# H264 Test Manual

## Platform and Architecture

This manual is based on the following platforms and configurations:

ARM platform:

- Architecture: ARM(aarch64)

- Server version: Kunpeng 920-7260

- Kernel version: Linux kunpeng2 (5.10.0-209.0.0.117.oe2203sp3.aarch64 #1 SMP Wed Jun 19 18:00:16 CST 2024)

- GCC&G++ version: 10.3.0

- JAVA version: 1.8.0\_412

- Python version: 3.12.2

RISC-V Platform:

- Architecture: RISCV (RISC-V64)

- Server version: SG2042

- Kernel version: Linux openeuler-riscv-4-2 6.6.0 #1 SMP Tue Jul 2 11:21:06 CST 2024 riscv64 riscv64 riscv64 GNU/Linux

- GCC&G++ version: 12.3.1

- JAVA version: 11.0.22

- Python version: 3.12.2

## Install ffmpeg

The H264 test is based on ffmpeg, so you need to install ffmpeg for testing:

```

./ffmpeg.sh

```

For more details on verification, see the ffmpeg documentation.

## Get test data and scripts

We need to download the benchmark framework and execute the script directly:

```sh

./build.sh

```

## Test items

### Test case 1: fate-h264-conformance-aud\_mw\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-aud\_mw\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-aud\_mw\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 2: fate-h264-conformance-ba1\_ft\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-ba1\_ft\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-ba1\_ft\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 3: fate-h264-conformance-ba1\_sony\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-ba1\_sony\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-ba1\_sony\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 4: fate-h264-conformance-ba2\_sony\_f

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-ba2\_sony\_f SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-ba2\_sony\_f

```

The ARM platform is the same as the RISCV platform.

### Test case 5: fate-h264-conformance-ba3\_sva\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-ba3\_sva\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-ba3\_sva\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 6: fate-h264-conformance-ba\_mw\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-ba\_mw\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-ba\_mw\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 7: fate-h264-conformance-bamq1\_jvc\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-bamq1\_jvc\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-bamq1\_jvc\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 8: fate-h264-conformance-bamq2\_jvc\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-bamq2\_jvc\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-bamq2\_jvc\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 9: fate-h264-conformance-banm\_mw\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-banm\_mw\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-banm\_mw\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 10: fate-h264-conformance-basqp1\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-basqp1\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-basqp1\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 11: fate-h264-conformance-caba1\_sony\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-caba1\_sony\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-caba1\_sony\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 12: fate-h264-conformance-caba1\_sva\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-caba1\_sva\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-caba1\_sva\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 13: fate-h264-conformance-caba2\_sony\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-caba2\_sony\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-caba2\_sony\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 14: fate-h264-conformance-caba2\_sva\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-caba2\_sva\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-caba2\_sva\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 15: fate-h264-conformance-caba3\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-caba3\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-caba3\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 16: fate-h264-conformance-caba3\_sva\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-caba3\_sva\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-caba3\_sva\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 17: fate-h264-conformance-caba3\_toshiba\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-caba3\_toshiba\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-caba3\_toshiba\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 18: fate-h264-conformance-cabac\_mot\_fld0\_full

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cabac\_mot\_fld0\_full SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cabac\_mot\_fld0\_full

```

The ARM platform is the same as the RISCV platform.

### Test case 19: fate-h264-conformance-cabac\_mot\_frm0\_full

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cabac\_mot\_frm0\_full SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cabac\_mot\_frm0\_full

```

The ARM platform is the same as the RISCV platform.

### Test case 20: fate-h264-conformance-cabac\_mot\_mbaff0\_full

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cabac\_mot\_mbaff0\_full SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cabac\_mot\_mbaff0\_full

```

The ARM platform is the same as the RISCV platform.

### Test case 21: fate-h264-conformance-cabac\_mot\_picaff0\_full

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cabac\_mot\_picaff0\_full SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cabac\_mot\_picaff0\_full

```

The ARM platform is the same as the RISCV platform.

### Test case 22: fate-h264-conformance-cabaci3\_sony\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cabaci3\_sony\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cabaci3\_sony\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 23: fate-h264-conformance-cabast3\_sony\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cabast3\_sony\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cabast3\_sony\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 24: fate-h264-conformance-cabastbr3\_sony\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cabastbr3\_sony\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cabastbr3\_sony\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 25: fate-h264-conformance-cabref3\_sand\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cabref3\_sand\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cabref3\_sand\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 26: fate-h264-conformance-cacqp3\_sony\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cacqp3\_sony\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cacqp3\_sony\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 27: fate-h264-conformance-cafi1\_sva\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cafi1\_sva\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cafi1\_sva\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 28: fate-h264-conformance-cama1\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cama1\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cama1\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 29: fate-h264-conformance-cama1\_toshiba\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cama1\_toshiba\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cama1\_toshiba\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 30: fate-h264-conformance-cama1\_vtc\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cama1\_vtc\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cama1\_vtc\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 31: fate-h264-conformance-cama2\_vtc\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cama2\_vtc\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cama2\_vtc\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 32: fate-h264-conformance-cama3\_sand\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cama3\_sand\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cama3\_sand\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 33: fate-h264-conformance-cama3\_vtc\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cama3\_vtc\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cama3\_vtc\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 34: fate-h264-conformance-camaci3\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-camaci3\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-camaci3\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 35: fate-h264-conformance-camanl1\_toshiba\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-camanl1\_toshiba\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-camanl1\_toshiba\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 36: fate-h264-conformance-camanl2\_toshiba\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-camanl2\_toshiba\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-camanl2\_toshiba\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 37: fate-h264-conformance-camanl3\_sand\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-camanl3\_sand\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-camanl3\_sand\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 38: fate-h264-conformance-camasl3\_sony\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-camasl3\_sony\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-camasl3\_sony\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 39: fate-h264-conformance-camp\_mot\_mbaff\_l30

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-camp\_mot\_mbaff\_l30 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-camp\_mot\_mbaff\_l30

```

The ARM platform is the same as the RISCV platform.

### Test case 40: fate-h264-conformance-camp\_mot\_mbaff\_l31

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-camp\_mot\_mbaff\_l31 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-camp\_mot\_mbaff\_l31

```

The ARM platform is the same as the RISCV platform.

### Test case 41: fate-h264-conformance-canl1\_sony\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-canl1\_sony\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-canl1\_sony\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 42: fate-h264-conformance-canl1\_sva\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-canl1\_sva\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-canl1\_sva\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 43: fate-h264-conformance-canl1\_toshiba\_g

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-canl1\_toshiba\_g SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-canl1\_toshiba\_g

```

The ARM platform is the same as the RISCV platform.

### Test case 44: fate-h264-conformance-canl2\_sony\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-canl2\_sony\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-canl2\_sony\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 45: fate-h264-conformance-canl2\_sva\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-canl2\_sva\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-canl2\_sva\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 46: fate-h264-conformance-canl3\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-canl3\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-canl3\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 47: fate-h264-conformance-canl3\_sva\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-canl3\_sva\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-canl3\_sva\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 48: fate-h264-conformance-canl4\_sva\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-canl4\_sva\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-canl4\_sva\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 49: fate-h264-conformance-canlma2\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-canlma2\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-canlma2\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 50: fate-h264-conformance-canlma3\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-canlma3\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-canlma3\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 51: fate-h264-conformance-capa1\_toshiba\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-capa1\_toshiba\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-capa1\_toshiba\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 52: fate-h264-conformance-capama3\_sand\_f

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-capama3\_sand\_f SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-capama3\_sand\_f

```

The ARM platform is the same as the RISCV platform.

### Test case 53: fate-h264-conformance-capcm1\_sand\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-capcm1\_sand\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-capcm1\_sand\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 54: fate-h264-conformance-capcmnl1\_sand\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-capcmnl1\_sand\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-capcmnl1\_sand\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 55: fate-h264-conformance-capm3\_sony\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-capm3\_sony\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-capm3\_sony\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 56: fate-h264-conformance-caqp1\_sony\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-caqp1\_sony\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-caqp1\_sony\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 57: fate-h264-conformance-cavlc\_mot\_fld0\_full\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cavlc\_mot\_fld0\_full\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cavlc\_mot\_fld0\_full\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 58: fate-h264-conformance-cavlc\_mot\_frm0\_full\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cavlc\_mot\_frm0\_full\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cavlc\_mot\_frm0\_full\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 59: fate-h264-conformance-cavlc\_mot\_mbaff0\_full\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cavlc\_mot\_mbaff0\_full\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cavlc\_mot\_mbaff0\_full\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 60: fate-h264-conformance-cavlc\_mot\_picaff0\_full\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cavlc\_mot\_picaff0\_full\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cavlc\_mot\_picaff0\_full\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 61: fate-h264-conformance-cawp1\_toshiba\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cawp1\_toshiba\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cawp1\_toshiba\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 62: fate-h264-conformance-cawp5\_toshiba\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cawp5\_toshiba\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cawp5\_toshiba\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 63: fate-h264-conformance-ci1\_ft\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-ci1\_ft\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-ci1\_ft\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 64: fate-h264-conformance-ci\_mw\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-ci\_mw\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-ci\_mw\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 65: fate-h264-conformance-cvbs3\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvbs3\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvbs3\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 66: fate-h264-conformance-cvcanlma2\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvcanlma2\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvcanlma2\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 67: fate-h264-conformance-cvfc1\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvfc1\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvfc1\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 68: fate-h264-conformance-cvfi1\_sony\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvfi1\_sony\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvfi1\_sony\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 69: fate-h264-conformance-cvfi1\_sva\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvfi1\_sva\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvfi1\_sva\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 70: fate-h264-conformance-cvfi2\_sony\_h

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvfi2\_sony\_h SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvfi2\_sony\_h

```

The ARM platform is the same as the RISCV platform.

### Test case 71: fate-h264-conformance-cvfi2\_sva\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvfi2\_sva\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvfi2\_sva\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 72: fate-h264-conformance-cvma1\_sony\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvma1\_sony\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvma1\_sony\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 73: fate-h264-conformance-cvma1\_toshiba\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvma1\_toshiba\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvma1\_toshiba\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 74: fate-h264-conformance-cvmanl1\_toshiba\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvmanl1\_toshiba\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvmanl1\_toshiba\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 75: fate-h264-conformance-cvmanl2\_toshiba\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvmanl2\_toshiba\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvmanl2\_toshiba\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 76: fate-h264-conformance-cvmapaqp3\_sony\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvmapaqp3\_sony\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvmapaqp3\_sony\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 77: fate-h264-conformance-cvmaqp2\_sony\_g

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvmaqp2\_sony\_g SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvmaqp2\_sony\_g

```

The ARM platform is the same as the RISCV platform.

### Test case 78: fate-h264-conformance-cvmaqp3\_sony\_d

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvmaqp3\_sony\_d SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvmaqp3\_sony\_d

```

The ARM platform is the same as the RISCV platform.

### Test case 79: fate-h264-conformance-cvmp\_mot\_fld\_l30\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvmp\_mot\_fld\_l30\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvmp\_mot\_fld\_l30\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 80: fate-h264-conformance-cvmp\_mot\_frm\_l31\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvmp\_mot\_frm\_l31\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvmp\_mot\_frm\_l31\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 81: fate-h264-conformance-cvnlfi1\_sony\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvnlfi1\_sony\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvnlfi1\_sony\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 82: fate-h264-conformance-cvnlfi2\_sony\_h

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvnlfi2\_sony\_h SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvnlfi2\_sony\_h

```

The ARM platform is the same as the RISCV platform.

### Test case 83: fate-h264-conformance-cvpa1\_toshiba\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvpa1\_toshiba\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvpa1\_toshiba\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 84: fate-h264-conformance-cvpcmnl1\_sva\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvpcmnl1\_sva\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvpcmnl1\_sva\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 85: fate-h264-conformance-cvpcmnl2\_sva\_c

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvpcmnl2\_sva\_c SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvpcmnl2\_sva\_c

```

The ARM platform is the same as the RISCV platform.

### Test case 86: fate-h264-conformance-cvwp1\_toshiba\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvwp1\_toshiba\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvwp1\_toshiba\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 87: fate-h264-conformance-cvwp2\_toshiba\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvwp2\_toshiba\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvwp2\_toshiba\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 88: fate-h264-conformance-cvwp3\_toshiba\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvwp3\_toshiba\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvwp3\_toshiba\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 89: fate-h264-conformance-cvwp5\_toshiba\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-cvwp5\_toshiba\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-cvwp5\_toshiba\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 90: fate-h264-conformance-fi1\_sony\_e

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-fi1\_sony\_e SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-fi1\_sony\_e

```

The ARM platform is the same as the RISCV platform.

### Test case 91: fate-h264-conformance-frext-alphaconformanceg

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-alphaconformanceg SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-alphaconformanceg

```

The ARM platform is the same as the RISCV platform.

### Test case 92: fate-h264-conformance-frext-bcrm\_freh10

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-bcrm\_freh10 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-bcrm\_freh10

```

The ARM platform is the same as the RISCV platform.

### Test case 93: fate-h264-conformance-frext-brcm\_freh11

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-brcm\_freh11 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-brcm\_freh11

```

The ARM platform is the same as the RISCV platform.

### Test case 94: fate-h264-conformance-frext-brcm\_freh3

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-brcm\_freh3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-brcm\_freh3

```

The ARM platform is the same as the RISCV platform.

### Test case 95: fate-h264-conformance-frext-brcm\_freh4

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-brcm\_freh4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-brcm\_freh4

```

The ARM platform is the same as the RISCV platform.

### Test case 96: fate-h264-conformance-frext-brcm\_freh5

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-brcm\_freh5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-brcm\_freh5

```

The ARM platform is the same as the RISCV platform.

### Test case 97: fate-h264-conformance-frext-brcm\_freh8

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-brcm\_freh8 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-brcm\_freh8

```

The ARM platform is the same as the RISCV platform.

### Test case 98: fate-h264-conformance-frext-brcm\_freh9

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-brcm\_freh9 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-brcm\_freh9

```

The ARM platform is the same as the RISCV platform.

### Test case 99: fate-h264-conformance-frext-freh12\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-freh12\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-freh12\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 100: fate-h264-conformance-frext-freh1\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-freh1\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-freh1\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 101: fate-h264-conformance-frext-freh2\_b

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-freh2\_b SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-freh2\_b

```

The ARM platform is the same as the RISCV platform.

### Test case 102: fate-h264-conformance-frext-freh6

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-conformance-frext-freh6 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-conformance-frext-freh6

```

The ARM platform is the same as the RISCV platform.

### Test case 103: fate-h264-reinit-large\_420\_8-to-small\_420\_8

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-reinit-large\_420\_8-to-small\_420\_8 SAMPLES=fate\_samples/

```

The running results are:

```sh

HOSTCC tests/base64.o

HOSTLD tests/base64

HOSTCC tests/tiny\_psnr.o

HOSTLD tests/tiny\_psnr

HOSTCC tests/tiny\_ssim.o

HOSTLD tests/tiny\_ssim

HOSTCC tests/audiomatch.o

HOSTLD tests/audiomatch

TEST h264-reinit-large\_420\_8-to-small\_420\_8

```

The ARM platform is the same as the RISCV platform.

### Test case 104: fate-h264-reinit-small\_420\_8-to-large\_444\_10

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-reinit-small\_420\_8-to-large\_444\_10 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-reinit-small\_420\_8-to-large\_444\_10

```

The ARM platform is the same as the RISCV platform.

### Test case 105: fate-h264-reinit-small\_420\_9-to-small\_420\_8

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-reinit-small\_420\_9-to-small\_420\_8 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-reinit-small\_420\_9-to-small\_420\_8

```

The ARM platform is the same as the RISCV platform.

### Test case 106: fate-h264-reinit-small\_422\_9-to-small\_420\_9

After executing `build.sh`, enter the `ffmpeg-7.0.1` directory and run the test case:

```sh

make fate-h264-reinit-small\_422\_9-to-small\_420\_9 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST h264-reinit-small\_422\_9-to-small\_420\_9

```

The ARM platform is the same as the RISCV platform.

## perf automated script test

Currently, perf's automated script supports testing all passing samples with one click.

In the current directory, instead of entering the directory of `ffmpeg-7.0.1` (that is, the script and ffmpeg folder are in the same directory), execute the test script:

```bash

./test.sh

```

### Test result file description

After executing the `./test.sh` script, many files and folders are generated.

First is `run\_result`, which stores the output of all successfully run tests, such as:

```

Warning: the sample format the decoder produced is planar (s16p). This example will output the first channel only.

Play the output audio file with the command:

ffplay -f s16le -ac 2 -ar 44100 4.mp4

```

This is the output of the executable file. Each test item is different, and some test items have very long output.

Then there is the folder `test\_result` folder, which saves all the results of the perf command output, such as:

```

# started on Tue Aug 20 23:05:40 2024

Performance counter stats for './exe/avio\_list\_dir ./exe':

8,406,770 ns duration\_time: # 1.120 G/sec

7.51 msec task-clock: # 0.893 CPUs utilized

7,606,253 cycles: # 1.013 GHz

6,055,388 instructions: # 0.80 insn per cycle

2,704,812 cache-references: # 360.288 M/sec

141,025 cache-misses: # 5.21% of all cache refs

1,355,112 branches: # 180.504 M/sec

47,621 branch-misses: # 3.51% of all branches

2,704,812 L1-dcache-loads: # 360.288 M/sec

141,025 L1-dcache-load-misses: # 5.21% of all L1-dcache accesses

12,072 LLC-load-misses: # 25.90% of all L1-icache accesses

46,610 LLC-loads: # 6.209 M/sec

0.008406770 seconds time elapsed

0.004263000 seconds user

0.004221000 seconds sys

```

This is also what will be printed on the terminal if you use the `perf xxxx....` command in the script.

At this point, all the tests are finished and the required performance data are in the various txt files in the folder.

### Result formatting

The test results are saved in a txt file. Here is another script to format the results of the txt file into an xlsx file to match the format in Tencent Documents.

First, open a virtual environment or use system python and make sure you have the following dependencies:

```sh

pip install pandas openpyxl

```

Then you need to select the corresponding regular expression according to the two situations, because the perf output data may be a number such as `2222120` without `,`, or it may be a number such as `2,222,120`, so you need to match them separately.

Then execute the script:

```sh

python xlsx.py

```

Then an xlsx file `result.xlsx` will be generated in the current directory, in which all the results are summarized and the corresponding data are counted.