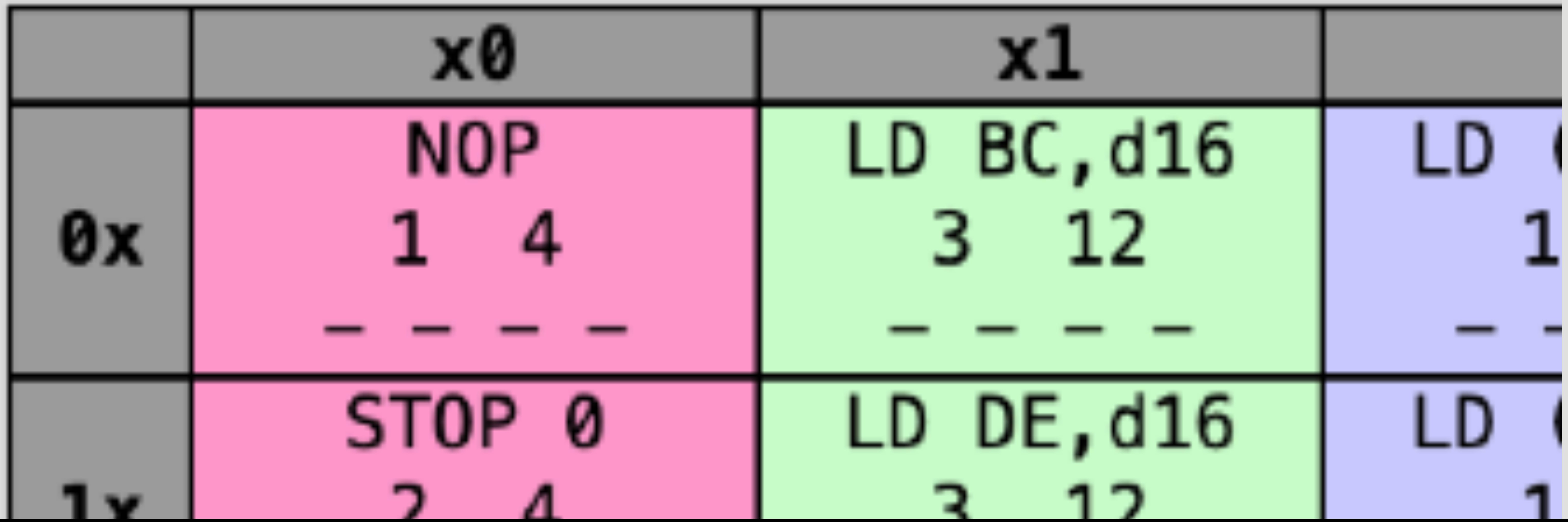

	x0	x1	
0x	NOP 1 4 - - - -	LD BC,d16 3 12 - - - -	LD (1 - -
1x	STOP 0 2 4	LD DE,d16 3 12	LD (1

```
0x00 => 1,
0x01 => {
    let value = self.memory.read_word(self.registers.pc);
    self.registers.pc = self.registers.pc.wrapping_add(2);
    cpu.registers.b = (value >> 8) as u8;
    cpu.registers.c = (value & 0x00FF) as u8;
    3
}
```

src/cpu/**mod.rs**

```rust
let byte = self.memory.read_byte(self.registers.pc);
self.registers.pc = self.registers.pc.wrapping_add(1);
let ticks = match byte {
    0x00 => 1,
    0x01 => { ld::bcnn(self); 3 }
    0x02 => { ld::bcm_a(self); 2 }
    0x03 => { data::incbc(self); 2 }

    // ... (rest of the instructions)

    0xFF => { stack::rst(self, 0x38); 4 }
    _ => { panic!("{:#06x} not implemented", op);
}
self.memory.cycle(ticks * 4);
```