



Quadra™ Performance Test Report V5.8

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: 5806tCr2

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	106	7
HEVC to YUV	8k	1	0	8	1	10	0	104	8
VP9 to YUV	8k	1	0	8	1	24	0	38	4
YUV to AVC	8k	1	0	8	1	0	95	67	54
YUV to HEVC	8k	1	0	8	1	0	98	83	72
AVC to AVC	8k	1	1	8	1	8	99	54	3
AVC to HEVC	8k	1	1	8	1	8	99	70	6
HEVC to AVC	8k	1	1	8	1	7	100	53	4
HEVC to HEVC	8k	1	1	8	1	7	100	70	3
VP9 to AVC	8k	1	1	8	1	24	43	35	2
VP9 to HEVC	8k	1	1	8	1	24	42	37	4
AVC to YUV	8k	1	0	10	1	10	0	61	9
HEVC to YUV	8k	1	0	10	1	10	0	56	7
VP9 to YUV	8k	1	0	10	1	24	0	33	15
YUV to AVC	8k	1	0	10	1	0	90	48	76
YUV to HEVC	8k	1	0	10	1	0	72	59	93
AVC to YUV	4k	1	0	8	1	18	0	315	16
HEVC to YUV	4k	1	0	8	1	16	0	326	19
VP9 to YUV	4k	1	0	8	1	24	0	154	4
AVC to YUV	4k	16	0	8	0	98	0	480	3
HEVC to YUV	4k	16	0	8	0	99	0	504	3
VP9 to YUV	4k	16	0	8	0	98	0	485	1
YUV to AVC	4k	1	0	8	1	0	95	294	40
YUV to HEVC	4k	1	0	8	1	0	96	327	49
YUV to AV1	4k	1	0	8	1	0	94	281	43
YUV to AVC	4k	4	0	8	0	0	94	298	23
YUV to HEVC	4k	4	0	8	0	0	96	332	13
YUV to AV1	4k	4	0	8	0	0	96	288	14
YUV to AVC	4k	8	0	8	0	0	99	320	8
YUV to HEVC	4k	8	0	8	0	0	99	344	10
YUV to AV1	4k	8	0	8	0	0	100	296	9
AVC to AVC	4k	1	1	8	1	14	93	221	13
AVC to HEVC	4k	1	1	8	1	14	89	271	15
AVC to AV1	4k	1	1	8	1	16	93	262	16
HEVC to AVC	4k	1	1	8	1	14	93	219	18
HEVC to HEVC	4k	1	1	8	1	15	92	279	15
HEVC to AV1	4k	1	1	8	1	13	93	260	16
VP9 to AVC	4k	1	1	8	1	24	45	144	3
VP9 to HEVC	4k	1	1	8	1	24	43	148	4

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	50	148	3
AVC to AVC	4k	4	1	8	0	62	94	236	7
AVC to HEVC	4k	4	1	8	0	68	96	300	3
AVC to AV1	4k	4	1	8	0	59	95	276	7
HEVC to AVC	4k	4	1	8	0	57	96	236	8
HEVC to HEVC	4k	4	1	8	0	60	96	300	3
HEVC to AV1	4k	4	1	8	0	46	95	276	9
VP9 to AVC	4k	4	1	8	0	63	96	240	3
VP9 to HEVC	4k	4	1	8	0	66	96	304	3
VP9 to AV1	4k	4	1	8	0	60	92	260	3
AVC to AVC	4k	8	1	8	0	68	99	212	2
AVC to HEVC	4k	8	1	8	0	76	99	280	2
AVC to AV1	4k	8	1	8	0	67	100	272	3
HEVC to AVC	4k	8	1	8	0	64	100	216	2
HEVC to HEVC	4k	8	1	8	0	68	100	280	2
HEVC to AV1	4k	8	1	8	0	58	100	274	2
VP9 to AVC	4k	8	1	8	0	68	100	233	2
VP9 to HEVC	4k	8	1	8	0	72	99	296	2
VP9 to AV1	4k	8	1	8	0	64	100	280	2
AVC to YUV	4k	1	0	10	1	16	0	224	9
HEVC to YUV	4k	1	0	10	1	15	0	226	9
VP9 to YUV	4k	1	0	10	1	24	0	124	4
AVC to YUV	4k	16	0	10	0	100	0	282	0
HEVC to YUV	4k	16	0	10	0	98	0	279	0
VP9 to YUV	4k	16	0	10	0	99	0	264	0
YUV to AVC	4k	1	0	10	1	0	69	198	61
YUV to HEVC	4k	1	0	10	1	0	60	206	58
YUV to AV1	4k	1	0	10	1	0	66	198	60
YUV to AVC	4k	4	0	10	0	0	92	218	30
YUV to HEVC	4k	4	0	10	0	0	76	248	36
YUV to AV1	4k	4	0	10	0	0	81	240	42
AVC to YUV	1080p	1	0	8	1	35	0	864	21
HEVC to YUV	1080p	1	0	8	1	30	0	835	28
VP9 to YUV	1080p	1	0	8	1	22	0	554	6
AVC to YUV	1080p	40	0	8	0	87	0	1720	0
HEVC to YUV	1080p	40	0	8	0	94	0	1800	1
VP9 to YUV	1080p	40	0	8	0	81	0	1761	0
YUV to AVC	1080p	1	0	8	1	0	53	693	28
YUV to HEVC	1080p	1	0	8	1	0	51	699	27
YUV to AV1	1080p	1	0	8	1	0	54	636	24

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1280	3
YUV to HEVC	1080p	32	0	8	0	0	99	1344	6
YUV to AV1	1080p	32	0	8	0	0	99	1175	3
AVC to AVC	1080p	1	1	8	1	42	81	964	21
AVC to HEVC	1080p	1	1	8	1	43	78	1011	26
AVC to AV1	1080p	1	1	8	1	38	79	897	22
HEVC to AVC	1080p	1	1	8	1	34	76	894	30
HEVC to HEVC	1080p	1	1	8	1	37	76	969	32
HEVC to AV1	1080p	1	1	8	1	33	77	849	30
VP9 to AVC	1080p	1	1	8	1	21	42	543	6
VP9 to HEVC	1080p	1	1	8	1	22	39	539	9
VP9 to AV1	1080p	1	1	8	1	21	46	537	9
AVC to AVC	1080p	32	1	8	0	76	99	931	2
AVC to HEVC	1080p	32	1	8	0	83	100	1063	1
AVC to AV1	1080p	32	1	8	0	77	100	1027	1
HEVC to AVC	1080p	32	1	8	0	68	100	992	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1120	1
HEVC to AV1	1080p	32	1	8	0	70	99	1057	1
VP9 to AVC	1080p	32	1	8	0	70	99	1088	1
VP9 to HEVC	1080p	32	1	8	0	74	99	1216	1
VP9 to AV1	1080p	32	1	8	0	71	99	1120	1
AVC to YUV	1080p	1	0	10	1	28	0	669	7
HEVC to YUV	1080p	1	0	10	1	26	0	696	13
VP9 to YUV	1080p	1	0	10	1	22	0	451	7
AVC to YUV	1080p	40	0	10	0	65	0	1081	0
HEVC to YUV	1080p	40	0	10	0	70	0	1080	0
VP9 to YUV	1080p	40	0	10	0	73	0	1046	0
YUV to AVC	1080p	1	0	10	1	0	38	497	44
YUV to HEVC	1080p	1	0	10	1	0	36	499	38
YUV to AV1	1080p	1	0	10	1	0	39	464	37
YUV to AVC	1080p	32	0	10	0	0	61	790	5
YUV to HEVC	1080p	32	0	10	0	0	57	780	6
YUV to AV1	1080p	32	0	10	0	0	64	768	5
AVC to YUV	720p	1	0	8	1	43	0	1193	16
HEVC to YUV	720p	1	0	8	1	37	0	1193	22
VP9 to YUV	720p	1	0	8	1	32	0	1014	8
AVC to YUV	720p	100	0	8	0	100	0	2519	0
HEVC to YUV	720p	100	0	8	0	100	0	2915	2
VP9 to YUV	720p	100	0	8	0	100	0	2613	0
YUV to AVC	720p	1	0	8	1	0	32	898	20

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	30	876	16
YUV to AV1	720p	1	0	8	1	0	34	793	18
YUV to AVC	720p	64	0	8	0	0	95	2304	1
YUV to HEVC	720p	64	0	8	0	0	94	2346	1
YUV to AV1	720p	64	0	8	0	0	99	1995	1
AVC to AVC	720p	1	1	8	1	47	48	1251	19
AVC to HEVC	720p	1	1	8	1	47	48	1263	22
AVC to AV1	720p	1	1	8	1	40	52	1075	18
HEVC to AVC	720p	1	1	8	1	40	46	1214	24
HEVC to HEVC	720p	1	1	8	1	40	46	1221	25
HEVC to AV1	720p	1	1	8	1	34	48	1055	25
VP9 to AVC	720p	1	1	8	1	32	36	997	13
VP9 to HEVC	720p	1	1	8	1	32	35	983	12
VP9 to AV1	720p	1	1	8	1	32	45	985	13
AVC to AVC	720p	64	1	8	0	94	100	2036	0
AVC to HEVC	720p	64	1	8	0	95	100	2112	0
AVC to AV1	720p	64	1	8	0	75	100	1766	0
HEVC to AVC	720p	64	1	8	0	83	100	2048	0
HEVC to HEVC	720p	64	1	8	0	83	100	2114	0
HEVC to AV1	720p	64	1	8	0	67	100	1789	0
VP9 to AVC	720p	64	1	8	0	96	100	2240	0
VP9 to HEVC	720p	64	1	8	0	97	100	2305	0
VP9 to AV1	720p	64	1	8	0	75	100	1905	0

2. T1A – Libxcoder Throughput

2.1 Decoding

2.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

2.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

2.2 Encoding

2.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

2.2.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

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

2.3 Transcoding

2.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

2.3.2 Command line

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

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

2.4 Libxcodec Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	10	0	92	7
HEVC to YUV	8k	1	0	8	1	10	0	96	6
VP9 to YUV	8k	1	0	8	1	25	0	37	3
YUV to AVC	8k	1	0	8	1	0	96	67	29
YUV to HEVC	8k	1	0	8	1	0	95	81	34
AVC to AVC	8k	1	1	8	1	9	100	54	5
AVC to HEVC	8k	1	1	8	1	10	99	71	5
HEVC to AVC	8k	1	1	8	1	10	100	52	5
HEVC to HEVC	8k	1	1	8	1	10	100	70	4
VP9 to AVC	8k	1	1	8	1	24	46	35	3
VP9 to HEVC	8k	1	1	8	1	24	43	37	4
AVC to YUV	8k	1	0	10	1	9	0	54	6
HEVC to YUV	8k	1	0	10	1	9	0	51	6
VP9 to YUV	8k	1	0	10	1	24	0	32	4
YUV to AVC	8k	1	0	10	1	0	68	39	36
YUV to HEVC	8k	1	0	10	1	0	45	39	36
AVC to YUV	4k	1	0	8	1	17	0	319	8
HEVC to YUV	4k	1	0	8	1	16	0	329	6
VP9 to YUV	4k	1	0	8	1	23	0	153	3
AVC to YUV	4k	16	0	8	0	99	0	485	0
HEVC to YUV	4k	16	0	8	0	99	0	510	0
VP9 to YUV	4k	16	0	8	0	99	0	490	0
YUV to AVC	4k	1	0	8	1	0	90	281	24
YUV to HEVC	4k	1	0	8	1	0	86	295	26
YUV to AV1	4k	1	0	8	1	0	89	264	24
YUV to AVC	4k	4	0	8	0	0	96	311	11
YUV to HEVC	4k	4	0	8	0	0	96	334	11
YUV to AV1	4k	4	0	8	0	0	96	289	10
YUV to AVC	4k	8	0	8	0	0	100	324	7
YUV to HEVC	4k	8	0	8	0	0	99	347	9
YUV to AV1	4k	8	0	8	0	0	100	301	7
AVC to AVC	4k	1	1	8	0	17	97	224	7
AVC to HEVC	4k	1	1	8	0	17	96	276	8
AVC to AV1	4k	1	1	8	0	17	96	263	8
HEVC to AVC	4k	1	1	8	0	16	97	224	5
HEVC to HEVC	4k	1	1	8	0	17	96	285	6
HEVC to AV1	4k	1	1	8	0	17	97	266	7
VP9 to AVC	4k	1	1	8	0	24	45	148	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	150	4
VP9 to AV1	4k	1	1	8	0	24	50	148	4
AVC to AVC	4k	4	1	8	0	62	96	242	5
AVC to HEVC	4k	4	1	8	0	68	96	301	5
AVC to AV1	4k	4	1	8	0	61	95	275	5
HEVC to AVC	4k	4	1	8	0	57	96	239	5
HEVC to HEVC	4k	4	1	8	0	61	96	294	5
HEVC to AV1	4k	4	1	8	0	50	95	274	5
VP9 to AVC	4k	4	1	8	0	63	97	245	4
VP9 to HEVC	4k	4	1	8	0	66	96	302	4
VP9 to AV1	4k	4	1	8	0	58	95	277	4
AVC to AVC	4k	8	1	8	0	68	99	217	3
AVC to HEVC	4k	8	1	8	0	76	100	280	4
AVC to AV1	4k	8	1	8	0	68	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	59	100	275	3
VP9 to AVC	4k	8	1	8	0	67	100	235	3
VP9 to HEVC	4k	8	1	8	0	72	100	301	3
VP9 to AV1	4k	8	1	8	0	63	100	286	3
AVC to YUV	4k	1	0	10	1	15	0	213	7
HEVC to YUV	4k	1	0	10	1	14	0	207	6
VP9 to YUV	4k	1	0	10	1	24	0	124	4
AVC to YUV	4k	16	0	10	0	91	0	284	0
HEVC to YUV	4k	16	0	10	0	91	0	283	0
VP9 to YUV	4k	16	0	10	0	99	0	267	0
YUV to AVC	4k	1	0	10	1	0	49	158	27
YUV to HEVC	4k	1	0	10	1	0	46	161	27
YUV to AV1	4k	1	0	10	1	0	50	152	26
YUV to AVC	4k	4	0	10	0	0	67	208	25
YUV to HEVC	4k	4	0	10	0	0	57	211	27
YUV to AV1	4k	4	0	10	0	0	57	187	28
AVC to YUV	1080p	1	0	8	1	33	0	829	14
HEVC to YUV	1080p	1	0	8	1	31	0	868	10
VP9 to YUV	1080p	1	0	8	1	22	0	553	4
AVC to YUV	1080p	40	0	8	0	81	0	1623	0
HEVC to YUV	1080p	40	0	8	0	89	0	1741	0
VP9 to YUV	1080p	40	0	8	0	71	0	1652	0
YUV to AVC	1080p	1	0	8	1	0	55	718	15

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	53	723	17
YUV to AV1	1080p	1	0	8	1	0	49	579	14
YUV to AVC	1080p	32	0	8	0	0	99	1325	2
YUV to HEVC	1080p	32	0	8	0	0	99	1400	3
YUV to AV1	1080p	32	0	8	0	0	99	1210	2
AVC to AVC	1080p	1	1	8	0	43	84	998	16
AVC to HEVC	1080p	1	1	8	0	45	82	1059	17
AVC to AV1	1080p	1	1	8	0	41	81	929	16
HEVC to AVC	1080p	1	1	8	0	39	86	1002	12
HEVC to HEVC	1080p	1	1	8	0	41	85	1090	13
HEVC to AV1	1080p	1	1	8	0	36	83	932	13
VP9 to AVC	1080p	1	1	8	0	21	42	542	6
VP9 to HEVC	1080p	1	1	8	0	22	39	535	6
VP9 to AV1	1080p	1	1	8	0	22	46	536	7
AVC to AVC	1080p	32	1	8	0	76	100	977	1
AVC to HEVC	1080p	32	1	8	0	84	99	1104	1
AVC to AV1	1080p	32	1	8	0	76	100	1065	1
HEVC to AVC	1080p	32	1	8	0	70	99	1021	1
HEVC to HEVC	1080p	32	1	8	0	77	99	1157	1
HEVC to AV1	1080p	32	1	8	0	68	100	1091	1
VP9 to AVC	1080p	32	1	8	0	70	99	1117	0
VP9 to HEVC	1080p	32	1	8	0	74	99	1260	0
VP9 to AV1	1080p	32	1	8	0	72	99	1158	1
AVC to YUV	1080p	1	0	10	1	28	0	666	7
HEVC to YUV	1080p	1	0	10	1	25	0	673	6
VP9 to YUV	1080p	1	0	10	1	22	0	451	4
AVC to YUV	1080p	40	0	10	0	62	0	1077	0
HEVC to YUV	1080p	40	0	10	0	70	0	1072	0
VP9 to YUV	1080p	40	0	10	0	69	0	1027	0
YUV to AVC	1080p	1	0	10	1	0	36	467	20
YUV to HEVC	1080p	1	0	10	1	0	34	472	23
YUV to AV1	1080p	1	0	10	1	0	35	419	20
YUV to AVC	1080p	32	0	10	0	0	65	854	6
YUV to HEVC	1080p	32	0	10	0	0	59	845	6
YUV to AV1	1080p	32	0	10	0	0	66	801	8
AVC to YUV	720p	1	0	8	1	41	0	1157	12
HEVC to YUV	720p	1	0	8	1	38	0	1222	8
VP9 to YUV	720p	1	0	8	1	32	0	1014	6
AVC to YUV	720p	100	0	8	0	100	0	2756	0

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	95	0	2945	0
VP9 to YUV	720p	100	0	8	0	95	0	2662	0
YUV to AVC	720p	1	0	8	1	0	33	980	12
YUV to HEVC	720p	1	0	8	1	0	34	1025	11
YUV to AV1	720p	1	0	8	1	0	32	772	9
YUV to AVC	720p	64	0	8	0	0	99	2641	2
YUV to HEVC	720p	64	0	8	0	0	99	2710	2
YUV to AV1	720p	64	0	8	0	0	100	2178	1
AVC to AVC	720p	1	1	8	0	51	50	1345	15
AVC to HEVC	720p	1	1	8	0	51	50	1349	16
AVC to AV1	720p	1	1	8	0	43	49	1062	15
HEVC to AVC	720p	1	1	8	0	45	50	1370	11
HEVC to HEVC	720p	1	1	8	0	45	50	1373	11
HEVC to AV1	720p	1	1	8	0	38	48	1064	11
VP9 to AVC	720p	1	1	8	0	31	35	977	7
VP9 to HEVC	720p	1	1	8	0	31	34	982	7
VP9 to AV1	720p	1	1	8	0	31	45	983	8
AVC to AVC	720p	64	1	8	0	100	100	2171	0
AVC to HEVC	720p	64	1	8	0	100	100	2256	0
AVC to AV1	720p	64	1	8	0	94	100	1874	0
HEVC to AVC	720p	64	1	8	0	94	100	2173	0
HEVC to HEVC	720p	64	1	8	0	95	100	2259	0
HEVC to AV1	720p	64	1	8	0	88	100	1874	0
VP9 to AVC	720p	64	1	8	0	100	100	2413	0
VP9 to HEVC	720p	64	1	8	0	100	100	2483	0
VP9 to AV1	720p	64	1	8	0	98	100	2006	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	59.24	57.43	0.04
YUV to HEVC	8k	1	55.46	60.32	54.14	0.96
YUV to AVC	4k	1	15.34	17.77	15.06	0.06
YUV to HEVC	4k	1	16.57	18.87	15.66	0.1
YUV to AV1	4k	1	21.75	25.56	16.07	0.65
YUV to AVC	4k	4	17.02	21.3	15	2.9
YUV to HEVC	4k	4	18.37	22.46	15.57	2.21
YUV to AV1	4k	4	23.01	27.62	16.27	1.01
YUV to AVC	4k	8	20.36	34.15	15.05	19.85
YUV to HEVC	4k	8	22.35	35.11	15.61	20.66
YUV to AV1	4k	8	37.24	45.74	21.92	10.51
YUV to AVC	1080p	1	4.68	6.2	4.45	0.12
YUV to HEVC	1080p	1	4.94	5.41	4.69	0.02
YUV to AV1	1080p	1	6.7	7.48	5.04	0.04
YUV to AVC	1080p	32	6.05	10.3	4.63	0.36
YUV to HEVC	1080p	32	6.81	11.7	4.84	1.18
YUV to AV1	1080p	32	41.02	46.63	34.71	1.37
YUV to AVC	720p	1	2.82	3.45	2.48	0.01
YUV to HEVC	720p	1	2.85	3.2	2.67	0.01
YUV to AV1	720p	1	3.8	4.17	2.95	0.01
YUV to AVC	720p	64	5.45	10.3	3.2	0.89
YUV to HEVC	720p	64	5.63	10.24	3.39	0.92
YUV to AV1	720p	64	39.77	47.38	31.67	3.37

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	6
HEVC to RGBA	8k	1	4	0	154	7
VP9 to RGBA	8k	1	22	0	40	3
AVC to RGBA	4k	1	21	0	145	8
AVC to RGBA	4k	16	93	3	585	2
HEVC to RGBA	4k	1	22	0	174	10
HEVC to RGBA	4k	16	92	4	670	2
VP9 to RGBA	4k	1	22	0	166	4
VP9 to RGBA	4k	16	94	4	680	1
AVC to RGBA	1080p	40	92	14	1894	1
HEVC to RGBA	1080p	40	94	14	2011	1
VP9 to RGBA	1080p	40	94	17	2440	0
AVC to RGBA	720p	100	97	18	2599	0
HEVC to RGBA	720p	100	90	20	2823	0
VP9 to RGBA	720p	64	94	18	2752	0

5. T1A – Streaming Ladder Generation

5.1 Transcoding

5.1.1 Description

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

Decoded YUV frame is kept on device.

The YUV frames are encoded with hardware encoder.

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

5.1.2 Command line

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

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

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

Input: 1080p

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

5.2 Streaming Ladder Generation Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	8	33	94	2	433	4
AVC to HEVC	8	32	94	2	468	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	395	4
VP9 to AVC	8	39	93	2	432	4
VP9 to HEVC	8	38	93	2	466	4
VP9 to AV1	8	28	93	2	392	3

6. T1A – RGBA Encoding

6.1 Encoding

6.1.1 Description

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

RGBA frame is uploaded and encoded by hardware encoder.

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

6.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

6.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	52	161	66
RGBA to HEVC	4k	1	1	46	161	66
RGBA to AV1	4k	1	1	55	164	67
RGBA to AVC	4k	4	0	69	164	38
RGBA to HEVC	4k	4	0	61	169	40
RGBA to AV1	4k	4	0	61	172	41
RGBA to AVC	4k	8	0	59	176	32
RGBA to HEVC	4k	8	0	56	176	28
RGBA to AV1	4k	8	0	62	177	29
RGBA to AVC	1080p	1	1	33	427	44
RGBA to HEVC	1080p	1	1	31	426	48
RGBA to AV1	1080p	1	1	34	409	42
RGBA to AVC	1080p	16	0	58	647	15
RGBA to HEVC	1080p	16	0	53	651	15
RGBA to AV1	1080p	16	0	56	641	15
RGBA to AVC	1080p	32	0	57	668	8
RGBA to HEVC	1080p	32	0	53	671	8
RGBA to AV1	1080p	32	0	57	643	9
RGBA to AVC	720p	1	1	28	682	57
RGBA to HEVC	720p	1	1	29	770	63
RGBA to AV1	720p	1	1	34	739	63
RGBA to AVC	720p	16	0	48	1214	34
RGBA to HEVC	720p	16	0	46	1195	33
RGBA to AV1	720p	16	0	55	1136	30
RGBA to AVC	720p	32	0	45	1165	31
RGBA to HEVC	720p	32	0	45	1170	30
RGBA to AV1	720p	32	0	57	1169	27

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	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	0	0	1	96	304	16
YUV to HEVC	4k	4	0	0	1	96	332	13
YUV to AV1	4k	4	0	0	1	96	288	13
YUV to AVC	4k	4	0	0	2	95	296	20
YUV to HEVC	4k	4	0	0	2	97	172	8
YUV to AV1	4k	4	0	0	2	98	140	7
YUV to AVC	4k	4	0	0	3	95	300	20
YUV to HEVC	4k	4	0	0	3	99	100	5
YUV to AV1	4k	4	0	0	3	97	76	4
YUV to AVC	4k	4	0	1	1	98	188	8
YUV to HEVC	4k	4	0	1	1	95	236	21
YUV to AVC	4k	4	0	1	2	98	188	8
YUV to HEVC	4k	4	0	1	2	98	108	5
YUV to AVC	4k	4	0	1	3	98	188	9
YUV to HEVC	4k	4	0	1	3	99	68	4
YUV to AVC	4k	4	4	0	1	100	198	10
YUV to HEVC	4k	4	4	0	1	99	228	10
YUV to AV1	4k	4	4	0	1	99	200	9
YUV to AVC	4k	4	4	0	2	100	198	9
YUV to HEVC	4k	4	4	0	2	100	136	6
YUV to AV1	4k	4	4	0	2	99	116	6
YUV to AVC	4k	4	4	0	3	99	198	9
YUV to HEVC	4k	4	4	0	3	99	88	4
YUV to AV1	4k	4	4	0	3	99	72	4
YUV to AVC	4k	4	4	1	1	100	140	6
YUV to HEVC	4k	4	4	1	1	99	180	8
YUV to AVC	4k	4	4	1	2	100	140	7
YUV to HEVC	4k	4	4	1	2	98	96	5
YUV to AVC	4k	4	4	1	3	99	140	6
YUV to HEVC	4k	4	4	1	3	98	64	4
YUV to AVC	4k	4	16	0	1	99	196	9
YUV to HEVC	4k	4	16	0	1	100	228	10
YUV to AV1	4k	4	16	0	1	100	200	9
YUV to AVC	4k	4	16	0	2	100	196	10
YUV to HEVC	4k	4	16	0	2	99	136	7
YUV to AV1	4k	4	16	0	2	100	116	6
YUV to AVC	4k	4	16	0	3	100	196	9
YUV to HEVC	4k	4	16	0	3	100	88	5

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	4k	4	16	0	3	100	72	4
YUV to AVC	4k	4	16	1	1	100	140	6
YUV to HEVC	4k	4	16	1	1	100	180	9
YUV to AVC	4k	4	16	1	2	99	140	6
YUV to HEVC	4k	4	16	1	2	100	96	6
YUV to AVC	4k	4	16	1	3	100	140	8
YUV to HEVC	4k	4	16	1	3	98	64	4
YUV to AVC	4k	4	40	0	1	100	196	9
YUV to HEVC	4k	4	40	0	1	99	224	10
YUV to AV1	4k	4	40	0	1	100	196	11
YUV to AVC	4k	4	40	0	2	100	196	10
YUV to HEVC	4k	4	40	0	2	99	136	7
YUV to AV1	4k	4	40	0	2	100	112	5
YUV to AVC	4k	4	40	0	3	99	196	9
YUV to HEVC	4k	4	40	0	3	99	88	4
YUV to AV1	4k	4	40	0	3	100	72	4
YUV to AVC	4k	4	40	1	1	99	136	6
YUV to HEVC	4k	4	40	1	1	99	176	8
YUV to AVC	4k	4	40	1	2	99	136	6
YUV to HEVC	4k	4	40	1	2	99	92	5
YUV to AVC	4k	4	40	1	3	99	136	7
YUV to HEVC	4k	4	40	1	3	100	64	3
YUV to AVC	1080p	20	0	0	1	99	1280	5
YUV to HEVC	1080p	20	0	0	1	99	1360	5
YUV to AV1	1080p	20	0	0	1	99	1180	4
YUV to AVC	1080p	20	0	0	2	99	1280	4
YUV to HEVC	1080p	20	0	0	2	99	686	2
YUV to AV1	1080p	20	0	0	2	99	560	2
YUV to AVC	1080p	20	0	0	3	99	1280	4
YUV to HEVC	1080p	20	0	0	3	99	400	2
YUV to AV1	1080p	20	0	0	3	100	300	1
YUV to AVC	1080p	20	0	1	1	100	760	3
YUV to HEVC	1080p	20	0	1	1	100	960	3
YUV to AVC	1080p	20	0	1	2	99	760	2
YUV to HEVC	1080p	20	0	1	2	99	440	2
YUV to AVC	1080p	20	0	1	3	100	760	2
YUV to HEVC	1080p	20	0	1	3	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	100	516	2
YUV to AV1	1080p	20	4	0	2	99	420	2
YUV to AVC	1080p	20	4	0	3	100	700	2
YUV to HEVC	1080p	20	4	0	3	99	340	2
YUV to AV1	1080p	20	4	0	3	99	266	1
YUV to AVC	1080p	20	4	1	1	100	501	2
YUV to HEVC	1080p	20	4	1	1	100	658	2
YUV to AVC	1080p	20	4	1	2	99	500	2
YUV to HEVC	1080p	20	4	1	2	100	360	1
YUV to AVC	1080p	20	4	1	3	100	500	2
YUV to HEVC	1080p	20	4	1	3	100	240	1
YUV to AVC	1080p	20	16	0	1	100	700	2
YUV to HEVC	1080p	20	16	0	1	99	820	3
YUV to AV1	1080p	20	16	0	1	99	719	2
YUV to AVC	1080p	20	16	0	2	99	700	2
YUV to HEVC	1080p	20	16	0	2	100	502	2
YUV to AV1	1080p	20	16	0	2	100	420	1
YUV to AVC	1080p	20	16	0	3	99	700	2
YUV to HEVC	1080p	20	16	0	3	99	340	1
YUV to AV1	1080p	20	16	0	3	100	261	1
YUV to AVC	1080p	20	16	1	1	99	500	2
YUV to HEVC	1080p	20	16	1	1	99	642	2
YUV to AVC	1080p	20	16	1	2	100	500	2
YUV to HEVC	1080p	20	16	1	2	99	360	1
YUV to AVC	1080p	20	16	1	3	99	500	2
YUV to HEVC	1080p	20	16	1	3	99	240	1
YUV to AVC	1080p	20	40	0	1	99	690	2
YUV to HEVC	1080p	20	40	0	1	99	803	3
YUV to AV1	1080p	20	40	0	1	99	701	2
YUV to AVC	1080p	20	40	0	2	99	691	2
YUV to HEVC	1080p	20	40	0	2	99	500	2
YUV to AV1	1080p	20	40	0	2	100	420	1
YUV to AVC	1080p	20	40	0	3	99	688	2
YUV to HEVC	1080p	20	40	0	3	100	337	1
YUV to AV1	1080p	20	40	0	3	100	260	1
YUV to AVC	1080p	20	40	1	1	100	500	2
YUV to HEVC	1080p	20	40	1	1	100	640	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	2	100	500	2
YUV to HEVC	1080p	20	40	1	2	100	360	2
YUV to AVC	1080p	20	40	1	3	100	500	2
YUV to HEVC	1080p	20	40	1	3	100	240	1
YUV to AVC	720p	40	0	0	1	93	2302	3
YUV to HEVC	720p	40	0	0	1	91	2321	3
YUV to AV1	720p	40	0	0	1	96	2012	2
YUV to AVC	720p	40	0	0	2	92	2298	3
YUV to HEVC	720p	40	0	0	2	99	1520	1
YUV to AV1	720p	40	0	0	2	99	1203	1
YUV to AVC	720p	40	0	0	3	93	2298	3
YUV to HEVC	720p	40	0	0	3	99	920	1
YUV to AV1	720p	40	0	0	3	100	654	1
YUV to AVC	720p	40	0	1	1	99	1690	1
YUV to HEVC	720p	40	0	1	1	99	2080	2
YUV to AVC	720p	40	0	1	2	99	1687	1
YUV to HEVC	720p	40	0	1	2	99	960	1
YUV to AVC	720p	40	0	1	3	99	1691	1
YUV to HEVC	720p	40	0	1	3	100	611	1
YUV to AVC	720p	40	4	0	1	100	1421	1
YUV to HEVC	720p	40	4	0	1	100	1359	1
YUV to AV1	720p	40	4	0	1	100	1003	1
YUV to AVC	720p	40	4	0	2	100	1414	1
YUV to HEVC	720p	40	4	0	2	99	1080	1
YUV to AV1	720p	40	4	0	2	99	880	1
YUV to AVC	720p	40	4	0	3	100	1422	1
YUV to HEVC	720p	40	4	0	3	99	720	1
YUV to AV1	720p	40	4	0	3	99	560	0
YUV to AVC	720p	40	4	1	1	99	1077	1
YUV to HEVC	720p	40	4	1	1	100	1321	1
YUV to AVC	720p	40	4	1	2	99	1075	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	100	520	1
YUV to AVC	720p	40	16	0	1	100	1400	1
YUV to HEVC	720p	40	16	0	1	100	1357	1
YUV to AV1	720p	40	16	0	1	100	1048	1
YUV to AVC	720p	40	16	0	2	100	1400	1
YUV to HEVC	720p	40	16	0	2	99	1080	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	720p	40	16	0	2	99	880	1
YUV to AVC	720p	40	16	0	3	100	1400	1
YUV to HEVC	720p	40	16	0	3	99	720	1
YUV to AV1	720p	40	16	0	3	99	560	1
YUV to AVC	720p	40	16	1	1	99	1046	1
YUV to HEVC	720p	40	16	1	1	100	1320	1
YUV to AVC	720p	40	16	1	2	99	1050	1
YUV to HEVC	720p	40	16	1	2	99	760	1
YUV to AVC	720p	40	16	1	3	99	1049	1
YUV to HEVC	720p	40	16	1	3	100	520	1
YUV to AVC	720p	40	40	0	1	100	1360	1
YUV to HEVC	720p	40	40	0	1	100	1320	1
YUV to AV1	720p	40	40	0	1	100	1040	1
YUV to AVC	720p	40	40	0	2	100	1360	1
YUV to HEVC	720p	40	40	0	2	99	1040	1
YUV to AV1	720p	40	40	0	2	99	846	1
YUV to AVC	720p	40	40	0	3	100	1360	1
YUV to HEVC	720p	40	40	0	3	99	720	1
YUV to AV1	720p	40	40	0	3	99	560	1
YUV to AVC	720p	40	40	1	1	99	1040	1
YUV to HEVC	720p	40	40	1	1	100	1319	1
YUV to AVC	720p	40	40	1	2	99	1040	1
YUV to HEVC	720p	40	40	1	2	99	759	1
YUV to AVC	720p	40	40	1	3	99	1040	1
YUV to HEVC	720p	40	40	1	3	99	520	1

8. T1A – Capped CRF

8.1 Encoding with lookaheadDepth

8.1.1 Description

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

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

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

8.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

8.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	19	100	702	2
YUV to HEVC	1080p	20	0	0	1	19	99	820	3
YUV to AV1	1080p	20	0	0	1	19	99	720	2
YUV to AVC	1080p	20	0	0	2	19	99	700	2
YUV to HEVC	1080p	20	0	0	2	19	100	520	2
YUV to AV1	1080p	20	0	0	2	19	99	440	1
YUV to AVC	1080p	20	0	0	3	19	99	702	2
YUV to HEVC	1080p	20	0	0	3	19	99	340	1
YUV to AV1	1080p	20	0	0	3	19	99	280	1
YUV to AVC	1080p	20	0	1	1	19	99	501	2
YUV to HEVC	1080p	20	0	1	1	19	99	660	2
YUV to AVC	1080p	20	0	1	2	19	99	501	2
YUV to HEVC	1080p	20	0	1	2	19	100	360	1
YUV to AVC	1080p	20	0	1	3	19	99	503	2
YUV to HEVC	1080p	20	0	1	3	19	100	240	1
YUV to AVC	1080p	20	4	0	1	19	99	700	2
YUV to HEVC	1080p	20	4	0	1	19	99	820	3
YUV to AV1	1080p	20	4	0	1	19	100	720	2
YUV to AVC	1080p	20	4	0	2	19	99	700	2
YUV to HEVC	1080p	20	4	0	2	19	99	518	2
YUV to AV1	1080p	20	4	0	2	19	99	420	1
YUV to AVC	1080p	20	4	0	3	19	99	700	2
YUV to HEVC	1080p	20	4	0	3	19	100	340	1
YUV to AV1	1080p	20	4	0	3	19	100	268	1
YUV to AVC	1080p	20	4	1	1	19	99	500	2
YUV to HEVC	1080p	20	4	1	1	19	99	660	2
YUV to AVC	1080p	20	4	1	2	19	99	501	2
YUV to HEVC	1080p	20	4	1	2	19	99	360	1
YUV to AVC	1080p	20	4	1	3	19	99	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	820	3
YUV to AV1	1080p	20	16	0	1	19	99	720	2
YUV to AVC	1080p	20	16	0	2	19	99	700	3
YUV to HEVC	1080p	20	16	0	2	19	100	500	2
YUV to AV1	1080p	20	16	0	2	19	100	420	1
YUV to AVC	1080p	20	16	0	3	19	99	700	3
YUV to HEVC	1080p	20	16	0	3	19	100	340	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	3	19	99	260	1
YUV to AVC	1080p	20	16	1	1	19	100	500	2
YUV to HEVC	1080p	20	16	1	1	19	99	653	2
YUV to AVC	1080p	20	16	1	2	19	99	500	2
YUV to HEVC	1080p	20	16	1	2	19	100	360	1
YUV to AVC	1080p	20	16	1	3	19	100	500	2
YUV to HEVC	1080p	20	16	1	3	19	99	240	1
YUV to AVC	1080p	20	40	0	1	19	99	693	2
YUV to HEVC	1080p	20	40	0	1	19	99	800	3
YUV to AV1	1080p	20	40	0	1	19	99	701	2
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	99	420	2
YUV to AVC	1080p	20	40	0	3	19	99	692	2
YUV to HEVC	1080p	20	40	0	3	19	100	340	1
YUV to AV1	1080p	20	40	0	3	19	99	260	1
YUV to AVC	1080p	20	40	1	1	19	99	500	2
YUV to HEVC	1080p	20	40	1	1	19	99	640	2
YUV to AVC	1080p	20	40	1	2	19	100	500	2
YUV to HEVC	1080p	20	40	1	2	19	100	360	1
YUV to AVC	1080p	20	40	1	3	19	100	500	2
YUV to HEVC	1080p	20	40	1	3	19	100	240	1
YUV to AVC	1080p	20	0	0	1	23	99	700	3
YUV to HEVC	1080p	20	0	0	1	23	99	820	3
YUV to AV1	1080p	20	0	0	1	23	99	720	2
YUV to AVC	1080p	20	0	0	2	23	99	700	2
YUV to HEVC	1080p	20	0	0	2	23	99	520	2
YUV to AV1	1080p	20	0	0	2	23	100	438	1
YUV to AVC	1080p	20	0	0	3	23	99	701	2
YUV to HEVC	1080p	20	0	0	3	23	99	340	1
YUV to AV1	1080p	20	0	0	3	23	99	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	501	2
YUV to HEVC	1080p	20	0	1	2	23	100	360	1
YUV to AVC	1080p	20	0	1	3	23	99	502	2
YUV to HEVC	1080p	20	0	1	3	23	100	240	1
YUV to AVC	1080p	20	4	0	1	23	99	700	2
YUV to HEVC	1080p	20	4	0	1	23	99	820	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	23	99	720	2
YUV to AVC	1080p	20	4	0	2	23	99	700	2
YUV to HEVC	1080p	20	4	0	2	23	99	520	2
YUV to AV1	1080p	20	4	0	2	23	99	420	1
YUV to AVC	1080p	20	4	0	3	23	99	700	2
YUV to HEVC	1080p	20	4	0	3	23	99	340	1
YUV to AV1	1080p	20	4	0	3	23	100	268	1
YUV to AVC	1080p	20	4	1	1	23	99	501	2
YUV to HEVC	1080p	20	4	1	1	23	99	660	2
YUV to AVC	1080p	20	4	1	2	23	100	500	2
YUV to HEVC	1080p	20	4	1	2	23	99	360	1
YUV to AVC	1080p	20	4	1	3	23	99	500	2
YUV to HEVC	1080p	20	4	1	3	23	100	240	1
YUV to AVC	1080p	20	16	0	1	23	100	700	2
YUV to HEVC	1080p	20	16	0	1	23	99	820	3
YUV to AV1	1080p	20	16	0	1	23	99	720	2
YUV to AVC	1080p	20	16	0	2	23	99	700	2
YUV to HEVC	1080p	20	16	0	2	23	99	501	2
YUV to AV1	1080p	20	16	0	2	23	100	420	2
YUV to AVC	1080p	20	16	0	3	23	99	700	2
YUV to HEVC	1080p	20	16	0	3	23	99	340	1
YUV to AV1	1080p	20	16	0	3	23	99	261	1
YUV to AVC	1080p	20	16	1	1	23	100	500	2
YUV to HEVC	1080p	20	16	1	1	23	99	645	2
YUV to AVC	1080p	20	16	1	2	23	100	500	2
YUV to HEVC	1080p	20	16	1	2	23	100	360	1
YUV to AVC	1080p	20	16	1	3	23	100	500	2
YUV to HEVC	1080p	20	16	1	3	23	100	240	1
YUV to AVC	1080p	20	40	0	1	23	99	691	2
YUV to HEVC	1080p	20	40	0	1	23	99	801	3
YUV to AV1	1080p	20	40	0	1	23	99	702	3
YUV to AVC	1080p	20	40	0	2	23	99	690	2
YUV to HEVC	1080p	20	40	0	2	23	99	500	2
YUV to AV1	1080p	20	40	0	2	23	99	420	1
YUV to AVC	1080p	20	40	0	3	23	99	690	2
YUV to HEVC	1080p	20	40	0	3	23	99	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	100	360	1
YUV to AVC	1080p	20	40	1	3	23	99	500	2
YUV to HEVC	1080p	20	40	1	3	23	99	240	1
YUV to AVC	1080p	20	0	0	1	27	99	700	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	702	2
YUV to HEVC	1080p	20	0	0	2	27	100	520	2
YUV to AV1	1080p	20	0	0	2	27	100	440	1
YUV to AVC	1080p	20	0	0	3	27	99	701	2
YUV to HEVC	1080p	20	0	0	3	27	100	340	1
YUV to AV1	1080p	20	0	0	3	27	99	280	1
YUV to AVC	1080p	20	0	1	1	27	99	501	2
YUV to HEVC	1080p	20	0	1	1	27	99	660	2
YUV to AVC	1080p	20	0	1	2	27	99	500	1
YUV to HEVC	1080p	20	0	1	2	27	99	360	1
YUV to AVC	1080p	20	0	1	3	27	99	502	2
YUV to HEVC	1080p	20	0	1	3	27	99	240	1
YUV to AVC	1080p	20	4	0	1	27	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	520	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	267	1
YUV to AVC	1080p	20	4	1	1	27	100	500	2
YUV to HEVC	1080p	20	4	1	1	27	99	660	2
YUV to AVC	1080p	20	4	1	2	27	100	500	2
YUV to HEVC	1080p	20	4	1	2	27	100	360	1
YUV to AVC	1080p	20	4	1	3	27	99	500	2
YUV to HEVC	1080p	20	4	1	3	27	100	240	1
YUV to AVC	1080p	20	16	0	1	27	99	700	2
YUV to HEVC	1080p	20	16	0	1	27	99	820	3
YUV to AV1	1080p	20	16	0	1	27	99	720	2
YUV to AVC	1080p	20	16	0	2	27	99	700	2
YUV to HEVC	1080p	20	16	0	2	27	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	699	3
YUV to HEVC	1080p	20	16	0	3	27	100	340	2
YUV to AV1	1080p	20	16	0	3	27	100	262	1
YUV to AVC	1080p	20	16	1	1	27	99	500	2
YUV to HEVC	1080p	20	16	1	1	27	99	648	2
YUV to AVC	1080p	20	16	1	2	27	99	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	99	240	1
YUV to AVC	1080p	20	40	0	1	27	99	688	2
YUV to HEVC	1080p	20	40	0	1	27	99	803	3
YUV to AV1	1080p	20	40	0	1	27	99	701	3
YUV to AVC	1080p	20	40	0	2	27	99	689	2
YUV to HEVC	1080p	20	40	0	2	27	100	500	2
YUV to AV1	1080p	20	40	0	2	27	99	420	2
YUV to AVC	1080p	20	40	0	3	27	99	693	2
YUV to HEVC	1080p	20	40	0	3	27	100	338	2
YUV to AV1	1080p	20	40	0	3	27	99	261	1
YUV to AVC	1080p	20	40	1	1	27	99	500	2
YUV to HEVC	1080p	20	40	1	1	27	99	640	2
YUV to AVC	1080p	20	40	1	2	27	99	500	2
YUV to HEVC	1080p	20	40	1	2	27	100	360	1
YUV to AVC	1080p	20	40	1	3	27	99	500	2
YUV to HEVC	1080p	20	40	1	3	27	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	308	12
AVC to AV1	1	13	20	4	263	11
HEVC to AVC	1	13	20	4	289	15
HEVC to HEVC	1	13	21	5	307	16
HEVC to AV1	1	12	20	4	262	14
VP9 to AVC	1	17	20	4	288	11
VP9 to HEVC	1	18	21	5	305	12
VP9 to AV1	1	15	20	4	262	12
AVC to AVC	16	71	93	24	1043	3
AVC to HEVC	16	78	94	27	1154	3
AVC to AV1	16	70	92	25	1108	3
HEVC to AVC	16	71	93	24	1088	4
HEVC to HEVC	16	78	94	28	1215	4
HEVC to AV1	16	68	93	25	1136	4
VP9 to AVC	16	83	94	24	1040	3
VP9 to HEVC	16	89	92	26	1168	3
VP9 to AV1	16	84	94	25	1120	3
AVC to AVC	32	74	94	23	960	1
AVC to HEVC	32	80	92	26	1085	1
AVC to AV1	32	76	93	25	1056	1
HEVC to AVC	32	73	95	25	998	1
HEVC to HEVC	32	78	92	27	1139	2
HEVC to AV1	32	73	94	26	1090	2
VP9 to AVC	32	80	92	23	992	1
VP9 to HEVC	32	89	93	26	1120	1
VP9 to AV1	32	85	95	26	1089	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	JOB	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	279
deeplabv3_FRP	8	10000	rgb	nchw	image	257x257x3	359
resnet50	8	10000	rgb	nchw	image	224x224x3	228
mobilenetv2	8	10000	rgb	nchw	image	224x224x3	1110
deeplabv3	8	10000	rgb	nchw	image	257x257x3	160
yolov4	8	10000	bgr	nchw	image	416x416x3	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	836
mobilenetv2_nchw_keras_96x160	1	10000	rgb	nchw	image	96x160x3	2343
mobilenetv2_nchw_keras_96x160	8	10000	rgb	nchw	image	96x160x3	2221.26
mobilenetv2_nchw_keras_96x160	16	10000	rgb	nchw	image	96x160x3	2240.17

11.2 AI Encoding with 2D Engine

11.2.1 Description BG Filter

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

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

11.2.2 Description ROI Filter

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

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

11.2.3 Description PRE Filter

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

11.2.4 Command line BG

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

11.2.5 Command line ROI

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

11.2.6 Command line PRE

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

11.2.7 AI Encoding with 2D Engine Performance Results

Filter	Model	Resolution	Session Number	Average FPS per session
ROI	network_binary_yolov4_head	1920x1080	1	79
ROI	network_binary_yolov4_head	1920x1080	8	22
ROI	network_binary_yolov4_head	1920x1080	32	5
BG	segm32_tflite_nchw_bgr	1920x1080	1	58
BG	segm32_tflite_nchw_bgr	1920x1080	8	39
BG	segm32_tflite_nchw_bgr	1920x1080	32	14
PRE	hw_lanczos_network_binary	1920x1080	8	66
PRE	hw_lanczos_network_binary	1920x1080	16	34
PRE	hw_bicubic_network_binary	1920x1080	8	67
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	43	65
YUV to AVC	8k	1	0	8	1	0	27	29	105
YUV to HEVC	8k	1	0	8	1	0	40	29	105
AVC to AVC	8k	1	1	8	1	3	92	54	6
AVC to HEVC	8k	1	1	8	1	3	94	71	7
HEVC to AVC	8k	1	1	8	1	2	97	53	6
HEVC to HEVC	8k	1	1	8	1	3	98	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	15	15	104
YUV to HEVC	8k	1	0	10	1	0	10	14	104
AVC to YUV	4k	1	0	8	1	10	0	168	61
HEVC to YUV	4k	1	0	8	1	9	0	170	61
AVC to YUV	4k	16	0	8	0	97	0	481	29
HEVC to YUV	4k	16	0	8	0	99	0	500	33
YUV to AVC	4k	1	0	8	1	0	41	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	247	42
YUV to HEVC	4k	4	0	8	0	0	76	256	46
YUV to AV1	4k	4	0	8	0	0	81	233	40
YUV to AVC	4k	8	0	8	0	0	99	309	33
YUV to HEVC	4k	8	0	8	0	0	99	339	37
YUV to AV1	4k	8	0	8	0	0	100	288	29
AVC to AVC	4k	1	1	8	1	10	91	224	7
AVC to HEVC	4k	1	1	8	1	12	92	279	8
AVC to AV1	4k	1	1	8	1	11	94	264	8
HEVC to AVC	4k	1	1	8	1	9	93	225	7
HEVC to HEVC	4k	1	1	8	1	10	94	282	7
HEVC to AV1	4k	1	1	8	1	10	95	264	7
AVC to AVC	4k	4	1	8	0	62	96	240	3
AVC to HEVC	4k	4	1	8	0	67	96	295	3
AVC to AV1	4k	4	1	8	0	57	95	272	3
HEVC to AVC	4k	4	1	8	0	58	97	234	3
HEVC to HEVC	4k	4	1	8	0	58	96	293	3
HEVC to AV1	4k	4	1	8	0	45	94	273	3
AVC to AVC	4k	8	1	8	0	69	99	218	2

AVC to HEVC	4k	8	1	8	0	75	100	281	3
AVC to AV1	4k	8	1	8	0	67	99	271	2
HEVC to AVC	4k	8	1	8	0	63	100	220	2
HEVC to HEVC	4k	8	1	8	0	67	100	284	2
HEVC to AV1	4k	8	1	8	0	58	100	277	2
AVC to YUV	4k	1	0	10	1	8	0	108	59
HEVC to YUV	4k	1	0	10	1	7	0	109	59
AVC to YUV	4k	16	0	10	0	90	0	272	30
HEVC to YUV	4k	16	0	10	0	84	0	271	33
YUV to AVC	4k	1	0	10	1	0	24	73	82
YUV to HEVC	4k	1	0	10	1	0	20	73	82
YUV to AV1	4k	1	0	10	1	0	22	73	81
YUV to AVC	4k	4	0	10	0	0	54	177	70
YUV to HEVC	4k	4	0	10	0	0	52	183	73
YUV to AV1	4k	4	0	10	0	0	55	165	66
AVC to YUV	1080p	1	0	8	1	21	0	484	49
HEVC to YUV	1080p	1	0	8	1	19	0	496	49
AVC to YUV	1080p	40	0	8	0	82	0	1530	14
HEVC to YUV	1080p	40	0	8	0	77	0	1592	16
YUV to AVC	1080p	1	0	8	1	0	33	431	69
YUV to HEVC	1080p	1	0	8	1	0	31	432	70
YUV to AV1	1080p	1	0	8	1	0	36	410	66
YUV to AVC	1080p	32	0	8	0	0	99	1278	20
YUV to HEVC	1080p	32	0	8	0	0	99	1349	21
YUV to AV1	1080p	32	0	8	0	0	99	1139	15
AVC to AVC	1080p	1	1	8	1	29	57	729	13
AVC to HEVC	1080p	1	1	8	1	27	52	718	13
AVC to AV1	1080p	1	1	8	1	25	57	654	12
HEVC to AVC	1080p	1	1	8	1	26	61	761	14
HEVC to HEVC	1080p	1	1	8	1	26	57	772	14
HEVC to AV1	1080p	1	1	8	1	23	60	685	13
AVC to AVC	1080p	32	1	8	0	74	99	953	2
AVC to HEVC	1080p	32	1	8	0	82	99	1074	2
AVC to AV1	1080p	32	1	8	0	74	99	1032	2
HEVC to AVC	1080p	32	1	8	0	68	99	994	2
HEVC to HEVC	1080p	32	1	8	0	76	99	1127	2
HEVC to AV1	1080p	32	1	8	0	68	99	1056	2
AVC to YUV	1080p	1	0	10	1	16	0	361	49
HEVC to YUV	1080p	1	0	10	1	15	0	364	51
AVC to YUV	1080p	40	0	10	0	58	0	1028	15
HEVC to YUV	1080p	40	0	10	0	73	0	1018	15

YUV to AVC	1080p	1	0	10	1	0	17	235	71
YUV to HEVC	1080p	1	0	10	1	0	16	235	71
YUV to AV1	1080p	1	0	10	1	0	19	229	70
YUV to AVC	1080p	32	0	10	0	0	52	707	32
YUV to HEVC	1080p	32	0	10	0	0	50	702	32
YUV to AV1	1080p	32	0	10	0	0	57	694	31
AVC to YUV	720p	1	0	8	1	28	0	756	39
HEVC to YUV	720p	1	0	8	1	26	0	799	39
AVC to YUV	720p	100	0	8	0	100	0	2513	4
HEVC to YUV	720p	100	0	8	0	93	0	2775	5
YUV to AVC	720p	1	0	8	1	0	32	897	26
YUV to HEVC	720p	1	0	8	1	0	32	879	27
YUV to AV1	720p	1	0	8	1	0	36	789	26
YUV to AVC	720p	64	0	8	0	0	99	2280	4
YUV to HEVC	720p	64	0	8	0	0	99	2304	4
YUV to AV1	720p	64	0	8	0	0	99	1937	3
AVC to AVC	720p	1	1	8	1	28	27	784	11
AVC to HEVC	720p	1	1	8	1	29	27	785	12
AVC to AV1	720p	1	1	8	1	25	30	699	10
HEVC to AVC	720p	1	1	8	1	27	29	844	12
HEVC to HEVC	720p	1	1	8	1	26	29	836	11
HEVC to AV1	720p	1	1	8	1	23	32	744	11
AVC to AVC	720p	64	1	8	0	94	100	2030	1
AVC to HEVC	720p	64	1	8	0	93	100	2113	1
AVC to AV1	720p	64	1	8	0	74	100	1770	1
HEVC to AVC	720p	64	1	8	0	82	99	2024	1
HEVC to HEVC	720p	64	1	8	0	83	100	2118	1
HEVC to AV1	720p	64	1	8	0	65	100	1786	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	57.87	59.69	57.38	0.06
YUV to HEVC	8k	1	54.77	59.54	54.05	0.65
YUV to AVC	4k	1	15.29	17.37	14.96	0.06
YUV to HEVC	4k	1	15.92	18.48	15.04	0.19
YUV to AV1	4k	1	20.05	25.59	16.44	1.23
YUV to AVC	4k	4	15.64	21.82	15	0.51
YUV to HEVC	4k	4	16.27	21.03	15.06	0.54
YUV to AV1	4k	4	20.46	28.58	16.5	1.6
YUV to AVC	4k	8	20.99	36.63	15	24.95
YUV to HEVC	4k	8	21.87	38.45	14.97	26.42
YUV to AV1	4k	8	29.01	45.2	16.68	42.37
YUV to AVC	1080p	1	4.51	6.56	4.4	0.02
YUV to HEVC	1080p	1	4.72	6.21	4.44	0.03
YUV to AV1	1080p	1	6.11	7.56	4.99	0.1
YUV to AVC	1080p	32	19.8	64.27	4.37	24.34
YUV to HEVC	1080p	32	21.41	83.48	4.48	28.93
YUV to AV1	1080p	32	31.5	44.03	5.13	60.68
YUV to AVC	720p	1	2.58	4.17	2.4	0.04
YUV to HEVC	720p	1	2.76	4.2	2.54	0.01
YUV to AV1	720p	1	3.5	4.97	2.96	0.03
YUV to AVC	720p	64	22.39	116.31	2.42	63.04
YUV to HEVC	720p	64	22.3	107.47	2.6	55.75
YUV to AV1	720p	64	29.17	67.48	3.16	88.44

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.rgbx 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	17	56	103
RGBA to HEVC	4k	1	1	15	56	103
RGBA to AVC	4k	4	0	49	145	87
RGBA to HEVC	4k	4	0	45	147	87
RGBA to AVC	4k	8	0	53	152	97
RGBA to HEVC	4k	8	0	43	150	96
RGBA to AVC	1080p	1	1	14	195	76
RGBA to HEVC	1080p	1	1	14	194	76
RGBA to AVC	1080p	16	0	47	596	63
RGBA to HEVC	1080p	16	0	43	590	61
RGBA to AVC	1080p	32	0	46	554	63
RGBA to HEVC	1080p	32	0	41	557	62
RGBA to AVC	720p	1	1	15	373	69
RGBA to HEVC	720p	1	1	14	374	69
RGBA to AVC	720p	16	0	50	1223	33
RGBA to HEVC	720p	16	0	50	1208	32
RGBA to AVC	720p	32	0	48	1165	32
RGBA to HEVC	720p	32	0	49	1180	31

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.

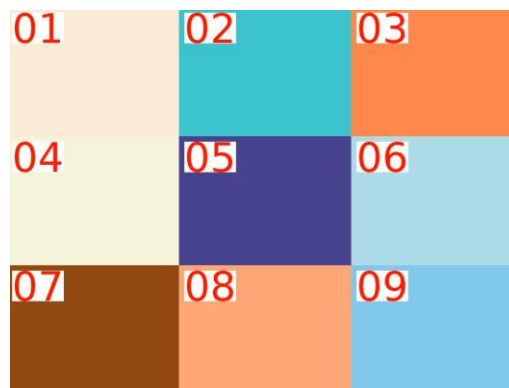
The encoded bitstream is read out through PCIe and written into an output file.

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	150.18	29	67	13	16
1920x1080	4x4	1920x1080	480x270	105.4	43	84	10	16
1920x1080	4x8	1920x1080	480x135	57.18	61	87	5	8
1920x1080	3x3	3840x2160	1280x720	69.76	15	32	22	19
1920x1080	4x4	3840x2160	960x540	55.83	22	46	19	21
1920x1080	4x8	3840x2160	960x270	43.62	38	72	15	18
1920x1080	3x3	7680x4320	2560x1440	19.83	8	8	22	16
1920x1080	4x4	7680x4320	1920x1080	19.11	10	15	22	17
1920x1080	4x8	7680x4320	1920x540	19.32	18	33	22	20
1920x1080	6x6	1920x1080	320x180	50.6	60	90	5	10
1920x1080	7x7	1920x1080	274x154 276x154 274x156 276x156*	37.47	61	90	4	9

*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	18.97	57	1	90

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	104	9
HEVC to YUV	8k	1	0	8	1	10	0	96	16
VP9 to YUV	8k	1	0	8	1	24	0	38	6
YUV to AVC	8k	1	0	8	1	0	97	62	85
YUV to HEVC	8k	1	0	8	1	0	97	82	123
AVC to AVC	8k	1	1	8	1	7	99	49	3
AVC to HEVC	8k	1	1	8	1	7	99	68	8
HEVC to AVC	8k	1	1	8	1	7	100	48	2
HEVC to HEVC	8k	1	1	8	1	6	100	66	12
VP9 to AVC	8k	1	1	8	1	24	45	35	1
VP9 to HEVC	8k	1	1	8	1	24	44	37	1
AVC to YUV	8k	1	0	10	1	10	0	59	9
HEVC to YUV	8k	1	0	10	1	9	0	54	7
VP9 to YUV	8k	1	0	10	1	24	0	33	15
YUV to AVC	8k	1	0	10	1	0	80	42	118
YUV to HEVC	8k	1	0	10	1	0	49	42	122
AVC to YUV	4k	1	0	8	1	19	0	298	18
HEVC to YUV	4k	1	0	8	1	18	0	297	18
VP9 to YUV	4k	1	0	8	1	23	0	137	7
AVC to YUV	4k	16	0	8	0	99	0	482	4
HEVC to YUV	4k	16	0	8	0	101	0	499	5
VP9 to YUV	4k	16	0	8	0	99	0	464	1
YUV to AVC	4k	1	0	8	1	0	92	286	48
YUV to HEVC	4k	1	0	8	1	0	94	322	51
YUV to AV1	4k	1	0	8	1	0	94	281	48
YUV to AVC	4k	4	0	8	0	0	95	301	20
YUV to HEVC	4k	4	0	8	0	0	96	332	14
YUV to AV1	4k	4	0	8	0	0	96	288	13
YUV to AVC	4k	8	0	8	0	0	100	320	8
YUV to HEVC	4k	8	0	8	0	0	99	344	10
YUV to AV1	4k	8	0	8	0	0	100	296	7
AVC to AVC	4k	1	1	8	1	14	93	213	15
AVC to HEVC	4k	1	1	8	1	14	90	264	13
AVC to AV1	4k	1	1	8	1	16	93	254	12
HEVC to AVC	4k	1	1	8	1	13	97	211	14
HEVC to HEVC	4k	1	1	8	1	13	92	264	18
HEVC to AV1	4k	1	1	8	1	14	92	251	15
VP9 to AVC	4k	1	1	8	1	23	40	133	4
VP9 to HEVC	4k	1	1	8	1	24	38	133	6

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	45	133	7
AVC to AVC	4k	4	1	8	0	62	96	232	1
AVC to HEVC	4k	4	1	8	0	68	96	296	7
AVC to AV1	4k	4	1	8	0	57	95	272	2
HEVC to AVC	4k	4	1	8	0	58	96	228	1
HEVC to HEVC	4k	4	1	8	0	61	96	292	8
HEVC to AV1	4k	4	1	8	0	47	95	275	2
VP9 to AVC	4k	4	1	8	0	66	96	228	2
VP9 to HEVC	4k	4	1	8	0	73	96	288	6
VP9 to AV1	4k	4	1	8	0	63	91	260	3
AVC to AVC	4k	8	1	8	0	70	100	203	2
AVC to HEVC	4k	8	1	8	0	76	100	266	2
AVC to AV1	4k	8	1	8	0	70	99	264	1
HEVC to AVC	4k	8	1	8	0	65	99	203	2
HEVC to HEVC	4k	8	1	8	0	70	99	271	2
HEVC to AV1	4k	8	1	8	0	61	100	270	2
VP9 to AVC	4k	8	1	8	0	69	100	211	1
VP9 to HEVC	4k	8	1	8	0	75	99	272	1
VP9 to AV1	4k	8	1	8	0	70	99	272	2
AVC to YUV	4k	1	0	10	1	17	0	217	9
HEVC to YUV	4k	1	0	10	1	16	0	214	8
VP9 to YUV	4k	1	0	10	1	24	0	126	4
AVC to YUV	4k	16	0	10	0	98	0	271	0
HEVC to YUV	4k	16	0	10	0	100	0	268	1
VP9 to YUV	4k	16	0	10	0	100	0	249	0
YUV to AVC	4k	1	0	10	1	0	64	184	63
YUV to HEVC	4k	1	0	10	1	0	55	188	68
YUV to AV1	4k	1	0	10	1	0	61	185	64
YUV to AVC	4k	4	0	10	0	0	88	210	30
YUV to HEVC	4k	4	0	10	0	0	77	241	45
YUV to AV1	4k	4	0	10	0	0	81	236	45
AVC to YUV	1080p	1	0	8	1	35	0	804	15
HEVC to YUV	1080p	1	0	8	1	30	0	779	28
VP9 to YUV	1080p	1	0	8	1	22	0	556	7
AVC to YUV	1080p	40	0	8	0	87	0	1720	0
HEVC to YUV	1080p	40	0	8	0	97	0	1805	1
VP9 to YUV	1080p	40	0	8	0	82	0	1800	0
YUV to AVC	1080p	1	0	8	1	0	52	672	27
YUV to HEVC	1080p	1	0	8	1	0	49	676	26
YUV to AV1	1080p	1	0	8	1	0	52	613	21

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1282	3
YUV to HEVC	1080p	32	0	8	0	0	99	1362	3
YUV to AV1	1080p	32	0	8	0	0	100	1183	2
AVC to AVC	1080p	1	1	8	1	41	79	939	18
AVC to HEVC	1080p	1	1	8	1	43	76	989	21
AVC to AV1	1080p	1	1	8	1	38	78	888	15
HEVC to AVC	1080p	1	1	8	1	34	76	879	30
HEVC to HEVC	1080p	1	1	8	1	36	73	956	30
HEVC to AV1	1080p	1	1	8	1	33	78	842	25
VP9 to AVC	1080p	1	1	8	1	22	41	536	6
VP9 to HEVC	1080p	1	1	8	1	22	39	542	4
VP9 to AV1	1080p	1	1	8	1	21	46	542	6
AVC to AVC	1080p	32	1	8	0	75	99	910	1
AVC to HEVC	1080p	32	1	8	0	84	99	1028	1
AVC to AV1	1080p	32	1	8	0	78	99	998	0
HEVC to AVC	1080p	32	1	8	0	72	100	952	3
HEVC to HEVC	1080p	32	1	8	0	78	100	1070	1
HEVC to AV1	1080p	32	1	8	0	70	100	1024	1
VP9 to AVC	1080p	32	1	8	0	70	100	1056	0
VP9 to HEVC	1080p	32	1	8	0	74	99	1214	0
VP9 to AV1	1080p	32	1	8	0	71	99	1120	0
AVC to YUV	1080p	1	0	10	1	28	0	574	9
HEVC to YUV	1080p	1	0	10	1	26	0	570	7
VP9 to YUV	1080p	1	0	10	1	22	0	423	8
AVC to YUV	1080p	40	0	10	0	66	0	1082	0
HEVC to YUV	1080p	40	0	10	0	69	0	1080	0
VP9 to YUV	1080p	40	0	10	0	78	0	1040	0
YUV to AVC	1080p	1	0	10	1	0	35	460	39
YUV to HEVC	1080p	1	0	10	1	0	32	458	37
YUV to AV1	1080p	1	0	10	1	0	36	430	34
YUV to AVC	1080p	32	0	10	0	0	62	803	7
YUV to HEVC	1080p	32	0	10	0	0	58	802	6
YUV to AV1	1080p	32	0	10	0	0	64	774	7
AVC to YUV	720p	1	0	8	1	43	0	1140	15
HEVC to YUV	720p	1	0	8	1	37	0	1144	24
VP9 to YUV	720p	1	0	8	1	33	0	1024	7
AVC to YUV	720p	100	0	8	0	100	0	2539	0
HEVC to YUV	720p	100	0	8	0	100	0	2921	1
VP9 to YUV	720p	100	0	8	0	100	0	2693	0
YUV to AVC	720p	1	0	8	1	0	31	891	15

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	31	888	15
YUV to AV1	720p	1	0	8	1	0	35	804	13
YUV to AVC	720p	64	0	8	0	0	98	2369	2
YUV to HEVC	720p	64	0	8	0	0	98	2432	2
YUV to AV1	720p	64	0	8	0	0	98	2005	1
AVC to AVC	720p	1	1	8	1	46	47	1243	18
AVC to HEVC	720p	1	1	8	1	47	47	1262	19
AVC to AV1	720p	1	1	8	1	40	51	1092	20
HEVC to AVC	720p	1	1	8	1	40	46	1211	24
HEVC to HEVC	720p	1	1	8	1	39	46	1212	24
HEVC to AV1	720p	1	1	8	1	34	49	1058	21
VP9 to AVC	720p	1	1	8	1	31	35	980	12
VP9 to HEVC	720p	1	1	8	1	32	35	989	10
VP9 to AV1	720p	1	1	8	1	31	44	990	10
AVC to AVC	720p	64	1	8	0	94	100	1986	0
AVC to HEVC	720p	64	1	8	0	95	100	2061	0
AVC to AV1	720p	64	1	8	0	75	100	1742	0
HEVC to AVC	720p	64	1	8	0	84	100	1984	0
HEVC to HEVC	720p	64	1	8	0	84	100	2056	0
HEVC to AV1	720p	64	1	8	0	67	100	1740	0
VP9 to AVC	720p	64	1	8	0	96	100	2239	0
VP9 to HEVC	720p	64	1	8	0	99	100	2304	0
VP9 to AV1	720p	64	1	8	0	76	100	1900	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	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	9	0	91	8
HEVC to YUV	8k	1	0	8	1	9	0	91	6
VP9 to YUV	8k	1	0	8	1	24	0	38	2
YUV to AVC	8k	1	0	8	1	0	88	59	27
YUV to HEVC	8k	1	0	8	1	0	87	74	33
AVC to AVC	8k	1	1	8	1	10	100	49	13
AVC to HEVC	8k	1	1	8	1	9	99	68	4
HEVC to AVC	8k	1	1	8	1	11	100	48	3
HEVC to HEVC	8k	1	1	8	1	9	100	66	2
VP9 to AVC	8k	1	1	8	1	24	45	34	2
VP9 to HEVC	8k	1	1	8	1	24	45	37	2
AVC to YUV	8k	1	0	10	1	9	0	54	7
HEVC to YUV	8k	1	0	10	1	8	0	48	7
VP9 to YUV	8k	1	0	10	1	24	0	33	4
YUV to AVC	8k	1	0	10	1	0	59	34	35
YUV to HEVC	8k	1	0	10	1	0	41	35	35
AVC to YUV	4k	1	0	8	1	19	0	292	7
HEVC to YUV	4k	1	0	8	1	18	0	293	6
VP9 to YUV	4k	1	0	8	1	23	0	136	3
AVC to YUV	4k	16	0	8	0	99	0	484	0
HEVC to YUV	4k	16	0	8	0	99	0	501	0
VP9 to YUV	4k	16	0	8	0	98	0	468	0
YUV to AVC	4k	1	0	8	1	0	74	238	23
YUV to HEVC	4k	1	0	8	1	0	70	244	25
YUV to AV1	4k	1	0	8	1	0	77	230	22
YUV to AVC	4k	4	0	8	0	0	94	305	11
YUV to HEVC	4k	4	0	8	0	0	95	330	11
YUV to AV1	4k	4	0	8	0	0	96	288	11
YUV to AVC	4k	8	0	8	0	0	90	287	17
YUV to HEVC	4k	8	0	8	0	0	99	346	8
YUV to AV1	4k	8	0	8	0	0	100	301	7
AVC to AVC	4k	1	1	8	0	14	97	217	5
AVC to HEVC	4k	1	1	8	0	16	96	273	6
AVC to AV1	4k	1	1	8	0	17	96	260	6
HEVC to AVC	4k	1	1	8	0	16	97	216	5
HEVC to HEVC	4k	1	1	8	0	16	96	277	5
HEVC to AV1	4k	1	1	8	0	16	97	261	5
VP9 to AVC	4k	1	1	8	0	24	41	132	2

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	0	23	38	133	3
VP9 to AV1	4k	1	1	8	0	23	45	133	2
AVC to AVC	4k	4	1	8	0	62	96	237	4
AVC to HEVC	4k	4	1	8	0	68	96	297	4
AVC to AV1	4k	4	1	8	0	57	94	275	4
HEVC to AVC	4k	4	1	8	0	58	96	231	4
HEVC to HEVC	4k	4	1	8	0	62	96	287	4
HEVC to AV1	4k	4	1	8	0	52	95	272	4
VP9 to AVC	4k	4	1	8	0	65	96	232	3
VP9 to HEVC	4k	4	1	8	0	71	96	289	3
VP9 to AV1	4k	4	1	8	0	61	96	273	3
AVC to AVC	4k	8	1	8	0	69	100	209	3
AVC to HEVC	4k	8	1	8	0	77	100	271	3
AVC to AV1	4k	8	1	8	0	72	99	267	3
HEVC to AVC	4k	8	1	8	0	66	99	207	3
HEVC to HEVC	4k	8	1	8	0	69	99	269	3
HEVC to AV1	4k	8	1	8	0	64	99	268	3
VP9 to AVC	4k	8	1	8	0	68	101	213	2
VP9 to HEVC	4k	8	1	8	0	75	100	278	3
VP9 to AV1	4k	8	1	8	0	68	99	275	3
AVC to YUV	4k	1	0	10	1	15	0	201	7
HEVC to YUV	4k	1	0	10	1	14	0	194	6
VP9 to YUV	4k	1	0	10	1	24	0	125	4
AVC to YUV	4k	16	0	10	0	99	0	275	0
HEVC to YUV	4k	16	0	10	0	100	0	273	0
VP9 to YUV	4k	16	0	10	0	100	0	254	0
YUV to AVC	4k	1	0	10	1	0	45	144	26
YUV to HEVC	4k	1	0	10	1	0	42	148	26
YUV to AV1	4k	1	0	10	1	0	47	141	26
YUV to AVC	4k	4	0	10	0	0	56	174	26
YUV to HEVC	4k	4	0	10	0	0	47	177	27
YUV to AV1	4k	4	0	10	0	0	53	166	28
AVC to YUV	1080p	1	0	8	1	34	0	791	13
HEVC to YUV	1080p	1	0	8	1	32	0	826	10
VP9 to YUV	1080p	1	0	8	1	24	0	556	4
AVC to YUV	1080p	40	0	8	0	80	0	1615	0
HEVC to YUV	1080p	40	0	8	0	89	0	1719	0
VP9 to YUV	1080p	40	0	8	0	72	0	1641	0
YUV to AVC	1080p	1	0	8	1	0	51	665	18

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	49	672	15
YUV to AV1	1080p	1	0	8	1	0	46	549	14
YUV to AVC	1080p	32	0	8	0	0	99	1322	2
YUV to HEVC	1080p	32	0	8	0	0	99	1397	3
YUV to AV1	1080p	32	0	8	0	0	99	1208	2
AVC to AVC	1080p	1	1	8	0	43	83	982	15
AVC to HEVC	1080p	1	1	8	0	44	80	1035	15
AVC to AV1	1080p	1	1	8	0	41	81	922	16
HEVC to AVC	1080p	1	1	8	0	39	86	975	14
HEVC to HEVC	1080p	1	1	8	0	41	86	1071	14
HEVC to AV1	1080p	1	1	8	0	36	83	928	13
VP9 to AVC	1080p	1	1	8	0	21	42	546	4
VP9 to HEVC	1080p	1	1	8	0	22	39	540	4
VP9 to AV1	1080p	1	1	8	0	21	46	540	5
AVC to AVC	1080p	32	1	8	0	77	100	948	0
AVC to HEVC	1080p	32	1	8	0	85	99	1070	0
AVC to AV1	1080p	32	1	8	0	79	99	1036	1
HEVC to AVC	1080p	32	1	8	0	71	99	982	0
HEVC to HEVC	1080p	32	1	8	0	77	99	1107	1
HEVC to AV1	1080p	32	1	8	0	71	99	1055	1
VP9 to AVC	1080p	32	1	8	0	70	100	1097	0
VP9 to HEVC	1080p	32	1	8	0	74	100	1240	0
VP9 to AV1	1080p	32	1	8	0	71	100	1146	0
AVC to YUV	1080p	1	0	10	1	27	0	547	6
HEVC to YUV	1080p	1	0	10	1	26	0	525	5
VP9 to YUV	1080p	1	0	10	1	22	0	422	4
AVC to YUV	1080p	40	0	10	0	70	0	1074	0
HEVC to YUV	1080p	40	0	10	0	73	0	1063	0
VP9 to YUV	1080p	40	0	10	0	77	0	1016	0
YUV to AVC	1080p	1	0	10	1	0	30	389	19
YUV to HEVC	1080p	1	0	10	1	0	28	389	20
YUV to AV1	1080p	1	0	10	1	0	30	346	18
YUV to AVC	1080p	32	0	10	0	0	59	805	6
YUV to HEVC	1080p	32	0	10	0	0	56	814	5
YUV to AV1	1080p	32	0	10	0	0	61	747	10
AVC to YUV	720p	1	0	8	1	41	0	1100	11
HEVC to YUV	720p	1	0	8	1	38	0	1160	7
VP9 to YUV	720p	1	0	8	1	33	0	1023	5
AVC to YUV	720p	100	0	8	0	100	0	2759	0

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	95	0	2943	0
VP9 to YUV	720p	100	0	8	0	94	0	2662	0
YUV to AVC	720p	1	0	8	1	0	33	966	13
YUV to HEVC	720p	1	0	8	1	0	32	959	13
YUV to AV1	720p	1	0	8	1	0	32	754	10
YUV to AVC	720p	64	0	8	0	0	100	2617	1
YUV to HEVC	720p	64	0	8	0	0	98	2657	2
YUV to AV1	720p	64	0	8	0	0	100	2192	1
AVC to AVC	720p	1	1	8	0	50	49	1333	15
AVC to HEVC	720p	1	1	8	0	50	48	1339	15
AVC to AV1	720p	1	1	8	0	42	49	1071	14
HEVC to AVC	720p	1	1	8	0	45	51	1365	13
HEVC to HEVC	720p	1	1	8	0	45	51	1369	12
HEVC to AV1	720p	1	1	8	0	38	49	1076	11
VP9 to AVC	720p	1	1	8	0	32	35	1007	6
VP9 to HEVC	720p	1	1	8	0	31	34	993	6
VP9 to AV1	720p	1	1	8	0	31	45	994	7
AVC to AVC	720p	64	1	8	0	100	100	2121	0
AVC to HEVC	720p	64	1	8	0	100	100	2217	0
AVC to AV1	720p	64	1	8	0	93	100	1854	0
HEVC to AVC	720p	64	1	8	0	85	99	2115	0
HEVC to HEVC	720p	64	1	8	0	95	100	2201	0
HEVC to AV1	720p	64	1	8	0	87	100	1850	0
VP9 to AVC	720p	64	1	8	0	100	100	2387	0
VP9 to HEVC	720p	64	1	8	0	100	100	2467	0
VP9 to AV1	720p	64	1	8	0	99	100	2005	0

19. T1U – FFmpeg Latency

19.1 Encoding

19.1.1 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.

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 -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

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	58.45	60.73	57.74	0.14
YUV to HEVC	8k	1	56.08	61.64	54.36	1.08
YUV to AVC	4k	1	15.58	17.64	15.06	0.06
YUV to HEVC	4k	1	16.85	19.05	15.66	0.1
YUV to AV1	4k	1	21.96	26.79	16.5	0.66
YUV to AVC	4k	4	17.41	23.39	15.18	1.61
YUV to HEVC	4k	4	18.85	23.65	15.76	2.65
YUV to AV1	4k	4	23.43	28.28	16.69	1.2
YUV to AVC	4k	8	20.74	34.06	15.18	25.44
YUV to HEVC	4k	8	23.4	37.39	15.96	32.45
YUV to AV1	4k	8	37.22	46.38	25.75	7.86
YUV to AVC	1080p	1	4.84	5.67	4.67	0.02
YUV to HEVC	1080p	1	5.15	7.25	4.69	0.03
YUV to AV1	1080p	1	6.81	8.86	5.14	0.05
YUV to AVC	1080p	32	7.44	13.22	4.91	1.25
YUV to HEVC	1080p	32	8.53	14.72	5.35	2.48
YUV to AV1	1080p	32	41.01	46.34	33.84	1.45
YUV to AVC	720p	1	2.79	3.41	2.46	0.01
YUV to HEVC	720p	1	2.79	3.14	2.71	0.01
YUV to AV1	720p	1	3.75	4.3	3.14	0.01
YUV to AVC	720p	64	5.62	11.58	3.24	1.13
YUV to HEVC	720p	64	5.71	9.84	3.34	1.08
YUV to AV1	720p	64	39.77	48.05	33.35	2.98

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	5	0	142	5
HEVC to RGBA	8k	1	4	0	153	7
VP9 to RGBA	8k	1	22	0	40	2
AVC to RGBA	4k	1	21	0	145	7
AVC to RGBA	4k	16	92	3	585	2
HEVC to RGBA	4k	1	22	0	174	9
HEVC to RGBA	4k	16	92	4	657	2
VP9 to RGBA	4k	1	22	0	144	6
VP9 to RGBA	4k	16	93	3	591	1
AVC to RGBA	1080p	40	92	14	1887	1
HEVC to RGBA	1080p	40	95	14	1993	1
VP9 to RGBA	1080p	40	94	17	2440	0
AVC to RGBA	720p	100	96	18	2613	0
HEVC to RGBA	720p	100	91	20	2844	0
VP9 to RGBA	720p	64	94	18	2752	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	33	93	2	433	3
AVC to HEVC	8	32	93	2	464	4
AVC to AV1	8	23	95	2	389	3
HEVC to AVC	8	34	94	3	440	4
HEVC to HEVC	8	32	92	2	471	4
HEVC to AV1	8	23	93	2	396	3
VP9 to AVC	8	39	93	2	432	3
VP9 to HEVC	8	39	94	2	464	4
VP9 to AV1	8	29	95	2	393	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	46	147	64
RGBA to HEVC	4k	1	1	42	146	69
RGBA to AV1	4k	1	1	49	147	69
RGBA to AVC	4k	4	0	72	159	44
RGBA to HEVC	4k	4	0	58	163	40
RGBA to AV1	4k	4	0	63	165	43
RGBA to AVC	4k	8	0	65	167	35
RGBA to HEVC	4k	8	0	58	170	36
RGBA to AV1	4k	8	0	60	170	34
RGBA to AVC	1080p	1	1	30	395	46
RGBA to HEVC	1080p	1	1	28	393	50
RGBA to AV1	1080p	1	1	32	380	43
RGBA to AVC	1080p	16	0	61	622	22
RGBA to HEVC	1080p	16	0	53	624	24
RGBA to AV1	1080p	16	0	57	614	24
RGBA to AVC	1080p	32	0	56	641	7
RGBA to HEVC	1080p	32	0	53	651	11
RGBA to AV1	1080p	32	0	56	641	13
RGBA to AVC	720p	1	1	28	660	55
RGBA to HEVC	720p	1	1	28	664	52
RGBA to AV1	720p	1	1	32	632	52
RGBA to AVC	720p	16	0	47	1202	35
RGBA to HEVC	720p	16	0	45	1174	34
RGBA to AV1	720p	16	0	54	1138	33
RGBA to AVC	720p	32	0	44	1155	32
RGBA to HEVC	720p	32	0	44	1158	31
RGBA to AV1	720p	32	0	54	1161	32

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	95	299	19
YUV to HEVC	4k	4	0	0	1	96	332	14
YUV to AV1	4k	4	0	0	1	96	288	14
YUV to AVC	4k	4	0	0	2	94	300	22
YUV to HEVC	4k	4	0	0	2	98	172	7
YUV to AV1	4k	4	0	0	2	96	136	8
YUV to AVC	4k	4	0	0	3	96	300	17
YUV to HEVC	4k	4	0	0	3	99	100	5
YUV to AV1	4k	4	0	0	3	100	76	4
YUV to AVC	4k	4	0	1	1	98	188	9
YUV to HEVC	4k	4	0	1	1	97	236	12
YUV to AVC	4k	4	0	1	2	98	188	9
YUV to HEVC	4k	4	0	1	2	98	108	5
YUV to AVC	4k	4	0	1	3	98	188	9
YUV to HEVC	4k	4	0	1	3	97	68	5
YUV to AVC	4k	4	4	0	1	99	196	8
YUV to HEVC	4k	4	4	0	1	100	228	11
YUV to AV1	4k	4	4	0	1	99	200	8
YUV to AVC	4k	4	4	0	2	99	196	9
YUV to HEVC	4k	4	4	0	2	99	136	6
YUV to AV1	4k	4	4	0	2	100	116	5
YUV to AVC	4k	4	4	0	3	99	196	11
YUV to HEVC	4k	4	4	0	3	101	88	5
YUV to AV1	4k	4	4	0	3	102	72	3
YUV to AVC	4k	4	4	1	1	100	140	7
YUV to HEVC	4k	4	4	1	1	100	180	10
YUV to AVC	4k	4	4	1	2	100	140	6
YUV to HEVC	4k	4	4	1	2	99	96	4
YUV to AVC	4k	4	4	1	3	100	140	6
YUV to HEVC	4k	4	4	1	3	99	64	3
YUV to AVC	4k	4	16	0	1	100	196	16
YUV to HEVC	4k	4	16	0	1	100	228	11
YUV to AV1	4k	4	16	0	1	100	199	12
YUV to AVC	4k	4	16	0	2	99	196	10
YUV to HEVC	4k	4	16	0	2	99	136	6
YUV to AV1	4k	4	16	0	2	100	116	6
YUV to AVC	4k	4	16	0	3	99	196	10
YUV to HEVC	4k	4	16	0	3	99	88	5
YUV to AV1	4k	4	16	0	3	99	72	5

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	16	1	1	99	140	8
YUV to HEVC	4k	4	16	1	1	99	180	8
YUV to AVC	4k	4	16	1	2	100	140	6
YUV to HEVC	4k	4	16	1	2	100	94	5
YUV to AVC	4k	4	16	1	3	99	140	7
YUV to HEVC	4k	4	16	1	3	101	64	3
YUV to AVC	4k	4	40	0	1	100	196	9
YUV to HEVC	4k	4	40	0	1	99	224	13
YUV to AV1	4k	4	40	0	1	100	196	7
YUV to AVC	4k	4	40	0	2	99	196	9
YUV to HEVC	4k	4	40	0	2	100	136	7
YUV to AV1	4k	4	40	0	2	100	112	6
YUV to AVC	4k	4	40	0	3	99	196	10
YUV to HEVC	4k	4	40	0	3	100	88	5
YUV to AV1	4k	4	40	0	3	100	72	3
YUV to AVC	4k	4	40	1	1	99	136	6
YUV to HEVC	4k	4	40	1	1	99	176	8
YUV to AVC	4k	4	40	1	2	100	136	9
YUV to HEVC	4k	4	40	1	2	100	92	5
YUV to AVC	4k	4	40	1	3	100	136	8
YUV to HEVC	4k	4	40	1	3	100	64	3
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	99	682	2
YUV to AV1	1080p	20	0	0	2	100	560	2
YUV to AVC	1080p	20	0	0	3	99	1280	4
YUV to HEVC	1080p	20	0	0	3	100	400	1
YUV to AV1	1080p	20	0	0	3	100	300	1
YUV to AVC	1080p	20	0	1	1	100	760	2
YUV to HEVC	1080p	20	0	1	1	99	961	3
YUV to AVC	1080p	20	0	1	2	100	760	2
YUV to HEVC	1080p	20	0	1	2	100	440	1
YUV to AVC	1080p	20	0	1	3	99	760	2
YUV to HEVC	1080p	20	0	1	3	99	280	1
YUV to AVC	1080p	20	4	0	1	99	700	2
YUV to HEVC	1080p	20	4	0	1	99	820	3
YUV to AV1	1080p	20	4	0	1	99	720	2
YUV to AVC	1080p	20	4	0	2	99	700	2

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	4	0	2	99	514	1
YUV to AV1	1080p	20	4	0	2	100	420	1
YUV to AVC	1080p	20	4	0	3	99	700	2
YUV to HEVC	1080p	20	4	0	3	99	340	1
YUV to AV1	1080p	20	4	0	3	100	272	1
YUV to AVC	1080p	20	4	1	1	99	500	1
YUV to HEVC	1080p	20	4	1	1	99	660	2
YUV to AVC	1080p	20	4	1	2	99	500	1
YUV to HEVC	1080p	20	4	1	2	99	360	1
YUV to AVC	1080p	20	4	1	3	99	501	2
YUV to HEVC	1080p	20	4	1	3	100	240	1
YUV to AVC	1080p	20	16	0	1	100	700	2
YUV to HEVC	1080p	20	16	0	1	99	820	2
YUV to AV1	1080p	20	16	0	1	99	719	2
YUV to AVC	1080p	20	16	0	2	99	700	2
YUV to HEVC	1080p	20	16	0	2	99	502	1
YUV to AV1	1080p	20	16	0	2	100	420	1
YUV to AVC	1080p	20	16	0	3	100	700	2
YUV to HEVC	1080p	20	16	0	3	99	340	1
YUV to AV1	1080p	20	16	0	3	99	266	1
YUV to AVC	1080p	20	16	1	1	99	500	2
YUV to HEVC	1080p	20	16	1	1	99	647	2
YUV to AVC	1080p	20	16	1	2	99	500	2
YUV to HEVC	1080p	20	16	1	2	100	360	1
YUV to AVC	1080p	20	16	1	3	100	500	2
YUV to HEVC	1080p	20	16	1	3	100	240	1
YUV to AVC	1080p	20	40	0	1	99	683	2
YUV to HEVC	1080p	20	40	0	1	99	801	2
YUV to AV1	1080p	20	40	0	1	99	703	2
YUV to AVC	1080p	20	40	0	2	100	690	2
YUV to HEVC	1080p	20	40	0	2	100	500	1
YUV to AV1	1080p	20	40	0	2	100	420	1
YUV to AVC	1080p	20	40	0	3	99	689	2
YUV to HEVC	1080p	20	40	0	3	100	340	1
YUV to AV1	1080p	20	40	0	3	99	261	1
YUV to AVC	1080p	20	40	1	1	99	500	1
YUV to HEVC	1080p	20	40	1	1	99	640	2
YUV to AVC	1080p	20	40	1	2	99	500	1
YUV to HEVC	1080p	20	40	1	2	100	360	1
YUV to AVC	1080p	20	40	1	3	99	500	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	40	1	3	100	240	1
YUV to AVC	720p	40	0	0	1	94	2322	2
YUV to HEVC	720p	40	0	0	1	92	2338	2
YUV to AV1	720p	40	0	0	1	96	2002	2
YUV to AVC	720p	40	0	0	2	93	2324	3
YUV to HEVC	720p	40	0	0	2	99	1520	1
YUV to AV1	720p	40	0	0	2	99	1203	1
YUV to AVC	720p	40	0	0	3	94	2323	2
YUV to HEVC	720p	40	0	0	3	100	920	1
YUV to AV1	720p	40	0	0	3	99	655	0
YUV to AVC	720p	40	0	1	1	99	1684	1
YUV to HEVC	720p	40	0	1	1	99	2080	2
YUV to AVC	720p	40	0	1	2	99	1694	1
YUV to HEVC	720p	40	0	1	2	99	960	1
YUV to AVC	720p	40	0	1	3	99	1689	1
YUV to HEVC	720p	40	0	1	3	99	607	1
YUV to AVC	720p	40	4	0	1	100	1438	1
YUV to HEVC	720p	40	4	0	1	100	1360	1
YUV to AV1	720p	40	4	0	1	100	1006	1
YUV to AVC	720p	40	4	0	2	100	1440	1
YUV to HEVC	720p	40	4	0	2	99	1080	1
YUV to AV1	720p	40	4	0	2	99	880	1
YUV to AVC	720p	40	4	0	3	100	1420	1
YUV to HEVC	720p	40	4	0	3	99	720	0
YUV to AV1	720p	40	4	0	3	99	560	0
YUV to AVC	720p	40	4	1	1	99	1075	1
YUV to HEVC	720p	40	4	1	1	100	1320	1
YUV to AVC	720p	40	4	1	2	99	1077	0
YUV to HEVC	720p	40	4	1	2	99	760	1
YUV to AVC	720p	40	4	1	3	99	1075	1
YUV to HEVC	720p	40	4	1	3	100	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	1059	1
YUV to AVC	720p	40	16	0	2	100	1400	1
YUV to HEVC	720p	40	16	0	2	99	1079	1
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	0
YUV to AV1	720p	40	16	0	3	99	560	0

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	40	16	1	1	99	1049	1
YUV to HEVC	720p	40	16	1	1	99	1322	1
YUV to AVC	720p	40	16	1	2	99	1047	1
YUV to HEVC	720p	40	16	1	2	99	760	0
YUV to AVC	720p	40	16	1	3	99	1048	1
YUV to HEVC	720p	40	16	1	3	99	520	0
YUV to AVC	720p	40	40	0	1	100	1360	1
YUV to HEVC	720p	40	40	0	1	100	1320	1
YUV to AV1	720p	40	40	0	1	100	1040	1
YUV to AVC	720p	40	40	0	2	100	1360	1
YUV to HEVC	720p	40	40	0	2	99	1040	1
YUV to AV1	720p	40	40	0	2	99	840	0
YUV to AVC	720p	40	40	0	3	100	1360	1
YUV to HEVC	720p	40	40	0	3	99	720	0
YUV to AV1	720p	40	40	0	3	99	559	0
YUV to AVC	720p	40	40	1	1	99	1040	1
YUV to HEVC	720p	40	40	1	1	100	1319	1
YUV to AVC	720p	40	40	1	2	99	1040	1
YUV to HEVC	720p	40	40	1	2	99	753	1
YUV to AVC	720p	40	40	1	3	99	1040	1
YUV to HEVC	720p	40	40	1	3	99	520	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	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	702	2
YUV to HEVC	1080p	20	0	0	2	19	100	520	2
YUV to AV1	1080p	20	0	0	2	19	100	440	2
YUV to AVC	1080p	20	0	0	3	19	99	702	2
YUV to HEVC	1080p	20	0	0	3	19	100	340	1
YUV to AV1	1080p	20	0	0	3	19	100	280	1
YUV to AVC	1080p	20	0	1	1	19	100	507	1
YUV to HEVC	1080p	20	0	1	1	19	100	660	2
YUV to AVC	1080p	20	0	1	2	19	100	504	1
YUV to HEVC	1080p	20	0	1	2	19	100	360	1
YUV to AVC	1080p	20	0	1	3	19	100	502	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	100	820	2
YUV to AV1	1080p	20	4	0	1	19	99	720	3
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

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	3	19	100	701	3
YUV to HEVC	1080p	20	4	0	3	19	100	340	1
YUV to AV1	1080p	20	4	0	3	19	99	267	1
YUV to AVC	1080p	20	4	1	1	19	99	500	1
YUV to HEVC	1080p	20	4	1	1	19	99	660	2
YUV to AVC	1080p	20	4	1	2	19	99	500	1
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	242	1
YUV to AVC	1080p	20	16	0	1	19	99	700	2
YUV to HEVC	1080p	20	16	0	1	19	99	819	3
YUV to AV1	1080p	20	16	0	1	19	99	720	2
YUV to AVC	1080p	20	16	0	2	19	99	700	2
YUV to HEVC	1080p	20	16	0	2	19	99	505	2
YUV to AV1	1080p	20	16	0	2	19	99	420	2
YUV to AVC	1080p	20	16	0	3	19	99	700	2
YUV to HEVC	1080p	20	16	0	3	19	99	340	1
YUV to AV1	1080p	20	16	0	3	19	99	261	1
YUV to AVC	1080p	20	16	1	1	19	100	500	1
YUV to HEVC	1080p	20	16	1	1	19	100	650	2
YUV to AVC	1080p	20	16	1	2	19	100	500	2
YUV to HEVC	1080p	20	16	1	2	19	99	360	1
YUV to AVC	1080p	20	16	1	3	19	100	500	1
YUV to HEVC	1080p	20	16	1	3	19	99	240	1
YUV to AVC	1080p	20	40	0	1	19	99	691	3
YUV to HEVC	1080p	20	40	0	1	19	99	801	3
YUV to AV1	1080p	20	40	0	1	19	99	701	2
YUV to AVC	1080p	20	40	0	2	19	99	691	2
YUV to HEVC	1080p	20	40	0	2	19	100	500	1
YUV to AV1	1080p	20	40	0	2	19	100	420	1
YUV to AVC	1080p	20	40	0	3	19	99	688	3
YUV to HEVC	1080p	20	40	0	3	19	99	339	1
YUV to AV1	1080p	20	40	0	3	19	100	260	1
YUV to AVC	1080p	20	40	1	1	19	99	500	1
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	99	500	2
YUV to HEVC	1080p	20	40	1	3	19	100	240	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	23	99	700	2
YUV to HEVC	1080p	20	0	0	1	23	99	820	3
YUV to AV1	1080p	20	0	0	1	23	99	720	2
YUV to AVC	1080p	20	0	0	2	23	99	702	3
YUV to HEVC	1080p	20	0	0	2	23	100	520	2
YUV to AV1	1080p	20	0	0	2	23	99	439	1
YUV to AVC	1080p	20	0	0	3	23	99	703	2
YUV to HEVC	1080p	20	0	0	3	23	100	340	1
YUV to AV1	1080p	20	0	0	3	23	99	280	1
YUV to AVC	1080p	20	0	1	1	23	100	505	1
YUV to HEVC	1080p	20	0	1	1	23	99	660	2
YUV to AVC	1080p	20	0	1	2	23	100	504	1
YUV to HEVC	1080p	20	0	1	2	23	100	360	1
YUV to AVC	1080p	20	0	1	3	23	100	509	2
YUV to HEVC	1080p	20	0	1	3	23	100	240	1
YUV to AVC	1080p	20	4	0	1	23	99	701	3
YUV to HEVC	1080p	20	4	0	1	23	99	820	3
YUV to AV1	1080p	20	4	0	1	23	99	720	2
YUV to AVC	1080p	20	4	0	2	23	99	700	2
YUV to HEVC	1080p	20	4	0	2	23	99	520	1
YUV to AV1	1080p	20	4	0	2	23	99	421	1
YUV to AVC	1080p	20	4	0	3	23	100	700	2
YUV to HEVC	1080p	20	4	0	3	23	100	340	1
YUV to AV1	1080p	20	4	0	3	23	99	267	1
YUV to AVC	1080p	20	4	1	1	23	99	502	1
YUV to HEVC	1080p	20	4	1	1	23	99	660	2
YUV to AVC	1080p	20	4	1	2	23	100	505	1
YUV to HEVC	1080p	20	4	1	2	23	100	360	1
YUV to AVC	1080p	20	4	1	3	23	99	500	1
YUV to HEVC	1080p	20	4	1	3	23	99	240	1
YUV to AVC	1080p	20	16	0	1	23	100	700	2
YUV to HEVC	1080p	20	16	0	1	23	99	819	3
YUV to AV1	1080p	20	16	0	1	23	99	720	2
YUV to AVC	1080p	20	16	0	2	23	99	700	2
YUV to HEVC	1080p	20	16	0	2	23	99	501	1
YUV to AV1	1080p	20	16	0	2	23	99	420	1
YUV to AVC	1080p	20	16	0	3	23	99	700	2
YUV to HEVC	1080p	20	16	0	3	23	99	340	1
YUV to AV1	1080p	20	16	0	3	23	100	263	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	16	1	1	23	99	502	2
YUV to HEVC	1080p	20	16	1	1	23	99	645	2
YUV to AVC	1080p	20	16	1	2	23	100	500	2
YUV to HEVC	1080p	20	16	1	2	23	100	360	1
YUV to AVC	1080p	20	16	1	3	23	99	500	1
YUV to HEVC	1080p	20	16	1	3	23	99	241	1
YUV to AVC	1080p	20	40	0	1	23	99	689	2
YUV to HEVC	1080p	20	40	0	1	23	99	802	3
YUV to AV1	1080p	20	40	0	1	23	99	701	2
YUV to AVC	1080p	20	40	0	2	23	99	687	2
YUV to HEVC	1080p	20	40	0	2	23	100	500	1
YUV to AV1	1080p	20	40	0	2	23	100	420	1
YUV to AVC	1080p	20	40	0	3	23	99	694	2
YUV to HEVC	1080p	20	40	0	3	23	99	335	1
YUV to AV1	1080p	20	40	0	3	23	99	260	1
YUV to AVC	1080p	20	40	1	1	23	100	500	2
YUV to HEVC	1080p	20	40	1	1	23	99	640	2
YUV to AVC	1080p	20	40	1	2	23	99	500	2
YUV to HEVC	1080p	20	40	1	2	23	99	360	1
YUV to AVC	1080p	20	40	1	3	23	99	500	2
YUV to HEVC	1080p	20	40	1	3	23	99	240	1
YUV to AVC	1080p	20	0	0	1	27	99	700	2
YUV to HEVC	1080p	20	0	0	1	27	99	820	2
YUV to AV1	1080p	20	0	0	1	27	100	720	2
YUV to AVC	1080p	20	0	0	2	27	99	700	2
YUV to HEVC	1080p	20	0	0	2	27	100	520	2
YUV to AV1	1080p	20	0	0	2	27	99	440	1
YUV to AVC	1080p	20	0	0	3	27	99	703	3
YUV to HEVC	1080p	20	0	0	3	27	99	340	1
YUV to AV1	1080p	20	0	0	3	27	100	280	1
YUV to AVC	1080p	20	0	1	1	27	100	503	1
YUV to HEVC	1080p	20	0	1	1	27	100	660	2
YUV to AVC	1080p	20	0	1	2	27	99	500	1
YUV to HEVC	1080p	20	0	1	2	27	99	360	1
YUV to AVC	1080p	20	0	1	3	27	99	502	1
YUV to HEVC	1080p	20	0	1	3	27	100	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

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	2	27	99	700	2
YUV to HEVC	1080p	20	4	0	2	27	99	520	1
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	100	340	1
YUV to AV1	1080p	20	4	0	3	27	99	268	1
YUV to AVC	1080p	20	4	1	1	27	99	501	1
YUV to HEVC	1080p	20	4	1	1	27	99	660	2
YUV to AVC	1080p	20	4	1	2	27	99	501	1
YUV to HEVC	1080p	20	4	1	2	27	100	360	1
YUV to AVC	1080p	20	4	1	3	27	99	500	2
YUV to HEVC	1080p	20	4	1	3	27	100	241	1
YUV to AVC	1080p	20	16	0	1	27	99	700	2
YUV to HEVC	1080p	20	16	0	1	27	99	820	3
YUV to AV1	1080p	20	16	0	1	27	99	719	3
YUV to AVC	1080p	20	16	0	2	27	99	700	2
YUV to HEVC	1080p	20	16	0	2	27	100	504	2
YUV to AV1	1080p	20	16	0	2	27	99	420	1
YUV to AVC	1080p	20	16	0	3	27	99	700	3
YUV to HEVC	1080p	20	16	0	3	27	99	340	1
YUV to AV1	1080p	20	16	0	3	27	100	261	1
YUV to AVC	1080p	20	16	1	1	27	99	500	1
YUV to HEVC	1080p	20	16	1	1	27	100	642	2
YUV to AVC	1080p	20	16	1	2	27	99	500	1
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	99	240	1
YUV to AVC	1080p	20	40	0	1	27	99	692	2
YUV to HEVC	1080p	20	40	0	1	27	99	802	3
YUV to AV1	1080p	20	40	0	1	27	99	702	3
YUV to AVC	1080p	20	40	0	2	27	99	691	2
YUV to HEVC	1080p	20	40	0	2	27	100	500	2
YUV to AV1	1080p	20	40	0	2	27	99	420	1
YUV to AVC	1080p	20	40	0	3	27	99	691	2
YUV to HEVC	1080p	20	40	0	3	27	100	340	1
YUV to AV1	1080p	20	40	0	3	27	100	260	1
YUV to AVC	1080p	20	40	1	1	27	100	500	2
YUV to HEVC	1080p	20	40	1	1	27	99	640	2
YUV to AVC	1080p	20	40	1	2	27	99	500	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	40	1	2	27	99	360	1
YUV to AVC	1080p	20	40	1	3	27	100	500	1
YUV to HEVC	1080p	20	40	1	3	27	100	240	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	20	5	289	10
AVC to HEVC	1	14	21	5	307	11
AVC to AV1	1	13	20	4	263	10
HEVC to AVC	1	12	21	4	289	13
HEVC to HEVC	1	13	20	5	306	14
HEVC to AV1	1	12	21	4	263	13
VP9 to AVC	1	17	20	4	288	9
VP9 to HEVC	1	18	21	5	305	10
VP9 to AV1	1	16	20	4	262	10
AVC to AVC	16	72	94	24	1026	2
AVC to HEVC	16	78	93	27	1136	2
AVC to AV1	16	73	94	26	1088	3
HEVC to AVC	16	71	93	24	1056	3
HEVC to HEVC	16	79	95	28	1183	4
HEVC to AV1	16	70	94	26	1120	3
VP9 to AVC	16	80	92	23	1024	2
VP9 to HEVC	16	90	94	26	1151	3
VP9 to AV1	16	83	92	25	1106	3
AVC to AVC	32	74	93	23	928	1
AVC to HEVC	32	80	92	26	1038	1
AVC to AV1	32	78	93	25	1024	1
HEVC to AVC	32	74	95	24	960	1
HEVC to HEVC	32	80	93	27	1088	1
HEVC to AV1	32	75	94	26	1056	1
VP9 to AVC	32	81	93	23	960	1
VP9 to HEVC	32	87	92	25	1088	1
VP9 to AV1	32	84	94	25	1056	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	160	35
HEVC to YUV	4k	1	0	8	1	9	0	159	48
VP9 to YUV	4k	1	0	8	1	24	0	83	25
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	12
YUV to AVC	4k	1	0	8	1	0	92	184	61
YUV to HEVC	4k	1	0	8	1	0	86	191	64
YUV to AV1	4k	1	0	8	1	0	93	179	58
YUV to AVC	4k	3	0	8	0	0	73	153	39
YUV to HEVC	4k	3	0	8	0	0	73	162	41
YUV to AV1	4k	3	0	8	0	0	73	138	38
YUV to AVC	4k	5	0	8	0	0	97	191	27
YUV to HEVC	4k	5	0	8	0	0	97	215	32
YUV to AV1	4k	5	0	8	0	0	98	185	29
AVC to AVC	4k	1	1	8	1	13	95	127	32
AVC to HEVC	4k	1	1	8	1	15	95	157	46
AVC to AV1	4k	1	1	8	1	13	89	154	42
HEVC to AVC	4k	1	1	8	1	14	93	125	37
HEVC to HEVC	4k	1	1	8	1	15	97	156	48
HEVC to AV1	4k	1	1	8	1	14	97	149	52
VP9 to AVC	4k	1	1	8	1	24	40	83	21
VP9 to HEVC	4k	1	1	8	1	24	38	83	20
VP9 to AV1	4k	1	1	8	1	24	44	83	24
AVC to AVC	4k	3	1	8	0	48	71	123	32
AVC to HEVC	4k	3	1	8	0	49	72	147	32
AVC to AV1	4k	3	1	8	0	40	72	135	30
HEVC to AVC	4k	3	1	8	0	41	72	123	33
HEVC to HEVC	4k	3	1	8	0	41	72	150	33
HEVC to AV1	4k	3	1	8	0	34	72	135	28
VP9 to AVC	4k	3	1	8	0	46	72	120	29
VP9 to HEVC	4k	3	1	8	0	49	72	147	30
VP9 to AV1	4k	3	1	8	0	42	72	135	25
AVC to AVC	4k	5	1	8	0	66	97	119	16
AVC to HEVC	4k	5	1	8	0	70	96	158	23
AVC to AV1	4k	5	1	8	0	64	96	156	18
HEVC to AVC	4k	5	1	8	0	56	97	121	13
HEVC to HEVC	4k	5	1	8	0	60	97	160	18
HEVC to AV1	4k	5	1	8	0	52	97	163	23

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AVC	4k	5	1	8	0	58	96	121	20
VP9 to HEVC	4k	5	1	8	0	66	98	159	25
VP9 to AV1	4k	5	1	8	0	62	97	159	27
AVC to YUV	4k	1	0	10	1	8	0	103	32
HEVC to YUV	4k	1	0	10	1	8	0	102	31
VP9 to YUV	4k	1	0	10	1	24	0	83	25
AVC to YUV	4k	16	0	10	0	50	0	135	10
HEVC to YUV	4k	16	0	10	0	68	0	135	9
VP9 to YUV	4k	16	0	10	0	67	0	135	8
YUV to AVC	4k	1	0	10	1	0	51	107	56
YUV to HEVC	4k	1	0	10	1	0	48	109	60
YUV to AV1	4k	1	0	10	1	0	55	106	52
YUV to AVC	4k	3	0	10	0	0	62	117	58
YUV to HEVC	4k	3	0	10	0	0	53	119	58
YUV to AV1	4k	3	0	10	0	0	61	117	56
AVC to YUV	1080p	1	0	8	1	18	0	429	58
HEVC to YUV	1080p	1	0	8	1	15	0	405	68
VP9 to YUV	1080p	1	0	8	1	23	0	211	32
AVC to YUV	1080p	18	0	8	0	93	0	959	8
HEVC to YUV	1080p	18	0	8	0	91	0	999	11
VP9 to YUV	1080p	18	0	8	0	98	0	877	7
YUV to AVC	1080p	1	0	8	1	0	55	445	71
YUV to HEVC	1080p	1	0	8	1	0	52	449	64
YUV to AV1	1080p	1	0	8	1	0	52	408	81
YUV to AVC	1080p	20	0	8	0	0	99	819	13
YUV to HEVC	1080p	20	0	8	0	0	99	859	15
YUV to AV1	1080p	20	0	8	0	0	99	740	12
AVC to AVC	1080p	1	1	8	1	35	70	542	73
AVC to HEVC	1080p	1	1	8	1	35	69	583	74
AVC to AV1	1080p	1	1	8	1	30	74	544	67
HEVC to AVC	1080p	1	1	8	1	31	72	520	82
HEVC to HEVC	1080p	1	1	8	1	32	73	575	83
HEVC to AV1	1080p	1	1	8	1	25	73	526	82
VP9 to AVC	1080p	1	1	8	1	23	25	209	30
VP9 to HEVC	1080p	1	1	8	1	23	24	209	32
VP9 to AV1	1080p	1	1	8	1	22	28	210	34
AVC to AVC	1080p	20	1	8	0	72	99	570	8
AVC to HEVC	1080p	20	1	8	0	80	99	656	10
AVC to AV1	1080p	20	1	8	0	74	99	642	12
HEVC to AVC	1080p	20	1	8	0	69	99	600	8

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to HEVC	1080p	20	1	8	0	71	100	684	8
HEVC to AV1	1080p	20	1	8	0	66	99	660	7
VP9 to AVC	1080p	20	1	8	0	78	99	580	9
VP9 to HEVC	1080p	20	1	8	0	87	100	660	11
VP9 to AV1	1080p	20	1	8	0	83	99	652	8
AVC to YUV	1080p	1	0	10	1	15	0	319	40
HEVC to YUV	1080p	1	0	10	1	14	0	312	44
VP9 to YUV	1080p	1	0	10	1	21	0	299	43
AVC to YUV	1080p	18	0	10	0	52	0	541	4
HEVC to YUV	1080p	18	0	10	0	53	0	542	4
VP9 to YUV	1080p	18	0	10	0	56	0	540	4
YUV to AVC	1080p	1	0	10	1	0	37	305	52
YUV to HEVC	1080p	1	0	10	1	0	34	304	72
YUV to AV1	1080p	1	0	10	1	0	38	284	68
YUV to AVC	1080p	20	0	10	0	0	57	460	13
YUV to HEVC	1080p	20	0	10	0	0	53	462	14
YUV to AV1	1080p	20	0	10	0	0	59	460	14
AVC to YUV	720p	1	0	8	1	24	0	641	63
HEVC to YUV	720p	1	0	8	1	21	0	629	78
VP9 to YUV	720p	1	0	8	1	21	0	371	40
AVC to YUV	720p	40	0	8	0	98	0	1895	6
HEVC to YUV	720p	40	0	8	0	89	0	1917	5
VP9 to YUV	720p	40	0	8	0	93	0	1614	4
YUV to AVC	720p	1	0	8	1	0	29	553	60
YUV to HEVC	720p	1	0	8	1	0	29	551	68
YUV to AV1	720p	1	0	8	1	0	32	500	66
YUV to AVC	720p	40	0	8	0	0	90	1622	7
YUV to HEVC	720p	40	0	8	0	0	90	1685	8
YUV to AV1	720p	40	0	8	0	0	90	1476	7
AVC to AVC	720p	1	1	8	1	37	46	856	76
AVC to HEVC	720p	1	1	8	1	40	49	886	68
AVC to AV1	720p	1	1	8	1	31	48	766	70
HEVC to AVC	720p	1	1	8	1	33	46	819	79
HEVC to HEVC	720p	1	1	8	1	33	47	837	62
HEVC to AV1	720p	1	1	8	1	27	48	747	82
VP9 to AVC	720p	1	1	8	1	21	20	368	43
VP9 to HEVC	720p	1	1	8	1	21	19	366	47
VP9 to AV1	720p	1	1	8	1	21	23	366	49
AVC to AVC	720p	40	1	8	0	91	100	1290	4
AVC to HEVC	720p	40	1	8	0	93	100	1415	5

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	720p	40	1	8	0	75	100	1291	3
HEVC to AVC	720p	40	1	8	0	83	100	1298	4
HEVC to HEVC	720p	40	1	8	0	85	100	1427	6
HEVC to AV1	720p	40	1	8	0	68	100	1290	6
VP9 to AVC	720p	40	1	8	0	100	95	1251	4
VP9 to HEVC	720p	40	1	8	0	100	93	1365	4
VP9 to AV1	720p	40	1	8	0	90	100	1286	2

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	34
HEVC to YUV	4k	1	0	8	1	10	0	160	26
VP9 to YUV	4k	1	0	8	1	24	0	82	17
AVC to YUV	4k	5	0	8	0	84	0	249	12
HEVC to YUV	4k	5	0	8	0	87	0	263	9
VP9 to YUV	4k	5	0	8	0	92	0	262	8
YUV to AVC	4k	1	0	8	1	0	60	121	49
YUV to HEVC	4k	1	0	8	1	0	53	115	52
YUV to AV1	4k	1	0	8	1	0	54	107	49
YUV to AVC	4k	3	0	8	0	0	72	152	30
YUV to HEVC	4k	3	0	8	0	0	73	161	29
YUV to AV1	4k	3	0	8	0	0	73	139	29
YUV to AVC	4k	5	0	8	0	0	98	186	26
YUV to HEVC	4k	5	0	8	0	0	97	216	26
YUV to AV1	4k	5	0	8	0	0	97	187	26
AVC to AVC	4k	1	1	8	0	15	96	128	50
AVC to HEVC	4k	1	1	8	0	15	96	163	54
AVC to AV1	4k	1	1	8	0	16	96	159	51
HEVC to AVC	4k	1	1	8	0	17	97	128	50
HEVC to HEVC	4k	1	1	8	0	16	97	159	49
HEVC to AV1	4k	1	1	8	0	15	97	152	46
VP9 to AVC	4k	1	1	8	0	24	40	82	41
VP9 to HEVC	4k	1	1	8	0	24	37	82	37
VP9 to AV1	4k	1	1	8	0	24	43	82	42
AVC to AVC	4k	3	1	8	0	45	72	121	52
AVC to HEVC	4k	3	1	8	0	48	72	152	27
AVC to AV1	4k	3	1	8	0	40	72	137	34
HEVC to AVC	4k	3	1	8	0	40	72	120	31
HEVC to HEVC	4k	3	1	8	0	40	72	145	26
HEVC to AV1	4k	3	1	8	0	34	73	133	26
VP9 to AVC	4k	3	1	8	0	46	73	122	24
VP9 to HEVC	4k	3	1	8	0	48	72	148	26
VP9 to AV1	4k	3	1	8	0	40	73	135	25
AVC to AVC	4k	5	1	8	0	66	96	122	12
AVC to HEVC	4k	5	1	8	0	69	97	159	11
AVC to AV1	4k	5	1	8	0	64	96	158	12
HEVC to AVC	4k	5	1	8	0	58	98	123	12
HEVC to HEVC	4k	5	1	8	0	61	97	159	17

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to AV1	4k	5	1	8	0	54	97	157	11
VP9 to AVC	4k	5	1	8	0	62	98	124	11
VP9 to HEVC	4k	5	1	8	0	68	97	161	11
VP9 to AV1	4k	5	1	8	0	63	97	159	10
AVC to YUV	4k	1	0	10	1	8	0	96	23
HEVC to YUV	4k	1	0	10	1	7	0	95	23
VP9 to YUV	4k	1	0	10	1	24	0	82	20
AVC to YUV	4k	5	0	10	0	55	0	131	7
HEVC to YUV	4k	5	0	10	0	64	0	135	7
VP9 to YUV	4k	5	0	10	0	72	0	134	7
YUV to AVC	4k	1	0	10	1	0	32	66	48
YUV to HEVC	4k	1	0	10	1	0	29	66	49
YUV to AV1	4k	1	0	10	1	0	34	64	49
YUV to AVC	4k	3	0	10	0	0	40	89	39
YUV to HEVC	4k	3	0	10	0	0	43	88	39
YUV to AV1	4k	3	0	10	0	0	42	83	38
AVC to YUV	1080p	1	0	8	1	15	0	371	41
HEVC to YUV	1080p	1	0	8	1	16	0	422	33
VP9 to YUV	1080p	1	0	8	1	23	0	211	17
AVC to YUV	1080p	20	0	8	0	90	0	929	8
HEVC to YUV	1080p	20	0	8	0	89	0	965	5
VP9 to YUV	1080p	20	0	8	0	99	0	889	3
YUV to AVC	1080p	1	0	8	1	0	37	316	45
YUV to HEVC	1080p	1	0	8	1	0	35	328	46
YUV to AV1	1080p	1	0	8	1	0	34	277	36
YUV to AVC	1080p	20	0	8	0	0	99	827	10
YUV to HEVC	1080p	20	0	8	0	0	99	875	11
YUV to AV1	1080p	20	0	8	0	0	99	763	10
AVC to AVC	1080p	1	1	8	0	40	81	590	66
AVC to HEVC	1080p	1	1	8	0	39	78	641	69
AVC to AV1	1080p	1	1	8	0	33	82	597	62
HEVC to AVC	1080p	1	1	8	0	35	83	588	65
HEVC to HEVC	1080p	1	1	8	0	29	74	654	57
HEVC to AV1	1080p	1	1	8	0	30	85	596	68
VP9 to AVC	1080p	1	1	8	0	23	25	210	47
VP9 to HEVC	1080p	1	1	8	0	23	24	211	38
VP9 to AV1	1080p	1	1	8	0	23	28	211	34
AVC to AVC	1080p	20	1	8	0	72	99	585	6
AVC to HEVC	1080p	20	1	8	0	79	99	667	6

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	1080p	20	1	8	0	72	99	652	5
HEVC to AVC	1080p	20	1	8	0	67	99	611	5
HEVC to HEVC	1080p	20	1	8	0	71	99	697	7
HEVC to AV1	1080p	20	1	8	0	65	99	668	4
VP9 to AVC	1080p	20	1	8	0	77	100	591	4
VP9 to HEVC	1080p	20	1	8	0	86	99	678	6
VP9 to AV1	1080p	20	1	8	0	82	99	663	7
AVC to YUV	1080p	1	0	10	1	16	0	318	28
HEVC to YUV	1080p	1	0	10	1	14	0	311	26
VP9 to YUV	1080p	1	0	10	1	21	0	297	27
AVC to YUV	1080p	20	0	10	0	52	0	542	2
HEVC to YUV	1080p	20	0	10	0	66	0	539	2
VP9 to YUV	1080p	20	0	10	0	100	0	520	2
YUV to AVC	1080p	1	0	10	1	0	24	215	45
YUV to HEVC	1080p	1	0	10	1	0	24	216	45
YUV to AV1	1080p	1	0	10	1	0	24	190	46
YUV to AVC	1080p	20	0	10	0	0	56	468	12
YUV to HEVC	1080p	20	0	10	0	0	52	471	11
YUV to AV1	1080p	20	0	10	0	0	58	448	12
AVC to YUV	720p	1	0	8	1	23	0	562	40
HEVC to YUV	720p	1	0	8	1	21	0	640	33
VP9 to YUV	720p	1	0	8	1	21	0	377	19
AVC to YUV	720p	40	0	8	0	93	0	1910	4
HEVC to YUV	720p	40	0	8	0	80	0	1743	2
VP9 to YUV	720p	40	0	8	0	90	0	1555	1
YUV to AVC	720p	1	0	8	1	0	27	518	28
YUV to HEVC	720p	1	0	8	1	0	26	518	34
YUV to AV1	720p	1	0	8	1	0	24	415	42
YUV to AVC	720p	40	0	8	0	0	93	1770	5
YUV to HEVC	720p	40	0	8	0	0	96	1828	6
YUV to AV1	720p	40	0	8	0	0	97	1606	5
AVC to AVC	720p	1	1	8	0	50	57	908	58
AVC to HEVC	720p	1	1	8	0	33	44	882	64
AVC to AV1	720p	1	1	8	0	39	57	836	62
HEVC to AVC	720p	1	1	8	0	32	47	923	47
HEVC to HEVC	720p	1	1	8	0	40	57	894	58
HEVC to AV1	720p	1	1	8	0	29	52	814	69
VP9 to AVC	720p	1	1	8	0	21	20	376	49
VP9 to HEVC	720p	1	1	8	0	21	20	376	45

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	40
AVC to AVC	720p	40	1	8	0	100	100	1344	0
AVC to HEVC	720p	40	1	8	0	100	100	1474	1
AVC to AV1	720p	40	1	8	0	97	100	1332	1
HEVC to AVC	720p	40	1	8	0	100	100	1349	0
HEVC to HEVC	720p	40	1	8	0	100	100	1478	1
HEVC to AV1	720p	40	1	8	0	94	100	1337	1
VP9 to AVC	720p	40	1	8	0	100	98	1297	1
VP9 to HEVC	720p	40	1	8	0	100	98	1407	1
VP9 to AV1	720p	40	1	8	0	100	100	1340	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	24.75	28.31	24.1	0.16
YUV to HEVC	4k	1	26.72	30.14	25.06	0.23
YUV to AV1	4k	1	34.81	41	26.7	1.58
YUV to AVC	4k	3	26.62	31.29	24.18	1.76
YUV to HEVC	4k	3	27.92	36.97	25.48	2.15
YUV to AV1	4k	3	35	41.97	26.55	1.56
YUV to AVC	4k	5	28.43	33.27	24.65	4.86
YUV to HEVC	4k	5	33.89	57.11	25.5	67.98
YUV to AV1	4k	5	43.86	77.19	26.34	132.44
YUV to AVC	1080p	1	7.66	9.25	7.13	0.06
YUV to HEVC	1080p	1	8.09	8.85	7.4	0.05
YUV to AV1	1080p	1	10.7	12	7.84	0.15
YUV to AVC	1080p	20	11.94	21.44	7.44	6.76
YUV to HEVC	1080p	20	12.94	24.17	7.86	9.52
YUV to AV1	1080p	20	39.97	55.9	29.43	15.34
YUV to AVC	720p	1	4.38	5.54	3.9	0.06
YUV to HEVC	720p	1	4.54	5.64	3.96	0.07
YUV to AV1	720p	1	5.92	6.44	4.88	0.05
YUV to AVC	720p	40	7.2	15.66	4.05	1.97
YUV to HEVC	720p	40	6.97	12.71	4.43	1.58
YUV to AV1	720p	40	38.9	47.18	31.08	1.73

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	21
AVC to RGBA	4k	8	93	1	341	16
HEVC to RGBA	4k	1	22	0	100	26
HEVC to RGBA	4k	8	93	2	408	23
VP9 to RGBA	4k	1	23	0	83	21
VP9 to RGBA	4k	8	93	1	336	17
AVC to RGBA	1080p	20	92	9	1194	7
HEVC to RGBA	1080p	20	92	8	1171	7
VP9 to RGBA	1080p	20	94	6	897	5
AVC to RGBA	720p	50	96	19	2216	3
HEVC to RGBA	720p	50	91	20	2303	4
VP9 to RGBA	720p	32	93	12	1672	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	34	92	4	250	30
AVC to HEVC	5	33	94	3	280	32
AVC to AV1	5	26	93	2	250	32
HEVC to AVC	5	35	92	4	255	32
HEVC to HEVC	5	34	94	3	285	35
HEVC to AV1	5	26	93	2	255	34
VP9 to AVC	5	40	95	4	250	30
VP9 to HEVC	5	39	94	3	281	32
VP9 to AV1	5	31	93	2	251	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	38	80	63
RGBA to HEVC	4k	1	1	37	80	54
RGBA to AV1	4k	1	1	42	80	59
RGBA to AVC	4k	3	0	47	90	49
RGBA to HEVC	4k	3	0	41	90	52
RGBA to AV1	4k	3	0	47	89	48
RGBA to AVC	4k	5	0	50	91	35
RGBA to HEVC	4k	5	0	45	90	37
RGBA to AV1	4k	5	0	49	90	39
RGBA to AVC	1080p	1	1	29	244	81
RGBA to HEVC	1080p	1	1	28	247	72
RGBA to AV1	1080p	1	1	32	241	63
RGBA to AVC	1080p	10	0	45	334	24
RGBA to HEVC	1080p	10	0	40	337	24
RGBA to AV1	1080p	10	0	45	337	25
RGBA to AVC	1080p	20	0	45	338	13
RGBA to HEVC	1080p	20	0	41	340	13
RGBA to AV1	1080p	20	0	47	341	13
RGBA to AVC	720p	1	1	19	374	100
RGBA to HEVC	720p	1	1	19	374	100
RGBA to AV1	720p	1	1	22	373	108
RGBA to AVC	720p	10	0	37	682	44
RGBA to HEVC	720p	10	0	36	690	45
RGBA to AV1	720p	10	0	42	668	44
RGBA to AVC	720p	20	0	40	730	23
RGBA to HEVC	720p	20	0	39	739	20
RGBA to AV1	720p	20	0	44	719	20

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	40
YUV to HEVC	4k	2	0	0	1	48	108	40
YUV to AV1	4k	2	0	0	1	48	94	39
YUV to AVC	4k	2	0	0	2	49	102	40
YUV to HEVC	4k	2	0	0	2	49	54	26
YUV to AV1	4k	2	0	0	2	48	44	24
YUV to AVC	4k	2	0	0	3	49	102	36
YUV to HEVC	4k	2	0	0	3	49	32	19
YUV to AV1	4k	2	0	0	3	49	24	18
YUV to AVC	4k	2	0	1	1	48	60	30
YUV to HEVC	4k	2	0	1	1	49	76	33
YUV to AVC	4k	2	0	1	2	50	60	28
YUV to HEVC	4k	2	0	1	2	51	34	20
YUV to AVC	4k	2	0	1	3	50	60	30
YUV to HEVC	4k	2	0	1	3	49	22	14
YUV to AVC	4k	2	4	0	1	80	100	44
YUV to HEVC	4k	2	4	0	1	74	108	44
YUV to AV1	4k	2	4	0	1	73	94	42
YUV to AVC	4k	2	4	0	2	79	100	41
YUV to HEVC	4k	2	4	0	2	63	54	26
YUV to AV1	4k	2	4	0	2	61	44	25
YUV to AVC	4k	2	4	0	3	79	100	38
YUV to HEVC	4k	2	4	0	3	56	32	20
YUV to AV1	4k	2	4	0	3	56	26	20
YUV to AVC	4k	2	4	1	1	66	60	30
YUV to HEVC	4k	2	4	1	1	67	76	34
YUV to AVC	4k	2	4	1	2	66	60	31
YUV to HEVC	4k	2	4	1	2	57	34	20
YUV to AVC	4k	2	4	1	3	67	60	31
YUV to HEVC	4k	2	4	1	3	54	22	19
YUV to AVC	4k	2	16	0	1	78	100	40
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	79	100	42
YUV to HEVC	4k	2	16	0	2	62	54	25
YUV to AV1	4k	2	16	0	2	60	44	26
YUV to AVC	4k	2	16	0	3	78	100	42
YUV to HEVC	4k	2	16	0	3	56	32	18
YUV to AV1	4k	2	16	0	3	58	26	19

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	16	1	1	66	60	28
YUV to HEVC	4k	2	16	1	1	67	76	36
YUV to AVC	4k	2	16	1	2	67	60	30
YUV to HEVC	4k	2	16	1	2	59	34	22
YUV to AVC	4k	2	16	1	3	67	60	30
YUV to HEVC	4k	2	16	1	3	54	22	16
YUV to AVC	4k	2	40	0	1	78	99	41
YUV to HEVC	4k	2	40	0	1	74	106	43
YUV to AV1	4k	2	40	0	1	73	92	37
YUV to AVC	4k	2	40	0	2	78	98	42
YUV to HEVC	4k	2	40	0	2	62	54	29
YUV to AV1	4k	2	40	0	2	61	44	24
YUV to AVC	4k	2	40	0	3	79	98	44
YUV to HEVC	4k	2	40	0	3	58	32	19
YUV to AV1	4k	2	40	0	3	57	26	19
YUV to AVC	4k	2	40	1	1	66	60	30
YUV to HEVC	4k	2	40	1	1	67	76	36
YUV to AVC	4k	2	40	1	2	67	60	30
YUV to HEVC	4k	2	40	1	2	59	34	24
YUV to AVC	4k	2	40	1	3	67	60	28
YUV to HEVC	4k	2	40	1	3	54	22	18
YUV to AVC	1080p	10	0	0	1	99	812	25
YUV to HEVC	1080p	10	0	0	1	95	830	26
YUV to AV1	1080p	10	0	0	1	99	749	23
YUV to AVC	1080p	10	0	0	2	99	808	26
YUV to HEVC	1080p	10	0	0	2	100	439	21
YUV to AV1	1080p	10	0	0	2	99	355	18
YUV to AVC	1080p	10	0	0	3	99	806	26
YUV to HEVC	1080p	10	0	0	3	99	260	14
YUV to AV1	1080p	10	0	0	3	100	189	16
YUV to AVC	1080p	10	0	1	1	100	481	20
YUV to HEVC	1080p	10	0	1	1	99	616	23
YUV to AVC	1080p	10	0	1	2	99	482	22
YUV to HEVC	1080p	10	0	1	2	99	280	19
YUV to AVC	1080p	10	0	1	3	100	482	21
YUV to HEVC	1080p	10	0	1	3	99	178	15
YUV to AVC	1080p	10	4	0	1	99	449	19
YUV to HEVC	1080p	10	4	0	1	99	524	20
YUV to AV1	1080p	10	4	0	1	99	460	18
YUV to AVC	1080p	10	4	0	2	99	450	19

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	4	0	2	100	330	17
YUV to AV1	1080p	10	4	0	2	100	270	17
YUV to AVC	1080p	10	4	0	3	99	450	19
YUV to HEVC	1080p	10	4	0	3	99	217	16
YUV to AV1	1080p	10	4	0	3	99	171	16
YUV to AVC	1080p	10	4	1	1	100	323	18
YUV to HEVC	1080p	10	4	1	1	99	419	19
YUV to AVC	1080p	10	4	1	2	99	322	18
YUV to HEVC	1080p	10	4	1	2	100	230	17
YUV to AVC	1080p	10	4	1	3	99	325	18
YUV to HEVC	1080p	10	4	1	3	99	155	15
YUV to AVC	1080p	10	16	0	1	99	449	20
YUV to HEVC	1080p	10	16	0	1	99	524	19
YUV to AV1	1080p	10	16	0	1	99	460	20
YUV to AVC	1080p	10	16	0	2	99	447	21
YUV to HEVC	1080p	10	16	0	2	99	329	18
YUV to AV1	1080p	10	16	0	2	99	270	16
YUV to AVC	1080p	10	16	0	3	99	448	19
YUV to HEVC	1080p	10	16	0	3	99	216	17
YUV to AV1	1080p	10	16	0	3	100	170	15
YUV to AVC	1080p	10	16	1	1	100	321	18
YUV to HEVC	1080p	10	16	1	1	100	419	19
YUV to AVC	1080p	10	16	1	2	100	321	19
YUV to HEVC	1080p	10	16	1	2	99	230	16
YUV to AVC	1080p	10	16	1	3	100	321	18
YUV to HEVC	1080p	10	16	1	3	99	155	16
YUV to AVC	1080p	10	40	0	1	99	442	19
YUV to HEVC	1080p	10	40	0	1	99	519	21
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	100	320	19
YUV to AV1	1080p	10	40	0	2	99	270	17
YUV to AVC	1080p	10	40	0	3	99	441	21
YUV to HEVC	1080p	10	40	0	3	99	213	17
YUV to AV1	1080p	10	40	0	3	99	171	14
YUV to AVC	1080p	10	40	1	1	99	320	19
YUV to HEVC	1080p	10	40	1	1	99	410	20
YUV to AVC	1080p	10	40	1	2	100	321	17
YUV to HEVC	1080p	10	40	1	2	99	230	17
YUV to AVC	1080p	10	40	1	3	100	320	17

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	1	3	99	153	16
YUV to AVC	720p	20	0	0	1	80	1508	15
YUV to HEVC	720p	20	0	0	1	83	1553	14
YUV to AV1	720p	20	0	0	1	86	1402	12
YUV to AVC	720p	20	0	0	2	85	1546	14
YUV to HEVC	720p	20	0	0	2	99	961	10
YUV to AV1	720p	20	0	0	2	99	780	8
YUV to AVC	720p	20	0	0	3	83	1524	14
YUV to HEVC	720p	20	0	0	3	99	580	6
YUV to AV1	720p	20	0	0	3	99	420	9
YUV to AVC	720p	20	0	1	1	99	1080	13
YUV to HEVC	720p	20	0	1	1	99	1322	15
YUV to AVC	720p	20	0	1	2	99	1080	12
YUV to HEVC	720p	20	0	1	2	99	620	6
YUV to AVC	720p	20	0	1	3	99	1080	11
YUV to HEVC	720p	20	0	1	3	99	400	10
YUV to AVC	720p	20	4	0	1	99	921	14
YUV to HEVC	720p	20	4	0	1	99	1060	13
YUV to AV1	720p	20	4	0	1	99	880	7
YUV to AVC	720p	20	4	0	2	99	922	14
YUV to HEVC	720p	20	4	0	2	99	682	8
YUV to AV1	720p	20	4	0	2	99	560	12
YUV to AVC	720p	20	4	0	3	99	920	11
YUV to HEVC	720p	20	4	0	3	99	460	7
YUV to AV1	720p	20	4	0	3	100	360	7
YUV to AVC	720p	20	4	1	1	99	680	13
YUV to HEVC	720p	20	4	1	1	99	860	11
YUV to AVC	720p	20	4	1	2	99	680	13
YUV to HEVC	720p	20	4	1	2	99	480	10
YUV to AVC	720p	20	4	1	3	99	680	8
YUV to HEVC	720p	20	4	1	3	100	340	9
YUV to AVC	720p	20	16	0	1	99	920	11
YUV to HEVC	720p	20	16	0	1	99	1059	13
YUV to AV1	720p	20	16	0	1	100	918	7
YUV to AVC	720p	20	16	0	2	99	920	14
YUV to HEVC	720p	20	16	0	2	99	680	15
YUV to AV1	720p	20	16	0	2	99	560	9
YUV to AVC	720p	20	16	0	3	99	920	13
YUV to HEVC	720p	20	16	0	3	99	460	11
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	13
YUV to HEVC	720p	20	16	1	1	99	860	10
YUV to AVC	720p	20	16	1	2	99	680	11
YUV to HEVC	720p	20	16	1	2	99	480	15
YUV to AVC	720p	20	16	1	3	99	680	14
YUV to HEVC	720p	20	16	1	3	100	340	8
YUV to AVC	720p	20	40	0	1	99	915	14
YUV to HEVC	720p	20	40	0	1	99	1043	13
YUV to AV1	720p	20	40	0	1	99	900	7
YUV to AVC	720p	20	40	0	2	99	914	12
YUV to HEVC	720p	20	40	0	2	99	680	14
YUV to AV1	720p	20	40	0	2	99	560	12
YUV to AVC	720p	20	40	0	3	99	917	14
YUV to HEVC	720p	20	40	0	3	99	460	8
YUV to AV1	720p	20	40	0	3	99	359	7
YUV to AVC	720p	20	40	1	1	99	673	13
YUV to HEVC	720p	20	40	1	1	99	841	12
YUV to AVC	720p	20	40	1	2	99	675	9
YUV to HEVC	720p	20	40	1	2	99	480	15
YUV to AVC	720p	20	40	1	3	99	677	11
YUV to HEVC	720p	20	40	1	3	100	340	10

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	451	20
YUV to HEVC	1080p	10	0	0	1	19	99	524	20
YUV to AV1	1080p	10	0	0	1	19	99	461	19
YUV to AVC	1080p	10	0	0	2	19	99	450	19
YUV to HEVC	1080p	10	0	0	2	19	99	330	19
YUV to AV1	1080p	10	0	0	2	19	100	280	17
YUV to AVC	1080p	10	0	0	3	19	99	451	20
YUV to HEVC	1080p	10	0	0	3	19	99	218	16
YUV to AV1	1080p	10	0	0	3	19	99	177	15
YUV to AVC	1080p	10	0	1	1	19	99	324	17
YUV to HEVC	1080p	10	0	1	1	19	99	420	19
YUV to AVC	1080p	10	0	1	2	19	99	326	17
YUV to HEVC	1080p	10	0	1	2	19	99	230	16
YUV to AVC	1080p	10	0	1	3	19	99	324	18
YUV to HEVC	1080p	10	0	1	3	19	100	157	16
YUV to AVC	1080p	10	4	0	1	19	99	450	19
YUV to HEVC	1080p	10	4	0	1	19	99	523	20
YUV to AV1	1080p	10	4	0	1	19	99	460	17
YUV to AVC	1080p	10	4	0	2	19	99	449	19
YUV to HEVC	1080p	10	4	0	2	19	99	330	18
YUV to AV1	1080p	10	4	0	2	19	100	270	17
YUV to AVC	1080p	10	4	0	3	19	99	450	18
YUV to HEVC	1080p	10	4	0	3	19	99	216	16
YUV to AV1	1080p	10	4	0	3	19	100	170	15
YUV to AVC	1080p	10	4	1	1	19	99	324	17
YUV to HEVC	1080p	10	4	1	1	19	99	420	19
YUV to AVC	1080p	10	4	1	2	19	99	325	18
YUV to HEVC	1080p	10	4	1	2	19	99	230	16
YUV to AVC	1080p	10	4	1	3	19	99	323	18
YUV to HEVC	1080p	10	4	1	3	19	99	157	15
YUV to AVC	1080p	10	16	0	1	19	99	448	19
YUV to HEVC	1080p	10	16	0	1	19	99	521	20
YUV to AV1	1080p	10	16	0	1	19	99	461	19
YUV to AVC	1080p	10	16	0	2	19	99	449	18
YUV to HEVC	1080p	10	16	0	2	19	100	329	18
YUV to AV1	1080p	10	16	0	2	19	100	270	18
YUV to AVC	1080p	10	16	0	3	19	99	448	18
YUV to HEVC	1080p	10	16	0	3	19	100	215	15

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	3	19	100	170	14
YUV to AVC	1080p	10	16	1	1	19	99	321	17
YUV to HEVC	1080p	10	16	1	1	19	100	418	18
YUV to AVC	1080p	10	16	1	2	19	100	323	16
YUV to HEVC	1080p	10	16	1	2	19	99	230	17
YUV to AVC	1080p	10	16	1	3	19	99	321	18
YUV to HEVC	1080p	10	16	1	3	19	100	156	16
YUV to AVC	1080p	10	40	0	1	19	99	441	19
YUV to HEVC	1080p	10	40	0	1	19	99	520	19
YUV to AV1	1080p	10	40	0	1	19	99	453	17
YUV to AVC	1080p	10	40	0	2	19	99	442	20
YUV to HEVC	1080p	10	40	0	2	19	99	320	17
YUV to AV1	1080p	10	40	0	2	19	99	270	16
YUV to AVC	1080p	10	40	0	3	19	99	442	18
YUV to HEVC	1080p	10	40	0	3	19	100	210	16
YUV to AV1	1080p	10	40	0	3	19	99	170	16
YUV to AVC	1080p	10	40	1	1	19	99	320	17
YUV to HEVC	1080p	10	40	1	1	19	99	411	18
YUV to AVC	1080p	10	40	1	2	19	100	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	19
YUV to HEVC	1080p	10	40	1	3	19	99	153	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	19
YUV to HEVC	1080p	10	0	0	2	23	99	330	18
YUV to AV1	1080p	10	0	0	2	23	99	279	17
YUV to AVC	1080p	10	0	0	3	23	99	450	20
YUV to HEVC	1080p	10	0	0	3	23	99	216	17
YUV to AV1	1080p	10	0	0	3	23	99	176	15
YUV to AVC	1080p	10	0	1	1	23	99	325	19
YUV to HEVC	1080p	10	0	1	1	23	99	420	20
YUV to AVC	1080p	10	0	1	2	23	99	325	18
YUV to HEVC	1080p	10	0	1	2	23	100	230	17
YUV to AVC	1080p	10	0	1	3	23	99	325	17
YUV to HEVC	1080p	10	0	1	3	23	100	156	10
YUV to AVC	1080p	10	4	0	1	23	99	449	19
YUV to HEVC	1080p	10	4	0	1	23	99	521	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	4	0	1	23	99	459	18
YUV to AVC	1080p	10	4	0	2	23	99	451	19
YUV to HEVC	1080p	10	4	0	2	23	99	330	17
YUV to AV1	1080p	10	4	0	2	23	99	270	17
YUV to AVC	1080p	10	4	0	3	23	99	450	20
YUV to HEVC	1080p	10	4	0	3	23	99	216	16
YUV to AV1	1080p	10	4	0	3	23	99	170	15
YUV to AVC	1080p	10	4	1	1	23	99	324	17
YUV to HEVC	1080p	10	4	1	1	23	100	419	19
YUV to AVC	1080p	10	4	1	2	23	100	323	17
YUV to HEVC	1080p	10	4	1	2	23	100	230	16
YUV to AVC	1080p	10	4	1	3	23	100	324	19
YUV to HEVC	1080p	10	4	1	3	23	100	157	17
YUV to AVC	1080p	10	16	0	1	23	99	449	18
YUV to HEVC	1080p	10	16	0	1	23	99	522	20
YUV to AV1	1080p	10	16	0	1	23	99	460	19
YUV to AVC	1080p	10	16	0	2	23	99	446	20
YUV to HEVC	1080p	10	16	0	2	23	99	327	19
YUV to AV1	1080p	10	16	0	2	23	99	270	18
YUV to AVC	1080p	10	16	0	3	23	99	447	19
YUV to HEVC	1080p	10	16	0	3	23	99	215	16
YUV to AV1	1080p	10	16	0	3	23	100	170	15
YUV to AVC	1080p	10	16	1	1	23	100	321	19
YUV to HEVC	1080p	10	16	1	1	23	99	418	18
YUV to AVC	1080p	10	16	1	2	23	99	320	18
YUV to HEVC	1080p	10	16	1	2	23	99	230	17
YUV to AVC	1080p	10	16	1	3	23	99	323	17
YUV to HEVC	1080p	10	16	1	3	23	101	157	16
YUV to AVC	1080p	10	40	0	1	23	99	443	19
YUV to HEVC	1080p	10	40	0	1	23	99	520	17
YUV to AV1	1080p	10	40	0	1	23	99	455	18
YUV to AVC	1080p	10	40	0	2	23	99	440	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	443	19
YUV to HEVC	1080p	10	40	0	3	23	99	213	16
YUV to AV1	1080p	10	40	0	3	23	99	170	15
YUV to AVC	1080p	10	40	1	1	23	99	320	16
YUV to HEVC	1080p	10	40	1	1	23	99	410	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	40	1	2	23	99	320	17
YUV to HEVC	1080p	10	40	1	2	23	100	230	16
YUV to AVC	1080p	10	40	1	3	23	100	320	18
YUV to HEVC	1080p	10	40	1	3	23	99	153	15
YUV to AVC	1080p	10	0	0	1	27	99	450	19
YUV to HEVC	1080p	10	0	0	1	27	99	523	20
YUV to AV1	1080p	10	0	0	1	27	99	461	19
YUV to AVC	1080p	10	0	0	2	27	99	450	19
YUV to HEVC	1080p	10	0	0	2	27	99	330	18
YUV to AV1	1080p	10	0	0	2	27	100	280	16
YUV to AVC	1080p	10	0	0	3	27	99	450	18
YUV to HEVC	1080p	10	0	0	3	27	99	218	17
YUV to AV1	1080p	10	0	0	3	27	99	178	15
YUV to AVC	1080p	10	0	1	1	27	99	327	18
YUV to HEVC	1080p	10	0	1	1	27	100	420	19
YUV to AVC	1080p	10	0	1	2	27	99	326	16
YUV to HEVC	1080p	10	0	1	2	27	99	230	16
YUV to AVC	1080p	10	0	1	3	27	99	325	19
YUV to HEVC	1080p	10	0	1	3	27	100	157	17
YUV to AVC	1080p	10	4	0	1	27	99	450	20
YUV to HEVC	1080p	10	4	0	1	27	99	522	20
YUV to AV1	1080p	10	4	0	1	27	99	460	18
YUV to AVC	1080p	10	4	0	2	27	99	449	19
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	450	19
YUV to HEVC	1080p	10	4	0	3	27	100	218	16
YUV to AV1	1080p	10	4	0	3	27	100	171	15
YUV to AVC	1080p	10	4	1	1	27	99	321	19
YUV to HEVC	1080p	10	4	1	1	27	99	420	19
YUV to AVC	1080p	10	4	1	2	27	99	324	18
YUV to HEVC	1080p	10	4	1	2	27	99	230	16
YUV to AVC	1080p	10	4	1	3	27	99	324	18
YUV to HEVC	1080p	10	4	1	3	27	100	157	16
YUV to AVC	1080p	10	16	0	1	27	99	448	19
YUV to HEVC	1080p	10	16	0	1	27	99	521	20
YUV to AV1	1080p	10	16	0	1	27	99	459	18
YUV to AVC	1080p	10	16	0	2	27	99	449	20
YUV to HEVC	1080p	10	16	0	2	27	99	328	17

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	2	27	99	270	16
YUV to AVC	1080p	10	16	0	3	27	99	448	18
YUV to HEVC	1080p	10	16	0	3	27	99	215	16
YUV to AV1	1080p	10	16	0	3	27	99	170	15
YUV to AVC	1080p	10	16	1	1	27	99	320	18
YUV to HEVC	1080p	10	16	1	1	27	99	419	18
YUV to AVC	1080p	10	16	1	2	27	99	321	18
YUV to HEVC	1080p	10	16	1	2	27	100	230	17
YUV to AVC	1080p	10	16	1	3	27	99	320	18
YUV to HEVC	1080p	10	16	1	3	27	100	156	16
YUV to AVC	1080p	10	40	0	1	27	99	441	19
YUV to HEVC	1080p	10	40	0	1	27	99	520	18
YUV to AV1	1080p	10	40	0	1	27	99	454	18
YUV to AVC	1080p	10	40	0	2	27	99	442	18
YUV to HEVC	1080p	10	40	0	2	27	99	320	18
YUV to AV1	1080p	10	40	0	2	27	100	270	16
YUV to AVC	1080p	10	40	0	3	27	99	441	20
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	18
YUV to AVC	1080p	10	40	1	2	27	100	320	17
YUV to HEVC	1080p	10	40	1	2	27	99	230	17
YUV to AVC	1080p	10	40	1	3	27	99	320	16
YUV to HEVC	1080p	10	40	1	3	27	101	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	190	55
AVC to HEVC	1	15	21	3	202	57
AVC to AV1	1	13	21	2	174	52
HEVC to AVC	1	14	21	3	187	63
HEVC to HEVC	1	15	21	3	200	65
HEVC to AV1	1	13	20	2	172	57
VP9 to AVC	1	19	21	3	190	49
VP9 to HEVC	1	20	20	3	201	52
VP9 to AV1	1	17	20	2	174	47
AVC to AVC	8	71	92	18	536	24
AVC to HEVC	8	77	92	20	608	24
AVC to AV1	8	73	92	19	608	28
HEVC to AVC	8	66	92	18	560	31
HEVC to HEVC	8	72	93	21	638	30
HEVC to AV1	8	67	95	21	632	30
VP9 to AVC	8	76	94	18	536	25
VP9 to HEVC	8	83	93	19	611	26
VP9 to AV1	8	82	94	20	608	27
AVC to AVC	16	73	95	19	529	11
AVC to HEVC	16	79	94	21	608	12
AVC to AV1	16	76	95	21	601	11
HEVC to AVC	16	65	92	20	560	11
HEVC to HEVC	16	74	95	22	640	12
HEVC to AV1	16	70	95	21	624	13
VP9 to AVC	16	77	95	19	534	11
VP9 to HEVC	16	82	92	20	608	9
VP9 to AV1	16	82	94	20	608	12

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	59	182	50
YUV to HEVC	4k	1	0	8	1	0	55	188	55
YUV to AV1	4k	1	0	8	1	0	60	182	59
YUV to AVC	4k	3	0	8	0	0	71	198	47
YUV to HEVC	4k	3	0	8	0	0	66	222	57
YUV to AV1	4k	3	0	8	0	0	71	213	44
YUV to AVC	4k	5	0	8	0	0	88	205	35
YUV to HEVC	4k	5	0	8	0	0	69	228	46
YUV to AV1	4k	5	0	8	0	0	78	226	44
YUV to AVC	4k	1	0	10	1	0	35	106	48
YUV to HEVC	4k	1	0	10	1	0	31	108	54
YUV to AV1	4k	1	0	10	1	0	35	106	55
YUV to AVC	4k	3	0	10	0	0	50	115	55
YUV to HEVC	4k	3	0	10	0	0	35	117	59
YUV to AV1	4k	3	0	10	0	0	40	116	60
YUV to AVC	1080p	1	0	8	1	0	34	450	88
YUV to HEVC	1080p	1	0	8	1	0	33	456	80
YUV to AV1	1080p	1	0	8	1	0	32	405	85
YUV to AVC	1080p	20	0	8	0	0	72	812	14
YUV to HEVC	1080p	20	0	8	0	0	65	852	15
YUV to AV1	1080p	20	0	8	0	0	70	818	14
YUV to AVC	1080p	1	0	10	1	0	23	309	65
YUV to HEVC	1080p	1	0	10	1	0	22	310	63
YUV to AV1	1080p	1	0	10	1	0	24	288	70
YUV to AVC	1080p	20	0	10	0	0	39	463	13
YUV to HEVC	1080p	20	0	10	0	0	34	465	13
YUV to AV1	1080p	20	0	10	0	0	39	460	13
YUV to AVC	720p	1	0	8	1	0	20	567	63
YUV to HEVC	720p	1	0	8	1	0	19	576	68
YUV to AV1	720p	1	0	8	1	0	23	512	57
YUV to AVC	720p	40	0	8	0	0	84	1574	8
YUV to HEVC	720p	40	0	8	0	0	81	1676	8
YUV to AV1	720p	40	0	8	0	0	85	1452	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	33	114	53
YUV to HEVC	4k	1	0	8	1	0	33	123	49
YUV to AV1	4k	1	0	8	1	0	35	107	54
YUV to AVC	4k	3	0	8	0	0	71	172	37
YUV to HEVC	4k	3	0	8	0	0	49	163	39
YUV to AV1	4k	3	0	8	0	0	53	158	37
YUV to AVC	4k	5	0	8	0	0	96	194	29
YUV to HEVC	4k	5	0	8	0	0	64	212	34
YUV to AV1	4k	5	0	8	0	0	65	194	34
YUV to AVC	4k	1	0	10	1	0	20	66	51
YUV to HEVC	4k	1	0	10	1	0	18	67	51
YUV to AV1	4k	1	0	10	1	0	21	64	50
YUV to AVC	4k	3	0	10	0	0	30	92	40
YUV to HEVC	4k	3	0	10	0	0	26	91	40
YUV to AV1	4k	3	0	10	0	0	27	85	37
YUV to AVC	1080p	1	0	8	1	0	31	350	46
YUV to HEVC	1080p	1	0	8	1	0	23	339	47
YUV to AV1	1080p	1	0	8	1	0	25	296	42
YUV to AVC	1080p	20	0	8	0	0	73	837	11
YUV to HEVC	1080p	20	0	8	0	0	66	870	11
YUV to AV1	1080p	20	0	8	0	0	70	827	11
YUV to AVC	1080p	1	0	10	1	0	16	209	51
YUV to HEVC	1080p	1	0	10	1	0	14	208	52
YUV to AV1	1080p	1	0	10	1	0	15	183	52
YUV to AVC	1080p	20	0	10	0	0	39	469	12
YUV to HEVC	1080p	20	0	10	0	0	34	471	12
YUV to AV1	1080p	20	0	10	0	0	38	442	12
YUV to AVC	720p	1	0	8	1	0	18	523	36
YUV to HEVC	720p	1	0	8	1	0	17	526	42
YUV to AV1	720p	1	0	8	1	0	18	415	45
YUV to AVC	720p	40	0	8	0	0	84	1705	5
YUV to HEVC	720p	40	0	8	0	0	79	1787	6
YUV to AV1	720p	40	0	8	0	0	88	1567	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	19.97	17.28	0.09
YUV to HEVC	4k	1	19.09	21.24	18.02	0.13
YUV to AV1	4k	1	24.27	28.12	18.82	0.67
YUV to AVC	4k	3	19.57	23.97	17.32	1.37
YUV to HEVC	4k	3	20.93	23.77	17.94	1
YUV to AV1	4k	3	25.84	30.88	20.86	1.48
YUV to AVC	4k	5	23	29.12	17.46	11.79
YUV to HEVC	4k	5	22.8	28.57	18.34	2.21
YUV to AV1	4k	5	28.64	36.78	19.38	3.41
YUV to AVC	1080p	1	5.8	6.99	5.09	0.08
YUV to HEVC	1080p	1	6.01	6.59	5.55	0.04
YUV to AV1	1080p	1	7.73	8.68	6.33	0.06
YUV to AVC	1080p	20	7.46	10.61	5.38	0.92
YUV to HEVC	1080p	20	8.19	13.66	5.52	1.48
YUV to AV1	1080p	20	11.64	20.3	6.83	5.57
YUV to AVC	720p	1	3.62	4.71	3.12	0.07
YUV to HEVC	720p	1	3.61	4.08	3.24	0.02
YUV to AV1	720p	1	4.5	5.07	3.68	0.05
YUV to AVC	720p	40	5.37	10.75	3.04	1.1
YUV to HEVC	720p	40	5.54	9.36	3.47	1.07
YUV to AV1	720p	40	7.5	14.82	4.13	1.7

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	84	56
RGBA to HEVC	4k	1	1	24	83	65
RGBA to AV1	4k	1	1	28	82	61
RGBA to AVC	4k	3	0	38	90	49
RGBA to HEVC	4k	3	0	33	89	51
RGBA to AV1	4k	3	0	33	89	49
RGBA to AVC	4k	5	0	48	91	38
RGBA to HEVC	4k	5	0	39	90	34
RGBA to AV1	4k	5	0	40	90	37
RGBA to AVC	1080p	1	1	19	255	63
RGBA to HEVC	1080p	1	1	18	256	77
RGBA to AV1	1080p	1	1	21	248	81
RGBA to AVC	1080p	10	0	33	331	23
RGBA to HEVC	1080p	10	0	30	333	24
RGBA to AV1	1080p	10	0	31	332	23
RGBA to AVC	1080p	20	0	38	330	13
RGBA to HEVC	1080p	20	0	31	340	13
RGBA to AV1	1080p	20	0	34	336	13
RGBA to AVC	720p	1	1	13	351	110
RGBA to HEVC	720p	1	1	17	379	101
RGBA to AV1	720p	1	1	16	378	116
RGBA to AVC	720p	10	0	34	698	46
RGBA to HEVC	720p	10	0	33	707	47
RGBA to AV1	720p	10	0	39	688	46
RGBA to AVC	720p	20	0	36	718	23
RGBA to HEVC	720p	20	0	35	727	25
RGBA to AV1	720p	20	0	43	714	21

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	47	154	58
YUV to HEVC	4k	2	0	0	1	48	166	53
YUV to AV1	4k	2	0	0	1	48	144	49
YUV to AVC	4k	2	0	0	2	48	154	51
YUV to HEVC	4k	2	0	0	2	49	86	35
YUV to AV1	4k	2	0	0	2	49	70	34
YUV to AVC	4k	2	0	0	3	48	154	49
YUV to HEVC	4k	2	0	0	3	49	50	23
YUV to AV1	4k	2	0	0	3	49	38	23
YUV to AVC	4k	2	0	1	1	49	94	41
YUV to HEVC	4k	2	0	1	1	48	120	44
YUV to AVC	4k	2	0	1	2	49	94	39
YUV to HEVC	4k	2	0	1	2	49	54	28
YUV to AVC	4k	2	0	1	3	49	94	41
YUV to HEVC	4k	2	0	1	3	50	34	18
YUV to AVC	4k	2	4	0	1	74	132	46
YUV to HEVC	4k	2	4	0	1	72	166	38
YUV to AV1	4k	2	4	0	1	72	144	52
YUV to AVC	4k	2	4	0	2	74	132	50
YUV to HEVC	4k	2	4	0	2	61	84	36
YUV to AV1	4k	2	4	0	2	60	70	33
YUV to AVC	4k	2	4	0	3	74	132	50
YUV to HEVC	4k	2	4	0	3	57	50	28
YUV to AV1	4k	2	4	0	3	56	40	23
YUV to AVC	4k	2	4	1	1	67	93	38
YUV to HEVC	4k	2	4	1	1	66	120	43
YUV to AVC	4k	2	4	1	2	66	92	39
YUV to HEVC	4k	2	4	1	2	58	54	26
YUV to AVC	4k	2	4	1	3	67	92	38
YUV to HEVC	4k	2	4	1	3	56	34	23
YUV to AVC	4k	2	20	0	1	73	132	45
YUV to HEVC	4k	2	20	0	1	73	166	52
YUV to AV1	4k	2	20	0	1	72	144	50

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	20	0	2	73	132	52
YUV to HEVC	4k	2	20	0	2	60	84	40
YUV to AV1	4k	2	20	0	2	60	70	35
YUV to AVC	4k	2	20	0	3	74	130	48
YUV to HEVC	4k	2	20	0	3	57	50	26
YUV to AV1	4k	2	20	0	3	56	40	25
YUV to AVC	4k	2	20	1	1	66	92	38
YUV to HEVC	4k	2	20	1	1	66	120	49
YUV to AVC	4k	2	20	1	2	67	92	39
YUV to HEVC	4k	2	20	1	2	58	54	28
YUV to AVC	4k	2	20	1	3	66	92	36
YUV to HEVC	4k	2	20	1	3	53	34	22
YUV to AVC	4k	2	40	0	1	73	126	50
YUV to HEVC	4k	2	40	0	1	73	164	58
YUV to AV1	4k	2	40	0	1	72	142	41
YUV to AVC	4k	2	40	0	2	74	130	49
YUV to HEVC	4k	2	40	0	2	62	84	34
YUV to AV1	4k	2	40	0	2	60	70	37
YUV to AVC	4k	2	40	0	3	74	126	42
YUV to HEVC	4k	2	40	0	3	56	50	29
YUV to AV1	4k	2	40	0	3	56	40	26
YUV to AVC	4k	2	40	1	1	67	92	38
YUV to HEVC	4k	2	40	1	1	66	118	46
YUV to AVC	4k	2	40	1	2	66	92	39
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	55	34	20
YUV to AVC	1080p	10	0	0	1	66	784	25
YUV to HEVC	1080p	10	0	0	1	61	816	26
YUV to AV1	1080p	10	0	0	1	65	770	25
YUV to AVC	1080p	10	0	0	2	65	776	25
YUV to HEVC	1080p	10	0	0	2	99	689	24
YUV to AV1	1080p	10	0	0	2	100	559	20
YUV to AVC	1080p	10	0	0	3	65	774	23

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	100	408	20
YUV to AV1	1080p	10	0	0	3	100	299	18
YUV to AVC	1080p	10	0	1	1	100	753	24
YUV to HEVC	1080p	10	0	1	1	85	825	26
YUV to AVC	1080p	10	0	1	2	99	753	24
YUV to HEVC	1080p	10	0	1	2	99	437	19
YUV to AVC	1080p	10	0	1	3	100	754	24
YUV to HEVC	1080p	10	0	1	3	100	278	18
YUV to AVC	1080p	10	4	0	1	100	648	20
YUV to HEVC	1080p	10	4	0	1	87	704	21
YUV to AV1	1080p	10	4	0	1	93	680	21
YUV to AVC	1080p	10	4	0	2	99	647	19
YUV to HEVC	1080p	10	4	0	2	99	510	21
YUV to AV1	1080p	10	4	0	2	100	430	18
YUV to AVC	1080p	10	4	0	3	99	648	21
YUV to HEVC	1080p	10	4	0	3	99	340	18
YUV to AV1	1080p	10	4	0	3	100	271	16
YUV to AVC	1080p	10	4	1	1	100	509	19
YUV to HEVC	1080p	10	4	1	1	99	654	21
YUV to AVC	1080p	10	4	1	2	99	510	21
YUV to HEVC	1080p	10	4	1	2	99	360	20
YUV to AVC	1080p	10	4	1	3	99	510	21
YUV to HEVC	1080p	10	4	1	3	100	248	18
YUV to AVC	1080p	10	20	0	1	99	640	19
YUV to HEVC	1080p	10	20	0	1	88	713	22
YUV to AV1	1080p	10	20	0	1	92	671	19
YUV to AVC	1080p	10	20	0	2	99	640	18
YUV to HEVC	1080p	10	20	0	2	99	510	21
YUV to AV1	1080p	10	20	0	2	99	430	18
YUV to AVC	1080p	10	20	0	3	99	641	20
YUV to HEVC	1080p	10	20	0	3	99	340	18
YUV to AV1	1080p	10	20	0	3	100	270	17
YUV to AVC	1080p	10	20	1	1	99	508	21
YUV to HEVC	1080p	10	20	1	1	99	650	21

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	20	1	2	99	508	20
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	99	244	17
YUV to AVC	1080p	10	40	0	1	99	630	19
YUV to HEVC	1080p	10	40	0	1	87	698	20
YUV to AV1	1080p	10	40	0	1	90	640	18
YUV to AVC	1080p	10	40	0	2	99	630	19
YUV to HEVC	1080p	10	40	0	2	99	508	20
YUV to AV1	1080p	10	40	0	2	100	421	18
YUV to AVC	1080p	10	40	0	3	99	630	19
YUV to HEVC	1080p	10	40	0	3	100	335	20
YUV to AV1	1080p	10	40	0	3	99	270	17
YUV to AVC	1080p	10	40	1	1	99	501	19
YUV to HEVC	1080p	10	40	1	1	99	648	20
YUV to AVC	1080p	10	40	1	2	99	500	21
YUV to HEVC	1080p	10	40	1	2	99	357	19
YUV to AVC	1080p	10	40	1	3	100	500	20
YUV to HEVC	1080p	10	40	1	3	100	241	17
YUV to AVC	720p	20	0	0	1	73	1463	13
YUV to HEVC	720p	20	0	0	1	70	1541	14
YUV to AV1	720p	20	0	0	1	77	1388	13
YUV to AVC	720p	20	0	0	2	77	1502	14
YUV to HEVC	720p	20	0	0	2	99	1503	15
YUV to AV1	720p	20	0	0	2	99	1220	15
YUV to AVC	720p	20	0	0	3	75	1476	14
YUV to HEVC	720p	20	0	0	3	100	904	11
YUV to AV1	720p	20	0	0	3	99	660	15
YUV to AVC	720p	20	0	1	1	90	1539	14
YUV to HEVC	720p	20	0	1	1	75	1561	15
YUV to AVC	720p	20	0	1	2	91	1576	15
YUV to HEVC	720p	20	0	1	2	100	960	10
YUV to AVC	720p	20	0	1	3	92	1581	14
YUV to HEVC	720p	20	0	1	3	100	620	9

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	20	4	0	1	100	1098	9
YUV to HEVC	720p	20	4	0	1	99	1104	9
YUV to AV1	720p	20	4	0	1	100	879	7
YUV to AVC	720p	20	4	0	2	99	1081	9
YUV to HEVC	720p	20	4	0	2	99	1080	13
YUV to AV1	720p	20	4	0	2	99	862	7
YUV to AVC	720p	20	4	0	3	97	1080	10
YUV to HEVC	720p	20	4	0	3	99	740	8
YUV to AV1	720p	20	4	0	3	99	560	14
YUV to AVC	720p	20	4	1	1	99	1066	12
YUV to HEVC	720p	20	4	1	1	100	1107	9
YUV to AVC	720p	20	4	1	2	99	1064	13
YUV to HEVC	720p	20	4	1	2	99	760	13
YUV to AVC	720p	20	4	1	3	99	1068	12
YUV to HEVC	720p	20	4	1	3	99	531	8
YUV to AVC	720p	20	20	0	1	99	1075	9
YUV to HEVC	720p	20	20	0	1	100	1103	10
YUV to AV1	720p	20	20	0	1	100	902	7
YUV to AVC	720p	20	20	0	2	98	1061	8
YUV to HEVC	720p	20	20	0	2	99	1076	12
YUV to AV1	720p	20	20	0	2	99	880	10
YUV to AVC	720p	20	20	0	3	97	1063	9
YUV to HEVC	720p	20	20	0	3	99	724	11
YUV to AV1	720p	20	20	0	3	99	560	10
YUV to AVC	720p	20	20	1	1	99	1060	13
YUV to HEVC	720p	20	20	1	1	99	1113	9
YUV to AVC	720p	20	20	1	2	99	1060	12
YUV to HEVC	720p	20	20	1	2	100	760	10
YUV to AVC	720p	20	20	1	3	99	1061	13
YUV to HEVC	720p	20	20	1	3	99	520	11
YUV to AVC	720p	20	40	0	1	99	1045	9
YUV to HEVC	720p	20	40	0	1	100	1081	10
YUV to AV1	720p	20	40	0	1	100	899	7
YUV to AVC	720p	20	40	0	2	99	1042	9

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	720p	20	40	0	2	99	1060	12
YUV to AV1	720p	20	40	0	2	99	875	9
YUV to AVC	720p	20	40	0	3	95	1041	9
YUV to HEVC	720p	20	40	0	3	99	720	10
YUV to AV1	720p	20	40	0	3	100	558	14
YUV to AVC	720p	20	40	1	1	99	1058	13
YUV to HEVC	720p	20	40	1	1	100	1088	9
YUV to AVC	720p	20	40	1	2	99	1057	13
YUV to HEVC	720p	20	40	1	2	99	760	15
YUV to AVC	720p	20	40	1	3	99	1059	12
YUV to HEVC	720p	20	40	1	3	100	520	12

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	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	0	0	1	19	99	681	21
YUV to HEVC	1080p	10	0	0	1	19	93	763	24
YUV to AV1	1080p	10	0	0	1	19	98	709	22
YUV to AVC	1080p	10	0	0	2	19	99	683	21
YUV to HEVC	1080p	10	0	0	2	19	99	512	20
YUV to AV1	1080p	10	0	0	2	19	99	431	18
YUV to AVC	1080p	10	0	0	3	19	100	682	21
YUV to HEVC	1080p	10	0	0	3	19	99	340	18
YUV to AV1	1080p	10	0	0	3	19	99	279	17
YUV to AVC	1080p	10	0	1	1	19	99	510	22
YUV to HEVC	1080p	10	0	1	1	19	99	658	23
YUV to AVC	1080p	10	0	1	2	19	100	510	20
YUV to HEVC	1080p	10	0	1	2	19	100	360	19
YUV to AVC	1080p	10	0	1	3	19	100	510	19
YUV to HEVC	1080p	10	0	1	3	19	100	245	17
YUV to AVC	1080p	10	4	0	1	19	99	641	19
YUV to HEVC	1080p	10	4	0	1	19	84	689	21
YUV to AV1	1080p	10	4	0	1	19	87	631	18
YUV to AVC	1080p	10	4	0	2	19	99	643	20
YUV to HEVC	1080p	10	4	0	2	19	99	510	20
YUV to AV1	1080p	10	4	0	2	19	99	430	20
YUV to AVC	1080p	10	4	0	3	19	99	642	19
YUV to HEVC	1080p	10	4	0	3	19	99	340	18
YUV to AV1	1080p	10	4	0	3	19	100	271	16
YUV to AVC	1080p	10	4	1	1	19	100	510	20
YUV to HEVC	1080p	10	4	1	1	19	99	658	21
YUV to AVC	1080p	10	4	1	2	19	99	510	18
YUV to HEVC	1080p	10	4	1	2	19	99	360	18
YUV to AVC	1080p	10	4	1	3	19	100	510	19
YUV to HEVC	1080p	10	4	1	3	19	100	246	17
YUV to AVC	1080p	10	16	0	1	19	92	599	18
YUV to HEVC	1080p	10	16	0	1	19	83	672	20
YUV to AV1	1080p	10	16	0	1	19	86	617	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	0	2	19	91	597	18
YUV to HEVC	1080p	10	16	0	2	19	100	510	20
YUV to AV1	1080p	10	16	0	2	19	99	428	18
YUV to AVC	1080p	10	16	0	3	19	94	599	17
YUV to HEVC	1080p	10	16	0	3	19	99	339	18
YUV to AV1	1080p	10	16	0	3	19	100	271	17
YUV to AVC	1080p	10	16	1	1	19	100	507	20
YUV to HEVC	1080p	10	16	1	1	19	100	651	20
YUV to AVC	1080p	10	16	1	2	19	99	509	18
YUV to HEVC	1080p	10	16	1	2	19	99	360	17
YUV to AVC	1080p	10	16	1	3	19	100	504	20
YUV to HEVC	1080p	10	16	1	3	19	99	243	18
YUV to AVC	1080p	10	40	0	1	19	92	582	18
YUV to HEVC	1080p	10	40	0	1	19	82	645	21
YUV to AV1	1080p	10	40	0	1	19	85	584	17
YUV to AVC	1080p	10	40	0	2	19	91	584	17
YUV to HEVC	1080p	10	40	0	2	19	99	507	19
YUV to AV1	1080p	10	40	0	2	19	100	420	17
YUV to AVC	1080p	10	40	0	3	19	93	584	17
YUV to HEVC	1080p	10	40	0	3	19	99	334	18
YUV to AV1	1080p	10	40	0	3	19	100	270	16
YUV to AVC	1080p	10	40	1	1	19	100	500	20
YUV to HEVC	1080p	10	40	1	1	19	100	645	21
YUV to AVC	1080p	10	40	1	2	19	99	501	19
YUV to HEVC	1080p	10	40	1	2	19	99	358	20
YUV to AVC	1080p	10	40	1	3	19	99	500	18
YUV to HEVC	1080p	10	40	1	3	19	100	240	16
YUV to AVC	1080p	10	0	0	1	23	99	682	20
YUV to HEVC	1080p	10	0	0	1	23	93	758	24
YUV to AV1	1080p	10	0	0	1	23	97	707	22
YUV to AVC	1080p	10	0	0	2	23	99	682	21
YUV to HEVC	1080p	10	0	0	2	23	99	510	19
YUV to AV1	1080p	10	0	0	2	23	100	430	18
YUV to AVC	1080p	10	0	0	3	23	99	682	21

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	23	100	340	18
YUV to AV1	1080p	10	0	0	3	23	100	278	17
YUV to AVC	1080p	10	0	1	1	23	100	510	20
YUV to HEVC	1080p	10	0	1	1	23	99	658	22
YUV to AVC	1080p	10	0	1	2	23	100	510	21
YUV to HEVC	1080p	10	0	1	2	23	99	360	18
YUV to AVC	1080p	10	0	1	3	23	99	510	20
YUV to HEVC	1080p	10	0	1	3	23	99	244	18
YUV to AVC	1080p	10	4	0	1	23	99	644	19
YUV to HEVC	1080p	10	4	0	1	23	86	699	21
YUV to AV1	1080p	10	4	0	1	23	88	641	19
YUV to AVC	1080p	10	4	0	2	23	99	644	19
YUV to HEVC	1080p	10	4	0	2	23	100	510	19
YUV to AV1	1080p	10	4	0	2	23	100	430	18
YUV to AVC	1080p	10	4	0	3	23	100	643	20
YUV to HEVC	1080p	10	4	0	3	23	100	340	18
YUV to AV1	1080p	10	4	0	3	23	100	270	17
YUV to AVC	1080p	10	4	1	1	23	100	510	22
YUV to HEVC	1080p	10	4	1	1	23	99	657	22
YUV to AVC	1080p	10	4	1	2	23	99	510	20
YUV to HEVC	1080p	10	4	1	2	23	99	360	18
YUV to AVC	1080p	10	4	1	3	23	99	510	20
YUV to HEVC	1080p	10	4	1	3	23	100	244	17
YUV to AVC	1080p	10	16	0	1	23	96	621	18
YUV to HEVC	1080p	10	16	0	1	23	86	686	21
YUV to AV1	1080p	10	16	0	1	23	87	629	19
YUV to AVC	1080p	10	16	0	2	23	94	618	20
YUV to HEVC	1080p	10	16	0	2	23	100	510	20
YUV to AV1	1080p	10	16	0	2	23	99	429	18
YUV to AVC	1080p	10	16	0	3	23	96	616	18
YUV to HEVC	1080p	10	16	0	3	23	100	340	17
YUV to AV1	1080p	10	16	0	3	23	99	270	17
YUV to AVC	1080p	10	16	1	1	23	100	507	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	507	19
YUV to HEVC	1080p	10	16	1	2	23	100	360	18
YUV to AVC	1080p	10	16	1	3	23	99	508	18
YUV to HEVC	1080p	10	16	1	3	23	100	243	16
YUV to AVC	1080p	10	40	0	1	23	94	604	19
YUV to HEVC	1080p	10	40	0	1	23	83	665	19
YUV to AV1	1080p	10	40	0	1	23	85	601	18
YUV to AVC	1080p	10	40	0	2	23	95	603	17
YUV to HEVC	1080p	10	40	0	2	23	99	508	20
YUV to AV1	1080p	10	40	0	2	23	99	420	18
YUV to AVC	1080p	10	40	0	3	23	93	601	18
YUV to HEVC	1080p	10	40	0	3	23	99	335	17
YUV to AV1	1080p	10	40	0	3	23	100	270	17
YUV to AVC	1080p	10	40	1	1	23	99	500	19
YUV to HEVC	1080p	10	40	1	1	23	99	644	21
YUV to AVC	1080p	10	40	1	2	23	99	500	19
YUV to HEVC	1080p	10	40	1	2	23	100	357	19
YUV to AVC	1080p	10	40	1	3	23	99	500	20
YUV to HEVC	1080p	10	40	1	3	23	100	241	18
YUV to AVC	1080p	10	0	0	1	27	99	682	23
YUV to HEVC	1080p	10	0	0	1	27	93	760	23
YUV to AV1	1080p	10	0	0	1	27	98	712	21
YUV to AVC	1080p	10	0	0	2	27	99	682	20
YUV to HEVC	1080p	10	0	0	2	27	99	511	20
YUV to AV1	1080p	10	0	0	2	27	99	430	18
YUV to AVC	1080p	10	0	0	3	27	99	682	21
YUV to HEVC	1080p	10	0	0	3	27	99	340	19
YUV to AV1	1080p	10	0	0	3	27	100	278	18
YUV to AVC	1080p	10	0	1	1	27	100	510	21
YUV to HEVC	1080p	10	0	1	1	27	99	659	21
YUV to AVC	1080p	10	0	1	2	27	100	510	20
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	245	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	4	0	1	27	99	646	19
YUV to HEVC	1080p	10	4	0	1	27	88	712	20
YUV to AV1	1080p	10	4	0	1	27	89	658	19
YUV to AVC	1080p	10	4	0	2	27	99	647	20
YUV to HEVC	1080p	10	4	0	2	27	99	511	20
YUV to AV1	1080p	10	4	0	2	27	99	430	18
YUV to AVC	1080p	10	4	0	3	27	99	645	20
YUV to HEVC	1080p	10	4	0	3	27	99	340	19
YUV to AV1	1080p	10	4	0	3	27	99	270	17
YUV to AVC	1080p	10	4	1	1	27	99	510	19
YUV to HEVC	1080p	10	4	1	1	27	99	656	22
YUV to AVC	1080p	10	4	1	2	27	99	510	19
YUV to HEVC	1080p	10	4	1	2	27	99	360	18
YUV to AVC	1080p	10	4	1	3	27	99	510	20
YUV to HEVC	1080p	10	4	1	3	27	100	245	17
YUV to AVC	1080p	10	16	0	1	27	99	639	18
YUV to HEVC	1080p	10	16	0	1	27	86	694	21
YUV to AV1	1080p	10	16	0	1	27	89	637	18
YUV to AVC	1080p	10	16	0	2	27	99	640	19
YUV to HEVC	1080p	10	16	0	2	27	99	510	20
YUV to AV1	1080p	10	16	0	2	27	99	428	18
YUV to AVC	1080p	10	16	0	3	27	99	640	19
YUV to HEVC	1080p	10	16	0	3	27	99	340	19
YUV to AV1	1080p	10	16	0	3	27	99	270	17
YUV to AVC	1080p	10	16	1	1	27	100	507	20
YUV to HEVC	1080p	10	16	1	1	27	99	650	20
YUV to AVC	1080p	10	16	1	2	27	99	506	19
YUV to HEVC	1080p	10	16	1	2	27	99	360	19
YUV to AVC	1080p	10	16	1	3	27	100	505	20
YUV to HEVC	1080p	10	16	1	3	27	100	243	17
YUV to AVC	1080p	10	40	0	1	27	99	624	19
YUV to HEVC	1080p	10	40	0	1	27	84	678	20
YUV to AV1	1080p	10	40	0	1	27	86	612	19
YUV to AVC	1080p	10	40	0	2	27	98	622	19

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	0	2	27	100	508	20
YUV to AV1	1080p	10	40	0	2	27	100	421	17
YUV to AVC	1080p	10	40	0	3	27	98	624	19
YUV to HEVC	1080p	10	40	0	3	27	100	335	18
YUV to AV1	1080p	10	40	0	3	27	99	270	16
YUV to AVC	1080p	10	40	1	1	27	99	500	20
YUV to HEVC	1080p	10	40	1	1	27	99	646	21
YUV to AVC	1080p	10	40	1	2	27	99	501	21
YUV to HEVC	1080p	10	40	1	2	27	100	357	19
YUV to AVC	1080p	10	40	1	3	27	99	500	18
YUV to HEVC	1080p	10	40	1	3	27	100	241	17

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