MIPI CCS camera sensor driver

The MIPI CCS camera sensor driver is a generic driver for MIPI CCS compliant camera sensors. It exposes three sub-devices representing the pixel array, the binner and the scaler.

As the capabilities of individual devices vary, the driver exposes interfaces based on the capabilities that exist in hardware.

Pixel Array sub-device

The pixel array sub-device represents the camera sensor's pixel matrix, as well as analogue crop functionality present in many compliant devices. The analogue crop is configured using the `V4L2_SEL_TGT_CROP` on the source pad (0) of the entity. The size of the pixel matrix can be obtained by getting the `V4L2_SEL_TGT_NATIVE_SIZE` target.

Binner

The binner sub-device represents the binning functionality on the sensor. For that purpose, selection target `V4L2_SEL_TGT_COMPOSE` is supported on the sink pad (0).

Additionally, if a device has no scaler or digital crop functionality, the source pad (1) expses another digital crop selection rectangle that can only crop at the end of the lines and frames.

Scaler

The scaler sub-device represents the digital crop and scaling functionality of the sensor. The V4L2 selection target `V4L2_SEL_TGT_CROP` is used to configure the digital crop on the sink pad (0) when digital crop is supported. Scaling is configured using selection target `V4L2_SEL_TGT_COMPOSE` on the sink pad (0) as well.

Additionally, if the scaler sub-device exists, its source pad (1) exposes another digital crop selection rectangle that can only crop at the end of the lines and frames.

Digital and analogue crop

Digital crop functionality is referred to as cropping that effectively works by dropping some data on the floor. Analogue crop, on the other hand, means that the cropped information is never retrieved. In case of camera sensors, the analogue data is never read from the pixel matrix that are outside the configured selection rectangle that designates crop. The difference has an effect in device timing and likely also in power consumption.

Register definition generator

The ccs-regs.asc file contains MIPI CCS register definitions that are used to produce C source code files for definitions that can be better used by programs written in C language. As there are many dependencies between the produced files, please do not modify them manually as it's error-prone and in vain, but instead change the script producing them.

Usage

Conventionally the script is called this way to update the CCS driver definitions:

System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\media\drivers\ccs\[linux-master][Documentation][driver-api][media][drivers][ccs]ccs.rst, line 74)

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none

        $ Documentation/driver-api/media/drivers/ccs/mk-ccs-regs -k \
                -e drivers/media/i2c/ccs/ccs-regs.h \
                -L drivers/media/i2c/ccs/ccs-limits.h \
                -l drivers/media/i2c/ccs/ccs-limits.c \
                -c Documentation/driver-api/media/drivers/ccs/ccs-regs.asc
```

CCS PLL calculator

The CCS PLL calculator is used to compute the PLL configuration, given sensor's capabilities as well as board configuration and user specified configuration. As the configuration space that encompasses all these configurations is vast, the PLL calculator isn't entirely trivial. Yet it is relatively simple to use for a driver.

The PLL model implemented by the PLL calculator corresponds to MIPI CCS 1.1.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-

master\Documentation\driver-api\media\drivers\ccs\[linux-master][Documentation][driver-api][media][drivers][ccs]ccs.rst, line 93)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/media/i2c/ccs-pll.h
```