

```
&usOrgX, &usOrgY);
```

```
G2D_CheckFifo(17);
```

```
Outp16(rG2D_ROT_OC_X, usOrgX);
```

```
Outp16(rG2D_ROT_OC_Y, usOrgY);
```

```
uRotDegree =
```

```
    (eRotDegree == ROT_0) ? G2D_ROTATION_0_DEG_BIT :
```

```
    (eRotDegree == ROT_90) ? G2D_ROTATION_90_DEG_BIT :
```

```
    (eRotDegree == ROT_180) ? G2D_ROTATION_180_DEG_BIT : G2D_ROTATION_270_DEG_BIT;
```

```
Outp32(rG2D_ROT_MODE, uRotDegree);
```

```
}
```

```
//BitBlt旋转设置
```

```
void G2D_RotateWithBitBlt(
```

```
    u16 usSrcX1, u16 usSrcY1, u16 usSrcX2, u16 usSrcY2,
```

```
    u16 usDestX1, u16 usDestY1, ROT_DEG eRotDegree)
```

```
{
```

```
    u16 usOrgX, usOrgY;
```

```
    u32 uRotDegree;
```

```
    G2D_GetRotationOrgXY(usSrcX1, usSrcY1, usSrcX2, usSrcY2, usDestX1, usDestY1, eRotDegree,
```

```
&usOrgX, &usOrgY);
```

```
    G2D_CheckFifo(17);
```

```
    Outp16(rG2D_ROT_OC_X, usOrgX);
```

```
    Outp16(rG2D_ROT_OC_Y, usOrgY);
```

```
    uRotDegree =
```

```
        (eRotDegree == ROT_0) ? G2D_ROTATION_0_DEG_BIT :
```

```
        (eRotDegree == ROT_90) ? G2D_ROTATION_90_DEG_BIT :
```

```
        (eRotDegree == ROT_180) ? G2D_ROTATION_180_DEG_BIT : G2D_ROTATION_270_DEG_BIT ;
```

```
    Outp32(rG2D_ROT_MODE, uRotDegree);
```

```
G2D_BitBlt(usSrcX1, usSrcY1, usSrcX2, usSrcY2, usSrcX1, usSrcY1, usSrcX2, usSrcY2, false);
}
```

6. 裁剪

裁剪将放弃裁剪窗口以外的像素（旋转后）。放弃的像素的色彩数据不写入帧缓冲器。

相关寄存器如表18-10所示。

表18-10 裁剪相关寄存器

| | |
|-----------|---------------|
| CW_LT_REG | 裁剪窗口最左上方的点的坐标 |
| CW_RB_REG | 裁剪窗口最右下方的点的坐标 |

18.4 2D 图形寄存器定义

地址映射

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|---------------|------------|-----|--------------------------|-------------|
| 通用寄存器 | | | | |
| INTEN_REG | 0x76100004 | 读/写 | 中断有效寄存器。 | 0x0000_0000 |
| FIFO_INTC_REG | 0x76100008 | 读/写 | 中断控制寄存器。 | 0x0000_0018 |
| INTC_PEND_REG | 0x7610000C | 读/写 | 中断控制等待寄存器。 | 0x0000_0000 |
| FIFO_STAT_REG | 0x76100010 | 读 | 指令FIFO状态寄存器。 | 0x0000_0600 |
| FB_BA_REG | 0x76100020 | 读/写 | 帧缓冲器基址寄存器。 | 0x0000_0000 |
| 指令寄存器 | | | | |
| CMD0_REG | 0x76100100 | 写 | 用于线/点绘图的指令寄存器。 | - |
| CMD1_REG | 0x76100104 | 写 | 用于BitBLT的指令寄存器。 | - |
| CMD2_REG | 0x76100108 | 写 | 用于主机到屏Bitblt传送开始点的指令寄存器。 | - |
| CMD3_REG | 0x7610010C | 写 | 用于主机到屏Bitblt传送恢复的指令寄存器。 | - |
| CMD4_REG | 0x76100110 | 写 | 用于彩色扩展的指令寄存器(主 | - |

| | | | | |
|----------------|------------|-----|--------------------------|--------------|
| | | | 机到屏，字型开始点)。 | |
| CMD5_REG | 0x76100114 | 写 | 用于彩色扩展的指令寄存器(主机到屏，字型恢复)。 | - |
| CMD7_REG | 0x7610011C | 写 | 用于彩色扩展的指令寄存器(存储器到屏)。 | - |
| 指令资源寄存器 | | | | |
| COLOR_MODE_REG | 0x76100200 | 读/写 | 色彩模式寄存器。 | 0x0000_0008 |
| HORI_REG_REG | 0x76100204 | 读/写 | 水平分辨率寄存器。 | 0x0000_0000 |
| SC_WIN_REG | 0x76100210 | 读/写 | 屏幕裁剪窗口寄存器。 | 0x0000_0000 |
| SC_WIN_X_REG | 0x76100214 | 写 | 屏幕裁剪窗口的最大X寄存器。 | 0x0000_0000 |
| SC_WIN_Y_REG | 0x76100218 | 写 | 屏幕裁剪窗口的最大Y寄存器。 | 0x0000_0000 |
| CW_LT_REG | 0x76100220 | 读/写 | 裁剪窗口左上方坐标。 | 0x0000_0000 |
| CW_LT_X_REG | 0x76100224 | 写 | 裁剪窗口左X坐标。 | 0x0000_0000- |
| CW_LT_Y_REG | 0x76100228 | 写 | 裁剪窗口上Y坐标。 | 0x0000_0000- |
| CW_RB_REG | 0x76100230 | 读/写 | 裁剪窗口右下方坐标。 | 0x0000_0000 |
| CW_RB_X_REG | 0x76100234 | 写 | 裁剪窗口右X坐标。 | 0x0000_0000- |
| CW_RB_Y_REG | 0x76100238 | 写 | 裁剪窗口下Y坐标。 | 0x0000_0000- |
| COORD0_REG | 0x76100300 | 读/写 | 坐标0寄存器。 | 0x0000_0000 |
| COORD0_X_REG | 0x76100304 | 写 | 坐标0的X坐标。 | 0x0000_0000- |
| COORD0_Y_REG | 0x76100308 | 写 | 坐标0的Y坐标。 | 0x0000_0000- |
| COORD1_REG | 0x76100310 | 读/写 | 坐标1寄存器。 | 0x0000_0000 |
| COORD1_X_REG | 0x76100314 | 写 | 坐标1的X坐标。 | 0x0000_0000- |
| COORD1_Y_REG | 0x76100318 | 写 | 坐标1的Y坐标。 | 0x0000_0000- |
| COORD2_REG | 0x76100320 | 读/写 | 坐标2寄存器。 | 0x0000_0000 |
| COORD2_X_REG | 0x76100324 | 写 | 坐标2的X坐标。 | 0x0000_0000- |
| COORD2_Y_REG | 0x76100328 | 写 | 坐标2的Y坐标。 | 0x0000_0000- |
| COORD3_REG | 0x76100330 | 读/写 | 坐标3寄存器。 | 0x0000_0000 |
| COORD3_X_REG | 0x76100334 | 写 | 坐标3的X坐标。 | 0x0000_0000- |

| | | | | |
|----------------|---------------------------|-----|--------------------|--------------|
| COORD3_Y_REG | 0x76100338 | 写 | 坐标3的Y坐标。 | 0x0000_0000- |
| ROT_OC_REG | 0x76100340 | 读/写 | 旋转原点坐标。 | 0x0000_0000 |
| ROT_OC_X_REG | 0x76100344 | 写 | 旋转原点坐标的X坐标。 | 0x0000_0000- |
| ROT_OC_Y_REG | 0x76100348 | 写 | 旋转原点坐标的Y坐标。 | 0x0000_0000- |
| ROTATE_REG | 0x7610034C | 读/写 | 旋转模式寄存器。 | 0x0000_0001 |
| ENDIA_READSIZE | 0x76100350 | 读/写 | 大端或小端选择，读取的顺序。 | 0x0000_0001 |
| X_INCR_REG | 0x76100400 | 读/写 | X增量寄存器。 | 0x0000_0000 |
| Y_INCR_REG | 0x76100404 | 读/写 | Y增量寄存器。 | 0x0000_0000 |
| ROP_REG | 0x76100410 | 读/写 | 光栅操作寄存器。 | 0x0000_0000 |
| ALPHA_REG | 0x76100420 | 读/写 | α 值，衰退偏移量。 | 0x0000_0000 |
| FG_COLOR_REG | 0x76100500 | 读/写 | 前景色/ α 寄存器。 | 0x0000_0000 |
| BG_COLOR_REG | 0x76100504 | 读/写 | 背景色寄存器。 | 0x0000_0000 |
| BS_COLOR_REG | 0x76100508 | 读/写 | 蓝屏颜色寄存器。 | 0x0000_0000 |
| PATTERN_REG | 0x76100600 ~0x7610067C | 读/写 | 图案寄存器。 | - |
| PATOFF_REG | 0x76100700 | 读/写 | 图案偏移量XY寄存器。 | 0x0000_0000 |
| PATOFF_X_REG | 0x76100704 | 写 | 图案偏移量X寄存器。 | 0x0000_0000 |
| PATOFF_Y_REG | 0x76100708 | 写 | 图案偏移量Y寄存器。 | 0x0000_0000 |

18.4.1. 通用中断有效寄存器 (INTEN_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|-----------|------------|-----|----------|-------------|
| INTEN_REG | 0x76100004 | 读/写 | 中断有效寄存器。 | 0x0000_0000 |

| INTEN_REG | 位 | 描述 | 初始状态 |
|-----------|---------|--------------------------------|------|
| Reserved | [31:10] | | 0x0 |
| DF | [10] | 绘图引擎完成中断有效。 | 0x0 |
| F | [9] | 所有的指令完成中断有效。当所有指令执行后（指令FIFO中没有 | 0x0 |

| | | | |
|----------|-------|---|-----|
| | | 指令)，设置该位。 | |
| OV | [8] | 溢出中断有效。当指令FIFO溢出，该位被设置。 | 0x0 |
| Reserved | [7:1] | | 0x0 |
| E | [0] | FIFO电平中断有效。如果该位设置为1，当FIFO_INT_LEVEL和FIFO_NO_USED相同时，图形引擎将INTREQ信号标志为高位。 | 0x0 |

18.4.2. 通用 FIFO 中断控制寄存器(FIFO_INTC_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|---------------|------------|-----|----------|-------------|
| FIFO_INTC_REG | 0x76100008 | 读/写 | 中断控制寄存器。 | 0x0000_0018 |

| FIFO_INTC_REG | 位 | 描述 | 初始值 |
|----------------|--------|---|------|
| Reserved | [31:6] | | 0x0 |
| FIFO_INT_LEVEL | [5:0] | 当 FIFO 的数目的使用是 FIFO_INT_LEVEL，中断启动位被设置为 1 时，图形引擎请求中断。 | 0x18 |

18.4.3. 通用断控制等待寄存器(INT_PEND_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|---------------|------------|-----|------------|-------------|
| INTC_PEND_REG | 0x7610000C | 读/写 | 中断控制等待寄存器。 | 0x0000_0000 |

| INTC_PEND_REG | 位 | 描述 | 初始值 |
|-----------------|--------|--|-----|
| CLRSEL | [31] | 水平中断&脉冲中断模式选择： 1：水平模式选择（中断清除启动） 0：脉冲中断模式选择 | |
| Reserved | 30:11] | | |
| INTP_DE_FIN | [10] | 图形绘图引擎完成。 | |
| INTP_FINISH_ALL | [9] | 图形引擎空闲状态。 | |
| INTP_OVERFLOW | [8] | 溢出中断。 | |

| | | | |
|-----------------|-------|--|--|
| Reserved | [7:1] | | |
| INTP_FIFO_LEVEL | [0] | 当 FIFO_INT_LEVEL 与 FIFO_NO_USED 一样时，这个位将被设置。 | |

18.4.4. 通用指令 FIFO 状态寄存器(FIFO_STAT_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|---------------|------------|-----|----------------|-------------|
| FIFO_STAT_REG | 0x76100010 | 读 | 指令 FIFO 状态寄存器。 | 0x0000_0600 |

| FIFO_STAT_REG | 位 | 描述 | 初始值 |
|---------------|---------|--------------|-----|
| Reserved | [31:13] | | 0x0 |
| DE_FIN | [10] | 图形绘图引擎完成。 | 1 |
| G2D_IDLE | [9] | 图形引擎空闲状态。 | 1 |
| OVR_INT | [8] | 溢出中断。 | 0 |
| Reserved | [7:6] | | 0x0 |
| FIFO_NO_USED | [5:0] | 使用的 FIFO 数目。 | 0x0 |

18.4.5. 通用帧缓冲器基址寄存器(FB_BA_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|-----------|------------|-----|------------|-------------|
| FB_BA_REG | 0x76100020 | 读/写 | 帧缓冲器基址寄存器。 | 0x0000_0000 |

| FB_BA_REG | 位 | 描述 | 初始值 |
|--------------|---------|--|-----|
| FrameBufAddr | [31:10] | 帧缓冲器地址上面的 22 位。8 位最高有效位 (MSB) 决定着帧缓冲器的上限。例如，如果用户设置帧缓冲器地址为 0x60800000，则最大存储器分配帧缓冲器是 [0x60800000, 0x60FFFFFF]。 | 0x0 |
| Reserved | [9:0] | 最低为 10 的帧缓冲器地址的位强制置为 0，这意味帧缓冲器地址应该定位到 1KB。 | 0x0 |

18.4.6. 指令线绘图寄存器 (CMD0_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|----------|------------|-----|-------------|-----|
| CMD0_REG | 0x76100100 | 写 | 用于线/点绘图寄存器。 | - |

| CMD0_REG | 位 | 描述 | 初始值 |
|----------|---------|-------------------------|-----|
| 保留 | [31:10] | | - |
| D | [9] | 0: 绘图最后的点 1: 未绘图最后的点 | - |
| M | [8] | 0: 主轴是 Y 1: 主轴是 X | - |
| Reserved | [7:2] | | - |
| L | [1] | 0: 无关 1: 线绘图 | - |
| P | [0] | 0: 无关 1: 点绘图 | - |

18.4.7. 指令 BITBLT 寄存器 (CMD1_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|----------|------------|-----|-------------------|-----|
| CMD1_REG | 0x76100104 | 写 | 用于 BitBLT 的指令寄存器。 | - |

| CMD1_REG | 位 | 描述 | 初始值 |
|----------|--------|-----------------------|-----|
| Reserved | [31:2] | | - |
| S | [1] | 0: 无关 1: 伸展 BitBLT | - |
| N | [0] | 0: 无关 1: 标准 BitBLT | - |

18.4.8. 指令主机到屏幕优先 BITBLT 寄存器 (CMD2_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|----------|------------|-----|----------------------------|-----|
| CMD2_REG | 0x76100108 | 写 | 用于主机到屏幕 Bitblt 传输优先的指令寄存器。 | - |

| CMD2_REG | 位 | 描述 | 初始值 |
|----------|--------|--------------|-----|
| Data | [31:0] | BitBLT 优先数据。 | - |

18.4.9. 指令主机到屏幕连续 BITBLT 寄存器 (CMD3_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|----------|------------|-----|----------------------------|-----|
| CMD3_REG | 0x7610010C | 写 | 用于主机到屏幕 Bitblt 传输连续的指令寄存器。 | - |

| CMD3_REG | 位 | 描述 | 初始值 |
|----------|--------|--------------|-----|
| Data | [31:0] | BitBLT 连续数据。 | - |

18.4.10. 指令主机到屏幕优先颜色扩充寄存器 (CMD4_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|----------|------------|-----|---------------------------|-----|
| CMD4_REG | 0x76100110 | 写 | 用于颜色扩充的指令寄存器（主机到屏幕，字体优先）。 | - |

| CMD4_REG | 位 | 描述 | 初始值 |
|----------|--------|-----------|-----|
| Data | [31:0] | 压缩格式位图数据。 | - |

18.4.11. 指令主机到屏幕连续颜色扩充寄存器 (CMD5_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|----------|------------|-----|---------------------------|-----|
| CMD5_REG | 0x76100114 | 写 | 用于颜色扩充的指令寄存器（主机到屏幕，字体连续）。 | - |

| CMD5_REG | 位 | 描述 | 初始值 |
|----------|--------|-----------|-----|
| Data | [31:0] | 压缩格式位图数据。 | - |

18.4.12. 指令内存到屏幕颜色扩充寄存器 (CMD7_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|----------|------------|-----|----------------------|-----|
| CMD7_REG | 0x7610011C | 写 | 用于颜色扩充的指令寄存器（内存到屏幕）。 | - |

| CMD7_REG | 位 | 描述 | 初始值 |
|----------------|---------|--------------|-----|
| Reserved | [31:25] | | - |
| Memory Address | [24:2] | 字定位地址到点位图数据。 | - |
| Reserved | [1:0] | | - |

18.4.13. 指令资源颜色模式 (COLOR_MODE_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|----------------|------------|-----|----------|-------------|
| COLOR_MODE_REG | 0x76100200 | 读/写 | 色彩模式寄存器。 | 0x0000_0008 |

| COLOR_MODE_REG | 位 | 描述 | 初始值 |
|----------------|--------|-------|-----|
| Reserved | [31:4] | | 0x0 |
| C3 | [3] | 24/32 | 0x1 |
| C2 | [2] | 18 | 0x0 |

| | | | |
|----|-----|----|-----|
| C1 | [1] | 16 | 0x0 |
| C0 | [0] | 15 | 0x0 |

18.4.14. 指令资源水平分辨率 (HORI_RES_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|--------------|------------|-----|-----------|-------------|
| HORI_RES_REG | 0x76100204 | 读/写 | 水平分辨率寄存器。 | 0x0000_0000 |

| HORI_RES_REG | 位 | 描述 | 初始值 |
|--------------|----------|------------------|-----|
| Reserved | [31: 12] | | 0x0 |
| HoriRes | [11: 0] | 水平分辨率（应为 4 的倍数）。 | 0x0 |

18.4.15. 指令资源屏幕剪切窗口 (SC_WIN_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|------------|------------|-----|------------|-------------|
| SC_WIN_REG | 0x76100210 | 读/写 | 屏幕剪切窗口寄存器。 | 0x0000_0000 |

| SC_WIN_REG | 位 | 描述 | 初始值 |
|------------|---------|--------------|-----|
| Reserved | [31:27] | | 0x0 |
| MaxSY | [26:16] | 最大屏幕剪切 Y 窗口。 | 0x0 |
| Reserved | [15:11] | | 0x0 |
| MaxSX | [10:0] | 最大屏幕剪切 X 窗口。 | 0x0 |

18.4.16. 指令资源屏幕剪切最大 X 窗口 (SC_WIN_X_REG)

| 寄存器 | 偏移量 | 读/写 | 描述 | 复位值 |
|--------------|------------|-----|----------------|-------------|
| SC_WIN_X_REG | 0x76100214 | 写 | 屏幕剪切的 X 窗口寄存器。 | 0x0000_0000 |