

src/mmu.rs

```
pub fn new(data: Vec<u8>) → MemoryManagementUnit {  
    let mbc = MemoryBankController::new(data);  
  
    let mut res = MemoryManagementUnit {  
        wram: [0; WRAM_SIZE],  
        zram: [0; ZRAM_SIZE],  
        wrambank: 1,  
        inte: 0,  
        intf: 0,  
        input: Input::default(),  
        gpu: Gpu::new(),  
        mbc,  
    };  
}
```

src/mmu.rs

Map of the initial memory

```
res.write_byte(0xFF05, 0);
res.write_byte(0xFF06, 0);
res.write_byte(0xFF07, 0);
res.write_byte(0xFF10, 0x80);
res.write_byte(0xFF11, 0xBF);
res.write_byte(0xFF12, 0xF3);
res.write_byte(0xFF14, 0xBF);
res.write_byte(0xFF16, 0x3F);
res.write_byte(0xFF16, 0x3F);
res.write_byte(0xFF17, 0);
res.write_byte(0xFF19, 0xBF);
res.write_byte(0xFF1A, 0x7F);
res.write_byte(0xFF1B, 0xFF);
res.write_byte(0xFF1C, 0x9F);
res.write_byte(0xFF1E, 0xFF);
res.write_byte(0xFF20, 0xFF);
```

```
res.write_byte(0xFF21, 0);
res.write_byte(0xFF22, 0);
res.write_byte(0xFF23, 0xBF);
res.write_byte(0xFF24, 0x77);
res.write_byte(0xFF25, 0xF3);
res.write_byte(0xFF26, 0xF1);
res.write_byte(0xFF40, 0x91);
res.write_byte(0xFF42, 0);
res.write_byte(0xFF43, 0);
res.write_byte(0xFF45, 0);
res.write_byte(0xFF47, 0xFC);
res.write_byte(0xFF48, 0xFF);
res.write_byte(0xFF49, 0xFF);
res.write_byte(0xFF4A, 0);
res.write_byte(0xFF4B, 0);
```