# FFmpeg 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

## Test-build executable file

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

```sh

./build.sh

```

## Test item-example

### Test case 1: avio\_http\_serve\_files

Currently, there is a problem with this test case. It requires a network connection and needs to open ports such as `http:localhost:9941`. At the same time, another terminal is required to connect to the server. However, using localhost to test will result in 404. The reason is currently unknown.

### Test case 2: avio\_list\_dir

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

./avio\_list\_dir exe

```

The running results are:

```sh

TYPE SIZE NAME UID(GID) UGO MODIFIED ACCESSED STATUS\_CHANGED

<FILE> 127752 decode\_filter\_audio 1018(1018) 755 1724068186000000 1724069222000000 1724068186000000

<FILE> 129144 mux 1018(1018) 755 1724068229000000 1724071108000000 1724068229000000

<FILE> 114016 encode\_audio 1018(1018) 755 1724068208000000 1724070239000000 1724068208000000

<FILE> 124832 demux\_decode 1018(1018) 755 1724068203000000 1724070022000000 1724068203000000

<FILE> 113736 remux 1018(1018) 755 1724068234000000 1724071192000000 1724068234000000

........omitted........

```

The ARM platform is the same as the RISCV platform.

### Test case 3: avio\_read\_callback

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./avio\_read\_callback temp.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

ptr:0xffff89800000 size:68509954

ptr:0xffff89801000 size:68505858

ptr:0xffff89802000 size:68501762

ptr:0xffff89803157 size:68497323

ptr:0xffff89804157 size:68493227

ptr:0xffff898274eb size:68348951

ptr:0xffff89843eae size:68231764

ptr:0xffff89862b69 size:68105625

ptr:0xffff8987d0bb size:67997767

ptr:0xffff89894df1 size:67900177

ptr:0xffff898aea52 size:67794608

ptr:0xffff898c8d7b size:67687303

ptr:0xffff898decfa size:67597320

ptr:0xffff898f87be size:67492164

ptr:0xffff89911764 size:67389854

ptr:0xffff89927d5f size:67298211

ptr:0xffff89942143 size:67190719

ptr:0xffff89958135 size:67100621

Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'temp.mp4':

Metadata:

major\_brand : mp42

minor\_version : 0

compatible\_brands: mp42mp41isomavc1

creation\_time : 2018-06-22T10:58:33.000000Z

Duration: 00:00:24.90, start: 0.000000, bitrate: N/A

Stream #0:0[0x1](und): Video: h264 (High) (avc1 / 0x31637661), yuv420p(tv, bt709, progressive), 4096x2160, 21892 kb/s, 24 fps, 24 tbr, 24 tbn (default)

Metadata:

creation\_time : 2018-06-22T10:58:33.000000Z

handler\_name : L-SMASH Video Handler

vendor\_id : [0][0][0][0]

encoder : AVC Coding

Stream #0:1[0x2](und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 137 kb/s (default)

Metadata:

creation\_time : 2018-06-22T10:58:33.000000Z

handler\_name : L-SMASH Audio Handler

vendor\_id : [0][0][0][0]

```

The ARM platform is the same as the RISCV platform.

### Test case 4: decode\_audio

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./decode\_audio temp.mp4 4.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

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

```

The ARM platform is the same as the RISCV platform.

### Test case 5: decode\_filter\_audio

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./decode\_filter\_audio temp.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

Output: srate:8000Hz fmt:s16 chlayout:mono

```

The ARM platform is the same as the RISCV platform.

### Test case 6: decode\_filter\_video

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./decode\_filter\_video temp.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

[in @ 0x21a1620] Changing video frame properties on the fly is not supported by all filters.

[in @ 0x21a1620] filter context - w: 4096 h: 2160 fmt: 0 csp: unknown range: unknown, incoming frame - w: 4096 h: 2160 fmt: 0 csp: bt709 range: tv pts\_time: 0

-------.--.-----...-----

-------..-------...-----

-----.-..-------...-----

-----....-------...-----

-----....------. ..-----

----....-------. ..-----

----....-------. ..----.

----....----+--. ..----.

----.-..----+-.. ..----.

----...-----+-...-.----.

.---...-----+-...-.----.

Then there are many symbols like ----, which are omitted here.

```

The ARM platform is the same as the RISCV platform.

### Test case 7: decode\_video

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./decode\_video temp.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

No output, no error means successful execution

```

The ARM platform is the same as the RISCV platform.

### Test case 8: demux\_decode

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./demux\_decode temp.mp4 2.mp4 3.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'temp.mp4':

Metadata:

major\_brand : mp42

minor\_version : 0

compatible\_brands: mp42mp41isomavc1

creation\_time : 2018-06-22T10:58:33.000000Z

Duration: 00:00:24.90, start: 0.000000, bitrate: 22015 kb/s

Stream #0:0[0x1](und): Video: h264 (High) (avc1 / 0x31637661), yuv420p(tv, bt709, progressive), 4096x2160, 21892 kb/s, 24 fps, 24 tbr, 24 tbn (default)

Metadata:

creation\_time : 2018-06-22T10:58:33.000000Z

handler\_name : L-SMASH Video Handler

vendor\_id : [0][0][0][0]

encoder : AVC Coding

Stream #0:1[0x2](und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 137 kb/s (default)

Metadata:

creation\_time : 2018-06-22T10:58:33.000000Z

handler\_name : L-SMASH Audio Handler

vendor\_id : [0][0][0][0]

Demuxing video from file 'temp.mp4' into '2.mp4'

Demuxing audio from file 'temp.mp4' into '3.mp4'

video\_frame n:0

video\_frame n:1

video\_frame n:2

video\_frame n:3

video\_frame n:4

video\_frame n:5

video\_frame n:6

video\_frame n:7

video\_frame n:8

video\_frame n:9

video\_frame n:10

audio\_frame n:0 nb\_samples:1024 pts:0

audio\_frame n:1 nb\_samples:1024 pts:0.0213333

audio\_frame n:2 nb\_samples:1024 pts:0.0426667

audio\_frame n:3 nb\_samples:1024 pts:0.064

audio\_frame n:4 nb\_samples:1024 pts:0.0853333

audio\_frame n:5 nb\_samples:1024 pts:0.106667

.................

audio\_frame n:1159 nb\_samples:1024 pts:24.725333

audio\_frame n:1160 nb\_samples:1024 pts:24.746667

audio\_frame n:1161 nb\_samples:1024 pts:24.768

audio\_frame n:1162 nb\_samples:1024 pts:24.789333

audio\_frame n:1163 nb\_samples:1024 pts:24.810667

audio\_frame n:1164 nb\_samples:1024 pts:24.832

audio\_frame n:1165 nb\_samples:1024 pts:24.853333

audio\_frame n:1166 nb\_samples:1024 pts:24.874667

video\_frame n:595

video\_frame n:596

Demuxing succeeded.

Play the output video file with the command:

ffplay -f rawvideo -pix\_fmt yuv420p -video\_size 4096x2160 2.mp4

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

Play the output audio file with the command:

ffplay -f f32le -ac 1 -ar 48000 3.mp4

```

The ARM platform is the same as the RISCV platform.

### Test case 9: encode\_audio

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./encode\_audio temp.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

No output, no error means successful execution

```

The ARM platform is the same as the RISCV platform.

### Test case 10: encode\_video

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./encode\_video temp.mp4 libx264

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

Send frame 0

Send frame 1

Send frame 2

Send frame 3

Send frame 4

Send frame 5

Send frame 6

Send frame 7

Send frame 8

Send frame 9

Send frame 10

Send frame 11

Send frame 12

Send frame 13

Send frame 14

Send frame 15

Send frame 16

Send frame 17

Send frame 18

Send frame 19

Send frame 20

Send frame 21

Write packet 0 (size= 2068)

Send frame 22

Write packet 2 (size= 674)

Send frame 23

Write packet 1 (size= 133)

Send frame 24

Write packet 4 (size= 767)

Write packet 3 (size= 193)

Write packet 6 (size= 682)

Write packet 5 (size= 521)

Write packet 8 (size= 737)

Write packet 7 (size= 437)

Write packet 9 (size= 506)

Write packet 10 (size= 2238)

Write packet 12 (size= 1028)

Write packet 11 (size= 696)

Write packet 14 (size= 903)

Write packet 13 (size= 482)

Write packet 16 (size= 916)

Write packet 15 (size= 678)

Write packet 18 (size= 1243)

Write packet 17 (size= 573)

Write packet 19 (size= 622)

Write packet 20 (size= 2361)

Write packet 22 (size= 1119)

Write packet 21 (size= 758)

Write packet 24 (size= 708)

Write packet 23 (size= 691)

[libx264 @ 0x20fa6080] frame I:3 Avg QP:25.33 size: 2222

[libx264 @ 0x20fa6080] frame P:12 Avg QP:24.22 size: 825

[libx264 @ 0x20fa6080] frame B:10 Avg QP:28.28 size: 516

[libx264 @ 0x20fa6080] consecutive B-frames: 20.0% 80.0%

[libx264 @ 0x20fa6080] mb I I16..4: 78.3% 11.6% 10.1%

[libx264 @ 0x20fa6080] mb P I16..4: 76.8% 0.9% 0.1% P16..4: 20.9% 0.6% 0.4% 0.0% 0.0% skip: 0.3%

[libx264 @ 0x20fa6080] mb B I16..4: 0.0% 0.0% 0.0% B16..8: 11.6% 0.3% 0.0% direct:11.2% skip:77.0% L0:24.6% L1:38.6% BI:36.9%

[libx264 @ 0x20fa6080] final ratefactor: 15.78

[libx264 @ 0x20fa6080] 8x8 transform intra:3.7% inter:16.7%

[libx264 @ 0x20fa6080] direct mvs spatial:0.0% temporal:100.0%

[libx264 @ 0x20fa6080] coded y,uvDC,uvAC intra: 4.8% 32.0% 4.2% inter: 1.3% 35.6% 8.8%

[libx264 @ 0x20fa6080] i16 v,h,dc,p: 0% 0% 0% 100%

[libx264 @ 0x20fa6080] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 3% 32% 16% 48% 0% 0% 0% 0% 0%

[libx264 @ 0x20fa6080] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 10% 6% 15% 53% 1% 5% 2% 7% 1%

[libx264 @ 0x20fa6080] i8c dc,h,v,p: 1% 6% 5% 88%

[libx264 @ 0x20fa6080] Weighted P-Frames: Y:0.0% UV:0.0%

[libx264 @ 0x20fa6080] ref P L0: 97.3% 0.7% 1.5% 0.3% 0.1% 0.1%

[libx264 @ 0x20fa6080] ref B L0: 66.6% 30.3% 3.1%

[libx264 @ 0x20fa6080] kb/s:173.87

```

The ARM platform is the same as the RISCV platform.

### Test case 11: extract\_mvs

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./extract\_mvs temp.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'temp.mp4':

Metadata:

major\_brand : mp42

minor\_version : 0

compatible\_brands: mp42mp41isomavc1

creation\_time : 2018-06-22T10:58:33.000000Z

Duration: 00:00:24.90, start: 0.000000, bitrate: 22015 kb/s

Stream #0:0[0x1](und): Video: h264 (High) (avc1 / 0x31637661), yuv420p(tv, bt709, progressive), 4096x2160, 21892 kb/s, 24 fps, 24 tbr, 24 tbn (default)

Metadata:

creation\_time : 2018-06-22T10:58:33.000000Z

handler\_name : L-SMASH Video Handler

vendor\_id : [0][0][0][0]

encoder : AVC Coding

Stream #0:1[0x2](und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 137 kb/s (default)

Metadata:

creation\_time : 2018-06-22T10:58:33.000000Z

handler\_name : L-SMASH Audio Handler

vendor\_id : [0][0][0][0]

framenum,source,blockw,blockh,srcx,srcy,dstx,dsty,flags,motion\_x,motion\_y,motion\_scale

2,-1,16,16, 15, 8, 8, 8,0x0, 30, 2, 4

2, 1,16,16, 11, 8, 40, 8,0x0,-117, 0, 4

2,-1,16,16, 63, 2, 56, 8,0x0, 30, -24, 4

2,-1, 8,16, 78, -2, 68, 8,0x0, 40, -42, 4

2,-1, 8,16, 86, -2, 76, 8,0x0, 40, -42, 4

2, 1,16,16, 64, 8, 88, 8,0x0, -96, 0, 4

2, 1,16,16, 76, 8, 104, 8,0x0,-112, 0, 4

2, 1,16,16, 92, 8, 120, 8,0x0,-112, 0, 4

2,-1,16,16, 224, 25, 136, 8,0x0, 353, 71, 4

2,-1,16,16, 240, 23, 152, 8,0x0, 353, 60, 4

...............

597,-1,16,16,2788,2165,2808,2152,0x0, -83, 52, 4

597,-1,16,16,2812,2163,2840,2152,0x0,-113, 45, 4

597,-1,16,16,2896,2156,2904,2152,0x0, -34, 19, 4

597,-1,16,16,2904,2155,2920,2152,0x0, -64, 15, 4

597,-1,16,16,2902,2147,2984,2152,0x0,-331, -21, 4

597,-1,16,16,2969,2151,3000,2152,0x0,-126, -6, 4

597,-1,16,16,3498,2152,3464,2152,0x0, 136, 0, 4

597,-1,16,16,3514,2155,3480,2152,0x0, 136, 15, 4

597,-1,16,16,3704,2152,3704,2152,0x0, 0, 0, 4

597,-1,16,16,3720,2153,3720,2152,0x0, 0, 6, 4

597,-1, 8,16,3747,2151,3732,2152,0x0, 63, -6, 4

597,-1, 8,16,3749,2152,3740,2152,0x0, 37, -2, 4

597,-1,16,16,3933,2144,3944,2152,0x0, -44, -33, 4

597,-1,16,16,4075,2151,4040,2152,0x0, 140, -4, 4

597,-1,16,16,4091,2151,4056,2152,0x0, 140, -4, 4

597,-1,16,16,4107,2151,4072,2152,0x0, 140, -4, 4

597,-1,16,16,4082,2153,4088,2152,0x0, -25, 6, 4

```

The ARM platform is the same as the RISCV platform.

### Test case 12: filter\_audio

You need to rebuild ffmpeg, which requires support from additional dependent libraries, and the default compilation options will not compile this use case.

The ARM platform is the same as the RISCV platform.

### Test case 13: hw\_decode

This is a hardware accelerated decoding example, which requires a device like CUDA, but it is not found on the server.

The ARM platform is the same as the RISCV platform.

### Test case 14: mux

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./mux temp.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

[libx264 @ 0x2de81a00] using cpu capabilities: ARMv8 NEON

[libx264 @ 0x2de81a00] profile High, level 1.3, 4:2:0, 8-bit

[libx264 @ 0x2de81a00] 264 - core 164 r3191 4613ac3 - H.264/MPEG-4 AVC codec - Copyleft 2003-2024 - http://www.videolan.org/x264.html - options: cabac=1 ref=3 deblock=1:0:0 analyze=0x3:0x113 me=hex subme=7 psy=1 psy\_rd=1.00:0.00 mixed\_ref=1 me\_range=16 chroma\_me=1 trellis=1 8x8dct=1 cqm=0 deadzone=21,11 fast\_pskip=1 chroma\_qp\_offset=-2 threads=9 lookahead\_threads=1 sliced\_threads=0 nr=0 decimate=1 interlaced=0 bluray\_compat=0 constrained\_intra=0 bframes=3 b\_pyramid=2 b\_adapt=1 b\_bias=0 direct=1 weightb=1 open\_gop=0 weightp=2 keyint=12 keyint\_min=1 scenecut=40 intra\_refresh=0 rc\_lookahead=12 rc=abr mbtree=1 bitrate=400 ratetol=1.0 qcomp=0.60 qpmin=0 qpmax=69 qpstep=4 ip\_ratio=1.40 aq=1:1.00

Output #0, mp4, to 'temp.mp4':

Stream #0:0: Video: h264, yuv420p, 352x288, q=2-31, 400 kb/s, 25 tbn

Side data:

cpb: bitrate max/min/avg: 0/0/400000 buffer size: 0 vbv\_delay: N/A

Stream #0:1: Audio: aac (LC), 44100 Hz, stereo, fltp, 64 kb/s

pts:-1024 pts\_time:-0.02322 dts:-1024 dts\_time:-0.02322 duration:1024 duration\_time:0.02322 stream\_index:1

pts:0 pts\_time:0 dts:0 dts\_time:0 duration:1024 duration\_time:0.02322 stream\_index:1

pts:1024 pts\_time:0.02322 dts:1024 dts\_time:0.02322 duration:1024 duration\_time:0.02322 stream\_index:1

pts:2048 pts\_time:0.0464399 dts:2048 dts\_time:0.0464399 duration:1024 duration\_time:0.02322 stream\_index:1

pts:3072 pts\_time:0.0696599 dts:3072 dts\_time:0.0696599 duration:1024 duration\_time:0.02322 stream\_index:1

pts:4096 pts\_time:0.0928798 dts:4096 dts\_time:0.0928798 duration:1024 duration\_time:0.02322 stream\_index:1

pts:5120 pts\_time:0.1161 dts:5120 dts\_time:0.1161 duration:1024 duration\_time:0.02322 stream\_index:1

pts:6144 pts\_time:0.13932 dts:6144 dts\_time:0.13932 duration:1024 duration\_time:0.02322 stream\_index:1

pts:7168 pts\_time:0.16254 dts:7168 dts\_time:0.16254 duration:1024 duration\_time:0.02322 stream\_index:1

pts:8192 pts\_time:0.18576 dts:8192 dts\_time:0.18576 duration:1024 duration\_time:0.02322 stream\_index:1

pts:9216 pts\_time:0.20898 dts:9216 dts\_time:0.20898 duration:1024 duration\_time:0.02322 stream\_index:1

pts:10240 pts\_time:0.2322 dts:10240 dts\_time:0.2322 duration:1024 duration\_time:0.02322 stream\_index:1

pts:11264 pts\_time:0.25542 dts:11264 dts\_time:0.25542 duration:1024 duration\_time:0.02322 stream\_index:1

pts:12288 pts\_time:0.278639 dts:12288 dts\_time:0.278639 duration:1024 duration\_time:0.02322 stream\_index:1

pts:13312 pts\_time:0.301859 dts:13312 dts\_time:0.301859 duration:1024 duration\_time:0.02322 stream\_index:1

pts:14336 pts\_time:0.325079 dts:14336 dts\_time:0.325079 duration:1024 duration\_time:0.02322 stream\_index:1

pts:15360 pts\_time:0.348299 dts:15360 dts\_time:0.348299 duration:1024 duration\_time:0.02322 stream\_index:1

.................

pts:123904 pts\_time:9.68 dts:122880 dts\_time:9.6 duration:0 duration\_time:0 stream\_index:0

pts:123392 pts\_time:9.64 dts:123392 dts\_time:9.64 duration:0 duration\_time:0 stream\_index:0

pts:124416 pts\_time:9.72 dts:123904 dts\_time:9.68 duration:0 duration\_time:0 stream\_index:0

pts:126976 pts\_time:9.92 dts:124416 dts\_time:9.72 duration:0 duration\_time:0 stream\_index:0

pts:125952 pts\_time:9.84 dts:124928 dts\_time:9.76 duration:0 duration\_time:0 stream\_index:0

pts:125440 pts\_time:9.8 dts:125440 dts\_time:9.8 duration:0 duration\_time:0 stream\_index:0

pts:126464 pts\_time:9.88 dts:125952 dts\_time:9.84 duration:0 duration\_time:0 stream\_index:0

pts:128000 pts\_time:10 dts:126464 dts\_time:9.88 duration:0 duration\_time:0 stream\_index:0

pts:127488 pts\_time:9.96 dts:126976 dts\_time:9.92 duration:0 duration\_time:0 stream\_index:0

[libx264 @ 0x2de81a00] frame I:21 Avg QP: 7.20 size: 4064

[libx264 @ 0x2de81a00] frame P:63 Avg QP: 8.31 size: 963

[libx264 @ 0x2de81a00] frame B:167 Avg QP:11.59 size: 1006

[libx264 @ 0x2de81a00] consecutive B-frames: 8.4% 0.8% 23.9% 66.9%

[libx264 @ 0x2de81a00] mb I I16..4: 78.0% 12.5% 9.5%

[libx264 @ 0x2de81a00] mb P I16..4: 76.4% 4.5% 0.2% P16..4: 18.4% 0.3% 0.2% 0.0% 0.0% skip: 0.0%

[libx264 @ 0x2de81a00] mb B I16..4: 0.0% 0.0% 0.0% B16..8: 10.5% 0.3% 0.0% direct:64.0% skip:25.1% L0:36.9% L1:60.6% BI: 2.5%

[libx264 @ 0x2de81a00] final ratefactor: 4.69

[libx264 @ 0x2de81a00] 8x8 transform intra:7.6% inter:40.9%

[libx264 @ 0x2de81a00] coded y,uvDC,uvAC intra: 6.7% 14.8% 9.8% inter: 0.2% 76.4% 7.6%

[libx264 @ 0x2de81a00] i16 v,h,dc,p: 0% 0% 0% 100%

[libx264 @ 0x2de81a00] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 0% 35% 5% 59% 0% 0% 0% 0% 0%

[libx264 @ 0x2de81a00] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 1% 18% 6% 69% 3% 0% 0% 2% 1%

[libx264 @ 0x2de81a00] i8c dc,h,v,p: 0% 4% 7% 89%

[libx264 @ 0x2de81a00] Weighted P-Frames: Y:0.0% UV:0.0%

[libx264 @ 0x2de81a00] ref P L0: 98.0% 0.5% 0.8% 0.7%

[libx264 @ 0x2de81a00] ref B L0: 88.3% 8.9% 2.8%

[libx264 @ 0x2de81a00] ref B L1: 96.1% 3.9%

[libx264 @ 0x2de81a00] kb/s:250.14

[aac @ 0x2de82a50] Qavg: 360.258

```

The ARM platform is the same as the RISCV platform.

### Test case 15: qsv\_decode

You need to rebuild and add ffmpeg, which requires support from additional dependent libraries. The default compilation options will not compile this use case.

The ARM platform is the same as the RISCV platform.

### Test case 16: qsv\_transcode

You need to rebuild and add ffmpeg, which requires support from additional dependent libraries. The default compilation options will not compile this use case.

The ARM platform is the same as the RISCV platform.

### Test case 17: remux

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./remux temp.mp4 1.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'temp.mp4':

Metadata:

major\_brand : mp42

minor\_version : 0

compatible\_brands: mp42mp41isomavc1

creation\_time : 2018-06-22T10:58:33.000000Z

Duration: 00:00:24.90, start: 0.000000, bitrate: 22015 kb/s

Stream #0:0[0x1](und): Video: h264 (High) (avc1 / 0x31637661), yuv420p(tv, bt709, progressive), 4096x2160, 21892 kb/s, 24 fps, 24 tbr, 24 tbn (default)

Metadata:

creation\_time : 2018-06-22T10:58:33.000000Z

handler\_name : L-SMASH Video Handler

vendor\_id : [0][0][0][0]

encoder : AVC Coding

Stream #0:1[0x2](und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 137 kb/s (default)

Metadata:

creation\_time : 2018-06-22T10:58:33.000000Z

handler\_name : L-SMASH Audio Handler

vendor\_id : [0][0][0][0]

Output #0, mp4, to '1.mp4':

Stream #0:0: Video: h264 (High), yuv420p(tv, bt709, progressive), 4096x2160, q=2-31, 21892 kb/s

Stream #0:1: Audio: aac (LC), 48000 Hz, stereo, fltp, 137 kb/s

in: pts:0 pts\_time:0 dts:-1 dts\_time:-0.0416667 duration:1 duration\_time:0.0416667 stream\_index:0

out: pts:0 pts\_time:0 dts:-3750 dts\_time:-0.0416667 duration:3750 duration\_time:0.0416667 stream\_index:0

in: pts:2 pts\_time:0.0833333 dts:0 dts\_time:0 duration:1 duration\_time:0.0416667 stream\_index:0

out: pts:7500 pts\_time:0.0833333 dts:0 dts\_time:0 duration:3750 duration\_time:0.0416667 stream\_index:0

in: pts:1 pts\_time:0.0416667 dts:1 dts\_time:0.0416667 duration:1 duration\_time:0.0416667 stream\_index:0

out: pts:3750 pts\_time:0.0416667 dts:3750 dts\_time:0.0416667 duration:3750 duration\_time:0.0416667 stream\_index:0

in: pts:4 pts\_time:0.166667 dts:2 dts\_time:0.0833333 duration:1 duration\_time:0.0416667 stream\_index:0

out: pts:15000 pts\_time:0.166667 dts:7500 dts\_time:0.0833333 duration:3750 duration\_time:0.0416667 stream\_index:0

in: pts:3 pts\_time:0.125 dts:3 dts\_time:0.125 duration:1 duration\_time:0.0416667 stream\_index:0

out: pts:11250 pts\_time:0.125 dts:11250 dts\_time:0.125 duration:3750 duration\_time:0.0416667 stream\_index:0

in: pts:5 pts\_time:0.208333 dts:4 dts\_time:0.166667 duration:1 duration\_time:0.0416667 stream\_index:0

out: pts:18750 pts\_time:0.208333 dts:15000 dts\_time:0.166667 duration:3750 duration\_time:0.0416667 stream\_index:0

in: pts:7 pts\_time:0.291667 dts:5 dts\_time:0.208333 duration:1 duration\_time:0.0416667 stream\_index:0

out: pts:26250 pts\_time:0.291667 dts:18750 dts\_time:0.208333 duration:3750 duration\_time:0.0416667 stream\_index:0

in: pts:6 pts\_time:0.25 dts:6 dts\_time:0.25 duration:1 duration\_time:0.0416667 stream\_index:0

......................................

out: pts:1190912 pts\_time:24.810667 dts:1190912 dts\_time:24.810667 duration:1024 duration\_time:0.0213333 stream\_index:1

in: pts:1191936 pts\_time:24.832 dts:1191936 dts\_time:24.832 duration:1024 duration\_time:0.0213333 stream\_index:1

out: pts:1191936 pts\_time:24.832 dts:1191936 dts\_time:24.832 duration:1024 duration\_time:0.0213333 stream\_index:1

in: pts:1192960 pts\_time:24.853333 dts:1192960 dts\_time:24.853333 duration:1024 duration\_time:0.0213333 stream\_index:1

out: pts:1192960 pts\_time:24.853333 dts:1192960 dts\_time:24.853333 duration:1024 duration\_time:0.0213333 stream\_index:1

in: pts:1193984 pts\_time:24.874667 dts:1193984 dts\_time:24.874667 duration:976 duration\_time:0.0203333 stream\_index:1

out: pts:1193984 pts\_time:24.874667 dts:1193984 dts\_time:24.874667 duration:976 duration\_time:0.0203333 stream\_index:1

```

The ARM platform is the same as the RISCV platform.

### Test case 18: resample\_audio

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./resample\_audio temp.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

t:0.021333 in:1024 out:925

t:0.042667 in:1024 out:941

t:0.064000 in:1024 out:940

t:0.085333 in:1024 out:941

t:0.106667 in:1024 out:941

t:0.128000 in:1024 out:941

t:0.149333 in:1024 out:941

........omitted........

t:9.877333 in:1024 out:940

t:9.898667 in:1024 out:941

t:9.920000 in:1024 out:941

t:9.941333 in:1024 out:941

t:9.962667 in:1024 out:941

t:9.984000 in:1024 out:940

t:10.005333 in:1024 out:941

Resampling succeeded. Play the output file with the command:

ffplay -f s16le -channel\_layout 3.0 -channels 3 -ar 44100 temp.mp4

```

The ARM platform is the same as the RISCV platform.

### Test case 19: scale\_video

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./scale\_video temp.mp4 1080\*720

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

Scaling succeeded. Play the output file with the command:

ffplay -f rawvideo -pix\_fmt rgb24 -video\_size 1080x720 temp.mp4

```

The ARM platform is the same as the RISCV platform.

### Test case 20: show\_metadata

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./show\_metadata temp.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

major\_brand=mp42

minor\_version=0

compatible\_brands=mp42mp41isomavc1

creation\_time=2018-06-22T10:58:33.000000Z

```

The ARM platform is the same as the RISCV platform.

### Test case 21: transcode

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./transcode temp.mp4 5.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'temp.mp4':

Metadata:

major\_brand : mp42

minor\_version : 0

compatible\_brands: mp42mp41isomavc1

creation\_time : 2018-06-22T10:58:33.000000Z

Duration: 00:00:24.90, start: 0.000000, bitrate: 22015 kb/s

Stream #0:0[0x1](und): Video: h264 (High) (avc1 / 0x31637661), yuv420p(tv, bt709, progressive), 4096x2160, 21892 kb/s, 24 fps, 24 tbr, 24 tbn (default)

Metadata:

creation\_time : 2018-06-22T10:58:33.000000Z

handler\_name : L-SMASH Video Handler

vendor\_id : [0][0][0][0]

encoder : AVC Coding

Stream #0:1[0x2](und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 137 kb/s (default)

Metadata:

creation\_time : 2018-06-22T10:58:33.000000Z

handler\_name : L-SMASH Audio Handler

vendor\_id : [0][0][0][0]

[libx264 @ 0x39ea1d10] using cpu capabilities: ARMv8 NEON

[libx264 @ 0x39ea1d10] profile High, level 5.1, 4:2:0, 8-bit

[libx264 @ 0x39ea1d10] 264 - core 164 r3191 4613ac3 - H.264/MPEG-4 AVC codec - Copyleft 2003-2024 - http://www.videolan.org/x264.html - options: cabac=1 ref=3 deblock=1:0:0 analyze=0x3:0x113 me=hex subme=7 psy=1 psy\_rd=1.00:0.00 mixed\_ref=1 me\_range=16 chroma\_me=1 trellis=1 8x8dct=1 cqm=0 deadzone=21,11 fast\_pskip=1 chroma\_qp\_offset=-2 threads=67 lookahead\_threads=11 sliced\_threads=0 nr=0 decimate=1 interlaced=0 bluray\_compat=0 constrained\_intra=0 bframes=3 b\_pyramid=2 b\_adapt=1 b\_bias=0 direct=1 weightb=1 open\_gop=0 weightp=2 keyint=250 keyint\_min=24 scenecut=40 intra\_refresh=0 rc\_lookahead=40 rc=crf mbtree=1 crf=23.0 qcomp=0.60 qpmin=0 qpmax=69 qpstep=4 ip\_ratio=1.40 aq=1:1.00

Output #0, mp4, to '5.mp4':

Stream #0:0: Video: h264, yuv420p, 4096x2160, q=2-31, 24 tbn

Side data:

cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv\_delay: N/A

Stream #0:1: Audio: aac (LC), 48000 Hz, stereo, fltp, 128 kb/s

Pushing decoded frame to filters

[in @ 0x39fc2230] Changing video frame properties on the fly is not supported by all filters.

[in @ 0x39fc2230] filter context - w: 4096 h: 2160 fmt: 0 csp: unknown range: unknown, incoming frame - w: 4096 h: 2160 fmt: 0 csp: bt709 range: tv pts\_time: 0

Pulling filtered frame from filters

Encoding frame

Pulling filtered frame from filters

Pushing decoded frame to filters

Pulling filtered frame from filters

Encoding frame

Pulling filtered frame from filters

Pushing decoded frame to filters

Pulling filtered frame from filters

Encoding frame

Pulling filtered frame from filters

Pushing decoded frame to filters

Pulling filtered frame from filters

Encoding frame

Pulling filtered frame from filters

Pushing decoded frame to filters

................................

Encoding frame

Pulling filtered frame from filters

Pushing decoded frame to filters

Pulling filtered frame from filters

Flushing stream #0 encoder

Encoding frame

Flushing stream 1 decoder

Pushing decoded frame to filters

Pulling filtered frame from filters

Flushing stream #1 encoder

Encoding frame

[libx264 @ 0x39ea1d10] frame I:26 Avg QP:23.13 size:178220

[libx264 @ 0x39ea1d10] frame P:416 Avg QP:25.82 size:135277

[libx264 @ 0x39ea1d10] frame B:155 Avg QP:26.22 size:120419

[libx264 @ 0x39ea1d10] consecutive B-frames: 48.6% 49.9% 1.5% 0.0%

[libx264 @ 0x39ea1d10] mb I I16..4: 31.1% 62.9% 6.0%

[libx264 @ 0x39ea1d10] mb P I16..4: 31.9% 48.4% 2.2% P16..4: 12.4% 2.6% 0.2% 0.0% 0.0% skip: 2.3%

[libx264 @ 0x39ea1d10] mb B I16..4: 14.0% 19.2% 0.9% B16..8: 33.1% 9.8% 0.6% direct: 3.0% skip:19.5% L0:45.4% L1:49.0% BI: 5.6%

[libx264 @ 0x39ea1d10] 8x8 transform intra:58.7% inter:91.2%

[libx264 @ 0x39ea1d10] coded y,uvDC,uvAC intra: 36.4% 31.0% 0.2% inter: 22.5% 26.8% 0.0%

[libx264 @ 0x39ea1d10] i16 v,h,dc,p: 2% 56% 2% 40%

[libx264 @ 0x39ea1d10] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 8% 62% 10% 2% 2% 1% 5% 1% 8%

[libx264 @ 0x39ea1d10] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 7% 66% 9% 1% 3% 1% 8% 1% 4%

[libx264 @ 0x39ea1d10] i8c dc,h,v,p: 44% 46% 7% 4%

[libx264 @ 0x39ea1d10] Weighted P-Frames: Y:1.2% UV:0.5%

[libx264 @ 0x39ea1d10] ref P L0: 68.5% 15.2% 9.7% 6.6% 0.1%

[libx264 @ 0x39ea1d10] ref B L0: 89.0% 10.8% 0.2%

[libx264 @ 0x39ea1d10] kb/s:25591.52

[aac @ 0x3aca8370] Qavg: 65536.000

```

The ARM platform is the same as the RISCV platform.

### Test case 22: transcode\_aac

After executing `build.sh`, go to the `exe` directory and run the test case:

```sh

cp test.mp4 temp.mp4

./encode\_audio temp.mp4

./transcode\_aac temp.mp4 6.mp4

rm temp.mp4 # This prevents this sample from affecting other samples

```

The running results are:

```sh

[mp3 @ 0x8e3ceb0] Estimating duration from bitrate, this may be inaccurate

[aac @ 0x8e41ef0] Qavg: 304.467

```

The ARM platform is the same as the RISCV platform.

### Test case 23: vaapi\_encode

You need to rebuild and add ffmpeg, which requires support from additional dependent libraries. The default compilation options will not compile this use case.

The ARM platform is the same as the RISCV platform.

### Test case 24: vaapi\_transcode

You need to rebuild and add ffmpeg, which requires support from additional dependent libraries. The default compilation options will not compile this use case.

The ARM platform is the same as the RISCV platform.

## Test item - Fate

### Test case 1: fate-ac3

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

```sh

make fate-ac3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST ac3-2.0

TEST ac3-4.0

TEST ac3-4.0-downmix-mono

TEST ac3-4.0-downmix-stereo

TEST ac3-5.1

TEST ac3-5.1-downmix-mono

TEST ac3-5.1-downmix-stereo

TEST ac3-fixed-2.0

TEST ac3-fixed-4.0-downmix-mono

TEST ac3-fixed-5.1-downmix-mono

TEST ac3-fixed-5.1-downmix-stereo

TEST ac3-encode

TEST ac3-fixed-encode

TEST eac3-1

TEST eac3-2

TEST eac3-3

TEST eac3-4

TEST eac3-5

TEST eac3-encode

TEST eac3-core-bsf

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 2: fate-acodec

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

```sh

make fate-acodec SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST acodec-pcm-alaw

TEST acodec-pcm-mulaw

TEST acodec-pcm-s16be

TEST acodec-pcm-s16le

TEST acodec-pcm-u16be

TEST acodec-pcm-u16le

TEST acodec-pcm-s8

TEST acodec-pcm-u8

TEST acodec-pcm-s24be

TEST acodec-pcm-s24le

TEST acodec-pcm-u24be

TEST acodec-pcm-u24le

TEST acodec-pcm-s32be

TEST acodec-pcm-s32le

TEST acodec-pcm-u32be

TEST acodec-pcm-u32le

TEST acodec-pcm-f32be

TEST acodec-pcm-f32le

TEST acodec-pcm-f64be

TEST acodec-pcm-f64le

TEST acodec-pcm-s8\_planar

TEST acodec-pcm-s16be\_planar

TEST acodec-pcm-s16le\_planar

TEST acodec-pcm-s24le\_planar

TEST acodec-pcm-s32le\_planar

TEST acodec-adpcm-ima\_apm

TEST acodec-adpcm-ima\_alp

TEST acodec-adpcm-ima\_ssi

TEST acodec-adpcm-ima\_ws

TEST acodec-adpcm-ms

TEST acodec-adpcm-swf

TEST acodec-adpcm-swf-wav

TEST acodec-adpcm-yamaha

TEST acodec-adpcm-adx

TEST acodec-adpcm-argo

TEST acodec-adpcm-ima\_qt

TEST acodec-adpcm-ima\_wav

TEST acodec-adpcm-ima\_qt-trellis

TEST acodec-adpcm-ima\_wav-trellis

TEST acodec-adpcm-ms-trellis

TEST acodec-adpcm-swf-trellis

TEST acodec-adpcm-yamaha-trellis

TEST acodec-mp2

TEST acodec-mp2fixed

TEST acodec-alac

TEST acodec-dca

TEST acodec-dca2

TEST acodec-flac

TEST acodec-flac-exact-rice

TEST acodec-g723\_1

TEST acodec-ra144

TEST acodec-roqaudio

TEST acodec-s302m

TEST acodec-wavpack

TEST acodec-tta

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 3: fate-adpcm

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

```sh

make fate-adpcm SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST adpcm-4xm

TEST adpcm-afc

TEST adpcm-creative

TEST adpcm-creative-8-2bit

TEST adpcm-creative-8-2.6bit

TEST adpcm-creative-8-4bit

TEST adpcm-dtk

TEST adpcm-ea-1

TEST adpcm-ea-2

TEST adpcm-ea-maxis-xa

TEST adpcm-ea-r1

TEST adpcm-ea-r2

TEST adpcm-ea-r3

TEST adpcm-ima-amv

TEST adpcm-ima-apc

TEST adpcm-ima-dk3

TEST adpcm-ima-dk4

TEST adpcm-ima-ea-eacs

TEST adpcm-ima-ea-sead

TEST adpcm-ima-iss

TEST adpcm-ima-oki

TEST adpcm-ima-rad

TEST adpcm-ima-smjpeg

TEST adpcm-ima\_wav-stereo

TEST adpcm-ima-ws

TEST adpcm-ima-ws-vqa3

TEST adpcm-ms-mono

TEST adpcm\_ms-stereo

TEST adpcm-thp

TEST adpcm-vima

TEST adpcm-xa

TEST adpcm-argo-mono

TEST adpcm-argo-stereo

TEST adpcm-ima-ssi-mono

TEST adpcm-ima-ssi-stereo

TEST adpcm-ima-apm-mono

TEST adpcm-ima-apm-stereo

TEST adpcm-ima-alp-mono

TEST adpcm-ima-alp-stereo

TEST adpcm-ima-cunning-single

TEST adpcm-ima-cunning-track0

TEST adpcm-ima-cunning-track1

TEST adpcm-ima-cunning-trunc-t1

TEST adpcm-ima-cunning-trunc-t2-track0

TEST adpcm-ima-cunning-trunc-t2-track1

TEST adpcm-ima-cunning-trunc-t2a-track0

TEST adpcm-ima-cunning-trunc-t2a-track1

TEST adpcm-ima-cunning-trunc-h2

TEST adpcm-ima-cunning-stereo

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 4: fate-afilter

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

```sh

make fate-afilter SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST filter-adelay

TEST filter-aecho

TEST filter-aemphasis-50fm

TEST filter-aemphasis-75kf

TEST filter-afade-qsin

TEST filter-afade-iqsin

TEST filter-afade-esin

TEST filter-afade-hsin

TEST filter-afade-exp

TEST filter-afade-log

TEST filter-agate

TEST filter-alimiter

TEST filter-amerge

TEST filter-apad

COPY tests/data/filtergraphs/anequalizer

TEST filter-anequalizer

TEST filter-asetnsamples-pad

TEST filter-asetnsamples-nopad

TEST filter-asetrate

GEN tests/data/asynth-22050-1.wav

TEST filter-chorus

TEST filter-crystalizer

TEST filter-dcshift

TEST filter-earwax

TEST filter-extrastereo

COPY tests/data/filtergraphs/firequalizer

TEST filter-firequalizer

TEST filter-pan-mono1

TEST filter-pan-mono2

GEN tests/data/asynth-44100-3.wav

TEST filter-pan-stereo1

TEST filter-pan-stereo2

TEST filter-pan-stereo3

TEST filter-pan-stereo4

TEST filter-pan-upmix1

TEST filter-pan-upmix2

TEST filter-pan-downmix1

GEN tests/data/asynth-44100-11.wav

TEST filter-pan-downmix2

TEST filter-silenceremove

COPY tests/data/filtergraphs/compand

TEST filter-compand

GEN tests/data/hls-list.m3u8

TEST filter-hls

GEN tests/data/hls-list-append.m3u8

TEST filter-hls-append

TEST filter-atrim-duration

TEST filter-atrim-mixed

TEST filter-atrim-samples

TEST filter-atrim-time

TEST filter-asegment-samples-absolute

TEST filter-asegment-samples-relative

TEST filter-asegment-timestamps-absolute

TEST filter-asegment-timestamps-relative

COPY tests/data/filtergraphs/channelmap\_one\_int

GEN tests/data/asynth-44100-6.wav

TEST filter-channelmap-one-int

COPY tests/data/filtergraphs/channelmap\_one\_str

TEST filter-channelmap-one-str

TEST filter-channelsplit

TEST filter-join

TEST filter-volume

TEST filter-formats

TEST filter-acrossfade

TEST filter-stereotools

TEST filter-tremolo

GEN tests/data/asynth-44100-2-2.wav

TEST filter-amix-simple

TEST filter-amix-first

GEN tests/data/asynth-44100-2-3.wav

TEST filter-amix-transition

TEST filter-aresample

TEST filter-hdcd-mix

TEST filter-hdcd-mix-psoff

TEST filter-hdcd-analyze-pe

TEST filter-hdcd-analyze-lle

TEST filter-hdcd-analyze-cdt

TEST filter-hdcd-analyze-tgm

TEST filter-hdcd-analyze-ltgm

TEST filter-hdcd-analyze-pel

TEST filter-hdcd-false-positive

TEST filter-hdcd-detect-errors

TEST filter-hdcd-20bit

TEST filter-hdcd-mono

TEST filter-hdcd-s16p

TEST filter-hdcd-s32p

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 5: fate-alac

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

```sh

make fate-alac SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST alac-16-level-0

TEST alac-16-level-1

TEST alac-16-level-2

TEST alac-16-lpc-orders

TEST alac-24-level-0

TEST alac-24-level-1

TEST alac-24-level-2

TEST alac-24-lpc-orders

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 6: fate-als

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

```sh

make fate-als SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST mpeg4-als-conformance-00

TEST mpeg4-als-conformance-01

TEST mpeg4-als-conformance-02

TEST mpeg4-als-conformance-03

TEST mpeg4-als-conformance-04

TEST mpeg4-als-conformance-05

TEST mpeg4-als-conformance-09

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 7: fate-amrnb

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

```sh

make fate-amrnb SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST amrnb-4k75

TEST amrnb-5k15

TEST amrnb-5k9

TEST amrnb-6k7

TEST amrnb-7k4

TEST amrnb-7k95

TEST amrnb-10k2

TEST amrnb-12k2

TEST amrnb-remux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 8: fate-amrwb

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

```sh

make fate-amrwb SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST amrwb-6k60

TEST amrwb-8k85

TEST amrwb-12k65

TEST amrwb-14k25

TEST amrwb-15k85

TEST amrwb-18k25

TEST amrwb-19k85

TEST amrwb-23k05

TEST amrwb-23k85

TEST amrwb-23k85-2

TEST amrwb-remux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 9: fate-api

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

```sh

make fate-api SAMPLES=fate\_samples/

```

The running results are:

```sh

CC tests/api/api-threadmessage-test.o

LD tests/api/api-threadmessage-test

TEST api-threadmessage

CC tests/api/api-flac-test.o

LD tests/api/api-flac-test

TEST api-flac

CC tests/api/api-seek-test.o

LD tests/api/api-seek-test

HOSTCC tests/audiogen.o

HOSTLD tests/audiogen

GEN tests/data/asynth1.sw

TEST lavf-flv

TEST api-seek

CC tests/api/api-band-test.o

LD tests/api/api-band-test

TEST api-band

CC tests/api/api-h264-test.o

LD tests/api/api-h264-test

TEST api-h264

CC tests/api/api-h264-slice-test.o

LD tests/api/api-h264-slice-test

TEST api-h264-slice

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 10: fate-apng

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

```sh

make fate-apng SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST apng-clock

TEST apng-osample

TEST apng-dispose-previous

TEST apng-dispose-background

TEST apng-dispose-background2

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 11: fate-atrac

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

```sh

make fate-atrac SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST atrac1-1

TEST atrac1-2

TEST atrac3-1

TEST atrac3-2

TEST atrac3-3

TEST atrac3p-1

TEST atrac3p-2

TEST atrac-aea-remux

TEST atrac-matroska-remux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 12: fate-audio

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

```sh

make fate-audio SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST binkaudio-dct

TEST binkaudio-rdft

TEST aptx

TEST aptx-hd

TEST bmv-audio

TEST delphine-cin-audio

TEST dolby-e

TEST dss-lp

TEST dss-sp

TEST dsf-dst

TEST imc

TEST msnsiren

TEST nellymoser

TEST nellymoser-aref-encode

TEST on2avc

TEST paf-audio

TEST sierra-vmd-audio

TEST smacker-audio

TEST ws\_snd

TEST flcl1905

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 13: fate-bmp

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

```sh

make fate-bmp SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST bmp-24bit

TEST bmp-1bit

TEST bmp-4bit

TEST bmp-4bit-os2

TEST bmp-8bit

TEST bmp-8bit-os2

TEST bmp-15bit

TEST bmp-15bit-mask

TEST bmp-16bit-mask

TEST bmp-32bit

TEST bmp-32bit-mask

TEST bmp-rle4

TEST bmp-rle8

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 14: fate-build

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

```sh

make fate-build SAMPLES=fate\_samples/

```

The running results are:

```sh

CC tools/enc\_recon\_frame\_test.o

LD tools/enc\_recon\_frame\_test

CC tools/enum\_options.o

LD tools/enum\_options

CC tools/qt-faststart.o

LD tools/qt-faststart

CC tools/trasher.o

LD tools/trasher

CC tools/uncoded\_frame.o

LD tools/uncoded\_frame

CC tools/cws2fws.o

LD tools/cws2fws

CC tools/graph2dot.o

tools/graph2dot.c: In function 'main':

tools/graph2dot.c:120:46: warning: comparison is always true due to limited range of data type [-Wtype-limits]

120 | while ((c = getopt(argc, argv, "hi:o:")) != -1) {

| ^~

LD tools/graph2dot

CC tools/aviocat.o

LD tools/aviocat

CC tools/ismindex.o

tools/ismindex.c: In function 'main':

tools/ismindex.c:184:49: warning: '.ismf' directive output may be truncated writing 5 bytes into a region of size between 1 and 2048 [-Wformat-truncation=]

184 | snprintf(idxname, sizeof(idxname), "%s%s.ismf", output\_prefix, basename);

| ^~~~~

In function 'write\_fragments',

inlined from 'read\_mfra' at tools/ismindex.c:442:15,

inlined from 'handle\_file' at tools/ismindex.c:608:11,

inlined from 'main' at tools/ismindex.c:821:17:

tools/ismindex.c:184:9: note: 'snprintf' output 6 or more bytes (assuming 2053) into a destination of size 2048

184 | snprintf(idxname, sizeof(idxname), "%s%s.ismf", output\_prefix, basename);

| ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

tools/ismindex.c: In function 'main':

tools/ismindex.c:195:47: warning: 'QualityLevels(' directive output may be truncated writing 14 bytes into a region of size between 1 and 2048 [-Wformat-truncation=]

195 | snprintf(dirname, sizeof(dirname), "%sQualityLevels(%d)", output\_prefix, track->bitrate);

| ^~~~~~~~~~~~~~

In function 'write\_fragments',

inlined from 'read\_mfra' at tools/ismindex.c:442:15,

inlined from 'handle\_file' at tools/ismindex.c:608:11,

inlined from 'main' at tools/ismindex.c:821:17:

tools/ismindex.c:195:9: note: 'snprintf' output between 17 and 2074 bytes into a destination of size 2048

195 | snprintf(dirname, sizeof(dirname), "%sQualityLevels(%d)", output\_prefix, track->bitrate);

| ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

tools/ismindex.c: In function 'main':

tools/ismindex.c:204:53: warning: '/Fragments(' directive output may be truncated writing 11 bytes into a region of size between 1 and 2048 [-Wformat-truncation=]

204 | snprintf(filename, sizeof(filename), "%s/Fragments(%s=%"PRId64")",

| ^~~~~~~~~~~

In function 'write\_fragments',

inlined from 'read\_mfra' at tools/ismindex.c:442:15,

inlined from 'handle\_file' at tools/ismindex.c:608:11,

inlined from 'main' at tools/ismindex.c:821:17:

tools/ismindex.c:204:13: note: 'snprintf' output between 20 and 2086 bytes into a destination of size 2048

204 | snprintf(filename, sizeof(filename), "%s/Fragments(%s=%"PRId64")",

| ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

205 | dirname, type, track->offsets[j].time);

| ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

tools/ismindex.c: In function 'main':

tools/ismindex.c:698:47: warning: '%s' directive output may be truncated writing up to 2047 bytes into a region of size 1000 [-Wformat-truncation=]

698 | snprintf(filename, sizeof(filename), "%sManifest", output\_prefix);

| ^~

In function 'output\_client\_manifest',

inlined from 'main' at tools/ismindex.c:832:5:

tools/ismindex.c:698:9: note: 'snprintf' output between 9 and 2056 bytes into a destination of size 1000

698 | snprintf(filename, sizeof(filename), "%sManifest", output\_prefix);

| ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

tools/ismindex.c: In function 'main':

tools/ismindex.c:626:43: warning: '%s' directive output may be truncated writing up to 2047 bytes into a region of size 1000 [-Wformat-truncation=]

626 | snprintf(filename, sizeof(filename), "%s%s.ism", output\_prefix, basename);

| ^~

In function 'output\_server\_manifest',

inlined from 'main' at tools/ismindex.c:830:9:

tools/ismindex.c:626:5: note: 'snprintf' output 5 or more bytes (assuming 2052) into a destination of size 1000

626 | snprintf(filename, sizeof(filename), "%s%s.ism", output\_prefix, basename);

| ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

tools/ismindex.c: In function 'main':

tools/ismindex.c:700:47: warning: '%s' directive output may be truncated writing up to 2047 bytes into a region of size 1000 [-Wformat-truncation=]

700 | snprintf(filename, sizeof(filename), "%s%s.ismc", output\_prefix, basename);

| ^~

In function 'output\_client\_manifest',

inlined from 'main' at tools/ismindex.c:832:5:

tools/ismindex.c:700:9: note: 'snprintf' output 6 or more bytes (assuming 2053) into a destination of size 1000

700 | snprintf(filename, sizeof(filename), "%s%s.ismc", output\_prefix, basename);

| ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

LD tools/ismindex

CC tools/pktdumper.o

LD tools/pktdumper

CC tools/probetest.o

LD tools/probetest

CC tools/seek\_print.o

LD tools/seek\_print

CC tools/sidxindex.o

LD tools/sidxindex

CC tools/venc\_data\_dump.o

LD tools/venc\_data\_dump

CC tools/fourcc2pixfmt.o

tools/fourcc2pixfmt.c: In function 'main':

tools/fourcc2pixfmt.c:71:46: warning: comparison is always true due to limited range of data type [-Wtype-limits]

71 | while ((c = getopt(argc, argv, "hp:lL")) != -1) {

| ^~

LD tools/fourcc2pixfmt

CC tools/crypto\_bench.o

LD tools/crypto\_bench

CC tools/ffhash.o

LD tools/ffhash

CC tools/ffeval.o

LD tools/ffeval

CC tools/ffescape.o

LD tools/ffescape

CC libavdevice/avdevice.ho

CC libavdevice/fbdev\_common.ho

CC libavdevice/internal.ho

CC libavdevice/opengl\_enc\_shaders.ho

CC libavdevice/oss.ho

CC libavdevice/timefilter.ho

CC libavdevice/v4l2-common.ho

CC libavdevice/version.ho

CC libavdevice/version\_major.ho

CC libavfilter/af\_afir.ho

CC libavfilter/af\_afirdsp.ho

CC libavfilter/af\_anlmdndsp.ho

CC libavfilter/af\_volume.ho

CC libavfilter/atadenoise.ho

CC libavfilter/audio.ho

CC libavfilter/avf\_showcqt.ho

CC libavfilter/avfilter.ho

CC libavfilter/avfilter\_internal.ho

CC libavfilter/bbox.ho

CC libavfilter/blend.ho

CC libavfilter/boxblur.ho

CC libavfilter/bufferqueue.ho

CC libavfilter/buffersink.ho

CC libavfilter/buffersrc.ho

CC libavfilter/bwdifdsp.ho

CC libavfilter/ccfifo.ho

CC libavfilter/colorspace.ho

CC libavfilter/colorspacedsp.ho

CC libavfilter/convolution.ho

CC libavfilter/dnn\_filter\_common.ho

CC libavfilter/dnn\_interface.ho

CC libavfilter/drawutils.ho

CC libavfilter/ebur128.ho

CC libavfilter/edge\_common.ho

CC libavfilter/filters.ho

CC libavfilter/formats.ho

CC libavfilter/framepool.ho

CC libavfilter/framequeue.ho

CC libavfilter/framerate.ho

CC libavfilter/framesync.ho

CC libavfilter/gblur.ho

CC libavfilter/generate\_wave\_table.ho

CC libavfilter/gradfun.ho

CC libavfilter/hermite.ho

CC libavfilter/hflip.ho

CC libavfilter/internal.ho

CC libavfilter/lavfutils.ho

CC libavfilter/limiter.ho

CC libavfilter/lswsutils.ho

CC libavfilter/lut3d.ho

CC libavfilter/maskedclamp.ho

CC libavfilter/maskedmerge.ho

CC libavfilter/median.ho

CC libavfilter/motion\_estimation.ho

CC libavfilter/opencl\_source.ho

CC libavfilter/palette.ho

CC libavfilter/preserve\_color.ho

CC libavfilter/psnr.ho

CC libavfilter/qp\_table.ho

CC libavfilter/removegrain.ho

CC libavfilter/scale\_eval.ho

CC libavfilter/scene\_sad.ho

CC libavfilter/signature.ho

CC libavfilter/ssim.ho

CC libavfilter/stereo3d.ho

CC libavfilter/textutils.ho

CC libavfilter/threshold.ho

CC libavfilter/tinterlace.ho

CC libavfilter/transform.ho

CC libavfilter/transpose.ho

CC libavfilter/unsharp.ho

CC libavfilter/v360.ho

CC libavfilter/version.ho

CC libavfilter/version\_major.ho

CC libavfilter/vf\_blend\_init.ho

CC libavfilter/vf\_eq.ho

CC libavfilter/vf\_fspp.ho

CC libavfilter/vf\_gblur\_init.ho

CC libavfilter/vf\_hflip\_init.ho

CC libavfilter/vf\_hqdn3d.ho

CC libavfilter/vf\_idet.ho

CC libavfilter/vf\_nlmeans.ho

CC libavfilter/vf\_nlmeans\_init.ho

CC libavfilter/vf\_noise.ho

CC libavfilter/vf\_overlay.ho

CC libavfilter/vf\_pp7.ho

CC libavfilter/vf\_pullup.ho

CC libavfilter/vf\_scale\_cuda.ho

CC libavfilter/vf\_spp.ho

CC libavfilter/vf\_threshold\_init.ho

CC libavfilter/video.ho

CC libavfilter/vmaf\_motion.ho

CC libavfilter/vsrc\_ddagrab\_shaders.ho

CC libavfilter/w3fdif.ho

CC libavfilter/window\_func.ho

CC libavfilter/yadif.ho

CC libavformat/aiff.ho

CC libavformat/apetag.ho

CC libavformat/argo\_asf.ho

CC libavformat/asf.ho

CC libavformat/asfcrypt.ho

CC libavformat/ast.ho

CC libavformat/av1.ho

CC libavformat/avc.ho

CC libavformat/avformat.ho

CC libavformat/avi.ho

CC libavformat/avio.ho

CC libavformat/avio\_internal.ho

CC libavformat/avlanguage.ho

CC libavformat/caf.ho

CC libavformat/dash.ho

CC libavformat/demux.ho

CC libavformat/dovi\_isom.ho

CC libavformat/dv.ho

CC libavformat/dvdclut.ho

CC libavformat/evc.ho

CC libavformat/ffmeta.ho

CC libavformat/flac\_picture.ho

CC libavformat/flacenc.ho

CC libavformat/flv.ho

CC libavformat/gxf.ho

CC libavformat/hevc.ho

CC libavformat/hls\_sample\_encryption.ho

CC libavformat/hlsplaylist.ho

CC libavformat/http.ho

CC libavformat/httpauth.ho

CC libavformat/iamf.ho

CC libavformat/iamf\_parse.ho

CC libavformat/iamf\_reader.ho

CC libavformat/iamf\_writer.ho

CC libavformat/id3v1.ho

CC libavformat/id3v2.ho

CC libavformat/img2.ho

CC libavformat/internal.ho

CC libavformat/ip.ho

CC libavformat/ircam.ho

CC libavformat/isom.ho

CC libavformat/lrc.ho

CC libavformat/matroska.ho

CC libavformat/metadata.ho

CC libavformat/mms.ho

CC libavformat/mov\_chan.ho

CC libavformat/movenc.ho

CC libavformat/movenc\_ttml.ho

CC libavformat/movenccenc.ho

CC libavformat/mpeg.ho

CC libavformat/mpegts.ho

CC libavformat/mux.ho

CC libavformat/mxf.ho

CC libavformat/network.ho

CC libavformat/nut.ho

CC libavformat/oggdec.ho

CC libavformat/oma.ho

CC libavformat/options\_table.ho

CC libavformat/os\_support.ho

CC libavformat/pcm.ho

CC libavformat/qtpalette.ho

CC libavformat/rawdec.ho

CC libavformat/rawenc.ho

CC libavformat/rawutils.ho

CC libavformat/rdt.ho

CC libavformat/replaygain.ho

CC libavformat/riff.ho

CC libavformat/rm.ho

CC libavformat/rmsipr.ho

CC libavformat/rso.ho

CC libavformat/rtmp.ho

CC libavformat/rtmpcrypt.ho

CC libavformat/rtmppkt.ho

CC libavformat/rtp.ho

CC libavformat/rtpdec.ho

CC libavformat/rtpdec\_formats.ho

CC libavformat/rtpenc.ho

CC libavformat/rtpenc\_chain.ho

CC libavformat/rtpproto.ho

CC libavformat/rtsp.ho

CC libavformat/rtspcodes.ho

CC libavformat/sauce.ho

CC libavformat/smjpeg.ho

CC libavformat/sox.ho

CC libavformat/spdif.ho

CC libavformat/srtp.ho

CC libavformat/subtitles.ho

CC libavformat/swf.ho

CC libavformat/tee\_common.ho

CC libavformat/tls.ho

CC libavformat/ttmlenc.ho

CC libavformat/url.ho

CC libavformat/urldecode.ho

CC libavformat/version.ho

CC libavformat/version\_major.ho

CC libavformat/voc.ho

CC libavformat/vorbiscomment.ho

CC libavformat/vpcc.ho

CC libavformat/vvc.ho

CC libavformat/w64.ho

CC libavformat/wtv.ho

CC libavformat/wv.ho

CC libavformat/yuv4mpeg.ho

CC libavcodec/a64colors.ho

CC libavcodec/a64tables.ho

CC libavcodec/aac.ho

CC libavcodec/aac\_ac3\_parser.ho

CC libavcodec/aac\_defines.ho

CC libavcodec/aacdec.ho

CC libavcodec/aacdectab.ho

CC libavcodec/aacenc.ho

CC libavcodec/aacenc\_is.ho

CC libavcodec/aacenc\_ltp.ho

CC libavcodec/aacenc\_pred.ho

CC libavcodec/aacenc\_tns.ho

CC libavcodec/aacenc\_utils.ho

CC libavcodec/aacencdsp.ho

CC libavcodec/aacenctab.ho

CC libavcodec/aacps.ho

CC libavcodec/aacpsdsp.ho

CC libavcodec/aacsbr.ho

CC libavcodec/aacsbrdata.ho

CC libavcodec/aactab.ho

CC libavcodec/aandcttab.ho

CC libavcodec/ac3.ho

CC libavcodec/ac3\_channel\_layout\_tab.ho

CC libavcodec/ac3\_parser.ho

CC libavcodec/ac3\_parser\_internal.ho

CC libavcodec/ac3dec.ho

In file included from libavcodec/ac3dec.hc:1:

./libavcodec/ac3dec.h:262:12: warning: 'ff\_eac3\_parse\_header' declared 'static' but never defined [-Wunused-function]

262 | static int ff\_eac3\_parse\_header(AC3DecodeContext \*s);

| ^~~~~~~~~~~~~~~~~~~~

./libavcodec/ac3dec.h:268:13: warning: 'ff\_eac3\_decode\_transform\_coeffs\_aht\_ch' declared 'static' but never defined [-Wunused-function]

268 | static void ff\_eac3\_decode\_transform\_coeffs\_aht\_ch(AC3DecodeContext \*s, int ch);

| ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

./libavcodec/ac3dec.h:275:13: warning: 'ff\_eac3\_apply\_spectral\_extension' declared 'static' but never defined [-Wunused-function]

275 | static void ff\_eac3\_apply\_spectral\_extension(AC3DecodeContext \*s);

| ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

CC libavcodec/ac3dec\_data.ho

CC libavcodec/ac3defs.ho

CC libavcodec/ac3dsp.ho

CC libavcodec/ac3enc.ho

CC libavcodec/ac3tab.ho

CC libavcodec/acelp\_filters.ho

CC libavcodec/acelp\_pitch\_delay.ho

CC libavcodec/acelp\_vectors.ho

CC libavcodec/adpcm.ho

CC libavcodec/adpcm\_data.ho

CC libavcodec/adts\_header.ho

CC libavcodec/adts\_parser.ho

CC libavcodec/adx.ho

CC libavcodec/alac\_data.ho

CC libavcodec/alacdsp.ho

CC libavcodec/amr.ho

CC libavcodec/amrnbdata.ho

CC libavcodec/amrwbdata.ho

CC libavcodec/aom\_film\_grain.ho

CC libavcodec/apng.ho

CC libavcodec/aptx.ho

CC libavcodec/ass.ho

CC libavcodec/ass\_split.ho

CC libavcodec/asv.ho

CC libavcodec/atrac.ho

CC libavcodec/atrac1data.ho

CC libavcodec/atrac3data.ho

CC libavcodec/atrac3plus.ho

CC libavcodec/atrac3plus\_data.ho

CC libavcodec/atrac9tab.ho

CC libavcodec/atsc\_a53.ho

CC libavcodec/audio\_frame\_queue.ho

CC libavcodec/audiodsp.ho

CC libavcodec/av1.ho

CC libavcodec/av1\_levels.ho

CC libavcodec/av1\_parse.ho

CC libavcodec/av1dec.ho

CC libavcodec/avcodec.ho

CC libavcodec/avcodec\_internal.ho

CC libavcodec/avdct.ho

CC libavcodec/avfft.ho

CC libavcodec/avs2.ho

CC libavcodec/avs3.ho

CC libavcodec/bethsoftvideo.ho

CC libavcodec/bgmc.ho

CC libavcodec/binkdata.ho

CC libavcodec/binkdsp.ho

CC libavcodec/bintext.ho

CC libavcodec/bitstream.ho

CC libavcodec/blockdsp.ho

CC libavcodec/bmp.ho

CC libavcodec/bsf.ho

CC libavcodec/bsf\_internal.ho

CC libavcodec/bswapdsp.ho

CC libavcodec/bytestream.ho

CC libavcodec/cabac.ho

CC libavcodec/cabac\_functions.ho

CC libavcodec/canopus.ho

CC libavcodec/cavs.ho

CC libavcodec/cavsdsp.ho

CC libavcodec/cbrt\_data.ho

CC libavcodec/cbs.ho

CC libavcodec/cbs\_av1.ho

CC libavcodec/cbs\_bsf.ho

CC libavcodec/cbs\_h264.ho

CC libavcodec/cbs\_h2645.ho

CC libavcodec/cbs\_h265.ho

CC libavcodec/cbs\_h266.ho

CC libavcodec/cbs\_internal.ho

CC libavcodec/cbs\_jpeg.ho

CC libavcodec/cbs\_mpeg2.ho

CC libavcodec/cbs\_sei.ho

CC libavcodec/cbs\_vp8.ho

CC libavcodec/cbs\_vp9.ho

CC libavcodec/celp\_filters.ho

CC libavcodec/celp\_math.ho

CC libavcodec/cfhd.ho

CC libavcodec/cfhddsp.ho

CC libavcodec/cfhdencdsp.ho

CC libavcodec/cga\_data.ho

CC libavcodec/clearvideodata.ho

CC libavcodec/codec.ho

CC libavcodec/codec2utils.ho

CC libavcodec/codec\_desc.ho

CC libavcodec/codec\_id.ho

CC libavcodec/codec\_internal.ho

CC libavcodec/codec\_par.ho

CC libavcodec/cookdata.ho

CC libavcodec/copy\_block.ho

CC libavcodec/dca.ho

CC libavcodec/dca\_core.ho

CC libavcodec/dca\_exss.ho

CC libavcodec/dca\_lbr.ho

CC libavcodec/dca\_sample\_rate\_tab.ho

CC libavcodec/dca\_syncwords.ho

CC libavcodec/dca\_xll.ho

CC libavcodec/dcaadpcm.ho

CC libavcodec/dcadata.ho

CC libavcodec/dcadct.ho

CC libavcodec/dcadec.ho

CC libavcodec/dcadsp.ho

CC libavcodec/dcaenc.ho

CC libavcodec/dcahuff.ho

CC libavcodec/dcamath.ho

CC libavcodec/dct.ho

CC libavcodec/dct32.ho

CC libavcodec/dctref.ho

CC libavcodec/decode.ho

CC libavcodec/defs.ho

CC libavcodec/dirac.ho

CC libavcodec/dirac\_arith.ho

CC libavcodec/dirac\_dwt.ho

CC libavcodec/dirac\_vlc.ho

CC libavcodec/diracdsp.ho

CC libavcodec/diractab.ho

CC libavcodec/dnxhddata.ho

CC libavcodec/dnxhdenc.ho

CC libavcodec/dolby\_e.ho

CC libavcodec/dovi\_rpu.ho

CC libavcodec/dsd.ho

CC libavcodec/dv.ho

CC libavcodec/dv\_internal.ho

CC libavcodec/dv\_profile.ho

CC libavcodec/dv\_profile\_internal.ho

CC libavcodec/dvaudio.ho

CC libavcodec/dvbtxt.ho

CC libavcodec/dvdata.ho

CC libavcodec/dvdsub.ho

CC libavcodec/dxv.ho

CC libavcodec/dynamic\_hdr\_vivid.ho

CC libavcodec/eac3\_data.ho

CC libavcodec/eac3enc.ho

CC libavcodec/eaidct.ho

CC libavcodec/elbg.ho

CC libavcodec/elsdec.ho

CC libavcodec/encode.ho

CC libavcodec/error\_resilience.ho

CC libavcodec/evc.ho

CC libavcodec/evc\_parse.ho

CC libavcodec/evc\_ps.ho

CC libavcodec/evrcdata.ho

CC libavcodec/exif.ho

CC libavcodec/exrdsp.ho

CC libavcodec/faandct.ho

CC libavcodec/faanidct.ho

CC libavcodec/faxcompr.ho

CC libavcodec/fdctdsp.ho

CC libavcodec/ffv1.ho

CC libavcodec/fits.ho

CC libavcodec/flac.ho

CC libavcodec/flac\_parse.ho

CC libavcodec/flacdata.ho

CC libavcodec/flacdsp.ho

CC libavcodec/flacencdsp.ho

CC libavcodec/flvdec.ho

CC libavcodec/flvenc.ho

CC libavcodec/fmtconvert.ho

CC libavcodec/frame\_thread\_encoder.ho

CC libavcodec/g722.ho

CC libavcodec/g722dsp.ho

CC libavcodec/g723\_1.ho

CC libavcodec/g729.ho

CC libavcodec/g729data.ho

CC libavcodec/g729postfilter.ho

CC libavcodec/get\_bits.ho

CC libavcodec/gif.ho

CC libavcodec/golomb.ho

CC libavcodec/gsm.ho

CC libavcodec/gsmdec\_data.ho

CC libavcodec/h261.ho

CC libavcodec/h261enc.ho

CC libavcodec/h263.ho

CC libavcodec/h263data.ho

CC libavcodec/h263dec.ho

CC libavcodec/h263dsp.ho

CC libavcodec/h263enc.ho

CC libavcodec/h264.ho

CC libavcodec/h2645\_parse.ho

CC libavcodec/h2645\_sei.ho

CC libavcodec/h2645\_vui.ho

CC libavcodec/h2645data.ho

CC libavcodec/h264\_levels.ho

CC libavcodec/h264\_mvpred.ho

CC libavcodec/h264\_parse.ho

CC libavcodec/h264\_ps.ho

CC libavcodec/h264\_sei.ho

CC libavcodec/h264chroma.ho

CC libavcodec/h264data.ho

CC libavcodec/h264dec.ho

CC libavcodec/h264dsp.ho

CC libavcodec/h264idct.ho

CC libavcodec/h264pred.ho

CC libavcodec/h264qpel.ho

CC libavcodec/h265\_profile\_level.ho

CC libavcodec/h274.ho

CC libavcodec/hap.ho

CC libavcodec/hca\_data.ho

CC libavcodec/hevc.ho

CC libavcodec/hevc\_data.ho

CC libavcodec/hevc\_parse.ho

CC libavcodec/hevc\_ps.ho

CC libavcodec/hevc\_sei.ho

CC libavcodec/hevcdec.ho

CC libavcodec/hevcdsp.ho

CC libavcodec/hevcpred.ho

CC libavcodec/hpeldsp.ho

CC libavcodec/hq\_hqadata.ho

CC libavcodec/hq\_hqadsp.ho

CC libavcodec/hqx.ho

CC libavcodec/hqxdsp.ho

CC libavcodec/htmlsubtitles.ho

CC libavcodec/huffman.ho

CC libavcodec/huffyuv.ho

CC libavcodec/huffyuvdsp.ho

CC libavcodec/huffyuvencdsp.ho

CC libavcodec/hwaccel\_internal.ho

CC libavcodec/hwaccels.ho

CC libavcodec/hwconfig.ho

CC libavcodec/idctdsp.ho

CC libavcodec/iirfilter.ho

CC libavcodec/ilbcdata.ho

CC libavcodec/imcdata.ho

CC libavcodec/indeo2data.ho

CC libavcodec/indeo3data.ho

CC libavcodec/indeo4data.ho

CC libavcodec/indeo5data.ho

CC libavcodec/internal.ho

CC libavcodec/intrax8.ho

CC libavcodec/intrax8dsp.ho

CC libavcodec/intrax8huf.ho

CC libavcodec/itut35.ho

CC libavcodec/ivi.ho

CC libavcodec/ivi\_dsp.ho

CC libavcodec/jacosub.ho

CC libavcodec/jni.ho

CC libavcodec/jpeg2000.ho

CC libavcodec/jpeg2000dec.ho

CC libavcodec/jpeg2000dsp.ho

CC libavcodec/jpeg2000dwt.ho

CC libavcodec/jpeg2000htdec.ho

CC libavcodec/jpegls.ho

CC libavcodec/jpeglsdec.ho

CC libavcodec/jpegquanttables.ho

CC libavcodec/jpegtables.ho

CC libavcodec/jpegtabs.ho

CC libavcodec/jpegxl.ho

CC libavcodec/jpegxl\_parse.ho

CC libavcodec/kbdwin.ho

CC libavcodec/lagarithrac.ho

CC libavcodec/lcl.ho

CC libavcodec/leaddata.ho

CC libavcodec/leb.ho

CC libavcodec/libopenh264.ho

CC libavcodec/libopus.ho

CC libavcodec/lossless\_audiodsp.ho

CC libavcodec/lossless\_videodsp.ho

CC libavcodec/lossless\_videoencdsp.ho

CC libavcodec/lpc.ho

CC libavcodec/lpc\_functions.ho

CC libavcodec/lsp.ho

CC libavcodec/lzf.ho

CC libavcodec/lzw.ho

CC libavcodec/mathops.ho

CC libavcodec/me\_cmp.ho

CC libavcodec/mediacodec.ho

CC libavcodec/metasound\_data.ho

CC libavcodec/metasound\_twinvq\_data.ho

CC libavcodec/mjpeg.ho

CC libavcodec/mjpegdec.ho

CC libavcodec/mjpegenc.ho

CC libavcodec/mjpegenc\_common.ho

CC libavcodec/mjpegenc\_huffman.ho

CC libavcodec/mlp.ho

CC libavcodec/mlp\_parse.ho

CC libavcodec/mlpdsp.ho

CC libavcodec/mlz.ho

CC libavcodec/motion\_est.ho

CC libavcodec/mpc.ho

CC libavcodec/mpc7data.ho

CC libavcodec/mpc8data.ho

CC libavcodec/mpc8huff.ho

CC libavcodec/mpcdata.ho

CC libavcodec/mpeg12.ho

CC libavcodec/mpeg12codecs.ho

CC libavcodec/mpeg12data.ho

CC libavcodec/mpeg12dec.ho

CC libavcodec/mpeg12enc.ho

CC libavcodec/mpeg12vlc.ho

CC libavcodec/mpeg4audio.ho

CC libavcodec/mpeg4audio\_copy\_pce.ho

CC libavcodec/mpeg4audio\_sample\_rates.ho

CC libavcodec/mpeg4data.ho

CC libavcodec/mpeg4video.ho

CC libavcodec/mpeg4videodata.ho

CC libavcodec/mpeg4videodec.ho

CC libavcodec/mpeg4videodefs.ho

CC libavcodec/mpeg4videodsp.ho

CC libavcodec/mpeg4videoenc.ho

CC libavcodec/mpeg\_er.ho

CC libavcodec/mpegaudio.ho

CC libavcodec/mpegaudiodata.ho

CC libavcodec/mpegaudiodecheader.ho

CC libavcodec/mpegaudiodsp.ho

CC libavcodec/mpegaudiotab.ho

CC libavcodec/mpegaudiotabs.ho

CC libavcodec/mpegpicture.ho

CC libavcodec/mpegutils.ho

CC libavcodec/mpegvideo.ho

CC libavcodec/mpegvideodata.ho

CC libavcodec/mpegvideodec.ho

CC libavcodec/mpegvideoenc.ho

CC libavcodec/mpegvideoencdsp.ho

CC libavcodec/mqc.ho

CC libavcodec/msgsmdec.ho

CC libavcodec/msmpeg4.ho

CC libavcodec/msmpeg4\_vc1\_data.ho

CC libavcodec/msmpeg4data.ho

CC libavcodec/msmpeg4dec.ho

CC libavcodec/msmpeg4enc.ho

CC libavcodec/msrledec.ho

CC libavcodec/mss12.ho

CC libavcodec/mss2dsp.ho

CC libavcodec/mss34dsp.ho

CC libavcodec/nellymoser.ho

CC libavcodec/on2avcdata.ho

CC libavcodec/options\_table.ho

CC libavcodec/opus.ho

CC libavcodec/opus\_celt.ho

CC libavcodec/opus\_parse.ho

CC libavcodec/opus\_pvq.ho

CC libavcodec/opus\_rc.ho

CC libavcodec/opus\_silk.ho

CC libavcodec/opusdsp.ho

CC libavcodec/opusenc.ho

CC libavcodec/opusenc\_psy.ho

CC libavcodec/opusenc\_utils.ho

CC libavcodec/opustab.ho

CC libavcodec/packet.ho

CC libavcodec/packet\_internal.ho

CC libavcodec/paf.ho

CC libavcodec/parser.ho

CC libavcodec/pixblockdsp.ho

CC libavcodec/pixels.ho

CC libavcodec/png.ho

CC libavcodec/pngdsp.ho

CC libavcodec/pnm.ho

CC libavcodec/profiles.ho

CC libavcodec/proresdata.ho

CC libavcodec/proresdec.ho

CC libavcodec/proresdsp.ho

CC libavcodec/psymodel.ho

CC libavcodec/pthread\_internal.ho

CC libavcodec/put\_bits.ho

CC libavcodec/put\_golomb.ho

CC libavcodec/qcelpdata.ho

CC libavcodec/qdm2data.ho

CC libavcodec/qoi.ho

CC libavcodec/qpeldsp.ho

CC libavcodec/ra144.ho

CC libavcodec/ra288.ho

CC libavcodec/ralfdata.ho

CC libavcodec/rangecoder.ho

CC libavcodec/ratecontrol.ho

CC libavcodec/raw.ho

CC libavcodec/rectangle.ho

CC libavcodec/refstruct.ho

CC libavcodec/rl.ho

CC libavcodec/rle.ho

CC libavcodec/rnd\_avg.ho

CC libavcodec/roqvideo.ho

CC libavcodec/rtjpeg.ho

CC libavcodec/rv10dec.ho

CC libavcodec/rv10enc.ho

CC libavcodec/rv30data.ho

CC libavcodec/rv34.ho

CC libavcodec/rv34data.ho

CC libavcodec/rv34dsp.ho

CC libavcodec/rv34vlc.ho

CC libavcodec/rv40data.ho

CC libavcodec/rv40vlc2.ho

CC libavcodec/sbc.ho

CC libavcodec/sbcdec\_data.ho

CC libavcodec/sbcdsp.ho

CC libavcodec/sbcdsp\_data.ho

CC libavcodec/sbr.ho

CC libavcodec/sbrdsp.ho

CC libavcodec/scpr.ho

In file included from libavcodec/scpr.hc:1:

./libavcodec/scpr.h:217:12: warning: 'decode\_run\_p' defined but not used [-Wunused-function]

217 | static int decode\_run\_p(AVCodecContext \*avctx, uint32\_t ptype, int run,

| ^~~~~~~~~~~~

./libavcodec/scpr.h:75:12: warning: 'decode\_run\_i' defined but not used [-Wunused-function]

75 | static int decode\_run\_i(AVCodecContext \*avctx, uint32\_t ptype, int run,

| ^~~~~~~~~~~~

CC libavcodec/scpr3.ho

CC libavcodec/sei.ho

CC libavcodec/sgi.ho

CC libavcodec/sheervideodata.ho

CC libavcodec/simple\_idct.ho

CC libavcodec/sinewin.ho

CC libavcodec/sipr.ho

CC libavcodec/sipr16kdata.ho

CC libavcodec/siprdata.ho

CC libavcodec/snappy.ho

CC libavcodec/snow.ho

CC libavcodec/snow\_dwt.ho

CC libavcodec/snowdata.ho

CC libavcodec/sp5x.ho

CC libavcodec/speedhq.ho

CC libavcodec/speedhqenc.ho

CC libavcodec/speexdata.ho

CC libavcodec/startcode.ho

CC libavcodec/sunrast.ho

CC libavcodec/svq1.ho

CC libavcodec/svq1\_cb.ho

CC libavcodec/svq1\_vlc.ho

CC libavcodec/svq1enc\_cb.ho

CC libavcodec/svq1encdsp.ho

CC libavcodec/synth\_filter.ho

CC libavcodec/tak.ho

CC libavcodec/takdsp.ho

CC libavcodec/targa.ho

CC libavcodec/texturedsp.ho

CC libavcodec/thread.ho

CC libavcodec/threadframe.ho

CC libavcodec/tiff.ho

CC libavcodec/tiff\_common.ho

CC libavcodec/tiff\_data.ho

CC libavcodec/to\_upper4.ho

CC libavcodec/tpeldsp.ho

CC libavcodec/truemotion1data.ho

CC libavcodec/truespeech\_data.ho

CC libavcodec/tscc2data.ho

CC libavcodec/ttadata.ho

CC libavcodec/ttadsp.ho

CC libavcodec/ttaencdsp.ho

CC libavcodec/ttmlenc.ho

CC libavcodec/twinvq.ho

CC libavcodec/twinvq\_data.ho

CC libavcodec/ulti\_cb.ho

CC libavcodec/unary.ho

CC libavcodec/utvideo.ho

CC libavcodec/utvideodsp.ho

CC libavcodec/v210dec.ho

CC libavcodec/v210dec\_init.ho

CC libavcodec/v210enc.ho

CC libavcodec/v210enc\_init.ho

CC libavcodec/v4l2\_buffers.ho

CC libavcodec/v4l2\_context.ho

CC libavcodec/v4l2\_fmt.ho

CC libavcodec/v4l2\_m2m.ho

CC libavcodec/vbn.ho

CC libavcodec/vc1.ho

CC libavcodec/vc1\_common.ho

CC libavcodec/vc1\_pred.ho

CC libavcodec/vc1\_vlc\_data.ho

CC libavcodec/vc1acdata.ho

CC libavcodec/vc1data.ho

CC libavcodec/vc1dsp.ho

CC libavcodec/vc2enc\_dwt.ho

CC libavcodec/version.ho

CC libavcodec/version\_major.ho

CC libavcodec/videodsp.ho

CC libavcodec/vlc.ho

CC libavcodec/vorbis.ho

CC libavcodec/vorbis\_data.ho

CC libavcodec/vorbis\_enc\_data.ho

CC libavcodec/vorbis\_parser.ho

CC libavcodec/vorbis\_parser\_internal.ho

CC libavcodec/vorbisdsp.ho

CC libavcodec/vp3data.ho

CC libavcodec/vp3dsp.ho

CC libavcodec/vp4data.ho

CC libavcodec/vp56.ho

CC libavcodec/vp56data.ho

CC libavcodec/vp56dsp.ho

CC libavcodec/vp5data.ho

CC libavcodec/vp6data.ho

CC libavcodec/vp8.ho

CC libavcodec/vp89\_rac.ho

CC libavcodec/vp8data.ho

CC libavcodec/vp8dsp.ho

CC libavcodec/vp9.ho

CC libavcodec/vp9data.ho

CC libavcodec/vp9dec.ho

CC libavcodec/vp9dsp.ho

CC libavcodec/vp9shared.ho

CC libavcodec/vpx\_rac.ho

CC libavcodec/vvc.ho

CC libavcodec/wavpack.ho

CC libavcodec/wavpackenc.ho

CC libavcodec/wma.ho

CC libavcodec/wma\_common.ho

CC libavcodec/wma\_freqs.ho

CC libavcodec/wmadata.ho

CC libavcodec/wmaprodata.ho

CC libavcodec/wmavoice\_data.ho

CC libavcodec/wmv2.ho

CC libavcodec/wmv2data.ho

CC libavcodec/wmv2dec.ho

CC libavcodec/wmv2dsp.ho

CC libavcodec/wmv2enc.ho

CC libavcodec/xface.ho

CC libavcodec/xiph.ho

CC libavcodec/xvididct.ho

CC libavcodec/xwd.ho

CC libavcodec/zlib\_wrapper.ho

CC libavcodec/aarch64/cabac.ho

CC libavcodec/aarch64/idct.ho

CC libavcodec/aarch64/vp8dsp.ho

CC libavcodec/aarch64/vp9dsp\_init.ho

CC libpostproc/postprocess.ho

CC libpostproc/postprocess\_internal.ho

CC libpostproc/version.ho

CC libpostproc/version\_major.ho

CC libswresample/audioconvert.ho

CC libswresample/resample.ho

CC libswresample/swresample.ho

CC libswresample/swresample\_internal.ho

CC libswresample/version.ho

CC libswresample/version\_major.ho

CC libswscale/rgb2rgb.ho

CC libswscale/swscale.ho

CC libswscale/swscale\_internal.ho

CC libswscale/version.ho

CC libswscale/version\_major.ho

CC libavutil/adler32.ho

CC libavutil/aes.ho

CC libavutil/aes\_ctr.ho

CC libavutil/aes\_internal.ho

CC libavutil/ambient\_viewing\_environment.ho

CC libavutil/attributes.ho

CC libavutil/attributes\_internal.ho

CC libavutil/audio\_fifo.ho

CC libavutil/avassert.ho

CC libavutil/avconfig.ho

CC libavutil/avstring.ho

CC libavutil/avutil.ho

CC libavutil/base64.ho

CC libavutil/blowfish.ho

CC libavutil/bprint.ho

CC libavutil/bswap.ho

CC libavutil/buffer.ho

CC libavutil/buffer\_internal.ho

CC libavutil/camellia.ho

CC libavutil/cast5.ho

CC libavutil/channel\_layout.ho

CC libavutil/colorspace.ho

CC libavutil/common.ho

CC libavutil/cpu.ho

CC libavutil/cpu\_internal.ho

CC libavutil/crc.ho

CC libavutil/csp.ho

CC libavutil/des.ho

CC libavutil/detection\_bbox.ho

CC libavutil/dict.ho

CC libavutil/dict\_internal.ho

CC libavutil/display.ho

CC libavutil/dovi\_meta.ho

CC libavutil/downmix\_info.ho

CC libavutil/dynarray.ho

CC libavutil/emms.ho

CC libavutil/encryption\_info.ho

CC libavutil/error.ho

CC libavutil/eval.ho

CC libavutil/executor.ho

CC libavutil/ffmath.ho

CC libavutil/ffversion.ho

CC libavutil/fifo.ho

CC libavutil/file.ho

CC libavutil/file\_open.ho

CC libavutil/film\_grain\_params.ho

CC libavutil/fixed\_dsp.ho

CC libavutil/float2half.ho

CC libavutil/float\_dsp.ho

CC libavutil/frame.ho

CC libavutil/getenv\_utf8.ho

CC libavutil/half2float.ho

CC libavutil/hash.ho

CC libavutil/hdr\_dynamic\_metadata.ho

CC libavutil/hdr\_dynamic\_vivid\_metadata.ho

CC libavutil/hmac.ho

CC libavutil/hwcontext.ho

CC libavutil/hwcontext\_d3d12va\_internal.ho

CC libavutil/hwcontext\_drm.ho

CC libavutil/hwcontext\_internal.ho

CC libavutil/hwcontext\_mediacodec.ho

CC libavutil/iamf.ho

CC libavutil/imgutils.ho

CC libavutil/imgutils\_internal.ho

CC libavutil/integer.ho

CC libavutil/internal.ho

CC libavutil/intfloat.ho

CC libavutil/intmath.ho

CC libavutil/intreadwrite.ho

CC libavutil/lfg.ho

CC libavutil/libm.ho

CC libavutil/lls.ho

CC libavutil/log.ho

CC libavutil/lzo.ho

CC libavutil/macos\_kperf.ho

CC libavutil/macros.ho

CC libavutil/mastering\_display\_metadata.ho

CC libavutil/mathematics.ho

CC libavutil/md5.ho

CC libavutil/mem.ho

CC libavutil/mem\_internal.ho

CC libavutil/motion\_vector.ho

CC libavutil/murmur3.ho

CC libavutil/opt.ho

CC libavutil/parseutils.ho

CC libavutil/pca.ho

CC libavutil/pixdesc.ho

CC libavutil/pixelutils.ho

CC libavutil/pixfmt.ho

CC libavutil/qsort.ho

CC libavutil/random\_seed.ho

CC libavutil/rational.ho

CC libavutil/rc4.ho

CC libavutil/replaygain.ho

CC libavutil/reverse.ho

CC libavutil/ripemd.ho

CC libavutil/samplefmt.ho

CC libavutil/sfc64.ho

CC libavutil/sha.ho

CC libavutil/sha512.ho

CC libavutil/slicethread.ho

CC libavutil/softfloat.ho

CC libavutil/softfloat\_ieee754.ho

CC libavutil/softfloat\_tables.ho

CC libavutil/spherical.ho

CC libavutil/stereo3d.ho

CC libavutil/tablegen.ho

CC libavutil/tea.ho

CC libavutil/thread.ho

CC libavutil/threadmessage.ho

CC libavutil/time.ho

CC libavutil/time\_internal.ho

CC libavutil/timecode.ho

CC libavutil/timer.ho

CC libavutil/timestamp.ho

CC libavutil/tree.ho

CC libavutil/twofish.ho

CC libavutil/tx.ho

CC libavutil/tx\_priv.ho

CC libavutil/uuid.ho

CC libavutil/version.ho

CC libavutil/version\_major.ho

CC libavutil/video\_enc\_params.ho

CC libavutil/video\_hint.ho

CC libavutil/wchar\_filename.ho

CC libavutil/xga\_font\_data.ho

CC libavutil/xtea.ho

CC libavutil/aarch64/cpu.ho

CC libavutil/aarch64/neontest.ho

LD doc/examples/avio\_list\_dir\_g

STRIP doc/examples/avio\_list\_dir

LD doc/examples/avio\_read\_callback\_g

STRIP doc/examples/avio\_read\_callback

LD doc/examples/decode\_audio\_g

STRIP doc/examples/decode\_audio

LD doc/examples/decode\_filter\_audio\_g

STRIP doc/examples/decode\_filter\_audio

LD doc/examples/decode\_filter\_video\_g

STRIP doc/examples/decode\_filter\_video

LD doc/examples/decode\_video\_g

STRIP doc/examples/decode\_video

LD doc/examples/demux\_decode\_g

STRIP doc/examples/demux\_decode

LD doc/examples/encode\_audio\_g

STRIP doc/examples/encode\_audio

LD doc/examples/encode\_video\_g

STRIP doc/examples/encode\_video

LD doc/examples/extract\_mvs\_g

STRIP doc/examples/extract\_mvs

CC doc/examples/filter\_audio.o

LD doc/examples/filter\_audio\_g

STRIP doc/examples/filter\_audio

LD doc/examples/hw\_decode\_g

STRIP doc/examples/hw\_decode

LD doc/examples/mux\_g

STRIP doc/examples/mux

LD doc/examples/remux\_g

STRIP doc/examples/remux

LD doc/examples/resample\_audio\_g

STRIP doc/examples/resample\_audio

LD doc/examples/scale\_video\_g

STRIP doc/examples/scale\_video

LD doc/examples/show\_metadata\_g

STRIP doc/examples/show\_metadata

LD doc/examples/transcode\_aac\_g

STRIP doc/examples/transcode\_aac

LD doc/examples/transcode\_g

STRIP doc/examples/transcode

CC libavfilter/tests/drawutils.o

LD libavfilter/tests/drawutils

CC libavfilter/tests/filtfmts.o

LD libavfilter/tests/filtfmts

CC libavfilter/tests/formats.o

LD libavfilter/tests/forma```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 15: fate-caf

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

```sh

make fate-caf SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST caf-demux

TEST caf-amr\_nb-remux

TEST caf-qdm2-remux

TEST caf-pcm\_s24le-remux

TEST caf-pcm\_s24-remux

TEST caf-mace6-remux

TEST caf-alac-remux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 16: fate-canopus-cllc

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

```sh

make fate-canopus-cllc SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST canopus-cllc-argb

TEST canopus-cllc-rgb

TEST canopus-cllc-yuy2-noblock

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 17: fate-canopus-hq\_hqa

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

```sh

make fate-canopus-hq\_hqa SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST canopus-hq\_hqa-hq

TEST canopus-hq\_hqa-hqa

TEST canopus-hq\_hqa-inter

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 18: fate-canopus-hqx

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

```sh

make fate-canopus-hqx SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST canopus-hqx422

TEST canopus-hqx422a

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 19: fate-cbs

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

```sh

make fate-cbs SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST cbs-av1-av1-1-b8-02-allintra

TEST cbs-av1-av1-1-b8-03-sizedown

TEST cbs-av1-av1-1-b8-03-sizeup

TEST cbs-av1-av1-1-b8-04-cdfupdate

TEST cbs-av1-av1-1-b8-05-mv

TEST cbs-av1-av1-1-b8-06-mfmv

TEST cbs-av1-av1-1-b8-22-svc-L1T2

TEST cbs-av1-av1-1-b8-22-svc-L2T1

TEST cbs-av1-av1-1-b8-22-svc-L2T2

TEST cbs-av1-av1-1-b8-23-film\_grain-50

TEST cbs-av1-av1-1-b10-23-film\_grain-50

TEST cbs-av1-decode\_model

TEST cbs-av1-frames\_refs\_short\_signaling

TEST cbs-av1-non\_uniform\_tiling

TEST cbs-av1-seq\_hdr\_op\_param\_info

TEST cbs-av1-switch\_frame

TEST cbs-h264-SVA\_Base\_B

TEST cbs-h264-BASQP1\_Sony\_C

TEST cbs-h264-FM1\_BT\_B

TEST cbs-h264-CVFC1\_Sony\_C

TEST cbs-h264-AUD\_MW\_E

TEST cbs-h264-CVBS3\_Sony\_C

TEST cbs-h264-MR1\_BT\_A

TEST cbs-h264-CVWP1\_TOSHIBA\_E

TEST cbs-h264-CVNLFI1\_Sony\_C

TEST cbs-h264-Sharp\_MP\_PAFF\_1r2

TEST cbs-h264-CVMANL1\_TOSHIBA\_B

TEST cbs-h264-sp1\_bt\_a

TEST cbs-h264-CVSE2\_Sony\_B

TEST cbs-h264-CABACI3\_Sony\_B

TEST cbs-h264-sei-1

TEST cbs-h264-discard-nonref

TEST cbs-h264-discard-bidir

TEST cbs-h264-discard-nonintra

TEST cbs-h264-discard-nonkey

TEST h264\_redundant\_pps-mov

TEST h264\_redundant\_pps-annexb

TEST h264\_redundant\_pps-side\_data

TEST h264\_redundant\_pps-side\_data2

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

TEST cbs-hevc-discard-nonref

TEST cbs-hevc-discard-bidir

TEST cbs-hevc-discard-nonintra

TEST cbs-hevc-discard-nonkey

TEST cbs-mpeg2-hhi\_burst\_422\_short

TEST cbs-mpeg2-sony-ct3

TEST cbs-mpeg2-tcela-6

TEST cbs-vp9-vp90-2-03-deltaq

TEST cbs-vp9-vp90-2-05-resize

TEST cbs-vp9-vp90-2-06-bilinear

TEST cbs-vp9-vp90-2-09-lf\_deltas

TEST cbs-vp9-vp90-2-10-show-existing-frame

TEST cbs-vp9-vp90-2-10-show-existing-frame2

TEST cbs-vp9-vp90-2-segmentation-aq-akiyo

TEST cbs-vp9-vp90-2-segmentation-sf-akiyo

TEST cbs-vp9-vp90-2-tiling-pedestrian

TEST cbs-vp9-vp91-2-04-yuv440

TEST cbs-vp9-vp91-2-04-yuv444

TEST cbs-vp9-vp92-2-20-10bit-yuv420

TEST cbs-vp9-vp93-2-20-10bit-yuv422

TEST cbs-vp9-vp93-2-20-12bit-yuv444

TEST cbs-vvc-APSALF\_A\_2

TEST cbs-vvc-APSLMCS\_D\_1

TEST cbs-vvc-APSMULT\_A\_4

TEST cbs-vvc-AUD\_A\_3

TEST cbs-vvc-BOUNDARY\_A\_3

TEST cbs-vvc-BUMP\_A\_2

TEST cbs-vvc-CodingToolsSets\_A\_2

TEST cbs-vvc-CROP\_B\_4

TEST cbs-vvc-DCI\_A\_3

TEST cbs-vvc-HRD\_A\_3

TEST cbs-vvc-OPI\_B\_3

TEST cbs-vvc-PHSH\_B\_1

TEST cbs-vvc-POC\_A\_1

TEST cbs-vvc-PPS\_B\_1

TEST cbs-vvc-RAP\_A\_1

TEST cbs-vvc-SAO\_A\_3

TEST cbs-vvc-SCALING\_A\_1

TEST cbs-vvc-SLICES\_A\_3

TEST cbs-vvc-SPS\_B\_1

TEST cbs-vvc-STILL\_B\_1

TEST cbs-vvc-SUBPIC\_A\_3

TEST cbs-vvc-TILE\_A\_2

TEST cbs-vvc-VPS\_A\_3

TEST cbs-vvc-WP\_A\_3

TEST cbs-vvc-WPP\_A\_3

TEST cbs-vvc-WRAP\_A\_4

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 20: fate-cdxl

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

```sh

make fate-cdxl SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST cdxl-pal8

TEST cdxl-pal8-small

TEST cdxl-bitline-ham6

TEST cdxl-ham6

TEST cdxl-ham8

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 21: fate-checkasm

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

```sh

make fate-checkasm SAMPLES=fate\_samples/

```

The running results are:

```sh

CC tests/checkasm/aacencdsp.o

CC tests/checkasm/aacpsdsp.o

AS tests/checkasm/aarch64/checkasm.o

CC tests/checkasm/ac3dsp.o

CC tests/checkasm/af\_afir.o

CC tests/checkasm/alacdsp.o

CC tests/checkasm/audiodsp.o

CC tests/checkasm/av\_tx.o

CC tests/checkasm/blockdsp.o

CC tests/checkasm/bswapdsp.o

CC tests/checkasm/checkasm.o

CC tests/checkasm/exrdsp.o

CC tests/checkasm/fixed\_dsp.o

CC tests/checkasm/flacdsp.o

CC tests/checkasm/float\_dsp.o

CC tests/checkasm/fmtconvert.o

CC tests/checkasm/g722dsp.o

CC tests/checkasm/h264chroma.o

tests/checkasm/h264chroma.c: In function 'check\_chroma\_mc':

tests/checkasm/h264chroma.c:54:18: warning: 'size' is used uninitialized [-Wuninitialized]

54 | for (int size = 0; size < 4; size++) {

| ^~~~

CC tests/checkasm/h264dsp.o

CC tests/checkasm/h264pred.o

CC tests/checkasm/h264qpel.o

CC tests/checkasm/hevc\_add\_res.o

CC tests/checkasm/hevc\_deblock.o

CC tests/checkasm/hevc\_idct.o

CC tests/checkasm/hevc\_pel.o

CC tests/checkasm/hevc\_sao.o

CC tests/checkasm/huffyuvdsp.o

CC tests/checkasm/idctdsp.o

CC tests/checkasm/jpeg2000dsp.o

CC tests/checkasm/llauddsp.o

CC tests/checkasm/llviddsp.o

CC tests/checkasm/llviddspenc.o

CC tests/checkasm/lpc.o

CC tests/checkasm/motion.o

CC tests/checkasm/opusdsp.o

CC tests/checkasm/pixblockdsp.o

CC tests/checkasm/rv34dsp.o

CC tests/checkasm/sbrdsp.o

CC tests/checkasm/svq1enc.o

CC tests/checkasm/sw\_gbrp.o

CC tests/checkasm/sw\_rgb.o

CC tests/checkasm/sw\_scale.o

CC tests/checkasm/synth\_filter.o

CC tests/checkasm/takdsp.o

CC tests/checkasm/utvideodsp.o

CC tests/checkasm/v210dec.o

CC tests/checkasm/v210enc.o

CC tests/checkasm/vc1dsp.o

CC tests/checkasm/vf\_blend.o

CC tests/checkasm/vf\_bwdif.o

CC tests/checkasm/vf\_colorspace.o

CC tests/checkasm/vf\_convolution.o

CC tests/checkasm/vf\_eq.o

CC tests/checkasm/vf\_gblur.o

CC tests/checkasm/vf\_hflip.o

CC tests/checkasm/vf\_nlmeans.o

CC tests/checkasm/vf\_threshold.o

CC tests/checkasm/videodsp.o

CC tests/checkasm/vorbisdsp.o

CC tests/checkasm/vp8dsp.o

CC tests/checkasm/vp9dsp.o

CC tests/checkasm/vvc\_mc.o

LD tests/checkasm/checkasm

TEST checkasm-aacencdsp

TEST checkasm-aacpsdsp

TEST checkasm-ac3dsp

TEST checkasm-af\_afir

TEST checkasm-alacdsp

TEST checkasm-audiodsp

TEST checkasm-av\_tx

TEST checkasm-blockdsp

TEST checkasm-bswapdsp

TEST checkasm-exrdsp

TEST checkasm-fixed\_dsp

TEST checkasm-flacdsp

TEST checkasm-float\_dsp

TEST checkasm-fmtconvert

TEST checkasm-g722dsp

TEST checkasm-h264chroma

TEST checkasm-h264dsp

TEST checkasm-h264pred

TEST checkasm-h264qpel

TEST checkasm-hevc\_add\_res

TEST checkasm-hevc\_deblock

TEST checkasm-hevc\_idct

TEST checkasm-hevc\_pel

TEST checkasm-hevc\_sao

TEST checkasm-huffyuvdsp

TEST checkasm-idctdsp

TEST checkasm-jpeg2000dsp

TEST checkasm-llauddsp

TEST checkasm-llviddsp

TEST checkasm-llviddspenc

TEST checkasm-lpc

TEST checkasm-motion

TEST checkasm-opusdsp

TEST checkasm-pixblockdsp

TEST checkasm-sbrdsp

TEST checkasm-rv34dsp

TEST checkasm-svq1enc

TEST checkasm-synth\_filter

TEST checkasm-sw\_gbrp

TEST checkasm-sw\_rgb

TEST checkasm-sw\_scale

TEST checkasm-takdsp

TEST checkasm-utvideodsp

TEST checkasm-v210dec

TEST checkasm-v210enc

TEST checkasm-vc1dsp

TEST checkasm-vf\_blend

TEST checkasm-vf\_bwdif

TEST checkasm-vf\_colorspace

TEST checkasm-vf\_eq

TEST checkasm-vf\_gblur

TEST checkasm-vf\_hflip

TEST checkasm-vf\_nlmeans

TEST checkasm-vf\_threshold

TEST checkasm-vf\_sobel

TEST checkasm-videodsp

TEST checkasm-vorbisdsp

TEST checkasm-vp8dsp

TEST checkasm-vp9dsp

TEST checkasm-vvc\_mc

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 22: fate-cover-art

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

```sh

make fate-cover-art SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST cover-art-ape

TEST cover-art-flac

TEST cover-art-m4a

TEST cover-art-ogg

TEST cover-art-wma

TEST cover-art-wma-id3

TEST cover-art-wma-metadatalib

TEST cover-art-wv

TEST cover-art-aiff-id3v2-remux

TEST cover-art-mp3-id3v2-remux

TEST cover-art-flac-remux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 23: fate-dca

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

```sh

make fate-dca SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST dca-xll\_51\_16\_192\_768\_0

TEST dca-xll\_51\_16\_192\_768\_0-dmix\_2

TEST dca-xll\_51\_16\_192\_768\_0-dmix\_6

TEST dca-xll\_51\_16\_192\_768\_1

TEST dca-xll\_51\_16\_192\_768\_1-dmix\_2

TEST dca-xll\_51\_16\_192\_768\_1-dmix\_6

TEST dca-xll\_51\_24\_48\_768

TEST dca-xll\_51\_24\_48\_768-dmix\_2

TEST dca-xll\_51\_24\_48\_768-dmix\_6

TEST dca-xll\_51\_24\_48\_none

TEST dca-xll\_51\_24\_48\_none-dmix\_2

TEST dca-xll\_51\_24\_48\_none-dmix\_6

TEST dca-xll\_71\_24\_48\_768\_0

TEST dca-xll\_71\_24\_48\_768\_0-dmix\_2

TEST dca-xll\_71\_24\_48\_768\_0-dmix\_6

TEST dca-xll\_71\_24\_48\_768\_1

TEST dca-xll\_71\_24\_48\_768\_1-dmix\_2

TEST dca-xll\_71\_24\_48\_768\_1-dmix\_6

TEST dca-xll\_71\_24\_96\_768

TEST dca-xll\_71\_24\_96\_768-dmix\_2

TEST dca-xll\_71\_24\_96\_768-dmix\_6

TEST dca-xll\_x96\_51\_24\_96\_1509

TEST dca-xll\_x96\_51\_24\_96\_1509-dmix\_2

TEST dca-xll\_x96\_51\_24\_96\_1509-dmix\_6

TEST dca-xll\_xch\_61\_24\_48\_768

TEST dca-xll\_xch\_61\_24\_48\_768-dmix\_2

TEST dca-xll\_xch\_61\_24\_48\_768-dmix\_6

TEST dca-core\_51\_24\_48\_768\_0

TEST dca-core\_51\_24\_48\_768\_1

TEST dca-x96\_51\_24\_96\_1509

TEST dca-x96\_xch\_61\_24\_96\_3840

TEST dca-x96\_xxch\_71\_24\_96\_3840

TEST dca-xbr\_51\_24\_48\_3840

TEST dca-xbr\_xch\_61\_24\_48\_3840

TEST dca-xbr\_xxch\_71\_24\_48\_3840

TEST dca-xch\_61\_24\_48\_768

TEST dca-xxch\_71\_24\_48\_2046

TEST dca-core\_51\_24\_48\_768\_1-dmix\_2

TEST dca-x96\_xxch\_71\_24\_96\_3840-dmix\_2

TEST dca-x96\_xxch\_71\_24\_96\_3840-dmix\_6

TEST dca-xch\_61\_24\_48\_768-dmix\_6

TEST dca-core

TEST dca-xll

TEST dts\_es

TEST dca-core-bsf

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 24: fate-demux

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

```sh

make fate-demux SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST avio-direct

TEST adts-demux

TEST adts-id3v1-demux

TEST adts-id3v2-demux

TEST adts-id3v2-two-tags-demux

TEST aa-demux

TEST aea-demux

TEST av1-annexb-demux

TEST ast

TEST bink-demux

TEST bfstm

TEST bcstm

TEST brstm

TEST cdxl-demux

TEST cine-demux

TEST d-cinema-demux

TEST d-eavp6-demux

TEST fits-demux

TEST gif-demux

TEST iv8-demux

TEST jv-demux

TEST lmlm4-demux

TEST maxis-xa

TEST mkv

TEST mkv-1242

TEST mlv-demux

TEST mov-mp3-demux

TEST mtv

TEST mxf-demux

TEST nc-demux

TEST nistsphere-demux

TEST nsv-demux

TEST oggvp8-demux

TEST paf-demux

TEST pmp-demux

TEST rsd-demux

TEST redspark-demux

TEST psx-str-demux

TEST pva-demux

TEST qcp-demux

TEST redcode-demux

TEST s337m-demux

TEST siff-demux

TEST smjpeg-demux

TEST wav-ac3

TEST westwood-aud

TEST wtv-demux

TEST xmv-demux

TEST xwma-demux

TEST cavs-demux

TEST flv-demux

TEST ts-opus-demux

TEST ts-small-demux

TEST oggopus-demux

TEST ts-demux

TEST ts-timed-id3-demux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 25: fate-dfa

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

```sh

make fate-dfa SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST dfa1

TEST dfa2

TEST dfa3

TEST dfa4

TEST dfa5

TEST dfa6

TEST dfa7

TEST dfa8

TEST dfa9

TEST dfa10

TEST dfa11

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 26: fate-dnxhd

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

```sh

make fate-dnxhd SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST dnxhr-parse

TEST dnxhr-prefix1

TEST dnxhr-prefix2

TEST dnxhr-prefix3

TEST dnxhr-prefix4

TEST dnxhr-prefix5

TEST dnxhd-mbaff

TEST dnxhr-444

TEST dnxhr-12bit

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 27: fate-dpcm

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

```sh

make fate-dpcm SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST dpcm-idroq

TEST dpcm-interplay

TEST dpcm-sierra

TEST dpcm-xan

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 28: fate-dvvideo

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

```sh

make fate-dvvideo SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST bsf-dv-error-marker

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 29: fate-ea

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

```sh

make fate-ea SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST ea-cdata

TEST ea-cmv

TEST ea-mad

TEST ea-tgq

TEST ea-tgv-1

TEST ea-tgv-2

TEST ea-tqi

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 30: fate-enc-external

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

```sh

make fate-enc-external SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST libx264-simple

TEST libx264-hdr10

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 31: fate-exif-image-embedded

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

```sh

make fate-exif-image-embedded SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST exif-image-embedded

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 32: fate-exif-image-jpg

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

```sh

make fate-exif-image-jpg SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST exif-image-jpg

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 33: fate-exif-image-tiff

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

```sh

make fate-exif-image-tiff SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST exif-image-tiff

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 34: fate-exif-image-webp

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

```sh

make fate-exif-image-webp SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST exif-image-webp

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 35: fate-ffprobe

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

```sh

make fate-ffprobe SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST lavf-mxf

TEST concat-demuxer-simple1-lavf-mxf

TEST lavf-mxf\_d10

TEST concat-demuxer-simple1-lavf-mxf\_d10

TEST lavf-ts

TEST concat-demuxer-simple2-lavf-ts

TEST concat-demuxer-extended-lavf-mxf

TEST concat-demuxer-extended-lavf-mxf\_d10

COPY tests/test\_copy.ffmeta

GEN tests/data/ffprobe-test.nut

TEST ffprobe\_compact

TEST ffprobe\_csv

TEST ffprobe\_default

TEST ffprobe\_flat

TEST ffprobe\_ini

TEST ffprobe\_json

TEST ffprobe\_xml

TEST ffprobe\_xsd

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 36: fate-fifo-muxer

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

```sh

make fate-fifo-muxer SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST fifo-muxer-tst

TEST fifo-muxer-h264

TEST fifo-muxer-wav

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 37: fate-fits

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

```sh

make fate-fits SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST fitsdec-ext\_data\_min\_max

TEST fitsdec-blank\_bitpix32

TEST fitsdec-bitpix-32

TEST fitsdec-bitpix-64

GEN tests/data/fits-multi.fits

TEST fitsdec-multi

TEST fitsdec-gray

TEST fitsdec-gbrp

TEST fitsdec-gbrp16be

TEST fitsdec-gbrap16be

TEST fitsenc-gray

TEST fitsenc-gray16be

TEST fitsenc-gbrp

TEST fitsenc-gbrap

TEST fitsenc-gbrp16be

TEST fitsenc-gbrap16be

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 38: fate-flac

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

```sh

make fate-flac SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST flac-16-chmode-indep

TEST flac-16-chmode-left\_side

TEST flac-16-chmode-mid\_side

TEST flac-16-chmode-right\_side

TEST flac-16-fixed

TEST flac-16-lpc-cholesky

TEST flac-16-lpc-levinson

TEST flac-24-comp-8

TEST flac-32-wasted-bits

TEST flac-rice-params

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 39: fate-flvenc

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

```sh

make fate-flvenc SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST flv-add\_keyframe\_index

TEST enhanced-flv-hevc

TEST enhanced-flv-vp9

TEST enhanced-flv-av1

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 40: fate-gapless

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

```sh

make fate-gapless SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST gapless-mp3

TEST audiomatch-square-mp3

TEST audiomatch-square-aac

TEST audiomatch-afconvert-16000-mono-lc-adts

TEST audiomatch-afconvert-16000-mono-lc-m4a

TEST audiomatch-afconvert-44100-mono-lc-adts

TEST audiomatch-afconvert-44100-mono-lc-m4a

TEST audiomatch-afconvert-16000-mono-he-adts

TEST audiomatch-afconvert-16000-mono-he-m4a

TEST audiomatch-afconvert-44100-mono-he-adts

TEST audiomatch-afconvert-44100-mono-he-m4a

TEST audiomatch-afconvert-16000-stereo-he-adts

TEST audiomatch-afconvert-16000-stereo-he-m4a

TEST audiomatch-afconvert-44100-stereo-he-adts

TEST audiomatch-afconvert-44100-stereo-he-m4a

TEST audiomatch-afconvert-16000-stereo-he2-adts

TEST audiomatch-afconvert-16000-stereo-he2-m4a

TEST audiomatch-afconvert-44100-stereo-he2-adts

TEST audiomatch-afconvert-44100-stereo-he2-m4a

TEST audiomatch-afconvert-16000-stereo-lc-adts

TEST audiomatch-afconvert-16000-stereo-lc-m4a

TEST audiomatch-afconvert-44100-stereo-lc-adts

TEST audiomatch-afconvert-44100-stereo-lc-m4a

TEST audiomatch-faac-16000-mono-lc-adts

TEST audiomatch-faac-16000-mono-lc-m4a

TEST audiomatch-faac-44100-mono-lc-adts

TEST audiomatch-faac-44100-mono-lc-m4a

TEST audiomatch-faac-16000-stereo-lc-adts

TEST audiomatch-faac-16000-stereo-lc-m4a

TEST audiomatch-faac-44100-stereo-lc-adts

TEST audiomatch-faac-44100-stereo-lc-m4a

TEST audiomatch-dolby-44100-mono-lc-mp4

TEST audiomatch-dolby-44100-mono-he-mp4

TEST audiomatch-dolby-44100-stereo-he-mp4

TEST audiomatch-dolby-44100-stereo-he2-mp4

TEST audiomatch-dolby-44100-stereo-lc-mp4

TEST audiomatch-nero-16000-mono-lc-m4a

TEST audiomatch-nero-44100-mono-lc-m4a

TEST audiomatch-nero-16000-mono-he-m4a

TEST audiomatch-nero-44100-mono-he-m4a

TEST audiomatch-nero-16000-stereo-he-m4a

TEST audiomatch-nero-44100-stereo-he-m4a

TEST audiomatch-nero-16000-stereo-he2-m4a

TEST audiomatch-nero-44100-stereo-he2-m4a

TEST audiomatch-nero-16000-stereo-lc-m4a

TEST audiomatch-nero-44100-stereo-lc-m4a

TEST audiomatch-quicktime7-44100-stereo-lc-mp4

TEST audiomatch-quicktimeX-44100-stereo-lc-m4a

TEST gapless-mp3-side-data

TEST gaplessinfo-itunes1

TEST gaplessinfo-itunes2

TEST gaplessenc-itunes-to-ipod-aac

TEST gaplessenc-pcm-to-mov-aac

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 41: fate-gif

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

```sh

make fate-gif SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST gif-color

TEST gif-disposal-background

TEST gif-disposal-restore

TEST gif-gray

TEST gif-deal

TEST gifenc-rgb8

TEST gifenc-bgr8

TEST gifenc-rgb4\_byte

TEST gifenc-bgr4\_byte

TEST gifenc-gray

TEST gifenc-pal8

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 42: fate-h264

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

```sh

make fate-h264 SAMPLES=fate\_samples/

```

The running results are:

```sh

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

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

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

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

TEST h264-conformance-aud\_mw\_e

TEST h264-conformance-ba1\_ft\_c

TEST h264-conformance-ba1\_sony\_d

TEST h264-conformance-ba2\_sony\_f

TEST h264-conformance-ba3\_sva\_c

TEST h264-conformance-ba\_mw\_d

TEST h264-conformance-bamq1\_jvc\_c

TEST h264-conformance-bamq2\_jvc\_c

TEST h264-conformance-banm\_mw\_d

TEST h264-conformance-basqp1\_sony\_c

TEST h264-conformance-caba1\_sony\_d

TEST h264-conformance-caba1\_sva\_b

TEST h264-conformance-caba2\_sony\_e

TEST h264-conformance-caba2\_sva\_b

TEST h264-conformance-caba3\_sony\_c

TEST h264-conformance-caba3\_sva\_b

TEST h264-conformance-caba3\_toshiba\_e

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

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

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

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

TEST h264-conformance-cabaci3\_sony\_b

TEST h264-conformance-cabast3\_sony\_e

TEST h264-conformance-cabastbr3\_sony\_b

TEST h264-conformance-cabref3\_sand\_d

TEST h264-conformance-cacqp3\_sony\_d

TEST h264-conformance-cafi1\_sva\_c

TEST h264-conformance-cama1\_sony\_c

TEST h264-conformance-cama1\_toshiba\_b

TEST h264-conformance-cama1\_vtc\_c

TEST h264-conformance-cama2\_vtc\_b

TEST h264-conformance-cama3\_sand\_e

TEST h264-conformance-cama3\_vtc\_b

TEST h264-conformance-camaci3\_sony\_c

TEST h264-conformance-camanl1\_toshiba\_b

TEST h264-conformance-camanl2\_toshiba\_b

TEST h264-conformance-camanl3\_sand\_e

TEST h264-conformance-camasl3\_sony\_b

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

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

TEST h264-conformance-canl1\_sony\_e

TEST h264-conformance-canl1\_sva\_b

TEST h264-conformance-canl1\_toshiba\_g

TEST h264-conformance-canl2\_sony\_e

TEST h264-conformance-canl2\_sva\_b

TEST h264-conformance-canl3\_sony\_c

TEST h264-conformance-canl3\_sva\_b

TEST h264-conformance-canl4\_sva\_b

TEST h264-conformance-canlma2\_sony\_c

TEST h264-conformance-canlma3\_sony\_c

TEST h264-conformance-capa1\_toshiba\_b

TEST h264-conformance-capama3\_sand\_f

TEST h264-conformance-capcm1\_sand\_e

TEST h264-conformance-capcmnl1\_sand\_e

TEST h264-conformance-capm3\_sony\_d

TEST h264-conformance-caqp1\_sony\_b

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

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

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

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

TEST h264-conformance-cawp1\_toshiba\_e

TEST h264-conformance-cawp5\_toshiba\_e

TEST h264-conformance-ci1\_ft\_b

TEST h264-conformance-ci\_mw\_d

TEST h264-conformance-cvbs3\_sony\_c

TEST h264-conformance-cvcanlma2\_sony\_c

TEST h264-conformance-cvfc1\_sony\_c

TEST h264-conformance-cvfi1\_sony\_d

TEST h264-conformance-cvfi1\_sva\_c

TEST h264-conformance-cvfi2\_sony\_h

TEST h264-conformance-cvfi2\_sva\_c

TEST h264-conformance-cvma1\_sony\_d

TEST h264-conformance-cvma1\_toshiba\_b

TEST h264-conformance-cvmanl1\_toshiba\_b

TEST h264-conformance-cvmanl2\_toshiba\_b

TEST h264-conformance-cvmapaqp3\_sony\_e

TEST h264-conformance-cvmaqp2\_sony\_g

TEST h264-conformance-cvmaqp3\_sony\_d

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

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

TEST h264-conformance-cvnlfi1\_sony\_c

TEST h264-conformance-cvnlfi2\_sony\_h

TEST h264-conformance-cvpa1\_toshiba\_b

TEST h264-conformance-cvpcmnl1\_sva\_c

TEST h264-conformance-cvpcmnl2\_sva\_c

TEST h264-conformance-cvwp1\_toshiba\_e

TEST h264-conformance-cvwp2\_toshiba\_e

TEST h264-conformance-cvwp3\_toshiba\_e

TEST h264-conformance-cvwp5\_toshiba\_e

TEST h264-conformance-fi1\_sony\_e

TEST h264-conformance-frext-alphaconformanceg

TEST h264-conformance-frext-bcrm\_freh10

TEST h264-conformance-frext-brcm\_freh11

TEST h264-conformance-frext-brcm\_freh3

TEST h264-conformance-frext-brcm\_freh4

TEST h264-conformance-frext-brcm\_freh5

TEST h264-conformance-frext-brcm\_freh8

TEST h264-conformance-frext-brcm\_freh9

TEST h264-conformance-frext-freh12\_b

TEST h264-conformance-frext-freh1\_b

TEST h264-conformance-frext-freh2\_b

TEST h264-conformance-frext-freh6```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 43: fate-happy

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

```sh

make fate-hap SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST hap1

TEST hap5

TEST hapy

TEST hap-chunk

TEST hapqa-nosnappy-127x71

TEST hapqa-snappy1-127x71

TEST hapqa-snappy16-127x71

TEST hap-alpha-only-nosnappy-128x72

TEST hap-alpha-only-snappy-127x71

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 44: fate-hevc

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

```sh

make fate-hevc SAMPLES=fate\_samples/

```

The running results are:

```sh

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

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

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

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

TEST hevc-conformance-AMP\_D\_Hisilicon

TEST hevc-conformance-AMP\_E\_Hisilicon

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

TEST hevc-paramchange-yuv420p-yuv420p10

GEN tests/data/hevc-mp4.mov

TEST hevc-bsf-mp4toannexb

TEST hevc-skiploopfilter

TEST hevc-extradata-reload

TEST hevc-two-first-slice

TEST hevc-cabac-tudepth

TEST hevc-small422chroma

TEST hevc-paired-fields

TEST hevc-monochrome-crop

TEST hevc-afd-tc-sei

TEST hevc-hdr10-plus-metadata

TEST hevc-hdr-vivid-metadata

TEST hevc-dv-rpu

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 45: fate-hlsenc

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

```sh

make fate-hlsenc SAMPLES=fate\_samples/

```

The running results are:

```sh

GEN tests/data/live\_no\_endlist.m3u8

TEST hls-live-no-endlist

GEN tests/data/live\_last\_endlist.m3u8

TEST hls-live-last-endlist

GEN tests/data/live\_endlist.m3u8

TEST hls-live-endlist

GEN tests/data/hls\_segment\_size.m3u8

TEST hls-segment-size

GEN tests/data/hls\_segment\_single.m3u8

TEST hls-segment-single

GEN tests/data/hls\_init\_time.m3u8

TEST hls-init-time

GEN tests/data/hls\_list\_size.m3u8

TEST hls-list-size

GEN tests/data/hls\_fmp4.m3u8

TEST hls-fmp4

GEN tests/data/hls\_fmp4\_ac3.m3u8

TEST hls-fmp4\_ac3

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 46: fate-iamf

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

```sh

make fate-iamf SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST iamf-stereo

TEST iamf-5\_1\_4

TEST iamf-7\_1\_4

TEST iamf-ambisonic\_1

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 47: fate-id3v2

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

```sh

make fate-id3v2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST id3v2-priv

TEST id3v2-priv-remux

TEST id3v2-chapters

TEST id3v2-utf16-bom

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 48: fate-image

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

```sh

make fate-image SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST aliaspix-bgr

TEST aliaspix-gray

TEST brenderpix-defpal

TEST brenderpix-intpal

TEST brenderpix-24

TEST brenderpix-565

TEST brenderpix-y400a

TEST dds-alpha8

TEST dds-argb

TEST dds-argb-aexp

TEST dds-dx10-bc1

TEST dds-dx10-bc1a

TEST dds-dx10-bc2

TEST dds-dx10-bc3

TEST dds-dx10-bc4

TEST dds-dx10-bc5

TEST dds-dxt1

TEST dds-dxt1a

TEST dds-dxt1-normalmap

TEST dds-dxt2

TEST dds-dxt3

TEST dds-dxt4

TEST dds-dxt5

TEST dds-dxt5-aexp

TEST dds-dxt5-normalmap

TEST dds-dxt5-normalmap-ati

TEST dds-dxt5-rbxg

TEST dds-dxt5-rgxb

TEST dds-dxt5-rxbg

TEST dds-dxt5-rxgb

TEST dds-dxt5-xgbr

TEST dds-dxt5-xgxr

TEST dds-dxt5-xrbg

TEST dds-dxt5-ycocg

TEST dds-dxt5-ycocg-scaled

TEST dds-monob

TEST dds-pal

TEST dds-pal-ati

TEST dds-rgb1555

TEST dds-rgb16

TEST dds-rgb24

TEST dds-rgb555

TEST dds-rgba

TEST dds-rgtc1s

TEST dds-rgtc1u

TEST dds-rgtc2s

TEST dds-rgtc2u

TEST dds-rgtc2u-xy

TEST dds-uyvy

TEST dds-xbgr

TEST dds-xrgb

TEST dds-y

TEST dds-ya

TEST dds-ycocg

TEST dds-yuyv

TEST exr-rgb-scanline-pxr24-half-uint32-13x9

TEST exr-slice-raw

TEST exr-slice-rle

TEST exr-slice-zip1

TEST exr-slice-zip16

TEST exr-slice-pxr24

TEST exr-rgb-scanline-pxr24-float-12x8

TEST exr-rgba-multiscanline-half-b44

TEST exr-rgb-scanline-float-b44

TEST exr-rgb-scanline-half-b44-12x8

TEST exr-rgb-scanline-half-b44-13x9

TEST exr-rgb-tile-float-raw-12x8

TEST exr-rgb-tile-float-raw-150x130

TEST exr-rgb-tile-half-raw-12x8

TEST exr-rgba-scanline-float-half-b44-13x9-l1

TEST exr-rgba-scanline-float-half-b44-13x9-l2

TEST exr-rgba-scanline-float-half-b44-12x8-l1

TEST exr-rgba-scanline-float-half-b44-12x8-l2

TEST exr-rgba-scanline-float-half-b44a-12x8-l1

TEST exr-rgba-scanline-float-half-b44a-12x8-l2

TEST exr-rgba-scanline-float-half-b44a-13x9-l1

TEST exr-rgba-scanline-float-half-b44a-13x9-l2

TEST exr-rgb-tile-pxr24-float-half-l1

TEST exr-rgb-tile-pxr24-float-half-l2

TEST exr-rgb-tile-pxr24-half-float-l1

TEST exr-rgb-tile-pxr24-half-float-l2

TEST exr-rgb-tile-half-float-b44-12x8-l1

TEST exr-rgb-tile-half-float-b44-12x8-l2

TEST exr-rgb-tile-zip-half-float-l1

TEST exr-rgb-tile-zip-half-float-l2

TEST exr-rgb-tile-zip1-half-float-l1

TEST exr-rgb-tile-zip1-half-float-l2

TEST exr-rgb-tile-rle-half-float-l1

TEST exr-rgb-tile-rle-half-float-l2

TEST exr-rgb-tile-raw-half-float-l1

TEST exr-rgb-tile-raw-half-float-l2

TEST exr-rgb-scanline-b44-half-float-12x8-l1

TEST exr-rgb-scanline-b44-half-float-12x8-l2

TEST exr-rgb-scanline-pxr24-half-float-l1

TEST exr-rgb-scanline-pxr24-half-float-l2

TEST exr-rgb-scanline-pxr24-float-half-l1

TEST exr-rgb-scanline-pxr24-float-half-l2

TEST exr-rgb-scanline-zip-half-float-l1

TEST exr-rgb-scanline-zip-half-float-l2

TEST exr-rgb-scanline-zip1-half-float-l1

TEST exr-rgb-scanline-zip1-half-float-l2

TEST exr-rgb-scanline-rle-half-float-l1

TEST exr-rgb-scanline-rle-half-float-l2

TEST exr-rgb-scanline-raw-half-float-l1

TEST exr-rgb-scanline-raw-half-float-l2

TEST exr-rgb-scanline-b44-uint32

TEST exr-rgb-scanline-pxr24-uint32

TEST exr-rgb-scanline-zip1-half-float-l1-zero-offsets

TEST exr-rgb-scanline-half-piz-bw

TEST exr-rgb-scanline-half-piz-color

TEST exr-rgb-scanline-half-piz-dw-t01

TEST exr-rgb-scanline-float-piz-48x32

TEST exr-rgb-scanline-none-negative-red

TEST exr-rgb-b44a-half-negative-4x4

TEST exr-y-tile-zip-half-12x8

TEST exr-y-scanline-zip-half-12x8

TEST exr-rgb-scanline-half-piz-dw-t08

TEST exr-rgba-zip16-16x32-flag4

TEST exr-ya-scanline-zip-half-12x8

TEST exr-rgb-tile-half-zip

TEST exr-rgb-scanline-float-zip-dw-large

TEST exr-rgb-scanline-half-piz-dw-large

TEST exr-rgb-scanline-half-zip-dw-large

TEST exr-rgb-scanline-uint32-piz-dw-large

TEST exr-rgb-tile-half-piz-dw-large

TEST exr-rgb-tile-uint32-piz-dw-large

TEST exr-rgb-scanline-half-zip-dw-outside

TEST exr-rgb-tile-half-zip-dw-outside

TEST exr-rgb-scanline-zip-half-0x0-0xFFFF

TEST jpg-12bpp

TEST jpg-jfif

TEST jpg-rgb-baseline

TEST jpg-rgb-progressive

TEST jpg-rgb-221

TEST jpg-rgb-1

TEST jpg-rgb-2

TEST jpg-rgb-3

TEST jpg-rgb-4

TEST jpg-rgb-5

TEST jpegls-2bpc

TEST jpegls-3bpc

TEST jpegls-5bpc

TEST jpegls-7bpc

TEST png-gray8

TEST png-gray16

TEST png-rgb24

TEST png-rgb48

TEST png-rgba

TEST png-rgba64

TEST png-ya8

TEST png-ya16

TEST png-int-rgb24

TEST psd-gray8

TEST psd-gray16

TEST psd-rgb24

TEST psd-rgb48

TEST psd-rgba

TEST psd-rgba64

TEST psd-ya8

TEST psd-ya16

TEST psd-lena-127x127-rgb24

TEST psd-lena-rgb-rle-127x127-16b

TEST psd-lena-rgb-rle-127x127-8b

TEST psd-lena-rgba-rle-128x128-8b

TEST psd-lena-256c

TEST psd-lena-bitmap

TEST psd-duo-tone-color

TEST sgi-gray8

TEST sgi-gray8-rle

TEST sgi-gray16

TEST sgi-gray16-rle

TEST sgi-rgb24

TEST sgi-rgb24-rle

TEST sgi-rgb48

TEST sgi-rgb48-rle

TEST sgi-rgba

TEST sgi-rgba-rle

TEST sgi-rgba64

TEST sgi-rgba64-rle

TEST sunraster-1bit-raw

TEST sunraster-1bit-rle

TEST sunraster-8bit-raw

TEST sunraster-8bit-rle

TEST sunraster-24bit-raw

TEST sunraster-24bit-rle

TEST sunraster-8bit\_gray-raw

TEST targa-conformance-CBW8

TEST targa-conformance-CCM8

TEST targa-conformance-CTC16

TEST targa-conformance-CTC24

TEST targa-conformance-CTC32

TEST targa-conformance-UBW8

TEST targa-conformance-UCM8

TEST targa-conformance-UTC16

TEST targa-conformance-UTC24

TEST targa-conformance-UTC32

TEST targa-top-to-bottom

TEST tiff-zip-rgbf32le

TEST tiff-zip-rgbaf32le

TEST tiff-fax-g3

TEST tiff-fax-g3s

TEST tiff-uncompressed-rgbf32le

TEST tiff-uncompressed-rgbaf32le

TEST tiff-lzw-rgbf32le

TEST tiff-lzw-rgbaf32le

TEST webp-rgb-lena-lossless-rgb24

TEST webp-rgb-lossless

TEST webp-rgb-lena-lossless

TEST webp-rgba-lossless

TEST webp-rgb-lossless-palette-predictor

TEST webp-rgb-lossy-q80

TEST webp-rgba-lossy-q80

TEST xbm10

TEST xbm11

TEST bmpparser

TEST dpx

TEST dpxparser

TEST pict

TEST pictor

TEST pngparser

TEST ptx

TEST xface

TEST png-frame-metadata

TEST png-side-data

TEST dpx-probe

TEST jpg-icc

TEST png-icc

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 49: fate-imfdec

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

```sh

make fate-imfdec SAMPLES=fate\_samples/

```

The running results are:

```sh

make: \*\*\* No rule to make target 'fate-imf-cpl-with-repeat', needed by 'fate-imfdec'. Stop.

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 50: fate-indeo

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

```sh

make fate-indeo 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 indeo2-delta

TEST indeo2-intra

TEST indeo3-1

TEST indeo3-2

TEST indeo4

TEST indeo5

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 51: fate-jpeg2000dec

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

```sh

make fate-jpeg2000dec SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST jpeg2000dec-p0\_01

TEST jpeg2000dec-p0\_02

TEST jpeg2000dec-p0\_03

TEST jpeg2000dec-p0\_04

TEST jpeg2000dec-p0\_05

TEST jpeg2000dec-p0\_09

TEST jpeg2000dec-p0\_11

TEST jpeg2000dec-p0\_12

TEST jpeg2000dec-p0\_14

TEST jpeg2000dec-p0\_15

TEST jpeg2000dec-p0\_16

TEST jpeg2000dec-ds0\_ht\_01\_b11

TEST jpeg2000dec-p0\_07

TEST jpeg2000dec-p0\_08

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 52: fate-jxl-anim-demux

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

```sh

make fate-jxl-anim-demux SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST jxl-anim-demux-newton

TEST jxl-anim-demux-icos4d

TEST jxl-anim-demux-belgium

TEST jxl-anim-demux-lenna256

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 53: fate-jxl-parse

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

```sh

make fate-jxl-parse SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST jxl-small-ext-box

TEST jxl-multiframe-permuted-toc

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 54: fate-lavf-audio

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

```sh

make fate-lavf-audio SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST lavf-aiff

TEST lavf-al

TEST lavf-au

TEST lavf-caf

TEST lavf-mmf

TEST lavf-ogg

TEST lavf-ul

TEST lavf-ircam

TEST lavf-s16.voc

TEST lavf-wav

TEST lavf-peak.wav

TEST lavf-peak\_only.wav

TEST lavf-w64

TEST lavf-tta

TEST lavf-mka

TEST lavf-ast

TEST lavf-dfpwm

TEST lavf-rso

TEST lavf-sox

TEST lavf-voc

TEST lavf-wv

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 55: fate-lavf-container

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

```sh

make fate-lavf-container SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST lavf-asf

TEST lavf-avi

TEST lavf-dv

TEST lavf-dv\_pal

TEST lavf-dv\_ntsc

TEST lavf-flv

TEST lavf-flm

TEST lavf-gxf

TEST lavf-gxf\_pal

TEST lavf-gxf\_ntsc

TEST lavf-mkv

TEST lavf-mkv\_attachment

TEST lavf-mov

TEST lavf-mov\_rtphint

TEST lavf-ismv

TEST lavf-mp4

TEST lavf-mpg

TEST lavf-mxf\_ffv1

TEST lavf-mxf

TEST lavf-mxf\_dv25

TEST lavf-mxf\_dvcpro50

TEST lavf-mxf\_dvcpro100

TEST lavf-mxf\_d10

TEST lavf-mxf\_opatom

TEST lavf-mxf\_opatom\_audio

TEST lavf-nut

TEST lavf-rm

TEST lavf-smjpeg

TEST lavf-swf

TEST lavf-ts

TEST lavf-wtv

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 56: fate-lavf-image2pipe

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

```sh

make fate-lavf-image2pipe SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST lavf-pbmpipe

TEST lavf-pgmpipe

TEST lavf-ppmpipe

TEST lavf-qoipipe

TEST lavf-xbmpipe

TEST lavf-xwdpipe

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 57: fate-lavf-images

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

```sh

make fate-lavf-images SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST lavf-bmp

TEST lavf-dpx

TEST lavf-gbrp10le.dpx

TEST lavf-gbrp12le.dpx

TEST lavf-rgb48le.dpx

TEST lavf-rgb48le\_10.dpx

TEST lavf-rgba64le.dpx

TEST lavf-none.grayf32le.exr

TEST lavf-none.gbrpf32le.exr

TEST lavf-none.gbrapf32le.exr

TEST lavf-rle.grayf32le.exr

TEST lavf-rle.gbrpf32le.exr

TEST lavf-rle.gbrapf32le.exr

TEST lavf-zip1.grayf32le.exr

TEST lavf-zip1.gbrpf32le.exr

TEST lavf-zip1.gbrapf32le.exr

TEST lavf-zip16.grayf32le.exr

TEST lavf-zip16.gbrpf32le.exr

TEST lavf-zip16.gbrapf32le.exr

TEST lavf-jpg

TEST lavf-pam

TEST lavf-rgba.pam

TEST lavf-gray.pam

TEST lavf-gray16be.pam

TEST lavf-rgb48be.pam

TEST lavf-monob.pam

TEST lavf-pcx

TEST lavf-grayf32le.pfm

TEST lavf-gbrpf32le.pfm

TEST lavf-grayf32be.pfm

TEST lavf-gbrpf32be.pfm

TEST lavf-pgm

TEST lavf-png

TEST lavf-gray16be.png

TEST lavf-rgb48be.png

TEST lavf-ppm

TEST lavf-sgi

TEST lavf-sun

TEST lavf-tga

TEST lavf-tiff

TEST lavf-qoi

TEST lavf-wbmp

TEST lavf-xbm

TEST lavf-xwd

TEST lavf-rgba.xwd

TEST lavf-rgb565be.xwd

TEST lavf-rgb555be.xwd

TEST lavf-rgb8.xwd

TEST lavf-rgb4\_byte.xwd

TEST lavf-gray.xwd

TEST lavf-monow.xwd

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 58: fate-lavf-video

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

```sh

make fate-lavf-video SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST lavf-y4m

TEST lavf-apng

TEST lavf-apng.png

TEST lavf-gray.fits

TEST lavf-gray16be.fits

TEST lavf-gbrp.fits

TEST lavf-gbrap.fits

TEST lavf-gbrp16be.fits

TEST lavf-gbrap16be.fits

TEST lavf-gif

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 59: fate-libavcodec

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

```sh

make fate-libavcodec SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST avpacket

TEST bitstream-be

TEST bitstream-le

TEST cabac

TEST celp\_math

TEST codec\_desc

TEST golomb

TEST idct8x8-0

TEST idct8x8-1

TEST idct8x8-2

TEST idct248

TEST dct8x8

TEST h264-levels

TEST h265-levels

TEST iirfilter

TEST mpeg12framerate

TEST rangecoder

TEST mathops

TEST j2k-dwt

TEST libavcodec-avcodec

TEST libavcodec-huffman

TEST libavcodec-htmlsubtitles

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 60: fate-libavformat

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

```sh

make fate-libavformat SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST seek\_utils

TEST noproxy

TEST srtp

TEST url

TEST movenc

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 61: fate-libavutil

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

```sh

make fate-libavutil SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST adler32

TEST aes

TEST aes\_ctr

TEST camellia

TEST cast5

TEST channel\_layout

TEST audio\_fifo

TEST avstring

TEST base64

TEST blowfish

TEST bprint

TEST cpu

cpu\_flags(raw) = 0x00000160

cpu\_flags\_str(raw) = armv8 neon dotprod

cpu\_flags(effective) = 0x00000160

cpu\_flags\_str(effective) = armv8 neon dotprod

threads = 1 (cpu\_count = 96)

TEST crc

TEST color\_utils

TEST

TEST dict

TEST encryption-info

TEST eval

TEST fifo

TEST hash

TEST hmac

TEST imgutils

TEST integer

TEST lfg

TEST md5

TEST murmur3

TEST parseutils

TEST pixfmt\_best

TEST display

TEST random\_seed

TEST ripemd

TEST sha

TEST sha512

TEST side\_data\_array

TEST tree

TEST twofish

TEST xtea

TEST tea

TEST opt

TEST uuid

TEST cpu\_init

TEST pixelutils

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 62: fate-libswscale

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

```sh

make fate-libswscale SAMPLES=fate\_samples/

```

The running results are:

```sh

CC libswscale/tests/pixdesc\_query.o

LD libswscale/tests/pixdesc\_query

TEST sws-pixdesc-query

CC libswscale/tests/floatimg\_cmp.o

LD libswscale/tests/floatimg\_cmp

TEST sws-floatimg-cmp

GEN tests/data/vsynth1.yuv

TEST sws-yuv-colorspace

TEST sws-yuv-range

CC tools/scale\_slice\_test.o

CC tools/decode\_simple.o

LD tools/scale\_slice\_test

TEST sws-slice-yuv422-12bit-rgb48

TEST sws-slice-bgr0-nv12

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 63: fate-lossless-audio

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

```sh

make fate-lossless-audio SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST lossless-alac

TEST lossless-meridianaudio

TEST ralf

TEST lossless-shorten

TEST lossless-rka

TEST lossless-osq

TEST lossless-tak

TEST lossless-tta

TEST lossless-tta-encrypted

TEST lossless-wma

TEST lossless-wma24-1

TEST lossless-wma24-2

TEST lossless-wma24-rawtile

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 64: fate-lossless-monkeysaudio

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

```sh

make fate-lossless-monkeysaudio SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST lossless-monkeysaudio-380-normal

TEST lossless-monkeysaudio-380-extrahigh

TEST lossless-monkeysaudio-388-normal

TEST lossless-monkeysaudio-388-extrahigh

TEST lossless-monkeysaudio-389b1-normal

TEST lossless-monkeysaudio-389b1-extrahigh

TEST lossless-monkeysaudio-391b1-normal

TEST lossless-monkeysaudio-391b1-extrahigh

TEST lossless-monkeysaudio-392b2-normal

TEST lossless-monkeysaudio-392b2-extrahigh

TEST lossless-monkeysaudio-394b1-normal

TEST lossless-monkeysaudio-394b1-extrahigh

TEST lossless-monkeysaudio-399

TEST lossless-monkeysaudio-legacy

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 65: fate-lossless-video

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

```sh

make fate-lossless-video SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST lagarith-rgb32

TEST lagarith-rgb24

TEST lagarith-yuy2

TEST lagarith-yv12

TEST lagarith-red

TEST lagarith-ticket4119

TEST lagarith-ticket4119-cfr

TEST lagarith-ticket4119-vfr

TEST lagarith-ticket4119-pass

TEST loco-rgb

TEST loco-yuy2

TEST msrle-8bit

TEST

TEST vble

TEST zerocodec

TEST zlib

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 66: fate-matroska

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

```sh

make fate-matroska SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST matroska-prores-zlib

TEST matroska-prores-header-insertion-bz2

TEST matroska-remux

TEST matroska-xiph-lacing

TEST matroska-wavpack-missing-codecprivate

TEST matroska-zlib-decompression

TEST matroska-lzo-decompression

TEST matroska-flac-channel-mapping

TEST matroska-flac-extradata-update

TEST webm-av1-extradata-update

TEST matroska-move-cues-to-front

TEST matroska-avoid-negative-ts

TEST matroska-ms-mode

TEST matroska-qt-mode

TEST matroska-pgs-remux

TEST matroska-pgs-remux-durations

TEST matroska-dvbsub-remux

TEST matroska-spherical-mono

TEST matroska-side-data-pref-codec

TEST matroska-side-data-pref-packet

TEST matroska-alac-remux

TEST webm-dash-chapters

TEST matroska-zero-length-block

TEST matroska-non-rotation-displaymatrix

TEST matroska-dovi-write-config7

TEST matroska-dovi-write-config8

TEST matroska-spherical-mono-remux

TEST matroska-mastering-display-metadata

TEST matroska-h264-remux

TEST matroska-vp8-alpha-remux

TEST matroska-mpegts-remux

TEST matroska-opus-remux

TEST matroska-ogg-opus-remux

TEST matroska-encoding-delay

TEST matroska-stereo\_mode

TEST webm-webvtt-remux

TEST webm-hdr10-plus-remux

TEST matroska-hdr10-plus-remux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 67: fate-microsoft

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

```sh

make fate-microsoft SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST mss2-rgb555

TEST mss2-rgb555s

TEST mss2-pal

TEST mss2-pals

TEST mss2-wmv

TEST mss2-region

TEST mts2-xesc

TEST msvideo1-8bit

TEST msvideo1-16bit

TEST wmv3-drm-dec

TEST wmv3-drm-nodec

TEST vc1\_sa00040

TEST vc1\_sa00050

TEST vc1\_sa10091

TEST vc1\_sa10143

TEST vc1\_sa20021

TEST vc1\_ilaced\_twomv

TEST vc1test\_smm0005

TEST vc1test\_smm0015

TEST vc1-ism

TEST msmpeg4v1

TEST mss1-pal

TEST mts2

TEST wmv8-x8intra

TEST asf-repldata

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 68: fate-mov

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

```sh

make fate-mov SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST mov-3elist

TEST mov-3elist-1ctts

TEST mov-1elist-1ctts

TEST mov-1elist-noctts

TEST mov-elist-starts-ctts-2ndsample

TEST mov-1elist-ends-last-bframe

TEST mov-2elist-elist1-ends-bframe

TEST mov-3elist-encrypted

TEST mov-frag-encrypted

TEST mov-tenc-only-encrypted

TEST mov-invalid-elst-entry-count

TEST mov-gpmf-remux

TEST mov-ibi-elst-starts-b

TEST mov-elst-ends-betn-b-and-i

TEST mov-frag-overlap

TEST mov-neg-firstpts-discard-frames

TEST mov-stream-shorter-than-movie

GEN tests/data/asynth-44100-1.wav

TEST mov-pcm-remux

COPY tests/data/filtergraphs/mov-channel-description

TEST mov-channel-description

COPY tests/data/filtergraphs/mov-mp4-pcm

TEST mov-mp4-pcm

TEST mov-mp4-pcm-float

GEN tests/data/asynth-44100-2.wav

COPY tests/data/streamgroups/audio\_element-stereo

COPY tests/data/streamgroups/mix\_presentation-stereo

TEST mov-mp4-iamf-stereo

GEN tests/data/asynth-44100-10.wav

COPY tests/data/filtergraphs/iamf\_5\_1\_4

COPY tests/data/streamgroups/audio\_element-5\_1\_4

COPY tests/data/streamgroups/mix\_presentation-5\_1\_4

TEST mov-mp4-iamf-5\_1\_4

GEN tests/data/asynth-44100-12.wav

COPY tests/data/filtergraphs/iamf\_7\_1\_4

COPY tests/data/streamgroups/audio\_element-7\_1\_4

COPY tests/data/streamgroups/mix\_presentation-7\_1\_4

TEST mov-mp4-iamf-7\_1\_4

GEN tests/data/asynth-44100-4.wav

COPY tests/data/filtergraphs/iamf\_ambisonic\_1

COPY tests/data/streamgroups/audio\_element-ambisonic\_1

COPY tests/data/streamgroups/mix\_presentation-ambisonic\_1

TEST mov-mp4-iamf-ambisonic\_1

TEST mov-neg-firstpts-discard

TEST mov-neg-firstpts-discard-vorbis

TEST mov-aac-2048-priming

TEST mov-zombie

TEST mov-init-nonkeyframe

TEST mov-displaymatrix

TEST mov-read-amve

TEST mov-spherical-mono

TEST mov-guess-delay-1

TEST mov-guess-delay-2

TEST mov-guess-delay-3

TEST mov-mp4-with-mov-in24-ver

TEST mov-mp4-extended-atom

TEST mov-faststart-4gb-overflow

TEST mov-440hz-10ms

TEST mov-bbi-elst-starts-b

TEST mov-avif-demux-still-image-1-item

TEST mov-avif-demux-still-image-multiple-items

TEST mov-heic-demux-still-image-1-item

TEST mov-heic-demux-still-image-multiple-items

TEST mov-mp4-chapters

TEST mov-cover-image

TEST mov-mp4-ttml-stpp

TEST mov-mp4-ttml-dfxp

TEST mov-heic-demux-still-image-grid

TEST mov-heic-demux-still-image-iovl

TEST mov-heic-demux-still-image-iovl-2

TEST mov-mp4-disposition-mpegts-remux

TEST mov-write-amve

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 69: fate-mp3

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

```sh

make fate-mp3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST mp3-float-conf-compl

TEST mp3-float-conf-he\_32khz

TEST mp3-float-conf-he\_44khz

TEST mp3-float-conf-he\_48khz

TEST mp3-float-conf-hecommon

TEST mp3-float-conf-si

TEST mp3-float-conf-si\_block

TEST mp3-float-extra\_overread

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 70: fate-mpc

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

```sh

make fate-mpc SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST mpc7-demux

TEST mpc8-demux

TEST musepack7

TEST musepack8

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 71: fate-mpeg4

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

```sh

make fate-mpeg4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST mpeg4-resolution-change-down-down

TEST mpeg4-resolution-change-down-up

TEST mpeg4-resolution-change-up-down

TEST mpeg4-resolution-change-up-up

TEST mpeg4-bsf-unpack-bframes

TEST mpeg4-packed

TEST mpeg4-simple-studio-profile

TEST m4v

TEST m4v-cfr

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 72: fate-mpegps

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

```sh

make fate-mpegps SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST mpegps-remuxed-pcm-demux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 73: fate-mpegts

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

```sh

make fate-mpegts SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST mpegts-probe-latm

TEST mpegts-probe-program

TEST mpegts-probe-pmt-merge

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 74: fate-mxf

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

```sh

make fate-mxf SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST mxf-missing-index-demux

TEST mxf-essencegroup-demux

TEST mxf-multiple-components-demux

TEST mxf-metadata-source-ref1

TEST mxf-metadata-source-ref2

TEST mxf-reel\_name

TEST mxf-user-comments

TEST mxf-opatom-user-comments

TEST mxf-probe-d10

TEST mxf-probe-dnxhd

TEST mxf-probe-j2k

TEST mxf-probe-dv25

TEST mxf-probe-applehdr10

TEST mxf-remux-applehdr10

TEST mxf-d10-user-comments

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 75: fate-oma

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

```sh

make fate-oma SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST oma-demux

TEST oma-atrac3-remux

TEST oma-atrac3p-remux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 76: fate-opus

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

```sh

make fate-opus SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST opus-testvector01

TEST opus-testvector11

TEST opus-tron.6ch.tinypkts

TEST opus-testvector05

TEST opus-testvector06

TEST opus-testvector02

TEST opus-testvector03

TEST opus-testvector04

TEST opus-silk-lbrr

TEST opus-testvector07

TEST opus-testvector08

TEST opus-testvector09

TEST opus-testvector10

TEST opus-testvector12

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 77: fate-pcm

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

```sh

make fate-pcm SAMPLES=fate\_samples/

```

The running results are:

```sh

GEN tests/data/asynth-96000-6.wav

TEST dcinema-encode

GEN tests/data/asynth-96000-1.wav

TEST pcm\_dvd-16-1-96000

TEST iff-pcm

TEST pcm\_dvd

TEST pcm-planar

TEST pcm\_s16be-stereo

TEST pcm\_s16le-stereo

TEST pcm\_u8-mono

TEST pcm\_u8-stereo

TEST w64

TEST pcm\_dvd-24-7.1-48000

TEST pcm\_dvd-16-7.1-48000

TEST pcm\_dvd-16-5.1-96000

TEST pcm\_dvd-24-5.1-48000

TEST pcm\_dvd-16-5.1-48000

TEST pcm\_dvd-24-2-48000

TEST pcm\_dvd-16-2-48000

TEST pcm\_dvd-24-1-48000

TEST pcm\_dvd-16-1-48000

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 78: fate-pixfmt

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

```sh

make fate-pixfmt SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST pixfmt-bgr24

TEST pixfmt-gray

TEST pixfmt-monob

TEST pixfmt-monow

TEST pixfmt-rgb24

TEST pixfmt-rgb32

TEST pixfmt-rgb555

TEST pixfmt-rgb565

TEST pixfmt-xyz12le

TEST pixfmt-yuv410p

TEST pixfmt-yuv411p

TEST pixfmt-yuv420p

TEST pixfmt-yuv422p

TEST pixfmt-yuv440p

TEST pixfmt-yuv444p

TEST pixfmt-yuvj420p

TEST pixfmt-yuvj422p

TEST pixfmt-yuvj440p

TEST pixfmt-yuvj444p

TEST pixfmt-yuyv422

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 79: fate-pixlet

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

```sh

make fate-pixlet SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST pixlet-rgb

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 80: fate-probe-format

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

```sh

make fate-probe-format SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST probe-format-roundup997

TEST probe-format-roundup1383

TEST probe-format-roundup1414

TEST probe-format-roundup2015

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 81: fate-prores

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

```sh

make fate-prores SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST prores-422

TEST prores-422\_hq

TEST prores-422\_lt

TEST prores-422\_proxy

TEST prores-alpha

TEST prores-alpha\_skip

TEST prores-transparency

TEST prores-transparency\_skip

TEST prores-gray

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 82: fate-qoa

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

```sh

make fate-qoa SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST qoa-152

TEST qoa-278

TEST qoa-303

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 83: fate-qt

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

```sh

make fate-qt SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST 8bps

TEST qdm2

TEST qt-alaw-mono

TEST qt-alaw-stereo

TEST qt-ima4-mono

TEST qt-ima4-stereo

TEST qt-mac3-mono

TEST qt-mac3-stereo

TEST qt-mac6-mono

TEST qt-mac6-stereo

TEST qt-ulaw-mono

TEST qt-ulaw-stereo

TEST quickdraw

TEST rpza

TEST svq1

TEST svq1-headerswap

TEST svq3-1

TEST svq3-watermark

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 84: fate-qtrle

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

```sh

make fate-qtrle SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST qtrle-1bit

TEST qtrle-24bit

TEST qtrle-2bit

TEST qtrle-4bit

TEST qtrle-8bit

TEST qtrle-16bit

TEST qtrle-32bit

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 85: fate-real

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

```sh

make fate-real SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST ra3-144

TEST ra-144

TEST ra-288

TEST ra-cook

TEST sipr-5k0

TEST sipr-6k5

TEST sipr-8k5

TEST sipr-16k

TEST rv20-1239

TEST rv30

TEST rv40

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 86: fate-screen

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

```sh

make fate-screen SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST fmvc-type1

TEST fmvc-type2

TEST fraps-v3

TEST fraps-v0

TEST fraps-v1

TEST fraps-v2

TEST fraps-v4

TEST fraps-v5

TEST rscc-8bit

TEST iscc

TEST rscc-16bit

TEST rscc-24bit

TEST rscc-32bit

TEST screenpresso-16bit

TEST screenpresso-24bit

TEST screenpresso-32bit

TEST tscc-15bit

TEST tscc-32bit

TEST tscc2-avi

TEST tscc2-mov

TEST vmnc-16bit

TEST vmnc-32bit

TEST zmbv-8bit

TEST zmbv-15bit

TEST zmbv-16bit

TEST zmbv-32bit

TEST cscd

TEST dxtory

TEST fic-avi

TEST g2m2

TEST g2m3

TEST g2m4

TEST tdsc

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 87: fate-seek

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

```sh

make fate-seek SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST acodec-adpcm-ima\_qt

TEST seek-acodec-adpcm-ima\_qt

TEST acodec-adpcm-ima\_qt-trellis

TEST seek-acodec-adpcm-ima\_qt-trellis

TEST acodec-adpcm-ima\_wav

TEST seek-acodec-adpcm-ima\_wav

TEST acodec-adpcm-ima\_wav-trellis

TEST seek-acodec-adpcm-ima\_wav-trellis

TEST acodec-adpcm-ms

TEST seek-acodec-adpcm-ms

TEST acodec-adpcm-ms-trellis

TEST seek-acodec-adpcm-ms-trellis

TEST acodec-adpcm-swf

TEST seek-acodec-adpcm-swf

TEST acodec-adpcm-swf-trellis

TEST seek-acodec-adpcm-swf-trellis

TEST acodec-adpcm-yamaha

TEST seek-acodec-adpcm-yamaha

TEST acodec-adpcm-yamaha-trellis

TEST seek-acodec-adpcm-yamaha-trellis

TEST acodec-alac

TEST seek-acodec-alac

TEST acodec-flac

TEST seek-acodec-flac

TEST acodec-mp2

TEST seek-acodec-mp2

TEST acodec-pcm-alaw

TEST seek-acodec-pcm-alaw

TEST acodec-pcm-mulaw

TEST seek-acodec-pcm-mulaw

TEST acodec-pcm-s8

TEST seek-acodec-pcm-s8

TEST acodec-pcm-u8

TEST seek-acodec-pcm-u8

TEST acodec-pcm-s16be

TEST seek-acodec-pcm-s16be

TEST acodec-pcm-s16le

TEST seek-acodec-pcm-s16le

TEST acodec-pcm-s24be

TEST seek-acodec-pcm-s24be

TEST acodec-pcm-s24le

TEST seek-acodec-pcm-s24le

TEST acodec-pcm-s32be

TEST seek-acodec-pcm-s32be

TEST acodec-pcm-s32le

TEST seek-acodec-pcm-s32le

TEST acodec-pcm-f32be

TEST seek-acodec-pcm-f32be

TEST acodec-pcm-f32le

TEST seek-acodec-pcm-f32le

TEST acodec-pcm-f64be

TEST seek-acodec-pcm-f64be

TEST acodec-pcm-f64le

TEST seek-acodec-pcm-f64le

TEST lavf-aiff

TEST seek-lavf-aiff

TEST lavf-al

TEST seek-lavf-al

TEST lavf-au

TEST seek-lavf-au

TEST lavf-mmf

TEST seek-lavf-mmf

TEST lavf-ogg

TEST seek-lavf-ogg

TEST lavf-ul

TEST seek-lavf-ul

TEST lavf-voc

TEST seek-lavf-voc

TEST lavf-wav

TEST seek-lavf-wav

TEST lavf-asf

TEST seek-lavf-asf

TEST lavf-avi

TEST seek-lavf-avi

TEST lavf-dv

TEST seek-lavf-dv

TEST lavf-flv

TEST seek-lavf-flv

TEST lavf-gxf

TEST seek-lavf-gxf

TEST lavf-mkv

TEST seek-lavf-mkv

TEST lavf-mov

TEST seek-lavf-mov

TEST lavf-mpg

TEST seek-lavf-mpg

TEST lavf-mxf

TEST seek-lavf-mxf

TEST lavf-mxf\_d10

TEST seek-lavf-mxf\_d10

TEST lavf-mxf\_dv25

TEST seek-lavf-mxf\_dv25

TEST lavf-mxf\_dvcpro50

TEST seek-lavf-mxf\_dvcpro50

TEST lavf-mxf\_opatom

TEST seek-lavf-mxf\_opatom

TEST lavf-mxf\_opatom\_audio

TEST seek-lavf-mxf\_opatom\_audio

TEST lavf-nut

TEST seek-lavf-nut

TEST lavf-swf

TEST seek-lavf-swf

TEST lavf-ts

TEST seek-lavf-ts

TEST lavf-wtv

TEST seek-lavf-wtv

TEST lavf-rm

TEST seek-lavf-rm

TEST lavf-gif

TEST seek-lavf-gif

TEST lavf-y4m

TEST seek-lavf-y4m

TEST lavf-bmp

TEST seek-lavf-bmp

TEST lavf-jpg

TEST seek-lavf-jpg

TEST lavf-pcx

TEST seek-lavf-pcx

TEST lavf-pgm

TEST seek-lavf-pgm

TEST lavf-ppm

TEST seek-lavf-ppm

TEST lavf-sgi

TEST seek-lavf-sgi

TEST lavf-tga

TEST seek-lavf-tga

TEST lavf-tiff

TEST seek-lavf-tiff

TEST lavf-pbmpipe

TEST seek-lavf-pbmpipe

TEST lavf-pgmpipe

TEST seek-lavf-pgmpipe

TEST lavf-ppmpipe

TEST seek-lavf-ppmpipe

HOSTCC tests/rotozoom.o

HOSTLD tests/rotozoom

GEN tests/data/vsynth\_lena.yuv

TEST vsynth\_lena-asv1

TEST seek-vsynth\_lena-asv1

TEST vsynth\_lena-asv2

TEST seek-vsynth\_lena-asv2

TEST vsynth\_lena-dnxhd-720p

TEST seek-vsynth\_lena-dnxhd-720p

TEST vsynth\_lena-dnxhd-720p-rd

TEST seek-vsynth\_lena-dnxhd-720p-rd

TEST vsynth\_lena-dnxhd-1080i

TEST seek-vsynth\_lena-dnxhd-1080i

TEST vsynth\_lena-dnxhd-4k-hr-lb

TEST seek-vsynth\_lena-dnxhd-4k-hr-lb

TEST vsynth\_lena-dv

TEST seek-vsynth\_lena-dv

TEST vsynth\_lena-dv-411

TEST seek-vsynth\_lena-dv-411

TEST vsynth\_lena-dv-50

TEST seek-vsynth\_lena-dv-50

TEST vsynth\_lena-ffv1

TEST seek-vsynth\_lena-ffv1

TEST vsynth\_lena-flashsv

TEST seek-vsynth\_lena-flashsv

TEST vsynth\_lena-flv

TEST seek-vsynth\_lena-flv

TEST vsynth\_lena-h261

TEST seek-vsynth\_lena-h261

TEST vsynth\_lena-h263

TEST seek-vsynth\_lena-h263

TEST vsynth\_lena-h263p

TEST seek-vsynth\_lena-h263p

TEST vsynth\_lena-huffyuv

TEST seek-vsynth\_lena-huffyuv

TEST vsynth\_lena-jpegls

TEST seek-vsynth\_lena-jpegls

TEST vsynth\_lena-ljpeg

TEST seek-vsynth\_lena-ljpeg

TEST vsynth\_lena-mjpeg

TEST seek-vsynth\_lena-mjpeg

TEST vsynth\_lena-mpeg1

TEST seek-vsynth\_lena-mpeg1

TEST vsynth\_lena-mpeg1b

TEST seek-vsynth\_lena-mpeg1b

TEST vsynth\_lena-mpeg2-422

TEST seek-vsynth\_lena-mpeg2-422

TEST vsynth\_lena-mpeg2-idct-int

TEST seek-vsynth\_lena-mpeg2-idct-int

TEST vsynth\_lena-mpeg2-ilace

TEST seek-vsynth\_lena-mpeg2-ilace

TEST vsynth\_lena-mpeg2-ivlc-qprd

TEST seek-vsynth\_lena-mpeg2-ivlc-qprd

TEST vsynth\_lena-mpeg2-thread

TEST seek-vsynth\_lena-mpeg2-thread

TEST vsynth\_lena-mpeg2-thread-ivlc

TEST seek-vsynth\_lena-mpeg2-thread-ivlc

TEST vsynth\_lena-mpeg4

TEST seek-vsynth\_lena-mpeg4

TEST vsynth\_lena-mpeg4-rc

TEST seek-vsynth\_lena-mpeg4-rc

TEST vsynth\_lena-mpeg4-adv

TEST seek-vsynth\_lena-mpeg4-adv

TEST vsynth\_lena-mpeg4-qprd

TEST seek-vsynth\_lena-mpeg4-qprd

TEST vsynth\_lena-mpeg4-adap

TEST seek-vsynth\_lena-mpeg4-adap

TEST vsynth\_lena-mpeg4-qpel

TEST seek-vsynth\_lena-mpeg4-qpel

TEST vsynth\_lena-mpeg4-thread

TEST seek-vsynth\_lena-mpeg4-thread

TEST vsynth\_lena-mpeg4-error

TEST seek-vsynth\_lena-mpeg4-error

TEST vsynth\_lena-mpeg4-nr

TEST seek-vsynth\_lena-mpeg4-nr

TEST vsynth\_lena-mpeg4-nsse

TEST seek-vsynth\_lena-mpeg4-nsse

TEST vsynth\_lena-msmpeg4

TEST seek-vsynth\_lena-msmpeg4

TEST vsynth\_lena-msmpeg4v2

TEST seek-vsynth\_lena-msmpeg4v2

TEST vsynth\_lena-rgb

TEST seek-vsynth\_lena-rgb

TEST vsynth\_lena-roqvideo

TEST seek-vsynth\_lena-roqvideo

TEST vsynth\_lena-rv10

TEST seek-vsynth\_lena-rv10

TEST vsynth\_lena-rv20

TEST seek-vsynth\_lena-rv20

TEST vsynth\_lena-snow

TEST seek-vsynth\_lena-snow

TEST vsynth\_lena-snow-ll

TEST seek-vsynth\_lena-snow-ll

TEST vsynth\_lena-svq1

TEST seek-vsynth\_lena-svq1

TEST vsynth\_lena-wmv1

TEST seek-vsynth\_lena-wmv1

TEST vsynth\_lena-wmv2

TEST seek-vsynth\_lena-wmv2

TEST vsynth\_lena-yuv

TEST seek-vsynth\_lena-yuv

TEST seek-extra-mp3

TEST seek-cache-pipe

TEST seek-mkv-codec-delay

TEST seek-extra-mp4

TEST seek-empty-edit-mp4

TEST seek-test-iibbibb-mp4

TEST seek-test-iibbibb-neg-ctts-mp4

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 88: fate-segafilm

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

```sh

make fate-segafilm SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST segafilm-adx-remux

TEST segafilm-s8-remux

TEST segafilm-cinepak-mux

TEST segafilm-rawvideo-mux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 89: fate-segment

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

```sh

make fate-segment SAMPLES=fate\_samples/

```

The running results are:

```sh

GEN tests/data/mp4-to-ts.m3u8

TEST segment-mp4-to-ts

GEN tests/data/adts-to-mkv.m3u8

TEST segment-adts-to-mkv

GEN tests/data/adts-to-mkv-header.mkv

GEN tests/data/adts-to-mkv-cated-000.mkv

TEST segment-adts-to-mkv-header-000

GEN tests/data/adts-to-mkv-cated-001.mkv

TEST segment-adts-to-mkv-header-001

GEN tests/data/adts-to-mkv-cated-002.mkv

TEST segment-adts-to-mkv-header-002

GEN tests/data/adts-to-mkv-cated-all.mkv

TEST segment-adts-to-mkv-header-all

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 90: fate-source

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

```sh

make fate-source SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST source

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 91: fate-spdif

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

```sh

make fate-spdif SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST spdif-dca-core-bswap

TEST spdif-dca-master

TEST spdif-dca-master-core

TEST spdif-eac3

TEST spdif-mlp

TEST spdif-truehd

TEST spdif-aac-remux

TEST spdif-ac3-remux

TEST spdif-dca-core-remux

TEST spdif-dca-master-core-remux

TEST spdif-eac3-remux

TEST spdif-mp2-remux

TEST spdif-mp3-remux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 92: fate-speedhq

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

```sh

make fate-speedhq SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST speedhq-422

TEST speedhq-422-singlefield

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 93: fate-subtitles

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

```sh

make fate-subtitles SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST binsub-mksenc

TEST sub-jacosub-remux

TEST sub-lrc-remux

TEST sub-microdvd-remux

TEST binsub-movtextenc

TEST sub-pgs-remux

TEST sub-srt-rrn-remux

TEST sub-srt-madness-timeshift

TEST sub-srt-empty-events

TEST sub-subripenc

TEST sub-webvttenc

TEST sub-text

TEST sub-scc

TEST sub-scc-remux

TEST sub-rcwt

TEST sub-dvb

TEST sub-ttmlenc

TEST sub-aqtitle

TEST sub-cc

TEST sub-cc-realtime

TEST sub-cc-scte20

TEST sub-ass-to-ass-transcode

TEST sub-ssa-to-ass-remux

TEST sub-jacosub

TEST sub-microdvd

TEST sub-movtext

TEST sub-mpl2

TEST sub-mpsub

TEST sub-mpsub-frames

TEST sub-pjs

TEST sub-realtext

TEST sub-sami

TEST sub-sami2

TEST sub-srt

TEST sub-srt-badsyntax

TEST sub-stl

TEST sub-subviewer1

TEST sub-subviewer

TEST sub-vplayer

TEST sub-webvtt

TEST sub-webvtt2

TEST sub-charenc

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 94: fate-swr

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

```sh

make fate-swr SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST swr-resample\_async-s16p-8000-44100

TEST swr-resample\_async-s16p-44100-8000

TEST swr-resample\_async-fltp-8000-44100

TEST swr-resample\_async-fltp-44100-8000

TEST swr-resample\_exact\_async-s16p-8000-44100

TEST swr-resample\_exact\_async-s16p-8000-48000

TEST swr-resample\_exact\_async-s16p-44100-8000

TEST swr-resample\_exact\_async-s16p-44100-48000

GEN tests/data/asynth-48000-1.wav

TEST swr-resample\_exact\_async-s16p-48000-8000

TEST swr-resample\_exact\_async-s16p-48000-44100

TEST swr-resample\_exact\_async-s32p-8000-44100

TEST swr-resample\_exact\_async-s32p-8000-48000

TEST swr-resample\_exact\_async-s32p-44100-8000

TEST swr-resample\_exact\_async-s32p-44100-48000

TEST swr-resample\_exact\_async-s32p-48000-8000

TEST swr-resample\_exact\_async-s32p-48000-44100

TEST swr-resample\_exact\_async-fltp-8000-44100

TEST swr-resample\_exact\_async-fltp-8000-48000

TEST swr-resample\_exact\_async-fltp-44100-8000

TEST swr-resample\_exact\_async-fltp-44100-48000

TEST swr-resample\_exact\_async-fltp-48000-8000

TEST swr-resample\_exact\_async-fltp-48000-44100

TEST swr-resample\_exact\_async-dblp-8000-44100

TEST swr-resample\_exact\_async-dblp-8000-48000

TEST swr-resample\_exact\_async-dblp-44100-8000

TEST swr-resample\_exact\_async-dblp-44100-48000

TEST swr-resample\_exact\_async-dblp-48000-8000

TEST swr-resample\_exact\_async-dblp-48000-44100

TEST swr-resample\_exact\_lin\_async-s16p-8000-44100

TEST swr-resample\_exact\_lin\_async-s16p-8000-48000

TEST swr-resample\_exact\_lin\_async-s16p-44100-8000

TEST swr-resample\_exact\_lin\_async-s16p-44100-48000

TEST swr-resample\_exact\_lin\_async-s16p-48000-8000

TEST swr-resample\_exact\_lin\_async-s16p-48000-44100

TEST swr-resample\_exact\_lin\_async-s32p-8000-44100

TEST swr-resample\_exact\_lin\_async-s32p-8000-48000

TEST swr-resample\_exact\_lin\_async-s32p-44100-8000

TEST swr-resample\_exact\_lin\_async-s32p-44100-48000

TEST swr-resample\_exact\_lin\_async-s32p-48000-8000

TEST swr-resample\_exact\_lin\_async-s32p-48000-44100

TEST swr-resample\_exact\_lin\_async-fltp-8000-44100

TEST swr-resample\_exact\_lin\_async-fltp-8000-48000

TEST swr-resample\_exact\_lin\_async-fltp-44100-8000

TEST swr-resample\_exact\_lin\_async-fltp-44100-48000

TEST swr-resample\_exact\_lin\_async-fltp-48000-8000

TEST swr-resample\_exact\_lin\_async-fltp-48000-44100

TEST swr-resample\_exact\_lin\_async-dblp-8000-44100

TEST swr-resample\_exact\_lin\_async-dblp-8000-48000

TEST swr-resample\_exact\_lin\_async-dblp-44100-8000

TEST swr-resample\_exact\_lin\_async-dblp-44100-48000

TEST swr-resample\_exact\_lin\_async-dblp-48000-8000

TEST swr-resample\_exact\_lin\_async-dblp-48000-44100

TEST swr-async-firstpts

GEN tests/data/asynth-2626-1.wav

TEST swr-resample-s16p-2626-8000

TEST swr-resample-s16p-2626-44100

TEST swr-resample-s16p-2626-48000

TEST swr-resample-s16p-2626-96000

TEST swr-resample-s16p-8000-2626

TEST swr-resample-s16p-8000-44100

TEST swr-resample-s16p-8000-48000

TEST swr-resample-s16p-8000-96000

TEST swr-resample-s16p-44100-2626

TEST swr-resample-s16p-44100-8000

TEST swr-resample-s16p-44100-48000

TEST swr-resample-s16p-44100-96000

TEST swr-resample-s16p-48000-2626

TEST swr-resample-s16p-48000-8000

TEST swr-resample-s16p-48000-44100

TEST swr-resample-s16p-48000-96000

TEST swr-resample-s16p-96000-2626

TEST swr-resample-s16p-96000-8000

TEST swr-resample-s16p-96000-44100

TEST swr-resample-s16p-96000-48000

TEST swr-resample-s32p-2626-8000

TEST swr-resample-s32p-2626-44100

TEST swr-resample-s32p-2626-48000

TEST swr-resample-s32p-2626-96000

TEST swr-resample-s32p-8000-2626

TEST swr-resample-s32p-8000-44100

TEST swr-resample-s32p-8000-48000

TEST swr-resample-s32p-8000-96000

TEST swr-resample-s32p-44100-2626

TEST swr-resample-s32p-44100-8000

TEST swr-resample-s32p-44100-48000

TEST swr-resample-s32p-44100-96000

TEST swr-resample-s32p-48000-2626

TEST swr-resample-s32p-48000-8000

TEST swr-resample-s32p-48000-44100

TEST swr-resample-s32p-48000-96000

TEST swr-resample-s32p-96000-2626

TEST swr-resample-s32p-96000-8000

TEST swr-resample-s32p-96000-44100

TEST swr-resample-s32p-96000-48000

TEST swr-resample-fltp-2626-8000

TEST swr-resample-fltp-2626-44100

TEST swr-resample-fltp-2626-48000

TEST swr-resample-fltp-2626-96000

TEST swr-resample-fltp-8000-2626

TEST swr-resample-fltp-8000-44100

TEST swr-resample-fltp-8000-48000

TEST swr-resample-fltp-8000-96000

TEST swr-resample-fltp-44100-2626

TEST swr-resample-fltp-44100-8000

TEST swr-resample-fltp-44100-48000

TEST swr-resample-fltp-44100-96000

TEST swr-resample-fltp-48000-2626

TEST swr-resample-fltp-48000-8000

TEST swr-resample-fltp-48000-44100

TEST swr-resample-fltp-48000-96000

TEST swr-resample-fltp-96000-2626

TEST swr-resample-fltp-96000-8000

TEST swr-resample-fltp-96000-44100

TEST swr-resample-fltp-96000-48000

TEST swr-resample-dblp-2626-8000

TEST swr-resample-dblp-2626-44100

TEST swr-resample-dblp-2626-48000

TEST swr-resample-dblp-2626-96000

TEST swr-resample-dblp-8000-2626

TEST swr-resample-dblp-8000-44100

TEST swr-resample-dblp-8000-48000

TEST swr-resample-dblp-8000-96000

TEST swr-resample-dblp-44100-2626

TEST swr-resample-dblp-44100-8000

TEST swr-resample-dblp-44100-48000

TEST swr-resample-dblp-44100-96000

TEST swr-resample-dblp-48000-2626

TEST swr-resample-dblp-48000-8000

TEST swr-resample-dblp-48000-44100

TEST swr-resample-dblp-48000-96000

TEST swr-resample-dblp-96000-2626

TEST swr-resample-dblp-96000-8000

TEST swr-resample-dblp-96000-44100

TEST swr-resample-dblp-96000-48000

TEST swr-resample\_lin-s16p-8000-44100

TEST swr-resample\_lin-s16p-8000-48000

TEST swr-resample\_lin-s16p-44100-8000

TEST swr-resample\_lin-s16p-44100-48000

TEST swr-resample\_lin-s16p-48000-8000

TEST swr-resample\_lin-s16p-48000-44100

TEST swr-resample\_lin-fltp-8000-44100

TEST swr-resample\_lin-fltp-8000-48000

TEST swr-resample\_lin-fltp-44100-8000

TEST swr-resample\_lin-fltp-44100-48000

TEST swr-resample\_lin-fltp-48000-8000

TEST swr-resample\_lin-fltp-48000-44100

TEST swr-resample\_lin-dblp-8000-44100

TEST swr-resample\_lin-dblp-8000-48000

TEST swr-resample\_lin-dblp-44100-8000

TEST swr-resample\_lin-dblp-44100-48000

TEST swr-resample\_lin-dblp-48000-8000

TEST swr-resample\_lin-dblp-48000-44100

TEST swr-resample\_nn-s16p-8000-44100

TEST swr-resample\_nn-s16p-44100-8000

TEST swr-resample\_nn-fltp-8000-44100

TEST swr-resample\_nn-fltp-44100-8000

TEST swr-resample\_exact-s16p-2626-8000

TEST swr-resample\_exact-s16p-2626-44100

TEST swr-resample\_exact-s16p-2626-48000

TEST swr-resample\_exact-s16p-2626-96000

TEST swr-resample\_exact-s16p-8000-2626

TEST swr-resample\_exact-s16p-8000-44100

TEST swr-resample\_exact-s16p-8000-48000

TEST swr-resample\_exact-s16p-8000-96000

TEST swr-resample\_exact-s16p-44100-2626

TEST swr-resample\_exact-s16p-44100-8000

TEST swr-resample\_exact-s16p-44100-48000

TEST swr-resample\_exact-s16p-44100-96000

TEST swr-resample\_exact-s16p-48000-2626

TEST swr-resample\_exact-s16p-48000-8000

TEST swr-resample\_exact-s16p-48000-44100

TEST swr-resample\_exact-s16p-48000-96000

TEST swr-resample\_exact-s16p-96000-2626

TEST swr-resample\_exact-s16p-96000-8000

TEST swr-resample\_exact-s16p-96000-44100

TEST swr-resample\_exact-s16p-96000-48000

TEST swr-resample\_exact-s32p-2626-8000

TEST swr-resample\_exact-s32p-2626-44100

TEST swr-resample\_exact-s32p-2626-48000

TEST swr-resample\_exact-s32p-2626-96000

TEST swr-resample\_exact-s32p-8000-2626

TEST swr-resample\_exact-s32p-8000-44100

TEST swr-resample\_exact-s32p-8000-48000

TEST swr-resample\_exact-s32p-8000-96000

TEST swr-resample\_exact-s32p-44100-2626

TEST swr-resample\_exact-s32p-44100-8000

TEST swr-resample\_exact-s32p-44100-48000

TEST swr-resample\_exact-s32p-44100-96000

TEST swr-resample\_exact-s32p-48000-2626

TEST swr-resample\_exact-s32p-48000-8000

TEST swr-resample\_exact-s32p-48000-44100

TEST swr-resample\_exact-s32p-48000-96000

TEST swr-resample\_exact-s32p-96000-2626

TEST swr-resample\_exact-s32p-96000-8000

TEST swr-resample\_exact-s32p-96000-44100

TEST swr-resample\_exact-s32p-96000-48000

TEST swr-resample\_exact-fltp-2626-8000

TEST swr-resample\_exact-fltp-2626-44100

TEST swr-resample\_exact-fltp-2626-48000

TEST swr-resample\_exact-fltp-2626-96000

TEST swr-resample\_exact-fltp-8000-2626

TEST swr-resample\_exact-fltp-8000-44100

TEST swr-resample\_exact-fltp-8000-48000

TEST swr-resample\_exact-fltp-8000-96000

TEST swr-resample\_exact-fltp-44100-2626

TEST swr-resample\_exact-fltp-44100-8000

TEST swr-resample\_exact-fltp-44100-48000

TEST swr-resample\_exact-fltp-44100-96000

TEST swr-resample\_exact-fltp-48000-2626

TEST swr-resample\_exact-fltp-48000-8000

TEST swr-resample\_exact-fltp-48000-44100

TEST swr-resample\_exact-fltp-48000-96000

TEST swr-resample\_exact-fltp-96000-2626

TEST swr-resample\_exact-fltp-96000-8000

TEST swr-resample\_exact-fltp-96000-44100

TEST swr-resample\_exact-fltp-96000-48000

TEST swr-resample\_exact-dblp-2626-8000

TEST swr-resample\_exact-dblp-2626-44100

TEST swr-resample\_exact-dblp-2626-48000

TEST swr-resample\_exact-dblp-2626-96000

TEST swr-resample\_exact-dblp-8000-2626

TEST swr-resample\_exact-dblp-8000-44100

TEST swr-resample\_exact-dblp-8000-48000

TEST swr-resample\_exact-dblp-8000-96000

TEST swr-resample\_exact-dblp-44100-2626

TEST swr-resample\_exact-dblp-44100-8000

TEST swr-resample\_exact-dblp-44100-48000

TEST swr-resample\_exact-dblp-44100-96000

TEST swr-resample\_exact-dblp-48000-2626

TEST swr-resample\_exact-dblp-48000-8000

TEST swr-resample\_exact-dblp-48000-44100

TEST swr-resample\_exact-dblp-48000-96000

TEST swr-resample\_exact-dblp-96000-2626

TEST swr-resample\_exact-dblp-96000-8000

TEST swr-resample\_exact-dblp-96000-44100

TEST swr-resample\_exact-dblp-96000-48000

TEST swr-resample\_exact\_lin-s16p-2626-8000

TEST swr-resample\_exact\_lin-s16p-2626-44100

TEST swr-resample\_exact\_lin-s16p-2626-48000

TEST swr-resample\_exact\_lin-s16p-2626-96000

TEST swr-resample\_exact\_lin-s16p-8000-2626

TEST swr-resample\_exact\_lin-s16p-8000-44100

TEST swr-resample\_exact\_lin-s16p-8000-48000

TEST swr-resample\_exact\_lin-s16p-8000-96000

TEST swr-resample\_exact\_lin-s16p-44100-2626

TEST swr-resample\_exact\_lin-s16p-44100-8000

TEST swr-resample\_exact\_lin-s16p-44100-48000

TEST swr-resample\_exact\_lin-s16p-44100-96000

TEST swr-resample\_exact\_lin-s16p-48000-2626

TEST swr-resample\_exact\_lin-s16p-48000-8000

TEST swr-resample\_exact\_lin-s16p-48000-44100

TEST swr-resample\_exact\_lin-s16p-48000-96000

TEST swr-resample\_exact\_lin-s16p-96000-2626

TEST swr-resample\_exact\_lin-s16p-96000-8000

TEST swr-resample\_exact\_lin-s16p-96000-44100

TEST swr-resample\_exact\_lin-s16p-96000-48000

TEST swr-resample\_exact\_lin-s32p-2626-8000

TEST swr-resample\_exact\_lin-s32p-2626-44100

TEST swr-resample\_exact\_lin-s32p-2626-48000

TEST swr-resample\_exact\_lin-s32p-2626-96000

TEST swr-resample\_exact\_lin-s32p-8000-2626

TEST swr-resample\_exact\_lin-s32p-8000-44100

TEST swr-resample\_exact\_lin-s32p-8000-48000

TEST swr-resample\_exact\_lin-s32p-8000-96000

TEST swr-resample\_exact\_lin-s32p-44100-2626

TEST swr-resample\_exact\_lin-s32p-44100-8000

TEST swr-resample\_exact\_lin-s32p-44100-48000

TEST swr-resample\_exact\_lin-s32p-44100-96000

TEST swr-resample\_exact\_lin-s32p-48000-2626

TEST swr-resample\_exact\_lin-s32p-48000-8000

TEST swr-resample\_exact\_lin-s32p-48000-44100

TEST swr-resample\_exact\_lin-s32p-48000-96000

TEST swr-resample\_exact\_lin-s32p-96000-2626

TEST swr-resample\_exact\_lin-s32p-96000-8000

TEST swr-resample\_exact\_lin-s32p-96000-44100

TEST swr-resample\_exact\_lin-s32p-96000-48000

TEST swr-resample\_exact\_lin-fltp-2626-8000

TEST swr-resample\_exact\_lin-fltp-2626-44100

TEST swr-resample\_exact\_lin-fltp-2626-48000

TEST swr-resample\_exact\_lin-fltp-2626-96000

TEST swr-resample\_exact\_lin-fltp-8000-2626

TEST swr-resample\_exact\_lin-fltp-8000-44100

TEST swr-resample\_exact\_lin-fltp-8000-48000

TEST swr-resample\_exact\_lin-fltp-8000-96000

TEST swr-resample\_exact\_lin-fltp-44100-2626

TEST swr-resample\_exact\_lin-fltp-44100-8000

TEST swr-resample\_exact\_lin-fltp-44100-48000

TEST swr-resample\_exact\_lin-fltp-44100-96000

TEST swr-resample\_exact\_lin-fltp-48000-2626

TEST swr-resample\_exact\_lin-fltp-48000-8000

TEST swr-resample\_exact\_lin-fltp-48000-44100

TEST swr-resample\_exact\_lin-fltp-48000-96000

TEST swr-resample\_exact\_lin-fltp-96000-2626

TEST swr-resample\_exact\_lin-fltp-96000-8000

TEST swr-resample\_exact\_lin-fltp-96000-44100

TEST swr-resample\_exact\_lin-fltp-96000-48000

TEST swr-resample\_exact\_lin-dblp-2626-8000

TEST swr-resample\_exact\_lin-dblp-2626-44100

TEST swr-resample\_exact\_lin-dblp-2626-48000

TEST swr-resample\_exact\_lin-dblp-2626-96000

TEST swr-resample\_exact\_lin-dblp-8000-2626

TEST swr-resample\_exact\_lin-dblp-8000-44100

TEST swr-resample\_exact\_lin-dblp-8000-48000

TEST swr-resample\_exact\_lin-dblp-8000-96000

TEST swr-resample\_exact\_lin-dblp-44100-2626

TEST swr-resample\_exact\_lin-dblp-44100-8000

TEST swr-resample\_exact\_lin-dblp-44100-48000

TEST swr-resample\_exact\_lin-dblp-44100-96000

TEST swr-resample\_exact\_lin-dblp-48000-2626

TEST swr-resample\_exact\_lin-dblp-48000-8000

TEST swr-resample\_exact\_lin-dblp-48000-44100

TEST swr-resample\_exact\_lin-dblp-48000-96000

TEST swr-resample\_exact\_lin-dblp-96000-2626

TEST swr-resample\_exact\_lin-dblp-96000-8000

TEST swr-resample\_exact\_lin-dblp-96000-44100

TEST swr-resample\_exact\_lin-dblp-96000-48000

TEST swr-audioconvert

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 95: fate-truehd

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

```sh

make fate-truehd SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST truehd-5.1

TEST truehd-5.1-downmix-2.0

TEST truehd-core-bsf

TEST truehd-mono1726

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 96: fate-truemotion1

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

```sh

make fate-truemotion1 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST truemotion1-15

TEST truemotion1-24

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 97: fate-truemotion2

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

```sh

make fate-truemotion2 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST truemotion2

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 98: fate-truemotion2rt

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

```sh

make fate-truemotion2rt SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST truemotion2rt-low

TEST truemotion2rt-mid

TEST truemotion2rt-high

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 99: fate-utvideo

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

```sh

make fate-utvideo SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST utvideo\_rgb\_left

TEST utvideo\_rgb\_median

TEST utvideo\_rgba\_left

TEST utvideo\_rgba\_median

TEST utvideo\_rgb\_int\_median

TEST utvideo\_rgba\_gradient

TEST utvideo\_rgb\_int\_gradient

TEST utvideo\_rgba\_single\_symbol

TEST utvideo\_yuv420\_left

TEST utvideo\_yuv420\_median

TEST utvideo\_yuv420\_int\_median

TEST utvideo\_yuv420\_gradient

TEST utvideo\_yuv420\_int\_gradient

TEST utvideo\_yuv422\_left

TEST utvideo\_yuv422\_median

TEST utvideo\_yuv422\_int\_median

TEST utvideo\_yuv422\_gradient

TEST utvideo\_yuv422\_int\_gradient

TEST utvideo\_yuv444\_709\_median

TEST utvideo\_yuv444\_709\_int\_median

TEST utvideo\_yuv444\_709\_gradient

TEST utvideo\_yuv444\_709\_int\_gradient

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 100: fate-utvideoenc

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

```sh

make fate-utvideoenc SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST utvideoenc\_rgba\_left

TEST utvideoenc\_rgba\_median

TEST utvideoenc\_rgba\_none

TEST utvideoenc\_rgb\_left

TEST utvideoenc\_rgb\_median

TEST utvideoenc\_rgb\_none

TEST utvideoenc\_yuv420\_left

TEST utvideoenc\_yuv420\_median

TEST utvideoenc\_yuv420\_none

TEST utvideoenc\_yuv422\_left

TEST utvideoenc\_yuv422\_median

TEST utvideoenc\_yuv422\_none

TEST utvideoenc\_yuv444\_left

TEST utvideoenc\_yuv444\_median

TEST utvideoenc\_yuv444\_none

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 101: fate-vbn

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

```sh

make fate-vbn SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST vbn-dxt1

TEST vbn-dxt5

TEST vbn-raw-rgba

TEST vbn-raw-rgb24

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 102: fate-vcodec

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

```sh

make fate-vcodec SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST vsynth1-asv1

TEST vsynth1-asv2

TEST vsynth1-vc2-420p

TEST vsynth1-ffv1

TEST vsynth1-ffv1-v0

TEST vsynth1-ffv1-v3-yuv420p

TEST vsynth1-ffv1-2pass

TEST vsynth1-ffvhuff

TEST vsynth1-flv

TEST vsynth1-h261

TEST vsynth1-h261-trellis

TEST vsynth1-h263

TEST vsynth1-h263-obmc

TEST vsynth1-h263p

TEST vsynth1-ljpeg

TEST vsynth1-mpeg1

TEST vsynth1-mpeg1b

TEST vsynth1-mpeg2

TEST vsynth1-mpeg2-422

TEST vsynth1-mpeg2-idct-int

TEST vsynth1-mpeg2-ilace

TEST vsynth1-mpeg2-ivlc-qprd

TEST vsynth1-mpeg2-thread

TEST vsynth1-mpeg2-thread-ivlc

TEST vsynth1-mpeg4

TEST vsynth1-mpeg4-rc

TEST vsynth1-mpeg4-adv

TEST vsynth1-mpeg4-qprd

TEST vsynth1-mpeg4-adap

TEST vsynth1-mpeg4-qpel

TEST vsynth1-mpeg4-thread

TEST vsynth1-mpeg4-error

TEST vsynth1-mpeg4-nr

TEST vsynth1-mpeg4-nsse

TEST vsynth1-msmpeg4

TEST vsynth1-msmpeg4v2

TEST vsynth1-rv10

TEST vsynth1-rv20

TEST vsynth1-snow-ll

TEST vsynth1-speedhq-420p

TEST vsynth1-wmv1

TEST vsynth1-wmv2

TEST vsynth1-yuv

TEST vsynth1-yuv4

TEST vsynth1-amv

TEST vsynth1-cinepak

TEST vsynth1-cljr

TEST vsynth1-vc2-420p10

TEST vsynth1-vc2-420p12

TEST vsynth1-vc2-422p

TEST vsynth1-vc2-422p10

TEST vsynth1-vc2-422p12

TEST vsynth1-vc2-444p

TEST vsynth1-vc2-444p10

TEST vsynth1-vc2-444p12

TEST vsynth1-vc2-thaar

TEST vsynth1-vc2-t5\_3

TEST vsynth1-dnxhd-720p

TEST vsynth1-dnxhd-720p-rd

TEST vsynth1-dnxhd-720p-10bit

TEST vsynth1-dnxhd-720p-hr-lb

TEST vsynth1-dnxhd-edge1-hr

TEST vsynth1-dnxhd-edge2-hr

TEST vsynth1-dnxhd-edge3-hr

TEST vsynth1-dnxhd-4k-hr-lb

TEST vsynth1-dnxhd-2k-hr-hq

TEST vsynth1-dnxhd-uhd-hr-sq

TEST vsynth1-dnxhd-1080i

TEST vsynth1-dnxhd-1080i-10bit

TEST vsynth1-dnxhd-1080i-colr

TEST vsynth1-dnxhd-hr-lb-mov

TEST vsynth1-dnxhd-hr-sq-mov

TEST vsynth1-dnxhd-hr-hq-mov

TEST vsynth1-dv

TEST vsynth1-dv-411

TEST vsynth1-dv-50

TEST vsynth1-dv-hd

TEST vsynth1-dv-fhd

TEST vsynth1-ffv1-v3-yuv422p10

TEST vsynth1-ffv1-v3-yuv444p16

TEST vsynth1-ffv1-v3-bgr0

TEST vsynth1-ffv1-v3-rgb48

TEST vsynth1-ffvhuff444

TEST vsynth1-ffvhuff420p12

TEST vsynth1-ffvhuff422p10left

TEST vsynth1-ffvhuff444p16

TEST vsynth1-flashsv

TEST vsynth1-flashsv2

TEST vsynth1-huffyuv

TEST vsynth1-huffyuvbgr24

TEST vsynth1-huffyuvbgra

TEST vsynth1-jpegls

TEST vsynth1-jpeg2000

TEST vsynth1-jpeg2000-97

TEST vsynth1-jpeg2000-gbrp12

TEST vsynth1-jpeg2000-yuva444p16

TEST vsynth1-mjpeg

TEST vsynth1-mjpeg-422

TEST vsynth1-mjpeg-444

TEST vsynth1-mjpeg-trell

TEST vsynth1-mjpeg-huffman

TEST vsynth1-mjpeg-trell-huffman

TEST vsynth1-msrle

TEST vsynth1-mpng

TEST vsynth1-msvideo1

TEST vsynth1-prores

TEST vsynth1-prores\_int

TEST vsynth1-prores\_444

TEST vsynth1-prores\_444\_int

TEST vsynth1-prores\_ks

TEST vsynth1-qtrle

TEST vsynth1-qtrlegray

TEST vsynth1-rgb

TEST vsynth1-bpp1

TEST vsynth1-bpp15

TEST vsynth1-mov-bgr24

TEST vsynth1-mov-bpp15

TEST vsynth1-mov-bpp16

TEST vsynth1-roqvideo

TEST vsynth1-rpza

TEST vsynth1-smc

TEST vsynth1-snow

TEST vsynth1-snow-hpel

TEST vsynth1-speedhq-422p

TEST vsynth1-speedhq-444p

TEST vsynth1-svq1

TEST vsynth1-r210

TEST vsynth1-v210

TEST vsynth1-v210-10

TEST vsynth1-v308

TEST vsynth1-v408

TEST vsynth1-avui

TEST vsynth1-xface

TEST vsynth1-y41p

TEST vsynth1-zlib

TEST vsynth\_lena-asv1

TEST vsynth\_lena-asv2

TEST vsynth\_lena-vc2-420p

TEST vsynth\_lena-ffv1

TEST vsynth\_lena-ffv1-v0

TEST vsynth\_lena-ffv1-v3-yuv420p

TEST vsynth\_lena-ffv1-2pass

TEST vsynth\_lena-ffvhuff

TEST vsynth\_lena-flv

TEST vsynth\_lena-h261

TEST vsynth\_lena-h261-trellis

TEST vsynth\_lena-h263

TEST vsynth\_lena-h263-obmc

TEST vsynth\_lena-h263p

TEST vsynth\_lena-ljpeg

TEST vsynth\_lena-mpeg1

TEST vsynth\_lena-mpeg1b

TEST vsynth\_lena-mpeg2

TEST vsynth\_lena-mpeg2-422

TEST vsynth\_lena-mpeg2-idct-int

TEST vsynth\_lena-mpeg2-ilace

TEST vsynth\_lena-mpeg2-ivlc-qprd

TEST vsynth\_lena-mpeg2-thread

TEST vsynth\_lena-mpeg2-thread-ivlc

TEST vsynth\_lena-mpeg4

TEST vsynth\_lena-mpeg4-rc

TEST vsynth\_lena-mpeg4-adv

TEST vsynth\_lena-mpeg4-qprd

TEST vsynth\_lena-mpeg4-adap

TEST vsynth\_lena-mpeg4-qpel

TEST vsynth\_lena-mpeg4-thread

TEST vsynth\_lena-mpeg4-error

TEST vsynth\_lena-mpeg4-nr

TEST vsynth\_lena-mpeg4-nsse

TEST vsynth\_lena-msmpeg4

TEST vsynth\_lena-msmpeg4v2

TEST vsynth\_lena-rv10

TEST vsynth\_lena-rv20

TEST vsynth\_lena-snow-ll

TEST vsynth\_lena-speedhq-420p

TEST vsynth\_lena-wmv1

TEST vsynth\_lena-wmv2

TEST vsynth\_lena-yuv

TEST vsynth\_lena-yuv4

TEST vsynth\_lena-amv

TEST vsynth\_lena-cinepak

TEST vsynth\_lena-cljr

TEST vsynth\_lena-vc2-420p10

TEST vsynth\_lena-vc2-420p12

TEST vsynth\_lena-vc2-422p

TEST vsynth\_lena-vc2-422p10

TEST vsynth\_lena-vc2-422p12

TEST vsynth\_lena-vc2-444p

TEST vsynth\_lena-vc2-444p10

TEST vsynth\_lena-vc2-444p12

TEST vsynth\_lena-vc2-thaar

TEST vsynth\_lena-vc2-t5\_3

TEST vsynth\_lena-dnxhd-720p

TEST vsynth\_lena-dnxhd-720p-rd

TEST vsynth\_lena-dnxhd-720p-10bit

TEST vsynth\_lena-dnxhd-720p-hr-lb

TEST vsynth\_lena-dnxhd-edge1-hr

TEST vsynth\_lena-dnxhd-edge2-hr

TEST vsynth\_lena-dnxhd-edge3-hr

TEST vsynth\_lena-dnxhd-4k-hr-lb

TEST vsynth\_lena-dnxhd-2k-hr-hq

TEST vsynth\_lena-dnxhd-uhd-hr-sq

TEST vsynth\_lena-dnxhd-1080i

TEST vsynth\_lena-dnxhd-1080i-10bit

TEST vsynth\_lena-dnxhd-1080i-colr

TEST vsynth\_lena-dnxhd-hr-lb-mov

TEST vsynth\_lena-dnxhd-hr-sq-mov

TEST vsynth\_lena-dnxhd-hr-hq-mov

TEST vsynth\_lena-dv

TEST vsynth\_lena-dv-411

TEST vsynth\_lena-dv-50

TEST vsynth\_lena-dv-hd

TEST vsynth\_lena-dv-fhd

TEST vsynth\_lena-ffv1-v3-yuv422p10

TEST vsynth\_lena-ffv1-v3-yuv444p16

TEST vsynth\_lena-ffv1-v3-bgr0

TEST vsynth\_lena-ffv1-v3-rgb48

TEST vsynth\_lena-ffvhuff444

TEST vsynth\_lena-ffvhuff420p12

TEST vsynth\_lena-ffvhuff422p10left

TEST vsynth\_lena-ffvhuff444p16

TEST vsynth\_lena-flashsv

TEST vsynth\_lena-flashsv2

TEST vsynth\_lena-huffyuv

TEST vsynth\_lena-huffyuvbgr24

TEST vsynth\_lena-huffyuvbgra

TEST vsynth\_lena-jpegls

TEST vsynth\_lena-jpeg2000

TEST vsynth\_lena-jpeg2000-97

TEST vsynth\_lena-jpeg2000-gbrp12

TEST vsynth\_lena-jpeg2000-yuva444p16

TEST vsynth\_lena-mjpeg

TEST vsynth\_lena-mjpeg-422

TEST vsynth\_lena-mjpeg-444

TEST vsynth\_lena-mjpeg-trell

TEST vsynth\_lena-mjpeg-huffman

TEST vsynth\_lena-mjpeg-trell-huffman

TEST vsynth\_lena-msrle

TEST vsynth\_lena-mpng

TEST vsynth\_lena-msvideo1

TEST vsynth\_lena-prores

TEST vsynth\_lena-prores\_int

TEST vsynth\_lena-prores\_444

TEST vsynth\_lena-prores\_444\_int

TEST vsynth\_lena-prores\_ks

TEST vsynth\_lena-qtrle

TEST vsynth\_lena-qtrlegray

TEST vsynth\_lena-rgb

TEST vsynth\_lena-bpp1

TEST vsynth\_lena-bpp15

TEST vsynth\_lena-mov-bgr24

TEST vsynth\_lena-mov-bpp15

TEST vsynth\_lena-mov-bpp16

TEST vsynth\_lena-roqvideo

TEST vsynth\_lena-rpza

TEST vsynth\_lena-smc

TEST vsynth\_lena-snow

TEST vsynth\_lena-snow-hpel

TEST vsynth\_lena-speedhq-422p

TEST vsynth\_lena-speedhq-444p

TEST vsynth\_lena-svq1

TEST vsynth\_lena-r210

TEST vsynth\_lena-v210

TEST vsynth\_lena-v210-10

TEST vsynth\_lena-v308

TEST vsynth\_lena-v408

TEST vsynth\_lena-avui

TEST vsynth\_lena-xface

TEST vsynth\_lena-y41p

TEST vsynth\_lena-zlib

GEN tests/data/vsynth2.yuv

TEST vsynth2-asv1

TEST vsynth2-asv2

TEST vsynth2-vc2-420p

TEST vsynth2-ffv1

TEST vsynth2-ffv1-v0

TEST vsynth2-ffv1-v3-yuv420p

TEST vsynth2-ffv1-2pass

TEST vsynth2-ffvhuff

TEST vsynth2-flv

TEST vsynth2-h261

TEST vsynth2-h261-trellis

TEST vsynth2-h263

TEST vsynth2-h263-obmc

TEST vsynth2-h263p

TEST vsynth2-ljpeg

TEST vsynth2-mpeg1

TEST vsynth2-mpeg1b

TEST vsynth2-mpeg2

TEST vsynth2-mpeg2-422

TEST vsynth2-mpeg2-idct-int

TEST vsynth2-mpeg2-ilace

TEST vsynth2-mpeg2-ivlc-qprd

TEST vsynth2-mpeg2-thread

TEST vsynth2-mpeg2-thread-ivlc

TEST vsynth2-mpeg4

TEST vsynth2-mpeg4-rc

TEST vsynth2-mpeg4-adv

TEST vsynth2-mpeg4-qprd

TEST vsynth2-mpeg4-adap

TEST vsynth2-mpeg4-qpel

TEST vsynth2-mpeg4-thread

TEST vsynth2-mpeg4-error

TEST vsynth2-mpeg4-nr

TEST vsynth2-mpeg4-nsse

TEST vsynth2-msmpeg4

TEST vsynth2-msmpeg4v2

TEST vsynth2-rv10

TEST vsynth2-rv20

TEST vsynth2-snow-ll

TEST vsynth2-speedhq-420p

TEST vsynth2-wmv1

TEST vsynth2-wmv2

TEST vsynth2-yuv

TEST vsynth2-yuv4

TEST vsynth2-amv

TEST vsynth2-cinepak

TEST vsynth2-cljr

TEST vsynth2-vc2-420p10

TEST vsynth2-vc2-420p12

TEST vsynth2-vc2-422p

TEST vsynth2-vc2-422p10

TEST vsynth2-vc2-422p12

TEST vsynth2-vc2-444p

TEST vsynth2-vc2-444p10

TEST vsynth2-vc2-444p12

TEST vsynth2-vc2-thaar

TEST vsynth2-vc2-t5\_3

TEST vsynth2-dnxhd-720p

TEST vsynth2-dnxhd-720p-rd

TEST vsynth2-dnxhd-720p-10bit

TEST vsynth2-dnxhd-720p-hr-lb

TEST vsynth2-dnxhd-edge1-hr

TEST vsynth2-dnxhd-edge2-hr

TEST vsynth2-dnxhd-edge3-hr

TEST vsynth2-dnxhd-4k-hr-lb

TEST vsynth2-dnxhd-2k-hr-hq

TEST vsynth2-dnxhd-uhd-hr-sq

TEST vsynth2-dnxhd-1080i

TEST vsynth2-dnxhd-1080i-10bit

TEST vsynth2-dnxhd-1080i-colr

TEST vsynth2-dnxhd-hr-lb-mov

TEST vsynth2-dnxhd-hr-sq-mov

TEST vsynth2-dnxhd-hr-hq-mov

TEST vsynth2-dv

TEST vsynth2-dv-411

TEST vsynth2-dv-50

TEST vsynth2-dv-hd

TEST vsynth2-dv-fhd

TEST vsynth2-ffv1-v3-yuv422p10

TEST vsynth2-ffv1-v3-yuv444p16

TEST vsynth2-ffv1-v3-bgr0

TEST vsynth2-ffv1-v3-rgb48

TEST vsynth2-ffvhuff444

TEST vsynth2-ffvhuff420p12

TEST vsynth2-ffvhuff422p10left

TEST vsynth2-ffvhuff444p16

TEST vsynth2-flashsv

TEST vsynth2-flashsv2

TEST vsynth2-huffyuv

TEST vsynth2-huffyuvbgr24

TEST vsynth2-huffyuvbgra

TEST vsynth2-jpegls

TEST vsynth2-jpeg2000

TEST vsynth2-jpeg2000-97

TEST vsynth2-jpeg2000-gbrp12

TEST vsynth2-jpeg2000-yuva444p16

TEST vsynth2-mjpeg

TEST vsynth2-mjpeg-422

TEST vsynth2-mjpeg-444

TEST vsynth2-mjpeg-trell

TEST vsynth2-mjpeg-huffman

TEST vsynth2-mjpeg-trell-huffman

TEST vsynth2-msrle

TEST vsynth2-mpng

TEST vsynth2-msvideo1

TEST vsynth2-prores

TEST vsynth2-prores\_int

TEST vsynth2-prores\_444

TEST vsynth2-prores\_444\_int

TEST vsynth2-prores\_ks

TEST vsynth2-qtrle

TEST vsynth2-qtrlegray

TEST vsynth2-rgb

TEST vsynth2-bpp1

TEST vsynth2-bpp15

TEST vsynth2-mov-bgr24

TEST vsynth2-mov-bpp15

TEST vsynth2-mov-bpp16

TEST vsynth2-roqvideo

TEST vsynth2-rpza

TEST vsynth2-smc

TEST vsynth2-snow

TEST vsynth2-snow-hpel

TEST vsynth2-speedhq-422p

TEST vsynth2-speedhq-444p

TEST vsynth2-svq1

TEST vsynth2-r210

TEST vsynth2-v210

TEST vsynth2-v210-10

TEST vsynth2-v308

TEST vsynth2-v408

TEST vsynth2-avui

TEST vsynth2-xface

TEST vsynth2-y41p

TEST vsynth2-zlib

GEN tests/data/vsynth3.yuv

TEST vsynth3-asv1

TEST vsynth3-asv2

TEST vsynth3-ffv1

TEST vsynth3-ffv1-v0

TEST vsynth3-ffv1-v3-yuv420p

TEST vsynth3-ffv1-2pass

TEST vsynth3-ffvhuff

TEST vsynth3-flv

TEST vsynth3-ljpeg

TEST vsynth3-mpeg1

TEST vsynth3-mpeg1b

TEST vsynth3-mpeg2

TEST vsynth3-mpeg2-422

TEST vsynth3-mpeg2-idct-int

TEST vsynth3-mpeg2-ilace

TEST vsynth3-mpeg2-ivlc-qprd

TEST vsynth3-mpeg2-thread

TEST vsynth3-mpeg2-thread-ivlc

TEST vsynth3-mpeg4

TEST vsynth3-mpeg4-rc

TEST vsynth3-mpeg4-adv

TEST vsynth3-mpeg4-qprd

TEST vsynth3-mpeg4-adap

TEST vsynth3-mpeg4-qpel

TEST vsynth3-mpeg4-thread

TEST vsynth3-mpeg4-error

TEST vsynth3-mpeg4-nr

TEST vsynth3-mpeg4-nsse

TEST vsynth3-msmpeg4

TEST vsynth3-msmpeg4v2

TEST vsynth3-wmv1

TEST vsynth3-wmv2

TEST vsynth3-yuv

TEST vsynth3-yuv4

TEST vsynth3-amv

TEST vsynth3-cljr

TEST vsynth3-dnxhd-720p-hr-lb

TEST vsynth3-dnxhd-edge1-hr

TEST vsynth3-dnxhd-edge2-hr

TEST vsynth3-dnxhd-edge3-hr

TEST vsynth3-dnxhd-4k-hr-lb

TEST vsynth3-dnxhd-2k-hr-hq

TEST vsynth3-dnxhd-uhd-hr-sq

TEST vsynth3-dnxhd-1080i-10bit

TEST vsynth3-dnxhd-1080i-colr

TEST vsynth3-dnxhd-hr-lb-mov

TEST vsynth3-dnxhd-hr-sq-mov

TEST vsynth3-dnxhd-hr-hq-mov

TEST vsynth3-dv-hd

TEST vsynth3-dv-fhd

TEST vsynth3-ffv1-v3-yuv422p10

TEST vsynth3-ffv1-v3-yuv444p16

TEST vsynth3-ffv1-v3-bgr0

TEST vsynth3-ffv1-v3-rgb48

TEST vsynth3-ffvhuff444

TEST vsynth3-ffvhuff420p12

TEST vsynth3-ffvhuff422p10left

TEST vsynth3-ffvhuff444p16

TEST vsynth3-flashsv

TEST vsynth3-flashsv2

TEST vsynth3-huffyuv

TEST vsynth3-huffyuvbgr24

TEST vsynth3-huffyuvbgra

TEST vsynth3-jpegls

TEST vsynth3-jpeg2000

TEST vsynth3-jpeg2000-97

TEST vsynth3-jpeg2000-gbrp12

TEST vsynth3-jpeg2000-yuva444p16

TEST vsynth3-mjpeg

TEST vsynth3-mjpeg-422

TEST vsynth3-mjpeg-444

TEST vsynth3-mjpeg-trell

TEST vsynth3-mjpeg-huffman

TEST vsynth3-mjpeg-trell-huffman

TEST vsynth3-msrle

TEST vsynth3-mpng

TEST vsynth3-prores

TEST vsynth3-prores\_int

TEST vsynth3-prores\_444

TEST vsynth3-prores\_444\_int

TEST vsynth3-prores\_ks

TEST vsynth3-qtrle

TEST vsynth3-rgb

TEST vsynth3-bpp1

TEST vsynth3-bpp15

TEST vsynth3-mov-bgr24

TEST vsynth3-mov-bpp15

TEST vsynth3-mov-bpp16

TEST vsynth3-rpza

TEST vsynth3-smc

TEST vsynth3-svq1

TEST vsynth3-r210

TEST vsynth3-v210

TEST vsynth3-v210-10

TEST vsynth3-v308

TEST vsynth3-v408

TEST vsynth3-xface

TEST vsynth3-zlib

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 103: fate-vfilter

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

```sh

make fate-vfilter SAMPLES=fate\_samples/

```

The running results are:

```sh

GEN tests/pixfmts.mak

TEST filter-lavd-life

TEST filter-lavd-testsrc

TEST filter-testsrc2-yuv420p

TEST filter-testsrc2-yuv444p

TEST filter-testsrc2-rgb24

TEST filter-testsrc2-rgba

TEST filter-allrgb

TEST filter-allyuv

TEST filter-pal75bars

TEST filter-pal100bars

TEST filter-rgbtestsrc

TEST filter-smptebars

TEST filter-smptehdbars

TEST filter-yuvtestsrc-yuv444p

TEST filter-yuvtestsrc-yuv444p12

COPY tests/data/filtergraphs/scalenorm

TEST filter-lavd-scalenorm

TEST filter-framerate-up

TEST filter-framerate-down

TEST filter-framerate-12bit-up

TEST filter-framerate-12bit-down

TEST filter-minterpolate-up

TEST filter-minterpolate-down

TEST filter-tpad-add

TEST filter-tpad-clone

TEST filter-tpad-add-duration

TEST filter-untile

TEST filter-untile-yuv422p

COPY tests/data/filtergraphs/concat

TEST filter-concat

COPY tests/data/filtergraphs/concat-vfr

TEST filter-concat-vfr

TEST filter-chromashift-smear

TEST filter-chromashift-wrap

TEST filter-decimate

TEST filter-mpdecimate

TEST filter-fps-up

TEST filter-fps-up-round-down

TEST filter-fps-up-round-up

TEST filter-fps-down

TEST filter-fps-down-round-down

TEST filter-fps-down-round-up

TEST filter-fps-down-eof-pass

TEST filter-fps-start-drop

TEST filter-fps-start-fill

TEST filter-refcmp-blockdetect-yuv

TEST filter-refcmp-blurdetect-yuv

TEST filter-refcmp-siti-yuv

TEST filter-refcmp-psnr-rgb

TEST filter-refcmp-psnr-yuv

TEST filter-refcmp-ssim-rgb

TEST filter-refcmp-ssim-yuv

TEST filter-owdenoise-sample

TEST filter-delogo

TEST filter-bwdif-mode0

TEST filter-bwdif-mode1

TEST filter-bwdif10

TEST filter-yadif-mode0

TEST filter-yadif-mode1

TEST filter-yadif10

TEST filter-yadif16

TEST filter-w3fdif-simple

TEST filter-w3fdif-complex

TEST filter-mcdeint-fast

TEST filter-mcdeint-medium

TEST filter-codecview-mvs

TEST filter-showpalette

TEST filter-palettegen-1

TEST filter-palettegen-2

TEST filter-paletteuse-nodither

TEST filter-paletteuse-bayer

TEST filter-paletteuse-bayer0

TEST filter-paletteuse-sierra2\_4a

COPY tests/data/filtergraphs/overlay-dvdsub-2397

TEST filter-overlay-dvdsub-2397

TEST filter-overlay\_yuv420\_yuva420

TEST filter-overlay\_yuv422\_yuva422

TEST filter-overlay\_yuv444\_yuva444

TEST filter-overlay\_gbrp\_gbrap

TEST filter-overlay\_yuva420\_yuva420

TEST filter-overlay\_yuva422\_yuva422

TEST filter-overlay\_yuva444\_yuva444

TEST filter-overlay\_gbrap\_gbrap

TEST filter-overlay\_rgb\_rgba

TEST filter-overlay\_rgba\_rgba

COPY tests/data/filtergraphs/hqdn3d

TEST filter-hqdn3d-sample

TEST filter-ep2x

TEST filter-ep3x

TEST filter-hq2x

TEST filter-hq3x

TEST filter-hq4x

TEST filter-2xbr

TEST filter-3xbr

TEST filter-4xbr

TEST filter-curves

COPY tests/data/filtergraphs/gradfun

TEST filter-gradfun-sample

TEST filter-fps-cfr

TEST filter-fps

COPY tests/data/maps/fsync-up

TEST filter-fsync-up

COPY tests/data/maps/fsync-down

TEST filter-fsync-down

TEST filter-meta-4560-rotate0

COPY tests/data/filtergraphs/overlay

TEST filter-overlay

COPY tests/data/filtergraphs/overlay\_rgb

TEST filter-overlay\_rgb

COPY tests/data/filtergraphs/overlay\_yuv420

TEST filter-overlay\_yuv420

COPY tests/data/filtergraphs/overlay\_yuv420p10

TEST filter-overlay\_yuv420p10

COPY tests/data/filtergraphs/overlay\_nv12

TEST filter-overlay\_nv12

COPY tests/data/filtergraphs/overlay\_nv21

TEST filter-overlay\_nv21

COPY tests/data/filtergraphs/overlay\_yuv422

TEST filter-overlay\_yuv422

COPY tests/data/filtergraphs/overlay\_yuv422p10

TEST filter-overlay\_yuv422p10

COPY tests/data/filtergraphs/overlay\_yuv444

TEST filter-overlay\_yuv444

COPY tests/data/filtergraphs/overlay\_yuv444p10

TEST filter-overlay\_yuv444p10

TEST filter-removegrain-mode-00

TEST filter-removegrain-mode-01

TEST filter-removegrain-mode-02

TEST filter-removegrain-mode-03

TEST filter-removegrain-mode-04

TEST filter-removegrain-mode-05

TEST filter-removegrain-mode-06

TEST filter-removegrain-mode-07

TEST filter-removegrain-mode-08

TEST filter-removegrain-mode-09

TEST filter-removegrain-mode-10

TEST filter-removegrain-mode-11

TEST filter-removegrain-mode-12

TEST filter-removegrain-mode-13

TEST filter-removegrain-mode-14

TEST filter-removegrain-mode-15

TEST filter-removegrain-mode-16

TEST filter-removegrain-mode-17

TEST filter-removegrain-mode-18

TEST filter-removegrain-mode-19

TEST filter-removegrain-mode-20

TEST filter-removegrain-mode-21

TEST filter-removegrain-mode-22

TEST filter-removegrain-mode-23

TEST filter-removegrain-mode-24

TEST filter-trim-duration

TEST filter-trim-frame

TEST filter-trim-mixed

TEST filter-trim-time

TEST filter-unsharp-yuv420p10

COPY tests/data/filtergraphs/scale2ref\_keep\_aspect

TEST filter-scale2ref\_keep\_aspect

TEST filter-scalechroma

TEST filter-idet

TEST vsynth1-mpeg4-qprd

TEST filter-pp

TEST filter-pp7

TEST filter-spp

TEST filter-codecview

TEST filter-select

TEST filter-stereo3d-al-sbsl

TEST filter-stereo3d-ar-abl

TEST filter-stereo3d-abr-mr

TEST filter-stereo3d-abr-ml

TEST filter-stereo3d-sbsl-abl

TEST filter-stereo3d-sbsl-abr

TEST filter-stereo3d-sbsl-al

TEST filter-stereo3d-sbsl-sbsr

TEST filter-stereo3d-sbsl-agmc

TEST filter-stereo3d-sbsl-agmd

TEST filter-stereo3d-sbsl-agmg

TEST filter-stereo3d-sbsl-agmh

TEST filter-stereo3d-sbsl-arbg

TEST filter-stereo3d-sbsl-arcc

TEST filter-stereo3d-sbsl-arcd

TEST filter-stereo3d-sbsl-arcg

TEST filter-stereo3d-sbsl-arch

TEST filter-stereo3d-sbsl-argg

TEST filter-stereo3d-sbsl-aybc

TEST filter-stereo3d-sbsl-aybd

TEST filter-stereo3d-sbsl-aybg

TEST filter-stereo3d-sbsl-aybh

TEST filter-pixdesc-yuv420p

TEST filter-pixdesc-yuyv422

TEST filter-pixdesc-rgb24

TEST filter-pixdesc-bgr24

TEST filter-pixdesc-yuv422p

TEST filter-pixdesc-yuv444p

TEST filter-pixdesc-yuv410p

TEST filter-pixdesc-yuv411p

TEST filter-pixdesc-gray

TEST filter-pixdesc-monow

TEST filter-pixdesc-monob

TEST filter-pixdesc-yuvj420p

TEST filter-pixdesc-yuvj422p

TEST filter-pixdesc-yuvj444p

TEST filter-pixdesc-uyvy422

TEST filter-pixdesc-bgr8

TEST filter-pixdesc-bgr4\_byte

TEST filter-pixdesc-rgb8

TEST filter-pixdesc-rgb4\_byte

TEST filter-pixdesc-nv12

TEST filter-pixdesc-nv21

TEST filter-pixdesc-argb

TEST filter-pixdesc-rgba

TEST filter-pixdesc-abgr

TEST filter-pixdesc-bgra

TEST filter-pixdesc-gray16be

TEST filter-pixdesc-gray16le

TEST filter-pixdesc-yuv440p

TEST filter-pixdesc-yuvj440p

TEST filter-pixdesc-yuva420p

TEST filter-pixdesc-rgb48be

TEST filter-pixdesc-rgb48le

TEST filter-pixdesc-rgb565be

TEST filter-pixdesc-rgb565le

TEST filter-pixdesc-rgb555be

TEST filter-pixdesc-rgb555le

TEST filter-pixdesc-bgr565be

TEST filter-pixdesc-bgr565le

TEST filter-pixdesc-bgr555be

TEST filter-pixdesc-bgr555le

TEST filter-pixdesc-yuv420p16le

TEST filter-pixdesc-yuv420p16be

TEST filter-pixdesc-yuv422p16le

TEST filter-pixdesc-yuv422p16be

TEST filter-pixdesc-yuv444p16le

TEST filter-pixdesc-yuv444p16be

TEST filter-pixdesc-rgb444le

TEST filter-pixdesc-rgb444be

TEST filter-pixdesc-bgr444le

TEST filter-pixdesc-bgr444be

TEST filter-pixdesc-ya8

TEST filter-pixdesc-bgr48be

TEST filter-pixdesc-bgr48le

TEST filter-pixdesc-yuv420p9be

TEST filter-pixdesc-yuv420p9le

TEST filter-pixdesc-yuv420p10be

TEST filter-pixdesc-yuv420p10le

TEST filter-pixdesc-yuv422p10be

TEST filter-pixdesc-yuv422p10le

TEST filter-pixdesc-yuv444p9be

TEST filter-pixdesc-yuv444p9le

TEST filter-pixdesc-yuv444p10be

TEST filter-pixdesc-yuv444p10le

TEST filter-pixdesc-yuv422p9be

TEST filter-pixdesc-yuv422p9le

TEST filter-pixdesc-gbrp

TEST filter-pixdesc-gbrp9be

TEST filter-pixdesc-gbrp9le

TEST filter-pixdesc-gbrp10be

TEST filter-pixdesc-gbrp10le

TEST filter-pixdesc-gbrp16be

TEST filter-pixdesc-gbrp16le

TEST filter-pixdesc-yuva422p

TEST filter-pixdesc-yuva444p

TEST filter-pixdesc-yuva420p9be

TEST filter-pixdesc-yuva420p9le

TEST filter-pixdesc-yuva422p9be

TEST filter-pixdesc-yuva422p9le

TEST filter-pixdesc-yuva444p9be

TEST filter-pixdesc-yuva444p9le

TEST filter-pixdesc-yuva420p10be

TEST filter-pixdesc-yuva420p10le

TEST filter-pixdesc-yuva422p10be

TEST filter-pixdesc-yuva422p10le

TEST filter-pixdesc-yuva444p10be

TEST filter-pixdesc-yuva444p10le

TEST filter-pixdesc-yuva420p16be

TEST filter-pixdesc-yuva420p16le

TEST filter-pixdesc-yuva422p16be

TEST filter-pixdesc-yuva422p16le

TEST filter-pixdesc-yuva444p16be

TEST filter-pixdesc-yuva444p16le

TEST filter-pixdesc-xyz12le

TEST filter-pixdesc-xyz12be

TEST filter-pixdesc-nv16

TEST filter-pixdesc-rgba64be

TEST filter-pixdesc-rgba64le

TEST filter-pixdesc-bgra64be

TEST filter-pixdesc-bgra64le

TEST filter-pixdesc-yvyu422

TEST filter-pixdesc-ya16be

TEST filter-pixdesc-ya16le

TEST filter-pixdesc-gbrap

TEST filter-pixdesc-gbrap16be

TEST filter-pixdesc-gbrap16le

TEST filter-pixdesc-0rgb

TEST filter-pixdesc-rgb0

TEST filter-pixdesc-0bgr

TEST filter-pixdesc-bgr0

TEST filter-pixdesc-yuv420p12be

TEST filter-pixdesc-yuv420p12le

TEST filter-pixdesc-yuv420p14be

TEST filter-pixdesc-yuv420p14le

TEST filter-pixdesc-yuv422p12be

TEST filter-pixdesc-yuv422p12le

TEST filter-pixdesc-yuv422p14be

TEST filter-pixdesc-yuv422p14le

TEST filter-pixdesc-yuv444p12be

TEST filter-pixdesc-yuv444p12le

TEST filter-pixdesc-yuv444p14be

TEST filter-pixdesc-yuv444p14le

TEST filter-pixdesc-gbrp12be

TEST filter-pixdesc-gbrp12le

TEST filter-pixdesc-gbrp14be

TEST filter-pixdesc-gbrp14le

TEST filter-pixdesc-yuvj411p

TEST filter-pixdesc-yuv440p10le

TEST filter-pixdesc-yuv440p10be

TEST filter-pixdesc-yuv440p12le

TEST filter-pixdesc-yuv440p12be

TEST filter-pixdesc-ayuv64le

TEST filter-pixdesc-p010le

TEST filter-pixdesc-p010be

TEST filter-pixdesc-gbrap12be

TEST filter-pixdesc-gbrap12le

TEST filter-pixdesc-gbrap10be

TEST filter-pixdesc-gbrap10le

TEST filter-pixdesc-gray12be

TEST filter-pixdesc-gray12le

TEST filter-pixdesc-gray10be

TEST filter-pixdesc-gray10le

TEST filter-pixdesc-p016le

TEST filter-pixdesc-p016be

TEST filter-pixdesc-gray9be

TEST filter-pixdesc-gray9le

TEST filter-pixdesc-gbrpf32be

TEST filter-pixdesc-gbrpf32le

TEST filter-pixdesc-gbrapf32be

TEST filter-pixdesc-gbrapf32le

TEST filter-pixdesc-gray14be

TEST filter-pixdesc-gray14le

TEST filter-pixdesc-grayf32be

TEST filter-pixdesc-grayf32le

TEST filter-pixdesc-yuva422p12be

TEST filter-pixdesc-yuva422p12le

TEST filter-pixdesc-yuva444p12be

TEST filter-pixdesc-yuva444p12le

TEST filter-pixdesc-nv24

TEST filter-pixdesc-nv42

TEST filter-pixdesc-y210le

TEST filter-pixdesc-x2rgb10le

TEST filter-pixdesc-x2bgr10le

TEST filter-pixdesc-p210be

TEST filter-pixdesc-p210le

TEST filter-pixdesc-p410be

TEST filter-pixdesc-p410le

TEST filter-pixdesc-p216be

TEST filter-pixdesc-p216le

TEST filter-pixdesc-p416be

TEST filter-pixdesc-p416le

TEST filter-pixdesc-vuya

TEST filter-pixdesc-vuyx

TEST filter-pixdesc-p012le

TEST filter-pixdesc-p012be

TEST filter-pixdesc-y212le

TEST filter-pixdesc-xv30le

TEST filter-pixdesc-xv36le

TEST filter-pixdesc-p212be

TEST filter-pixdesc-p212le

TEST filter-pixdesc-p412be

TEST filter-pixdesc-p412le

TEST filter-pixdesc-gbrap14be

TEST filter-pixdesc-gbrap14le

TEST filter-pixfmts-copy

TEST filter-pixfmts-crop

TEST filter-pixfmts-field

TEST filter-pixfmts-fieldmatch

TEST filter-pixfmts-fieldorder

TEST filter-pixfmts-hflip

TEST filter-pixfmts-il

TEST filter-pixfmts-kerndeint

TEST filter-pixfmts-lut

TEST filter-pixfmts-null

TEST filter-pixfmts-pad

TEST filter-pixfmts-pullup

TEST filter-pixfmts-rotate

TEST filter-pixfmts-scale

TEST filter-pixfmts-super2xsai

TEST filter-pixfmts-swapuv

TEST filter-pixfmts-tinterlace\_cvlpf

TEST filter-pixfmts-tinterlace\_merge

TEST filter-pixfmts-tinterlace\_pad

TEST filter-pixfmts-tinterlace\_vlpf

TEST filter-pixfmts-transpose

TEST filter-pixfmts-vflip

TEST filter-crop

TEST filter-crop\_scale

TEST filter-crop\_scale\_vflip

TEST filter-crop\_vflip

TEST filter-null

TEST filter-scale200

TEST filter-scale500

TEST filter-vflip

TEST filter-colormatrix1

TEST filter-colormatrix2

TEST filter-vflip\_crop

TEST filter-vflip\_vflip

TEST filter-edgedetect

TEST filter-edgedetect-colormix

TEST filter-median

TEST filter-hue1

TEST filter-hue2

TEST filter-hue3

TEST filter-hue4

TEST filter-pad

TEST filter-pp1

TEST filter-pp4

TEST filter-pp5

TEST filter-pp6

TEST filter-pp2

TEST filter-pp3

TEST filter-qp

TEST filter-setdar

TEST filter-setsar

TEST filter-thumbnail

TEST filter-tile

TEST filter-pixelize-avg

TEST filter-pixelize-min

TEST filter-pixelize-max

TEST filter-tiltandshift

TEST filter-boxblur

TEST filter-colorchannelmixer

TEST filter-drawbox

TEST filter-fade

TEST filter-fieldorder

TEST filter-framepack-columns

TEST filter-framepack-frameseq

TEST filter-framepack-lines

TEST filter-framepack-sbs

TEST filter-framepack-tab

TEST filter-gradfun

TEST filter-hqdn3d

TEST filter-interlace

TEST filter-interlace-complex

TEST filter-negate

TEST filter-histogram-levels

TEST filter-waveform\_column

TEST filter-waveform\_row

TEST filter-waveform\_envelope

TEST filter-waveform\_uv

TEST filter-vectorscope\_gray

TEST filter-vectorscope\_color

TEST filter-vectorscope\_color2

TEST filter-vectorscope\_color3

TEST filter-vectorscope\_color4

TEST filter-vectorscope\_xy

COPY tests/data/filtergraphs/mergeplanes

TEST filter-mergeplanes

COPY tests/data/filtergraphs/hstack

TEST filter-hstack

COPY tests/data/filtergraphs/vstack

TEST filter-vstack

TEST filter-phase

TEST filter-separatefields

TEST filter-weave

COPY tests/data/filtergraphs/select-alternate

TEST filter-select-alternate

COPY tests/data/filtergraphs/setpts

TEST filter-setpts

TEST filter-shuffleframes

TEST filter-shuffleplanes-dup-luma

TEST filter-shuffleplanes-swapuv

TEST filter-swaprect

TEST filter-tblend

TEST filter-telecine

TEST filter-transpose

TEST filter-unsharp

COPY tests/data/filtergraphs/alphamerge\_alphaextract\_rgb

TEST filter-alphaextract\_alphamerge\_rgb

COPY tests/data/filtergraphs/alphamerge\_alphaextract\_yuv

TEST filter-alphaextract\_alphamerge\_yuv

TEST filter-colorlevels

TEST filter-colorlevels-16

TEST filter-colorbalance

TEST filter-colorbalance-gbrap

TEST filter-colorbalance-rgba64

TEST filter-colorbalance-gbrap-16

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 104: fate-video

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

```sh

make fate-video SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST 4xm-1

TEST 4xm-2

TEST cfhd-1

TEST cfhd-2

TEST cfhd-3

TEST cvid-palette

TEST cvid-partial

TEST cvid-grayscale

TEST dxa-feeble

TEST dxa-scummvm

TEST flic-af11-palette-change

TEST flic-af12

TEST flic-magiccarpet

TEST txd-pal8

TEST txd-16bpp

TEST txd-odd

TEST 012v

TEST aasc

TEST aic

TEST aic-oddsize

TEST alg-mm

TEST amv

TEST ansi

TEST ansi256

TEST armovie-escape124

TEST armovie-escape130

TEST auravision-v1

TEST auravision-v2

TEST avid-interlaced

TEST avid-meridian

TEST bethsoft-vid

TEST bfi

TEST bink-video-b

TEST bink-video-f

TEST bink-video-i

TEST bmv-video

TEST cavs

TEST cdgraphics

TEST cljr

TEST corepng

TEST rgbapng-4816

TEST creatureshock-avs

TEST cyberia-c93

TEST cyuv

TEST delphine-cin-video

TEST deluxepaint-anm

TEST dirac

TEST dirac-low-delay

TEST dxv-dxt1

TEST dxv-dxt5

TEST dxv3-dxt1

TEST dxv3-dxt5

TEST dxv3-ycg6

TEST dxv3-yg10

TEST dxv3enc-dxt1

TEST film-cvid

TEST frwu

TEST id-cin-video

TEST idroq-video-encode

TEST iff-byterun1

TEST iff-fibonacci

TEST iff-ilbm

TEST interplay-mve-8bit

TEST interplay-mve-16bit

TEST jpeg2000-dcinema

TEST jv

TEST kgv1

TEST kmvc

TEST lscr

TEST magicyuv-y4444i

TEST magicyuv-y400i

TEST magicyuv-y420

TEST magicyuv-y422i

TEST magicyuv-y444

TEST magicyuv-rgba

TEST magicyuv-rgb

TEST mdec

TEST mdec-v3

TEST mimic

TEST mjpegb

TEST mjpeg-ticket3229

TEST motionpixels

TEST mpeg2-field-enc

TEST mpeg2-ticket186

TEST mpeg2-ticket6677

TEST mv-mvc1

TEST mv-mvc2

TEST mv-sgirle

TEST mxpeg

TEST nuv-rtjpeg

TEST nuv-rtjpeg-fh

TEST paf-video

TEST qpeg

TEST r210

TEST rl2

TEST roqvideo

TEST sanm

TEST sierra-vmd-video

TEST smacker-video

TEST smc

TEST sp5x

TEST thp

TEST tiertex-seq

TEST tmv

TEST ULTI

TEST v210

TEST v410dec

HOSTCC tests/videogen.o

HOSTLD tests/videogen

GEN tests/vsynth1/00.pgm

TEST v410enc

TEST vb

TEST vcr1

TEST vcr2

TEST videoxl

TEST vqa-cc

TEST wc3movie-xan

TEST wnv1

TEST yop

TEST xxan-wc4

TEST smvjpeg

TEST

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 105: fate-voice

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

```sh

make fate-voice SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST g722dec-1

GEN tests/data/asynth-16000-1.wav

TEST g722-encode

TEST g723\_1-dec-1

TEST g723\_1-dec-2

TEST g723\_1-dec-3

TEST g723\_1-dec-4

TEST g723\_1-dec-5

TEST g723\_1-dec-6

TEST g723\_1-dec-7

TEST g723\_1-dec-8

GEN tests/data/asynth-8000-1.wav

TEST g726-encode-2bit

TEST g726-encode-3bit

TEST g726-encode-4bit

TEST g726-encode-5bit

TEST gsm-ms

TEST gsm-toast

TEST qcelp

TEST truespeech

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 106: fate-vorbis

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

```sh

make fate-vorbis SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST vorbis-encode

TEST vorbis-1

TEST vorbis-2

TEST vorbis-3

TEST vorbis-4

TEST vorbis-5

TEST vorbis-6

TEST vorbis-7

TEST vorbis-8

TEST vorbis-9

TEST vorbis-10

TEST vorbis-11

TEST vorbis-12

TEST vorbis-13

TEST vorbis-14

TEST vorbis-15

TEST vorbis-16

TEST vorbis-17

TEST vorbis-18

TEST vorbis-19

TEST vorbis-20

TEST vorbis-1833-chapters

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 107: fate-vp3

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

```sh

make fate-vp3 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST theora-coeff-level64

TEST theora-offset

TEST vp31

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 108: fate-vp4

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

```sh

make fate-vp4 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST vp4

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 109: fate-vp5

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

```sh

make fate-vp5 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST vp5

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 110: fate-vp6

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

```sh

make fate-vp6 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST vp60

TEST vp61

TEST vp6a

TEST vp6a-skip\_alpha

TEST vp6f

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 111: fate-vp7

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

```sh

make fate-vp7 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST vp7

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 112: fate-vp8

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

```sh

make fate-vp8 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST vp8-2451

TEST vp8-test-vector-001

TEST vp8-test-vector-002

TEST vp8-test-vector-003

TEST vp8-test-vector-004

TEST vp8-test-vector-005

TEST vp8-test-vector-006

TEST vp8-test-vector-007

TEST vp8-test-vector-008

TEST vp8-test-vector-009

TEST vp8-test-vector-010

TEST vp8-test-vector-011

TEST vp8-test-vector-012

TEST vp8-test-vector-013

TEST vp8-test-vector-014

TEST vp8-test-vector-015

TEST vp8-test-vector-016

TEST vp8-test-vector-017

TEST vp8-sign-bias

TEST vp8-size-change

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 113: fate-vp9

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

```sh

make fate-vp9 SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST vp9-00-quantizer-00

TEST vp9-00-quantizer-01

TEST vp9-00-quantizer-02

TEST vp9-00-quantizer-03

TEST vp9-00-quantizer-04

TEST vp9-00-quantizer-05

TEST vp9-00-quantizer-06

TEST vp9-00-quantizer-07

TEST vp9-00-quantizer-08

TEST vp9-00-quantizer-09

TEST vp9-00-quantizer-10

TEST vp9-00-quantizer-11

TEST vp9-00-quantizer-12

TEST vp9-00-quantizer-13

TEST vp9-00-quantizer-14

TEST vp9-00-quantizer-15

TEST vp9-00-quantizer-16

TEST vp9-00-quantizer-17

TEST vp9-00-quantizer-18

TEST vp9-00-quantizer-19

TEST vp9-00-quantizer-20

TEST vp9-00-quantizer-21

TEST vp9-00-quantizer-22

TEST vp9-00-quantizer-23

TEST vp9-00-quantizer-24

TEST vp9-00-quantizer-25

TEST vp9-00-quantizer-26

TEST vp9-00-quantizer-27

TEST vp9-00-quantizer-28

TEST vp9-00-quantizer-29

TEST vp9-00-quantizer-30

TEST vp9-00-quantizer-31

TEST vp9-00-quantizer-32

TEST vp9-00-quantizer-33

TEST vp9-00-quantizer-34

TEST vp9-00-quantizer-35

TEST vp9-00-quantizer-36

TEST vp9-00-quantizer-37

TEST vp9-00-quantizer-38

TEST vp9-00-quantizer-39

TEST vp9-00-quantizer-40

TEST vp9-00-quantizer-41

TEST vp9-00-quantizer-42

TEST vp9-00-quantizer-43

TEST vp9-00-quantizer-44

TEST vp9-00-quantizer-45

TEST vp9-00-quantizer-46

TEST vp9-00-quantizer-47

TEST vp9-00-quantizer-48

TEST vp9-00-quantizer-49

TEST vp9-00-quantizer-50

TEST vp9-00-quantizer-51

TEST vp9-00-quantizer-52

TEST vp9-00-quantizer-53

TEST vp9-00-quantizer-54

TEST vp9-00-quantizer-55

TEST vp9-00-quantizer-56

TEST vp9-00-quantizer-57

TEST vp9-00-quantizer-58

TEST vp9-00-quantizer-59

TEST vp9-00-quantizer-60

TEST vp9-00-quantizer-61

TEST vp9-00-quantizer-62

TEST vp9-00-quantizer-63

TEST vp9-01-sharpness-1

TEST vp9-01-sharpness-2

TEST vp9-01-sharpness-3

TEST vp9-01-sharpness-4

TEST vp9-01-sharpness-5

TEST vp9-01-sharpness-6

TEST vp9-01-sharpness-7

TEST vp9-02-size-08x08

TEST vp9-02-size-08x10

TEST vp9-02-size-08x16

TEST vp9-02-size-08x18

TEST vp9-02-size-08x32

TEST vp9-02-size-08x34

TEST vp9-02-size-08x64

TEST vp9-02-size-08x66

TEST vp9-02-size-10x08

TEST vp9-02-size-10x10

TEST vp9-02-size-10x16

TEST vp9-02-size-10x18

TEST vp9-02-size-10x32

TEST vp9-02-size-10x34

TEST vp9-02-size-10x64

TEST vp9-02-size-10x66

TEST vp9-02-size-16x08

TEST vp9-02-size-16x10

TEST vp9-02-size-16x16

TEST vp9-02-size-16x18

TEST vp9-02-size-16x32

TEST vp9-02-size-16x34

TEST vp9-02-size-16x64

TEST vp9-02-size-16x66

TEST vp9-02-size-18x08

TEST vp9-02-size-18x10

TEST vp9-02-size-18x16

TEST vp9-02-size-18x18

TEST vp9-02-size-18x32

TEST vp9-02-size-18x34

TEST vp9-02-size-18x64

TEST vp9-02-size-18x66

TEST vp9-02-size-32x08

TEST vp9-02-size-32x10

TEST vp9-02-size-32x16

TEST vp9-02-size-32x18

TEST vp9-02-size-32x32

TEST vp9-02-size-32x34

TEST vp9-02-size-32x64

TEST vp9-02-size-32x66

TEST vp9-02-size-34x08

TEST vp9-02-size-34x10

TEST vp9-02-size-34x16

TEST vp9-02-size-34x18

TEST vp9-02-size-34x32

TEST vp9-02-size-34x34

TEST vp9-02-size-34x64

TEST vp9-02-size-34x66

TEST vp9-02-size-64x08

TEST vp9-02-size-64x10

TEST vp9-02-size-64x16

TEST vp9-02-size-64x18

TEST vp9-02-size-64x32

TEST vp9-02-size-64x34

TEST vp9-02-size-64x64

TEST vp9-02-size-64x66

TEST vp9-02-size-66x08

TEST vp9-02-size-66x10

TEST vp9-02-size-66x16

TEST vp9-02-size-66x18

TEST vp9-02-size-66x32

TEST vp9-02-size-66x34

TEST vp9-02-size-66x64

TEST vp9-02-size-66x66

TEST vp9-03-size-196x196

TEST vp9-03-size-196x198

TEST vp9-03-size-196x200

TEST vp9-03-size-196x202

TEST vp9-03-size-196x208

TEST vp9-03-size-196x210

TEST vp9-03-size-196x224

TEST vp9-03-size-196x226

TEST vp9-03-size-198x196

TEST vp9-03-size-198x198

TEST vp9-03-size-198x200

TEST vp9-03-size-198x202

TEST vp9-03-size-198x208

TEST vp9-03-size-198x210

TEST vp9-03-size-198x224

TEST vp9-03-size-198x226

TEST vp9-03-size-200x196

TEST vp9-03-size-200x198

TEST vp9-03-size-200x200

TEST vp9-03-size-200x202

TEST vp9-03-size-200x208

TEST vp9-03-size-200x210

TEST vp9-03-size-200x224

TEST vp9-03-size-200x226

TEST vp9-03-size-202x196

TEST vp9-03-size-202x198

TEST vp9-03-size-202x200

TEST vp9-03-size-202x202

TEST vp9-03-size-202x208

TEST vp9-03-size-202x210

TEST vp9-03-size-202x224

TEST vp9-03-size-202x226

TEST vp9-03-size-208x196

TEST vp9-03-size-208x198

TEST vp9-03-size-208x200

TEST vp9-03-size-208x202

TEST vp9-03-size-208x208

TEST vp9-03-size-208x210

TEST vp9-03-size-208x224

TEST vp9-03-size-208x226

TEST vp9-03-size-210x196

TEST vp9-03-size-210x198

TEST vp9-03-size-210x200

TEST vp9-03-size-210x202

TEST vp9-03-size-210x208

TEST vp9-03-size-210x210

TEST vp9-03-size-210x224

TEST vp9-03-size-210x226

TEST vp9-03-size-224x196

TEST vp9-03-size-224x198

TEST vp9-03-size-224x200

TEST vp9-03-size-224x202

TEST vp9-03-size-224x208

TEST vp9-03-size-224x210

TEST vp9-03-size-224x224

TEST vp9-03-size-224x226

TEST vp9-03-size-226x196

TEST vp9-03-size-226x198

TEST vp9-03-size-226x200

TEST vp9-03-size-226x202

TEST vp9-03-size-226x208

TEST vp9-03-size-226x210

TEST vp9-03-size-226x224

TEST vp9-03-size-226x226

TEST vp9-03-deltaq

TEST vp9p1-04-yuv422

TEST vp9p1-04-yuv440

TEST vp9p1-04-yuv444

TEST vp9p2-20-10bit-yuv420

TEST vp9p2-20-12bit-yuv420

TEST vp9p3-20-10bit-yuv422

TEST vp9p3-20-10bit-yuv440

TEST vp9p3-20-10bit-yuv444

TEST vp9p3-20-12bit-yuv422

TEST vp9p3-20-12bit-yuv440

TEST vp9p3-20-12bit-yuv444

TEST vp9-06-bilinear

TEST vp9-09-lf\_deltas

TEST vp9-10-show-existing-frame

TEST vp9-10-show-existing-frame2

TEST vp9-15-segkey\_adpq

TEST vp9-16-intra-only

TEST vp9-2pass-akiyo

TEST vp9-parallelmode-akiyo

TEST vp9-segmentation-aq-akiyo

TEST vp9-segmentation-sf-akiyo

TEST vp9-tiling-pedestrian

TEST vp9-trac3849

TEST vp9-trac4359

TEST vp9-05-resize

TEST vp9-superframe-bsf

TEST vp9-encparams

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 114: fate-vqf

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

```sh

make fate-vqf SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST twinvq

TEST vqf-demux

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 115: fate-vvc

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

```sh

make fate-vvc SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST vvc-conformance-CodingToolsSets\_A\_2

TEST vvc-conformance-APSALF\_A\_2

TEST vvc-conformance-APSLLMCS\_D\_1

TEST vvc-conformance-APSMULT\_A\_4

TEST vvc-conformance-AUD\_A\_3

TEST vvc-conformance-BUMP\_A\_2

TEST vvc-conformance-DCI\_A\_3

TEST vvc-conformance-HRD\_A\_3

TEST vvc-conformance-PHSH\_B\_1

TEST vvc-conformance-POC\_A\_1

TEST vvc-conformance-PPS\_B\_1

TEST vvc-conformance-RAP\_A\_1

TEST vvc-conformance-SAO\_A\_3

TEST vvc-conformance-SCALING\_A\_1

TEST vvc-conformance-SLICES\_A\_3

TEST vvc-conformance-SPS\_B\_1

TEST vvc-conformance-STILL\_B\_1

TEST vvc-conformance-SUBPIC\_A\_3

TEST vvc-conformance-TILE\_A\_2

TEST vvc-conformance-WP\_A\_3

TEST vvc-conformance-WPP\_A\_3

TEST vvc-conformance-WRAP\_A\_4

TEST vvc-conformance-CROP\_B\_4

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 116: fate-wavpack

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

```sh

make fate-wavpack SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST wavpack-matroskamode

TEST wavpack-matroska\_mux-mono

TEST wavpack-matroska\_mux-61

TEST wavpack-lossless-8bit

TEST wavpack-lossy-8bit

TEST wavpack-lossless-12bit

TEST wavpack-lossless-16bit

TEST wavpack-lossy-16bit

TEST wavpack-channels-monoint

TEST wavpack-channels-4.0

TEST wavpack-channels-5.1

TEST wavpack-channels-6.1

TEST wavpack-channels-7.1

TEST wavpack-speed-default

TEST wavpack-speed-fast

TEST wavpack-speed-high

TEST wavpack-speed-vhigh

TEST wavpack-clipping

TEST wavpack-cuesheet

TEST wavpack-falsestereo

TEST wavpack-zerolsbs

TEST wavpack-lossless-24bit

TEST wavpack-lossy-24bit

TEST wavpack-lossless-32bit

TEST wavpack-lossy-32bit

TEST wavpack-lossless-float

TEST wavpack-lossy-float

TEST wavpack-channels-monofloat

TEST wavpack-cuesheet-tags

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 117: fate-webm-dash-manifests

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

```sh

make fate-webm-dash-manifests SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST webm-dash-manifest

TEST webm-dash-manifest-unaligned-video-streams

TEST webm-dash-manifest-unaligned-audio-streams

TEST webm-dash-manifest-representations

TEST webm-dash-manifest-live

TEST webm-dash-manifest-live-bandwidth

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 118: fate-wma-encode

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

```sh

make fate-wma-encode SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST wmav1-encode

TEST wmav2-encode

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 119: fate-wmapro

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

```sh

make fate-wmapro SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST wmapro-2ch

TEST wmapro-5.1

TEST wmapro-ism

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 120: fate-wmavoice

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

```sh

make fate-wmavoice SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST wmavoice-7k

TEST wmavoice-11k

TEST wmavoice-19k

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

### Test case 121: fate-xvid

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

```sh

make fate-xvid SAMPLES=fate\_samples/

```

The running results are:

```sh

TEST xvid-custom-matrix

TEST xvid-idct

```

Note that the output shown here is the first run, because some build processes will not be executed again in the second run.

The ARM platform is the same as the RISCV platform.

## perf automated script test

Currently, the perf automation script supports one-click testing of all samples that can pass. Execute the test script:

```bash

./ffmpeg.sh

```

### Test result file description

After executing the `./ffmpeg.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.