# How to use

## 1. Test script brief introduction

**Git Repos**:

<https://sqbu-github.cisco.com/huashi/OpenH264MemCheckTool>

**Test Script:**

For more detail about scripts, please refer to 5.1.2~5.1.4

* **run\_ParseMemCheckLog.sh**

parse memory allocate/free/leak size for one test case

* **run\_MememAnalyseForOneYUV.sh**

test all cases for one YUV and output memory/FPS statistic info

* **run\_Main.sh**

test all YUVs for all cases and output all memory/FPS statistic info

* Other scripts as test tool called by above script

run\_GetYUVList.sh

run\_GetYUVPath.sh

run\_ParseYUVInfo.sh

**Test case in configuration file:**

Using default setting in below cfg files which copied from openh264/testbin/

* Welsenc.cfg
* Layer2.cfg

And combine with below parameters in run\_MememAnalyseForOneYUV.sh

  SliceMode=(0  1  2 3)

  SliceNum=(1 4  4 0 )

  ThreadNum=(1 2 3 4)

  SliceSize=(1500 600)

**Test codec:**

* Origin design

Codec version info, refer to Codec\_OriginDesign/Codec\_Info.txt

* New design

Codec version info, refer to Codec\_NewDesign/Codec\_Info.txt

* New design with reallocate step ¼ MaxSliceNumOld

Codec version info, refer to

Codec\_NewDesign\_Step\_1-4/Codec\_Info.txt

* New design with reallocate step 1/3 MaxSliceNumOld

Codec version info, refer to

Codec\_NewDesign\_Step\_1-3/Codec\_Info.txt

**Test data:**

* Overall data structure

**TestSpace ="TestData/${Codec}/${Arch}"**

Example:

TestData/Codec\_OriginDesign/x86

TestData/Codec\_OriginDesign/x64

TestData/Codec\_NewDesign/x86

TestData/Codec\_NewDesign/x64

TestData/Codec\_NewDesign\_Step\_1-3/x86

TestData/Codec\_NewDesign\_Step\_1-3/x64

TestData/Codec\_NewDesign\_Step\_1-4/x86

TestData/Codec\_NewDesign\_Step\_1-4/x64

* Test data file generate by **encoder** for one case

**TestMemLogFile="${TestSpace}/**enc\_mem\_check\_point\_${YUVName}\_${SliceMode[$i]}\_${SliceNum[$i]}\_${iThrdNum}\_${iSlcSize}.txt"

* Test data file generate by **run\_ParseMemCheckLog.sh** for one case

**TestAnalyseResult**="**${TestSpace}**/MemAnalyseResut\_${YUVName}\_${SliceMode[$i]}\_${SliceNum[$i]}\_${iThrdNum}\_${iSlcSize}.txt"

* Test data file for one YUV by **run\_MememAnalyseForOneYUV.sh**

**TestReport**="${**TestSpace**}/MemReport\_For\_${YUVName}.csv"

* Test data file for all YUVs by **run\_Main.sh**

**TestReportForAllYUV**="${**TestSpace**}/MemReport\_For\_AllYUVs.csv"

**Test data example:**

For **new design** with **x64** encoder,

**TestSpace=** TestData/Codec\_NewDesign/x64/

**case is**: slice mode = 1, slice num = 4,

thread num = 3, target bit rate = 600

**YUV is** : Zhuling\_1280x720.yuv

* memory file generated by encoder, rename and remove to:

TestData/Codec\_NewDesign/x64/enc\_mem\_check\_point\_Zhuling\_1280x720.yuv\_1\_4\_3\_600.txt

* Memory analyse result by **run\_ParseMemCheckLog.sh** is:

TestData/Codec\_NewDesign/x64/MemAnalyseResut\_Zhuling\_1280x720.yuv\_1\_4\_3\_600.txt

* Test report by **run\_MememAnalyseForOneYUV.sh** for Zhuling\_1280x720.yuv is:

TestData/Codec\_NewDesign/x64/MemReport\_For\_Zhuling\_1280x720.yuv.csv

* Ater test all YUV by **run\_Main.sh**; report for all YUVs is:

TestData/Codec\_NewDesign/x64/MemReport\_For\_AllYUVs.csv

## 2. Memory analyse for encode one case

**run\_ParseMemCheckLog.sh**

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#  brief:  1. parse memory allocate/free info based on memory statistic log

#             ./enc\_mem\_check\_point.txt

#

#          2. to generate above log file,

#              need to enable below macro in codec/common/memory\_align.h

#              #define MEMORY\_CHECK 1

#

#          3. the output may contain the summary of memory allocate/free

#             statistic info during encoding process, and detail info of all

#             buffers which have been allocated/freed during/after’

# encoding process

#

#  usage:   ./run\_ParseMemCheckLog.sh $LogFile $OutFile $Option

#            ----LogFile: should be enc\_mem\_check\_point.txt

# or other rename log file

#            ----OutFile:  the final report file for memory statistic info

#            ----Option:

#                       1)MemCheck     :   all detail info

#                       2)OverallCheck :   summary info only

#

#

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Example:**

Analyse new design x64 encoder’s memory info, with

**Case** slice mode = 1, slice num = 4,

thread num = 3, target bit rate = 600

**LogFile**=”TestData/Codec\_NewDesign/x64/enc\_mem\_check\_point\_Zhuling\_1280x720.yuv\_1\_4\_3\_600.txt”

**OutFile**=”TestData/Codec\_NewDesign/x64/MemAnalyseResut\_Zhuling\_1280x720.yuv\_1\_4\_3\_600.txt”

* **./run\_ParseMemCheckLog.sh ${LogFile} ${OutFile} OverallCheck**
* **./run\_ParseMemCheckLog.sh ${LogFile} ${OutFile} MemCheck**

if using OverallCheck, will only output below summary info

TestData/Codec\_NewDesign/x64/MemAnalyseResut\_Zhuling\_1280x720.yuv\_1\_4\_3\_600.txt

MemLeakStatus False: AllocatedNum--FreeNum (116--116) : AllocateSize==FreeSize==LeakSize (25615532==25615532==0)

If using OverallCheck, will output detail info about all buffer allocate size, free size, allocate num, free num etc

## 3. memory analyse for one YUV

**run\_MememAnalyseForOneYUV.sh** #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#  brief:  Analyse memroy allocate statistic info for one YUV with

#              all test cases

#

#  usage:   ./run\_MememAnalyseForOneYUV.sh \$YUVName \$YUVDir \$TestSpace

#            ----YUVName:  Test YUV’s name

#            ----YUVDir:   Test YUV's folder, script will search YUV

# within this folder

#            ----TestSpace: Test data output folder

#

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Test case for one YUV in script(you can modify it if you want):**

**SliceMode=(0  1  2 3)**

**SliceNum=(1 4  4 0 )**

**ThreadNum=(1 2 3 4)**

**SliceSize=(1500 600)**

**Example:**

**Test yuv with new design with arch=x64 encoder,**

* **Step 1:**

copy encoder h264enc from Codec\_NewDesign/x64/

* **Step 2:**

YUVName=” Zhuling\_1280x720.yuv”

YUVDir=”../../YUV”

TestSpace=” TestData/Codec\_NewDesign/x64”

* **Step 3:**

**./run\_MememAnalyseForOneYUV.sh ${YUVName} ${YUVDir} ${TestSpace}**

**Then all test data for** Zhuling\_1280x720.yuv is under folder

TestData/Codec\_NewDesign/x64/

**Test report file for zhuling is:**

TestData/Codec\_NewDesign/x64/MemReport\_For\_Zhuling\_1280x720.yuv.csv

## 4. Memory analyse for all YUVs

**run\_Main.sh**

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#  brief:  Analyse memroy allocate statistic info for all YUVs with

#              all test cases

#

#  usage:   ./run\_Main.sh \$TestYUVListDir

#            ----TestYUVListDir: Test YUV's folder, script will search YUV

# in this folder

#

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Test all YUVs with all test cases using all encoder in encoder set**

You can modify below test sets if you want

  TestYUVList=("CiscoVT2people\_320x192\_12fps.yuv"  \

               "xuemei\_640x360.yuv" \

               "Zhuling\_1280x720.yuv" \

               "desktop\_dialog\_1920x1080\_i420.yuv" \

               "SlideShowFast\_SCC\_2880x1800.yuv")

  TestCodecList=("Codec\_NewDesign" \

                 "Codec\_NewDesign\_Step\_1-3" \

                 "Codec\_NewDesign\_Step\_1-4" \

                 "Codec\_OriginDesign")

  TestCodecArchList=("x64" "x86")

**And combine with cases in** **run\_MememAnalyseForOneYUV.sh**

SliceMode=(0  1  2 3)

  SliceNum=(1 4  4 0 )

  ThreadNum=(1 2 3 4)

  SliceSize=(1500 600)

**Example:**

**./run\_Main.sh ../../YUV**

**Then all test data will be generated under all below folders:**

TestData/Codec\_OriginDesign/x86

TestData/Codec\_OriginDesign/x64

TestData/Codec\_NewDesign/x86

TestData/Codec\_NewDesign/x64

TestData/Codec\_NewDesign\_Step\_1-3/x86

TestData/Codec\_NewDesign\_Step\_1-3/x64

TestData/Codec\_NewDesign\_Step\_1-4/x86

TestData/Codec\_NewDesign\_Step\_1-4/x64

## 5. Example for final Test Data

**All test data for all test encoder all arch wit all test YUVs for all cases:**

<https://sqbu-github.cisco.com/huashi/OpenH264MemCheckTool/tree/master/TestData>

**Final Test report for all YUVs:**

<https://sqbu-github.cisco.com/huashi/OpenH264MemCheckTool/blob/master/NewDesignTestData.xlsx>

example for SlideShowFast\_SCC\_2880x1800.yuv



-15.39% means that new design decrease overall allocate size;

43.10% means that new design increase overall allocate size due to reallocate modue which is not allowed in origin design

**Final test summary:**

New design most of time save the overall memory size,

FPS is the same with origin disign