



# Quadra™ Performance Test Report V5.5

## Contents

Contents .....	2
Environment Overview .....	4
Definitions .....	6
1. T1A – FFmpeg Throughput.....	7
2. T1A – Libxcodec Throughput.....	13
3. T1A – FFmpeg Latency .....	19
4. T1A – Decoder PPU Scaling .....	21
5. T1A – Streaming Ladder Generation.....	22
6. T1A – RGBA Encoding.....	23
7. T1A – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth .....	25
8. T1A – Capped CRF .....	31
9. T1A – Inplace Overlay .....	37
10. 2x T2A – MultiThread P2P DMA on AMD GPU .....	39
11. T1A – AI .....	40
12. T1A – GStreamer XStack Throughput .....	44
13. T1A – GStreamer Ladder Generation.....	46
14. T1U – FFmpeg Throughput .....	47
15. T1U – Libxcodec Throughput .....	53
16. T1U – FFmpeg Latency.....	59
17. T1U – Decoder PPU Scaling.....	61
18. T1U – Streaming Ladder Generation .....	62
19. T1U – RGBA Encoding .....	63
20. T1U – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth .....	65
21. T1U – Capped CRF.....	71
22. T1U – Inplace Overlay .....	77
23. A Note on T1M Persistent Configurations .....	79
24. T1M (Persistent config “F”) – FFmpeg Throughput .....	80
25. T1M (Persistent config “F”) – Libxcodec Throughput .....	86
26. T1M (Persistent config “F”) – FFmpeg Latency.....	92
27. T1M (Persistent config “F”) – Decoder PPU Scaling.....	94
28. T1M (Persistent config “F”) – Streaming Ladder Generation .....	95

29. T1M (Persistent config “F”) – RGBA Encoding .....	96
30. T1M (Persistent config “F”) – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth .....	98
31. T1M (Persistent config “F”) – Capped CRF.....	104
32. T1M (Persistent config “F”) – Inplace Overlay .....	110
33. T1M (Persistent config “E”) – FFmpeg Throughput .....	112
34. T1M (Persistent config “E”) – Libxcoder Throughput .....	114
35. T1M (Persistent config “E”) – FFmpeg Latency.....	116
36. T1M (Persistent config “E”) – RGBA Encoding .....	118
37. T1M (Persistent config “E”) – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth.....	120
38. T1M (Persistent config “E”) – Capped CRF .....	127
Appendix A: GStreamer XStack Command .....	134
Appendix B: 7x7 Grid Layout.....	136
Appendix C: GStreamer Ladder Command .....	137

## Environment Overview

Revision: 5506sqr2

### Setup #1:

- Server: AMD Ryzen 5 5600 6-core Processor; CPU(s) 12; Motherboard MPG X570 GAMING EDGE WIFI (MS-7C37); Memory 16GiB System Memory 2x 8GiB DIMM DDR4 Synchronous Unbuffered (Unregistered) 3200 MHz (0.3 ns)
- DUT: 1x T1A or 1x T1U
- FFmpeg Version: 7.1
- Gstreamer Version: 1.22.2
- Tests:
  - FFmpeg Throughput
  - Libxcodec Throughput
  - FFmpeg Latency
  - Decoder PPU Scaling
  - Streaming Ladder Generation
  - RGBA Encoding
  - Encoding EnableRdoQuant/rdoLevel/lookaheadDepth
  - Capped CRF
  - Inplace Overlay
  - Gstreamer XStack Throughput (T1A only)
  - Gstreamer Ladder Generation (T1A only)

### Setup #2:

- Server: Intel Core i7-7700 4-core Processor; CPU(s) 8; Motherboard ASUSTeK COMPUTER INC. H110I-PLUS (Rev X.0x); Memory 16GiB System Memory, 2x8GiB DIMM DDR4 Synchronous Unbuffered (Unregistered) 2133 MHz (0.5 ns)
- DUT: 1x T1M
- FFmpeg Version: 7.1
- Tests:
  - FFmpeg Throughput
  - Libxcodec Throughput
  - FFmpeg Latency
  - Decoder PPU Scaling
  - Streaming Ladder Generation
  - RGBA Encoding
  - Encoding EnableRdoQuant/rdoLevel/lookaheadDepth
  - Capped CRF
  - Inplace Overlay

#### Setup #3:

- Server: AMD EPYC 7763 64-Core Processor; CPU(s) 128; Motherboard TYAN S8030GM2NE-NFX; Memory 256GiB System Memory, 8x32GiB DIMM DDR4 Synchronous Registered (Buffered) 3200 MHz (0.3 ns)
- DUT: 2x T2A
- FFmpeg Version: 4.3.1
- Tests:
  - MultiThread P2P DMA on AMD GPU

#### Setup #4:

- Server: AMD Ryzen 5 5600X 6-Core Processor; CPU(s) 12; Motherboard TUF GAMING X570-PLUS (WI-FI); Memory 16GiB System Memory, 2x8GiB DIMM DDR4 Synchronous Unbuffered (Unregistered) 2133 MHz (0.5 ns)
- DUT: 1x T1A
- FFmpeg Version: 4.3.1
- Tests:
  - AI

## Definitions

- CPU: Average per instance CPU usage.
  - $(\text{System-wide CPU usage} * \text{number of CPU}) / (\text{number of devices} * \text{number of instances per device})$
- FPS: Total frames processed per second across all processes ( $\Sigma$  of average FPS per process)
- Jobs: Number of instances running concurrently
- HW Frame: Decoded YUV is kept on the device
- Bit: Input video's bit depth
- Resolution: Input video's resolution
- Load: Maximum load between FW Load and VPU Load during traffic

# 1. T1A – FFmpeg Throughput

## 1.1 Decoding

### 1.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe.

Bitstream is decoded by hardware decoder.

Decoded YUV frame is read out through PCIe and written into an output file.

### 1.1.2 Command Line

```
ffmpeg -nostdin -f concat -safe 0 -c:v <dec>_ni_quadra_dec -dec 0 -  
xcoder-params multicoreJointMode=<*> -i /media/ramdisk/input.list -f  
null /dev/null -
```

<dec> is the decoder codec. eg h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

## 1.2 Encoding

### 1.2.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

### 1.2.2 Command Line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f null  
/dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 8k, bitrate = 50000000, framerate = 24

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

## 1.3 Transcoding

### 1.3.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder.

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

### 1.3.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -c:v <dec>_ni_quadra_dec -dec 0 -  
xcoder-params out=hw:sempianar0=1:multicoreJointMode=<*> -i  
/media/ramdisk/input.list -c:v <enc>_ni_quadra_enc -enc 0 -xcoder-  
params intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f  
null /dev/null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 8k, bitrate = 50000000, framerate = 24

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)



## 1.4 FFmpeg Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	95	0	114	10
HEVC to YUV	8k	1	0	8	1	96	0	115	12
VP9 to YUV	8k	1	0	8	1	24	0	38	6
YUV to AVC	8k	1	0	8	1	0	96	67	52
YUV to HEVC	8k	1	0	8	1	0	98	83	67
AVC to AVC	8k	1	1	8	1	74	99	54	4
AVC to HEVC	8k	1	1	8	1	80	98	71	2
HEVC to AVC	8k	1	1	8	1	64	100	53	6
HEVC to HEVC	8k	1	1	8	1	68	100	71	6
VP9 to AVC	8k	1	1	8	1	24	44	35	2
VP9 to HEVC	8k	1	1	8	1	24	43	37	2
AVC to YUV	8k	1	0	10	1	56	0	59	8
HEVC to YUV	8k	1	0	10	1	98	0	61	6
VP9 to YUV	8k	1	0	10	1	24	0	33	15
YUV to AVC	8k	1	0	10	1	0	89	48	74
YUV to HEVC	8k	1	0	10	1	0	74	59	92
AVC to YUV	4k	1	0	8	1	56	0	318	19
HEVC to YUV	4k	1	0	8	1	51	0	339	20
VP9 to YUV	4k	1	0	8	1	24	0	155	4
AVC to YUV	4k	16	0	8	0	100	0	484	4
HEVC to YUV	4k	16	0	8	0	98	0	506	3
VP9 to YUV	4k	16	0	8	0	99	0	487	1
YUV to AVC	4k	1	0	8	1	0	95	294	39
YUV to HEVC	4k	1	0	8	1	0	95	327	49
YUV to AV1	4k	1	0	8	1	0	94	282	38
YUV to AVC	4k	4	0	8	0	0	93	288	26
YUV to HEVC	4k	4	0	8	0	0	96	332	14
YUV to AV1	4k	4	0	8	0	0	96	288	13
YUV to AVC	4k	8	0	8	0	0	100	320	8
YUV to HEVC	4k	8	0	8	0	0	99	344	9
YUV to AV1	4k	8	0	8	0	0	100	296	9
AVC to AVC	4k	1	1	8	1	66	93	222	15
AVC to HEVC	4k	1	1	8	1	69	92	272	15
AVC to AV1	4k	1	1	8	1	62	94	263	15
HEVC to AVC	4k	1	1	8	1	56	93	222	15
HEVC to HEVC	4k	1	1	8	1	60	92	275	15
HEVC to AV1	4k	1	1	8	1	52	92	257	9
VP9 to AVC	4k	1	1	8	1	24	45	144	4
VP9 to HEVC	4k	1	1	8	1	24	43	151	7

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	51	151	6
AVC to AVC	4k	4	1	8	0	62	96	240	7
AVC to HEVC	4k	4	1	8	0	68	96	300	3
AVC to AV1	4k	4	1	8	0	56	95	276	8
HEVC to AVC	4k	4	1	8	0	56	96	236	3
HEVC to HEVC	4k	4	1	8	0	60	96	296	3
HEVC to AV1	4k	4	1	8	0	58	93	252	8
VP9 to AVC	4k	4	1	8	0	63	97	243	4
VP9 to HEVC	4k	4	1	8	0	66	97	301	3
VP9 to AV1	4k	4	1	8	0	56	95	275	3
AVC to AVC	4k	8	1	8	0	68	99	211	2
AVC to HEVC	4k	8	1	8	0	76	99	280	3
AVC to AV1	4k	8	1	8	0	68	100	272	3
HEVC to AVC	4k	8	1	8	0	63	99	214	2
HEVC to HEVC	4k	8	1	8	0	68	99	280	2
HEVC to AV1	4k	8	1	8	0	59	100	278	3
VP9 to AVC	4k	8	1	8	0	68	99	235	3
VP9 to HEVC	4k	8	1	8	0	73	100	296	2
VP9 to AV1	4k	8	1	8	0	60	100	280	2
AVC to YUV	4k	1	0	10	1	44	0	226	10
HEVC to YUV	4k	1	0	10	1	51	0	233	9
VP9 to YUV	4k	1	0	10	1	24	0	126	4
AVC to YUV	4k	16	0	10	0	100	0	282	0
HEVC to YUV	4k	16	0	10	0	99	0	282	0
VP9 to YUV	4k	16	0	10	0	99	0	263	0
YUV to AVC	4k	1	0	10	1	0	69	200	58
YUV to HEVC	4k	1	0	10	1	0	60	207	62
YUV to AV1	4k	1	0	10	1	0	67	199	56
YUV to AVC	4k	4	0	10	0	0	93	219	26
YUV to HEVC	4k	4	0	10	0	0	77	249	41
YUV to AV1	4k	4	0	10	0	0	81	241	35
AVC to YUV	1080p	1	0	8	1	41	0	861	25
HEVC to YUV	1080p	1	0	8	1	43	0	841	27
VP9 to YUV	1080p	1	0	8	1	22	0	557	6
AVC to YUV	1080p	40	0	8	0	86	0	1720	0
HEVC to YUV	1080p	40	0	8	0	96	0	1839	1
VP9 to YUV	1080p	40	0	8	0	82	0	1778	0
YUV to AVC	1080p	1	0	8	1	0	54	705	28
YUV to HEVC	1080p	1	0	8	1	0	52	714	24
YUV to AV1	1080p	1	0	8	1	0	55	647	24

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1280	3
YUV to HEVC	1080p	32	0	8	0	0	99	1353	3
YUV to AV1	1080p	32	0	8	0	0	99	1175	3
AVC to AVC	1080p	1	1	8	1	66	80	967	22
AVC to HEVC	1080p	1	1	8	1	66	79	1017	26
AVC to AV1	1080p	1	1	8	1	53	79	898	20
HEVC to AVC	1080p	1	1	8	1	54	76	898	31
HEVC to HEVC	1080p	1	1	8	1	55	73	978	32
HEVC to AV1	1080p	1	1	8	1	47	76	852	28
VP9 to AVC	1080p	1	1	8	1	22	42	544	9
VP9 to HEVC	1080p	1	1	8	1	22	39	539	6
VP9 to AV1	1080p	1	1	8	1	22	46	541	8
AVC to AVC	1080p	32	1	8	0	76	100	938	2
AVC to HEVC	1080p	32	1	8	0	83	100	1065	1
AVC to AV1	1080p	32	1	8	0	77	99	1028	1
HEVC to AVC	1080p	32	1	8	0	68	100	992	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1121	1
HEVC to AV1	1080p	32	1	8	0	69	99	1057	1
VP9 to AVC	1080p	32	1	8	0	71	99	1088	1
VP9 to HEVC	1080p	32	1	8	0	74	99	1216	1
VP9 to AV1	1080p	32	1	8	0	71	99	1120	1
AVC to YUV	1080p	1	0	10	1	30	0	663	12
HEVC to YUV	1080p	1	0	10	1	27	0	697	8
VP9 to YUV	1080p	1	0	10	1	22	0	457	6
AVC to YUV	1080p	40	0	10	0	64	0	1086	0
HEVC to YUV	1080p	40	0	10	0	68	0	1080	0
VP9 to YUV	1080p	40	0	10	0	72	0	1063	0
YUV to AVC	1080p	1	0	10	1	0	39	505	40
YUV to HEVC	1080p	1	0	10	1	0	36	501	44
YUV to AV1	1080p	1	0	10	1	0	39	467	36
YUV to AVC	1080p	32	0	10	0	0	62	800	6
YUV to HEVC	1080p	32	0	10	0	0	58	800	6
YUV to AV1	1080p	32	0	10	0	0	64	769	5
AVC to YUV	720p	1	0	8	1	44	0	1208	19
HEVC to YUV	720p	1	0	8	1	38	0	1216	22
VP9 to YUV	720p	1	0	8	1	31	0	1022	8
AVC to YUV	720p	100	0	8	0	100	0	2534	0
HEVC to YUV	720p	100	0	8	0	100	0	2918	1
VP9 to YUV	720p	100	0	8	0	100	0	2633	0
YUV to AVC	720p	1	0	8	1	0	30	885	16

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	30	900	18
YUV to AV1	720p	1	0	8	1	0	34	805	14
YUV to AVC	720p	64	0	8	0	0	97	2368	1
YUV to HEVC	720p	64	0	8	0	0	94	2393	1
YUV to AV1	720p	64	0	8	0	0	98	2048	1
AVC to AVC	720p	1	1	8	1	48	48	1283	19
AVC to HEVC	720p	1	1	8	1	48	48	1280	19
AVC to AV1	720p	1	1	8	1	41	51	1094	21
HEVC to AVC	720p	1	1	8	1	41	46	1232	24
HEVC to HEVC	720p	1	1	8	1	41	46	1250	24
HEVC to AV1	720p	1	1	8	1	35	48	1079	22
VP9 to AVC	720p	1	1	8	1	32	35	1006	14
VP9 to HEVC	720p	1	1	8	1	32	34	993	12
VP9 to AV1	720p	1	1	8	1	31	44	994	12
AVC to AVC	720p	64	1	8	0	97	100	2051	0
AVC to HEVC	720p	64	1	8	0	98	100	2123	0
AVC to AV1	720p	64	1	8	0	77	100	1795	0
HEVC to AVC	720p	64	1	8	0	85	100	2058	0
HEVC to HEVC	720p	64	1	8	0	86	100	2146	0
HEVC to AV1	720p	64	1	8	0	69	100	1799	0
VP9 to AVC	720p	64	1	8	0	98	100	2287	0
VP9 to HEVC	720p	64	1	8	0	100	100	2368	0
VP9 to AV1	720p	64	1	8	0	79	100	1920	0

## 2. T1A – Libxcoder Throughput

### 2.1 Decoding

#### 2.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe.

Bitstream is decoded by hardware decoder.

Decoded YUV frame is read out through PCIe and written into an output file.

#### 2.1.2 Command Line

```
./ni_xcoder_decode -c 0 -r 1000 -i /media/ramdisk/input.<ext> -m  
<test_type> -o /dev/null -d multicoreJointMode=<*>
```

<test\_type> = test codecs. ie. a (avc), h (hevc), etc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

Note: Libxcoder decoding tests were run without multi-threading (but with multicoreJointMode enabled where noted)

### 2.2 Encoding

#### 2.2.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 2.2.2 Command Line

```
./ni_xcoder_encode -c 0 -s <resolution> -r 1000 -i  
/media/ramdisk/input.yuv -m <test_type> -o /dev/null -e  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*>
```

<test\_type> = test codecs. ie. a (avc), h (hevc), etc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 8k, bitrate = 50000000, framerate = 24

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

Note: Libxcoder encoding tests were run without multi-threading (but with multicoreJointMode enabled where noted)

## 2.3 Transcoding

### 2.3.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder.

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

### 2.3.2 Command line

```
./ni_xcoder_multithread_transcode -c 0 -r 1000 -i  
/media/ramdisk/input.<ext> -m <dec_test_type> -n <enc_test_type> -o  
/dev/null -e  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*> -d out=hw:semiplanar0=1:multicoreJointMode=1
```

<dec\_test\_type> = decoding test codecs. ie. a (avc), h (hevc), etc

<enc\_test\_type> = encoding test codecs. ie. a (avc), h (hevc), etc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 8k, bitrate = 50000000, framerate = 24

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

## 2.4 Libxcode Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	66	0	92	7
HEVC to YUV	8k	1	0	8	1	60	0	96	6
VP9 to YUV	8k	1	0	8	1	24	0	38	3
YUV to AVC	8k	1	0	8	1	0	98	68	28
YUV to HEVC	8k	1	0	8	1	0	95	82	33
AVC to AVC	8k	1	1	8	1	69	100	54	5
AVC to HEVC	8k	1	1	8	1	80	99	71	5
HEVC to AVC	8k	1	1	8	1	63	100	52	5
HEVC to HEVC	8k	1	1	8	1	77	99	70	4
VP9 to AVC	8k	1	1	8	1	24	46	35	4
VP9 to HEVC	8k	1	1	8	1	24	44	37	4
AVC to YUV	8k	1	0	10	1	55	0	54	7
HEVC to YUV	8k	1	0	10	1	52	0	51	6
VP9 to YUV	8k	1	0	10	1	24	0	33	4
YUV to AVC	8k	1	0	10	1	0	63	40	35
YUV to HEVC	8k	1	0	10	1	0	46	39	35
AVC to YUV	4k	1	0	8	1	57	0	320	8
HEVC to YUV	4k	1	0	8	1	50	0	330	6
VP9 to YUV	4k	1	0	8	1	24	0	154	3
AVC to YUV	4k	16	0	8	0	99	0	488	0
HEVC to YUV	4k	16	0	8	0	98	0	512	0
VP9 to YUV	4k	16	0	8	0	100	0	492	0
YUV to AVC	4k	1	0	8	1	0	91	284	26
YUV to HEVC	4k	1	0	8	1	0	86	297	25
YUV to AV1	4k	1	0	8	1	0	90	270	23
YUV to AVC	4k	4	0	8	0	0	96	311	11
YUV to HEVC	4k	4	0	8	0	0	96	334	11
YUV to AV1	4k	4	0	8	0	0	96	289	10
YUV to AVC	4k	8	0	8	0	0	100	326	7
YUV to HEVC	4k	8	0	8	0	0	100	347	9
YUV to AV1	4k	8	0	8	0	0	100	301	7
AVC to AVC	4k	1	1	8	0	69	97	226	6
AVC to HEVC	4k	1	1	8	0	73	96	278	8
AVC to AV1	4k	1	1	8	0	65	97	263	8
HEVC to AVC	4k	1	1	8	0	58	97	225	5
HEVC to HEVC	4k	1	1	8	0	64	97	286	6
HEVC to AV1	4k	1	1	8	0	55	97	268	6
VP9 to AVC	4k	1	1	8	0	24	46	145	4

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	0	24	43	149	4
VP9 to AV1	4k	1	1	8	0	24	51	150	4
AVC to AVC	4k	4	1	8	0	62	97	243	5
AVC to HEVC	4k	4	1	8	0	67	96	302	5
AVC to AV1	4k	4	1	8	0	56	95	276	5
HEVC to AVC	4k	4	1	8	0	56	96	239	5
HEVC to HEVC	4k	4	1	8	0	60	95	294	5
HEVC to AV1	4k	4	1	8	0	48	95	275	5
VP9 to AVC	4k	4	1	8	0	63	97	246	4
VP9 to HEVC	4k	4	1	8	0	67	96	303	4
VP9 to AV1	4k	4	1	8	0	55	96	278	4
AVC to AVC	4k	8	1	8	0	68	100	217	3
AVC to HEVC	4k	8	1	8	0	75	100	280	3
AVC to AV1	4k	8	1	8	0	68	100	274	3
HEVC to AVC	4k	8	1	8	0	65	100	219	3
HEVC to HEVC	4k	8	1	8	0	67	100	282	3
HEVC to AV1	4k	8	1	8	0	60	100	275	3
VP9 to AVC	4k	8	1	8	0	68	99	236	3
VP9 to HEVC	4k	8	1	8	0	71	100	303	3
VP9 to AV1	4k	8	1	8	0	62	99	287	3
AVC to YUV	4k	1	0	10	1	47	0	214	7
HEVC to YUV	4k	1	0	10	1	41	0	207	6
VP9 to YUV	4k	1	0	10	1	24	0	125	4
AVC to YUV	4k	16	0	10	0	98	0	286	0
HEVC to YUV	4k	16	0	10	0	99	0	285	0
VP9 to YUV	4k	16	0	10	0	100	0	267	0
YUV to AVC	4k	1	0	10	1	0	51	163	27
YUV to HEVC	4k	1	0	10	1	0	46	161	26
YUV to AV1	4k	1	0	10	1	0	51	155	26
YUV to AVC	4k	4	0	10	0	0	62	203	26
YUV to HEVC	4k	4	0	10	0	0	59	209	27
YUV to AV1	4k	4	0	10	0	0	63	184	28
AVC to YUV	1080p	1	0	8	1	40	0	834	13
HEVC to YUV	1080p	1	0	8	1	46	0	886	10
VP9 to YUV	1080p	1	0	8	1	22	0	557	4
AVC to YUV	1080p	40	0	8	0	83	0	1635	0
HEVC to YUV	1080p	40	0	8	0	86	0	1746	0
VP9 to YUV	1080p	40	0	8	0	70	0	1667	0
YUV to AVC	1080p	1	0	8	1	0	57	742	16



TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	55	750	18
YUV to AV1	1080p	1	0	8	1	0	50	595	14
YUV to AVC	1080p	32	0	8	0	0	99	1325	2
YUV to HEVC	1080p	32	0	8	0	0	100	1402	3
YUV to AV1	1080p	32	0	8	0	0	99	1211	2
AVC to AVC	1080p	1	1	8	0	70	85	1011	16
AVC to HEVC	1080p	1	1	8	0	68	82	1064	16
AVC to AV1	1080p	1	1	8	0	55	82	935	16
HEVC to AVC	1080p	1	1	8	0	61	86	992	12
HEVC to HEVC	1080p	1	1	8	0	65	86	1095	12
HEVC to AV1	1080p	1	1	8	0	52	83	935	13
VP9 to AVC	1080p	1	1	8	0	21	42	546	6
VP9 to HEVC	1080p	1	1	8	0	22	39	541	6
VP9 to AV1	1080p	1	1	8	0	22	46	540	7
AVC to AVC	1080p	32	1	8	0	76	100	978	1
AVC to HEVC	1080p	32	1	8	0	84	99	1107	1
AVC to AV1	1080p	32	1	8	0	78	99	1065	1
HEVC to AVC	1080p	32	1	8	0	70	99	1023	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1158	1
HEVC to AV1	1080p	32	1	8	0	68	99	1091	1
VP9 to AVC	1080p	32	1	8	0	70	100	1117	0
VP9 to HEVC	1080p	32	1	8	0	74	100	1263	0
VP9 to AV1	1080p	32	1	8	0	71	99	1158	0
AVC to YUV	1080p	1	0	10	1	29	0	667	7
HEVC to YUV	1080p	1	0	10	1	26	0	673	6
VP9 to YUV	1080p	1	0	10	1	22	0	456	4
AVC to YUV	1080p	40	0	10	0	62	0	1075	0
HEVC to YUV	1080p	40	0	10	0	68	0	1077	0
VP9 to YUV	1080p	40	0	10	0	66	0	1033	0
YUV to AVC	1080p	1	0	10	1	0	35	456	22
YUV to HEVC	1080p	1	0	10	1	0	35	484	18
YUV to AV1	1080p	1	0	10	1	0	36	425	20
YUV to AVC	1080p	32	0	10	0	0	63	852	6
YUV to HEVC	1080p	32	0	10	0	0	60	852	6
YUV to AV1	1080p	32	0	10	0	0	65	802	8
AVC to YUV	720p	1	0	8	1	42	0	1166	12
HEVC to YUV	720p	1	0	8	1	39	0	1241	8
VP9 to YUV	720p	1	0	8	1	31	0	1021	6
AVC to YUV	720p	100	0	8	0	100	0	2768	0

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	94	0	2960	0
VP9 to YUV	720p	100	0	8	0	92	0	2676	0
YUV to AVC	720p	1	0	8	1	0	35	1025	12
YUV to HEVC	720p	1	0	8	1	0	33	985	11
YUV to AV1	720p	1	0	8	1	0	32	774	10
YUV to AVC	720p	64	0	8	0	0	99	2692	2
YUV to HEVC	720p	64	0	8	0	0	98	2765	2
YUV to AV1	720p	64	0	8	0	0	100	2220	2
AVC to AVC	720p	1	1	8	0	52	50	1347	15
AVC to HEVC	720p	1	1	8	0	52	49	1365	16
AVC to AV1	720p	1	1	8	0	44	49	1081	15
HEVC to AVC	720p	1	1	8	0	46	50	1399	11
HEVC to HEVC	720p	1	1	8	0	46	50	1396	12
HEVC to AV1	720p	1	1	8	0	38	48	1078	11
VP9 to AVC	720p	1	1	8	0	32	35	1007	7
VP9 to HEVC	720p	1	1	8	0	31	34	995	7
VP9 to AV1	720p	1	1	8	0	31	44	995	8
AVC to AVC	720p	64	1	8	0	94	99	2186	0
AVC to HEVC	720p	64	1	8	0	100	100	2294	0
AVC to AV1	720p	64	1	8	0	92	100	1904	0
HEVC to AVC	720p	64	1	8	0	88	99	2201	0
HEVC to HEVC	720p	64	1	8	0	96	100	2300	0
HEVC to AV1	720p	64	1	8	0	89	100	1913	0
VP9 to AVC	720p	64	1	8	0	100	100	2453	0
VP9 to HEVC	720p	64	1	8	0	100	100	2512	0
VP9 to AV1	720p	64	1	8	0	98	100	2049	0

## 3. T1A – FFmpeg Latency

### 3.1 Encoding

#### 3.1.1 Description

Libxcodec is compiled and installed with parameter `--with-latency-display`

```
$ bash build.sh --with-latency-display
```

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

For each frame, the encoder latency (eLat) value is provided in the output log.

All eLat values are parsed from the output log and the last 50% of frame data before killing ffmpeg instances is used to calculate the Average, Min, Max, and Variance.

The first 50% of frame data are ignored to reach stability while launching multiple jobs.

#### 3.1.2 Command Line

```
ffmpeg -re -loglevel info -f rawvideo -pix_fmt yuv420p -stream_loop  
1000 -s:v <resolution> -i /media/ramdisk/input.yuv -c:v  
<enc>_ni_quadra_enc -enc 0 -xcodec-params gopPresetIdx=9:lowDelay=1 -f  
null -
```

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<resolution> is resolution of input

### 3.2 FFmpeg Latency Performance Results

TYPE	RESOLUTION	JOBS	ELAT_AVG (ms)	ELAT_MAX (ms)	ELAT_MIN (ms)	ELAT_VAR (ms)
YUV to AVC	8k	1	57.8	59.25	57.4	0.04
YUV to HEVC	8k	1	55.46	60.48	53.99	1.03
YUV to AVC	4k	1	15.4	17.76	15.03	0.11
YUV to HEVC	4k	1	16.59	18.79	15.65	0.1
YUV to AV1	4k	1	21.89	26.97	16.5	0.79
YUV to AVC	4k	4	16.76	20.91	14.97	2.21
YUV to HEVC	4k	4	17.75	21.85	15.66	1.56
YUV to AV1	4k	4	22.81	27.28	16.3	0.86
YUV to AVC	4k	8	19.89	32.26	15.03	16.76
YUV to HEVC	4k	8	22.93	34.58	15.67	21.61
YUV to AV1	4k	8	37.25	47.32	22.91	10.25
YUV to AVC	1080p	1	4.72	6.51	4.41	0.17
YUV to HEVC	1080p	1	5.01	5.76	4.74	0.03
YUV to AV1	1080p	1	6.69	7.43	4.84	0.06
YUV to AVC	1080p	32	5.75	10.16	4.58	0.37
YUV to HEVC	1080p	32	6.65	12.01	4.8	1.12
YUV to AV1	1080p	32	41.02	47.64	34.18	1.15
YUV to AVC	720p	1	2.81	3.49	2.47	0.01
YUV to HEVC	720p	1	2.85	3.26	2.73	0.01
YUV to AV1	720p	1	3.75	4.09	2.92	0.01
YUV to AVC	720p	64	5.05	9.94	2.9	1.07
YUV to HEVC	720p	64	5.29	10.3	2.99	1.08
YUV to AV1	720p	64	39.77	52.02	30.73	2.96

## 4. T1A – Decoder PPU Scaling

### 4.1 Decoding

#### 4.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe.

Bitstream is decoded by hardware decoder and scaled to 224x224 with decoder post processing unit.

Decoded YUV is kept on device.

The YUV frame is converted to RGBA format with 2D Engine.

The RGBA frame is read out through PCIe and written into an output file.

#### 4.1.2 Command Line

```
ffmpeg -vsync 0 -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params  
out=hw:scale0=224x224:multicoreJointMode=<resolution=8k?1:0> -f concat  
-safe 0 -i /media/ramdisk/input.list -vf  
ni_quadra_scale=iw:ih:format=rgba,hwdownload,format=rgba -f null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

### 4.2 Decoder PPU Scaling Performance Results

TYPE	RESOLUTION	JOBS	DEC_LOAD	SCALER_LOAD	FPS	CPU
AVC to RGBA	8k	1	92	0	143	0
HEVC to RGBA	8k	1	92	0	154	0
VP9 to RGBA	8k	1	22	0	40	0
AVC to RGBA	4k	1	21	0	145	0
AVC to RGBA	4k	16	93	3	587	0
HEVC to RGBA	4k	1	21	0	174	0
HEVC to RGBA	4k	16	93	4	670	0
VP9 to RGBA	4k	1	22	0	166	0
VP9 to RGBA	4k	16	95	4	682	0
AVC to RGBA	1080p	40	92	14	1900	0
HEVC to RGBA	1080p	40	92	14	2010	0
VP9 to RGBA	1080p	40	95	17	2441	0
AVC to RGBA	720p	100	98	18	2602	0
HEVC to RGBA	720p	100	93	20	2869	0
VP9 to RGBA	720p	64	94	18	2752	0

## 5. T1A – Streaming Ladder Generation

### 5.1 Transcoding

#### 5.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder split and scaled to smaller resolutions with decoder post processing unit or 2D Engine.

Decoded YUV frame is kept on device.

The YUV frames are encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

#### 5.1.2 Command line

```
ffmpeg -vsync 0 -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params  
out=hw:sempianar0=1:enableOut1=1:sempianar1=1:scale1=1280x720:enableO  
ut2=1:sempianar2=1:scale2=960x540 -f concat -safe 0 -i  
/media/ramdisk/input.list -filter_complex  
'[0:v]ni_quadra_split=2:1:2[1080p][1080p_1][720p][540p][540p_1];[540p_1  
]ni_quadra_scale=640x360[360p]'-map [1080p] -xcoder-params  
RcEnable=1:bitrate=3500000 -c:v <enc>_ni_quadra_enc -enc 0 -f null - -  
map [1080p_1] -xcoder-params RcEnable=1:bitrate=1800000 -c:v  
<enc>_ni_quadra_enc -enc 0 -f null - -map [720p] -xcoder-params  
RcEnable=1:bitrate=1000000 -c:v <enc>_ni_quadra_enc -enc 0 -f null - -  
map [540p] -xcoder-params RcEnable=1:bitrate=800000 -c:v  
<enc>_ni_quadra_enc -enc 0 -f null - -map [360p] -xcoder-params  
RcEnable=1:bitrate=500000 -c:v <enc>_ni_quadra_enc -enc 0 -f null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

Input: 1080p

Output: 1080p, 1080p, 720p(PPU Scale), 540p(PPU Scale), 360p(2D Scale)

### 5.2 Streaming Ladder Generation Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	8	32	92	2	435	4
AVC to HEVC	8	31	92	2	468	4
AVC to AV1	8	23	96	2	400	3
HEVC to AVC	8	34	94	2	440	5
HEVC to HEVC	8	33	94	2	472	5
HEVC to AV1	8	25	95	2	401	4
VP9 to AVC	8	40	94	2	432	4
VP9 to HEVC	8	38	93	2	466	4
VP9 to AV1	8	29	94	2	400	4

## 6. T1A – RGBA Encoding

### 6.1 Encoding

#### 6.1.1 Description

RGBA frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

RGBA frame is uploaded and encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 6.1.2 Command line

```
ffmpeg -nostdin -stream_loop -1 -f rawvideo -pix_fmt rgba -s:v  
<resolution> -r 30 -i /media/ramdisk/input.rgb -vf  
"ni_quadra_hwupload=0" -c:v <enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f null  
/dev/null
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30

<resolution> == 720p, bitrate = 1500000, framerate = 30

## 6.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	53	162	61
RGBA to HEVC	4k	1	1	46	161	60
RGBA to AV1	4k	1	1	55	165	72
RGBA to AVC	4k	4	0	67	165	35
RGBA to HEVC	4k	4	0	60	169	39
RGBA to AV1	4k	4	0	61	172	44
RGBA to AVC	4k	8	0	61	176	28
RGBA to HEVC	4k	8	0	56	177	28
RGBA to AV1	4k	8	0	62	178	35
RGBA to AVC	1080p	1	1	34	445	44
RGBA to HEVC	1080p	1	1	32	442	46
RGBA to AV1	1080p	1	1	36	425	43
RGBA to AVC	1080p	16	0	58	651	15
RGBA to HEVC	1080p	16	0	53	655	14
RGBA to AV1	1080p	16	0	56	640	14
RGBA to AVC	1080p	32	0	56	669	8
RGBA to HEVC	1080p	32	0	52	675	8
RGBA to AV1	1080p	32	0	58	653	9
RGBA to AVC	720p	1	1	29	709	63
RGBA to HEVC	720p	1	1	29	785	67
RGBA to AV1	720p	1	1	34	702	60
RGBA to AVC	720p	16	0	48	1235	36
RGBA to HEVC	720p	16	0	46	1221	35
RGBA to AV1	720p	16	0	54	1160	33
RGBA to AVC	720p	32	0	45	1172	31
RGBA to HEVC	720p	32	0	44	1170	31
RGBA to AV1	720p	32	0	55	1178	28



## 7. T1A – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

### 7.1 Encoding

#### 7.1.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder with a mix of xcoder-params EnableRdoQuant, rdoLevel, and lookaheadDepth.

Encoded bitstream is read out through PCIe and written into an output file.

#### 7.1.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:lookaheadDepth=<*>:EnableRdoQuant=  
<*>:rdoLevel=<*> -f null /dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30

## 7.2 Encoding EnableRdoQuant/rdoLevel/lookaheadDepth Performance Results

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	0	0	1	96	304	15
YUV to HEVC	4k	4	0	0	1	96	332	13
YUV to AV1	4k	4	0	0	1	96	288	13
YUV to AVC	4k	4	0	0	2	95	298	23
YUV to HEVC	4k	4	0	0	2	99	171	7
YUV to AV1	4k	4	0	0	2	98	140	6
YUV to AVC	4k	4	0	0	3	96	298	13
YUV to HEVC	4k	4	0	0	3	98	100	6
YUV to AV1	4k	4	0	0	3	99	76	3
YUV to AVC	4k	4	0	1	1	98	188	7
YUV to HEVC	4k	4	0	1	1	98	240	10
YUV to AVC	4k	4	0	1	2	98	188	8
YUV to HEVC	4k	4	0	1	2	97	108	6
YUV to AVC	4k	4	0	1	3	98	188	8
YUV to HEVC	4k	4	0	1	3	100	68	4
YUV to AVC	4k	4	4	0	1	99	196	9
YUV to HEVC	4k	4	4	0	1	99	228	10
YUV to AV1	4k	4	4	0	1	99	200	12
YUV to AVC	4k	4	4	0	2	100	197	8
YUV to HEVC	4k	4	4	0	2	99	136	7
YUV to AV1	4k	4	4	0	2	100	116	5
YUV to AVC	4k	4	4	0	3	100	197	9
YUV to HEVC	4k	4	4	0	3	99	88	5
YUV to AV1	4k	4	4	0	3	99	72	3
YUV to AVC	4k	4	4	1	1	100	140	6
YUV to HEVC	4k	4	4	1	1	100	180	7
YUV to AVC	4k	4	4	1	2	100	140	6
YUV to HEVC	4k	4	4	1	2	100	96	5
YUV to AVC	4k	4	4	1	3	99	140	6
YUV to HEVC	4k	4	4	1	3	100	64	3
YUV to AVC	4k	4	16	0	1	100	196	9
YUV to HEVC	4k	4	16	0	1	99	228	10
YUV to AV1	4k	4	16	0	1	100	198	8
YUV to AVC	4k	4	16	0	2	100	196	9
YUV to HEVC	4k	4	16	0	2	99	136	6
YUV to AV1	4k	4	16	0	2	99	116	5
YUV to AVC	4k	4	16	0	3	99	196	9
YUV to HEVC	4k	4	16	0	3	99	88	4

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	4k	4	16	0	3	100	72	4
YUV to AVC	4k	4	16	1	1	100	140	6
YUV to HEVC	4k	4	16	1	1	99	180	8
YUV to AVC	4k	4	16	1	2	99	140	6
YUV to HEVC	4k	4	16	1	2	99	96	4
YUV to AVC	4k	4	16	1	3	100	140	6
YUV to HEVC	4k	4	16	1	3	99	64	3
YUV to AVC	4k	4	40	0	1	100	196	9
YUV to HEVC	4k	4	40	0	1	99	224	9
YUV to AV1	4k	4	40	0	1	100	196	8
YUV to AVC	4k	4	40	0	2	99	196	9
YUV to HEVC	4k	4	40	0	2	100	136	6
YUV to AV1	4k	4	40	0	2	99	112	5
YUV to AVC	4k	4	40	0	3	100	196	10
YUV to HEVC	4k	4	40	0	3	99	88	5
YUV to AV1	4k	4	40	0	3	98	72	4
YUV to AVC	4k	4	40	1	1	99	136	7
YUV to HEVC	4k	4	40	1	1	100	176	8
YUV to AVC	4k	4	40	1	2	99	136	6
YUV to HEVC	4k	4	40	1	2	99	92	4
YUV to AVC	4k	4	40	1	3	99	136	6
YUV to HEVC	4k	4	40	1	3	100	64	3
YUV to AVC	1080p	20	0	0	1	99	1280	4
YUV to HEVC	1080p	20	0	0	1	99	1360	5
YUV to AV1	1080p	20	0	0	1	99	1178	4
YUV to AVC	1080p	20	0	0	2	99	1280	4
YUV to HEVC	1080p	20	0	0	2	100	685	2
YUV to AV1	1080p	20	0	0	2	99	560	2
YUV to AVC	1080p	20	0	0	3	99	1280	4
YUV to HEVC	1080p	20	0	0	3	99	400	1
YUV to AV1	1080p	20	0	0	3	99	300	1
YUV to AVC	1080p	20	0	1	1	99	760	2
YUV to HEVC	1080p	20	0	1	1	100	960	3
YUV to AVC	1080p	20	0	1	2	100	760	3
YUV to HEVC	1080p	20	0	1	2	99	440	2
YUV to AVC	1080p	20	0	1	3	99	760	3
YUV to HEVC	1080p	20	0	1	3	99	280	1
YUV to AVC	1080p	20	4	0	1	99	700	2
YUV to HEVC	1080p	20	4	0	1	99	820	3

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	99	720	2
YUV to AVC	1080p	20	4	0	2	99	700	2
YUV to HEVC	1080p	20	4	0	2	99	515	2
YUV to AV1	1080p	20	4	0	2	99	421	1
YUV to AVC	1080p	20	4	0	3	99	700	3
YUV to HEVC	1080p	20	4	0	3	99	340	1
YUV to AV1	1080p	20	4	0	3	99	271	1
YUV to AVC	1080p	20	4	1	1	99	501	2
YUV to HEVC	1080p	20	4	1	1	100	660	2
YUV to AVC	1080p	20	4	1	2	99	500	2
YUV to HEVC	1080p	20	4	1	2	99	360	2
YUV to AVC	1080p	20	4	1	3	99	500	2
YUV to HEVC	1080p	20	4	1	3	100	240	1
YUV to AVC	1080p	20	16	0	1	99	700	2
YUV to HEVC	1080p	20	16	0	1	99	820	3
YUV to AV1	1080p	20	16	0	1	100	719	2
YUV to AVC	1080p	20	16	0	2	99	700	2
YUV to HEVC	1080p	20	16	0	2	99	502	2
YUV to AV1	1080p	20	16	0	2	99	420	1
YUV to AVC	1080p	20	16	0	3	100	701	2
YUV to HEVC	1080p	20	16	0	3	100	340	1
YUV to AV1	1080p	20	16	0	3	100	261	1
YUV to AVC	1080p	20	16	1	1	100	500	2
YUV to HEVC	1080p	20	16	1	1	100	643	2
YUV to AVC	1080p	20	16	1	2	99	500	2
YUV to HEVC	1080p	20	16	1	2	100	360	1
YUV to AVC	1080p	20	16	1	3	100	500	2
YUV to HEVC	1080p	20	16	1	3	99	240	1
YUV to AVC	1080p	20	40	0	1	99	687	2
YUV to HEVC	1080p	20	40	0	1	99	800	2
YUV to AV1	1080p	20	40	0	1	99	702	2
YUV to AVC	1080p	20	40	0	2	99	691	2
YUV to HEVC	1080p	20	40	0	2	99	500	2
YUV to AV1	1080p	20	40	0	2	100	420	2
YUV to AVC	1080p	20	40	0	3	99	690	2
YUV to HEVC	1080p	20	40	0	3	100	340	1
YUV to AV1	1080p	20	40	0	3	100	260	1
YUV to AVC	1080p	20	40	1	1	99	500	2
YUV to HEVC	1080p	20	40	1	1	99	640	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	2	99	500	2
YUV to HEVC	1080p	20	40	1	2	99	360	1
YUV to AVC	1080p	20	40	1	3	99	500	2
YUV to HEVC	1080p	20	40	1	3	100	240	1
YUV to AVC	720p	40	0	0	1	93	2347	3
YUV to HEVC	720p	40	0	0	1	90	2363	3
YUV to AV1	720p	40	0	0	1	95	2044	2
YUV to AVC	720p	40	0	0	2	93	2360	2
YUV to HEVC	720p	40	0	0	2	99	1520	1
YUV to AV1	720p	40	0	0	2	99	1201	1
YUV to AVC	720p	40	0	0	3	93	2342	3
YUV to HEVC	720p	40	0	0	3	100	920	1
YUV to AV1	720p	40	0	0	3	100	653	1
YUV to AVC	720p	40	0	1	1	99	1691	1
YUV to HEVC	720p	40	0	1	1	99	2080	2
YUV to AVC	720p	40	0	1	2	99	1692	1
YUV to HEVC	720p	40	0	1	2	100	960	1
YUV to AVC	720p	40	0	1	3	99	1688	1
YUV to HEVC	720p	40	0	1	3	100	609	1
YUV to AVC	720p	40	4	0	1	99	1440	1
YUV to HEVC	720p	40	4	0	1	100	1364	1
YUV to AV1	720p	40	4	0	1	100	1040	1
YUV to AVC	720p	40	4	0	2	100	1440	1
YUV to HEVC	720p	40	4	0	2	99	1080	1
YUV to AV1	720p	40	4	0	2	99	880	1
YUV to AVC	720p	40	4	0	3	99	1440	1
YUV to HEVC	720p	40	4	0	3	99	720	1
YUV to AV1	720p	40	4	0	3	99	560	1
YUV to AVC	720p	40	4	1	1	99	1079	1
YUV to HEVC	720p	40	4	1	1	99	1333	1
YUV to AVC	720p	40	4	1	2	99	1076	1
YUV to HEVC	720p	40	4	1	2	99	760	1
YUV to AVC	720p	40	4	1	3	99	1076	1
YUV to HEVC	720p	40	4	1	3	99	520	1
YUV to AVC	720p	40	16	0	1	100	1411	1
YUV to HEVC	720p	40	16	0	1	100	1371	1
YUV to AV1	720p	40	16	0	1	100	1080	1
YUV to AVC	720p	40	16	0	2	99	1404	1
YUV to HEVC	720p	40	16	0	2	99	1080	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	720p	40	16	0	2	99	880	1
YUV to AVC	720p	40	16	0	3	100	1409	1
YUV to HEVC	720p	40	16	0	3	99	720	1
YUV to AV1	720p	40	16	0	3	99	560	1
YUV to AVC	720p	40	16	1	1	99	1047	1
YUV to HEVC	720p	40	16	1	1	99	1320	1
YUV to AVC	720p	40	16	1	2	99	1050	1
YUV to HEVC	720p	40	16	1	2	99	760	1
YUV to AVC	720p	40	16	1	3	99	1048	1
YUV to HEVC	720p	40	16	1	3	99	520	1
YUV to AVC	720p	40	40	0	1	100	1393	1
YUV to HEVC	720p	40	40	0	1	100	1327	1
YUV to AV1	720p	40	40	0	1	100	1044	1
YUV to AVC	720p	40	40	0	2	100	1378	1
YUV to HEVC	720p	40	40	0	2	99	1040	1
YUV to AV1	720p	40	40	0	2	99	844	1
YUV to AVC	720p	40	40	0	3	100	1393	1
YUV to HEVC	720p	40	40	0	3	100	720	1
YUV to AV1	720p	40	40	0	3	99	560	1
YUV to AVC	720p	40	40	1	1	99	1040	1
YUV to HEVC	720p	40	40	1	1	99	1320	1
YUV to AVC	720p	40	40	1	2	99	1040	1
YUV to HEVC	720p	40	40	1	2	99	760	1
YUV to AVC	720p	40	40	1	3	99	1040	1
YUV to HEVC	720p	40	40	1	3	100	520	1

## 8. T1A – Capped CRF

### 8.1 Encoding with lookaheadDepth

#### 8.1.1 Description

YUV frame is read from an input file on ramdisk and fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder with a mix of xcoder-params EnableRdoQuant, rdoLevel, lookaheadDepth, CRF, bitrate, and vbvBufferSize.

Encoded bitstream is read out through PCIe and written into an output file.

#### 8.1.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:vbvBufferSize=1000:bitrate=<*>:lookaheadDepth=<*>:EnableR  
doQuant=<*>:rdoLevel=<*>:crf=<*> -f null /dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30

## 8.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	19	99	700	2
YUV to HEVC	1080p	20	0	0	1	19	99	820	2
YUV to AV1	1080p	20	0	0	1	19	99	720	2
YUV to AVC	1080p	20	0	0	2	19	99	701	2
YUV to HEVC	1080p	20	0	0	2	19	100	520	2
YUV to AV1	1080p	20	0	0	2	19	99	440	1
YUV to AVC	1080p	20	0	0	3	19	99	700	2
YUV to HEVC	1080p	20	0	0	3	19	100	340	1
YUV to AV1	1080p	20	0	0	3	19	100	280	1
YUV to AVC	1080p	20	0	1	1	19	99	502	2
YUV to HEVC	1080p	20	0	1	1	19	99	660	2
YUV to AVC	1080p	20	0	1	2	19	99	500	2
YUV to HEVC	1080p	20	0	1	2	19	99	360	2
YUV to AVC	1080p	20	0	1	3	19	99	501	2
YUV to HEVC	1080p	20	0	1	3	19	100	240	1
YUV to AVC	1080p	20	4	0	1	19	99	700	3
YUV to HEVC	1080p	20	4	0	1	19	99	820	3
YUV to AV1	1080p	20	4	0	1	19	99	720	2
YUV to AVC	1080p	20	4	0	2	19	99	700	2
YUV to HEVC	1080p	20	4	0	2	19	99	520	2
YUV to AV1	1080p	20	4	0	2	19	99	420	2
YUV to AVC	1080p	20	4	0	3	19	99	700	2
YUV to HEVC	1080p	20	4	0	3	19	100	340	1
YUV to AV1	1080p	20	4	0	3	19	99	269	1
YUV to AVC	1080p	20	4	1	1	19	99	500	2
YUV to HEVC	1080p	20	4	1	1	19	99	660	2
YUV to AVC	1080p	20	4	1	2	19	100	500	2
YUV to HEVC	1080p	20	4	1	2	19	99	360	1
YUV to AVC	1080p	20	4	1	3	19	100	500	2
YUV to HEVC	1080p	20	4	1	3	19	99	240	1
YUV to AVC	1080p	20	16	0	1	19	99	700	3
YUV to HEVC	1080p	20	16	0	1	19	99	820	3
YUV to AV1	1080p	20	16	0	1	19	99	719	2
YUV to AVC	1080p	20	16	0	2	19	99	700	2
YUV to HEVC	1080p	20	16	0	2	19	100	501	2
YUV to AV1	1080p	20	16	0	2	19	99	420	1
YUV to AVC	1080p	20	16	0	3	19	99	700	3
YUV to HEVC	1080p	20	16	0	3	19	100	340	1



TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	3	19	100	261	1
YUV to AVC	1080p	20	16	1	1	19	99	500	2
YUV to HEVC	1080p	20	16	1	1	19	99	641	2
YUV to AVC	1080p	20	16	1	2	19	99	500	2
YUV to HEVC	1080p	20	16	1	2	19	100	360	1
YUV to AVC	1080p	20	16	1	3	19	100	500	2
YUV to HEVC	1080p	20	16	1	3	19	99	240	1
YUV to AVC	1080p	20	40	0	1	19	99	693	3
YUV to HEVC	1080p	20	40	0	1	19	99	800	3
YUV to AV1	1080p	20	40	0	1	19	100	701	2
YUV to AVC	1080p	20	40	0	2	19	99	691	3
YUV to HEVC	1080p	20	40	0	2	19	99	500	2
YUV to AV1	1080p	20	40	0	2	19	100	420	1
YUV to AVC	1080p	20	40	0	3	19	99	692	2
YUV to HEVC	1080p	20	40	0	3	19	100	338	1
YUV to AV1	1080p	20	40	0	3	19	100	260	1
YUV to AVC	1080p	20	40	1	1	19	100	500	2
YUV to HEVC	1080p	20	40	1	1	19	99	640	2
YUV to AVC	1080p	20	40	1	2	19	100	500	2
YUV to HEVC	1080p	20	40	1	2	19	100	360	1
YUV to AVC	1080p	20	40	1	3	19	100	500	2
YUV to HEVC	1080p	20	40	1	3	19	100	240	1
YUV to AVC	1080p	20	0	0	1	23	100	700	2
YUV to HEVC	1080p	20	0	0	1	23	99	820	3
YUV to AV1	1080p	20	0	0	1	23	99	720	2
YUV to AVC	1080p	20	0	0	2	23	99	700	2
YUV to HEVC	1080p	20	0	0	2	23	99	520	2
YUV to AV1	1080p	20	0	0	2	23	99	440	1
YUV to AVC	1080p	20	0	0	3	23	100	700	2
YUV to HEVC	1080p	20	0	0	3	23	99	340	1
YUV to AV1	1080p	20	0	0	3	23	100	280	1
YUV to AVC	1080p	20	0	1	1	23	99	501	2
YUV to HEVC	1080p	20	0	1	1	23	100	660	2
YUV to AVC	1080p	20	0	1	2	23	100	503	2
YUV to HEVC	1080p	20	0	1	2	23	100	360	2
YUV to AVC	1080p	20	0	1	3	23	99	504	2
YUV to HEVC	1080p	20	0	1	3	23	100	240	1
YUV to AVC	1080p	20	4	0	1	23	99	700	2
YUV to HEVC	1080p	20	4	0	1	23	99	820	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	23	99	720	2
YUV to AVC	1080p	20	4	0	2	23	99	701	2
YUV to HEVC	1080p	20	4	0	2	23	99	520	2
YUV to AV1	1080p	20	4	0	2	23	99	420	2
YUV to AVC	1080p	20	4	0	3	23	100	700	2
YUV to HEVC	1080p	20	4	0	3	23	99	340	1
YUV to AV1	1080p	20	4	0	3	23	99	271	1
YUV to AVC	1080p	20	4	1	1	23	99	502	2
YUV to HEVC	1080p	20	4	1	1	23	99	660	2
YUV to AVC	1080p	20	4	1	2	23	99	501	2
YUV to HEVC	1080p	20	4	1	2	23	100	360	1
YUV to AVC	1080p	20	4	1	3	23	99	500	2
YUV to HEVC	1080p	20	4	1	3	23	99	240	1
YUV to AVC	1080p	20	16	0	1	23	99	700	2
YUV to HEVC	1080p	20	16	0	1	23	99	818	3
YUV to AV1	1080p	20	16	0	1	23	99	720	2
YUV to AVC	1080p	20	16	0	2	23	99	700	3
YUV to HEVC	1080p	20	16	0	2	23	100	501	2
YUV to AV1	1080p	20	16	0	2	23	100	420	1
YUV to AVC	1080p	20	16	0	3	23	99	700	2
YUV to HEVC	1080p	20	16	0	3	23	100	340	1
YUV to AV1	1080p	20	16	0	3	23	100	262	1
YUV to AVC	1080p	20	16	1	1	23	100	500	2
YUV to HEVC	1080p	20	16	1	1	23	99	643	2
YUV to AVC	1080p	20	16	1	2	23	99	500	2
YUV to HEVC	1080p	20	16	1	2	23	100	360	1
YUV to AVC	1080p	20	16	1	3	23	100	500	2
YUV to HEVC	1080p	20	16	1	3	23	100	240	1
YUV to AVC	1080p	20	40	0	1	23	100	691	2
YUV to HEVC	1080p	20	40	0	1	23	99	802	3
YUV to AV1	1080p	20	40	0	1	23	99	700	2
YUV to AVC	1080p	20	40	0	2	23	99	692	2
YUV to HEVC	1080p	20	40	0	2	23	100	500	2
YUV to AV1	1080p	20	40	0	2	23	100	420	1
YUV to AVC	1080p	20	40	0	3	23	100	692	2
YUV to HEVC	1080p	20	40	0	3	23	100	340	1
YUV to AV1	1080p	20	40	0	3	23	99	260	1
YUV to AVC	1080p	20	40	1	1	23	99	500	2
YUV to HEVC	1080p	20	40	1	1	23	100	640	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	2	23	99	500	2
YUV to HEVC	1080p	20	40	1	2	23	100	359	1
YUV to AVC	1080p	20	40	1	3	23	99	500	2
YUV to HEVC	1080p	20	40	1	3	23	100	240	1
YUV to AVC	1080p	20	0	0	1	27	100	701	2
YUV to HEVC	1080p	20	0	0	1	27	99	820	3
YUV to AV1	1080p	20	0	0	1	27	100	720	2
YUV to AVC	1080p	20	0	0	2	27	99	701	2
YUV to HEVC	1080p	20	0	0	2	27	99	520	2
YUV to AV1	1080p	20	0	0	2	27	100	440	1
YUV to AVC	1080p	20	0	0	3	27	99	700	3
YUV to HEVC	1080p	20	0	0	3	27	99	340	1
YUV to AV1	1080p	20	0	0	3	27	100	280	1
YUV to AVC	1080p	20	0	1	1	27	99	501	2
YUV to HEVC	1080p	20	0	1	1	27	99	660	2
YUV to AVC	1080p	20	0	1	2	27	99	503	2
YUV to HEVC	1080p	20	0	1	2	27	99	360	1
YUV to AVC	1080p	20	0	1	3	27	100	501	2
YUV to HEVC	1080p	20	0	1	3	27	99	240	1
YUV to AVC	1080p	20	4	0	1	27	100	700	2
YUV to HEVC	1080p	20	4	0	1	27	99	820	3
YUV to AV1	1080p	20	4	0	1	27	99	720	2
YUV to AVC	1080p	20	4	0	2	27	99	700	3
YUV to HEVC	1080p	20	4	0	2	27	100	520	2
YUV to AV1	1080p	20	4	0	2	27	100	420	2
YUV to AVC	1080p	20	4	0	3	27	99	700	2
YUV to HEVC	1080p	20	4	0	3	27	99	340	1
YUV to AV1	1080p	20	4	0	3	27	99	270	1
YUV to AVC	1080p	20	4	1	1	27	100	501	2
YUV to HEVC	1080p	20	4	1	1	27	99	660	2
YUV to AVC	1080p	20	4	1	2	27	100	500	2
YUV to HEVC	1080p	20	4	1	2	27	99	360	1
YUV to AVC	1080p	20	4	1	3	27	100	500	2
YUV to HEVC	1080p	20	4	1	3	27	99	240	1
YUV to AVC	1080p	20	16	0	1	27	99	700	2
YUV to HEVC	1080p	20	16	0	1	27	99	819	3
YUV to AV1	1080p	20	16	0	1	27	99	719	2
YUV to AVC	1080p	20	16	0	2	27	99	700	2
YUV to HEVC	1080p	20	16	0	2	27	99	503	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	2	27	100	420	1
YUV to AVC	1080p	20	16	0	3	27	99	700	3
YUV to HEVC	1080p	20	16	0	3	27	99	340	1
YUV to AV1	1080p	20	16	0	3	27	99	260	1
YUV to AVC	1080p	20	16	1	1	27	99	500	2
YUV to HEVC	1080p	20	16	1	1	27	100	646	2
YUV to AVC	1080p	20	16	1	2	27	99	500	2
YUV to HEVC	1080p	20	16	1	2	27	100	360	2
YUV to AVC	1080p	20	16	1	3	27	100	500	2
YUV to HEVC	1080p	20	16	1	3	27	99	240	1
YUV to AVC	1080p	20	40	0	1	27	99	686	2
YUV to HEVC	1080p	20	40	0	1	27	99	802	3
YUV to AV1	1080p	20	40	0	1	27	99	701	2
YUV to AVC	1080p	20	40	0	2	27	99	688	2
YUV to HEVC	1080p	20	40	0	2	27	100	500	2
YUV to AV1	1080p	20	40	0	2	27	99	420	1
YUV to AVC	1080p	20	40	0	3	27	99	691	2
YUV to HEVC	1080p	20	40	0	3	27	100	340	1
YUV to AV1	1080p	20	40	0	3	27	99	260	1
YUV to AVC	1080p	20	40	1	1	27	99	500	2
YUV to HEVC	1080p	20	40	1	1	27	99	640	2
YUV to AVC	1080p	20	40	1	2	27	99	500	2
YUV to HEVC	1080p	20	40	1	2	27	100	360	2
YUV to AVC	1080p	20	40	1	3	27	99	500	2
YUV to HEVC	1080p	20	40	1	3	27	100	240	1

## 9. T1A – Inplace Overlay

### 9.1 Transcoding

#### 9.1.1 Description

A bitstream is read from an input file on ramdisk and then fed into the hardware decoder through PCIe. The bitstream is decoded by the hardware decoder. The decoded YUV frame is kept on the device.

An RGBA image is also uploaded to the device and overlayed onto the video stream via the 2D Engine. The overlayed YUV frames are encoded with the hardware encoder. The encoded bitstream is then read out through PCIe and written into an output file.

#### 9.1.2 Command line

```
ffmpeg -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params "out=hw" -f
concat -safe 0 -i /media/ramdisk/input.list -f rawvideo -s:v 128x128 -
pix_fmt rgba -i /media/ramdisk/img.rgb -filter_complex
"[1:v]format=rgba,ni_quadra_hwupload=0[a];[0:v][a]ni_quadra_overlay=0:0
:alpha=1:inplace=1[b]" -c:a copy -map "[b]" -c:v <enc>_ni_quadra_enc -
enc 0 -xcoder-params "RcEnable=1:bitrate=2000000" -f null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

Input Video: 1080p

Input Image: 128x128

## 9.2 Inplace Overlay Performance Results

TYPE	JOB5	FPS	CPU	DEC_LOAD	ENC_LOAD	SCALER_LOAD
AVC to AVC	1	290	11	14	21	5
AVC to HEVC	1	308	12	14	20	5
AVC to AV1	1	263	12	13	20	4
HEVC to AVC	1	289	15	12	20	5
HEVC to HEVC	1	307	16	13	21	5
HEVC to AV1	1	262	15	12	20	4
VP9 to AVC	1	288	10	17	20	4
VP9 to HEVC	1	305	12	18	21	5
VP9 to AV1	1	263	11	15	20	4
AVC to AVC	16	1047	3	71	93	24
AVC to HEVC	16	1159	3	79	94	27
AVC to AV1	16	1109	3	71	92	25
HEVC to AVC	16	1088	4	71	94	24
HEVC to HEVC	16	1216	4	78	95	28
HEVC to AV1	16	1136	4	68	93	25
VP9 to AVC	16	1040	3	83	95	24
VP9 to HEVC	16	1169	3	89	92	26
VP9 to AV1	16	1120	3	84	94	25
AVC to AVC	32	960	1	75	94	24
AVC to HEVC	32	1082	1	80	92	26
AVC to AV1	32	1056	1	76	93	25
HEVC to AVC	32	997	1	73	94	24
HEVC to HEVC	32	1140	2	78	92	27
HEVC to AV1	32	1094	2	73	94	26
VP9 to AVC	32	992	1	80	92	23
VP9 to HEVC	32	1120	1	89	93	26
VP9 to AV1	32	1088	1	84	95	26

## 10. 2x T2A – MultiThread P2P DMA on AMD GPU

### 10.1 Encoding

#### 10.1.1 Description

GPU renders frames in its video memory and will convert it from RGB to YUV.

YUV is transferred directly to Quadra device through peer-to-peer DMA without host PC intervention.

The YUV frame is encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

#### 10.1.2 Command line

```
sudo python3 ~/FFmpegXcoder/amd-multi-  
thread/scripts/run_multiple_encoding.py --frames 1000 --instance  
<num_jobs> --codec 0
```

<num\_jobs> = number of instances running concurrently

### 10.2 Multi Thread P2P DMA on AMD GPU Performance Results

TYPE	RES	JOBS	Frames	FPS	CPU	Enc Load	P2P MEM	Latency Avg	Latency Dev
P2A	720p	1	1001	30	2.4	0.5	1	2.1	0.01
P2H	720p	1	1001	30	2.4	0.5	1	2.08	0.01
P2A	720p	180	180180	29.6	2.26	46	56	2.44	0.33
P2H	720p	180	180180	29.8	2.33	45	56	2.52	0.37
P2A	1080p	1	1001	30	2.8	1	1.5	3.82	0.01
P2H	1080p	1	1001	30	2.7	1	1.5	3.72	0.02
P2A	1080p	80	80080	29.2	2.75	46	50	4.15	0.48
P2H	1080p	80	80080	30	2.62	43	50	3.93	0.43

## 11. T1A – AI

### 11.1 AI Model

#### 11.1.1 Description

aiperf reads the network binary file provided on the command line and sends the data to the device through the PCIe bus. At the device side, the network binary is unfolded into memory and initializes the AI hardware.

aiperf sends and receives the model input and output parameters from the device through the PCIe bus.

aiperf performs any data format conversion expected by the hardware.

After conversion, aiperf writes the input data to the device through the PCIe bus. The device receives the input data buffer address, then trigger the hardware to start the inference.

When the device has completed the inference, aiperf then reads the output data from the device through the PCIe bus.

The output data is converted to tensor data or binary data, based on the hardware and model format.

#### 11.1.2 Command line

```
sudo ./aiperf -conf_file config_example.json
```

In config\_example.json, user need to specify the following arguments

```
{  
    "nb": "/path/to/network_binary_0.nb",  
    "dataset": "/path/to/dataset0.txt",  
    "outdir": "/path/to/output0",  
    "format": "nchw",  
    "order": "rgb",  
    "devid": "0",  
    "loop": "10000"  
}
```

In dataset0.txt, user need to specify the path to the input batch (image or tensor)  
/path/to/image.png



### 11.1.3 AI Model Performance Results

Model	Instance Number	Loops	Channel Order	File Format	File Type	Input Size	Total FPS
yolov5s_640	8	10000	rgb	nchw	image	640x640x3	78
yolov5s_320	8	10000	rgb	nchw	image	320x320x3	278
deeplabv3_FRP	8	10000	rgb	nchw	image	257x257x3	360
resnet50	8	10000	rgb	nchw	image	224x224x3	228
mobilenetv2	8	10000	rgb	nchw	image	224x224x3	1113
deeplabv3	8	10000	rgb	nchw	image	257x257x3	160
yolov4	8	10000	bgr	nchw	image	416x416x3	257
fsrcnn	8	2000	bgr	nchw	image	360x640x1	31
BiSeNetv1	8	10000	rgb	nchw	image	512x512x3	76
HrNet	8	10000	rgb	nchw	image	256x192x3	74
usm_1656x1920	8	10000	rgb	nchw	image	1920x1656x1	251
usm_3240x3840	8	10000	rgb	nchw	image	3840x3240x1	61
lpips	8	2000	rgb	nchw	image	720x480x3	1
PaddleOCR-512_onnx	8	10000	rgb	nchw	image	512x48x3	16
segm32_tflite_kl_mle	8	10000	rgb	nchw	image	256x144x3	837
mobilenetv2_nchw_keras_96x160	1	10000	rgb	nchw	image	96x160x3	2355.5
mobilenetv2_nchw_keras_96x160	8	10000	rgb	nchw	image	96x160x3	2349.41
mobilenetv2_nchw_keras_96x160	16	10000	rgb	nchw	image	96x160x3	2346.52

## 11.2 AI Encoding with 2D Engine

### 11.2.1 Description BG Filter

The FFmpeg Background Removal filter analyses input frames, inferences these input images using the AI module (segm32), segments the foreground and background of the input images, and then removes the background.

With the features of 2D scale, AI inference, alpha merge, and 2D overlay, the background removal filter can remove the background of the input frame.

### 11.2.2 Description ROI Filter

The FFmpeg ROI filter makes inferences from input frames using the AI module in Quadra. It identifies the bounding coordinates of chosen objects and classes within the images, and then wraps the coordinates into ROI side data.

All ROI side data within an image is appended to, then passed down to the encoder along with the actual images themselves.

### 11.2.3 Description PRE Filter

The FFMPEG PRE filter makes YUV previous processing by custom AI model. The input and output are both a single Quadra HW Frame. The actual effect is determined by the AI model.

### 11.2.4 Command line BG

```
ffmpeg -vsync 0 -init_hw_device ni_quadra=foo:0 -dec 0 -c:v  
h264_ni_quadra_dec -xcoder-params 'out=hw' -i bg_1920x1080.h264 -  
filter_hw_device foo -vf  
'ni_quadra_bg=nb=segm32_tflite_nchw_bgr.nb:use_default_bg=1' -enc 0 -  
c:v h264_ni_quadra_enc -xcoder-params "RcEnable=1" -f null -
```

### 11.2.5 Command line ROI

```
ffmpeg -vsync 0 -init_hw_device ni_quadra=foo:0 -dec 0 -c:v  
h264_ni_quadra_dec -xcoder-params 'out=hw' -i cr7_1920x1080.h264 -  
filter_hw_device foo -vf  
'ni_quadra_roi=nb=network_binary_yolov4_head.nb:qpoffset=-0.3' -enc 0 -  
c:v h264_ni_quadra_enc -xcoder-params 'roiEnable=1:RcEnable=1' -f null  
-
```

### 11.2.6 Command line PRE

```
ffmpeg -vsync 0 -dec 0 -c:v h264_ni_quadra_dec -xcoder-params 'out=hw'  
-f concat -safe 0 -i pre_1920x1080.h264.list -vf  
ni_quadra_ai_pre=nb=<hw_*_network_binary>:width=1280:height=720 -enc 0  
-c:v h265_ni_quadra_enc -xcoder-params RcEnable=1:bitrate=1000000 -f  
null -
```

### 11.2.7 AI Encoding with 2D Engine Performance Results

Filter	Model	Resolution	Session Number	Average FPS per session
ROI	network_binary_yolov4_head	1920x1080	1	77
ROI	network_binary_yolov4_head	1920x1080	8	23
ROI	network_binary_yolov4_head	1920x1080	32	5
BG	segm32_tflite_nchw_bgr	1920x1080	1	68
BG	segm32_tflite_nchw_bgr	1920x1080	8	40
BG	segm32_tflite_nchw_bgr	1920x1080	32	14
PRE	hw_lanczos_network_binary	1920x1080	8	66
PRE	hw_lanczos_network_binary	1920x1080	16	34
PRE	hw_bicubic_network_binary	1920x1080	8	67
PRE	hw_bicubic_network_binary	1920x1080	16	34

## 12. T1A – GStreamer XStack Throughput

### 12.1 Transcoding

#### 12.1.1 Description

Bitstreams are read from multiple input files on ramdisk and then fed into hardware decoder through PCIe. Bitstreams are decoded by hardware decoder.

Decoded YUV frames are all kept on device and are sent through the ni\_quadra\_xstack filter to produce a single YUV output.

The YUV frame is encoded with hardware encoder.

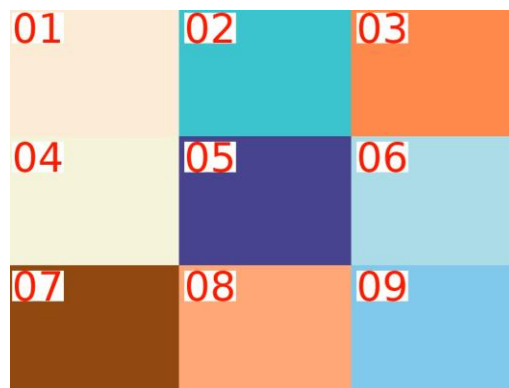
The encoded bitstream is read out through PCIe and written into an output file.

In this test, XStack will generate a single video output in a 3x3, 4x4, or 4x8 grid format generated from 9, 16, or 32 inputs, respectively.

Each input will scale to a cell size and be placed in the grid layout.

The grid layout and cell size will determine the output resolution.

This test is HEVC to AVC only.



*Example output in a 3x3 layout with 9 inputs*

#### 12.1.2 Command line

See Appendix A: GStreamer XStack Command

## 12.2 GStreamer XStack Performance Results

Input Res	Grid	Output Res	Cell Size	FPS	CPU	Dec Load	Enc Load	Scaler Load
1920x1080	3x3	1920x1080	640x360	145.1	36	63	13	14
1920x1080	4x4	1920x1080	480x270	99.56	53	80	10	14
1920x1080	4x8	1920x1080	480x135	55.01	73	87	5	8
1920x1080	3x3	3840x2160	1280x720	69.91	19	32	22	18
1920x1080	4x4	3840x2160	960x540	54.69	28	46	19	20
1920x1080	4x8	3840x2160	960x270	41.33	52	70	14	16
1920x1080	3x3	7680x4320	2560x1440	19.86	8	8	22	15
1920x1080	4x4	7680x4320	1920x1080	19.11	12	15	22	16
1920x1080	4x8	7680x4320	1920x540	19.23	23	33	22	19
1920x1080	6x6	1920x1080	320x180	48.89	76	88	5	9
1920x1080	7x7	1920x1080	274x154 276x154 274x156 276x156*	36.39	74	88	3	8

\*7x7 uses multiple cell sizes. See Appendix B: 7x7 Grid Layout for a visual

## 13. T1A – GStreamer Ladder Generation

### 13.1 Transcoding

#### 13.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder.

Decoded YUV is split to multiple pads.

The YUV frames are encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

This test will generate 64 outputs of 1080p from a single 1080p input

This test is AVC to HEVC only

#### 13.1.2 Command line

See Appendix C: GStreamer Ladder Command

### 13.2 GStreamer Ladder Performance Results

Jobs	Outputs	FPS	CPU	Dec Load	Enc Load
1	64	18.97	74	1	92

## 14. T1U – FFmpeg Throughput

### 14.1 Decoding

#### 14.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe.

Bitstream is decoded by hardware decoder.

Decoded YUV frame is read out through PCIe and written into an output file.

#### 14.1.2 Command Line

```
ffmpeg -nostdin -f concat -safe 0 -c:v <dec>_ni_quadra_dec -dec 0 -  
xcoder-params multicoreJointMode=<*> -i /media/ramdisk/input.list -f  
null /dev/null -
```

<dec> is the decoder codec. eg h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

### 14.2 Encoding

#### 14.2.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 14.2.2 Command Line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f null  
/dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 8k, bitrate = 50000000, framerate = 24

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

## 14.3 Transcoding

### 14.3.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder.

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

### 14.3.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -c:v <dec>_ni_quadra_dec -dec 0 -  
xcoder-params out=hw:sempianar0=1:multicoreJointMode=<*> -i  
/media/ramdisk/input.list -c:v <enc>_ni_quadra_enc -enc 0 -xcoder-  
params intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f  
null /dev/null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 8k, bitrate = 50000000, framerate = 24

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)



#### 14.4 FFmpeg Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	94	0	102	14
HEVC to YUV	8k	1	0	8	1	95	0	100	16
VP9 to YUV	8k	1	0	8	1	25	0	30	2
YUV to AVC	8k	1	0	8	1	0	97	58	86
YUV to HEVC	8k	1	0	8	1	0	100	72	96
AVC to AVC	8k	1	1	8	1	70	99	48	3
AVC to HEVC	8k	1	1	8	1	75	99	63	8
HEVC to AVC	8k	1	1	8	1	60	100	47	2
HEVC to HEVC	8k	1	1	8	1	68	100	64	9
VP9 to AVC	8k	1	1	8	1	24	41	29	3
VP9 to HEVC	8k	1	1	8	1	24	40	30	0
AVC to YUV	8k	1	0	10	1	95	0	56	8
HEVC to YUV	8k	1	0	10	1	97	0	57	9
VP9 to YUV	8k	1	0	10	1	24	0	28	4
YUV to AVC	8k	1	0	10	1	0	80	41	118
YUV to HEVC	8k	1	0	10	1	0	57	42	114
AVC to YUV	4k	1	0	8	1	61	0	286	19
HEVC to YUV	4k	1	0	8	1	54	0	293	21
VP9 to YUV	4k	1	0	8	1	24	0	110	6
AVC to YUV	4k	16	0	8	0	100	0	420	0
HEVC to YUV	4k	16	0	8	0	99	0	452	1
VP9 to YUV	4k	16	0	8	0	99	0	400	0
YUV to AVC	4k	1	0	8	1	0	95	260	42
YUV to HEVC	4k	1	0	8	1	0	96	287	48
YUV to AV1	4k	1	0	8	1	0	94	246	38
YUV to AVC	4k	4	0	8	0	0	96	266	17
YUV to HEVC	4k	4	0	8	0	0	96	288	15
YUV to AV1	4k	4	0	8	0	0	92	240	23
YUV to AVC	4k	8	0	8	0	0	99	280	7
YUV to HEVC	4k	8	0	8	0	0	100	296	7
YUV to AV1	4k	8	0	8	0	0	99	256	7
AVC to AVC	4k	1	1	8	1	69	94	201	15
AVC to HEVC	4k	1	1	8	1	71	93	247	9
AVC to AV1	4k	1	1	8	1	59	96	235	12
HEVC to AVC	4k	1	1	8	1	57	97	203	9
HEVC to HEVC	4k	1	1	8	1	56	92	256	12
HEVC to AV1	4k	1	1	8	1	47	93	234	19
VP9 to AVC	4k	1	1	8	1	24	39	108	4
VP9 to HEVC	4k	1	1	8	1	23	36	109	2

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	42	109	4
AVC to AVC	4k	4	1	8	0	61	95	220	1
AVC to HEVC	4k	4	1	8	0	67	96	272	2
AVC to AV1	4k	4	1	8	0	58	95	244	6
HEVC to AVC	4k	4	1	8	0	54	95	216	2
HEVC to HEVC	4k	4	1	8	0	56	96	272	2
HEVC to AV1	4k	4	1	8	0	48	95	240	8
VP9 to AVC	4k	4	1	8	0	64	96	216	2
VP9 to HEVC	4k	4	1	8	0	69	96	264	2
VP9 to AV1	4k	4	1	8	0	60	92	228	2
AVC to AVC	4k	8	1	8	0	67	99	195	1
AVC to HEVC	4k	8	1	8	0	75	99	256	1
AVC to AV1	4k	8	1	8	0	68	99	241	6
HEVC to AVC	4k	8	1	8	0	61	99	195	1
HEVC to HEVC	4k	8	1	8	0	66	99	256	1
HEVC to AV1	4k	8	1	8	0	55	99	246	7
VP9 to AVC	4k	8	1	8	0	67	99	199	1
VP9 to HEVC	4k	8	1	8	0	73	99	256	1
VP9 to AV1	4k	8	1	8	0	62	100	248	5
AVC to YUV	4k	1	0	10	1	48	0	207	9
HEVC to YUV	4k	1	0	10	1	53	0	212	8
VP9 to YUV	4k	1	0	10	1	24	0	106	2
AVC to YUV	4k	16	0	10	0	96	0	256	0
HEVC to YUV	4k	16	0	10	0	100	0	256	0
VP9 to YUV	4k	16	0	10	0	100	0	235	0
YUV to AVC	4k	1	0	10	1	0	72	176	58
YUV to HEVC	4k	1	0	10	1	0	64	188	63
YUV to AV1	4k	1	0	10	1	0	70	183	67
YUV to AVC	4k	4	0	10	0	0	90	196	29
YUV to HEVC	4k	4	0	10	0	0	81	231	39
YUV to AV1	4k	4	0	10	0	0	89	224	39
AVC to YUV	1080p	1	0	8	1	48	0	796	19
HEVC to YUV	1080p	1	0	8	1	50	0	764	27
VP9 to YUV	1080p	1	0	8	1	22	0	456	6
AVC to YUV	1080p	40	0	8	0	97	0	1561	2
HEVC to YUV	1080p	40	0	8	0	99	0	1544	1
VP9 to YUV	1080p	40	0	8	0	93	0	1680	0
YUV to AVC	1080p	1	0	8	1	0	58	662	31
YUV to HEVC	1080p	1	0	8	1	0	56	669	32
YUV to AV1	1080p	1	0	8	1	0	58	608	28

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1121	3
YUV to HEVC	1080p	32	0	8	0	0	99	1184	3
YUV to AV1	1080p	32	0	8	0	0	99	1024	2
AVC to AVC	1080p	1	1	8	1	67	82	879	19
AVC to HEVC	1080p	1	1	8	1	67	80	933	19
AVC to AV1	1080p	1	1	8	1	54	82	798	16
HEVC to AVC	1080p	1	1	8	1	60	78	826	27
HEVC to HEVC	1080p	1	1	8	1	63	76	879	27
HEVC to AV1	1080p	1	1	8	1	52	77	759	22
VP9 to AVC	1080p	1	1	8	1	22	40	454	6
VP9 to HEVC	1080p	1	1	8	1	22	38	454	4
VP9 to AV1	1080p	1	1	8	1	22	44	455	7
AVC to AVC	1080p	32	1	8	0	76	99	864	1
AVC to HEVC	1080p	32	1	8	0	84	99	970	1
AVC to AV1	1080p	32	1	8	0	77	99	928	2
HEVC to AVC	1080p	32	1	8	0	71	99	897	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1013	1
HEVC to AV1	1080p	32	1	8	0	71	99	943	3
VP9 to AVC	1080p	32	1	8	0	70	99	992	0
VP9 to HEVC	1080p	32	1	8	0	73	99	1111	0
VP9 to AV1	1080p	32	1	8	0	68	99	992	0
AVC to YUV	1080p	1	0	10	1	30	0	490	7
HEVC to YUV	1080p	1	0	10	1	28	0	494	7
VP9 to YUV	1080p	1	0	10	1	23	0	362	4
AVC to YUV	1080p	40	0	10	0	72	0	1080	0
HEVC to YUV	1080p	40	0	10	0	77	0	1080	0
VP9 to YUV	1080p	40	0	10	0	89	0	1038	0
YUV to AVC	1080p	1	0	10	1	0	36	418	36
YUV to HEVC	1080p	1	0	10	1	0	34	417	37
YUV to AV1	1080p	1	0	10	1	0	38	393	32
YUV to AVC	1080p	32	0	10	0	0	71	801	6
YUV to HEVC	1080p	32	0	10	0	0	67	803	8
YUV to AV1	1080p	32	0	10	0	0	76	778	6
AVC to YUV	720p	1	0	8	1	45	0	1119	15
HEVC to YUV	720p	1	0	8	1	39	0	1112	20
VP9 to YUV	720p	1	0	8	1	29	0	863	7
AVC to YUV	720p	100	0	8	0	100	0	2400	0
HEVC to YUV	720p	100	0	8	0	98	0	2713	0
VP9 to YUV	720p	100	0	8	0	100	0	2500	0
YUV to AVC	720p	1	0	8	1	0	33	863	20

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	34	875	19
YUV to AV1	720p	1	0	8	1	0	35	779	14
YUV to AVC	720p	64	0	8	0	0	97	2305	2
YUV to HEVC	720p	64	0	8	0	0	98	2366	2
YUV to AV1	720p	64	0	8	0	0	97	1941	2
AVC to AVC	720p	1	1	8	1	49	48	1209	20
AVC to HEVC	720p	1	1	8	1	48	48	1214	19
AVC to AV1	720p	1	1	8	1	41	51	1052	15
HEVC to AVC	720p	1	1	8	1	41	46	1170	21
HEVC to HEVC	720p	1	1	8	1	41	46	1176	22
HEVC to AV1	720p	1	1	8	1	35	49	1024	22
VP9 to AVC	720p	1	1	8	1	28	34	862	7
VP9 to HEVC	720p	1	1	8	1	28	33	856	10
VP9 to AV1	720p	1	1	8	1	28	40	856	7
AVC to AVC	720p	64	1	8	0	96	100	1876	1
AVC to HEVC	720p	64	1	8	0	98	100	1965	0
AVC to AV1	720p	64	1	8	0	78	100	1666	0
HEVC to AVC	720p	64	1	8	0	86	100	1880	1
HEVC to HEVC	720p	64	1	8	0	88	100	1984	0
HEVC to AV1	720p	64	1	8	0	68	100	1669	0
VP9 to AVC	720p	64	1	8	0	100	100	2112	0
VP9 to HEVC	720p	64	1	8	0	100	100	2182	0
VP9 to AV1	720p	64	1	8	0	78	100	1795	0

## 15. T1U – Libxcoder Throughput

### 15.1 Decoding

#### 15.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe.

Bitstream is decoded by hardware decoder.

Decoded YUV frame is read out through PCIe and written into an output file.

#### 15.1.2 Command Line

```
./ni_xcoder_decode -c 0 -r 1000 -i /media/ramdisk/input.<ext> -m  
<test_type> -o /dev/null -d multicoreJointMode=<*>
```

<test\_type> = test codecs. ie. a (avc), h (hevc), etc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

Note: Libxcoder decoding tests were run without multi-threading (but with multicoreJointMode enabled where noted)

### 15.2 Encoding

#### 15.2.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 15.2.2 Command Line

```
./ni_xcoder_encode -c 0 -s <resolution> -r 1000 -i  
/media/ramdisk/input.yuv -m <test_type> -o /dev/null -e  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*>
```

<test\_type> = test codecs. ie. a (avc), h (hevc), etc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 8k, bitrate = 50000000, framerate = 24

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

Note: Libxcoder encoding tests were run without multi-threading (but with multicoreJointMode enabled where noted)

## 15.3 Transcoding

### 15.3.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder.

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

### 15.3.2 Command line

```
./ni_xcoder_multithread_transcode -c 0 -r 1000 -i  
/media/ramdisk/input.<ext> -m <dec_test_type> -n <enc_test_type> -o  
/dev/null -e  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*> -d out=hw:semiplanar0=1:multicoreJointMode=1
```

<dec\_test\_type> = decoding test codecs. ie. a (avc), h (hevc), etc

<enc\_test\_type> = encoding test codecs. ie. a (avc), h (hevc), etc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 8k, bitrate = 50000000, framerate = 24

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

## 15.4 Libxcode Throughput Performance Results

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	79	0	89	7
HEVC to YUV	8k	1	0	8	1	79	0	91	7
VP9 to YUV	8k	1	0	8	1	25	0	30	2
YUV to AVC	8k	1	0	8	1	0	88	57	26
YUV to HEVC	8k	1	0	8	1	0	95	72	32
AVC to AVC	8k	1	1	8	1	76	99	48	3
AVC to HEVC	8k	1	1	8	1	78	99	64	4
HEVC to AVC	8k	1	1	8	1	64	100	47	3
HEVC to HEVC	8k	1	1	8	1	74	100	62	3
VP9 to AVC	8k	1	1	8	1	24	42	29	1
VP9 to HEVC	8k	1	1	8	1	24	38	30	2
AVC to YUV	8k	1	0	10	1	66	0	51	8
HEVC to YUV	8k	1	0	10	1	55	0	47	7
VP9 to YUV	8k	1	0	10	1	25	0	28	4
YUV to AVC	8k	1	0	10	1	0	63	35	35
YUV to HEVC	8k	1	0	10	1	0	49	36	37
AVC to YUV	4k	1	0	8	1	58	0	268	7
HEVC to YUV	4k	1	0	8	1	51	0	278	6
VP9 to YUV	4k	1	0	8	1	24	0	109	2
AVC to YUV	4k	16	0	8	0	99	0	425	0
HEVC to YUV	4k	16	0	8	0	99	0	456	0
VP9 to YUV	4k	16	0	8	0	100	0	406	0
YUV to AVC	4k	1	0	8	1	0	84	232	23
YUV to HEVC	4k	1	0	8	1	0	80	240	23
YUV to AV1	4k	1	0	8	1	0	86	224	23
YUV to AVC	4k	4	0	8	0	0	96	272	10
YUV to HEVC	4k	4	0	8	0	0	97	291	11
YUV to AV1	4k	4	0	8	0	0	97	252	10
YUV to AVC	4k	8	0	8	0	0	95	271	15
YUV to HEVC	4k	8	0	8	0	0	99	302	8
YUV to AV1	4k	8	0	8	0	0	100	262	6
AVC to AVC	4k	1	1	8	0	68	97	205	5
AVC to HEVC	4k	1	1	8	0	72	96	251	6
AVC to AV1	4k	1	1	8	0	64	97	236	6
HEVC to AVC	4k	1	1	8	0	58	97	205	3
HEVC to HEVC	4k	1	1	8	0	61	97	256	4
HEVC to AV1	4k	1	1	8	0	53	97	237	4
VP9 to AVC	4k	1	1	8	0	24	38	108	2

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	0	24	36	109	2
VP9 to AV1	4k	1	1	8	0	24	42	109	2
AVC to AVC	4k	4	1	8	0	60	96	225	3
AVC to HEVC	4k	4	1	8	0	67	95	275	4
AVC to AV1	4k	4	1	8	0	53	95	246	4
HEVC to AVC	4k	4	1	8	0	55	96	219	3
HEVC to HEVC	4k	4	1	8	0	58	96	265	3
HEVC to AV1	4k	4	1	8	0	50	96	242	3
VP9 to AVC	4k	4	1	8	0	62	96	219	3
VP9 to HEVC	4k	4	1	8	0	70	96	268	3
VP9 to AV1	4k	4	1	8	0	57	95	242	3
AVC to AVC	4k	8	1	8	0	68	99	198	2
AVC to HEVC	4k	8	1	8	0	74	99	256	2
AVC to AV1	4k	8	1	8	0	67	100	247	2
HEVC to AVC	4k	8	1	8	0	62	100	198	2
HEVC to HEVC	4k	8	1	8	0	65	100	254	2
HEVC to AV1	4k	8	1	8	0	56	100	244	2
VP9 to AVC	4k	8	1	8	0	67	100	202	2
VP9 to HEVC	4k	8	1	8	0	75	100	261	2
VP9 to AV1	4k	8	1	8	0	65	100	249	2
AVC to YUV	4k	1	0	10	1	48	0	191	6
HEVC to YUV	4k	1	0	10	1	42	0	186	6
VP9 to YUV	4k	1	0	10	1	24	0	105	4
AVC to YUV	4k	16	0	10	0	96	0	258	0
HEVC to YUV	4k	16	0	10	0	97	0	256	0
VP9 to YUV	4k	16	0	10	0	99	0	238	0
YUV to AVC	4k	1	0	10	1	0	47	132	25
YUV to HEVC	4k	1	0	10	1	0	42	130	26
YUV to AV1	4k	1	0	10	1	0	47	123	23
YUV to AVC	4k	4	0	10	0	0	56	167	26
YUV to HEVC	4k	4	0	10	0	0	62	170	27
YUV to AV1	4k	4	0	10	0	0	57	157	28
AVC to YUV	1080p	1	0	8	1	41	0	666	13
HEVC to YUV	1080p	1	0	8	1	46	0	696	10
VP9 to YUV	1080p	1	0	8	1	22	0	456	3
AVC to YUV	1080p	40	0	8	0	91	0	1497	0
HEVC to YUV	1080p	40	0	8	0	97	0	1582	0
VP9 to YUV	1080p	40	0	8	0	81	0	1538	0
YUV to AVC	1080p	1	0	8	1	0	55	625	16



TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	52	627	15
YUV to AV1	1080p	1	0	8	1	0	50	509	17
YUV to AVC	1080p	32	0	8	0	0	99	1150	2
YUV to HEVC	1080p	32	0	8	0	0	99	1216	2
YUV to AV1	1080p	32	0	8	0	0	99	1053	2
AVC to AVC	1080p	1	1	8	0	70	86	922	15
AVC to HEVC	1080p	1	1	8	0	70	84	970	14
AVC to AV1	1080p	1	1	8	0	55	85	826	15
HEVC to AVC	1080p	1	1	8	0	68	87	910	10
HEVC to HEVC	1080p	1	1	8	0	70	85	970	11
HEVC to AV1	1080p	1	1	8	0	57	85	824	11
VP9 to AVC	1080p	1	1	8	0	22	40	457	3
VP9 to HEVC	1080p	1	1	8	0	22	38	454	3
VP9 to AV1	1080p	1	1	8	0	22	44	454	5
AVC to AVC	1080p	32	1	8	0	75	99	898	0
AVC to HEVC	1080p	32	1	8	0	84	99	1012	0
AVC to AV1	1080p	32	1	8	0	76	99	967	0
HEVC to AVC	1080p	32	1	8	0	72	99	931	0
HEVC to HEVC	1080p	32	1	8	0	77	99	1044	0
HEVC to AV1	1080p	32	1	8	0	70	100	974	0
VP9 to AVC	1080p	32	1	8	0	69	99	1024	0
VP9 to HEVC	1080p	32	1	8	0	73	99	1142	0
VP9 to AV1	1080p	32	1	8	0	70	99	1026	0
AVC to YUV	1080p	1	0	10	1	30	0	479	5
HEVC to YUV	1080p	1	0	10	1	27	0	475	6
VP9 to YUV	1080p	1	0	10	1	24	0	362	4
AVC to YUV	1080p	40	0	10	0	70	0	1039	0
HEVC to YUV	1080p	40	0	10	0	77	0	1039	0
VP9 to YUV	1080p	40	0	10	0	78	0	987	0
YUV to AVC	1080p	1	0	10	1	0	33	374	19
YUV to HEVC	1080p	1	0	10	1	0	31	374	19
YUV to AV1	1080p	1	0	10	1	0	32	334	18
YUV to AVC	1080p	32	0	10	0	0	63	726	4
YUV to HEVC	1080p	32	0	10	0	0	60	734	5
YUV to AV1	1080p	32	0	10	0	0	64	690	8
AVC to YUV	720p	1	0	8	1	42	0	1069	10
HEVC to YUV	720p	1	0	8	1	39	0	1110	8
VP9 to YUV	720p	1	0	8	1	29	0	863	3
AVC to YUV	720p	100	0	8	0	100	0	2584	0

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	90	0	2653	0
VP9 to YUV	720p	100	0	8	0	94	0	2500	0
YUV to AVC	720p	1	0	8	1	0	37	942	11
YUV to HEVC	720p	1	0	8	1	0	36	936	10
YUV to AV1	720p	1	0	8	1	0	32	726	10
YUV to AVC	720p	64	0	8	0	0	95	2440	1
YUV to HEVC	720p	64	0	8	0	0	94	2471	1
YUV to AV1	720p	64	0	8	0	0	100	2117	1
AVC to AVC	720p	1	1	8	0	52	50	1271	13
AVC to HEVC	720p	1	1	8	0	51	49	1271	13
AVC to AV1	720p	1	1	8	0	44	50	1047	13
HEVC to AVC	720p	1	1	8	0	45	51	1287	10
HEVC to HEVC	720p	1	1	8	0	45	51	1292	10
HEVC to AV1	720p	1	1	8	0	39	49	1048	11
VP9 to AVC	720p	1	1	8	0	29	33	860	5
VP9 to HEVC	720p	1	1	8	0	28	33	855	5
VP9 to AV1	720p	1	1	8	0	28	39	856	6
AVC to AVC	720p	64	1	8	0	100	100	2014	0
AVC to HEVC	720p	64	1	8	0	100	100	2104	0
AVC to AV1	720p	64	1	8	0	95	100	1770	0
HEVC to AVC	720p	64	1	8	0	87	99	2013	0
HEVC to HEVC	720p	64	1	8	0	96	100	2104	0
HEVC to AV1	720p	64	1	8	0	89	100	1773	0
VP9 to AVC	720p	64	1	8	0	100	100	2278	0
VP9 to HEVC	720p	64	1	8	0	100	100	2342	0
VP9 to AV1	720p	64	1	8	0	99	100	1927	0

## 16. T1U – FFmpeg Latency

### 16.1 Encoding

#### 16.1.1 Description

Libxcodec is compiled and installed with parameter `--with-latency-display`

```
$ bash build.sh --with-latency-display
```

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

For each frame, the encoder latency (eLat) value is provided in the output log.

All eLat values are parsed from the output log and the last 50% of frame data before killing ffmpeg instances is used to calculate the Average, Min, Max, and Variance.

The first 50% of frame data are ignored to reach stability while launching multiple jobs.

#### 16.1.2 Command Line

```
ffmpeg -re -loglevel info -f rawvideo -pix_fmt yuv420p -stream_loop  
1000 -s:v <resolution> -i /media/ramdisk/input.yuv -c:v  
<enc>_ni_quadra_enc -enc 0 -xcodec-params gopPresetIdx=9:lowDelay=1 -f  
null -
```

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<resolution> is resolution of input

## 16.2 FFmpeg Latency Performance Results

TYPE	RESOLUTION	JOBS	ELAT_AVG (ms)	ELAT_MAX (ms)	ELAT_MIN (ms)	ELAT_VAR (ms)
YUV to AVC	8k	1	65.89	66.68	65.2	0.09
YUV to HEVC	8k	1	63.12	69.05	61.18	1.35
YUV to AVC	4k	1	17.51	20.05	17.03	0.08
YUV to HEVC	4k	1	18.95	21.4	17.7	0.13
YUV to AV1	4k	1	24.8	29.06	18.8	0.84
YUV to AVC	4k	4	17.73	22.44	16.91	0.21
YUV to HEVC	4k	4	19.53	22.51	17.72	1.08
YUV to AV1	4k	4	25.03	29.59	18.48	0.83
YUV to AVC	4k	8	20.63	31.82	17.14	23.28
YUV to HEVC	4k	8	24.83	36.15	17.73	29.82
YUV to AV1	4k	8	42.82	52.75	27.27	11.29
YUV to AVC	1080p	1	5.46	6.39	5.28	0.01
YUV to HEVC	1080p	1	5.75	8.83	5.35	0.05
YUV to AV1	1080p	1	7.73	8.55	5.79	0.07
YUV to AVC	1080p	32	8.15	14.51	5.34	2.2
YUV to HEVC	1080p	32	10.07	15.38	6.09	2.43
YUV to AV1	1080p	32	47.14	55.83	40.27	2.58
YUV to AVC	720p	1	3.11	3.73	2.8	0.02
YUV to HEVC	720p	1	3.17	3.52	2.89	0
YUV to AV1	720p	1	4.16	4.75	3.28	0.01
YUV to AVC	720p	64	7.24	11.71	4.23	0.91
YUV to HEVC	720p	64	7.5	13.72	4.45	0.91
YUV to AV1	720p	64	45.7	53.22	36.97	3.43

## 17. T1U – Decoder PPU Scaling

### 17.1 Decoding

#### 17.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe.

Bitstream is decoded by hardware decoder and scaled to 224x224 with decoder post processing unit.

Decoded YUV is kept on device.

The YUV frame is converted to RGBA format with 2D Engine.

The RGBA frame is read out through PCIe and written into an output file.

#### 17.1.2 Command Line

```
ffmpeg -vsync 0 -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params  
out=hw:scale0=224x224:multicoreJointMode=<resolution=8k?1:0> -f concat  
-safe 0 -i /media/ramdisk/input.list -vf  
ni_quadra_scale=iw:ih:format=rgba,hwdownload,format=rgba -f null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

### 17.2 Decoder PPU Scaling Performance Results

TYPE	RESOLUTION	JOBS	DEC_LOAD	SCALER_LOAD	FPS	CPU
AVC to RGBA	8k	1	93	0	117	5
HEVC to RGBA	8k	1	90	0	120	6
VP9 to RGBA	8k	1	23	0	31	1
AVC to RGBA	4k	1	22	0	114	6
AVC to RGBA	4k	16	94	2	468	1
HEVC to RGBA	4k	1	22	0	137	8
HEVC to RGBA	4k	16	95	3	549	2
VP9 to RGBA	4k	1	22	0	112	4
VP9 to RGBA	4k	16	93	2	462	1
AVC to RGBA	1080p	40	93	12	1603	1
HEVC to RGBA	1080p	40	94	11	1598	1
VP9 to RGBA	1080p	40	93	15	2060	0
AVC to RGBA	720p	100	99	18	2433	0
HEVC to RGBA	720p	100	91	20	2695	0
VP9 to RGBA	720p	64	97	19	2613	0

## 18. T1U – Streaming Ladder Generation

### 18.1 Transcoding

#### 18.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder split and scaled to smaller resolutions with decoder post processing unit or 2D Engine.

Decoded YUV frame is kept on device.

The YUV frames are encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

#### 18.1.2 Command line

```
ffmpeg -vsync 0 -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params  
out=hw:semiplanar0=1:enableOut1=1:semiplanar1=1:scale1=1280x720:enableO  
ut2=1:semiplanar2=1:scale2=960x540 -f concat -safe 0 -i  
/media/ramdisk/input.list -filter_complex  
'[0:v]ni_quadra_split=2:1:2[1080p][1080p_1][720p][540p][540p_1];[540p_1  
]ni_quadra_scale=640x360[360p]' -map [1080p] -xcoder-params  
RcEnable=1:bitrate=3500000 -c:v <enc>_ni_quadra_enc -enc 0 -f null - -  
map [1080p_1] -xcoder-params RcEnable=1:bitrate=1800000 -c:v  
<enc>_ni_quadra_enc -enc 0 -f null - -map [720p] -xcoder-params  
RcEnable=1:bitrate=1000000 -c:v <enc>_ni_quadra_enc -enc 0 -f null - -  
map [540p] -xcoder-params RcEnable=1:bitrate=800000 -c:v  
<enc>_ni_quadra_enc -enc 0 -f null - -map [360p] -xcoder-params  
RcEnable=1:bitrate=500000 -c:v <enc>_ni_quadra_enc -enc 0 -f null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

Input: 1080p

Output: 1080p, 1080p, 720p(PPU Scale), 540p(PPU Scale), 360p(2D Scale)

### 18.2 Streaming Ladder Generation Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	8	29	93	2	386	3
AVC to HEVC	8	29	94	2	411	3
AVC to AV1	8	23	94	1	352	3
HEVC to AVC	8	31	95	2	392	4
HEVC to HEVC	8	30	92	2	414	3
HEVC to AV1	8	23	92	1	352	3
VP9 to AVC	8	37	93	2	384	3
VP9 to HEVC	8	37	93	2	410	3
VP9 to AV1	8	30	94	1	352	3

## 19. T1U – RGBA Encoding

### 19.1 Encoding

#### 19.1.1 Description

RGBA frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

RGBA frame is uploaded and encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 19.1.2 Command line

```
ffmpeg -nostdin -stream_loop -1 -f rawvideo -pix_fmt rgba -s:v  
<resolution> -r 30 -i /media/ramdisk/input.rgb -vf  
"ni_quadra_hwupload=0" -c:v <enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f null  
/dev/null
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30

<resolution> == 720p, bitrate = 1500000, framerate = 30

## 19.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	51	142	68
RGBA to HEVC	4k	1	1	48	145	72
RGBA to AV1	4k	1	1	56	145	72
RGBA to AVC	4k	4	0	70	152	37
RGBA to HEVC	4k	4	0	63	158	37
RGBA to AV1	4k	4	0	68	160	42
RGBA to AVC	4k	8	0	66	161	34
RGBA to HEVC	4k	8	0	61	168	33
RGBA to AV1	4k	8	0	66	169	37
RGBA to AVC	1080p	1	1	34	385	46
RGBA to HEVC	1080p	1	1	32	386	46
RGBA to AV1	1080p	1	1	36	372	46
RGBA to AVC	1080p	16	0	63	598	18
RGBA to HEVC	1080p	16	0	59	608	18
RGBA to AV1	1080p	16	0	61	605	21
RGBA to AVC	1080p	32	0	62	640	12
RGBA to HEVC	1080p	32	0	58	640	14
RGBA to AV1	1080p	32	0	62	641	14
RGBA to AVC	720p	1	1	28	648	58
RGBA to HEVC	720p	1	1	29	648	57
RGBA to AV1	720p	1	1	33	626	50
RGBA to AVC	720p	16	0	49	1186	35
RGBA to HEVC	720p	16	0	48	1185	36
RGBA to AV1	720p	16	0	56	1133	33
RGBA to AVC	720p	32	0	47	1145	32
RGBA to HEVC	720p	32	0	46	1147	31
RGBA to AV1	720p	32	0	56	1154	31



## 20. T1U – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

### 20.1 Encoding

#### 20.1.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder with a mix of xcoder-params EnableRdoQuant, rdoLevel, and lookaheadDepth.

Encoded bitstream is read out through PCIe and written into an output file.

#### 20.1.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:lookaheadDepth=<*>:EnableRdoQuant=  
<*>:rdoLevel=<*> -f null /dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30

## 20.2 Encoding EnableRdoQuant/rdoLevel/lookaheadDepth Performance Results

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	0	0	1	96	262	18
YUV to HEVC	4k	4	0	0	1	97	292	12
YUV to AV1	4k	4	0	0	1	97	252	11
YUV to AVC	4k	4	0	0	2	95	264	18
YUV to HEVC	4k	4	0	0	2	99	148	6
YUV to AV1	4k	4	0	0	2	99	120	5
YUV to AVC	4k	4	0	0	3	96	268	15
YUV to HEVC	4k	4	0	0	3	99	88	5
YUV to AV1	4k	4	0	0	3	95	64	3
YUV to AVC	4k	4	0	1	1	98	164	7
YUV to HEVC	4k	4	0	1	1	97	208	9
YUV to AVC	4k	4	0	1	2	98	164	7
YUV to HEVC	4k	4	0	1	2	97	96	4
YUV to AVC	4k	4	0	1	3	98	164	7
YUV to HEVC	4k	4	0	1	3	99	60	4
YUV to AVC	4k	4	4	0	1	100	172	9
YUV to HEVC	4k	4	4	0	1	99	200	12
YUV to AV1	4k	4	4	0	1	99	172	8
YUV to AVC	4k	4	4	0	2	99	172	8
YUV to HEVC	4k	4	4	0	2	100	120	6
YUV to AV1	4k	4	4	0	2	100	100	6
YUV to AVC	4k	4	4	0	3	99	172	9
YUV to HEVC	4k	4	4	0	3	98	76	3
YUV to AV1	4k	4	4	0	3	100	64	3
YUV to AVC	4k	4	4	1	1	99	120	6
YUV to HEVC	4k	4	4	1	1	99	156	7
YUV to AVC	4k	4	4	1	2	100	120	6
YUV to HEVC	4k	4	4	1	2	98	83	5
YUV to AVC	4k	4	4	1	3	99	120	6
YUV to HEVC	4k	4	4	1	3	97	56	3
YUV to AVC	4k	4	16	0	1	99	172	8
YUV to HEVC	4k	4	16	0	1	100	196	11
YUV to AV1	4k	4	16	0	1	99	172	11
YUV to AVC	4k	4	16	0	2	99	172	8
YUV to HEVC	4k	4	16	0	2	100	120	6
YUV to AV1	4k	4	16	0	2	99	100	5
YUV to AVC	4k	4	16	0	3	99	172	9
YUV to HEVC	4k	4	16	0	3	98	76	4
YUV to AV1	4k	4	16	0	3	99	64	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	16	1	1	100	120	6
YUV to HEVC	4k	4	16	1	1	100	156	8
YUV to AVC	4k	4	16	1	2	99	120	5
YUV to HEVC	4k	4	16	1	2	99	80	4
YUV to AVC	4k	4	16	1	3	100	120	6
YUV to HEVC	4k	4	16	1	3	100	56	3
YUV to AVC	4k	4	40	0	1	99	171	7
YUV to HEVC	4k	4	40	0	1	99	196	12
YUV to AV1	4k	4	40	0	1	100	172	9
YUV to AVC	4k	4	40	0	2	99	171	9
YUV to HEVC	4k	4	40	0	2	99	116	6
YUV to AV1	4k	4	40	0	2	100	100	5
YUV to AVC	4k	4	40	0	3	99	171	10
YUV to HEVC	4k	4	40	0	3	99	76	5
YUV to AV1	4k	4	40	0	3	100	60	3
YUV to AVC	4k	4	40	1	1	99	120	6
YUV to HEVC	4k	4	40	1	1	99	152	6
YUV to AVC	4k	4	40	1	2	100	120	6
YUV to HEVC	4k	4	40	1	2	98	80	4
YUV to AVC	4k	4	40	1	3	99	120	5
YUV to HEVC	4k	4	40	1	3	100	55	4
YUV to AVC	1080p	20	0	0	1	99	1120	4
YUV to HEVC	1080p	20	0	0	1	99	1184	4
YUV to AV1	1080p	20	0	0	1	99	1020	3
YUV to AVC	1080p	20	0	0	2	99	1120	4
YUV to HEVC	1080p	20	0	0	2	100	600	2
YUV to AV1	1080p	20	0	0	2	99	481	2
YUV to AVC	1080p	20	0	0	3	99	1120	5
YUV to HEVC	1080p	20	0	0	3	100	360	1
YUV to AV1	1080p	20	0	0	3	100	260	1
YUV to AVC	1080p	20	0	1	1	99	660	2
YUV to HEVC	1080p	20	0	1	1	99	840	2
YUV to AVC	1080p	20	0	1	2	99	660	2
YUV to HEVC	1080p	20	0	1	2	99	380	1
YUV to AVC	1080p	20	0	1	3	100	660	2
YUV to HEVC	1080p	20	0	1	3	99	240	1
YUV to AVC	1080p	20	4	0	1	99	615	2
YUV to HEVC	1080p	20	4	0	1	99	717	2
YUV to AV1	1080p	20	4	0	1	99	621	2
YUV to AVC	1080p	20	4	0	2	99	618	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	4	0	2	100	442	2
YUV to AV1	1080p	20	4	0	2	99	374	1
YUV to AVC	1080p	20	4	0	3	99	619	2
YUV to HEVC	1080p	20	4	0	3	99	300	1
YUV to AV1	1080p	20	4	0	3	99	240	1
YUV to AVC	1080p	20	4	1	1	99	440	2
YUV to HEVC	1080p	20	4	1	1	99	563	2
YUV to AVC	1080p	20	4	1	2	99	440	1
YUV to HEVC	1080p	20	4	1	2	100	320	1
YUV to AVC	1080p	20	4	1	3	100	440	1
YUV to HEVC	1080p	20	4	1	3	99	220	1
YUV to AVC	1080p	20	16	0	1	99	604	2
YUV to HEVC	1080p	20	16	0	1	99	714	2
YUV to AV1	1080p	20	16	0	1	99	622	2
YUV to AVC	1080p	20	16	0	2	99	607	2
YUV to HEVC	1080p	20	16	0	2	99	440	1
YUV to AV1	1080p	20	16	0	2	99	364	1
YUV to AVC	1080p	20	16	0	3	99	606	2
YUV to HEVC	1080p	20	16	0	3	99	297	1
YUV to AV1	1080p	20	16	0	3	99	237	1
YUV to AVC	1080p	20	16	1	1	99	440	1
YUV to HEVC	1080p	20	16	1	1	99	566	2
YUV to AVC	1080p	20	16	1	2	99	440	2
YUV to HEVC	1080p	20	16	1	2	99	310	1
YUV to AVC	1080p	20	16	1	3	99	440	1
YUV to HEVC	1080p	20	16	1	3	99	215	1
YUV to AVC	1080p	20	40	0	1	99	600	2
YUV to HEVC	1080p	20	40	0	1	99	700	2
YUV to AV1	1080p	20	40	0	1	99	620	2
YUV to AVC	1080p	20	40	0	2	99	600	2
YUV to HEVC	1080p	20	40	0	2	99	440	1
YUV to AV1	1080p	20	40	0	2	99	360	1
YUV to AVC	1080p	20	40	0	3	99	600	2
YUV to HEVC	1080p	20	40	0	3	100	288	1
YUV to AV1	1080p	20	40	0	3	99	229	1
YUV to AVC	1080p	20	40	1	1	99	440	1
YUV to HEVC	1080p	20	40	1	1	99	560	2
YUV to AVC	1080p	20	40	1	2	99	440	1
YUV to HEVC	1080p	20	40	1	2	100	302	1
YUV to AVC	1080p	20	40	1	3	99	436	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	40	1	3	100	204	1
YUV to AVC	720p	40	0	0	1	95	2275	2
YUV to HEVC	720p	40	0	0	1	92	2295	3
YUV to AV1	720p	40	0	0	1	98	1991	2
YUV to AVC	720p	40	0	0	2	94	2281	2
YUV to HEVC	720p	40	0	0	2	99	1320	1
YUV to AV1	720p	40	0	0	2	99	1044	1
YUV to AVC	720p	40	0	0	3	96	2280	2
YUV to HEVC	720p	40	0	0	3	99	800	1
YUV to AV1	720p	40	0	0	3	99	562	0
YUV to AVC	720p	40	0	1	1	99	1480	1
YUV to HEVC	720p	40	0	1	1	99	1804	1
YUV to AVC	720p	40	0	1	2	99	1480	1
YUV to HEVC	720p	40	0	1	2	99	840	1
YUV to AVC	720p	40	0	1	3	99	1480	1
YUV to HEVC	720p	40	0	1	3	99	529	0
YUV to AVC	720p	40	4	0	1	99	1247	1
YUV to HEVC	720p	40	4	0	1	100	1320	1
YUV to AV1	720p	40	4	0	1	100	976	0
YUV to AVC	720p	40	4	0	2	99	1251	1
YUV to HEVC	720p	40	4	0	2	99	921	1
YUV to AV1	720p	40	4	0	2	99	760	0
YUV to AVC	720p	40	4	0	3	99	1256	1
YUV to HEVC	720p	40	4	0	3	99	640	0
YUV to AV1	720p	40	4	0	3	99	480	0
YUV to AVC	720p	40	4	1	1	99	921	1
YUV to HEVC	720p	40	4	1	1	99	1160	0
YUV to AVC	720p	40	4	1	2	99	920	1
YUV to HEVC	720p	40	4	1	2	99	665	0
YUV to AVC	720p	40	4	1	3	99	920	1
YUV to HEVC	720p	40	4	1	3	99	447	0
YUV to AVC	720p	40	16	0	1	99	1240	1
YUV to HEVC	720p	40	16	0	1	100	1320	1
YUV to AV1	720p	40	16	0	1	100	1040	0
YUV to AVC	720p	40	16	0	2	99	1240	1
YUV to HEVC	720p	40	16	0	2	99	920	1
YUV to AV1	720p	40	16	0	2	99	760	0
YUV to AVC	720p	40	16	0	3	99	1241	1
YUV to HEVC	720p	40	16	0	3	99	640	0
YUV to AV1	720p	40	16	0	3	99	480	0

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	40	16	1	1	99	920	1
YUV to HEVC	720p	40	16	1	1	99	1160	1
YUV to AVC	720p	40	16	1	2	99	920	1
YUV to HEVC	720p	40	16	1	2	99	643	0
YUV to AVC	720p	40	16	1	3	99	920	0
YUV to HEVC	720p	40	16	1	3	99	441	0
YUV to AVC	720p	40	40	0	1	99	1240	1
YUV to HEVC	720p	40	40	0	1	100	1280	1
YUV to AV1	720p	40	40	0	1	100	1002	1
YUV to AVC	720p	40	40	0	2	99	1239	1
YUV to HEVC	720p	40	40	0	2	99	920	1
YUV to AV1	720p	40	40	0	2	99	758	0
YUV to AVC	720p	40	40	0	3	99	1239	1
YUV to HEVC	720p	40	40	0	3	99	630	0
YUV to AV1	720p	40	40	0	3	99	480	0
YUV to AVC	720p	40	40	1	1	99	917	1
YUV to HEVC	720p	40	40	1	1	99	1142	1
YUV to AVC	720p	40	40	1	2	99	918	1
YUV to HEVC	720p	40	40	1	2	99	640	0
YUV to AVC	720p	40	40	1	3	99	918	1
YUV to HEVC	720p	40	40	1	3	99	440	0

## 21. T1U – Capped CRF

### 21.1 Encoding with lookaheadDepth

#### 21.1.1 Description

YUV frame is read from an input file on ramdisk and fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder with a mix of xcoder-params EnableRdoQuant, rdoLevel, lookaheadDepth, CRF, bitrate, and vbvBufferSize.

Encoded bitstream is read out through PCIe and written into an output file.

#### 21.1.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:vbvBufferSize=1000:bitrate=<*>:lookaheadDepth=<*>:EnableR  
doQuant=<*>:rdoLevel=<*>:crf=<*> -f null /dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30

### 21.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	19	99	619	2
YUV to HEVC	1080p	20	0	0	1	19	99	719	2
YUV to AV1	1080p	20	0	0	1	19	99	627	2
YUV to AVC	1080p	20	0	0	2	19	99	619	2
YUV to HEVC	1080p	20	0	0	2	19	99	440	1
YUV to AV1	1080p	20	0	0	2	19	99	380	1
YUV to AVC	1080p	20	0	0	3	19	99	619	2
YUV to HEVC	1080p	20	0	0	3	19	99	300	1
YUV to AV1	1080p	20	0	0	3	19	100	240	1
YUV to AVC	1080p	20	0	1	1	19	100	441	1
YUV to HEVC	1080p	20	0	1	1	19	99	579	2
YUV to AVC	1080p	20	0	1	2	19	99	440	1
YUV to HEVC	1080p	20	0	1	2	19	99	320	1
YUV to AVC	1080p	20	0	1	3	19	99	440	1
YUV to HEVC	1080p	20	0	1	3	19	100	220	1
YUV to AVC	1080p	20	4	0	1	19	99	617	2
YUV to HEVC	1080p	20	4	0	1	19	99	720	2
YUV to AV1	1080p	20	4	0	1	19	99	620	2
YUV to AVC	1080p	20	4	0	2	19	99	617	2
YUV to HEVC	1080p	20	4	0	2	19	99	445	1
YUV to AV1	1080p	20	4	0	2	19	99	368	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	3	19	99	614	2
YUV to HEVC	1080p	20	4	0	3	19	99	300	1
YUV to AV1	1080p	20	4	0	3	19	100	240	1
YUV to AVC	1080p	20	4	1	1	19	100	440	1
YUV to HEVC	1080p	20	4	1	1	19	99	571	2
YUV to AVC	1080p	20	4	1	2	19	100	440	1
YUV to HEVC	1080p	20	4	1	2	19	100	320	1
YUV to AVC	1080p	20	4	1	3	19	99	440	1
YUV to HEVC	1080p	20	4	1	3	19	99	220	1
YUV to AVC	1080p	20	16	0	1	19	99	604	2
YUV to HEVC	1080p	20	16	0	1	19	99	712	2
YUV to AV1	1080p	20	16	0	1	19	99	620	2
YUV to AVC	1080p	20	16	0	2	19	99	605	2
YUV to HEVC	1080p	20	16	0	2	19	99	440	1
YUV to AV1	1080p	20	16	0	2	19	99	365	1
YUV to AVC	1080p	20	16	0	3	19	99	603	2
YUV to HEVC	1080p	20	16	0	3	19	100	300	1
YUV to AV1	1080p	20	16	0	3	19	100	240	1
YUV to AVC	1080p	20	16	1	1	19	100	440	1
YUV to HEVC	1080p	20	16	1	1	19	99	560	2
YUV to AVC	1080p	20	16	1	2	19	100	440	1
YUV to HEVC	1080p	20	16	1	2	19	99	307	1
YUV to AVC	1080p	20	16	1	3	19	99	440	1
YUV to HEVC	1080p	20	16	1	3	19	100	214	1
YUV to AVC	1080p	20	40	0	1	19	99	600	2
YUV to HEVC	1080p	20	40	0	1	19	99	700	3
YUV to AV1	1080p	20	40	0	1	19	99	620	2
YUV to AVC	1080p	20	40	0	2	19	99	600	2
YUV to HEVC	1080p	20	40	0	2	19	99	440	2
YUV to AV1	1080p	20	40	0	2	19	100	360	1
YUV to AVC	1080p	20	40	0	3	19	99	600	2
YUV to HEVC	1080p	20	40	0	3	19	99	284	1
YUV to AV1	1080p	20	40	0	3	19	100	227	1
YUV to AVC	1080p	20	40	1	1	19	100	440	1
YUV to HEVC	1080p	20	40	1	1	19	99	560	2
YUV to AVC	1080p	20	40	1	2	19	99	438	1
YUV to HEVC	1080p	20	40	1	2	19	100	300	1
YUV to AVC	1080p	20	40	1	3	19	99	439	1
YUV to HEVC	1080p	20	40	1	3	19	100	206	1



TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	23	99	619	2
YUV to HEVC	1080p	20	0	0	1	23	99	720	2
YUV to AV1	1080p	20	0	0	1	23	99	629	2
YUV to AVC	1080p	20	0	0	2	23	99	618	2
YUV to HEVC	1080p	20	0	0	2	23	99	444	1
YUV to AV1	1080p	20	0	0	2	23	100	380	1
YUV to AVC	1080p	20	0	0	3	23	99	620	2
YUV to HEVC	1080p	20	0	0	3	23	100	300	1
YUV to AV1	1080p	20	0	0	3	23	99	240	1
YUV to AVC	1080p	20	0	1	1	23	99	440	1
YUV to HEVC	1080p	20	0	1	1	23	99	574	1
YUV to AVC	1080p	20	0	1	2	23	99	440	1
YUV to HEVC	1080p	20	0	1	2	23	99	320	1
YUV to AVC	1080p	20	0	1	3	23	99	440	1
YUV to HEVC	1080p	20	0	1	3	23	100	220	1
YUV to AVC	1080p	20	4	0	1	23	99	614	2
YUV to HEVC	1080p	20	4	0	1	23	99	718	2
YUV to AV1	1080p	20	4	0	1	23	99	620	2
YUV to AVC	1080p	20	4	0	2	23	99	615	2
YUV to HEVC	1080p	20	4	0	2	23	99	440	1
YUV to AV1	1080p	20	4	0	2	23	100	365	1
YUV to AVC	1080p	20	4	0	3	23	99	615	2
YUV to HEVC	1080p	20	4	0	3	23	100	300	1
YUV to AV1	1080p	20	4	0	3	23	99	239	1
YUV to AVC	1080p	20	4	1	1	23	100	440	1
YUV to HEVC	1080p	20	4	1	1	23	99	571	2
YUV to AVC	1080p	20	4	1	2	23	99	440	1
YUV to HEVC	1080p	20	4	1	2	23	100	320	1
YUV to AVC	1080p	20	4	1	3	23	100	440	1
YUV to HEVC	1080p	20	4	1	3	23	99	220	1
YUV to AVC	1080p	20	16	0	1	23	99	604	2
YUV to HEVC	1080p	20	16	0	1	23	99	712	2
YUV to AV1	1080p	20	16	0	1	23	99	620	2
YUV to AVC	1080p	20	16	0	2	23	99	606	2
YUV to HEVC	1080p	20	16	0	2	23	99	440	1
YUV to AV1	1080p	20	16	0	2	23	99	362	1
YUV to AVC	1080p	20	16	0	3	23	99	606	2
YUV to HEVC	1080p	20	16	0	3	23	99	300	1
YUV to AV1	1080p	20	16	0	3	23	100	238	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	16	1	1	23	100	440	1
YUV to HEVC	1080p	20	16	1	1	23	99	564	2
YUV to AVC	1080p	20	16	1	2	23	99	440	1
YUV to HEVC	1080p	20	16	1	2	23	99	313	1
YUV to AVC	1080p	20	16	1	3	23	99	440	1
YUV to HEVC	1080p	20	16	1	3	23	99	217	1
YUV to AVC	1080p	20	40	0	1	23	99	600	2
YUV to HEVC	1080p	20	40	0	1	23	99	700	2
YUV to AV1	1080p	20	40	0	1	23	99	619	2
YUV to AVC	1080p	20	40	0	2	23	99	600	2
YUV to HEVC	1080p	20	40	0	2	23	99	440	1
YUV to AV1	1080p	20	40	0	2	23	99	360	1
YUV to AVC	1080p	20	40	0	3	23	99	600	2
YUV to HEVC	1080p	20	40	0	3	23	100	283	1
YUV to AV1	1080p	20	40	0	3	23	99	229	1
YUV to AVC	1080p	20	40	1	1	23	100	440	2
YUV to HEVC	1080p	20	40	1	1	23	99	560	2
YUV to AVC	1080p	20	40	1	2	23	99	439	1
YUV to HEVC	1080p	20	40	1	2	23	100	300	1
YUV to AVC	1080p	20	40	1	3	23	99	440	1
YUV to HEVC	1080p	20	40	1	3	23	99	201	1
YUV to AVC	1080p	20	0	0	1	27	99	619	2
YUV to HEVC	1080p	20	0	0	1	27	99	720	2
YUV to AV1	1080p	20	0	0	1	27	99	626	2
YUV to AVC	1080p	20	0	0	2	27	99	619	2
YUV to HEVC	1080p	20	0	0	2	27	99	443	1
YUV to AV1	1080p	20	0	0	2	27	100	380	1
YUV to AVC	1080p	20	0	0	3	27	99	619	2
YUV to HEVC	1080p	20	0	0	3	27	99	300	1
YUV to AV1	1080p	20	0	0	3	27	100	240	1
YUV to AVC	1080p	20	0	1	1	27	99	440	1
YUV to HEVC	1080p	20	0	1	1	27	99	574	2
YUV to AVC	1080p	20	0	1	2	27	99	440	1
YUV to HEVC	1080p	20	0	1	2	27	99	320	1
YUV to AVC	1080p	20	0	1	3	27	99	440	1
YUV to HEVC	1080p	20	0	1	3	27	99	220	1
YUV to AVC	1080p	20	4	0	1	27	99	613	2
YUV to HEVC	1080p	20	4	0	1	27	99	717	2
YUV to AV1	1080p	20	4	0	1	27	99	620	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	2	27	99	617	2
YUV to HEVC	1080p	20	4	0	2	27	99	444	1
YUV to AV1	1080p	20	4	0	2	27	99	367	1
YUV to AVC	1080p	20	4	0	3	27	99	616	2
YUV to HEVC	1080p	20	4	0	3	27	100	300	1
YUV to AV1	1080p	20	4	0	3	27	99	239	1
YUV to AVC	1080p	20	4	1	1	27	99	440	1
YUV to HEVC	1080p	20	4	1	1	27	99	569	2
YUV to AVC	1080p	20	4	1	2	27	99	440	1
YUV to HEVC	1080p	20	4	1	2	27	100	320	1
YUV to AVC	1080p	20	4	1	3	27	99	440	1
YUV to HEVC	1080p	20	4	1	3	27	100	220	1
YUV to AVC	1080p	20	16	0	1	27	99	604	2
YUV to HEVC	1080p	20	16	0	1	27	99	712	2
YUV to AV1	1080p	20	16	0	1	27	99	620	2
YUV to AVC	1080p	20	16	0	2	27	99	606	2
YUV to HEVC	1080p	20	16	0	2	27	99	440	1
YUV to AV1	1080p	20	16	0	2	27	99	364	1
YUV to AVC	1080p	20	16	0	3	27	99	603	2
YUV to HEVC	1080p	20	16	0	3	27	99	298	1
YUV to AV1	1080p	20	16	0	3	27	100	234	1
YUV to AVC	1080p	20	16	1	1	27	99	440	1
YUV to HEVC	1080p	20	16	1	1	27	99	560	2
YUV to AVC	1080p	20	16	1	2	27	100	440	1
YUV to HEVC	1080p	20	16	1	2	27	99	308	1
YUV to AVC	1080p	20	16	1	3	27	100	440	1
YUV to HEVC	1080p	20	16	1	3	27	99	219	1
YUV to AVC	1080p	20	40	0	1	27	99	600	2
YUV to HEVC	1080p	20	40	0	1	27	99	700	2
YUV to AV1	1080p	20	40	0	1	27	99	619	2
YUV to AVC	1080p	20	40	0	2	27	99	600	2
YUV to HEVC	1080p	20	40	0	2	27	99	440	1
YUV to AV1	1080p	20	40	0	2	27	99	360	1
YUV to AVC	1080p	20	40	0	3	27	99	600	2
YUV to HEVC	1080p	20	40	0	3	27	99	285	1
YUV to AV1	1080p	20	40	0	3	27	99	223	1
YUV to AVC	1080p	20	40	1	1	27	100	440	1
YUV to HEVC	1080p	20	40	1	1	27	99	560	2
YUV to AVC	1080p	20	40	1	2	27	99	440	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	40	1	2	27	99	300	1
YUV to AVC	1080p	20	40	1	3	27	99	438	1
YUV to HEVC	1080p	20	40	1	3	27	100	204	1

## 22. T1U – Inplace Overlay

### 22.1 Transcoding

#### 22.1.1 Description

A bitstream is read from an input file on ramdisk and then fed into the hardware decoder through PCIe. The bitstream is decoded by the hardware decoder. The decoded YUV frame is kept on the device.

An RGBA image is also uploaded to the device and overlayed onto the video stream via the 2D Engine. The overlayed YUV frames are encoded with the hardware encoder. The encoded bitstream is then read out through PCIe and written into an output file.

#### 22.1.2 Command line

```
ffmpeg -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params "out=hw" -f
concat -safe 0 -i /media/ramdisk/input.list -f rawvideo -s:v 128x128 -
pix_fmt rgba -i /media/ramdisk/img.rgb -filter_complex
"[1:v]format=rgba,ni_quadra_hwupload=0[a];[0:v][a]ni_quadra_overlay=0:0
:alpha=1:inplace=1[b]" -c:a copy -map "[b]" -c:v <enc>_ni_quadra_enc -
enc 0 -xcoder-params "RcEnable=1:bitrate=2000000" -f null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

Input Video: 1080p

Input Image: 128x128

## 22.2 Inplace Overlay Performance Results

TYPE	JOB5	FPS	CPU	DEC_LOAD	ENC_LOAD	SCALER_LOAD
AVC to AVC	1	254	8	14	20	4
AVC to HEVC	1	270	8	15	20	4
AVC to AV1	1	231	9	13	21	4
HEVC to AVC	1	254	11	14	20	4
HEVC to HEVC	1	270	13	15	20	4
HEVC to AV1	1	231	11	13	20	4
VP9 to AVC	1	253	8	19	20	4
VP9 to HEVC	1	269	9	20	21	4
VP9 to AV1	1	231	9	17	20	4
AVC to AVC	16	960	2	72	94	24
AVC to HEVC	16	1072	2	77	92	27
AVC to AV1	16	992	2	70	93	24
HEVC to AVC	16	992	3	73	94	25
HEVC to HEVC	16	1088	3	77	93	27
HEVC to AV1	16	1008	3	68	94	24
VP9 to AVC	16	975	2	86	95	24
VP9 to HEVC	16	1054	2	91	89	25
VP9 to AV1	16	1008	2	84	92	24
AVC to AVC	32	864	1	73	92	23
AVC to HEVC	32	991	1	81	92	26
AVC to AV1	32	931	1	76	94	25
HEVC to AVC	32	906	1	72	92	24
HEVC to HEVC	32	1024	1	79	92	27
HEVC to AV1	32	960	1	72	94	25
VP9 to AVC	32	922	1	86	92	23
VP9 to HEVC	32	1020	1	93	92	26
VP9 to AV1	32	992	1	85	92	24

## 23. A Note on T1M Persistent Configurations

T1M currently supports two configurations:

- Config 'E' or "Encoding Only" mode
- config 'F' or "Full Feature" mode

The feature set of each is summarized in the table below:

	Encoding	Decoding	AI	Capped Bitrate (CRF)	2D Engine	PPU	8K Support
Persistent Config 'E'	√	×	×	√	√	×	×
Persistent Config 'F'	√	√	×	√	√	√	×

## 24. T1M (Persistent config “F”) – FFmpeg Throughput

### 24.1 Decoding

#### 24.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe.

Bitstream is decoded by hardware decoder.

Decoded YUV frame is read out through PCIe and written into an output file.

#### 24.1.2 Command Line

```
ffmpeg -nostdin -f concat -safe 0 -c:v <dec>_ni_quadra_dec -dec 0 -  
xcoder-params multicoreJointMode=<*> -i /media/ramdisk/input.list -f  
null /dev/null -
```

<dec> is the decoder codec. eg h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

### 24.2 Encoding

#### 24.2.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 24.2.2 Command Line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f null  
/dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

### 24.3 Transcoding

#### 24.3.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder.



Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

#### 24.3.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -c:v <dec>_ni_quadra_dec -dec 0 -  
xcoder-params out=hw:sempianar0=1:multicoreJointMode=<*> -i  
/media/ramdisk/input.list -c:v <enc>_ni_quadra_enc -enc 0 -xcoder-  
params intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f  
null /dev/null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

## 24.4 FFmpeg Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	4k	1	0	8	1	45	0	159	35
HEVC to YUV	4k	1	0	8	1	48	0	164	46
VP9 to YUV	4k	1	0	8	1	24	0	83	26
AVC to YUV	4k	5	0	8	0	85	0	250	18
HEVC to YUV	4k	5	0	8	0	91	0	269	15
VP9 to YUV	4k	5	0	8	0	93	0	266	11
YUV to AVC	4k	1	0	8	1	0	92	184	56
YUV to HEVC	4k	1	0	8	1	0	87	192	62
YUV to AV1	4k	1	0	8	1	0	94	179	64
YUV to AVC	4k	3	0	8	0	0	73	153	37
YUV to HEVC	4k	3	0	8	0	0	72	162	38
YUV to AV1	4k	3	0	8	0	0	73	140	37
YUV to AVC	4k	5	0	8	0	0	98	191	27
YUV to HEVC	4k	5	0	8	0	0	97	215	31
YUV to AV1	4k	5	0	8	0	0	98	185	26
AVC to AVC	4k	1	1	8	1	75	96	127	32
AVC to HEVC	4k	1	1	8	1	77	92	159	43
AVC to AV1	4k	1	1	8	1	76	96	156	44
HEVC to AVC	4k	1	1	8	1	51	92	126	42
HEVC to HEVC	4k	1	1	8	1	56	96	158	38
HEVC to AV1	4k	1	1	8	1	46	87	155	47
VP9 to AVC	4k	1	1	8	1	24	39	83	22
VP9 to HEVC	4k	1	1	8	1	24	37	83	21
VP9 to AV1	4k	1	1	8	1	24	44	83	26
AVC to AVC	4k	3	1	8	0	49	72	123	30
AVC to HEVC	4k	3	1	8	0	48	72	153	35
AVC to AV1	4k	3	1	8	0	40	72	137	27
HEVC to AVC	4k	3	1	8	0	41	72	122	28
HEVC to HEVC	4k	3	1	8	0	41	73	126	26
HEVC to AV1	4k	3	1	8	0	34	73	135	28
VP9 to AVC	4k	3	1	8	0	44	72	123	28
VP9 to HEVC	4k	3	1	8	0	49	72	147	29
VP9 to AV1	4k	3	1	8	0	40	72	135	22
AVC to AVC	4k	5	1	8	0	65	97	120	16
AVC to HEVC	4k	5	1	8	0	71	97	158	24
AVC to AV1	4k	5	1	8	0	65	97	156	20
HEVC to AVC	4k	5	1	8	0	55	97	122	22
HEVC to HEVC	4k	5	1	8	0	60	98	160	23
HEVC to AV1	4k	5	1	8	0	53	96	162	20

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AVC	4k	5	1	8	0	60	97	122	21
VP9 to HEVC	4k	5	1	8	0	66	98	160	24
VP9 to AV1	4k	5	1	8	0	62	98	159	26
AVC to YUV	4k	1	0	10	1	37	0	102	34
HEVC to YUV	4k	1	0	10	1	36	0	101	32
VP9 to YUV	4k	1	0	10	1	24	0	83	28
AVC to YUV	4k	16	0	10	0	50	0	135	9
HEVC to YUV	4k	16	0	10	0	63	0	135	8
VP9 to YUV	4k	16	0	10	0	65	0	134	9
YUV to AVC	4k	1	0	10	1	0	52	107	48
YUV to HEVC	4k	1	0	10	1	0	48	108	52
YUV to AV1	4k	1	0	10	1	0	55	106	59
YUV to AVC	4k	3	0	10	0	0	62	117	54
YUV to HEVC	4k	3	0	10	0	0	52	118	63
YUV to AV1	4k	3	0	10	0	0	61	117	57
AVC to YUV	1080p	1	0	8	1	35	0	434	53
HEVC to YUV	1080p	1	0	8	1	38	0	417	79
VP9 to YUV	1080p	1	0	8	1	23	0	211	29
AVC to YUV	1080p	18	0	8	0	92	0	961	9
HEVC to YUV	1080p	18	0	8	0	93	0	1005	11
VP9 to YUV	1080p	18	0	8	0	98	0	879	8
YUV to AVC	1080p	1	0	8	1	0	53	453	77
YUV to HEVC	1080p	1	0	8	1	0	53	449	75
YUV to AV1	1080p	1	0	8	1	0	57	400	55
YUV to AVC	1080p	20	0	8	0	0	99	820	12
YUV to HEVC	1080p	20	0	8	0	0	99	862	13
YUV to AV1	1080p	20	0	8	0	0	99	743	13
AVC to AVC	1080p	1	1	8	1	62	72	544	75
AVC to HEVC	1080p	1	1	8	1	62	71	592	80
AVC to AV1	1080p	1	1	8	1	51	75	548	78
HEVC to AVC	1080p	1	1	8	1	51	74	524	91
HEVC to HEVC	1080p	1	1	8	1	57	74	582	79
HEVC to AV1	1080p	1	1	8	1	49	74	527	78
VP9 to AVC	1080p	1	1	8	1	23	25	209	34
VP9 to HEVC	1080p	1	1	8	1	23	24	210	34
VP9 to AV1	1080p	1	1	8	1	23	29	210	40
AVC to AVC	1080p	20	1	8	0	73	99	570	5
AVC to HEVC	1080p	20	1	8	0	81	99	660	11
AVC to AV1	1080p	20	1	8	0	74	99	641	8
HEVC to AVC	1080p	20	1	8	0	70	99	600	10

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to HEVC	1080p	20	1	8	0	71	99	684	10
HEVC to AV1	1080p	20	1	8	0	67	99	660	9
VP9 to AVC	1080p	20	1	8	0	77	99	580	9
VP9 to HEVC	1080p	20	1	8	0	86	99	661	7
VP9 to AV1	1080p	20	1	8	0	83	99	654	10
AVC to YUV	1080p	1	0	10	1	23	0	316	42
HEVC to YUV	1080p	1	0	10	1	20	0	312	40
VP9 to YUV	1080p	1	0	10	1	22	0	299	36
AVC to YUV	1080p	18	0	10	0	56	0	542	3
HEVC to YUV	1080p	18	0	10	0	53	0	541	4
VP9 to YUV	1080p	18	0	10	0	56	0	540	4
YUV to AVC	1080p	1	0	10	1	0	37	306	64
YUV to HEVC	1080p	1	0	10	1	0	34	306	76
YUV to AV1	1080p	1	0	10	1	0	36	283	68
YUV to AVC	1080p	20	0	10	0	0	57	462	13
YUV to HEVC	1080p	20	0	10	0	0	53	462	13
YUV to AV1	1080p	20	0	10	0	0	59	460	12
AVC to YUV	720p	1	0	8	1	27	0	650	56
HEVC to YUV	720p	1	0	8	1	26	0	633	64
VP9 to YUV	720p	1	0	8	1	21	0	372	39
AVC to YUV	720p	40	0	8	0	100	0	1900	6
HEVC to YUV	720p	40	0	8	0	92	0	1913	6
VP9 to YUV	720p	40	0	8	0	96	0	1612	4
YUV to AVC	720p	1	0	8	1	0	28	544	57
YUV to HEVC	720p	1	0	8	1	0	28	550	72
YUV to AV1	720p	1	0	8	1	0	31	513	45
YUV to AVC	720p	40	0	8	0	0	89	1630	8
YUV to HEVC	720p	40	0	8	0	0	90	1688	8
YUV to AV1	720p	40	0	8	0	0	91	1481	7
AVC to AVC	720p	1	1	8	1	38	46	879	87
AVC to HEVC	720p	1	1	8	1	36	45	886	87
AVC to AV1	720p	1	1	8	1	31	49	776	87
HEVC to AVC	720p	1	1	8	1	40	46	831	67
HEVC to HEVC	720p	1	1	8	1	40	45	856	71
HEVC to AV1	720p	1	1	8	1	35	49	751	87
VP9 to AVC	720p	1	1	8	1	21	20	370	37
VP9 to HEVC	720p	1	1	8	1	21	19	368	48
VP9 to AV1	720p	1	1	8	1	21	23	370	40
AVC to AVC	720p	40	1	8	0	93	100	1291	3
AVC to HEVC	720p	40	1	8	0	95	100	1417	2

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	720p	40	1	8	0	76	100	1295	4
HEVC to AVC	720p	40	1	8	0	85	100	1307	5
HEVC to HEVC	720p	40	1	8	0	86	100	1436	4
HEVC to AV1	720p	40	1	8	0	68	100	1303	3
VP9 to AVC	720p	40	1	8	0	100	94	1257	4
VP9 to HEVC	720p	40	1	8	0	100	90	1367	4
VP9 to AV1	720p	40	1	8	0	91	100	1295	2

## 25. T1M (Persistent config “F”) – Libxcoder Throughput

### 25.1 Decoding

#### 25.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe.

Bitstream is decoded by hardware decoder.

Decoded YUV frame is read out through PCIe and written into an output file.

#### 25.1.2 Command Line

```
./ni_xcoder_decode -c 0 -r 1000 -i /media/ramdisk/input.<ext> -m  
<test_type> -o /dev/null -d multicoreJointMode=<*>
```

<test\_type> = test codecs. ie. a (avc), h (hevc), etc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

Note: Libxcoder decoding tests were run without multi-threading (but with multicoreJointMode enabled where noted)

### 25.2 Encoding

#### 25.2.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 25.2.2 Command Line

```
./ni_xcoder_encode -c 0 -s <resolution> -r 1000 -i  
/media/ramdisk/input.yuv -m <test_type> -o /dev/null -e  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*>
```

<test\_type> = test codecs. ie. a (avc), h (hevc), etc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

Note: Libxcoder encoding tests were run without multi-threading (but with multicoreJointMode enabled where noted)

## 25.3 Transcoding

### 25.3.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder.

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

### 25.3.2 Command line

```
./ni_xcoder_multithread_transcode -c 0 -r 1000 -i  
/media/ramdisk/input.<ext> -m <dec_test_type> -n <enc_test_type> -o  
/dev/null -e  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*> -d out=hw:semiplanar0=1:multicoreJointMode=1
```

<dec\_test\_type> = decoding test codecs. ie. a (avc), h (hevc), etc

<enc\_test\_type> = encoding test codecs. ie. a (avc), h (hevc), etc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

## 25.4 Libxcode Throughput Performance Results

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	4k	1	0	8	1	47	0	155	31
HEVC to YUV	4k	1	0	8	1	44	0	158	26
VP9 to YUV	4k	1	0	8	1	24	0	82	16
AVC to YUV	4k	5	0	8	0	84	0	250	11
HEVC to YUV	4k	5	0	8	0	84	0	262	9
VP9 to YUV	4k	5	0	8	0	90	0	262	8
YUV to AVC	4k	1	0	8	1	0	58	123	48
YUV to HEVC	4k	1	0	8	1	0	56	123	48
YUV to AV1	4k	1	0	8	1	0	58	112	50
YUV to AVC	4k	3	0	8	0	0	73	150	34
YUV to HEVC	4k	3	0	8	0	0	73	161	32
YUV to AV1	4k	3	0	8	0	0	73	140	30
YUV to AVC	4k	5	0	8	0	0	98	186	27
YUV to HEVC	4k	5	0	8	0	0	98	216	26
YUV to AV1	4k	5	0	8	0	0	97	187	26
AVC to AVC	4k	1	1	8	0	75	96	128	52
AVC to HEVC	4k	1	1	8	0	83	96	163	57
AVC to AV1	4k	1	1	8	0	76	96	159	55
HEVC to AVC	4k	1	1	8	0	54	97	128	48
HEVC to HEVC	4k	1	1	8	0	58	97	159	50
HEVC to AV1	4k	1	1	8	0	54	97	152	51
VP9 to AVC	4k	1	1	8	0	24	39	82	38
VP9 to HEVC	4k	1	1	8	0	24	38	82	42
VP9 to AV1	4k	1	1	8	0	24	44	82	44
AVC to AVC	4k	3	1	8	0	48	71	122	31
AVC to HEVC	4k	3	1	8	0	49	72	149	25
AVC to AV1	4k	3	1	8	0	39	72	137	43
HEVC to AVC	4k	3	1	8	0	41	72	120	28
HEVC to HEVC	4k	3	1	8	0	40	72	144	31
HEVC to AV1	4k	3	1	8	0	35	72	133	39
VP9 to AVC	4k	3	1	8	0	46	72	122	34
VP9 to HEVC	4k	3	1	8	0	47	73	148	35
VP9 to AV1	4k	3	1	8	0	39	72	135	32
AVC to AVC	4k	5	1	8	0	65	96	122	13
AVC to HEVC	4k	5	1	8	0	69	97	159	11
AVC to AV1	4k	5	1	8	0	66	95	159	23
HEVC to AVC	4k	5	1	8	0	54	95	123	12
HEVC to HEVC	4k	5	1	8	0	59	96	159	14



TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to AV1	4k	5	1	8	0	52	96	157	11
VP9 to AVC	4k	5	1	8	0	59	98	123	12
VP9 to HEVC	4k	5	1	8	0	67	97	160	11
VP9 to AV1	4k	5	1	8	0	64	97	160	11
AVC to YUV	4k	1	0	10	1	40	0	95	23
HEVC to YUV	4k	1	0	10	1	34	0	92	22
VP9 to YUV	4k	1	0	10	1	24	0	82	21
AVC to YUV	4k	5	0	10	0	59	0	133	7
HEVC to YUV	4k	5	0	10	0	60	0	135	7
VP9 to YUV	4k	5	0	10	0	70	0	134	6
YUV to AVC	4k	1	0	10	1	0	32	66	50
YUV to HEVC	4k	1	0	10	1	0	29	67	48
YUV to AV1	4k	1	0	10	1	0	33	64	48
YUV to AVC	4k	3	0	10	0	0	47	89	39
YUV to HEVC	4k	3	0	10	0	0	37	89	38
YUV to AV1	4k	3	0	10	0	0	43	82	37
AVC to YUV	1080p	1	0	8	1	32	0	392	40
HEVC to YUV	1080p	1	0	8	1	37	0	412	36
VP9 to YUV	1080p	1	0	8	1	23	0	211	18
AVC to YUV	1080p	20	0	8	0	88	0	931	6
HEVC to YUV	1080p	20	0	8	0	90	0	968	6
VP9 to YUV	1080p	20	0	8	0	97	0	889	3
YUV to AVC	1080p	1	0	8	1	0	44	348	44
YUV to HEVC	1080p	1	0	8	1	0	38	340	44
YUV to AV1	1080p	1	0	8	1	0	38	295	42
YUV to AVC	1080p	20	0	8	0	0	99	829	10
YUV to HEVC	1080p	20	0	8	0	0	99	882	10
YUV to AV1	1080p	20	0	8	0	0	99	764	10
AVC to AVC	1080p	1	1	8	0	69	80	591	60
AVC to HEVC	1080p	1	1	8	0	66	78	638	70
AVC to AV1	1080p	1	1	8	0	58	84	598	68
HEVC to AVC	1080p	1	1	8	0	60	86	591	68
HEVC to HEVC	1080p	1	1	8	0	60	77	648	65
HEVC to AV1	1080p	1	1	8	0	57	85	597	68
VP9 to AVC	1080p	1	1	8	0	23	25	211	46
VP9 to HEVC	1080p	1	1	8	0	23	24	211	41
VP9 to AV1	1080p	1	1	8	0	23	28	211	39
AVC to AVC	1080p	20	1	8	0	73	100	585	6
AVC to HEVC	1080p	20	1	8	0	79	99	667	7

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	1080p	20	1	8	0	73	99	651	7
HEVC to AVC	1080p	20	1	8	0	67	99	611	7
HEVC to HEVC	1080p	20	1	8	0	70	99	697	6
HEVC to AV1	1080p	20	1	8	0	65	100	668	7
VP9 to AVC	1080p	20	1	8	0	77	100	591	6
VP9 to HEVC	1080p	20	1	8	0	86	99	678	5
VP9 to AV1	1080p	20	1	8	0	82	99	662	5
AVC to YUV	1080p	1	0	10	1	23	0	318	27
HEVC to YUV	1080p	1	0	10	1	21	0	312	26
VP9 to YUV	1080p	1	0	10	1	21	0	296	25
AVC to YUV	1080p	20	0	10	0	51	0	544	2
HEVC to YUV	1080p	20	0	10	0	61	0	537	2
VP9 to YUV	1080p	20	0	10	0	100	0	522	2
YUV to AVC	1080p	1	0	10	1	0	24	210	49
YUV to HEVC	1080p	1	0	10	1	0	23	211	49
YUV to AV1	1080p	1	0	10	1	0	24	188	47
YUV to AVC	1080p	20	0	10	0	0	56	470	12
YUV to HEVC	1080p	20	0	10	0	0	52	472	12
YUV to AV1	1080p	20	0	10	0	0	60	450	12
AVC to YUV	720p	1	0	8	1	23	0	576	38
HEVC to YUV	720p	1	0	8	1	27	0	635	36
VP9 to YUV	720p	1	0	8	1	22	0	377	19
AVC to YUV	720p	40	0	8	0	98	0	1903	4
HEVC to YUV	720p	40	0	8	0	78	0	1757	2
VP9 to YUV	720p	40	0	8	0	92	0	1555	1
YUV to AVC	720p	1	0	8	1	0	27	522	33
YUV to HEVC	720p	1	0	8	1	0	28	532	34
YUV to AV1	720p	1	0	8	1	0	25	423	43
YUV to AVC	720p	40	0	8	0	0	93	1767	5
YUV to HEVC	720p	40	0	8	0	0	95	1824	6
YUV to AV1	720p	40	0	8	0	0	95	1587	5
AVC to AVC	720p	1	1	8	0	51	58	911	56
AVC to HEVC	720p	1	1	8	0	50	59	946	54
AVC to AV1	720p	1	1	8	0	38	57	866	56
HEVC to AVC	720p	1	1	8	0	53	57	876	60
HEVC to HEVC	720p	1	1	8	0	50	53	884	61
HEVC to AV1	720p	1	1	8	0	44	59	861	46
VP9 to AVC	720p	1	1	8	0	21	20	376	46
VP9 to HEVC	720p	1	1	8	0	21	20	376	43

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	720p	1	1	8	0	21	24	377	48
AVC to AVC	720p	40	1	8	0	100	100	1349	0
AVC to HEVC	720p	40	1	8	0	100	100	1481	1
AVC to AV1	720p	40	1	8	0	97	100	1341	1
HEVC to AVC	720p	40	1	8	0	100	100	1354	0
HEVC to HEVC	720p	40	1	8	0	100	100	1484	0
HEVC to AV1	720p	40	1	8	0	94	100	1342	1
VP9 to AVC	720p	40	1	8	0	100	98	1298	1
VP9 to HEVC	720p	40	1	8	0	99	96	1407	1
VP9 to AV1	720p	40	1	8	0	100	100	1347	0

## 26. T1M (Persistent config “F”) – FFmpeg Latency

### 26.1 Encoding

#### 26.1.1 Description

Libxcodec is compiled and installed with parameter `--with-latency-display`

```
$ bash build.sh --with-latency-display
```

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

For each frame, the encoder latency (eLat) value is provided in the output log.

All eLat values are parsed from the output log and the last 50% of frame data before killing ffmpeg instances is used to calculate the Average, Min, Max, and Variance.

The first 50% of frame data are ignored to reach stability while launching multiple jobs.

#### 26.1.2 Command Line

```
ffmpeg -re -loglevel info -f rawvideo -pix_fmt yuv420p -stream_loop  
1000 -s:v <resolution> -i /media/ramdisk/input.yuv -c:v  
<enc>_ni_quadra_enc -enc 0 -xcodec-params gopPresetIdx=9:lowDelay=1 -f  
null -
```

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<resolution> is resolution of input

## 26.2 FFmpeg Latency Performance Results

TYPE	RESOLUTION	JOBS	ELAT_AVG (ms)	ELAT_MAX (ms)	ELAT_MIN (ms)	ELAT_VAR (ms)
YUV to AVC	4k	1	24.73	27.98	24.09	0.16
YUV to HEVC	4k	1	26.71	29.91	25.19	0.24
YUV to AV1	4k	1	34.83	40.81	26.32	1.61
YUV to AVC	4k	3	25.08	28.66	24.13	0.18
YUV to HEVC	4k	3	26.91	33.68	25.04	0.37
YUV to AV1	4k	3	35.02	40.9	26.2	1.57
YUV to AVC	4k	5	26.07	30.81	24.23	1.95
YUV to HEVC	4k	5	33.79	57.5	25.3	68.63
YUV to AV1	4k	5	43.66	76.73	26.38	131.84
YUV to AVC	1080p	1	7.63	9.07	7.08	0.06
YUV to HEVC	1080p	1	8.11	9.02	7.59	0.04
YUV to AV1	1080p	1	10.7	12.01	7.9	0.14
YUV to AVC	1080p	20	10.36	17.57	7.35	3.12
YUV to HEVC	1080p	20	10.46	18.73	8.01	3.13
YUV to AV1	1080p	20	40	51.04	30.49	14.51
YUV to AVC	720p	1	4.43	5.61	3.72	0.06
YUV to HEVC	720p	1	4.58	5.18	4.13	0.03
YUV to AV1	720p	1	5.91	6.58	4.76	0.05
YUV to AVC	720p	40	6.1	11.04	4.01	1.06
YUV to HEVC	720p	40	6.41	12.06	4.3	1.37
YUV to AV1	720p	40	38.88	45.71	32.71	2.27

## 27. T1M (Persistent config “F”) – Decoder PPU Scaling

### 27.1 Decoding

#### 27.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe.

Bitstream is decoded by hardware decoder and scaled to 224x224 with decoder post processing unit.

Decoded YUV is kept on device.

The YUV frame is converted to RGBA format with 2D Engine.

The RGBA frame is read out through PCIe and written into an output file.

#### 27.1.2 Command Line

```
ffmpeg -vsync 0 -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params  
out=hw:scale0=224x224:multicoreJointMode=<resolution=8k?1:0> -f concat  
-safe 0 -i /media/ramdisk/input.list -vf  
ni_quadra_scale=iw:ih:format=rgba,hwdownload,format=rgba -f null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

### 27.2 Decoder PPU Scaling Performance Results

TYPE	RESOLUTION	JOBS	DEC_LOAD	SCALER_LOAD	FPS	CPU
AVC to RGBA	4k	1	22	0	84	23
AVC to RGBA	4k	8	93	1	341	16
HEVC to RGBA	4k	1	22	0	100	27
HEVC to RGBA	4k	8	94	2	410	20
VP9 to RGBA	4k	1	22	0	83	20
VP9 to RGBA	4k	8	93	2	336	16
AVC to RGBA	1080p	20	93	9	1193	5
HEVC to RGBA	1080p	20	94	9	1172	8
VP9 to RGBA	1080p	20	93	6	897	4
AVC to RGBA	720p	50	94	18	2213	3
HEVC to RGBA	720p	50	87	19	2310	4
VP9 to RGBA	720p	32	94	18	1672	6

## 28. T1M (Persistent config “F”) – Streaming Ladder Generation

### 28.1 Transcoding

#### 28.1.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder split and scaled to smaller resolutions with decoder post processing unit or 2D Engine.

Decoded YUV frame is kept on device.

The YUV frames are encoded with hardware encoder.

The encoded bitstream is read out through PCIe and written into an output file.

#### 28.1.2 Command line

```
ffmpeg -vsync 0 -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params  
out=hw:sempianar0=1:enableOut1=1:sempianar1=1:scale1=1280x720:enableO  
ut2=1:sempianar2=1:scale2=960x540 -f concat -safe 0 -i  
/media/ramdisk/input.list -filter_complex  
'[0:v]ni_quadra_split=2:1:2[1080p][1080p_1][720p][540p][540p_1];[540p_1  
]ni_quadra_scale=640x360[360p]'-map [1080p] -xcoder-params  
RcEnable=1:bitrate=3500000 -c:v <enc>_ni_quadra_enc -enc 0 -f null - -  
map [1080p_1] -xcoder-params RcEnable=1:bitrate=1800000 -c:v  
<enc>_ni_quadra_enc -enc 0 -f null - -map [720p] -xcoder-params  
RcEnable=1:bitrate=1000000 -c:v <enc>_ni_quadra_enc -enc 0 -f null - -  
map [540p] -xcoder-params RcEnable=1:bitrate=800000 -c:v  
<enc>_ni_quadra_enc -enc 0 -f null - -map [360p] -xcoder-params  
RcEnable=1:bitrate=500000 -c:v <enc>_ni_quadra_enc -enc 0 -f null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

Input: 1080p

Output: 1080p, 1080p, 720p(PPU Scale), 540p(PPU Scale), 360p(2D Scale)

### 28.2 Streaming Ladder Generation Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	5	35	95	4	250	29
AVC to HEVC	5	32	92	3	280	31
AVC to AV1	5	26	93	2	250	30
HEVC to AVC	5	35	93	4	255	32
HEVC to HEVC	5	34	94	3	285	34
HEVC to AV1	5	26	95	2	255	33
VP9 to AVC	5	39	92	4	250	28
VP9 to HEVC	5	38	93	3	284	31
VP9 to AV1	5	32	94	2	253	29

## 29. T1M (Persistent config “F”) – RGBA Encoding

### 29.1 Encoding

#### 29.1.1 Description

RGBA frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

RGBA frame is uploaded and encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 29.1.2 Command line

```
ffmpeg -nostdin -stream_loop -1 -f rawvideo -pix_fmt rgba -s:v  
<resolution> -r 30 -i /media/ramdisk/input.rgb -vf  
"ni_quadra_hwupload=0" -c:v <enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f null  
/dev/null
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30

<resolution> == 720p, bitrate = 1500000, framerate = 30



## 29.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	39	81	57
RGBA to HEVC	4k	1	1	37	82	68
RGBA to AV1	4k	1	1	42	81	49
RGBA to AVC	4k	3	0	46	89	49
RGBA to HEVC	4k	3	0	43	90	48
RGBA to AV1	4k	3	0	46	90	51
RGBA to AVC	4k	5	0	48	90	33
RGBA to HEVC	4k	5	0	45	91	35
RGBA to AV1	4k	5	0	49	90	39
RGBA to AVC	1080p	1	1	29	245	66
RGBA to HEVC	1080p	1	1	28	244	70
RGBA to AV1	1080p	1	1	31	239	84
RGBA to AVC	1080p	10	0	44	335	23
RGBA to HEVC	1080p	10	0	40	339	25
RGBA to AV1	1080p	10	0	45	337	25
RGBA to AVC	1080p	20	0	45	340	13
RGBA to HEVC	1080p	20	0	41	340	13
RGBA to AV1	1080p	20	0	46	340	14
RGBA to AVC	720p	1	1	18	335	112
RGBA to HEVC	720p	1	1	17	334	111
RGBA to AV1	720p	1	1	20	339	107
RGBA to AVC	720p	10	0	38	693	47
RGBA to HEVC	720p	10	0	37	698	47
RGBA to AV1	720p	10	0	42	682	47
RGBA to AVC	720p	20	0	41	721	18
RGBA to HEVC	720p	20	0	40	739	25
RGBA to AV1	720p	20	0	44	720	22

## 30. T1M (Persistent config “F”) – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

### 30.1 Encoding

#### 30.1.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder with a mix of xcoder-params EnableRdoQuant, rdoLevel, and lookaheadDepth.

Encoded bitstream is read out through PCIe and written into an output file.

#### 30.1.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:lookaheadDepth=<*>:EnableRdoQuant=  
<*>:rdoLevel=<*> -f null /dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30

### 30.2 Encoding EnableRdoQuant/rdoLevel/lookaheadDepth Performance Results

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	0	0	1	48	102	44
YUV to HEVC	4k	2	0	0	1	48	108	41
YUV to AV1	4k	2	0	0	1	49	94	39
YUV to AVC	4k	2	0	0	2	49	102	41
YUV to HEVC	4k	2	0	0	2	50	54	29
YUV to AV1	4k	2	0	0	2	49	44	22
YUV to AVC	4k	2	0	0	3	49	102	42
YUV to HEVC	4k	2	0	0	3	50	32	22
YUV to AV1	4k	2	0	0	3	49	24	19
YUV to AVC	4k	2	0	1	1	49	60	29
YUV to HEVC	4k	2	0	1	1	49	76	33
YUV to AVC	4k	2	0	1	2	48	60	27
YUV to HEVC	4k	2	0	1	2	51	34	20
YUV to AVC	4k	2	0	1	3	49	60	28
YUV to HEVC	4k	2	0	1	3	49	22	18
YUV to AVC	4k	2	4	0	1	79	100	40
YUV to HEVC	4k	2	4	0	1	74	108	42
YUV to AV1	4k	2	4	0	1	73	94	46
YUV to AVC	4k	2	4	0	2	78	100	41
YUV to HEVC	4k	2	4	0	2	62	54	28
YUV to AV1	4k	2	4	0	2	62	44	26
YUV to AVC	4k	2	4	0	3	80	102	39
YUV to HEVC	4k	2	4	0	3	56	32	19
YUV to AV1	4k	2	4	0	3	57	26	18
YUV to AVC	4k	2	4	1	1	68	60	27
YUV to HEVC	4k	2	4	1	1	66	76	35
YUV to AVC	4k	2	4	1	2	69	60	28
YUV to HEVC	4k	2	4	1	2	58	34	20
YUV to AVC	4k	2	4	1	3	68	60	29
YUV to HEVC	4k	2	4	1	3	54	22	19
YUV to AVC	4k	2	16	0	1	78	100	42
YUV to HEVC	4k	2	16	0	1	74	108	43
YUV to AV1	4k	2	16	0	1	73	94	39
YUV to AVC	4k	2	16	0	2	78	100	39
YUV to HEVC	4k	2	16	0	2	63	54	26
YUV to AV1	4k	2	16	0	2	62	44	28
YUV to AVC	4k	2	16	0	3	79	100	39
YUV to HEVC	4k	2	16	0	3	58	32	17
YUV to AV1	4k	2	16	0	3	55	26	22

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	16	1	1	69	60	27
YUV to HEVC	4k	2	16	1	1	67	76	35
YUV to AVC	4k	2	16	1	2	67	60	28
YUV to HEVC	4k	2	16	1	2	56	34	21
YUV to AVC	4k	2	16	1	3	67	60	30
YUV to HEVC	4k	2	16	1	3	54	22	16
YUV to AVC	4k	2	40	0	1	78	100	39
YUV to HEVC	4k	2	40	0	1	74	107	44
YUV to AV1	4k	2	40	0	1	71	90	41
YUV to AVC	4k	2	40	0	2	79	100	40
YUV to HEVC	4k	2	40	0	2	62	54	27
YUV to AV1	4k	2	40	0	2	61	44	26
YUV to AVC	4k	2	40	0	3	79	100	40
YUV to HEVC	4k	2	40	0	3	56	32	21
YUV to AV1	4k	2	40	0	3	55	26	22
YUV to AVC	4k	2	40	1	1	66	60	29
YUV to HEVC	4k	2	40	1	1	67	76	34
YUV to AVC	4k	2	40	1	2	66	60	29
YUV to HEVC	4k	2	40	1	2	59	34	21
YUV to AVC	4k	2	40	1	3	66	60	29
YUV to HEVC	4k	2	40	1	3	54	22	16
YUV to AVC	1080p	10	0	0	1	99	815	25
YUV to HEVC	1080p	10	0	0	1	95	833	26
YUV to AV1	1080p	10	0	0	1	99	749	24
YUV to AVC	1080p	10	0	0	2	99	814	24
YUV to HEVC	1080p	10	0	0	2	100	439	19
YUV to AV1	1080p	10	0	0	2	99	357	19
YUV to AVC	1080p	10	0	0	3	98	814	25
YUV to HEVC	1080p	10	0	0	3	99	259	18
YUV to AV1	1080p	10	0	0	3	100	189	16
YUV to AVC	1080p	10	0	1	1	99	482	21
YUV to HEVC	1080p	10	0	1	1	99	616	24
YUV to AVC	1080p	10	0	1	2	100	482	20
YUV to HEVC	1080p	10	0	1	2	99	281	18
YUV to AVC	1080p	10	0	1	3	99	482	20
YUV to HEVC	1080p	10	0	1	3	99	176	17
YUV to AVC	1080p	10	4	0	1	99	450	20
YUV to HEVC	1080p	10	4	0	1	99	526	20
YUV to AV1	1080p	10	4	0	1	99	461	20
YUV to AVC	1080p	10	4	0	2	99	451	21

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	4	0	2	99	330	18
YUV to AV1	1080p	10	4	0	2	100	270	17
YUV to AVC	1080p	10	4	0	3	99	450	19
YUV to HEVC	1080p	10	4	0	3	99	218	17
YUV to AV1	1080p	10	4	0	3	99	171	15
YUV to AVC	1080p	10	4	1	1	99	325	19
YUV to HEVC	1080p	10	4	1	1	99	420	21
YUV to AVC	1080p	10	4	1	2	99	323	19
YUV to HEVC	1080p	10	4	1	2	100	230	16
YUV to AVC	1080p	10	4	1	3	100	324	18
YUV to HEVC	1080p	10	4	1	3	99	157	16
YUV to AVC	1080p	10	16	0	1	99	447	18
YUV to HEVC	1080p	10	16	0	1	99	523	20
YUV to AV1	1080p	10	16	0	1	99	460	19
YUV to AVC	1080p	10	16	0	2	99	448	19
YUV to HEVC	1080p	10	16	0	2	99	328	19
YUV to AV1	1080p	10	16	0	2	100	271	17
YUV to AVC	1080p	10	16	0	3	99	447	18
YUV to HEVC	1080p	10	16	0	3	99	216	15
YUV to AV1	1080p	10	16	0	3	99	170	16
YUV to AVC	1080p	10	16	1	1	99	320	17
YUV to HEVC	1080p	10	16	1	1	100	419	18
YUV to AVC	1080p	10	16	1	2	99	321	18
YUV to HEVC	1080p	10	16	1	2	99	230	16
YUV to AVC	1080p	10	16	1	3	99	320	18
YUV to HEVC	1080p	10	16	1	3	100	156	15
YUV to AVC	1080p	10	40	0	1	99	445	20
YUV to HEVC	1080p	10	40	0	1	99	520	21
YUV to AV1	1080p	10	40	0	1	99	455	19
YUV to AVC	1080p	10	40	0	2	99	442	20
YUV to HEVC	1080p	10	40	0	2	99	320	17
YUV to AV1	1080p	10	40	0	2	99	270	17
YUV to AVC	1080p	10	40	0	3	99	443	19
YUV to HEVC	1080p	10	40	0	3	100	212	17
YUV to AV1	1080p	10	40	0	3	100	170	16
YUV to AVC	1080p	10	40	1	1	100	320	18
YUV to HEVC	1080p	10	40	1	1	99	410	19
YUV to AVC	1080p	10	40	1	2	99	320	18
YUV to HEVC	1080p	10	40	1	2	100	230	16
YUV to AVC	1080p	10	40	1	3	99	320	17

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	1	3	99	154	15
YUV to AVC	720p	20	0	0	1	81	1535	14
YUV to HEVC	720p	20	0	0	1	84	1588	15
YUV to AV1	720p	20	0	0	1	89	1427	12
YUV to AVC	720p	20	0	0	2	84	1565	15
YUV to HEVC	720p	20	0	0	2	99	963	16
YUV to AV1	720p	20	0	0	2	99	780	14
YUV to AVC	720p	20	0	0	3	84	1568	15
YUV to HEVC	720p	20	0	0	3	99	580	10
YUV to AV1	720p	20	0	0	3	99	420	9
YUV to AVC	720p	20	0	1	1	99	1080	15
YUV to HEVC	720p	20	0	1	1	99	1320	15
YUV to AVC	720p	20	0	1	2	99	1084	12
YUV to HEVC	720p	20	0	1	2	99	620	10
YUV to AVC	720p	20	0	1	3	99	1080	12
YUV to HEVC	720p	20	0	1	3	99	400	7
YUV to AVC	720p	20	4	0	1	99	924	11
YUV to HEVC	720p	20	4	0	1	99	1063	13
YUV to AV1	720p	20	4	0	1	99	886	7
YUV to AVC	720p	20	4	0	2	99	923	15
YUV to HEVC	720p	20	4	0	2	99	683	8
YUV to AV1	720p	20	4	0	2	99	560	13
YUV to AVC	720p	20	4	0	3	99	923	14
YUV to HEVC	720p	20	4	0	3	100	463	6
YUV to AV1	720p	20	4	0	3	99	360	7
YUV to AVC	720p	20	4	1	1	99	681	15
YUV to HEVC	720p	20	4	1	1	99	860	14
YUV to AVC	720p	20	4	1	2	99	680	11
YUV to HEVC	720p	20	4	1	2	99	480	9
YUV to AVC	720p	20	4	1	3	99	680	11
YUV to HEVC	720p	20	4	1	3	99	340	7
YUV to AVC	720p	20	16	0	1	99	920	14
YUV to HEVC	720p	20	16	0	1	99	1060	14
YUV to AV1	720p	20	16	0	1	99	940	9
YUV to AVC	720p	20	16	0	2	99	919	13
YUV to HEVC	720p	20	16	0	2	99	680	8
YUV to AV1	720p	20	16	0	2	99	560	14
YUV to AVC	720p	20	16	0	3	99	920	14
YUV to HEVC	720p	20	16	0	3	99	463	6
YUV to AV1	720p	20	16	0	3	99	360	6

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	20	16	1	1	99	680	14
YUV to HEVC	720p	20	16	1	1	99	860	14
YUV to AVC	720p	20	16	1	2	99	680	9
YUV to HEVC	720p	20	16	1	2	99	480	12
YUV to AVC	720p	20	16	1	3	99	680	14
YUV to HEVC	720p	20	16	1	3	100	340	8
YUV to AVC	720p	20	40	0	1	99	913	11
YUV to HEVC	720p	20	40	0	1	99	1043	13
YUV to AV1	720p	20	40	0	1	99	914	7
YUV to AVC	720p	20	40	0	2	99	917	9
YUV to HEVC	720p	20	40	0	2	99	680	8
YUV to AV1	720p	20	40	0	2	99	560	10
YUV to AVC	720p	20	40	0	3	99	913	13
YUV to HEVC	720p	20	40	0	3	99	460	9
YUV to AV1	720p	20	40	0	3	99	360	6
YUV to AVC	720p	20	40	1	1	99	677	14
YUV to HEVC	720p	20	40	1	1	99	841	14
YUV to AVC	720p	20	40	1	2	99	676	9
YUV to HEVC	720p	20	40	1	2	99	480	10
YUV to AVC	720p	20	40	1	3	99	677	10
YUV to HEVC	720p	20	40	1	3	99	340	6

## 31. T1M (Persistent config “F”) – Capped CRF

### 31.1 Encoding with lookaheadDepth

#### 31.1.1 Description

YUV frame is read from an input file on ramdisk and fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder with a mix of xcoder-params EnableRdoQuant, rdoLevel, lookaheadDepth, CRF, bitrate, and vbvBufferSize.

Encoded bitstream is read out through PCIe and written into an output file.

#### 31.1.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:vbvBufferSize=1000:bitrate=<*>:lookaheadDepth=<*>:EnableR  
doQuant=<*>:rdoLevel=<*>:crf=<*> -f null /dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30



### 31.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	0	0	1	19	99	450	18
YUV to HEVC	1080p	10	0	0	1	19	99	525	20
YUV to AV1	1080p	10	0	0	1	19	99	460	20
YUV to AVC	1080p	10	0	0	2	19	99	450	19
YUV to HEVC	1080p	10	0	0	2	19	100	330	18
YUV to AV1	1080p	10	0	0	2	19	99	279	16
YUV to AVC	1080p	10	0	0	3	19	99	451	20
YUV to HEVC	1080p	10	0	0	3	19	99	216	16
YUV to AV1	1080p	10	0	0	3	19	99	178	15
YUV to AVC	1080p	10	0	1	1	19	99	327	19
YUV to HEVC	1080p	10	0	1	1	19	99	420	19
YUV to AVC	1080p	10	0	1	2	19	99	325	19
YUV to HEVC	1080p	10	0	1	2	19	100	230	17
YUV to AVC	1080p	10	0	1	3	19	100	327	18
YUV to HEVC	1080p	10	0	1	3	19	100	158	14
YUV to AVC	1080p	10	4	0	1	19	99	450	20
YUV to HEVC	1080p	10	4	0	1	19	99	522	19
YUV to AV1	1080p	10	4	0	1	19	99	460	18
YUV to AVC	1080p	10	4	0	2	19	99	450	18
YUV to HEVC	1080p	10	4	0	2	19	99	330	18
YUV to AV1	1080p	10	4	0	2	19	100	270	16
YUV to AVC	1080p	10	4	0	3	19	99	450	19
YUV to HEVC	1080p	10	4	0	3	19	99	217	17
YUV to AV1	1080p	10	4	0	3	19	100	171	14
YUV to AVC	1080p	10	4	1	1	19	99	325	18
YUV to HEVC	1080p	10	4	1	1	19	99	420	18
YUV to AVC	1080p	10	4	1	2	19	99	324	18
YUV to HEVC	1080p	10	4	1	2	19	100	230	17
YUV to AVC	1080p	10	4	1	3	19	99	324	17
YUV to HEVC	1080p	10	4	1	3	19	99	158	14
YUV to AVC	1080p	10	16	0	1	19	99	448	18
YUV to HEVC	1080p	10	16	0	1	19	99	520	19
YUV to AV1	1080p	10	16	0	1	19	99	460	17
YUV to AVC	1080p	10	16	0	2	19	99	448	18
YUV to HEVC	1080p	10	16	0	2	19	99	329	18
YUV to AV1	1080p	10	16	0	2	19	99	270	16
YUV to AVC	1080p	10	16	0	3	19	99	449	18
YUV to HEVC	1080p	10	16	0	3	19	99	217	16

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	3	19	99	171	15
YUV to AVC	1080p	10	16	1	1	19	100	321	17
YUV to HEVC	1080p	10	16	1	1	19	99	419	19
YUV to AVC	1080p	10	16	1	2	19	99	321	18
YUV to HEVC	1080p	10	16	1	2	19	100	230	15
YUV to AVC	1080p	10	16	1	3	19	99	321	17
YUV to HEVC	1080p	10	16	1	3	19	100	155	15
YUV to AVC	1080p	10	40	0	1	19	99	443	19
YUV to HEVC	1080p	10	40	0	1	19	99	520	19
YUV to AV1	1080p	10	40	0	1	19	99	455	18
YUV to AVC	1080p	10	40	0	2	19	99	442	19
YUV to HEVC	1080p	10	40	0	2	19	99	320	18
YUV to AV1	1080p	10	40	0	2	19	100	270	15
YUV to AVC	1080p	10	40	0	3	19	99	442	18
YUV to HEVC	1080p	10	40	0	3	19	99	214	16
YUV to AV1	1080p	10	40	0	3	19	100	170	15
YUV to AVC	1080p	10	40	1	1	19	99	320	18
YUV to HEVC	1080p	10	40	1	1	19	99	412	19
YUV to AVC	1080p	10	40	1	2	19	99	320	17
YUV to HEVC	1080p	10	40	1	2	19	99	230	16
YUV to AVC	1080p	10	40	1	3	19	99	320	18
YUV to HEVC	1080p	10	40	1	3	19	100	153	16
YUV to AVC	1080p	10	0	0	1	23	99	450	19
YUV to HEVC	1080p	10	0	0	1	23	99	526	20
YUV to AV1	1080p	10	0	0	1	23	99	460	18
YUV to AVC	1080p	10	0	0	2	23	99	450	19
YUV to HEVC	1080p	10	0	0	2	23	100	330	17
YUV to AV1	1080p	10	0	0	2	23	100	279	17
YUV to AVC	1080p	10	0	0	3	23	99	450	19
YUV to HEVC	1080p	10	0	0	3	23	99	218	16
YUV to AV1	1080p	10	0	0	3	23	99	175	16
YUV to AVC	1080p	10	0	1	1	23	100	327	17
YUV to HEVC	1080p	10	0	1	1	23	99	420	20
YUV to AVC	1080p	10	0	1	2	23	99	326	19
YUV to HEVC	1080p	10	0	1	2	23	100	230	18
YUV to AVC	1080p	10	0	1	3	23	99	328	16
YUV to HEVC	1080p	10	0	1	3	23	100	156	15
YUV to AVC	1080p	10	4	0	1	23	99	450	18
YUV to HEVC	1080p	10	4	0	1	23	99	524	19

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	4	0	1	23	99	460	17
YUV to AVC	1080p	10	4	0	2	23	99	449	20
YUV to HEVC	1080p	10	4	0	2	23	99	330	18
YUV to AV1	1080p	10	4	0	2	23	99	270	17
YUV to AVC	1080p	10	4	0	3	23	99	450	19
YUV to HEVC	1080p	10	4	0	3	23	99	219	17
YUV to AV1	1080p	10	4	0	3	23	99	171	16
YUV to AVC	1080p	10	4	1	1	23	99	324	18
YUV to HEVC	1080p	10	4	1	1	23	99	419	18
YUV to AVC	1080p	10	4	1	2	23	99	325	17
YUV to HEVC	1080p	10	4	1	2	23	100	230	16
YUV to AVC	1080p	10	4	1	3	23	100	325	16
YUV to HEVC	1080p	10	4	1	3	23	99	156	15
YUV to AVC	1080p	10	16	0	1	23	99	449	19
YUV to HEVC	1080p	10	16	0	1	23	99	523	20
YUV to AV1	1080p	10	16	0	1	23	99	460	19
YUV to AVC	1080p	10	16	0	2	23	99	447	18
YUV to HEVC	1080p	10	16	0	2	23	100	329	19
YUV to AV1	1080p	10	16	0	2	23	100	270	17
YUV to AVC	1080p	10	16	0	3	23	99	449	18
YUV to HEVC	1080p	10	16	0	3	23	99	214	17
YUV to AV1	1080p	10	16	0	3	23	99	170	15
YUV to AVC	1080p	10	16	1	1	23	99	322	17
YUV to HEVC	1080p	10	16	1	1	23	99	418	19
YUV to AVC	1080p	10	16	1	2	23	100	321	18
YUV to HEVC	1080p	10	16	1	2	23	99	230	17
YUV to AVC	1080p	10	16	1	3	23	99	322	17
YUV to HEVC	1080p	10	16	1	3	23	100	156	15
YUV to AVC	1080p	10	40	0	1	23	99	443	18
YUV to HEVC	1080p	10	40	0	1	23	99	520	19
YUV to AV1	1080p	10	40	0	1	23	99	456	18
YUV to AVC	1080p	10	40	0	2	23	99	444	19
YUV to HEVC	1080p	10	40	0	2	23	100	323	18
YUV to AV1	1080p	10	40	0	2	23	100	270	16
YUV to AVC	1080p	10	40	0	3	23	99	442	18
YUV to HEVC	1080p	10	40	0	3	23	99	212	16
YUV to AV1	1080p	10	40	0	3	23	100	170	15
YUV to AVC	1080p	10	40	1	1	23	99	320	17
YUV to HEVC	1080p	10	40	1	1	23	100	413	19

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	40	1	2	23	99	320	18
YUV to HEVC	1080p	10	40	1	2	23	99	229	16
YUV to AVC	1080p	10	40	1	3	23	99	320	17
YUV to HEVC	1080p	10	40	1	3	23	98	153	15
YUV to AVC	1080p	10	0	0	1	27	99	450	19
YUV to HEVC	1080p	10	0	0	1	27	99	524	20
YUV to AV1	1080p	10	0	0	1	27	99	460	19
YUV to AVC	1080p	10	0	0	2	27	100	450	18
YUV to HEVC	1080p	10	0	0	2	27	100	330	18
YUV to AV1	1080p	10	0	0	2	27	99	279	16
YUV to AVC	1080p	10	0	0	3	27	99	450	19
YUV to HEVC	1080p	10	0	0	3	27	100	217	16
YUV to AV1	1080p	10	0	0	3	27	100	176	15
YUV to AVC	1080p	10	0	1	1	27	99	326	16
YUV to HEVC	1080p	10	0	1	1	27	99	420	19
YUV to AVC	1080p	10	0	1	2	27	99	326	19
YUV to HEVC	1080p	10	0	1	2	27	99	230	18
YUV to AVC	1080p	10	0	1	3	27	99	327	18
YUV to HEVC	1080p	10	0	1	3	27	99	158	15
YUV to AVC	1080p	10	4	0	1	27	99	450	20
YUV to HEVC	1080p	10	4	0	1	27	99	521	20
YUV to AV1	1080p	10	4	0	1	27	99	459	18
YUV to AVC	1080p	10	4	0	2	27	99	450	19
YUV to HEVC	1080p	10	4	0	2	27	99	330	18
YUV to AV1	1080p	10	4	0	2	27	99	270	16
YUV to AVC	1080p	10	4	0	3	27	99	450	20
YUV to HEVC	1080p	10	4	0	3	27	100	216	16
YUV to AV1	1080p	10	4	0	3	27	100	172	15
YUV to AVC	1080p	10	4	1	1	27	100	325	17
YUV to HEVC	1080p	10	4	1	1	27	99	420	19
YUV to AVC	1080p	10	4	1	2	27	99	325	19
YUV to HEVC	1080p	10	4	1	2	27	99	230	17
YUV to AVC	1080p	10	4	1	3	27	100	324	17
YUV to HEVC	1080p	10	4	1	3	27	100	157	15
YUV to AVC	1080p	10	16	0	1	27	99	448	20
YUV to HEVC	1080p	10	16	0	1	27	99	522	21
YUV to AV1	1080p	10	16	0	1	27	99	460	17
YUV to AVC	1080p	10	16	0	2	27	99	449	20
YUV to HEVC	1080p	10	16	0	2	27	99	330	17

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	2	27	100	270	17
YUV to AVC	1080p	10	16	0	3	27	99	449	19
YUV to HEVC	1080p	10	16	0	3	27	100	215	16
YUV to AV1	1080p	10	16	0	3	27	99	170	14
YUV to AVC	1080p	10	16	1	1	27	99	322	17
YUV to HEVC	1080p	10	16	1	1	27	99	419	19
YUV to AVC	1080p	10	16	1	2	27	99	321	17
YUV to HEVC	1080p	10	16	1	2	27	100	230	17
YUV to AVC	1080p	10	16	1	3	27	99	322	18
YUV to HEVC	1080p	10	16	1	3	27	100	156	15
YUV to AVC	1080p	10	40	0	1	27	99	443	20
YUV to HEVC	1080p	10	40	0	1	27	99	520	21
YUV to AV1	1080p	10	40	0	1	27	100	453	18
YUV to AVC	1080p	10	40	0	2	27	99	445	19
YUV to HEVC	1080p	10	40	0	2	27	99	320	19
YUV to AV1	1080p	10	40	0	2	27	100	270	16
YUV to AVC	1080p	10	40	0	3	27	99	443	19
YUV to HEVC	1080p	10	40	0	3	27	99	213	16
YUV to AV1	1080p	10	40	0	3	27	100	170	16
YUV to AVC	1080p	10	40	1	1	27	99	321	19
YUV to HEVC	1080p	10	40	1	1	27	100	412	20
YUV to AVC	1080p	10	40	1	2	27	99	320	17
YUV to HEVC	1080p	10	40	1	2	27	99	230	15
YUV to AVC	1080p	10	40	1	3	27	99	320	18
YUV to HEVC	1080p	10	40	1	3	27	99	154	16

## 32. T1M (Persistent config “F”) – Inplace Overlay

### 32.1 Transcoding

#### 32.1.1 Description

A bitstream is read from an input file on ramdisk and then fed into the hardware decoder through PCIe. The bitstream is decoded by the hardware decoder. The decoded YUV frame is kept on the device.

An RGBA image is also uploaded to the device and overlayed onto the video stream via the 2D Engine. The overlayed YUV frames are encoded with the hardware encoder. The encoded bitstream is then read out through PCIe and written into an output file.

#### 32.1.2 Command line

```
ffmpeg -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params "out=hw" -f
concat -safe 0 -i /media/ramdisk/input.list -f rawvideo -s:v 128x128 -
pix_fmt rgba -i /media/ramdisk/img.rgb -filter_complex
"[1:v]format=rgba,ni_quadra_hwupload=0[a];[0:v][a]ni_quadra_overlay=0:0
:alpha=1:inplace=1[b]" -c:a copy -map "[b]" -c:v <enc>_ni_quadra_enc -
enc 0 -xcoder-params "RcEnable=1:bitrate=2000000" -f null -
```

<dec> is the decoder codec. ie h264\_ni\_quadra\_dec, h265\_ni\_quadra\_dec, vp9\_ni\_quadra\_dec

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

Input Video: 1080p

Input Image: 128x128

### 32.2 Inplace Overlay Performance Results

TYPE	JOB5	FPS	CPU	DEC_LOAD	ENC_LOAD	SCALER_LOAD
AVC to AVC	1	191	53	14	22	3
AVC to HEVC	1	203	59	15	21	3
AVC to AV1	1	174	55	13	21	2
HEVC to AVC	1	188	60	14	21	3
HEVC to HEVC	1	200	60	15	20	3
HEVC to AV1	1	172	59	13	21	2
VP9 to AVC	1	189	48	19	21	3
VP9 to HEVC	1	202	50	20	20	3
VP9 to AV1	1	174	47	17	20	2
AVC to AVC	8	536	26	70	92	18
AVC to HEVC	8	608	26	78	93	20
AVC to AV1	8	607	25	75	95	20
HEVC to AVC	8	560	29	67	94	19
HEVC to HEVC	8	638	29	73	95	21
HEVC to AV1	8	628	27	68	93	20
VP9 to AVC	8	538	19	74	92	17
VP9 to HEVC	8	615	26	84	95	20
VP9 to AV1	8	608	26	80	92	19
AVC to AVC	16	531	11	72	93	19
AVC to HEVC	16	608	10	77	93	20
AVC to AV1	16	600	12	75	93	20
HEVC to AVC	16	560	11	67	94	20
HEVC to HEVC	16	639	11	73	94	22
HEVC to AV1	16	624	11	70	94	21
VP9 to AVC	16	535	10	77	95	19
VP9 to HEVC	16	608	11	83	93	20
VP9 to AV1	16	608	10	81	93	20

## 33. T1M (Persistent config “E”) – FFmpeg Throughput

### 33.1 Encoding

#### 33.1.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 33.1.2 Command Line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f null  
/dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)



### 33.2 FFmpeg Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	4k	1	0	8	1	0	60	182	56
YUV to HEVC	4k	1	0	8	1	0	54	188	64
YUV to AV1	4k	1	0	8	1	0	60	181	66
YUV to AVC	4k	3	0	8	0	0	71	198	45
YUV to HEVC	4k	3	0	8	0	0	66	222	54
YUV to AV1	4k	3	0	8	0	0	71	213	51
YUV to AVC	4k	5	0	8	0	0	94	207	35
YUV to HEVC	4k	5	0	8	0	0	69	229	44
YUV to AV1	4k	5	0	8	0	0	76	226	42
YUV to AVC	4k	1	0	10	1	0	35	107	52
YUV to HEVC	4k	1	0	10	1	0	31	108	55
YUV to AV1	4k	1	0	10	1	0	35	106	54
YUV to AVC	4k	3	0	10	0	0	45	114	59
YUV to HEVC	4k	3	0	10	0	0	36	117	57
YUV to AV1	4k	3	0	10	0	0	39	116	60
YUV to AVC	1080p	1	0	8	1	0	34	451	70
YUV to HEVC	1080p	1	0	8	1	0	31	457	88
YUV to AV1	1080p	1	0	8	1	0	34	406	64
YUV to AVC	1080p	20	0	8	0	0	72	814	13
YUV to HEVC	1080p	20	0	8	0	0	64	847	14
YUV to AV1	1080p	20	0	8	0	0	70	816	14
YUV to AVC	1080p	1	0	10	1	0	23	309	64
YUV to HEVC	1080p	1	0	10	1	0	22	310	72
YUV to AV1	1080p	1	0	10	1	0	24	287	60
YUV to AVC	1080p	20	0	10	0	0	38	465	13
YUV to HEVC	1080p	20	0	10	0	0	34	463	13
YUV to AV1	1080p	20	0	10	0	0	38	460	13
YUV to AVC	720p	1	0	8	1	0	19	560	66
YUV to HEVC	720p	1	0	8	1	0	20	550	72
YUV to AV1	720p	1	0	8	1	0	24	511	56
YUV to AVC	720p	40	0	8	0	0	83	1582	8
YUV to HEVC	720p	40	0	8	0	0	81	1675	8
YUV to AV1	720p	40	0	8	0	0	84	1455	6

## 34. T1M (Persistent config “E”) – Libxcoder Throughput

### 34.1 Encoding

#### 34.1.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 34.1.2 Command Line

```
./ni_xcoder_encode -c 0 -s <resolution> -r 1000 -i  
/media/ramdisk/input.yuv -m <test_type> -o /dev/null -e  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*>
```

<test\_type> = test codecs. ie. a (avc), h (hevc), etc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 4k, bitrate = 12000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 1080p, bitrate = 3000000, framerate = 30 (8bit) / 60 (10bit)

<resolution> == 720p, bitrate = 1500000, framerate = 30 (8bit) / 60 (10bit)

Note: Libxcoder encoding tests were run without multi-threading (but with multicoreJointMode enabled where noted)

## 34.2 Libxcode Throughput Performance Results

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	4k	1	0	8	1	0	38	124	49
YUV to HEVC	4k	1	0	8	1	0	35	123	49
YUV to AV1	4k	1	0	8	1	0	36	110	51
YUV to AVC	4k	3	0	8	0	0	57	171	37
YUV to HEVC	4k	3	0	8	0	0	46	164	38
YUV to AV1	4k	3	0	8	0	0	51	156	36
YUV to AVC	4k	5	0	8	0	0	94	194	30
YUV to HEVC	4k	5	0	8	0	0	64	212	34
YUV to AV1	4k	5	0	8	0	0	65	194	33
YUV to AVC	4k	1	0	10	1	0	19	67	49
YUV to HEVC	4k	1	0	10	1	0	19	66	49
YUV to AV1	4k	1	0	10	1	0	21	65	48
YUV to AVC	4k	3	0	10	0	0	27	88	38
YUV to HEVC	4k	3	0	10	0	0	27	92	39
YUV to AV1	4k	3	0	10	0	0	27	85	37
YUV to AVC	1080p	1	0	8	1	0	26	335	47
YUV to HEVC	1080p	1	0	8	1	0	24	351	42
YUV to AV1	1080p	1	0	8	1	0	23	276	39
YUV to AVC	1080p	20	0	8	0	0	72	834	10
YUV to HEVC	1080p	20	0	8	0	0	64	867	11
YUV to AV1	1080p	20	0	8	0	0	70	824	10
YUV to AVC	1080p	1	0	10	1	0	16	213	48
YUV to HEVC	1080p	1	0	10	1	0	15	215	47
YUV to AV1	1080p	1	0	10	1	0	15	186	48
YUV to AVC	1080p	20	0	10	0	0	36	469	12
YUV to HEVC	1080p	20	0	10	0	0	34	471	12
YUV to AV1	1080p	20	0	10	0	0	37	440	12
YUV to AVC	720p	1	0	8	1	0	19	529	37
YUV to HEVC	720p	1	0	8	1	0	18	529	31
YUV to AV1	720p	1	0	8	1	0	18	421	45
YUV to AVC	720p	40	0	8	0	0	83	1716	5
YUV to HEVC	720p	40	0	8	0	0	80	1805	6
YUV to AV1	720p	40	0	8	0	0	87	1578	5

## 35. T1M (Persistent config “E”) – FFmpeg Latency

### 35.1 Encoding

#### 35.1.1 Description

Libxcodec is compiled and installed with parameter `--with-latency-display`

```
$ bash build.sh --with-latency-display
```

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

For each frame, the encoder latency (eLat) value is provided in the output log.

All eLat values are parsed from the output log and the last 50% of frame data before killing ffmpeg instances is used to calculate the Average, Min, Max, and Variance.

The first 50% of frame data are ignored to reach stability while launching multiple jobs.

#### 35.1.2 Command Line

```
ffmpeg -re -loglevel info -f rawvideo -pix_fmt yuv420p -stream_loop  
1000 -s:v <resolution> -i /media/ramdisk/input.yuv -c:v  
<enc>_ni_quadra_enc -enc 0 -xcodec-params gopPresetIdx=9:lowDelay=1 -f  
null -
```

<enc> is the encoder codec. ie h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<resolution> is resolution of input

### 35.2 FFmpeg Latency Performance Results

TYPE	RESOLUTION	JOBS	ELAT_AVG (ms)	ELAT_MAX (ms)	ELAT_MIN (ms)	ELAT_VAR (ms)
YUV to AVC	4k	1	17.86	20.05	17.17	0.1
YUV to HEVC	4k	1	19.02	21.1	18.02	0.12
YUV to AV1	4k	1	24.21	27.94	18.48	0.74
YUV to AVC	4k	3	19.1	23.86	17.18	0.82
YUV to HEVC	4k	3	19.62	24.81	17.82	0.62
YUV to AV1	4k	3	25.18	31.9	18.55	1.52
YUV to AVC	4k	5	19.52	25.43	17.38	0.78
YUV to HEVC	4k	5	19.78	25.7	18.07	0.49
YUV to AV1	4k	5	25.74	32.63	18.87	1.96
YUV to AVC	1080p	1	5.81	6.67	5.15	0.04
YUV to HEVC	1080p	1	6.17	8.01	5.47	0.19
YUV to AV1	1080p	1	7.76	8.64	6.05	0.07
YUV to AVC	1080p	20	7.47	11.2	5.31	0.75
YUV to HEVC	1080p	20	8.01	16.17	5.58	2.49
YUV to AV1	1080p	20	11.53	20.82	6.7	4.96
YUV to AVC	720p	1	3.5	4.23	2.94	0.04
YUV to HEVC	720p	1	3.55	4.09	2.98	0.04
YUV to AV1	720p	1	4.54	5.03	3.7	0.03
YUV to AVC	720p	40	4.59	8.06	2.94	0.48
YUV to HEVC	720p	40	5.39	11.61	3.18	1.43
YUV to AV1	720p	40	7.19	13.64	4.53	1.32

## 36. T1M (Persistent config “E”) – RGBA Encoding

### 36.1 Encoding

#### 36.1.1 Description

RGBA frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

RGBA frame is uploaded and encoded by hardware encoder.

Encoded bitstream is read out through PCIe and written into an output file.

#### 36.1.2 Command line

```
ffmpeg -nostdin -stream_loop -1 -f rawvideo -pix_fmt rgba -s:v  
<resolution> -r 30 -i /media/ramdisk/input.rgb -vf  
"ni_quadra_hwupload=0" -c:v <enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f null  
/dev/null
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num\_jobs> == 1, multicoreJointMode = 1

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30

<resolution> == 720p, bitrate = 1500000, framerate = 30

## 36.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	25	82	55
RGBA to HEVC	4k	1	1	24	83	71
RGBA to AV1	4k	1	1	27	82	53
RGBA to AVC	4k	3	0	37	88	51
RGBA to HEVC	4k	3	0	33	88	49
RGBA to AV1	4k	3	0	34	89	48
RGBA to AVC	4k	5	0	45	91	42
RGBA to HEVC	4k	5	0	38	90	37
RGBA to AV1	4k	5	0	37	90	36
RGBA to AVC	1080p	1	1	19	246	80
RGBA to HEVC	1080p	1	1	17	246	81
RGBA to AV1	1080p	1	1	20	241	80
RGBA to AVC	1080p	10	0	33	331	23
RGBA to HEVC	1080p	10	0	29	333	24
RGBA to AV1	1080p	10	0	30	329	24
RGBA to AVC	1080p	20	0	36	331	13
RGBA to HEVC	1080p	20	0	31	331	13
RGBA to AV1	1080p	20	0	34	329	14
RGBA to AVC	720p	1	1	13	332	108
RGBA to HEVC	720p	1	1	13	332	105
RGBA to AV1	720p	1	1	15	330	112
RGBA to AVC	720p	10	0	34	699	47
RGBA to HEVC	720p	10	0	33	710	47
RGBA to AV1	720p	10	0	39	689	47
RGBA to AVC	720p	20	0	36	721	26
RGBA to HEVC	720p	20	0	34	729	24
RGBA to AV1	720p	20	0	42	712	23

## 37. T1M (Persistent config “E”) – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

### 37.1 Encoding

#### 37.1.1 Description

YUV frame is read from an input file on ramdisk and then fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder with a mix of xcoder-params EnableRdoQuant, rdoLevel, and lookaheadDepth.

Encoded bitstream is read out through PCIe and written into an output file.

#### 37.1.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:lookaheadDepth=<*>:EnableRdoQuant=  
<*>:rdoLevel=<*> -f null /dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30



### 37.2 Encoding EnableRdoQuant/rdoLevel/lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	0	0	1	48	155	52
YUV to HEVC	4k	2	0	0	1	48	166	50
YUV to AV1	4k	2	0	0	1	48	144	51
YUV to AVC	4k	2	0	0	2	48	156	55
YUV to HEVC	4k	2	0	0	2	49	86	34
YUV to AV1	4k	2	0	0	2	49	70	32
YUV to AVC	4k	2	0	0	3	47	154	53
YUV to HEVC	4k	2	0	0	3	50	50	26
YUV to AV1	4k	2	0	0	3	48	38	21
YUV to AVC	4k	2	0	1	1	48	94	38
YUV to HEVC	4k	2	0	1	1	48	120	43
YUV to AVC	4k	2	0	1	2	49	94	38
YUV to HEVC	4k	2	0	1	2	48	54	26
YUV to AVC	4k	2	0	1	3	49	94	37
YUV to HEVC	4k	2	0	1	3	50	34	21
YUV to AVC	4k	2	4	0	1	73	130	49
YUV to HEVC	4k	2	4	0	1	73	166	51
YUV to AV1	4k	2	4	0	1	72	144	55
YUV to AVC	4k	2	4	0	2	75	134	48
YUV to HEVC	4k	2	4	0	2	61	84	34
YUV to AV1	4k	2	4	0	2	60	70	32
YUV to AVC	4k	2	4	0	3	73	132	45
YUV to HEVC	4k	2	4	0	3	56	50	29
YUV to AV1	4k	2	4	0	3	56	40	23
YUV to AVC	4k	2	4	1	1	66	92	38
YUV to HEVC	4k	2	4	1	1	67	120	47
YUV to AVC	4k	2	4	1	2	67	92	40
YUV to HEVC	4k	2	4	1	2	58	54	28
YUV to AVC	4k	2	4	1	3	67	92	38
YUV to HEVC	4k	2	4	1	3	55	34	20
YUV to AVC	4k	2	20	0	1	74	132	51
YUV to HEVC	4k	2	20	0	1	73	166	56
YUV to AV1	4k	2	20	0	1	72	144	42

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	20	0	2	73	132	49
YUV to HEVC	4k	2	20	0	2	61	84	36
YUV to AV1	4k	2	20	0	2	61	70	36
YUV to AVC	4k	2	20	0	3	73	132	48
YUV to HEVC	4k	2	20	0	3	57	50	26
YUV to AV1	4k	2	20	0	3	55	40	25
YUV to AVC	4k	2	20	1	1	66	92	39
YUV to HEVC	4k	2	20	1	1	67	120	45
YUV to AVC	4k	2	20	1	2	67	92	37
YUV to HEVC	4k	2	20	1	2	57	54	32
YUV to AVC	4k	2	20	1	3	66	92	41
YUV to HEVC	4k	2	20	1	3	54	34	23
YUV to AVC	4k	2	40	0	1	74	130	50
YUV to HEVC	4k	2	40	0	1	72	164	59
YUV to AV1	4k	2	40	0	1	72	142	51
YUV to AVC	4k	2	40	0	2	73	128	52
YUV to HEVC	4k	2	40	0	2	61	84	35
YUV to AV1	4k	2	40	0	2	60	68	33
YUV to AVC	4k	2	40	0	3	72	128	52
YUV to HEVC	4k	2	40	0	3	57	50	28
YUV to AV1	4k	2	40	0	3	56	40	28
YUV to AVC	4k	2	40	1	1	66	90	39
YUV to HEVC	4k	2	40	1	1	66	118	46
YUV to AVC	4k	2	40	1	2	66	92	37
YUV to HEVC	4k	2	40	1	2	57	54	26
YUV to AVC	4k	2	40	1	3	67	92	35
YUV to HEVC	4k	2	40	1	3	55	34	24
YUV to AVC	1080p	10	0	0	1	66	785	25
YUV to HEVC	1080p	10	0	0	1	60	816	26
YUV to AV1	1080p	10	0	0	1	65	772	25
YUV to AVC	1080p	10	0	0	2	65	783	24
YUV to HEVC	1080p	10	0	0	2	99	689	23
YUV to AV1	1080p	10	0	0	2	100	559	21
YUV to AVC	1080p	10	0	0	3	66	786	25

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	99	408	19
YUV to AV1	1080p	10	0	0	3	100	298	18
YUV to AVC	1080p	10	0	1	1	99	754	23
YUV to HEVC	1080p	10	0	1	1	86	831	26
YUV to AVC	1080p	10	0	1	2	100	755	25
YUV to HEVC	1080p	10	0	1	2	100	439	20
YUV to AVC	1080p	10	0	1	3	99	754	23
YUV to HEVC	1080p	10	0	1	3	100	277	18
YUV to AVC	1080p	10	4	0	1	99	649	20
YUV to HEVC	1080p	10	4	0	1	87	711	21
YUV to AV1	1080p	10	4	0	1	94	684	20
YUV to AVC	1080p	10	4	0	2	99	648	20
YUV to HEVC	1080p	10	4	0	2	100	512	19
YUV to AV1	1080p	10	4	0	2	99	430	20
YUV to AVC	1080p	10	4	0	3	99	649	19
YUV to HEVC	1080p	10	4	0	3	99	340	17
YUV to AV1	1080p	10	4	0	3	100	270	17
YUV to AVC	1080p	10	4	1	1	99	510	19
YUV to HEVC	1080p	10	4	1	1	100	653	20
YUV to AVC	1080p	10	4	1	2	99	510	20
YUV to HEVC	1080p	10	4	1	2	99	360	18
YUV to AVC	1080p	10	4	1	3	99	510	20
YUV to HEVC	1080p	10	4	1	3	100	245	17
YUV to AVC	1080p	10	20	0	1	99	640	20
YUV to HEVC	1080p	10	20	0	1	89	716	22
YUV to AV1	1080p	10	20	0	1	93	671	20
YUV to AVC	1080p	10	20	0	2	99	640	18
YUV to HEVC	1080p	10	20	0	2	99	510	19
YUV to AV1	1080p	10	20	0	2	100	429	19
YUV to AVC	1080p	10	20	0	3	99	641	19
YUV to HEVC	1080p	10	20	0	3	99	339	19
YUV to AV1	1080p	10	20	0	3	100	270	17
YUV to AVC	1080p	10	20	1	1	99	508	20
YUV to HEVC	1080p	10	20	1	1	99	652	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	20	1	2	100	510	20
YUV to HEVC	1080p	10	20	1	2	99	360	17
YUV to AVC	1080p	10	20	1	3	99	510	19
YUV to HEVC	1080p	10	20	1	3	99	243	16
YUV to AVC	1080p	10	40	0	1	99	630	19
YUV to HEVC	1080p	10	40	0	1	89	703	21
YUV to AV1	1080p	10	40	0	1	90	646	19
YUV to AVC	1080p	10	40	0	2	99	631	19
YUV to HEVC	1080p	10	40	0	2	99	510	19
YUV to AV1	1080p	10	40	0	2	99	421	18
YUV to AVC	1080p	10	40	0	3	100	630	18
YUV to HEVC	1080p	10	40	0	3	99	336	17
YUV to AV1	1080p	10	40	0	3	99	268	17
YUV to AVC	1080p	10	40	1	1	99	501	20
YUV to HEVC	1080p	10	40	1	1	99	648	22
YUV to AVC	1080p	10	40	1	2	100	500	19
YUV to HEVC	1080p	10	40	1	2	99	357	18
YUV to AVC	1080p	10	40	1	3	99	501	19
YUV to HEVC	1080p	10	40	1	3	99	240	17
YUV to AVC	720p	20	0	0	1	75	1487	13
YUV to HEVC	720p	20	0	0	1	72	1570	14
YUV to AV1	720p	20	0	0	1	78	1408	12
YUV to AVC	720p	20	0	0	2	76	1505	14
YUV to HEVC	720p	20	0	0	2	99	1506	15
YUV to AV1	720p	20	0	0	2	99	1220	15
YUV to AVC	720p	20	0	0	3	77	1500	13
YUV to HEVC	720p	20	0	0	3	99	907	13
YUV to AV1	720p	20	0	0	3	99	660	9
YUV to AVC	720p	20	0	1	1	91	1565	15
YUV to HEVC	720p	20	0	1	1	76	1589	15
YUV to AVC	720p	20	0	1	2	93	1585	15
YUV to HEVC	720p	20	0	1	2	99	960	11
YUV to AVC	720p	20	0	1	3	92	1587	14
YUV to HEVC	720p	20	0	1	3	100	620	10

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	20	4	0	1	98	1102	10
YUV to HEVC	720p	20	4	0	1	99	1117	9
YUV to AV1	720p	20	4	0	1	100	880	7
YUV to AVC	720p	20	4	0	2	99	1101	10
YUV to HEVC	720p	20	4	0	2	99	1080	13
YUV to AV1	720p	20	4	0	2	99	880	7
YUV to AVC	720p	20	4	0	3	99	1103	9
YUV to HEVC	720p	20	4	0	3	99	740	14
YUV to AV1	720p	20	4	0	3	100	560	13
YUV to AVC	720p	20	4	1	1	99	1070	12
YUV to HEVC	720p	20	4	1	1	99	1123	9
YUV to AVC	720p	20	4	1	2	99	1070	13
YUV to HEVC	720p	20	4	1	2	99	760	10
YUV to AVC	720p	20	4	1	3	99	1066	14
YUV to HEVC	720p	20	4	1	3	99	533	8
YUV to AVC	720p	20	20	0	1	97	1081	9
YUV to HEVC	720p	20	20	0	1	99	1120	9
YUV to AV1	720p	20	20	0	1	100	921	7
YUV to AVC	720p	20	20	0	2	98	1080	9
YUV to HEVC	720p	20	20	0	2	99	1079	13
YUV to AV1	720p	20	20	0	2	99	880	11
YUV to AVC	720p	20	20	0	3	98	1081	10
YUV to HEVC	720p	20	20	0	3	99	729	13
YUV to AV1	720p	20	20	0	3	100	560	14
YUV to AVC	720p	20	20	1	1	99	1061	12
YUV to HEVC	720p	20	20	1	1	99	1125	9
YUV to AVC	720p	20	20	1	2	99	1062	12
YUV to HEVC	720p	20	20	1	2	99	760	14
YUV to AVC	720p	20	20	1	3	99	1062	12
YUV to HEVC	720p	20	20	1	3	100	522	8
YUV to AVC	720p	20	40	0	1	99	1060	9
YUV to HEVC	720p	20	40	0	1	99	1095	10
YUV to AV1	720p	20	40	0	1	100	900	8
YUV to AVC	720p	20	40	0	2	98	1060	9

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	720p	20	40	0	2	99	1060	12
YUV to AV1	720p	20	40	0	2	99	876	10
YUV to AVC	720p	20	40	0	3	99	1053	9
YUV to HEVC	720p	20	40	0	3	100	720	9
YUV to AV1	720p	20	40	0	3	99	560	8
YUV to AVC	720p	20	40	1	1	99	1060	12
YUV to HEVC	720p	20	40	1	1	99	1103	9
YUV to AVC	720p	20	40	1	2	99	1060	12
YUV to HEVC	720p	20	40	1	2	99	759	14
YUV to AVC	720p	20	40	1	3	99	1060	13
YUV to HEVC	720p	20	40	1	3	100	520	10

## 38. T1M (Persistent config “E”) – Capped CRF

### 38.1 Encoding with lookaheadDepth

#### 38.1.1 Description

YUV frame is read from an input file on ramdisk and fed into hardware encoder through PCIe.

YUV frame is encoded by hardware encoder with a mix of xcoder-params EnableRdoQuant, rdoLevel, lookaheadDepth, CRF, bitrate, and vbvBufferSize.

Encoded bitstream is read out through PCIe and written into an output file.

#### 38.1.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -i /media/ramdisk/input.list -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:vbvBufferSize=1000:bitrate=<*>:lookaheadDepth=<*>:EnableR  
doQuant=<*>:rdoLevel=<*>:crf=<*> -f null /dev/null -
```

<enc> is the encoder codec. eg h264\_ni\_quadra\_enc, h265\_ni\_quadra\_enc, av1\_ni\_quadra\_enc

<num\_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, bitrate = 12000000, framerate = 30

<resolution> == 1080p, bitrate = 3000000, framerate = 30

## 38.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	0	0	1	19	99	684	22
YUV to HEVC	1080p	10	0	0	1	19	94	770	21
YUV to AV1	1080p	10	0	0	1	19	97	711	21
YUV to AVC	1080p	10	0	0	2	19	99	686	21
YUV to HEVC	1080p	10	0	0	2	19	99	515	19
YUV to AV1	1080p	10	0	0	2	19	100	431	18
YUV to AVC	1080p	10	0	0	3	19	100	685	20
YUV to HEVC	1080p	10	0	0	3	19	100	340	18
YUV to AV1	1080p	10	0	0	3	19	99	278	17
YUV to AVC	1080p	10	0	1	1	19	99	510	21
YUV to HEVC	1080p	10	0	1	1	19	99	659	22
YUV to AVC	1080p	10	0	1	2	19	99	510	20
YUV to HEVC	1080p	10	0	1	2	19	99	360	18
YUV to AVC	1080p	10	0	1	3	19	100	510	19
YUV to HEVC	1080p	10	0	1	3	19	99	247	17
YUV to AVC	1080p	10	4	0	1	19	99	647	19
YUV to HEVC	1080p	10	4	0	1	19	85	691	20
YUV to AV1	1080p	10	4	0	1	19	88	632	18
YUV to AVC	1080p	10	4	0	2	19	99	647	19
YUV to HEVC	1080p	10	4	0	2	19	99	512	20
YUV to AV1	1080p	10	4	0	2	19	100	430	18
YUV to AVC	1080p	10	4	0	3	19	99	647	20
YUV to HEVC	1080p	10	4	0	3	19	99	340	19
YUV to AV1	1080p	10	4	0	3	19	100	272	17
YUV to AVC	1080p	10	4	1	1	19	100	510	20
YUV to HEVC	1080p	10	4	1	1	19	99	657	20
YUV to AVC	1080p	10	4	1	2	19	100	510	20
YUV to HEVC	1080p	10	4	1	2	19	99	360	18
YUV to AVC	1080p	10	4	1	3	19	99	510	18
YUV to HEVC	1080p	10	4	1	3	19	99	242	16
YUV to AVC	1080p	10	16	0	1	19	93	601	17
YUV to HEVC	1080p	10	16	0	1	19	84	673	20
YUV to AV1	1080p	10	16	0	1	19	85	616	18



TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	0	2	19	94	601	18
YUV to HEVC	1080p	10	16	0	2	19	100	510	20
YUV to AV1	1080p	10	16	0	2	19	100	430	17
YUV to AVC	1080p	10	16	0	3	19	94	603	19
YUV to HEVC	1080p	10	16	0	3	19	100	339	18
YUV to AV1	1080p	10	16	0	3	19	99	270	17
YUV to AVC	1080p	10	16	1	1	19	99	509	19
YUV to HEVC	1080p	10	16	1	1	19	99	650	21
YUV to AVC	1080p	10	16	1	2	19	99	507	20
YUV to HEVC	1080p	10	16	1	2	19	100	360	18
YUV to AVC	1080p	10	16	1	3	19	100	507	19
YUV to HEVC	1080p	10	16	1	3	19	99	243	16
YUV to AVC	1080p	10	40	0	1	19	93	587	18
YUV to HEVC	1080p	10	40	0	1	19	80	651	19
YUV to AV1	1080p	10	40	0	1	19	86	589	17
YUV to AVC	1080p	10	40	0	2	19	91	588	17
YUV to HEVC	1080p	10	40	0	2	19	100	508	19
YUV to AV1	1080p	10	40	0	2	19	100	420	17
YUV to AVC	1080p	10	40	0	3	19	92	591	18
YUV to HEVC	1080p	10	40	0	3	19	99	334	17
YUV to AV1	1080p	10	40	0	3	19	99	268	16
YUV to AVC	1080p	10	40	1	1	19	99	500	18
YUV to HEVC	1080p	10	40	1	1	19	99	648	20
YUV to AVC	1080p	10	40	1	2	19	100	500	19
YUV to HEVC	1080p	10	40	1	2	19	100	359	19
YUV to AVC	1080p	10	40	1	3	19	99	500	19
YUV to HEVC	1080p	10	40	1	3	19	99	241	16
YUV to AVC	1080p	10	0	0	1	23	99	685	20
YUV to HEVC	1080p	10	0	0	1	23	92	764	22
YUV to AV1	1080p	10	0	0	1	23	98	711	21
YUV to AVC	1080p	10	0	0	2	23	99	684	21
YUV to HEVC	1080p	10	0	0	2	23	100	510	20
YUV to AV1	1080p	10	0	0	2	23	99	430	19
YUV to AVC	1080p	10	0	0	3	23	99	685	21

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	23	99	340	17
YUV to AV1	1080p	10	0	0	3	23	100	279	16
YUV to AVC	1080p	10	0	1	1	23	100	510	19
YUV to HEVC	1080p	10	0	1	1	23	99	654	22
YUV to AVC	1080p	10	0	1	2	23	100	510	20
YUV to HEVC	1080p	10	0	1	2	23	100	360	19
YUV to AVC	1080p	10	0	1	3	23	99	510	20
YUV to HEVC	1080p	10	0	1	3	23	100	246	17
YUV to AVC	1080p	10	4	0	1	23	99	646	18
YUV to HEVC	1080p	10	4	0	1	23	86	702	21
YUV to AV1	1080p	10	4	0	1	23	89	647	18
YUV to AVC	1080p	10	4	0	2	23	99	647	18
YUV to HEVC	1080p	10	4	0	2	23	100	512	19
YUV to AV1	1080p	10	4	0	2	23	100	430	18
YUV to AVC	1080p	10	4	0	3	23	99	648	19
YUV to HEVC	1080p	10	4	0	3	23	100	340	18
YUV to AV1	1080p	10	4	0	3	23	99	270	16
YUV to AVC	1080p	10	4	1	1	23	99	509	20
YUV to HEVC	1080p	10	4	1	1	23	99	658	21
YUV to AVC	1080p	10	4	1	2	23	100	510	20
YUV to HEVC	1080p	10	4	1	2	23	100	360	18
YUV to AVC	1080p	10	4	1	3	23	99	510	19
YUV to HEVC	1080p	10	4	1	3	23	99	246	16
YUV to AVC	1080p	10	16	0	1	23	95	625	18
YUV to HEVC	1080p	10	16	0	1	23	86	687	21
YUV to AV1	1080p	10	16	0	1	23	88	631	18
YUV to AVC	1080p	10	16	0	2	23	96	625	18
YUV to HEVC	1080p	10	16	0	2	23	99	510	21
YUV to AV1	1080p	10	16	0	2	23	100	430	19
YUV to AVC	1080p	10	16	0	3	23	98	626	18
YUV to HEVC	1080p	10	16	0	3	23	99	338	17
YUV to AV1	1080p	10	16	0	3	23	99	270	16
YUV to AVC	1080p	10	16	1	1	23	99	509	19
YUV to HEVC	1080p	10	16	1	1	23	99	650	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	1	2	23	100	509	19
YUV to HEVC	1080p	10	16	1	2	23	100	360	18
YUV to AVC	1080p	10	16	1	3	23	99	508	20
YUV to HEVC	1080p	10	16	1	3	23	100	244	17
YUV to AVC	1080p	10	40	0	1	23	96	611	17
YUV to HEVC	1080p	10	40	0	1	23	83	668	18
YUV to AV1	1080p	10	40	0	1	23	86	604	18
YUV to AVC	1080p	10	40	0	2	23	96	611	18
YUV to HEVC	1080p	10	40	0	2	23	99	509	21
YUV to AV1	1080p	10	40	0	2	23	100	420	18
YUV to AVC	1080p	10	40	0	3	23	94	611	18
YUV to HEVC	1080p	10	40	0	3	23	100	335	16
YUV to AV1	1080p	10	40	0	3	23	99	270	16
YUV to AVC	1080p	10	40	1	1	23	99	500	19
YUV to HEVC	1080p	10	40	1	1	23	99	648	20
YUV to AVC	1080p	10	40	1	2	23	100	501	19
YUV to HEVC	1080p	10	40	1	2	23	100	357	17
YUV to AVC	1080p	10	40	1	3	23	99	500	19
YUV to HEVC	1080p	10	40	1	3	23	100	242	16
YUV to AVC	1080p	10	0	0	1	27	100	685	21
YUV to HEVC	1080p	10	0	0	1	27	94	771	23
YUV to AV1	1080p	10	0	0	1	27	98	711	21
YUV to AVC	1080p	10	0	0	2	27	99	685	20
YUV to HEVC	1080p	10	0	0	2	27	99	512	20
YUV to AV1	1080p	10	0	0	2	27	100	430	18
YUV to AVC	1080p	10	0	0	3	27	99	683	21
YUV to HEVC	1080p	10	0	0	3	27	99	340	17
YUV to AV1	1080p	10	0	0	3	27	99	279	16
YUV to AVC	1080p	10	0	1	1	27	100	510	20
YUV to HEVC	1080p	10	0	1	1	27	100	657	20
YUV to AVC	1080p	10	0	1	2	27	100	510	20
YUV to HEVC	1080p	10	0	1	2	27	99	360	17
YUV to AVC	1080p	10	0	1	3	27	99	510	20
YUV to HEVC	1080p	10	0	1	3	27	99	247	17

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	4	0	1	27	99	648	20
YUV to HEVC	1080p	10	4	0	1	27	88	716	22
YUV to AV1	1080p	10	4	0	1	27	91	658	18
YUV to AVC	1080p	10	4	0	2	27	99	649	18
YUV to HEVC	1080p	10	4	0	2	27	100	512	20
YUV to AV1	1080p	10	4	0	2	27	99	430	18
YUV to AVC	1080p	10	4	0	3	27	99	649	19
YUV to HEVC	1080p	10	4	0	3	27	99	340	18
YUV to AV1	1080p	10	4	0	3	27	99	271	16
YUV to AVC	1080p	10	4	1	1	27	99	510	20
YUV to HEVC	1080p	10	4	1	1	27	99	659	21
YUV to AVC	1080p	10	4	1	2	27	99	510	20
YUV to HEVC	1080p	10	4	1	2	27	99	360	19
YUV to AVC	1080p	10	4	1	3	27	100	509	19
YUV to HEVC	1080p	10	4	1	3	27	100	245	17
YUV to AVC	1080p	10	16	0	1	27	99	641	19
YUV to HEVC	1080p	10	16	0	1	27	87	701	20
YUV to AV1	1080p	10	16	0	1	27	89	640	18
YUV to AVC	1080p	10	16	0	2	27	99	640	19
YUV to HEVC	1080p	10	16	0	2	27	100	510	20
YUV to AV1	1080p	10	16	0	2	27	99	429	18
YUV to AVC	1080p	10	16	0	3	27	99	640	19
YUV to HEVC	1080p	10	16	0	3	27	99	339	18
YUV to AV1	1080p	10	16	0	3	27	100	269	16
YUV to AVC	1080p	10	16	1	1	27	100	507	19
YUV to HEVC	1080p	10	16	1	1	27	99	651	20
YUV to AVC	1080p	10	16	1	2	27	99	508	19
YUV to HEVC	1080p	10	16	1	2	27	100	360	18
YUV to AVC	1080p	10	16	1	3	27	100	509	19
YUV to HEVC	1080p	10	16	1	3	27	99	244	16
YUV to AVC	1080p	10	40	0	1	27	99	631	19
YUV to HEVC	1080p	10	40	0	1	27	85	678	21
YUV to AV1	1080p	10	40	0	1	27	86	614	18
YUV to AVC	1080p	10	40	0	2	27	99	631	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	0	2	27	99	510	20
YUV to AV1	1080p	10	40	0	2	27	99	420	17
YUV to AVC	1080p	10	40	0	3	27	99	630	18
YUV to HEVC	1080p	10	40	0	3	27	99	334	18
YUV to AV1	1080p	10	40	0	3	27	99	269	17
YUV to AVC	1080p	10	40	1	1	27	99	502	19
YUV to HEVC	1080p	10	40	1	1	27	99	647	20
YUV to AVC	1080p	10	40	1	2	27	99	500	20
YUV to HEVC	1080p	10	40	1	2	27	99	358	17
YUV to AVC	1080p	10	40	1	3	27	99	500	19
YUV to HEVC	1080p	10	40	1	3	27	99	242	16

## Appendix A: GStreamer XStack Command

Example of a 4x8 grid with 32 inputs and an output resolution of 1920x1080 with each cell 480x135.

[illegible]

```
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack. multifilesrc  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec  
xcoder-params='out=hw' ! xstack.  
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec
```

## Appendix B: 7x7 Grid Layout

Size of each cell in a 7x7 grid with 49 outputs. Overall output resolution is 1080p

	274x154		274x154		274x154		274x154		274x154		274x154		276x154	
	274x154		274x154		274x154		274x154		274x154		274x154		276x154	
	274x154		274x154		274x154		274x154		274x154		274x154		276x154	
	274x154		274x154		274x154		274x154		274x154		274x154		276x154	
	274x154		274x154		274x154		274x154		274x154		274x154		276x154	
	274x154		274x154		274x154		274x154		274x154		274x154		276x154	
	274x156		274x156		274x156		274x156		274x156		274x156		276x156	



## Appendix C: GStreamer Ladder Command

### Example of single input with 64 outputs

```
gst-launch-1.0 multiqueue sync-by-running-time=TRUE max-size-bytes=0 max-  
size-buffers=0 max-size-time=0 name=mq multifilesrc  
location=/media/ramdisk/input.h264 loop=true ! h264parse ! niquadrah264dec !  
tee name=t ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink  
sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-  
sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink  
video-sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc !  
fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq. mq. !  
niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq.  
mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t.  
! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -  
v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink  
sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-  
sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink  
video-sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc !  
fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq. mq. !  
niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq.  
mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t.  
! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -  
v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink  
sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-  
sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink  
video-sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc !  
fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq. mq. !  
niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq.  
mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t.  
! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -  
v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink  
sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-  
sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink  
video-sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc !  
fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq. mq. !  
niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq.  
mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t.  
! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -  
v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink  
sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-  
sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink  
video-sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc !  
fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq. mq. !  
niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq.
```

```

mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t.
! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -
v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink
sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-
sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink
video-sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc !
fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq. mq. !
niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq.
mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t.
! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -
v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink
sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-
sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink
video-sink=fakesink sync=false -v t. ! mq. mq. ! niquadrah265enc !
fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq. mq. !
niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t. ! mq.
mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -v t.
! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink sync=false -
v t. ! mq. mq. ! niquadrah265enc ! fpsdisplaysink video-sink=fakesink
sync=false -v

```