



Quadra™ Performance Test Report V5.9

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: 5906tFr1

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
- RES: Input video's resolution
- Load: Maximum load between FW Load and VPU Load during traffic
- DEC_LOAD: Maximum Load on the Decoder Engine
- ENC_LOAD: Maximum Load on the Encoder Engine
- SCALER_LOAD: Maximum Load on the 2D Engine (Scaler)

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	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	11	0	102	7
HEVC to YUV	8k	1	0	8	1	11	0	105	8
VP9 to YUV	8k	1	0	8	1	24	0	38	6
YUV to AVC	8k	1	0	8	1	0	96	67	54
YUV to HEVC	8k	1	0	8	1	0	98	83	73
AVC to AVC	8k	1	1	8	1	7	99	54	6
AVC to HEVC	8k	1	1	8	1	9	98	70	6
HEVC to AVC	8k	1	1	8	1	6	100	52	6
HEVC to HEVC	8k	1	1	8	1	7	100	70	6
VP9 to AVC	8k	1	1	8	1	24	44	35	2
VP9 to HEVC	8k	1	1	8	1	24	41	37	2
AVC to YUV	8k	1	0	10	1	11	0	60	6
HEVC to YUV	8k	1	0	10	1	10	0	56	7
VP9 to YUV	8k	1	0	10	1	25	0	33	14
YUV to AVC	8k	1	0	10	1	0	93	48	76
YUV to HEVC	8k	1	0	10	1	0	71	59	94
AVC to YUV	4k	1	0	8	1	18	0	313	16
HEVC to YUV	4k	1	0	8	1	17	0	313	18
VP9 to YUV	4k	1	0	8	1	23	0	155	6
AVC to YUV	4k	16	0	8	0	97	0	482	4
HEVC to YUV	4k	16	0	8	0	100	0	506	2
VP9 to YUV	4k	16	0	8	0	99	0	485	1
YUV to AVC	4k	1	0	8	1	0	94	294	43
YUV to HEVC	4k	1	0	8	1	0	95	326	51
YUV to AV1	4k	1	0	8	1	0	94	281	43
YUV to AVC	4k	4	0	8	0	0	96	304	18
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	10
YUV to HEVC	4k	8	0	8	0	0	100	344	9
YUV to AV1	4k	8	0	8	0	0	100	296	12
AVC to AVC	4k	1	1	8	1	14	94	221	13
AVC to HEVC	4k	1	1	8	1	15	91	274	15
AVC to AV1	4k	1	1	8	1	16	93	261	14
HEVC to AVC	4k	1	1	8	1	13	92	219	18
HEVC to HEVC	4k	1	1	8	1	16	92	277	13
HEVC to AV1	4k	1	1	8	1	14	92	260	15
VP9 to AVC	4k	1	1	8	1	24	46	144	3
VP9 to HEVC	4k	1	1	8	1	24	44	151	6

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	50	151	3
AVC to AVC	4k	4	1	8	0	61	95	236	3
AVC to HEVC	4k	4	1	8	0	68	96	300	3
AVC to AV1	4k	4	1	8	0	56	96	276	5
HEVC to AVC	4k	4	1	8	0	58	96	236	8
HEVC to HEVC	4k	4	1	8	0	60	96	296	3
HEVC to AV1	4k	4	1	8	0	46	96	276	8
VP9 to AVC	4k	4	1	8	0	65	97	243	3
VP9 to HEVC	4k	4	1	8	0	66	96	304	2
VP9 to AV1	4k	4	1	8	0	59	95	256	3
AVC to AVC	4k	8	1	8	0	68	99	211	2
AVC to HEVC	4k	8	1	8	0	76	99	276	2
AVC to AV1	4k	8	1	8	0	67	99	268	3
HEVC to AVC	4k	8	1	8	0	63	100	215	2
HEVC to HEVC	4k	8	1	8	0	68	100	280	2
HEVC to AV1	4k	8	1	8	0	57	100	273	2
VP9 to AVC	4k	8	1	8	0	67	99	230	2
VP9 to HEVC	4k	8	1	8	0	70	99	297	2
VP9 to AV1	4k	8	1	8	0	62	100	280	2
AVC to YUV	4k	1	0	10	1	16	0	225	7
HEVC to YUV	4k	1	0	10	1	15	0	226	9
VP9 to YUV	4k	1	0	10	1	23	0	126	7
AVC to YUV	4k	16	0	10	0	100	0	282	0
HEVC to YUV	4k	16	0	10	0	98	0	280	0
VP9 to YUV	4k	16	0	10	0	101	0	264	0
YUV to AVC	4k	1	0	10	1	0	68	198	58
YUV to HEVC	4k	1	0	10	1	0	60	206	60
YUV to AV1	4k	1	0	10	1	0	66	198	57
YUV to AVC	4k	4	0	10	0	0	91	219	27
YUV to HEVC	4k	4	0	10	0	0	74	249	37
YUV to AV1	4k	4	0	10	0	0	81	239	39
AVC to YUV	1080p	1	0	8	1	36	0	850	20
HEVC to YUV	1080p	1	0	8	1	30	0	827	28
VP9 to YUV	1080p	1	0	8	1	22	0	557	6
AVC to YUV	1080p	40	0	8	0	84	0	1722	1
HEVC to YUV	1080p	40	0	8	0	96	0	1816	1
VP9 to YUV	1080p	40	0	8	0	80	0	1772	0
YUV to AVC	1080p	1	0	8	1	0	53	685	28
YUV to HEVC	1080p	1	0	8	1	0	50	688	30
YUV to AV1	1080p	1	0	8	1	0	53	631	27

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1280	3
YUV to HEVC	1080p	32	0	8	0	0	100	1353	5
YUV to AV1	1080p	32	0	8	0	0	99	1182	3
AVC to AVC	1080p	1	1	8	1	41	79	954	24
AVC to HEVC	1080p	1	1	8	1	43	76	1012	22
AVC to AV1	1080p	1	1	8	1	39	79	897	20
HEVC to AVC	1080p	1	1	8	1	35	77	908	28
HEVC to HEVC	1080p	1	1	8	1	37	76	972	30
HEVC to AV1	1080p	1	1	8	1	32	75	850	28
VP9 to AVC	1080p	1	1	8	1	22	42	544	8
VP9 to HEVC	1080p	1	1	8	1	22	39	540	9
VP9 to AV1	1080p	1	1	8	1	22	46	540	7
AVC to AVC	1080p	32	1	8	0	75	100	938	2
AVC to HEVC	1080p	32	1	8	0	84	99	1066	1
AVC to AV1	1080p	32	1	8	0	77	100	1027	1
HEVC to AVC	1080p	32	1	8	0	68	99	992	1
HEVC to HEVC	1080p	32	1	8	0	76	100	1122	1
HEVC to AV1	1080p	32	1	8	0	68	99	1058	1
VP9 to AVC	1080p	32	1	8	0	70	100	1088	1
VP9 to HEVC	1080p	32	1	8	0	75	99	1216	1
VP9 to AV1	1080p	32	1	8	0	72	99	1120	1
AVC to YUV	1080p	1	0	10	1	30	0	643	9
HEVC to YUV	1080p	1	0	10	1	27	0	651	9
VP9 to YUV	1080p	1	0	10	1	22	0	456	6
AVC to YUV	1080p	40	0	10	0	62	0	1082	0
HEVC to YUV	1080p	40	0	10	0	70	0	1080	0
VP9 to YUV	1080p	40	0	10	0	70	0	1044	0
YUV to AVC	1080p	1	0	10	1	0	37	482	36
YUV to HEVC	1080p	1	0	10	1	0	34	480	40
YUV to AV1	1080p	1	0	10	1	0	37	449	33
YUV to AVC	1080p	32	0	10	0	0	61	791	5
YUV to HEVC	1080p	32	0	10	0	0	57	796	6
YUV to AV1	1080p	32	0	10	0	0	64	769	6
AVC to YUV	720p	1	0	8	1	43	0	1125	18
HEVC to YUV	720p	1	0	8	1	37	0	1131	25
VP9 to YUV	720p	1	0	8	1	33	0	1023	8
AVC to YUV	720p	100	0	8	0	100	0	2521	0
HEVC to YUV	720p	100	0	8	0	100	0	2911	1
VP9 to YUV	720p	100	0	8	0	99	0	2617	0
YUV to AVC	720p	1	0	8	1	0	30	860	18

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	30	863	18
YUV to AV1	720p	1	0	8	1	0	34	772	16
YUV to AVC	720p	64	0	8	0	0	97	2307	2
YUV to HEVC	720p	64	0	8	0	0	95	2347	2
YUV to AV1	720p	64	0	8	0	0	99	1984	1
AVC to AVC	720p	1	1	8	1	48	48	1255	18
AVC to HEVC	720p	1	1	8	1	47	47	1265	21
AVC to AV1	720p	1	1	8	1	40	51	1084	16
HEVC to AVC	720p	1	1	8	1	40	46	1216	24
HEVC to HEVC	720p	1	1	8	1	40	46	1221	24
HEVC to AV1	720p	1	1	8	1	34	49	1054	24
VP9 to AVC	720p	1	1	8	1	32	36	997	12
VP9 to HEVC	720p	1	1	8	1	32	35	993	14
VP9 to AV1	720p	1	1	8	1	32	45	994	12
AVC to AVC	720p	64	1	8	0	93	100	2007	0
AVC to HEVC	720p	64	1	8	0	95	100	2101	0
AVC to AV1	720p	64	1	8	0	76	100	1735	0
HEVC to AVC	720p	64	1	8	0	83	100	2037	0
HEVC to HEVC	720p	64	1	8	0	84	100	2112	0
HEVC to AV1	720p	64	1	8	0	67	100	1769	0
VP9 to AVC	720p	64	1	8	0	94	100	2240	0
VP9 to HEVC	720p	64	1	8	0	97	100	2304	0
VP9 to AV1	720p	64	1	8	0	76	100	1866	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	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	10	0	93	7
HEVC to YUV	8k	1	0	8	1	10	0	96	6
VP9 to YUV	8k	1	0	8	1	24	0	38	3
YUV to AVC	8k	1	0	8	1	0	96	67	29
YUV to HEVC	8k	1	0	8	1	0	94	81	33
AVC to AVC	8k	1	1	8	1	11	100	54	6
AVC to HEVC	8k	1	1	8	1	9	99	71	7
HEVC to AVC	8k	1	1	8	1	9	100	52	5
HEVC to HEVC	8k	1	1	8	1	11	100	70	5
VP9 to AVC	8k	1	1	8	1	24	46	35	4
VP9 to HEVC	8k	1	1	8	1	24	44	37	3
AVC to YUV	8k	1	0	10	1	10	0	54	7
HEVC to YUV	8k	1	0	10	1	9	0	51	6
VP9 to YUV	8k	1	0	10	1	24	0	33	4
YUV to AVC	8k	1	0	10	1	0	67	40	36
YUV to HEVC	8k	1	0	10	1	0	46	39	35
AVC to YUV	4k	1	0	8	1	18	0	311	8
HEVC to YUV	4k	1	0	8	1	17	0	319	6
VP9 to YUV	4k	1	0	8	1	24	0	154	4
AVC to YUV	4k	16	0	8	0	99	0	486	0
HEVC to YUV	4k	16	0	8	0	99	0	510	0
VP9 to YUV	4k	16	0	8	0	99	0	492	0
YUV to AVC	4k	1	0	8	1	0	89	279	25
YUV to HEVC	4k	1	0	8	1	0	83	287	27
YUV to AV1	4k	1	0	8	1	0	87	260	24
YUV to AVC	4k	4	0	8	0	0	96	312	11
YUV to HEVC	4k	4	0	8	0	0	97	334	11
YUV to AV1	4k	4	0	8	0	0	96	289	11
YUV to AVC	4k	8	0	8	0	0	99	325	7
YUV to HEVC	4k	8	0	8	0	0	100	347	7
YUV to AV1	4k	8	0	8	0	0	100	301	9
AVC to AVC	4k	1	1	8	1	17	97	225	8
AVC to HEVC	4k	1	1	8	1	17	96	277	9
AVC to AV1	4k	1	1	8	1	17	96	262	9
HEVC to AVC	4k	1	1	8	1	16	97	224	6
HEVC to HEVC	4k	1	1	8	1	16	96	285	7
HEVC to AV1	4k	1	1	8	1	17	97	268	7
VP9 to AVC	4k	1	1	8	1	24	46	149	4

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	1	24	43	149	4
VP9 to AV1	4k	1	1	8	1	24	50	149	4
AVC to AVC	4k	4	1	8	0	62	96	243	6
AVC to HEVC	4k	4	1	8	0	68	95	303	6
AVC to AV1	4k	4	1	8	0	57	95	275	5
HEVC to AVC	4k	4	1	8	0	57	97	239	5
HEVC to HEVC	4k	4	1	8	0	59	96	294	5
HEVC to AV1	4k	4	1	8	0	49	96	274	5
VP9 to AVC	4k	4	1	8	0	62	97	246	4
VP9 to HEVC	4k	4	1	8	0	67	96	303	4
VP9 to AV1	4k	4	1	8	0	54	96	277	4
AVC to AVC	4k	8	1	8	0	68	100	217	3
AVC to HEVC	4k	8	1	8	0	76	100	280	4
AVC to AV1	4k	8	1	8	0	67	100	273	4
HEVC to AVC	4k	8	1	8	0	63	99	219	3
HEVC to HEVC	4k	8	1	8	0	68	100	282	3
HEVC to AV1	4k	8	1	8	0	59	99	275	3
VP9 to AVC	4k	8	1	8	0	66	99	236	3
VP9 to HEVC	4k	8	1	8	0	71	100	303	3
VP9 to AV1	4k	8	1	8	0	61	100	288	3
AVC to YUV	4k	1	0	10	1	15	0	214	6
HEVC to YUV	4k	1	0	10	1	14	0	207	6
VP9 to YUV	4k	1	0	10	1	24	0	125	4
AVC to YUV	4k	16	0	10	0	98	0	284	0
HEVC to YUV	4k	16	0	10	0	100	0	284	0
VP9 to YUV	4k	16	0	10	0	101	0	267	0
YUV to AVC	4k	1	0	10	1	0	49	159	28
YUV to HEVC	4k	1	0	10	1	0	45	160	28
YUV to AV1	4k	1	0	10	1	0	51	152	27
YUV to AVC	4k	4	0	10	0	0	70	209	26
YUV to HEVC	4k	4	0	10	0	0	62	211	27
YUV to AV1	4k	4	0	10	0	0	63	186	29
AVC to YUV	1080p	1	0	8	1	35	0	800	13
HEVC to YUV	1080p	1	0	8	1	33	0	847	10
VP9 to YUV	1080p	1	0	8	1	22	0	556	5
AVC to YUV	1080p	40	0	8	0	80	0	1611	0
HEVC to YUV	1080p	40	0	8	0	85	0	1699	0
VP9 to YUV	1080p	40	0	8	0	72	0	1642	0
YUV to AVC	1080p	1	0	8	1	0	53	683	16

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	51	704	16
YUV to AV1	1080p	1	0	8	1	0	51	595	16
YUV to AVC	1080p	32	0	8	0	0	99	1325	2
YUV to HEVC	1080p	32	0	8	0	0	99	1399	3
YUV to AV1	1080p	32	0	8	0	0	100	1209	2
AVC to AVC	1080p	1	1	8	1	43	84	1003	20
AVC to HEVC	1080p	1	1	8	1	45	81	1058	21
AVC to AV1	1080p	1	1	8	1	41	81	926	20
HEVC to AVC	1080p	1	1	8	1	39	86	1005	13
HEVC to HEVC	1080p	1	1	8	1	41	85	1084	14
HEVC to AV1	1080p	1	1	8	1	37	83	933	14
VP9 to AVC	1080p	1	1	8	1	21	42	546	6
VP9 to HEVC	1080p	1	1	8	1	22	39	539	6
VP9 to AV1	1080p	1	1	8	1	21	46	540	7
AVC to AVC	1080p	32	1	8	0	74	99	977	1
AVC to HEVC	1080p	32	1	8	0	82	99	1105	1
AVC to AV1	1080p	32	1	8	0	75	99	1066	1
HEVC to AVC	1080p	32	1	8	0	69	99	1021	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1156	1
HEVC to AV1	1080p	32	1	8	0	67	99	1092	1
VP9 to AVC	1080p	32	1	8	0	72	99	1116	0
VP9 to HEVC	1080p	32	1	8	0	75	100	1261	0
VP9 to AV1	1080p	32	1	8	0	73	99	1157	1
AVC to YUV	1080p	1	0	10	1	29	0	639	6
HEVC to YUV	1080p	1	0	10	1	26	0	637	6
VP9 to YUV	1080p	1	0	10	1	22	0	456	5
AVC to YUV	1080p	40	0	10	0	67	0	1069	0
HEVC to YUV	1080p	40	0	10	0	70	0	1064	0
VP9 to YUV	1080p	40	0	10	0	70	0	1029	0
YUV to AVC	1080p	1	0	10	1	0	36	471	22
YUV to HEVC	1080p	1	0	10	1	0	34	480	23
YUV to AV1	1080p	1	0	10	1	0	34	403	21
YUV to AVC	1080p	32	0	10	0	0	63	834	6
YUV to HEVC	1080p	32	0	10	0	0	58	827	6
YUV to AV1	1080p	32	0	10	0	0	64	774	8
AVC to YUV	720p	1	0	8	1	41	0	1092	12
HEVC to YUV	720p	1	0	8	1	38	0	1155	8
VP9 to YUV	720p	1	0	8	1	33	0	1022	6
AVC to YUV	720p	100	0	8	0	100	0	2746	0

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	93	0	2896	0
VP9 to YUV	720p	100	0	8	0	93	0	2636	0
YUV to AVC	720p	1	0	8	1	0	33	977	12
YUV to HEVC	720p	1	0	8	1	0	33	990	11
YUV to AV1	720p	1	0	8	1	0	31	753	10
YUV to AVC	720p	64	0	8	0	0	100	2600	2
YUV to HEVC	720p	64	0	8	0	0	99	2663	2
YUV to AV1	720p	64	0	8	0	0	100	2149	1
AVC to AVC	720p	1	1	8	1	50	49	1313	18
AVC to HEVC	720p	1	1	8	1	49	48	1312	17
AVC to AV1	720p	1	1	8	1	42	49	1060	17
HEVC to AVC	720p	1	1	8	1	46	51	1368	13
HEVC to HEVC	720p	1	1	8	1	45	51	1366	13
HEVC to AV1	720p	1	1	8	1	38	49	1059	12
VP9 to AVC	720p	1	1	8	1	31	35	993	7
VP9 to HEVC	720p	1	1	8	1	31	34	990	7
VP9 to AV1	720p	1	1	8	1	31	45	990	9
AVC to AVC	720p	64	1	8	0	100	100	2163	0
AVC to HEVC	720p	64	1	8	0	100	100	2248	0
AVC to AV1	720p	64	1	8	0	93	100	1863	0
HEVC to AVC	720p	64	1	8	0	94	100	2164	0
HEVC to HEVC	720p	64	1	8	0	95	100	2253	0
HEVC to AV1	720p	64	1	8	0	89	100	1873	0
VP9 to AVC	720p	64	1	8	0	100	100	2402	0
VP9 to HEVC	720p	64	1	8	0	100	100	2467	0
VP9 to AV1	720p	64	1	8	0	98	100	2001	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	57.89	59.50	57.45	0.05
YUV to HEVC	8k	1	55.49	60.43	54.05	0.91
YUV to AVC	4k	1	15.46	17.80	15.10	0.08
YUV to HEVC	4k	1	16.64	18.44	15.66	0.09
YUV to AV1	4k	1	21.84	25.51	16.20	0.71
YUV to AVC	4k	4	16.62	21.49	14.98	1.98
YUV to HEVC	4k	4	18.32	22.18	15.63	1.84
YUV to AV1	4k	4	23.04	27.62	16.41	1.12
YUV to AVC	4k	8	20.23	33.86	15.05	20.77
YUV to HEVC	4k	8	22.66	34.59	15.82	20.66
YUV to AV1	4k	8	37.24	46.03	23.82	13.05
YUV to AVC	1080p	1	4.65	5.53	4.47	0.02
YUV to HEVC	1080p	1	4.98	5.60	4.75	0.02
YUV to AV1	1080p	1	6.73	7.49	4.92	0.06
YUV to AVC	1080p	32	6.11	9.83	4.75	0.33
YUV to HEVC	1080p	32	7.09	12.09	5.09	1.37
YUV to AV1	1080p	32	41.02	46.40	33.65	1.97
YUV to AVC	720p	1	2.84	3.51	2.48	0.01
YUV to HEVC	720p	1	2.89	3.41	2.75	0.01
YUV to AV1	720p	1	3.83	4.17	3.23	0.01
YUV to AVC	720p	64	6.16	11.54	3.41	0.98
YUV to HEVC	720p	64	6.40	10.83	3.56	1.04
YUV to AV1	720p	64	39.77	47.41	31.38	3.47

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	23	0	40	2
AVC to RGBA	4k	1	22	0	145	8
AVC to RGBA	4k	16	92	3	586	2
HEVC to RGBA	4k	1	22	0	174	10
HEVC to RGBA	4k	16	94	4	669	2
VP9 to RGBA	4k	1	22	0	166	4
VP9 to RGBA	4k	16	93	4	680	1
AVC to RGBA	1080p	40	94	14	1897	1
HEVC to RGBA	1080p	40	93	14	2017	1
VP9 to RGBA	1080p	40	93	17	2440	0
AVC to RGBA	720p	100	96	18	2600	0
HEVC to RGBA	720p	100	90	20	2833	0
VP9 to RGBA	720p	64	96	19	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	93	2	432	4
AVC to HEVC	8	32	93	2	465	4
AVC to AV1	8	22	94	1	384	3
HEVC to AVC	8	34	92	2	440	5
HEVC to HEVC	8	32	92	2	472	5
HEVC to AV1	8	23	93	2	392	4
VP9 to AVC	8	40	95	2	432	4
VP9 to HEVC	8	39	95	2	464	4
VP9 to AV1	8	29	95	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	64
RGBA to AV1	4k	1	1	55	164	70
RGBA to AVC	4k	4	0	69	165	41
RGBA to HEVC	4k	4	0	61	168	40
RGBA to AV1	4k	4	0	61	172	45
RGBA to AVC	4k	8	0	59	176	31
RGBA to HEVC	4k	8	0	57	176	31
RGBA to AV1	4k	8	0	61	176	32
RGBA to AVC	1080p	1	1	33	433	45
RGBA to HEVC	1080p	1	1	31	427	50
RGBA to AV1	1080p	1	1	34	410	45
RGBA to AVC	1080p	16	0	57	641	15
RGBA to HEVC	1080p	16	0	53	644	15
RGBA to AV1	1080p	16	0	55	636	16
RGBA to AVC	1080p	32	0	55	667	9
RGBA to HEVC	1080p	32	0	51	666	9
RGBA to AV1	1080p	32	0	57	647	8
RGBA to AVC	720p	1	1	29	761	62
RGBA to HEVC	720p	1	1	29	796	49
RGBA to AV1	720p	1	1	34	726	63
RGBA to AVC	720p	16	0	47	1178	32
RGBA to HEVC	720p	16	0	46	1177	32
RGBA to AV1	720p	16	0	53	1102	29
RGBA to AVC	720p	32	0	46	1153	32
RGBA to HEVC	720p	32	0	45	1157	30
RGBA to AV1	720p	32	0	56	1158	29

7. T1A – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

7.1 Encoding

7.1.1 Description

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

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

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

7.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

7.2 Encoding EnableRdoQuant/rdoLevel/lookaheadDepth Performance Results

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	0	0	1	93	300	27
YUV to HEVC	4k	4	0	0	1	97	332	15
YUV to AV1	4k	4	0	0	1	96	288	13
YUV to AVC	4k	4	0	0	2	93	296	28
YUV to HEVC	4k	4	0	0	2	96	168	14
YUV to AV1	4k	4	0	0	2	98	140	7
YUV to AVC	4k	4	0	0	3	96	302	22
YUV to HEVC	4k	4	0	0	3	100	100	7
YUV to AV1	4k	4	0	0	3	97	76	4
YUV to AVC	4k	4	0	1	1	99	188	8
YUV to HEVC	4k	4	0	1	1	97	240	10
YUV to AVC	4k	4	0	1	2	97	188	7
YUV to HEVC	4k	4	0	1	2	97	108	6
YUV to AVC	4k	4	0	1	3	97	188	9
YUV to HEVC	4k	4	0	1	3	98	68	4
YUV to AVC	4k	4	4	0	1	99	196	9
YUV to HEVC	4k	4	4	0	1	100	228	11
YUV to AV1	4k	4	4	0	1	100	200	10
YUV to AVC	4k	4	4	0	2	99	196	9
YUV to HEVC	4k	4	4	0	2	100	136	6
YUV to AV1	4k	4	4	0	2	99	116	6
YUV to AVC	4k	4	4	0	3	99	197	9
YUV to HEVC	4k	4	4	0	3	99	88	5
YUV to AV1	4k	4	4	0	3	100	72	5
YUV to AVC	4k	4	4	1	1	99	140	7
YUV to HEVC	4k	4	4	1	1	99	180	9
YUV to AVC	4k	4	4	1	2	100	140	6
YUV to HEVC	4k	4	4	1	2	99	96	5
YUV to AVC	4k	4	4	1	3	100	140	7
YUV to HEVC	4k	4	4	1	3	99	64	3
YUV to AVC	4k	4	16	0	1	99	196	10
YUV to HEVC	4k	4	16	0	1	99	228	11
YUV to AV1	4k	4	16	0	1	100	198	15
YUV to AVC	4k	4	16	0	2	99	196	10
YUV to HEVC	4k	4	16	0	2	99	136	8
YUV to AV1	4k	4	16	0	2	99	116	6
YUV to AVC	4k	4	16	0	3	100	196	9
YUV to HEVC	4k	4	16	0	3	99	88	5

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	4k	4	16	0	3	102	72	4
YUV to AVC	4k	4	16	1	1	100	140	7
YUV to HEVC	4k	4	16	1	1	100	180	12
YUV to AVC	4k	4	16	1	2	100	140	6
YUV to HEVC	4k	4	16	1	2	100	95	5
YUV to AVC	4k	4	16	1	3	100	140	7
YUV to HEVC	4k	4	16	1	3	101	64	4
YUV to AVC	4k	4	40	0	1	100	196	14
YUV to HEVC	4k	4	40	0	1	100	224	11
YUV to AV1	4k	4	40	0	1	100	196	10
YUV to AVC	4k	4	40	0	2	100	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	100	196	9
YUV to HEVC	4k	4	40	0	3	100	88	4
YUV to AV1	4k	4	40	0	3	98	72	4
YUV to AVC	4k	4	40	1	1	100	136	7
YUV to HEVC	4k	4	40	1	1	100	176	9
YUV to AVC	4k	4	40	1	2	100	136	7
YUV to HEVC	4k	4	40	1	2	98	92	6
YUV to AVC	4k	4	40	1	3	99	136	6
YUV to HEVC	4k	4	40	1	3	100	64	4
YUV to AVC	1080p	20	0	0	1	99	1280	4
YUV to HEVC	1080p	20	0	0	1	99	1360	7
YUV to AV1	1080p	20	0	0	1	99	1180	4
YUV to AVC	1080p	20	0	0	2	99	1280	4
YUV to HEVC	1080p	20	0	0	2	100	682	2
YUV to AV1	1080p	20	0	0	2	99	560	2
YUV to AVC	1080p	20	0	0	3	99	1280	5
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	99	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	100	440	2
YUV to AVC	1080p	20	0	1	3	100	760	2
YUV to HEVC	1080p	20	0	1	3	100	280	1
YUV to AVC	1080p	20	4	0	1	99	700	2
YUV to HEVC	1080p	20	4	0	1	99	820	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	100	720	2
YUV to AVC	1080p	20	4	0	2	99	700	2
YUV to HEVC	1080p	20	4	0	2	100	518	2
YUV to AV1	1080p	20	4	0	2	99	420	2
YUV to AVC	1080p	20	4	0	3	99	700	2
YUV to HEVC	1080p	20	4	0	3	100	340	2
YUV to AV1	1080p	20	4	0	3	100	268	1
YUV to AVC	1080p	20	4	1	1	100	501	2
YUV to HEVC	1080p	20	4	1	1	99	660	2
YUV to AVC	1080p	20	4	1	2	100	501	2
YUV to HEVC	1080p	20	4	1	2	100	360	1
YUV to AVC	1080p	20	4	1	3	99	500	2
YUV to HEVC	1080p	20	4	1	3	99	240	1
YUV to AVC	1080p	20	16	0	1	99	700	2
YUV to HEVC	1080p	20	16	0	1	99	820	3
YUV to AV1	1080p	20	16	0	1	99	720	3
YUV to AVC	1080p	20	16	0	2	99	700	2
YUV to HEVC	1080p	20	16	0	2	99	500	2
YUV to AV1	1080p	20	16	0	2	99	420	1
YUV to AVC	1080p	20	16	0	3	99	700	2
YUV to HEVC	1080p	20	16	0	3	100	340	1
YUV to AV1	1080p	20	16	0	3	99	262	1
YUV to AVC	1080p	20	16	1	1	100	500	2
YUV to HEVC	1080p	20	16	1	1	99	645	2
YUV to AVC	1080p	20	16	1	2	99	500	2
YUV to HEVC	1080p	20	16	1	2	100	360	1
YUV to AVC	1080p	20	16	1	3	100	500	2
YUV to HEVC	1080p	20	16	1	3	99	240	1
YUV to AVC	1080p	20	40	0	1	99	691	3
YUV to HEVC	1080p	20	40	0	1	99	802	3
YUV to AV1	1080p	20	40	0	1	100	702	2
YUV to AVC	1080p	20	40	0	2	100	691	2
YUV to HEVC	1080p	20	40	0	2	100	500	2
YUV to AV1	1080p	20	40	0	2	100	420	1
YUV to AVC	1080p	20	40	0	3	99	686	2
YUV to HEVC	1080p	20	40	0	3	99	340	1
YUV to AV1	1080p	20	40	0	3	99	260	1
YUV to AVC	1080p	20	40	1	1	99	500	2
YUV to HEVC	1080p	20	40	1	1	99	640	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	2	100	500	2
YUV to HEVC	1080p	20	40	1	2	100	360	1
YUV to AVC	1080p	20	40	1	3	99	500	2
YUV to HEVC	1080p	20	40	1	3	99	240	1
YUV to AVC	720p	40	0	0	1	93	2273	2
YUV to HEVC	720p	40	0	0	1	89	2283	3
YUV to AV1	720p	40	0	0	1	97	2002	2
YUV to AVC	720p	40	0	0	2	93	2269	3
YUV to HEVC	720p	40	0	0	2	99	1520	1
YUV to AV1	720p	40	0	0	2	100	1205	1
YUV to AVC	720p	40	0	0	3	93	2257	2
YUV to HEVC	720p	40	0	0	3	99	920	1
YUV to AV1	720p	40	0	0	3	99	654	1
YUV to AVC	720p	40	0	1	1	99	1695	1
YUV to HEVC	720p	40	0	1	1	99	2080	2
YUV to AVC	720p	40	0	1	2	99	1686	1
YUV to HEVC	720p	40	0	1	2	99	960	1
YUV to AVC	720p	40	0	1	3	99	1682	1
YUV to HEVC	720p	40	0	1	3	100	607	1
YUV to AVC	720p	40	4	0	1	100	1401	1
YUV to HEVC	720p	40	4	0	1	100	1323	1
YUV to AV1	720p	40	4	0	1	100	1000	1
YUV to AVC	720p	40	4	0	2	100	1406	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	1408	1
YUV to HEVC	720p	40	4	0	3	99	720	1
YUV to AV1	720p	40	4	0	3	99	560	1
YUV to AVC	720p	40	4	1	1	99	1079	1
YUV to HEVC	720p	40	4	1	1	100	1320	1
YUV to AVC	720p	40	4	1	2	99	1073	1
YUV to HEVC	720p	40	4	1	2	99	760	1
YUV to AVC	720p	40	4	1	3	99	1076	1
YUV to HEVC	720p	40	4	1	3	100	520	1
YUV to AVC	720p	40	16	0	1	100	1377	1
YUV to HEVC	720p	40	16	0	1	100	1325	1
YUV to AV1	720p	40	16	0	1	100	1043	1
YUV to AVC	720p	40	16	0	2	100	1368	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	1372	1
YUV to HEVC	720p	40	16	0	3	100	720	1
YUV to AV1	720p	40	16	0	3	99	560	1
YUV to AVC	720p	40	16	1	1	99	1048	1
YUV to HEVC	720p	40	16	1	1	100	1322	1
YUV to AVC	720p	40	16	1	2	99	1049	1
YUV to HEVC	720p	40	16	1	2	99	760	1
YUV to AVC	720p	40	16	1	3	99	1051	1
YUV to HEVC	720p	40	16	1	3	100	520	0
YUV to AVC	720p	40	40	0	1	100	1328	1
YUV to HEVC	720p	40	40	0	1	100	1288	1
YUV to AV1	720p	40	40	0	1	100	1040	1
YUV to AVC	720p	40	40	0	2	100	1327	1
YUV to HEVC	720p	40	40	0	2	99	1040	1
YUV to AV1	720p	40	40	0	2	99	845	1
YUV to AVC	720p	40	40	0	3	100	1322	1
YUV to HEVC	720p	40	40	0	3	99	720	1
YUV to AV1	720p	40	40	0	3	99	558	1
YUV to AVC	720p	40	40	1	1	99	1040	1
YUV to HEVC	720p	40	40	1	1	100	1280	1
YUV to AVC	720p	40	40	1	2	99	1040	1
YUV to HEVC	720p	40	40	1	2	99	758	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	99	701	2
YUV to HEVC	1080p	20	0	0	1	19	99	820	3
YUV to AV1	1080p	20	0	0	1	19	99	720	2
YUV to AVC	1080p	20	0	0	2	19	99	700	2
YUV to HEVC	1080p	20	0	0	2	19	99	520	2
YUV to AV1	1080p	20	0	0	2	19	99	440	2
YUV to AVC	1080p	20	0	0	3	19	100	701	2
YUV to HEVC	1080p	20	0	0	3	19	100	340	2
YUV to AV1	1080p	20	0	0	3	19	100	280	1
YUV to AVC	1080p	20	0	1	1	19	99	500	2
YUV to HEVC	1080p	20	0	1	1	19	100	660	2
YUV to AVC	1080p	20	0	1	2	19	99	501	2
YUV to HEVC	1080p	20	0	1	2	19	99	360	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	99	720	2
YUV to AVC	1080p	20	4	0	2	19	99	700	2
YUV to HEVC	1080p	20	4	0	2	19	100	520	2
YUV to AV1	1080p	20	4	0	2	19	99	420	2
YUV to AVC	1080p	20	4	0	3	19	99	700	2
YUV to HEVC	1080p	20	4	0	3	19	99	340	2
YUV to AV1	1080p	20	4	0	3	19	100	269	1
YUV to AVC	1080p	20	4	1	1	19	99	500	2
YUV to HEVC	1080p	20	4	1	1	19	99	660	2
YUV to AVC	1080p	20	4	1	2	19	100	500	2
YUV to HEVC	1080p	20	4	1	2	19	99	360	1
YUV to AVC	1080p	20	4	1	3	19	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	819	3
YUV to AV1	1080p	20	16	0	1	19	99	720	3
YUV to AVC	1080p	20	16	0	2	19	99	700	2
YUV to HEVC	1080p	20	16	0	2	19	100	500	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	100	340	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	3	19	100	263	1
YUV to AVC	1080p	20	16	1	1	19	100	500	2
YUV to HEVC	1080p	20	16	1	1	19	99	647	2
YUV to AVC	1080p	20	16	1	2	19	99	500	2
YUV to HEVC	1080p	20	16	1	2	19	100	360	1
YUV to AVC	1080p	20	16	1	3	19	100	500	2
YUV to HEVC	1080p	20	16	1	3	19	100	240	1
YUV to AVC	1080p	20	40	0	1	19	99	685	3
YUV to HEVC	1080p	20	40	0	1	19	99	801	3
YUV to AV1	1080p	20	40	0	1	19	99	701	3
YUV to AVC	1080p	20	40	0	2	19	99	688	3
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	691	3
YUV to HEVC	1080p	20	40	0	3	19	100	340	1
YUV to AV1	1080p	20	40	0	3	19	100	260	1
YUV to AVC	1080p	20	40	1	1	19	99	500	2
YUV to HEVC	1080p	20	40	1	1	19	99	640	2
YUV to AVC	1080p	20	40	1	2	19	100	500	2
YUV to HEVC	1080p	20	40	1	2	19	99	360	1
YUV to AVC	1080p	20	40	1	3	19	100	500	2
YUV to HEVC	1080p	20	40	1	3	19	100	240	1
YUV to AVC	1080p	20	0	0	1	23	99	700	2
YUV to HEVC	1080p	20	0	0	1	23	99	820	3
YUV to AV1	1080p	20	0	0	1	23	99	720	2
YUV to AVC	1080p	20	0	0	2	23	99	700	3
YUV to HEVC	1080p	20	0	0	2	23	99	519	2
YUV to AV1	1080p	20	0	0	2	23	100	439	1
YUV to AVC	1080p	20	0	0	3	23	99	700	2
YUV to HEVC	1080p	20	0	0	3	23	100	340	1
YUV to AV1	1080p	20	0	0	3	23	99	280	1
YUV to AVC	1080p	20	0	1	1	23	100	502	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	503	2
YUV to HEVC	1080p	20	0	1	3	23	100	240	1
YUV to AVC	1080p	20	4	0	1	23	99	700	2
YUV to HEVC	1080p	20	4	0	1	23	99	820	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	23	99	720	2
YUV to AVC	1080p	20	4	0	2	23	99	701	2
YUV to HEVC	1080p	20	4	0	2	23	100	520	2
YUV to AV1	1080p	20	4	0	2	23	99	420	2
YUV to AVC	1080p	20	4	0	3	23	99	700	3
YUV to HEVC	1080p	20	4	0	3	23	99	340	1
YUV to AV1	1080p	20	4	0	3	23	100	267	1
YUV to AVC	1080p	20	4	1	1	23	99	502	2
YUV to HEVC	1080p	20	4	1	1	23	99	660	2
YUV to AVC	1080p	20	4	1	2	23	99	500	2
YUV to HEVC	1080p	20	4	1	2	23	99	360	1
YUV to AVC	1080p	20	4	1	3	23	99	500	2
YUV to HEVC	1080p	20	4	1	3	23	100	240	1
YUV to AVC	1080p	20	16	0	1	23	99	699	2
YUV to HEVC	1080p	20	16	0	1	23	99	820	3
YUV to AV1	1080p	20	16	0	1	23	99	719	3
YUV to AVC	1080p	20	16	0	2	23	99	700	3
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	100	700	3
YUV to HEVC	1080p	20	16	0	3	23	99	340	1
YUV to AV1	1080p	20	16	0	3	23	100	266	1
YUV to AVC	1080p	20	16	1	1	23	99	500	2
YUV to HEVC	1080p	20	16	1	1	23	100	646	2
YUV to AVC	1080p	20	16	1	2	23	99	500	2
YUV to HEVC	1080p	20	16	1	2	23	99	360	1
YUV to AVC	1080p	20	16	1	3	23	99	500	2
YUV to HEVC	1080p	20	16	1	3	23	99	240	1
YUV to AVC	1080p	20	40	0	1	23	99	693	3
YUV to HEVC	1080p	20	40	0	1	23	99	800	3
YUV to AV1	1080p	20	40	0	1	23	99	701	3
YUV to AVC	1080p	20	40	0	2	23	99	685	3
YUV to HEVC	1080p	20	40	0	2	23	99	500	2
YUV to AV1	1080p	20	40	0	2	23	100	420	2
YUV to AVC	1080p	20	40	0	3	23	100	691	3
YUV to HEVC	1080p	20	40	0	3	23	100	339	1
YUV to AV1	1080p	20	40	0	3	23	99	260	1
YUV to AVC	1080p	20	40	1	1	23	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	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	100	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	100	700	2
YUV to HEVC	1080p	20	0	0	2	27	99	520	2
YUV to AV1	1080p	20	0	0	2	27	100	439	1
YUV to AVC	1080p	20	0	0	3	27	99	701	3
YUV to HEVC	1080p	20	0	0	3	27	100	340	1
YUV to AV1	1080p	20	0	0	3	27	100	280	1
YUV to AVC	1080p	20	0	1	1	27	99	502	2
YUV to HEVC	1080p	20	0	1	1	27	100	660	2
YUV to AVC	1080p	20	0	1	2	27	100	502	2
YUV to HEVC	1080p	20	0	1	2	27	100	360	1
YUV to AVC	1080p	20	0	1	3	27	99	502	2
YUV to HEVC	1080p	20	0	1	3	27	99	240	1
YUV to AVC	1080p	20	4	0	1	27	100	700	2
YUV to HEVC	1080p	20	4	0	1	27	99	820	3
YUV to AV1	1080p	20	4	0	1	27	99	720	2
YUV to AVC	1080p	20	4	0	2	27	99	700	2
YUV to HEVC	1080p	20	4	0	2	27	100	520	2
YUV to AV1	1080p	20	4	0	2	27	99	420	2
YUV to AVC	1080p	20	4	0	3	27	99	700	2
YUV to HEVC	1080p	20	4	0	3	27	100	340	1
YUV to AV1	1080p	20	4	0	3	27	99	266	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	99	500	2
YUV to HEVC	1080p	20	4	1	2	27	99	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	100	720	2
YUV to AVC	1080p	20	16	0	2	27	99	700	2
YUV to HEVC	1080p	20	16	0	2	27	99	506	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	2	27	99	420	2
YUV to AVC	1080p	20	16	0	3	27	99	700	2
YUV to HEVC	1080p	20	16	0	3	27	100	340	1
YUV to AV1	1080p	20	16	0	3	27	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	645	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	99	500	2
YUV to HEVC	1080p	20	16	1	3	27	99	240	1
YUV to AVC	1080p	20	40	0	1	27	99	685	2
YUV to HEVC	1080p	20	40	0	1	27	99	801	3
YUV to AV1	1080p	20	40	0	1	27	100	702	3
YUV to AVC	1080p	20	40	0	2	27	99	690	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	692	3
YUV to HEVC	1080p	20	40	0	3	27	99	339	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	100	640	2
YUV to AVC	1080p	20	40	1	2	27	100	500	2
YUV to HEVC	1080p	20	40	1	2	27	99	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	21	5	289	11
AVC to HEVC	1	14	20	5	308	12
AVC to AV1	1	13	20	4	262	11
HEVC to AVC	1	13	21	5	289	14
HEVC to HEVC	1	13	21	5	307	15
HEVC to AV1	1	12	20	4	262	14
VP9 to AVC	1	17	20	5	288	11
VP9 to HEVC	1	18	21	5	306	11
VP9 to AV1	1	15	20	4	262	11
AVC to AVC	16	71	93	24	1046	3
AVC to HEVC	16	79	94	28	1159	3
AVC to AV1	16	70	92	26	1111	3
HEVC to AVC	16	71	94	25	1088	4
HEVC to HEVC	16	78	95	28	1216	4
HEVC to AV1	16	67	93	26	1136	4
VP9 to AVC	16	83	95	24	1040	3
VP9 to HEVC	16	89	92	27	1168	3
VP9 to AV1	16	83	94	25	1120	3
AVC to AVC	32	75	94	24	960	1
AVC to HEVC	32	82	95	27	1081	1
AVC to AV1	32	76	93	26	1056	1
HEVC to AVC	32	73	94	25	993	1
HEVC to HEVC	32	80	95	29	1140	2
HEVC to AV1	32	72	93	26	1088	2
VP9 to AVC	32	82	95	24	992	1
VP9 to HEVC	32	88	93	26	1120	1
VP9 to AV1	32	84	95	26	1088	1

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

10.1 Encoding

10.1.1 Description

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

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

The YUV frame is encoded with hardware encoder.

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

10.1.2 Command line

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

<num_jobs> = number of instances running concurrently

10.2 Multi Thread P2P DMA on AMD GPU Performance Results

TYPE	RES	JOB	Frames	FPS	CPU	Enc Load	P2P MEM	Latency Avg (ms)	Latency Dev (%)
P2A	720p	1	1001	29	1	0	0.5	2.07	0.01
P2H	720p	1	1001	29	1	0.25	0.5	2.05	0.01
P2A	720p	180	180180	29.7	0	46	56	2.34	0.12
P2H	720p	180	180180	29.6	0	45	56	2.34	0.14
P2A	1080p	1	1001	29	1	0.5	0.75	3.77	0.01
P2H	1080p	1	1001	29	1	0.5	0.75	3.58	0.01
P2A	1080p	80	80080	29.4	0	46	50	4.01	0.42
P2H	1080p	80	80080	30	0	43	50	3.74	0.18

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	77.59
yolov5s_320	8	10000	rgb	nchw	image	320x320x3	278.22
deeplabv3_FRP	8	10000	rgb	nchw	image	257x257x3	359.41
resnet50	8	10000	rgb	nchw	image	224x224x3	228.03
mobilenetv2	8	10000	rgb	nchw	image	224x224x3	1110.53
deeplabv3	8	10000	rgb	nchw	image	257x257x3	160.33
yolov4	8	10000	bgr	nchw	image	416x416x3	257.43
fsrcnn	8	2000	bgr	nchw	image	360x640x1	31.33
BiSeNetv1	8	10000	rgb	nchw	image	512x512x3	76.10
HrNet	8	10000	rgb	nchw	image	256x192x3	73.71
usm_1656x1920	8	10000	rgb	nchw	image	1920x1656x1	250.67
usm_3240x3840	8	10000	rgb	nchw	image	3840x3240x1	61.03
lpips	8	2000	rgb	nchw	image	720x480x3	1.41
PaddleOCR-512_onnx	8	10000	rgb	nchw	image	512x48x3	16.05
segm32_tflite_kl_mle	8	10000	rgb	nchw	image	256x144x3	835.57
mobilenetv2_nchw_keras_96x160	1	10000	rgb	nchw	image	96x160x3	2343.50
mobilenetv2_nchw_keras_96x160	8	10000	rgb	nchw	image	96x160x3	2337.16
mobilenetv2_nchw_keras_96x160	16	10000	rgb	nchw	image	96x160x3	2335.12

11.2 AI Encoding with 2D Engine

11.2.1 Description BG Filter

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

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

11.2.2 Description ROI Filter

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

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

11.2.3 Description PRE Filter

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

11.2.4 Command line BG

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

11.2.5 Command line ROI

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

11.2.6 Command line PRE

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

11.2.7 AI Encoding with 2D Engine Performance Results

Filter	Model	Resolution	Session Number	Average FPS per session
ROI	network_binary_yolov4_head	1920x1080	1	79
ROI	network_binary_yolov4_head	1920x1080	8	22
ROI	network_binary_yolov4_head	1920x1080	32	5
BG	segm32_tflite_nchw_bgr	1920x1080	1	72
BG	segm32_tflite_nchw_bgr	1920x1080	8	46
BG	segm32_tflite_nchw_bgr	1920x1080	32	15
PRE	hw_lanczos_network_binary	1920x1080	8	66
PRE	hw_lanczos_network_binary	1920x1080	16	33
PRE	hw_bicubic_network_binary	1920x1080	8	66
PRE	hw_bicubic_network_binary	1920x1080	16	33

12. T1A – GStreamer Throughput

12.1 Decoding

12.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

12.1.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

12.2 Encoding

12.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

12.2.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

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

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

12.3 Transcoding

12.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

12.3.2 Command line

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

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

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

<num_jobs> is the number of instances running concurrently

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

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

12.4 GStreamer Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	5	0	44	67
HEVC to YUV	8k	1	0	8	1	5	0	44	65
YUV to AVC	8k	1	0	8	1	0	45	29	105
YUV to HEVC	8k	1	0	8	1	0	30	29	105
AVC to AVC	8k	1	1	8	1	3	91	54	7
AVC to HEVC	8k	1	1	8	1	3	96	71	7
HEVC to AVC	8k	1	1	8	1	2	96	52	6
HEVC to HEVC	8k	1	1	8	1	3	98	69	6
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	17	14	105
YUV to HEVC	8k	1	0	10	1	0	21	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	99	0	484	29
HEVC to YUV	4k	16	0	8	0	96	0	503	34
YUV to AVC	4k	1	0	8	1	0	44	133	73
YUV to HEVC	4k	1	0	8	1	0	41	138	76
YUV to AV1	4k	1	0	8	1	0	45	129	71
YUV to AVC	4k	4	0	8	0	0	80	249	42
YUV to HEVC	4k	4	0	8	0	0	76	261	46
YUV to AV1	4k	4	0	8	0	0	81	235	40
YUV to AVC	4k	8	0	8	0	0	99	308	32
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	9	93	224	7
AVC to HEVC	4k	1	1	8	1	12	92	280	8
AVC to AV1	4k	1	1	8	1	11	94	264	8
HEVC to AVC	4k	1	1	8	1	9	91	223	7
HEVC to HEVC	4k	1	1	8	1	10	95	282	7
HEVC to AV1	4k	1	1	8	1	9	96	263	7
AVC to AVC	4k	4	1	8	0	60	96	240	3
AVC to HEVC	4k	4	1	8	0	68	96	297	3
AVC to AV1	4k	4	1	8	0	55	95	274	3
HEVC to AVC	4k	4	1	8	0	60	96	234	3
HEVC to HEVC	4k	4	1	8	0	60	96	295	3
HEVC to AV1	4k	4	1	8	0	46	93	274	3
AVC to AVC	4k	8	1	8	0	69	99	216	2

AVC to HEVC	4k	8	1	8	0	75	99	280	2
AVC to AV1	4k	8	1	8	0	67	100	270	3
HEVC to AVC	4k	8	1	8	0	61	99	218	2
HEVC to HEVC	4k	8	1	8	0	67	99	282	2
HEVC to AV1	4k	8	1	8	0	58	99	277	2
AVC to YUV	4k	1	0	10	1	8	0	108	59
HEVC to YUV	4k	1	0	10	1	8	0	108	60
AVC to YUV	4k	16	0	10	0	97	0	274	30
HEVC to YUV	4k	16	0	10	0	93	0	271	34
YUV to AVC	4k	1	0	10	1	0	24	73	81
YUV to HEVC	4k	1	0	10	1	0	21	73	82
YUV to AV1	4k	1	0	10	1	0	25	72	81
YUV to AVC	4k	4	0	10	0	0	53	177	71
YUV to HEVC	4k	4	0	10	0	0	51	181	74
YUV to AV1	4k	4	0	10	0	0	56	167	67
AVC to YUV	1080p	1	0	8	1	20	0	475	49
HEVC to YUV	1080p	1	0	8	1	19	0	495	48
AVC to YUV	1080p	40	0	8	0	75	0	1530	14
HEVC to YUV	1080p	40	0	8	0	83	0	1590	15
YUV to AVC	1080p	1	0	8	1	0	34	437	68
YUV to HEVC	1080p	1	0	8	1	0	32	434	68
YUV to AV1	1080p	1	0	8	1	0	37	417	66
YUV to AVC	1080p	32	0	8	0	0	100	1281	19
YUV to HEVC	1080p	32	0	8	0	0	99	1338	22
YUV to AV1	1080p	32	0	8	0	0	99	1140	15
AVC to AVC	1080p	1	1	8	1	28	56	726	12
AVC to HEVC	1080p	1	1	8	1	27	52	714	13
AVC to AV1	1080p	1	1	8	1	25	58	652	12
HEVC to AVC	1080p	1	1	8	1	27	61	767	14
HEVC to HEVC	1080p	1	1	8	1	26	57	774	14
HEVC to AV1	1080p	1	1	8	1	23	60	686	13
AVC to AVC	1080p	32	1	8	0	75	99	951	2
AVC to HEVC	1080p	32	1	8	0	83	99	1073	2
AVC to AV1	1080p	32	1	8	0	77	99	1031	2
HEVC to AVC	1080p	32	1	8	0	70	99	991	2
HEVC to HEVC	1080p	32	1	8	0	78	99	1122	2
HEVC to AV1	1080p	32	1	8	0	67	100	1056	2
AVC to YUV	1080p	1	0	10	1	16	0	358	51
HEVC to YUV	1080p	1	0	10	1	14	0	363	49
AVC to YUV	1080p	40	0	10	0	56	0	1025	15
HEVC to YUV	1080p	40	0	10	0	69	0	1016	15

YUV to AVC	1080p	1	0	10	1	0	18	234	72
YUV to HEVC	1080p	1	0	10	1	0	16	234	71
YUV to AV1	1080p	1	0	10	1	0	18	228	70
YUV to AVC	1080p	32	0	10	0	0	52	689	32
YUV to HEVC	1080p	32	0	10	0	0	48	688	32
YUV to AV1	1080p	32	0	10	0	0	58	690	32
AVC to YUV	720p	1	0	8	1	28	0	753	38
HEVC to YUV	720p	1	0	8	1	26	0	795	37
AVC to YUV	720p	100	0	8	0	100	0	2540	4
HEVC to YUV	720p	100	0	8	0	93	0	2746	5
YUV to AVC	720p	1	0	8	1	0	32	885	27
YUV to HEVC	720p	1	0	8	1	0	32	889	27
YUV to AV1	720p	1	0	8	1	0	36	792	24
YUV to AVC	720p	64	0	8	0	0	99	2255	4
YUV to HEVC	720p	64	0	8	0	0	99	2282	4
YUV to AV1	720p	64	0	8	0	0	100	1925	3
AVC to AVC	720p	1	1	8	1	28	27	785	11
AVC to HEVC	720p	1	1	8	1	28	27	783	11
AVC to AV1	720p	1	1	8	1	25	31	697	11
HEVC to AVC	720p	1	1	8	1	26	29	837	11
HEVC to HEVC	720p	1	1	8	1	27	29	844	11
HEVC to AV1	720p	1	1	8	1	24	33	749	11
AVC to AVC	720p	64	1	8	0	93	99	2025	1
AVC to HEVC	720p	64	1	8	0	93	100	2105	1
AVC to AV1	720p	64	1	8	0	75	100	1771	1
HEVC to AVC	720p	64	1	8	0	83	99	2023	1
HEVC to HEVC	720p	64	1	8	0	84	98	2110	1
HEVC to AV1	720p	64	1	8	0	64	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	57.82	58.84	57.45	0.04
YUV to HEVC	8k	1	54.83	59.47	54.16	0.66
YUV to AVC	4k	1	15.32	17.48	14.99	0.07
YUV to HEVC	4k	1	15.93	18.55	15.02	0.20
YUV to AV1	4k	1	20.05	25.74	16.48	1.22
YUV to AVC	4k	4	15.63	24.55	14.99	0.51
YUV to HEVC	4k	4	16.30	22.02	15.09	0.60
YUV to AV1	4k	4	20.49	27.92	16.38	1.49
YUV to AVC	4k	8	20.87	37.78	15.02	24.58
YUV to HEVC	4k	8	21.58	36.61	15.25	27.09
YUV to AV1	4k	8	29.12	47.83	16.57	41.06
YUV to AVC	1080p	1	4.66	6.82	4.47	0.03
YUV to HEVC	1080p	1	4.74	6.16	4.45	0.02
YUV to AV1	1080p	1	6.11	7.68	4.90	0.11
YUV to AVC	1080p	32	19.76	94.81	4.42	32.67
YUV to HEVC	1080p	32	21.42	108.01	4.43	40.03
YUV to AV1	1080p	32	31.36	48.33	5.18	59.82
YUV to AVC	720p	1	2.58	4.07	2.40	0.03
YUV to HEVC	720p	1	2.77	4.16	2.57	0.01
YUV to AV1	720p	1	3.50	5.09	2.96	0.04
YUV to AVC	720p	64	22.47	107.48	2.43	63.62
YUV to HEVC	720p	64	22.48	118.40	2.57	63.85
YUV to AV1	720p	64	29.26	49.29	3.11	87.23

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	19	56	103
RGBA to HEVC	4k	1	1	15	54	104
RGBA to AVC	4k	4	0	47	143	86
RGBA to HEVC	4k	4	0	43	148	89
RGBA to AVC	4k	8	0	50	150	98
RGBA to HEVC	4k	8	0	46	151	98
RGBA to AVC	1080p	1	1	15	195	76
RGBA to AVC	1080p	8	0	47	587	67
RGBA to HEVC	1080p	1	1	14	193	76
RGBA to HEVC	1080p	8	0	43	584	63
RGBA to AVC	1080p	16	0	44	550	63
RGBA to HEVC	1080p	16	0	41	550	62
RGBA to AVC	720p	1	1	15	380	68
RGBA to HEVC	720p	1	1	15	379	68
RGBA to AVC	720p	16	0	50	1204	34
RGBA to HEVC	720p	16	0	50	1193	33
RGBA to AVC	720p	32	0	48	1165	32
RGBA to HEVC	720p	32	0	49	1168	32

15. T1A – GStreamer XStack Throughput

15.1 Transcoding

15.1.1 Description

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

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

The YUV frame is encoded with hardware encoder.

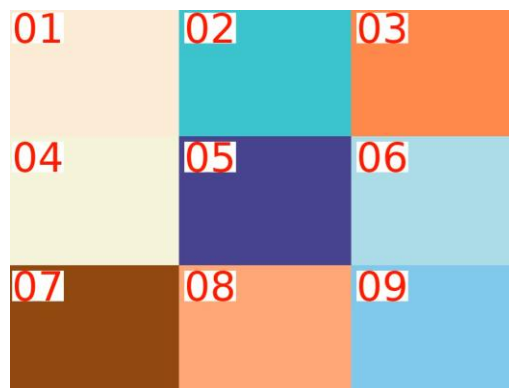
The encoded bitstream is read out through PCIe and written into an output file.

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

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

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

This test is HEVC to AVC only.



Example output in a 3x3 layout with 9 inputs

15.1.2 Command line

See Appendix A: GStreamer XStack Command

15.2 GStreamer XStack Performance Results

Input Res	Grid	Output Res	Cell Size	FPS	CPU	Dec Load	Enc Load	Scaler Load
1920x1080	3x3	1920x1080	640x360	149.69	30	66	13	15
1920x1080	4x4	1920x1080	480x270	105.74	41	84	10	15
1920x1080	4x8	1920x1080	480x135	57.06	63	90	5	8
1920x1080	3x3	3840x2160	1280x720	69.96	15	32	22	19
1920x1080	4x4	3840x2160	960x540	56.06	23	48	20	19
1920x1080	4x8	3840x2160	960x270	43.95	43	72	15	17
1920x1080	3x3	7680x4320	2560x1440	19.82	8	8	22	15
1920x1080	4x4	7680x4320	1920x1080	19.11	10	15	22	18
1920x1080	4x8	7680x4320	1920x540	19.34	19	33	22	20
1920x1080	6x6	1920x1080	320x180	50.66	62	89	5	10
1920x1080	7x7	1920x1080	274x154 276x154 274x156 276x156*	37.48	64	89	4	8

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

16. T1A – GStreamer Ladder Generation

16.1 Transcoding

16.1.1 Description

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

Decoded YUV is split to multiple pads.

The YUV frames are encoded with hardware encoder.

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

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

This test is AVC to HEVC only

16.1.2 Command line

See Appendix C: GStreamer Ladder Command

16.2 GStreamer Ladder Performance Results

Jobs	Outputs	FPS	CPU	Dec Load	Enc Load
1	64	18.99	58	1	92

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	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	12	0	102	9
HEVC to YUV	8k	1	0	8	1	12	0	96	18
VP9 to YUV	8k	1	0	8	1	24	0	38	6
YUV to AVC	8k	1	0	8	1	0	93	61	93
YUV to HEVC	8k	1	0	8	1	0	97	82	136
AVC to AVC	8k	1	1	8	1	7	99	50	4
AVC to HEVC	8k	1	1	8	1	6	99	67	9
HEVC to AVC	8k	1	1	8	1	7	99	49	3
HEVC to HEVC	8k	1	1	8	1	5	100	66	12
VP9 to AVC	8k	1	1	8	1	24	44	35	3
VP9 to HEVC	8k	1	1	8	1	24	42	37	2
AVC to YUV	8k	1	0	10	1	11	0	59	12
HEVC to YUV	8k	1	0	10	1	10	0	53	7
VP9 to YUV	8k	1	0	10	1	24	0	33	15
YUV to AVC	8k	1	0	10	1	0	81	41	117
YUV to HEVC	8k	1	0	10	1	0	47	41	123
AVC to YUV	4k	1	0	8	1	20	0	291	16
HEVC to YUV	4k	1	0	8	1	20	0	290	19
VP9 to YUV	4k	1	0	8	1	24	0	137	7
AVC to YUV	4k	16	0	8	0	98	0	478	3
HEVC to YUV	4k	16	0	8	0	100	0	499	5
VP9 to YUV	4k	16	0	8	0	99	0	463	1
YUV to AVC	4k	1	0	8	1	0	91	283	45
YUV to HEVC	4k	1	0	8	1	0	93	318	56
YUV to AV1	4k	1	0	8	1	0	93	280	48
YUV to AVC	4k	4	0	8	0	0	96	300	18
YUV to HEVC	4k	4	0	8	0	0	96	332	18
YUV to AV1	4k	4	0	8	0	0	96	288	15
YUV to AVC	4k	8	0	8	0	0	99	320	9
YUV to HEVC	4k	8	0	8	0	0	100	344	8
YUV to AV1	4k	8	0	8	0	0	99	296	7
AVC to AVC	4k	1	1	8	1	14	96	213	16
AVC to HEVC	4k	1	1	8	1	15	91	265	12
AVC to AV1	4k	1	1	8	1	15	93	256	15
HEVC to AVC	4k	1	1	8	1	13	93	211	16
HEVC to HEVC	4k	1	1	8	1	14	91	264	16
HEVC to AV1	4k	1	1	8	1	14	92	250	15
VP9 to AVC	4k	1	1	8	1	24	41	130	7
VP9 to HEVC	4k	1	1	8	1	23	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	23	45	134	9
AVC to AVC	4k	4	1	8	0	62	96	232	3
AVC to HEVC	4k	4	1	8	0	69	96	296	7
AVC to AV1	4k	4	1	8	0	58	95	272	3
HEVC to AVC	4k	4	1	8	0	58	96	228	2
HEVC to HEVC	4k	4	1	8	0	62	96	292	9
HEVC to AV1	4k	4	1	8	0	47	96	275	2
VP9 to AVC	4k	4	1	8	0	67	96	230	2
VP9 to HEVC	4k	4	1	8	0	72	96	288	6
VP9 to AV1	4k	4	1	8	0	63	94	272	2
AVC to AVC	4k	8	1	8	0	69	100	203	2
AVC to HEVC	4k	8	1	8	0	77	99	271	1
AVC to AV1	4k	8	1	8	0	70	100	264	2
HEVC to AVC	4k	8	1	8	0	65	100	203	1
HEVC to HEVC	4k	8	1	8	0	70	100	270	2
HEVC to AV1	4k	8	1	8	0	61	100	268	1
VP9 to AVC	4k	8	1	8	0	69	100	211	1
VP9 to HEVC	4k	8	1	8	0	78	99	272	2
VP9 to AV1	4k	8	1	8	0	68	100	272	2
AVC to YUV	4k	1	0	10	1	19	0	211	9
HEVC to YUV	4k	1	0	10	1	18	0	210	9
VP9 to YUV	4k	1	0	10	1	24	0	125	8
AVC to YUV	4k	16	0	10	0	98	0	274	0
HEVC to YUV	4k	16	0	10	0	98	0	268	0
VP9 to YUV	4k	16	0	10	0	98	0	249	0
YUV to AVC	4k	1	0	10	1	0	63	180	61
YUV to HEVC	4k	1	0	10	1	0	55	184	66
YUV to AV1	4k	1	0	10	1	0	61	181	64
YUV to AVC	4k	4	0	10	0	0	87	210	30
YUV to HEVC	4k	4	0	10	0	0	74	242	46
YUV to AV1	4k	4	0	10	0	0	80	235	46
AVC to YUV	1080p	1	0	8	1	35	0	752	18
HEVC to YUV	1080p	1	0	8	1	31	0	728	27
VP9 to YUV	1080p	1	0	8	1	24	0	555	7
AVC to YUV	1080p	40	0	8	0	87	0	1714	0
HEVC to YUV	1080p	40	0	8	0	97	0	1800	1
VP9 to YUV	1080p	40	0	8	0	79	0	1760	0
YUV to AVC	1080p	1	0	8	1	0	49	633	30
YUV to HEVC	1080p	1	0	8	1	0	46	631	27
YUV to AV1	1080p	1	0	8	1	0	48	581	30

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1280	4
YUV to HEVC	1080p	32	0	8	0	0	99	1357	4
YUV to AV1	1080p	32	0	8	0	0	99	1184	2
AVC to AVC	1080p	1	1	8	1	41	79	933	22
AVC to HEVC	1080p	1	1	8	1	43	76	987	20
AVC to AV1	1080p	1	1	8	1	38	78	889	22
HEVC to AVC	1080p	1	1	8	1	34	75	873	33
HEVC to HEVC	1080p	1	1	8	1	37	76	949	36
HEVC to AV1	1080p	1	1	8	1	33	76	840	27
VP9 to AVC	1080p	1	1	8	1	22	41	534	6
VP9 to HEVC	1080p	1	1	8	1	22	39	541	7
VP9 to AV1	1080p	1	1	8	1	22	46	540	8
AVC to AVC	1080p	32	1	8	0	75	100	911	1
AVC to HEVC	1080p	32	1	8	0	84	99	1028	0
AVC to AV1	1080p	32	1	8	0	78	99	998	0
HEVC to AVC	1080p	32	1	8	0	72	99	952	3
HEVC to HEVC	1080p	32	1	8	0	79	100	1071	1
HEVC to AV1	1080p	32	1	8	0	68	99	1024	1
VP9 to AVC	1080p	32	1	8	0	69	99	1056	0
VP9 to HEVC	1080p	32	1	8	0	75	99	1211	0
VP9 to AV1	1080p	32	1	8	0	71	99	1120	0
AVC to YUV	1080p	1	0	10	1	28	0	531	7
HEVC to YUV	1080p	1	0	10	1	27	0	527	7
VP9 to YUV	1080p	1	0	10	1	26	0	421	10
AVC to YUV	1080p	40	0	10	0	63	0	1080	0
HEVC to YUV	1080p	40	0	10	0	68	0	1080	0
VP9 to YUV	1080p	40	0	10	0	78	0	1040	0
YUV to AVC	1080p	1	0	10	1	0	33	428	39
YUV to HEVC	1080p	1	0	10	1	0	30	428	39
YUV to AV1	1080p	1	0	10	1	0	33	402	39
YUV to AVC	1080p	32	0	10	0	0	61	800	6
YUV to HEVC	1080p	32	0	10	0	0	58	800	7
YUV to AV1	1080p	32	0	10	0	0	64	768	6
AVC to YUV	720p	1	0	8	1	42	0	1124	19
HEVC to YUV	720p	1	0	8	1	37	0	1126	21
VP9 to YUV	720p	1	0	8	1	34	0	1022	6
AVC to YUV	720p	100	0	8	0	100	0	2545	0
HEVC to YUV	720p	100	0	8	0	100	0	2911	1
VP9 to YUV	720p	100	0	8	0	100	0	2686	0
YUV to AVC	720p	1	0	8	1	0	31	883	15

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	30	883	12
YUV to AV1	720p	1	0	8	1	0	35	805	14
YUV to AVC	720p	64	0	8	0	0	99	2313	1
YUV to HEVC	720p	64	0	8	0	0	97	2372	2
YUV to AV1	720p	64	0	8	0	0	99	1984	1
AVC to AVC	720p	1	1	8	1	47	48	1245	24
AVC to HEVC	720p	1	1	8	1	47	48	1247	21
AVC to AV1	720p	1	1	8	1	40	51	1086	19
HEVC to AVC	720p	1	1	8	1	40	46	1206	24
HEVC to HEVC	720p	1	1	8	1	40	46	1212	22
HEVC to AV1	720p	1	1	8	1	34	48	1049	26
VP9 to AVC	720p	1	1	8	1	31	35	979	10
VP9 to HEVC	720p	1	1	8	1	32	35	988	13
VP9 to AV1	720p	1	1	8	1	31	45	991	9
AVC to AVC	720p	64	1	8	0	93	100	1982	0
AVC to HEVC	720p	64	1	8	0	95	100	2059	0
AVC to AV1	720p	64	1	8	0	76	100	1733	0
HEVC to AVC	720p	64	1	8	0	83	100	1970	1
HEVC to HEVC	720p	64	1	8	0	84	100	2060	0
HEVC to AV1	720p	64	1	8	0	68	100	1738	0
VP9 to AVC	720p	64	1	8	0	94	100	2192	0
VP9 to HEVC	720p	64	1	8	0	94	100	2276	0
VP9 to AV1	720p	64	1	8	0	76	100	1866	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	11	0	90	8
HEVC to YUV	8k	1	0	8	1	11	0	90	8
VP9 to YUV	8k	1	0	8	1	24	0	37	3
YUV to AVC	8k	1	0	8	1	0	90	59	27
YUV to HEVC	8k	1	0	8	1	0	86	75	34
AVC to AVC	8k	1	1	8	1	8	100	50	5
AVC to HEVC	8k	1	1	8	1	10	99	67	5
HEVC to AVC	8k	1	1	8	1	9	100	48	3
HEVC to HEVC	8k	1	1	8	1	11	99	66	4
VP9 to AVC	8k	1	1	8	1	24	45	34	2
VP9 to HEVC	8k	1	1	8	1	24	43	37	2
AVC to YUV	8k	1	0	10	1	10	0	53	8
HEVC to YUV	8k	1	0	10	1	9	0	48	8
VP9 to YUV	8k	1	0	10	1	24	0	32	6
YUV to AVC	8k	1	0	10	1	0	60	34	35
YUV to HEVC	8k	1	0	10	1	0	40	35	36
AVC to YUV	4k	1	0	8	1	21	0	271	9
HEVC to YUV	4k	1	0	8	1	20	0	275	7
VP9 to YUV	4k	1	0	8	1	23	0	136	4
AVC to YUV	4k	16	0	8	0	98	0	483	0
HEVC to YUV	4k	16	0	8	0	98	0	501	0
VP9 to YUV	4k	16	0	8	0	100	0	467	0
YUV to AVC	4k	1	0	8	1	0	72	231	25
YUV to HEVC	4k	1	0	8	1	0	68	236	26
YUV to AV1	4k	1	0	8	1	0	75	226	24
YUV to AVC	4k	4	0	8	0	0	91	298	13
YUV to HEVC	4k	4	0	8	0	0	94	328	12
YUV to AV1	4k	4	0	8	0	0	96	287	12
YUV to AVC	4k	8	0	8	0	0	80	260	18
YUV to HEVC	4k	8	0	8	0	0	99	347	7
YUV to AV1	4k	8	0	8	0	0	100	301	9
AVC to AVC	4k	1	1	8	1	15	97	217	6
AVC to HEVC	4k	1	1	8	1	16	96	271	7
AVC to AV1	4k	1	1	8	1	16	96	258	8
HEVC to AVC	4k	1	1	8	1	16	97	216	4
HEVC to HEVC	4k	1	1	8	1	17	97	276	6
HEVC to AV1	4k	1	1	8	1	16	97	262	5
VP9 to AVC	4k	1	1	8	1	23	41	129	3

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	1	23	38	134	3
VP9 to AV1	4k	1	1	8	1	23	45	134	3
AVC to AVC	4k	4	1	8	0	62	97	238	4
AVC to HEVC	4k	4	1	8	0	68	96	298	4
AVC to AV1	4k	4	1	8	0	58	95	276	4
HEVC to AVC	4k	4	1	8	0	60	96	231	3
HEVC to HEVC	4k	4	1	8	0	62	96	287	3
HEVC to AV1	4k	4	1	8	0	51	96	271	4
VP9 to AVC	4k	4	1	8	0	65	97	232	3
VP9 to HEVC	4k	4	1	8	0	71	96	288	3
VP9 to AV1	4k	4	1	8	0	61	96	273	3
AVC to AVC	4k	8	1	8	0	69	99	208	2
AVC to HEVC	4k	8	1	8	0	78	99	271	3
AVC to AV1	4k	8	1	8	0	70	100	267	3
HEVC to AVC	4k	8	1	8	0	66	99	208	2
HEVC to HEVC	4k	8	1	8	0	70	100	270	3
HEVC to AV1	4k	8	1	8	0	65	99	266	3
VP9 to AVC	4k	8	1	8	0	68	100	212	2
VP9 to HEVC	4k	8	1	8	0	75	100	278	3
VP9 to AV1	4k	8	1	8	0	71	100	274	2
AVC to YUV	4k	1	0	10	1	18	0	197	8
HEVC to YUV	4k	1	0	10	1	16	0	189	7
VP9 to YUV	4k	1	0	10	1	24	0	124	5
AVC to YUV	4k	16	0	10	0	101	0	276	0
HEVC to YUV	4k	16	0	10	0	97	0	272	0
VP9 to YUV	4k	16	0	10	0	98	0	254	0
YUV to AVC	4k	1	0	10	1	0	41	134	28
YUV to HEVC	4k	1	0	10	1	0	38	136	27
YUV to AV1	4k	1	0	10	1	0	43	129	26
YUV to AVC	4k	4	0	10	0	0	51	165	26
YUV to HEVC	4k	4	0	10	0	0	46	165	27
YUV to AV1	4k	4	0	10	0	0	50	157	27
AVC to YUV	1080p	1	0	8	1	35	0	725	13
HEVC to YUV	1080p	1	0	8	1	33	0	759	10
VP9 to YUV	1080p	1	0	8	1	24	0	555	5
AVC to YUV	1080p	40	0	8	0	79	0	1582	0
HEVC to YUV	1080p	40	0	8	0	86	0	1674	0
VP9 to YUV	1080p	40	0	8	0	70	0	1618	0
YUV to AVC	1080p	1	0	8	1	0	51	660	19

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	48	665	16
YUV to AV1	1080p	1	0	8	1	0	46	547	16
YUV to AVC	1080p	32	0	8	0	0	92	1240	3
YUV to HEVC	1080p	32	0	8	0	0	91	1271	3
YUV to AV1	1080p	32	0	8	0	0	99	1199	3
AVC to AVC	1080p	1	1	8	1	43	84	981	15
AVC to HEVC	1080p	1	1	8	1	44	80	1035	17
AVC to AV1	1080p	1	1	8	1	41	80	919	16
HEVC to AVC	1080p	1	1	8	1	39	85	971	12
HEVC to HEVC	1080p	1	1	8	1	41	86	1071	12
HEVC to AV1	1080p	1	1	8	1	37	83	928	12
VP9 to AVC	1080p	1	1	8	1	22	42	545	5
VP9 to HEVC	1080p	1	1	8	1	22	39	539	5
VP9 to AV1	1080p	1	1	8	1	22	46	540	6
AVC to AVC	1080p	32	1	8	0	76	99	948	1
AVC to HEVC	1080p	32	1	8	0	85	99	1068	1
AVC to AV1	1080p	32	1	8	0	79	100	1037	1
HEVC to AVC	1080p	32	1	8	0	71	100	982	0
HEVC to HEVC	1080p	32	1	8	0	78	100	1107	0
HEVC to AV1	1080p	32	1	8	0	69	99	1056	0
VP9 to AVC	1080p	32	1	8	0	70	100	1097	0
VP9 to HEVC	1080p	32	1	8	0	74	99	1239	0
VP9 to AV1	1080p	32	1	8	0	72	99	1146	0
AVC to YUV	1080p	1	0	10	1	26	0	638	7
HEVC to YUV	1080p	1	0	10	1	27	0	495	7
VP9 to YUV	1080p	1	0	10	1	26	0	411	6
AVC to YUV	1080p	40	0	10	0	63	0	1059	0
HEVC to YUV	1080p	40	0	10	0	69	0	1056	0
VP9 to YUV	1080p	40	0	10	0	70	0	1006	0
YUV to AVC	1080p	1	0	10	1	0	29	380	21
YUV to HEVC	1080p	1	0	10	1	0	27	382	22
YUV to AV1	1080p	1	0	10	1	0	28	338	20
YUV to AVC	1080p	32	0	10	0	0	56	737	5
YUV to HEVC	1080p	32	0	10	0	0	53	744	5
YUV to AV1	1080p	32	0	10	0	0	58	703	8
AVC to YUV	720p	1	0	8	1	41	0	1092	11
HEVC to YUV	720p	1	0	8	1	39	0	1153	8
VP9 to YUV	720p	1	0	8	1	33	0	1022	5
AVC to YUV	720p	100	0	8	0	100	0	2743	0

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	98	0	2896	0
VP9 to YUV	720p	100	0	8	0	92	0	2639	0
YUV to AVC	720p	1	0	8	1	0	32	948	12
YUV to HEVC	720p	1	0	8	1	0	32	955	13
YUV to AV1	720p	1	0	8	1	0	31	742	13
YUV to AVC	720p	64	0	8	0	0	99	2559	2
YUV to HEVC	720p	64	0	8	0	0	97	2607	2
YUV to AV1	720p	64	0	8	0	0	100	2160	1
AVC to AVC	720p	1	1	8	1	51	49	1333	15
AVC to HEVC	720p	1	1	8	1	50	49	1338	16
AVC to AV1	720p	1	1	8	1	43	49	1067	14
HEVC to AVC	720p	1	1	8	1	46	51	1367	11
HEVC to HEVC	720p	1	1	8	1	46	51	1373	11
HEVC to AV1	720p	1	1	8	1	38	49	1073	11
VP9 to AVC	720p	1	1	8	1	31	35	984	6
VP9 to HEVC	720p	1	1	8	1	31	35	990	6
VP9 to AV1	720p	1	1	8	1	31	45	990	9
AVC to AVC	720p	64	1	8	0	100	100	2112	0
AVC to HEVC	720p	64	1	8	0	100	100	2207	0
AVC to AV1	720p	64	1	8	0	94	100	1845	0
HEVC to AVC	720p	64	1	8	0	94	100	2096	0
HEVC to HEVC	720p	64	1	8	0	96	100	2187	0
HEVC to AV1	720p	64	1	8	0	89	100	1841	0
VP9 to AVC	720p	64	1	8	0	100	100	2369	0
VP9 to HEVC	720p	64	1	8	0	100	100	2447	0
VP9 to AV1	720p	64	1	8	0	98	100	1992	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	58.86	60.68	57.80	0.15
YUV to HEVC	8k	1	56.46	61.73	54.75	1.09
YUV to AVC	4k	1	15.75	18.50	15.49	0.06
YUV to HEVC	4k	1	17.04	19.07	16.00	0.09
YUV to AV1	4k	1	22.08	25.83	16.61	0.64
YUV to AVC	4k	4	16.39	20.86	15.14	0.61
YUV to HEVC	4k	4	18.82	22.11	15.69	1.33
YUV to AV1	4k	4	23.41	27.94	16.39	0.91
YUV to AVC	4k	8	20.74	35.18	15.21	25.59
YUV to HEVC	4k	8	22.48	37.65	15.85	35.65
YUV to AV1	4k	8	37.24	45.77	26.33	12.12
YUV to AVC	1080p	1	4.83	5.69	4.72	0.02
YUV to HEVC	1080p	1	5.24	9.67	4.99	0.10
YUV to AV1	1080p	1	6.88	9.18	5.19	0.05
YUV to AVC	1080p	32	7.80	13.41	4.98	1.51
YUV to HEVC	1080p	32	8.60	15.42	5.55	2.13
YUV to AV1	1080p	32	41.01	46.90	33.02	2.12
YUV to AVC	720p	1	2.77	3.39	2.44	0.01
YUV to HEVC	720p	1	2.77	3.16	2.72	0.00
YUV to AV1	720p	1	3.75	8.44	2.94	0.09
YUV to AVC	720p	64	6.29	12.31	3.64	1.21
YUV to HEVC	720p	64	6.49	12.42	3.72	1.29
YUV to AV1	720p	64	39.77	47.47	31.88	3.39

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	6
HEVC to RGBA	8k	1	4	0	153	7
VP9 to RGBA	8k	1	23	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	93	4	657	2
VP9 to RGBA	4k	1	21	0	144	7
VP9 to RGBA	4k	16	94	3	590	1
AVC to RGBA	1080p	40	93	14	1884	1
HEVC to RGBA	1080p	40	92	14	1991	1
VP9 to RGBA	1080p	40	94	17	2440	0
AVC to RGBA	720p	100	96	18	2613	0
HEVC to RGBA	720p	100	90	20	2821	0
VP9 to RGBA	720p	64	95	19	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:sempplanar0=1:enableOut1=1:sempplanar1=1:scale1=1280x720:enableO  
ut2=1:sempplanar2=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	32	92	2	432	3
AVC to HEVC	8	32	93	2	466	4
AVC to AV1	8	22	94	1	384	2
HEVC to AVC	8	34	93	2	439	4
HEVC to HEVC	8	33	94	2	472	4
HEVC to AV1	8	24	96	2	392	3
VP9 to AVC	8	39	92	2	432	3
VP9 to HEVC	8	38	93	2	464	3
VP9 to AV1	8	28	94	2	392	2

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	141	67
RGBA to HEVC	4k	1	1	42	144	67
RGBA to AV1	4k	1	1	48	142	75
RGBA to AVC	4k	4	0	63	160	39
RGBA to HEVC	4k	4	0	60	164	40
RGBA to AV1	4k	4	0	62	164	44
RGBA to AVC	4k	8	0	65	167	35
RGBA to HEVC	4k	8	0	58	169	37
RGBA to AV1	4k	8	0	61	172	40
RGBA to AVC	1080p	1	1	28	377	51
RGBA to HEVC	1080p	1	1	27	377	45
RGBA to AV1	1080p	1	1	30	363	51
RGBA to AVC	1080p	16	0	61	624	23
RGBA to HEVC	1080p	16	0	54	624	24
RGBA to AV1	1080p	16	0	55	608	22
RGBA to AVC	1080p	32	0	56	643	11
RGBA to HEVC	1080p	32	0	52	643	10
RGBA to AV1	1080p	32	0	56	640	14
RGBA to AVC	720p	1	1	28	659	60
RGBA to HEVC	720p	1	1	27	657	60
RGBA to AV1	720p	1	1	32	627	61
RGBA to AVC	720p	16	0	47	1174	35
RGBA to HEVC	720p	16	0	46	1168	33
RGBA to AV1	720p	16	0	54	1113	31
RGBA to AVC	720p	32	0	45	1150	31
RGBA to HEVC	720p	32	0	46	1154	32
RGBA to AV1	720p	32	0	56	1151	30

23. T1U – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

23.1 Encoding

23.1.1 Description

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

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

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

23.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

23.2 Encoding EnableRdoQuant/rdoLevel/lookaheadDepth Performance Results

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	0	0	1	96	304	14
YUV to HEVC	4k	4	0	0	1	97	332	15
YUV to AV1	4k	4	0	0	1	96	288	13
YUV to AVC	4k	4	0	0	2	96	298	16
YUV to HEVC	4k	4	0	0	2	99	172	7
YUV to AV1	4k	4	0	0	2	98	140	6
YUV to AVC	4k	4	0	0	3	96	304	15
YUV to HEVC	4k	4	0	0	3	98	100	4
YUV to AV1	4k	4	0	0	3	99	76	4
YUV to AVC	4k	4	0	1	1	97	188	9
YUV to HEVC	4k	4	0	1	1	97	240	11
YUV to AVC	4k	4	0	1	2	98	188	8
YUV to HEVC	4k	4	0	1	2	99	108	6
YUV to AVC	4k	4	0	1	3	98	188	8
YUV to HEVC	4k	4	0	1	3	99	68	4
YUV to AVC	4k	4	4	0	1	99	196	9
YUV to HEVC	4k	4	4	0	1	99	228	15
YUV to AV1	4k	4	4	0	1	99	200	11
YUV to AVC	4k	4	4	0	2	99	196	12
YUV to HEVC	4k	4	4	0	2	99	136	8
YUV to AV1	4k	4	4	0	2	99	116	6
YUV to AVC	4k	4	4	0	3	100	196	12
YUV to HEVC	4k	4	4	0	3	100	88	4
YUV to AV1	4k	4	4	0	3	98	72	3
YUV to AVC	4k	4	4	1	1	99	140	9
YUV to HEVC	4k	4	4	1	1	100	180	8
YUV to AVC	4k	4	4	1	2	99	140	7
YUV to HEVC	4k	4	4	1	2	100	96	5
YUV to AVC	4k	4	4	1	3	100	140	7
YUV to HEVC	4k	4	4	1	3	101	64	3
YUV to AVC	4k	4	16	0	1	99	196	11
YUV to HEVC	4k	4	16	0	1	99	228	13
YUV to AV1	4k	4	16	0	1	99	200	13
YUV to AVC	4k	4	16	0	2	100	196	12
YUV to HEVC	4k	4	16	0	2	99	136	8
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	99	88	4
YUV to AV1	4k	4	16	0	3	99	72	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	16	1	1	100	140	7
YUV to HEVC	4k	4	16	1	1	99	179	9
YUV to AVC	4k	4	16	1	2	100	140	6
YUV to HEVC	4k	4	16	1	2	99	96	3
YUV to AVC	4k	4	16	1	3	100	140	6
YUV to HEVC	4k	4	16	1	3	101	64	4
YUV to AVC	4k	4	40	0	1	100	196	10
YUV to HEVC	4k	4	40	0	1	99	224	11
YUV to AV1	4k	4	40	0	1	100	196	10
YUV to AVC	4k	4	40	0	2	100	196	10
YUV to HEVC	4k	4	40	0	2	99	136	8
YUV to AV1	4k	4	40	0	2	100	112	6
YUV to AVC	4k	4	40	0	3	100	196	9
YUV to HEVC	4k	4	40	0	3	99	88	6
YUV to AV1	4k	4	40	0	3	98	72	3
YUV to AVC	4k	4	40	1	1	100	136	7
YUV to HEVC	4k	4	40	1	1	100	176	9
YUV to AVC	4k	4	40	1	2	99	136	7
YUV to HEVC	4k	4	40	1	2	99	92	5
YUV to AVC	4k	4	40	1	3	99	136	7
YUV to HEVC	4k	4	40	1	3	99	64	3
YUV to AVC	1080p	20	0	0	1	99	1280	4
YUV to HEVC	1080p	20	0	0	1	99	1360	6
YUV to AV1	1080p	20	0	0	1	99	1180	4
YUV to AVC	1080p	20	0	0	2	99	1280	4
YUV to HEVC	1080p	20	0	0	2	100	689	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	100	400	1
YUV to AV1	1080p	20	0	0	3	100	300	1
YUV to AVC	1080p	20	0	1	1	99	760	2
YUV to HEVC	1080p	20	0	1	1	99	960	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	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
YUV to AV1	1080p	20	4	0	1	99	720	3
YUV to AVC	1080p	20	4	0	2	99	700	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	20	4	0	2	100	515	1
YUV to AV1	1080p	20	4	0	2	99	421	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	267	1
YUV to AVC	1080p	20	4	1	1	99	501	1
YUV to HEVC	1080p	20	4	1	1	99	660	2
YUV to AVC	1080p	20	4	1	2	99	500	2
YUV to HEVC	1080p	20	4	1	2	99	360	1
YUV to AVC	1080p	20	4	1	3	99	500	2
YUV to HEVC	1080p	20	4	1	3	100	240	1
YUV to AVC	1080p	20	16	0	1	99	700	2
YUV to HEVC	1080p	20	16	0	1	99	819	3
YUV to AV1	1080p	20	16	0	1	99	718	2
YUV to AVC	1080p	20	16	0	2	99	700	2
YUV to HEVC	1080p	20	16	0	2	99	501	2
YUV to AV1	1080p	20	16	0	2	100	421	1
YUV to AVC	1080p	20	16	0	3	99	700	3
YUV to HEVC	1080p	20	16	0	3	99	340	1
YUV to AV1	1080p	20	16	0	3	100	262	1
YUV to AVC	1080p	20	16	1	1	99	500	1
YUV to HEVC	1080p	20	16	1	1	99	650	2
YUV to AVC	1080p	20	16	1	2	99	500	2
YUV to HEVC	1080p	20	16	1	2	99	360	1
YUV to AVC	1080p	20	16	1	3	99	500	1
YUV to HEVC	1080p	20	16	1	3	99	240	1
YUV to AVC	1080p	20	40	0	1	99	686	2
YUV to HEVC	1080p	20	40	0	1	99	801	3
YUV to AV1	1080p	20	40	0	1	99	700	2
YUV to AVC	1080p	20	40	0	2	99	687	2
YUV to HEVC	1080p	20	40	0	2	99	500	1
YUV to AV1	1080p	20	40	0	2	100	420	1
YUV to AVC	1080p	20	40	0	3	99	690	2
YUV to HEVC	1080p	20	40	0	3	99	340	1
YUV to AV1	1080p	20	40	0	3	100	260	1
YUV to AVC	1080p	20	40	1	1	99	500	2
YUV to HEVC	1080p	20	40	1	1	100	640	2
YUV to AVC	1080p	20	40	1	2	99	500	2
YUV to HEVC	1080p	20	40	1	2	99	360	1
YUV to AVC	1080p	20	40	1	3	99	500	1

TYPE	RES	JOB	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	93	2286	2
YUV to HEVC	720p	40	0	0	1	91	2295	3
YUV to AV1	720p	40	0	0	1	96	2004	2
YUV to AVC	720p	40	0	0	2	94	2287	3
YUV to HEVC	720p	40	0	0	2	99	1520	1
YUV to AV1	720p	40	0	0	2	99	1201	1
YUV to AVC	720p	40	0	0	3	93	2286	3
YUV to HEVC	720p	40	0	0	3	100	920	1
YUV to AV1	720p	40	0	0	3	99	657	0
YUV to AVC	720p	40	0	1	1	99	1696	1
YUV to HEVC	720p	40	0	1	1	99	2080	2
YUV to AVC	720p	40	0	1	2	99	1691	1
YUV to HEVC	720p	40	0	1	2	99	960	1
YUV to AVC	720p	40	0	1	3	99	1697	1
YUV to HEVC	720p	40	0	1	3	100	605	1
YUV to AVC	720p	40	4	0	1	100	1405	1
YUV to HEVC	720p	40	4	0	1	100	1326	1
YUV to AV1	720p	40	4	0	1	100	1000	1
YUV to AVC	720p	40	4	0	2	100	1400	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	1418	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	1321	1
YUV to AVC	720p	40	4	1	2	99	1076	1
YUV to HEVC	720p	40	4	1	2	99	760	0
YUV to AVC	720p	40	4	1	3	99	1069	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	1339	1
YUV to AV1	720p	40	16	0	1	100	1052	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	0
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	100	1320	1
YUV to AVC	720p	40	16	1	2	99	1045	1
YUV to HEVC	720p	40	16	1	2	99	760	0
YUV to AVC	720p	40	16	1	3	99	1046	1
YUV to HEVC	720p	40	16	1	3	99	520	0
YUV to AVC	720p	40	40	0	1	100	1325	1
YUV to HEVC	720p	40	40	0	1	100	1317	1
YUV to AV1	720p	40	40	0	1	100	1040	1
YUV to AVC	720p	40	40	0	2	100	1345	1
YUV to HEVC	720p	40	40	0	2	99	1040	1
YUV to AV1	720p	40	40	0	2	100	840	1
YUV to AVC	720p	40	40	0	3	100	1339	1
YUV to HEVC	720p	40	40	0	3	100	720	1
YUV to AV1	720p	40	40	0	3	99	560	1
YUV to AVC	720p	40	40	1	1	99	1040	1
YUV to HEVC	720p	40	40	1	1	100	1300	1
YUV to AVC	720p	40	40	1	2	99	1040	1
YUV to HEVC	720p	40	40	1	2	99	760	1
YUV to AVC	720p	40	40	1	3	99	1040	1
YUV to HEVC	720p	40	40	1	3	99	520	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	704	2
YUV to HEVC	1080p	20	0	0	1	19	99	820	2
YUV to AV1	1080p	20	0	0	1	19	99	720	2
YUV to AVC	1080p	20	0	0	2	19	99	701	2
YUV to HEVC	1080p	20	0	0	2	19	100	520	1
YUV to AV1	1080p	20	0	0	2	19	100	440	1
YUV to AVC	1080p	20	0	0	3	19	100	701	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	500	2
YUV to HEVC	1080p	20	0	1	1	19	100	660	2
YUV to AVC	1080p	20	0	1	2	19	100	508	2
YUV to HEVC	1080p	20	0	1	2	19	99	360	1
YUV to AVC	1080p	20	0	1	3	19	99	505	2
YUV to HEVC	1080p	20	0	1	3	19	100	240	1
YUV to AVC	1080p	20	4	0	1	19	99	700	2
YUV to HEVC	1080p	20	4	0	1	19	99	820	3
YUV to AV1	1080p	20	4	0	1	19	99	720	3
YUV to AVC	1080p	20	4	0	2	19	99	701	2
YUV to HEVC	1080p	20	4	0	2	19	99	520	1
YUV to AV1	1080p	20	4	0	2	19	100	420	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	3	19	99	701	2
YUV to HEVC	1080p	20	4	0	3	19	100	340	1
YUV to AV1	1080p	20	4	0	3	19	100	265	1
YUV to AVC	1080p	20	4	1	1	19	99	503	2
YUV to HEVC	1080p	20	4	1	1	19	99	660	2
YUV to AVC	1080p	20	4	1	2	19	100	501	2
YUV to HEVC	1080p	20	4	1	2	19	100	360	1
YUV to AVC	1080p	20	4	1	3	19	100	501	1
YUV to HEVC	1080p	20	4	1	3	19	100	240	1
YUV to AVC	1080p	20	16	0	1	19	99	700	3
YUV to HEVC	1080p	20	16	0	1	19	99	820	3
YUV to AV1	1080p	20	16	0	1	19	99	720	2
YUV to AVC	1080p	20	16	0	2	19	99	700	2
YUV to HEVC	1080p	20	16	0	2	19	99	507	1
YUV to AV1	1080p	20	16	0	2	19	99	420	1
YUV to AVC	1080p	20	16	0	3	19	99	700	2
YUV to HEVC	1080p	20	16	0	3	19	99	340	1
YUV to AV1	1080p	20	16	0	3	19	100	262	1
YUV to AVC	1080p	20	16	1	1	19	99	500	2
YUV to HEVC	1080p	20	16	1	1	19	99	641	2
YUV to AVC	1080p	20	16	1	2	19	100	501	2
YUV to HEVC	1080p	20	16	1	2	19	99	360	1
YUV to AVC	1080p	20	16	1	3	19	99	500	2
YUV to HEVC	1080p	20	16	1	3	19	100	240	1
YUV to AVC	1080p	20	40	0	1	19	100	689	3
YUV to HEVC	1080p	20	40	0	1	19	99	801	2
YUV to AV1	1080p	20	40	0	1	19	99	700	3
YUV to AVC	1080p	20	40	0	2	19	99	689	2
YUV to HEVC	1080p	20	40	0	2	19	99	500	2
YUV to AV1	1080p	20	40	0	2	19	99	420	1
YUV to AVC	1080p	20	40	0	3	19	100	690	3
YUV to HEVC	1080p	20	40	0	3	19	99	340	2
YUV to AV1	1080p	20	40	0	3	19	100	262	1
YUV to AVC	1080p	20	40	1	1	19	100	500	2
YUV to HEVC	1080p	20	40	1	1	19	99	640	2
YUV to AVC	1080p	20	40	1	2	19	99	500	2
YUV to HEVC	1080p	20	40	1	2	19	100	360	1
YUV to AVC	1080p	20	40	1	3	19	100	500	2
YUV to HEVC	1080p	20	40	1	3	19	99	240	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	0	0	1	23	99	703	2
YUV to HEVC	1080p	20	0	0	1	23	99	820	3
YUV to AV1	1080p	20	0	0	1	23	99	721	2
YUV to AVC	1080p	20	0	0	2	23	99	701	2
YUV to HEVC	1080p	20	0	0	2	23	100	520	1
YUV to AV1	1080p	20	0	0	2	23	99	437	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	100	280	1
YUV to AVC	1080p	20	0	1	1	23	100	502	1
YUV to HEVC	1080p	20	0	1	1	23	99	660	2
YUV to AVC	1080p	20	0	1	2	23	100	500	1
YUV to HEVC	1080p	20	0	1	2	23	99	360	1
YUV to AVC	1080p	20	0	1	3	23	100	504	2
YUV to HEVC	1080p	20	0	1	3	23	100	240	1
YUV to AVC	1080p	20	4	0	1	23	99	700	2
YUV to HEVC	1080p	20	4	0	1	23	99	820	3
YUV to AV1	1080p	20	4	0	1	23	99	720	3
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	100	420	1
YUV to AVC	1080p	20	4	0	3	23	99	700	3
YUV to HEVC	1080p	20	4	0	3	23	100	340	1
YUV to AV1	1080p	20	4	0	3	23	99	269	1
YUV to AVC	1080p	20	4	1	1	23	99	501	2
YUV to HEVC	1080p	20	4	1	1	23	100	660	2
YUV to AVC	1080p	20	4	1	2	23	100	504	2
YUV to HEVC	1080p	20	4	1	2	23	100	360	1
YUV to AVC	1080p	20	4	1	3	23	99	500	2
YUV to HEVC	1080p	20	4	1	3	23	99	240	1
YUV to AVC	1080p	20	16	0	1	23	99	700	2
YUV to HEVC	1080p	20	16	0	1	23	99	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	100	500	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	100	340	1
YUV to AV1	1080p	20	16	0	3	23	99	265	1

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
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	99	500	1
YUV to HEVC	1080p	20	16	1	2	23	99	360	1
YUV to AVC	1080p	20	16	1	3	23	99	500	1
YUV to HEVC	1080p	20	16	1	3	23	99	240	1
YUV to AVC	1080p	20	40	0	1	23	99	691	2
YUV to HEVC	1080p	20	40	0	1	23	99	803	3
YUV to AV1	1080p	20	40	0	1	23	99	701	3
YUV to AVC	1080p	20	40	0	2	23	99	689	2
YUV to HEVC	1080p	20	40	0	2	23	99	500	1
YUV to AV1	1080p	20	40	0	2	23	99	420	1
YUV to AVC	1080p	20	40	0	3	23	99	693	2
YUV to HEVC	1080p	20	40	0	3	23	99	339	1
YUV to AV1	1080p	20	40	0	3	23	100	260	1
YUV to AVC	1080p	20	40	1	1	23	100	500	1
YUV to HEVC	1080p	20	40	1	1	23	100	640	2
YUV to AVC	1080p	20	40	1	2	23	99	500	1
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	100	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	700	2
YUV to HEVC	1080p	20	0	0	2	27	99	520	2
YUV to AV1	1080p	20	0	0	2	27	99	439	1
YUV to AVC	1080p	20	0	0	3	27	99	701	2
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	502	2
YUV to HEVC	1080p	20	0	1	1	27	100	660	2
YUV to AVC	1080p	20	0	1	2	27	100	500	1
YUV to HEVC	1080p	20	0	1	2	27	100	360	1
YUV to AVC	1080p	20	0	1	3	27	99	505	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

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	2	27	100	701	2
YUV to HEVC	1080p	20	4	0	2	27	100	520	1
YUV to AV1	1080p	20	4	0	2	27	99	420	1
YUV to AVC	1080p	20	4	0	3	27	99	700	2
YUV to HEVC	1080p	20	4	0	3	27	99	340	1
YUV to AV1	1080p	20	4	0	3	27	99	272	1
YUV to AVC	1080p	20	4	1	1	27	100	500	1
YUV to HEVC	1080p	20	4	1	1	27	99	660	2
YUV to AVC	1080p	20	4	1	2	27	99	500	2
YUV to HEVC	1080p	20	4	1	2	27	100	360	1
YUV to AVC	1080p	20	4	1	3	27	100	500	2
YUV to HEVC	1080p	20	4	1	3	27	99	240	1
YUV to AVC	1080p	20	16	0	1	27	99	700	2
YUV to HEVC	1080p	20	16	0	1	27	99	818	2
YUV to AV1	1080p	20	16	0	1	27	99	719	2
YUV to AVC	1080p	20	16	0	2	27	99	700	2
YUV to HEVC	1080p	20	16	0	2	27	100	502	2
YUV to AV1	1080p	20	16	0	2	27	100	420	2
YUV to AVC	1080p	20	16	0	3	27	99	700	3
YUV to HEVC	1080p	20	16	0	3	27	99	340	1
YUV to AV1	1080p	20	16	0	3	27	99	261	1
YUV to AVC	1080p	20	16	1	1	27	99	500	2
YUV to HEVC	1080p	20	16	1	1	27	99	645	2
YUV to AVC	1080p	20	16	1	2	27	100	500	2
YUV to HEVC	1080p	20	16	1	2	27	100	360	1
YUV to AVC	1080p	20	16	1	3	27	99	500	1
YUV to HEVC	1080p	20	16	1	3	27	99	240	1
YUV to AVC	1080p	20	40	0	1	27	99	686	2
YUV to HEVC	1080p	20	40	0	1	27	99	801	3
YUV to AV1	1080p	20	40	0	1	27	99	703	2
YUV to AVC	1080p	20	40	0	2	27	99	688	2
YUV to HEVC	1080p	20	40	0	2	27	100	500	2
YUV to AV1	1080p	20	40	0	2	27	99	420	1
YUV to AVC	1080p	20	40	0	3	27	99	689	2
YUV to HEVC	1080p	20	40	0	3	27	100	340	1
YUV to AV1	1080p	20	40	0	3	27	100	261	1
YUV to AVC	1080p	20	40	1	1	27	100	500	1
YUV to HEVC	1080p	20	40	1	1	27	100	640	2
YUV to AVC	1080p	20	40	1	2	27	99	500	2

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	21	5	289	10
AVC to HEVC	1	14	20	5	308	11
AVC to AV1	1	13	21	4	263	10
HEVC to AVC	1	12	20	5	288	14
HEVC to HEVC	1	13	21	5	306	14
HEVC to AV1	1	12	20	4	262	13
VP9 to AVC	1	17	21	5	288	11
VP9 to HEVC	1	18	20	5	306	11
VP9 to AV1	1	15	20	4	262	10
AVC to AVC	16	71	92	24	1024	2
AVC to HEVC	16	80	95	28	1136	3
AVC to AV1	16	73	93	26	1099	3
HEVC to AVC	16	70	92	25	1056	3
HEVC to HEVC	16	78	94	28	1182	4
HEVC to AV1	16	69	93	26	1120	4
VP9 to AVC	16	80	92	24	1024	3
VP9 to HEVC	16	90	93	27	1151	3
VP9 to AV1	16	83	92	25	1104	3
AVC to AVC	32	74	94	24	928	1
AVC to HEVC	32	80	92	27	1032	1
AVC to AV1	32	78	93	26	1024	1
HEVC to AVC	32	71	92	24	960	1
HEVC to HEVC	32	80	93	27	1088	1
HEVC to AV1	32	75	95	27	1056	1
VP9 to AVC	32	81	93	23	960	1
VP9 to HEVC	32	87	92	26	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	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	4k	1	0	8	1	10	0	159	41
HEVC to YUV	4k	1	0	8	1	10	0	160	48
VP9 to YUV	4k	1	0	8	1	24	0	83	28
AVC to YUV	4k	5	0	8	0	84	0	250	16
HEVC to YUV	4k	5	0	8	0	90	0	268	12
VP9 to YUV	4k	5	0	8	0	93	0	267	11
YUV to AVC	4k	1	0	8	1	0	92	184	64
YUV to HEVC	4k	1	0	8	1	0	85	190	70
YUV to AV1	4k	1	0	8	1	0	94	179	66
YUV to AVC	4k	3	0	8	0	0	72	150	41
YUV to HEVC	4k	3	0	8	0	0	73	162	38
YUV to AV1	4k	3	0	8	0	0	73	139	36
YUV to AVC	4k	5	0	8	0	0	97	190	29
YUV to HEVC	4k	5	0	8	0	0	97	214	30
YUV to AV1	4k	5	0	8	0	0	98	185	28
AVC to AVC	4k	1	1	8	1	14	96	126	30
AVC to HEVC	4k	1	1	8	1	14	96	157	43
AVC to AV1	4k	1	1	8	1	14	95	155	40
HEVC to AVC	4k	1	1	8	1	14	97	125	36
HEVC to HEVC	4k	1	1	8	1	17	96	156	48
HEVC to AV1	4k	1	1	8	1	12	88	151	44
VP9 to AVC	4k	1	1	8	1	24	39	83	23
VP9 to HEVC	4k	1	1	8	1	24	38	83	23
VP9 to AV1	4k	1	1	8	1	24	44	83	23
AVC to AVC	4k	3	1	8	0	48	71	123	30
AVC to HEVC	4k	3	1	8	0	49	72	147	32
AVC to AV1	4k	3	1	8	0	40	72	135	26
HEVC to AVC	4k	3	1	8	0	41	72	120	31
HEVC to HEVC	4k	3	1	8	0	42	72	150	34
HEVC to AV1	4k	3	1	8	0	34	72	135	24
VP9 to AVC	4k	3	1	8	0	45	72	120	29
VP9 to HEVC	4k	3	1	8	0	48	72	149	28
VP9 to AV1	4k	3	1	8	0	40	72	135	20
AVC to AVC	4k	5	1	8	0	66	97	120	22
AVC to HEVC	4k	5	1	8	0	71	97	159	26
AVC to AV1	4k	5	1	8	0	65	96	157	23
HEVC to AVC	4k	5	1	8	0	56	97	121	18
HEVC to HEVC	4k	5	1	8	0	60	96	159	20
HEVC to AV1	4k	5	1	8	0	52	97	163	22

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AVC	4k	5	1	8	0	61	97	122	17
VP9 to HEVC	4k	5	1	8	0	67	98	159	26
VP9 to AV1	4k	5	1	8	0	62	96	160	23
AVC to YUV	4k	1	0	10	1	8	0	103	29
HEVC to YUV	4k	1	0	10	1	8	0	103	29
VP9 to YUV	4k	1	0	10	1	24	0	83	28
AVC to YUV	4k	5	0	10	0	50	0	135	9
HEVC to YUV	4k	5	0	10	0	66	0	134	9
VP9 to YUV	4k	5	0	10	0	63	0	135	8
YUV to AVC	4k	1	0	10	1	0	51	107	59
YUV to HEVC	4k	1	0	10	1	0	49	109	49
YUV to AV1	4k	1	0	10	1	0	55	106	48
YUV to AVC	4k	3	0	10	0	0	61	116	54
YUV to HEVC	4k	3	0	10	0	0	53	117	57
YUV to AV1	4k	3	0	10	0	0	61	118	50
AVC to YUV	1080p	1	0	8	1	17	0	426	58
HEVC to YUV	1080p	1	0	8	1	15	0	409	80
VP9 to YUV	1080p	1	0	8	1	23	0	210	32
AVC to YUV	1080p	18	0	8	0	93	0	962	9
HEVC to YUV	1080p	18	0	8	0	92	0	1005	12
VP9 to YUV	1080p	18	0	8	0	99	0	877	9
YUV to AVC	1080p	1	0	8	1	0	52	445	82
YUV to HEVC	1080p	1	0	8	1	0	53	453	72
YUV to AV1	1080p	1	0	8	1	0	53	402	76
YUV to AVC	1080p	20	0	8	0	0	99	818	12
YUV to HEVC	1080p	20	0	8	0	0	99	860	14
YUV to AV1	1080p	20	0	8	0	0	99	743	12
AVC to AVC	1080p	1	1	8	1	34	71	543	63
AVC to HEVC	1080p	1	1	8	1	37	72	590	70
AVC to AV1	1080p	1	1	8	1	29	74	542	80
HEVC to AVC	1080p	1	1	8	1	32	74	525	76
HEVC to HEVC	1080p	1	1	8	1	30	72	579	82
HEVC to AV1	1080p	1	1	8	1	28	81	527	76
VP9 to AVC	1080p	1	1	8	1	23	25	209	28
VP9 to HEVC	1080p	1	1	8	1	22	23	209	32
VP9 to AV1	1080p	1	1	8	1	23	28	209	32
AVC to AVC	1080p	20	1	8	0	73	99	571	10
AVC to HEVC	1080p	20	1	8	0	80	99	654	11
AVC to AV1	1080p	20	1	8	0	74	99	640	11
HEVC to AVC	1080p	20	1	8	0	69	100	601	10

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to HEVC	1080p	20	1	8	0	72	99	684	12
HEVC to AV1	1080p	20	1	8	0	67	100	660	6
VP9 to AVC	1080p	20	1	8	0	80	100	580	9
VP9 to HEVC	1080p	20	1	8	0	87	99	660	10
VP9 to AV1	1080p	20	1	8	0	82	100	651	10
AVC to YUV	1080p	1	0	10	1	15	0	318	39
HEVC to YUV	1080p	1	0	10	1	13	0	313	39
VP9 to YUV	1080p	1	0	10	1	21	0	300	37
AVC to YUV	1080p	18	0	10	0	55	0	544	4
HEVC to YUV	1080p	18	0	10	0	61	0	541	5
VP9 to YUV	1080p	18	0	10	0	53	0	540	4
YUV to AVC	1080p	1	0	10	1	0	36	302	71
YUV to HEVC	1080p	1	0	10	1	0	34	303	74
YUV to AV1	1080p	1	0	10	1	0	38	282	60
YUV to AVC	1080p	20	0	10	0	0	57	462	13
YUV to HEVC	1080p	20	0	10	0	0	53	465	13
YUV to AV1	1080p	20	0	10	0	0	60	460	14
AVC to YUV	720p	1	0	8	1	23	0	644	61
HEVC to YUV	720p	1	0	8	1	21	0	630	75
VP9 to YUV	720p	1	0	8	1	21	0	372	57
AVC to YUV	720p	40	0	8	0	98	0	1896	5
HEVC to YUV	720p	40	0	8	0	90	0	1894	7
VP9 to YUV	720p	40	0	8	0	94	0	1613	4
YUV to AVC	720p	1	0	8	1	0	29	545	60
YUV to HEVC	720p	1	0	8	1	0	30	553	63
YUV to AV1	720p	1	0	8	1	0	31	498	87
YUV to AVC	720p	40	0	8	0	0	88	1612	8
YUV to HEVC	720p	40	0	8	0	0	91	1679	8
YUV to AV1	720p	40	0	8	0	0	91	1474	7
AVC to AVC	720p	1	1	8	1	37	46	850	65
AVC to HEVC	720p	1	1	8	1	38	48	873	59
AVC to AV1	720p	1	1	8	1	31	49	766	73
HEVC to AVC	720p	1	1	8	1	33	45	826	68
HEVC to HEVC	720p	1	1	8	1	30	43	835	87
HEVC to AV1	720p	1	1	8	1	28	49	734	70
VP9 to AVC	720p	1	1	8	1	21	19	367	44
VP9 to HEVC	720p	1	1	8	1	21	20	366	40
VP9 to AV1	720p	1	1	8	1	20	23	365	49
AVC to AVC	720p	40	1	8	0	90	100	1288	3
AVC to HEVC	720p	40	1	8	0	92	100	1410	4

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	720p	40	1	8	0	75	100	1288	3
HEVC to AVC	720p	40	1	8	0	83	100	1291	5
HEVC to HEVC	720p	40	1	8	0	85	100	1417	5
HEVC to AV1	720p	40	1	8	0	69	100	1290	3
VP9 to AVC	720p	40	1	8	0	100	97	1256	4
VP9 to HEVC	720p	40	1	8	0	100	94	1363	5
VP9 to AV1	720p	40	1	8	0	90	100	1286	5

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

28.1 Decoding

28.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

28.1.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

28.2 Encoding

28.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

28.2.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

28.3 Transcoding

28.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

28.3.2 Command line

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

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

28.4 Libxcode Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	4k	1	0	8	1	10	0	156	32
HEVC to YUV	4k	1	0	8	1	9	0	160	26
VP9 to YUV	4k	1	0	8	1	24	0	82	16
AVC to YUV	4k	5	0	8	0	82	0	249	11
HEVC to YUV	4k	5	0	8	0	87	0	262	9
VP9 to YUV	4k	5	0	8	0	90	0	262	8
YUV to AVC	4k	1	0	8	1	0	55	119	48
YUV to HEVC	4k	1	0	8	1	0	56	125	47
YUV to AV1	4k	1	0	8	1	0	59	108	49
YUV to AVC	4k	3	0	8	0	0	72	151	32
YUV to HEVC	4k	3	0	8	0	0	73	161	33
YUV to AV1	4k	3	0	8	0	0	72	140	30
YUV to AVC	4k	5	0	8	0	0	98	186	26
YUV to HEVC	4k	5	0	8	0	0	97	215	27
YUV to AV1	4k	5	0	8	0	0	97	187	26
AVC to AVC	4k	1	1	8	1	16	96	128	49
AVC to HEVC	4k	1	1	8	1	16	96	163	55
AVC to AV1	4k	1	1	8	1	15	96	159	54
HEVC to AVC	4k	1	1	8	1	19	97	128	45
HEVC to HEVC	4k	1	1	8	1	16	97	159	50
HEVC to AV1	4k	1	1	8	1	15	97	151	50
VP9 to AVC	4k	1	1	8	1	24	38	82	42
VP9 to HEVC	4k	1	1	8	1	24	36	82	40
VP9 to AV1	4k	1	1	8	1	24	44	82	42
AVC to AVC	4k	3	1	8	0	48	72	122	38
AVC to HEVC	4k	3	1	8	0	47	72	153	41
AVC to AV1	4k	3	1	8	0	40	72	137	46
HEVC to AVC	4k	3	1	8	0	40	72	119	40
HEVC to HEVC	4k	3	1	8	0	42	72	145	24
HEVC to AV1	4k	3	1	8	0	35	72	133	34
VP9 to AVC	4k	3	1	8	0	45	73	122	30
VP9 to HEVC	4k	3	1	8	0	49	73	148	27
VP9 to AV1	4k	3	1	8	0	40	73	135	24
AVC to AVC	4k	5	1	8	0	66	97	122	12
AVC to HEVC	4k	5	1	8	0	69	97	159	11
AVC to AV1	4k	5	1	8	0	64	96	159	10
HEVC to AVC	4k	5	1	8	0	56	97	123	12
HEVC to HEVC	4k	5	1	8	0	59	96	158	11

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to AV1	4k	5	1	8	0	55	96	157	11
VP9 to AVC	4k	5	1	8	0	60	99	123	29
VP9 to HEVC	4k	5	1	8	0	67	96	160	10
VP9 to AV1	4k	5	1	8	0	62	97	159	11
AVC to YUV	4k	1	0	10	1	8	0	96	24
HEVC to YUV	4k	1	0	10	1	7	0	96	22
VP9 to YUV	4k	1	0	10	1	24	0	82	21
AVC to YUV	4k	5	0	10	0	53	0	133	7
HEVC to YUV	4k	5	0	10	0	63	0	134	7
VP9 to YUV	4k	5	0	10	0	65	0	134	7
YUV to AVC	4k	1	0	10	1	0	32	66	47
YUV to HEVC	4k	1	0	10	1	0	29	68	47
YUV to AV1	4k	1	0	10	1	0	34	64	47
YUV to AVC	4k	3	0	10	0	0	39	88	39
YUV to HEVC	4k	3	0	10	0	0	42	89	39
YUV to AV1	4k	3	0	10	0	0	45	83	37
AVC to YUV	1080p	1	0	8	1	16	0	385	39
HEVC to YUV	1080p	1	0	8	1	16	0	422	35
VP9 to YUV	1080p	1	0	8	1	23	0	211	19
AVC to YUV	1080p	20	0	8	0	87	0	929	6
HEVC to YUV	1080p	20	0	8	0	89	0	966	5
VP9 to YUV	1080p	20	0	8	0	98	0	889	3
YUV to AVC	1080p	1	0	8	1	0	38	322	44
YUV to HEVC	1080p	1	0	8	1	0	39	326	45
YUV to AV1	1080p	1	0	8	1	0	36	292	41
YUV to AVC	1080p	20	0	8	0	0	99	827	10
YUV to HEVC	1080p	20	0	8	0	0	99	881	11
YUV to AV1	1080p	20	0	8	0	0	100	763	10
AVC to AVC	1080p	1	1	8	1	40	79	591	68
AVC to HEVC	1080p	1	1	8	1	34	72	628	72
AVC to AV1	1080p	1	1	8	1	35	84	596	66
HEVC to AVC	1080p	1	1	8	1	34	83	589	67
HEVC to HEVC	1080p	1	1	8	1	33	80	644	66
HEVC to AV1	1080p	1	1	8	1	29	84	596	63
VP9 to AVC	1080p	1	1	8	1	23	25	210	33
VP9 to HEVC	1080p	1	1	8	1	23	24	211	35
VP9 to AV1	1080p	1	1	8	1	23	28	211	41
AVC to AVC	1080p	20	1	8	0	72	99	585	5
AVC to HEVC	1080p	20	1	8	0	80	99	667	6

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to AV1	1080p	20	1	8	0	72	99	651	4
HEVC to AVC	1080p	20	1	8	0	68	99	611	5
HEVC to HEVC	1080p	20	1	8	0	70	99	697	7
HEVC to AV1	1080p	20	1	8	0	66	99	667	5
VP9 to AVC	1080p	20	1	8	0	78	99	590	5
VP9 to HEVC	1080p	20	1	8	0	86	99	677	6
VP9 to AV1	1080p	20	1	8	0	82	99	662	3
AVC to YUV	1080p	1	0	10	1	16	0	321	28
HEVC to YUV	1080p	1	0	10	1	14	0	313	25
VP9 to YUV	1080p	1	0	10	1	21	0	297	26
AVC to YUV	1080p	20	0	10	0	50	0	539	2
HEVC to YUV	1080p	20	0	10	0	61	0	536	2
VP9 to YUV	1080p	20	0	10	0	100	0	522	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	25	193	45
YUV to AVC	1080p	20	0	10	0	0	57	466	12
YUV to HEVC	1080p	20	0	10	0	0	52	472	12
YUV to AV1	1080p	20	0	10	0	0	59	450	12
AVC to YUV	720p	1	0	8	1	22	0	590	37
HEVC to YUV	720p	1	0	8	1	21	0	635	33
VP9 to YUV	720p	1	0	8	1	21	0	377	20
AVC to YUV	720p	40	0	8	0	96	0	1905	3
HEVC to YUV	720p	40	0	8	0	80	0	1737	2
VP9 to YUV	720p	40	0	8	0	88	0	1552	1
YUV to AVC	720p	1	0	8	1	0	27	522	29
YUV to HEVC	720p	1	0	8	1	0	27	520	31
YUV to AV1	720p	1	0	8	1	0	24	407	44
YUV to AVC	720p	40	0	8	0	0	93	1763	6
YUV to HEVC	720p	40	0	8	0	0	95	1828	6
YUV to AV1	720p	40	0	8	0	0	96	1602	5
AVC to AVC	720p	1	1	8	1	34	45	859	65
AVC to HEVC	720p	1	1	8	1	35	47	935	56
AVC to AV1	720p	1	1	8	1	30	49	844	61
HEVC to AVC	720p	1	1	8	1	43	57	969	31
HEVC to HEVC	720p	1	1	8	1	32	47	928	47
HEVC to AV1	720p	1	1	8	1	29	52	799	68
VP9 to AVC	720p	1	1	8	1	21	20	376	48
VP9 to HEVC	720p	1	1	8	1	21	20	376	48

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	720p	1	1	8	1	21	24	376	49
AVC to AVC	720p	40	1	8	0	100	100	1340	0
AVC to HEVC	720p	40	1	8	0	100	100	1468	1
AVC to AV1	720p	40	1	8	0	97	100	1328	1
HEVC to AVC	720p	40	1	8	0	100	100	1342	0
HEVC to HEVC	720p	40	1	8	0	100	100	1470	0
HEVC to AV1	720p	40	1	8	0	94	100	1330	1
VP9 to AVC	720p	40	1	8	0	100	100	1297	1
VP9 to HEVC	720p	40	1	8	0	100	99	1406	1
VP9 to AV1	720p	40	1	8	0	100	100	1334	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	24.80	28.34	24.11	0.17
YUV to HEVC	4k	1	26.75	30.03	25.27	0.24
YUV to AV1	4k	1	34.71	40.93	26.09	1.61
YUV to AVC	4k	3	26.63	31.13	24.14	1.59
YUV to HEVC	4k	3	27.68	33.70	25.37	0.85
YUV to AV1	4k	3	35.01	41.64	25.73	1.66
YUV to AVC	4k	5	28.13	31.37	24.88	1.88
YUV to HEVC	4k	5	33.69	58.58	25.28	66.53
YUV to AV1	4k	5	44.04	74.39	26.37	132.05
YUV to AVC	1080p	1	7.64	9.27	7.10	0.05
YUV to HEVC	1080p	1	8.16	9.09	7.36	0.07
YUV to AV1	1080p	1	10.73	12.08	8.03	0.13
YUV to AVC	1080p	20	10.67	19.38	7.31	3.72
YUV to HEVC	1080p	20	12.09	21.91	7.78	7.03
YUV to AV1	1080p	20	39.99	50.17	27.52	13.68
YUV to AVC	720p	1	4.33	5.39	3.71	0.05
YUV to HEVC	720p	1	4.59	5.31	4.05	0.03
YUV to AV1	720p	1	5.94	6.58	4.90	0.05
YUV to AVC	720p	40	6.20	22.09	4.26	1.34
YUV to HEVC	720p	40	6.45	11.22	4.38	1.41
YUV to AV1	720p	40	38.90	47.29	32.22	1.75

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	83	24
AVC to RGBA	4k	8	91	1	340	18
HEVC to RGBA	4k	1	22	0	100	25
HEVC to RGBA	4k	8	94	2	409	22
VP9 to RGBA	4k	1	22	0	83	20
VP9 to RGBA	4k	8	94	1	335	17
AVC to RGBA	1080p	20	92	9	1196	7
HEVC to RGBA	1080p	20	93	8	1173	8
VP9 to RGBA	1080p	20	92	6	900	6
AVC to RGBA	720p	50	96	19	2211	4
HEVC to RGBA	720p	50	89	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	35	94	4	250	29
AVC to HEVC	5	33	93	3	280	33
AVC to AV1	5	25	92	2	250	32
HEVC to AVC	5	36	94	4	255	32
HEVC to HEVC	5	33	93	3	285	37
HEVC to AV1	5	26	92	2	255	35
VP9 to AVC	5	40	95	4	250	29
VP9 to HEVC	5	39	94	3	280	32
VP9 to AV1	5	31	93	2	253	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	60
RGBA to HEVC	4k	1	1	37	81	56
RGBA to AV1	4k	1	1	42	80	60
RGBA to AVC	4k	3	0	44	88	52
RGBA to HEVC	4k	3	0	40	90	49
RGBA to AV1	4k	3	0	46	89	54
RGBA to AVC	4k	5	0	54	90	38
RGBA to HEVC	4k	5	0	44	90	36
RGBA to AV1	4k	5	0	48	90	40
RGBA to AVC	1080p	1	1	29	244	84
RGBA to HEVC	1080p	1	1	27	245	82
RGBA to AV1	1080p	1	1	31	237	88
RGBA to AVC	1080p	10	0	43	334	25
RGBA to HEVC	1080p	10	0	40	338	26
RGBA to AV1	1080p	10	0	45	335	26
RGBA to AVC	1080p	20	0	45	336	13
RGBA to HEVC	1080p	20	0	41	340	13
RGBA to AV1	1080p	20	0	46	340	14
RGBA to AVC	720p	1	1	21	362	104
RGBA to HEVC	720p	1	1	20	366	108
RGBA to AV1	720p	1	1	23	400	111
RGBA to AVC	720p	10	0	38	687	44
RGBA to HEVC	720p	10	0	37	698	47
RGBA to AV1	720p	10	0	42	677	47
RGBA to AVC	720p	20	0	40	723	26
RGBA to HEVC	720p	20	0	40	733	24
RGBA to AV1	720p	20	0	44	719	22

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	38
YUV to HEVC	4k	2	0	0	1	48	108	43
YUV to AV1	4k	2	0	0	1	48	94	36
YUV to AVC	4k	2	0	0	2	49	102	38
YUV to HEVC	4k	2	0	0	2	48	54	25
YUV to AV1	4k	2	0	0	2	49	44	23
YUV to AVC	4k	2	0	0	3	49	102	41
YUV to HEVC	4k	2	0	0	3	49	32	22
YUV to AV1	4k	2	0	0	3	49	24	18
YUV to AVC	4k	2	0	1	1	49	60	28
YUV to HEVC	4k	2	0	1	1	49	76	35
YUV to AVC	4k	2	0	1	2	48	60	31
YUV to HEVC	4k	2	0	1	2	48	34	18
YUV to AVC	4k	2	0	1	3	48	60	30
YUV to HEVC	4k	2	0	1	3	49	22	17
YUV to AVC	4k	2	4	0	1	78	100	44
YUV to HEVC	4k	2	4	0	1	74	108	42
YUV to AV1	4k	2	4	0	1	73	94	44
YUV to AVC	4k	2	4	0	2	77	100	48
YUV to HEVC	4k	2	4	0	2	62	54	29
YUV to AV1	4k	2	4	0	2	61	44	25
YUV to AVC	4k	2	4	0	3	78	100	43
YUV to HEVC	4k	2	4	0	3	56	32	20
YUV to AV1	4k	2	4	0	3	55	26	18
YUV to AVC	4k	2	4	1	1	66	60	30
YUV to HEVC	4k	2	4	1	1	67	76	40
YUV to AVC	4k	2	4	1	2	66	60	34
YUV to HEVC	4k	2	4	1	2	56	34	21
YUV to AVC	4k	2	4	1	3	67	60	31
YUV to HEVC	4k	2	4	1	3	54	22	15
YUV to AVC	4k	2	16	0	1	79	100	38
YUV to HEVC	4k	2	16	0	1	74	108	44
YUV to AV1	4k	2	16	0	1	72	94	38
YUV to AVC	4k	2	16	0	2	80	100	38
YUV to HEVC	4k	2	16	0	2	61	54	27
YUV to AV1	4k	2	16	0	2	60	44	22
YUV to AVC	4k	2	16	0	3	80	100	42
YUV to HEVC	4k	2	16	0	3	57	32	19
YUV to AV1	4k	2	16	0	3	58	26	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	16	1	1	67	60	27
YUV to HEVC	4k	2	16	1	1	67	76	35
YUV to AVC	4k	2	16	1	2	66	60	31
YUV to HEVC	4k	2	16	1	2	56	34	19
YUV to AVC	4k	2	16	1	3	67	60	27
YUV to HEVC	4k	2	16	1	3	54	22	16
YUV to AVC	4k	2	40	0	1	78	98	39
YUV to HEVC	4k	2	40	0	1	74	106	44
YUV to AV1	4k	2	40	0	1	74	92	40
YUV to AVC	4k	2	40	0	2	77	98	41
YUV to HEVC	4k	2	40	0	2	61	54	28
YUV to AV1	4k	2	40	0	2	61	44	24
YUV to AVC	4k	2	40	0	3	78	98	35
YUV to HEVC	4k	2	40	0	3	57	32	20
YUV to AV1	4k	2	40	0	3	57	26	18
YUV to AVC	4k	2	40	1	1	67	58	30
YUV to HEVC	4k	2	40	1	1	67	76	36
YUV to AVC	4k	2	40	1	2	66	60	29
YUV to HEVC	4k	2	40	1	2	56	34	20
YUV to AVC	4k	2	40	1	3	67	60	28
YUV to HEVC	4k	2	40	1	3	56	22	19
YUV to AVC	1080p	10	0	0	1	99	811	25
YUV to HEVC	1080p	10	0	0	1	94	827	26
YUV to AV1	1080p	10	0	0	1	100	749	23
YUV to AVC	1080p	10	0	0	2	98	806	25
YUV to HEVC	1080p	10	0	0	2	100	440	21
YUV to AV1	1080p	10	0	0	2	100	356	19
YUV to AVC	1080p	10	0	0	3	99	807	24
YUV to HEVC	1080p	10	0	0	3	99	261	17
YUV to AV1	1080p	10	0	0	3	99	188	16
YUV to AVC	1080p	10	0	1	1	99	480	21
YUV to HEVC	1080p	10	0	1	1	99	614	22
YUV to AVC	1080p	10	0	1	2	100	481	21
YUV to HEVC	1080p	10	0	1	2	100	281	17
YUV to AVC	1080p	10	0	1	3	99	481	20
YUV to HEVC	1080p	10	0	1	3	100	177	16
YUV to AVC	1080p	10	4	0	1	99	450	20
YUV to HEVC	1080p	10	4	0	1	99	525	21
YUV to AV1	1080p	10	4	0	1	99	460	18
YUV to AVC	1080p	10	4	0	2	99	449	19

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	4	0	2	99	330	18
YUV to AV1	1080p	10	4	0	2	99	270	17
YUV to AVC	1080p	10	4	0	3	100	450	19
YUV to HEVC	1080p	10	4	0	3	99	217	16
YUV to AV1	1080p	10	4	0	3	100	171	15
YUV to AVC	1080p	10	4	1	1	99	322	18
YUV to HEVC	1080p	10	4	1	1	99	420	19
YUV to AVC	1080p	10	4	1	2	99	324	17
YUV to HEVC	1080p	10	4	1	2	99	230	17
YUV to AVC	1080p	10	4	1	3	99	324	19
YUV to HEVC	1080p	10	4	1	3	100	155	17
YUV to AVC	1080p	10	16	0	1	99	448	20
YUV to HEVC	1080p	10	16	0	1	99	520	21
YUV to AV1	1080p	10	16	0	1	99	460	20
YUV to AVC	1080p	10	16	0	2	99	446	20
YUV to HEVC	1080p	10	16	0	2	100	326	17
YUV to AV1	1080p	10	16	0	2	100	270	17
YUV to AVC	1080p	10	16	0	3	99	448	18
YUV to HEVC	1080p	10	16	0	3	99	215	16
YUV to AV1	1080p	10	16	0	3	99	171	15
YUV to AVC	1080p	10	16	1	1	99	321	19
YUV to HEVC	1080p	10	16	1	1	100	419	18
YUV to AVC	1080p	10	16	1	2	99	320	19
YUV to HEVC	1080p	10	16	1	2	99	230	16
YUV to AVC	1080p	10	16	1	3	99	320	18
YUV to HEVC	1080p	10	16	1	3	100	155	16
YUV to AVC	1080p	10	40	0	1	99	440	21
YUV to HEVC	1080p	10	40	0	1	99	519	20
YUV to AV1	1080p	10	40	0	1	99	453	19
YUV to AVC	1080p	10	40	0	2	99	442	18
YUV to HEVC	1080p	10	40	0	2	99	320	17
YUV to AV1	1080p	10	40	0	2	100	270	17
YUV to AVC	1080p	10	40	0	3	99	443	20
YUV to HEVC	1080p	10	40	0	3	99	212	17
YUV to AV1	1080p	10	40	0	3	100	171	15
YUV to AVC	1080p	10	40	1	1	99	320	20
YUV to HEVC	1080p	10	40	1	1	100	411	20
YUV to AVC	1080p	10	40	1	2	99	320	18
YUV to HEVC	1080p	10	40	1	2	99	230	16
YUV to AVC	1080p	10	40	1	3	99	320	18

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	1	3	100	152	16
YUV to AVC	720p	20	0	0	1	82	1504	14
YUV to HEVC	720p	20	0	0	1	82	1552	15
YUV to AV1	720p	20	0	0	1	86	1394	12
YUV to AVC	720p	20	0	0	2	85	1543	14
YUV to HEVC	720p	20	0	0	2	99	960	13
YUV to AV1	720p	20	0	0	2	99	780	11
YUV to AVC	720p	20	0	0	3	83	1521	14
YUV to HEVC	720p	20	0	0	3	99	580	9
YUV to AV1	720p	20	0	0	3	99	420	6
YUV to AVC	720p	20	0	1	1	99	1080	10
YUV to HEVC	720p	20	0	1	1	99	1324	15
YUV to AVC	720p	20	0	1	2	99	1080	15
YUV to HEVC	720p	20	0	1	2	100	620	9
YUV to AVC	720p	20	0	1	3	99	1080	11
YUV to HEVC	720p	20	0	1	3	99	400	5
YUV to AVC	720p	20	4	0	1	99	922	14
YUV to HEVC	720p	20	4	0	1	99	1060	12
YUV to AV1	720p	20	4	0	1	100	870	7
YUV to AVC	720p	20	4	0	2	99	922	14
YUV to HEVC	720p	20	4	0	2	99	682	13
YUV to AV1	720p	20	4	0	2	99	560	8
YUV to AVC	720p	20	4	0	3	99	920	14
YUV to HEVC	720p	20	4	0	3	99	460	8
YUV to AV1	720p	20	4	0	3	99	360	7
YUV to AVC	720p	20	4	1	1	99	680	8
YUV to HEVC	720p	20	4	1	1	99	860	14
YUV to AVC	720p	20	4	1	2	99	680	11
YUV to HEVC	720p	20	4	1	2	99	480	7
YUV to AVC	720p	20	4	1	3	99	680	13
YUV to HEVC	720p	20	4	1	3	99	340	9
YUV to AVC	720p	20	16	0	1	99	919	12
YUV to HEVC	720p	20	16	0	1	99	1060	13
YUV to AV1	720p	20	16	0	1	100	903	7
YUV to AVC	720p	20	16	0	2	99	920	12
YUV to HEVC	720p	20	16	0	2	99	680	11
YUV to AV1	720p	20	16	0	2	99	560	8
YUV to AVC	720p	20	16	0	3	99	920	14
YUV to HEVC	720p	20	16	0	3	99	461	7
YUV to AV1	720p	20	16	0	3	99	360	6

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	20	16	1	1	99	680	12
YUV to HEVC	720p	20	16	1	1	99	859	11
YUV to AVC	720p	20	16	1	2	99	680	14
YUV to HEVC	720p	20	16	1	2	99	480	7
YUV to AVC	720p	20	16	1	3	99	680	11
YUV to HEVC	720p	20	16	1	3	99	340	6
YUV to AVC	720p	20	40	0	1	99	915	14
YUV to HEVC	720p	20	40	0	1	99	1042	13
YUV to AV1	720p	20	40	0	1	99	888	7
YUV to AVC	720p	20	40	0	2	99	916	13
YUV to HEVC	720p	20	40	0	2	99	680	11
YUV to AV1	720p	20	40	0	2	100	560	8
YUV to AVC	720p	20	40	0	3	99	915	14
YUV to HEVC	720p	20	40	0	3	99	460	8
YUV to AV1	720p	20	40	0	3	100	360	6
YUV to AVC	720p	20	40	1	1	99	673	12
YUV to HEVC	720p	20	40	1	1	99	840	10
YUV to AVC	720p	20	40	1	2	99	677	12
YUV to HEVC	720p	20	40	1	2	100	480	7
YUV to AVC	720p	20	40	1	3	99	674	11
YUV to HEVC	720p	20	40	1	3	99	340	7

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	100	450	19
YUV to HEVC	1080p	10	0	0	1	19	99	523	20
YUV to AV1	1080p	10	0	0	1	19	99	460	19
YUV to AVC	1080p	10	0	0	2	19	99	450	19
YUV to HEVC	1080p	10	0	0	2	19	99	330	18
YUV to AV1	1080p	10	0	0	2	19	99	279	18
YUV to AVC	1080p	10	0	0	3	19	99	451	19
YUV to HEVC	1080p	10	0	0	3	19	100	218	18
YUV to AV1	1080p	10	0	0	3	19	99	175	16
YUV to AVC	1080p	10	0	1	1	19	99	327	18
YUV to HEVC	1080p	10	0	1	1	19	99	420	20
YUV to AVC	1080p	10	0	1	2	19	99	325	19
YUV to HEVC	1080p	10	0	1	2	19	100	230	17
YUV to AVC	1080p	10	0	1	3	19	99	325	19
YUV to HEVC	1080p	10	0	1	3	19	99	156	16
YUV to AVC	1080p	10	4	0	1	19	99	450	19
YUV to HEVC	1080p	10	4	0	1	19	99	523	21
YUV to AV1	1080p	10	4	0	1	19	99	459	19
YUV to AVC	1080p	10	4	0	2	19	99	450	20
YUV to HEVC	1080p	10	4	0	2	19	99	330	18
YUV to AV1	1080p	10	4	0	2	19	100	270	16
YUV to AVC	1080p	10	4	0	3	19	99	450	19
YUV to HEVC	1080p	10	4	0	3	19	99	217	17
YUV to AV1	1080p	10	4	0	3	19	100	172	15
YUV to AVC	1080p	10	4	1	1	19	99	324	18
YUV to HEVC	1080p	10	4	1	1	19	99	420	19
YUV to AVC	1080p	10	4	1	2	19	99	324	18
YUV to HEVC	1080p	10	4	1	2	19	99	230	18
YUV to AVC	1080p	10	4	1	3	19	99	322	18
YUV to HEVC	1080p	10	4	1	3	19	99	156	16
YUV to AVC	1080p	10	16	0	1	19	99	448	18
YUV to HEVC	1080p	10	16	0	1	19	99	522	22
YUV to AV1	1080p	10	16	0	1	19	99	460	18
YUV to AVC	1080p	10	16	0	2	19	99	447	18
YUV to HEVC	1080p	10	16	0	2	19	100	328	18
YUV to AV1	1080p	10	16	0	2	19	100	270	19
YUV to AVC	1080p	10	16	0	3	19	99	448	20
YUV to HEVC	1080p	10	16	0	3	19	99	216	17

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	3	19	99	170	15
YUV to AVC	1080p	10	16	1	1	19	99	322	18
YUV to HEVC	1080p	10	16	1	1	19	99	417	20
YUV to AVC	1080p	10	16	1	2	19	99	321	16
YUV to HEVC	1080p	10	16	1	2	19	100	230	17
YUV to AVC	1080p	10	16	1	3	19	99	320	19
YUV to HEVC	1080p	10	16	1	3	19	100	155	16
YUV to AVC	1080p	10	40	0	1	19	99	442	19
YUV to HEVC	1080p	10	40	0	1	19	99	518	20
YUV to AV1	1080p	10	40	0	1	19	99	451	16
YUV to AVC	1080p	10	40	0	2	19	99	441	19
YUV to HEVC	1080p	10	40	0	2	19	99	320	18
YUV to AV1	1080p	10	40	0	2	19	100	270	17
YUV to AVC	1080p	10	40	0	3	19	99	440	19
YUV to HEVC	1080p	10	40	0	3	19	99	213	18
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	410	17
YUV to AVC	1080p	10	40	1	2	19	99	320	18
YUV to HEVC	1080p	10	40	1	2	19	99	230	16
YUV to AVC	1080p	10	40	1	3	19	99	320	16
YUV to HEVC	1080p	10	40	1	3	19	100	153	15
YUV to AVC	1080p	10	0	0	1	23	99	450	20
YUV to HEVC	1080p	10	0	0	1	23	99	524	22
YUV to AV1	1080p	10	0	0	1	23	99	460	19
YUV to AVC	1080p	10	0	0	2	23	99	450	19
YUV to HEVC	1080p	10	0	0	2	23	100	330	18
YUV to AV1	1080p	10	0	0	2	23	99	279	17
YUV to AVC	1080p	10	0	0	3	23	99	451	19
YUV to HEVC	1080p	10	0	0	3	23	99	216	16
YUV to AV1	1080p	10	0	0	3	23	99	177	15
YUV to AVC	1080p	10	0	1	1	23	99	326	19
YUV to HEVC	1080p	10	0	1	1	23	99	420	21
YUV to AVC	1080p	10	0	1	2	23	99	325	18
YUV to HEVC	1080p	10	0	1	2	23	99	230	17
YUV to AVC	1080p	10	0	1	3	23	99	327	18
YUV to HEVC	1080p	10	0	1	3	23	100	157	15
YUV to AVC	1080p	10	4	0	1	23	99	449	19
YUV to HEVC	1080p	10	4	0	1	23	99	522	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	4	0	1	23	99	460	17
YUV to AVC	1080p	10	4	0	2	23	99	450	19
YUV to HEVC	1080p	10	4	0	2	23	99	330	17
YUV to AV1	1080p	10	4	0	2	23	99	270	18
YUV to AVC	1080p	10	4	0	3	23	99	449	19
YUV to HEVC	1080p	10	4	0	3	23	100	219	17
YUV to AV1	1080p	10	4	0	3	23	100	171	15
YUV to AVC	1080p	10	4	1	1	23	100	325	17
YUV to HEVC	1080p	10	4	1	1	23	99	420	18
YUV to AVC	1080p	10	4	1	2	23	100	326	18
YUV to HEVC	1080p	10	4	1	2	23	100	230	16
YUV to AVC	1080p	10	4	1	3	23	99	324	19
YUV to HEVC	1080p	10	4	1	3	23	99	158	17
YUV to AVC	1080p	10	16	0	1	23	99	448	19
YUV to HEVC	1080p	10	16	0	1	23	99	521	21
YUV to AV1	1080p	10	16	0	1	23	99	460	20
YUV to AVC	1080p	10	16	0	2	23	99	450	18
YUV to HEVC	1080p	10	16	0	2	23	99	326	18
YUV to AV1	1080p	10	16	0	2	23	99	270	16
YUV to AVC	1080p	10	16	0	3	23	99	448	19
YUV to HEVC	1080p	10	16	0	3	23	99	214	15
YUV to AV1	1080p	10	16	0	3	23	99	170	15
YUV to AVC	1080p	10	16	1	1	23	99	321	19
YUV to HEVC	1080p	10	16	1	1	23	99	419	18
YUV to AVC	1080p	10	16	1	2	23	100	321	19
YUV to HEVC	1080p	10	16	1	2	23	99	230	17
YUV to AVC	1080p	10	16	1	3	23	100	320	17
YUV to HEVC	1080p	10	16	1	3	23	100	155	15
YUV to AVC	1080p	10	40	0	1	23	99	443	18
YUV to HEVC	1080p	10	40	0	1	23	99	519	20
YUV to AV1	1080p	10	40	0	1	23	99	454	19
YUV to AVC	1080p	10	40	0	2	23	99	444	19
YUV to HEVC	1080p	10	40	0	2	23	99	321	19
YUV to AV1	1080p	10	40	0	2	23	99	270	17
YUV to AVC	1080p	10	40	0	3	23	99	442	19
YUV to HEVC	1080p	10	40	0	3	23	100	212	17
YUV to AV1	1080p	10	40	0	3	23	99	170	16
YUV to AVC	1080p	10	40	1	1	23	99	320	18
YUV to HEVC	1080p	10	40	1	1	23	99	411	19

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	40	1	2	23	100	320	18
YUV to HEVC	1080p	10	40	1	2	23	100	230	17
YUV to AVC	1080p	10	40	1	3	23	100	320	18
YUV to HEVC	1080p	10	40	1	3	23	100	154	15
YUV to AVC	1080p	10	0	0	1	27	99	451	19
YUV to HEVC	1080p	10	0	0	1	27	99	523	21
YUV to AV1	1080p	10	0	0	1	27	99	460	19
YUV to AVC	1080p	10	0	0	2	27	99	450	19
YUV to HEVC	1080p	10	0	0	2	27	99	330	19
YUV to AV1	1080p	10	0	0	2	27	100	279	18
YUV to AVC	1080p	10	0	0	3	27	99	450	19
YUV to HEVC	1080p	10	0	0	3	27	99	218	17
YUV to AV1	1080p	10	0	0	3	27	99	176	17
YUV to AVC	1080p	10	0	1	1	27	100	326	18
YUV to HEVC	1080p	10	0	1	1	27	99	420	20
YUV to AVC	1080p	10	0	1	2	27	99	325	20
YUV to HEVC	1080p	10	0	1	2	27	99	230	17
YUV to AVC	1080p	10	0	1	3	27	99	325	17
YUV to HEVC	1080p	10	0	1	3	27	99	158	15
YUV to AVC	1080p	10	4	0	1	27	99	450	19
YUV to HEVC	1080p	10	4	0	1	27	99	523	19
YUV to AV1	1080p	10	4	0	1	27	99	460	17
YUV to AVC	1080p	10	4	0	2	27	99	450	20
YUV to HEVC	1080p	10	4	0	2	27	100	330	17
YUV to AV1	1080p	10	4	0	2	27	99	271	18
YUV to AVC	1080p	10	4	0	3	27	99	449	20
YUV to HEVC	1080p	10	4	0	3	27	99	216	15
YUV to AV1	1080p	10	4	0	3	27	99	170	16
YUV to AVC	1080p	10	4	1	1	27	99	323	17
YUV to HEVC	1080p	10	4	1	1	27	99	420	20
YUV to AVC	1080p	10	4	1	2	27	99	324	18
YUV to HEVC	1080p	10	4	1	2	27	100	230	16
YUV to AVC	1080p	10	4	1	3	27	99	323	18
YUV to HEVC	1080p	10	4	1	3	27	100	158	13
YUV to AVC	1080p	10	16	0	1	27	99	448	19
YUV to HEVC	1080p	10	16	0	1	27	99	523	20
YUV to AV1	1080p	10	16	0	1	27	99	460	18
YUV to AVC	1080p	10	16	0	2	27	99	449	20
YUV to HEVC	1080p	10	16	0	2	27	100	329	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	10	16	0	2	27	100	270	17
YUV to AVC	1080p	10	16	0	3	27	99	448	20
YUV to HEVC	1080p	10	16	0	3	27	99	214	17
YUV to AV1	1080p	10	16	0	3	27	100	171	15
YUV to AVC	1080p	10	16	1	1	27	100	321	17
YUV to HEVC	1080p	10	16	1	1	27	99	418	18
YUV to AVC	1080p	10	16	1	2	27	99	321	18
YUV to HEVC	1080p	10	16	1	2	27	99	230	17
YUV to AVC	1080p	10	16	1	3	27	99	320	17
YUV to HEVC	1080p	10	16	1	3	27	99	155	16
YUV to AVC	1080p	10	40	0	1	27	99	442	18
YUV to HEVC	1080p	10	40	0	1	27	99	519	20
YUV to AV1	1080p	10	40	0	1	27	99	453	18
YUV to AVC	1080p	10	40	0	2	27	99	442	19
YUV to HEVC	1080p	10	40	0	2	27	99	320	18
YUV to AV1	1080p	10	40	0	2	27	99	270	17
YUV to AVC	1080p	10	40	0	3	27	99	442	20
YUV to HEVC	1080p	10	40	0	3	27	99	212	16
YUV to AV1	1080p	10	40	0	3	27	100	170	16
YUV to AVC	1080p	10	40	1	1	27	100	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	19
YUV to HEVC	1080p	10	40	1	2	27	99	230	17
YUV to AVC	1080p	10	40	1	3	27	99	320	17
YUV to HEVC	1080p	10	40	1	3	27	99	152	16

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

35.1 Transcoding

35.1.1 Description

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

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

35.1.2 Command line

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

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

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

Input Video: 1080p

Input Image: 128x128

35.2 Inplace Overlay Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	1	14	21	3	191	57
AVC to HEVC	1	15	21	3	203	57
AVC to AV1	1	13	21	2	174	54
HEVC to AVC	1	14	21	3	188	61
HEVC to HEVC	1	15	20	3	199	61
HEVC to AV1	1	13	21	2	172	60
VP9 to AVC	1	18	21	3	190	52
VP9 to HEVC	1	21	21	3	202	51
VP9 to AV1	1	17	20	2	173	49
AVC to AVC	8	72	95	19	536	26
AVC to HEVC	8	78	94	21	608	21
AVC to AV1	8	75	94	20	606	28
HEVC to AVC	8	67	94	19	560	26
HEVC to HEVC	8	72	93	21	635	28
HEVC to AV1	8	69	94	21	629	28
VP9 to AVC	8	74	95	18	538	25
VP9 to HEVC	8	82	92	19	612	25
VP9 to AV1	8	81	93	19	608	27
AVC to AVC	16	72	94	19	530	11
AVC to HEVC	16	77	92	21	608	11
AVC to AV1	16	76	94	21	598	12
HEVC to AVC	16	66	92	20	560	12
HEVC to HEVC	16	73	93	22	635	13
HEVC to AV1	16	70	95	22	624	11
VP9 to AVC	16	77	95	19	532	9
VP9 to HEVC	16	85	95	21	608	10
VP9 to AV1	16	83	94	21	608	11

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

36.1 Encoding

36.1.1 Description

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

YUV frame is encoded by hardware encoder.

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

36.1.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 4k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

36.2 FFmpeg Throughput Performance Results

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	4k	1	0	8	1	0	59	182	58
YUV to HEVC	4k	1	0	8	1	0	54	188	56
YUV to AV1	4k	1	0	8	1	0	61	182	63
YUV to AVC	4k	3	0	8	0	0	71	198	46
YUV to HEVC	4k	3	0	8	0	0	66	222	58
YUV to AV1	4k	3	0	8	0	0	71	213	43
YUV to AVC	4k	5	0	8	0	0	95	208	31
YUV to HEVC	4k	5	0	8	0	0	70	230	45
YUV to AV1	4k	5	0	8	0	0	76	225	41
YUV to AVC	4k	1	0	10	1	0	36	106	48
YUV to HEVC	4k	1	0	10	1	0	30	108	55
YUV to AV1	4k	1	0	10	1	0	35	106	61
YUV to AVC	4k	3	0	10	0	0	49	115	51
YUV to HEVC	4k	3	0	10	0	0	35	118	58
YUV to AV1	4k	3	0	10	0	0	39	116	56
YUV to AVC	1080p	1	0	8	1	0	32	438	79
YUV to HEVC	1080p	1	0	8	1	0	33	453	72
YUV to AV1	1080p	1	0	8	1	0	35	405	65
YUV to AVC	1080p	20	0	8	0	0	73	808	13
YUV to HEVC	1080p	20	0	8	0	0	65	847	14
YUV to AV1	1080p	20	0	8	0	0	71	815	13
YUV to AVC	1080p	1	0	10	1	0	24	309	56
YUV to HEVC	1080p	1	0	10	1	0	22	311	64
YUV to AV1	1080p	1	0	10	1	0	24	289	66
YUV to AVC	1080p	20	0	10	0	0	38	463	13
YUV to HEVC	1080p	20	0	10	0	0	34	463	13
YUV to AV1	1080p	20	0	10	0	0	39	460	13
YUV to AVC	720p	1	0	8	1	0	21	559	67
YUV to HEVC	720p	1	0	8	1	0	19	568	60
YUV to AV1	720p	1	0	8	1	0	23	506	54
YUV to AVC	720p	40	0	8	0	0	85	1563	8
YUV to HEVC	720p	40	0	8	0	0	83	1655	8
YUV to AV1	720p	40	0	8	0	0	85	1444	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	39	127	47
YUV to HEVC	4k	1	0	8	1	0	35	123	47
YUV to AV1	4k	1	0	8	1	0	39	115	47
YUV to AVC	4k	3	0	8	0	0	54	171	37
YUV to HEVC	4k	3	0	8	0	0	52	164	37
YUV to AV1	4k	3	0	8	0	0	52	157	36
YUV to AVC	4k	5	0	8	0	0	94	195	28
YUV to HEVC	4k	5	0	8	0	0	65	212	33
YUV to AV1	4k	5	0	8	0	0	64	194	33
YUV to AVC	4k	1	0	10	1	0	20	66	49
YUV to HEVC	4k	1	0	10	1	0	19	67	48
YUV to AV1	4k	1	0	10	1	0	21	63	48
YUV to AVC	4k	3	0	10	0	0	27	91	39
YUV to HEVC	4k	3	0	10	0	0	27	90	39
YUV to AV1	4k	3	0	10	0	0	30	84	38
YUV to AVC	1080p	1	0	8	1	0	23	316	45
YUV to HEVC	1080p	1	0	8	1	0	23	320	44
YUV to AV1	1080p	1	0	8	1	0	23	278	40
YUV to AVC	1080p	20	0	8	0	0	73	835	11
YUV to HEVC	1080p	20	0	8	0	0	66	868	11
YUV to AV1	1080p	20	0	8	0	0	69	822	10
YUV to AVC	1080p	1	0	10	1	0	15	203	50
YUV to HEVC	1080p	1	0	10	1	0	14	212	47
YUV to AV1	1080p	1	0	10	1	0	15	180	48
YUV to AVC	1080p	20	0	10	0	0	38	470	12
YUV to HEVC	1080p	20	0	10	0	0	34	471	11
YUV to AV1	1080p	20	0	10	0	0	38	443	12
YUV to AVC	720p	1	0	8	1	0	20	524	32
YUV to HEVC	720p	1	0	8	1	0	18	525	36
YUV to AV1	720p	1	0	8	1	0	19	419	43
YUV to AVC	720p	40	0	8	0	0	87	1699	5
YUV to HEVC	720p	40	0	8	0	0	81	1785	6
YUV to AV1	720p	40	0	8	0	0	89	1560	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	17.80	20.14	17.19	0.09
YUV to HEVC	4k	1	19.09	21.58	17.69	0.13
YUV to AV1	4k	1	24.28	28.48	18.70	0.70
YUV to AVC	4k	3	19.73	23.84	17.29	1.66
YUV to HEVC	4k	3	22.25	26.01	17.87	1.00
YUV to AV1	4k	3	26.39	30.99	18.70	2.31
YUV to AVC	4k	5	21.15	28.44	17.48	3.97
YUV to HEVC	4k	5	22.76	28.33	17.97	7.82
YUV to AV1	4k	5	28.43	36.31	18.93	2.97
YUV to AVC	1080p	1	5.86	6.90	5.31	0.05
YUV to HEVC	1080p	1	6.04	6.73	5.56	0.04
YUV to AV1	1080p	1	7.83	8.94	6.12	0.08
YUV to AVC	1080p	20	7.78	11.95	5.39	1.25
YUV to HEVC	1080p	20	8.94	18.09	5.60	3.65
YUV to AV1	1080p	20	10.80	19.27	6.16	3.93
YUV to AVC	720p	1	3.60	4.18	2.98	0.03
YUV to HEVC	720p	1	3.65	4.32	3.15	0.05
YUV to AV1	720p	1	4.55	5.07	3.68	0.05
YUV to AVC	720p	40	4.63	9.36	3.08	0.33
YUV to HEVC	720p	40	5.38	10.39	3.36	0.88
YUV to AV1	720p	40	7.52	13.40	4.34	1.81

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	81	56
RGBA to HEVC	4k	1	1	24	81	64
RGBA to AV1	4k	1	1	27	81	59
RGBA to AVC	4k	3	0	37	88	49
RGBA to HEVC	4k	3	0	31	88	49
RGBA to AV1	4k	3	0	33	87	51
RGBA to AVC	4k	5	0	49	90	36
RGBA to HEVC	4k	5	0	42	90	35
RGBA to AV1	4k	5	0	38	90	38
RGBA to AVC	1080p	1	1	19	245	64
RGBA to HEVC	1080p	1	1	17	243	87
RGBA to AV1	1080p	1	1	20	238	98
RGBA to AVC	1080p	10	0	33	330	24
RGBA to HEVC	1080p	10	0	30	333	24
RGBA to AV1	1080p	10	0	31	329	24
RGBA to AVC	1080p	20	0	38	324	13
RGBA to HEVC	1080p	20	0	31	336	13
RGBA to AV1	1080p	20	0	33	336	14
RGBA to AVC	720p	1	1	17	369	90
RGBA to HEVC	720p	1	1	14	350	112
RGBA to AV1	720p	1	1	17	345	110
RGBA to AVC	720p	10	0	34	693	46
RGBA to HEVC	720p	10	0	33	696	46
RGBA to AV1	720p	10	0	39	673	44
RGBA to AVC	720p	20	0	37	716	22
RGBA to HEVC	720p	20	0	35	724	24
RGBA to AV1	720p	20	0	42	711	22

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

40.1 Encoding

40.1.1 Description

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

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

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

40.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

40.2 Encoding EnableRdoQuant/rdoLevel/lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	0	0	1	48	154	53
YUV to HEVC	4k	2	0	0	1	48	166	38
YUV to AV1	4k	2	0	0	1	48	144	42
YUV to AVC	4k	2	0	0	2	48	154	54
YUV to HEVC	4k	2	0	0	2	49	85	36
YUV to AV1	4k	2	0	0	2	49	70	32
YUV to AVC	4k	2	0	0	3	48	154	60
YUV to HEVC	4k	2	0	0	3	50	50	27
YUV to AV1	4k	2	0	0	3	48	38	23
YUV to AVC	4k	2	0	1	1	49	94	40
YUV to HEVC	4k	2	0	1	1	49	120	44
YUV to AVC	4k	2	0	1	2	49	94	38
YUV to HEVC	4k	2	0	1	2	49	54	26
YUV to AVC	4k	2	0	1	3	49	94	41
YUV to HEVC	4k	2	0	1	3	50	34	23
YUV to AVC	4k	2	4	0	1	73	132	49
YUV to HEVC	4k	2	4	0	1	72	166	50
YUV to AV1	4k	2	4	0	1	71	144	51
YUV to AVC	4k	2	4	0	2	74	134	47
YUV to HEVC	4k	2	4	0	2	61	84	36
YUV to AV1	4k	2	4	0	2	60	70	32
YUV to AVC	4k	2	4	0	3	73	134	51
YUV to HEVC	4k	2	4	0	3	56	50	26
YUV to AV1	4k	2	4	0	3	55	40	23
YUV to AVC	4k	2	4	1	1	66	92	39
YUV to HEVC	4k	2	4	1	1	66	120	48
YUV to AVC	4k	2	4	1	2	67	92	42
YUV to HEVC	4k	2	4	1	2	58	54	28
YUV to AVC	4k	2	4	1	3	67	93	40
YUV to HEVC	4k	2	4	1	3	57	34	22
YUV to AVC	4k	2	16	0	1	74	132	43
YUV to HEVC	4k	2	16	0	1	73	166	53
YUV to AV1	4k	2	16	0	1	72	144	49

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	2	16	0	2	73	128	52
YUV to HEVC	4k	2	16	0	2	61	84	38
YUV to AV1	4k	2	16	0	2	60	70	29
YUV to AVC	4k	2	16	0	3	73	132	47
YUV to HEVC	4k	2	16	0	3	56	50	26
YUV to AV1	4k	2	16	0	3	55	40	24
YUV to AVC	4k	2	16	1	1	67	92	40
YUV to HEVC	4k	2	16	1	1	66	118	48
YUV to AVC	4k	2	16	1	2	66	92	37
YUV to HEVC	4k	2	16	1	2	57	54	30
YUV to AVC	4k	2	16	1	3	68	92	36
YUV to HEVC	4k	2	16	1	3	55	34	21
YUV to AVC	4k	2	40	0	1	72	126	48
YUV to HEVC	4k	2	40	0	1	73	164	42
YUV to AV1	4k	2	40	0	1	72	142	51
YUV to AVC	4k	2	40	0	2	74	130	41
YUV to HEVC	4k	2	40	0	2	61	84	37
YUV to AV1	4k	2	40	0	2	60	70	33
YUV to AVC	4k	2	40	0	3	72	126	45
YUV to HEVC	4k	2	40	0	3	57	50	26
YUV to AV1	4k	2	40	0	3	56	40	22
YUV to AVC	4k	2	40	1	1	67	92	38
YUV to HEVC	4k	2	40	1	1	66	118	50
YUV to AVC	4k	2	40	1	2	67	92	38
YUV to HEVC	4k	2	40	1	2	56	54	27
YUV to AVC	4k	2	40	1	3	65	90	42
YUV to HEVC	4k	2	40	1	3	53	34	21
YUV to AVC	1080p	10	0	0	1	66	782	25
YUV to HEVC	1080p	10	0	0	1	60	813	25
YUV to AV1	1080p	10	0	0	1	66	773	24
YUV to AVC	1080p	10	0	0	2	65	781	25
YUV to HEVC	1080p	10	0	0	2	99	688	24
YUV to AV1	1080p	10	0	0	2	100	559	20
YUV to AVC	1080p	10	0	0	3	65	779	25

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	100	408	21
YUV to AV1	1080p	10	0	0	3	99	298	18
YUV to AVC	1080p	10	0	1	1	100	754	23
YUV to HEVC	1080p	10	0	1	1	84	824	27
YUV to AVC	1080p	10	0	1	2	100	753	23
YUV to HEVC	1080p	10	0	1	2	100	438	19
YUV to AVC	1080p	10	0	1	3	99	755	24
YUV to HEVC	1080p	10	0	1	3	100	276	17
YUV to AVC	1080p	10	4	0	1	99	647	20
YUV to HEVC	1080p	10	4	0	1	85	702	20
YUV to AV1	1080p	10	4	0	1	91	677	20
YUV to AVC	1080p	10	4	0	2	99	647	18
YUV to HEVC	1080p	10	4	0	2	99	510	21
YUV to AV1	1080p	10	4	0	2	99	430	19
YUV to AVC	1080p	10	4	0	3	99	648	20
YUV to HEVC	1080p	10	4	0	3	99	340	19
YUV to AV1	1080p	10	4	0	3	99	270	17
YUV to AVC	1080p	10	4	1	1	99	510	21
YUV to HEVC	1080p	10	4	1	1	99	654	22
YUV to AVC	1080p	10	4	1	2	99	510	19
YUV to HEVC	1080p	10	4	1	2	100	360	18
YUV to AVC	1080p	10	4	1	3	100	510	20
YUV to HEVC	1080p	10	4	1	3	99	247	17
YUV to AVC	1080p	10	16	0	1	99	640	18
YUV to HEVC	1080p	10	16	0	1	88	712	22
YUV to AV1	1080p	10	16	0	1	93	666	21
YUV to AVC	1080p	10	16	0	2	99	640	19
YUV to HEVC	1080p	10	16	0	2	100	511	19
YUV to AV1	1080p	10	16	0	2	99	430	18
YUV to AVC	1080p	10	16	0	3	99	638	19
YUV to HEVC	1080p	10	16	0	3	99	339	18
YUV to AV1	1080p	10	16	0	3	100	270	17
YUV to AVC	1080p	10	16	1	1	99	509	20
YUV to HEVC	1080p	10	16	1	1	99	651	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	1	2	99	508	19
YUV to HEVC	1080p	10	16	1	2	99	360	18
YUV to AVC	1080p	10	16	1	3	100	508	19
YUV to HEVC	1080p	10	16	1	3	99	243	16
YUV to AVC	1080p	10	40	0	1	99	629	20
YUV to HEVC	1080p	10	40	0	1	88	694	21
YUV to AV1	1080p	10	40	0	1	89	640	18
YUV to AVC	1080p	10	40	0	2	99	630	19
YUV to HEVC	1080p	10	40	0	2	99	509	20
YUV to AV1	1080p	10	40	0	2	99	420	18
YUV to AVC	1080p	10	40	0	3	99	630	18
YUV to HEVC	1080p	10	40	0	3	99	334	19
YUV to AV1	1080p	10	40	0	3	100	268	16
YUV to AVC	1080p	10	40	1	1	99	500	20
YUV to HEVC	1080p	10	40	1	1	99	647	20
YUV to AVC	1080p	10	40	1	2	100	500	19
YUV to HEVC	1080p	10	40	1	2	99	360	18
YUV to AVC	1080p	10	40	1	3	99	500	20
YUV to HEVC	1080p	10	40	1	3	99	241	17
YUV to AVC	720p	20	0	0	1	75	1461	13
YUV to HEVC	720p	20	0	0	1	73	1540	15
YUV to AV1	720p	20	0	0	1	79	1380	13
YUV to AVC	720p	20	0	0	2	78	1488	14
YUV to HEVC	720p	20	0	0	2	99	1505	15
YUV to AV1	720p	20	0	0	2	99	1220	14
YUV to AVC	720p	20	0	0	3	76	1470	14
YUV to HEVC	720p	20	0	0	3	99	902	14
YUV to AV1	720p	20	0	0	3	99	660	7
YUV to AVC	720p	20	0	1	1	90	1532	14
YUV to HEVC	720p	20	0	1	1	74	1559	15
YUV to AVC	720p	20	0	1	2	93	1568	14
YUV to HEVC	720p	20	0	1	2	99	960	14
YUV to AVC	720p	20	0	1	3	92	1569	14
YUV to HEVC	720p	20	0	1	3	100	620	13

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	20	4	0	1	99	1081	10
YUV to HEVC	720p	20	4	0	1	99	1095	9
YUV to AV1	720p	20	4	0	1	100	862	7
YUV to AVC	720p	20	4	0	2	98	1080	10
YUV to HEVC	720p	20	4	0	2	99	1080	11
YUV to AV1	720p	20	4	0	2	99	860	7
YUV to AVC	720p	20	4	0	3	99	1082	9
YUV to HEVC	720p	20	4	0	3	99	740	10
YUV to AV1	720p	20	4	0	3	99	560	9
YUV to AVC	720p	20	4	1	1	99	1066	13
YUV to HEVC	720p	20	4	1	1	100	1100	9
YUV to AVC	720p	20	4	1	2	99	1068	12
YUV to HEVC	720p	20	4	1	2	99	760	14
YUV to AVC	720p	20	4	1	3	99	1065	13
YUV to HEVC	720p	20	4	1	3	100	534	8
YUV to AVC	720p	20	16	0	1	99	1060	9
YUV to HEVC	720p	20	16	0	1	99	1100	10
YUV to AV1	720p	20	16	0	1	100	900	7
YUV to AVC	720p	20	16	0	2	98	1058	10
YUV to HEVC	720p	20	16	0	2	99	1076	13
YUV to AV1	720p	20	16	0	2	99	880	10
YUV to AVC	720p	20	16	0	3	97	1056	9
YUV to HEVC	720p	20	16	0	3	99	727	15
YUV to AV1	720p	20	16	0	3	99	560	9
YUV to AVC	720p	20	16	1	1	99	1060	12
YUV to HEVC	720p	20	16	1	1	100	1101	9
YUV to AVC	720p	20	16	1	2	99	1060	13
YUV to HEVC	720p	20	16	1	2	99	760	14
YUV to AVC	720p	20	16	1	3	99	1060	13
YUV to HEVC	720p	20	16	1	3	99	520	9
YUV to AVC	720p	20	40	0	1	99	1028	9
YUV to HEVC	720p	20	40	0	1	99	1068	9
YUV to AV1	720p	20	40	0	1	100	880	8
YUV to AVC	720p	20	40	0	2	99	1040	9

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to HEVC	720p	20	40	0	2	99	1060	11
YUV to AV1	720p	20	40	0	2	99	874	10
YUV to AVC	720p	20	40	0	3	100	1037	9
YUV to HEVC	720p	20	40	0	3	100	720	14
YUV to AV1	720p	20	40	0	3	100	560	10
YUV to AVC	720p	20	40	1	1	99	1058	13
YUV to HEVC	720p	20	40	1	1	99	1080	10
YUV to AVC	720p	20	40	1	2	99	1056	13
YUV to HEVC	720p	20	40	1	2	99	760	15
YUV to AVC	720p	20	40	1	3	99	1059	11
YUV to HEVC	720p	20	40	1	3	100	520	7

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

41.1 Encoding with lookaheadDepth

41.1.1 Description

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

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

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

41.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

41.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	0	0	1	19	99	683	22
YUV to HEVC	1080p	10	0	0	1	19	94	758	23
YUV to AV1	1080p	10	0	0	1	19	98	703	20
YUV to AVC	1080p	10	0	0	2	19	99	682	20
YUV to HEVC	1080p	10	0	0	2	19	99	511	20
YUV to AV1	1080p	10	0	0	2	19	99	430	18
YUV to AVC	1080p	10	0	0	3	19	99	682	21
YUV to HEVC	1080p	10	0	0	3	19	100	340	18
YUV to AV1	1080p	10	0	0	3	19	100	278	17
YUV to AVC	1080p	10	0	1	1	19	99	510	20
YUV to HEVC	1080p	10	0	1	1	19	99	658	21
YUV to AVC	1080p	10	0	1	2	19	100	510	20
YUV to HEVC	1080p	10	0	1	2	19	100	360	18
YUV to AVC	1080p	10	0	1	3	19	99	510	20
YUV to HEVC	1080p	10	0	1	3	19	100	246	16
YUV to AVC	1080p	10	4	0	1	19	99	641	19
YUV to HEVC	1080p	10	4	0	1	19	84	684	20
YUV to AV1	1080p	10	4	0	1	19	87	625	19
YUV to AVC	1080p	10	4	0	2	19	99	642	19
YUV to HEVC	1080p	10	4	0	2	19	100	511	19
YUV to AV1	1080p	10	4	0	2	19	100	430	18
YUV to AVC	1080p	10	4	0	3	19	99	643	20
YUV to HEVC	1080p	10	4	0	3	19	99	339	18
YUV to AV1	1080p	10	4	0	3	19	100	270	18
YUV to AVC	1080p	10	4	1	1	19	99	510	21
YUV to HEVC	1080p	10	4	1	1	19	100	655	20
YUV to AVC	1080p	10	4	1	2	19	99	510	20
YUV to HEVC	1080p	10	4	1	2	19	100	360	19
YUV to AVC	1080p	10	4	1	3	19	100	509	20
YUV to HEVC	1080p	10	4	1	3	19	100	246	16
YUV to AVC	1080p	10	16	0	1	19	92	591	19
YUV to HEVC	1080p	10	16	0	1	19	83	669	20
YUV to AV1	1080p	10	16	0	1	19	85	613	17

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	0	2	19	92	593	18
YUV to HEVC	1080p	10	16	0	2	19	99	510	19
YUV to AV1	1080p	10	16	0	2	19	99	429	18
YUV to AVC	1080p	10	16	0	3	19	91	594	18
YUV to HEVC	1080p	10	16	0	3	19	100	339	17
YUV to AV1	1080p	10	16	0	3	19	99	270	17
YUV to AVC	1080p	10	16	1	1	19	99	506	19
YUV to HEVC	1080p	10	16	1	1	19	100	650	20
YUV to AVC	1080p	10	16	1	2	19	99	508	19
YUV to HEVC	1080p	10	16	1	2	19	99	360	18
YUV to AVC	1080p	10	16	1	3	19	100	508	19
YUV to HEVC	1080p	10	16	1	3	19	99	244	17
YUV to AVC	1080p	10	40	0	1	19	90	579	18
YUV to HEVC	1080p	10	40	0	1	19	81	645	20
YUV to AV1	1080p	10	40	0	1	19	85	582	18
YUV to AVC	1080p	10	40	0	2	19	91	581	18
YUV to HEVC	1080p	10	40	0	2	19	99	506	18
YUV to AV1	1080p	10	40	0	2	19	100	420	18
YUV to AVC	1080p	10	40	0	3	19	91	580	18
YUV to HEVC	1080p	10	40	0	3	19	99	335	18
YUV to AV1	1080p	10	40	0	3	19	100	268	17
YUV to AVC	1080p	10	40	1	1	19	99	500	19
YUV to HEVC	1080p	10	40	1	1	19	99	647	20
YUV to AVC	1080p	10	40	1	2	19	100	500	18
YUV to HEVC	1080p	10	40	1	2	19	100	357	18
YUV to AVC	1080p	10	40	1	3	19	100	500	18
YUV to HEVC	1080p	10	40	1	3	19	100	241	17
YUV to AVC	1080p	10	0	0	1	23	99	683	22
YUV to HEVC	1080p	10	0	0	1	23	92	753	23
YUV to AV1	1080p	10	0	0	1	23	96	702	21
YUV to AVC	1080p	10	0	0	2	23	99	683	21
YUV to HEVC	1080p	10	0	0	2	23	100	510	21
YUV to AV1	1080p	10	0	0	2	23	99	430	18
YUV to AVC	1080p	10	0	0	3	23	100	681	21

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	0	0	3	23	99	340	19
YUV to AV1	1080p	10	0	0	3	23	100	277	17
YUV to AVC	1080p	10	0	1	1	23	100	509	19
YUV to HEVC	1080p	10	0	1	1	23	99	655	21
YUV to AVC	1080p	10	0	1	2	23	99	510	20
YUV to HEVC	1080p	10	0	1	2	23	99	360	19
YUV to AVC	1080p	10	0	1	3	23	99	510	20
YUV to HEVC	1080p	10	0	1	3	23	100	245	17
YUV to AVC	1080p	10	4	0	1	23	99	643	19
YUV to HEVC	1080p	10	4	0	1	23	86	699	21
YUV to AV1	1080p	10	4	0	1	23	88	637	20
YUV to AVC	1080p	10	4	0	2	23	99	644	20
YUV to HEVC	1080p	10	4	0	2	23	100	511	19
YUV to AV1	1080p	10	4	0	2	23	100	430	18
YUV to AVC	1080p	10	4	0	3	23	99	643	20
YUV to HEVC	1080p	10	4	0	3	23	100	340	18
YUV to AV1	1080p	10	4	0	3	23	100	271	17
YUV to AVC	1080p	10	4	1	1	23	100	510	19
YUV to HEVC	1080p	10	4	1	1	23	100	658	20
YUV to AVC	1080p	10	4	1	2	23	100	510	19
YUV to HEVC	1080p	10	4	1	2	23	100	360	19
YUV to AVC	1080p	10	4	1	3	23	100	510	21
YUV to HEVC	1080p	10	4	1	3	23	99	244	19
YUV to AVC	1080p	10	16	0	1	23	95	616	18
YUV to HEVC	1080p	10	16	0	1	23	84	680	20
YUV to AV1	1080p	10	16	0	1	23	86	623	19
YUV to AVC	1080p	10	16	0	2	23	95	616	17
YUV to HEVC	1080p	10	16	0	2	23	100	510	19
YUV to AV1	1080p	10	16	0	2	23	100	429	17
YUV to AVC	1080p	10	16	0	3	23	95	613	18
YUV to HEVC	1080p	10	16	0	3	23	100	340	19
YUV to AV1	1080p	10	16	0	3	23	100	270	17
YUV to AVC	1080p	10	16	1	1	23	99	508	20
YUV to HEVC	1080p	10	16	1	1	23	99	650	20

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	16	1	2	23	99	506	20
YUV to HEVC	1080p	10	16	1	2	23	99	360	18
YUV to AVC	1080p	10	16	1	3	23	99	508	20
YUV to HEVC	1080p	10	16	1	3	23	99	243	17
YUV to AVC	1080p	10	40	0	1	23	95	601	17
YUV to HEVC	1080p	10	40	0	1	23	82	662	20
YUV to AV1	1080p	10	40	0	1	23	86	598	17
YUV to AVC	1080p	10	40	0	2	23	95	601	17
YUV to HEVC	1080p	10	40	0	2	23	100	507	20
YUV to AV1	1080p	10	40	0	2	23	100	420	19
YUV to AVC	1080p	10	40	0	3	23	92	601	17
YUV to HEVC	1080p	10	40	0	3	23	100	334	18
YUV to AV1	1080p	10	40	0	3	23	100	269	17
YUV to AVC	1080p	10	40	1	1	23	99	500	18
YUV to HEVC	1080p	10	40	1	1	23	99	643	20
YUV to AVC	1080p	10	40	1	2	23	99	501	18
YUV to HEVC	1080p	10	40	1	2	23	99	358	19
YUV to AVC	1080p	10	40	1	3	23	99	500	20
YUV to HEVC	1080p	10	40	1	3	23	100	241	17
YUV to AVC	1080p	10	0	0	1	27	100	681	21
YUV to HEVC	1080p	10	0	0	1	27	93	760	23
YUV to AV1	1080p	10	0	0	1	27	97	704	20
YUV to AVC	1080p	10	0	0	2	27	99	680	22
YUV to HEVC	1080p	10	0	0	2	27	99	510	21
YUV to AV1	1080p	10	0	0	2	27	100	431	18
YUV to AVC	1080p	10	0	0	3	27	99	684	23
YUV to HEVC	1080p	10	0	0	3	27	100	340	19
YUV to AV1	1080p	10	0	0	3	27	99	278	17
YUV to AVC	1080p	10	0	1	1	27	100	510	22
YUV to HEVC	1080p	10	0	1	1	27	99	657	23
YUV to AVC	1080p	10	0	1	2	27	99	510	20
YUV to HEVC	1080p	10	0	1	2	27	100	360	19
YUV to AVC	1080p	10	0	1	3	27	100	510	21
YUV to HEVC	1080p	10	0	1	3	27	99	246	18

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	10	4	0	1	27	99	645	19
YUV to HEVC	1080p	10	4	0	1	27	87	708	20
YUV to AV1	1080p	10	4	0	1	27	91	651	18
YUV to AVC	1080p	10	4	0	2	27	99	644	19
YUV to HEVC	1080p	10	4	0	2	27	99	511	20
YUV to AV1	1080p	10	4	0	2	27	100	430	18
YUV to AVC	1080p	10	4	0	3	27	99	645	19
YUV to HEVC	1080p	10	4	0	3	27	99	340	19
YUV to AV1	1080p	10	4	0	3	27	99	271	18
YUV to AVC	1080p	10	4	1	1	27	100	509	19
YUV to HEVC	1080p	10	4	1	1	27	100	657	20
YUV to AVC	1080p	10	4	1	2	27	99	509	20
YUV to HEVC	1080p	10	4	1	2	27	99	360	20
YUV to AVC	1080p	10	4	1	3	27	99	509	20
YUV to HEVC	1080p	10	4	1	3	27	99	244	18
YUV to AVC	1080p	10	16	0	1	27	99	639	19
YUV to HEVC	1080p	10	16	0	1	27	86	691	20
YUV to AV1	1080p	10	16	0	1	27	88	634	18
YUV to AVC	1080p	10	16	0	2	27	99	640	19
YUV to HEVC	1080p	10	16	0	2	27	100	510	19
YUV to AV1	1080p	10	16	0	2	27	99	430	17
YUV to AVC	1080p	10	16	0	3	27	98	634	18
YUV to HEVC	1080p	10	16	0	3	27	100	340	19
YUV to AV1	1080p	10	16	0	3	27	100	270	17
YUV to AVC	1080p	10	16	1	1	27	100	508	20
YUV to HEVC	1080p	10	16	1	1	27	100	650	21
YUV to AVC	1080p	10	16	1	2	27	99	508	20
YUV to HEVC	1080p	10	16	1	2	27	100	360	19
YUV to AVC	1080p	10	16	1	3	27	99	507	19
YUV to HEVC	1080p	10	16	1	3	27	99	245	16
YUV to AVC	1080p	10	40	0	1	27	98	621	18
YUV to HEVC	1080p	10	40	0	1	27	84	671	20
YUV to AV1	1080p	10	40	0	1	27	85	609	17
YUV to AVC	1080p	10	40	0	2	27	98	619	19

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to HEVC	1080p	10	40	0	2	27	99	505	19
YUV to AV1	1080p	10	40	0	2	27	100	421	18
YUV to AVC	1080p	10	40	0	3	27	97	622	18
YUV to HEVC	1080p	10	40	0	3	27	100	335	17
YUV to AV1	1080p	10	40	0	3	27	100	267	18
YUV to AVC	1080p	10	40	1	1	27	100	500	19
YUV to HEVC	1080p	10	40	1	1	27	99	646	20
YUV to AVC	1080p	10	40	1	2	27	100	502	19
YUV to HEVC	1080p	10	40	1	2	27	100	357	19
YUV to AVC	1080p	10	40	1	3	27	99	500	20
YUV to HEVC	1080p	10	40	1	3	27	99	240	16

Appendix A: GStreamer XStack Command

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

[illegible]

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

Appendix B: 7x7 Grid Layout

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

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

Appendix C: GStreamer Ladder Command

Example of single input with 64 outputs

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

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