# H265 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 H265 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-hevc-afd-tc-sei

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

```sh

make fate-hevc-afd-tc-sei SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-afd-tc-sei

```

The ARM platform is the same as the RISCV platform.

### Test case 2: fate-hevc-bsf-mp4toannexb

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

```sh

make fate-hevc-bsf-mp4toannexb SAMPLES=fate\_samples/

```

The running results are:

```sh

GEN tests/data/hevc-mp4.mov

TEST hevc-bsf-mp4toannexb

```

The ARM platform is the same as the RISCV platform.

### Test case 3: fate-hevc-cabac-tudepth

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

```sh

make fate-hevc-cabac-tudepth SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-cabac-tudepth

```

The ARM platform is the same as the RISCV platform.

### Test case 4: fate-hevc-conformance-ADJUST\_IPRED\_ANGLE\_A\_RExt\_Mitsubishi\_1

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

```sh

make fate-hevc-conformance-ADJUST\_IPRED\_ANGLE\_A\_RExt\_Mitsubishi\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-ADJUST\_IPRED\_ANGLE\_A\_RExt\_Mitsubishi\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 5: fate-hevc-conformance-AMP\_A\_Samsung\_4

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

```sh

make fate-hevc-conformance-AMP\_A\_Samsung\_4 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 hevc-conformance-AMP\_A\_Samsung\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 6: fate-hevc-conformance-AMP\_A\_Samsung\_6

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

```sh

make fate-hevc-conformance-AMP\_A\_Samsung\_6 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-AMP\_A\_Samsung\_6

```

The ARM platform is the same as the RISCV platform.

### Test case 7: fate-hevc-conformance-AMP\_B\_Samsung\_4

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

```sh

make fate-hevc-conformance-AMP\_B\_Samsung\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-AMP\_B\_Samsung\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 8: fate-hevc-conformance-AMP\_B\_Samsung\_6

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

```sh

make fate-hevc-conformance-AMP\_B\_Samsung\_6 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-AMP\_B\_Samsung\_6

```

The ARM platform is the same as the RISCV platform.

### Test case 9: fate-hevc-conformance-AMP\_D\_Hisilicon

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

```sh

make fate-hevc-conformance-AMP\_D\_Hisilicon SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-AMP\_D\_Hisilicon

```

The ARM platform is the same as the RISCV platform.

### Test case 10: fate-hevc-conformance-AMP\_E\_Hisilicon

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

```sh

make fate-hevc-conformance-AMP\_E\_Hisilicon SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-AMP\_E\_Hisilicon

```

The ARM platform is the same as the RISCV platform.

### Test case 11: fate-hevc-conformance-AMP\_F\_Hisilicon\_3

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

```sh

make fate-hevc-conformance-AMP\_F\_Hisilicon\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-AMP\_F\_Hisilicon\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 12: fate-hevc-conformance-AMVP\_A\_MTK\_4

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

```sh

make fate-hevc-conformance-AMVP\_A\_MTK\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-AMVP\_A\_MTK\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 13: fate-hevc-conformance-AMVP\_B\_MTK\_4

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

```sh

make fate-hevc-conformance-AMVP\_B\_MTK\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-AMVP\_B\_MTK\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 14: fate-hevc-conformance-AMVP\_C\_Samsung\_4

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

```sh

make fate-hevc-conformance-AMVP\_C\_Samsung\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-AMVP\_C\_Samsung\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 15: fate-hevc-conformance-AMVP\_C\_Samsung\_6

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

```sh

make fate-hevc-conformance-AMVP\_C\_Samsung\_6 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-AMVP\_C\_Samsung\_6

```

The ARM platform is the same as the RISCV platform.

### Test case 16: fate-hevc-conformance-BUMPING\_A\_ericsson\_1

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

```sh

make fate-hevc-conformance-BUMPING\_A\_ericsson\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-BUMPING\_A\_ericsson\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 17: fate-hevc-conformance-CAINIT\_A\_SHARP\_4

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

```sh

make fate-hevc-conformance-CAINIT\_A\_SHARP\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CAINIT\_A\_SHARP\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 18: fate-hevc-conformance-CAINIT\_B\_SHARP\_4

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

```sh

make fate-hevc-conformance-CAINIT\_B\_SHARP\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CAINIT\_B\_SHARP\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 19: fate-hevc-conformance-CAINIT\_C\_SHARP\_3

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

```sh

make fate-hevc-conformance-CAINIT\_C\_SHARP\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CAINIT\_C\_SHARP\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 20: fate-hevc-conformance-CAINIT\_D\_SHARP\_3

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

```sh

make fate-hevc-conformance-CAINIT\_D\_SHARP\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CAINIT\_D\_SHARP\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 21: fate-hevc-conformance-CAINIT\_E\_SHARP\_3

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

```sh

make fate-hevc-conformance-CAINIT\_E\_SHARP\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CAINIT\_E\_SHARP\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 22: fate-hevc-conformance-CAINIT\_F\_SHARP\_3

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

```sh

make fate-hevc-conformance-CAINIT\_F\_SHARP\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CAINIT\_F\_SHARP\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 23: fate-hevc-conformance-CAINIT\_G\_SHARP\_3

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

```sh

make fate-hevc-conformance-CAINIT\_G\_SHARP\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CAINIT\_G\_SHARP\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 24: fate-hevc-conformance-CAINIT\_H\_SHARP\_3

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

```sh

make fate-hevc-conformance-CAINIT\_H\_SHARP\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CAINIT\_H\_SHARP\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 25: fate-hevc-conformance-CIP\_A\_Panasonic\_3

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

```sh

make fate-hevc-conformance-CIP\_A\_Panasonic\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CIP\_A\_Panasonic\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 26: fate-hevc-conformance-CIP\_C\_Panasonic\_2

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

```sh

make fate-hevc-conformance-CIP\_C\_Panasonic\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CIP\_C\_Panasonic\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 27: fate-hevc-conformance-CONFWIN\_A\_Sony\_1

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

```sh

make fate-hevc-conformance-CONFWIN\_A\_Sony\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-CONFWIN\_A\_Sony\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 28: fate-hevc-conformance-DBLK\_A\_MAIN10\_VIXS\_3

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

```sh

make fate-hevc-conformance-DBLK\_A\_MAIN10\_VIXS\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DBLK\_A\_MAIN10\_VIXS\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 29: fate-hevc-conformance-DBLK\_A\_SONY\_3

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

```sh

make fate-hevc-conformance-DBLK\_A\_SONY\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DBLK\_A\_SONY\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 30: fate-hevc-conformance-DBLK\_B\_SONY\_3

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

```sh

make fate-hevc-conformance-DBLK\_B\_SONY\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DBLK\_B\_SONY\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 31: fate-hevc-conformance-DBLK\_C\_SONY\_3

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

```sh

make fate-hevc-conformance-DBLK\_C\_SONY\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DBLK\_C\_SONY\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 32: fate-hevc-conformance-DBLK\_D\_VIXS\_2

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

```sh

make fate-hevc-conformance-DBLK\_D\_VIXS\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DBLK\_D\_VIXS\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 33: fate-hevc-conformance-DBLK\_E\_VIXS\_2

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

```sh

make fate-hevc-conformance-DBLK\_E\_VIXS\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DBLK\_E\_VIXS\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 34: fate-hevc-conformance-DBLK\_F\_VIXS\_2

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

```sh

make fate-hevc-conformance-DBLK\_F\_VIXS\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DBLK\_F\_VIXS\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 35: fate-hevc-conformance-DBLK\_G\_VIXS\_2

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

```sh

make fate-hevc-conformance-DBLK\_G\_VIXS\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DBLK\_G\_VIXS\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 36: fate-hevc-conformance-DELTAQP\_A\_BRCM\_4

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

```sh

make fate-hevc-conformance-DELTAQP\_A\_BRCM\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DELTAQP\_A\_BRCM\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 37: fate-hevc-conformance-DELTAQP\_B\_SONY\_3

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

```sh

make fate-hevc-conformance-DELTAQP\_B\_SONY\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DELTAQP\_B\_SONY\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 38: fate-hevc-conformance-DELTAQP\_C\_SONY\_3

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

```sh

make fate-hevc-conformance-DELTAQP\_C\_SONY\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DELTAQP\_C\_SONY\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 39: fate-hevc-conformance-DSLICE\_A\_HHI\_5

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

```sh

make fate-hevc-conformance-DSLICE\_A\_HHI\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DSLICE\_A\_HHI\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 40: fate-hevc-conformance-DSLICE\_B\_HHI\_5

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

```sh

make fate-hevc-conformance-DSLICE\_B\_HHI\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DSLICE\_B\_HHI\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 41: fate-hevc-conformance-DSLICE\_C\_HHI\_5

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

```sh

make fate-hevc-conformance-DSLICE\_C\_HHI\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-DSLICE\_C\_HHI\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 42: fate-hevc-conformance-ENTP\_A\_Qualcomm\_1

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

```sh

make fate-hevc-conformance-ENTP\_A\_Qualcomm\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-ENTP\_A\_Qualcomm\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 43: fate-hevc-conformance-ENTP\_B\_Qualcomm\_1

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

```sh

make fate-hevc-conformance-ENTP\_B\_Qualcomm\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-ENTP\_B\_Qualcomm\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 44: fate-hevc-conformance-ENTP\_C\_Qualcomm\_1

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

```sh

make fate-hevc-conformance-ENTP\_C\_Qualcomm\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-ENTP\_C\_Qualcomm\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 45: fate-hevc-conformance-EXT\_A\_ericsson\_4

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

```sh

make fate-hevc-conformance-EXT\_A\_ericsson\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-EXT\_A\_ericsson\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 46: fate-hevc-conformance-FILLER\_A\_Sony\_1

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

```sh

make fate-hevc-conformance-FILLER\_A\_Sony\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-FILLER\_A\_Sony\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 47: fate-hevc-conformance-HRD\_A\_Fujitsu\_2

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

```sh

make fate-hevc-conformance-HRD\_A\_Fujitsu\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-HRD\_A\_Fujitsu\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 48: fate-hevc-conformance-HRD\_A\_Fujitsu\_3

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

```sh

make fate-hevc-conformance-HRD\_A\_Fujitsu\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-HRD\_A\_Fujitsu\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 49: fate-hevc-conformance-INITQP\_A\_Sony\_1

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

```sh

make fate-hevc-conformance-INITQP\_A\_Sony\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-INITQP\_A\_Sony\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 50: fate-hevc-conformance-INITQP\_B\_Sony\_1

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

```sh

make fate-hevc-conformance-INITQP\_B\_Sony\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-INITQP\_B\_Sony\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 51: fate-hevc-conformance-IPCM\_A\_RExt\_NEC

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

```sh

make fate-hevc-conformance-IPCM\_A\_RExt\_NEC SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-IPCM\_A\_RExt\_NEC

```

The ARM platform is the same as the RISCV platform.

### Test case 52: fate-hevc-conformance-IPCM\_B\_RExt\_NEC

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

```sh

make fate-hevc-conformance-IPCM\_B\_RExt\_NEC SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-IPCM\_B\_RExt\_NEC

```

The ARM platform is the same as the RISCV platform.

### Test case 53: fate-hevc-conformance-IPRED\_A\_docomo\_2

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

```sh

make fate-hevc-conformance-IPRED\_A\_docomo\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-IPRED\_A\_docomo\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 54: fate-hevc-conformance-IPRED\_B\_Nokia\_3

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

```sh

make fate-hevc-conformance-IPRED\_B\_Nokia\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-IPRED\_B\_Nokia\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 55: fate-hevc-conformance-IPRED\_C\_Mitsubishi\_3

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

```sh

make fate-hevc-conformance-IPRED\_C\_Mitsubishi\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-IPRED\_C\_Mitsubishi\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 56: fate-hevc-conformance-LS\_A\_Orange\_2

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

```sh

make fate-hevc-conformance-LS\_A\_Orange\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-LS\_A\_Orange\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 57: fate-hevc-conformance-LS\_B\_ORANGE\_4

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

```sh

make fate-hevc-conformance-LS\_B\_ORANGE\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-LS\_B\_ORANGE\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 58: fate-hevc-conformance-LTRPSPS\_A\_Qualcomm\_1

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

```sh

make fate-hevc-conformance-LTRPSPS\_A\_Qualcomm\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-LTRPSPS\_A\_Qualcomm\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 59: fate-hevc-conformance-MAXBINS\_A\_TI\_4

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

```sh

make fate-hevc-conformance-MAXBINS\_A\_TI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MAXBINS\_A\_TI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 60: fate-hevc-conformance-MAXBINS\_B\_TI\_4

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

```sh

make fate-hevc-conformance-MAXBINS\_B\_TI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MAXBINS\_B\_TI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 61: fate-hevc-conformance-MAXBINS\_C\_TI\_4

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

```sh

make fate-hevc-conformance-MAXBINS\_C\_TI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MAXBINS\_C\_TI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 62: fate-hevc-conformance-MERGE\_A\_TI\_3

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

```sh

make fate-hevc-conformance-MERGE\_A\_TI\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MERGE\_A\_TI\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 63: fate-hevc-conformance-MERGE\_B\_TI\_3

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

```sh

make fate-hevc-conformance-MERGE\_B\_TI\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MERGE\_B\_TI\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 64: fate-hevc-conformance-MERGE\_C\_TI\_3

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

```sh

make fate-hevc-conformance-MERGE\_C\_TI\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MERGE\_C\_TI\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 65: fate-hevc-conformance-MERGE\_D\_TI\_3

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

```sh

make fate-hevc-conformance-MERGE\_D\_TI\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MERGE\_D\_TI\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 66: fate-hevc-conformance-MERGE\_E\_TI\_3

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

```sh

make fate-hevc-conformance-MERGE\_E\_TI\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MERGE\_E\_TI\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 67: fate-hevc-conformance-MERGE\_F\_MTK\_4

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

```sh

make fate-hevc-conformance-MERGE\_F\_MTK\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MERGE\_F\_MTK\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 68: fate-hevc-conformance-MERGE\_G\_HHI\_4

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

```sh

make fate-hevc-conformance-MERGE\_G\_HHI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MERGE\_G\_HHI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 69: fate-hevc-conformance-MVCLIP\_A\_qualcomm\_3

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

```sh

make fate-hevc-conformance-MVCLIP\_A\_qualcomm\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MVCLIP\_A\_qualcomm\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 70: fate-hevc-conformance-MVDL1ZERO\_A\_docomo\_3

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

```sh

make fate-hevc-conformance-MVDL1ZERO\_A\_docomo\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MVDL1ZERO\_A\_docomo\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 71: fate-hevc-conformance-MVEDGE\_A\_qualcomm\_3

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

```sh

make fate-hevc-conformance-MVEDGE\_A\_qualcomm\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-MVEDGE\_A\_qualcomm\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 72: fate-hevc-conformance-Main\_422\_10\_A\_RExt\_Sony\_1

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

```sh

make fate-hevc-conformance-Main\_422\_10\_A\_RExt\_Sony\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-Main\_422\_10\_A\_RExt\_Sony\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 73: fate-hevc-conformance-Main\_422\_10\_B\_RExt\_Sony\_1

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

```sh

make fate-hevc-conformance-Main\_422\_10\_B\_RExt\_Sony\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-Main\_422\_10\_B\_RExt\_Sony\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 74: fate-hevc-conformance-NUT\_A\_ericsson\_5

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

```sh

make fate-hevc-conformance-NUT\_A\_ericsson\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-NUT\_A\_ericsson\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 75: fate-hevc-conformance-NoOutPrior\_A\_Qualcomm\_1

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

```sh

make fate-hevc-conformance-NoOutPrior\_A\_Qualcomm\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-NoOutPrior\_A\_Qualcomm\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 76: fate-hevc-conformance-NoOutPrior\_B\_Qualcomm\_1

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

```sh

make fate-hevc-conformance-NoOutPrior\_B\_Qualcomm\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-NoOutPrior\_B\_Qualcomm\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 77: fate-hevc-conformance-OPFLAG\_A\_Qualcomm\_1

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

```sh

make fate-hevc-conformance-OPFLAG\_A\_Qualcomm\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-OPFLAG\_A\_Qualcomm\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 78: fate-hevc-conformance-OPFLAG\_B\_Qualcomm\_1

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

```sh

make fate-hevc-conformance-OPFLAG\_B\_Qualcomm\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-OPFLAG\_B\_Qualcomm\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 79: fate-hevc-conformance-OPFLAG\_C\_Qualcomm\_1

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

```sh

make fate-hevc-conformance-OPFLAG\_C\_Qualcomm\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-OPFLAG\_C\_Qualcomm\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 80: fate-hevc-conformance-PERSIST\_RPARAM\_A\_RExt\_Sony\_1

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

```sh

make fate-hevc-conformance-PERSIST\_RPARAM\_A\_RExt\_Sony\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PERSIST\_RPARAM\_A\_RExt\_Sony\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 81: fate-hevc-conformance-PERSIST\_RPARAM\_A\_RExt\_Sony\_3

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

```sh

make fate-hevc-conformance-PERSIST\_RPARAM\_A\_RExt\_Sony\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PERSIST\_RPARAM\_A\_RExt\_Sony\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 82: fate-hevc-conformance-PICSIZE\_A\_Bossen\_1

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

```sh

make fate-hevc-conformance-PICSIZE\_A\_Bossen\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PICSIZE\_A\_Bossen\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 83: fate-hevc-conformance-PICSIZE\_B\_Bossen\_1

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

```sh

make fate-hevc-conformance-PICSIZE\_B\_Bossen\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PICSIZE\_B\_Bossen\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 84: fate-hevc-conformance-PICSIZE\_C\_Bossen\_1

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

```sh

make fate-hevc-conformance-PICSIZE\_C\_Bossen\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PICSIZE\_C\_Bossen\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 85: fate-hevc-conformance-PICSIZE\_D\_Bossen\_1

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

```sh

make fate-hevc-conformance-PICSIZE\_D\_Bossen\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PICSIZE\_D\_Bossen\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 86: fate-hevc-conformance-PMERGE\_A\_TI\_3

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

```sh

make fate-hevc-conformance-PMERGE\_A\_TI\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PMERGE\_A\_TI\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 87: fate-hevc-conformance-PMERGE\_B\_TI\_3

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

```sh

make fate-hevc-conformance-PMERGE\_B\_TI\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PMERGE\_B\_TI\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 88: fate-hevc-conformance-PMERGE\_C\_TI\_3

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

```sh

make fate-hevc-conformance-PMERGE\_C\_TI\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PMERGE\_C\_TI\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 89: fate-hevc-conformance-PMERGE\_D\_TI\_3

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

```sh

make fate-hevc-conformance-PMERGE\_D\_TI\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PMERGE\_D\_TI\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 90: fate-hevc-conformance-PMERGE\_E\_TI\_3

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

```sh

make fate-hevc-conformance-PMERGE\_E\_TI\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PMERGE\_E\_TI\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 91: fate-hevc-conformance-POC\_A\_Bossen\_3

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

```sh

make fate-hevc-conformance-POC\_A\_Bossen\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-POC\_A\_Bossen\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 92: fate-hevc-conformance-PPS\_A\_qualcomm\_7

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

```sh

make fate-hevc-conformance-PPS\_A\_qualcomm\_7 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PPS\_A\_qualcomm\_7

```

The ARM platform is the same as the RISCV platform.

### Test case 93: fate-hevc-conformance-PS\_A\_VIDYO\_3

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

```sh

make fate-hevc-conformance-PS\_A\_VIDYO\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PS\_A\_VIDYO\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 94: fate-hevc-conformance-PS\_B\_VIDYO\_3

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

```sh

make fate-hevc-conformance-PS\_B\_VIDYO\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-PS\_B\_VIDYO\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 95: fate-hevc-conformance-QMATRIX\_A\_RExt\_Sony\_1

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

```sh

make fate-hevc-conformance-QMATRIX\_A\_RExt\_Sony\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-QMATRIX\_A\_RExt\_Sony\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 96: fate-hevc-conformance-RAP\_A\_docomo\_4

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

```sh

make fate-hevc-conformance-RAP\_A\_docomo\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RAP\_A\_docomo\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 97: fate-hevc-conformance-RAP\_B\_Bossen\_1

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

```sh

make fate-hevc-conformance-RAP\_B\_Bossen\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RAP\_B\_Bossen\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 98: fate-hevc-conformance-RPLM\_A\_qualcomm\_4

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

```sh

make fate-hevc-conformance-RPLM\_A\_qualcomm\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RPLM\_A\_qualcomm\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 99: fate-hevc-conformance-RPLM\_B\_qualcomm\_4

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

```sh

make fate-hevc-conformance-RPLM\_B\_qualcomm\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RPLM\_B\_qualcomm\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 100: fate-hevc-conformance-RPS\_A\_docomo\_4

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

```sh

make fate-hevc-conformance-RPS\_A\_docomo\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RPS\_A\_docomo\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 101: fate-hevc-conformance-RPS\_B\_qualcomm\_5

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

```sh

make fate-hevc-conformance-RPS\_B\_qualcomm\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RPS\_B\_qualcomm\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 102: fate-hevc-conformance-RPS\_C\_ericsson\_5

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

```sh

make fate-hevc-conformance-RPS\_C\_ericsson\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RPS\_C\_ericsson\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 103: fate-hevc-conformance-RPS\_D\_ericsson\_6

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

```sh

make fate-hevc-conformance-RPS\_D\_ericsson\_6 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RPS\_D\_ericsson\_6

```

The ARM platform is the same as the RISCV platform.

### Test case 104: fate-hevc-conformance-RPS\_E\_qualcomm\_5

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

```sh

make fate-hevc-conformance-RPS\_E\_qualcomm\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RPS\_E\_qualcomm\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 105: fate-hevc-conformance-RPS\_F\_docomo\_1

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

```sh

make fate-hevc-conformance-RPS\_F\_docomo\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RPS\_F\_docomo\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 106: fate-hevc-conformance-RQT\_A\_HHI\_4

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

```sh

make fate-hevc-conformance-RQT\_A\_HHI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RQT\_A\_HHI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 107: fate-hevc-conformance-RQT\_B\_HHI\_4

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

```sh

make fate-hevc-conformance-RQT\_B\_HHI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RQT\_B\_HHI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 108: fate-hevc-conformance-RQT\_C\_HHI\_4

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

```sh

make fate-hevc-conformance-RQT\_C\_HHI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RQT\_C\_HHI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 109: fate-hevc-conformance-RQT\_D\_HHI\_4

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

```sh

make fate-hevc-conformance-RQT\_D\_HHI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RQT\_D\_HHI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 110: fate-hevc-conformance-RQT\_E\_HHI\_4

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

```sh

make fate-hevc-conformance-RQT\_E\_HHI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RQT\_E\_HHI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 111: fate-hevc-conformance-RQT\_F\_HHI\_4

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

```sh

make fate-hevc-conformance-RQT\_F\_HHI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RQT\_F\_HHI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 112: fate-hevc-conformance-RQT\_G\_HHI\_4

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

```sh

make fate-hevc-conformance-RQT\_G\_HHI\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-RQT\_G\_HHI\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 113: fate-hevc-conformance-SAO\_A\_MediaTek\_4

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

```sh

make fate-hevc-conformance-SAO\_A\_MediaTek\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SAO\_A\_MediaTek\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 114: fate-hevc-conformance-SAO\_A\_RExt\_MediaTek\_1

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

```sh

make fate-hevc-conformance-SAO\_A\_RExt\_MediaTek\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SAO\_A\_RExt\_MediaTek\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 115: fate-hevc-conformance-SAO\_B\_MediaTek\_5

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

```sh

make fate-hevc-conformance-SAO\_B\_MediaTek\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SAO\_B\_MediaTek\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 116: fate-hevc-conformance-SAO\_C\_Samsung\_4

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

```sh

make fate-hevc-conformance-SAO\_C\_Samsung\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SAO\_C\_Samsung\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 117: fate-hevc-conformance-SAO\_C\_Samsung\_5

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

```sh

make fate-hevc-conformance-SAO\_C\_Samsung\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SAO\_C\_Samsung\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 118: fate-hevc-conformance-SAO\_D\_Samsung\_4

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

```sh

make fate-hevc-conformance-SAO\_D\_Samsung\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SAO\_D\_Samsung\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 119: fate-hevc-conformance-SAO\_D\_Samsung\_5

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

```sh

make fate-hevc-conformance-SAO\_D\_Samsung\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SAO\_D\_Samsung\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 120: fate-hevc-conformance-SAO\_E\_Canon\_4

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

```sh

make fate-hevc-conformance-SAO\_E\_Canon\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SAO\_E\_Canon\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 121: fate-hevc-conformance-SAO\_F\_Canon\_3

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

```sh

make fate-hevc-conformance-SAO\_F\_Canon\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SAO\_F\_Canon\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 122: fate-hevc-conformance-SAO\_G\_Canon\_3

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

```sh

make fate-hevc-conformance-SAO\_G\_Canon\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SAO\_G\_Canon\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 123: fate-hevc-conformance-SDH\_A\_Orange\_3

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

```sh

make fate-hevc-conformance-SDH\_A\_Orange\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SDH\_A\_Orange\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 124: fate-hevc-conformance-SLICES\_A\_Rovi\_3

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

```sh

make fate-hevc-conformance-SLICES\_A\_Rovi\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SLICES\_A\_Rovi\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 125: fate-hevc-conformance-SLIST\_A\_Sony\_4

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

```sh

make fate-hevc-conformance-SLIST\_A\_Sony\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SLIST\_A\_Sony\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 126: fate-hevc-conformance-SLIST\_B\_Sony\_8

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

```sh

make fate-hevc-conformance-SLIST\_B\_Sony\_8 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SLIST\_B\_Sony\_8

```

The ARM platform is the same as the RISCV platform.

### Test case 127: fate-hevc-conformance-SLIST\_C\_Sony\_3

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

```sh

make fate-hevc-conformance-SLIST\_C\_Sony\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SLIST\_C\_Sony\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 128: fate-hevc-conformance-SLIST\_D\_Sony\_9

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

```sh

make fate-hevc-conformance-SLIST\_D\_Sony\_9 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SLIST\_D\_Sony\_9

```

The ARM platform is the same as the RISCV platform.

### Test case 129: fate-hevc-conformance-SLPPLP\_A\_VIDYO\_1

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

```sh

make fate-hevc-conformance-SLPPLP\_A\_VIDYO\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SLPPLP\_A\_VIDYO\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 130: fate-hevc-conformance-SLPPLP\_A\_VIDYO\_2

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

```sh

make fate-hevc-conformance-SLPPLP\_A\_VIDYO\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-SLPPLP\_A\_VIDYO\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 131: fate-hevc-conformance-STRUCT\_A\_Samsung\_5

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

```sh

make fate-hevc-conformance-STRUCT\_A\_Samsung\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-STRUCT\_A\_Samsung\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 132: fate-hevc-conformance-STRUCT\_B\_Samsung\_4

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

```sh

make fate-hevc-conformance-STRUCT\_B\_Samsung\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-STRUCT\_B\_Samsung\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 133: fate-hevc-conformance-STRUCT\_B\_Samsung\_6

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

```sh

make fate-hevc-conformance-STRUCT\_B\_Samsung\_6 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-STRUCT\_B\_Samsung\_6

```

The ARM platform is the same as the RISCV platform.

### Test case 134: fate-hevc-conformance-TILES\_A\_Cisco\_2

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

```sh

make fate-hevc-conformance-TILES\_A\_Cisco\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-TILES\_A\_Cisco\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 135: fate-hevc-conformance-TILES\_B\_Cisco\_1

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

```sh

make fate-hevc-conformance-TILES\_B\_Cisco\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-TILES\_B\_Cisco\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 136: fate-hevc-conformance-TMVP\_A\_MS\_3

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

```sh

make fate-hevc-conformance-TMVP\_A\_MS\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-TMVP\_A\_MS\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 137: fate-hevc-conformance-TSCL\_A\_VIDYO\_5

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

```sh

make fate-hevc-conformance-TSCL\_A\_VIDYO\_5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-TSCL\_A\_VIDYO\_5

```

The ARM platform is the same as the RISCV platform.

### Test case 138: fate-hevc-conformance-TSCL\_B\_VIDYO\_4

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

```sh

make fate-hevc-conformance-TSCL\_B\_VIDYO\_4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-TSCL\_B\_VIDYO\_4

```

The ARM platform is the same as the RISCV platform.

### Test case 139: fate-hevc-conformance-TSKIP\_A\_MS\_3

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

```sh

make fate-hevc-conformance-TSKIP\_A\_MS\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-TSKIP\_A\_MS\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 140: fate-hevc-conformance-TUSIZE\_A\_Samsung\_1

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

```sh

make fate-hevc-conformance-TUSIZE\_A\_Samsung\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-TUSIZE\_A\_Samsung\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 141: fate-hevc-conformance-VPSID\_A\_VIDYO\_1

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

```sh

make fate-hevc-conformance-VPSID\_A\_VIDYO\_1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-VPSID\_A\_VIDYO\_1

```

The ARM platform is the same as the RISCV platform.

### Test case 142: fate-hevc-conformance-VPSID\_A\_VIDYO\_2

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

```sh

make fate-hevc-conformance-VPSID\_A\_VIDYO\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-VPSID\_A\_VIDYO\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 143: fate-hevc-conformance-WPP\_A\_ericsson\_MAIN10\_2

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

```sh

make fate-hevc-conformance-WPP\_A\_ericsson\_MAIN10\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_A\_ericsson\_MAIN10\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 144: fate-hevc-conformance-WPP\_A\_ericsson\_MAIN\_2

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

```sh

make fate-hevc-conformance-WPP\_A\_ericsson\_MAIN\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_A\_ericsson\_MAIN\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 145: fate-hevc-conformance-WPP\_B\_ericsson\_MAIN10\_2

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

```sh

make fate-hevc-conformance-WPP\_B\_ericsson\_MAIN10\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_B\_ericsson\_MAIN10\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 146: fate-hevc-conformance-WPP\_B\_ericsson\_MAIN\_2

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

```sh

make fate-hevc-conformance-WPP\_B\_ericsson\_MAIN\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_B\_ericsson\_MAIN\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 147: fate-hevc-conformance-WPP\_C\_ericsson\_MAIN10\_2

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

```sh

make fate-hevc-conformance-WPP\_C\_ericsson\_MAIN10\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_C\_ericsson\_MAIN10\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 148: fate-hevc-conformance-WPP\_C\_ericsson\_MAIN\_2

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

```sh

make fate-hevc-conformance-WPP\_C\_ericsson\_MAIN\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_C\_ericsson\_MAIN\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 149: fate-hevc-conformance-WPP\_D\_ericsson\_MAIN10\_2

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

```sh

make fate-hevc-conformance-WPP\_D\_ericsson\_MAIN10\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_D\_ericsson\_MAIN10\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 150: fate-hevc-conformance-WPP\_D\_ericsson\_MAIN\_2

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

```sh

make fate-hevc-conformance-WPP\_D\_ericsson\_MAIN\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_D\_ericsson\_MAIN\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 151: fate-hevc-conformance-WPP\_E\_ericsson\_MAIN10\_2

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

```sh

make fate-hevc-conformance-WPP\_E\_ericsson\_MAIN10\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_E\_ericsson\_MAIN10\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 152: fate-hevc-conformance-WPP\_E\_ericsson\_MAIN\_2

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

```sh

make fate-hevc-conformance-WPP\_E\_ericsson\_MAIN\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_E\_ericsson\_MAIN\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 153: fate-hevc-conformance-WPP\_F\_ericsson\_MAIN10\_2

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

```sh

make fate-hevc-conformance-WPP\_F\_ericsson\_MAIN10\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_F\_ericsson\_MAIN10\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 154: fate-hevc-conformance-WPP\_F\_ericsson\_MAIN\_2

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

```sh

make fate-hevc-conformance-WPP\_F\_ericsson\_MAIN\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_F\_ericsson\_MAIN\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 155: fate-hevc-conformance-WPP\_HIGH\_TP\_444\_8BIT\_RExt\_Apple\_2

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

```sh

make fate-hevc-conformance-WPP\_HIGH\_TP\_444\_8BIT\_RExt\_Apple\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WPP\_HIGH\_TP\_444\_8BIT\_RExt\_Apple\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 156: fate-hevc-conformance-WP\_A\_MAIN10\_Toshiba\_3

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

```sh

make fate-hevc-conformance-WP\_A\_MAIN10\_Toshiba\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WP\_A\_MAIN10\_Toshiba\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 157: fate-hevc-conformance-WP\_A\_Toshiba\_3

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

```sh

make fate-hevc-conformance-WP\_A\_Toshiba\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WP\_A\_Toshiba\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 158: fate-hevc-conformance-WP\_B\_Toshiba\_3

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

```sh

make fate-hevc-conformance-WP\_B\_Toshiba\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WP\_B\_Toshiba\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 159: fate-hevc-conformance-WP\_MAIN10\_B\_Toshiba\_3

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

```sh

make fate-hevc-conformance-WP\_MAIN10\_B\_Toshiba\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-WP\_MAIN10\_B\_Toshiba\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 160: fate-hevc-conformance-cip\_B\_NEC\_3

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

```sh

make fate-hevc-conformance-cip\_B\_NEC\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-cip\_B\_NEC\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 161: fate-hevc-conformance-ipcm\_A\_NEC\_3

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

```sh

make fate-hevc-conformance-ipcm\_A\_NEC\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-ipcm\_A\_NEC\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 162: fate-hevc-conformance-ipcm\_B\_NEC\_3

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

```sh

make fate-hevc-conformance-ipcm\_B\_NEC\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-ipcm\_B\_NEC\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 163: fate-hevc-conformance-ipcm\_C\_NEC\_3

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

```sh

make fate-hevc-conformance-ipcm\_C\_NEC\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-ipcm\_C\_NEC\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 164: fate-hevc-conformance-ipcm\_D\_NEC\_3

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

```sh

make fate-hevc-conformance-ipcm\_D\_NEC\_3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-ipcm\_D\_NEC\_3

```

The ARM platform is the same as the RISCV platform.

### Test case 165: fate-hevc-conformance-ipcm\_E\_NEC\_2

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

```sh

make fate-hevc-conformance-ipcm\_E\_NEC\_2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-conformance-ipcm\_E\_NEC\_2

```

The ARM platform is the same as the RISCV platform.

### Test case 166: fate-hevc-dv-rpu

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

```sh

make fate-hevc-dv-rpu SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-dv-rpu

```

The ARM platform is the same as the RISCV platform.

### Test case 167: fate-hevc-extradata-reload

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

```sh

make fate-hevc-extradata-reload SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-extradata-reload

```

The ARM platform is the same as the RISCV platform.

### Test case 168: fate-hevc-hdr-vivid-metadata

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

```sh

make fate-hevc-hdr-vivid-metadata SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-hdr-vivid-metadata

```

The ARM platform is the same as the RISCV platform.

### Test case 169: fate-hevc-hdr10-plus-metadata

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

```sh

make fate-hevc-hdr10-plus-metadata SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-hdr10-plus-metadata

```

The ARM platform is the same as the RISCV platform.

### Test case 170: fate-hevc-monochrome-crop

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

```sh

make fate-hevc-monochrome-crop SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-monochrome-crop

```

The ARM platform is the same as the RISCV platform.

### Test case 171: fate-hevc-paired-fields

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

```sh

make fate-hevc-paired-fields SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-paired-fields

```

The ARM platform is the same as the RISCV platform.

### Test case 172: fate-hevc-paramchange-yuv420p-yuv420p10

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

```sh

make fate-hevc-paramchange-yuv420p-yuv420p10 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-paramchange-yuv420p-yuv420p10

```

The ARM platform is the same as the RISCV platform.

### Test case 173: fate-hevc-skiploopfilter

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

```sh

make fate-hevc-skiploopfilter SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-skiploopfilter

```

The ARM platform is the same as the RISCV platform.

### Test case 174: fate-hevc-small422chroma

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

```sh

make fate-hevc-small422chroma SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-small422chroma

```

The ARM platform is the same as the RISCV platform.

### Test case 175: fate-hevc-two-first-slice

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

```sh

make fate-hevc-two-first-slice SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hevc-two-first-slice

```

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.