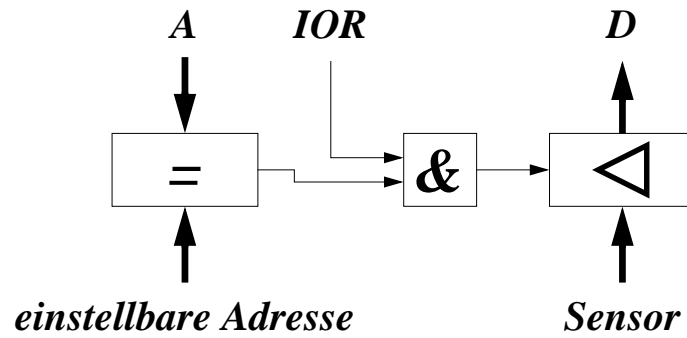


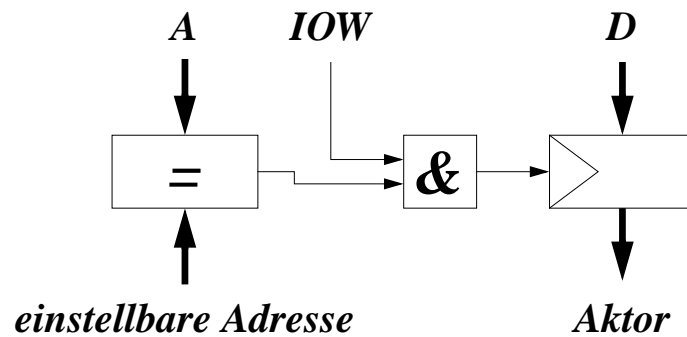
I/O (Input)

Aufbau eines I/O-Input-Registers (schematisch):



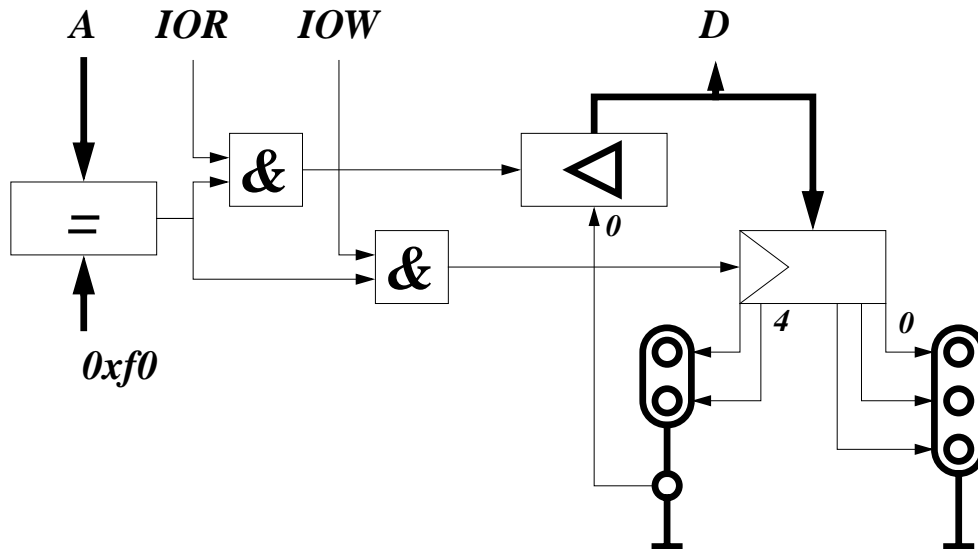
I/O (Output)

Aufbau eines I/O-Output-Registers (schematisch):



I/O (Beispiel)

Fußgänger-Ampel:



I/O (Beispiel)

Ampel-Kommandos:

<i>Fußgänger-Ampel</i>	<i>Fahrzeug-Ampel</i>	<i>Code</i>
<i>aus</i>	<i>aus</i>	<i>0x00</i>
<i>rot</i>	<i>grün</i>	<i>0x24</i>
<i>rot</i>	<i>gelb</i>	<i>0x22</i>
<i>rot</i>	<i>rot</i>	<i>0x21</i>
<i>grün</i>	<i>rot</i>	<i>0x11</i>
<i>rot</i>	<i>rot</i>	<i>0x21</i>
<i>rot</i>	<i>rot-gelb</i>	<i>0x23</i>

I/O (Beispiel)

Ampel-Programm:

```
    outb $0x24, 0xf0                /* Fahrzeug-Ampel rot */
    /* Fahrzeug-Ampel grün*/       /* Fußgänger-Ampel grün */
    /* Fußgänger-Ampel rot */     call wait_1min

    /* Auf Knopfdruck warten */    /* Umschalten */
loop:                               outb 0x21, 0xf0
    inb 0xf0, %al                  call wait_5sec
    cmpb $1, %al                   outb 0x23, 0xf0
    jne loop                        call wait_5sec
                                    outb 0x24, 0xf0

    /* Umschalten */              /* Fahrzeug-Ampel grün*/
    outb $0x22, 0xf0               /* Fußgänger-Ampel rot */
    call wait_5sec                  call wait_5min
    outb $0x21, 0xf0
    call wait_5sec
    outb $0x11, 0xf0                jmp loop
```