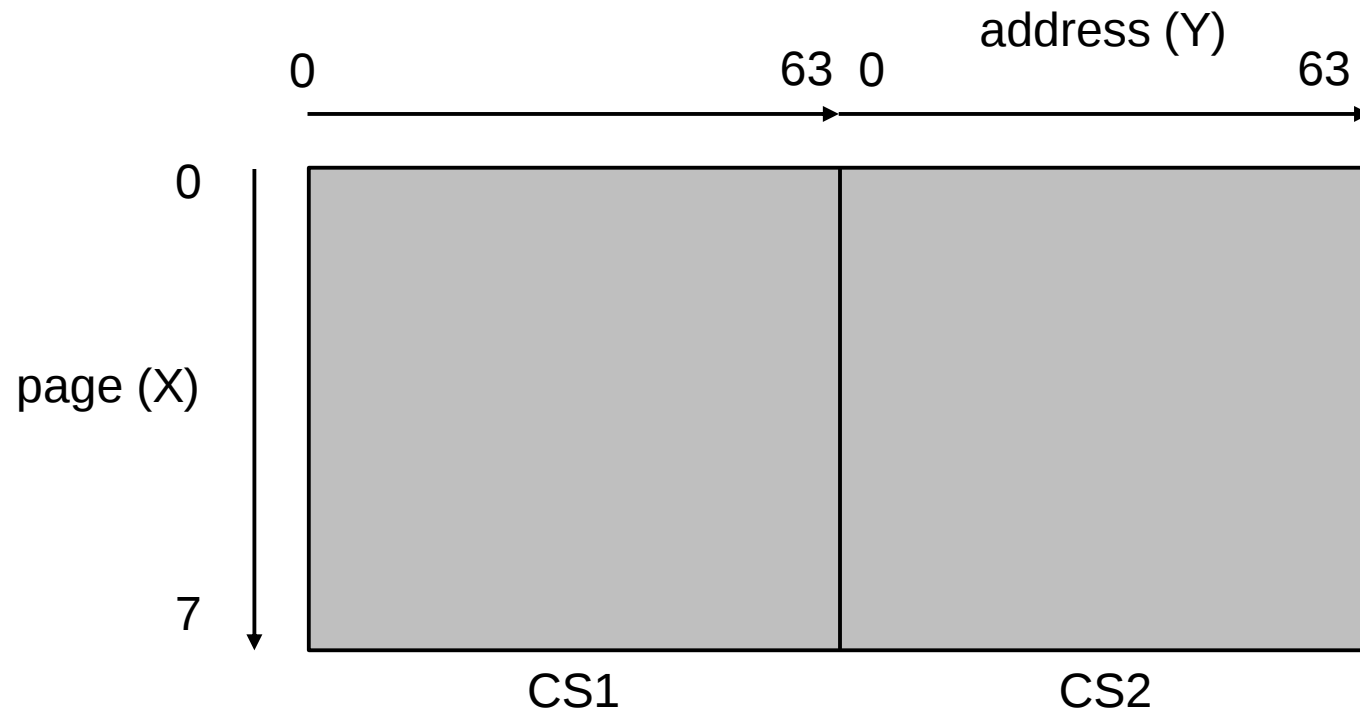


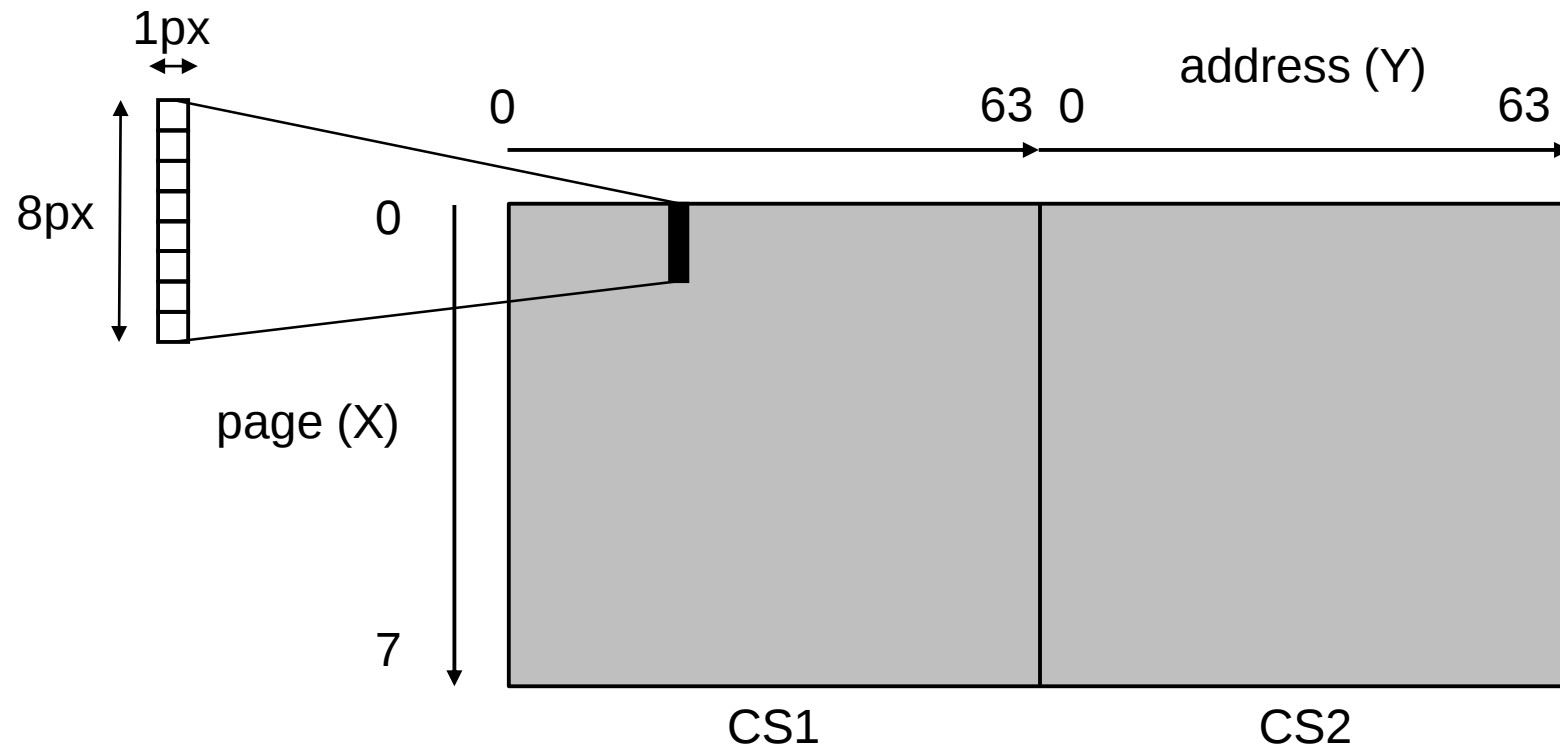
Egenskaper

- Vi kan bara läsa och skriva i hela alignade bytes i vertikalled!
- Exempel:



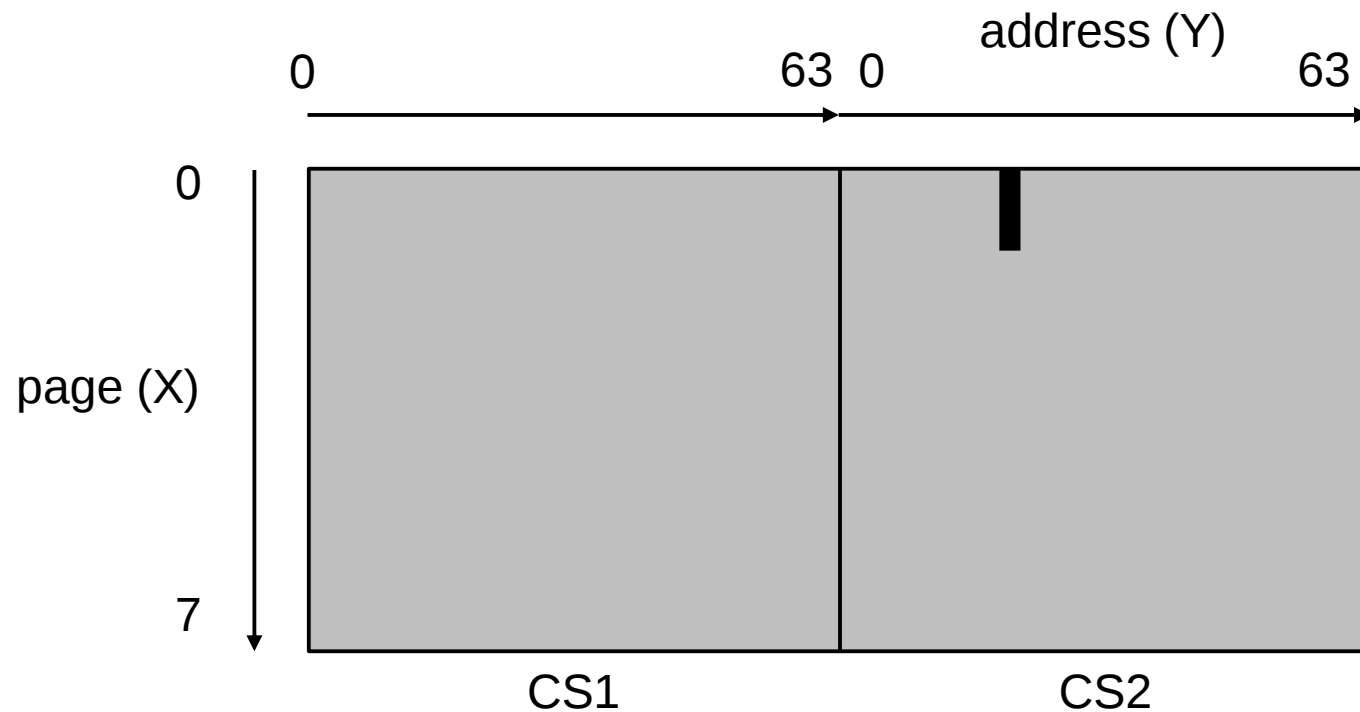
Egenskaper

- Vi kan bara läsa och skriva i hela alignade bytes i vertikalled!
- Exempel: CS1 = 1, CS2 = 0, page = 0, address = 20.



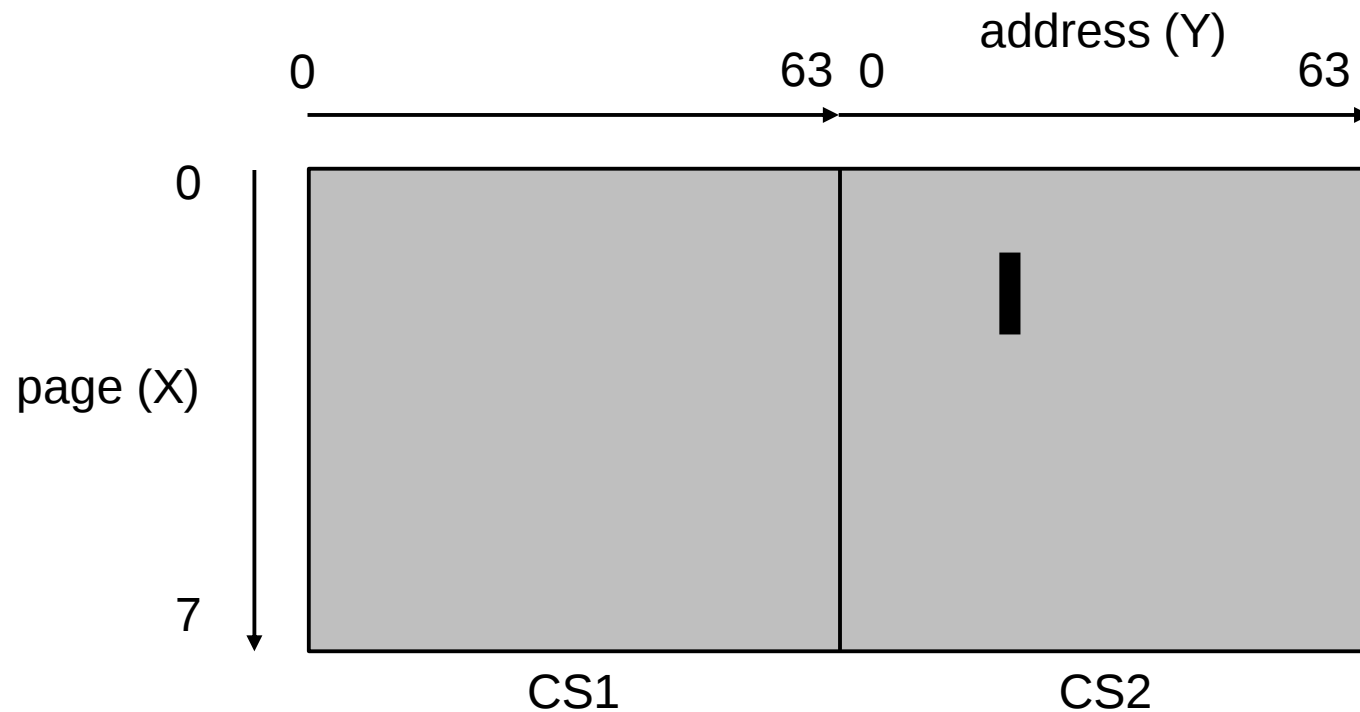
Egenskaper

- Vi kan bara läsa och skriva i hela alignade bytes i vertikalled!
- Exempel: CS1 = 0, CS2 = 1, page = 0, address = 20.



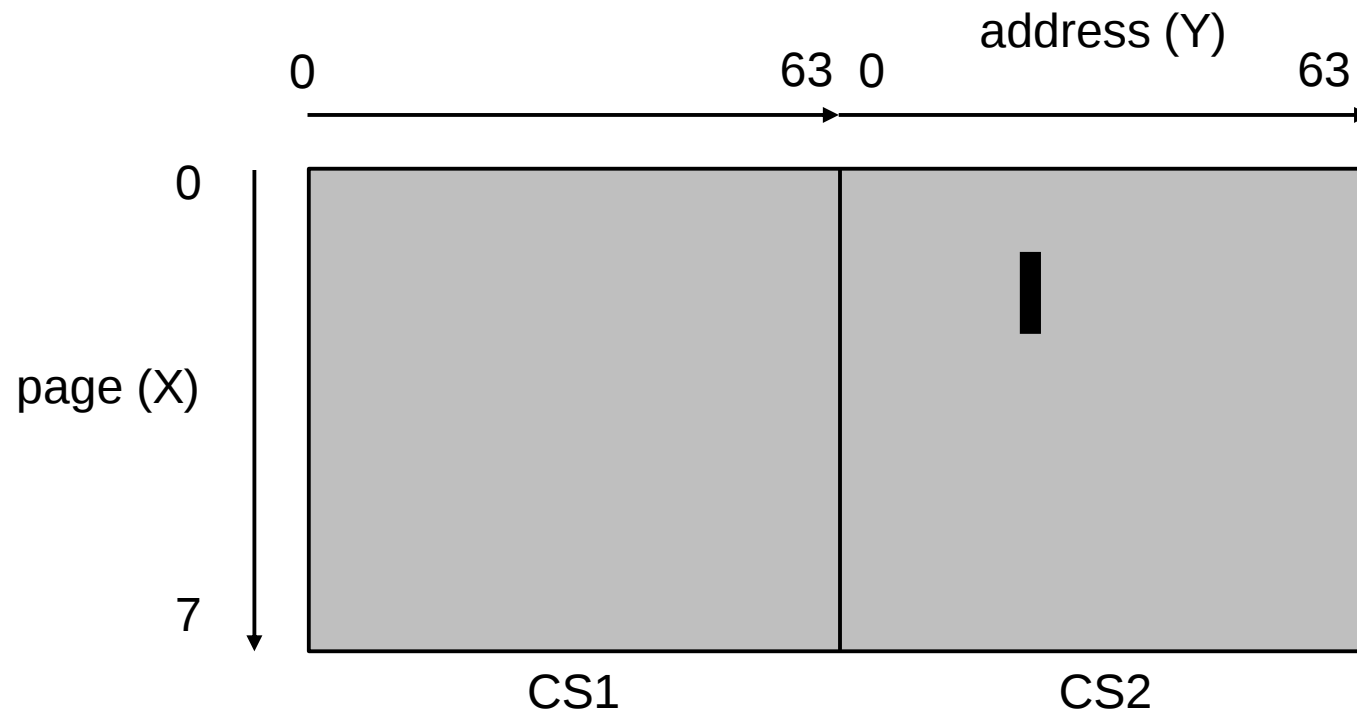
Egenskaper

- Vi kan bara läsa och skriva i hela alignade bytes i vertikalled!
- Exempel: CS1 = 0, CS2 = 1, page = 1, address = 20.



Egenskaper

- Vi kan bara läsa och skriva i hela alignade bytes i vertikalled!
- Exempel: CS1 = 0, CS2 = 1, page = 1, address = 21.



Implementation av drivrutiner

```
static void graphic_ctrl_bit_set(uint8_t x) {  
    uint8_t c;  
    c = *portOdrLow;  
    c &= ~B_SELECT;  
    c |= (~B_SELECT & x);  
    *portOdrLow = c;  
}
```

```
static void graphic_ctrl_bit_clear(uint8_t x) {  
    uint8_t c;  
    c = *portOdrLow;  
    c &= ~B_SELECT;  
    c &= ~x;  
    *portOdrLow = c;  
}
```

Implementation av drivrutiner

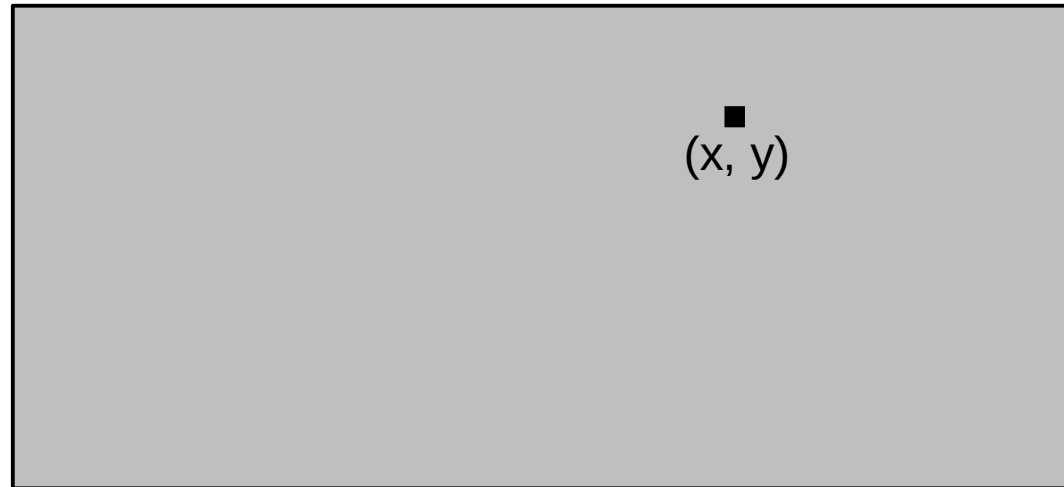
```
static void graphic_write(uint8_t value, uint8_t controller) {  
    *portOdrHigh = value;  
    select_controller(controller);  
    delay_500ns();  
    graphic_ctrl_bit_set(B_E);  
    delay_500ns();  
    graphic_ctrl_bit_clear( B_E );  
  
    if(controller & B_CS1) {  
        select_controller( B_CS1);  
        graphic_wait_ready();  
    }  
    if(controller & B_CS2) {  
        select_controller( B_CS2);  
        graphic_wait_ready();  
    }  
}
```

Skriva till display

```
void graphic_clear_screen(void) {  
    uint8_t i, j;  
  
    for(j = 0; j < 8; j++) {  
        graphic_write_command(LCD_SET_PAGE | j, B_CS1 | B_CS2);  
        graphic_write_command(LCD_SET_ADD | 0, B_CS1 | B_CS2);  
        for(i = 0; i <= 63; i++){  
            graphic_write_data(0, B_CS1 | B_CS2);  
        }  
    }  
}
```

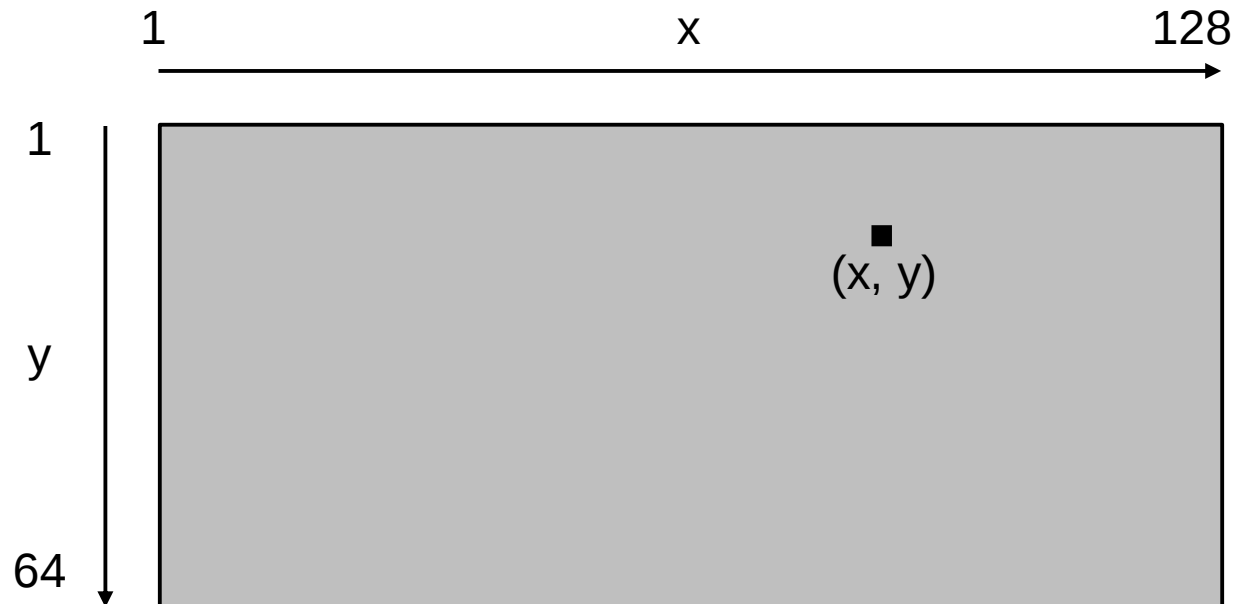
Skriva till display

- Funkar ok om vi vill rensa hela displayen, men inte så smidigt annars.
- Vi vill kunna skriva till displayen varje pixel för sig.



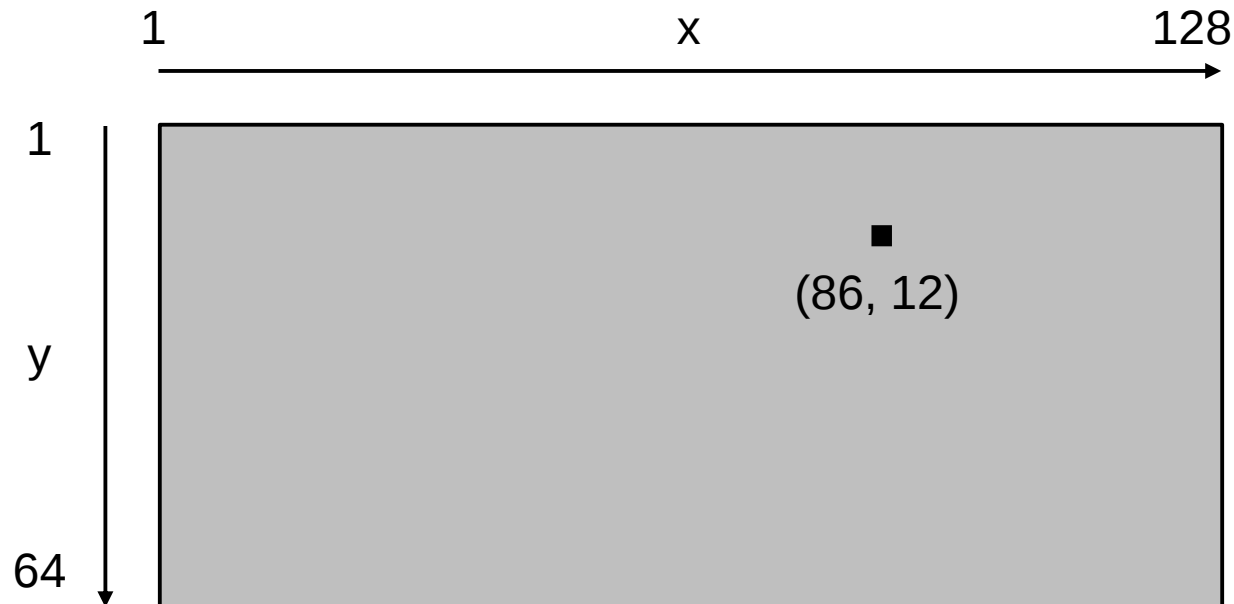
Skriva till display

- Vi vill ha en funktion `void pixel(int x, int y, int set)`
- Inför *logiska koordinater* med x i $[1..128]$ och y i $[1..64]$



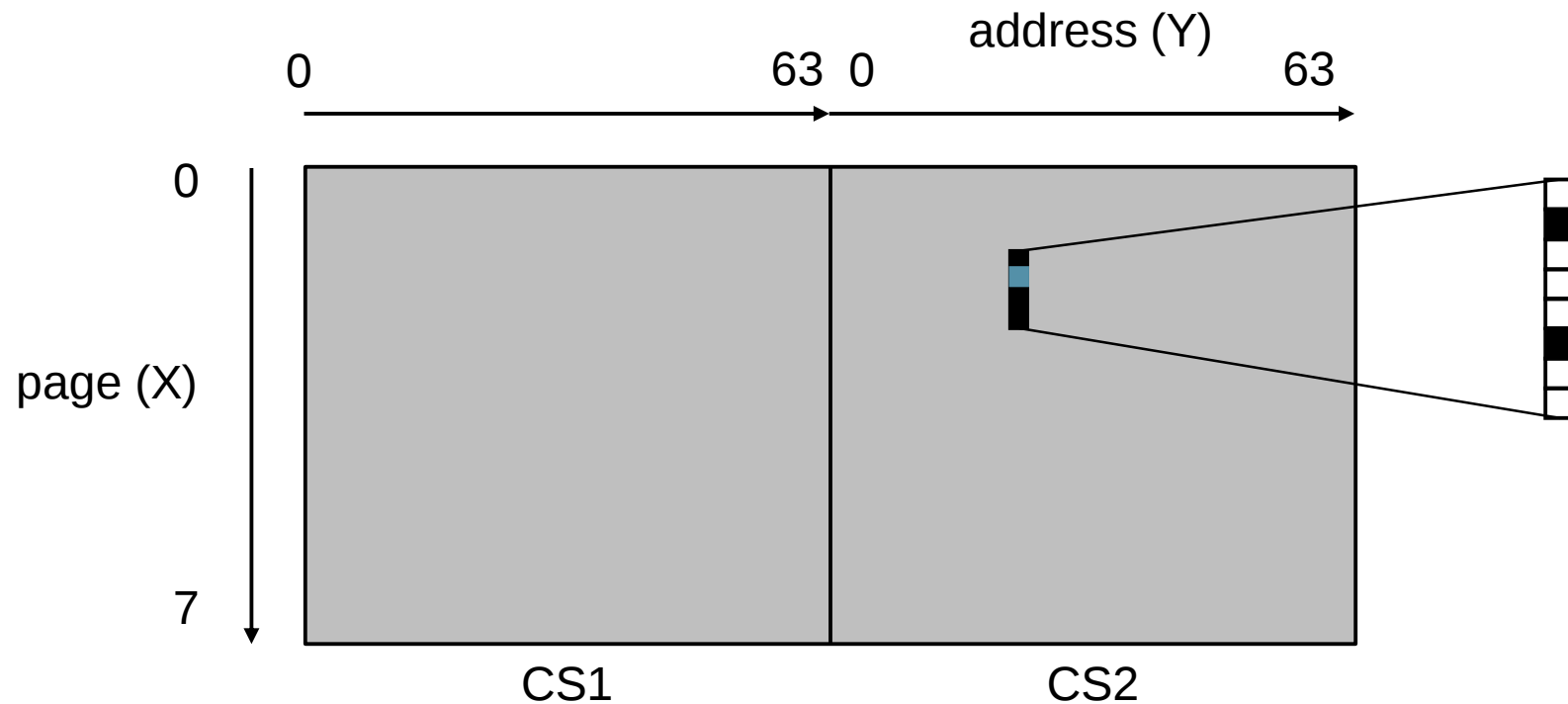
Skriva till display

- Hur skriver vi en sådan funktion när vi bara kan läsa/skriva i hela bytes?
- Exempel: Vi vill skriva till logiska koordinater (86, 12).



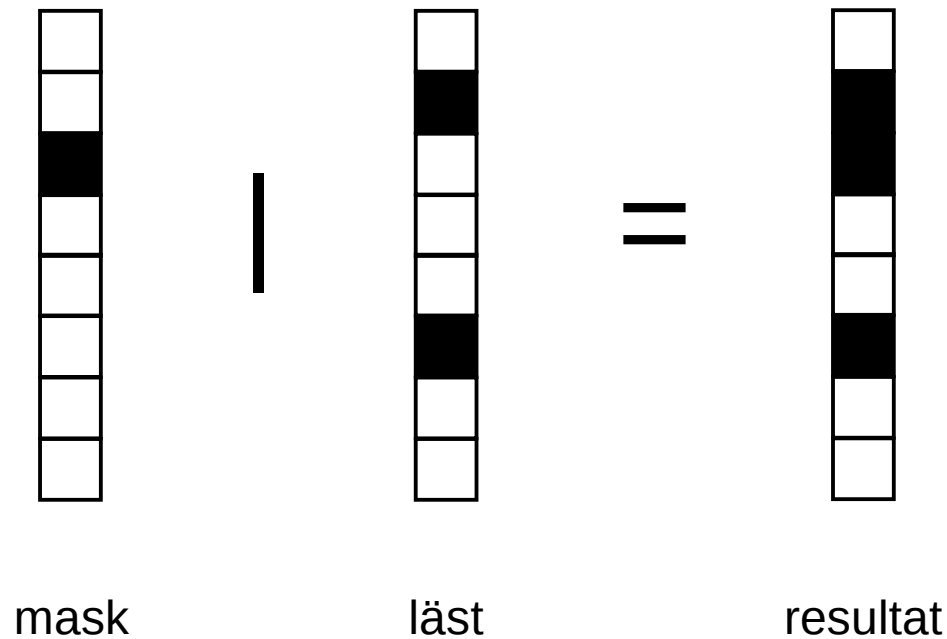
Egenskaper

- Vår address blir då $86 - 65 = 21$. Page blir $(12-1) / 8 = 1$.
- Vår bit i vår vertikala byte blir $(12-1) \% 8 = 3$.



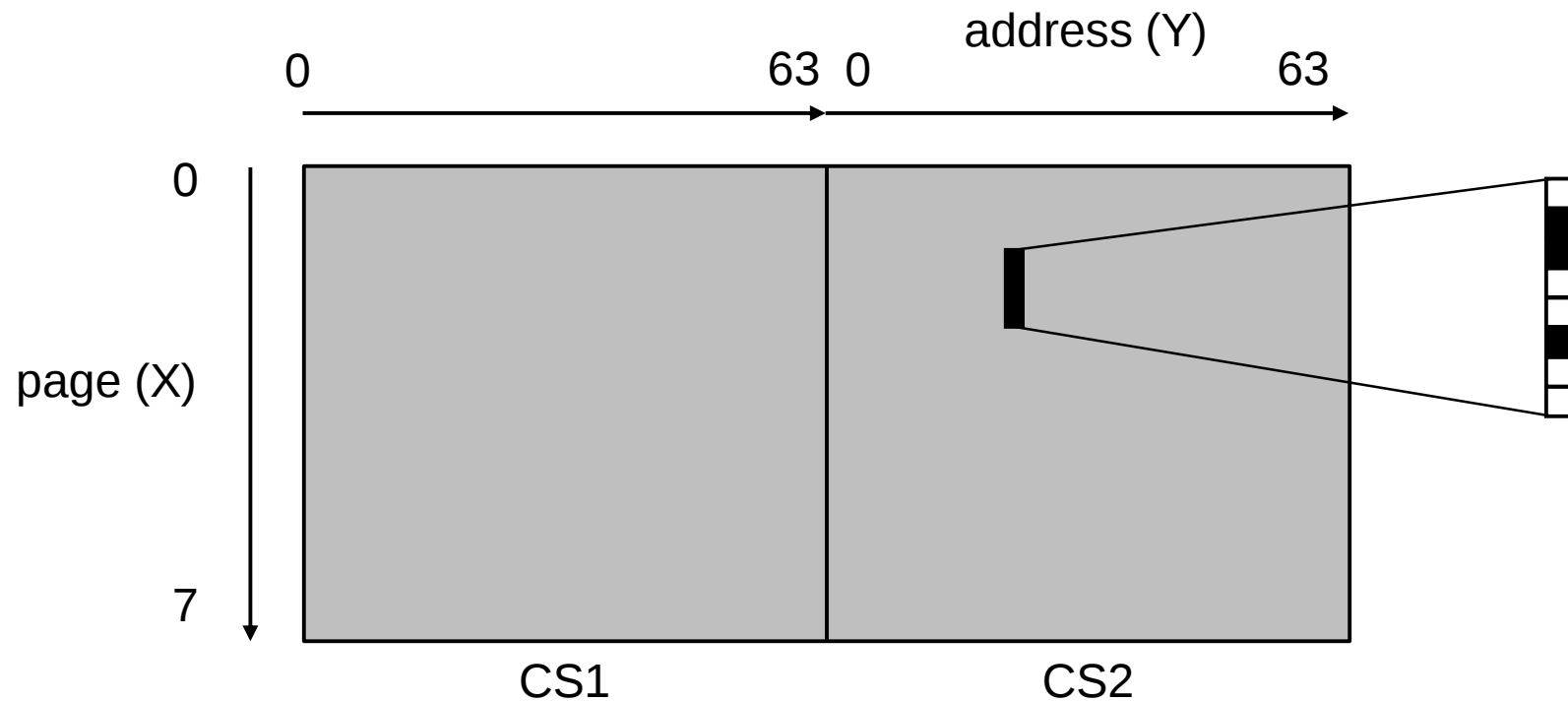
Egenskaper

- Vi läser då en byte med fysiska koordinater (CS2, 21, 1) .
- Vi vill skriva till bit 3 i denna byte.



Egenskaper

- Skriv denna resulterande byte tillbaka till displayen.



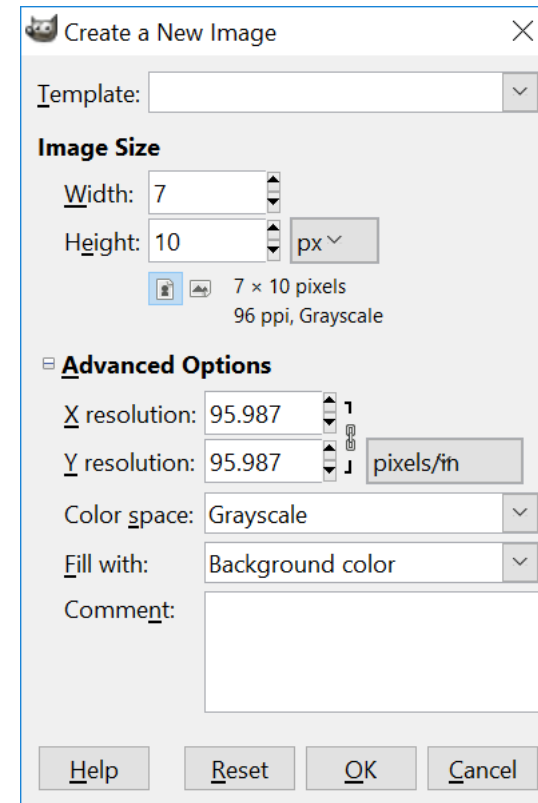
Design av grafik

Designa egen sprite

Gå till File->New...

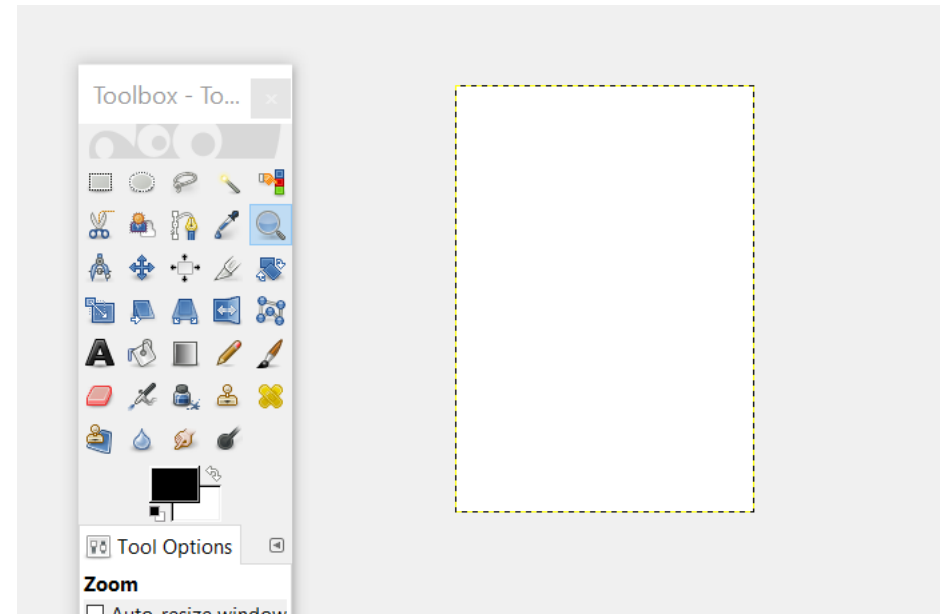
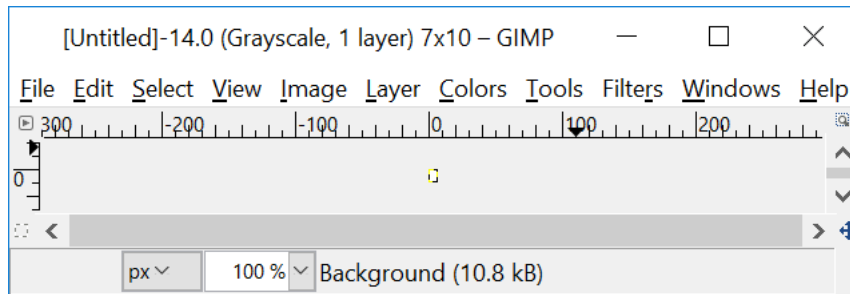
Sätt bredd och höjd i pixlar

Välj Color space: Grayscale



Design av grafik

Bilden är extremt liten på vanlig bildskärm.
Zooma in.

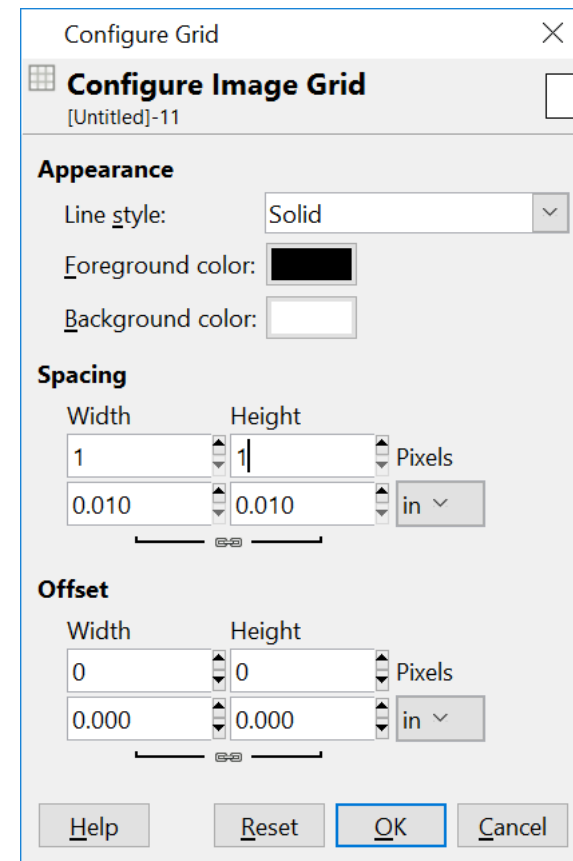


Design av grafik

1x1 pixels rutnät hjälper till vid ritning.

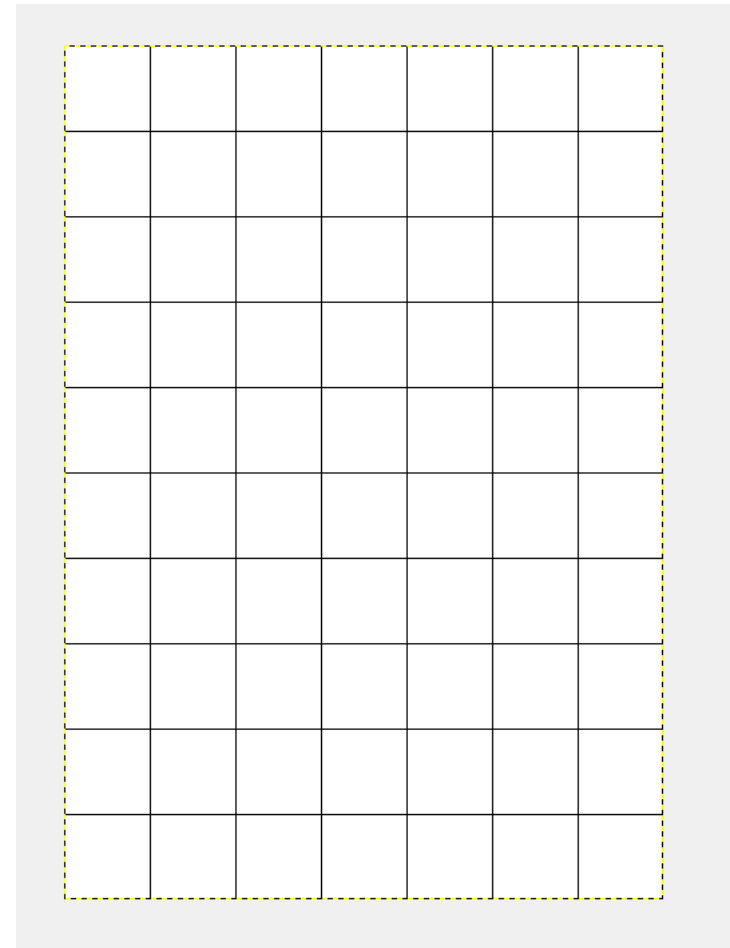
Välj Image->Configure Grid...

Välj Width = Height = 1 px



Design av grafik

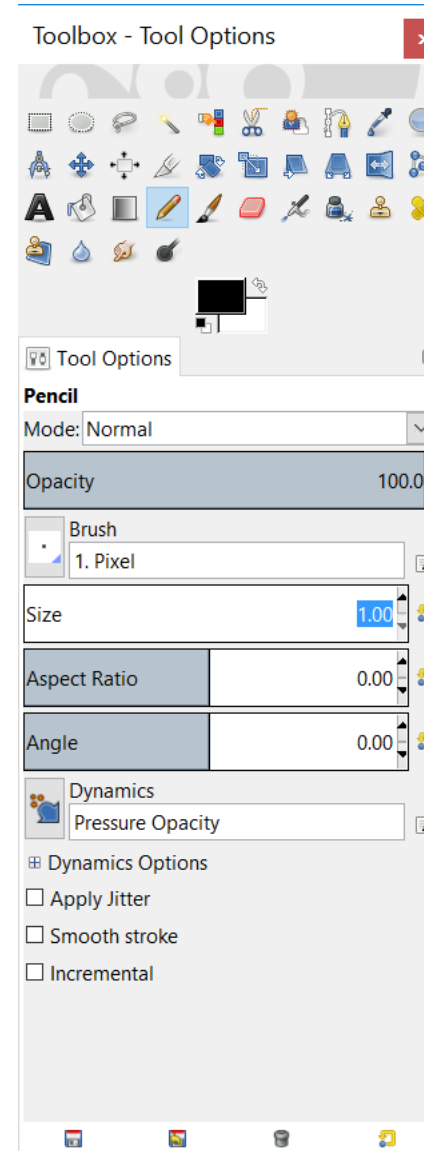
Togglar View->Show Grid



Design av grafik

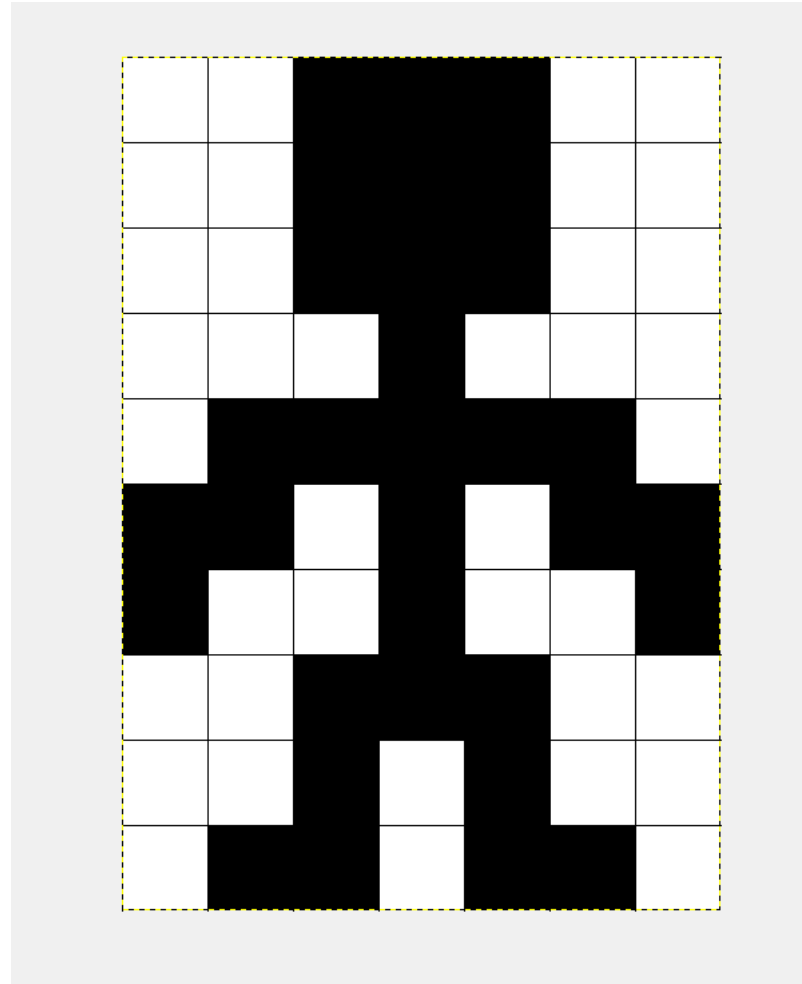
Använd "Pencil Tool"

Sätt Size = 1 px



Design av grafik

Rita!



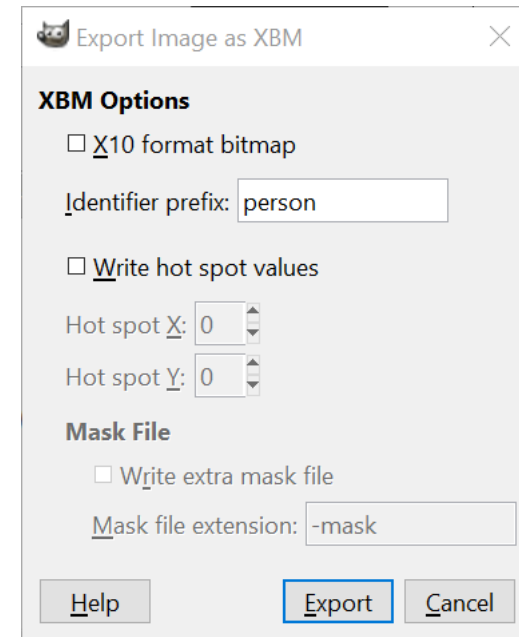
Design av grafik

Spara sedan som X Bitmap-fil.

Välj File->Export as...

Spara med ändelsen .xbm

Klicka på Export i dialogruten



Design av grafik

Resultat:

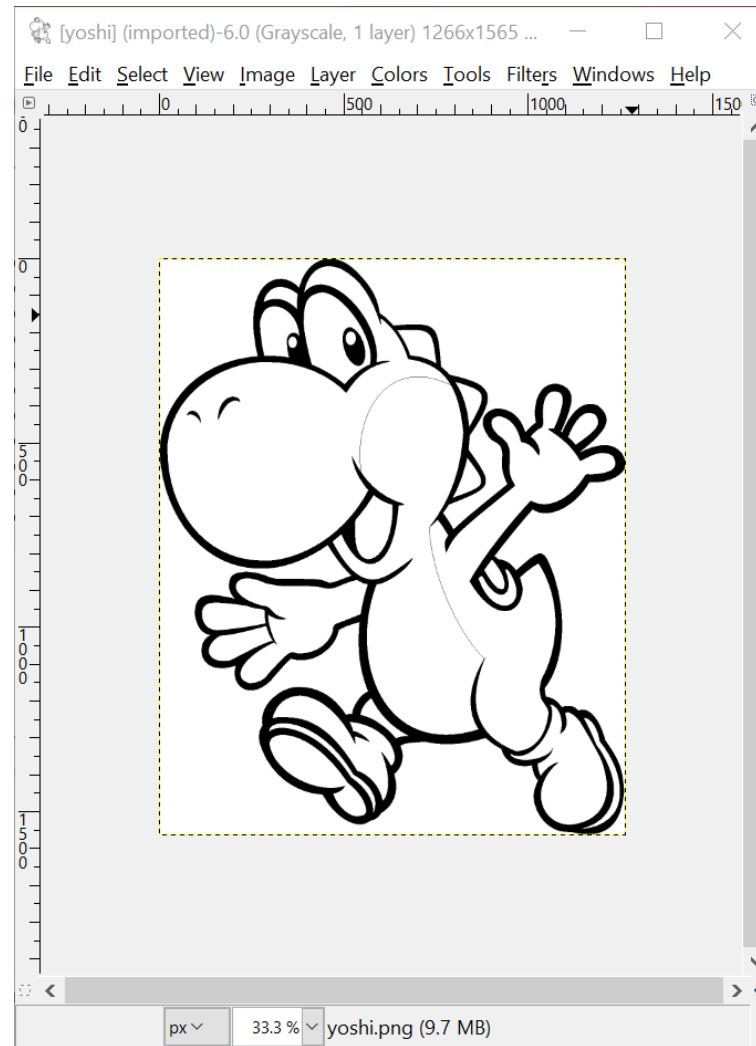
```
person.xbm X
1  #define person_width 7
2  #define person_height 10
3  static unsigned char person_bits[] = {
4      0x1c, 0x1c, 0x1c, 0x08, 0x3e, 0x6b, 0x49, 0x1c, 0x14, 0x36 };
5
```

Design av grafik

Vill designa en större eller mer komplicerad sprite.

Hitta en bild du gillar på internet och sampla ner färgrymd och upplösning!

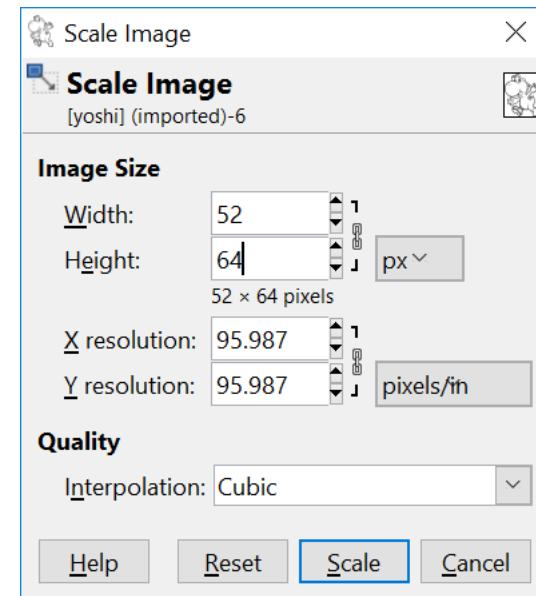
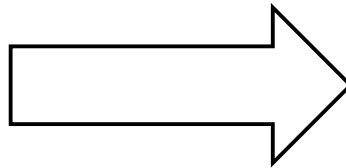
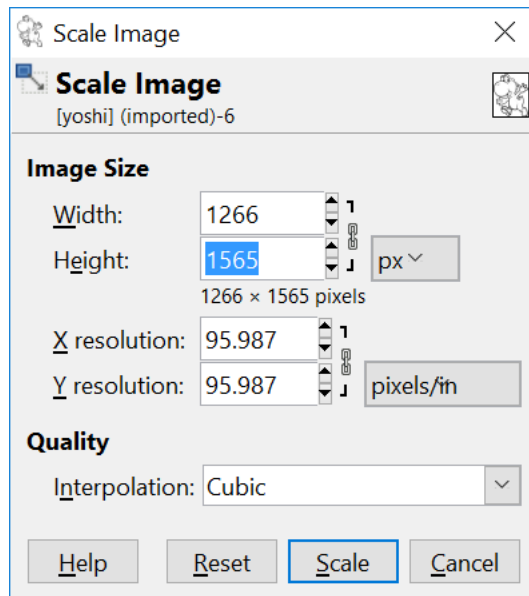
Exempel: Yoshi.



Design av grafik

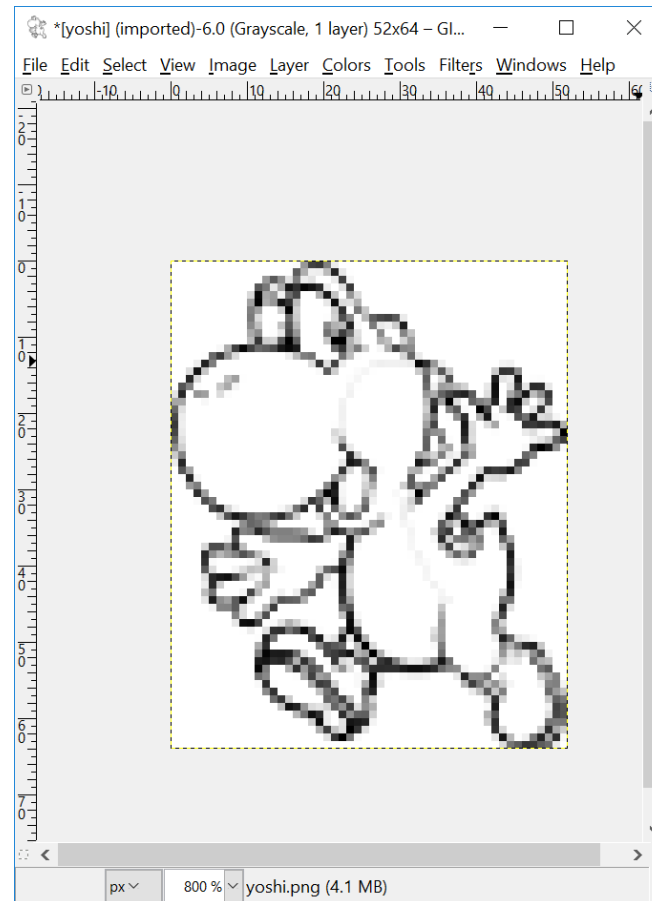
Välj Image->Scale Image...

Skala om till önskad storlek.



Design av grafik

Resultat
(efter rejäl inzoomning)

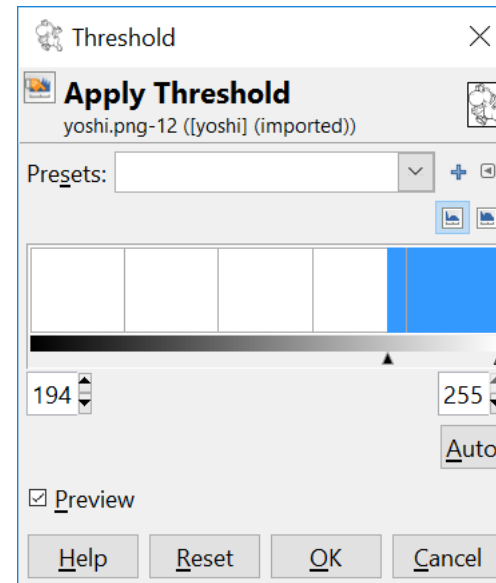


Design av grafik

Trunkera till 1-bits färg.

Välj Colors->Threshold...

Dra i slidern tills det ser bra ut.



Design av grafik

Nu är sprite:n färdig att använda i programmet.

Notera: utgick från en svartvit bild. Kan även ta färgbild och konvertera till svartvit, men kan vara svårt att få bra resultat.

