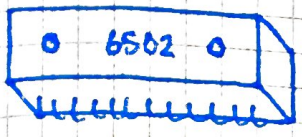


Nes Emulation: notes on the 6502 microprocessor

For more info go to wiki.nesdev.com

Bitwise Operators

& And allows you to apply a mask to a binary # like this

↳ 00011000 & 00001111

gives us 00001000

>> << shifting bits is sort of like dividing a byte by 2

↳ 00010000 >> 3

gives us 00000010

this allows us to use branching logic like so:

```
if (x&(1<<5)) ...
```

this tests the fifth bit

| the or operator allows us to set values:

```
y = 0b101
x |= (y<<2) // 00010100
```

~ allows us to clear bits:

```
x &= ~(1<<5)
```

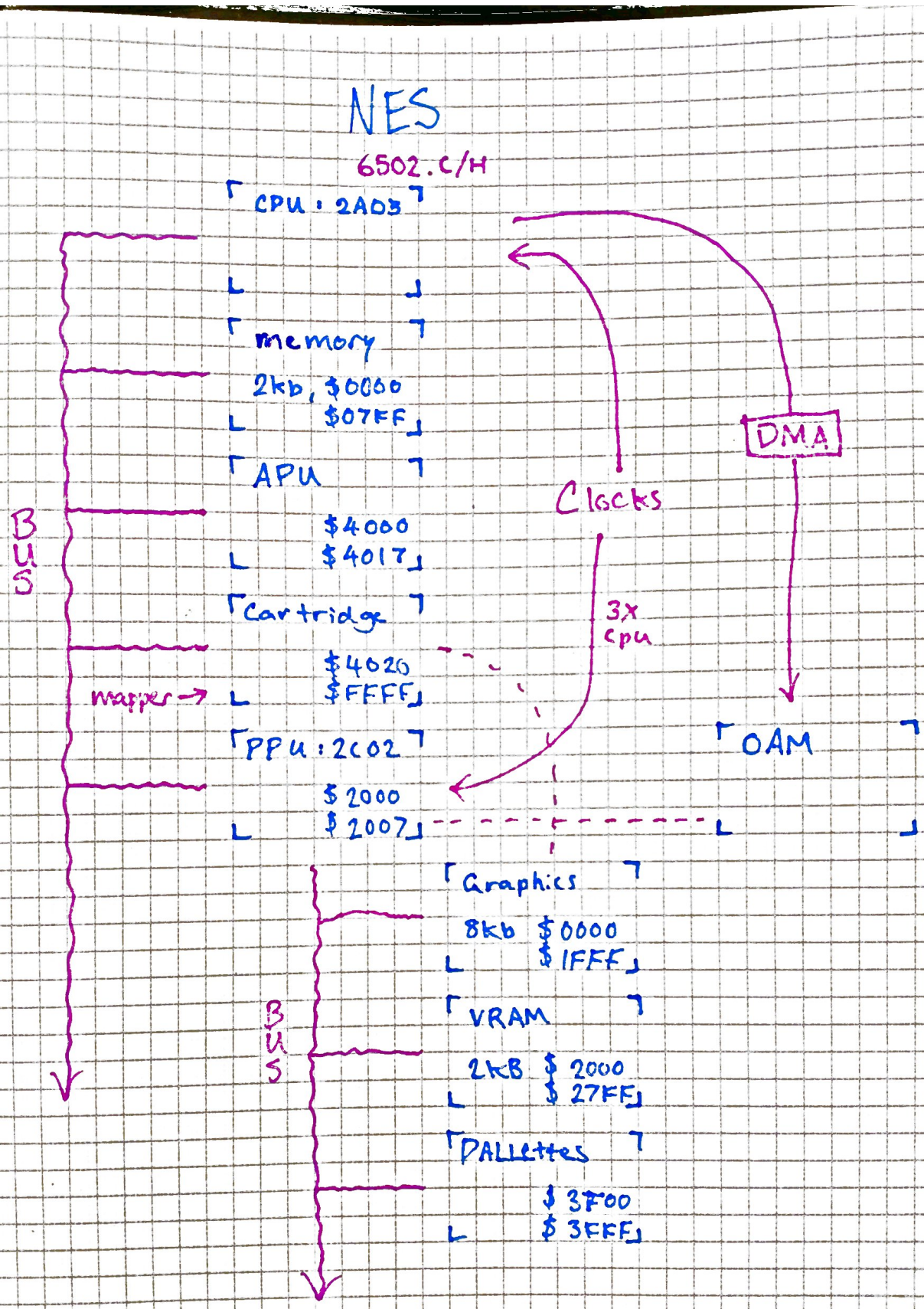
^ xor allows us to toggle bits on/off

A Bitfield will allow us to easily use bits in C/C++

```
{
  char sw1 : 1;
  char sw2 : 1;
  char val : 6;
}
```

```
A.sw1 = 1;
A.val = 25;
```

- We'll use these with unions


NES
6502.C/H
CPU : 2A03
memory
2kb, $0000
$07FF
APU
$4000
$4017
Cartridge
$4020
$FFFF
mapper
PPU : 2C02
$2000
$2007
BUS
Clocks
3x cpu
DMA
OAM
Graphics
8kb $0000
$1FFF
VRAM
2kB $2000
$27FF
Pallettes
$3F00
$3FFF
BUS