



Quadra™ Performance Test Report V5.4

Contents

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

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

Environment Overview

Revision: 5306sBr2

Setup #1:

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

Setup #2:

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

Setup #3:

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

Setup #4:

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

Definitions

- CPU: Average per instance CPU usage.
 - $(\text{System-wide CPU usage} * \text{number of CPU}) / (\text{number of devices} * \text{number of instances per device})$
- FPS: Average of all FPS reported 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	71	0	115	9
HEVC to YUV	8k	1	0	8	1	96	0	115	12
VP9 to YUV	8k	1	0	8	1	24	0	38	4
YUV to AVC	8k	1	0	8	1	0	100	67	52
YUV to HEVC	8k	1	0	8	1	0	96	83	68
AVC to AVC	8k	1	1	8	1	67	99	54	4
AVC to HEVC	8k	1	1	8	1	82	98	71	4
HEVC to AVC	8k	1	1	8	1	61	100	53	6
HEVC to HEVC	8k	1	1	8	1	70	100	71	6
VP9 to AVC	8k	1	1	8	1	24	43	35	2
VP9 to HEVC	8k	1	1	8	1	24	43	37	4
AVC to YUV	8k	1	0	10	1	55	0	59	8
HEVC to YUV	8k	1	0	10	1	100	0	62	9
VP9 to YUV	8k	1	0	10	1	25	0	33	14
YUV to AVC	8k	1	0	10	1	0	93	48	74
YUV to HEVC	8k	1	0	10	1	0	72	59	92
AVC to YUV	4k	1	0	8	1	55	0	317	18
HEVC to YUV	4k	1	0	8	1	48	0	339	24
VP9 to YUV	4k	1	0	8	1	24	0	155	7
AVC to YUV	4k	16	0	8	0	98	0	480	3
HEVC to YUV	4k	16	0	8	0	100	0	506	2
VP9 to YUV	4k	16	0	8	0	98	0	486	1
YUV to AVC	4k	1	0	8	1	0	95	294	43
YUV to HEVC	4k	1	0	8	1	0	95	327	46
YUV to AV1	4k	1	0	8	1	0	94	282	39
YUV to AVC	4k	4	0	8	0	0	96	304	15
YUV to HEVC	4k	4	0	8	0	0	96	332	14
YUV to AV1	4k	4	0	8	0	0	96	286	12
YUV to AVC	4k	8	0	8	0	0	100	320	8
YUV to HEVC	4k	8	0	8	0	0	100	344	9
YUV to AV1	4k	8	0	8	0	0	100	296	8
AVC to AVC	4k	1	1	8	1	67	93	221	13
AVC to HEVC	4k	1	1	8	1	71	91	271	15
AVC to AV1	4k	1	1	8	1	61	93	258	12
HEVC to AVC	4k	1	1	8	1	56	97	219	15
HEVC to HEVC	4k	1	1	8	1	58	92	281	18
HEVC to AV1	4k	1	1	8	1	48	92	264	18
VP9 to AVC	4k	1	1	8	1	24	45	145	7
VP9 to HEVC	4k	1	1	8	1	24	43	149	6

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	51	151	7
AVC to AVC	4k	4	1	8	0	62	95	236	3
AVC to HEVC	4k	4	1	8	0	68	96	300	3
AVC to AV1	4k	4	1	8	0	55	95	276	4
HEVC to AVC	4k	4	1	8	0	57	97	240	9
HEVC to HEVC	4k	4	1	8	0	60	96	300	3
HEVC to AV1	4k	4	1	8	0	51	95	268	4
VP9 to AVC	4k	4	1	8	0	65	97	242	3
VP9 to HEVC	4k	4	1	8	0	65	96	301	3
VP9 to AV1	4k	4	1	8	0	59	92	252	3
AVC to AVC	4k	8	1	8	0	68	99	211	2
AVC to HEVC	4k	8	1	8	0	76	100	280	3
AVC to AV1	4k	8	1	8	0	67	100	272	3
HEVC to AVC	4k	8	1	8	0	64	99	216	2
HEVC to HEVC	4k	8	1	8	0	68	99	280	3
HEVC to AV1	4k	8	1	8	0	58	100	276	3
VP9 to AVC	4k	8	1	8	0	67	99	235	2
VP9 to HEVC	4k	8	1	8	0	72	100	297	2
VP9 to AV1	4k	8	1	8	0	62	100	280	2
AVC to YUV	4k	1	0	10	0	46	0	226	8
HEVC to YUV	4k	1	0	10	0	51	0	232	9
VP9 to YUV	4k	1	0	10	0	24	0	158	6
AVC to YUV	4k	16	0	10	0	98	0	284	0
HEVC to YUV	4k	16	0	10	0	98	0	282	0
VP9 to YUV	4k	16	0	10	0	99	0	500	0
YUV to AVC	4k	1	0	10	0	0	69	199	57
YUV to HEVC	4k	1	0	10	0	0	60	206	58
YUV to AV1	4k	1	0	10	0	0	67	199	61
YUV to AVC	4k	4	0	10	0	0	91	219	26
YUV to HEVC	4k	4	0	10	0	0	73	248	36
YUV to AV1	4k	4	0	10	0	0	82	240	40
AVC to YUV	1080p	1	0	8	1	41	0	865	21
HEVC to YUV	1080p	1	0	8	1	43	0	845	27
VP9 to YUV	1080p	1	0	8	1	22	0	558	7
AVC to YUV	1080p	40	0	8	0	86	0	1723	1
HEVC to YUV	1080p	40	0	8	0	96	0	1828	1
VP9 to YUV	1080p	40	0	8	0	82	0	1789	0
YUV to AVC	1080p	1	0	8	1	0	54	702	28
YUV to HEVC	1080p	1	0	8	1	0	51	710	27
YUV to AV1	1080p	1	0	8	1	0	54	648	28

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	100	1280	3
YUV to HEVC	1080p	32	0	8	0	0	99	1349	3
YUV to AV1	1080p	32	0	8	0	0	100	1180	3
AVC to AVC	1080p	1	1	8	1	66	81	961	22
AVC to HEVC	1080p	1	1	8	1	67	78	1025	25
AVC to AV1	1080p	1	1	8	1	53	80	900	21
HEVC to AVC	1080p	1	1	8	1	55	76	899	31
HEVC to HEVC	1080p	1	1	8	1	56	76	970	33
HEVC to AV1	1080p	1	1	8	1	47	76	852	30
VP9 to AVC	1080p	1	1	8	1	21	42	547	8
VP9 to HEVC	1080p	1	1	8	1	22	39	543	8
VP9 to AV1	1080p	1	1	8	1	22	46	542	7
AVC to AVC	1080p	32	1	8	0	76	99	938	2
AVC to HEVC	1080p	32	1	8	0	83	99	1062	1
AVC to AV1	1080p	32	1	8	0	77	99	1024	1
HEVC to AVC	1080p	32	1	8	0	68	99	992	1
HEVC to HEVC	1080p	32	1	8	0	77	100	1120	1
HEVC to AV1	1080p	32	1	8	0	69	99	1056	1
VP9 to AVC	1080p	32	1	8	0	71	99	1088	1
VP9 to HEVC	1080p	32	1	8	0	74	100	1216	1
VP9 to AV1	1080p	32	1	8	0	71	99	1120	1
AVC to YUV	1080p	1	0	10	0	29	0	659	8
HEVC to YUV	1080p	1	0	10	0	26	0	695	9
VP9 to YUV	1080p	1	0	10	0	22	0	456	7
AVC to YUV	1080p	40	0	10	0	66	0	1084	0
HEVC to YUV	1080p	40	0	10	0	71	0	1080	0
VP9 to YUV	1080p	40	0	10	0	73	0	1059	0
YUV to AVC	1080p	1	0	10	0	0	39	505	44
YUV to HEVC	1080p	1	0	10	0	0	36	502	37
YUV to AV1	1080p	1	0	10	0	0	39	470	37
YUV to AVC	1080p	32	0	10	0	0	61	800	7
YUV to HEVC	1080p	32	0	10	0	0	57	800	6
YUV to AV1	1080p	32	0	10	0	0	65	771	6
AVC to YUV	720p	1	0	8	1	43	0	1210	16
HEVC to YUV	720p	1	0	8	1	38	0	1216	25
VP9 to YUV	720p	1	0	8	1	31	0	1023	10
AVC to YUV	720p	100	0	8	0	100	0	2530	0
HEVC to YUV	720p	100	0	8	0	100	0	2912	1
VP9 to YUV	720p	100	0	8	0	99	0	2613	0
YUV to AVC	720p	1	0	8	1	0	30	886	19

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	30	895	20
YUV to AV1	720p	1	0	8	1	0	34	805	15
YUV to AVC	720p	64	0	8	0	0	94	2368	2
YUV to HEVC	720p	64	0	8	0	0	93	2378	1
YUV to AV1	720p	64	0	8	0	0	98	2049	1
AVC to AVC	720p	1	1	8	1	49	48	1289	21
AVC to HEVC	720p	1	1	8	1	48	47	1292	25
AVC to AV1	720p	1	1	8	1	41	51	1113	19
HEVC to AVC	720p	1	1	8	1	41	45	1241	27
HEVC to HEVC	720p	1	1	8	1	41	46	1247	28
HEVC to AV1	720p	1	1	8	1	35	49	1079	22
VP9 to AVC	720p	1	1	8	1	32	35	1005	14
VP9 to HEVC	720p	1	1	8	1	31	34	991	12
VP9 to AV1	720p	1	1	8	1	31	43	991	14
AVC to AVC	720p	64	1	8	0	100	99	2057	0
AVC to HEVC	720p	64	1	8	0	99	100	2129	0
AVC to AV1	720p	64	1	8	0	77	100	1798	0
HEVC to AVC	720p	64	1	8	0	89	99	2073	0
HEVC to HEVC	720p	64	1	8	0	86	100	2163	0
HEVC to AV1	720p	64	1	8	0	69	100	1803	0
VP9 to AVC	720p	64	1	8	0	100	100	2304	0
VP9 to HEVC	720p	64	1	8	0	100	100	2368	0
VP9 to AV1	720p	64	1	8	0	79	100	1926	0

2. T1A – Libxcoder Throughput

2.1 Decoding

2.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

2.1.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

2.2 Encoding

2.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

2.2.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

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

2.3 Transcoding

2.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

2.3.2 Command line

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

<dec_test_type> = decoding test codecs. ie. a (avc), h (hevc), etc

<enc_test_type> = encoding test codecs. ie. a (avc), h (hevc), etc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

2.4 Libxcode Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	67	0	93	7
HEVC to YUV	8k	1	0	8	1	61	0	97	7
VP9 to YUV	8k	1	0	8	1	24	0	38	3
YUV to AVC	8k	1	0	8	1	0	96	68	27
YUV to HEVC	8k	1	0	8	1	0	96	82	32
AVC to AVC	8k	1	1	8	1	73	100	54	6
AVC to HEVC	8k	1	1	8	1	82	99	71	6
HEVC to AVC	8k	1	1	8	1	65	100	53	5
HEVC to HEVC	8k	1	1	8	1	73	99	70	6
VP9 to AVC	8k	1	1	8	1	24	45	35	4
VP9 to HEVC	8k	1	1	8	1	24	44	37	4
AVC to YUV	8k	1	0	10	1	54	0	54	7
HEVC to YUV	8k	1	0	10	1	51	0	51	6
VP9 to YUV	8k	1	0	10	1	24	0	33	4
YUV to AVC	8k	1	0	10	1	0	68	40	35
YUV to HEVC	8k	1	0	10	1	0	47	40	34
AVC to YUV	4k	1	0	8	1	57	0	319	8
HEVC to YUV	4k	1	0	8	1	50	0	330	7
VP9 to YUV	4k	1	0	8	1	23	0	154	3
AVC to YUV	4k	16	0	8	0	98	0	487	0
HEVC to YUV	4k	16	0	8	0	98	0	511	0
VP9 to YUV	4k	16	0	8	0	99	0	492	0
YUV to AVC	4k	1	0	8	1	0	91	284	26
YUV to HEVC	4k	1	0	8	1	0	88	304	27
YUV to AV1	4k	1	0	8	1	0	91	272	23
YUV to AVC	4k	4	0	8	0	0	96	311	11
YUV to HEVC	4k	4	0	8	0	0	96	334	11
YUV to AV1	4k	4	0	8	0	0	96	289	11
YUV to AVC	4k	8	0	8	0	0	99	327	7
YUV to HEVC	4k	8	0	8	0	0	100	347	8
YUV to AV1	4k	8	0	8	0	0	100	301	6
AVC to AVC	4k	1	1	8	0	67	97	225	8
AVC to HEVC	4k	1	1	8	0	73	96	278	8
AVC to AV1	4k	1	1	8	0	65	96	263	8
HEVC to AVC	4k	1	1	8	0	58	97	225	6
HEVC to HEVC	4k	1	1	8	0	64	97	287	7
HEVC to AV1	4k	1	1	8	0	56	97	266	7
VP9 to AVC	4k	1	1	8	0	24	46	149	5

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	0	24	43	149	4
VP9 to AV1	4k	1	1	8	0	24	50	149	5
AVC to AVC	4k	4	1	8	0	61	97	243	5
AVC to HEVC	4k	4	1	8	0	67	96	303	5
AVC to AV1	4k	4	1	8	0	56	95	276	6
HEVC to AVC	4k	4	1	8	0	57	97	240	5
HEVC to HEVC	4k	4	1	8	0	60	96	294	5
HEVC to AV1	4k	4	1	8	0	48	95	274	5
VP9 to AVC	4k	4	1	8	0	64	97	245	5
VP9 to HEVC	4k	4	1	8	0	66	96	303	5
VP9 to AV1	4k	4	1	8	0	55	96	277	5
AVC to AVC	4k	8	1	8	0	69	99	217	3
AVC to HEVC	4k	8	1	8	0	75	99	280	4
AVC to AV1	4k	8	1	8	0	67	100	274	4
HEVC to AVC	4k	8	1	8	0	63	99	218	3
HEVC to HEVC	4k	8	1	8	0	68	100	282	4
HEVC to AV1	4k	8	1	8	0	61	100	275	3
VP9 to AVC	4k	8	1	8	0	69	100	235	3
VP9 to HEVC	4k	8	1	8	0	72	100	304	3
VP9 to AV1	4k	8	1	8	0	60	99	286	3
AVC to YUV	4k	1	0	10	1	45	0	215	7
HEVC to YUV	4k	1	0	10	1	42	0	207	6
VP9 to YUV	4k	1	0	10	1	24	0	158	3
AVC to YUV	4k	16	0	10	0	98	0	286	0
HEVC to YUV	4k	16	0	10	0	99	0	285	0
VP9 to YUV	4k	16	0	10	0	100	0	506	0
YUV to AVC	4k	1	0	10	1	0	51	164	26
YUV to HEVC	4k	1	0	10	1	0	47	163	27
YUV to AV1	4k	1	0	10	1	0	52	156	25
YUV to AVC	4k	4	0	10	0	0	67	208	25
YUV to HEVC	4k	4	0	10	0	0	59	211	26
YUV to AV1	4k	4	0	10	0	0	63	186	28
AVC to YUV	1080p	1	0	8	1	40	0	836	14
HEVC to YUV	1080p	1	0	8	1	45	0	882	11
VP9 to YUV	1080p	1	0	8	1	22	0	557	5
AVC to YUV	1080p	40	0	8	0	81	0	1647	0
HEVC to YUV	1080p	40	0	8	0	86	0	1767	0
VP9 to YUV	1080p	40	0	8	0	76	0	1684	0
YUV to AVC	1080p	1	0	8	1	0	57	732	16

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	54	744	15
YUV to AV1	1080p	1	0	8	1	0	50	600	16
YUV to AVC	1080p	32	0	8	0	0	99	1326	2
YUV to HEVC	1080p	32	0	8	0	0	99	1402	3
YUV to AV1	1080p	32	0	8	0	0	100	1211	2
AVC to AVC	1080p	1	1	8	0	70	85	1011	18
AVC to HEVC	1080p	1	1	8	0	68	82	1064	19
AVC to AV1	1080p	1	1	8	0	55	82	934	18
HEVC to AVC	1080p	1	1	8	0	61	86	994	13
HEVC to HEVC	1080p	1	1	8	0	64	86	1091	14
HEVC to AV1	1080p	1	1	8	0	52	84	937	14
VP9 to AVC	1080p	1	1	8	0	22	41	535	7
VP9 to HEVC	1080p	1	1	8	0	22	39	541	11
VP9 to AV1	1080p	1	1	8	0	22	46	543	7
AVC to AVC	1080p	32	1	8	0	74	99	978	1
AVC to HEVC	1080p	32	1	8	0	83	99	1105	1
AVC to AV1	1080p	32	1	8	0	77	99	1066	1
HEVC to AVC	1080p	32	1	8	0	70	99	1021	1
HEVC to HEVC	1080p	32	1	8	0	77	99	1157	1
HEVC to AV1	1080p	32	1	8	0	69	99	1091	1
VP9 to AVC	1080p	32	1	8	0	70	100	1115	1
VP9 to HEVC	1080p	32	1	8	0	74	100	1262	0
VP9 to AV1	1080p	32	1	8	0	71	100	1158	1
AVC to YUV	1080p	1	0	10	1	30	0	656	7
HEVC to YUV	1080p	1	0	10	1	29	0	657	7
VP9 to YUV	1080p	1	0	10	1	22	0	456	5
AVC to YUV	1080p	40	0	10	0	66	0	1078	0
HEVC to YUV	1080p	40	0	10	0	69	0	1071	0
VP9 to YUV	1080p	40	0	10	0	71	0	1034	0
YUV to AVC	1080p	1	0	10	1	0	37	478	20
YUV to HEVC	1080p	1	0	10	1	0	34	473	19
YUV to AV1	1080p	1	0	10	1	0	36	425	20
YUV to AVC	1080p	32	0	10	0	0	64	852	6
YUV to HEVC	1080p	32	0	10	0	0	60	847	6
YUV to AV1	1080p	32	0	10	0	0	65	798	8
AVC to YUV	720p	1	0	8	1	42	0	1179	13
HEVC to YUV	720p	1	0	8	1	39	0	1250	10
VP9 to YUV	720p	1	0	8	1	31	0	1022	7
AVC to YUV	720p	100	0	8	0	100	0	2771	0

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	93	0	2981	0
VP9 to YUV	720p	100	0	8	0	93	0	2671	0
YUV to AVC	720p	1	0	8	1	0	35	1023	14
YUV to HEVC	720p	1	0	8	1	0	33	986	9
YUV to AV1	720p	1	0	8	1	0	32	776	11
YUV to AVC	720p	64	0	8	0	0	99	2703	2
YUV to HEVC	720p	64	0	8	0	0	98	2767	2
YUV to AV1	720p	64	0	8	0	0	99	2234	2
AVC to AVC	720p	1	1	8	0	52	50	1374	18
AVC to HEVC	720p	1	1	8	0	52	49	1361	18
AVC to AV1	720p	1	1	8	0	43	49	1086	16
HEVC to AVC	720p	1	1	8	0	46	50	1390	14
HEVC to HEVC	720p	1	1	8	0	46	50	1399	14
HEVC to AV1	720p	1	1	8	0	39	48	1091	13
VP9 to AVC	720p	1	1	8	0	32	35	1009	9
VP9 to HEVC	720p	1	1	8	0	31	34	995	8
VP9 to AV1	720p	1	1	8	0	31	44	996	10
AVC to AVC	720p	64	1	8	0	95	99	2187	0
AVC to HEVC	720p	64	1	8	0	100	100	2301	0
AVC to AV1	720p	64	1	8	0	95	100	1915	0
HEVC to AVC	720p	64	1	8	0	86	98	2202	0
HEVC to HEVC	720p	64	1	8	0	95	100	2308	0
HEVC to AV1	720p	64	1	8	0	89	100	1918	0
VP9 to AVC	720p	64	1	8	0	100	100	2464	0
VP9 to HEVC	720p	64	1	8	0	100	100	2520	0
VP9 to AV1	720p	64	1	8	0	99	100	2060	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	57.83	59.46	57.29	0.05
YUV to HEVC	8k	1	55.47	60.55	53.95	0.97
YUV to AVC	4k	1	15.33	17.32	14.96	0.06
YUV to HEVC	4k	1	16.54	18.94	15.55	0.1
YUV to AV1	4k	1	21.72	25.39	16.32	0.64
YUV to AVC	4k	4	16.7	21.37	14.97	2.23
YUV to HEVC	4k	4	17.96	22.55	15.68	1.82
YUV to AV1	4k	4	22.98	27.38	16.34	0.96
YUV to AVC	4k	8	20.19	35.34	15.02	21.63
YUV to HEVC	4k	8	22.47	33.4	15.66	22.88
YUV to AV1	4k	8	37.25	46.69	22.07	10.14
YUV to AVC	1080p	1	4.59	5.57	4.41	0.02
YUV to HEVC	1080p	1	4.95	5.55	4.7	0.03
YUV to AV1	1080p	1	6.73	7.6	4.98	0.05
YUV to AVC	1080p	32	5.91	10.73	4.59	0.47
YUV to HEVC	1080p	32	6.49	11.66	4.75	0.98
YUV to AV1	1080p	32	41.02	48.45	33.18	1.24
YUV to AVC	720p	1	2.81	3.42	2.49	0.01
YUV to HEVC	720p	1	2.85	3.18	2.66	0.01
YUV to AV1	720p	1	3.79	4.19	3.01	0.02
YUV to AVC	720p	64	5.03	9.06	2.77	0.82
YUV to HEVC	720p	64	5.3	11.11	2.79	1.17
YUV to AV1	720p	64	39.78	46.84	31.01	2.81

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	91	0	142	8
HEVC to RGBA	8k	1	89	0	154	8
VP9 to RGBA	8k	1	22	0	40	3
AVC to RGBA	4k	1	21	0	145	9
AVC to RGBA	4k	16	93	3	583	2
HEVC to RGBA	4k	1	21	0	174	11
HEVC to RGBA	4k	16	92	4	668	3
VP9 to RGBA	4k	1	22	0	166	5
VP9 to RGBA	4k	16	94	4	679	1
AVC to RGBA	1080p	40	95	14	1900	1
HEVC to RGBA	1080p	40	93	14	2009	1
VP9 to RGBA	1080p	40	93	17	2444	0
AVC to RGBA	720p	100	96	18	2601	0
HEVC to RGBA	720p	100	91	21	2893	0
VP9 to RGBA	720p	64	95	19	2754	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	93	2	435	4
AVC to HEVC	8	32	92	2	465	4
AVC to AV1	8	24	95	2	400	3
HEVC to AVC	8	34	93	2	440	5
HEVC to HEVC	8	32	92	2	472	5
HEVC to AV1	8	25	95	2	400	5
VP9 to AVC	8	40	93	2	432	4
VP9 to HEVC	8	38	93	2	466	4
VP9 to AV1	8	30	95	2	400	4

6. T1A – RGBA Encoding

6.1 Encoding

6.1.1 Description

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

RGBA frame is uploaded and encoded by hardware encoder.

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

6.1.2 Command line

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

<enc> is the encoder codec. eg h264_ni_quadra_enc, h265_ni_quadra_enc, av1_ni_quadra_enc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

6.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	53	162	62
RGBA to HEVC	4k	1	1	46	161	63
RGBA to AV1	4k	1	1	56	166	70
RGBA to AVC	4k	4	0	67	166	37
RGBA to HEVC	4k	4	0	60	169	37
RGBA to AV1	4k	4	0	61	172	41
RGBA to AVC	4k	8	0	62	176	30
RGBA to HEVC	4k	8	0	55	176	28
RGBA to AV1	4k	8	0	61	177	30
RGBA to AVC	1080p	1	1	33	437	46
RGBA to HEVC	1080p	1	1	31	434	45
RGBA to AV1	1080p	1	1	35	417	42
RGBA to AVC	1080p	16	0	60	653	14
RGBA to HEVC	1080p	16	0	53	656	14
RGBA to AV1	1080p	16	0	56	642	15
RGBA to AVC	1080p	32	0	57	669	9
RGBA to HEVC	1080p	32	0	53	672	8
RGBA to AV1	1080p	32	0	57	652	9
RGBA to AVC	720p	1	1	29	722	54
RGBA to HEVC	720p	1	1	28	787	66
RGBA to AV1	720p	1	1	34	728	61
RGBA to AVC	720p	16	0	48	1244	41
RGBA to HEVC	720p	16	0	47	1234	36
RGBA to AV1	720p	16	0	54	1159	30
RGBA to AVC	720p	32	0	46	1173	31
RGBA to HEVC	720p	32	0	44	1179	31
RGBA to AV1	720p	32	0	55	1175	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	96	302	21
YUV to HEVC	4k	4	0	0	1	96	332	14
YUV to AV1	4k	4	0	0	1	96	288	12
YUV to AVC	4k	4	0	0	2	93	296	27
YUV to HEVC	4k	4	0	0	2	98	170	7
YUV to AV1	4k	4	0	0	2	98	140	7
YUV to AVC	4k	4	0	0	3	96	299	15
YUV to HEVC	4k	4	0	0	3	98	100	6
YUV to AV1	4k	4	0	0	3	97	76	4
YUV to AVC	4k	4	0	1	1	97	188	8
YUV to HEVC	4k	4	0	1	1	97	237	10
YUV to AVC	4k	4	0	1	2	98	188	8
YUV to HEVC	4k	4	0	1	2	99	108	5
YUV to AVC	4k	4	0	1	3	98	188	7
YUV to HEVC	4k	4	0	1	3	97	68	5
YUV to AVC	4k	4	4	0	1	100	196	9
YUV to HEVC	4k	4	4	0	1	100	228	16
YUV to AV1	4k	4	4	0	1	99	200	9
YUV to AVC	4k	4	4	0	2	100	196	10
YUV to HEVC	4k	4	4	0	2	100	136	6
YUV to AV1	4k	4	4	0	2	99	116	6
YUV to AVC	4k	4	4	0	3	99	197	9
YUV to HEVC	4k	4	4	0	3	99	88	4
YUV to AV1	4k	4	4	0	3	100	72	3
YUV to AVC	4k	4	4	1	1	99	140	6
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	99	96	5
YUV to AVC	4k	4	4	1	3	99	140	6
YUV to HEVC	4k	4	4	1	3	98	64	3
YUV to AVC	4k	4	16	0	1	99	196	9
YUV to HEVC	4k	4	16	0	1	100	228	12
YUV to AV1	4k	4	16	0	1	99	200	11
YUV to AVC	4k	4	16	0	2	99	196	9
YUV to HEVC	4k	4	16	0	2	100	136	7
YUV to AV1	4k	4	16	0	2	100	116	5
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	6
YUV to HEVC	4k	4	16	1	1	100	180	7
YUV to AVC	4k	4	16	1	2	100	140	7
YUV to HEVC	4k	4	16	1	2	99	95	5
YUV to AVC	4k	4	16	1	3	100	140	7
YUV to HEVC	4k	4	16	1	3	100	64	3
YUV to AVC	4k	4	40	0	1	100	196	10
YUV to HEVC	4k	4	40	0	1	99	224	9
YUV to AV1	4k	4	40	0	1	100	196	8
YUV to AVC	4k	4	40	0	2	99	196	8
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	99	196	9
YUV to HEVC	4k	4	40	0	3	99	88	5
YUV to AV1	4k	4	40	0	3	100	72	3
YUV to AVC	4k	4	40	1	1	100	136	6
YUV to HEVC	4k	4	40	1	1	100	176	11
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	99	136	6
YUV to HEVC	4k	4	40	1	3	99	64	3
YUV to AVC	1080p	20	0	0	1	99	1280	5
YUV to HEVC	1080p	20	0	0	1	99	1360	5
YUV to AV1	1080p	20	0	0	1	100	1179	4
YUV to AVC	1080p	20	0	0	2	100	1280	4
YUV to HEVC	1080p	20	0	0	2	100	686	2
YUV to AV1	1080p	20	0	0	2	100	560	2
YUV to AVC	1080p	20	0	0	3	99	1280	4
YUV to HEVC	1080p	20	0	0	3	100	403	2
YUV to AV1	1080p	20	0	0	3	100	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	2
YUV to HEVC	1080p	20	0	1	2	100	440	2
YUV to AVC	1080p	20	0	1	3	99	760	2
YUV to HEVC	1080p	20	0	1	3	99	280	1
YUV to AVC	1080p	20	4	0	1	100	700	2
YUV to HEVC	1080p	20	4	0	1	99	820	3

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	99	720	2
YUV to AVC	1080p	20	4	0	2	99	700	2
YUV to HEVC	1080p	20	4	0	2	99	517	2
YUV to AV1	1080p	20	4	0	2	99	420	2
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	267	1
YUV to AVC	1080p	20	4	1	1	99	500	2
YUV to HEVC	1080p	20	4	1	1	99	659	2
YUV to AVC	1080p	20	4	1	2	99	500	2
YUV to HEVC	1080p	20	4	1	2	100	360	1
YUV to AVC	1080p	20	4	1	3	100	501	2
YUV to HEVC	1080p	20	4	1	3	100	240	1
YUV to AVC	1080p	20	16	0	1	99	700	2
YUV to HEVC	1080p	20	16	0	1	99	820	3
YUV to AV1	1080p	20	16	0	1	100	719	2
YUV to AVC	1080p	20	16	0	2	99	700	3
YUV to HEVC	1080p	20	16	0	2	100	503	2
YUV to AV1	1080p	20	16	0	2	99	420	2
YUV to AVC	1080p	20	16	0	3	99	700	2
YUV to HEVC	1080p	20	16	0	3	100	340	1
YUV to AV1	1080p	20	16	0	3	99	264	1
YUV to AVC	1080p	20	16	1	1	100	500	2
YUV to HEVC	1080p	20	16	1	1	99	646	2
YUV to AVC	1080p	20	16	1	2	99	500	2
YUV to HEVC	1080p	20	16	1	2	100	360	1
YUV to AVC	1080p	20	16	1	3	99	500	2
YUV to HEVC	1080p	20	16	1	3	99	240	1
YUV to AVC	1080p	20	40	0	1	99	688	2
YUV to HEVC	1080p	20	40	0	1	99	800	3
YUV to AV1	1080p	20	40	0	1	99	703	3
YUV to AVC	1080p	20	40	0	2	99	688	2
YUV to HEVC	1080p	20	40	0	2	99	500	2
YUV to AV1	1080p	20	40	0	2	100	420	1
YUV to AVC	1080p	20	40	0	3	99	690	2
YUV to HEVC	1080p	20	40	0	3	99	340	1
YUV to AV1	1080p	20	40	0	3	100	260	1
YUV to AVC	1080p	20	40	1	1	99	500	2
YUV to HEVC	1080p	20	40	1	1	100	640	2

TYPE	RES	JOB	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	90	2284	2
YUV to HEVC	720p	40	0	0	1	88	2291	2
YUV to AV1	720p	40	0	0	1	95	2018	2
YUV to AVC	720p	40	0	0	2	92	2287	3
YUV to HEVC	720p	40	0	0	2	99	1520	1
YUV to AV1	720p	40	0	0	2	99	1202	1
YUV to AVC	720p	40	0	0	3	91	2284	3
YUV to HEVC	720p	40	0	0	3	99	920	1
YUV to AV1	720p	40	0	0	3	100	650	1
YUV to AVC	720p	40	0	1	1	99	1685	1
YUV to HEVC	720p	40	0	1	1	99	2080	2
YUV to AVC	720p	40	0	1	2	99	1692	1
YUV to HEVC	720p	40	0	1	2	99	960	1
YUV to AVC	720p	40	0	1	3	99	1692	1
YUV to HEVC	720p	40	0	1	3	99	611	1
YUV to AVC	720p	40	4	0	1	100	1440	1
YUV to HEVC	720p	40	4	0	1	100	1360	1
YUV to AV1	720p	40	4	0	1	100	1008	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	100	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	1079	1
YUV to HEVC	720p	40	4	1	1	100	1322	1
YUV to AVC	720p	40	4	1	2	99	1078	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	1400	1
YUV to HEVC	720p	40	16	0	1	100	1357	1
YUV to AV1	720p	40	16	0	1	100	1060	1
YUV to AVC	720p	40	16	0	2	100	1400	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	100	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	1048	1
YUV to HEVC	720p	40	16	1	1	100	1321	1
YUV to AVC	720p	40	16	1	2	99	1051	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	98	1360	1
YUV to HEVC	720p	40	40	0	1	100	1320	1
YUV to AV1	720p	40	40	0	1	100	1040	1
YUV to AVC	720p	40	40	0	2	100	1360	1
YUV to HEVC	720p	40	40	0	2	99	1040	1
YUV to AV1	720p	40	40	0	2	99	846	1
YUV to AVC	720p	40	40	0	3	100	1360	1
YUV to HEVC	720p	40	40	0	3	99	720	1
YUV to AV1	720p	40	40	0	3	99	560	1
YUV to AVC	720p	40	40	1	1	99	1040	1
YUV to HEVC	720p	40	40	1	1	100	1316	1
YUV to AVC	720p	40	40	1	2	99	1040	1
YUV to HEVC	720p	40	40	1	2	99	760	1
YUV to AVC	720p	40	40	1	3	99	1040	1
YUV to HEVC	720p	40	40	1	3	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	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	19	99	702	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	701	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	99	340	1
YUV to AV1	1080p	20	0	0	3	19	100	280	1
YUV to AVC	1080p	20	0	1	1	19	99	501	2
YUV to HEVC	1080p	20	0	1	1	19	99	660	2
YUV to AVC	1080p	20	0	1	2	19	99	501	2
YUV to HEVC	1080p	20	0	1	2	19	99	360	2
YUV to AVC	1080p	20	0	1	3	19	100	503	2
YUV to HEVC	1080p	20	0	1	3	19	100	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	266	1
YUV to AVC	1080p	20	4	1	1	19	99	500	2
YUV to HEVC	1080p	20	4	1	1	19	100	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	2
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	3
YUV to HEVC	1080p	20	16	0	1	19	99	820	3
YUV to AV1	1080p	20	16	0	1	19	99	719	3
YUV to AVC	1080p	20	16	0	2	19	99	700	2
YUV to HEVC	1080p	20	16	0	2	19	100	503	2
YUV to AV1	1080p	20	16	0	2	19	100	420	1
YUV to AVC	1080p	20	16	0	3	19	99	700	3
YUV to HEVC	1080p	20	16	0	3	19	100	340	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	3	19	99	264	1
YUV to AVC	1080p	20	16	1	1	19	99	500	2
YUV to HEVC	1080p	20	16	1	1	19	99	646	2
YUV to AVC	1080p	20	16	1	2	19	100	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	691	2
YUV to HEVC	1080p	20	40	0	1	19	99	800	3
YUV to AV1	1080p	20	40	0	1	19	100	701	2
YUV to AVC	1080p	20	40	0	2	19	100	691	3
YUV to HEVC	1080p	20	40	0	2	19	100	500	2
YUV to AV1	1080p	20	40	0	2	19	100	420	2
YUV to AVC	1080p	20	40	0	3	19	99	690	3
YUV to HEVC	1080p	20	40	0	3	19	100	340	1
YUV to AV1	1080p	20	40	0	3	19	99	260	1
YUV to AVC	1080p	20	40	1	1	19	100	500	2
YUV to HEVC	1080p	20	40	1	1	19	99	640	2
YUV to AVC	1080p	20	40	1	2	19	100	500	2
YUV to HEVC	1080p	20	40	1	2	19	99	360	1
YUV to AVC	1080p	20	40	1	3	19	99	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	100	700	2
YUV to HEVC	1080p	20	0	0	2	23	100	520	2
YUV to AV1	1080p	20	0	0	2	23	99	439	1
YUV to AVC	1080p	20	0	0	3	23	99	701	2
YUV to HEVC	1080p	20	0	0	3	23	100	340	1
YUV to AV1	1080p	20	0	0	3	23	100	280	1
YUV to AVC	1080p	20	0	1	1	23	99	501	2
YUV to HEVC	1080p	20	0	1	1	23	99	660	2
YUV to AVC	1080p	20	0	1	2	23	99	505	2
YUV to HEVC	1080p	20	0	1	2	23	100	360	2
YUV to AVC	1080p	20	0	1	3	23	99	503	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	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	23	99	720	2
YUV to AVC	1080p	20	4	0	2	23	99	701	2
YUV to HEVC	1080p	20	4	0	2	23	100	519	2
YUV to AV1	1080p	20	4	0	2	23	99	420	2
YUV to AVC	1080p	20	4	0	3	23	99	700	2
YUV to HEVC	1080p	20	4	0	3	23	99	340	1
YUV to AV1	1080p	20	4	0	3	23	99	268	1
YUV to AVC	1080p	20	4	1	1	23	99	500	2
YUV to HEVC	1080p	20	4	1	1	23	100	660	2
YUV to AVC	1080p	20	4	1	2	23	99	501	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	719	2
YUV to AVC	1080p	20	16	0	2	23	99	700	2
YUV to HEVC	1080p	20	16	0	2	23	99	504	2
YUV to AV1	1080p	20	16	0	2	23	99	420	2
YUV to AVC	1080p	20	16	0	3	23	99	700	2
YUV to HEVC	1080p	20	16	0	3	23	99	340	1
YUV to AV1	1080p	20	16	0	3	23	100	262	1
YUV to AVC	1080p	20	16	1	1	23	99	500	2
YUV to HEVC	1080p	20	16	1	1	23	99	648	2
YUV to AVC	1080p	20	16	1	2	23	99	500	2
YUV to HEVC	1080p	20	16	1	2	23	100	360	1
YUV to AVC	1080p	20	16	1	3	23	100	500	2
YUV to HEVC	1080p	20	16	1	3	23	99	240	1
YUV to AVC	1080p	20	40	0	1	23	99	695	2
YUV to HEVC	1080p	20	40	0	1	23	99	800	3
YUV to AV1	1080p	20	40	0	1	23	99	701	2
YUV to AVC	1080p	20	40	0	2	23	99	689	2
YUV to HEVC	1080p	20	40	0	2	23	99	500	2
YUV to AV1	1080p	20	40	0	2	23	99	420	1
YUV to AVC	1080p	20	40	0	3	23	99	690	2
YUV to HEVC	1080p	20	40	0	3	23	100	340	1
YUV to AV1	1080p	20	40	0	3	23	99	260	1
YUV to AVC	1080p	20	40	1	1	23	99	500	2
YUV to HEVC	1080p	20	40	1	1	23	99	640	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	2	23	100	500	2
YUV to HEVC	1080p	20	40	1	2	23	100	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	701	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	701	2
YUV to HEVC	1080p	20	0	0	2	27	100	520	2
YUV to AV1	1080p	20	0	0	2	27	99	440	2
YUV to AVC	1080p	20	0	0	3	27	100	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	501	2
YUV to HEVC	1080p	20	0	1	1	27	99	660	2
YUV to AVC	1080p	20	0	1	2	27	99	503	2
YUV to HEVC	1080p	20	0	1	2	27	100	360	1
YUV to AVC	1080p	20	0	1	3	27	99	501	2
YUV to HEVC	1080p	20	0	1	3	27	100	240	1
YUV to AVC	1080p	20	4	0	1	27	99	702	2
YUV to HEVC	1080p	20	4	0	1	27	99	820	3
YUV to AV1	1080p	20	4	0	1	27	99	720	2
YUV to AVC	1080p	20	4	0	2	27	99	700	3
YUV to HEVC	1080p	20	4	0	2	27	100	520	2
YUV to AV1	1080p	20	4	0	2	27	99	420	2
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	268	1
YUV to AVC	1080p	20	4	1	1	27	99	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	100	360	1
YUV to AVC	1080p	20	4	1	3	27	100	500	2
YUV to HEVC	1080p	20	4	1	3	27	99	241	1
YUV to AVC	1080p	20	16	0	1	27	99	700	2
YUV to HEVC	1080p	20	16	0	1	27	99	820	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	100	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	99	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	99	645	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	99	500	2
YUV to HEVC	1080p	20	16	1	3	27	100	240	1
YUV to AVC	1080p	20	40	0	1	27	99	691	2
YUV to HEVC	1080p	20	40	0	1	27	99	802	3
YUV to AV1	1080p	20	40	0	1	27	99	700	2
YUV to AVC	1080p	20	40	0	2	27	100	692	2
YUV to HEVC	1080p	20	40	0	2	27	100	500	2
YUV to AV1	1080p	20	40	0	2	27	99	420	2
YUV to AVC	1080p	20	40	0	3	27	99	689	2
YUV to HEVC	1080p	20	40	0	3	27	100	339	1
YUV to AV1	1080p	20	40	0	3	27	99	260	1
YUV to AVC	1080p	20	40	1	1	27	99	500	2
YUV to HEVC	1080p	20	40	1	1	27	99	640	2
YUV to AVC	1080p	20	40	1	2	27	99	500	2
YUV to HEVC	1080p	20	40	1	2	27	100	360	1
YUV to AVC	1080p	20	40	1	3	27	99	500	2
YUV to HEVC	1080p	20	40	1	3	27	99	240	1

9. T1A – Inplace Overlay

9.1 Transcoding

9.1.1 Description

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

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

9.1.2 Command line

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

<dec> is the decoder codec. ie h264_ni_quadra_dec, h265_ni_quadra_dec, vp9_ni_quadra_dec

<enc> is the encoder codec. ie h264_ni_quadra_enc, h265_ni_quadra_enc, av1_ni_quadra_enc

Input Video: 1080p

Input Image: 128x128

9.2 Inplace Overlay Performance Results

TYPE	JOB5	FPS	CPU	DEC_LOAD	ENC_LOAD	SCALER_LOAD
AVC to AVC	1	290	12	14	21	5
AVC to HEVC	1	308	12	14	21	5
AVC to AV1	1	263	12	13	21	4
HEVC to AVC	1	289	13	12	20	5
HEVC to HEVC	1	308	16	13	20	5
HEVC to AV1	1	263	15	12	20	4
VP9 to AVC	1	289	11	16	20	5
VP9 to HEVC	1	306	12	17	20	5
VP9 to AV1	1	263	12	16	20	4
AVC to AVC	16	1046	3	71	93	24
AVC to HEVC	16	1157	3	78	93	27
AVC to AV1	16	1109	3	72	94	26
HEVC to AVC	16	1088	4	72	94	25
HEVC to HEVC	16	1216	4	78	94	28
HEVC to AV1	16	1138	4	69	95	26
VP9 to AVC	16	1040	3	81	92	23
VP9 to HEVC	16	1168	3	89	93	26
VP9 to AV1	16	1120	3	83	93	25
AVC to AVC	32	960	1	73	93	23
AVC to HEVC	32	1087	1	80	92	26
AVC to AV1	32	1056	1	76	93	26
HEVC to AVC	32	997	1	72	93	24
HEVC to HEVC	32	1139	2	79	92	28
HEVC to AV1	32	1089	2	72	93	26
VP9 to AVC	32	992	1	81	94	23
VP9 to HEVC	32	1120	1	89	93	26
VP9 to AV1	32	1088	1	84	94	25

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	29	1.3	0.25	0.5	2.07	0.01
P2H	720p	1	1001	29	1.2	0.25	0.5	2.07	0.02
P2A	720p	180	180180	29.7	0.8	46	56	2.3	0.11
P2H	720p	180	180180	29.7	0.8	45	56	2.26	0.09
P2A	1080p	1	1001	29	1.3	0.5	0.75	3.79	0.01
P2H	1080p	1	1001	30	1.3	0.5	0.75	3.58	0.02
P2A	1080p	80	80080	29.8	0.89	46	50	3.92	0.27
P2H	1080p	80	80080	29.4	0.9	43	50	3.78	0.27

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	Session Number	Loops	Channel Order	File Format	File Type	Input Size	FPS per session
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	359
resnet50	8	10000	rgb	nchw	image	224x224x3	228
mobilenetv2	8	10000	rgb	nchw	image	224x224x3	1110
deeplabv3	8	10000	rgb	nchw	image	257x257x3	160
yolov4	8	10000	bgr	nchw	image	416x416x3	256
fsrcnn	8	2000	bgr	nchw	image	360x640x1	31
BiSeNetv1	8	10000	rgb	nchw	image	512x512x3	77
HrNet	8	10000	rgb	nchw	image	256x192x3	74
usm_1656x1920	8	10000	rgb	nchw	image	1920x1656x1	252
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	2354.5
mobilenetv2_nchw_keras_96x160	8	10000	rgb	nchw	image	96x160x3	2343.88
mobilenetv2_nchw_keras_96x160	16	10000	rgb	nchw	image	96x160x3	2345.74

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:qppoffset=-0.3' -enc 0 -
c:v h264_ni_quadra_enc -xcoder-params 'roiEnable=1:RcEnable=1' -f null
-
```

11.2.6 Command line PRE

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

11.2.7 AI Encoding with 2D Engine Performance Results

Filter	Model	Resolution	Session Number	Average FPS per session
ROI	network_binary_yolov4_head	1920x1080	1	77
ROI	network_binary_yolov4_head	1920x1080	8	23
ROI	network_binary_yolov4_head	1920x1080	32	5
BG	segm32_tflite_nchw_bgr	1920x1080	1	68
BG	segm32_tflite_nchw_bgr	1920x1080	8	41
BG	segm32_tflite_nchw_bgr	1920x1080	32	14
PRE	hw_lanczos_network_binary	1920x1080	8	66
PRE	hw_lanczos_network_binary	1920x1080	16	34
PRE	hw_bicubic_network_binary	1920x1080	8	67
PRE	hw_bicubic_network_binary	1920x1080	16	34

12. T1A – GStreamer XStack Throughput

12.1 Transcoding

12.1.1 Description

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

Decoded YUV frames are all kept on device and are sent through the `ni_quadra_xstack` filter to produce a single YUV output.

The YUV frame is encoded with hardware encoder.

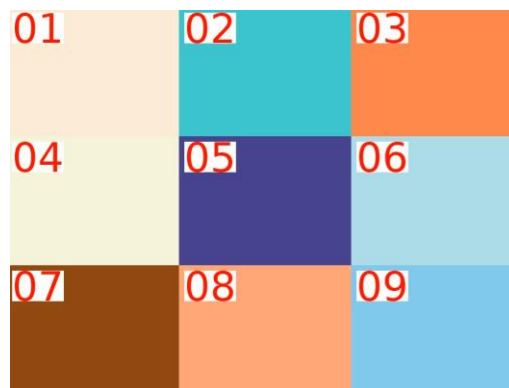
The encoded bitstream is read out through PCIe and written into an output file.

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

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

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

This test is HEVC to AVC only.



Example output in a 3x3 layout with 9 inputs

12.1.2 Command line

See Appendix A: GStreamer XStack Command

12.2 GStreamer XStack Performance Results

Input Res	Grid	Output Res	Cell Size	FPS	CPU	Dec Load	Enc Load	Scaler Load
1920x1080	3x3	1920x1080	640x360	148.89	33	64	13	15
1920x1080	4x4	1920x1080	480x270	102.48	44	79	9	14
1920x1080	4x8	1920x1080	480x135	56.83	52	87	5	7
1920x1080	3x3	3840x2160	1280x720	70.49	16	31	21	18
1920x1080	4x4	3840x2160	960x540	55.6	26	45	18	18
1920x1080	4x8	3840x2160	960x270	42.27	46	70	15	17
1920x1080	3x3	7680x4320	2560x1440	19.88	10	8	22	15
1920x1080	4x4	7680x4320	1920x1080	19.18	13	15	22	16
1920x1080	4x8	7680x4320	1920x540	19.39	21	32	22	19
1920x1080	6x6	1920x1080	320x180	50.14	62	87	5	9
1920x1080	7x7	1920x1080	274x154 276x154 274x156 276x156*	37.16	61	87	4	8

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

13. T1A – GStreamer Ladder Generation

13.1 Transcoding

13.1.1 Description

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

Decoded YUV is split to multiple pads.

The YUV frames are encoded with hardware encoder.

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

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

This test is AVC to HEVC only

13.1.2 Command line

See Appendix C: GStreamer Ladder Command

13.2 GStreamer Ladder Performance Results

Jobs	Outputs	FPS	CPU	Dec Load	Enc Load
1	64	18.99	61	1	89

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	93	0	99	13
HEVC to YUV	8k	1	0	8	1	88	0	98	18
VP9 to YUV	8k	1	0	8	1	25	0	30	3
YUV to AVC	8k	1	0	8	1	0	95	57	81
YUV to HEVC	8k	1	0	8	1	0	98	71	105
AVC to AVC	8k	1	1	8	1	73	99	47	4
AVC to HEVC	8k	1	1	8	1	81	99	63	9
HEVC to AVC	8k	1	1	8	1	64	100	47	3
HEVC to HEVC	8k	1	1	8	1	71	100	63	9
VP9 to AVC	8k	1	1	8	1	25	42	29	2
VP9 to HEVC	8k	1	1	8	1	24	39	30	1
AVC to YUV	8k	1	0	10	1	61	0	54	7
HEVC to YUV	8k	1	0	10	1	90	0	56	7
VP9 to YUV	8k	1	0	10	1	25	0	28	4
YUV to AVC	8k	1	0	10	1	0	89	39	112
YUV to HEVC	8k	1	0	10	1	0	55	40	122
AVC to YUV	4k	1	0	8	1	59	0	269	16
HEVC to YUV	4k	1	0	8	1	52	0	275	21
VP9 to YUV	4k	1	0	8	1	24	0	108	8
AVC to YUV	4k	16	0	8	0	99	0	417	1
HEVC to YUV	4k	16	0	8	0	98	0	449	1
VP9 to YUV	4k	16	0	8	0	100	0	401	0
YUV to AVC	4k	1	0	8	1	0	94	253	44
YUV to HEVC	4k	1	0	8	1	0	95	278	55
YUV to AV1	4k	1	0	8	1	0	94	239	43
YUV to AVC	4k	4	0	8	0	0	97	264	13
YUV to HEVC	4k	4	0	8	0	0	96	284	13
YUV to AV1	4k	4	0	8	0	0	95	246	12
YUV to AVC	4k	8	0	8	0	0	99	280	7
YUV to HEVC	4k	8	0	8	0	0	100	296	9
YUV to AV1	4k	8	0	8	0	0	100	256	7
AVC to AVC	4k	1	1	8	1	70	94	198	16
AVC to HEVC	4k	1	1	8	1	70	93	243	9
AVC to AV1	4k	1	1	8	1	61	94	232	13
HEVC to AVC	4k	1	1	8	1	56	97	199	13
HEVC to HEVC	4k	1	1	8	1	56	92	251	15
HEVC to AV1	4k	1	1	8	1	47	93	229	14
VP9 to AVC	4k	1	1	8	1	24	38	107	1
VP9 to HEVC	4k	1	1	8	1	24	36	108	6

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	40	109	7
AVC to AVC	4k	4	1	8	0	61	97	220	1
AVC to HEVC	4k	4	1	8	0	69	97	252	2
AVC to AV1	4k	4	1	8	0	56	95	244	9
HEVC to AVC	4k	4	1	8	0	55	96	216	2
HEVC to HEVC	4k	4	1	8	0	61	96	260	2
HEVC to AV1	4k	4	1	8	0	46	94	224	2
VP9 to AVC	4k	4	1	8	0	65	97	214	2
VP9 to HEVC	4k	4	1	8	0	70	96	264	2
VP9 to AV1	4k	4	1	8	0	61	94	237	6
AVC to AVC	4k	8	1	8	0	67	100	195	2
AVC to HEVC	4k	8	1	8	0	75	99	256	1
AVC to AV1	4k	8	1	8	0	69	100	240	6
HEVC to AVC	4k	8	1	8	0	61	99	195	1
HEVC to HEVC	4k	8	1	8	0	62	100	255	2
HEVC to AV1	4k	8	1	8	0	55	99	240	8
VP9 to AVC	4k	8	1	8	0	66	100	198	1
VP9 to HEVC	4k	8	1	8	0	74	99	256	2
VP9 to AV1	4k	8	1	8	0	67	99	248	6
AVC to YUV	4k	1	0	10	0	47	0	198	12
HEVC to YUV	4k	1	0	10	0	49	0	199	8
VP9 to YUV	4k	1	0	10	0	24	0	125	4
AVC to YUV	4k	16	0	10	0	99	0	252	0
HEVC to YUV	4k	16	0	10	0	96	0	253	0
VP9 to YUV	4k	16	0	10	0	100	0	434	0
YUV to AVC	4k	1	0	10	0	0	68	168	63
YUV to HEVC	4k	1	0	10	0	0	63	177	63
YUV to AV1	4k	1	0	10	0	0	69	174	61
YUV to AVC	4k	4	0	10	0	0	90	190	29
YUV to HEVC	4k	4	0	10	0	0	81	224	40
YUV to AV1	4k	4	0	10	0	0	88	218	38
AVC to YUV	1080p	1	0	8	1	42	0	676	18
HEVC to YUV	1080p	1	0	8	1	46	0	660	24
VP9 to YUV	1080p	1	0	8	1	22	0	447	6
AVC to YUV	1080p	40	0	8	0	97	0	1527	2
HEVC to YUV	1080p	40	0	8	0	97	0	1520	3
VP9 to YUV	1080p	40	0	8	0	92	0	1616	0
YUV to AVC	1080p	1	0	8	1	0	54	603	28
YUV to HEVC	1080p	1	0	8	1	0	51	601	27
YUV to AV1	1080p	1	0	8	1	0	55	553	24

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1120	3
YUV to HEVC	1080p	32	0	8	0	0	99	1179	3
YUV to AV1	1080p	32	0	8	0	0	100	1024	2
AVC to AVC	1080p	1	1	8	1	68	82	865	19
AVC to HEVC	1080p	1	1	8	1	66	78	906	20
AVC to AV1	1080p	1	1	8	1	54	81	780	18
HEVC to AVC	1080p	1	1	8	1	60	78	806	28
HEVC to HEVC	1080p	1	1	8	1	63	76	860	31
HEVC to AV1	1080p	1	1	8	1	52	77	744	28
VP9 to AVC	1080p	1	1	8	1	22	40	445	6
VP9 to HEVC	1080p	1	1	8	1	22	38	445	4
VP9 to AV1	1080p	1	1	8	1	22	44	443	6
AVC to AVC	1080p	32	1	8	0	76	100	864	1
AVC to HEVC	1080p	32	1	8	0	84	99	967	0
AVC to AV1	1080p	32	1	8	0	78	99	928	2
HEVC to AVC	1080p	32	1	8	0	71	99	896	3
HEVC to HEVC	1080p	32	1	8	0	76	99	1001	1
HEVC to AV1	1080p	32	1	8	0	66	99	935	3
VP9 to AVC	1080p	32	1	8	0	70	99	992	1
VP9 to HEVC	1080p	32	1	8	0	74	99	1093	0
VP9 to AV1	1080p	32	1	8	0	69	99	992	1
AVC to YUV	1080p	1	0	10	0	30	0	479	9
HEVC to YUV	1080p	1	0	10	0	28	0	482	6
VP9 to YUV	1080p	1	0	10	0	22	0	453	6
AVC to YUV	1080p	40	0	10	0	70	0	1040	0
HEVC to YUV	1080p	40	0	10	0	81	0	1040	0
VP9 to YUV	1080p	40	0	10	0	93	0	1680	0
YUV to AVC	1080p	1	0	10	0	0	37	409	38
YUV to HEVC	1080p	1	0	10	0	0	34	411	39
YUV to AV1	1080p	1	0	10	0	0	37	386	39
YUV to AVC	1080p	32	0	10	0	0	70	774	8
YUV to HEVC	1080p	32	0	10	0	0	67	785	7
YUV to AV1	1080p	32	0	10	0	0	76	766	7
AVC to YUV	720p	1	0	8	1	45	0	1101	15
HEVC to YUV	720p	1	0	8	1	39	0	1090	24
VP9 to YUV	720p	1	0	8	1	29	0	841	6
AVC to YUV	720p	100	0	8	0	100	0	2324	0
HEVC to YUV	720p	100	0	8	0	98	0	2643	0
VP9 to YUV	720p	100	0	8	0	100	0	2411	0
YUV to AVC	720p	1	0	8	1	0	32	814	16

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	32	813	16
YUV to AV1	720p	1	0	8	1	0	34	738	13
YUV to AVC	720p	64	0	8	0	0	98	2245	1
YUV to HEVC	720p	64	0	8	0	0	95	2304	2
YUV to AV1	720p	64	0	8	0	0	97	1877	1
AVC to AVC	720p	1	1	8	1	49	48	1187	21
AVC to HEVC	720p	1	1	8	1	49	48	1201	20
AVC to AV1	720p	1	1	8	1	41	50	1028	16
HEVC to AVC	720p	1	1	8	1	41	46	1151	21
HEVC to HEVC	720p	1	1	8	1	41	46	1157	22
HEVC to AV1	720p	1	1	8	1	35	48	1006	22
VP9 to AVC	720p	1	1	8	1	29	33	839	9
VP9 to HEVC	720p	1	1	8	1	28	33	834	7
VP9 to AV1	720p	1	1	8	1	28	40	836	9
AVC to AVC	720p	64	1	8	0	98	100	1856	1
AVC to HEVC	720p	64	1	8	0	98	100	1928	0
AVC to AV1	720p	64	1	8	0	79	100	1622	0
HEVC to AVC	720p	64	1	8	0	87	100	1856	1
HEVC to HEVC	720p	64	1	8	0	87	100	1942	0
HEVC to AV1	720p	64	1	8	0	68	100	1664	0
VP9 to AVC	720p	64	1	8	0	100	100	2104	0
VP9 to HEVC	720p	64	1	8	0	98	100	2176	0
VP9 to AV1	720p	64	1	8	0	79	100	1792	0

15. T1U – Libxcoder Throughput

15.1 Decoding

15.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

15.1.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

15.2 Encoding

15.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

15.2.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

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

15.3 Transcoding

15.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

15.3.2 Command line

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

<dec_test_type> = decoding test codecs. ie. a (avc), h (hevc), etc

<enc_test_type> = encoding test codecs. ie. a (avc), h (hevc), etc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

15.4 Libxcode Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	76	0	86	8
HEVC to YUV	8k	1	0	8	1	73	0	88	7
VP9 to YUV	8k	1	0	8	1	24	0	30	2
YUV to AVC	8k	1	0	8	1	0	89	55	25
YUV to HEVC	8k	1	0	8	1	0	94	69	31
AVC to AVC	8k	1	1	8	1	72	100	47	3
AVC to HEVC	8k	1	1	8	1	77	99	64	3
HEVC to AVC	8k	1	1	8	1	61	100	46	2
HEVC to HEVC	8k	1	1	8	1	74	100	62	3
VP9 to AVC	8k	1	1	8	1	24	41	29	1
VP9 to HEVC	8k	1	1	8	1	25	40	30	2
AVC to YUV	8k	1	0	10	1	69	0	50	7
HEVC to YUV	8k	1	0	10	1	55	0	47	7
VP9 to YUV	8k	1	0	10	1	25	0	27	4
YUV to AVC	8k	1	0	10	1	0	65	34	34
YUV to HEVC	8k	1	0	10	1	0	49	34	34
AVC to YUV	4k	1	0	8	1	56	0	253	8
HEVC to YUV	4k	1	0	8	1	50	0	261	6
VP9 to YUV	4k	1	0	8	1	24	0	108	2
AVC to YUV	4k	16	0	8	0	100	0	421	0
HEVC to YUV	4k	16	0	8	0	100	0	447	0
VP9 to YUV	4k	16	0	8	0	99	0	402	0
YUV to AVC	4k	1	0	8	1	0	82	223	23
YUV to HEVC	4k	1	0	8	1	0	78	230	22
YUV to AV1	4k	1	0	8	1	0	84	216	21
YUV to AVC	4k	4	0	8	0	0	96	265	10
YUV to HEVC	4k	4	0	8	0	0	96	283	12
YUV to AV1	4k	4	0	8	0	0	96	246	10
YUV to AVC	4k	8	0	8	0	0	90	256	16
YUV to HEVC	4k	8	0	8	0	0	99	299	9
YUV to AV1	4k	8	0	8	0	0	99	258	9
AVC to AVC	4k	1	1	8	0	69	97	203	6
AVC to HEVC	4k	1	1	8	0	73	96	249	6
AVC to AV1	4k	1	1	8	0	62	95	235	6
HEVC to AVC	4k	1	1	8	0	56	97	202	4
HEVC to HEVC	4k	1	1	8	0	61	97	252	4
HEVC to AV1	4k	1	1	8	0	53	97	233	5
VP9 to AVC	4k	1	1	8	0	24	38	108	2

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	0	24	36	108	2
VP9 to AV1	4k	1	1	8	0	24	42	108	2
AVC to AVC	4k	4	1	8	0	61	96	223	4
AVC to HEVC	4k	4	1	8	0	68	96	270	4
AVC to AV1	4k	4	1	8	0	55	95	244	4
HEVC to AVC	4k	4	1	8	0	54	96	219	4
HEVC to HEVC	4k	4	1	8	0	59	96	265	4
HEVC to AV1	4k	4	1	8	0	48	95	241	4
VP9 to AVC	4k	4	1	8	0	62	96	218	3
VP9 to HEVC	4k	4	1	8	0	71	96	267	3
VP9 to AV1	4k	4	1	8	0	58	96	241	3
AVC to AVC	4k	8	1	8	0	68	99	197	2
AVC to HEVC	4k	8	1	8	0	74	99	254	3
AVC to AV1	4k	8	1	8	0	66	100	245	3
HEVC to AVC	4k	8	1	8	0	62	99	197	2
HEVC to HEVC	4k	8	1	8	0	67	99	253	2
HEVC to AV1	4k	8	1	8	0	55	100	244	3
VP9 to AVC	4k	8	1	8	0	66	99	202	2
VP9 to HEVC	4k	8	1	8	0	74	100	261	2
VP9 to AV1	4k	8	1	8	0	67	99	249	3
AVC to YUV	4k	1	0	10	1	47	0	185	7
HEVC to YUV	4k	1	0	10	1	42	0	181	6
VP9 to YUV	4k	1	0	10	1	24	0	124	3
AVC to YUV	4k	16	0	10	0	100	0	255	0
HEVC to YUV	4k	16	0	10	0	100	0	254	0
VP9 to YUV	4k	16	0	10	0	92	0	435	0
YUV to AVC	4k	1	0	10	1	0	46	126	24
YUV to HEVC	4k	1	0	10	1	0	43	128	25
YUV to AV1	4k	1	0	10	1	0	47	122	24
YUV to AVC	4k	4	0	10	0	0	60	160	26
YUV to HEVC	4k	4	0	10	0	0	54	160	26
YUV to AV1	4k	4	0	10	0	0	58	148	27
AVC to YUV	1080p	1	0	8	1	41	0	660	14
HEVC to YUV	1080p	1	0	8	1	48	0	699	8
VP9 to YUV	1080p	1	0	8	1	22	0	445	3
AVC to YUV	1080p	40	0	8	0	94	0	1482	0
HEVC to YUV	1080p	40	0	8	0	99	0	1539	0
VP9 to YUV	1080p	40	0	8	0	81	0	1527	0
YUV to AVC	1080p	1	0	8	1	0	58	648	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	641	17
YUV to AV1	1080p	1	0	8	1	0	53	530	15
YUV to AVC	1080p	32	0	8	0	0	99	1150	2
YUV to HEVC	1080p	32	0	8	0	0	99	1217	2
YUV to AV1	1080p	32	0	8	0	0	99	1052	2
AVC to AVC	1080p	1	1	8	0	70	85	896	15
AVC to HEVC	1080p	1	1	8	0	70	84	950	16
AVC to AV1	1080p	1	1	8	0	55	85	808	15
HEVC to AVC	1080p	1	1	8	0	68	87	887	10
HEVC to HEVC	1080p	1	1	8	0	71	85	952	12
HEVC to AV1	1080p	1	1	8	0	59	86	809	12
VP9 to AVC	1080p	1	1	8	0	22	40	443	4
VP9 to HEVC	1080p	1	1	8	0	22	38	447	3
VP9 to AV1	1080p	1	1	8	0	22	44	445	4
AVC to AVC	1080p	32	1	8	0	77	99	890	0
AVC to HEVC	1080p	32	1	8	0	86	99	1006	0
AVC to AV1	1080p	32	1	8	0	78	100	962	1
HEVC to AVC	1080p	32	1	8	0	71	99	925	0
HEVC to HEVC	1080p	32	1	8	0	77	99	1037	0
HEVC to AV1	1080p	32	1	8	0	69	99	970	0
VP9 to AVC	1080p	32	1	8	0	70	99	1015	0
VP9 to HEVC	1080p	32	1	8	0	74	99	1133	0
VP9 to AV1	1080p	32	1	8	0	70	99	1023	0
AVC to YUV	1080p	1	0	10	1	30	0	474	6
HEVC to YUV	1080p	1	0	10	1	28	0	480	6
VP9 to YUV	1080p	1	0	10	1	22	0	454	3
AVC to YUV	1080p	40	0	10	0	70	0	1012	0
HEVC to YUV	1080p	40	0	10	0	71	0	1016	0
VP9 to YUV	1080p	40	0	10	0	80	0	1545	0
YUV to AVC	1080p	1	0	10	1	0	33	365	19
YUV to HEVC	1080p	1	0	10	1	0	30	362	20
YUV to AV1	1080p	1	0	10	1	0	32	324	18
YUV to AVC	1080p	32	0	10	0	0	65	735	5
YUV to HEVC	1080p	32	0	10	0	0	62	741	5
YUV to AV1	1080p	32	0	10	0	0	67	691	8
AVC to YUV	720p	1	0	8	1	41	0	1029	13
HEVC to YUV	720p	1	0	8	1	39	0	1109	7
VP9 to YUV	720p	1	0	8	1	29	0	838	4
AVC to YUV	720p	100	0	8	0	100	0	2532	0

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	91	0	2602	0
VP9 to YUV	720p	100	0	8	0	94	0	2461	0
YUV to AVC	720p	1	0	8	1	0	35	884	11
YUV to HEVC	720p	1	0	8	1	0	35	877	11
YUV to AV1	720p	1	0	8	1	0	32	689	9
YUV to AVC	720p	64	0	8	0	0	96	2441	1
YUV to HEVC	720p	64	0	8	0	0	95	2464	2
YUV to AV1	720p	64	0	8	0	0	100	2093	1
AVC to AVC	720p	1	1	8	0	52	50	1253	15
AVC to HEVC	720p	1	1	8	0	52	50	1262	15
AVC to AV1	720p	1	1	8	0	44	50	1028	15
HEVC to AVC	720p	1	1	8	0	46	51	1257	11
HEVC to HEVC	720p	1	1	8	0	46	51	1275	11
HEVC to AV1	720p	1	1	8	0	39	49	1035	11
VP9 to AVC	720p	1	1	8	0	28	33	839	6
VP9 to HEVC	720p	1	1	8	0	28	33	841	6
VP9 to AV1	720p	1	1	8	0	28	39	831	7
AVC to AVC	720p	64	1	8	0	100	100	1999	0
AVC to HEVC	720p	64	1	8	0	100	100	2087	0
AVC to AV1	720p	64	1	8	0	95	100	1757	0
HEVC to AVC	720p	64	1	8	0	85	99	2002	0
HEVC to HEVC	720p	64	1	8	0	97	100	2083	0
HEVC to AV1	720p	64	1	8	0	89	100	1758	0
VP9 to AVC	720p	64	1	8	0	100	100	2253	0
VP9 to HEVC	720p	64	1	8	0	100	100	2314	0
VP9 to AV1	720p	64	1	8	0	100	100	1900	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	66.11	67.06	65.18	0.14
YUV to HEVC	8k	1	63.44	69.33	61.48	1.38
YUV to AVC	4k	1	17.69	20.03	17.32	0.06
YUV to HEVC	4k	1	19.02	21.38	17.94	0.14
YUV to AV1	4k	1	24.93	29.34	18.87	0.85
YUV to AVC	4k	4	17.82	22.32	16.96	0.2
YUV to HEVC	4k	4	19.42	23.22	17.55	1.23
YUV to AV1	4k	4	25.17	29.68	18.89	0.83
YUV to AVC	4k	8	20.86	30.39	17.21	23.46
YUV to HEVC	4k	8	24.85	36.4	17.79	30.85
YUV to AV1	4k	8	42.82	52.52	29.71	7.48
YUV to AVC	1080p	1	5.44	6.39	5.26	0.01
YUV to HEVC	1080p	1	5.74	9.76	5.37	0.07
YUV to AV1	1080p	1	7.72	8.51	5.79	0.06
YUV to AVC	1080p	32	8.56	16.57	5.54	2.17
YUV to HEVC	1080p	32	10.26	15.43	5.91	2.4
YUV to AV1	1080p	32	47.14	54.91	39.65	2.97
YUV to AVC	720p	1	3.11	3.79	2.78	0.02
YUV to HEVC	720p	1	3.16	3.57	2.89	0.01
YUV to AV1	720p	1	4.16	4.54	3.33	0.01
YUV to AVC	720p	64	7.44	12.73	4.65	0.92
YUV to HEVC	720p	64	7.95	13.26	5.04	0.92
YUV to AV1	720p	64	45.7	54.09	38.63	3.44

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	88	0	113	4
HEVC to RGBA	8k	1	89	0	117	5
VP9 to RGBA	8k	1	22	0	31	1
AVC to RGBA	4k	1	21	0	109	5
AVC to RGBA	4k	16	94	2	468	1
HEVC to RGBA	4k	1	22	0	134	8
HEVC to RGBA	4k	16	94	3	549	2
VP9 to RGBA	4k	1	21	0	110	5
VP9 to RGBA	4k	16	92	2	461	1
AVC to RGBA	1080p	40	92	12	1572	1
HEVC to RGBA	1080p	40	94	11	1590	1
VP9 to RGBA	1080p	40	90	14	2002	0
AVC to RGBA	720p	100	93	18	2367	0
HEVC to RGBA	720p	100	89	19	2592	0
VP9 to RGBA	720p	64	89	17	2432	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: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)

18.2 Streaming Ladder Generation Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	8	28	90	2	376	4
AVC to HEVC	8	28	90	2	400	3
AVC to AV1	8	22	89	1	336	3
HEVC to AVC	8	31	91	2	376	4
HEVC to HEVC	8	30	91	2	399	4
HEVC to AV1	8	23	92	1	344	4
VP9 to AVC	8	38	93	2	375	3
VP9 to HEVC	8	37	90	2	400	4
VP9 to AV1	8	29	90	1	342	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	53	139	70
RGBA to HEVC	4k	1	1	48	140	67
RGBA to AV1	4k	1	1	58	143	73
RGBA to AVC	4k	4	0	72	148	36
RGBA to HEVC	4k	4	0	64	155	40
RGBA to AV1	4k	4	0	65	157	43
RGBA to AVC	4k	8	0	66	160	32
RGBA to HEVC	4k	8	0	61	162	33
RGBA to AV1	4k	8	0	66	162	38
RGBA to AVC	1080p	1	1	35	379	46
RGBA to HEVC	1080p	1	1	33	386	46
RGBA to AV1	1080p	1	1	37	376	48
RGBA to AVC	1080p	16	0	61	580	21
RGBA to HEVC	1080p	16	0	58	592	20
RGBA to AV1	1080p	16	0	61	591	22
RGBA to AVC	1080p	32	0	62	608	12
RGBA to HEVC	1080p	32	0	58	635	10
RGBA to AV1	1080p	32	0	64	622	13
RGBA to AVC	720p	1	1	29	632	56
RGBA to HEVC	720p	1	1	28	635	52
RGBA to AV1	720p	1	1	34	623	52
RGBA to AVC	720p	16	0	48	1128	34
RGBA to HEVC	720p	16	0	47	1118	33
RGBA to AV1	720p	16	0	55	1080	32
RGBA to AVC	720p	32	0	48	1109	31
RGBA to HEVC	720p	32	0	45	1099	31
RGBA to AV1	720p	32	0	54	1112	32

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	97	264	15
YUV to HEVC	4k	4	0	0	1	96	284	13
YUV to AV1	4k	4	0	0	1	96	248	12
YUV to AVC	4k	4	0	0	2	97	263	14
YUV to HEVC	4k	4	0	0	2	98	148	7
YUV to AV1	4k	4	0	0	2	98	120	6
YUV to AVC	4k	4	0	0	3	96	260	15
YUV to HEVC	4k	4	0	0	3	99	88	4
YUV to AV1	4k	4	0	0	3	99	64	4
YUV to AVC	4k	4	0	1	1	97	164	8
YUV to HEVC	4k	4	0	1	1	96	204	12
YUV to AVC	4k	4	0	1	2	98	164	9
YUV to HEVC	4k	4	0	1	2	100	96	5
YUV to AVC	4k	4	0	1	3	98	164	9
YUV to HEVC	4k	4	0	1	3	99	60	3
YUV to AVC	4k	4	4	0	1	100	172	9
YUV to HEVC	4k	4	4	0	1	99	198	11
YUV to AV1	4k	4	4	0	1	100	172	9
YUV to AVC	4k	4	4	0	2	99	172	9
YUV to HEVC	4k	4	4	0	2	99	120	7
YUV to AV1	4k	4	4	0	2	99	100	6
YUV to AVC	4k	4	4	0	3	99	172	9
YUV to HEVC	4k	4	4	0	3	100	76	3
YUV to AV1	4k	4	4	0	3	97	64	4
YUV to AVC	4k	4	4	1	1	99	120	7
YUV to HEVC	4k	4	4	1	1	99	156	9
YUV to AVC	4k	4	4	1	2	100	120	6
YUV to HEVC	4k	4	4	1	2	100	80	4
YUV to AVC	4k	4	4	1	3	99	120	5
YUV to HEVC	4k	4	4	1	3	99	56	3
YUV to AVC	4k	4	16	0	1	99	172	9
YUV to HEVC	4k	4	16	0	1	100	196	10
YUV to AV1	4k	4	16	0	1	99	172	9
YUV to AVC	4k	4	16	0	2	99	172	10
YUV to HEVC	4k	4	16	0	2	99	120	6
YUV to AV1	4k	4	16	0	2	100	100	6
YUV to AVC	4k	4	16	0	3	99	172	9
YUV to HEVC	4k	4	16	0	3	99	76	7

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	4k	4	16	0	3	99	64	4
YUV to AVC	4k	4	16	1	1	100	120	6
YUV to HEVC	4k	4	16	1	1	100	156	8
YUV to AVC	4k	4	16	1	2	99	120	6
YUV to HEVC	4k	4	16	1	2	100	80	5
YUV to AVC	4k	4	16	1	3	99	120	6
YUV to HEVC	4k	4	16	1	3	98	56	3
YUV to AVC	4k	4	40	0	1	99	171	9
YUV to HEVC	4k	4	40	0	1	99	196	11
YUV to AV1	4k	4	40	0	1	99	172	10
YUV to AVC	4k	4	40	0	2	100	169	9
YUV to HEVC	4k	4	40	0	2	98	116	6
YUV to AV1	4k	4	40	0	2	99	100	5
YUV to AVC	4k	4	40	0	3	99	171	11
YUV to HEVC	4k	4	40	0	3	100	76	3
YUV to AV1	4k	4	40	0	3	100	60	3
YUV to AVC	4k	4	40	1	1	100	120	6
YUV to HEVC	4k	4	40	1	1	99	152	9
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	100	55	2
YUV to AVC	1080p	20	0	0	1	99	1100	4
YUV to HEVC	1080p	20	0	0	1	99	1160	4
YUV to AV1	1080p	20	0	0	1	99	1005	3
YUV to AVC	1080p	20	0	0	2	99	1100	5
YUV to HEVC	1080p	20	0	0	2	100	600	2
YUV to AV1	1080p	20	0	0	2	100	480	2
YUV to AVC	1080p	20	0	0	3	99	1100	4
YUV to HEVC	1080p	20	0	0	3	99	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	100	660	2
YUV to HEVC	1080p	20	0	1	2	100	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	615	2
YUV to HEVC	1080p	20	4	0	1	99	720	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	99	620	2
YUV to AVC	1080p	20	4	0	2	99	618	2
YUV to HEVC	1080p	20	4	0	2	99	440	1
YUV to AV1	1080p	20	4	0	2	99	377	1
YUV to AVC	1080p	20	4	0	3	99	615	2
YUV to HEVC	1080p	20	4	0	3	99	300	1
YUV to AV1	1080p	20	4	0	3	99	239	1
YUV to AVC	1080p	20	4	1	1	100	440	1
YUV to HEVC	1080p	20	4	1	1	99	564	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	711	2
YUV to AV1	1080p	20	16	0	1	99	620	2
YUV to AVC	1080p	20	16	0	2	99	607	2
YUV to HEVC	1080p	20	16	0	2	99	440	1
YUV to AV1	1080p	20	16	0	2	99	361	1
YUV to AVC	1080p	20	16	0	3	99	604	2
YUV to HEVC	1080p	20	16	0	3	99	296	1
YUV to AV1	1080p	20	16	0	3	100	237	1
YUV to AVC	1080p	20	16	1	1	99	440	2
YUV to HEVC	1080p	20	16	1	1	100	560	1
YUV to AVC	1080p	20	16	1	2	99	440	1
YUV to HEVC	1080p	20	16	1	2	100	319	1
YUV to AVC	1080p	20	16	1	3	100	440	2
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	700	2
YUV to AV1	1080p	20	40	0	1	99	620	2
YUV to AVC	1080p	20	40	0	2	99	600	2
YUV to HEVC	1080p	20	40	0	2	99	440	1
YUV to AV1	1080p	20	40	0	2	100	360	1
YUV to AVC	1080p	20	40	0	3	99	600	2
YUV to HEVC	1080p	20	40	0	3	99	282	1
YUV to AV1	1080p	20	40	0	3	100	229	1
YUV to AVC	1080p	20	40	1	1	99	440	1
YUV to HEVC	1080p	20	40	1	1	99	560	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	2	100	440	2
YUV to HEVC	1080p	20	40	1	2	99	300	1
YUV to AVC	1080p	20	40	1	3	99	440	1
YUV to HEVC	1080p	20	40	1	3	100	206	1
YUV to AVC	720p	40	0	0	1	94	2209	2
YUV to HEVC	720p	40	0	0	1	90	2232	2
YUV to AV1	720p	40	0	0	1	96	1924	2
YUV to AVC	720p	40	0	0	2	93	2202	2
YUV to HEVC	720p	40	0	0	2	99	1320	1
YUV to AV1	720p	40	0	0	2	99	1040	1
YUV to AVC	720p	40	0	0	3	94	2213	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	2
YUV to AVC	720p	40	0	1	2	99	1480	1
YUV to HEVC	720p	40	0	1	2	99	840	0
YUV to AVC	720p	40	0	1	3	99	1480	1
YUV to HEVC	720p	40	0	1	3	99	520	0
YUV to AVC	720p	40	4	0	1	99	1247	1
YUV to HEVC	720p	40	4	0	1	100	1280	1
YUV to AV1	720p	40	4	0	1	100	960	1
YUV to AVC	720p	40	4	0	2	99	1249	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	1244	1
YUV to HEVC	720p	40	4	0	3	99	640	0
YUV to AV1	720p	40	4	0	3	99	480	0
YUV to AVC	720p	40	4	1	1	99	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	655	0
YUV to AVC	720p	40	4	1	3	99	920	1
YUV to HEVC	720p	40	4	1	3	99	451	0
YUV to AVC	720p	40	16	0	1	99	1240	1
YUV to HEVC	720p	40	16	0	1	100	1281	1
YUV to AV1	720p	40	16	0	1	100	1002	1
YUV to AVC	720p	40	16	0	2	99	1240	1
YUV to HEVC	720p	40	16	0	2	99	920	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
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	99	480	0
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	1
YUV to HEVC	720p	40	16	1	3	100	440	0
YUV to AVC	720p	40	40	0	1	99	1240	1
YUV to HEVC	720p	40	40	0	1	100	1244	1
YUV to AV1	720p	40	40	0	1	100	1000	1
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	630	1
YUV to AV1	720p	40	40	0	3	99	480	0
YUV to AVC	720p	40	40	1	1	99	920	1
YUV to HEVC	720p	40	40	1	1	99	1142	1
YUV to AVC	720p	40	40	1	2	99	919	1
YUV to HEVC	720p	40	40	1	2	99	640	1
YUV to AVC	720p	40	40	1	3	99	920	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

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	620	2
YUV to HEVC	1080p	20	0	0	1	19	99	720	2
YUV to AV1	1080p	20	0	0	1	19	99	622	2
YUV to AVC	1080p	20	0	0	2	19	99	620	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	618	2
YUV to HEVC	1080p	20	0	0	3	19	99	300	1
YUV to AV1	1080p	20	0	0	3	19	99	240	1
YUV to AVC	1080p	20	0	1	1	19	99	440	1
YUV to HEVC	1080p	20	0	1	1	19	99	577	2
YUV to AVC	1080p	20	0	1	2	19	99	440	2
YUV to HEVC	1080p	20	0	1	2	19	100	320	1
YUV to AVC	1080p	20	0	1	3	19	100	440	2
YUV to HEVC	1080p	20	0	1	3	19	100	220	1
YUV to AVC	1080p	20	4	0	1	19	99	614	2
YUV to HEVC	1080p	20	4	0	1	19	99	718	3
YUV to AV1	1080p	20	4	0	1	19	99	620	2
YUV to AVC	1080p	20	4	0	2	19	99	616	2
YUV to HEVC	1080p	20	4	0	2	19	99	440	1
YUV to AV1	1080p	20	4	0	2	19	99	368	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	3	19	99	614	2
YUV to HEVC	1080p	20	4	0	3	19	99	300	1
YUV to AV1	1080p	20	4	0	3	19	100	240	1
YUV to AVC	1080p	20	4	1	1	19	99	440	1
YUV to HEVC	1080p	20	4	1	1	19	99	570	2
YUV to AVC	1080p	20	4	1	2	19	99	440	2
YUV to HEVC	1080p	20	4	1	2	19	100	320	1
YUV to AVC	1080p	20	4	1	3	19	99	440	1
YUV to HEVC	1080p	20	4	1	3	19	99	219	1
YUV to AVC	1080p	20	16	0	1	19	99	602	2
YUV to HEVC	1080p	20	16	0	1	19	99	708	2
YUV to AV1	1080p	20	16	0	1	19	99	620	2
YUV to AVC	1080p	20	16	0	2	19	99	606	2
YUV to HEVC	1080p	20	16	0	2	19	99	440	1
YUV to AV1	1080p	20	16	0	2	19	100	360	1
YUV to AVC	1080p	20	16	0	3	19	99	607	2
YUV to HEVC	1080p	20	16	0	3	19	99	297	1
YUV to AV1	1080p	20	16	0	3	19	99	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	99	320	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	620	2
YUV to AVC	1080p	20	40	0	2	19	99	600	2
YUV to HEVC	1080p	20	40	0	2	19	99	440	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	284	1
YUV to AV1	1080p	20	40	0	3	19	100	228	1
YUV to AVC	1080p	20	40	1	1	19	100	440	1
YUV to HEVC	1080p	20	40	1	1	19	99	560	2
YUV to AVC	1080p	20	40	1	2	19	99	440	1
YUV to HEVC	1080p	20	40	1	2	19	99	300	1
YUV to AVC	1080p	20	40	1	3	19	100	440	2
YUV to HEVC	1080p	20	40	1	3	19	99	205	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	23	99	617	2
YUV to HEVC	1080p	20	0	0	1	23	99	720	3
YUV to AV1	1080p	20	0	0	1	23	99	629	2
YUV to AVC	1080p	20	0	0	2	23	99	618	2
YUV to HEVC	1080p	20	0	0	2	23	100	440	1
YUV to AV1	1080p	20	0	0	2	23	99	380	1
YUV to AVC	1080p	20	0	0	3	23	99	619	2
YUV to HEVC	1080p	20	0	0	3	23	100	300	1
YUV to AV1	1080p	20	0	0	3	23	100	240	1
YUV to AVC	1080p	20	0	1	1	23	99	440	1
YUV to HEVC	1080p	20	0	1	1	23	99	577	1
YUV to AVC	1080p	20	0	1	2	23	99	440	1
YUV to HEVC	1080p	20	0	1	2	23	100	320	1
YUV to AVC	1080p	20	0	1	3	23	99	440	1
YUV to HEVC	1080p	20	0	1	3	23	100	220	1
YUV to AVC	1080p	20	4	0	1	23	99	616	2
YUV to HEVC	1080p	20	4	0	1	23	99	717	3
YUV to AV1	1080p	20	4	0	1	23	99	620	2
YUV to AVC	1080p	20	4	0	2	23	99	617	2
YUV to HEVC	1080p	20	4	0	2	23	99	440	1
YUV to AV1	1080p	20	4	0	2	23	99	361	1
YUV to AVC	1080p	20	4	0	3	23	99	617	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	99	565	1
YUV to AVC	1080p	20	4	1	2	23	99	440	2
YUV to HEVC	1080p	20	4	1	2	23	99	320	1
YUV to AVC	1080p	20	4	1	3	23	99	440	1
YUV to HEVC	1080p	20	4	1	3	23	99	219	1
YUV to AVC	1080p	20	16	0	1	23	99	602	2
YUV to HEVC	1080p	20	16	0	1	23	99	708	2
YUV to AV1	1080p	20	16	0	1	23	99	620	2
YUV to AVC	1080p	20	16	0	2	23	99	604	2
YUV to HEVC	1080p	20	16	0	2	23	99	440	1
YUV to AV1	1080p	20	16	0	2	23	100	360	1
YUV to AVC	1080p	20	16	0	3	23	99	602	2
YUV to HEVC	1080p	20	16	0	3	23	99	300	1
YUV to AV1	1080p	20	16	0	3	23	100	240	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	16	1	1	23	99	440	1
YUV to HEVC	1080p	20	16	1	1	23	100	560	2
YUV to AVC	1080p	20	16	1	2	23	99	440	1
YUV to HEVC	1080p	20	16	1	2	23	99	318	1
YUV to AVC	1080p	20	16	1	3	23	99	440	1
YUV to HEVC	1080p	20	16	1	3	23	100	220	1
YUV to AVC	1080p	20	40	0	1	23	99	600	2
YUV to HEVC	1080p	20	40	0	1	23	99	700	3
YUV to AV1	1080p	20	40	0	1	23	99	619	2
YUV to AVC	1080p	20	40	0	2	23	99	600	2
YUV to HEVC	1080p	20	40	0	2	23	99	440	1
YUV to AV1	1080p	20	40	0	2	23	99	360	1
YUV to AVC	1080p	20	40	0	3	23	99	600	2
YUV to HEVC	1080p	20	40	0	3	23	99	287	1
YUV to AV1	1080p	20	40	0	3	23	100	226	1
YUV to AVC	1080p	20	40	1	1	23	99	439	1
YUV to HEVC	1080p	20	40	1	1	23	100	560	2
YUV to AVC	1080p	20	40	1	2	23	99	440	1
YUV to HEVC	1080p	20	40	1	2	23	100	300	1
YUV to AVC	1080p	20	40	1	3	23	99	440	1
YUV to HEVC	1080p	20	40	1	3	23	99	202	1
YUV to AVC	1080p	20	0	0	1	27	99	620	2
YUV to HEVC	1080p	20	0	0	1	27	99	720	3
YUV to AV1	1080p	20	0	0	1	27	99	625	3
YUV to AVC	1080p	20	0	0	2	27	99	620	2
YUV to HEVC	1080p	20	0	0	2	27	99	440	1
YUV to AV1	1080p	20	0	0	2	27	99	380	1
YUV to AVC	1080p	20	0	0	3	27	99	619	2
YUV to HEVC	1080p	20	0	0	3	27	99	300	1
YUV to AV1	1080p	20	0	0	3	27	99	240	1
YUV to AVC	1080p	20	0	1	1	27	99	440	1
YUV to HEVC	1080p	20	0	1	1	27	99	573	2
YUV to AVC	1080p	20	0	1	2	27	99	440	2
YUV to HEVC	1080p	20	0	1	2	27	100	320	1
YUV to AVC	1080p	20	0	1	3	27	100	440	1
YUV to HEVC	1080p	20	0	1	3	27	99	220	1
YUV to AVC	1080p	20	4	0	1	27	99	616	2
YUV to HEVC	1080p	20	4	0	1	27	99	718	3
YUV to AV1	1080p	20	4	0	1	27	99	621	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	2	27	99	617	2
YUV to HEVC	1080p	20	4	0	2	27	99	440	1
YUV to AV1	1080p	20	4	0	2	27	99	363	1
YUV to AVC	1080p	20	4	0	3	27	99	617	2
YUV to HEVC	1080p	20	4	0	3	27	99	300	1
YUV to AV1	1080p	20	4	0	3	27	100	240	1
YUV to AVC	1080p	20	4	1	1	27	99	440	1
YUV to HEVC	1080p	20	4	1	1	27	99	571	1
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	100	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	1
YUV to HEVC	1080p	20	16	0	1	27	99	708	2
YUV to AV1	1080p	20	16	0	1	27	99	620	2
YUV to AVC	1080p	20	16	0	2	27	99	605	2
YUV to HEVC	1080p	20	16	0	2	27	99	440	1
YUV to AV1	1080p	20	16	0	2	27	99	360	1
YUV to AVC	1080p	20	16	0	3	27	99	601	2
YUV to HEVC	1080p	20	16	0	3	27	99	300	1
YUV to AV1	1080p	20	16	0	3	27	99	238	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	100	319	1
YUV to AVC	1080p	20	16	1	3	27	99	440	1
YUV to HEVC	1080p	20	16	1	3	27	100	220	1
YUV to AVC	1080p	20	40	0	1	27	99	600	2
YUV to HEVC	1080p	20	40	0	1	27	99	700	3
YUV to AV1	1080p	20	40	0	1	27	99	619	2
YUV to AVC	1080p	20	40	0	2	27	99	600	2
YUV to HEVC	1080p	20	40	0	2	27	100	440	1
YUV to AV1	1080p	20	40	0	2	27	99	360	1
YUV to AVC	1080p	20	40	0	3	27	99	600	2
YUV to HEVC	1080p	20	40	0	3	27	100	286	1
YUV to AV1	1080p	20	40	0	3	27	99	229	1
YUV to AVC	1080p	20	40	1	1	27	99	439	1
YUV to HEVC	1080p	20	40	1	1	27	99	560	2
YUV to AVC	1080p	20	40	1	2	27	99	439	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	40	1	2	27	100	300	1
YUV to AVC	1080p	20	40	1	3	27	99	440	1
YUV to HEVC	1080p	20	40	1	3	27	99	201	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	JOBS	FPS	CPU	DEC_LOAD	ENC_LOAD	SCALER_LOAD
AVC to AVC	1	241	9	13	19	4
AVC to HEVC	1	256	9	14	19	4
AVC to AV1	1	221	9	12	19	3
HEVC to AVC	1	241	11	14	20	4
HEVC to HEVC	1	256	12	14	19	4
HEVC to AV1	1	220	12	12	19	3
VP9 to AVC	1	241	8	18	19	4
VP9 to HEVC	1	254	8	19	19	4
VP9 to AV1	1	218	8	16	19	3
AVC to AVC	16	944	2	70	91	23
AVC to HEVC	16	1040	2	78	93	27
AVC to AV1	16	976	2	69	92	24
HEVC to AVC	16	960	3	71	92	24
HEVC to HEVC	16	1060	4	75	90	26
HEVC to AV1	16	976	3	67	93	24
VP9 to AVC	16	944	2	86	92	23
VP9 to HEVC	16	1023	2	88	86	24
VP9 to AV1	16	992	2	86	94	24
AVC to AVC	32	864	1	74	92	23
AVC to HEVC	32	961	1	82	94	27
AVC to AV1	32	930	1	75	93	25
HEVC to AVC	32	896	1	73	92	24
HEVC to HEVC	32	1017	1	80	93	27
HEVC to AV1	32	960	1	71	92	25
VP9 to AVC	32	896	1	85	92	23
VP9 to HEVC	32	1005	1	92	91	26
VP9 to AV1	32	972	1	85	92	24

23. A Note on T1M Persistent Configurations

T1M currently supports two configurations:

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

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

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

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

24.1 Decoding

24.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

24.1.2 Command Line

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

<dec> is the decoder codec. eg h264_ni_quadra_dec, h265_ni_quadra_dec, vp9_ni_quadra_dec

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

24.2 Encoding

24.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

24.2.2 Command Line

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

<enc> is the encoder codec. eg h264_ni_quadra_enc, h265_ni_quadra_enc, av1_ni_quadra_enc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

24.3 Transcoding

24.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

24.3.2 Command line

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

<dec> is the decoder codec. ie h264_ni_quadra_dec, h265_ni_quadra_dec, vp9_ni_quadra_dec

<enc> is the encoder codec. ie h264_ni_quadra_enc, h265_ni_quadra_enc, av1_ni_quadra_enc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

24.4 FFmpeg Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	4k	1	0	8	1	43	0	151	36
HEVC to YUV	4k	1	0	8	1	46	0	156	40
VP9 to YUV	4k	1	0	8	1	24	0	83	28
AVC to YUV	4k	5	0	8	0	83	0	248	18
HEVC to YUV	4k	5	0	8	0	89	0	264	13
VP9 to YUV	4k	5	0	8	0	93	0	260	12
YUV to AVC	4k	1	0	8	1	0	93	181	68
YUV to HEVC	4k	1	0	8	1	0	86	188	68
YUV to AV1	4k	1	0	8	1	0	93	177	68
YUV to AVC	4k	3	0	8	0	0	73	150	38
YUV to HEVC	4k	3	0	8	0	0	72	159	43
YUV to AV1	4k	3	0	8	0	0	72	138	37
YUV to AVC	4k	5	0	8	0	0	97	190	32
YUV to HEVC	4k	5	0	8	0	0	98	214	33
YUV to AV1	4k	5	0	8	0	0	98	185	28
AVC to AVC	4k	1	1	8	1	77	96	127	42
AVC to HEVC	4k	1	1	8	1	76	88	157	48
AVC to AV1	4k	1	1	8	1	76	94	154	48
HEVC to AVC	4k	1	1	8	1	53	97	125	40
HEVC to HEVC	4k	1	1	8	1	56	97	156	42
HEVC to AV1	4k	1	1	8	1	49	92	150	41
VP9 to AVC	4k	1	1	8	1	24	39	83	23
VP9 to HEVC	4k	1	1	8	1	24	37	83	24
VP9 to AV1	4k	1	1	8	1	24	44	83	27
AVC to AVC	4k	3	1	8	0	48	71	123	26
AVC to HEVC	4k	3	1	8	0	49	72	150	33
AVC to AV1	4k	3	1	8	0	40	72	135	30
HEVC to AVC	4k	3	1	8	0	41	73	120	29
HEVC to HEVC	4k	3	1	8	0	41	73	147	33
HEVC to AV1	4k	3	1	8	0	37	72	132	24
VP9 to AVC	4k	3	1	8	0	44	72	122	18
VP9 to HEVC	4k	3	1	8	0	48	73	147	30
VP9 to AV1	4k	3	1	8	0	42	72	135	28
AVC to AVC	4k	5	1	8	0	64	98	120	22
AVC to HEVC	4k	5	1	8	0	71	97	157	25
AVC to AV1	4k	5	1	8	0	65	96	157	24
HEVC to AVC	4k	5	1	8	0	56	97	121	20
HEVC to HEVC	4k	5	1	8	0	59	96	160	25
HEVC to AV1	4k	5	1	8	0	52	97	160	24

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AVC	4k	5	1	8	0	60	96	122	21
VP9 to HEVC	4k	5	1	8	0	66	97	160	24
VP9 to AV1	4k	5	1	8	0	60	96	158	25
AVC to YUV	4k	1	0	10	0	37	0	100	32
HEVC to YUV	4k	1	0	10	0	33	0	100	32
VP9 to YUV	4k	1	0	10	0	24	0	97	25
AVC to YUV	4k	16	0	10	0	50	0	133	10
HEVC to YUV	4k	16	0	10	0	71	0	131	9
VP9 to YUV	4k	16	0	10	0	78	0	268	10
YUV to AVC	4k	1	0	10	0	0	52	105	54
YUV to HEVC	4k	1	0	10	0	0	48	107	60
YUV to AV1	4k	1	0	10	0	0	55	105	60
YUV to AVC	4k	3	0	10	0	0	62	114	58
YUV to HEVC	4k	3	0	10	0	0	53	115	56
YUV to AV1	4k	3	0	10	0	0	61	115	57
AVC to YUV	1080p	1	0	8	1	33	0	415	62
HEVC to YUV	1080p	1	0	8	1	36	0	400	64
VP9 to YUV	1080p	1	0	8	1	22	0	351	46
AVC to YUV	1080p	18	0	8	0	95	0	962	9
HEVC to YUV	1080p	18	0	8	0	92	0	1005	12
VP9 to YUV	1080p	18	0	8	0	76	0	990	6
YUV to AVC	1080p	1	0	8	1	0	53	429	73
YUV to HEVC	1080p	1	0	8	1	0	50	433	68
YUV to AV1	1080p	1	0	8	1	0	54	397	66
YUV to AVC	1080p	20	0	8	0	0	99	816	13
YUV to HEVC	1080p	20	0	8	0	0	99	850	15
YUV to AV1	1080p	20	0	8	0	0	99	740	13
AVC to AVC	1080p	1	1	8	1	62	72	537	76
AVC to HEVC	1080p	1	1	8	1	63	71	587	75
AVC to AV1	1080p	1	1	8	1	55	79	543	72
HEVC to AVC	1080p	1	1	8	1	55	79	519	91
HEVC to HEVC	1080p	1	1	8	1	55	71	572	88
HEVC to AV1	1080p	1	1	8	1	53	80	525	87
VP9 to AVC	1080p	1	1	8	1	22	42	350	45
VP9 to HEVC	1080p	1	1	8	1	22	40	351	39
VP9 to AV1	1080p	1	1	8	1	22	47	351	41
AVC to AVC	1080p	20	1	8	0	73	99	570	9
AVC to HEVC	1080p	20	1	8	0	80	99	654	9
AVC to AV1	1080p	20	1	8	0	75	99	640	8
HEVC to AVC	1080p	20	1	8	0	69	100	600	11

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to HEVC	1080p	20	1	8	0	71	99	682	10
HEVC to AV1	1080p	20	1	8	0	68	99	660	6
VP9 to AVC	1080p	20	1	8	0	75	100	640	8
VP9 to HEVC	1080p	20	1	8	0	76	99	740	8
VP9 to AV1	1080p	20	1	8	0	70	99	684	11
AVC to YUV	1080p	1	0	10	0	22	0	315	40
HEVC to YUV	1080p	1	0	10	0	22	0	309	40
VP9 to YUV	1080p	1	0	10	0	22	0	296	44
AVC to YUV	1080p	18	0	10	0	57	0	540	4
HEVC to YUV	1080p	18	0	10	0	60	0	540	4
VP9 to YUV	1080p	18	0	10	0	55	0	540	4
YUV to AVC	1080p	1	0	10	0	0	36	308	66
YUV to HEVC	1080p	1	0	10	0	0	34	305	74
YUV to AV1	1080p	1	0	10	0	0	39	288	69
YUV to AVC	1080p	20	0	10	0	0	57	460	14
YUV to HEVC	1080p	20	0	10	0	0	53	461	13
YUV to AV1	1080p	20	0	10	0	0	60	454	13
AVC to YUV	720p	1	0	8	1	24	0	643	62
HEVC to YUV	720p	1	0	8	1	27	0	627	68
VP9 to YUV	720p	1	0	8	1	22	0	658	57
AVC to YUV	720p	40	0	8	0	99	0	1884	5
HEVC to YUV	720p	40	0	8	0	92	0	1876	6
VP9 to YUV	720p	40	0	8	0	99	0	1961	6
YUV to AVC	720p	1	0	8	1	0	29	544	63
YUV to HEVC	720p	1	0	8	1	0	29	555	73
YUV to AV1	720p	1	0	8	1	0	30	495	93
YUV to AVC	720p	40	0	8	0	0	88	1613	8
YUV to HEVC	720p	40	0	8	0	0	90	1663	8
YUV to AV1	720p	40	0	8	0	0	92	1471	7
AVC to AVC	720p	1	1	8	1	38	47	853	80
AVC to HEVC	720p	1	1	8	1	34	44	873	79
AVC to AV1	720p	1	1	8	1	30	47	767	85
HEVC to AVC	720p	1	1	8	1	38	44	832	82
HEVC to HEVC	720p	1	1	8	1	38	43	828	71
HEVC to AV1	720p	1	1	8	1	36	51	749	68
VP9 to AVC	720p	1	1	8	1	24	36	669	61
VP9 to HEVC	720p	1	1	8	1	23	36	670	70
VP9 to AV1	720p	1	1	8	1	23	43	672	60
AVC to AVC	720p	40	1	8	0	94	100	1290	5
AVC to HEVC	720p	40	1	8	0	96	100	1412	6

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	720p	40	1	8	0	76	100	1291	6
HEVC to AVC	720p	40	1	8	0	84	98	1300	6
HEVC to HEVC	720p	40	1	8	0	87	100	1422	6
HEVC to AV1	720p	40	1	8	0	71	100	1292	5
VP9 to AVC	720p	40	1	8	0	99	100	1401	4
VP9 to HEVC	720p	40	1	8	0	99	100	1528	5
VP9 to AV1	720p	40	1	8	0	80	100	1379	5

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	48	0	152	35
HEVC to YUV	4k	1	0	8	1	43	0	157	28
VP9 to YUV	4k	1	0	8	1	24	0	82	19
AVC to YUV	4k	5	0	8	0	82	0	247	13
HEVC to YUV	4k	5	0	8	0	87	0	259	9
VP9 to YUV	4k	5	0	8	0	91	0	258	9
YUV to AVC	4k	1	0	8	1	0	65	123	48
YUV to HEVC	4k	1	0	8	1	0	55	122	49
YUV to AV1	4k	1	0	8	1	0	56	114	49
YUV to AVC	4k	3	0	8	0	0	72	150	33
YUV to HEVC	4k	3	0	8	0	0	73	160	34
YUV to AV1	4k	3	0	8	0	0	73	139	31
YUV to AVC	4k	5	0	8	0	0	98	185	28
YUV to HEVC	4k	5	0	8	0	0	98	213	28
YUV to AV1	4k	5	0	8	0	0	98	186	27
AVC to AVC	4k	1	1	8	0	74	96	128	55
AVC to HEVC	4k	1	1	8	0	82	96	162	59
AVC to AV1	4k	1	1	8	0	75	96	158	58
HEVC to AVC	4k	1	1	8	0	52	97	127	52
HEVC to HEVC	4k	1	1	8	0	58	96	157	52
HEVC to AV1	4k	1	1	8	0	53	97	150	54
VP9 to AVC	4k	1	1	8	0	24	39	82	47
VP9 to HEVC	4k	1	1	8	0	24	37	82	44
VP9 to AV1	4k	1	1	8	0	24	42	82	45
AVC to AVC	4k	3	1	8	0	47	72	122	55
AVC to HEVC	4k	3	1	8	0	47	72	152	60
AVC to AV1	4k	3	1	8	0	43	71	135	59
HEVC to AVC	4k	3	1	8	0	41	72	119	55
HEVC to HEVC	4k	3	1	8	0	41	72	144	57
HEVC to AV1	4k	3	1	8	0	34	72	132	59
VP9 to AVC	4k	3	1	8	0	47	72	122	54
VP9 to HEVC	4k	3	1	8	0	48	72	147	55
VP9 to AV1	4k	3	1	8	0	42	72	134	57
AVC to AVC	4k	5	1	8	0	65	97	122	31
AVC to HEVC	4k	5	1	8	0	72	97	158	30
AVC to AV1	4k	5	1	8	0	67	97	157	27
HEVC to AVC	4k	5	1	8	0	52	98	123	30
HEVC to HEVC	4k	5	1	8	0	60	97	159	29

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to AV1	4k	5	1	8	0	54	96	157	21
VP9 to AVC	4k	5	1	8	0	62	98	124	25
VP9 to HEVC	4k	5	1	8	0	67	98	160	30
VP9 to AV1	4k	5	1	8	0	63	96	159	29
AVC to YUV	4k	1	0	10	1	42	0	95	26
HEVC to YUV	4k	1	0	10	1	36	0	95	24
VP9 to YUV	4k	1	0	10	1	24	0	96	19
AVC to YUV	4k	5	0	10	0	56	0	130	8
HEVC to YUV	4k	5	0	10	0	64	0	132	8
VP9 to YUV	4k	5	0	10	0	80	0	263	9
YUV to AVC	4k	1	0	10	1	0	33	67	49
YUV to HEVC	4k	1	0	10	1	0	29	66	50
YUV to AV1	4k	1	0	10	1	0	36	64	49
YUV to AVC	4k	3	0	10	0	0	41	89	41
YUV to HEVC	4k	3	0	10	0	0	37	93	40
YUV to AV1	4k	3	0	10	0	0	45	83	38
AVC to YUV	1080p	1	0	8	1	29	0	366	44
HEVC to YUV	1080p	1	0	8	1	38	0	422	35
VP9 to YUV	1080p	1	0	8	1	22	0	350	27
AVC to YUV	1080p	20	0	8	0	85	0	923	8
HEVC to YUV	1080p	20	0	8	0	87	0	954	5
VP9 to YUV	1080p	20	0	8	0	73	0	933	3
YUV to AVC	1080p	1	0	8	1	0	39	340	47
YUV to HEVC	1080p	1	0	8	1	0	37	346	46
YUV to AV1	1080p	1	0	8	1	0	38	294	43
YUV to AVC	1080p	20	0	8	0	0	99	827	11
YUV to HEVC	1080p	20	0	8	0	0	99	863	12
YUV to AV1	1080p	20	0	8	0	0	100	763	11
AVC to AVC	1080p	1	1	8	0	67	78	582	73
AVC to HEVC	1080p	1	1	8	0	68	78	634	72
AVC to AV1	1080p	1	1	8	0	58	83	594	72
HEVC to AVC	1080p	1	1	8	0	59	84	583	71
HEVC to HEVC	1080p	1	1	8	0	66	84	648	68
HEVC to AV1	1080p	1	1	8	0	58	85	591	73
VP9 to AVC	1080p	1	1	8	0	22	42	350	53
VP9 to HEVC	1080p	1	1	8	0	22	40	350	54
VP9 to AV1	1080p	1	1	8	0	22	47	351	51
AVC to AVC	1080p	20	1	8	0	73	99	583	6
AVC to HEVC	1080p	20	1	8	0	78	99	664	7

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	1080p	20	1	8	0	74	99	651	7
HEVC to AVC	1080p	20	1	8	0	67	99	609	7
HEVC to HEVC	1080p	20	1	8	0	70	99	694	7
HEVC to AV1	1080p	20	1	8	0	65	99	664	7
VP9 to AVC	1080p	20	1	8	0	72	99	645	7
VP9 to HEVC	1080p	20	1	8	0	74	99	736	7
VP9 to AV1	1080p	20	1	8	0	71	99	696	7
AVC to YUV	1080p	1	0	10	1	23	0	317	30
HEVC to YUV	1080p	1	0	10	1	21	0	311	28
VP9 to YUV	1080p	1	0	10	1	22	0	294	28
AVC to YUV	1080p	20	0	10	0	51	0	532	2
HEVC to YUV	1080p	20	0	10	0	64	0	531	2
VP9 to YUV	1080p	20	0	10	0	100	0	514	2
YUV to AVC	1080p	1	0	10	1	0	25	215	46
YUV to HEVC	1080p	1	0	10	1	0	24	214	46
YUV to AV1	1080p	1	0	10	1	0	25	188	48
YUV to AVC	1080p	20	0	10	0	0	56	465	12
YUV to HEVC	1080p	20	0	10	0	0	52	465	12
YUV to AV1	1080p	20	0	10	0	0	61	440	12
AVC to YUV	720p	1	0	8	1	21	0	549	43
HEVC to YUV	720p	1	0	8	1	30	0	644	36
VP9 to YUV	720p	1	0	8	1	22	0	669	27
AVC to YUV	720p	40	0	8	0	94	0	1876	4
HEVC to YUV	720p	40	0	8	0	79	0	1734	3
VP9 to YUV	720p	40	0	8	0	92	0	1857	2
YUV to AVC	720p	1	0	8	1	0	27	514	41
YUV to HEVC	720p	1	0	8	1	0	28	518	37
YUV to AV1	720p	1	0	8	1	0	25	428	44
YUV to AVC	720p	40	0	8	0	0	93	1749	6
YUV to HEVC	720p	40	0	8	0	0	95	1799	6
YUV to AV1	720p	40	0	8	0	0	96	1571	5
AVC to AVC	720p	1	1	8	0	50	57	882	68
AVC to HEVC	720p	1	1	8	0	35	47	892	67
AVC to AV1	720p	1	1	8	0	34	54	821	75
HEVC to AVC	720p	1	1	8	0	42	47	864	66
HEVC to HEVC	720p	1	1	8	0	40	46	865	69
HEVC to AV1	720p	1	1	8	0	40	53	808	73
VP9 to AVC	720p	1	1	8	0	23	37	681	61
VP9 to HEVC	720p	1	1	8	0	23	37	682	60

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	720p	1	1	8	0	23	43	681	65
AVC to AVC	720p	40	1	8	0	100	100	1350	1
AVC to HEVC	720p	40	1	8	0	100	100	1474	1
AVC to AV1	720p	40	1	8	0	98	100	1337	1
HEVC to AVC	720p	40	1	8	0	100	100	1351	1
HEVC to HEVC	720p	40	1	8	0	100	100	1480	0
HEVC to AV1	720p	40	1	8	0	94	100	1339	1
VP9 to AVC	720p	40	1	8	0	100	100	1451	0
VP9 to HEVC	720p	40	1	8	0	100	100	1598	0
VP9 to AV1	720p	40	1	8	0	99	100	1429	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	28.21	24.07	0.16
YUV to HEVC	4k	1	26.73	30	25.16	0.26
YUV to AV1	4k	1	34.79	42.09	26.33	1.68
YUV to AVC	4k	3	25.14	28.6	24.12	0.19
YUV to HEVC	4k	3	26.94	33.79	25.18	0.41
YUV to AV1	4k	3	34.99	41.42	26.25	1.61
YUV to AVC	4k	5	29.79	53.77	24.26	27.37
YUV to HEVC	4k	5	33.76	57.99	25.55	66.98
YUV to AV1	4k	5	43.87	74.77	26.27	134.29
YUV to AVC	1080p	1	7.52	9.11	6.98	0.05
YUV to HEVC	1080p	1	8.02	10.28	7.29	0.18
YUV to AV1	1080p	1	10.56	12.07	7.83	0.12
YUV to AVC	1080p	20	10.64	19.59	7.36	3.75
YUV to HEVC	1080p	20	11.37	21.58	7.59	5.95
YUV to AV1	1080p	20	40	50.86	29.53	14.2
YUV to AVC	720p	1	4.3	5.26	3.72	0.05
YUV to HEVC	720p	1	4.48	5.04	4.05	0.04
YUV to AV1	720p	1	5.91	6.58	4.78	0.05
YUV to AVC	720p	40	6.6	13.47	4.23	1.89
YUV to HEVC	720p	40	7.06	13.24	4.12	1.79
YUV to AV1	720p	40	38.9	45.99	32.01	2.25

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	21	0	81	32
AVC to RGBA	4k	8	91	1	338	20
HEVC to RGBA	4k	1	22	0	99	34
HEVC to RGBA	4k	8	92	2	408	24
VP9 to RGBA	4k	1	22	0	82	29
VP9 to RGBA	4k	8	93	1	335	20
AVC to RGBA	1080p	20	94	9	1187	9
HEVC to RGBA	1080p	20	93	8	1171	9
VP9 to RGBA	1080p	20	93	12	1502	10
AVC to RGBA	720p	50	94	18	2143	3
HEVC to RGBA	720p	50	88	19	2211	4
VP9 to RGBA	720p	32	92	18	2248	6

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

28.1 Transcoding

28.1.1 Description

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

Decoded YUV frame is kept on device.

The YUV frames are encoded with hardware encoder.

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

28.1.2 Command line

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

<dec> is the decoder codec. ie h264_ni_quadra_dec, h265_ni_quadra_dec, vp9_ni_quadra_dec

<enc> is the encoder codec. ie h264_ni_quadra_enc, h265_ni_quadra_enc, av1_ni_quadra_enc

Input: 1080p

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

28.2 Streaming Ladder Generation Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	5	34	92	4	246	36
AVC to HEVC	5	33	93	3	275	39
AVC to AV1	5	27	94	2	250	39
HEVC to AVC	5	35	92	4	250	38
HEVC to HEVC	5	33	92	3	280	42
HEVC to AV1	5	26	93	2	250	40
VP9 to AVC	5	40	94	4	245	35
VP9 to HEVC	5	40	95	3	275	38
VP9 to AV1	5	31	92	2	250	37

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

29.1 Encoding

29.1.1 Description

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

RGBA frame is uploaded and encoded by hardware encoder.

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

29.1.2 Command line

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

<enc> is the encoder codec. eg h264_ni_quadra_enc, h265_ni_quadra_enc, av1_ni_quadra_enc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

29.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	39	80	56
RGBA to HEVC	4k	1	1	37	79	68
RGBA to AV1	4k	1	1	41	78	63
RGBA to AVC	4k	3	0	45	87	55
RGBA to HEVC	4k	3	0	41	87	59
RGBA to AV1	4k	3	0	46	87	48
RGBA to AVC	4k	5	0	50	90	39
RGBA to HEVC	4k	5	0	44	90	39
RGBA to AV1	4k	5	0	49	90	35
RGBA to AVC	1080p	1	1	29	242	78
RGBA to HEVC	1080p	1	1	27	241	69
RGBA to AV1	1080p	1	1	31	236	89
RGBA to AVC	1080p	10	0	44	331	25
RGBA to HEVC	1080p	10	0	39	335	24
RGBA to AV1	1080p	10	0	46	333	25
RGBA to AVC	1080p	20	0	45	336	13
RGBA to HEVC	1080p	20	0	41	340	13
RGBA to AV1	1080p	20	0	46	340	14
RGBA to AVC	720p	1	1	20	374	93
RGBA to HEVC	720p	1	1	19	366	102
RGBA to AV1	720p	1	1	23	378	90
RGBA to AVC	720p	10	0	37	676	45
RGBA to HEVC	720p	10	0	36	680	45
RGBA to AV1	720p	10	0	42	660	44
RGBA to AVC	720p	20	0	40	719	25
RGBA to HEVC	720p	20	0	39	714	24
RGBA to AV1	720p	20	0	45	702	21

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	49	102	44
YUV to HEVC	4k	2	0	0	1	48	107	43
YUV to AV1	4k	2	0	0	1	49	92	40
YUV to AVC	4k	2	0	0	2	49	102	42
YUV to HEVC	4k	2	0	0	2	48	54	30
YUV to AV1	4k	2	0	0	2	48	44	26
YUV to AVC	4k	2	0	0	3	49	102	43
YUV to HEVC	4k	2	0	0	3	49	32	24
YUV to AV1	4k	2	0	0	3	51	24	22
YUV to AVC	4k	2	0	1	1	48	60	31
YUV to HEVC	4k	2	0	1	1	49	76	37
YUV to AVC	4k	2	0	1	2	49	60	30
YUV to HEVC	4k	2	0	1	2	48	34	24
YUV to AVC	4k	2	0	1	3	49	60	28
YUV to HEVC	4k	2	0	1	3	49	22	18
YUV to AVC	4k	2	4	0	1	79	100	42
YUV to HEVC	4k	2	4	0	1	74	108	42
YUV to AV1	4k	2	4	0	1	73	94	42
YUV to AVC	4k	2	4	0	2	78	100	41
YUV to HEVC	4k	2	4	0	2	62	54	28
YUV to AV1	4k	2	4	0	2	60	44	25
YUV to AVC	4k	2	4	0	3	78	100	42
YUV to HEVC	4k	2	4	0	3	56	32	24
YUV to AV1	4k	2	4	0	3	56	26	20
YUV to AVC	4k	2	4	1	1	66	60	30
YUV to HEVC	4k	2	4	1	1	67	76	33
YUV to AVC	4k	2	4	1	2	67	60	30
YUV to HEVC	4k	2	4	1	2	56	34	22
YUV to AVC	4k	2	4	1	3	67	60	32
YUV to HEVC	4k	2	4	1	3	56	22	18
YUV to AVC	4k	2	16	0	1	77	100	46
YUV to HEVC	4k	2	16	0	1	74	108	41
YUV to AV1	4k	2	16	0	1	74	94	38
YUV to AVC	4k	2	16	0	2	79	100	44
YUV to HEVC	4k	2	16	0	2	61	54	30
YUV to AV1	4k	2	16	0	2	61	44	27
YUV to AVC	4k	2	16	0	3	78	100	47
YUV to HEVC	4k	2	16	0	3	56	32	21

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	4k	2	16	0	3	56	26	19
YUV to AVC	4k	2	16	1	1	65	60	31
YUV to HEVC	4k	2	16	1	1	67	76	34
YUV to AVC	4k	2	16	1	2	66	60	29
YUV to HEVC	4k	2	16	1	2	57	34	23
YUV to AVC	4k	2	16	1	3	67	60	30
YUV to HEVC	4k	2	16	1	3	54	22	19
YUV to AVC	4k	2	40	0	1	78	98	37
YUV to HEVC	4k	2	40	0	1	74	106	43
YUV to AV1	4k	2	40	0	1	73	92	41
YUV to AVC	4k	2	40	0	2	78	100	42
YUV to HEVC	4k	2	40	0	2	62	54	30
YUV to AV1	4k	2	40	0	2	62	44	26
YUV to AVC	4k	2	40	0	3	78	100	41
YUV to HEVC	4k	2	40	0	3	57	32	19
YUV to AV1	4k	2	40	0	3	55	26	18
YUV to AVC	4k	2	40	1	1	66	60	31
YUV to HEVC	4k	2	40	1	1	66	76	36
YUV to AVC	4k	2	40	1	2	67	60	28
YUV to HEVC	4k	2	40	1	2	58	34	22
YUV to AVC	4k	2	40	1	3	67	60	30
YUV to HEVC	4k	2	40	1	3	54	22	17
YUV to AVC	1080p	10	0	0	1	99	803	26
YUV to HEVC	1080p	10	0	0	1	95	822	27
YUV to AV1	1080p	10	0	0	1	99	744	26
YUV to AVC	1080p	10	0	0	2	99	806	26
YUV to HEVC	1080p	10	0	0	2	100	439	23
YUV to AV1	1080p	10	0	0	2	100	355	19
YUV to AVC	1080p	10	0	0	3	99	804	25
YUV to HEVC	1080p	10	0	0	3	99	260	18
YUV to AV1	1080p	10	0	0	3	100	190	17
YUV to AVC	1080p	10	0	1	1	99	481	22
YUV to HEVC	1080p	10	0	1	1	99	615	24
YUV to AVC	1080p	10	0	1	2	100	479	23
YUV to HEVC	1080p	10	0	1	2	99	280	19
YUV to AVC	1080p	10	0	1	3	100	480	22
YUV to HEVC	1080p	10	0	1	3	100	179	17
YUV to AVC	1080p	10	4	0	1	99	449	21
YUV to HEVC	1080p	10	4	0	1	99	525	21

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	4	0	1	99	460	20
YUV to AVC	1080p	10	4	0	2	99	449	19
YUV to HEVC	1080p	10	4	0	2	99	330	18
YUV to AV1	1080p	10	4	0	2	99	270	17
YUV to AVC	1080p	10	4	0	3	99	449	20
YUV to HEVC	1080p	10	4	0	3	99	216	19
YUV to AV1	1080p	10	4	0	3	100	171	17
YUV to AVC	1080p	10	4	1	1	100	323	18
YUV to HEVC	1080p	10	4	1	1	99	420	20
YUV to AVC	1080p	10	4	1	2	99	324	19
YUV to HEVC	1080p	10	4	1	2	99	230	17
YUV to AVC	1080p	10	4	1	3	100	325	18
YUV to HEVC	1080p	10	4	1	3	99	156	16
YUV to AVC	1080p	10	16	0	1	99	449	21
YUV to HEVC	1080p	10	16	0	1	99	521	22
YUV to AV1	1080p	10	16	0	1	99	460	20
YUV to AVC	1080p	10	16	0	2	99	450	20
YUV to HEVC	1080p	10	16	0	2	100	330	18
YUV to AV1	1080p	10	16	0	2	99	270	16
YUV to AVC	1080p	10	16	0	3	99	449	20
YUV to HEVC	1080p	10	16	0	3	99	215	18
YUV to AV1	1080p	10	16	0	3	100	170	16
YUV to AVC	1080p	10	16	1	1	99	320	19
YUV to HEVC	1080p	10	16	1	1	99	420	21
YUV to AVC	1080p	10	16	1	2	100	322	19
YUV to HEVC	1080p	10	16	1	2	99	230	17
YUV to AVC	1080p	10	16	1	3	99	323	19
YUV to HEVC	1080p	10	16	1	3	99	155	16
YUV to AVC	1080p	10	40	0	1	100	443	21
YUV to HEVC	1080p	10	40	0	1	99	520	21
YUV to AV1	1080p	10	40	0	1	99	455	19
YUV to AVC	1080p	10	40	0	2	99	441	21
YUV to HEVC	1080p	10	40	0	2	99	320	20
YUV to AV1	1080p	10	40	0	2	99	270	17
YUV to AVC	1080p	10	40	0	3	99	441	20
YUV to HEVC	1080p	10	40	0	3	99	213	18
YUV to AV1	1080p	10	40	0	3	99	170	16
YUV to AVC	1080p	10	40	1	1	99	320	19
YUV to HEVC	1080p	10	40	1	1	99	411	20

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	40	1	2	99	320	18
YUV to HEVC	1080p	10	40	1	2	100	230	17
YUV to AVC	1080p	10	40	1	3	99	320	19
YUV to HEVC	1080p	10	40	1	3	99	153	17
YUV to AVC	720p	20	0	0	1	84	1535	15
YUV to HEVC	720p	20	0	0	1	85	1570	15
YUV to AV1	720p	20	0	0	1	88	1413	14
YUV to AVC	720p	20	0	0	2	84	1540	15
YUV to HEVC	720p	20	0	0	2	99	960	14
YUV to AV1	720p	20	0	0	2	99	780	15
YUV to AVC	720p	20	0	0	3	82	1511	15
YUV to HEVC	720p	20	0	0	3	99	580	13
YUV to AV1	720p	20	0	0	3	99	420	8
YUV to AVC	720p	20	0	1	1	99	1081	14
YUV to HEVC	720p	20	0	1	1	99	1320	15
YUV to AVC	720p	20	0	1	2	99	1080	16
YUV to HEVC	720p	20	0	1	2	99	620	8
YUV to AVC	720p	20	0	1	3	99	1080	13
YUV to HEVC	720p	20	0	1	3	99	400	7
YUV to AVC	720p	20	4	0	1	99	923	11
YUV to HEVC	720p	20	4	0	1	99	1062	14
YUV to AV1	720p	20	4	0	1	100	886	8
YUV to AVC	720p	20	4	0	2	99	924	15
YUV to HEVC	720p	20	4	0	2	99	681	12
YUV to AV1	720p	20	4	0	2	99	560	14
YUV to AVC	720p	20	4	0	3	99	924	15
YUV to HEVC	720p	20	4	0	3	99	461	7
YUV to AV1	720p	20	4	0	3	99	360	6
YUV to AVC	720p	20	4	1	1	99	680	13
YUV to HEVC	720p	20	4	1	1	99	860	14
YUV to AVC	720p	20	4	1	2	99	681	15
YUV to HEVC	720p	20	4	1	2	99	480	7
YUV to AVC	720p	20	4	1	3	99	680	15
YUV to HEVC	720p	20	4	1	3	99	340	7
YUV to AVC	720p	20	16	0	1	99	920	14
YUV to HEVC	720p	20	16	0	1	99	1060	14
YUV to AV1	720p	20	16	0	1	99	939	10
YUV to AVC	720p	20	16	0	2	99	920	15
YUV to HEVC	720p	20	16	0	2	99	680	12

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	720p	20	16	0	2	99	560	10
YUV to AVC	720p	20	16	0	3	99	920	15
YUV to HEVC	720p	20	16	0	3	99	460	15
YUV to AV1	720p	20	16	0	3	100	360	6
YUV to AVC	720p	20	16	1	1	99	680	11
YUV to HEVC	720p	20	16	1	1	99	859	10
YUV to AVC	720p	20	16	1	2	99	680	16
YUV to HEVC	720p	20	16	1	2	99	480	10
YUV to AVC	720p	20	16	1	3	99	680	11
YUV to HEVC	720p	20	16	1	3	99	340	6
YUV to AVC	720p	20	40	0	1	99	913	15
YUV to HEVC	720p	20	40	0	1	99	1047	12
YUV to AV1	720p	20	40	0	1	99	916	8
YUV to AVC	720p	20	40	0	2	99	918	11
YUV to HEVC	720p	20	40	0	2	99	680	10
YUV to AV1	720p	20	40	0	2	99	560	13
YUV to AVC	720p	20	40	0	3	99	916	12
YUV to HEVC	720p	20	40	0	3	100	460	12
YUV to AV1	720p	20	40	0	3	99	358	8
YUV to AVC	720p	20	40	1	1	99	679	13
YUV to HEVC	720p	20	40	1	1	99	843	12
YUV to AVC	720p	20	40	1	2	99	675	12
YUV to HEVC	720p	20	40	1	2	100	480	10
YUV to AVC	720p	20	40	1	3	99	678	9
YUV to HEVC	720p	20	40	1	3	100	340	8

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	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	0	0	1	19	99	450	20
YUV to HEVC	1080p	10	0	0	1	19	99	523	21
YUV to AV1	1080p	10	0	0	1	19	99	461	20
YUV to AVC	1080p	10	0	0	2	19	99	450	18
YUV to HEVC	1080p	10	0	0	2	19	99	330	19
YUV to AV1	1080p	10	0	0	2	19	99	280	18
YUV to AVC	1080p	10	0	0	3	19	99	451	20
YUV to HEVC	1080p	10	0	0	3	19	99	217	17
YUV to AV1	1080p	10	0	0	3	19	99	178	16
YUV to AVC	1080p	10	0	1	1	19	100	326	18
YUV to HEVC	1080p	10	0	1	1	19	99	420	21
YUV to AVC	1080p	10	0	1	2	19	100	327	20
YUV to HEVC	1080p	10	0	1	2	19	99	230	17
YUV to AVC	1080p	10	0	1	3	19	99	326	20
YUV to HEVC	1080p	10	0	1	3	19	100	157	16
YUV to AVC	1080p	10	4	0	1	19	99	450	20
YUV to HEVC	1080p	10	4	0	1	19	99	524	20
YUV to AV1	1080p	10	4	0	1	19	99	460	19
YUV to AVC	1080p	10	4	0	2	19	99	449	19
YUV to HEVC	1080p	10	4	0	2	19	99	330	18
YUV to AV1	1080p	10	4	0	2	19	99	270	17
YUV to AVC	1080p	10	4	0	3	19	99	450	20
YUV to HEVC	1080p	10	4	0	3	19	99	217	16
YUV to AV1	1080p	10	4	0	3	19	99	171	17
YUV to AVC	1080p	10	4	1	1	19	99	323	19
YUV to HEVC	1080p	10	4	1	1	19	99	420	19
YUV to AVC	1080p	10	4	1	2	19	100	323	18
YUV to HEVC	1080p	10	4	1	2	19	99	230	17
YUV to AVC	1080p	10	4	1	3	19	99	323	18
YUV to HEVC	1080p	10	4	1	3	19	99	155	16
YUV to AVC	1080p	10	16	0	1	19	99	448	19
YUV to HEVC	1080p	10	16	0	1	19	99	522	20
YUV to AV1	1080p	10	16	0	1	19	99	460	19
YUV to AVC	1080p	10	16	0	2	19	99	449	20
YUV to HEVC	1080p	10	16	0	2	19	99	326	18
YUV to AV1	1080p	10	16	0	2	19	100	270	16
YUV to AVC	1080p	10	16	0	3	19	99	449	20
YUV to HEVC	1080p	10	16	0	3	19	99	215	17

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	3	19	99	170	17
YUV to AVC	1080p	10	16	1	1	19	100	320	17
YUV to HEVC	1080p	10	16	1	1	19	99	419	19
YUV to AVC	1080p	10	16	1	2	19	99	321	19
YUV to HEVC	1080p	10	16	1	2	19	100	230	19
YUV to AVC	1080p	10	16	1	3	19	99	321	18
YUV to HEVC	1080p	10	16	1	3	19	100	158	16
YUV to AVC	1080p	10	40	0	1	19	99	442	20
YUV to HEVC	1080p	10	40	0	1	19	99	520	20
YUV to AV1	1080p	10	40	0	1	19	99	454	19
YUV to AVC	1080p	10	40	0	2	19	99	441	19
YUV to HEVC	1080p	10	40	0	2	19	99	321	18
YUV to AV1	1080p	10	40	0	2	19	99	270	17
YUV to AVC	1080p	10	40	0	3	19	99	441	20
YUV to HEVC	1080p	10	40	0	3	19	99	213	18
YUV to AV1	1080p	10	40	0	3	19	100	170	16
YUV to AVC	1080p	10	40	1	1	19	99	320	18
YUV to HEVC	1080p	10	40	1	1	19	99	411	21
YUV to AVC	1080p	10	40	1	2	19	99	320	19
YUV to HEVC	1080p	10	40	1	2	19	99	230	18
YUV to AVC	1080p	10	40	1	3	19	100	320	18
YUV to HEVC	1080p	10	40	1	3	19	99	153	16
YUV to AVC	1080p	10	0	0	1	23	99	450	20
YUV to HEVC	1080p	10	0	0	1	23	99	524	21
YUV to AV1	1080p	10	0	0	1	23	99	460	20
YUV to AVC	1080p	10	0	0	2	23	99	450	20
YUV to HEVC	1080p	10	0	0	2	23	100	330	19
YUV to AV1	1080p	10	0	0	2	23	99	280	17
YUV to AVC	1080p	10	0	0	3	23	99	450	19
YUV to HEVC	1080p	10	0	0	3	23	99	218	17
YUV to AV1	1080p	10	0	0	3	23	100	177	16
YUV to AVC	1080p	10	0	1	1	23	99	327	18
YUV to HEVC	1080p	10	0	1	1	23	99	420	19
YUV to AVC	1080p	10	0	1	2	23	99	324	19
YUV to HEVC	1080p	10	0	1	2	23	99	230	18
YUV to AVC	1080p	10	0	1	3	23	99	326	19
YUV to HEVC	1080p	10	0	1	3	23	99	157	16
YUV to AVC	1080p	10	4	0	1	23	99	450	20
YUV to HEVC	1080p	10	4	0	1	23	99	523	21

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	4	0	1	23	99	460	20
YUV to AVC	1080p	10	4	0	2	23	99	450	20
YUV to HEVC	1080p	10	4	0	2	23	99	330	19
YUV to AV1	1080p	10	4	0	2	23	100	270	18
YUV to AVC	1080p	10	4	0	3	23	99	450	19
YUV to HEVC	1080p	10	4	0	3	23	100	216	18
YUV to AV1	1080p	10	4	0	3	23	100	171	17
YUV to AVC	1080p	10	4	1	1	23	100	323	19
YUV to HEVC	1080p	10	4	1	1	23	99	420	21
YUV to AVC	1080p	10	4	1	2	23	99	323	18
YUV to HEVC	1080p	10	4	1	2	23	99	230	18
YUV to AVC	1080p	10	4	1	3	23	99	322	18
YUV to HEVC	1080p	10	4	1	3	23	99	156	17
YUV to AVC	1080p	10	16	0	1	23	99	448	20
YUV to HEVC	1080p	10	16	0	1	23	99	521	21
YUV to AV1	1080p	10	16	0	1	23	99	460	20
YUV to AVC	1080p	10	16	0	2	23	99	449	19
YUV to HEVC	1080p	10	16	0	2	23	100	328	18
YUV to AV1	1080p	10	16	0	2	23	100	270	17
YUV to AVC	1080p	10	16	0	3	23	99	447	20
YUV to HEVC	1080p	10	16	0	3	23	100	214	16
YUV to AV1	1080p	10	16	0	3	23	100	170	16
YUV to AVC	1080p	10	16	1	1	23	100	322	18
YUV to HEVC	1080p	10	16	1	1	23	99	419	19
YUV to AVC	1080p	10	16	1	2	23	99	320	17
YUV to HEVC	1080p	10	16	1	2	23	99	230	17
YUV to AVC	1080p	10	16	1	3	23	99	320	18
YUV to HEVC	1080p	10	16	1	3	23	99	157	17
YUV to AVC	1080p	10	40	0	1	23	99	442	20
YUV to HEVC	1080p	10	40	0	1	23	99	520	21
YUV to AV1	1080p	10	40	0	1	23	99	453	19
YUV to AVC	1080p	10	40	0	2	23	99	441	20
YUV to HEVC	1080p	10	40	0	2	23	99	321	18
YUV to AV1	1080p	10	40	0	2	23	99	270	16
YUV to AVC	1080p	10	40	0	3	23	99	443	20
YUV to HEVC	1080p	10	40	0	3	23	100	212	18
YUV to AV1	1080p	10	40	0	3	23	98	170	17
YUV to AVC	1080p	10	40	1	1	23	99	320	19
YUV to HEVC	1080p	10	40	1	1	23	99	411	19

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	40	1	2	23	100	320	18
YUV to HEVC	1080p	10	40	1	2	23	100	230	17
YUV to AVC	1080p	10	40	1	3	23	99	320	19
YUV to HEVC	1080p	10	40	1	3	23	99	153	16
YUV to AVC	1080p	10	0	0	1	27	99	450	20
YUV to HEVC	1080p	10	0	0	1	27	99	525	21
YUV to AV1	1080p	10	0	0	1	27	99	460	20
YUV to AVC	1080p	10	0	0	2	27	99	451	20
YUV to HEVC	1080p	10	0	0	2	27	99	330	19
YUV to AV1	1080p	10	0	0	2	27	99	279	18
YUV to AVC	1080p	10	0	0	3	27	99	451	20
YUV to HEVC	1080p	10	0	0	3	27	100	219	17
YUV to AV1	1080p	10	0	0	3	27	99	176	16
YUV to AVC	1080p	10	0	1	1	27	99	324	19
YUV to HEVC	1080p	10	0	1	1	27	99	420	20
YUV to AVC	1080p	10	0	1	2	27	99	326	20
YUV to HEVC	1080p	10	0	1	2	27	99	230	17
YUV to AVC	1080p	10	0	1	3	27	99	324	18
YUV to HEVC	1080p	10	0	1	3	27	99	157	16
YUV to AVC	1080p	10	4	0	1	27	99	451	19
YUV to HEVC	1080p	10	4	0	1	27	99	523	22
YUV to AV1	1080p	10	4	0	1	27	99	460	19
YUV to AVC	1080p	10	4	0	2	27	99	449	20
YUV to HEVC	1080p	10	4	0	2	27	99	330	17
YUV to AV1	1080p	10	4	0	2	27	99	270	17
YUV to AVC	1080p	10	4	0	3	27	99	450	20
YUV to HEVC	1080p	10	4	0	3	27	99	216	18
YUV to AV1	1080p	10	4	0	3	27	99	172	17
YUV to AVC	1080p	10	4	1	1	27	99	324	18
YUV to HEVC	1080p	10	4	1	1	27	99	420	20
YUV to AVC	1080p	10	4	1	2	27	99	322	19
YUV to HEVC	1080p	10	4	1	2	27	100	230	18
YUV to AVC	1080p	10	4	1	3	27	99	325	19
YUV to HEVC	1080p	10	4	1	3	27	100	157	17
YUV to AVC	1080p	10	16	0	1	27	99	448	20
YUV to HEVC	1080p	10	16	0	1	27	99	520	21
YUV to AV1	1080p	10	16	0	1	27	99	460	19
YUV to AVC	1080p	10	16	0	2	27	99	446	22
YUV to HEVC	1080p	10	16	0	2	27	99	328	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	2	27	99	270	17
YUV to AVC	1080p	10	16	0	3	27	99	448	19
YUV to HEVC	1080p	10	16	0	3	27	100	216	17
YUV to AV1	1080p	10	16	0	3	27	100	171	16
YUV to AVC	1080p	10	16	1	1	27	99	321	19
YUV to HEVC	1080p	10	16	1	1	27	99	419	20
YUV to AVC	1080p	10	16	1	2	27	99	321	19
YUV to HEVC	1080p	10	16	1	2	27	100	230	17
YUV to AVC	1080p	10	16	1	3	27	99	320	20
YUV to HEVC	1080p	10	16	1	3	27	100	156	16
YUV to AVC	1080p	10	40	0	1	27	99	444	20
YUV to HEVC	1080p	10	40	0	1	27	99	520	21
YUV to AV1	1080p	10	40	0	1	27	99	455	18
YUV to AVC	1080p	10	40	0	2	27	99	440	19
YUV to HEVC	1080p	10	40	0	2	27	99	320	19
YUV to AV1	1080p	10	40	0	2	27	99	270	19
YUV to AVC	1080p	10	40	0	3	27	99	441	21
YUV to HEVC	1080p	10	40	0	3	27	99	211	18
YUV to AV1	1080p	10	40	0	3	27	100	170	17
YUV to AVC	1080p	10	40	1	1	27	99	320	18
YUV to HEVC	1080p	10	40	1	1	27	99	412	21
YUV to AVC	1080p	10	40	1	2	27	99	320	17
YUV to HEVC	1080p	10	40	1	2	27	99	230	17
YUV to AVC	1080p	10	40	1	3	27	99	320	18
YUV to HEVC	1080p	10	40	1	3	27	100	153	16

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

32.1 Transcoding

32.1.1 Description

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

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

32.1.2 Command line

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

<dec> is the decoder codec. ie h264_ni_quadra_dec, h265_ni_quadra_dec, vp9_ni_quadra_dec

<enc> is the encoder codec. ie h264_ni_quadra_enc, h265_ni_quadra_enc, av1_ni_quadra_enc

Input Video: 1080p

Input Image: 128x128

32.2 Inplace Overlay Performance Results

TYPE	JOB5	FPS	CPU	DEC_LOAD	ENC_LOAD	SCALER_LOAD
AVC to AVC	1	184	61	13	20	3
AVC to HEVC	1	195	60	14	20	3
AVC to AV1	1	168	57	12	20	2
HEVC to AVC	1	182	63	14	20	3
HEVC to HEVC	1	192	70	14	20	3
HEVC to AV1	1	166	65	12	20	2
VP9 to AVC	1	183	58	18	20	3
VP9 to HEVC	1	194	57	19	20	3
VP9 to AV1	1	167	55	17	20	2
AVC to AVC	8	523	25	70	92	18
AVC to HEVC	8	592	26	77	93	20
AVC to AV1	8	592	26	72	90	20
HEVC to AVC	8	547	27	65	92	18
HEVC to HEVC	8	616	29	72	93	21
HEVC to AV1	8	609	31	67	91	20
VP9 to AVC	8	527	24	74	92	17
VP9 to HEVC	8	596	26	82	93	19
VP9 to AV1	8	595	26	81	93	19
AVC to AVC	16	528	11	72	93	19
AVC to HEVC	16	592	11	79	94	21
AVC to AV1	16	592	12	77	95	21
HEVC to AVC	16	549	12	67	94	20
HEVC to HEVC	16	624	13	74	94	22
HEVC to AV1	16	611	13	70	94	21
VP9 to AVC	16	528	10	76	94	18
VP9 to HEVC	16	608	13	84	94	20
VP9 to AV1	16	594	12	82	93	20

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

33.1 Encoding

33.1.1 Description

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

YUV frame is encoded by hardware encoder.

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

33.1.2 Command Line

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

<enc> is the encoder codec. eg h264_ni_quadra_enc, h265_ni_quadra_enc, av1_ni_quadra_enc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

33.2 FFmpeg Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	4k	1	0	8	1	0	58	179	60
YUV to HEVC	4k	1	0	8	1	0	54	185	58
YUV to AV1	4k	1	0	8	1	0	61	180	54
YUV to AVC	4k	3	0	8	0	0	72	198	47
YUV to HEVC	4k	3	0	8	0	0	67	222	58
YUV to AV1	4k	3	0	8	0	0	70	210	51
YUV to AVC	4k	5	0	8	0	0	94	206	36
YUV to HEVC	4k	5	0	8	0	0	69	226	48
YUV to AV1	4k	5	0	8	0	0	77	224	43
YUV to AVC	4k	1	0	10	0	0	35	104	52
YUV to HEVC	4k	1	0	10	0	0	31	106	52
YUV to AV1	4k	1	0	10	0	0	35	105	54
YUV to AVC	4k	3	0	10	0	0	48	111	56
YUV to HEVC	4k	3	0	10	0	0	35	116	61
YUV to AV1	4k	3	0	10	0	0	39	113	62
YUV to AVC	1080p	1	0	8	1	0	33	440	78
YUV to HEVC	1080p	1	0	8	1	0	31	442	74
YUV to AV1	1080p	1	0	8	1	0	33	397	80
YUV to AVC	1080p	20	0	8	0	0	73	802	14
YUV to HEVC	1080p	20	0	8	0	0	65	843	14
YUV to AV1	1080p	20	0	8	0	0	70	804	14
YUV to AVC	1080p	1	0	10	0	0	24	308	65
YUV to HEVC	1080p	1	0	10	0	0	22	307	58
YUV to AV1	1080p	1	0	10	0	0	24	284	68
YUV to AVC	1080p	20	0	10	0	0	38	461	14
YUV to HEVC	1080p	20	0	10	0	0	34	460	13
YUV to AV1	1080p	20	0	10	0	0	39	454	14
YUV to AVC	720p	1	0	8	1	0	19	558	64
YUV to HEVC	720p	1	0	8	1	0	20	551	57
YUV to AV1	720p	1	0	8	1	0	23	506	65
YUV to AVC	720p	40	0	8	0	0	82	1561	8
YUV to HEVC	720p	40	0	8	0	0	80	1651	8
YUV to AV1	720p	40	0	8	0	0	82	1441	7

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	36	123	50
YUV to HEVC	4k	1	0	8	1	0	35	124	49
YUV to AV1	4k	1	0	8	1	0	37	111	53
YUV to AVC	4k	3	0	8	0	0	67	172	38
YUV to HEVC	4k	3	0	8	0	0	50	169	39
YUV to AV1	4k	3	0	8	0	0	53	156	37
YUV to AVC	4k	5	0	8	0	0	89	193	30
YUV to HEVC	4k	5	0	8	0	0	61	210	35
YUV to AV1	4k	5	0	8	0	0	69	192	34
YUV to AVC	4k	1	0	10	1	0	20	66	50
YUV to HEVC	4k	1	0	10	1	0	19	66	50
YUV to AV1	4k	1	0	10	1	0	21	63	51
YUV to AVC	4k	3	0	10	0	0	28	89	39
YUV to HEVC	4k	3	0	10	0	0	26	90	40
YUV to AV1	4k	3	0	10	0	0	28	83	39
YUV to AVC	1080p	1	0	8	1	0	26	359	45
YUV to HEVC	1080p	1	0	8	1	0	23	315	47
YUV to AV1	1080p	1	0	8	1	0	25	300	41
YUV to AVC	1080p	20	0	8	0	0	73	828	11
YUV to HEVC	1080p	20	0	8	0	0	66	860	12
YUV to AV1	1080p	20	0	8	0	0	70	816	11
YUV to AVC	1080p	1	0	10	1	0	16	214	49
YUV to HEVC	1080p	1	0	10	1	0	14	205	51
YUV to AV1	1080p	1	0	10	1	0	16	191	47
YUV to AVC	1080p	20	0	10	0	0	37	463	12
YUV to HEVC	1080p	20	0	10	0	0	34	466	12
YUV to AV1	1080p	20	0	10	0	0	38	433	13
YUV to AVC	720p	1	0	8	1	0	18	544	37
YUV to HEVC	720p	1	0	8	1	0	18	520	42
YUV to AV1	720p	1	0	8	1	0	18	417	46
YUV to AVC	720p	40	0	8	0	0	84	1709	6
YUV to HEVC	720p	40	0	8	0	0	80	1791	6
YUV to AV1	720p	40	0	8	0	0	85	1561	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	20.1	17.14	0.09
YUV to HEVC	4k	1	19.22	21.39	18	0.27
YUV to AV1	4k	1	24.21	28.22	18.64	0.7
YUV to AVC	4k	3	19.05	21.55	17.23	0.73
YUV to HEVC	4k	3	19.52	24.54	17.98	0.5
YUV to AV1	4k	3	25.42	31.24	19.05	1.57
YUV to AVC	4k	5	19.73	24.49	17.63	0.98
YUV to HEVC	4k	5	20.68	25.46	18.02	2.13
YUV to AV1	4k	5	27.59	47.47	19.3	4.13
YUV to AVC	1080p	1	5.81	6.81	5.09	0.08
YUV to HEVC	1080p	1	6.03	6.83	5.53	0.04
YUV to AV1	1080p	1	7.71	8.53	5.96	0.07
YUV to AVC	1080p	20	8.05	13.21	5.32	2.11
YUV to HEVC	1080p	20	9.09	15.41	5.93	1.37
YUV to AV1	1080p	20	11.89	22.08	6.38	5.74
YUV to AVC	720p	1	3.55	5.57	3.02	0.06
YUV to HEVC	720p	1	3.71	5.82	3.2	0.28
YUV to AV1	720p	1	4.51	5.17	3.81	0.05
YUV to AVC	720p	40	4.65	14.28	3.14	0.43
YUV to HEVC	720p	40	5.22	10.85	3.35	0.87
YUV to AV1	720p	40	7.63	15.15	4.53	1.98

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

36.1 Encoding

36.1.1 Description

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

RGBA frame is uploaded and encoded by hardware encoder.

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

36.1.2 Command line

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

<enc> is the encoder codec. eg h264_ni_quadra_enc, h265_ni_quadra_enc, av1_ni_quadra_enc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

36.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	25	80	77
RGBA to HEVC	4k	1	1	23	81	70
RGBA to AV1	4k	1	1	26	79	68
RGBA to AVC	4k	3	0	36	86	50
RGBA to HEVC	4k	3	0	32	85	52
RGBA to AV1	4k	3	0	32	87	51
RGBA to AVC	4k	5	0	46	89	43
RGBA to HEVC	4k	5	0	40	90	36
RGBA to AV1	4k	5	0	38	88	34
RGBA to AVC	1080p	1	1	19	243	76
RGBA to HEVC	1080p	1	1	18	244	59
RGBA to AV1	1080p	1	1	20	237	73
RGBA to AVC	1080p	10	0	34	326	25
RGBA to HEVC	1080p	10	0	29	326	25
RGBA to AV1	1080p	10	0	31	327	25
RGBA to AVC	1080p	20	0	36	320	13
RGBA to HEVC	1080p	20	0	32	335	13
RGBA to AV1	1080p	20	0	34	334	14
RGBA to AVC	720p	1	1	16	362	100
RGBA to HEVC	720p	1	1	13	333	108
RGBA to AV1	720p	1	1	21	368	101
RGBA to AVC	720p	10	0	34	679	45
RGBA to HEVC	720p	10	0	32	685	46
RGBA to AV1	720p	10	0	39	677	45
RGBA to AVC	720p	20	0	35	707	22
RGBA to HEVC	720p	20	0	33	718	20
RGBA to AV1	720p	20	0	42	701	21

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	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	0	0	1	48	152	55
YUV to HEVC	4k	2	0	0	1	48	164	46
YUV to AV1	4k	2	0	0	1	48	142	51
YUV to AVC	4k	2	0	0	2	48	152	59
YUV to HEVC	4k	2	0	0	2	48	84	38
YUV to AV1	4k	2	0	0	2	49	70	38
YUV to AVC	4k	2	0	0	3	48	152	48
YUV to HEVC	4k	2	0	0	3	49	50	25
YUV to AV1	4k	2	0	0	3	50	38	22
YUV to AVC	4k	2	0	1	1	49	94	40
YUV to HEVC	4k	2	0	1	1	49	118	50
YUV to AVC	4k	2	0	1	2	48	94	41
YUV to HEVC	4k	2	0	1	2	49	54	28
YUV to AVC	4k	2	0	1	3	48	94	40
YUV to HEVC	4k	2	0	1	3	49	34	21
YUV to AVC	4k	2	4	0	1	74	134	45
YUV to HEVC	4k	2	4	0	1	70	162	67
YUV to AV1	4k	2	4	0	1	72	142	56
YUV to AVC	4k	2	4	0	2	74	130	53
YUV to HEVC	4k	2	4	0	2	61	84	41
YUV to AV1	4k	2	4	0	2	60	70	30
YUV to AVC	4k	2	4	0	3	74	132	51
YUV to HEVC	4k	2	4	0	3	57	50	26
YUV to AV1	4k	2	4	0	3	56	40	24
YUV to AVC	4k	2	4	1	1	67	92	41
YUV to HEVC	4k	2	4	1	1	64	118	54
YUV to AVC	4k	2	4	1	2	68	92	40
YUV to HEVC	4k	2	4	1	2	58	54	27
YUV to AVC	4k	2	4	1	3	67	92	39
YUV to HEVC	4k	2	4	1	3	53	34	24
YUV to AVC	4k	2	20	0	1	72	130	51
YUV to HEVC	4k	2	20	0	1	73	166	48
YUV to AV1	4k	2	20	0	1	70	142	58
YUV to AVC	4k	2	20	0	2	73	130	50

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	4k	2	20	0	2	61	84	38
YUV to AV1	4k	2	20	0	2	60	70	33
YUV to AVC	4k	2	20	0	3	74	130	50
YUV to HEVC	4k	2	20	0	3	57	50	27
YUV to AV1	4k	2	20	0	3	56	40	23
YUV to AVC	4k	2	20	1	1	66	92	38
YUV to HEVC	4k	2	20	1	1	66	119	46
YUV to AVC	4k	2	20	1	2	66	92	42
YUV to HEVC	4k	2	20	1	2	57	54	28
YUV to AVC	4k	2	20	1	3	67	92	39
YUV to HEVC	4k	2	20	1	3	53	34	24
YUV to AVC	4k	2	40	0	1	73	130	50
YUV to HEVC	4k	2	40	0	1	73	162	54
YUV to AV1	4k	2	40	0	1	72	140	45
YUV to AVC	4k	2	40	0	2	73	130	46
YUV to HEVC	4k	2	40	0	2	61	84	35
YUV to AV1	4k	2	40	0	2	60	70	31
YUV to AVC	4k	2	40	0	3	73	130	48
YUV to HEVC	4k	2	40	0	3	57	50	27
YUV to AV1	4k	2	40	0	3	55	40	26
YUV to AVC	4k	2	40	1	1	67	92	40
YUV to HEVC	4k	2	40	1	1	66	118	48
YUV to AVC	4k	2	40	1	2	66	92	40
YUV to HEVC	4k	2	40	1	2	57	54	27
YUV to AVC	4k	2	40	1	3	66	92	38
YUV to HEVC	4k	2	40	1	3	55	34	25
YUV to AVC	1080p	10	0	0	1	65	773	27
YUV to HEVC	1080p	10	0	0	1	60	804	28
YUV to AV1	1080p	10	0	0	1	66	762	25
YUV to AVC	1080p	10	0	0	2	65	771	26
YUV to HEVC	1080p	10	0	0	2	99	686	25
YUV to AV1	1080p	10	0	0	2	100	557	22
YUV to AVC	1080p	10	0	0	3	66	775	26
YUV to HEVC	1080p	10	0	0	3	100	406	19

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	0	0	3	100	298	18
YUV to AVC	1080p	10	0	1	1	99	745	26
YUV to HEVC	1080p	10	0	1	1	86	821	27
YUV to AVC	1080p	10	0	1	2	99	748	25
YUV to HEVC	1080p	10	0	1	2	100	439	21
YUV to AVC	1080p	10	0	1	3	100	748	26
YUV to HEVC	1080p	10	0	1	3	99	279	20
YUV to AVC	1080p	10	4	0	1	99	650	21
YUV to HEVC	1080p	10	4	0	1	87	702	22
YUV to AV1	1080p	10	4	0	1	93	677	21
YUV to AVC	1080p	10	4	0	2	100	648	21
YUV to HEVC	1080p	10	4	0	2	99	510	20
YUV to AV1	1080p	10	4	0	2	99	430	19
YUV to AVC	1080p	10	4	0	3	99	646	21
YUV to HEVC	1080p	10	4	0	3	99	340	20
YUV to AV1	1080p	10	4	0	3	100	270	18
YUV to AVC	1080p	10	4	1	1	100	510	20
YUV to HEVC	1080p	10	4	1	1	100	653	22
YUV to AVC	1080p	10	4	1	2	100	510	20
YUV to HEVC	1080p	10	4	1	2	99	360	18
YUV to AVC	1080p	10	4	1	3	99	510	21
YUV to HEVC	1080p	10	4	1	3	100	244	17
YUV to AVC	1080p	10	20	0	1	99	640	21
YUV to HEVC	1080p	10	20	0	1	89	712	22
YUV to AV1	1080p	10	20	0	1	94	661	20
YUV to AVC	1080p	10	20	0	2	99	641	19
YUV to HEVC	1080p	10	20	0	2	100	510	22
YUV to AV1	1080p	10	20	0	2	100	430	20
YUV to AVC	1080p	10	20	0	3	99	641	20
YUV to HEVC	1080p	10	20	0	3	100	340	18
YUV to AV1	1080p	10	20	0	3	100	270	17
YUV to AVC	1080p	10	20	1	1	100	508	21
YUV to HEVC	1080p	10	20	1	1	100	653	22
YUV to AVC	1080p	10	20	1	2	100	508	22

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	20	1	2	100	360	19
YUV to AVC	1080p	10	20	1	3	100	507	20
YUV to HEVC	1080p	10	20	1	3	100	242	18
YUV to AVC	1080p	10	40	0	1	100	630	20
YUV to HEVC	1080p	10	40	0	1	89	699	21
YUV to AV1	1080p	10	40	0	1	91	640	20
YUV to AVC	1080p	10	40	0	2	99	630	21
YUV to HEVC	1080p	10	40	0	2	99	509	21
YUV to AV1	1080p	10	40	0	2	99	420	19
YUV to AVC	1080p	10	40	0	3	99	630	20
YUV to HEVC	1080p	10	40	0	3	100	335	18
YUV to AV1	1080p	10	40	0	3	100	268	18
YUV to AVC	1080p	10	40	1	1	99	501	20
YUV to HEVC	1080p	10	40	1	1	99	647	22
YUV to AVC	1080p	10	40	1	2	99	500	20
YUV to HEVC	1080p	10	40	1	2	99	359	19
YUV to AVC	1080p	10	40	1	3	99	500	21
YUV to HEVC	1080p	10	40	1	3	100	242	15
YUV to AVC	720p	20	0	0	1	74	1474	14
YUV to HEVC	720p	20	0	0	1	72	1551	15
YUV to AV1	720p	20	0	0	1	79	1401	14
YUV to AVC	720p	20	0	0	2	74	1484	14
YUV to HEVC	720p	20	0	0	2	99	1500	16
YUV to AV1	720p	20	0	0	2	99	1205	15
YUV to AVC	720p	20	0	0	3	73	1464	13
YUV to HEVC	720p	20	0	0	3	99	903	13
YUV to AV1	720p	20	0	0	3	100	660	7
YUV to AVC	720p	20	0	1	1	92	1553	15
YUV to HEVC	720p	20	0	1	1	77	1582	15
YUV to AVC	720p	20	0	1	2	91	1554	15
YUV to HEVC	720p	20	0	1	2	100	960	11
YUV to AVC	720p	20	0	1	3	90	1539	15
YUV to HEVC	720p	20	0	1	3	100	620	7
YUV to AVC	720p	20	4	0	1	99	1100	9

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	720p	20	4	0	1	98	1103	10
YUV to AV1	720p	20	4	0	1	100	880	7
YUV to AVC	720p	20	4	0	2	97	1090	10
YUV to HEVC	720p	20	4	0	2	99	1080	13
YUV to AV1	720p	20	4	0	2	98	871	8
YUV to AVC	720p	20	4	0	3	99	1096	10
YUV to HEVC	720p	20	4	0	3	99	740	12
YUV to AV1	720p	20	4	0	3	99	560	14
YUV to AVC	720p	20	4	1	1	99	1068	14
YUV to HEVC	720p	20	4	1	1	99	1119	10
YUV to AVC	720p	20	4	1	2	99	1070	13
YUV to HEVC	720p	20	4	1	2	99	760	11
YUV to AVC	720p	20	4	1	3	99	1072	14
YUV to HEVC	720p	20	4	1	3	99	536	13
YUV to AVC	720p	20	20	0	1	99	1073	10
YUV to HEVC	720p	20	20	0	1	99	1110	10
YUV to AV1	720p	20	20	0	1	100	909	8
YUV to AVC	720p	20	20	0	2	99	1075	10
YUV to HEVC	720p	20	20	0	2	99	1079	13
YUV to AV1	720p	20	20	0	2	99	880	11
YUV to AVC	720p	20	20	0	3	99	1072	9
YUV to HEVC	720p	20	20	0	3	100	727	12
YUV to AV1	720p	20	20	0	3	99	560	8
YUV to AVC	720p	20	20	1	1	99	1061	14
YUV to HEVC	720p	20	20	1	1	99	1111	10
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	13
YUV to HEVC	720p	20	20	1	3	99	527	10
YUV to AVC	720p	20	40	0	1	100	1054	10
YUV to HEVC	720p	20	40	0	1	99	1084	11
YUV to AV1	720p	20	40	0	1	100	899	9
YUV to AVC	720p	20	40	0	2	99	1047	9
YUV to HEVC	720p	20	40	0	2	99	1060	13

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	720p	20	40	0	2	99	877	11
YUV to AVC	720p	20	40	0	3	99	1055	10
YUV to HEVC	720p	20	40	0	3	99	720	15
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	99	1090	11
YUV to AVC	720p	20	40	1	2	99	1059	13
YUV to HEVC	720p	20	40	1	2	99	760	15
YUV to AVC	720p	20	40	1	3	99	1060	13
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	99	683	22
YUV to HEVC	1080p	10	0	0	1	19	93	753	23
YUV to AV1	1080p	10	0	0	1	19	97	700	21
YUV to AVC	1080p	10	0	0	2	19	99	685	23
YUV to HEVC	1080p	10	0	0	2	19	99	512	21
YUV to AV1	1080p	10	0	0	2	19	100	430	19
YUV to AVC	1080p	10	0	0	3	19	99	681	22
YUV to HEVC	1080p	10	0	0	3	19	100	340	18
YUV to AV1	1080p	10	0	0	3	19	100	279	18
YUV to AVC	1080p	10	0	1	1	19	100	511	20
YUV to HEVC	1080p	10	0	1	1	19	99	659	24
YUV to AVC	1080p	10	0	1	2	19	99	511	21
YUV to HEVC	1080p	10	0	1	2	19	100	360	19
YUV to AVC	1080p	10	0	1	3	19	100	510	22
YUV to HEVC	1080p	10	0	1	3	19	99	246	18
YUV to AVC	1080p	10	4	0	1	19	99	645	21
YUV to HEVC	1080p	10	4	0	1	19	84	683	22
YUV to AV1	1080p	10	4	0	1	19	87	622	19
YUV to AVC	1080p	10	4	0	2	19	99	644	21
YUV to HEVC	1080p	10	4	0	2	19	99	512	20
YUV to AV1	1080p	10	4	0	2	19	100	430	19
YUV to AVC	1080p	10	4	0	3	19	99	643	20
YUV to HEVC	1080p	10	4	0	3	19	99	340	18
YUV to AV1	1080p	10	4	0	3	19	99	271	18
YUV to AVC	1080p	10	4	1	1	19	100	510	21
YUV to HEVC	1080p	10	4	1	1	19	99	657	22
YUV to AVC	1080p	10	4	1	2	19	100	510	20
YUV to HEVC	1080p	10	4	1	2	19	99	361	19
YUV to AVC	1080p	10	4	1	3	19	99	510	20
YUV to HEVC	1080p	10	4	1	3	19	100	246	16
YUV to AVC	1080p	10	16	0	1	19	92	599	18
YUV to HEVC	1080p	10	16	0	1	19	84	668	20
YUV to AV1	1080p	10	16	0	1	19	85	611	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	0	2	19	94	601	20
YUV to HEVC	1080p	10	16	0	2	19	99	510	19
YUV to AV1	1080p	10	16	0	2	19	100	429	17
YUV to AVC	1080p	10	16	0	3	19	93	594	19
YUV to HEVC	1080p	10	16	0	3	19	100	340	20
YUV to AV1	1080p	10	16	0	3	19	99	270	19
YUV to AVC	1080p	10	16	1	1	19	100	508	20
YUV to HEVC	1080p	10	16	1	1	19	99	651	22
YUV to AVC	1080p	10	16	1	2	19	100	510	20
YUV to HEVC	1080p	10	16	1	2	19	100	360	19
YUV to AVC	1080p	10	16	1	3	19	99	507	20
YUV to HEVC	1080p	10	16	1	3	19	100	243	17
YUV to AVC	1080p	10	40	0	1	19	91	581	18
YUV to HEVC	1080p	10	40	0	1	19	79	641	20
YUV to AV1	1080p	10	40	0	1	19	85	581	18
YUV to AVC	1080p	10	40	0	2	19	91	585	19
YUV to HEVC	1080p	10	40	0	2	19	99	509	20
YUV to AV1	1080p	10	40	0	2	19	99	420	19
YUV to AVC	1080p	10	40	0	3	19	90	582	18
YUV to HEVC	1080p	10	40	0	3	19	99	335	18
YUV to AV1	1080p	10	40	0	3	19	100	270	17
YUV to AVC	1080p	10	40	1	1	19	99	500	21
YUV to HEVC	1080p	10	40	1	1	19	99	647	21
YUV to AVC	1080p	10	40	1	2	19	99	500	20
YUV to HEVC	1080p	10	40	1	2	19	99	357	19
YUV to AVC	1080p	10	40	1	3	19	99	501	20
YUV to HEVC	1080p	10	40	1	3	19	100	241	18
YUV to AVC	1080p	10	0	0	1	23	99	682	21
YUV to HEVC	1080p	10	0	0	1	23	93	759	24
YUV to AV1	1080p	10	0	0	1	23	97	702	21
YUV to AVC	1080p	10	0	0	2	23	99	684	22
YUV to HEVC	1080p	10	0	0	2	23	100	511	21
YUV to AV1	1080p	10	0	0	2	23	100	430	18
YUV to AVC	1080p	10	0	0	3	23	99	683	23

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	23	100	340	20
YUV to AV1	1080p	10	0	0	3	23	99	278	17
YUV to AVC	1080p	10	0	1	1	23	100	510	20
YUV to HEVC	1080p	10	0	1	1	23	99	657	22
YUV to AVC	1080p	10	0	1	2	23	99	510	20
YUV to HEVC	1080p	10	0	1	2	23	100	360	19
YUV to AVC	1080p	10	0	1	3	23	99	511	21
YUV to HEVC	1080p	10	0	1	3	23	99	244	20
YUV to AVC	1080p	10	4	0	1	23	99	647	20
YUV to HEVC	1080p	10	4	0	1	23	86	694	21
YUV to AV1	1080p	10	4	0	1	23	88	639	19
YUV to AVC	1080p	10	4	0	2	23	99	649	20
YUV to HEVC	1080p	10	4	0	2	23	99	511	20
YUV to AV1	1080p	10	4	0	2	23	99	430	19
YUV to AVC	1080p	10	4	0	3	23	99	647	21
YUV to HEVC	1080p	10	4	0	3	23	100	340	18
YUV to AV1	1080p	10	4	0	3	23	100	270	17
YUV to AVC	1080p	10	4	1	1	23	100	510	21
YUV to HEVC	1080p	10	4	1	1	23	99	657	22
YUV to AVC	1080p	10	4	1	2	23	99	510	20
YUV to HEVC	1080p	10	4	1	2	23	99	360	20
YUV to AVC	1080p	10	4	1	3	23	99	509	20
YUV to HEVC	1080p	10	4	1	3	23	100	244	18
YUV to AVC	1080p	10	16	0	1	23	97	620	19
YUV to HEVC	1080p	10	16	0	1	23	84	679	21
YUV to AV1	1080p	10	16	0	1	23	87	623	19
YUV to AVC	1080p	10	16	0	2	23	96	619	19
YUV to HEVC	1080p	10	16	0	2	23	99	510	21
YUV to AV1	1080p	10	16	0	2	23	99	430	19
YUV to AVC	1080p	10	16	0	3	23	96	621	19
YUV to HEVC	1080p	10	16	0	3	23	99	339	17
YUV to AV1	1080p	10	16	0	3	23	100	270	17
YUV to AVC	1080p	10	16	1	1	23	99	508	21
YUV to HEVC	1080p	10	16	1	1	23	99	652	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	1	2	23	99	507	21
YUV to HEVC	1080p	10	16	1	2	23	100	360	18
YUV to AVC	1080p	10	16	1	3	23	100	504	21
YUV to HEVC	1080p	10	16	1	3	23	100	242	18
YUV to AVC	1080p	10	40	0	1	23	96	612	19
YUV to HEVC	1080p	10	40	0	1	23	84	661	20
YUV to AV1	1080p	10	40	0	1	23	88	599	19
YUV to AVC	1080p	10	40	0	2	23	96	607	19
YUV to HEVC	1080p	10	40	0	2	23	99	508	21
YUV to AV1	1080p	10	40	0	2	23	100	420	19
YUV to AVC	1080p	10	40	0	3	23	96	605	19
YUV to HEVC	1080p	10	40	0	3	23	100	334	19
YUV to AV1	1080p	10	40	0	3	23	99	269	18
YUV to AVC	1080p	10	40	1	1	23	99	500	20
YUV to HEVC	1080p	10	40	1	1	23	99	646	22
YUV to AVC	1080p	10	40	1	2	23	99	500	22
YUV to HEVC	1080p	10	40	1	2	23	99	357	20
YUV to AVC	1080p	10	40	1	3	23	99	500	20
YUV to HEVC	1080p	10	40	1	3	23	99	241	18
YUV to AVC	1080p	10	0	0	1	27	99	682	22
YUV to HEVC	1080p	10	0	0	1	27	94	750	23
YUV to AV1	1080p	10	0	0	1	27	96	700	22
YUV to AVC	1080p	10	0	0	2	27	99	683	23
YUV to HEVC	1080p	10	0	0	2	27	99	511	21
YUV to AV1	1080p	10	0	0	2	27	99	430	19
YUV to AVC	1080p	10	0	0	3	27	100	685	22
YUV to HEVC	1080p	10	0	0	3	27	100	339	20
YUV to AV1	1080p	10	0	0	3	27	100	277	19
YUV to AVC	1080p	10	0	1	1	27	100	510	20
YUV to HEVC	1080p	10	0	1	1	27	99	658	23
YUV to AVC	1080p	10	0	1	2	27	99	511	21
YUV to HEVC	1080p	10	0	1	2	27	99	360	20
YUV to AVC	1080p	10	0	1	3	27	100	509	21
YUV to HEVC	1080p	10	0	1	3	27	100	245	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	4	0	1	27	99	648	21
YUV to HEVC	1080p	10	4	0	1	27	88	706	22
YUV to AV1	1080p	10	4	0	1	27	90	650	20
YUV to AVC	1080p	10	4	0	2	27	100	646	21
YUV to HEVC	1080p	10	4	0	2	27	100	514	20
YUV to AV1	1080p	10	4	0	2	27	100	430	19
YUV to AVC	1080p	10	4	0	3	27	99	648	21
YUV to HEVC	1080p	10	4	0	3	27	100	340	19
YUV to AV1	1080p	10	4	0	3	27	100	271	17
YUV to AVC	1080p	10	4	1	1	27	99	510	20
YUV to HEVC	1080p	10	4	1	1	27	99	657	22
YUV to AVC	1080p	10	4	1	2	27	99	510	20
YUV to HEVC	1080p	10	4	1	2	27	99	361	18
YUV to AVC	1080p	10	4	1	3	27	100	510	20
YUV to HEVC	1080p	10	4	1	3	27	99	246	17
YUV to AVC	1080p	10	16	0	1	27	99	640	20
YUV to HEVC	1080p	10	16	0	1	27	86	690	22
YUV to AV1	1080p	10	16	0	1	27	88	631	19
YUV to AVC	1080p	10	16	0	2	27	100	640	21
YUV to HEVC	1080p	10	16	0	2	27	99	510	20
YUV to AV1	1080p	10	16	0	2	27	99	429	19
YUV to AVC	1080p	10	16	0	3	27	99	640	20
YUV to HEVC	1080p	10	16	0	3	27	100	338	18
YUV to AV1	1080p	10	16	0	3	27	99	270	17
YUV to AVC	1080p	10	16	1	1	27	99	508	21
YUV to HEVC	1080p	10	16	1	1	27	100	651	21
YUV to AVC	1080p	10	16	1	2	27	99	507	20
YUV to HEVC	1080p	10	16	1	2	27	100	360	18
YUV to AVC	1080p	10	16	1	3	27	100	509	20
YUV to HEVC	1080p	10	16	1	3	27	100	244	17
YUV to AVC	1080p	10	40	0	1	27	98	625	20
YUV to HEVC	1080p	10	40	0	1	27	85	669	21
YUV to AV1	1080p	10	40	0	1	27	86	608	18
YUV to AVC	1080p	10	40	0	2	27	98	627	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	0	2	27	99	508	22
YUV to AV1	1080p	10	40	0	2	27	100	420	20
YUV to AVC	1080p	10	40	0	3	27	98	626	19
YUV to HEVC	1080p	10	40	0	3	27	99	336	18
YUV to AV1	1080p	10	40	0	3	27	99	268	18
YUV to AVC	1080p	10	40	1	1	27	100	501	21
YUV to HEVC	1080p	10	40	1	1	27	99	647	22
YUV to AVC	1080p	10	40	1	2	27	99	500	20
YUV to HEVC	1080p	10	40	1	2	27	99	358	18
YUV to AVC	1080p	10	40	1	3	27	100	501	21
YUV to HEVC	1080p	10	40	1	3	27	99	240	18

Appendix A: GStreamer XStack Command

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

[illegible]

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

Appendix B: 7x7 Grid Layout

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

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

Appendix C: GStreamer Ladder Command

Example of single input with 64 outputs

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

```

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

```