Quadra^{TM} Performance Test Report V5.2

Contents

Environment Overview

Revision: 5206s5r3

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: 6.1
- Gstreamer Version: 1.22.2
- Tests:
  - FFmpeg Throughput
  - Libxcoder 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 0PYVT1; 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.
  - (System-wide CPU usage * number of CPU) / (number of devices * number of instances per device)
- FPS: Average of all FPS reported per process
- Jobs: Number of instances running concurrently
- HW Frame: Decoded YUV is kept on the device
- Bit: Input video's bit depth
- Resolution: Input video's resolution
- Load: Maximum load between FW Load and VPU Load during traffic

1. T1A – FFmpeg Throughput

1.1 Decoding

1.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

1.1.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

1.2 Encoding

1.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

1.2.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

1.3 Transcoding

1.3.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder.
Decoded YUV frame is kept on device.
The YUV frame is encoded with hardware encoder.
The encoded bitstream is read out through PCIe and written into an output file.

1.3.2 Command line

```
ffmpeg -nostdin -f concat -safe 0 -c:v <dec>_ni_quadra_dec -dec 0 -
xcoder-params out=hw:semiplanar0=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	96	0	114	13
HEVC to YUV	8k	1	0	8	1	98	0	114	12
VP9 to YUV	8k	1	0	8	1	24	0	38	3
YUV to AVC	8k	1	0	8	1	0	96	67	62
YUV to HEVC	8k	1	0	8	1	0	98	83	81
AVC to AVC	8k	1	1	8	1	70	99	54	7
AVC to HEVC	8k	1	1	8	1	83	98	71	3
HEVC to AVC	8k	1	1	8	1	63	95	52	4
HEVC to HEVC	8k	1	1	8	1	65	100	70	8
VP9 to AVC	8k	1	1	8	1	24	43	35	3
VP9 to HEVC	8k	1	1	8	1	25	44	37	4
AVC to YUV	8k	1	0	10	1	56	0	59	10
HEVC to YUV	8k	1	0	10	1	58	0	61	8
VP9 to YUV	8k	1	0	10	1	22	0	33	15
YUV to AVC	8k	1	0	10	1	0	92	48	88
YUV to HEVC	8k	1	0	10	1	0	73	59	106
AVC to YUV	4k	1	0	8	1	53	0	303	21
HEVC to YUV	4k	1	0	8	1	48	0	319	25
VP9 to YUV	4k	1	0	8	1	24	0	155	8
AVC to YUV	4k	16	0	8	0	99	0	485	3
HEVC to YUV	4k	16	0	8	0	99	0	506	4
VP9 to YUV	4k	16	0	8	0	98	0	487	1
YUV to AVC	4k	1	0	8	1	0	94	293	44
YUV to HEVC	4k	1	0	8	1	0	94	321	50
YUV to AV1	4k	1	0	8	1	0	93	281	45
YUV to AVC	4k	4	0	8	0	0	96	304	23
YUV to HEVC	4k	4	0	8	0	0	96	332	16
YUV to AV1	4k	4	0	8	0	0	96	288	15
YUV to AVC	4k	8	0	8	0	0	99	321	10
YUV to HEVC	4k	8	0	8	0	0	100	344	12
YUV to AV1	4k	8	0	8	0	0	100	296	8
AVC to AVC	4k	1	1	8	1	64	87	217	13
AVC to HEVC	4k	1	1	8	1	70	88	271	16
AVC to AV1	4k	1	1	8	1	59	89	256	14
HEVC to AVC	4k	1	1	8	1	52	86	213	15
HEVC to HEVC	4k	1	1	8	1	56	88	266	16
HEVC to AV1	4k	1	1	8	1	49	87	250	18
VP9 to AVC	4k	1	1	8	1	24	46	149	6
VP9 to HEVC	4k	1	1	8	1	24	43	151	8

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	6
AVC to AVC	4k	4	1	8	0	63	97	240	8
AVC to HEVC	4k	4	1	8	0	66	92	294	9
AVC to AV1	4k	4	1	8	0	54	95	272	4
HEVC to AVC	4k	4	1	8	0	57	96	236	4
HEVC to HEVC	4k	4	1	8	0	60	93	293	9
HEVC to AV1	4k	4	1	8	0	46	95	270	4
VP9 to AVC	4k	4	1	8	0	64	97	236	4
VP9 to HEVC	4k	4	1	8	0	65	96	299	4
VP9 to AV1	4k	4	1	8	0	62	95	272	4
AVC to AVC	4k	8	1	8	0	68	100	219	3
AVC to HEVC	4k	8	1	8	0	76	99	280	3
AVC to AV1	4k	8	1	8	0	68	100	272	3
HEVC to AVC	4k	8	1	8	0	63	99	219	3
HEVC to HEVC	4k	8	1	8	0	67	99	286	3
HEVC to AV1	4k	8	1	8	0	61	99	272	3
VP9 to AVC	4k	8	1	8	0	68	99	235	3
VP9 to HEVC	4k	8	1	8	0	72	100	304	3
VP9 to AV1	4k	8	1	8	0	59	100	281	3
AVC to YUV	4k	1	0	10	0	43	0	219	12
HEVC to YUV	4k	1	0	10	0	51	0	227	10
VP9 to YUV	4k	1	0	10	0	24	0	158	7
AVC to YUV	4k	16	0	10	0	100	0	283	1
HEVC to YUV	4k	16	0	10	0	99	0	280	1
VP9 to YUV	4k	16	0	10	0	99	0	500	1
YUV to AVC	4k	1	0	10	0	0	69	196	62
YUV to HEVC	4k	1	0	10	0	0	57	201	66
YUV to AV1	4k	1	0	10	0	0	64	194	60
YUV to AVC	4k	4	0	10	0	0	94	220	33
YUV to HEVC	4k	4	0	10	0	0	73	249	44
YUV to AV1	4k	4	0	10	0	0	81	240	45
AVC to YUV	1080p	1	0	8	1	41	0	837	27
HEVC to YUV	1080p	1	0	8	1	43	0	811	30
VP9 to YUV	1080p	1	0	8	1	22	0	557	10
AVC to YUV	1080p	40	0	8	0	87	0	1722	1
HEVC to YUV	1080p	40	0	8	0	98	0	1834	1
VP9 to YUV	1080p	40	0	8	0	78	0	1791	0
YUV to AVC	1080p	1	0	8	1	0	53	687	30
YUV to HEVC	1080p	1	0	8	1	0	50	689	32
YUV to AV1	1080p	1	0	8	1	0	53	625	25

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to AVC	1080p	32	0	8	0	0	99	1280	3
YUV to HEVC	1080p	32	0	8	0	0	99	1369	3
YUV to AV1	1080p	32	0	8	0	0	99	1184	3
AVC to AVC	1080p	1	1	8	1	37	55	717	25
AVC to HEVC	1080p	1	1	8	1	37	52	714	21
AVC to AV1	1080p	1	1	8	1	33	57	648	20
HEVC to AVC	1080p	1	1	8	1	36	53	705	28
HEVC to HEVC	1080p	1	1	8	1	35	50	710	32
HEVC to AV1	1080p	1	1	8	1	32	55	638	28
VP9 to AVC	1080p	1	1	8	1	21	42	549	12
VP9 to HEVC	1080p	1	1	8	1	21	39	542	10
VP9 to AV1	1080p	1	1	8	1	22	46	542	13
AVC to AVC	1080p	32	1	8	0	76	99	960	3
AVC to HEVC	1080p	32	1	8	0	82	99	1080	1
AVC to AV1	1080p	32	1	8	0	76	99	1031	1
HEVC to AVC	1080p	32	1	8	0	70	100	992	4
HEVC to HEVC	1080p	32	1	8	0	78	99	1120	1
HEVC to AV1	1080p	32	1	8	0	70	99	1056	1
VP9 to AVC	1080p	32	1	8	0	65	100	1088	1
VP9 to HEVC	1080p	32	1	8	0	69	100	1216	3
VP9 to AV1	1080p	32	1	8	0	59	99	1120	1
AVC to YUV	1080p	1	0	10	0	28	0	618	12
HEVC to YUV	1080p	1	0	10	0	27	0	641	14
VP9 to YUV	1080p	1	0	10	0	22	0	456	10
AVC to YUV	1080p	40	0	10	0	68	0	1084	0
HEVC to YUV	1080p	40	0	10	0	68	0	1086	0
VP9 to YUV	1080p	40	0	10	0	70	0	1070	0
YUV to AVC	1080p	1	0	10	0	0	37	488	38
YUV to HEVC	1080p	1	0	10	0	0	34	477	34
YUV to AV1	1080p	1	0	10	0	0	38	455	36
YUV to AVC	1080p	32	0	10	0	0	64	832	7
YUV to HEVC	1080p	32	0	10	0	0	60	832	7
YUV to AV1	1080p	32	0	10	0	0	67	798	7
AVC to YUV	720p	1	0	8	1	41	0	1103	22
HEVC to YUV	720p	1	0	8	1	36	0	1099	26
VP9 to YUV	720p	1	0	8	1	33	0	1021	13
AVC to YUV	720p	100	0	8	0	100	0	2515	0
HEVC to YUV	720p	100	0	8	0	100	0	2925	1
VP9 to YUV	720p	100	0	8	0	99	0	2629	0
YUV to AVC	720p	1	0	8	1	0	30	879	26

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	30	879	26
YUV to AV1	720p	1	0	8	1	0	34	789	21
YUV to AVC	720p	64	0	8	0	0	91	2304	2
YUV to HEVC	720p	64	0	8	0	0	90	2318	2
YUV to AV1	720p	64	0	8	0	0	97	2046	1
AVC to AVC	720p	1	1	8	1	28	28	781	20
AVC to HEVC	720p	1	1	8	1	28	27	787	19
AVC to AV1	720p	1	1	8	1	25	31	707	19
HEVC to AVC	720p	1	1	8	1	25	28	793	24
HEVC to HEVC	720p	1	1	8	1	24	28	793	22
HEVC to AV1	720p	1	1	8	1	22	31	711	24
VP9 to AVC	720p	1	1	8	1	26	29	815	15
VP9 to HEVC	720p	1	1	8	1	27	28	818	14
VP9 to AV1	720p	1	1	8	1	24	32	725	16
AVC to AVC	720p	64	1	8	0	95	100	2048	0
AVC to HEVC	720p	64	1	8	0	96	100	2114	0
AVC to AV1	720p	64	1	8	0	76	100	1792	0
HEVC to AVC	720p	64	1	8	0	84	99	2036	0
HEVC to HEVC	720p	64	1	8	0	84	99	2121	0
HEVC to AV1	720p	64	1	8	0	68	100	1798	0
VP9 to AVC	720p	64	1	8	0	95	100	2242	0
VP9 to HEVC	720p	64	1	8	0	98	98	2307	0
VP9 to AV1	720p	64	1	8	0	78	100	1921	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 Libxcoder 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	65	0	93	9
HEVC to YUV	8k	1	0	8	1	64	0	96	8
VP9 to YUV	8k	1	0	8	1	24	0	38	4
YUV to AVC	8k	1	0	8	1	0	97	67	35
YUV to HEVC	8k	1	0	8	1	0	90	76	40
AVC to AVC	8k	1	1	8	1	67	99	54	13
AVC to HEVC	8k	1	1	8	1	84	99	71	13
HEVC to AVC	8k	1	1	8	1	63	100	52	12
HEVC to HEVC	8k	1	1	8	1	74	99	70	13
VP9 to AVC	8k	1	1	8	1	25	48	35	8
VP9 to HEVC	8k	1	1	8	1	24	44	37	8
AVC to YUV	8k	1	0	10	1	55	0	54	9
HEVC to YUV	8k	1	0	10	1	51	0	51	8
VP9 to YUV	8k	1	0	10	1	24	0	33	6
YUV to AVC	8k	1	0	10	1	0	56	37	40
YUV to HEVC	8k	1	0	10	1	0	43	37	40
AVC to YUV	4k	1	0	8	1	56	0	312	11
HEVC to YUV	4k	1	0	8	1	48	0	322	9
VP9 to YUV	4k	1	0	8	1	24	0	154	6
AVC to YUV	4k	16	0	8	0	99	0	487	1
HEVC to YUV	4k	16	0	8	0	100	0	511	1
VP9 to YUV	4k	16	0	8	0	99	0	492	1
YUV to AVC	4k	1	0	8	1	0	83	263	33
YUV to HEVC	4k	1	0	8	1	0	79	272	33
YUV to AV1	4k	1	0	8	1	0	83	249	33
YUV to AVC	4k	4	0	8	0	0	96	310	14
YUV to HEVC	4k	4	0	8	0	0	96	334	14
YUV to AV1	4k	4	0	8	0	0	96	289	14
YUV to AVC	4k	8	0	8	0	0	100	326	9
YUV to HEVC	4k	8	0	8	0	0	99	347	9
YUV to AV1	4k	8	0	8	0	0	100	301	9
AVC to AVC	4k	1	1	8	0	67	96	225	16
AVC to HEVC	4k	1	1	8	0	73	95	279	16
AVC to AV1	4k	1	1	8	0	64	96	262	17
HEVC to AVC	4k	1	1	8	0	57	97	225	14
HEVC to HEVC	4k	1	1	8	0	63	96	287	15
HEVC to AV1	4k	1	1	8	0	56	96	267	15
VP9 to AVC	4k	1	1	8	0	24	46	144	10

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	0	24	43	151	9
VP9 to AV1	4k	1	1	8	0	24	51	151	10
AVC to AVC	4k	4	1	8	0	61	96	243	10
AVC to HEVC	4k	4	1	8	0	67	96	302	10
AVC to AV1	4k	4	1	8	0	55	95	276	10
HEVC to AVC	4k	4	1	8	0	57	96	239	9
HEVC to HEVC	4k	4	1	8	0	59	96	294	9
HEVC to AV1	4k	4	1	8	0	50	95	273	9
VP9 to AVC	4k	4	1	8	0	81	97	246	9
VP9 to HEVC	4k	4	1	8	0	80	96	304	9
VP9 to AV1	4k	4	1	8	0	82	95	277	8
AVC to AVC	4k	8	1	8	0	68	99	217	6
AVC to HEVC	4k	8	1	8	0	74	100	280	6
AVC to AV1	4k	8	1	8	0	67	100	273	6
HEVC to AVC	4k	8	1	8	0	65	99	218	5
HEVC to HEVC	4k	8	1	8	0	67	100	282	6
HEVC to AV1	4k	8	1	8	0	61	100	275	6
VP9 to AVC	4k	8	1	8	0	100	99	234	5
VP9 to HEVC	4k	8	1	8	0	99	99	302	5
VP9 to AV1	4k	8	1	8	0	100	100	286	5
AVC to YUV	4k	1	0	10	1	48	0	209	9
HEVC to YUV	4k	1	0	10	1	42	0	203	9
VP9 to YUV	4k	1	0	10	1	24	0	158	5
AVC to YUV	4k	16	0	10	0	98	0	285	1
HEVC to YUV	4k	16	0	10	0	99	0	284	0
VP9 to YUV	4k	16	0	10	0	99	0	505	1
YUV to AVC	4k	1	0	10	1	0	45	146	34
YUV to HEVC	4k	1	0	10	1	0	43	150	35
YUV to AV1	4k	1	0	10	1	0	46	140	33
YUV to AVC	4k	4	0	10	0	0	64	197	30
YUV to HEVC	4k	4	0	10	0	0	56	196	31
YUV to AV1	4k	4	0	10	0	0	56	175	31
AVC to YUV	1080p	1	0	8	1	40	0	835	18
HEVC to YUV	1080p	1	0	8	1	46	0	890	15
VP9 to YUV	1080p	1	0	8	1	22	0	557	9
AVC to YUV	1080p	40	0	8	0	83	0	1645	1
HEVC to YUV	1080p	40	0	8	0	88	0	1762	0
VP9 to YUV	1080p	40	0	8	0	71	0	1675	0
YUV to AVC	1080p	1	0	8	1	0	54	703	23

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	50	692	25
YUV to AV1	1080p	1	0	8	1	0	47	556	21
YUV to AVC	1080p	32	0	8	0	0	100	1326	3
YUV to HEVC	1080p	32	0	8	0	0	99	1403	3
YUV to AV1	1080p	32	0	8	0	0	100	1211	3
AVC to AVC	1080p	1	1	8	0	69	84	1002	29
AVC to HEVC	1080p	1	1	8	0	68	82	1058	29
AVC to AV1	1080p	1	1	8	0	55	81	926	29
HEVC to AVC	1080p	1	1	8	0	62	86	1005	23
HEVC to HEVC	1080p	1	1	8	0	64	85	1091	24
HEVC to AV1	1080p	1	1	8	0	51	82	932	24
VP9 to AVC	1080p	1	1	8	0	21	42	546	14
VP9 to HEVC	1080p	1	1	8	0	22	39	541	14
VP9 to AV1	1080p	1	1	8	0	22	46	539	15
AVC to AVC	1080p	32	1	8	0	75	100	976	1
AVC to HEVC	1080p	32	1	8	0	84	99	1104	1
AVC to AV1	1080p	32	1	8	0	77	99	1064	1
HEVC to AVC	1080p	32	1	8	0	70	99	1020	1
HEVC to HEVC	1080p	32	1	8	0	77	99	1156	1
HEVC to AV1	1080p	32	1	8	0	68	100	1091	1
VP9 to AVC	1080p	32	1	8	0	100	99	1107	1
VP9 to HEVC	1080p	32	1	8	0	100	99	1254	1
VP9 to AV1	1080p	32	1	8	0	100	99	1156	1
AVC to YUV	1080p	1	0	10	1	28	0	630	12
HEVC to YUV	1080p	1	0	10	1	25	0	615	10
VP9 to YUV	1080p	1	0	10	1	22	0	455	9
AVC to YUV	1080p	40	0	10	0	63	0	1077	0
HEVC to YUV	1080p	40	0	10	0	72	0	1073	0
VP9 to YUV	1080p	40	0	10	0	71	0	1036	0
YUV to AVC	1080p	1	0	10	1	0	33	438	30
YUV to HEVC	1080p	1	0	10	1	0	32	440	28
YUV to AV1	1080p	1	0	10	1	0	33	391	27
YUV to AVC	1080p	32	0	10	0	0	64	853	7
YUV to HEVC	1080p	32	0	10	0	0	59	846	7
YUV to AV1	1080p	32	0	10	0	0	65	796	8
AVC to YUV	720p	1	0	8	1	41	0	1135	16
HEVC to YUV	720p	1	0	8	1	38	0	1196	14
VP9 to YUV	720p	1	0	8	1	33	0	1022	11
AVC to YUV	720p	100	0	8	0	100	0	2761	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	2954	0
VP9 to YUV	720p	100	0	8	0	94	0	2655	0
YUV to AVC	720p	1	0	8	1	0	32	954	17
YUV to HEVC	720p	1	0	8	1	0	33	972	17
YUV to AV1	720p	1	0	8	1	0	31	765	15
YUV to AVC	720p	64	0	8	0	0	100	2676	2
YUV to HEVC	720p	64	0	8	0	0	99	2758	2
YUV to AV1	720p	64	0	8	0	0	100	2214	2
AVC to AVC	720p	1	1	8	0	49	46	1299	27
AVC to HEVC	720p	1	1	8	0	49	46	1302	28
AVC to AV1	720p	1	1	8	0	42	48	1067	27
HEVC to AVC	720p	1	1	8	0	45	48	1357	23
HEVC to HEVC	720p	1	1	8	0	44	48	1358	23
HEVC to AV1	720p	1	1	8	0	37	47	1068	23
VP9 to AVC	720p	1	1	8	0	32	35	1007	17
VP9 to HEVC	720p	1	1	8	0	32	34	994	17
VP9 to AV1	720p	1	1	8	0	32	44	994	19
AVC to AVC	720p	64	1	8	0	94	99	2176	1
AVC to HEVC	720p	64	1	8	0	100	100	2290	1
AVC to AV1	720p	64	1	8	0	94	100	1899	0
HEVC to AVC	720p	64	1	8	0	87	99	2191	0
HEVC to HEVC	720p	64	1	8	0	95	100	2291	0
HEVC to AV1	720p	64	1	8	0	89	100	1908	0
VP9 to AVC	720p	64	1	8	0	100	100	2245	1
VP9 to HEVC	720p	64	1	8	0	100	96	2281	1
VP9 to AV1	720p	64	1	8	0	100	100	2024	0

3. T1A – FFmpeg Latency

3.1 Encoding

3.1.1 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.

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 -xcoder-params gopPresetIdx=9:lowDelay=1 -f
null -
```

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

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	57.9	59.56	57.44	0.07
YUV to HEVC	8k	1	55.54	60.66	54.01	0.99
YUV to AVC	4k	1	15.46	17.3	15.09	0.05
YUV to HEVC	4k	1	16.67	18.87	15.8	0.1
YUV to AV1	4k	1	21.83	25.62	16.57	0.63
YUV to AVC	4k	4	16.84	21.53	15.04	2.35
YUV to HEVC	4k	4	18.64	22.25	15.72	2.64
YUV to AV1	4k	4	22.94	27.3	16.19	0.77
YUV to AVC	4k	8	20.81	33.31	15.1	23.85
YUV to HEVC	4k	8	23	34.27	15.67	26.21
YUV to AV1	4k	8	37.24	45.74	21.75	8.36
YUV to AVC	1080p	1	4.73	5.68	4.44	0.03
YUV to HEVC	1080p	1	5.15	5.63	4.73	0.02
YUV to AV1	1080p	1	6.8	7.44	5.08	0.06
YUV to AVC	1080p	32	6.19	11.6	4.53	1.12
YUV to HEVC	1080p	32	7.11	13.78	4.88	1.77
YUV to AV1	1080p	32	40.99	46.41	34.39	2
YUV to AVC	720p	1	2.9	3.5	2.52	0.02
YUV to HEVC	720p	1	2.92	3.39	2.69	0.02
YUV to AV1	720p	1	3.9	4.34	3.15	0.02
YUV to AVC	720p	64	5.34	9.77	3.64	0.44
YUV to HEVC	720p	64	5.78	10.47	3.9	0.56
YUV to AV1	720p	64	39.72	48.2	32.64	3.39

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	88	0	141	9
HEVC to RGBA	8k	1	89	0	152	10
VP9 to RGBA	8k	1	22	0	40	3
AVC to RGBA	4k	1	21	0	141	11
AVC to RGBA	4k	16	93	3	584	3
HEVC to RGBA	4k	1	21	0	170	13
HEVC to RGBA	4k	16	94	4	664	4
VP9 to RGBA	4k	1	22	0	166	6
VP9 to RGBA	4k	16	95	4	680	2
AVC to RGBA	1080p	40	94	14	1893	1
HEVC to RGBA	1080p	40	94	14	2004	2
VP9 to RGBA	1080p	40	90	17	2373	1
AVC to RGBA	720p	100	95	18	2581	0
HEVC to RGBA	720p	100	87	19	2715	0
VP9 to RGBA	720p	64	95	18	2750	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:semiplanar0=1:enableOut1=1:semiplanar1=1:scale1=1280x720:enableO
ut2=1:semiplanar2=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	94	3	432	4
AVC to HEVC	8	30	91	2	456	4
AVC to AV1	8	20	88	1	375	3
HEVC to AVC	8	34	94	3	433	5
HEVC to HEVC	8	32	92	3	464	5
HEVC to AV1	8	21	88	1	376	4
VP9 to AVC	8	39	94	3	432	4
VP9 to HEVC	8	39	93	3	460	4
VP9 to AV1	8	26	88	1	376	4

6. T1A – RGBA Encoding

6.1 Encoding

6.1.1 Description

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

RGBA frame is uploaded and encoded by hardware encoder.

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

6.1.2 Command line

```
ffmpeg –nostdin –stream_loop –1 –f rawvideo –pix_fmt rgba –s:v
<resolution> -r 30 -i /media/ramdisk/input.rgba -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	47	150	72
RGBA to HEVC	4k	1	1	43	150	74
RGBA to AV1	4k	1	1	49	146	69
RGBA to AVC	4k	4	0	61	166	52
RGBA to HEVC	4k	4	0	56	169	55
RGBA to AV1	4k	4	0	60	169	49
RGBA to AVC	4k	8	0	63	174	42
RGBA to HEVC	4k	8	0	55	176	43
RGBA to AV1	4k	8	0	59	176	33
RGBA to AVC	1080p	1	1	29	385	38
RGBA to HEVC	1080p	1	1	28	385	42
RGBA to AV1	1080p	1	1	31	369	38
RGBA to AVC	1080p	16	0	59	639	16
RGBA to HEVC	1080p	16	0	52	640	16
RGBA to AV1	1080p	16	0	55	624	14
RGBA to AVC	1080p	32	0	55	640	9
RGBA to HEVC	1080p	32	0	49	655	9
RGBA to AV1	1080p	32	0	54	640	8
RGBA to AVC	720p	1	1	21	578	40
RGBA to HEVC	720p	1	1	21	578	39
RGBA to AV1	720p	1	1	23	484	27
RGBA to AVC	720p	16	0	40	1038	33
RGBA to HEVC	720p	16	0	38	1031	31
RGBA to AV1	720p	16	0	47	1009	29
RGBA to AVC	720p	32	0	42	1116	22
RGBA to HEVC	720p	32	0	40	1093	21
RGBA to AV1	720p	32	0	49	1058	20

7. T1A – Encoding EnableRdoQuant/rdoLevel/lookaheadDepth

7.1 Encoding

7.1.1 Description

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

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

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

7.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

7.2 Encoding EnableRdoQuant/rdoLevel/lookaheadDepth Performance Results

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	0	0	1	93	288	26
YUV to HEVC	4k	4	0	0	1	96	332	16
YUV to AV1	4k	4	0	0	1	96	288	15
YUV to AVC	4k	4	0	0	2	95	298	24
YUV to HEVC	4k	4	0	0	2	97	168	13
YUV to AV1	4k	4	0	0	2	95	140	13
YUV to AVC	4k	4	0	0	3	95	302	22
YUV to HEVC	4k	4	0	0	3	98	100	7
YUV to AV1	4k	4	0	0	3	99	76	6
YUV to AVC	4k	4	0	1	1	98	188	10
YUV to HEVC	4k	4	0	1	1	97	240	12
YUV to AVC	4k	4	0	1	2	97	188	10
YUV to HEVC	4k	4	0	1	2	100	108	8
YUV to AVC	4k	4	0	1	3	97	188	12
YUV to HEVC	4k	4	0	1	3	98	68	7
YUV to AVC	4k	4	4	0	1	100	196	13
YUV to HEVC	4k	4	4	0	1	99	228	17
YUV to AV1	4k	4	4	0	1	99	200	13
YUV to AVC	4k	4	4	0	2	99	197	11
YUV to HEVC	4k	4	4	0	2	100	136	11
YUV to AV1	4k	4	4	0	2	100	116	8
YUV to AVC	4k	4	4	0	3	99	198	11
YUV to HEVC	4k	4	4	0	3	99	88	7
YUV to AV1	4k	4	4	0	3	100	72	6
YUV to AVC	4k	4	4	1	1	100	140	9
YUV to HEVC	4k	4	4	1	1	100	180	10
YUV to AVC	4k	4	4	1	2	99	140	9
YUV to HEVC	4k	4	4	1	2	99	96	7
YUV to AVC	4k	4	4	1	3	100	140	9
YUV to HEVC	4k	4	4	1	3	98	64	6
YUV to AVC	4k	4	16	0	1	99	196	11
YUV to HEVC	4k	4	16	0	1	100	228	15
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	8
YUV to AVC	4k	4	16	0	3	99	196	12
YUV to HEVC	4k	4	16	0	3	99	88	6

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AV1	4k	4	16	0	3	100	72	6
YUV to AVC	4k	4	16	1	1	100	140	9
YUV to HEVC	4k	4	16	1	1	100	180	11
YUV to AVC	4k	4	16	1	2	100	140	8
YUV to HEVC	4k	4	16	1	2	99	96	7
YUV to AVC	4k	4	16	1	3	100	140	9
YUV to HEVC	4k	4	16	1	3	99	64	6
YUV to AVC	4k	4	40	0	1	100	196	11
YUV to HEVC	4k	4	40	0	1	100	225	12
YUV to AV1	4k	4	40	0	1	100	196	10
YUV to AVC	4k	4	40	0	2	100	196	12
YUV to HEVC	4k	4	40	0	2	99	136	8
YUV to AV1	4k	4	40	0	2	98	112	9
YUV to AVC	4k	4	40	0	3	99	196	11
YUV to HEVC	4k	4	40	0	3	99	88	6
YUV to AV1	4k	4	40	0	3	100	72	6
YUV to AVC	4k	4	40	1	1	100	140	9
YUV to HEVC	4k	4	40	1	1	100	176	10
YUV to AVC	4k	4	40	1	2	99	138	9
YUV to HEVC	4k	4	40	1	2	99	92	6
YUV to AVC	4k	4	40	1	3	99	137	9
YUV to HEVC	4k	4	40	1	3	100	64	6
YUV to AVC	1080p	20	0	0	1	100	1280	5
YUV to HEVC	1080p	20	0	0	1	99	1360	5
YUV to AV1	1080p	20	0	0	1	99	1180	4
YUV to AVC	1080p	20	0	0	2	99	1280	5
YUV to HEVC	1080p	20	0	0	2	99	698	3
YUV to AV1	1080p	20	0	0	2	99	560	3
YUV to AVC	1080p	20	0	0	3	99	1280	4
YUV to HEVC	1080p	20	0	0	3	99	400	3
YUV to AV1	1080p	20	0	0	3	100	300	2
YUV to AVC	1080p	20	0	1	1	99	760	3
YUV to HEVC	1080p	20	0	1	1	99	960	4
YUV to AVC	1080p	20	0	1	2	99	760	3
YUV to HEVC	1080p	20	0	1	2	99	440	3
YUV to AVC	1080p	20	0	1	3	99	760	3
YUV to HEVC	1080p	20	0	1	3	100	280	2
YUV to AVC	1080p	20	4	0	1	99	700	3
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	99	720	3
YUV to AVC	1080p	20	4	0	2	99	701	3
YUV to HEVC	1080p	20	4	0	2	99	520	3
YUV to AV1	1080p	20	4	0	2	100	420	2
YUV to AVC	1080p	20	4	0	3	99	700	3
YUV to HEVC	1080p	20	4	0	3	99	340	2
YUV to AV1	1080p	20	4	0	3	99	277	2
YUV to AVC	1080p	20	4	1	1	99	502	3
YUV to HEVC	1080p	20	4	1	1	99	660	3
YUV to AVC	1080p	20	4	1	2	99	500	3
YUV to HEVC	1080p	20	4	1	2	99	360	2
YUV to AVC	1080p	20	4	1	3	100	500	3
YUV to HEVC	1080p	20	4	1	3	100	240	2
YUV to AVC	1080p	20	16	0	1	99	700	3
YUV to HEVC	1080p	20	16	0	1	99	820	4
YUV to AV1	1080p	20	16	0	1	99	720	3
YUV to AVC	1080p	20	16	0	2	99	700	3
YUV to HEVC	1080p	20	16	0	2	100	508	2
YUV to AV1	1080p	20	16	0	2	100	420	3
YUV to AVC	1080p	20	16	0	3	100	700	3
YUV to HEVC	1080p	20	16	0	3	100	340	2
YUV to AV1	1080p	20	16	0	3	99	265	2
YUV to AVC	1080p	20	16	1	1	99	500	3
YUV to HEVC	1080p	20	16	1	1	99	656	3
YUV to AVC	1080p	20	16	1	2	100	500	3
YUV to HEVC	1080p	20	16	1	2	99	360	2
YUV to AVC	1080p	20	16	1	3	99	500	3
YUV to HEVC	1080p	20	16	1	3	99	240	2
YUV to AVC	1080p	20	40	0	1	99	695	3
YUV to HEVC	1080p	20	40	0	1	99	804	3
YUV to AV1	1080p	20	40	0	1	99	704	3
YUV to AVC	1080p	20	40	0	2	99	695	3
YUV to HEVC	1080p	20	40	0	2	99	500	3
YUV to AV1	1080p	20	40	0	2	99	420	2
YUV to AVC	1080p	20	40	0	3	99	696	3
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	100	500	3
YUV to HEVC	1080p	20	40	1	1	99	640	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	2	99	500	3
YUV to HEVC	1080p	20	40	1	2	99	360	2
YUV to AVC	1080p	20	40	1	3	99	500	3
YUV to HEVC	1080p	20	40	1	3	100	240	2
YUV to AVC	720p	40	0	0	1	92	2320	3
YUV to HEVC	720p	40	0	0	1	89	2319	3
YUV to AV1	720p	40	0	0	1	95	2040	2
YUV to AVC	720p	40	0	0	2	92	2320	3
YUV to HEVC	720p	40	0	0	2	99	1520	2
YUV to AV1	720p	40	0	0	2	99	1200	1
YUV to AVC	720p	40	0	0	3	92	2320	3
YUV to HEVC	720p	40	0	0	3	100	920	1
YUV to AV1	720p	40	0	0	3	100	655	1
YUV to AVC	720p	40	0	1	1	99	1684	2
YUV to HEVC	720p	40	0	1	1	99	2080	2
YUV to AVC	720p	40	0	1	2	99	1684	2
YUV to HEVC	720p	40	0	1	2	99	960	1
YUV to AVC	720p	40	0	1	3	99	1684	2
YUV to HEVC	720p	40	0	1	3	99	623	1
YUV to AVC	720p	40	4	0	1	99	1440	1
YUV to HEVC	720p	40	4	0	1	100	1360	1
YUV to AV1	720p	40	4	0	1	100	1000	1
YUV to AVC	720p	40	4	0	2	99	1440	1
YUV to HEVC	720p	40	4	0	2	99	1080	1
YUV to AV1	720p	40	4	0	2	99	880	1
YUV to AVC	720p	40	4	0	3	100	1440	1
YUV to HEVC	720p	40	4	0	3	99	720	1
YUV to AV1	720p	40	4	0	3	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	1078	1
YUV to HEVC	720p	40	4	1	2	99	760	1
YUV to AVC	720p	40	4	1	3	99	1080	1
YUV to HEVC	720p	40	4	1	3	99	520	1
YUV to AVC	720p	40	16	0	1	100	1360	1
YUV to HEVC	720p	40	16	0	1	100	1360	1
YUV to AV1	720p	40	16	0	1	100	1079	1
YUV to AVC	720p	40	16	0	2	100	1362	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	1394	1
YUV to HEVC	720p	40	16	0	3	99	720	1
YUV to AV1	720p	40	16	0	3	99	560	1
YUV to AVC	720p	40	16	1	1	99	1049	1
YUV to HEVC	720p	40	16	1	1	100	1323	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	1048	1
YUV to HEVC	720p	40	16	1	3	100	520	1
YUV to AVC	720p	40	40	0	1	100	1353	1
YUV to HEVC	720p	40	40	0	1	100	1320	1
YUV to AV1	720p	40	40	0	1	100	1040	1
YUV to AVC	720p	40	40	0	2	100	1325	1
YUV to HEVC	720p	40	40	0	2	99	1040	1
YUV to AV1	720p	40	40	0	2	99	856	1
YUV to AVC	720p	40	40	0	3	100	1352	1
YUV to HEVC	720p	40	40	0	3	99	720	1
YUV to AV1	720p	40	40	0	3	100	560	1
YUV to AVC	720p	40	40	1	1	99	1040	1
YUV to HEVC	720p	40	40	1	1	100	1320	1
YUV to AVC	720p	40	40	1	2	99	1040	1
YUV to HEVC	720p	40	40	1	2	99	760	1
YUV to AVC	720p	40	40	1	3	99	1040	1
YUV to HEVC	720p	40	40	1	3	100	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	3
YUV to HEVC	1080p	20	0	0	1	19	99	820	4
YUV to AV1	1080p	20	0	0	1	19	99	720	3
YUV to AVC	1080p	20	0	0	2	19	100	700	3
YUV to HEVC	1080p	20	0	0	2	19	100	520	3
YUV to AV1	1080p	20	0	0	2	19	99	440	2
YUV to AVC	1080p	20	0	0	3	19	99	702	3
YUV to HEVC	1080p	20	0	0	3	19	99	340	3
YUV to AV1	1080p	20	0	0	3	19	100	280	2
YUV to AVC	1080p	20	0	1	1	19	99	511	3
YUV to HEVC	1080p	20	0	1	1	19	99	660	3
YUV to AVC	1080p	20	0	1	2	19	100	507	3
YUV to HEVC	1080p	20	0	1	2	19	100	360	2
YUV to AVC	1080p	20	0	1	3	19	99	509	3
YUV to HEVC	1080p	20	0	1	3	19	100	240	2
YUV to AVC	1080p	20	4	0	1	19	99	700	3
YUV to HEVC	1080p	20	4	0	1	19	99	820	4
YUV to AV1	1080p	20	4	0	1	19	99	720	3
YUV to AVC	1080p	20	4	0	2	19	99	700	3
YUV to HEVC	1080p	20	4	0	2	19	99	520	2
YUV to AV1	1080p	20	4	0	2	19	99	420	2
YUV to AVC	1080p	20	4	0	3	19	99	701	3
YUV to HEVC	1080p	20	4	0	3	19	100	340	2
YUV to AV1	1080p	20	4	0	3	19	100	275	2
YUV to AVC	1080p	20	4	1	1	19	99	501	3
YUV to HEVC	1080p	20	4	1	1	19	100	660	3
YUV to AVC	1080p	20	4	1	2	19	99	500	3
YUV to HEVC	1080p	20	4	1	2	19	100	360	2
YUV to AVC	1080p	20	4	1	3	19	99	500	3
YUV to HEVC	1080p	20	4	1	3	19	100	240	2
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	3
YUV to AVC	1080p	20	16	0	2	19	100	700	3
YUV to HEVC	1080p	20	16	0	2	19	99	504	2
YUV to AV1	1080p	20	16	0	2	19	99	420	2
YUV to AVC	1080p	20	16	0	3	19	99	700	3
YUV to HEVC	1080p	20	16	0	3	19	100	340	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	3	19	99	263	2
YUV to AVC	1080p	20	16	1	1	19	100	500	3
YUV to HEVC	1080p	20	16	1	1	19	99	658	3
YUV to AVC	1080p	20	16	1	2	19	99	500	3
YUV to HEVC	1080p	20	16	1	2	19	100	360	2
YUV to AVC	1080p	20	16	1	3	19	99	500	3
YUV to HEVC	1080p	20	16	1	3	19	99	240	2
YUV to AVC	1080p	20	40	0	1	19	99	694	3
YUV to HEVC	1080p	20	40	0	1	19	99	804	4
YUV to AV1	1080p	20	40	0	1	19	99	704	3
YUV to AVC	1080p	20	40	0	2	19	99	697	3
YUV to HEVC	1080p	20	40	0	2	19	100	500	3
YUV to AV1	1080p	20	40	0	2	19	99	420	3
YUV to AVC	1080p	20	40	0	3	19	99	698	3
YUV to HEVC	1080p	20	40	0	3	19	100	340	2
YUV to AV1	1080p	20	40	0	3	19	100	260	3
YUV to AVC	1080p	20	40	1	1	19	100	500	3
YUV to HEVC	1080p	20	40	1	1	19	100	640	3
YUV to AVC	1080p	20	40	1	2	19	100	500	3
YUV to HEVC	1080p	20	40	1	2	19	100	360	2
YUV to AVC	1080p	20	40	1	3	19	100	500	3
YUV to HEVC	1080p	20	40	1	3	19	100	240	2
YUV to AVC	1080p	20	0	0	1	23	99	702	3
YUV to HEVC	1080p	20	0	0	1	23	99	820	4
YUV to AV1	1080p	20	0	0	1	23	99	720	3
YUV to AVC	1080p	20	0	0	2	23	99	702	3
YUV to HEVC	1080p	20	0	0	2	23	100	520	3
YUV to AV1	1080p	20	0	0	2	23	100	440	2
YUV to AVC	1080p	20	0	0	3	23	99	703	3
YUV to HEVC	1080p	20	0	0	3	23	99	340	3
YUV to AV1	1080p	20	0	0	3	23	100	280	2
YUV to AVC	1080p	20	0	1	1	23	99	508	3
YUV to HEVC	1080p	20	0	1	1	23	99	660	3
YUV to AVC	1080p	20	0	1	2	23	100	509	2
YUV to HEVC	1080p	20	0	1	2	23	100	360	2
YUV to AVC	1080p	20	0	1	3	23	99	510	3
YUV to HEVC	1080p	20	0	1	3	23	100	240	2
YUV to AVC	1080p	20	4	0	1	23	100	700	3
YUV to HEVC	1080p	20	4	0	1	23	99	820	4

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	23	99	720	3
YUV to AVC	1080p	20	4	0	2	23	99	700	3
YUV to HEVC	1080p	20	4	0	2	23	99	520	3
YUV to AV1	1080p	20	4	0	2	23	100	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	2
YUV to AV1	1080p	20	4	0	3	23	99	273	2
YUV to AVC	1080p	20	4	1	1	23	99	501	3
YUV to HEVC	1080p	20	4	1	1	23	99	660	3
YUV to AVC	1080p	20	4	1	2	23	100	501	3
YUV to HEVC	1080p	20	4	1	2	23	99	360	2
YUV to AVC	1080p	20	4	1	3	23	100	501	3
YUV to HEVC	1080p	20	4	1	3	23	100	240	2
YUV to AVC	1080p	20	16	0	1	23	99	700	3
YUV to HEVC	1080p	20	16	0	1	23	99	820	4
YUV to AV1	1080p	20	16	0	1	23	99	720	3
YUV to AVC	1080p	20	16	0	2	23	99	700	3
YUV to HEVC	1080p	20	16	0	2	23	99	509	3
YUV to AV1	1080p	20	16	0	2	23	99	420	2
YUV to AVC	1080p	20	16	0	3	23	99	700	3
YUV to HEVC	1080p	20	16	0	3	23	99	340	3
YUV to AV1	1080p	20	16	0	3	23	99	264	2
YUV to AVC	1080p	20	16	1	1	23	99	500	3
YUV to HEVC	1080p	20	16	1	1	23	99	655	3
YUV to AVC	1080p	20	16	1	2	23	100	500	3
YUV to HEVC	1080p	20	16	1	2	23	99	360	2
YUV to AVC	1080p	20	16	1	3	23	99	500	3
YUV to HEVC	1080p	20	16	1	3	23	100	240	2
YUV to AVC	1080p	20	40	0	1	23	99	698	3
YUV to HEVC	1080p	20	40	0	1	23	99	803	4
YUV to AV1	1080p	20	40	0	1	23	99	706	3
YUV to AVC	1080p	20	40	0	2	23	99	698	3
YUV to HEVC	1080p	20	40	0	2	23	99	500	3
YUV to AV1	1080p	20	40	0	2	23	99	420	3
YUV to AVC	1080p	20	40	0	3	23	100	695	3
YUV to HEVC	1080p	20	40	0	3	23	100	340	2
YUV to AV1	1080p	20	40	0	3	23	100	260	2
YUV to AVC	1080p	20	40	1	1	23	99	500	3
YUV to HEVC	1080p	20	40	1	1	23	99	640	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	2	23	99	500	3
YUV to HEVC	1080p	20	40	1	2	23	99	360	2
YUV to AVC	1080p	20	40	1	3	23	99	500	3
YUV to HEVC	1080p	20	40	1	3	23	100	240	2
YUV to AVC	1080p	20	0	0	1	27	99	702	3
YUV to HEVC	1080p	20	0	0	1	27	99	820	3
YUV to AV1	1080p	20	0	0	1	27	99	720	3
YUV to AVC	1080p	20	0	0	2	27	99	701	3
YUV to HEVC	1080p	20	0	0	2	27	99	520	3
YUV to AV1	1080p	20	0	0	2	27	99	440	2
YUV to AVC	1080p	20	0	0	3	27	99	702	3
YUV to HEVC	1080p	20	0	0	3	27	99	340	3
YUV to AV1	1080p	20	0	0	3	27	99	280	2
YUV to AVC	1080p	20	0	1	1	27	99	509	3
YUV to HEVC	1080p	20	0	1	1	27	99	660	3
YUV to AVC	1080p	20	0	1	2	27	100	510	3
YUV to HEVC	1080p	20	0	1	2	27	99	360	2
YUV to AVC	1080p	20	0	1	3	27	99	510	3
YUV to HEVC	1080p	20	0	1	3	27	100	240	2
YUV to AVC	1080p	20	4	0	1	27	99	701	3
YUV to HEVC	1080p	20	4	0	1	27	99	820	3
YUV to AV1	1080p	20	4	0	1	27	99	720	3
YUV to AVC	1080p	20	4	0	2	27	99	700	3
YUV to HEVC	1080p	20	4	0	2	27	99	520	3
YUV to AV1	1080p	20	4	0	2	27	99	420	2
YUV to AVC	1080p	20	4	0	3	27	99	700	3
YUV to HEVC	1080p	20	4	0	3	27	99	340	2
YUV to AV1	1080p	20	4	0	3	27	100	275	2
YUV to AVC	1080p	20	4	1	1	27	100	500	3
YUV to HEVC	1080p	20	4	1	1	27	99	660	3
YUV to AVC	1080p	20	4	1	2	27	99	502	3
YUV to HEVC	1080p	20	4	1	2	27	100	360	2
YUV to AVC	1080p	20	4	1	3	27	100	502	3
YUV to HEVC	1080p	20	4	1	3	27	100	240	2
YUV to AVC	1080p	20	16	0	1	27	100	700	3
YUV to HEVC	1080p	20	16	0	1	27	99	820	4
YUV to AV1	1080p	20	16	0	1	27	99	720	3
YUV to AVC	1080p	20	16	0	2	27	100	700	3
YUV to HEVC	1080p	20	16	0	2	27	99	506	3

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
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	100	340	2
YUV to AV1	1080p	20	16	0	3	27	99	265	2
YUV to AVC	1080p	20	16	1	1	27	99	500	3
YUV to HEVC	1080p	20	16	1	1	27	100	655	3
YUV to AVC	1080p	20	16	1	2	27	99	500	3
YUV to HEVC	1080p	20	16	1	2	27	99	360	2
YUV to AVC	1080p	20	16	1	3	27	100	500	3
YUV to HEVC	1080p	20	16	1	3	27	100	240	2
YUV to AVC	1080p	20	40	0	1	27	99	695	3
YUV to HEVC	1080p	20	40	0	1	27	99	807	3
YUV to AV1	1080p	20	40	0	1	27	99	706	3
YUV to AVC	1080p	20	40	0	2	27	99	699	3
YUV to HEVC	1080p	20	40	0	2	27	99	500	3
YUV to AV1	1080p	20	40	0	2	27	99	420	2
YUV to AVC	1080p	20	40	0	3	27	99	694	3
YUV to HEVC	1080p	20	40	0	3	27	99	340	2
YUV to AV1	1080p	20	40	0	3	27	99	260	2
YUV to AVC	1080p	20	40	1	1	27	100	500	3
YUV to HEVC	1080p	20	40	1	1	27	100	640	3
YUV to AVC	1080p	20	40	1	2	27	100	500	3
YUV to HEVC	1080p	20	40	1	2	27	100	360	2
YUV to AVC	1080p	20	40	1	3	27	99	500	3
YUV to HEVC	1080p	20	40	1	3	27	99	240	2

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.rgba -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	FPS	CPU	DEC_LOAD	ENC_LOAD	SCALER_LOAD
AVC to AVC	1	289	15	12	21	5
AVC to HEVC	1	306	16	13	20	5
AVC to AV1	1	262	15	11	20	4
HEVC to AVC	1	285	19	12	20	5
HEVC to HEVC	1	301	20	13	20	5
HEVC to AV1	1	261	23	11	20	4
VP9 to AVC	1	288	15	17	20	5
VP9 to HEVC	1	306	15	17	20	5
VP9 to AV1	1	262	15	15	20	4
AVC to AVC	16	1047	4	72	95	24
AVC to HEVC	16	1168	4	77	93	27
AVC to AV1	16	1120	4	72	94	25
HEVC to AVC	16	1088	5	70	92	24
HEVC to HEVC	16	1216	5	76	93	27
HEVC to AV1	16	1148	5	68	95	26
VP9 to AVC	16	1043	4	81	93	23
VP9 to HEVC	16	1171	4	91	95	26
VP9 to AV1	16	1127	4	82	93	25
AVC to AVC	32	960	2	74	93	23
AVC to HEVC	32	1088	2	81	93	26
AVC to AV1	32	1056	2	76	93	25
HEVC to AVC	32	1014	2	71	92	24
HEVC to HEVC	32	1149	2	78	92	27
HEVC to AV1	32	1088	2	74	95	26
VP9 to AVC	32	992	2	82	95	23
VP9 to HEVC	32	1120	2	90	94	26
VP9 to AV1	32	1088	2	84	94	25

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

10.1 Encoding

10.1.1 Description

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

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

The YUV frame is encoded with hardware encoder.

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

10.1.2 Command line

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

<num_jobs> = number of instances running concurrently

10.2 Multi Thread P2P DMA on AMD GPU Performance Results

TYPE	RES	JOBS	Frames	FPS	CPU	Enc Load	P2P MEM	Latency Avg	Latency Dev
P2A	720p	1	1001	30	1.3	0.25	0.5	1.99	0
P2H	720p	1	1001	30	1.1	0.25	0.5	2	0.02
P2A	720p	180	180180	29.6	0.77	46	56	2.38	0.18
P2H	720p	180	180180	29.7	0.75	45	56	2.33	0.1
P2A	1080p	1	1001	29	1.3	0.5	0.75	3.73	0.01
P2H	1080p	1	1001	30	1.2	0.5	0.75	3.51	0.02
P2A	1080p	80	80080	29.6	0.88	46	50	3.88	0.11
P2H	1080p	80	80080	29.9	0.82	43	50	3.7	0.12

11. T1A – AI

11.1 AI Model

11.1.1 Description

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

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

aiperf performs any data format conversion expected by the hardware.

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

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

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

11.1.2 Command line

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

In config_example.json, user need to specify the following arguments
```
{
        "nb":"/path/to/network_binary_0.nb",
        "dataset":"/path/to/dataset0.txt",
        "outdir":"/path/to/output0",
        "format":"nchw",
        "order":"rgb",
        "devid":"0",
        "loop": "10000"
}
```
In dataset0.txt, user need to specify the path to the input batch (image or tensor)
```
/path/to/image.png
```

11.1.3 AI Model Performance Results

Model	Session Number	Loops	Channel Order	File Format	File Type	Input Size	FPS per session
yolov5s_640	8	10000	rgb	nchw	image	640x640x3	78
yolov5s_320	8	10000	rgb	nchw	image	320x320x3	278
deeplabv3_FRP	8	10000	rgb	nchw	image	257x257x3	359
resnet50	8	10000	rgb	nchw	image	224x224x3	228
mobilenetv2	8	10000	rgb	nchw	image	224x224x3	1111
deeplabv3	8	10000	rgb	nchw	image	257x257x3	160
yolov4	8	10000	bgr	nchw	image	416x416x3	256
fsrcnn	8	2000	bgr	nchw	image	360x640x1	31
BiSeNetv1	8	10000	rgb	nchw	image	512x512x3	77
HrNet	8	10000	rgb	nchw	image	256x192x3	74
usm_1656x1920	8	10000	rgb	nchw	image	1920x1656x1	251
usm_3240x3840	8	10000	rgb	nchw	image	3840x3240x1	61
lpips	8	2000	rgb	nchw	image	720x480x3	1
PaddleOCR-512_onnx	8	10000	rgb	nchw	image	512x48x3	16
segm32_tflite_kl_mle	8	10000	rgb	nchw	image	256x144x3	837
mobilenetv2_nchw_keras_96x160	1	10000	rgb	nchw	image	96x160x3	2349.5
mobilenetv2_nchw_keras_96x160	8	10000	rgb	nchw	image	96x160x3	2341.75
mobilenetv2_nchw_keras_96x160	16	10000	rgb	nchw	image	96x160x3	2330

11.2 AI Encoding with 2D Engine

11.2.1 Description BG Filter

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

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

11.2.2 Description ROI Filter

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

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

11.2.3 Description PRE Filter

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

11.2.4 Command line BG

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

11.2.5 Command line ROI

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

11.2.6 Command line PRE

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

11.2.7 AI Encoding with 2D Engine Performance Results

Filter	Model	Resolution	Session Number	Average FPS per session
ROI	network_binary_yolov4_head	1920x1080	1	76
ROI	network_binary_yolov4_head	1920x1080	8	24
ROI	network_binary_yolov4_head	1920x1080	32	5
BG	segm32_tflite_nchw_bgr	1920x1080	1	69
BG	segm32_tflite_nchw_bgr	1920x1080	8	41
BG	segm32_tflite_nchw_bgr	1920x1080	32	14
PRE	hw_lanczos_network_binary	1920x1080	8	66
PRE	hw_lanczos_network_binary	1920x1080	16	34
PRE	hw_bicubic_network_binary	1920x1080	8	67
PRE	hw_bicubic_network_binary	1920x1080	16	34

12. T1A – GStreamer XStack Throughput

12.1 Transcoding

12.1.1 Description

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

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

The YUV frame is encoded with hardware encoder.

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

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

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

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

This test is HEVC to AVC only.
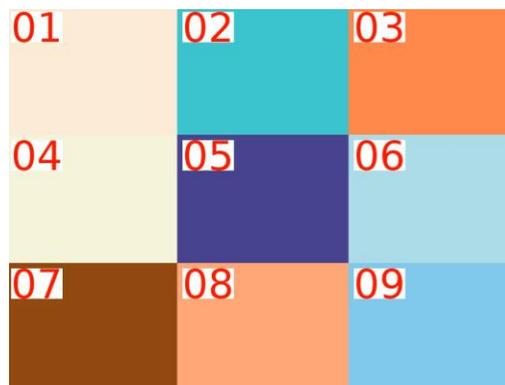
Example output in a 3x3 layout with 9 inputs

12.1.2 Command line

See Appendix A: GStreamer XStack Command

12.2 GStreamer XStack Performance Results

Input Res	Grid	Output Res	Cell Size	FPS	CPU	Dec Load	Enc Load	Scaler Load
1920x1080	3x3	1920x1080	640x360	152.31	61	65	13	11
1920x1080	4x4	1920x1080	480x270	104.13	73	79	10	11
1920x1080	4x8	1920x1080	480x135	57.04	75	87	5	6
1920x1080	3x3	3840x2160	1280x720	69.92	30	30	21	11
1920x1080	4x4	3840x2160	960x540	61.09	43	51	21	15
1920x1080	4x8	3840x2160	960x270	46.29	59	75	17	14
1920x1080	3x3	7680x4320	2560x1440	19.91	16	8	22	9
1920x1080	4x4	7680x4320	1920x1080	19.21	18	15	22	10
1920x1080	4x8	7680x4320	1920x540	19.4	28	31	22	11
1920x1080	6x6	1920x1080	320x180	50.42	82	88	5	8
1920x1080	7x7	1920x1080	274x154 276x154 274x156 276x156*	37.22	80	86	3	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	15.1	1034	0	72

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:semiplanar0=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	87	0	98	14
HEVC to YUV	8k	1	0	8	1	81	0	96	15
VP9 to YUV	8k	1	0	8	1	25	0	30	2
YUV to AVC	8k	1	0	8	1	0	94	57	78
YUV to HEVC	8k	1	0	8	1	0	97	71	100
AVC to AVC	8k	1	1	8	1	76	99	48	2
AVC to HEVC	8k	1	1	8	1	72	93	64	9
HEVC to AVC	8k	1	1	8	1	62	100	47	2
HEVC to HEVC	8k	1	1	8	1	66	96	63	10
VP9 to AVC	8k	1	1	8	1	24	41	29	1
VP9 to HEVC	8k	1	1	8	1	25	41	30	1
AVC to YUV	8k	1	0	10	1	61	0	54	7
HEVC to YUV	8k	1	0	10	1	95	0	56	8
VP9 to YUV	8k	1	0	10	1	24	0	28	4
YUV to AVC	8k	1	0	10	1	0	88	40	112
YUV to HEVC	8k	1	0	10	1	0	56	41	118
AVC to YUV	4k	1	0	8	1	58	0	267	16
HEVC to YUV	4k	1	0	8	1	52	0	281	18
VP9 to YUV	4k	1	0	8	1	24	0	108	7
AVC to YUV	4k	16	0	8	0	100	0	419	1
HEVC to YUV	4k	16	0	8	0	97	0	447	0
VP9 to YUV	4k	16	0	8	0	100	0	403	1
YUV to AVC	4k	1	0	8	1	0	95	255	33
YUV to HEVC	4k	1	0	8	1	0	95	283	39
YUV to AV1	4k	1	0	8	1	0	94	242	28
YUV to AVC	4k	4	0	8	0	0	95	264	18
YUV to HEVC	4k	4	0	8	0	0	97	286	13
YUV to AV1	4k	4	0	8	0	0	97	248	11
YUV to AVC	4k	8	0	8	0	0	99	280	8
YUV to HEVC	4k	8	0	8	0	0	99	296	8
YUV to AV1	4k	8	0	8	0	0	99	256	7
AVC to AVC	4k	1	1	8	1	72	94	196	9
AVC to HEVC	4k	1	1	8	1	69	89	244	8
AVC to AV1	4k	1	1	8	1	58	89	225	8
HEVC to AVC	4k	1	1	8	1	51	86	192	12
HEVC to HEVC	4k	1	1	8	1	55	87	236	14
HEVC to AV1	4k	1	1	8	1	47	88	219	15
VP9 to AVC	4k	1	1	8	1	23	38	107	2
VP9 to HEVC	4k	1	1	8	1	24	36	108	3

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to AV1	4k	1	1	8	1	24	42	108	7
AVC to AVC	4k	4	1	8	0	63	96	220	2
AVC to HEVC	4k	4	1	8	0	68	96	264	1
AVC to AV1	4k	4	1	8	0	58	95	240	6
HEVC to AVC	4k	4	1	8	0	55	96	212	1
HEVC to HEVC	4k	4	1	8	0	56	96	252	6
HEVC to AV1	4k	4	1	8	0	46	91	240	7
VP9 to AVC	4k	4	1	8	0	65	96	218	1
VP9 to HEVC	4k	4	1	8	0	69	96	264	1
VP9 to AV1	4k	4	1	8	0	60	93	236	5
AVC to AVC	4k	8	1	8	0	67	99	195	1
AVC to HEVC	4k	8	1	8	0	75	100	256	1
AVC to AV1	4k	8	1	8	0	71	100	240	5
HEVC to AVC	4k	8	1	8	0	59	99	196	1
HEVC to HEVC	4k	8	1	8	0	62	100	256	1
HEVC to AV1	4k	8	1	8	0	59	99	240	7
VP9 to AVC	4k	8	1	8	0	66	99	203	1
VP9 to HEVC	4k	8	1	8	0	73	100	259	1
VP9 to AV1	4k	8	1	8	0	65	100	248	5
AVC to YUV	4k	1	0	10	0	47	0	201	8
HEVC to YUV	4k	1	0	10	0	52	0	208	6
VP9 to YUV	4k	1	0	10	0	24	0	125	3
AVC to YUV	4k	16	0	10	0	99	0	254	0
HEVC to YUV	4k	16	0	10	0	99	0	253	0
VP9 to YUV	4k	16	0	10	0	100	0	435	0
YUV to AVC	4k	1	0	10	0	0	68	172	50
YUV to HEVC	4k	1	0	10	0	0	62	181	58
YUV to AV1	4k	1	0	10	0	0	70	177	54
YUV to AVC	4k	4	0	10	0	0	91	193	26
YUV to HEVC	4k	4	0	10	0	0	80	227	33
YUV to AV1	4k	4	0	10	0	0	87	218	31
AVC to YUV	1080p	1	0	8	1	44	0	721	18
HEVC to YUV	1080p	1	0	8	1	47	0	709	24
VP9 to YUV	1080p	1	0	8	1	22	0	448	3
AVC to YUV	1080p	40	0	8	0	97	0	1523	1
HEVC to YUV	1080p	40	0	8	0	98	0	1528	1
VP9 to YUV	1080p	40	0	8	0	94	0	1639	0
YUV to AVC	1080p	1	0	8	1	0	58	653	19
YUV to HEVC	1080p	1	0	8	1	0	55	651	19
YUV to AV1	1080p	1	0	8	1	0	60	603	19

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	44	61	681	19
AVC to HEVC	1080p	1	1	8	1	43	56	677	14
AVC to AV1	1080p	1	1	8	1	39	63	618	15
HEVC to AVC	1080p	1	1	8	1	46	61	664	26
HEVC to HEVC	1080p	1	1	8	1	45	57	671	24
HEVC to AV1	1080p	1	1	8	1	42	63	615	22
VP9 to AVC	1080p	1	1	8	1	22	40	446	6
VP9 to HEVC	1080p	1	1	8	1	22	37	447	4
VP9 to AV1	1080p	1	1	8	1	22	44	445	4
AVC to AVC	1080p	32	1	8	0	74	99	864	0
AVC to HEVC	1080p	32	1	8	0	85	99	973	1
AVC to AV1	1080p	32	1	8	0	79	99	928	2
HEVC to AVC	1080p	32	1	8	0	72	99	898	1
HEVC to HEVC	1080p	32	1	8	0	75	99	1000	1
HEVC to AV1	1080p	32	1	8	0	71	99	938	3
VP9 to AVC	1080p	32	1	8	0	62	99	992	0
VP9 to HEVC	1080p	32	1	8	0	68	99	1088	0
VP9 to AV1	1080p	32	1	8	0	55	99	992	1
AVC to YUV	1080p	1	0	10	0	30	0	507	9
HEVC to YUV	1080p	1	0	10	0	27	0	519	6
VP9 to YUV	1080p	1	0	10	0	22	0	457	6
AVC to YUV	1080p	40	0	10	0	71	0	1040	0
HEVC to YUV	1080p	40	0	10	0	77	0	1047	0
VP9 to YUV	1080p	40	0	10	0	94	0	1680	0
YUV to AVC	1080p	1	0	10	0	0	36	431	31
YUV to HEVC	1080p	1	0	10	0	0	34	430	27
YUV to AV1	1080p	1	0	10	0	0	37	402	24
YUV to AVC	1080p	32	0	10	0	0	72	796	6
YUV to HEVC	1080p	32	0	10	0	0	67	800	6
YUV to AV1	1080p	32	0	10	0	0	74	768	5
AVC to YUV	720p	1	0	8	1	43	0	1068	16
HEVC to YUV	720p	1	0	8	1	38	0	1053	25
VP9 to YUV	720p	1	0	8	1	29	0	846	4
AVC to YUV	720p	100	0	8	0	100	0	2322	0
HEVC to YUV	720p	100	0	8	0	99	0	2670	0
VP9 to YUV	720p	100	0	8	0	100	0	2421	0
YUV to AVC	720p	1	0	8	1	0	32	811	18

TYPE	RES	JOB	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	720p	1	0	8	1	0	32	824	19
YUV to AV1	720p	1	0	8	1	0	34	743	8
YUV to AVC	720p	64	0	8	0	0	99	2231	2
YUV to HEVC	720p	64	0	8	0	0	96	2268	2
YUV to AV1	720p	64	0	8	0	0	99	1902	1
AVC to AVC	720p	1	1	8	1	28	30	754	15
AVC to HEVC	720p	1	1	8	1	28	30	752	14
AVC to AV1	720p	1	1	8	1	25	32	678	13
HEVC to AVC	720p	1	1	8	1	26	30	762	19
HEVC to HEVC	720p	1	1	8	1	26	30	765	18
HEVC to AV1	720p	1	1	8	1	22	32	687	15
VP9 to AVC	720p	1	1	8	1	27	32	796	8
VP9 to HEVC	720p	1	1	8	1	27	32	806	8
VP9 to AV1	720p	1	1	8	1	24	33	705	6
AVC to AVC	720p	64	1	8	0	93	100	1847	1
AVC to HEVC	720p	64	1	8	0	96	100	1926	1
AVC to AV1	720p	64	1	8	0	75	100	1620	0
HEVC to AVC	720p	64	1	8	0	83	99	1839	1
HEVC to HEVC	720p	64	1	8	0	85	100	1921	1
HEVC to AV1	720p	64	1	8	0	67	100	1635	0
VP9 to AVC	720p	64	1	8	0	96	100	2052	0
VP9 to HEVC	720p	64	1	8	0	99	100	2119	0
VP9 to AV1	720p	64	1	8	0	77	100	1765	0

15. T1U – Libxcoder Throughput

15.1 Decoding

15.1.1 Description

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

Bitstream is decoded by hardware decoder.

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

15.1.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

15.2 Encoding

15.2.1 Description

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

YUV frame is encoded by hardware encoder.

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

15.2.2 Command Line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

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

15.3 Transcoding

15.3.1 Description

Bitstream is read from an input file on ramdisk and then fed into hardware decoder through PCIe. Bitstream is decoded by hardware decoder.
Decoded YUV frame is kept on device.
The YUV frame is encoded with hardware encoder.
The encoded bitstream is read out through PCIe and written into an output file.

15.3.2 Command line

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

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<resolution> == 8k, multicoreJointMode = 1

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

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

15.4   Libxcoder 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	78	0	86	7
HEVC to YUV	8k	1	0	8	1	81	0	89	6
VP9 to YUV	8k	1	0	8	1	24	0	30	3
YUV to AVC	8k	1	0	8	1	0	90	56	25
YUV to HEVC	8k	1	0	8	1	0	97	70	32
AVC to AVC	8k	1	1	8	1	73	100	48	3
AVC to HEVC	8k	1	1	8	1	77	99	64	4
HEVC to AVC	8k	1	1	8	1	56	100	47	2
HEVC to HEVC	8k	1	1	8	1	71	100	62	3
VP9 to AVC	8k	1	1	8	1	25	41	29	2
VP9 to HEVC	8k	1	1	8	1	24	39	30	2
AVC to YUV	8k	1	0	10	1	60	0	49	6
HEVC to YUV	8k	1	0	10	1	55	0	47	6
VP9 to YUV	8k	1	0	10	1	24	0	28	3
YUV to AVC	8k	1	0	10	1	0	62	34	35
YUV to HEVC	8k	1	0	10	1	0	49	35	36
AVC to YUV	4k	1	0	8	1	58	0	265	8
HEVC to YUV	4k	1	0	8	1	52	0	274	6
VP9 to YUV	4k	1	0	8	1	24	0	108	2
AVC to YUV	4k	16	0	8	0	98	0	420	1
HEVC to YUV	4k	16	0	8	0	96	0	449	0
VP9 to YUV	4k	16	0	8	0	100	0	401	0
YUV to AVC	4k	1	0	8	1	0	84	229	22
YUV to HEVC	4k	1	0	8	1	0	81	238	22
YUV to AV1	4k	1	0	8	1	0	86	219	21
YUV to AVC	4k	4	0	8	0	0	96	267	10
YUV to HEVC	4k	4	0	8	0	0	96	287	11
YUV to AV1	4k	4	0	8	0	0	96	246	10
YUV to AVC	4k	8	0	8	0	0	100	280	8
YUV to HEVC	4k	8	0	8	0	0	99	299	9
YUV to AV1	4k	8	0	8	0	0	100	259	8
AVC to AVC	4k	1	1	8	0	70	97	202	6
AVC to HEVC	4k	1	1	8	0	73	96	248	7
AVC to AV1	4k	1	1	8	0	62	97	232	8
HEVC to AVC	4k	1	1	8	0	57	97	203	4
HEVC to HEVC	4k	1	1	8	0	62	95	253	4
HEVC to AV1	4k	1	1	8	0	53	97	233	4
VP9 to AVC	4k	1	1	8	0	23	38	108	2

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
VP9 to HEVC	4k	1	1	8	0	24	36	109	2
VP9 to AV1	4k	1	1	8	0	24	42	108	2
AVC to AVC	4k	4	1	8	0	62	97	223	4
AVC to HEVC	4k	4	1	8	0	68	96	270	4
AVC to AV1	4k	4	1	8	0	54	95	243	4
HEVC to AVC	4k	4	1	8	0	55	96	219	3
HEVC to HEVC	4k	4	1	8	0	59	96	264	3
HEVC to AV1	4k	4	1	8	0	48	96	241	3
VP9 to AVC	4k	4	1	8	0	64	96	218	3
VP9 to HEVC	4k	4	1	8	0	69	95	266	3
VP9 to AV1	4k	4	1	8	0	58	95	241	3
AVC to AVC	4k	8	1	8	0	67	100	198	3
AVC to HEVC	4k	8	1	8	0	76	100	255	3
AVC to AV1	4k	8	1	8	0	64	99	245	3
HEVC to AVC	4k	8	1	8	0	63	99	198	2
HEVC to HEVC	4k	8	1	8	0	67	100	253	3
HEVC to AV1	4k	8	1	8	0	58	99	243	3
VP9 to AVC	4k	8	1	8	0	93	99	202	2
VP9 to HEVC	4k	8	1	8	0	80	99	261	2
VP9 to AV1	4k	8	1	8	0	81	99	249	3
AVC to YUV	4k	1	0	10	1	49	0	187	6
HEVC to YUV	4k	1	0	10	1	44	0	184	6
VP9 to YUV	4k	1	0	10	1	24	0	126	2
AVC to YUV	4k	16	0	10	0	98	0	256	0
HEVC to YUV	4k	16	0	10	0	90	0	255	0
VP9 to YUV	4k	16	0	10	0	98	0	437	0
YUV to AVC	4k	1	0	10	1	0	50	139	25
YUV to HEVC	4k	1	0	10	1	0	47	140	26
YUV to AV1	4k	1	0	10	1	0	53	138	26
YUV to AVC	4k	4	0	10	0	0	60	166	26
YUV to HEVC	4k	4	0	10	0	0	54	169	28
YUV to AV1	4k	4	0	10	0	0	61	161	29
AVC to YUV	1080p	1	0	8	1	45	0	720	15
HEVC to YUV	1080p	1	0	8	1	53	0	759	9
VP9 to YUV	1080p	1	0	8	1	22	0	443	3
AVC to YUV	1080p	40	0	8	0	92	0	1510	0
HEVC to YUV	1080p	40	0	8	0	99	0	1540	0
VP9 to YUV	1080p	40	0	8	0	83	0	1543	0
YUV to AVC	1080p	1	0	8	1	0	58	648	16

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
YUV to HEVC	1080p	1	0	8	1	0	55	648	17
YUV to AV1	1080p	1	0	8	1	0	53	537	14
YUV to AVC	1080p	32	0	8	0	0	99	1152	2
YUV to HEVC	1080p	32	0	8	0	0	99	1221	2
YUV to AV1	1080p	32	0	8	0	0	99	1052	2
AVC to AVC	1080p	1	1	8	0	70	86	899	19
AVC to HEVC	1080p	1	1	8	0	70	83	953	19
AVC to AV1	1080p	1	1	8	0	56	84	805	19
HEVC to AVC	1080p	1	1	8	0	68	87	895	11
HEVC to HEVC	1080p	1	1	8	0	70	85	950	12
HEVC to AV1	1080p	1	1	8	0	58	85	809	12
VP9 to AVC	1080p	1	1	8	0	22	40	449	4
VP9 to HEVC	1080p	1	1	8	0	22	38	446	4
VP9 to AV1	1080p	1	1	8	0	22	44	445	6
AVC to AVC	1080p	32	1	8	0	76	99	892	1
AVC to HEVC	1080p	32	1	8	0	88	99	1003	1
AVC to AV1	1080p	32	1	8	0	76	99	960	1
HEVC to AVC	1080p	32	1	8	0	72	100	926	0
HEVC to HEVC	1080p	32	1	8	0	78	99	1036	0
HEVC to AV1	1080p	32	1	8	0	70	99	970	1
VP9 to AVC	1080p	32	1	8	0	75	99	1016	0
VP9 to HEVC	1080p	32	1	8	0	73	99	1134	0
VP9 to AV1	1080p	32	1	8	0	72	99	1021	0
AVC to YUV	1080p	1	0	10	1	29	0	508	6
HEVC to YUV	1080p	1	0	10	1	28	0	495	6
VP9 to YUV	1080p	1	0	10	1	22	0	453	3
AVC to YUV	1080p	40	0	10	0	64	0	1017	0
HEVC to YUV	1080p	40	0	10	0	76	0	1022	0
VP9 to YUV	1080p	40	0	10	0	81	0	1580	0
YUV to AVC	1080p	1	0	10	1	0	33	374	20
YUV to HEVC	1080p	1	0	10	1	0	31	372	19
YUV to AV1	1080p	1	0	10	1	0	33	330	18
YUV to AVC	1080p	32	0	10	0	0	69	770	5
YUV to HEVC	1080p	32	0	10	0	0	64	781	5
YUV to AV1	1080p	32	0	10	0	0	70	726	8
AVC to YUV	720p	1	0	8	1	42	0	1042	13
HEVC to YUV	720p	1	0	8	1	39	0	1116	7
VP9 to YUV	720p	1	0	8	1	29	0	845	4
AVC to YUV	720p	100	0	8	0	100	0	2519	0

TYPE	RES	JOBS	HW FRAME	Bit	Joint Mode	DEC_LOAD	ENC_LOAD	FPS	CPU
HEVC to YUV	720p	100	0	8	0	91	0	2626	0
VP9 to YUV	720p	100	0	8	0	95	0	2475	0
YUV to AVC	720p	1	0	8	1	0	35	879	12
YUV to HEVC	720p	1	0	8	1	0	35	887	12
YUV to AV1	720p	1	0	8	1	0	31	690	9
YUV to AVC	720p	64	0	8	0	0	100	2495	1
YUV to HEVC	720p	64	0	8	0	0	98	2541	1
YUV to AV1	720p	64	0	8	0	0	100	2064	1
AVC to AVC	720p	1	1	8	0	51	50	1241	18
AVC to HEVC	720p	1	1	8	0	50	49	1234	18
AVC to AV1	720p	1	1	8	0	43	49	1012	15
HEVC to AVC	720p	1	1	8	0	45	50	1257	11
HEVC to HEVC	720p	1	1	8	0	45	50	1255	12
HEVC to AV1	720p	1	1	8	0	39	48	1020	10
VP9 to AVC	720p	1	1	8	0	29	33	837	6
VP9 to HEVC	720p	1	1	8	0	29	33	832	6
VP9 to AV1	720p	1	1	8	0	29	39	836	7
AVC to AVC	720p	64	1	8	0	100	100	1975	0
AVC to HEVC	720p	64	1	8	0	100	100	2071	0
AVC to AV1	720p	64	1	8	0	96	100	1735	0
HEVC to AVC	720p	64	1	8	0	85	99	1984	0
HEVC to HEVC	720p	64	1	8	0	97	100	2065	0
HEVC to AV1	720p	64	1	8	0	90	100	1733	0
VP9 to AVC	720p	64	1	8	0	100	100	2219	0
VP9 to HEVC	720p	64	1	8	0	100	100	2280	0
VP9 to AV1	720p	64	1	8	0	95	97	1882	0

16.   T1U – FFmpeg Latency

16.1   Encoding

16.1.1 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.

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 -xcoder-params gopPresetIdx=9:lowDelay=1 -f
null -
```

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

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	65.86	66.7	65.27	0.05
YUV to HEVC	8k	1	63.12	68.98	61.43	1.19
YUV to AVC	4k	1	17.47	19.97	16.89	0.08
YUV to HEVC	4k	1	18.85	21.16	17.64	0.13
YUV to AV1	4k	1	24.62	29.13	18.3	0.85
YUV to AVC	4k	4	18.71	22.84	17.03	0.75
YUV to HEVC	4k	4	20.07	22.62	17.73	1.26
YUV to AV1	4k	4	24.94	29.29	18.36	0.75
YUV to AVC	4k	8	21.02	31.22	17.17	24.55
YUV to HEVC	4k	8	24.63	34.79	17.64	34.63
YUV to AV1	4k	8	42.84	62.09	30.46	9.88
YUV to AVC	1080p	1	5.23	6.2	5.05	0.02
YUV to HEVC	1080p	1	5.59	6.15	5.22	0.01
YUV to AV1	1080p	1	7.54	8.44	5.72	0.06
YUV to AVC	1080p	32	8.11	13.21	5.3	2.17
YUV to HEVC	1080p	32	9.77	14.38	6.41	2.29
YUV to AV1	1080p	32	47.17	76.5	40.77	2.29
YUV to AVC	720p	1	3.03	3.85	2.65	0.01
YUV to HEVC	720p	1	3.07	3.6	2.87	0.01
YUV to AV1	720p	1	4.19	4.98	3.36	0.05
YUV to AVC	720p	64	6.96	11.49	4.05	0.91
YUV to HEVC	720p	64	7.44	12.44	4.67	0.89
YUV to AV1	720p	64	45.76	90.86	37.43	5.72

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	89	0	113	4
HEVC to RGBA	8k	1	89	0	116	4
VP9 to RGBA	8k	1	22	0	31	1
AVC to RGBA	4k	1	20	0	107	6
AVC to RGBA	4k	16	94	2	462	1
HEVC to RGBA	4k	1	21	0	131	8
HEVC to RGBA	4k	16	92	3	548	2
VP9 to RGBA	4k	1	21	0	109	4
VP9 to RGBA	4k	16	95	2	460	1
AVC to RGBA	1080p	40	93	12	1564	1
HEVC to RGBA	1080p	40	93	11	1583	1
VP9 to RGBA	1080p	40	91	15	1994	0
AVC to RGBA	720p	100	93	17	2303	0
HEVC to RGBA	720p	100	84	18	2399	0
VP9 to RGBA	720p	64	88	17	2376	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:semiplanar0=1:enableOut1=1:semiplanar1=1:scale1=1280x720:enableO
ut2=1:semiplanar2=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	90	2	368	2
AVC to HEVC	8	28	90	2	392	2
AVC to AV1	8	21	87	1	325	2
HEVC to AVC	8	31	91	3	368	3
HEVC to HEVC	8	29	90	2	387	3
HEVC to AV1	8	22	88	1	333	2
VP9 to AVC	8	35	88	3	367	2
VP9 to HEVC	8	36	90	2	389	2
VP9 to AV1	8	29	88	2	328	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 and encoded by hardware encoder.

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

19.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

<num_jobs> == 1, multicoreJointMode = 1

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

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

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

19.2 RGBA Encoding Performance Results

TYPE	RES	JOBS	Joint Mode	ENC_LOAD	FPS	CPU
RGBA to AVC	4k	1	1	48	132	78
RGBA to HEVC	4k	1	1	45	134	81
RGBA to AV1	4k	1	1	52	133	79
RGBA to AVC	4k	4	0	60	147	42
RGBA to HEVC	4k	4	0	58	154	43
RGBA to AV1	4k	4	0	64	153	43
RGBA to AVC	4k	8	0	61	157	36
RGBA to HEVC	4k	8	0	60	160	32
RGBA to AV1	4k	8	0	67	165	42
RGBA to AVC	1080p	1	1	30	331	20
RGBA to HEVC	1080p	1	1	28	330	24
RGBA to AV1	1080p	1	1	31	318	25
RGBA to AVC	1080p	16	0	61	591	20
RGBA to HEVC	1080p	16	0	58	602	20
RGBA to AV1	1080p	16	0	63	593	18
RGBA to AVC	1080p	32	0	62	608	8
RGBA to HEVC	1080p	32	0	57	611	9
RGBA to AV1	1080p	32	0	60	640	8
RGBA to AVC	720p	1	1	22	498	24
RGBA to HEVC	720p	1	1	22	495	46
RGBA to AV1	720p	1	1	24	466	32
RGBA to AVC	720p	16	0	38	914	31
RGBA to HEVC	720p	16	0	37	919	33
RGBA to AV1	720p	16	0	43	895	29
RGBA to AVC	720p	32	0	35	889	20
RGBA to HEVC	720p	32	0	36	898	21
RGBA to AV1	720p	32	0	42	874	19

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	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	0	0	1	96	264	11
YUV to HEVC	4k	4	0	0	1	97	288	13
YUV to AV1	4k	4	0	0	1	97	248	12
YUV to AVC	4k	4	0	0	2	97	264	14
YUV to HEVC	4k	4	0	0	2	98	148	6
YUV to AV1	4k	4	0	0	2	97	122	6
YUV to AVC	4k	4	0	0	3	97	265	16
YUV to HEVC	4k	4	0	0	3	99	88	4
YUV to AV1	4k	4	0	0	3	96	64	3
YUV to AVC	4k	4	0	1	1	98	164	8
YUV to HEVC	4k	4	0	1	1	97	208	10
YUV to AVC	4k	4	0	1	2	98	164	8
YUV to HEVC	4k	4	0	1	2	98	96	6
YUV to AVC	4k	4	0	1	3	98	164	7
YUV to HEVC	4k	4	0	1	3	99	60	3
YUV to AVC	4k	4	4	0	1	100	172	8
YUV to HEVC	4k	4	4	0	1	99	200	12
YUV to AV1	4k	4	4	0	1	99	172	8
YUV to AVC	4k	4	4	0	2	99	172	9
YUV to HEVC	4k	4	4	0	2	99	120	5
YUV to AV1	4k	4	4	0	2	100	100	6
YUV to AVC	4k	4	4	0	3	99	172	9
YUV to HEVC	4k	4	4	0	3	100	76	3
YUV to AV1	4k	4	4	0	3	99	64	3
YUV to AVC	4k	4	4	1	1	99	120	5
YUV to HEVC	4k	4	4	1	1	100	156	8
YUV to AVC	4k	4	4	1	2	100	120	5
YUV to HEVC	4k	4	4	1	2	100	84	5
YUV to AVC	4k	4	4	1	3	99	120	6
YUV to HEVC	4k	4	4	1	3	100	56	3
YUV to AVC	4k	4	16	0	1	99	172	9
YUV to HEVC	4k	4	16	0	1	99	196	7
YUV to AV1	4k	4	16	0	1	99	172	9
YUV to AVC	4k	4	16	0	2	99	172	9
YUV to HEVC	4k	4	16	0	2	100	120	6
YUV to AV1	4k	4	16	0	2	100	100	5

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	4k	4	16	0	3	100	172	9
YUV to HEVC	4k	4	16	0	3	100	76	6
YUV to AV1	4k	4	16	0	3	99	64	3
YUV to AVC	4k	4	16	1	1	100	120	5
YUV to HEVC	4k	4	16	1	1	99	156	7
YUV to AVC	4k	4	16	1	2	100	120	5
YUV to HEVC	4k	4	16	1	2	100	81	4
YUV to AVC	4k	4	16	1	3	99	120	6
YUV to HEVC	4k	4	16	1	3	98	56	3
YUV to AVC	4k	4	40	0	1	99	169	9
YUV to HEVC	4k	4	40	0	1	99	196	12
YUV to AV1	4k	4	40	0	1	99	172	9
YUV to AVC	4k	4	40	0	2	99	170	8
YUV to HEVC	4k	4	40	0	2	100	116	5
YUV to AV1	4k	4	40	0	2	100	100	4
YUV to AVC	4k	4	40	0	3	99	171	8
YUV to HEVC	4k	4	40	0	3	99	76	4
YUV to AV1	4k	4	40	0	3	100	64	3
YUV to AVC	4k	4	40	1	1	99	120	6
YUV to HEVC	4k	4	40	1	1	99	154	7
YUV to AVC	4k	4	40	1	2	99	120	6
YUV to HEVC	4k	4	40	1	2	99	80	3
YUV to AVC	4k	4	40	1	3	100	120	6
YUV to HEVC	4k	4	40	1	3	100	56	2
YUV to AVC	1080p	20	0	0	1	99	1119	3
YUV to HEVC	1080p	20	0	0	1	99	1180	3
YUV to AV1	1080p	20	0	0	1	99	1020	3
YUV to AVC	1080p	20	0	0	2	99	1117	3
YUV to HEVC	1080p	20	0	0	2	99	600	2
YUV to AV1	1080p	20	0	0	2	99	480	1
YUV to AVC	1080p	20	0	0	3	99	1113	3
YUV to HEVC	1080p	20	0	0	3	100	360	1
YUV to AV1	1080p	20	0	0	3	99	260	1
YUV to AVC	1080p	20	0	1	1	99	660	2
YUV to HEVC	1080p	20	0	1	1	99	840	3
YUV to AVC	1080p	20	0	1	2	100	660	2
YUV to HEVC	1080p	20	0	1	2	100	380	1
YUV to AVC	1080p	20	0	1	3	99	660	2
YUV to HEVC	1080p	20	0	1	3	99	240	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	4	0	1	99	618	2
YUV to HEVC	1080p	20	4	0	1	99	719	2
YUV to AV1	1080p	20	4	0	1	99	622	2
YUV to AVC	1080p	20	4	0	2	99	615	2
YUV to HEVC	1080p	20	4	0	2	99	440	1
YUV to AV1	1080p	20	4	0	2	99	380	1
YUV to AVC	1080p	20	4	0	3	99	618	2
YUV to HEVC	1080p	20	4	0	3	99	300	1
YUV to AV1	1080p	20	4	0	3	100	240	1
YUV to AVC	1080p	20	4	1	1	99	440	1
YUV to HEVC	1080p	20	4	1	1	99	566	2
YUV to AVC	1080p	20	4	1	2	100	440	2
YUV to HEVC	1080p	20	4	1	2	100	320	1
YUV to AVC	1080p	20	4	1	3	99	440	1
YUV to HEVC	1080p	20	4	1	3	100	220	1
YUV to AVC	1080p	20	16	0	1	99	608	2
YUV to HEVC	1080p	20	16	0	1	99	717	2
YUV to AV1	1080p	20	16	0	1	99	621	2
YUV to AVC	1080p	20	16	0	2	99	604	2
YUV to HEVC	1080p	20	16	0	2	99	440	1
YUV to AV1	1080p	20	16	0	2	100	370	1
YUV to AVC	1080p	20	16	0	3	99	607	2
YUV to HEVC	1080p	20	16	0	3	99	300	1
YUV to AV1	1080p	20	16	0	3	99	240	1
YUV to AVC	1080p	20	16	1	1	99	440	1
YUV to HEVC	1080p	20	16	1	1	99	560	2
YUV to AVC	1080p	20	16	1	2	99	440	1
YUV to HEVC	1080p	20	16	1	2	100	320	1
YUV to AVC	1080p	20	16	1	3	99	440	1
YUV to HEVC	1080p	20	16	1	3	99	220	1
YUV to AVC	1080p	20	40	0	1	99	600	2
YUV to HEVC	1080p	20	40	0	1	99	701	2
YUV to AV1	1080p	20	40	0	1	99	620	2
YUV to AVC	1080p	20	40	0	2	99	600	2
YUV to HEVC	1080p	20	40	0	2	99	440	1
YUV to AV1	1080p	20	40	0	2	99	360	1
YUV to AVC	1080p	20	40	0	3	99	600	2
YUV to HEVC	1080p	20	40	0	3	99	300	1
YUV to AV1	1080p	20	40	0	3	100	232	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	1	99	440	2
YUV to HEVC	1080p	20	40	1	1	99	560	1
YUV to AVC	1080p	20	40	1	2	99	440	1
YUV to HEVC	1080p	20	40	1	2	99	300	1
YUV to AVC	1080p	20	40	1	3	99	440	1
YUV to HEVC	1080p	20	40	1	3	99	219	1
YUV to AVC	720p	40	0	0	1	94	2209	2
YUV to HEVC	720p	40	0	0	1	88	2202	2
YUV to AV1	720p	40	0	0	1	97	1942	2
YUV to AVC	720p	40	0	0	2	93	2222	2
YUV to HEVC	720p	40	0	0	2	99	1320	1
YUV to AV1	720p	40	0	0	2	99	1043	1
YUV to AVC	720p	40	0	0	3	94	2231	2
YUV to HEVC	720p	40	0	0	3	99	800	0
YUV to AV1	720p	40	0	0	3	99	560	0
YUV to AVC	720p	40	0	1	1	99	1480	1
YUV to HEVC	720p	40	0	1	1	99	1800	1
YUV to AVC	720p	40	0	1	2	99	1480	1
YUV to HEVC	720p	40	0	1	2	99	840	0
YUV to AVC	720p	40	0	1	3	99	1480	1
YUV to HEVC	720p	40	0	1	3	100	541	0
YUV to AVC	720p	40	4	0	1	99	1251	1
YUV to HEVC	720p	40	4	0	1	100	1262	1
YUV to AV1	720p	40	4	0	1	100	948	0
YUV to AVC	720p	40	4	0	2	99	1254	1
YUV to HEVC	720p	40	4	0	2	99	920	1
YUV to AV1	720p	40	4	0	2	99	760	0
YUV to AVC	720p	40	4	0	3	99	1248	1
YUV to HEVC	720p	40	4	0	3	99	640	0
YUV to AV1	720p	40	4	0	3	99	480	0
YUV to AVC	720p	40	4	1	1	99	920	1
YUV to HEVC	720p	40	4	1	1	99	1160	1
YUV to AVC	720p	40	4	1	2	99	920	1
YUV to HEVC	720p	40	4	1	2	99	678	0
YUV to AVC	720p	40	4	1	3	99	920	1
YUV to HEVC	720p	40	4	1	3	99	479	0
YUV to AVC	720p	40	16	0	1	99	1240	1
YUV to HEVC	720p	40	16	0	1	100	1280	1
YUV to AV1	720p	40	16	0	1	100	1001	0

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	ENC LOAD	FPS	CPU
YUV to AVC	720p	40	16	0	2	99	1240	1
YUV to HEVC	720p	40	16	0	2	99	920	1
YUV to AV1	720p	40	16	0	2	99	760	0
YUV to AVC	720p	40	16	0	3	99	1240	1
YUV to HEVC	720p	40	16	0	3	99	640	0
YUV to AV1	720p	40	16	0	3	99	480	1
YUV to AVC	720p	40	16	1	1	99	920	1
YUV to HEVC	720p	40	16	1	1	99	1160	1
YUV to AVC	720p	40	16	1	2	99	920	1
YUV to HEVC	720p	40	16	1	2	99	642	1
YUV to AVC	720p	40	16	1	3	99	920	1
YUV to HEVC	720p	40	16	1	3	99	441	0
YUV to AVC	720p	40	40	0	1	99	1238	1
YUV to HEVC	720p	40	40	0	1	100	1240	1
YUV to AV1	720p	40	40	0	1	100	962	1
YUV to AVC	720p	40	40	0	2	99	1238	1
YUV to HEVC	720p	40	40	0	2	99	920	1
YUV to AV1	720p	40	40	0	2	99	760	0
YUV to AVC	720p	40	40	0	3	99	1239	1
YUV to HEVC	720p	40	40	0	3	99	640	0
YUV to AV1	720p	40	40	0	3	99	480	0
YUV to AVC	720p	40	40	1	1	99	919	1
YUV to HEVC	720p	40	40	1	1	99	1150	1
YUV to AVC	720p	40	40	1	2	99	920	1
YUV to HEVC	720p	40	40	1	2	99	640	0
YUV to AVC	720p	40	40	1	3	99	919	1
YUV to HEVC	720p	40	40	1	3	100	440	0

21. T1U – Capped CRF

21.1 Encoding with lookaheadDepth

21.1.1 Description

YUV frame is read from an input file on ramdisk and fed into hardware encoder through PCIe.
YUV frame is encoded by hardware encoder with a mix of xcoder-params EnableRdoQuant,
rdoLevel, lookaheadDepth, CRF, bitrate, and vbvBufferSize.
Encoded bitstream is read out through PCIe and written into an output file.

21.1.2 Command line

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

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

<num_jobs> is the number of instances running concurrently

<resolution> is resolution of input

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

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

21.2   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	620	2
YUV to HEVC	1080p	20	0	0	1	19	99	718	2
YUV to AV1	1080p	20	0	0	1	19	99	623	2
YUV to AVC	1080p	20	0	0	2	19	99	620	2
YUV to HEVC	1080p	20	0	0	2	19	99	440	1
YUV to AV1	1080p	20	0	0	2	19	99	380	1
YUV to AVC	1080p	20	0	0	3	19	99	620	2
YUV to HEVC	1080p	20	0	0	3	19	100	300	1
YUV to AV1	1080p	20	0	0	3	19	100	240	1
YUV to AVC	1080p	20	0	1	1	19	99	440	1
YUV to HEVC	1080p	20	0	1	1	19	99	580	2
YUV to AVC	1080p	20	0	1	2	19	99	440	1
YUV to HEVC	1080p	20	0	1	2	19	100	320	1
YUV to AVC	1080p	20	0	1	3	19	99	440	1
YUV to HEVC	1080p	20	0	1	3	19	99	220	1
YUV to AVC	1080p	20	4	0	1	19	99	616	2
YUV to HEVC	1080p	20	4	0	1	19	99	720	2
YUV to AV1	1080p	20	4	0	1	19	99	620	2
YUV to AVC	1080p	20	4	0	2	19	99	614	2
YUV to HEVC	1080p	20	4	0	2	19	99	440	1
YUV to AV1	1080p	20	4	0	2	19	99	376	1
YUV to AVC	1080p	20	4	0	3	19	99	620	2
YUV to HEVC	1080p	20	4	0	3	19	99	300	1
YUV to AV1	1080p	20	4	0	3	19	100	240	1
YUV to AVC	1080p	20	4	1	1	19	100	440	1
YUV to HEVC	1080p	20	4	1	1	19	99	577	2
YUV to AVC	1080p	20	4	1	2	19	100	440	2
YUV to HEVC	1080p	20	4	1	2	19	99	320	1
YUV to AVC	1080p	20	4	1	3	19	99	440	1
YUV to HEVC	1080p	20	4	1	3	19	99	220	1
YUV to AVC	1080p	20	16	0	1	19	99	608	2
YUV to HEVC	1080p	20	16	0	1	19	99	715	2
YUV to AV1	1080p	20	16	0	1	19	99	620	2
YUV to AVC	1080p	20	16	0	2	19	99	606	2
YUV to HEVC	1080p	20	16	0	2	19	99	440	1
YUV to AV1	1080p	20	16	0	2	19	99	362	1
YUV to AVC	1080p	20	16	0	3	19	99	607	2
YUV to HEVC	1080p	20	16	0	3	19	99	300	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	3	19	99	240	1
YUV to AVC	1080p	20	16	1	1	19	99	440	1
YUV to HEVC	1080p	20	16	1	1	19	99	560	1
YUV to AVC	1080p	20	16	1	2	19	99	440	1
YUV to HEVC	1080p	20	16	1	2	19	99	320	1
YUV to AVC	1080p	20	16	1	3	19	99	440	1
YUV to HEVC	1080p	20	16	1	3	19	99	220	1
YUV to AVC	1080p	20	40	0	1	19	99	600	2
YUV to HEVC	1080p	20	40	0	1	19	99	700	3
YUV to AV1	1080p	20	40	0	1	19	99	620	2
YUV to AVC	1080p	20	40	0	2	19	99	600	2
YUV to HEVC	1080p	20	40	0	2	19	99	440	1
YUV to AV1	1080p	20	40	0	2	19	99	360	1
YUV to AVC	1080p	20	40	0	3	19	99	600	2
YUV to HEVC	1080p	20	40	0	3	19	99	294	1
YUV to AV1	1080p	20	40	0	3	19	100	233	1
YUV to AVC	1080p	20	40	1	1	19	100	440	1
YUV to HEVC	1080p	20	40	1	1	19	99	560	2
YUV to AVC	1080p	20	40	1	2	19	99	440	1
YUV to HEVC	1080p	20	40	1	2	19	99	300	1
YUV to AVC	1080p	20	40	1	3	19	99	440	1
YUV to HEVC	1080p	20	40	1	3	19	100	214	1
YUV to AVC	1080p	20	0	0	1	23	99	620	1
YUV to HEVC	1080p	20	0	0	1	23	99	719	2
YUV to AV1	1080p	20	0	0	1	23	99	630	1
YUV to AVC	1080p	20	0	0	2	23	99	619	2
YUV to HEVC	1080p	20	0	0	2	23	99	440	1
YUV to AV1	1080p	20	0	0	2	23	99	380	1
YUV to AVC	1080p	20	0	0	3	23	99	620	2
YUV to HEVC	1080p	20	0	0	3	23	99	300	1
YUV to AV1	1080p	20	0	0	3	23	100	240	1
YUV to AVC	1080p	20	0	1	1	23	99	440	1
YUV to HEVC	1080p	20	0	1	1	23	99	575	1
YUV to AVC	1080p	20	0	1	2	23	99	440	1
YUV to HEVC	1080p	20	0	1	2	23	99	320	1
YUV to AVC	1080p	20	0	1	3	23	99	440	1
YUV to HEVC	1080p	20	0	1	3	23	99	220	1
YUV to AVC	1080p	20	4	0	1	23	99	616	2
YUV to HEVC	1080p	20	4	0	1	23	99	720	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	4	0	1	23	99	622	2
YUV to AVC	1080p	20	4	0	2	23	99	618	2
YUV to HEVC	1080p	20	4	0	2	23	99	440	1
YUV to AV1	1080p	20	4	0	2	23	99	377	1
YUV to AVC	1080p	20	4	0	3	23	99	614	2
YUV to HEVC	1080p	20	4	0	3	23	100	300	1
YUV to AV1	1080p	20	4	0	3	23	99	240	1
YUV to AVC	1080p	20	4	1	1	23	99	440	1
YUV to HEVC	1080p	20	4	1	1	23	99	572	2
YUV to AVC	1080p	20	4	1	2	23	99	440	1
YUV to HEVC	1080p	20	4	1	2	23	99	320	1
YUV to AVC	1080p	20	4	1	3	23	99	440	1
YUV to HEVC	1080p	20	4	1	3	23	99	220	1
YUV to AVC	1080p	20	16	0	1	23	99	608	2
YUV to HEVC	1080p	20	16	0	1	23	99	713	2
YUV to AV1	1080p	20	16	0	1	23	99	620	2
YUV to AVC	1080p	20	16	0	2	23	99	606	2
YUV to HEVC	1080p	20	16	0	2	23	99	440	1
YUV to AV1	1080p	20	16	0	2	23	99	363	1
YUV to AVC	1080p	20	16	0	3	23	99	604	2
YUV to HEVC	1080p	20	16	0	3	23	99	300	1
YUV to AV1	1080p	20	16	0	3	23	100	239	1
YUV to AVC	1080p	20	16	1	1	23	99	440	1
YUV to HEVC	1080p	20	16	1	1	23	99	560	2
YUV to AVC	1080p	20	16	1	2	23	99	440	1
YUV to HEVC	1080p	20	16	1	2	23	99	320	1
YUV to AVC	1080p	20	16	1	3	23	99	440	1
YUV to HEVC	1080p	20	16	1	3	23	99	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	618	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	1
YUV to AVC	1080p	20	40	0	3	23	99	600	2
YUV to HEVC	1080p	20	40	0	3	23	100	295	1
YUV to AV1	1080p	20	40	0	3	23	99	233	1
YUV to AVC	1080p	20	40	1	1	23	99	440	1
YUV to HEVC	1080p	20	40	1	1	23	99	560	2

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AVC	1080p	20	40	1	2	23	100	440	1
YUV to HEVC	1080p	20	40	1	2	23	99	300	1
YUV to AVC	1080p	20	40	1	3	23	99	440	1
YUV to HEVC	1080p	20	40	1	3	23	100	210	1
YUV to AVC	1080p	20	0	0	1	27	99	620	2
YUV to HEVC	1080p	20	0	0	1	27	99	720	3
YUV to AV1	1080p	20	0	0	1	27	99	631	2
YUV to AVC	1080p	20	0	0	2	27	99	620	2
YUV to HEVC	1080p	20	0	0	2	27	99	440	1
YUV to AV1	1080p	20	0	0	2	27	99	380	1
YUV to AVC	1080p	20	0	0	3	27	99	619	2
YUV to HEVC	1080p	20	0	0	3	27	99	300	1
YUV to AV1	1080p	20	0	0	3	27	99	240	1
YUV to AVC	1080p	20	0	1	1	27	99	440	1
YUV to HEVC	1080p	20	0	1	1	27	99	579	2
YUV to AVC	1080p	20	0	1	2	27	100	440	1
YUV to HEVC	1080p	20	0	1	2	27	100	320	1
YUV to AVC	1080p	20	0	1	3	27	100	440	1
YUV to HEVC	1080p	20	0	1	3	27	99	220	1
YUV to AVC	1080p	20	4	0	1	27	99	615	2
YUV to HEVC	1080p	20	4	0	1	27	99	719	2
YUV to AV1	1080p	20	4	0	1	27	99	620	2
YUV to AVC	1080p	20	4	0	2	27	99	619	2
YUV to HEVC	1080p	20	4	0	2	27	99	440	1
YUV to AV1	1080p	20	4	0	2	27	99	377	1
YUV to AVC	1080p	20	4	0	3	27	99	613	2
YUV to HEVC	1080p	20	4	0	3	27	99	300	1
YUV to AV1	1080p	20	4	0	3	27	100	240	1
YUV to AVC	1080p	20	4	1	1	27	99	440	1
YUV to HEVC	1080p	20	4	1	1	27	99	574	2
YUV to AVC	1080p	20	4	1	2	27	99	440	1
YUV to HEVC	1080p	20	4	1	2	27	100	320	1
YUV to AVC	1080p	20	4	1	3	27	99	440	1
YUV to HEVC	1080p	20	4	1	3	27	99	220	1
YUV to AVC	1080p	20	16	0	1	27	99	606	2
YUV to HEVC	1080p	20	16	0	1	27	99	714	2
YUV to AV1	1080p	20	16	0	1	27	99	620	2
YUV to AVC	1080p	20	16	0	2	27	99	605	2
YUV to HEVC	1080p	20	16	0	2	27	99	440	1

TYPE	RES	JOBS	lookaheadDepth	enableRdoQuant	rdoLevel	CRF	ENC LOAD	FPS	CPU
YUV to AV1	1080p	20	16	0	2	27	99	361	1
YUV to AVC	1080p	20	16	0	3	27	99	606	2
YUV to HEVC	1080p	20	16	0	3	27	100	300	1
YUV to AV1	1080p	20	16	0	3	27	99	239	1
YUV to AVC	1080p	20	16	1	1	27	100	440	1
YUV to HEVC	1080p	20	16	1	1	27	99	560	2
YUV to AVC	1080p	20	16	1	2	27	99	440	1
YUV to HEVC	1080p	20	16	1	2	27	100	320	1
YUV to AVC	1080p	20	16	1	3	27	100	440	1
YUV to HEVC	1080p	20	16	1	3	27	100	220	1
YUV to AVC	1080p	20	40	0	1	27	99	600	2
YUV to HEVC	1080p	20	40	0	1	27	99	700	2
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	1
YUV to AV1	1080p	20	40	0	2	27	99	360	1
YUV to AVC	1080p	20	40	0	3	27	99	600	2
YUV to HEVC	1080p	20	40	0	3	27	99	296	1
YUV to AV1	1080p	20	40	0	3	27	99	233	1
YUV to AVC	1080p	20	40	1	1	27	99	440	1
YUV to HEVC	1080p	20	40	1	1	27	99	560	2
YUV to AVC	1080p	20	40	1	2	27	99	440	1
YUV to HEVC	1080p	20	40	1	2	27	100	300	1
YUV to AVC	1080p	20	40	1	3	27	100	440	1
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.rgba -filter_complex
"[1:v]format=rgba,ni_quadra_hwupload=0[a];[0:v][a]ni_quadra_overlay=0:0
:alpha=1:inplace=1[b]" -c:a copy -map "[b]" -c:v <enc>_ni_quadra_enc -
enc 0 -xcoder-params "RcEnable=1:bitrate=2000000" -f null -
```

<dec> is the decoder codec. ie h264_ni_quadra_dec, h265_ni_quadra_dec, vp9_ni_quadra_dec
<enc> is the encoder codec. ie h264_ni_quadra_enc, h265_ni_quadra_enc, av1_ni_quadra_enc
Input Video: 1080p
Input Image: 128x128

22.2   Inplace Overlay Performance Results

TYPE	JOBS	FPS	CPU	DEC_LOAD	ENC_LOAD	SCALER_LOAD
AVC to AVC	1	239	8	13	19	4
AVC to HEVC	1	254	8	14	19	4
AVC to AV1	1	219	7	12	19	3
HEVC to AVC	1	239	11	13	19	4
HEVC to HEVC	1	252	11	14	19	4
HEVC to AV1	1	219	11	12	19	3
VP9 to AVC	1	240	7	17	19	4
VP9 to HEVC	1	255	8	18	19	4
VP9 to AV1	1	219	7	16	19	3
AVC to AVC	16	923	2	70	91	23
AVC to HEVC	16	1024	2	75	89	25
AVC to AV1	16	947	2	66	89	23
HEVC to AVC	16	943	3	68	89	22
HEVC to HEVC	16	1040	4	74	89	25
HEVC to AV1	16	960	3	65	90	23
VP9 to AVC	16	929	2	83	89	22
VP9 to HEVC	16	997	2	85	83	23
VP9 to AV1	16	956	2	82	90	23
AVC to AVC	32	864	1	73	92	23
AVC to HEVC	32	961	1	80	91	26
AVC to AV1	32	928	1	74	91	24
HEVC to AVC	32	896	1	73	93	24
HEVC to HEVC	32	1010	2	78	92	26
HEVC to AV1	32	960	1	70	91	24
VP9 to AVC	32	902	1	86	93	23
VP9 to HEVC	32	991	1	89	88	25
VP9 to AV1	32	960	1	84	91	23

Appendix A: GStreamer XStack Command

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

```
gst-launch-1.0 multifilesrc location=/media/ramdisk/input.h265 loop=true !
h265parse ! niquadrah265dec  xcoder-params='out=hw' ! niquadrastack
inputs=32
layout="0_0|w0_0|w0+w1_0|w0+w1+w2_0|0_h0|w0_h0|w0+w1_h0|w0+w1+w2_h0|0_h0+h1|w
0_h0+h1|w0+w1_h0+h1|w0+w1+w2_h0+h1|0_h0+h1+h2|w0_h0+h1+h2|w0+w1_h0+h1+h2|w0+w
1+w2_h0+h1+h2|0_h0+h1+h2+h3|w0_h0+h1+h2+h3|w0+w1_h0+h1+h2+h3|w0+w1+w2_h0+h1+h
2+h3|0_h0+h1+h2+h3|w0_h0+h1+h2+h3|w0+w1_h0+h1+h2+h3|w0+w1+w2_h0+h1+h2+h3|0_h0
+h1+h2+h3+h4|w0_h0+h1+h2+h3+h4|w0+w1_h0+h1+h2+h3+h4|w0+w1+w2_h0+h1+h2+h3+h4|0
_h0+h1+h2+h3+h4+h5|w0_h0+h1+h2+h3+h4+h5|w0+w1_h0+h1+h2+h3+h4+h5|w0+w1+w2_h0+h
1+h2+h3+h4+h5|0_h0+h1+h2+h3+h4+h5+h6|w0_h0+h1+h2+h3+h4+h5+h6|w0+w1_h0+h1+h2+h
3+h4+h5+h6|w0+w1+w2_h0+h1+h2+h3+h4+h5+h6"
size="480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135
|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_
135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|480_135|4
80_135|480_135|480_135|480_135" shortest=1 name=xstack ! niquadrah264enc  !
fpsdisplaysink video-sink=fakesink sync=false -v   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. 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. multifilesrc
location=/media/ramdisk/input.h265 loop=true ! h265parse ! niquadrah265dec
xcoder-params='out=hw' ! xstack.
```

Appendix B: 7x7 Grid Layout

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

```
| 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 276x154 |

| 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 276x154 |

| 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 276x154 |

| 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 276x154 |

| 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 276x154 |

| 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 274x154 | 276x154 |

| 274x156 | 274x156 | 274x156 | 274x156 | 274x156 | 274x156 | 276x156 |
```

Appendix C: GStreamer Ladder Command

Example of single input with 64 outputs

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

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