White Paper

Intel® Quick Capture Technology for the Intel® PXA27x Processor Family

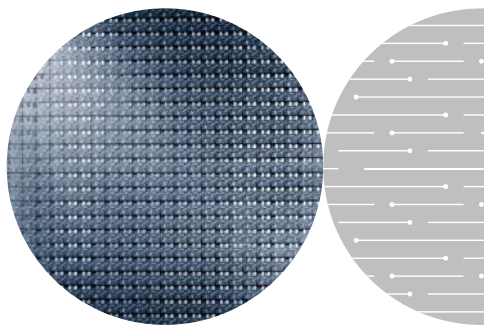


Table of Contents

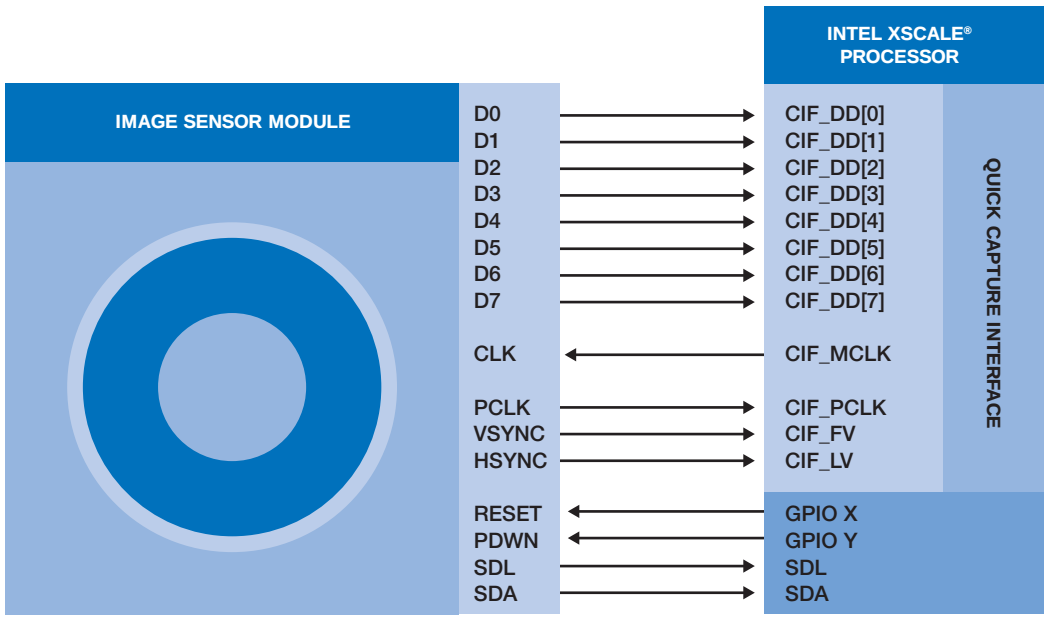


Figure 1. Typical 8 bit master parallel interface connections

5.0 Mode Summary

The Intel Quick Capture interface is highly configurable and can operate in a variety of modes: Master mode, slave mode, embedded mode, parallel and serial modes. If you are not familiar with the differences between these modes, please take a few moments to read the rest of this section. Note that the terms master and slave are from the image sensor's point of view. In master mode, the image sensor is the master and in slave mode, the image sensor is the slave.

5.1 What is Master Mode?

Master mode refers to a mode of operation where the image sensor module provides the line and frame synchronization signals. The line synchronization signal is commonly referred to as HSYNC or "line valid" and the frame synchronization signal is commonly referred to as VSYNC or "frame valid." For the Intel PXA27x master mode, this means the line valid and frame valid signals are inputs to the Intel PXA27x Quick Capture interface. Master mode is a common option for many of today's image sensors.

5.2 What is Slave Mode?

Slave mode refers to a mode of operation where the image sensor receives the synchronization signals externally. For the Intel PXA27x slave mode, this means the line valid and frame valid signals are outputs from the Intel Quick Capture interface.

5.3 What is Embedded Mode?

Embedded mode is a mode of operation where the line valid and frame valid signals are embedded in the data stream. This mode of operation provides a reduced pin count solution because separate pins are not required for the line and frame synchronization signals. The embedded mode follows the ITU-R BT.656-4 standard for Start-of-Active-Video, SAV, and End-of-Active-Video, EAV encoding convention. Embedded mode is a master mode and therefore the embedded synchronization encoding is generated by the image sensor and sent to the Intel PXA27x processor.

5.4 What is the Difference Between Parallel and Serial Modes?

The parallel option is what you might expect for a typical parallel interface. For example, if you have an 8 bit data bus and 8-bit data, all 8-bits get sent at the same time. The serial mode is slightly different than what you might normally think about regarding a serial interface. Serial mode is really a 4- or 5-bit data bus solution for 8- or 10-bit data. For serial mode, 8-bit data is transferred in two 4-bit transfers over a 4-bit data bus. For 10-bit data, two 5-bit transfers are performed over a 5-bit data bus. Therefore, the data is not transferred over a single data line that is typical of serial interfaces. While the serial option provides a reduced pin count solution, currently a parallel interface is more common.

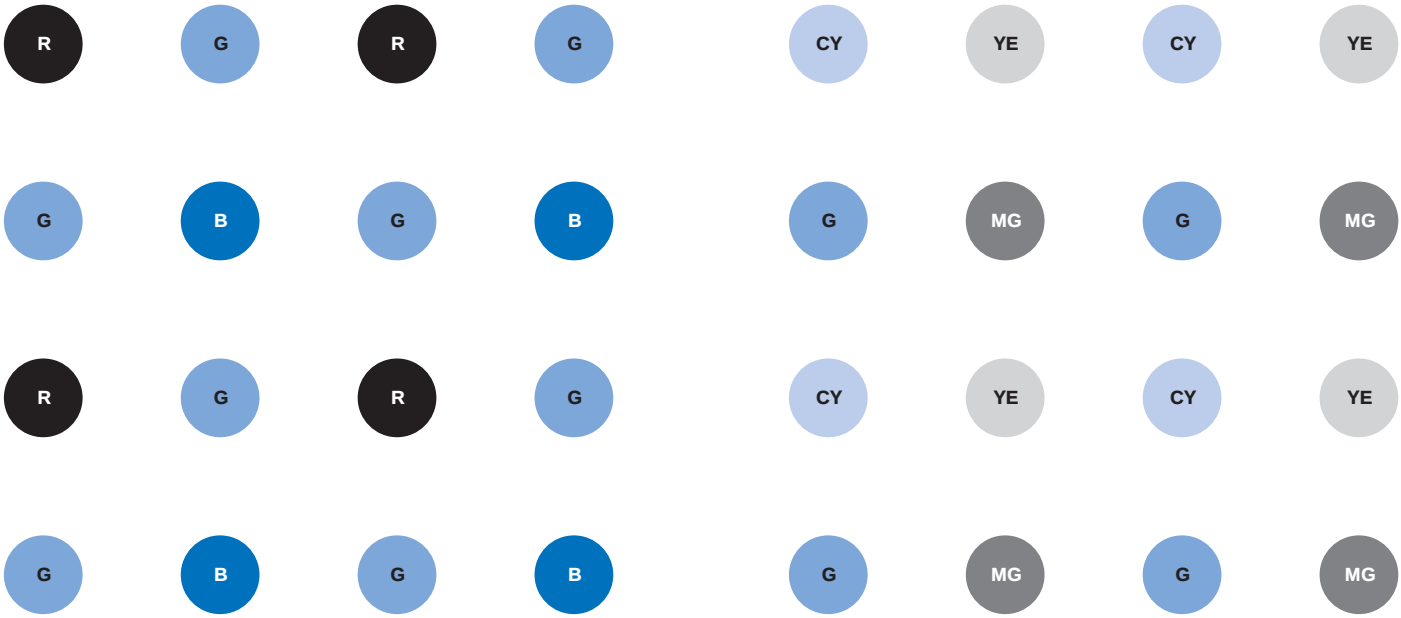


Figure 2. Illustration of RAW Bayer Formats

6.0 Color Space Discussion

The choice of color space is a very important part of any imaging solution. Your choice of color space format can have a significant impact on the overall performance of your imaging solution.

6.1 Color Space Formats

The color sensitivity for modern image sensors is provided by a color filter array (CFA) applied over an array of photo detectors. The detectors convert the light that passes through the CFA material into a signal that is measured and digitized. There is a lot of variation between manufacturers of color image sensors when it comes to the implementations of CFAs. The filter array material is usually applied in an interleaved fashion. The pattern may be either in stripes or a structured mosaic. The component filter material is designed to pass either the primary colors red, green, and blue, or the secondary colors cyan, yellow, magenta wavelengths. A mosaic pattern for RGB and the CYMG color filters is illustrated in Figure 2, “Illustration of RAW Bayer Formats.”

The RGB color space provides a scheme for three numbers representing the relative proportions of red, green, and blue at each pixel location. When combined, these three additive primary colors can be used to produce any other color. The RGB color space is used for both the capture and display of image data since the varying proportions can be combined easily. However, the human visual system is more sensitive to the luminance, or brightness of the scene, than to the color, so you can realize a reduction in data in the compression process by taking advantage of this characteristic.

The separation of the luminance from the color information is accomplished through the color space transformation of RGB to YCbCr. In this transformation, the luminance, Y, is the weighted average of the R, G, and B components, and each chrominance sample is the weighted difference between the luminance and the R and G components in Figure 3, next page.

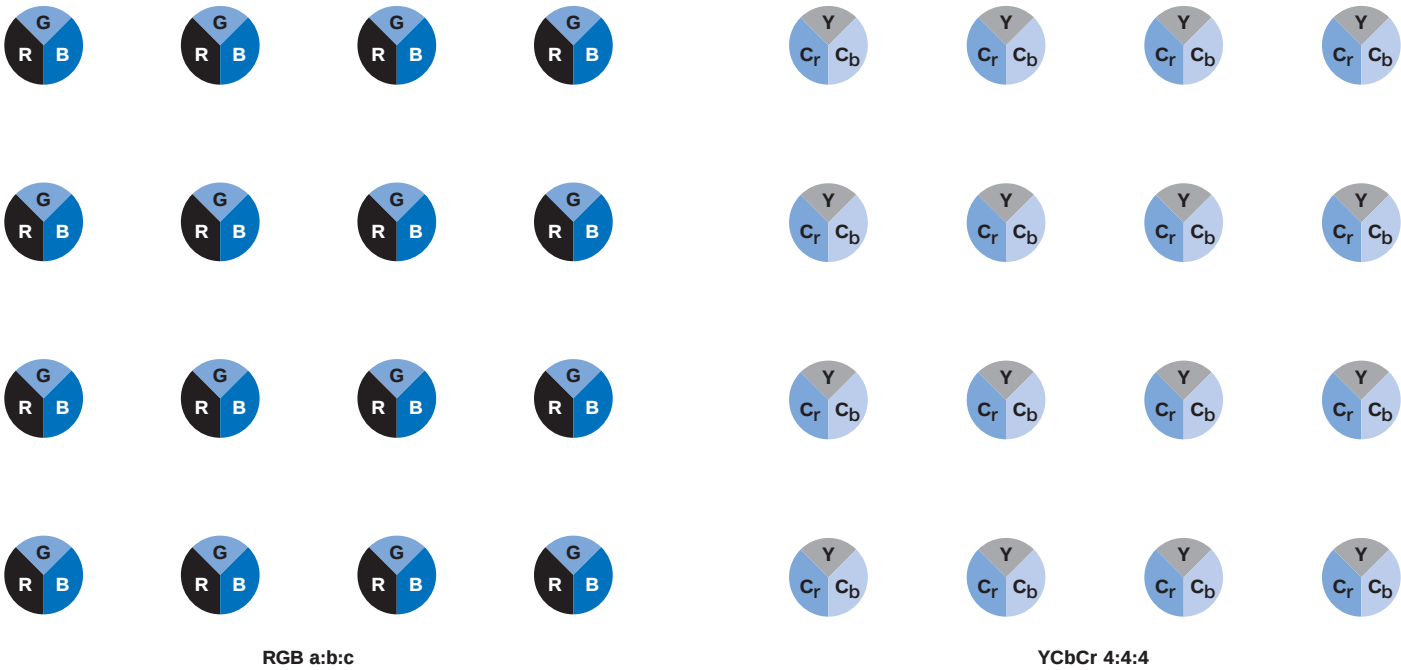


Figure 3. Illustration of RGB a:b:c and YCbCr 4:4:4 Formats.

The key advantage of using the YCbCr color space is that we can use a lower spatial resolution when compressing or storing the images since the visual system is less sensitive to the color than luminance. The two formats that are most popular use the 4:2:2 and 4:2:0 sampling patterns. In the 4:2:2 format, the chrominance samples are averaged across two spatial locations whereas the 4:2:0 averages them across four sample locations. The 4:2:2 and 4:2:0 sampling formats are illustrated in Figure 4,next page, “Illustration of YCbCr 4:2:2 and YCbCr 4:2:0.”

6.2 Choice of Color Space

One of the most important decisions to be made when designing your camera solution is to determine what color space format(s) you are going to use. This decision can greatly impact the amount of processing that is done via hardware and how much is done via software. Intel Quick Capture Technology in the Intel PXA27x processor includes some features which can help with the decision process. This is a good place in the discussion to look at an example user scenario for still capture. See Figure 5, “Still capture scenario.” Additional example scenarios for video capture and video conferencing can be found in Section 9.0, “Example user scenarios” on page 11.

The situation we have with mobile devices is that the user will typically want to display the camera image on their device's screen for preview as well as encode the data into JPEG or MPEG4 formats so the data can be stored on their device as pictures or video clips. The RGB color space format is normally required for viewing images on a display screen like during preview, and the YCbCr color space format is required for encoding. Hence, we have a need for the same data to be in both YCbCr and RGB color space formats. The Intel PXA27x contains a solution for this situation.

Another component of Intel Quick Capture Technology is the hardware color space conversion engine in overlay 2 of the LCD controller. The color space conversion engine is capable of accepting YCbCr data and converting it to the RGB color space format.

Question:

Why is it important that Overlay 2 can accept YCbCr data?

Answer:

Efficiency.

1) This allows incoming YCbCr data to be streamed directly to overlay 2 without first having to do any RGB conversion. The color space conversion engine in overlay 2 can efficiently convert the YCbCr data to RGB for display. Be aware that the converted data is available for display only by the LCD controller and cannot be accessed outside of the LCD controller.

2) The other benefit is that the incoming YCbCr image data can easily be reformatted from YCbCR 4:2:2 to YCbCr 4:2:0, the preferred format for input into an encoder. Another component of Intel Quick Capture Technology is Intel® Wireless MMX™ Technology. It can be used to efficiently perform the 4:2:2 to 4:2:0 conversion.

Question:

What if the RGB color space is used as the format for the sensor output?

Answer:

The Intel Quick Capture Interface is easily configured to receive data in a variety of RGB formats. However, if your user scenario is similar to the scenario above where you want to preview data as well as perform encoding, there will be consequences. If the data is received in RGB format, the processor-intensive task of software-based color space conversion of RGB to YCbCr will be required. If software-based color space conversion is required, check the latest version of the Intel® Integrated Performance Primitives for optimized functions to help with that conversion task. More information about Intel Integrated Performance Primitives can be found at the following URL:

http://www.intel.com/software/products/ipp/

Because of the conditions outlined above, it is obvious that YCbCr color space format is the most efficient and therefore the preferred input format for the Intel PXA27x. That is why all of the user scenarios in this document use the YCbCr colorspace format.

6.3 Using a RAW format

For applications where the highest quality still images are required, the use of RAW mode along with a software-based RAW processing chain may be desirable. This is another example of where Intel Wireless MMX Technology and Intel Integrated Performance Primitives can be used to increase the performance of your imaging solution. See the latest release of the Intel Integrated Performance Primitives library for image processing related functions.

In RAW mode, image data on the interface data bus is treated as just data so no assumptions are made about the format of that data. This allows the Intel Quick Capture Interface to accept data in most any raw format that can be transferred over an 8, 9,or 10 bit data interface.

The key issue to remember with the use of RAW mode is that the software running on the Intel PXA27x must be programmed to understand the particular RAW format in which the sensor is providing data. Figure 2, "Illustration of RAW Bayer Formats." on page 5 shows examples of RAW color output formats.

7.0 Can I Use My Particular Sensor with the Intel Quick Capture Interface on the Intel® PXA27x Processor?

To help you determine if your sensor will work with the Intel PXA27x processor, review the image sensor's datasheet to answer the list of questions that follow.

Question 1

What mode(s) of operation does your sensor support?

If parallel mode is supported go to question 1A, if serial mode is supported, go to question 1B

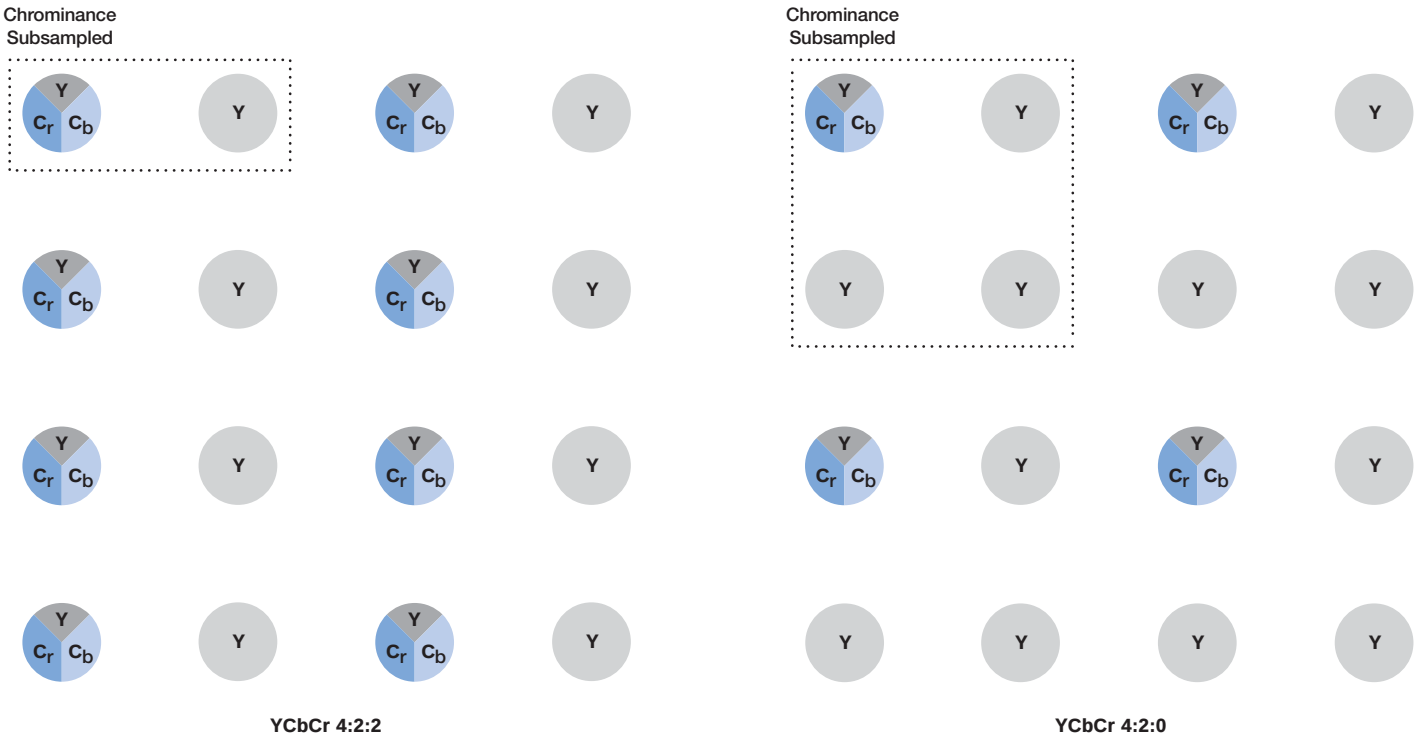
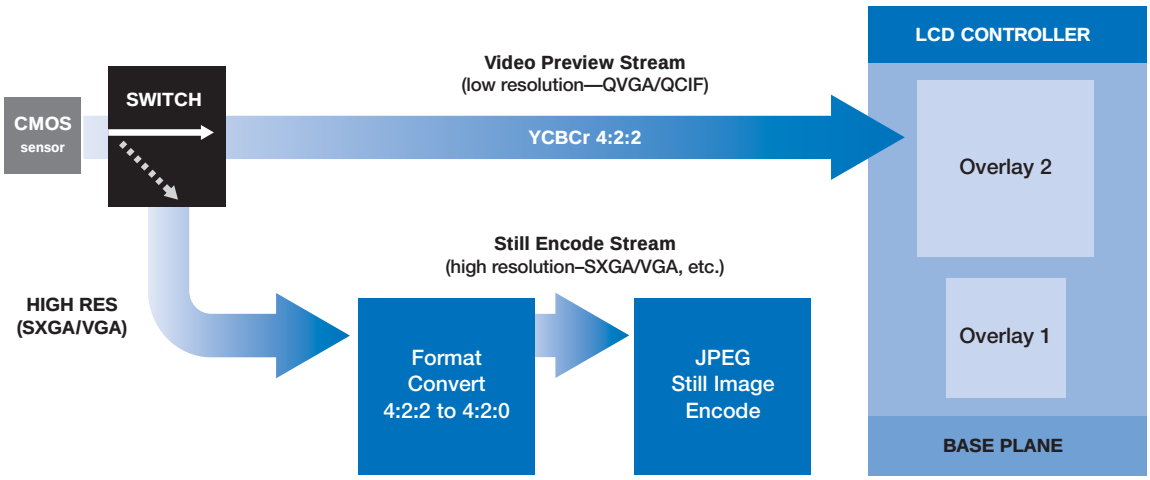


Figure 4. Illustration of YCbCr 4:2:2 and YCbCr 4:2:0



Intel® Wireless MMX™ Technology Routines (tentative, subject to change without notice)

Figure 5. Still capture scenario

Question 1A

If parallel, the Intel PXA27x processor supports 8-, 9-, and 10-bit formats.

- The Intel PXA27x processor supports the following parallel modes:
  - master parallel
  - embedded-parallel
  - slave parallel

If your sensor supports one of the three parallel modes listed above, mode compatibility should not be an issue. Proceed to question 2

Question 1B

If serial, the Intel PXA27x processor supports 4- or 5-bit data bus solutions.

- The Intel PXA27x processor supports the following serial modes:
  - master serial
  - embedded-serial (a master mode)

If your sensor supports one of the two serial modes listed above, mode compatibility should not be an issue. Proceed to question 2

Question 2

In what output format does your image sensor provide data?

If your image sensor provides pre-processed data go to question 2A, if it provides RAW, go to question 2B. If your sensor is capable of providing both pre-processed and RAW, you should read the options for both.

Question 2A

Pre-processed format?

- The Intel PXA27x processor can accept pre-processed data in the following formats:
  - RGB 8:8:8, includes support for un-packed and packed
  - RGB 6:6:6, includes support for un-packed and packed
  - RGB 5:6:5, packed
  - RGB 5:5:5, packed
  - RGB 4:4:4, packed
  - YCbCr 4:2:2, includes support for planer and interleaved

If your image sensor supports one of the pre-processed modes listed above, go to Section 8.0. The section describes in what order the Intel PXA27x processor expects the color components to be organized. If your sensor's output color component order matches what the Intel PXA27x processor expects, the Intel PXA27x processor should be able to capture its output correctly. If it does not, it may still be possible to interface your image sensor to the Intel PXA27x processor but re-formatting of the data via software may be required. Proceed to question 3.

Question 2B

Raw format?

If you have read section Section 6.3, "Using a RAW format" on page 8, you are already aware that the Intel Quick Capture interface is designed to be very flexible when used in the RAW capture mode. Because the Intel Quick Capture interface makes no assumptions about the format of that data, the Intel Quick Capture interface can accept data in most any raw format that can be transferred over an 8-, 9-,or 10-bit data interface. The key issue to remember is that the software running on the Intel PXA27x processor must be designed to understand the particular RAW format in which the sensor is providing data.

If your image sensor provides raw data output and you can support that format via software on the Intel PXA27x processor, data format should not be an issue. Proceed to question 3.

Question 3

What interface does your camera sensor require for sensor programming and control? The typical programming and control interface for images sensors is I^2C. The Intel PXA27x processor includes support for I^2C. If your sensor requires a different programming and control interface, you must determine if any of the other standard interfaces on the Intel PXA27x processor can be used.

If you have reviewed all of the questions and were able to find a match for mode, format, and control, the Intel Quick Capture interface will likely be able to work with your sensor. If you found there was an issue during your assessment, it might still be possible to use your sensor with the Intel PXA27x processor but the specific details required to work a solution may be outside the scope of this document. If you have questions or need assistance, please contact your Intel technical representative through http://premier.intel.com.

8.0 YCbCr and RGB Color Component Output Order

When evaluating a sensor, it is very important that you are aware that not all sensor manufacturers output YCbCr and RGB color components in the same order. The following tables show the preferred order for the Intel PXA27x processor to receive data for the most popular output formats. The tables are intended to be used along with your image sensor data sheet to determine if your sensor and the Intel PXA27x processor are both using the same format. Programming your sensor to output data as

DATA BUS	RGB 8:8:8 BYTE SEQUENCE					
CIF_DD[7]	R7(0)	G7(0)	B2(0)	R7(1)	G7(1)	B2(1)
CIF_DD[6]	R6(0)	G6(0)	B1(0)	R6(1)	G6(1)	B1(1)
CIF_DD[5]	R5(0)	G5(0)	B0(0)	R5(1)	G5(1)	B0(1)
CIF_DD[4]	R4(0)	G4(0)	B4(0)	R4(1)	G4(1)	B4(1)
CIF_DD[3]	R3(0)	G3(0)	B3(0)	R3(1)	G3(1)	B3(1)
CIF_DD[2]	R2(0)	G2(0)	B2(0)	R2(1)	G2(1)	B2(1)
CIF_DD[1]	R1(0)	G1(0)	B1(0)	R1(1)	G1(1)	B1(1)
CIF_DD[0]	R0(0)	G0(0)	B0(0)	R0(1)	G0(1)	B0(1)
Byte sequence	0	1	3	4	5	6
Pixel	Pixel 0			Pixel 1		

Table 5. 8-bit Data Capture Sequence for RGB 8:8:8

shown below will result in the most efficient design as no reformatting or re-organizing of the data via software is required.

Question:

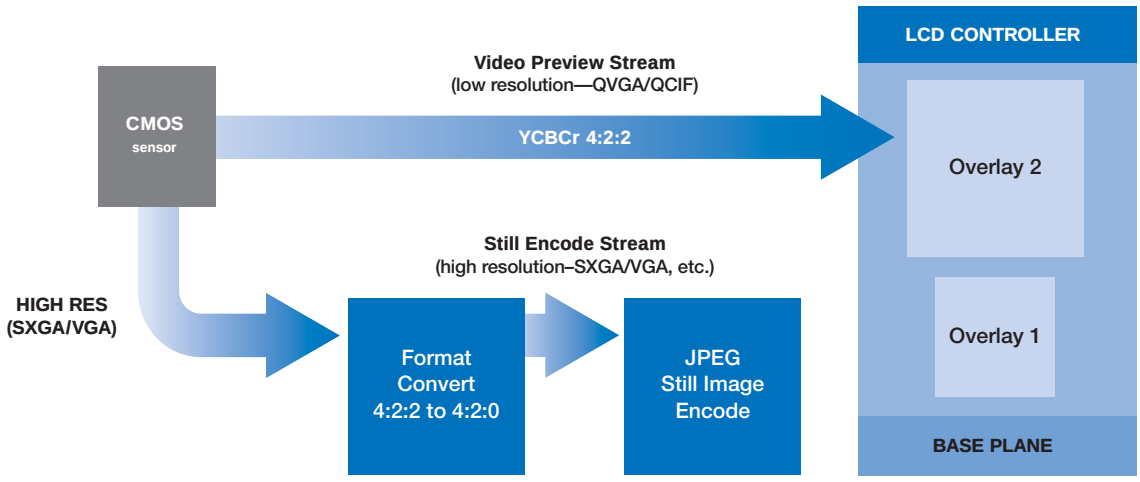
What if my sensor does not output packed RGB 5:6:5 data in format shown in Table 4?

Answer:

The Intel PXA27x camera interface can do color component reduction in hardware. Check to see if your sensor can output RGB 8:8:8. The Intel PXA27x can then convert the RGB 8:8:8 into RGB 5:6:5 as the data is received.

9.0 Example User Scenarios

The figures in this section are some additional user scenarios for mobile wireless devices.



Intel® Wireless MMX™ Technology Routines (tentative, subject to change without notice)

Figure 6. Video capture scenario

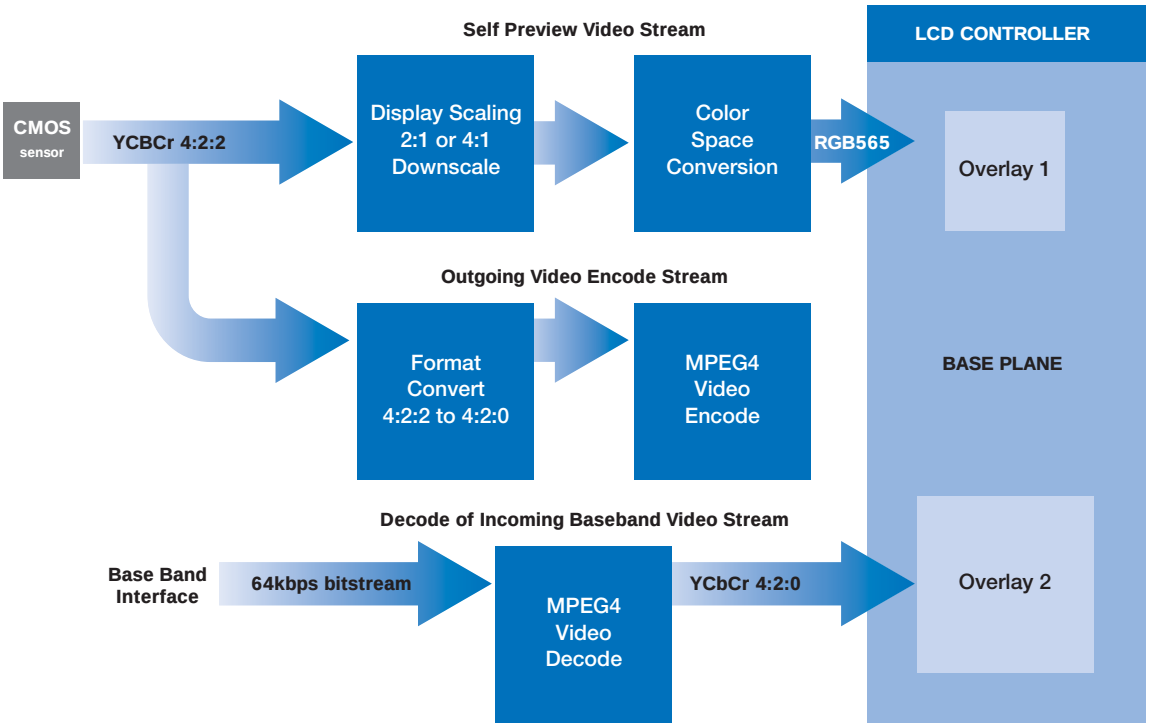


Figure 6. Video capture scenario

10.0 Interface Bandwidth and Frame Rate Estimation

This section provides an example on how to estimate the Intel Quick Capture interface bandwidth and maximum frame rate. These estimates are for a given resolution, color depth, and pixel clock. Achievable rates in an actual system will depend on such issues as system design, bus loading, and application performance. Please note that the intention of the examples here are to show you how to determine if a particular interface configuration is capable of providing the desired bandwidth or frame rate. For example, if you desire a frame rate of 25 fps but your calculation shows the configuration has a limit of 15 fps, your goal of 25 fps may not be achievable. Also, don't forget to note the frame rate limit of the image sensor you are considering. For example, some sensors have an internal limits of 15fps when the sensor is programmed for its maximum resolution.

CAMERA INTERFACE BANDWIDTH ESTIMATOR		
bit width of data interface	8	bits
number of bits clocked per cycle	8	bits/cycle
pixel clock frequency (cannot exceed 25 MHz)	25,000,000	cycles/sec
bits/cycle* cycles/sec	200,000,000	bits/sec
conversion from bits to bytes	25,000,000	bytes/sec
Maximum possible Interface bandwidth	**25.00**	**MB/sec**
Interface frame rate estimator		
color depth—programmable	16	bits/pixel
X resolution—programmable	640	pixels/frame
Y resolution—programmable	480	pixels/frame
total pixels in a frame = X resolution * Y resolution	307,200	pixels/frame
bits per frame = pixels per frame * bits per pixel =	4,915,200	bits/frame
conversion from bits to bytes	614,400	bytes/frame
Maximum interface frame rate for the resolution and color depth listed above	**40.00**	**frames/sec**

11.0 CCD Compatibility

Question:

Can I use a CCD type image sensor with the Intel PXA27x processor?

Answer:

CCD type sensor interface requirements can vary significantly so CCD compatibility must be determined on a sensor by sensor basis. However, some CCD image sensors have an interface that is very similar to the typical CMOS sensor. In this case typical means:

- Line valid and frame valid signals are used
- The sensor can provide data in YCbCr or RGB color space
- A similar number of data pins
- Uses I²C for control communication

One of the more critical issues to focus on is the clock timing. Some CCD sensors have complicated clock and timing requirements. Others can use a single clock input like the one typically used by CMOS sensors. If the CCD sensor you are considering provides similar interface signals and only requires a simple clock, it is likely the sensor can be used with the Intel PXA27x processor. Through the use of software, it may be possible to use GPIO pins to create a missing interface signal (if required) or for additional signals like a sensor power-down pin.

12.0 Conclusion

Intel Quick Capture Technology included in the Intel PXA27x processor provides components designed to make integrating image sensor technology into handheld devices easy and efficient.

1) The Intel Quick Capture interface provides a highly flexible physical interface.

2) The hardware color space conversion engine in overlay 2 of the LCD controller simplifies the requirements for generating a preview image.

3) Intel Wireless MMX Technology and Intel Integrated Performance Primitives provide optimized functions for performing image processing and accelerating application development.

For more information, visit the Intel Web site at: **developer.intel.com**