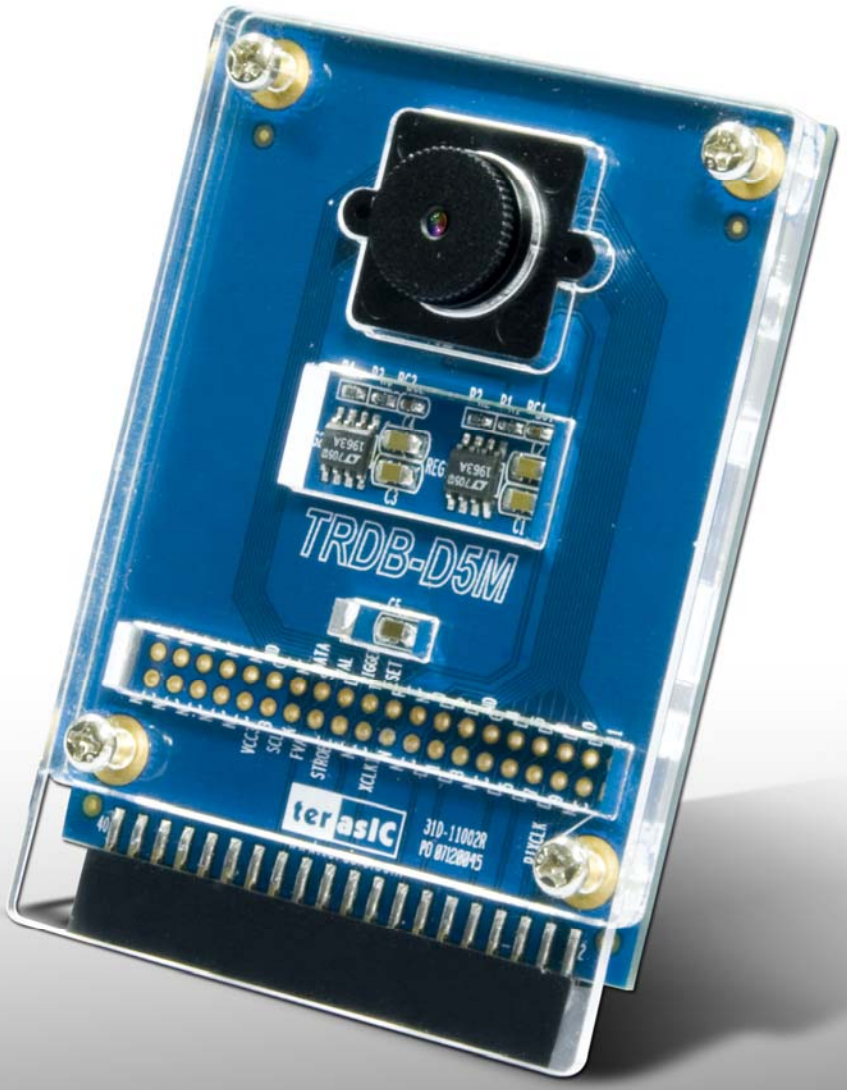
www.terasic.com

THDB-D5M

Terasic D5M Hardware specification



Document Version 0.1 APR. 18, 2008 by Terasic

Page Index

CHAPTER 1

Pixel Data Format

This chapter describes the output data format for D5M.

Pixel Array Structure

The D5M pixel array consists of a 2,752-column by 2,004-row matrix of pixels addressed by column and row. The address (column 0, row 0) represents the upper-right corner of the entire array, looking at the sensor as shown in Figure 1.1.

The array consists of a 2,592-column by 1,944-row active region in the center representing the default output image, surrounded by a boundary region (also active), surrounded by a border of dark pixels (see Table 1.1 and Table 1.2). The boundary region can be used to avoid edge effects when doing color processing to achieve a 2,592 x 1,944 result image, while the optically black column and rows can be used to monitor the black level.

Pixels are output in a Bayer pattern format consisting of four “colors”—Green1, Green2, Red, and Blue (G1, G2, R, B)—representing three filter colors. When no mirror modes are enabled, the first row output alternates between G1 and R pixels, and the second row output alternates between B and G2 pixels. The Green1 and Green2 pixels have the same color filter, but they are treated as separate colors by the data path and analog signal chain.

Table 1.1 Pixel Type by Column

Column	Pixel Type
0–9	Dark (10)
10–15	Active boundary (6)
16–2,607	Active image (2592)
2,608–2,617	Active boundary (10)
2,618–2,751	Dark (134)

Table 1.2 Pixel Type by Row

Row	Pixel Type
0–49	Dark (50)
50–53	Active boundary (4)
54–1997	Active image (1944)
1,998–2,001	Active boundary (4)
2,002–2003	Dark (2)

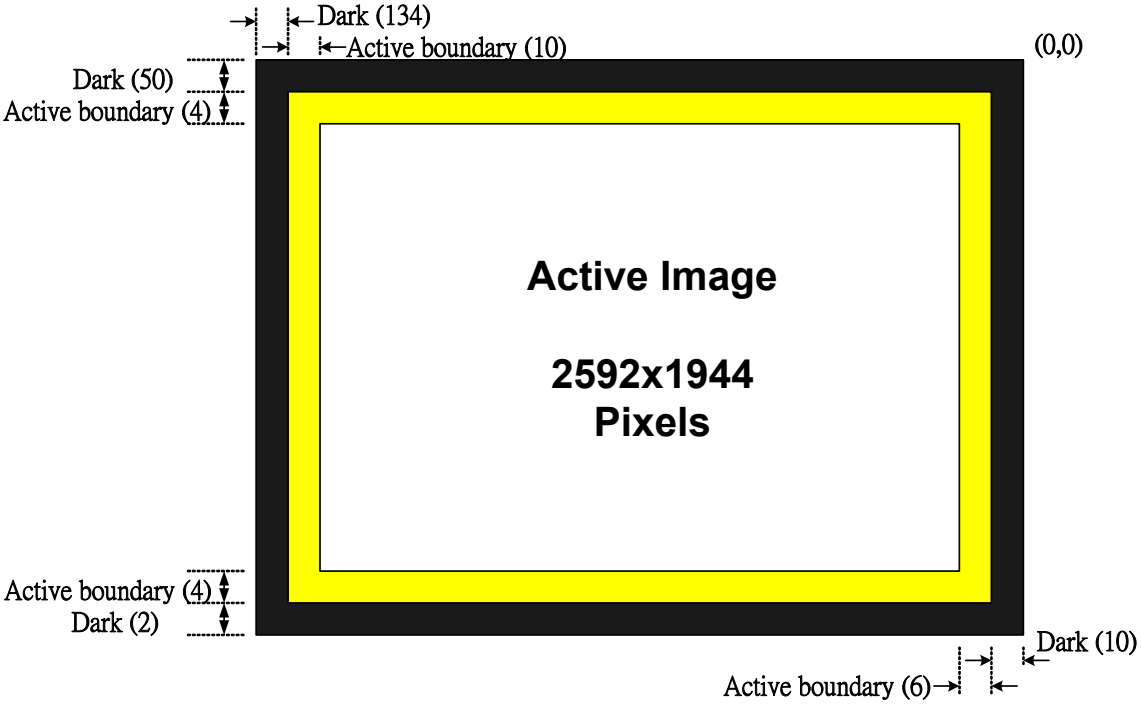


Figure 1.1 Pixel Array Description

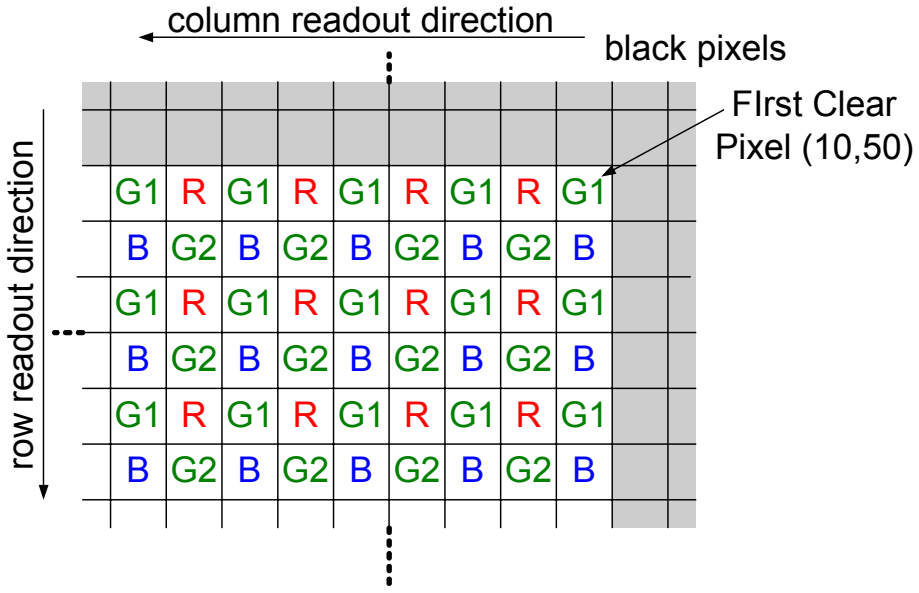


Figure 1.2 Pixel Color Pattern Detail (Top Right Corner)

Default Readout Order

By convention, the sensor core pixel array is shown with pixel (0,0) in the top right corner (see Figure 1.1). This reflects the actual layout of the array on the die. Also, the first pixel data read out of the sensor in default condition is that of pixel (16,54).

When the sensor is imaging, the active surface of the sensor faces the scene as shown in Figure 1.3. When the image is read out of the sensor, it is read one row at a time, with the rows and columns sequenced as shown in Figure 1.2.

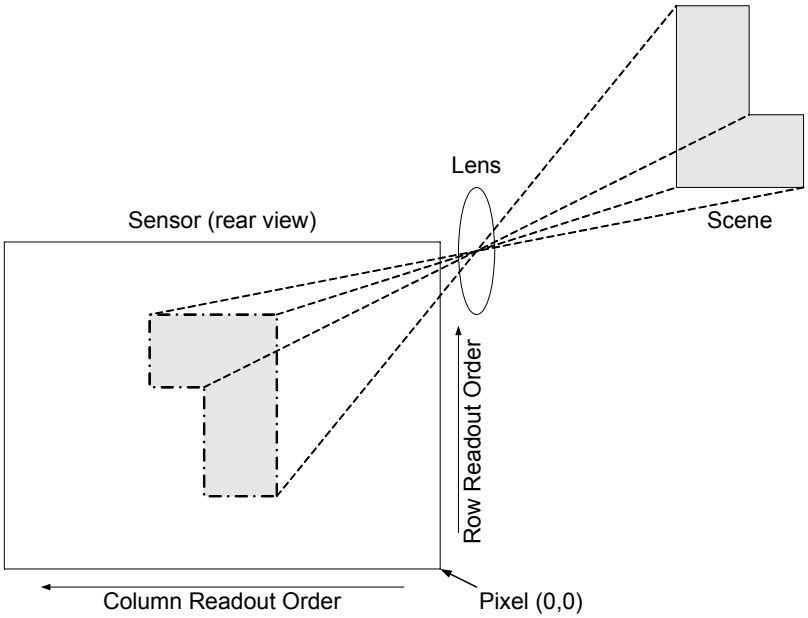


Figure 1.3 Imaging a Scene

Output Data Format (Default Mode)

The D5M image data is read out in a progressive scan. Valid image data is surrounded by horizontal blanking and vertical blanking, as shown in Figure 1.4. LINE_VALID is HIGH during the shaded region of the figure. FRAME_VALID timing is described in “Output Data Timing” on page 5.

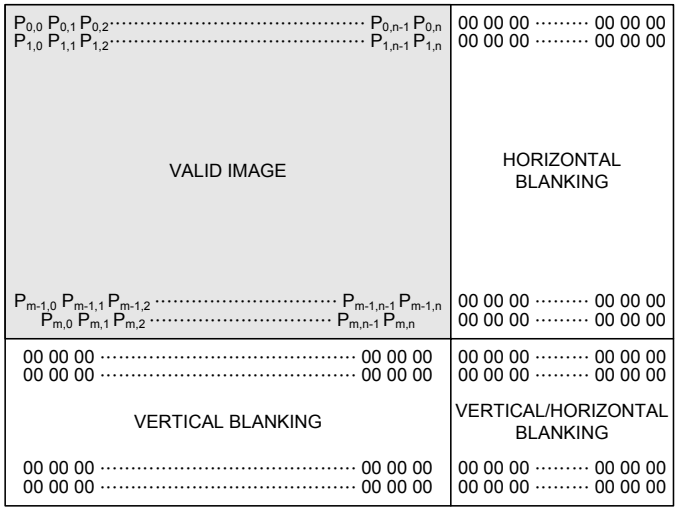


Figure 1.4 Spatial Illustration of Image Readout

Readout Sequence

Typically, the readout window is set to a region including only active pixels. The user has the option of reading out dark regions of the array, but if this is done, consideration must be given to how the sensor reads the dark regions for its own purposes.

Rows are read from the array in the following order:

1. Dark rows: If Show_Dark_Rows is set, or if Manual_BLC is clear, dark rows on the top of the array is read out. The set of rows sampled are adjusted based on the Row_Bin setting such that there are 8 rows after binning, as shown in the Table 1.3.

The Row_Skip setting is ignored for the dark row region.

If Show_Dark_Rows is clear and Manual_BLC is set, no dark rows are read from the array as part of this step, allowing all rows to be part of the active image. This does not change the frame time, as HDR is included in the vertical blank period.

2. Active image: The rows defined by the row start, row size, bin, skip, and row mirror settings are read out. If this set of rows includes rows read out above, those rows are resampled, meaning that the data is invalid.

Table 1.3 Dark Rows Sampled as a Function of Row_Bin

Row_Bin	HDR (Dark rows after binning)
0	8
1	8
3	8

Columns are read out in the following order:

1. Dark columns: If either Show_Dark_Columns or Row_BLC is set, dark columns on the left side of the image is read out followed by those on the right side. The set of columns read is shown in Table 1.4.

The Column_Skip setting is ignored for the dark columns.

If neither Show_Dark_Columns nor Row_BLC is set, no dark columns are read, allowing all columns to be part of the active image. This does not change the row time, as WDC is included in the vertical blank period.

2. Active image: The columns defined by column start, column size, bin, skip, and column mirror settings are read out. If this set of columns includes the columns read out above, these columns are resampled, meaning the data is invalid.

Table 1.4 Dark Columns Sampled as a Function of Column_Bin

Column_Bin	WDC (Dark columns after binning)
0	80
1	40
3	20

Output Data Timing

The output images are divided into frames, which are further divided into lines. By default, the sensor produces 1,944 rows of 2,592 columns each. The FRAME_VALID and LINE_VALID signals indicate the boundaries between frames and lines, respectively.
PIXCLK can be used as a clock to latch the data. For each PIXCLK cycle, one 12-bit pixel datum outputs on the DOUT pins. When both FRAME_VALID and LINE_VALID are asserted, the pixel is valid. PIXCLK cycles that occur when FRAME_VALID is negated are called vertical blanking. PIXCLK cycles that occur when only LINE_VALID is negated are called horizontal blanking.

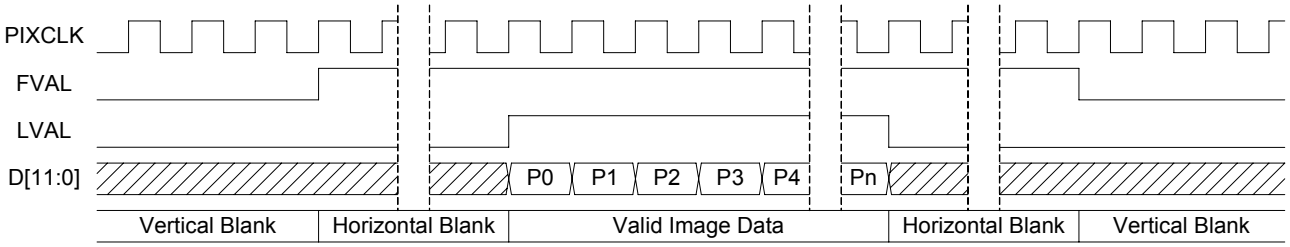


Figure 1.5 Default Pixel Output Timing

LINE_VALID and FRAME_VALID

The timing of the FRAME_VALID and LINE_VALID outputs is closely related to the row time and the frame time.

The FRAME_VALID pin will be asserted for an integral number of row times, which will normally be equal to the height of the output image. If Show_Dark_Rows is set, the dark sample rows will be output before the active image, and FRAME_VALID will be extended to include them. In this case, FRAME_VALID 's leading edge happens at time 0.

The LINE_VALID pin will be asserted during the valid pixels of each row. The leading edge of LINE_VALID will be offset from the leading edge of FRAME_VALID by 609 pixclks. If Show_Dark_Columns is set, the dark columns will be output before the image pixels, and LINE_VALID will be extended back to include them; in this case, the first pixel of the active image still occurs at the same position relative to the leading edge of FRAME_VALID. Normally, LINE_VALID will only be asserted if FRAME_VALID is asserted; this is configurable as described below.

LINE_VALID Format Options

The default situation is for LINE_VALID to be negated when FRAME_VALID is negated. The other option available is shown in Figure 1.6. If Continuous_LINE_VALID is set, LINE_VALID is asserted even when FRAME_VALID is not, with the same period and duty cycle. If XOR_Line_Valid is set, but not Continuous_Line_Valid, the resulting LV will be the XOR of FV and the continuous LV.

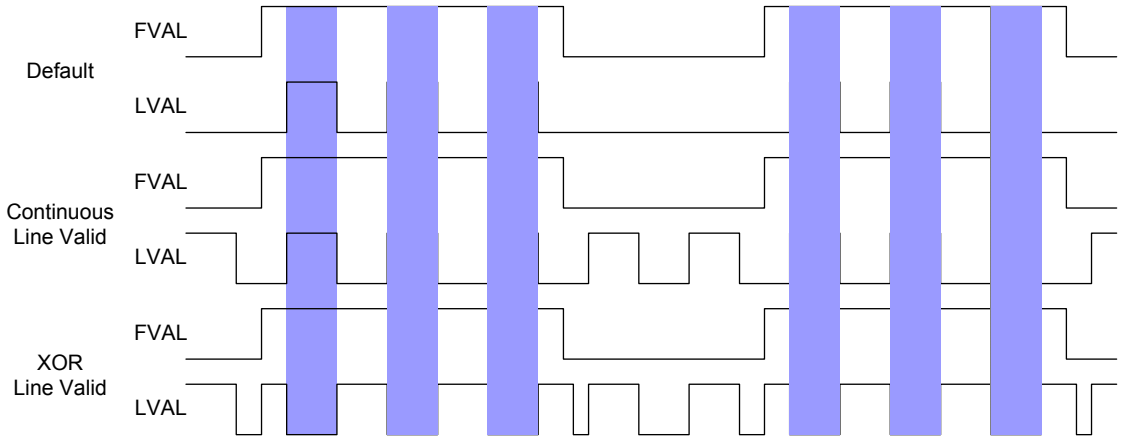


Figure 1.6 LINE_VALID Format Options

Frame Time

The pixel clock (PIXCLK) represents the time needed to sample 1 pixel from the array, and is typically equal to 1 EXTCLK period. The sensor outputs data at the maximum rate of one pixel per PIXCLK. One row time (^{t}ROW) is the period from the first pixel output in a row to the first pixel output in the next row. The row time and frame time are defined by equations in Table 1.5 on page 7.

The timing of an entire frame is shown in Figure 1.7.

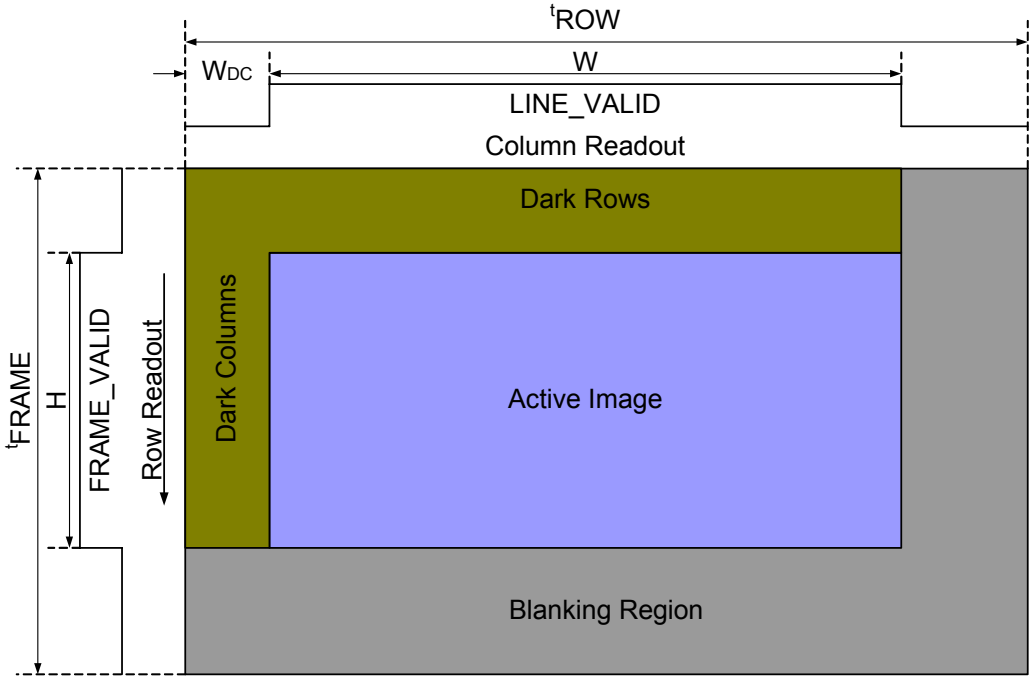


Figure 1.7 Frame Timing

Table 1.5 Frame Time

Parameter	Name	Equation	Default Timing at EXTCLK = 96 MHz
fps	Frame Rate	1/tFRAME	15
^{t}FRAME	Frame Time	(H + max(VB, VBMIN)) × tROW	66ms
^{t}ROW	Row Time	2 × tPIXCLK x max(((W/2) + max(HB, HBMIN)), (41 + 208 x (Row_Bin+1) + 99))	33.5µs
W	Output Image Width	2 × ceil((Column_Size + 1) / (2 × (Column_Skip + 1)))	2592 PIXCLK
H	Output Image Height	2 × ceil((Row_Size + 1) / (2 × (Row_Skip + 1)))	1944 rows
SW	Shutter Width	max(1, (2*16 × Shutter_Width_Upper) + Shutter_Width_Lower)	1943 rows
HB	Horizontal Blanking	Horizontal_Blank + 1	1 PIXCLK
VB	Vertical Blanking	Vertical_Blank + 1	26 rows
HBMIN	Minimum Horizontal Blanking	208 × (Row_Bin + 1) + 64 + (WDC/2)	312 PIXCLK
VBMIN	Minimum Vertical Blanking	max(8, SW - H) + 1	9 rows
^{t}PIXCLK	Pixclk Period	1/fPIXCLK	10.42ns

The minimum horizontal blanking (HB_{MIN}) values for various Row_Bin and Column_Bin settings are shown in Table 1.6.

Table 1.6 HB_{MIN} Values for Row_bin vs. Column_bin Settings

	Column_bin (WDC)			
Row_bin		**0**	**1**	**3**
	0	312	292	282
	1	520	500	490
	3	936	916	906

Frame Rates at Common Resolutions

Table 1.7 and Table 1.8 show examples of register settings to achieve common resolutions and their frame rates. Frame rates are shown both with subsampling enabled and disabled.

Table 1.7 Standard Resolutions

Resolution	Frame Rate	Sub-sampling Mode	Column_Size (R0x04)	Row_Size (R0x03)	Shutter_Width_Lower (R0x09)	Row_Bin (R0x22 [5:4])	Row_Skip (R0x22 [2:0])	Column_Bin (R0x23 [5:4])	Column_Skip (R0x23 [2:0])
2592 x 1944 (Full Resolution)	15.15	N/A	2591	1943	<1943	0	0	0	0
2,048 x 1,536 QXGA	23	N/A	2047	1535	<1535	0	0	0	0
1,600 x 1,200 UXGA	35.2	N/A	1599	1199	<1199	0	0	0	0
1,280 x 1,024 SXGA	48	N/A	1279	1023	<1023	0	0	0	0
	48	skipping	2559	2047		0	1	0	1
	40.1	binning	2559	2047		1	1	1	1
1,024 x 768 XGA	73.4	N/A	1023	767	<767	0	0	0	0
	73.4	skipping	2047	1535		0	1	0	1
	59.7	binning	2047	1535		1	1	1	1
800 x 600 VGA	107.7	N/A	799	599	<599	0	0	0	0
	107.7	skipping	1599	1199		0	1	0	1
	85.2	binning	1599	1199		1	1	1	1
640 x 480 VGA	150	N/A	639	479	<479	0	0	0	0
	150	skipping	2559	1919		0	3	0	3
	77.4	binning	2559	1919	3	3	3	3	3

Table 1.8 Wide Screen (16:9) Resolutions

Resolution	Frame Rate	Sub-sampling Mode	Column_Size (R0x04)	Row_Size (R0x03)	Shutter_Width_Lower (R0x09)	Row_Bin (R0x22 [5:4])	Row_Skip (R0x22 [2:0])	Column_Bin (R0x23 [5:4])	Column_Skip (R0x23 [2:0])
1,920 x 1,080 HDTV	34.1	N/A	1919	1079	<1079	0	0	0	0
1,280 x 720 HDTV	67.6	N/A	1279	719	<719	0	0	0	0
	67.6	skipping	2559	1439	<719	0	1	0	1
	56.4	binning	2559	1439	<719	1	1	1	1

- **It is assumed that the minimum horizontal blanking and the minimum vertical blanking conditions are met, and that all other registers are set to default values.**

CHAPTER 2 Registers

This chapter lists the register map and detailed instruction for D5M.

Register List and Default Values

Tabl 2.1 lists D5M sensor registers and their default values.

Tabl 2.1 Register List and Default Values

1 = read-only，always 1； 0 = read-only，always 0； d = programmable； ? = read-only，dynamic

Register # Dec (Hex)	Register Description	Data Format (Binary)	Default Value Dec(Hex)
R0:0(R0x000)	Chip Version	???? ???? ???? ????	6145 (0x1801)
R1:0(R0x001)	Row Start	0000 0ddd dddd dddd	54 (0x0036)
R2:0(R0x002)	Column Star	0000 dddd dddd dddd	16 (0x0010)
R3:0(R0x003)	Row Size	0000 0ddd dddd dddd	1943 (0x0797)
R4:0(R0x004)	Column Size	0000 dddd dddd dddd	2591 (0x0A1F)
R5:0(R0x005)	Horizontal Blank	0000 dddd dddd dddd	0 (0x0000)
R6:0(R0x006)	Vertical Blank	0000 0ddd dddd dddd	25 (0x0019)
R7:0(R0x007)	Output Control	0d0d dddd dddd dddd	8066 (0x1F82)
R8:0(R0x008)	Shutter Width Upper	0000 0000 0000 dddd	0 (0x0000)
R9:0(R0x009)	Shutter Width Lower	dddd dddd dddd dddd	1943 (0x0797)
R10:0(R0x00A)	Pixel Clock Control	d000 0ddd 0ddd dddd	0 (0x0000)
R11:0(R0x00B)	Restart	0000 0000 0000 0ddd	0 (0x0000)
R12:0(R0x00C)	Shutter Delay	000d dddd dddd dddd	0 (0x0000)
R13:0(R0x00D)	Reset	0000 0000 0000 000d	0 (0x0000)
R15:0(R0x00F)	Reserved	-	0 (0x0000)
R16:0(R0x010)	PLL Control	ddd0 000d dddd 00dd	80 (0x0050)
R17:0(R0x011)	PLL Config 1	dddd dddd 00dd dddd	25604 (0x6404)
R18:0(R0x012)	PLL Config 2	000d dddd 000d dddd	0 (0x0000)
R20:0(R0x014)	Reserved	-	54 (0x0036)
R21:0(R0x015)	Reserved	-	16 (0x0010)
R30:0(R0x01E)	Read Mode 1	0ddd dddd dddd dddd	16390 (0x4006

Tabl 2.1 Register List and Default Values

1 = read-only，always 1； 0 = read-only，always 0； d = programmable； ? = read-only，dynamic

Register # Dec (Hex)	Register Description	Data Format (Binary)	Default Value Dec(Hex)
R32:0(R0x020)	Read Mode 2	dddd d000 0ddd 00d0	64 (0x0040)
R34:0(R0x022)	Row Address Mode	0ddd 0ddd 00dd 0ddd	0 (0x0000)
R35:0(R0x023)	Column Address Mode	0000 0ddd 00dd 0ddd	0 (0x0000)
R36:0(R0x024)	Reserved	-	2 (0x0002)
R39:0(R0x027)	Reserved	-	11 (0x000B)
R41:0(R0x029)	Reserved	-	1153 (0x0481)
R42:0(R0x02A)	Reserved	-	4230 (0x1086)
R43:0(R0x02B)	Green1 Gain	0ddd dddd dddd dddd	8 (0x0008)
R44:0(R0x02C)	Blue Gain	0ddd dddd dddd dddd	8 (0x0008)
R45:0(R0x02D)	Red Gain	0ddd dddd dddd dddd	8 (0x0008)
R46:0(R0x02E)	Green2 Gain	0ddd dddd dddd dddd	8 (0x0008)
R48:0(R0x030)	Reserved	-	0 (0x0000)
R50:0(R0x032)	Reserved	-	0 (0x0000)
R53:0(R0x035)	Global Gain	dddd dddd dddd dddd	8 (0x0008)
R60:0(R0x03C)	Reserved	-	4112 (0x1010)
R61:0(R0x03D)	Reserved	-	5 (0x0005)
R62:0(R0x03E)	Reserved	-	32967 (0x80C7)
R63:0(R0x03F)	Reserved	-	4 (0x0004)
R64:0(R0x040)	Reserved	-	7 (0x0007)
R65:0(R0x041)	Reserved	-	0 (0x0000)
R66:0(R0x042)	Reserved	-	3 (0x0003)
R67:0(R0x043)	Reserved	-	3 (0x0003)
R68:0(R0x044)	Reserved	-	515 (0x0203)
R69:0(R0x045)	Reserved	-	4112 (0x1010)
R70:0(R0x046)	Reserved	-	4112 (0x1010)
R71:0(R0x047)	Reserved	-	4112 (0x1010)
R72:0(R0x048)	Reserved	-	16 (0x0010)
R73:0(R0x049)	Row Black Target	0000 dddd dddd dddd	168 (0x00A8)
R74:0(R0x04A)	Reserved	-	16 (0x0010)
R75:0(R0x04B)	Row Black Default Offset	0000 dddd dddd dddd	40 (0x0028)
R76:0(R0x04C)	Reserved	-	16 (0x0010)
R77:0(R0x04D)	Reserved	-	8224 (0x2020)
R78:0(R0x04E)	Reserved	-	4112 (0x1010)
R79:0(R0x04F)	Reserved	-	20 (0x0014)
R80:0(R0x050)	Reserved	-	32768 (0x8000)

Table 2.1 Register List and Default Values

1 = read-only，always 1； 0 = read-only，always 0； d = programmable； ? = read-only，dynamic

Register # Dec (Hex)	Register Description	Data Format (Binary)	Default Value Dec(Hex)
R81:0(R0x051)	Reserved	-	7 (0x0007)
R82:0(R0x052)	Reserved	-	32768 (0x8000)
R83:0(R0x053)	Reserved	-	7 (0x0007)
R84:0(R0x054)	Reserved	-	8 (0x0008)
R86:0(R0x056)	Reserved	-	32 (0x0020)
R87:0(R0x057)	Reserved	-	4 (0x0004)
R88:0(R0x058)	Reserved	-	32768 (0x8000)
R89:0(R0x059)	Reserved	-	7 (0x0007)
R90:0(R0x05A)	Reserved	-	4 (0x0004)
R91:0(R0x05B)	BLC_Sample_Size	0000 0000 0000 000d	1 (0x0001)
R92:0(R0x05C)	BLC_Tune_1	0000 dddd dddd dddd	90 (0x005A)
R93:0(R0x05D)	BLC_Delta_Thresholds	0ddd dddd 0ddd dddd	11539 (0x2D13)
R94:0(R0x05E)	BLC_Tune_2	0ddd 000d dddd dddd	16895 (0x41FF)
R95:0(R0x05F)	BLC_Target_Thresholds	0ddd dddd 0ddd dddd	8989 (0x231D)
R96:0(R0x060)	Green1_Offset	0000 000d dddd dddd	32 (0x0020)
R97:0(R0x061)	Green2_Offset	0000 000d dddd dddd	32 (0x0020)
R98:0(R0x062)	Black_Level_Calibration	dddd d000 0000 00dd	0 (0x0000)
R99:0(R0x063)	Red_Offset	0000 000d dddd dddd	32 (0x0020)
R100:0(R0x064)	Blue_Offset	0000 000d dddd dddd	32 (0x0020)
R101:0(R0x065)	Reserved	-	0 (0x0000)
R104:0(R0x068)	Reserved	-	0 (0x0000)
R105:0(R0x069)	Reserved	-	0 (0x0000)
R106:0(R0x06A)	Reserved	-	0 (0x0000)
R107:0(R0x06B)	Reserved	-	0 (0x0000)
R108:0(R0x06C)	Reserved	-	0 (0x0000)
R109:0(R0x06D)	Reserved	-	0 (0x0000)
R112:0(R0x070)	Reserved	-	103 (0x0067)
R113:0(R0x071)	Reserved	-	42752 (0xA700)
R114:0(R0x072)	Reserved	-	42752 (0xA700)
R115:0(R0x073)	Reserved	-	3072 (0x0C00)
R116:0(R0x074)	Reserved	-	1536 (0x0600)
R117:0(R0x075)	Reserved	-	22039 (0x5617)
R118:0(R0x076)	Reserved	-	27509 (0x6B75)
R119:0(R0x077)	Reserved	-	27509 (0x6B75)
R120:0(R0x078)	Reserved	-	42240 (0xA500)

Table 2.1 Register List and Default Values

1 = read-only，always 1； 0 = read-only，always 0； d = programmable； ? = read-only，dynamic

Register # Dec (Hex)	Register Description	Data Format (Binary)	Default Value Dec(Hex)
R121:0(R0x079)	Reserved	-	43776 (0xAB00)
R122:0(R0x07A)	Reserved	-	43268 (0xA904)
R123:0(R0x07B)	Reserved	-	42761 (0xA709)
R124:0(R0x07C)	Reserved	-	42752 (0xA700)
R125:0(R0x07D)	Reserved	-	65280 (0xFF00)
R126:0(R0x07E)	Reserved	-	43264 (0xA900)
R127:0(R0x07F)	Reserved	-	43264 (0xA900)
R128:0(R0x080)	Reserved	-	34 (0x0022)
R129:0(R0x081)	Reserved	-	7940 (0x1F04)
R130:0(R0x082)	Reserved	-	0 (0x0000)
R131:0(R0x083)	Reserved	-	6918 (0x1B06)
R132:0(R0x084)	Reserved	-	7432 (0x1D08)
R134:0(R0x086)	Reserved	-	6150 (0x1806)
R135:0(R0x087)	Reserved	-	6664 (0x1A08)
R144:0(R0x090)	Reserved	-	2000 (0x07D0)
R145:0(R0x091)	Reserved	-	0 (0x0000)
R146:0(R0x092)	Reserved	-	1 (0x0001)
R147:0(R0x093)	Reserved	-	0 (0x0000)
R149:0(R0x095)	Reserved	-	0 (0x0000)
R150:0(R0x096)	Reserved	-	0 (0x0000)
R151:0(R0x097)	Reserved	-	0 (0x0000)
R152:0(R0x098)	Reserved	-	0 (0x0000)
R153:0(R0x099)	Reserved	-	0 (0x0000)
R154:0(R0x09A)	Reserved	-	0 (0x0000)
R155:0(R0x09B)	Reserved	-	0 (0x0000)
R156:0(R0x09C)	Reserved	-	0 (0x0000)
R160:0(R0x0A0)	Test_Pattern_Control	-	0 (0x0000)
R161:0(R0x0A1)	Test_Pattern_Green	-	0 (0x0000)
R162:0(R0x0A2)	Test_Pattern_Red	-	0 (0x0000)
R163:0(R0x0A3)	Test_Pattern_Blue	-	0 (0x0000)
R164:0(R0x0A4)	Test_Pattern_Bar_Width	-	0 (0x0000)
R165:0(R0x0A5)	Reserved	-	0 (0x0000)
R166:0(R0x0A6)	Reserved	-	0 (0x0000)
R167:0(R0x0A7)	Reserved	-	0 (0x0000)
R168:0(R0x0A8)	Reserved	-	0 (0x0000)

Table2.1 Register List and Default Values

1 = read-only，always 1； 0 = read-only，always 0； d = programmable； ? = read-only，dynamic

Register # Dec (Hex)	Register Description	Data Format (Binary)	Default Value Dec(Hex)
R169:0(R0x0A9)	Reserved	-	0 (0x0000)
R170:0(R0x0AA)	Reserved	-	0 (0x0000)
R171:0(R0x0AB)	Reserved	-	0 (0x0000)
R172:0(R0x0AC)	Reserved	-	0 (0x0000)
R173:0(R0x0AD)	Reserved	-	0 (0x0000)
R174:0(R0x0AE)	Reserved	-	32 (0x0020)
R175:0(R0x0AF)	Reserved	-	0 (0x0000)
R176:0(R0x0B0)	Reserved	-	0 (0x0000)
R177:0(R0x0B1)	Reserved	-	0 (0x0000)
R178:0(R0x0B2)	Reserved	-	0 (0x0000)
R179:0(R0x0B3)	Reserved	-	0 (0x0000)
R180:0(R0x0B4)	Reserved	-	0 (0x0000)
R181:0(R0x0B5)	Reserved	-	0 (0x0000)
R182:0(R0x0B6)	Reserved	-	0 (0x0000)
R183:0(R0x0B7)	Reserved	-	0 (0x0000)
R184:0(R0x0B8)	Reserved	-	0 (0x0000)
R185:0(R0x0B9)	Reserved	-	0 (0x0000)
R186:0(R0x0BA)	Reserved	-	0 (0x0000)
R187:0(R0x0BB)	Reserved	-	0 (0x0000)
R188:0(R0x0BC)	Reserved	-	0 (0x0000)
R189:0(R0x0BD)	Reserved	-	0 (0x0000)
R190:0(R0x0BE)	Reserved	-	0 (0x0000)
R191:0(R0x0BF)	Reserved	-	0 (0x0000)
R192:0(R0x0C0)	Reserved	-	0 (0x0000)
R193:0(R0x0C1)	Reserved	-	0 (0x0000)
R194:0(R0x0C2)	Reserved	-	0 (0x0000)
R195:0(R0x0C3)	Reserved	-	0 (0x0000)
R196:0(R0x0C4)	Reserved	-	0 (0x0000)
R197:0(R0x0C5)	Reserved	-	0 (0x0000)
R198:0(R0x0C6)	Reserved	-	0 (0x0000)
R199:0(R0x0C7)	Reserved	-	0 (0x0000)
R200:0(R0x0C8)	Reserved	-	0 (0x0000)
R201:0(R0x0C9)	Reserved	-	0 (0x0000)
R202:0(R0x0CA)	Reserved	-	0 (0x0000)
R203:0(R0x0CB)	Reserved	-	0 (0x0000)

Table 2.1 Register List and Default Values

1 = read-only，always 1； 0 = read-only，always 0； d = programmable； ? = read-only，dynamic

Register # Dec (Hex)	Register Description	Data Format (Binary)	Default Value Dec(Hex)
R204:0(R0x0CC)	Reserved	-	0 (0x0000)
R205:0(R0x0CD)	Reserved	-	0 (0x0000)
R206:0(R0x0CE)	Reserved	-	0 (0x0000)
R207:0(R0x0CF)	Reserved	-	0 (0x0000)
R208:0(R0x0D0)	Reserved	-	0 (0x0000)
R209:0(R0x0D1)	Reserved	-	0 (0x0000)
R210:0(R0x0D2)	Reserved	-	0 (0x0000)
R211:0(R0x0D3)	Reserved	-	0 (0x0000)
R212:0(R0x0D4)	Reserved	-	0 (0x0000)
R213:0(R0x0D5)	Reserved	-	0 (0x0000)
R214:0(R0x0D6)	Reserved	-	0 (0x0000)
R215:0(R0x0D7)	Reserved	-	0 (0x0000)
R216:0(R0x0D8)	Reserved	-	0 (0x0000)
R217:0(R0x0D9)	Reserved	-	0 (0x0000)
R218:0(R0x0DA)	Reserved	-	0 (0x0000)
R219:0(R0x0DB)	Reserved	-	0 (0x0000)
R220:0(R0x0DC)	Reserved	-	0 (0x0000)
R221:0(R0x0DD)	Reserved	-	0 (0x0000)
R222:0(R0x0DE)	Reserved	-	0 (0x0000)
R223:0(R0x0DF)	Reserved	-	0 (0x0000)
R224:0(R0x0E0)	Reserved	-	0 (0x0000)
R225:0(R0x0E1)	Reserved	-	0 (0x0000)
R226:0(R0x0E2)	Reserved	-	0 (0x0000)
R227:0(R0x0E3)	Reserved	-	0 (0x0000)
R228:0(R0x0E4)	Reserved	-	0 (0x0000)
R229:0(R0x0E5)	Reserved	-	0 (0x0000)
R230:0(R0x0E6)	Reserved	-	0 (0x0000)
R231:0(R0x0E7)	Reserved	-	0 (0x0000)
R232:0(R0x0E8)	Reserved	-	0 (0x0000)
R233:0(R0x0E9)	Reserved	-	0 (0x0000)
R234:0(R0x0EA)	Reserved	-	0 (0x0000)
R235:0(R0x0EB)	Reserved	-	0 (0x0000)
R236:0(R0x0EC)	Reserved	-	0 (0x0000)
R237:0(R0x0ED)	Reserved	-	0 (0x0000)
R238:0(R0x0EE)	Reserved	-	0 (0x0000)

Table 2.1 Register List and Default Values

1 = read-only，always 1； 0 = read-only，always 0； d = programmable； ? = read-only，dynamic

Register # Dec (Hex)	Register Description	Data Format (Binary)	Default Value Dec(Hex)
R239:0(R0x0EF)	Reserved	-	0 (0x0000)
R240:0(R0x0F0)	Reserved	-	0 (0x0000)
R241:0(R0x0F1)	Reserved	-	0 (0x0000)
R248:0(R0x0F8)	Reserved	-	0 (0x0000)
R250:0(R0x0FA)	Reserved	-	0 (0x0000)
R251:0(R0x0FB)	Reserved	-	0 (0x0000)
R252:0(R0x0FC)	Reserved	-	0 (0x0000)
R253:0(R0x0FD)	Reserved	-	0 (0x0000)
R255:0(R0x0FF)	Chip_Version_Alt	???? ???? ???? ????	6145 (0x1801)

Register Description

Table 2.2 lists sensor register descriptions

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R0:0 R0x000	15:0	0x1801	Chip Version (RO)
	15:8	**RO**	**Part ID** **Two-digit BCD value typically derived from the reticle ID code.** **Legal values: [0, 255].**
	7:4	**RO**	**Analog Revision** **Constant value incremented with each mask change for the same Part ID.** **Legal values: [0, 15].**
	3:0	**RO**	**Digital Revision** **Constant value incremented with each digital functionality change for the same Part ID.** **Legal values: [0, 15].**
	Chip version.		
R1:0 R0x001	15:0	0x0036	Row Start (RW)
	The Y coordinate of the upper-left corner of the Field of View. If this register is set to an odd value, the next lower even value will be used. Writes are synchronized to frame boundaries. Affected by **Synchronize_Changes.** **Causes a Bad Frame if written. Legal values: [0, 2004], even.**		
R2:0 R0x002	15:0	0x0010	Column Start (RW)
	The X coordinate of the upper-left corner of the Field of View. The value will be rounded down to the nearest multiple of 2 times the column bin factor. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Legal values: [0, 2750], even. **Note: Set Column_Start such that it is in the form shown below, where n is an integer:** **Mirror_Column = 0 — Mirror_Column = 1** **no bin — 4n — 4n + 2** **Bin — 2x 8n — 8n + 4** **Bin — 4x 16n — 16n + 8**		
R3:0 R0x003	15:0	0x0797	Row Size (RW)
	The height of the field of view minus one. If this register is set to an even value, the next higher odd value will be used. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. **Causes a Bad Frame if written. Legal values: [1, 2005], odd.**		
R4:0 R0x004	15:0	0x0A1F	Column Size (RW)
	The width of the field of view minus one. If this register is set to an even value, the next higher odd value will be used. In other words, it should be (2*n*(Column_Bin + 1) - 1) for some integer n. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. **Causes a Bad Frame if written.** **Legal values: [1, 2751], odd.**		
R5:0 R0x005	15:0	0x0000	Horizontal Blank (RW)
	Extra time added to the end of each row, in pixel clocks. Incrementing this register will increase exposure and decrease frame rate. Setting a value less than the minimum will use the minumum horizontal blank. The minimum horizontal blank depends on the mode of the sensor. Writes are synchronized to frame boundaries.Affected by Synchronize_Changes. **Causes a Bad Frame if written. Legal values: [0, 4095].**		

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R6:0 R0x006	15:0	0x0019	Vertical Blank (RW)
	Extra time added to the end of each frame in rows minus one. Incrementing this register will decrease frame rate, but not affect exposure. Setting a value less than the minimum will use the minimum vertical blank. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Legal values: [8, 2047].		
R7:0 R0x007	15:0	0x1F82	Output Control (RW)
	15	**X**	**Reserved**
	14	**0x0000**	**Reserved**
	13	**X**	**Reserved**
	12:10	**0x0007**	**Output_Slew_Rate** **Controls the slew rate on digital output pads except for PIXCLK. Higher values imply faster transition times. Legal values: [0, 7].**
	9:7	**0x0007**	**PIXCLK_Slew_Rate** **Controls the slew rate on the PIXCLK pad. Higher values imply faster transition times.** **Legal values: [0, 7].**
	6	**0x0000**	**Reserved**
	5:4	**X**	**Reserved**
	3	**0x0000**	**Reserved**
	2	**0x0000**	**FIFO_Parallel_Data** **When set, pixels will be sent through the output FIFO before being sent off chip. This allows the output port to be running at a slower speed than f_PIXCLK, since the FIFO allows for pixels to be output during horizontal blank. Use of this mode requires the PLL to be set up properly.**
	1	**0x0001**	**Chip Enable** **When clear, sensor readout is stopped and analog circuitry is put in a state which draws minimal power. When set, the chip operates according to the current mode. Writing this bit does not affect the values of any other registers.**
	0	**0x0000**	**Synchronize Changes** **When set, changes to certain registers (those with the SC attribute) are delayed until the bit is clear. When cleared, all the delayed writes will happen immediately. Registers with the F attribute will still have the update synchronized to the next frame boundary.**
R8:0 R0x008	15:0	0x0000	Shutter Width Upper (RW)
	The most significant bits of the shutter width, which are combined with Shutter Width Lower (R9).		
R9:0 R0x009	15:0	0x0797	Shutter Width Lower (RW)
	The least significant bits of the shutter width. This is combined with Shutter_Width_Upper and Shutter_Delay for the effective shutter width. If set to zero, a value of "1" will be used.		

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R10:0 R0x00A	15:0	0x0000	Pixel Clock Control (RW)
	15	**0x0000X**	**Invert Pixel Clock** **When set, LVAL, FVAL, and D[11:0] should be captured on the rising edge of PIXCLK. When clear, they should be captured on the falling edge. This is accomplished by inverting the PIXCLK output.** **NOTE: This field is not reset by the soft Reset (R13).**
	14:11	**X**	**Reserved**
	10:8	**0x0000**	**Shift Pixel Clock** **Two's complement value representing how far to shift the PIXCLK output pin relative to D, in XCLKIN cycles. Positive values shift PIXCLK later in time relative to D (and thus relative to the internal array/datapath clock). No effect unless PIXCLK is divided by Divide Pixel Clock.** **NOTE: This field is not reset by the soft Reset (R13).** **Legal values: [-2, 2].**
	7	**X**	**Reserved**
	6:0	**0x0000**	**Divide Pixel Clock** **Produces a PIXCLK that is divided by the value times two. The value must be zero or a power of 2. This will slow down the internal clock in the array control and datapath blocks, including pixel readout. It will not affect the two-wire serial interface clock. A value of "0" corresponds to a PIXCLK with the same frequency as XCLKIN. A value of 1 means f_PIXCLK = (f_XCLKIN / 2); 2 means f_PIXCLK = (f_XCLKIN / 4); 64 means f_PIXCLK = (f_XCLKIN / 128); etc.** **NOTE: This field is not reset by the soft Reset (R13). This field should not be written while in streaming mode. Instead, Pause_Restart should be used to suspend output while the divider is being changed. Legal values: {0, 1, 2, 4, 8, 16, 32, 64}.**
R11:0 R0x00B	15:0	0x0000	Restart (RW)
	15:3	**X**	**Reserved**
	2	**0x0000**	**Trigger** **Setting this bit in Snapshot mode will cause the next trigger to occur as if the TRIGGER pin were properly asserted/negated. Ineffective if not in Snapshot mode. The sense of this bit is NOT affected by Invert Trigger.** **When using this bit instead of the TRIGGER pin, make sure that either the trigger pin is continuously asserted, or that the pad is continuously negated and Invert_Trigger is set.**
	1	**0x0000**	**Pause Restart** **When set, Restart will not automatically be cleared. Instead, the sensor will pause at row 0 after Restart is set. When Pause_Restart is cleared, the sensor will resume. This allows for a repeatable delay from clearing restart to FV. When clearing this bit, be sure not to clear Restart as well: it will be cleared automatically when the device has restarted.**
	0	**0x0000**	**Restart** **Setting this bit will cause the sensor to abandon the current frame and restart from the first row. It will take up to 2*t_ROW for the restart to take effect. This bit resets to 0 automatically unless Pause_Restart is set. Manually setting this bit to zero will cause undefined behavior.** **Volatile.**

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R12:0 R0x00C	15:0	0x0000	Shutter Delay (RW)
	A negative adjustment to the effective shutter width in ACLKs. See Shutter_Width_Lower. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Legal values: [0, 8191].		
R13:0 R0x00D	15:0	0x0050	PLL Control (RW)
	15	**0x0000**	**Reserved**
	14:13	**0x0000**	**Reserved**
	12:9	**X**	**Reserved**
	8	**0x0000**	**Reserved**
	7:4	**0x0005**	**Reserved**
	3:2	**X**	**Reserved**
	1	**0x0000**	**Use PLL** **When set, use the PLL output as the system clock. When clear, use XCLKIN as the system clock.**
	0	**0x0000**	**Power PLL** **When set, the PLL is powered. When clear, it is not powered.**
R17:0 R0x011	15:0	0x6404	PLL Config 1 (RW)
	15:8	**0x0064**	**PLL m Factor** **PLL output frequency multiplier.** **Legal values: [16, 255].**
	7:6	**X**	**Reserved**
	5:0	**0x0004**	**PLL n Divider** **PLL output frequency divider minus 1.** **Legal values: [0, 63].**
R18:0 R0x012	15:0	0x0000	PLL Config 2 (RW)
	15:13	**X**	**Reserved**
	12:8	**0x0000**	**Reserved**
	7:5	**X**	**Reserved**
	4:0	**0x0000**	**PLL p1 Divider** **PLL system clock divider minus 1. If this is set to an even number, the system clock duty cycle will not be 50:50. In this case, set all bits in R101 or slow down XCLKIN. Legal values: [0, 127].**

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R30:0 R0x01E	15:0	0x4006	Read Mode 1 (RW)
	15	**X**	**Reserved**
	14	**0x0001**	**Reserved**
	13	**0x0000**	**Reserved**
	12	**0x0000**	**Reserved**
	11	**0x0000**	**XOR Line Valid** **When set, produce a LVAL signal that is the XOR of FVAL and the normal line_valid. Ineffective if Continuous Line Valid is set. When clear, produce a normal LVAL.**
	10	**0x0000**	**Continuous Line Valid** **When set, produce the LVAL signal even during the vertical blank period. When clear, produce LVAL only when active rows are being read out (that is, only when FVAL is high). Ineffective if FIFO_Parallel_Data is set.**
	9	**0x0000**	**Invert Trigger** **When set, the sense of the TRIGGER input pin will be inverted.**
	8	**0x0000**	**Snapshot** **When set, the sensor enters snapshot mode, and will wait for a trigger event between frames. When clear, the sensor is in continuous mode. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes.**
	7	**0x0000**	**Global Reset** **When set, the Global Reset Release shutter will be used. When clear, the Electronic Rolling Shutter will be used. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes.**
	6	**0x0000**	**Bulb Exposure** **When set, exposure time will be controlled by an external trigger. When clear, exposure time will be controlled by the Shutter_Width_Lower and Shutter_Width_Upper registers. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes.**
	5	**0x0000**	**Invert Strobe** **When set, the STROBE signal will be active LOW (during exposure). When clear, the STROBE signal is active HIGH.**
	4	**0x0000**	**Strobe Enable** **When set, a strobe signal will be generated by the digital logic during integration. When clear, the strobe pin will be set to the value of Invert_Strobe.**
	3:2	**0x0001**	**Strobe Start** **Determines the timepoint when the strobe is asserted.** **0 – first trigger** **1 – start of simultaneous exposure** **2 – shutter width** **3 – second trigger** **Writes are synchronized to frame boundaries. Affected by Synchronize_Changes.**
	1:0	**0x0002**	**Strobe End** **Determines the timepoint when the strobe is negated. If this is set equal to or less than Strobe_Start, the width of the strobe pulse will be t_ROW. See Strobe_Start. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes.**

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R32:0 R0x020	15:0	0x0040	Read Mode 2 (RW)
	15	**0x0000**	**Mirror Row** **When set, row readout in the active image occurs in reverse numerical order starting from (Row_Start + Row_Size). When clear, row readout of the active image occurs in numerical order. This has no effect on the readout of the dark rows. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Causes a Bad Frame if written.**
	14	**0x0000**	**Mirror Column** **When set, column readout in the active image occurs in reverse numerical order, starting from (Column_Start + Column_Size). When clear, column readout of the active image occurs in numerical order. This has no effect on the readout of the dark columns. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes.**
	13	**0x0000**	**Reserved**
	12	**0x0000**	**Show Dark Columns** **When set, the dark columns will be output to the left of the active image, making the output image wider. This has no effect on integration time or frame rate. When clear, only columns that are part of the active image will be output. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes.**
	11	**0x0000**	**Show Dark Rows** **When set, the dark rows will be output before the active image rows, making the output image taller. This has no effect on integration time or frame rate. When clear, only rows from the active image will be output. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes.**
	10:7	**X**	**Reserved**
	6	**0x0001**	**Row BLC** **When set, digitally compensate for differing black levels between rows by adding Dark Target (R73) and subtracting the average value of the 8 same-color dark pixels at the beginning of the row. When clear, digitally add Row Black Default Offset (R75) to the value of each pixel.**
	5	**0x0000**	**Column Sum** **When set, column summing will be enabled, and in column bin modes, all sampled capacitors will be enabled for column readout, resulting in an effective gain equal to the column bin factor. When clear, column averaging will be done, and there will be no additional gain related to the column bin factor.** **Writes are synchronized to frame boundaries. Affected by Synchronize_Changes.**
	4	**0x0000**	**Reserved**
	3:0	**X**	**Reserved**

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R34:0 R0x022	15:0	0x0000	Row Address Mode (RW)
	15	**X**	**Reserved**
	14:12	**0x0000**	**Reserved**
	11	**X**	**Reserved**
	10:8	**0x0000**	**Reserved**
	7:6	**X**	**Reserved**
	5:4	**0x0000**	**Row Bin** **The number of rows to be read and averaged per row output minus one. The rows will be read independently into sampling capacitors, then averaged together before column readout. For normal readout, this should be 0. For Bin 2X, it should be 1; for Bin 4X, it should be 3. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Causes a Bad Frame if written. Legal values: [0, 3].**
	3	**X**	**Reserved**
	2:0	**0x0000**	**Row Skip** **The number of row-pairs to skip for every row-pair output. A value of zero means to read every row. For Skip 2X, this should be 1; for Skip 3X, it should be 2, and so on. This value should be no less than Row_Bin. For full binning, Row_Skip should equal Row_Bin. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Causes a Bad Frame if written. Legal values: [0, 7].**
R35:0 R0x023	15:0	0x0000	Column Address Mode (RW)
	15:11	**X**	**Reserved**
	10:8	**0x0000**	**Reserved**
	7:6	**X**	**Reserved**
	5:4	**0x0000**	**Column Bin** **The number of columns to be read and averaged per column output minus one. For normal readout, this should be zero. For Bin 2X, it shoud be 1; for Bin 4X, it should be 3. Writes are synchronized to frame boundaries. Affected by .Synchronize_Changes. Causes a Bad Frame if written. Legal values: {0, 1, 3}.**
	3	**X**	**Reserved**
	2:0	**0x0000**	**Column Skip** **The number of column-pairs to skip for every column-pair output. A value of zero means to read every column in the active image. For Skip 2X, this should be 1; for Skip 3X, this should be 2, and so on. This value should be no less than Column_Bin. For full binning, Column_Skip should equal Column_Bin. Writes are synchronized to frame boundaries. Affected by .Synchronize_Changes. Causes a Bad Frame if written.** **Legal values: [0, 6].**

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R43:0 R0x02B	15:0	0x0008	Green1 Gain (RW)
	15	**X**	**Reserved**
	14:8	**0x0000**	**Green1 Digital Gain** **Digital Gain for the Green1 channel minus 1 times 8. The actual digital gain is (1 + value/8), and can range from 1 (a setting of 0) to 16 (a setting of 120) in increments of 1/8. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile. Legal values: [0, 120].**
	7	**X**	**Reserved**
	6	**0x0000**	**Green1 Analog Multiplier** **Analog gain multiplier for the Green1 channel minus 1. If 1, an additional analog gain of 2x will be applied. If 0, no additional gain is applied. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile.**
	5:0	**0x0008**	**Green1 Analogl Gain** **Analog gain setting for the Green1 channel times 8. The effective gain for the channel is (((Green1_Digital_Gain/8) + 1) * (Green1_Analog_Multiplier + 1) * (Green1_Analog_Gain/ 8)). Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile. Legal values: [8, 63].**
R44:0 R0x02C	15:0	0x0008	Blue Gain (RW)
	15	**X**	**Reserved**
	14:8	**0x0000**	**Blue Digital Gain** **Digital Gain for the Blue channel minus 1 times 8. The actual digital gain is (1 + value/8), and can range from 1 (a setting of 0) to 16 (a setting of 120) in increments of 1/8. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile. Legal values: [0, 120].**
	7	**X**	**Reserved**
	6	**0x0000**	**Blue Analog Multiplier** **Analog gain multiplier for the Blue channel minus 1. If 1, an additional analog gain of 2X will be applied. If 0, no additional gain is applied. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile.**
	5:0	**0x0008**	**Blue Analog Gain** **Analog gain setting for the Blue channel times 8. The effective gain for the channel is (((Blue_Digital_Gain/8) + 1) * (Blue_Analog_Multiplier + 1) * (Blue_Analog_Gain/8)). Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile. Legal values: [8, 63].**

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R45:0 R0x02D	15:0	0x0008	Red Gain (RW)
	15	**X**	**Reserved**
	14:8	**0x0000**	**Red Digital Gain** **Digital Gain for the Red channel minus 1 times 8. The actual digital gain is (1 + value/8), and can range from 1 (a setting of 0) to 16 (a setting of 120) in increments of 1/8. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile. Legal values: [0, 120].**
	7	**X**	**Reserved**
	6	**0x0000**	**Red Analog Multiplier** **Analog gain multiplier for the Red channel minus 1. If 1, an additional analog gain of 2X will be applied. If 0, no additional gain is applied. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile.**
	5:0	**0x0008**	**Red Analog Gain** **Analog gain setting for the Red channel times 8. The effective gain for the channel is (((Red_Digital_Gain/8) + 1) * (Red_Analog_Multiplier + 1) * (Red_Analog_Gain/8)). Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile. Legal values: [8, 63].**
R46:0 R0x02E	15:0	0x0008	Green2 Gain (RW)
	15	**X**	**Reserved**
	14:8	**0x0000**	**Green2 Digital Gain** **Digital Gain for the Green2 channel minus 1 times 8. The actual digital gain is (1 + value/8), and can range from 1 (a setting of 0) to 16 (a setting of 120) in increments of 1/8 . Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile. Legal values: [0, 120].**
	7	**X**	**Reserved**
	6	**0x0000**	**Green2 Analog Multiplier** **Analog gain multiplier for the Green2 channel minus 1. If 1, an additional analog gain of 2x will be applied. If 0, no additional gain is applied. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile.**
	5:0	**X**	**Green2 Analog Gain** **Analog gain setting for the Green2 channel times 8. The effective gain for the channel is (((Green2_Digital_Gain/8) + 1) * (Green2_Analog_Multiplier + 1) * (Green2_Analog_Gain/ 8)). Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Volatile. Legal values: [8, 63].**
R53:0 R0x035	15:0	0x0008	Global Gain (WO)
	Writing the Global_Gain sets all four individual gain registers R43-R46 to the value. This register should not be read. **See Green1_Gain (R43) for a description of the various fields. Affected by Synchronize_Changes. Duplicate. Legal values: special**		
R73:0 R0x049	15:0	0x00A8	Row Black Target (RW)
	The target black level for the Row BLC algorithm.		

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R75:0 R0x04B	15:0	0x0028	Row Black Default Offset (RW)
	A two's-compliment offset digitally added to all active image pixel values when Row BLC (R30[6]) is disabled. Writes are synchronized to frame boundaries. Affected by Synchronize_Changes. Legal values: [0, 4095].		
R91:0 R0x05B	15:0	0x0001	BLC_Sample_Size (RW)
	If set, the "moving average" calculation in the BLC algorithm will use a sample size of 32. If clear, it will use a sample size of 1 (that is, each frame's black level will be considered independent of other frames).		
R92:0 R0x05C	15:0	0x005A	BLC_Tune_1 (RW)
	15:12	**X**	**Reserved**
	11:8	**0x0000**	**BLC_Delta_Damping A number subtracted from the calculated correction's magnitude when in delta mode. Setting this to a positive number will correct by that much less than the delta value. A negative (two's compliment) number will correct by more (possibly worsening the overshoot). This applies to the magnitude of the delta, so a positive damping value will be subtracted from a positive delta and added to a negative delta. Writes are synchronized to frame boundaries. Legal values: [-8, 7].**
	7:0	**0x005A**	**BLC_DAC_Settling_Time The number of pixclks it takes for a newly set offset to take effect divided by 2. Used to configure the fast sample algorithm. After setting a calibration value in fast sample mode, (value * 2) pixclks will elapse before the next sample is taken. Writes are synchronized to frame boundaries. Legal values: [0, 255].**
R93:0 R0x05D	15:0	0x2D13	BLC_Delta_Thresholds (RW)
	15	**X**	**Reserved**
	14:8	**0x002D**	**BLC_High_Delta_Threshold Upper delta threshold divided by 4. If the average black value for a color is higher than this value times 4 or lower than BLC_Low_Delta_Threshold times 4, the fast sampling and binary search modes will be activated (if enabled). Once the black level is between the BLC_High_Delta_Threshold and the BLC_Low_Delta_Threshold, the delta adjustment mode will be used (though fast sample mode will continue until the end of the frame). This value should be set no lower than BLC High Target Threshold. Writes are synchronized to frame boundaries. Legal values: [0, 127].**
	7	**X**	**Reserved**
	6:0	**0x0013**	**BLC_Low_Delta_Threshold Lower delta threshold divided by 4. See BLC_High_Delta_Threshold. Should be no higher than BLC_Low_Target_Threshold. Writes are synchronized to frame boundaries. Legal values: [0, 127].**

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R94:0 R0x05E	15:0	0x41FF	BLC_Tune_2 (RW)
	15	**X**	**Reserved**
	14:12	**0x0004**	**BLC_Step_Size** **Base 2 log of the change in pixel value (in LSBs) of a pixel when the analog offset is changed by one. Legal values: [0, 4].**
	11:9	**X**	**Reserved**
	8:0	**0x01FF**	**BLC_Max_Adjust** **The maximum adjustment (positive or negative) that the BLC delta adjustment mode is allowed to make to the analog offset. Legal values: [1, 511].**
R95:0 R0x05F	15:0	0x231D	BLC_Target_Thresholds (RW)
	15	**X**	**Reserved**
	14:8	**0x0023**	**BLC_High_Target_Threshold** **The upper target threshold of the BLC algorithm divided by 4. The target black value is 4 times the average of the BLC_High_Target_Threshold and the BLC_Low_Target_Threshold. When the black value for a color is within these thresholds, it will be considered to be on target. Writes are synchronized to frame boundaries. Legal values: [0, 127].**
	7	**X**	**Reserved**
	6:0	**0x001D**	**BLC_Low_Target_Threshold** **The lower target threshold for the BLC algorithm divided by 4. See BLC High Target Threshold above. Writes are synchronized to frame boundaries. Legal values: [0, 127].**
R96:0 R0x060	15:0	0x0020	Green1_Offset (RW)
	Two's-compliment representation of the analog offset value for Green1. If Manual_BLC (R98[0]) is set, this value will be used as the analog offset. Otherwise, the value may be overridden by the BLC algorithm. When read, this register returns the offset currently in use. The user-programmed value is always retained internally, and may be read by setting Manual_BLC. A value of -256 will set the offset to -255. Writes are synchronized to frame boundaries. Legal values: [-255, 255].		
R97:0 R0x061	15:0	0x0020	Green2_Offset (RW)
	Two's-complement representation of the analog offset value for Green2. See Green1_Offset. Writes are synchronized to frame boundaries. Legal values: [-255, 255].		

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R98:0 R0x062	15:0	0x0000	Black_Level_Calibration (RW)
	15	**0x0000**	**Disable_Fast_Sample** **When set, the fast sampling mode (multiple samples per frame) will not be used if the black level falls outside the delta thresholds; instead, only one sample-adjust will take place per frame. Binary search mode may still be used. When clear, fast sample mode will be used when necessary. Writes are synchronized to frame boundaries**
	14	**0x0000**	**Lock_Green_Calibration** **When set, the calibration offset chosen for Green1 will be used for Green2 pixels as well. Only effective if Green1_Analog_Gain equals Green2_Analog_Gain and Green1_Analog_Multiplier equals Green2_Analog_Multiplier. Writes are synchronized to frame boundaries.**
	13	**0x0000**	**Lock_Red_Blue_Calibration** **When set, the calibration offset chosen for Red will be used for Blue pixels as well. Only effective if Red_Analog_Gain equals Blue_Analog_Gain and Red_Analog_Multiplier equals Blue_Analog_Multiplier. Writes are synchronized to frame boundaries.**
	12	**0x0000**	**Recalculate_Black_Level** **When set, any running averages will be reset and the fast sample and binary search modes will be activated (if enabled). This bit always reads "0." Writes are synchronized to frame boundaries.**
	11	**0x0000**	**Disable_Binary_Search** **When set, binary search mode will not be used when the black level falls outside the delta thresholds; instead the delta mode will be used. Fast sampling mode may still be used if enabled. Writes are synchronized to frame boundaries.**
	10:2	**X**	**Reserved**
	1	**0x0000**	**Disable_Calibration** **When set, analog calibration is disabled. When clear, the programmed or automatic offsets will be used.**
	0	**0x0000**	**Manual_BLC** **When set, the user programmed calibration offsets from R96-R97 and R99-R100 will be used. Also, black level calculation will be disabled. When clear, the BLC algorithm will adjust the offsets to maintain the target black level. Issue a Restart after clearing this register to avoid updating offsets based on corrupt black rows.** **If this bit is 1, Show_Dark_Rows must be set to allow channel offset correction to function properly.** **Writes are synchronized to frame boundaries.**
R99:0 R0x063	15:0	0x0020	Red_Offset (RW)
	Two's-compliment representation of the analog offset value for Red. See Green1_Offset. Writes are synchronized to frame boundaries. Legal values: [-255, 255].		
R100:0 R0x064	15:0	0x0020	Blue_Offset (RW)
	Two's-compliment representation of the analog offset value for Blue. See Green1_Offset. Writes are synchronized to frame boundaries. Legal values: [-255, 255].		

Table 2.2 Register Description

Reg. #	Bits	Default	Name
R160:0 R0x0A0	6:3	0x0000	Test_Pattern_Control
			Sets the test pattern mode: **0: color field** **1: horizontal gradient** **2: vertical gradient** **3: diagonal** **4: classic** **5: marching 1's** **6: monochrome horizontal bars** **7: monochrome vertical bars** **8: vertical color bars** **Legal values: [0, 15].**
	2	**0x0**	**Reserved**
	1	**0x0**	**Reserved**
	0	**0x011**	**Enable_Test_Pattern** **Enables the test pattern. When set, data from the ADC will be replaced with a digitally generated test pattern specified by Test_Pattern_Mode.**
R161:0 R0x0A1	11:0	0x0000	Test_Pattern_Green
	Value used for green pixels of dark rows and columns in all test patterns, and for the color field. **Legal values: [0, 4095].**		
R162:0 R0x0A2	11:0	0x0000	Test_Pattern_Red
	As above for red. **Legal values: [0, 4095].**		
R163:0 R0x0A3	11:0	0x0000	Test_Pattern_Blue
	As above for blue. **Legal values: [0, 4095].**		
R164:0 R0x0A4	11:0	0x0000	Test_Pattern_Bar_Width
	The width of the monochrome color bars in test modes 6 and 7. This should be set to an odd value. **Legal values: [0, 4095], odd.**		
R255:0 R0x0FF	11:0	0x1800	Chip_Version_Alt
	Mirror of R0[15:0]. **Read-only. Duplicate. Appears in all pages. Legal values: special.**		

CHAPTER 3 Features

This chapter describes the features of the D5M board.

Reset

The D5M may be reset by either using RESETn (active LOW) or the Reset register.

Hard Reset

Assert (LOW) RESETn pin, it is not necessary to clock the device. All registers return to the factory defaults. When the pin is negated (HIGH), the chip resumes normal operation.

Soft Reset

Set the Reset register field to “1” (Reg0x0D[0] = 1). All registers except the following Will be reset:

- **Chip_Enable**
- **Synchronize_Changes**
- **Reset**
- **Use_PLL**
- **Power_PLL**
- **PLL_m_Factor**
- **PLL_n_Divider**
- **PLL_p1_Divider**

When the field is returned to “0,” the chip resumes normal operation.

Power Up and Power Down

When first powering on the D5M, follow this sequence:

1. Ensure RESETn is asserted (LOW).
2. Bring up the supplies. If both the analog and the digital supplies cannot be brought up simultaneously, ensure the digital supply comes up first.
3. Negate RESETn (HIGH) to bring up the sensor.

When powering down, be sure to follow this sequence to ensure that I/Os do not load any busses that they are connected to:

1. Assert RESETn.
2. Remove the supplies.

Clocks

XCLKIN and PIXCLK:

The D5M requires one clock (XCLKIN), which is nominally 96 MHz. By default, this results in pixels being output on the D[11:0] pins at a maximum data rate of 96 Mp/s. The XCLKIN clock can be divided down internally by setting Divide_Pixel_Clock to a non-zero value. This slows down the operation of the chip as though XCLKIN had been divided externally.

The D[11:0], LVAL, FVAL, and STROBE outputs are launched on the rising edge of PIXCLK, and should be captured on the falling edge of PIXCLK. The specific relationship of PIXCLK to these other outputs can be adjusted in two ways. If Invert_Pixel_Clock is set, the sense of PIXCLK is inverted from that shown in Figure 1.5 on page 5.. In addition, if the pixel clock has been divided by Divide_Pixel_Clock, it can be shifted relative to the other outputs by setting Shift_Pixel_Clock.

$$f_{PIXCLK} = \begin{cases} f_{XCLKIN} & \text{if Divide\_Pixel\_Clock} = 0 \\ f_{XCLKIN} / (2 \times \text{Divide\_Pixel\_Clock}) & \text{otherwise} \end{cases}$$

PLL

PLL Generated Master Clock

The PLL contains a prescaler to divide the input clock applied on XCLKIN, a VCO to multiply the prescaler output, and a another divider stage to generate the output clock. The clocking structure is shown is Figure 3.1. PLL control registers (in courier font) can be programmed to generate desired master clock frequency.

- **The PLL control registers must be programmed while the sensor is in the software Standby state. The effect of programming the PLL divisors while the sensor is in the streaming state is UNDEFINED.**

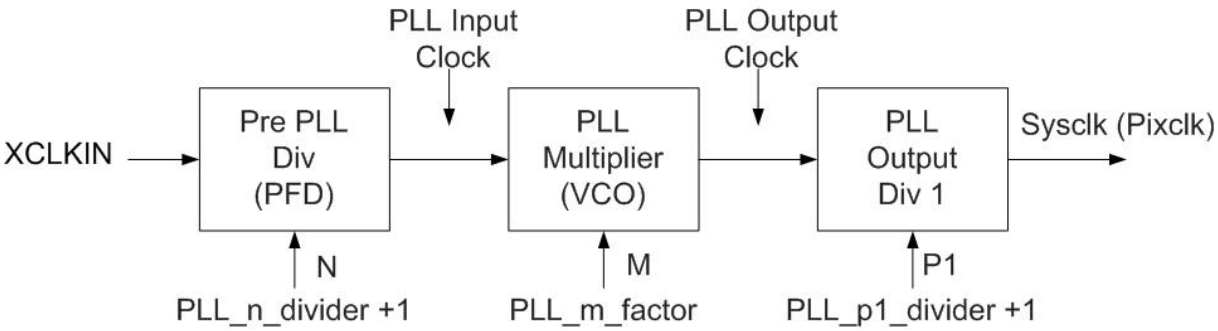


Figure 3.1 PLL Generated Master Clock

PLL Setup

The D5M has a PLL which can be used to generate the pixel clock internally.

To use the PLL:

1. Bring the D5M up as normal, make sure that ^{f}XCLKIN is between 6 MHz and 27 MHz and then power on the PLL by setting Power_PLL (Reg0x10[0] = 1).
2. Set PLL_m_Factor, PLL_n_Divider, and PLL_p1_Divider based on the desired input (^{f}XCLKIN) and output (^{f}PIXCLK) frequencies. Determine the M, N, and P1 values to achieve the desired ^{f}PIXCLK using these formula:

$$^{f}PIXCLK = (^{f}XCLKIN \times M) / (N \times P1)$$

where

M = PLL_m_Factor

N = PLL_n_Divider + 1

P1 = PLL_p1_Divider + 1

- If P1 is odd (that is, .PLL_p1_Divider is even), the duty cycle of the internal system clock will not be 50:50. In this case, it is important that either a slower clock is used or all clock enable bits are set in R101.

2 MHz < ^{f}XCLKIN / N < 13.5 MHz

180 MHz < (^{f}XCLKIN × M) / N < 360 MHz

It is desirable to keep (^{f}XCLKIN / n) as large as possible within the limits. Also, "m" must be between 16 and 255, inclusive.

3. Wait 1ms to ensure that the VCO has locked.
4. Set Use_PLL (Reg0x10[1] = 1) to switch from XCLKIN to the PLL-generated clock.

PLL Programming Example

The following is an example of register settings for input ^{f}XCLKIN = 24 MHz and output ^{f}PIXCLK = 96MHz.

```
REG=0, 0x10, 0x0051        // PLL_CONTROL; POWER UP PLL
REG=0,0x11, 0x4805         // PLL_CONFIG_1; CONFIG_1; PLL_n_Divider=5,
                           //PLL_m_Factor=72 (N=6, M=72)
REG=0,0x12, 0x0002         // PLL_CONFIG_2; PLL_p1_Divider=2, (P1=3)
Delay=1,                   // Wait 1ms for VCO to lock
REG=0, 0x10, 0x0053        // PLL_CONTROL; USE PLL
DELAY=1,                   // Wait 1ms
```

In this example,

^{f}PIXCLK = (^{f}XCLKIN × M) / (N × P1)

= 24 MHz x 72 / 6 x 3

= 96 MHz

Standby and Chip Enable

Standby and Chip Enable

The D5M can be put in a low-power Standby state by clearing the Chip_Enable register field (Reg0x07[1] = 0).

When the sensor is put in standby, all internal clocks are gated, and analog circuitry is put in a state that it draws minimal power. The two-wire serial interface remains minimally active so that the Chip_Enable bit can subsequently be cleared. Reads cannot be performed and only the Chip_Enable and Invert_Standby registers are writable.

If the sensor was in continuous mode when put in standby, it resumes from where it was when standby is deactivated. Naturally, this frame and the next frame are corrupted, though the sensor itself does not realize this. As this could affect automatic black level calibration, it is recommended that either the chip be paused (by setting Restart_Pause) before being put in standby mode, or it be restarted (setting Restart) upon resumption of operation.

For maximum power savings in standby mode, XCLKIN should not be toggling.

When standby mode is entered, by clearing Chip_Enable, the PLL is disabled automatically or powered down. It must be manually re-enabled when leaving standby as needed.

Full-Array Readout

The entire array, including dark pixels, can be read out without digital processing or automatic black level adjustments. This can be accomplished as follows:

1. Set Row_Start and Column_Start to 0.
2. Set Row_Size to 2,003.
3. Set Column_Size to 2,751.
4. Set Manual_BLC to 1.
5. Set Row_BLC to 0.
6. Set Row_Black_Default_Offset to 0.
7. Set Show_Dark_Rows and Show_Dark_Columns to 0.

If automatic analog (coarse) BLC is desired, but no digital processing, modify the above settings as follows

1. Set Row_Start to 12.
2. Set Row_Size to 1,993.
3. Set Manual_BLC to 0.

These settings result in the same array layout as above, but only 22 dark rows are available at the top of the array; the first eight are used in the black level algorithm, and there should be a two-row buffer between the black region and the active region.

Windows Control

Window Control

The output image window of the pixel (the field of view or FOV) is defined by four register fields. Column_Start and Row_Start define the X and Y coordinates of the upperleft corner of the FOV. Column_Size defines the width of the FOV, and Row_Size defines the height of the FOV in array pixels.

The Column_Start and Row_Start fields must be set to an even number. The Column_Size and Row_Size fields must be set to odd numbers (resulting in an even size for the FOV). The Row_Start register should be set no lower than 12 if either Manual_BLC is clear or Show_Dark_Rows is set.

If no special resolution modes are set (see below), the width of the output image, W, is (Column_Size + 1) and the height, H, is (Row_Size + 1).

Readout Modes

Subsampling

By default, the resolution of the output image is the full width and height of the FOV as defined above. The output resolution can be reduced by two methods: Skipping and Binning.

Row and column skip modes use subsampling to reduce the output resolution without reducing field-of-view. The D5M also has row and column binning modes, which can reduce the impact of aliasing introduced by the use of skip modes. This is achieved by the averaging of 2 or 3 adjacent rows and columns (adjacent same-color pixels). Both 2X and 4X binning modes are supported. Rows and columns can be binned independently.

Skipping

Skipping reduces resolution by using only selected pixels from the FOV in the output image. In skip mode, entire rows and columns of pixels are not sampled, resulting in a lower resolution output image. A skip 2X mode skips one Bayer pair of pixels for every pair output. Skip 3x skips two pairs for each one pair output. Rows and columns are always read out in pairs. If skip 2x mode is enabled with otherwise default sensor settings, the columns in the output image corresponds to the pixel array columns 16, 17, 20, 21, 24, 25...

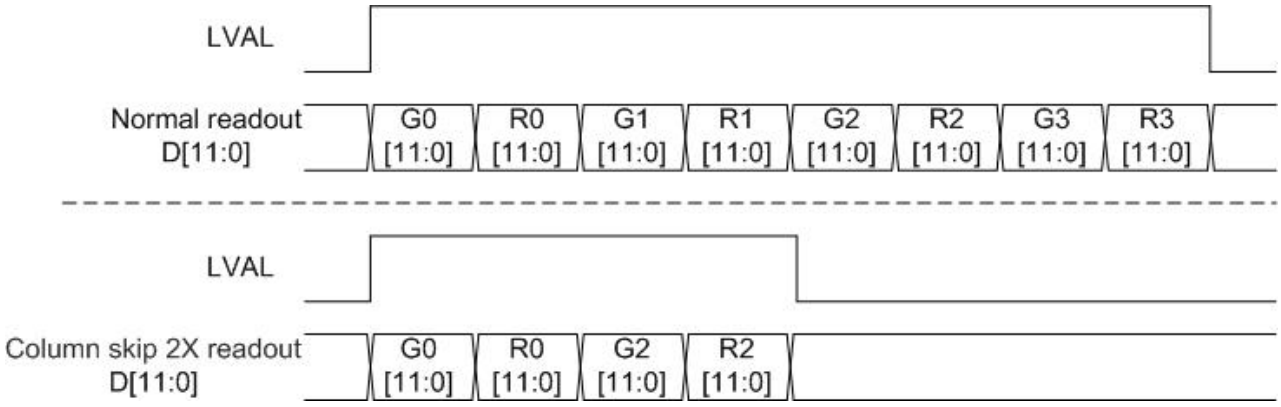


Figure 3.2 Eight Pixels in Normal and Column Skip 2X Readout Modes

Skipping can be enabled separately for rows and columns. To enable skip mode, set either or both of Row_Skip and Column_Skip to the number of pixel pairs that should be skipped for each pair used in the output image. For example, to set column skip 2x mode, set Column_Skip to “1.”

The size of the output image is reduced by the skip mode as shown in the following two equations:

W = 2 x ceil((Column_Size + 1) / (2 x (Column_Skip + 1)))

H = 2 x ceil((Row_Size + 1) / (2 x (Row_Skip + 1)))

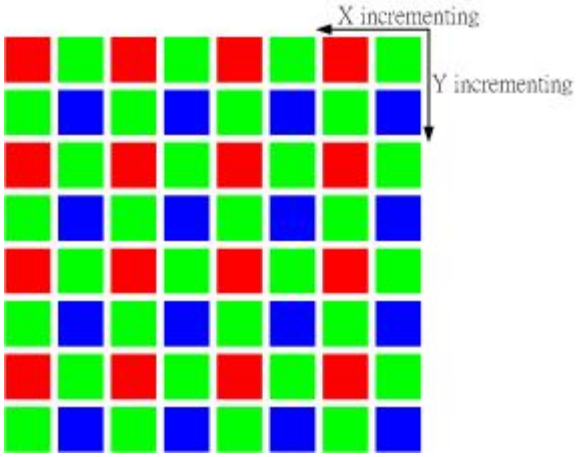


Figure 3.3: Pixel Readout (no skipping)

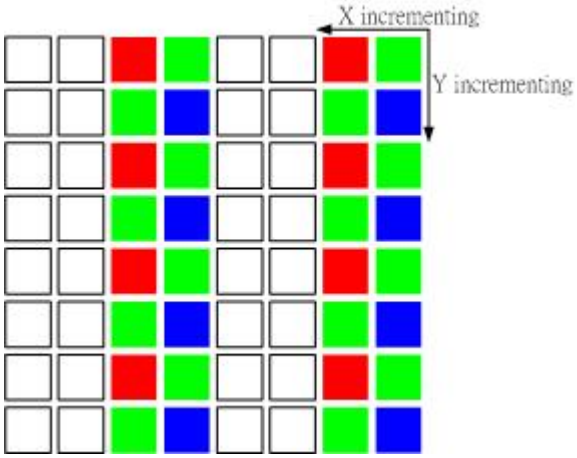


Figure 3.4: Pixel Readout (Column Skip 2X)

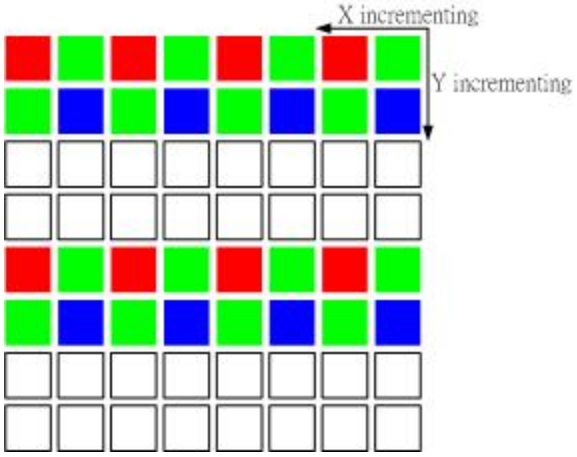


Figure 3.5: Pixel Readout (Row Skip 2X)

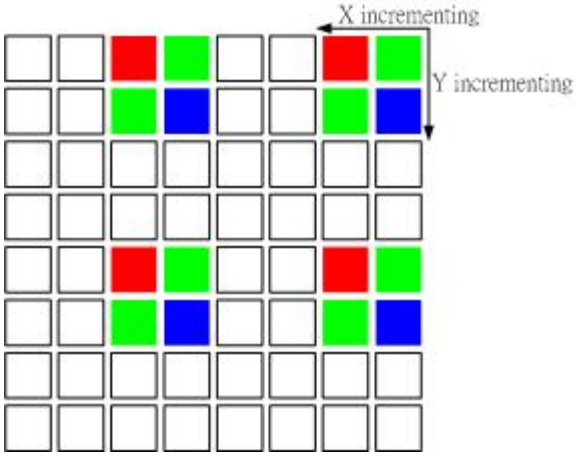


Figure 3.6: Pixel Readout (Column Skip 2X, Row Skip 2X)

Binning

Binning reduces resolution by combining adjacent same-color imager pixels to produce one output pixel. All of the pixels in the FOV contribute to the output image in bin mode.This can result in a more pleasing output image with reduced subsampling artifacts. It also improves low-light performance. For columns, the combination step can be either an averaging or summing operation. Depending on lighting conditions, one or the other may be desirable. In low-light conditions, summing produces a gain roughly equivalent to the column bin factor. Column summing may be enabled by setting Column_Sum.

Binning works in conjunction with skipping. Pixels that would be skipped because of the Column_Skip and Row_Skip settings can be averaged instead by setting Column_Bin and Row_Bin to the number of neighbor pixels to be averaged with each output pixel. For example, to set bin 2x mode, set Column_Skip and Column_Bin to 1. Additionally, Column_Start must be a multiple of (2 * (Column_Bin + 1)) and Row_Start must be a multiple of (2 * (Row_Bin + 1)).

Only certain combinations of binning and skipping are allowed.

These are shown in Table 3.1. If an illegal skip value is selected for a bin mode, a legal value is selected instead.

Table 3.1 Legal Values for Column_Skip Based on Column_Bin

Column_Bin	Legal Values for Column_Skip
0 (no binning)	**0, 1, 2, 3, 4, 5, 6**
1 (Bin 2x)	**1, 3, 5**
3 (Bin 4x)	**3**

◆ **Ensure that Column_Start (R0x02) is set in the form shown below, where n is an integer:**

	Mirror Column = 0	Mirror Column = 1
no bin	4n	4n + 2
Bin 2x	8n	8n + 4
Bin 4x	16n	16n + 8

Bin mode is illustrated in Figure 3.7 and Figure 3.8.

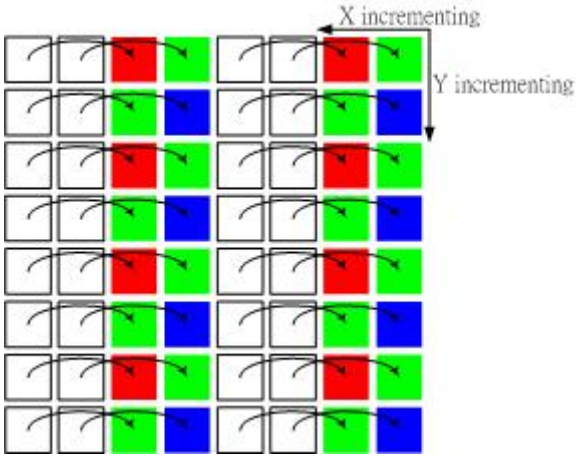


Figure 3.7: Pixel Readout (Column Bin 2X)

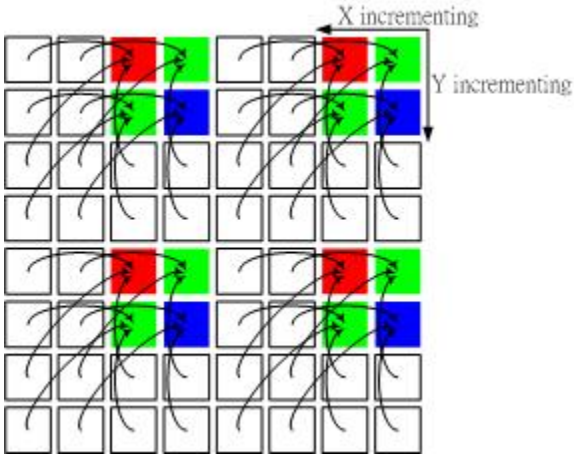


Figure 3.8: Pixel Readout (Column Bin 2X, Row Bin 2X)

Mirror

Column Mirror Image

By setting Reg0x20[14] = 1, the readout order of the columns are reversed as shown in Figure 3.9. The starting color, thus Bayer pattern, is preserved when mirroring the columns.

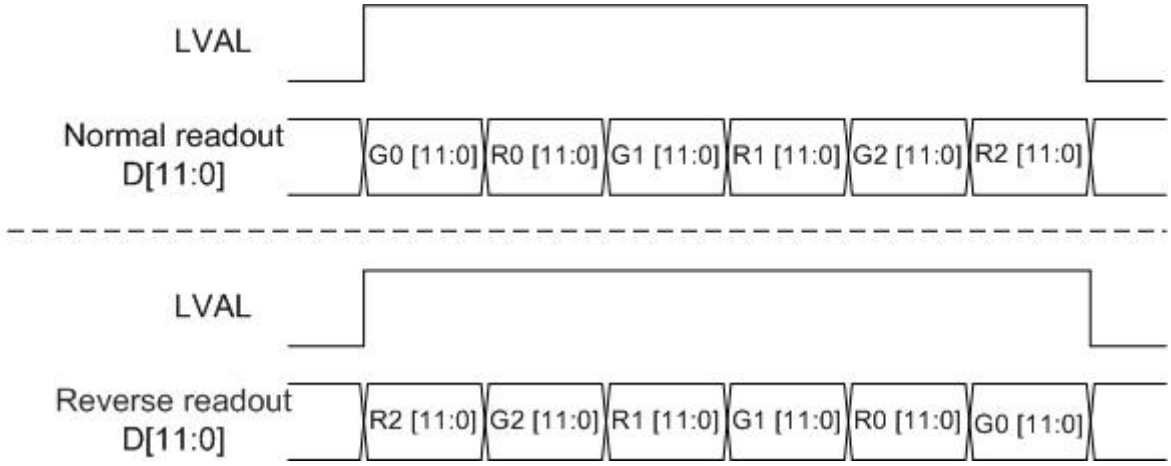


Figure 3.9 Six Pixels in Normal and Column Mirror Readout Modes

Row Mirror Image

By setting Reg0x20[15] = 1, the readout order of the rows are reversed as shown in Figure 3.10. The starting color, thus Bayer pattern, is preserved when mirroring the rows.

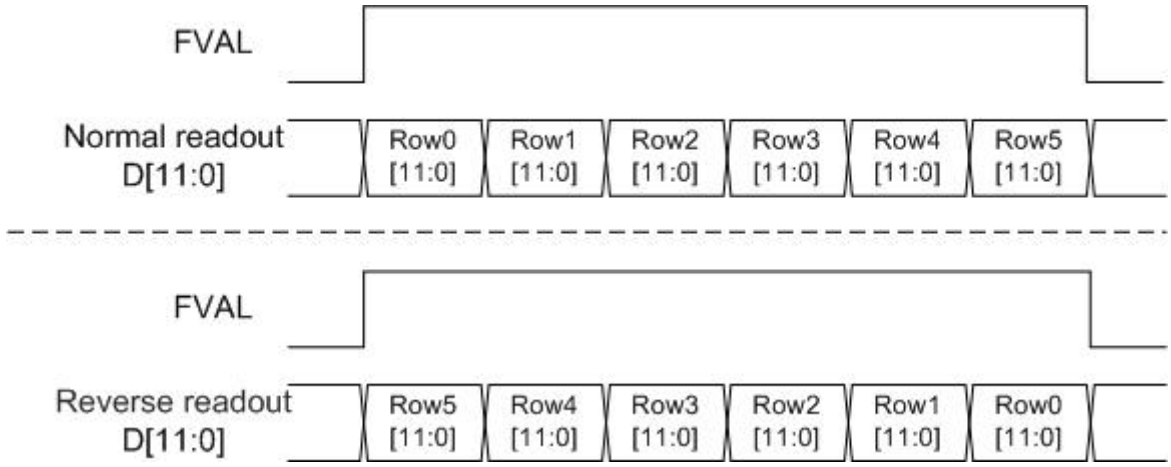


Figure 3.10 Six Rows in Normal and Row Mirror Readout Modes

By default, active pixels in the resulting image are output in row-major order (an entire row is output before the next row is begun), from lowest row/column number to highest. If desired, the output (and sampling) order of the rows and columns can be reversed. This only affects pixels in the active region defined above, not any pixels read out as dark rows or dark columns. When the readout direction is reversed, the color order is reversed as well (red, green, red, etc., instead of green, red, green, etc. for example).

If row binning is combined with row mirroring, the binning is still done in the positive direction. Therefore, if the first output row in bin 2x + row mirror was 1,997, pixels on rows 1,997 and 1,999 would be averaged together. The next pixel output would be from rows 1,996 and 1,998, followed by the average of 1,993 and 1,995.

For column mirroring plus binning, the span of pixels used should be the same as with non-mirror mode.

Maintaining a Constant Frame Rate

Maintaining a constant frame rate while continuing to have the ability to adjust certain parameters is the desired scenario. This is not always possible, however, since register updates are synchronized to the read pointer, and the shutter pointer for a frame is usually active during the readout of the previous frame. Therefore, any register changes that could affect the row time or the set of rows sampled causes the shutter pointer to start over at the beginning of the next frame.

By default, the following register fields cause a "bubble" in the output rate (i. e. the vertical blank increases for one frame) if they are written in continuous mode, even if the new value would not change the resulting frame rate:

- **Row_Start**
- **Row_Size**
- **Column_Size**
- **Horizontal_Blank**
- **Vertical_Blank**
- **Shutter_Delay**
- **Mirror_Row**
- **Row_Bin**
- **Row_Skip**
- **Column_Skip**

The size of this bubble is (SW × ^{t}ROW), calculating the row time according to the new settings.

The Shutter_Width_Lower and Shutter_Width_Upper fields may be written without causing a bubble in the output rate under certain circumstances. Since the shutter sequence for the next frame often is active during the output of the current frame, this would not be possible without special provisions in the hardware. Writes to these registers take effect two frames after the frame they are written, which allows the shutter width to increase without interrupting the output or producing a corrupt frame (as long as the change in shutter width does not affect the frame time).

Synchronizing Register Writes to Frame Boundaries

Changes to most register fields that affect the size or brightness of an image take effect on the frame after the one during which they are written. These fields are noted as "synchronized to frame boundaries" in Table 2.1 Register List and Default Values on page 9. To ensure that a register update takes effect on the next frame, the write operation must be completed after the leading edge of FRAME_VALID and before the trailing edge of FRAME_VALID.

As a special case, in Snapshot modes (see below), register writes that occur after Frame_Valid but before the next trigger will take effect immediately on the next frame, as if there had been a Restart However, if the trigger for the next frame in ERS Snapshot mode occurs during Frame_Valid , register writes take effect as with continuous mode.

Additional control over the timing of register updates can be achieved by using Synchronize_Changes. If this bit is set, writes to certain register fields that affect the brightness of the output image do not take effect immediately. Instead, the new value is remembered internally. When Synchronize_Changes is cleared, all the updates simultaneously take effect on the next frame (as if they had all been written the instant Synchronize_Changes was cleared). Register fields affected by this bit are identified in Table 2.2 Register Description on page 16.

Fields not identified as being frame-synchronized or affected by Synchronize_Changes are updated immediately after the register write is completed. The effect of these registers on the next frame can be difficult to predict if they affect the shutter pointer.

Restart

Restart the D5M

To restart the D5M At any time during the operation of the sensor, write a “1” to the Restart register (Reg0x0B[0]=1). This has two effects: first, the current frame is interrupted immediately. Second, any writes to frame-synchronized registers and the shutter width registers takes effect immediately, and a new frame starts (in continuous mode). Register updates being held by Synchronize_Changes do not take effect until that bit is cleared. The current row and one following row completes before the new frame is started, so the time between issuing the Restart and the beginning of the next frame can vary by about ^{t}ROW.

If Pause_Restart is set, rather than immediately beginning the next frame after a Restart in continuous mode, the sensor pauses at the beginning of the next frame until Pause_Restart is cleared. This can be used to achieve a deterministic time period from clearing the Pause_Restart bit to the beginning of the first frame, meaning that the controller does not need to be tightly synchronized to LVAL or FVAL.

- **When Pause_Restart is cleared, be sure to leave Restart set to “1” for proper operation. The Restart bit will be cleared automatically by the device.**

Image Acquisition Modes

The D5M supports two image acquisition modes (Shutter Types), electronic rolling shutter and global reset release.

Electronic Rolling Shutter

The ERS modes take pictures by scanning the rows of the sensor twice in the order described in “Full-Array Readout” on page 34. On the first scan, each row is released from reset, starting the exposure. On the second scan, the row is sampled, processed, and returned to the reset state. The exposure for any row is therefore the time between the first and second scans. Each row is exposed for the same duration, but at slightly different point in time, which can cause a shear in moving subjects.

Whenever the mode is changed to an ERS mode (even from another ERS mode), and before the first frame following reset, there is an anti blooming sequence where all rows are placed in reset. This sequence must complete before continuous readout begins.
This delay is:

$$^{t}ALLRESET = 16 \times 2004 \times tACLK$$

Global Reset Release

The GRR modes attempt to address the shearing effect by starting all rows' exposures at the same time. Instead of the first scan used in ERS mode, the reset to each row is released simultaneously. The second scan occurs as normal, so the exposure time for each row would different. Typically, an external mechanical shutter would be used to stop the exposure of all rows simultaneously.

In GRR modes, there is a startup overhead before each frame as all rows are initially placed in the reset state ($^{t}ALLRESET$). Unlike ERS mode, this delay always occurs before each frame. However, it occurs as soon as possible after the preceding frame, so typically the time from trigger to the start of exposure does not include this delay. To ensure that this is the case, the first trigger must occur no sooner than $^{t}ALLRESET$ after the previous frame is read out.

Exposure

The nominal exposure time, ^{t}EXP, is the effective shutter time in ERS modes, and is defined by the shutter width, SW, and the shutter overhead, SO, which includes the effect of Shutter_Delay. Exposure time for other modes is defined relative to this time. Increasing Shutter_Delay (SD) decreases the exposure time. Exposure times are typically specified in units of row time, although it is possible to fine tune exposures in units of tACLKs (where ^{t}ACLK is 2* ^{t}PIXCLK).

$$^{t}EXP = SW \times {}^{t}ROW - SO \times 2 \times {}^{t}PIXCLK$$

where:

SW = max(1, (2*16 × Shutter_Width_Upper) + Shutter_Width_Lower)

SO = 208 × (Row_Bin + 1) + 98 + min(SD, SDmax) – 94

SD = Shutter_Delay + 1

SDmax = 1232; if SW < 3

1504, else

The exposure time is calculated by determining the reset time of each pixel row (with time 0 being the start of the first row time), and subtracting it from the sample time. Under normal conditions in ERS modes, every pixel should end up with the same exposure time. In global shutter release modes, or in row binning modes, the exposure times of individual pixels can vary.

In global shutter release modes (described later) exposure time starts simultaneously for all rows, but still ends as defined above. In a real system, the exposure would be stopped by a mechanical shutter, which would effectively stop the exposure to all rows simultaneously. Since this specification does not consider the effect of an external shutter, each output row's exposure time will differ by ^{t}ROW from the previous row.

Global shutter modes also introduce a constant added to the shutter time for each row, since the exposure starts during the global shutter sequence, and not during any row's shutter sequence. For each additional row in a row bin, this offset will increase by the length of the shutter sequence.

In Bulb_Exposure modes (also detailed later), the exposure time is determined by the width of the TRIGGER pulse rather than the shutter width registers. In ERS Bulb mode, it is still be a multiple of row times, and the shutter overhead equation still applies. In GRR Bulb mode, the exposure time is granular to ACLKs, and shutter overhead (and thus Shutter_Delay) have no effect.

Operating Modes

In the default operating mode, the D5M continuously samples and outputs frames. It can be put in "snapshot" or triggered mode by setting Snapshot, which means that it samples and outputs a frame only when triggered. To leave snapshot mode, it is necessary to first clear Snapshot then issue a Restart.

When in snapshot mode, the sensor can use the ERS or the GRR. The exposure can be controlled as normal, with the Shutter_Width_Lower and Shutter_Width_Upper registers, or it can be controlled using the external TRIGGER signal. The various operating modes are summarized in Table 3.2.

Table 3.2 Operating Modes

Mode	Settings	Description
ERS Continuous	**Default**	**Frames are output continuously at the frame rate defined by ^{t}FRAME. ERS is used, and the exposure time is electronically controlled to be ^{t}EXP.**
ERS Snapshot	**Snapshot = 1**	**Frames are output one at a time, with each frame initiated by a trigger. ERS is used, and the exposure time is electronically controlled to be ^{t}EXP.**
ERS Bulb	**Snapshot = 1; Bulb_Exposure = 1**	**Frames are output one at a time, with each frame's exposure initiated by a trigger. ERS is used. End of exposure and readout are initiated by a second trigger.**
GRR Snapshot	**Snapshot = 1; Global_Reset = 1**	**Frames are output one at a time, with each frame initiated by a trigger. GRR is used. Readout is electronically triggered based on SW.**
GRR Bulb	**Snapshot = 1; Bulb_Exposure = 1; Global_Reset = 1**	**Frames are output one at a time, with each frame initiated by a trigger. GRR is used. Readout is initiated by a second trigger.**

- **In ERS Bulb mode, SW must be greater than 4 (use trigger wider than ^{t}ROW*4).**

All operating modes share a common set of operations:

1. Wait for the first trigger, then start the exposure.
2. Wait for the second trigger, then start the readout.

The first trigger is by default automatic, producing continuous images. If Snapshot is set, the first trigger can either be a low level on the TRIGGER pin or writing a “1” to the Trigger register field. If Invert_Trigger is set, the first trigger is a high level on TRIGGER pin (or a “1” written to Trigger register field). Since TRIGGER is level-sensitive, multiple frames can be output (with a frame rate of ^{t}FRAME) by holding TRIGGER pin at the triggering level.

The second trigger is also normally automatic, and generally occurs SW row times after the exposure is started. If Bulb_Exposure is set, the second trigger can either be a high level on TRIGGER or a write to Restart. If Invert_Trigger is set, the second trigger is a low level on TRIGGER (or a Restart). In bulb modes, the minimum possible exposure time depends on the mechanical shutter used.

After one frame has been output, the chip will reset back to step 1 above, eventually waiting for the first trigger again. The next trigger may be issued after ((VB - 8) x ^{t}ROW) in ERS modes or ^{t}ALLREST in GRR modes.

The choice of shutter type is made by Global_Reset. If it is set, the GRR shutter is used; otherwise, ERS is used. The two shutters are described in “Electronic Rolling Shutter” on page 42 and “Global Reset Release” on page 42.

The default ERS continuous mode is shown in Figure 1.5 on page 5. Figure 3.11 shows default signal timing for ERS snapshot modes, while Figure 3.12 on page 46 shows default signal timing for GRR snapshot modes.

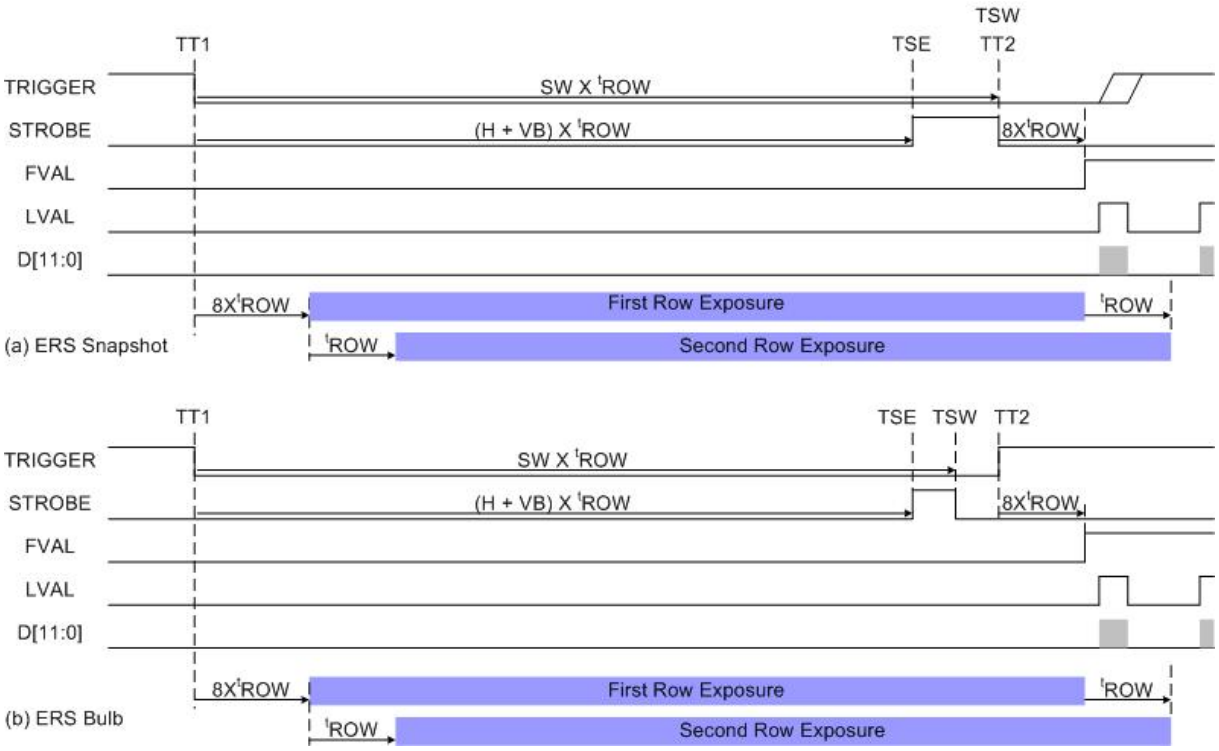


Figure 3.11 ERS Snapshot Timing

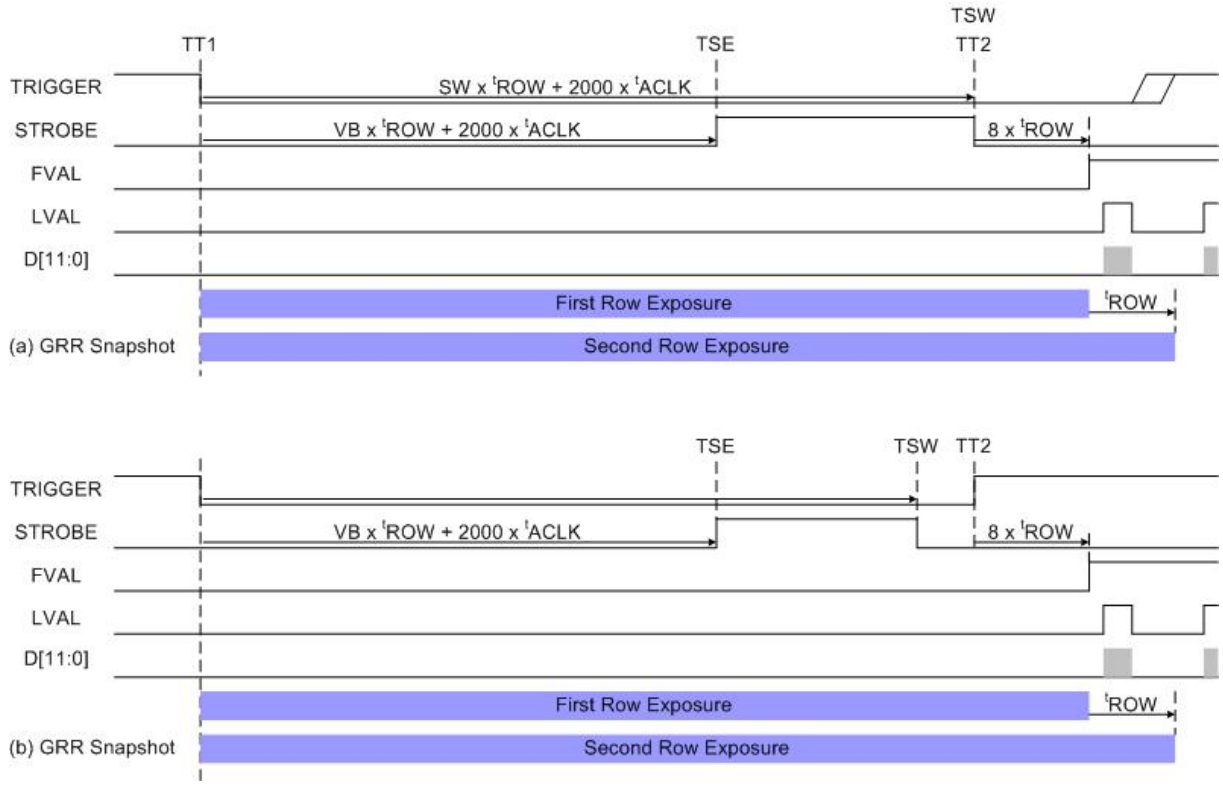


Figure 3.12 GRR Snapshot Timing

Strobe Control

To support synchronization of the exposure with external events such as a flash or mechanical shutter, the D5M produces a STROBE output. By default, this signal is asserted for approximately the time that all rows are simultaneously exposing, minus the vertical blank time, as shown in Figure 3.11 on page 45 and Figure 3.12 on page 46. Also indicated in these figures are the leading and trailing edges of STROBE which an be configured to occur at one of several timepoints. The leading edge of STROBE occurs at STROBE_Start, and the trailing edge at STROBE_End, which are set to codes described in Table 3.3.

Table 3.3 STROBE Timepoints

Symbol	Timepoint	Code
TT1	**Trigger 1 (start of shutter scan)**	**–**
TSE	**Start of exposure (all rows simultaneously exposing) offset by VB**	**1**
TSW	**End of Shutter width (expiration of the internal shutter width counter)**	**2**
TT2	**Trigger 2 (start of readout scan)**	**3**

If STROBE_Start and STROBE_End are set to the same timepoint, the strobe is a ^{t}ROW wide pulse starting at the STROBE_Start timepoint. If the settings are such that the strobe would occur after the trailing edge of FVAL, the strobe may be only ^{t}ACLK wide; however, since there is no concept of a row at that time. The sense of the STROBE signal can be inverted by setting Invert_Strobe (Reg0x1E[5] = 1). To use strobe as a flash in snapshot modes or with mechanical shutter, set the Strobe_Enable register bit field (Reg0x1E[4] = 1).

Signal Chain and Datapath

The signal chain and datapath are shown in Figure 3.13. Each color is processed independently, including separate gain and offset settings. Voltages sampled from the pixel array are first passed through an analog gain stage, which can produce gain factors between 1 and 8. An analog offset is then applied, and the signal is sent through a 12-bit analog-to-digital converter. In the digital space, a digital gain factor of between 1 and 16 is applied, and then a digital offset of between -2048 and 2047 is added. The resulting 12-bit pixel value is then output on the D[11:0] ports.

The analog offset applied is determined automatically by the black level calibration algorithm, which attempts to shift the output of the analog signal chain so that black is at a particular level. The digital offset is a fine-tuning of the analog offset.

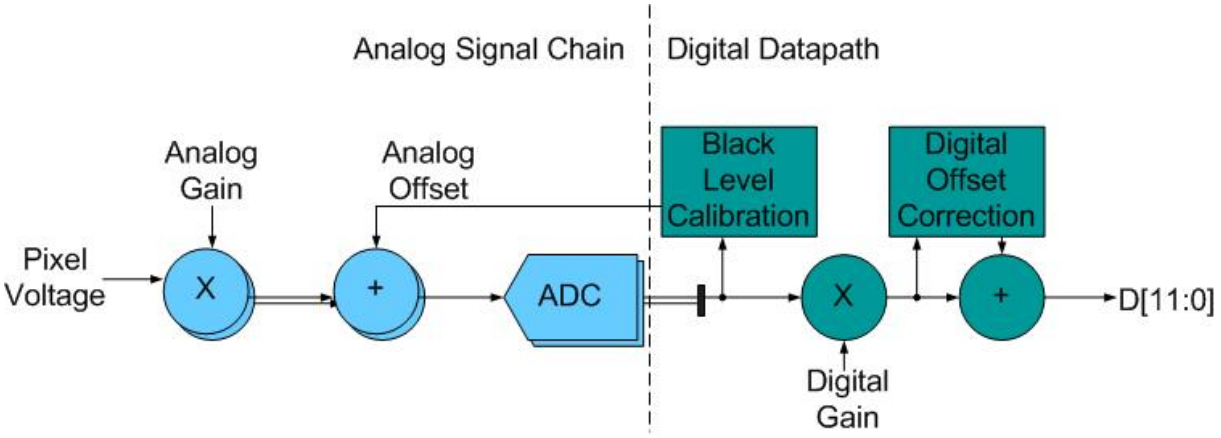


Figure 3.13 Signal Path

Gain

There are two types of gain supported: analog gain and digital gain. Combined, gains of between 1 and 128 are possible. The recommended gain settings are shown in Table 3.4.

Table 3.4 Gain Increment Settings

Gain Range	Increments	Digital Gain	Analog Multipier	Analog Gain
1– 4	**0.125**	**0**	**0**	**8–32**
4.25–8	**0.25**	**0**	**1**	**17–32**
9–128	**1**	**1–120**	**1**	**32**

- **Analog gain should be maximized before applying digital gain.**

The combined gain for a color C is given by:

$G_C = AG_C \times DG_C$.

Analog Gain

The analog gain is specified independently for each color channel. There are two components, the gain and the multiplier. The gain is specified by Green1_Analog_Gain, Red_Analog_Gain, Blue_Analog_Gain, and Green2_Analog_Gain in steps of 0.125. The analog multiplier is specified by Green1_Analog_Multiplier, Red_Analog_Multiplier, Blue_Analog_Multiplier, and Green2_Analog_Multiplier. These combine to form the analog gain for a given color C as shown in this equation:

$$AG_C = (1 + C\_Analog\_Multiplier) \times (C\_Analog\_Gain / 8)$$

The gain component can range from 0 to 7,875 in steps of 0.125, and the multiplier component can be either 0 or 1 (resulting in a multiplier of 1 or 2). However, it is best to keep the "gain" component between 1 and 4 for the best noise performance, and use the multiplier for gains between 4 and 8.

Digital Gain

The digital gain is specified independently for each color channel in steps of 0.125. It is controlled by the register fields Green1_Digital_Gain, Red_Digital_Gain, Blue_Digital_Gain, and Green2_Digital_Gain. The digital gain for a color C is given by:

$$DG_C = 1 + (C\_Digital\_Gain / 8)$$

Offset

The D5M sensor can apply an offset or shift to the image data in a number of ways:

An analog offset can be applied on a color-wise basis to the pixel voltage as it enters the ADC. This makes it possible to adjust for offset introduced in the pixel sampling and gain stages to be removed, centering the resulting voltage swing in the ADC's range. This offset can be automatically determined by the sensor using the automatic black level calibration (BLC) circuit, or it can be set manually by the user. It is a fairly coarse adjustment, with adjustment step sizes of 4 - 8 LSBs.

Digital offset is also added on a color-wise and line-wise basis to fine-tune the black level of the output image. This offset is based on an average black level taken from each row's dark columns, and is automatically determined by the digital row-wise black level calibration (RBLC) circuit. If the RBLC circuit is not used, a user defined offset can be applied instead. This offset has a resolution of 1 LSB.

A digital offset is added on a color-wise basis to account for channel offsets that can be introduced due to "even" and "odd" pixels of the same color going through a slightly different ADC chain. This offset is automatically determined based on dark row data, but it can also be manually set.

Analog Black Level Calibration

The D5M black level calibration circuitry provides a feedback control system since adjustments to the analog offset are imprecise by nature. The goal is that within the dark row region of any supported output image size, the offset should have been adjusted such that the average black level falls within the specified target thresholds.

The analog offsets normally need a major adjustment only when leaving the Reset state or when there has been a change to a color's analog gain. Factors like shutter width and temperature have lower-order impact, and generally only require a minor adjustment to the analog offsets. The D5M has various calibration modes to keep the system stable while still supporting the need for rapid offset adjustments when necessary.

The two basic steps of black level calibration are:

1. Take a sample.
2. If necessary, adjust the analog offset.

Black level calibration is normally done separately for each color channel, and different channels can be using different sample or adjustment methods at the same time. However, since both Green1 and Green2 pixels go through the same signal chain, and Red and Blue pixels likewise go through the same signal chain, it is expected that the chosen offset for these pairs should be the same as long as the gains are the same. If Lock_Green_Calibration is set, and (Green1_Analog_Gain = Green2_Analog_Gain) and (Green1_Analog_Multiplier = Green2_Analog_Multiplier), the calculated or user-specified Green1_Offset is used for both green channels. Similarly, if Lock_Red/ Blue_Calibration is set, and (Red_Analog_Gain = Blue_Analog_Gain) and (Red_Analog_Multiplier = Blue_Analog_Multiplier), the calculated or user-specified Red_Offset is used for both the red and blue channels.

The currently-in-effect values of the offsets can be read from the Green1_Offset, Red_Offset, Blue_Offset, and Green2_Offset registers. Writes to these registers when Manual_BLC is set changes the offsets being used. In automatic BLC mode, writes to these registers are effective when manual mode is re-entered. In Manual_BLC mode, no sampling or adjusting takes place for any color.

Digital Black Level Calibration

Digital black level calibration is the final calculation applied to pixel data before it is output. It provides a precise black level to complement the coarser-grained analog black level calibration, and also corrects for black level shift introduced by digital gain. This correction applies to the active columns for all rows, including dark rows.

Test Patterns

The D5M has the capability of injecting a number of test patterns into the top of the datapath in order to debug the digital logic. With one of the test patterns activated, any of the datapath functions can be enabled in order exercise it in a deterministic fashion. Test patterns are enabled when Enable_Test_Pattern is set. Only one of the test patterns can be enabled at a given point in time by setting the Test_Pattern_Mode register according to Table 3.5. When test patterns are enabled the active area will receive the value specified by the selected test pattern and the dark pixels will receive the value in Test_Pattern_Green for green pixels, Test_Pattern_Blue for blue pixels, and Test_Pattern_Red for red pixels.

- **It is recommended to turn off black level calibration (BLC) when Test Pattern is enabled.**

Table 3.5 Test Pattern Modes

Test_Pattern_Mode	Test Pattern Output
0	**Color Field (Normal Operation)**
1	**Horizontal Gradient**
2	**Vertical Gradient**
3	**Diagonal Gradient**
4	**Classic Test Pattern**
5	**Marching 1's**
6	**Monochrome Horizontal Bars**
7	**Monochrome Vertical Bars**
8	**Vertical Color Bars**

Classic Test Pattern

When selected, a value from Test_Data will be sent through the digital pipeline instead of sampled data from the sensor. The value will alternate between Test_Data for even columns and ~(Test_Data) for odd columns.

Color Field

When selected, the value for each pixel is determined by its color. Green pixels will receive the value in Test_Pattern_Green, red pixels will receive the value in Test_Pattern_Red, and blue pixels will receive the value in Test_Pattern_Blue.

Vertical Color Bars

When selected, a typical color bar pattern will be sent through the digital pipeline.

Horizontal Gradient

When selected, a horizontal gradient will be produced based on a counter which increments on every active pixel.

Vertical Gradient

When selected, a vertical gradient will be produced based on a counter which increments on every active row.

Diagonal Gradient

When selected, a diagonal gradient will be produced based on the counter used by the horizontal and vertical gradients.

Marching 1's

When selected, a marching 1's pattern will be sent through the digital pipeline. The first value in each row is 1.

Monochrome Vertical Bars

When selected, vertical monochrome bars will be sent through the digital pipeline. The width of each bar can be set in Test_Pattern_Bar_Width and the intensity of each bar is set by Test_Pattern_Green for even bars and Test_Pattern_Blue for odd bars.

Monochrome Horizontal Bars

When selected, horizontal monochrome bars will be sent through the digital pipeline. The width of each bar can be set in Test_Pattern_Bar_Width and the intensity of each bar is set by Test_Pattern_Green for even bars and Test_Pattern_Blue for odd bars.

CHAPTER 4

D5M Serial Bus Description

Registers are written to and read from the D5M through the two-wire serial interface bus. The D5M is a serial interface slave and is controlled by the serial clock (SCLK), which is driven by the serial interface master. Data is transferred into and out of the D5M through the serial data (SDATA) line. The SDATA line is pulled up to VDDQ off-chip by a 1.5KΩ resistor. Either the slave or master device can pull the SDATA line LOW— the serial interface protocol determines which device is allowed to pull the SDATA line down at any given time.

Serial Bus Description

Protocol

The two-wire serial defines several different transmission codes, as follows:

- a start bit
- the slave device 8-bit address
- a(n) (no) acknowledge bit
- an 8-bit message
- a stop bit

Sequence

A typical read or write sequence begins by the master sending a start bit. After the start bit, the master sends the slave device's 8-bit address. The last bit of the address determines if the request is a read or a write, where a “0” indicates a write and a “1” indicates a read. The slave device acknowledges its address by sending an acknowledge bit back to the master.

If the request was a write, the master then transfers the 8-bit register address to which a write should take place. The slave sends an acknowledge bit to indicate that the register address has been received. The master then transfers the data eight bits at a time, with the slave sending an acknowledge bit after each eight bits. The D5M uses 16-bit data for its internal registers, thus requiring two 8-bit transfers to write to one register. After 16 bits are transferred, the register address is automatically incremented, so that the next 16 bits are written to the next register address. The master stops writing by sending a start or stop bit.

A typical read sequence is executed as follows. First the master sends the write-mode slave address and 8-bit register address, just as in the write request. The master then sends a start bit and the read-mode slave address. The master then clocks out the register data eight bits at a time. The master sends an acknowledge bit after each 8-bit transfer. The register address is auto-incremented after every 16 bits is transferred. The data transfer is stopped when the master sends a no-acknowledge bit.

Bus Idle State

The bus is idle when both the data and clock lines are HIGH. Control of the bus is initiated with a start bit, and the bus is released with a stop bit. Only the master can generate the start and stop bits.

Start Bit

The start bit is defined as a HIGH-to-LOW transition of the data line while the clock line is HIGH.

Stop Bit

The stop bit is defined as a LOW-to-HIGH transition of the data line while the clock line is HIGH.

Slave Address

The 8-bit address of a two-wire serial interface device consists of seven bits of address and 1 bit of direction. A “0” in the LSB (least significant bit) of the address indicates write mode (0xBA), and a “1” indicates read mode (0xBB).

Data Bit Transfer

One data bit is transferred during each clock pulse. The serial interface clock pulse is provided by the master. The data must be stable during the HIGH period of the two-wire serial interface clock—it can only change when the serial clock is LOW. Data is transferred eight bits at a time, followed by an acknowledge bit.

Acknowledge Bit

The master generates the acknowledge clock pulse. The transmitter (which is the master when writing, or the slave when reading) releases the data line, and the receiver indicates an acknowledge bit by pulling the data line LOW during the acknowledge clock pulse.

No-Acknowledge Bit

The no-acknowledge bit is generated when the data line is not pulled down by the receiver during the acknowledge clock pulse. A no-acknowledge bit is used to terminate a read sequence.

Two-Wire Serial Register Interface

The electrical characteristics of the two-wire serial register interface (SCLK, SDATA) are shown in Table 4.1 and Figure 4.1.

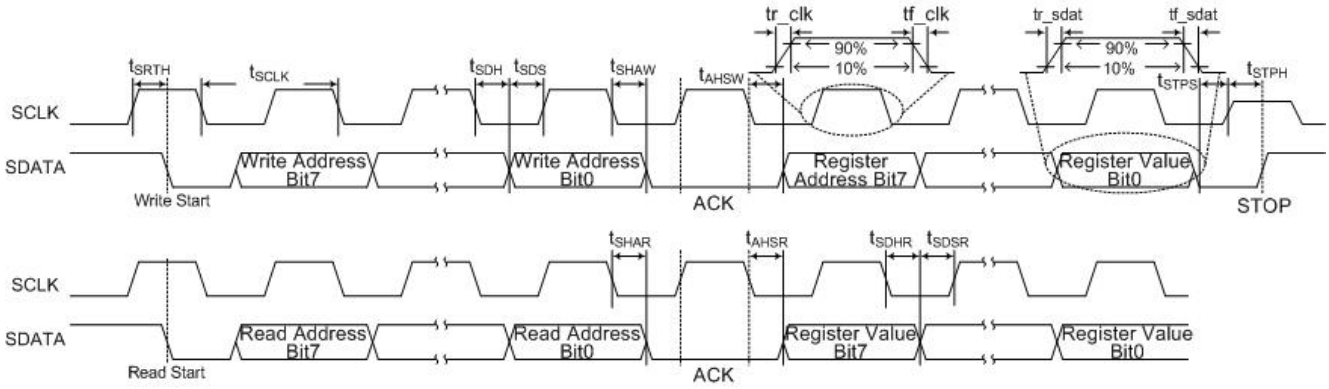


Figure 4.1 Two-Wire Serial Bus Timing Parameters

Table 4.1 Two-Wire Serial Bus Characteristics

Symbol	Definition	Condition	Min	Typ	Max	Unit
^{f}SCLK	Serial interface input clock frequency	—	—	—	400	kHz
^{t}SCLK	Serial Input clock period	—	—	—	2.5	µsec
	SCLK Duty Cycle	—	40	50	60	%
tr_sclk	SCLK rise time		—	34	—	ns
tf_sclk	SCLK fall time		—	8	—	ns
tr_sdat	SDATA rise time		—	34	—	ns
tf_sdat	SDATA fall time		—	10	—	ns
^{t}SRTH	Start hold time	WRITE/READ	0	10	28	ns
^{t}SDH	SDATA hold	WRITE	0	0	0	ns
^{t}SDS	SDATA setup	WRITE	0	19.9	59.9	ns
^{t}SHAW	SDATA hold to ACK	WRITE	279	281	300	ns
^{t}AHSW	ACK hold to SDATA	WRITE	279	281	300	ns
^{t}STPS	Stop setup time	WRITE/READ	0	0	0	ns
^{t}STPH	Stop hold time	WRITE/READ	0	0	0	ns
^{t}SHAR	SDATA hold to ACK	READ	279	284	300	ns
^{t}AHSR	ACK hold to SDATA	READ	279	284	300	ns
^{t}SDHR	SDATA hold	READ	0	0	0	ns
^{t}SDSR	SDATA setup	READ	0	19.9	59.9	ns
CIN_SI	Serial interface input pin capacitance	—	—	3.5	—	pF
CLOAD_SD	SDATA max load capacitance	—	—	15	—	pF
RSD	SDATA pull-up resistor	—	—	1.5	—	kΩ

I/O Timing

By default, the D5M launches pixel data, FVAL and LVAL with the rising edge of PIXCLK. The expectation is that the user captures D[11:0], FVAL and LVAL using the falling edge of PIXCLK.

See Figure 4.2 and Table 4.2 for I/O timing (AC) characteristics.

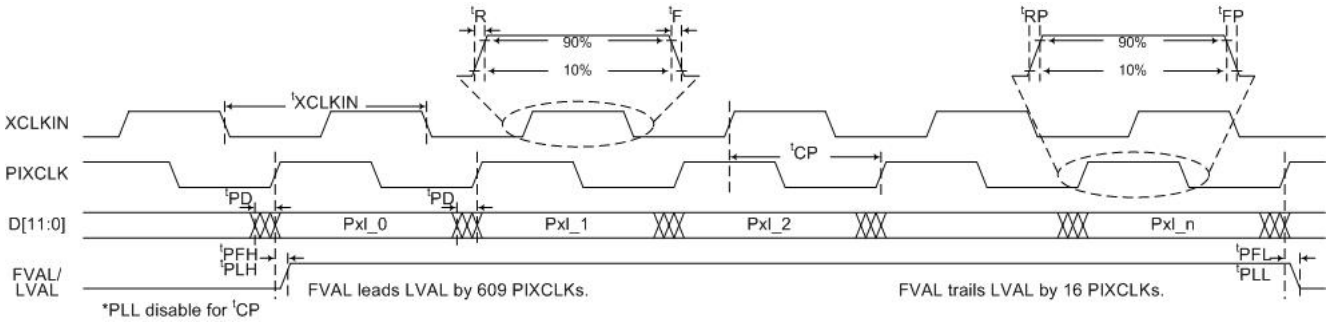


Figure 4.2 I/O Timing Diagram

Table 4.2 I/O Timing Characteristics

Symbol	Definition	Conditions	Min	Typ	Max	Units
$f_{XCLKIN1}$	**Input clock frequency**	**PLL enabled**	**6**	**—**	**27**	**MHz**
$t_{XCLKIN1}$	**Input clock period**	**PLL enabled**	**166**	**—**	**37**	**ns**
$f_{XCLKIN2}$	**Input clock frequency**	**PLL disabled**	**6**	**—**	**96**	**MHz**
$t_{XCLKIN2}$	**Input clock period**	**PLL disabled**	**125**	**—**	**10.4**	**ns**
t_R	**Input clock rise time**		**0.03**	**—**	**1**	**V/ns**
t_F	**Input clock fall time**		**0.03**	**—**	**1**	**V/ns**
t_{RP}	**Pixclk rise time**		**0.03**	**—**	**1**	**V/ns**
t_{FP}	**Pixclk fall time**		**0.03**	**—**	**1**	**V/ns**
	Clock duty cycle		**40**	**50**	**60**	**%**
$t_{(PIX\ JITTER)}$	**Jitter on PIXCLK**		**—**	**1.026**	**—**	**ps**
t_{JITTER}	**Input clock jitter 96 MHz**		**—**	**220**	**—**	**ps**
t_{CP}	**XCLKIN to PIXCLK propagation delay**	**Nominal voltages**	**11.5**	**17.7**	**19.1**	**ns**
f_{PIXCLK}	**PIXCLK frequency**	**Default**	**6**	**—**	**96**	**MHz**
t_{PD}	**PIXCLK to data valid**	**Default**	**0.6**	**1.2**	**2.2**	**ns**
t_{PFH}	**PIXCLK to FV HIGH**	**Default**	**2.8**	**3.6**	**4.6**	**ns**
t_{PLH}	**PIXCLK to LV HIGH**	**Default**	**2.2**	**3.2**	**4.2**	**ns**
t_{PFL}	**PIXCLK to FV LOW**	**Default**	**2.4**	**3.4**	**4.2**	**ns**
t_{PLL}	**PIXCLK to LV LOW**	**Default**	**2.6**	**3.4**	**4.2**	**ns**
CLOAD	**Output load capacitance**		**—**	**<10**	**—**	**pF**
CIN	**Input pin capacitance**		**—**	**2.5**	**—**	**pF**