NXP Semiconductors
Application Note

Document Number: AN12060
Rev. 2, 12/2018

Software ISP Application Note

1. Introduction

This document describes the software-based image signal processing application (SW-ISP) and its implementation on the i.MX8 family device System-on-Chip [1] (SoC) processors. The application is used for image processing in i.MX8 family device. This pipelined image processing engine is optimized by on-chip GPU. It includes functions of Bad Pixel Correction, White Balance, Histogram Equalization, High-Quality Demosaicing, and High-Quality Noise Reduction. Its purpose is to use on-chip GPU to implement ISP function with a high-speed image processing.

Contents

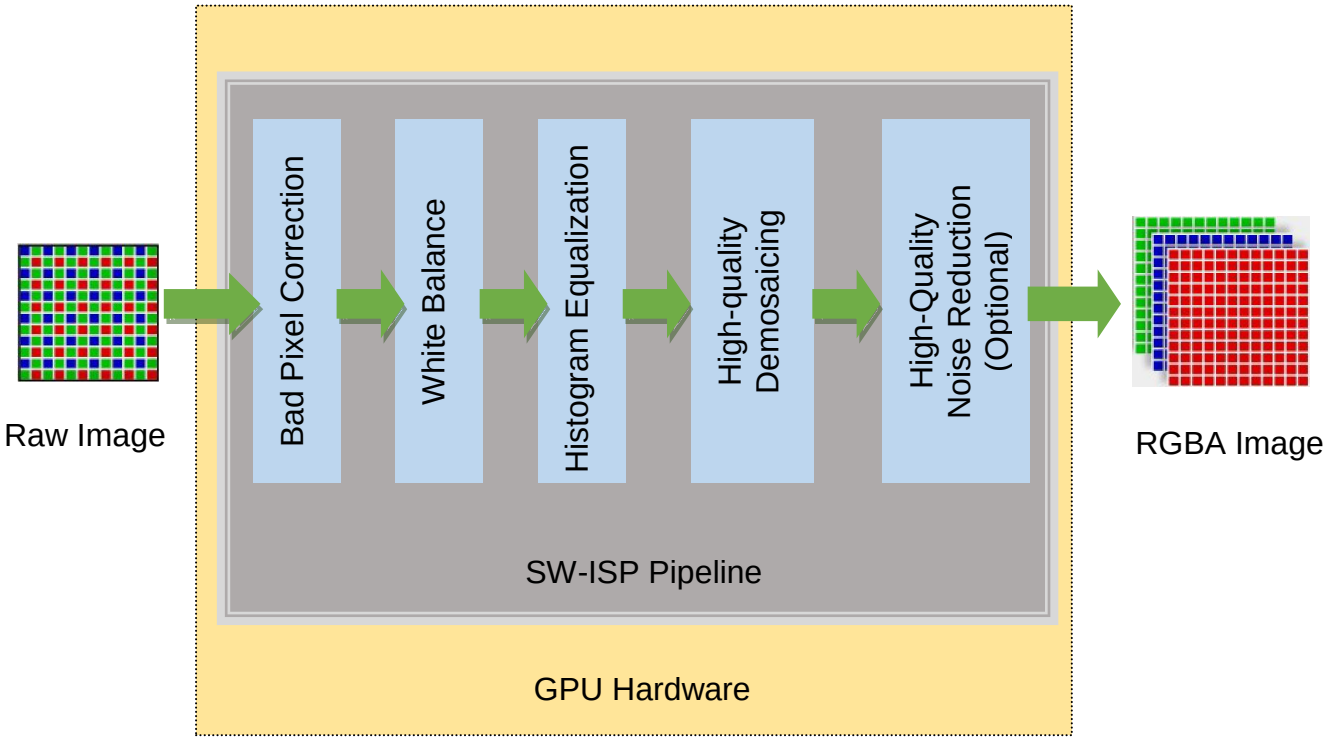

Figure 1. SW-ISP pipeline block diagram

2. Software theory

From the software point of view, the full SW-ISP application consists of these parts:

- Linux OS: The SD card image is created by using the Yocto Project[1].
- SW-ISP pipeline optimized by OpenCL1.2[2].
- SW-ISP pipeline optimized by OpenVX1.1[2].

SW-ISP pipeline includes five functions of image processing. Bad Pixel Correction algorithm[3], White Balance algorithm[4] , Histogram Equalization algorithm[5] and High-Quality Demosaicing algorithm[6] are applied into Bayer image[1], while High-Quality Noise Reduction algorithm[7] is applied into Y-channel alone. The High-Quality Noise Reduction process consists of following steps:

1) The RGBA image is converted to YUV image[1].
2) Apply bilateral filtering[7] to Y-Channel to reduce noise of image.
3) Filtered Y-Channel image is merged into UV-Channels to covert to RGBA image.

In both OpenCL and OpenVX pipelines, input raw images captured by Bayer camera (Used color filter array over Bayer camera is shown in *Figure* 2) are read from file. The raw values would be uploaded to a GPU as a grayscale texture.

Every function in SW-ISP pipeline is independent, so they can be combined willfully. All the functions except High-Quality Noise Reduction are enable in both SW-ISP OpenCL pipeline and SW-ISP OpenVX pipeline. That's because the kernels running on GPU are compiled and built online and High-

Quality Noise Reduction is so complex that it takes long time in OpenCL/OpenVX pipeline when compiling the kernel. This function can be enable by command line “--Enable”.

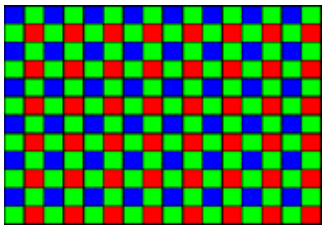

Figure 2. Bayer color filter array

NOTE

The color filter array is blue-green-green-red (BG/GR). It is 50% green, 25% blue, and 25% red.

Figure 3 shows effects of each algorithm:

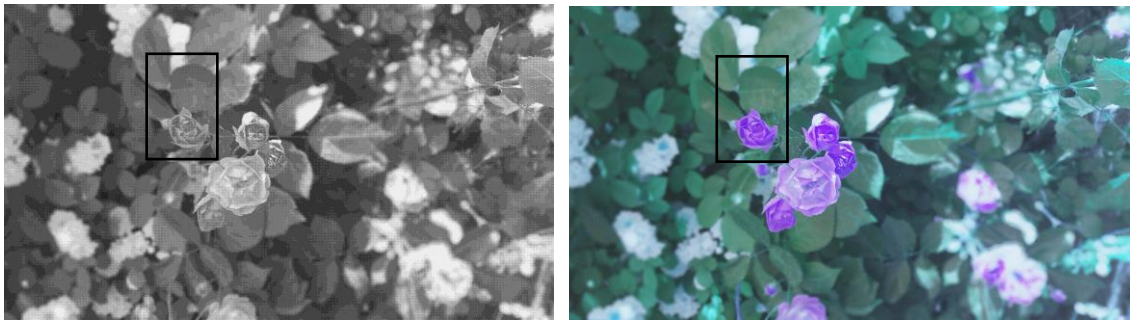

Figure 3. Comparison of algorithms applied to an image.

NOTE

Because the currently released software ISP works in the demo mode only, the input array of Bayer image is BG/GR mode.

2.1. SW-ISP Pipeline Optimized by OpenCL

In OpenCL pipeline, combine different functions of SW-ISP by setting kernel arguments. The detailed directory structure of the SW-ISP of OpenCL application is shown in *Figure* 4:

```
SoftISP
├── build
│   └── Yocto
├── Content
│   ├── bayer.data
│   ├── isp_kernel.cl
│   └── License.json
├── Example.jpg
├── Fsl.gen
├── GNUmakefile_Yocto
├── README.md
├── SoftISP
└── source
    ├── SoftISP.cpp
    ├── SoftISP.hpp
    └── SoftISP_Register.cpp
```

Figure 4. Directory structure of the SW-ISP of OpenCL application

The root directory is called *SoftISP* under *OpenCL*. There are three subdirectories and an executable binary file in the root directory:

- Executable binary *"SoftISP"* is in folder of SoftISP.
- *Build* - contains OBJ files generated by the SW-ISP.
- *Content* - contains the prepared raw images ("*bayer.data*") in the **.data* format. The raw image was captured by Bayer camera (resolution: 1920 x 1080, Bayer array: BG/GR). The application can read raw data by the filename of *"bayer.data"*.
- *Source* - contains source code running on CPU and GPU. *"isp_kernel.cl"* is C code of kernels.

2.2. SW-ISP Pipeline Optimized by OpenVX

In OpenVX pipeline, combination of different functions are controlled by graphs in OpenVX. The detailed directory structure of the SW-ISP of OpenVX application is shown in *Figure* 5:

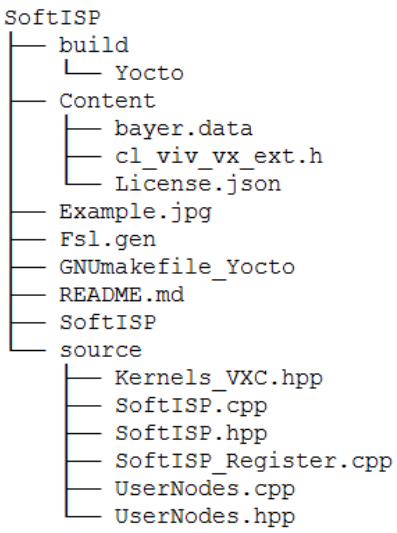

Figure 5. Directory structure of the SW-ISP of OpenVX application

The root directory is called *SoftISP* under *OpenVX*. There are three subdirectories and an executable binary file in the root directory:

- Executable binary *"SoftISP"* is in folder of SoftISP.
- *Build* - contains OBJ files generated by the SW-ISP.
- *Content* - contains the prepared raw images ("*bayer.data*") in the **.data* format. The raw image was captured by Bayer camera (resolution: 1920 x 1080, Bayer array: BG/GR). The application can read raw data by the filename of *"bayer.data"*. "*cl_viv_vx_ext.h*" is header of OpenVX API used in i.MX8.
- *Source* - contains source code running on CPU and GPU. *"Kernels_VXC.hpp"* is code of kernels.

3. Performance measurement for i.MX8 Series applications processors

This section describes the profiling results of the SW-ISP application. Two implementation versions are available: first one based on OpenCL framework and second one based on OpenVX platform.

The measurements were performed on all families of the i.MX8 Series Applcations Processors with OpenCL support. The boards used for the measurements are: : i.MX8QuadMax, i.MX8MQuad and i.MX8QuadXPlus. i.MX8QuadMax supports both OpenCL and OpenVX, while i.MX8MQuad and i.MX8QuadXPlus support only OpenCL.

Two main use cases are compared:

- OpenCL vs OpenVX results on i.MX8QuadMax
- Results obtained with OpenCL on i.MX8QuadMax, i.MX8MQuad and i.MX8QuadXPlus

The profiling was performed in the following environment:

Table 1. Common SW-ISP profiling conditions

Feature	Input method	Graphical backend	Platform	BSP	GPU driver	OpenVX (just i.MX8QuadMax)	OpenCL	Used HDMI resolution	Doc
Description	Image (1920 x 1080, Bayer)	wayland	OpenCL, OpenVX	4.9.88	6.2.4	1.1	1.2FP	1920 x 1080 (1080p)	Software ISP Application

Please find below a relevant selection of the hw/sw capabilities of the two boards:

Table 2. Hardware/Software capabilities of the boards

FEATURE	i.MX8QM	i.MX8MQ	i.MX8QXP
CPU	2 x A72 4 x A53	4 x A53	4 x A35
GPU	2 x GC7000XSVX	1 x GC7000Lite	1 x GC7000Lite
DDR	LPDDR4 @ 1600 MHz 2 x 32b 6 GB	LPDDR4 @ 1600 MHz 1 x 32b 4 GB	LPDDR4 @ 1200 MHz (no ECC) 1 x 32b 3 GB
OpenCL	1.2 FP	1.2 FP	1.2 FP
OpenVX	1.1	n/a	n/a

NOTE

- SoftISP runs on one CPU when GPU is running.
- SoftISP runs on one GPU (even if 8QuadMax has two available).
- The CPU is used to load the data. The GPU is used to run the pipeline algorithms.
- The CPUs can run at certain frequencies. There is no common frequency between all three processors, so the measurements were performed in similar but not identical conditions:
- i.MX8QuadMax: A72@1.5 GHz; A53@1.2 GHz

3.1. SW-ISP Pipeline Optimized by OpenCL/OpenVX for i.MX8QuadMax

Table 3 and *Table* 4 show system memory profiling by Linux tool.

To obtain the results in this section, the application was run on A72 core @ 1.6 GHz. It was modified to load the data and run the algorithms in a loop, to be able to compute average results.

- CPU and memory usage

Table 3. Memory used and CPU usage

	VIRTUAL (MB)	PHYSICAL (MB)	SHARED (MB)	%CPU	%PHYSICAL
OpenCL SoftISP	414	58	11	11	1
OpenVX SoftISP	320	37	10	25	0.6

- DDR bandwidth.

Table 4. DDR bandwidth

	Read-Cycles (/s)	Read (MB/s)	Write-Cycles (/s)	Write (MB/s)
OpenCL SoftISP	56,621,872	863	23,245,245	355
OpenVX SoftISP	86,753,842	1,324	60,161,806	918

Table 5 shows the statistic results of distinct functions. The default pipeline includes bad pixel correction node, white balance node and high-quality demosaicing node.

Table 5. Comparison of SW-ISP OpenCL and OpenVX on i.MX8QM on A72@1.6GHz

SW-ISP Functions	Consumed time of OpenCL (ms)	Consumed time of OpenVX(ms)
Bad Pixel Correction Node	34	5
White Balance Node	4	7
Histogram Equalization Node	5	3
Demosaicing Node	9	2
RGBA2YUV Node	7	4
High-quality Noise Reduction Node	10,923	1,673
YUV2RGBA Node	9	3
Pipeline(default)	52	20

3.2. SW-ISP Performance comparison between between i.MX8 Series Application Processors for OpenCL 1.2

There are only a few fixed frequencies that can be set for the CPUs for each board. There is no common frequency between all three boards, so the measurements were performed in similar but not identical conditions:

- i.MX8QuadMax: A53@1.2 GHz
- i.MX8MQuad: A53@1 GHz
- i.MX8QuadXPlus: A35@1 GHz

For i.MX8QuadMax A53 core was chosen to run the application, as it could be set at a frequency value closer to the ones available on the other two boards (1.2 GHz vs 1 GHz).

- CPU and memory usage

Table 6. Memory and CPU usage

	VIRTUAL (MB)	PHYSICAL (MB)	SHARED (MB)	%CPU	%PHYSICAL
i.MX8QuadMax A53@1.2GHz	414	58	11	12	1
i.M8MQuad A53@1GHz	414	59	11	8	2
i.MX8QuadXPlus A35@1GHz	413	58	11	6.7	2

The algorithm uses the CPU to load the input data. Only one of the available CPUs is used to perform this operation.

- DDR bandwidth

Table 7. DDR bandwidth

	Read-Cycles (/s)	Read (MB/s)	Write-Cycles (/s)	Write (MB/s)
i.MX8QuadMax LPDDR4 @ 1600 MHz 2 x 32b 6 GB	1,632,089,432	844	695,038,942	359
i.M8MQuad LPDDR4 @ 1600 MHz 1 x 32b 4 GB	1,744,614,872	467	680,330,891	182
i.MX8QuadXPlus LPDDR4 @ 1200 MHz (no ECC) 1 x 32b 3 GB	1,657,446,637	460	1,013,582,601	281

- Pipeline stages

Table 8. Pipeline stages

	i.MX8QuadMax GC7000XSVX	i.M8MQuad GC7000L	i.MX8QuadXPlus GC7000Lite
Bad Pixel Correction Node	34	68	68
White balance Node	4	3	4
Histogram Equalization Node	7	11	10
Demosaicing Node	9	21	20
RGBA2YUV Node	7	15	15
High-quality Noise Reduction Node	7,360	21,674	21,159
YUV2RGBA Node	9	21	21
Pipeline	54	104	102

NOTE

- Pipeline (default) value is sum of Bad Pixel Correction, White Balance, Histogram Equalization, Demosaicing.
- SoftISP runs on only one GPU (even if 8QM has two available).
- Different GPU's are used for each board: GC7000XSVX for i.MX8QuadMax and GC7000L for i.MX8MQuad.
- GFLOPS on GC7000XSVX is twice the value of GC7000L. This explains why most of the algorithms running on i.MX8MQuad last twice longer than the ones running on i.MX8QuadMax.
- The performance of an application can be influenced by several factors including GFLOPS and caching method.

3.3. Profiling guidelines

- CPU and memory usage were performed using Linux 'top' command.
- DDR bandwidth was measured using Linux 'perf stat' command.
  # perf stat -I 1000 -a -e ddr0/read-cycles/,ddr0/write-cycles/

NOTE

i.MX8QaudMax has 2 DDR controllers that must be taken into consideration:

 # perf stat -I 1000 -a -e ddr0/read-cycles/,ddr0/write-cycles/,ddr1/read-cycles/,ddr1/write-cycles/

- To run an application on a specific CPU use taskset command. It uses a bitmask, with the lowest order bit corresponding to the first logical CPU and the highest order bit corresponding to the last logical CPU. E.g., to run SoftISP on core 3:
  root@imx8qmmek:~# taskset 0x8 ./SoftISP
  To obtain the index of each CPU:
  root@imx8qmmek:~# cat /proc/cpuinfo
  To check that the process is really running on the specified core, check PSR column:

  root@imx8qmmek:~# ps -aF

4. Conclusion

This application represents a software ISP solution.

Its performance is influenced by the hardware capabilities: GPU type, CPU frequency, DDR features.

The SW-ISP application has these main advantages:

- High application performance.
- Algorithm of ISP optimized deeply by OpenCL/OpenVX.
- Support different combination of functions.
- SW-ISP optimized by OpenCL support most OpenCL platform.

5. References

1. System on Chip, Yocto Project, Bayer Image, YUV/RGB Convert, available at https://en.wikipedia.org.
2. OpenCL 1.2 and OpenVX 1.1 Reference Pages, available at https://www.khronos.org/.
3. Bad pixel detection and correction algorithm is referred to http://ieeexplore.ieee.org/abstract/document/6419046/.
4. White balance algorithm is referred to https://courses.cs.washington.edu/courses/cse467/08au/labs/l5/whiteBalance.pdf.
5. Histogram Equalization algorithm is referred to https://en.wikipedia.org/wiki/Histogram_equalization.
6. High-quality demosaicing algorithm is referred to http://www.tandfonline.com/doi/abs/10.1080/2151237X.2008.10129267.
7. High-quality noise reduction algorithm is referred to https://en.wikipedia.org/wiki/Bilateral_filter.
8. i.MX8 Series Applications Processors

   https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/i.mx-applications-processors/i.mx-8-processors:IMX8-SERIES
9. Vivante GC7000 GPUs

   http://www.vivantecorp.com/index.php/en/media-article/news/277-20140403-vivante-gc7000-delivers-desktop-graphics-to-mobile.html

6. Revision history

Table 9. Sample revision history

Revision number	Date	Substantive changes
0	10/2017	Initial release
1	09/2018	Source release
2	12/2018	Chapter 3 & 4 updated

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AN12060
Rev. 2
12/2018