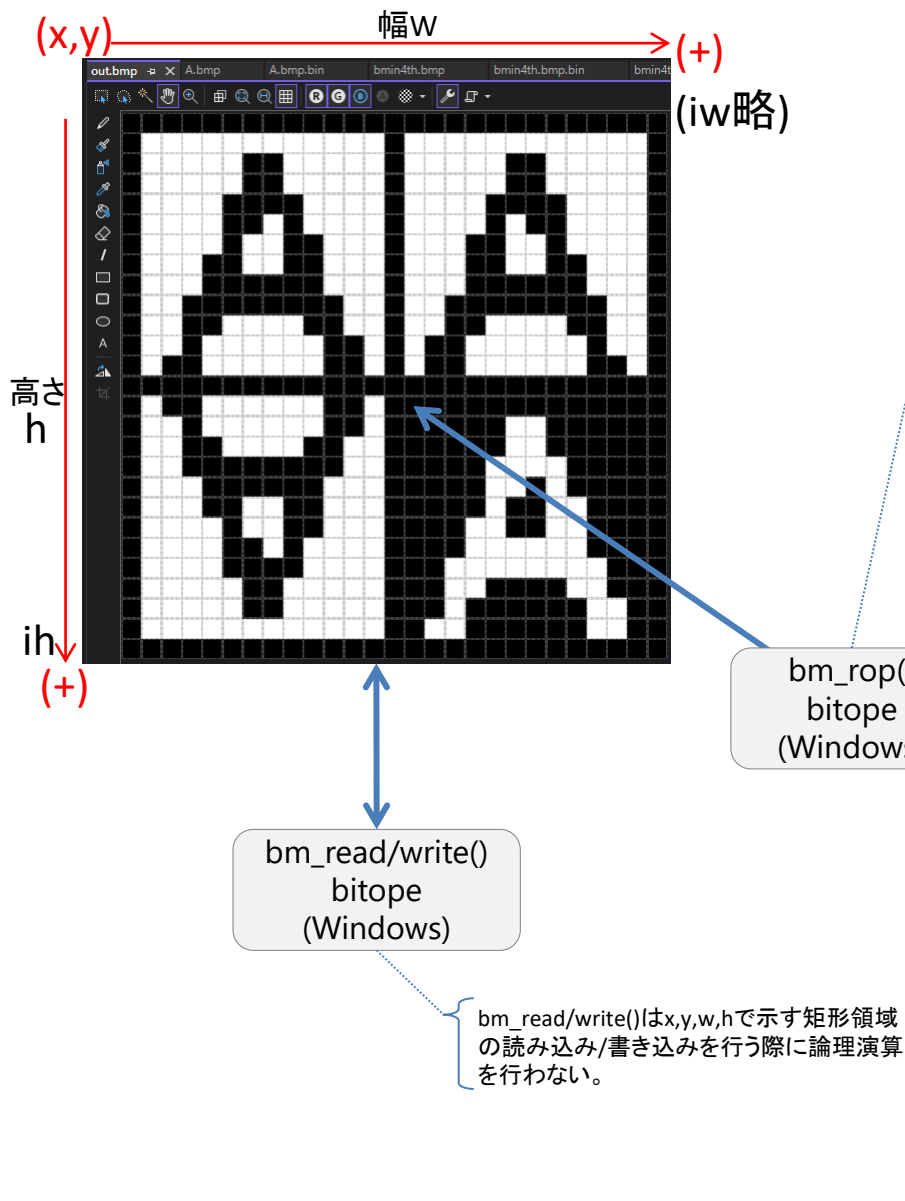




モジュール設計仕様書	システム	bitope	Bit-mapオペレーション	作成日	2025/02/04
補足	モジュール	ex_bitope	Bit-mapを操作する	作成者	HosiYamaKaze

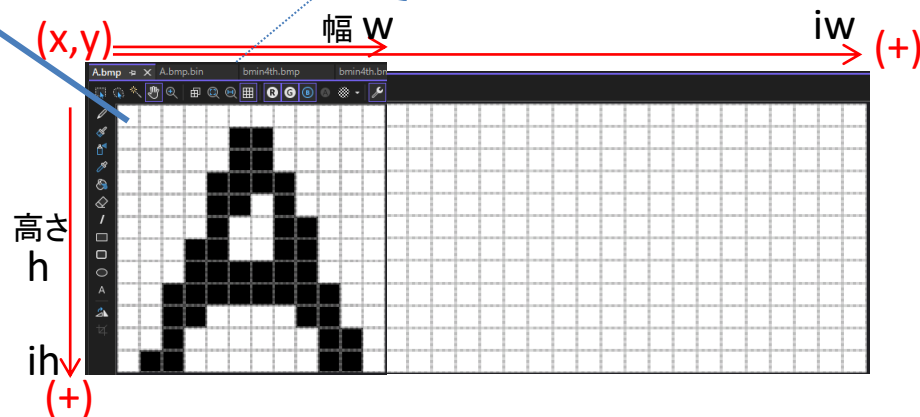
DFD:ex\_bitope.cpp (2) Bit-map操作、bm\_{rop,read,write}()



1. bm\_rop()は、複写先(dst)のx,y,w,hで示す矩形領域に、複写元(src)のx,y,w,hで示す矩形領域の対応するbit間で論理演算を行い、その結果を複写先(dst)にセットする。
2. x,yは左上隅を原点(0,0)とする(第四象限的)。
3. 矩形領域の大きさ(w,h)は一致する必要がある。
4. 複写元の定数として、BM\_SRC0(全ビット0:白)、BM\_SRC1(全ビット1:黒)を使用できる。

```
BM_0,/* 0 */
BM_1,/* 1 */
BM_and,/* dst & src */
BM_nand,/* ~dst & src */
BM_andN,/* dst & ~src */
BM_nandN,/* ~dst & ~src */
BM_or,/* dst | src */
BM_nor,/* ~dst | src */
BM_orN,/* dst | ~src */
BM_norN,/* ~dst | ~src */
BM_xor,/* dst xor src */
BM_xorN,/* dst xor ~src */
BM_noop,/* dst */
BM_N,/* ~dst */
BM_set,/* src */
BM_Nset,/* ~src */
```

1. 処理範囲をx,y,w,hの矩形領域で示すため、これらの値は必要により処理中に変化する。
2. ih,iwは、矩形領域の最大値である。
3. iwは32bit単位のパディング境界までを含む。



モジュール設計仕様書	システム	bitope	Bit-mapオペレーション	作成日	2025/02/05
補足	モジュール	ex_base64	Bit-mapを使ってbit列を操作する	作成者	HosiYamaKaze

DFD:全体像(ex\_base64.cpp:高さh=1のbit-mapとして、bit列を操作する)

Enter source text (for Enter to Base64): 0123abcABC  
Base64 Encode: MDEyM2FiY0FCQw==  
Base64 Decode: 0123abcABC  
Enter source text (for Enter to Base64):  
Enter Base64 text (for Enter to exit): MDEyM2FiY0FCQw==  
Base64 Decode: 0123abcABC  
Base64 Encode: MDEyM2FiY0FCQw==  
Enter source text (for Enter to Base64):  
Enter Base64 text (for Enter to exit):  
Done.

1. 入力されたtextのbit列を高さh:1のビットマップを利用して、幅w:6bitまたは8bitで取り扱う。
2. Bit操作例を示すため、下記bitmapイメージ化した。

