

Portbit einzeln setzen und rücksetzen



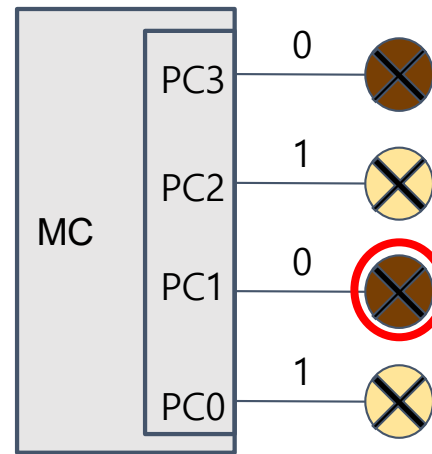
Bitmanipulationen

Hallo, ich bin Mik, Dein
Mikrocontroller

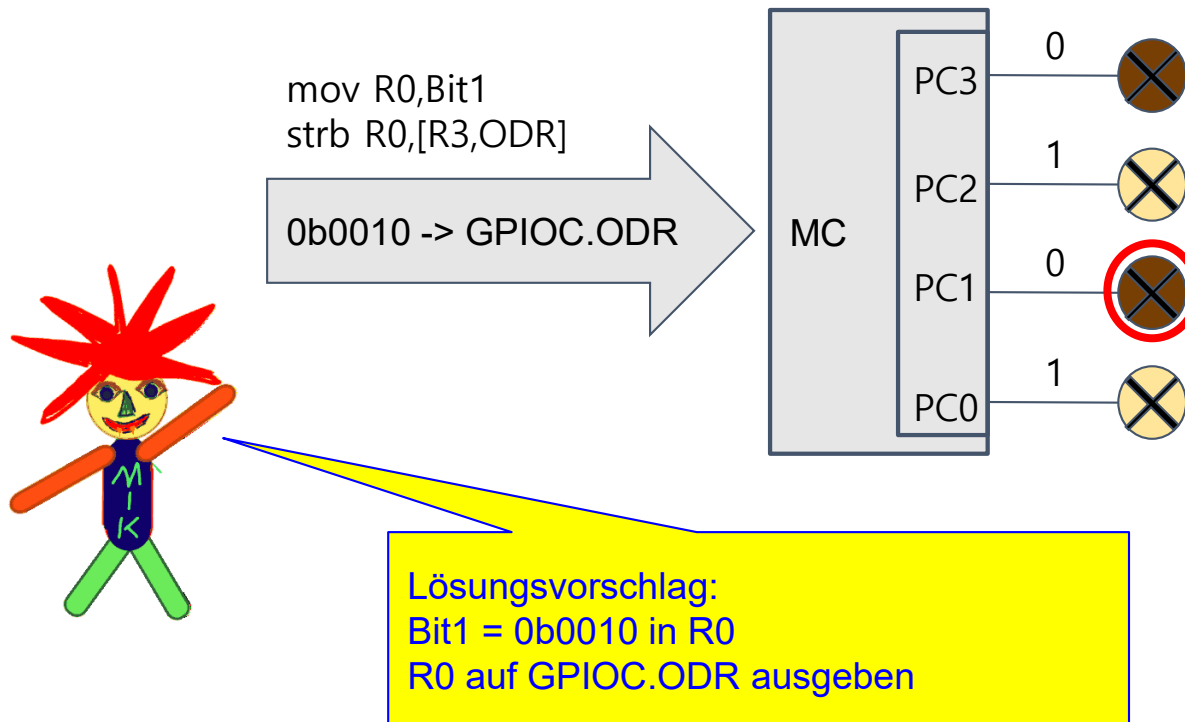


Portbit einzeln setzen und rücksetzen

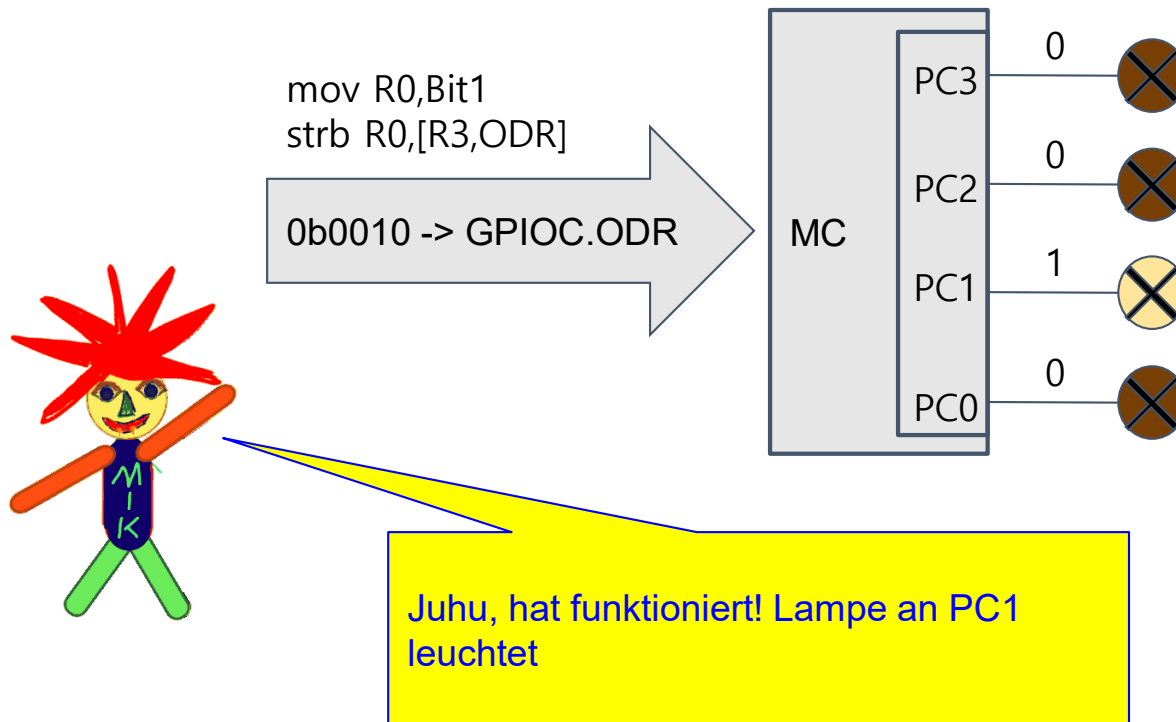
Aufgabe: Die
Lampe an
PC1 soll
leuchten



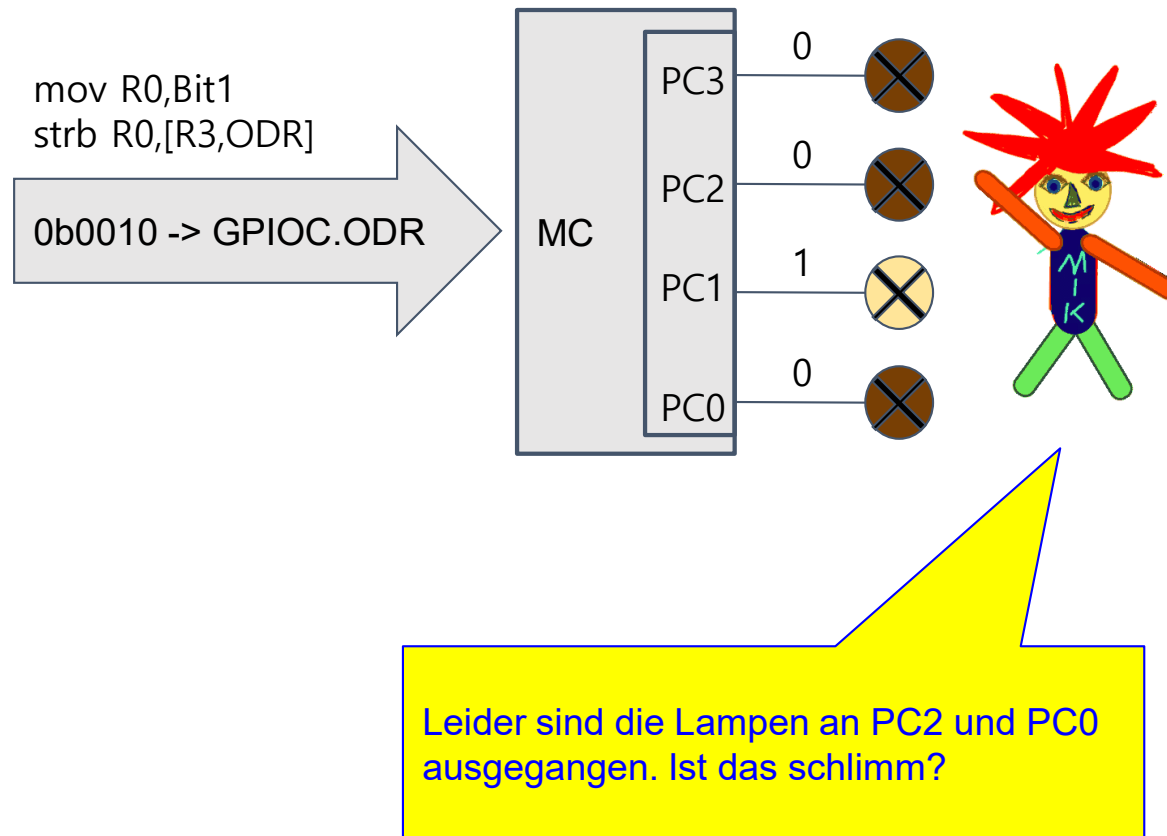
Portbit einzeln setzen und rücksetzen



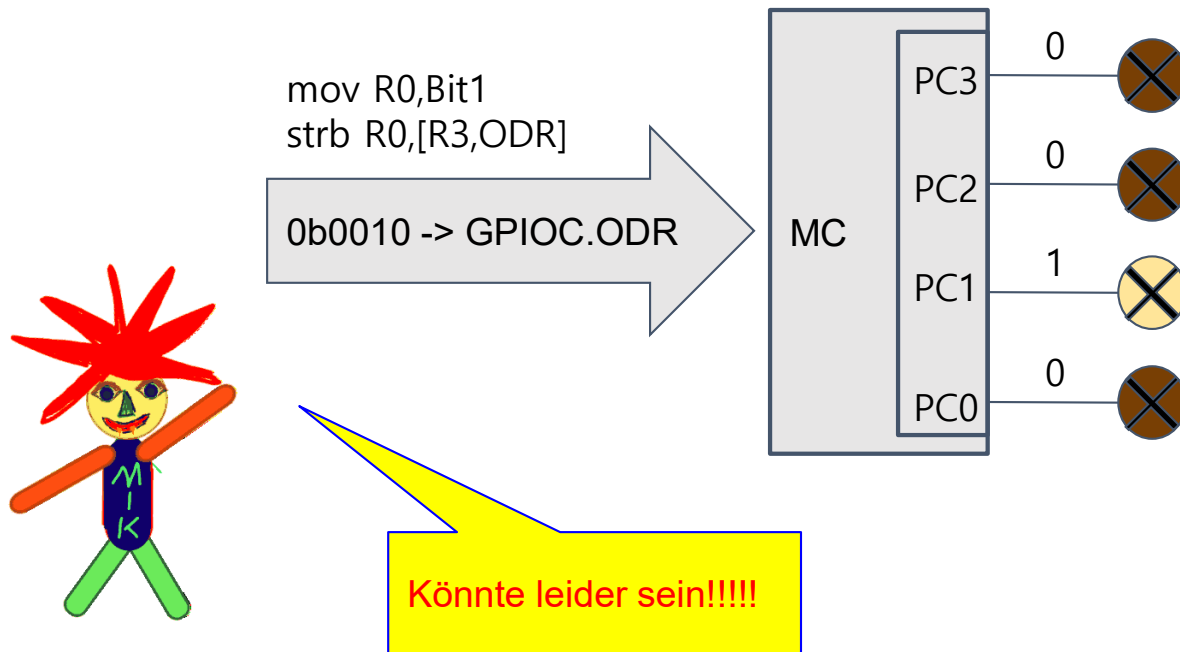
Portbit einzeln setzen und rücksetzen



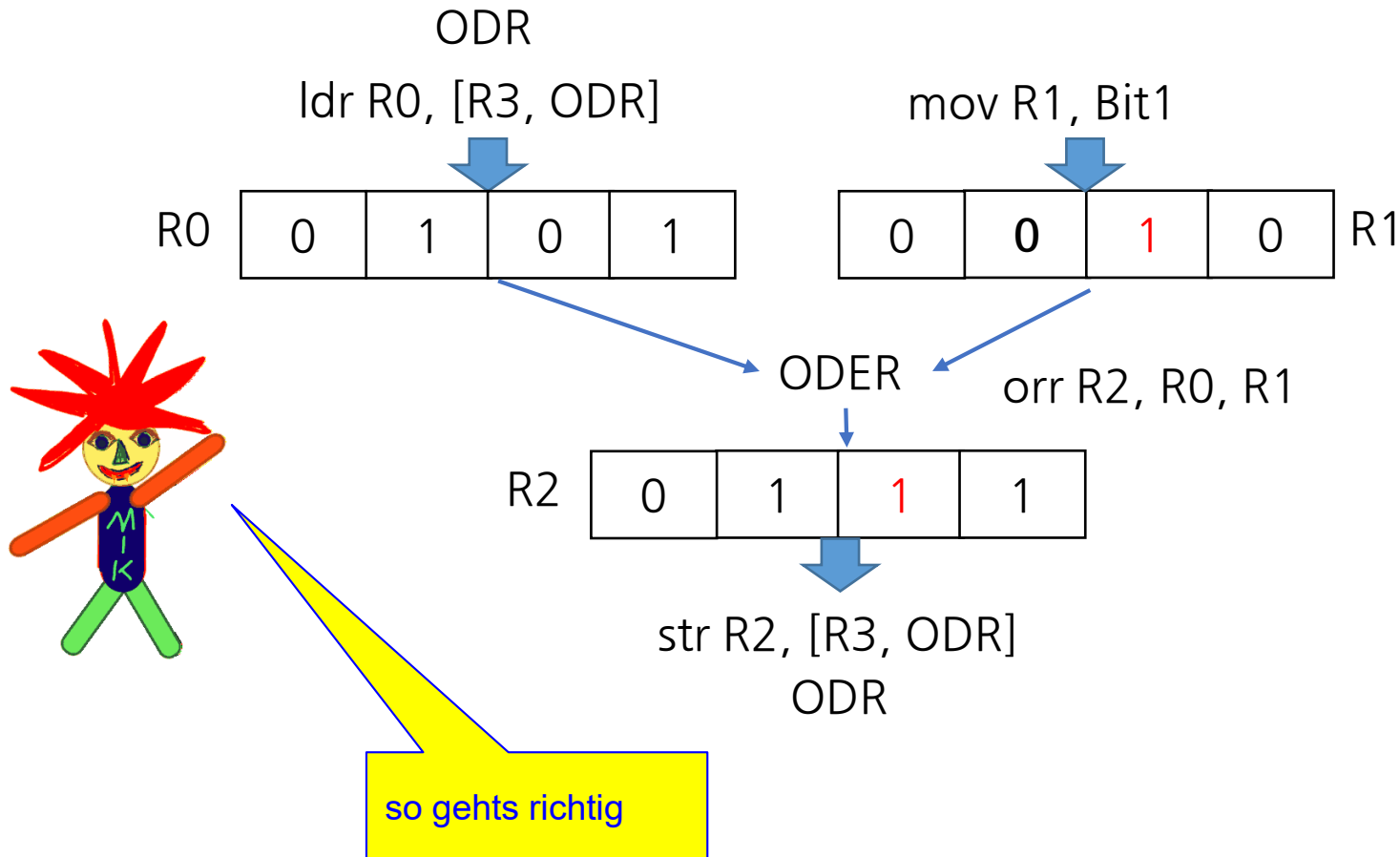
Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen

alter Inhalt von ODR in R0
laden

ODR

ldr R0, [R3, ODR]

R0

0	1	0	1
---	---	---	---

mov R1, Bit1

0	0	1	0
---	---	---	---

R1

ODR

orr R2, R0, R1

R2

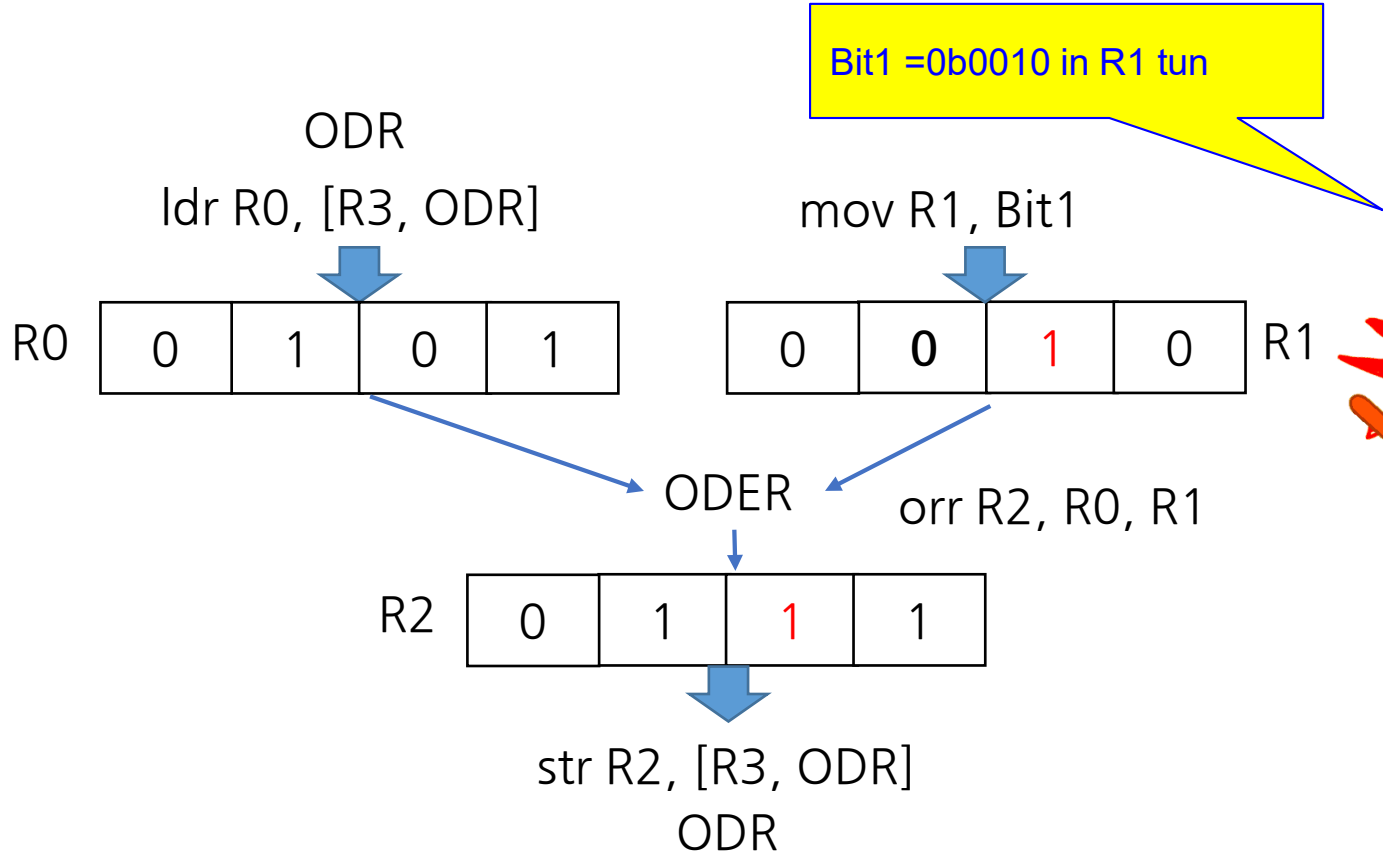
0	1	1	1
---	---	---	---

str R2, [R3, ODR]

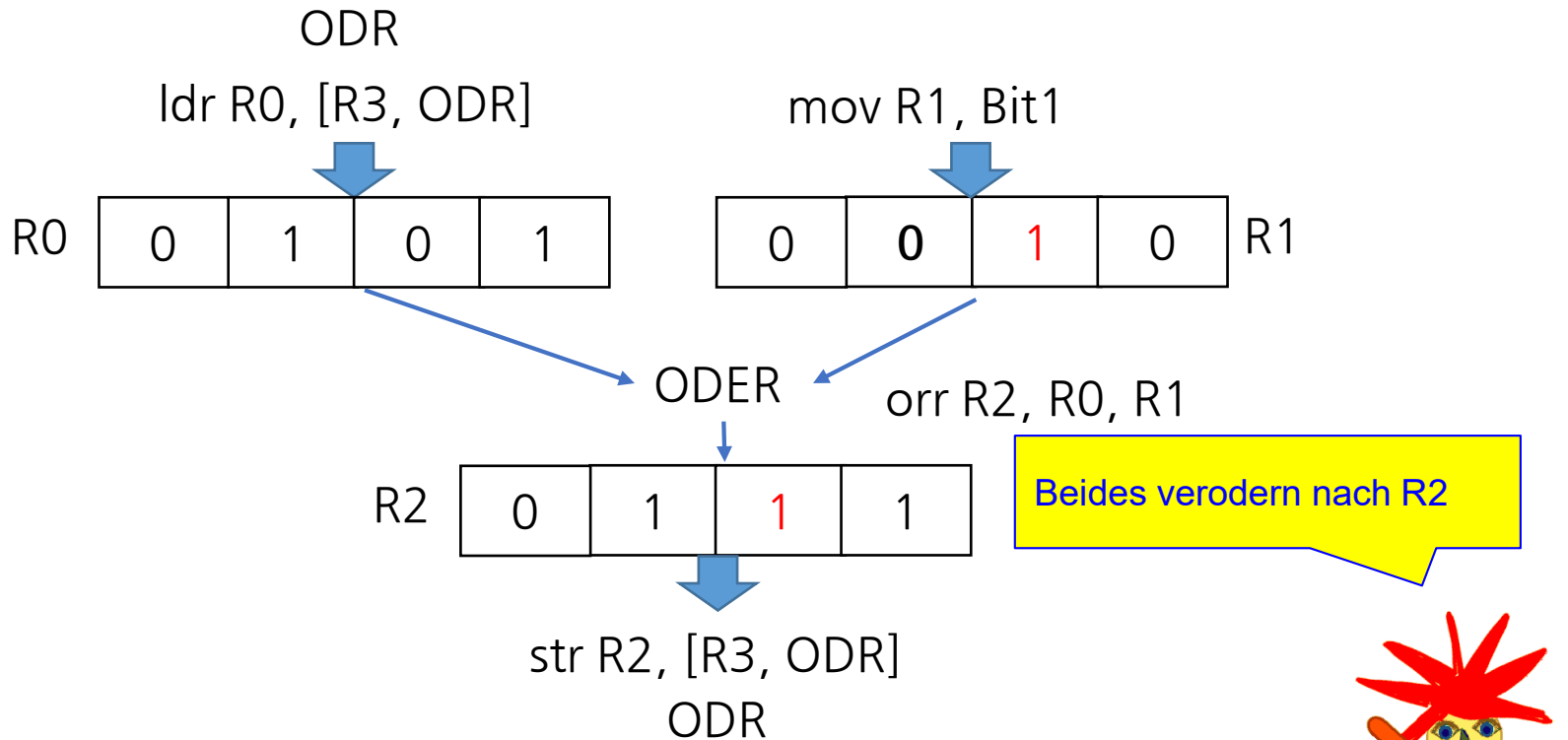
ODR



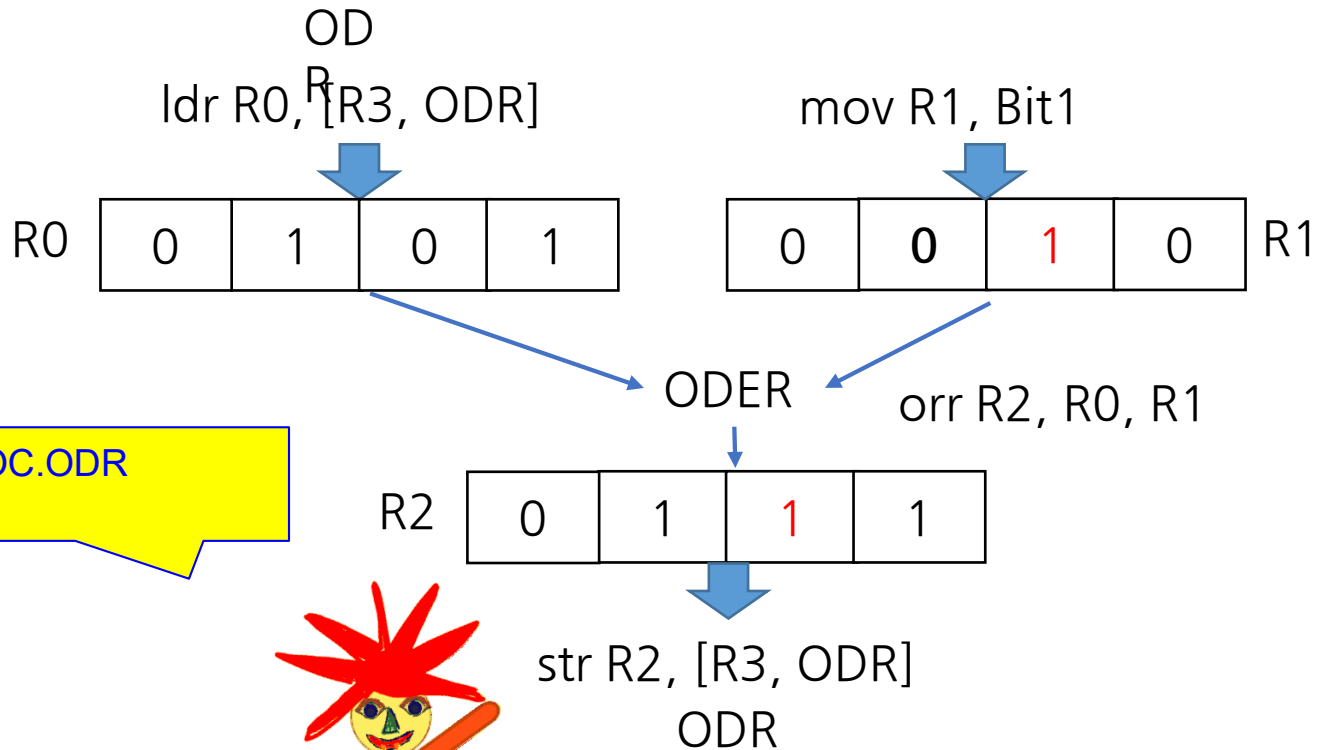
Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen



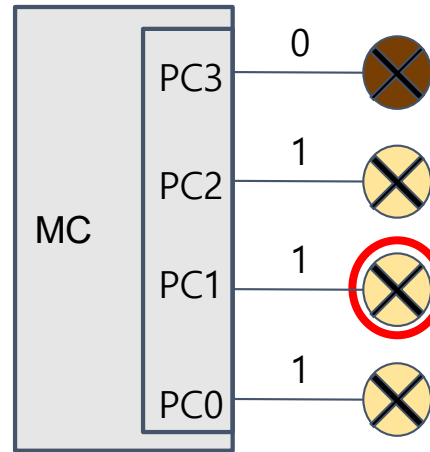
Portbit einzeln setzen und rücksetzen



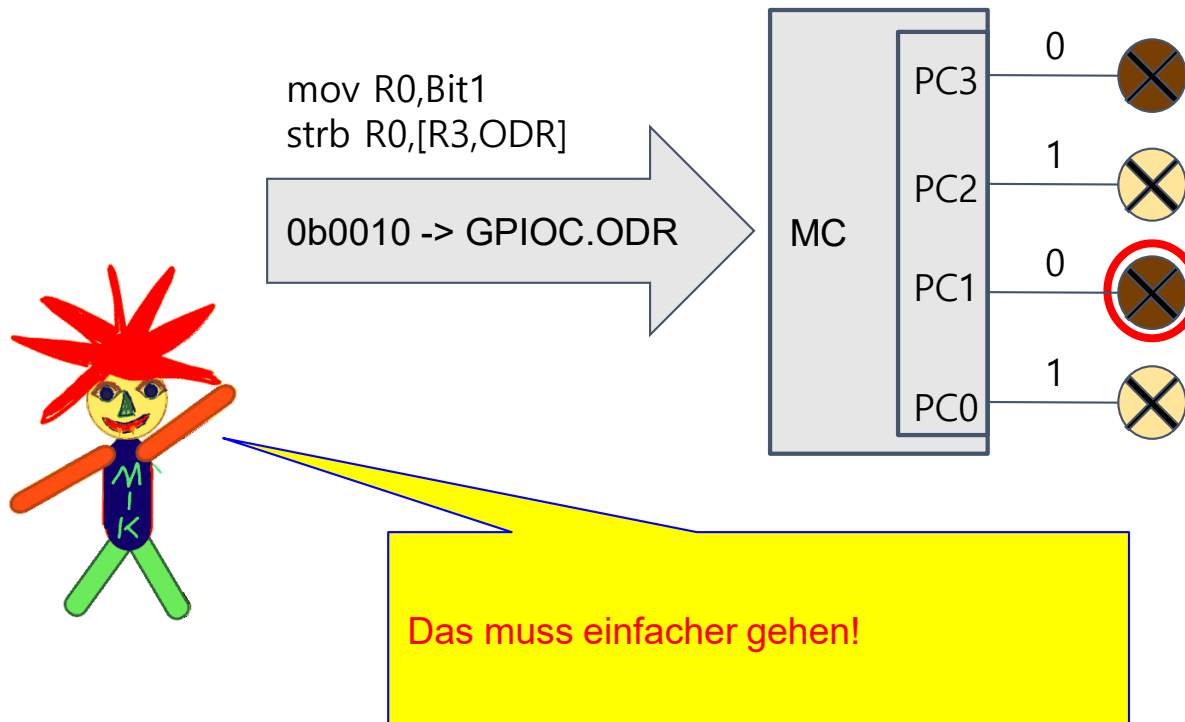
Portbit einzeln setzen und rücksetzen



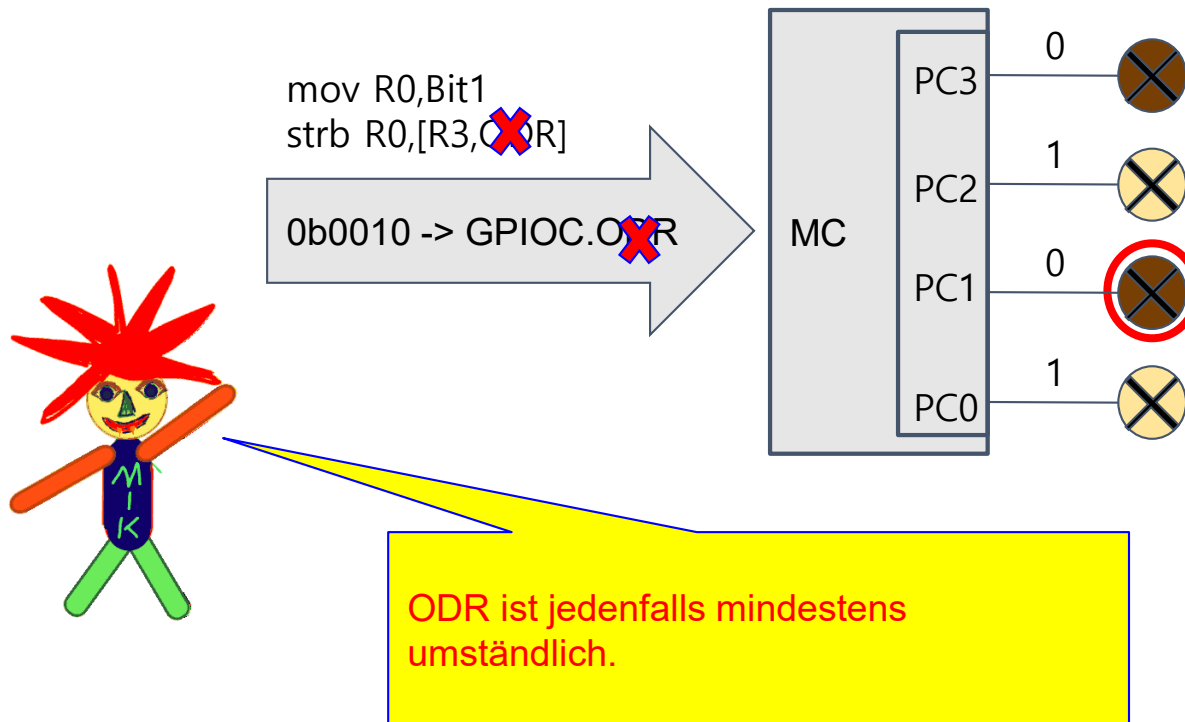
So gehört
das!!



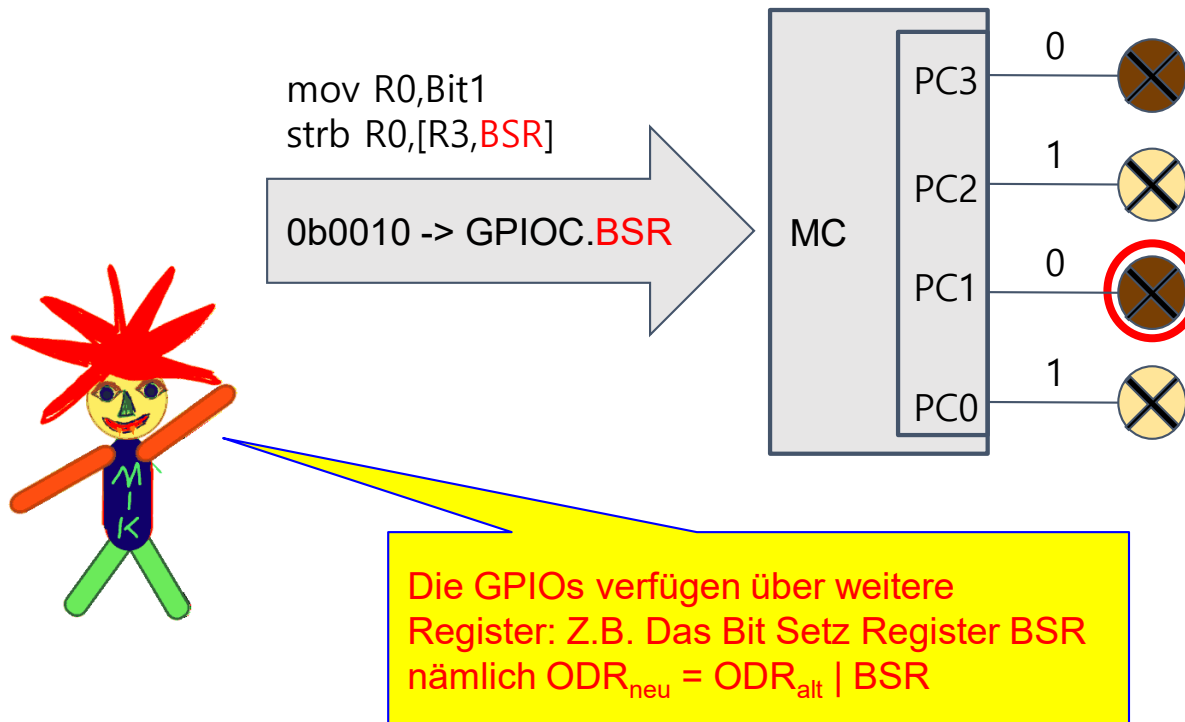
Portbit einzeln setzen und rücksetzen



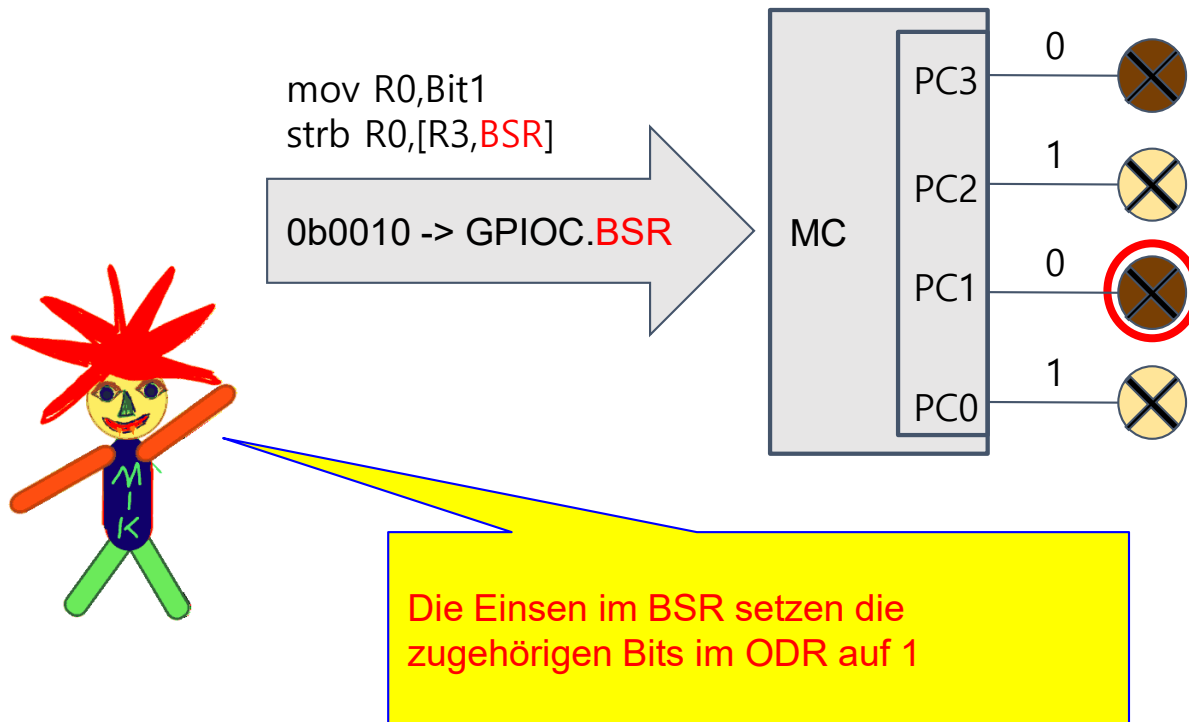
Portbit einzeln setzen und rücksetzen



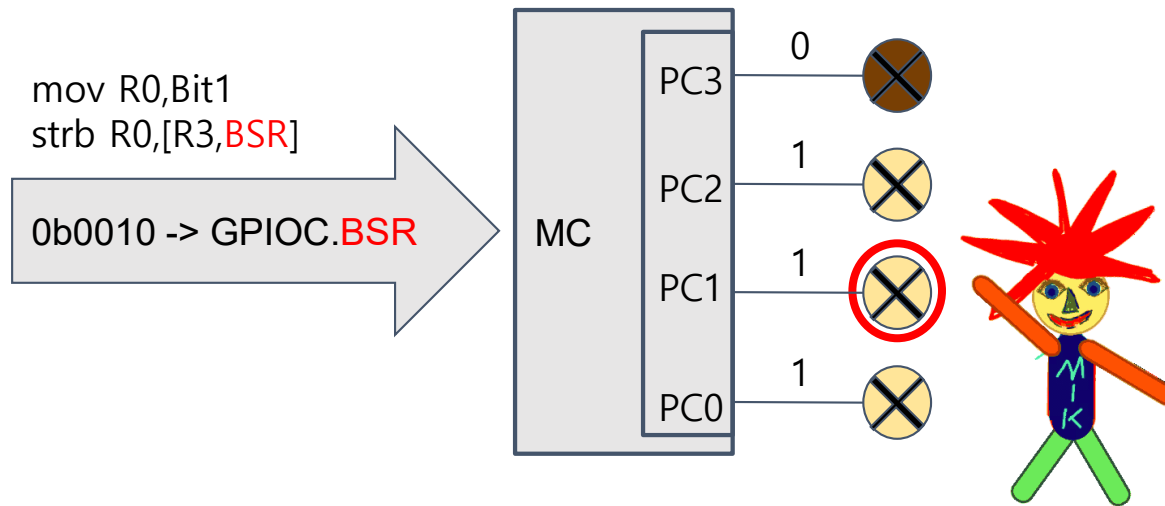
Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen

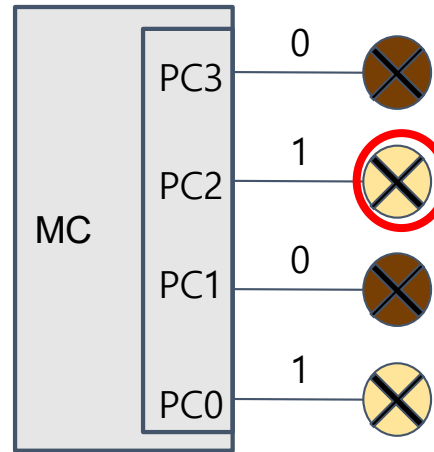


Ergebnis: PC1 wird 1, die übrigen Portbits bleiben unverändert!!

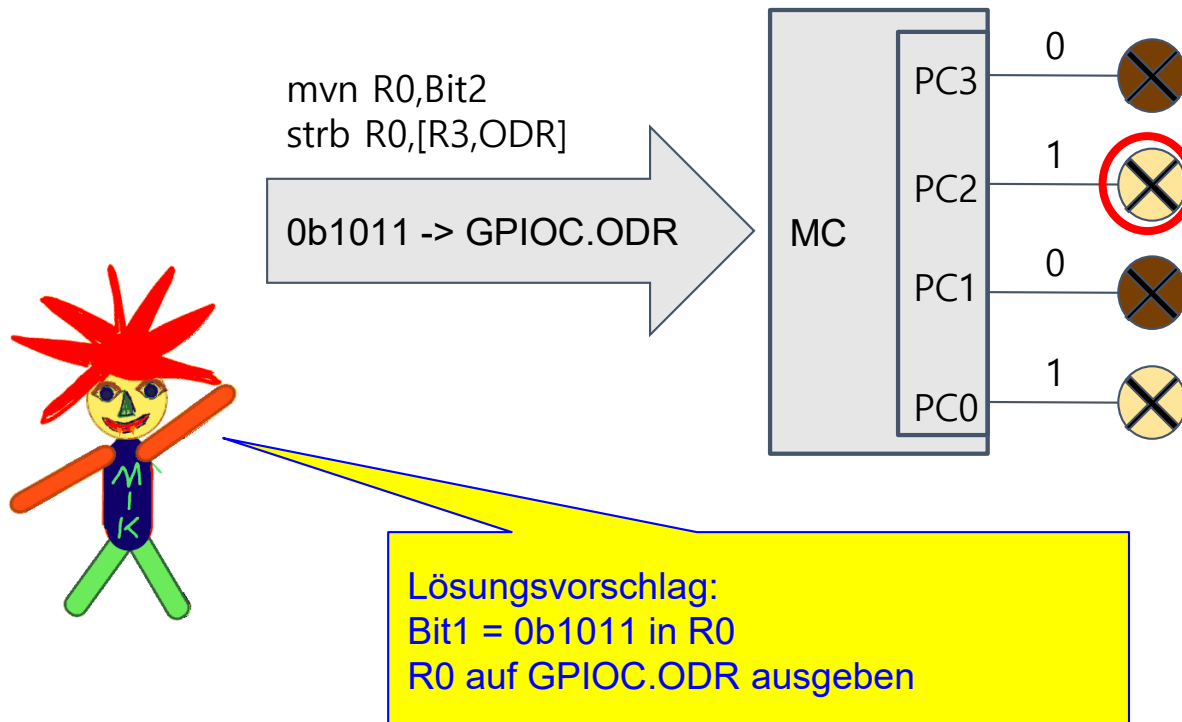


Portbit einzeln setzen und rücksetzen

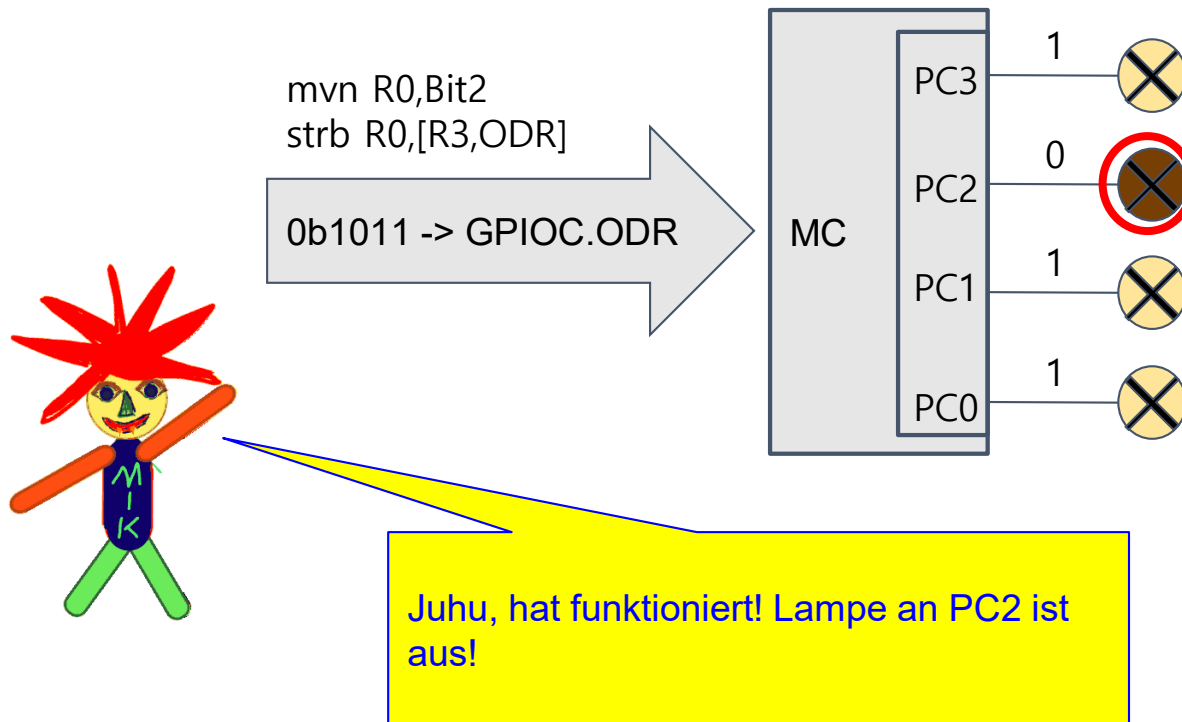
Aufgabe: Die
Lampe an PC2
soll nicht mehr
leuchten



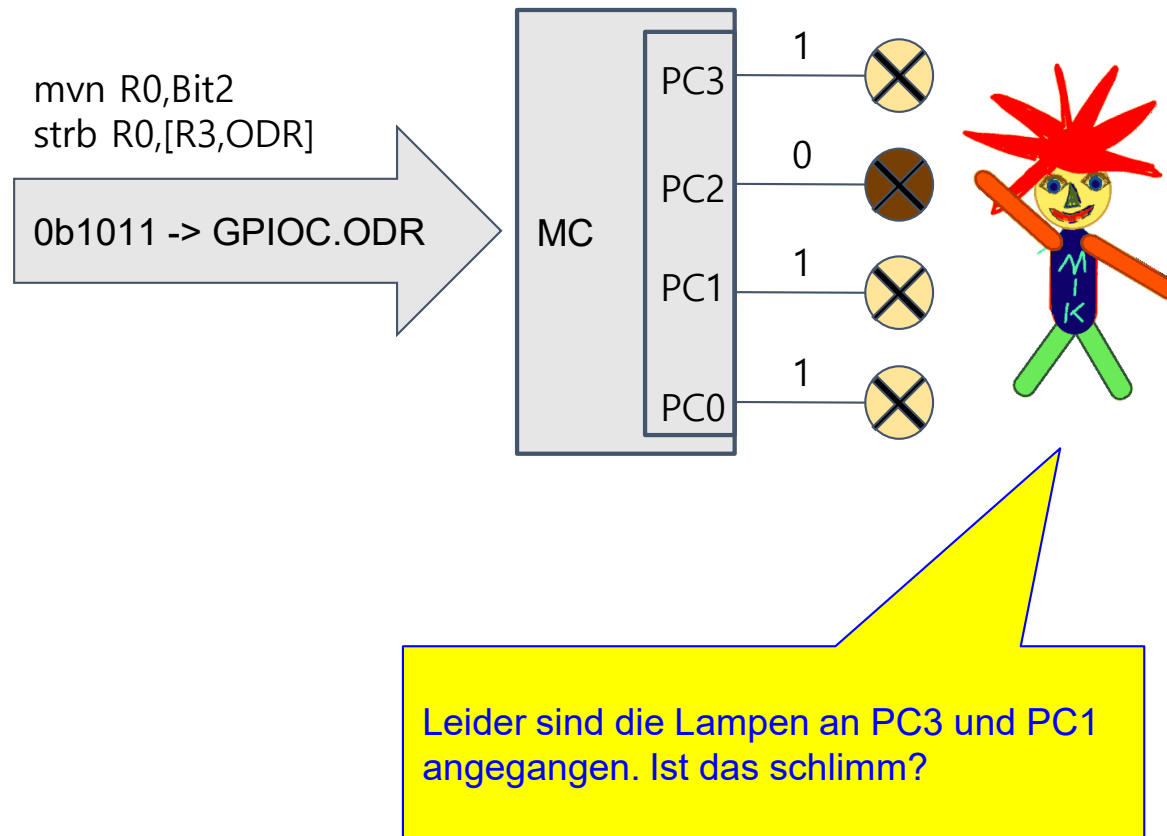
Portbit einzeln setzen und rücksetzen



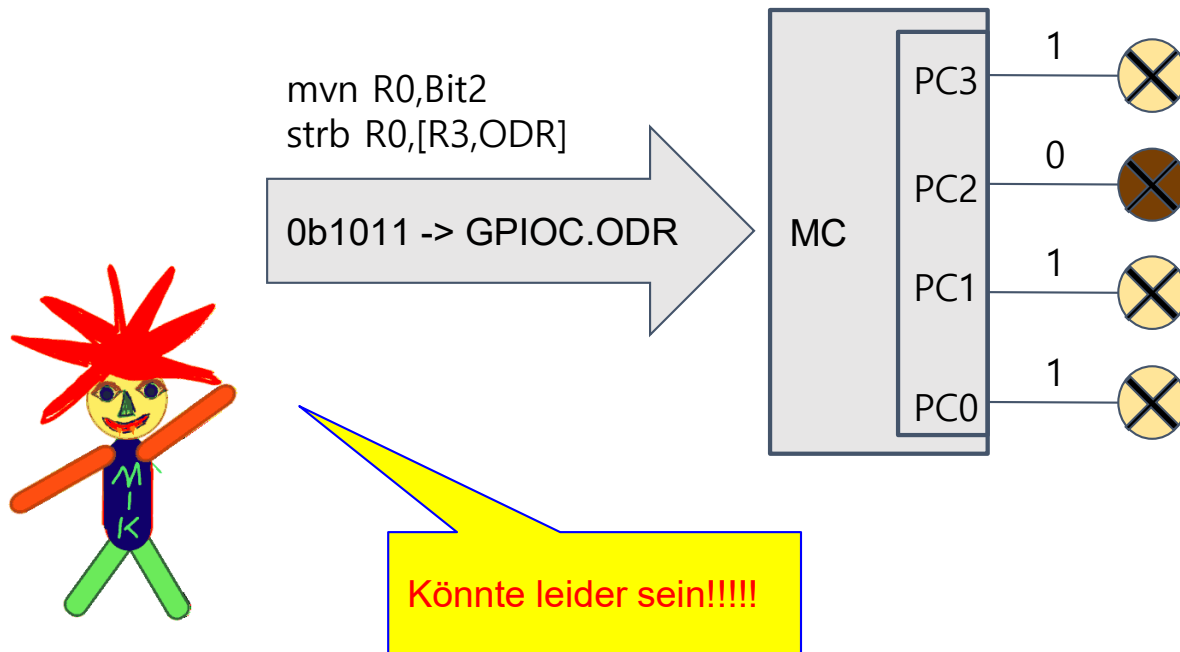
Portbit einzeln setzen und rücksetzen



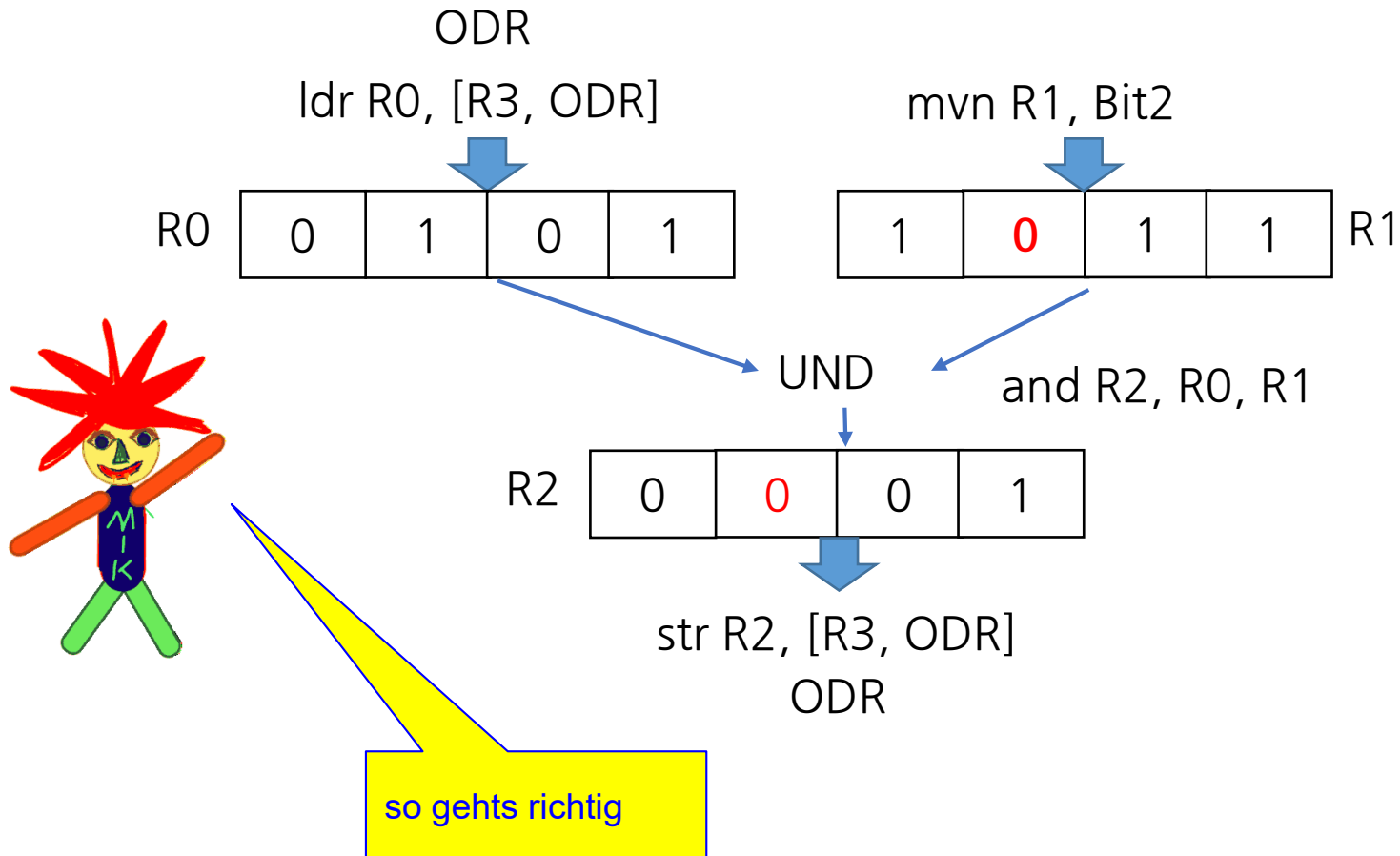
Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen

alter Inhalt von ODR in R0
laden

ODR

ldr R0, [R3, ODR]

R0

0	1	0	1
---	---	---	---

mvn R1, Bit2

1	0	1	1
---	---	---	---

R1

UND

and R2, R0, R1

R2

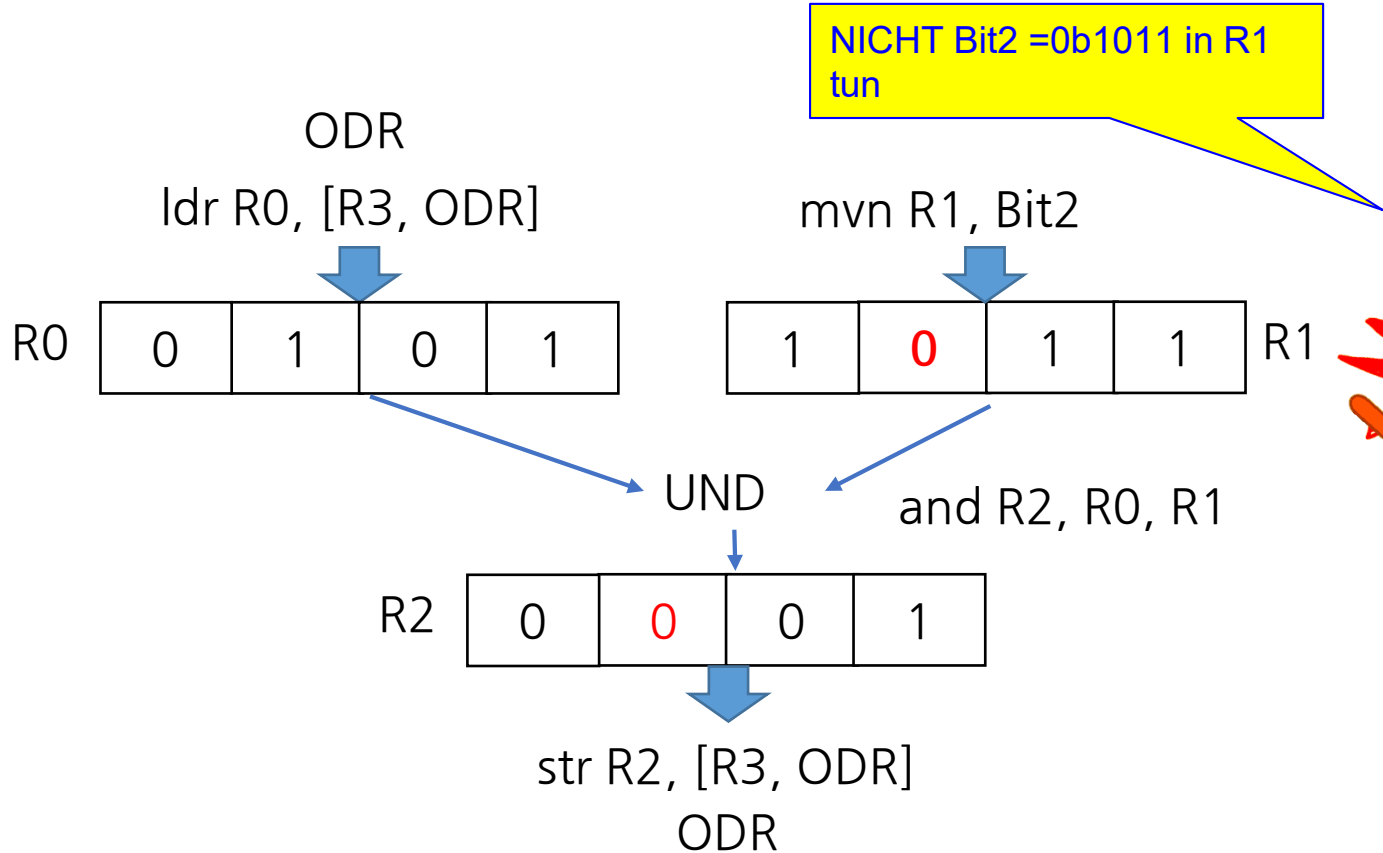
0	0	0	1
---	---	---	---

str R2, [R3, ODR]

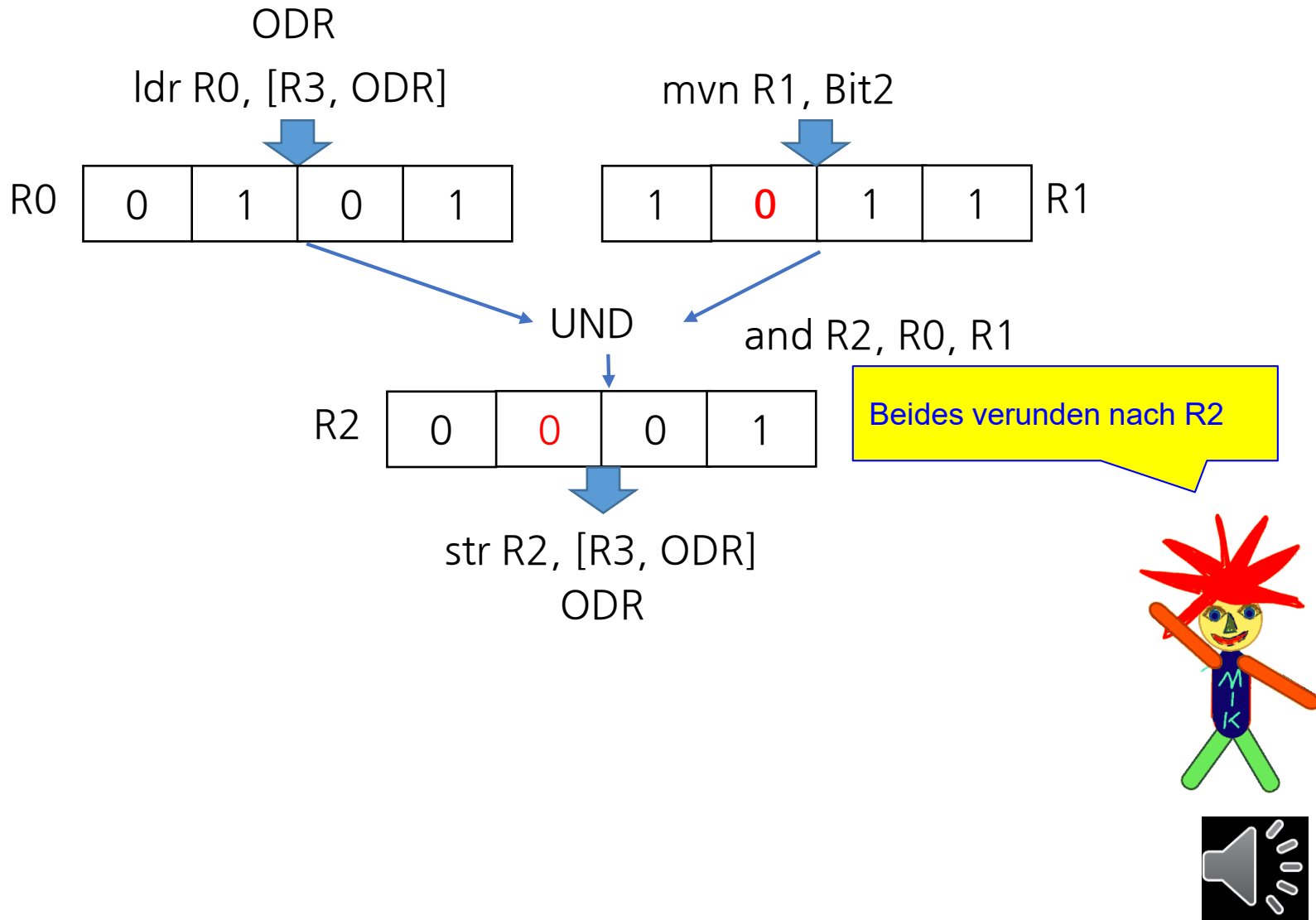
ODR



Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen

ODR

ldr R0, [R3, ODR]



mvn R1, Bit2



UND

and R2, R0, R1

R2



str R2, [R3, ODR]
ODR

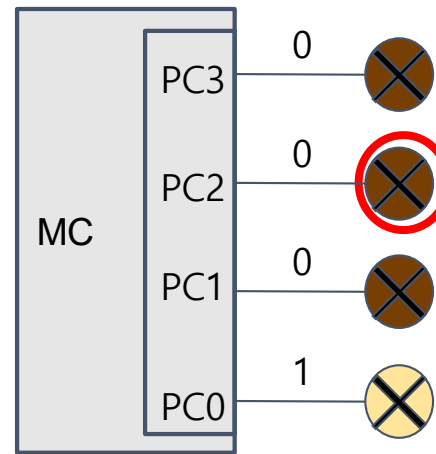
R2 auf GPIOC.ODR
ausgeben



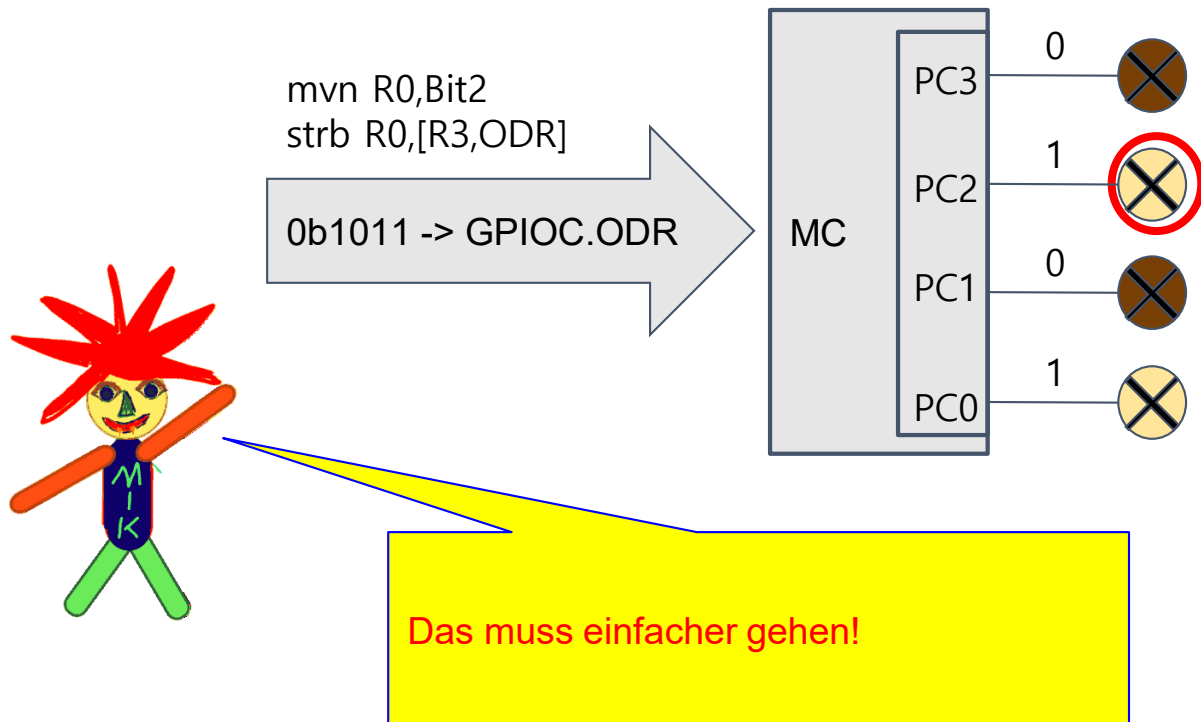
Portbit einzeln setzen und rücksetzen



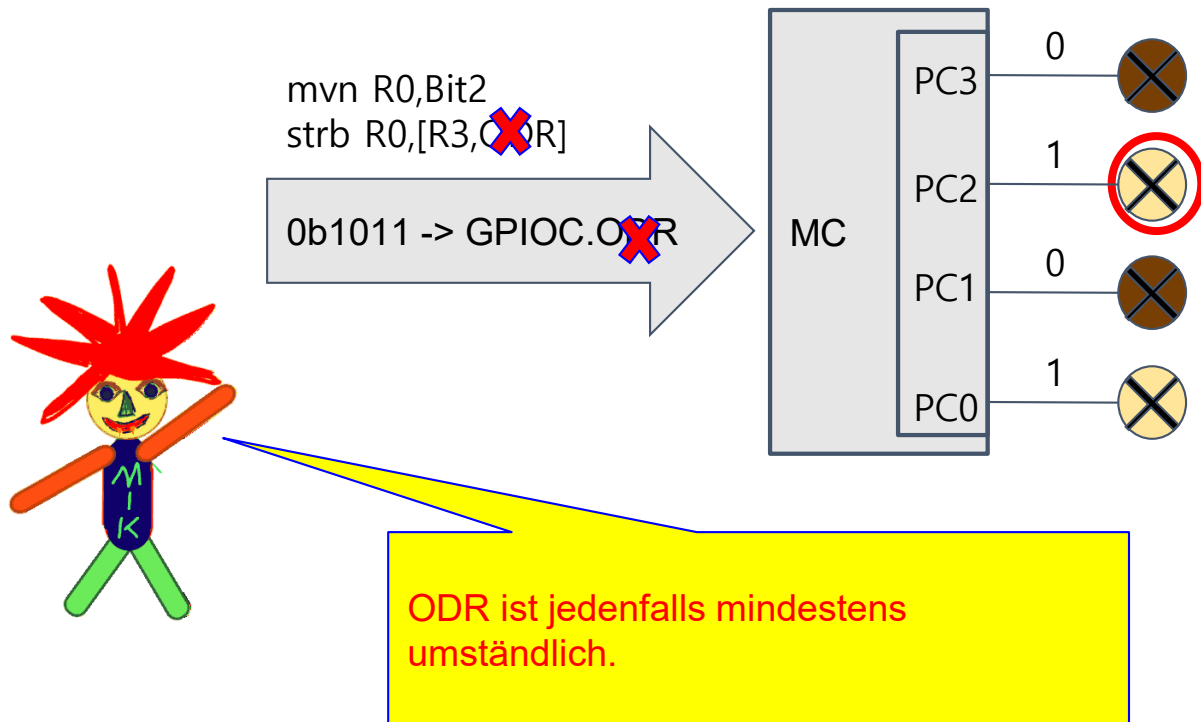
So gehört
das!!



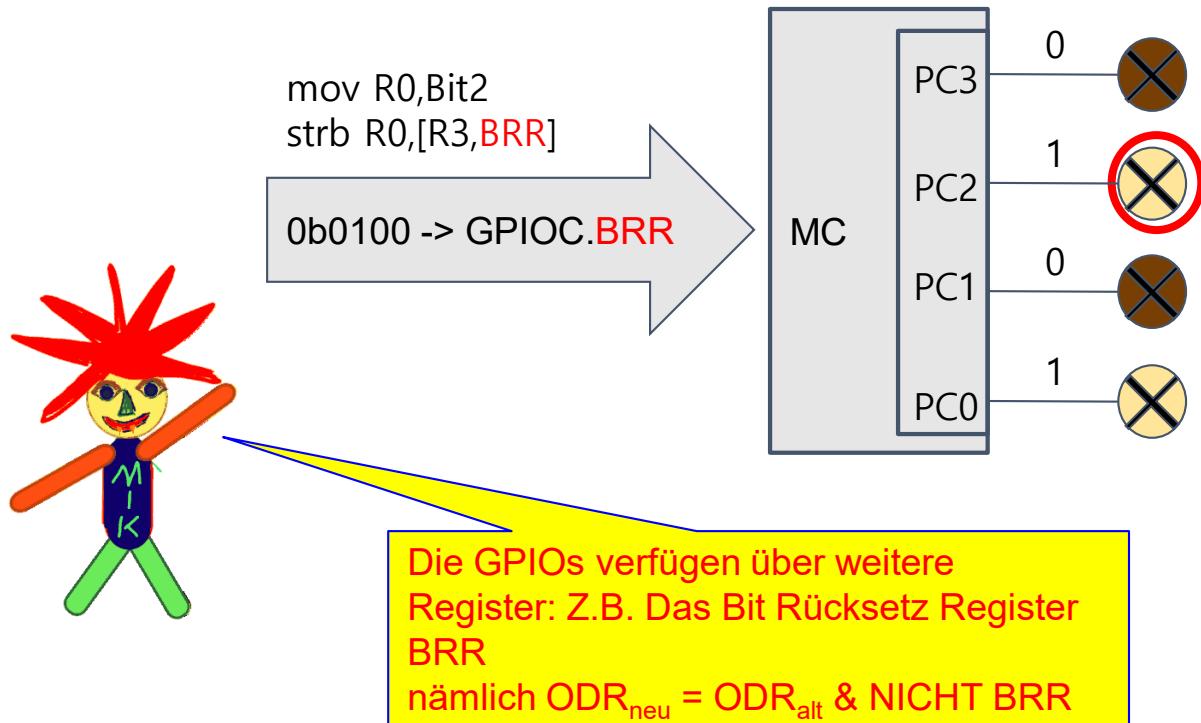
Portbit einzeln setzen und rücksetzen



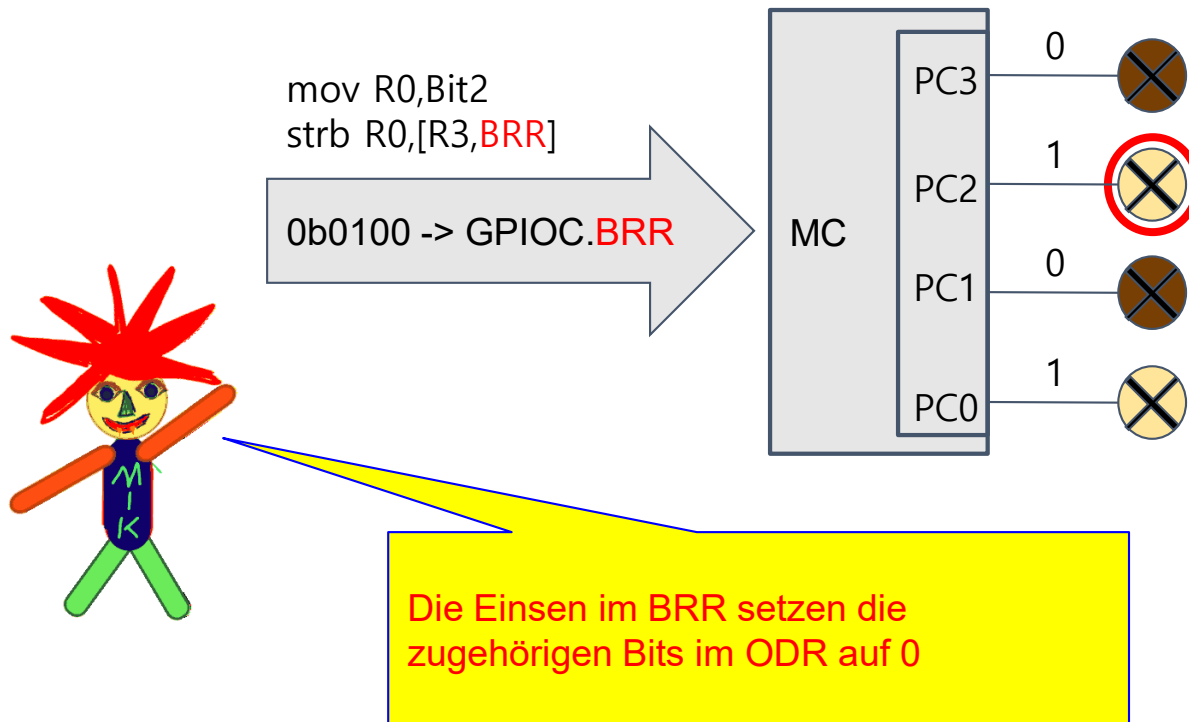
Portbit einzeln setzen und rücksetzen



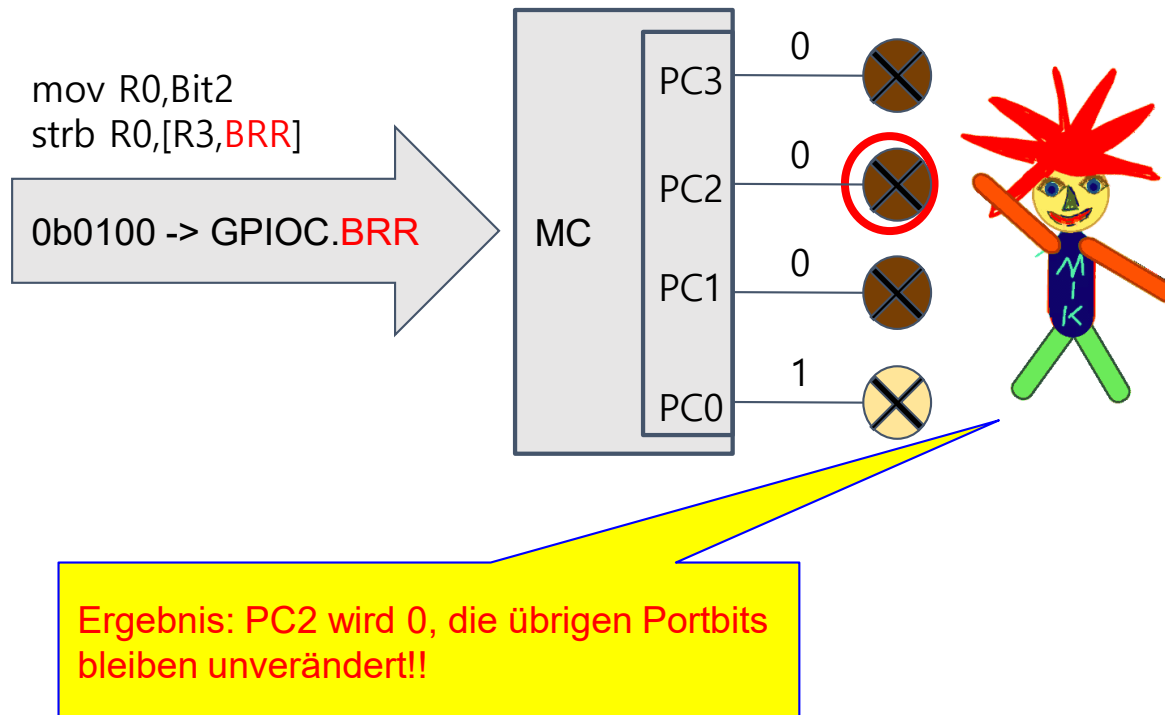
Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen



Portbit einzeln setzen und rücksetzen



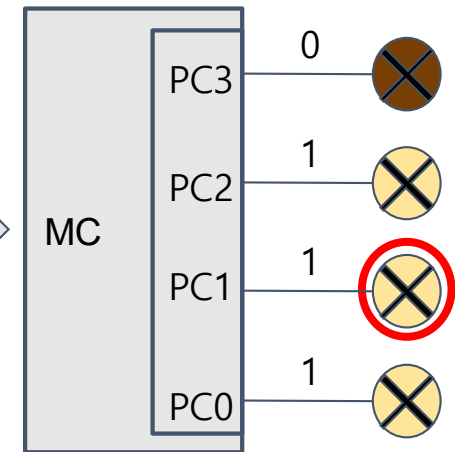
Portbit einzeln setzen und rücksetzen

Ergebnis: Eine 1 im BSR setzt das betreffende Bit im ODR auf 1 und ändert sonst nichts



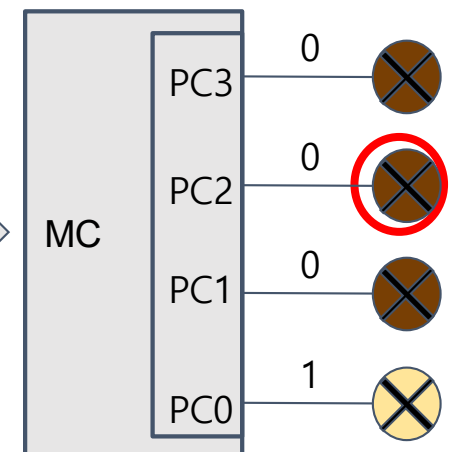
```
mov R0,Bit1  
strb R0,[R3,BSR]
```

0b0010 -> GPIOC.BSR



```
mov R0,Bit2  
strb R0,[R3,BRR]
```

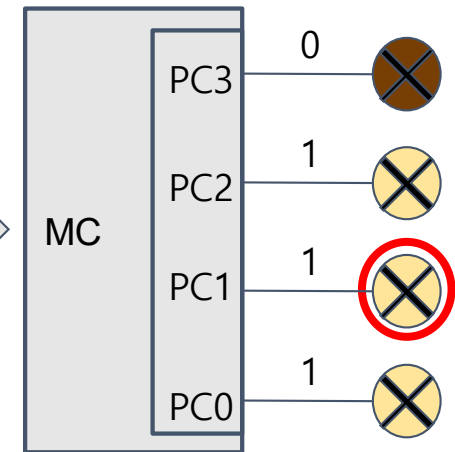
0b0100 -> GPIOC.BRR



Portbit einzeln setzen und rücksetzen

```
mov R0,Bit1  
strb R0,[R3,BSR]
```

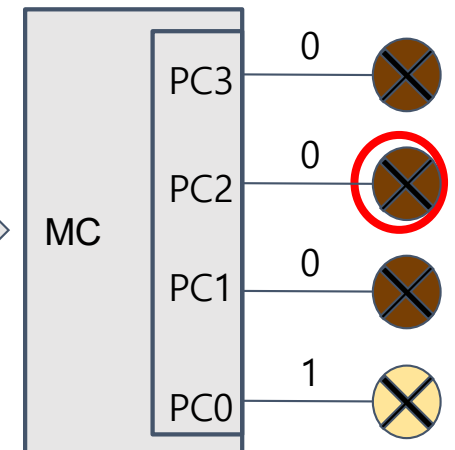
0b0010 -> GPIOC.**BSR**



Ergebnis: Eine 1 im BRR setzt das betreffende Bit im ODR auf 0 zurück und ändert sonst nichts

```
mov R0,Bit2  
strb R0,[R3,BRR]
```

0b0100 -> GPIOC.**BRR**



Portbit einzeln setzen und rücksetzen



Juhu! Mit den Registern GPIOC.BSR und GPIOC.BRR können wir einzelne Portbits sehr leicht auf 1 setzen oder auf Null zurücksetzen. Einfach die betreffenden Bits statt auf ODR auf BSR oder BRR ausgeben:

```
ldr R3,=GPIOC
mov R0,Bit1
strb R0,[R3,BSR]      //PC1 auf 1 setzen
mov R0,Bit2
strb R0,[R3,BRR]      //PC2 auf 0 rücksetzen
```

```
mov R0,Bit2
strb R0,[R3,BRR]
```

0b0100 -> GPIOC.BRR

