

NAME

blit – Copies a rectangular area from one bitmap to another. Allegro game programming library.

SYNOPSIS

```
#include <allegro.h>
```

```
void blit(BITMAP *source, BITMAP *dest, int source_x, int source_y, int dest_x, int dest_y, int
width, int height);
```

DESCRIPTION

Copies a rectangular area of the source bitmap to the destination bitmap. The `source_x` and `source_y` parameters are the top left corner of the area to copy from the source bitmap, and `dest_x` and `dest_y` are the corresponding position in the destination bitmap. This routine respects the destination clipping rectangle, and it will also clip if you try to blit from areas outside the source bitmap. Example:

```
BITMAP *bmp;
...
/* Blit src on the screen. */
blit(bmp, screen, 0, 0, 0, 0, bmp->w, bmp->h);

/* Now copy a chunk to a corner, slightly outside. */
blit(screen, screen, 100, 100, -10, -10, 25, 30);
```

You can blit between any parts of any two bitmaps, even if the two memory areas overlap (ie. source and dest are the same, or one is sub-bitmap of the other). You should be aware, however, that a lot of SVGA cards don't provide separate read and write banks, which means that blitting from one part of the screen to another requires the use of a temporary bitmap in memory, and is therefore extremely slow. As a general rule you should avoid blitting from the screen onto itself in SVGA modes.

In mode-X, on the other hand, blitting from one part of the screen to another can be significantly faster than blitting from memory onto the screen, as long as the source and destination are correctly aligned with each other. Copying between overlapping screen rectangles is slow, but if the areas don't overlap, and if they have the same plane alignment (ie. $(source_x \% 4) == (dest_x \% 4)$), the VGA latch registers can be used for a very fast data transfer. To take advantage of this, in mode-X it is often worth storing tile graphics in a hidden area of video memory (using a large virtual screen), and blitting them from there onto the visible part of the screen.

If the `GFX_HW_VRAM_BLIT` bit in the `gfx_capabilities` flag is set, the current driver supports hardware accelerated blits from one part of the screen onto another. This is extremely fast, so when this flag is set it may be worth storing some of your more frequently used graphics in an offscreen portion of the video memory.

Unlike most of the graphics routines, `blit()` allows the source and destination bitmaps to be of different color depths, so it can be used to convert images from one pixel format to another. In this case, the behavior is affected by the `COLORCONV_KEEP_TRANS` and `COLORCONV_DITHER*` flags of the current color conversion mode: see `set_color_conversion()` for more information.

SEE ALSO

masked_blit(3), stretch_blit(3), draw_sprite(3), gfx_capabilities(3), set_color_conversion(3)

