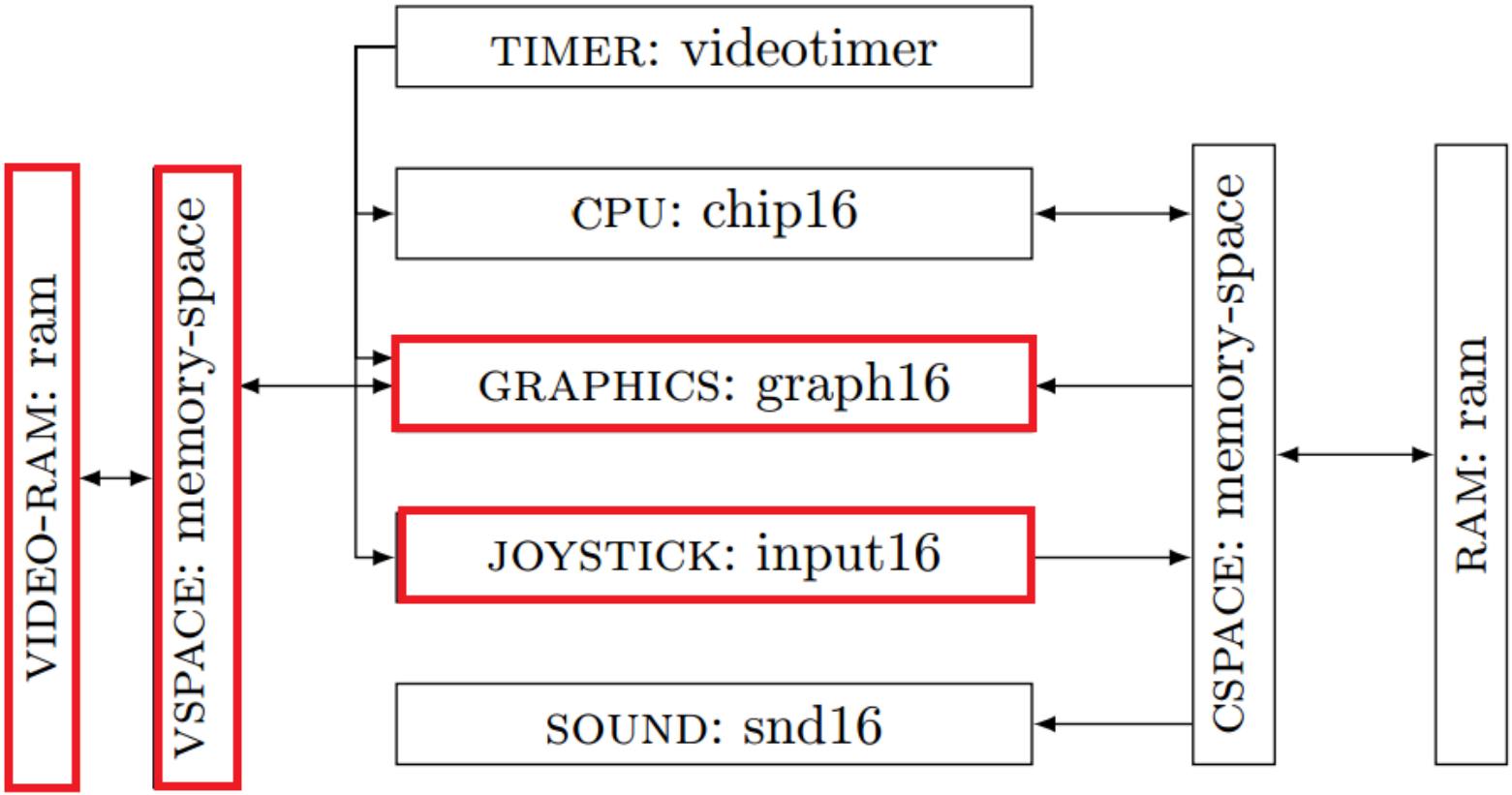
Graphics and Input

Чуйкин Олег

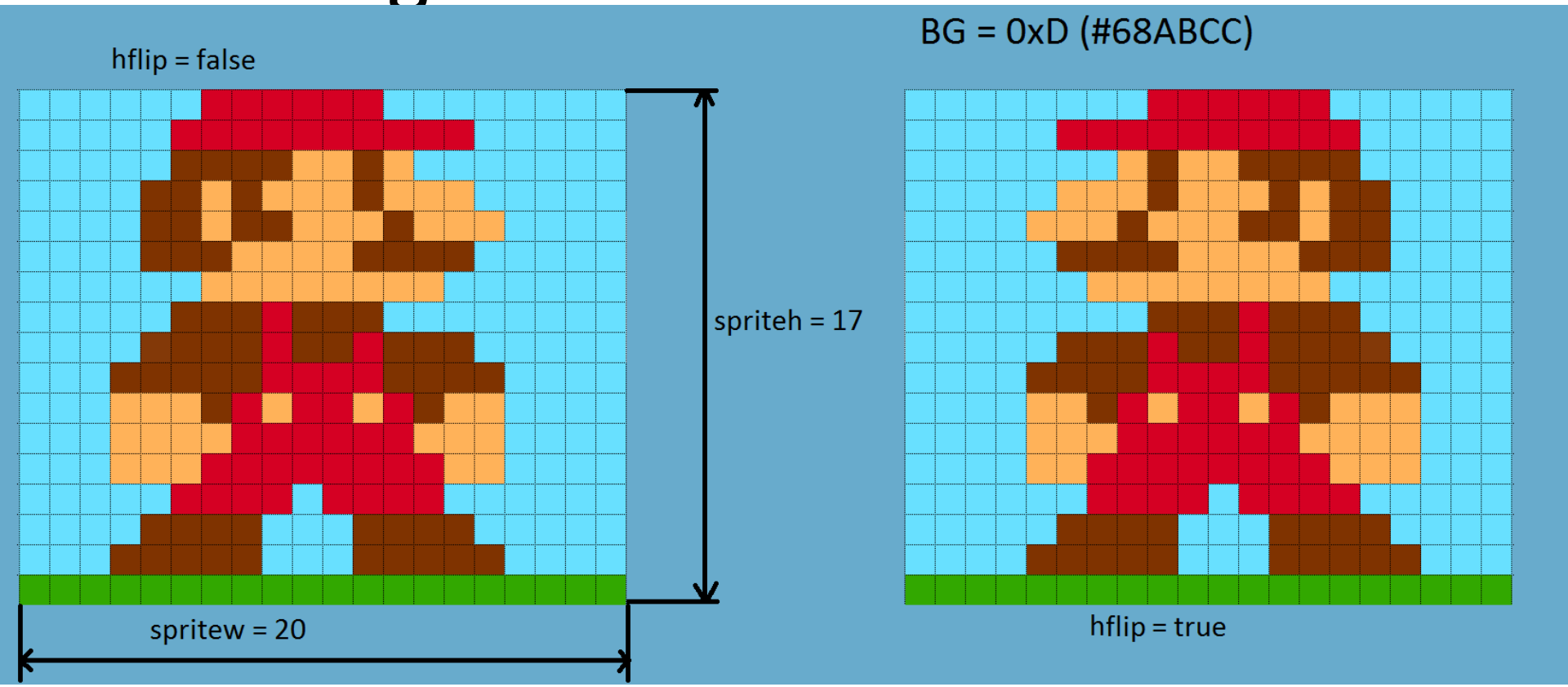
Лаборатория МФТИ-Интел

Chip16


TIMER: videotimer
CPU: chip16
GRAPHICS: graph16
JOYSTICK: input16
SOUND: snd16
VIDEO-RAM: ram
VSPACE: memory-space
CSPACE: memory-space
RAM: ram

1.Graphics

State registers



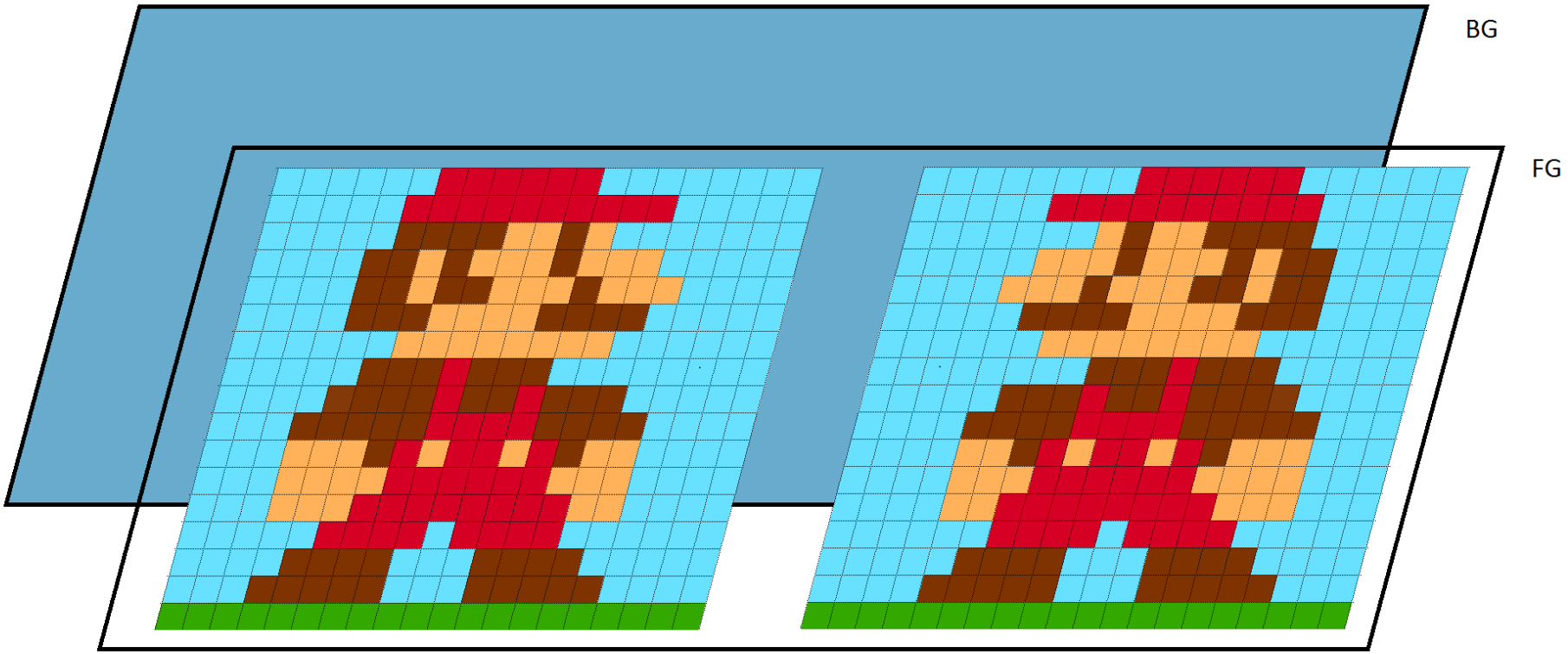
Register	Type
bg	Nibble
spritew	Unsigned byte

spriteh	Unsigned byte
hflip	Boolean
vflip	Boolean

Palette

Index	Value	Color
0x0	#000000	Black (transp. inFG)
0x1	#000000	Black
0x2	#888888	Gray

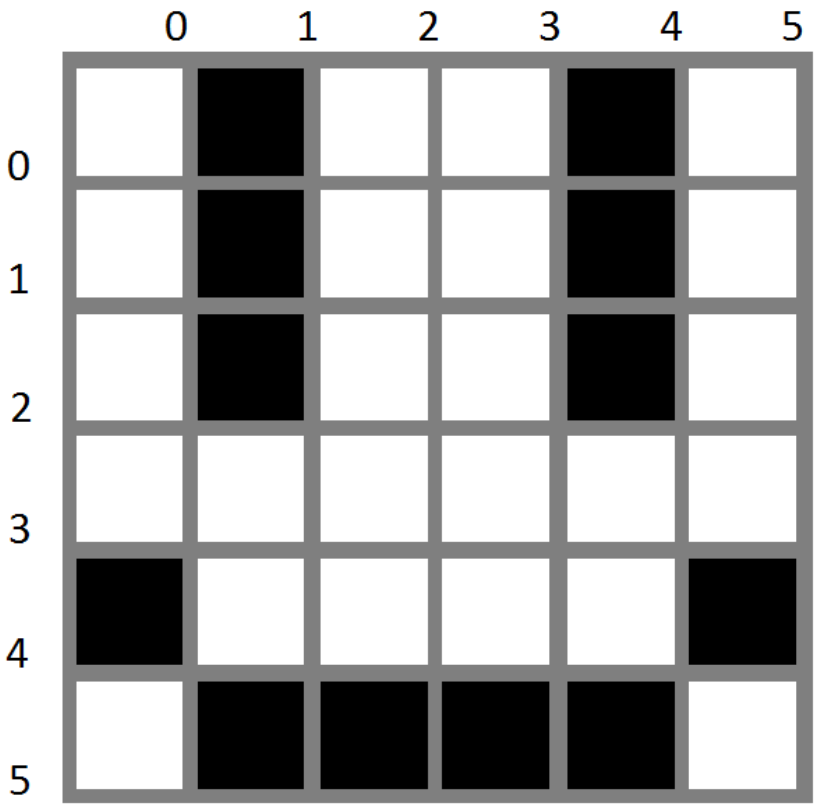
Display format



Screen details

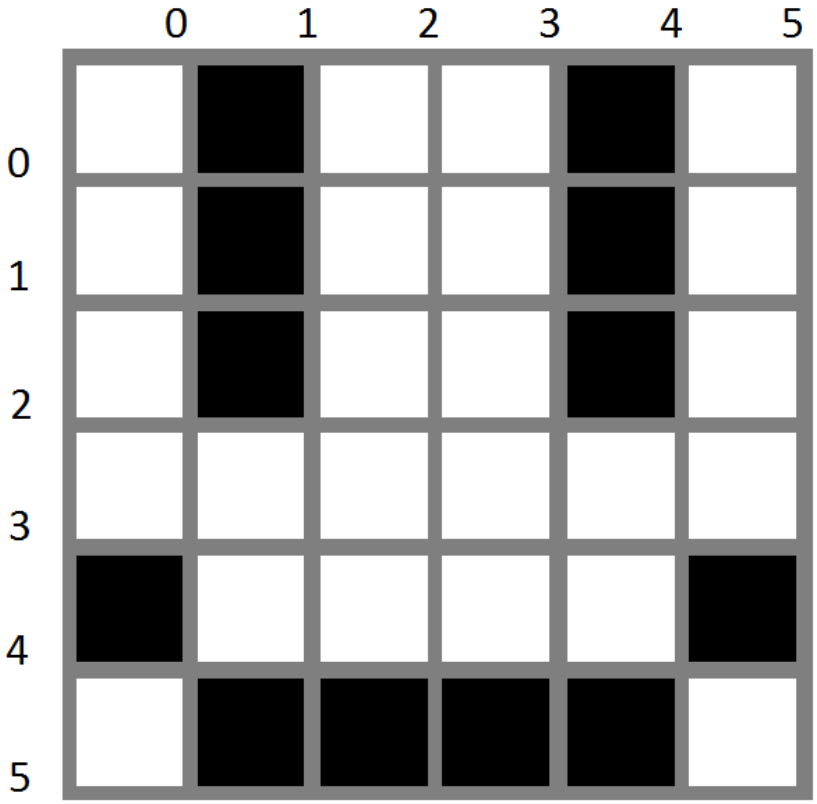
- 320x240 screen resolution.
- Refresh frequency of 60 Hz(~16.67ms).

Example



0xF – White
0x1 - Black

Example

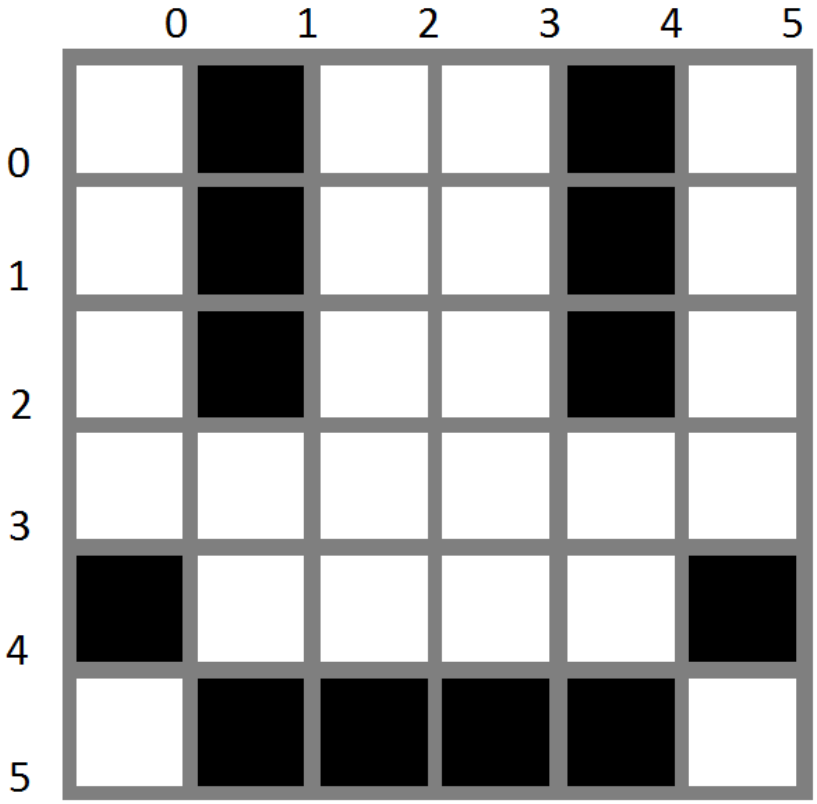


F 0 F F 0 F
F 0 F F 0 F
F 0 F F 0 F
F F F F F F
0 F F F F 0
F 0 0 0 0 F

→ 36 / 2 = 18 bytes

0xF – White
0x1 - Black

Example



36 / 2 = 18 bytes

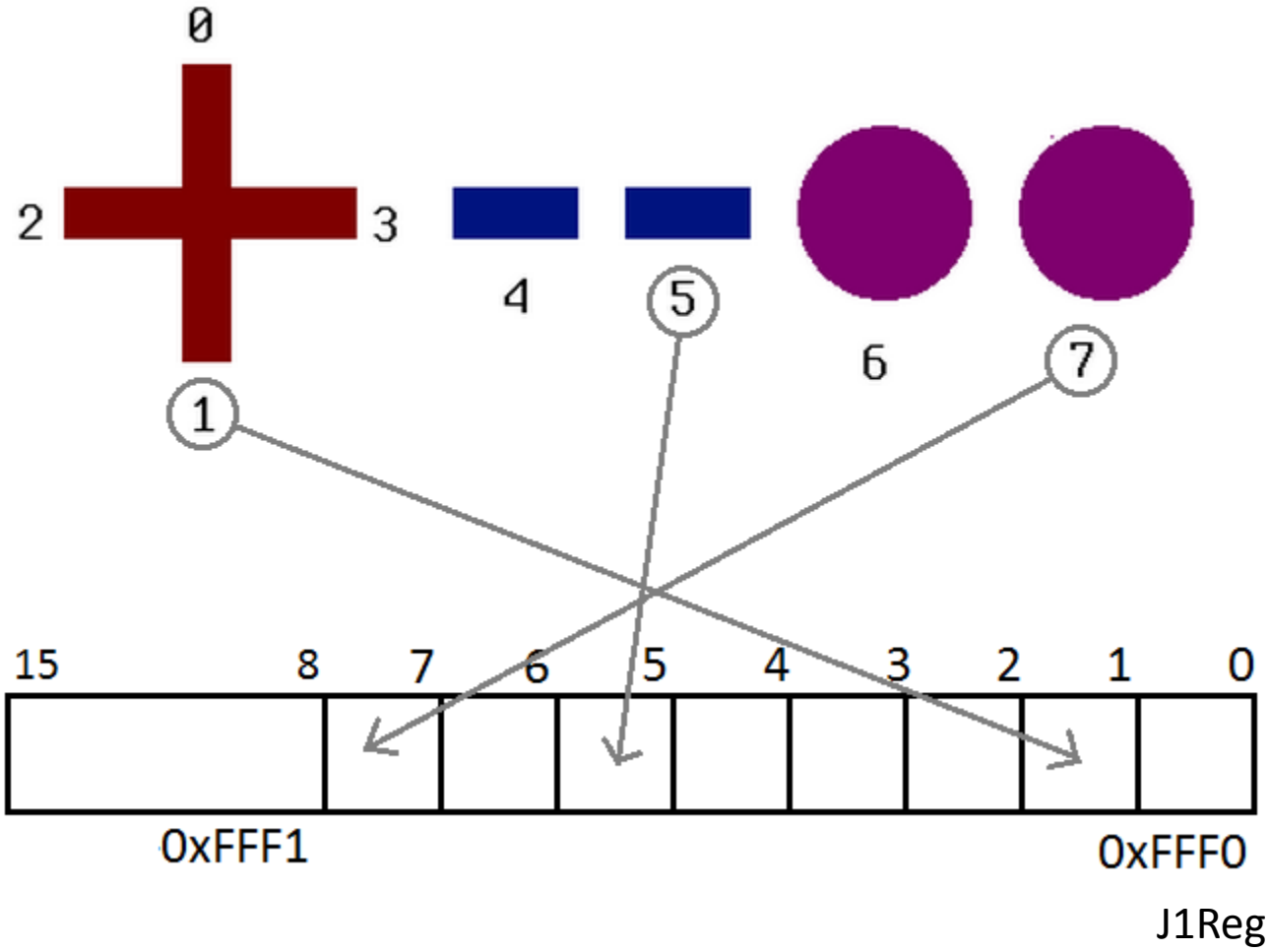
01 00 00 00 (CLS) – clear display
04 00 06 06 (SPR 06 06) – set H and L of sprite
08 00 00 00 (FLIP 0 0) – no flip
05 YX LL HH (DRW RX RY HHLL) – draw sprite from HHLL with coordination RX RY

0xF – White
0x1 - Black

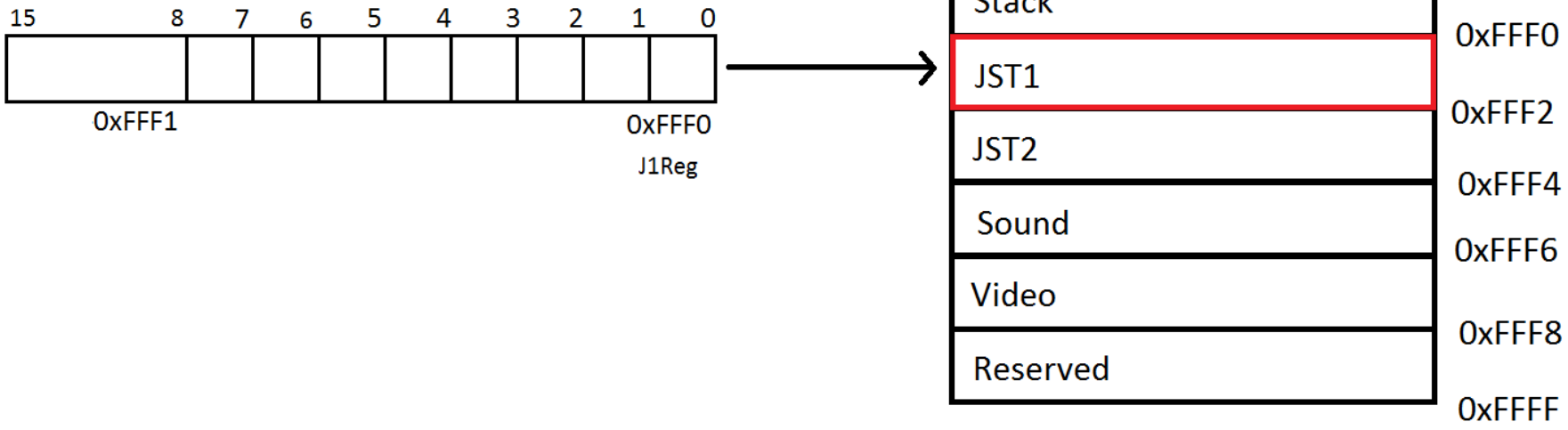
iLab

2.Input

Controller layout



Memory mapping



3. SDL2

Init

- **#include “path_to_sdl2/SDL.h”**

- **SDL_Init**(SDL_INIT_EVERYTHING)
- /*code*/
- **SDL_Quit**()

Create window

```
SDL_Window* SDL_CreateWindow(
    const char* title,
    int x,
    int y,
    int w,
    int h,
    Uint32 flags)
```

Events

SDL_Event event;

int **SDL_PollEvent**(SDL_Event* event)

Click handling

```
SDL_Event event;
Const Uint8 * kState = SDL_GetKeyboardState(NULL);

If (SDL_PollEvent(&event))
    switch (event.type){
        case event.key.keysym.sym:
            if (SDL_SCANCODE_UP)
                /*do smth*/
```

Scancodes

- http://wiki.libsdl.org/SDL_Scancode