



# Quadra™ Performance Test Report V5.6

## 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 .....	126
Appendix A: GStreamer XStack Command .....	132
Appendix B: 7x7 Grid Layout.....	135
Appendix C: GStreamer Ladder Command .....	136

## Environment Overview

Revision: 5606sVr3

### 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: AMD Ryzen 5 7600X 6-core Processor; CPU(s) 12; Motherboard ASUSTeK COMPUTER INC. Prime x670-P; Memory 32GiB System Memory, 2x16GiB DIMM DDR5 Synchronous Unbuffered (Unregistered) 4800 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) 2400 MHz (0.42 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

#### Setup #5:

- 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
  - RGBA Encoding
  - Encoding EnableRdoQuant/rdoLevel/lookaheadDepth
  - Capped CRF

## 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	80	0	104	7
HEVC to YUV	8k	1	0	8	1	71	0	105	8
VP9 to YUV	8k	1	0	8	1	24	0	38	7
YUV to AVC	8k	1	0	8	1	0	97	67	54
YUV to HEVC	8k	1	0	8	1	0	98	83	66
AVC to AVC	8k	1	1	8	1	68	99	54	3
AVC to HEVC	8k	1	1	8	1	82	98	71	6
HEVC to AVC	8k	1	1	8	1	64	99	52	3
HEVC to HEVC	8k	1	1	8	1	71	100	70	4
VP9 to AVC	8k	1	1	8	1	25	44	35	3
VP9 to HEVC	8k	1	1	8	1	24	43	37	2
AVC to YUV	8k	1	0	10	1	59	0	60	7
HEVC to YUV	8k	1	0	10	1	58	0	57	8
VP9 to YUV	8k	1	0	10	1	24	0	33	12
YUV to AVC	8k	1	0	10	1	0	89	48	72
YUV to HEVC	8k	1	0	10	1	0	71	59	91
AVC to YUV	4k	1	0	8	1	57	0	319	21
HEVC to YUV	4k	1	0	8	1	50	0	329	26
VP9 to YUV	4k	1	0	8	1	24	0	154	7
AVC to YUV	4k	16	0	8	0	98	0	481	4
HEVC to YUV	4k	16	0	8	0	98	0	504	4
VP9 to YUV	4k	16	0	8	0	100	0	486	1
YUV to AVC	4k	1	0	8	1	0	95	294	42
YUV to HEVC	4k	1	0	8	1	0	95	327	44
YUV to AV1	4k	1	0	8	1	0	94	281	37
YUV to AVC	4k	4	0	8	0	0	96	301	20
YUV to HEVC	4k	4	0	8	0	0	96	332	13
YUV to AV1	4k	4	0	8	0	0	92	284	24
YUV to AVC	4k	8	0	8	0	0	100	320	8
YUV to HEVC	4k	8	0	8	0	0	100	344	8
YUV to AV1	4k	8	0	8	0	0	100	296	7
AVC to AVC	4k	1	1	8	1	70	96	221	12
AVC to HEVC	4k	1	1	8	1	69	90	273	18
AVC to AV1	4k	1	1	8	1	62	95	263	14
HEVC to AVC	4k	1	1	8	1	56	93	220	14
HEVC to HEVC	4k	1	1	8	1	62	92	274	16
HEVC to AV1	4k	1	1	8	1	55	97	257	10
VP9 to AVC	4k	1	1	8	1	24	45	144	4
VP9 to HEVC	4k	1	1	8	1	24	43	148	3

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	50	150	3
AVC to AVC	4k	4	1	8	0	62	94	220	3
AVC to HEVC	4k	4	1	8	0	69	96	300	4
AVC to AV1	4k	4	1	8	0	60	95	276	7
HEVC to AVC	4k	4	1	8	0	57	96	236	8
HEVC to HEVC	4k	4	1	8	0	60	96	296	3
HEVC to AV1	4k	4	1	8	0	46	95	276	8
VP9 to AVC	4k	4	1	8	0	65	97	242	3
VP9 to HEVC	4k	4	1	8	0	67	96	304	3
VP9 to AV1	4k	4	1	8	0	56	95	272	3
AVC to AVC	4k	8	1	8	0	68	100	211	2
AVC to HEVC	4k	8	1	8	0	76	99	276	3
AVC to AV1	4k	8	1	8	0	67	99	271	3
HEVC to AVC	4k	8	1	8	0	63	100	217	2
HEVC to HEVC	4k	8	1	8	0	68	100	280	2
HEVC to AV1	4k	8	1	8	0	58	100	279	2
VP9 to AVC	4k	8	1	8	0	67	100	233	2
VP9 to HEVC	4k	8	1	8	0	73	100	298	2
VP9 to AV1	4k	8	1	8	0	62	99	280	2
AVC to YUV	4k	1	0	10	1	46	0	225	9
HEVC to YUV	4k	1	0	10	1	52	0	227	9
VP9 to YUV	4k	1	0	10	1	24	0	124	3
AVC to YUV	4k	16	0	10	0	99	0	284	0
HEVC to YUV	4k	16	0	10	0	100	0	282	0
VP9 to YUV	4k	16	0	10	0	98	0	263	0
YUV to AVC	4k	1	0	10	1	0	69	200	55
YUV to HEVC	4k	1	0	10	1	0	60	207	57
YUV to AV1	4k	1	0	10	1	0	66	199	55
YUV to AVC	4k	4	0	10	0	0	96	220	27
YUV to HEVC	4k	4	0	10	0	0	76	251	39
YUV to AV1	4k	4	0	10	0	0	82	240	42
AVC to YUV	1080p	1	0	8	1	43	0	890	22
HEVC to YUV	1080p	1	0	8	1	45	0	869	27
VP9 to YUV	1080p	1	0	8	1	22	0	555	6
AVC to YUV	1080p	40	0	8	0	89	0	1731	1
HEVC to YUV	1080p	40	0	8	0	96	0	1841	1
VP9 to YUV	1080p	40	0	8	0	83	0	1795	0
YUV to AVC	1080p	1	0	8	1	0	56	727	27
YUV to HEVC	1080p	1	0	8	1	0	53	729	25
YUV to AV1	1080p	1	0	8	1	0	56	666	21

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1281	3
YUV to HEVC	1080p	32	0	8	0	0	99	1350	3
YUV to AV1	1080p	32	0	8	0	0	99	1179	2
AVC to AVC	1080p	1	1	8	1	67	81	960	25
AVC to HEVC	1080p	1	1	8	1	66	77	1024	24
AVC to AV1	1080p	1	1	8	1	53	79	899	20
HEVC to AVC	1080p	1	1	8	1	55	76	896	30
HEVC to HEVC	1080p	1	1	8	1	57	76	970	30
HEVC to AV1	1080p	1	1	8	1	47	76	851	27
VP9 to AVC	1080p	1	1	8	1	22	41	532	8
VP9 to HEVC	1080p	1	1	8	1	22	39	539	9
VP9 to AV1	1080p	1	1	8	1	22	46	540	6
AVC to AVC	1080p	32	1	8	0	75	100	935	2
AVC to HEVC	1080p	32	1	8	0	84	99	1060	1
AVC to AV1	1080p	32	1	8	0	76	100	1026	1
HEVC to AVC	1080p	32	1	8	0	69	100	992	1
HEVC to HEVC	1080p	32	1	8	0	77	99	1121	1
HEVC to AV1	1080p	32	1	8	0	70	99	1057	1
VP9 to AVC	1080p	32	1	8	0	71	99	1088	1
VP9 to HEVC	1080p	32	1	8	0	75	99	1216	1
VP9 to AV1	1080p	32	1	8	0	72	99	1120	1
AVC to YUV	1080p	1	0	10	1	32	0	683	7
HEVC to YUV	1080p	1	0	10	1	26	0	694	10
VP9 to YUV	1080p	1	0	10	1	22	0	452	6
AVC to YUV	1080p	40	0	10	0	62	0	1090	0
HEVC to YUV	1080p	40	0	10	0	67	0	1080	0
VP9 to YUV	1080p	40	0	10	0	71	0	1075	0
YUV to AVC	1080p	1	0	10	1	0	39	514	31
YUV to HEVC	1080p	1	0	10	1	0	37	514	36
YUV to AV1	1080p	1	0	10	1	0	40	477	33
YUV to AVC	1080p	32	0	10	0	0	63	800	5
YUV to HEVC	1080p	32	0	10	0	0	59	800	6
YUV to AV1	1080p	32	0	10	0	0	64	770	5
AVC to YUV	720p	1	0	8	1	45	0	1245	16
HEVC to YUV	720p	1	0	8	1	40	0	1265	27
VP9 to YUV	720p	1	0	8	1	31	0	1014	8
AVC to YUV	720p	100	0	8	0	100	0	2512	0
HEVC to YUV	720p	100	0	8	0	100	0	2912	1
VP9 to YUV	720p	100	0	8	0	100	0	2620	0
YUV to AVC	720p	1	0	8	1	0	31	906	15

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	31	914	18
YUV to AV1	720p	1	0	8	1	0	35	820	13
YUV to AVC	720p	64	0	8	0	0	98	2371	1
YUV to HEVC	720p	64	0	8	0	0	97	2428	2
YUV to AV1	720p	64	0	8	0	0	98	2025	1
AVC to AVC	720p	1	1	8	1	48	48	1272	20
AVC to HEVC	720p	1	1	8	1	48	47	1267	22
AVC to AV1	720p	1	1	8	1	41	51	1097	18
HEVC to AVC	720p	1	1	8	1	41	46	1246	25
HEVC to HEVC	720p	1	1	8	1	41	46	1232	24
HEVC to AV1	720p	1	1	8	1	35	49	1078	22
VP9 to AVC	720p	1	1	8	1	31	35	974	9
VP9 to HEVC	720p	1	1	8	1	32	34	985	13
VP9 to AV1	720p	1	1	8	1	31	44	983	14
AVC to AVC	720p	64	1	8	0	96	100	2048	0
AVC to HEVC	720p	64	1	8	0	96	100	2116	0
AVC to AV1	720p	64	1	8	0	77	100	1792	0
HEVC to AVC	720p	64	1	8	0	84	100	2049	0
HEVC to HEVC	720p	64	1	8	0	85	100	2123	0
HEVC to AV1	720p	64	1	8	0	69	100	1792	0
VP9 to AVC	720p	64	1	8	0	97	100	2243	0
VP9 to HEVC	720p	64	1	8	0	100	100	2310	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 Libxcodec 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	65	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	37	2
YUV to AVC	8k	1	0	8	1	0	96	67	26
YUV to HEVC	8k	1	0	8	1	0	95	82	33
AVC to AVC	8k	1	1	8	1	72	100	54	5
AVC to HEVC	8k	1	1	8	1	79	99	71	5
HEVC to AVC	8k	1	1	8	1	62	99	53	4
HEVC to HEVC	8k	1	1	8	1	73	99	70	5
VP9 to AVC	8k	1	1	8	1	24	46	35	4
VP9 to HEVC	8k	1	1	8	1	25	42	36	4
AVC to YUV	8k	1	0	10	1	57	0	54	6
HEVC to YUV	8k	1	0	10	1	50	0	51	6
VP9 to YUV	8k	1	0	10	1	24	0	32	4
YUV to AVC	8k	1	0	10	1	0	63	40	34
YUV to HEVC	8k	1	0	10	1	0	46	40	34
AVC to YUV	4k	1	0	8	1	58	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	153	4
AVC to YUV	4k	16	0	8	0	98	0	485	0
HEVC to YUV	4k	16	0	8	0	99	0	510	0
VP9 to YUV	4k	16	0	8	0	99	0	491	0
YUV to AVC	4k	1	0	8	1	0	92	287	24
YUV to HEVC	4k	1	0	8	1	0	88	303	24
YUV to AV1	4k	1	0	8	1	0	92	278	23
YUV to AVC	4k	4	0	8	0	0	96	311	10
YUV to HEVC	4k	4	0	8	0	0	96	334	10
YUV to AV1	4k	4	0	8	0	0	96	289	10
YUV to AVC	4k	8	0	8	0	0	99	327	7
YUV to HEVC	4k	8	0	8	0	0	99	347	8
YUV to AV1	4k	8	0	8	0	0	100	301	10
AVC to AVC	4k	1	1	8	0	67	97	225	7
AVC to HEVC	4k	1	1	8	0	74	96	277	8
AVC to AV1	4k	1	1	8	0	64	97	262	8
HEVC to AVC	4k	1	1	8	0	58	97	225	5
HEVC to HEVC	4k	1	1	8	0	63	96	286	6
HEVC to AV1	4k	1	1	8	0	57	97	266	6
VP9 to AVC	4k	1	1	8	0	24	46	143	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	148	4
VP9 to AV1	4k	1	1	8	0	24	50	150	4
AVC to AVC	4k	4	1	8	0	61	96	243	5
AVC to HEVC	4k	4	1	8	0	67	96	302	5
AVC to AV1	4k	4	1	8	0	57	95	277	5
HEVC to AVC	4k	4	1	8	0	57	96	239	5
HEVC to HEVC	4k	4	1	8	0	61	95	293	5
HEVC to AV1	4k	4	1	8	0	49	95	274	5
VP9 to AVC	4k	4	1	8	0	63	97	246	4
VP9 to HEVC	4k	4	1	8	0	67	95	303	4
VP9 to AV1	4k	4	1	8	0	57	95	277	4
AVC to AVC	4k	8	1	8	0	69	99	217	3
AVC to HEVC	4k	8	1	8	0	76	100	280	4
AVC to AV1	4k	8	1	8	0	67	99	274	4
HEVC to AVC	4k	8	1	8	0	65	100	219	3
HEVC to HEVC	4k	8	1	8	0	66	99	282	3
HEVC to AV1	4k	8	1	8	0	61	99	275	4
VP9 to AVC	4k	8	1	8	0	68	99	235	3
VP9 to HEVC	4k	8	1	8	0	72	100	304	3
VP9 to AV1	4k	8	1	8	0	63	99	287	4
AVC to YUV	4k	1	0	10	1	46	0	215	6
HEVC to YUV	4k	1	0	10	1	41	0	208	6
VP9 to YUV	4k	1	0	10	1	24	0	124	4
AVC to YUV	4k	16	0	10	0	100	0	282	0
HEVC to YUV	4k	16	0	10	0	100	0	282	0
VP9 to YUV	4k	16	0	10	0	99	0	266	0
YUV to AVC	4k	1	0	10	1	0	51	165	25
YUV to HEVC	4k	1	0	10	1	0	47	166	26
YUV to AV1	4k	1	0	10	1	0	52	156	25
YUV to AVC	4k	4	0	10	0	0	66	209	24
YUV to HEVC	4k	4	0	10	0	0	63	210	26
YUV to AV1	4k	4	0	10	0	0	63	191	27
AVC to YUV	1080p	1	0	8	1	41	0	844	13
HEVC to YUV	1080p	1	0	8	1	46	0	893	10
VP9 to YUV	1080p	1	0	8	1	22	0	554	4
AVC to YUV	1080p	40	0	8	0	81	0	1660	0
HEVC to YUV	1080p	40	0	8	0	87	0	1780	0
VP9 to YUV	1080p	40	0	8	0	72	0	1674	0
YUV to AVC	1080p	1	0	8	1	0	58	753	16



TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	55	753	16
YUV to AV1	1080p	1	0	8	1	0	52	620	14
YUV to AVC	1080p	32	0	8	0	0	100	1326	2
YUV to HEVC	1080p	32	0	8	0	0	99	1402	3
YUV to AV1	1080p	32	0	8	0	0	99	1211	2
AVC to AVC	1080p	1	1	8	0	70	85	1008	16
AVC to HEVC	1080p	1	1	8	0	68	81	1063	16
AVC to AV1	1080p	1	1	8	0	55	82	930	16
HEVC to AVC	1080p	1	1	8	0	61	85	990	12
HEVC to HEVC	1080p	1	1	8	0	64	85	1092	13
HEVC to AV1	1080p	1	1	8	0	52	83	934	13
VP9 to AVC	1080p	1	1	8	0	21	42	546	6
VP9 to HEVC	1080p	1	1	8	0	22	39	538	6
VP9 to AV1	1080p	1	1	8	0	22	46	537	7
AVC to AVC	1080p	32	1	8	0	75	99	976	1
AVC to HEVC	1080p	32	1	8	0	84	99	1104	1
AVC to AV1	1080p	32	1	8	0	76	99	1064	1
HEVC to AVC	1080p	32	1	8	0	69	100	1020	1
HEVC to HEVC	1080p	32	1	8	0	77	100	1155	1
HEVC to AV1	1080p	32	1	8	0	68	99	1090	1
VP9 to AVC	1080p	32	1	8	0	70	99	1114	0
VP9 to HEVC	1080p	32	1	8	0	74	99	1258	0
VP9 to AV1	1080p	32	1	8	0	73	99	1156	0
AVC to YUV	1080p	1	0	10	1	29	0	666	7
HEVC to YUV	1080p	1	0	10	1	26	0	668	6
VP9 to YUV	1080p	1	0	10	1	22	0	451	4
AVC to YUV	1080p	40	0	10	0	66	0	1077	0
HEVC to YUV	1080p	40	0	10	0	66	0	1073	0
VP9 to YUV	1080p	40	0	10	0	68	0	1034	0
YUV to AVC	1080p	1	0	10	1	0	39	504	23
YUV to HEVC	1080p	1	0	10	1	0	36	504	20
YUV to AV1	1080p	1	0	10	1	0	36	419	18
YUV to AVC	1080p	32	0	10	0	0	65	863	6
YUV to HEVC	1080p	32	0	10	0	0	60	854	5
YUV to AV1	1080p	32	0	10	0	0	65	806	8
AVC to YUV	720p	1	0	8	1	42	0	1178	12
HEVC to YUV	720p	1	0	8	1	39	0	1248	8
VP9 to YUV	720p	1	0	8	1	31	0	1015	6
AVC to YUV	720p	100	0	8	0	100	0	2754	0

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	93	0	2954	0
VP9 to YUV	720p	100	0	8	0	91	0	2659	0
YUV to AVC	720p	1	0	8	1	0	33	980	12
YUV to HEVC	720p	1	0	8	1	0	34	1013	11
YUV to AV1	720p	1	0	8	1	0	32	787	10
YUV to AVC	720p	64	0	8	0	0	100	2661	2
YUV to HEVC	720p	64	0	8	0	0	99	2732	2
YUV to AV1	720p	64	0	8	0	0	100	2198	1
AVC to AVC	720p	1	1	8	0	51	50	1347	16
AVC to HEVC	720p	1	1	8	0	52	50	1373	16
AVC to AV1	720p	1	1	8	0	43	49	1078	15
HEVC to AVC	720p	1	1	8	0	47	51	1385	11
HEVC to HEVC	720p	1	1	8	0	47	50	1404	11
HEVC to AV1	720p	1	1	8	0	39	48	1082	12
VP9 to AVC	720p	1	1	8	0	31	34	986	8
VP9 to HEVC	720p	1	1	8	0	31	34	985	7
VP9 to AV1	720p	1	1	8	0	31	44	984	9
AVC to AVC	720p	64	1	8	0	94	99	2180	0
AVC to HEVC	720p	64	1	8	0	100	100	2274	0
AVC to AV1	720p	64	1	8	0	94	100	1884	0
HEVC to AVC	720p	64	1	8	0	95	100	2184	0
HEVC to HEVC	720p	64	1	8	0	96	100	2276	0
HEVC to AV1	720p	64	1	8	0	88	100	1891	0
VP9 to AVC	720p	64	1	8	0	100	100	2423	0
VP9 to HEVC	720p	64	1	8	0	100	100	2500	0
VP9 to AV1	720p	64	1	8	0	99	100	2028	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	57.84	59.08	57.26	0.04
YUV to HEVC	8k	1	55.45	60.52	54.11	0.93
YUV to AVC	4k	1	15.43	17.45	15.01	0.06
YUV to HEVC	4k	1	16.66	18.76	15.5	0.24
YUV to AV1	4k	1	21.69	25.34	16.2	0.64
YUV to AVC	4k	4	16.71	21.57	14.95	2.09
YUV to HEVC	4k	4	17.93	21.84	15.62	1.85
YUV to AV1	4k	4	22.76	27.06	16.34	0.77
YUV to AVC	4k	8	20.56	33.4	15.05	22.82
YUV to HEVC	4k	8	22.61	35.54	15.69	26.77
YUV to AV1	4k	8	37.24	45.09	24.73	10.11
YUV to AVC	1080p	1	4.57	5.47	4.4	0.02
YUV to HEVC	1080p	1	4.99	5.6	4.8	0.02
YUV to AV1	1080p	1	6.73	7.44	5.04	0.05
YUV to AVC	1080p	32	6.05	12.86	4.48	0.39
YUV to HEVC	1080p	32	6.86	11.73	5.09	1.46
YUV to AV1	1080p	32	41.03	48.68	34.34	1.12
YUV to AVC	720p	1	2.81	3.42	2.47	0.01
YUV to HEVC	720p	1	2.84	3.29	2.64	0.01
YUV to AV1	720p	1	3.79	4.19	3.01	0.01
YUV to AVC	720p	64	5.13	8.92	2.75	1.07
YUV to HEVC	720p	64	5.51	10.52	2.79	1.32
YUV to AV1	720p	64	39.78	46.96	33.18	2.74

## 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	90	0	142	6
HEVC to RGBA	8k	1	90	0	154	8
VP9 to RGBA	8k	1	22	0	40	2
AVC to RGBA	4k	1	22	0	145	8
AVC to RGBA	4k	16	93	3	584	2
HEVC to RGBA	4k	1	21	0	174	10
HEVC to RGBA	4k	16	93	4	669	2
VP9 to RGBA	4k	1	22	0	166	4
VP9 to RGBA	4k	16	92	4	679	1
AVC to RGBA	1080p	40	94	14	1901	1
HEVC to RGBA	1080p	40	94	14	2010	1
VP9 to RGBA	1080p	40	92	17	2440	0
AVC to RGBA	720p	100	96	18	2600	0
HEVC to RGBA	720p	100	91	20	2881	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	33	94	2	436	4
AVC to HEVC	8	32	94	2	469	4
AVC to AV1	8	23	94	2	392	3
HEVC to AVC	8	34	93	2	440	5
HEVC to HEVC	8	32	92	2	472	5
HEVC to AV1	8	24	96	2	400	4
VP9 to AVC	8	40	94	2	432	4
VP9 to HEVC	8	39	94	2	465	4
VP9 to AV1	8	29	94	2	400	3

## 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	52	162	61
RGBA to HEVC	4k	1	1	46	161	62
RGBA to AV1	4k	1	1	55	165	64
RGBA to AVC	4k	4	0	66	166	37
RGBA to HEVC	4k	4	0	61	169	40
RGBA to AV1	4k	4	0	62	172	42
RGBA to AVC	4k	8	0	63	176	30
RGBA to HEVC	4k	8	0	57	176	29
RGBA to AV1	4k	8	0	61	177	36
RGBA to AVC	1080p	1	1	34	447	42
RGBA to HEVC	1080p	1	1	32	439	50
RGBA to AV1	1080p	1	1	36	423	42
RGBA to AVC	1080p	16	0	58	645	15
RGBA to HEVC	1080p	16	0	53	654	15
RGBA to AV1	1080p	16	0	56	642	15
RGBA to AVC	1080p	32	0	57	671	8
RGBA to HEVC	1080p	32	0	53	672	8
RGBA to AV1	1080p	32	0	58	657	8
RGBA to AVC	720p	1	1	29	717	51
RGBA to HEVC	720p	1	1	29	728	63
RGBA to AV1	720p	1	1	34	751	64
RGBA to AVC	720p	16	0	49	1243	37
RGBA to HEVC	720p	16	0	47	1221	39
RGBA to AV1	720p	16	0	55	1166	32
RGBA to AVC	720p	32	0	46	1174	31
RGBA to HEVC	720p	32	0	45	1175	31
RGBA to AV1	720p	32	0	57	1181	29



## 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	95	293	20
YUV to HEVC	4k	4	0	0	1	96	332	13
YUV to AV1	4k	4	0	0	1	96	288	12
YUV to AVC	4k	4	0	0	2	96	302	18
YUV to HEVC	4k	4	0	0	2	98	171	8
YUV to AV1	4k	4	0	0	2	98	140	7
YUV to AVC	4k	4	0	0	3	93	296	27
YUV to HEVC	4k	4	0	0	3	98	100	11
YUV to AV1	4k	4	0	0	3	100	76	4
YUV to AVC	4k	4	0	1	1	96	188	10
YUV to HEVC	4k	4	0	1	1	97	240	9
YUV to AVC	4k	4	0	1	2	98	188	7
YUV to HEVC	4k	4	0	1	2	97	108	6
YUV to AVC	4k	4	0	1	3	98	188	7
YUV to HEVC	4k	4	0	1	3	100	68	4
YUV to AVC	4k	4	4	0	1	100	196	10
YUV to HEVC	4k	4	4	0	1	100	228	12
YUV to AV1	4k	4	4	0	1	100	200	9
YUV to AVC	4k	4	4	0	2	99	196	9
YUV to HEVC	4k	4	4	0	2	100	136	6
YUV to AV1	4k	4	4	0	2	99	116	5
YUV to AVC	4k	4	4	0	3	99	199	9
YUV to HEVC	4k	4	4	0	3	99	88	4
YUV to AV1	4k	4	4	0	3	100	72	4
YUV to AVC	4k	4	4	1	1	100	140	9
YUV to HEVC	4k	4	4	1	1	100	180	8
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	100	140	6
YUV to HEVC	4k	4	4	1	3	99	64	3
YUV to AVC	4k	4	16	0	1	100	196	9
YUV to HEVC	4k	4	16	0	1	99	228	12
YUV to AV1	4k	4	16	0	1	100	199	10
YUV to AVC	4k	4	16	0	2	100	196	9
YUV to HEVC	4k	4	16	0	2	99	136	8
YUV to AV1	4k	4	16	0	2	98	116	6
YUV to AVC	4k	4	16	0	3	100	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	5
YUV to HEVC	4k	4	16	1	1	99	180	7
YUV to AVC	4k	4	16	1	2	99	140	6
YUV to HEVC	4k	4	16	1	2	100	95	4
YUV to AVC	4k	4	16	1	3	99	140	6
YUV to HEVC	4k	4	16	1	3	99	64	4
YUV to AVC	4k	4	40	0	1	100	196	9
YUV to HEVC	4k	4	40	0	1	99	224	12
YUV to AV1	4k	4	40	0	1	100	196	11
YUV to AVC	4k	4	40	0	2	99	196	9
YUV to HEVC	4k	4	40	0	2	99	136	6
YUV to AV1	4k	4	40	0	2	100	112	6
YUV to AVC	4k	4	40	0	3	100	196	9
YUV to HEVC	4k	4	40	0	3	100	88	5
YUV to AV1	4k	4	40	0	3	100	72	3
YUV to AVC	4k	4	40	1	1	99	136	6
YUV to HEVC	4k	4	40	1	1	99	176	8
YUV to AVC	4k	4	40	1	2	100	136	6
YUV to HEVC	4k	4	40	1	2	100	92	5
YUV to AVC	4k	4	40	1	3	100	136	6
YUV to HEVC	4k	4	40	1	3	99	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	100	1180	4
YUV to AVC	1080p	20	0	0	2	99	1280	4
YUV to HEVC	1080p	20	0	0	2	99	685	2
YUV to AV1	1080p	20	0	0	2	99	560	2
YUV to AVC	1080p	20	0	0	3	100	1280	4
YUV to HEVC	1080p	20	0	0	3	99	402	2
YUV to AV1	1080p	20	0	0	3	99	300	1
YUV to AVC	1080p	20	0	1	1	100	760	2
YUV to HEVC	1080p	20	0	1	1	99	960	3
YUV to AVC	1080p	20	0	1	2	99	760	3
YUV to HEVC	1080p	20	0	1	2	99	440	2
YUV to AVC	1080p	20	0	1	3	100	760	2
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	100	720	2
YUV to AVC	1080p	20	4	0	2	99	700	2
YUV to HEVC	1080p	20	4	0	2	99	517	2
YUV to AV1	1080p	20	4	0	2	99	420	1
YUV to AVC	1080p	20	4	0	3	99	700	2
YUV to HEVC	1080p	20	4	0	3	99	340	2
YUV to AV1	1080p	20	4	0	3	99	270	1
YUV to AVC	1080p	20	4	1	1	99	500	2
YUV to HEVC	1080p	20	4	1	1	99	660	2
YUV to AVC	1080p	20	4	1	2	99	500	2
YUV to HEVC	1080p	20	4	1	2	99	360	1
YUV to AVC	1080p	20	4	1	3	99	501	2
YUV to HEVC	1080p	20	4	1	3	99	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	99	720	2
YUV to AVC	1080p	20	16	0	2	100	700	2
YUV to HEVC	1080p	20	16	0	2	100	502	2
YUV to AV1	1080p	20	16	0	2	100	420	1
YUV to AVC	1080p	20	16	0	3	99	700	2
YUV to HEVC	1080p	20	16	0	3	99	340	1
YUV to AV1	1080p	20	16	0	3	100	261	1
YUV to AVC	1080p	20	16	1	1	99	500	2
YUV to HEVC	1080p	20	16	1	1	99	644	2
YUV to AVC	1080p	20	16	1	2	100	500	2
YUV to HEVC	1080p	20	16	1	2	100	360	1
YUV to AVC	1080p	20	16	1	3	99	500	2
YUV to HEVC	1080p	20	16	1	3	99	240	1
YUV to AVC	1080p	20	40	0	1	100	693	2
YUV to HEVC	1080p	20	40	0	1	99	805	3
YUV to AV1	1080p	20	40	0	1	99	701	2
YUV to AVC	1080p	20	40	0	2	99	688	2
YUV to HEVC	1080p	20	40	0	2	100	500	2
YUV to AV1	1080p	20	40	0	2	99	420	2
YUV to AVC	1080p	20	40	0	3	99	690	2
YUV to HEVC	1080p	20	40	0	3	100	339	1
YUV to AV1	1080p	20	40	0	3	100	260	1
YUV to AVC	1080p	20	40	1	1	100	500	2
YUV to HEVC	1080p	20	40	1	1	100	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	94	2352	2
YUV to HEVC	720p	40	0	0	1	91	2369	3
YUV to AV1	720p	40	0	0	1	96	2036	2
YUV to AVC	720p	40	0	0	2	94	2360	3
YUV to HEVC	720p	40	0	0	2	99	1520	1
YUV to AV1	720p	40	0	0	2	100	1203	1
YUV to AVC	720p	40	0	0	3	94	2356	2
YUV to HEVC	720p	40	0	0	3	99	920	1
YUV to AV1	720p	40	0	0	3	100	651	1
YUV to AVC	720p	40	0	1	1	99	1687	1
YUV to HEVC	720p	40	0	1	1	99	2080	2
YUV to AVC	720p	40	0	1	2	99	1689	1
YUV to HEVC	720p	40	0	1	2	100	960	1
YUV to AVC	720p	40	0	1	3	99	1681	1
YUV to HEVC	720p	40	0	1	3	99	610	1
YUV to AVC	720p	40	4	0	1	99	1440	1
YUV to HEVC	720p	40	4	0	1	100	1360	1
YUV to AV1	720p	40	4	0	1	100	1027	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	100	560	1
YUV to AVC	720p	40	4	1	1	99	1080	1
YUV to HEVC	720p	40	4	1	1	99	1349	1
YUV to AVC	720p	40	4	1	2	99	1077	1
YUV to HEVC	720p	40	4	1	2	99	760	1
YUV to AVC	720p	40	4	1	3	99	1079	1
YUV to HEVC	720p	40	4	1	3	100	520	1
YUV to AVC	720p	40	16	0	1	100	1406	1
YUV to HEVC	720p	40	16	0	1	100	1362	1
YUV to AV1	720p	40	16	0	1	100	1079	1
YUV to AVC	720p	40	16	0	2	100	1405	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	1400	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	1053	1
YUV to HEVC	720p	40	16	1	1	99	1321	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	1050	1
YUV to HEVC	720p	40	16	1	3	100	520	1
YUV to AVC	720p	40	40	0	1	100	1362	1
YUV to HEVC	720p	40	40	0	1	100	1324	1
YUV to AV1	720p	40	40	0	1	100	1043	1
YUV to AVC	720p	40	40	0	2	100	1361	1
YUV to HEVC	720p	40	40	0	2	99	1040	1
YUV to AV1	720p	40	40	0	2	99	854	1
YUV to AVC	720p	40	40	0	3	100	1365	1
YUV to HEVC	720p	40	40	0	3	99	720	1
YUV to AV1	720p	40	40	0	3	100	560	1
YUV to AVC	720p	40	40	1	1	99	1040	1
YUV to HEVC	720p	40	40	1	1	100	1320	1
YUV to AVC	720p	40	40	1	2	99	1040	1
YUV to HEVC	720p	40	40	1	2	100	760	1
YUV to AVC	720p	40	40	1	3	99	1040	1
YUV to HEVC	720p	40	40	1	3	99	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	100	700	2
YUV to HEVC	1080p	20	0	0	1	19	99	820	3
YUV to AV1	1080p	20	0	0	1	19	99	720	2
YUV to AVC	1080p	20	0	0	2	19	99	700	2
YUV to HEVC	1080p	20	0	0	2	19	100	520	2
YUV to AV1	1080p	20	0	0	2	19	100	440	2
YUV to AVC	1080p	20	0	0	3	19	99	701	2
YUV to HEVC	1080p	20	0	0	3	19	100	340	1
YUV to AV1	1080p	20	0	0	3	19	99	280	1
YUV to AVC	1080p	20	0	1	1	19	99	500	2
YUV to HEVC	1080p	20	0	1	1	19	99	660	2
YUV to AVC	1080p	20	0	1	2	19	99	502	2
YUV to HEVC	1080p	20	0	1	2	19	99	360	1
YUV to AVC	1080p	20	0	1	3	19	99	501	2
YUV to HEVC	1080p	20	0	1	3	19	99	240	1
YUV to AVC	1080p	20	4	0	1	19	99	700	2
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	1
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	100	268	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	99	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	100	240	1
YUV to AVC	1080p	20	16	0	1	19	99	700	2
YUV to HEVC	1080p	20	16	0	1	19	99	819	3
YUV to AV1	1080p	20	16	0	1	19	99	720	2
YUV to AVC	1080p	20	16	0	2	19	99	700	3
YUV to HEVC	1080p	20	16	0	2	19	99	501	2
YUV to AV1	1080p	20	16	0	2	19	100	420	2
YUV to AVC	1080p	20	16	0	3	19	99	700	2
YUV to HEVC	1080p	20	16	0	3	19	99	340	1



TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	3	19	99	261	1
YUV to AVC	1080p	20	16	1	1	19	100	500	2
YUV to HEVC	1080p	20	16	1	1	19	99	647	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	99	500	2
YUV to HEVC	1080p	20	16	1	3	19	99	240	1
YUV to AVC	1080p	20	40	0	1	19	99	690	2
YUV to HEVC	1080p	20	40	0	1	19	99	801	3
YUV to AV1	1080p	20	40	0	1	19	100	700	3
YUV to AVC	1080p	20	40	0	2	19	99	692	2
YUV to HEVC	1080p	20	40	0	2	19	99	500	2
YUV to AV1	1080p	20	40	0	2	19	99	420	2
YUV to AVC	1080p	20	40	0	3	19	99	693	2
YUV to HEVC	1080p	20	40	0	3	19	100	340	1
YUV to AV1	1080p	20	40	0	3	19	100	260	1
YUV to AVC	1080p	20	40	1	1	19	99	500	2
YUV to HEVC	1080p	20	40	1	1	19	99	640	2
YUV to AVC	1080p	20	40	1	2	19	99	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	701	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	701	2
YUV to HEVC	1080p	20	0	0	2	23	99	520	2
YUV to AV1	1080p	20	0	0	2	23	99	439	2
YUV to AVC	1080p	20	0	0	3	23	99	700	2
YUV to HEVC	1080p	20	0	0	3	23	100	340	1
YUV to AV1	1080p	20	0	0	3	23	99	280	1
YUV to AVC	1080p	20	0	1	1	23	99	504	2
YUV to HEVC	1080p	20	0	1	1	23	100	660	2
YUV to AVC	1080p	20	0	1	2	23	100	502	2
YUV to HEVC	1080p	20	0	1	2	23	100	360	1
YUV to AVC	1080p	20	0	1	3	23	99	500	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	700	2
YUV to HEVC	1080p	20	4	0	2	23	99	520	2
YUV to AV1	1080p	20	4	0	2	23	99	420	1
YUV to AVC	1080p	20	4	0	3	23	99	700	2
YUV to HEVC	1080p	20	4	0	3	23	100	340	1
YUV to AV1	1080p	20	4	0	3	23	100	268	1
YUV to AVC	1080p	20	4	1	1	23	99	500	2
YUV to HEVC	1080p	20	4	1	1	23	99	660	2
YUV to AVC	1080p	20	4	1	2	23	99	500	2
YUV to HEVC	1080p	20	4	1	2	23	99	360	1
YUV to AVC	1080p	20	4	1	3	23	99	500	2
YUV to HEVC	1080p	20	4	1	3	23	100	240	1
YUV to AVC	1080p	20	16	0	1	23	99	700	2
YUV to HEVC	1080p	20	16	0	1	23	99	820	3
YUV to AV1	1080p	20	16	0	1	23	99	720	2
YUV to AVC	1080p	20	16	0	2	23	99	700	2
YUV to HEVC	1080p	20	16	0	2	23	99	507	2
YUV to AV1	1080p	20	16	0	2	23	99	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	99	265	1
YUV to AVC	1080p	20	16	1	1	23	99	500	2
YUV to HEVC	1080p	20	16	1	1	23	99	644	2
YUV to AVC	1080p	20	16	1	2	23	99	500	2
YUV to HEVC	1080p	20	16	1	2	23	99	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	99	687	2
YUV to HEVC	1080p	20	40	0	1	23	99	803	3
YUV to AV1	1080p	20	40	0	1	23	99	701	3
YUV to AVC	1080p	20	40	0	2	23	99	692	2
YUV to HEVC	1080p	20	40	0	2	23	99	500	2
YUV to AV1	1080p	20	40	0	2	23	99	420	2
YUV to AVC	1080p	20	40	0	3	23	99	691	2
YUV to HEVC	1080p	20	40	0	3	23	100	339	1
YUV to AV1	1080p	20	40	0	3	23	99	260	1
YUV to AVC	1080p	20	40	1	1	23	100	500	2
YUV to HEVC	1080p	20	40	1	1	23	99	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	99	360	1
YUV to AVC	1080p	20	40	1	3	23	99	500	2
YUV to HEVC	1080p	20	40	1	3	23	99	240	1
YUV to AVC	1080p	20	0	0	1	27	99	702	2
YUV to HEVC	1080p	20	0	0	1	27	99	820	3
YUV to AV1	1080p	20	0	0	1	27	99	720	2
YUV to AVC	1080p	20	0	0	2	27	99	700	2
YUV to HEVC	1080p	20	0	0	2	27	99	520	2
YUV to AV1	1080p	20	0	0	2	27	100	440	2
YUV to AVC	1080p	20	0	0	3	27	99	700	2
YUV to HEVC	1080p	20	0	0	3	27	100	340	1
YUV to AV1	1080p	20	0	0	3	27	100	280	1
YUV to AVC	1080p	20	0	1	1	27	99	502	2
YUV to HEVC	1080p	20	0	1	1	27	99	660	2
YUV to AVC	1080p	20	0	1	2	27	99	501	2
YUV to HEVC	1080p	20	0	1	2	27	100	360	1
YUV to AVC	1080p	20	0	1	3	27	99	502	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	2
YUV to HEVC	1080p	20	4	0	2	27	99	519	2
YUV to AV1	1080p	20	4	0	2	27	99	420	1
YUV to AVC	1080p	20	4	0	3	27	99	700	2
YUV to HEVC	1080p	20	4	0	3	27	100	340	1
YUV to AV1	1080p	20	4	0	3	27	100	267	1
YUV to AVC	1080p	20	4	1	1	27	100	500	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	99	501	2
YUV to HEVC	1080p	20	4	1	3	27	100	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	720	2
YUV to AVC	1080p	20	16	0	2	27	99	700	2
YUV to HEVC	1080p	20	16	0	2	27	99	501	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	2	27	99	420	2
YUV to AVC	1080p	20	16	0	3	27	99	700	2
YUV to HEVC	1080p	20	16	0	3	27	100	340	1
YUV to AV1	1080p	20	16	0	3	27	99	261	1
YUV to AVC	1080p	20	16	1	1	27	99	500	2
YUV to HEVC	1080p	20	16	1	1	27	100	644	2
YUV to AVC	1080p	20	16	1	2	27	100	500	2
YUV to HEVC	1080p	20	16	1	2	27	100	360	1
YUV to AVC	1080p	20	16	1	3	27	100	500	2
YUV to HEVC	1080p	20	16	1	3	27	100	240	1
YUV to AVC	1080p	20	40	0	1	27	99	690	2
YUV to HEVC	1080p	20	40	0	1	27	99	801	3
YUV to AV1	1080p	20	40	0	1	27	99	701	2
YUV to AVC	1080p	20	40	0	2	27	99	692	2
YUV to HEVC	1080p	20	40	0	2	27	100	500	2
YUV to AV1	1080p	20	40	0	2	27	100	420	1
YUV to AVC	1080p	20	40	0	3	27	99	690	2
YUV to HEVC	1080p	20	40	0	3	27	100	340	1
YUV to AV1	1080p	20	40	0	3	27	100	260	1
YUV to AVC	1080p	20	40	1	1	27	100	500	2
YUV to HEVC	1080p	20	40	1	1	27	99	640	2
YUV to AVC	1080p	20	40	1	2	27	100	500	2
YUV to HEVC	1080p	20	40	1	2	27	100	360	1
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	289	12	14	20	5
AVC to HEVC	1	308	12	14	21	5
AVC to AV1	1	263	12	13	20	4
HEVC to AVC	1	289	15	12	21	5
HEVC to HEVC	1	307	15	13	20	5
HEVC to AV1	1	263	15	12	21	4
VP9 to AVC	1	288	11	16	20	4
VP9 to HEVC	1	306	11	18	20	5
VP9 to AV1	1	262	12	15	20	4
AVC to AVC	16	1044	3	72	94	24
AVC to HEVC	16	1156	3	78	93	27
AVC to AV1	16	1110	3	72	94	26
HEVC to AVC	16	1088	4	72	94	25
HEVC to HEVC	16	1215	4	78	94	28
HEVC to AV1	16	1136	4	69	95	26
VP9 to AVC	16	1040	3	83	95	24
VP9 to HEVC	16	1168	3	92	95	27
VP9 to AV1	16	1120	3	85	95	26
AVC to AVC	32	960	1	75	95	24
AVC to HEVC	32	1085	1	81	94	26
AVC to AV1	32	1056	1	76	93	25
HEVC to AVC	32	994	1	73	95	25
HEVC to HEVC	32	1141	2	80	94	28
HEVC to AV1	32	1089	2	73	94	26
VP9 to AVC	32	992	1	80	93	23
VP9 to HEVC	32	1120	1	90	95	27
VP9 to AV1	32	1088	1	85	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	2.3	0.5	1	2.14	0
P2H	720p	1	1001	30	2.4	0	1	2.1	0.01
P2A	720p	180	180180	29.7	0.76	46	56	2.33	0.11
P2H	720p	180	180180	29.7	0.76	45	56	2.3	0.09
P2A	1080p	1	1001	30	1.3	0.5	0.75	3.8	0.02
P2H	1080p	1	1001	29	1.3	0.5	0.75	3.59	0.01
P2A	1080p	80	80080	30	0.88	46	50	3.96	0.29
P2H	1080p	80	80080	30	0.86	43	50	3.76	0.25

## 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	250
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	2351.5
mobilenetv2_nchw_keras_96x160	8	10000	rgb	nchw	image	96x160x3	2345.88
mobilenetv2_nchw_keras_96x160	16	10000	rgb	nchw	image	96x160x3	2342.44

## 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	79
ROI	network_binary_yolov4_head	1920x1080	8	22
ROI	network_binary_yolov4_head	1920x1080	32	5
BG	segm32_tflite_nchw_bgr	1920x1080	1	60
BG	segm32_tflite_nchw_bgr	1920x1080	8	39
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	66
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.

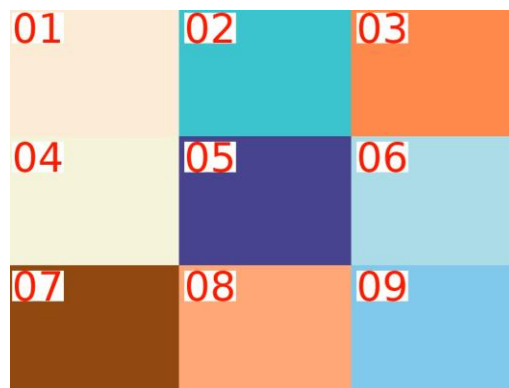
The encoded bitstream is read out through PCIe and written into an output file.

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	141.57	54	63	13	14
1920x1080	4x4	1920x1080	480x270	97.96	78	80	10	13
1920x1080	4x8	1920x1080	480x135	55.13	99	88	5	8
1920x1080	3x3	3840x2160	1280x720	69.5	26	32	22	18
1920x1080	4x4	3840x2160	960x540	54.32	40	47	19	20
1920x1080	4x8	3840x2160	960x270	40.68	77	69	14	16
1920x1080	3x3	7680x4320	2560x1440	19.8	15	8	23	15
1920x1080	4x4	7680x4320	1920x1080	19.07	20	15	22	16
1920x1080	4x8	7680x4320	1920x540	19.08	34	34	22	19
1920x1080	6x6	1920x1080	320x180	49	103	90	5	10
1920x1080	7x7	1920x1080	274x154 276x154 274x156 276x156*	36.09	109	89	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	19.02	126	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	89	0	100	12
HEVC to YUV	8k	1	0	8	1	96	0	97	18
VP9 to YUV	8k	1	0	8	1	25	0	30	4
YUV to AVC	8k	1	0	8	1	0	96	58	78
YUV to HEVC	8k	1	0	8	1	0	96	72	98
AVC to AVC	8k	1	1	8	1	73	99	47	3
AVC to HEVC	8k	1	1	8	1	76	99	63	4
HEVC to AVC	8k	1	1	8	1	62	100	47	4
HEVC to HEVC	8k	1	1	8	1	72	100	63	10
VP9 to AVC	8k	1	1	8	1	25	42	29	3
VP9 to HEVC	8k	1	1	8	1	25	41	30	4
AVC to YUV	8k	1	0	10	1	97	0	58	8
HEVC to YUV	8k	1	0	10	1	62	0	51	6
VP9 to YUV	8k	1	0	10	1	24	0	28	7
YUV to AVC	8k	1	0	10	1	0	88	42	141
YUV to HEVC	8k	1	0	10	1	0	56	42	124
AVC to YUV	4k	1	0	8	1	61	0	289	15
HEVC to YUV	4k	1	0	8	1	53	0	286	21
VP9 to YUV	4k	1	0	8	1	24	0	110	7
AVC to YUV	4k	16	0	8	0	98	0	422	1
HEVC to YUV	4k	16	0	8	0	99	0	452	1
VP9 to YUV	4k	16	0	8	0	99	0	401	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	34
YUV to AVC	4k	4	0	8	0	0	97	263	15
YUV to HEVC	4k	4	0	8	0	0	96	292	12
YUV to AV1	4k	4	0	8	0	0	97	248	11
YUV to AVC	4k	8	0	8	0	0	100	280	7
YUV to HEVC	4k	8	0	8	0	0	99	296	7
YUV to AV1	4k	8	0	8	0	0	99	256	5
AVC to AVC	4k	1	1	8	1	71	96	201	14
AVC to HEVC	4k	1	1	8	1	69	91	247	12
AVC to AV1	4k	1	1	8	1	61	94	234	15
HEVC to AVC	4k	1	1	8	1	57	97	202	16
HEVC to HEVC	4k	1	1	8	1	58	95	246	9
HEVC to AV1	4k	1	1	8	1	49	93	232	16
VP9 to AVC	4k	1	1	8	1	24	39	108	3
VP9 to HEVC	4k	1	1	8	1	24	37	109	4

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	6
AVC to AVC	4k	4	1	8	0	62	97	220	3
AVC to HEVC	4k	4	1	8	0	66	96	272	3
AVC to AV1	4k	4	1	8	0	56	95	244	8
HEVC to AVC	4k	4	1	8	0	54	96	216	3
HEVC to HEVC	4k	4	1	8	0	56	95	264	3
HEVC to AV1	4k	4	1	8	0	46	95	244	8
VP9 to AVC	4k	4	1	8	0	65	97	216	3
VP9 to HEVC	4k	4	1	8	0	70	96	264	3
VP9 to AV1	4k	4	1	8	0	61	91	224	3
AVC to AVC	4k	8	1	8	0	68	100	195	1
AVC to HEVC	4k	8	1	8	0	76	99	255	2
AVC to AV1	4k	8	1	8	0	69	99	240	7
HEVC to AVC	4k	8	1	8	0	61	100	195	2
HEVC to HEVC	4k	8	1	8	0	65	99	256	2
HEVC to AV1	4k	8	1	8	0	55	99	240	8
VP9 to AVC	4k	8	1	8	0	67	100	197	2
VP9 to HEVC	4k	8	1	8	0	74	99	256	1
VP9 to AV1	4k	8	1	8	0	63	99	248	6
AVC to YUV	4k	1	0	10	1	50	0	211	12
HEVC to YUV	4k	1	0	10	1	53	0	209	8
VP9 to YUV	4k	1	0	10	1	24	0	106	6
AVC to YUV	4k	16	0	10	0	100	0	254	0
HEVC to YUV	4k	16	0	10	0	99	0	253	0
VP9 to YUV	4k	16	0	10	0	98	0	235	0
YUV to AVC	4k	1	0	10	1	0	71	180	58
YUV to HEVC	4k	1	0	10	1	0	66	197	70
YUV to AV1	4k	1	0	10	1	0	73	188	63
YUV to AVC	4k	4	0	10	0	0	90	196	27
YUV to HEVC	4k	4	0	10	0	0	82	232	39
YUV to AV1	4k	4	0	10	0	0	89	225	40
AVC to YUV	1080p	1	0	8	1	48	0	793	19
HEVC to YUV	1080p	1	0	8	1	52	0	760	25
VP9 to YUV	1080p	1	0	8	1	22	0	456	3
AVC to YUV	1080p	40	0	8	0	98	0	1578	2
HEVC to YUV	1080p	40	0	8	0	98	0	1549	2
VP9 to YUV	1080p	40	0	8	0	94	0	1690	0
YUV to AVC	1080p	1	0	8	1	0	59	662	27
YUV to HEVC	1080p	1	0	8	1	0	55	664	25
YUV to AV1	1080p	1	0	8	1	0	59	610	31

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	2
YUV to HEVC	1080p	32	0	8	0	0	99	1186	2
YUV to AV1	1080p	32	0	8	0	0	99	1024	2
AVC to AVC	1080p	1	1	8	1	67	81	879	19
AVC to HEVC	1080p	1	1	8	1	67	80	928	27
AVC to AV1	1080p	1	1	8	1	54	82	798	15
HEVC to AVC	1080p	1	1	8	1	61	78	824	25
HEVC to HEVC	1080p	1	1	8	1	63	76	876	27
HEVC to AV1	1080p	1	1	8	1	53	78	760	30
VP9 to AVC	1080p	1	1	8	1	22	40	456	9
VP9 to HEVC	1080p	1	1	8	1	22	38	454	7
VP9 to AV1	1080p	1	1	8	1	22	44	454	6
AVC to AVC	1080p	32	1	8	0	76	100	864	0
AVC to HEVC	1080p	32	1	8	0	85	99	974	1
AVC to AV1	1080p	32	1	8	0	78	99	928	2
HEVC to AVC	1080p	32	1	8	0	72	100	896	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1007	1
HEVC to AV1	1080p	32	1	8	0	71	99	939	3
VP9 to AVC	1080p	32	1	8	0	70	99	992	0
VP9 to HEVC	1080p	32	1	8	0	74	99	1106	0
VP9 to AV1	1080p	32	1	8	0	69	99	992	1
AVC to YUV	1080p	1	0	10	1	30	0	569	6
HEVC to YUV	1080p	1	0	10	1	28	0	569	7
VP9 to YUV	1080p	1	0	10	1	22	0	363	8
AVC to YUV	1080p	40	0	10	0	70	0	1080	0
HEVC to YUV	1080p	40	0	10	0	84	0	1080	0
VP9 to YUV	1080p	40	0	10	0	92	0	1040	0
YUV to AVC	1080p	1	0	10	1	0	39	449	37
YUV to HEVC	1080p	1	0	10	1	0	37	454	38
YUV to AV1	1080p	1	0	10	1	0	40	421	36
YUV to AVC	1080p	32	0	10	0	0	71	808	7
YUV to HEVC	1080p	32	0	10	0	0	69	817	6
YUV to AV1	1080p	32	0	10	0	0	78	801	7
AVC to YUV	720p	1	0	8	1	45	0	1131	16
HEVC to YUV	720p	1	0	8	1	39	0	1124	20
VP9 to YUV	720p	1	0	8	1	29	0	863	4
AVC to YUV	720p	100	0	8	0	100	0	2401	0
HEVC to YUV	720p	100	0	8	0	97	0	2708	0
VP9 to YUV	720p	100	0	8	0	100	0	2502	0
YUV to AVC	720p	1	0	8	1	0	33	872	16

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	34	873	19
YUV to AV1	720p	1	0	8	1	0	35	785	16
YUV to AVC	720p	64	0	8	0	0	99	2301	2
YUV to HEVC	720p	64	0	8	0	0	99	2344	2
YUV to AV1	720p	64	0	8	0	0	98	1920	1
AVC to AVC	720p	1	1	8	1	48	48	1204	20
AVC to HEVC	720p	1	1	8	1	49	48	1223	20
AVC to AV1	720p	1	1	8	1	41	51	1054	16
HEVC to AVC	720p	1	1	8	1	41	46	1171	25
HEVC to HEVC	720p	1	1	8	1	41	46	1172	21
HEVC to AV1	720p	1	1	8	1	35	49	1018	19
VP9 to AVC	720p	1	1	8	1	28	33	861	12
VP9 to HEVC	720p	1	1	8	1	28	33	856	9
VP9 to AV1	720p	1	1	8	1	28	40	857	9
AVC to AVC	720p	64	1	8	0	95	100	1861	1
AVC to HEVC	720p	64	1	8	0	96	100	1947	0
AVC to AV1	720p	64	1	8	0	77	100	1665	0
HEVC to AVC	720p	64	1	8	0	85	100	1870	1
HEVC to HEVC	720p	64	1	8	0	86	100	1963	1
HEVC to AV1	720p	64	1	8	0	67	100	1667	0
VP9 to AVC	720p	64	1	8	0	98	100	2112	0
VP9 to HEVC	720p	64	1	8	0	97	100	2176	0
VP9 to AV1	720p	64	1	8	0	78	100	1792	0

## 15. T1U – Libxcodec 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: Libxcodec 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: Libxcodec 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	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	80	0	87	7
HEVC to YUV	8k	1	0	8	1	73	0	89	4
VP9 to YUV	8k	1	0	8	1	25	0	30	2
YUV to AVC	8k	1	0	8	1	0	94	58	25
YUV to HEVC	8k	1	0	8	1	0	97	72	31
AVC to AVC	8k	1	1	8	1	71	100	48	3
AVC to HEVC	8k	1	1	8	1	81	99	64	4
HEVC to AVC	8k	1	1	8	1	60	100	46	2
HEVC to HEVC	8k	1	1	8	1	71	100	63	3
VP9 to AVC	8k	1	1	8	1	24	42	29	1
VP9 to HEVC	8k	1	1	8	1	23	38	30	1
AVC to YUV	8k	1	0	10	1	57	0	50	5
HEVC to YUV	8k	1	0	10	1	54	0	47	4
VP9 to YUV	8k	1	0	10	1	24	0	28	3
YUV to AVC	8k	1	0	10	1	0	64	36	35
YUV to HEVC	8k	1	0	10	1	0	50	37	37
AVC to YUV	4k	1	0	8	1	69	0	318	8
HEVC to YUV	4k	1	0	8	1	55	0	301	4
VP9 to YUV	4k	1	0	8	1	24	0	109	2
AVC to YUV	4k	16	0	8	0	98	0	425	0
HEVC to YUV	4k	16	0	8	0	99	0	456	0
VP9 to YUV	4k	16	0	8	0	98	0	406	0
YUV to AVC	4k	1	0	8	1	0	92	251	20
YUV to HEVC	4k	1	0	8	1	0	94	283	22
YUV to AV1	4k	1	0	8	1	0	94	247	21
YUV to AVC	4k	4	0	8	0	0	97	274	8
YUV to HEVC	4k	4	0	8	0	0	97	292	8
YUV to AV1	4k	4	0	8	0	0	97	252	8
YUV to AVC	4k	8	0	8	0	0	99	285	5
YUV to HEVC	4k	8	0	8	0	0	99	302	6
YUV to AV1	4k	8	0	8	0	0	100	263	5
AVC to AVC	4k	1	1	8	0	67	97	204	4
AVC to HEVC	4k	1	1	8	0	72	97	251	6
AVC to AV1	4k	1	1	8	0	63	97	235	6
HEVC to AVC	4k	1	1	8	0	57	97	204	4
HEVC to HEVC	4k	1	1	8	0	62	97	256	4
HEVC to AV1	4k	1	1	8	0	54	97	236	5
VP9 to AVC	4k	1	1	8	0	24	38	109	2

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	0	23	36	109	2
VP9 to AV1	4k	1	1	8	0	24	42	109	2
AVC to AVC	4k	4	1	8	0	62	97	225	4
AVC to HEVC	4k	4	1	8	0	66	95	274	4
AVC to AV1	4k	4	1	8	0	57	95	245	4
HEVC to AVC	4k	4	1	8	0	54	96	219	3
HEVC to HEVC	4k	4	1	8	0	57	96	266	3
HEVC to AV1	4k	4	1	8	0	49	95	242	4
VP9 to AVC	4k	4	1	8	0	64	96	219	3
VP9 to HEVC	4k	4	1	8	0	68	97	268	3
VP9 to AV1	4k	4	1	8	0	60	94	243	3
AVC to AVC	4k	8	1	8	0	67	99	198	2
AVC to HEVC	4k	8	1	8	0	74	99	255	3
AVC to AV1	4k	8	1	8	0	64	100	246	3
HEVC to AVC	4k	8	1	8	0	62	99	198	2
HEVC to HEVC	4k	8	1	8	0	63	99	254	2
HEVC to AV1	4k	8	1	8	0	56	99	244	3
VP9 to AVC	4k	8	1	8	0	66	99	203	3
VP9 to HEVC	4k	8	1	8	0	76	100	261	2
VP9 to AV1	4k	8	1	8	0	66	100	249	3
AVC to YUV	4k	1	0	10	1	50	0	195	5
HEVC to YUV	4k	1	0	10	1	43	0	190	4
VP9 to YUV	4k	1	0	10	1	24	0	105	3
AVC to YUV	4k	16	0	10	0	99	0	258	0
HEVC to YUV	4k	16	0	10	0	100	0	257	0
VP9 to YUV	4k	16	0	10	0	100	0	238	0
YUV to AVC	4k	1	0	10	1	0	53	150	25
YUV to HEVC	4k	1	0	10	1	0	49	152	25
YUV to AV1	4k	1	0	10	1	0	55	145	24
YUV to AVC	4k	4	0	10	0	0	74	195	24
YUV to HEVC	4k	4	0	10	0	0	65	201	26
YUV to AV1	4k	4	0	10	0	0	75	183	29
AVC to YUV	1080p	1	0	8	1	49	0	785	14
HEVC to YUV	1080p	1	0	8	1	57	0	829	8
VP9 to YUV	1080p	1	0	8	1	22	0	456	2
AVC to YUV	1080p	40	0	8	0	98	0	1599	0
HEVC to YUV	1080p	40	0	8	0	99	0	1586	0
VP9 to YUV	1080p	40	0	8	0	87	0	1657	0
YUV to AVC	1080p	1	0	8	1	0	59	665	13



TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	57	689	15
YUV to AV1	1080p	1	0	8	1	0	54	563	12
YUV to AVC	1080p	32	0	8	0	0	99	1155	1
YUV to HEVC	1080p	32	0	8	0	0	99	1225	2
YUV to AV1	1080p	32	0	8	0	0	99	1056	1
AVC to AVC	1080p	1	1	8	0	70	85	919	14
AVC to HEVC	1080p	1	1	8	0	70	84	975	14
AVC to AV1	1080p	1	1	8	0	56	84	824	15
HEVC to AVC	1080p	1	1	8	0	68	87	910	12
HEVC to HEVC	1080p	1	1	8	0	71	85	971	13
HEVC to AV1	1080p	1	1	8	0	57	86	824	14
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	4
AVC to AVC	1080p	32	1	8	0	75	99	896	1
AVC to HEVC	1080p	32	1	8	0	83	99	1011	0
AVC to AV1	1080p	32	1	8	0	76	99	965	0
HEVC to AVC	1080p	32	1	8	0	72	99	930	0
HEVC to HEVC	1080p	32	1	8	0	77	99	1043	0
HEVC to AV1	1080p	32	1	8	0	69	99	974	1
VP9 to AVC	1080p	32	1	8	0	69	99	1019	0
VP9 to HEVC	1080p	32	1	8	0	74	99	1141	0
VP9 to AV1	1080p	32	1	8	0	69	99	1026	0
AVC to YUV	1080p	1	0	10	1	33	0	612	4
HEVC to YUV	1080p	1	0	10	1	30	0	608	4
VP9 to YUV	1080p	1	0	10	1	22	0	362	2
AVC to YUV	1080p	40	0	10	0	75	0	1067	0
HEVC to YUV	1080p	40	0	10	0	82	0	1067	0
VP9 to YUV	1080p	40	0	10	0	78	0	1009	0
YUV to AVC	1080p	1	0	10	1	0	35	400	18
YUV to HEVC	1080p	1	0	10	1	0	34	416	18
YUV to AV1	1080p	1	0	10	1	0	36	371	16
YUV to AVC	1080p	32	0	10	0	0	75	867	7
YUV to HEVC	1080p	32	0	10	0	0	71	878	7
YUV to AV1	1080p	32	0	10	0	0	77	810	8
AVC to YUV	720p	1	0	8	1	43	0	1144	13
HEVC to YUV	720p	1	0	8	1	42	0	1215	7
VP9 to YUV	720p	1	0	8	1	28	0	863	4
AVC to YUV	720p	100	0	8	0	100	0	2592	0

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	89	0	2686	0
VP9 to YUV	720p	100	0	8	0	93	0	2593	0
YUV to AVC	720p	1	0	8	1	0	38	978	11
YUV to HEVC	720p	1	0	8	1	0	38	980	10
YUV to AV1	720p	1	0	8	1	0	32	728	9
YUV to AVC	720p	64	0	8	0	0	100	2562	2
YUV to HEVC	720p	64	0	8	0	0	100	2636	2
YUV to AV1	720p	64	0	8	0	0	100	2125	1
AVC to AVC	720p	1	1	8	0	51	50	1268	14
AVC to HEVC	720p	1	1	8	0	51	50	1282	14
AVC to AV1	720p	1	1	8	0	44	50	1043	13
HEVC to AVC	720p	1	1	8	0	46	51	1284	11
HEVC to HEVC	720p	1	1	8	0	46	51	1290	10
HEVC to AV1	720p	1	1	8	0	39	49	1056	11
VP9 to AVC	720p	1	1	8	0	29	33	861	5
VP9 to HEVC	720p	1	1	8	0	29	33	854	5
VP9 to AV1	720p	1	1	8	0	28	39	854	7
AVC to AVC	720p	64	1	8	0	100	100	2004	0
AVC to HEVC	720p	64	1	8	0	100	100	2091	0
AVC to AV1	720p	64	1	8	0	95	100	1758	0
HEVC to AVC	720p	64	1	8	0	98	100	1997	0
HEVC to HEVC	720p	64	1	8	0	98	100	2087	0
HEVC to AV1	720p	64	1	8	0	90	100	1760	0
VP9 to AVC	720p	64	1	8	0	100	100	2264	0
VP9 to HEVC	720p	64	1	8	0	100	100	2341	0
VP9 to AV1	720p	64	1	8	0	100	100	1911	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	65.34	66.18	64.65	0.08
YUV to HEVC	8k	1	62.54	68.09	60.92	1.24
YUV to AVC	4k	1	17.42	19.77	16.95	0.08
YUV to HEVC	4k	1	18.86	21.22	17.8	0.13
YUV to AV1	4k	1	24.8	28.9	18.6	0.85
YUV to AVC	4k	4	19.09	22.41	16.86	1.13
YUV to HEVC	4k	4	20.22	22.67	18.95	0.39
YUV to AV1	4k	4	25.52	31.1	19.75	1.29
YUV to AVC	4k	8	24.42	35.21	17.05	28.89
YUV to HEVC	4k	8	26.81	40.79	18.95	43.19
YUV to AV1	4k	8	42.83	51.23	29.88	7.28
YUV to AVC	1080p	1	5.3	6.31	5	0.01
YUV to HEVC	1080p	1	5.57	6.36	5.2	0.02
YUV to AV1	1080p	1	7.57	8.4	5.67	0.07
YUV to AVC	1080p	32	8.68	15.9	5.69	2.58
YUV to HEVC	1080p	32	10.25	15.38	5.81	2.77
YUV to AV1	1080p	32	47.14	54.56	38.84	4.23
YUV to AVC	720p	1	3.09	3.82	2.79	0.02
YUV to HEVC	720p	1	3.15	3.48	2.81	0
YUV to AV1	720p	1	4.13	4.51	3.28	0.01
YUV to AVC	720p	64	5.89	11.98	3.27	1.21
YUV to HEVC	720p	64	6.46	11.65	3.73	1.1
YUV to AV1	720p	64	45.71	52.9	38.82	2.61

## 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	92	0	116	4
HEVC to RGBA	8k	1	90	0	120	5
VP9 to RGBA	8k	1	23	0	31	1
AVC to RGBA	4k	1	22	0	114	6
AVC to RGBA	4k	16	93	2	467	1
HEVC to RGBA	4k	1	22	0	137	7
HEVC to RGBA	4k	16	95	3	551	2
VP9 to RGBA	4k	1	22	0	112	4
VP9 to RGBA	4k	16	93	2	463	1
AVC to RGBA	1080p	40	95	13	1600	1
HEVC to RGBA	1080p	40	94	11	1596	1
VP9 to RGBA	1080p	40	92	15	2059	0
AVC to RGBA	720p	100	96	18	2421	0
HEVC to RGBA	720p	100	93	20	2702	0
VP9 to RGBA	720p	64	96	19	2624	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	384	3
AVC to HEVC	8	29	95	2	411	3
AVC to AV1	8	23	95	1	352	3
HEVC to AVC	8	31	94	2	392	4
HEVC to HEVC	8	31	94	2	411	4
HEVC to AV1	8	23	94	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	95	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	57	150	69
RGBA to HEVC	4k	1	1	51	150	73
RGBA to AV1	4k	1	1	60	153	69
RGBA to AVC	4k	4	0	74	152	37
RGBA to HEVC	4k	4	0	65	159	40
RGBA to AV1	4k	4	0	67	160	44
RGBA to AVC	4k	8	0	65	163	33
RGBA to HEVC	4k	8	0	61	167	35
RGBA to AV1	4k	8	0	69	168	33
RGBA to AVC	1080p	1	1	35	398	49
RGBA to HEVC	1080p	1	1	34	405	48
RGBA to AV1	1080p	1	1	37	388	46
RGBA to AVC	1080p	16	0	62	604	22
RGBA to HEVC	1080p	16	0	59	609	20
RGBA to AV1	1080p	16	0	64	608	19
RGBA to AVC	1080p	32	0	63	641	12
RGBA to HEVC	1080p	32	0	59	642	9
RGBA to AV1	1080p	32	0	65	641	12
RGBA to AVC	720p	1	1	29	677	52
RGBA to HEVC	720p	1	1	29	683	57
RGBA to AV1	720p	1	1	36	671	58
RGBA to AVC	720p	16	0	51	1237	39
RGBA to HEVC	720p	16	0	51	1250	40
RGBA to AV1	720p	16	0	58	1179	35
RGBA to AVC	720p	32	0	49	1179	33
RGBA to HEVC	720p	32	0	47	1171	32
RGBA to AV1	720p	32	0	58	1176	30



## 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	95	266	15
YUV to HEVC	4k	4	0	0	1	97	292	12
YUV to AV1	4k	4	0	0	1	97	252	10
YUV to AVC	4k	4	0	0	2	96	268	14
YUV to HEVC	4k	4	0	0	2	98	148	6
YUV to AV1	4k	4	0	0	2	97	120	5
YUV to AVC	4k	4	0	0	3	95	264	18
YUV to HEVC	4k	4	0	0	3	99	88	4
YUV to AV1	4k	4	0	0	3	100	64	3
YUV to AVC	4k	4	0	1	1	98	164	7
YUV to HEVC	4k	4	0	1	1	97	208	13
YUV to AVC	4k	4	0	1	2	98	164	6
YUV to HEVC	4k	4	0	1	2	99	96	3
YUV to AVC	4k	4	0	1	3	98	164	6
YUV to HEVC	4k	4	0	1	3	99	60	3
YUV to AVC	4k	4	4	0	1	100	172	7
YUV to HEVC	4k	4	4	0	1	99	199	8
YUV to AV1	4k	4	4	0	1	100	172	9
YUV to AVC	4k	4	4	0	2	100	172	7
YUV to HEVC	4k	4	4	0	2	99	120	6
YUV to AV1	4k	4	4	0	2	99	100	4
YUV to AVC	4k	4	4	0	3	100	172	7
YUV to HEVC	4k	4	4	0	3	99	76	5
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	100	156	7
YUV to AVC	4k	4	4	1	2	99	120	4
YUV to HEVC	4k	4	4	1	2	100	84	4
YUV to AVC	4k	4	4	1	3	100	120	5
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	99	196	9
YUV to AV1	4k	4	16	0	1	99	172	7
YUV to AVC	4k	4	16	0	2	100	172	9
YUV to HEVC	4k	4	16	0	2	100	120	6
YUV to AV1	4k	4	16	0	2	100	100	5
YUV to AVC	4k	4	16	0	3	99	172	8
YUV to HEVC	4k	4	16	0	3	100	76	3
YUV to AV1	4k	4	16	0	3	100	64	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	16	1	1	100	120	5
YUV to HEVC	4k	4	16	1	1	99	156	7
YUV to AVC	4k	4	16	1	2	100	120	5
YUV to HEVC	4k	4	16	1	2	100	80	4
YUV to AVC	4k	4	16	1	3	99	120	6
YUV to HEVC	4k	4	16	1	3	99	56	3
YUV to AVC	4k	4	40	0	1	100	170	9
YUV to HEVC	4k	4	40	0	1	100	196	9
YUV to AV1	4k	4	40	0	1	100	172	8
YUV to AVC	4k	4	40	0	2	100	171	8
YUV to HEVC	4k	4	40	0	2	99	116	7
YUV to AV1	4k	4	40	0	2	99	100	4
YUV to AVC	4k	4	40	0	3	99	170	7
YUV to HEVC	4k	4	40	0	3	100	76	4
YUV to AV1	4k	4	40	0	3	100	60	3
YUV to AVC	4k	4	40	1	1	99	120	5
YUV to HEVC	4k	4	40	1	1	99	153	7
YUV to AVC	4k	4	40	1	2	99	120	6
YUV to HEVC	4k	4	40	1	2	100	80	3
YUV to AVC	4k	4	40	1	3	99	120	6
YUV to HEVC	4k	4	40	1	3	98	56	3
YUV to AVC	1080p	20	0	0	1	99	1120	3
YUV to HEVC	1080p	20	0	0	1	99	1180	3
YUV to AV1	1080p	20	0	0	1	99	1020	3
YUV to AVC	1080p	20	0	0	2	99	1120	3
YUV to HEVC	1080p	20	0	0	2	99	600	1
YUV to AV1	1080p	20	0	0	2	99	480	1
YUV to AVC	1080p	20	0	0	3	99	1120	3
YUV to HEVC	1080p	20	0	0	3	100	360	1
YUV to AV1	1080p	20	0	0	3	99	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	99	660	2
YUV to HEVC	1080p	20	0	1	3	99	240	1
YUV to AVC	1080p	20	4	0	1	99	614	2
YUV to HEVC	1080p	20	4	0	1	99	718	2
YUV to AV1	1080p	20	4	0	1	99	621	2
YUV to AVC	1080p	20	4	0	2	99	612	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	4	0	2	99	440	1
YUV to AV1	1080p	20	4	0	2	99	373	1
YUV to AVC	1080p	20	4	0	3	99	617	2
YUV to HEVC	1080p	20	4	0	3	99	300	1
YUV to AV1	1080p	20	4	0	3	100	240	1
YUV to AVC	1080p	20	4	1	1	99	440	1
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	99	320	1
YUV to AVC	1080p	20	4	1	3	99	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	713	2
YUV to AV1	1080p	20	16	0	1	99	620	2
YUV to AVC	1080p	20	16	0	2	99	605	1
YUV to HEVC	1080p	20	16	0	2	100	440	1
YUV to AV1	1080p	20	16	0	2	99	366	1
YUV to AVC	1080p	20	16	0	3	99	603	2
YUV to HEVC	1080p	20	16	0	3	99	295	1
YUV to AV1	1080p	20	16	0	3	99	240	1
YUV to AVC	1080p	20	16	1	1	100	440	1
YUV to HEVC	1080p	20	16	1	1	99	560	2
YUV to AVC	1080p	20	16	1	2	99	440	1
YUV to HEVC	1080p	20	16	1	2	99	317	1
YUV to AVC	1080p	20	16	1	3	99	440	1
YUV to HEVC	1080p	20	16	1	3	100	220	1
YUV to AVC	1080p	20	40	0	1	99	600	2
YUV to HEVC	1080p	20	40	0	1	99	701	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	100	360	1
YUV to AVC	1080p	20	40	0	3	99	600	2
YUV to HEVC	1080p	20	40	0	3	100	284	1
YUV to AV1	1080p	20	40	0	3	99	232	1
YUV to AVC	1080p	20	40	1	1	99	440	2
YUV to HEVC	1080p	20	40	1	1	99	560	2
YUV to AVC	1080p	20	40	1	2	100	440	1
YUV to HEVC	1080p	20	40	1	2	100	300	1
YUV to AVC	1080p	20	40	1	3	99	440	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	40	1	3	100	202	1
YUV to AVC	720p	40	0	0	1	97	2319	3
YUV to HEVC	720p	40	0	0	1	95	2360	3
YUV to AV1	720p	40	0	0	1	96	1972	2
YUV to AVC	720p	40	0	0	2	96	2306	3
YUV to HEVC	720p	40	0	0	2	99	1320	1
YUV to AV1	720p	40	0	0	2	99	1046	1
YUV to AVC	720p	40	0	0	3	96	2322	3
YUV to HEVC	720p	40	0	0	3	99	800	0
YUV to AV1	720p	40	0	0	3	99	560	0
YUV to AVC	720p	40	0	1	1	99	1480	1
YUV to HEVC	720p	40	0	1	1	99	1800	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	522	1
YUV to AVC	720p	40	4	0	1	99	1244	1
YUV to HEVC	720p	40	4	0	1	100	1319	1
YUV to AV1	720p	40	4	0	1	100	974	1
YUV to AVC	720p	40	4	0	2	99	1247	1
YUV to HEVC	720p	40	4	0	2	99	920	1
YUV to AV1	720p	40	4	0	2	99	760	0
YUV to AVC	720p	40	4	0	3	99	1245	1
YUV to HEVC	720p	40	4	0	3	99	640	1
YUV to AV1	720p	40	4	0	3	99	480	0
YUV to AVC	720p	40	4	1	1	99	920	0
YUV to HEVC	720p	40	4	1	1	99	1160	1
YUV to AVC	720p	40	4	1	2	99	920	1
YUV to HEVC	720p	40	4	1	2	99	660	0
YUV to AVC	720p	40	4	1	3	99	920	1
YUV to HEVC	720p	40	4	1	3	99	443	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	1039	1
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	1240	1
YUV to HEVC	720p	40	16	0	3	99	640	0
YUV to AV1	720p	40	16	0	3	100	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	640	0
YUV to AVC	720p	40	16	1	3	99	920	0
YUV to HEVC	720p	40	16	1	3	100	440	0
YUV to AVC	720p	40	40	0	1	99	1239	1
YUV to HEVC	720p	40	40	0	1	100	1280	1
YUV to AV1	720p	40	40	0	1	100	1000	0
YUV to AVC	720p	40	40	0	2	99	1240	1
YUV to HEVC	720p	40	40	0	2	99	920	1
YUV to AV1	720p	40	40	0	2	99	759	0
YUV to AVC	720p	40	40	0	3	99	1240	1
YUV to HEVC	720p	40	40	0	3	99	626	0
YUV to AV1	720p	40	40	0	3	99	480	0
YUV to AVC	720p	40	40	1	1	99	919	0
YUV to HEVC	720p	40	40	1	1	99	1146	1
YUV to AVC	720p	40	40	1	2	99	920	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	617	2
YUV to HEVC	1080p	20	0	0	1	19	99	716	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	617	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	99	440	1
YUV to HEVC	1080p	20	0	1	1	19	99	574	1
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	99	220	1
YUV to AVC	1080p	20	4	0	1	19	99	614	2
YUV to HEVC	1080p	20	4	0	1	19	99	719	3
YUV to AV1	1080p	20	4	0	1	19	99	621	2
YUV to AVC	1080p	20	4	0	2	19	99	619	2
YUV to HEVC	1080p	20	4	0	2	19	99	440	1
YUV to AV1	1080p	20	4	0	2	19	100	369	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	3	19	99	616	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	99	440	1
YUV to HEVC	1080p	20	4	1	1	19	99	567	2
YUV to AVC	1080p	20	4	1	2	19	99	440	1
YUV to HEVC	1080p	20	4	1	2	19	99	320	1
YUV to AVC	1080p	20	4	1	3	19	99	440	1
YUV to HEVC	1080p	20	4	1	3	19	100	220	1
YUV to AVC	1080p	20	16	0	1	19	99	603	2
YUV to HEVC	1080p	20	16	0	1	19	99	711	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	360	1
YUV to AVC	1080p	20	16	0	3	19	99	602	2
YUV to HEVC	1080p	20	16	0	3	19	99	300	1
YUV to AV1	1080p	20	16	0	3	19	100	237	1
YUV to AVC	1080p	20	16	1	1	19	99	440	1
YUV to HEVC	1080p	20	16	1	1	19	99	560	1
YUV to AVC	1080p	20	16	1	2	19	99	440	1
YUV to HEVC	1080p	20	16	1	2	19	100	312	1
YUV to AVC	1080p	20	16	1	3	19	99	440	1
YUV to HEVC	1080p	20	16	1	3	19	99	220	1
YUV to AVC	1080p	20	40	0	1	19	99	600	2
YUV to HEVC	1080p	20	40	0	1	19	99	700	2
YUV to AV1	1080p	20	40	0	1	19	99	619	2
YUV to AVC	1080p	20	40	0	2	19	99	600	2
YUV to HEVC	1080p	20	40	0	2	19	99	440	1
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	286	1
YUV to AV1	1080p	20	40	0	3	19	99	228	1
YUV to AVC	1080p	20	40	1	1	19	99	438	1
YUV to HEVC	1080p	20	40	1	1	19	99	560	2
YUV to AVC	1080p	20	40	1	2	19	99	440	1
YUV to HEVC	1080p	20	40	1	2	19	99	302	1
YUV to AVC	1080p	20	40	1	3	19	99	440	1
YUV to HEVC	1080p	20	40	1	3	19	100	203	1



TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	23	99	618	2
YUV to HEVC	1080p	20	0	0	1	23	99	720	2
YUV to AV1	1080p	20	0	0	1	23	99	623	2
YUV to AVC	1080p	20	0	0	2	23	99	616	2
YUV to HEVC	1080p	20	0	0	2	23	100	440	1
YUV to AV1	1080p	20	0	0	2	23	100	380	1
YUV to AVC	1080p	20	0	0	3	23	99	618	2
YUV to HEVC	1080p	20	0	0	3	23	99	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	2
YUV to AVC	1080p	20	0	1	2	23	100	440	2
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	99	220	1
YUV to AVC	1080p	20	4	0	1	23	99	617	2
YUV to HEVC	1080p	20	4	0	1	23	99	717	2
YUV to AV1	1080p	20	4	0	1	23	99	620	2
YUV to AVC	1080p	20	4	0	2	23	99	616	2
YUV to HEVC	1080p	20	4	0	2	23	99	440	1
YUV to AV1	1080p	20	4	0	2	23	99	364	1
YUV to AVC	1080p	20	4	0	3	23	99	616	2
YUV to HEVC	1080p	20	4	0	3	23	99	300	1
YUV to AV1	1080p	20	4	0	3	23	99	240	1
YUV to AVC	1080p	20	4	1	1	23	99	440	1
YUV to HEVC	1080p	20	4	1	1	23	100	574	1
YUV to AVC	1080p	20	4	1	2	23	99	440	1
YUV to HEVC	1080p	20	4	1	2	23	99	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	603	1
YUV to HEVC	1080p	20	16	0	1	23	99	714	2
YUV to AV1	1080p	20	16	0	1	23	99	620	2
YUV to AVC	1080p	20	16	0	2	23	99	605	2
YUV to HEVC	1080p	20	16	0	2	23	99	440	1
YUV to AV1	1080p	20	16	0	2	23	100	364	1
YUV to AVC	1080p	20	16	0	3	23	99	603	2
YUV to HEVC	1080p	20	16	0	3	23	100	292	1
YUV to AV1	1080p	20	16	0	3	23	99	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	560	1
YUV to AVC	1080p	20	16	1	2	23	99	440	1
YUV to HEVC	1080p	20	16	1	2	23	100	317	1
YUV to AVC	1080p	20	16	1	3	23	99	440	1
YUV to HEVC	1080p	20	16	1	3	23	99	219	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	620	2
YUV to AVC	1080p	20	40	0	2	23	99	600	2
YUV to HEVC	1080p	20	40	0	2	23	99	440	2
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	284	1
YUV to AV1	1080p	20	40	0	3	23	99	225	1
YUV to AVC	1080p	20	40	1	1	23	99	438	1
YUV to HEVC	1080p	20	40	1	1	23	99	560	1
YUV to AVC	1080p	20	40	1	2	23	99	440	1
YUV to HEVC	1080p	20	40	1	2	23	99	300	1
YUV to AVC	1080p	20	40	1	3	23	99	440	1
YUV to HEVC	1080p	20	40	1	3	23	100	200	1
YUV to AVC	1080p	20	0	0	1	27	99	618	2
YUV to HEVC	1080p	20	0	0	1	27	99	719	2
YUV to AV1	1080p	20	0	0	1	27	99	624	2
YUV to AVC	1080p	20	0	0	2	27	99	618	2
YUV to HEVC	1080p	20	0	0	2	27	100	440	1
YUV to AV1	1080p	20	0	0	2	27	99	380	1
YUV to AVC	1080p	20	0	0	3	27	99	618	2
YUV to HEVC	1080p	20	0	0	3	27	100	300	1
YUV to AV1	1080p	20	0	0	3	27	99	240	1
YUV to AVC	1080p	20	0	1	1	27	100	440	1
YUV to HEVC	1080p	20	0	1	1	27	99	568	1
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	100	220	1
YUV to AVC	1080p	20	4	0	1	27	99	610	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	613	2
YUV to HEVC	1080p	20	4	0	2	27	99	440	1
YUV to AV1	1080p	20	4	0	2	27	99	366	1
YUV to AVC	1080p	20	4	0	3	27	99	614	2
YUV to HEVC	1080p	20	4	0	3	27	99	300	1
YUV to AV1	1080p	20	4	0	3	27	100	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	99	320	1
YUV to AVC	1080p	20	4	1	3	27	99	440	1
YUV to HEVC	1080p	20	4	1	3	27	99	220	1
YUV to AVC	1080p	20	16	0	1	27	99	604	2
YUV to HEVC	1080p	20	16	0	1	27	99	709	2
YUV to AV1	1080p	20	16	0	1	27	99	620	2
YUV to AVC	1080p	20	16	0	2	27	99	607	2
YUV to HEVC	1080p	20	16	0	2	27	99	440	1
YUV to AV1	1080p	20	16	0	2	27	100	362	1
YUV to AVC	1080p	20	16	0	3	27	99	601	2
YUV to HEVC	1080p	20	16	0	3	27	100	300	1
YUV to AV1	1080p	20	16	0	3	27	99	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	99	440	1
YUV to HEVC	1080p	20	16	1	2	27	99	308	1
YUV to AVC	1080p	20	16	1	3	27	99	440	1
YUV to HEVC	1080p	20	16	1	3	27	99	216	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	620	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	100	360	1
YUV to AVC	1080p	20	40	0	3	27	99	600	2
YUV to HEVC	1080p	20	40	0	3	27	99	286	1
YUV to AV1	1080p	20	40	0	3	27	99	226	1
YUV to AVC	1080p	20	40	1	1	27	99	440	1
YUV to HEVC	1080p	20	40	1	1	27	99	560	2
YUV to AVC	1080p	20	40	1	2	27	99	438	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	440	1
YUV to HEVC	1080p	20	40	1	3	27	99	200	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	9	14	21	4
AVC to HEVC	1	270	9	15	20	5
AVC to AV1	1	231	8	12	20	4
HEVC to AVC	1	254	12	14	21	4
HEVC to HEVC	1	269	13	15	21	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	962	2	72	93	24
AVC to HEVC	16	1069	2	78	92	27
AVC to AV1	16	993	2	69	92	24
HEVC to AVC	16	992	3	73	95	25
HEVC to HEVC	16	1088	3	77	93	27
HEVC to AV1	16	1008	3	67	93	24
VP9 to AVC	16	976	2	87	93	23
VP9 to HEVC	16	1054	2	91	88	25
VP9 to AV1	16	1008	3	85	93	24
AVC to AVC	32	864	1	74	95	24
AVC to HEVC	32	991	1	80	92	26
AVC to AV1	32	959	1	75	93	25
HEVC to AVC	32	913	1	74	95	24
HEVC to HEVC	32	1024	2	79	92	27
HEVC to AV1	32	960	1	72	94	25
VP9 to AVC	32	925	1	86	93	23
VP9 to HEVC	32	1021	1	94	92	26
VP9 to AV1	32	988	1	87	93	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	59	0	191	13
HEVC to YUV	4k	1	0	8	1	52	0	190	15
VP9 to YUV	4k	1	0	8	1	24	0	95	3
AVC to YUV	4k	5	0	8	0	84	0	251	5
HEVC to YUV	4k	5	0	8	0	82	0	258	5
VP9 to YUV	4k	5	0	8	0	80	0	266	2
YUV to AVC	4k	1	0	8	1	0	85	174	32
YUV to HEVC	4k	1	0	8	1	0	81	180	31
YUV to AV1	4k	1	0	8	1	0	91	175	33
YUV to AVC	4k	3	0	8	0	0	73	151	9
YUV to HEVC	4k	3	0	8	0	0	72	162	10
YUV to AV1	4k	3	0	8	0	0	73	138	8
YUV to AVC	4k	5	0	8	0	0	96	188	10
YUV to HEVC	4k	5	0	8	0	0	91	200	11
YUV to AV1	4k	5	0	8	0	0	97	186	7
AVC to AVC	4k	1	1	8	1	75	97	128	1
AVC to HEVC	4k	1	1	8	1	82	96	163	6
AVC to AV1	4k	1	1	8	1	77	96	159	9
HEVC to AVC	4k	1	1	8	1	52	97	128	6
HEVC to HEVC	4k	1	1	8	1	58	97	161	6
HEVC to AV1	4k	1	1	8	1	51	97	156	6
VP9 to AVC	4k	1	1	8	1	24	46	96	1
VP9 to HEVC	4k	1	1	8	1	24	43	96	1
VP9 to AV1	4k	1	1	8	1	24	51	96	1
AVC to AVC	4k	3	1	8	0	45	72	120	4
AVC to HEVC	4k	3	1	8	0	48	72	153	6
AVC to AV1	4k	3	1	8	0	41	72	136	1
HEVC to AVC	4k	3	1	8	0	42	72	123	5
HEVC to HEVC	4k	3	1	8	0	39	72	148	5
HEVC to AV1	4k	3	1	8	0	34	73	135	1
VP9 to AVC	4k	3	1	8	0	43	73	126	1
VP9 to HEVC	4k	3	1	8	0	42	72	153	1
VP9 to AV1	4k	3	1	8	0	35	72	135	1
AVC to AVC	4k	5	1	8	0	65	97	120	0
AVC to HEVC	4k	5	1	8	0	71	97	157	1
AVC to AV1	4k	5	1	8	0	64	97	155	0
HEVC to AVC	4k	5	1	8	0	55	98	122	0
HEVC to HEVC	4k	5	1	8	0	60	97	160	0
HEVC to AV1	4k	5	1	8	0	53	96	161	0

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AVC	4k	5	1	8	0	60	97	130	0
VP9 to HEVC	4k	5	1	8	0	65	97	170	0
VP9 to AV1	4k	5	1	8	0	56	97	165	1
AVC to YUV	4k	1	0	10	1	42	0	116	9
HEVC to YUV	4k	1	0	10	1	43	0	118	7
VP9 to YUV	4k	1	0	10	1	28	0	81	8
AVC to YUV	4k	16	0	10	0	58	0	142	2
HEVC to YUV	4k	16	0	10	0	68	0	138	2
VP9 to YUV	4k	16	0	10	0	60	0	136	2
YUV to AVC	4k	1	0	10	1	0	45	95	33
YUV to HEVC	4k	1	0	10	1	0	43	95	36
YUV to AV1	4k	1	0	10	1	0	49	94	36
YUV to AVC	4k	3	0	10	0	0	48	99	20
YUV to HEVC	4k	3	0	10	0	0	45	102	20
YUV to AV1	4k	3	0	10	0	0	51	100	20
AVC to YUV	1080p	1	0	8	1	50	0	598	14
HEVC to YUV	1080p	1	0	8	1	55	0	594	19
VP9 to YUV	1080p	1	0	8	1	23	0	211	7
AVC to YUV	1080p	18	0	8	0	92	0	863	0
HEVC to YUV	1080p	18	0	8	0	87	0	898	2
VP9 to YUV	1080p	18	0	8	0	90	0	821	0
YUV to AVC	1080p	1	0	8	1	0	58	483	21
YUV to HEVC	1080p	1	0	8	1	0	55	485	21
YUV to AV1	1080p	1	0	8	1	0	60	456	19
YUV to AVC	1080p	20	0	8	0	0	86	704	2
YUV to HEVC	1080p	20	0	8	0	0	82	720	2
YUV to AV1	1080p	20	0	8	0	0	94	700	2
AVC to AVC	1080p	1	1	8	1	74	83	597	10
AVC to HEVC	1080p	1	1	8	1	77	83	660	9
AVC to AV1	1080p	1	1	8	1	60	85	601	8
HEVC to AVC	1080p	1	1	8	1	59	83	576	13
HEVC to HEVC	1080p	1	1	8	1	64	84	643	14
HEVC to AV1	1080p	1	1	8	1	57	85	583	12
VP9 to AVC	1080p	1	1	8	1	23	25	211	4
VP9 to HEVC	1080p	1	1	8	1	23	24	211	6
VP9 to AV1	1080p	1	1	8	1	23	28	211	4
AVC to AVC	1080p	20	1	8	0	73	99	571	0
AVC to HEVC	1080p	20	1	8	0	79	99	656	0
AVC to AV1	1080p	20	1	8	0	73	99	640	0
HEVC to AVC	1080p	20	1	8	0	69	99	600	2

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to HEVC	1080p	20	1	8	0	73	99	681	0
HEVC to AV1	1080p	20	1	8	0	66	99	660	0
VP9 to AVC	1080p	20	1	8	0	79	99	580	1
VP9 to HEVC	1080p	20	1	8	0	87	99	660	0
VP9 to AV1	1080p	20	1	8	0	81	99	644	0
AVC to YUV	1080p	1	0	10	1	47	0	415	9
HEVC to YUV	1080p	1	0	10	1	46	0	415	10
VP9 to YUV	1080p	1	0	10	1	32	0	314	6
AVC to YUV	1080p	18	0	10	0	76	0	529	0
HEVC to YUV	1080p	18	0	10	0	73	0	522	0
VP9 to YUV	1080p	18	0	10	0	76	0	522	0
YUV to AVC	1080p	1	0	10	1	0	37	309	24
YUV to HEVC	1080p	1	0	10	1	0	34	308	26
YUV to AV1	1080p	1	0	10	1	0	39	295	26
YUV to AVC	1080p	20	0	10	0	0	47	393	3
YUV to HEVC	1080p	20	0	10	0	0	44	398	3
YUV to AV1	1080p	20	0	10	0	0	50	380	2
AVC to YUV	720p	1	0	8	1	52	0	940	9
HEVC to YUV	720p	1	0	8	1	47	0	926	14
VP9 to YUV	720p	1	0	8	1	34	0	691	6
AVC to YUV	720p	40	0	8	0	100	0	1520	0
HEVC to YUV	720p	40	0	8	0	100	0	1606	1
VP9 to YUV	720p	40	0	8	0	100	0	1560	0
YUV to AVC	720p	1	0	8	1	0	38	717	13
YUV to HEVC	720p	1	0	8	1	0	38	722	12
YUV to AV1	720p	1	0	8	1	0	40	656	10
YUV to AVC	720p	40	0	8	0	0	78	1401	1
YUV to HEVC	720p	40	0	8	0	0	76	1432	1
YUV to AV1	720p	40	0	8	0	0	91	1398	0
AVC to AVC	720p	1	1	8	1	49	56	1050	8
AVC to HEVC	720p	1	1	8	1	49	58	1089	9
AVC to AV1	720p	1	1	8	1	42	61	964	7
HEVC to AVC	720p	1	1	8	1	49	55	1017	10
HEVC to HEVC	720p	1	1	8	1	51	57	1049	10
HEVC to AV1	720p	1	1	8	1	43	61	945	10
VP9 to AVC	720p	1	1	8	1	24	36	686	6
VP9 to HEVC	720p	1	1	8	1	23	36	688	4
VP9 to AV1	720p	1	1	8	1	23	43	689	4
AVC to AVC	720p	40	1	8	0	92	100	1287	0
AVC to HEVC	720p	40	1	8	0	94	100	1413	0

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	720p	40	1	8	0	76	100	1286	0
HEVC to AVC	720p	40	1	8	0	87	100	1292	0
HEVC to HEVC	720p	40	1	8	0	87	100	1428	0
HEVC to AV1	720p	40	1	8	0	68	100	1288	0
VP9 to AVC	720p	40	1	8	0	94	100	1400	0
VP9 to HEVC	720p	40	1	8	0	97	100	1534	0
VP9 to AV1	720p	40	1	8	0	77	100	1391	0

## 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	57	0	182	11
HEVC to YUV	4k	1	0	8	1	50	0	182	9
VP9 to YUV	4k	1	0	8	1	24	0	95	5
AVC to YUV	4k	5	0	8	0	83	0	251	2
HEVC to YUV	4k	5	0	8	0	78	0	253	2
VP9 to YUV	4k	5	0	8	0	80	0	261	2
YUV to AVC	4k	1	0	8	1	0	77	159	19
YUV to HEVC	4k	1	0	8	1	0	73	162	19
YUV to AV1	4k	1	0	8	1	0	80	154	20
YUV to AVC	4k	3	0	8	0	0	72	151	8
YUV to HEVC	4k	3	0	8	0	0	73	161	7
YUV to AV1	4k	3	0	8	0	0	73	140	6
YUV to AVC	4k	5	0	8	0	0	97	184	6
YUV to HEVC	4k	5	0	8	0	0	91	201	8
YUV to AV1	4k	5	0	8	0	0	98	187	6
AVC to AVC	4k	1	1	8	0	75	97	128	3
AVC to HEVC	4k	1	1	8	0	81	96	163	2
AVC to AV1	4k	1	1	8	0	77	96	159	2
HEVC to AVC	4k	1	1	8	0	54	97	128	1
HEVC to HEVC	4k	1	1	8	0	58	97	159	2
HEVC to AV1	4k	1	1	8	0	53	97	151	2
VP9 to AVC	4k	1	1	8	0	24	46	95	2
VP9 to HEVC	4k	1	1	8	0	24	43	96	2
VP9 to AV1	4k	1	1	8	0	24	50	96	2
AVC to AVC	4k	3	1	8	0	45	72	122	2
AVC to HEVC	4k	3	1	8	0	47	72	154	2
AVC to AV1	4k	3	1	8	0	40	72	137	2
HEVC to AVC	4k	3	1	8	0	39	72	119	2
HEVC to HEVC	4k	3	1	8	0	41	72	145	2
HEVC to AV1	4k	3	1	8	0	36	73	133	2
VP9 to AVC	4k	3	1	8	0	43	73	127	2
VP9 to HEVC	4k	3	1	8	0	42	72	153	2
VP9 to AV1	4k	3	1	8	0	37	73	136	2
AVC to AVC	4k	5	1	8	0	65	98	122	2
AVC to HEVC	4k	5	1	8	0	71	97	159	2
AVC to AV1	4k	5	1	8	0	64	96	159	2
HEVC to AVC	4k	5	1	8	0	54	99	123	1
HEVC to HEVC	4k	5	1	8	0	59	97	159	2



TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to AV1	4k	5	1	8	0	54	97	157	2
VP9 to AVC	4k	5	1	8	0	58	98	130	1
VP9 to HEVC	4k	5	1	8	0	65	98	170	2
VP9 to AV1	4k	5	1	8	0	57	96	166	2
AVC to YUV	4k	1	0	10	1	53	0	114	9
HEVC to YUV	4k	1	0	10	1	43	0	109	9
VP9 to YUV	4k	1	0	10	1	31	0	81	7
AVC to YUV	4k	5	0	10	0	56	0	132	2
HEVC to YUV	4k	5	0	10	0	63	0	132	2
VP9 to YUV	4k	5	0	10	0	67	0	134	2
YUV to AVC	4k	1	0	10	1	0	40	84	22
YUV to HEVC	4k	1	0	10	1	0	38	84	21
YUV to AV1	4k	1	0	10	1	0	43	82	20
YUV to AVC	4k	3	0	10	0	0	46	96	13
YUV to HEVC	4k	3	0	10	0	0	42	96	14
YUV to AV1	4k	3	0	10	0	0	47	92	14
AVC to YUV	1080p	1	0	8	1	51	0	584	13
HEVC to YUV	1080p	1	0	8	1	58	0	606	10
VP9 to YUV	1080p	1	0	8	1	23	0	211	3
AVC to YUV	1080p	20	0	8	0	87	0	827	0
HEVC to YUV	1080p	20	0	8	0	79	0	861	0
VP9 to YUV	1080p	20	0	8	0	87	0	799	0
YUV to AVC	1080p	1	0	8	1	0	58	478	15
YUV to HEVC	1080p	1	0	8	1	0	53	471	17
YUV to AV1	1080p	1	0	8	1	0	54	413	14
YUV to AVC	1080p	20	0	8	0	0	89	745	2
YUV to HEVC	1080p	20	0	8	0	0	86	759	2
YUV to AV1	1080p	20	0	8	0	0	96	730	2
AVC to AVC	1080p	1	1	8	0	75	84	601	6
AVC to HEVC	1080p	1	1	8	0	76	84	667	6
AVC to AV1	1080p	1	1	8	0	60	85	597	7
HEVC to AVC	1080p	1	1	8	0	61	87	583	4
HEVC to HEVC	1080p	1	1	8	0	67	86	669	4
HEVC to AV1	1080p	1	1	8	0	58	88	594	5
VP9 to AVC	1080p	1	1	8	0	23	25	211	2
VP9 to HEVC	1080p	1	1	8	0	23	24	211	2
VP9 to AV1	1080p	1	1	8	0	23	28	211	2
AVC to AVC	1080p	20	1	8	0	73	99	585	0
AVC to HEVC	1080p	20	1	8	0	79	99	666	0

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	1080p	20	1	8	0	73	99	652	0
HEVC to AVC	1080p	20	1	8	0	68	99	610	0
HEVC to HEVC	1080p	20	1	8	0	71	99	697	0
HEVC to AV1	1080p	20	1	8	0	67	99	668	0
VP9 to AVC	1080p	20	1	8	0	77	99	591	0
VP9 to HEVC	1080p	20	1	8	0	86	99	678	0
VP9 to AV1	1080p	20	1	8	0	83	99	662	0
AVC to YUV	1080p	1	0	10	1	48	0	420	10
HEVC to YUV	1080p	1	0	10	1	49	0	402	10
VP9 to YUV	1080p	1	0	10	1	35	0	314	8
AVC to YUV	1080p	20	0	10	0	74	0	520	0
HEVC to YUV	1080p	20	0	10	0	70	0	510	0
VP9 to YUV	1080p	20	0	10	0	100	0	449	0
YUV to AVC	1080p	1	0	10	1	0	35	289	16
YUV to HEVC	1080p	1	0	10	1	0	33	293	18
YUV to AV1	1080p	1	0	10	1	0	35	263	17
YUV to AVC	1080p	20	0	10	0	0	48	408	2
YUV to HEVC	1080p	20	0	10	0	0	45	409	2
YUV to AV1	1080p	20	0	10	0	0	51	396	3
AVC to YUV	720p	1	0	8	1	51	0	915	10
HEVC to YUV	720p	1	0	8	1	46	0	888	7
VP9 to YUV	720p	1	0	8	1	34	0	690	4
AVC to YUV	720p	40	0	8	0	100	0	1579	0
HEVC to YUV	720p	40	0	8	0	92	0	1557	0
VP9 to YUV	720p	40	0	8	0	94	0	1530	0
YUV to AVC	720p	1	0	8	1	0	41	763	12
YUV to HEVC	720p	1	0	8	1	0	40	764	13
YUV to AV1	720p	1	0	8	1	0	38	617	11
YUV to AVC	720p	40	0	8	0	0	84	1559	1
YUV to HEVC	720p	40	0	8	0	0	82	1582	1
YUV to AV1	720p	40	0	8	0	0	96	1503	1
AVC to AVC	720p	1	1	8	0	51	59	1095	7
AVC to HEVC	720p	1	1	8	0	51	62	1143	7
AVC to AV1	720p	1	1	8	0	43	60	946	7
HEVC to AVC	720p	1	1	8	0	54	57	1035	4
HEVC to HEVC	720p	1	1	8	0	55	59	1056	4
HEVC to AV1	720p	1	1	8	0	46	61	936	5
VP9 to AVC	720p	1	1	8	0	24	36	687	3
VP9 to HEVC	720p	1	1	8	0	24	36	687	3

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	720p	1	1	8	0	23	43	689	3
AVC to AVC	720p	40	1	8	0	100	100	1347	0
AVC to HEVC	720p	40	1	8	0	100	100	1477	0
AVC to AV1	720p	40	1	8	0	96	100	1338	0
HEVC to AVC	720p	40	1	8	0	100	100	1350	0
HEVC to HEVC	720p	40	1	8	0	100	100	1478	0
HEVC to AV1	720p	40	1	8	0	94	100	1340	0
VP9 to AVC	720p	40	1	8	0	100	100	1438	0
VP9 to HEVC	720p	40	1	8	0	100	100	1572	0
VP9 to AV1	720p	40	1	8	0	99	100	1433	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	24.78	28.05	24.39	0.13
YUV to HEVC	4k	1	26.76	29.67	25.44	0.2
YUV to AV1	4k	1	34.93	40.61	26.32	1.53
YUV to AVC	4k	3	25.95	31.37	24.69	1.65
YUV to HEVC	4k	3	28.19	37.69	25.54	3.21
YUV to AV1	4k	3	35.15	41.12	26.41	1.51
YUV to AVC	4k	5	30.2	51.93	24.87	8.27
YUV to HEVC	4k	5	34.01	56.96	25.69	68.27
YUV to AV1	4k	5	44.07	75.76	26.67	131.37
YUV to AVC	1080p	1	7.11	8.41	6.89	0.03
YUV to HEVC	1080p	1	7.66	13.09	7.22	0.08
YUV to AV1	1080p	1	10.28	11.46	7.76	0.11
YUV to AVC	1080p	20	12.83	24.76	8.14	5.91
YUV to HEVC	1080p	20	13.88	27.23	8.27	10.08
YUV to AV1	1080p	20	40.21	54.32	27.32	12.24
YUV to AVC	720p	1	4.01	12.45	3.63	0.19
YUV to HEVC	720p	1	4.03	4.67	3.73	0.01
YUV to AV1	720p	1	5.49	5.93	4.34	0.04
YUV to AVC	720p	40	7.45	19.05	4.25	1.33
YUV to HEVC	720p	40	8.14	20.61	4.55	2.02
YUV to AV1	720p	40	39	51.83	26	4.18

## 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	4
AVC to RGBA	4k	8	91	1	342	1
HEVC to RGBA	4k	1	22	0	101	4
HEVC to RGBA	4k	8	94	2	408	2
VP9 to RGBA	4k	1	23	0	96	2
VP9 to RGBA	4k	8	93	2	387	0
AVC to RGBA	1080p	20	93	9	1191	1
HEVC to RGBA	1080p	20	92	8	1169	1
VP9 to RGBA	1080p	20	94	6	896	0
AVC to RGBA	720p	50	95	18	2200	0
HEVC to RGBA	720p	50	87	19	2252	0
VP9 to RGBA	720p	32	94	18	2368	0

## 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	94	4	250	1
AVC to HEVC	5	33	95	3	280	2
AVC to AV1	5	26	93	2	250	2
HEVC to AVC	5	36	94	4	255	2
HEVC to HEVC	5	34	95	3	285	2
HEVC to AV1	5	25	93	2	255	2
VP9 to AVC	5	40	94	4	250	1
VP9 to HEVC	5	39	95	3	284	1
VP9 to AV1	5	31	93	2	255	2

## 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	35	73	43
RGBA to HEVC	4k	1	1	33	74	43
RGBA to AV1	4k	1	1	38	73	43
RGBA to AVC	4k	3	0	38	75	17
RGBA to HEVC	4k	3	0	35	76	18
RGBA to AV1	4k	3	0	41	76	18
RGBA to AVC	4k	5	0	39	75	11
RGBA to HEVC	4k	5	0	36	75	11
RGBA to AV1	4k	5	0	40	75	12
RGBA to AVC	1080p	1	1	30	250	38
RGBA to HEVC	1080p	1	1	29	252	40
RGBA to AV1	1080p	1	1	32	247	38
RGBA to AVC	1080p	10	0	37	280	7
RGBA to HEVC	1080p	10	0	33	285	6
RGBA to AV1	1080p	10	0	38	285	5
RGBA to AVC	1080p	20	0	36	280	3
RGBA to HEVC	1080p	20	0	33	280	3
RGBA to AV1	1080p	20	0	37	280	3
RGBA to AVC	720p	1	1	33	522	49
RGBA to HEVC	720p	1	1	33	518	50
RGBA to AV1	720p	1	1	37	506	44
RGBA to AVC	720p	10	0	42	610	11
RGBA to HEVC	720p	10	0	41	614	11
RGBA to AV1	720p	10	0	47	609	9
RGBA to AVC	720p	20	0	45	618	8
RGBA to HEVC	720p	20	0	42	620	7
RGBA to AV1	720p	20	0	48	613	6

## 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	9
YUV to HEVC	4k	2	0	0	1	49	108	12
YUV to AV1	4k	2	0	0	1	48	94	9
YUV to AVC	4k	2	0	0	2	48	102	9
YUV to HEVC	4k	2	0	0	2	49	54	6
YUV to AV1	4k	2	0	0	2	49	44	4
YUV to AVC	4k	2	0	0	3	48	102	9
YUV to HEVC	4k	2	0	0	3	49	32	3
YUV to AV1	4k	2	0	0	3	49	24	2
YUV to AVC	4k	2	0	1	1	50	60	6
YUV to HEVC	4k	2	0	1	1	49	76	7
YUV to AVC	4k	2	0	1	2	49	60	7
YUV to HEVC	4k	2	0	1	2	50	34	4
YUV to AVC	4k	2	0	1	3	48	60	4
YUV to HEVC	4k	2	0	1	3	49	22	3
YUV to AVC	4k	2	4	0	1	79	100	9
YUV to HEVC	4k	2	4	0	1	74	108	10
YUV to AV1	4k	2	4	0	1	72	94	9
YUV to AVC	4k	2	4	0	2	78	100	9
YUV to HEVC	4k	2	4	0	2	62	54	5
YUV to AV1	4k	2	4	0	2	62	44	4
YUV to AVC	4k	2	4	0	3	78	100	9
YUV to HEVC	4k	2	4	0	3	56	32	3
YUV to AV1	4k	2	4	0	3	55	26	2
YUV to AVC	4k	2	4	1	1	66	60	6
YUV to HEVC	4k	2	4	1	1	67	76	7
YUV to AVC	4k	2	4	1	2	66	60	6
YUV to HEVC	4k	2	4	1	2	59	34	3
YUV to AVC	4k	2	4	1	3	67	60	5
YUV to HEVC	4k	2	4	1	3	54	22	3
YUV to AVC	4k	2	16	0	1	79	100	10
YUV to HEVC	4k	2	16	0	1	74	108	12
YUV to AV1	4k	2	16	0	1	73	92	8
YUV to AVC	4k	2	16	0	2	80	100	10
YUV to HEVC	4k	2	16	0	2	61	54	6
YUV to AV1	4k	2	16	0	2	60	44	4
YUV to AVC	4k	2	16	0	3	80	100	9
YUV to HEVC	4k	2	16	0	3	56	32	3
YUV to AV1	4k	2	16	0	3	55	26	3

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	16	1	1	68	60	6
YUV to HEVC	4k	2	16	1	1	67	76	7
YUV to AVC	4k	2	16	1	2	69	60	6
YUV to HEVC	4k	2	16	1	2	58	34	4
YUV to AVC	4k	2	16	1	3	69	60	6
YUV to HEVC	4k	2	16	1	3	54	22	3
YUV to AVC	4k	2	40	0	1	79	99	10
YUV to HEVC	4k	2	40	0	1	74	107	10
YUV to AV1	4k	2	40	0	1	73	92	7
YUV to AVC	4k	2	40	0	2	79	99	9
YUV to HEVC	4k	2	40	0	2	62	54	6
YUV to AV1	4k	2	40	0	2	60	44	4
YUV to AVC	4k	2	40	0	3	79	98	9
YUV to HEVC	4k	2	40	0	3	57	32	3
YUV to AV1	4k	2	40	0	3	55	26	2
YUV to AVC	4k	2	40	1	1	66	60	6
YUV to HEVC	4k	2	40	1	1	67	76	9
YUV to AVC	4k	2	40	1	2	66	60	6
YUV to HEVC	4k	2	40	1	2	59	34	3
YUV to AVC	4k	2	40	1	3	68	60	6
YUV to HEVC	4k	2	40	1	3	54	22	3
YUV to AVC	1080p	10	0	0	1	85	700	6
YUV to HEVC	1080p	10	0	0	1	81	710	7
YUV to AV1	1080p	10	0	0	1	91	690	5
YUV to AVC	1080p	10	0	0	2	84	700	6
YUV to HEVC	1080p	10	0	0	2	99	441	3
YUV to AV1	1080p	10	0	0	2	100	356	2
YUV to AVC	1080p	10	0	0	3	84	699	7
YUV to HEVC	1080p	10	0	0	3	99	261	2
YUV to AV1	1080p	10	0	0	3	100	190	1
YUV to AVC	1080p	10	0	1	1	99	481	3
YUV to HEVC	1080p	10	0	1	1	100	613	3
YUV to AVC	1080p	10	0	1	2	100	481	3
YUV to HEVC	1080p	10	0	1	2	99	278	1
YUV to AVC	1080p	10	0	1	3	99	479	2
YUV to HEVC	1080p	10	0	1	3	99	179	1
YUV to AVC	1080p	10	4	0	1	99	450	2
YUV to HEVC	1080p	10	4	0	1	99	525	3
YUV to AV1	1080p	10	4	0	1	99	460	2
YUV to AVC	1080p	10	4	0	2	99	450	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	4	0	2	99	330	1
YUV to AV1	1080p	10	4	0	2	99	270	1
YUV to AVC	1080p	10	4	0	3	100	449	2
YUV to HEVC	1080p	10	4	0	3	99	217	1
YUV to AV1	1080p	10	4	0	3	99	171	1
YUV to AVC	1080p	10	4	1	1	99	324	2
YUV to HEVC	1080p	10	4	1	1	99	420	2
YUV to AVC	1080p	10	4	1	2	99	323	2
YUV to HEVC	1080p	10	4	1	2	99	230	1
YUV to AVC	1080p	10	4	1	3	100	325	2
YUV to HEVC	1080p	10	4	1	3	99	156	1
YUV to AVC	1080p	10	16	0	1	99	448	2
YUV to HEVC	1080p	10	16	0	1	99	520	3
YUV to AV1	1080p	10	16	0	1	99	459	2
YUV to AVC	1080p	10	16	0	2	99	447	2
YUV to HEVC	1080p	10	16	0	2	100	325	2
YUV to AV1	1080p	10	16	0	2	99	270	1
YUV to AVC	1080p	10	16	0	3	99	448	2
YUV to HEVC	1080p	10	16	0	3	100	214	1
YUV to AV1	1080p	10	16	0	3	99	170	1
YUV to AVC	1080p	10	16	1	1	99	321	2
YUV to HEVC	1080p	10	16	1	1	99	417	2
YUV to AVC	1080p	10	16	1	2	99	321	2
YUV to HEVC	1080p	10	16	1	2	100	230	1
YUV to AVC	1080p	10	16	1	3	99	322	2
YUV to HEVC	1080p	10	16	1	3	99	155	1
YUV to AVC	1080p	10	40	0	1	99	441	3
YUV to HEVC	1080p	10	40	0	1	99	519	3
YUV to AV1	1080p	10	40	0	1	99	452	2
YUV to AVC	1080p	10	40	0	2	99	440	2
YUV to HEVC	1080p	10	40	0	2	99	320	1
YUV to AV1	1080p	10	40	0	2	99	270	1
YUV to AVC	1080p	10	40	0	3	99	441	2
YUV to HEVC	1080p	10	40	0	3	100	211	1
YUV to AV1	1080p	10	40	0	3	100	170	1
YUV to AVC	1080p	10	40	1	1	99	320	2
YUV to HEVC	1080p	10	40	1	1	99	411	2
YUV to AVC	1080p	10	40	1	2	99	320	2
YUV to HEVC	1080p	10	40	1	2	99	229	1
YUV to AVC	1080p	10	40	1	3	99	320	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	1	3	99	153	1
YUV to AVC	720p	20	0	0	1	78	1370	2
YUV to HEVC	720p	20	0	0	1	75	1394	2
YUV to AV1	720p	20	0	0	1	86	1338	2
YUV to AVC	720p	20	0	0	2	78	1371	2
YUV to HEVC	720p	20	0	0	2	99	960	1
YUV to AV1	720p	20	0	0	2	100	780	1
YUV to AVC	720p	20	0	0	3	78	1371	2
YUV to HEVC	720p	20	0	0	3	99	580	0
YUV to AV1	720p	20	0	0	3	99	420	0
YUV to AVC	720p	20	0	1	1	99	1080	1
YUV to HEVC	720p	20	0	1	1	99	1320	1
YUV to AVC	720p	20	0	1	2	99	1080	1
YUV to HEVC	720p	20	0	1	2	99	620	1
YUV to AVC	720p	20	0	1	3	99	1080	1
YUV to HEVC	720p	20	0	1	3	99	400	0
YUV to AVC	720p	20	4	0	1	99	920	1
YUV to HEVC	720p	20	4	0	1	99	1049	1
YUV to AV1	720p	20	4	0	1	100	823	1
YUV to AVC	720p	20	4	0	2	99	920	1
YUV to HEVC	720p	20	4	0	2	99	680	0
YUV to AV1	720p	20	4	0	2	99	560	0
YUV to AVC	720p	20	4	0	3	99	920	1
YUV to HEVC	720p	20	4	0	3	99	460	0
YUV to AV1	720p	20	4	0	3	99	360	0
YUV to AVC	720p	20	4	1	1	99	680	1
YUV to HEVC	720p	20	4	1	1	99	860	1
YUV to AVC	720p	20	4	1	2	99	680	1
YUV to HEVC	720p	20	4	1	2	99	480	0
YUV to AVC	720p	20	4	1	3	99	680	1
YUV to HEVC	720p	20	4	1	3	100	340	0
YUV to AVC	720p	20	16	0	1	99	919	1
YUV to HEVC	720p	20	16	0	1	99	1060	1
YUV to AV1	720p	20	16	0	1	100	862	1
YUV to AVC	720p	20	16	0	2	99	920	1
YUV to HEVC	720p	20	16	0	2	99	680	1
YUV to AV1	720p	20	16	0	2	99	560	0
YUV to AVC	720p	20	16	0	3	99	919	1
YUV to HEVC	720p	20	16	0	3	99	460	0
YUV to AV1	720p	20	16	0	3	99	360	0

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	20	16	1	1	99	680	1
YUV to HEVC	720p	20	16	1	1	99	859	1
YUV to AVC	720p	20	16	1	2	99	680	1
YUV to HEVC	720p	20	16	1	2	99	480	0
YUV to AVC	720p	20	16	1	3	99	680	1
YUV to HEVC	720p	20	16	1	3	99	340	0
YUV to AVC	720p	20	40	0	1	99	909	1
YUV to HEVC	720p	20	40	0	1	99	1025	1
YUV to AV1	720p	20	40	0	1	100	845	1
YUV to AVC	720p	20	40	0	2	99	910	1
YUV to HEVC	720p	20	40	0	2	99	680	1
YUV to AV1	720p	20	40	0	2	99	558	0
YUV to AVC	720p	20	40	0	3	99	908	1
YUV to HEVC	720p	20	40	0	3	99	460	0
YUV to AV1	720p	20	40	0	3	99	354	0
YUV to AVC	720p	20	40	1	1	99	669	1
YUV to HEVC	720p	20	40	1	1	99	840	1
YUV to AVC	720p	20	40	1	2	99	672	1
YUV to HEVC	720p	20	40	1	2	99	480	0
YUV to AVC	720p	20	40	1	3	99	672	1
YUV to HEVC	720p	20	40	1	3	100	340	0

## 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	2
YUV to HEVC	1080p	10	0	0	1	19	99	522	3
YUV to AV1	1080p	10	0	0	1	19	99	460	2
YUV to AVC	1080p	10	0	0	2	19	99	450	2
YUV to HEVC	1080p	10	0	0	2	19	99	330	1
YUV to AV1	1080p	10	0	0	2	19	99	280	1
YUV to AVC	1080p	10	0	0	3	19	99	451	2
YUV to HEVC	1080p	10	0	0	3	19	100	216	1
YUV to AV1	1080p	10	0	0	3	19	99	176	1
YUV to AVC	1080p	10	0	1	1	19	100	326	2
YUV to HEVC	1080p	10	0	1	1	19	100	420	2
YUV to AVC	1080p	10	0	1	2	19	99	325	2
YUV to HEVC	1080p	10	0	1	2	19	99	230	1
YUV to AVC	1080p	10	0	1	3	19	100	326	2
YUV to HEVC	1080p	10	0	1	3	19	98	158	1
YUV to AVC	1080p	10	4	0	1	19	99	450	2
YUV to HEVC	1080p	10	4	0	1	19	99	521	3
YUV to AV1	1080p	10	4	0	1	19	99	460	2
YUV to AVC	1080p	10	4	0	2	19	99	449	2
YUV to HEVC	1080p	10	4	0	2	19	99	330	2
YUV to AV1	1080p	10	4	0	2	19	99	270	1
YUV to AVC	1080p	10	4	0	3	19	99	450	2
YUV to HEVC	1080p	10	4	0	3	19	99	216	1
YUV to AV1	1080p	10	4	0	3	19	99	170	1
YUV to AVC	1080p	10	4	1	1	19	99	323	2
YUV to HEVC	1080p	10	4	1	1	19	99	420	2
YUV to AVC	1080p	10	4	1	2	19	100	323	2
YUV to HEVC	1080p	10	4	1	2	19	100	230	1
YUV to AVC	1080p	10	4	1	3	19	99	323	2
YUV to HEVC	1080p	10	4	1	3	19	99	159	1
YUV to AVC	1080p	10	16	0	1	19	99	447	3
YUV to HEVC	1080p	10	16	0	1	19	99	521	3
YUV to AV1	1080p	10	16	0	1	19	100	460	2
YUV to AVC	1080p	10	16	0	2	19	99	447	3
YUV to HEVC	1080p	10	16	0	2	19	99	327	1
YUV to AV1	1080p	10	16	0	2	19	99	270	1
YUV to AVC	1080p	10	16	0	3	19	100	448	2
YUV to HEVC	1080p	10	16	0	3	19	100	215	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	3	19	100	171	1
YUV to AVC	1080p	10	16	1	1	19	99	320	2
YUV to HEVC	1080p	10	16	1	1	19	99	419	2
YUV to AVC	1080p	10	16	1	2	19	99	320	2
YUV to HEVC	1080p	10	16	1	2	19	100	230	1
YUV to AVC	1080p	10	16	1	3	19	99	320	2
YUV to HEVC	1080p	10	16	1	3	19	99	156	1
YUV to AVC	1080p	10	40	0	1	19	99	441	3
YUV to HEVC	1080p	10	40	0	1	19	99	517	3
YUV to AV1	1080p	10	40	0	1	19	99	452	3
YUV to AVC	1080p	10	40	0	2	19	99	441	2
YUV to HEVC	1080p	10	40	0	2	19	99	320	2
YUV to AV1	1080p	10	40	0	2	19	99	270	1
YUV to AVC	1080p	10	40	0	3	19	99	441	2
YUV to HEVC	1080p	10	40	0	3	19	99	212	1
YUV to AV1	1080p	10	40	0	3	19	99	170	1
YUV to AVC	1080p	10	40	1	1	19	99	320	2
YUV to HEVC	1080p	10	40	1	1	19	99	410	2
YUV to AVC	1080p	10	40	1	2	19	99	320	2
YUV to HEVC	1080p	10	40	1	2	19	100	230	1
YUV to AVC	1080p	10	40	1	3	19	100	320	2
YUV to HEVC	1080p	10	40	1	3	19	99	152	1
YUV to AVC	1080p	10	0	0	1	23	99	451	3
YUV to HEVC	1080p	10	0	0	1	23	99	526	3
YUV to AV1	1080p	10	0	0	1	23	99	460	2
YUV to AVC	1080p	10	0	0	2	23	99	450	3
YUV to HEVC	1080p	10	0	0	2	23	99	330	2
YUV to AV1	1080p	10	0	0	2	23	99	279	1
YUV to AVC	1080p	10	0	0	3	23	99	450	2
YUV to HEVC	1080p	10	0	0	3	23	100	217	1
YUV to AV1	1080p	10	0	0	3	23	100	178	1
YUV to AVC	1080p	10	0	1	1	23	99	326	2
YUV to HEVC	1080p	10	0	1	1	23	99	420	2
YUV to AVC	1080p	10	0	1	2	23	99	326	2
YUV to HEVC	1080p	10	0	1	2	23	99	230	1
YUV to AVC	1080p	10	0	1	3	23	99	325	2
YUV to HEVC	1080p	10	0	1	3	23	100	157	1
YUV to AVC	1080p	10	4	0	1	23	99	450	2
YUV to HEVC	1080p	10	4	0	1	23	99	523	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	4	0	1	23	99	460	2
YUV to AVC	1080p	10	4	0	2	23	99	448	3
YUV to HEVC	1080p	10	4	0	2	23	99	330	1
YUV to AV1	1080p	10	4	0	2	23	99	270	1
YUV to AVC	1080p	10	4	0	3	23	99	449	2
YUV to HEVC	1080p	10	4	0	3	23	100	217	1
YUV to AV1	1080p	10	4	0	3	23	100	171	1
YUV to AVC	1080p	10	4	1	1	23	99	322	2
YUV to HEVC	1080p	10	4	1	1	23	99	420	2
YUV to AVC	1080p	10	4	1	2	23	99	325	2
YUV to HEVC	1080p	10	4	1	2	23	100	230	1
YUV to AVC	1080p	10	4	1	3	23	99	323	2
YUV to HEVC	1080p	10	4	1	3	23	100	159	1
YUV to AVC	1080p	10	16	0	1	23	99	448	3
YUV to HEVC	1080p	10	16	0	1	23	99	521	3
YUV to AV1	1080p	10	16	0	1	23	99	459	2
YUV to AVC	1080p	10	16	0	2	23	99	448	2
YUV to HEVC	1080p	10	16	0	2	23	99	325	2
YUV to AV1	1080p	10	16	0	2	23	100	270	2
YUV to AVC	1080p	10	16	0	3	23	99	449	3
YUV to HEVC	1080p	10	16	0	3	23	99	215	1
YUV to AV1	1080p	10	16	0	3	23	99	170	1
YUV to AVC	1080p	10	16	1	1	23	99	321	2
YUV to HEVC	1080p	10	16	1	1	23	99	419	2
YUV to AVC	1080p	10	16	1	2	23	99	321	2
YUV to HEVC	1080p	10	16	1	2	23	99	230	1
YUV to AVC	1080p	10	16	1	3	23	100	321	2
YUV to HEVC	1080p	10	16	1	3	23	100	155	1
YUV to AVC	1080p	10	40	0	1	23	99	441	3
YUV to HEVC	1080p	10	40	0	1	23	99	518	3
YUV to AV1	1080p	10	40	0	1	23	99	452	3
YUV to AVC	1080p	10	40	0	2	23	99	442	3
YUV to HEVC	1080p	10	40	0	2	23	99	320	2
YUV to AV1	1080p	10	40	0	2	23	99	270	1
YUV to AVC	1080p	10	40	0	3	23	99	441	3
YUV to HEVC	1080p	10	40	0	3	23	100	211	1
YUV to AV1	1080p	10	40	0	3	23	99	170	1
YUV to AVC	1080p	10	40	1	1	23	99	320	2
YUV to HEVC	1080p	10	40	1	1	23	100	410	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	40	1	2	23	99	320	2
YUV to HEVC	1080p	10	40	1	2	23	100	230	1
YUV to AVC	1080p	10	40	1	3	23	99	320	2
YUV to HEVC	1080p	10	40	1	3	23	99	151	1
YUV to AVC	1080p	10	0	0	1	27	99	451	3
YUV to HEVC	1080p	10	0	0	1	27	99	525	3
YUV to AV1	1080p	10	0	0	1	27	99	461	3
YUV to AVC	1080p	10	0	0	2	27	99	450	3
YUV to HEVC	1080p	10	0	0	2	27	99	330	2
YUV to AV1	1080p	10	0	0	2	27	99	279	1
YUV to AVC	1080p	10	0	0	3	27	99	451	3
YUV to HEVC	1080p	10	0	0	3	27	99	217	1
YUV to AV1	1080p	10	0	0	3	27	100	177	1
YUV to AVC	1080p	10	0	1	1	27	100	325	2
YUV to HEVC	1080p	10	0	1	1	27	99	420	2
YUV to AVC	1080p	10	0	1	2	27	100	328	2
YUV to HEVC	1080p	10	0	1	2	27	99	230	1
YUV to AVC	1080p	10	0	1	3	27	99	327	2
YUV to HEVC	1080p	10	0	1	3	27	100	157	1
YUV to AVC	1080p	10	4	0	1	27	100	450	3
YUV to HEVC	1080p	10	4	0	1	27	99	522	3
YUV to AV1	1080p	10	4	0	1	27	99	460	2
YUV to AVC	1080p	10	4	0	2	27	99	449	2
YUV to HEVC	1080p	10	4	0	2	27	99	330	1
YUV to AV1	1080p	10	4	0	2	27	100	270	1
YUV to AVC	1080p	10	4	0	3	27	99	450	3
YUV to HEVC	1080p	10	4	0	3	27	99	217	1
YUV to AV1	1080p	10	4	0	3	27	100	171	1
YUV to AVC	1080p	10	4	1	1	27	99	324	2
YUV to HEVC	1080p	10	4	1	1	27	99	420	2
YUV to AVC	1080p	10	4	1	2	27	99	323	2
YUV to HEVC	1080p	10	4	1	2	27	100	230	1
YUV to AVC	1080p	10	4	1	3	27	99	325	1
YUV to HEVC	1080p	10	4	1	3	27	99	157	1
YUV to AVC	1080p	10	16	0	1	27	99	448	2
YUV to HEVC	1080p	10	16	0	1	27	99	521	3
YUV to AV1	1080p	10	16	0	1	27	99	459	2
YUV to AVC	1080p	10	16	0	2	27	99	448	3
YUV to HEVC	1080p	10	16	0	2	27	99	327	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	2	27	99	270	2
YUV to AVC	1080p	10	16	0	3	27	99	447	2
YUV to HEVC	1080p	10	16	0	3	27	99	214	1
YUV to AV1	1080p	10	16	0	3	27	99	171	1
YUV to AVC	1080p	10	16	1	1	27	99	321	2
YUV to HEVC	1080p	10	16	1	1	27	100	420	2
YUV to AVC	1080p	10	16	1	2	27	99	322	2
YUV to HEVC	1080p	10	16	1	2	27	100	230	1
YUV to AVC	1080p	10	16	1	3	27	99	320	2
YUV to HEVC	1080p	10	16	1	3	27	100	155	1
YUV to AVC	1080p	10	40	0	1	27	99	440	2
YUV to HEVC	1080p	10	40	0	1	27	99	518	3
YUV to AV1	1080p	10	40	0	1	27	99	452	2
YUV to AVC	1080p	10	40	0	2	27	99	441	3
YUV to HEVC	1080p	10	40	0	2	27	99	320	2
YUV to AV1	1080p	10	40	0	2	27	99	270	1
YUV to AVC	1080p	10	40	0	3	27	99	440	2
YUV to HEVC	1080p	10	40	0	3	27	100	210	1
YUV to AV1	1080p	10	40	0	3	27	99	170	1
YUV to AVC	1080p	10	40	1	1	27	99	320	2
YUV to HEVC	1080p	10	40	1	1	27	100	410	2
YUV to AVC	1080p	10	40	1	2	27	99	320	2
YUV to HEVC	1080p	10	40	1	2	27	100	230	1
YUV to AVC	1080p	10	40	1	3	27	100	320	2
YUV to HEVC	1080p	10	40	1	3	27	100	153	1

## 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	21	3
AVC to HEVC	1	203	59	14	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	16	21	3
HEVC to AV1	1	172	59	13	20	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	18	21	2
AVC to AVC	8	536	26	71	92	17
AVC to HEVC	8	608	26	77	92	20
AVC to AV1	8	607	25	74	93	19
HEVC to AVC	8	560	29	66	93	18
HEVC to HEVC	8	638	29	72	93	20
HEVC to AV1	8	628	27	68	94	20
VP9 to AVC	8	538	19	76	94	18
VP9 to HEVC	8	615	26	84	94	20
VP9 to AV1	8	608	26	82	94	19
AVC to AVC	16	531	11	73	94	19
AVC to HEVC	16	608	10	77	92	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	74	95	22
HEVC to AV1	16	624	11	68	92	20
VP9 to AVC	16	535	10	76	93	18
VP9 to HEVC	16	608	11	84	94	21
VP9 to AV1	16	608	10	81	92	19

### 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	53	170	31
YUV to HEVC	4k	1	0	8	1	0	51	177	32
YUV to AV1	4k	1	0	8	1	0	57	172	37
YUV to AVC	4k	3	0	8	0	0	62	183	17
YUV to HEVC	4k	3	0	8	0	0	57	195	19
YUV to AV1	4k	3	0	8	0	0	64	192	18
YUV to AVC	4k	5	0	8	0	0	63	184	15
YUV to HEVC	4k	5	0	8	0	0	59	198	17
YUV to AV1	4k	5	0	8	0	0	65	194	17
YUV to AVC	4k	1	0	10	1	0	29	93	34
YUV to HEVC	4k	1	0	10	1	0	27	95	34
YUV to AV1	4k	1	0	10	1	0	31	93	37
YUV to AVC	4k	3	0	10	0	0	32	99	19
YUV to HEVC	4k	3	0	10	0	0	29	102	19
YUV to AV1	4k	3	0	10	0	0	33	102	19
YUV to AVC	1080p	1	0	8	1	0	37	483	20
YUV to HEVC	1080p	1	0	8	1	0	35	485	19
YUV to AV1	1080p	1	0	8	1	0	38	457	19
YUV to AVC	1080p	20	0	8	0	0	56	701	2
YUV to HEVC	1080p	20	0	8	0	0	53	720	2
YUV to AV1	1080p	20	0	8	0	0	60	700	2
YUV to AVC	1080p	1	0	10	1	0	25	307	26
YUV to HEVC	1080p	1	0	10	1	0	25	309	28
YUV to AV1	1080p	1	0	10	1	0	27	294	26
YUV to AVC	1080p	20	0	10	0	0	35	399	3
YUV to HEVC	1080p	20	0	10	0	0	35	400	3
YUV to AV1	1080p	20	0	10	0	0	38	380	3
YUV to AVC	720p	1	0	8	1	0	32	717	16
YUV to HEVC	720p	1	0	8	1	0	32	719	18
YUV to AV1	720p	1	0	8	1	0	35	655	10
YUV to AVC	720p	40	0	8	0	0	78	1400	1
YUV to HEVC	720p	40	0	8	0	0	77	1436	1
YUV to AV1	720p	40	0	8	0	0	93	1400	0

## 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	49	157	20
YUV to HEVC	4k	1	0	8	1	0	46	161	19
YUV to AV1	4k	1	0	8	1	0	51	153	19
YUV to AVC	4k	3	0	8	0	0	63	175	10
YUV to HEVC	4k	3	0	8	0	0	55	190	11
YUV to AV1	4k	3	0	8	0	0	61	183	11
YUV to AVC	4k	5	0	8	0	0	94	196	29
YUV to HEVC	4k	5	0	8	0	0	62	213	34
YUV to AV1	4k	5	0	8	0	0	67	195	34
YUV to AVC	4k	1	0	10	1	0	25	84	22
YUV to HEVC	4k	1	0	10	1	0	24	85	21
YUV to AV1	4k	1	0	10	1	0	27	82	20
YUV to AVC	4k	3	0	10	0	0	28	96	13
YUV to HEVC	4k	3	0	10	0	0	27	97	13
YUV to AV1	4k	3	0	10	0	0	30	93	14
YUV to AVC	1080p	1	0	8	1	0	37	482	15
YUV to HEVC	1080p	1	0	8	1	0	35	486	14
YUV to AV1	1080p	1	0	8	1	0	35	418	14
YUV to AVC	1080p	20	0	8	0	0	73	839	11
YUV to HEVC	1080p	20	0	8	0	0	65	871	11
YUV to AV1	1080p	20	0	8	0	0	70	827	10
YUV to AVC	1080p	1	0	10	1	0	23	289	18
YUV to HEVC	1080p	1	0	10	1	0	23	292	18
YUV to AV1	1080p	1	0	10	1	0	24	265	15
YUV to AVC	1080p	20	0	10	0	0	37	471	12
YUV to HEVC	1080p	20	0	10	0	0	35	473	12
YUV to AV1	1080p	20	0	10	0	0	38	446	12
YUV to AVC	720p	1	0	8	1	0	33	760	12
YUV to HEVC	720p	1	0	8	1	0	33	768	12
YUV to AV1	720p	1	0	8	1	0	32	614	11
YUV to AVC	720p	40	0	8	0	0	86	1706	5
YUV to HEVC	720p	40	0	8	0	0	83	1805	6
YUV to AV1	720p	40	0	8	0	0	88	1566	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	17.79	20.03	17.52	0.06
YUV to HEVC	4k	1	19.19	21.29	18.12	0.2
YUV to AV1	4k	1	24.31	28.13	18.89	0.65
YUV to AVC	4k	3	21.32	25.23	17.62	1.21
YUV to HEVC	4k	3	21.83	27.47	18.39	1.56
YUV to AV1	4k	3	25.72	32.63	19.1	1.7
YUV to AVC	4k	5	24.27	37.63	17.79	16.29
YUV to HEVC	4k	5	25.65	38.21	20.63	10.05
YUV to AV1	4k	5	29.96	37.65	21.27	5.91
YUV to AVC	1080p	1	5.29	6.25	5.13	0.02
YUV to HEVC	1080p	1	5.59	8.82	5.24	0.03
YUV to AV1	1080p	1	7.36	8.14	5.6	0.05
YUV to AVC	1080p	20	9.56	22.24	5.59	2.12
YUV to HEVC	1080p	20	10.16	20.23	6.3	2.42
YUV to AV1	1080p	20	12.92	25.8	7.82	5.16
YUV to AVC	720p	1	3.1	3.74	2.76	0.01
YUV to HEVC	720p	1	3.1	3.46	2.99	0
YUV to AV1	720p	1	4.08	4.44	3.21	0.01
YUV to AVC	720p	40	6.08	16.89	3.27	1.06
YUV to HEVC	720p	40	6.14	17.1	3.49	0.9
YUV to AV1	720p	40	9.11	17.82	6	1.45

## 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	26	83	64
RGBA to HEVC	4k	1	1	24	83	61
RGBA to AV1	4k	1	1	27	82	64
RGBA to AVC	4k	3	0	34	89	46
RGBA to HEVC	4k	3	0	33	90	52
RGBA to AV1	4k	3	0	34	89	52
RGBA to AVC	4k	5	0	45	91	36
RGBA to HEVC	4k	5	0	40	90	35
RGBA to AV1	4k	5	0	37	90	36
RGBA to AVC	1080p	1	1	20	259	69
RGBA to HEVC	1080p	1	1	19	258	64
RGBA to AV1	1080p	1	1	21	250	81
RGBA to AVC	1080p	10	0	34	331	25
RGBA to HEVC	1080p	10	0	30	332	24
RGBA to AV1	1080p	10	0	31	332	25
RGBA to AVC	1080p	20	0	34	331	13
RGBA to HEVC	1080p	20	0	34	339	13
RGBA to AV1	1080p	20	0	33	334	14
RGBA to AVC	720p	1	1	12	375	117
RGBA to HEVC	720p	1	1	14	357	114
RGBA to AV1	720p	1	1	17	362	111
RGBA to AVC	720p	10	0	34	698	48
RGBA to HEVC	720p	10	0	33	703	48
RGBA to AV1	720p	10	0	39	684	46
RGBA to AVC	720p	20	0	36	722	26
RGBA to HEVC	720p	20	0	35	723	20
RGBA to AV1	720p	20	0	43	717	22

## 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	152	59
YUV to HEVC	4k	2	0	0	1	48	166	54
YUV to AV1	4k	2	0	0	1	48	144	52
YUV to AVC	4k	2	0	0	2	48	154	58
YUV to HEVC	4k	2	0	0	2	49	86	36
YUV to AV1	4k	2	0	0	2	49	70	35
YUV to AVC	4k	2	0	0	3	48	154	56
YUV to HEVC	4k	2	0	0	3	49	50	25
YUV to AV1	4k	2	0	0	3	48	38	22
YUV to AVC	4k	2	0	1	1	48	94	43
YUV to HEVC	4k	2	0	1	1	48	120	48
YUV to AVC	4k	2	0	1	2	49	94	40
YUV to HEVC	4k	2	0	1	2	49	54	30
YUV to AVC	4k	2	0	1	3	49	94	42
YUV to HEVC	4k	2	0	1	3	49	34	23
YUV to AVC	4k	2	4	0	1	72	130	47
YUV to HEVC	4k	2	4	0	1	73	166	42
YUV to AV1	4k	2	4	0	1	70	139	51
YUV to AVC	4k	2	4	0	2	73	132	50
YUV to HEVC	4k	2	4	0	2	61	84	36
YUV to AV1	4k	2	4	0	2	60	70	31
YUV to AVC	4k	2	4	0	3	74	134	47
YUV to HEVC	4k	2	4	0	3	56	50	28
YUV to AV1	4k	2	4	0	3	55	40	25
YUV to AVC	4k	2	4	1	1	67	94	38
YUV to HEVC	4k	2	4	1	1	66	120	45
YUV to AVC	4k	2	4	1	2	67	92	41
YUV to HEVC	4k	2	4	1	2	58	54	26
YUV to AVC	4k	2	4	1	3	67	94	38
YUV to HEVC	4k	2	4	1	3	53	34	22
YUV to AVC	4k	2	20	0	1	73	132	47
YUV to HEVC	4k	2	20	0	1	73	166	47
YUV to AV1	4k	2	20	0	1	72	144	50

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	20	0	2	72	132	50
YUV to HEVC	4k	2	20	0	2	61	84	40
YUV to AV1	4k	2	20	0	2	60	70	34
YUV to AVC	4k	2	20	0	3	73	132	50
YUV to HEVC	4k	2	20	0	3	57	50	28
YUV to AV1	4k	2	20	0	3	55	40	22
YUV to AVC	4k	2	20	1	1	67	92	38
YUV to HEVC	4k	2	20	1	1	66	120	45
YUV to AVC	4k	2	20	1	2	67	92	41
YUV to HEVC	4k	2	20	1	2	57	54	27
YUV to AVC	4k	2	20	1	3	67	92	38
YUV to HEVC	4k	2	20	1	3	53	34	21
YUV to AVC	4k	2	40	0	1	74	130	50
YUV to HEVC	4k	2	40	0	1	72	164	49
YUV to AV1	4k	2	40	0	1	72	142	52
YUV to AVC	4k	2	40	0	2	73	130	48
YUV to HEVC	4k	2	40	0	2	61	84	38
YUV to AV1	4k	2	40	0	2	60	70	33
YUV to AVC	4k	2	40	0	3	73	130	49
YUV to HEVC	4k	2	40	0	3	56	50	27
YUV to AV1	4k	2	40	0	3	56	40	24
YUV to AVC	4k	2	40	1	1	67	92	41
YUV to HEVC	4k	2	40	1	1	66	118	44
YUV to AVC	4k	2	40	1	2	66	92	40
YUV to HEVC	4k	2	40	1	2	58	54	28
YUV to AVC	4k	2	40	1	3	68	92	40
YUV to HEVC	4k	2	40	1	3	55	34	20
YUV to AVC	1080p	10	0	0	1	67	784	24
YUV to HEVC	1080p	10	0	0	1	62	819	26
YUV to AV1	1080p	10	0	0	1	66	771	25
YUV to AVC	1080p	10	0	0	2	67	789	25
YUV to HEVC	1080p	10	0	0	2	99	689	23
YUV to AV1	1080p	10	0	0	2	100	557	21
YUV to AVC	1080p	10	0	0	3	68	789	25

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	100	408	19
YUV to AV1	1080p	10	0	0	3	100	298	17
YUV to AVC	1080p	10	0	1	1	99	754	23
YUV to HEVC	1080p	10	0	1	1	86	833	26
YUV to AVC	1080p	10	0	1	2	100	753	23
YUV to HEVC	1080p	10	0	1	2	100	437	19
YUV to AVC	1080p	10	0	1	3	100	753	24
YUV to HEVC	1080p	10	0	1	3	99	278	17
YUV to AVC	1080p	10	4	0	1	100	649	19
YUV to HEVC	1080p	10	4	0	1	88	713	21
YUV to AV1	1080p	10	4	0	1	94	683	21
YUV to AVC	1080p	10	4	0	2	99	649	21
YUV to HEVC	1080p	10	4	0	2	100	510	20
YUV to AV1	1080p	10	4	0	2	99	430	18
YUV to AVC	1080p	10	4	0	3	99	647	20
YUV to HEVC	1080p	10	4	0	3	99	339	18
YUV to AV1	1080p	10	4	0	3	100	270	16
YUV to AVC	1080p	10	4	1	1	100	510	20
YUV to HEVC	1080p	10	4	1	1	99	653	21
YUV to AVC	1080p	10	4	1	2	100	510	21
YUV to HEVC	1080p	10	4	1	2	99	360	19
YUV to AVC	1080p	10	4	1	3	100	510	20
YUV to HEVC	1080p	10	4	1	3	100	245	17
YUV to AVC	1080p	10	20	0	1	99	641	20
YUV to HEVC	1080p	10	20	0	1	87	712	22
YUV to AV1	1080p	10	20	0	1	93	670	19
YUV to AVC	1080p	10	20	0	2	99	640	19
YUV to HEVC	1080p	10	20	0	2	100	511	20
YUV to AV1	1080p	10	20	0	2	100	430	17
YUV to AVC	1080p	10	20	0	3	99	640	20
YUV to HEVC	1080p	10	20	0	3	100	338	18
YUV to AV1	1080p	10	20	0	3	99	270	17
YUV to AVC	1080p	10	20	1	1	100	507	20
YUV to HEVC	1080p	10	20	1	1	99	652	21

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	20	1	2	99	508	21
YUV to HEVC	1080p	10	20	1	2	100	360	19
YUV to AVC	1080p	10	20	1	3	99	507	20
YUV to HEVC	1080p	10	20	1	3	100	244	16
YUV to AVC	1080p	10	40	0	1	100	631	19
YUV to HEVC	1080p	10	40	0	1	89	706	21
YUV to AV1	1080p	10	40	0	1	91	647	19
YUV to AVC	1080p	10	40	0	2	99	630	19
YUV to HEVC	1080p	10	40	0	2	99	507	20
YUV to AV1	1080p	10	40	0	2	99	420	17
YUV to AVC	1080p	10	40	0	3	99	630	19
YUV to HEVC	1080p	10	40	0	3	100	336	18
YUV to AV1	1080p	10	40	0	3	100	268	17
YUV to AVC	1080p	10	40	1	1	100	500	20
YUV to HEVC	1080p	10	40	1	1	99	647	21
YUV to AVC	1080p	10	40	1	2	99	501	19
YUV to HEVC	1080p	10	40	1	2	99	360	18
YUV to AVC	1080p	10	40	1	3	99	501	21
YUV to HEVC	1080p	10	40	1	3	100	240	16
YUV to AVC	720p	20	0	0	1	76	1487	13
YUV to HEVC	720p	20	0	0	1	73	1569	14
YUV to AV1	720p	20	0	0	1	79	1414	13
YUV to AVC	720p	20	0	0	2	79	1506	14
YUV to HEVC	720p	20	0	0	2	99	1501	16
YUV to AV1	720p	20	0	0	2	99	1220	16
YUV to AVC	720p	20	0	0	3	76	1502	14
YUV to HEVC	720p	20	0	0	3	99	901	13
YUV to AV1	720p	20	0	0	3	100	660	10
YUV to AVC	720p	20	0	1	1	92	1567	15
YUV to HEVC	720p	20	0	1	1	76	1597	14
YUV to AVC	720p	20	0	1	2	93	1586	14
YUV to HEVC	720p	20	0	1	2	100	960	13
YUV to AVC	720p	20	0	1	3	93	1589	15
YUV to HEVC	720p	20	0	1	3	100	620	7

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	20	4	0	1	99	1099	10
YUV to HEVC	720p	20	4	0	1	99	1104	10
YUV to AV1	720p	20	4	0	1	100	877	7
YUV to AVC	720p	20	4	0	2	98	1098	9
YUV to HEVC	720p	20	4	0	2	99	1080	12
YUV to AV1	720p	20	4	0	2	99	864	7
YUV to AVC	720p	20	4	0	3	97	1092	10
YUV to HEVC	720p	20	4	0	3	99	740	10
YUV to AV1	720p	20	4	0	3	100	560	14
YUV to AVC	720p	20	4	1	1	99	1067	13
YUV to HEVC	720p	20	4	1	1	100	1117	9
YUV to AVC	720p	20	4	1	2	99	1066	12
YUV to HEVC	720p	20	4	1	2	99	760	9
YUV to AVC	720p	20	4	1	3	99	1067	13
YUV to HEVC	720p	20	4	1	3	100	524	8
YUV to AVC	720p	20	20	0	1	100	1072	9
YUV to HEVC	720p	20	20	0	1	99	1108	9
YUV to AV1	720p	20	20	0	1	100	905	7
YUV to AVC	720p	20	20	0	2	99	1065	9
YUV to HEVC	720p	20	20	0	2	99	1078	12
YUV to AV1	720p	20	20	0	2	99	880	11
YUV to AVC	720p	20	20	0	3	99	1077	9
YUV to HEVC	720p	20	20	0	3	99	726	11
YUV to AV1	720p	20	20	0	3	99	560	9
YUV to AVC	720p	20	20	1	1	99	1063	13
YUV to HEVC	720p	20	20	1	1	99	1118	9
YUV to AVC	720p	20	20	1	2	99	1061	13
YUV to HEVC	720p	20	20	1	2	99	760	13
YUV to AVC	720p	20	20	1	3	99	1062	12
YUV to HEVC	720p	20	20	1	3	99	520	9
YUV to AVC	720p	20	40	0	1	100	1049	9
YUV to HEVC	720p	20	40	0	1	99	1082	9
YUV to AV1	720p	20	40	0	1	100	900	7
YUV to AVC	720p	20	40	0	2	99	1052	9

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	720p	20	40	0	2	99	1060	13
YUV to AV1	720p	20	40	0	2	99	876	11
YUV to AVC	720p	20	40	0	3	98	1058	9
YUV to HEVC	720p	20	40	0	3	99	720	9
YUV to AV1	720p	20	40	0	3	99	560	11
YUV to AVC	720p	20	40	1	1	99	1059	13
YUV to HEVC	720p	20	40	1	1	100	1092	10
YUV to AVC	720p	20	40	1	2	99	1059	13
YUV to HEVC	720p	20	40	1	2	99	760	13
YUV to AVC	720p	20	40	1	3	99	1060	12
YUV to HEVC	720p	20	40	1	3	99	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	94	647	4
YUV to HEVC	1080p	10	0	0	1	19	83	680	4
YUV to AV1	1080p	10	0	0	1	19	90	641	4
YUV to AVC	1080p	10	0	0	2	19	94	646	4
YUV to HEVC	1080p	10	0	0	2	19	100	512	2
YUV to AV1	1080p	10	0	0	2	19	99	430	2
YUV to AVC	1080p	10	0	0	3	19	94	646	4
YUV to HEVC	1080p	10	0	0	3	19	99	339	2
YUV to AV1	1080p	10	0	0	3	19	99	278	1
YUV to AVC	1080p	10	0	1	1	19	99	511	3
YUV to HEVC	1080p	10	0	1	1	19	99	658	4
YUV to AVC	1080p	10	0	1	2	19	99	510	3
YUV to HEVC	1080p	10	0	1	2	19	99	360	2
YUV to AVC	1080p	10	0	1	3	19	100	510	3
YUV to HEVC	1080p	10	0	1	3	19	100	246	1
YUV to AVC	1080p	10	4	0	1	19	89	591	3
YUV to HEVC	1080p	10	4	0	1	19	84	626	4
YUV to AV1	1080p	10	4	0	1	19	92	581	3
YUV to AVC	1080p	10	4	0	2	19	90	594	4
YUV to HEVC	1080p	10	4	0	2	19	100	512	3
YUV to AV1	1080p	10	4	0	2	19	100	430	2
YUV to AVC	1080p	10	4	0	3	19	90	594	3
YUV to HEVC	1080p	10	4	0	3	19	100	339	2
YUV to AV1	1080p	10	4	0	3	19	99	270	2
YUV to AVC	1080p	10	4	1	1	19	100	510	3
YUV to HEVC	1080p	10	4	1	1	19	94	630	4
YUV to AVC	1080p	10	4	1	2	19	99	510	3
YUV to HEVC	1080p	10	4	1	2	19	99	360	2
YUV to AVC	1080p	10	4	1	3	19	99	510	3
YUV to HEVC	1080p	10	4	1	3	19	100	245	1
YUV to AVC	1080p	10	16	0	1	19	82	550	3
YUV to HEVC	1080p	10	16	0	1	19	84	614	4
YUV to AV1	1080p	10	16	0	1	19	92	566	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	0	2	19	83	550	3
YUV to HEVC	1080p	10	16	0	2	19	99	510	3
YUV to AV1	1080p	10	16	0	2	19	99	428	2
YUV to AVC	1080p	10	16	0	3	19	82	551	3
YUV to HEVC	1080p	10	16	0	3	19	99	340	2
YUV to AV1	1080p	10	16	0	3	19	99	270	1
YUV to AVC	1080p	10	16	1	1	19	99	507	3
YUV to HEVC	1080p	10	16	1	1	19	94	618	4
YUV to AVC	1080p	10	16	1	2	19	99	507	3
YUV to HEVC	1080p	10	16	1	2	19	100	360	2
YUV to AVC	1080p	10	16	1	3	19	99	507	3
YUV to HEVC	1080p	10	16	1	3	19	99	243	1
YUV to AVC	1080p	10	40	0	1	19	81	536	3
YUV to HEVC	1080p	10	40	0	1	19	86	590	3
YUV to AV1	1080p	10	40	0	1	19	91	537	3
YUV to AVC	1080p	10	40	0	2	19	83	538	3
YUV to HEVC	1080p	10	40	0	2	19	100	501	2
YUV to AV1	1080p	10	40	0	2	19	99	420	2
YUV to AVC	1080p	10	40	0	3	19	84	537	3
YUV to HEVC	1080p	10	40	0	3	19	99	332	2
YUV to AV1	1080p	10	40	0	3	19	100	268	2
YUV to AVC	1080p	10	40	1	1	19	99	500	3
YUV to HEVC	1080p	10	40	1	1	19	92	593	3
YUV to AVC	1080p	10	40	1	2	19	99	500	3
YUV to HEVC	1080p	10	40	1	2	19	100	353	2
YUV to AVC	1080p	10	40	1	3	19	99	500	2
YUV to HEVC	1080p	10	40	1	3	19	99	241	1
YUV to AVC	1080p	10	0	0	1	23	94	647	4
YUV to HEVC	1080p	10	0	0	1	23	83	680	4
YUV to AV1	1080p	10	0	0	1	23	89	641	4
YUV to AVC	1080p	10	0	0	2	23	94	650	4
YUV to HEVC	1080p	10	0	0	2	23	99	511	3
YUV to AV1	1080p	10	0	0	2	23	99	430	2
YUV to AVC	1080p	10	0	0	3	23	94	648	4



TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	23	100	340	2
YUV to AV1	1080p	10	0	0	3	23	99	278	1
YUV to AVC	1080p	10	0	1	1	23	100	510	3
YUV to HEVC	1080p	10	0	1	1	23	99	656	4
YUV to AVC	1080p	10	0	1	2	23	99	510	3
YUV to HEVC	1080p	10	0	1	2	23	99	360	2
YUV to AVC	1080p	10	0	1	3	23	100	510	3
YUV to HEVC	1080p	10	0	1	3	23	100	247	1
YUV to AVC	1080p	10	4	0	1	23	91	598	4
YUV to HEVC	1080p	10	4	0	1	23	84	639	4
YUV to AV1	1080p	10	4	0	1	23	91	591	3
YUV to AVC	1080p	10	4	0	2	23	91	600	3
YUV to HEVC	1080p	10	4	0	2	23	100	510	3
YUV to AV1	1080p	10	4	0	2	23	99	430	2
YUV to AVC	1080p	10	4	0	3	23	90	600	3
YUV to HEVC	1080p	10	4	0	3	23	100	339	2
YUV to AV1	1080p	10	4	0	3	23	99	270	1
YUV to AVC	1080p	10	4	1	1	23	100	510	3
YUV to HEVC	1080p	10	4	1	1	23	96	635	4
YUV to AVC	1080p	10	4	1	2	23	100	509	3
YUV to HEVC	1080p	10	4	1	2	23	100	360	2
YUV to AVC	1080p	10	4	1	3	23	99	510	3
YUV to HEVC	1080p	10	4	1	3	23	100	245	1
YUV to AVC	1080p	10	16	0	1	23	86	571	3
YUV to HEVC	1080p	10	16	0	1	23	84	624	4
YUV to AV1	1080p	10	16	0	1	23	91	579	3
YUV to AVC	1080p	10	16	0	2	23	87	571	3
YUV to HEVC	1080p	10	16	0	2	23	100	510	3
YUV to AV1	1080p	10	16	0	2	23	100	429	2
YUV to AVC	1080p	10	16	0	3	23	88	570	3
YUV to HEVC	1080p	10	16	0	3	23	99	339	2
YUV to AV1	1080p	10	16	0	3	23	100	270	2
YUV to AVC	1080p	10	16	1	1	23	99	506	2
YUV to HEVC	1080p	10	16	1	1	23	95	627	4

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	1	2	23	99	506	3
YUV to HEVC	1080p	10	16	1	2	23	100	360	2
YUV to AVC	1080p	10	16	1	3	23	100	509	3
YUV to HEVC	1080p	10	16	1	3	23	100	243	1
YUV to AVC	1080p	10	40	0	1	23	84	559	3
YUV to HEVC	1080p	10	40	0	1	23	86	601	3
YUV to AV1	1080p	10	40	0	1	23	91	553	3
YUV to AVC	1080p	10	40	0	2	23	87	561	3
YUV to HEVC	1080p	10	40	0	2	23	99	503	2
YUV to AV1	1080p	10	40	0	2	23	100	420	2
YUV to AVC	1080p	10	40	0	3	23	87	560	3
YUV to HEVC	1080p	10	40	0	3	23	100	333	2
YUV to AV1	1080p	10	40	0	3	23	100	270	2
YUV to AVC	1080p	10	40	1	1	23	100	500	3
YUV to HEVC	1080p	10	40	1	1	23	94	609	4
YUV to AVC	1080p	10	40	1	2	23	100	499	3
YUV to HEVC	1080p	10	40	1	2	23	99	354	2
YUV to AVC	1080p	10	40	1	3	23	99	500	2
YUV to HEVC	1080p	10	40	1	3	23	100	240	1
YUV to AVC	1080p	10	0	0	1	27	94	649	4
YUV to HEVC	1080p	10	0	0	1	27	83	680	4
YUV to AV1	1080p	10	0	0	1	27	88	640	4
YUV to AVC	1080p	10	0	0	2	27	93	646	4
YUV to HEVC	1080p	10	0	0	2	27	99	511	2
YUV to AV1	1080p	10	0	0	2	27	99	430	2
YUV to AVC	1080p	10	0	0	3	27	94	649	4
YUV to HEVC	1080p	10	0	0	3	27	99	339	2
YUV to AV1	1080p	10	0	0	3	27	99	278	2
YUV to AVC	1080p	10	0	1	1	27	99	510	3
YUV to HEVC	1080p	10	0	1	1	27	99	656	4
YUV to AVC	1080p	10	0	1	2	27	99	510	3
YUV to HEVC	1080p	10	0	1	2	27	100	360	2
YUV to AVC	1080p	10	0	1	3	27	99	510	3
YUV to HEVC	1080p	10	0	1	3	27	99	244	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	4	0	1	27	91	601	3
YUV to HEVC	1080p	10	4	0	1	27	82	641	4
YUV to AV1	1080p	10	4	0	1	27	90	602	3
YUV to AVC	1080p	10	4	0	2	27	92	600	3
YUV to HEVC	1080p	10	4	0	2	27	99	510	3
YUV to AV1	1080p	10	4	0	2	27	100	430	2
YUV to AVC	1080p	10	4	0	3	27	91	600	3
YUV to HEVC	1080p	10	4	0	3	27	100	340	2
YUV to AV1	1080p	10	4	0	3	27	100	270	1
YUV to AVC	1080p	10	4	1	1	27	99	510	3
YUV to HEVC	1080p	10	4	1	1	27	98	645	4
YUV to AVC	1080p	10	4	1	2	27	100	510	3
YUV to HEVC	1080p	10	4	1	2	27	99	360	2
YUV to AVC	1080p	10	4	1	3	27	99	510	3
YUV to HEVC	1080p	10	4	1	3	27	100	246	1
YUV to AVC	1080p	10	16	0	1	27	89	588	3
YUV to HEVC	1080p	10	16	0	1	27	83	632	4
YUV to AV1	1080p	10	16	0	1	27	90	590	3
YUV to AVC	1080p	10	16	0	2	27	92	591	3
YUV to HEVC	1080p	10	16	0	2	27	100	510	2
YUV to AV1	1080p	10	16	0	2	27	99	428	2
YUV to AVC	1080p	10	16	0	3	27	89	590	4
YUV to HEVC	1080p	10	16	0	3	27	100	339	2
YUV to AV1	1080p	10	16	0	3	27	100	270	2
YUV to AVC	1080p	10	16	1	1	27	99	505	3
YUV to HEVC	1080p	10	16	1	1	27	97	638	4
YUV to AVC	1080p	10	16	1	2	27	100	508	3
YUV to HEVC	1080p	10	16	1	2	27	100	360	2
YUV to AVC	1080p	10	16	1	3	27	99	507	3
YUV to HEVC	1080p	10	16	1	3	27	99	243	1
YUV to AVC	1080p	10	40	0	1	27	87	575	3
YUV to HEVC	1080p	10	40	0	1	27	84	611	3
YUV to AV1	1080p	10	40	0	1	27	91	564	3
YUV to AVC	1080p	10	40	0	2	27	89	574	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	0	2	27	99	503	3
YUV to AV1	1080p	10	40	0	2	27	99	420	2
YUV to AVC	1080p	10	40	0	3	27	88	571	3
YUV to HEVC	1080p	10	40	0	3	27	100	333	2
YUV to AV1	1080p	10	40	0	3	27	99	269	2
YUV to AVC	1080p	10	40	1	1	27	99	500	3
YUV to HEVC	1080p	10	40	1	1	27	95	617	4
YUV to AVC	1080p	10	40	1	2	27	100	500	3
YUV to HEVC	1080p	10	40	1	2	27	100	353	2
YUV to AVC	1080p	10	40	1	3	27	99	500	3
YUV to HEVC	1080p	10	40	1	3	27	99	240	1

## Appendix A: GStreamer XStack Command

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

```
gst-launch-1.0 multifilesrc location=/media/ramdisk/input.h265 loop=true !
h265parse ! niquadrah265dec xcoder-params='out=hw' ! niquadrastack
inputs=32
layout="0_0|w0_0|w0+w1_0|w0+w1+w2_0|0_h0|w0_h0|w0+w1_h0|w0+w1+w2_h0|0_h0+h1|w
0_h0+h1|w0+w1_h0+h1|w0+w1+w2_h0+h1|0_h0+h1+h2|w0_h0+h1+h2|w0+w1_h0+h1+h2|w0+w
1+w2_h0+h1+h2|0_h0+h1+h2+h3|w0_h0+h1+h2+h3|w0+w1_h0+h1+h2+h3|w0+w1+w2_h0+h1+h
2+h3|0_h0+h1+h2+h3|w0_h0+h1+h2+h3|w0+w1_h0+h1+h2+h3|w0+w1+w2_h0+h1+h2+h3|0_h0
+h1+h2+h3+h4|w0_h0+h1+h2+h3+h4|w0+w1_h0+h1+h2+h3+h4|w0+w1+w2_h0+h1+h2+h3+h4|0
_h0+h1+h2+h3+h4+h5|w0_h0+h1+h2+h3+h4+h5|w0+w1_h0+h1+h2+h3+h4+h5|w0+w1+w2_h0+h
1+h2+h3+h4+h5|0_h0+h1+h2+h3+h4+h5+h6|w0_h0+h1+h2+h3+h4+h5+h6|w0+w1_h0+h1+h2+h
3+h4+h5+h6|w0+w1+w2_h0+h1+h2+h3+h4+h5+h6"
size="480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135
|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_
135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|4
80_135|480_135|480_135|480_135" shortest=1 name=xstack ! niquadrah264enc !
```

[illegible]

```
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.
```

## 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	

### Example of single input with 64 outputs

Page 136 of 137



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