



拥抱变革：基于模型的通信系统设计实践

中国空间技术研究院西安分院

主讲人：许鹏飞



CAST Xian

创人类航天文明 铸民族科技丰碑



承载梦想
铸就辉煌



汇报内容

1

基本情况介绍

2

传统FPGA开发流程

3

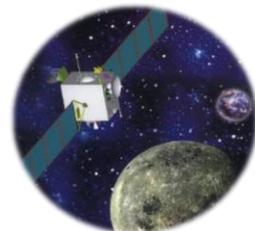
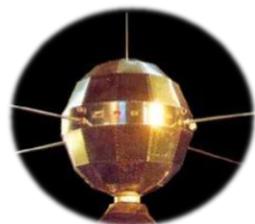
MATLAB解决方案

4

MATLAB应用实效

5

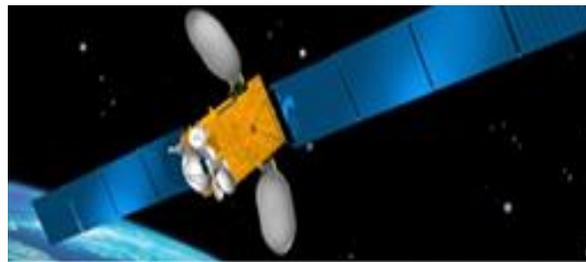
经验与总结



一、基本情况介绍

1 中国空间技术研究院西安分院简介

我国从事卫星有效载荷设计和空间电子设备研制的专业研究单位。



2 通信技术研究所简介

主要从事卫星通信系统论证设计及其核心产品研制。

3 演讲人

许鹏飞，高级工程师，主要研究方向为卫星通信等。





汇报内容

1

基本情况介绍

2

传统FPGA开发流程

3

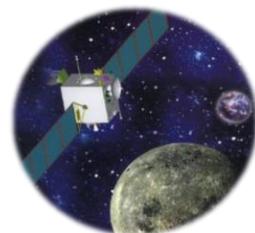
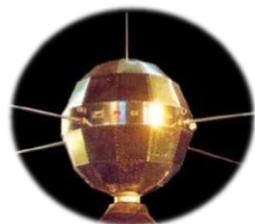
MATLAB解决方案

4

MATLAB应用实效

5

经验与总结



二、传统FPGA开发流程

1 传统开发流程



1.需求和技术指标

文本文档

```
1 % Add reference to the API wrapper.
2 apiroot = 'C:\Program Files\Audio Precision\APx500 2.4\api\';
3 NET.addAssembly([apiroot 'AudioPrecision.API2.dll'])
4
5 % Create an instance of the application, and make it visible (it doesn't
6 % have to be visible to work, it just lets us see what's happening).
7 apx = AudioPrecision.API.APx500_Application;
8 apx.Visible = true;
9
10 % Turn off the signal monitors.
11 apx.SignalMonitorsEnabled = false;
12
13 % Select the Frequency Response measurement.
14 apx.ShowMeasurement('Signal Path1', 'Frequency Response')
15
16 % Set the generator parameters. Matlab can only set a property on an
17 % explicit handle.
18 gen = apx.FrequencyResponse.Generator;
19 startFreq = gen.StartFrequency;
20 startFreq.Unit = 'Hz';
21 startFreq.Value = 30;
22 stopFreq = gen.StopFrequency;
23 stopFreq.Value = 18000;
24 levels = gen.AnalogLevels;
25 levels.Unit = 'Vrms';
26 levels.SetValue(AudioPrecision.API.OutputChannelIndex.Ch1, 1.5);
27
28 % Start the measurement. Success is true if the measurement worked.
29 success = apx.FrequencyResponse.Start;
```

2.设计

算法级系统设计

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY TestBench IS
END TestBench;
ARCHITECTURE TestBenchArch OF TestBench IS
COMPONENT UnitUnderTest IS
PORT (a, b: IN std_logic;
      c: OUT std_logic);
END COMPONENT;
SIGNAL x, y, z: std_logic;
BEGIN
CompToTest: UnitUnderTest PORT MAP (x, y, z);
PROCESS
BEGIN
x <= '0'; y <= '1';
WAIT FOR 5 ns;
x <= '1';
WAIT FOR 25 ns;
x <= '0'; y <= '0';
WAIT FOR 10 ns;
x <= '1';
WAIT FOR 20 ns;
x <= '0'; y <= '1';
WAIT FOR 5 ns;
END PROCESS;
END TestBenchArch;
```

3.实现

手工HDL编码



4.测试和验证

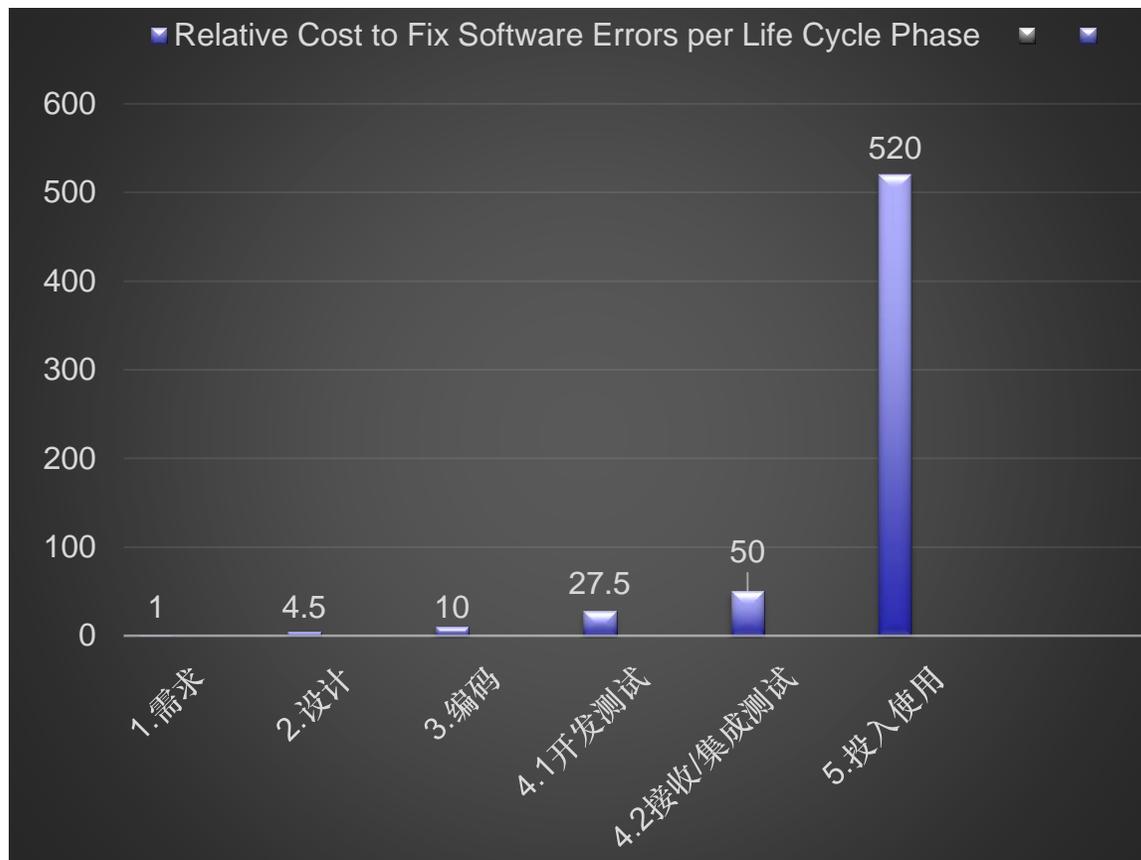
传统测试



二、传统FPGA开发流程

2 纠错成本高

Error Cost Escalation Through the Project Life Cycle – NASA



二、传统FPGA开发流程

3 开发效率低

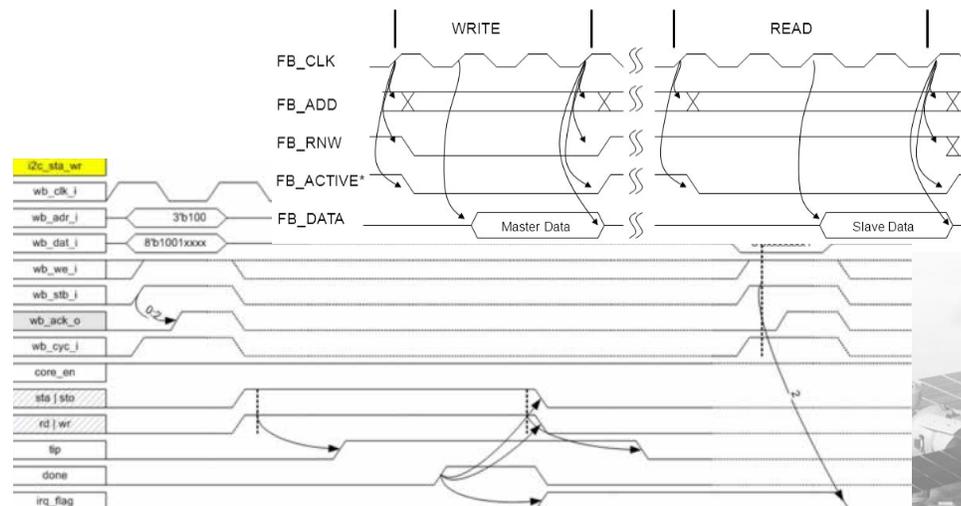
➤ 设计抽象级低



➤ 迭代更新慢

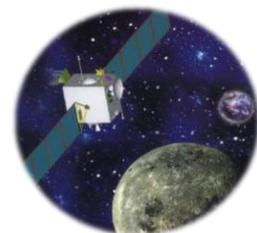
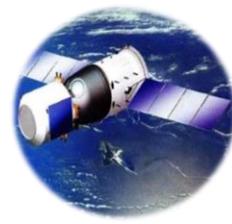
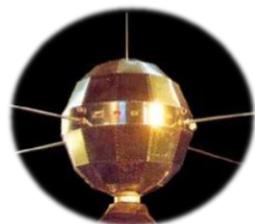


Please wait



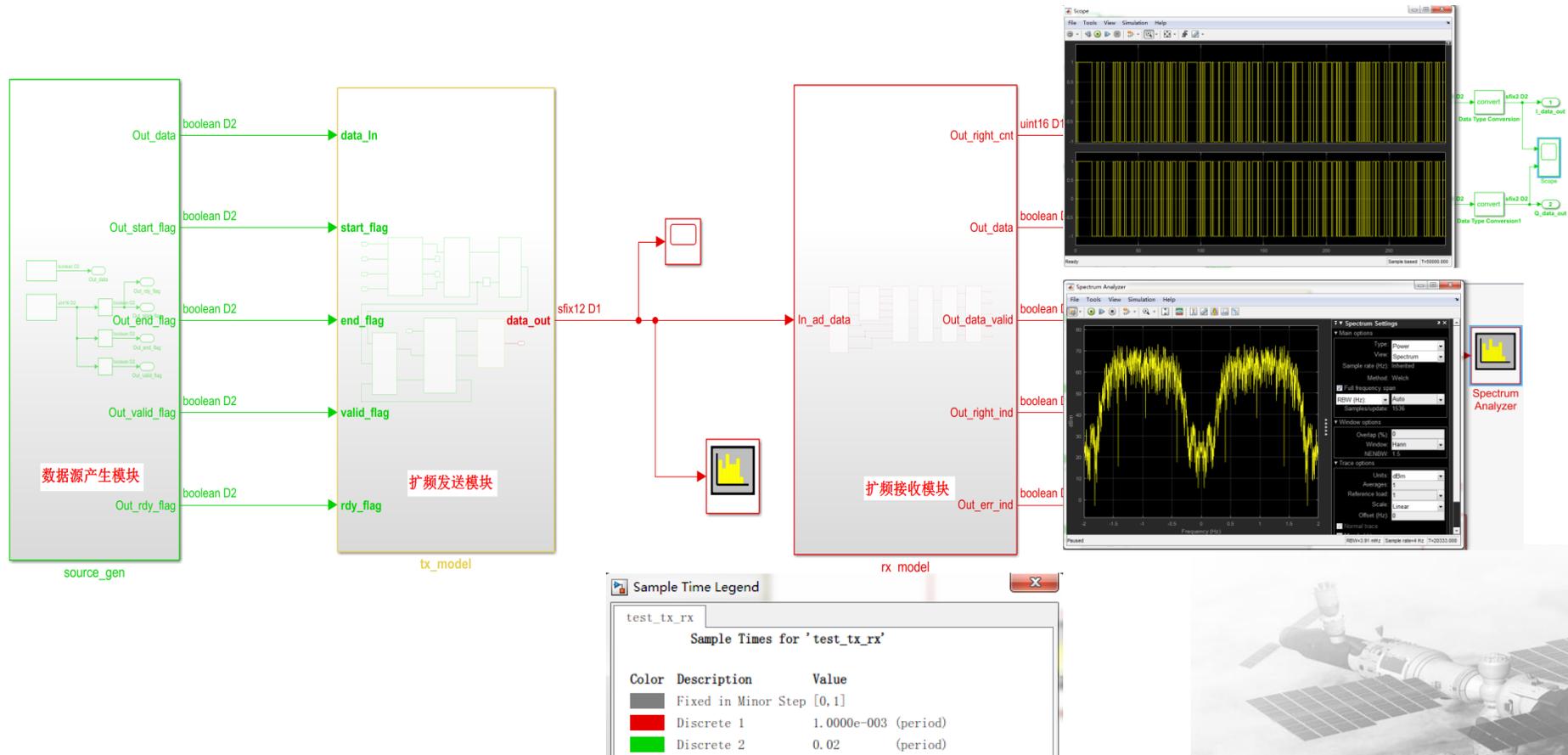
汇报内容

- 1 基本情况介绍
- 2 传统FPGA开发流程
- 3 **MATLAB解决方案**
- 4 MATLAB应用实效
- 5 经验与总结



三、MATLAB解决方案

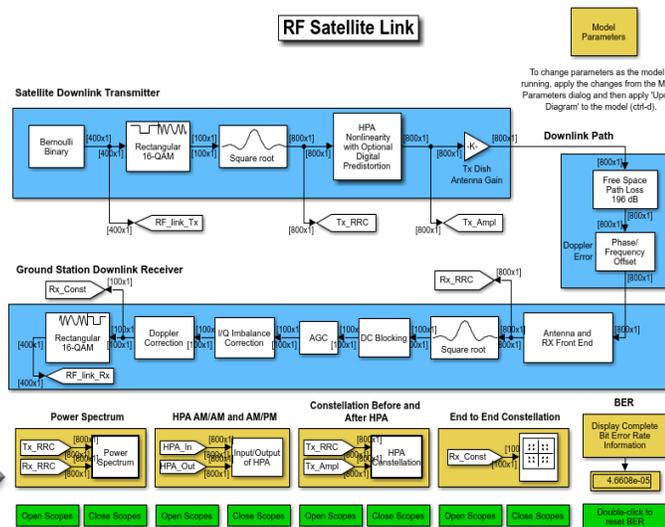
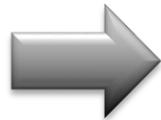
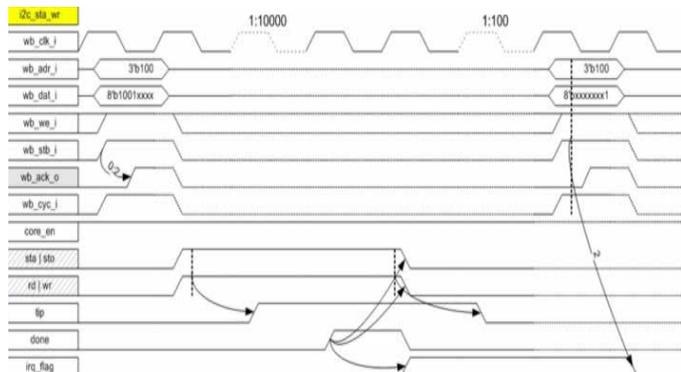
1 需求分析



三、MATLAB解决方案

2 设计

➤ 抽象级提高



✓ 接口简化;
✓ 时序简化;

➤ 建模对象: 门电路比特级信号

定点数、复数、向量、矩阵

➤ 行为: 加减乘, 与或非;
同/异步时序;

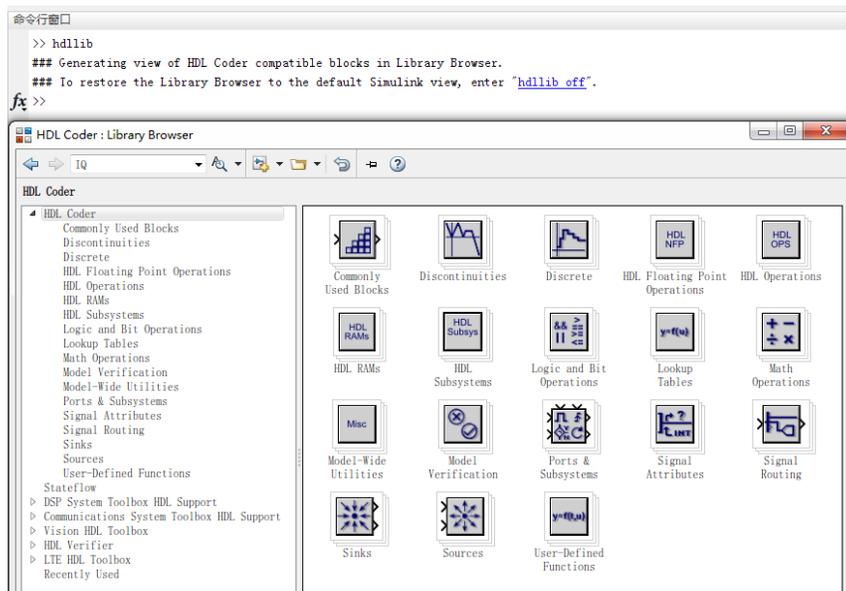
算法级行为



三、MATLAB解决方案

2 设计

➤ 模块库支持



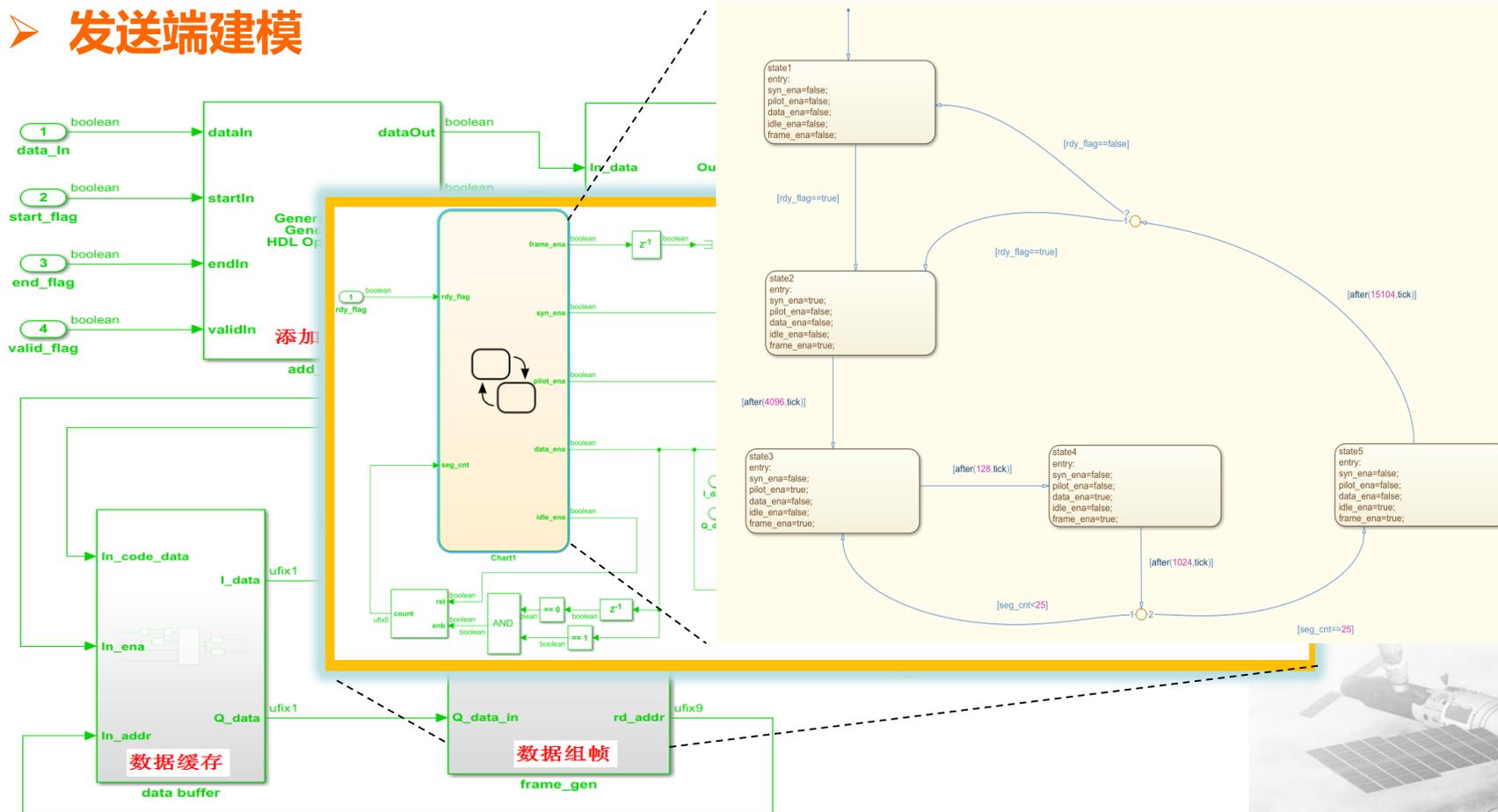
| 序号 | 模块类型 | 模块 |
|----|--------------|--|
| 1 | 存储模块 | Dual Port RAM、Dual Rate Dual Port RAM、FIFO、Simple Dual Port RAM、Simple Port RAM、Lookup Table (ROM) |
| 2 | 三角函数模块 | Sine/Cosine、NCO |
| 3 | 滤波模块 | Biquad Filter、Channelizer HDL Optimized、CIC Decimation/Interpolation、FIR filter、FIR Decimation/Interpolation、LMS Filter、Raised Cosine Receive/ Transmit Filter |
| 4 | FFT | FFT/IFFT |
| 5 | 编译码模块 | RS编译码、卷积码编译码、CRC校验 |
| 6 | 交织模块 | Convolutional Interleaver/Deinterleaver、General Multiplexed Interleaver/Deinterleaver |
| 7 | 调制模块 | BPSK、QPSK、M-PSK Modulator/Demodulator、Rectangular QAM Modulator/ Demodulator |
| 8 | 支持LTE标准的相关模块 | Convolutional Interleaver/Deinterleaver、CRC Encoder/Decoder、Turbo Encoder/Decoder |



三、MATLAB解决方案

2 设计

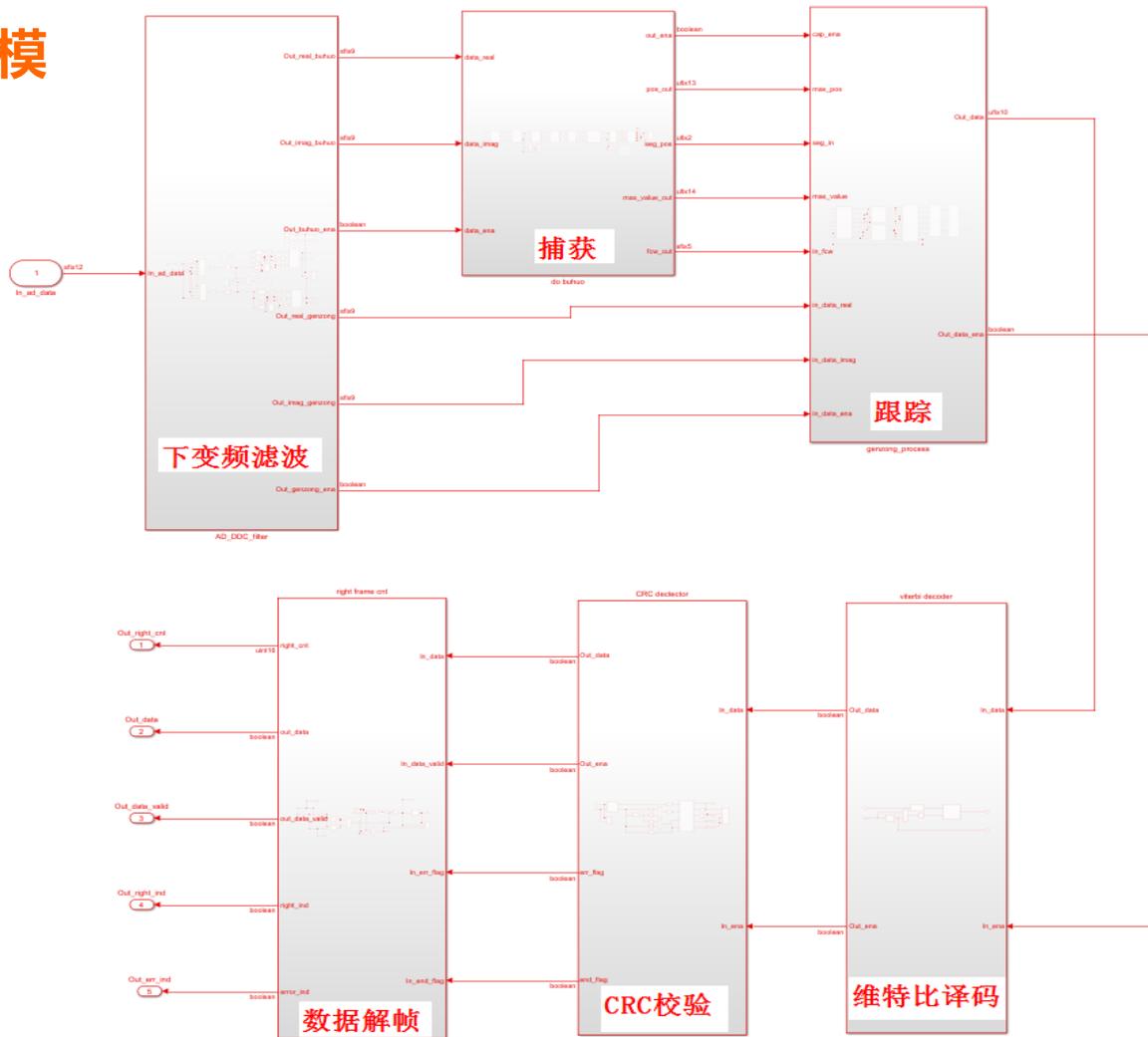
➤ 发送端建模



三、MATLAB解决方案

2 设计

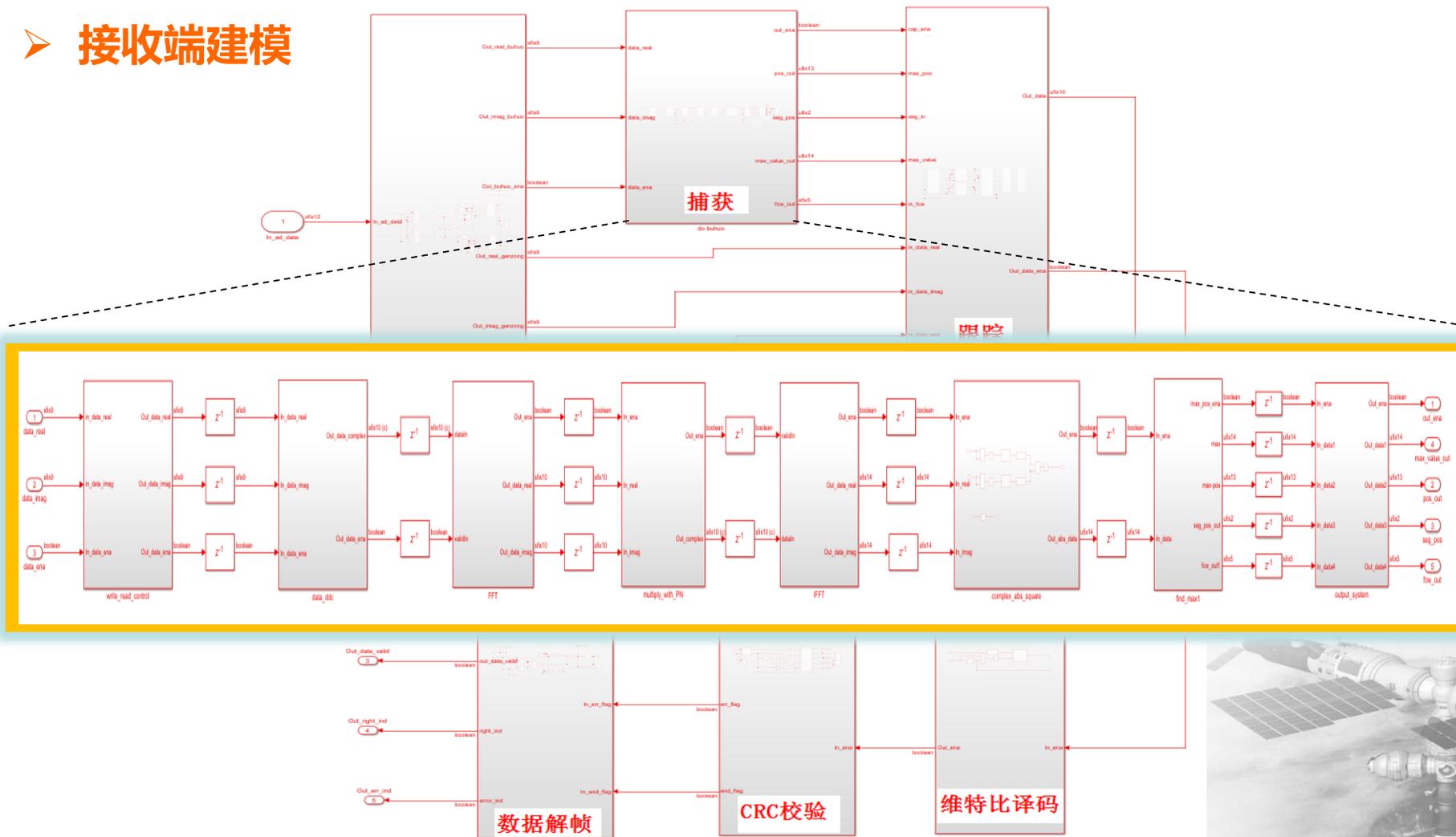
➤ 接收端建模



三、MATLAB解决方案

2 设计

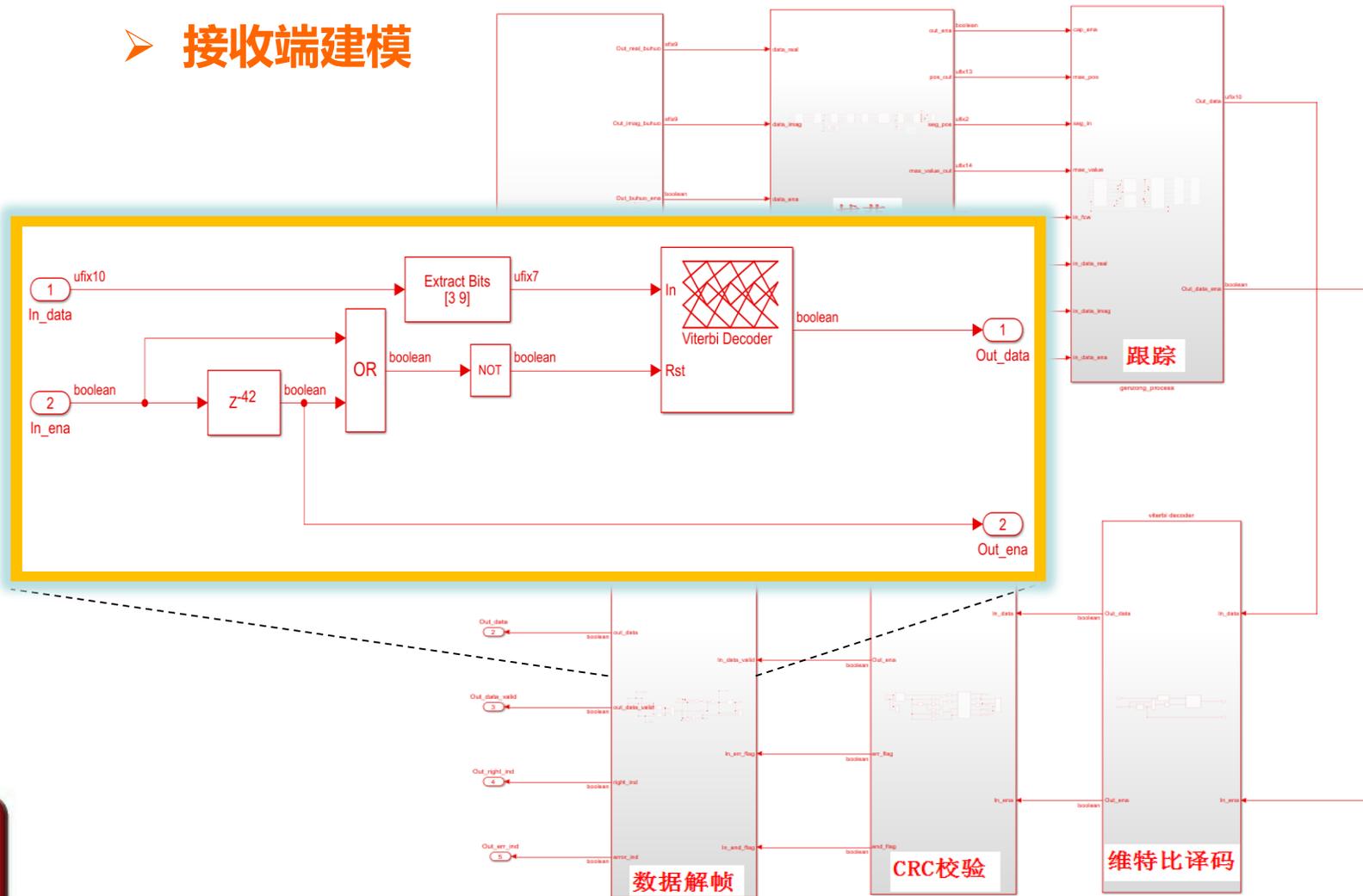
➤ 接收端建模



三、MATLAB解决方案

2 设计

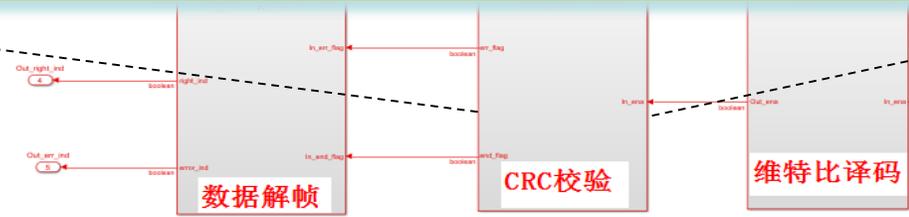
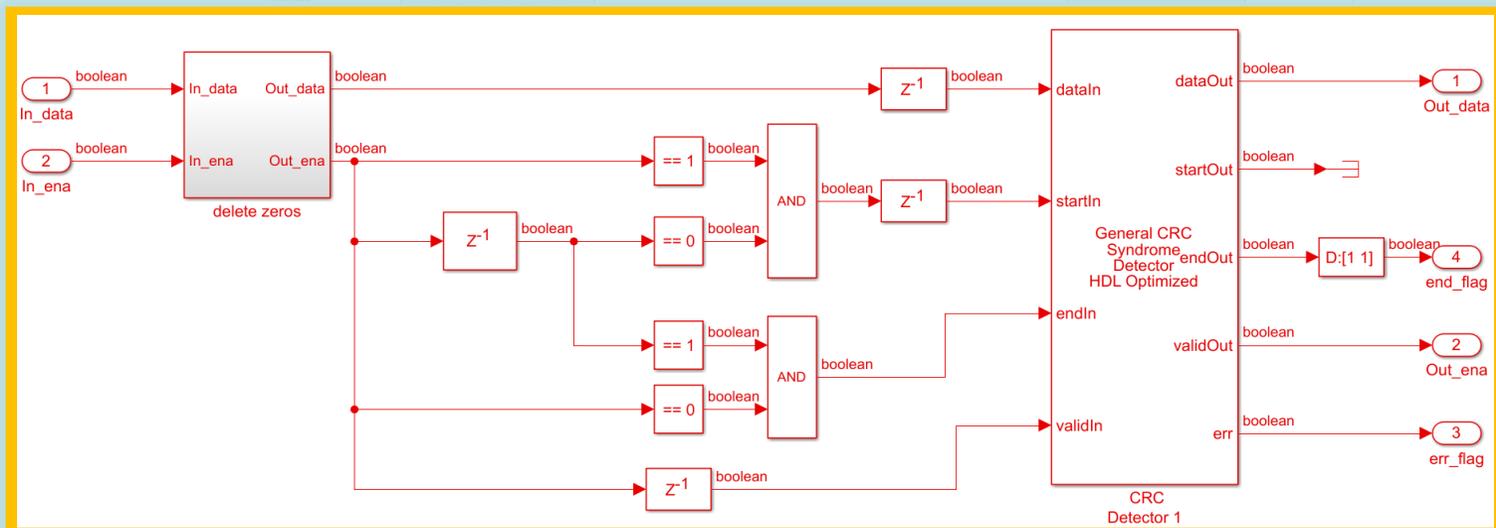
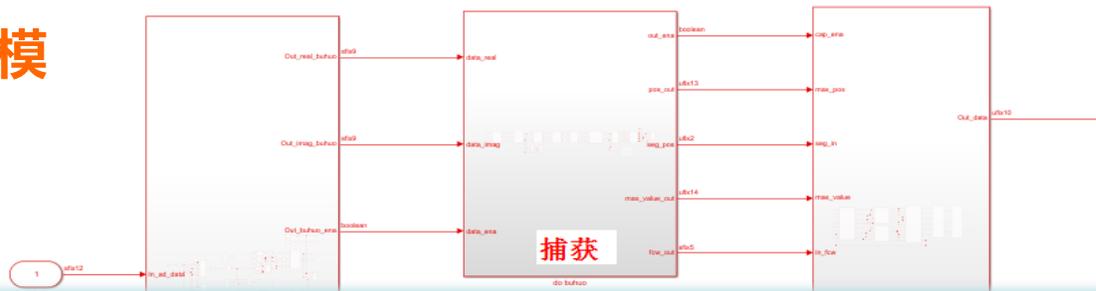
接收端建模



三、MATLAB解决方案

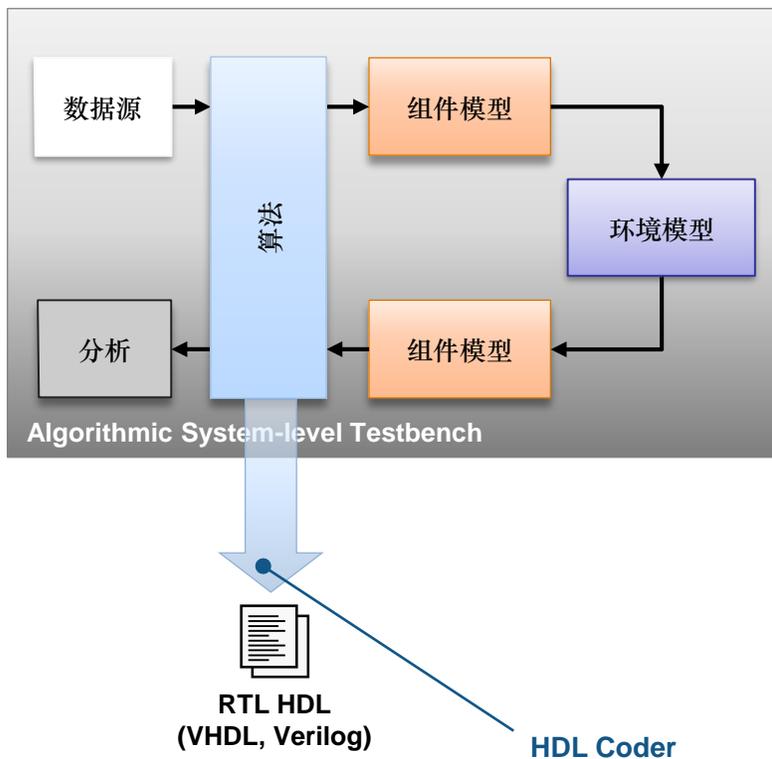
2 设计

➤ 接收端建模



三、MATLAB解决方案

3 实现



✓ 时序自动设计;

HDL Coder

RTL HDL (VHDL, Verilog)

✓ 快速迭代;

1需求 2设计
3实现 4测试

The diagram shows a 4-step iterative cycle: 1需求 (Requirements), 2设计 (Design), 3实现 (Implementation), and 4测试 (Testing). Arrows indicate a clockwise flow between these steps.



三、MATLAB解决方案

3 实现 (自动生成代码)

The screenshot displays the Simulink environment with two windows open:

- Code Generation Window (Left):** Shows the generated Verilog code for the `rx_model` module. The code includes library declarations, port definitions, and component instantiations.
- RTL Hierarchy Window (Right):** Shows the hierarchical structure of the Verilog RTL model, listing various sub-modules and their instantiations.

```
--- Module: rx_model
--- Source Path: test_tx_rx/rx_model
--- Hierarchy Level: 0
---
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;
USE work.rx_model_pkg.ALL;

ENTITY rx_model IS
    PORT ( clk           : IN    std_logic;
          reset         : IN    std_logic;
          clk_enable    : IN    std_logic;
          In_ad_data    : IN    std_logic_vector(11 DOWNTO 0); -- sfix12
          ce_out        : OUI   std_logic;
          Out_right_cnt : OUI   std_logic_vector(15 DOWNTO 0); -- uint16
          Out_data      : OUI   std_logic;
          Out_data_valid : OUI   std_logic;
          Out_right_ind : OUI   std_logic;
          Out_err_ind   : OUI   std_logic
        );
END rx_model;

ARCHITECTURE rtl OF rx_model IS
    -- Component Declarations
    COMPONENT AD_DDC_filter
        PORT ( clk           : IN    std_logic;
              reset         : IN    std_logic;
              enb          : IN    std_logic;
              In_ad_data    : IN    std_logic_vector(11 DOWNTO 0); -- sfix12
              Out_real_buhuo : OUI   std_logic_vector(8 DOWNTO 0); -- sfix9
              Out_imag_buhuo : OUI   std_logic_vector(8 DOWNTO 0); -- sfix9
              Out_buhuo_ena  : OUI   std_logic;
              Out_real_genzong : OUI   std_logic_vector(8 DOWNTO 0); -- sfix9
              Out_imag_genzong : OUI   std_logic_vector(8 DOWNTO 0); -- sfix9
              Out_genzong_ena : OUI   std_logic
            );
    END COMPONENT;
```

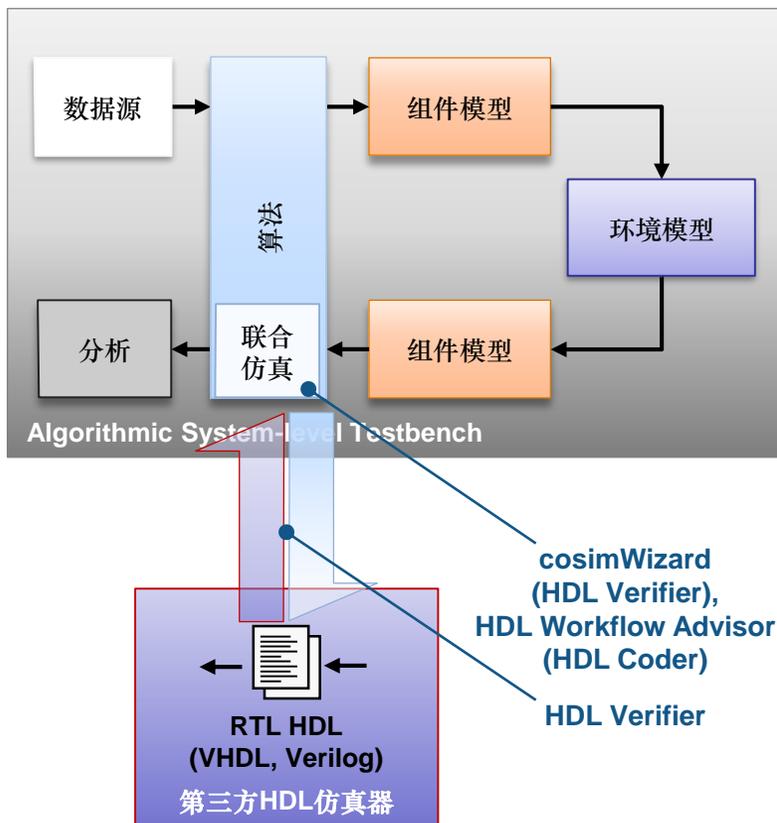
The RTL Hierarchy window shows the following structure:

- `rx_model_ise`
 - `xc5vfx130t-3ff1738`
 - `rx_model - rtl (rx_model.vhd)`
 - `u_AD_DDC_filter - AD_DDC_filter - rtl (AD_DDC_filter.vhd)`
 - `u_do_buhuo - do_buhuo - rtl (do_buhuo.vhd)`
 - `u_write_read_control - write_read_control - rtl (write_read_control.vhd)`
 - `u_data_ddc - data_ddc - rtl (data_ddc.vhd)`
 - `u_FFT - FFT - rtl (FFT.vhd)`
 - `u_multiply_with_PN - multiply_with_PN - rtl (multiply_with_PN.vhd)`
 - `u_IFFT - IFFT - rtl (IFFT.vhd)`
 - `u_complex_abs_square - complex_abs_square - rtl (complex_abs_square.vhd)`
 - `u_find_max1 - find_max1 - rtl (find_max1.vhd)`
 - `u_output_system - output_system_block - rtl (output_system_block.vhd)`
 - `u_genzong_process - genzong_process - rtl (genzong_process.vhd)`
 - `u_do_genzong2 - do_genzong2 - rtl (do_genzong2.vhd)`
 - `u_buhuo_to_genzong - buhuo_to_genzong - rtl (buhuo_to_genzong.vhd)`
 - `u_do_genzong1 - do_genzong1 - rtl (do_genzong1.vhd)`
 - `u_genzong_decision - genzong_decision - rtl (genzong_decision.vhd)`
 - `u_phase_correct - phase_correct - rtl (phase_correct.vhd)`
 - `u_data_get - data_get - rtl (data_get.vhd)`
 - `u_viterbi_decoder - viterbi_decoder_block - rtl (viterbi_decoder_block.vhd)`
 - `u_Viterbi_Decoder - Viterbi_Decoder - rtl (Viterbi_Decoder.vhd)`
 - `u_CRC_detector - CRC_detector - rtl (CRC_detector.vhd)`
 - `u_delete_zeros - delete_zeros - rtl (delete_zeros.vhd)`
 - `u_CRC_Detector_1 - alphaCRC_Detector_1 - rtl (alphaCRC_Detector_1.vhd)`
 - `u_right_frame_cnt - right_frame_cnt - rtl (right_frame_cnt.vhd)`
 - `clock_constraint.ucf`



三、MATLAB解决方案

4 测试与验证

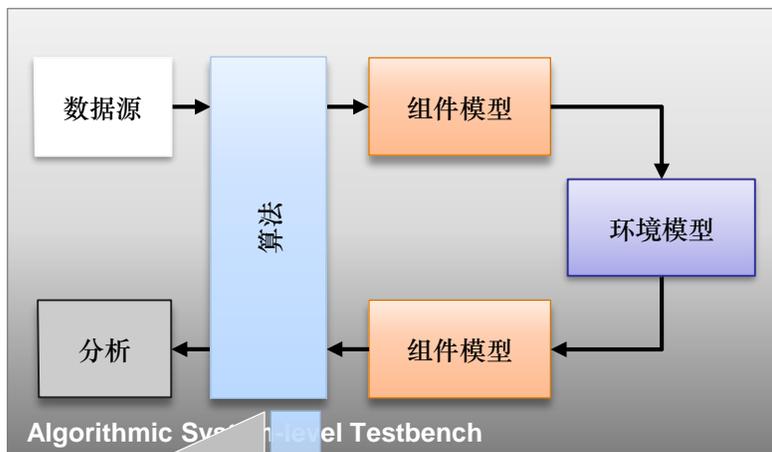


✓ 自动生成Testbench, 联合仿真

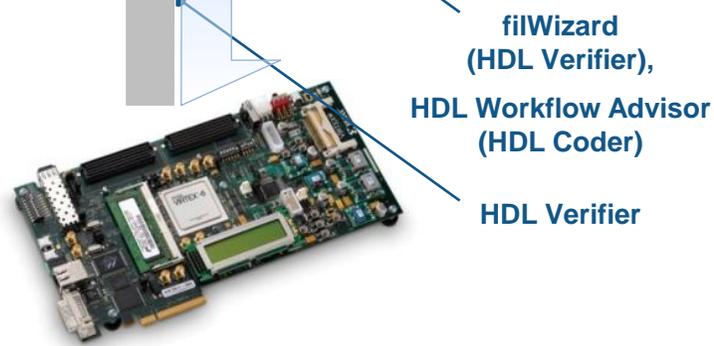


三、MATLAB解决方案

4 测试与验证



✓ 自动生成Testbench, FPGA硬件在环





汇报内容

1

基本情况介绍

2

传统FPGA开发流程

3

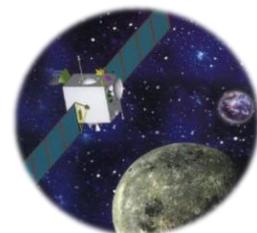
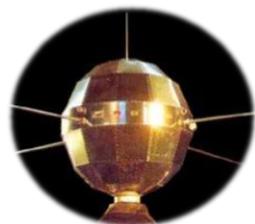
MATLAB解决方案

4

MATLAB应用实效

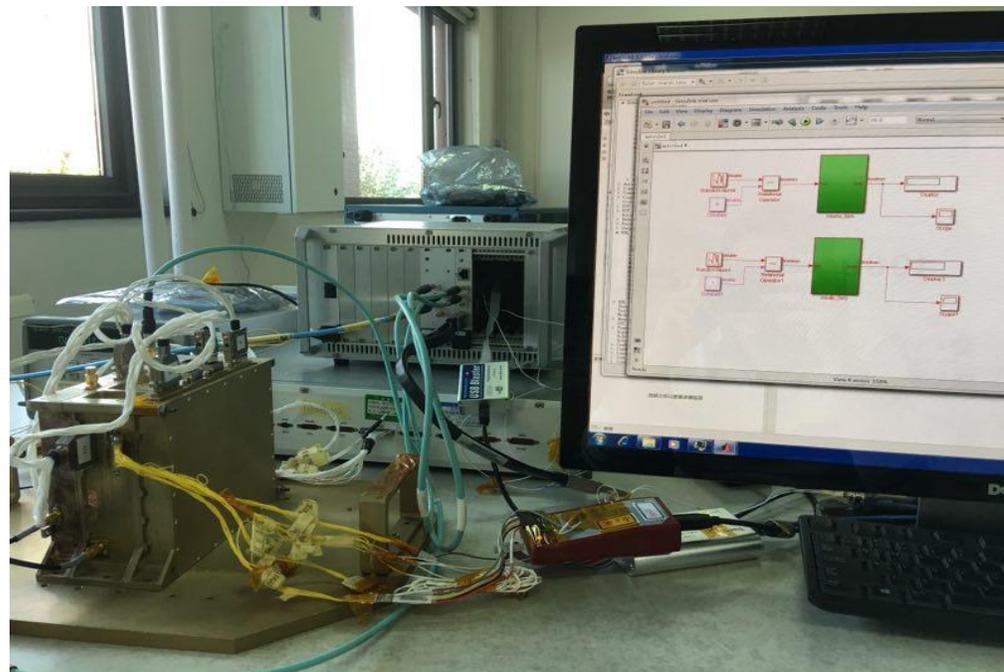
5

经验与总结



四、MATLAB应用实效

1 功能、性能指标测试



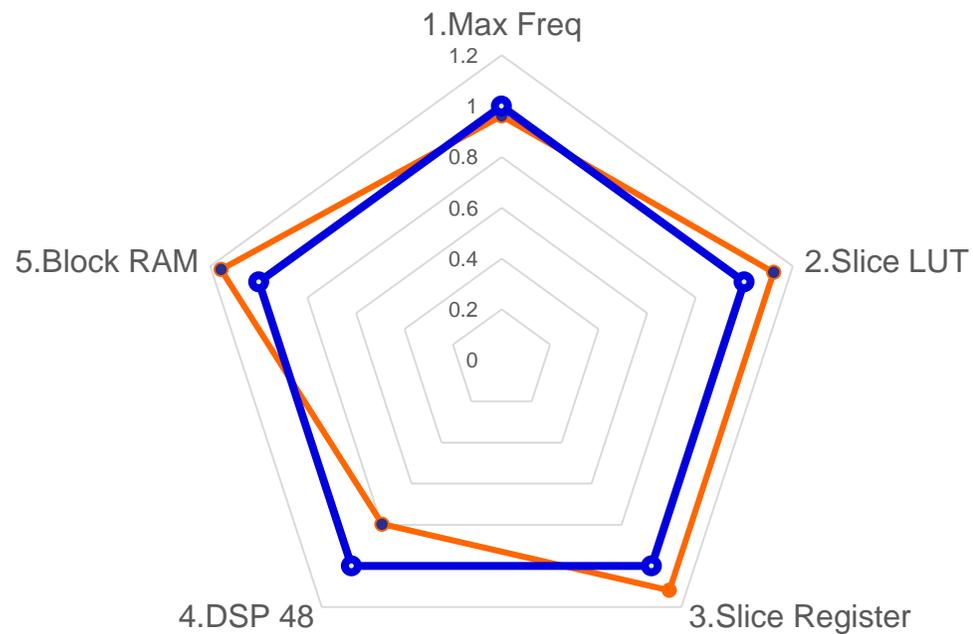
- 近4万行HDL代码，千万门电路规模。
- 功能及电性能指标完全满足要求！



2 代码性能

代码性能比较

—●— HDL Coder自动生成代码 —●— 手工优化代码





汇报内容

1

基本情况介绍

2

传统FPGA开发流程

3

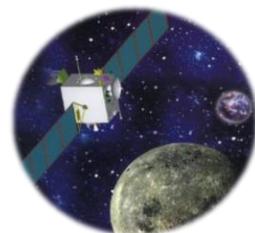
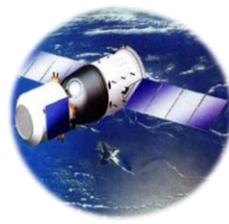
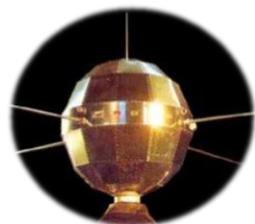
MATLAB解决方案

4

MATLAB应用实效

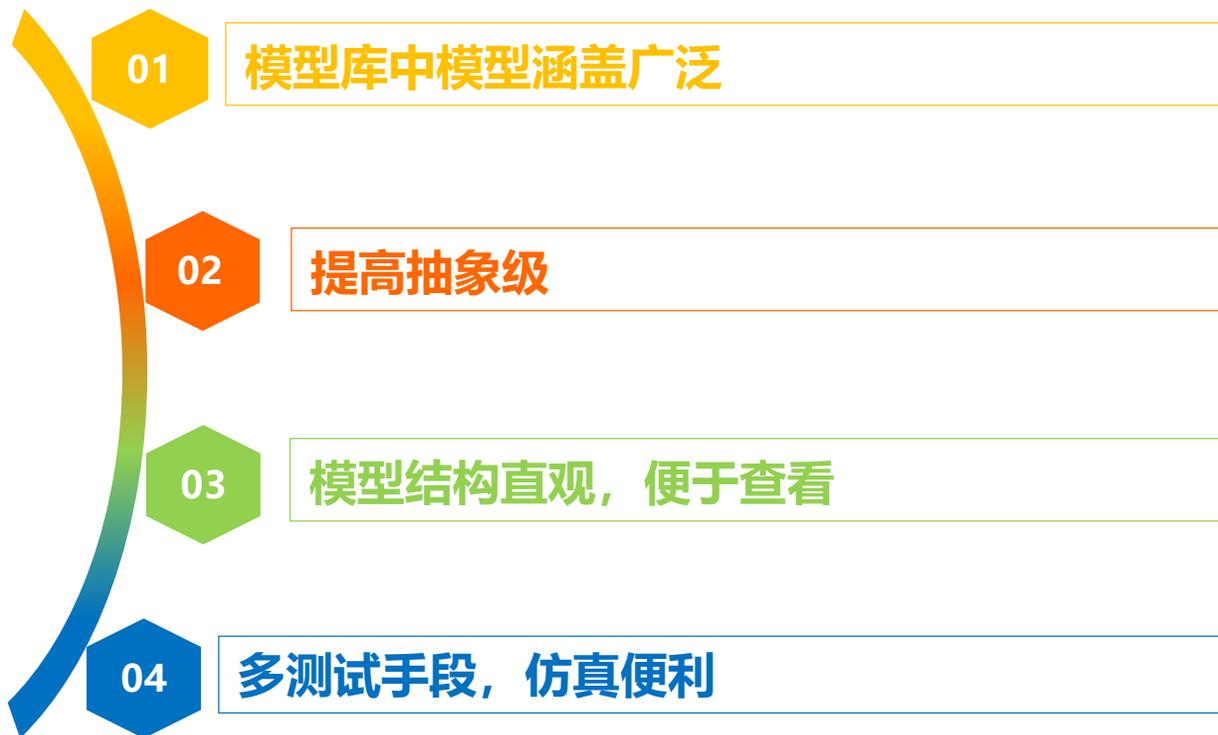
5

经验与总结



五、经验和总结

1 设计阶段



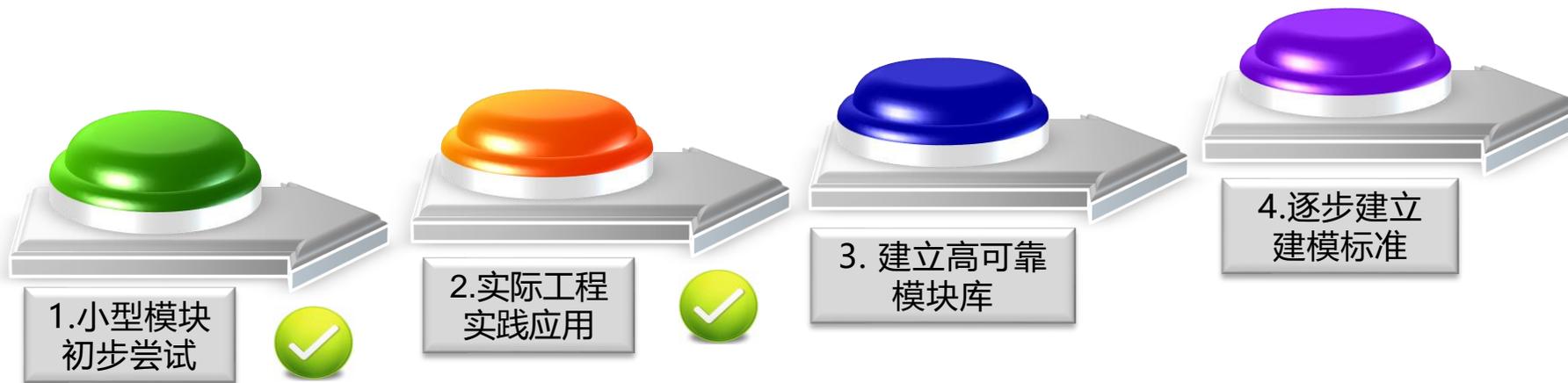
五、经验和总结

2 实现阶段



五、经验和总结

3 后续计划



谢谢!

