HiKey970

Vcode Development Guide

Issue 01

Date 2018-03-11

Trademarks and Permissions

 , HISILICON, and other HiSilicon icons are trademarks of HiSilicon Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

Change History

Changes between document issues are cumulative. The latest document issue contains all the changes made in earlier issues.

Issue 01 (2018-03-11)

The first version.

Contents

1 Description

1.1 VENC Function Description

1.1.1 Summary

The Hi3670 integrates an H.264/H265 hardware encoder and supports the H.264/H265/JPEG encoding standard.

VENC features:

Contains an H264 encoder.
Tools used by the H.264 Baseline Profile:
I and P slice.
4x4/16x16 partition support in frame, 16x16 support DC/V/H prediction mode, 4x4 support DC/V/H prediction mode
Inter-frame support division of 8x8/16x16
Support MB class rate control
Supports 1/2 and 1/4 pixel inter prediction.
The cosine transform supports the Hadamard transform.
Support deblocking.
H.264 Main Profile to increase the use of tools:
Supports CABAC.
H.264 High Profile added tools:
Add 8x8 transforms in addition to 4x4 transforms.
Supports intra 8x8 prediction.
Input image data format

planar YUV 4:2:0

planar YUV 4:2:2

semi-planar YUV 4:2:0

semi-planar YVU 4:2:0

package UYVY4:2:2，VYUY4:2:2

package YUYV4:2:2，YVYU4:2:2

ARGB/BGRA8888

ABGR/RGBA8888

The output data format is the original code stream of each format.
Contains an H265 encoder.

Tools used by H.265 MainProfile:
I and P slice.
Intra-frame support 4x4/8x8/16x16/32x32 partition, 4x4/8x8/16x16 partition support 35 kinds of prediction modes, 32x32 support DC/Planar prediction mode;
Inter-frame support 8x8/16x16/32x32/64x64 partitioning
Supports CU-level rate control
Supports positive and negative levels of 512, vertical and negative 144 integer search
Supports 1/2 and 1/4 pixel inter prediction.
Supports DCT4/8/16/32, supports DST4.
Supports Merge/MergeSkip
Supports TMV
Support deblocking/SAO.
Input image data format

planar YUV 4:2:0

planar YUV 4:2:2

semi-planar YUV 4:2:0

semi-planar YVU 4:2:0

package UYVY4:2:2，VYUY4:2:2

package YUYV4:2:2，YVYU4:2:2

ARGB/BGRA8888

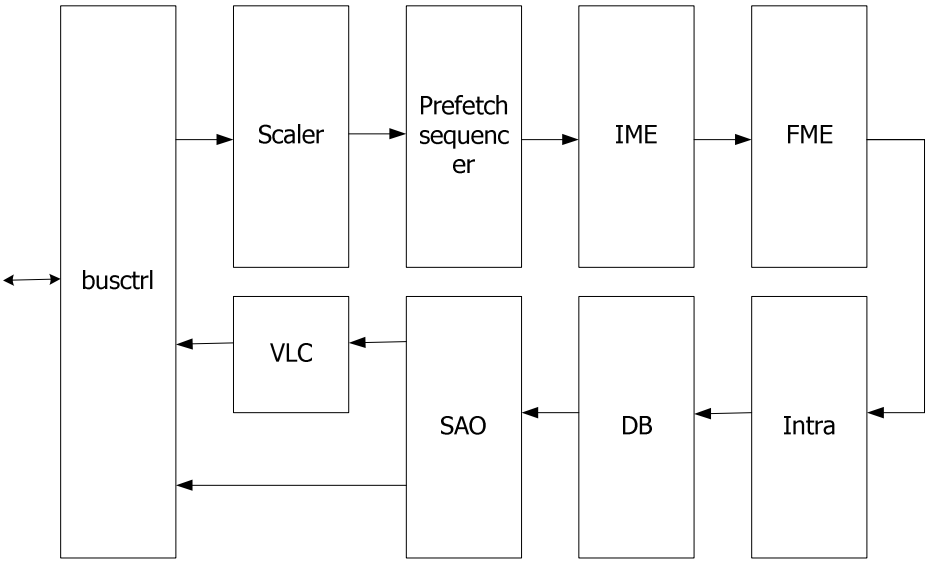
ABGR/RGBA8888

The output data format is the original code stream of each format

1.1.2 Functional block diagram

The architecture of the VENC core is shown in the figure. The frame to be encoded is directly read into the sequencer via busctrl, and subsequent encoding is started. After the encoding is completed, the busctrl is output to the bus.

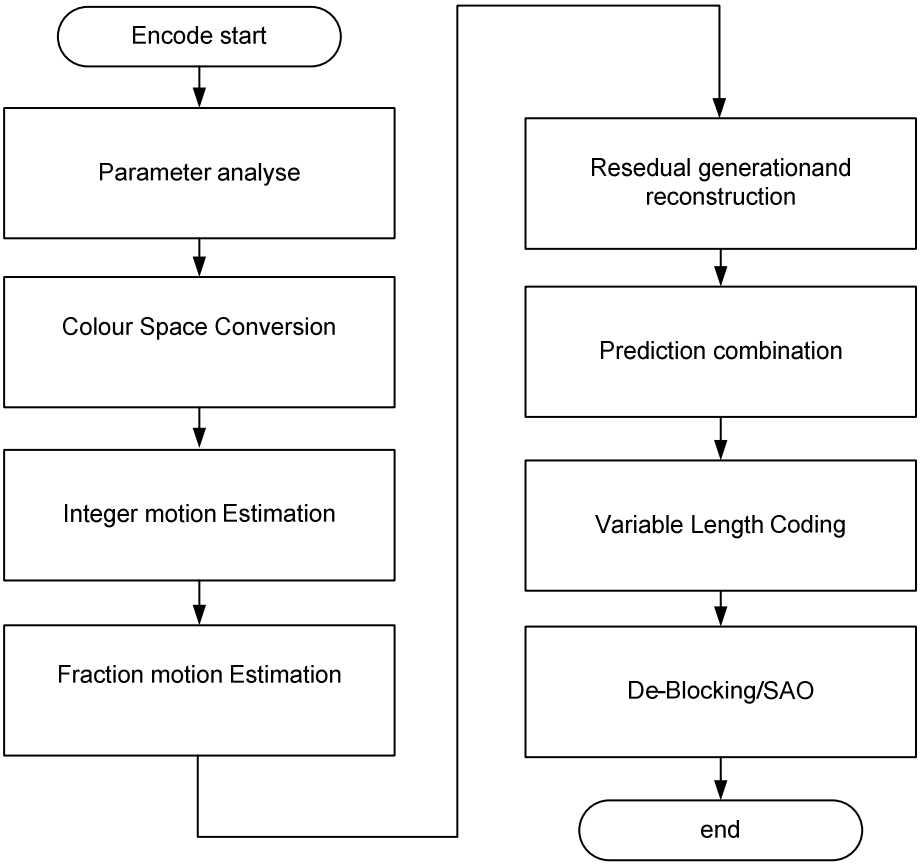
Figure 1 The functional architecture of VENC core



1.1.3 Coding process

VENC coding process as shown. The parameters are analyzed first after the coding starts, then the color space conversion of the input video source is performed through Color Space Conversion, followed by an integer motion vector search in the IME, and also followed by a sub-pixel motion vector search in FME, and finally result is combined after the Intra. Then we can reconstruct image input to DB/SAO, with the residual stream input to VLC for encoding.

Figure 2 VENC



1.2 VDEC Function Description

1.2.1 Summary

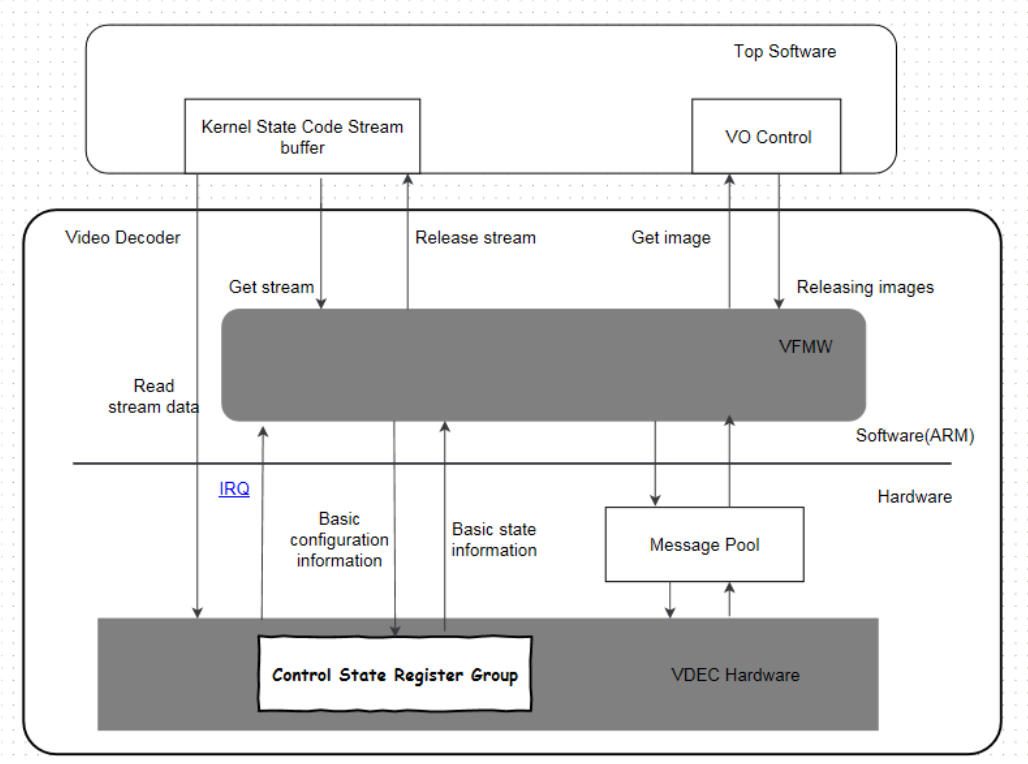
The Hi3670 integrates a video decoder VDEC that supports the H.265, H.264, MPEG2, MPEG4, VP8, and VP9 protocols.

VDEC is composed of VFMW (Video Firmware) and embedded hardware video decoding engine running on ARM processor. VFMW obtains code stream from upper layer software, parses the code stream and calls hardware video decoding engine to generate decoded image sequence. The decoded image sequence is output from the subsequent module to the monitor or other device under the control of the upper software.

1.2.2 function architecture

VDEC video decoder architecture as shown.

Figure 3 video decoder architecture



- VDEC: Multi-protocol video decoding engine
- VFMW: Video Firmware, video firmware, is actually a software component running on the host processor. It is responsible for scheduling video decoding engines and completing video decoding.
- Message pool: The storage space for information exchange between the VFMW and VDEC hardware, which is opened in the external SDRAM memory and can be read and written by VDEC and VFMW.

VDEC hardware and VFMW interactive mode:

- According to a batch of slices to complete the decoding interaction, VFMW complete the slice header and above decoding, VDEC hardware to complete the slice data and the following decoding.
- To support multi-channel decoding, VFMW starts the VDEC hardware by frame switching, and multiplex decoding is implemented in a time-multiplexed manner.

1.3 Development

1.3.1 DTS Configuration

vcode dts configuration: arch/arm64/boot/dts/hisilicon/kirin970-hikey970.dt

```
vdec {
    compatible = "hisi,kirin970-vdec";
    reg = <0x0 0xe8800000 0x0 0x40000>;
    interrupts = <0 290 4>, <0 291 4>, <0 292 4>;
```

```
        clocks = <&media2_crg KIRIN970_CLK_GATE_VDECFREQ>;
        clock-names = "clk_gate_vdecfreq";
        dec_clk_rate = <450000000>, <300000000>, <185000000>;
        vdec_fpga = <0x1>;
        status = "ok";
        iommu_info {
                    start-addr = <0x40000>;
                    size = <0xE0000000>;
       };
};
venc {
     compatible = "hisi,kirin970-venc";
     reg = <0x0 0xe8880000 0x0 0x40000>;
     interrupts = <0 296 4>, <0 297 4>, <0 298 4>;
     clocks = <&media2_crg KIRIN970_CLK_GATE_VENCFREQ>;
     clock-names = "clk_gate_vencfreq";
     enc_clk_rate = <554000000>, <450000000>, <238000000>;
     venc_fpga = <0x1>;
     status = "ok";
     iommu_info {
                 start-addr = <0x40000>;
                 size = <0xE0000000>;
     };
};
```

1.3.2 Device Driver Configuration

[Add driver configuration]

Kernel supports Vcode function, modify kernel：arch/arm64/configs/hikey970_defconfig

CONFIG_HI_VCODEC_VENC=y

CONFIG_HI_VCODEC_VDEC=y

Vcode-related driver specific locations such as drivers/vcodec, which is divided into encoder and decoder driver code

[Registration Driver]

Interfaces that call the standard driver are registered in the system, where the drive interface logic for VENC and VDEC is as shown in the figure below.