



# VGA CMOS Image Sensor GC0308

## Application Notes

2010-02-08

GalaxyCore Inc.

# 目 录

1. 简介 .....	4
2. Pixel Array 说明 .....	4
3. 系统应用 .....	5
3.1 外围连接 .....	6
3.2 应用时序 .....	6
3.3 芯片控制 .....	7
3.3.1 寄存器复位 .....	7
3.3.2 Standby 模式控制 .....	7
3.3.3 输出使能控制 .....	8
3.3.4 输出 Pin 驱动能力 .....	8
4. 芯片功能方面配置 .....	9
4.1 Pixel Array 控制 .....	9
4.2 时钟预分频 .....	9
4.3 输出时序说明及同步信号控制 .....	10
4.3.1 输出时序说明 .....	10
4.3.2 同步信号极性控制 .....	11
4.4 图像窗口设置 .....	12
4.5 Subsample 输出 .....	14
4.6 Anti_flicker 与 HB, VB 与 CLK 的关系。 .....	14
4.6.1 Anti_flicker 计算 .....	14
4.6.2 Exp Level 配置 .....	15
4.7 各种输出模式 .....	15
5. 图像功能模块调试 .....	16
5.1 AEC .....	16

---

5.2	AWB .....	17
5.3	LSC (Lens Shading Compensation) .....	17
5.4	去坏点 (delect defective points) .....	17
5.5	去噪 (denoise) .....	18
5.6	边缘加强 (Edge enhancement) .....	18
5.7	对比度.....	18
5.8	饱和度.....	19
5.9	特效参数.....	19
5.10	Gamma 调整.....	19
5.11	Color Matrix.....	19

## 1. 简介

此文档为应用系统设计者了解 GC0308 提供关键说明，针对 GC0308 芯片在寄存器设定及系统应用上给予指导，方便系统设计及调试工程师快速建立应用方案和调试效果。如果需要更详细的寄存器定义，请参考 GC0308 DataSheet 文档。

GC0308 是格科微电子有限公司研发的最新 VGA CMOS 图像传感器芯片。它采用了公司最新的 pixel 工艺和图像处理技术，为客户提供高性价比的拍照解决方案。

GC0308 的 two-wire serial interface 总线读写地址为 0x42/0x43，芯片判断通过 0x00 只读寄存器来实现，如果读出值为 0x9b，则为 GC0308。

GC0308 的功能框图：

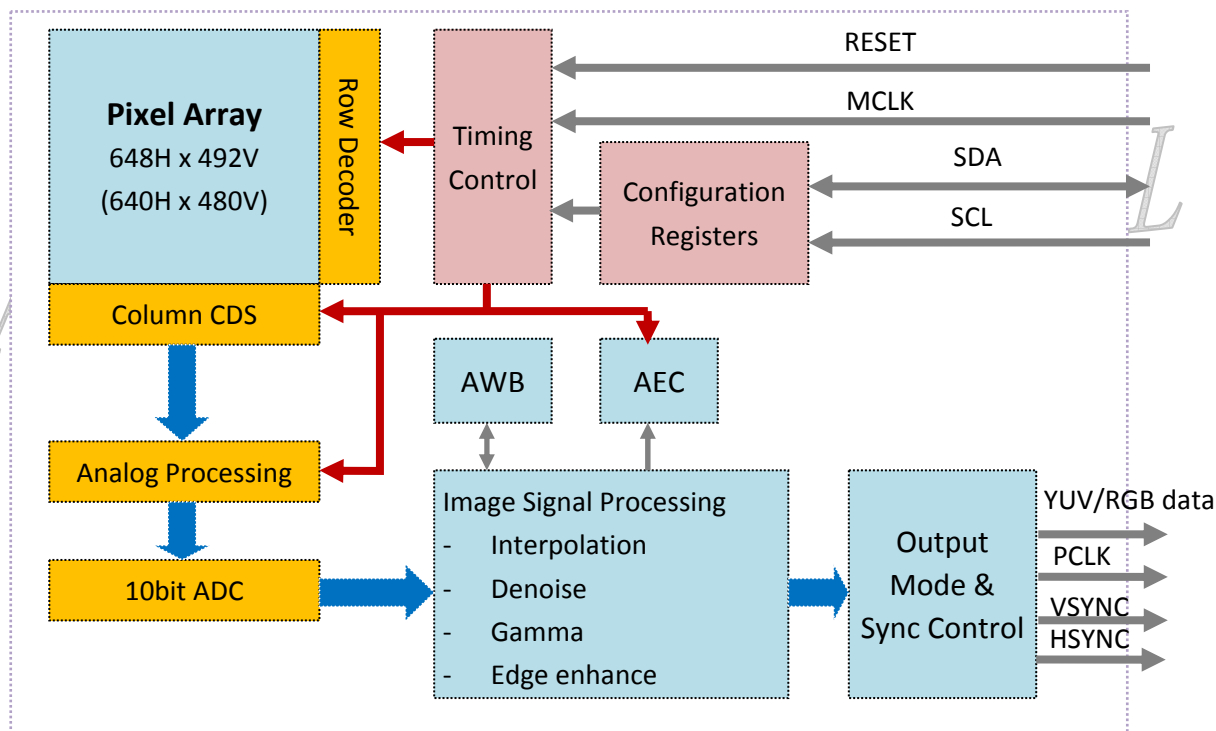


图 1-1 功能框图

## 2. Pixel Array 说明

GC0308 的像素阵列大小为 648 列、488 行，除此之外还有 4 行 dark row。

GC0308 的像素阵列上覆盖着彩色滤光片（Color Filter），并且彩色滤光片以 BG/GR 的方式每行交错排列，

像素的输出按照逐行读出的方式进行，读出的顺序见下图。

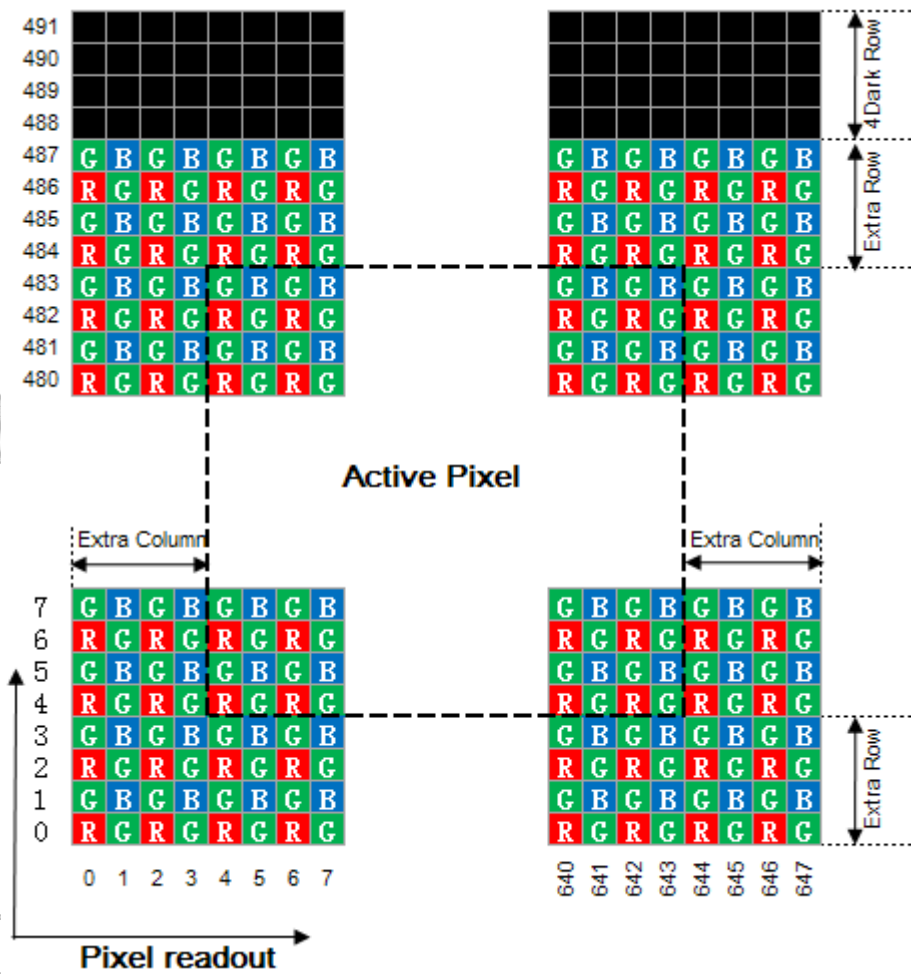


图 2-1 像素阵列图

### 3. 系统应用

### 3.1 外围连接

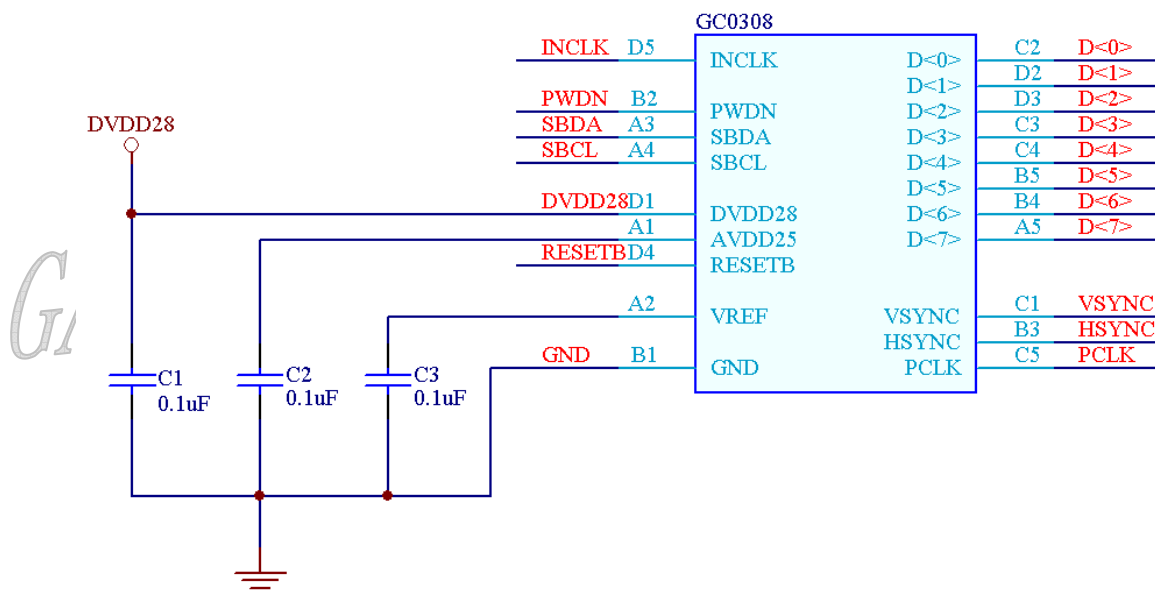


图 3-1 外围连接参考

- ◆ GC0308 芯片只需要单电源供电，DVDD28 = 2.8V，其余电源由芯片内部产生，不需要引出至模组连接器。AVDD25 及内部参考电源（VREF）管脚在模组内部通过电容接地。
- ◆ 电源上附近如图示 C1、C2、C3 滤波电容，容值均为 0.1uF 或 0.47uF。
- ◆ 电容摆放应尽量靠近 Pin 脚。
- ◆ DVDD18 pin 脚没有引出，由芯片内部产生。
- ◆ AGND/DGND 芯片内部相连。
- ◆ SBCL/SBDA pin 外部需要 5-10KΩ 的上拉电阻。

### 3.2 应用时序

- 1) PWDN 与 RESET 均是异步设计，生效时不需要 MCLK pin 有时钟提供。
- 2) PWDN 高有效，Low -> 正常工作，High -> 省电模式  
RESET 低有效，Low -> reset 芯片，High -> 正常工作
- 3) 主时钟必须在 sensor two-wire serial interface 读写前提供。

4) 建议的应用时序如下图所示

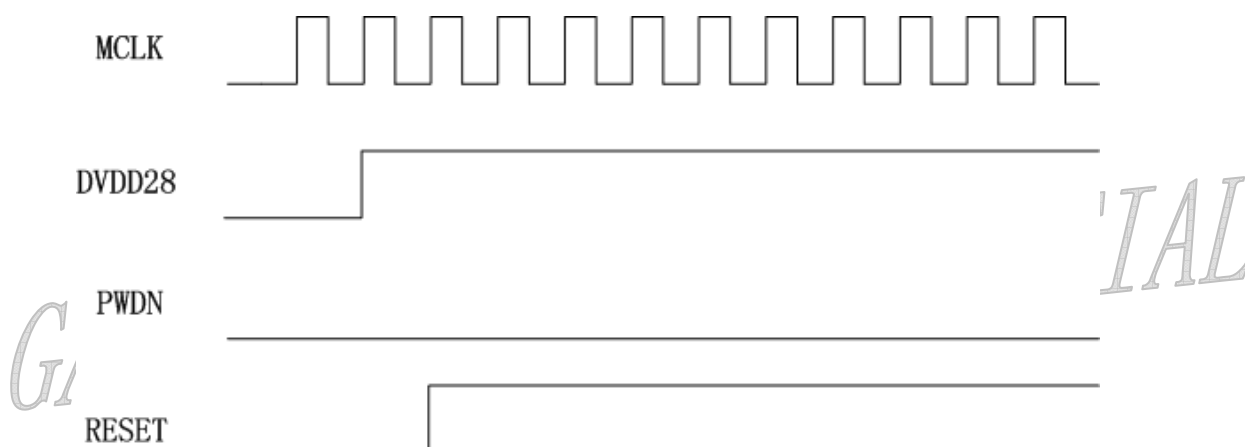


图 3-2 应用时序图

### 3.3 芯片控制

#### 3.3.1 寄存器复位

芯片内部寄存器复位默认值有两种方式：

- 1) RESET pin 接入低电平。
- 2) 将寄存器 0xfe[7]置 1。

#### 3.3.2 Standby 模式控制

使芯片置于 Standby 模式有两种方式：

- 1) PWDN pin 接入高电平。此方式会降低功耗，输出 pin 高阻，寄存器值保持不变。此时寄存器将无法读写。

要恢复 normal 工作模式，只需将 PWDN pin 接入低电平即可。

- 2) 将寄存器 0x1a[0]置 1，0x25 写为 0x00。此方式同样降低功耗，输出 pin 高阻，寄存器值保持不变。此时寄存器是可以读写的。

要恢复 normal 工作模式，需要将 0x1a[0]置 0，0x25 写为 0xff。

### 3.3.3 输出使能控制

GC0308 可以通过写寄存器来控制几组输出 pin 的输出使能。

Function	Register	Description
<b>VSync output enable</b>	0x25[0]	控制 VSYNC pin 输出 0 → VSYNC pin 高阻 1 → VSYNC pin 正常输出
<b>HSync output enable</b>	0x25[1]	控制 HSYNC pin 输出 0 → HSYNC pin 高阻 1 → HSYNC pin 正常输出
<b>PCLK output enable</b>	0x25[2]	控制 PCLK pin 输出 0 → PCLK pin 高阻 1 → PCLK pin 正常输出
<b>Pixel data[7:0] output enable</b>	0x25[3]	控制 data pin 输出 0 → data pin 高阻 1 → data pin 正常输出

表 3-1 输出使能控制

### 3.3.4 输出 Pin 驱动能力

GC0308 可以通过写寄存器来控制输出 pin 的驱动能力。

Function	Register	Description
<b>PCLK PIN 驱动能力</b>	0x1f[1:0]	控制 PCLK pin 输出驱动能力 00 → 2mA 01 → 4mA 10 → 8mA 11 → 10mA
<b>Data PIN 驱动能力</b>	0x1f[3:2]	控制 data pin 输出驱动能力 00 → 2mA 01 → 4mA 10 → 8mA 11 → 10mA
<b>SYNC PIN 驱动能力</b>	0x1f[5:4]	控制 HSYNC/VSYNC pin 输出驱动能力 00 → 4mA 01 → 8mA 10 → 12mA 11 → 16mA

表 3-2 输出驱动能力控制



## 4. 芯片功能方面配置

### 4.1 Pixel Array 控制

GC0308 采用逐行扫描的方式将阵列产生的信号依次输入到模拟信号处理模块中。最开始的行为 0 行。在默认寄存器设置下，Sensor 的阵列数据输出顺序为从下到上，从左到右（参看图 2-1）。

GC0308 可通过寄存器控制扫描顺序，实现镜像/垂直翻转。



原始图像      镜像翻转      垂直翻转      镜像垂直翻转

Function	Register Address	Register Value
正常图像	0x14[1:0]	00
镜像翻转	0x14[1:0]	01
垂直翻转	0x14[1:0]	10
镜像垂直翻转	0x14[1:0]	11

表 4-1 镜像/垂直翻转控制

### 4.2 时钟预分频

外部 MCLK 时钟输入后，通过 clock divider 模块对 MCLK 进行分频，芯片内部工作频率基于分频后的频率。GC0308 最大分频比为 1/8 分频。

Function	Register	Description
MCLK 内部分频比	0x28[6:4]	此值+1 = 实际的分频率，如 7 表示 8 分频。
分频后占空比	0x28[2:0]	分频后高电平个数。如果是 8 分频，0x28 设为 0x77，表示 8 分频后的波形占空比为 H:L = 7:1

一般分频的推荐设置如下表：

Function	Register	Description
内部分频设置	0x28	0x00 -> 不分频 0x11 -> 1/2 MCLK 0x21 -> 1/3 MCLK 0x32 -> 1/4 MCLK 0x42 -> 1/5 MCLK 0x53 -> 1/6 MCLK 0x63 -> 1/7 MCLK 0x74 -> 1/8 MCLK

表 4-2 内部分频设置

4.3 输出时序说明及同步信号控制

4.3.1 输出时序说明

假设帧同步信号 Vsync 低有效，行同步 Hsync 为高有效，而输出格式为 YCbCr 或 RGB565 图像的话，Vsync 和 Hsync 的关系如下：

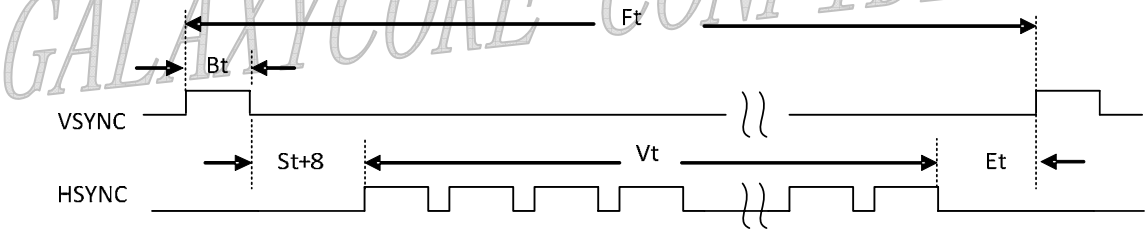


图 4-1 输出时序图

**Ft = VB + Vt + 8** （单位均为 row\_time，row\_time 在下面描述）

VB = Bt + St + Et，一般称为 Vblank/Dummy line，由寄存器 0x0f[7:4]和 0x02 设置，需要大于 St+Et+6

- ◆ Ft -> Frame time，一个帧周期的时间。
- ◆ Bt -> Blank time，Vsync 无效时间。
- ◆ St -> Start time，帧头与第一行有效数据开始之间的时间，由 0x0d 寄存器来设定。
- ◆ Et -> End time，最后一行有效数据与帧尾之间的时间，由 0x0e 寄存器来设定

- ◆  $Vt \rightarrow$  有效行的时间。如 VGA 为 480,  $Vt = win\_height - 8$ ,  $win\_height$  由寄存器 0x09 和 0x0a 所设定（设为 488）。

当  $exp\_time$ （曝光时间） $\leq win\_height + VB$  时， $Bt = VB - St - Et$ 。帧率由  $win\_height + VB$  控制。

当  $exp\_time > win\_height + VB$  时， $Bt = exp\_time - win\_height - St - Et$ 。帧率由  $exp\_time$  决定。

下面是一行时间（ $row\_time$ ）的计算方法：

$$row\_time = Hb + Sh\_delay + win\_width + 4.$$

- ◆  $Hb \rightarrow$  为 HBlank 或 dummy pixel，由 0x0f[3:0] 和 0x01 设定。
- ◆  $Sh\_delay \rightarrow$  由寄存器 0x12 设定。。
- ◆  $win\_width \rightarrow$  0x0b 和 0x0c 所设定，比实际需要的输出尺寸要大 8，如 VGA 要设为 648。

### 4.3.2 同步信号极性控制

VSYNC 为场同步信号，HSYNC 为行同步信号，PCLK 为输出 data 的同步时钟。

GC0308 可以通过寄存器来控制这三个信号的极性。默认配置下，PCLK 下降沿出数据，建议后端 DSP 用 PCLK 的上升沿采集数据。

Function	Register	Description
VSYNC 极性控制	0x26[0]	0 -> 低有效。 1 -> 高有效。表示 VSYNC 为高时 sensor 输出有效数据。
HSYNC 极性控制	0x26[1]	0 -> 低有效。 1 -> 高有效。表示 HSYNC 为高时 sensor 输出有效数据。
PCLK 极性控制	0x26[2]	0 -> 下降沿出 data， default 值。 1 -> 上升沿出 data。

表 4-3 同步信号极性控制

## 4.4 图像窗口设置

GC0308 可以截取任意尺寸( $\leq$ VGA)的窗口输出,且有两种模式实现窗口输出,这两种模式输出小于 VGA 窗口时,视角均会变小。如果想实现视角不变输出 QVGA/QQVGA/CIF/QCIF 等窗口,需要用到 subsample 模式,详见下节“Subsample 输出”

### 1) Windowing 模式。

- ◆ 此模式输出小于 VGA 尺寸时,会加快帧率.如 24M MCLK, CIF->80fps, QVGA->90fps, QQVGA->120fps, 上述为最高帧率。
- ◆ 使用此模式时, anti-flicker 的设置需要重新计算,不能与 VGA 一致。
- ◆ 需要重新配置 measure window 寄存器 (0xf7~0xf9)。
- ◆ 如果想实现高速小尺寸输出,建议用此模式。

Windowing 挖窗口时,用 column start 和 row start 来分别确定要挖窗口起始的 X/Y 坐标,用 window width-8 和 window height-8 (请注意寄存器设置要比实际输出多 8) 来确定所需要窗口的宽度和高度,下表是实现 640x480 和 320x240 的参考设置。

Register	Description	640x480	320x240
0x05	Row start[8]	0x00	0x00
0x06	Row start[7:0]	0x00	0x78
0x07	Column start[9:8]	0x00	0x00
0x08	Column start[7:0]	0x00	0xa0
0x09	Window height[8]	0x01	0x01
0x0a	Window height[7:0]	0xe8	0x48
0x0b	Window width[9:8]	0x02	0x00
0x0c	Window width[7:0]	0x88	0xf8

表 4-4 windowing 模式输出设置

### Measure window 配置:

Measure window 配置出来的窗口是用来做图像处理(如 AEC/AWB)的,所以设置的窗口比 Windowing 配出窗口(0x05~0x0c)略小即可。请注意 0xf7~0xfa 四个寄存器值不要为 0。

用 Meas\_win\_x0 和 Meas\_win\_y0 来分别确定要窗口起始的 X/Y 坐标，用 Meas\_win\_x1 和 Meas\_win\_y1 来确定窗口的宽度和高度

$\text{Meas\_win\_x0} = \text{Reg}[0xf7] * 4$ , // Reg[0xf7] 表示寄存器 0xf7 的值。

$\text{Meas\_win\_y0} = \text{Reg}[0xf8] * 4$

$\text{Meas\_win\_x1} = \text{Reg}[0xf9] * 4$

$\text{Meas\_win\_y1} = \text{Reg}[0xfa] * 4$

## 2) Crop window 模式。

- ◆ 此模式输出小于 VGA 尺寸时，帧率与 VGA 相同。
- ◆ Anti-flicker 的配置与 VGA 一致，不需要重新计算。
- ◆ Windowing (0x05-0x0c) 寄存器按 640x480 的配置。
- ◆ 不需要重新配置 measure window 寄存器 (0xf7~0xf9)。
- ◆ 要使用 crop window 模式，需要将 0x46[7]置 1。

用 Crop window 模式挖窗口时，用 Out window x0 和 Out window y0 来分别确定要挖窗口起始的 X/Y 坐标，用 Out window width 和 Out window height 来确定所需要窗口的宽度和高度，下表是实现 320x240 的参考设置。

Register	Description	320x240 Sample
<b>0x46</b>	[7] enable crop window mode [5:4] Out window y0[9:8] [2:0] Out window x0[10:8]	0x80
<b>0x47</b>	Out window y0[7:0]	0x78
<b>0x48</b>	Out window x0[7:0]	0xa0
<b>0x49</b>	Out window height[8]	0x00
<b>0x4a</b>	Out window height[7:0]	0xf0
<b>0x4b</b>	Out window width[9:8]	0x01
<b>0x4c</b>	Out window width[7:0]	0x40

表 4-5 Crop window 模式输出设置

## 4.5 Subsample 输出

Subsample 输出是在 windowing (0x05-0x0c 寄存器) 确定好图像窗口尺寸后, 采用抽点的方式输出更小尺寸的数据, 最终输出图像尺寸的行/列可以为设定窗口行/列的 1/2, 1/3, 1/4, 1/5, 1/6, 1/7 (表示行为原始尺寸的 1/7, 列也为原始尺寸的 1/7), 同时也能实现 3/5, 2/3, 4/7 等采样率。

- ◆ Subsample 后的帧率与原始尺寸的帧率一样, 不会加快帧率。
- ◆ Subsample 后的图像视角与原始尺寸一样, 不会缩小图像范围。
- ◆ 不需要重新配置 measure window 寄存器 (0xf7~0xf9)。
- ◆ 建议用 subsample 方式输出 QVGA/CIF/QQVGA, 可以保持图像视角。

下面给出几种常用 subsample 的设置。

(P1:0x54 表示 0x54 寄存器是 page1 的, page 选择由 P0:0xfe[1:0] 设置, 设为 0x00 表示 page0, 设为 0x01 表示 page1)

Subsample Ratio	P1:0x53[7]	P1:0x54	P1:0x56	P1:0x57	P1:0x58	P1:0x59	P1:0x55[0]
1/2	1	0x22	0x00	0x00	0x00	0x00	1
1/3	1	0x33	0x00	0x00	0x00	0x00	1
1/4	1	0x44	0x00	0x00	0x00	0x00	1
2/3	1	0x33	0x02	0x00	0x02	0x00	1
3/5	1	0x55	0x02	0x04	0x02	0x04	1
4/7	1	0x77	0x02	0x46	0x02	0x46	1

## 4.6 Anti\_flicker 与 HB, VB 与 CLK 的关系。

### 4.6.1 Anti\_flicker 计算

在 GC0308 中, 消除灯管频率导致的 flicker 问题, 需要同时配置 exposure step register (0xe3) 和 Hb ({0x0f[3:0], 0x01}), Vb ({0x0f[7:4], 0x02}), exp\_level(0xe4~0xeb) 来实现。消除 flicker 的原理即是曝光时间是灯管周期的整数倍。

计算方法为:  $\text{step} * \text{row\_time} = N * T$

- 1) step 便是 0xe3 要写入的值。
- 2) row\_time 在 4.3.1 章节（输出时序说明）中有详细阐述
- 3) N 为大于等于 1 的整数。
- 4) T 为灯管频率的周期。

即工频为 50Hz 时,  $T=0.01s=10ms$

工频为 60Hz 时,  $T=1/120s=8.333ms$

另外, 当曝光时间小于一个 step 的时候, 还是会出 flicker, 这时应该配合 VB 将相位固定住。

公式为:  $(488+Vb) * rowtime = N * T$

## 4.6.2 Exp Level 配置

GC0308 可以配置 4 档 Exp\_level, 用户可以通过修改 0xec[5:4]来设置用哪一档 exp\_level (范围为 0-3), 设置完成后, AEC 最大曝光时间只能达到 exp\_level 所设定的值, 从而达到控制帧率的目的。

Exp\_level 的配置很简单, 只要用是所计算出来的 step 的整数倍即可, 可按照想要的帧率配置, 每个 level 需要配两个寄存器, 如 0xe4[3:0]为高位, 0xe5 为低位。

Exp\_level0 { 0xe4[3:0], 0xe5 }

Exp\_level1 { 0xe6[3:0], 0xe7 }

Exp\_level2 { 0xe8[3:0], 0xe9 }

Exp\_level3 { 0xea[3:0], 0xeb }

## 4.7 各种输出模式

**YUV/RGB565 输出设置:**

Output Format		0x24	最大输出尺寸
YUV422	CbYCrY	0xa0	644 x 482
	CrYCbY	0xa1	644 x 482
	YCbYCr	0xa2	644 x 482
	YCrYCb	0xa3	644 x 482
RGB565		0xa6	644 x 482

**RawRGB 输出设置:**



Output Format	0x24	0x20	0x21	0x22	0xd2[7]	0x29[3]	最大输出尺寸
Raw RGB(LSC out)	0xb9	0x00	0x00	0x00	0	0	650 x 486
Raw RGB(DNDD out)	0xb8	0x00	0x00	0x00	0	0	648 x 484
Raw RGB(CISCTL out)	---	0x00	0x00	0x00	0	1	648 x 488

- ◆ CISCTL out raw RGB 是没有经过任何 ISP 处理的数据输出。
- ◆ LSC out raw RGB – 可以使能 AEC/AWB/黑电平校正/镜头暗角补偿模块。
- ◆ DNDD out raw RGB – 可以使能 AEC/AWB/黑电平校正/去噪/去坏点/镜头暗角补偿模块。
- ◆ LSC out/ DNDD out raw RGB 要使能某些功能模块，需要配置使能这些模块的寄存器。
- ◆ 要达到上述最大输出尺寸需要将 0x2f[1] (inbuf\_en) 置为 1，0x2d[0] (more\_boundary) 置为 1。

## 5. 图像功能模块调试

### 5.1 AEC

Function	Address	Value	Description
AEC enable	0xd2[7]	1'b0	使能 AEC 模块
Target Y	0xd3	0x50	AEC 目标亮度值
AEC action period	0xd1[6:4]	2	AEC 每隔几帧调一次。
Exp level	0xe4-0xeb	--	配置 exp level，具体见 4.6.2
Max exp level	0xec[5:4]	--	选择 exp level 0-3 中的一档
Exp min	0xed	0x04	设置 AEC 自动调节的最小 exp
Auto pre gain	0x51	--	AEC 使能时，为只读。
Auto post gain	0x52	--	AEC 不使能时，可读写。
Pre gain limit	0xef	0x40	The gain limit of Auto_pre_gain.
Post gain limit	0xee	0x80	The gain limit of Auto_post_gain.



## 5.2 AWB

Function	Address	Value	Description
AWB enable	0x22[1]	1'b0	使能 AWB 模块
AWB_R_gain	0x5a	--	AWB 使能时，为只读。 AWB 不使能时，可读写。。
AWB_G_gain	0x5b	--	
AWB_B_gain	0x5c	--	
AWB_R_gain_limit	P1: 0x1c	0x80	The gain limit of AWB_R_gain.
AWB_G_gain_limit	P1: 0x1d	0x80	The gain limit of AWB_G_gain.
AWB_B_gain_limit	P1: 0x1e	0x80	The gain limit of AWB_B_gain.
AWB action period	P1: 0x13 [5:4]	1'b0	AWB 每隔几帧调一次。
R_ratio	0x57	0x80	The gain on R channel after AWB
G_ratio	0x58	0x80	The gain on G channel after AWB
B_ratio	0x59	0x80	The gain on B channel after AWB

## 5.3 LSC (Lens Shading Compensation)

LSC 的算法公式为:  $LSC\ Gain = 1 + b2 * R^2 + b4 * R^4$  (R 为与 LSC center 的距离),  
调试时 0xb2 用于粗调, 0xb4 用于微调。

Function	Address	Value	Description
LSC enable	0x20[0]	1'b0	使能 LSC 模块
lsc_row_center	0x91[6:0]	0x3c	The row center of lens shading. x4
lsc_col_center	0x92	0x50	The column center of lens shading. x4
LSC red b2	0x8b	0x24	公式中 b2 项的值, 对应于 Red 分量。
LSC green b2	0x8c	0x20	公式中 b2 项的值, 对应于 green 分量。
LSC blue b2	0x8d	0x20	公式中 b2 项的值, 对应于 blue 分量。
LSC red b4	0x8e	0x10	公式中 b4 项的值, 对应于 Red 分量。
LSC green b4	0x8f	0x10	公式中 b4 项的值, 对应于 green 分量。
LSC blue b4	0x90	0x10	公式中 b4 项的值, 对应于 blue 分量。

## 5.4 去坏点 (delect defective points)

Function	Address	Value	Description
----------	---------	-------	-------------

<b>DD enable</b>	0x20[1]	1'b0	使能去坏点模块
<b>Dark threshold</b>	0x65[7:4]	0x01	暗点 threshold
<b>Bright threshold</b>	0x65[3:0]	0x07	亮点 threshold

注：要多去坏点：减小 Dark threshold / Bright threshold.

反之，要少去快点：放大 Dark threshold / Bright threshold.

## 5.5 去噪 (denoise)

Function	Address	Value	Description
<b>Denoise enable</b>	0x20[2]	1'b0	使能去噪模块
<b>DN_bilat_b</b>	0x62	0x0c	越大去噪越强
<b>DN_C_weight_base</b>	0x64[3:0]	0x03	越大去噪越弱
<b>C_weight_adaptive_ratio</b>	0x61[5:4]	2'b01	越大去噪越弱
<b>dn_b_mode_ratio</b>	0x61[1:0]	2'b10	越大去噪越强

## 5.6 边缘加强 (Edge enhancement)

Function	Address	Value	Description
<b>EE enable</b>	0x20[4]	1'b0	使能边缘加强模块
<b>Edge1_effect</b>	0x77[7:4]	--	Edge1_effect
<b>Edge2_effect</b>	0x77[3:0]	--	Edge2_effect
<b>ASDE_auto_EE1_effect_start</b>	0x6e[7:4]	0xa5	[7:4] ASDE_auto_EE1_effect_start, 4
<b>ASDE_auto_EE2_effect_start</b>	0x6f[7:4]	0xa8	[7:4] ASDE_auto_EE2_effect_start, 4

注：auto\_EE 打开时，要加大边缘加强，则增大 0x6e[7:4]以及 0x6f[7:4]。

Auto\_EE 关闭时，要加大去噪能力，则增大 0x77.

## 5.7 对比度

Function	Address	Value	Description
<b>Luma_contrast</b>	0xb3	0x40	对比度调整，0x40 为一倍
<b>Contrast_center</b>	0xb4	0x80	Contrast center value

## 5.8 饱和度

Function	Address	Value	Description
<b>Global saturation</b>	0xb0	0x40	Float 2.6 bits, 0x40 为一倍
<b>Cb saturation</b>	0xb1	0x20	Float S2.5 bits, 0x20 为一倍
<b>Cr saturation</b>	0xb2	0x20	Float S2.5 bits, 0x20 为一倍

## 5.9 特效参数

Function	Address	Value	Description
<b>Fixed_CbCr_en</b>	0x23[1]	1'b0	When enable, the Cb/Cr is fixed by the setting of 0x78/0x79。
<b>Fixed_Cb</b>	0xba	0x00	When Fixed_CbCr_en, the Cb is set by this register.
<b>Fixed_Cr</b>	0xbb	0x00	When Fixed_CbCr_en, the Cr is set by this register.
<b>Inverse_color</b>	0x23[0]	1'b0	When enable, the color of image is inversed.

## 5.10 Gamma 调整

GC0308 有两种 gamma: RGB gamma 和 Y gamma, 分别是在 RGB 域和 YUV 域上做 Gamma, 可以共同使用。

Function	Address	Value	Description
<b>RGB Gamma_en</b>	0x20[6]	1'b1	The RGB gamma function enable bit.
<b>RGB Gamma curve</b>	0x9f~0xaf	--	The 17 points of RGB gamma curve.
<b>Y Gamma_en</b>	0x21[1]	1'b1	The Y gamma function enable bit.
<b>RGB Gamma curve</b>	0xc0~0xcc	--	The 13 points of Y gamma curve.

## 5.11 Color Matrix

Function	Address	Value	Description
<b>CC enable</b>	0x20[5]	1'b1	The CC function enable bit.
<b>CC Matrix C11</b>	0x93	0x44	R channel coefficient 1, S1.6
<b>CC Matrix C12</b>	0x94	0xfe	R channel coefficient 2, S1.6

<b>CC Matrix C13</b>	0x95	0xfe	R channel coefficient 3, S1.6
<b>CC Matrix C21</b>	0x96	0xfe	G channel coefficient 1, S1.6
<b>CC Matrix C22</b>	0x97	0x44	G channel coefficient 2, S1.6
<b>CC Matrix C23</b>	0x98	0xfe	G channel coefficient 3, S1.6
<b>CC Matrix C41</b>	0x9c	0x00	R channel offset coefficient, S4
<b>CC Matrix C42</b>	0x9d	0x00	G channel offset coefficient, S4
<b>CC Matrix C43</b>	0x9e	0x00	B channel offset coefficient, S4

注：CC 矩阵算法的公式如下：

$$[R' G' B'] = [R G B] \times \begin{bmatrix} C11 & C12 & C13 \\ C21 & C22 & C23 \\ C31 & C32 & C33 \end{bmatrix} + [C41 C42 C43]$$

其中  $C31 = 1 - C21 - C11$ ,

$C32 = 1 - C22 - C12$ ,

$C33 = 1 - C23 - C13$ ,