

## DS90UB964 Pattern Generation

### 1.1 Pattern Generation Types

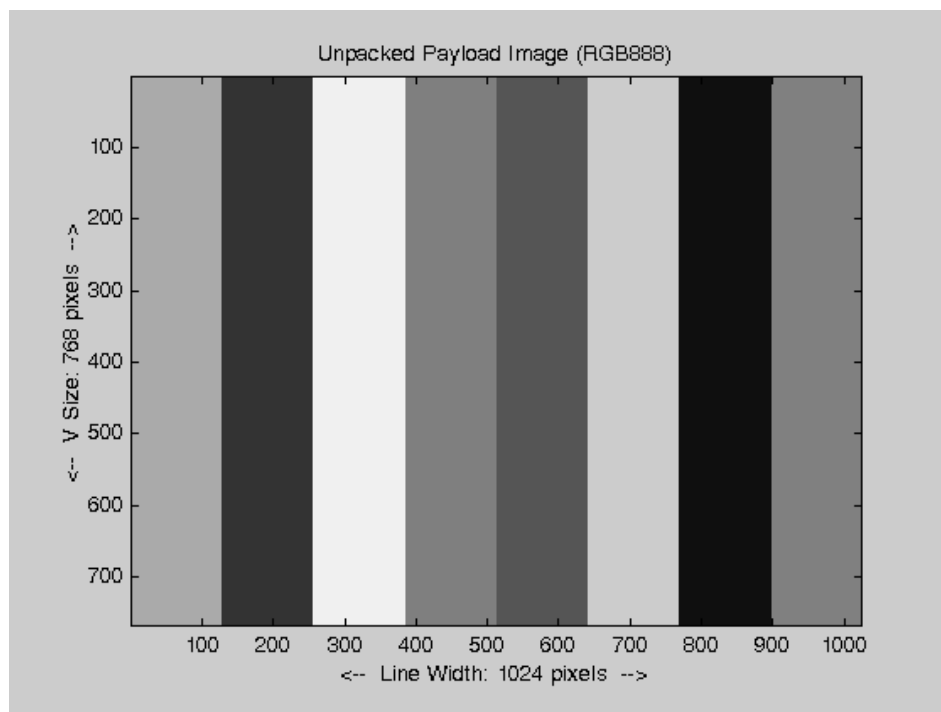
The DS90UB960 implements a Pattern Generator to provide a simple way to generate video test patterns for the CSI2 transmitter outputs. Two types of patterns are supported:

- Reference Color Bar pattern
- Fixed Color patterns

#### 1.1.1 Reference Color Bar Pattern

The Reference Color Bar Patterns are based on the pattern defined in Appendix D of the mipi\_CTS\_for\_D-PHY\_v1-1\_r03 specification. The pattern is an 8 color bar pattern designed to provide high, low, and medium frequency outputs on the CSI2 transmit data lanes.

**Figure 1. Example MIPI D-PHY test pattern**



The DSI Reference pattern provides 8 color bars by default with the following byte data for the color bars:

- X bytes of 0xAA (high-frequency pattern, inverted)
- X bytes of 0x33 (mid-frequency pattern)
- X bytes of 0xF0 (low-frequency pattern, inverted)
- X bytes of 0x7F (lone 0 pattern)
- X bytes of 0x55 (high-frequency pattern)
- X bytes of 0xCC (mid-frequency pattern, inverted)
- X bytes of 0x0F (low-frequency pattern)

Y bytes of 0x80 (lone 1 pattern)

In most cases, Y will be the same as X. For certain data types, the last color bar may need to be larger than the others to properly fill the video line dimensions.

The Pattern Generator is programmable with the following options:

- Number of color bars (1, 2, 4, or 8)
- Number of bytes per line
- Number of bytes per color bar
- CSI DataType field and VC-ID
- Number of active video lines per frame
- Number of total lines per frame (active plus blanking)
- line period (possibly program in units of 10ns)
- vertical front porch – number of blank lines prior to FrameEnd packet
- vertical back porch – number of blank lines following FrameStart packet

The pattern generator relies on proper programming by software to ensure the color bar widths are set to multiples of the block (or word) size required for the specified DataType. For example, for RGB888, the block size is 3 bytes which also matches the pixel size. In this case, the number of bytes per color bar must be a multiple of 3.

The Pattern Generator is implemented in the CSI Transmit clock domain, providing the pattern directly to the CSI Transmitter. The circuit generates the CSI formatted data.

## 1.2 Fixed Color Patterns

When programmed for Fixed Color Pattern mode, Pattern Generator can generate a video image with a programmable fixed data pattern. The basic programming fields for image dimensions are the same as used with the Color Bar Patterns. When sending Fixed Color Patterns, the color bar controls allow alternating between the fixed pattern data and the bit-wise inverse of the fixed pattern data.

The Fixed Color patterns assume a fixed block size for the byte pattern to be sent. The block size is programmable via register and is designed to support most 8-bit, 10-bit, and 12-bit pixel formats. The block size should be set based on the pixel size converted to blocks that are an integer multiple of bytes. For example, an RGB888 pattern would consist of 3-byte pixels and therefore require a 3-byte block size. A 2x12-bit pixel image would also require 3-byte block size, while a 3x12-bit pixel image would require 9 bytes (2 pixels) in order to send an integer number of bytes. Sending a RAW10 pattern typically requires a 5-byte block size for 4 pixels, so 1x10-bit and 2x10-bit could both be sent with a 5-byte block size. For 3x10-bit, a 15-byte block size would be required.

The Fixed Color patterns support block sizes up to 16 bytes in length, allowing additional options for patterns in some conditions. For example, an RGB888 image could alternate between four different pixels by using a twelve-byte block size. An alternating black/white RGB888 image could be sent with a block size of 6-bytes and setting first three bytes to 0xFF and next three bytes to 0x00.

To support up to 16-byte block sizes, a set of sixteen registers are implemented to allow programming the value for each data byte.

## 1.3 Packet Generator Programming

The following information provides details on how to program the Pattern Generator to provide a specific color bar pattern, based on datatype, frame size, and line size.

Most basic configuration information is determined directly from the expected video frame parameters. The requirements should include the datatype, frame rate (frames per second), number of active lines per frame, number of total lines per frame (active plus blanking), and number of pixels per line.

- PGEN\_ACT\_LPF – Number of active lines per frame
- PGEN\_TOT\_LPF – Number of total lines per frame
- PGEN\_LSIZE – Video line length size in bytes. Compute based on pixels per line multiplied by pixel size in bytes
- Optional: PGEN\_VBP – Vertical back porch. This is the number of lines of vertical blanking following VSync
- Optional: PGEN\_VFP – Vertical front porch. This is the number of lines of vertical blanking preceding VSync
- PGEN\_LINE\_PD – Line period in 10ns units. Compute based on Frame Rate and total lines per frame
- PGEN\_BAR\_SIZE – Color bar size in bytes. Compute based on datatype and line length in bytes (see details below)

Prior to enabling the Packet Generator, the following should be done:

1. Disable video forwarding by setting the upper 4 bits of the FWD\_CTL1 register (i.e. set register 0x20 to 0xF0).
2. Configure CSI Transmitter operating speed using the CSI\_PLL\_CTL register.
3. Enable the CSI Transmitter for port 0 using the CSI\_CTL register.

### 1.3.1 Determining Color Bar Size

The color bar pattern should be programmed in units of a block or word size dependent on the datatype of the video being sent. The sizes are defined in the Mipi CSI-2 specification. For example, RGB888 requires a 3-byte block size which is the same as the pixel size. RAW10 requires a 5-byte block size which is equal to 4 pixels. RAW12 requires a 3-byte block size which is equal to 2 pixels.

When programming the Pattern Generator, software should compute the required bar size in bytes based on the line size and the number of bars. For the standard 8 color bar pattern, that would require the following algorithm:

1. Select the desired datatype, and a valid length for that datatype (in pixels).
2. Convert pixels/line to blocks/line (by dividing by the number of pixels/block, as defined in the datatype specification).
3. Divide the blocks/line result by the number of color bars (8), giving blocks/bar
4. Round result down to the nearest integer
5. Convert blocks/bar to bytes/bar and program that value into the PGEN\_BAR\_SIZE register

As an alternative, the blocks/line can be computed by converting pixels/line to bytes/line and divide by bytes/block.

### 1.4 Default Color Bar Pattern

The default color bar pattern for the Pattern Generator is a 640x480 RGB888 pattern at 60 Hz. The following computations were used to define the default register settings:

Data:

Hactive : 640 pixels

Vactive : 480 lines

Vtotal : 525 lines

Vfront : 10 lines

Vback : 33 lines

Pixel\_size : 3 bytes

Block\_size : 3 bytes

Default Values:

PGEN\_ACT\_LPF : Set to Vactive = 480 (0x01E0)

PGEN\_TOT\_LPF : Set to Vtotal = 525 (0x020D)

PGEN\_LSIZE : Set to Hactive \* Pixel\_size = 640 \* 3 = 1920 (0x0780)

PGEN\_VBP : Set to Vback = 33 (0x21)

PGEN\_VFP : Set to Vactive = 10 (0x0A)

PGEN\_LINE\_PD : Set to (1sec)/(FrameRate\*Vtotal\*10ns) = 3175 (0x0C67)

PGEN\_BAR\_SIZE : Set to ((Hactive\*Pixel\_size/Block\_size) DIV 8) \* Block\_size = 80\*3 = 240 (0xF0)

## 1.5 Default Fixed Color Pattern

The default settings could also be used to send a fixed color pattern, using the same video parameters (640x480 RGB888 pattern at 60 Hz). This would require setting the PGEN\_FIXED\_EN bit in the PGEN\_CFG register. In addition, the first three PGEN\_COLORx registers could be set to send a specific color pattern.

For example, to send a blue screen, program the PGEN\_COLOR0/1/2 registers to 0xFF, 0x00, and 0x00 respectively. To send green, program to 0x00, 0xFF, 0x00. To send red, program to 0x00, 0x00, 0xFF.

With the default setting of eight color bars, the pattern will alternate between the fixed value and it's inverse. For example the default will alternate between color bars of {0xAA, 0x33, 0xF0} and {0x55, 0xCC, 0x0F} for colors {B, G, R}.