

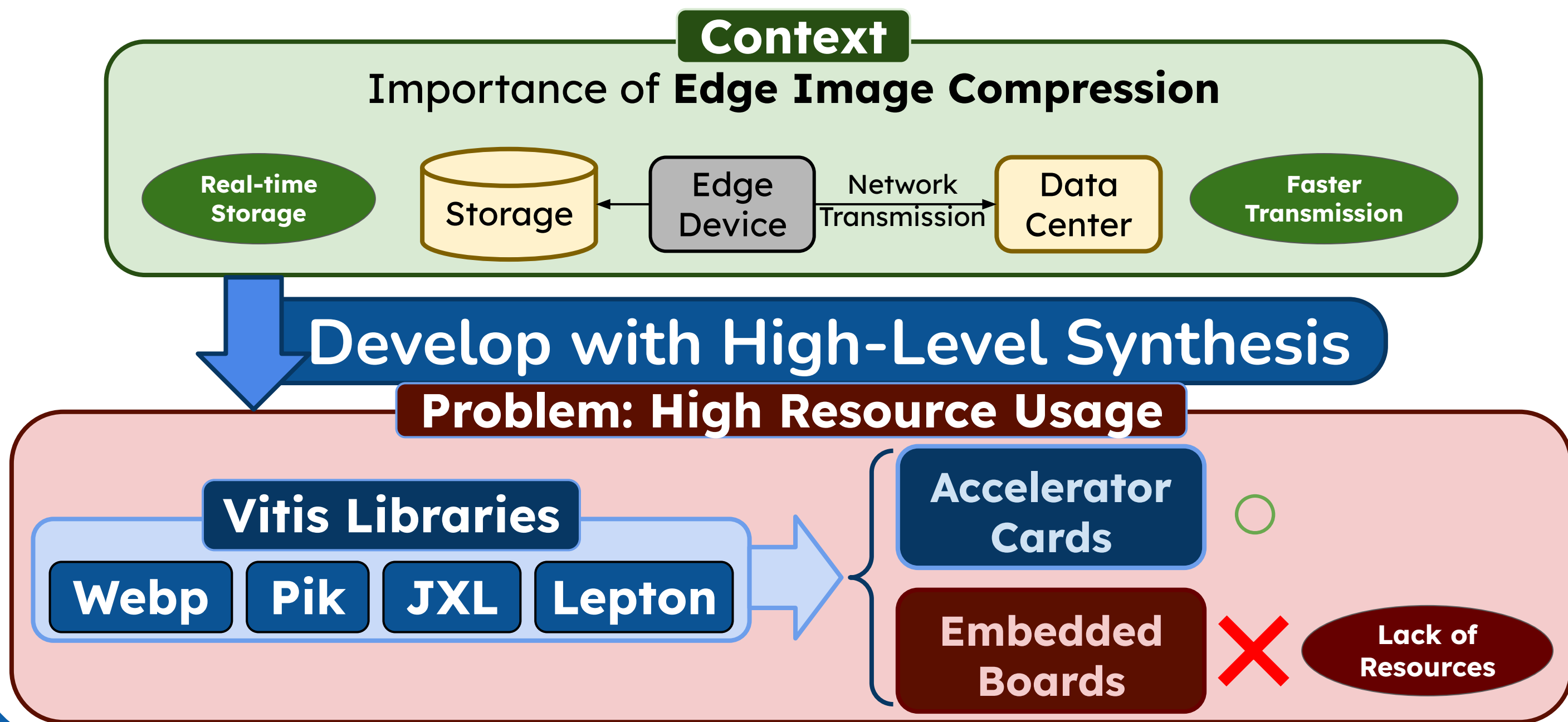
GStreamer-integrated HLS-based JPEG Encoder for Edge FPGA SoCs

Yuri Guimaraes Pereira Primo da Silva¹, Shinya Honda¹,
Sugako Otani¹², Masato Eda¹, Abraham Monrroy Cano³

¹Nagoya University, ²Renesas Electronics,
³Map IV Inc.

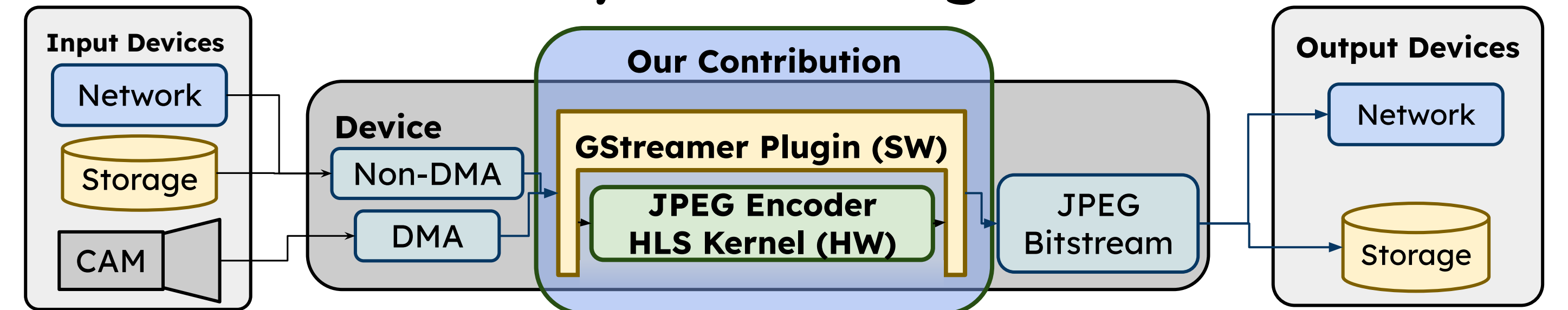
E-mail: yuri_gpps1@ert1.jp

Motivation



Approach

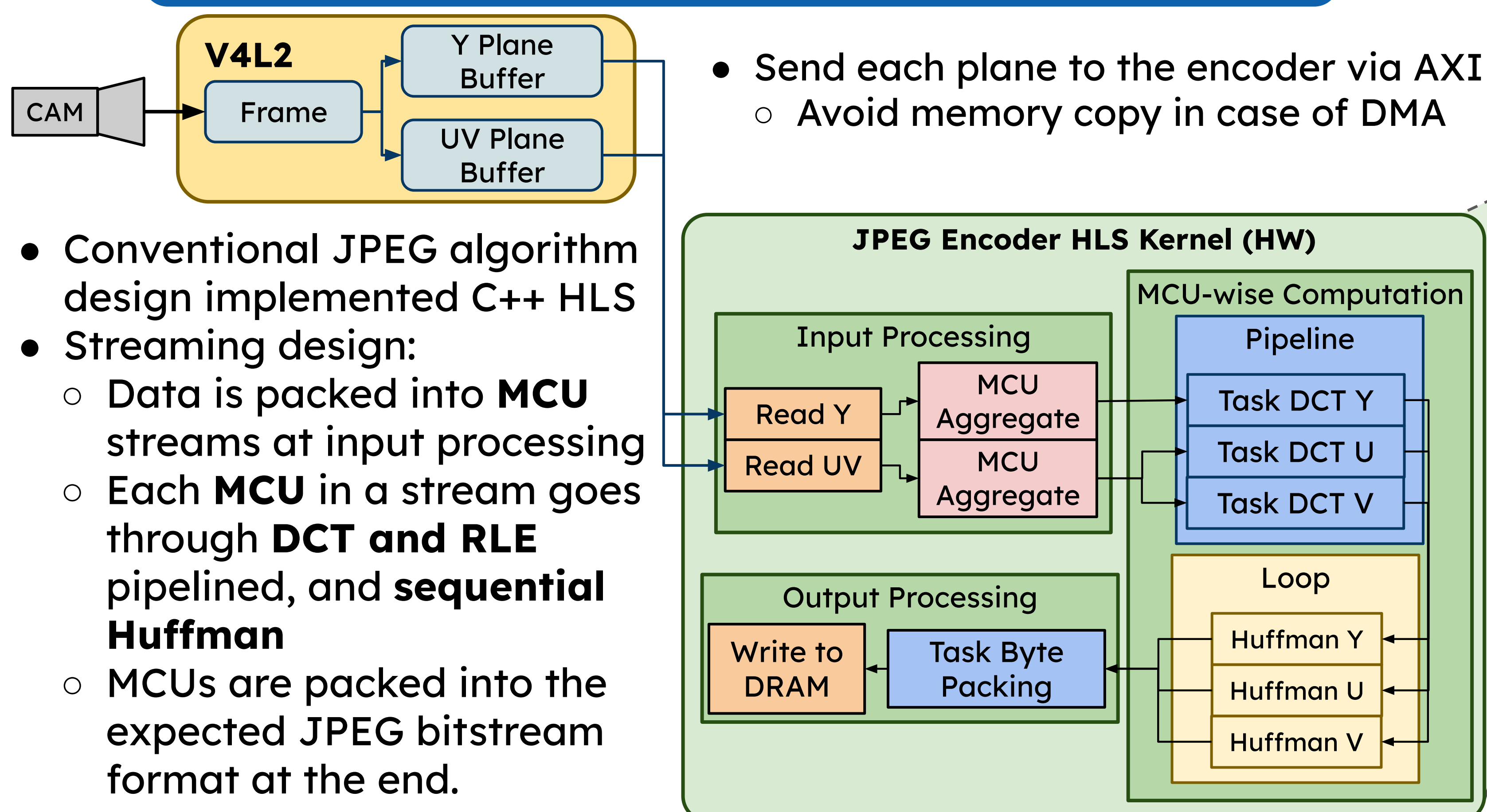
System Design



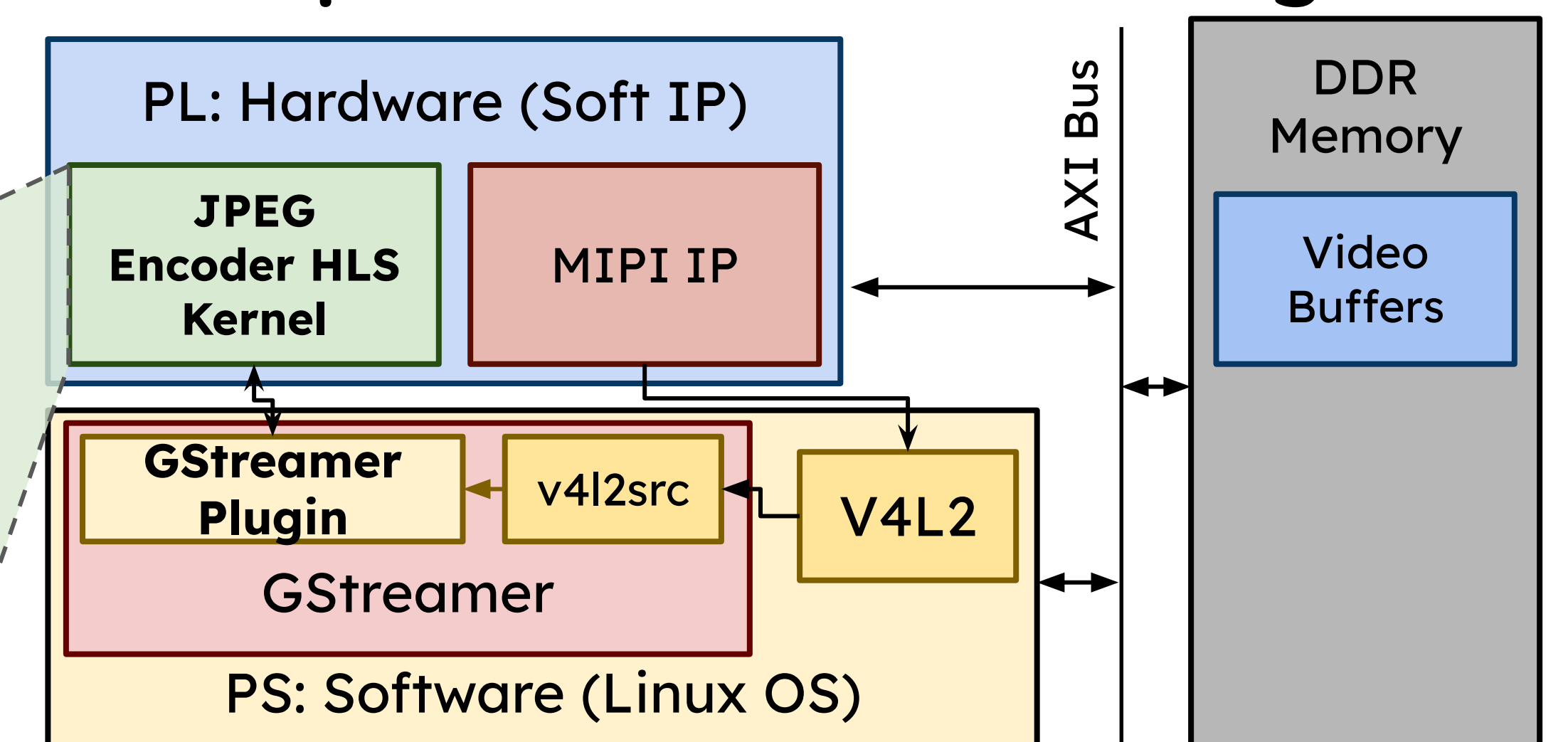
Contributions

- **HW JPEG Encoder** implemented in *High-level Synthesis*.
- **GStreamer plugin** interfaces with encoder.
 - Allows for mix and matching of pipeline elements for arbitrary media processing.
 - Easy usage of DMA capabilities.

Hardware Architecture



Top-level Hardware Design

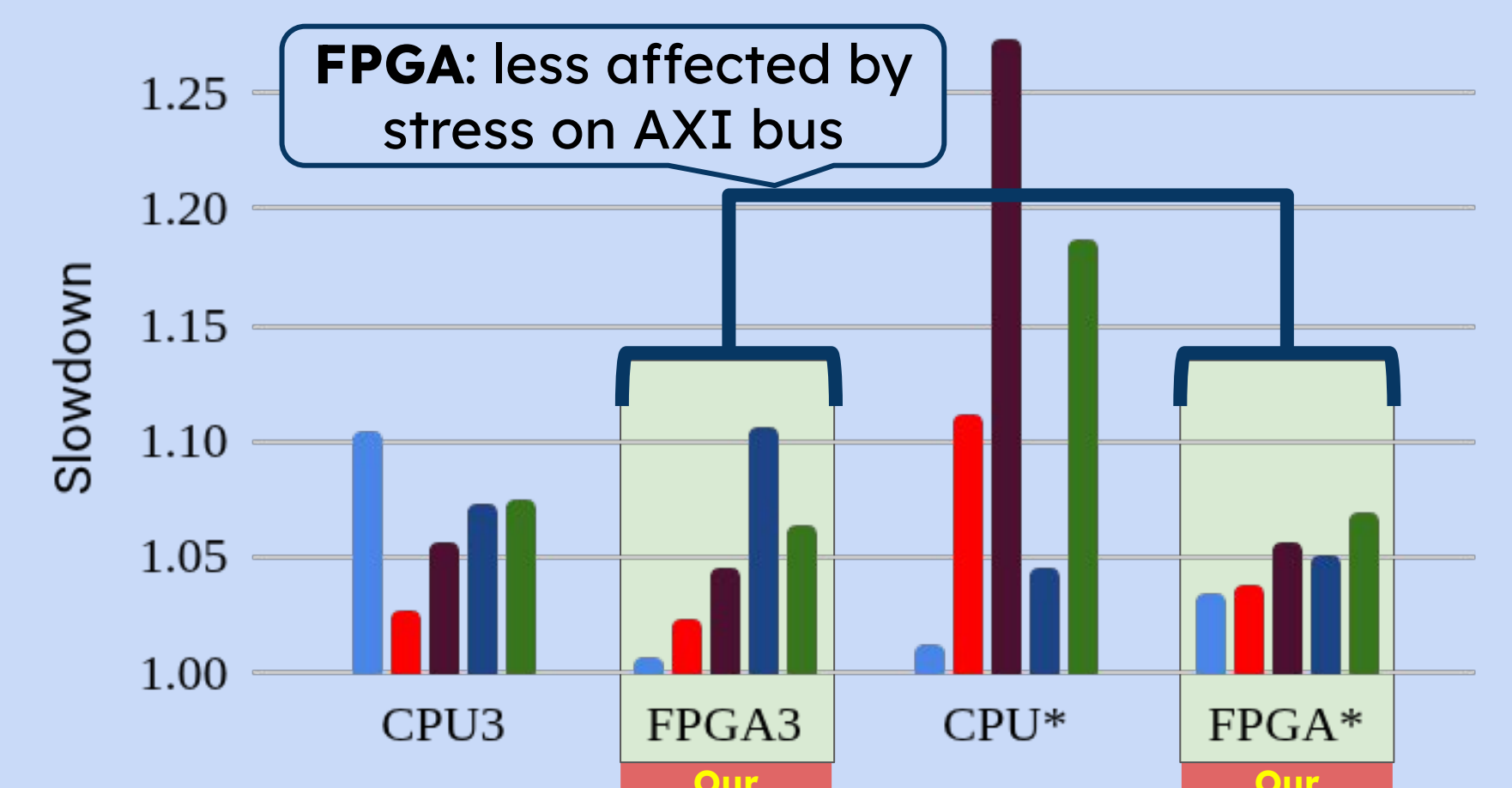
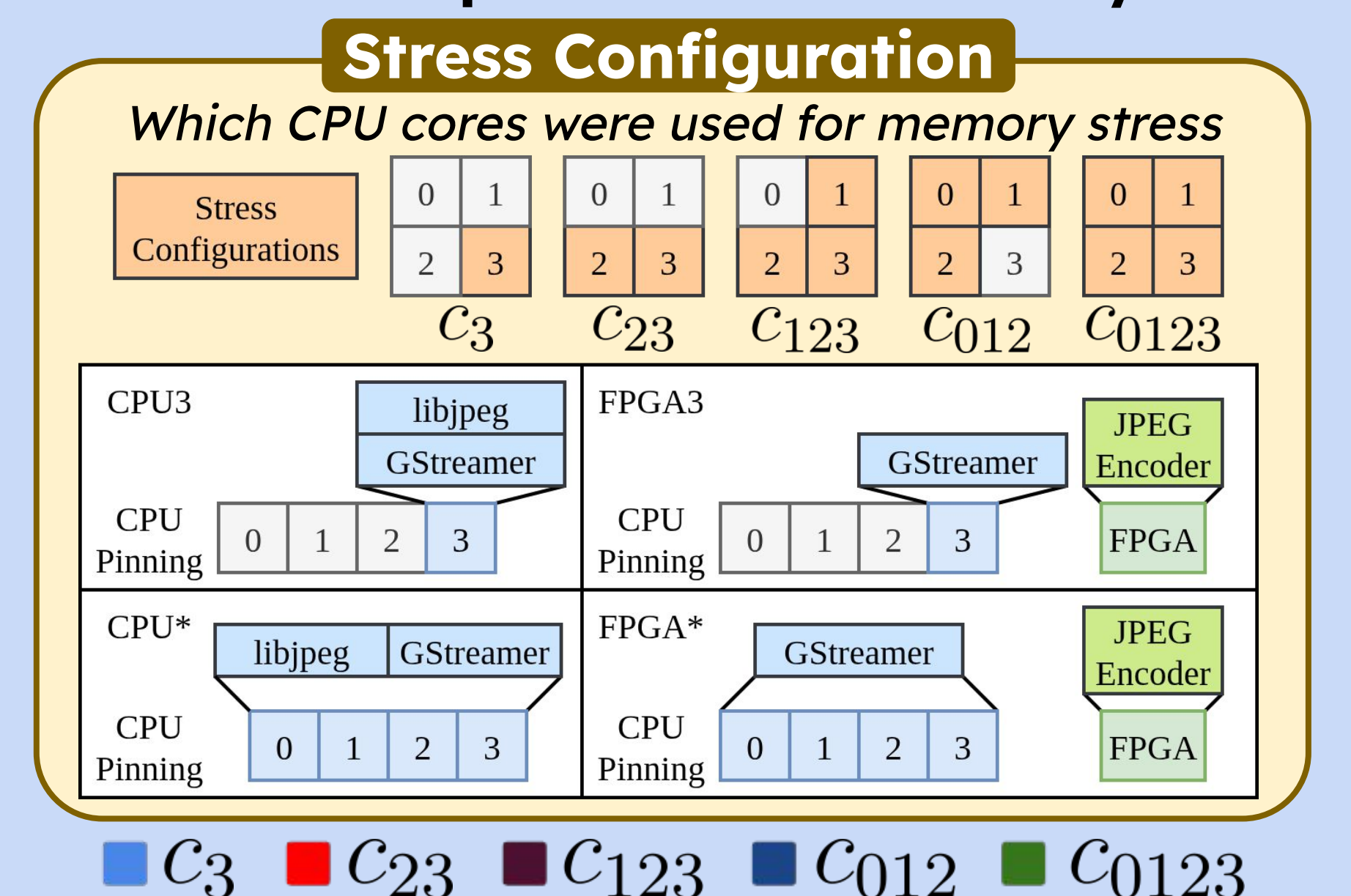


- Camera accessed through V4L2
 - Accesses must go through AXI bus

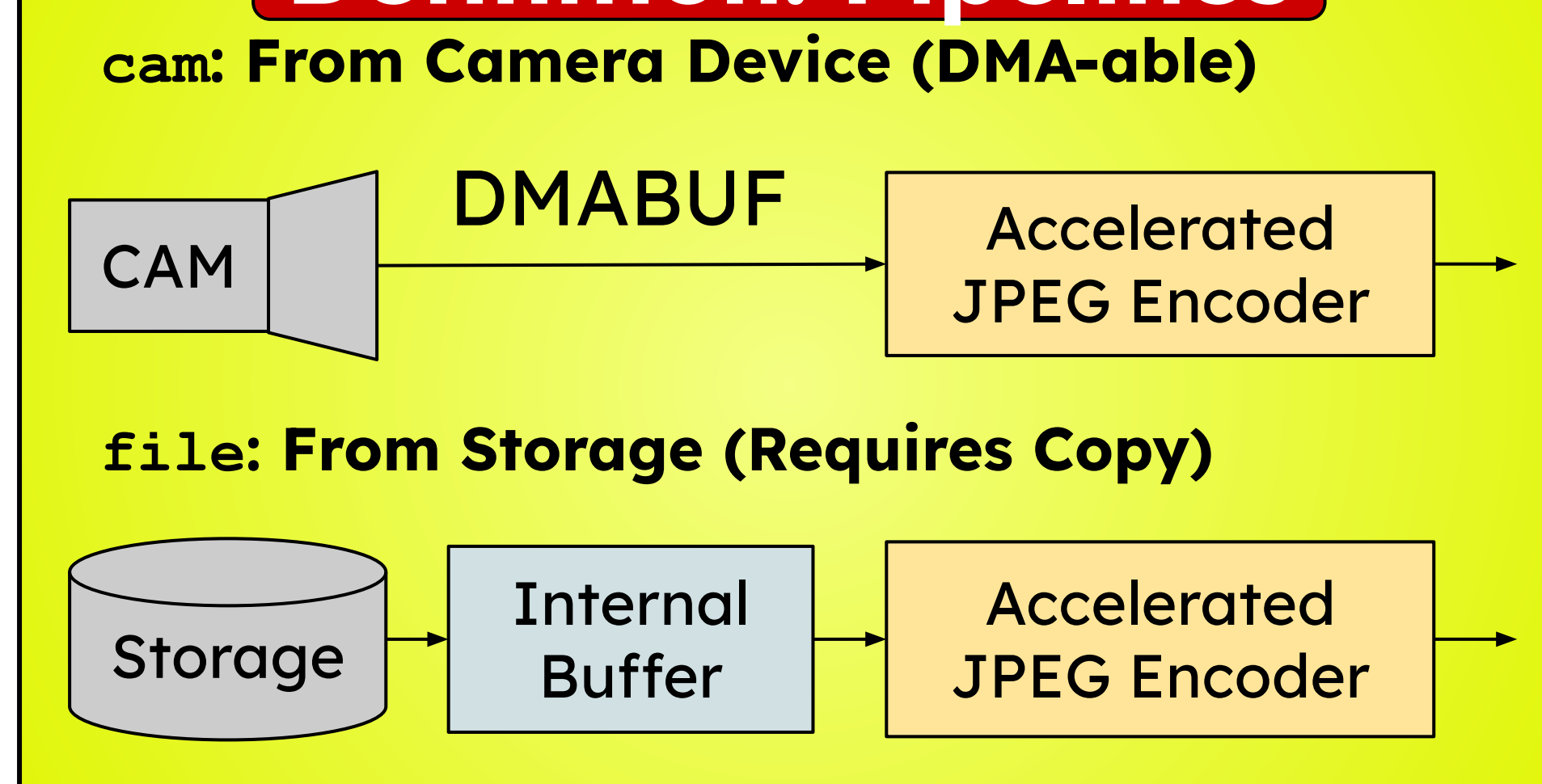
Experiments

Memory Stress Testing

Slowdown Comparison under Memory Stress



Definition: Pipelines



Performance

FPGA vs. CPU (jpegenc)

		Our		
		CPU	FPGA	FPGA/CPU
cam	FPS	2.6	19.5	7.5
	CPU %	93	5	0.054
file	FPS	22.6	21.7	.96
	CPU %	107	14	0.13

Resource Utilization

JPEG (our) vs. Vitis Codec Libraries

	Freq.	BRAM	URAM	DSP	FF	LUT
Our JPEG ¹	250MHz	6	12	85	18150	13868
Webp ²	100MHz	81	46	834	71227	68906
Webp ²	250MHz	229	10	414	92030	68755
JXL ²	260MHz	584	122	644	270565	199537
PIK ²	200MHz	614	473	2398	549987	449995

¹Our work

²Vitis Codec Libraries

Webp Encoder²

- **High Resource Usage**
 - Requires lowering target frequency to 100Hz

JPEG Encoder¹

- **7~25% of Webp Resources**
 - Can fit **four** JPEG encoders in place of a **single** Webp Encoder

Discussion and Conclusions

1. Suitability for Edge
 - Low resource usage
 - Realtime performance
2. Usability
 - GStreamer integration
 - Easy composition of media pipelines
3. Increased Predictability
 - Resilience against memory stress

Download this Poster!

