

# 2024 中国研究生创“芯”大赛·EDA 精英挑战赛 赛题指南

## 一、赛题名称

基于仿真波形的回归测试错误自动化诊断与分类

## 二、命题单位

芯华章科技股份有限公司

## 三、赛题主席

李曦（国防科技大学）

## 四、竞赛关键词

回归验证，数字电路，波形分析，模式识别

## 五、背景知识

随着芯片设计规模的不断扩大，回归测试 [1] 在芯片的验证环节中起到了不可替代的作用。在芯片设计的过程中，后加入的设计改动有可能改变某些已经测试通过的设计行为。另一方面，芯片规格的更新迭代也会导致设计团队与验证团队不同步。因此，验证工程师需要将已经通过的测试用例集成到一个特定的测试平台（Testbench）中并定期回归运行，以期及时发现项目开发中产生的问题和错误。

然而，回归失败的情况在芯片开发的过程中屡见不鲜，有的

问题是由于设计改动造成的，有的问题则是因为测试平台或测试环境发生变更所导致。而对于失败的回归测试用例，验证工程师需要结合调试工具对波形或日志进行反复分析，并最终将问题分诊并派发给相应的设计或验证人员。这个过程通常是手动的，在占用宝贵验证资源的同时，分诊准确率还常常无法得到保障，这对于整个项目的验证工作乃至产品上市时间（TTM）都是一个巨大的挑战。

## 六、参赛人员要求

1. 具备扎实的数字电路设计基础，能够对数字电路仿真波形进行分析；
2. 熟悉 Python 或 C/C++，具备良好的编程能力。

## 七、赛题描述

### （一）概览

本赛题聚焦在回归失败分析中的波形分析。对于某些回归测试用例，测试失败时的某些关键信号波形和测试通过时的波形存在着某些差异。验证工程师通常根据经验识别这些差异，并对造成失败的原因进行初步的诊断及分类。

本赛题将向参赛队伍提供一批 `xedb` 格式的波形文件，每个波形文件中均含有一定数量的关键信号数据。参赛队伍需要利用芯华章波形接口（支持 Python、C/C++ 及接口）读取波形文件以

获取相应的波形数据，并从这些波形数据中寻找规律或提取特征，继而设计出相应的分类算法或模型用以判断某个波形是否为错误波形（若是则需进一步指出其错误类型），以此达到自动化波形诊断与分类的目的。

## （二）可行性分析

测试失败时的某些关键信号波形和测试通过时的波形存在着某些差异。有的差异可以从波形图像上被识别，例如，图 1、图 2 是某 16bit 关键信号波形图，在图 1、图 2 中，该 16bit 数据被转化为 1 个整型数值进行绘图（纵轴为数值大小，横轴为时间），可以看出它们在波形图上存在着显著差异。除了通过波形图像本身识别错误，有的错误还可以通过某些关键特征或基于一定的规则被识别，例如，在错误波形中某个特殊值（例如 0）可能反复出现。因此，通过波形数据进行回归失败分析是可行的。

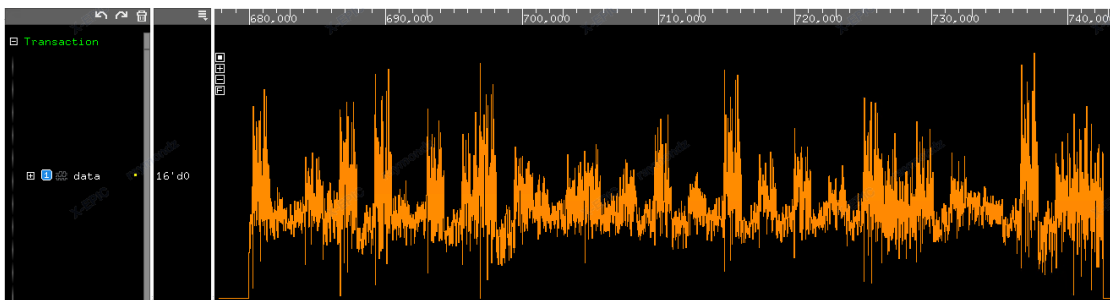
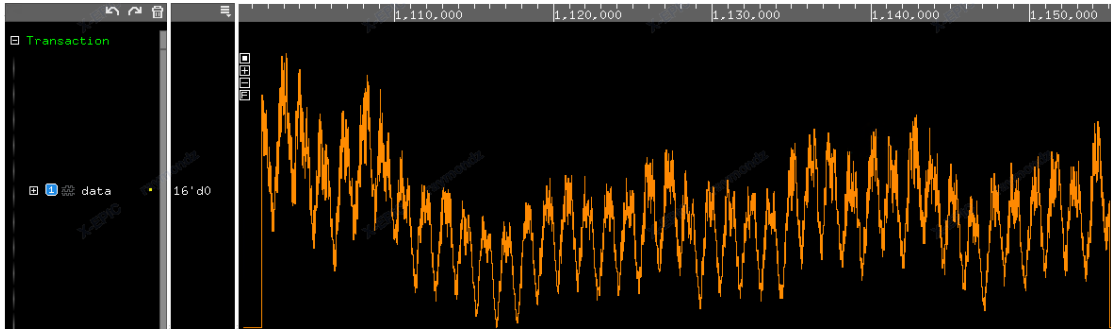
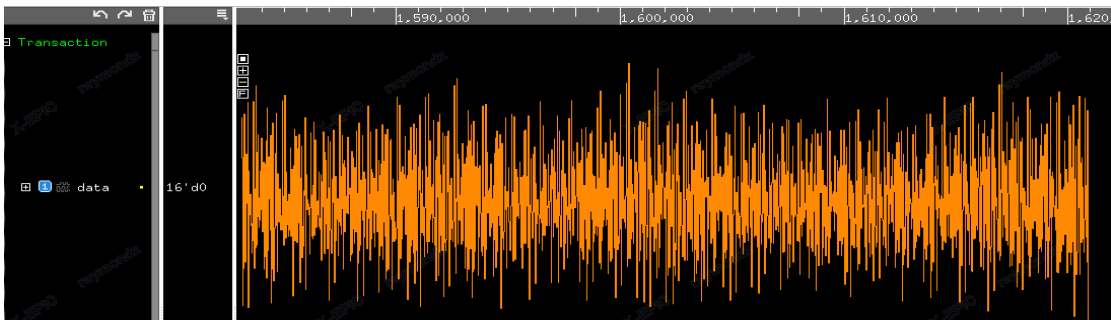


图 1. 测试通过时的关键信号波形示例



a)



b)

图 2. 测试失败时的关键信号波形示例。注意，a) 与 b) 是由不同的错误类型所导致

### (三) 输入数据

#### 1. Public Train Set:

**wave\_\$(id).xedb:** 仿真波形文件，约 5-10 个。文件名中的 id 用于唯一标识每个文件。每个波形文件中均含有一定数量的正确波形及错误波形(注：也可能全部为正确波形或全部为错误波形)。

每个波形文件均包含以下信号：

**vif.clk:** 时钟信号 (1 bit)

**vif.signal\_1:** 状态或控制信号 1 (1 bit)

**vif.signal\_2:** 状态或控制信号 2 (1 bit)

**vif.signal\_3:** 状态或控制信号 3 (1 bit)

**vif.data[i]:** 数据信号 (1 bit),  $i \in [0, 15]$

clk 上升沿, 且 signal\_1 及 signal\_2 同时为高电平时, data[i] 信号有效。signal\_3 为高电平时代表该帧数据为该笔数据的最后一帧数据。

如图 3 所示, signal\_1 及 signal\_2 在 152030 时刻由低电平跳变为高电平, 此时刻即为该笔数据的起始时刻。

如图 4 所示, signal\_3 在 231670 时刻由低电平跳变为高电平, 231670 时刻即为该笔数据的结束时刻(注: 时刻 152030 至 231670 期间, signal\_1 及 signal\_2 均保持高电平, 部分数据未在图中展示)

需要注意的是, 每笔数据的持续时间并不完全相同。在每笔数据的持续时间内含有若干帧数据, 每帧数据以时钟上升沿作为起始时刻, 每帧数据持续一个时钟周期。

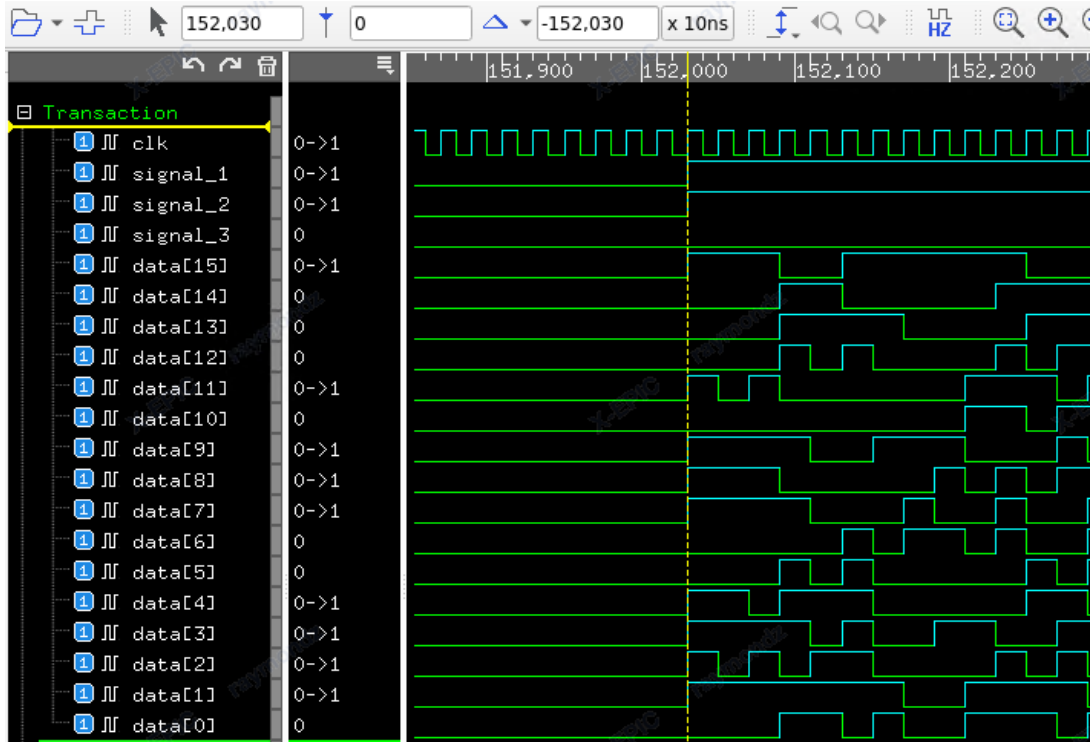


图 3. 数据起始说明: 黄色虚线标尺处 (152030) 为该笔数据起始时间

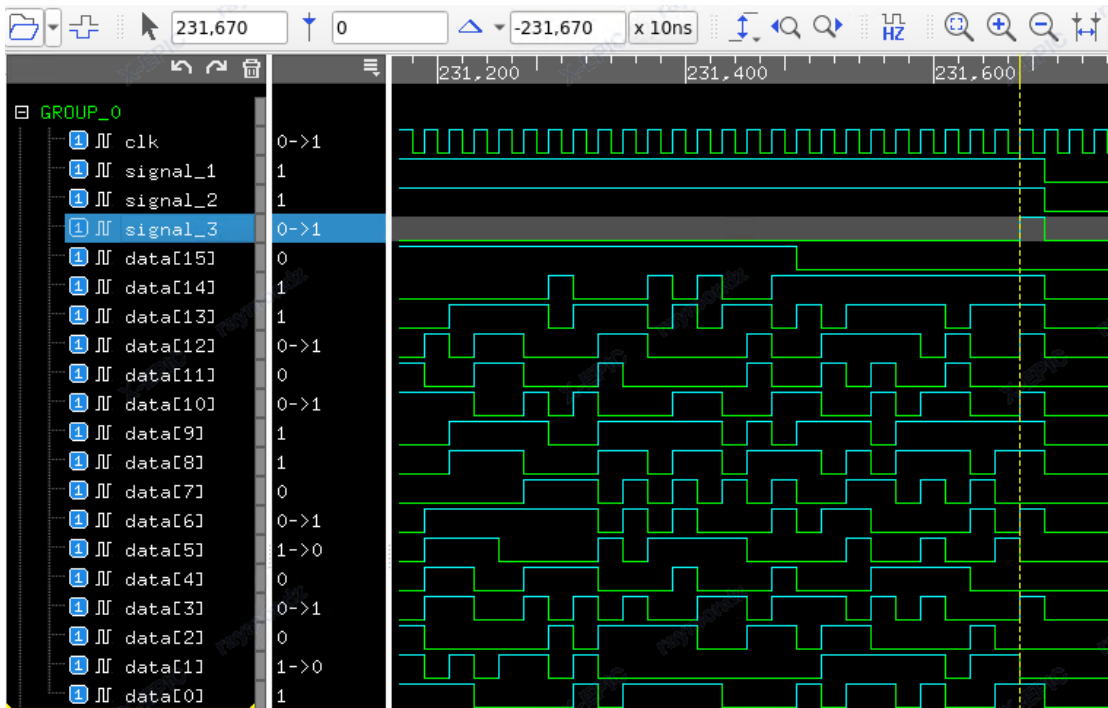


图 4. 数据结束说明: 黄色虚线标尺处 (231670) 为该笔数据结束时间

**wave\_\$id.csv**: 标签文件，数量同仿真波形文件，通过文件名上的 **id** 字段可以找到相应的仿真波形文件，该文件为标准 **csv** 文件，其内容示例见表 1。

start_time	end_time	label
152030	231670	0
324190	368210	0
439430	499510	1
576050	621850	2
698610	746290	3

表 1. 标签文件内容示例

标签文件字段说明：

**start\_time**：整型数据，代表数据起始时间（见图 3）；

**end\_time**: 整型数据，代表数据结束时间（见图 4）；

**label**: 整型数据，代表数据类型标签；标签有效范围: [0, 5], 标签说明见表 2。

label	说明
0	正确数据
1	错误数据，错误类型 1
2	错误数据，错误类型 2
3	错误数据，错误类型 3
4	错误数据，错误类型 4
5	错误数据，错误类型 5

表 2. 标签说明

## 2. Public Test Set

**wave\_\$id.xedb**: 仿真波形文件，约 20 个，内容同 public train set。

Public test set 标签信息将在比赛结束后将向参赛队伍公布。

### 3. Hidden Test Set

数据形式与 public test set 完全一致，但数据内容不同。该数据集在竞赛期间不向参赛者公布，数据集完整信息将在比赛结束后将向参赛队伍公布。

#### (四) 提交说明

作品以 zip 压缩文件的格式进行提交，作品名必须为“**submission.zip**”。**submission.zip** 内部需要存放作品运行所依赖的文件或数据。**submission.zip** 文件被提交至出题方服务器后，会被自动解压，并进入解压路径执行主程序。主程序可以是可执行文件或 **Python** 脚本文件，可执行文件必须以“**analyzer**”命名，**Python** 脚本必须以“**analyzer.py**”命名。

可执行文件或 **Python** 脚本需要接收一个 **xedb** 文件名作为输入参数，如：

```
./analyzer /data/path/wave_1.xedb
```

```
python ./analyzer.py /data/path/wave_1.xedb
```

可执行文件或 **Python** 脚本执行成功后在运行路径下的 **output** 路径中生成一个 **csv** 文件，该文件名前缀为输入文件名前缀，示例如下：



输入文件名：/data/path/wave\_1.xedb

输出文件名：./output/wave\_1.csv

该 csv 文件必须为标准 csv 格式（以逗号“,”作为分隔符，encoding 方式为“utf-8”），内容参见表 1。

submission.zip 文件大小不得大于 200MB。

主程序在单个波形文件上的运行时间不得大于 5 分钟。

## （五）评分标准

### 1. 数据识别得分（占比 20%）

某笔数据被正确识别的条件：满足 start\_time、end\_time 同时正确。以 F1-Score [2] 作为该项评分指标（计算示例参见附录 A）。

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (3)$$

### 2. 数据分类得分（占比 80%）

某笔数据被正确分类的条件：满足 start\_time、end\_time 及 label 同时正确。以 Macro F1-Score [3] 作为该项评分指标（计算示例参见附录 B）。

$$Macro\ F1 = \frac{1}{n} \sum_i F1_i \quad (4)$$

### 3. 效果总分

效果总分取“数据识别得分”与“数据分类得分”的加权后得分。

$$效果总分 = 数据识别得分 * 0.2 + 数据分类得分 * 0.8 \quad (5)$$

### 4. 性能得分（附加分，占比 3%）

“效果总分”排名前 30% 的队伍，将进行性能比拼。性能比拼以作品在数据集上的总运行时间作为评估指标，计算运行时间时，将限定最大 CPU 核数、最大内存占用等硬件资源。性能得分将按如下公式计算：

$$Performance_t = \frac{Time_{min}}{Time_t} \quad (6)$$

其中， $Time_{min}$  代表参与性能比拼的作品中性能最好作品的运行时间， $Time_t$  代表队伍  $t$  的作品运行时间。

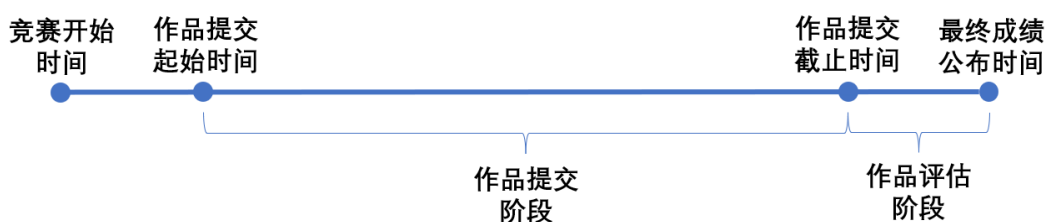
### 5. 最终得分

队伍的最终得分将按如下公式计算：

$$最终得分 = 效果总分 + 性能得分 * 0.03 \quad (7)$$

“效果总分”排名未进入前 30% 的队伍“性能得分”视做 0 分。

## (六) 竞赛流程



### ◆ 作品提交阶段

该阶段将在竞赛正式开始一段时间后开启。

参赛队伍在该阶段可以每天进行 2 次作品评估(评估方式将于后续公布), 评估数据集为 **public test set**, 需要注意的是, 在该阶段仅进行“效果总分”评估, 不会进行性能评估。各参赛队伍的最高“效果总分”将被用于队伍排名, 队伍排名将向所有参赛队伍公示(排名查询方式将于后续公布)。

在该阶段, 参赛队伍需要提交 1 份作品作为“最终作品”, 参赛队伍可随时修改“最终作品”的提交, “最终作品”将以最后一次提交为准。

注意, 该阶段的排名及成绩仅作为参考。

### ◆ 作品评估阶段

在该阶段, 参赛队伍将无法再提交“最终作品”。

参赛队伍的“最终作品”在该阶段会首先在 hidden test set 数据集上计算“效果总分”，并取排名前 30% 的队伍在 hidden test set 数据集上进行性能比拼（注 1：若前 30% 的队伍数量不足 10 支，则按 10 支计算；注 2：若前 30% 最后一名出现多支队伍同分，则同分队伍同时获得性能比拼资格），参与性能比拼的作品将获得额外的“性能得分”。所有队伍将按“最终得分”进行队伍排名。

## 八、参考资料

- [1] Cieplucha, M. Metric-Driven Verification Methodology with Regression Management. *J Electron Test* 35, 101–110 (2019).  
<https://doi.org/10.1007/s10836-019-05777-0>.
- [2] Nancy Chinchor, MUC-4 Evaluation Metrics, in *Proc. of the Fourth Message Understanding Conference*, pp. 22–29, 1992.
- [3] Juri Opitz and Sebastian Burst. Macro F1 and Macro F1. 2019. arXiv: 1911.03347.

## 附录

### A. 数据识别得分计算示例

	start_time	end_time	label
1	166950	231690	0
2	324190	368210	0
3	439430	499510	1
4	576050	621850	2
5	698610	746290	3
6	805450	868770	0
7	871910	912510	1
8	983170	1042550	2
9	1043570	1105410	0
10	1215590	1277410	0
11	1352810	1410310	4
12	1491670	1542670	0
13	1573590	1620890	4
14	1727370	1793630	5
15	1930390	1982630	0

a)

	start_time	end_time	label
1	166950	231690	0
2	324190	368210	0
3	439430	499370	1
4	576050	621850	1
5	698610	746290	2
6	805450	868770	0
7	871910	912510	0
8	983170	1042550	0
9	1043570	1105410	0
10	1215590	1277410	0
11	1352810	1410310	4
12	1491670	1542670	0
13	1573590	1620890	4
14	1723580	1793630	5
15	1823350	1892350	2
16	2062390	2103370	1

b)

表 3. 数据示例。a) 为实际标签；b) 为作品输出的结果，红色字体代表识别错误的时间

以表 3 为例，数据识别得分为：0.8387，计算细节见公式 8~10。注意，最终计算数据识别得分时会将所有输出文件合并在一起进行计算。

$$Precision = \frac{13}{13 + 3} \quad (8)$$

$$Recall = \frac{13}{13 + 2} \quad (9)$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = 0.8387 \quad (10)$$

### B. 数据分类得分计算示例

以表 3 为例，数据分类得分为：0.4667，计算明细见表 4。

注意，最终计算数据分类得分时会将所有输出文件合并在一起进行计算。

label	Precision	Recall	F1
0	$6/(6+2)$	$6/(6+1)$	0.8000
1	$0/(0+2)$	$0/(0+1)$	0.0000
2	$0/(0+2)$	$0/(0+2)$	0.0000
3	0	$0/(0+1)$	0.0000
4	$2/(2+0)$	$2/(2+0)$	1.0000
5	$1/(1+0)$	$1/(1+0)$	1.0000
<b>Macro F1</b>			0.4667

表 4. 数据分类得分 Macro F1-Score 计算明细

\*本赛题指南未尽问题，见赛题 Q&A 文件