



Quadra™ Performance Test Report V5.7

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 Throughput.....	44
13. T1A – GStreamer Latency.....	49
14. T1A – GStreamer RGBA Encoding	51
15. T1A – GStreamer XStack Throughput	53
16. T1A – GStreamer Ladder Generation.....	55
17. T1U – FFmpeg Throughput	56
18. T1U – Libxcodec Throughput	62
19. T1U – FFmpeg Latency	68
20. T1U – Decoder PPU Scaling.....	70
21. T1U – Streaming Ladder Generation	71
22. T1U – RGBA Encoding	72
23. T1U – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth	74
24. T1U – Capped CRF.....	80
25. T1U – Inplace Overlay	86
26. A Note on T1M Persistent Configurations	88
27. T1M (Persistent config “F”) – FFmpeg Throughput	89
28. T1M (Persistent config “F”) – Libxcodec Throughput	95

29. T1M (Persistent config "F") – FFmpeg Latency.....	101
30. T1M (Persistent config "F") – Decoder PPU Scaling.....	103
31. T1M (Persistent config "F") – Streaming Ladder Generation	104
32. T1M (Persistent config "F") – RGBA Encoding	105
33. T1M (Persistent config "F") – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth	107
34. T1M (Persistent config "F") – Capped CRF.....	113
35. T1M (Persistent config "F") – Inplace Overlay	119
36. T1M (Persistent config "E") – FFmpeg Throughput	121
37. T1M (Persistent config "E") – Libxcoder Throughput	123
38. T1M (Persistent config "E") – FFmpeg Latency.....	125
39. T1M (Persistent config "E") – RGBA Encoding	127
40. T1M (Persistent config "E") – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth.....	129
41. T1M (Persistent config "E") – Capped CRF	136
Appendix A: GStreamer XStack Command	143
Appendix B: 7x7 Grid Layout.....	145
Appendix C: GStreamer Ladder Command	146

Environment Overview

Revision: 5706t1r2

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
- Tests:
 - FFmpeg Throughput
 - Libxcodec Throughput
 - FFmpeg Latency
 - Decoder PPU Scaling
 - Streaming Ladder Generation
 - RGBA Encoding
 - Encoding EnableRdoQuant/rdoLevel/lookaheadDepth
 - Capped CRF
 - Inplace Overlay
 - Gstreamer Throughput
 - Gstreamer Latency
 - Gstreamer RGBA Encoding
 - Gstreamer XStack Throughput
 - Gstreamer Ladder Generation

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) 2400 MHz (0.42 ns)
- DUT: 2x T2A
- FFmpeg Version: 4.3.1
- Tests:
 - MultiThread P2P DMA on AMD GPU

Setup #4:

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

Definitions

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

1. T1A – FFmpeg Throughput

1.1 Decoding

1.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

1.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

1.2 Encoding

1.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

1.2.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

1.3 Transcoding

1.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

1.3.2 Command line

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

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

1.4 FFmpeg Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	11	0	105	6
HEVC to YUV	8k	1	0	8	1	11	0	104	6
VP9 to YUV	8k	1	0	8	1	24	0	38	4
YUV to AVC	8k	1	0	8	1	0	98	67	52
YUV to HEVC	8k	1	0	8	1	0	98	83	70
AVC to AVC	8k	1	1	8	1	7	99	54	4
AVC to HEVC	8k	1	1	8	1	8	98	71	7
HEVC to AVC	8k	1	1	8	1	7	100	53	6
HEVC to HEVC	8k	1	1	8	1	8	100	70	6
VP9 to AVC	8k	1	1	8	1	24	45	35	3
VP9 to HEVC	8k	1	1	8	1	24	42	37	2
AVC to YUV	8k	1	0	10	1	10	0	60	7
HEVC to YUV	8k	1	0	10	1	9	0	56	6
VP9 to YUV	8k	1	0	10	1	24	0	33	13
YUV to AVC	8k	1	0	10	1	0	92	48	75
YUV to HEVC	8k	1	0	10	1	0	73	59	94
AVC to YUV	4k	1	0	8	1	18	0	317	18
HEVC to YUV	4k	1	0	8	1	15	0	325	20
VP9 to YUV	4k	1	0	8	1	24	0	155	4
AVC to YUV	4k	16	0	8	0	100	0	485	4
HEVC to YUV	4k	16	0	8	0	99	0	504	3
VP9 to YUV	4k	16	0	8	0	100	0	487	1
YUV to AVC	4k	1	0	8	1	0	94	294	40
YUV to HEVC	4k	1	0	8	1	0	95	327	50
YUV to AV1	4k	1	0	8	1	0	94	282	38
YUV to AVC	4k	4	0	8	0	0	95	302	18
YUV to HEVC	4k	4	0	8	0	0	96	332	13
YUV to AV1	4k	4	0	8	0	0	96	288	13
YUV to AVC	4k	8	0	8	0	0	100	320	9
YUV to HEVC	4k	8	0	8	0	0	99	344	13
YUV to AV1	4k	8	0	8	0	0	100	296	7
AVC to AVC	4k	1	1	8	1	13	93	221	15
AVC to HEVC	4k	1	1	8	1	14	91	273	15
AVC to AV1	4k	1	1	8	1	16	97	260	12
HEVC to AVC	4k	1	1	8	1	13	93	220	15
HEVC to HEVC	4k	1	1	8	1	14	92	274	15
HEVC to AV1	4k	1	1	8	1	15	92	261	16
VP9 to AVC	4k	1	1	8	1	24	46	145	3
VP9 to HEVC	4k	1	1	8	1	24	43	149	8

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	51	151	6
AVC to AVC	4k	4	1	8	0	61	96	230	3
AVC to HEVC	4k	4	1	8	0	68	96	300	3
AVC to AV1	4k	4	1	8	0	55	95	276	6
HEVC to AVC	4k	4	1	8	0	56	96	236	6
HEVC to HEVC	4k	4	1	8	0	60	96	300	3
HEVC to AV1	4k	4	1	8	0	47	96	264	4
VP9 to AVC	4k	4	1	8	0	63	97	244	3
VP9 to HEVC	4k	4	1	8	0	66	96	304	2
VP9 to AV1	4k	4	1	8	0	54	96	276	3
AVC to AVC	4k	8	1	8	0	68	100	211	2
AVC to HEVC	4k	8	1	8	0	75	100	279	3
AVC to AV1	4k	8	1	8	0	68	100	272	3
HEVC to AVC	4k	8	1	8	0	63	100	216	2
HEVC to HEVC	4k	8	1	8	0	68	99	280	2
HEVC to AV1	4k	8	1	8	0	58	100	272	2
VP9 to AVC	4k	8	1	8	0	67	100	233	2
VP9 to HEVC	4k	8	1	8	0	73	100	296	2
VP9 to AV1	4k	8	1	8	0	65	100	280	2
AVC to YUV	4k	1	0	10	1	16	0	225	10
HEVC to YUV	4k	1	0	10	1	15	0	227	9
VP9 to YUV	4k	1	0	10	1	24	0	126	7
AVC to YUV	4k	16	0	10	0	98	0	283	0
HEVC to YUV	4k	16	0	10	0	103	0	279	0
VP9 to YUV	4k	16	0	10	0	99	0	264	0
YUV to AVC	4k	1	0	10	1	0	70	199	56
YUV to HEVC	4k	1	0	10	1	0	60	207	62
YUV to AV1	4k	1	0	10	1	0	67	199	62
YUV to AVC	4k	4	0	10	0	0	96	218	27
YUV to HEVC	4k	4	0	10	0	0	77	248	37
YUV to AV1	4k	4	0	10	0	0	81	240	39
AVC to YUV	1080p	1	0	8	1	36	0	868	22
HEVC to YUV	1080p	1	0	8	1	30	0	850	26
VP9 to YUV	1080p	1	0	8	1	22	0	557	4
AVC to YUV	1080p	40	0	8	0	88	0	1721	1
HEVC to YUV	1080p	40	0	8	0	95	0	1824	1
VP9 to YUV	1080p	40	0	8	0	81	0	1770	0
YUV to AVC	1080p	1	0	8	1	0	54	700	27
YUV to HEVC	1080p	1	0	8	1	0	50	700	30
YUV to AV1	1080p	1	0	8	1	0	54	644	24

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1280	3
YUV to HEVC	1080p	32	0	8	0	0	100	1357	3
YUV to AV1	1080p	32	0	8	0	0	100	1169	3
AVC to AVC	1080p	1	1	8	1	42	81	958	21
AVC to HEVC	1080p	1	1	8	1	43	77	1020	26
AVC to AV1	1080p	1	1	8	1	38	80	900	21
HEVC to AVC	1080p	1	1	8	1	35	77	903	32
HEVC to HEVC	1080p	1	1	8	1	37	77	981	32
HEVC to AV1	1080p	1	1	8	1	32	76	852	28
VP9 to AVC	1080p	1	1	8	1	21	42	548	7
VP9 to HEVC	1080p	1	1	8	1	22	39	542	8
VP9 to AV1	1080p	1	1	8	1	22	46	540	10
AVC to AVC	1080p	32	1	8	0	76	100	940	2
AVC to HEVC	1080p	32	1	8	0	84	99	1065	1
AVC to AV1	1080p	32	1	8	0	76	99	1026	1
HEVC to AVC	1080p	32	1	8	0	68	99	992	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1120	1
HEVC to AV1	1080p	32	1	8	0	68	99	1056	1
VP9 to AVC	1080p	32	1	8	0	70	100	1088	1
VP9 to HEVC	1080p	32	1	8	0	74	99	1216	1
VP9 to AV1	1080p	32	1	8	0	70	99	1120	1
AVC to YUV	1080p	1	0	10	1	28	0	671	7
HEVC to YUV	1080p	1	0	10	1	26	0	697	9
VP9 to YUV	1080p	1	0	10	1	22	0	456	7
AVC to YUV	1080p	40	0	10	0	65	0	1084	0
HEVC to YUV	1080p	40	0	10	0	67	0	1080	0
VP9 to YUV	1080p	40	0	10	0	75	0	1060	0
YUV to AVC	1080p	1	0	10	1	0	38	502	33
YUV to HEVC	1080p	1	0	10	1	0	36	503	42
YUV to AV1	1080p	1	0	10	1	0	38	467	37
YUV to AVC	1080p	32	0	10	0	0	62	799	6
YUV to HEVC	1080p	32	0	10	0	0	57	798	6
YUV to AV1	1080p	32	0	10	0	0	64	768	5
AVC to YUV	720p	1	0	8	1	43	0	1172	15
HEVC to YUV	720p	1	0	8	1	37	0	1162	22
VP9 to YUV	720p	1	0	8	1	33	0	1022	7
AVC to YUV	720p	100	0	8	0	100	0	2513	0
HEVC to YUV	720p	100	0	8	0	100	0	2921	1
VP9 to YUV	720p	100	0	8	0	100	0	2682	0
YUV to AVC	720p	1	0	8	1	0	30	872	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	879	20
YUV to AV1	720p	1	0	8	1	0	34	789	15
YUV to AVC	720p	64	0	8	0	0	97	2357	2
YUV to HEVC	720p	64	0	8	0	0	96	2378	2
YUV to AV1	720p	64	0	8	0	0	99	2036	1
AVC to AVC	720p	1	1	8	1	48	48	1276	19
AVC to HEVC	720p	1	1	8	1	48	48	1272	24
AVC to AV1	720p	1	1	8	1	40	51	1097	18
HEVC to AVC	720p	1	1	8	1	40	46	1233	26
HEVC to HEVC	720p	1	1	8	1	40	46	1239	24
HEVC to AV1	720p	1	1	8	1	34	49	1075	24
VP9 to AVC	720p	1	1	8	1	32	35	996	10
VP9 to HEVC	720p	1	1	8	1	32	34	989	14
VP9 to AV1	720p	1	1	8	1	32	44	992	12
AVC to AVC	720p	64	1	8	0	95	100	2048	0
AVC to HEVC	720p	64	1	8	0	96	100	2113	0
AVC to AV1	720p	64	1	8	0	76	100	1795	0
HEVC to AVC	720p	64	1	8	0	83	100	2052	0
HEVC to HEVC	720p	64	1	8	0	83	100	2121	0
HEVC to AV1	720p	64	1	8	0	68	100	1793	0
VP9 to AVC	720p	64	1	8	0	95	100	2245	0
VP9 to HEVC	720p	64	1	8	0	100	100	2317	0
VP9 to AV1	720p	64	1	8	0	78	100	1920	0

2. T1A – Libxcoder Throughput

2.1 Decoding

2.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

2.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

2.2 Encoding

2.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

2.2.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

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

2.3 Transcoding

2.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

2.3.2 Command line

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

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

2.4 Libxcode Throughput Performance Results

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	10	0	92	7
HEVC to YUV	8k	1	0	8	1	10	0	97	6
VP9 to YUV	8k	1	0	8	1	24	0	38	3
YUV to AVC	8k	1	0	8	1	0	98	68	30
YUV to HEVC	8k	1	0	8	1	0	95	81	35
AVC to AVC	8k	1	1	8	1	10	100	54	5
AVC to HEVC	8k	1	1	8	1	8	99	71	5
HEVC to AVC	8k	1	1	8	1	10	100	53	4
HEVC to HEVC	8k	1	1	8	1	9	99	70	5
VP9 to AVC	8k	1	1	8	1	24	46	35	4
VP9 to HEVC	8k	1	1	8	1	24	44	37	4
AVC to YUV	8k	1	0	10	1	9	0	54	7
HEVC to YUV	8k	1	0	10	1	8	0	51	6
VP9 to YUV	8k	1	0	10	1	24	0	33	4
YUV to AVC	8k	1	0	10	1	0	65	39	37
YUV to HEVC	8k	1	0	10	1	0	46	39	37
AVC to YUV	4k	1	0	8	1	17	0	320	8
HEVC to YUV	4k	1	0	8	1	16	0	331	6
VP9 to YUV	4k	1	0	8	1	24	0	154	3
AVC to YUV	4k	16	0	8	0	97	0	488	0
HEVC to YUV	4k	16	0	8	0	100	0	511	0
VP9 to YUV	4k	16	0	8	0	100	0	493	0
YUV to AVC	4k	1	0	8	1	0	89	278	26
YUV to HEVC	4k	1	0	8	1	0	83	289	27
YUV to AV1	4k	1	0	8	1	0	88	262	24
YUV to AVC	4k	4	0	8	0	0	96	311	12
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	326	7
YUV to HEVC	4k	8	0	8	0	0	100	347	8
YUV to AV1	4k	8	0	8	0	0	99	301	8
AVC to AVC	4k	1	1	8	0	16	97	224	7
AVC to HEVC	4k	1	1	8	0	17	96	277	8
AVC to AV1	4k	1	1	8	0	17	96	263	8
HEVC to AVC	4k	1	1	8	0	16	97	225	6
HEVC to HEVC	4k	1	1	8	0	16	96	286	6
HEVC to AV1	4k	1	1	8	0	16	97	266	7
VP9 to AVC	4k	1	1	8	0	24	46	149	4

TYPE	RES	JOBS	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	4
AVC to AVC	4k	4	1	8	0	61	96	241	5
AVC to HEVC	4k	4	1	8	0	67	96	302	5
AVC to AV1	4k	4	1	8	0	54	95	278	5
HEVC to AVC	4k	4	1	8	0	57	96	239	5
HEVC to HEVC	4k	4	1	8	0	59	96	294	5
HEVC to AV1	4k	4	1	8	0	48	95	274	5
VP9 to AVC	4k	4	1	8	0	63	96	246	4
VP9 to HEVC	4k	4	1	8	0	66	96	304	4
VP9 to AV1	4k	4	1	8	0	55	95	276	4
AVC to AVC	4k	8	1	8	0	68	99	217	3
AVC to HEVC	4k	8	1	8	0	75	99	280	3
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	3
HEVC to AV1	4k	8	1	8	0	60	100	275	3
VP9 to AVC	4k	8	1	8	0	68	100	235	3
VP9 to HEVC	4k	8	1	8	0	71	99	304	3
VP9 to AV1	4k	8	1	8	0	61	99	287	3
AVC to YUV	4k	1	0	10	1	15	0	214	7
HEVC to YUV	4k	1	0	10	1	13	0	207	6
VP9 to YUV	4k	1	0	10	1	24	0	125	3
AVC to YUV	4k	16	0	10	0	98	0	285	0
HEVC to YUV	4k	16	0	10	0	99	0	284	0
VP9 to YUV	4k	16	0	10	0	102	0	267	0
YUV to AVC	4k	1	0	10	1	0	49	158	28
YUV to HEVC	4k	1	0	10	1	0	45	158	28
YUV to AV1	4k	1	0	10	1	0	51	154	27
YUV to AVC	4k	4	0	10	0	0	73	208	26
YUV to HEVC	4k	4	0	10	0	0	59	212	27
YUV to AV1	4k	4	0	10	0	0	60	185	29
AVC to YUV	1080p	1	0	8	1	34	0	837	14
HEVC to YUV	1080p	1	0	8	1	31	0	888	11
VP9 to YUV	1080p	1	0	8	1	22	0	557	5
AVC to YUV	1080p	40	0	8	0	81	0	1614	0
HEVC to YUV	1080p	40	0	8	0	86	0	1721	0
VP9 to YUV	1080p	40	0	8	0	71	0	1653	0
YUV to AVC	1080p	1	0	8	1	0	57	738	17

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	50	697	18
YUV to AV1	1080p	1	0	8	1	0	50	597	15
YUV to AVC	1080p	32	0	8	0	0	99	1325	2
YUV to HEVC	1080p	32	0	8	0	0	99	1401	3
YUV to AV1	1080p	32	0	8	0	0	99	1210	2
AVC to AVC	1080p	1	1	8	0	43	85	1005	16
AVC to HEVC	1080p	1	1	8	0	45	82	1069	16
AVC to AV1	1080p	1	1	8	0	41	82	936	17
HEVC to AVC	1080p	1	1	8	0	39	86	1006	12
HEVC to HEVC	1080p	1	1	8	0	41	86	1089	13
HEVC to AV1	1080p	1	1	8	0	37	83	935	13
VP9 to AVC	1080p	1	1	8	0	22	41	535	6
VP9 to HEVC	1080p	1	1	8	0	21	39	541	6
VP9 to AV1	1080p	1	1	8	0	22	46	540	7
AVC to AVC	1080p	32	1	8	0	76	99	977	1
AVC to HEVC	1080p	32	1	8	0	83	100	1105	1
AVC to AV1	1080p	32	1	8	0	77	100	1064	1
HEVC to AVC	1080p	32	1	8	0	70	99	1021	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1157	1
HEVC to AV1	1080p	32	1	8	0	69	99	1091	1
VP9 to AVC	1080p	32	1	8	0	70	99	1118	0
VP9 to HEVC	1080p	32	1	8	0	74	100	1260	0
VP9 to AV1	1080p	32	1	8	0	72	99	1157	1
AVC to YUV	1080p	1	0	10	1	28	0	669	7
HEVC to YUV	1080p	1	0	10	1	26	0	678	7
VP9 to YUV	1080p	1	0	10	1	22	0	456	4
AVC to YUV	1080p	40	0	10	0	61	0	1075	0
HEVC to YUV	1080p	40	0	10	0	67	0	1071	0
VP9 to YUV	1080p	40	0	10	0	69	0	1029	0
YUV to AVC	1080p	1	0	10	1	0	37	486	20
YUV to HEVC	1080p	1	0	10	1	0	32	448	23
YUV to AV1	1080p	1	0	10	1	0	34	401	21
YUV to AVC	1080p	32	0	10	0	0	65	855	6
YUV to HEVC	1080p	32	0	10	0	0	60	846	6
YUV to AV1	1080p	32	0	10	0	0	66	799	8
AVC to YUV	720p	1	0	8	1	42	0	1174	12
HEVC to YUV	720p	1	0	8	1	39	0	1241	9
VP9 to YUV	720p	1	0	8	1	31	0	1021	6
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	94	0	2944	0
VP9 to YUV	720p	100	0	8	0	92	0	2681	0
YUV to AVC	720p	1	0	8	1	0	33	975	12
YUV to HEVC	720p	1	0	8	1	0	33	968	12
YUV to AV1	720p	1	0	8	1	0	31	746	11
YUV to AVC	720p	64	0	8	0	0	100	2658	2
YUV to HEVC	720p	64	0	8	0	0	99	2733	2
YUV to AV1	720p	64	0	8	0	0	100	2199	1
AVC to AVC	720p	1	1	8	0	52	50	1376	16
AVC to HEVC	720p	1	1	8	0	51	50	1369	16
AVC to AV1	720p	1	1	8	0	43	49	1073	14
HEVC to AVC	720p	1	1	8	0	46	51	1386	11
HEVC to HEVC	720p	1	1	8	0	46	51	1396	12
HEVC to AV1	720p	1	1	8	0	38	49	1078	11
VP9 to AVC	720p	1	1	8	0	32	35	1007	7
VP9 to HEVC	720p	1	1	8	0	31	34	992	8
VP9 to AV1	720p	1	1	8	0	31	44	993	9
AVC to AVC	720p	64	1	8	0	100	100	2187	0
AVC to HEVC	720p	64	1	8	0	100	100	2278	0
AVC to AV1	720p	64	1	8	0	95	100	1891	0
HEVC to AVC	720p	64	1	8	0	86	99	2195	0
HEVC to HEVC	720p	64	1	8	0	95	100	2282	0
HEVC to AV1	720p	64	1	8	0	88	100	1901	0
VP9 to AVC	720p	64	1	8	0	100	100	2432	0
VP9 to HEVC	720p	64	1	8	0	100	100	2508	0
VP9 to AV1	720p	64	1	8	0	98	100	2035	0

3. T1A – FFmpeg Latency

3.1 Encoding

3.1.1 Description

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

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

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

YUV frame is encoded by hardware encoder.

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

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

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

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

3.1.2 Command line

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

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

<resolution> is resolution of input

3.2 FFmpeg Latency Performance Results

TYPE	RESOLUTION	JOBS	ELAT_AVG (ms)	ELAT_MAX (ms)	ELAT_MIN (ms)	ELAT_VAR (ms)
YUV to AVC	8k	1	57.84	59.45	57.33	0.04
YUV to HEVC	8k	1	55.46	60.36	54.11	0.91
YUV to AVC	4k	1	15.38	17.44	14.99	0.07
YUV to HEVC	4k	1	16.6	18.59	15.66	0.17
YUV to AV1	4k	1	21.7	25.42	16.37	0.61
YUV to AVC	4k	4	16.97	21.4	14.97	2.91
YUV to HEVC	4k	4	18.51	22.04	15.58	1.97
YUV to AV1	4k	4	22.89	27.73	16.47	0.89
YUV to AVC	4k	8	18.23	23.62	15.05	3.85
YUV to HEVC	4k	8	19.12	24.42	15.61	3.12
YUV to AV1	4k	8	37.25	47.01	25.53	8.64
YUV to AVC	1080p	1	4.58	5.46	4.4	0.02
YUV to HEVC	1080p	1	4.88	5.46	4.55	0.01
YUV to AV1	1080p	1	6.7	7.44	5.04	0.05
YUV to AVC	1080p	32	6.08	10.35	4.63	0.33
YUV to HEVC	1080p	32	6.77	11.6	4.87	1.09
YUV to AV1	1080p	32	41.02	46.92	34.45	1.34
YUV to AVC	720p	1	2.81	3.5	2.5	0.01
YUV to HEVC	720p	1	2.83	3.23	2.65	0.01
YUV to AV1	720p	1	3.75	4.15	3.07	0.01
YUV to AVC	720p	64	5.95	10.49	3.35	1.06
YUV to HEVC	720p	64	6.08	10.37	3.43	1.15
YUV to AV1	720p	64	39.78	47.92	32.28	3.29

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	5	0	142	7
HEVC to RGBA	8k	1	4	0	153	8
VP9 to RGBA	8k	1	23	0	40	2
AVC to RGBA	4k	1	21	0	145	8
AVC to RGBA	4k	16	94	3	584	2
HEVC to RGBA	4k	1	22	0	174	10
HEVC to RGBA	4k	16	94	4	668	3
VP9 to RGBA	4k	1	22	0	166	3
VP9 to RGBA	4k	16	92	4	679	1
AVC to RGBA	1080p	40	93	14	1896	1
HEVC to RGBA	1080p	40	93	14	2012	1
VP9 to RGBA	1080p	40	92	16	2440	0
AVC to RGBA	720p	100	95	18	2604	0
HEVC to RGBA	720p	100	90	20	2843	0
VP9 to RGBA	720p	64	95	18	2752	0

5. T1A – Streaming Ladder Generation

5.1 Transcoding

5.1.1 Description

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

Decoded YUV frame is kept on device.

The YUV frames are encoded with hardware encoder.

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

5.1.2 Command line

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

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

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

Input: 1080p

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

5.2 Streaming Ladder Generation Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	8	33	94	2	437	4
AVC to HEVC	8	32	94	2	464	4
AVC to AV1	8	23	95	2	392	3
HEVC to AVC	8	34	93	2	440	5
HEVC to HEVC	8	32	93	2	472	5
HEVC to AV1	8	24	94	2	401	4
VP9 to AVC	8	39	93	2	432	4
VP9 to HEVC	8	38	92	2	466	4
VP9 to AV1	8	29	93	2	400	3

6. T1A – RGBA Encoding

6.1 Encoding

6.1.1 Description

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

RGBA frame is uploaded and encoded by hardware encoder.

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

6.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

6.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	53	162	66
RGBA to HEVC	4k	1	1	46	161	67
RGBA to AV1	4k	1	1	56	165	72
RGBA to AVC	4k	4	0	69	165	40
RGBA to HEVC	4k	4	0	61	170	40
RGBA to AV1	4k	4	0	61	172	43
RGBA to AVC	4k	8	0	63	176	26
RGBA to HEVC	4k	8	0	55	176	30
RGBA to AV1	4k	8	0	61	176	30
RGBA to AVC	1080p	1	1	34	443	46
RGBA to HEVC	1080p	1	1	32	441	48
RGBA to AV1	1080p	1	1	35	422	44
RGBA to AVC	1080p	16	0	59	651	14
RGBA to HEVC	1080p	16	0	53	656	15
RGBA to AV1	1080p	16	0	56	640	15
RGBA to AVC	1080p	32	0	56	671	8
RGBA to HEVC	1080p	32	0	52	672	8
RGBA to AV1	1080p	32	0	57	648	9
RGBA to AVC	720p	1	1	29	717	60
RGBA to HEVC	720p	1	1	29	791	67
RGBA to AV1	720p	1	1	35	724	63
RGBA to AVC	720p	16	0	49	1227	37
RGBA to HEVC	720p	16	0	46	1226	36
RGBA to AV1	720p	16	0	54	1159	32
RGBA to AVC	720p	32	0	45	1169	31
RGBA to HEVC	720p	32	0	44	1166	31
RGBA to AV1	720p	32	0	56	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	93	288	26
YUV to HEVC	4k	4	0	0	1	96	332	13
YUV to AV1	4k	4	0	0	1	96	288	12
YUV to AVC	4k	4	0	0	2	96	305	17
YUV to HEVC	4k	4	0	0	2	99	171	9
YUV to AV1	4k	4	0	0	2	98	140	8
YUV to AVC	4k	4	0	0	3	96	298	19
YUV to HEVC	4k	4	0	0	3	98	100	5
YUV to AV1	4k	4	0	0	3	100	76	4
YUV to AVC	4k	4	0	1	1	98	188	12
YUV to HEVC	4k	4	0	1	1	97	236	14
YUV to AVC	4k	4	0	1	2	98	188	7
YUV to HEVC	4k	4	0	1	2	99	108	6
YUV to AVC	4k	4	0	1	3	97	188	8
YUV to HEVC	4k	4	0	1	3	100	68	4
YUV to AVC	4k	4	4	0	1	99	196	10
YUV to HEVC	4k	4	4	0	1	100	228	12
YUV to AV1	4k	4	4	0	1	99	200	10
YUV to AVC	4k	4	4	0	2	99	196	8
YUV to HEVC	4k	4	4	0	2	100	136	6
YUV to AV1	4k	4	4	0	2	100	116	6
YUV to AVC	4k	4	4	0	3	99	196	10
YUV to HEVC	4k	4	4	0	3	99	88	4
YUV to AV1	4k	4	4	0	3	102	72	4
YUV to AVC	4k	4	4	1	1	99	140	5
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	6
YUV to AVC	4k	4	4	1	3	99	140	6
YUV to HEVC	4k	4	4	1	3	101	64	4
YUV to AVC	4k	4	16	0	1	100	196	9
YUV to HEVC	4k	4	16	0	1	99	228	11
YUV to AV1	4k	4	16	0	1	100	198	11
YUV to AVC	4k	4	16	0	2	100	196	9
YUV to HEVC	4k	4	16	0	2	100	136	6
YUV to AV1	4k	4	16	0	2	99	116	6
YUV to AVC	4k	4	16	0	3	99	196	10
YUV to HEVC	4k	4	16	0	3	99	88	5

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	4k	4	16	0	3	99	72	4
YUV to AVC	4k	4	16	1	1	99	140	6
YUV to HEVC	4k	4	16	1	1	100	180	9
YUV to AVC	4k	4	16	1	2	100	140	6
YUV to HEVC	4k	4	16	1	2	99	96	5
YUV to AVC	4k	4	16	1	3	99	140	6
YUV to HEVC	4k	4	16	1	3	99	64	3
YUV to AVC	4k	4	40	0	1	99	196	9
YUV to HEVC	4k	4	40	0	1	100	224	12
YUV to AV1	4k	4	40	0	1	100	196	10
YUV to AVC	4k	4	40	0	2	99	196	10
YUV to HEVC	4k	4	40	0	2	100	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	101	88	5
YUV to AV1	4k	4	40	0	3	99	72	4
YUV to AVC	4k	4	40	1	1	99	136	6
YUV to HEVC	4k	4	40	1	1	100	176	9
YUV to AVC	4k	4	40	1	2	99	136	7
YUV to HEVC	4k	4	40	1	2	99	92	5
YUV to AVC	4k	4	40	1	3	99	136	6
YUV to HEVC	4k	4	40	1	3	96	64	5
YUV to AVC	1080p	20	0	0	1	100	1280	4
YUV to HEVC	1080p	20	0	0	1	99	1360	5
YUV to AV1	1080p	20	0	0	1	99	1180	4
YUV to AVC	1080p	20	0	0	2	99	1280	4
YUV to HEVC	1080p	20	0	0	2	100	684	2
YUV to AV1	1080p	20	0	0	2	100	560	2
YUV to AVC	1080p	20	0	0	3	100	1280	4
YUV to HEVC	1080p	20	0	0	3	99	401	2
YUV to AV1	1080p	20	0	0	3	99	300	1
YUV to AVC	1080p	20	0	1	1	99	760	2
YUV to HEVC	1080p	20	0	1	1	100	960	3
YUV to AVC	1080p	20	0	1	2	100	760	2
YUV to HEVC	1080p	20	0	1	2	100	440	2
YUV to AVC	1080p	20	0	1	3	100	760	2
YUV to HEVC	1080p	20	0	1	3	100	280	1
YUV to AVC	1080p	20	4	0	1	99	700	2
YUV to HEVC	1080p	20	4	0	1	99	820	3

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	99	720	2
YUV to AVC	1080p	20	4	0	2	99	700	2
YUV to HEVC	1080p	20	4	0	2	99	517	2
YUV to AV1	1080p	20	4	0	2	100	420	2
YUV to AVC	1080p	20	4	0	3	99	700	2
YUV to HEVC	1080p	20	4	0	3	100	340	1
YUV to AV1	1080p	20	4	0	3	99	269	1
YUV to AVC	1080p	20	4	1	1	100	500	2
YUV to HEVC	1080p	20	4	1	1	100	659	2
YUV to AVC	1080p	20	4	1	2	99	501	2
YUV to HEVC	1080p	20	4	1	2	99	360	1
YUV to AVC	1080p	20	4	1	3	100	500	2
YUV to HEVC	1080p	20	4	1	3	100	240	1
YUV to AVC	1080p	20	16	0	1	99	700	2
YUV to HEVC	1080p	20	16	0	1	99	820	3
YUV to AV1	1080p	20	16	0	1	99	720	2
YUV to AVC	1080p	20	16	0	2	99	700	2
YUV to HEVC	1080p	20	16	0	2	99	502	2
YUV to AV1	1080p	20	16	0	2	100	420	1
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	100	263	1
YUV to AVC	1080p	20	16	1	1	99	500	2
YUV to HEVC	1080p	20	16	1	1	99	648	2
YUV to AVC	1080p	20	16	1	2	99	500	2
YUV to HEVC	1080p	20	16	1	2	99	360	2
YUV to AVC	1080p	20	16	1	3	100	500	2
YUV to HEVC	1080p	20	16	1	3	99	240	1
YUV to AVC	1080p	20	40	0	1	99	688	2
YUV to HEVC	1080p	20	40	0	1	99	801	3
YUV to AV1	1080p	20	40	0	1	99	703	3
YUV to AVC	1080p	20	40	0	2	99	690	2
YUV to HEVC	1080p	20	40	0	2	100	500	2
YUV to AV1	1080p	20	40	0	2	100	420	1
YUV to AVC	1080p	20	40	0	3	99	693	2
YUV to HEVC	1080p	20	40	0	3	100	339	2
YUV to AV1	1080p	20	40	0	3	100	260	1
YUV to AVC	1080p	20	40	1	1	99	500	2
YUV to HEVC	1080p	20	40	1	1	99	640	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	2	100	500	2
YUV to HEVC	1080p	20	40	1	2	100	360	1
YUV to AVC	1080p	20	40	1	3	99	500	2
YUV to HEVC	1080p	20	40	1	3	99	240	1
YUV to AVC	720p	40	0	0	1	95	2333	3
YUV to HEVC	720p	40	0	0	1	91	2356	3
YUV to AV1	720p	40	0	0	1	97	2044	2
YUV to AVC	720p	40	0	0	2	95	2329	3
YUV to HEVC	720p	40	0	0	2	99	1520	1
YUV to AV1	720p	40	0	0	2	99	1200	1
YUV to AVC	720p	40	0	0	3	94	2336	3
YUV to HEVC	720p	40	0	0	3	99	920	1
YUV to AV1	720p	40	0	0	3	99	650	1
YUV to AVC	720p	40	0	1	1	99	1694	1
YUV to HEVC	720p	40	0	1	1	99	2080	2
YUV to AVC	720p	40	0	1	2	99	1691	1
YUV to HEVC	720p	40	0	1	2	99	960	1
YUV to AVC	720p	40	0	1	3	99	1684	1
YUV to HEVC	720p	40	0	1	3	100	613	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	1040	1
YUV to AVC	720p	40	4	0	2	100	1440	1
YUV to HEVC	720p	40	4	0	2	99	1080	1
YUV to AV1	720p	40	4	0	2	99	880	1
YUV to AVC	720p	40	4	0	3	99	1440	1
YUV to HEVC	720p	40	4	0	3	99	720	1
YUV to AV1	720p	40	4	0	3	100	560	1
YUV to AVC	720p	40	4	1	1	99	1077	1
YUV to HEVC	720p	40	4	1	1	100	1339	1
YUV to AVC	720p	40	4	1	2	99	1076	1
YUV to HEVC	720p	40	4	1	2	99	760	1
YUV to AVC	720p	40	4	1	3	99	1077	1
YUV to HEVC	720p	40	4	1	3	99	520	0
YUV to AVC	720p	40	16	0	1	100	1400	1
YUV to HEVC	720p	40	16	0	1	100	1360	1
YUV to AV1	720p	40	16	0	1	100	1080	1
YUV to AVC	720p	40	16	0	2	100	1402	1
YUV to HEVC	720p	40	16	0	2	99	1079	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	720p	40	16	0	2	99	880	1
YUV to AVC	720p	40	16	0	3	100	1400	1
YUV to HEVC	720p	40	16	0	3	99	720	1
YUV to AV1	720p	40	16	0	3	100	560	1
YUV to AVC	720p	40	16	1	1	99	1049	1
YUV to HEVC	720p	40	16	1	1	99	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	1047	1
YUV to HEVC	720p	40	16	1	3	99	520	1
YUV to AVC	720p	40	40	0	1	100	1362	1
YUV to HEVC	720p	40	40	0	1	100	1323	1
YUV to AV1	720p	40	40	0	1	100	1042	1
YUV to AVC	720p	40	40	0	2	100	1361	1
YUV to HEVC	720p	40	40	0	2	99	1040	1
YUV to AV1	720p	40	40	0	2	99	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	100	557	1
YUV to AVC	720p	40	40	1	1	99	1040	1
YUV to HEVC	720p	40	40	1	1	100	1320	1
YUV to AVC	720p	40	40	1	2	99	1040	1
YUV to HEVC	720p	40	40	1	2	100	760	1
YUV to AVC	720p	40	40	1	3	99	1040	1
YUV to HEVC	720p	40	40	1	3	99	520	0

8. T1A – Capped CRF

8.1 Encoding with lookaheadDepth

8.1.1 Description

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

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

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

8.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

8.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	19	99	701	2
YUV to HEVC	1080p	20	0	0	1	19	99	820	3
YUV to AV1	1080p	20	0	0	1	19	99	720	2
YUV to AVC	1080p	20	0	0	2	19	99	700	2
YUV to HEVC	1080p	20	0	0	2	19	99	520	2
YUV to AV1	1080p	20	0	0	2	19	100	439	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	503	2
YUV to HEVC	1080p	20	0	1	1	19	99	660	2
YUV to AVC	1080p	20	0	1	2	19	99	500	2
YUV to HEVC	1080p	20	0	1	2	19	99	360	1
YUV to AVC	1080p	20	0	1	3	19	99	500	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	100	420	2
YUV to AVC	1080p	20	4	0	3	19	99	700	2
YUV to HEVC	1080p	20	4	0	3	19	99	340	2
YUV to AV1	1080p	20	4	0	3	19	99	271	1
YUV to AVC	1080p	20	4	1	1	19	99	500	2
YUV to HEVC	1080p	20	4	1	1	19	99	660	2
YUV to AVC	1080p	20	4	1	2	19	99	500	2
YUV to HEVC	1080p	20	4	1	2	19	100	360	1
YUV to AVC	1080p	20	4	1	3	19	100	500	2
YUV to HEVC	1080p	20	4	1	3	19	100	240	1
YUV to AVC	1080p	20	16	0	1	19	99	700	2
YUV to HEVC	1080p	20	16	0	1	19	99	819	3
YUV to AV1	1080p	20	16	0	1	19	99	720	3
YUV to AVC	1080p	20	16	0	2	19	99	700	2
YUV to HEVC	1080p	20	16	0	2	19	100	501	2
YUV to AV1	1080p	20	16	0	2	19	99	420	1
YUV to AVC	1080p	20	16	0	3	19	99	700	2
YUV to HEVC	1080p	20	16	0	3	19	99	340	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	3	19	100	262	1
YUV to AVC	1080p	20	16	1	1	19	99	500	2
YUV to HEVC	1080p	20	16	1	1	19	100	646	2
YUV to AVC	1080p	20	16	1	2	19	99	500	2
YUV to HEVC	1080p	20	16	1	2	19	99	360	1
YUV to AVC	1080p	20	16	1	3	19	99	500	2
YUV to HEVC	1080p	20	16	1	3	19	100	240	1
YUV to AVC	1080p	20	40	0	1	19	99	688	3
YUV to HEVC	1080p	20	40	0	1	19	99	801	3
YUV to AV1	1080p	20	40	0	1	19	99	701	3
YUV to AVC	1080p	20	40	0	2	19	99	690	2
YUV to HEVC	1080p	20	40	0	2	19	99	500	2
YUV to AV1	1080p	20	40	0	2	19	100	420	1
YUV to AVC	1080p	20	40	0	3	19	99	694	2
YUV to HEVC	1080p	20	40	0	3	19	100	339	1
YUV to AV1	1080p	20	40	0	3	19	99	261	1
YUV to AVC	1080p	20	40	1	1	19	100	500	2
YUV to HEVC	1080p	20	40	1	1	19	100	640	2
YUV to AVC	1080p	20	40	1	2	19	99	500	2
YUV to HEVC	1080p	20	40	1	2	19	99	360	1
YUV to AVC	1080p	20	40	1	3	19	100	500	2
YUV to HEVC	1080p	20	40	1	3	19	100	240	1
YUV to AVC	1080p	20	0	0	1	23	99	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	3
YUV to AVC	1080p	20	0	0	2	23	99	700	2
YUV to HEVC	1080p	20	0	0	2	23	100	520	2
YUV to AV1	1080p	20	0	0	2	23	100	439	1
YUV to AVC	1080p	20	0	0	3	23	99	700	2
YUV to HEVC	1080p	20	0	0	3	23	100	340	1
YUV to AV1	1080p	20	0	0	3	23	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	502	2
YUV to HEVC	1080p	20	0	1	2	23	99	360	1
YUV to AVC	1080p	20	0	1	3	23	99	501	2
YUV to HEVC	1080p	20	0	1	3	23	99	240	1
YUV to AVC	1080p	20	4	0	1	23	100	700	2
YUV to HEVC	1080p	20	4	0	1	23	100	820	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	23	99	720	2
YUV to AVC	1080p	20	4	0	2	23	99	700	2
YUV to HEVC	1080p	20	4	0	2	23	99	518	2
YUV to AV1	1080p	20	4	0	2	23	99	420	2
YUV to AVC	1080p	20	4	0	3	23	100	700	2
YUV to HEVC	1080p	20	4	0	3	23	99	340	1
YUV to AV1	1080p	20	4	0	3	23	100	268	1
YUV to AVC	1080p	20	4	1	1	23	100	500	2
YUV to HEVC	1080p	20	4	1	1	23	99	660	2
YUV to AVC	1080p	20	4	1	2	23	99	501	2
YUV to HEVC	1080p	20	4	1	2	23	100	360	2
YUV to AVC	1080p	20	4	1	3	23	99	500	2
YUV to HEVC	1080p	20	4	1	3	23	100	240	1
YUV to AVC	1080p	20	16	0	1	23	99	700	2
YUV to HEVC	1080p	20	16	0	1	23	99	820	3
YUV to AV1	1080p	20	16	0	1	23	99	720	2
YUV to AVC	1080p	20	16	0	2	23	99	700	2
YUV to HEVC	1080p	20	16	0	2	23	100	501	2
YUV to AV1	1080p	20	16	0	2	23	99	420	2
YUV to AVC	1080p	20	16	0	3	23	99	700	3
YUV to HEVC	1080p	20	16	0	3	23	99	340	2
YUV to AV1	1080p	20	16	0	3	23	99	262	1
YUV to AVC	1080p	20	16	1	1	23	99	500	2
YUV to HEVC	1080p	20	16	1	1	23	100	650	2
YUV to AVC	1080p	20	16	1	2	23	99	500	2
YUV to HEVC	1080p	20	16	1	2	23	99	360	1
YUV to AVC	1080p	20	16	1	3	23	99	500	2
YUV to HEVC	1080p	20	16	1	3	23	99	240	1
YUV to AVC	1080p	20	40	0	1	23	99	694	2
YUV to HEVC	1080p	20	40	0	1	23	99	802	3
YUV to AV1	1080p	20	40	0	1	23	99	704	2
YUV to AVC	1080p	20	40	0	2	23	99	687	2
YUV to HEVC	1080p	20	40	0	2	23	100	500	2
YUV to AV1	1080p	20	40	0	2	23	99	420	2
YUV to AVC	1080p	20	40	0	3	23	99	691	2
YUV to HEVC	1080p	20	40	0	3	23	100	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	JOBS	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	99	360	1
YUV to AVC	1080p	20	40	1	3	23	100	500	2
YUV to HEVC	1080p	20	40	1	3	23	100	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	700	2
YUV to HEVC	1080p	20	0	0	2	27	99	520	2
YUV to AV1	1080p	20	0	0	2	27	100	439	2
YUV to AVC	1080p	20	0	0	3	27	99	702	2
YUV to HEVC	1080p	20	0	0	3	27	100	340	2
YUV to AV1	1080p	20	0	0	3	27	99	280	1
YUV to AVC	1080p	20	0	1	1	27	100	503	2
YUV to HEVC	1080p	20	0	1	1	27	99	660	2
YUV to AVC	1080p	20	0	1	2	27	100	502	2
YUV to HEVC	1080p	20	0	1	2	27	99	360	1
YUV to AVC	1080p	20	0	1	3	27	100	502	2
YUV to HEVC	1080p	20	0	1	3	27	99	240	1
YUV to AVC	1080p	20	4	0	1	27	99	700	2
YUV to HEVC	1080p	20	4	0	1	27	99	820	3
YUV to AV1	1080p	20	4	0	1	27	99	720	2
YUV to AVC	1080p	20	4	0	2	27	99	700	2
YUV to HEVC	1080p	20	4	0	2	27	100	518	2
YUV to AV1	1080p	20	4	0	2	27	100	420	1
YUV to AVC	1080p	20	4	0	3	27	99	700	2
YUV to HEVC	1080p	20	4	0	3	27	99	340	1
YUV to AV1	1080p	20	4	0	3	27	100	268	1
YUV to AVC	1080p	20	4	1	1	27	99	501	2
YUV to HEVC	1080p	20	4	1	1	27	99	660	2
YUV to AVC	1080p	20	4	1	2	27	99	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	100	240	1
YUV to AVC	1080p	20	16	0	1	27	100	700	2
YUV to HEVC	1080p	20	16	0	1	27	99	819	3
YUV to AV1	1080p	20	16	0	1	27	99	720	3
YUV to AVC	1080p	20	16	0	2	27	99	700	2
YUV to HEVC	1080p	20	16	0	2	27	99	502	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	100	264	1
YUV to AVC	1080p	20	16	1	1	27	99	500	2
YUV to HEVC	1080p	20	16	1	1	27	99	646	2
YUV to AVC	1080p	20	16	1	2	27	100	500	2
YUV to HEVC	1080p	20	16	1	2	27	100	360	1
YUV to AVC	1080p	20	16	1	3	27	100	500	2
YUV to HEVC	1080p	20	16	1	3	27	100	240	1
YUV to AVC	1080p	20	40	0	1	27	99	692	2
YUV to HEVC	1080p	20	40	0	1	27	99	800	3
YUV to AV1	1080p	20	40	0	1	27	99	703	3
YUV to AVC	1080p	20	40	0	2	27	99	691	2
YUV to HEVC	1080p	20	40	0	2	27	99	500	2
YUV to AV1	1080p	20	40	0	2	27	99	420	1
YUV to AVC	1080p	20	40	0	3	27	99	693	2
YUV to HEVC	1080p	20	40	0	3	27	99	340	1
YUV to AV1	1080p	20	40	0	3	27	99	260	1
YUV to AVC	1080p	20	40	1	1	27	99	500	2
YUV to HEVC	1080p	20	40	1	1	27	99	640	2
YUV to AVC	1080p	20	40	1	2	27	99	500	2
YUV to HEVC	1080p	20	40	1	2	27	100	360	1
YUV to AVC	1080p	20	40	1	3	27	100	500	2
YUV to HEVC	1080p	20	40	1	3	27	100	240	1

9. T1A – Inplace Overlay

9.1 Transcoding

9.1.1 Description

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

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

9.1.2 Command line

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

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

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

Input Video: 1080p

Input Image: 128x128

9.2 Inplace Overlay Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	1	14	20	5	289	11
AVC to HEVC	1	14	21	5	307	12
AVC to AV1	1	13	20	4	262	11
HEVC to AVC	1	12	20	5	289	15
HEVC to HEVC	1	13	21	5	307	15
HEVC to AV1	1	12	20	4	263	14
VP9 to AVC	1	17	20	4	288	11
VP9 to HEVC	1	18	20	5	306	11
VP9 to AV1	1	16	20	4	262	11
AVC to AVC	16	71	92	24	1043	3
AVC to HEVC	16	78	93	27	1157	3
AVC to AV1	16	73	95	26	1107	3
HEVC to AVC	16	71	93	24	1088	4
HEVC to HEVC	16	78	94	28	1216	4
HEVC to AV1	16	67	92	25	1137	4
VP9 to AVC	16	82	94	23	1040	3
VP9 to HEVC	16	91	95	27	1168	3
VP9 to AV1	16	83	93	25	1120	3
AVC to AVC	32	74	93	23	960	2
AVC to HEVC	32	82	94	27	1073	1
AVC to AV1	32	75	92	25	1056	1
HEVC to AVC	32	72	93	24	996	1
HEVC to HEVC	32	80	94	28	1135	2
HEVC to AV1	32	71	92	26	1088	2
VP9 to AVC	32	82	94	23	992	1
VP9 to HEVC	32	90	95	27	1120	1
VP9 to AV1	32	83	93	25	1088	1

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

10.1 Encoding

10.1.1 Description

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

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

The YUV frame is encoded with hardware encoder.

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

10.1.2 Command line

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

<num_jobs> = number of instances running concurrently

10.2 Multi Thread P2P DMA on AMD GPU Performance Results

TYPE	RES	JOBS	Frames	FPS	CPU	Enc Load	P2P MEM	Latency Avg	Latency Dev
P2A	720p	1	1001	30	1.4	0.25	0.5	2.08	0.01
P2H	720p	1	1001	30	1.2	0.25	0.5	2.07	0.01
P2A	720p	180	180180	29.7	0.79	46	56	2.3	0.08
P2H	720p	180	180180	29.6	0.77	45	56	2.3	0.11
P2A	1080p	1	1001	30	1.3	0.5	0.75	3.78	0.01
P2H	1080p	1	1001	30	1.2	0.5	0.75	3.58	0.01
P2A	1080p	80	80080	30	0.9	46	50	3.92	0.24
P2H	1080p	80	80080	29.7	0.89	43	50	3.77	0.24

11. T1A – AI

11.1 AI Model

11.1.1 Description

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

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

aiperf performs any data format conversion expected by the hardware.

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

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

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

11.1.2 Command line

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

In config_example.json, user need to specify the following arguments

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

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

11.1.3 AI Model Performance Results

Model	Instance Number	Loops	Channel Order	File Format	File Type	Input Size	Total FPS
yolov5s_640	8	10000	rgb	nchw	image	640x640x3	78
yolov5s_320	8	10000	rgb	nchw	image	320x320x3	278
deeplabv3_FRP	8	10000	rgb	nchw	image	257x257x3	359
resnet50	8	10000	rgb	nchw	image	224x224x3	228
mobilenetv2	8	10000	rgb	nchw	image	224x224x3	1112
deeplabv3	8	10000	rgb	nchw	image	257x257x3	160
yolov4	8	10000	bgr	nchw	image	416x416x3	257
fsrcnn	8	2000	bgr	nchw	image	360x640x1	31
BiSeNetv1	8	10000	rgb	nchw	image	512x512x3	76
HrNet	8	10000	rgb	nchw	image	256x192x3	74
usm_1656x1920	8	10000	rgb	nchw	image	1920x1656x1	251
usm_3240x3840	8	10000	rgb	nchw	image	3840x3240x1	61
lpips	8	2000	rgb	nchw	image	720x480x3	1
PaddleOCR-512_onnx	8	10000	rgb	nchw	image	512x48x3	16
segm32_tflite_kl_mle	8	10000	rgb	nchw	image	256x144x3	837
mobilenetv2_nchw_keras_96x160	1	10000	rgb	nchw	image	96x160x3	2353.5
mobilenetv2_nchw_keras_96x160	8	10000	rgb	nchw	image	96x160x3	2347.75
mobilenetv2_nchw_keras_96x160	16	10000	rgb	nchw	image	96x160x3	2345.12

11.2 AI Encoding with 2D Engine

11.2.1 Description BG Filter

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

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

11.2.2 Description ROI Filter

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

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

11.2.3 Description PRE Filter

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

11.2.4 Command line BG

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

11.2.5 Command line ROI

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

11.2.6 Command line PRE

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

11.2.7 AI Encoding with 2D Engine Performance Results

Filter	Model	Resolution	Session Number	Average FPS per session
ROI	network_binary_yolov4_head	1920x1080	1	80
ROI	network_binary_yolov4_head	1920x1080	8	22
ROI	network_binary_yolov4_head	1920x1080	32	5
BG	segm32_tflite_nchw_bgr	1920x1080	1	59
BG	segm32_tflite_nchw_bgr	1920x1080	8	40
BG	segm32_tflite_nchw_bgr	1920x1080	32	15
PRE	hw_lanczos_network_binary	1920x1080	8	66
PRE	hw_lanczos_network_binary	1920x1080	16	34
PRE	hw_bicubic_network_binary	1920x1080	8	66
PRE	hw_bicubic_network_binary	1920x1080	16	34

12. T1A – GStreamer Throughput

12.1 Decoding

12.1.1 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.

12.1.2 Command Line

```
gst-launch-1.0 -v --gst-debug=3 multifilesrc
location=/media/ramdisk/input.list loop=true ! <dec>parse ! multiqueue
! niquadra<dec>dec dec=0 ! fpsdisplaysink video-sink=fakesink
sync=false
```

<dec> is the decoder codec. eg h264, h265

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

12.2 Encoding

12.2.1 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.

12.2.2 Command Line

```
gst-launch-1.0 -v --gst-debug=3 multifilesrc
location=/media/ramdisk/input.yuv loop=true ! rawvideoparse format=43
width=<width> height=<height> ! multiqueue ! niquadra<enc>enc enc=0
xcoder-params=intraPeriod=0:RcEnable=1:bitrate=<*> ! fpsdisplaysink
video-sink=fakesink sync=false
```

<enc> is the encoder codec. eg h264, h265, av1

<num_jobs> is the number of instances running concurrently

<width> and <height> make up the 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)

12.3 Transcoding

12.3.1 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.

12.3.2 Command line

```
gst-launch-1.0 -v --gst-debug=3 multifilesrc
location=/media/ramdisk/input.list loop=true ! <dec>parse ! multiqueue
! niquadra<dec>dec dec=0 xcoder-
params=out=hw:sempianar0=1:multicoreJointMode=<*> ! niquadra<enc>enc
enc=0 xcoder-
params=intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> !
fpsdisplaysink video-sink=fakesink sync=false
```

<dec> is the decoder codec. ie h264, h265

<enc> is the encoder codec. ie h264, h265, av1

<num_jobs> is the number of instances running concurrently

<width> and <height> make up the 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)

12.4 GStreamer 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	5	0	44	67
HEVC to YUV	8k	1	0	8	1	5	0	44	65
YUV to AVC	8k	1	0	8	1	0	26	29	104
YUV to HEVC	8k	1	0	8	1	0	33	29	104
AVC to AVC	8k	1	1	8	1	3	93	54	6
AVC to HEVC	8k	1	1	8	1	3	97	71	7
HEVC to AVC	8k	1	1	8	1	2	93	52	6
HEVC to HEVC	8k	1	1	8	1	3	97	70	7
AVC to YUV	8k	1	0	10	1	5	0	27	63
HEVC to YUV	8k	1	0	10	1	5	0	27	63
YUV to AVC	8k	1	0	10	1	0	13	14	104
YUV to HEVC	8k	1	0	10	1	0	17	15	104
AVC to YUV	4k	1	0	8	1	9	0	169	62
HEVC to YUV	4k	1	0	8	1	9	0	172	61
AVC to YUV	4k	16	0	8	0	98	0	484	29
HEVC to YUV	4k	16	0	8	0	99	0	504	33
YUV to AVC	4k	1	0	8	1	0	42	131	73
YUV to HEVC	4k	1	0	8	1	0	40	137	77
YUV to AV1	4k	1	0	8	1	0	47	128	71
YUV to AVC	4k	4	0	8	0	0	79	245	44
YUV to HEVC	4k	4	0	8	0	0	76	258	46
YUV to AV1	4k	4	0	8	0	0	81	233	41
YUV to AVC	4k	8	0	8	0	0	100	307	33
YUV to HEVC	4k	8	0	8	0	0	99	339	39
YUV to AV1	4k	8	0	8	0	0	100	288	30
AVC to AVC	4k	1	1	8	1	9	89	224	7
AVC to HEVC	4k	1	1	8	1	12	93	281	7
AVC to AV1	4k	1	1	8	1	11	95	263	7
HEVC to AVC	4k	1	1	8	1	9	93	224	7
HEVC to HEVC	4k	1	1	8	1	10	94	283	7
HEVC to AV1	4k	1	1	8	1	9	97	263	7
AVC to AVC	4k	4	1	8	0	62	97	239	3
AVC to HEVC	4k	4	1	8	0	67	95	295	3
AVC to AV1	4k	4	1	8	0	59	95	273	3
HEVC to AVC	4k	4	1	8	0	58	97	234	3
HEVC to HEVC	4k	4	1	8	0	60	96	295	3
HEVC to AV1	4k	4	1	8	0	46	96	275	3
AVC to AVC	4k	8	1	8	0	68	100	217	2

AVC to HEVC	4k	8	1	8	0	76	100	280	3
AVC to AV1	4k	8	1	8	0	68	99	270	3
HEVC to AVC	4k	8	1	8	0	64	100	219	2
HEVC to HEVC	4k	8	1	8	0	67	99	283	2
HEVC to AV1	4k	8	1	8	0	58	100	273	2
AVC to YUV	4k	1	0	10	1	8	0	108	60
HEVC to YUV	4k	1	0	10	1	7	0	109	59
AVC to YUV	4k	16	0	10	0	83	0	272	31
HEVC to YUV	4k	16	0	10	0	86	0	271	33
YUV to AVC	4k	1	0	10	1	0	24	75	81
YUV to HEVC	4k	1	0	10	1	0	22	73	82
YUV to AV1	4k	1	0	10	1	0	25	74	81
YUV to AVC	4k	4	0	10	0	0	56	179	71
YUV to HEVC	4k	4	0	10	0	0	51	183	74
YUV to AV1	4k	4	0	10	0	0	54	167	67
AVC to YUV	1080p	1	0	8	1	21	0	490	52
HEVC to YUV	1080p	1	0	8	1	19	0	499	54
AVC to YUV	1080p	40	0	8	0	79	0	1556	15
HEVC to YUV	1080p	40	0	8	0	77	0	1620	16
YUV to AVC	1080p	1	0	8	1	0	36	462	68
YUV to HEVC	1080p	1	0	8	1	0	33	461	66
YUV to AV1	1080p	1	0	8	1	0	39	438	67
YUV to AVC	1080p	32	0	8	0	0	99	1280	17
YUV to HEVC	1080p	32	0	8	0	0	99	1353	19
YUV to AV1	1080p	32	0	8	0	0	99	1140	14
AVC to AVC	1080p	1	1	8	1	28	57	735	14
AVC to HEVC	1080p	1	1	8	1	28	53	732	13
AVC to AV1	1080p	1	1	8	1	25	57	664	12
HEVC to AVC	1080p	1	1	8	1	27	61	773	14
HEVC to HEVC	1080p	1	1	8	1	26	58	786	14
HEVC to AV1	1080p	1	1	8	1	23	61	696	13
AVC to AVC	1080p	32	1	8	0	75	100	950	2
AVC to HEVC	1080p	32	1	8	0	82	99	1074	2
AVC to AV1	1080p	32	1	8	0	73	99	1031	2
HEVC to AVC	1080p	32	1	8	0	71	99	992	2
HEVC to HEVC	1080p	32	1	8	0	76	99	1124	2
HEVC to AV1	1080p	32	1	8	0	68	99	1054	2
AVC to YUV	1080p	1	0	10	1	16	0	360	48
HEVC to YUV	1080p	1	0	10	1	15	0	366	53
AVC to YUV	1080p	40	0	10	0	60	0	1028	15
HEVC to YUV	1080p	40	0	10	0	69	0	1025	15

YUV to AVC	1080p	1	0	10	1	0	19	249	70
YUV to HEVC	1080p	1	0	10	1	0	17	247	71
YUV to AV1	1080p	1	0	10	1	0	19	240	69
YUV to AVC	1080p	32	0	10	0	0	57	743	30
YUV to HEVC	1080p	32	0	10	0	0	51	736	31
YUV to AV1	1080p	32	0	10	0	0	62	738	29
AVC to YUV	720p	1	0	8	1	29	0	775	37
HEVC to YUV	720p	1	0	8	1	26	0	821	41
AVC to YUV	720p	100	0	8	0	100	0	2548	4
HEVC to YUV	720p	100	0	8	0	97	0	2802	5
YUV to AVC	720p	1	0	8	1	0	32	906	26
YUV to HEVC	720p	1	0	8	1	0	32	908	25
YUV to AV1	720p	1	0	8	1	0	36	813	22
YUV to AVC	720p	64	0	8	0	0	99	2309	4
YUV to HEVC	720p	64	0	8	0	0	99	2337	4
YUV to AV1	720p	64	0	8	0	0	99	1959	3
AVC to AVC	720p	1	1	8	1	29	28	799	11
AVC to HEVC	720p	1	1	8	1	28	27	797	11
AVC to AV1	720p	1	1	8	1	26	31	712	10
HEVC to AVC	720p	1	1	8	1	27	29	853	11
HEVC to HEVC	720p	1	1	8	1	27	29	856	11
HEVC to AV1	720p	1	1	8	1	24	32	760	10
AVC to AVC	720p	64	1	8	0	95	100	2043	1
AVC to HEVC	720p	64	1	8	0	94	100	2130	1
AVC to AV1	720p	64	1	8	0	75	99	1792	1
HEVC to AVC	720p	64	1	8	0	82	99	2042	1
HEVC to HEVC	720p	64	1	8	0	83	100	2138	1
HEVC to AV1	720p	64	1	8	0	67	100	1811	1

13. T1A – GStreamer Latency

13.1 Encoding

13.1.1 Description

Libxcoder 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.

13.1.2 Command Line

```
gst-launch-1.0 multifilesrc location=/media/ramdisk/input.yuv
loop=true ! videoparse width=<width> height=<height> format=2 !
niquadra<enc>enc enc=0 xcoder-params=gopPresetIdx=9:lowDelay=1 !
fpsdisplaysink video-sink=fakesink sync=false
```

<enc> is the encoder codec. ie niquadrah264enc, niquadrah265enc, niquadraav1enc

<width> and <height> make up the resolution of input

13.2 GStreamer 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	58.8	57.53	0.03
YUV to HEVC	8k	1	54.88	59.3	54.08	0.56
YUV to AVC	4k	1	15.24	17.43	14.91	0.07
YUV to HEVC	4k	1	15.83	18.48	15.01	0.18
YUV to AV1	4k	1	19.96	25.43	16.21	1.23
YUV to AVC	4k	4	15.6	20.01	14.94	0.44
YUV to HEVC	4k	4	16.26	22.03	15.07	0.53
YUV to AV1	4k	4	20.43	30.07	16.39	1.59
YUV to AVC	4k	8	20.92	34.95	15.12	26.07
YUV to HEVC	4k	8	21.53	37.6	15	26.86
YUV to AV1	4k	8	29.15	45.81	16.47	42.13
YUV to AVC	1080p	1	4.51	6.18	4.38	0.02
YUV to HEVC	1080p	1	4.72	6.16	4.43	0.03
YUV to AV1	1080p	1	6.09	7.7	4.93	0.11
YUV to AVC	1080p	32	20.01	88.17	4.41	28.07
YUV to HEVC	1080p	32	21.35	54.95	4.44	28.36
YUV to AV1	1080p	32	31.52	45.29	5.03	61.46
YUV to AVC	720p	1	2.57	4.12	2.38	0.04
YUV to HEVC	720p	1	2.8	4.16	2.56	0.01
YUV to AV1	720p	1	3.48	5.04	2.95	0.04
YUV to AVC	720p	64	21.46	90.34	2.4	50.32
YUV to HEVC	720p	64	21.46	90.11	2.56	50.67
YUV to AV1	720p	64	29.38	50.44	3.08	85.65

14. T1A – GStreamer RGBA Encoding

14.1 Encoding

14.1.1 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.

14.1.2 Command line

```
gst-launch-1.0 -v --gst-debug=3 multifilesrc
location=/media/ramdisk/input.rgba loop=true ! rawvideoparse format=11
width=<width> height=<height> ! multiqueue ! videoconvert !
niquadra<enc>enc enc=0 xcoder-
params=intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> !
fpsdisplaysink video-sink=fakesink sync=false
```

<enc> is the encoder codec. eg h264, h265

<num_jobs> is the number of instances running concurrently

<width> and <height> make up the 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

14.2 GStreamer RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	18	54	104
RGBA to HEVC	4k	1	1	13	54	104
RGBA to AVC	4k	4	0	45	144	87
RGBA to HEVC	4k	4	0	43	146	90
RGBA to AVC	4k	8	0	50	150	96
RGBA to HEVC	4k	8	0	43	150	98
RGBA to AVC	1080p	1	1	15	195	76
RGBA to HEVC	1080p	1	1	13	194	77
RGBA to AVC	1080p	16	0	47	585	72
RGBA to HEVC	1080p	16	0	43	580	68
RGBA to AVC	1080p	32	0	42	546	63
RGBA to HEVC	1080p	32	0	41	548	63
RGBA to AVC	720p	1	1	15	381	69
RGBA to HEVC	720p	1	1	14	382	69
RGBA to AVC	720p	16	0	49	1208	36
RGBA to HEVC	720p	16	0	50	1204	35
RGBA to AVC	720p	32	0	46	1160	32
RGBA to HEVC	720p	32	0	48	1164	32

15. T1A – GStreamer XStack Throughput

15.1 Transcoding

15.1.1 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.

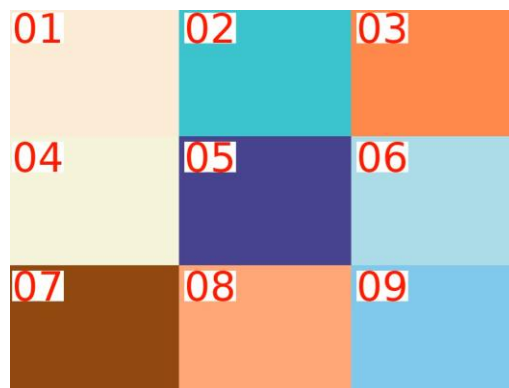
The encoded bitstream is read out through PCIe and written into an output file.

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

15.1.2 Command line

See Appendix A: GStreamer XStack Command

15.2 GStreamer XStack Performance Results

Input Res	Grid	Output Res	Cell Size	FPS	CPU	Dec Load	Enc Load	Scaler Load
1920x1080	3x3	1920x1080	640x360	151.91	29	66	13	16
1920x1080	4x4	1920x1080	480x270	105.99	42	82	10	15
1920x1080	4x8	1920x1080	480x135	57.1	63	87	5	8
1920x1080	3x3	3840x2160	1280x720	69.96	14	33	22	20
1920x1080	4x4	3840x2160	960x540	56.63	23	47	19	19
1920x1080	4x8	3840x2160	960x270	44.07	42	73	15	18
1920x1080	3x3	7680x4320	2560x1440	19.87	8	8	22	15
1920x1080	4x4	7680x4320	1920x1080	19.15	10	15	22	17
1920x1080	4x8	7680x4320	1920x540	19.46	20	33	22	20
1920x1080	6x6	1920x1080	320x180	50.83	63	89	5	10
1920x1080	7x7	1920x1080	274x154 276x154 274x156 276x156*	37.54	63	88	4	8

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

16. T1A – GStreamer Ladder Generation

16.1 Transcoding

16.1.1 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

16.1.2 Command line

See Appendix C: GStreamer Ladder Command

16.2 GStreamer Ladder Performance Results

Jobs	Outputs	FPS	CPU	Dec Load	Enc Load
1	64	18.99	59	1	91

17. T1U – FFmpeg Throughput

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.

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

17.1.2 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

17.2 Encoding

17.2.1 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.

17.2.2 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)

17.3 Transcoding

17.3.1 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.

17.3.2 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)

17.4 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	10	0	96	12
HEVC to YUV	8k	1	0	8	1	9	0	96	14
VP9 to YUV	8k	1	0	8	1	24	0	30	3
YUV to AVC	8k	1	0	8	1	0	97	58	78
YUV to HEVC	8k	1	0	8	1	0	98	72	102
AVC to AVC	8k	1	1	8	1	7	99	47	1
AVC to HEVC	8k	1	1	8	1	8	99	63	8
HEVC to AVC	8k	1	1	8	1	8	100	47	0
HEVC to HEVC	8k	1	1	8	1	8	100	62	3
VP9 to AVC	8k	1	1	8	1	25	42	29	2
VP9 to HEVC	8k	1	1	8	1	25	40	30	1
AVC to YUV	8k	1	0	10	1	10	0	56	6
HEVC to YUV	8k	1	0	10	1	8	0	51	4
VP9 to YUV	8k	1	0	10	1	25	0	28	6
YUV to AVC	8k	1	0	10	1	0	92	43	114
YUV to HEVC	8k	1	0	10	1	0	59	44	124
AVC to YUV	4k	1	0	8	1	17	0	302	15
HEVC to YUV	4k	1	0	8	1	15	0	294	19
VP9 to YUV	4k	1	0	8	1	24	0	109	8
AVC to YUV	4k	16	0	8	0	98	0	422	0
HEVC to YUV	4k	16	0	8	0	99	0	451	1
VP9 to YUV	4k	16	0	8	0	98	0	400	0
YUV to AVC	4k	1	0	8	1	0	96	260	36
YUV to HEVC	4k	1	0	8	1	0	96	287	43
YUV to AV1	4k	1	0	8	1	0	94	246	34
YUV to AVC	4k	4	0	8	0	0	96	264	17
YUV to HEVC	4k	4	0	8	0	0	97	292	11
YUV to AV1	4k	4	0	8	0	0	96	252	11
YUV to AVC	4k	8	0	8	0	0	99	280	6
YUV to HEVC	4k	8	0	8	0	0	99	296	6
YUV to AV1	4k	8	0	8	0	0	100	256	7
AVC to AVC	4k	1	1	8	1	15	96	202	13
AVC to HEVC	4k	1	1	8	1	14	92	246	12
AVC to AV1	4k	1	1	8	1	15	94	234	15
HEVC to AVC	4k	1	1	8	1	14	94	201	15
HEVC to HEVC	4k	1	1	8	1	14	92	248	8
HEVC to AV1	4k	1	1	8	1	14	93	230	13
VP9 to AVC	4k	1	1	8	1	23	39	108	0
VP9 to HEVC	4k	1	1	8	1	24	36	109	3

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	42	109	7
AVC to AVC	4k	4	1	8	0	62	96	220	3
AVC to HEVC	4k	4	1	8	0	67	96	272	2
AVC to AV1	4k	4	1	8	0	54	95	244	6
HEVC to AVC	4k	4	1	8	0	54	96	220	2
HEVC to HEVC	4k	4	1	8	0	56	96	272	3
HEVC to AV1	4k	4	1	8	0	45	94	240	7
VP9 to AVC	4k	4	1	8	0	64	96	216	1
VP9 to HEVC	4k	4	1	8	0	69	96	264	2
VP9 to AV1	4k	4	1	8	0	60	94	240	7
AVC to AVC	4k	8	1	8	0	68	100	195	1
AVC to HEVC	4k	8	1	8	0	76	100	256	1
AVC to AV1	4k	8	1	8	0	68	100	240	6
HEVC to AVC	4k	8	1	8	0	61	99	195	1
HEVC to HEVC	4k	8	1	8	0	65	99	256	1
HEVC to AV1	4k	8	1	8	0	53	99	248	7
VP9 to AVC	4k	8	1	8	0	68	98	199	1
VP9 to HEVC	4k	8	1	8	0	74	100	256	1
VP9 to AV1	4k	8	1	8	0	66	100	248	6
AVC to YUV	4k	1	0	10	1	15	0	209	6
HEVC to YUV	4k	1	0	10	1	13	0	208	7
VP9 to YUV	4k	1	0	10	1	24	0	105	3
AVC to YUV	4k	16	0	10	0	100	0	253	0
HEVC to YUV	4k	16	0	10	0	100	0	255	0
VP9 to YUV	4k	16	0	10	0	100	0	235	0
YUV to AVC	4k	1	0	10	1	0	75	184	55
YUV to HEVC	4k	1	0	10	1	0	67	202	62
YUV to AV1	4k	1	0	10	1	0	74	193	60
YUV to AVC	4k	4	0	10	0	0	94	195	24
YUV to HEVC	4k	4	0	10	0	0	83	230	40
YUV to AV1	4k	4	0	10	0	0	90	226	38
AVC to YUV	1080p	1	0	8	1	34	0	810	16
HEVC to YUV	1080p	1	0	8	1	27	0	760	27
VP9 to YUV	1080p	1	0	8	1	22	0	455	4
AVC to YUV	1080p	40	0	8	0	98	0	1586	2
HEVC to YUV	1080p	40	0	8	0	99	0	1544	2
VP9 to YUV	1080p	40	0	8	0	95	0	1711	0
YUV to AVC	1080p	1	0	8	1	0	62	709	27
YUV to HEVC	1080p	1	0	8	1	0	59	708	27
YUV to AV1	1080p	1	0	8	1	0	62	648	25

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	2
YUV to HEVC	1080p	32	0	8	0	0	99	1184	2
YUV to AV1	1080p	32	0	8	0	0	99	1024	2
AVC to AVC	1080p	1	1	8	1	41	81	880	16
AVC to HEVC	1080p	1	1	8	1	43	80	928	21
AVC to AV1	1080p	1	1	8	1	37	82	797	18
HEVC to AVC	1080p	1	1	8	1	35	81	829	25
HEVC to HEVC	1080p	1	1	8	1	35	76	878	27
HEVC to AV1	1080p	1	1	8	1	30	77	760	25
VP9 to AVC	1080p	1	1	8	1	22	40	450	4
VP9 to HEVC	1080p	1	1	8	1	22	38	453	6
VP9 to AV1	1080p	1	1	8	1	22	44	453	4
AVC to AVC	1080p	32	1	8	0	76	99	864	1
AVC to HEVC	1080p	32	1	8	0	84	99	974	1
AVC to AV1	1080p	32	1	8	0	77	99	928	2
HEVC to AVC	1080p	32	1	8	0	72	100	896	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1010	1
HEVC to AV1	1080p	32	1	8	0	71	99	942	3
VP9 to AVC	1080p	32	1	8	0	69	99	992	0
VP9 to HEVC	1080p	32	1	8	0	73	99	1099	0
VP9 to AV1	1080p	32	1	8	0	68	100	992	0
AVC to YUV	1080p	1	0	10	1	27	0	623	6
HEVC to YUV	1080p	1	0	10	1	24	0	629	6
VP9 to YUV	1080p	1	0	10	1	22	0	361	4
AVC to YUV	1080p	40	0	10	0	72	0	1080	0
HEVC to YUV	1080p	40	0	10	0	81	0	1080	0
VP9 to YUV	1080p	40	0	10	0	92	0	1040	0
YUV to AVC	1080p	1	0	10	1	0	42	475	32
YUV to HEVC	1080p	1	0	10	1	0	39	477	36
YUV to AV1	1080p	1	0	10	1	0	43	444	36
YUV to AVC	1080p	32	0	10	0	0	71	815	6
YUV to HEVC	1080p	32	0	10	0	0	69	830	7
YUV to AV1	1080p	32	0	10	0	0	77	801	6
AVC to YUV	720p	1	0	8	1	45	0	1185	18
HEVC to YUV	720p	1	0	8	1	39	0	1187	24
VP9 to YUV	720p	1	0	8	1	28	0	859	8
AVC to YUV	720p	100	0	8	0	100	0	2400	0
HEVC to YUV	720p	100	0	8	0	98	0	2715	0
VP9 to YUV	720p	100	0	8	0	100	0	2500	0
YUV to AVC	720p	1	0	8	1	0	33	865	15

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	33	869	15
YUV to AV1	720p	1	0	8	1	0	35	780	13
YUV to AVC	720p	64	0	8	0	0	99	2247	2
YUV to HEVC	720p	64	0	8	0	0	99	2309	2
YUV to AV1	720p	64	0	8	0	0	98	1920	1
AVC to AVC	720p	1	1	8	1	48	49	1210	18
AVC to HEVC	720p	1	1	8	1	48	48	1213	20
AVC to AV1	720p	1	1	8	1	41	52	1051	15
HEVC to AVC	720p	1	1	8	1	41	47	1164	20
HEVC to HEVC	720p	1	1	8	1	41	46	1172	24
HEVC to AV1	720p	1	1	8	1	35	49	1025	24
VP9 to AVC	720p	1	1	8	1	29	33	857	9
VP9 to HEVC	720p	1	1	8	1	28	33	852	7
VP9 to AV1	720p	1	1	8	1	28	40	851	10
AVC to AVC	720p	64	1	8	0	94	100	1865	1
AVC to HEVC	720p	64	1	8	0	97	100	1935	0
AVC to AV1	720p	64	1	8	0	77	100	1664	0
HEVC to AVC	720p	64	1	8	0	83	100	1870	1
HEVC to HEVC	720p	64	1	8	0	85	100	1953	1
HEVC to AV1	720p	64	1	8	0	67	100	1666	0
VP9 to AVC	720p	64	1	8	0	99	100	2112	0
VP9 to HEVC	720p	64	1	8	0	100	100	2176	0
VP9 to AV1	720p	64	1	8	0	78	100	1792	0

18. T1U – Libxcoder Throughput

18.1 Decoding

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.

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

18.1.2 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)

18.2 Encoding

18.2.1 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.

18.2.2 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)

18.3 Transcoding

18.3.1 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.

18.3.2 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)

18.4 Libxcode Throughput Performance Results

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	9	0	87	5
HEVC to YUV	8k	1	0	8	1	8	0	89	5
VP9 to YUV	8k	1	0	8	1	24	0	30	2
YUV to AVC	8k	1	0	8	1	0	94	57	25
YUV to HEVC	8k	1	0	8	1	0	96	72	31
AVC to AVC	8k	1	1	8	1	11	100	48	4
AVC to HEVC	8k	1	1	8	1	9	99	64	4
HEVC to AVC	8k	1	1	8	1	12	100	46	3
HEVC to HEVC	8k	1	1	8	1	9	100	63	3
VP9 to AVC	8k	1	1	8	1	25	42	29	1
VP9 to HEVC	8k	1	1	8	1	24	40	30	2
AVC to YUV	8k	1	0	10	1	8	0	50	4
HEVC to YUV	8k	1	0	10	1	7	0	47	4
VP9 to YUV	8k	1	0	10	1	24	0	28	2
YUV to AVC	8k	1	0	10	1	0	65	35	35
YUV to HEVC	8k	1	0	10	1	0	49	36	36
AVC to YUV	4k	1	0	8	1	17	0	295	7
HEVC to YUV	4k	1	0	8	1	16	0	294	5
VP9 to YUV	4k	1	0	8	1	24	0	109	2
AVC to YUV	4k	16	0	8	0	99	0	425	0
HEVC to YUV	4k	16	0	8	0	101	0	456	0
VP9 to YUV	4k	16	0	8	0	99	0	405	0
YUV to AVC	4k	1	0	8	1	0	90	248	22
YUV to HEVC	4k	1	0	8	1	0	89	268	23
YUV to AV1	4k	1	0	8	1	0	94	245	19
YUV to AVC	4k	4	0	8	0	0	96	273	8
YUV to HEVC	4k	4	0	8	0	0	97	291	8
YUV to AV1	4k	4	0	8	0	0	97	252	8
YUV to AVC	4k	8	0	8	0	0	99	285	5
YUV to HEVC	4k	8	0	8	0	0	99	302	5
YUV to AV1	4k	8	0	8	0	0	100	262	6
AVC to AVC	4k	1	1	8	0	16	97	205	5
AVC to HEVC	4k	1	1	8	0	17	96	253	5
AVC to AV1	4k	1	1	8	0	17	97	236	6
HEVC to AVC	4k	1	1	8	0	16	97	204	4
HEVC to HEVC	4k	1	1	8	0	16	97	256	4
HEVC to AV1	4k	1	1	8	0	17	97	237	4
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	23	36	109	2
VP9 to AV1	4k	1	1	8	0	24	42	109	3
AVC to AVC	4k	4	1	8	0	60	96	225	4
AVC to HEVC	4k	4	1	8	0	67	96	275	4
AVC to AV1	4k	4	1	8	0	56	95	246	4
HEVC to AVC	4k	4	1	8	0	56	96	220	3
HEVC to HEVC	4k	4	1	8	0	57	96	265	3
HEVC to AV1	4k	4	1	8	0	48	96	242	4
VP9 to AVC	4k	4	1	8	0	64	97	218	3
VP9 to HEVC	4k	4	1	8	0	69	96	268	3
VP9 to AV1	4k	4	1	8	0	57	95	243	3
AVC to AVC	4k	8	1	8	0	67	99	197	3
AVC to HEVC	4k	8	1	8	0	75	100	256	2
AVC to AV1	4k	8	1	8	0	64	99	247	3
HEVC to AVC	4k	8	1	8	0	61	98	198	2
HEVC to HEVC	4k	8	1	8	0	65	99	254	2
HEVC to AV1	4k	8	1	8	0	57	100	244	3
VP9 to AVC	4k	8	1	8	0	66	99	202	2
VP9 to HEVC	4k	8	1	8	0	76	99	262	3
VP9 to AV1	4k	8	1	8	0	65	100	249	3
AVC to YUV	4k	1	0	10	1	14	0	198	5
HEVC to YUV	4k	1	0	10	1	13	0	193	5
VP9 to YUV	4k	1	0	10	1	24	0	105	2
AVC to YUV	4k	16	0	10	0	97	0	257	0
HEVC to YUV	4k	16	0	10	0	98	0	256	0
VP9 to YUV	4k	16	0	10	0	100	0	238	0
YUV to AVC	4k	1	0	10	1	0	53	149	25
YUV to HEVC	4k	1	0	10	1	0	49	151	25
YUV to AV1	4k	1	0	10	1	0	54	144	24
YUV to AVC	4k	4	0	10	0	0	60	185	25
YUV to HEVC	4k	4	0	10	0	0	68	196	26
YUV to AV1	4k	4	0	10	0	0	66	179	29
AVC to YUV	1080p	1	0	8	1	34	0	773	12
HEVC to YUV	1080p	1	0	8	1	31	0	810	8
VP9 to YUV	1080p	1	0	8	1	22	0	455	3
AVC to YUV	1080p	40	0	8	0	96	0	1590	0
HEVC to YUV	1080p	40	0	8	0	98	0	1585	0
VP9 to YUV	1080p	40	0	8	0	89	0	1633	0
YUV to AVC	1080p	1	0	8	1	0	59	667	14

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	55	664	11
YUV to AV1	1080p	1	0	8	1	0	57	586	14
YUV to AVC	1080p	32	0	8	0	0	99	1155	1
YUV to HEVC	1080p	32	0	8	0	0	99	1224	1
YUV to AV1	1080p	32	0	8	0	0	99	1056	1
AVC to AVC	1080p	1	1	8	0	43	85	915	14
AVC to HEVC	1080p	1	1	8	0	45	85	976	15
AVC to AV1	1080p	1	1	8	0	40	84	823	15
HEVC to AVC	1080p	1	1	8	0	38	87	906	12
HEVC to HEVC	1080p	1	1	8	0	38	85	968	12
HEVC to AV1	1080p	1	1	8	0	35	86	824	12
VP9 to AVC	1080p	1	1	8	0	22	40	454	3
VP9 to HEVC	1080p	1	1	8	0	22	38	453	3
VP9 to AV1	1080p	1	1	8	0	22	44	453	5
AVC to AVC	1080p	32	1	8	0	76	99	897	0
AVC to HEVC	1080p	32	1	8	0	85	99	1011	0
AVC to AV1	1080p	32	1	8	0	77	99	964	1
HEVC to AVC	1080p	32	1	8	0	72	100	930	0
HEVC to HEVC	1080p	32	1	8	0	78	99	1043	0
HEVC to AV1	1080p	32	1	8	0	69	99	974	1
VP9 to AVC	1080p	32	1	8	0	70	99	1020	0
VP9 to HEVC	1080p	32	1	8	0	74	99	1141	0
VP9 to AV1	1080p	32	1	8	0	71	99	1025	0
AVC to YUV	1080p	1	0	10	1	28	0	591	6
HEVC to YUV	1080p	1	0	10	1	24	0	582	5
VP9 to YUV	1080p	1	0	10	1	22	0	360	3
AVC to YUV	1080p	40	0	10	0	71	0	1061	0
HEVC to YUV	1080p	40	0	10	0	79	0	1059	0
VP9 to YUV	1080p	40	0	10	0	81	0	1003	0
YUV to AVC	1080p	1	0	10	1	0	34	391	18
YUV to HEVC	1080p	1	0	10	1	0	32	391	19
YUV to AV1	1080p	1	0	10	1	0	34	348	16
YUV to AVC	1080p	32	0	10	0	0	73	834	6
YUV to HEVC	1080p	32	0	10	0	0	70	849	6
YUV to AV1	1080p	32	0	10	0	0	73	780	9
AVC to YUV	720p	1	0	8	1	43	0	1131	11
HEVC to YUV	720p	1	0	8	1	39	0	1166	7
VP9 to YUV	720p	1	0	8	1	29	0	859	4
AVC to YUV	720p	100	0	8	0	100	0	2595	0

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	90	0	2682	0
VP9 to YUV	720p	100	0	8	0	100	0	2585	0
YUV to AVC	720p	1	0	8	1	0	37	965	11
YUV to HEVC	720p	1	0	8	1	0	37	958	13
YUV to AV1	720p	1	0	8	1	0	33	742	10
YUV to AVC	720p	64	0	8	0	0	100	2534	2
YUV to HEVC	720p	64	0	8	0	0	100	2605	1
YUV to AV1	720p	64	0	8	0	0	100	2101	1
AVC to AVC	720p	1	1	8	0	52	51	1295	14
AVC to HEVC	720p	1	1	8	0	52	51	1291	14
AVC to AV1	720p	1	1	8	0	44	50	1037	13
HEVC to AVC	720p	1	1	8	0	45	51	1278	12
HEVC to HEVC	720p	1	1	8	0	45	51	1292	11
HEVC to AV1	720p	1	1	8	0	39	50	1045	11
VP9 to AVC	720p	1	1	8	0	29	33	851	5
VP9 to HEVC	720p	1	1	8	0	29	33	849	5
VP9 to AV1	720p	1	1	8	0	28	39	852	6
AVC to AVC	720p	64	1	8	0	100	100	1998	0
AVC to HEVC	720p	64	1	8	0	100	100	2087	0
AVC to AV1	720p	64	1	8	0	96	100	1754	0
HEVC to AVC	720p	64	1	8	0	96	100	1995	0
HEVC to HEVC	720p	64	1	8	0	97	100	2088	0
HEVC to AV1	720p	64	1	8	0	89	100	1758	0
VP9 to AVC	720p	64	1	8	0	100	100	2254	0
VP9 to HEVC	720p	64	1	8	0	100	100	2331	0
VP9 to AV1	720p	64	1	8	0	99	100	1905	0

19. T1U – FFmpeg Latency

19.1 Encoding

19.1.1 Description

Libxcoder 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.

19.1.2 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 -xcoder-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

19.2 FFmpeg Latency Performance Results

TYPE	RESOLUTION	JOBS	ELAT_AVG (ms)	ELAT_MAX (ms)	ELAT_MIN (ms)	ELAT_VAR (ms)
YUV to AVC	8k	1	65.34	66.68	64.7	0.09
YUV to HEVC	8k	1	62.53	68.23	60.82	1.24
YUV to AVC	4k	1	17.47	19.77	16.97	0.08
YUV to HEVC	4k	1	18.85	21.16	17.64	0.13
YUV to AV1	4k	1	24.77	29.07	18.37	0.87
YUV to AVC	4k	4	19.47	23.13	17.02	2.19
YUV to HEVC	4k	4	20.52	22.95	18.89	0.62
YUV to AV1	4k	4	25.56	30.81	19.5	1.24
YUV to AVC	4k	8	24.28	35.9	17.12	27.37
YUV to HEVC	4k	8	28.1	37.81	18.03	26.95
YUV to AV1	4k	8	42.82	51.04	30.99	7.07
YUV to AVC	1080p	1	5.37	6.65	5.01	0.09
YUV to HEVC	1080p	1	5.61	9.76	5.19	0.08
YUV to AV1	1080p	1	7.58	11.73	5.65	0.11
YUV to AVC	1080p	32	8.98	14.76	5.56	2.86
YUV to HEVC	1080p	32	10.36	18	5.99	2.95
YUV to AV1	1080p	32	47.14	54.89	39.44	4.8
YUV to AVC	720p	1	3.11	3.84	2.82	0.02
YUV to HEVC	720p	1	3.17	3.62	2.82	0.01
YUV to AV1	720p	1	4.16	4.53	3.26	0.01
YUV to AVC	720p	64	6.04	10.38	3.33	1.19
YUV to HEVC	720p	64	6.55	13.28	3.78	1.05
YUV to AV1	720p	64	45.7	53.98	38.1	2.83

20. T1U – Decoder PPU Scaling

20.1 Decoding

20.1.1 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.

20.1.2 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

20.2 Decoder PPU Scaling Performance Results

TYPE	RESOLUTION	JOBS	DEC_LOAD	SCALER_LOAD	FPS	CPU
AVC to RGBA	8k	1	4	0	116	5
HEVC to RGBA	8k	1	3	0	120	5
VP9 to RGBA	8k	1	22	0	31	1
AVC to RGBA	4k	1	22	0	114	6
AVC to RGBA	4k	16	93	2	468	2
HEVC to RGBA	4k	1	22	0	137	8
HEVC to RGBA	4k	16	95	3	551	2
VP9 to RGBA	4k	1	22	0	112	5
VP9 to RGBA	4k	16	95	2	462	1
AVC to RGBA	1080p	40	92	12	1600	1
HEVC to RGBA	1080p	40	94	11	1596	1
VP9 to RGBA	1080p	40	94	15	2060	0
AVC to RGBA	720p	100	97	18	2435	0
HEVC to RGBA	720p	100	93	20	2703	0
VP9 to RGBA	720p	64	96	19	2624	0

21. T1U – Streaming Ladder Generation

21.1 Transcoding

21.1.1 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.

21.1.2 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)

21.2 Streaming Ladder Generation Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	8	29	92	2	384	3
AVC to HEVC	8	29	92	2	409	3
AVC to AV1	8	22	92	1	352	3
HEVC to AVC	8	30	92	2	392	4
HEVC to HEVC	8	30	92	2	412	4
HEVC to AV1	8	23	92	1	352	3
VP9 to AVC	8	37	93	2	384	3
VP9 to HEVC	8	37	93	2	411	3
VP9 to AV1	8	29	92	1	352	3

22. T1U – RGBA Encoding

22.1 Encoding

22.1.1 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.

22.1.2 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

22.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	55	146	64
RGBA to HEVC	4k	1	1	50	149	66
RGBA to AV1	4k	1	1	59	150	69
RGBA to AVC	4k	4	0	68	152	35
RGBA to HEVC	4k	4	0	63	158	39
RGBA to AV1	4k	4	0	67	160	42
RGBA to AVC	4k	8	0	67	162	33
RGBA to HEVC	4k	8	0	62	168	37
RGBA to AV1	4k	8	0	69	170	34
RGBA to AVC	1080p	1	1	35	390	44
RGBA to HEVC	1080p	1	1	33	393	45
RGBA to AV1	1080p	1	1	36	378	44
RGBA to AVC	1080p	16	0	63	601	20
RGBA to HEVC	1080p	16	0	59	609	21
RGBA to AV1	1080p	16	0	62	607	21
RGBA to AVC	1080p	32	0	62	641	15
RGBA to HEVC	1080p	32	0	57	643	13
RGBA to AV1	1080p	32	0	64	642	12
RGBA to AVC	720p	1	1	29	645	57
RGBA to HEVC	720p	1	1	29	649	56
RGBA to AV1	720p	1	1	33	631	49
RGBA to AVC	720p	16	0	51	1233	39
RGBA to HEVC	720p	16	0	50	1237	41
RGBA to AV1	720p	16	0	58	1171	37
RGBA to AVC	720p	32	0	49	1172	31
RGBA to HEVC	720p	32	0	47	1172	32
RGBA to AV1	720p	32	0	57	1172	30

23. T1U – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

23.1 Encoding

23.1.1 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.

23.1.2 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

23.2 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	268	14
YUV to HEVC	4k	4	0	0	1	97	292	13
YUV to AV1	4k	4	0	0	1	97	252	11
YUV to AVC	4k	4	0	0	2	93	256	24
YUV to HEVC	4k	4	0	0	2	98	148	6
YUV to AV1	4k	4	0	0	2	97	120	5
YUV to AVC	4k	4	0	0	3	96	267	17
YUV to HEVC	4k	4	0	0	3	99	88	4
YUV to AV1	4k	4	0	0	3	99	64	3
YUV to AVC	4k	4	0	1	1	98	164	7
YUV to HEVC	4k	4	0	1	1	98	208	8
YUV to AVC	4k	4	0	1	2	98	164	9
YUV to HEVC	4k	4	0	1	2	100	96	4
YUV to AVC	4k	4	0	1	3	98	164	7
YUV to HEVC	4k	4	0	1	3	99	60	3
YUV to AVC	4k	4	4	0	1	99	172	9
YUV to HEVC	4k	4	4	0	1	99	197	9
YUV to AV1	4k	4	4	0	1	99	172	7
YUV to AVC	4k	4	4	0	2	99	172	7
YUV to HEVC	4k	4	4	0	2	99	120	5
YUV to AV1	4k	4	4	0	2	99	100	5
YUV to AVC	4k	4	4	0	3	99	172	9
YUV to HEVC	4k	4	4	0	3	100	76	5
YUV to AV1	4k	4	4	0	3	100	64	3
YUV to AVC	4k	4	4	1	1	99	120	6
YUV to HEVC	4k	4	4	1	1	99	156	7
YUV to AVC	4k	4	4	1	2	99	120	5
YUV to HEVC	4k	4	4	1	2	101	83	5
YUV to AVC	4k	4	4	1	3	100	120	6
YUV to HEVC	4k	4	4	1	3	99	56	3
YUV to AVC	4k	4	16	0	1	100	172	7
YUV to HEVC	4k	4	16	0	1	100	196	11
YUV to AV1	4k	4	16	0	1	99	172	10
YUV to AVC	4k	4	16	0	2	100	172	8
YUV to HEVC	4k	4	16	0	2	100	120	6
YUV to AV1	4k	4	16	0	2	99	100	5
YUV to AVC	4k	4	16	0	3	99	172	8
YUV to HEVC	4k	4	16	0	3	99	76	5
YUV to AV1	4k	4	16	0	3	100	64	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	16	1	1	100	120	6
YUV to HEVC	4k	4	16	1	1	100	156	8
YUV to AVC	4k	4	16	1	2	101	120	12
YUV to HEVC	4k	4	16	1	2	98	80	3
YUV to AVC	4k	4	16	1	3	99	120	5
YUV to HEVC	4k	4	16	1	3	100	56	2
YUV to AVC	4k	4	40	0	1	100	172	9
YUV to HEVC	4k	4	40	0	1	99	196	9
YUV to AV1	4k	4	40	0	1	100	172	10
YUV to AVC	4k	4	40	0	2	99	170	7
YUV to HEVC	4k	4	40	0	2	100	116	5
YUV to AV1	4k	4	40	0	2	99	100	4
YUV to AVC	4k	4	40	0	3	99	172	10
YUV to HEVC	4k	4	40	0	3	99	76	4
YUV to AV1	4k	4	40	0	3	100	60	3
YUV to AVC	4k	4	40	1	1	100	120	7
YUV to HEVC	4k	4	40	1	1	100	154	8
YUV to AVC	4k	4	40	1	2	99	120	5
YUV to HEVC	4k	4	40	1	2	99	80	4
YUV to AVC	4k	4	40	1	3	100	120	5
YUV to HEVC	4k	4	40	1	3	101	56	4
YUV to AVC	1080p	20	0	0	1	99	1120	3
YUV to HEVC	1080p	20	0	0	1	99	1182	4
YUV to AV1	1080p	20	0	0	1	99	1020	3
YUV to AVC	1080p	20	0	0	2	99	1120	3
YUV to HEVC	1080p	20	0	0	2	99	600	1
YUV to AV1	1080p	20	0	0	2	99	481	2
YUV to AVC	1080p	20	0	0	3	99	1120	3
YUV to HEVC	1080p	20	0	0	3	99	360	1
YUV to AV1	1080p	20	0	0	3	100	260	1
YUV to AVC	1080p	20	0	1	1	99	660	2
YUV to HEVC	1080p	20	0	1	1	99	840	2
YUV to AVC	1080p	20	0	1	2	99	660	2
YUV to HEVC	1080p	20	0	1	2	99	380	1
YUV to AVC	1080p	20	0	1	3	99	660	2
YUV to HEVC	1080p	20	0	1	3	100	240	1
YUV to AVC	1080p	20	4	0	1	99	620	2
YUV to HEVC	1080p	20	4	0	1	99	719	2
YUV to AV1	1080p	20	4	0	1	99	621	2
YUV to AVC	1080p	20	4	0	2	99	616	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	4	0	2	99	444	1
YUV to AV1	1080p	20	4	0	2	99	369	1
YUV to AVC	1080p	20	4	0	3	99	614	2
YUV to HEVC	1080p	20	4	0	3	99	300	1
YUV to AV1	1080p	20	4	0	3	100	238	1
YUV to AVC	1080p	20	4	1	1	99	440	1
YUV to HEVC	1080p	20	4	1	1	99	561	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	441	1
YUV to HEVC	1080p	20	4	1	3	100	220	1
YUV to AVC	1080p	20	16	0	1	99	601	1
YUV to HEVC	1080p	20	16	0	1	99	712	2
YUV to AV1	1080p	20	16	0	1	99	621	2
YUV to AVC	1080p	20	16	0	2	99	602	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	300	1
YUV to AV1	1080p	20	16	0	3	99	236	1
YUV to AVC	1080p	20	16	1	1	99	440	1
YUV to HEVC	1080p	20	16	1	1	99	560	1
YUV to AVC	1080p	20	16	1	2	99	440	1
YUV to HEVC	1080p	20	16	1	2	99	315	1
YUV to AVC	1080p	20	16	1	3	99	440	1
YUV to HEVC	1080p	20	16	1	3	99	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	99	360	1
YUV to AVC	1080p	20	40	0	3	99	600	1
YUV to HEVC	1080p	20	40	0	3	99	288	1
YUV to AV1	1080p	20	40	0	3	99	225	1
YUV to AVC	1080p	20	40	1	1	99	440	1
YUV to HEVC	1080p	20	40	1	1	99	560	2
YUV to AVC	1080p	20	40	1	2	99	438	1
YUV to HEVC	1080p	20	40	1	2	99	301	1
YUV to AVC	1080p	20	40	1	3	99	439	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	40	1	3	100	202	1
YUV to AVC	720p	40	0	0	1	98	2289	2
YUV to HEVC	720p	40	0	0	1	95	2321	3
YUV to AV1	720p	40	0	0	1	97	1963	1
YUV to AVC	720p	40	0	0	2	98	2287	2
YUV to HEVC	720p	40	0	0	2	99	1320	1
YUV to AV1	720p	40	0	0	2	99	1040	0
YUV to AVC	720p	40	0	0	3	96	2282	2
YUV to HEVC	720p	40	0	0	3	99	800	1
YUV to AV1	720p	40	0	0	3	99	560	0
YUV to AVC	720p	40	0	1	1	99	1480	1
YUV to HEVC	720p	40	0	1	1	99	1800	1
YUV to AVC	720p	40	0	1	2	99	1480	1
YUV to HEVC	720p	40	0	1	2	99	840	0
YUV to AVC	720p	40	0	1	3	99	1480	1
YUV to HEVC	720p	40	0	1	3	100	529	0
YUV to AVC	720p	40	4	0	1	99	1244	1
YUV to HEVC	720p	40	4	0	1	100	1291	1
YUV to AV1	720p	40	4	0	1	100	961	1
YUV to AVC	720p	40	4	0	2	99	1244	1
YUV to HEVC	720p	40	4	0	2	99	920	0
YUV to AV1	720p	40	4	0	2	99	760	1
YUV to AVC	720p	40	4	0	3	99	1243	1
YUV to HEVC	720p	40	4	0	3	99	640	0
YUV to AV1	720p	40	4	0	3	100	480	0
YUV to AVC	720p	40	4	1	1	99	920	1
YUV to HEVC	720p	40	4	1	1	99	1160	0
YUV to AVC	720p	40	4	1	2	99	920	1
YUV to HEVC	720p	40	4	1	2	99	651	0
YUV to AVC	720p	40	4	1	3	99	920	1
YUV to HEVC	720p	40	4	1	3	100	441	0
YUV to AVC	720p	40	16	0	1	99	1240	1
YUV to HEVC	720p	40	16	0	1	100	1294	1
YUV to AV1	720p	40	16	0	1	100	1015	1
YUV to AVC	720p	40	16	0	2	99	1240	1
YUV to HEVC	720p	40	16	0	2	99	920	1
YUV to AV1	720p	40	16	0	2	99	760	0
YUV to AVC	720p	40	16	0	3	99	1240	1
YUV to HEVC	720p	40	16	0	3	99	640	0
YUV to AV1	720p	40	16	0	3	99	480	0

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	40	16	1	1	99	920	1
YUV to HEVC	720p	40	16	1	1	99	1160	1
YUV to AVC	720p	40	16	1	2	99	920	1
YUV to HEVC	720p	40	16	1	2	99	640	0
YUV to AVC	720p	40	16	1	3	99	920	0
YUV to HEVC	720p	40	16	1	3	99	440	0
YUV to AVC	720p	40	40	0	1	99	1240	1
YUV to HEVC	720p	40	40	0	1	100	1262	1
YUV to AV1	720p	40	40	0	1	100	1000	0
YUV to AVC	720p	40	40	0	2	99	1240	1
YUV to HEVC	720p	40	40	0	2	99	920	1
YUV to AV1	720p	40	40	0	2	99	759	0
YUV to AVC	720p	40	40	0	3	99	1240	1
YUV to HEVC	720p	40	40	0	3	99	617	0
YUV to AV1	720p	40	40	0	3	99	480	0
YUV to AVC	720p	40	40	1	1	99	919	0
YUV to HEVC	720p	40	40	1	1	99	1132	1
YUV to AVC	720p	40	40	1	2	99	920	0
YUV to HEVC	720p	40	40	1	2	99	640	0
YUV to AVC	720p	40	40	1	3	99	919	1
YUV to HEVC	720p	40	40	1	3	99	440	0

24. T1U – Capped CRF

24.1 Encoding with lookaheadDepth

24.1.1 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.

24.1.2 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

24.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	19	99	618	2
YUV to HEVC	1080p	20	0	0	1	19	99	717	2
YUV to AV1	1080p	20	0	0	1	19	99	624	2
YUV to AVC	1080p	20	0	0	2	19	99	617	2
YUV to HEVC	1080p	20	0	0	2	19	99	440	1
YUV to AV1	1080p	20	0	0	2	19	100	380	1
YUV to AVC	1080p	20	0	0	3	19	99	620	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	100	440	1
YUV to HEVC	1080p	20	0	1	1	19	99	574	1
YUV to AVC	1080p	20	0	1	2	19	99	440	1
YUV to HEVC	1080p	20	0	1	2	19	99	320	1
YUV to AVC	1080p	20	0	1	3	19	99	440	1
YUV to HEVC	1080p	20	0	1	3	19	100	220	1
YUV to AVC	1080p	20	4	0	1	19	99	611	2
YUV to HEVC	1080p	20	4	0	1	19	99	714	2
YUV to AV1	1080p	20	4	0	1	19	99	620	2
YUV to AVC	1080p	20	4	0	2	19	99	615	2
YUV to HEVC	1080p	20	4	0	2	19	100	440	1
YUV to AV1	1080p	20	4	0	2	19	99	360	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	3	19	99	610	2
YUV to HEVC	1080p	20	4	0	3	19	99	300	1
YUV to AV1	1080p	20	4	0	3	19	99	240	1
YUV to AVC	1080p	20	4	1	1	19	99	440	1
YUV to HEVC	1080p	20	4	1	1	19	100	560	2
YUV to AVC	1080p	20	4	1	2	19	99	440	1
YUV to HEVC	1080p	20	4	1	2	19	100	320	1
YUV to AVC	1080p	20	4	1	3	19	99	440	1
YUV to HEVC	1080p	20	4	1	3	19	99	220	1
YUV to AVC	1080p	20	16	0	1	19	99	603	2
YUV to HEVC	1080p	20	16	0	1	19	99	710	2
YUV to AV1	1080p	20	16	0	1	19	99	620	2
YUV to AVC	1080p	20	16	0	2	19	99	601	2
YUV to HEVC	1080p	20	16	0	2	19	99	440	1
YUV to AV1	1080p	20	16	0	2	19	99	360	1
YUV to AVC	1080p	20	16	0	3	19	99	604	2
YUV to HEVC	1080p	20	16	0	3	19	100	300	1
YUV to AV1	1080p	20	16	0	3	19	100	238	1
YUV to AVC	1080p	20	16	1	1	19	100	440	1
YUV to HEVC	1080p	20	16	1	1	19	99	560	2
YUV to AVC	1080p	20	16	1	2	19	99	440	1
YUV to HEVC	1080p	20	16	1	2	19	99	317	1
YUV to AVC	1080p	20	16	1	3	19	99	440	1
YUV to HEVC	1080p	20	16	1	3	19	99	218	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	99	360	1
YUV to AVC	1080p	20	40	0	3	19	99	600	2
YUV to HEVC	1080p	20	40	0	3	19	99	286	1
YUV to AV1	1080p	20	40	0	3	19	100	223	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	99	440	2
YUV to HEVC	1080p	20	40	1	3	19	100	202	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	718	2
YUV to AV1	1080p	20	0	0	1	23	99	624	2
YUV to AVC	1080p	20	0	0	2	23	99	619	2
YUV to HEVC	1080p	20	0	0	2	23	100	441	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	99	300	1
YUV to AV1	1080p	20	0	0	3	23	99	240	1
YUV to AVC	1080p	20	0	1	1	23	100	440	1
YUV to HEVC	1080p	20	0	1	1	23	99	567	1
YUV to AVC	1080p	20	0	1	2	23	100	440	1
YUV to HEVC	1080p	20	0	1	2	23	99	320	1
YUV to AVC	1080p	20	0	1	3	23	99	440	2
YUV to HEVC	1080p	20	0	1	3	23	99	220	1
YUV to AVC	1080p	20	4	0	1	23	99	614	1
YUV to HEVC	1080p	20	4	0	1	23	99	719	2
YUV to AV1	1080p	20	4	0	1	23	99	620	2
YUV to AVC	1080p	20	4	0	2	23	99	613	2
YUV to HEVC	1080p	20	4	0	2	23	99	440	1
YUV to AV1	1080p	20	4	0	2	23	100	366	1
YUV to AVC	1080p	20	4	0	3	23	99	613	2
YUV to HEVC	1080p	20	4	0	3	23	99	300	1
YUV to AV1	1080p	20	4	0	3	23	100	239	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	100	440	1
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	100	220	1
YUV to AVC	1080p	20	16	0	1	23	99	604	2
YUV to HEVC	1080p	20	16	0	1	23	99	713	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	100	440	1
YUV to AV1	1080p	20	16	0	2	23	99	361	1
YUV to AVC	1080p	20	16	0	3	23	99	600	2
YUV to HEVC	1080p	20	16	0	3	23	100	297	1
YUV to AV1	1080p	20	16	0	3	23	100	235	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	16	1	1	23	100	440	1
YUV to HEVC	1080p	20	16	1	1	23	99	560	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	99	215	1
YUV to AVC	1080p	20	40	0	1	23	99	601	2
YUV to HEVC	1080p	20	40	0	1	23	99	700	2
YUV to AV1	1080p	20	40	0	1	23	99	618	1
YUV to AVC	1080p	20	40	0	2	23	99	600	2
YUV to HEVC	1080p	20	40	0	2	23	100	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	281	1
YUV to AV1	1080p	20	40	0	3	23	100	225	1
YUV to AVC	1080p	20	40	1	1	23	100	440	2
YUV to HEVC	1080p	20	40	1	1	23	99	560	1
YUV to AVC	1080p	20	40	1	2	23	99	440	1
YUV to HEVC	1080p	20	40	1	2	23	100	300	1
YUV to AVC	1080p	20	40	1	3	23	99	439	1
YUV to HEVC	1080p	20	40	1	3	23	100	201	1
YUV to AVC	1080p	20	0	0	1	27	99	619	2
YUV to HEVC	1080p	20	0	0	1	27	99	719	2
YUV to AV1	1080p	20	0	0	1	27	99	627	1
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	618	2
YUV to HEVC	1080p	20	0	0	3	27	100	300	1
YUV to AV1	1080p	20	0	0	3	27	100	240	1
YUV to AVC	1080p	20	0	1	1	27	99	440	1
YUV to HEVC	1080p	20	0	1	1	27	99	573	1
YUV to AVC	1080p	20	0	1	2	27	99	440	1
YUV to HEVC	1080p	20	0	1	2	27	99	320	1
YUV to AVC	1080p	20	0	1	3	27	99	440	1
YUV to HEVC	1080p	20	0	1	3	27	99	220	1
YUV to AVC	1080p	20	4	0	1	27	99	613	2
YUV to HEVC	1080p	20	4	0	1	27	99	716	2
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	613	2
YUV to HEVC	1080p	20	4	0	2	27	99	440	1
YUV to AV1	1080p	20	4	0	2	27	100	367	1
YUV to AVC	1080p	20	4	0	3	27	99	616	2
YUV to HEVC	1080p	20	4	0	3	27	99	300	1
YUV to AV1	1080p	20	4	0	3	27	99	239	1
YUV to AVC	1080p	20	4	1	1	27	100	440	1
YUV to HEVC	1080p	20	4	1	1	27	99	565	2
YUV to AVC	1080p	20	4	1	2	27	99	440	1
YUV to HEVC	1080p	20	4	1	2	27	99	320	1
YUV to AVC	1080p	20	4	1	3	27	99	440	1
YUV to HEVC	1080p	20	4	1	3	27	100	220	1
YUV to AVC	1080p	20	16	0	1	27	99	603	2
YUV to HEVC	1080p	20	16	0	1	27	99	711	2
YUV to AV1	1080p	20	16	0	1	27	99	620	2
YUV to AVC	1080p	20	16	0	2	27	99	602	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	606	2
YUV to HEVC	1080p	20	16	0	3	27	99	299	1
YUV to AV1	1080p	20	16	0	3	27	99	235	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	311	1
YUV to AVC	1080p	20	16	1	3	27	99	440	1
YUV to HEVC	1080p	20	16	1	3	27	99	218	1
YUV to AVC	1080p	20	40	0	1	27	99	600	2
YUV to HEVC	1080p	20	40	0	1	27	99	700	2
YUV to AV1	1080p	20	40	0	1	27	99	619	2
YUV to AVC	1080p	20	40	0	2	27	99	600	2
YUV to HEVC	1080p	20	40	0	2	27	99	440	1
YUV to AV1	1080p	20	40	0	2	27	99	360	1
YUV to AVC	1080p	20	40	0	3	27	99	600	2
YUV to HEVC	1080p	20	40	0	3	27	99	283	1
YUV to AV1	1080p	20	40	0	3	27	100	226	1
YUV to AVC	1080p	20	40	1	1	27	99	440	1
YUV to HEVC	1080p	20	40	1	1	27	99	560	1
YUV to AVC	1080p	20	40	1	2	27	100	440	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	40	1	2	27	99	300	1
YUV to AVC	1080p	20	40	1	3	27	99	440	1
YUV to HEVC	1080p	20	40	1	3	27	100	202	1

25. T1U – Inplace Overlay

25.1 Transcoding

25.1.1 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.

25.1.2 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

25.2 Inplace Overlay Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	1	14	21	4	254	8
AVC to HEVC	1	15	20	5	270	9
AVC to AV1	1	13	20	4	231	9
HEVC to AVC	1	14	20	4	253	11
HEVC to HEVC	1	15	20	4	269	12
HEVC to AV1	1	13	21	4	230	11
VP9 to AVC	1	18	20	4	253	8
VP9 to HEVC	1	20	20	4	269	9
VP9 to AV1	1	17	21	4	230	8
AVC to AVC	16	71	92	24	960	2
AVC to HEVC	16	77	92	27	1070	2
AVC to AV1	16	68	92	24	992	2
HEVC to AVC	16	71	92	24	992	3
HEVC to HEVC	16	77	92	27	1088	3
HEVC to AV1	16	68	95	25	1007	3
VP9 to AVC	16	89	95	24	970	2
VP9 to HEVC	16	93	90	25	1053	2
VP9 to AV1	16	85	92	24	1008	2
AVC to AVC	32	72	92	23	864	1
AVC to HEVC	32	80	92	26	992	1
AVC to AV1	32	75	94	25	949	1
HEVC to AVC	32	73	92	24	910	1
HEVC to HEVC	32	79	93	27	1024	1
HEVC to AV1	32	72	94	25	960	1
VP9 to AVC	32	86	93	23	917	1
VP9 to HEVC	32	94	92	26	1022	1
VP9 to AV1	32	87	93	24	991	1

26. 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'	√	√	×	√	√	√	×

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

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.

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

27.1.2 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

27.2 Encoding

27.2.1 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.

27.2.2 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)

27.3 Transcoding

27.3.1 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.

27.3.2 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)

27.4 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	10	0	158	37
HEVC to YUV	4k	1	0	8	1	10	0	158	44
VP9 to YUV	4k	1	0	8	1	24	0	83	20
AVC to YUV	4k	5	0	8	0	83	0	250	17
HEVC to YUV	4k	5	0	8	0	92	0	269	13
VP9 to YUV	4k	5	0	8	0	94	0	266	11
YUV to AVC	4k	1	0	8	1	0	92	184	60
YUV to HEVC	4k	1	0	8	1	0	87	191	56
YUV to AV1	4k	1	0	8	1	0	93	179	60
YUV to AVC	4k	3	0	8	0	0	71	150	48
YUV to HEVC	4k	3	0	8	0	0	73	162	42
YUV to AV1	4k	3	0	8	0	0	73	138	37
YUV to AVC	4k	5	0	8	0	0	98	192	28
YUV to HEVC	4k	5	0	8	0	0	98	213	30
YUV to AV1	4k	5	0	8	0	0	97	187	28
AVC to AVC	4k	1	1	8	1	14	96	127	39
AVC to HEVC	4k	1	1	8	1	15	96	158	43
AVC to AV1	4k	1	1	8	1	13	94	154	40
HEVC to AVC	4k	1	1	8	1	15	96	126	39
HEVC to HEVC	4k	1	1	8	1	13	89	154	38
HEVC to AV1	4k	1	1	8	1	12	97	148	36
VP9 to AVC	4k	1	1	8	1	24	39	83	23
VP9 to HEVC	4k	1	1	8	1	24	38	83	27
VP9 to AV1	4k	1	1	8	1	24	44	83	23
AVC to AVC	4k	3	1	8	0	45	72	120	31
AVC to HEVC	4k	3	1	8	0	50	73	149	32
AVC to AV1	4k	3	1	8	0	41	73	129	24
HEVC to AVC	4k	3	1	8	0	41	72	120	29
HEVC to HEVC	4k	3	1	8	0	41	72	150	34
HEVC to AV1	4k	3	1	8	0	34	73	134	26
VP9 to AVC	4k	3	1	8	0	45	72	123	27
VP9 to HEVC	4k	3	1	8	0	49	72	147	29
VP9 to AV1	4k	3	1	8	0	41	72	135	22
AVC to AVC	4k	5	1	8	0	66	96	120	21
AVC to HEVC	4k	5	1	8	0	71	98	157	23
AVC to AV1	4k	5	1	8	0	65	97	157	17
HEVC to AVC	4k	5	1	8	0	55	96	121	21
HEVC to HEVC	4k	5	1	8	0	59	97	160	22
HEVC to AV1	4k	5	1	8	0	53	96	161	24

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AVC	4k	5	1	8	0	61	98	122	20
VP9 to HEVC	4k	5	1	8	0	67	97	160	24
VP9 to AV1	4k	5	1	8	0	61	97	159	18
AVC to YUV	4k	1	0	10	1	8	0	101	28
HEVC to YUV	4k	1	0	10	1	8	0	100	26
VP9 to YUV	4k	1	0	10	1	24	0	83	23
AVC to YUV	4k	16	0	10	0	48	0	135	9
HEVC to YUV	4k	16	0	10	0	72	0	135	8
VP9 to YUV	4k	16	0	10	0	62	0	134	9
YUV to AVC	4k	1	0	10	1	0	51	107	55
YUV to HEVC	4k	1	0	10	1	0	49	109	52
YUV to AV1	4k	1	0	10	1	0	55	106	56
YUV to AVC	4k	3	0	10	0	0	62	117	57
YUV to HEVC	4k	3	0	10	0	0	53	119	58
YUV to AV1	4k	3	0	10	0	0	60	117	60
AVC to YUV	1080p	1	0	8	1	18	0	424	52
HEVC to YUV	1080p	1	0	8	1	14	0	412	71
VP9 to YUV	1080p	1	0	8	1	23	0	211	29
AVC to YUV	1080p	18	0	8	0	92	0	965	8
HEVC to YUV	1080p	18	0	8	0	94	0	1005	11
VP9 to YUV	1080p	18	0	8	0	97	0	878	8
YUV to AVC	1080p	1	0	8	1	0	57	448	84
YUV to HEVC	1080p	1	0	8	1	0	54	446	72
YUV to AV1	1080p	1	0	8	1	0	53	398	83
YUV to AVC	1080p	20	0	8	0	0	99	820	12
YUV to HEVC	1080p	20	0	8	0	0	99	860	14
YUV to AV1	1080p	20	0	8	0	0	99	740	12
AVC to AVC	1080p	1	1	8	1	36	74	542	65
AVC to HEVC	1080p	1	1	8	1	36	71	588	73
AVC to AV1	1080p	1	1	8	1	30	75	548	80
HEVC to AVC	1080p	1	1	8	1	34	77	521	86
HEVC to HEVC	1080p	1	1	8	1	31	73	580	84
HEVC to AV1	1080p	1	1	8	1	28	80	529	73
VP9 to AVC	1080p	1	1	8	1	23	25	210	32
VP9 to HEVC	1080p	1	1	8	1	23	24	209	26
VP9 to AV1	1080p	1	1	8	1	22	28	209	36
AVC to AVC	1080p	20	1	8	0	73	99	570	8
AVC to HEVC	1080p	20	1	8	0	80	99	657	11
AVC to AV1	1080p	20	1	8	0	75	99	640	11
HEVC to AVC	1080p	20	1	8	0	68	99	600	9

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to HEVC	1080p	20	1	8	0	70	99	681	9
HEVC to AV1	1080p	20	1	8	0	66	99	661	8
VP9 to AVC	1080p	20	1	8	0	79	99	580	9
VP9 to HEVC	1080p	20	1	8	0	88	99	660	9
VP9 to AV1	1080p	20	1	8	0	83	99	651	7
AVC to YUV	1080p	1	0	10	1	14	0	318	40
HEVC to YUV	1080p	1	0	10	1	13	0	312	36
VP9 to YUV	1080p	1	0	10	1	22	0	299	45
AVC to YUV	1080p	18	0	10	0	56	0	541	4
HEVC to YUV	1080p	18	0	10	0	57	0	541	3
VP9 to YUV	1080p	18	0	10	0	53	0	540	4
YUV to AVC	1080p	1	0	10	1	0	36	306	63
YUV to HEVC	1080p	1	0	10	1	0	34	306	74
YUV to AV1	1080p	1	0	10	1	0	38	289	64
YUV to AVC	1080p	20	0	10	0	0	57	464	12
YUV to HEVC	1080p	20	0	10	0	0	53	463	14
YUV to AV1	1080p	20	0	10	0	0	59	460	14
AVC to YUV	720p	1	0	8	1	24	0	655	59
HEVC to YUV	720p	1	0	8	1	21	0	631	75
VP9 to YUV	720p	1	0	8	1	21	0	371	39
AVC to YUV	720p	40	0	8	0	100	0	1902	6
HEVC to YUV	720p	40	0	8	0	91	0	1907	6
VP9 to YUV	720p	40	0	8	0	94	0	1615	4
YUV to AVC	720p	1	0	8	1	0	30	562	69
YUV to HEVC	720p	1	0	8	1	0	29	562	57
YUV to AV1	720p	1	0	8	1	0	31	501	63
YUV to AVC	720p	40	0	8	0	0	88	1621	8
YUV to HEVC	720p	40	0	8	0	0	91	1680	8
YUV to AV1	720p	40	0	8	0	0	91	1469	6
AVC to AVC	720p	1	1	8	1	35	44	875	75
AVC to HEVC	720p	1	1	8	1	38	48	889	78
AVC to AV1	720p	1	1	8	1	32	51	776	70
HEVC to AVC	720p	1	1	8	1	33	45	840	80
HEVC to HEVC	720p	1	1	8	1	29	42	842	81
HEVC to AV1	720p	1	1	8	1	29	50	749	77
VP9 to AVC	720p	1	1	8	1	21	19	368	38
VP9 to HEVC	720p	1	1	8	1	21	19	367	39
VP9 to AV1	720p	1	1	8	1	21	23	369	44
AVC to AVC	720p	40	1	8	0	91	100	1292	2
AVC to HEVC	720p	40	1	8	0	93	100	1416	3

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	720p	40	1	8	0	75	100	1289	6
HEVC to AVC	720p	40	1	8	0	84	100	1297	5
HEVC to HEVC	720p	40	1	8	0	85	100	1418	6
HEVC to AV1	720p	40	1	8	0	68	100	1291	6
VP9 to AVC	720p	40	1	8	0	100	95	1257	4
VP9 to HEVC	720p	40	1	8	0	100	93	1367	5
VP9 to AV1	720p	40	1	8	0	91	100	1290	5

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

28.1 Decoding

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.

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

28.1.2 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)

28.2 Encoding

28.2.1 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.

28.2.2 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)

28.3 Transcoding

28.3.1 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.

28.3.2 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)

28.4 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	10	0	155	30
HEVC to YUV	4k	1	0	8	1	9	0	158	26
VP9 to YUV	4k	1	0	8	1	24	0	82	16
AVC to YUV	4k	5	0	8	0	84	0	250	11
HEVC to YUV	4k	5	0	8	0	81	0	263	9
VP9 to YUV	4k	5	0	8	0	92	0	262	8
YUV to AVC	4k	1	0	8	1	0	57	122	48
YUV to HEVC	4k	1	0	8	1	0	54	120	50
YUV to AV1	4k	1	0	8	1	0	60	111	49
YUV to AVC	4k	3	0	8	0	0	73	152	32
YUV to HEVC	4k	3	0	8	0	0	73	161	32
YUV to AV1	4k	3	0	8	0	0	73	140	30
YUV to AVC	4k	5	0	8	0	0	98	186	26
YUV to HEVC	4k	5	0	8	0	0	98	216	26
YUV to AV1	4k	5	0	8	0	0	98	187	26
AVC to AVC	4k	1	1	8	0	17	97	128	49
AVC to HEVC	4k	1	1	8	0	16	96	163	51
AVC to AV1	4k	1	1	8	0	16	96	159	54
HEVC to AVC	4k	1	1	8	0	17	97	128	46
HEVC to HEVC	4k	1	1	8	0	16	97	159	49
HEVC to AV1	4k	1	1	8	0	15	97	152	48
VP9 to AVC	4k	1	1	8	0	24	39	82	37
VP9 to HEVC	4k	1	1	8	0	24	37	82	39
VP9 to AV1	4k	1	1	8	0	24	43	82	38
AVC to AVC	4k	3	1	8	0	48	72	122	28
AVC to HEVC	4k	3	1	8	0	47	72	153	53
AVC to AV1	4k	3	1	8	0	40	72	136	54
HEVC to AVC	4k	3	1	8	0	39	72	119	24
HEVC to HEVC	4k	3	1	8	0	40	71	145	24
HEVC to AV1	4k	3	1	8	0	34	72	133	27
VP9 to AVC	4k	3	1	8	0	46	72	122	37
VP9 to HEVC	4k	3	1	8	0	48	72	148	27
VP9 to AV1	4k	3	1	8	0	40	72	134	24
AVC to AVC	4k	5	1	8	0	65	97	122	12
AVC to HEVC	4k	5	1	8	0	71	98	159	22
AVC to AV1	4k	5	1	8	0	64	97	159	11
HEVC to AVC	4k	5	1	8	0	57	98	123	23
HEVC to HEVC	4k	5	1	8	0	59	97	159	11

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to AV1	4k	5	1	8	0	56	97	158	11
VP9 to AVC	4k	5	1	8	0	61	97	123	11
VP9 to HEVC	4k	5	1	8	0	68	97	160	11
VP9 to AV1	4k	5	1	8	0	61	97	159	11
AVC to YUV	4k	1	0	10	1	7	0	93	22
HEVC to YUV	4k	1	0	10	1	7	0	92	20
VP9 to YUV	4k	1	0	10	1	24	0	82	20
AVC to YUV	4k	5	0	10	0	56	0	133	7
HEVC to YUV	4k	5	0	10	0	56	0	135	7
VP9 to YUV	4k	5	0	10	0	64	0	134	7
YUV to AVC	4k	1	0	10	1	0	35	66	49
YUV to HEVC	4k	1	0	10	1	0	29	66	50
YUV to AV1	4k	1	0	10	1	0	32	64	48
YUV to AVC	4k	3	0	10	0	0	40	88	38
YUV to HEVC	4k	3	0	10	0	0	43	90	38
YUV to AV1	4k	3	0	10	0	0	44	84	36
AVC to YUV	1080p	1	0	8	1	16	0	390	38
HEVC to YUV	1080p	1	0	8	1	15	0	414	36
VP9 to YUV	1080p	1	0	8	1	23	0	211	17
AVC to YUV	1080p	20	0	8	0	88	0	932	6
HEVC to YUV	1080p	20	0	8	0	89	0	969	5
VP9 to YUV	1080p	20	0	8	0	98	0	890	3
YUV to AVC	1080p	1	0	8	1	0	39	348	45
YUV to HEVC	1080p	1	0	8	1	0	37	332	44
YUV to AV1	1080p	1	0	8	1	0	37	291	36
YUV to AVC	1080p	20	0	8	0	0	99	828	10
YUV to HEVC	1080p	20	0	8	0	0	99	882	11
YUV to AV1	1080p	20	0	8	0	0	99	764	10
AVC to AVC	1080p	1	1	8	0	39	79	597	60
AVC to HEVC	1080p	1	1	8	0	37	75	649	62
AVC to AV1	1080p	1	1	8	0	35	83	602	64
HEVC to AVC	1080p	1	1	8	0	40	87	590	57
HEVC to HEVC	1080p	1	1	8	0	34	84	653	61
HEVC to AV1	1080p	1	1	8	0	30	86	597	66
VP9 to AVC	1080p	1	1	8	0	23	25	211	36
VP9 to HEVC	1080p	1	1	8	0	23	24	211	36
VP9 to AV1	1080p	1	1	8	0	23	28	211	42
AVC to AVC	1080p	20	1	8	0	73	99	585	6
AVC to HEVC	1080p	20	1	8	0	79	99	667	5

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	1080p	20	1	8	0	73	99	652	6
HEVC to AVC	1080p	20	1	8	0	67	99	611	6
HEVC to HEVC	1080p	20	1	8	0	70	99	697	6
HEVC to AV1	1080p	20	1	8	0	65	99	667	6
VP9 to AVC	1080p	20	1	8	0	75	99	591	5
VP9 to HEVC	1080p	20	1	8	0	87	99	678	3
VP9 to AV1	1080p	20	1	8	0	83	99	662	4
AVC to YUV	1080p	1	0	10	1	15	0	320	27
HEVC to YUV	1080p	1	0	10	1	14	0	312	25
VP9 to YUV	1080p	1	0	10	1	21	0	297	25
AVC to YUV	1080p	20	0	10	0	52	0	543	2
HEVC to YUV	1080p	20	0	10	0	62	0	538	2
VP9 to YUV	1080p	20	0	10	0	100	0	521	2
YUV to AVC	1080p	1	0	10	1	0	24	202	49
YUV to HEVC	1080p	1	0	10	1	0	23	219	46
YUV to AV1	1080p	1	0	10	1	0	26	193	46
YUV to AVC	1080p	20	0	10	0	0	57	469	12
YUV to HEVC	1080p	20	0	10	0	0	52	473	12
YUV to AV1	1080p	20	0	10	0	0	59	454	12
AVC to YUV	720p	1	0	8	1	21	0	584	40
HEVC to YUV	720p	1	0	8	1	21	0	638	36
VP9 to YUV	720p	1	0	8	1	21	0	377	18
AVC to YUV	720p	40	0	8	0	96	0	1914	3
HEVC to YUV	720p	40	0	8	0	78	0	1740	2
VP9 to YUV	720p	40	0	8	0	89	0	1561	1
YUV to AVC	720p	1	0	8	1	0	28	536	30
YUV to HEVC	720p	1	0	8	1	0	28	533	33
YUV to AV1	720p	1	0	8	1	0	25	418	43
YUV to AVC	720p	40	0	8	0	0	93	1758	6
YUV to HEVC	720p	40	0	8	0	0	94	1811	6
YUV to AV1	720p	40	0	8	0	0	95	1584	5
AVC to AVC	720p	1	1	8	0	37	48	1010	42
AVC to HEVC	720p	1	1	8	0	50	61	1020	45
AVC to AV1	720p	1	1	8	0	32	52	908	37
HEVC to AVC	720p	1	1	8	0	31	45	865	66
HEVC to HEVC	720p	1	1	8	0	32	47	965	40
HEVC to AV1	720p	1	1	8	0	31	53	823	60
VP9 to AVC	720p	1	1	8	0	21	20	376	37
VP9 to HEVC	720p	1	1	8	0	21	20	376	38

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	720p	1	1	8	0	21	24	376	49
AVC to AVC	720p	40	1	8	0	100	100	1346	1
AVC to HEVC	720p	40	1	8	0	100	100	1475	1
AVC to AV1	720p	40	1	8	0	96	100	1335	1
HEVC to AVC	720p	40	1	8	0	100	100	1350	0
HEVC to HEVC	720p	40	1	8	0	100	100	1478	0
HEVC to AV1	720p	40	1	8	0	93	100	1337	0
VP9 to AVC	720p	40	1	8	0	100	99	1299	1
VP9 to HEVC	720p	40	1	8	0	100	98	1409	1
VP9 to AV1	720p	40	1	8	0	100	100	1342	0

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

29.1 Encoding

29.1.1 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.

29.1.2 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

29.2 FFmpeg Latency Performance Results

TYPE	RESOLUTION	JOBS	ELAT_AVG (ms)	ELAT_MAX (ms)	ELAT_MIN (ms)	ELAT_VAR (ms)
YUV to AVC	4k	1	24.72	28.01	24.08	0.14
YUV to HEVC	4k	1	26.72	30.23	24.97	0.24
YUV to AV1	4k	1	34.79	40.94	26.04	1.64
YUV to AVC	4k	3	25.69	28.41	24.16	1.08
YUV to HEVC	4k	3	26.94	33.46	25.23	0.38
YUV to AV1	4k	3	34.95	41.11	26.34	1.55
YUV to AVC	4k	5	26.49	31.86	24.36	2.49
YUV to HEVC	4k	5	33.81	58.5	25.37	67.41
YUV to AV1	4k	5	43.76	77.93	26.35	133.09
YUV to AVC	1080p	1	7.55	8.77	7.07	0.05
YUV to HEVC	1080p	1	8.01	9.06	7.39	0.06
YUV to AV1	1080p	1	10.68	12.02	7.71	0.14
YUV to AVC	1080p	20	10.92	18.92	7.18	4.34
YUV to HEVC	1080p	20	11.45	19.01	7.46	5.33
YUV to AV1	1080p	20	39.99	50.12	28.17	13.79
YUV to AVC	720p	1	4.45	5.79	3.81	0.06
YUV to HEVC	720p	1	4.61	5.08	3.68	0.04
YUV to AV1	720p	1	5.96	6.55	4.55	0.06
YUV to AVC	720p	40	6.15	10.94	4.15	1.12
YUV to HEVC	720p	40	6.18	10.18	3.89	1.18
YUV to AV1	720p	40	38.9	45.5	33.02	2.31

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

30.1 Decoding

30.1.1 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.

30.1.2 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

30.2 Decoder PPU Scaling Performance Results

TYPE	RESOLUTION	JOBS	DEC_LOAD	SCALER_LOAD	FPS	CPU
AVC to RGBA	4k	1	22	0	84	24
AVC to RGBA	4k	8	92	1	341	19
HEVC to RGBA	4k	1	22	0	100	26
HEVC to RGBA	4k	8	96	2	408	22
VP9 to RGBA	4k	1	22	0	83	22
VP9 to RGBA	4k	8	94	1	336	17
AVC to RGBA	1080p	20	94	9	1195	6
HEVC to RGBA	1080p	20	94	9	1173	9
VP9 to RGBA	1080p	20	94	6	896	6
AVC to RGBA	720p	50	96	19	2213	4
HEVC to RGBA	720p	50	89	19	2297	4
VP9 to RGBA	720p	32	92	12	1674	6

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

31.1 Transcoding

31.1.1 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.

31.1.2 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)

31.2 Streaming Ladder Generation Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	5	35	94	4	250	30
AVC to HEVC	5	32	93	3	280	32
AVC to AV1	5	25	92	2	250	31
HEVC to AVC	5	36	94	4	255	32
HEVC to HEVC	5	33	93	3	285	34
HEVC to AV1	5	25	92	2	255	34
VP9 to AVC	5	40	94	4	250	29
VP9 to HEVC	5	38	93	3	281	30
VP9 to AV1	5	31	92	1	254	30

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

32.1 Encoding

32.1.1 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.

32.1.2 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

32.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	39	81	58
RGBA to HEVC	4k	1	1	37	82	60
RGBA to AV1	4k	1	1	42	81	61
RGBA to AVC	4k	3	0	47	90	54
RGBA to HEVC	4k	3	0	42	90	48
RGBA to AV1	4k	3	0	46	89	53
RGBA to AVC	4k	5	0	49	90	34
RGBA to HEVC	4k	5	0	45	91	38
RGBA to AV1	4k	5	0	49	90	35
RGBA to AVC	1080p	1	1	29	244	76
RGBA to HEVC	1080p	1	1	28	245	70
RGBA to AV1	1080p	1	1	32	239	80
RGBA to AVC	1080p	10	0	44	336	24
RGBA to HEVC	1080p	10	0	40	340	25
RGBA to AV1	1080p	10	0	45	336	25
RGBA to AVC	1080p	20	0	46	340	13
RGBA to HEVC	1080p	20	0	42	340	14
RGBA to AV1	1080p	20	0	46	340	13
RGBA to AVC	720p	1	1	17	329	110
RGBA to HEVC	720p	1	1	17	333	111
RGBA to AV1	720p	1	1	20	338	104
RGBA to AVC	720p	10	0	38	689	45
RGBA to HEVC	720p	10	0	37	706	45
RGBA to AV1	720p	10	0	43	685	44
RGBA to AVC	720p	20	0	41	731	24
RGBA to HEVC	720p	20	0	38	736	24
RGBA to AV1	720p	20	0	45	720	25

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

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 with a mix of xcoder-params EnableRdoQuant, rdoLevel, and lookaheadDepth.

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=<*>: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

33.2 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	42
YUV to HEVC	4k	2	0	0	1	48	108	44
YUV to AV1	4k	2	0	0	1	48	94	36
YUV to AVC	4k	2	0	0	2	49	102	40
YUV to HEVC	4k	2	0	0	2	49	54	28
YUV to AV1	4k	2	0	0	2	48	44	23
YUV to AVC	4k	2	0	0	3	48	102	40
YUV to HEVC	4k	2	0	0	3	49	32	21
YUV to AV1	4k	2	0	0	3	48	24	16
YUV to AVC	4k	2	0	1	1	48	60	29
YUV to HEVC	4k	2	0	1	1	48	76	34
YUV to AVC	4k	2	0	1	2	49	60	29
YUV to HEVC	4k	2	0	1	2	49	34	20
YUV to AVC	4k	2	0	1	3	50	60	29
YUV to HEVC	4k	2	0	1	3	49	22	15
YUV to AVC	4k	2	4	0	1	79	100	44
YUV to HEVC	4k	2	4	0	1	73	108	40
YUV to AV1	4k	2	4	0	1	73	94	40
YUV to AVC	4k	2	4	0	2	79	100	42
YUV to HEVC	4k	2	4	0	2	62	54	26
YUV to AV1	4k	2	4	0	2	60	44	24
YUV to AVC	4k	2	4	0	3	77	100	40
YUV to HEVC	4k	2	4	0	3	56	32	20
YUV to AV1	4k	2	4	0	3	56	26	21
YUV to AVC	4k	2	4	1	1	68	60	30
YUV to HEVC	4k	2	4	1	1	67	76	37
YUV to AVC	4k	2	4	1	2	67	60	26
YUV to HEVC	4k	2	4	1	2	57	34	22
YUV to AVC	4k	2	4	1	3	69	60	30
YUV to HEVC	4k	2	4	1	3	54	22	16
YUV to AVC	4k	2	16	0	1	80	100	39
YUV to HEVC	4k	2	16	0	1	74	108	43
YUV to AV1	4k	2	16	0	1	73	94	39
YUV to AVC	4k	2	16	0	2	80	100	39
YUV to HEVC	4k	2	16	0	2	61	54	26
YUV to AV1	4k	2	16	0	2	61	44	22
YUV to AVC	4k	2	16	0	3	79	100	35
YUV to HEVC	4k	2	16	0	3	56	32	21
YUV to AV1	4k	2	16	0	3	56	26	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	16	1	1	67	60	27
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	58	34	19
YUV to AVC	4k	2	16	1	3	66	60	29
YUV to HEVC	4k	2	16	1	3	54	22	18
YUV to AVC	4k	2	40	0	1	80	100	42
YUV to HEVC	4k	2	40	0	1	74	106	44
YUV to AV1	4k	2	40	0	1	73	92	37
YUV to AVC	4k	2	40	0	2	78	100	41
YUV to HEVC	4k	2	40	0	2	62	54	26
YUV to AV1	4k	2	40	0	2	61	44	25
YUV to AVC	4k	2	40	0	3	78	100	39
YUV to HEVC	4k	2	40	0	3	57	32	20
YUV to AV1	4k	2	40	0	3	56	26	16
YUV to AVC	4k	2	40	1	1	66	60	30
YUV to HEVC	4k	2	40	1	1	66	76	34
YUV to AVC	4k	2	40	1	2	66	60	30
YUV to HEVC	4k	2	40	1	2	58	34	20
YUV to AVC	4k	2	40	1	3	66	60	29
YUV to HEVC	4k	2	40	1	3	56	22	16
YUV to AVC	1080p	10	0	0	1	99	812	25
YUV to HEVC	1080p	10	0	0	1	95	837	25
YUV to AV1	1080p	10	0	0	1	99	749	22
YUV to AVC	1080p	10	0	0	2	99	814	26
YUV to HEVC	1080p	10	0	0	2	99	440	21
YUV to AV1	1080p	10	0	0	2	100	357	18
YUV to AVC	1080p	10	0	0	3	99	814	24
YUV to HEVC	1080p	10	0	0	3	99	260	18
YUV to AV1	1080p	10	0	0	3	99	191	16
YUV to AVC	1080p	10	0	1	1	100	480	20
YUV to HEVC	1080p	10	0	1	1	99	614	23
YUV to AVC	1080p	10	0	1	2	99	481	21
YUV to HEVC	1080p	10	0	1	2	99	281	18
YUV to AVC	1080p	10	0	1	3	100	482	21
YUV to HEVC	1080p	10	0	1	3	99	179	17
YUV to AVC	1080p	10	4	0	1	99	450	19
YUV to HEVC	1080p	10	4	0	1	99	525	20
YUV to AV1	1080p	10	4	0	1	99	460	18
YUV to AVC	1080p	10	4	0	2	99	448	19

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	4	0	2	100	330	19
YUV to AV1	1080p	10	4	0	2	99	270	16
YUV to AVC	1080p	10	4	0	3	99	450	19
YUV to HEVC	1080p	10	4	0	3	99	216	16
YUV to AV1	1080p	10	4	0	3	100	171	16
YUV to AVC	1080p	10	4	1	1	100	324	18
YUV to HEVC	1080p	10	4	1	1	99	420	19
YUV to AVC	1080p	10	4	1	2	99	323	19
YUV to HEVC	1080p	10	4	1	2	100	230	16
YUV to AVC	1080p	10	4	1	3	99	325	17
YUV to HEVC	1080p	10	4	1	3	100	159	15
YUV to AVC	1080p	10	16	0	1	99	448	19
YUV to HEVC	1080p	10	16	0	1	99	521	19
YUV to AV1	1080p	10	16	0	1	99	459	20
YUV to AVC	1080p	10	16	0	2	99	449	19
YUV to HEVC	1080p	10	16	0	2	99	329	17
YUV to AV1	1080p	10	16	0	2	99	270	17
YUV to AVC	1080p	10	16	0	3	99	448	20
YUV to HEVC	1080p	10	16	0	3	99	217	16
YUV to AV1	1080p	10	16	0	3	100	170	15
YUV to AVC	1080p	10	16	1	1	99	321	18
YUV to HEVC	1080p	10	16	1	1	99	419	18
YUV to AVC	1080p	10	16	1	2	99	320	18
YUV to HEVC	1080p	10	16	1	2	99	230	18
YUV to AVC	1080p	10	16	1	3	99	321	17
YUV to HEVC	1080p	10	16	1	3	99	155	16
YUV to AVC	1080p	10	40	0	1	99	441	19
YUV to HEVC	1080p	10	40	0	1	99	520	20
YUV to AV1	1080p	10	40	0	1	99	454	18
YUV to AVC	1080p	10	40	0	2	99	442	19
YUV to HEVC	1080p	10	40	0	2	99	320	19
YUV to AV1	1080p	10	40	0	2	99	270	17
YUV to AVC	1080p	10	40	0	3	99	442	20
YUV to HEVC	1080p	10	40	0	3	99	212	17
YUV to AV1	1080p	10	40	0	3	100	170	15
YUV to AVC	1080p	10	40	1	1	100	320	18
YUV to HEVC	1080p	10	40	1	1	99	413	20
YUV to AVC	1080p	10	40	1	2	99	320	18
YUV to HEVC	1080p	10	40	1	2	99	230	16
YUV to AVC	1080p	10	40	1	3	99	320	18

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	1	3	99	155	15
YUV to AVC	720p	20	0	0	1	82	1531	14
YUV to HEVC	720p	20	0	0	1	84	1591	15
YUV to AV1	720p	20	0	0	1	88	1422	13
YUV to AVC	720p	20	0	0	2	85	1555	15
YUV to HEVC	720p	20	0	0	2	99	960	16
YUV to AV1	720p	20	0	0	2	99	780	11
YUV to AVC	720p	20	0	0	3	84	1557	14
YUV to HEVC	720p	20	0	0	3	99	580	7
YUV to AV1	720p	20	0	0	3	100	420	6
YUV to AVC	720p	20	0	1	1	99	1080	13
YUV to HEVC	720p	20	0	1	1	99	1320	15
YUV to AVC	720p	20	0	1	2	99	1080	10
YUV to HEVC	720p	20	0	1	2	99	620	9
YUV to AVC	720p	20	0	1	3	99	1080	13
YUV to HEVC	720p	20	0	1	3	99	400	5
YUV to AVC	720p	20	4	0	1	99	922	15
YUV to HEVC	720p	20	4	0	1	99	1061	13
YUV to AV1	720p	20	4	0	1	100	882	7
YUV to AVC	720p	20	4	0	2	99	921	10
YUV to HEVC	720p	20	4	0	2	99	680	12
YUV to AV1	720p	20	4	0	2	99	560	8
YUV to AVC	720p	20	4	0	3	99	921	14
YUV to HEVC	720p	20	4	0	3	100	462	10
YUV to AV1	720p	20	4	0	3	99	360	14
YUV to AVC	720p	20	4	1	1	99	680	13
YUV to HEVC	720p	20	4	1	1	99	860	13
YUV to AVC	720p	20	4	1	2	99	680	12
YUV to HEVC	720p	20	4	1	2	99	480	7
YUV to AVC	720p	20	4	1	3	99	680	9
YUV to HEVC	720p	20	4	1	3	100	340	5
YUV to AVC	720p	20	16	0	1	99	920	12
YUV to HEVC	720p	20	16	0	1	99	1060	13
YUV to AV1	720p	20	16	0	1	100	923	7
YUV to AVC	720p	20	16	0	2	99	920	10
YUV to HEVC	720p	20	16	0	2	99	680	8
YUV to AV1	720p	20	16	0	2	99	560	12
YUV to AVC	720p	20	16	0	3	99	920	14
YUV to HEVC	720p	20	16	0	3	99	460	8
YUV to AV1	720p	20	16	0	3	99	360	7

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	20	16	1	1	99	680	8
YUV to HEVC	720p	20	16	1	1	99	860	12
YUV to AVC	720p	20	16	1	2	99	680	10
YUV to HEVC	720p	20	16	1	2	99	480	14
YUV to AVC	720p	20	16	1	3	99	680	9
YUV to HEVC	720p	20	16	1	3	99	340	5
YUV to AVC	720p	20	40	0	1	99	916	10
YUV to HEVC	720p	20	40	0	1	99	1044	14
YUV to AV1	720p	20	40	0	1	100	902	8
YUV to AVC	720p	20	40	0	2	99	916	13
YUV to HEVC	720p	20	40	0	2	99	680	13
YUV to AV1	720p	20	40	0	2	99	560	13
YUV to AVC	720p	20	40	0	3	99	917	12
YUV to HEVC	720p	20	40	0	3	99	460	10
YUV to AV1	720p	20	40	0	3	99	360	6
YUV to AVC	720p	20	40	1	1	99	678	14
YUV to HEVC	720p	20	40	1	1	99	840	11
YUV to AVC	720p	20	40	1	2	99	678	13
YUV to HEVC	720p	20	40	1	2	99	480	7
YUV to AVC	720p	20	40	1	3	99	677	8
YUV to HEVC	720p	20	40	1	3	99	340	6

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

34.1 Encoding with lookaheadDepth

34.1.1 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.

34.1.2 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

34.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	0	0	1	19	99	450	18
YUV to HEVC	1080p	10	0	0	1	19	99	523	19
YUV to AV1	1080p	10	0	0	1	19	99	460	19
YUV to AVC	1080p	10	0	0	2	19	99	450	19
YUV to HEVC	1080p	10	0	0	2	19	99	330	18
YUV to AV1	1080p	10	0	0	2	19	99	280	16
YUV to AVC	1080p	10	0	0	3	19	99	450	19
YUV to HEVC	1080p	10	0	0	3	19	99	218	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	19
YUV to AVC	1080p	10	0	1	2	19	99	327	18
YUV to HEVC	1080p	10	0	1	2	19	100	230	16
YUV to AVC	1080p	10	0	1	3	19	99	327	18
YUV to HEVC	1080p	10	0	1	3	19	99	158	16
YUV to AVC	1080p	10	4	0	1	19	99	451	18
YUV to HEVC	1080p	10	4	0	1	19	99	522	21
YUV to AV1	1080p	10	4	0	1	19	100	460	17
YUV to AVC	1080p	10	4	0	2	19	99	449	19
YUV to HEVC	1080p	10	4	0	2	19	100	330	17
YUV to AV1	1080p	10	4	0	2	19	99	270	17
YUV to AVC	1080p	10	4	0	3	19	99	449	19
YUV to HEVC	1080p	10	4	0	3	19	100	217	17
YUV to AV1	1080p	10	4	0	3	19	99	170	16
YUV to AVC	1080p	10	4	1	1	19	99	324	18
YUV to HEVC	1080p	10	4	1	1	19	100	420	18
YUV to AVC	1080p	10	4	1	2	19	100	323	18
YUV to HEVC	1080p	10	4	1	2	19	100	230	17
YUV to AVC	1080p	10	4	1	3	19	100	324	18
YUV to HEVC	1080p	10	4	1	3	19	99	157	15
YUV to AVC	1080p	10	16	0	1	19	99	448	18
YUV to HEVC	1080p	10	16	0	1	19	99	521	21
YUV to AV1	1080p	10	16	0	1	19	99	459	19
YUV to AVC	1080p	10	16	0	2	19	99	448	19
YUV to HEVC	1080p	10	16	0	2	19	99	327	17
YUV to AV1	1080p	10	16	0	2	19	99	270	17
YUV to AVC	1080p	10	16	0	3	19	99	447	18
YUV to HEVC	1080p	10	16	0	3	19	99	216	17

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	3	19	99	170	15
YUV to AVC	1080p	10	16	1	1	19	99	320	17
YUV to HEVC	1080p	10	16	1	1	19	99	419	19
YUV to AVC	1080p	10	16	1	2	19	100	320	17
YUV to HEVC	1080p	10	16	1	2	19	99	230	16
YUV to AVC	1080p	10	16	1	3	19	99	321	17
YUV to HEVC	1080p	10	16	1	3	19	99	155	17
YUV to AVC	1080p	10	40	0	1	19	99	442	18
YUV to HEVC	1080p	10	40	0	1	19	99	519	19
YUV to AV1	1080p	10	40	0	1	19	99	454	18
YUV to AVC	1080p	10	40	0	2	19	99	443	18
YUV to HEVC	1080p	10	40	0	2	19	99	320	17
YUV to AV1	1080p	10	40	0	2	19	99	270	17
YUV to AVC	1080p	10	40	0	3	19	99	442	18
YUV to HEVC	1080p	10	40	0	3	19	100	211	16
YUV to AV1	1080p	10	40	0	3	19	100	170	15
YUV to AVC	1080p	10	40	1	1	19	100	321	17
YUV to HEVC	1080p	10	40	1	1	19	99	411	18
YUV to AVC	1080p	10	40	1	2	19	99	320	17
YUV to HEVC	1080p	10	40	1	2	19	100	230	16
YUV to AVC	1080p	10	40	1	3	19	99	320	18
YUV to HEVC	1080p	10	40	1	3	19	99	151	16
YUV to AVC	1080p	10	0	0	1	23	99	450	19
YUV to HEVC	1080p	10	0	0	1	23	99	525	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	18
YUV to AV1	1080p	10	0	0	2	23	100	280	15
YUV to AVC	1080p	10	0	0	3	23	99	450	19
YUV to HEVC	1080p	10	0	0	3	23	100	216	16
YUV to AV1	1080p	10	0	0	3	23	99	176	15
YUV to AVC	1080p	10	0	1	1	23	100	325	19
YUV to HEVC	1080p	10	0	1	1	23	99	420	20
YUV to AVC	1080p	10	0	1	2	23	99	327	18
YUV to HEVC	1080p	10	0	1	2	23	100	230	17
YUV to AVC	1080p	10	0	1	3	23	99	327	18
YUV to HEVC	1080p	10	0	1	3	23	99	156	15
YUV to AVC	1080p	10	4	0	1	23	99	450	19
YUV to HEVC	1080p	10	4	0	1	23	99	523	19

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	4	0	1	23	99	460	19
YUV to AVC	1080p	10	4	0	2	23	99	451	20
YUV to HEVC	1080p	10	4	0	2	23	99	330	18
YUV to AV1	1080p	10	4	0	2	23	99	270	17
YUV to AVC	1080p	10	4	0	3	23	99	450	19
YUV to HEVC	1080p	10	4	0	3	23	99	217	16
YUV to AV1	1080p	10	4	0	3	23	100	171	15
YUV to AVC	1080p	10	4	1	1	23	100	324	18
YUV to HEVC	1080p	10	4	1	1	23	99	419	19
YUV to AVC	1080p	10	4	1	2	23	99	323	18
YUV to HEVC	1080p	10	4	1	2	23	99	230	17
YUV to AVC	1080p	10	4	1	3	23	100	324	19
YUV to HEVC	1080p	10	4	1	3	23	99	158	16
YUV to AVC	1080p	10	16	0	1	23	99	448	19
YUV to HEVC	1080p	10	16	0	1	23	99	521	19
YUV to AV1	1080p	10	16	0	1	23	99	460	20
YUV to AVC	1080p	10	16	0	2	23	99	449	20
YUV to HEVC	1080p	10	16	0	2	23	99	326	19
YUV to AV1	1080p	10	16	0	2	23	99	270	17
YUV to AVC	1080p	10	16	0	3	23	99	446	19
YUV to HEVC	1080p	10	16	0	3	23	99	217	16
YUV to AV1	1080p	10	16	0	3	23	100	171	14
YUV to AVC	1080p	10	16	1	1	23	99	320	17
YUV to HEVC	1080p	10	16	1	1	23	99	417	18
YUV to AVC	1080p	10	16	1	2	23	99	320	18
YUV to HEVC	1080p	10	16	1	2	23	99	230	16
YUV to AVC	1080p	10	16	1	3	23	99	321	16
YUV to HEVC	1080p	10	16	1	3	23	99	155	15
YUV to AVC	1080p	10	40	0	1	23	99	442	19
YUV to HEVC	1080p	10	40	0	1	23	99	520	18
YUV to AV1	1080p	10	40	0	1	23	99	455	19
YUV to AVC	1080p	10	40	0	2	23	99	442	19
YUV to HEVC	1080p	10	40	0	2	23	99	320	18
YUV to AV1	1080p	10	40	0	2	23	99	270	16
YUV to AVC	1080p	10	40	0	3	23	99	442	19
YUV to HEVC	1080p	10	40	0	3	23	100	211	15
YUV to AV1	1080p	10	40	0	3	23	100	171	16
YUV to AVC	1080p	10	40	1	1	23	99	320	16
YUV to HEVC	1080p	10	40	1	1	23	99	410	19

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	40	1	2	23	99	320	18
YUV to HEVC	1080p	10	40	1	2	23	99	230	16
YUV to AVC	1080p	10	40	1	3	23	99	320	19
YUV to HEVC	1080p	10	40	1	3	23	99	154	15
YUV to AVC	1080p	10	0	0	1	27	99	450	19
YUV to HEVC	1080p	10	0	0	1	27	99	522	20
YUV to AV1	1080p	10	0	0	1	27	99	460	19
YUV to AVC	1080p	10	0	0	2	27	99	450	20
YUV to HEVC	1080p	10	0	0	2	27	99	330	18
YUV to AV1	1080p	10	0	0	2	27	99	278	16
YUV to AVC	1080p	10	0	0	3	27	99	451	19
YUV to HEVC	1080p	10	0	0	3	27	99	217	16
YUV to AV1	1080p	10	0	0	3	27	100	178	16
YUV to AVC	1080p	10	0	1	1	27	100	326	18
YUV to HEVC	1080p	10	0	1	1	27	99	420	20
YUV to AVC	1080p	10	0	1	2	27	99	326	19
YUV to HEVC	1080p	10	0	1	2	27	99	230	17
YUV to AVC	1080p	10	0	1	3	27	99	327	19
YUV to HEVC	1080p	10	0	1	3	27	100	157	16
YUV to AVC	1080p	10	4	0	1	27	99	449	19
YUV to HEVC	1080p	10	4	0	1	27	99	523	20
YUV to AV1	1080p	10	4	0	1	27	99	460	17
YUV to AVC	1080p	10	4	0	2	27	99	450	18
YUV to HEVC	1080p	10	4	0	2	27	99	330	19
YUV to AV1	1080p	10	4	0	2	27	99	270	17
YUV to AVC	1080p	10	4	0	3	27	99	449	19
YUV to HEVC	1080p	10	4	0	3	27	100	216	16
YUV to AV1	1080p	10	4	0	3	27	99	172	15
YUV to AVC	1080p	10	4	1	1	27	99	324	17
YUV to HEVC	1080p	10	4	1	1	27	99	420	18
YUV to AVC	1080p	10	4	1	2	27	100	323	17
YUV to HEVC	1080p	10	4	1	2	27	100	230	16
YUV to AVC	1080p	10	4	1	3	27	99	324	19
YUV to HEVC	1080p	10	4	1	3	27	100	157	16
YUV to AVC	1080p	10	16	0	1	27	99	449	19
YUV to HEVC	1080p	10	16	0	1	27	99	520	19
YUV to AV1	1080p	10	16	0	1	27	99	460	19
YUV to AVC	1080p	10	16	0	2	27	99	448	19
YUV to HEVC	1080p	10	16	0	2	27	99	327	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	2	27	100	270	16
YUV to AVC	1080p	10	16	0	3	27	99	449	20
YUV to HEVC	1080p	10	16	0	3	27	99	215	16
YUV to AV1	1080p	10	16	0	3	27	99	170	16
YUV to AVC	1080p	10	16	1	1	27	100	322	17
YUV to HEVC	1080p	10	16	1	1	27	99	419	18
YUV to AVC	1080p	10	16	1	2	27	100	321	18
YUV to HEVC	1080p	10	16	1	2	27	99	230	16
YUV to AVC	1080p	10	16	1	3	27	99	320	17
YUV to HEVC	1080p	10	16	1	3	27	99	156	15
YUV to AVC	1080p	10	40	0	1	27	99	442	19
YUV to HEVC	1080p	10	40	0	1	27	99	519	20
YUV to AV1	1080p	10	40	0	1	27	99	452	18
YUV to AVC	1080p	10	40	0	2	27	99	442	19
YUV to HEVC	1080p	10	40	0	2	27	100	321	17
YUV to AV1	1080p	10	40	0	2	27	100	270	16
YUV to AVC	1080p	10	40	0	3	27	99	443	19
YUV to HEVC	1080p	10	40	0	3	27	99	211	15
YUV to AV1	1080p	10	40	0	3	27	100	170	15
YUV to AVC	1080p	10	40	1	1	27	99	320	18
YUV to HEVC	1080p	10	40	1	1	27	99	411	19
YUV to AVC	1080p	10	40	1	2	27	100	320	19
YUV to HEVC	1080p	10	40	1	2	27	100	230	17
YUV to AVC	1080p	10	40	1	3	27	100	320	19
YUV to HEVC	1080p	10	40	1	3	27	99	152	15

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

35.1 Transcoding

35.1.1 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.

35.1.2 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

35.2 Inplace Overlay Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	1	14	21	3	191	55
AVC to HEVC	1	15	21	3	203	55
AVC to AV1	1	13	21	2	174	52
HEVC to AVC	1	14	21	3	188	60
HEVC to HEVC	1	15	21	3	199	60
HEVC to AV1	1	13	20	2	172	58
VP9 to AVC	1	19	21	3	190	50
VP9 to HEVC	1	20	20	3	202	48
VP9 to AV1	1	18	21	2	174	48
AVC to AVC	8	72	94	18	538	24
AVC to HEVC	8	79	95	20	608	29
AVC to AV1	8	74	93	19	607	29
HEVC to AVC	8	66	93	18	560	28
HEVC to HEVC	8	73	94	21	638	27
HEVC to AV1	8	67	92	19	626	28
VP9 to AVC	8	76	94	18	537	26
VP9 to HEVC	8	83	93	19	614	27
VP9 to AV1	8	80	93	18	608	25
AVC to AVC	16	72	93	19	532	9
AVC to HEVC	16	77	93	20	608	10
AVC to AV1	16	74	93	20	599	11
HEVC to AVC	16	68	95	20	560	11
HEVC to HEVC	16	74	94	22	639	12
HEVC to AV1	16	69	93	20	624	11
VP9 to AVC	16	76	93	18	532	10
VP9 to HEVC	16	82	92	20	608	11
VP9 to AV1	16	83	95	20	608	11

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

36.1 Encoding

36.1.1 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.

36.1.2 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)

36.2 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	183	56
YUV to HEVC	4k	1	0	8	1	0	54	188	61
YUV to AV1	4k	1	0	8	1	0	60	182	64
YUV to AVC	4k	3	0	8	0	0	72	198	44
YUV to HEVC	4k	3	0	8	0	0	67	224	58
YUV to AV1	4k	3	0	8	0	0	71	213	37
YUV to AVC	4k	5	0	8	0	0	91	206	34
YUV to HEVC	4k	5	0	8	0	0	69	230	44
YUV to AV1	4k	5	0	8	0	0	77	226	42
YUV to AVC	4k	1	0	10	1	0	35	107	53
YUV to HEVC	4k	1	0	10	1	0	31	108	48
YUV to AV1	4k	1	0	10	1	0	35	106	52
YUV to AVC	4k	3	0	10	0	0	51	114	57
YUV to HEVC	4k	3	0	10	0	0	35	117	58
YUV to AV1	4k	3	0	10	0	0	40	116	57
YUV to AVC	1080p	1	0	8	1	0	32	451	80
YUV to HEVC	1080p	1	0	8	1	0	33	458	80
YUV to AV1	1080p	1	0	8	1	0	36	410	66
YUV to AVC	1080p	20	0	8	0	0	73	813	13
YUV to HEVC	1080p	20	0	8	0	0	65	854	14
YUV to AV1	1080p	20	0	8	0	0	71	817	14
YUV to AVC	1080p	1	0	10	1	0	24	312	54
YUV to HEVC	1080p	1	0	10	1	0	22	312	75
YUV to AV1	1080p	1	0	10	1	0	24	289	64
YUV to AVC	1080p	20	0	10	0	0	38	462	13
YUV to HEVC	1080p	20	0	10	0	0	34	463	12
YUV to AV1	1080p	20	0	10	0	0	39	459	13
YUV to AVC	720p	1	0	8	1	0	20	572	58
YUV to HEVC	720p	1	0	8	1	0	19	559	65
YUV to AV1	720p	1	0	8	1	0	24	518	43
YUV to AVC	720p	40	0	8	0	0	84	1566	8
YUV to HEVC	720p	40	0	8	0	0	82	1664	8
YUV to AV1	720p	40	0	8	0	0	85	1449	7

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

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.

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

37.1.2 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)

37.2 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	35	118	50
YUV to HEVC	4k	1	0	8	1	0	34	121	50
YUV to AV1	4k	1	0	8	1	0	36	110	50
YUV to AVC	4k	3	0	8	0	0	56	168	38
YUV to HEVC	4k	3	0	8	0	0	48	167	38
YUV to AV1	4k	3	0	8	0	0	53	157	36
YUV to AVC	4k	5	0	8	0	0	93	195	28
YUV to HEVC	4k	5	0	8	0	0	62	212	34
YUV to AV1	4k	5	0	8	0	0	67	195	33
YUV to AVC	4k	1	0	10	1	0	20	67	48
YUV to HEVC	4k	1	0	10	1	0	19	68	46
YUV to AV1	4k	1	0	10	1	0	20	65	47
YUV to AVC	4k	3	0	10	0	0	29	88	38
YUV to HEVC	4k	3	0	10	0	0	23	88	38
YUV to AV1	4k	3	0	10	0	0	28	85	37
YUV to AVC	1080p	1	0	8	1	0	26	353	43
YUV to HEVC	1080p	1	0	8	1	0	23	337	42
YUV to AV1	1080p	1	0	8	1	0	24	291	40
YUV to AVC	1080p	20	0	8	0	0	73	835	11
YUV to HEVC	1080p	20	0	8	0	0	66	873	11
YUV to AV1	1080p	20	0	8	0	0	69	825	10
YUV to AVC	1080p	1	0	10	1	0	16	198	50
YUV to HEVC	1080p	1	0	10	1	0	15	204	48
YUV to AV1	1080p	1	0	10	1	0	15	185	49
YUV to AVC	1080p	20	0	10	0	0	37	470	12
YUV to HEVC	1080p	20	0	10	0	0	35	472	12
YUV to AV1	1080p	20	0	10	0	0	38	444	12
YUV to AVC	720p	1	0	8	1	0	17	497	61
YUV to HEVC	720p	1	0	8	1	0	17	502	58
YUV to AV1	720p	1	0	8	1	0	18	414	40
YUV to AVC	720p	40	0	8	0	0	84	1704	5
YUV to HEVC	720p	40	0	8	0	0	80	1786	6
YUV to AV1	720p	40	0	8	0	0	87	1563	5

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

38.1 Encoding

38.1.1 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.

38.1.2 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

38.2 FFmpeg Latency Performance Results

TYPE	RESOLUTION	JOBS	ELAT_AVG (ms)	ELAT_MAX (ms)	ELAT_MIN (ms)	ELAT_VAR (ms)
YUV to AVC	4k	1	17.8	19.98	17.26	0.08
YUV to HEVC	4k	1	19.11	21.46	18.02	0.15
YUV to AV1	4k	1	24.31	28.53	18.73	0.69
YUV to AVC	4k	3	20.53	25.1	17.13	6.18
YUV to HEVC	4k	3	20.38	25.47	18.09	1.44
YUV to AV1	4k	3	25.2	31.71	18.82	1.88
YUV to AVC	4k	5	22.48	28.87	17.5	11.64
YUV to HEVC	4k	5	23.24	28.75	18.55	4.95
YUV to AV1	4k	5	27.93	34.89	19.07	3
YUV to AVC	1080p	1	5.86	7	5.08	0.05
YUV to HEVC	1080p	1	6.11	7.07	5.21	0.08
YUV to AV1	1080p	1	7.72	8.83	6.21	0.06
YUV to AVC	1080p	20	9.68	18.28	5.52	6.31
YUV to HEVC	1080p	20	8.24	14.58	5.58	1.66
YUV to AV1	1080p	20	11.49	21.3	6.61	4.42
YUV to AVC	720p	1	3.62	4.57	3.15	0.05
YUV to HEVC	720p	1	3.58	4.13	3.25	0.04
YUV to AV1	720p	1	4.46	5.05	3.74	0.05
YUV to AVC	720p	40	4.94	9.22	3.2	0.61
YUV to HEVC	720p	40	5.27	10.19	3.12	1.09
YUV to AV1	720p	40	6.92	20.48	3.84	1.22

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

39.1 Encoding

39.1.1 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.

39.1.2 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

39.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	25	82	54
RGBA to HEVC	4k	1	1	24	82	55
RGBA to AV1	4k	1	1	27	82	51
RGBA to AVC	4k	3	0	37	88	48
RGBA to HEVC	4k	3	0	30	89	52
RGBA to AV1	4k	3	0	33	89	53
RGBA to AVC	4k	5	0	48	91	33
RGBA to HEVC	4k	5	0	41	90	36
RGBA to AV1	4k	5	0	38	90	37
RGBA to AVC	1080p	1	1	19	248	80
RGBA to HEVC	1080p	1	1	18	248	64
RGBA to AV1	1080p	1	1	20	243	84
RGBA to AVC	1080p	10	0	33	331	24
RGBA to HEVC	1080p	10	0	29	331	23
RGBA to AV1	1080p	10	0	30	329	24
RGBA to AVC	1080p	20	0	38	330	12
RGBA to HEVC	1080p	20	0	33	338	13
RGBA to AV1	1080p	20	0	34	332	13
RGBA to AVC	720p	1	1	16	416	88
RGBA to HEVC	720p	1	1	12	351	119
RGBA to AV1	720p	1	1	13	351	113
RGBA to AVC	720p	10	0	34	703	47
RGBA to HEVC	720p	10	0	34	709	47
RGBA to AV1	720p	10	0	40	686	47
RGBA to AVC	720p	20	0	36	720	21
RGBA to HEVC	720p	20	0	35	723	18
RGBA to AV1	720p	20	0	43	712	20

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

40.1 Encoding

40.1.1 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.

40.1.2 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

40.2 Encoding EnableRdoQuant/rdoLevel/lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	0	0	1	48	155	55
YUV to HEVC	4k	2	0	0	1	48	166	48
YUV to AV1	4k	2	0	0	1	48	144	52
YUV to AVC	4k	2	0	0	2	48	156	50
YUV to HEVC	4k	2	0	0	2	49	86	36
YUV to AV1	4k	2	0	0	2	49	70	32
YUV to AVC	4k	2	0	0	3	48	154	54
YUV to HEVC	4k	2	0	0	3	49	50	27
YUV to AV1	4k	2	0	0	3	49	38	22
YUV to AVC	4k	2	0	1	1	48	94	37
YUV to HEVC	4k	2	0	1	1	48	120	46
YUV to AVC	4k	2	0	1	2	49	94	41
YUV to HEVC	4k	2	0	1	2	50	54	28
YUV to AVC	4k	2	0	1	3	49	94	37
YUV to HEVC	4k	2	0	1	3	49	34	21
YUV to AVC	4k	2	4	0	1	74	132	46
YUV to HEVC	4k	2	4	0	1	73	166	50
YUV to AV1	4k	2	4	0	1	72	144	48
YUV to AVC	4k	2	4	0	2	74	134	47
YUV to HEVC	4k	2	4	0	2	61	84	33
YUV to AV1	4k	2	4	0	2	60	70	34
YUV to AVC	4k	2	4	0	3	74	134	50
YUV to HEVC	4k	2	4	0	3	57	50	24
YUV to AV1	4k	2	4	0	3	57	40	24
YUV to AVC	4k	2	4	1	1	67	94	36
YUV to HEVC	4k	2	4	1	1	66	120	44
YUV to AVC	4k	2	4	1	2	68	92	41
YUV to HEVC	4k	2	4	1	2	57	54	27
YUV to AVC	4k	2	4	1	3	66	92	40
YUV to HEVC	4k	2	4	1	3	53	34	22
YUV to AVC	4k	2	20	0	1	74	132	52
YUV to HEVC	4k	2	20	0	1	72	166	48
YUV to AV1	4k	2	20	0	1	72	144	48

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	20	0	2	72	132	52
YUV to HEVC	4k	2	20	0	2	61	84	38
YUV to AV1	4k	2	20	0	2	60	70	32
YUV to AVC	4k	2	20	0	3	74	132	52
YUV to HEVC	4k	2	20	0	3	56	50	25
YUV to AV1	4k	2	20	0	3	57	40	22
YUV to AVC	4k	2	20	1	1	67	92	38
YUV to HEVC	4k	2	20	1	1	66	120	46
YUV to AVC	4k	2	20	1	2	67	92	37
YUV to HEVC	4k	2	20	1	2	58	54	26
YUV to AVC	4k	2	20	1	3	67	92	40
YUV to HEVC	4k	2	20	1	3	55	34	18
YUV to AVC	4k	2	40	0	1	73	126	46
YUV to HEVC	4k	2	40	0	1	73	164	53
YUV to AV1	4k	2	40	0	1	72	142	49
YUV to AVC	4k	2	40	0	2	73	131	46
YUV to HEVC	4k	2	40	0	2	62	84	35
YUV to AV1	4k	2	40	0	2	60	70	32
YUV to AVC	4k	2	40	0	3	74	130	41
YUV to HEVC	4k	2	40	0	3	56	50	26
YUV to AV1	4k	2	40	0	3	55	40	21
YUV to AVC	4k	2	40	1	1	66	92	40
YUV to HEVC	4k	2	40	1	1	66	118	49
YUV to AVC	4k	2	40	1	2	66	92	38
YUV to HEVC	4k	2	40	1	2	57	54	26
YUV to AVC	4k	2	40	1	3	66	92	39
YUV to HEVC	4k	2	40	1	3	53	34	19
YUV to AVC	1080p	10	0	0	1	67	785	26
YUV to HEVC	1080p	10	0	0	1	61	816	27
YUV to AV1	1080p	10	0	0	1	66	773	24
YUV to AVC	1080p	10	0	0	2	66	785	24
YUV to HEVC	1080p	10	0	0	2	100	691	24
YUV to AV1	1080p	10	0	0	2	100	558	20
YUV to AVC	1080p	10	0	0	3	66	781	25

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	100	408	19
YUV to AV1	1080p	10	0	0	3	100	298	19
YUV to AVC	1080p	10	0	1	1	100	752	24
YUV to HEVC	1080p	10	0	1	1	86	834	26
YUV to AVC	1080p	10	0	1	2	100	755	24
YUV to HEVC	1080p	10	0	1	2	100	439	19
YUV to AVC	1080p	10	0	1	3	99	752	24
YUV to HEVC	1080p	10	0	1	3	99	277	18
YUV to AVC	1080p	10	4	0	1	99	649	19
YUV to HEVC	1080p	10	4	0	1	87	711	20
YUV to AV1	1080p	10	4	0	1	93	680	19
YUV to AVC	1080p	10	4	0	2	99	648	19
YUV to HEVC	1080p	10	4	0	2	100	510	19
YUV to AV1	1080p	10	4	0	2	100	430	19
YUV to AVC	1080p	10	4	0	3	99	650	20
YUV to HEVC	1080p	10	4	0	3	99	340	19
YUV to AV1	1080p	10	4	0	3	100	271	17
YUV to AVC	1080p	10	4	1	1	100	510	19
YUV to HEVC	1080p	10	4	1	1	99	656	21
YUV to AVC	1080p	10	4	1	2	100	510	19
YUV to HEVC	1080p	10	4	1	2	100	360	19
YUV to AVC	1080p	10	4	1	3	99	509	19
YUV to HEVC	1080p	10	4	1	3	100	246	16
YUV to AVC	1080p	10	20	0	1	99	642	20
YUV to HEVC	1080p	10	20	0	1	89	716	22
YUV to AV1	1080p	10	20	0	1	93	670	18
YUV to AVC	1080p	10	20	0	2	99	641	20
YUV to HEVC	1080p	10	20	0	2	100	510	21
YUV to AV1	1080p	10	20	0	2	100	428	17
YUV to AVC	1080p	10	20	0	3	99	641	18
YUV to HEVC	1080p	10	20	0	3	100	340	17
YUV to AV1	1080p	10	20	0	3	99	270	16
YUV to AVC	1080p	10	20	1	1	99	507	20
YUV to HEVC	1080p	10	20	1	1	99	651	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	20	1	2	100	508	19
YUV to HEVC	1080p	10	20	1	2	99	360	19
YUV to AVC	1080p	10	20	1	3	99	508	20
YUV to HEVC	1080p	10	20	1	3	100	245	18
YUV to AVC	1080p	10	40	0	1	99	630	19
YUV to HEVC	1080p	10	40	0	1	89	704	22
YUV to AV1	1080p	10	40	0	1	91	645	20
YUV to AVC	1080p	10	40	0	2	100	630	18
YUV to HEVC	1080p	10	40	0	2	99	509	20
YUV to AV1	1080p	10	40	0	2	99	420	18
YUV to AVC	1080p	10	40	0	3	99	630	18
YUV to HEVC	1080p	10	40	0	3	99	334	17
YUV to AV1	1080p	10	40	0	3	99	270	16
YUV to AVC	1080p	10	40	1	1	100	500	20
YUV to HEVC	1080p	10	40	1	1	99	647	21
YUV to AVC	1080p	10	40	1	2	99	500	19
YUV to HEVC	1080p	10	40	1	2	100	358	18
YUV to AVC	1080p	10	40	1	3	100	500	19
YUV to HEVC	1080p	10	40	1	3	100	240	17
YUV to AVC	720p	20	0	0	1	77	1475	14
YUV to HEVC	720p	20	0	0	1	72	1552	15
YUV to AV1	720p	20	0	0	1	78	1398	12
YUV to AVC	720p	20	0	0	2	78	1501	13
YUV to HEVC	720p	20	0	0	2	99	1503	15
YUV to AV1	720p	20	0	0	2	100	1220	15
YUV to AVC	720p	20	0	0	3	77	1495	13
YUV to HEVC	720p	20	0	0	3	99	900	8
YUV to AV1	720p	20	0	0	3	99	660	8
YUV to AVC	720p	20	0	1	1	91	1548	14
YUV to HEVC	720p	20	0	1	1	76	1576	15
YUV to AVC	720p	20	0	1	2	92	1585	15
YUV to HEVC	720p	20	0	1	2	99	960	12
YUV to AVC	720p	20	0	1	3	92	1585	14
YUV to HEVC	720p	20	0	1	3	99	620	12

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	20	4	0	1	100	1100	9
YUV to HEVC	720p	20	4	0	1	99	1107	9
YUV to AV1	720p	20	4	0	1	100	878	7
YUV to AVC	720p	20	4	0	2	98	1096	10
YUV to HEVC	720p	20	4	0	2	99	1080	12
YUV to AV1	720p	20	4	0	2	99	867	7
YUV to AVC	720p	20	4	0	3	99	1099	9
YUV to HEVC	720p	20	4	0	3	99	740	14
YUV to AV1	720p	20	4	0	3	99	560	13
YUV to AVC	720p	20	4	1	1	99	1067	13
YUV to HEVC	720p	20	4	1	1	100	1116	9
YUV to AVC	720p	20	4	1	2	99	1069	13
YUV to HEVC	720p	20	4	1	2	99	760	14
YUV to AVC	720p	20	4	1	3	99	1067	12
YUV to HEVC	720p	20	4	1	3	100	530	8
YUV to AVC	720p	20	20	0	1	100	1071	9
YUV to HEVC	720p	20	20	0	1	100	1118	10
YUV to AV1	720p	20	20	0	1	100	903	7
YUV to AVC	720p	20	20	0	2	98	1069	9
YUV to HEVC	720p	20	20	0	2	99	1077	12
YUV to AV1	720p	20	20	0	2	99	880	11
YUV to AVC	720p	20	20	0	3	99	1076	9
YUV to HEVC	720p	20	20	0	3	99	723	9
YUV to AV1	720p	20	20	0	3	99	560	10
YUV to AVC	720p	20	20	1	1	99	1060	12
YUV to HEVC	720p	20	20	1	1	99	1120	9
YUV to AVC	720p	20	20	1	2	99	1061	13
YUV to HEVC	720p	20	20	1	2	99	760	10
YUV to AVC	720p	20	20	1	3	99	1062	13
YUV to HEVC	720p	20	20	1	3	100	520	10
YUV to AVC	720p	20	40	0	1	99	1043	9
YUV to HEVC	720p	20	40	0	1	99	1082	9
YUV to AV1	720p	20	40	0	1	100	899	7
YUV to AVC	720p	20	40	0	2	97	1054	9

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	720p	20	40	0	2	99	1060	13
YUV to AV1	720p	20	40	0	2	99	874	10
YUV to AVC	720p	20	40	0	3	99	1053	9
YUV to HEVC	720p	20	40	0	3	99	720	13
YUV to AV1	720p	20	40	0	3	100	560	8
YUV to AVC	720p	20	40	1	1	99	1059	13
YUV to HEVC	720p	20	40	1	1	99	1098	10
YUV to AVC	720p	20	40	1	2	99	1059	12
YUV to HEVC	720p	20	40	1	2	99	760	14
YUV to AVC	720p	20	40	1	3	99	1059	11
YUV to HEVC	720p	20	40	1	3	99	520	9

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

41.1 Encoding with lookaheadDepth

41.1.1 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.

41.1.2 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

41.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	0	0	1	19	99	684	22
YUV to HEVC	1080p	10	0	0	1	19	93	771	22
YUV to AV1	1080p	10	0	0	1	19	97	711	21
YUV to AVC	1080p	10	0	0	2	19	99	683	21
YUV to HEVC	1080p	10	0	0	2	19	100	512	21
YUV to AV1	1080p	10	0	0	2	19	100	431	20
YUV to AVC	1080p	10	0	0	3	19	99	683	21
YUV to HEVC	1080p	10	0	0	3	19	100	340	19
YUV to AV1	1080p	10	0	0	3	19	100	277	18
YUV to AVC	1080p	10	0	1	1	19	99	510	20
YUV to HEVC	1080p	10	0	1	1	19	99	658	21
YUV to AVC	1080p	10	0	1	2	19	100	511	19
YUV to HEVC	1080p	10	0	1	2	19	100	360	18
YUV to AVC	1080p	10	0	1	3	19	99	510	20
YUV to HEVC	1080p	10	0	1	3	19	100	246	16
YUV to AVC	1080p	10	4	0	1	19	99	644	19
YUV to HEVC	1080p	10	4	0	1	19	85	692	20
YUV to AV1	1080p	10	4	0	1	19	88	633	19
YUV to AVC	1080p	10	4	0	2	19	99	647	18
YUV to HEVC	1080p	10	4	0	2	19	99	512	21
YUV to AV1	1080p	10	4	0	2	19	99	430	18
YUV to AVC	1080p	10	4	0	3	19	99	648	18
YUV to HEVC	1080p	10	4	0	3	19	99	340	18
YUV to AV1	1080p	10	4	0	3	19	99	270	16
YUV to AVC	1080p	10	4	1	1	19	100	509	20
YUV to HEVC	1080p	10	4	1	1	19	99	656	21
YUV to AVC	1080p	10	4	1	2	19	100	509	18
YUV to HEVC	1080p	10	4	1	2	19	99	360	17
YUV to AVC	1080p	10	4	1	3	19	99	510	19
YUV to HEVC	1080p	10	4	1	3	19	99	245	16
YUV to AVC	1080p	10	16	0	1	19	92	601	18
YUV to HEVC	1080p	10	16	0	1	19	85	679	20
YUV to AV1	1080p	10	16	0	1	19	87	618	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	0	2	19	92	602	17
YUV to HEVC	1080p	10	16	0	2	19	99	510	19
YUV to AV1	1080p	10	16	0	2	19	100	428	17
YUV to AVC	1080p	10	16	0	3	19	91	599	18
YUV to HEVC	1080p	10	16	0	3	19	100	340	19
YUV to AV1	1080p	10	16	0	3	19	100	270	17
YUV to AVC	1080p	10	16	1	1	19	99	510	18
YUV to HEVC	1080p	10	16	1	1	19	100	651	20
YUV to AVC	1080p	10	16	1	2	19	100	507	18
YUV to HEVC	1080p	10	16	1	2	19	100	360	18
YUV to AVC	1080p	10	16	1	3	19	100	507	18
YUV to HEVC	1080p	10	16	1	3	19	99	243	16
YUV to AVC	1080p	10	40	0	1	19	91	580	17
YUV to HEVC	1080p	10	40	0	1	19	81	651	19
YUV to AV1	1080p	10	40	0	1	19	87	588	17
YUV to AVC	1080p	10	40	0	2	19	92	588	18
YUV to HEVC	1080p	10	40	0	2	19	99	509	19
YUV to AV1	1080p	10	40	0	2	19	99	420	18
YUV to AVC	1080p	10	40	0	3	19	90	583	17
YUV to HEVC	1080p	10	40	0	3	19	99	336	18
YUV to AV1	1080p	10	40	0	3	19	100	270	15
YUV to AVC	1080p	10	40	1	1	19	99	500	19
YUV to HEVC	1080p	10	40	1	1	19	100	646	21
YUV to AVC	1080p	10	40	1	2	19	99	500	19
YUV to HEVC	1080p	10	40	1	2	19	100	358	18
YUV to AVC	1080p	10	40	1	3	19	100	500	19
YUV to HEVC	1080p	10	40	1	3	19	99	241	16
YUV to AVC	1080p	10	0	0	1	23	99	682	21
YUV to HEVC	1080p	10	0	0	1	23	94	767	23
YUV to AV1	1080p	10	0	0	1	23	98	714	20
YUV to AVC	1080p	10	0	0	2	23	99	683	22
YUV to HEVC	1080p	10	0	0	2	23	99	510	20
YUV to AV1	1080p	10	0	0	2	23	99	430	18
YUV to AVC	1080p	10	0	0	3	23	99	683	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	23	99	340	18
YUV to AV1	1080p	10	0	0	3	23	100	279	17
YUV to AVC	1080p	10	0	1	1	23	99	510	20
YUV to HEVC	1080p	10	0	1	1	23	99	656	21
YUV to AVC	1080p	10	0	1	2	23	100	510	20
YUV to HEVC	1080p	10	0	1	2	23	100	360	18
YUV to AVC	1080p	10	0	1	3	23	99	510	20
YUV to HEVC	1080p	10	0	1	3	23	100	247	16
YUV to AVC	1080p	10	4	0	1	23	99	647	20
YUV to HEVC	1080p	10	4	0	1	23	87	706	21
YUV to AV1	1080p	10	4	0	1	23	89	649	19
YUV to AVC	1080p	10	4	0	2	23	99	649	20
YUV to HEVC	1080p	10	4	0	2	23	100	510	21
YUV to AV1	1080p	10	4	0	2	23	100	430	18
YUV to AVC	1080p	10	4	0	3	23	99	647	19
YUV to HEVC	1080p	10	4	0	3	23	99	340	19
YUV to AV1	1080p	10	4	0	3	23	99	270	18
YUV to AVC	1080p	10	4	1	1	23	100	510	20
YUV to HEVC	1080p	10	4	1	1	23	100	657	20
YUV to AVC	1080p	10	4	1	2	23	100	510	20
YUV to HEVC	1080p	10	4	1	2	23	100	360	19
YUV to AVC	1080p	10	4	1	3	23	100	510	20
YUV to HEVC	1080p	10	4	1	3	23	100	245	15
YUV to AVC	1080p	10	16	0	1	23	96	623	19
YUV to HEVC	1080p	10	16	0	1	23	86	691	21
YUV to AV1	1080p	10	16	0	1	23	86	631	18
YUV to AVC	1080p	10	16	0	2	23	96	621	18
YUV to HEVC	1080p	10	16	0	2	23	100	510	18
YUV to AV1	1080p	10	16	0	2	23	100	430	17
YUV to AVC	1080p	10	16	0	3	23	96	623	17
YUV to HEVC	1080p	10	16	0	3	23	99	338	18
YUV to AV1	1080p	10	16	0	3	23	99	270	16
YUV to AVC	1080p	10	16	1	1	23	100	508	19
YUV to HEVC	1080p	10	16	1	1	23	99	651	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	1	2	23	100	506	20
YUV to HEVC	1080p	10	16	1	2	23	100	360	18
YUV to AVC	1080p	10	16	1	3	23	99	509	20
YUV to HEVC	1080p	10	16	1	3	23	100	243	16
YUV to AVC	1080p	10	40	0	1	23	96	608	17
YUV to HEVC	1080p	10	40	0	1	23	83	670	20
YUV to AV1	1080p	10	40	0	1	23	87	604	17
YUV to AVC	1080p	10	40	0	2	23	95	610	19
YUV to HEVC	1080p	10	40	0	2	23	99	508	19
YUV to AV1	1080p	10	40	0	2	23	100	420	17
YUV to AVC	1080p	10	40	0	3	23	95	609	18
YUV to HEVC	1080p	10	40	0	3	23	99	335	20
YUV to AV1	1080p	10	40	0	3	23	100	269	17
YUV to AVC	1080p	10	40	1	1	23	99	500	19
YUV to HEVC	1080p	10	40	1	1	23	100	645	20
YUV to AVC	1080p	10	40	1	2	23	99	500	20
YUV to HEVC	1080p	10	40	1	2	23	100	359	17
YUV to AVC	1080p	10	40	1	3	23	99	500	19
YUV to HEVC	1080p	10	40	1	3	23	100	241	16
YUV to AVC	1080p	10	0	0	1	27	99	684	21
YUV to HEVC	1080p	10	0	0	1	27	94	770	23
YUV to AV1	1080p	10	0	0	1	27	97	712	22
YUV to AVC	1080p	10	0	0	2	27	99	685	22
YUV to HEVC	1080p	10	0	0	2	27	100	510	20
YUV to AV1	1080p	10	0	0	2	27	99	430	17
YUV to AVC	1080p	10	0	0	3	27	99	683	21
YUV to HEVC	1080p	10	0	0	3	27	99	341	17
YUV to AV1	1080p	10	0	0	3	27	100	278	17
YUV to AVC	1080p	10	0	1	1	27	99	510	21
YUV to HEVC	1080p	10	0	1	1	27	99	656	22
YUV to AVC	1080p	10	0	1	2	27	100	510	19
YUV to HEVC	1080p	10	0	1	2	27	100	360	19
YUV to AVC	1080p	10	0	1	3	27	99	510	20
YUV to HEVC	1080p	10	0	1	3	27	100	244	16

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	4	0	1	27	99	646	18
YUV to HEVC	1080p	10	4	0	1	27	89	717	21
YUV to AV1	1080p	10	4	0	1	27	92	663	20
YUV to AVC	1080p	10	4	0	2	27	100	647	18
YUV to HEVC	1080p	10	4	0	2	27	100	510	18
YUV to AV1	1080p	10	4	0	2	27	100	430	17
YUV to AVC	1080p	10	4	0	3	27	99	646	20
YUV to HEVC	1080p	10	4	0	3	27	99	340	18
YUV to AV1	1080p	10	4	0	3	27	100	271	17
YUV to AVC	1080p	10	4	1	1	27	99	510	19
YUV to HEVC	1080p	10	4	1	1	27	100	658	21
YUV to AVC	1080p	10	4	1	2	27	100	509	20
YUV to HEVC	1080p	10	4	1	2	27	100	360	18
YUV to AVC	1080p	10	4	1	3	27	99	510	19
YUV to HEVC	1080p	10	4	1	3	27	100	245	17
YUV to AVC	1080p	10	16	0	1	27	99	640	18
YUV to HEVC	1080p	10	16	0	1	27	87	700	20
YUV to AV1	1080p	10	16	0	1	27	90	644	19
YUV to AVC	1080p	10	16	0	2	27	99	641	19
YUV to HEVC	1080p	10	16	0	2	27	99	510	19
YUV to AV1	1080p	10	16	0	2	27	100	429	17
YUV to AVC	1080p	10	16	0	3	27	99	640	19
YUV to HEVC	1080p	10	16	0	3	27	99	338	17
YUV to AV1	1080p	10	16	0	3	27	100	270	17
YUV to AVC	1080p	10	16	1	1	27	99	506	19
YUV to HEVC	1080p	10	16	1	1	27	99	652	21
YUV to AVC	1080p	10	16	1	2	27	99	506	20
YUV to HEVC	1080p	10	16	1	2	27	99	360	18
YUV to AVC	1080p	10	16	1	3	27	99	505	20
YUV to HEVC	1080p	10	16	1	3	27	99	245	16
YUV to AVC	1080p	10	40	0	1	27	100	630	19
YUV to HEVC	1080p	10	40	0	1	27	86	680	21
YUV to AV1	1080p	10	40	0	1	27	86	615	18
YUV to AVC	1080p	10	40	0	2	27	97	627	19

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	0	2	27	99	508	21
YUV to AV1	1080p	10	40	0	2	27	100	420	17
YUV to AVC	1080p	10	40	0	3	27	99	629	18
YUV to HEVC	1080p	10	40	0	3	27	100	336	17
YUV to AV1	1080p	10	40	0	3	27	100	269	16
YUV to AVC	1080p	10	40	1	1	27	100	500	18
YUV to HEVC	1080p	10	40	1	1	27	99	648	20
YUV to AVC	1080p	10	40	1	2	27	100	500	20
YUV to HEVC	1080p	10	40	1	2	27	99	356	18
YUV to AVC	1080p	10	40	1	3	27	99	501	19
YUV to HEVC	1080p	10	40	1	3	27	100	242	16

Appendix A: GStreamer XStack Command

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

[illegible]

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


Appendix B: 7x7 Grid Layout

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

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

Appendix C: GStreamer Ladder Command

Example of single input with 64 outputs

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

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