



Quadra™ Performance Test Report V4.8

Contents

Contents	2
Environment Overview	3
Definitions	4
1. T1A – FFmpeg Throughput.....	5
2. T1A – Libxcodec Throughput.....	11
3. T1A – FFmpeg Latency	17
4. T1A – Decoder PPU Scaling	19
5. T1A – Streaming Ladder Generation.....	20
6. T1A – RGBA Encoding.....	21
7. T1A – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth	23
8. T1A – Capped CRF	28
9. T1A – Inplace Overlay	32
10. 2x T2A – MultiThread P2P DMA on AMD GPU	34
11. T1A – AI	35
12. T1A – GStreamer XStack Throughput	39
13. T1A – GStreamer Ladder Generation.....	40
14. T1U – FFmpeg Throughput	41
15. T1U – Libxcodec Throughput	47
16. T1U – FFmpeg Latency.....	53
17. T1U – Decoder PPU Scaling.....	55
18. T1U – Streaming Ladder Generation	56
19. T1U – RGBA Encoding	57
20. T1U – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth	59
21. T1U – Capped CRF.....	64
22. T1U – Inplace Overlay	68
Appendix A: GStreamer XStack Command	70
Appendix B: 7x7 Grid Layout.....	72
Appendix C: GStreamer Ladder Command	73

Environment Overview

FW Revision: 4806rJr1

SW Revision: 4806rjr2

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) 2133 MHz (0.5 ns)
- DUT: 1x T1A or 1x T1U
- FFmpeg Version: 6.0
- Gstreamer Version: 1.22.2
- Tests:
 - FFmpeg Throughput
 - Libxcodec Throughput
 - FFmpeg Latency
 - Decoder PPU Scaling
 - Streaming Ladder Generation
 - Inplace Overlay
 - Encoding EnableRdoQuant/rdoLevel/lookaheadDepth (T1A only)
 - Gstreamer XStack Throughput (T1A only)
 - Gstreamer Ladder Generation (T1A only)

Setup #2:

- Server: AMD EPYC 7763 64-Core Processor; CPU(s) 128; Motherboard OPYVT1; Memory 256GiB System Memory, 8x32GiB DIMM DDR4 Synchronous Registered (Buffered) 3200 MHz (0.3 ns)
- DUT: 2x T2A
- FFmpeg Version: 4.3.1
- Tests:
 - MultiThread P2P DMA on AMD GPU

Setup #3:

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

Definitions

- CPU: Average per instance CPU usage.
 - $(\text{System-wide CPU usage} * \text{number of CPU}) / (\text{number of devices} * \text{number of instances per device})$
- FPS: Average of all FPS reported per process
- Jobs: Number of instances running concurrently
- HW Frame: Decoded YUV is kept on the device
- Bit: Input video's bit depth
- Resolution: Input video's resolution

1. T1A – FFmpeg Throughput

1.1 Decoding

1.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

1.1.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

1.2 Encoding

1.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

1.2.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

1.3 Transcoding

1.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

1.3.2 Command line

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

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

1.4 FFmpeg Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	95	0	115	9
HEVC to YUV	8k	1	0	8	1	65	0	104	9
VP9 to YUV	8k	1	0	8	1	24	0	38	3
YUV to AVC	8k	1	0	8	1	0	97	68	52
YUV to HEVC	8k	1	0	8	1	0	98	83	66
AVC to AVC	8k	1	1	8	1	67	91	54	3
AVC to HEVC	8k	1	1	8	1	80	96	72	4
HEVC to AVC	8k	1	1	8	1	64	97	53	3
HEVC to HEVC	8k	1	1	8	1	72	98	70	4
VP9 to AVC	8k	1	1	8	1	25	43	35	2
VP9 to HEVC	8k	1	1	8	1	24	43	37	1
AVC to YUV	8k	1	0	10	1	50	0	56	8
HEVC to YUV	8k	1	0	10	1	51	0	59	6
VP9 to YUV	8k	1	0	10	1	23	0	33	6
YUV to AVC	8k	1	0	10	1	0	95	49	74
YUV to HEVC	8k	1	0	10	1	0	74	60	93
AVC to YUV	4k	1	0	8	1	55	0	313	16
HEVC to YUV	4k	1	0	8	1	50	0	330	19
VP9 to YUV	4k	1	0	8	1	24	0	155	4
AVC to YUV	4k	16	0	8	0	98	0	481	2
HEVC to YUV	4k	16	0	8	0	99	0	505	2
VP9 to YUV	4k	16	0	8	0	98	0	485	1
YUV to AVC	4k	1	0	8	1	0	93	300	33
YUV to HEVC	4k	1	0	8	1	0	97	333	37
YUV to AV1	4k	1	0	8	1	0	95	290	33
YUV to AVC	4k	4	0	8	0	0	98	304	22
YUV to HEVC	4k	4	0	8	0	0	98	336	18
YUV to AV1	4k	4	0	8	0	0	97	292	11
YUV to AVC	4k	8	0	8	0	0	99	323	8
YUV to HEVC	4k	8	0	8	0	0	99	344	7
YUV to AV1	4k	8	0	8	0	0	100	296	6
AVC to AVC	4k	1	1	8	1	64	85	216	10
AVC to HEVC	4k	1	1	8	1	71	88	271	12
AVC to AV1	4k	1	1	8	1	60	88	254	12
HEVC to AVC	4k	1	1	8	1	53	88	213	11
HEVC to HEVC	4k	1	1	8	1	58	88	267	14
HEVC to AV1	4k	1	1	8	1	50	88	251	13
VP9 to AVC	4k	1	1	8	1	24	46	149	3
VP9 to HEVC	4k	1	1	8	1	24	43	151	3

T1A – FFmpeg Throughput

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	51	151	3
AVC to AVC	4k	4	1	8	0	61	97	236	3
AVC to HEVC	4k	4	1	8	0	68	97	296	4
AVC to AV1	4k	4	1	8	0	61	96	273	4
HEVC to AVC	4k	4	1	8	0	58	97	236	4
HEVC to HEVC	4k	4	1	8	0	57	91	292	5
HEVC to AV1	4k	4	1	8	0	49	97	276	4
VP9 to AVC	4k	4	1	8	0	64	97	244	2
VP9 to HEVC	4k	4	1	8	0	66	97	308	2
VP9 to AV1	4k	4	1	8	0	61	97	276	2
AVC to AVC	4k	8	1	8	0	69	99	211	2
AVC to HEVC	4k	8	1	8	0	76	99	280	3
AVC to AV1	4k	8	1	8	0	68	99	272	3
HEVC to AVC	4k	8	1	8	0	64	99	216	2
HEVC to HEVC	4k	8	1	8	0	66	99	280	3
HEVC to AV1	4k	8	1	8	0	58	100	275	3
VP9 to AVC	4k	8	1	8	0	67	99	235	1
VP9 to HEVC	4k	8	1	8	0	72	100	304	1
VP9 to AV1	4k	8	1	8	0	64	100	280	1
AVC to YUV	4k	1	0	10	0	45	0	217	7
HEVC to YUV	4k	1	0	10	0	42	0	220	7
VP9 to YUV	4k	1	0	10	0	24	0	158	3
AVC to YUV	4k	16	0	10	0	100	0	285	1
HEVC to YUV	4k	16	0	10	0	98	0	281	1
VP9 to YUV	4k	16	0	10	0	100	0	500	1
YUV to AVC	4k	1	0	10	0	0	73	204	51
YUV to HEVC	4k	1	0	10	0	0	62	214	54
YUV to AV1	4k	1	0	10	0	0	70	206	52
YUV to AVC	4k	4	0	10	0	0	95	221	28
YUV to HEVC	4k	4	0	10	0	0	76	254	37
YUV to AV1	4k	4	0	10	0	0	83	244	37
AVC to YUV	1080p	1	0	8	1	42	0	853	18
HEVC to YUV	1080p	1	0	8	1	45	0	873	27
VP9 to YUV	1080p	1	0	8	1	22	0	556	6
AVC to YUV	1080p	40	0	8	0	84	0	1677	1
HEVC to YUV	1080p	40	0	8	0	99	0	1870	1
VP9 to YUV	1080p	40	0	8	0	79	0	1789	0
YUV to AVC	1080p	1	0	8	1	0	64	832	21
YUV to HEVC	1080p	1	0	8	1	0	59	826	22
YUV to AV1	1080p	1	0	8	1	0	64	758	19

T1A – FFmpeg Throughput

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1280	3
YUV to HEVC	1080p	32	0	8	0	0	99	1376	3
YUV to AV1	1080p	32	0	8	0	0	100	1184	2
AVC to AVC	1080p	1	1	8	1	44	60	760	15
AVC to HEVC	1080p	1	1	8	1	42	57	771	15
AVC to AV1	1080p	1	1	8	1	39	62	710	15
HEVC to AVC	1080p	1	1	8	1	42	61	779	24
HEVC to HEVC	1080p	1	1	8	1	41	58	795	24
HEVC to AV1	1080p	1	1	8	1	36	61	724	23
VP9 to AVC	1080p	1	1	8	1	21	42	548	5
VP9 to HEVC	1080p	1	1	8	1	22	39	542	5
VP9 to AV1	1080p	1	1	8	1	21	46	542	6
AVC to AVC	1080p	32	1	8	0	75	100	960	1
AVC to HEVC	1080p	32	1	8	0	83	99	1078	1
AVC to AV1	1080p	32	1	8	0	76	100	1038	1
HEVC to AVC	1080p	32	1	8	0	68	99	993	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1123	1
HEVC to AV1	1080p	32	1	8	0	68	99	1061	1
VP9 to AVC	1080p	32	1	8	0	64	100	1088	1
VP9 to HEVC	1080p	32	1	8	0	68	99	1231	1
VP9 to AV1	1080p	32	1	8	0	59	99	1120	1
AVC to YUV	1080p	1	0	10	0	32	0	683	8
HEVC to YUV	1080p	1	0	10	0	28	0	714	8
VP9 to YUV	1080p	1	0	10	0	22	0	456	6
AVC to YUV	1080p	40	0	10	0	70	0	1120	0
HEVC to YUV	1080p	40	0	10	0	73	0	1129	0
VP9 to YUV	1080p	40	0	10	0	74	0	1120	0
YUV to AVC	1080p	1	0	10	0	0	43	561	31
YUV to HEVC	1080p	1	0	10	0	0	40	557	31
YUV to AV1	1080p	1	0	10	0	0	44	521	28
YUV to AVC	1080p	32	0	10	0	0	66	859	8
YUV to HEVC	1080p	32	0	10	0	0	61	849	8
YUV to AV1	1080p	32	0	10	0	0	70	832	8
AVC to YUV	720p	1	0	8	1	29	0	1264	16
HEVC to YUV	720p	1	0	8	1	32	0	1279	22
VP9 to YUV	720p	1	0	8	1	19	0	1021	7
AVC to YUV	720p	100	0	8	0	55	0	2400	0
HEVC to YUV	720p	100	0	8	0	67	0	2768	0
VP9 to YUV	720p	100	0	8	0	46	0	2400	0
YUV to AVC	720p	1	0	8	1	0	39	1144	25

T1A – FFmpeg Throughput

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	38	1125	22
YUV to AV1	720p	1	0	8	1	0	40	1014	19
YUV to AVC	720p	64	0	8	0	0	82	2385	8
YUV to HEVC	720p	64	0	8	0	0	83	2440	8
YUV to AV1	720p	64	0	8	0	0	80	1991	5
AVC to AVC	720p	1	1	8	1	21	30	885	12
AVC to HEVC	720p	1	1	8	1	21	30	886	12
AVC to AV1	720p	1	1	8	1	19	32	802	12
HEVC to AVC	720p	1	1	8	1	22	31	901	17
HEVC to HEVC	720p	1	1	8	1	22	30	898	16
HEVC to AV1	720p	1	1	8	1	20	33	812	16
VP9 to AVC	720p	1	1	8	1	19	34	995	8
VP9 to HEVC	720p	1	1	8	1	19	33	988	9
VP9 to AV1	720p	1	1	8	1	17	36	908	8
AVC to AVC	720p	64	1	8	0	66	82	1934	0
AVC to HEVC	720p	64	1	8	0	61	77	1992	0
AVC to AV1	720p	64	1	8	0	47	71	1669	0
HEVC to AVC	720p	64	1	8	0	65	83	1945	0
HEVC to HEVC	720p	64	1	8	0	61	80	2010	0
HEVC to AV1	720p	64	1	8	0	48	72	1675	0
VP9 to AVC	720p	64	1	8	0	51	77	2068	0
VP9 to HEVC	720p	64	1	8	0	51	76	2117	0
VP9 to AV1	720p	64	1	8	0	39	71	1792	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

```
~/FFmpegXcoder/libxcoder/build/xcoder -c 0 -s <resolution> -r 1000 -i  
/media/ramdisk/input.<ext> -m <test_type> -o /dev/null -b <bit> -t -x  
keepAliveTimeout=2 -d multicoreJointMode=<*>
```

<test_type> = test codecs. ie. a2y, a2h, etc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

```
~/FFmpegXcoder/libxcoder/build/xcoder -c 0 -s <resolution> -r 1000 -i  
/media/ramdisk/input.yuv -m <test_type> -o /dev/null -b <bit> -t -x  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*>
```

<test_type> = test codecs. ie. a2y, a2h, 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)

2.3 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

```
~/FFmpegXcoder/libxcoder/build/xcoder -c 0 -s <resolution> -r 1000 -i  
/media/ramdisk/input.<ext> -m <test_type> -o /dev/null -b <bit> -t -x  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*> -d out=hw:semitplanar0=1:multicoreJointMode=1
```

<test_type> = test codecs. ie. a2y, a2h, 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 Libxcoder 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	64	0	98	7
HEVC to YUV	8k	1	0	8	1	74	0	103	6
VP9 to YUV	8k	1	0	8	1	24	0	37	3
YUV to AVC	8k	1	0	8	1	0	97	65	26
YUV to HEVC	8k	1	0	8	1	0	99	83	32
AVC to AVC	8k	1	1	8	1	71	90	52	4
AVC to HEVC	8k	1	1	8	1	76	97	69	5
HEVC to AVC	8k	1	1	8	1	60	94	51	3
HEVC to HEVC	8k	1	1	8	1	70	98	70	4
VP9 to AVC	8k	1	1	8	1	24	45	35	2
VP9 to HEVC	8k	1	1	8	1	24	45	37	2
AVC to YUV	8k	1	0	10	1	53	0	56	6
HEVC to YUV	8k	1	0	10	1	48	0	54	6
VP9 to YUV	8k	1	0	10	1	25	0	32	4
YUV to AVC	8k	1	0	10	1	0	65	41	35
YUV to HEVC	8k	1	0	10	1	0	47	40	34
AVC to YUV	4k	1	0	8	1	62	0	336	8
HEVC to YUV	4k	1	0	8	1	53	0	348	6
VP9 to YUV	4k	1	0	8	1	24	0	151	2
AVC to YUV	4k	16	0	8	0	98	0	478	1
HEVC to YUV	4k	16	0	8	0	99	0	501	0
VP9 to YUV	4k	16	0	8	0	99	0	483	0
YUV to AVC	4k	1	0	8	1	0	96	302	25
YUV to HEVC	4k	1	0	8	1	0	96	328	31
YUV to AV1	4k	1	0	8	1	0	95	282	23
YUV to AVC	4k	4	0	8	0	0	97	311	9
YUV to HEVC	4k	4	0	8	0	0	96	333	10
YUV to AV1	4k	4	0	8	0	0	96	287	9
YUV to AVC	4k	8	0	8	0	0	99	322	6
YUV to HEVC	4k	8	0	8	0	0	100	342	6
YUV to AV1	4k	8	0	8	0	0	99	296	6
AVC to AVC	4k	1	1	8	0	67	92	218	6
AVC to HEVC	4k	1	1	8	0	73	93	272	6
AVC to AV1	4k	1	1	8	0	66	96	255	7
HEVC to AVC	4k	1	1	8	0	59	95	225	4
HEVC to HEVC	4k	1	1	8	0	65	97	286	6
HEVC to AV1	4k	1	1	8	0	56	96	266	5
VP9 to AVC	4k	1	1	8	0	24	46	147	2

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	0	24	43	147	3
VP9 to AV1	4k	1	1	8	0	24	51	146	3
AVC to AVC	4k	4	1	8	0	61	96	236	4
AVC to HEVC	4k	4	1	8	0	57	84	287	4
AVC to AV1	4k	4	1	8	0	55	93	265	5
HEVC to AVC	4k	4	1	8	0	51	88	229	4
HEVC to HEVC	4k	4	1	8	0	60	95	283	4
HEVC to AV1	4k	4	1	8	0	40	82	260	4
VP9 to AVC	4k	4	1	8	0	61	94	235	4
VP9 to HEVC	4k	4	1	8	0	61	90	291	4
VP9 to AV1	4k	4	1	8	0	48	88	264	4
AVC to AVC	4k	8	1	8	0	68	100	207	3
AVC to HEVC	4k	8	1	8	0	76	99	269	3
AVC to AV1	4k	8	1	8	0	70	100	265	3
HEVC to AVC	4k	8	1	8	0	63	100	216	3
HEVC to HEVC	4k	8	1	8	0	69	100	276	3
HEVC to AV1	4k	8	1	8	0	60	98	270	3
VP9 to AVC	4k	8	1	8	0	67	98	230	3
VP9 to HEVC	4k	8	1	8	0	74	100	295	3
VP9 to AV1	4k	8	1	8	0	62	99	280	3
AVC to YUV	4k	1	0	10	1	41	0	216	6
HEVC to YUV	4k	1	0	10	1	43	0	220	6
VP9 to YUV	4k	1	0	10	1	24	0	156	2
AVC to YUV	4k	16	0	10	0	97	0	281	0
HEVC to YUV	4k	16	0	10	0	99	0	280	0
VP9 to YUV	4k	16	0	10	0	100	0	496	0
YUV to AVC	4k	1	0	10	1	0	55	173	26
YUV to HEVC	4k	1	0	10	1	0	49	168	27
YUV to AV1	4k	1	0	10	1	0	58	171	27
YUV to AVC	4k	4	0	10	0	0	81	236	16
YUV to HEVC	4k	4	0	10	0	0	72	244	18
YUV to AV1	4k	4	0	10	0	0	84	248	17
AVC to YUV	1080p	1	0	8	1	41	0	816	12
HEVC to YUV	1080p	1	0	8	1	46	0	864	12
VP9 to YUV	1080p	1	0	8	1	22	0	556	3
AVC to YUV	1080p	40	0	8	0	90	0	1726	0
HEVC to YUV	1080p	40	0	8	0	98	0	1888	0
VP9 to YUV	1080p	40	0	8	0	76	0	1749	0
YUV to AVC	1080p	1	0	8	1	0	61	785	15

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	59	795	18
YUV to AV1	1080p	1	0	8	1	0	60	698	13
YUV to AVC	1080p	32	0	8	0	0	99	1304	2
YUV to HEVC	1080p	32	0	8	0	0	99	1384	2
YUV to AV1	1080p	32	0	8	0	0	99	1196	2
AVC to AVC	1080p	1	1	8	0	71	86	997	15
AVC to HEVC	1080p	1	1	8	0	72	85	1078	16
AVC to AV1	1080p	1	1	8	0	59	86	948	17
HEVC to AVC	1080p	1	1	8	0	63	88	989	14
HEVC to HEVC	1080p	1	1	8	0	68	89	1104	14
HEVC to AV1	1080p	1	1	8	0	55	87	942	15
VP9 to AVC	1080p	1	1	8	0	22	41	517	4
VP9 to HEVC	1080p	1	1	8	0	22	39	540	4
VP9 to AV1	1080p	1	1	8	0	21	46	539	5
AVC to AVC	1080p	32	1	8	0	75	99	964	1
AVC to HEVC	1080p	32	1	8	0	83	99	1089	1
AVC to AV1	1080p	32	1	8	0	77	99	1053	1
HEVC to AVC	1080p	32	1	8	0	72	99	1007	1
HEVC to HEVC	1080p	32	1	8	0	76	100	1137	1
HEVC to AV1	1080p	32	1	8	0	67	99	1075	1
VP9 to AVC	1080p	32	1	8	0	66	99	1099	1
VP9 to HEVC	1080p	32	1	8	0	69	99	1240	1
VP9 to AV1	1080p	32	1	8	0	59	99	1142	1
AVC to YUV	1080p	1	0	10	1	29	0	658	6
HEVC to YUV	1080p	1	0	10	1	27	0	709	6
VP9 to YUV	1080p	1	0	10	1	22	0	449	4
AVC to YUV	1080p	40	0	10	0	70	0	1174	0
HEVC to YUV	1080p	40	0	10	0	70	0	1176	0
VP9 to YUV	1080p	40	0	10	0	76	0	1163	0
YUV to AVC	1080p	1	0	10	1	0	43	554	21
YUV to HEVC	1080p	1	0	10	1	0	40	555	25
YUV to AV1	1080p	1	0	10	1	0	44	509	25
YUV to AVC	1080p	32	0	10	0	0	68	878	2
YUV to HEVC	1080p	32	0	10	0	0	63	875	2
YUV to AV1	1080p	32	0	10	0	0	72	877	3
AVC to YUV	720p	1	0	8	1	28	0	1150	12
HEVC to YUV	720p	1	0	8	1	30	0	1179	10
VP9 to YUV	720p	1	0	8	1	19	0	1020	4
AVC to YUV	720p	100	0	8	0	60	0	2699	0

T1A – Libxcoder Throughput

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	71	0	3060	0
VP9 to YUV	720p	100	0	8	0	52	0	2802	0
YUV to AVC	720p	1	0	8	1	0	40	1158	14
YUV to HEVC	720p	1	0	8	1	0	39	1133	14
YUV to AV1	720p	1	0	8	1	0	41	1047	16
YUV to AVC	720p	64	0	8	0	0	83	2511	5
YUV to HEVC	720p	64	0	8	0	0	85	2581	5
YUV to AV1	720p	64	0	8	0	0	81	2150	4
AVC to AVC	720p	1	1	8	0	39	51	1458	15
AVC to HEVC	720p	1	1	8	0	38	50	1461	16
AVC to AV1	720p	1	1	8	0	31	51	1221	16
HEVC to AVC	720p	1	1	8	0	42	53	1500	14
HEVC to HEVC	720p	1	1	8	0	42	53	1496	14
HEVC to AV1	720p	1	1	8	0	32	51	1250	13
VP9 to AVC	720p	1	1	8	0	19	34	1002	7
VP9 to HEVC	720p	1	1	8	0	19	33	984	7
VP9 to AV1	720p	1	1	8	0	19	39	977	7
AVC to AVC	720p	64	1	8	0	66	83	2059	0
AVC to HEVC	720p	64	1	8	0	63	78	2129	0
AVC to AV1	720p	64	1	8	0	46	68	1745	0
HEVC to AVC	720p	64	1	8	0	64	85	2056	0
HEVC to HEVC	720p	64	1	8	0	61	78	2117	0
HEVC to AV1	720p	64	1	8	0	47	70	1763	0
VP9 to AVC	720p	64	1	8	0	56	83	2258	0
VP9 to HEVC	720p	64	1	8	0	53	79	2316	0
VP9 to AV1	720p	64	1	8	0	39	72	1879	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	57.62	58.02	56.97	0.03
YUV to HEVC	8k	1	55.31	60.31	53.95	1.01
YUV to AVC	4k	1	15.17	17.27	14.77	0.07
YUV to HEVC	4k	1	16.38	18.22	15.36	0.11
YUV to AV1	4k	1	21.55	25.29	16.07	0.65
YUV to AVC	4k	4	15.31	19.14	14.75	0.22
YUV to HEVC	4k	4	17.07	20.99	15.32	0.83
YUV to AV1	4k	4	21.84	27.83	16.07	0.88
YUV to AVC	4k	8	16.28	31.71	14.77	4.51
YUV to HEVC	4k	8	20.07	35.93	15.51	17.14
YUV to AV1	4k	8	37.24	45.92	24.77	9.27
YUV to AVC	1080p	1	4.51	5.43	4.23	0.03
YUV to HEVC	1080p	1	4.8	5.34	4.47	0.02
YUV to AV1	1080p	1	6.49	7.17	4.75	0.06
YUV to AVC	1080p	32	5.96	11.65	4.26	1.41
YUV to HEVC	1080p	32	6.77	11.95	4.47	2.16
YUV to AV1	1080p	32	40.99	48.32	34.63	3.27
YUV to AVC	720p	1	2.52	3.08	2.15	0.01
YUV to HEVC	720p	1	2.62	2.9	2.24	0.01
YUV to AV1	720p	1	3.53	4.34	2.73	0.02
YUV to AVC	720p	64	3.64	7.16	2.18	0.51
YUV to HEVC	720p	64	3.89	7.47	2.34	0.62
YUV to AV1	720p	64	39.31	47.07	33.93	1.78

4. T1A – Decoder PPU Scaling

4.1 Decoding

4.1.1 Description

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

Bitstream is decoded by hardware decoder and scaled to 224x224 with decoder post processing unit.

Decoded YUV is kept on device.

The YUV frame is converted to RGBA format with 2D Engine.

The RGBA frame is read out through PCIe and written into an output file.

4.1.2 Command Line

```
ffmpeg -vsync 0 -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params
out=hw:scale0=224x224:multicoreJointMode=<resolution=8k?1:0> -f concat
-safe 0 -i /media/ramdisk/input.list -vf
ni_quadra_scale=iw:ih:format=rgba,hwdownload,format=rgba -f null -
```

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

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

4.2 Decoder PPU Scaling Performance Results

TYPE	RESOLUTION	JOBS	DEC_LOAD	SCALER_LOAD	FPS	CPU
AVC to RGBA	8k	1	90	0	140	6
HEVC to RGBA	8k	1	90	0	153	7
VP9 to RGBA	8k	1	22	0	40	2
AVC to RGBA	4k	1	22	0	144	7
AVC to RGBA	4k	16	95	0	582	2
HEVC to RGBA	4k	1	21	0	170	9
HEVC to RGBA	4k	16	92	0	662	2
VP9 to RGBA	4k	1	22	0	166	4
VP9 to RGBA	4k	16	98	0	680	1
AVC to RGBA	1080p	40	94	3	1883	1
HEVC to RGBA	1080p	40	95	3	2011	1
VP9 to RGBA	1080p	40	88	4	2437	0
AVC to RGBA	720p	100	61	4	2659	0
HEVC to RGBA	720p	100	70	4	3016	0
VP9 to RGBA	720p	64	48	4	2760	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	32	93	3	424	2
AVC to HEVC	8	28	88	2	432	2
AVC to AV1	8	20	82	1	360	2
HEVC to AVC	8	33	91	3	432	3
HEVC to HEVC	8	29	87	2	440	3
HEVC to AV1	8	21	84	1	360	2
VP9 to AVC	8	39	93	3	431	2
VP9 to HEVC	8	34	89	2	440	2
VP9 to AV1	8	24	82	1	360	2

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 to the 2D Engine and formatted to YUV with ni_quadra_scale.

YUV frame is 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,ni_quadra_scale=iw:ih:format=yuv420p" -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	JOB	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	27	87	34
RGBA to HEVC	4k	1	1	25	88	34
RGBA to AV1	4k	1	1	28	88	34
RGBA to AVC	4k	4	0	49	146	23
RGBA to HEVC	4k	4	0	49	160	25
RGBA to AV1	4k	4	0	54	158	26
RGBA to AVC	4k	8	0	79	152	11
RGBA to HEVC	4k	8	0	68	160	12
RGBA to AV1	4k	8	0	71	160	13
RGBA to AVC	1080p	1	1	21	283	19
RGBA to HEVC	1080p	1	1	20	283	20
RGBA to AV1	1080p	1	1	23	274	20
RGBA to AVC	1080p	16	0	78	592	6
RGBA to HEVC	1080p	16	0	70	624	5
RGBA to AV1	1080p	16	0	73	624	7
RGBA to AVC	1080p	32	0	76	640	4
RGBA to HEVC	1080p	32	0	69	670	4
RGBA to AV1	1080p	32	0	72	672	5
RGBA to AVC	720p	1	1	15	459	28
RGBA to HEVC	720p	1	1	15	456	28
RGBA to AV1	720p	1	1	17	433	27
RGBA to AVC	720p	16	0	40	1047	35
RGBA to HEVC	720p	16	0	38	1043	34
RGBA to AV1	720p	16	0	42	1038	31
RGBA to AVC	720p	32	0	44	1100	25
RGBA to HEVC	720p	32	0	42	1098	25
RGBA to AV1	720p	32	0	45	1083	22

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	98	305	20
YUV to HEVC	4k	4	0	0	1	97	336	18
YUV to AV1	4k	4	0	0	1	97	292	17
YUV to AVC	4k	4	0	0	2	97	305	20
YUV to HEVC	4k	4	0	0	2	98	172	8
YUV to AV1	4k	4	0	0	2	99	140	7
YUV to AVC	4k	4	0	0	3	96	308	19
YUV to HEVC	4k	4	0	0	3	99	100	5
YUV to AV1	4k	4	0	0	3	99	76	4
YUV to AVC	4k	4	0	1	1	98	188	7
YUV to HEVC	4k	4	0	1	1	98	240	12
YUV to AVC	4k	4	0	1	2	98	188	8
YUV to HEVC	4k	4	0	1	2	99	108	5
YUV to AVC	4k	4	0	1	3	98	188	8
YUV to HEVC	4k	4	0	1	3	97	68	5
YUV to AVC	4k	4	16	0	1	99	196	9
YUV to HEVC	4k	4	16	0	1	99	228	10
YUV to AV1	4k	4	16	0	1	100	200	9
YUV to AVC	4k	4	16	0	2	100	196	9
YUV to HEVC	4k	4	16	0	2	100	136	6
YUV to AV1	4k	4	16	0	2	100	116	5
YUV to AVC	4k	4	16	0	3	99	196	8
YUV to HEVC	4k	4	16	0	3	100	88	4
YUV to AV1	4k	4	16	0	3	100	72	4
YUV to AVC	4k	4	16	1	1	100	140	7
YUV to HEVC	4k	4	16	1	1	99	180	8
YUV to AVC	4k	4	16	1	2	99	140	7
YUV to HEVC	4k	4	16	1	2	99	96	4
YUV to AVC	4k	4	16	1	3	100	140	7
YUV to HEVC	4k	4	16	1	3	99	64	3
YUV to AVC	4k	4	40	0	1	100	196	9
YUV to HEVC	4k	4	40	0	1	100	224	11
YUV to AV1	4k	4	40	0	1	99	196	9
YUV to AVC	4k	4	40	0	2	100	196	9
YUV to HEVC	4k	4	40	0	2	100	136	6
YUV to AV1	4k	4	40	0	2	99	112	6
YUV to AVC	4k	4	40	0	3	100	196	8
YUV to HEVC	4k	4	40	0	3	99	88	4

T1A – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	4k	4	40	0	3	100	72	4
YUV to AVC	4k	4	40	1	1	99	137	6
YUV to HEVC	4k	4	40	1	1	99	176	9
YUV to AVC	4k	4	40	1	2	99	139	6
YUV to HEVC	4k	4	40	1	2	100	92	5
YUV to AVC	4k	4	40	1	3	100	138	6
YUV to HEVC	4k	4	40	1	3	99	64	3
YUV to AVC	1080p	20	0	0	1	99	1280	4
YUV to HEVC	1080p	20	0	0	1	99	1360	5
YUV to AV1	1080p	20	0	0	1	100	1180	3
YUV to AVC	1080p	20	0	0	2	99	1280	4
YUV to HEVC	1080p	20	0	0	2	100	700	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	2
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	99	760	3
YUV to HEVC	1080p	20	0	1	2	100	440	2
YUV to AVC	1080p	20	0	1	3	99	760	2
YUV to HEVC	1080p	20	0	1	3	100	280	1
YUV to AVC	1080p	20	16	0	1	99	700	2
YUV to HEVC	1080p	20	16	0	1	99	820	3
YUV to AV1	1080p	20	16	0	1	99	720	2
YUV to AVC	1080p	20	16	0	2	99	700	2
YUV to HEVC	1080p	20	16	0	2	99	509	2
YUV to AV1	1080p	20	16	0	2	99	420	2
YUV to AVC	1080p	20	16	0	3	99	700	2
YUV to HEVC	1080p	20	16	0	3	99	340	1
YUV to AV1	1080p	20	16	0	3	99	267	1
YUV to AVC	1080p	20	16	1	1	99	500	2
YUV to HEVC	1080p	20	16	1	1	100	659	2
YUV to AVC	1080p	20	16	1	2	100	500	2
YUV to HEVC	1080p	20	16	1	2	100	360	1
YUV to AVC	1080p	20	16	1	3	99	500	2
YUV to HEVC	1080p	20	16	1	3	100	240	1
YUV to AVC	1080p	20	40	0	1	99	699	2
YUV to HEVC	1080p	20	40	0	1	99	814	3

T1A – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	40	0	1	99	708	2
YUV to AVC	1080p	20	40	0	2	100	699	2
YUV to HEVC	1080p	20	40	0	2	99	500	2
YUV to AV1	1080p	20	40	0	2	99	420	2
YUV to AVC	1080p	20	40	0	3	99	697	2
YUV to HEVC	1080p	20	40	0	3	100	340	2
YUV to AV1	1080p	20	40	0	3	99	260	2
YUV to AVC	1080p	20	40	1	1	99	500	2
YUV to HEVC	1080p	20	40	1	1	99	640	2
YUV to AVC	1080p	20	40	1	2	100	500	2
YUV to HEVC	1080p	20	40	1	2	100	360	2
YUV to AVC	1080p	20	40	1	3	100	500	2
YUV to HEVC	1080p	20	40	1	3	100	240	1
YUV to AVC	720p	40	0	0	1	81	2438	11
YUV to HEVC	720p	40	0	0	1	84	2482	11
YUV to AV1	720p	40	0	0	1	81	2040	8
YUV to AVC	720p	40	0	0	2	82	2433	11
YUV to HEVC	720p	40	0	0	2	99	1520	3
YUV to AV1	720p	40	0	0	2	99	1202	2
YUV to AVC	720p	40	0	0	3	81	2436	11
YUV to HEVC	720p	40	0	0	3	99	920	2
YUV to AV1	720p	40	0	0	3	99	669	1
YUV to AVC	720p	40	0	1	1	99	1711	4
YUV to HEVC	720p	40	0	1	1	99	2080	6
YUV to AVC	720p	40	0	1	2	99	1715	4
YUV to HEVC	720p	40	0	1	2	100	960	2
YUV to AVC	720p	40	0	1	3	99	1709	4
YUV to HEVC	720p	40	0	1	3	99	640	1
YUV to AVC	720p	40	16	0	1	90	1319	4
YUV to HEVC	720p	40	16	0	1	79	1280	4
YUV to AV1	720p	40	16	0	1	69	1000	3
YUV to AVC	720p	40	16	0	2	90	1317	4
YUV to HEVC	720p	40	16	0	2	99	1080	2
YUV to AV1	720p	40	16	0	2	99	880	2
YUV to AVC	720p	40	16	0	3	89	1320	4
YUV to HEVC	720p	40	16	0	3	100	720	2
YUV to AV1	720p	40	16	0	3	99	560	1
YUV to AVC	720p	40	16	1	1	99	1064	2
YUV to HEVC	720p	40	16	1	1	97	1270	4

T1A – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	40	16	1	2	99	1062	2
YUV to HEVC	720p	40	16	1	2	99	760	2
YUV to AVC	720p	40	16	1	3	99	1061	2
YUV to HEVC	720p	40	16	1	3	100	520	1
YUV to AVC	720p	40	40	0	1	86	1240	4
YUV to HEVC	720p	40	40	0	1	74	1238	4
YUV to AV1	720p	40	40	0	1	67	965	3
YUV to AVC	720p	40	40	0	2	87	1273	4
YUV to HEVC	720p	40	40	0	2	99	1040	3
YUV to AV1	720p	40	40	0	2	99	866	2
YUV to AVC	720p	40	40	0	3	88	1243	4
YUV to HEVC	720p	40	40	0	3	99	720	2
YUV to AV1	720p	40	40	0	3	99	560	1
YUV to AVC	720p	40	40	1	1	99	1040	2
YUV to HEVC	720p	40	40	1	1	88	1214	4
YUV to AVC	720p	40	40	1	2	99	1040	3
YUV to HEVC	720p	40	40	1	2	99	760	2
YUV to AVC	720p	40	40	1	3	99	1040	2
YUV to HEVC	720p	40	40	1	3	99	520	1

8. T1A – Capped CRF

8.1 Encoding with lookaheadDepth

8.1.1 Description

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

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

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

8.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

8.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	16	0	1	19	99	700	2
YUV to HEVC	1080p	20	16	0	1	19	99	820	3
YUV to AV1	1080p	20	16	0	1	19	99	720	2
YUV to AVC	1080p	20	16	0	2	19	99	700	2
YUV to HEVC	1080p	20	16	0	2	19	100	510	2
YUV to AV1	1080p	20	16	0	2	19	100	420	2
YUV to AVC	1080p	20	16	0	3	19	99	700	2
YUV to HEVC	1080p	20	16	0	3	19	100	340	1
YUV to AV1	1080p	20	16	0	3	19	100	266	1
YUV to AVC	1080p	20	16	1	1	19	100	500	2
YUV to HEVC	1080p	20	16	1	1	19	100	658	2
YUV to AVC	1080p	20	16	1	2	19	100	500	2
YUV to HEVC	1080p	20	16	1	2	19	99	360	1
YUV to AVC	1080p	20	16	1	3	19	100	500	2
YUV to HEVC	1080p	20	16	1	3	19	100	240	2
YUV to AVC	1080p	20	40	0	1	19	99	697	2
YUV to HEVC	1080p	20	40	0	1	19	99	818	3
YUV to AV1	1080p	20	40	0	1	19	99	720	2
YUV to AVC	1080p	20	40	0	2	19	99	698	2
YUV to HEVC	1080p	20	40	0	2	19	99	506	2
YUV to AV1	1080p	20	40	0	2	19	99	420	2
YUV to AVC	1080p	20	40	0	3	19	99	700	2
YUV to HEVC	1080p	20	40	0	3	19	100	340	2
YUV to AV1	1080p	20	40	0	3	19	100	262	1
YUV to AVC	1080p	20	40	1	1	19	99	500	2
YUV to HEVC	1080p	20	40	1	1	19	99	660	2
YUV to AVC	1080p	20	40	1	2	19	100	500	2
YUV to HEVC	1080p	20	40	1	2	19	100	360	2
YUV to AVC	1080p	20	40	1	3	19	99	500	2
YUV to HEVC	1080p	20	40	1	3	19	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	820	3
YUV to AV1	1080p	20	16	0	1	23	99	720	2
YUV to AVC	1080p	20	16	0	2	23	99	700	2
YUV to HEVC	1080p	20	16	0	2	23	99	508	2
YUV to AV1	1080p	20	16	0	2	23	100	420	2
YUV to AVC	1080p	20	16	0	3	23	100	700	2
YUV to HEVC	1080p	20	16	0	3	23	99	340	2

T1A – Capped CRF

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	3	23	99	268	1
YUV to AVC	1080p	20	16	1	1	23	99	500	2
YUV to HEVC	1080p	20	16	1	1	23	100	656	2
YUV to AVC	1080p	20	16	1	2	23	100	500	2
YUV to HEVC	1080p	20	16	1	2	23	100	360	1
YUV to AVC	1080p	20	16	1	3	23	100	500	2
YUV to HEVC	1080p	20	16	1	3	23	99	240	1
YUV to AVC	1080p	20	40	0	1	23	99	699	2
YUV to HEVC	1080p	20	40	0	1	23	99	813	3
YUV to AV1	1080p	20	40	0	1	23	99	713	2
YUV to AVC	1080p	20	40	0	2	23	99	699	2
YUV to HEVC	1080p	20	40	0	2	23	100	500	2
YUV to AV1	1080p	20	40	0	2	23	99	420	2
YUV to AVC	1080p	20	40	0	3	23	99	699	2
YUV to HEVC	1080p	20	40	0	3	23	99	340	2
YUV to AV1	1080p	20	40	0	3	23	100	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
YUV to AVC	1080p	20	40	1	2	23	100	500	2
YUV to HEVC	1080p	20	40	1	2	23	100	360	2
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	16	0	1	27	99	700	2
YUV to HEVC	1080p	20	16	0	1	27	99	820	2
YUV to AV1	1080p	20	16	0	1	27	99	720	2
YUV to AVC	1080p	20	16	0	2	27	99	700	2
YUV to HEVC	1080p	20	16	0	2	27	99	510	2
YUV to AV1	1080p	20	16	0	2	27	100	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	270	1
YUV to AVC	1080p	20	16	1	1	27	100	500	2
YUV to HEVC	1080p	20	16	1	1	27	99	658	2
YUV to AVC	1080p	20	16	1	2	27	99	500	2
YUV to HEVC	1080p	20	16	1	2	27	99	360	1
YUV to AVC	1080p	20	16	1	3	27	100	500	2
YUV to HEVC	1080p	20	16	1	3	27	99	240	1
YUV to AVC	1080p	20	40	0	1	27	99	698	2
YUV to HEVC	1080p	20	40	0	1	27	99	820	3

T1A – Capped CRF

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	40	0	1	27	99	720	2
YUV to AVC	1080p	20	40	0	2	27	100	698	2
YUV to HEVC	1080p	20	40	0	2	27	100	519	2
YUV to AV1	1080p	20	40	0	2	27	99	420	2
YUV to AVC	1080p	20	40	0	3	27	99	698	2
YUV to HEVC	1080p	20	40	0	3	27	99	340	2
YUV to AV1	1080p	20	40	0	3	27	100	261	1
YUV to AVC	1080p	20	40	1	1	27	99	500	2
YUV to HEVC	1080p	20	40	1	1	27	99	660	2
YUV to AVC	1080p	20	40	1	2	27	99	500	2
YUV to HEVC	1080p	20	40	1	2	27	99	360	2
YUV to AVC	1080p	20	40	1	3	27	100	500	2
YUV to HEVC	1080p	20	40	1	3	27	100	240	1

9. T1A – Inplace Overlay

9.1 Transcoding

9.1.1 Description

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

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

9.1.2 Command line

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

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

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

Input Video: 1080p

Input Image: 128x128

9.2 Inplace Overlay Performance Results

TYPE	JOB5	FPS	CPU	DEC_LOAD	ENC_LOAD	SCALER_LOAD
AVC to AVC	1	295	9	12	20	0
AVC to HEVC	1	314	10	13	21	0
AVC to AV1	1	268	9	11	20	0
HEVC to AVC	1	292	12	13	20	0
HEVC to HEVC	1	310	13	14	21	0
HEVC to AV1	1	267	12	11	21	0
VP9 to AVC	1	295	9	16	20	0
VP9 to HEVC	1	314	9	18	20	0
VP9 to AV1	1	268	9	16	21	0
AVC to AVC	16	1040	2	73	95	5
AVC to HEVC	16	1152	2	80	95	6
AVC to AV1	16	1104	2	73	95	5
HEVC to AVC	16	1076	3	72	94	5
HEVC to HEVC	16	1200	4	78	94	6
HEVC to AV1	16	1136	3	71	97	5
VP9 to AVC	16	1040	2	82	93	5
VP9 to HEVC	16	1167	2	90	93	7
VP9 to AV1	16	1120	2	83	93	5
AVC to AVC	32	960	1	75	95	5
AVC to HEVC	32	1088	1	81	93	5
AVC to AV1	32	1056	1	78	95	5
HEVC to AVC	32	1014	1	72	93	5
HEVC to HEVC	32	1151	1	80	94	6
HEVC to AV1	32	1092	1	72	92	5
VP9 to AVC	32	992	1	82	94	5
VP9 to HEVC	32	1120	1	88	92	6
VP9 to AV1	32	1088	1	84	94	5

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

10.1 Encoding

10.1.1 Description

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

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

The YUV frame is encoded with hardware encoder.

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

10.1.2 Command line

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

<num_jobs> = number of instances running concurrently

10.2 Multi Thread P2P DMA on AMD GPU Performance Results

TYPE	RES	JOBS	Frames	FPS	CPU	Enc Load	P2P MEM	Latency Avg	Latency Dev
P2A	720p	1	1001	30	1.1	0.25	0.5	1.9	0
P2H	720p	1	1001	30	1.1	0.25	0.5	1.9	0
P2A	720p	180	180180	29.6	0.71	46	56	2.14	0.06
P2H	720p	180	180180	29.8	0.7	45.5	56	2.3	5.23
P2A	1080p	1	1001	29	1.2	0.5	0.75	3.61	0
P2H	1080p	1	1001	30	1.2	0.5	0.75	3.37	0
P2A	1080p	80	80080	29.8	0.85	46	50	3.8	0.19
P2H	1080p	80	80080	29.6	0.84	43	50	3.59	0.21

11. T1A – AI

11.1 AI Model

11.1.1 Description

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

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

aiperf performs any data format conversion expected by the hardware.

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

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

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

11.1.2 Command line

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

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

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

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

11.1.3 AI Model Performance Results

Model	Session Number	Loops	Channel Order	File Format	File Type	Input Size	FPS per session
yolov5s_640	8	10000	rgb	nchw	image	640x640x3	77
yolov5s_320	8	10000	rgb	nchw	image	320x320x3	277
deeplabv3_FRP	8	10000	rgb	nchw	image	257x257x3	360
resnet50	8	10000	rgb	nchw	image	224x224x3	228
mobilenetv2	8	10000	rgb	nchw	image	224x224x3	1108
deeplabv3	8	10000	rgb	nchw	image	257x257x3	160
yolov4	8	10000	bgr	nchw	image	416x416x3	257
fsrcnn	8	2000	bgr	nchw	image	360x640x1	31
BiSeNetv1	8	10000	rgb	nchw	image	512x512x3	76
HrNet	8	10000	rgb	nchw	image	256x192x3	74
usm_1656x1920	8	10000	rgb	nchw	image	1920x1656x1	243
usm_3240x3840	8	10000	rgb	nchw	image	3840x3240x1	61
lpips	8	2000	rgb	nchw	image	720x480x3	1
PaddleOCR-512_onnx	8	10000	rgb	nchw	image	512x48x3	16
segm32_tflite_kl_mle	8	10000	rgb	nchw	image	256x144x3	837
mobilenetv2_nchw_keras_96x160	1	10000	rgb	nchw	image	96x160x3	2342.5
mobilenetv2_nchw_keras_96x160	8	10000	rgb	nchw	image	96x160x3	2331.78
mobilenetv2_nchw_keras_96x160	16	10000	rgb	nchw	image	96x160x3	2332.76

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:qoffset=-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	24
ROI	network_binary_yolov4_head	1920x1080	32	5
BG	segm32_tflite_nchw_bgr	1920x1080	1	53
BG	segm32_tflite_nchw_bgr	1920x1080	8	35
BG	segm32_tflite_nchw_bgr	1920x1080	32	14
PRE	hw_lanczos_network_binary	1920x1080	8	67
PRE	hw_lanczos_network_binary	1920x1080	16	34
PRE	hw_bicubic_network_binary	1920x1080	8	67
PRE	hw_bicubic_network_binary	1920x1080	16	34

12. T1A – GStreamer XStack Throughput

12.1 Transcoding

12.1.1 Description

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

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

The YUV frame is encoded with hardware encoder.

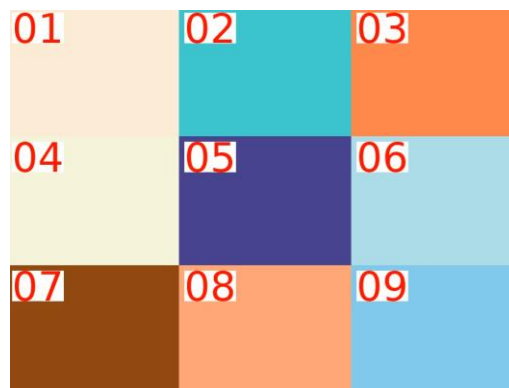
The encoded bitstream is read out through PCIe and written into an output file.

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

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

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

This test is HEVC to AVC only.



Example output in a 3x3 layout with 9 inputs

12.1.2 Command line

See Appendix A: GStreamer XStack Command

12.2 GStreamer XStack Performance Results

Input Res	Grid	Output Res	Cell Size	FPS	CPU	Dec Load	Enc Load	Scaler Load
1920x1080	3x3	1920x1080	640x360	155.1	56	67	14	12
1920x1080	4x4	1920x1080	480x270	104.48	64	81	10	11
1920x1080	4x8	1920x1080	480x135	56.57	75	86	5	6
1920x1080	3x3	3840x2160	1280x720	70.34	27	31	22	12
1920x1080	4x4	3840x2160	960x540	61.31	41	52	22	15
1920x1080	4x8	3840x2160	960x270	46.83	58	79	17	15
1920x1080	3x3	7680x4320	2560x1440	19.94	14	8	22	9
1920x1080	4x4	7680x4320	1920x1080	19.17	17	15	22	10
1920x1080	4x8	7680x4320	1920x540	19.32	26	31	21	11
1920x1080	6x6	1920x1080	320x180	50.11	80	85	5	8
1920x1080	7x7	1920x1080	274x154 276x154 274x156 276x156*	37.11	82	91	4	7

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

13. T1A – GStreamer Ladder Generation

13.1 Transcoding

13.1.1 Description

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

Decoded YUV is split to multiple pads.

The YUV frames are encoded with hardware encoder.

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

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

This test is AVC to HEVC only

13.1.2 Command line

See Appendix C: GStreamer Ladder Command

13.2 GStreamer Ladder Performance Results

Jobs	Outputs	FPS	CPU	Dec Load	Enc Load
1	64	17.9	1001	0	84

14. T1U – FFmpeg Throughput

14.1 Decoding

14.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

14.1.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

14.2 Encoding

14.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

14.2.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

14.3 Transcoding

14.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

14.3.2 Command line

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

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

14.4 FFmpeg Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	89	0	97	10
HEVC to YUV	8k	1	0	8	1	84	0	93	9
VP9 to YUV	8k	1	0	8	1	25	0	30	5
YUV to AVC	8k	1	0	8	1	0	96	58	78
YUV to HEVC	8k	1	0	8	1	0	97	72	97
AVC to AVC	8k	1	1	8	1	72	85	48	7
AVC to HEVC	8k	1	1	8	1	72	96	64	7
HEVC to AVC	8k	1	1	8	1	62	95	47	7
HEVC to HEVC	8k	1	1	8	1	72	100	63	7
VP9 to AVC	8k	1	1	8	1	25	41	29	5
VP9 to HEVC	8k	1	1	8	1	25	40	30	6
AVC to YUV	8k	1	0	10	1	57	0	52	9
HEVC to YUV	8k	1	0	10	1	54	0	50	8
VP9 to YUV	8k	1	0	10	1	24	0	28	7
YUV to AVC	8k	1	0	10	1	0	92	42	118
YUV to HEVC	8k	1	0	10	1	0	60	43	122
AVC to YUV	4k	1	0	8	1	63	0	288	17
HEVC to YUV	4k	1	0	8	1	56	0	296	19
VP9 to YUV	4k	1	0	8	1	24	0	108	8
AVC to YUV	4k	16	0	8	0	99	0	420	2
HEVC to YUV	4k	16	0	8	0	99	0	450	2
VP9 to YUV	4k	16	0	8	0	100	0	400	2
YUV to AVC	4k	1	0	8	1	0	95	263	33
YUV to HEVC	4k	1	0	8	1	0	97	288	33
YUV to AV1	4k	1	0	8	1	0	96	249	31
YUV to AVC	4k	4	0	8	0	0	98	268	15
YUV to HEVC	4k	4	0	8	0	0	97	291	14
YUV to AV1	4k	4	0	8	0	0	96	252	13
YUV to AVC	4k	8	0	8	0	0	99	280	7
YUV to HEVC	4k	8	0	8	0	0	100	296	7
YUV to AV1	4k	8	0	8	0	0	99	257	7
AVC to AVC	4k	1	1	8	1	69	90	197	13
AVC to HEVC	4k	1	1	8	1	66	87	242	14
AVC to AV1	4k	1	1	8	1	58	89	226	14
HEVC to AVC	4k	1	1	8	1	51	87	194	14
HEVC to HEVC	4k	1	1	8	1	56	87	239	16
HEVC to AV1	4k	1	1	8	1	52	95	222	15
VP9 to AVC	4k	1	1	8	1	24	38	108	8
VP9 to HEVC	4k	1	1	8	1	23	35	107	8

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	41	108	8
AVC to AVC	4k	4	1	8	0	63	97	216	4
AVC to HEVC	4k	4	1	8	0	68	97	268	5
AVC to AV1	4k	4	1	8	0	58	93	240	5
HEVC to AVC	4k	4	1	8	0	55	97	216	5
HEVC to HEVC	4k	4	1	8	0	57	97	260	5
HEVC to AV1	4k	4	1	8	0	48	97	240	5
VP9 to AVC	4k	4	1	8	0	64	98	216	4
VP9 to HEVC	4k	4	1	8	0	71	97	260	4
VP9 to AV1	4k	4	1	8	0	63	96	240	4
AVC to AVC	4k	8	1	8	0	67	99	195	3
AVC to HEVC	4k	8	1	8	0	75	100	254	3
AVC to AV1	4k	8	1	8	0	65	100	240	3
HEVC to AVC	4k	8	1	8	0	61	99	195	3
HEVC to HEVC	4k	8	1	8	0	65	100	248	3
HEVC to AV1	4k	8	1	8	0	59	99	240	3
VP9 to AVC	4k	8	1	8	0	66	99	202	2
VP9 to HEVC	4k	8	1	8	0	73	100	256	2
VP9 to AV1	4k	8	1	8	0	63	99	248	3
AVC to YUV	4k	1	0	10	0	47	0	196	9
HEVC to YUV	4k	1	0	10	0	47	0	196	10
VP9 to YUV	4k	1	0	10	0	24	0	126	5
AVC to YUV	4k	16	0	10	0	97	0	255	1
HEVC to YUV	4k	16	0	10	0	99	0	254	1
VP9 to YUV	4k	16	0	10	0	98	0	434	1
YUV to AVC	4k	1	0	10	0	0	75	183	57
YUV to HEVC	4k	1	0	10	0	0	69	204	64
YUV to AV1	4k	1	0	10	0	0	76	194	60
YUV to AVC	4k	4	0	10	0	0	92	194	25
YUV to HEVC	4k	4	0	10	0	0	83	231	36
YUV to AV1	4k	4	0	10	0	0	90	226	34
AVC to YUV	1080p	1	0	8	1	50	0	801	19
HEVC to YUV	1080p	1	0	8	1	54	0	793	27
VP9 to YUV	1080p	1	0	8	1	22	0	452	8
AVC to YUV	1080p	40	0	8	0	98	0	1535	1
HEVC to YUV	1080p	40	0	8	0	97	0	1524	1
VP9 to YUV	1080p	40	0	8	0	96	0	1679	0
YUV to AVC	1080p	1	0	8	1	0	70	787	24
YUV to HEVC	1080p	1	0	8	1	0	66	785	22
YUV to AV1	1080p	1	0	8	1	0	69	717	22

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1120	2
YUV to HEVC	1080p	32	0	8	0	0	99	1184	2
YUV to AV1	1080p	32	0	8	0	0	99	1024	2
AVC to AVC	1080p	1	1	8	1	48	64	702	18
AVC to HEVC	1080p	1	1	8	1	48	61	725	17
AVC to AV1	1080p	1	1	8	1	45	68	669	18
HEVC to AVC	1080p	1	1	8	1	50	66	729	26
HEVC to HEVC	1080p	1	1	8	1	52	64	740	26
HEVC to AV1	1080p	1	1	8	1	46	68	692	26
VP9 to AVC	1080p	1	1	8	1	22	40	450	7
VP9 to HEVC	1080p	1	1	8	1	22	38	448	7
VP9 to AV1	1080p	1	1	8	1	22	44	448	9
AVC to AVC	1080p	32	1	8	0	75	99	864	1
AVC to HEVC	1080p	32	1	8	0	84	99	979	1
AVC to AV1	1080p	32	1	8	0	78	99	928	1
HEVC to AVC	1080p	32	1	8	0	72	99	896	1
HEVC to HEVC	1080p	32	1	8	0	76	99	1007	1
HEVC to AV1	1080p	32	1	8	0	71	99	933	1
VP9 to AVC	1080p	32	1	8	0	62	99	992	1
VP9 to HEVC	1080p	32	1	8	0	68	99	1109	1
VP9 to AV1	1080p	32	1	8	0	55	99	992	1
AVC to YUV	1080p	1	0	10	0	35	0	630	10
HEVC to YUV	1080p	1	0	10	0	33	0	640	10
VP9 to YUV	1080p	1	0	10	0	22	0	458	8
AVC to YUV	1080p	40	0	10	0	79	0	1087	0
HEVC to YUV	1080p	40	0	10	0	82	0	1099	0
VP9 to YUV	1080p	40	0	10	0	92	0	1722	0
YUV to AVC	1080p	1	0	10	0	0	50	556	28
YUV to HEVC	1080p	1	0	10	0	0	46	551	28
YUV to AV1	1080p	1	0	10	0	0	49	506	27
YUV to AVC	1080p	32	0	10	0	0	76	835	9
YUV to HEVC	1080p	32	0	10	0	0	71	848	8
YUV to AV1	1080p	32	0	10	0	0	80	832	9
AVC to YUV	720p	1	0	8	1	36	0	1228	17
HEVC to YUV	720p	1	0	8	1	41	0	1217	24
VP9 to YUV	720p	1	0	8	1	20	0	847	10
AVC to YUV	720p	100	0	8	0	70	0	2317	0
HEVC to YUV	720p	100	0	8	0	80	0	2609	0
VP9 to YUV	720p	100	0	8	0	53	0	2300	0
YUV to AVC	720p	1	0	8	1	0	43	1091	29

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	42	1075	22
YUV to AV1	720p	1	0	8	1	0	44	971	15
YUV to AVC	720p	64	0	8	0	0	90	2182	7
YUV to HEVC	720p	64	0	8	0	0	91	2249	7
YUV to AV1	720p	64	0	8	0	0	87	1856	5
AVC to AVC	720p	1	1	8	1	25	33	836	14
AVC to HEVC	720p	1	1	8	1	25	33	837	15
AVC to AV1	720p	1	1	8	1	22	35	751	15
HEVC to AVC	720p	1	1	8	1	27	33	853	20
HEVC to HEVC	720p	1	1	8	1	27	33	846	19
HEVC to AV1	720p	1	1	8	1	25	36	761	18
VP9 to AVC	720p	1	1	8	1	20	33	836	11
VP9 to HEVC	720p	1	1	8	1	20	33	840	10
VP9 to AV1	720p	1	1	8	1	20	39	838	11
AVC to AVC	720p	64	1	8	0	65	82	1730	0
AVC to HEVC	720p	64	1	8	0	63	77	1796	0
AVC to AV1	720p	64	1	8	0	52	72	1532	0
HEVC to AVC	720p	64	1	8	0	68	84	1737	0
HEVC to HEVC	720p	64	1	8	0	64	80	1807	0
HEVC to AV1	720p	64	1	8	0	52	73	1539	0
VP9 to AVC	720p	64	1	8	0	55	82	1854	0
VP9 to HEVC	720p	64	1	8	0	51	78	1916	0
VP9 to AV1	720p	64	1	8	0	43	80	1644	0

15. T1U – Libxcoder Throughput

15.1 Decoding

15.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

15.1.2 Command Line

```
~/FFmpegXcoder/libxcoder/build/xcoder -c 0 -s <resolution> -r 1000 -i  
/media/ramdisk/input.<ext> -m <test_type> -o /dev/null -b <bit> -t -x  
keepAliveTimeout=2 -d multicoreJointMode=<*>
```

<test_type> = test codecs. ie. a2y, a2h, etc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

15.2 Encoding

15.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

15.2.2 Command Line

```
~/FFmpegXcoder/libxcoder/build/xcoder -c 0 -s <resolution> -r 1000 -i  
/media/ramdisk/input.yuv -m <test_type> -o /dev/null -b <bit> -t -x  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*>
```

<test_type> = test codecs. ie. a2y, a2h, etc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

15.3 Transcoding

15.3.1 Description

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

Decoded YUV frame is kept on device.

The YUV frame is encoded with hardware encoder.

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

15.3.2 Command line

```
~/FFmpegXcoder/libxcoder/build/xcoder -c 0 -s <resolution> -r 1000 -i  
/media/ramdisk/input.<ext> -m <test_type> -o /dev/null -b <bit> -t -x  
intraPeriod=0:RcEnable=1:bitrate=<*>:keepAliveTimeout=2:multicoreJointM  
ode=<*> -d out=hw:semiplanar0=1:multicoreJointMode=1
```

<test_type> = test codecs. ie. a2y, a2h, etc

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

15.4 Libxcode Throughput Performance Results

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
AVC to YUV	8k	1	0	8	1	86	0	87	9
HEVC to YUV	8k	1	0	8	1	79	0	91	8
VP9 to YUV	8k	1	0	8	1	24	0	30	5
YUV to AVC	8k	1	0	8	1	0	95	57	28
YUV to HEVC	8k	1	0	8	1	0	97	72	33
AVC to AVC	8k	1	1	8	1	80	97	46	6
AVC to HEVC	8k	1	1	8	1	76	100	62	8
HEVC to AVC	8k	1	1	8	1	58	94	45	5
HEVC to HEVC	8k	1	1	8	1	69	98	61	5
VP9 to AVC	8k	1	1	8	1	24	41	28	5
VP9 to HEVC	8k	1	1	8	1	24	40	30	5
AVC to YUV	8k	1	0	10	1	56	0	51	8
HEVC to YUV	8k	1	0	10	1	50	0	50	8
VP9 to YUV	8k	1	0	10	1	24	0	27	5
YUV to AVC	8k	1	0	10	1	0	61	35	41
YUV to HEVC	8k	1	0	10	1	0	50	36	39
AVC to YUV	4k	1	0	8	1	69	0	308	13
HEVC to YUV	4k	1	0	8	1	59	0	316	9
VP9 to YUV	4k	1	0	8	1	24	0	107	5
AVC to YUV	4k	16	0	8	0	100	0	414	1
HEVC to YUV	4k	16	0	8	0	99	0	444	0
VP9 to YUV	4k	16	0	8	0	99	0	398	0
YUV to AVC	4k	1	0	8	1	0	97	253	23
YUV to HEVC	4k	1	0	8	1	0	96	286	27
YUV to AV1	4k	1	0	8	1	0	95	243	22
YUV to AVC	4k	4	0	8	0	0	97	271	9
YUV to HEVC	4k	4	0	8	0	0	97	288	9
YUV to AV1	4k	4	0	8	0	0	97	248	8
YUV to AVC	4k	8	0	8	0	0	99	279	5
YUV to HEVC	4k	8	0	8	0	0	99	296	6
YUV to AV1	4k	8	0	8	0	0	99	257	5
AVC to AVC	4k	1	1	8	0	70	94	197	7
AVC to HEVC	4k	1	1	8	0	74	95	243	9
AVC to AV1	4k	1	1	8	0	63	96	234	8
HEVC to AVC	4k	1	1	8	0	57	96	203	7
HEVC to HEVC	4k	1	1	8	0	62	96	253	8
HEVC to AV1	4k	1	1	8	0	54	97	234	8
VP9 to AVC	4k	1	1	8	0	24	37	107	6

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	0	24	36	107	6
VP9 to AV1	4k	1	1	8	0	23	41	107	5
AVC to AVC	4k	4	1	8	0	63	96	215	5
AVC to HEVC	4k	4	1	8	0	66	95	261	5
AVC to AV1	4k	4	1	8	0	53	90	233	5
HEVC to AVC	4k	4	1	8	0	55	96	209	4
HEVC to HEVC	4k	4	1	8	0	57	92	251	4
HEVC to AV1	4k	4	1	8	0	47	96	225	4
VP9 to AVC	4k	4	1	8	0	63	97	210	4
VP9 to HEVC	4k	4	1	8	0	66	91	259	4
VP9 to AV1	4k	4	1	8	0	55	91	233	4
AVC to AVC	4k	8	1	8	0	68	99	190	3
AVC to HEVC	4k	8	1	8	0	77	100	247	3
AVC to AV1	4k	8	1	8	0	67	99	240	3
HEVC to AVC	4k	8	1	8	0	62	99	195	3
HEVC to HEVC	4k	8	1	8	0	66	99	249	3
HEVC to AV1	4k	8	1	8	0	59	99	240	3
VP9 to AVC	4k	8	1	8	0	66	99	198	3
VP9 to HEVC	4k	8	1	8	0	72	99	255	3
VP9 to AV1	4k	8	1	8	0	65	100	244	3
AVC to YUV	4k	1	0	10	1	47	0	192	8
HEVC to YUV	4k	1	0	10	1	47	0	194	7
VP9 to YUV	4k	1	0	10	1	24	0	123	5
AVC to YUV	4k	16	0	10	0	97	0	254	1
HEVC to YUV	4k	16	0	10	0	100	0	251	0
VP9 to YUV	4k	16	0	10	0	99	0	431	0
YUV to AVC	4k	1	0	10	1	0	59	158	30
YUV to HEVC	4k	1	0	10	1	0	54	160	30
YUV to AV1	4k	1	0	10	1	0	64	161	31
YUV to AVC	4k	4	0	10	0	0	86	209	15
YUV to HEVC	4k	4	0	10	0	0	84	242	18
YUV to AV1	4k	4	0	10	0	0	92	234	17
AVC to YUV	1080p	1	0	8	1	48	0	759	19
HEVC to YUV	1080p	1	0	8	1	55	0	797	14
VP9 to YUV	1080p	1	0	8	1	22	0	443	6
AVC to YUV	1080p	40	0	8	0	99	0	1562	0
HEVC to YUV	1080p	40	0	8	0	98	0	1534	0
VP9 to YUV	1080p	40	0	8	0	95	0	1706	0
YUV to AVC	1080p	1	0	8	1	0	68	742	21

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	62	740	20
YUV to AV1	1080p	1	0	8	1	0	70	728	20
YUV to AVC	1080p	32	0	8	0	0	99	1134	1
YUV to HEVC	1080p	32	0	8	0	0	99	1202	2
YUV to AV1	1080p	32	0	8	0	0	99	1038	2
AVC to AVC	1080p	1	1	8	0	73	88	900	21
AVC to HEVC	1080p	1	1	8	0	73	86	973	22
AVC to AV1	1080p	1	1	8	0	59	87	830	21
HEVC to AVC	1080p	1	1	8	0	69	89	895	15
HEVC to HEVC	1080p	1	1	8	0	73	87	964	16
HEVC to AV1	1080p	1	1	8	0	61	88	827	16
VP9 to AVC	1080p	1	1	8	0	22	40	438	5
VP9 to HEVC	1080p	1	1	8	0	22	37	436	6
VP9 to AV1	1080p	1	1	8	0	22	44	435	8
AVC to AVC	1080p	32	1	8	0	74	99	877	1
AVC to HEVC	1080p	32	1	8	0	84	99	986	1
AVC to AV1	1080p	32	1	8	0	76	99	947	1
HEVC to AVC	1080p	32	1	8	0	72	99	912	1
HEVC to HEVC	1080p	32	1	8	0	77	99	1022	1
HEVC to AV1	1080p	32	1	8	0	70	99	958	1
VP9 to AVC	1080p	32	1	8	0	62	99	1003	0
VP9 to HEVC	1080p	32	1	8	0	68	99	1119	1
VP9 to AV1	1080p	32	1	8	0	56	99	1008	1
AVC to YUV	1080p	1	0	10	1	34	0	600	9
HEVC to YUV	1080p	1	0	10	1	32	0	649	9
VP9 to YUV	1080p	1	0	10	1	22	0	450	6
AVC to YUV	1080p	40	0	10	0	79	0	1114	0
HEVC to YUV	1080p	40	0	10	0	87	0	1119	0
VP9 to YUV	1080p	40	0	10	0	90	0	1746	0
YUV to AVC	1080p	1	0	10	1	0	48	533	25
YUV to HEVC	1080p	1	0	10	1	0	44	516	26
YUV to AV1	1080p	1	0	10	1	0	46	456	22
YUV to AVC	1080p	32	0	10	0	0	77	861	2
YUV to HEVC	1080p	32	0	10	0	0	72	866	2
YUV to AV1	1080p	32	0	10	0	0	85	863	3
AVC to YUV	720p	1	0	8	1	34	0	1100	17
HEVC to YUV	720p	1	0	8	1	39	0	1168	13
VP9 to YUV	720p	1	0	8	1	20	0	839	7
AVC to YUV	720p	100	0	8	0	71	0	2477	0

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	83	0	2718	0
VP9 to YUV	720p	100	0	8	0	60	0	2566	0
YUV to AVC	720p	1	0	8	1	0	43	1052	19
YUV to HEVC	720p	1	0	8	1	0	43	1033	18
YUV to AV1	720p	1	0	8	1	0	45	1000	20
YUV to AVC	720p	64	0	8	0	0	90	2315	5
YUV to HEVC	720p	64	0	8	0	0	92	2387	5
YUV to AV1	720p	64	0	8	0	0	88	1982	4
AVC to AVC	720p	1	1	8	0	44	56	1361	21
AVC to HEVC	720p	1	1	8	0	44	56	1361	22
AVC to AV1	720p	1	1	8	0	36	56	1167	19
HEVC to AVC	720p	1	1	8	0	52	56	1320	16
HEVC to HEVC	720p	1	1	8	0	51	55	1366	16
HEVC to AV1	720p	1	1	8	0	41	55	1158	14
VP9 to AVC	720p	1	1	8	0	20	33	823	8
VP9 to HEVC	720p	1	1	8	0	20	33	828	8
VP9 to AV1	720p	1	1	8	0	20	39	825	8
AVC to AVC	720p	64	1	8	0	68	84	1851	0
AVC to HEVC	720p	64	1	8	0	65	79	1908	0
AVC to AV1	720p	64	1	8	0	50	72	1596	0
HEVC to AVC	720p	64	1	8	0	66	86	1841	0
HEVC to HEVC	720p	64	1	8	0	66	80	1907	0
HEVC to AV1	720p	64	1	8	0	51	73	1591	0
VP9 to AVC	720p	64	1	8	0	57	86	2051	0
VP9 to HEVC	720p	64	1	8	0	56	84	2109	0
VP9 to AV1	720p	64	1	8	0	42	77	1711	0

16. T1U – FFmpeg Latency

16.1 Encoding

16.1.1 Description

Libxcodec is compiled and installed with parameter `--with-latency-display`
`$ bash build.sh --with-latency-display`

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

YUV frame is encoded by hardware encoder.

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

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

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

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

16.1.2 Command Line

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

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

<resolution> is resolution of input

16.2 FFmpeg Latency Performance Results

TYPE	RESOLUTION	JOBS	ELAT_AVG (ms)	ELAT_MAX (ms)	ELAT_MIN (ms)	ELAT_VAR (ms)
YUV to AVC	8k	1	65.15	66	64.5	0.07
YUV to HEVC	8k	1	62.44	68.7	60.69	1.35
YUV to AVC	4k	1	17.14	19.41	16.73	0.07
YUV to HEVC	4k	1	18.57	20.78	17.41	0.12
YUV to AV1	4k	1	24.54	28.63	18.3	0.82
YUV to AVC	4k	4	17.91	21.95	16.67	0.62
YUV to HEVC	4k	4	18.96	22.48	17.36	0.48
YUV to AV1	4k	4	24.84	30.67	18.02	1.05
YUV to AVC	4k	8	21.15	36.97	16.68	19.42
YUV to HEVC	4k	8	24.91	41.5	17.48	31.71
YUV to AV1	4k	8	42.86	75.17	30.49	9.02
YUV to AVC	1080p	1	5.07	6.66	4.76	0.03
YUV to HEVC	1080p	1	5.37	5.9	4.99	0.01
YUV to AV1	1080p	1	7.35	8.36	5.4	0.07
YUV to AVC	1080p	32	7.46	13.37	4.83	2.56
YUV to HEVC	1080p	32	8.96	15.59	5.23	2.83
YUV to AV1	1080p	32	47.17	86.15	39.58	10.07
YUV to AVC	720p	1	2.72	3.5	2.37	0.02
YUV to HEVC	720p	1	2.8	3.21	2.48	0.01
YUV to AV1	720p	1	3.83	4.99	2.93	0.03
YUV to AVC	720p	64	4.44	8.59	2.56	0.89
YUV to HEVC	720p	64	4.94	9.43	2.67	1.04
YUV to AV1	720p	64	45.29	120.75	39.15	4.93

17. T1U – Decoder PPU Scaling

17.1 Decoding

17.1.1 Description

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

Bitstream is decoded by hardware decoder and scaled to 224x224 with decoder post processing unit.

Decoded YUV is kept on device.

The YUV frame is converted to RGBA format with 2D Engine.

The RGBA frame is read out through PCIe and written into an output file.

17.1.2 Command Line

```
ffmpeg -vsync 0 -c:v <dec>_ni_quadra_dec -dec 0 -xcoder-params
out=hw:scale0=224x224:multicoreJointMode=<resolution=8k?1:0> -f concat
-safe 0 -i /media/ramdisk/input.list -vf
ni_quadra_scale=iw:ih:format=rgba,hwdownload,format=rgba -f null -
```

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

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

17.2 Decoder PPU Scaling Performance Results

TYPE	RESOLUTION	JOBS	DEC_LOAD	SCALER_LOAD	FPS	CPU
AVC to RGBA	8k	1	87	0	112	8
HEVC to RGBA	8k	1	88	0	116	8
VP9 to RGBA	8k	1	23	0	31	5
AVC to RGBA	4k	1	21	0	109	9
AVC to RGBA	4k	16	95	0	465	2
HEVC to RGBA	4k	1	22	0	131	11
HEVC to RGBA	4k	16	93	0	549	2
VP9 to RGBA	4k	1	21	0	109	8
VP9 to RGBA	4k	16	94	0	462	1
AVC to RGBA	1080p	40	92	2	1580	1
HEVC to RGBA	1080p	40	94	2	1576	1
VP9 to RGBA	1080p	40	92	2	2001	0
AVC to RGBA	720p	100	65	3	2311	0
HEVC to RGBA	720p	100	77	3	2695	0
VP9 to RGBA	720p	64	51	3	2432	0

18. T1U – Streaming Ladder Generation

18.1 Transcoding

18.1.1 Description

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

Decoded YUV frame is kept on device.

The YUV frames are encoded with hardware encoder.

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

18.1.2 Command line

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

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

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

Input: 1080p

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

18.2 Streaming Ladder Generation Performance Results

TYPE	JOBS	DEC_LOAD	ENC_LOAD	SCALER_LOAD	FPS	CPU
AVC to AVC	8	29	89	3	360	3
AVC to HEVC	8	27	86	2	373	2
AVC to AV1	8	19	82	1	312	2
HEVC to AVC	8	31	89	3	360	3
HEVC to HEVC	8	29	86	2	378	3
HEVC to AV1	8	21	86	1	319	3
VP9 to AVC	8	37	90	3	360	2
VP9 to HEVC	8	35	89	2	384	2
VP9 to AV1	8	26	84	1	316	2

19. T1U – RGBA Encoding

19.1 Encoding

19.1.1 Description

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

RGBA frame is uploaded to the 2D Engine and formatted to YUV with `ni_quadra_scale`.

YUV frame is encoded by hardware encoder.

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

19.1.2 Command line

```
ffmpeg -nostdin -stream_loop -1 -f rawvideo -pix_fmt rgba -s:v  
<resolution> -r 30 -i /media/ramdisk/input.rgb -vf  
"ni_quadra_hwupload=0,ni_quadra_scale=iw:ih:format=yuv420p" -c:v  
<enc>_ni_quadra_enc -enc 0 -xcoder-params  
intraPeriod=0:RcEnable=1:bitrate=<*>:multicoreJointMode=<*> -f null  
/dev/null
```

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num_jobs> == 1, `multicoreJointMode` = 1

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

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

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

19.2 RGBA Encoding Performance Results

TYPE	RES	JOB	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	28	79	49
RGBA to HEVC	4k	1	1	27	80	49
RGBA to AV1	4k	1	1	30	79	48
RGBA to AVC	4k	4	0	55	136	26
RGBA to HEVC	4k	4	0	53	148	28
RGBA to AV1	4k	4	0	59	144	28
RGBA to AVC	4k	8	0	79	144	13
RGBA to HEVC	4k	8	0	70	152	14
RGBA to AV1	4k	8	0	73	152	14
RGBA to AVC	1080p	1	1	23	256	23
RGBA to HEVC	1080p	1	1	22	259	24
RGBA to AV1	1080p	1	1	24	250	23
RGBA to AVC	1080p	16	0	79	544	6
RGBA to HEVC	1080p	16	0	73	576	6
RGBA to AV1	1080p	16	0	74	576	6
RGBA to AVC	1080p	32	0	79	576	3
RGBA to HEVC	1080p	32	0	72	608	4
RGBA to AV1	1080p	32	0	76	608	4
RGBA to AVC	720p	1	1	17	438	27
RGBA to HEVC	720p	1	1	16	435	28
RGBA to AV1	720p	1	1	19	417	25
RGBA to AVC	720p	16	0	44	1008	35
RGBA to HEVC	720p	16	0	44	1007	35
RGBA to AV1	720p	16	0	46	995	31
RGBA to AVC	720p	32	0	50	1037	24
RGBA to HEVC	720p	32	0	48	1043	24
RGBA to AV1	720p	32	0	51	1025	22

20. T1U – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

20.1 Encoding

20.1.1 Description

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

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

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

20.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

20.2 Encoding EnableRdoQuant/rdoLevel/lookaheadDepth Performance Results

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	0	0	1	98	268	16
YUV to HEVC	4k	4	0	0	1	97	288	15
YUV to AV1	4k	4	0	0	1	96	252	15
YUV to AVC	4k	4	0	0	2	98	268	16
YUV to HEVC	4k	4	0	0	2	98	148	8
YUV to AV1	4k	4	0	0	2	97	124	7
YUV to AVC	4k	4	0	0	3	96	268	17
YUV to HEVC	4k	4	0	0	3	98	88	5
YUV to AV1	4k	4	0	0	3	98	68	4
YUV to AVC	4k	4	0	1	1	99	164	8
YUV to HEVC	4k	4	0	1	1	97	208	11
YUV to AVC	4k	4	0	1	2	99	164	8
YUV to HEVC	4k	4	0	1	2	100	96	5
YUV to AVC	4k	4	0	1	3	100	164	9
YUV to HEVC	4k	4	0	1	3	99	60	4
YUV to AVC	4k	4	16	0	1	100	172	9
YUV to HEVC	4k	4	16	0	1	100	198	10
YUV to AV1	4k	4	16	0	1	100	172	8
YUV to AVC	4k	4	16	0	2	99	172	9
YUV to HEVC	4k	4	16	0	2	99	120	6
YUV to AV1	4k	4	16	0	2	100	100	5
YUV to AVC	4k	4	16	0	3	100	172	9
YUV to HEVC	4k	4	16	0	3	99	76	4
YUV to AV1	4k	4	16	0	3	100	64	5
YUV to AVC	4k	4	16	1	1	100	120	6
YUV to HEVC	4k	4	16	1	1	99	156	8
YUV to AVC	4k	4	16	1	2	99	120	6
YUV to HEVC	4k	4	16	1	2	99	83	5
YUV to AVC	4k	4	16	1	3	100	120	6
YUV to HEVC	4k	4	16	1	3	100	56	3
YUV to AVC	4k	4	40	0	1	99	171	9
YUV to HEVC	4k	4	40	0	1	99	196	10
YUV to AV1	4k	4	40	0	1	99	172	9
YUV to AVC	4k	4	40	0	2	99	171	9
YUV to HEVC	4k	4	40	0	2	100	116	7
YUV to AV1	4k	4	40	0	2	100	100	5
YUV to AVC	4k	4	40	0	3	100	171	9
YUV to HEVC	4k	4	40	0	3	100	76	5

T1U – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	4k	4	40	0	3	99	64	4
YUV to AVC	4k	4	40	1	1	99	120	6
YUV to HEVC	4k	4	40	1	1	99	154	9
YUV to AVC	4k	4	40	1	2	99	120	6
YUV to HEVC	4k	4	40	1	2	100	80	5
YUV to AVC	4k	4	40	1	3	99	120	7
YUV to HEVC	4k	4	40	1	3	100	56	4
YUV to AVC	1080p	20	0	0	1	99	1105	3
YUV to HEVC	1080p	20	0	0	1	99	1180	4
YUV to AV1	1080p	20	0	0	1	99	1020	3
YUV to AVC	1080p	20	0	0	2	99	1119	3
YUV to HEVC	1080p	20	0	0	2	99	600	2
YUV to AV1	1080p	20	0	0	2	99	481	2
YUV to AVC	1080p	20	0	0	3	99	1103	3
YUV to HEVC	1080p	20	0	0	3	99	360	1
YUV to AV1	1080p	20	0	0	3	100	260	1
YUV to AVC	1080p	20	0	1	1	99	660	2
YUV to HEVC	1080p	20	0	1	1	99	840	2
YUV to AVC	1080p	20	0	1	2	99	660	2
YUV to HEVC	1080p	20	0	1	2	100	380	1
YUV to AVC	1080p	20	0	1	3	99	660	2
YUV to HEVC	1080p	20	0	1	3	100	240	1
YUV to AVC	1080p	20	16	0	1	99	611	2
YUV to HEVC	1080p	20	16	0	1	99	716	2
YUV to AV1	1080p	20	16	0	1	99	620	2
YUV to AVC	1080p	20	16	0	2	99	610	2
YUV to HEVC	1080p	20	16	0	2	99	440	2
YUV to AV1	1080p	20	16	0	2	100	368	1
YUV to AVC	1080p	20	16	0	3	99	608	2
YUV to HEVC	1080p	20	16	0	3	99	300	1
YUV to AV1	1080p	20	16	0	3	100	240	1
YUV to AVC	1080p	20	16	1	1	100	440	2
YUV to HEVC	1080p	20	16	1	1	99	561	2
YUV to AVC	1080p	20	16	1	2	99	440	2
YUV to HEVC	1080p	20	16	1	2	99	320	1
YUV to AVC	1080p	20	16	1	3	99	440	2
YUV to HEVC	1080p	20	16	1	3	100	220	1
YUV to AVC	1080p	20	40	0	1	99	600	2
YUV to HEVC	1080p	20	40	0	1	99	700	2

T1U – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	40	0	1	99	620	2
YUV to AVC	1080p	20	40	0	2	99	600	2
YUV to HEVC	1080p	20	40	0	2	99	440	2
YUV to AV1	1080p	20	40	0	2	99	360	2
YUV to AVC	1080p	20	40	0	3	99	600	2
YUV to HEVC	1080p	20	40	0	3	100	294	1
YUV to AV1	1080p	20	40	0	3	99	231	1
YUV to AVC	1080p	20	40	1	1	99	440	2
YUV to HEVC	1080p	20	40	1	1	100	560	2
YUV to AVC	1080p	20	40	1	2	99	440	2
YUV to HEVC	1080p	20	40	1	2	99	305	1
YUV to AVC	1080p	20	40	1	3	99	440	2
YUV to HEVC	1080p	20	40	1	3	100	218	1
YUV to AVC	720p	40	0	0	1	90	2265	11
YUV to HEVC	720p	40	0	0	1	92	2296	12
YUV to AV1	720p	40	0	0	1	88	1878	9
YUV to AVC	720p	40	0	0	2	90	2246	11
YUV to HEVC	720p	40	0	0	2	99	1320	3
YUV to AV1	720p	40	0	0	2	99	1052	2
YUV to AVC	720p	40	0	0	3	89	2269	12
YUV to HEVC	720p	40	0	0	3	100	800	1
YUV to AV1	720p	40	0	0	3	99	560	1
YUV to AVC	720p	40	0	1	1	99	1480	3
YUV to HEVC	720p	40	0	1	1	99	1800	5
YUV to AVC	720p	40	0	1	2	99	1480	3
YUV to HEVC	720p	40	0	1	2	99	840	2
YUV to AVC	720p	40	0	1	3	99	1480	3
YUV to HEVC	720p	40	0	1	3	99	560	1
YUV to AVC	720p	40	16	0	1	97	1205	4
YUV to HEVC	720p	40	16	0	1	83	1200	4
YUV to AV1	720p	40	16	0	1	69	922	3
YUV to AVC	720p	40	16	0	2	98	1224	5
YUV to HEVC	720p	40	16	0	2	99	920	2
YUV to AV1	720p	40	16	0	2	99	760	2
YUV to AVC	720p	40	16	0	3	97	1219	4
YUV to HEVC	720p	40	16	0	3	99	640	1
YUV to AV1	720p	40	16	0	3	99	480	1
YUV to AVC	720p	40	16	1	1	99	920	2
YUV to HEVC	720p	40	16	1	1	99	1160	4

T1U – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	40	16	1	2	99	920	2
YUV to HEVC	720p	40	16	1	2	99	660	1
YUV to AVC	720p	40	16	1	3	99	920	2
YUV to HEVC	720p	40	16	1	3	99	453	1
YUV to AVC	720p	40	40	0	1	92	1160	4
YUV to HEVC	720p	40	40	0	1	81	1160	4
YUV to AV1	720p	40	40	0	1	70	920	3
YUV to AVC	720p	40	40	0	2	92	1142	4
YUV to HEVC	720p	40	40	0	2	99	920	2
YUV to AV1	720p	40	40	0	2	99	760	2
YUV to AVC	720p	40	40	0	3	92	1160	4
YUV to HEVC	720p	40	40	0	3	99	640	1
YUV to AV1	720p	40	40	0	3	99	480	1
YUV to AVC	720p	40	40	1	1	99	919	2
YUV to HEVC	720p	40	40	1	1	99	1121	5
YUV to AVC	720p	40	40	1	2	99	920	2
YUV to HEVC	720p	40	40	1	2	99	640	1
YUV to AVC	720p	40	40	1	3	99	920	2
YUV to HEVC	720p	40	40	1	3	99	440	1

21. T1U – Capped CRF

21.1 Encoding with lookaheadDepth

21.1.1 Description

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

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

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

21.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

21.2 Capped CRF Encoding with lookaheadDepth Performance Results

TYPE	RES	JOB	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	16	0	1	19	99	611	2
YUV to HEVC	1080p	20	16	0	1	19	99	718	2
YUV to AV1	1080p	20	16	0	1	19	99	621	2
YUV to AVC	1080p	20	16	0	2	19	99	611	2
YUV to HEVC	1080p	20	16	0	2	19	100	440	2
YUV to AV1	1080p	20	16	0	2	19	99	365	1
YUV to AVC	1080p	20	16	0	3	19	99	611	2
YUV to HEVC	1080p	20	16	0	3	19	99	300	1
YUV to AV1	1080p	20	16	0	3	19	99	239	1
YUV to AVC	1080p	20	16	1	1	19	99	440	2
YUV to HEVC	1080p	20	16	1	1	19	99	561	2
YUV to AVC	1080p	20	16	1	2	19	99	440	2
YUV to HEVC	1080p	20	16	1	2	19	100	320	1
YUV to AVC	1080p	20	16	1	3	19	99	440	2
YUV to HEVC	1080p	20	16	1	3	19	99	220	1
YUV to AVC	1080p	20	40	0	1	19	99	601	2
YUV to HEVC	1080p	20	40	0	1	19	99	702	2
YUV to AV1	1080p	20	40	0	1	19	99	620	2
YUV to AVC	1080p	20	40	0	2	19	99	600	2
YUV to HEVC	1080p	20	40	0	2	19	99	440	2
YUV to AV1	1080p	20	40	0	2	19	99	378	2
YUV to AVC	1080p	20	40	0	3	19	99	600	2
YUV to HEVC	1080p	20	40	0	3	19	100	296	1
YUV to AV1	1080p	20	40	0	3	19	100	235	1
YUV to AVC	1080p	20	40	1	1	19	99	440	2
YUV to HEVC	1080p	20	40	1	1	19	99	567	2
YUV to AVC	1080p	20	40	1	2	19	99	440	2
YUV to HEVC	1080p	20	40	1	2	19	100	320	1
YUV to AVC	1080p	20	40	1	3	19	99	440	2
YUV to HEVC	1080p	20	40	1	3	19	100	216	1
YUV to AVC	1080p	20	16	0	1	23	99	608	2
YUV to HEVC	1080p	20	16	0	1	23	99	718	2
YUV to AV1	1080p	20	16	0	1	23	99	620	2
YUV to AVC	1080p	20	16	0	2	23	99	611	2
YUV to HEVC	1080p	20	16	0	2	23	99	440	2
YUV to AV1	1080p	20	16	0	2	23	100	365	2
YUV to AVC	1080p	20	16	0	3	23	99	605	2
YUV to HEVC	1080p	20	16	0	3	23	99	300	1

T1U – Capped CRF

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	3	23	100	239	1
YUV to AVC	1080p	20	16	1	1	23	99	440	2
YUV to HEVC	1080p	20	16	1	1	23	99	560	2
YUV to AVC	1080p	20	16	1	2	23	99	440	2
YUV to HEVC	1080p	20	16	1	2	23	99	320	1
YUV to AVC	1080p	20	16	1	3	23	99	440	2
YUV to HEVC	1080p	20	16	1	3	23	100	220	1
YUV to AVC	1080p	20	40	0	1	23	99	600	2
YUV to HEVC	1080p	20	40	0	1	23	99	700	2
YUV to AV1	1080p	20	40	0	1	23	99	620	2
YUV to AVC	1080p	20	40	0	2	23	99	600	2
YUV to HEVC	1080p	20	40	0	2	23	99	440	2
YUV to AV1	1080p	20	40	0	2	23	99	360	2
YUV to AVC	1080p	20	40	0	3	23	99	600	2
YUV to HEVC	1080p	20	40	0	3	23	99	293	1
YUV to AV1	1080p	20	40	0	3	23	100	232	1
YUV to AVC	1080p	20	40	1	1	23	99	440	2
YUV to HEVC	1080p	20	40	1	1	23	99	560	2
YUV to AVC	1080p	20	40	1	2	23	100	440	2
YUV to HEVC	1080p	20	40	1	2	23	99	304	1
YUV to AVC	1080p	20	40	1	3	23	99	440	2
YUV to HEVC	1080p	20	40	1	3	23	100	217	1
YUV to AVC	1080p	20	16	0	1	27	99	603	2
YUV to HEVC	1080p	20	16	0	1	27	99	716	2
YUV to AV1	1080p	20	16	0	1	27	99	620	2
YUV to AVC	1080p	20	16	0	2	27	99	610	2
YUV to HEVC	1080p	20	16	0	2	27	99	440	2
YUV to AV1	1080p	20	16	0	2	27	100	365	1
YUV to AVC	1080p	20	16	0	3	27	99	608	2
YUV to HEVC	1080p	20	16	0	3	27	99	300	1
YUV to AV1	1080p	20	16	0	3	27	100	239	1
YUV to AVC	1080p	20	16	1	1	27	99	440	2
YUV to HEVC	1080p	20	16	1	1	27	100	560	2
YUV to AVC	1080p	20	16	1	2	27	100	440	2
YUV to HEVC	1080p	20	16	1	2	27	99	320	1
YUV to AVC	1080p	20	16	1	3	27	100	440	2
YUV to HEVC	1080p	20	16	1	3	27	100	220	1
YUV to AVC	1080p	20	40	0	1	27	99	600	2
YUV to HEVC	1080p	20	40	0	1	27	99	705	3

T1U – Capped CRF

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	40	0	1	27	99	620	2
YUV to AVC	1080p	20	40	0	2	27	99	600	2
YUV to HEVC	1080p	20	40	0	2	27	99	440	2
YUV to AV1	1080p	20	40	0	2	27	99	374	2
YUV to AVC	1080p	20	40	0	3	27	99	600	2
YUV to HEVC	1080p	20	40	0	3	27	99	298	1
YUV to AV1	1080p	20	40	0	3	27	100	233	1
YUV to AVC	1080p	20	40	1	1	27	99	440	2
YUV to HEVC	1080p	20	40	1	1	27	99	579	2
YUV to AVC	1080p	20	40	1	2	27	99	440	2
YUV to HEVC	1080p	20	40	1	2	27	99	320	1
YUV to AVC	1080p	20	40	1	3	27	99	440	2
YUV to HEVC	1080p	20	40	1	3	27	99	218	1

22. T1U – Inplace Overlay

22.1 Transcoding

22.1.1 Description

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

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

22.1.2 Command line

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

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

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

Input Video: 1080p

Input Image: 128x128

22.2 Inplace Overlay Performance Results

TYPE	JOB5	FPS	CPU	DEC_LOAD	ENC_LOAD	SCALER_LOAD
AVC to AVC	1	244	11	13	19	0
AVC to HEVC	1	259	11	14	19	0
AVC to AV1	1	223	10	12	20	0
HEVC to AVC	1	243	13	13	20	0
HEVC to HEVC	1	257	14	14	19	0
HEVC to AV1	1	221	13	12	19	0
VP9 to AVC	1	243	11	18	20	0
VP9 to HEVC	1	260	11	19	19	0
VP9 to AV1	1	223	10	17	20	0
AVC to AVC	16	912	2	69	89	4
AVC to HEVC	16	1008	2	75	89	5
AVC to AV1	16	960	2	69	91	4
HEVC to AVC	16	944	3	71	91	5
HEVC to HEVC	16	1040	3	76	91	5
HEVC to AV1	16	975	3	68	92	4
VP9 to AVC	16	922	2	84	90	5
VP9 to HEVC	16	999	2	87	85	6
VP9 to AV1	16	961	2	82	89	4
AVC to AVC	32	863	1	72	92	4
AVC to HEVC	32	960	1	81	93	6
AVC to AV1	32	928	1	76	93	5
HEVC to AVC	32	896	1	75	95	5
HEVC to HEVC	32	1017	1	81	94	6
HEVC to AV1	32	960	1	73	94	5
VP9 to AVC	32	895	1	85	92	5
VP9 to HEVC	32	997	1	91	90	7
VP9 to AV1	32	961	1	87	93	5

Appendix A: GStreamer XStack Command

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

[illegible]

[illegible]

Appendix B: 7x7 Grid Layout

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

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

Example of single input with 64 outputs

Page 73 of 74

```

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

```