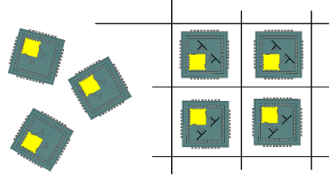


success



Silicon-based Ultra Compact Cost-efficient System Design for mmWave Sensors “SUCCESS”

Deliverable
D5.2

Mm-wave SoC Integration report

By: IHP

Contributors:

Miroslav Marinkovic, Xin Fan (IHP)

Abstract

This document describes the architecture, basic testing environment and preliminary measurement results of the mm-wave SoC radar. This chip includes the RF front-end (with digital control) as well as the baseband processor (BB) integrated into a single-die.

Keywords

Mm-wave sensor, SoC, RF front-end, BB processor, Globally Asynchronous Locally Synchronous (GALS) design, Design-For-Testability (DFT)

Silicon-based Ultra Compact Cost-efficient System Design for mmWave Sensors “SUCCESS”	1
Abstract	1
1. Introduction.....	3
2. Architecture of Lighthouse chip	4
2.1 Building blocks and pin description	4
2.2 GALS FMCW coprocessor.....	8
2.2.1 GALS partitioning of the FMCW Coprocessor	9
2.2.2 Interface circuits design	10
2.2.3 Timing Convergence on handshake signals.....	11
2.2.4 Working mode configuration.....	12
2.3 DFT in BB processor	13
2.3.1 Scan Test.....	13
2.3.2 BIST Test	15
3. Testing Environment Setup and Preliminary Test Results	15
3.1 Testing Environment Setup.....	15
3.2 Preliminary Test Results	16
4. Conclusion and Further Work.....	18

1. Introduction

This report presents the architecture, the basic testing environment and preliminary measurement results of the mm-wave SoC radar (the “*Lighthouse*” chip). A description of the chip architecture is first shown with a special focus on the GALS part of the *Lighthouse* chip. Then the testing environment which was set up in IHP was discussed. Furthermore, we exhibit the preliminary measurement results on the main part of the *Lighthouse* chip - GALS and synchronous FMCW coprocessor of the BB processor. Finally, we are providing conclusion and further work.

Since we did not implement a standalone BB processor, the deliverable D4.4 “Embedded Baseband Processor in Silicon Test Report” is included in this D5.2 Deliverable. We have integrated the RF frond-end and BB processor into a single die (the “*Lighthouse*” chip) in the first shot.

2. Architecture of Lighthouse chip

2.1 Building blocks and pin description

The top-level architecture of the *Lighthouse* chip is shown in Fig. 1.

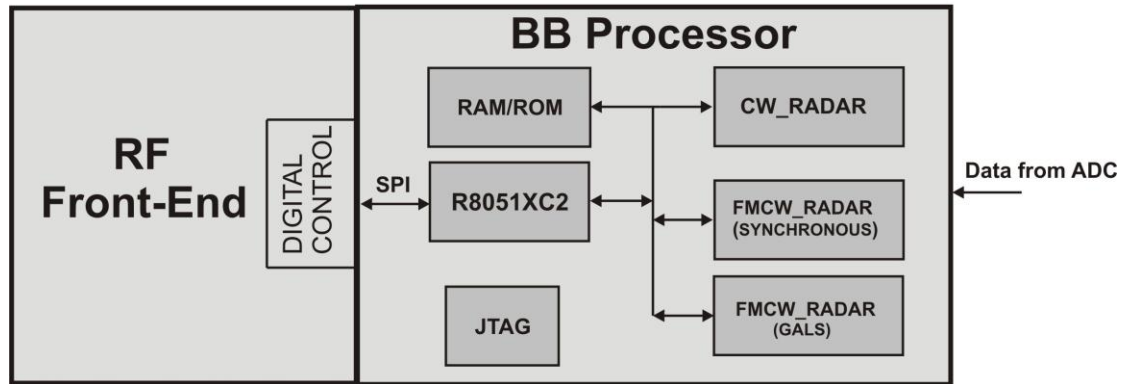


Figure 1. Architecture of Lighthouse chip

The Lighthouse chip consists of the three main components:

- RF Front-End
- Digital control of RF Front-End
- BB Processor

The layout view of the *Lighthouse* chip is shown in Fig. 2, whereas the basic chip parameters are summarized in Table 1. Additionally, the pin descriptions are summarized in Table 2.

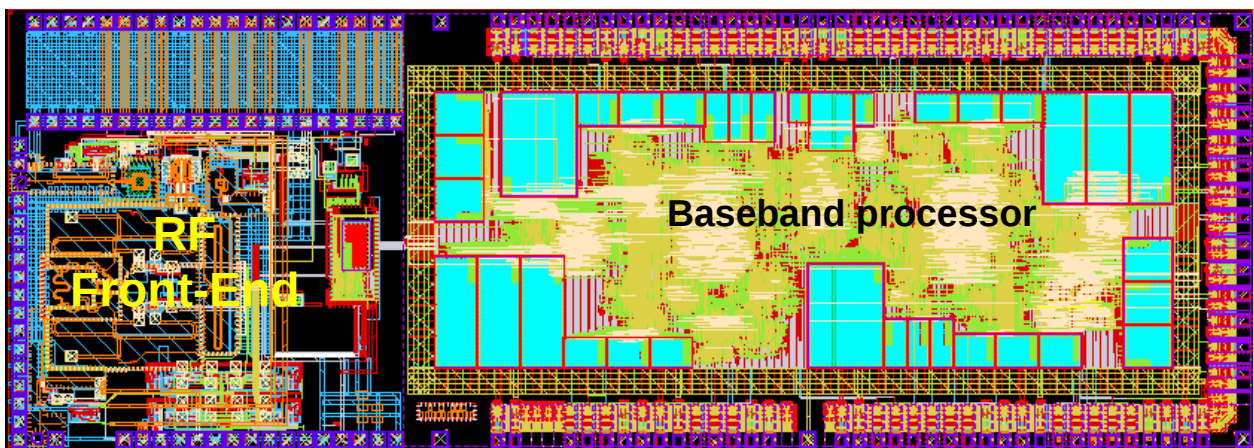


Figure 2. Layout view of Lighthouse chip

Table 1. Lighthouse chip parameters

Lighthouse chip	
Total chip area	17.1 mm ²
BB processor chip area	11.8 mm ²
Total number of pads	149
Supply Voltage	1.2V, 2.5V, 3.3V
Number of power pads	80
Number of I/O pads	64
Number of EMI and substrate pads	5

Table 2. Lighthouse pin description

Nr.	Name	Type	Dir	Str	Pol	Description
1	gnd	AGND				analog ground
2	Rfin	NC	I			122GHz input, leave it open in digital test
3	gnd	AGND				
4	vct_LNA	NC				LNA current control input, default open
5	vdd33	VDD5				high voltage CMOS supply 3.3V
6	vdd12	VDD6				1.2V CMOS supply
7	gnd	AGND				
8	vdd25	VDD7				2.5V LNA supply 20mA
9	gnd	AGND				
10	oib	ASIG	O			output signal I bar, DC measurement
11	oi	ASIG	O			output signal I,DC measurement
12	gnd	AGND				
13	oqb	ASIG	O			output signal Q bar, DC measurement
14	oq	ASIG	O			output signal Q,DC measurement
15	gnd	AGND				
16	vdd12	VDD6				1.2V CMOS supply
17	gnd	AGND				
18	gnd	AGND				
19	gnd	AGND				
20	sub_pad_ul	NC	I/O			substrate contact pad upper left
21	vss	DGND				
22	vdd	VDD2				1.2V CMOS supply
23	adc_sdo	DSIG	I			ADC serial interface data
24	adc_clk	DSIG	I			Clock used to generate adc_sck
25	adc_sck	DSIG	O	8mA		ADC serial interface clock
26	vddio	VDD1				3.3V CMOS supply
27	vss	DGND				
28	vss	DGND				
29	vdd	VDD2				1.2V CMOS supply
30	gp_out15	DSIG	O	8mA		General purpose output
31	eeeprom_misoi	DSIG	I			EEPROM SPI data in
32	eeeprom_scko	DSIG	O	8mA		EEPROM SPI clock
33	vddio	VDD1				3.3V CMOS supply
34	vss	DGND				

35	emi_pad	NC	I/O			emi pad
36	vss	DGND				
37	vdd	VDD2				1.2V CMOS supply
38	eeeprom_mosio	DSIG	O	8mA		EEPROM SPI data out
39	eeeprom_ssn1	DSIG	O	8mA		EEPROM SPI chip select
40	op_state[0]	DSIG	O	8mA		Indication of baseband processor status, bit 0
41	op_state[1]	DSIG	O	8mA		Indication of baseband processor status, bit 1
42	op_state[2]	DSIG	O	8mA		Indication of baseband processor status, bit 2
43	op_state[3]	DSIG	O	8mA		Indication of baseband processor status, bit 3
44	vddio	VDD1				3.3V CMOS supply
45	vss	DGND				
46	vss	DGND				
47	vdd	VDD2				1.2V CMOS supply
48	tcclk[0]	DSIG	O	8mA		GALS clock 0
49	tcclk[1]	DSIG	O	8mA		GALS clock 1
50	tcclk[2]	DSIG	O	8mA		GALS clock 2
51	tcclk[3]	DSIG	O	8mA		GALS clock 3
52	tcclk[4]	DSIG	O	8mA		GALS clock 4
53	vddio	VDD1				3.3V CMOS supply
54	vss	DGND				
55	vss	DGND				
56	vdd	VDD2				1.2V CMOS supply
57	sub_pad_ll	NC	I/O			substrate contact pad lower left
58	reset	DSIG	I		Hi	reset
59	test_mode	DSIG	I		Hi	scan-chain test mode
60	test_se	DSIG	I			scan-chain test enable
61	tck	DSIG	I			JTAG clock
62	vddio	VDD1				3.3V CMOS supply
63	vss	DGND				
64	vss	DGND				
65	vdd	VDD2				1.2V CMOS supply
66	trst	DSIG	I		Hi	JTAG reset
67	tms	DSIG	I			JTAG test mode
68	tdi	DSIG	I			JTAG data in
69	tdo	DSIG	O	8mA		JTAG data out
70	bist_ok	DSIG	O	8mA		BIST OK of baseband processor, FMCW mode
71	sub_pad_lr	NC	I/O			substrate contact pad lower right
72	vddio	VDD1				3.3V CMOS supply
73	vss	DGND				
74	vss	DGND				
75	vdd	VDD2				1.2V CMOS supply
76	msck	DSIG	I			HOST interface serial clock
77	mcs	DSIG	I			HOST interface chip select
78	msda	DSIG	I			HOST interface data in
79	ssda	DSIG	O	8mA		HOST interface data out
80	proc_en	DSIG	O	8mA		Enable signal for ADC operation
81	pll_le	DSIG	O	8mA		PLL serial interface load enable
82	vddio	VDD1				3.3V CMOS supply

83	vss	DGND				
84	vss	DGND				
85	vdd	VDD2				1.2V CMOS supply
86	pll_clock	DSIG	O	8mA		PLL serial interface clock
87	pll_data	DSIG	O	8mA		PLL serial interface data
88	pll_lock	DSIG	I			PLL lock
89	host_int	DSIG	O	8mA		Interrupt output
90	auxdac_sclk	DSIG	O	8mA		Auxiliary DAC serial clock
91	auxdac_syncn	DSIG	O	8mA		Auxiliary DAC serial frame synchronization
92	vddio	VDD1				3.3V CMOS supply
93	vss	DGND				
94	vss	DGND				
95	vdd	VDD2				1.2V CMOS supply
96	auxdac_din	DSIG	O	8mA		Auxiliary DAC serial data in
97	auxdac_dout	DSIG	I			Auxiliary DAC serial data out
98	auxadc_sclk	DSIG	O	8mA		Auxiliary ADC serial clock
99	auxadc_csn	DSIG	O	8mA		Auxiliary ADC chip select
100	auxadc_dout	DSIG	I			Auxiliary ADC data out
101	adc_conv	DSIG	O	8mA		ADC serial conversion start
102	vddio	VDD1				3.3V CMOS supply
103	vss	DGND				
104	vss	DGND				
105	vdd	VDD2				1.2V CMOS supply
106	auxadc_din	DSIG	O	8mA		Auxiliary ADC data in
107	start_bist_dp	DSIG	I		Hi	Start BIST of digital control
108	bist_ok_dp	DSIG	O	8mA		BIST OK of digital control
109	clk	DSIG	I			clock 50 MHz
110	vss	DGND				
111	vdd	VDD2				1.2V CMOS supply
112	sub_pad_ur	NC	I/O			substrate contact pad upper right
113	vdd33	VDD5				3.3V supply FMU 90mA
114	vdd12	VDD6				1.2V cmos FMU supply, 8mA
115	vctl_vco	ASIG	O			0-3.3V VCO control output, direct feed to vfine 122. pin , DC measurement
116	vctl_vco_1.2V	ASIG	O			0-1.2V VCO control output, DC measurement
117	gnd	AGND				
118	vdd25	VDD7				2.5V receiver supply, 80mA
119	gnd	AGND				
120	vdd33	VDD5				3.3V VCO supply, 80mA
121	vbuf60	NC	I			60GHz buffer amplifier current control, default open
122	vfine	ASIG	I			VCO fine tuning input (from pin115), no test program, but a needle contact is present, connect to vctl_vco pin
123	div_o	NC	O			divider output, 2GHz, leave it open in digital test
124	div_ob	NC	O			divider output bar, 2GHz, leave it open in digital test
125	gnd	AGND				
126	vdd25	VDD7				2.5V doubler supply, 30 mA

127	out	ASIG	O			multiplex output, DC measurement
128	outb	ASIG	O			multiplex output bar,DC measurement
129	vt_doubler	NC	I			doubler current control, default open
130	gnd	AGND				
131	gnd	AGND				
132	gnd	AGND				
133	gnd	AGND				
134	gnd	AGND				
135	gnd	AGND				
136	Rfout	NC	O			122GHz transmitter output, leave it open in digital test
137	gnd	AGND				
138	gnd	AGND				
139	gnd	AGND				
140	gnd	AGND				
141	gnd	AGND				
142	gnd	AGND				
143	gnd	AGND				
144	gnd	AGND				
145	gnd	AGND				
146	gnd	AGND				
147	gnd	AGND				
148	gnd	AGND				
149	gnd	AGND				

NC – not connected pin to the probe card

The architecture of the RF Frond-End with digital control as well as the BB processor architecture is provided in deliverable D4.1. Additionally, more details about the BB processor architecture are provided in deliverable D4.2. In comparison to that BB processor developed by Evatronix, a new module included in the BB processor (and consequently in the *Lighthouse* chip) is the GALS FMCW coprocessor.

2.2 GALS FMCW coprocessor

The GALS FMCW coprocessor has been designed and implemented on *Lighthouse* baseband processor, in parallel with the synchronous counterpart, to evaluate its advantages in terms of on-chip switching noise suppression. It contributes to (1) improving the performance of common-die analog/RF frond-end circuits in both time domain and frequency domain, and (2) facilitating the system-level integration of digital and analog/RF blocks.

The critical design issues, including system partitioning strategy, asynchronous interface design, timing analysis on key paths, are highlighted. The preliminary measurement results of SYNC/GALS FMCW processor (working on BIST mode) on the *Lighthouse* chip are presented as well.

2.2.1 GALS partitioning of the FMCW Coprocessor

The starting point of our work is a synchronous FMCW coprocessor. Its signal flow diagram at the top level is shown in Fig.3. Radix-4 butterfly structure is used as elemental block and 6 cascaded stages of Radix-4 FFT tiers are employed for processing each data frame of 4096 points. Two control modules for functional configuration and data pre/post-processing are applied.

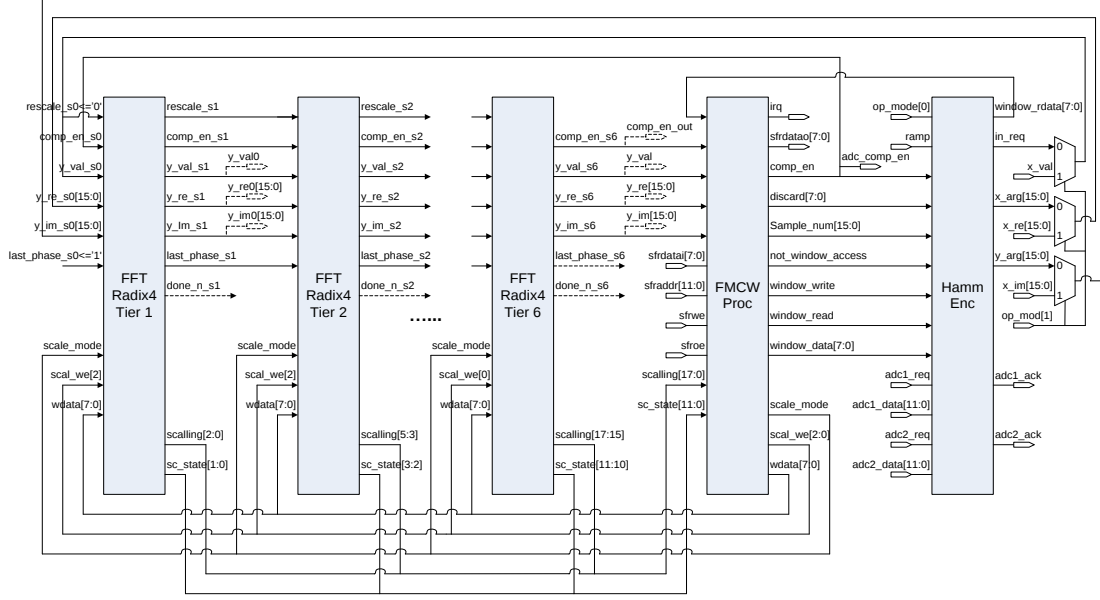


Figure 3. Signal Flow diagram of FMCW Coprocessor

The area, power and memory occupation of each functional module have been estimated according to the post-synthesis netlist using the IHP 130-nm CMOS process, as shown in Table 3. GALS partitioning scheme is further explored to balance the power consumption in each GALS clock domain (Fig.4).

Table 3. Lighthouse pin description

		FMCW RADAR							
Memory	Size	Radix4 T1	Radix4 T2	Radix4 T3	Radix4 T4	Radix4 T5	Radix4 T6	FMCW Proc	Hamm Enc
	Power	3X(1024X32)	3X(256X32)	3X(64X32)	3X(16X32)	0	0	2048X32	3X(256X12)
		7.4mW	6.7mW	6.5mW	6.6mW	0	0	3.4mW	3.3mW
Area		17.4%	15.7%	15.4%	15.5%	0	0	8%	7.9%
		0.56mm ²	0.27mm ²	0.21mm ²	0.19mm ²	0.10mm ²	0.04mm ²	0.31mm ²	0.12mm ²
		31.1%	15.0%	11.7%	10.6%	5.5%	2.2%	17.2%	6.7%
Power		9.4mW	7.7mW	7.6mW	8.6mW	1.9mW	0.5mW	3.4mW	3.3mW
		22.2%	18.2%	17.9%	20.3%	5.5%	1.1%	8.0%	7.8%
		100%	100%	100%	100%	100%	100%	100%	100%
		GALS FMCW RADAR							
Memory	Size	GALS B1	GALS B2	GALS B3	GALS B4	GALS B5			Total
	Power	3X(1024X32)	3X(256X32)	3X(64X32)	3X(16X32)	2048X32 + 3X(256X12)			15.0625KB
		7.4mW	6.7mW	6.5mW	6.6mW	6.7mW			33.9mW
Area		17.4%	15.7%	15.4%	15.5%	15.9%			80.0%
		0.56mm ²	0.27mm ²	0.21mm ²	0.19mm ²	0.57mm ²			1.80mm ²
		31.1%	15.0%	11.7%	10.6%	31.7%			100%
Power		9.4mW	7.7mW	7.6mW	8.6mW	9.1mW			42.4mW
		22.2%	18.2%	17.9%	20.3%	21.4%			100%
		100%	100%	100%	100%	100%			100%

The asynchronous communication in GALS FMCW design is achieved via three different types of data link: 2-stage DFF synchronizer, dual-clock FIFO, and pausable

clocking scheme. In particular, data transfer between GALS clock domains is done through double-flipflop timed by the handshake signals from pausable clock generators.

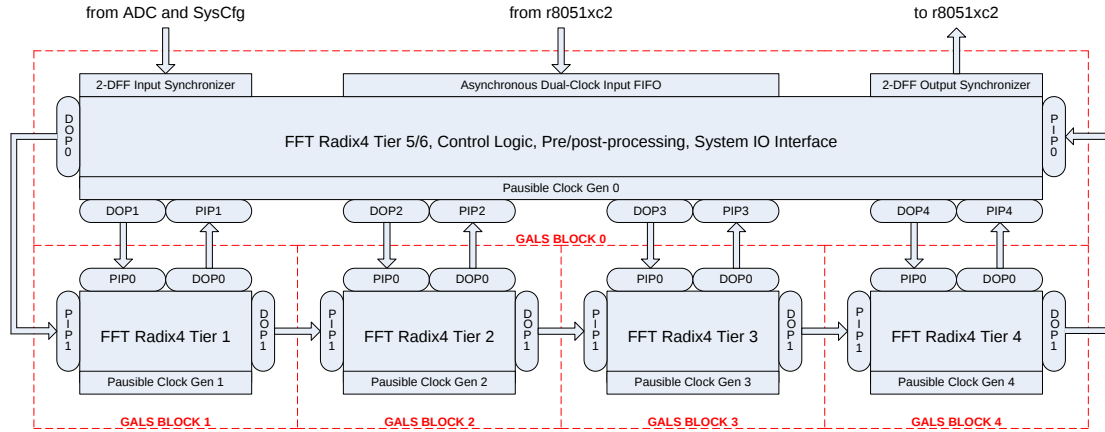


Figure 4. GALS partition scheme of FMCW coprocessor

2.2.2 Interface circuits design

In pausable clocking based GALS design, the arrival time of input data is fully asynchronous with regards to the RX local clock. A MUTEX is therefore applied as an arbiter in the clock generator to determine when the input data can be safely sampled by the RX clock. Two cascaded flipflops, which are triggered by the MUTEX output signals, are inserted on the data link. The fundamental scheme of pausable clocking based GALS data link, along with the input synchronization and IO flow control units, is shown in Fig.5.

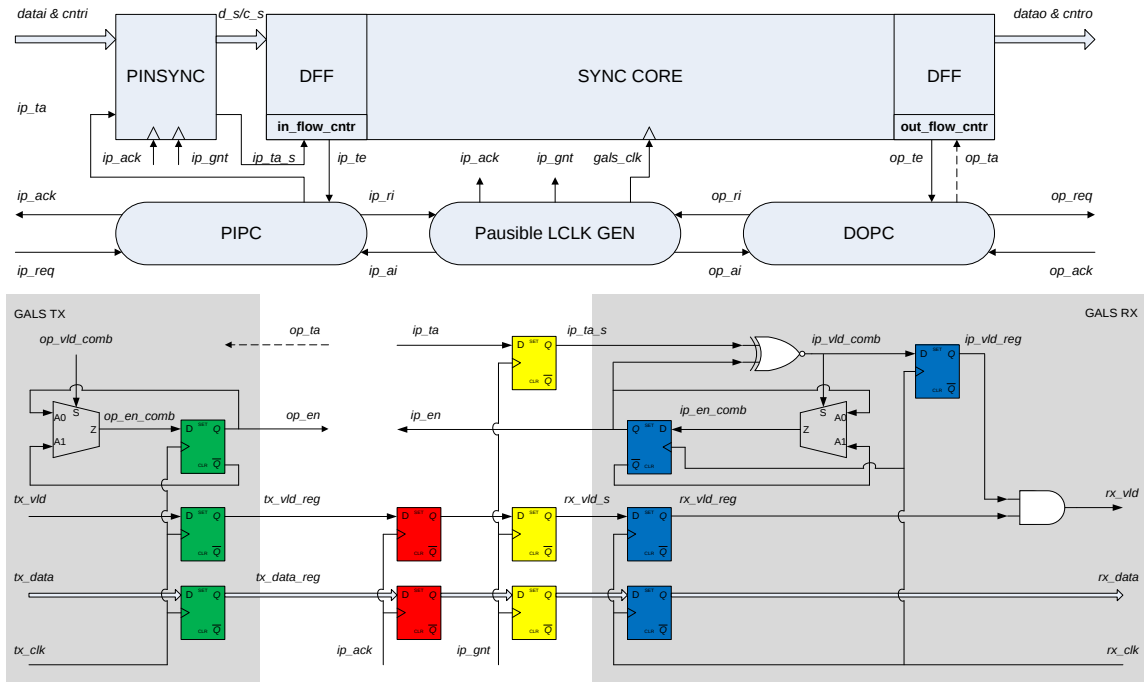
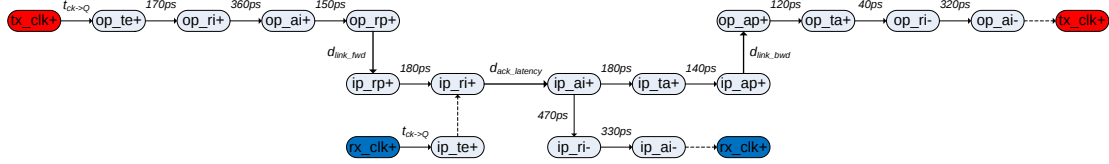


Figure 5. Pausible clocking based GALS data link

2.2.3 Timing Convergence on handshake signals

The asynchronous FSM of I/O port controllers are of importance for the performance of pausable clocking based GALS design. Following presents the delay corresponding to each signal transition of a DOP-to-PIP asynchronous channel at gate level synthesized by the IHP 130 nm process. Based on the back annotated propagation delays, timing analysis on critical paths can be performed as shown below.

$$\begin{aligned} d_{Link_fwd} &> 0; \\ d_{Link_bwd} &> 0; \\ d_{MUTEX} &< d_{ack_latency} < RAW + d_{MUTEX}. \end{aligned}$$



- I. **Clock stretching on TX.** For demand-type output port, the local clock on TX has to be paused for the whole communication. Consequently, TX clock will be stretched when the asynchronous handshaking loop delay exceeds the clock period, as shown below. The propagation delay of asynchronous FSM is negligible, and the RX clock acknowledged latency and I/O port interconnect delay dominate the handshaking loop delay.

$$T_{tx_clk} < d_{op_te+ \Rightarrow op_ai-} = d_{Link} + d_{ack_latency} + 1.6ns.$$

Clock stretching on RX. For poll-type input port, the RX local clock continues running until receiving an input port request from TX. RX clock is stretched only when ip_ri is high beyond RAW . More important, the stretching is tiny and deterministic, which can be ignored in practice.

$$RAW_{rx} < d_{ip_ri+ \Rightarrow ip_ri-}, Stretch_{rx} = d_{ip_ri+ \Rightarrow ip_ri-} - RAW = d_{ip_ai+ \Rightarrow ip_ri-} < 0.5ns.$$

- II. **Bundled-data protocol constraint.** As bundled-data protocol is applied in the data link, the input data must be valid on input port no later than being latched by the handshaking signals on the RX side. This is guaranteed by restricting the datapath interconnect delay between TX and RX less than the corresponding handshaking propagation delay. It leads to a constraint on the maximum acceptable datapath delay when taking the minimum d_{Link} and $d_{ack_latency}$ into account, as shown below. Indeed, it is a pretty loose constraint for the layout of asynchronous communication link.

$$d_{data} < d_{op_te+ \Rightarrow op_ai+} - t_{setup} = d_{Link_fwd} + d_{ack_latency} + 0.86ns - t_{setup}.$$

- III. **Setup/hold time constraints on ip_ta signal.** Among all the above handshaking signals, ip_ta needs particular attention in timing analysis since it is the only signal synchronized by ip_gi+ (in double-FF mutually exclusive mechanism) on the RX side. For setup time analysis, the worst case happens when ip_ri rises simultaneously with $rclk$ and is granted by the MUTEX first. Under this circumstance ip_gi+ happens immediately after ip_ai- . Hence the setup timing constraint on ip_ta can be derived as follows. According to above timing arcs, it's easy to be guaranteed. Hold time violation happens on ip_ta if it changes too close after ip_gi+ . The minimum interval from ip_gi+ to $ip_ta+/-$ occurs when ip_ri+ rises with $rclk+$ and the MUTEX responds to ip_ri after consuming all the resolution time associated with the target MTBF. In that case, there is still an interval between ip_gi+ and rx_clk+ which is reserved (by adjusting RAW) to accommodate the combinational logic to update ip_te , and this interval itself is actually larger than the hold time. As a result, it can be concluded that the

minimum delay from ip_gi+ (ip_ai-) to $ip_ta+/-$ is always much larger than t_{hold} , and therefore there is no hold timing constraint on ip_ta .

$$d_{ip_ai+ \Rightarrow ip_ta+} < d_{ip_ai+ \Rightarrow ip_ai-} - t_{setup}$$

- IV. **RAW specification.** The request acknowledged window on RX clock is critical for the timing as well as performance analysis of pausable clocking based GALS design. An optimal RAW should cover (1) the resolution time of MUTEX at target MTBF so as to avoid unnecessary clock stretching on RX; (2) in this situation clock stretching happens if all the resolution time is consumed by MUTEX and the stretching duration is predictable to be $d_{ip_ai+ \Rightarrow ip_ai-}$, i.e., the active-phase of ip_ai ; (3) to accommodate the combinational logic on updating ip_te , additional duration of RAW is required to meet $d_{ip_gi+} + d_{comb_ip_te} < d_{ip_gi+ \Rightarrow rx_clk+} - t_{setup}$; (4) as a result, the optimal RAW is slightly larger than the MUTEX resolution time, but the clock stretching remains to be the active-phase of ip_ai .

2.2.4 Working mode configuration

Static working modes configuration is supported. A 32-bit register, which is programmed via the JTAG interface, is reserved to set the working modes of 4096-point GALS FFT processor. The configuration bit assignment and corresponding mode selection are pre-defined as shown in Table 4.

An external reset $nRST$ is applied as a global signal to activate 4096-point GALS FFT processor. As the distribution of local clocks is crucial for the analysis and evaluation of low-noise GALS design, 5 probe pads are reserved for the measurement of local clocks. Furthermore, Clock Working Mode is in particular defined with $nRST=1$ and $FEN=0$, where only the local clock generators get enabled while all the functional modules are kept in reset. BIST mode is also integrated in the GALS design, which supports continuously functional testing with internally generated pseudo-random data.

Table 4. Configuration bit assignment and mode selection

31	30	29	28	27:22			21:20	19:18	17:12		11:10	09:06	05:00		
FEN	BIST	PEN	TCK	D8			D7	D6/D5	D4		D3	D2		D1	
63	62	61:58			57:32										
NV		G1/S0	OP_MODE											RESERVED	
BIT		Polarity		Definitions											
BIST		HIGH		Built-in self-test enable (otherwise data valid from ADC)											
FEN		HIGH		Functional modules enable (otherwise only local clocks get enabled)											
PEN		HIGH		Local clock interleaving enable (adaptive phase detection and compensation)											
TCK		HIGH		Frequency/2 on output testing clock enable (otherwise output local clock directly for test)											
nRST		FEN		BIST	PEN	TCK	Testing Mode Selection								
0		X		X	X	X	IDLE								
1		0		X	X	X	Clock working mode								
1		1		1	X	X	BIST working mode								
1		1		0	X	X	Normal working mode								

2.3 DFT in BB processor

In order to provide a high level of testability of the BB processor embedded in the Lighthouse chip, two DFT approaches have been implemented: scan-chain test and BIST test of both synchronous and GALS FMCW coprocessors.

2.3.1 Scan Test

The scan-chain test has been implemented using Synopsys DFT Compiler tool whereas the test patterns have been created using TetraMAX ATPG tool. After design synthesis using Synopsys Design Compiler, a mapped netlist has been created. This netlist has been used to create a fully optimized design with internal scan circuitry. A typical design flow which we have followed to implement scan test is shown in Fig. 6. At the end of the design flow, TetraMAX ATPG produces a set of high fault-coverage test vectors that can be readily adapted to a tester.

Taking into account the BB processor architecture, we have implemented five scan-chains. Two scan-chains are driven by the system clock (pin '*clk*'). The other three scan-chains are driven by clocks '*adc_clk*', '*msck*' and '*tck*', respectively. The complete synchronous part has been included into the scan test. On the other hand, we have decided to exclude the GALS FMCW coprocessor from the scan test. The reason is driven by the fact that we cannot control the internal GALS clocks by the system clock '*clk*'. In order to keep the number of I/O pins as low as possible, we have not introduced additional pins for the scan-in and scan-out signals. We have multiplexed those signals with some of existing functional I/O pins. The scan-chain parameters are summarized in Table 5.

Table 5. Scan-chain parameters

Scan-Chain	Clock	Length (number of scan cell)	Scan-In	Scan-Out
1	clk	2514	adc_sdo	op_state[0]
2	clk	2513	auxdac_dout	auxdac_din
3	adc_clk	56	pll_clock	op_state[1]
4	tck	208	eeeprom_miso	op_state[2]
5	msck	57	mcs	op_state[3]

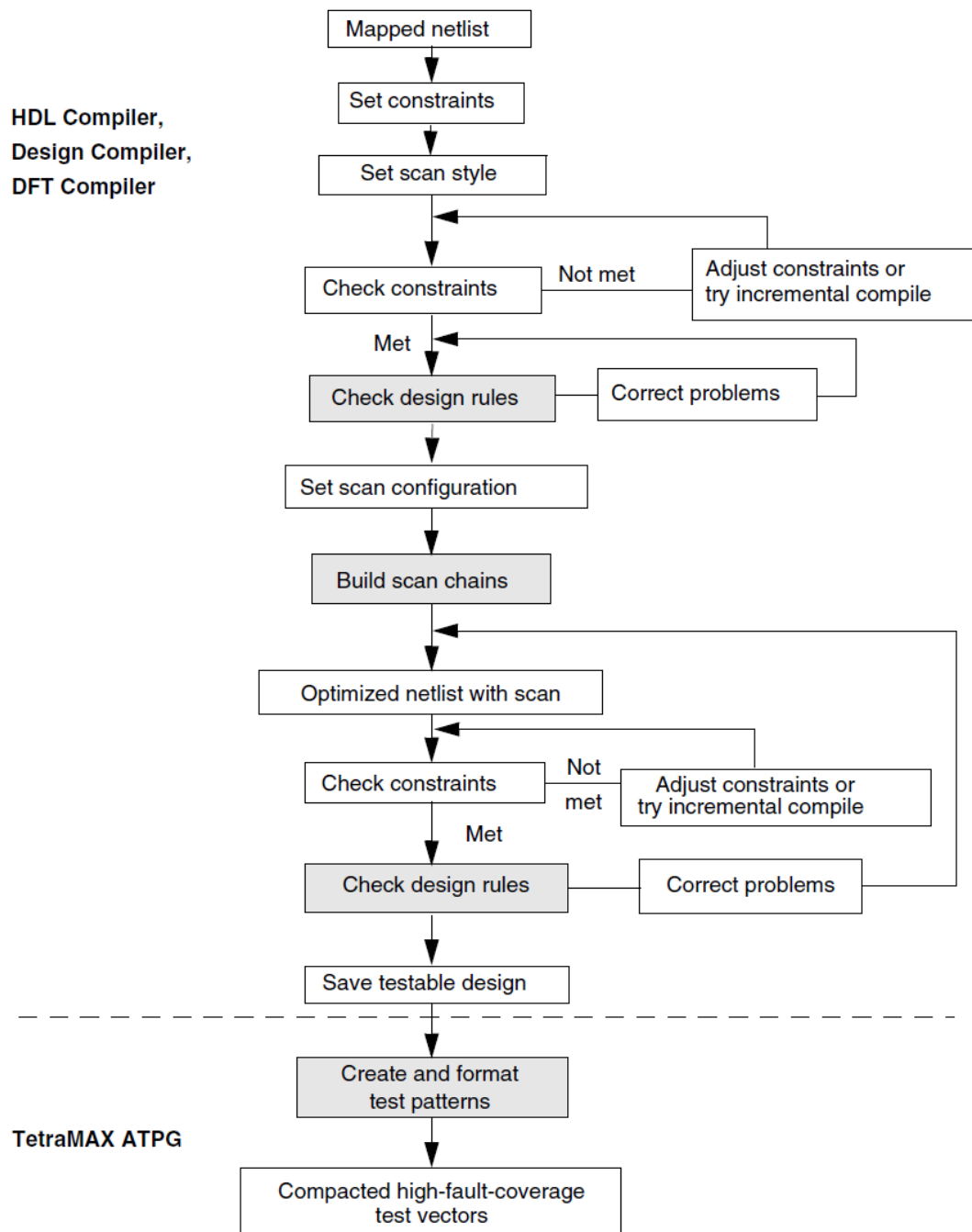


Figure 6. Typical Scan Synthesis Flow from a mapped design

2.3.2 BIST Test

As we already have mentioned, the FMCW coprocessor is by far the most complex component of the BB processor (almost 80 % cell area of the BB processor). Therefore, we have decided to implement BIST for this component in both synchronous and GALS version. The BIST concept of FMCW coprocessor is illustrated in Fig. 7.

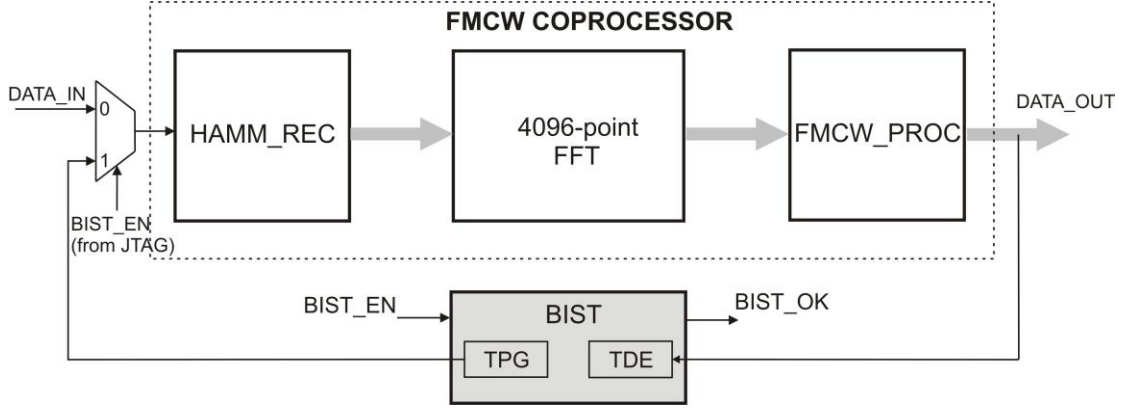


Figure 7. BIST concept of the FMCW coprocessor

The BIST function can be activated over the JTAG interface. A test pattern generator (TPG) consisting of a linear feedback shift register (LFSR) generates the test input data. Similarly, a test data evaluator (TDE) checks the output test data. The TDE consists of a test response compression circuit and a comparator. The test response compression block is based on signature analysis and, accordingly, incorporates one LFSR in its structure. The presence of repetitive pulses at *BIST_OK* output indicates the success of the test. Every pulse corresponds to one FFT frame correctly processed by the FMCW coprocessor.

3. Testing Environment Setup and Preliminary Test Results

3.1 Testing Environment Setup

So far, the Lighthouse tests have been conducted on Advantest 9300 SOC system. We have firstly decided to test the chip at the wafer level. In order to that testing, a probe card is required. Therefore, we have ordered and got the probe card produced by an external company. In the next step, the chip will be packaged and again tested. The photos of Advantest 9300 SOC test environment and the probe card are shown in Fig. 8 and Fig. 9, respectively.

The Advantest 93000 SOC is a high performance production test system. We have a digital-dominant configuration with licensed speeds up to 800MB/s. The hardware is capable of up to 3.6GB/s per channel. The test system provides a set of commonly used standard test functions such as functional test, current measurements, sweep tests

etc. Low level programming for user/device specific requirements is available through a rich C++ API as well as direct firmware access.



Figure 8. Advantest 9300 SOC system

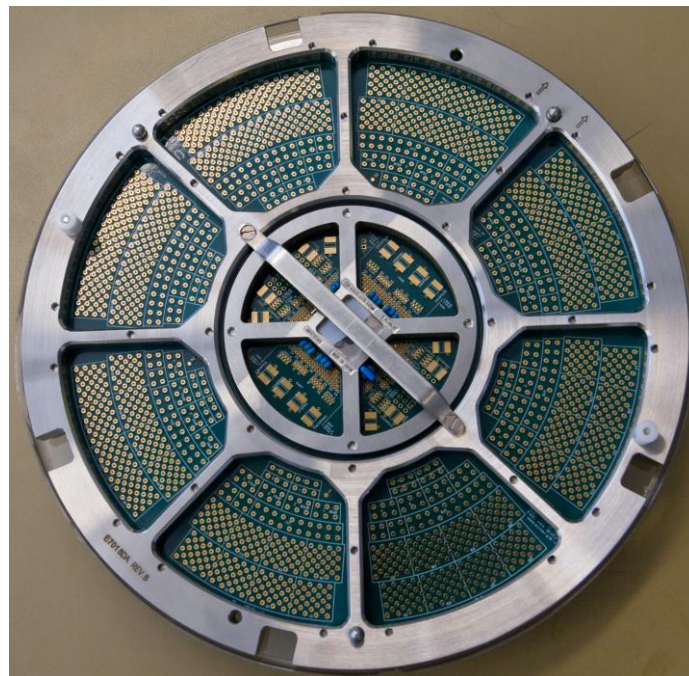


Figure 9. Probe card

3.2 Preliminary Test Results

So far, the test flow of Lighthouse chip is structured as follows:

- Continuity: Parallel and serial pin continuity tests to check for proper bonding
- Scan Test
- BIST GALS : Test of the GALS FMCW coprocessor
- BIST SYNC : Test of the SYNC FMCW coprocessor
- BIST of Digital Control
- SPI Test of Digital Control
- D/A Convertor Test

The analysis tool is able to generate the waveforms and to find error locations. One example of such waveform for the SPI test is shown in Figure 10.

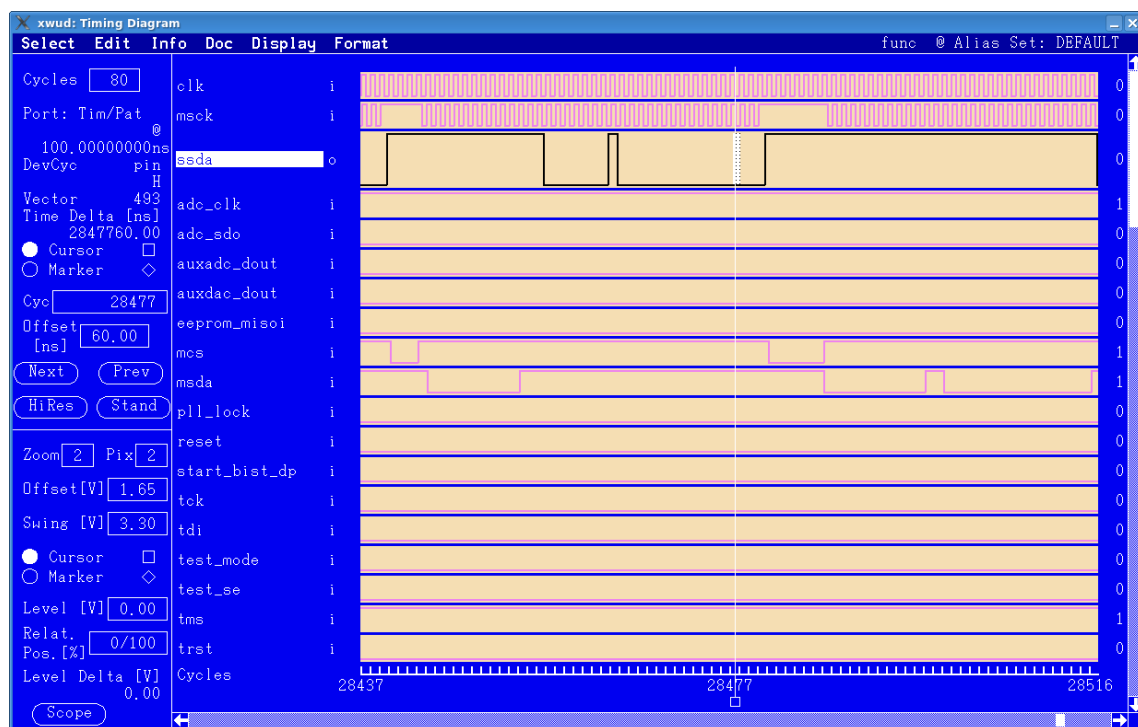


Figure 10. Timing waveform – SPI test

The BIST test of FMCW coprocessor in the both GALS and synchronous version has passed successfully.

For the single pulse pattern on BIST_OK signal, we detected the rising edge and falling edge which are slightly earlier than the simulation. For the multiple pulses pattern, we further detected the continuous pulses on the chip. A test report automatically generated by the test machine is shown below. As can be seen, three pulses (six transitions) on BIST_OK were detected by the tester, with the exact begin/end cycle time and pulse width. The variation in pulse width also indicated the drifting in working frequency due to asynchronous design. Also, we noticed that the working speed of GALS design seems higher than the synchronous one. Again, we would like to stress that the FMCW coprocessor is by far the most complex block of BB processor.

```

BIST_GALS_capt
Pulse 1:
  StartCycle = 134332 @ (3.26049e+06ns - 3.26051e+06ns)
  PulseWidth = 66991 Cycles
  EndCycle   = 201322 @ (4.88646e+06ns - 4.88648e+06ns )

Pulse 2:
  StartCycle = 268312 @ (6.51243e+06ns - 6.51245e+06ns)
  PulseWidth = 66986 Cycles
  EndCycle   = 335297 @ (8.13828e+06ns - 8.1383e+06ns )

Pulse 3:
  StartCycle = 402288 @ (9.76427e+06ns - 9.7643e+06ns)
  PulseWidth = 66993 Cycles
  EndCycle   = 469280 @ (1.13903e+07ns - 1.13903e+07ns )

```

The scan test passed successfully for the four scan-chains. Only one scan test for the chain no. 2 (see Table 5) has failed. With respect to this scan-chain, we have discovered that the probe card has a short to VDD for input pin ‘auxdac_dout’. This pin is the scan-in pin for scan-chain no. 2. Therefore, we believe that this defect is a cause of scan-chain failure.

Unfortunately, we have found an error (a short for clock signal) in the layout of digital control component. On the other hand, we have found that the host-interface of BB processor works properly, which is essential for programming the SPI registers of digital control. However, due to the short in digital control, we have not been able to successfully test the digital control of RF Front-End and the D/A convertor. This short can be fixed by changing one metal mask, which is work in progress.

4. Conclusion and Further Work

The mm-wave SoC radar chip (‘Lighthouse’ chip) has been designed, fabricated and tested on wafer. We have confirmed the correct operation of FMCW coprocessor in both synchronous and GALS mode. This is already significant results, which should enable EMI measurements. We can expect some interesting results on the noise suppression by the GALS design, which will be addressed in the future package tests.

In order to fix defect (a short) in the probe card, it has been sent for repair. We expect that the probe card will be again available very soon for further testing. Additionally, when the short in the chip is removed by a metal fix, we will package the chips and do the further tests. Therefore, new test results will be available in the new release of this document.