl3
56 Koff Lout Loff Koff
57 Kout L3 Lout Kout
58 Loff offin offout Loff
59 Lout qout 0 Lout
60 L5 20 clk L5
61 LRB6 23 0 LRB6
62 RB6 20 23 RB6
63 LB3 22 20 LB3
64 LP6 21 0 LP6
65 B6 20 21 B6 jjmit area=B6
66 L4 20 18 L4
67 LRB5 19 11 LRB5
68 RB5 18 19 RB5
69 B5 18 11 B5 jjmit area=B5
70 L7 14 dout L7
71 LRB7 17 0 LRB7
72 RB7 14 17 RB7
73 LB4 16 14 LB4
74 LP7 15 0 LP7
75 B7 14 15 B7 jjmit area=B7
76 L6 11 14 L6
77 LRB4 13 0 LRB4
78 RB4 11 13 RB4

```

```

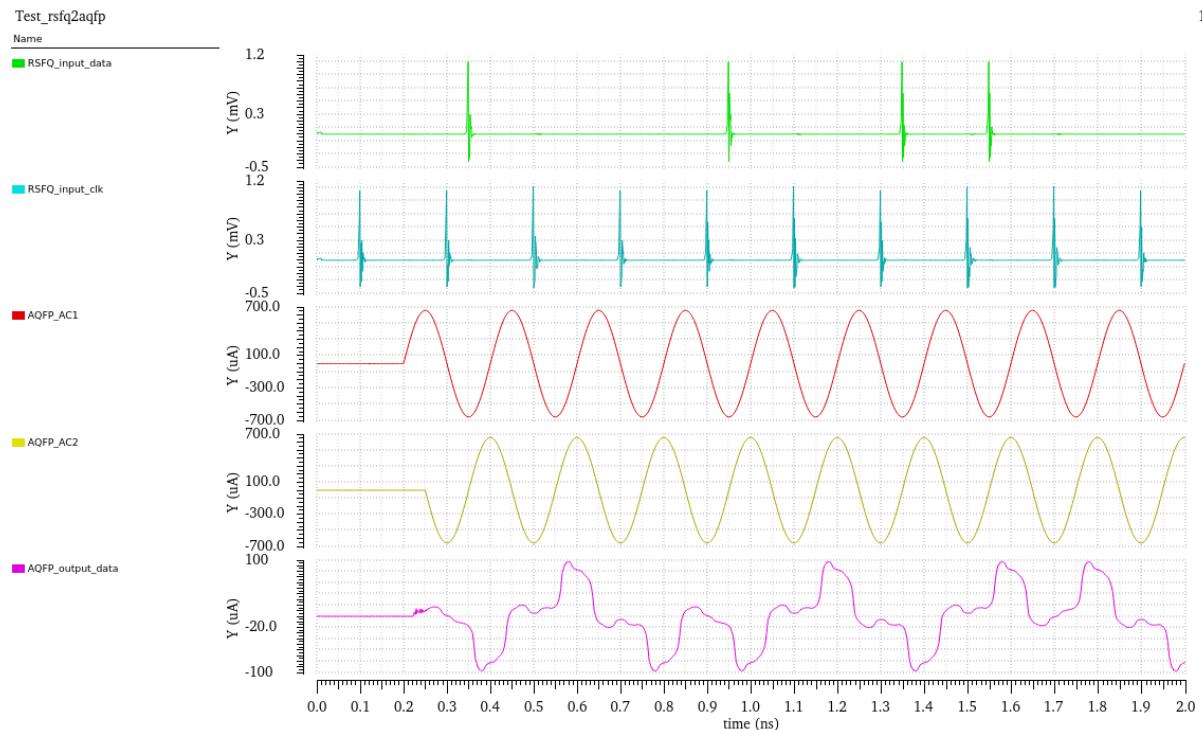
79 | LP4 12 0 LP4
80 | B4 11 12 B4 jjmit area=B4
81 | L3 5 11 L3
82 | LRB3 10 0 LRB3
83 | RB3 5 10 RB3
84 | LB2 9 5 LB2
85 | LP3 8 0 LP3
86 | B3 5 8 B3 jjmit area=B3
87 | L1 din 1 L1
88 | LRB2 6 5 LRB2
89 | LRB1 7 0 LRB1
90 | RB1 1 7 RB1
91 | RB2 3 6 RB2
92 | B2 3 5 B2 jjmit area=B2
93 | LB1 4 1 LB1
94 | L2 1 3 L2
95 | LP1 2 0 LP1
96 | B1 1 2 B1 jjmit area=B1
97 | .ends rsfq2aqfp

```

Listing 2.33: rsfq2aqfp netlist (.cir).

Simulation result

Simulation waveform of the rsfq2aqfp. The data input is provided through JTLs connected to the RSFQ-based DFF with an input pattern of 01001011. An SFQ clock pulse is provided at the same frequency as the 5 GHz AQFP AC excitation clock. The AC amplitude is 800 μ A, and DC is set to 1.2 mA. The bipolar AQFP output current matches the same data as the SFQ-based encoding of 01001011.

**Figure 2.86:** rsfq2aqfp analog waveform.

Switching energy

To be investigated in the future.

2.6 Sub-Cells

2.6.1 Constant

Constant cells contain constant 0 (const0) and constant 1 (const1) that are building blocks to develop AQFP logic AND and OR cells. Both const0 and const1 are designed by changing the symmetric architecture of a SQUID into asymmetric to generate the constant positive current (as logic ‘1’) or negative current (as logic ‘0’). Here we only introduce the constant 0.

Symbol

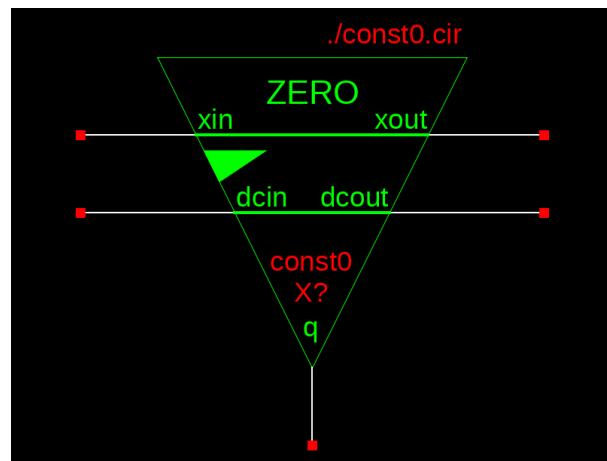


Figure 2.87: const0 symbol.

Table 2.35: const0 pin list.

Pin	Description
XIN	serial clock input
DCIN	dc offset input
Q	data output
XOUT	serial clock output
DCOUT	dc offset output

Schematic

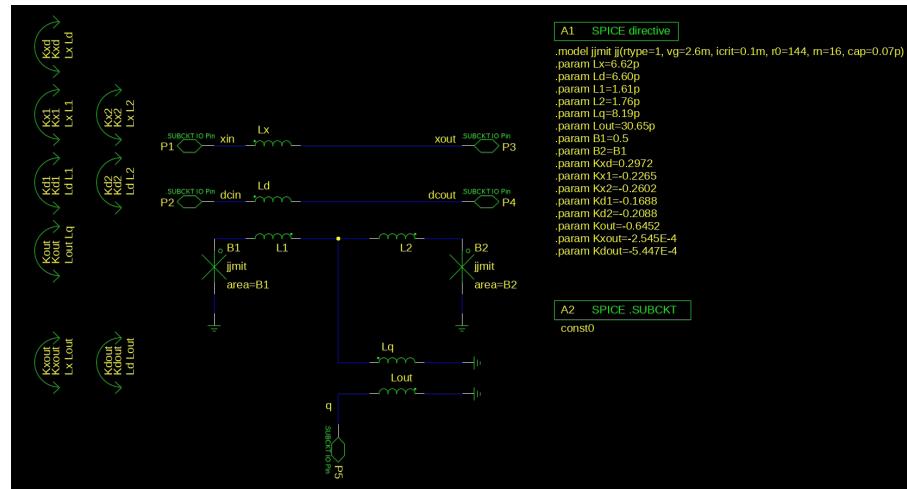


Figure 2.88: const0 schematic.

Layout

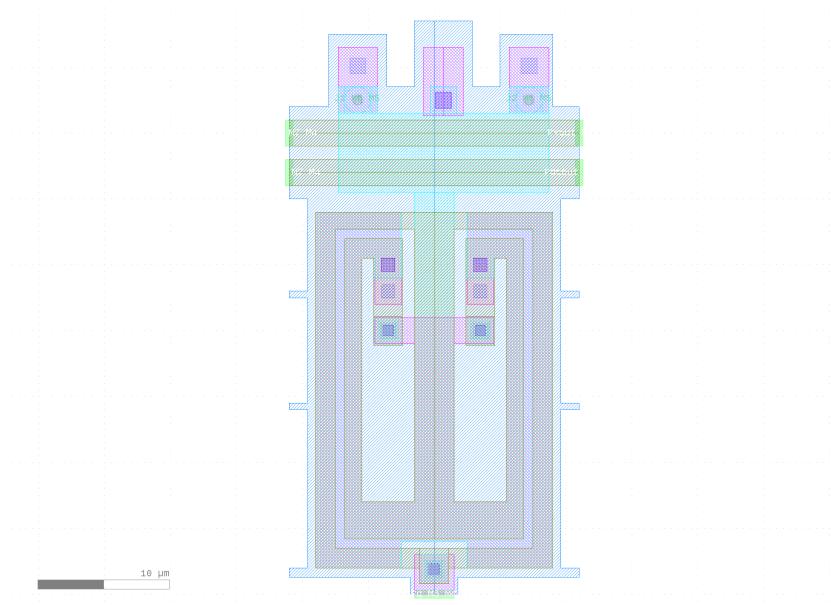


Figure 2.89: const0 layout.

Analog model

```

1 | .SUBCKT const0 xin dcin xout dcout q
2 | .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
3 | .param Lx=6.62p
4 | .param Ld=6.60p
5 | .param L1=1.61p
6 | .param L2=1.76p
7 | .param Lq=8.19p
8 | .param Lout=30.65p

```

```

9  .param B1=0.5
10 .param B2=B1
11 .param Kxd=0.2972
12 .param Kx1=-0.2265
13 .param Kx2=-0.2602
14 .param Kd1=-0.1688
15 .param Kd2=-0.2088
16 .param Kout=-0.6452
17 .param Kxout=-2.545E-4
18 .param Kdout=-5.447E-4
19 Kxout Lx Lout Kxout
20 Kdout Ld Lout Kdout
21 Kout Lout Lq Kout
22 Kd2 Ld L2 Kd2
23 Kx2 Lx L2 Kx2
24 Kd1 Ld L1 Kd1
25 Kx1 Lx L1 Kx1
26 Kxd Lx Ld Kxd
27 B2 3 0 B2 jjmit area=B2
28 B1 1 0 B1 jjmit area=B1
29 Lout 0 q Lout
30 Lq 2 0 Lq
31 L2 3 2 L2
32 L1 2 1 L1
33 Ld dcin dcout Ld
34 Lx xin xout Lx
35 .ends const0

```

Listing 2.34: const0 netlist (.cir).

2.6.2 Branch

A 1-to-n (or n-to-1) branch, which consists of (n+1) superconducting inductors, is used to split or merge AQFP current. It is a basic building block to make AQFP logic cells such as AND, OR, Majority and Splitter introduced in section II. Here we only introduce 1-to-2 branch, others can be found as separate files in the deliverables.

Symbol

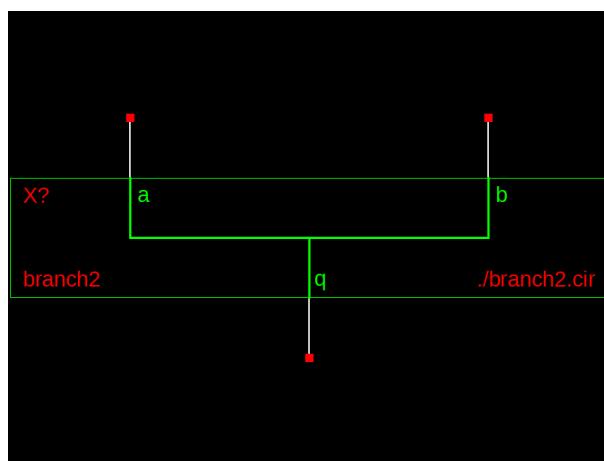
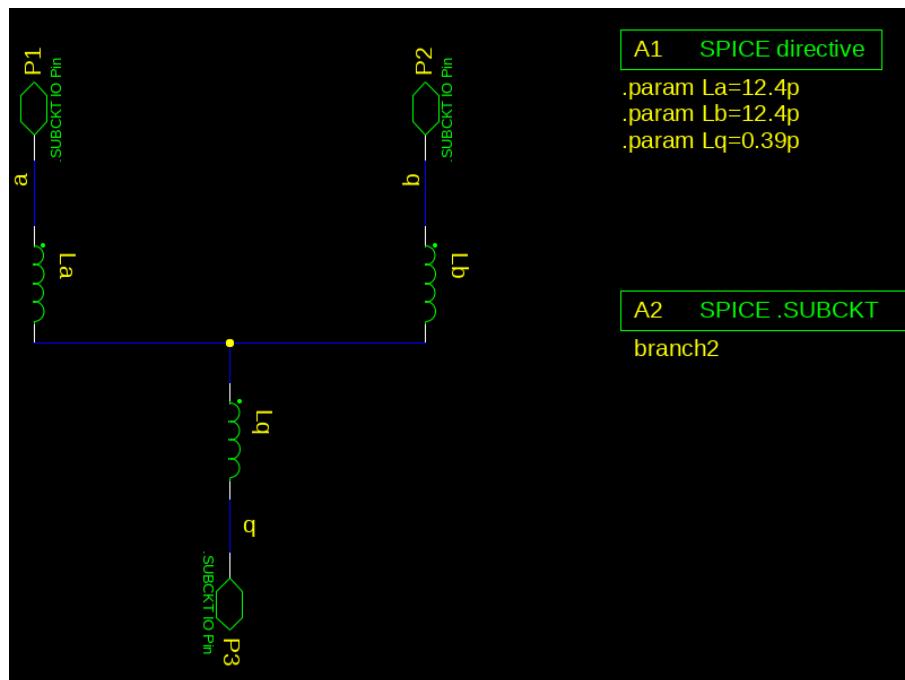
**Figure 2.90:** branch2 symbol.

Table 2.36: branch2 pin list.

Pin	Description
A	data inout
B	data inout
C	data inout

Schematic**Figure 2.91:** branch2 schematic.

Layout

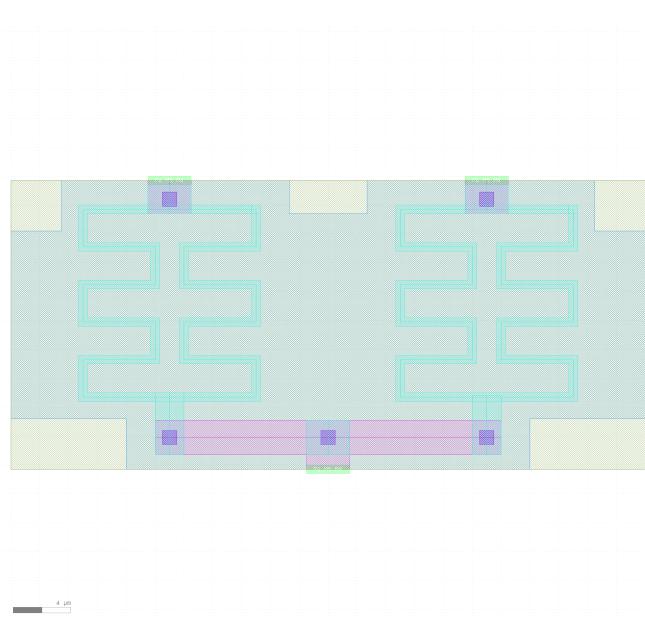


Figure 2.92: branch2 layout.

Analog model

```

1 | ****
2 | * Begin .SUBCKT model *
3 | * spice-sdb ver 4.28.2007 *
4 | ****
5 | .SUBCKT branch2 a b q
6 | ===== Begin SPICE netlist of main design =====
7 | .param La=12.4p
8 | .param Lb=12.4p
9 | .param Lq=0.39p
10| Lq 1 q Lq
11| Lb b 1 Lb
12| La a 1 La
13| .ends branch2
14| ****

```

Listing 2.35: branch2 netlist (.cir).

2.6.3 Storage gate

An AQFP storage gate is a building block to develop AQFP sequential logic cells such as latch (`qfp1`) and non-destructive read-out (`ndro_qfp1`).

Symbol

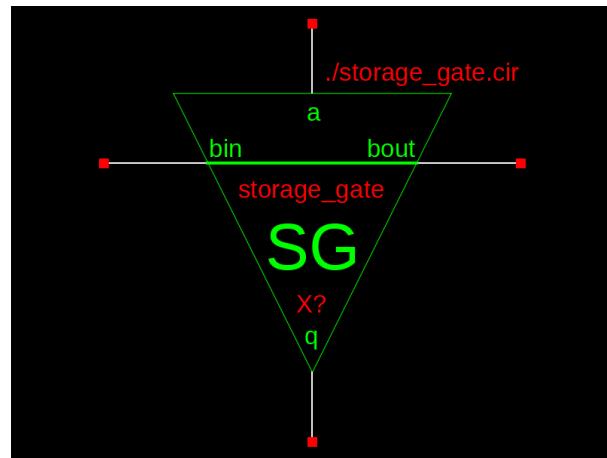


Figure 2.93: storage_gate symbol.

Table 2.37: storage_gate pin list.

Pin	Description
A	data inout
B	data inout
C	data inout

Schematic

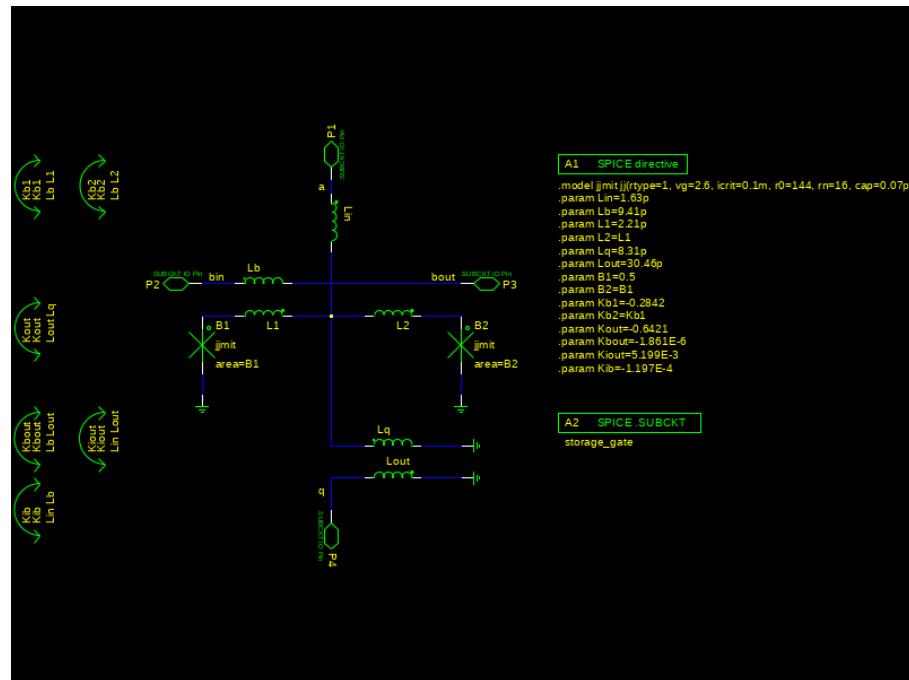


Figure 2.94: storage_gate schematic.

Layout

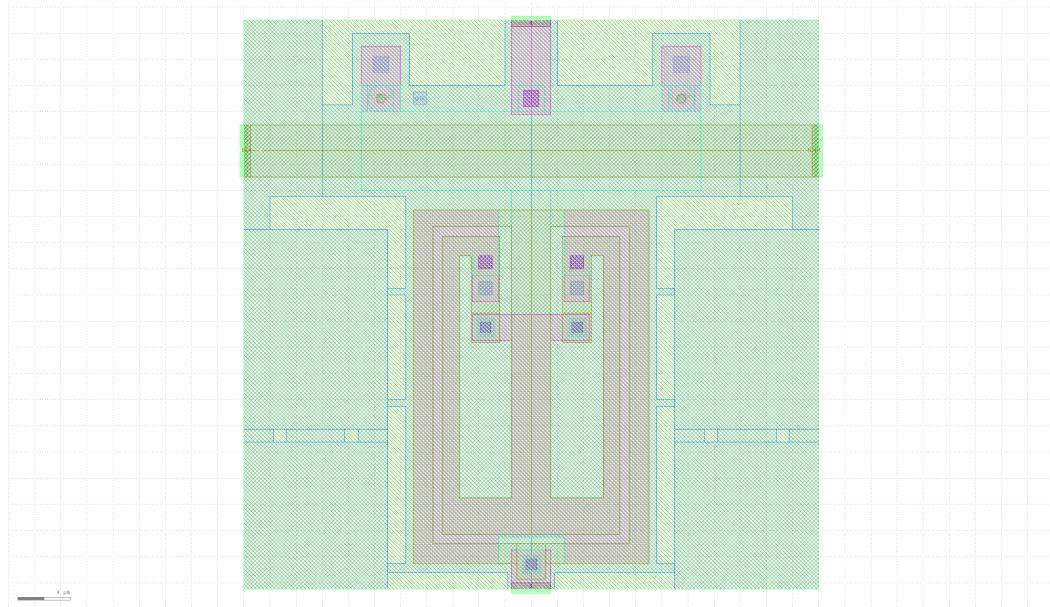


Figure 2.95: storage_gate layout.

Analog model

```

1  .SUBCKT storage_gate a bin bout q
2  Kiout Lin Lout Kiout
3  Kbout Lb Lout Kbout
4  .model jjmit jj(rtype=1, vg=2.6m, icrit=0.1m, r0=144, rn=16, cap=0.07p)
5  .param Lin=1.63p
6  .param Lb=9.41p
7  .param L1=2.21p
8  .param L2=L1
9  .param Lq=8.31p
10 .param Lout=30.46p
11 .param B1=0.5
12 .param B2=B1
13 .param Kb1=-0.2842
14 .param Kb2=Kb1
15 .param Kout=-0.6421
16 .param Kbout=-1.663E-5
17 .param Kiout=5.199E-3
18 .param Kib=-4.608E-5
19
20 Kib Lin Lb Kib
21 Kout Lout Lq Kout
22 Kb2 Lb L2 Kb2
23 Kb1 Lb L1 Kb1
24 Lin a 2 Lin
25 B2 3 0 B2 jjmit area=B2
26 B1 1 0 B1 jjmit area=B1
27 Lout 0 q Lout
28 Lq 2 0 Lq
29 L2 3 2 L2
30 L1 2 1 L1
31 Lb bin bout Lb
32 .ends storage_gate

```

Listing 2.36: storage_gate netlist (.cir).

2.6.4 HDL ac/dc interface

Verilog module ‘biasDir_b.v’ is an internal module for each AQFP logic gate for handling the excitation ac clock (AC/xout) and dc bias (dcin/dcout). Because some cells such as ‘bfr’ can be placed in a normal or flipped orientation, it was necessary to develop a bi-directional interface to handle the ac/dc bias. For gates such as AND or OR which can only be placed in a single direction relative to the bias direction, we enforce a uni-directional version of this interface. This interface also determines the relative directions of ac and dc bias applied to the cell. If their directions are the same, the ac clock (modeled as a digital square wave) will be applied to the logic cell as is. If their directions are different, the gate will receive an inverted version of the ac clock. This allows us to model 4-phase clocking at the HDL level using two ac clocks (ac1 with a 0 degree shift, and ac2 with a -90 degree shift) and a dc bias (modeled as a 0 to 1 transition at the beginning of simulation).

Digital model

```

1 //-----
2 // Design Name : biasDir
3 // File Name   : biasDir.v
4 // Function    : determines direction of excitation
5 //               bias and creates normalized gate
6 //               excitation signal
7 // Developer   : Christopher Ayala & Olivia Chen
8 //                  (olivia.chen@ieee.org)
9 //-----
10 'timescale 1ps/10fs
11
12 module biasDir_b(xin, xout, dcin, dcout, gatex);
13
14   parameter propagationDelay = 0.198;
15
16   inout xin, xout;
17   inout dcin, dcout;
18   output gatex;
19   reg xiFrst, xoFrst, dciFrst, dcoFrst, xiReg, xoReg, dciReg, dcoReg;
20   reg fndDirx, fndDirdc, gatex, sameDir, clock;
21
22   initial begin
23     xiFrst = 1'b0;
24     xoFrst = 1'b0;
25     dciFrst = 1'b0;
26     dcoFrst = 1'b0;
27     xiReg = 1'b0;
28     xoReg = 1'b0;
29     dciReg = 1'b0;
30     dcoReg = 1'b0;
31     fndDirx = 1'b0;
32     fndDirdc = 1'b0;
33     gatex = 1'b0;
34     sameDir = 1'b0;
35     clock = 1'b0;
36   end
37
38   always@(AC or xout or dcin or dcout) begin
39     if((AC==1'b1 || xin==1'b0) && !fndDirx) begin
40       xiFrst = 1'b1;
41       fndDirx = 1'b1;
42     end
43
44     if((xout==1'b1 || xout==1'b0) && !fndDirx) begin
45       xoFrst = 1'b1;

```

```

46     fndDirx = 1'b1;
47   end
48
49   if((dcin==1'b1) && !fndDirdc) begin
50     dciFrst = 1'b1;
51     fndDirdc = 1'b1;
52   end
53
54   if((dcout==1'b1) && !fndDirdc) begin
55     dcoFrst = 1'b1;
56     fndDirdc = 1'b1;
57   end
58
59   xiReg <= #propagationDelay xin;
60   xoReg <= #propagationDelay xout;
61   dciReg <= dcin;
62   dcoReg <= dcout;
63   clock <= xiFrst ? xin : xout;
64   sameDir <= dciFrst ^~ xiFrst;
65   gatex <= sameDir? clock : !clock;
66 end
67
68 assign xin = xoFrst ? xoReg : 1'bz;
69 assign xout = xiFrst ? xiReg : 1'bz;
70 assign dcin = dcoFrst ? dcoReg : 1'bz;
71 assign dcout = dciFrst ? dciReg : 1'bz;
72
73 endmodule

```

Listing 2.37: HDL ac/dc interface Verilog model code.

2.6.5 HDL dc interface

Verilog module ‘biasDC.v’ is an internal module for special cells such as the `storage_gate` which only have a dc bias line. This module does not apply any special normalization but it is necessary for resolving the bi-directional ports of the dc bias in HDL gate-level modeling.

Digital model

```

1 //-----
2 // Design Name : biasDC
3 // File Name   : biasDC.v
4 // Function    : determines direction of DC bias and
5 //                resolves direction of bi-directional ports
6 // Developer   : Christopher Ayala (chris.ayala@ieee.org)
7 //-----
8
9 `timescale 1ps/10fs
10
11 module biasDC(a, b);
12
13 parameter propagationDelay = 0.033;
14 parameter length = 1;
15
16 inout a, b;
17
18 reg aFrst, bFrst, aReg, bReg, fndDir;
19
20 initial begin
21   aFrst = 1'b0;
22   bFrst = 1'b0;
23   aReg  = 1'b0;
24   bReg  = 1'b0;
25   fndDir = 1'b0;
26 end

```

```

27
28 always@(a or b) begin
29   if((a==1'b1 || a==1'b0) && !fnDir) begin
30     aFrst = 1'b1;
31     fnDir = 1'b1;
32   end
33
34   if((b==1'b1 || b==1'b0) && !fnDir) begin
35     bFrst = 1'b1;
36     fnDir = 1'b1;
37   end
38
39   aReg <= #(propagationDelay*length) a;
40   bReg <= #(propagationDelay*length) b;
41 end
42
43 assign a = bFrst ? bReg : 1'bz;
44 assign b = aFrst ? aReg : 1'bz;
45
46 endmodule

```

Listing 2.38: HDL dc interface Verilog model code.

2.7 Standard Delay Format (SDF)

Included in the cell library under the `verilog` sub-directory, we include a global Standard Delay Format (SDF) file which includes timing parameters extracted from AQFPTX [7]. This SDF file is for the standard 5 GHz clock rate under nominal bias levels. The listing is shown in Listing 2.39. In Phase 3, we will implement a timing characterization methodology that also covers higher-order effects.

```

1  (
2    (DELAYFILE
3      (SDFVERSION "4.0")
4      (DESIGN "top")
5      (DATE "Wednesday June 22 GMT 2021")
6      (VENDOR "")
7      (PROGRAM "AQFPTX")
8      (VERSION "1.3")
9      (DIVIDER /)
10     (PROCESS "mit")
11     (TEMPERATURE 1:2:4)
12     (TIMESCALE 1ps)
13     (CELL
14       (CELLTYPE "inv")
15       (INSTANCE *)
16       (DELAY
17         (ABSOLUTE
18           (COND xin
19             (IOPATH xin q (13.33))
20           )
21         )
22       )
23     (TIMINGCHECK
24       (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
25       (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
26       (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
27       (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
28     )
29   )
30 )
31
32 (CELL
33   (CELLTYPE "and2_nn")
34   (INSTANCE *)
35   (DELAY
36     (ABSOLUTE

```

```

37          (COND xin
38              (IOPATH xin q (13.33))
39          )
40      )
41  )
42 (TIMINGCHECK
43     (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
44     (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
45     (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
46     (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
47     (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
48     (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
49     (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
50     (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
51
52   )
53 )
54
55 (CELL
56   (CELLTYPE "and2_np")
57   (INSTANCE *)
58   (DELAY
59     (ABSOLUTE
60       (COND xin
61           (IOPATH xin q (13.33))
62       )
63     )
64   )
65 (TIMINGCHECK
66     (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
67     (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
68     (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
69     (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
70     (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
71     (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
72     (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
73     (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
74
75   )
76 )
77
78 (CELL
79   (CELLTYPE "and2_pn")
80   (INSTANCE *)
81   (DELAY
82     (ABSOLUTE
83       (COND xin
84           (IOPATH xin q (13.33))
85       )
86     )
87   )
88 (TIMINGCHECK
89     (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
90     (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
91     (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
92     (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
93     (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
94     (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
95     (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
96     (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
97
98   )
99 )
100
101 (CELL
102   (CELLTYPE "and2_pp")
103   (INSTANCE *)
104   (DELAY
105     (ABSOLUTE
106       (COND xin

```

```

107           (IOPATH xin q (13.33))
108       )
109   )
110 )
111 (TIMINGCHECK
112   (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
113   (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
114   (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
115   (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
116   (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
117   (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
118   (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
119   (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
120 )
121 )
122 )
123 )
124 (CELL
125   (CELLTYPE "and3_ppp")
126   (INSTANCE *)
127   (DELAY
128     (ABSOLUTE
129       (COND xin
130         (IOPATH xin q (13.33))
131       )
132     )
133   )
134 (TIMINGCHECK
135   (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
136   (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
137   (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
138   (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
139   (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
140   (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
141   (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
142   (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
143   (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
144   (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
145   (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
146   (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
147 )
148 )
149 )
150 (CELL
151   (CELLTYPE "bfr")
152   (INSTANCE *)
153   (DELAY
154     (ABSOLUTE
155       (COND xin
156         (IOPATH xin q (13.33))
157       )
158     )
159   )
160 (TIMINGCHECK
161   (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
162   (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
163   (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
164   (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
165 )
166 )
167 )
168 )
169 (CELL
170   (CELLTYPE "const0")
171   (INSTANCE *)
172   (DELAY
173     (ABSOLUTE
174       (COND xin
175         (IOPATH xin q (13.33))
176       )

```

```

177      )
178    )
179  )
180
181 (CELL
182   (CELLTYPE "const1")
183   (INSTANCE *)
184   (DELAY
185     (ABSOLUTE
186       (COND xin
187         (IOPATH xin q (13.33))
188       )
189     )
190   )
191 )
192
193 (CELL
194   (CELLTYPE "maj3_nnn")
195   (INSTANCE *)
196   (DELAY
197     (ABSOLUTE
198       (COND xin
199         (IOPATH xin q (13.33))
200       )
201     )
202   )
203 (TIMINGCHECK
204   (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
205   (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
206   (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
207   (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
208   (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
209   (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
210   (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
211   (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
212   (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
213   (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
214   (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
215   (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
216 )
217 )
218
219 (CELL
220   (CELLTYPE "maj3_nnp")
221   (INSTANCE *)
222   (DELAY
223     (ABSOLUTE
224       (COND xin
225         (IOPATH xin q (13.33))
226       )
227     )
228   )
229 (TIMINGCHECK
230   (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
231   (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
232   (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
233   (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
234   (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
235   (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
236   (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
237   (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
238   (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
239   (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
240   (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
241   (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
242 )
243 )
244
245 (CELL
246   (CELLTYPE "maj3_npn")

```

```

247     (INSTANCE *)
248     (DELAY
249         (ABSOLUTE
250             (COND xin
251                 (IOPATH xin q (13.33))
252             )
253         )
254     )
255     (TIMINGCHECK
256         (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
257         (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
258         (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
259         (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
260         (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
261         (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
262         (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
263         (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
264         (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
265         (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
266         (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
267         (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
268     )
269   )
270
271   (CELL
272     (CELLTYPE "maj3_npp")
273     (INSTANCE *)
274     (DELAY
275         (ABSOLUTE
276             (COND xin
277                 (IOPATH xin q (13.33))
278             )
279         )
280     )
281     (TIMINGCHECK
282         (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
283         (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
284         (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
285         (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
286         (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
287         (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
288         (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
289         (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
290         (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
291         (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
292         (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
293         (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
294     )
295   )
296
297
298   (CELL
299     (CELLTYPE "maj3_pnn")
300     (INSTANCE *)
301     (DELAY
302         (ABSOLUTE
303             (COND xin
304                 (IOPATH xin q (13.33))
305             )
306         )
307     )
308     (TIMINGCHECK
309         (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
310         (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
311         (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
312         (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
313         (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
314         (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
315         (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
316         (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))

```

```

317      (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
318      (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
319      (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
320      (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
321    )
322  )
323
324  (CELL
325    (CELLTYPE "maj3_pnp")
326    (INSTANCE *)
327    (DELAY
328      (ABSOLUTE
329        (COND xin
330          (IOPATH xin q (13.33)))
331        )
332      )
333    )
334  (TIMINGCHECK
335    (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
336    (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
337    (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
338    (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
339    (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
340    (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
341    (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
342    (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
343    (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
344    (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
345    (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
346    (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
347  )
348  )
349
350  (CELL
351    (CELLTYPE "maj3_ppn")
352    (INSTANCE *)
353    (DELAY
354      (ABSOLUTE
355        (COND xin
356          (IOPATH xin q (13.33)))
357        )
358      )
359    )
360  (TIMINGCHECK
361    (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
362    (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
363    (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
364    (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
365    (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
366    (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
367    (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
368    (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
369    (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
370    (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
371    (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
372    (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
373  )
374  )
375
376  (CELL
377    (CELLTYPE "maj3_ppp")
378    (INSTANCE *)
379    (DELAY
380      (ABSOLUTE
381        (COND xin
382          (IOPATH xin q (13.33)))
383        )
384      )
385    )
386  (TIMINGCHECK

```

```

387      (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
388      (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
389      (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
390      (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
391      (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
392      (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
393      (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
394      (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
395      (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
396      (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
397      (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
398      (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
399    )
400  )
401
402 (CELL
403   (CELLTYPE "maj5_ppppp")
404   (INSTANCE *)
405   (DELAY
406     (ABSOLUTE
407       (COND xin
408         (IOPATH xin q (13.33))
409       )
410     )
411   )
412   (TIMINGCHECK
413     (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
414     (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
415     (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
416     (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
417     (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
418     (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
419     (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
420     (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
421     (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
422     (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
423     (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
424     (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
425     (SETUP (COND d (posedge d)) (posedge gatex) (-8.10))
426     (SETUP (COND not_d (negedge d)) (posedge gatex) (-8.05))
427     (HOLD (posedge gatex) (COND d (negedge d)) (10.90))
428     (HOLD (posedge gatex) (COND not_d (posedge d)) (10.95))
429     (SETUP (COND e (posedge e)) (posedge gatex) (-8.10))
430     (SETUP (COND not_e (negedge e)) (posedge gatex) (-8.05))
431     (HOLD (posedge gatex) (COND e (negedge e)) (10.90))
432     (HOLD (posedge gatex) (COND not_e (posedge e)) (10.95))
433   )
434 )
435
436 (CELL
437   (CELLTYPE "qfp1")
438   (INSTANCE *)
439   (DELAY
440     (ABSOLUTE
441       (COND xin
442         (IOPATH xin q (13.33))
443       )
444     )
445   )
446   (TIMINGCHECK
447     (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
448     (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
449     (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
450     (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
451     (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
452     (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
453     (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
454     (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
455   )
456 )

```

```

457
458     (CELL
459         (CELLTYPE "or2_nn")
460         (INSTANCE *)
461         (DELAY
462             (ABSOLUTE
463                 (COND xin
464                     (IOPATH xin q (13.33))
465                 )
466             )
467         )
468     (TIMINGCHECK
469         (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
470         (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
471         (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
472         (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
473         (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
474         (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
475         (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
476         (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
477     )
478   )
479 )
480
481 (CELL
482     (CELLTYPE "or2_np")
483     (INSTANCE *)
484     (DELAY
485         (ABSOLUTE
486             (COND xin
487                 (IOPATH xin q (13.33))
488             )
489         )
490     )
491     (TIMINGCHECK
492         (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
493         (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
494         (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
495         (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
496         (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
497         (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
498         (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
499         (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
500     )
501   )
502 )
503
504 (CELL
505     (CELLTYPE "or2_pn")
506     (INSTANCE *)
507     (DELAY
508         (ABSOLUTE
509             (COND xin
510                 (IOPATH xin q (13.33))
511             )
512         )
513     )
514     (TIMINGCHECK
515         (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
516         (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
517         (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
518         (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
519         (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
520         (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
521         (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
522         (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
523     )
524   )
525 )
526

```

```

527   (CELL
528     (CELLTYPE "or2_pp")
529     (INSTANCE *)
530     (DELAY
531       (ABSOLUTE
532         (COND xin
533           (IOPATH xin q (13.33))
534         )
535       )
536     )
537     (TIMINGCHECK
538       (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
539       (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
540       (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
541       (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
542       (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
543       (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
544       (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
545       (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
546     )
547   )
548 )
549
550 (CELL
551   (CELLTYPE "or3_ppp")
552   (INSTANCE *)
553   (DELAY
554     (ABSOLUTE
555       (COND xin
556         (IOPATH xin q (13.33))
557       )
558     )
559   )
560   (TIMINGCHECK
561     (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
562     (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
563     (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
564     (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
565     (SETUP (COND b (posedge b)) (posedge gatex) (-8.10))
566     (SETUP (COND not_b (negedge b)) (posedge gatex) (-8.05))
567     (HOLD (posedge gatex) (COND b (negedge b)) (10.90))
568     (HOLD (posedge gatex) (COND not_b (posedge b)) (10.95))
569     (SETUP (COND c (posedge c)) (posedge gatex) (-8.10))
570     (SETUP (COND not_c (negedge c)) (posedge gatex) (-8.05))
571     (HOLD (posedge gatex) (COND c (negedge c)) (10.90))
572     (HOLD (posedge gatex) (COND not_c (posedge c)) (10.95))
573   )
574 )
575
576 (CELL
577   (CELLTYPE "spl2")
578   (INSTANCE *)
579   (DELAY
580     (ABSOLUTE
581       (COND xin
582         (IOPATH xin q0 (13.33))
583       )
584       (COND xin
585         (IOPATH xin q1 (13.33))
586       )
587     )
588   )
589   (TIMINGCHECK
590     (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
591     (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
592     (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
593     (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
594   )
595 )
596

```

```

597
598     (CELL
599         (CELLTYPE "spl3")
600         (INSTANCE *)
601         (DELAY
602             (ABSOLUTE
603                 (COND xin
604                     (IOPATH xin q0 (13.33))
605                 )
606                 (COND xin
607                     (IOPATH xin q1 (13.33))
608                 )
609                 (COND xin
610                     (IOPATH xin q2 (13.33))
611                 )
612             )
613         )
614     (TIMINGCHECK
615         (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
616         (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
617         (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
618         (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
619
620     )
621   )
622
623   (CELL
624       (CELLTYPE "spl4")
625       (INSTANCE *)
626       (DELAY
627           (ABSOLUTE
628               (COND xin
629                   (IOPATH xin q0 (13.33))
630               )
631               (COND xin
632                   (IOPATH xin q1 (13.33))
633               )
634               (COND xin
635                   (IOPATH xin q2 (13.33))
636               )
637
638               (COND xin
639                   (IOPATH xin q3 (13.33))
640               )
641           )
642       )
643     (TIMINGCHECK
644         (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
645         (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
646         (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
647         (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
648     )
649   )
650
651   (CELL
652       (CELLTYPE "boost1")
653       (INSTANCE *)
654       (DELAY
655           (ABSOLUTE
656               (COND xin
657                   (IOPATH xin q (13.33))
658               )
659           )
660       )
661     (TIMINGCHECK
662         (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
663         (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
664         (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
665         (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
666

```

```

667      )
668  )
669
670  (CELL
671    (CELLTYPE "boost2f2")
672    (INSTANCE *)
673    (DELAY
674      (ABSOLUTE
675        (COND xin
676          (IOPATH xin q0 (13.33))
677        )
678        (COND xin
679          (IOPATH xin q1 (13.33))
680        )
681      )
682    )
683    (TIMINGCHECK
684      (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
685      (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
686      (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
687      (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
688
689    )
690  )
691
692  (CELL
693    (CELLTYPE "boost2f4")
694    (INSTANCE *)
695    (DELAY
696      (ABSOLUTE
697        (COND xin
698          (IOPATH xin q0 (13.33))
699        )
700        (COND xin
701          (IOPATH xin q1 (13.33))
702        )
703        (COND xin
704          (IOPATH xin q2 (13.33))
705        )
706
707        (COND xin
708          (IOPATH xin q3 (13.33))
709        )
710      )
711    )
712    (TIMINGCHECK
713      (SETUP (COND a (posedge a)) (posedge gatex) (-8.10))
714      (SETUP (COND not_a (negedge a)) (posedge gatex) (-8.05))
715      (HOLD (posedge gatex) (COND a (negedge a)) (10.90))
716      (HOLD (posedge gatex) (COND not_a (posedge a)) (10.95))
717    )
718  )
719)

```

Listing 2.39: SDF file included in the library.

Part II

RSFQ Logic

3. Introduction and Setup

3.1 Introduction

This RSFQ cell library is developed under the IARPA SuperTools/ColdFlux contract via the U.S. Army Research Office grant W911NF-17-1-0120. The aim is to create a generic and open-source cell library with RSFQ logic [12] as part of the IARPA SuperTools Program [13], [14]. The cell library is continually updated and the latest version of the library can always be found at: <https://github.com/sunmagnetics/RSFQlib>.

The free and open-source tools *XIC* [15], *JoSIM* [6], [16], *JoSIM-tools* [17], *KLayout* and *TimEx* [18], [19] are used to develop and test the RSFQ cells. The circuit schematics are drawn using *XIC*. *JoSIM* is used as the SPICE engine for simulating the cells, while *JoSIM-tools* is used for operating margin analysis as well as cell parameter optimization. *KLayout* is used to construct the cell layouts. *TimEx* is used to extract the characteristics of the cell to generate the Mealy Finite State Machine diagram and Verilog files. Icarus *Verilog* [8] and *GTKWave* [9] can be used to simulate and view the verilog files for each cell. Additionally, *InductEx* [5], [20] is used for impedance extraction during cell layout design. A free version of *InductEx* is available, but has limited capacity.

Version 2.1 of the RSFQ cell library includes two versions of each cell: one with standard connections designed to be connected directly with other cells, and a version designed to be connected to Passive Transmission Lines (PTLs). The version of the cell designed to be connected to PTLs include integrated PTL transmitter and receiver cells. To indicate the integration of PTL transmitters and receivers within a cell, the letter ‘T’ is added at the end of a cell name, for example the DFF with integrated PTL transmitters and receivers will be referred to as DF^TF.

The following core cells are included in the RSFQ cell library:

- Interconnects: JTL, JTLT, SPLIT, SPLITT, MERGE, MERGET, PTLLTX, PTLLRX, Always0 (synchronous and asynchronous) and Always0T (synchronous and asynchronous).
- Logic cells: AND2, AND2T, OR2, OR2T, XOR, XORT, NOT, NOTT, XNOR, XNORT.
- Buffers: DFF, DF^TF, NDRO, NDROT, BUFF and BUFFT.
- Interfacing cells: DCSFQ, DCSFQ-PTLLTX, PTLLRX-SFQDC and SFQDC.

More complex functions can be constructed through connecting several core cells. The cells are currently optimized to run at a maximum clock frequency of 50 GHz. Each delivered cell is documented in 5 parts:

1. **Schematic:** The schematic of a cell is constructed using *Xic* and is delivered in the native *Xic* format.
2. **Layout:** The physical layout of the cells can be constructed using *Xic* or *KLayout* and is delivered in standard GDSII format.
3. **Analog model:**
 - (a) **Netlist:** The netlist presents the device-level construction of a circuit. Each cell is delivered with both a “base” and “optimized” circuit netlist files. The base netlist shows how the cell is designed from first principles using phase-based equations. The cells are then optimized before the physical layout is done. The optimized circuit netlists provides the back-annotated netlist extracted through *InductEx*.
 - (b) **Pin list:** The pin labels and function of each pin is listed.
 - (c) **Simulation results:** JoSIM is used for all circuit simulations. The simulation uses the cell testbench to verify cell functionality.
4. **Digital model:**
 - (a) **Verilog model:** The behavior-level model of a cell with timing specifications included within the model. The verilog models are delivered in two parts – a basic verilog model accompanied by a SDF file to include timing delays, and a self-contained verilog model which contains the timing delays of the cell within the verilog model itself. All verilog models are extracted using *TimEx* and is delivered in standard HDL Verilog format. The verilog model included within this document is the self-contained verilog model.
 - (b) **Simulation results:** The digital simulation testbench is generated through *TimEx* and is run using *Icarus Verilog* and wave viewer *GTKWave*. Each edge event indicates an SFQ pulse.
 - (c) **Mealy finite state machine diagram:** The state machine diagram is extracted using *TimEx* and is delivered in standard PDF format.
5. **Power consumption:** The power consumption of each cell is calculated in terms of static and dynamic power consumption. As a rough estimate, it is assumed that each junction switches with every clock signal. For asynchronous cells, the power consumption is calculated through assuming that an input pulse train is applied at the same frequency as the specified clock frequency. Following [21], dynamic power consumption can be calculated as $P_d = f\Phi_0I_c$ and static power consumption can be calculated as $P_s = I_bV_b$.

3.2 Setup

The latest version of the RSFQ cell library can be found at: <https://github.com/sunmagnetics/RSFQlib>. The RSFQ cell library is simulated and tested using several free and/or open-source tools:

- *Xic* is part of *XicTools* and can be found at <http://www.wrcad.com/xic.html>.
- *JoSIM* can be found at <https://github.com/JoeyDelp/JoSIM/>.
- *JoSIM-tools* can be found at <https://github.com/pleroux0/josim-tools>.
- *TimEx* can be found at <https://github.com/sunmagnetics/TimEx>.
- *KLayout* can be found at <https://www.klayout.de/>.
- *InductEx* can be found at <https://www.inductex.info>.
- *Icarus Verilog* can be found at <http://iverilog.icarus.com/>.
- *GTKWave* can be found at <http://gtkwave.sourceforge.net/>.

No additional setup is required to use the RSFQ cell library.

3.3 License

The generic RSFQ cell library is free to distribute and/or modify under the terms of the MIT license.

4. RSFQ Cell Library: Standard Connections

4.1 Interconnects

4.1.1 JTL

The JTL, Josephson transmission line, cell is commonly used to re-establish and propagate SFQ pulses. The cell is not designed to be directly connected to a PTL.

Schematic

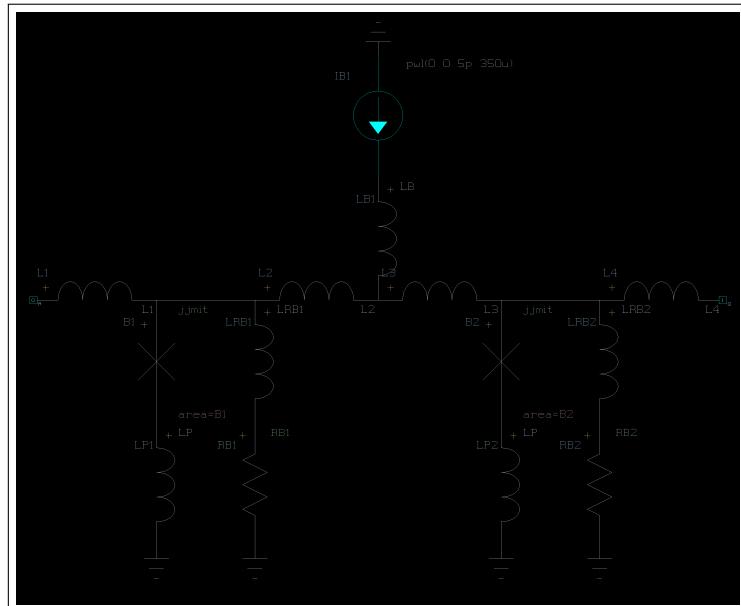


Figure 4.1: Schematic of RSFQ JTL.

Layout

The physical layout for the RSFQ JTL is shown in Fig. 4.2. The layout height is $70 \mu m$ and the width is $30 \mu m$. The cell includes an integrated bias line on M0 within the top row of track blocks. A bias pillar from M0 to M6 connects the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

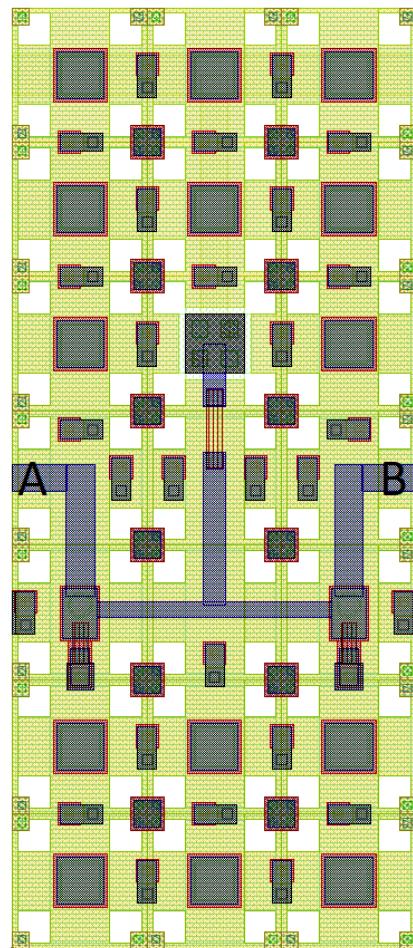


Figure 4.2: RSFQ JTL Layout

Table 4.1: RSFQ JTL pin list.

Pin	Description
a	Data input
q	Data output

Analog model

```

1 * Back-annotated simulation file written by InductEx v.6.0.4 on 10-3-21.
2 * Author: L. Schindler
3 * Version: 2.1
4 * Last modification date: 12 January 2021
5 * Last modification by: L. Schindler
6
7 *$Ports a q
8 .subckt LSMITLL_JTL a q
9 .model jjmit jj(rtype=1, vg=2.8mV, cap=0.07pF, r0=160, rn=16, icrit=0.1mA)
10 .param Phi0=2.067833848E-15
11 .param B0=1
12 .param Ic0=0.0001
13 .param IcRs=100u*6.859904418
14 .param B0Rs=IcRs/Ic0*B0
15 .param Rsheet=2
16 .param Lsheet=1.13e-12
17 .param LP=0.2p
18 .param IC=2.5
19 .param Lptl=2p
20 .param LB=2p
21 .param BiasCoef=0.7
22
23 .param B1=IC
24 .param B2=IC
25 .param IB1=(B1+B2)*Ic0*BiasCoef
26 .param LB1=LB
27 .param L1=Phi0/(4*B1*Ic0)
28 .param L2=Phi0/(4*B1*Ic0)
29 .param L3=Phi0/(4*B1*Ic0)
30 .param L4=Phi0/(4*B2*Ic0)
31 .param RB1=B0Rs/B1
32 .param RB2=B0Rs/B2
33 .param LRB1=(RB1/Rsheet)*Lsheet+LP
34 .param LRB2=(RB2/Rsheet)*Lsheet+LP
35 .param LP1=LP
36 .param LP2=LP
37
38 B1 1 2 jjmit area=B1
39 B2 6 7 jjmit area=B2
40 IB1 0 5 pwl(0 0 5p IB1)
41 L1 a 1 2.082E-12
42 L2 1 4 2.06E-12
43 L3 4 6 2.067E-12
44 L4 6 q 2.075E-12
45 LP1 2 0 4.998E-13
46 LP2 7 0 5.011E-13
47 LB1 5 4 LB1
48 RB1 1 3 RB1
49 RB2 6 8 RB2
50 LRB1 3 0 LRB1
51 LRB2 8 0 LRB2
52 .end

```

Listing 4.1: RSFQ JTL JoSIM netlist.

The simulation results for the RSFQ JTL using JoSIM is shown in Fig. 4.3. The testbench is included within the cell library for user verification. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the phase over the output JJ of pin **q**, and
- (d) the phase over the input JJ of the load circuit.

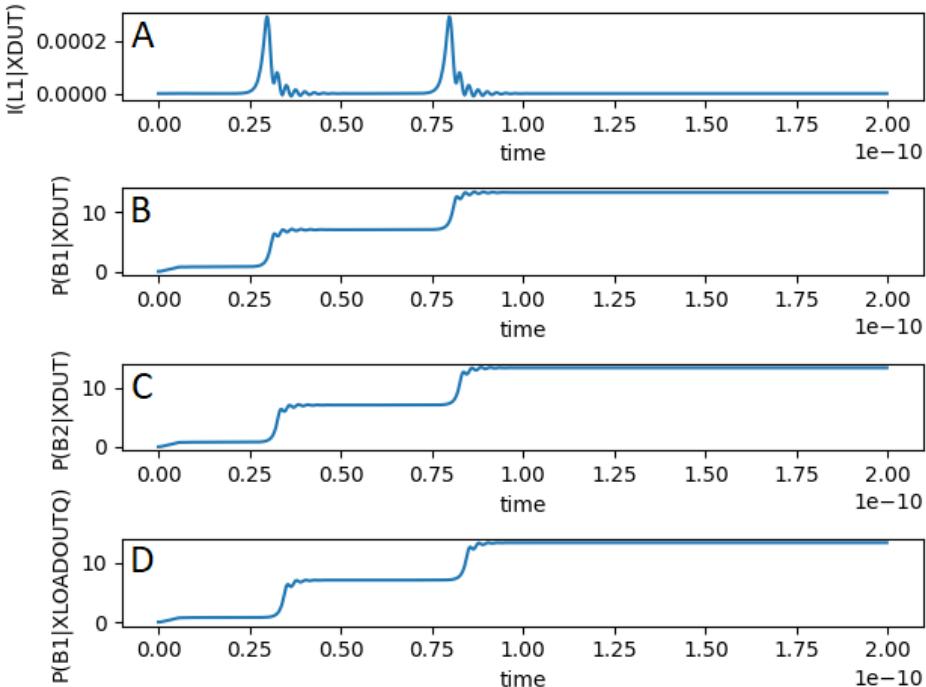


Figure 4.3: RSFQ JTL analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 12 January 2021
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmitll_JTL (a, q);
9
10 input
11   a;
12
13 output
14   q;
15
16 reg
17   q;
18
19 real
20   delay_state0_a_q = 3.5,
21   ct_state0_a_a = 3.5;
22
23 reg
24   errorsignal_a;
25
26 integer
27   outfile,
28   cell_state; // internal state of the cell
29
30 initial
31 begin
32   errorsignal_a = 0;
33   cell_state = 0; // Startup state
34   q = 0; // All outputs start at 0
35 end
36
37 always @(posedge a or negedge a) // execute at positive and negative edges of input
38 begin
39   if ($time>4) // arbitrary steady-state time)
40     begin
41       if (errorsignal_a == 1'b1) // A critical timing is active for this input
42         begin
43           outfile = $fopen("errors.txt", "a");
44           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
45           ↪ ", $stime);
46           $fclose(outfile);
47           q <= 1'bX; // Set all outputs to unknown
48         end
49       if (errorsignal_a == 0)
50         begin
51           case (cell_state)
52             0: begin
53               q <= #(delay_state0_a_q) !q;
54               errorsignal_a = 1; // Critical timing on this input; assign
55               ↪ immediately
56               errorsignal_a <= #(ct_state0_a_a) 0; // Clear error signal
57               ↪ after critical timing expires
58             end
59           endcase
60         end
61     end
62   end
63 endmodule

```

Listing 4.2: RSFQ JTL verilog model with self-contained timing.

The digital simulation results for the RSFQ JTL is shown in Fig. 4.4 and the Mealy finite state machine diagram, extracted using *TimEx*, is shown in Fig. 4.5.

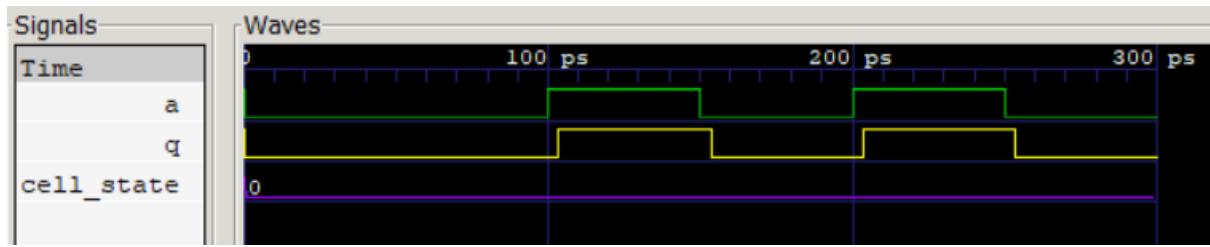


Figure 4.4: RSFQ JTL digital simulation results.

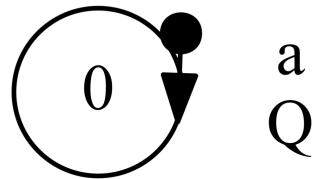


Figure 4.5: RSFQ JTL Mealy finite state machine diagram.

Power Consumption

Table 4.2: RSFQ JTL power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	910	1.03
2	910	2.07
5	910	5.17
10	910	10.3
20	910	20.7
50	910	51.7

4.1.2 SPLIT

The SPLIT cell is used to split a single pulse signal line into two duplicate output pulse signal lines. The cell is not designed to be directly connected to a PTL.

Schematic

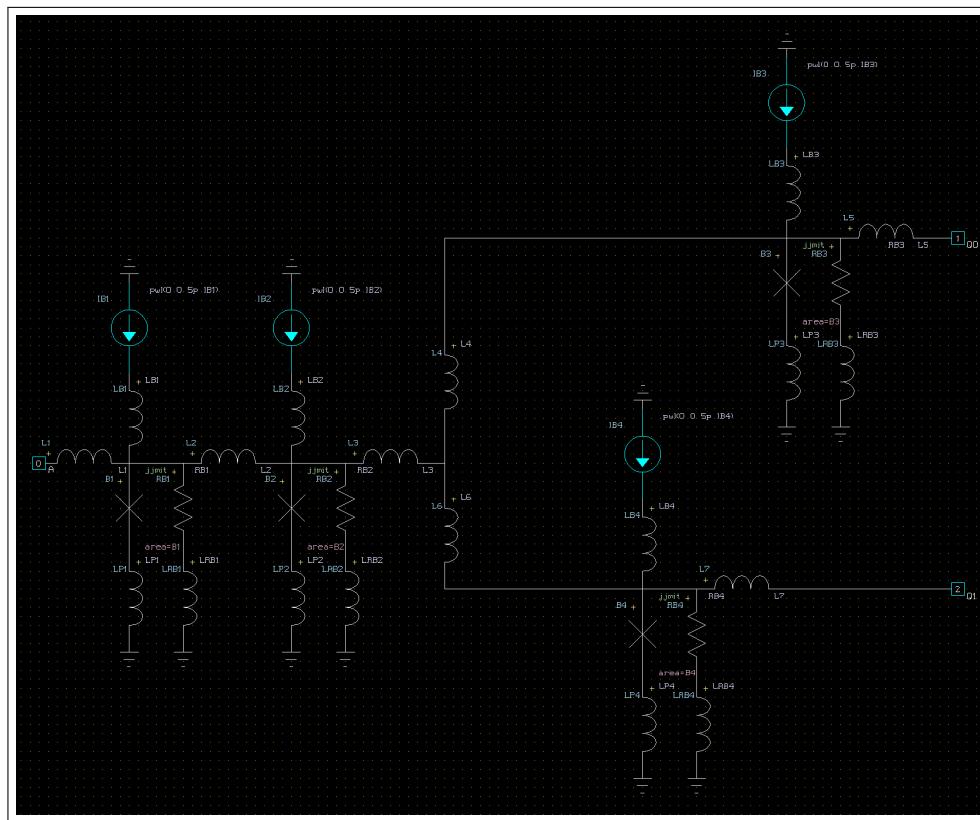


Figure 4.6: Schematic of RSFQ SPLIT.

Layout

The physical layout for the RSFQ SPLIT is shown in Fig. 4.7. The layout height is $70 \mu\text{m}$ and the width is $50 \mu\text{m}$. The cell includes an integrated bias line on M0 within the top row of track blocks. Two bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

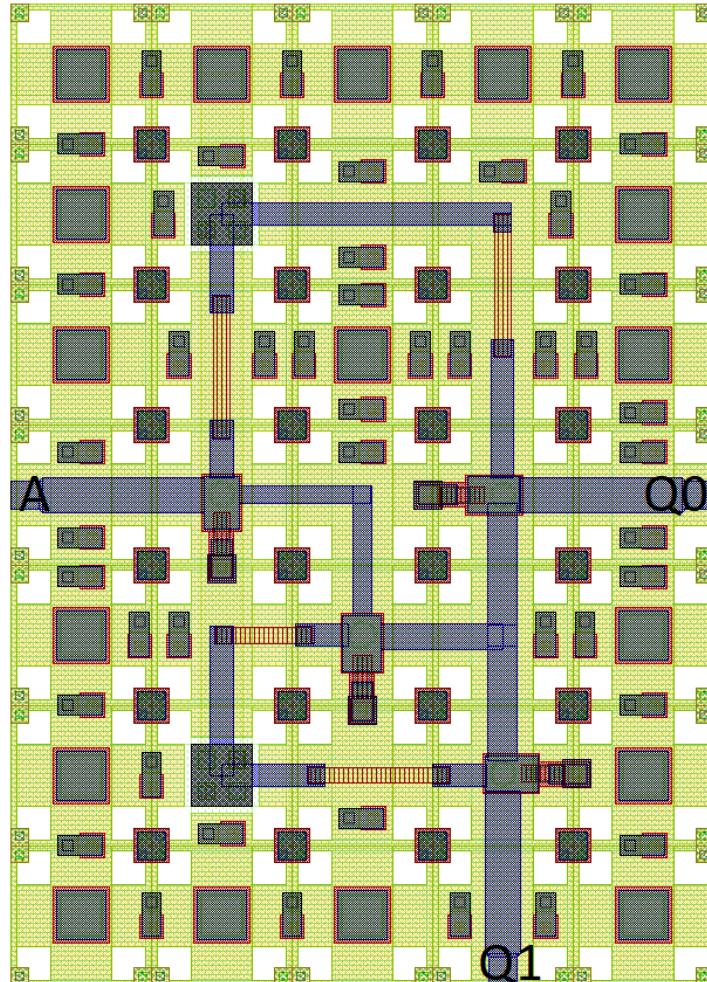


Figure 4.7: RSFQ SPLIT layout.

Analog model

```

1  * Back-annotated simulation file written      43 | .param RB2=B0Rs/B2
2   ↪ by InductEx v.6.0.4 on 1-4-21.          44 | .param RB3=B0Rs/B3
3  * Author: L. Schindler                     45 | .param RB4=B0Rs/B4
4  * Version: 2.1                           46 | .param LRB1=(RB1/Rsheet)*Lsheet
5  * Last modification date: 9 March 2021    47 | .param LRB2=(RB2/Rsheet)*Lsheet
6  * Last modification by: L. Schindler       48 | .param LRB3=(RB3/Rsheet)*Lsheet
7  *$Ports a q0 q1                         49 | .param LRB4=(RB4/Rsheet)*Lsheet
8 .subckt LSmitll_SPLIT a q0 q1           50 |
9 .model jjmit jj(rtype=1, vg=2.8mV, cap   51 | IB1 0 3 pwl(0 0 5p IB1)
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA   52 | IB2 0 6 pwl(0 0 5p IB2)
   ↪ )                                         53 | IB3 0 10 pwl(0 0 5p IB3)
10 .param Phi0=2.067833848E-15            54 | IB4 0 13 pwl(0 0 5p IB4)
11 .param B0=1                            55 | LB1 3 1 9.175E-13
12 .param Ic0=0.0001                      56 | LB2 6 4 7.666E-13
13 .param IcRs=100u*6.859904418          57 | LB3 10 8 1.928E-12
14 .param B0Rs=IcRs/Ic0*B0              58 | LB4 13 11 8.786E-13
15 .param Rsheet=2                        59 |
16 .param Lsheet=1.13e-12                  60 | B1 1 2 jjmit area=B1
17 .param LP=0.2p                         61 | B2 4 5 jjmit area=B2
18 .param IC=2.5                          62 | B3 8 9 jjmit area=B3
19 .param Lptl=2p                         63 | B4 11 12 jjmit area=B4
20 .param LB=2p                           64 | L1 a 1 2.063E-12
21 .param BiasCoef=0.7                   65 | L2 1 4 3.637E-12
22 .param RD=1.36                         66 | L3 4 7 1.278E-12
23                               67 | L4 7 8 1.305E-12
24 .param B1=2.5                          68 | L5 8 q0 2.05E-12
25 .param B2=3.0                           69 | L6 7 11 1.315E-12
26 .param B3=2.5                          70 | L7 11 q1 2.06E-12
27 .param B4=2.5                          71 |
28                               72 | LP1 2 0 4.676E-13
29 .param IB1=175u                        73 | LP2 5 0 4.498E-13
30 .param IB2=280u                        74 | LP3 9 0 5.183E-13
31 .param IB3=175u                        75 | LP4 12 0 4.639E-13
32 .param IB4=175u                        76 | RB1 1 101 RB1
33                               77 | LRB1 101 0 LRB1
34 .param L1=Lptl                         78 | RB2 4 104 RB2
35 .param L2=Phi0/(2*B1*Ic0)             79 | LRB2 104 0 LRB2
36 .param L3=(Phi0/(2*B2*Ic0))/2        80 | RB3 8 108 RB3
37 .param L4=L3                           81 | LRB3 108 0 LRB3
38 .param L5=Lptl                         82 | RB4 11 111 RB4
39 .param L6=L3                           83 | LRB4 111 0 LRB4
40 .param L7=Lptl                         84 | .ends
41                               84 |
42 .param RB1=B0Rs/B1

```

Listing 4.3: RSFQ SPLIT JoSIM netlist.

Table 4.3: RSFQ SPLIT pin list.

Pin	Description
a	Data input
q0	Data output
q1	Data output

The JoSIM simulation results for the RSFQ SPLIT are shown in Fig. 4.8. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the phase over the output JJ of pin **q0**,
- (d) the phase over the output JJ of pin **q1**,
- (e) the phase over the input JJ of the load cell connected to pin **q0**, and
- (f) the phase over the input JJ of the load cell connected to pin **q1**.

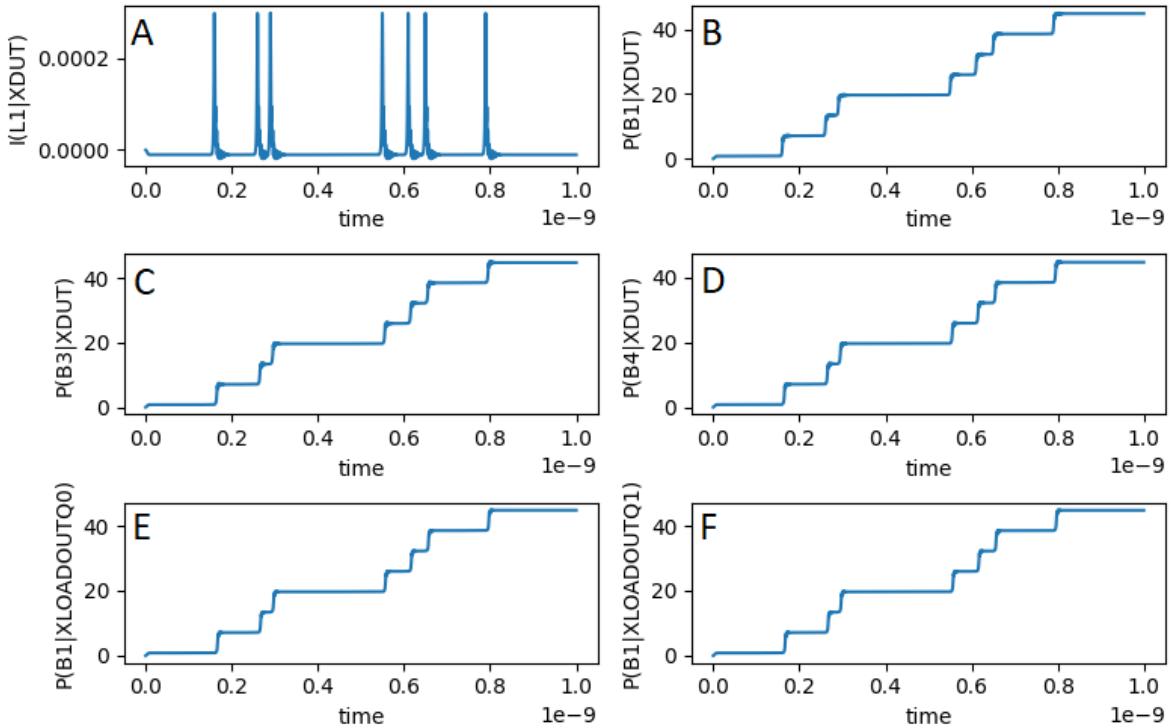


Figure 4.8: RSFQ SPLIT analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 3 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 'timescale 1ps/100fs
8 module LSmitll_SPLIT_v2p1_optimized (a, q0, q1);
9
10 input
11   a;
12
13 output
14   q0, q1;
15
16 reg
17   q0, q1;
18
19 real
20   delay_state0_a_q0 = 6.8,
21   delay_state0_a_q1 = 6.5,
22   ct_state0_a_a = 3.4;
23
24 reg
25   errorsignal_a;
26 integer
27   outfile,
28   cell_state; // internal state of the cell
29
30 initial
31 begin
32   errorsignal_a = 0;
33   cell_state = 0; // Startup state
34   q0 = 0; // All outputs start at 0
35   q1 = 0; // All outputs start at 0
36 end
37
38 always @(posedge a or negedge a) // execute at positive and negative edges of input
39 begin
40   if ($time>4) // arbitrary steady-state time)
41     begin
42       if (errorsignal_a == 1'b1) // A critical timing is active for this input
43         begin
44           outfile = $fopen("errors.txt", "a");
45           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
46           ↪ ", $stime);
47           $fclose(outfile);
48           q0 <= 1'bX; // Set all outputs to unknown
49           q1 <= 1'bX; // Set all outputs to unknown
50         end
51       if (errorsignal_a == 0)
52         begin
53           case (cell_state)
54             0: begin
55               q0 <= #(delay_state0_a_q0) !q0;
56               q1 <= #(delay_state0_a_q1) !q1;
57               errorsignal_a = 1; // Critical timing on this input; assign
58               ↪ immediately
59               errorsignal_a <= #(ct_state0_a_a) 0; // Clear error signal
59               ↪ after critical timing expires
60             end
61           endcase
62         end
63     end
64   end
65 endmodule

```

Listing 4.4: RSFQ SPLIT verilog model.

The digital simulation results for the RSFQ SPLIT is shown in Fig. 4.9 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 4.10.

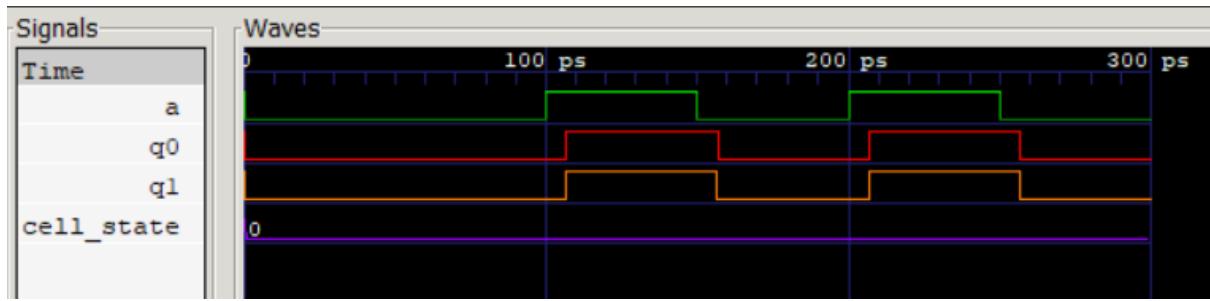


Figure 4.9: RSFQ SPLIT digital simulation results.

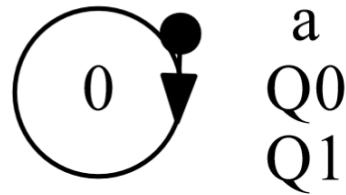


Figure 4.10: RSFQ SPLIT Mealy finite state diagram.

Power consumption

Table 4.4: RSFQ SPLIT power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2093	2.17
2	2093	4.34
5	2093	10.9
10	2093	21.7
20	2093	43.4
50	2093	108.6

4.1.3 MERGE

The MERGE joins two input pulse signal lines and provides a single output pulse signal line. If there is a pulse on either input lines, the MERGE will generate a pulse on the output signal line. The MERGE is not designed to be directly connected to a PTL.

Schematic

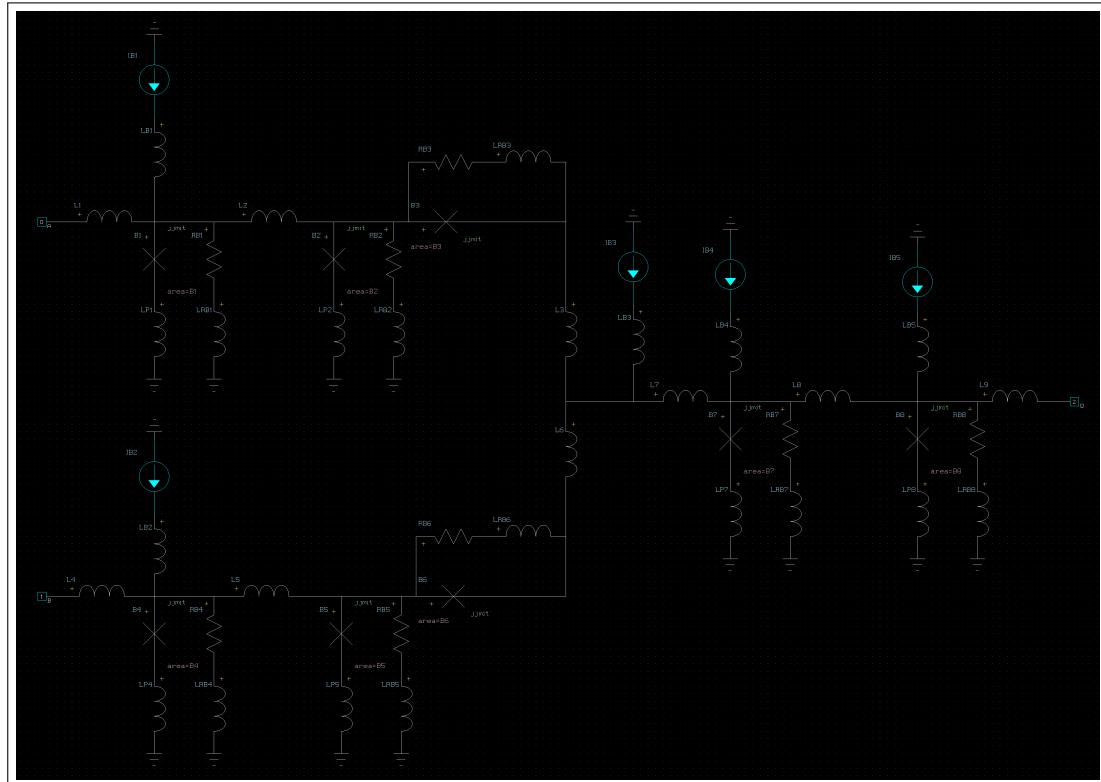


Figure 4.11: Schematic of RSFQ MERGE.

Layout

The physical layout of the RSFQ MERGE is shown in Fig. 4.12. The height of the layout is $70 \mu m$ and the width is $70 \mu m$. The cell includes an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

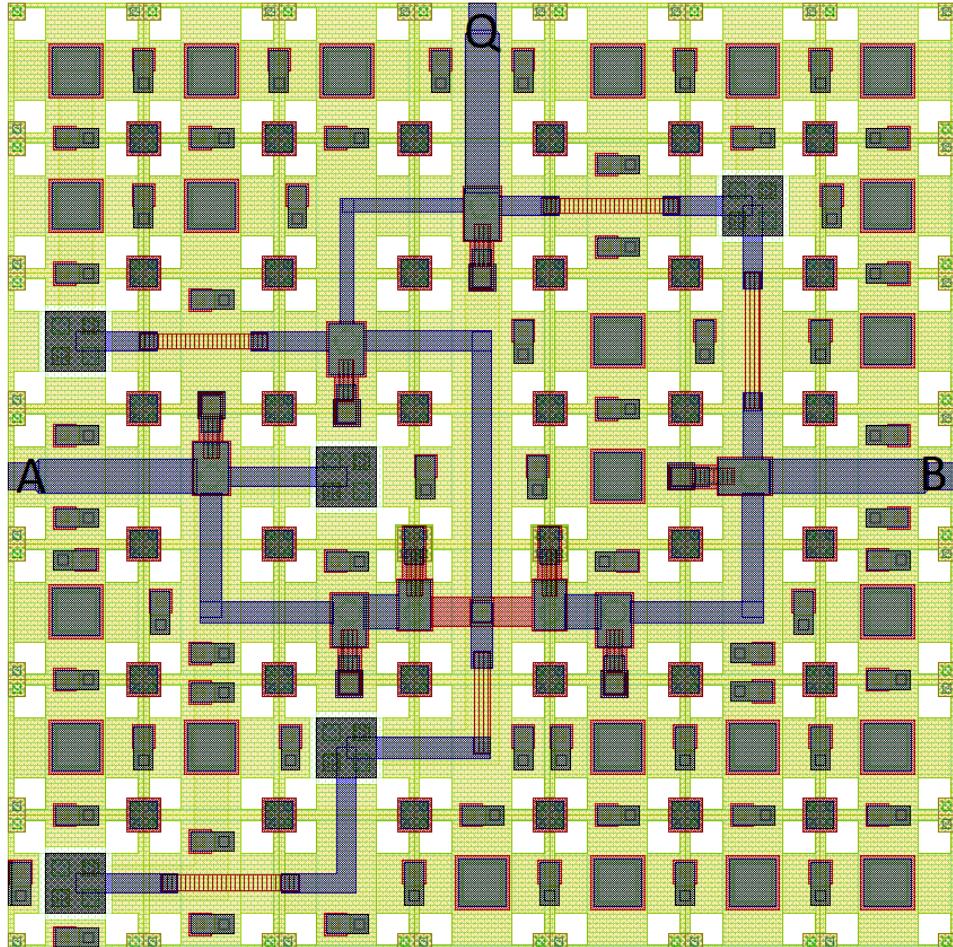


Figure 4.12: RSFQ MERGE layout.

Analog model

```

1  * Back-annotated simulation file written      65 | .param RB7=B0Rs/B7
2   ↪ by InductEx v.6.0.4 on 19-3-21.          66 | .param RB8=B0Rs/B8
3  * Author: L. Schindler                      67 | .param LRB1=(RB1/Rsheet)*Lsheet
4  * Version: 2.1                               68 | .param LRB2=(RB2/Rsheet)*Lsheet
5  * Last modification date: 3 June 2021       69 | .param LRB3=(RB3/Rsheet)*Lsheet
6  * Last modification by: L. Schindler        70 | .param LRB4=(RB4/Rsheet)*Lsheet
7  *$Ports a b q                                71 | .param LRB5=(RB5/Rsheet)*Lsheet
8 .subckt LSmitll_MERGE a b q                  72 | .param LRB6=(RB6/Rsheet)*Lsheet
9 .model jjmit jj(rtype=1, vg=2.8mV, cap     73 | .param LRB7=(RB7/Rsheet)*Lsheet
10    ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    74 | .param LRB8=(RB8/Rsheet)*Lsheet
11 .param Phi0=2.067833848E-15                 75 |
12 .param B0=1                                    76 | B1 1 2 jjmit area=B1
13 .param Ic0=0.0001                            77 | B2 4 5 jjmit area=B2
14 .param IcRs=100u*6.859904418                 78 | B3 4 6 jjmit area=B3
15 .param B0Rs=IcRs/Ic0*B0                     79 | B4 8 9 jjmit area=B4
16 .param Rsheet=2                             80 | B5 11 12 jjmit area=B5
17 .param Lsheet=1.13e-12                      81 | B6 11 13 jjmit area=B6
18 .param LP=0.2p                                82 | B7 15 16 jjmit area=B7
19 .param IC=2.5                                83 | B8 18 19 jjmit area=B8
20 .param LB=2p                                 84 |
21 .param BiasCoef=0.70                         85 | IB1 0 3 pw1(0 0 5p IB1)
22 .param RD=1.36                                86 | IB2 0 10 pw1(0 0 5p IB2)
23 .param B1=IC                                 87 | IB3 0 14 pw1(0 0 5p IB3)
24 .param B2=2.5                                88 | IB4 0 17 pw1(0 0 5p IB4)
25 .param B3=1.92                                89 | IB5 0 20 pw1(0 0 5p IB5)
26 .param B4=B1                                 90 |
27 .param B5=B2                                 91 | L1 a 1 2.117E-12
28 .param B6=B3                                 92 | L2 1 4 3.17E-12
29 .param B7=2.53                               93 | L3 6 7 1.234E-12
30 .param B8=IC                                 94 | L4 b 8 2.082E-12
31 |                                         95 | L5 8 11 3.165E-12
32 .param IB1=BiasCoef*Ic0*B1                 96 | L6 13 7 1.224E-12
33 .param IB2=IB1                               97 | L7 7 15 5.299E-12
34 .param IB3=254E-6                           98 | L8 15 18 4.489E-12
35 .param IB4=192E-6                           99 | L9 18 q 2.077E-12
36 .param IB5=BiasCoef*Ic0*B8                 100 |
37 |                                         101 | LP1 2 0 4.652E-13
38 .param L1=Phi0/(4*IC*Ic0)                   102 | LP2 5 0 4.457E-13
39 .param L2=3.173E-12                         103 | LP4 9 0 5.293E-13
40 .param L3=1.2E-12                           104 | LP5 12 0 4.452E-13
41 .param L4=L1                                105 | LP7 16 0 5.039E-13
42 .param L5=L2                                106 | LP8 19 0 4.984E-13
43 .param L6=L3                                107 | LB1 1 3 LB1
44 .param L7=5.354E-12                         108 | LB2 8 10 LB2
45 .param L8=4.456E-12                         109 | LB3 7 14 LB3
46 .param L9=Phi0/(4*B8*Ic0)                   110 | LB4 15 17 LB4
47 |                                         111 | LB5 18 20 LB5
48 .param LB1=LB                                112 | RB1 1 101 RB1
49 .param LB2=LB                                113 | LRB1 101 0 LRB1
50 .param LB3=LB                                114 | RB2 4 104 RB2
51 .param LB4=LB                                115 | LRB2 104 0 LRB2
52 .param LB5=LB                                116 | RB3 4 106 RB3
53 .param LP1=LP                                117 | LRB3 106 6 LRB3
54 .param LP2=LP                                118 | RB4 8 108 RB4
55 .param LP4=LP                                119 | LRB4 108 0 LRB4
56 .param LP5=LP                                120 | RB5 11 111 RB5
57 .param LP7=LP                                121 | LRB5 111 0 LRB5
58 .param LP8=LP                                122 | RB6 11 113 RB6
59 .param RB1=B0Rs/B1                          123 | LRB6 113 13 LRB6
60 .param RB2=B0Rs/B2                          124 | RB7 15 115 RB7
61 .param RB3=B0Rs/B3                          125 | LRB7 115 0 LRB7
62 .param RB4=B0Rs/B4                          126 | RB8 18 118 RB8
63 .param RB5=B0Rs/B5                          127 | LRB8 118 0 LRB8
64 .param RB6=B0Rs/B6                          128 | .ends

```

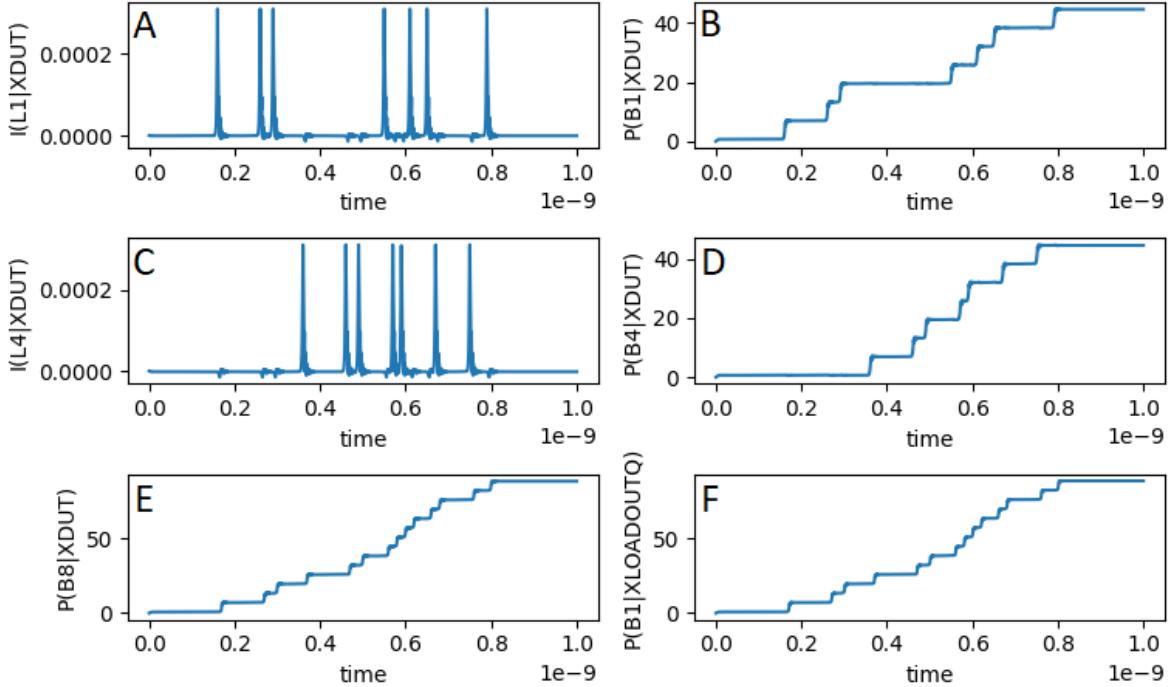
Listing 4.5: RSFQ MERGE JoSIM netlist.

Table 4.5: RSFQ MERGE pin list.

Pin	Description
a	Data input
b	Data input
q	Data output

The JoSIM simulation results for the RSFQ MERGE are shown in Fig. 4.13. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b**,
- (d) the phase over the input JJ of pin **b**,
- (e) the phase over the output JJ of pin **q**,
- (f) the phase over the input JJ of the load cell connected to pin **q**.

**Figure 4.13:** RSFQ MERGE analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 3 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 //
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 //
15 'timescale 1ps/100fs
16 module LSmitll_MERGE_v2p1_optimized (a, b, q);
17
18 input
19   a, b;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state0_a_q = 10.7,
29   delay_state0_b_q = 10.7,
30   ct_state0_a_a = 8.4,
31   ct_state0_a_b = 3.0,
32   ct_state0_b_a = 3.0,
33   ct_state0_b_b = 8.9;
34
35 reg
36   errorsignal_a,
37   errorsignal_b;
38
39 integer
40   outfile,
41   cell_state; // internal state of the cell
42
43 initial
44 begin
45   errorsignal_a = 0;
46   errorsignal_b = 0;
47   cell_state = 0; // Startup state
48   q = 0; // All outputs start at 0
49 end
50
51 always @ (posedge a or negedge a) // execute at positive and negative edges of input
52 begin
53   if ($time > 4) // arbitrary steady-state time)
54     begin
55       if (errorsignal_a == 1'b1) // A critical timing is active for this input
56         begin
57           outfile = $fopen("errors.txt", "a");
58           $fdisplay(outfile, "Violation of critical timing in module %m; %0d ps.\n"
59           ↪ ", $stime);
60           $fclose(outfile);
61           q <= 1'bX; // Set all outputs to unknown
62         end
63       if (errorsignal_a == 0)
64         begin
65           case (cell_state)
66             0: begin
67               q <= #(delay_state0_a_q) !q;

```

```

67          errorsignal_a = 1; // Critical timing on this input; assign
68          //      ↪ immediately
69          errorsignal_a <= #(ct_state0_a_a) 0; // Clear error signal
70          //      ↪ after critical timing expires
71          errorsignal_b = 1; // Critical timing on this input; assign
72          //      ↪ immediately
73          errorsignal_b <= #(ct_state0_a_b) 0; // Clear error signal
74          //      ↪ after critical timing expires
75      end
76  endcase
77 end
78
79 always @(posedge b or negedge b) // execute at positive and negative edges of input
80 begin
81     if ($time>4) // arbitrary steady-state time)
82     begin
83         if (errorsignal_b == 1'b1) // A critical timing is active for this input
84         begin
85             outfile = $fopen("errors.txt", "a");
86             $fdisplay(outfile, "Violation of critical timing in module %m; %0d ps.\n"
87                         //      ↪ ", $stime);
88             $fclose(outfile);
89             q <= 1'bX; // Set all outputs to unknown
90         end
91         if (errorsignal_b == 0)
92         begin
93             case (cell_state)
94                 0: begin
95                     q <= #(delay_state0_b_q) !q;
96                     errorsignal_a = 1; // Critical timing on this input; assign
97                     //      ↪ immediately
98                     errorsignal_a <= #(ct_state0_b_a) 0; // Clear error signal
99                     //      ↪ after critical timing expires
100                    errorsignal_b = 1; // Critical timing on this input; assign
101                    //      ↪ immediately
102                    errorsignal_b <= #(ct_state0_b_b) 0; // Clear error signal
103                    //      ↪ after critical timing expires
104                end
105            endcase
106        end
107    end
108 endmodule

```

Listing 4.6: RSFQ MERGE verilog model.

The digital simulation results for the RSFQ MERGE is shown in Fig. 4.14 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 4.15.



Figure 4.14: RSFQ MERGE digital simulation results.

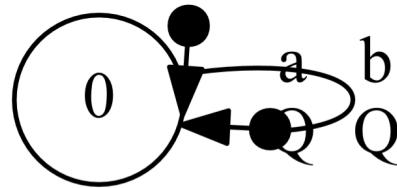


Figure 4.15: RSFQ MERGE Mealy finite state diagram.

Power consumption

Table 4.6: RSFQ MERGE power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2526	3.09
2	2526	7.80
5	2526	19.5
10	2526	39.0
20	2526	78.0
50	2526	195

4.1.4 PTLTX

The RSFQ PTLTX is a cell which transmits a pulse signal over a PTL. It is connected to a cell which is not designed to connect to PTLs, if a PTL connection is required.

Schematic

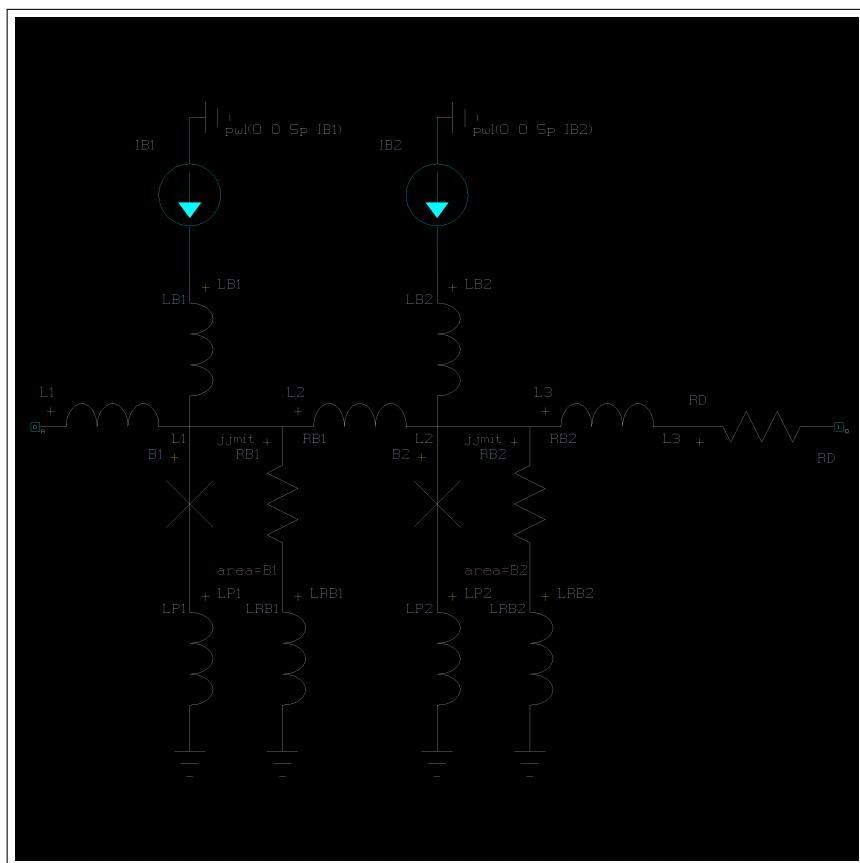


Figure 4.16: Schematic of RSFQ PTLTX.

Layout

The physical layout for the RSFQ PTLTX is shown in Fig. 4.17. The layout height is $70 \mu m$ and the width is $30 \mu m$. The cell includes an integrated bias line on M0 within the top row of track blocks. A bias pillar from M0 to M6 connects the circuit to the integrated bias line. The PTL connection is only valid for a PTL attached on M3. The connection does not extend to M1, as this layer is used as a sky plane for the bias line on M0. All bias resistors are designed to be connected to a 2.6 mV voltage source.

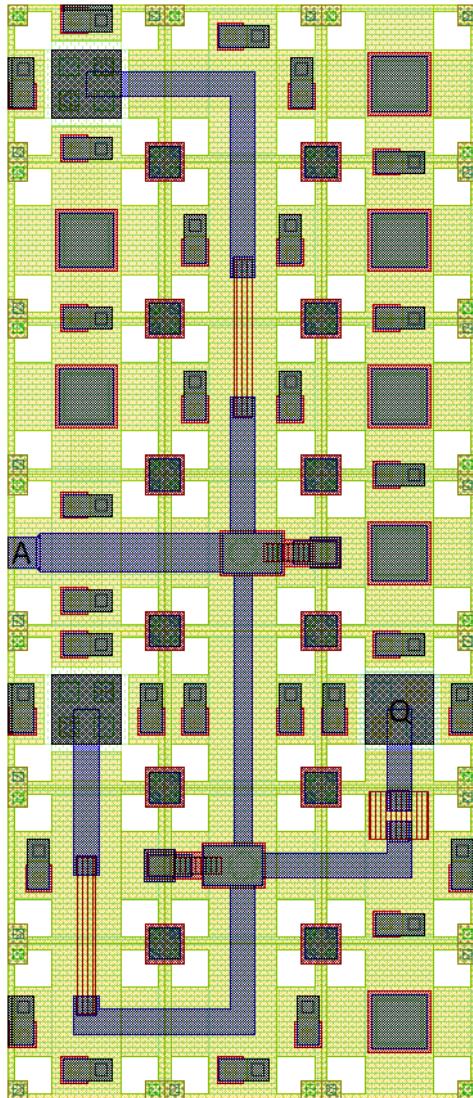


Figure 4.17: RSFQ PTLTX Layout.

Analog model

```

1  * Back-annotated simulation file written      29 | .param L2=Phi0/(2*B1*Ic0)
2   ↪ by InductEx v.6.0 on 2021/06/22.          30 | .param L3=Lptl
3  * Author: L. Schindler                      31 | .param LB1=LB
4  * Version: 2.1                               32 | .param LB2=LB
5  * Last modification date: 22 June 2021       33 | .param LP1=LP
6  * Last modification by: L. Schindler         34 | .param LP2=LP
7  *$Ports a q                                35 | .param RB1=B0Rs/B1
8 .subckt LSmitll_PTLTX a q                  36 | .param RB2=B0Rs/B2
9 .model jjmit jj(rtype=1, vg=2.8mV, cap    37 | .param LRB1=(RB1/Rsheet)*Lsheet
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA     38 | .param LRB2=(RB2/Rsheet)*Lsheet
   ↪ )                                         39 |
10 .param Phi0=2.067833848E-15                40 | B1 1 2 jjmit area=B1
11 .param B0=1                                 41 | B2 4 5 jjmit area=B2
12 .param Ic0=0.0001                           42 | IB1 0 3 pw1(0 0 5p IB1)
13 .param IcRs=100u*6.859904418               43 | IB2 0 6 pw1(0 0 5p IB2)
14 .param B0Rs=IcRs/Ic0*B0                   44 | LB1 1 3 1.684E-012
15 .param Rsheet=2                            45 | LB2 4 6 3.596E-012
16 .param Lsheet=1.13e-12                     46 | L1 a 1 2.063E-012
17 .param LP=0.5p                            47 | L2 1 4 4.123E-012
18 .param LB=2p                             48 | L3 4 7 2.193E-012
19 .param Lptl=2p                           49 | RD 7 q RD
20 .param RD=1.36                           50 | LP1 2 0 5.254E-013
21 .param IC=2.5                            51 | LP2 5 0 5.141E-013
22 .param BiasCoef=0.7                      52 | RB1 1 101 RB1
23 .param B1=IC                            53 | RB2 4 104 RB2
24 .param B2=IC                            54 | LRB1 101 0 LRB1
25 .param IB1=BiasCoef*Ic0*B1             55 | LRB2 104 0 LRB2
26 .param IB2=BiasCoef*Ic0*B2             56 | .ends
27 .param L1=Phi0/(4*B1*Ic0)
28

```

Listing 4.7: RSFQ PTLTX JoSIM netlist.

Table 4.7: RSFQ PTLTX pin list.

Pin	Description
a	Data input
q	Data output

The simulation results for the RSFQ PTLTX using JoSIM is shown in Fig. 4.18. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the phase over the output JJ of pin **q**, and
- (d) the phase over the input JJ of the load circuit connected via a PTL to the PTLTX.

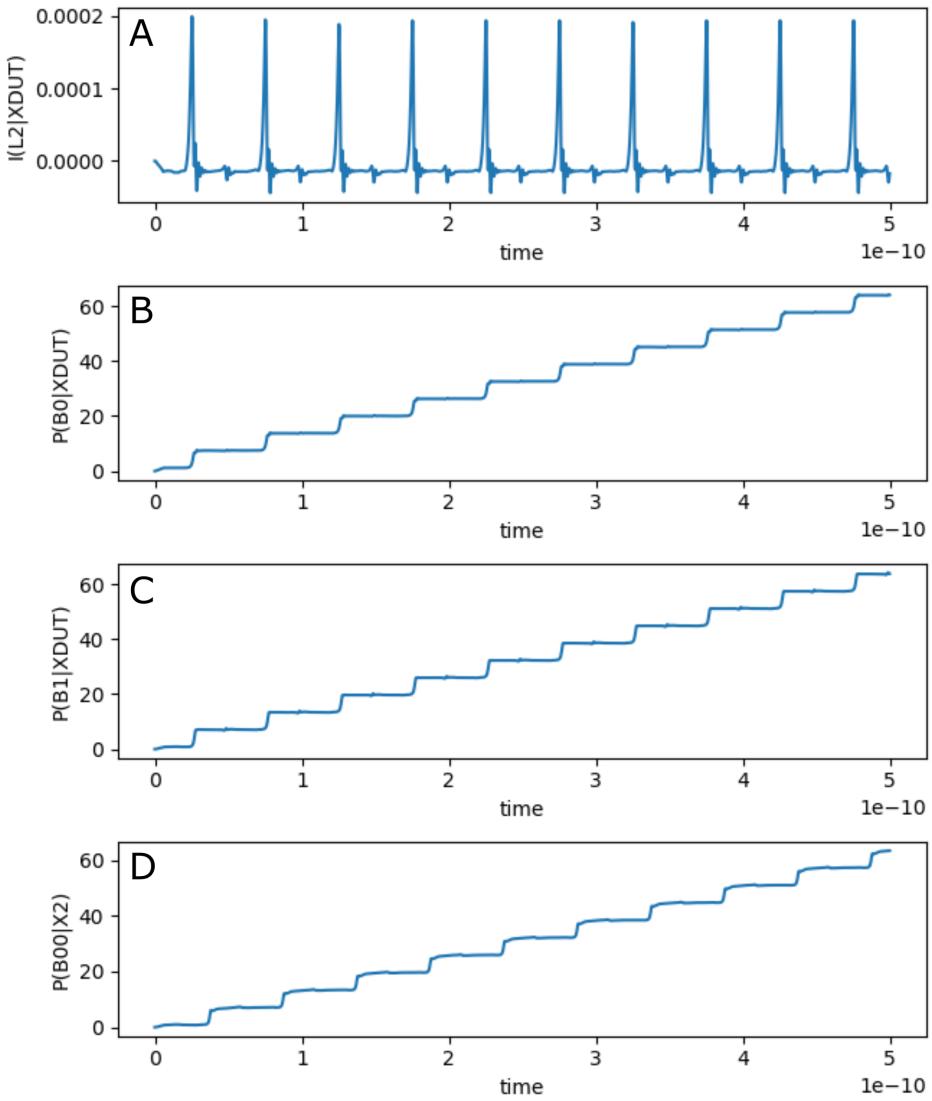


Figure 4.18: RSFQ PTLTX analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 22 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmitll_ptltx_v2p1_optimized (a, q);
9
10 input
11   a;
12
13 output
14   q;
15
16 reg
17   q;
18
19 real
20   delay_state0_a_q = 3.3,
21   ct_state0_a_a = 7.2;
22
23 reg
24   errorsignal_a;
25
26 integer
27   outfile,
28   cell_state; // internal state of the cell
29
30 initial
31 begin
32   errorsignal_a = 0;
33   cell_state = 0; // Startup state
34   q = 0; // All outputs start at 0
35 end
36
37 always @(posedge a or negedge a) // execute at positive and negative edges of input
38 begin
39   if ($time>4) // arbitrary steady-state time)
40     begin
41       if (errorsignal_a == 1'b1) // A critical timing is active for this input
42         begin
43           outfile = $fopen("errors.txt", "a");
44           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
45           ↪ ", $stime);
46           $fclose(outfile);
47           q <= 1'bX; // Set all outputs to unknown
48         end
49       if (errorsignal_a == 0)
50         begin
51           case (cell_state)
52             0: begin
53               q <= #(delay_state0_a_q) !q;
54               errorsignal_a = 1; // Critical timing on this input; assign
55               ↪ immediately
56               errorsignal_a <= #(ct_state0_a_a) 0; // Clear error signal
57               ↪ after critical timing expires
58             end
59           endcase
60         end
61     end
62   end
63 endmodule

```

Listing 4.8: RSFQ PTLTX verilog model.

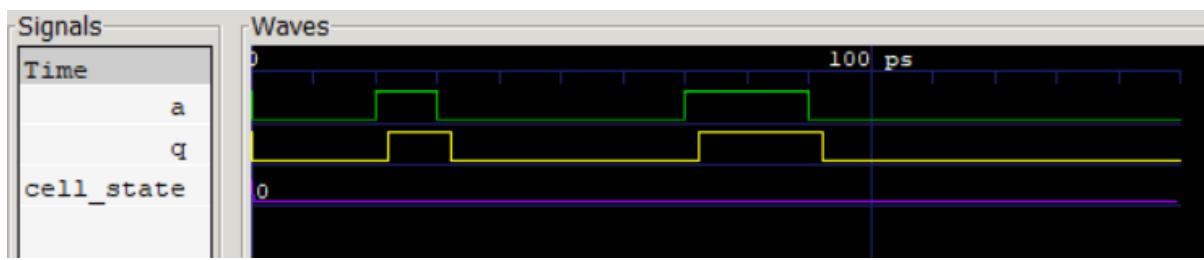


Figure 4.19: RSFQ PTLTX digital simulation results.

The digital simulation results for the RSFQ PTLTX is shown in Fig. 4.19.

Power Consumption

Table 4.8: RSFQ PTLTX power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	910	1.03
2	910	2.07
5	910	5.17
10	910	10.3
20	910	20.7
50	910	51.8

4.1.5 PTLRX

The PTLRX is a receiver cell which receives a pulse signal from a PTL. It is connected to cells that are not designed to connect to PTLs when a PTL connection is required.

Schematic

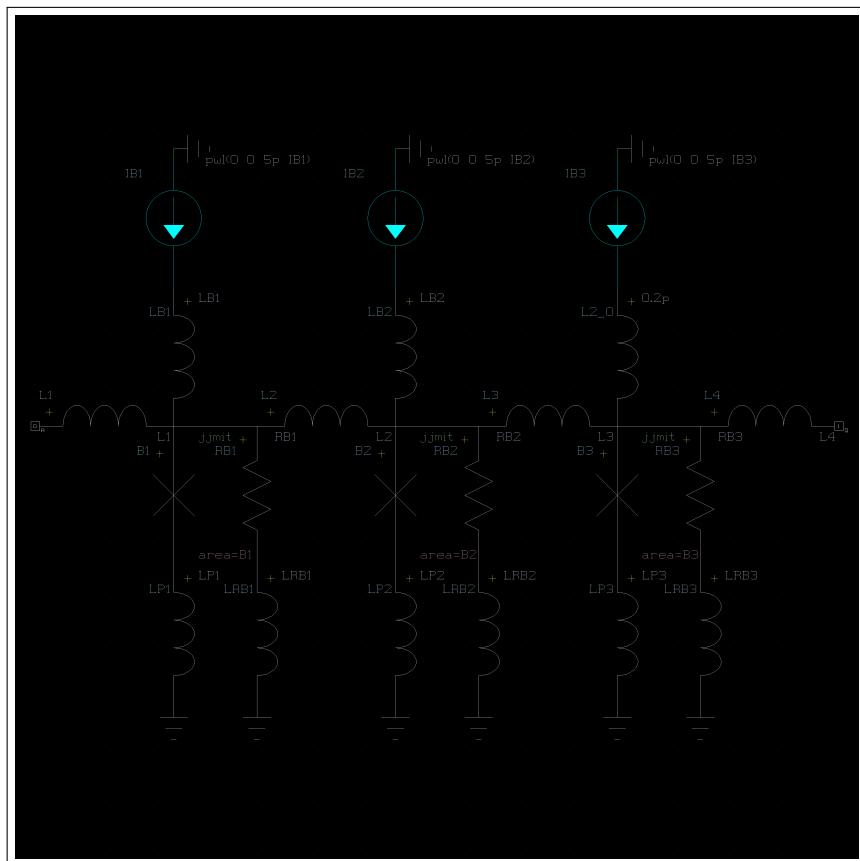


Figure 4.20: Schematic of RSFQ PTLRX.

Layout

The physical layout for the RSFQ PTLRX is shown in Fig. 4.21. The layout height is $70 \mu m$ and the width is $40 \mu m$. The cell includes an integrated bias line on M0 within the top row of track blocks. A bias pillar from M0 to M6 connects the circuit to the integrated bias line. The PTL connection is only valid for a PTL attached on M3. The connection does not extend to M1, as this layer is used as a sky plane for the bias line on M0. All bias resistors are designed to be connected to a 2.6 mV voltage source.

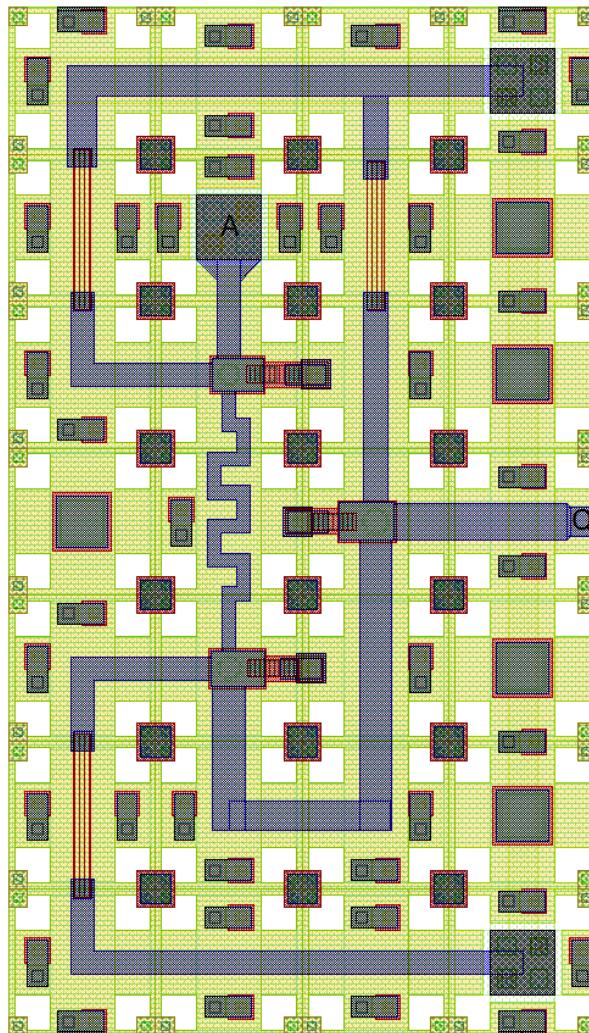


Figure 4.21: RSFQ PTLRX Layout.

Analog model

```

1  * Back-annotated simulation file written      35 | .param LB3=LB
2   ↪ by InductEx v.6.0 on 2021/06/22.          36 | .param LP1=LP
3  * Author: L. Schindler                      37 | .param LP2=LP
4  * Version: 2.1                               38 | .param LP3=LP
5  * Last modification date: 22 June 2021       39 | .param RB1=B0Rs/B1
6  * Last modification by: L. Schindler        40 | .param RB2=B0Rs/B2
7  *$Ports a q                                41 | .param RB3=B0Rs/B3
8 .subckt LSmitll_PTLRX a q                   42 | .param LRB1=(RB1/Rsheet)*Lsheet
9 .model jjmit jj(rtype=1, vg=2.8mV, cap     43 | .param LRB2=(RB2/Rsheet)*Lsheet
10    ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA   44 | .param LRB3=(RB3/Rsheet)*Lsheet
11   ↪ )
12 .param Phi0=2.067833848E-15                 45 |
13 .param B0=1                                  46 | B1 1 2 jjmit area=B1
14 .param Ic0=0.0001                            47 | B2 4 5 jjmit area=B2
15 .param IcRs=100u*6.859904418                48 | B3 7 8 jjmit area=B3
16 .param B0Rs=IcRs/Ic0*B0                     49 | IB1 0 3 pw1(0 0 5p IB1)
17 .param Rsheet=2                             50 | LB1 1 3 2.777E-012
18 .param Lsheet=1.13e-12                       51 | IB2 0 6 pw1(0 0 5p IB2)
19 .param LP=0.5p                             52 | LB2 4 6 2.685E-012
20 .param LB=2p                               53 | IB3 0 9 pw1(0 0 5p IB3)
21 .param Lptl=2p                            54 | LB3 7 9 2.764E-012
22 .param IC=2.5                             55 | L1 a 1 1.346E-012
23 .param B1=IC/1.54                           56 | L2 1 4 6.348E-012
24 .param B2=IC/1.25                           57 | L3 4 7 5.197E-012
25 .param B3=IC                               58 | L4 7 q 2.058E-012
26 .param IB1=Ic0*B1                          59 | LP1 2 0 4.795E-013
27 .param IB2=BiasCoef*Ic0*B2                60 | LP2 5 0 5.431E-013
28 .param IB3=BiasCoef*Ic0*B3                61 | LP3 8 0 5.339E-013
29 .param L1=Lptl                            62 | RB1 1 101 RB1
30 .param L2=Phi0/(2*B1*Ic0)                  63 | RB2 4 104 RB2
31 .param L3=Phi0/(2*B2*Ic0)                  64 | RB3 7 107 RB3
32 .param L4=Phi0/(4*IC*Ic0)                  65 | LRB1 101 0 LRB1
33 .param LB1=LB                            66 | LRB2 104 0 LRB2
34 .param LB2=LB                            67 | LRB3 107 0 LRB3
35 |
36 |
37 |
38 |
39 |
40 |
41 |
42 |
43 |
44 |
45 |
46 |
47 |
48 |
49 |
50 |
51 |
52 |
53 |
54 |
55 |
56 |
57 |
58 |
59 |
60 |
61 |
62 |
63 |
64 |
65 |
66 |
67 |
68 |

```

Listing 4.9: RSFQ PTLRX JoSIM netlist.

Table 4.9: RSFQ PTLRX pin list.

Pin	Description
a	Data input
q	Data output

The simulation results for the RSFQ PTLRX using JoSIM is shown in Fig. 4.22. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the phase over the output JJ of pin **q**, and
- (d) the phase over the input JJ of the load circuit connected to pin **q**.

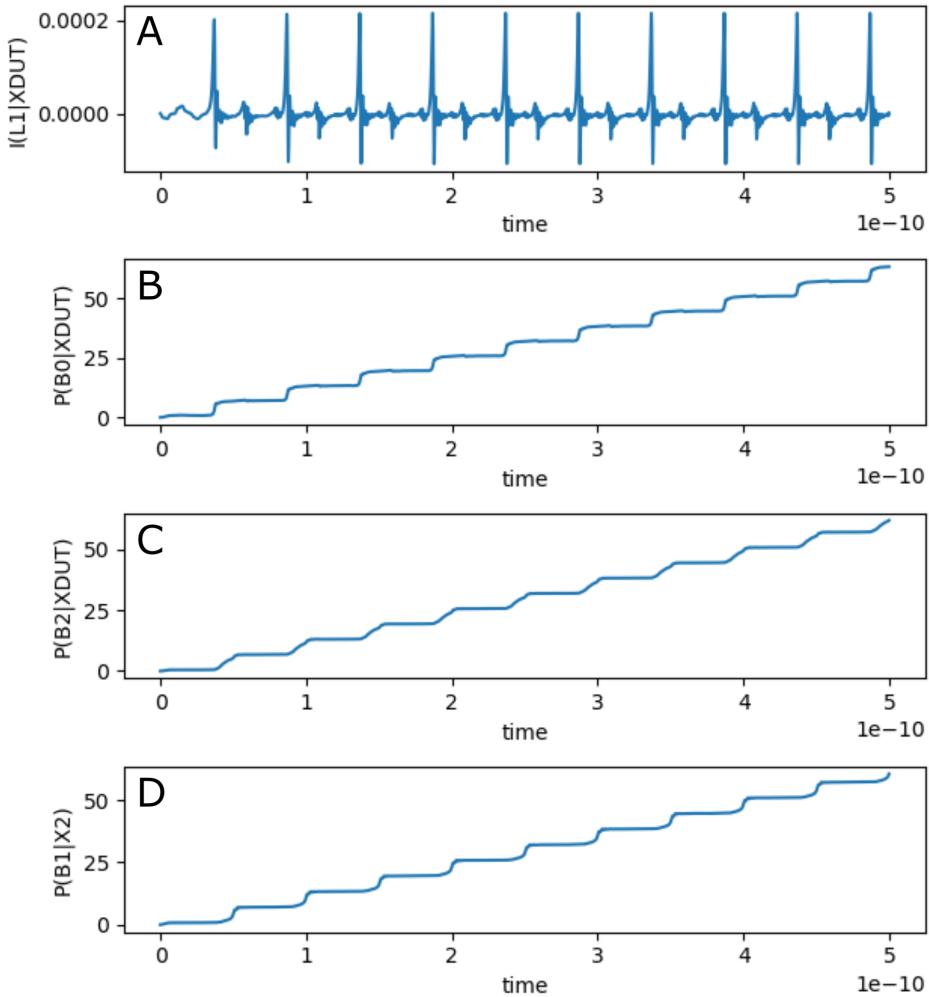


Figure 4.22: RSFQ PTLRX analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 22 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmitll_ptlrx_v2p1_optimized (a, q);
9
10 input
11   a;
12
13 output
14   q;
15
16 reg
17   q;
18
19 real
20   delay_state0_a_q = 5.5,
21   ct_state0_a_a = 6.8;
22
23 reg
24   errorsignal_a;
25
26 integer
27   outfile,
28   cell_state; // internal state of the cell
29
30 initial
31 begin
32   errorsignal_a = 0;
33   cell_state = 0; // Startup state
34   q = 0; // All outputs start at 0
35 end
36
37 always @(posedge a or negedge a) // execute at positive and negative edges of input
38 begin
39   if ($time>4) // arbitrary steady-state time)
40     begin
41       if (errorsignal_a == 1'b1) // A critical timing is active for this input
42         begin
43           outfile = $fopen("errors.txt", "a");
44           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
45           ↪ ", $stime);
46           $fclose(outfile);
47           q <= 1'bX; // Set all outputs to unknown
48         end
49       if (errorsignal_a == 0)
50         begin
51           case (cell_state)
52             0: begin
53               q <= #(delay_state0_a_q) !q;
54               errorsignal_a = 1; // Critical timing on this input; assign
55               ↪ immediately
56               errorsignal_a <= #(ct_state0_a_a) 0; // Clear error signal
57               ↪ after critical timing expires
58             end
59           endcase
60         end
61     end
62   end
63 endmodule

```

Listing 4.10: RSFQ PTLRX verilog model.



Figure 4.23: RSFQ PTLRX digital simulation results.

The digital simulation results for the RSFQ PTLRX is shown in Fig. 4.23.

Power Consumption

Table 4.10: RSFQ PTLRX power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	782	1.27
2	782	2.53
5	782	6.33
10	782	12.7
20	782	25.3
50	782	63.3

4.1.6 Always0 Asynchronous

The Always0 Asynchronous cell provides an output which is always zero. Two versions of the cell exist: One with an **a** input port, as seen in Fig. 4.24, and one without an **a** input port, as seen in Fig. 4.25. The Always0 Asynchronous is not designed to be directly connected to a PTL.

Schematic

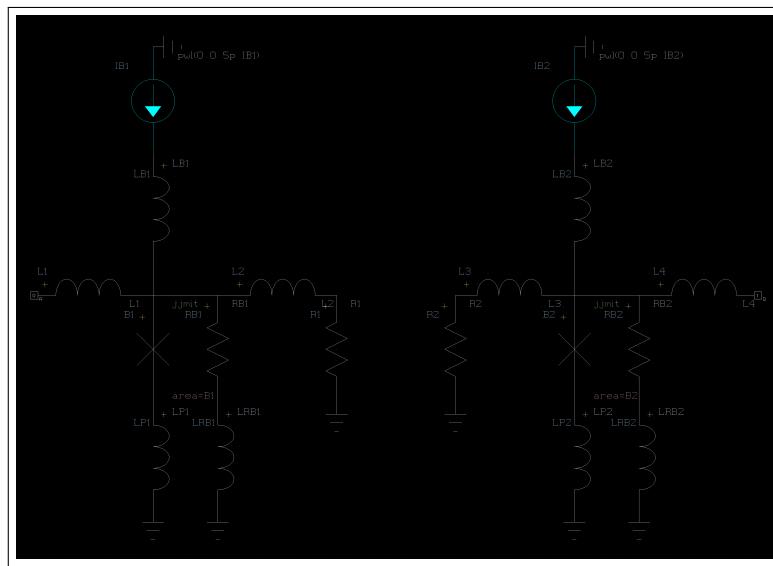


Figure 4.24: Schematic of RSFQ Always0 Asynchronous.

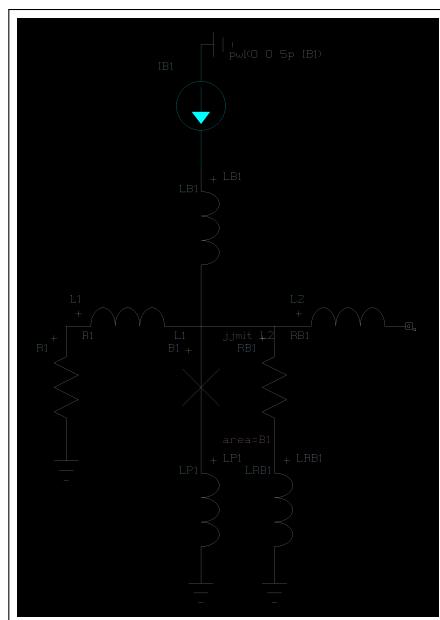


Figure 4.25: Schematic of RSFQ Always0 Asynchronous without an **a** input port.

Layout

The physical layouts of the two RSFQ Always0 Asynchronous cell versions are shown in Fig. 4.26a and 4.26b respectively. The height of the Always0 Asynchronous layout is $70 \mu m$ and the width is $40 \mu m$. For the version without the **a** input port, the width of the layout is $30 \mu m$. The cells include an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

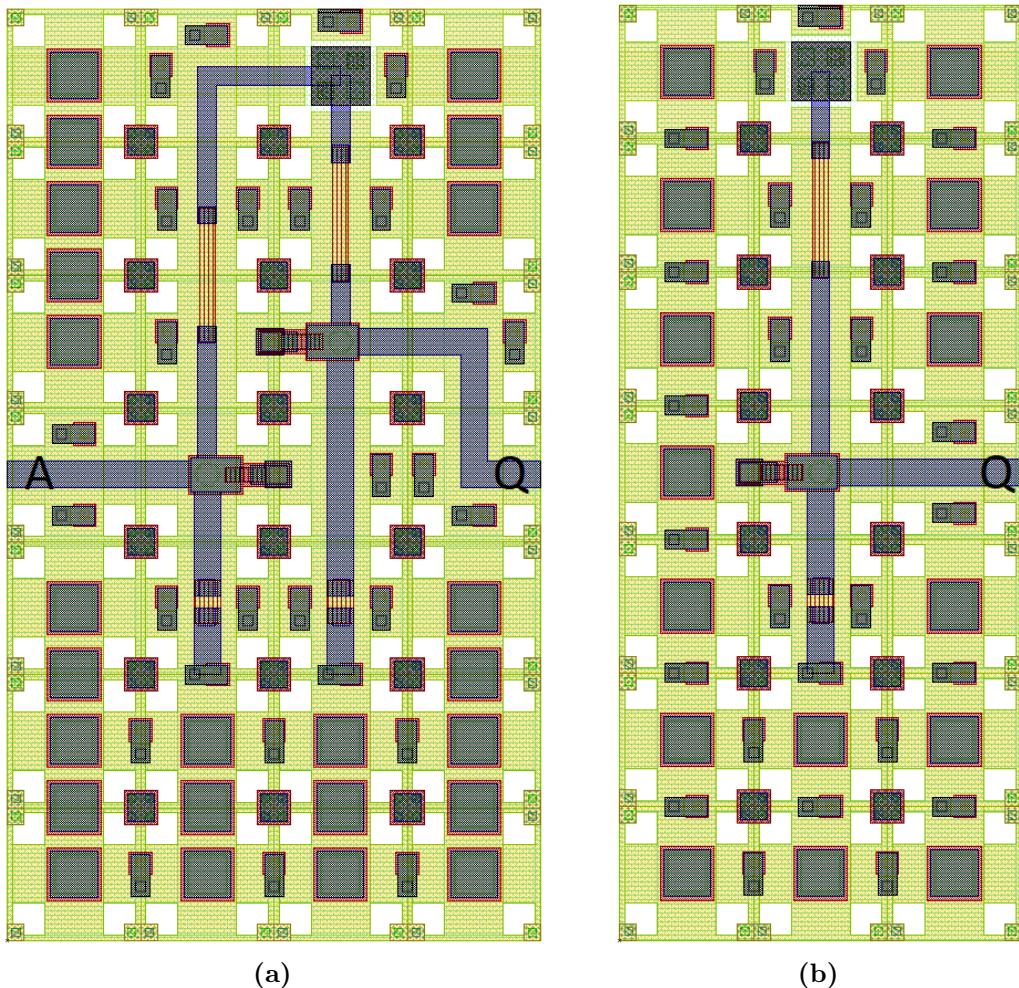


Figure 4.26: The physical layouts for (a) the RSFQ Always0 Asynchronous and (b) the RSFQ Always0 Asynchronous without an **a** input port.

Analog model

```

1  * Back-annotated simulation file written      31 | .param LB2=LB
2  *   ↪ by InductEx v.6.0.4 on 2-6-21.          32 | .param RB1=B0Rs/B1
3  * Author: L. Schindler                      33 | .param RB2=B0Rs/B2
4  * Version: 2.1                                34 | .param LRB1=(RB1/Rsheet)*Lsheet+LP
5  * Last modification date: 2 June 2021        35 | .param LRB2=(RB2/Rsheet)*Lsheet+LP
6  * Last modification by: L. Schindler          36 | .param LP1=LP
7  *$Ports                                         37 | .param LP2=LP
8  .subckt LSMITLL_Always0_async a q           38 | .param R1=2
9  .model jjmit jj(rtype=1, vg=2.8mV, cap     39 | .param R2=2
10 *=0.07pF, r0=160, rn=16, icrit=0.1mA       40 |
11 .param Phi0=2.067833848E-15                  41 | B1 1 2 jjmit area=B1
12 .param B0=1                                    42 | B2 6 7 jjmit area=B2
13 .param Ic0=0.0001                             43 | IB1 0 3 pwl(0 0 5p IB1)
14 .param IcRs=100u*6.859904418                 44 | IB2 0 8 pwl(0 0 5p IB2)
15 .param B0Rs=IcRs/Ic0*B0                      45 | L1 a 1 2.47E-12
16 .param Rsheet=2                              46 | L2 1 4 1.4E-12
17 .param Lsheet=1.13e-12                        47 | L3 5 6 2.948E-12
18 .param LP=0.2p                               48 | L4 6 q 3.768E-12
19 .param IC=2.5                                49 | R1 4 0 R1
20 .param Lptl=2p                               50 | R2 5 0 R2
21 .param BiasCoef=0.7                          51 | LB1 1 3 2.199E-12
22 .param B1=IC                                 52 | LB2 6 8 1.087E-12
23 .param B2=IC                                 53 | LP1 2 0 5.282E-13
24 .param IB1=B1*Ic0*BiasCoef                  54 | LP2 7 0 4.908E-13
25 .param IB2=B2*Ic0*BiasCoef                  55 | RB1 1 101 RB1
26 .param L1=Phi0/(4*B1*Ic0)                   56 | LRB1 101 0 LRB1
27 .param L2=Phi0/(2*B1*Ic0)                   57 | RB2 6 106 RB2
28 .param L3=Phi0/(2*B2*Ic0)                   58 | LRB2 106 0 LRB2
29 .param L4=Phi0/(4*B2*Ic0)                   59 | .ends
30 .param LB1=LB

```

Listing 4.11: RSFQ Always0 Asynchronous JoSIM netlist.

Table 4.11: RSFQ Always0 Asynchronous pin list.

Pin	Description
a	Data input
q	Data output

```

1  * Back-annotated simulation file written      22 | .param B1=IC
2  *   ↪ by InductEx v.6.0.4 on 2-6-21.          23 | .param IB1=B1*Ic0*BiasCoef
3  * Author: L. Schindler                      24 | .param L1=Phi0/(2*B1*Ic0)
4  * Version: 2.1                                25 | .param L2=Phi0/(4*B1*Ic0)
5  * Last modification date: 2 June 2021        26 | .param LB1=LB
6  * Last modification by: L. Schindler         27 | .param RB1=B0Rs/B1
7  *$Ports                         q           28 | .param LRB1=(RB1/Rsheet)*Lsheet+LP
8  .subckt LSMITLL_Always0_async_noA q          29 | .param LP1=LP
9  .model jjmit jj(rtype=1, vg=2.8mV, cap     30 | .param R1=2
10 .param Phi0=2.067833848E-15                  31 | B1 2 3 jjmit area=B1
11 .param B0=1                                    32 | IB1 0 5 pwl(0 0 5p IB1)
12 .param Ic0=0.0001                            33 | L1 1 2 1.397E-12
13 .param IcRs=100u*6.859904418                 34 | L2 2 q 2.465E-12
14 .param B0Rs=IcRs/Ic0*B0                     35 | R1 1 0 R1
15 .param Rsheet=2                             36 | LB1 2 4 3.177E-12
16 .param Lsheet=1.13e-12                      37 | LP1 3 0 5.306E-13
17 .param LP=0.2p                           38 | RB1 2 102 RB1
18 .param IC=2.5                           39 | LRB1 102 0 LRB1
19 .param Lptl=2p                          40 | .ends
20 .param LB=2p                           41 |
21 .param BiasCoef=0.7

```

Listing 4.12: RSFQ Always0 Asynchronous, without an **a** input port, JoSIM netlist.

Table 4.12: RSFQ Always0 Asynchronous, without an **a** input port, pin list.

Pin	Description
q	Data output

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 2 June 2021
5 // Last modification by: L. Schindler
6 'timescale 1ps/100fs
7 module LSmitll_Always0_async (a, q);
8
9 input
10   a;
11
12 output
13   q;
14
15 reg
16   q;
17
18 initial
19   begin
20     q = 0; // Output always 0
21   end
22
23 always
24   begin
25     #10 q = 0; // Output always 0
26   end
27 endmodule

```

Listing 4.13: RSFQ Always0 Asynchronous verilog model.

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 2 June 2021
5 // Last modification by: L. Schindler
6 'timescale 1ps/100fs
7 module LSmitll_Always0_async_noA (q);
8
9
10 output
11   q;
12
13 reg
14   q;
15
16 initial
17   begin
18     q = 0; // Output always 0
19   end
20
21 always
22   begin
23     #10 q = 0; // Output always 0
24   end
25 endmodule

```

Listing 4.14: RSFQ Always0 Asynchronous, without an **a** input port, verilog model.

Power consumption

Table 4.13: RSFQ Always0 Asynchronous power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	910	0.52
2	910	1.03
5	910	2.58
10	910	5.17
20	910	10.3
50	910	25.8

Table 4.14: RSFQ Always0 Asynchronous, without an **a** input port, power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	455	N/A
2	455	N/A
5	455	N/A
10	455	N/A
20	455	N/A
50	455	N/A

4.1.7 Always0 Synchronous

The Always0 Synchronous cell provides an output which is always zero synchronous to a clock signal. Two versions of the cell exist: One with an **a** input port, as seen in Fig. 4.27, and one without an **a** input port, as seen in Fig. 4.28. The Always0 Synchronous is not designed to be directly connected to a PTL.

Schematic

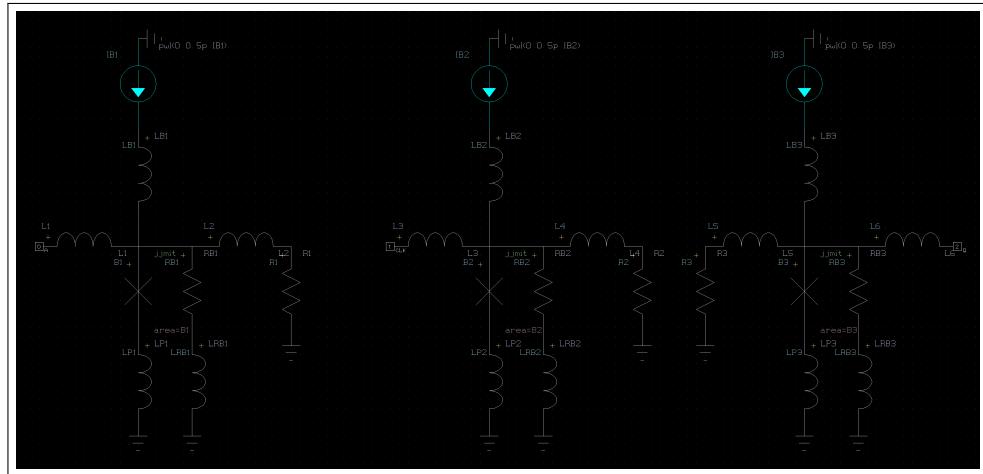


Figure 4.27: Schematic of RSFQ Always0 Synchronous.

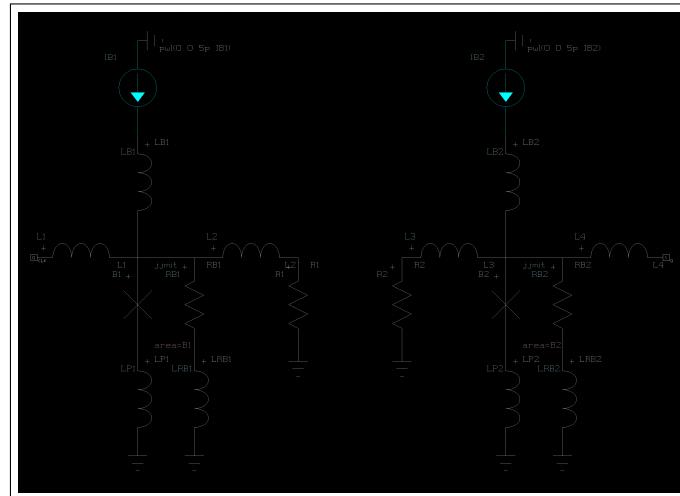


Figure 4.28: Schematic of RSFQ Always0 Synchronous without an **a** input port.

Layout

The physical layouts of the two RSFQ Always0 Synchronous cell versions are shown in Fig. 4.29a and 4.29b respectively. The height of the Always0 Synchronous layout is $70 \mu\text{m}$ and the width is $50 \mu\text{m}$. For the version without the **a** input port, the width of the layout is $30 \mu\text{m}$. The cells include an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

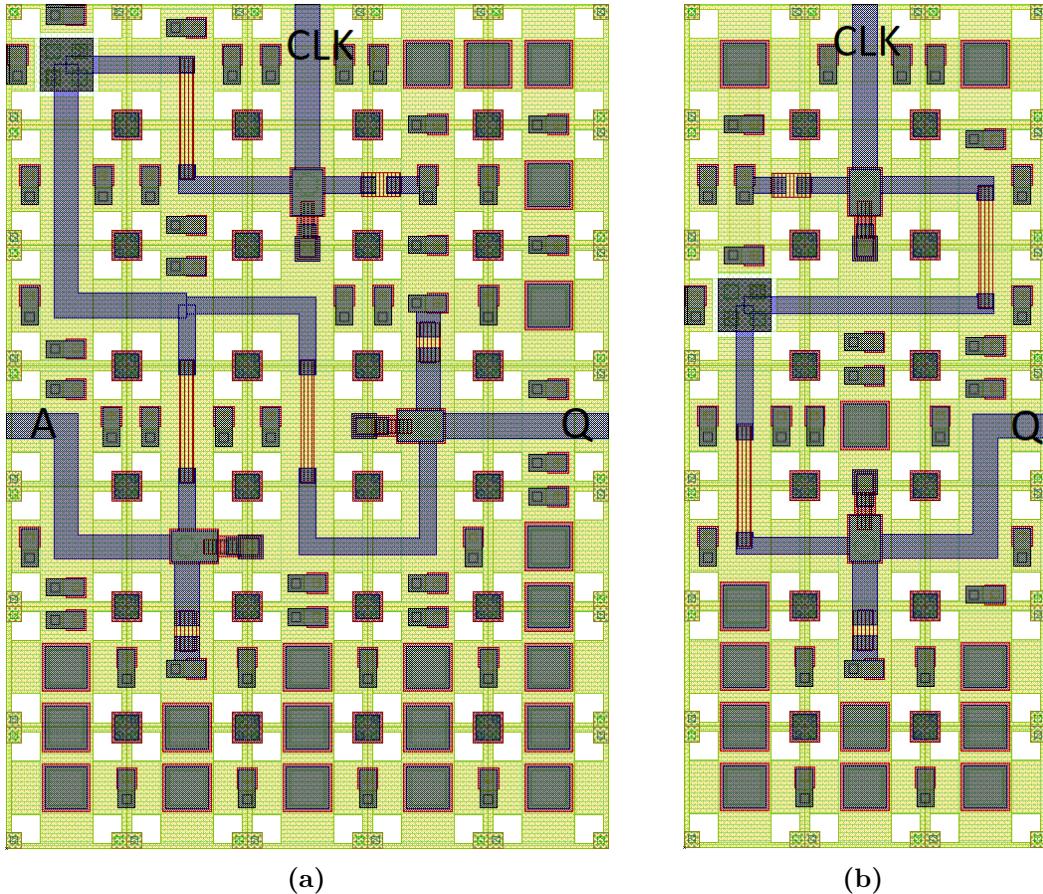


Figure 4.29: The physical layouts for (a) the RSFQ Always0 Synchronous and (b) the RSFQ Always0 Synchronous without an **a** input port.

Analog model

```

1  * Back-annotated simulation file written      40 | .param LRB1=(RB1/Rsheet)*Lsheet+LP
2  *   ↪ by InductEx v.6.0.4 on 2-6-21.          41 | .param LRB2=(RB2/Rsheet)*Lsheet+LP
3  * Author: L. Schindler                      42 | .param LRB3=(RB3/Rsheet)*Lsheet+LP
4  * Version: 2.1                                43 | .param LP1=LP
5  * Last modification date: 2 June 2021        44 | .param LP2=LP
6  * Last modification by: L. Schindler         45 | .param LP3=LP
7  *$Ports                                         a clk q    46 | .param R1=2
8  .subckt LSMITLL_Always0_sync a clk q        47 | .param R2=2
9  .model jjmit jj(rttype=1, vg=2.8mV, cap     48 | .param R3=2
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA      49 |
   ↪ )                                             
10 .param Phi0=2.067833848E-15                  50 | B1 1 2 jjmit area=B1
11 .param B0=1                                    51 | B2 5 6 jjmit area=B2
12 .param Ic0=0.0001                            52 | B3 10 11 jjmit area=B3
13 .param IcRs=100u*6.859904418                 53 | IB1 0 3 pw1(0 0 5p IB1)
14 .param B0Rs=IcRs/Ic0*B0                      54 | IB2 0 7 pw1(0 0 5p IB2)
15 .param Rsheet=2                             55 | IB3 0 12 pw1(0 0 5p IB3)
16 .param Lsheet=1.13e-12                        56 | L1 a 1 3.78E-12
17 .param LP=0.2p                               57 | L2 1 4 1.016E-12
18 .param IC=2.5                                58 | L3 clk 5 2.5E-12
19 .param Lptl=2p                               59 | L4 5 8 1.048E-12
20 .param LB=2p                                 60 | L5 9 10 1.005E-12
21 .param BiasCoef=0.7                         61 | L6 10 q 2.488E-12
22 .param B1=IC                                 62 | R1 4 0 R1
23 .param B2=IC                                 63 | R2 8 0 R2
24 .param B3=IC                                 64 | R3 9 0 R3
25 .param IB1=B1*Ic0*BiasCoef                 65 | LB1 1 3 1.243E-12
26 .param IB2=B2*Ic0*BiasCoef                 66 | LB2 5 7 2.181E-12
27 .param IB3=B3*Ic0*BiasCoef                 67 | LB3 10 12 5.092E-12
28 .param L1=Phi0/(4*B1*Ic0)                   68 | LP1 2 0 4.931E-13
29 .param L2=Phi0/(2*B1*Ic0)                   69 | LP2 6 0 4.919E-13
30 .param L3=Phi0/(4*B2*Ic0)                   70 | LP3 11 0 5.015E-13
31 .param L4=Phi0/(2*B2*Ic0)                   71 | RB1 1 101 RB1
32 .param L5=Phi0/(2*B3*Ic0)                   72 | LRB1 101 0 LRB1
33 .param L6=Phi0/(4*B3*Ic0)                   73 | RB2 5 105 RB2
34 .param LB1=LB                                74 | LRB2 105 0 LRB2
35 .param LB2=LB                                75 | RB3 10 110 RB3
36 .param LB3=LB                                76 | LRB3 110 0 LRB3
37 .param RB1=B0Rs/B1                          77 | .ends
38 .param RB2=B0Rs/B2
39 .param RB3=B0Rs/B3

```

Listing 4.15: RSFQ Always0 Synchronous JoSIM netlist.

Table 4.15: RSFQ Always0 Synchronous pin list.

Pin	Description
a	Data input
clk	Clock input
q	Data output

```

1  * Back-annotated simulation file written      31 | .param LB2=LB
2  *   ↪ by InductEx v.6.0.4 on 2-6-21.          32 | .param RB1=B0Rs/B1
3  * Author: L. Schindler                      33 | .param RB2=B0Rs/B2
4  * Version: 2.1                                34 | .param LRB1=(RB1/Rsheet)*Lsheet+LP
5  * Last modification date: 2 June 2021        35 | .param LRB2=(RB2/Rsheet)*Lsheet+LP
6  * Last modification by: L. Schindler         36 | .param LP1=LP
7  *$Ports                                         37 | .param LP2=LP
8  .subckt LSMITLL_Always0_sync_noA clk q      38 | .param R1=2
9  .model jjmit jj(rtype=1, vg=2.8mV, cap     39 | .param R2=2
10 |   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    40 |
11 |   ↪ )                                         41 | B1 1 2 jjmit area=B1
12 | .param Phi0=2.067833848E-15                 42 | B2 6 7 jjmit area=B2
13 | .param B0=1                                  43 | IB1 0 3 pw1(0 0 5p IB1)
14 | .param Ic0=0.0001                            44 | IB2 0 8 pw1(0 0 5p IB2)
15 | .param IcRs=100u*6.859904418                45 | L1 clk 1 2.507E-12
16 | .param B0Rs=IcRs/Ic0*B0                     46 | L2 1 4 1.057E-12
17 | .param Rsheet=2                             47 | L3 5 6 1.062E-12
18 | .param LP=0.2p                             48 | L4 6 q 3.747E-12
19 | .param IC=2.5                               49 | R1 4 0 R1
20 | .param Lptl=2p                             50 | R2 5 0 R2
21 | .param LB=2p                               51 | LB1 1 3 2.133E-12
22 | .param BiasCoef=0.7                         52 | LB2 6 8 2.17E-12
23 | .param B1=IC                               53 | LP1 2 0 4.927E-13
24 | .param B2=IC                               54 | LP2 7 0 4.925E-13
25 | .param IB1=B1*Ic0*BiasCoef               55 | RB1 1 101 RB1
26 | .param IB2=B2*Ic0*BiasCoef               56 | LRB1 101 0 LRB1
27 | .param L1=Phi0/(4*B1*Ic0)                  57 | RB2 6 106 RB2
28 | .param L2=Phi0/(2*B1*Ic0)                  58 | LRB2 106 0 LRB2
29 | .param L3=Phi0/(2*B2*Ic0)                  59 | .ends
30 | .param LB1=LB

```

Listing 4.16: RSFQ Always0 Synchronous, without an **a** input port, JoSIM netlist.

Table 4.16: RSFQ Always0 Synchronous, without an **a** input port, pin list.

Pin	Description
clk	Clock input
q	Data output

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 2 June 2021
5 // Last modification by: L. Schindler
6 //
7 `timescale 1ps/100fs
8 module LSmitll_Always0_sync (a, clk, q);
9
10 input
11   a;
12   clk;
13
14 output
15   q;
16
17 reg
18   q;
19
20 initial
21 begin
22   q = 0; // Output always 0
23 end
24
25 always
26 begin
27   #10 q = 0; // Output always 0
28 end
29 endmodule

```

Listing 4.17: RSFQ Always0 Synchronous verilog model.

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 2 June 2021
5 // Last modification by: L. Schindler
6 //
7 // Timing description and structural design for IARPA-BAA-14-03 via
8 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
9 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
10 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
11 // (c) 2016-2020 Stellenbosch University
12 //
13 `timescale 1ps/100fs
14 module LSmitll_Always0_sync_noA (clk, q);
15
16 input
17   clk;
18
19 output
20   q;
21
22 reg
23   q;
24
25 initial
26 begin
27   q = 0; // Output always 0
28 end
29
30 always
31 begin
32   #10 q = 0; // Output always 0
33 end
34 endmodule

```

Listing 4.18: RSFQ Always0 Synchronous, without an **a** input port, verilog model.

Power consumption

Table 4.17: RSFQ Always0 Synchronous power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	1365	1.03
2	1365	2.07
5	1365	5.17
10	1365	10.3
20	1365	20.7
50	1365	51.7

Table 4.18: RSFQ Always0 Synchronous, without an **a** input port, power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	910	0.52
2	910	1.03
5	910	2.58
10	910	5.17
20	910	10.3
50	910	25.8

4.2 Logic Cells

4.2.1 AND2

The RSFQ AND2 cell generates an output pulse if pulses from both input signal lines were received before the clock signal. The cell is not designed to be directly connected to a PTL.

Schematic

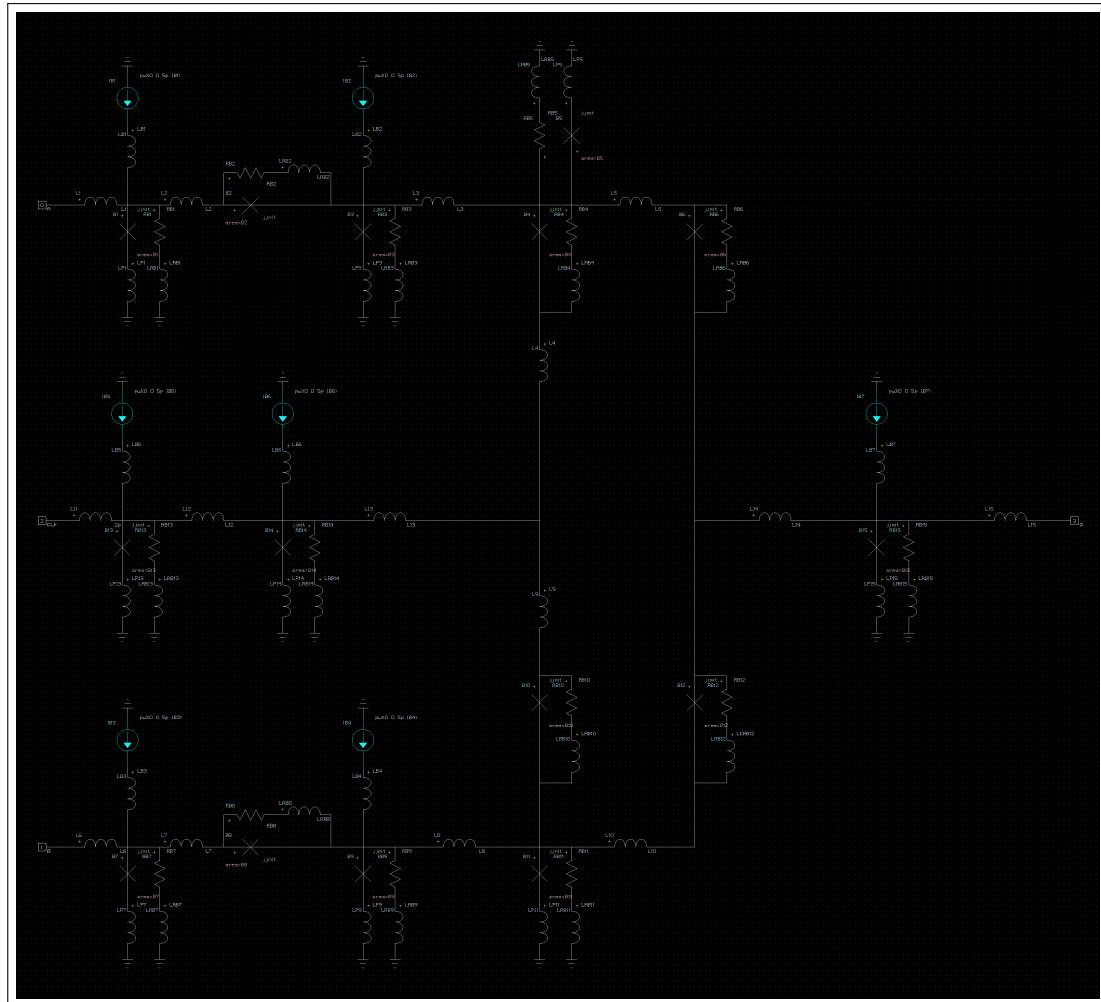


Figure 4.30: Schematic of RSFQ AND2.

Layout

The physical layout for the RSFQ AND2 is shown in Fig. 4.31. The layout height is $70 \mu\text{m}$ and the width is $90 \mu\text{m}$. The cell includes an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

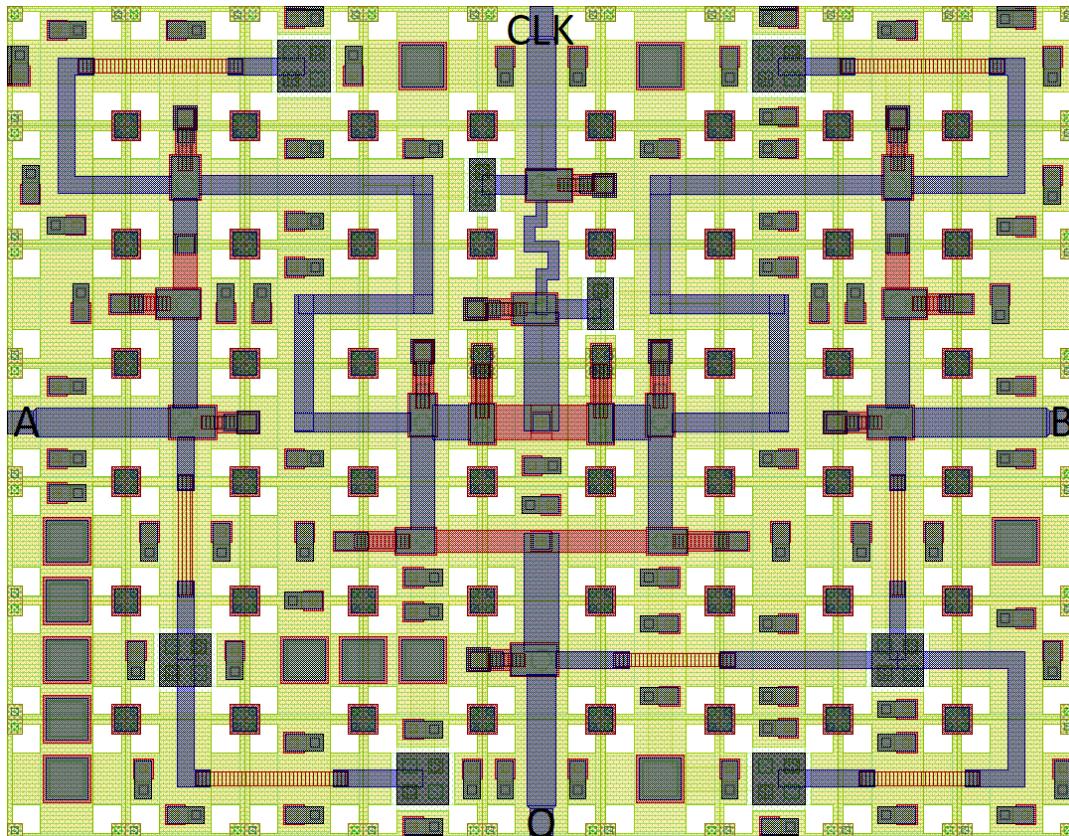


Figure 4.31: RSFQ AND2 Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param L10=L5
2   ↪ by InductEx v.6.0.4 on 25-3-21.          65  .param L11=Phi0/(4*IC*Ic0)
3  * Author: L. Schindler                      66  .param L12=Phi0/(2*B13*Ic0)
4  * Version: 2.1                               67  .param L13=1p
5  * Last modification date: 7 January 2021    68  .param L14=1p
6  * Last modification by: L. Schindler        69  .param L15=Phi0/(4*IC*Ic0)
7  *$Ports a b clk q                         70
8 .subckt LSMITLL_AND2 a b clk q            71  .param LP1=LP
9 .model jjmit jj(rtype=1, vg=2.8mV, cap    72  .param LP3=LP
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    73  .param LP5=LP
   ↪ )                                         74  .param LP7=LP
10 .param Phi0=2.067833848E-15               75  .param LP9=LP
11 .param B0=1                                76  .param LP11=LP
12 .param Ic0=0.0001                          77  .param LP13=LP
13 .param IcRs=100u*6.859904418              78  .param LP14=LP
14 .param B0Rs=IcRs/Ic0*B0                   79  .param LP15=LP
15 .param Rsheet=2                            80
16 .param Lsheet=1.13e-12                     81  .param RB1=B0Rs/B1
17 .param LP=0.2p                            82  .param RB2=B0Rs/B2
18 .param IC=2.5                            83  .param RB3=B0Rs/B3
19 .param Lptl=2p                           84  .param RB4=B0Rs/B4
20 .param LB=2p                             85  .param RB5=B0Rs/B5
21 .param BiasCoef=0.7                      86  .param RB6=B0Rs/B6
22
23 .param B1=IC                             87  .param RB7=B0Rs/B7
24 .param B2=2.01                           88  .param RB8=B0Rs/B8
25 .param B3=1.91                           89  .param RB9=B0Rs/B9
26 .param B4=1.26                           90  .param RB10=B0Rs/B10
27 .param B5=1.57                           91  .param RB11=B0Rs/B11
28 .param B6=1.19                           92  .param RB12=B0Rs/B12
29 .param B7=B1                            93  .param RB13=B0Rs/B13
30 .param B8=B2                            94  .param RB14=B0Rs/B14
31 .param B9=B3                            95  .param RB15=B0Rs/B15
32 .param B10=B4                           96  .param LRB1=(RB1/Rsheet)*Lsheet
33 .param B11=B5                           97  .param LRB2=(RB2/Rsheet)*Lsheet
34 .param B12=B6                           98  .param LRB3=(RB3/Rsheet)*Lsheet
35 .param B13=IC                           99  .param LRB4=(RB4/Rsheet)*Lsheet
36 .param B14=2.06                           100 .param LRB5=(RB5/Rsheet)*Lsheet
37 .param B15=IC                           101 .param LRB6=(RB6/Rsheet)*Lsheet
38
39 .param IB1=BiasCoef*Ic0*B1             102 .param LRB7=(RB7/Rsheet)*Lsheet
40 .param IB2=123u                           103 .param LRB8=(RB8/Rsheet)*Lsheet
41 .param IB3=IB1                           104 .param LRB9=(RB9/Rsheet)*Lsheet
42 .param IB4=IB2                           105 .param LRB10=(RB10/Rsheet)*Lsheet
43 .param IB5=BiasCoef*Ic0*B13            106 .param LRB11=(RB11/Rsheet)*Lsheet
44 .param IB6=133u                           107 .param LRB12=(RB12/Rsheet)*Lsheet
45 .param IB7=BiasCoef*Ic0*B15            108 .param LRB13=(RB13/Rsheet)*Lsheet
46
47 .param LB1=LB                           109 .param LRB14=(RB14/Rsheet)*Lsheet
48 .param LB2=LB                           110 .param LRB15=(RB15/Rsheet)*Lsheet
49 .param LB3=LB                           111
50 .param LB4=LB                           112 B1 1 2 jjmit area=B1
51 .param LB5=LB                           113 B2 4 5 jjmit area=B2
52 .param LB6=LB                           114 B3 5 6 jjmit area=B3
53 .param LB7=LB                           115 B4 8 9 jjmit area=B4
54
55 .param L1=Phi0/(4*IC*Ic0)             116 B5 8 11 jjmit area=B5
56 .param L2=Phi0/(2*B1*Ic0)             117 B6 12 13 jjmit area=B6
57 .param L3=Phi0/(B3*Ic0)              118 B7 14 15 jjmit area=B7
58 .param L4=1p                            119 B8 17 18 jjmit area=B8
59 .param L5=Phi0/(2*B5*Ic0)             120 B9 18 19 jjmit area=B9
60 .param L6=L1                           121 B10 21 22 jjmit area=B10
61 .param L7=L2                           122 B11 21 23 jjmit area=B11
62 .param L8=L3                           123 B12 24 13 jjmit area=B12
63 .param L9=L4                           124 B13 25 26 jjmit area=B13
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

```

```

130 | IB3 0 16 pwl(0 0 5p IB3)      167 | L14 13 28 1.06E-12
131 | IB4 0 20 pwl(0 0 5p IB4)      168 | L15 28 q 2.069E-12
132 | IB5 0 27 pwl(0 0 5p IB5)      169
133 | IB6 0 33 pwl(0 0 5p IB6)      170 | RB1 1 101 RB1
134 | IB7 0 30 pwl(0 0 5p IB7)      171 | LRB1 101 0 LRB1
135
136 | LB1 3 1 LB1                   172 | RB2 4 104 RB2
137 | LB2 7 5 LB2                   173 | LRB2 104 5 LRB2
138 | LB3 16 14 LB3                  174 | RB3 5 105 RB3
139 | LB4 20 18 LB4                  175 | LRB3 105 0 LRB3
140 | LB5 27 25 LB5                  176 | RB4 8 109 RB4
141 | LB6 30 28 LB6                  177 | LRB4 109 9 LRB4
142 | LB7 33 31 LB7                  178 | RB5 8 108 RB5
143
144 | LP1 2 0 4.864E-13             179 | LRB5 108 0 LRB5
145 | LP3 6 0 5.474E-13             180 | RB6 12 112 RB6
146 | LP5 11 0 5.279E-13            181 | LRB6 112 13 LRB6
147 | LP7 15 0 4.901E-13            182 | RB7 14 114 RB7
148 | LP9 19 0 5.414E-13            183 | LRB7 114 0 LRB7
149 | LP11 23 0 5.306E-13           184 | RB8 17 117 RB8
150 | LP13 26 0 5.084E-13           185 | LRB8 117 18 LRB8
151 | LP14 32 0 5.329E-13           186 | RB9 18 118 RB9
152 | LP15 29 0 4.92E-13            187 | LRB9 118 0 LRB9
153
154 | L1 a 1 2.075E-12              188 | RB10 21 122 RB10
155 | L2 1 4 2.812E-12              189 | LRB10 122 22 LRB10
156 | L3 5 8 9.756E-12              190 | RB11 21 121 RB11
157 | L4 9 10 1.079E-12             191 | LRB11 121 0 LRB11
158 | L5 8 12 3.105E-12             192 | RB12 24 124 RB12
159 | L6 b 14 2.073E-12             193 | LRB12 124 13 LRB12
160 | L7 14 17 2.811E-12             194 | RB13 25 125 RB13
161 | L8 18 21 9.768E-12             195 | LRB13 125 0 LRB13
162 | L9 22 10 1.084E-12             196 | RB14 31 131 RB14
163 | L10 21 24 3.095E-12            197 | LRB14 131 0 LRB14
164 | L11 clk 25 2.063E-12            198 | RB15 28 128 RB15
165 | L12 25 31 2.96E-12             199 | LRB15 128 0 LRB15
166 | L13 31 10 1.002E-12            200 | .ends

```

Listing 4.19: RSFQ AND2 JoSIM netlist.**Table 4.19:** RSFQ AND2 pin list.

Pin	Description
a	Data input
b	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ AND2 using JoSIM is shown in Fig. 4.32. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b**,
- (d) the phase over the input JJ of pin **b**,
- (e) the current through the input inductor connected to pin **clk**,
- (f) the phase over the input JJ of pin **clk**,
- (g) the phase over the output JJ of pin **q**, and
- (h) the phase over the input JJ of the load circuit connected to pin **q**.

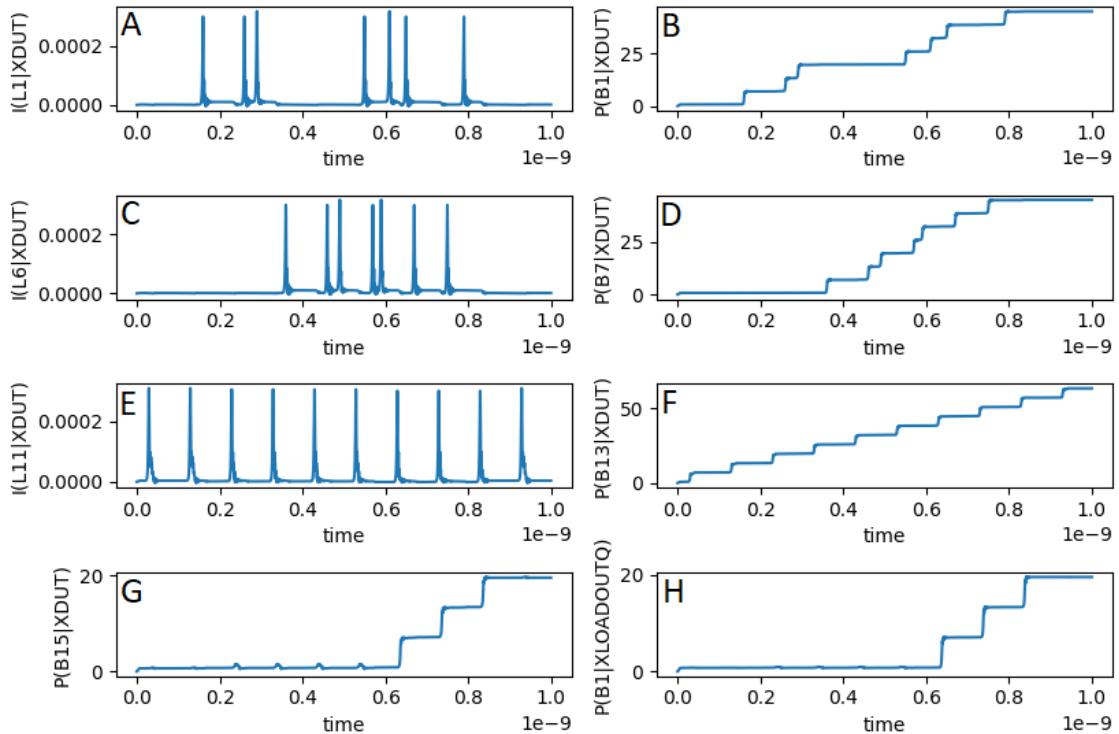


Figure 4.32: RSFQ AND2 analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 2 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 // -----
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 // -----
15 'timescale 1ps/100fs
16 module LSmitll_AND2 (a, b, clk, q);
17
18 input
19   a, b, clk;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state3_clk_q = 7.0,
29   ct_state0_clk_a = 2.2,
30   ct_state0_clk_b = 2.2,
31   ct_state1_clk_a = 1.6,
32   ct_state1_clk_b = 1.8,
33   ct_state2_clk_a = 1.8,
34   ct_state2_clk_b = 1.6,
35   ct_state3_clk_a = 0.7,
36   ct_state3_clk_b = 0.7;
37
38 reg
39   errorsignal_a,
40   errorsignal_b,
41   errorsignal_clk;
42
43 integer
44   outfile,
45   cell_state; // internal state of the cell
46
47 initial
48 begin
49   errorsignal_a = 0;
50   errorsignal_b = 0;
51   errorsignal_clk = 0;
52   cell_state = 0; // Startup state
53   q = 0; // All outputs start at 0
54 end
55
56 always @ (posedge a or negedge a) // execute at positive and negative edges of input
57 begin
58   if ($time > 4) // arbitrary steady-state time)
59   begin
60     if (errorsignal_a == 1'b1) // A critical timing is active for this input
61     begin
62       outfile = $fopen("errors.txt", "a");
63       $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
64       ↪ ", $stime);
65       $fclose(outfile);
66       q <= 1'bX; // Set all outputs to unknown
67     end

```

```

67         if (errorsignal_a == 0)
68             begin
69                 case (cell_state)
70                     0: begin
71                         cell_state = 1; // Blocking statement -- immediately
72                     end
73                     1: begin
74                         end
75                     2: begin
76                         cell_state = 3; // Blocking statement -- immediately
77                     end
78                     3: begin
79                         end
80                 endcase
81             end
82         end
83     end
84
85 always @(posedge b or negedge b) // execute at positive and negative edges of input
86 begin
87     if ($time>4) // arbitrary steady-state time)
88     begin
89         if (errorsignal_b == 1'b1) // A critical timing is active for this input
90             begin
91                 outfile = $fopen("errors.txt", "a");
92                 $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
93                             ↪ ", $stime);
94                 $fclose(outfile);
95                 q <= 1'bX; // Set all outputs to unknown
96             end
97         if (errorsignal_b == 0)
98             begin
99                 case (cell_state)
100                     0: begin
101                         cell_state = 2; // Blocking statement -- immediately
102                     end
103                     1: begin
104                         cell_state = 3; // Blocking statement -- immediately
105                     end
106                     2: begin
107                         end
108                     3: begin
109                         end
110                     endcase
111             end
112         end
113     end
114 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
115 begin
116     if ($time>4) // arbitrary steady-state time)
117     begin
118         if (errorsignal_clk == 1'b1) // A critical timing is active for this input
119             begin
120                 outfile = $fopen("errors.txt", "a");
121                 $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
122                             ↪ ", $stime);
123                 $fclose(outfile);
124                 q <= 1'bX; // Set all outputs to unknown
125             end
126         if (errorsignal_clk == 0)
127             begin
128                 case (cell_state)
129                     0: begin
130                         errorsignal_a = 1; // Critical timing on this input; assign
131                             ↪ immediately
130                         errorsignal_a <= #(ct_state0_clk_a) 0; // Clear error signal
131                             ↪ after critical timing expires
131                         errorsignal_b = 1; // Critical timing on this input; assign
131                             ↪ immediately

```

```

132           errorsignal_b <= #(ct_state0_clk_b) 0; // Clear error signal
133           // after critical timing expires
134       end
135   1: begin
136       cell_state = 0; // Blocking statement -- immediately
137       errorsignal_a = 1; // Critical timing on this input; assign
138           // immediately
139       errorsignal_a <= #(ct_state1_clk_a) 0; // Clear error signal
140           // after critical timing expires
141       errorsignal_b = 1; // Critical timing on this input; assign
142           // immediately
143       errorsignal_b <= #(ct_state1_clk_b) 0; // Clear error signal
144           // after critical timing expires
145   end
146   2: begin
147       cell_state = 0; // Blocking statement -- immediately
148       errorsignal_a = 1; // Critical timing on this input; assign
149           // immediately
150       errorsignal_a <= #(ct_state2_clk_a) 0; // Clear error signal
151           // after critical timing expires
152       errorsignal_b = 1; // Critical timing on this input; assign
153           // immediately
154       errorsignal_b <= #(ct_state2_clk_b) 0; // Clear error signal
155           // after critical timing expires
156   end
157   endcase
158 end
159
160
161 endmodule

```

Listing 4.20: RSFQ AND2 verilog model.

The digital simulation results for the RSFQ AND2 is shown in Fig. 4.33 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 4.34.

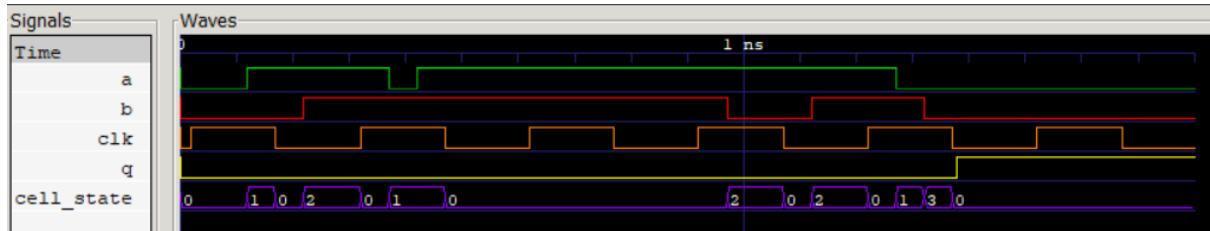


Figure 4.33: RSFQ AND2 digital simulation results.

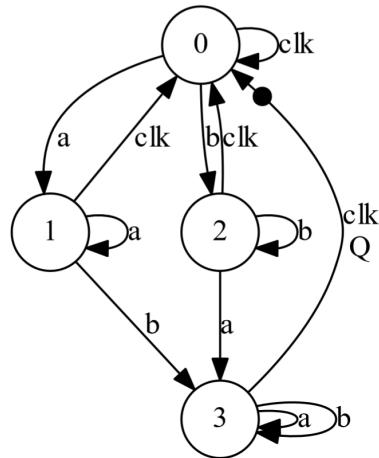


Figure 4.34: RSFQ AND2 Mealy finite state machine diagram.

Power Consumption

Table 4.20: RSFQ AND2 power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2805	5.78
2	2805	11.6
5	2805	28.9
10	2805	57.8
20	2805	116
50	2805	289

4.2.2 OR2

The RSFQ OR2 cell generates an output pulse if an input pulse from either input lines was received before the clock signal. The OR2 cell is not designed to be directly connected to a PTL.

Schematic

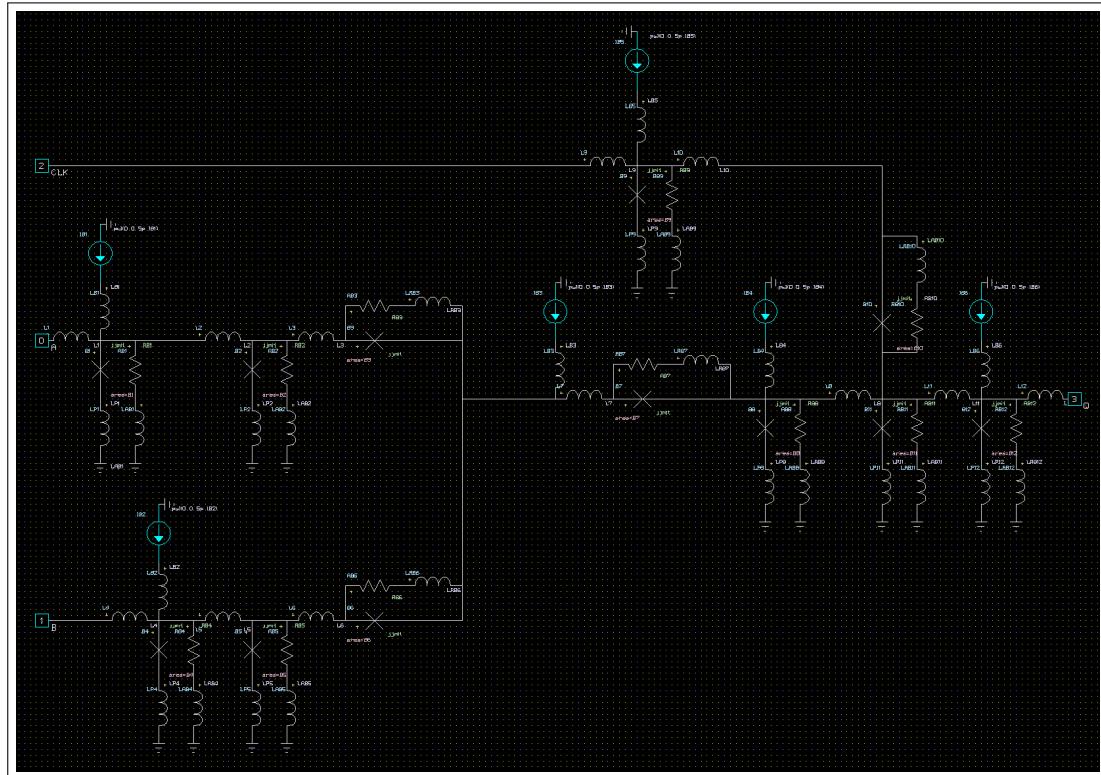


Figure 4.35: Schematic of RSFQ OR2.

Layout

The physical layout for the RSFQ OR2 is shown in Fig. 4.36. The layout height is $70 \mu\text{m}$ and the width is $80 \mu\text{m}$. The cell includes an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

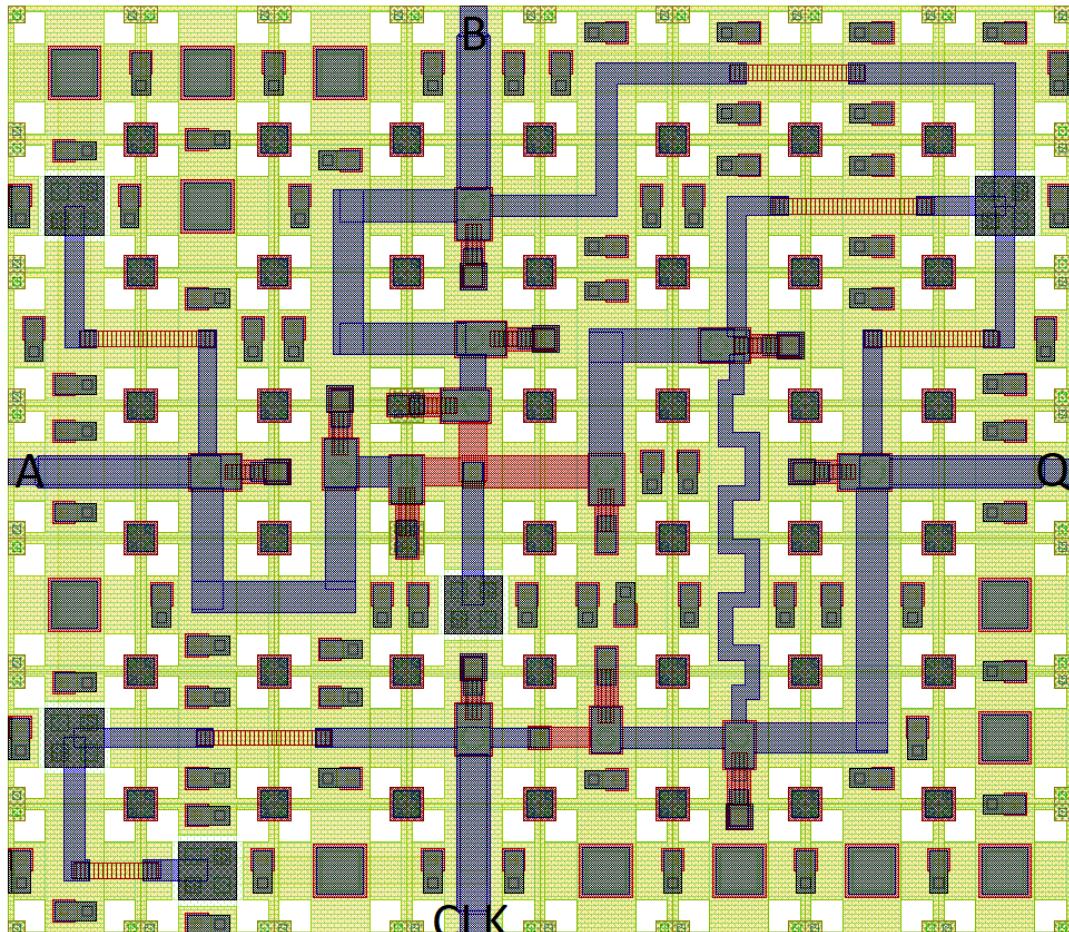


Figure 4.36: RSFQ OR2 Layout

Analog model

```

1  * Back-annotated simulation file written      64  .param RB9=B0Rs/B9
2   ↪ by InductEx v.6.0.4 on 29-3-21.          65  .param RB10=B0Rs/B10
3  * Author: L. Schindler                      66  .param RB11=B0Rs/B11
4  * Version: 2.1                               67  .param RB12=B0Rs/B12
5  * Last modification date: 3 June 2021       68  .param LRB1=(RB1/Rsheet)*Lsheet+LP
6  * Last modification by: L. Schindler        69  .param LRB2=(RB2/Rsheet)*Lsheet+LP
7  *$Ports a b clk q                         70  .param LRB3=(RB3/Rsheet)*Lsheet+LP
8  .SUBCKT LSMITLL_OR2 a b clk q             71  .param LRB4=(RB4/Rsheet)*Lsheet+LP
9   .model jjmit jj(rtype=1, vg=2.8mV, cap    72  .param LRB5=(RB5/Rsheet)*Lsheet+LP
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA     73  .param LRB6=(RB6/Rsheet)*Lsheet+LP
   ↪ )                                         74  .param LRB7=(RB7/Rsheet)*Lsheet+LP
10 .param Phi0=2.067833848E-15                75  .param LRB8=(RB8/Rsheet)*Lsheet+LP
11 .param B0=1                                 76  .param LRB9=(RB9/Rsheet)*Lsheet+LP
12 .param Ic0=0.0001                           77  .param LRB10=(RB10/Rsheet)*Lsheet+LP
13 .param IcRs=100u*6.859904418               78  .param LRB11=(RB11/Rsheet)*Lsheet+LP
14 .param B0Rs=IcRs/Ic0*B0                   79  .param LRB12=(RB12/Rsheet)*Lsheet+LP
15 .param Rsheet=2                            80
16 .param Lsheet=1.13e-12                     81 L1 a 1 2.051E-12
17 .param LP=0.2p                            82 L2 1 4 3.232E-12
18 .param IC=2.5                            83 L3 4 6 1.198E-12
19 .param Lptl=2p                           84 L4 b 8 2.046E-12
20 .param LB=2p                             85 L5 8 11 3.242E-12
21 .param BiasCoef=0.7                      86 L6 11 13 1.191E-12
22                               87 L7 7 15 3.341E-12
23 .param B1=2.5                            88 L8 16 19 7.998E-12
24 .param B2=2.22                           89 L9 clk 20 2.062E-12
25 .param B3=1.86                           90 L10 20 23 3.408E-12
26 .param B4=B1                            91 L11 19 25 3.54E-12
27 .param B5=B2                            92 L12 25 q 2.05E-12
28 .param B6=B3                            93
29 .param B7=2.28                           94 B1 1 2 jjmit area=B1
30 .param B8=2.09                           95 B2 4 5 jjmit area=B2
31 .param B9=2.5                            96 B3 6 7 jjmit area=B3
32 .param B10=1.52                           97 B4 8 9 jjmit area=B4
33 .param B11=1.60                           98 B5 11 12 jjmit area=B5
34 .param B12=2.5                           99 B6 13 7 jjmit area=B6
35                               100 B7 15 16 jjmit area=B7
36 .param IB1=175u                           101 B8 16 17 jjmit area=B8
37 .param IB2=IB1                           102 B9 20 21 jjmit area=B9
38 .param IB3=304u                           103 B10 23 19 jjmit area=B10
39 .param IB4=142u                           104 B11 19 24 jjmit area=B11
40 .param IB5=175u                           105 B12 25 26 jjmit area=B12
41 .param IB6=175u                           106
42                               107 IB1 0 3 pw1(0 0 5p IB1)
43 .param L1=Phi0/(4*IC*Ic0)                 108 IB2 0 10 pw1(0 0 5p IB2)
44 .param L2=Phi0/(2*B1*Ic0)                 109 IB3 0 14 pw1(0 0 5p IB3)
45 .param L3=1p                             110 IB4 0 18 pw1(0 0 5p IB4)
46 .param L4=L1                            111 IB5 0 22 pw1(0 0 5p IB5)
47 .param L5=L2                            112 IB6 0 27 pw1(0 0 5p IB6)
48 .param L6=L3                           113
49 .param L7=Phi0/(2*B2*Ic0)                 114 LB1 1 3 LB
50 .param L8=Phi0/(B8*Ic0)                  115 LB2 8 10 LB
51 .param L9=Phi0/(4*IC*Ic0)                 116 LB3 7 14 LB
52 .param L10=Phi0/(2*B9*Ic0)                117 LB4 16 18 LB
53 .param L11=Phi0/(2*B11*Ic0)               118 LB5 20 22 LB
54 .param L12=Phi0/(4*IC*Ic0)                 119 LB6 25 27 LB
55                               120
56 .param RB1=B0Rs/B1                      121 LP1 2 0 5.253E-13
57 .param RB2=B0Rs/B2                      122 LP2 5 0 5.141E-13
58 .param RB3=B0Rs/B3                      123 LP4 9 0 5.352E-13
59 .param RB4=B0Rs/B4                      124 LP5 12 0 5.154E-13
60 .param RB5=B0Rs/B5                      125 LP8 17 0 4.905E-13
61 .param RB6=B0Rs/B6                      126 LP9 21 0 5.216E-13
62 .param RB7=B0Rs/B7                      127 LP11 24 0 5.649E-13
63 .param RB8=B0Rs/B8                      128 LP12 26 0 5.28E-13
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

```

```

130 | RB1 1 101 RB1
131 | LRB1 101 0 LRB1
132 | RB2 4 104 RB2
133 | LRB2 104 0 LRB2
134 | RB3 6 105 RB3
135 | LRB3 105 7 LRB3
136 | RB4 8 108 RB4
137 | LRB4 108 0 LRB4
138 | RB5 11 111 RB5
139 | LRB5 111 0 LRB5
140 | RB6 13 113 RB6
141 | LRB6 113 7 LRB6
142 | RB7 15 115 RB7
143 | LRB7 115 16 LRB7
144 | RB8 16 116 RB8
145 | LRB8 116 0 LRB8
146 | RB9 20 120 RB9
147 | LRB9 120 0 LRB9
148 | RB10 23 123 RB10
149 | LRB10 123 19 LRB10
150 | RB11 19 119 RB11
151 | LRB11 119 0 LRB11
152 | RB12 25 125 RB12
153 | LRB12 125 0 LRB12
154 | .ends

```

Listing 4.21: RSFQ OR2 JoSIM netlist.**Table 4.21:** RSFQ OR2 pin list.

Pin	Description
a	Data input
b	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ OR2 using JoSIM is shown in Fig. 4.37. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b**,
- (d) the phase over the input JJ of pin **b**,
- (e) the current through the input inductor connected to pin **clk**,
- (f) the phase over the input JJ of pin **clk**,
- (g) the phase over the output JJ of pin **q**, and
- (h) the phase over the input JJ of the load circuit connected to pin **q**.

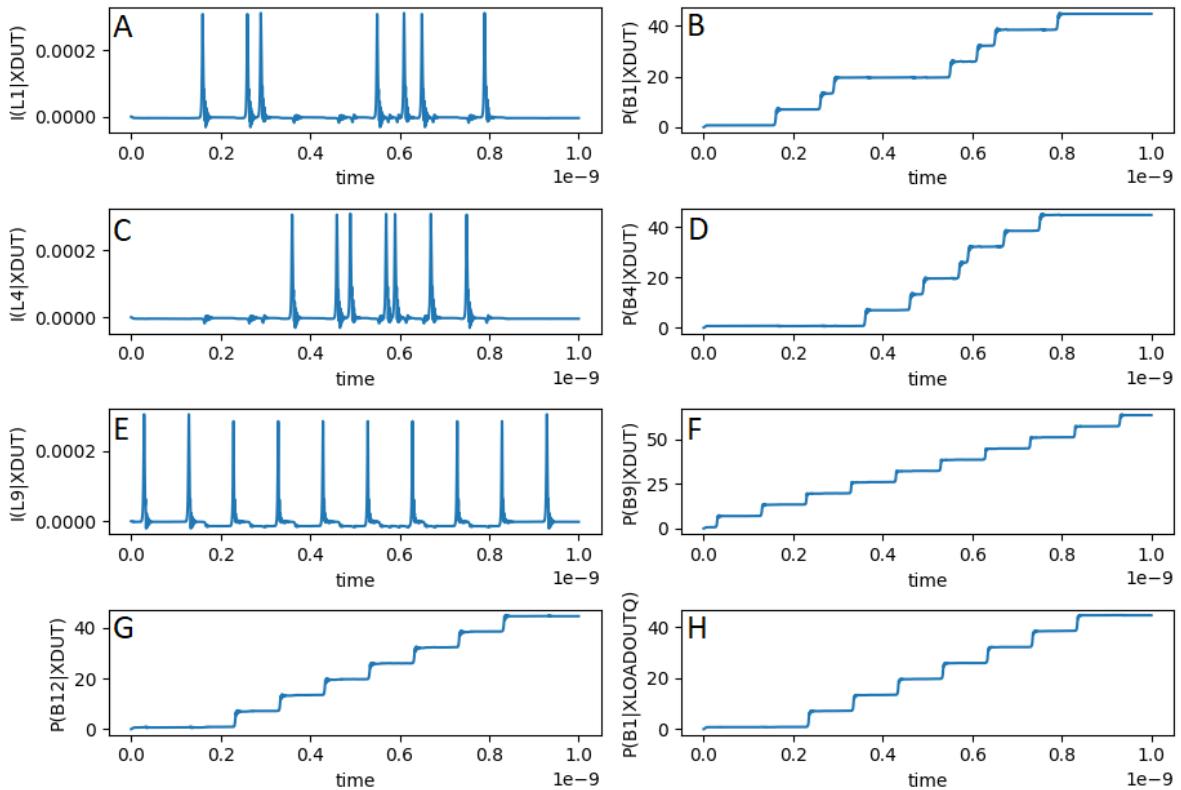


Figure 4.37: RSFQ OR2 analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 3 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 'timescale 1ps/100fs
8 module LSmitll_OR2_v2p1_optimized (a, b, clk, q);
9
10 input
11   a, b, clk;
12
13 output
14   q;
15
16 reg
17   q;
18
19 real
20   delay_state1_clk_q = 5.5,
21   ct_state0_a_clk = 3.2,
22   ct_state0_b_clk = 3.2,
23   ct_state1_a_clk = 5.6,
24   ct_state1_b_clk = 5.6;
25
26 reg
27   errorsignal_a,
28   errorsignal_b,
29   errorsignal_clk;
30
31 integer
32   outfile,
33   cell_state; // internal state of the cell
34
35 initial
36 begin
37   errorsignal_a = 0;
38   errorsignal_b = 0;
39   errorsignal_clk = 0;
40   cell_state = 0; // Startup state
41   q = 0; // All outputs start at 0
42 end
43 always @ (posedge a or negedge a) // execute at positive and negative edges of input
44 begin
45   if ($time > 4) // arbitrary steady-state time)
46     begin
47       if (errorsignal_a == 1'b1) // A critical timing is active for this input
48         begin
49           outfile = $fopen("errors.txt", "a");
50           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
51           ↪ ", $stime);
52           $fclose(outfile);
53           q <= 1'bX; // Set all outputs to unknown
54         end
55       if (errorsignal_a == 0)
56         begin
57           case (cell_state)
58             0: begin
59               cell_state = 1; // Blocking statement -- immediately
60               errorsignal_clk = 1; // Critical timing on this input; assign
61               ↪ immediately
62               errorsignal_clk <= #(ct_state0_a_clk) 0; // Clear error signal
63               ↪ after critical timing expires
64             end
65           1: begin
66               errorsignal_clk = 1; // Critical timing on this input; assign
67               ↪ immediately
68             end
69           endcase
70         end
71       end
72     end
73   end
74 end
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
737
738
739
739
740
741
742
743
744
745
745
746
747
748
749
749
750
751
752
753
754
755
756
756
757
758
759
759
760
761
762
763
764
765
765
766
767
768
768
769
769
770
771
772
773
774
775
775
776
777
778
778
779
779
780
781
782
783
783
784
785
785
786
787
787
788
788
789
789
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
157
```

```

64           errorsignal_clk <= #(ct_state1_a_clk) 0; // Clear error signal
65           // after critical timing expires
66       end
67   endcase
68 end
69
70
71 always @(posedge b or negedge b) // execute at positive and negative edges of input
72 begin
73     if ($time>4) // arbitrary steady-state time)
74     begin
75         if (errorsignal_b == 1'b1) // A critical timing is active for this input
76         begin
77             outfile = $fopen("errors.txt", "a");
78             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n
79             ", $stime);
80             $fclose(outfile);
81             q <= 1'bX; // Set all outputs to unknown
82         end
83         if (errorsignal_b == 0)
84         begin
85             case (cell_state)
86             0: begin
87                 cell_state = 1; // Blocking statement -- immediately
88                 errorsignal_clk = 1; // Critical timing on this input; assign
89                 // immediately
90                 errorsignal_clk <= #(ct_state0_b_clk) 0; // Clear error signal
91                 // after critical timing expires
92             end
93         end
94     endcase
95 end
96
97 end
98
99 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
100 begin
101     if ($time>4) // arbitrary steady-state time)
102     begin
103         if (errorsignal_clk == 1'b1) // A critical timing is active for this input
104         begin
105             outfile = $fopen("errors.txt", "a");
106             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n
107             ", $stime);
108             $fclose(outfile);
109             q <= 1'bX; // Set all outputs to unknown
110         end
111         if (errorsignal_clk == 0)
112         begin
113             case (cell_state)
114             0: begin
115                 end
116             1: begin
117                 q <= #(delay_state1_clk_q) !q;
118                 cell_state = 0; // Blocking statement -- immediately
119             end
120         endcase
121     end
122 end
123 endmodule

```

Listing 4.22: RSFQ OR2 verilog model.

The digital simulation results for the RSFQ OR2 is shown in Fig. 4.38 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 4.39.

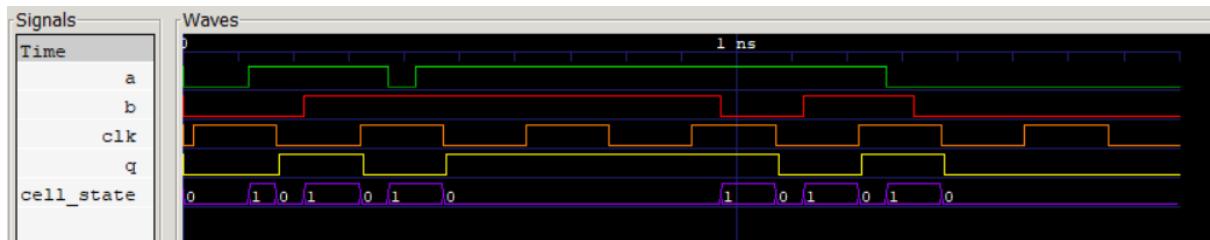


Figure 4.38: RSFQ OR2 digital simulation results.

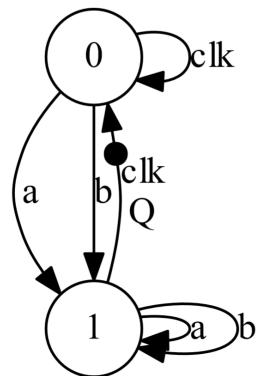


Figure 4.39: RSFQ OR2 Mealy finite state machine diagram.

Power Consumption

Table 4.22: RSFQ OR2 power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2980	5.30
2	2980	10.6
5	2980	26.5
10	2980	53.0
20	2980	106
50	2980	265

4.2.3 XOR

The RSFQ XOR cell generates an output pulse exclusively if a pulse from a single input line was received before the clock signal. The XOR cell is not designed to be directly connected to a PTL.

Schematic

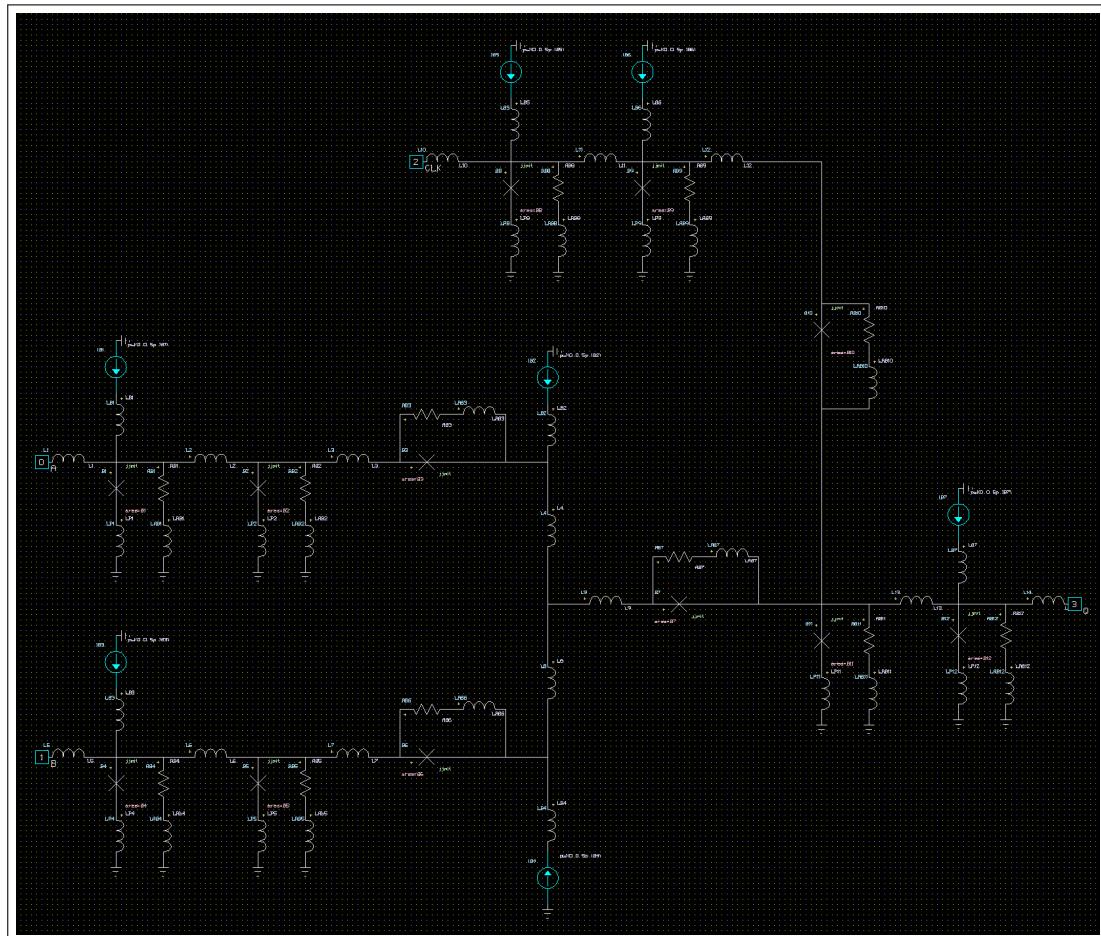


Figure 4.40: Schematic of RSFQ XOR.

Layout

The physical layout for the RSFQ XOR is shown in Fig. 4.41. The layout height is $70 \mu\text{m}$ and the width is $70 \mu\text{m}$. The cell includes an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

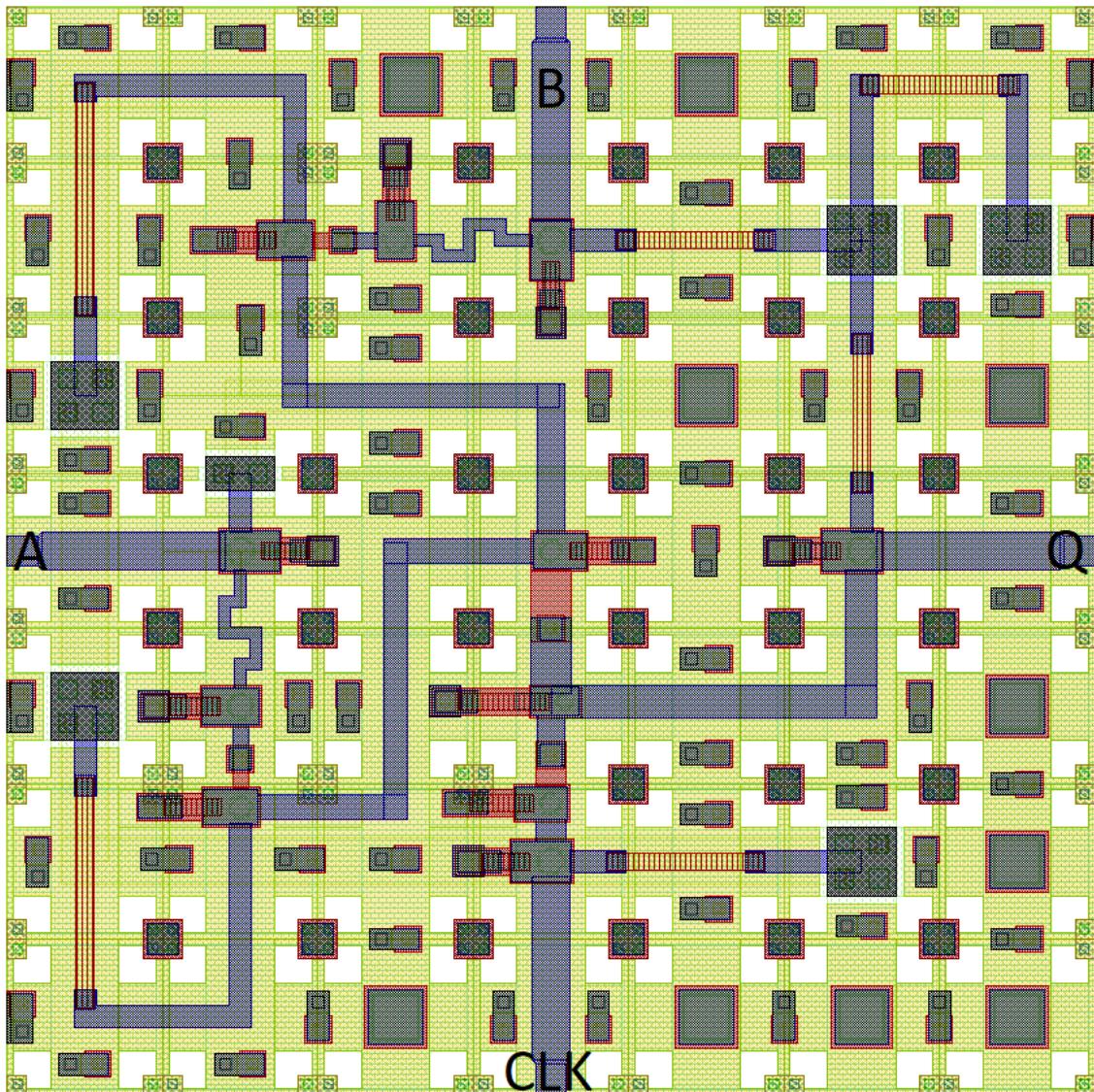


Figure 4.41: RSFQ XOR Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param LRB3=(RB3/Rsheet)*Lsheet+LP
2   ↪ by InductEx v.6.0.4 on 1-4-21.          65  .param LRB4=(RB4/Rsheet)*Lsheet+LP
3  * Author: L. Schindler                     66  .param LRB5=(RB5/Rsheet)*Lsheet+LP
4  * Version: 2.1                           67  .param LRB6=(RB6/Rsheet)*Lsheet+LP
5  * Last modification date: 4 June 2021     68  .param LRB7=(RB7/Rsheet)*Lsheet+LP
6  * Last modification by: L. Schindler       69  .param LRB8=(RB8/Rsheet)*Lsheet+LP
7  *$Ports a b clk q                      70  .param LRB9=(RB9/Rsheet)*Lsheet+LP
8  .subckt LSmitll_XOR      a b clk q        71  .param LRB10=(RB10/Rsheet)*Lsheet+LP
9  .model jjmit jj(rtype=1, vg=2.8mV, cap    72  .param LRB11=(RB11/Rsheet)*Lsheet+LP
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    73  B1 1 2 jjmit area=B1
   ↪ )                                         74  B2 4 5 jjmit area=B2
10 .param Phi0=2.067833848E-15                75  B3 25 6 jjmit area=B3
11 .param B0=1                                76  B4 9 10 jjmit area=B4
12 .param Ic0=0.0001                          77  B5 12 13 jjmit area=B5
13 .param IcRs=100u*6.859904418               78  B6 26 14 jjmit area=B6
14 .param B0Rs=IcRs/Ic0*B0                   79  B7 27 16 jjmit area=B7
15 .param Rsheet=2                            80  B8 17 18 jjmit area=B8
16 .param Lsheet=1.13e-12                     81  B9 20 16 jjmit area=B9
17 .param LP=0.5p                            82  B10 16 21 jjmit area=B10
18 .param IC=2.5                             83  B11 22 23 jjmit area=B11
19 .param LB=2p                             84  IB1 0 3 pwl(0 0 5p IB1)
20 .param BiasCoef=0.70                     85  IB2 0 7 pwl(0 0 5p IB2)
21 .param B1=2.5                            86  IB3 0 11 pwl(0 0 5p IB3)
22 .param B2=2.09                           87  IB4 0 15 pwl(0 0 5p IB4)
23 .param B3=1.71                           88  IB5 0 19 pwl(0 0 5p IB5)
24 .param B4=B1                            89  IB6 0 24 pwl(0 0 5p IB6)
25 .param B5=B2                            90  LB1 3 1 LB
26 .param B6=B3                            91  LB2 7 6 LB
27 .param B7=1.62                           92  LB3 11 9 LB
28 .param B8=2.5                            93  LB4 15 14 LB
29 .param B9=1.45                           94  LB5 19 17 LB
30 .param B10=0.89                           95  LB6 24 22 LB
31 .param B11=2.5                           96  L1 a 1 2.06E-12
32 .param IB1=175u                           97  L2 1 4 3.233E-12
33 .param IB2=112u                           98  L3 4 25 1.419E-12
34 .param IB3=IB1                           99  L4 6 8 6.051E-12
35 .param IB4=IB2                           100 L5 b 9 2.092E-12
36 .param IB5=175u                           101 L6 9 12 3.221E-12
37 .param IB6=175u                           102 L7 12 26 1.384E-12
38 .param L1=Phi0/(4*IC*Ic0)                103 L8 14 8 6.059E-12
39 .param L2=Phi0/(2*B1*Ic0)                104 L9 8 27 1.301E-12
40 .param L3=1.2p                            105 L10 clk 17 2.082E-12
41 .param L4=Phi0/(B2*Ic0)                  106 L11 17 20 1.43E-12
42 .param L5=L1                            107 L12 16 22 3.892E-12
43 .param L6=L2                            108 L13 22 q 2.077E-12
44 .param L7=L3                            109 LP1 2 0 5.508E-13
45 .param L8=L4                            110 LP2 5 0 4.769E-13
46 .param L9=1.2p                           111 LP4 10 0 4.767E-13
47 .param L10=Phi0/(4*IC*Ic0)               112 LP5 13 0 4.812E-13
48 .param L11=Phi0/(2*B8*Ic0)               113 LP8 18 0 4.526E-13
49 .param L12=Phi0/(2*B10*Ic0)              114 LP10 21 0 5.69E-13
50 .param L13=Phi0/(4*IC*Ic0)               115 LP11 23 0 4.746E-13
51 .param RB1=B0Rs/B1                      116 RB1 1 101 RB1
52 .param RB2=B0Rs/B2                      117 LRB1 101 0 LRB1
53 .param RB3=B0Rs/B3                      118 RB2 4 104 RB2
54 .param RB4=B0Rs/B4                      119 LRB2 104 0 LRB2
55 .param RB5=B0Rs/B5                      120 RB3 4 106 RB3
56 .param RB6=B0Rs/B6                      121 LRB3 106 6 LRB3
57 .param RB7=B0Rs/B7                      122 RB4 9 109 RB4
58 .param RB8=B0Rs/B8                      123 LRB4 109 0 LRB4
59 .param RB9=B0Rs/B9                      124 RB5 12 112 RB5
60 .param RB10=B0Rs/B10                     125 LRB5 112 0 LRB5
61 .param RB11=B0Rs/B11                     126 RB6 12 114 RB6
62 .param LRB1=(RB1/Rsheet)*Lsheet+LP      127 LRB6 114 14 LRB6
63 .param LRB2=(RB2/Rsheet)*Lsheet+LP      128 RB7 8 108 RB7
                                         129 LRB7 108 16 LRB7

```

```

130 | RB8 17 117 RB8
131 | LRB8 117 0 LRB8
132 | RB9 20 120 RB9
133 | LRB9 120 16 LRB9
134 | RB10 16 116 RB10
135 | LRB10 116 0 LRB10
136 | RB11 22 122 RB11
137 | LRB11 122 0 LRB11
138 | .ends

```

Listing 4.23: RSFQ XOR JoSIM netlist.**Table 4.23:** RSFQ XOR pin list.

Pin	Description
a	Data input
b	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ XOR using JoSIM is shown in Fig. 4.42. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b**,
- (d) the phase over the input JJ of pin **b**,
- (e) the current through the input inductor connected to pin **clk**,
- (f) the phase over the input JJ of pin **clk**,
- (g) the phase over the output JJ of pin **q**, and
- (h) the phase over the input JJ of the load circuit connected to pin **q**.

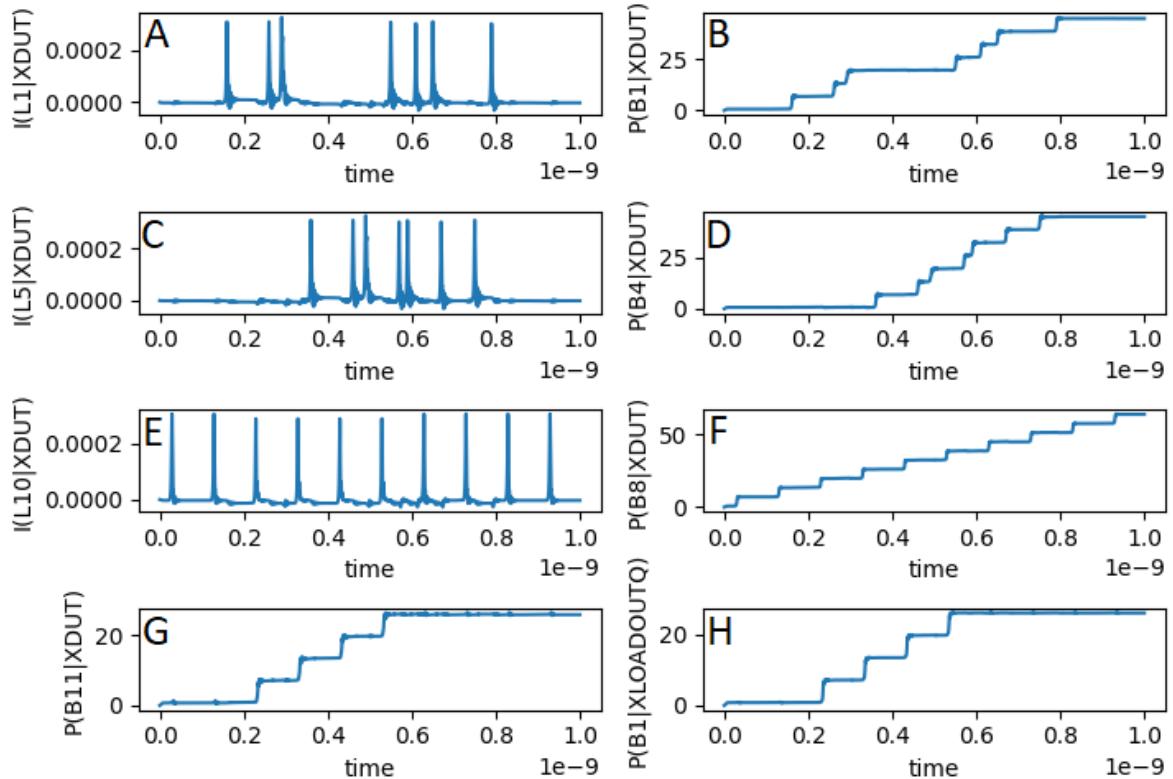


Figure 4.42: RSFQ XOR analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 4 June 2021
5 // Last modification by: L. Schindler
6 // -----
7
8 // -----
9 // Automatically extracted verilog file, created with TimEx v2.05
10 // Timing description and structural design for IARPA-BAA-14-03 via
11 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
12 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
13 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
14 // (c) 2016-2018 Stellenbosch University
15 // -----
16 `timescale 1ps/100fs
17 module LSmitll_XOR_v2p1_optimized (a, b, clk, q);
18
19 input
20   a, b, clk;
21
22 output
23   q;
24
25 reg
26   q;
27
28 real
29   delay_state1_clk_q = 5.8,
30   delay_state2_clk_q = 5.8,
31   ct_state0_a_clk = 0.2,
32   ct_state1_a_b = 8.0,
33   ct_state1_a_clk = 8.1,
34   ct_state1_b_b = 3.5,
35   ct_state1_clk_b = 4.3,
36   ct_state2_a_a = 3.5,
37   ct_state2_b_a = 7.9,
38   ct_state2_b_clk = 7.9,
39   ct_state2_clk_a = 4.4;
40
41 reg
42   errorsignal_a,
43   errorsignal_b,
44   errorsignal_clk;
45
46 integer
47   outfile,
48   cell_state; // internal state of the cell
49
50 initial
51 begin
52   errorsignal_a = 0;
53   errorsignal_b = 0;
54   errorsignal_clk = 0;
55   cell_state = 0; // Startup state
56   q = 0; // All outputs start at 0
57 end
58
59 always @ (posedge a or negedge a) // execute at positive and negative edges of input
60 begin
61   if ($time>4) // arbitrary steady-state time)
62     begin
63       if (errorsignal_a == 1'b1) // A critical timing is active for this input
64         begin
65           outfile = $fopen("errors.txt", "a");
66           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
67             , $stime);

```

```

67          $fclose(outfile);
68          q <= 1'bX; // Set all outputs to unknown
69      end
70      if (errorsignal_a == 0)
71      begin
72          case (cell_state)
73              0: begin
74                  cell_state = 1; // Blocking statement -- immediately
75                  errorsignal_clk = 1; // Critical timing on this input; assign
76                      // immediately
77                  errorsignal_clk <= #(ct_state0_a_clk) 0; // Clear error signal
78                      // after critical timing expires
79              end
80              1: begin
81                  errorsignal_b = 1; // Critical timing on this input; assign
82                      // immediately
83                  errorsignal_b <= #(ct_state1_a_b) 0; // Clear error signal
84                      // after critical timing expires
85                  errorsignal_clk = 1; // Critical timing on this input; assign
86                      // immediately
87                  errorsignal_clk <= #(ct_state1_a_clk) 0; // Clear error signal
88                      // after critical timing expires
89              end
90          endcase
91      end
92  end
93
94 always @ (posedge b or negedge b) // execute at positive and negative edges of input
95 begin
96     if ($time>4) // arbitrary steady-state time)
97     begin
98         if (errorsignal_b == 1'b1) // A critical timing is active for this input
99         begin
100             outfile = $fopen("errors.txt", "a");
101             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
102                         // ", $stime);
103             $fclose(outfile);
104             q <= 1'bX; // Set all outputs to unknown
105         end
106         if (errorsignal_b == 0)
107         begin
108             case (cell_state)
109                 0: begin
110                     cell_state = 2; // Blocking statement -- immediately
111                 end
112                 1: begin
113                     cell_state = 0; // Blocking statement -- immediately
114                     errorsignal_b = 1; // Critical timing on this input; assign
115                         // immediately
116                     errorsignal_b <= #(ct_state1_b_b) 0; // Clear error signal
117                         // after critical timing expires
118                 end
119                 2: begin
120                     errorsignal_a = 1; // Critical timing on this input; assign
121                         // immediately
122                     errorsignal_a <= #(ct_state2_b_a) 0; // Clear error signal
123                         // after critical timing expires
124                     errorsignal_clk = 1; // Critical timing on this input; assign
125                         // immediately
126                     errorsignal_clk <= #(ct_state2_b_clk) 0; // Clear error signal
127                         // after critical timing expires
128                 end
129             endcase
130         end
131     end
132 
```

```

122           endcase
123       end
124   end
125
126
127 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
128 begin
129     if ($time>4) // arbitrary steady-state time)
130     begin
131         if (errorsignal_clk == 1'b1) // A critical timing is active for this input
132         begin
133             outfile = $fopen("errors.txt", "a");
134             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
135                         ↪ ", $stime);
136             $fclose(outfile);
137             q <= 1'bX; // Set all outputs to unknown
138         end
139         if (errorsignal_clk == 0)
140         begin
141             case (cell_state)
142                 0: begin
143                     end
144                 1: begin
145                     q <= #(delay_state1_clk_q) !q;
146                     cell_state = 0; // Blocking statement -- immediately
147                     errorsignal_b = 1; // Critical timing on this input; assign
148                         ↪ immediately
149                     errorsignal_b <= #(ct_state1_clk_b) 0; // Clear error signal
150                         ↪ after critical timing expires
151                 end
152                 2: begin
153                     q <= #(delay_state2_clk_q) !q;
154                     cell_state = 0; // Blocking statement -- immediately
155                     errorsignal_a = 1; // Critical timing on this input; assign
156                         ↪ immediately
157                     errorsignal_a <= #(ct_state2_clk_a) 0; // Clear error signal
158                         ↪ after critical timing expires
159                 end
160             endcase
161         end
162     end
163 endmodule

```

Listing 4.24: RSFQ XOR verilog model.

The digital simulation results for the RSFQ XOR is shown in Fig. 4.43 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 4.44.

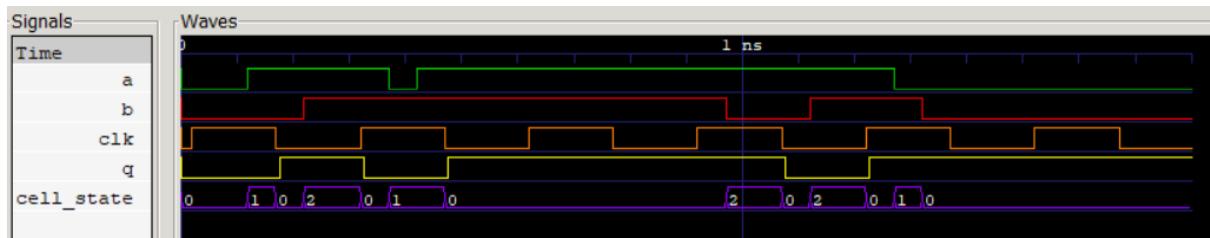


Figure 4.43: RSFQ XOR digital simulation results.

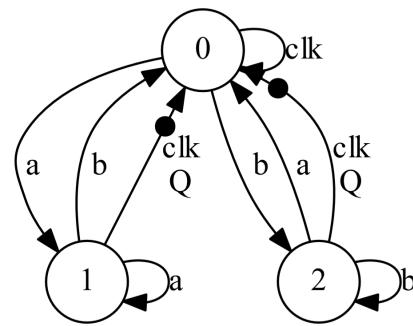


Figure 4.44: RSFQ XOR Mealy finite state machine diagram.

Power Consumption

Table 4.24: RSFQ XOR power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2402	4.64
2	2402	9.29
5	2402	23.2
10	2402	46.4
20	2402	92.9
50	2402	232

4.2.4 NOT

The RSFQ NOT cell is a signal inverting cell driven by a clock pulse signal line. The NOT cell is not designed to be directly connected to a PTL.

Schematic

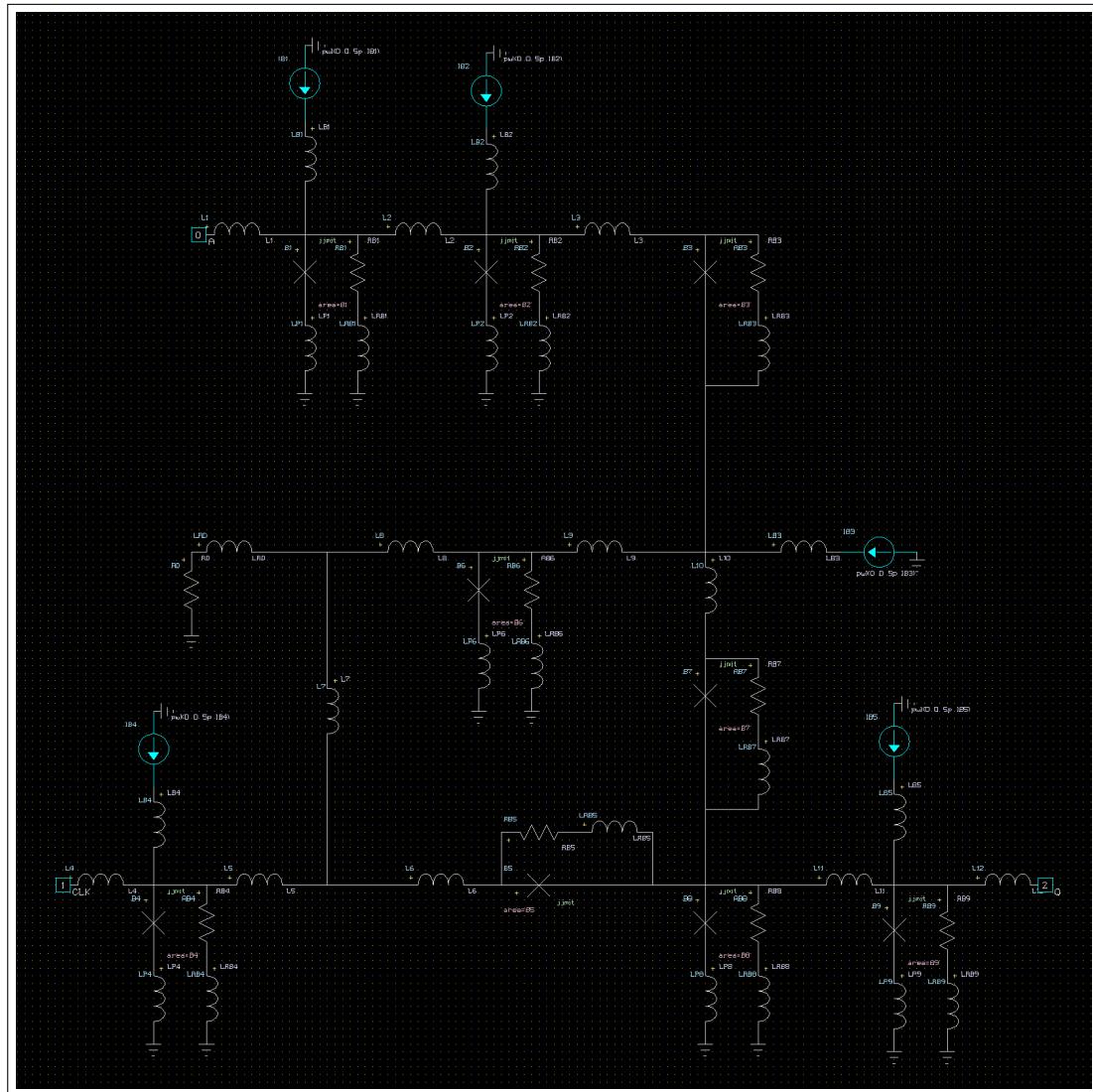


Figure 4.45: Schematic of RSFQ NOT.

Layout

The physical layout for the RSFQ NOT is shown in Fig. 4.46. The layout height is $70 \mu\text{m}$ and the width is $70 \mu\text{m}$. The cell includes an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

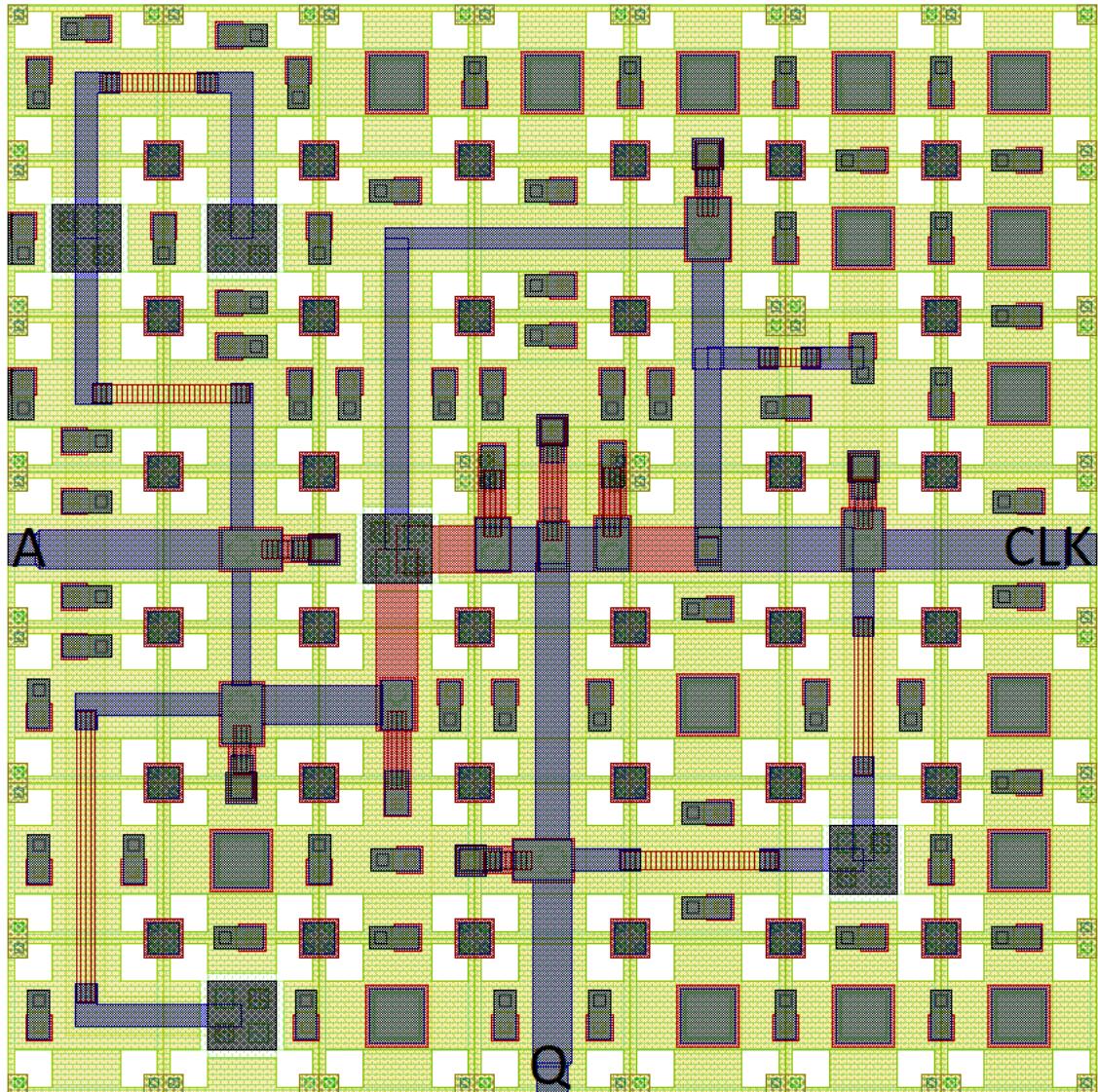


Figure 4.46: RSFQ NOT Layout.

Analog model

```

1  * Back-annotated simulation file written      65 | .param LRD=2p
2   ↪ by InductEx v.6.0.4 on 8-4-21.          66 | B1 1 2 jjmit area=B1
3  * Author: L. Schindler                     67 | B2 4 5 jjmit area=B2
4  * Version: 2.1                            68 | B3 7 8 jjmit area=B3
5  * Last modification date: 3 June 2021     69 | B4 13 14 jjmit area=B4
6  * Last modification by: L. Schindler       70 | B5 17 18 jjmit area=B5
7  *$Ports a clk q                         71 | B6 10 11 jjmit area=B6
8 .subckt LSMITLL_NOT a clk q             72 | B7 20 18 jjmit area=B7
9 .model jjmit jj(rtype=1, vg=2.8mV, cap    73 | B8 18 19 jjmit area=B8
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    74 | B9 21 22 jjmit area=B9
   ↪ )                                         75 |
10 .param Phi0=2.067833848E-15            76 |
11 .param B0=1                           77 | IB1 0 3 pw1(0 0 5p IB1)
12 .param Ic0=0.0001                     78 | IB2 0 6 pw1(0 0 5p IB2)
13 .param IcRs=100u*6.859904418        79 | IB3 0 9 pw1(0 0 5p IB3)
14 .param B0Rs=IcRs/Ic0*B0              80 | IB4 0 15 pw1(0 0 5p IB4)
15 .param Rsheet=2                      81 | IB5 0 23 pw1(0 0 5p IB5)
16 .param Lsheet=1.13e-12                82 |
17 .param LP=0.2p                       83 | LB1 3 1 LB1
18 .param IC=2.5                        84 | LB2 6 4 LB2
19 .param LB=2p                         85 | LB3 8 9 LB3
20 .param BiasCoef=0.7                 86 | LB4 13 15 LB4
21                                         87 | LB5 21 23 LB5
22 .param B1=IC                         88 |
23 .param B2=2.57                       89 | L1 a 1 2.062E-12
24 .param B3=1.07                       90 | L2 1 4 1.889E-12
25 .param B4=IC                         91 | L3 4 7 2.72E-12
26 .param B5=1.34                       92 | L4 clk 13 2.057E-12
27 .param B6=3.03                       93 | L5 13 16 1.029E-12
28 .param B7=1.38                       94 | L6 16 17 1.241E-12
29 .param B8=0.8                         95 | L7 16 12 1.973E-12
30 .param B9=IC                         96 | L8 10 12 1.003E-12
31                                         97 | L9 10 8 7.524E-12
32 .param IB1=BiasCoef*Ic0*B1          98 | L10 8 20 1.234E-12
33 .param IB2=87u                       99 | L11 18 21 2.607E-12
34 .param IB3=257u                      100 | L12 21 q 2.062E-12
35 .param IB4=BiasCoef*Ic0*B4          101 |
36 .param IB5=BiasCoef*Ic0*B9          102 | LP1 2 0 5.271E-13
37                                         103 | LP2 5 0 5.237E-13
38 .param LB1=LB                         104 | LP4 14 0 4.759E-13
39 .param LB2=LB                         105 | LP6 11 0 5.021E-13
40 .param LB3=LB                         106 | LP8 19 0 6.33E-13
41 .param LB4=LB                         107 | LP9 22 0 4.749E-13
42 .param LB5=LB                         108 |
43                                         109 | RB1 1 101 RB1
44 .param RB1=B0Rs/B1                  110 | LRB1 101 0 LRB1
45 .param RB2=B0Rs/B2                  111 | RB2 4 104 RB2
46 .param RB3=B0Rs/B3                  112 | LRB2 104 5 LRB2
47 .param RB4=B0Rs/B4                  113 | RB3 7 107 RB3
48 .param RB5=B0Rs/B5                  114 | LRB3 107 8 LRB3
49 .param RB6=B0Rs/B6                  115 | RB4 13 113 RB4
50 .param RB7=B0Rs/B7                  116 | LRB4 113 0 LRB4
51 .param RB8=B0Rs/B8                  117 | RB5 17 117 RB5
52 .param RB9=B0Rs/B9                  118 | LRB5 117 18 LRB5
53                                         119 | RB6 10 110 RB6
54 .param LRB1=(RB1/Rsheet)*Lsheet     120 | LRB6 110 0 LRB6
55 .param LRB2=(RB2/Rsheet)*Lsheet     121 | RB7 20 120 RB7
56 .param LRB3=(RB3/Rsheet)*Lsheet     122 | LRB7 120 18 LRB7
57 .param LRB4=(RB4/Rsheet)*Lsheet     123 | RB8 18 118 RB8
58 .param LRB5=(RB5/Rsheet)*Lsheet     124 | LRB8 118 0 LRB8
59 .param LRB6=(RB6/Rsheet)*Lsheet     125 | RB9 21 121 RB9
60 .param LRB7=(RB7/Rsheet)*Lsheet     126 | LRB9 121 0 LRB9
61 .param LRB8=(RB8/Rsheet)*Lsheet     127 | LRD 12 112 LRD
62 .param LRB9=(RB9/Rsheet)*Lsheet     128 | RD 112 0 RD
63                                         129 | .ends
64 .param RD=4

```

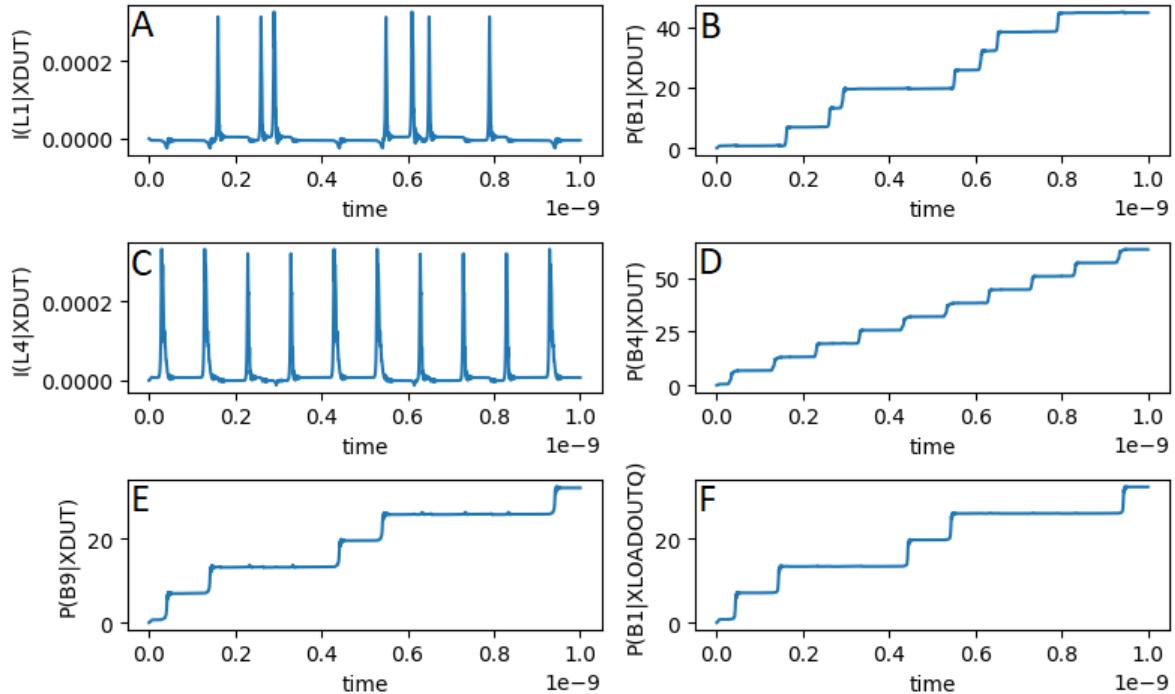
Listing 4.25: RSFQ NOT JoSIM netlist.

Table 4.25: RSFQ NOT pin list.

Pin	Description
a	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ NOT using JoSIM is shown in Fig. 4.47. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **clk**,
- (d) the phase over the input JJ of pin **clk**,
- (e) the phase over the output JJ of pin **q**, and
- (f) the phase over the input JJ of the load circuit connected to pin **q**.

**Figure 4.47:** RSFQ NOT analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 8 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 //
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 //
15 'timescale 1ps/100fs
16 module LSmitll_NOT_v2p1_optimized (a, clk, q);
17
18 input
19   a, clk;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state0_clk_q = 10.8,
29   ct_state0_a_clk = 2.6,
30   ct_state0_clk_a = 8.4,
31   ct_state0_clk_clk = 12.8,
32   ct_state1_a_clk = 4.2;
33
34 reg
35   errorsignal_a,
36   errorsignal_clk;
37
38 integer
39   outfile,
40   cell_state; // internal state of the cell
41
42 initial
43 begin
44   errorsignal_a = 0;
45   errorsignal_clk = 0;
46   cell_state = 0; // Startup state
47   q = 0; // All outputs start at 0
48 end
49
50 always @ (posedge a or negedge a) // execute at positive and negative edges of input
51 begin
52   if ($time > 4) // arbitrary steady-state time)
53     begin
54       if (errorsignal_a == 1'b1) // A critical timing is active for this input
55         begin
56           outfile = $fopen("errors.txt", "a");
57           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
58           ↪ ", $stime);
59           $fclose(outfile);
60           q <= 1'bX; // Set all outputs to unknown
61         end
62       if (errorsignal_a == 0)
63         begin
64           case (cell_state)
65             0: begin
66               cell_state = 1; // Blocking statement -- immediately
67               errorsignal_clk = 1; // Critical timing on this input; assign

```

```

67           → immediately
68           errorsignal_clk <= #(ct_state0_a_clk) 0; // Clear error signal
69           → after critical timing expires
70       end
71   1: begin
72       errorsignal_clk = 1; // Critical timing on this input; assign
73           → immediately
74       errorsignal_clk <= #(ct_state1_a_clk) 0; // Clear error signal
75           → after critical timing expires
76   end
77 endcase
78 end
79
80 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
81 begin
82     if ($time>4) // arbitrary steady-state time)
83     begin
84         if (errorsignal_clk == 1'b1) // A critical timing is active for this input
85         begin
86             outfile = $fopen("errors.txt", "a");
87             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
88             → ", $time);
89             $fclose(outfile);
90             q <= 1'bX; // Set all outputs to unknown
91         end
92         if (errorsignal_clk == 0)
93         begin
94             case (cell_state)
95             0: begin
96                 q <= #(delay_state0_clk_q) !q;
97                 errorsignal_a = 1; // Critical timing on this input; assign
98                 → immediately
99                 errorsignal_a <= #(ct_state0_clk_a) 0; // Clear error signal
100                → after critical timing expires
101                errorsignal_clk = 1; // Critical timing on this input; assign
102                → immediately
103                errorsignal_clk <= #(ct_state0_clk_clk) 0; // Clear error
104                → signal after critical timing expires
105            end
106        end
107    endmodule

```

Listing 4.26: RSFQ NOT verilog model.

The digital simulation results for the RSFQ NOT is shown in Fig. 4.48 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 4.49.

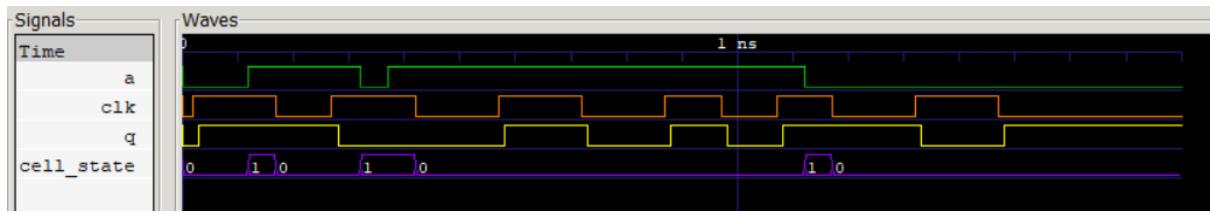


Figure 4.48: RSFQ NOT digital simulation results.

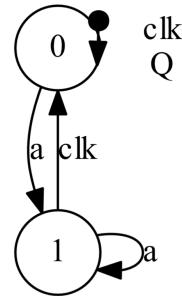


Figure 4.49: RSFQ NOT Mealy finite state machine diagram.

Power Consumption

Table 4.26: RSFQ NOT power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2259	3.66
2	2259	7.32
5	2259	18.3
10	2259	36.6
20	2259	73.2
50	2259	183

4.2.5 XNOR

The RSFQ XNOR cell is a combination of a NOT and XOR cell. It generates an output pulse exclusively if no input pulse was received or if a pulse from both input lines was received before the clock signal. The XNOR cell is not designed to be directly connected to a PTL.

Schematic

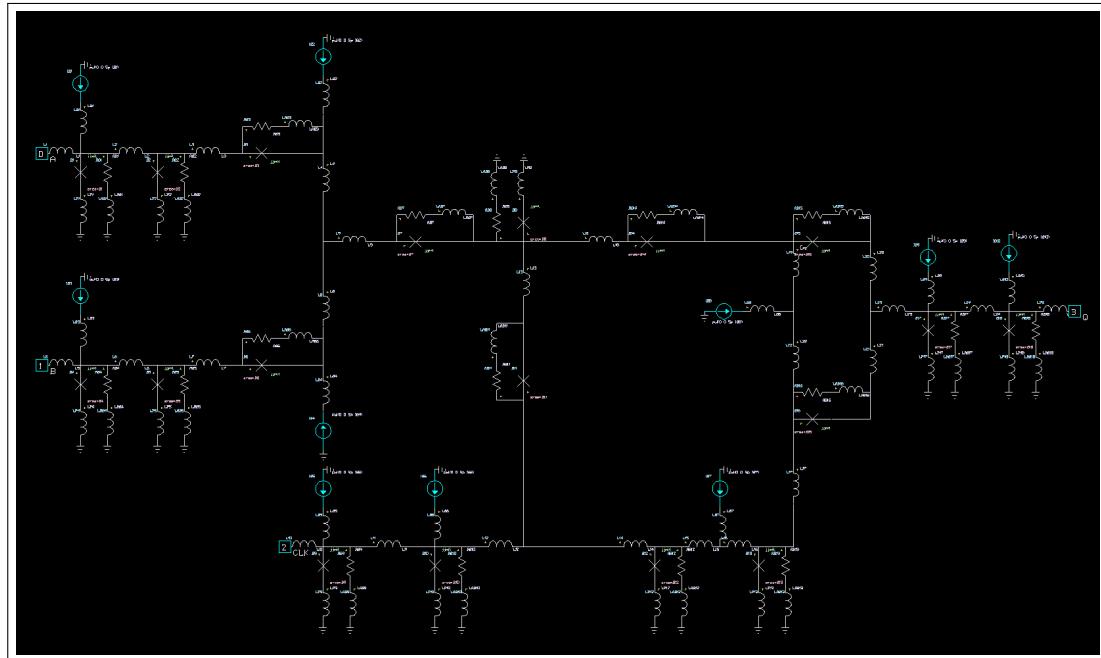


Figure 4.50: Schematic of RSFQ XNOR.

Layout

The physical layout for the RSFQ XNOR is shown in Fig. 4.51. The layout height is $70 \mu m$ and the width is $100 \mu m$. The cell includes an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

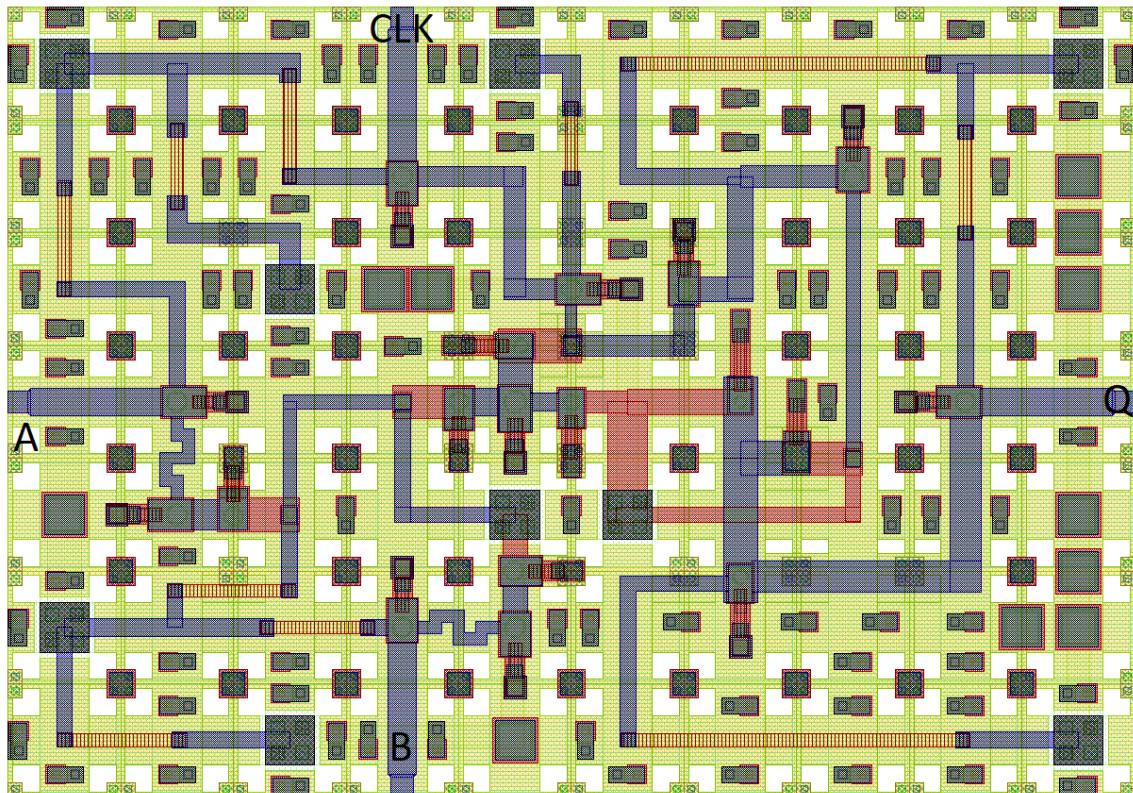


Figure 4.51: RSFQ XNOR Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param L12=1p
2   ↪ by InductEx v.6.0.4 on 16-4-21.          65  .param L13=1p
3  * Author: L. Schindler                      66  .param L14=Phi0/(2*B10*Ic0)
4  * Version: 2.1                               67  .param L15=Phi0/(4*B12*Ic0)
5  * Last modification date: 3 June 2021       68  .param L16=Phi0/(4*B12*Ic0)
6  * Last modification by: L. Schindler        69  .param L17=Phi0/(B13*Ic0)
7  *$Ports a b clk q                         70  .param L18=1p
8 .subckt LSMITLL_XNOR a b clk q            71  .param L20=1p
9 .model jjmit jj(rtype=1, vg=2.8mV, cap    72  .param L21=0.5p
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    73  .param L22=Phi0/(B16*Ic0)
   ↪ )                                         74  .param L23=Phi0/(4*B16*Ic0)
10 .param Phi0=2.067833848E-15                75  .param L24=Phi0/(2*B17*Ic0)
11 .param B0=1                                 76  .param L25=Lptl
12 .param Ic0=0.0001                           77
13 .param IcRs=100u*6.859904418               78  .param RB1=B0Rs/B1
14 .param B0Rs=IcRs/Ic0*B0                   79  .param RB2=B0Rs/B2
15 .param Rsheet=2                            80  .param RB3=B0Rs/B3
16 .param Lsheet=1.13e-12                     81  .param RB4=B0Rs/B4
17 .param LP=0.2p                            82  .param RB5=B0Rs/B5
18 .param IC=2.5                             83  .param RB6=B0Rs/B6
19 .param Lptl=2p                            84  .param RB7=B0Rs/B7
20 .param LB=2p                             85  .param RB8=B0Rs/B8
21 .param BiasCoef=0.7                      86  .param RB9=B0Rs/B9
22
23 .param B1=2.5                            87  .param RB10=B0Rs/B10
24 .param B2=2.66                           88  .param RB11=B0Rs/B11
25 .param B3=2.34                           89  .param RB12=B0Rs/B12
26 .param B4=B1                            90  .param RB13=B0Rs/B13
27 .param B5=B2                            91  .param RB14=B0Rs/B14
28 .param B6=B3                            92  .param RB15=B0Rs/B15
29 .param B7=2.31                           93  .param RB16=B0Rs/B16
30 .param B8=3.15                           94  .param RB17=B0Rs/B17
31 .param B9=2.5                            95  .param RB18=B0Rs/B18
32 .param B10=2.58                          96  .param LRB1=(RB1/Rsheet)*Lsheet
33 .param B11=1.44                          97  .param LRB2=(RB2/Rsheet)*Lsheet
34 .param B12=2.60                          98  .param LRB3=(RB3/Rsheet)*Lsheet
35 .param B13=2.71                          99  .param LRB4=(RB4/Rsheet)*Lsheet
36 .param B14=1.63                          100 .param LRB5=(RB5/Rsheet)*Lsheet
37 .param B15=0.95                          101 .param LRB6=(RB6/Rsheet)*Lsheet
38 .param B16=1.44                          102 .param LRB7=(RB7/Rsheet)*Lsheet
39 .param B17=1.40                          103 .param LRB8=(RB8/Rsheet)*Lsheet
40 .param B18=2.5                           104 .param LRB9=(RB9/Rsheet)*Lsheet
41
42 .param IB1=175E-6                         105 .param LRB10=(RB10/Rsheet)*Lsheet
43 .param IB2=153E-6                         106 .param LRB11=(RB11/Rsheet)*Lsheet
44 .param IB3=IB1                           107 .param LRB12=(RB12/Rsheet)*Lsheet
45 .param IB4=IB2                           108 .param LRB13=(RB13/Rsheet)*Lsheet
46 .param IB5=175E-6                         109 .param LRB14=(RB14/Rsheet)*Lsheet
47 .param IB6=260E-6                         110 .param LRB15=(RB15/Rsheet)*Lsheet
48 .param IB7=56E-6                          111 .param LRB16=(RB16/Rsheet)*Lsheet
49 .param IB8=248E-6                         112 .param LRB17=(RB17/Rsheet)*Lsheet
50 .param IB9=51E-6                          113 .param LRB18=(RB18/Rsheet)*Lsheet
51 .param IB10=175E-6                        114
52
53 .param L1=Lptl                           115 B1 1 2 jjmit area=B1
54 .param L2=Phi0/(2*B1*Ic0)                 116 B2 3 4 jjmit area=B2
55 .param L3=1p                            117 B3 5 6 jjmit area=B3
56 .param L4=Phi0/(B2*Ic0)                  118 B4 8 9 jjmit area=B4
57 .param L5=L1                            119 B5 10 11 jjmit area=B5
58 .param L6=L2                            120 B6 12 13 jjmit area=B6
59 .param L7=L3                            121 B7 14 15 jjmit area=B7
60 .param L8=L4                            122 B8 15 16 jjmit area=B8
61 .param L9=1p                            123 B9 17 18 jjmit area=B9
62 .param L10=Lptl                          124 B10 19 20 jjmit area=B10
63 .param L11=Phi0/(2*B9*Ic0)              125 B11 22 15 jjmit area=B11
                                         126 B12 23 24 jjmit area=B12
                                         127 B13 26 27 jjmit area=B13
                                         128 B14 29 34 jjmit area=B14
                                         129 B15 34 31 jjmit area=B15

```

```

130 | B16 28 33 jjmit area=B16
131 | B17 35 36 jjmit area=B17
132 | B18 37 38 jjmit area=B18
133 |
134 | LP1 2 0 5.29E-13
135 | LP2 4 0 4.956E-13
136 | LP4 9 0 4.424E-13
137 | LP5 11 0 5.101E-13
138 | LP8 16 0 5.051E-13
139 | LP9 18 0 5.321E-13
140 | LP10 20 0 5.07E-13
141 | LP12 24 0 5.287E-13
142 | LP13 27 0 5.393E-13
143 | LP17 36 0 5.753E-13
144 | LP18 38 0 5.361E-13
145 |
146 | IB1 0 1 pwl(0 0 5p IB1)
147 | IB2 0 6 pwl(0 0 5p IB2)
148 | IB3 0 8 pwl(0 0 5p IB3)
149 | IB4 0 13 pwl(0 0 5p IB4)
150 | IB5 0 17 pwl(0 0 5p IB5)
151 | IB6 0 19 pwl(0 0 5p IB6)
152 | IB7 0 25 pwl(0 0 5p IB7)
153 | IB8 0 34 pwl(0 0 5p IB8)
154 | IB9 0 35 pwl(0 0 5p IB9)
155 | IB10 0 37 pwl(0 0 5p IB10)
156 |
157 | L1 a 1 2.058E-12
158 | L2 1 3 2.395E-12
159 | L3 3 5 1.108E-12
160 | L4 6 7 4.1E-12
161 | L5 b 8 2.104E-12
162 | L6 8 10 2.374E-12
163 | L7 10 12 9.887E-13
164 | L8 13 7 4.048E-12
165 | L9 7 14 1.15E-12
166 | L10 clk 17 2.041E-12
167 | L11 17 19 3.442E-12
168 | L12 19 21 8.703E-13
169 | L13 21 22 1.117E-12
170 | L14 21 23 2.107E-12
171 | L15 23 25 1.641E-12
172 | L16 25 26 1.335E-12
173 | L17 26 28 5.203E-12
174 | L18 15 29 1.494E-12
175 | L20 31 32 2.081E-12

176 | L21 33 32 9.13E-13
177 | L22 34 28 6.448E-12
178 | L23 32 35 1.012E-12
179 | L24 35 37 3.546E-12
180 | L25 37 q 2.034E-12
181 |
182 | RB1 1 101 RB1
183 | LRB1 101 0 LRB1
184 | RB2 3 103 RB2
185 | LRB2 103 0 LRB2
186 | RB3 5 105 RB3
187 | LRB3 105 6 LRB3
188 | RB4 8 108 RB4
189 | LRB4 108 0 LRB4
190 | RB5 10 110 RB5
191 | LRB5 110 0 LRB5
192 | RB6 12 112 RB6
193 | LRB6 112 13 LRB6
194 | RB7 14 114 RB7
195 | LRB7 114 15 LRB7
196 | RB8 15 115 RB8
197 | LRB8 115 0 LRB8
198 | RB9 17 117 RB9
199 | LRB9 117 0 LRB9
200 | RB10 19 119 RB10
201 | LRB10 119 0 LRB10
202 | RB11 22 122 RB11
203 | LRB11 122 15 LRB11
204 | RB12 23 123 RB12
205 | LRB12 123 0 LRB12
206 | RB13 26 126 RB13
207 | LRB13 126 0 LRB13
208 | RB14 29 129 RB14
209 | LRB14 129 34 LRB14
210 | RB15 34 130 RB15
211 | LRB15 130 31 LRB15
212 | RB16 28 128 RB16
213 | LRB16 128 33 LRB16
214 | RB17 35 135 RB17
215 | LRB17 135 0 LRB17
216 | RB18 37 137 RB18
217 | LRB18 137 0 LRB18
218 | .ends

```

Listing 4.27: RSFQ XNOR JoSIM netlist.**Table 4.27:** RSFQ XNOR pin list.

Pin	Description
a	Data input
b	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ XNOR using JoSIM is shown in Fig. 4.52. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b**,
- (d) the phase over the input JJ of pin **b**,
- (e) the current through the input inductor connected to pin **clk**,
- (f) the phase over the input JJ of pin **clk**,
- (g) the phase over the output JJ of pin **q**, and
- (h) the phase over the input JJ of the load circuit connected to pin **q**.

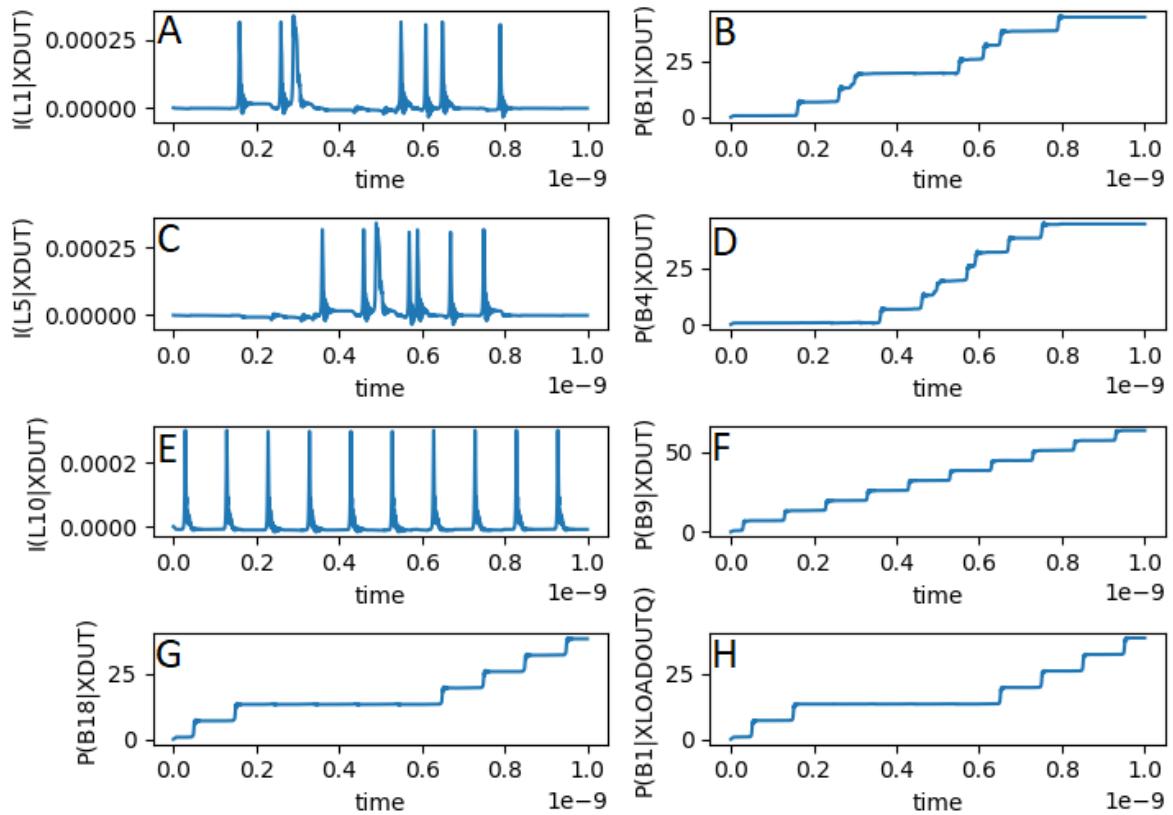


Figure 4.52: RSFQ XNOR analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 3 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 // -----
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 // -----
15 `timescale 1ps/100fs
16 module LSmitll_XNOR_v2p1_optimized (a, b, clk, q);
17
18 input
19   a, b, clk;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state0_clk_q = 21.8,
29   ct_state0_clk_a = 6.6,
30   ct_state0_clk_b = 6.7,
31   ct_state0_clk_clk = 13.4,
32   ct_state1_b_b = 3.5,
33   ct_state1_clk_b = 8.5,
34   ct_state2_a_a = 3.4,
35   ct_state2_a_clk = 2.6,
36   ct_state2_b_a = 16.2,
37   ct_state2_b_clk = 12.9,
38   ct_state2_clk_a = 8.6;
39
40 reg
41   errorsignal_a,
42   errorsignal_b,
43   errorsignal_clk;
44
45 integer
46   outfile,
47   cell_state; // internal state of the cell
48
49 initial
50 begin
51   errorsignal_a = 0;
52   errorsignal_b = 0;
53   errorsignal_clk = 0;
54   cell_state = 0; // Startup state
55   q = 0; // All outputs start at 0
56 end
57
58 always @ (posedge a or negedge a) // execute at positive and negative edges of input
59 begin
60   if ($time > 4) // arbitrary steady-state time)
61     begin
62       if (errorsignal_a == 1'b1) // A critical timing is active for this input
63         begin
64           outfile = $fopen("errors.txt", "a");
65           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m; %0d_ps.\n",
66           $stime);
67           $fclose(outfile);
68         end
69     end
70   end
71 end

```

```

67           q <= 1'bX; // Set all outputs to unknown
68       end
69   if (errorsignal_a == 0)
70   begin
71       case (cell_state)
72       0: begin
73           cell_state = 1; // Blocking statement -- immediately
74       end
75       1: begin
76           end
77       2: begin
78           cell_state = 0; // Blocking statement -- immediately
79           errorsignal_a = 1; // Critical timing on this input; assign
79           ↪ immediately
80           errorsignal_a <= #(ct_state2_a_a) 0; // Clear error signal
80           ↪ after critical timing expires
81           errorsignal_clk = 1; // Critical timing on this input; assign
81           ↪ immediately
82           errorsignal_clk <= #(ct_state2_a_clk) 0; // Clear error signal
82           ↪ after critical timing expires
83       end
84   endcase
85   end
86 end
87
88
89 always @(posedge b or negedge b) // execute at positive and negative edges of input
90 begin
91     if ($time>4) // arbitrary steady-state time)
92     begin
93         if (errorsignal_b == 1'b1) // A critical timing is active for this input
94         begin
95             outfile = $fopen("errors.txt", "a");
96             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
96             ↪ ", $stime);
97             $fclose(outfile);
98             q <= 1'bX; // Set all outputs to unknown
99         end
100    if (errorsignal_b == 0)
101    begin
102        case (cell_state)
103        0: begin
104            cell_state = 2; // Blocking statement -- immediately
105        end
106        1: begin
107            cell_state = 0; // Blocking statement -- immediately
108            errorsignal_b = 1; // Critical timing on this input; assign
108            ↪ immediately
109            errorsignal_b <= #(ct_state1_b_b) 0; // Clear error signal
109            ↪ after critical timing expires
110        end
111        2: begin
112            errorsignal_a = 1; // Critical timing on this input; assign
112            ↪ immediately
113            errorsignal_a <= #(ct_state2_b_a) 0; // Clear error signal
113            ↪ after critical timing expires
114            errorsignal_clk = 1; // Critical timing on this input; assign
114            ↪ immediately
115            errorsignal_clk <= #(ct_state2_b_clk) 0; // Clear error signal
115            ↪ after critical timing expires
116        end
117    endcase
118    end
119 end
120
121
122 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
123 begin
124     if ($time>4) // arbitrary steady-state time)
125     begin

```

```

126      if (errorsignal_clk == 1'b1) // A critical timing is active for this input
127      begin
128          outfile = $fopen("errors.txt", "a");
129          $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
130                      ↪ ", $stime);
131          $fclose(outfile);
132          q <= 1'bX; // Set all outputs to unknown
133      end
134      if (errorsignal_clk == 0)
135      begin
136          case (cell_state)
137              0: begin
138                  q <= #(delay_state0_clk_q) !q;
139                  errorsignal_a = 1; // Critical timing on this input; assign
140                      ↪ immediately
141                  errorsignal_a <= #(ct_state0_clk_a) 0; // Clear error signal
142                      ↪ after critical timing expires
143                  errorsignal_b = 1; // Critical timing on this input; assign
144                      ↪ immediately
145                  errorsignal_b <= #(ct_state0_clk_b) 0; // Clear error signal
146                      ↪ after critical timing expires
147                  errorsignal_clk = 1; // Critical timing on this input; assign
148                      ↪ immediately
149                  errorsignal_clk <= #(ct_state0_clk_clk) 0; // Clear error
150                      ↪ signal after critical timing expires
151              end
152              1: begin
153                  cell_state = 0; // Blocking statement -- immediately
154                  errorsignal_b = 1; // Critical timing on this input; assign
155                      ↪ immediately
156                  errorsignal_b <= #(ct_state1_clk_b) 0; // Clear error signal
157                      ↪ after critical timing expires
158              end
159          endcase
160      end
161  endmodule

```

Listing 4.28: RSFQ XNOR verilog model.

The digital simulation results for the RSFQ XNOR is shown in Fig. 4.53 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 4.54.



Figure 4.53: RSFQ XNOR digital simulation results.

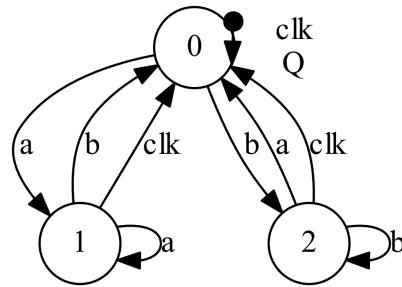


Figure 4.54: RSFQ XNOR Mealy finite state machine diagram.

Power Consumption

Table 4.28: RSFQ XNOR power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	4215	8.31
2	4215	16.6
5	4215	41.6
10	4215	83.1
20	4215	166
50	4215	416

4.3 Buffers

4.3.1 DFF

The RSFQ DFF, D flip-flop, is a multi-state device used to transmit an input set pulse synchronised with a reset (typically clock) signal. The DFF cell is not designed to be directly connected to a PTL.

Schematic

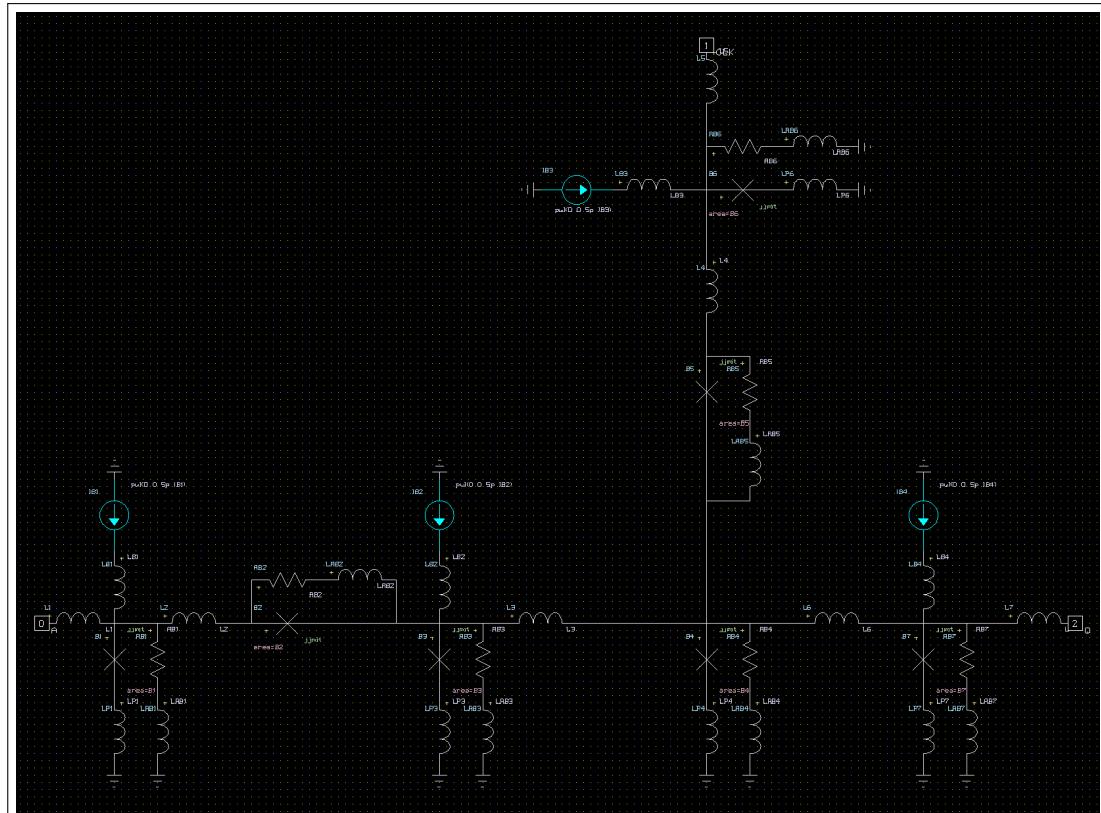


Figure 4.55: Schematic of RSFQ DFF.

Layout

The physical layout for the RSFQ DFF is shown in Fig. 4.56. The layout height is $70 \mu\text{m}$ and the width is $60 \mu\text{m}$. The cell includes an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

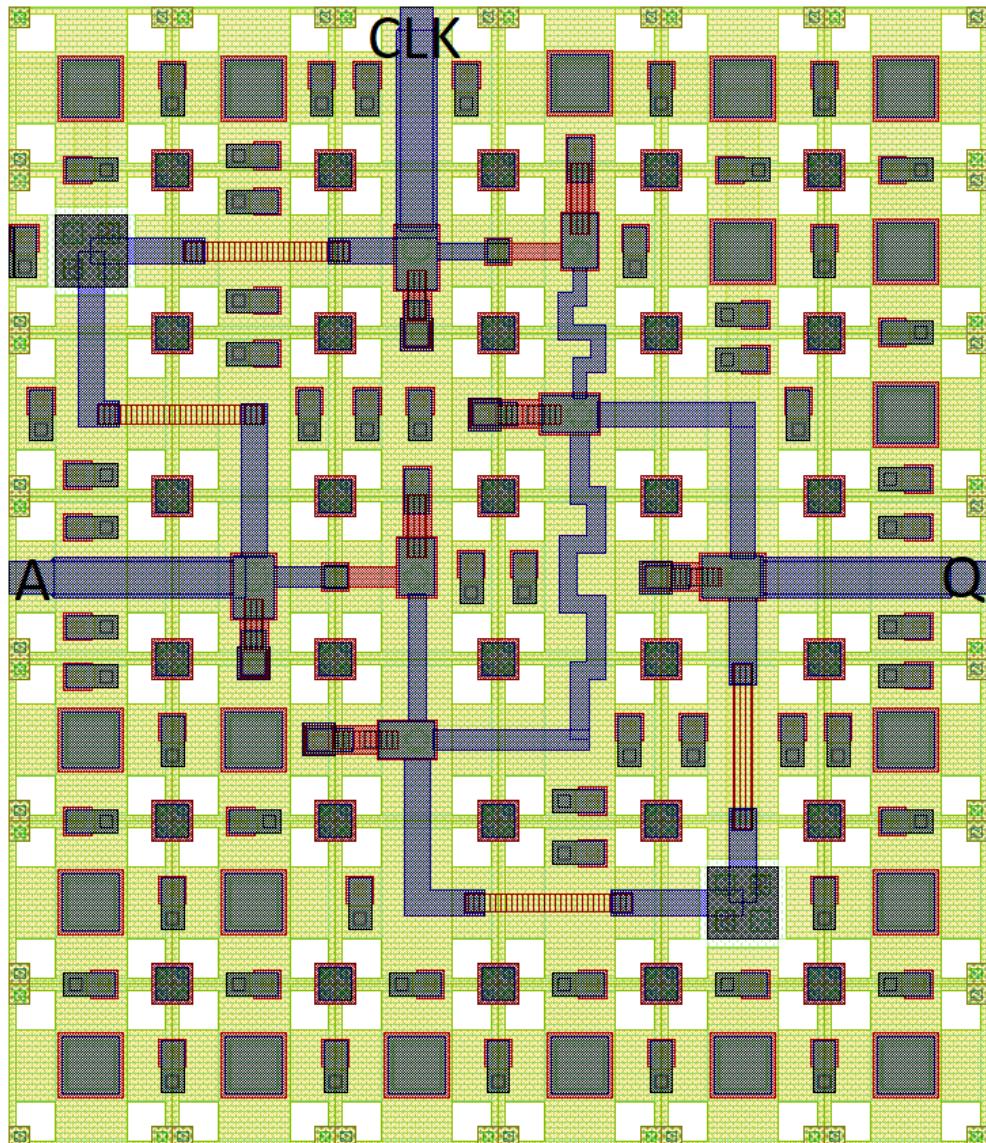


Figure 4.56: RSFQ DFF Layout.

Analog model

```

1  * Back-annotated simulation file written      57  .param RB7=B0Rs/B7
2   ↪ by InductEx v.6.0.4 on 12-3-21.          58  .param LRB1=(RB1/Rsheet)*Lsheet
3  * Author: L. Schindler                      59  .param LRB2=(RB2/Rsheet)*Lsheet
4  * Version: 2.1                               60  .param LRB3=(RB3/Rsheet)*Lsheet
5  * Last modification date: 3 June 2021       61  .param LRB4=(RB4/Rsheet)*Lsheet
6  * Last modification by: L. Schindler        62  .param LRB5=(RB5/Rsheet)*Lsheet
7  *$Ports a clk q                           63  .param LRB6=(RB6/Rsheet)*Lsheet
8  .subckt LSmitll_DFF      a      clk q      64  .param LRB7=(RB7/Rsheet)*Lsheet
9   .model jjmit jj(rtype=1, vg=2.8mV, cap    65
10  ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA     66  B1 1 2 jjmit area=B1
11  .param Phi0=2.067833848E-15                67  B2 4 5 jjmit area=B2
12  .param B0=1                                68  B3 5 6 jjmit area=B3
13  .param Ic0=0.0001                          69  B4 8 9 jjmit area=B4
14  .param IcRs=100u*6.859904418               70  B5 10 8 jjmit area=B5
15  .param B0Rs=IcRs/Ic0*B0                   71  B6 11 12 jjmit area=B6
16  .param Rsheet=2                            72  B7 14 15 jjmit area=B7
17  .param Lsheet=1.13e-12                     73
18  .param LP=0.2p                            74  IB1 0 3 pwl(0 0 5p IB1)
19  .param IC=2.5                            75  IB2 0 7 pwl(0 0 5p IB2)
20  .param LB=2p                             76  IB3 0 13 pwl(0 0 5p IB3)
21  .param BiasCoef=0.70                     77  IB4 0 16 pwl(0 0 5p IB4)
22  .param B1=2.5                            78
23  .param B2=1.61                           79  LB1 3 1 LB1
24  .param B3=1.54                           80  LB2 7 5 LB2
25  .param B4=1.69                           81  LB3 11 13 LB3
26  .param B5=1.38                           82  LB4 16 14 LB4
27  .param B6=2.5                            83
28  .param B7=2.5                            84  L1 a 1 2.059E-12
29                                         85  L2 1 4 4.123E-12
30  .param IB1=175u                          86  L3 5 8 6.873E-12
31  .param IB2=173u                          87  L4 10 11 5.195E-12
32  .param IB3=175u                          88  L5 clk 11 2.071E-12
33  .param IB4=175u                          89  L6 8 14 3.287E-12
34                                         90  L7 14 q 2.066E-12
35  .param L1=Phi0/(4*IC*Ic0)                91
36  .param L2=Phi0/(2*B1*Ic0)                92  LP1 2 0 5.042E-13
37  .param L3=Phi0/(B3*Ic0)                  93  LP3 6 0 5.799E-13
38  .param L4=Phi0/(2*B6*Ic0)                94  LP4 9 0 5.733E-13
39  .param L5=Phi0/(4*IC*Ic0)                95  LP6 12 0 4.605E-13
40  .param L6=Phi0/(2*B4*Ic0)                96  LP7 15 0 4.961E-13
41  .param L7=Phi0/(4*B7*Ic0)                97
42  .param LB1=LB                           98  RB1 1 101 RB1
43  .param LB2=LB                           99  LRB1 101 0 LRB1
44  .param LB3=LB                           100 RB2 4 104 RB2
45  .param LB4=LB                           101 LRB2 104 5 LRB2
46  .param LP1=LP                           102 RB3 5 105 RB3
47  .param LP3=LP                           103 LRB3 105 0 LRB3
48  .param LP4=LP                           104 RB4 8 108 RB4
49  .param LP6=LP                           105 LRB4 108 0 LRB4
50  .param LP7=LP                           106 RB5 10 110 RB5
51  .param RB1=B0Rs/B1                     107 LRB5 110 8 LRB5
52  .param RB2=B0Rs/B2                     108 RB6 11 111 RB6
53  .param RB3=B0Rs/B3                     109 LRB6 111 0 LRB6
54  .param RB4=B0Rs/B4                     110 RB7 14 114 RB7
55  .param RB5=B0Rs/B5                     111 LRB7 114 0 LRB7
56  .param RB6=B0Rs/B6                     112 .ends

```

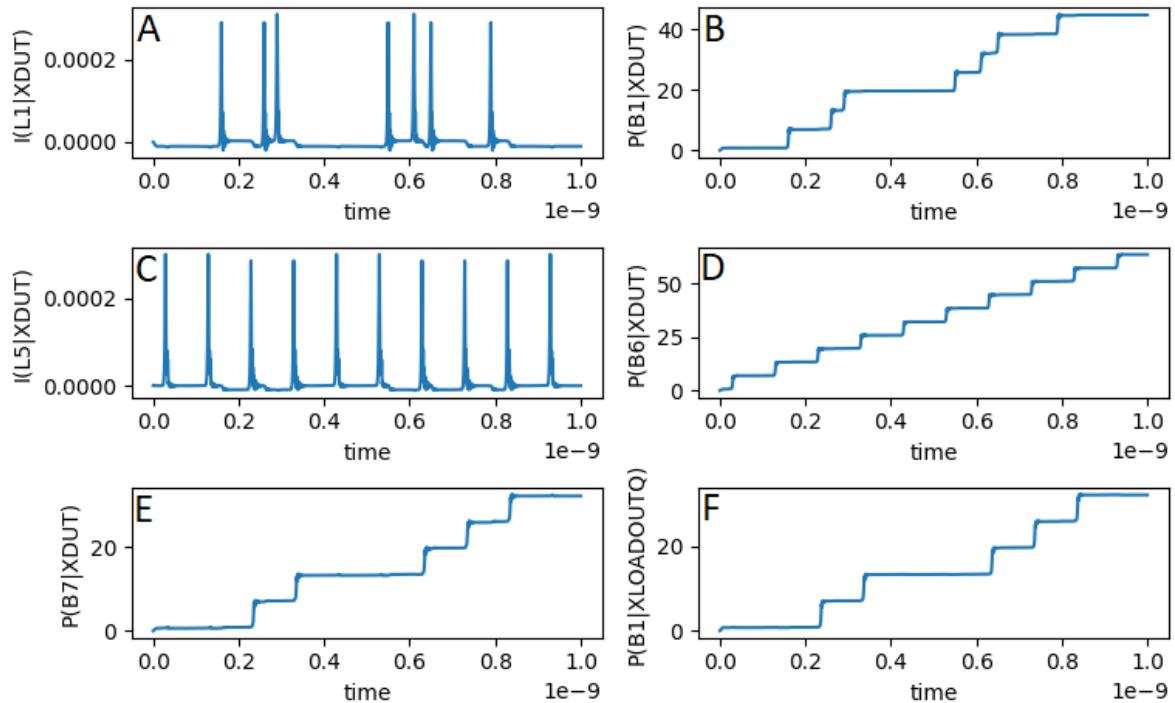
Listing 4.29: RSFQ DFF JoSIM netlist.

Table 4.29: RSFQ DFF pin list.

Pin	Description
a	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ DFF using JoSIM is shown in Fig. 4.57. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **clk**,
- (d) the phase over the input JJ of pin **clk**,
- (e) the phase over the output JJ of pin **q**, and
- (f) the phase over the input JJ of the load circuit connected to pin **q**.

**Figure 4.57:** RSFQ DFF analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 4 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 // -----
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 // -----
15 `timescale 1ps/100fs
16 module LSmitll_DFF_v2p1_optimized (a, clk, q);
17
18 input
19   a, clk;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state1_clk_q = 5.8,
29   ct_state0_clk_a = 0.9,
30   ct_state1_a_clk = 0.6;
31
32 reg
33   errorsignal_a,
34   errorsignal_clk;
35
36 integer
37   outfile,
38   cell_state; // internal state of the cell
39
40 initial
41 begin
42   errorsignal_a = 0;
43   errorsignal_clk = 0;
44   cell_state = 0; // Startup state
45   q = 0; // All outputs start at 0
46 end
47
48 always @ (posedge a or negedge a) // execute at positive and negative edges of input
49 begin
50   if ($time>4) // arbitrary steady-state time)
51     begin
52       if (errorsignal_a == 1'b1) // A critical timing is active for this input
53         begin
54           outfile = $fopen("errors.txt", "a");
55           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
56           ↪ ", $stime);
57           $fclose(outfile);
58           q <= 1'bX; // Set all outputs to unknown
59         end
60       if (errorsignal_a == 0)
61         begin
62           case (cell_state)
63             0: begin
64               cell_state = 1; // Blocking statement -- immediately
65             end
66             1: begin
67               errorsignal_clk = 1; // Critical timing on this input; assign

```

```

67           ↪ immediately
68           errorsignal_clk <= #(ct_state1_a_clk) 0; // Clear error signal
69           ↪ after critical timing expires
70       end
71   endcase
72 end
73
74 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
75 begin
76     if ($time>4) // arbitrary steady-state time)
77     begin
78         if (errorsignal_clk == 1'b1) // A critical timing is active for this input
79         begin
80             outfile = $fopen("errors.txt", "a");
81             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
82             ↪ ", $stime);
83             $fclose(outfile);
84             q <= 1'bX; // Set all outputs to unknown
85         end
86         if (errorsignal_clk == 0)
87         begin
88             case (cell_state)
89             0: begin
90                 errorsignal_a = 1; // Critical timing on this input; assign
91                 ↪ immediately
92                 errorsignal_a <= #(ct_state0_clk_a) 0; // Clear error signal
93                 ↪ after critical timing expires
94             end
95             1: begin
96                 q <= #(delay_state1_clk_q) !q;
97                 cell_state = 0; // Blocking statement -- immediately
98             end
99         endcase
100    end
101 endmodule

```

Listing 4.30: RSFQ DFF verilog model.

The digital simulation results for the RSFQ DFF is shown in Fig. 4.58 and the Mealy finite state machine diagram, extracted using *TimEx*, is shown in Fig. 4.59.

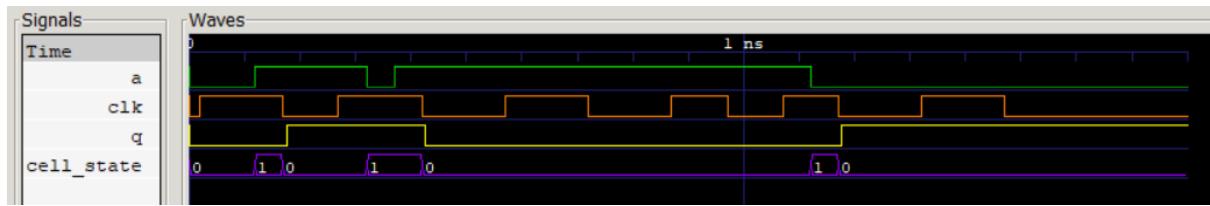


Figure 4.58: RSFQ DFF digital simulation results.

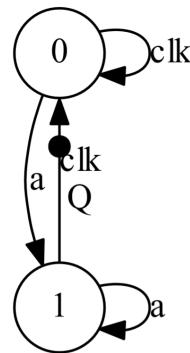


Figure 4.59: RSFQ DFF Mealy finite state machine diagram.

Power Consumption

Table 4.30: RSFQ DFF power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	1815	2.84
2	1815	5.67
5	1815	14.2
10	1815	28.4
20	1815	56.7
50	1815	142

4.3.2 NDRO

The NDRO, non-destructive readout, cell is a memory device controlled by a set, reset and clock input signal. When an input set signal is received, the NDRO will generate an output pulse after each clock signal until an input reset signal is received. The NDRO cell is not designed to be directly connected to a PTL.

Schematic

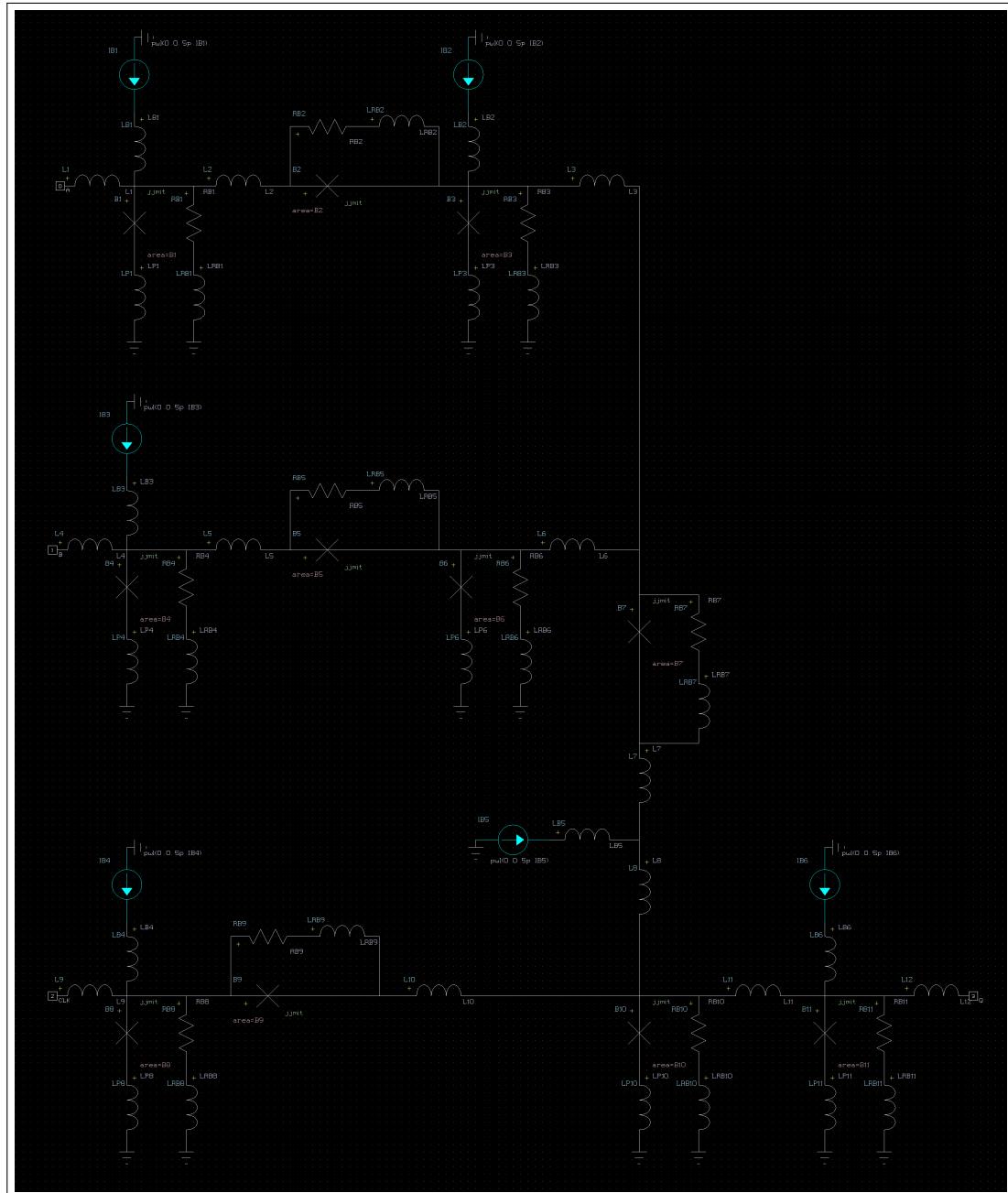


Figure 4.60: Schematic of RSFQ NDRO.

Layout

The physical layout for the RSFQ NDRO is shown in Fig. 4.61. The layout height is $70 \mu m$ and the width is $70 \mu m$. The cell includes an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

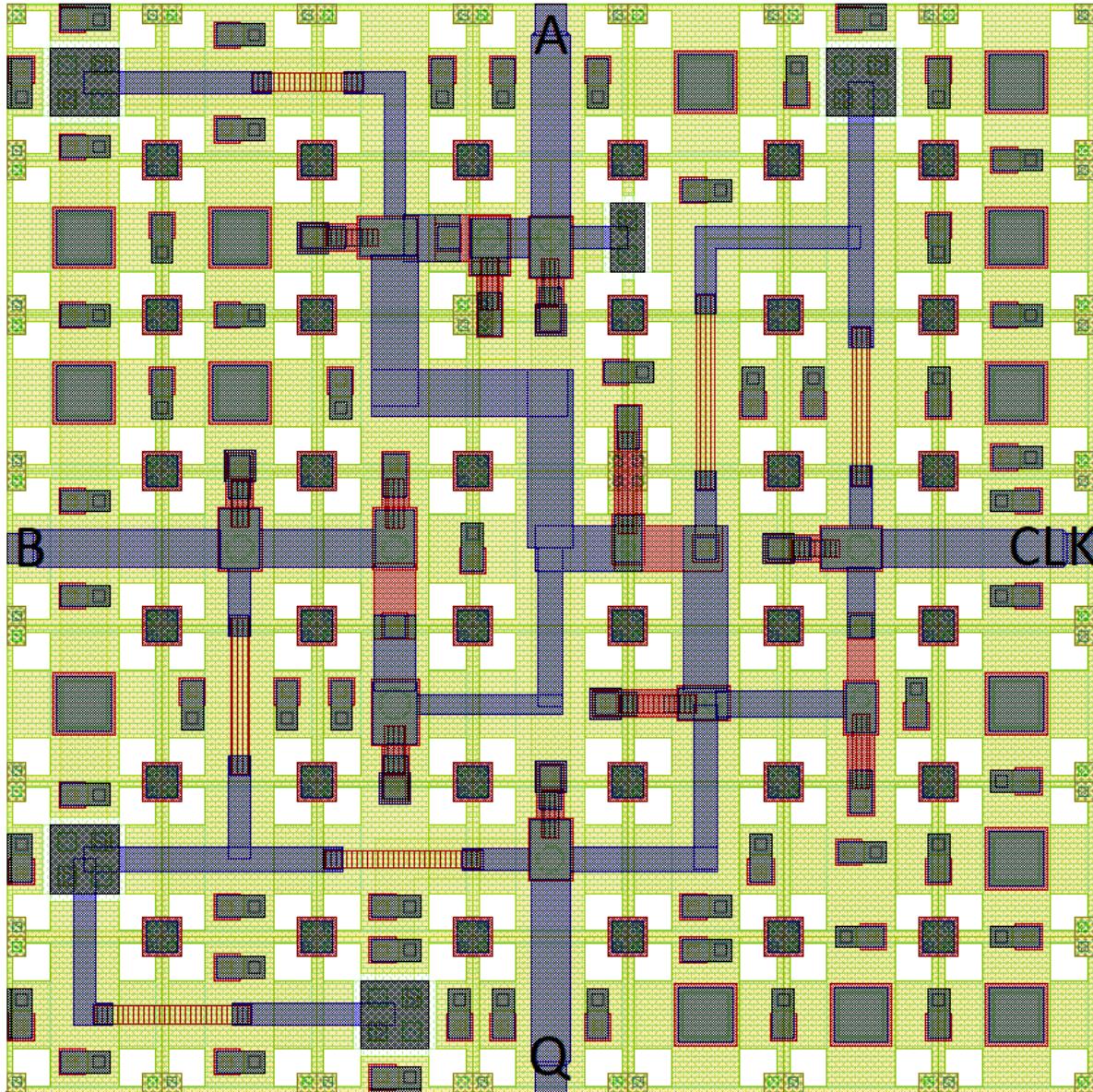


Figure 4.61: RSFQ NDRO Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param RB11=B0Rs/B11
2   ↪ by InductEx v.6.0.4 on 15-4-21.          65  .param LRB1=(RB1/Rsheet)*Lsheet+LP
3  * Author: L. Schindler                      66  .param LRB2=(RB2/Rsheet)*Lsheet+LP
4  * Version: 2.1                               67  .param LRB3=(RB3/Rsheet)*Lsheet+LP
5  * Last modification date: 9 June 2021       68  .param LRB4=(RB4/Rsheet)*Lsheet+LP
6  * Last modification by: L. Schindler        69  .param LRB5=(RB5/Rsheet)*Lsheet+LP
7  *$Ports a b clk q                         70  .param LRB6=(RB6/Rsheet)*Lsheet+LP
8  .subckt LSmitll_NDRO a b clk q           71  .param LRB7=(RB7/Rsheet)*Lsheet+LP
9  .model jjmit jj(rtype=1, vg=2.8mV, cap    72  .param LRB8=(RB8/Rsheet)*Lsheet+LP
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA     73  .param LRB9=(RB9/Rsheet)*Lsheet+LP
   ↪ )                                         74  .param LRB10=(RB10/Rsheet)*Lsheet+LP
10 .param Phi0=2.067833848E-15                75  .param LRB11=(RB11/Rsheet)*Lsheet+LP
11 .param B0=1                                 76
12 .param Ic0=0.0001                           77
13 .param IcRs=100u*6.859904418               78  .param LB1=LB
14 .param B0Rs=IcRs/Ic0*B0                   79  .param LB2=LB
15 .param Rsheet=2                            80  .param LB3=LB
16 .param Lsheet=1.13e-12                     81  .param LB4=LB
17 .param LP=0.2p                            82  .param LB5=LB
18 .param IC=2.5                            83  .param LB6=LB
19 .param LB=2p                             84
20 .param BiasCoef=0.7                       85  B1 1 2 jjmit area=B1
21
22 .param B1=2.5                            86  B2 3 4 jjmit area=B2
23 .param B2=1.99                           87  B3 4 5 jjmit area=B3
24 .param B3=2.2                            88  B4 7 8 jjmit area=B4
25 .param B4=2.5                            89  B5 9 10 jjmit area=B5
26 .param B5=2.35                           90  B6 10 11 jjmit area=B6
27 .param B6=3.24                           91  B7 6 12 jjmit area=B7
28 .param B7=0.74                           92  B8 14 15 jjmit area=B8
29 .param B8=2.5                            93  B9 14 16 jjmit area=B9
30 .param B9=1.17                           94  B10 17 18 jjmit area=B10
31 .param B10=1.09                          95  B11 19 20 jjmit area=B11
32 .param B11=2.5                           96
33
34 .param IB1=175u                           97  IB1 0 21 pw1(0 0 5p IB1)
35 .param IB2=271u                           98  IB2 0 22 pw1(0 0 5p IB2)
36 .param IB3=175u                           99  IB3 0 23 pw1(0 0 5p IB3)
37 .param IB4=175u                           100 IB4 0 24 pw1(0 0 5p IB4)
38 .param IB5=136u                           101 IB5 0 25 pw1(0 0 5p IB5)
39 .param IB6=175u                           102 IB6 0 26 pw1(0 0 5p IB6)
40
41 .param L1=Phi0/(4*IC*Ic0)                 103
42 .param L2=Phi0/(2*B1*Ic0)                 104  LB1 1 21 LB1
43 .param L3=Phi0/(2*B3*Ic0)                 105  LB2 4 22 LB2
44 .param L4=Phi0/(4*IC*Ic0)                 106  LB3 7 23 LB3
45 .param L5=Phi0/(2*B4*Ic0)                 107  LB4 14 24 LB4
46 .param L6=Phi0/(2*B6*Ic0)                 108  LB5 13 25 LB5
47 .param L7=1p                             109  LB6 19 26 LB6
48 .param L8=1p                             110
49 .param L9=Phi0/(4*IC*Ic0)                 111  L1 a 1 2.067E-12
50 .param L10=Phi0/(2*B8*Ic0)                112  L2 1 3 9.786E-13
51 .param L11=Phi0/(2*B10*Ic0)               113  L3 4 6 2.693E-12
52 .param L12=Phi0/(4*IC*Ic0)                114  L4 b 7 2.068E-12
53
54 .param RB1=B0Rs/B1                        115  L5 7 9 2.319E-12
55 .param RB2=B0Rs/B2                        116  L6 10 6 3.438E-12
56 .param RB3=B0Rs/B3                        117  L7 12 13 1.12E-12
57 .param RB4=B0Rs/B4                        118  L8 13 17 1.028E-12
58 .param RB5=B0Rs/B5                        119  L9 clk 14 2.106E-12
59 .param RB6=B0Rs/B6                        120  L10 16 17 3.135E-12
60 .param RB7=B0Rs/B7                        121  L11 17 19 3.223E-12
61 .param RB8=B0Rs/B8                        122  L12 19 q 2.071E-12
62 .param RB9=B0Rs/B9                        123
63 .param RB10=B0Rs/B10                       124  LP1 2 0 5.174E-13
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

```

```

130 | LP11 20 0 4.832E-13
131 | RB1 1 101 RB1
132 | LRB1 101 0 LRB1
133 | RB2 3 103 RB2
134 | LRB2 103 4 LRB2
135 | RB3 4 104 RB3
136 | LRB3 104 0 LRB3
137 | RB4 7 107 RB4
138 | LRB4 107 0 LRB4
139 | RB5 9 109 RB5
140 | LRB5 109 10 LRB5
141 | RB6 10 110 RB6
142 | LRB6 110 0 LRB6
144 | RB7 6 106 RB7
145 | LRB7 106 12 LRB7
146 | RB8 14 114 RB8
147 | LRB8 114 0 LRB8
148 | RB9 14 116 RB9
149 | LRB9 116 16 LRB9
150 | RB10 17 117 RB10
151 | LRB10 117 0 LRB10
152 | RB11 19 119 RB11
153 | LRB11 119 0 LRB11
154 | .ends

```

Listing 4.31: RSFQ NDRO JoSIM netlist.**Table 4.31:** RSFQ NDRO pin list.

Pin	Description
a	Data input (set signal)
b	Data input (reset signal)
clk	Clock input
q	Data output

The simulation results for the RSFQ NDRO using JoSIM is shown in Fig. 4.62. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a** (set signal),
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b** (reset signal),
- (d) the phase over the input JJ of pin **b**,
- (e) the current through the input inductor connected to pin **clk**,
- (f) the phase over the input JJ of pin **clk**,
- (g) the phase over the output JJ of pin **q**, and
- (h) the phase over the input JJ of the load circuit connected to pin **q**.

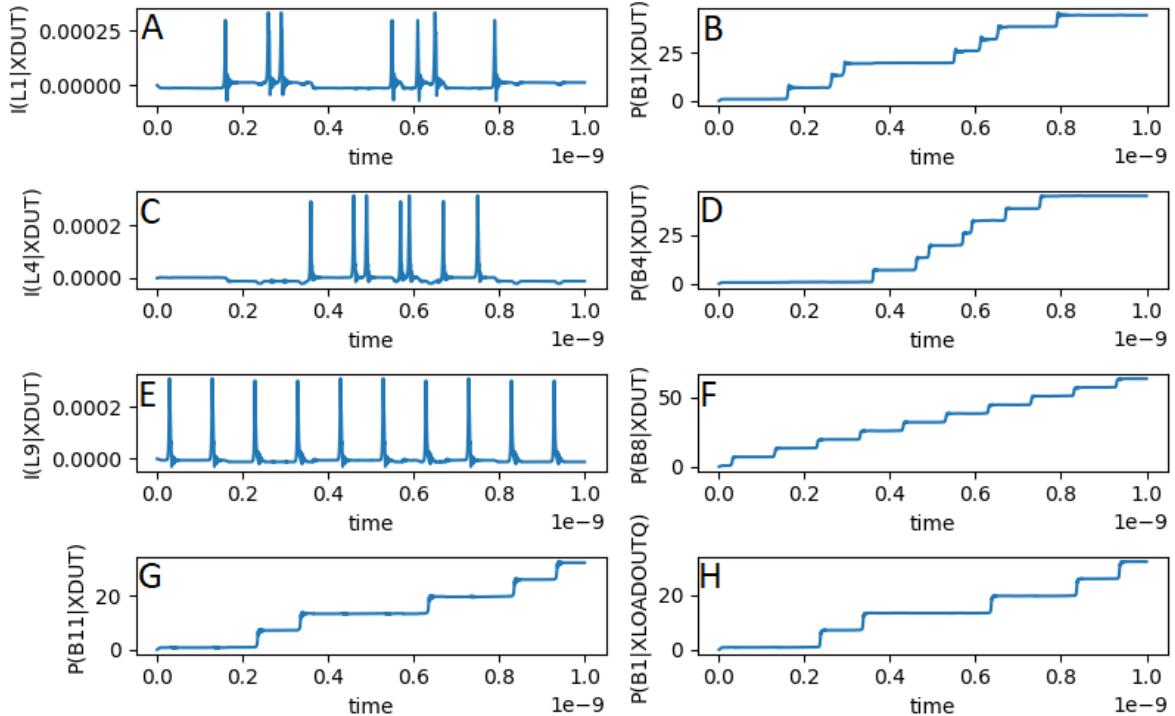


Figure 4.62: RSFQ NDRO analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 11 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmit11_NDRO_v2p1_optimized (a, b, clk, q);
9
10 input
11     a, b, clk;
12
13 output
14     q;
15
16 reg
17     q;
18
19 real
20     delay_state1_clk_q = 5.0,
21     ct_state0_b_a = 1.9,
22     ct_state1_a_b = 1.5,
23     ct_state1_clk_clk = 6.9;
24
25 reg
26     errorsignal_a,
27     errorsignal_b,
28     errorsignal_clk;
29
30 integer
31     outfile,
32     cell_state; // internal state of the cell
33
34 initial
35 begin
36     errorsignal_a = 0;
37     errorsignal_b = 0;
38     errorsignal_clk = 0;
39     cell_state = 0; // Startup state
40     q = 0; // All outputs start at 0
41 end
42
43 always @(posedge a or negedge a) // execute at positive and negative edges of input
44 begin
45     if ($time>4) // arbitrary steady-state time)
46         begin
47             if (errorsignal_a == 1'b1) // A critical timing is active for this input
48                 begin
49                     outfile = $fopen("errors.txt", "a");
50                     $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m; %0d_ps.\n"
51                                     ↪ ", $time);
52                     $fclose(outfile);
53                     q <= 1'bX; // Set all outputs to unknown
54                 end
55             if (errorsignal_a == 0)
56                 begin
57                     case (cell_state)
58                         0: begin
59                             cell_state = 1; // Blocking statement -- immediately
60                         end
61                         1: begin
62                             errorsignal_b = 1; // Critical timing on this input; assign
63                                         ↪ immediately
64                             errorsignal_b <= #(ct_state1_a_b) 0; // Clear error signal
65                                         ↪ after critical timing expires
66                         end
67                     endcase
68                 end
69             end
70         end
71     end
72 end

```

```

65           end
66       end
67   end
68
69 always @(posedge b or negedge b) // execute at positive and negative edges of input
70   begin
71     if ($time>4) // arbitrary steady-state time)
72     begin
73       if (errorsignal_b == 1'b1) // A critical timing is active for this input
74       begin
75         outfile = $fopen("errors.txt", "a");
76         $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
77                     ↪ ", $stime);
78         $fclose(outfile);
79         q <= 1'bX; // Set all outputs to unknown
80       end
81     if (errorsignal_b == 0)
82     begin
83       case (cell_state)
84         0: begin
85           errorsignal_a = 1; // Critical timing on this input; assign
86           ↪ immediately
87           errorsignal_a <= #(ct_state0_b_a) 0; // Clear error signal
88           ↪ after critical timing expires
89         end
90       1: begin
91         cell_state = 0; // Blocking statement -- immediately
92       end
93     endcase
94   end
95
96 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
97   begin
98     if ($time>4) // arbitrary steady-state time)
99     begin
100       if (errorsignal_clk == 1'b1) // A critical timing is active for this input
101       begin
102         outfile = $fopen("errors.txt", "a");
103         $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
104                     ↪ ", $stime);
105         $fclose(outfile);
106         q <= 1'bX; // Set all outputs to unknown
107       end
108     if (errorsignal_clk == 0)
109     begin
110       case (cell_state)
111         0: begin
112           end
113         1: begin
114           q <= #(delay_state1_clk_q) !q;
115           errorsignal_clk = 1; // Critical timing on this input; assign
116           ↪ immediately
117           errorsignal_clk <= #(ct_state1_clk_clk) 0; // Clear error
118           ↪ signal after critical timing expires
119         end
120       endcase
121     end
122   end
123 endmodule

```

Listing 4.32: RSFQ NDRO verilog model.

The digital simulation results for the RSFQ NDRO is shown in Fig. 4.63 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 4.64.

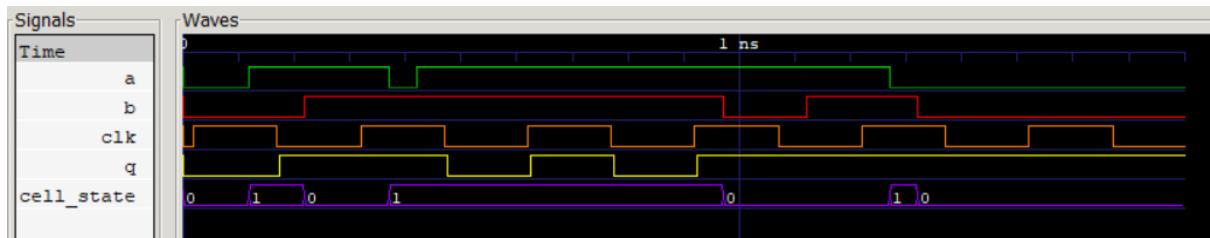


Figure 4.63: RSFQ NDRO digital simulation results.

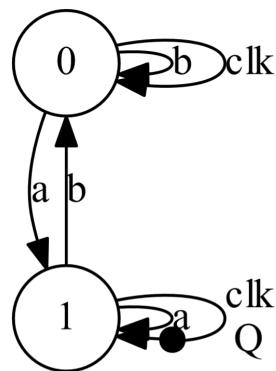


Figure 4.64: RSFQ NDRO Mealy finite state machine diagram.

Power Consumption

Table 4.32: RSFQ NDRO power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2878	4.71
2	2878	9.42
5	2878	23.6
10	2878	47.1
20	2878	94.2
50	2878	236

4.3.3 BUFF

The RSFQ BUFF cell is a buffer cell intended for clock balancing. It is designed to have the same a-to-q delay as the SPLIT cell. The BUFF does not have integrated PTL transmitters and receivers and connecting the cell directly to a PTL is not recommended.

Schematic

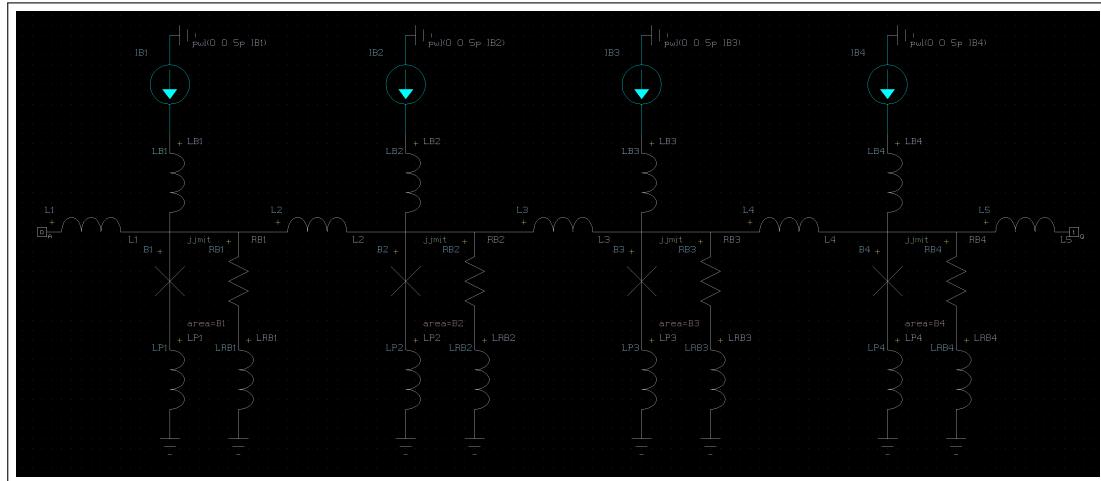


Figure 4.65: Schematic of RSFQ BUFF.

Layout

The physical layout for the RSFQ BUFF is shown in Fig. 4.66. The layout height is $70 \mu m$ and the width is $50 \mu m$. The cell includes an integrated bias line on M0 within the top row of track blocks. Bias pillars from M0 to M6 connect the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

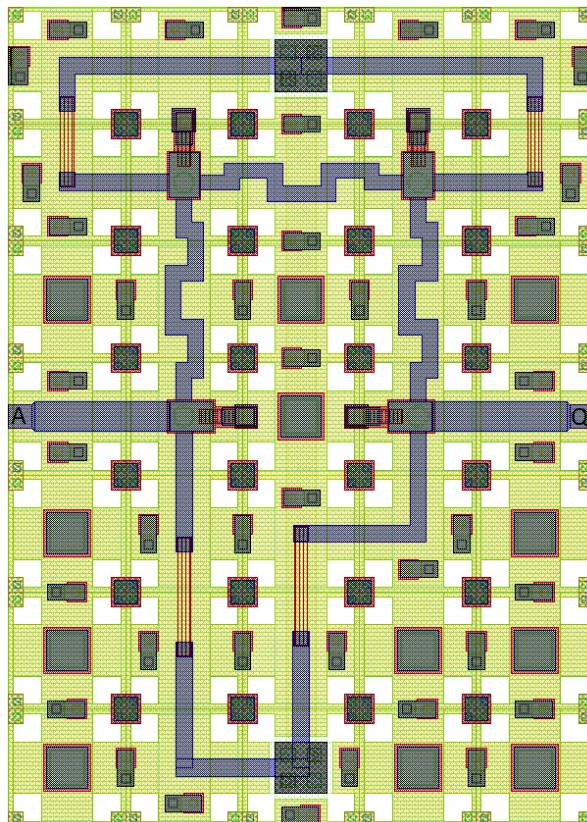


Figure 4.66: RSFQ BUFF Layout.

Analog model

```

1  * Back-annotated simulation file written      45 | .param LRB2=(RB2/Rsheet)*Lsheet+LP
2   ↪ by InductEx v.6.0.4 on 11-6-21.          46 | .param LRB3=(RB3/Rsheet)*Lsheet+LP
3  * Author: L. Schindler                      47 | .param LRB4=(RB4/Rsheet)*Lsheet+LP
4  * Version: 2.1                               48 | .param LP1=LP
5  * Last modification date: 11 June 2021       49 | .param LP2=LP
6  * Last modification by: L. Schindler         50 | .param LP3=LP
7  *$Ports a q                                51 | .param LP4=LP
8 .subckt LSMITLL_BUFF a q                     52 |
9 .model jjmit jj(rtype=1, vg=2.8mV, cap      53 | B1 1 2 jjmit area=B1
10    ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    54 | B2 4 5 jjmit area=B2
11    ↪ )                                         55 | B3 7 8 jjmit area=B3
12 .param Phi0=2.067833848E-15                 56 | B4 10 11 jjmit area=B4
13 .param B0=1                                  57 |
14 .param Ic0=0.0001                            58 | IB1 0 3 pwl(0 0 5p IB1)
15 .param IcRs=100u*6.859904418                  59 | IB2 0 6 pwl(0 0 5p IB2)
16 .param B0Rs=IcRs/Ic0*B0                      60 | IB3 0 9 pwl(0 0 5p IB3)
17 .param Rsheet=2                             61 | IB4 0 12 pwl(0 0 5p IB4)
18 .param Lsheet=1.13e-12                         62 |
19 .param LP=0.2p                               63 | L1 a 1 2.057E-12
20 .param IC=2.5                               64 | L2 1 4 4.026E-12
21 .param Lptl=2p                             65 | L3 4 7 4.155E-12
22 .param LB=2p                               66 | L4 7 10 4.057E-12
23 .param BiasCoef=0.7                         67 | L5 10 q 2.057E-12
24 .param B1=IC                                68 |
25 .param B2=IC                                69 | LP1 2 0 5.283E-13
26 .param B3=IC                                70 | LP2 5 0 5.327E-13
27 .param B4=IC                                71 | LP3 8 0 5.084E-13
28 .param IB1=B1*Ic0*BiasCoef                72 | LP4 11 0 5.269E-13
29 .param IB2=B2*Ic0*0.95                      73 |
30 .param IB3=B3*Ic0*0.95                      74 | LB1 1 3 LB1
31 .param IB4=B4*Ic0*BiasCoef                75 | LB2 4 6 LB2
32 .param LB1=LB                               76 | LB3 7 9 LB3
33 .param LB2=LB                               77 | LB4 10 12 LB4
34 .param LB3=LB                               78 |
35 .param LB4=LB                               79 | RB1 1 101 RB1
36 .param L1=Phi0/(4*IC*Ic0)                   80 | RB2 4 104 RB2
37 .param L2=Phi0/(2*B1*Ic0)                   81 | RB3 7 107 RB3
38 .param L3=Phi0/(2*B2*Ic0)                   82 | RB4 10 110 RB4
39 .param L4=Phi0/(2*B3*Ic0)                   83 |
40 .param RB1=B0Rs/B1                         84 | LRB1 101 0 LRB1
41 .param RB2=B0Rs/B2                         85 | LRB2 104 0 LRB2
42 .param RB3=B0Rs/B3                         86 | LRB3 107 0 LRB3
43 .param RB4=B0Rs/B4                         87 | LRB4 110 0 LRB4
44 .param LRB1=(RB1/Rsheet)*Lsheet+LP          88 | .ends

```

Listing 4.33: RSFQ BUFF JoSIM netlist.

Table 4.33: RSFQ BUFF pin list.

Pin	Description
a	Data input
q	Data output

The simulation results for the RSFQ BUFF using JoSIM is shown in Fig. 4.67. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the phase over the output JJ of pin **q**, and
- (d) the phase over the input JJ of the load circuit connected to pin **q**.

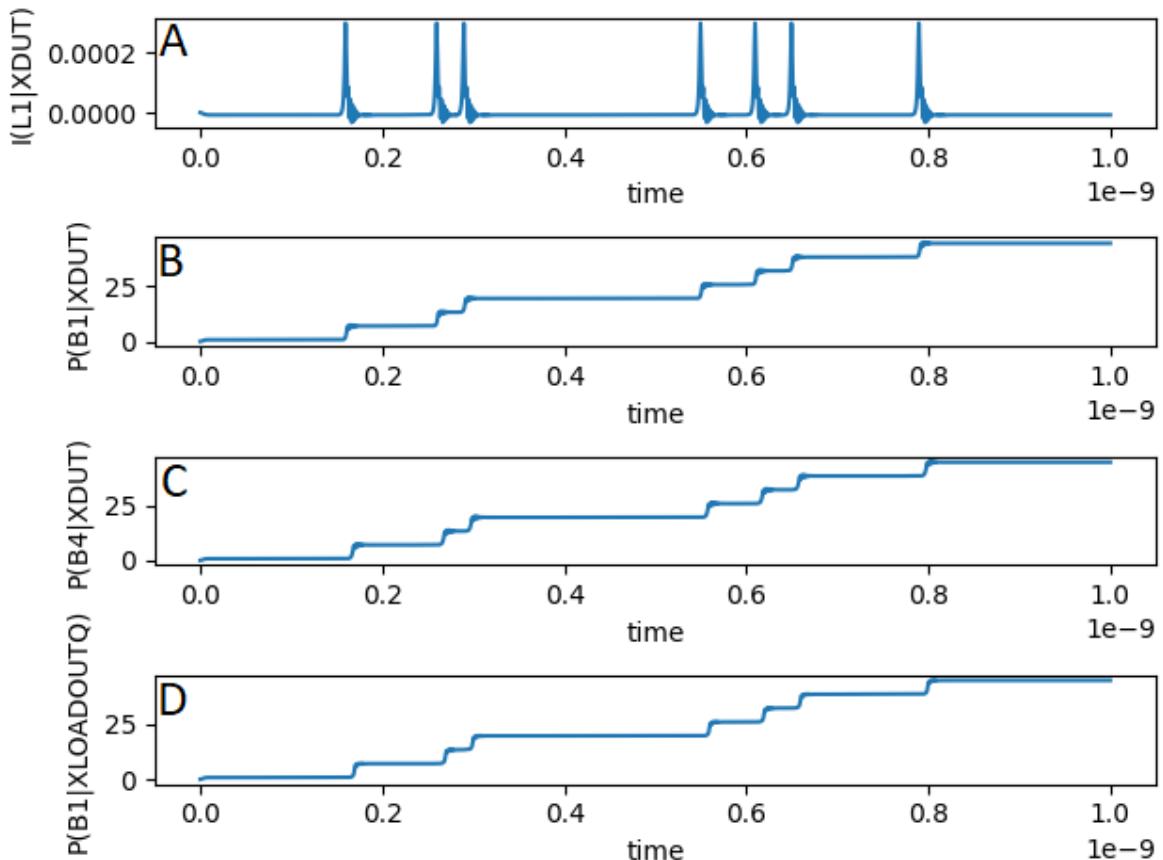


Figure 4.67: RSFQ BUFF analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 11 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmit11_BUFF_v2p1_annotated (a, q);
9
10 input
11   a;
12
13 output
14   q;
15
16 reg
17   q;
18
19 real
20   delay_state0_a_q = 6.8,
21   ct_state0_a_a = 3.4;
22
23 reg
24   errorsignal_a;
25
26 integer
27   outfile,
28   cell_state; // internal state of the cell
29
30 initial
31 begin
32   errorsignal_a = 0;
33   cell_state = 0; // Startup state
34   q = 0; // All outputs start at 0
35 end
36
37 always @(posedge a or negedge a) // execute at positive and negative edges of input
38 begin
39   if ($time>4) // arbitrary steady-state time)
40     begin
41       if (errorsignal_a == 1'b1) // A critical timing is active for this input
42         begin
43           outfile = $fopen("errors.txt", "a");
44           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
45           ↪ ", $stime);
46           $fclose(outfile);
47           q <= 1'bX; // Set all outputs to unknown
48         end
49       if (errorsignal_a == 0)
50         begin
51           case (cell_state)
52             0: begin
53               q <= #(delay_state0_a_q) !q;
54               errorsignal_a = 1; // Critical timing on this input; assign
55               ↪ immediately
56               errorsignal_a <= #(ct_state0_a_a) 0; // Clear error signal
57               ↪ after critical timing expires
58             end
59           endcase
60         end
61     end
62   end
63 endmodule

```

Listing 4.34: RSFQ BUFF verilog model.

The digital simulation results for the RSFQ BUFF is shown in Fig. 4.68 and the Mealy finite state machine diagram, extracted using *TimEx*, is shown in Fig. 4.69.

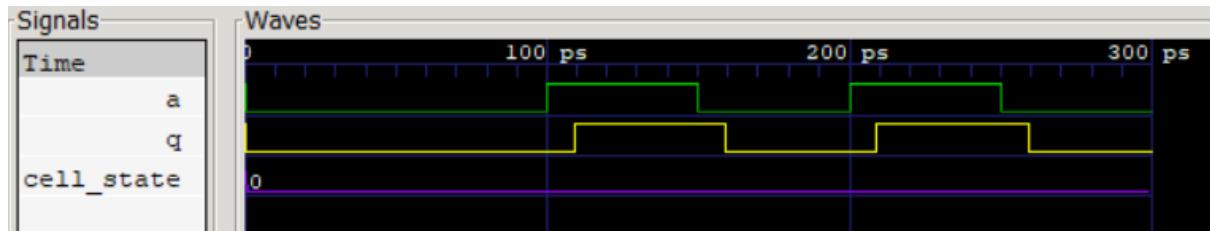


Figure 4.68: RSFQ BUFF digital simulation results.

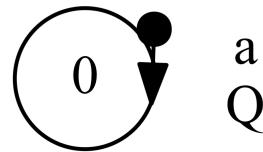


Figure 4.69: RSFQ BUFF Mealy finite state machine diagram.

Power Consumption

Table 4.34: RSFQ BUFF power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2132	2.07
2	2132	4.14
5	2132	10.3
10	2132	20.7
20	2132	41.4
50	2132	103

4.4 Interface cells

4.4.1 DCSFQ

The RSFQ DCSFQ is an interface cell designed to convert input voltage pulses into SFQ pulses. The DCSFQ does not have an integrated PTL transmitter and is not intended to connect directly to a PTL output.

Schematic

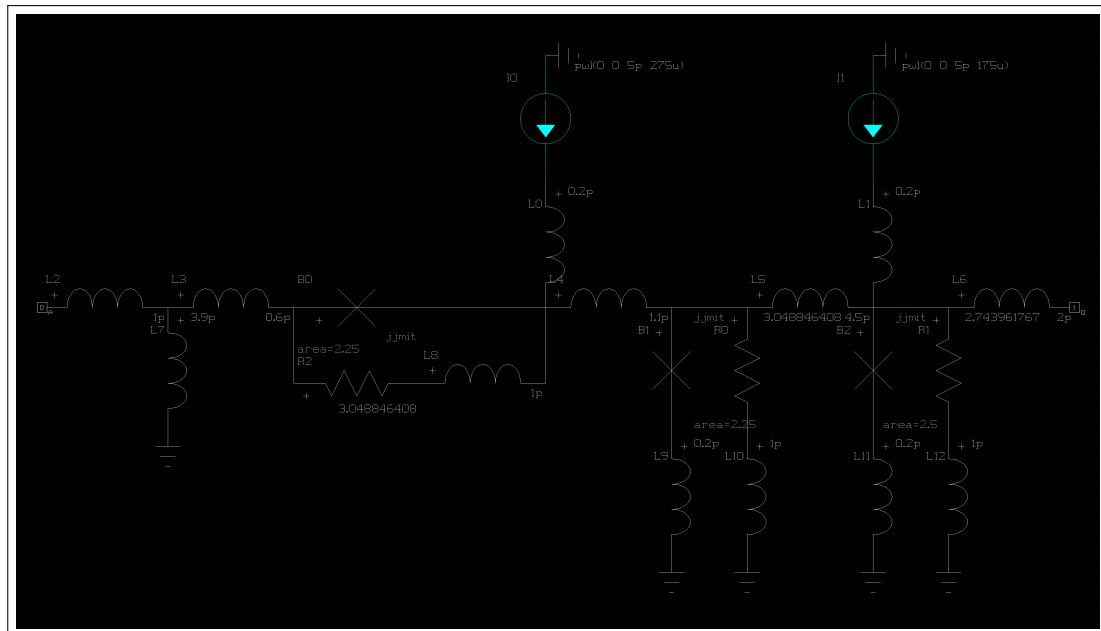


Figure 4.70: Schematic of RSFQ DCSFQ.

Layout

The physical layout for the RSFQ DCSFQ is shown in Fig. 4.71. The layout height is $70 \mu m$ and the width is $30 \mu m$. The cell includes an integrated bias line on M0 within the top row of track blocks. A bias pillar from M0 to M6 connects the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

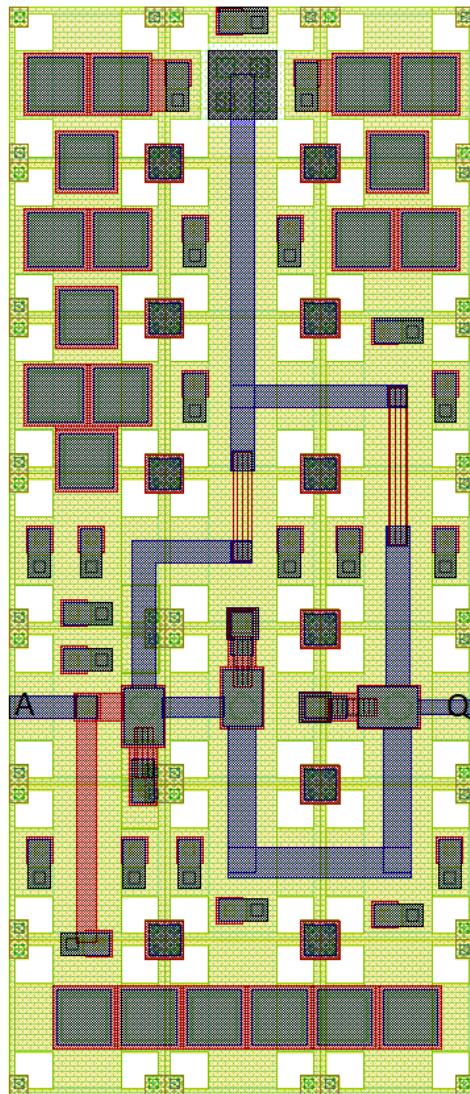


Figure 4.71: RSFQ DCSFQ Layout.

Analog model

```

1 * Back-annotated simulation file written by InductEx v.6.0.4 on 2-6-21.
2 * Author: L. Schindler
3 * Version: 2.1
4 * Last modification date: 2 June 2021
5 * Last modification by: L. Schindler
6
7 *$Ports a q
8 .subckt LSmitll_DCSFQ a q
9 .model jjm1 jj(rtype=1, vg=2.8mV, cap=0.07pF, r0=160, rn=16, icrit=0.1mA)
10 .param B0=1.0
11 .param Ic0=0.0001
12 .param IcRs=100u*6.859904418
13 .param B0Rs=IcRs/Ic0*B0
14 .param Rsheet=2
15 .param Lsheet=1.13e-12
16 .param LB=2p
17 .param LP=0.2p
18
19 .param B1=2.25
20 .param B2=2.25
21 .param B3=2.5
22 .param L1=1p
23 .param L2=3.9p
24 .param L3=0.6p
25 .param L4=1.1p
26 .param L5=4.5p
27 .param L6=2p
28 .param IB1=275u
29 .param IB2=175u
30 .param LB1=LB
31 .param LB2=LB
32 .param LP2=LP
33 .param LP3=LP
34 .param RB1=B0Rs/B1
35 .param RB2=B0Rs/B2
36 .param RB3=B0Rs/B3
37 .param LRB1=(RB1/Rsheet)*Lsheet
38 .param LRB2=(RB2/Rsheet)*Lsheet
39 .param LRB3=(RB3/Rsheet)*Lsheet
40
41 B1 2 3 jjmit area=B1
42 B2 5 6 jjmit area=B2
43 B3 7 8 jjmit area=B3
44 IB1 0 4 pw1(0 0 5p IB1)
45 IB2 0 9 pw1(0 0 5p IB2)
46 LB1 3 4 3.12E-12
47 LB2 7 9 2.281E-12
48 L1 a 1 1.295E-12
49 L2 1 0 3.904E-12
50 L3 1 2 5.998E-13
51 L4 3 5 1.095E-12
52 L5 5 7 4.448E-12
53 L6 7 q 1.334E-12
54 LP2 6 0 5.16E-13
55 LP3 8 0 4.733E-13
56 RB1 2 102 RB1
57 LRB1 102 3 LRB1
58 RB2 5 105 RB2
59 LRB2 105 0 LRB2
60 RB3 7 107 RB3
61 LRB3 107 0 LRB3
62 .ends

```

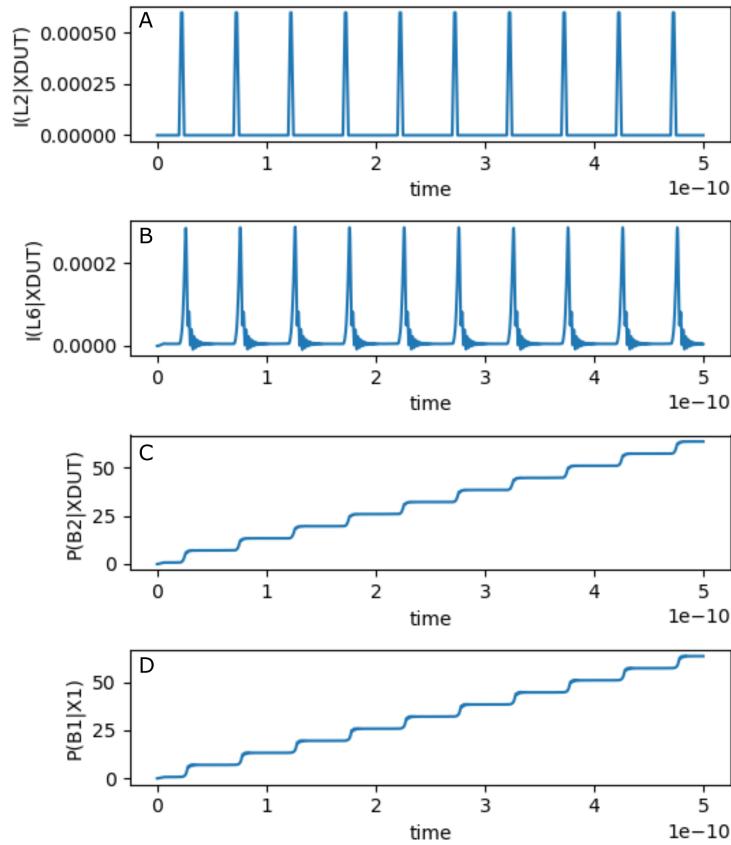
Listing 4.35: RSFQ DCSFQ JoSIM netlist.

Table 4.35: RSFQ DCSFQ pin list.

Pin	Description
a	Data input
q	Data output

The simulation results for the RSFQ DCSFQ using JoSIM is shown in Fig. 4.72. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the current through the output inductor connected to pin **q**,
- (c) the phase over the output JJ of pin **q**, and
- (d) the phase over the input JJ of the load circuit connected to pin **q**.

**Figure 4.72:** RSFQ DCSFQ analog simulation results.

4.4.2 SFQDC

The RSFQ SFQDC is an interface cell designed to convert SFQ pulses to an output voltage level which can be measured by standard equipment. The SFQDC does not have an integrated PTL receiver and is not intended to be connected directly to a PTL input.

Schematic

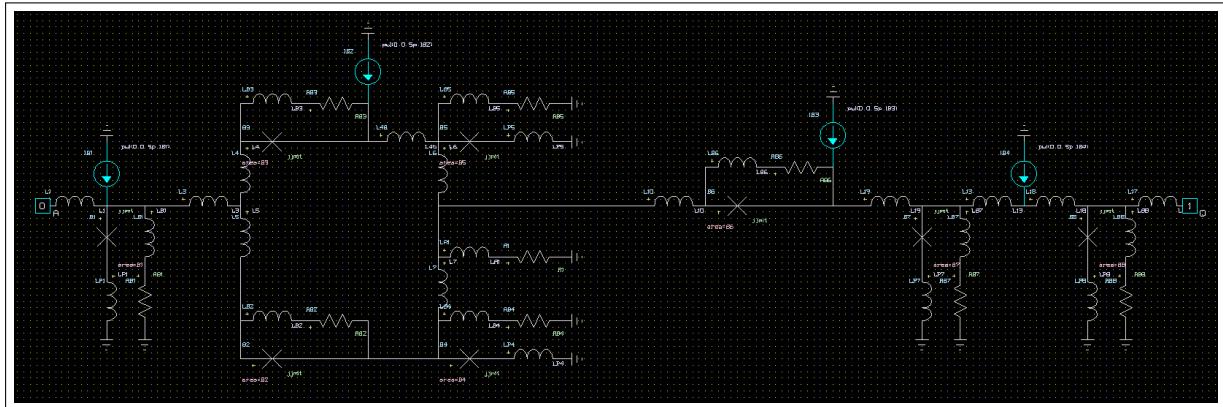


Figure 4.73: Schematic of RSFQ SFQDC.

Layout

The physical layout for the RSFQ SFQDC is shown in Fig. 4.74. The layout height is $70 \mu m$ and the width is $60 \mu m$. The cell includes an integrated bias line on M0 within the top row of track blocks. A bias pillar from M0 to M6 connects the circuit to the integrated bias line. All bias resistors are designed to be connected to a 2.6 mV voltage source.

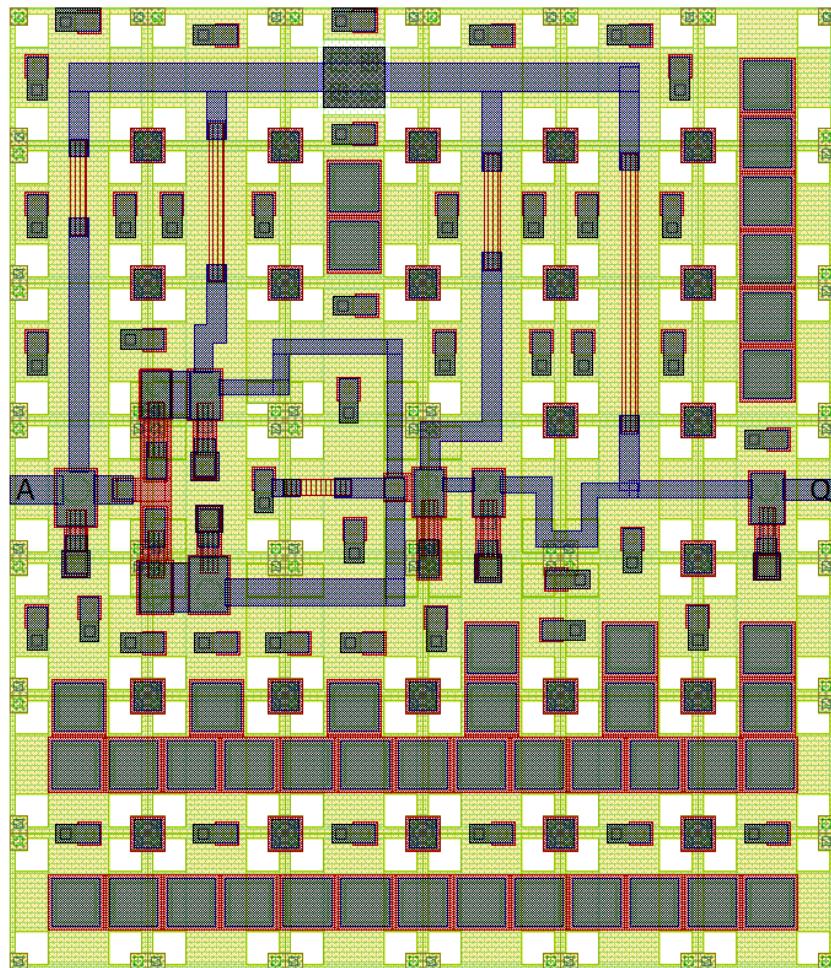


Figure 4.74: RSFQ SFQDC Layout.

Analog model

```

1  * Adapted from Fluxonics SDQDC_v5
2  * Author: L. Schindler
3  * Version: 2.1
4  * Last modification date: 27 April 2021
5  * Last modification by: L. Schindler
6
7  *$Ports a q
8 .subckt LSmitll_SFQDC a q
9 .model jjmit jj(rtype=1, vg=2.8mV, cap
  ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA
  ↪ )
10 .param B0=1
11 .param Ic0=0.0001
12 .param IcRs=100u*6.859904418
13 .param B0Rs=IcRs/Ic0*B0
14 .param Rsheet=2
15 .param Lsheet=1.13e-12
16 .param B1=3.25
17 .param B2=2.00
18 .param B3=1.50
19 .param B4=3.00
20 .param B5=1.75
21 .param B6=1.50
22 .param B7=1.50
23 .param B8=2.00
24 .param L1=1.522p
25 .param L3=0.827p
26 .param L4=1.12884p
27 .param L5=1.11098p
28 .param L6=5.940p
29 .param L7=3.216p
30 .param L10=0.215p
31 .param L13=3.699p
32 .param L17=1.510p
33 .param L18=2.010p
34 .param L19=0.954p
35 .param L4b=0.178p
36 .param LRB1=(RB1/Rsheet)*Lsheet
37 .param LRB2=(RB2/Rsheet)*Lsheet
38 .param LRB3=(RB3/Rsheet)*Lsheet
39 .param LRB4=(RB4/Rsheet)*Lsheet
40 .param LRB5=(RB5/Rsheet)*Lsheet
41 .param LRB6=(RB6/Rsheet)*Lsheet
42 .param LRB7=(RB7/Rsheet)*Lsheet
43 .param LRB8=(RB8/Rsheet)*Lsheet
44 .param LP1=0.140p
45 .param LP4=0.524p
46 .param LP5=0.516p
47 .param LP7=0.086p
48 .param LP8=0.226p
49 .param LR1=0.91p
50 .param R1=0.375
51 .param RB1=B0Rs/B1
52 .param RB2=B0Rs/B2
53 .param RB3=B0Rs/B3
54 .param RB4=B0Rs/B4
55 .param RB5=B0Rs/B5
56 .param RB6=B0Rs/B6
57 .param RB7=B0Rs/B7
58 .param RB8=B0Rs/B8
59 .param IB1=280u
60 .param IB2=150u
61 .param IB3=220u
62 .param IB4=80u
63 B1 8 20 jjmit area=B1
64 B2 12 13 jjmit area=B2
65 B3 3 4 jjmit area=B3
66 B4 13 29 jjmit area=B4
67 B5 5 16 jjmit area=B5
68 B6 6 7 jjmit area=B6
69 B7 10 22 jjmit area=B7
70 B8 11 24 jjmit area=B8
71 IB1 0 8 pw1(0 0 5p IB1)
72 IB2 0 4 pw1(0 0 5p IB2)
73 IB3 0 7 pw1(0 0 5p IB3)
74 IB4 0 18 pw1(0 0 5p IB4)
75 L1 a 8 L1
76 L3 8 17 L3
77 L4 3 17 L4
78 L5 17 12 L5
79 L6 5 9 L6
80 L7 9 13 L7
81 L10 9 6 L10
82 L13 10 18 L13
83 L17 11 q L17
84 L18 18 11 L18
85 L19 7 10 L19
86 L4b 4 5 L4b
87 LRB1 8 21 LRB1
88 LRB2 12 27 LRB2
89 LRB3 3 14 LRB3
90 LRB4 13 28 LRB4
91 LRB5 5 15 LRB5
92 LRB6 6 19 LRB6
93 LRB7 10 23 LRB7
94 LRB8 11 25 LRB8
95 LP1 20 0 LP1
96 LP4 29 0 LP4
97 LP5 16 0 LP5
98 LP7 22 0 LP7
99 LP8 24 0 LP8
100 LR1 9 26 LR1
101 R1 26 0 R1
102 RB1 21 0 RB1
103 RB2 27 13 RB2
104 RB3 14 4 RB3
105 RB4 28 0 RB4
106 RB5 15 0 RB5
107 RB6 19 7 RB6
108 RB7 23 0 RB7
109 RB8 25 0 RB8
110 .ends

```

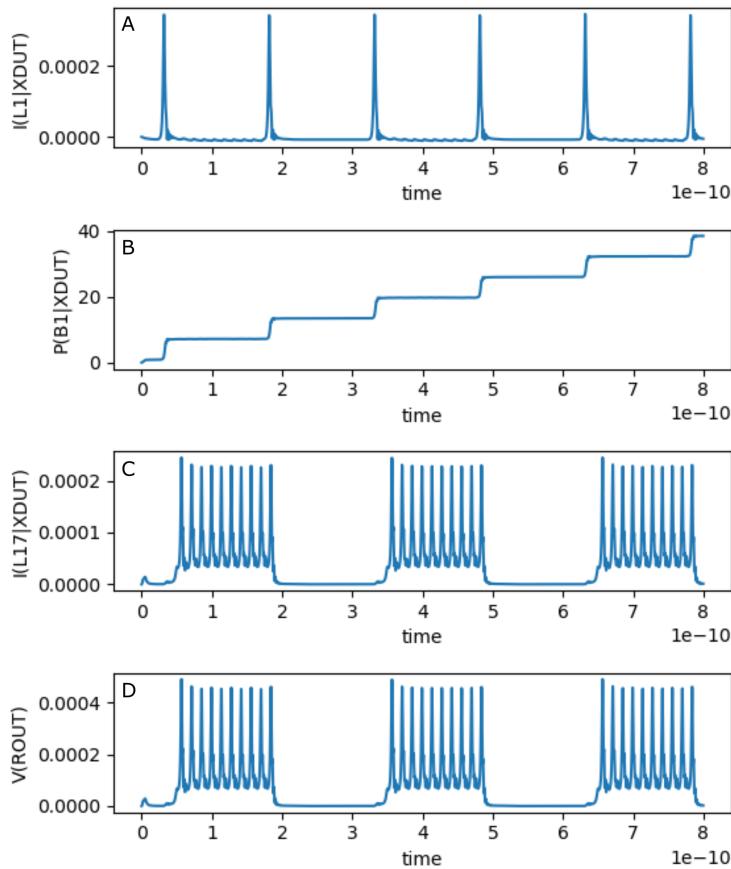
Listing 4.36: RSFQ SFQDC JoSIM netlist.

Table 4.36: RSFQ SFQDC pin list.

Pin	Description
a	Data input
q	Data output

The simulation results for the RSFQ SFQDC using JoSIM is shown in Fig. 4.75. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the output inductor connected to pin **q**, and
- (c) the voltage over the load resistor connected to pin **q**.

**Figure 4.75:** RSFQ SFQDC analog simulation results.

5. RSFQ Cell Library: PTL Connections

5.1 Interconnects

5.1.1 JTLT

The JTLT, Josephson transmission line, cell is commonly used to reestablish and propagate RSFQ pulses when long PTL connections are required. The cell has integrated PTL transmitters and receivers and is meant to connect directly to a PTL.

Schematic

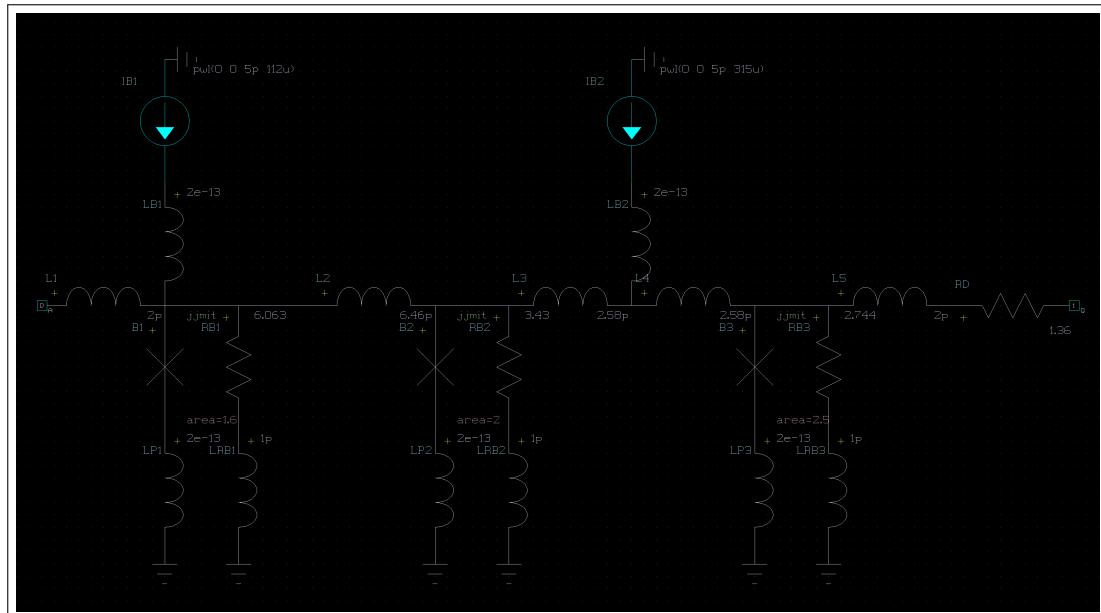


Figure 5.1: Schematic of RSFQ JTLT.

Layout

The physical layout for the RSFQ JTLT is shown in Fig. 5.2. The layout height is $70 \mu m$ and the width is $40 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

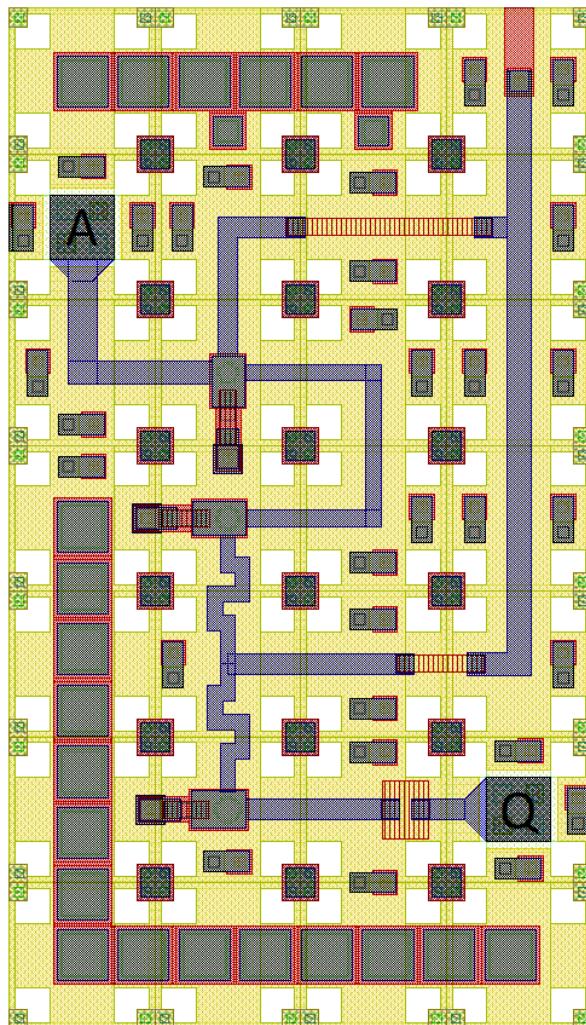


Figure 5.2: RSFQ JTLT Layout.

Table 5.1: RSFQ JTLT pin list.

Pin	Description
a	Data input
q	Data output

Analog model

```

1 * Back-annotated simulation file written      34 .param LP1=LP
2   ↳ by InductEx v.6.0.4 on 26-4-21.          35 .param LP2=LP
3 * Author: L. Schindler                      36 .param LP3=LP
4 * Version: 2.1                               37 .param LB1=LB
5 * Last modification date: 8 June 2021       38 .param LB2=LB
6 * Last modification by: L. Schindler        39 .param RB1=B0Rs/B1
7 *$Ports a q                                40 .param RB2=B0Rs/B2
8 .subckt LSMITLL_JTLT a q                   41 .param RB3=B0Rs/B3
9 .model jjmit jj(rtype=1, vg=2.8mV, cap    42 .param LRB1=(RB1/Rsheet)*Lsheet+LP
10   ↳ =0.07pF, r0=160, rn=16, icrit=0.1mA   43 .param LRB2=(RB2/Rsheet)*Lsheet+LP
11   ↳ )                                         44 .param LRB3=(RB3/Rsheet)*Lsheet+LP
12 .param Phi0=2.067833848E-15                45 B1 6 7 jjmit area=B1
13 .param B0=1                                 46 B2 9 10 jjmit area=B2
14 .param Ic0=0.0001                           47 B3 12 13 jjmit area=B3
15 .param IcRs=100u*6.859904418               48 IB1 0 18 pwl(0 0 5p IB1)
16 .param B0Rs=IcRs/Ic0*B0                     49 IB2 0 19 pwl(0 0 5p IB2)
17 .param Rsheet=2                            50 L1 a 6 2.914E-12
18 .param Lsheet=1.13e-12                      51 L2 6 9 6.425E-12
19 .param LP=0.2p                             52 L3 9 16 2.566E-12
20 .param Lptl=2p                            53 L4 16 12 2.538E-12
21 .param LB=2p                               54 L5 12 17 2.312E-12
22 .param BiasCoef=0.7                        55 LB1 6 18 2.904E-12
23 .param RD=1.36                            56 LB2 16 19 2.517E-12
24 .param B1=1.62                            57 LP1 0 7 5.462E-13
25 .param B2=2.0                            58 LP2 0 10 5.434E-13
26 .param B3=2.5                            59 LP3 0 13 5.144E-13
27 .param IB1=112u                           60 LRB1 0 8 LRB1
28 .param IB2=315u                           61 LRB2 0 11 LRB2
29 .param L1=Lptl                            62 LRB3 0 14 LRB3
30 .param L2=Phi0/(2*B1*Ic0)                 63 RB1 8 6 RB1
31 .param L3=(Phi0/(2*B2*Ic0))/2            64 RB2 11 9 RB2
32 .param L4=L3                             65 RB3 14 12 RB3
33 .param L5=Lptl                           66 RD 17 q RD
34 .ends

```

Listing 5.1: RSFQ JTLT JoSIM netlist.

The simulation results for the RSFQ JTLT using JoSIM is shown in Fig. 5.3. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the phase over the output JJ of pin **q**, and
- (d) the phase over the input JJ of the load circuit connected via a PTL to the JTLT.

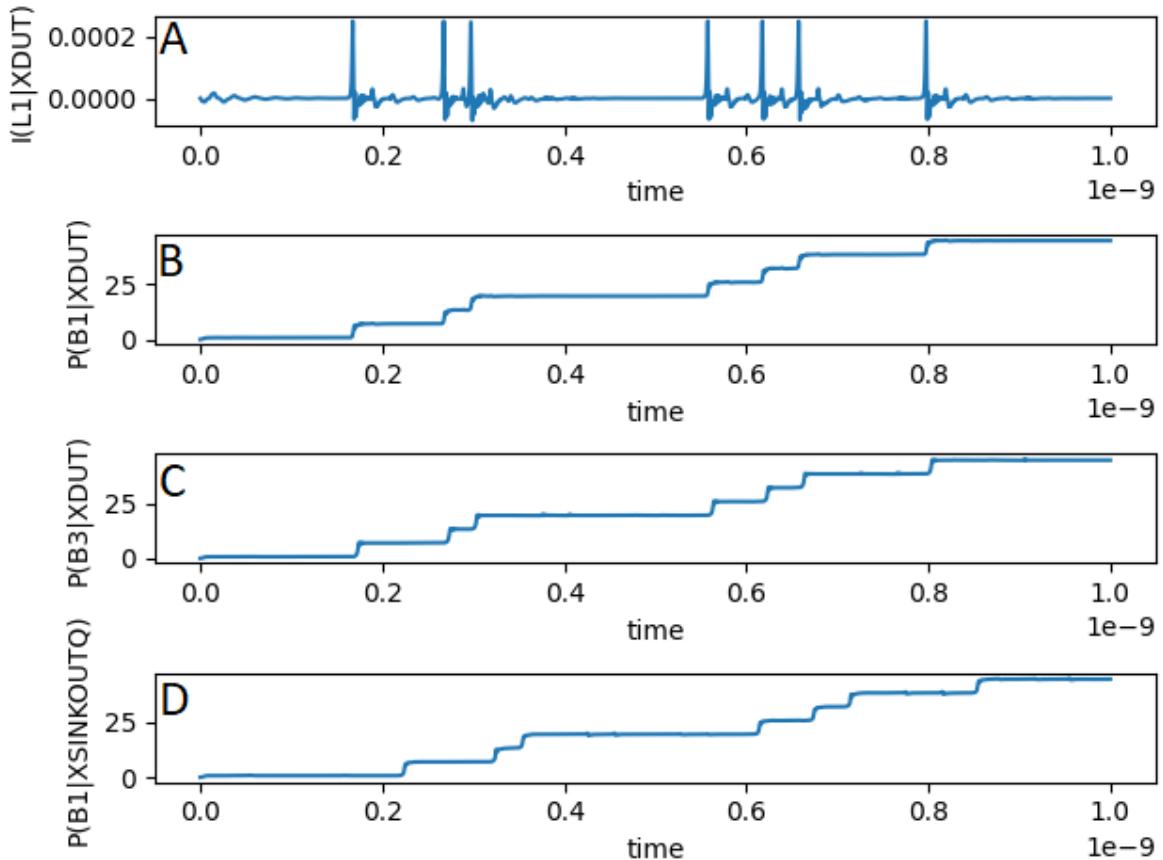


Figure 5.3: RSFQ JTLT analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 3 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmitll_JTLT_v2p1_optimized (a, q);
9
10 input
11   a;
12
13 output
14   q;
15
16 reg
17   q;
18
19 real
20   delay_state0_a_q = 6.2,
21   ct_state0_a_a = 6.8;
22
23 reg
24   errorsignal_a;
25
26 integer
27   outfile,
28   cell_state; // internal state of the cell
29
30 initial
31 begin
32   errorsignal_a = 0;
33   cell_state = 0; // Startup state
34   q = 0; // All outputs start at 0
35 end
36
37 always @(posedge a or negedge a) // execute at positive and negative edges of input
38 begin
39   if ($time>4) // arbitrary steady-state time)
40     begin
41       if (errorsignal_a == 1'b1) // A critical timing is active for this input
42         begin
43           outfile = $fopen("errors.txt", "a");
44           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
45           ↪ ", $stime);
46           $fclose(outfile);
47           q <= 1'bX; // Set all outputs to unknown
48         end
49       if (errorsignal_a == 0)
50         begin
51           case (cell_state)
52             0: begin
53               q <= #(delay_state0_a_q) !q;
54               errorsignal_a = 1; // Critical timing on this input; assign
55               ↪ immediately
56               errorsignal_a <= #(ct_state0_a_a) 0; // Clear error signal
57               ↪ after critical timing expires
58             end
59           endcase
60         end
61     end
62   end
63 endmodule

```

Listing 5.2: RSFQ JTLT verilog model.

The digital simulation results for the RSFQ JTLT is shown in Fig. 5.4 and the Mealy finite state machine diagram, extracted using *TimEx*, is shown in Fig. 5.5.

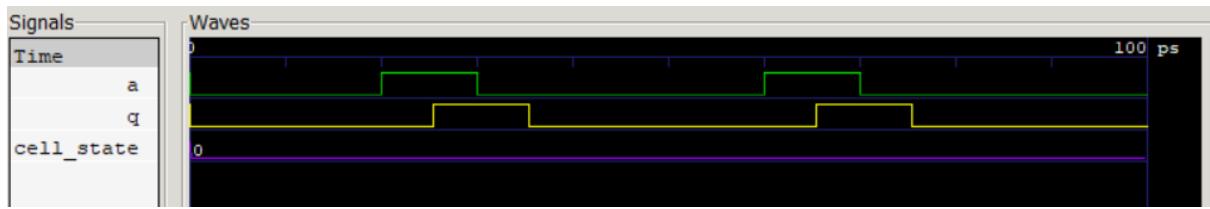


Figure 5.4: RSFQ JTLT digital simulation results.

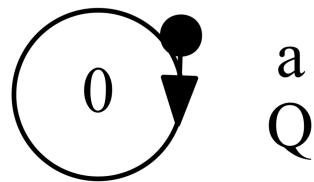


Figure 5.5: RSFQ JTLT Mealy finite state machine diagram.

Power Consumption

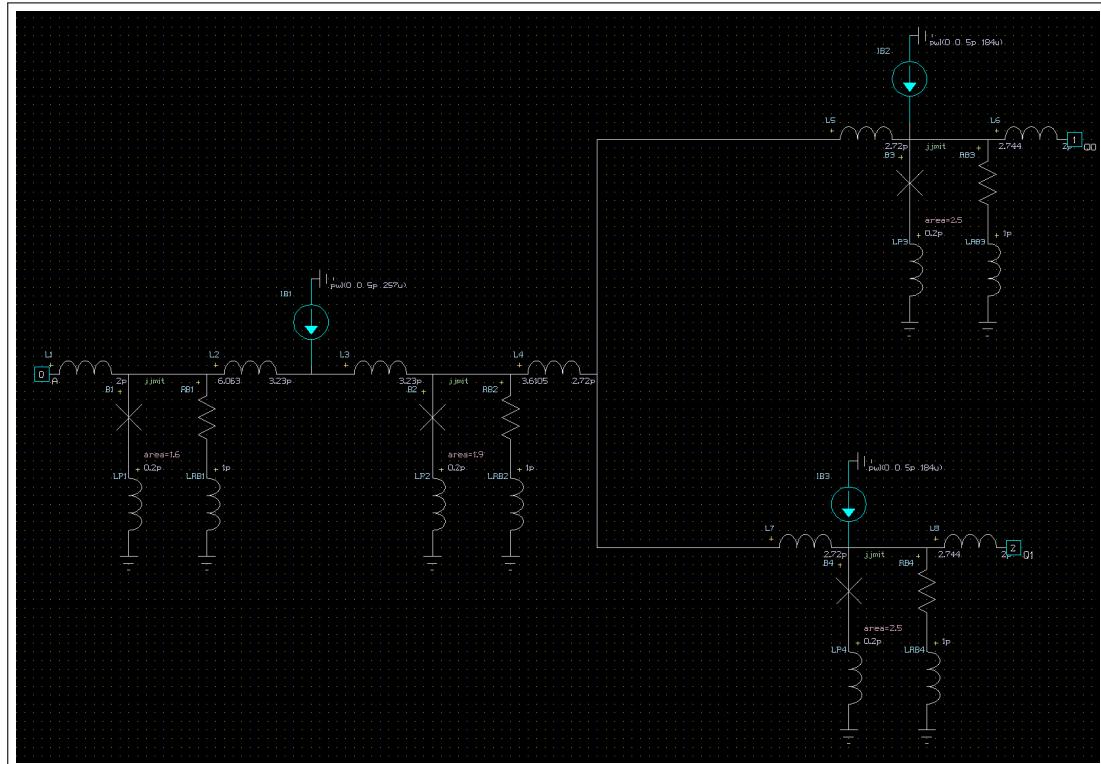
Table 5.2: RSFQ JTLT power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	1110	1.26
2	1110	2.52
5	1110	6.31
10	1110	12.6
20	1110	25.2
50	1110	63.1

5.1.2 SPLITT

The SPLITT cell is used to split a single pulse signal line into two duplicate output pulse signal lines. The cell has integrated PTL transmitters and receivers and is meant to connect directly to a PTL.

Schematic



Layout

The physical layout for the RSFQ SPLITT is shown in Fig. 5.7. The layout height is $70 \mu m$ and the width is $50 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

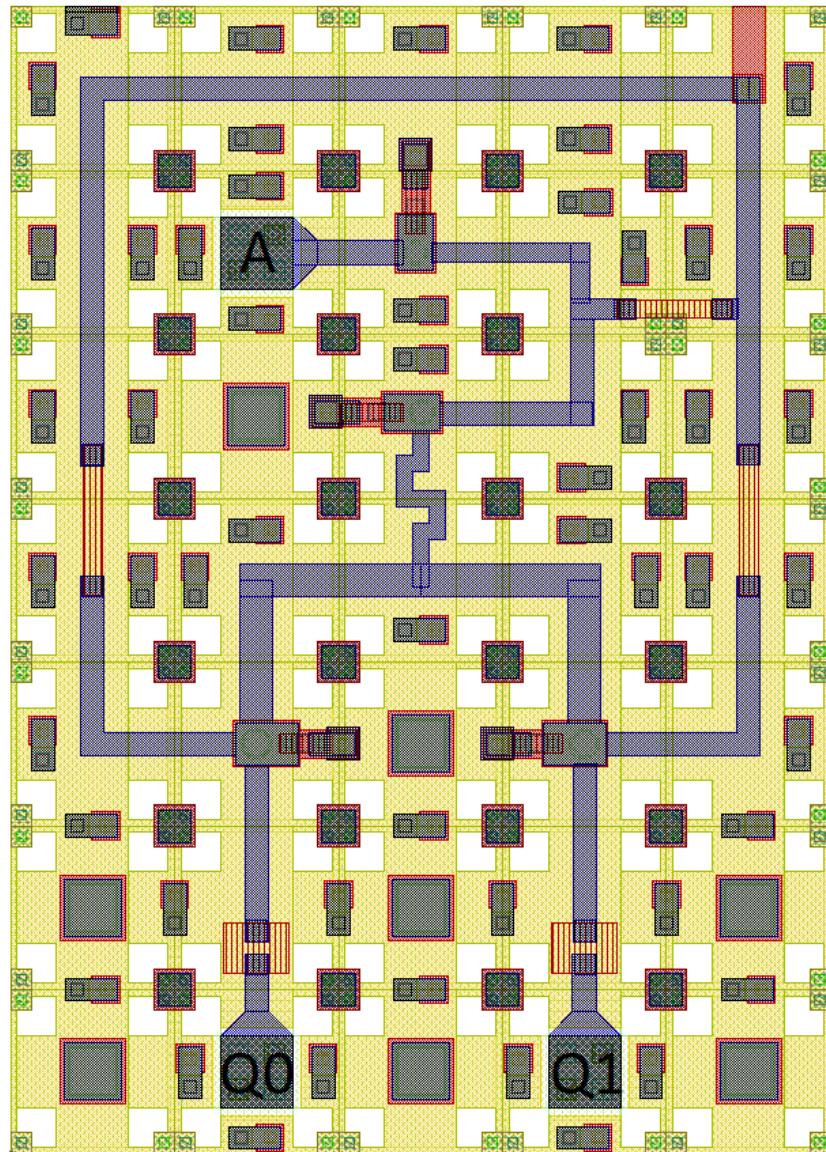


Figure 5.7: RSFQ SPLITT layout.

Analog model

```

1  * Back-annotated simulation file written      39 | .param L8=Lptl
2   ↪ by InductEx v.6.0.4 on 26-4-21.          40 | .param RB1=B0Rs/B1
3  * Author: L. Schindler                      41 | .param RB2=B0Rs/B2
4  * Version: 2.1                               42 | .param RB3=B0Rs/B3
5  * Last modification date: 26 April 2021     43 | .param RB4=B0Rs/B4
6  * Last modification by: L. Schindler        44 | .param LRB1=(RB1/Rsheet)*Lsheet
7  *$Ports                                         a q0 q1    45 | .param LRB2=(RB2/Rsheet)*Lsheet
8 .subckt LSMITLL_SPLITT a q0 q1              46 | .param LRB3=(RB3/Rsheet)*Lsheet
9 .model jjmit jj(rtype=1, vg=2.8mV, cap     47 | .param LRB4=(RB4/Rsheet)*Lsheet
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    48 | IB1 0 4 pwl(0 0 5p IB1)
   ↪ )                                           49 | IB2 0 8 pwl(0 0 5p IB2)
10 .param Phi0=2.067833848E-15                50 | IB3 0 11 pwl(0 0 5p IB3)
11 .param B0=1                                 51 | B1 2 3 jjmit area=B1
12 .param Ic0=0.0001                            52 | B2 5 6 jjmit area=B2
13 .param IcRs=100u*6.859904418               53 | B3 8 9 jjmit area=B3
14 .param B0Rs=IcRs/Ic0*B0                     54 | B4 11 12 jjmit area=B4
15 .param Rsheet=2                             55 | L1 a 2 1.515E-12
16 .param Lsheet=1.13e-12                       56 | L2 2 4 2.836E-12
17 .param LP=0.2p                             57 | L3 4 5 2.85E-12
18 .param IC=2.5                             58 | L4 5 7 2.68E-12
19 .param ICreceive=1.6                        59 | L5 7 8 2.699E-12
20 .param ICtrans=2.5                         60 | L6 8 10 2.364E-12
21 .param Lptl=2p                            61 | L7 7 11 2.704E-12
22 .param BiasCoef=0.7                        62 | L8 11 13 2.369E-12
23 .param RD=1.36                            63 | LP1 3 0 4.521E-13
24                                         64 | LP2 6 0 5.395E-13
25 .param B1=1.6                            65 | LP3 9 0 5.061E-13
26 .param B2=1.67                           66 | LP4 12 0 5.078E-13
27 .param B3=2.5                            67 | RB1 2 102 RB1
28 .param B4=2.5                            68 | LRB1 102 0 LRB1
29 .param IB1=225u                           69 | RB2 5 105 RB2
30 .param IB2=185u                           70 | LRB2 105 0 LRB2
31 .param IB3=185u                           71 | RB3 8 108 RB3
32 .param L1=Lptl                           72 | LRB3 108 0 LRB3
33 .param L2=(Phi0/(2*B1*Ic0))/2            73 | RB4 11 111 RB4
34 .param L3=(Phi0/(2*B1*Ic0))/2            74 | LRB4 111 0 LRB4
35 .param L4=(Phi0/(2*B2*Ic0))/2            75 | RD1 13 q0 RD
36 .param L5=(Phi0/(2*B2*Ic0))/2            76 | RD2 10 q1 RD
37 .param L6=Lptl                           77 | .ends
38 .param L7=(Phi0/(2*B2*Ic0))/2

```

Listing 5.3: RSFQ SPLITT JoSIM netlist.

Table 5.3: RSFQ SPLITT pin list.

Pin	Description
a	Data input
q0	Data output
q1	Data output

The JoSIM simulation results for the RSFQ SPLITT are shown in Fig. 5.8. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the phase over the output JJ of pin **q0**,
- (d) the phase over the output JJ of pin **q1**,
- (e) the phase over the input JJ of the load cell connected to pin **q0** through a PTL, and
- (f) the phase over the input JJ of the load cell connected to pin **q1** through a PTL.

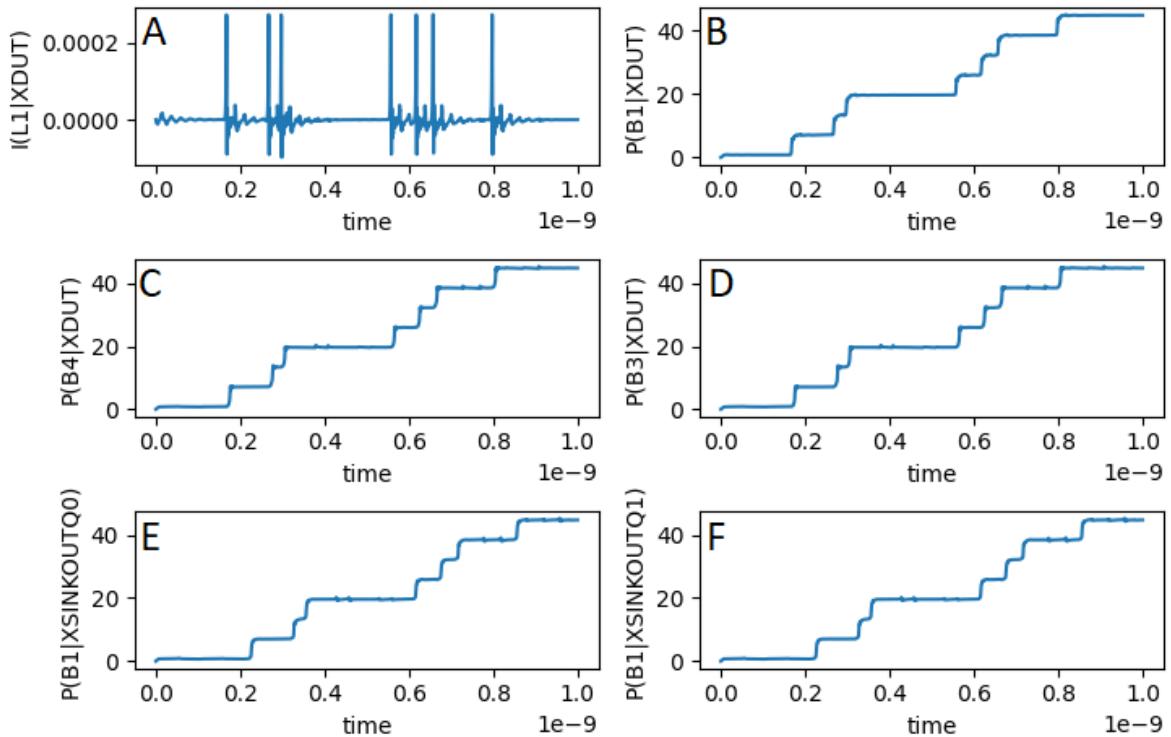


Figure 5.8: RSFQ SPLITT analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 8 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmitll_SPLITT_v2p1_optimized (a, q0, q1);
9   input
10    a;
11   output
12    q0, q1;
13   reg
14    q0, q1;
15   real
16    delay_state0_a_q0 = 7.3,
17    delay_state0_a_q1 = 7.3,
18    ct_state0_a_a = 9.1;
19
20   reg
21    errorsignal_a;
22
23   integer
24    outfile,
25    cell_state; // internal state of the cell
26   initial
27    begin
28      errorsignal_a = 0;
29      cell_state = 0; // Startup state
30      q0 = 0; // All outputs start at 0
31      q1 = 0; // All outputs start at 0
32    end
33
34   always @(posedge a or negedge a) // execute at positive and negative edges of input
35   begin
36     if ($time>4) // arbitrary steady-state time)
37     begin
38       if (errorsignal_a == 1'b1) // A critical timing is active for this input
39         begin
40           outfile = $fopen("errors.txt", "a");
41           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m; %d_ps.\n"
42                         , $stime);
43           $fclose(outfile);
44           q0 <= 1'bX; // Set all outputs to unknown
45           q1 <= 1'bX; // Set all outputs to unknown
46         end
47       if (errorsignal_a == 0)
48         begin
49           case (cell_state)
50             0: begin
51               q0 <= #(delay_state0_a_q0) !q0;
52               q1 <= #(delay_state0_a_q1) !q1;
53               errorsignal_a = 1; // Critical timing on this input; assign
54                                         // immediately
55               errorsignal_a <= #(ct_state0_a_a) 0; // Clear error signal
56                                         // after critical timing expires
57             end
58           endcase
59         end
60       end
61     end
62   endmodule

```

Listing 5.4: RSFQ SPLITT verilog model.

The digital simulation results for the RSFQ SPLITT is shown in Fig. 5.9 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 5.10.

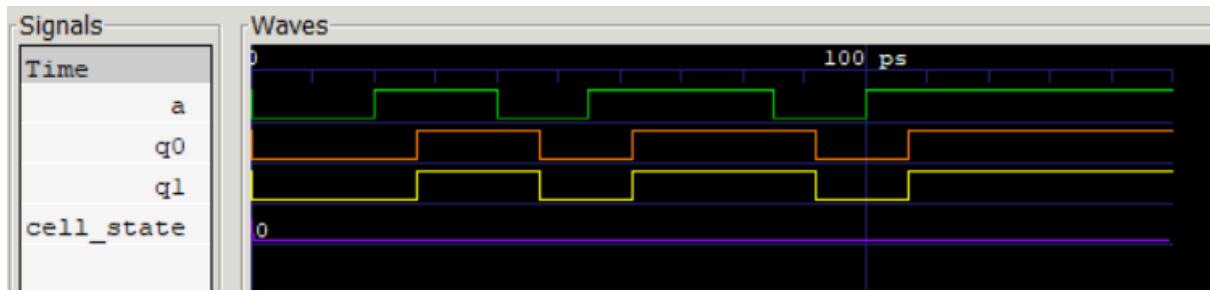


Figure 5.9: RSFQ SPLITT digital simulation results.

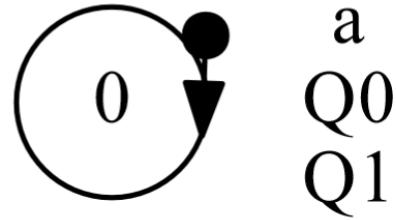


Figure 5.10: RSFQ SPLITT Mealy finite state diagram.

Power consumption

Table 5.4: RSFQ SPLITT power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	1547	1.71
2	1547	3.42
5	1547	8.55
10	1547	17.1
20	1547	34.2
50	1547	85.5

5.1.3 MERGET

The MERGET joins two input pulse signal lines and provides a single output pulse signal line. If there is a pulse on either input lines, the MERGET will generate a pulse on the output signal line. The MERGET has integrated PTL transmitters and receivers is designed to be directly connected to a PTL.

Schematic

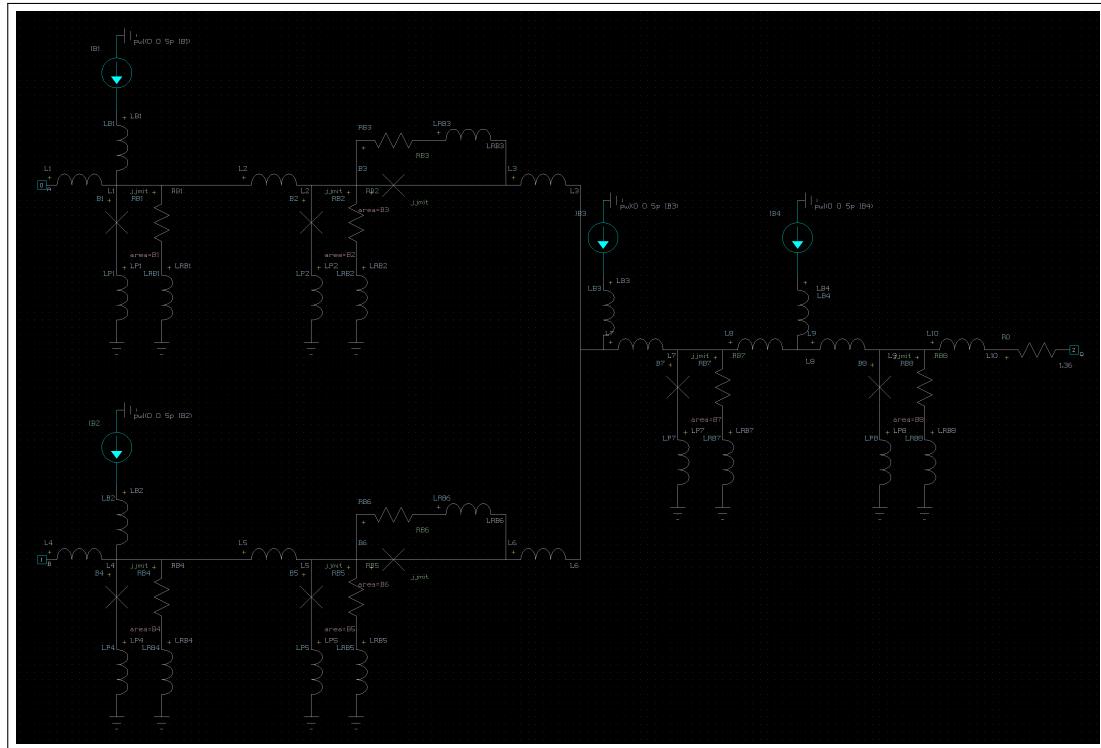


Figure 5.11: Schematic of RSFQ MERGET.

Layout

The physical layout of the RSFQ MERGET is shown in Fig. 5.12. The height of the layout is $70 \mu\text{m}$ and the width is $70 \mu\text{m}$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

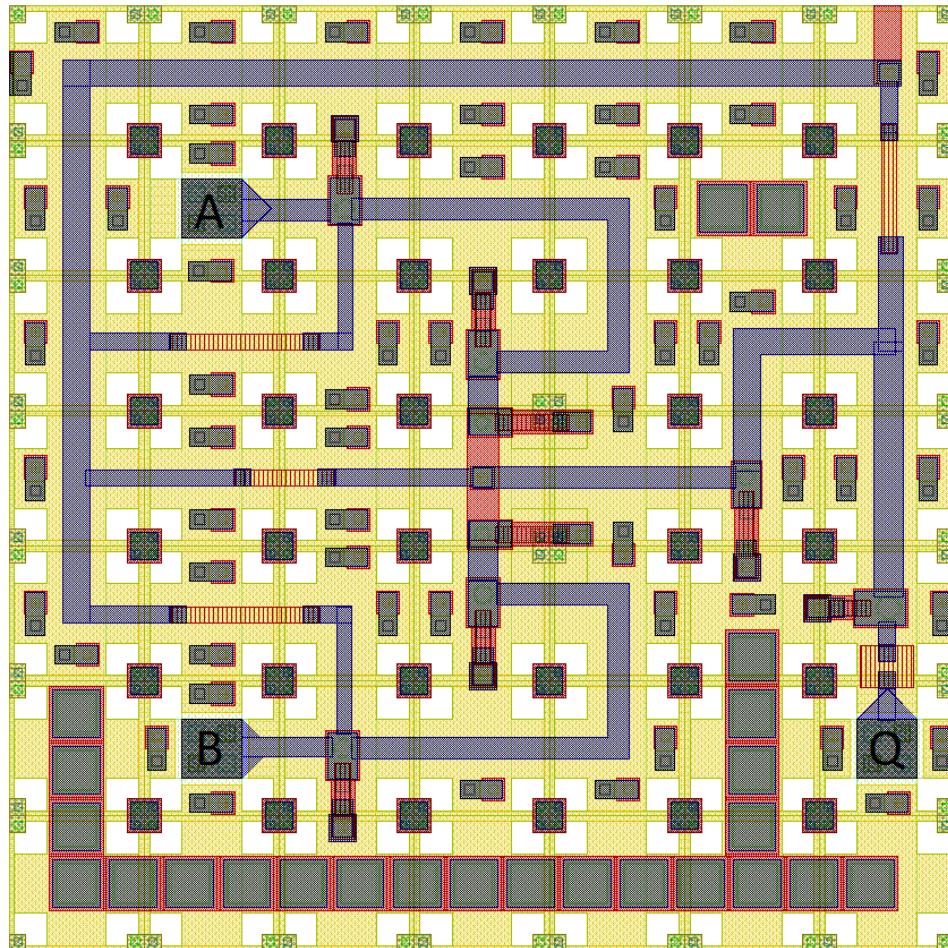


Figure 5.12: RSFQ MERGET layout.

Analog model

```

1  * Back-annotated simulation file written      58 | .param LRB3=(RB3/Rsheet)*Lsheet
2  *   ↪ by InductEx v.6.0.4 on 27-4-21.        59 | .param LRB4=(RB4/Rsheet)*Lsheet
3  * Author: L. Schindler                      60 | .param LRB5=(RB5/Rsheet)*Lsheet
4  * Version: 2.1                                61 | .param LRB6=(RB6/Rsheet)*Lsheet
5  * Last modification date: 27 April 2021       62 | .param LRB7=(RB7/Rsheet)*Lsheet
6  * Last modification by: L. Schindler          63 | .param LRB8=(RB8/Rsheet)*Lsheet
7  *$ports a b q
8  .subckt LSmitll_MERGET a b q
9  .model jjmit jj(rtype=1, vg=2.8mV, cap
10    ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA
11    ↪ )
12 .param Phi0=2.067833848E-15
13 .param B0=1
14 .param Ic0=0.0001
15 .param IcRs=100u*6.859904418
16 .param B0Rs=IcRs/Ic0*B0
17 .param Rsheet=2
18 .param Lsheet=1.13e-12
19 .param LP=0.2p
20 .param IC=2.5
21 .param ICreceive=1.6
22 .param ICtrans=2.5
23 .param Lptl=2p
24 .param LB=2p
25 .param BiasCoef=0.70
26 .param RD=1.36
27 .param B1=1.6
28 .param B2=1.54
29 .param B3=0.95
30 .param B4=B1
31 .param B5=B2
32 .param B6=B3
33 .param B7=1.16
34 .param B8=2.5
35 .param L1=Lptl
36 .param L2=Phi0/(2*B1*Ic0)
37 .param L3=Phi0/(2*B2*Ic0)*(B7/(B2+B7))
38 .param L4=L1
39 .param L5=L2
40 .param L6=L3
41 .param L7=Phi0/(2*B2*Ic0)*(B2/(B2+B7))
42 .param L8=(Phi0/(2*B7*Ic0))*(B8/(B7+B8))
43 .param L9=(Phi0/(2*B7*Ic0))*(B7/(B7+B8))
44 .param L10=Lptl
45 .param IB1=148u
46 .param IB2=148u
47 .param IB3=241u
48 .param IB4=176u
49 .param RB1=B0Rs/B1
50 .param RB2=B0Rs/B2
51 .param RB3=B0Rs/B3
52 .param RB4=B0Rs/B4
53 .param RB5=B0Rs/B5
54 .param RB6=B0Rs/B6
55 .param RB7=B0Rs/B7
56 .param RB8=B0Rs/B8
57 .param LRB1=(RB1/Rsheet)*Lsheet
58 .param LRB2=(RB2/Rsheet)*Lsheet
59 .param LRB3=(RB3/Rsheet)*Lsheet
60 .param LRB4=(RB4/Rsheet)*Lsheet
61 .param LRB5=(RB5/Rsheet)*Lsheet
62 .param LRB6=(RB6/Rsheet)*Lsheet
63 .param LRB7=(RB7/Rsheet)*Lsheet
64 .param LRB8=(RB8/Rsheet)*Lsheet
65 L1 a 1 1.439E-12
66 L2 1 4 6.953E-12
67 L3 4 6 1.156E-12
68 L4 b 8 1.462E-12
69 L5 8 11 6.998E-12
70 L6 11 13 1.166E-12
71 L7 7 15 3.436E-12
72 L8 15 19 2.713E-12
73 L9 17 19 2.713E-12
74 L10 19 21 7.598E-13
75 RD 21 q RD
76 IB1 0 3 pwl(0 0 5p IB1)
77 IB2 0 10 pwl(0 0 5p IB2)
78 IB3 0 14 pwl(0 0 5p IB3)
79 IB4 0 18 pwl(0 0 5p IB4)
80 B1 1 2 jjmit area=B1
81 B2 4 5 jjmit area=B2
82 B3 6 7 jjmit area=B3
83 B4 8 9 jjmit area=B4
84 B5 11 12 jjmit area=B5
85 B6 13 7 jjmit area=B6
86 B7 15 16 jjmit area=B7
87 B8 19 20 jjmit area=B8
88 LP1 2 0 4.853E-13
89 LP2 5 0 5.381E-13
90 LP4 9 0 4.789E-13
91 LP5 12 0 5.351E-13
92 LP7 16 9 5.981E-13
93 LP8 20 0 3.813E-13
94 RB1 1 101 RB1
95 RB2 4 104 RB2
96 RB3 6 106 RB3
97 RB4 8 108 RB4
98 RB5 11 111 RB5
99 RB6 13 113 RB6
100 RB7 15 115 RB7
101 RB8 19 119 RB8
102 LRB1 101 0 LRB1
103 LRB2 104 0 LRB2
104 LRB3 106 7 LRB3
105 LRB4 108 0 LRB4
106 LRB5 111 0 LRB5
107 LRB6 113 7 LRB6
108 LRB7 115 0 LRB7
109 LRB8 119 0 LRB8
110 LB1 1 3 2.952E-12
111 LB2 8 10 2.928E-12
112 LB3 7 14 2.643E-12
113 LB4 17 18 1.428E-12
114 .ends

```

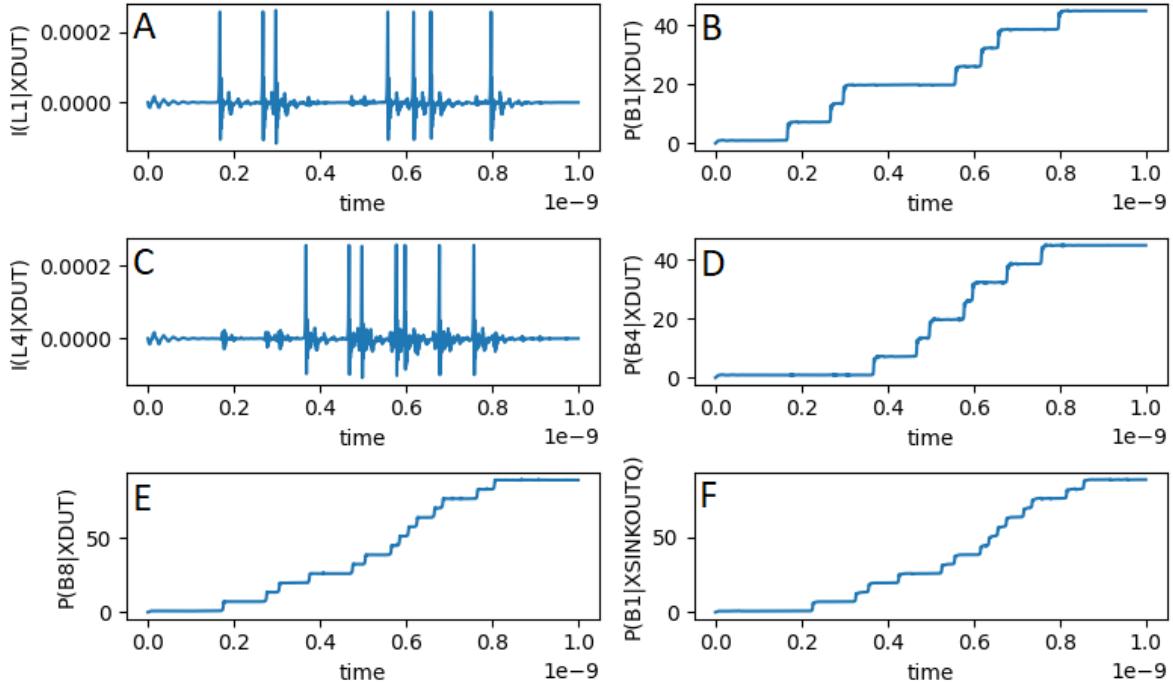
Listing 5.5: RSFQ MERGET JoSIM netlist.

Table 5.5: RSFQ MERGET pin list.

Pin	Description
a	Data input
b	Data input
q	Data output

The JoSIM simulation results for the RSFQ MERGET are shown in Fig. 5.13. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b**,
- (d) the phase over the input JJ of pin **b**,
- (e) the phase over the output JJ of pin **q**,
- (f) the phase over the input JJ of the load cell connected to pin **q** via a PTL.

**Figure 5.13:** RSFQ MERGET analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 8 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 // -----
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 // -----
15 'timescale 1ps/100fs
16 module LSmitll_MERGET_v2p1_optimized (a, b, q);
17
18 input
19   a, b;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state0_a_q = 8.0,
29   delay_state0_b_q = 8.0,
30   ct_state0_a_a = 3.3,
31   ct_state0_a_b = 3.2,
32   ct_state0_b_a = 2.3,
33   ct_state0_b_b = 3.3;
34
35 reg
36   errorsignal_a,
37   errorsignal_b;
38
39 integer
40   outfile,
41   cell_state; // internal state of the cell
42
43 initial
44 begin
45   errorsignal_a = 0;
46   errorsignal_b = 0;
47   cell_state = 0; // Startup state
48   q = 0; // All outputs start at 0
49 end
50
51 always @(posedge a or negedge a) // execute at positive and negative edges of input
52 begin
53   if ($time>4) // arbitrary steady-state time)
54     begin
55       if (errorsignal_a == 1'b1) // A critical timing is active for this input
56         begin
57           outfile = $fopen("errors.txt", "a");
58           $fdisplay(outfile, "Violation of critical timing in module %m; %0d ps.\n",
59           $stime);
60           $fclose(outfile);
61           q <= 1'bX; // Set all outputs to unknown
62         end
63       if (errorsignal_a == 0)
64         begin
65           case (cell_state)
66             0: begin
67               q <= #(delay_state0_a_q) !q;
68             end
69           endcase
70         end
71     end
72   end
73 end
74
75 endmodule

```

```

67          errorsignal_a = 1; // Critical timing on this input; assign
68          //      ↪ immediately
69          errorsignal_a <= #(ct_state0_a_a) 0; // Clear error signal
70          //      ↪ after critical timing expires
71          errorsignal_b = 1; // Critical timing on this input; assign
72          //      ↪ immediately
73          errorsignal_b <= #(ct_state0_a_b) 0; // Clear error signal
74          //      ↪ after critical timing expires
75      end
76  endcase
77 end
78
79 always @(posedge b or negedge b) // execute at positive and negative edges of input
80 begin
81     if ($time>4) // arbitrary steady-state time)
82     begin
83         if (errorsignal_b == 1'b1) // A critical timing is active for this input
84         begin
85             outfile = $fopen("errors.txt", "a");
86             $fdisplay(outfile, "Violation of critical timing in module %m; %0d ps.\n"
87                         //      ↪ ", $stime);
88             $fclose(outfile);
89             q <= 1'bX; // Set all outputs to unknown
90         end
91         if (errorsignal_b == 0)
92         begin
93             case (cell_state)
94                 0: begin
95                     q <= #(delay_state0_b_q) !q;
96                     errorsignal_a = 1; // Critical timing on this input; assign
97                     //      ↪ immediately
98                     errorsignal_a <= #(ct_state0_b_a) 0; // Clear error signal
99                     //      ↪ after critical timing expires
100                    errorsignal_b = 1; // Critical timing on this input; assign
101                    //      ↪ immediately
102                    errorsignal_b <= #(ct_state0_b_b) 0; // Clear error signal
103                    //      ↪ after critical timing expires
104                end
105            endcase
106        end
107    end
108 endmodule

```

Listing 5.6: RSFQ MERGET verilog model.

The digital simulation results for the RSFQ MERGET is shown in Fig. 5.14 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 5.15.

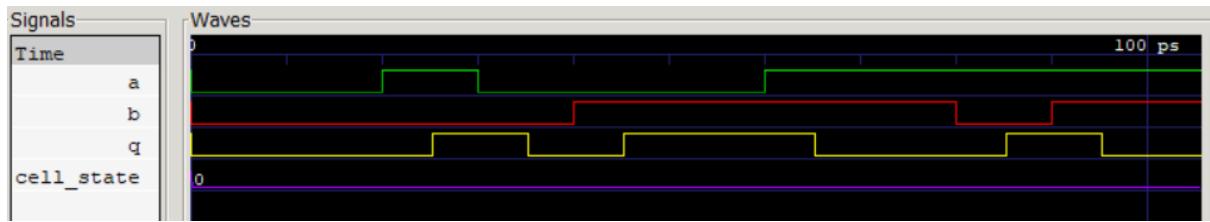


Figure 5.14: RSFQ MERGET digital simulation results.

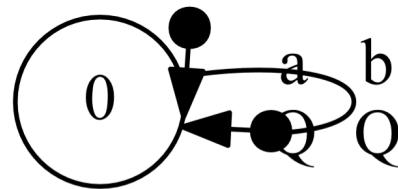


Figure 5.15: RSFQ MERGET Mealy finite state diagram.

Power consumption

Table 5.6: RSFQ MERGET power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	1851	2.45
2	1851	4.90
5	1851	12.2
10	1851	24.5
20	1851	49.0
50	1851	122

5.1.4 Always0T Asynchronous

The Always0T Asynchronous cell provides an output which is always zero. Two versions of the cell exist: One with an **a** input port, as seen in Fig. 5.16, and one without an **a** input port, as seen in Fig. 5.17. The Always0T Asynchronous is designed to be directly connected to a PTL.

Schematic

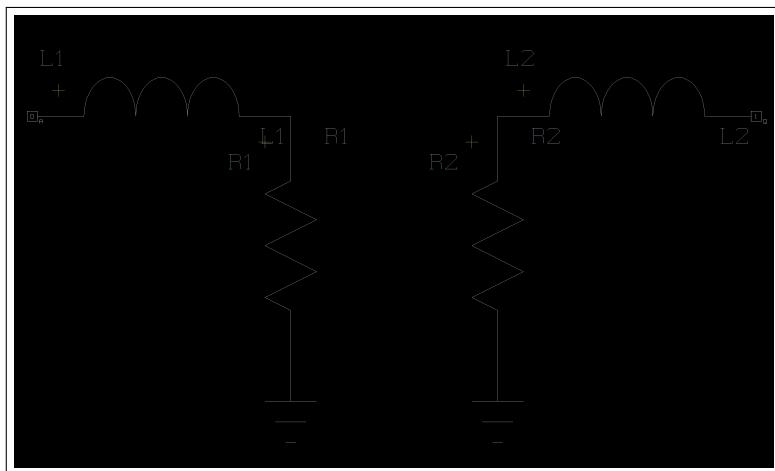


Figure 5.16: Schematic of RSFQ Always0T Asynchronous.

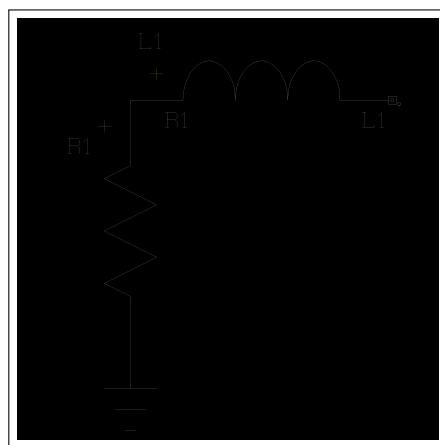


Figure 5.17: Schematic of RSFQ Always0T Asynchronous without an **a** input port.

Layout

The physical layouts of the two RSFQ Always0T Asynchronous cell versions are shown in Fig. 5.18a and 5.18b respectively. The height of the layout for both versions of the cell is $70 \mu m$ and the width is $10 \mu m$. The PTL connection extends to connect to a PTL on either M1 or M3.

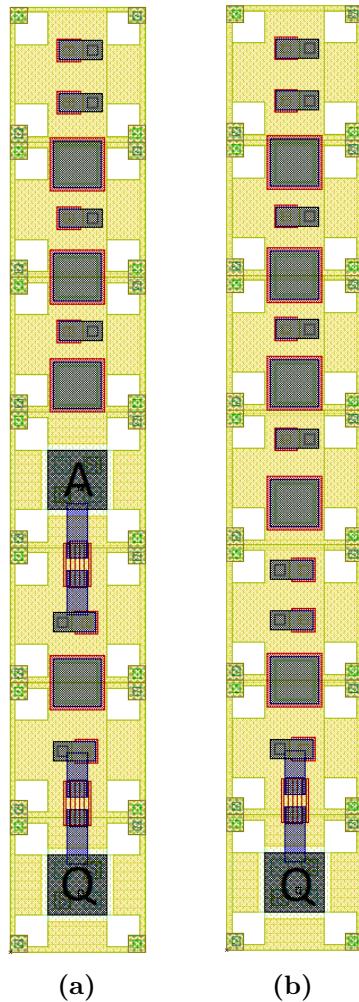


Figure 5.18: The physical layouts for (a) the RSFQ Always0T Asynchronous and (b) the RSFQ Always0T Asynchronous without an **a** input port.

Analog model

```

1 * Author: L. Schindler
2 * Version: 2.1
3 * Last modification date: 13 August 2020
4 * Last modification by: L. Schindler
5
6 *$Ports           a q
7 .subckt LSMITLL_ALWAYS0T_async a q
8 La a 1 2p
9 Lq 2 q 2p
10 RA 1 0 2
11 RQ 2 0 2
12 .ends

```

Listing 5.7: RSFQ Always0T Asynchronous JoSIM netlist.

Table 5.7: RSFQ Always0T Asynchronous pin list.

Pin	Description
a	Data input
q	Data output

```

1 * Author: L. Schindler
2 * Version: 2.1
3 * Last modification date: 17 August 2020
4 * Last modification by: L. Schindler
5
6 *$Ports q
7 .subckt LSMITLL_ALWAYS0T_async_noA q
8 Lq 2 q 2p
9 RQ 2 0 2
10 .ends

```

Listing 5.8: RSFQ Always0T Asynchronous, without an **a** input port, JoSIM netlist.

Table 5.8: RSFQ Always0T Asynchronous, without an **a** input port, pin list.

Pin	Description
q	Data output

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 13 August 2020
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmitll_always0T_async (a, q);
9
10 input
11   a;
12 output
13   q;
14 reg
15   q;
16
17 initial
18   begin
19     q = 0; // Output always 0
20   end
21
22 always
23   begin
24     #10 q = 0; // Output always 0
25   end
26
27 endmodule

```

Listing 5.9: RSFQ Always0T Asynchronous verilog model.

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 17 August 2020
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmitll_Always0T_async_noA (q);
9
10 output
11   q;
12
13 reg
14   q;
15
16 initial
17   begin
18     q = 0; // Output always 0
19   end
20
21 always
22   begin
23     #10 q = 0; // Output always 0
24   end
25
26 endmodule

```

Listing 5.10: RSFQ Always0T Asynchronous, without an **a** input port, verilog model.

5.1.5 Always0T Synchronous

The Always0T Synchronous cell provides an output which is always zero synchronous to a clock signal. Two versions of the cell exist: One with an **a** input port, as seen in Fig. 5.19, and one without an **a** input port, as seen in Fig. 5.20. The Always0T Synchronous is designed to be directly connected to a PTL.

Schematic

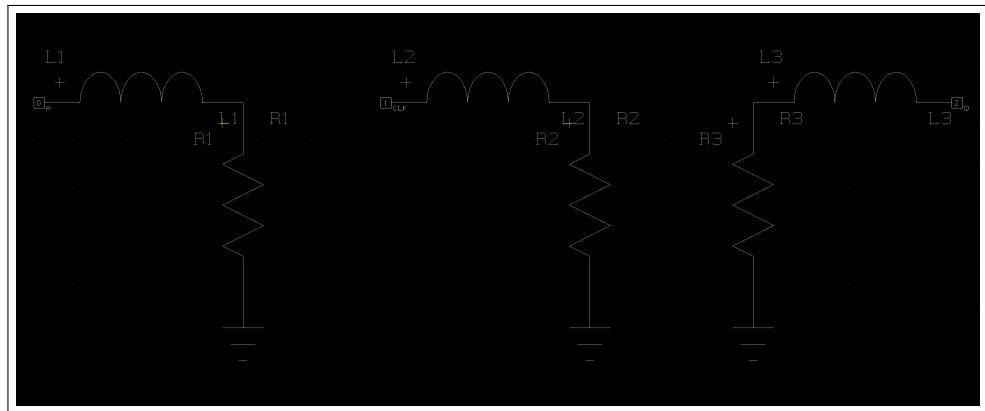


Figure 5.19: Schematic of RSFQ Always0T Synchronous.

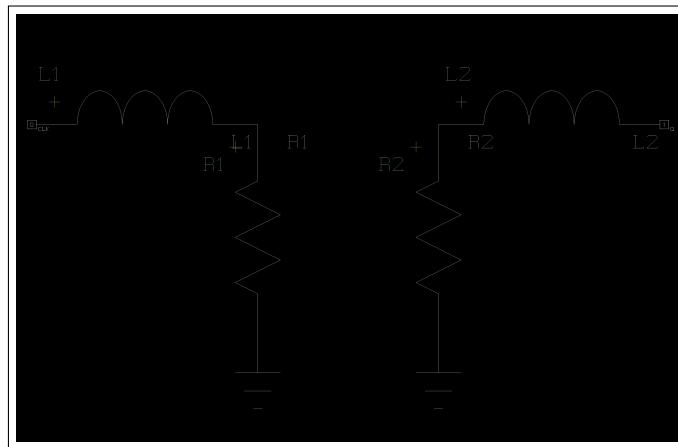


Figure 5.20: Schematic of RSFQ Always0T Synchronous without an **a** input port.

Layout

The physical layouts of the two RSFQ Always0T Synchronous cell versions are shown in Fig. 5.21a and 5.21b respectively. The height for both versions of the Always0T Synchronous layout is $70 \mu m$ and the width is $10 \mu m$. The PTL connection extends to connect to a PTL on either M1 or M3.

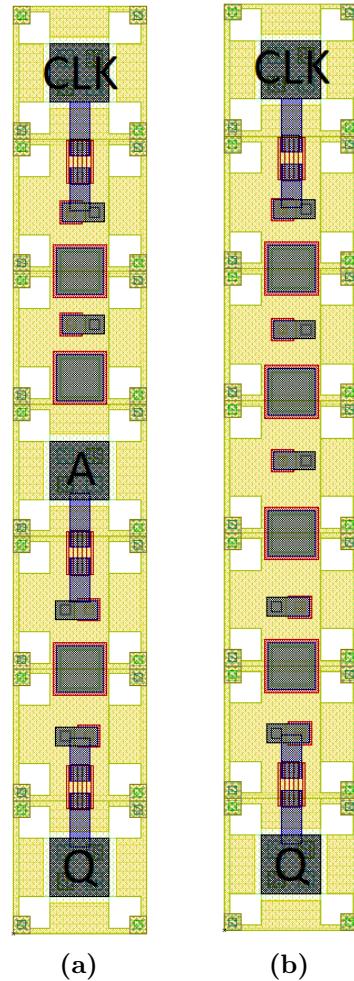


Figure 5.21: The physical layouts for (a) the RSFQ Always0T Synchronous and (b) the RSFQ Always0T Synchronous without an **a** input port.

Analog model

```

1 * Author: L. Schindler
2 * Version: 2.1
3 * Last modification date: 13 August 2020
4 * Last modification by: L. Schindler
5
6 *$Ports a clk q
7 .subckt LSMITLL_ALWAYS0T_SYNC a clk q
8 La a 1 2p
9 Lclk clk 2 2p
10 Lq 3 q 2p
11 RA 1 0 2
12 RCLK 2 0 2
13 RQ 3 0 2
14 .ends

```

Listing 5.11: RSFQ Always0T Synchronous JoSIM netlist.

Table 5.9: RSFQ Always0T Synchronous pin list.

Pin	Description
a	Data input
clk	Clock input
q	Data output

```

1 * Author: L. Schindler
2 * Version: 2.1
3 * Last modification date: 17 August 2020
4 * Last modification by: L. Schindler
5
6 *$Ports clk q
7 .subckt LSMITLL_ALWAYS0T_SYNC_noA clk q
8 Lclk clk 2 2p
9 Lq 3 q 2p
10 RCLK 2 0 2
11 RQ 3 0 2
12 .ends

```

Listing 5.12: RSFQ Always0T Synchronous, without an **a** input port, JoSIM netlist.

Table 5.10: RSFQ Always0T Synchronous, without an **a** input port, pin list.

Pin	Description
clk	Clock input
q	Data output

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 13 August 2020
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmitll_Always0T_sync (a, clk, q);
9
10 input
11   a, clk;
12
13 output
14   q;
15
16 reg
17   q;
18
19 initial
20   begin
21     q = 0; // Output always 0
22   end
23
24 always @(posedge clk or negedge clk)
25   begin
26     #10 q = 0; // Output always 0
27   end
28 endmodule

```

Listing 5.13: RSFQ Always0T Synchronous verilog model.

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 17 August 2020
5 // Last modification by: L. Schindler
6 // -----
7 `timescale 1ps/100fs
8 module LSmitll_Always0T_sync_noA ( clk, q);
9
10 input
11   clk;
12
13 output
14   q;
15
16 reg
17   q;
18
19 initial
20   begin
21     q = 0; // Output always 0
22   end
23
24 always @(posedge clk or negedge clk)
25   begin
26     #10 q = 0; // Output always 0
27   end
28 endmodule

```

Listing 5.14: RSFQ Always0T Synchronous, without an **a** input port, verilog model.

5.2 Logic Cells

5.2.1 AND2T

The RSFQ AND2T cell generates an output pulse if pulses from both input signal lines were received before the clock signal. The AND2T is designed with integrated PTL transmitters and receivers and is meant to be connected directly to a PTL.

Schematic

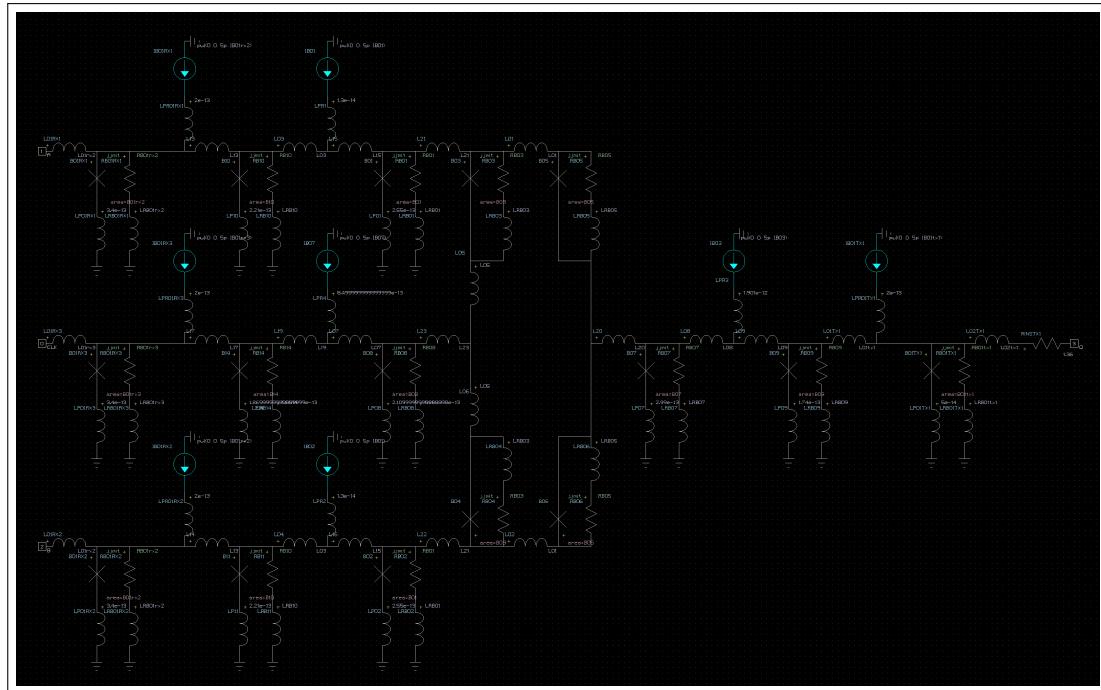


Figure 5.22: Schematic of RSFQ AND2T.

Layout

The physical layout for the RSFQ AND2T is shown in Fig. 5.23. The layout height is $70 \mu m$ and the width is $100 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

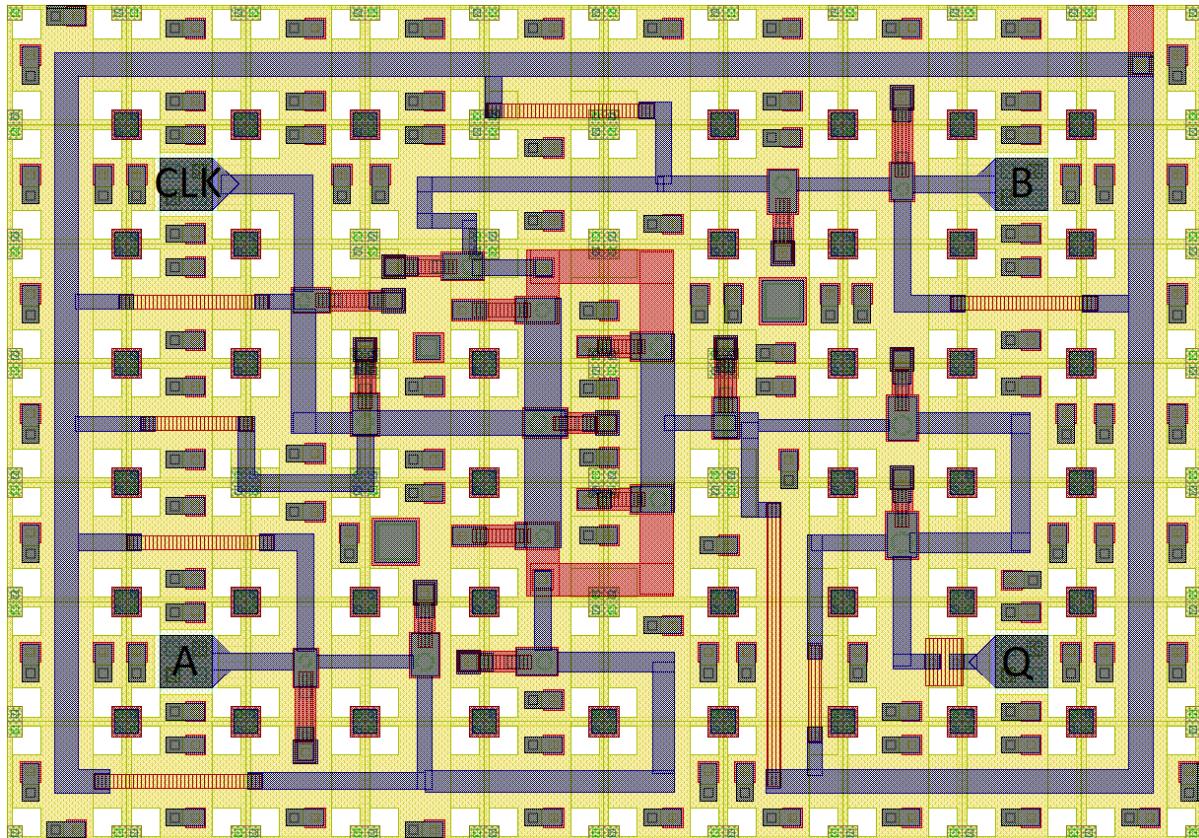


Figure 5.23: RSFQ AND2T Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param L19=1.1464p
2   ↪ by InductEx v.6.0.4 on 30-4-21.          65  .param L20=0.9p
3  * Author: L. Schindler                      66  .param L21=0.2p
4  * Version: 2.1                               67  .param L22=2.925p
5  * Last modification date: 30 April 2021     68  .param L23=4.644p
6  * Last modification by: L. Schindler        69  .param L24=Lptl
7
8  *$Ports a b clk q
9  .subckt LSmitll_AND2T a b clk q
10 .model jjmit jj(rtype=1, vg=2.8mV, cap
11   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA
12   ↪ )
13 .param B0=1.0
14 .param Ic0=0.0001
15 .param IcRs=100u*6.859904418
16 .param B0Rs=IcRs/Ic0*B0
17 .param Rsheet=2
18 .param Lsheet=1.13e-12
19 .param RD=1.36
20 .param LB=0.2p
21 .param Lptl=2p
22 .param LP=0.2p
23 .param B1=0.88
24 .param B2=1.76
25 .param B3=1.32
26 .param B4=1.13
27 .param B5=1.53
28 .param B6=0.90
29 .param B7=1.50
30 .param B8=1.76
31 .param B9=B1
32 .param B10=B2
33 .param B11=B3
34 .param B12=B4
35 .param B13=B5
36 .param B14=1.26
37 .param B15=2.04
38 .param B16=2.27
39 .param IB1=131u
40 .param IB2=113u
41 .param IB3=128u
42 .param IB4=179u
43 .param IB5=IB1
44 .param IB6=IB2
45 .param IB7=63u
46 .param IB8=214u
47 .param L1=Lptl
48 .param L2=2.23p
49 .param L3=1.9325p
50 .param L4=6.105p
51 .param L5=1.2909p
52 .param L6=2.58p
53 .param L7=1.1464p
54 .param L8=Lptl
55 .param L9=1.9428p
56 .param L10=0.2p
57 .param L11=1.9932p
58 .param L13=Lptl
59 .param L14=2.23p
60 .param L15=1.9325p
61 .param L16=6.105p
62 .param L17=1.2909p
63 .param L18=2.58p
64 .param L19=1.1464p
65 .param L20=0.9p
66 .param L21=0.2p
67 .param L22=2.925p
68 .param L23=4.644p
69 .param L24=Lptl
70
71 .param LB1=LB
72 .param LB2=LB
73 .param LB3=LB
74 .param LB4=LB
75 .param LB5=LB
76 .param LB6=LB
77 .param LB7=LB
78 .param LB8=LB
79
80 .param LP1=LP
81 .param LP2=LP
82 .param LP3=LP
83 .param LP6=LP
84 .param LP7=LP
85 .param LP8=LP
86 .param LP9=LP
87 .param LP10=LP
88 .param LP11=LP
89 .param LP14=LP
90 .param LP15=LP
91 .param LP16=LP
92
93 .param RB1=B0Rs/B1
94 .param RB2=B0Rs/B2
95 .param RB3=B0Rs/B3
96 .param RB4=B0Rs/B4
97 .param RB5=B0Rs/B5
98 .param RB6=B0Rs/B6
99 .param RB7=B0Rs/B7
100 .param RB8=B0Rs/B8
101 .param RB9=B0Rs/B9
102 .param RB10=B0Rs/B10
103 .param RB11=B0Rs/B11
104 .param RB12=B0Rs/B12
105 .param RB13=B0Rs/B13
106 .param RB14=B0Rs/B14
107 .param RB15=B0Rs/B15
108 .param RB16=B0Rs/B16
109
110 .param LRB1=(RB1/Rsheet)*Lsheet
111 .param LRB2=(RB2/Rsheet)*Lsheet
112 .param LRB3=(RB3/Rsheet)*Lsheet
113 .param LRB4=(RB4/Rsheet)*Lsheet
114 .param LRB5=(RB5/Rsheet)*Lsheet
115 .param LRB6=(RB6/Rsheet)*Lsheet
116 .param LRB7=(RB7/Rsheet)*Lsheet
117 .param LRB8=(RB8/Rsheet)*Lsheet
118 .param LRB9=(RB9/Rsheet)*Lsheet
119 .param LRB10=(RB10/Rsheet)*Lsheet
120 .param LRB11=(RB11/Rsheet)*Lsheet
121 .param LRB12=(RB12/Rsheet)*Lsheet
122 .param LRB13=(RB13/Rsheet)*Lsheet
123 .param LRB14=(RB14/Rsheet)*Lsheet
124 .param LRB15=(RB15/Rsheet)*Lsheet
125 .param LRB16=(RB16/Rsheet)*Lsheet
126
127 B1 2 4 jjmit area=B1
128 B2 5 6 jjmit area=B2
129 B3 9 10 jjmit area=B3

```

```

130 | B4 11 13 jjmit area=B4          183 | L24 45 48 2.426E-12
131 | B5 12 38 jjmit area=B5          184 |
132 | B6 15 17 jjmit area=B6          185 | LP1 4 0 5.322E-13
133 | B7 20 19 jjmit area=B7          186 | LP2 6 0 5.415E-13
134 | B8 24 23 jjmit area=B8          187 | LP3 10 0 5.529E-13
135 | B9 26 28 jjmit area=B9          188 | LP6 17 0 5.48E-13
136 | B10 29 30 jjmit area=B10         189 | LP7 19 0 5.392E-13
137 | B11 33 34 jjmit area=B11         190 | LP8 23 0 5.294E-13
138 | B12 36 35 jjmit area=B12         191 | LP9 28 0 5.186E-13
139 | B13 38 37 jjmit area=B13         192 | LP10 30 0 4.924E-13
140 | B14 39 40 jjmit area=B14         193 | LP11 34 0 5.748E-13
141 | B15 43 44 jjmit area=B15         194 | LP14 40 0 5.268E-13
142 | B16 45 47 jjmit area=B16         195 | LP15 44 0 5.107E-13
143 |                               196 | LP16 47 0 5.28E-13
144 | IB1 0 3 pwl(0 0 5p IB1)          197 |
145 | IB2 0 8 pwl(0 0 5p IB2)          198 | RD 48 q RD
146 | IB3 0 16 pwl(0 0 5p IB3)         199 |
147 | IB4 0 21 pwl(0 0 5p IB4)          200 | RB1 2 102 RB1
148 | IB5 0 27 pwl(0 0 5p IB5)          201 | LRB1 102 0 LRB1
149 | IB6 0 32 pwl(0 0 5p IB6)          202 | RB2 5 105 RB2
150 | IB7 0 42 pwl(0 0 5p IB7)          203 | LRB2 105 0 LRB2
151 | IB8 0 46 pwl(0 0 5p IB8)          204 | RB3 9 109 RB3
152 |                               205 | LRB3 109 0 LRB3
153 | LB1 2 3 2.705E-12               206 | RB4 11 111 RB4
154 | LB2 7 8 3.072E-12               207 | LRB4 111 13 LRB4
155 | LB3 15 16 8.269E-13              208 | RB5 12 112 RB5
156 | LB4 20 21 3.807E-12              209 | LRB5 112 38 LRB5
157 | LB5 26 27 2.958E-12              210 | RB6 15 115 RB6
158 | LB6 31 32 1.53E-12               211 | LRB6 115 0 LRB6
159 | LB7 41 42 1.761E-12              212 | RB7 20 120 RB7
160 | LB8 45 46 3.373E-12              213 | LRB7 120 0 LRB7
161 |                               214 | RB8 24 122 RB8
162 | L1 a 2 1.603E-12                215 | LRB8 122 0 LRB8
163 | L2 2 5 2.243E-12                216 | RB9 26 126 RB9
164 | L3 5 7 1.928E-12                217 | LRB9 126 0 LRB9
165 | L4 7 9 6.173E-12                218 | RB10 29 129 RB10
166 | L5 9 11 1.295E-12               219 | LRB10 129 0 LRB10
167 | L6 11 12 2.568E-12              220 | RB11 33 133 RB11
168 | L7 13 24 1.332E-12              221 | LRB11 133 0 LRB11
169 | L8 clk 15 3.135E-12              222 | RB12 35 135 RB12
170 | L9 15 20 2.099E-12              223 | LRB12 135 36 LRB12
171 | L11 20 24 1.991E-12             224 | RB13 37 137 RB13
172 | L13 b 26 1.592E-12              225 | LRB13 137 38 LRB13
173 | L14 26 29 2.218E-12             226 | RB14 39 141 RB14
174 | L15 29 31 1.949E-12             227 | LRB14 141 0 LRB14
175 | L16 31 33 6.082E-12             228 | RB15 43 143 RB15
176 | L17 33 35 1.263E-12             229 | LRB15 143 0 LRB15
177 | L18 35 37 2.595E-12              230 | RB16 45 145 RB16
178 | L19 36 24 1.287E-12              231 | LRB16 145 0 LRB16
179 | L20 38 39 9.063E-13              232 | .ends
180 | L21 39 41 2.265E-13
181 | L22 41 43 2.896E-12
182 | L23 43 45 4.653E-12

```

Listing 5.15: RSFQ AND2T JoSIM netlist.**Table 5.11:** RSFQ AND2T pin list.

Pin	Description
a	Data input
b	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ AND2T using JoSIM is shown in Fig. 5.24. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b**,
- (d) the phase over the input JJ of pin **b**,
- (e) the current through the input inductor connected to pin **clk**,
- (f) the phase over the input JJ of pin **clk**,
- (g) the phase over the output JJ of pin **q**, and
- (h) the phase over the input JJ of the load circuit connected via a PTL to pin **q**.

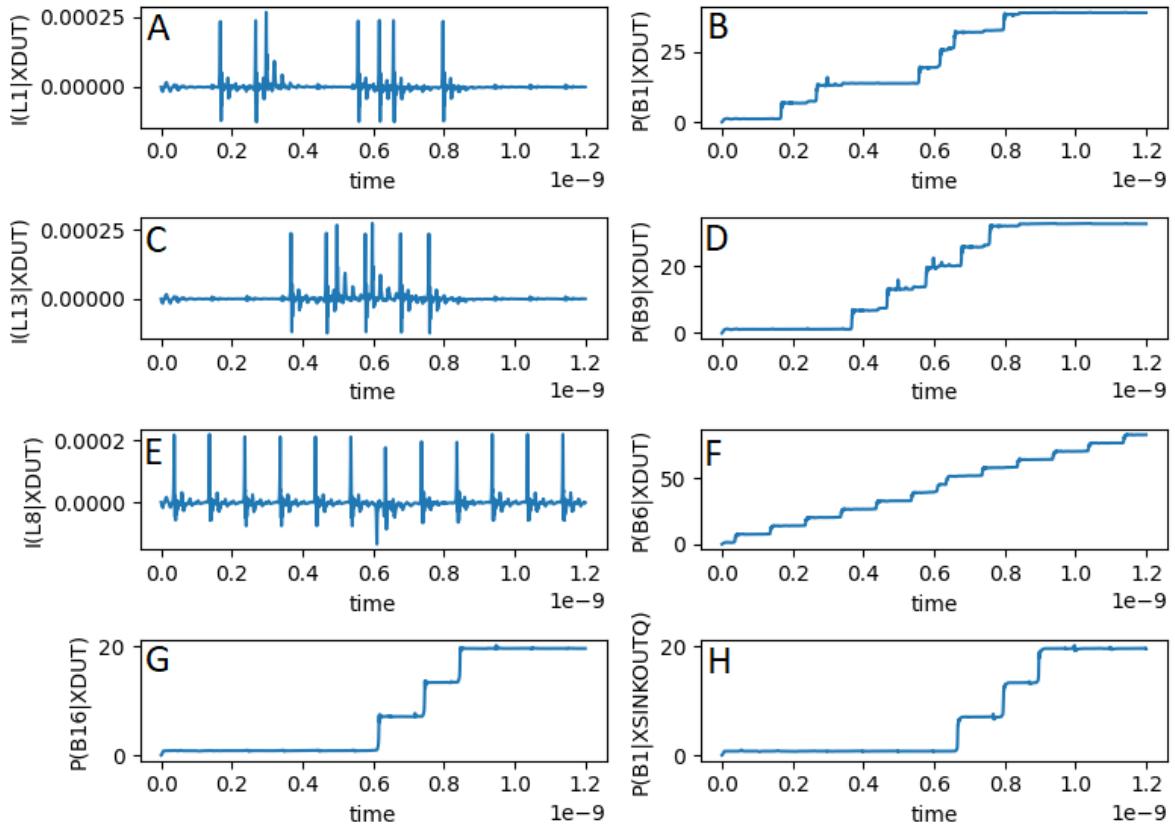


Figure 5.24: RSFQ AND2T analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 2 June 2021
5 // Last modification by: L. Schindler
6 // -----
7
8 // -----
9 // Automatically extracted verilog file, created with TimEx v2.05
10 // Timing description and structural design for IARPA-BAA-14-03 via
11 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
12 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
13 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
14 // (c) 2016-2018 Stellenbosch University
15 // -----
16 `timescale 1ps/100fs
17 module LSmitll_AND2T (a, b, clk, q);
18
19 input
20   a, b, clk;
21
22 output
23   q;
24
25 reg
26   q;
27
28 real
29   delay_state3_clk_q = 9.3,
30   ct_state0_clk_a = 3.8,
31   ct_state0_clk_b = 3.8,
32   ct_state1_clk_a = 3.7,
33   ct_state1_clk_b = 3.5,
34   ct_state2_clk_a = 3.5,
35   ct_state2_clk_b = 3.8,
36   ct_state3_clk_a = 2.4,
37   ct_state3_clk_b = 2.3;
38
39 reg
40   errorsignal_a,
41   errorsignal_b,
42   errorsignal_clk;
43
44 integer
45   outfile,
46   cell_state; // internal state of the cell
47
48 initial
49 begin
50   errorsignal_a = 0;
51   errorsignal_b = 0;
52   errorsignal_clk = 0;
53   cell_state = 0; // Startup state
54   q = 0; // All outputs start at 0
55 end
56
57 always @(posedge a or negedge a) // execute at positive and negative edges of input
58 begin
59   if ($time>4) // arbitrary steady-state time
60     begin
61       if (errorsignal_a == 1'b1) // A critical timing is active for this input
62         begin
63           outfile = $fopen("errors.txt", "a");
64           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m; %0d_ps.\n"
65           ↪ ", $stime);
66           $fclose(outfile);
67           q <= 1'bX; // Set all outputs to unknown

```

```

67         end
68     if (errorsignal_a == 0)
69     begin
70         case (cell_state)
71             0: begin
72                 cell_state = 1; // Blocking statement -- immediately
73             end
74             1: begin
75                 end
76             2: begin
77                 cell_state = 3; // Blocking statement -- immediately
78             end
79             3: begin
80                 end
81             endcase
82         end
83     end
84 end
85
86 always @(posedge b or negedge b) // execute at positive and negative edges of input
87 begin
88     if ($time>4) // arbitrary steady-state time)
89     begin
90         if (errorsignal_b == 1'b1) // A critical timing is active for this input
91         begin
92             outfile = $fopen("errors.txt", "a");
93             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
94             ↪ ", $stime);
95             $fclose(outfile);
96             q <= 1'bX; // Set all outputs to unknown
97         end
98         if (errorsignal_b == 0)
99         begin
100             case (cell_state)
101                 0: begin
102                     cell_state = 2; // Blocking statement -- immediately
103                 end
104                 1: begin
105                     cell_state = 3; // Blocking statement -- immediately
106                 end
107                 2: begin
108                     end
109                 3: begin
110                     end
111             endcase
112         end
113     end
114 end
115
116 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
117 begin
118     if ($time>4) // arbitrary steady-state time)
119     begin
120         if (errorsignal_clk == 1'b1) // A critical timing is active for this input
121         begin
122             outfile = $fopen("errors.txt", "a");
123             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
124             ↪ ", $stime);
125             $fclose(outfile);
126             q <= 1'bX; // Set all outputs to unknown
127         end
128         if (errorsignal_clk == 0)
129         begin
130             case (cell_state)
131                 0: begin
132                     errorsignal_a = 1; // Critical timing on this input; assign
133                     ↪ immediately
134                     errorsignal_a <= #(ct_state0_clk_a) 0; // Clear error signal
135                     ↪ after critical timing expires
136                     errorsignal_b = 1; // Critical timing on this input; assign

```

```

133           → immediately
134           errorsignal_b <= #(ct_state0_clk_b) 0; // Clear error signal
135           → after critical timing expires
136       end
137   begin
138       cell_state = 0; // Blocking statement -- immediately
139       errorsignal_a = 1; // Critical timing on this input; assign
140           → immediately
141           errorsignal_a <= #(ct_state1_clk_a) 0; // Clear error signal
142           → after critical timing expires
143       errorsignal_b = 1; // Critical timing on this input; assign
144           → immediately
145           errorsignal_b <= #(ct_state1_clk_b) 0; // Clear error signal
146           → after critical timing expires
147   end
148 end
149 begin
150     q <= #(delay_state3_clk_q) !q;
151     cell_state = 0; // Blocking statement -- immediately
152     errorsignal_a = 1; // Critical timing on this input; assign
153         → immediately
154         errorsignal_a <= #(ct_state3_clk_a) 0; // Clear error signal
155         → after critical timing expires
156         errorsignal_b = 1; // Critical timing on this input; assign
157             → immediately
158             errorsignal_b <= #(ct_state3_clk_b) 0; // Clear error signal
159             → after critical timing expires
160     end
161 end
162 endmodule

```

Listing 5.16: RSFQ AND2T verilog model.

The digital simulation results for the RSFQ AND2T is shown in Fig. 5.25 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 5.26.

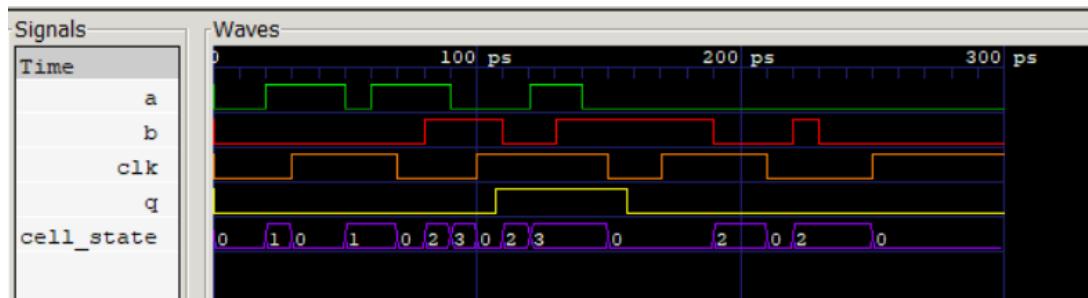


Figure 5.25: RSFQ AND2T digital simulation results.

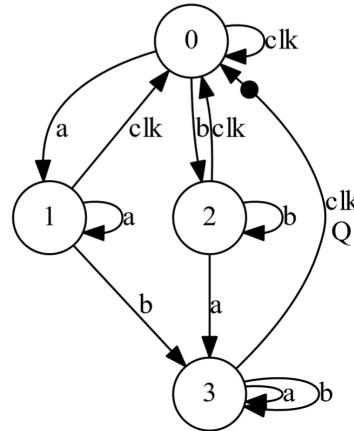


Figure 5.26: RSFQ AND2T Mealy finite state machine diagram.

Power Consumption

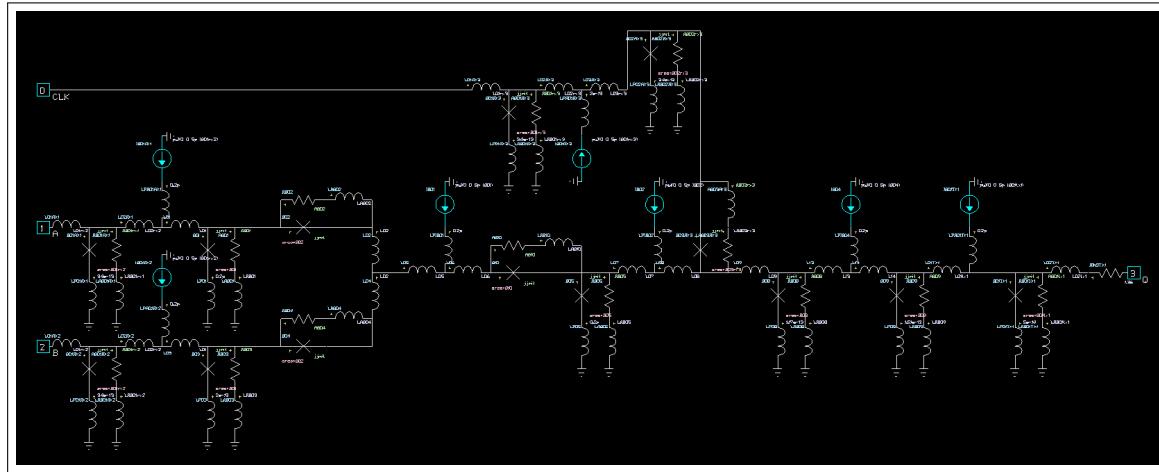
Table 5.12: RSFQ AND2T power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2787	4.75
2	2787	9.50
5	2787	23.7
10	2787	47.5
20	2787	95.0
50	2787	237

5.2.2 OR2T

The RSFQ OR2T cell generates an output pulse if an input pulse from either input lines was received before the clock signal. The OR2T cell is designed with integrated PTL transmitters and receivers and is intended to be connected directly to a PTL.

Schematic



Layout

The physical layout for the RSFQ OR2T is shown in Fig. 5.28. The layout height is $70 \mu m$ and the width is $100 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

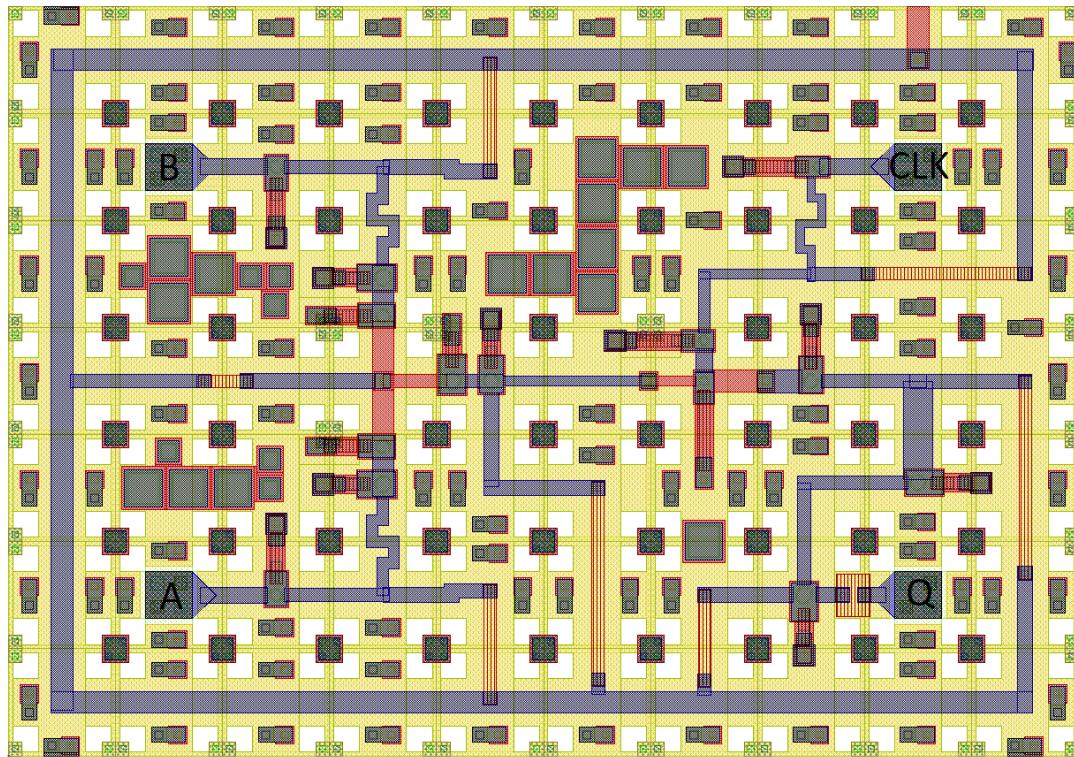


Figure 5.28: RSFQ OR2T Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param LB1=LB
2   ↪ by InductEx v.6.0.4 on 28-4-21.          65  .param LB2=LB
3  * Author: L. Schindler                      66  .param LB3=LB
4  * Version: 2.1                               67  .param LB4=LB
5  * Last modification date: 28 April 2021     68  .param LB5=LB
6  * Last modification by: L. Schindler        69  .param LB6=LB
7  *$ports a b clk q                         70  .param LB7=LB
8 .subckt LSMITLL_OR2T a b clk q            71  .param LP1=LP
9 .model jjmit jj(rtype=1, vg=2.8mV, cap    72  .param LP2=LP
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    73  .param LP4=LP
   ↪ )                                         74  .param LP5=LP
10 .param B0=1                                75  .param LP8=LP
11 .param Ic0=0.0001                           76  .param LP9=LP
12 .param IcRs=100u*6.859904418                77  .param LP10=LP
13 .param B0Rs=IcRs/Ic0*B0                     78  .param LP12=LP
14 .param Rsheet=2                            79  .param LP13=LP
15 .param Lsheet=1.13e-12                      80  .param LP14=LP
16 .param LP=0.2p                             81
17 .param Lptl=2p                            82  .param RB1=B0Rs/B1
18 .param LB=2p                               83  .param RB2=B0Rs/B2
19 .param RD=1.36                            84  .param RB3=B0Rs/B3
20
21 .param B1=1.17                            85  .param RB4=B0Rs/B4
22 .param B2=1.95                            86  .param RB5=B0Rs/B5
23 .param B3=1.31                            87  .param RB6=B0Rs/B6
24 .param B4=1.17                            88  .param RB7=B0Rs/B7
25 .param B5=1.95                            89  .param RB8=B0Rs/B8
26 .param B6=1.31                            90  .param RB9=B0Rs/B9
27 .param B7=2.20                            91  .param RB10=B0Rs/B10
28 .param B8=1.72                            92  .param RB11=B0Rs/B11
29 .param B9=0.81                            93  .param RB12=B0Rs/B12
30 .param B10=0.75                           94  .param RB13=B0Rs/B13
31 .param B11=0.63                           95  .param RB14=B0Rs/B14
32 .param B12=1.40                           96  .param LRB1=(RB1/Rsheet)*Lsheet+LP
33 .param B13=1.62                           97  .param LRB2=(RB2/Rsheet)*Lsheet+LP
34 .param B14=1.9                           98  .param LRB3=(RB3/Rsheet)*Lsheet+LP
35
36 .param IB1=141u                           99  .param LRB4=(RB4/Rsheet)*Lsheet+LP
37 .param IB2=141u                           100 .param LRB5=(RB5/Rsheet)*Lsheet+LP
38 .param IB3=328u                           101 .param LRB6=(RB6/Rsheet)*Lsheet+LP
39 .param IB4=81u                            102 .param LRB7=(RB7/Rsheet)*Lsheet+LP
40 .param IB5=98u                            103 .param LRB8=(RB8/Rsheet)*Lsheet+LP
41 .param IB6=81u                            104 .param LRB9=(RB9/Rsheet)*Lsheet+LP
42 .param IB7=177u                           105 .param LRB10=(RB10/Rsheet)*Lsheet+LP
43
44 .param L1=Lptl                           106 .param LRB11=(RB11/Rsheet)*Lsheet+LP
45 .param L2=2.0822p                          107 .param LRB12=(RB12/Rsheet)*Lsheet+LP
46 .param L3=2.6809p                          108 .param LRB13=(RB13/Rsheet)*Lsheet+LP
47 .param L4=1.3486p                          109 .param LRB14=(RB14/Rsheet)*Lsheet+LP
48 .param L5=Lptl                           110
49 .param L6=2.0822p                          111 B1 2 3      jjmit area=B1
50 .param L7=2.6809p                          112 B2 6 7      jjmit area=B2
51 .param L8=1.3486p                          113 B3 6 8      jjmit area=B3
52 .param L10=1.8890p                         114 B4 11 12     jjmit area=B4
53 .param L12=5.4916p                         115 B5 15 16     jjmit area=B5
54 .param L13=Lptl                           116 B6 15 43     jjmit area=B6
55 .param L14=3.3652p                         117 B7 19 22     jjmit area=B7
56 .param L15=4.0267p                         118 B8 22 21     jjmit area=B8
57 .param L16=0.7p                            119 B9 26 27     jjmit area=B9
58 .param L17=1.5727p                         120 B10 30 31    jjmit area=B10
59 .param L18=2.0776p                         121 B11 32 24    jjmit area=B11
60 .param L19=0.885p                          122 B12 33 34    jjmit area=B12
61 .param L20=4.2904p                         123 B13 37 38    jjmit area=B13
62 .param L21=Lptl                           124 B14 39 41    jjmit area=B14
63

```

```

130 | IB5 0 29 pwl(0 0 5p IB5)      167 | LP9 27 0          5.216E-13
131 | IB6 0 36 pwl(0 0 5p IB6)      168 | LP10 31 0         5.841E-13
132 | IB7 0 40 pwl(0 0 5p IB7)      169 | LP12 34 0         4.758E-13
133 |
134 | L1 a 2 1.593E-12             170 | LP13 38 0         5.26E-13
135 | L2 2 4 2.116E-12            171 | LP14 41 0         4.366E-13
136 | L3 4 6 2.655E-12            172
137 | L4 8 9 1.349E-12            173 RB1 2 102 RB1
138 | L5 b 11 1.596E-12           174 LRB1 102 0 LRB1
139 | L6 11 13 2.104E-12          175 RB2 6 106 RB2
140 | L7 13 15 2.655E-12          176 LRB2 106 0 LRB2
141 | L8 43 9 1.348E-12           177 RB3 6 108 RB3
142 | L10 9 19 1.883E-12          178 LRB3 108 8 LRB3
143 | L12 22 24 5.465E-12          179 RB4 11 111 RB4
144 | L13 clk 26 1.54E-12          180 LRB4 111 0 LRB4
145 | L14 26 28 3.367E-12          181 RB5 15 115 RB5
146 | L15 28 30 4.045E-12          182 LRB5 115 0 LRB5
147 | L16 30 32 5.696E-13          183 RB6 15 143 RB6
148 | L17 24 33 1.584E-12          184 LRB6 143 43 LRB6
149 | L18 33 35 2.075E-12          185 RB7 19 119 RB7
150 | L19 35 37 9.15E-13           186 LRB7 119 22 LRB7
151 | L20 37 39 4.26E-12           187 RB8 22 122 RB8
152 | L21 39 42 7.721E-13          188 LRB8 122 0 LRB8
153 | RD 42 q RD                  189 RB9 26 126 RB9
154 |
155 | LB1 4 5 2.162E-12            190 LRB9 126 0 LRB9
156 | LB2 13 14 2.177E-12          191 RB10 30 130 RB10
157 | LB3 9 18 2.831E-12           192 LRB10 130 0 LRB10
158 | LB4 22 23 3.933E-12          193 RB11 32 132 RB11
159 | LB5 28 29 1.367E-12          194 LRB11 132 24 LRB11
160 | LB6 35 36 2.221E-12          195 RB12 33 133 RB12
161 | LB7 39 40 2.035E-12          196 LRB12 133 0 LRB12
162 | LP1 3 0 4.886E-13            197 RB13 37 137 RB13
163 | LP2 7 0 4.645E-13            198 LRB13 137 0 LRB13
164 | LP4 12 0 4.935E-13            199 RB14 39 139 RB14
165 | LP5 16 0 4.65E-13             200 LRB14 139 0 LRB14
166 | LP8 21 0 5.102E-13            201 .ends

```

Listing 5.17: RSFQ OR2T JoSIM netlist.**Table 5.13:** RSFQ OR2T pin list.

Pin	Description
a	Data input
b	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ OR2T using JoSIM is shown in Fig. 5.29. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b**,
- (d) the phase over the input JJ of pin **b**,
- (e) the current through the input inductor connected to pin **clk**,
- (f) the phase over the input JJ of pin **clk**,
- (g) the phase over the output JJ of pin **q**, and
- (h) the phase over the input JJ of the load circuit connected via a PTL to pin **q**.

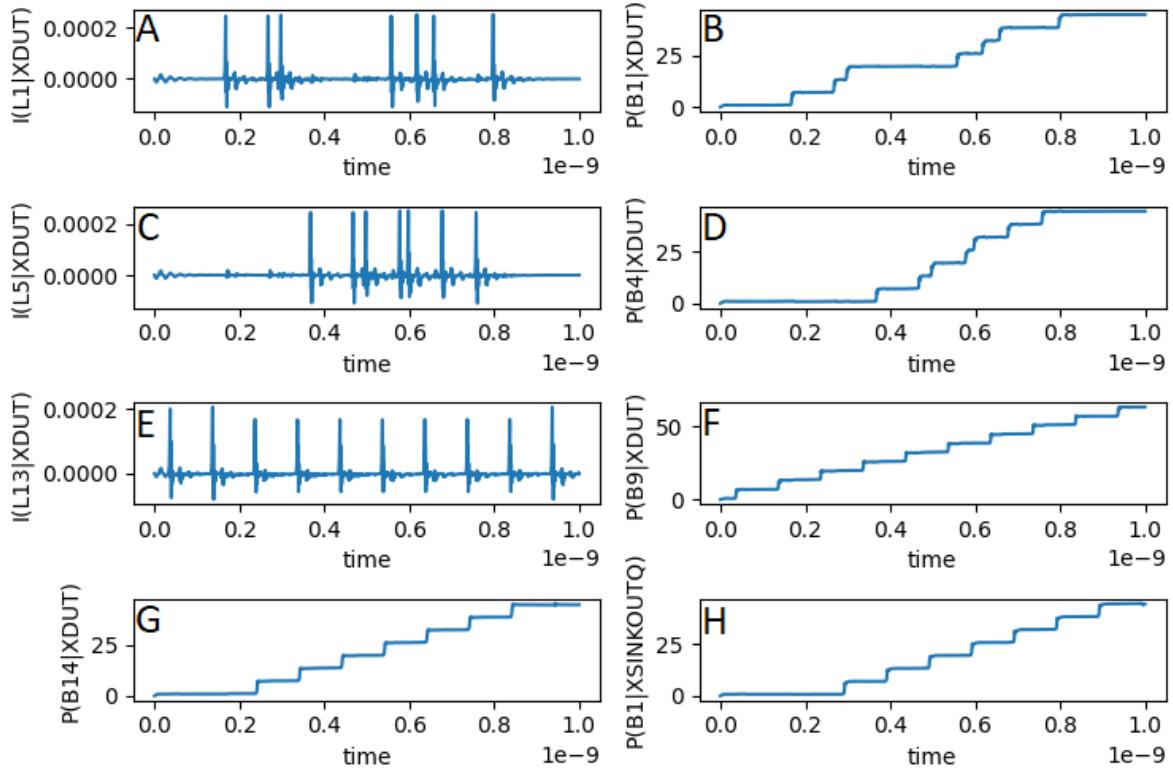


Figure 5.29: RSFQ OR2T analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 3 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 'timescale 1ps/100fs
8 module LSmitll_OR2T_v2p1_optimized (a, b, clk, q);
9
10 input
11   a, b, clk;
12
13 output
14   q;
15
16 reg
17   q;
18
19 real
20   delay_state1_clk_q = 5.5,
21   ct_state0_a_clk = 2.0,
22   ct_state0_b_clk = 2.0,
23   ct_state1_a_clk = 1.2,
24   ct_state1_b_clk = 1.2;
25
26 reg
27   errorsignal_a,
28   errorsignal_b,
29   errorsignal_clk;
30
31 integer
32   outfile,
33   cell_state; // internal state of the cell
34
35 initial
36 begin
37   errorsignal_a = 0;
38   errorsignal_b = 0;
39   errorsignal_clk = 0;
40   cell_state = 0; // Startup state
41   q = 0; // All outputs start at 0
42 end
43
44 always @ (posedge a or negedge a) // execute at positive and negative edges of input
45 begin
46   if ($time > 4) // arbitrary steady-state time)
47     begin
48       if (errorsignal_a == 1'b1) // A critical timing is active for this input
49         begin
50           outfile = $fopen("errors.txt", "a");
51           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
52           ↪ ", $stime);
53           $fclose(outfile);
54           q <= 1'bX; // Set all outputs to unknown
55         end
56       if (errorsignal_a == 0)
57         begin
58           case (cell_state)
59             0: begin
60               cell_state = 1; // Blocking statement -- immediately
61               errorsignal_clk = 1; // Critical timing on this input; assign
62               ↪ immediately
63               errorsignal_clk <= #(ct_state0_a_clk) 0; // Clear error signal
64               ↪ after critical timing expires
65             end
66           1: begin
67             errorsignal_clk = 1; // Critical timing on this input; assign

```

```

65           → immediately
66           errorsignal_clk <= #(ct_state1_a_clk) 0; // Clear error signal
67           → after critical timing expires
68       end
69   endcase
70 end
71 always @(posedge b or negedge b) // execute at positive and negative edges of input
72 begin
73   if ($time>4) // arbitrary steady-state time)
74     begin
75       if (errorsignal_b == 1'b1) // A critical timing is active for this input
76         begin
77           outfile = $fopen("errors.txt", "a");
78           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;%0d_ps.\n
79           ", $stime);
80           $fclose(outfile);
81           q <= 1'bX; // Set all outputs to unknown
82         end
83       if (errorsignal_b == 0)
84         begin
85           case (cell_state)
86             0: begin
87               cell_state = 1; // Blocking statement -- immediately
88               errorsignal_clk = 1; // Critical timing on this input; assign
89               → immediately
90               errorsignal_clk <= #(ct_state0_b_clk) 0; // Clear error signal
91               → after critical timing expires
92             end
93           endcase
94         end
95       end
96     end
97   end
98 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
99 begin
100   if ($time>4) // arbitrary steady-state time)
101     begin
102       if (errorsignal_clk == 1'b1) // A critical timing is active for this input
103         begin
104           outfile = $fopen("errors.txt", "a");
105           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;%0d_ps.\n
106           ", $stime);
107           $fclose(outfile);
108           q <= 1'bX; // Set all outputs to unknown
109         end
110       if (errorsignal_clk == 0)
111         begin
112           case (cell_state)
113             0: begin
114             end
115             1: begin
116               q <= #(delay_state1_clk_q) !q;
117               cell_state = 0; // Blocking statement -- immediately
118             end
119           endcase
120         end
121     end
122 endmodule

```

Listing 5.18: RSFQ OR2T verilog model.

The digital simulation results for the RSFQ OR2T is shown in Fig. 5.30 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 5.31.

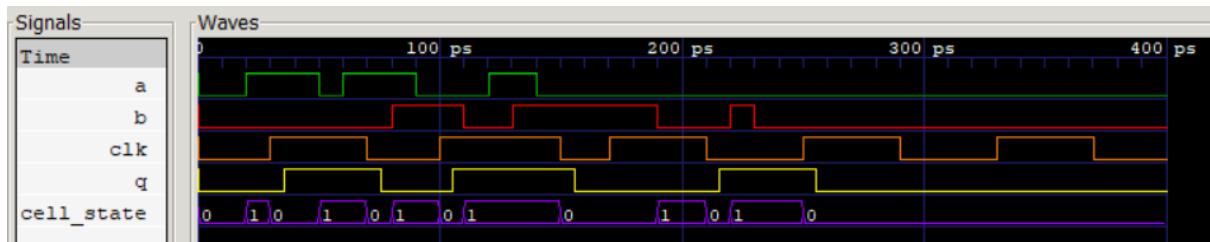


Figure 5.30: RSFQ OR2T digital simulation results.

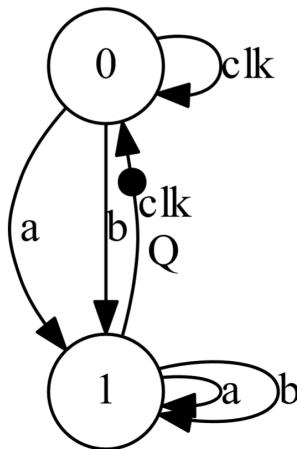


Figure 5.31: RSFQ OR2T Mealy finite state machine diagram.

Power Consumption

Table 5.14: RSFQ OR2T power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2722	4.11
2	2722	8.23
5	2722	20.6
10	2722	41.1
20	2722	82.3
50	2722	206

5.2.3 XORT

The RSFQ XORT cell generates an output pulse exclusively if a pulse from a single input line was received before the clock signal. The XORT cell is designed with integrated PTL transmitters and receivers and is intended to be connected directly to a PTL.

Schematic

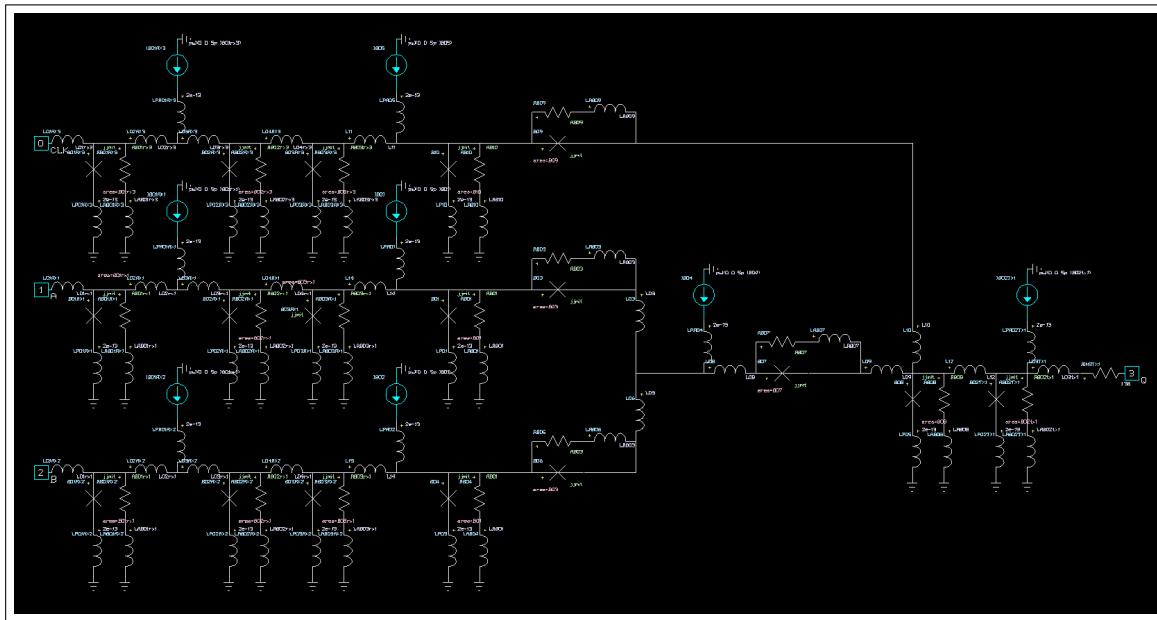


Figure 5.32: Schematic of RSFQ XORT.

Layout

The physical layout for the RSFQ XORT is shown in Fig. 5.33. The layout height is $70 \mu m$ and the width is $100 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

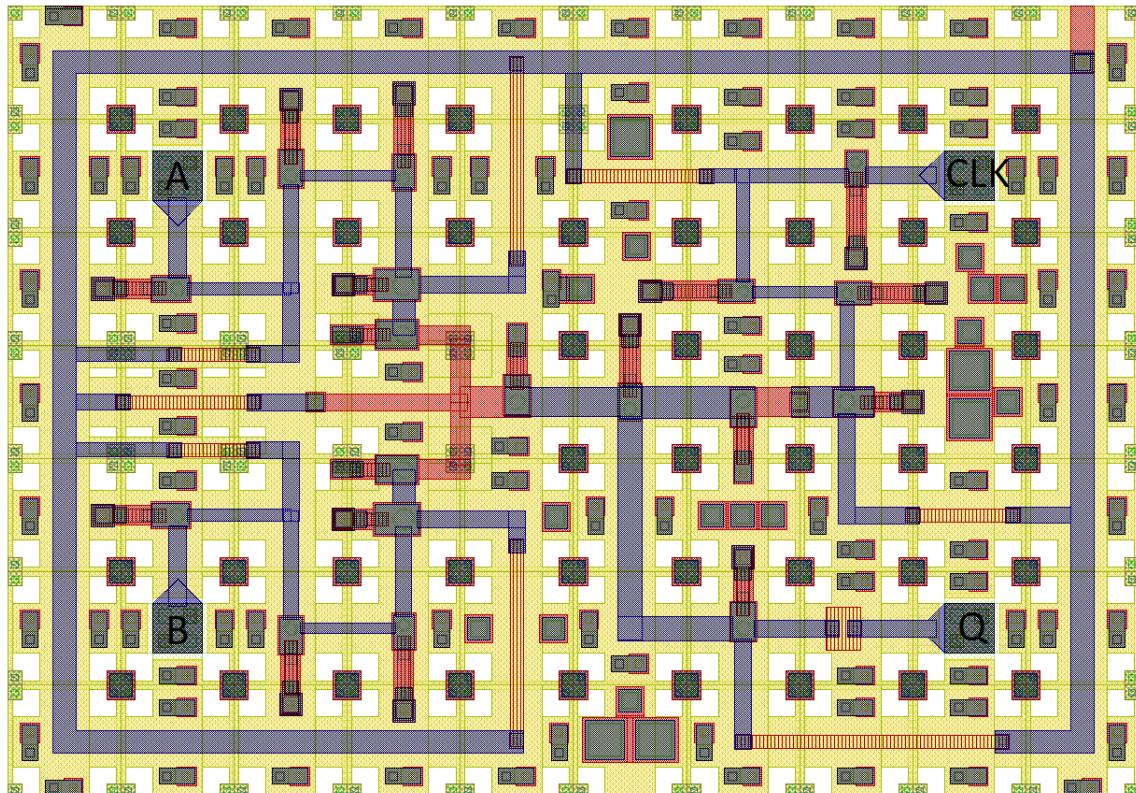


Figure 5.33: RSFQ XORT Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param L17=1.8033p
2  	↳ by InductEx v.6.0.4 on 29-4-21.          65  .param L18=2.2246p
3  * Author: L. Schindler                      66  .param L19=1.7515p
4  * Version: 2.1                               67  .param L20=3.8658p
5  * Last modification date: 29 April 2021     68  .param L21=Lptl
6  * Last modification by: L. Schindler        69
7  *$Ports a b clk q                         70  .param RB1=B0Rs/B1
8 .subckt LSmitll_XORT a b clk q            71  .param RB2=B0Rs/B2
9 .model jjmit jj(rttype=1, vg=2.8mV, cap    72  .param RB3=B0Rs/B3
   ↳ =0.07pF, r0=160, rn=16, icrit=0.1mA     73  .param RB4=B0Rs/B4
   ↳ )                                         74  .param RB5=B0Rs/B5
10 .param B0=1                                75  .param RB6=B0Rs/B6
11 .param Ic0=0.0001                           76  .param RB7=B0Rs/B7
12 .param IcRs=100u*6.859904418                77  .param RB8=B0Rs/B8
13 .param B0Rs=IcRs/Ic0*B0                     78  .param RB9=B0Rs/B9
14 .param Rsheet=2                            79  .param RB10=B0Rs/B10
15 .param Lsheet=1.13e-12                      80  .param RB11=B0Rs/B11
16 .param Lptl=2p                             81  .param RB12=B0Rs/B12
17 .param LB=2p                               82  .param RB13=B0Rs/B13
18 .param LP=0.5p                            83  .param RB14=B0Rs/B14
19
20 .param B1=1.21                            84  .param RB15=B0Rs/B15
21 .param B2=1.16                            85  .param RB16=B0Rs/B16
22 .param B3=0.90                            86  .param RB17=B0Rs/B17
23 .param B4=2.80                            87  .param RB18=B0Rs/B18
24 .param B5=1.92
25 .param B6=B1
26 .param B7=B2
27 .param B8=B3
28 .param B9=B4
29 .param B10=B5
30 .param B11=0.72
31 .param B12=0.77
32 .param B13=0.83
33 .param B14=1.69
34 .param B15=1.29
35 .param B16=1.49
36 .param B17=0.93
37 .param B18=1.37
38
39 .param IB1=230u
40 .param IB2=89u
41 .param IB3=IB1
42 .param IB4=IB2
43 .param IB5=132u
44 .param IB6=178u
45 .param IB7=134u
46 .param IB8=66u
47
48 .param L1=Lptl
49 .param L2=2.1529p
50 .param L3=1.9729p
51 .param L4=2.3966p
52 .param L5=1.6354p
53 .param L6=2.2793p
54 .param L7=L1
55 .param L8=L2
56 .param L9=L3
57 .param L10=L4
58 .param L11=L5
59 .param L12=L6
60 .param L13=Lptl
61 .param L14=2.2381p
62 .param L15=2.0205p
63 .param L16=2.0178p
64  .param L17=1.8033p
65  .param L18=2.2246p
66  .param L19=1.7515p
67  .param L20=3.8658p
68  .param L21=Lptl
69
70  .param RB1=B0Rs/B1
71  .param RB2=B0Rs/B2
72  .param RB3=B0Rs/B3
73  .param RB4=B0Rs/B4
74  .param RB5=B0Rs/B5
75  .param RB6=B0Rs/B6
76  .param RB7=B0Rs/B7
77  .param RB8=B0Rs/B8
78  .param RB9=B0Rs/B9
79  .param RB10=B0Rs/B10
80  .param RB11=B0Rs/B11
81  .param RB12=B0Rs/B12
82  .param RB13=B0Rs/B13
83  .param RB14=B0Rs/B14
84  .param RB15=B0Rs/B15
85  .param RB16=B0Rs/B16
86  .param RB17=B0Rs/B17
87  .param RB18=B0Rs/B18
88
89  .param LRB1=(RB1/Rsheet)*Lsheet
90  .param LRB2=(RB2/Rsheet)*Lsheet
91  .param LRB3=(RB3/Rsheet)*Lsheet
92  .param LRB4=(RB4/Rsheet)*Lsheet
93  .param LRB5=(RB5/Rsheet)*Lsheet
94  .param LRB6=(RB6/Rsheet)*Lsheet
95  .param LRB7=(RB7/Rsheet)*Lsheet
96  .param LRB8=(RB8/Rsheet)*Lsheet
97  .param LRB9=(RB9/Rsheet)*Lsheet
98  .param LRB10=(RB10/Rsheet)*Lsheet
99  .param LRB11=(RB11/Rsheet)*Lsheet
100 .param LRB12=(RB12/Rsheet)*Lsheet
101 .param LRB13=(RB13/Rsheet)*Lsheet
102 .param LRB14=(RB14/Rsheet)*Lsheet
103 .param LRB15=(RB15/Rsheet)*Lsheet
104 .param LRB16=(RB16/Rsheet)*Lsheet
105 .param LRB17=(RB17/Rsheet)*Lsheet
106 .param LRB18=(RB18/Rsheet)*Lsheet
107
108 .param LB1=LB
109 .param LB2=LB
110 .param LB3=LB
111 .param LB4=LB
112 .param LB5=LB
113 .param LB6=LB
114 .param LB7=LB
115 .param LB8=LB
116
117 .param LP1=LP
118 .param LP2=LP
119 .param LP3=LP
120 .param LP4=LP
121 .param LP6=LP
122 .param LP7=LP
123 .param LP8=LP
124 .param LP9=LP
125 .param LP11=LP
126 .param LP12=LP
127 .param LP13=LP
128 .param LP14=LP
129 .param LP17=LP

```

```

130 | .param LP18=LP
131 | B1 2 3 jjmit area=B1
132 | B2 6 7 jjmit area=B2
133 | B3 8 9 jjmit area=B3
134 | B4 10 11 jjmit area=B4
135 | B5 10 13 jjmit area=B5
136 | B6 16 17 jjmit area=B6
137 | B7 20 21 jjmit area=B7
138 | B8 22 23 jjmit area=B8
139 | B9 24 25 jjmit area=B9
140 | B10 24 27 jjmit area=B10
141 | B11 32 33 jjmit area=B11
142 | B12 36 37 jjmit area=B12
143 | B13 38 39 jjmit area=B13
144 | B14 40 41 jjmit area=B14
145 | B15 40 43 jjmit area=B15
146 | B16 29 30 jjmit area=B16
147 | B17 30 44 jjmit area=B17
148 | B18 45 46 jjmit area=B18
149 |
150 |
151 | IB1 0 5 pwl(0 0 5p IB1)
152 | IB2 0 12 pwl(0 0 5p IB2)
153 | IB3 0 19 pwl(0 0 5p IB3)
154 | IB4 0 26 pwl(0 0 5p IB4)
155 | IB5 0 35 pwl(0 0 5p IB5)
156 | IB6 0 42 pwl(0 0 5p IB6)
157 | IB7 0 28 pwl(0 0 5p IB7)
158 | IB8 0 47 pwl(0 0 5p IB8)
159 |
160 | L1 a 2 1.453E-12
161 | L2 2 4 2.162E-12
162 | L3 4 6 1.962E-12
163 | L4 6 8 2.392E-12
164 | L5 8 10 1.634E-12
165 | L6 13 14 2.263E-12
166 | L7 b 16 1.446E-12
167 | L8 16 18 2.158E-12
168 | L9 18 20 1.959E-12
169 | L10 20 22 2.389E-12
170 | L11 22 24 1.63E-12
171 | L12 27 14 2.268E-12
172 | L13 clk 32 1.608E-12
173 | L14 32 34 2.222E-12
174 | L15 34 36 2.02E-12
175 | L16 36 38 2.019E-12
176 | L17 38 40 1.796E-12
177 | L18 43 30 2.204E-12
178 | L19 14 29 1.754E-12
179 | L20 30 45 3.883E-12
180 | L21 45 48 1.589E-12
181 |
182 | LP1 3 0 5.123E-13
183 | LP2 7 0 5.755E-13
184 | LP3 9 0 5.799E-13
185 | LP4 11 0 4.826E-13
186 | LP6 17 0 5.124E-13
187 | LP7 21 0 5.734E-13
188 | LP8 23 0 5.803E-13
189 | LP9 25 0 4.745E-13
190 | LP11 33 0 5.105E-13
191 | LP12 37 0 6.091E-13
192 | LP13 39 0 5.955E-13
193 | LP14 41 0 5.33E-13
194 | LP17 44 0 6.204E-13
195 | LP18 46 0 5.407E-13
196 |
197 | LB1 4 5 1.862E-12
198 | LB2 10 12 2.299E-12
199 | LB3 18 19 1.858E-12
200 | LB4 24 26 2.318E-12
201 | LB5 34 35 9.457E-13
202 | LB6 40 42 2.935E-12
203 | LB7 14 28 4.089E-12
204 | LB8 45 47 2.014E-12
205 |
206 | RD 48 q 1.36
207 |
208 | RB1 2 102 RB1
209 | LRB1 102 0 LRB1
210 | RB2 6 106 RB2
211 | LRB2 106 0 LRB2
212 | RB3 8 108 RB3
213 | LRB3 108 0 LRB3
214 | RB4 10 110 RB4
215 | LRB4 110 0 LRB4
216 | RB5 10 113 RB5
217 | LRB5 113 13 LRB5
218 | RB6 16 116 RB6
219 | LRB6 116 0 LRB6
220 | RB7 20 120 RB7
221 | LRB7 120 0 LRB7
222 | RB8 22 122 RB8
223 | LRB8 122 0 LRB8
224 | RB9 24 124 RB9
225 | LRB9 124 0 LRB9
226 | RB10 24 127 RB10
227 | LRB10 127 27 LRB10
228 | RB11 32 132 RB11
229 | LRB11 132 0 LRB11
230 | RB12 36 136 RB12
231 | LRB12 136 0 LRB12
232 | RB13 38 138 RB13
233 | LRB13 138 0 LRB13
234 | RB14 40 140 RB14
235 | LRB14 140 0 LRB14
236 | RB15 40 143 RB15
237 | LRB15 143 43 LRB15
238 | RB16 29 129 RB16
239 | LRB16 129 30 LRB16
240 | RB17 30 130 RB17
241 | LRB17 130 0 LRB17
242 | RB18 45 145 RB18
243 | LRB18 145 0 LRB18
244 | .ends

```

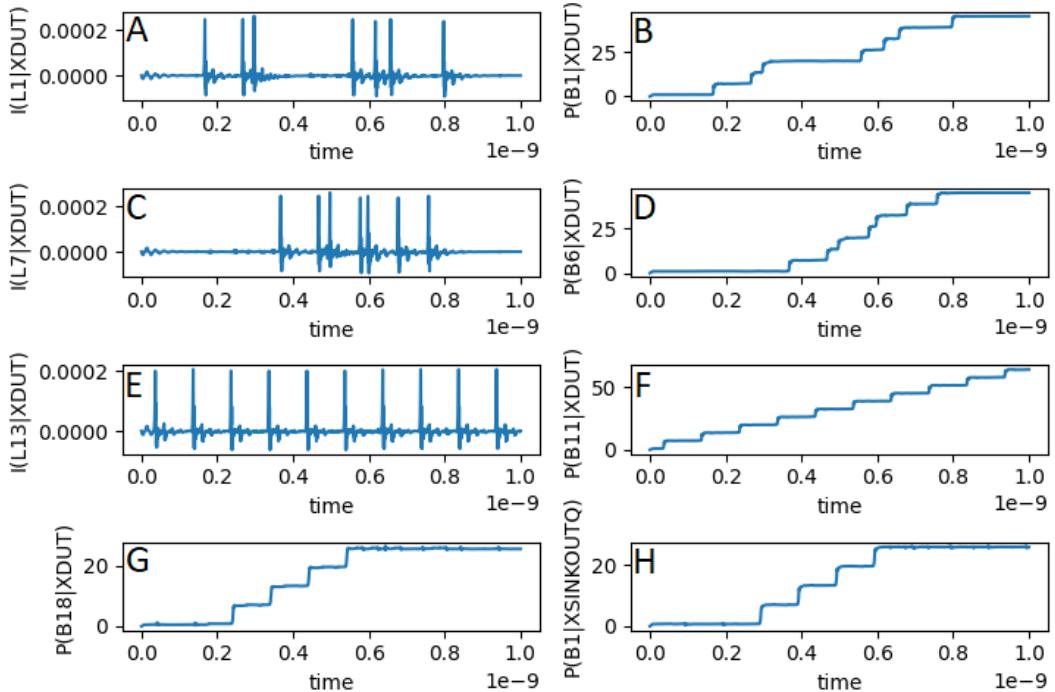
Listing 5.19: RSFQ XORT JoSIM netlist.

Table 5.15: RSFQ XORT pin list.

Pin	Description
a	Data input
b	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ XORT using JoSIM is shown in Fig. 5.34. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b**,
- (d) the phase over the input JJ of pin **b**,
- (e) the current through the input inductor connected to pin **clk**,
- (f) the phase over the input JJ of pin **clk**,
- (g) the phase over the output JJ of pin **q**, and
- (h) the phase over the input JJ of the load circuit connected via a PTL to pin **q**.

**Figure 5.34:** RSFQ XORT analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 4 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 //
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 //
15 'timescale 1ps/100fs
16 module LSmitll_XORT_v2p1_optimized (a, b, clk, q);
17
18 input
19   a, b, clk;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state1_clk_q = 5.5,
29   delay_state2_clk_q = 5.5,
30   ct_state0_a_clk = 2.3,
31   ct_state0_b_clk = 2.3,
32   ct_state1_a_b = 9.0,
33   ct_state1_a_clk = 11.4,
34   ct_state1_b_b = 3.3,
35   ct_state1_clk_b = 3.9,
36   ct_state2_a_a = 3.3,
37   ct_state2_b_a = 9.1,
38   ct_state2_b_clk = 11.6,
39   ct_state2_clk_a = 3.9;
40
41 reg
42   errorsignal_a,
43   errorsignal_b,
44   errorsignal_clk;
45
46 integer
47   outfile,
48   cell_state; // internal state of the cell
49
50 initial
51 begin
52   errorsignal_a = 0;
53   errorsignal_b = 0;
54   errorsignal_clk = 0;
55   cell_state = 0; // Startup state
56   q = 0; // All outputs start at 0
57 end
58
59 always @ (posedge a or negedge a) // execute at positive and negative edges of input
60 begin
61   if ($time>4) // arbitrary steady-state time)
62     begin
63       if (errorsignal_a == 1'b1) // A critical timing is active for this input
64         begin
65           outfile = $fopen("errors.txt", "a");
66           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
67             ↳ ", $stime);

```

```

67          $fclose(outfile);
68          q <= 1'bX; // Set all outputs to unknown
69      end
70      if (errorsignal_a == 0)
71      begin
72          case (cell_state)
73          0: begin
74              cell_state = 1; // Blocking statement -- immediately
75              errorsignal_clk = 1; // Critical timing on this input; assign
76                  // immediately
77              errorsignal_clk <= #(ct_state0_a_clk) 0; // Clear error signal
78                  // after critical timing expires
79          end
80          1: begin
81              errorsignal_b = 1; // Critical timing on this input; assign
82                  // immediately
83              errorsignal_b <= #(ct_state1_a_b) 0; // Clear error signal
84                  // after critical timing expires
85              errorsignal_clk = 1; // Critical timing on this input; assign
86                  // immediately
87              errorsignal_clk <= #(ct_state1_a_clk) 0; // Clear error signal
88                  // after critical timing expires
89          end
90          endcase
91      end
92  end
93
94 always @ (posedge b or negedge b) // execute at positive and negative edges of input
95 begin
96     if ($time>4) // arbitrary steady-state time)
97     begin
98         if (errorsignal_b == 1'b1) // A critical timing is active for this input
99         begin
100             outfile = $fopen("errors.txt", "a");
101             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
102                         // ", $stime);
103             $fclose(outfile);
104             q <= 1'bX; // Set all outputs to unknown
105         end
106         if (errorsignal_b == 0)
107         begin
108             case (cell_state)
109             0: begin
110                 cell_state = 2; // Blocking statement -- immediately
111                 errorsignal_clk = 1; // Critical timing on this input; assign
112                     // immediately
113                 errorsignal_clk <= #(ct_state0_b_clk) 0; // Clear error signal
114                     // after critical timing expires
115             end
116             1: begin
117                 cell_state = 0; // Blocking statement -- immediately
118                 errorsignal_b = 1; // Critical timing on this input; assign
119                     // immediately
120                 errorsignal_b <= #(ct_state1_b_b) 0; // Clear error signal
121                     // after critical timing expires
122             end
123             2: begin
124                 errorsignal_a = 1; // Critical timing on this input; assign
125                     // immediately
126                 errorsignal_a <= #(ct_state2_b_a) 0; // Clear error signal
127                     // after critical timing expires
128                 errorsignal_clk = 1; // Critical timing on this input; assign

```

```

122           ↪ immediately
123           errorsignal_clk <= #(ct_state2_b_clk) 0; // Clear error signal
124           ↪ after critical timing expires
125       end
126   endcase
127 end
128
129 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
130 begin
131   if ($time>4) // arbitrary steady-state time)
132   begin
133     if (errorsignal_clk == 1'b1) // A critical timing is active for this input
134     begin
135       outfile = $fopen("errors.txt", "a");
136       $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n
137           ↪ ", $stime);
138       $fclose(outfile);
139       q <= 1'bX; // Set all outputs to unknown
140     end
141     if (errorsignal_clk == 0)
142     begin
143       case (cell_state)
144         0: begin
145           end
146         1: begin
147           q <= #(delay_state1_clk_q) !q;
148           cell_state = 0; // Blocking statement -- immediately
149           errorsignal_b = 1; // Critical timing on this input; assign
150           ↪ immediately
151           errorsignal_b <= #(ct_state1_clk_b) 0; // Clear error signal
152           ↪ after critical timing expires
153         end
154         2: begin
155           q <= #(delay_state2_clk_q) !q;
156           cell_state = 0; // Blocking statement -- immediately
157           errorsignal_a = 1; // Critical timing on this input; assign
158           ↪ immediately
159           errorsignal_a <= #(ct_state2_clk_a) 0; // Clear error signal
160           ↪ after critical timing expires
161         end
162       endcase
163     end
164   end
165 endmodule

```

Listing 5.20: RSFQ XORT verilog model.

The digital simulation results for the RSFQ XORT is shown in Fig. 5.35 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 5.36.



Figure 5.35: RSFQ XORT digital simulation results.

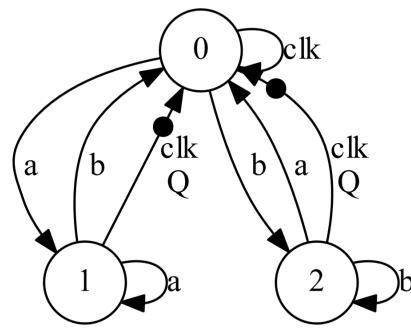


Figure 5.36: RSFQ XORT Mealy finite state machine diagram.

Power Consumption

Table 5.16: RSFQ XORT power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2985	5.18
2	2985	10.4
5	2985	25.9
10	2985	51.8
20	2985	104
50	2985	259

5.2.4 NOTT

The RSFQ NOTT cell is a signal inverting cell driven by a clock pulse signal line. The NOTT cell is designed with integrated PTL transmitters and receivers and is intended for direct connections with PTLs.

Schematic

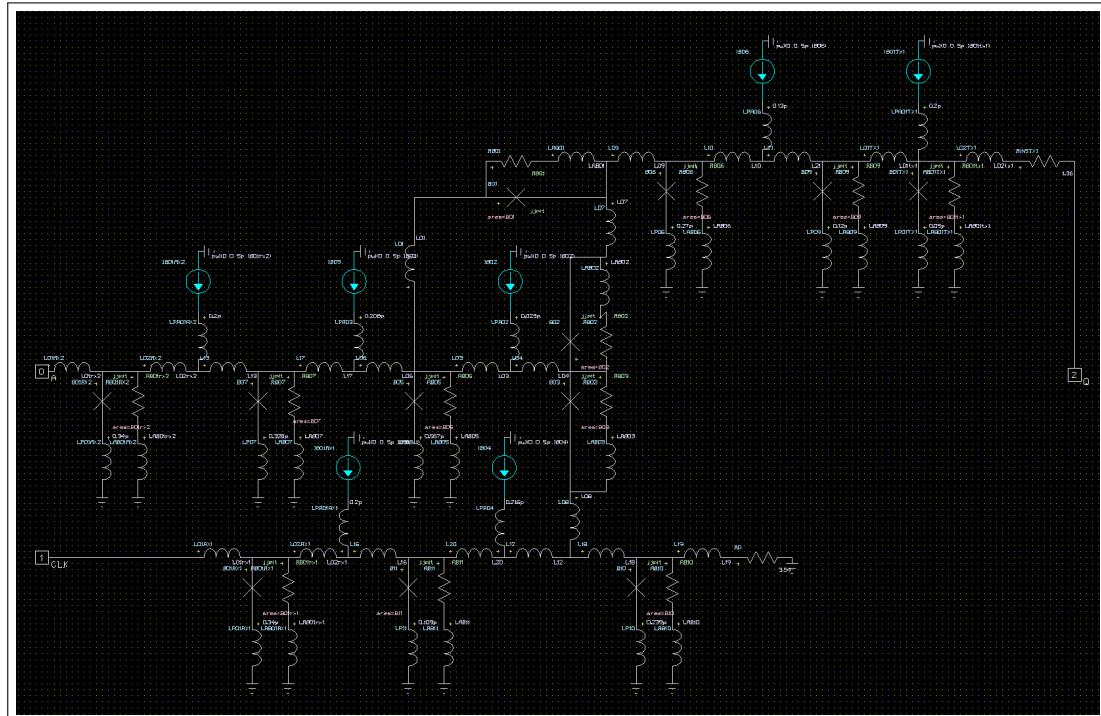


Figure 5.37: Schematic of RSFQ NOTT.

Layout

The physical layout for the RSFQ NOTT is shown in Fig. 5.38. The layout height is $70 \mu m$ and the width is $100 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

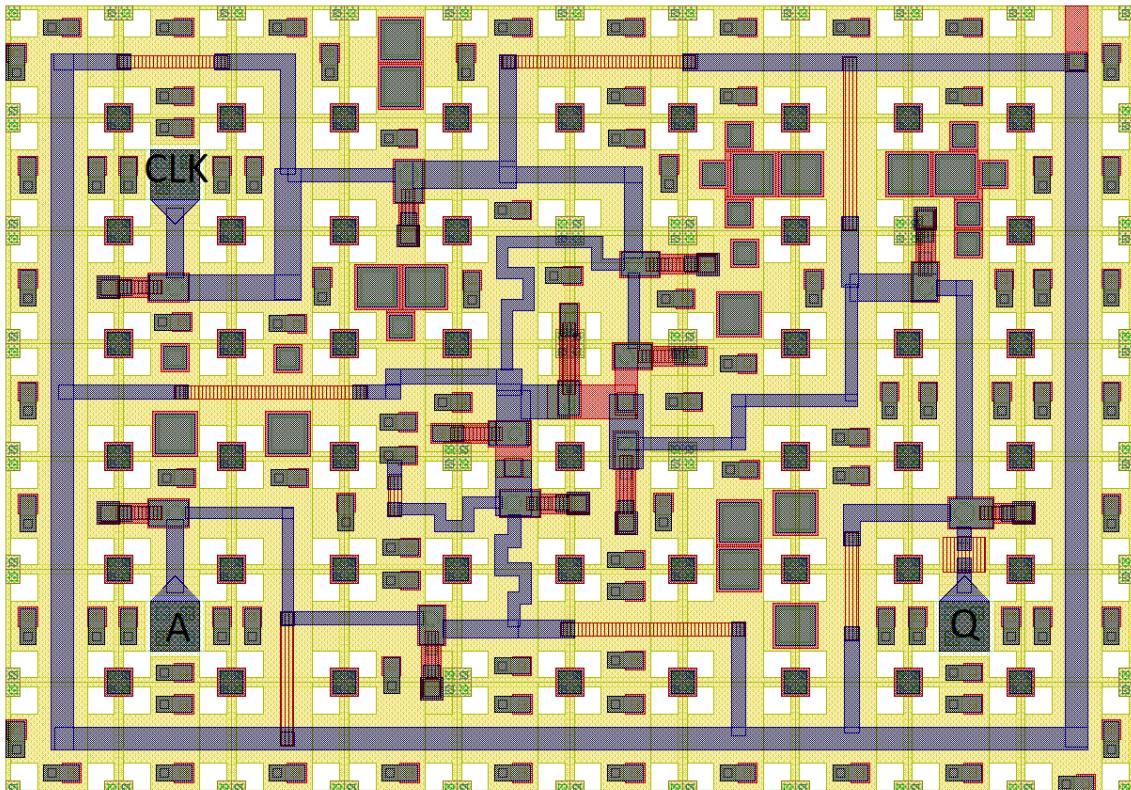


Figure 5.38: RSFQ NOTT Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param L16=2.2847p
2   ↪ by InductEx v.6.0 on 2021/06/30.          65  .param L17=0.49986p
3  * Author: L. Schindler                      66  .param L18=0.28417p
4  * Version: 2.1                               67  .param L19=7.3651p
5  * Last modification date: 30 June 2021       68  .param L20=0.74611p
6  * Last modification by: L. Schindler        69  .param L21=4.5195p
7  *$Ports a clk q                           70  .param L22=Lpt1
8 .subckt LSmitll_NOTT a clk q
9 .model jjmit jj(rtype=1, vg=2.8mV, cap
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA
   ↪ )
10 .param B0=1
11 .param Ic0=0.0001
12 .param IcRs=100u*6.859904418
13 .param B0Rs=IcRs/Ic0*B0
14 .param Rsheet=2
15 .param Lsheet=1.13e-12
16 .param RD=1.36
17 .param LB=0.2p
18 .param Lpt1=2p
19 .param LP=0.5p
20
21 .param B1=1.26
22 .param B2=1.42
23 .param B3=1.72
24 .param B4=1.22
25 .param B5=0.77
26 .param B6=1.40
27 .param B7=2.5
28 .param B8=2
29 .param B9=1.35
30 .param B10=1.04
31 .param B11=1.41
32 .param B12=2.5
33
34 .param IB1=173u
35 .param IB2=123u
36 .param IB3=230u
37 .param IB4=112u
38 .param IB5=112u
39 .param IB6=108u
40 .param IB7=187u
41
42 .param LB1=LB
43 .param LB2=LB
44 .param LB3=LB
45 .param LB4=LB
46 .param LB5=LB
47 .param LB6=LB
48 .param LB7=LB
49
50 .param L1=Lpt1
51 .param L2=4.4718p
52 .param L3=2.6117p
53 .param L4=1.1676p
54 .param L5=2.6532p
55 .param L7=3.1681p
56 .param L8=0.86946p
57 .param L9=Lpt1
58 .param L10=2.5468p
59 .param L11=2.1566p
60 .param L12=0.99180p
61 .param L13=3.286p
62 .param L14=6.5962p
63 .param L15=0.42413p
64  .param L16=2.2847p
65  .param L17=0.49986p
66  .param L18=0.28417p
67  .param L19=7.3651p
68  .param L20=0.74611p
69  .param L21=4.5195p
70  .param L22=Lpt1
71
72 .param RB1=B0Rs/B1
73 .param RB2=B0Rs/B2
74 .param RB3=B0Rs/B3
75 .param RB4=B0Rs/B4
76 .param RB5=B0Rs/B5
77 .param RB6=B0Rs/B6
78 .param RB7=B0Rs/B7
79 .param RB8=B0Rs/B8
80 .param RB9=B0Rs/B9
81 .param RB10=B0Rs/B10
82 .param RB11=B0Rs/B11
83 .param RB12=B0Rs/B12
84
85 .param LRB1=(RB1/Rsheet)*Lsheet
86 .param LRB2=(RB2/Rsheet)*Lsheet
87 .param LRB3=(RB3/Rsheet)*Lsheet
88 .param LRB4=(RB4/Rsheet)*Lsheet
89 .param LRB5=(RB5/Rsheet)*Lsheet
90 .param LRB6=(RB6/Rsheet)*Lsheet
91 .param LRB7=(RB7/Rsheet)*Lsheet
92 .param LRB8=(RB8/Rsheet)*Lsheet
93 .param LRB9=(RB9/Rsheet)*Lsheet
94 .param LRB10=(RB10/Rsheet)*Lsheet
95 .param LRB11=(RB11/Rsheet)*Lsheet
96 .param LRB12=(RB12/Rsheet)*Lsheet
97
98 B1 2 3 jjmit area=B1
99 B2 6 7 jjmit area=B2
100 B3 10 12 jjmit area=B3
101 B4 15 14 jjmit area=B4
102 B5 15 31 jjmit area=B5
103 B6 17 18 jjmit area=B6
104 B7 21 22 jjmit area=B7
105 B8 25 26 jjmit area=B8
106 B9 29 30 jjmit area=B9
107 B10 32 33 jjmit area=B10
108 B11 36 37 jjmit area=B11
109 B12 38 40 jjmit area=B12
110
111 IB1 0 5 pw1(0 0 5p IB1)
112 IB2 0 9 pw1(0 0 5p IB2)
113 IB3 0 20 pw1(0 0 5p IB3)
114 IB4 0 24 pw1(0 0 5p IB4)
115 IB5 0 28 pw1(0 0 5p IB5)
116 IB6 0 35 pw1(0 0 5p IB6)
117 IB7 0 39 pw1(0 0 5p IB7)
118
119 L1 a 2 1.515E-012
120 L2 2 4 4.518E-012
121 L3 4 6 2.597E-012
122 L4 6 8 1.171E-012
123 L5 8 10 2.698E-012
124 L7 10 13 3.034E-012
125 L8 10 14 9.587E-013
126 L9 clk 17 1.476E-012
127 L10 17 19 2.524E-012
128 L11 19 21 2.154E-012
129 L12 21 23 1.005E-012

```

```

130 | L13 23 25 3.241E-012      160 | LP12 40 0 3.86E-013
131 | L14 25 27 6.51E-012      161 |
132 | L15 27 15 3.051E-013      162 | RB1 2 102 RB1
133 | L16 25 29 2.268E-012      163 | LRB1 102 0 LRB1
134 | L17 30 31 9.642E-013      164 | RB2 6 106 RB2
135 | L18 30 32 4.92E-013      165 | LRB2 106 0 LRB2
136 | L19 32 34 7.312E-012      166 | RB3 10 110 RB3
137 | L20 34 36 6.792E-013      167 | LRB3 110 0 LRB3
138 | L21 36 38 4.551E-012      168 | RB4 14 114 RB4
139 | L22 38 41 5.96E-013      169 | LRB4 114 15 LRB4
140 |
141 | RN 13 0 3.54
142 | RD 41 q RD
143 |
144 | LB1 4 5 3.554E-013
145 | LB2 8 9 1.086E-012
146 | LB3 19 20 3.259E-012
147 | LB4 23 24 1.957E-012
148 | LB5 27 28 2.939E-012
149 | LB6 34 35 1.297E-012
150 | LB7 38 39 2.4E-012
151 |
152 | LP1 3 0 5.147E-013
153 | LP2 7 0 5.081E-013
154 | LP3 12 0 5.184E-013
155 | LP6 18 0 4.98E-013
156 | LP7 22 0 4.82E-013
157 | LP8 26 0 5.034E-013
158 | LP10 33 0 6.009E-013
159 | LP11 37 0 5.044E-013
160 | LRB5 115 31 LRB5
161 |
162 | RB6 17 117 RB6
163 | LRB6 117 0 LRB6
164 | RB7 21 121 RB7
165 | LRB7 121 0 LRB7
166 | RB8 25 125 RB8
167 | LRB8 125 0 LRB8
168 | RB9 29 129 RB9
169 | LRB9 129 30 LRB9
170 | RB10 32 132 RB10
171 | LRB10 132 0 LRB10
172 |
173 | RB11 36 136 RB11
174 | LRB11 136 0 LRB11
175 | RB12 38 138 RB12
176 | LRB12 138 0 LRB12
177 |
178 | .ends
179 |
180 |
181 |
182 |
183 |
184 |
185 |
186 |

```

Listing 5.21: RSFQ NOTT JoSIM netlist.**Table 5.17:** RSFQ NOTT pin list.

Pin	Description
a	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ NOTT using JoSIM is shown in Fig. 5.39. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **clk**,
- (d) the phase over the input JJ of pin **clk**,
- (e) the phase over the output JJ of pin **q**, and
- (f) the phase over the input JJ of the load circuit connected via a PTL to pin **q**.

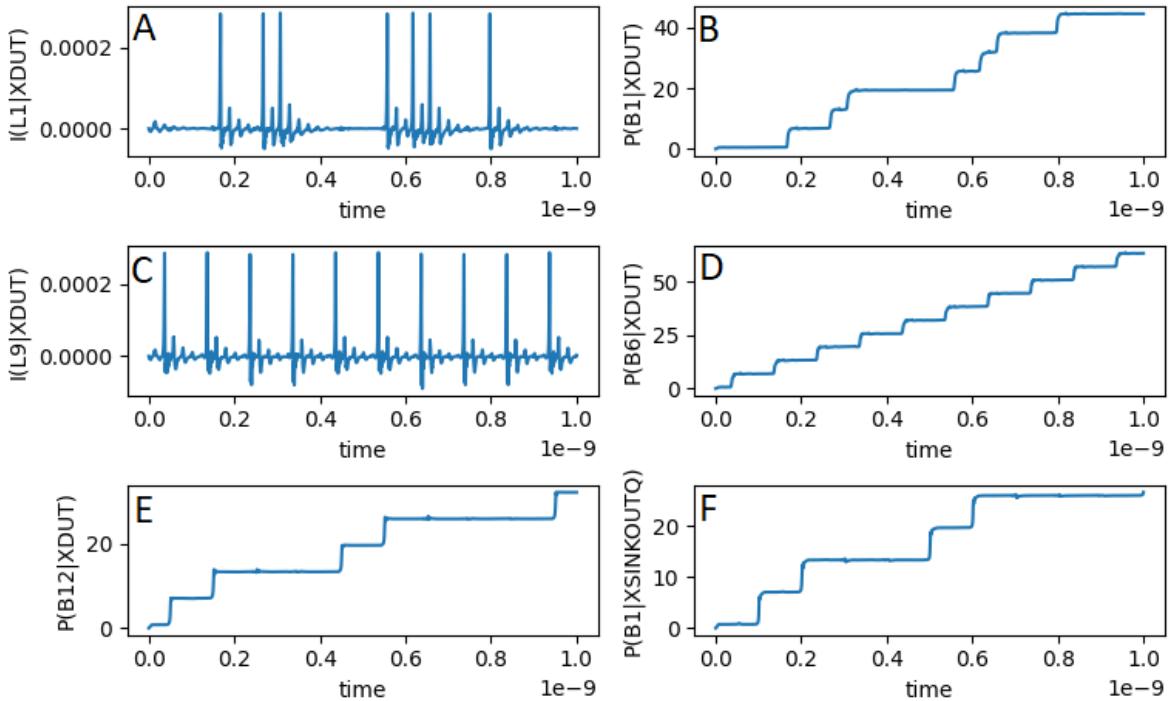


Figure 5.39: RSFQ NOTT analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 30 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 //
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 //
15 'timescale 1ps/100fs
16 module LSmitll_NOTT_v2p1_optimized (a, clk, q);
17
18 input
19   a, clk;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state0_clk_q = 13.5,
29   ct_state0_clk_a = 8.0,
30   ct_state0_clk_clk = 11.6,
31   ct_state1_a_clk = 7.5;
32
33 reg
34   errorsignal_a,
35   errorsignal_clk;
36
37 integer
38   outfile,
39   cell_state; // internal state of the cell
40
41 initial
42 begin
43   errorsignal_a = 0;
44   errorsignal_clk = 0;
45   cell_state = 0; // Startup state
46   q = 0; // All outputs start at 0
47 end
48
49 always @ (posedge a or negedge a) // execute at positive and negative edges of input
50 begin
51   if ($time > 4) // arbitrary steady-state time)
52     begin
53       if (errorsignal_a == 1'b1) // A critical timing is active for this input
54         begin
55           outfile = $fopen("errors.txt", "a");
56           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
57           ↪ ", $stime);
58           $fclose(outfile);
59           q <= 1'bX; // Set all outputs to unknown
60         end
61       if (errorsignal_a == 0)
62         begin
63           case (cell_state)
64             0: begin
65               cell_state = 1; // Blocking statement -- immediately
66             end
67             1: begin

```

```

67           errorsignal_clk = 1; // Critical timing on this input; assign
68           ↪ immediately
69           errorsignal_clk <= #(ct_state1_a_clk) 0; // Clear error signal
70           ↪ after critical timing expires
71       end
72   endcase
73 end
74
75 always @ (posedge clk or negedge clk) // execute at positive and negative edges of input
76 begin
77     if ($time>4) // arbitrary steady-state time)
78     begin
79         if (errorsignal_clk == 1'b1) // A critical timing is active for this input
80             begin
81                 outfile = $fopen("errors.txt", "a");
82                 $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
83                             ↪ ", $stime);
84                 $fclose(outfile);
85                 q <= 1'bX; // Set all outputs to unknown
86             end
87         if (errorsignal_clk == 0)
88             begin
89                 case (cell_state)
90                     0: begin
91                         q <= #(delay_state0_clk_q) !q;
92                         errorsignal_a = 1; // Critical timing on this input; assign
93                         ↪ immediately
94                         errorsignal_a <= #(ct_state0_clk_a) 0; // Clear error signal
95                         ↪ after critical timing expires
96                         errorsignal_clk = 1; // Critical timing on this input; assign
97                         ↪ immediately
98                         errorsignal_clk <= #(ct_state0_clk_clk) 0; // Clear error
99                         ↪ signal after critical timing expires
100                    end
101                1: begin
102                    cell_state = 0; // Blocking statement -- immediately
103                end
104            endcase
105        end
106    end
107 end
108
109 endmodule

```

Listing 5.22: RSFQ NOTT verilog model.

The digital simulation results for the RSFQ NOTT is shown in Fig. 5.40 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 5.41.

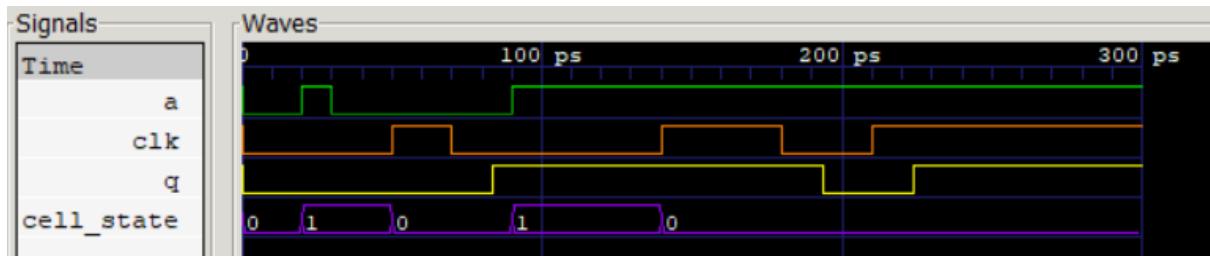


Figure 5.40: RSFQ NOTT digital simulation results.

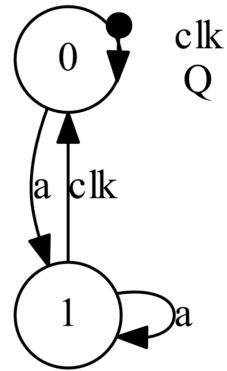


Figure 5.41: RSFQ NOTT Mealy finite state machine diagram.

Power Consumption

Table 5.18: RSFQ NOTT power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2545	3.82
2	2545	7.65
5	2545	19.1
10	2545	38.2
20	2545	76.5
50	2545	191

5.2.5 XNORT

The RSFQ XNORT cell is a combination of a NOTT and XORT cell. It generates an output pulse exclusively if no input pulse was received or if a pulse from both input lines was received before the clock signal. The XNORT cell is designed with integrated PTL transmitters and receivers and is intended for direct connections with PTLs.

Schematic

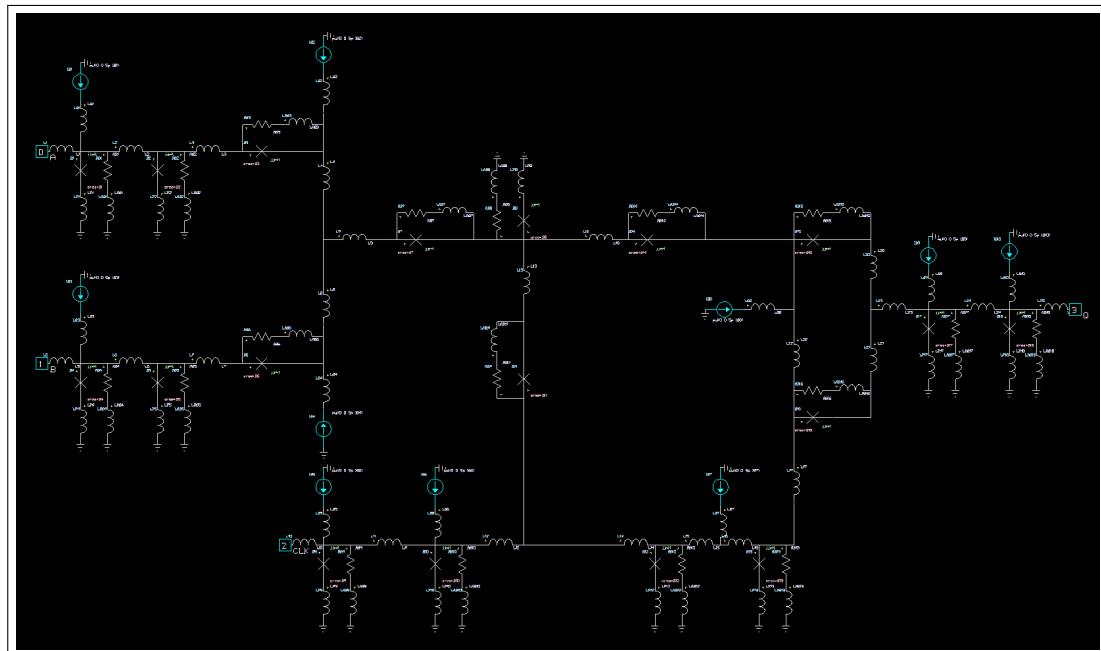


Figure 5.42: Schematic of RSFQ XNORT.

Layout

The physical layout for the RSFQ XNORT is shown in Fig. 5.43. The layout height is $70 \mu m$ and the width is $90 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

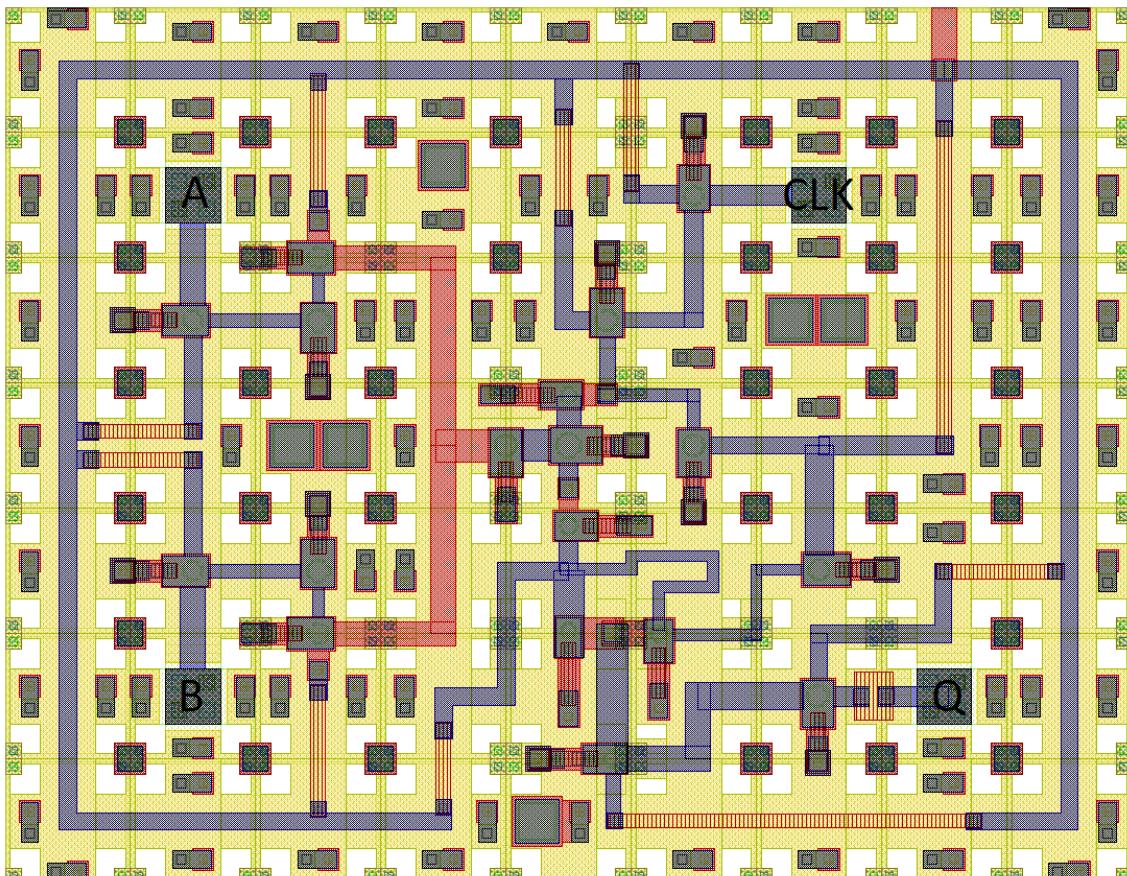


Figure 5.43: RSFQ XNORT Layout.

Analog model

```

1  * Author: L. Schindler
2  * Version: 2.1
3  * Last modification date: 7 June 2021
4  * Last modification by: L. Schindler
5
6  *$Ports a b clk q
7 .subckt LSMITLL_XNORT a b clk q
8 .model jjmit jj(rtype=1, vg=2.8mV, cap
  ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA
  ↪ )
9 .param Phi0=2.067833848E-15
10 .param B0=1
11 .param Ic0=0.0001
12 .param IcRs=100u*6.859904418
13 .param B0Rs=IcRs/Ic0*B0
14 .param Rsheet=2
15 .param Lsheet=1.13e-12
16 .param LP=0.2p
17 .param IC=2.5
18 .param Lptl=2p
19 .param LB=2p
20 .param BiasCoef=0.7
21
22 .param B1=2.0
23 .param B2=2.74
24 .param B3=2.23
25 .param B4=B1
26 .param B5=B2
27 .param B6=B3
28 .param B7=2.40
29 .param B8=3.29
30 .param B9=2.0
31 .param B10=2.42
32 .param B11=1.36
33 .param B12=2.40
34 .param B13=2.28
35 .param B14=1.75
36 .param B15=1.07
37 .param B16=1.54
38 .param B17=1.64
39 .param B18=2.5
40
41 .param IB1=195E-6
42 .param IB2=168E-6
43 .param IB3=IB1
44 .param IB4=IB2
45 .param IB5=175E-6
46 .param IB6=195E-6
47 .param IB7=60E-6
48 .param IB8=262E-6
49 .param IB9=53E-6
50 .param IB10=175E-6
51
52 .param L1=Lptl
53 .param L2=2.10e-12
54 .param L3=1p
55 .param L4=4.12e-12
56 .param L5=L1
57 .param L6=L2
58 .param L7=L3
59 .param L8=L4
60 .param L9=1p
61 .param L10=Lptl
62 .param L11=3.11e-12
63 .param L12=1.07E-12
64 .param L13=1.01E-12
65 .param L14=2.45e-12
66 .param L15=1.83e-12
67 .param L16=1.12e-12
68 .param L17=4.32e-12
69 .param L18=1.6p
70 .param L20=1.06E-12
71 .param L21=0.55E-12
72 .param L22=5.40e-12
73 .param L23=1.18e-12
74 .param L24=2.78e-12
75 .param L25=Lptl
76
77 .param RD=1.36
78
79 .param RB1=B0Rs/B1
80 .param RB2=B0Rs/B2
81 .param RB3=B0Rs/B3
82 .param RB4=B0Rs/B4
83 .param RB5=B0Rs/B5
84 .param RB6=B0Rs/B6
85 .param RB7=B0Rs/B7
86 .param RB8=B0Rs/B8
87 .param RB9=B0Rs/B9
88 .param RB10=B0Rs/B10
89 .param RB11=B0Rs/B11
90 .param RB12=B0Rs/B12
91 .param RB13=B0Rs/B13
92 .param RB14=B0Rs/B14
93 .param RB15=B0Rs/B15
94 .param RB16=B0Rs/B16
95 .param RB17=B0Rs/B17
96 .param RB18=B0Rs/B18
97 .param LRB1=(RB1/Rsheet)*Lsheet
98 .param LRB2=(RB2/Rsheet)*Lsheet
99 .param LRB3=(RB3/Rsheet)*Lsheet
100 .param LRB4=(RB4/Rsheet)*Lsheet
101 .param LRB5=(RB5/Rsheet)*Lsheet
102 .param LRB6=(RB6/Rsheet)*Lsheet
103 .param LRB7=(RB7/Rsheet)*Lsheet
104 .param LRB8=(RB8/Rsheet)*Lsheet
105 .param LRB9=(RB9/Rsheet)*Lsheet
106 .param LRB10=(RB10/Rsheet)*Lsheet
107 .param LRB11=(RB11/Rsheet)*Lsheet
108 .param LRB12=(RB12/Rsheet)*Lsheet
109 .param LRB13=(RB13/Rsheet)*Lsheet
110 .param LRB14=(RB14/Rsheet)*Lsheet
111 .param LRB15=(RB15/Rsheet)*Lsheet
112 .param LRB16=(RB16/Rsheet)*Lsheet
113 .param LRB17=(RB17/Rsheet)*Lsheet
114 .param LRB18=(RB18/Rsheet)*Lsheet
115
116 B1 1 2 jjmit area=B1
117 B2 3 4 jjmit area=B2
118 B3 5 6 jjmit area=B3
119 B4 8 9 jjmit area=B4
120 B5 10 11 jjmit area=B5
121 B6 12 13 jjmit area=B6
122 B7 14 15 jjmit area=B7
123 B8 15 16 jjmit area=B8
124 B9 17 18 jjmit area=B9
125 B10 19 20 jjmit area=B10
126 B11 22 15 jjmit area=B11
127 B12 23 24 jjmit area=B12
128 B13 26 27 jjmit area=B13
129 B14 29 34 jjmit area=B14
130 B15 34 31 jjmit area=B15

```

```

131 | B16 28 33 jjmit area=B16
132 | B17 35 36 jjmit area=B17
133 | B18 37 38 jjmit area=B18
134 |
135 | LP1 2 0 LP
136 | LP2 4 0 LP
137 | LP4 9 0 LP
138 | LP5 11 0 LP
139 | LP8 16 0 LP
140 | LP9 18 0 LP
141 | LP10 20 0 LP
142 | LP12 24 0 LP
143 | LP13 27 0 LP
144 | LP17 36 0 LP
145 | LP18 38 0 LP
146 |
147 | IB1 0 1 pwl(0 0 5p IB1)
148 | IB2 0 6 pwl(0 0 5p IB2)
149 | IB3 0 8 pwl(0 0 5p IB3)
150 | IB4 0 13 pwl(0 0 5p IB4)
151 | IB5 0 17 pwl(0 0 5p IB5)
152 | IB6 0 19 pwl(0 0 5p IB6)
153 | IB7 0 25 pwl(0 0 5p IB7)
154 | IB8 0 34 pwl(0 0 5p IB8)
155 | IB9 0 35 pwl(0 0 5p IB9)
156 | IB10 0 37 pwl(0 0 5p IB10)
157 |
158 | L1 a 1 L1
159 | L2 1 3 L2
160 | L3 3 5 L3
161 | L4 6 7 L4
162 | L5 b 8 L5
163 | L6 8 10 L6
164 | L7 10 12 L7
165 | L8 13 7 L8
166 | L9 7 14 L9
167 | L10 clk 17 L10
168 | L11 17 19 L11
169 | L12 19 21 L12
170 | L13 21 22 L13
171 | L14 21 23 L14
172 | L15 23 25 L15
173 | L16 25 26 L16
174 | L17 26 28 L17
175 | L18 15 29 L18
176 | L20 31 32 L20
177 | L21 33 32 L21
178 | L22 34 28 L22
179 | L23 32 35 L23
180 | L24 35 37 L24
181 | L25 37 39 L25
182 | RD 39 q RD
183 |
184 | RB1 1 101 RB1
185 | LRB1 101 0 LRB1
186 | RB2 3 103 RB2
187 | LRB2 103 0 LRB2
188 | RB3 5 105 RB3
189 | LRB3 105 6 LRB3
190 | RB4 8 108 RB4
191 | LRB4 108 0 LRB4
192 | RB5 10 110 RB5
193 | LRB5 110 0 LRB5
194 | RB6 12 112 RB6
195 | LRB6 112 13 LRB6
196 | RB7 14 114 RB7
197 | LRB7 114 15 LRB7
198 | RB8 15 115 RB8
199 | LRB8 115 0 LRB8
200 | RB9 17 117 RB9
201 | LRB9 117 0 LRB9
202 | RB10 19 119 RB10
203 | LRB10 119 0 LRB10
204 | RB11 22 122 RB11
205 | LRB11 122 15 LRB11
206 | RB12 23 123 RB12
207 | LRB12 123 0 LRB12
208 | RB13 26 126 RB13
209 | LRB13 126 0 LRB13
210 | RB14 29 129 RB14
211 | LRB14 129 34 LRB14
212 | RB15 34 130 RB15
213 | LRB15 130 31 LRB15
214 | RB16 28 128 RB16
215 | LRB16 128 33 LRB16
216 | RB17 35 135 RB17
217 | LRB17 135 0 LRB17
218 | RB18 37 137 RB18
219 | LRB18 137 0 LRB18
220 | .ends

```

Listing 5.23: RSFQ XNORT JoSIM netlist.**Table 5.19:** RSFQ XNORT pin list.

Pin	Description
a	Data input
b	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ XNORT using JoSIM is shown in Fig. 5.44. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b**,
- (d) the phase over the input JJ of pin **b**,
- (e) the current through the input inductor connected to pin **clk**,
- (f) the phase over the input JJ of pin **clk**,
- (g) the phase over the output JJ of pin **q**, and
- (h) the phase over the input JJ of the load circuit connected via a PTL to pin **q**.

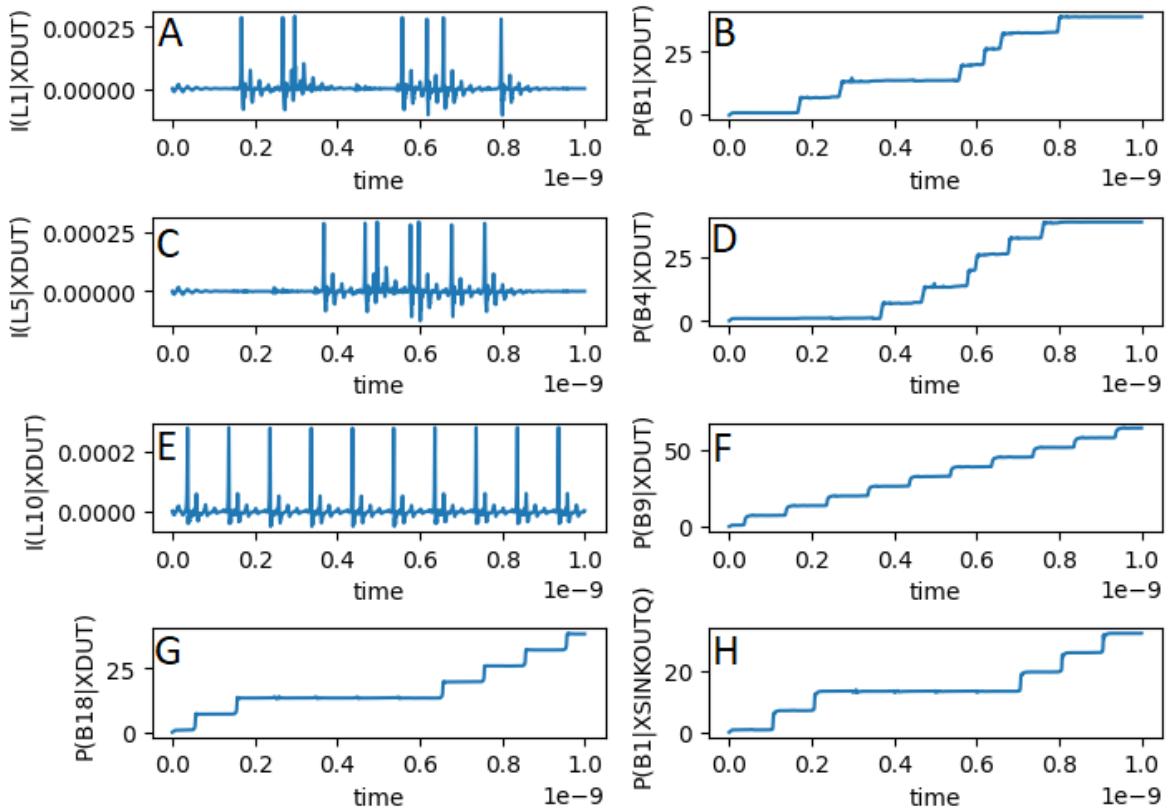


Figure 5.44: RSFQ XNORT analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 4 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 //
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 //
15 'timescale 1ps/100fs
16 module LSmitll_XNORT_v2p1_optimized (a, b, clk, q);
17
18 input
19   a, b, clk;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state0_clk_q = 19.0,
29   ct_state0_clk_a = 6.0,
30   ct_state0_clk_b = 6.0,
31   ct_state0_clk_clk = 30.0,
32   ct_state1_b_a = 11.3,
33   ct_state1_b_b = 7.5,
34   ct_state1_b_clk = 18.1,
35   ct_state1_clk_a = 12.5,
36   ct_state1_clk_b = 7.6,
37   ct_state2_a_a = 7.5,
38   ct_state2_a_b = 11.3,
39   ct_state2_a_clk = 18.1,
40   ct_state2_clk_a = 7.6,
41   ct_state2_clk_b = 12.5;
42
43 reg
44   errorsignal_a,
45   errorsignal_b,
46   errorsignal_clk;
47
48 integer
49   outfile,
50   cell_state; // internal state of the cell
51
52 initial
53 begin
54   errorsignal_a = 0;
55   errorsignal_b = 0;
56   errorsignal_clk = 0;
57   cell_state = 0; // Startup state
58   q = 0; // All outputs start at 0
59 end
60
61 always @(posedge a or negedge a) // execute at positive and negative edges of input
62 begin
63   if ($time>4) // arbitrary steady-state time)
64   begin
65     if (errorsignal_a == 1'b1) // A critical timing is active for this input
66     begin
67       outfile = $fopen("errors.txt", "a");

```

```

68         $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
69             ↪ ", $stime);
70         $fclose(outfile);
71         q <= 1'bX; // Set all outputs to unknown
72     end
73     if (errorsignal_a == 0)
74     begin
75         case (cell_state)
76             0: begin
77                 cell_state = 1; // Blocking statement -- immediately
78             end
79             1: begin
80                 end
81             2: begin
82                 cell_state = 0; // Blocking statement -- immediately
83                 errorsignal_a = 1; // Critical timing on this input; assign
84                     ↪ immediately
85                 errorsignal_a <= #(ct_state2_a_a) 0; // Clear error signal
86                     ↪ after critical timing expires
87                 errorsignal_b = 1; // Critical timing on this input; assign
88                     ↪ immediately
89                 errorsignal_b <= #(ct_state2_a_b) 0; // Clear error signal
90                     ↪ after critical timing expires
91                 errorsignal_clk = 1; // Critical timing on this input; assign
92                     ↪ immediately
93                 errorsignal_clk <= #(ct_state2_a_clk) 0; // Clear error signal
94                     ↪ after critical timing expires
95             end
96         endcase
97     end
98
99 always @(posedge b or negedge b) // execute at positive and negative edges of input
100 begin
101     if ($time>4) // arbitrary steady-state time)
102     begin
103         if (errorsignal_b == 1'b1) // A critical timing is active for this input
104         begin
105             outfile = $fopen("errors.txt", "a");
106             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
107                 ↪ ", $stime);
108             $fclose(outfile);
109             q <= 1'bX; // Set all outputs to unknown
110         end
111         if (errorsignal_b == 0)
112         begin
113             case (cell_state)
114                 0: begin
115                     cell_state = 2; // Blocking statement -- immediately
116                 end
117                 1: begin
118                     cell_state = 0; // Blocking statement -- immediately
119                     errorsignal_a = 1; // Critical timing on this input; assign
120                         ↪ immediately
121                     errorsignal_a <= #(ct_state1_b_a) 0; // Clear error signal
122                         ↪ after critical timing expires
123                     errorsignal_b = 1; // Critical timing on this input; assign
124                         ↪ immediately
125                     errorsignal_b <= #(ct_state1_b_b) 0; // Clear error signal
126                         ↪ after critical timing expires
127                     errorsignal_clk = 1; // Critical timing on this input; assign
128                         ↪ immediately
129                     errorsignal_clk <= #(ct_state1_b_clk) 0; // Clear error signal
130                         ↪ after critical timing expires
131                 end
132                 2: begin
133                     end
134             endcase
135         end
136     end

```

```

124         end
125     end
126
127     always @(posedge clk or negedge clk) // execute at positive and negative edges of input
128     begin
129         if ($time>4) // arbitrary steady-state time)
130             begin
131                 if (errorsignal_clk == 1'b1) // A critical timing is active for this input
132                     begin
133                         outfile = $fopen("errors.txt", "a");
134                         $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
135                                     ↪ ", $stime);
136                         $fclose(outfile);
137                         q <= 1'bX; // Set all outputs to unknown
138                     end
139                 if (errorsignal_clk == 0)
140                     begin
141                         case (cell_state)
142                             0: begin
143                                 q <= #(delay_state0_clk_q) !q;
144                                 errorsignal_a = 1; // Critical timing on this input; assign
145                                     ↪ immediately
146                                 errorsignal_a <= #(ct_state0_clk_a) 0; // Clear error signal
147                                     ↪ after critical timing expires
148                                 errorsignal_b = 1; // Critical timing on this input; assign
149                                     ↪ immediately
150                                 errorsignal_b <= #(ct_state0_clk_b) 0; // Clear error signal
151                                     ↪ after critical timing expires
152                                 errorsignal_clk = 1; // Critical timing on this input; assign
153                                     ↪ immediately
154                                 errorsignal_clk <= #(ct_state0_clk_clk) 0; // Clear error
155                                     ↪ signal after critical timing expires
156                             end
157                             1: begin
158                                 cell_state = 0; // Blocking statement -- immediately
159                                 errorsignal_a = 1; // Critical timing on this input; assign
160                                     ↪ immediately
161                                 errorsignal_a <= #(ct_state1_clk_a) 0; // Clear error signal
162                                     ↪ after critical timing expires
163                                 errorsignal_b = 1; // Critical timing on this input; assign
164                                     ↪ immediately
165                                 errorsignal_b <= #(ct_state1_clk_b) 0; // Clear error signal
166                                     ↪ after critical timing expires
167                             end
168                         endcase
169                     end
170                 end
171             end
172         end
173     endmodule

```

Listing 5.24: RSFQ XNORT verilog model.

The digital simulation results for the RSFQ XNORT is shown in Fig. 5.45 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 5.46.



Figure 5.45: RSFQ XNORT digital simulation results.

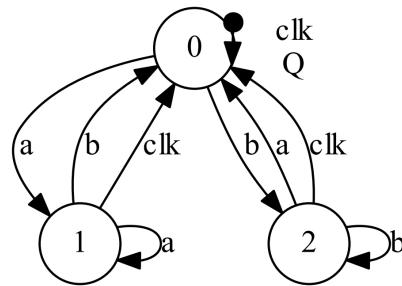


Figure 5.46: RSFQ XNORT Mealy finite state machine diagram.

Power Consumption

Table 5.20: RSFQ XNORT power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	4280	7.98
2	4280	16.0
5	4280	39.9
10	4280	79.8
20	4280	160
50	4280	399

5.3 Buffers

5.3.1 DFFT

The RSFQ DFFT, D flip-flop, is a multi-state device used to transmit an input set pulse synchronised with a reset (typically clock) signal. The DFFT is designed with integrated PTL transmitters and receivers and is intended to connect directly to PTLs.

Schematic

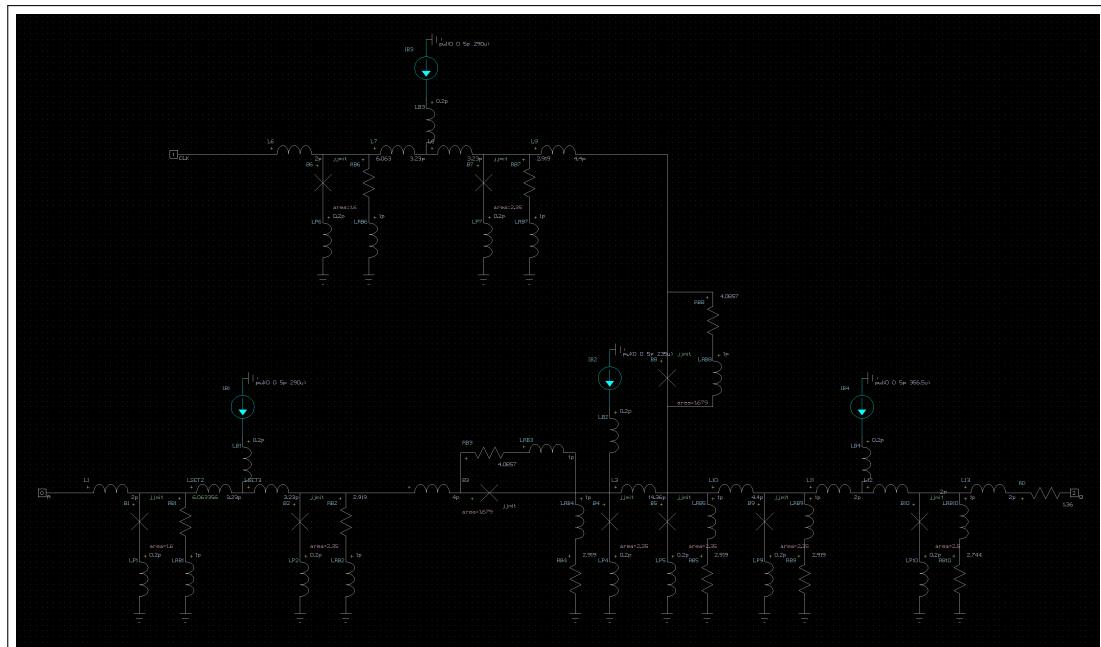


Figure 5.47: Schematic of RSFQ DFFT.

Layout

The physical layout for the RSFQ DFFT is shown in Fig. 5.48. The layout height is $70 \mu m$ and the width is $80 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

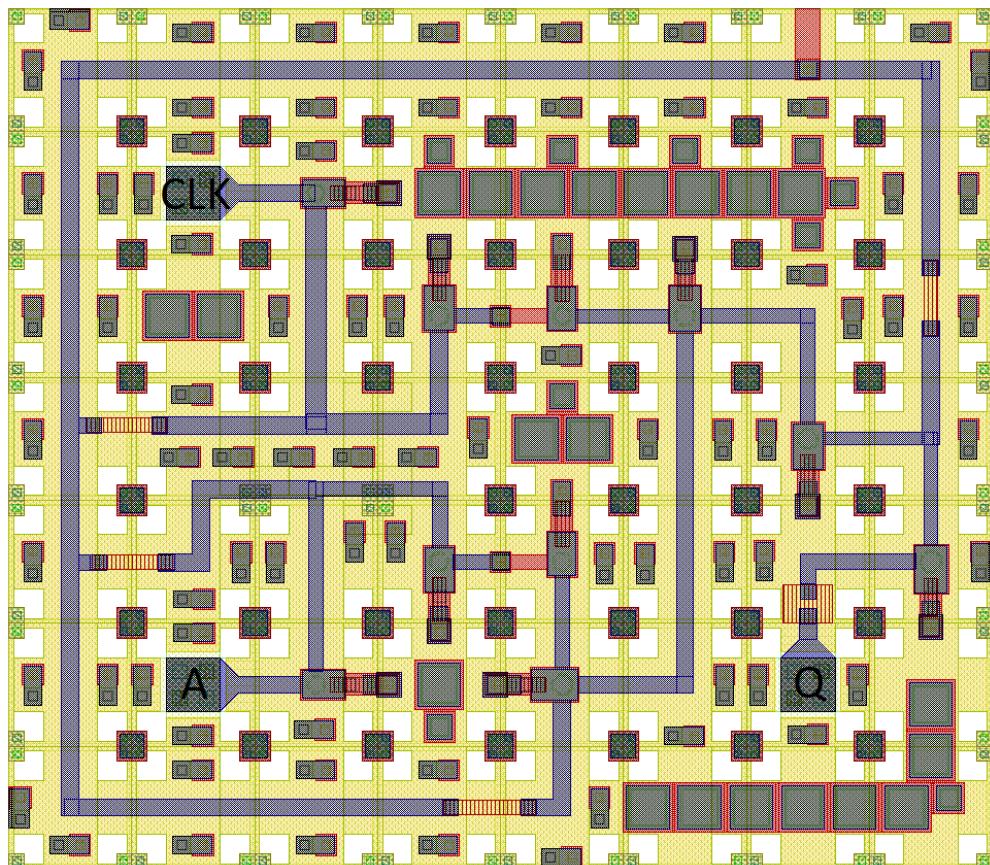


Figure 5.48: RSFQ DFFT Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param RB10=B0Rs/B10
2   ↪ by InductEx v.6.0.4 on 27-4-21.          65  .param LRB1=(RB1/Rsheet)*Lsheet+LP
3  * Author: L. Schindler                      66  .param LRB2=(RB2/Rsheet)*Lsheet+LP
4  * Version: 2.1                               67  .param LRB3=(RB3/Rsheet)*Lsheet+LP
5  * Last modification date: 8 June 2021       68  .param LRB4=(RB4/Rsheet)*Lsheet+LP
6  * Last modification by: L. Schindler        69  .param LRB5=(RB5/Rsheet)*Lsheet+LP
7  *$Ports a clk q                           70  .param LRB6=(RB6/Rsheet)*Lsheet+LP
8  .subckt LSMITLL_DFFT a clk q             71  .param LRB7=(RB7/Rsheet)*Lsheet+LP
9  .model jjmit jj(rtype=1, vg=2.8mV, cap    72  .param LRB8=(RB8/Rsheet)*Lsheet+LP
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA     73  .param LRB9=(RB9/Rsheet)*Lsheet+LP
   ↪ )                                         74  .param LRB10=(RB10/Rsheet)*Lsheet+LP
10 .param Phi0=2.067833848E-15                75  .param LP1=LP
11 .param B0=1                                 76  .param LP2=LP
12 .param Ic0=0.0001                          77  .param LP4=LP
13 .param IcRs=100u*6.859904418               78  .param LP5=LP
14 .param B0Rs=IcRs/Ic0*B0                   79  .param LP6=LP
15 .param Rsheet=2                            80  .param LP7=LP
16 .param Lsheet=1.13e-12                     81  .param LP9=LP
17 .param LP=0.2p                            82  .param LP10=LP
18 .param IC=2.5                             83  .param LB1=LB
19 .param ICreceive=2.0                      84  .param LB2=LB
20 .param ICtrans=2.5                        85  .param LB3=LB
21 .param Lptl=2p                           86  .param LB4=LB
22 .param LB=2p                            87  IB1 0 5 pwl(0 0 5p IB1)
23 .param BiasCoef=0.7                      88  IB2 0 11 pwl(0 0 5p IB2)
24 .param RD=1.36                           89  IB3 0 18 pwl(0 0 5p IB3)
25 .param B1=1.62                           90  IB4 0 25 pwl(0 0 5p IB4)
26 .param B2=1.89                           91  B1 2 3 jjmit area=B1
27 .param B3=1.72                           92  B2 6 7 jjmit area=B2
28 .param B4=2.32                           93  B3 8 9 jjmit area=B3
29 .param B5=2.12                           94  B4 9 10 jjmit area=B4
30 .param B6=1.62                           95  B5 12 13 jjmit area=B5
31 .param B7=1.98                           96  B6 15 16 jjmit area=B6
32 .param B8=1.71                           97  B7 19 20 jjmit area=B7
33 .param B9=2.12                           98  B8 21 12 jjmit area=B8
34 .param B10=2.5                          99  B9 22 23 jjmit area=B9
35 .param B10=2.5                         100 B10 26 27 jjmit area=B10
36 .param IB1=276u                          101 L1 a 2 1.587E-12
37 .param IB2=235u                          102 L2 2 4 3.253E-12
38 .param IB3=284u                          103 L3 4 6 3.304E-12
39 .param IB4=312u                          104 L4 6 8 3.979E-12
40 .param L1=Lptl                           105 L5 9 12 7.521E-12
41 .param L2=(Phi0/(2*B1*Ic0))/2           106 L6 clk 15 1.6E-12
42 .param L3=(Phi0/(2*B1*Ic0))/2           107 L7 15 17 3.042E-12
43 .param L4=(Phi0/(2*B2*Ic0))/2           108 L8 17 19 3.045E-12
44 .param L5=Phi0/(B4*Ic0)                 109 L9 19 21 4.21E-12
45 .param L6=Lptl                           110 L10 12 22 4.022E-12
46 .param L7=(Phi0/(2*B6*Ic0))/2           111 L11 22 24 2.164E-12
47 .param L8=(Phi0/(2*B6*Ic0))/2           112 L12 24 26 2.183E-12
48 .param L9=Phi0/(2*B7*Ic0)               113 L13 26 28 2.536E-12
49 .param L10=Phi0/(2*B5*Ic0)              114 LP1 3 0 5.021E-13
50 .param L11=(Phi0/(2*B9*Ic0))/2          115 LP2 7 0 5.102E-13
51 .param L12=(Phi0/(2*B9*Ic0))/2          116 LP4 10 0 5.275E-13
52 .param L13=Lptl                           117 LP5 13 0 5.37E-13
53 .param RB1=B0Rs/B1                      118 LP6 16 0 5.005E-13
54 .param RB2=B0Rs/B2                      119 LP7 20 0 5.161E-13
55 .param RB3=B0Rs/B3                      120 LP9 23 0 5.212E-13
56 .param RB4=B0Rs/B4                      121 LP10 27 0 5.039E-13
57 .param RB5=B0Rs/B5                      122 LB1 4 5 3.559E-12
58 .param RB6=B0Rs/B6                      123 LB2 9 11 2.45E-12
59 .param RB7=B0Rs/B7                      124 LB3 17 18 2.72E-12
60 .param RB8=B0Rs/B8                      125 LB4 24 25 1.988E-12
61 .param RB9=B0Rs/B9                      126 RB1 2 102 RB1
62 .param RB10=B0Rs/B10                     127 RB2 6 106 RB2
63 .param RB11=B0Rs/B11                     128 RB3 8 108 RB3
64 .param RB12=B0Rs/B12                     129 RB4 9 109 RB4

```

```

130 | RB5 12 112 RB5
131 | RB6 15 115 RB6
132 | RB7 19 119 RB7
133 | RB8 21 121 RB8
134 | RB9 22 122 RB9
135 | RB10 26 126 RB10
136 | LRB1 102 0 LRB1
137 | LRB2 106 0 LRB2
138 | LRB3 108 9 LRB3
139 | LRB4 109 0 LRB4
140 | LRB5 112 0 LRB5
141 | LRB6 115 0 LRB6
142 | LRB7 119 0 LRB7
143 | LRB8 121 12 LRB8
144 | LRB9 122 0 LRB9
145 | LRB10 126 0 LRB10
146 | RD 28 q RD
147 | .ends

```

Listing 5.25: RSFQ DFFT JoSIM netlist.**Table 5.21:** RSFQ DFFT pin list.

Pin	Description
a	Data input
clk	Clock input
q	Data output

The simulation results for the RSFQ DFFT using JoSIM is shown in Fig. 5.49. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **clk**,
- (d) the phase over the input JJ of pin **clk**,
- (e) the phase over the output JJ of pin **q**, and
- (f) the phase over the input JJ of the load circuit connected via a PTL to pin **q**.

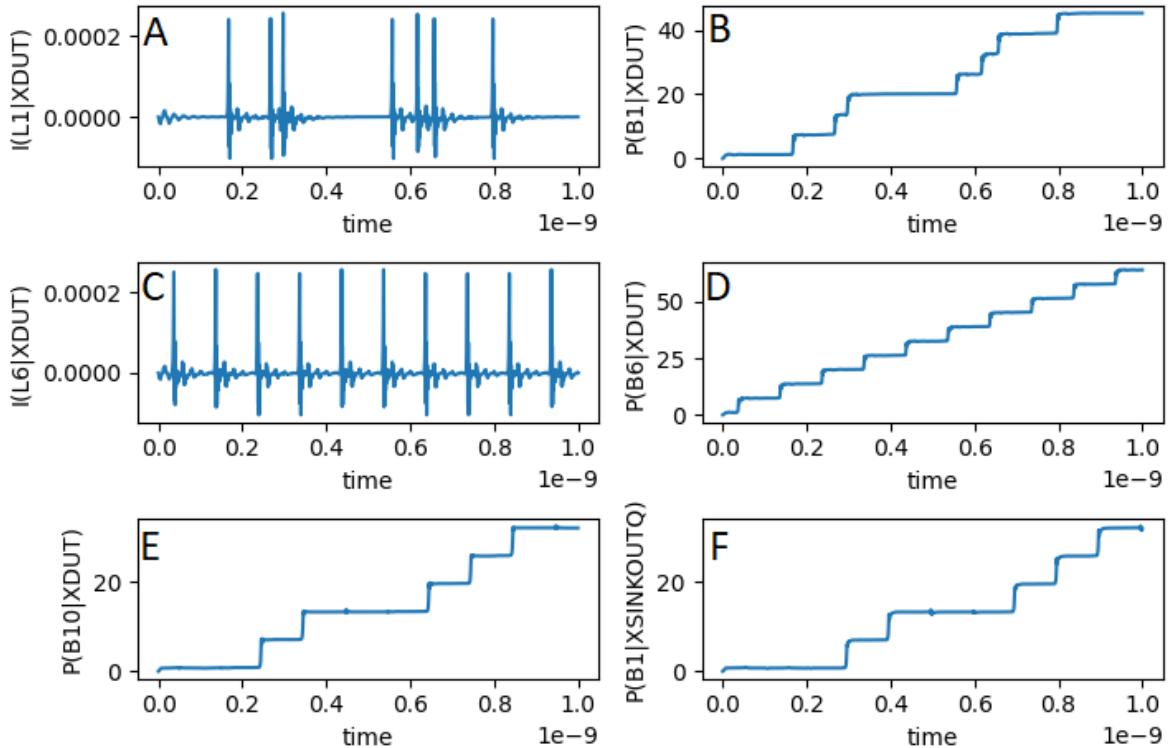


Figure 5.49: RSFQ DFFT analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 8 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 // -----
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 // -----
15 `timescale 1ps/100fs
16 module LSmitll_DFFT_v2p1_optimized (a, clk, q);
17
18 input
19   a, clk;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state1_clk_q = 8.5,
29   ct_state0_clk_a = 1.8,
30   ct_state1_a_clk = 1.8;
31
32 reg
33   errorsignal_a,
34   errorsignal_clk;
35
36 integer
37   outfile,
38   cell_state; // internal state of the cell
39
40 initial
41 begin
42   errorsignal_a = 0;
43   errorsignal_clk = 0;
44   cell_state = 0; // Startup state
45   q = 0; // All outputs start at 0
46 end
47
48 always @ (posedge a or negedge a) // execute at positive and negative edges of input
49 begin
50   if ($time>4) // arbitrary steady-state time)
51     begin
52       if (errorsignal_a == 1'b1) // A critical timing is active for this input
53         begin
54           outfile = $fopen("errors.txt", "a");
55           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
56           ↪ ", $stime);
57           $fclose(outfile);
58           q <= 1'bX; // Set all outputs to unknown
59         end
60       if (errorsignal_a == 0)
61         begin
62           case (cell_state)
63             0: begin
64               cell_state = 1; // Blocking statement -- immediately
65             end
66             1: begin
67               errorsignal_clk = 1; // Critical timing on this input; assign

```

```

67           ↪ immediately
68           errorsignal_clk <= #(ct_state1_a_clk) 0; // Clear error signal
69           ↪ after critical timing expires
70       end
71   endcase
72 end
73
74 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
75 begin
76     if ($time>4) // arbitrary steady-state time)
77     begin
78         if (errorsignal_clk == 1'b1) // A critical timing is active for this input
79         begin
80             outfile = $fopen("errors.txt", "a");
81             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
82             ↪ ", $stime);
83             $fclose(outfile);
84             q <= 1'bX; // Set all outputs to unknown
85         end
86         if (errorsignal_clk == 0)
87         begin
88             case (cell_state)
89             0: begin
90                 errorsignal_a = 1; // Critical timing on this input; assign
91                 ↪ immediately
92                 errorsignal_a <= #(ct_state0_clk_a) 0; // Clear error signal
93                 ↪ after critical timing expires
94             end
95             1: begin
96                 q <= #(delay_state1_clk_q) !q;
97                 cell_state = 0; // Blocking statement -- immediately
98             end
99         endcase
100    end
101 endmodule

```

Listing 5.26: RSFQ DFFT verilog model.

The digital simulation results for the RSFQ DFFT is shown in Fig. 5.50 and the Mealy finite state machine diagram, extracted using *TimEx*, is shown in Fig. 5.51.

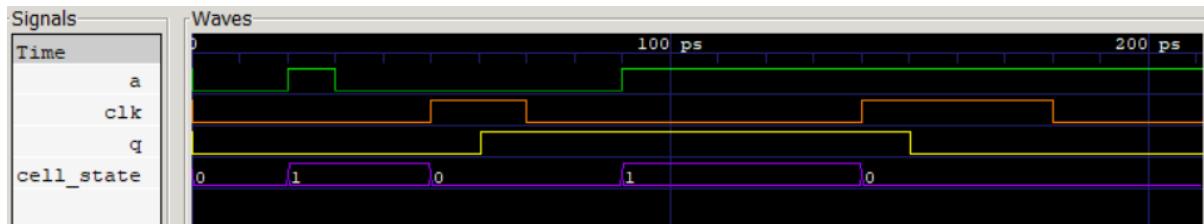


Figure 5.50: RSFQ DFFT digital simulation results.

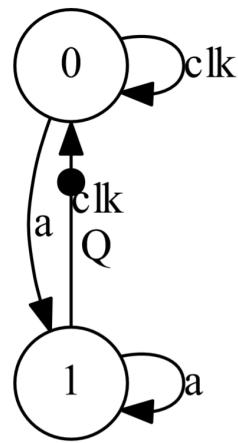


Figure 5.51: RSFQ DFFT Mealy finite state machine diagram.

Power Consumption

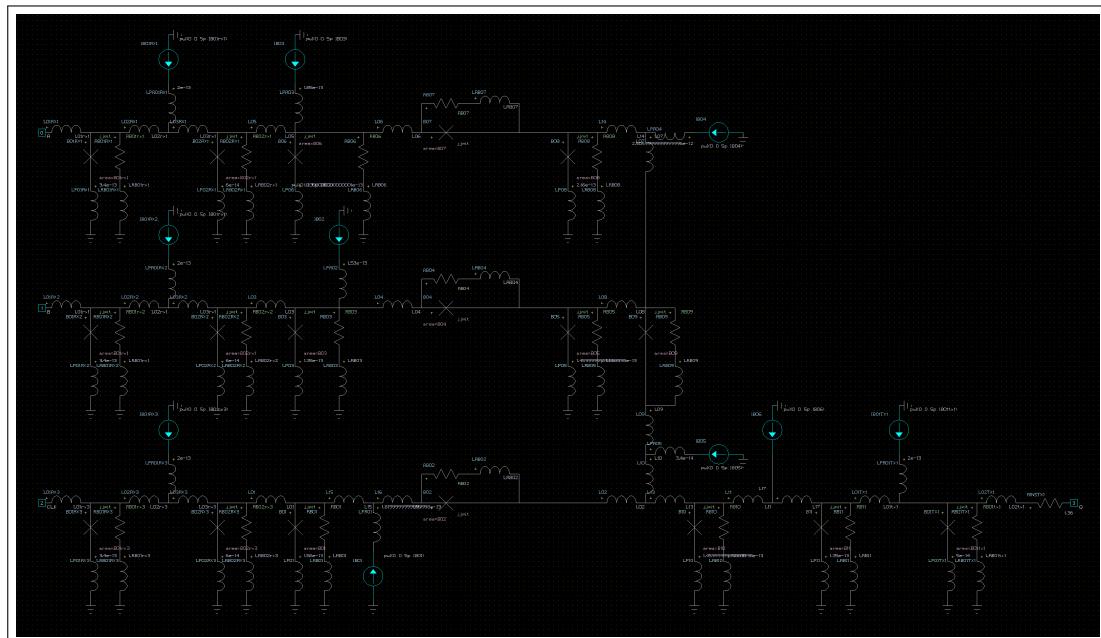
Table 5.22: RSFQ DFFT power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	2878	4.05
2	2878	8.11
5	2878	20.3
10	2878	40.5
20	2878	81.1
50	2878	203

5.3.2 NDROT

The NDROT, non-destructive readout, cell is a memory device controlled by a set, reset and clock input signal. When an input set signal is received, the NDROT will generate an output pulse after each clock signal until an input reset signal is received. The NDROT is designed with integrated PTL transmitters and receivers and is intended to connect directly to PTLs.

Schematic



Layout

The physical layout for the RSFQ NDROT is shown in Fig. 5.53. The layout height is $70 \mu m$ and the width is $120 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

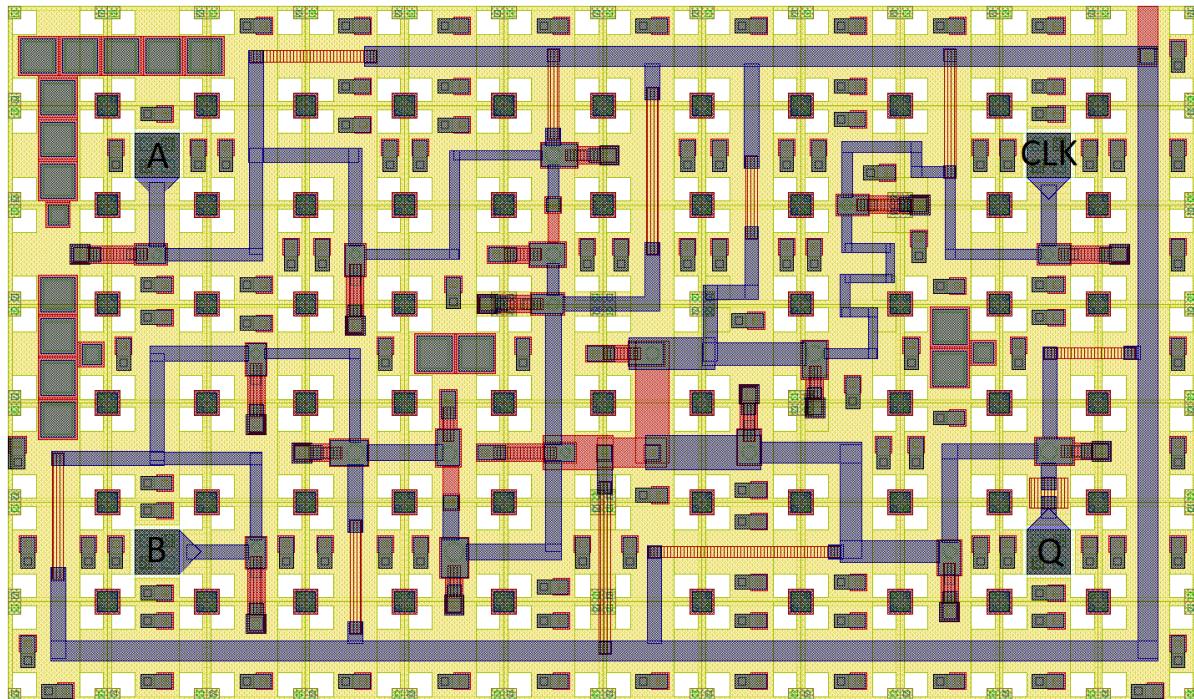


Figure 5.53: RSFQ NDROT Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param L16=3.926p
2   ↪ by InductEx v.6.0.4 on 29-4-21.          65  .param L17=7.5833p
3  * Author: L. Schindler                      66  .param L18=1.2875p
4  * Version: 2.1                               67  .param L19=1.0678p
5  * Last modification date: 29 April 2021     68  .param L21=0.37382p
6  * Last modification by: L. Schindler        69  .param L22=0.52995p
7  *$Ports a b clk q                         70  .param L23=0.95137p
8 .subckt LSmitll_NDROT a b clk q           71  .param L24=2.5089p
9 .model jjmit jj(rtype=1, vg=2.8mV, cap    72  .param L25=1.2791p
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    73  .param L26=3.5427p
   ↪ )                                         74  .param L27=Lptl
10 .param B0=1                                 75
11 .param Ic0=0.0001                           76  .param RB1=B0Rs/B1
12 .param IcRs=100u*6.859904418                77  .param RB2=B0Rs/B2
13 .param B0Rs=IcRs/Ic0*B0                     78  .param RB3=B0Rs/B3
14 .param Rsheet=2                            79  .param RB4=B0Rs/B4
15 .param Lsheet=1.13e-12                      80  .param RB5=B0Rs/B5
16 .param Lptl=2p                             81  .param RB6=B0Rs/B6
17 .param LB=2p                               82  .param RB7=B0Rs/B7
18 .param LP=0.2p                            83  .param RB8=B0Rs/B8
19
20 .param B1=0.86                            84  .param RB9=B0Rs/B9
21 .param B2=1.00                            85  .param RB10=B0Rs/B10
22 .param B3=1.91                            86  .param RB11=B0Rs/B11
23 .param B4=1.78                            87  .param RB12=B0Rs/B12
24 .param B5=1.16                            88  .param RB13=B0Rs/B13
25 .param B6=0.86                            89  .param RB14=B0Rs/B14
26 .param B7=1.00                            90  .param RB15=B0Rs/B15
27 .param B8=2.35                            91  .param RB16=B0Rs/B16
28 .param B9=1.96                            92  .param RB17=B0Rs/B17
29 .param B10=2.84                           93  .param RB18=B0Rs/B18
30 .param B11=0.78                           94
31 .param B12=0.99                           95  .param LRB1=(RB1/Rsheet)*Lsheet
32 .param B13=0.94                           96  .param LRB2=(RB2/Rsheet)*Lsheet
33 .param B14=2.18                           97  .param LRB3=(RB3/Rsheet)*Lsheet
34 .param B15=1.66                           98  .param LRB4=(RB4/Rsheet)*Lsheet
35 .param B16=1.63                           99  .param LRB5=(RB5/Rsheet)*Lsheet
36 .param B17=1.51                           100 .param LRB6=(RB6/Rsheet)*Lsheet
37 .param B18=2.36                           101 .param LRB7=(RB7/Rsheet)*Lsheet
38
39 .param IB1=134u                           102 .param LRB8=(RB8/Rsheet)*Lsheet
40 .param IB2=198u                           103 .param LRB9=(RB9/Rsheet)*Lsheet
41 .param IB3=99u                            104 .param LRB10=(RB10/Rsheet)*Lsheet
42 .param IB4=134u                           105 .param LRB11=(RB11/Rsheet)*Lsheet
43 .param IB5=152u                           106 .param LRB12=(RB12/Rsheet)*Lsheet
44 .param IB6=132u                           107 .param LRB13=(RB13/Rsheet)*Lsheet
45 .param IB7=224u                           108 .param LRB14=(RB14/Rsheet)*Lsheet
46 .param IB8=95u                            109 .param LRB15=(RB15/Rsheet)*Lsheet
47 .param IB9=64u                            110 .param LRB16=(RB16/Rsheet)*Lsheet
48 .param IB10=196u                           111 .param LRB17=(RB17/Rsheet)*Lsheet
49
50 .param L1=Lptl                            112 .param LRB18=(RB18/Rsheet)*Lsheet
51 .param L2=4.0481p                           113
52 .param L3=3.6036p                           114 .param LB1=LB
53 .param L4=7.2183p                           115 .param LB2=LB
54 .param L5=3.0677p                           116 .param LB3=LB
55 .param L7=2.5596p                           117 .param LB4=LB
56 .param L8=Lptl                            118 .param LB5=LB
57 .param L9=4.0481p                           119 .param LB6=LB
58 .param L10=3.6036p                           120 .param LB7=LB
59 .param L11=4.3879p                           121 .param LB8=LB
60 .param L12=3.217p                           122 .param LB9=LB
61 .param L13=3.2439p                           123 .param LB10=LB
62 .param L14=Lptl                           124
63 .param L15=4.3135p                           125 .param LP1=LP
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

```

```

130 .param LP7=LP          197 L2 2 4 4.036E-12
131 .param LP8=LP          198 L3 4 6 3.595E-12
132 .param LP10=LP         199 L4 6 8 7.246E-12
133 .param LP12=LP         200 L5 8 11 3.06E-12
134 .param LP13=LP         201 L7 12 29 2.546E-12
135 .param LP14=LP         202 L8 b 17 1.572E-12
136 .param LP16=LP         203 L9 17 19 4.01E-12
137 .param LP17=LP         204 L10 19 21 3.596E-12
138 .param LP18=LP         205 L11 21 23 4.373E-12
139
140 B1 2 3 jjmit area=B1 206 L12 23 26 3.223E-12
141 B2 6 7 jjmit area=B2 207 L13 27 29 3.265E-12
142 B3 8 9 jjmit area=B3 208 L14 clk 34 1.573E-12
143 B4 11 12 jjmit area=B4 209 L15 34 36 4.349E-12
144 B5 12 13 jjmit area=B5 210 L16 36 38 3.941E-12
145 B6 17 18 jjmit area=B6 211 L17 38 40 7.501E-12
146 B7 21 22 jjmit area=B7 212 L18 40 42 1.301E-12
147 B8 23 24 jjmit area=B8 213 L19 42 44 1.505E-12
148 B9 26 27 jjmit area=B9 214 L21 30 31 5.791E-13
149 B10 27 28 jjmit area=B10 215 L22 31 46 5.223E-13
150 B11 29 30 jjmit area=B11 216 L23 46 47 1.048E-12
151 B12 34 35 jjmit area=B12 217 L24 47 49 2.514E-12
152 B13 38 39 jjmit area=B13 218 L25 49 51 1.269E-12
153 B14 40 41 jjmit area=B14 219 L26 51 53 3.539E-12
154 B15 44 46 jjmit area=B15 220 L27 53 56 6.608E-13
155 B16 47 48 jjmit area=B16 221
156 B17 51 52 jjmit area=B17 222 RD 56 q 1.36
157 B18 53 54 jjmit area=B18 223
158
159 IB1 0 5 pwl(0 0 5p IB1) 224 RB1 2 102 RB1
160 IB2 0 10 pwl(0 0 5p IB2) 225 LRB1 102 0 LRB1
161 IB3 0 15 pwl(0 0 5p IB3) 226 RB2 6 106 RB2
162 IB4 0 20 pwl(0 0 5p IB4) 227 LRB2 106 0 LRB2
163 IB5 0 25 pwl(0 0 5p IB5) 228 RB3 8 108 RB3
164 IB6 0 37 pwl(0 0 5p IB6) 229 LRB3 108 0 LRB3
165 IB7 0 43 pwl(0 0 5p IB7) 230 RB4 11 111 RB4
166 IB8 0 32 pwl(0 0 5p IB8) 231 LRB4 111 12 LRB4
167 IB9 0 50 pwl(0 0 5p IB9) 232 RB5 12 112 RB5
168 IB10 0 55 pwl(0 0 5p IB10) 233 LRB5 112 0 LRB5
169
170 LB1 4 5 2.101E-12 234 RB6 17 117 RB6
171 LB2 8 10 4.183E-13 235 LRB6 117 0 LRB6
172 LB3 12 15 2.907E-12 236 RB7 21 121 RB7
173 LB4 19 20 2.228E-12 237 LRB7 121 0 LRB7
174 LB5 23 25 1.461E-12 238 RB8 23 123 RB8
175 LB6 36 37 2.777E-13 239 LRB8 123 0 LRB8
176 LB7 42 43 3.002E-12 240 RB9 26 126 RB9
177 LB8 31 32 9.476E-13 241 LRB9 126 27 LRB9
178 LB9 49 50 5.108E-13 242 RB10 27 127 RB10
179 LB10 53 55 2.133E-12 243 LRB10 127 0 LRB10
180
181 LP1 3 0 5.398E-13 244 RB11 29 129 RB11
182 LP2 7 0 5.963E-13 245 LRB11 129 30 LRB11
183 LP3 9 0 4.237E-13 246 RB12 34 134 RB12
184 LP5 13 0 6.003E-13 247 LRB12 134 0 LRB12
185 LP6 18 0 5.045E-13 248 RB13 38 138 RB13
186 LP7 22 0 5.781E-13 249 LRB13 138 0 LRB13
187 LP8 24 0 4.931E-13 250 RB14 40 140 RB14
188 LP10 28 0 4.784E-13 251 LRB14 140 0 LRB14
189 LP12 35 0 5.289E-13 252 RB15 44 144 RB15
190 LP13 39 0 5.509E-13 253 LRB15 144 46 LRB15
191 LP14 41 0 4.982E-13 254 RB16 47 147 RB16
192 LP16 48 0 5.004E-13 255 LRB16 147 0 LRB16
193 LP17 52 0 5.204E-13 256 RB17 51 151 RB17
194 LP18 54 0 4.135E-13 257 LRB17 151 0 LRB17
195
196 L1 a 2 1.51E-12 258 RB18 53 153 RB18
                                259 LRB18 153 0 LRB18
                                260 .ends

```

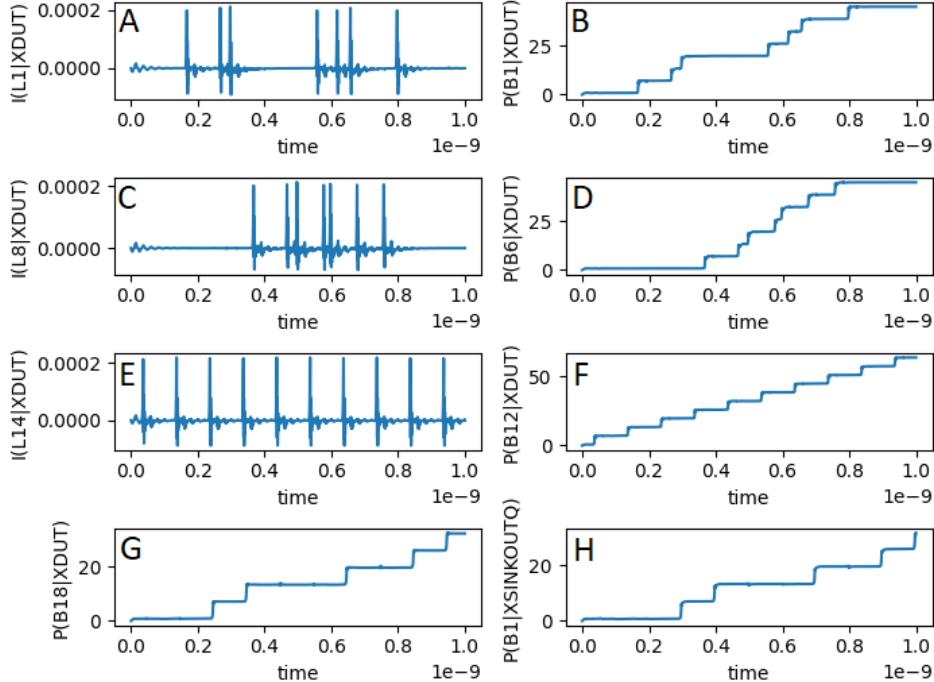
Listing 5.27: RSFQ NDROT JoSIM netlist.

Table 5.23: RSFQ NDROT pin list.

Pin	Description
a	Data input (set signal)
b	Data input (reset signal)
clk	Clock input
q	Data output

The simulation results for the RSFQ NDROT using JoSIM is shown in Fig. 5.54. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a** (set signal),
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the input inductor connected to pin **b** (reset signal),
- (d) the phase over the input JJ of pin **b**,
- (e) the current through the input inductor connected to pin **clk**,
- (f) the phase over the input JJ of pin **clk**,
- (g) the phase over the output JJ of pin **q**, and
- (h) the phase over the input JJ of the load circuit connected via a PTL to pin **q**.

**Figure 5.54:** RSFQ NDROT analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 3 June 2021
5 // Last modification by: L. Schindler
6 // -----
7 //
8 // Automatically extracted verilog file, created with TimEx v2.05
9 // Timing description and structural design for IARPA-BAA-14-03 via
10 // U.S. Air Force Research Laboratory contract FA8750-15-C-0203 and
11 // IARPA-BAA-16-03 via U.S. Army Research Office grant W911NF-17-1-0120.
12 // For questions about TimEx, contact CJ Fourie, coenrad@sun.ac.za
13 // (c) 2016-2018 Stellenbosch University
14 //
15 'timescale 1ps/100fs
16 module LSmitll_NDROT_v2p1_optimized (a, b, clk, q);
17
18 input
19   a, b, clk;
20
21 output
22   q;
23
24 reg
25   q;
26
27 real
28   delay_state1_clk_q = 9.0,
29   ct_state0_b_a = 5.0,
30   ct_state1_a_b = 2.2,
31   ct_state1_clk_clk = 10.1;
32
33 reg
34   errorsignal_a,
35   errorsignal_b,
36   errorsignal_clk;
37
38 integer
39   outfile,
40   cell_state; // internal state of the cell
41
42 initial
43 begin
44   errorsignal_a = 0;
45   errorsignal_b = 0;
46   errorsignal_clk = 0;
47   cell_state = 0; // Startup state
48   q = 0; // All outputs start at 0
49 end
50
51 always @ (posedge a or negedge a) // execute at positive and negative edges of input
52 begin
53   if ($time > 4) // arbitrary steady-state time)
54     begin
55       if (errorsignal_a == 1'b1) // A critical timing is active for this input
56         begin
57           outfile = $fopen("errors.txt", "a");
58           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
59             ↪ ", $stime);
60           $fclose(outfile);
61           q <= 1'bX; // Set all outputs to unknown
62         end
63       if (errorsignal_a == 0)
64         begin
65           case (cell_state)
66             0: begin
67               cell_state = 1; // Blocking statement -- immediately

```

```

67          end
68      1: begin
69          errorsignal_b = 1; // Critical timing on this input; assign
69          ↪ immediately
70          errorsignal_b <= #(ct_state1_a_b) 0; // Clear error signal
70          ↪ after critical timing expires
71      end
72  endcase
73  end
74 end
75
76 always @(posedge b or negedge b) // execute at positive and negative edges of input
77 begin
78     if ($time>4) // arbitrary steady-state time)
79     begin
80         if (errorsignal_b == 1'b1) // A critical timing is active for this input
81         begin
82             outfile = $fopen("errors.txt", "a");
83             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n
83             ↪ ", $stime);
84             $fclose(outfile);
85             q <= 1'bX; // Set all outputs to unknown
86         end
87         if (errorsignal_b == 0)
88         begin
89             case (cell_state)
90                 0: begin
91                     errorsignal_a = 1; // Critical timing on this input; assign
91                     ↪ immediately
92                     errorsignal_a <= #(ct_state0_b_a) 0; // Clear error signal
92                     ↪ after critical timing expires
93                 end
94                 1: begin
95                     cell_state = 0; // Blocking statement -- immediately
95                     end
96             endcase
97         end
98     end
99 end
100
101 always @(posedge clk or negedge clk) // execute at positive and negative edges of input
102 begin
103     if ($time>4) // arbitrary steady-state time)
104     begin
105         if (errorsignal_clk == 1'b1) // A critical timing is active for this input
106         begin
107             outfile = $fopen("errors.txt", "a");
108             $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n
108             ↪ ", $stime);
109             $fclose(outfile);
110             q <= 1'bX; // Set all outputs to unknown
111         end
112         if (errorsignal_clk == 0)
113         begin
114             case (cell_state)
115                 0: begin
116                     end
117                 1: begin
118                     q <= #(delay_state1_clk_q) !q;
119                     errorsignal_clk = 1; // Critical timing on this input; assign
119                     ↪ immediately
120                     errorsignal_clk <= #(ct_state1_clk_clk) 0; // Clear error
120                     ↪ signal after critical timing expires
121                 end
122             endcase
123         end
124     end
125 endmodule

```

Listing 5.28: RSFQ NDROT verilog model.

The digital simulation results for the RSFQ NDROT is shown in Fig. 5.55 and the Mealy finite state diagram, extracted using *TimEx*, is shown in Fig. 5.56.



Figure 5.55: RSFQ NDROT digital simulation results.

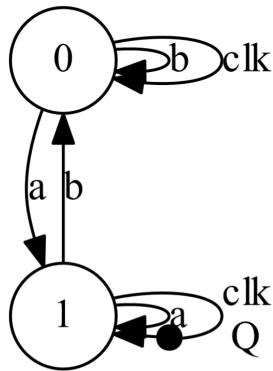


Figure 5.56: RSFQ NDROT Mealy finite state machine diagram.

Power Consumption

Table 5.24: RSFQ NDROT power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	3713	5.74
2	3713	11.5
5	3713	28.7
10	3713	57.4
20	3713	115
50	3713	287

5.3.3 BUFFT

The RSFQ BUFFT cell is a buffer cell intended for clock balancing. It is designed to have the same a-to-q delay as the SPLITT cell. The BUFFT is designed with integrated PTL drivers and receivers and is intended to be connected directly to PTLs.

Schematic

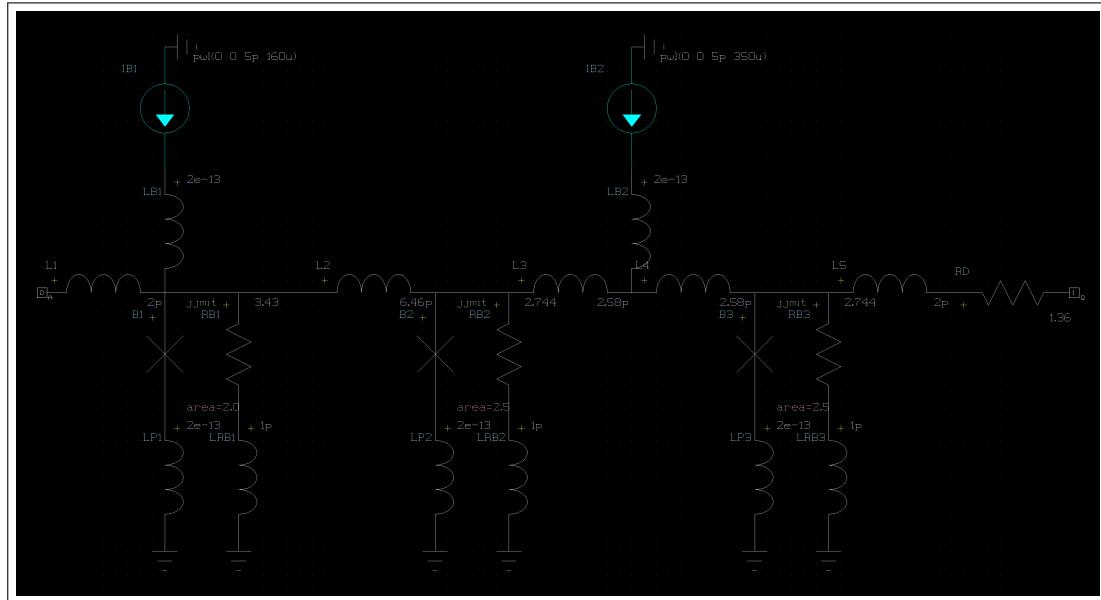


Figure 5.57: Schematic of RSFQ BUFFT.

Layout

The physical layout for the RSFQ BUFFT is shown in Fig. 5.58. The layout height is $70 \mu m$ and the width is $40 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

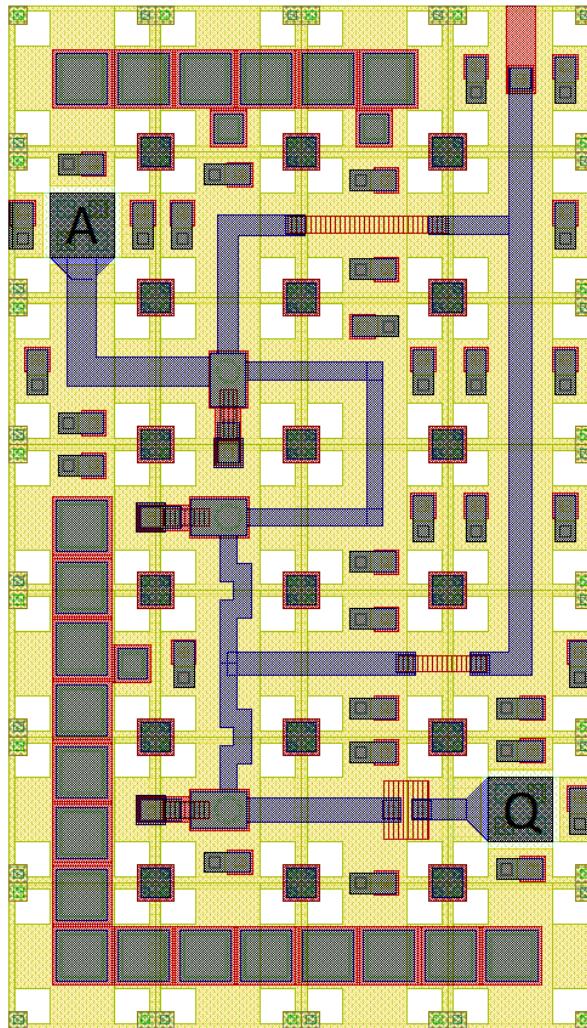


Figure 5.58: RSFQ BUFFT Layout.

Analog model

```

1  * Back-annotated simulation file written      33 | .param LP1=LP
2   ↪ by InductEx v.6.0.4 on 30-4-21.          34 | .param LP2=LP
3  * Author: L. Schindler                      35 | .param LP3=LP
4  * Version: 2.1                               36 | .param RB1=B0Rs/B1
5  * Last modification date: 11 June 2021       37 | .param RB2=B0Rs/B2
6  * Last modification by: L. Schindler        38 | .param RB3=B0Rs/B3
7  *$Ports a q                                39 | .param LRB1=(RB1/Rsheet)*Lsheet
8 .subckt LSmitll_bufft a q                   40 | .param LRB2=(RB2/Rsheet)*Lsheet
9 .model jjmit jj(rtype=1, vg=2.8mV, cap     41 | .param LRB3=(RB3/Rsheet)*Lsheet
10    ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    42
11    ↪ )
12 .param B0=1.0                               43 | B1 2 3 jjmit area=B1
13 .param Ic0=0.0001                           44 | B2 6 7 jjmit area=B2
14 .param IcRs=100u*6.859904418                45 | B3 11 12 jjmit area=B3
15 .param B0Rs=IcRs/Ic0*B0                     46 | IB1 0 5 pwl(0 0 5p IB1)
16 .param Rsheet=2                            47 | IB2 0 10 pwl(0 0 5p IB2)
17 .param Lsheet=1.13e-12                      48 | L1 a 2 2.727E-12
18 .param RD=1.36                             49 | L2 2 6 6.22E-12
19 .param LB=0.2p                            50 | L3 6 9 2.06E-12
20 .param Lpt1=2p                            51 | L4 9 11 2.051E-12
21 .param LP=0.2p                            52 | L5 11 14 2.041E-12
22 .param B1=2.0                               53 | RD 14 q RD
23 .param B2=2.5                               54 | LP1 3 0 5.275E-13
24 .param B3=2.5                               55 | LP2 7 0 5.212E-13
25 .param IB1=168u                            56 | LP3 12 0 5.126E-13
26 .param IB2=340u                            57 | RB1 2 4 RB1
27 .param LB1=LB                             58 | RB2 6 8 RB2
28 .param L1=Lpt1                           59 | RB3 11 13 RB3
29 .param L2=6.2p                            60 | LRB1 4 0 LRB1
30 .param L3=2.07p                           61 | LRB2 8 0 LRB2
31 .param L4=2.07p                           62 | LRB3 13 0 LRB3
32 .param L5=Lpt1                           63 | LB1 2 5 2.883E-12
                                         64 | LB2 9 10 2.393E-12
                                         65 | .ends

```

Listing 5.29: RSFQ BUFFT JoSIM netlist.

Table 5.25: RSFQ BUFFT pin list.

Pin	Description
a	Data input
q	Data output

The simulation results for the RSFQ BUFFT using JoSIM is shown in Fig. 5.59. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the phase over the output JJ of pin **q**, and
- (d) the phase over the input JJ of the load circuit connected via a PTL to pin **q**.

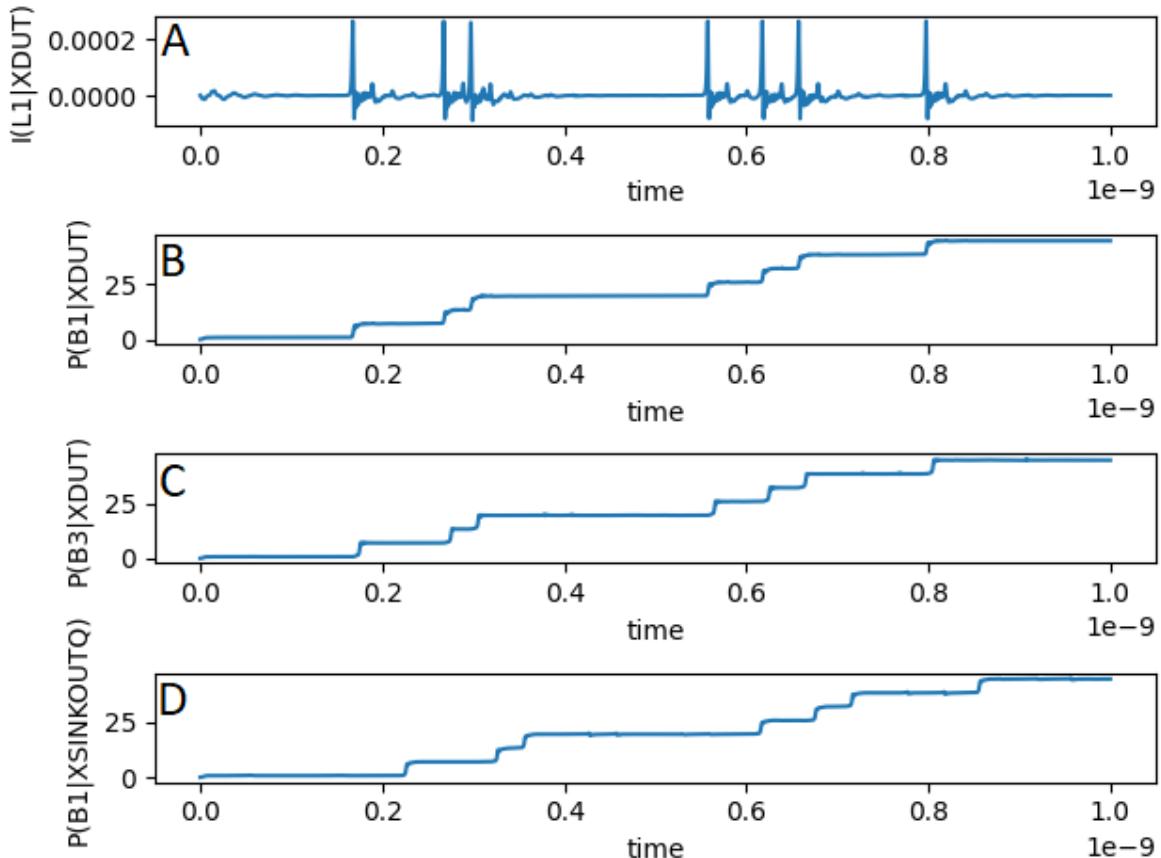


Figure 5.59: RSFQ BUFFT analog simulation results.

Digital model

```

1 // -----
2 // Author: L. Schindler
3 // Version: 2.1
4 // Last modification date: 11 June 2021
5 // Last modification by: L. Schindler
6 //
7 'timescale 1ps/100fs
8 module LSmitll_BUFFT_v2p1_annotated (a, q);
9
10 input
11   a;
12
13 output
14   q;
15
16 reg
17   q;
18
19 real
20   delay_state0_a_q = 7.3,
21   ct_state0_a_a = 7.9;
22
23 reg
24   errorsignal_a;
25
26 integerc
27   outfile,
28   cell_state; // internal state of the cell
29
30 initial
31 begin
32   errorsignal_a = 0;
33   cell_state = 0; // Startup state
34   q = 0; // All outputs start at 0
35 end
36
37 always @(posedge a or negedge a) // execute at positive and negative edges of input
38 begin
39   if ($time>4) // arbitrary steady-state time)
40     begin
41       if (errorsignal_a == 1'b1) // A critical timing is active for this input
42         begin
43           outfile = $fopen("errors.txt", "a");
44           $fdisplay(outfile, "Violation_of_critical_timing_in_module_%m;_%0d_ps.\n"
45           ↪ ", $stime);
46           $fclose(outfile);
47           q <= 1'bX; // Set all outputs to unknown
48         end
49       if (errorsignal_a == 0)
50         begin
51           case (cell_state)
52             0: begin
53               q <= #(delay_state0_a_q) !q;
54               errorsignal_a = 1; // Critical timing on this input; assign
55               ↪ immediately
56               errorsignal_a <= #(ct_state0_a_a) 0; // Clear error signal
57               ↪ after critical timing expires
58             end
59           endcase
60         end
61     end
62   end
63 endmodule

```

Listing 5.30: RSFQ BUFFT verilog model.

The digital simulation results for the RSFQ BUFFT is shown in Fig. 5.60 and the Mealy finite state machine diagram, extracted using *TimEx*, is shown in Fig. 5.61.

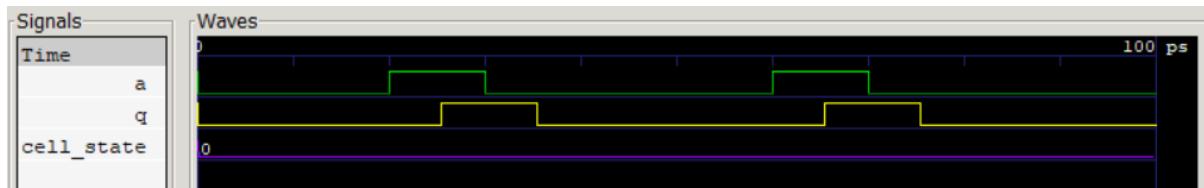


Figure 5.60: RSFQ BUFFT digital simulation results.

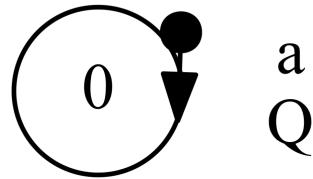


Figure 5.61: RSFQ BUFFT Mealy finite state machine diagram.

Power Consumption

Table 5.26: RSFQ BUFFT power consumption.

Clock rate (GHz)	Static power (nW)	Dynamic power (nW)
1	1321	1.45
2	1321	2.89
5	1321	7.24
10	1321	14.5
20	1321	28.9
50	1321	72.4

5.4 Interface cells

5.4.1 DCSFQ-PTLTX

The RSFQ DCSFQ-PTLTX is an interface cell designed to convert input voltage pulses into SFQ pulses. The DCSFQ-PTLTX has an integrated PTL transmitter and is intended to connect directly to a PTL output.

Schematic

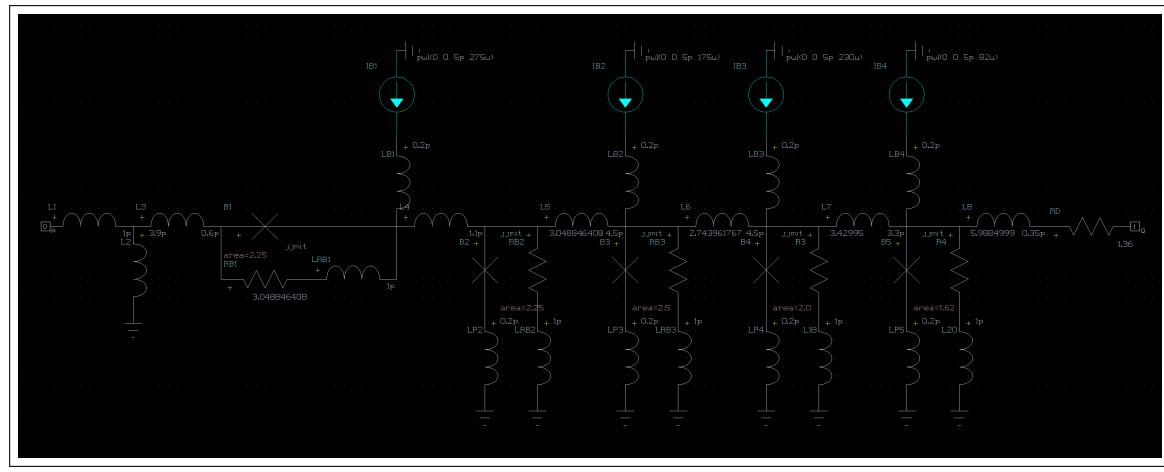


Figure 5.62: Schematic of RSFQ DCSFQ-PTLTX.

Layout

The physical layout for the RSFQ DCSFQ-PTLTX is shown in Fig. 5.63. The layout height is $50 \mu\text{m}$ and the width is $60 \mu\text{m}$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

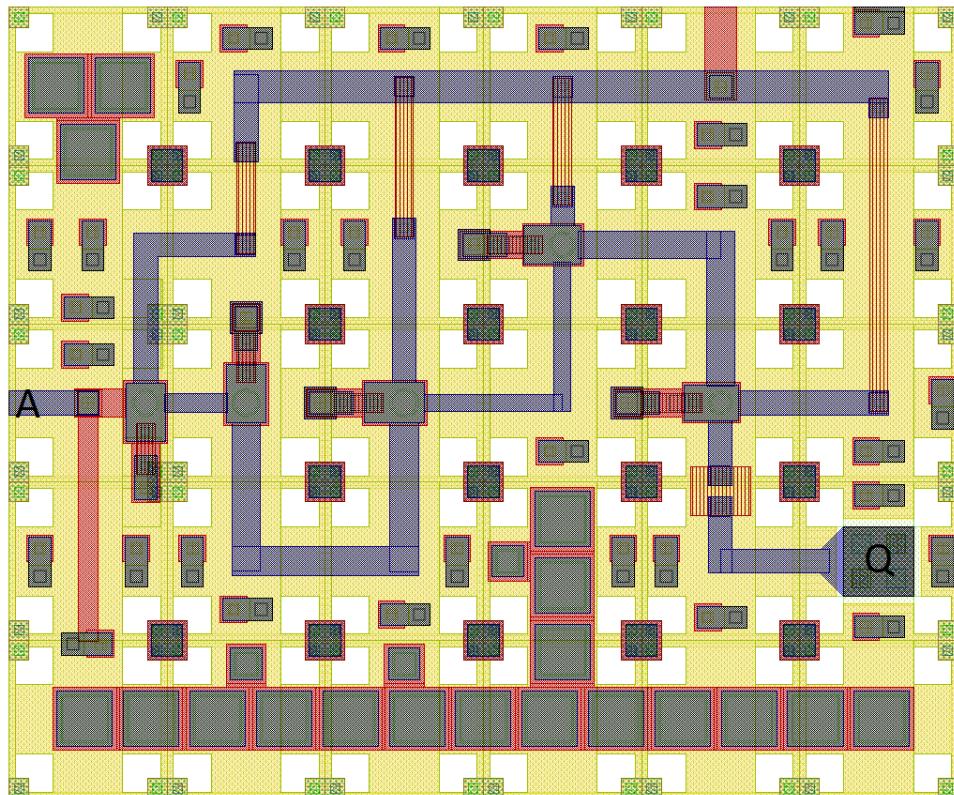


Figure 5.63: RSFQ DCSFQ-PTLTX Layout.

Analog model

```

1  * Back-annotated simulation file written      43 | .param LRB3=(RB3/Rsheet)*Lsheet
2  *   ↪ by InductEx v.6.0.4 on 2-6-21.          44 | .param LRB4=(RB4/Rsheet)*Lsheet
3  * Author: L. Schindler                      45 | .param LRB5=(RB5/Rsheet)*Lsheet
4  * Version: 2.1                                46 |
5  * Last modification date: 2 June 2021        47 | B1 2 3 jjmit area=B1
6  * Last modification by: L. Schindler         48 | B2 6 7 jjmit area=B2
7  *$Ports a q                                  49 | B3 9 10 jjmit area=B3
8  .subckt LSmitll_DCSFQ_PTLTX a q            50 | B4 13 14 jjmit area=B4
9  .model jjmit jj(rtype=1, vg=2.8mV, cap    51 | B5 17 18 jjmit area=B5
   ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA     52 | IB1 0 5 pwl(0 0 5p IB1)
   ↪ )                                         53 | IB2 0 12 pwl(0 0 5p IB2)
10 .param B0=1                                    54 | IB3 0 16 pwl(0 0 5p IB3)
11 .param Ic0=0.0001                            55 | IB4 0 20 pwl(0 0 5p IB4)
12 .param IcRs=100u*6.859904418                 56 | LB1 5 3 3.122E-12
13 .param B0Rs=IcRs/Ic0*B0                     57 | LB2 12 9 2.229E-12
14 .param Rsheet=2                             58 | LB3 16 13 6.294E-13
15 .param Lsheet=1.13e-12                      59 | LB4 20 17 2.139E-12
16 .param LB=0.2p                               60 | L1 a 1 1.293E-12
17 .param LP=0.2p                               61 | L2 1 0 3.9E-12
18 .param B1=2.25                             62 | L3 1 2 5.991E-13
19 .param B2=2.25                             63 | L4 3 6 1.088E-12
20 .param B3=2.5                               64 | L5 6 9 4.514E-12
21 .param B4=2                                65 | L6 9 13 4.542E-12
22 .param B5=1.62                            66 | L7 13 17 3.272E-12
23 .param IB1=275u                           67 | L8 17 21 1.009E-12
24 .param IB2=175u                           68 | LP2 7 0 5.238E-13
25 .param IB3=230u                           69 | LP3 10 0 5.263E-13
26 .param IB4=82u                            70 | LP4 14 0 4.571E-13
27 .param L1=1p                               71 | LP5 18 0 4.763E-13
28 .param L2=3.9p                            72 | LRB1 2 4 LRB1
29 .param L3=0.6p                            73 | LRB2 8 0 LRB2
30 .param L4=1.1p                            74 | LRB3 11 0 LRB3
31 .param L5=4.5p                            75 | LRB4 15 0 LRB4
32 .param L6=4.5p                            76 | LRB5 19 0 LRB5
33 .param L7=3.3p                            77 | RB1 4 3 RB1
34 .param L8=0.35p                           78 | RB2 6 8 RB2
35 .param RD=1.36                           79 | RB3 9 11 RB3
36 .param RB1=B0Rs/B1                      80 | RB4 13 15 RB4
37 .param RB2=B0Rs/B2                      81 | RB5 17 19 RB5
38 .param RB3=B0Rs/B3                      82 | RD 21 q RD
39 .param RB4=B0Rs/B4                      83 | .ends
40 .param RB5=B0Rs/B5                      83 |
41 .param LRB1=(RB1/Rsheet)*Lsheet
42 .param LRB2=(RB2/Rsheet)*Lsheet

```

Listing 5.31: RSFQ DCSFQ-PTLTX JoSIM netlist.

Table 5.27: RSFQ DCSFQ-PTLTX pin list.

Pin	Description
a	Data input
q	Data output

The simulation results for the RSFQ DCSFQ-PTLTX using JoSIM is shown in Fig. 5.64. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the output JJ of pin **q**, and
- (c) the phase over the input JJ of the load circuit connected through a PTL to pin **q**.

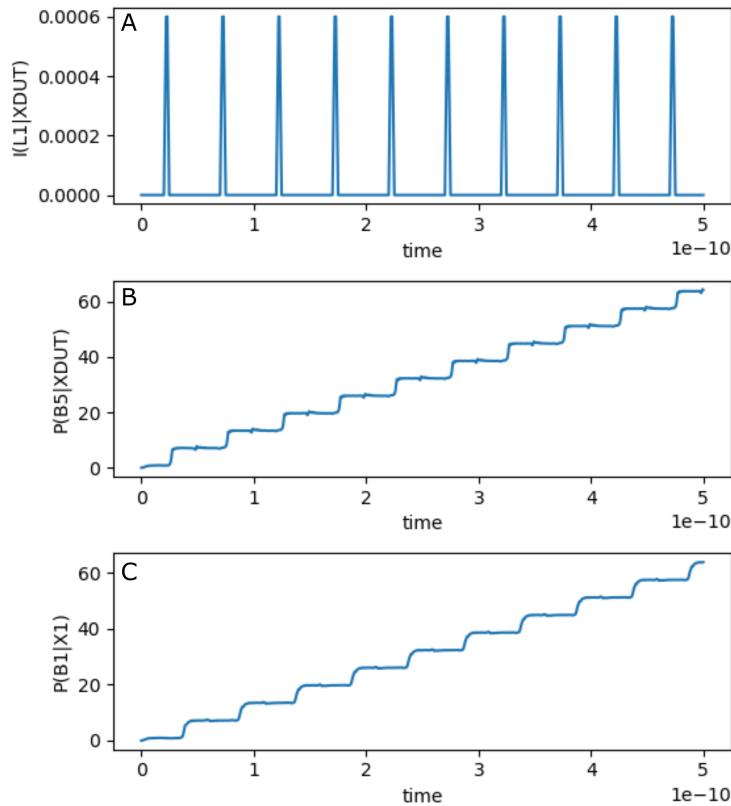


Figure 5.64: RSFQ DCSFQ-PTLTX analog simulation results.

5.4.2 PTLRX-SFQDC

The RSFQ PTLRX-SFQDC is an interface cell designed to convert SFQ pulses to an output voltage level which can be measured by standard equipment. The PTLRX-SFQDC has an integrated PTL receiver and is intended to be connected directly to a PTL input.

Schematic

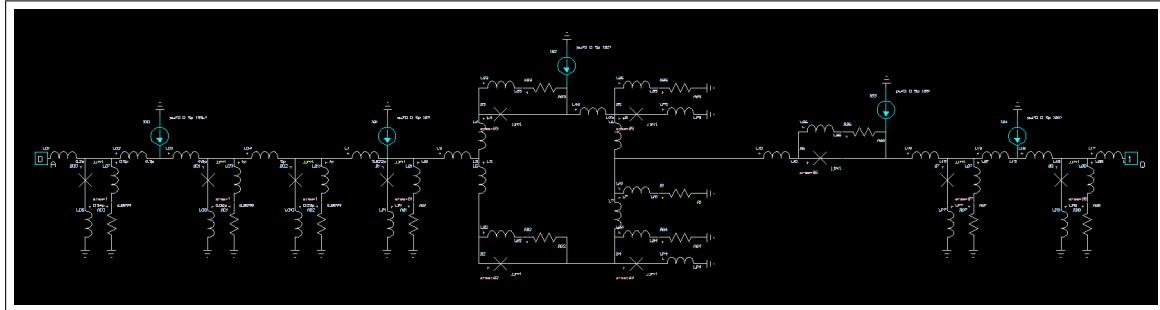


Figure 5.65: Schematic of RSFQ PTLRX-SFQDC.

Layout

The physical layout for the RSFQ PTLRX-SFQDC is shown in Fig. 5.66. The layout height is $70 \mu m$ and the width is $100 \mu m$. The bias line is located on M5 at the top right of the cell. The PTL connection extends to connect to a PTL on either M1 or M3. All bias resistors are designed to be connected to a 2.6 mV voltage source.

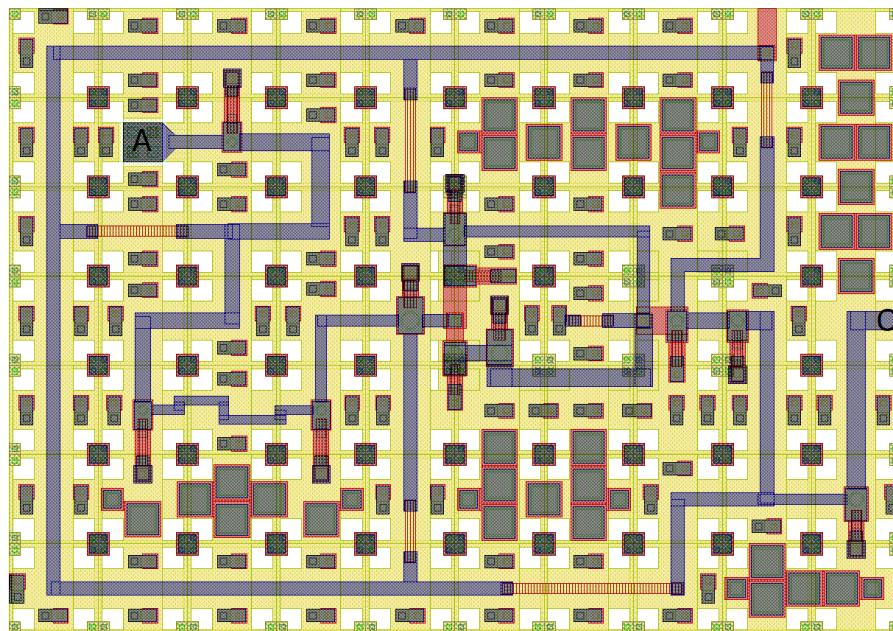


Figure 5.66: RSFQ PTLRX-SFQDC Layout.

Analog model

```

1  * Back-annotated simulation file written      64  .param LRB10=(RB10/Rsheet)*Lsheet
2   ↪ by InductEx v.6.0.4 on 27-4-21.          65  .param LRB11=(RB11/Rsheet)*Lsheet
3  * Author: L. Schindler                      66
4  * Version: 2.1                               67  .param R1=5.74
5  * Last modification date: 26 April 2021     68
6  * Last modification by: L. Schindler        69
7  *$Ports a q                                70  B1 2 3 jjmit area=B1
8 .subckt LSmitll_PTLRX_SFQDC a q           71  B2 6 7 jjmit area=B2
9 .model jjmit jj(rtype=1, vg=2.8mV, cap    72  B3 8 9 jjmit area=B3
10  ↪ =0.07pF, r0=160, rn=16, icrit=0.1mA    73  B4 10 11 jjmit area=B4
11  ↪ )                                         74  B5 13 14 jjmit area=B5
12 .param B0=1                                 75  B6 15 16 jjmit area=B6
13 .param Ic0=0.0001                           76  B7 19 20 jjmit area=B7
14 .param IcRs=100u*6.859904418                77  B8 20 21 jjmit area=B8
15 .param B0Rs=IcRs/Ic0*B0                   78  B9 23 24 jjmit area=B9
16 .param Rsheet=2                            79  B10 26 27 jjmit area=B10
17 .param Lsheet=1.13e-12                     80  B11 30 31 jjmit area=B11
18 .param LB=2p                               81
19 .param LP=0.2p                            82  LP1 3 0 5.068E-13
20 .param B1=1                               83  LP2 7 0 6.182E-13
21 .param B2=1                               84  LP3 9 0 6.097E-13
22 .param B3=1                               85  LP4 11 0 4.785E-13
23 .param B4=3.25                           86  LP6 16 0 4.984E-13
24 .param B5=1.50                           87  LP8 21 0 4.692E-13
25 .param B6=1.75                           88  LP10 27 0 5.018E-13
26 .param B7=2.00                           89  LP11 31 0 5.645E-13
27 .param B8=3.00                           90
28 .param B9=1.50                           91  IB1 0 5 pwl(0 0 5p IB1)
29 .param B10=1.50                          92  IB2 0 12 pwl(0 0 5p IB2)
30 .param B11=2.00                          93  IB3 0 17 pwl(0 0 5p IB3)
31 .param IB1=155u                          94  IB4 0 25 pwl(0 0 5p IB4)
32 .param IB2=280u                          95  IB5 0 29 pwl(0 0 5p IB5)
33 .param IB3=150u                          96
34 .param IB4=220u                          97  LB1 4 5 1.284E-12
35 .param IB5=80u                           98  LB2 10 12 3.934E-12
36 .param LB1=LB                            99  LB3 15 17 1.916E-12
37 .param LB2=LB                            100 LB4 24 25 5.378E-12
38 .param LB3=LB                            101 LB5 28 29 3.797E-12
39 .param LB4=LB                            102
40 .param LB5=LB                            103 L1 a 2 1.524E-12
41 .param RB1=B0Rs/B1                      104 L2 2 4 4.301E-12
42 .param RB2=B0Rs/B2                      105 L3 4 6 4.612E-12
43 .param RB3=B0Rs/B3                      106 L4 6 8 5.035E-12
44 .param RB4=B0Rs/B4                      107 L5 8 10 3.791E-12
45 .param RB5=B0Rs/B5                      108 L6 10 13 8.316E-13
46 .param RB6=B0Rs/B6                      109 L7 14 15 1.167E-12
47 .param RB7=B0Rs/B7                      110 L9 15 18 5.964E-12
48 .param RB8=B0Rs/B8                      111 L10 13 19 1.104E-12
49 .param RB9=B0Rs/B9                      112 L11 20 18 3.24E-12
50 .param RB10=B0Rs/B10                     113 L12 18 22 8.87E-13
51 .param RB11=B0Rs/B11                     114 L14 18 23 5.769E-13
52 .param LRB1=(RB1/Rsheet)*Lsheet         115 L15 24 26 9.383E-13
53 .param LRB2=(RB2/Rsheet)*Lsheet         116 L16 26 28 3.701E-12
54 .param LRB3=(RB3/Rsheet)*Lsheet         117 L17 28 30 2.06E-12
55 .param LRB4=(RB4/Rsheet)*Lsheet         118 L18 30 q 4.08E-12
56 .param LRB5=(RB5/Rsheet)*Lsheet         119
57 .param LRB6=(RB6/Rsheet)*Lsheet         120 R1 22 0 R1
58 .param LRB7=(RB7/Rsheet)*Lsheet         121
59 .param LRB8=(RB8/Rsheet)*Lsheet         122 RB1 2 102 RB1
60 .param LRB9=(RB9/Rsheet)*Lsheet         123 LRB1 102 0 LRB1
61 .param LRB10=(RB10/Rsheet)*Lsheet        124 RB2 6 106 RB2
62 .param LRB11=(RB11/Rsheet)*Lsheet        125 LRB2 106 0 LRB2
63 .param LRB12=(RB12/Rsheet)*Lsheet        126 RB3 8 108 RB3
64 .param LRB13=(RB13/Rsheet)*Lsheet        127 LRB3 108 0 LRB3
65 .param LRB14=(RB14/Rsheet)*Lsheet        128 RB4 10 110 RB4
66 .param LRB15=(RB15/Rsheet)*Lsheet        129 LRB4 110 0 LRB4

```

```

130 | RB5 13 113 RB5
131 | LRB5 113 14 LRB5
132 | RB6 15 115 RB6
133 | LRB6 115 0 LRB6
134 | RB7 19 119 RB7
135 | LRB7 119 20 LRB7
136 | RB8 20 120 RB8
137 | LRB8 120 0 LRB8
138 | RB9 23 123 RB9
139 | LRB9 123 24 LRB9
140 | RB10 26 126 RB10
141 | LRB10 126 0 LRB10
142 | RB11 30 130 RB11
143 | LRB11 130 0 LRB11
144 .ends

```

Listing 5.32: RSFQ PTLRX-SFQDC JoSIM netlist.

Table 5.28: RSFQ PTLRX-SFQDC pin list.

Pin	Description
a	Data input
q	Data output

The simulation results for the RSFQ PTLRX-SFQDC using JoSIM is shown in Fig. 5.67. The figure shows the graph for:

- (a) the current through the input inductor connected to pin **a**,
- (b) the phase over the input JJ of pin **a**,
- (c) the current through the output inductor connected to pin **q**, and
- (c) the voltage over the load resistor connected to pin **q**.

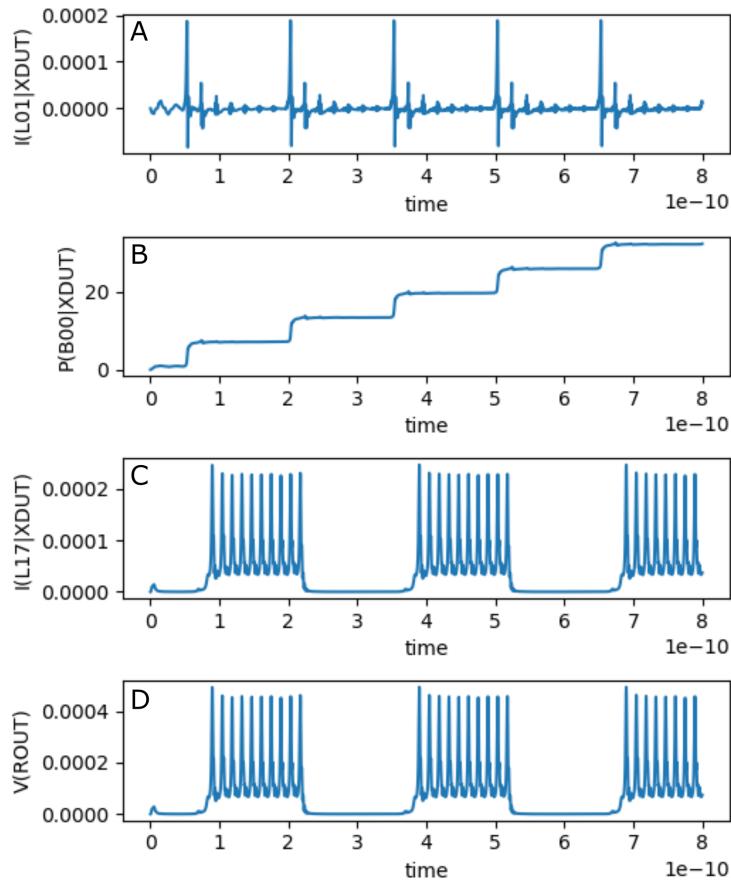


Figure 5.67: RSFQ PTLRX-SFQDC analog simulation results.

Appendices

A. RSFQ PCell Script Example

```

1 import sys
2 # Change this to the location that contains the subcells.py folder
3 subcell_path = '..\subcells'
4 if subcell_path not in sys.path:
5     sys.path.append(subcell_path)
6 import subcells as sc
7 import os
8 import spira.all as spira
9 from spira.technologies.mit.process.database import RDD
10
11 IXPORT = spira.RDD.PLAYER.IXPORT
12 TEXT = spira.Layer(number=182)
13
14 ## Parameterization
15 # Trackpitch in microns
16 tp = 10
17 sc.tp = tp
18
19 # Inductor widths
20 Scaling = (1+(tp-10)*0.25)
21 L1_width = 0.25*tp*Scaling
22 L2_width = 0.135*tp*Scaling
23 L3_width = 0.13*tp*Scaling
24 L4_width = 0.135*tp*Scaling
25 L5_width = 0.25*tp*Scaling
26 LB_width = 0.14*tp*Scaling
27
28 class PCELL(spira.PCell):
29     __name_prefix__ = "LSmitll_BUFF_v2p1"
30     def create_elements(self, elems):
31         IXports = spira.SRef(IX_ports())
32         M6Strips = spira.SRef(M6_strips())
33         M0tracks = spira.SRef(M0_tracks())
34         jjfill = spira.SRef(junction_fill())
35         M4M5M6M7conns = spira.SRef(M4M5M6M7_connections())
36         bias = spira.SRef(biasing())
37         jjs = spira.SRef(junctions())
38         M0blocks = spira.SRef(M0_blocks())
39         tblocks = spira.SRef(trackblocks())
40         elems += [IXports, M6Strips, M0tracks,
41                    jjfill, M4M5M6M7conns, bias, jjs, M0blocks, tblocks]
42         # Ports
43         # Bias1
44         PB1N = spira.Port(name="PB1N", midpoint=bias.reference.elements['bias1'].ports['M6:
45             ↪ PN'].midpoint, process=spira.RDD.PROCESS.M6)
46         PB1S = spira.Port(name="PB1S", midpoint=bias.reference.elements['bias1'].ports['M6:
47             ↪ PS'].midpoint, process=spira.RDD.PROCESS.M6)
48         # Bias2
49         PB2N = spira.Port(name="PB2N", midpoint=bias.reference.elements['bias2'].ports['M6:
50             ↪ PN'].midpoint, process=spira.RDD.PROCESS.M6)
51         PB2S = spira.Port(name="PB2S", midpoint=bias.reference.elements['bias2'].ports['M6:
52             ↪ PS'].midpoint, process=spira.RDD.PROCESS.M6)
53         # Bias3
54         PB3N = spira.Port(name="PB3N", midpoint=bias.reference.elements['bias3'].ports['M6:
55             ↪ PN'].midpoint, process=spira.RDD.PROCESS.M6)

```

```

51      PB3S = spira.Port(name="PB3S",midpoint=bias.reference.elements['bias3'].ports['M6':
52          ↪ PS'].midpoint,process=spira.RDD.PROCESS.M6)
53      # Bias4
54      PB4N = spira.Port(name="PB4N",midpoint=bias.reference.elements['bias4'].ports['M6':
55          ↪ PN'].midpoint,process=spira.RDD.PROCESS.M6)
56      PB4S = spira.Port(name="PB4S",midpoint=bias.reference.elements['bias4'].ports['M6':
57          ↪ PS'].midpoint,process=spira.RDD.PROCESS.M6)
58      # Bias Ends
59      PB1_4_end = spira.Port(name="PB1_4_end",midpoint=(2.5*tp,0.5*tp),process=spira.RDD
60          ↪ .PROCESS.M6)
61      PB2_3_end = spira.Port(name="PB2_3_end",midpoint=(2.5*tp,6.5*tp),process=spira.RDD
62          ↪ .PROCESS.M6)
63      # JJ1
64      PJ1 = spira.Port(name="PJ1",midpoint=jjj.reference.elements['J1'].ports['M6:PJ'].:
65          ↪ midpoint,process=spira.RDD.PROCESS.M6)
66      # JJ2
67      PJ2 = spira.Port(name="PJ2",midpoint=jjj.reference.elements['J2'].ports['M6:PJ'].:
68          ↪ midpoint,process=spira.RDD.PROCESS.M6)
69      # JJ3
70      PJ3 = spira.Port(name="PJ3",midpoint=jjj.reference.elements['J3'].ports['M6:PJ'].:
71          ↪ midpoint,process=spira.RDD.PROCESS.M6)
72      # JJ4
73      PJ4 = spira.Port(name="PJ4",midpoint=jjj.reference.elements['J4'].ports['M6:PJ'].:
74          ↪ midpoint,process=spira.RDD.PROCESS.M6)
75      # Pins
76      PA = spira.Port(name="PA",midpoint=IXports.reference.elements['A'].center,process=
77          ↪ spira.RDD.PROCESS.M6)
78      PA_post = spira.Port(name="PA_post",midpoint=IXports.reference.elements['A'].:
79          ↪ center+(0.2*tp,0),process=spira.RDD.PROCESS.M6)
80      PQ = spira.Port(name="PQ",midpoint=IXports.reference.elements['Q'].center,process=
81          ↪ spira.RDD.PROCESS.M6)
82      PQ_post = spira.Port(name="PQ_post",midpoint=IXports.reference.elements['Q'].:
83          ↪ center-(0.2*tp,0),process=spira.RDD.PROCESS.M6)
84      # Inductors
85      L1 = spira.RoutePath(port1=PA,port2=PJ1,path=[((PA.x+PJ1.x)/2,(PA.y+PJ1.y)/2)],
86          ↪ start_straight=False,end_straight=False,width=IXports.reference.elements['A'.
87          ↪ ].bbox_info.height,layer=sc.M6)
88      L1_post = spira.RoutePath(port1=PA_post,port2=PJ1,path=[((PA_post.x+PJ1.x)/2,(.
89          ↪ PA_post.y+PJ1.y)/2)],start_straight=False,end_straight=False,width=L1_width
90          ↪ ,layer=sc.M6)
91      L2 = spira.RoutePath(port1=PJ1,port2=PJ2,path=[(1.5*tp,3.8975*tp),(1.6*tp,3.8975*.
92          ↪ tp),(1.6*tp,4.25675*tp),(1.405*tp,4.2675*tp),(1.405*tp,4.7225*tp),(1.6*tp
93          ↪ ,4.7225*tp),(1.6*tp,5.0925*tp),(1.5*tp,5.0925*tp)],start_straight=False,
94          ↪ end_straight=False,width=L2_width,layer=sc.M6)
95      L3 = spira.RoutePath(port1=PJ3,port2=PJ2,path=[(3.1025*tp,5.5*tp),(3.1025*tp,5.6*.
96          ↪ tp),(2.7325*tp,5.6*tp),(2.7325*tp,5.405*tp),(2.2775*tp,5.405*tp),(2.2775*tp
97          ↪ ,5.6*tp),(1.9075*tp,5.6*tp),(1.9075*tp,5.5*tp)],start_straight=False,
98          ↪ end_straight=False,width=L3_width,layer=sc.M6)
99      L4 = spira.RoutePath(port1=PJ4,port2=PJ3,path=[(3.5*tp,3.8975*tp),(3.6*tp,3.8975*.
100         ↪ tp),(3.6*tp,4.25675*tp),(3.405*tp,4.2675*tp),(3.405*tp,4.7225*tp),(3.6*tp
101         ↪ ,4.7225*tp),(3.6*tp,5.0925*tp),(3.5*tp,5.0925*tp)],start_straight=False,
102         ↪ end_straight=False,width=L4_width,layer=sc.M6)
103      L5 = spira.RoutePath(port1=PJ4,port2=PQ,path=[((PJ4.x+PQ.x)/2,(PJ4.y+PQ.y)/2)],
104          ↪ start_straight=False,end_straight=False,width=IXports.reference.elements['Q'.
105          ↪ ].bbox_info.height,layer=sc.M6)
106      L5_post = spira.RoutePath(port1=PJ4,port2=PQ_post,path=[((PJ4.x+PQ_post.x)/2,(P.
107          ↪ J4.y+PQ_post.y)/2)],start_straight=False,end_straight=False,width=L5_width,
108          ↪ layer=sc.M6)
109      LB1_1 = spira.RoutePath(port1=PJ1,port2=PB1N,path=[(PB1N.x,PJ1.y)],start_straight=
110          ↪ False,end_straight=False,width=LB_width,layer=sc.M6)
111      LB1_2 = spira.RoutePath(port1=PB1S,port2=PB1_4_end,path=[(PB1S.x,PB1_4_end.y)],
112          ↪ start_straight=False,end_straight=False,width=LB_width,layer=sc.M6)
113      LB2_1 = spira.RoutePath(port1=PJ2,port2=PB2S,path=[(PB2S.x,PJ2.y)],start_straight=
114          ↪ False,end_straight=False,width=LB_width,layer=sc.M6)
115      LB2_2 = spira.RoutePath(port1=PB2N,port2=PB2_3_end,path=[(PB2N.x,PB2_3_end.y)],
116          ↪ start_straight=False,end_straight=False,width=LB_width,layer=sc.M6)
117      LB3_1 = spira.RoutePath(port1=PJ3,port2=PB3S,path=[(PB3S.x,PJ3.y)],start_straight=
118          ↪ False,end_straight=False,width=LB_width,layer=sc.M6)
119      LB3_2 = spira.RoutePath(port1=PB3N,port2=PB2_3_end,path=[(PB3N.x,PB2_3_end.y)],
120          ↪ start_straight=False,end_straight=False,width=LB_width,layer=sc.M6)

```

```

85     LB4_1 = spira.RoutePath(port1=PJ4, port2=PB4N, path=[(PJ4.x, PB4N.x)], start_straight=
86         ↪ False, end_straight=False, width=LB_width, layer=sc.M6)
87     LB4_2 = spira.RoutePath(port1=PB4S, port2=PB1_4_end, path=[(PB4S.x, PB1_4_end.y)],
88         ↪ start_straight=False, end_straight=False, width=LB_width, layer=sc.M6)
89     elems += [L1, L1_post, L2, L3, L4, L5, L5_post,
90             ↪ LB1_1, LB1_2, LB2_1, LB2_2, LB3_1, LB3_2, LB4_1, LB4_2]
91     # Text Labels
92     elems += spira.Label(text="a", position=(0*tp, 3.5*tp), layer=TEXT)
93     elems += spira.Label(text="P1_M6_M4", position=(0*tp, 3.5*tp), layer=TEXT)
94     elems += spira.Label(text="J1_M6_M5", position=(1.5*tp, 3.5025*tp), layer=TEXT)
95     elems += spira.Label(text="PB1_M6_M4", position=(1.5*tp, 2.4075*tp), layer=TEXT)
96     elems += spira.Label(text="J2_M6_M5", position=(1.505*tp, 5.4975*tp), layer=TEXT)
97     elems += spira.Label(text="PB2_M6_M4", position=(0.5*tp, 5.53*tp), layer=TEXT)
98     elems += spira.Label(text="J3_M6_M5", position=(3.495*tp, 5.5025*tp), layer=TEXT)
99     elems += spira.Label(text="PB3_M6_M4", position=(4.5*tp, 5.53*tp), layer=TEXT)
100    elems += spira.Label(text="J4_M6_M5", position=(3.4875*tp, 3.5*tp), layer=TEXT)
101    elems += spira.Label(text="PB4_M6_M4", position=(2.495*tp, 2.5075*tp), layer=TEXT)
102    elems += spira.Label(text="q", position=(5*tp, 3.5*tp), layer=TEXT)
103    elems += spira.Label(text="P2_M6_M4", position=(5*tp, 3.5*tp), layer=TEXT)
104    elems += spira.Label(text="bias_in", position=(0*tp, 6.5*tp), layer=TEXT)
105    elems += spira.Label(text="bias_out", position=(5*tp, 6.5*tp), layer=TEXT)
106    return elems
107
108 class M6_strips(spira.Cell):
109     __name_prefix__ = "M6_strips"
110     def create_elements(self, elems):
111         elems += spira.Box(layer=sc.M6, width=0.025*tp, height=0.315*tp, center=(0.0125*tp
112             ↪ , 6.4925*tp))
113         elems += spira.Box(layer=sc.M6, width=0.025*tp, height=0.315*tp, center=(4.9875*tp
114             ↪ , 6.4925*tp))
115         elems += spira.Box(layer=sc.M6, width=0.315*tp, height=0.025*tp, center=(2.5075*tp
116             ↪ , 6.9875*tp))
117         elems += spira.Box(layer=sc.M6, width=0.315*tp, height=0.025*tp, center=(2.4925*tp
118             ↪ , 0.0125*tp))
119         elems += spira.Box(layer=sc.M5, width=0.025*tp, height=0.315*tp, center=(0.0125*tp
120             ↪ , 6.4925*tp))
121         elems += spira.Box(layer=sc.M5, width=0.025*tp, height=0.315*tp, center=(4.9875*tp
122             ↪ , 6.4925*tp))
123         elems += spira.Box(layer=sc.M5, width=0.315*tp, height=0.025*tp, center=(2.5075*tp
124             ↪ , 6.9875*tp))
125         elems += spira.Box(layer=sc.M5, width=0.315*tp, height=0.025*tp, center=(2.4925*tp
126             ↪ , 0.0125*tp))
127     return elems
128
129 class IX_ports(spira.Cell):
130     __name_prefix__ = "IX_ports"
131     def create_elements(self, elems):
132         elems += spira.Box(layer=EXPORT, width=0*tp, height=0.2*tp, center=(0*tp, 3.5*tp),
133             ↪ alias='A')
134         elems += spira.Box(layer=EXPORT, width=0*tp, height=0.2*tp, center=(5*tp, 3.5*tp),
135             ↪ alias='Q')
136         elems += spira.Box(layer=EXPORT, width=0.052*tp, height=0.052*tp, center=(4.5*tp
137             ↪ , 5.53*tp))
138         elems += spira.Box(layer=EXPORT, width=0.052*tp, height=0.052*tp, center=(2.5*tp
139             ↪ , 2.502*tp))
140         elems += spira.Box(layer=EXPORT, width=0.052*tp, height=0.052*tp, center=(1.5*tp
141             ↪ , 2.407*tp))
142         elems += spira.Box(layer=EXPORT, width=0.052*tp, height=0.052*tp, center=(0.5*tp
143             ↪ , 5.53*tp))
144     return elems
145
146 class M0_tracks(spira.Cell):
147     __name_prefix__ = "M0_tracks"
148     def create_elements(self, elems):
149         shape = spira.Shape(points=[(2.375*tp, 0.5*tp), (2.375*tp, 6.5*tp), (2.625*tp, 6.5*tp)
150             ↪ , (2.625*tp, 0.5*tp)])
151         elems += spira.Polygon(shape, layer=sc.M0)
152         shape = spira.Shape(points=[(0*tp, 6.35*tp), (0*tp, 6.65*tp), (5*tp, 6.65*tp), (5*tp
153             ↪ , 6.35*tp)])
154         elems += spira.Polygon(shape, layer=sc.M0)

```

```

137     shape = spira.Shape(points=[(2*tp, 0.125*tp),(2*tp,5.875*tp),(2.04*tp,5.875*tp)
138                           ↪ ,(2.04*tp,0.125*tp)])
139     elems += spira.Polygon(shape,layer=sc.M0)
140     shape = spira.Shape(points=[(4.125*tp,4.96*tp),(4.125*tp,5*tp),(4.875*tp,5*tp)
141                           ↪ ,(4.875*tp,4.96*tp)])
142     elems += spira.Polygon(shape,layer=sc.M0)
143     shape = spira.Shape(points=[(0.125*tp,6*tp),(0.125*tp,6.04*tp),(1.875*tp,6.04*tp)
144                           ↪ ,(1.875*tp,6*tp)])
145     elems += spira.Polygon(shape,layer=sc.M0)
146     shape = spira.Shape(points=[(3.125*tp,6*tp),(3.125*tp,6.04*tp),(4.875*tp,6.04*tp)
147                           ↪ ,(4.875*tp,6*tp)])
148     elems += spira.Polygon(shape,layer=sc.M0)
149     shape = spira.Shape(points=[(0.28*tp,0.28*tp),(0.28*tp,0.72*tp),(0.72*tp,0.72*tp)
150                           ↪ ,(0.72*tp,0.28*tp)])
151     elems += spira.Polygon(shape,layer=sc.M0)
152     shape = spira.Shape(points=[(0*tp,0.875*tp),(0*tp,1*tp),(0.125*tp,1*tp),(0.125*tp
153                           ↪ ,0.875*tp)])
154     elems += spira.Polygon(shape,layer=sc.M0)
155     shape = spira.Shape(points=[(0.875*tp,0.875*tp),(0.875*tp,1*tp),(1*tp,1*tp),(1*tp
156                           ↪ ,0.875*tp)])
157     elems += spira.Polygon(shape,layer=sc.M0)
158     shape = spira.Shape(points=[(0.875*tp,0*tp),(0.875*tp,0.125*tp),(1*tp,0.125*tp)
159                           ↪ ,(1*tp,0*tp)])
160     elems += spira.Polygon(shape,layer=sc.M0)
161     shape = spira.Shape(points=[(0.875*tp,0.875*tp),(0.875*tp,1*tp),(1*tp,1*tp),(1*tp
162                           ↪ ,0.875*tp)])
163     elems += spira.Polygon(shape,layer=sc.M0)
164     shape = spira.Shape(points=[(0*tp,0.875*tp),(0*tp,1*tp),(0.125*tp,1*tp),(0.125*tp
165                           ↪ ,0.875*tp)])
166     elems += spira.Polygon(shape,layer=sc.M0)
167     return elems
168
169 class junction_fill(spira.Cell):
170     __name_prefix__ = "junction_fill"
171     def create_elements(self, elems):
172         sys.stdout.write("Adding_3um_junction_fill.\n")
173         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(4.5*tp,0.5*tp))
174         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(0.5*tp,0.5*tp))
175         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(4.5*tp,2.5*tp))
176         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(4.5*tp,1.5*tp))
177         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(2.5*tp,3.5*tp))
178         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(3.5*tp,1.5*tp))
179         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(2.5*tp,4.5*tp))
180         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(0.5*tp,4.5*tp))
181         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(0.5*tp,1.5*tp))
182         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(0.5*tp,2.5*tp))
183         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(4.5*tp,4.5*tp))
184         elems += spira.SRef(sc.ls_FakeJJ_3umx3um(),midpoint=(3.5*tp,0.5*tp))
185         sys.stdout.write("Adding_1.5um_junction_fill.\n")
186         for y in range(1, 7):
187             for x in range(1, 5):
188                 elems += spira.SRef(sc.ls_FakeJJ_1p5x1p5um(), midpoint=(0.0+x*tp,0.0+y*tp)
189                           ↪ )
190         return elems
191
192 class M4M5M6M7_connections(spira.Cell):
193     __name_prefix__ = "M4M5M6M7_connections"
194     def create_elements(self, elems):

```

```

192     sys.stdout.write('Adding_M4M5M6M7_connections.\n')
193     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(0.335*tp, 1.93*tp))
194     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(1.335*tp, 0.125*tp))
195     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(4.335*tp, 0.93*tp))
196     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(3.335*tp, 0.93*tp))
197     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(4.335*tp, 3.125*tp))
198     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(2.335*tp, 0.025*tp))
199     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(4.335*tp, 1.93*tp))
200     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(4.335*tp, 3.735*tp))
201     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(0.335*tp, 0.93*tp))
202     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(3.93*tp, 1.335*tp),
203                         ↪ transformation=sc.m45)
204     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(4.835*tp, 6.335*tp),
205                         ↪ transformation=sc.m45)
206     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(1.93*tp, 2.335*tp),
207                         ↪ transformation=sc.m45)
208     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(1.93*tp, 4.335*tp),
209                         ↪ transformation=sc.m45)
210     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(0.93*tp, 4.335*tp),
211                         ↪ transformation=sc.m45)
212     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(3.93*tp, 4.335*tp),
213                         ↪ transformation=sc.m45)
214     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(1.93*tp, 1.335*tp),
215                         ↪ transformation=sc.m45)
216     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(3.93*tp, 0.335*tp),
217                         ↪ transformation=sc.m45)
218     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(2.845*tp, 0.335*tp),
219                         ↪ transformation=sc.m45)
220     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(0.93*tp, 0.335*tp),
221                         ↪ transformation=sc.m45)
222     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(1.125*tp, 2.335*tp),
223                         ↪ transformation=sc.m45)
224     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(4.735*tp, 5.335*tp),
225                         ↪ transformation=sc.m45)
226     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(2.93*tp, 4.335*tp),
227                         ↪ transformation=sc.m45)
228     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(0.025*tp, 6.335*tp),
229                         ↪ transformation=sc.m45)
230     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(1.125*tp, 1.335*tp),
231                         ↪ transformation=sc.m45)
232     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(2.735*tp, 1.335*tp),
233                         ↪ transformation=sc.m45)
234     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(0.125*tp, 5.335*tp),
235                         ↪ transformation=sc.m45)
236     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(3.805*tp, 2.335*tp),
237                         ↪ transformation=sc.m45)
238     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(3.665*tp, 6.735*tp),
239                         ↪ transformation=sc.m90)
240     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(4.665*tp, 5.055*tp),
241                         ↪ transformation=sc.m90)
242     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(0.665*tp, 3.125*tp),
243                         ↪ transformation=sc.m90)
244     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(2.665*tp, 4.93*tp),
245                         ↪ transformation=sc.m90)
246     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(0.665*tp, 3.735*tp),
247                         ↪ transformation=sc.m90)
248     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(2.665*tp, 3.93*tp),
249                         ↪ transformation=sc.m90)
250     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(1.665*tp, 6.735*tp),
251                         ↪ transformation=sc.m90)
252     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(0.665*tp, 6.735*tp),
253                         ↪ transformation=sc.m90)
254     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(0.665*tp, 5.055*tp),
255                         ↪ transformation=sc.m90)
256     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(2.665*tp, 6.835*tp),
257                         ↪ transformation=sc.m90)
258     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(2.665*tp, 2.735*tp),
259                         ↪ transformation=sc.m90)
260     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(3.665*tp, 2.125*tp),
261                         ↪ transformation=sc.m90)

```

```

232     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(2.665*tp, 5.93*tp),
233         ↪ transformation=sc.m90)
234     elems += spira.SRef(sc.ls_conn_M4M5M6M7(), midpoint=(4.665*tp, 6.735*tp),
235         ↪ transformation=sc.m90)
236     return elems
237
238 class biasing(spira.Cell):
239     __name_prefix__ = "biasing"
240     def create_elements(self, elems):
241         elems += spira.SRef(sc.ls_ib_175(), midpoint=(2.5*tp, 1.55*tp), alias="bias4")
242         elems += spira.SRef(sc.ls_ib_175(), midpoint=(1.5*tp, 1.455*tp), alias="bias1")
243         elems += spira.SRef(sc.ls_ib_235(), midpoint=(4.5*tp, 5.475*tp), alias="bias3")
244         elems += spira.SRef(sc.ls_ib_235(), midpoint=(0.5*tp, 5.475*tp), alias="bias2")
245     return elems
246
247 class junctions(spira.Cell):
248     __name_prefix__ = "junctions"
249     def create_elements(self, elems):
250         elems += spira.SRef(sc.ls_jj_250_sg(), midpoint=(3.5*tp, 5.5*tp), alias="J3")
251         elems += spira.SRef(sc.ls_jj_250_sg(), midpoint=(3.5*tp, 3.5*tp), transformation=sc.
252             ↪ r90, alias="J4")
253         elems += spira.SRef(sc.ls_jj_250_sg(), midpoint=(1.5*tp, 3.5*tp), transformation=sc.
254             ↪ r270, alias="J1")
255         elems += spira.SRef(sc.ls_jj_250_sg(), midpoint=(1.5*tp, 5.5*tp), alias="J2")
256     return elems
257
258 class M0_blocks(spira.Cell):
259     __name_prefix__ = "M0_blocks"
260     def create_elements(self, elems):
261         sys.stdout.write("Adding_M0_trackblocks.\n")
262         for y in range(0, 6):
263             for x in range(0, 5):
264                 if (x != 2):
265                     elems += spira.SRef(sc.ls_tr_M0(), midpoint=(0+x*tp, 0+y*tp))
266     return elems
267
268 class trackblocks(spira.Cell):
269     __name_prefix__ = "trackblocks"
270     def create_elements(self, elems):
271         sys.stdout.write("Adding_trackblocks.\n")
272         for y in range(0, 7):
273             for x in range(0, 5):
274                 if ((x == 2) and (y in [0,6])):
275                     elems += spira.SRef(sc.ls_tr_bias_pillar_M0M6(), midpoint=(0+x*tp, 0+y*
276                         ↪ tp))
277                 else:
278                     elems += spira.SRef(sc.ls_tr_u_M4_alt(), midpoint=(0+x*tp, 0+y*tp))
279                     elems += spira.SRef(sc.ls_tr_M7(), midpoint=(0+x*tp, 0+y*tp))
280     return elems
281
282 sys.stdout.write("Adjusting_settings.\n")
283 F = RDD.FILTERS.OUTPUT.PORTS
284 F['cell_ports'] = False
285 F['edge_ports'] = False
286 F['contact_ports'] = False
287 F = RDD.FILTERS.PCELL.DEVICE
288 F['boolean'] = True
289 F['contact_attach'] = True
290 F = RDD.FILTERS.PCELL.CIRCUIT
291 F['boolean'] = False
292 D = PCELL()
293 sys.stdout.write("Writing_output.\n")
294 D.gdsii_output(os.path.splitext(__file__)[0])

```

Listing A.1: RSFQ BUFF PCell Script.

Bibliography

- [1] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, “Adiabatic quantum-flux-parametron cell library adopting minimalist design,” *J. Appl. Phys.*, vol. 117, no. 17, p. 173912, May 2015. DOI: [10.1063/1.4919838](https://doi.org/10.1063/1.4919838).
- [2] (2017). XicTools suite, [Online]. Available: <http://www.wrcad.com/xictools/index.html>.
- [3] gEDA Project. (2021), [Online]. Available: <http://wiki.geda-project.org/geda:gaf>.
- [4] KLayout. (2021), [Online]. Available: <https://www.klayout.de/>.
- [5] C. J. Fourie, “Full-Gate Verification of Superconducting Integrated Circuit Layouts With InductEx,” *IEEE Trans. Appl. Supercond.*, vol. 25, no. 1, Feb. 2015.
- [6] J. A. Delport, K. Jackman, P. Le Roux, and C. J. Fourie, “JoSIM—Superconductor SPICE Simulator,” *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, pp. 1–5, 2019.
- [7] C. L. Ayala, O. Chen, and N. Yoshikawa, “AQFPTX: Adiabatic Quantum-Flux-Parametron timing extraction tool,” in *2019 IEEE International Superconductive Electronics Conference (ISEC)*, Jul. 2019, pp. 1–3.
- [8] S. Williams. (2018). Icarus verilog, [Online]. Available: <http://iverilog.icarus.com/>.
- [9] (2018). Gtkwave, [Online]. Available: <http://gtkwave.sourceforge.net/>.
- [10] T. Yamae, N. Takeuchi, and N. Yoshikawa, “Systematic method to evaluate energy dissipation in adiabatic quantum-flux-parametron logic,” *J. Appl. Phys.*, vol. 126, no. 17, p. 173903, Nov. 2019.
- [11] C. J. Fourie, C. L. Ayala, L. Schindler, T. Tanaka, and N. Yoshikawa, “Design and characterization of track routing architecture for RSFQ and AQFP circuits in a multilayer process,” *IEEE Trans. Appl. Supercond.*, vol. 30, no. 6, pp. 1–9, Sep. 2020.
- [12] K. K. Likharev and V. K. Semenov, “RSFQ Logic/Memory Family: A New Josephson-Junction Technology for Sub-Terahertz-Clock-Frequency Digital Systems,” *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3–28, Mar. 1991.
- [13] Office of the Director of National Intelligence, *Proposers’ Day Notification for SuperTools, Intelligence advanced research projects activity*, Jan. 2016.

- [14] C. J. Fourie, K. Jackman, M. M. Botha, S. Razmkhah, P. Febvre, C. L. Ayala, Q. Xu, N. Yoshikawa, E. Patrick, M. Law, Y. Wang, M. Annavaram, P. Beerel, S. Gupta, S. Nazarian, and M. Pedram, “ColdFlux Superconducting EDA and TCAD Tools Project: Overview and Progress,” *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, pp. 1–7, Aug. 2019, Art. no. 1300407, ISSN: 1051-8223. DOI: 10.1109/TASC.2019.2892115.
- [15] Whiteley Research, Inc. (2017), [Online]. Available: <http://www.wrcad.com>.
- [16] J. A. Delport. (2020). JoSIM, [Online]. Available: <https://github.com/JoeuDelp/JoSIM/>.
- [17] P. le Roux. (2020). JoSIM-tools, [Online]. Available: <https://github.com/pleroux0/josim-tools>.
- [18] C. J. Fourie. (2018). TimEx, [Online]. Available: <https://github.com/sunmagnetics/TimEx>.
- [19] C. J. Fourie, “Extraction of DC-Biased SFQ Circuit Verilog Models,” *IEEE Trans. Appl. Supercond.*, vol. 28, no. 6, Sep. 2018, Art. no. 1300811, ISSN: 1051-8223. DOI: 10.1109/TASC.2018.2829776.
- [20] InductEx. (2018), [Online]. Available: <https://www.inductex.info>.
- [21] T. Ortlepp, O. Wetzstein, S. Engert, J. Kunert, and H. Toepfer, “Reduced power consumption in superconducting electronics,” *IEEE Transactions on Applied Superconductivity*, vol. 21, no. 3, pp. 770–775, 2011, Art. no. 5740379.