**CCSDS-LDPC编码器**

|  |  |  |  |
| --- | --- | --- | --- |
| Document Version Control | | | |
| Version | Date | Author | Changes |
| 1.0.0 | 2023.12.21 | 王雨霄 | 实现了(8160,7136)的CCSDS-LDPC编码器 |
| 1.0.1 | 2024.1.15 | 王雨霄 | 增添了(8176,7154)码型的编码器 |
| 1.0.2 | 2024.1.24 | 王雨霄 | 增添了深空AR4JA-LDPC码的编码器 |
| 1.0.3 | 2024.2.8 | 王雨霄 | 增添了AR4JA-LDPC码的并行编码功能 |
| 1.0.4 | 2024.2.22 | 王雨霄 | 增添了(8176,7154)、(8160,7136) 码的并行编码功能 |
| 1.0.5 | 2024.2.29 | 王雨霄 | 修改AR4JA码型下的接口逻辑，提升数据吞吐率 |

1. **功能概述**

本设计参照CCSDS 131.1-O-2标准，对近地通信所使用的（8176,7154）、（8160,7136）LDPC编码器及深空通信所使用的AR4JA-LDPC编码器进行了Verilog实现（采用标准AXIS接口）。

1. **接口概述**

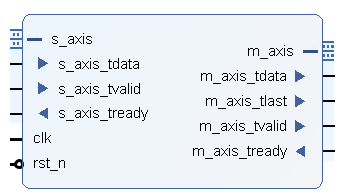
****

图1. 模块接口



图2. 接口功能表

1. **协议说明**

CCSDS 131.1-O-2标准的第二节<2 LOW DENSITY PARITY CHECK CODE OPTIMIZED FOR NEAR EARTH APPLICATIONS>给出了近地空间通信所采用的（8176,7154）及（8160,7136）QC-LDPC码的相关标准；标准中完整给出了（8176,7154）QC-LDPC码的生成矩阵及校验矩阵，并在（8176,7154）QC-LDPC码的基础上进一步规定了（8160,7136）QC-LDPC码，用以实现单个LDPC码块的n、k参数（k为编码前比特数，n为编码后比特数）均为32的整数倍；其中，（8176,7154）QC-LDPC码可由标准文件附录A中给出的生成矩阵直接编码得到，（8160,7136）QC-LDPC码则由（8176,7154）QC-LDPC码经如下环节变换而来：

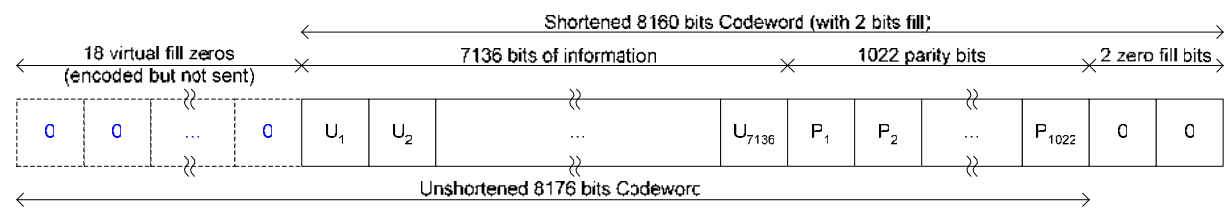


图3. 码字分割示意图



图4. 编码流程示意图

CCSDS 131.1-O-2标准的第三节<3 LOW DENSITY PARITY CHECK CODE FAMILY OPTIMIZED FOR DEEP SPACE APPLICATIONS>给出了深空通信所采用的AR4JA-QC-LDPC码的相关标准；标准中规定了如下图5所示的九种AR4JA码型，并给出了相应生成矩阵及校验矩阵的产生方法（相应代码请见MATLAB文件夹中的“AR4JA\_generator.m”文件）。

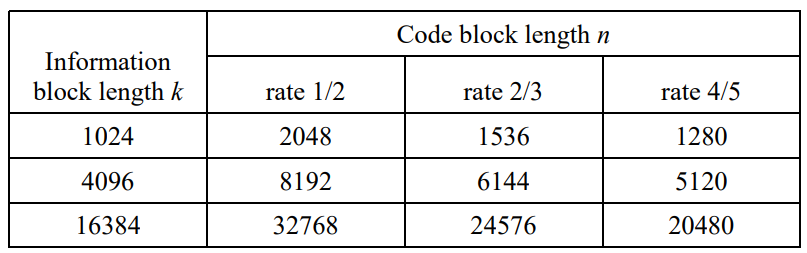


图5. AR4JA码型表

需要注意的是，由标准中的矩阵生成算法运算得到的生成矩阵需去除最后M列后才可作为编码矩阵，九种码型对应的M值如下图6所示：

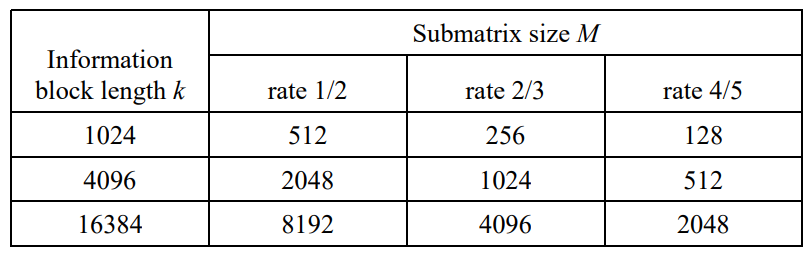


图6. M值表

以（2048，1024）AR4JA-QC-LDPC码为例，通过标准文件3.3及3.4小节规定的矩阵生成算法运算得到的生成矩阵G维度为1024\*2560，去除矩阵G的最后M（M=512）列后，得到一个维度为1024\*2048的矩阵G0，G0即为（2048，1024）码实际采用的编码矩阵。

1. **参数配置**

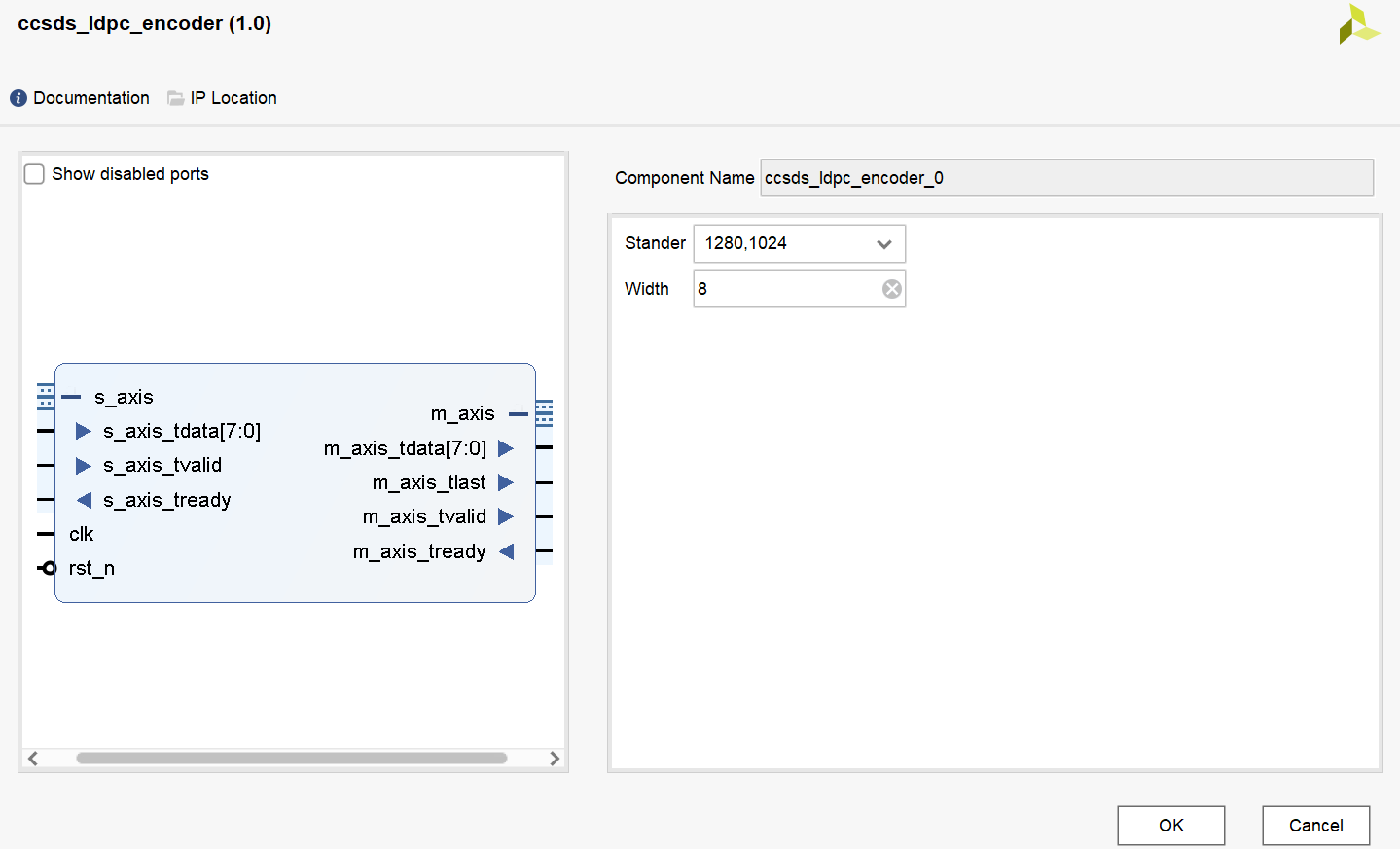


图3. 将设计导出为IP核后的参数配置界面

**1、stander**

设定LDPC编码所采用的码型，支持CCSDS 131.1-O-2标准中近地空间通信所采用的（8176,7154）、（8160,7136）两种QC-LDPC编码及深空通信所采用的（2048,1024）等九种AR4JA-QC-LDPC编码。

**2、width**

设定输入/输出位宽（即并行编码位宽），各码型支持的“width”参数如下：



**注：**并行编码时，每次输入中的MSB位被认为先进入编码器，每次输出中的MSB位被认为先离开编码器。

1. **脚本使用**

本节旨在描述通过脚本+源码文件来对设计进行Modelsim-RTL仿真并将设计导出为IP核的操作步骤。

**（1）.环境配置**

脚本在使用过程中会通过bat命令调用Vivado和Modelsim，因此，需要先将这些应用的相关路径添加至用户环境变量中（Vivado在安装完成后不会自动添加环境变量，需要手动添加其安装路径下的bin文件夹路径至用户环境变量中；Modelsim在安装过程中会自动添加环境变量，倘若安装完成后用户环境变量中未出现红框内的win64文件夹路径，可尝试重启电脑，若重启后仍未出现此路径，则需手动添加Modelsim安装路径下的win64文件夹路径至用户环境变量中）：

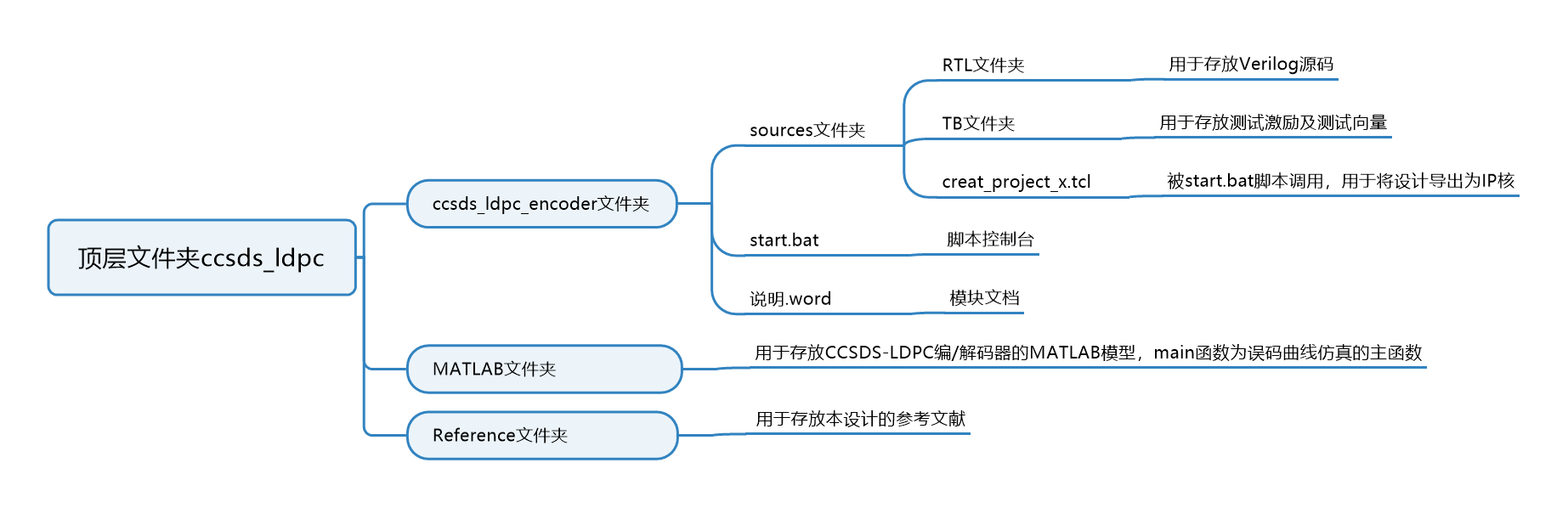




注：本工程使用Vivado **2021.2**、Modelsim **2020.4**、MATLAB **2020B**进行操作。

**（2）.RTL仿真**

总工程的文件结构目录如下：

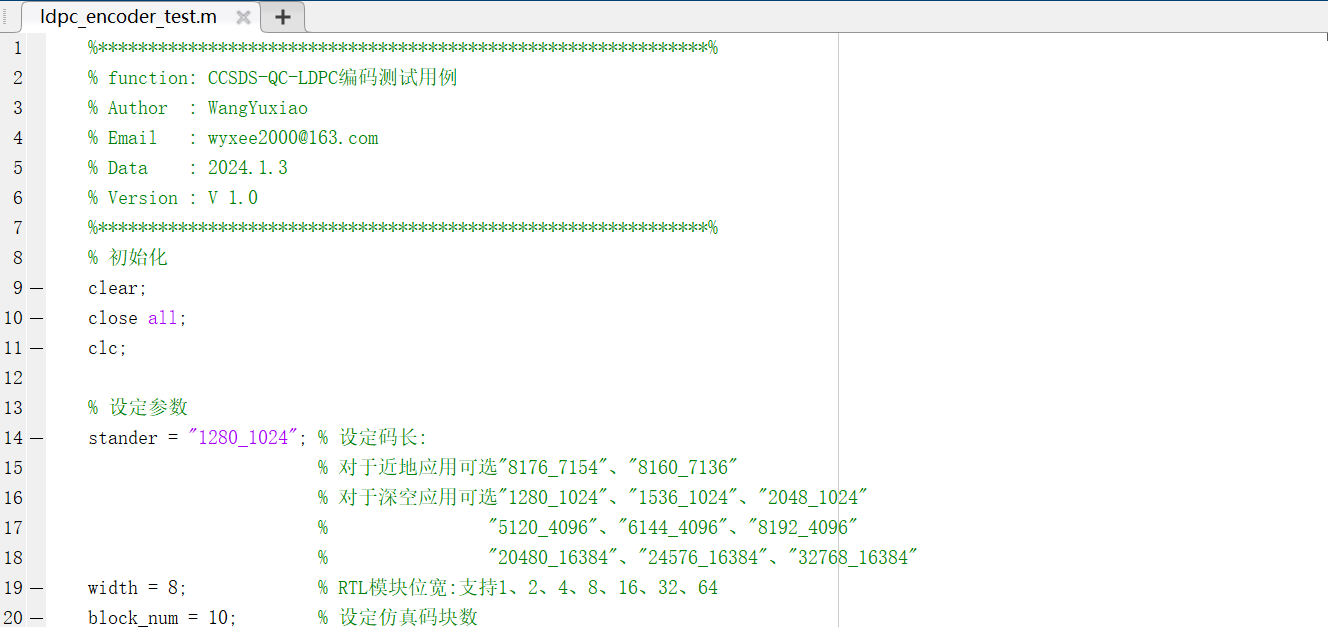


红框内即为本LDPC编码器的工程结构目录。

为运行RTL仿真，首先需通过如下链接将9个.mat文件下载至MATLAB文件夹中（也可通过运行MATLAB文件夹中的“AR4JA\_generator.m”得到这9个.mat文件）：

<https://pan.baidu.com/s/1jQmLIbyIN26tLI29vEFxkg?pwd=1ad3>

下载完成后，进入MATLAB文件夹中，通过MATLAB打开“ldpc\_encoder\_test.m”文件：

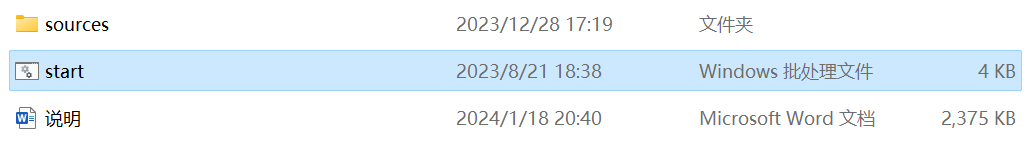


其中，第14行的stander值、第19行的width值应与IP核参数设置保持一致，第20行的block\_num用于设置仿真时LDPC编码的码块数量。

参数设定完成后，运行此.m文件，可见TB文件夹中生成了“stimulus.txt”与“response.txt”两个文件，即为RTL仿真所需的测试向量文件：

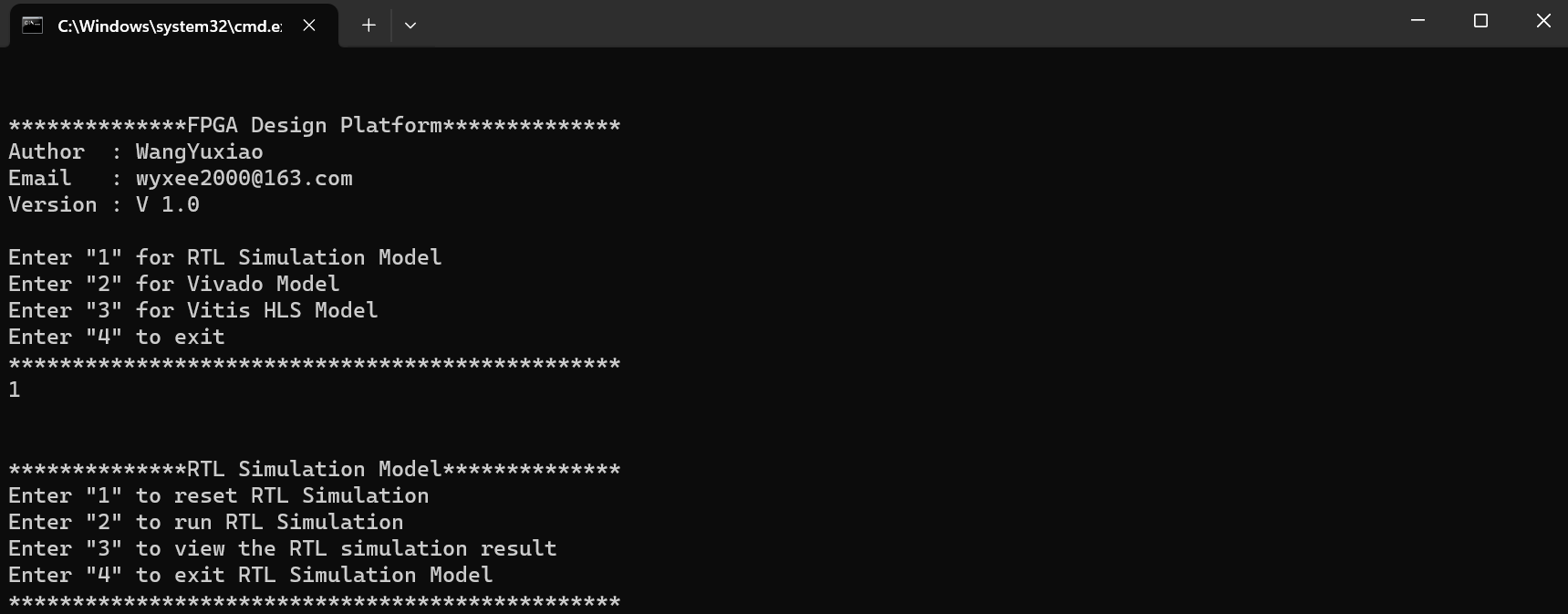


随后，进入ccsds\_ldpc\_encoder文件夹中，双击运行“start.bat”脚本：

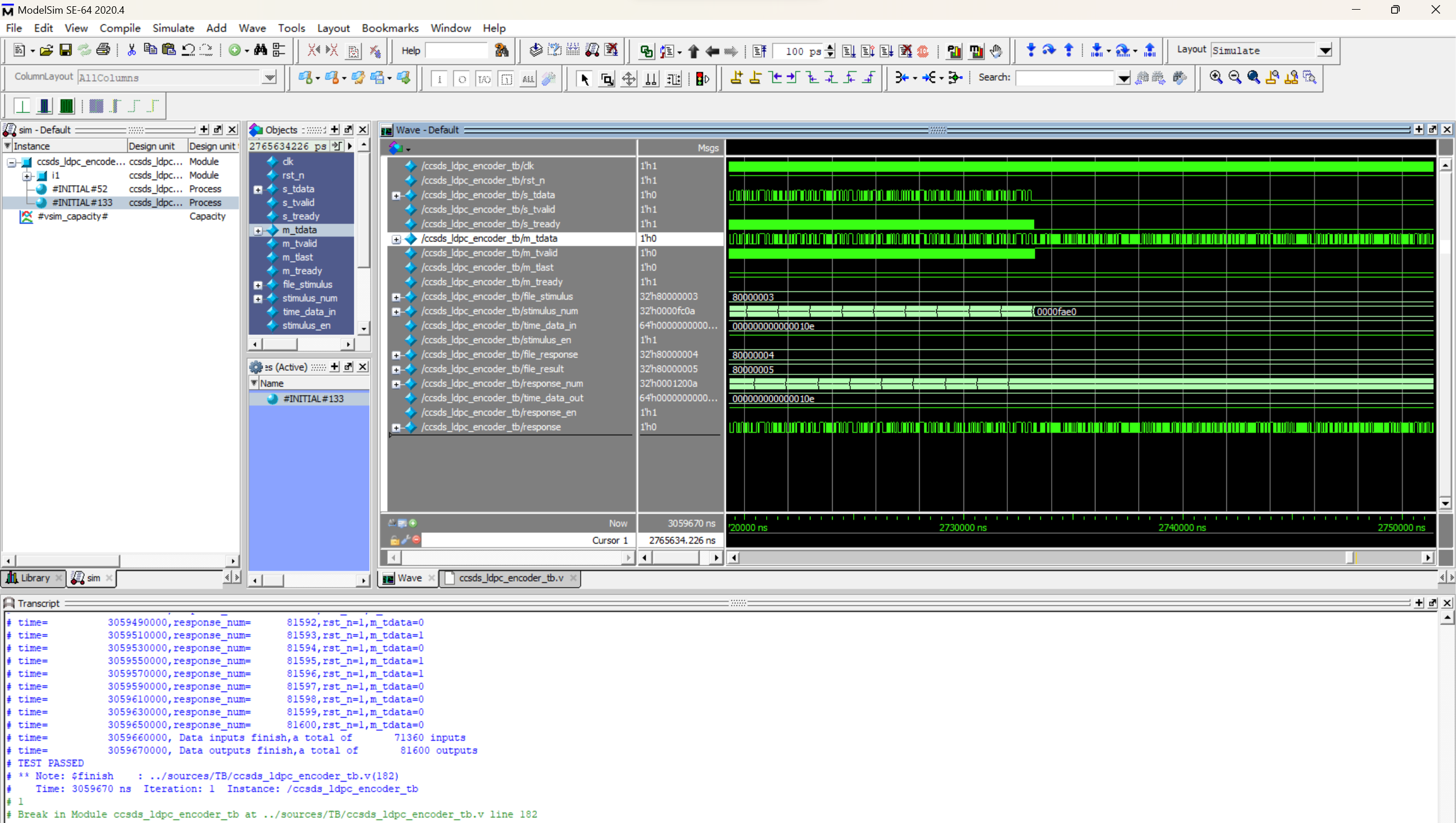




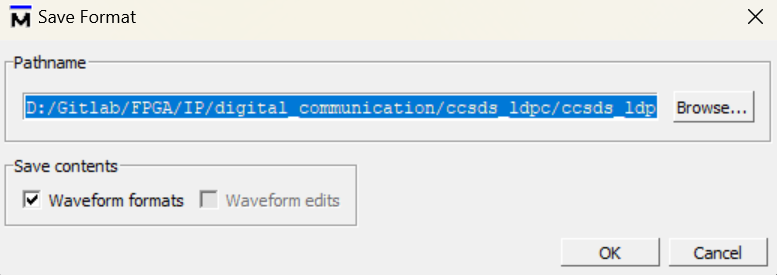
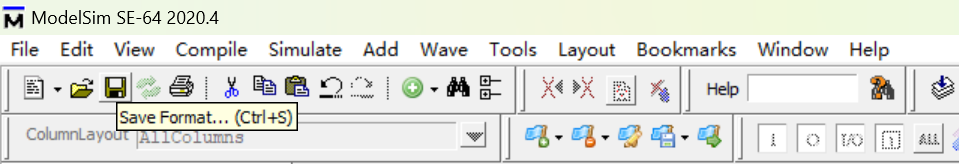
输入1：



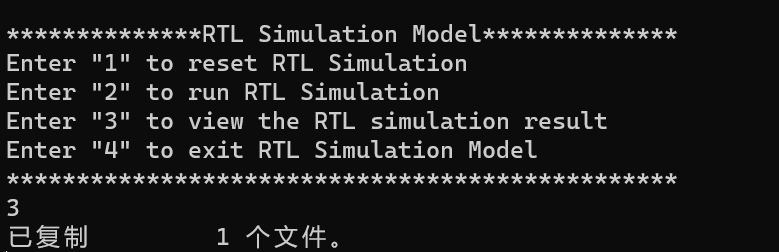
输入2，运行Modelsim仿真：

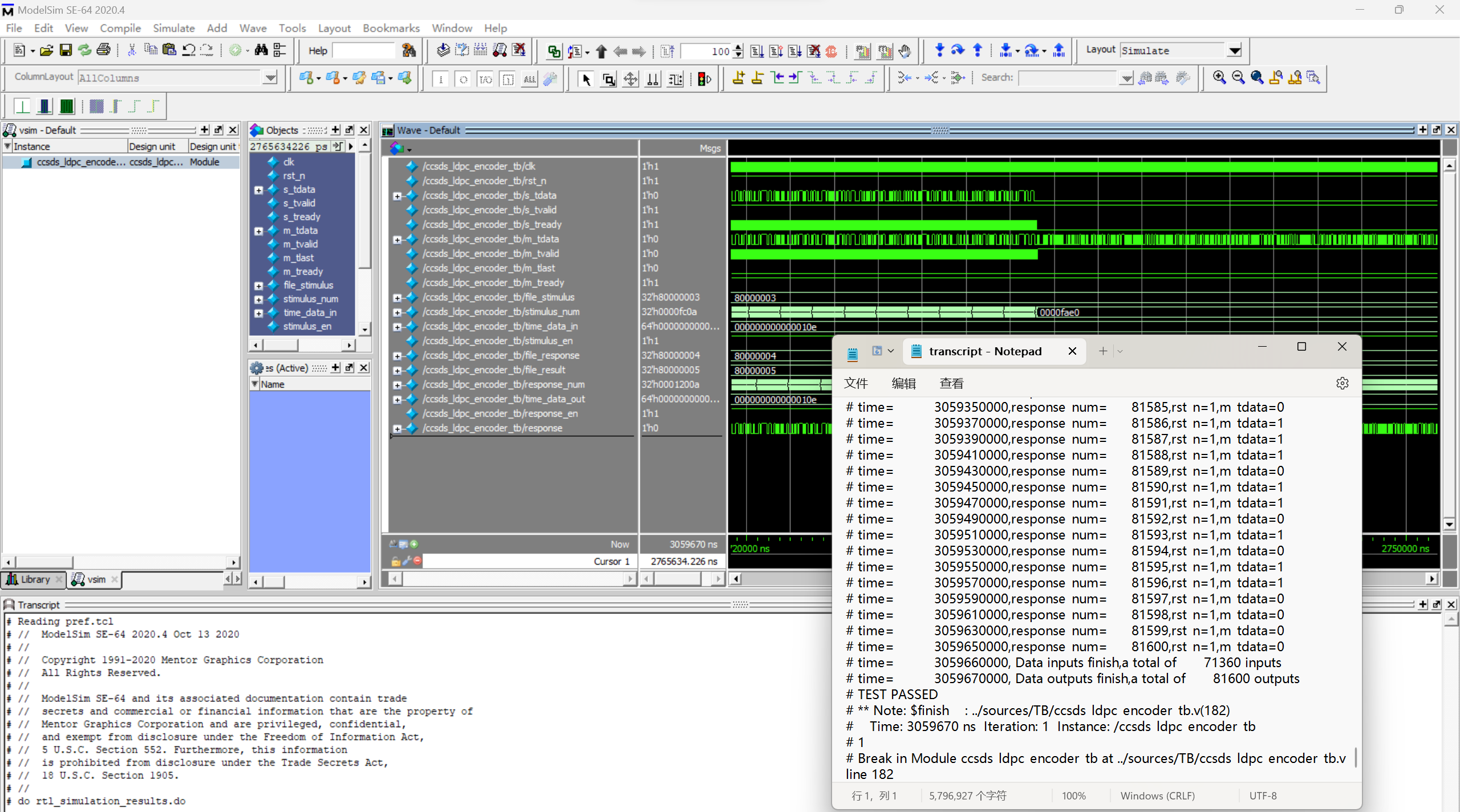


若想记录仿真结果以便后续查看，可通过菜单栏中的“Save Format”选项保存当前的波形设置：



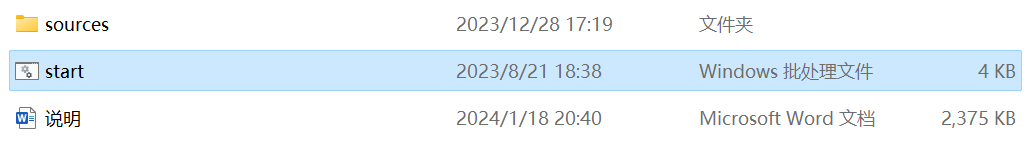
后续在脚本的RTL模式下输入3，即可恢复上次Modelsim仿真的波形及命令行信息：

****

****

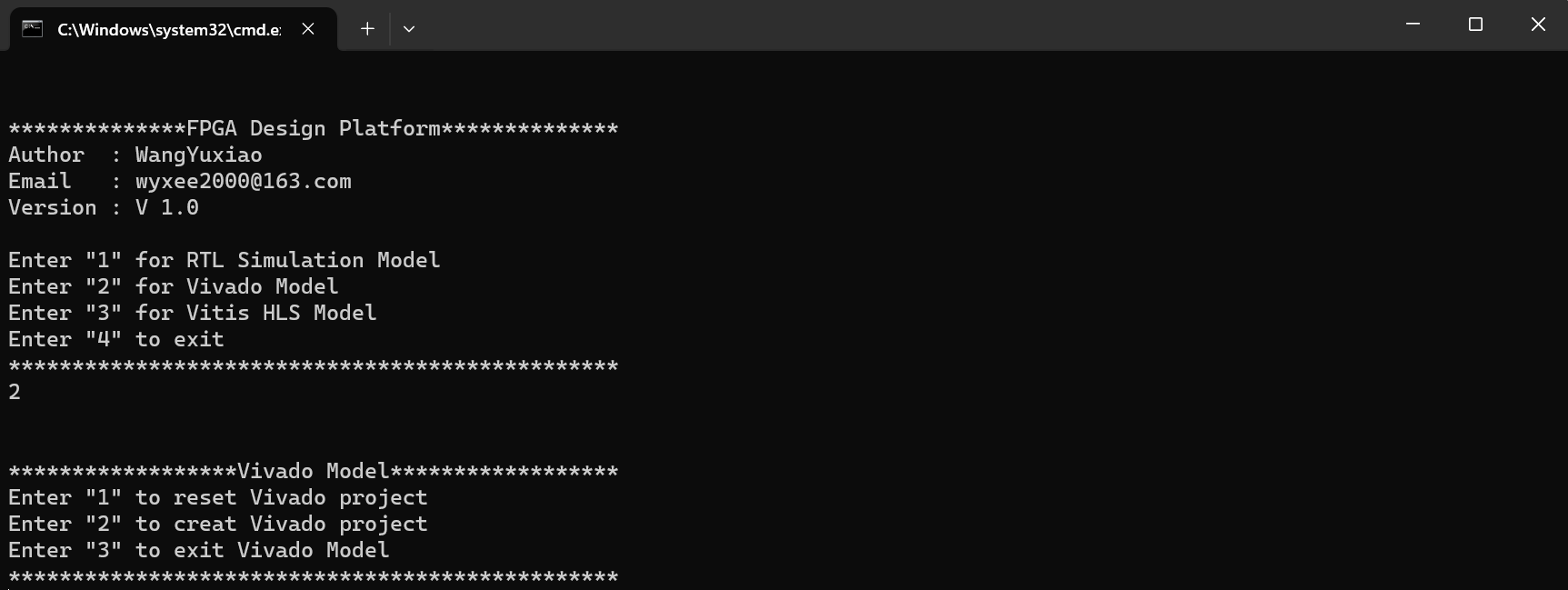
**（3）.导出IP**

进入ccsds\_ldpc\_encoder文件夹中，双击运行“start.bat”脚本：

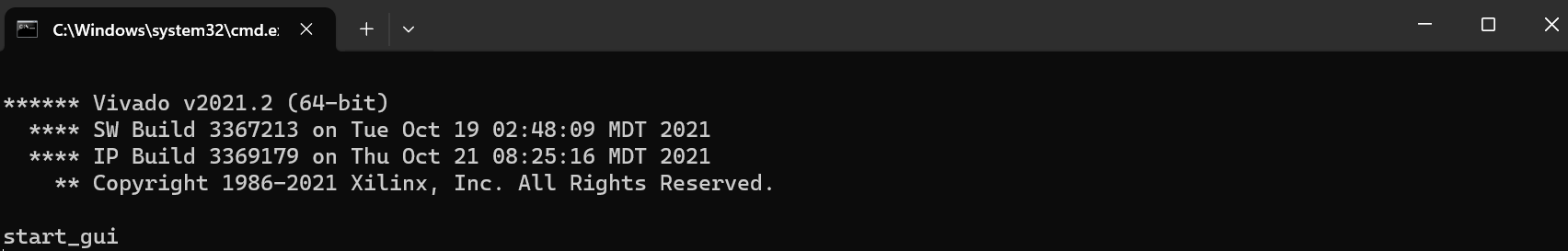
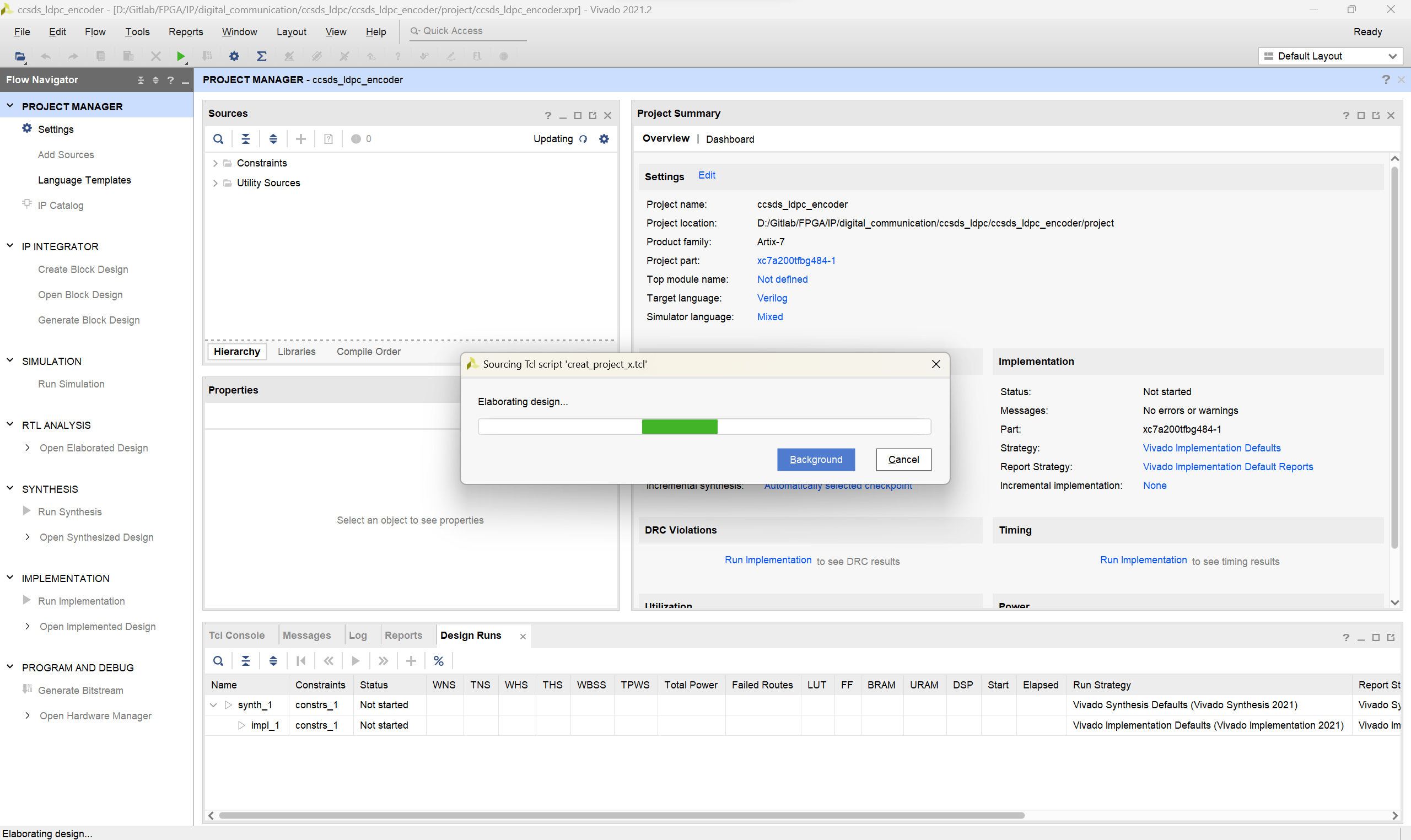




输入2：



再次输入2：

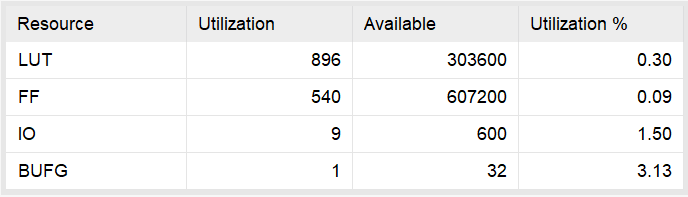
此时将弹出一个新的命令窗口，用于运行Vivado，导出“ccsds\_ldpc\_encoder” IP核。

运行完成后，ccsds\_ldpc\_encoder文件夹中出现了my\_ip的文件夹，刚刚生成的 IP核即存放在my\_ip文件夹内：

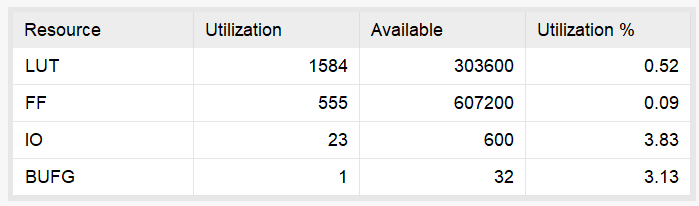


**附录：**

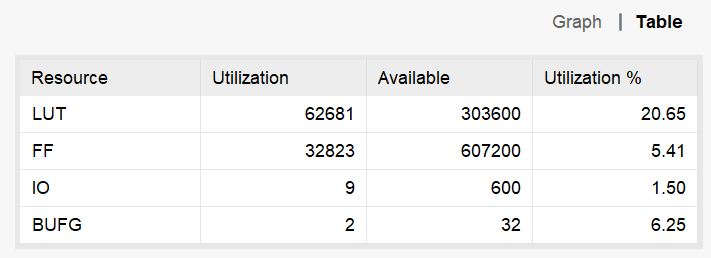
编码器在（1280,1024）码型、串行编码时的资源消耗：



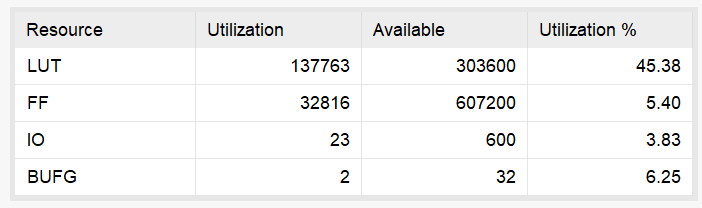
编码器在（1280,1024）码型、8路并行编码时的资源消耗：



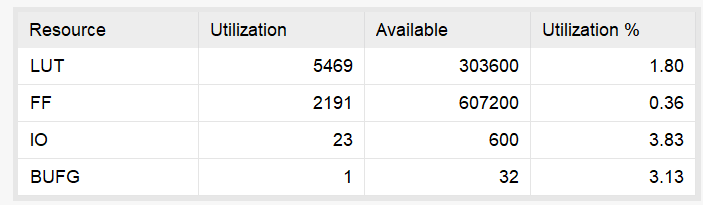
编码器在（32768,16384）码型、串行编码时的资源消耗：



编码器在（32768,16384）码型、8路并行编码时的资源消耗：



编码器在（8160,7136）码型、8路并行编码时的资源消耗：



width=8时，本编码器于xc7a200tfbg484-1平台上的实现结果：



其中，Fmax依照下图红框中的规则计算：

