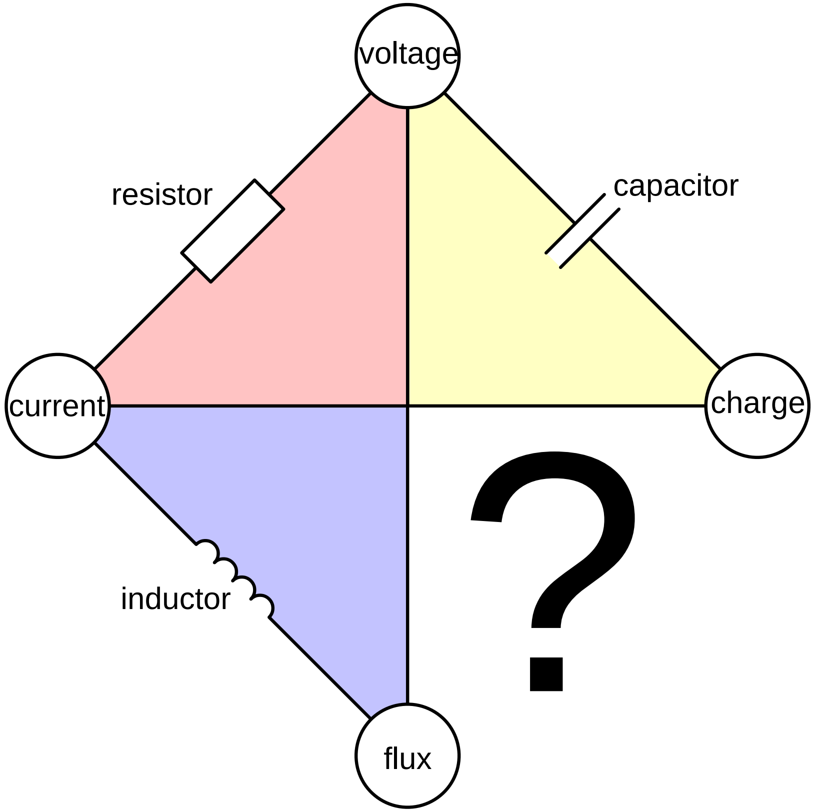
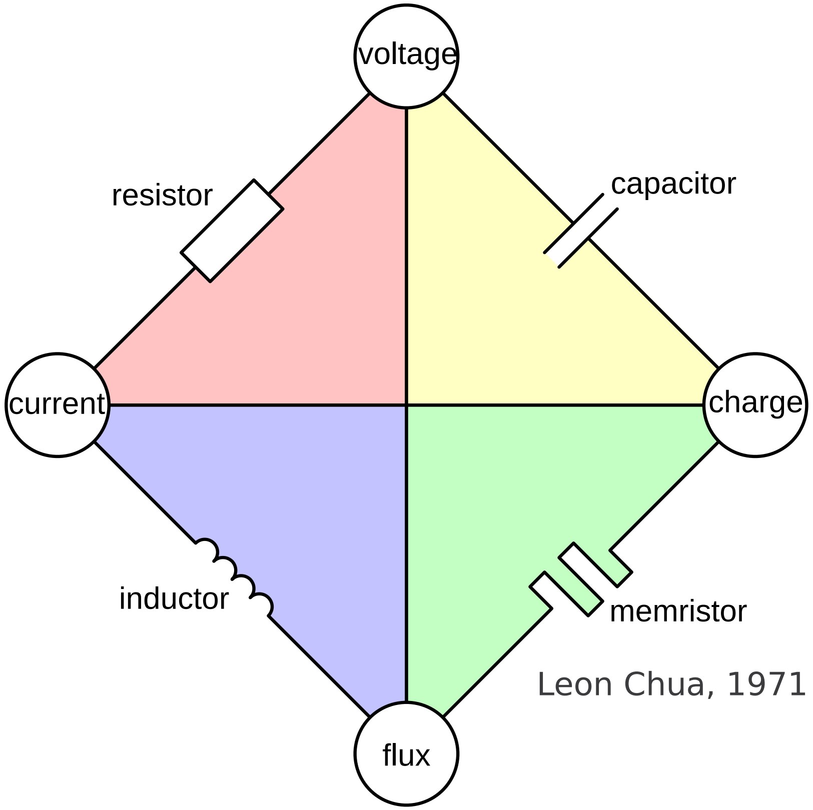
Stateless encoding in V4L2

Andrzej Pietrasiewicz
andrzej.p@collabora.com

Missing element?

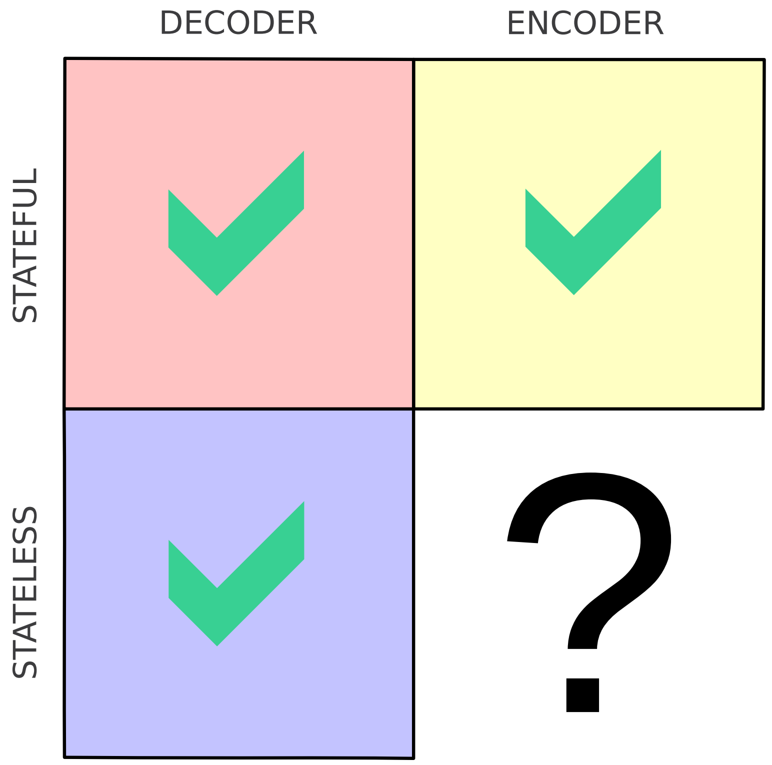


Memristor



Leon Chua, 1971

V4L2 codecs



V4L2 codecs

	DECODER	ENCODER
STATEFUL	✓	✓
STATELESS	✓	✓

Stateless encoders are coming to Linux

Agenda

- Definitions
- uAPI
- Rate control
- Future
- Q&A

COLLABORA

Definitions

General

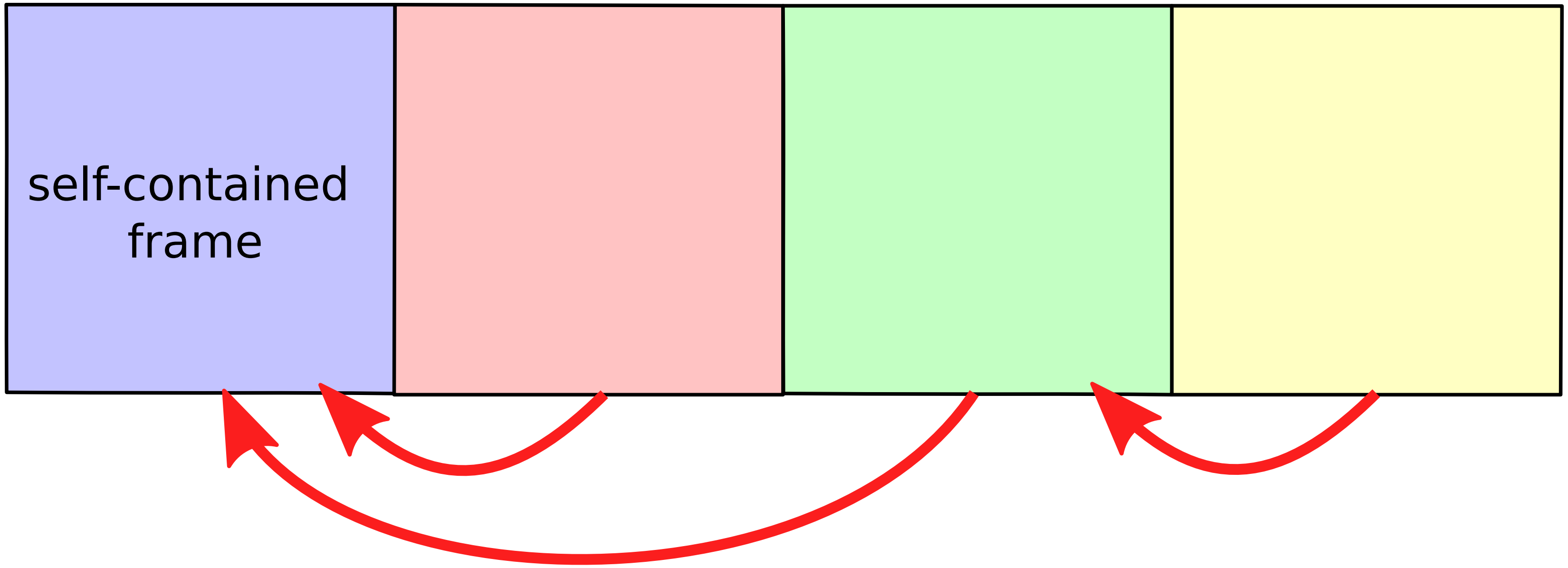
- CoDec
- mem2mem

Stateful vs Stateless

Stateful	Stateless
State kept and maintained **in** hardware	State kept and maintained **outside** hardware

So what?

Reference frames



Stateful vs Stateless

Stateful	Stateless
More complex hw	**Less complex hw**
Sw needs to interact with codec firmware	More registers to cope with
More expensive context change	**Less expensive context change**
Less flexibility	**More flexibility**

COLLABORA

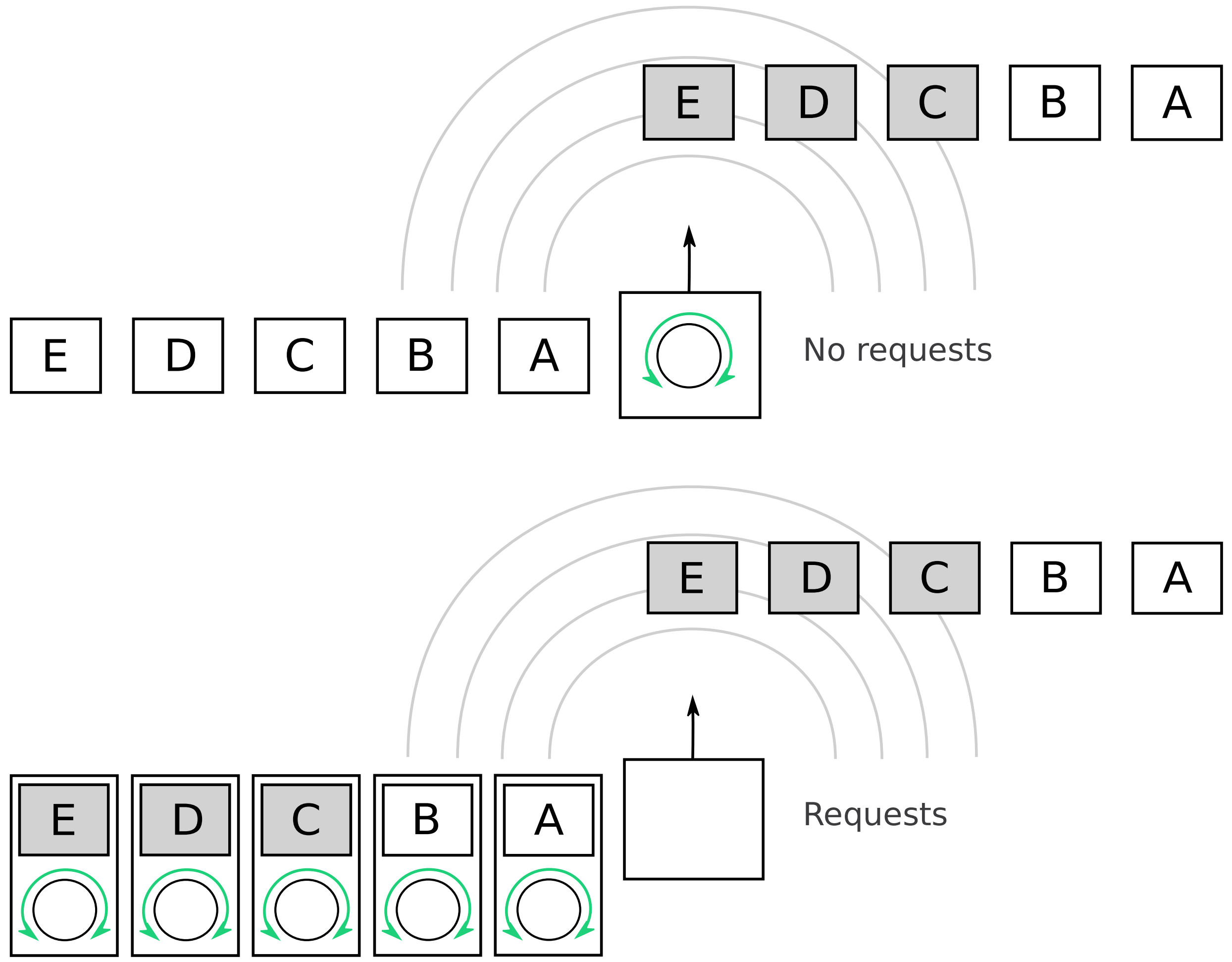
uAPI

First encoders

- 2020: H.264
  - https://github.com/bootlin/linux/tree/hantro/h264-encoding-v5.11
  - https://github.com/bootlin/v4l2-hantro-h264-encoder

- 2023: VP8
  - https://lore.kernel.org/linux-arm-kernel/20230309125651.23911-1-andrzej.p@collabora.com/T/
  - https://gitlab.collabora.com/linux/for-upstream/-/tree/vp8-rfc-v6.4-rc6
  - https://gitlab.freedesktop.org/gstreamer/gstreamer/-/merge_requests/3736

Request API



Using requests

```
ioctl(media_fd, MEDIA_IOC_REQUEST_ALLOC, &request_fd);

struct v4l2_ext_controls ext_ctrls;
struct v4l2_ext_control ctrls[1];
struct v4l2_ctrl_vp8_encode_params params;

ext_ctrls.controls = ctrls;
ext_ctrls.count = 1;

ctrls[0].id = V4L2_CID_STATELESS_VP8_ENCODE_PARAMS;
ctrls[0].ptr = &params;
ctrls[0].size = sizeof(params);

ext_ctrls.which = V4L2_CTRL_WHICH_REQUEST_VAL;
ext_ctrls.request_fd = request_fd;

ioctl(video_fd, VIDIOC_S_EXT_CTRLS, &ext_ctrls);

ioctl(request_fd, MEDIA_REQUEST_IOC_QUEUE, NULL);
```

VP8 encode params

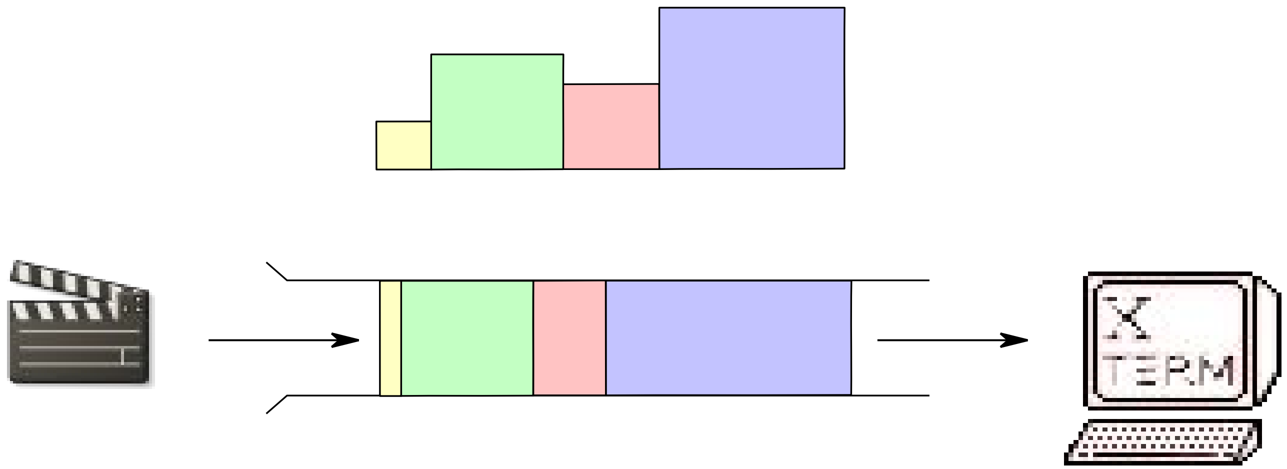
```
struct v4l2_ctrl_vp8_encode_params {
    __u32 flags;
    __u8 frame_type;
    __u8 color_space;
    __u8 clamping_type;
    __u8 loop_filter_type;
    __u8 loop_filter_level;
    __u8 sharpness_level;
    __u8 log2_nbr_of_dct_partitions;
    __u8 prob_intra;
    __u8 prob_last;
    __u8 prob_gf;
    __u8 copy_buffer_to_golden;
    __u8 copy_buffer_to_alternate;
    __u8 reference_type;
};
```

* Subject to change

COLLABORA

Rate control

Rate control



Rate control strategies

- Constant QP
- Constant Bitrate
- Average Bitrate
- ?

VP8 QP

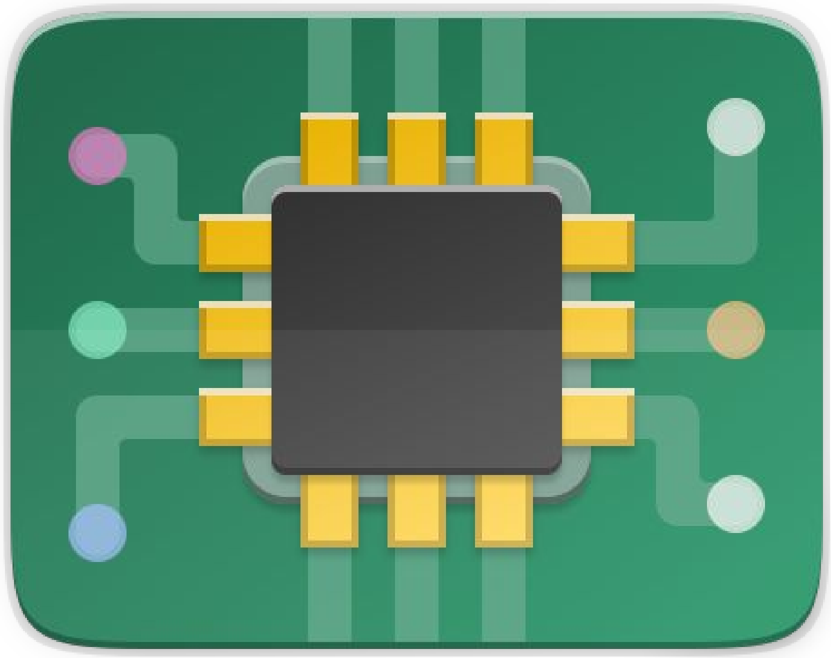
```
struct v4l2_ext_controls ext_ctrls;
struct v4l2_ext_control ctrls[1];

ext_ctrls.controls = ctrls;
ext_ctrls.count = 1;

ctrls[0].id = V4L2_CID_STATELESS_VP8_ENCODE_QP;
ctrls[0].value = fixed_qp;

ext_ctrls.which = V4L2_CTRL_WHICH_REQUEST_VAL;
ext_ctrls.request_fd = request_fd;

ioctl(video_fd, VIDIOC_S_EXT_CTRLS, &ext_ctrls);
```

* Subject to change

COLLABORA

Future

More VP8 hw?



Media Summit 26th Jun 2023

- 2 drivers rule?
- Frame header assembled in the kernel
- uAPI should support 3 reference frames
- Rate control:
  - Per-frame constant QP
  - Maybe have in-kernel algorithm for all encoders to use
  - Maybe support hardware-assisted rate control

AI?

- Improve encoders
  - Reference frame selection
  - Rate control

- Eliminate encoders 😱
  - Make guesses

Thank you!

We are hiring

col.la/careers