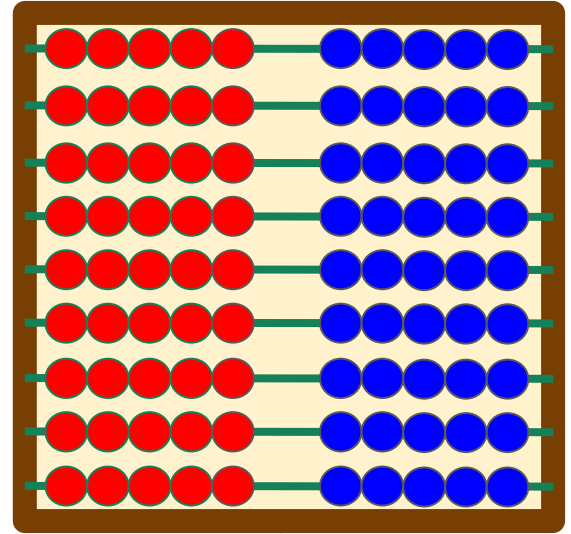


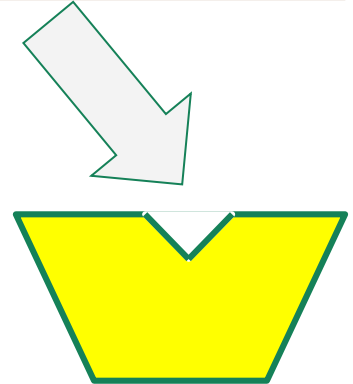
# Arithmetisch- Logische Einheit



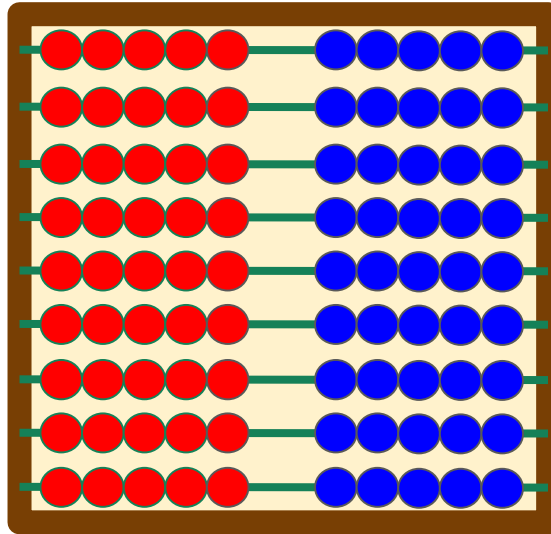
Mikrocontrollerbasics



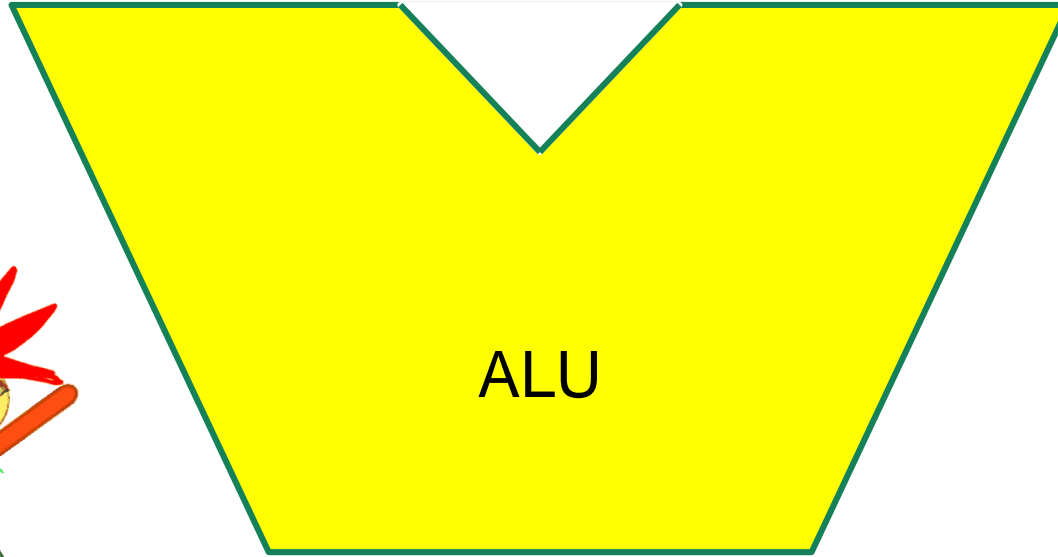
Ich bin Mik, Dein  
Mikrocontroller



Wie funktioniert eigentlich das Rechenwerk?



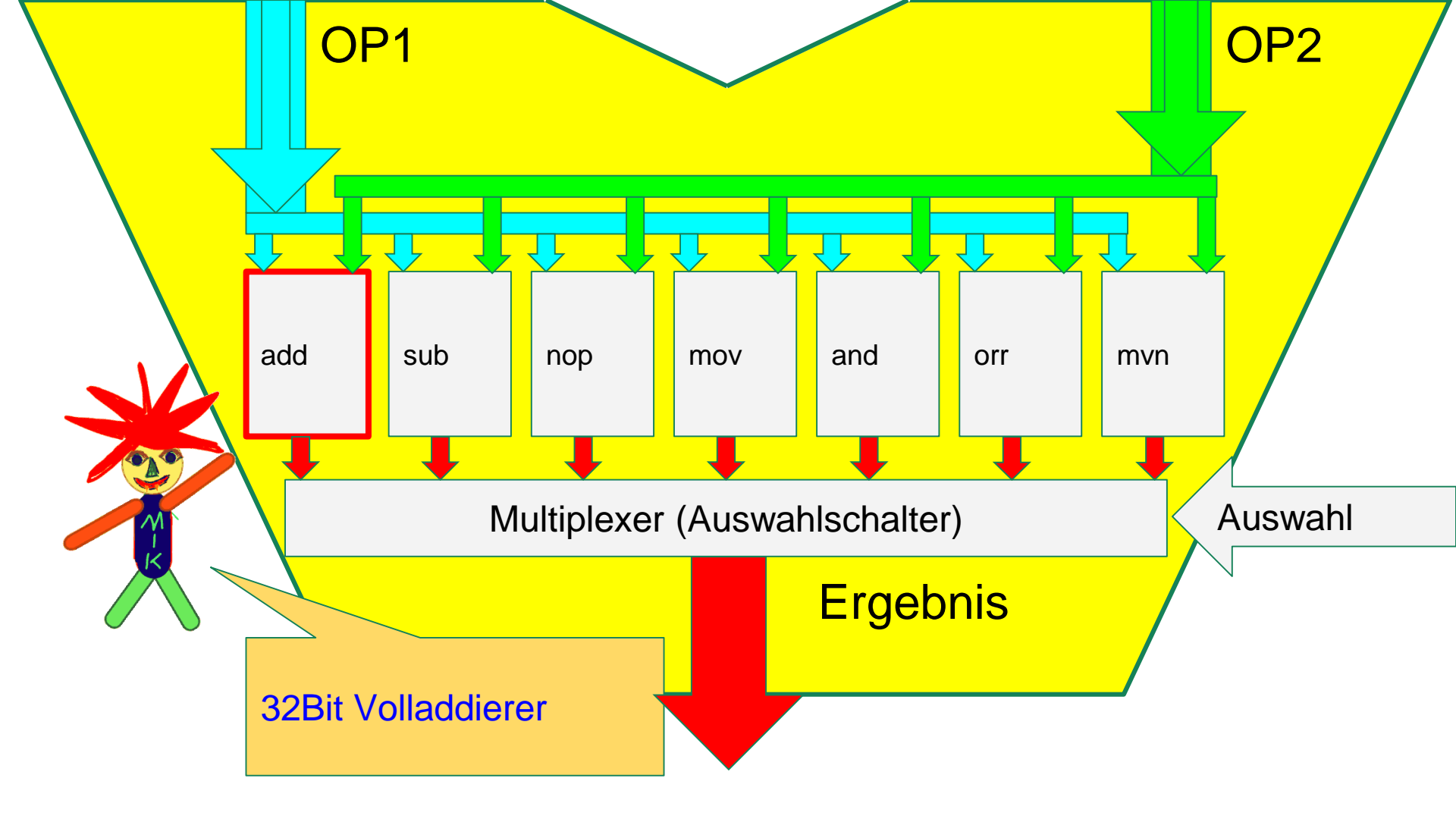
Das Rechenwerk

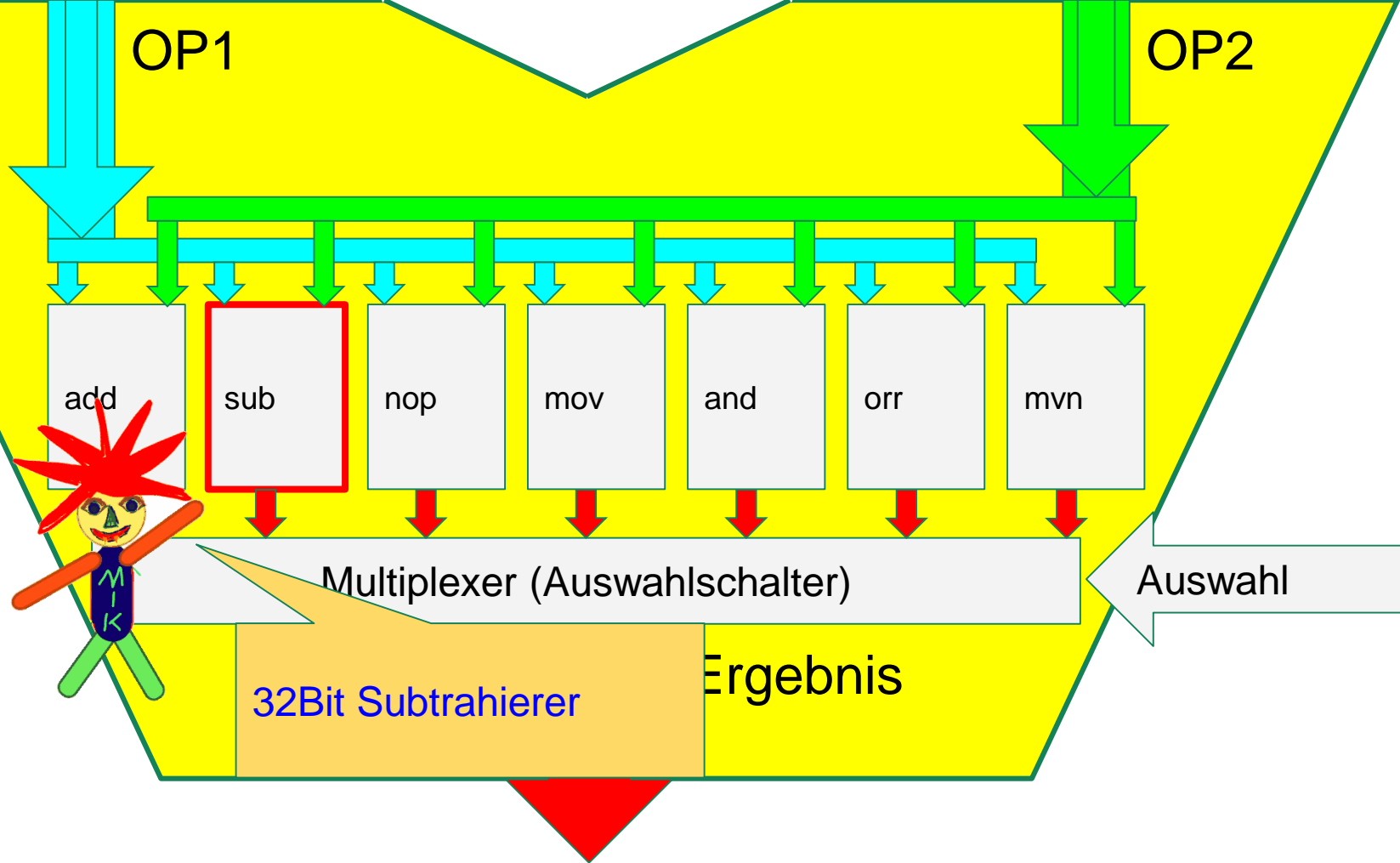


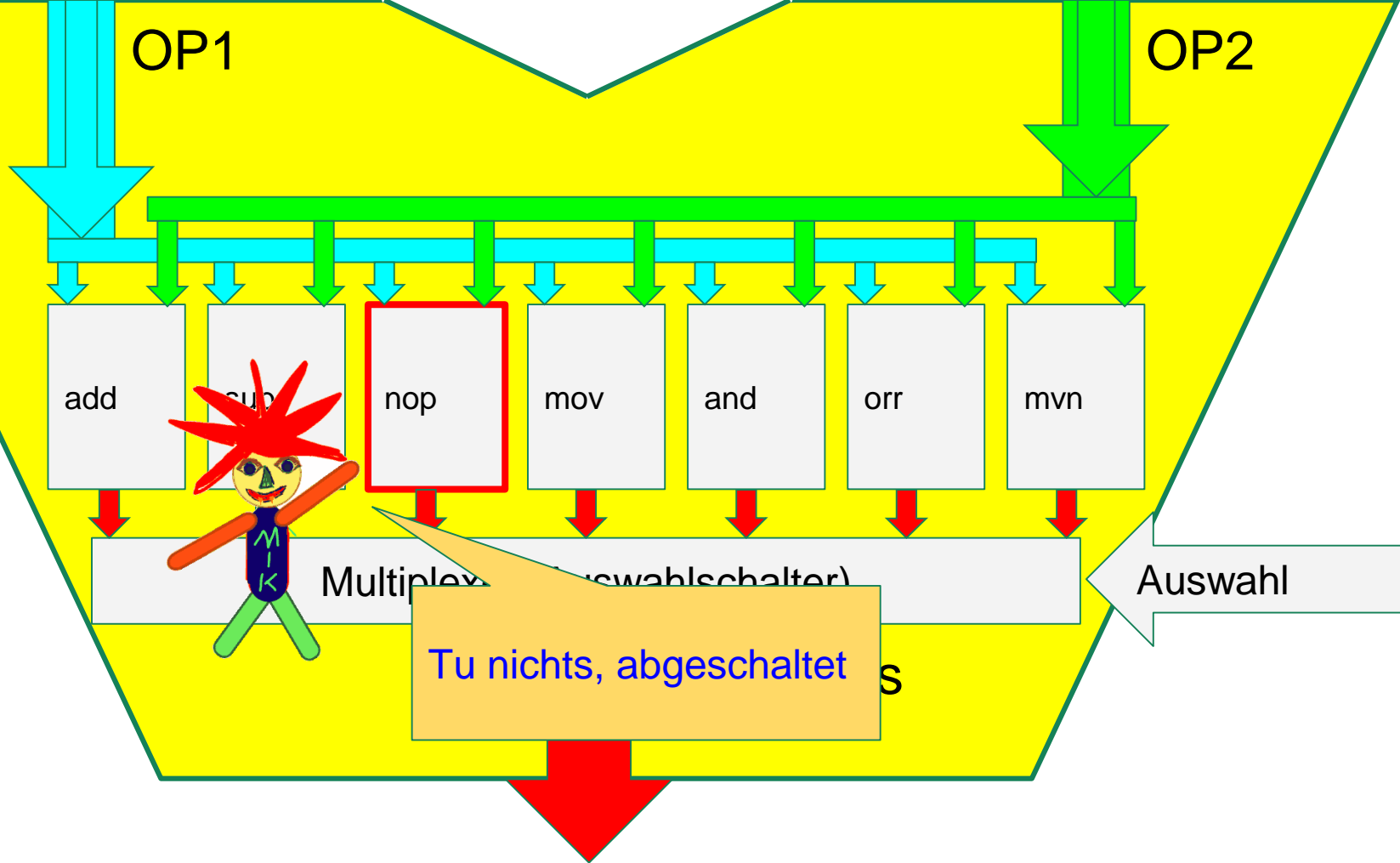
ALU

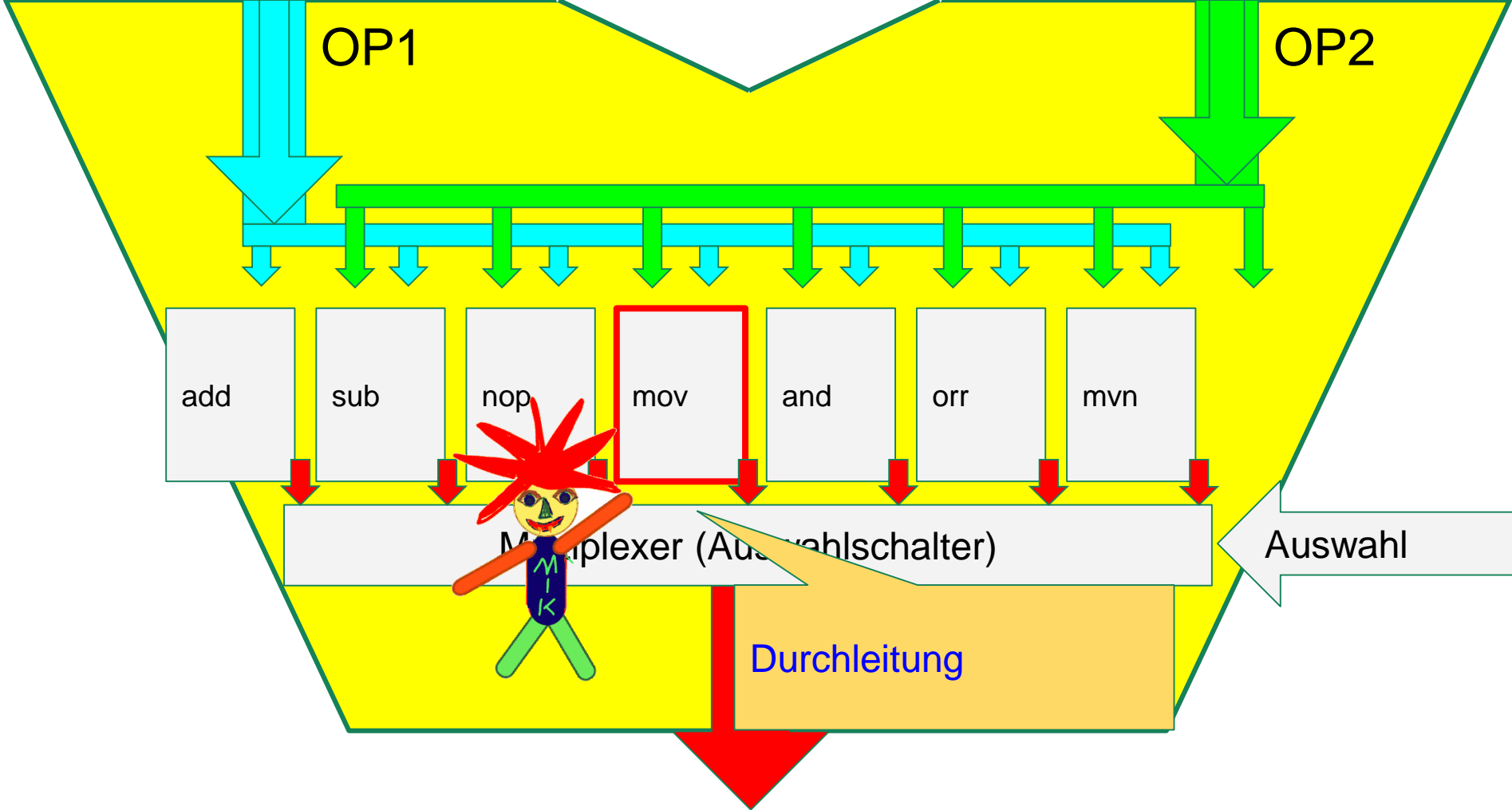
Das Rechenwerk

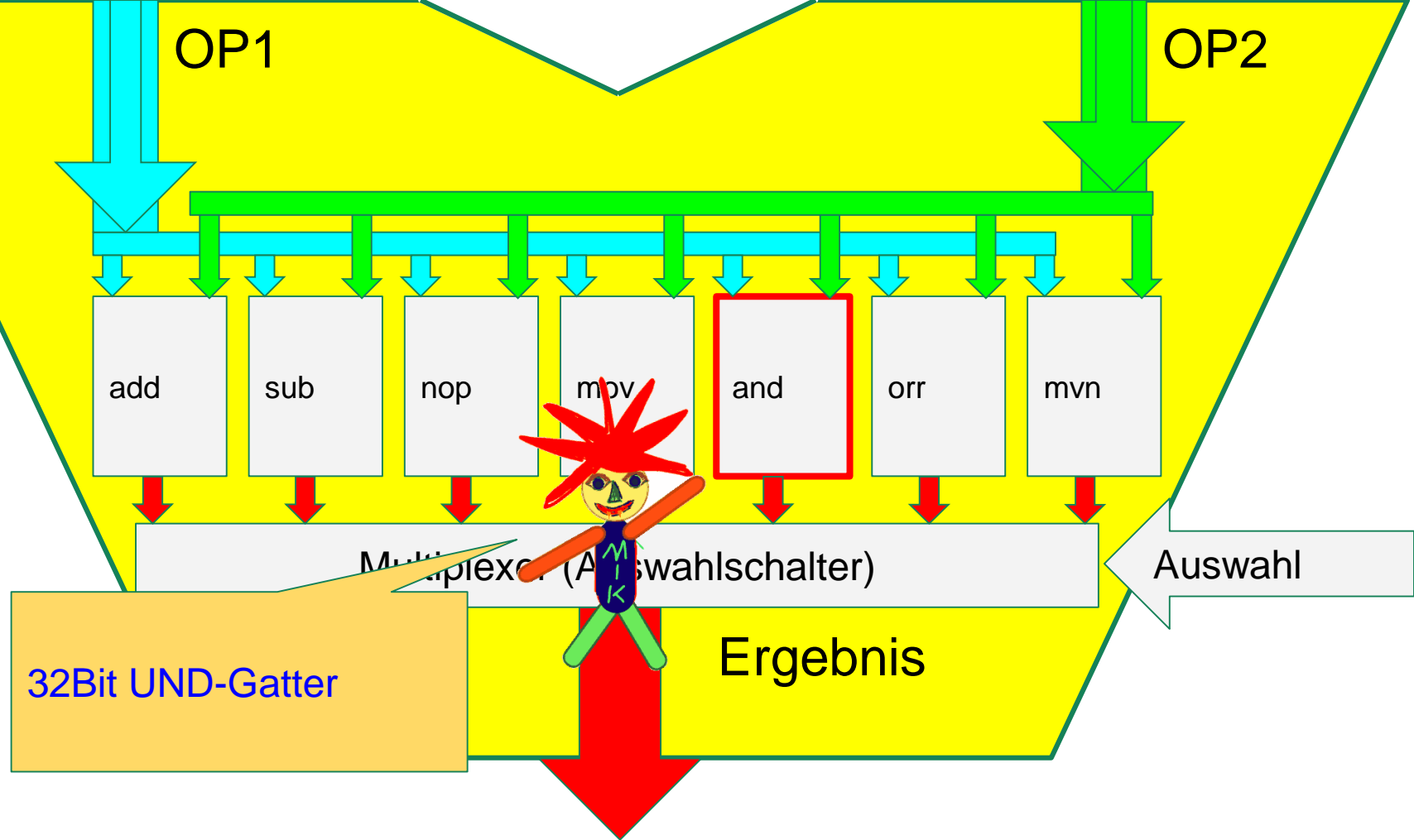
Übliche Darstellung,  
aber was ist drin?



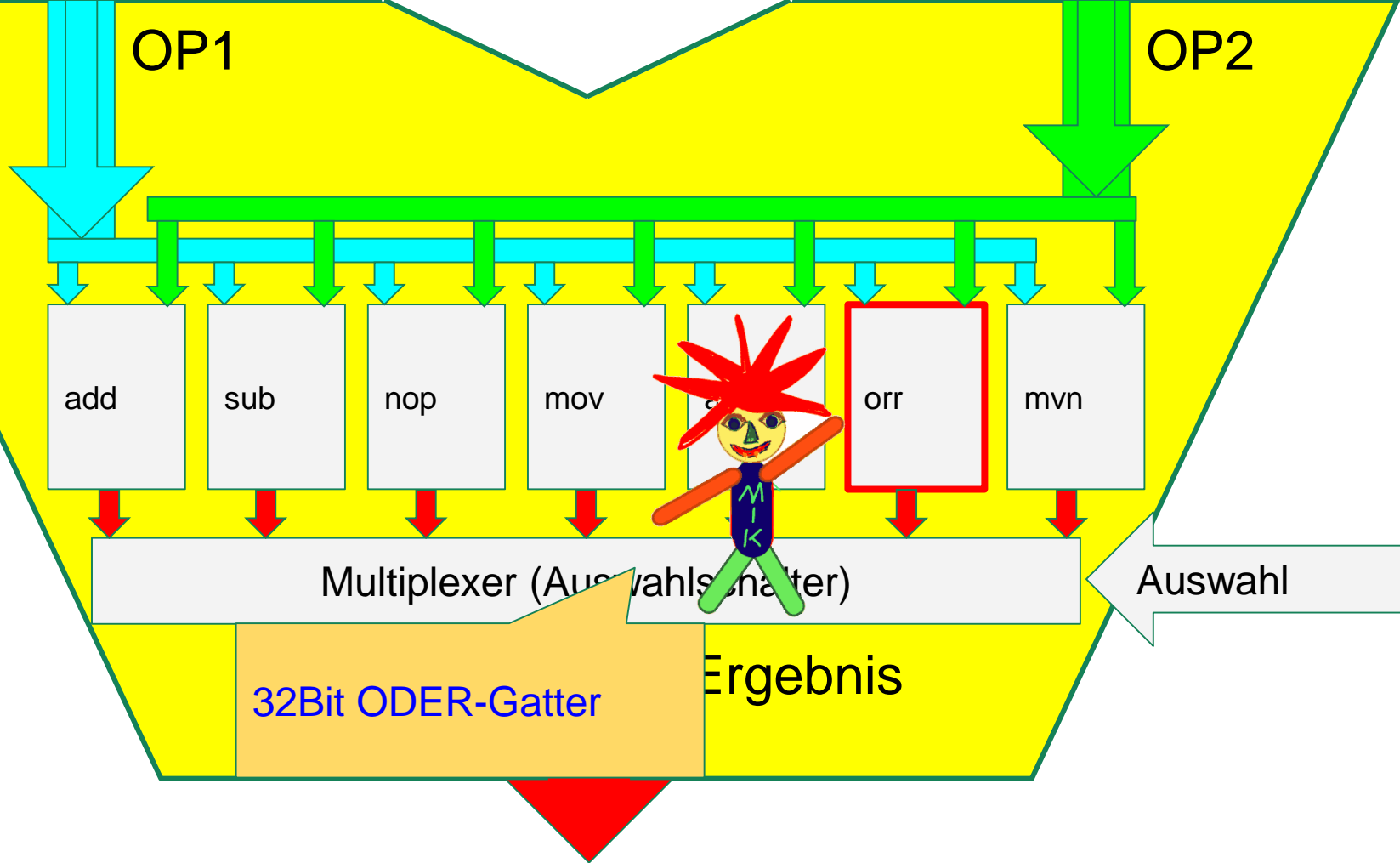


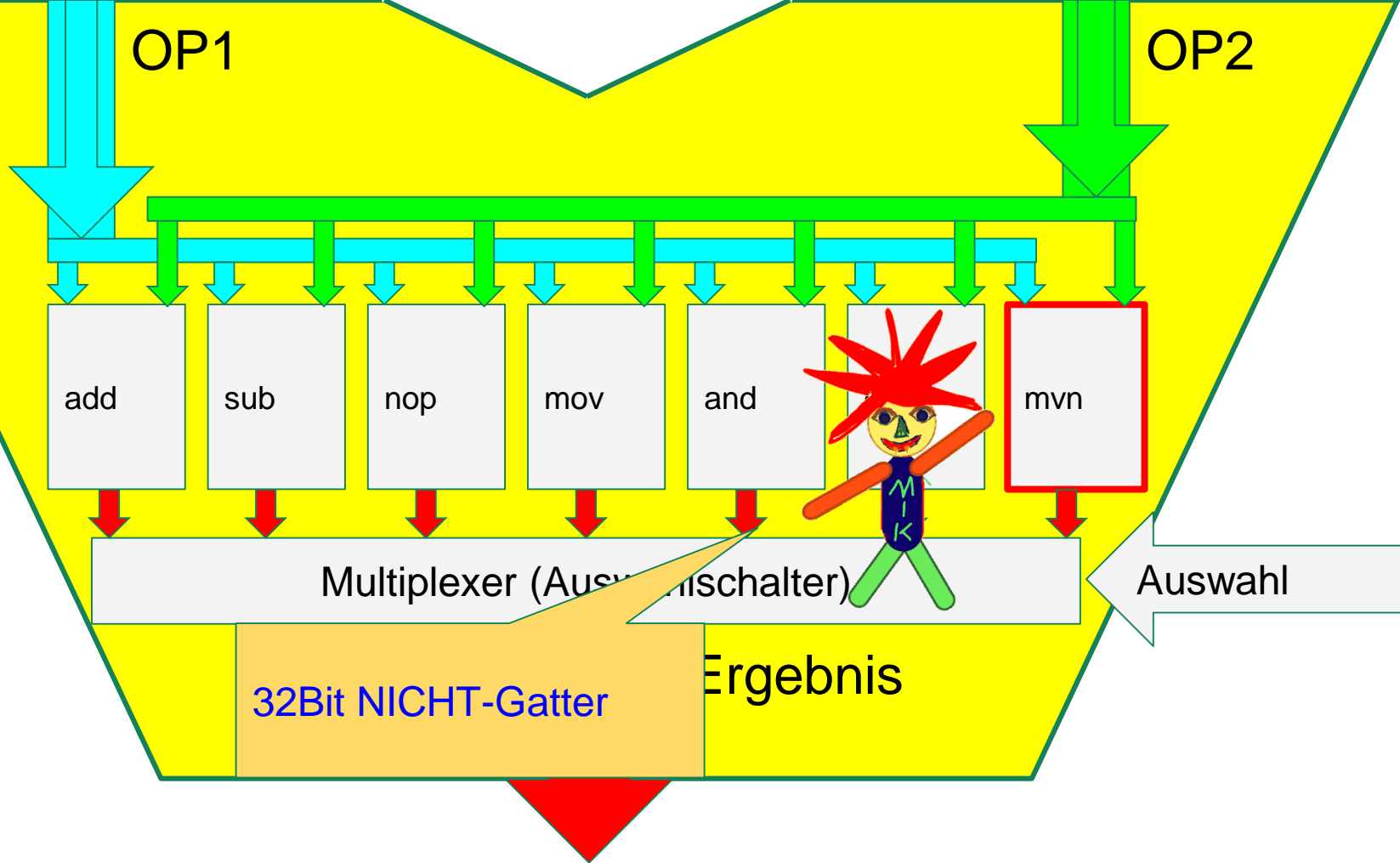








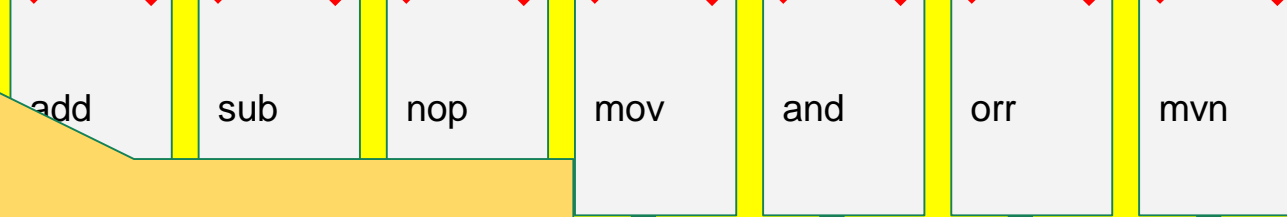
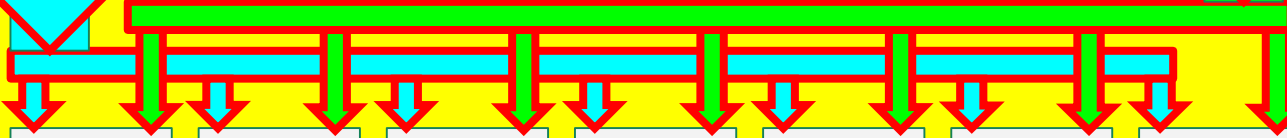




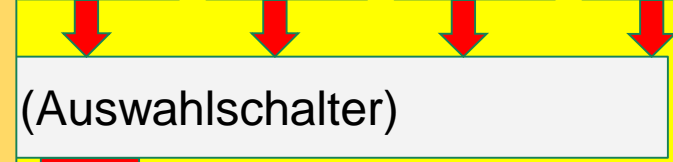


OP1

OP2

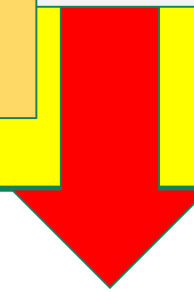


Beide Operanten werden  
gleichzeitig mit 7  
Rechenschaltnetzen verbunden



Auswahl

Ergebnis



OP1

OP2

Mit einem Multiplexer wird, je nach Anweisung, das passende Rechenergebnis ausgewählt

mov

and

orr

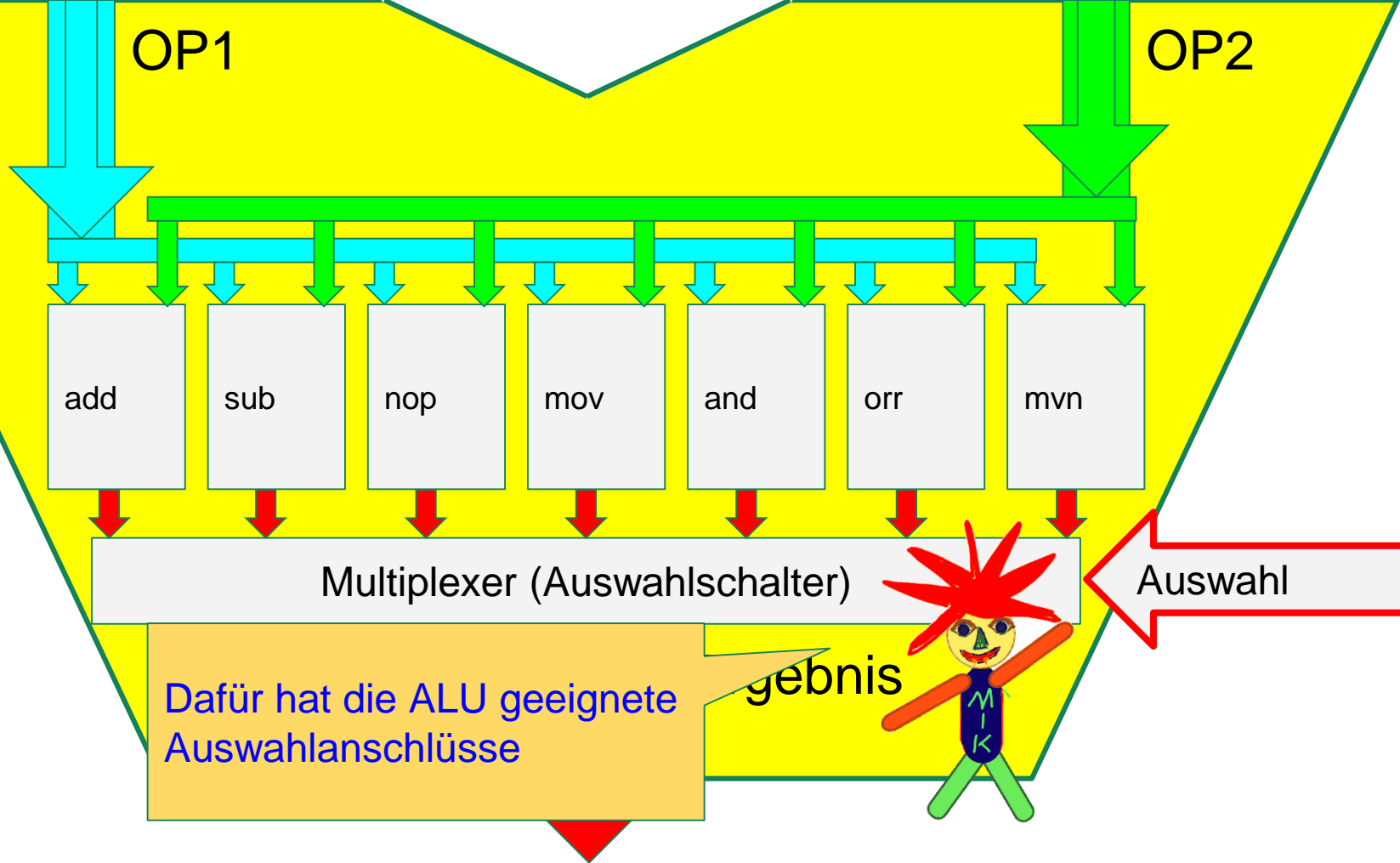
mvn

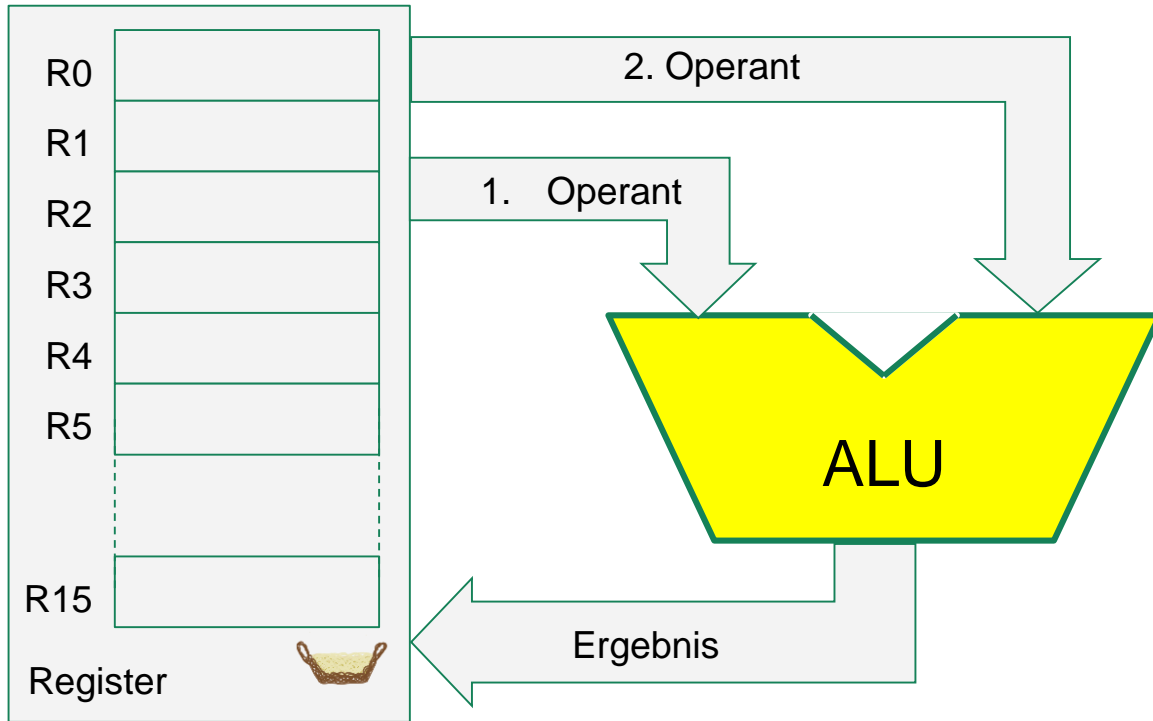
Multiplexer (Auswahlschalter)

Auswahl

Ergebnis

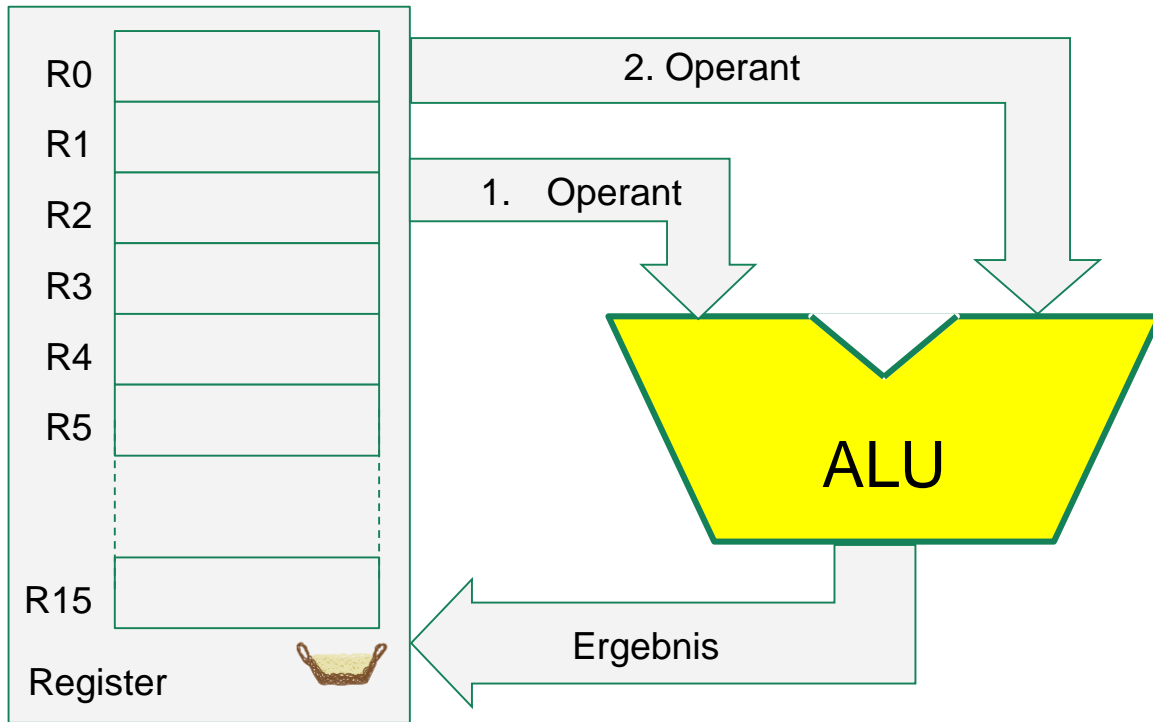






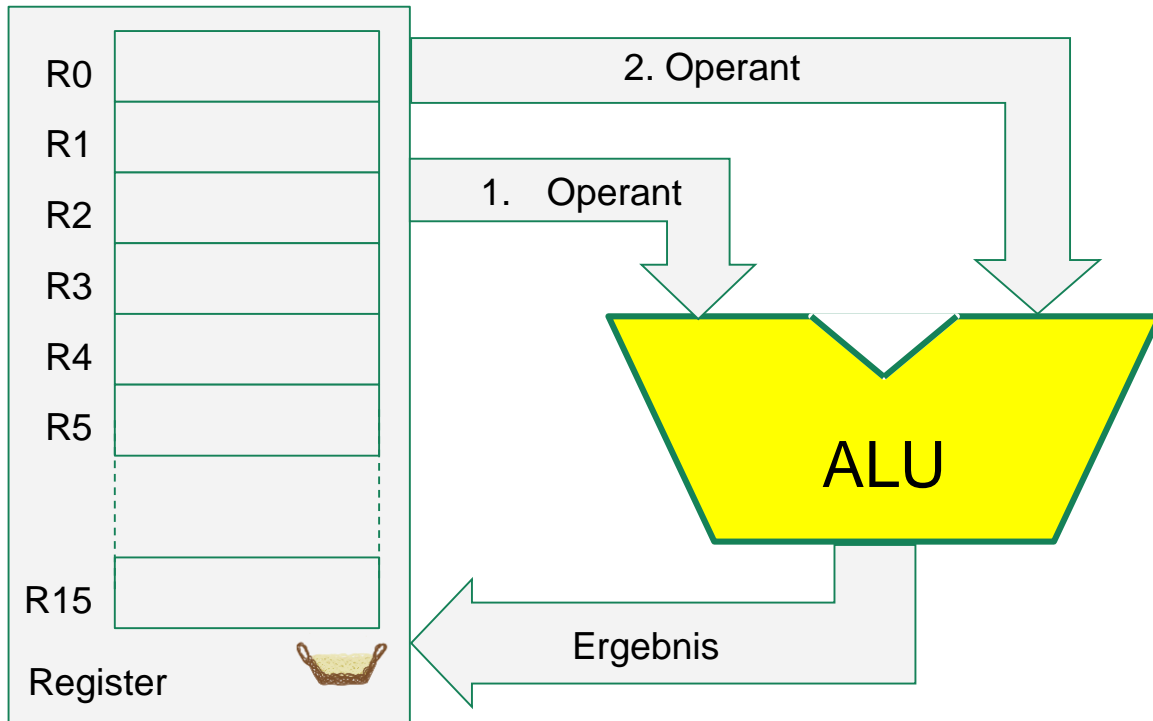
Die ALU berechnet also aus 2 Operanten ein Ergebnis.





Die Operanten können  
Register oder Zahlen sein

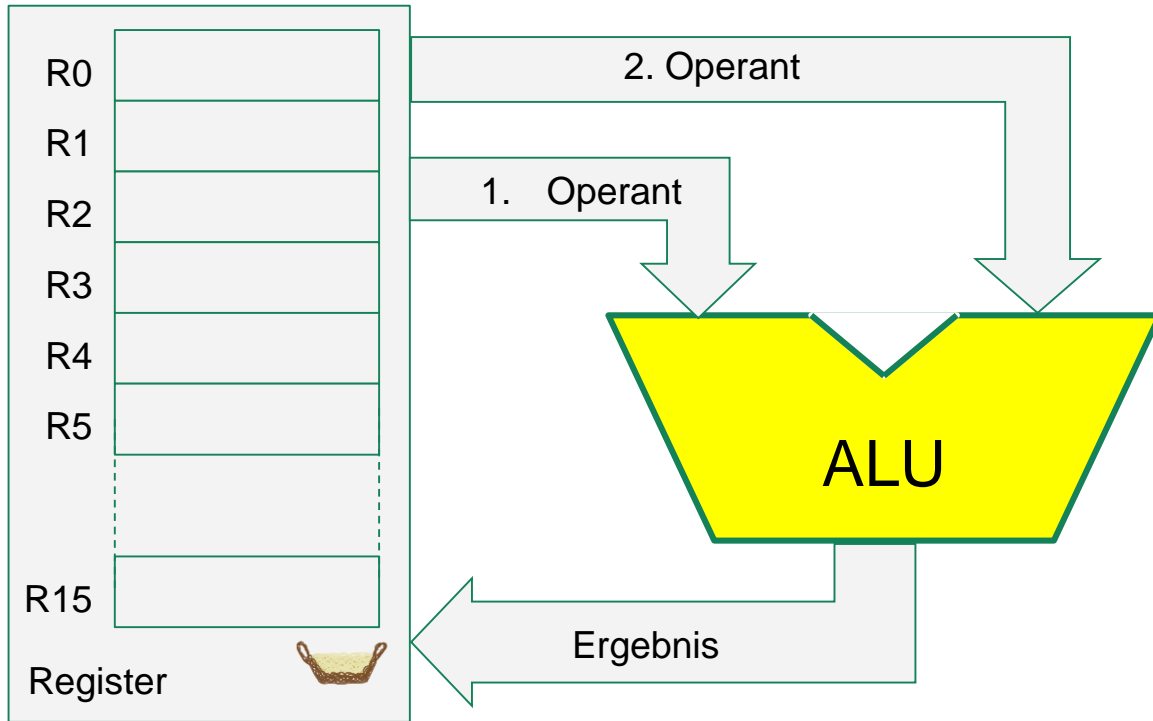




Das Ergebnis kann in einem Register abgelegt werden.



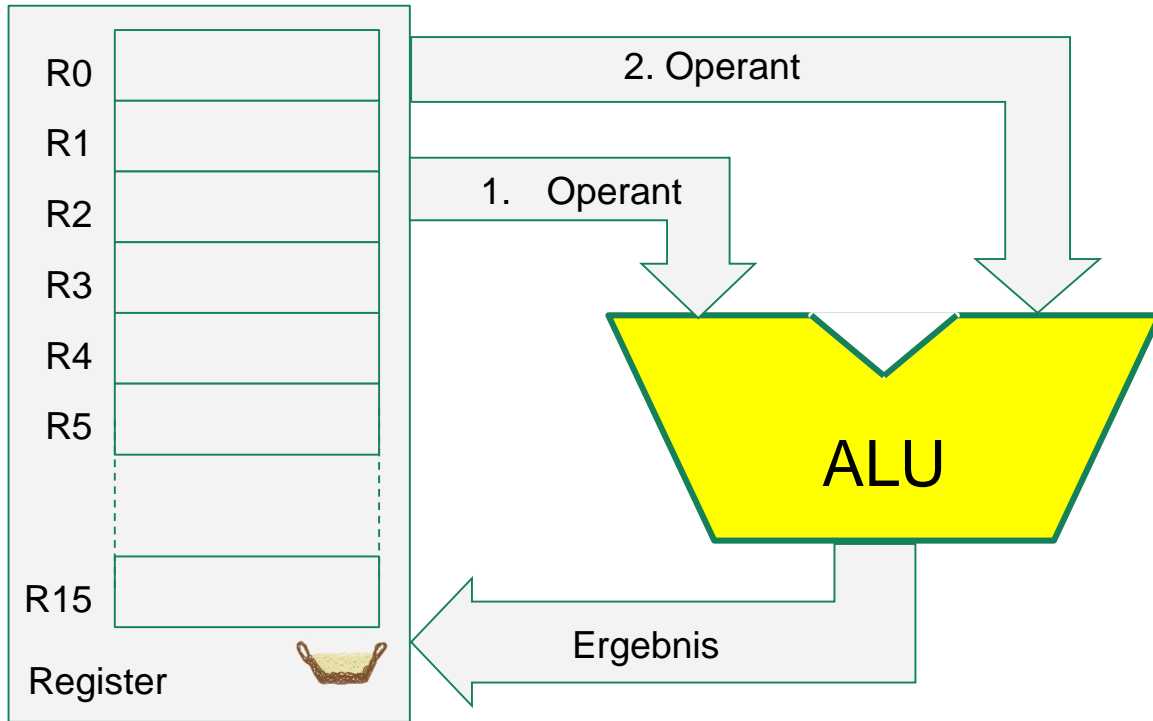




Z. B. Die  
Grundrechenarten

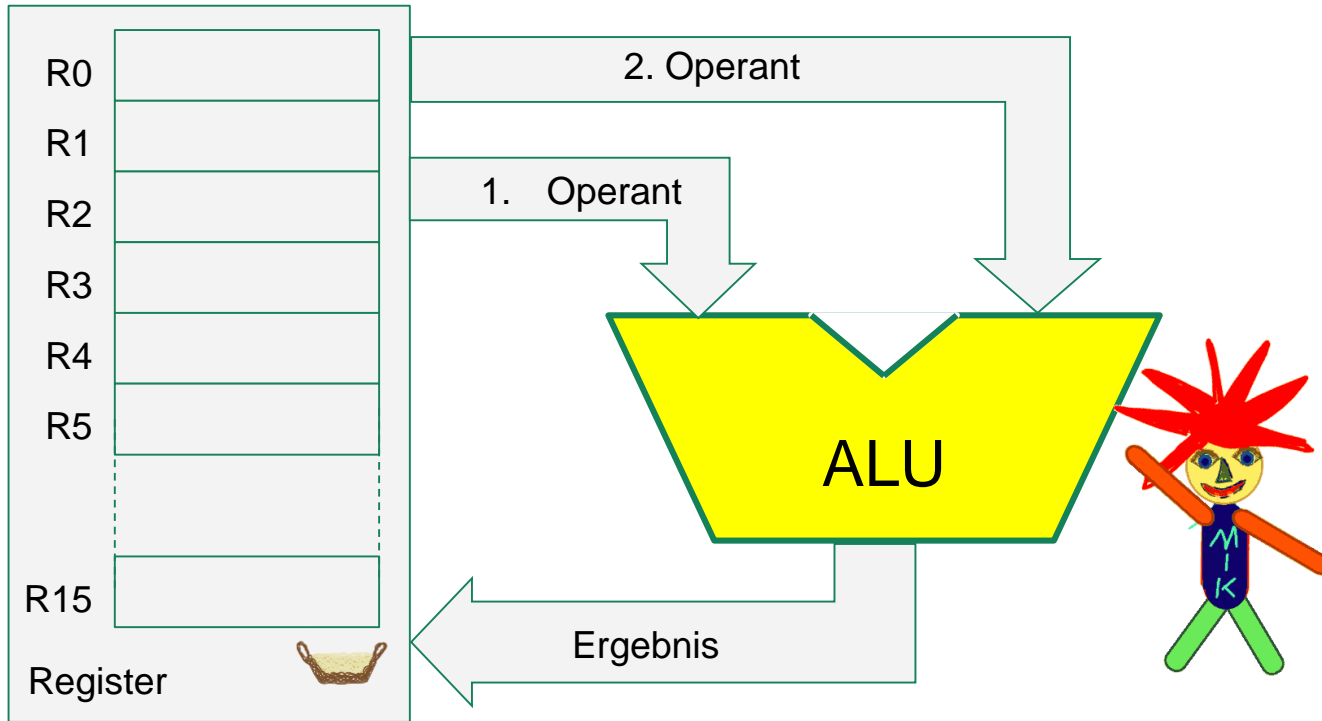
+ add  
- sub



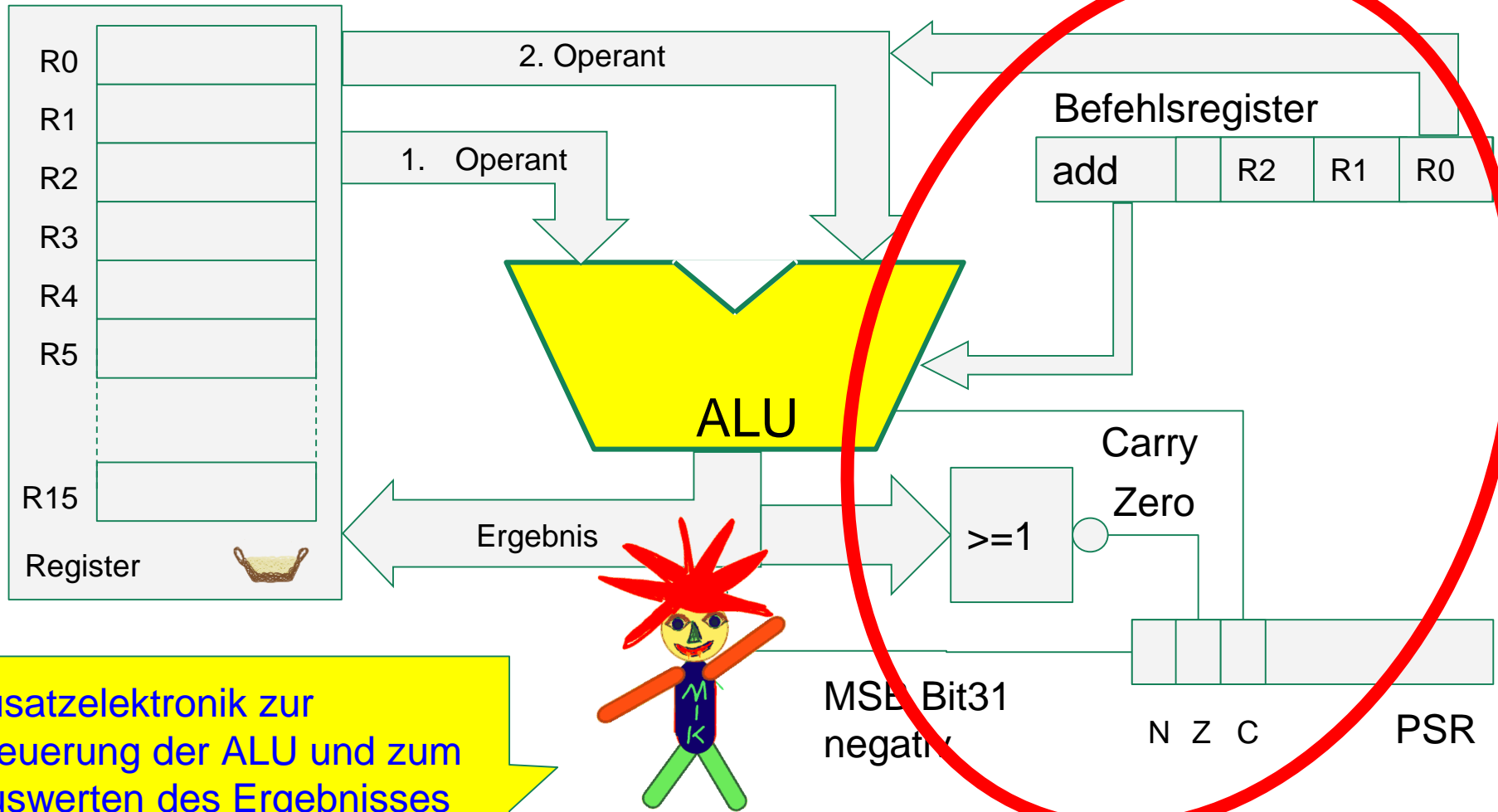


Z. B. Logische Operationen  
! mvn (mov not)  
& and  
| orr

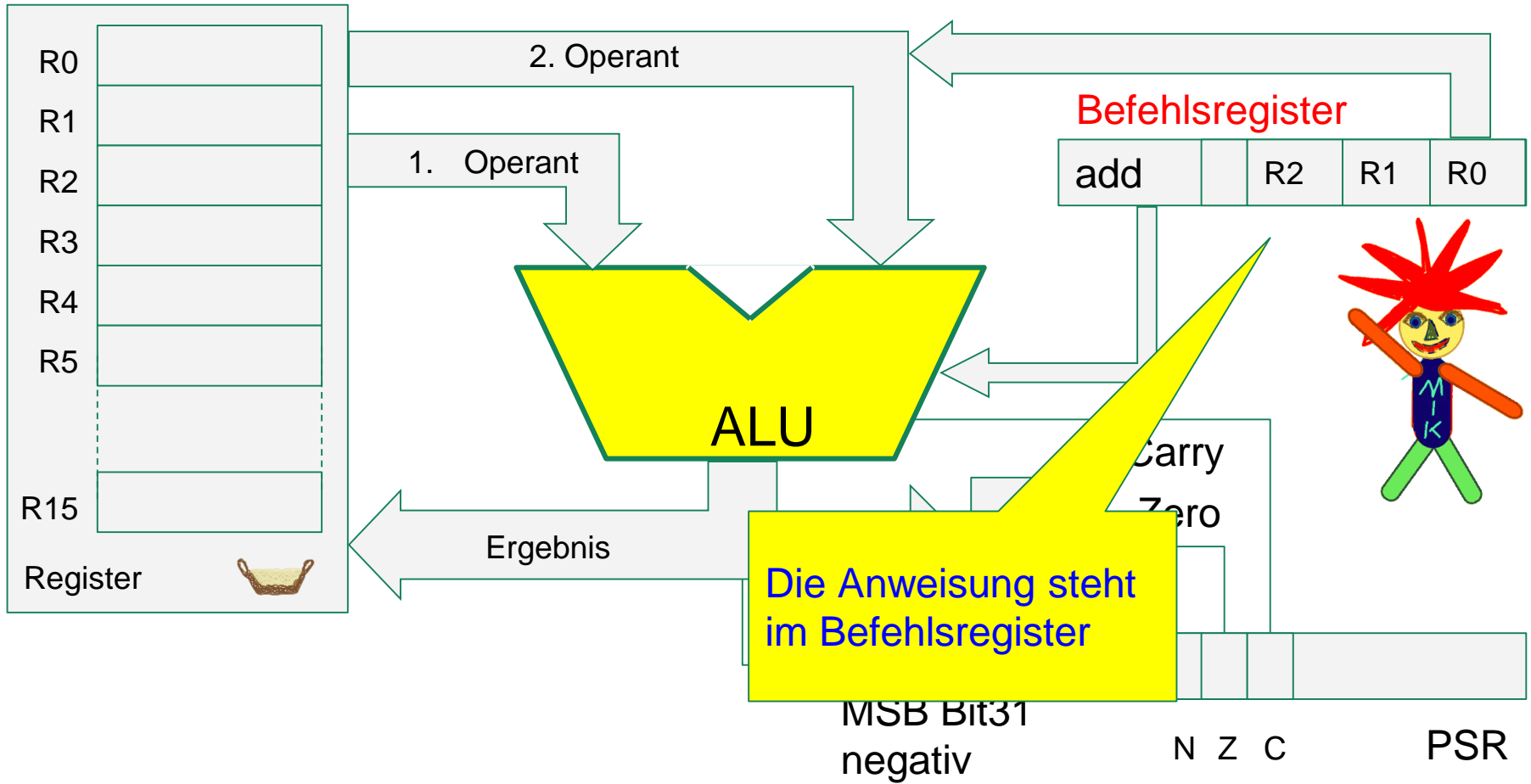


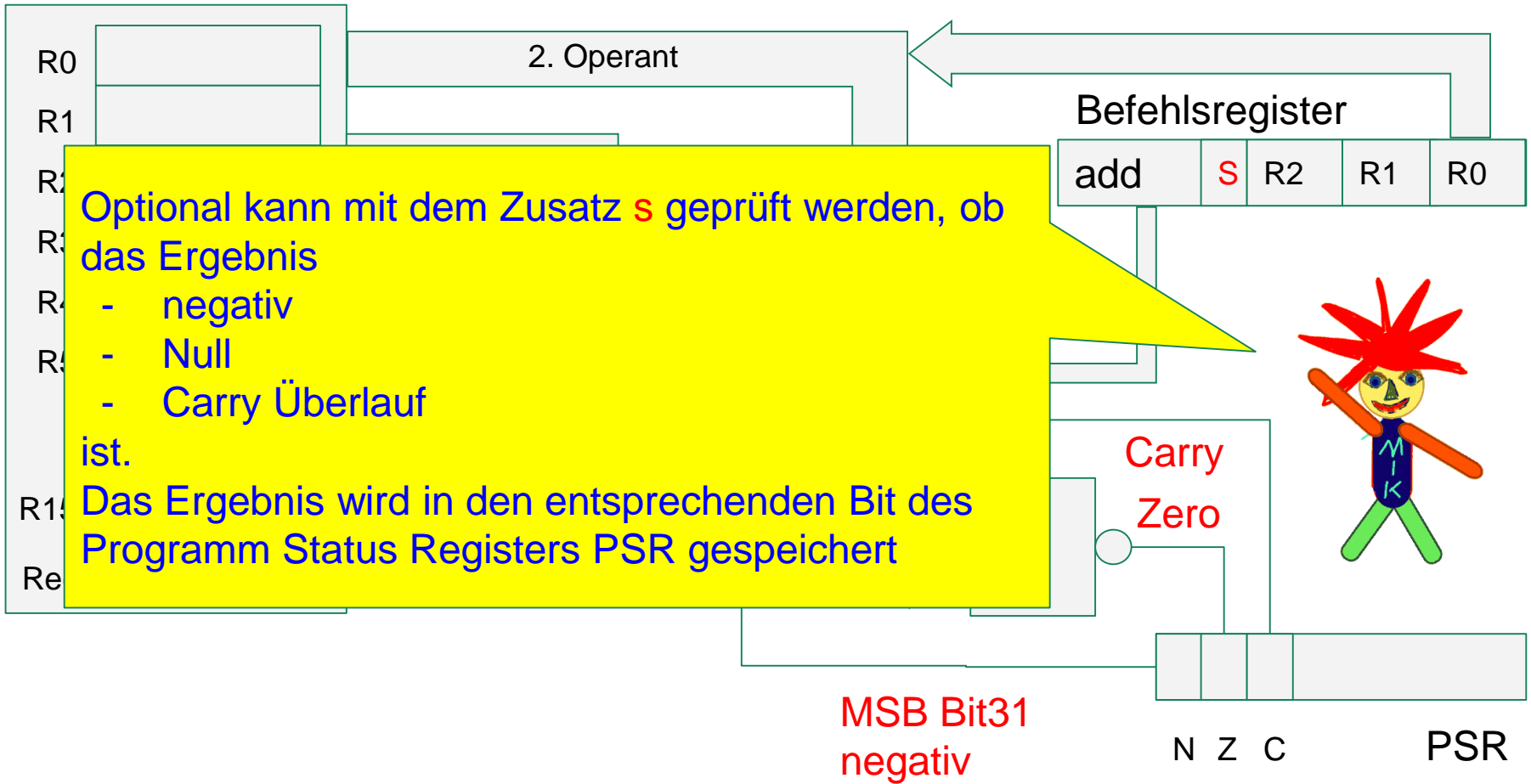


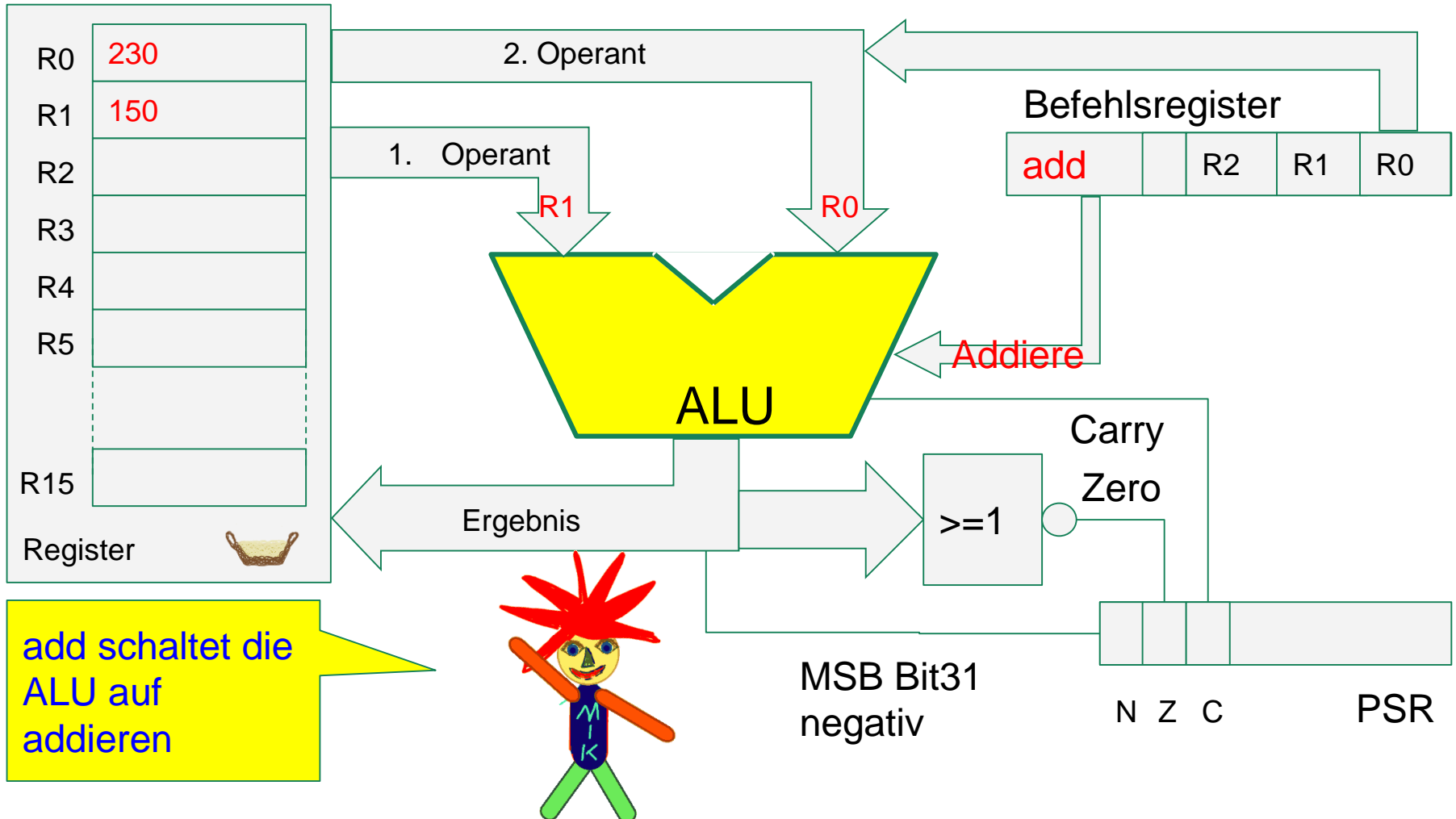
Ergebnis := Operant1  $\times$  Operant2  
Die ALU verknüpft 2 Operanten

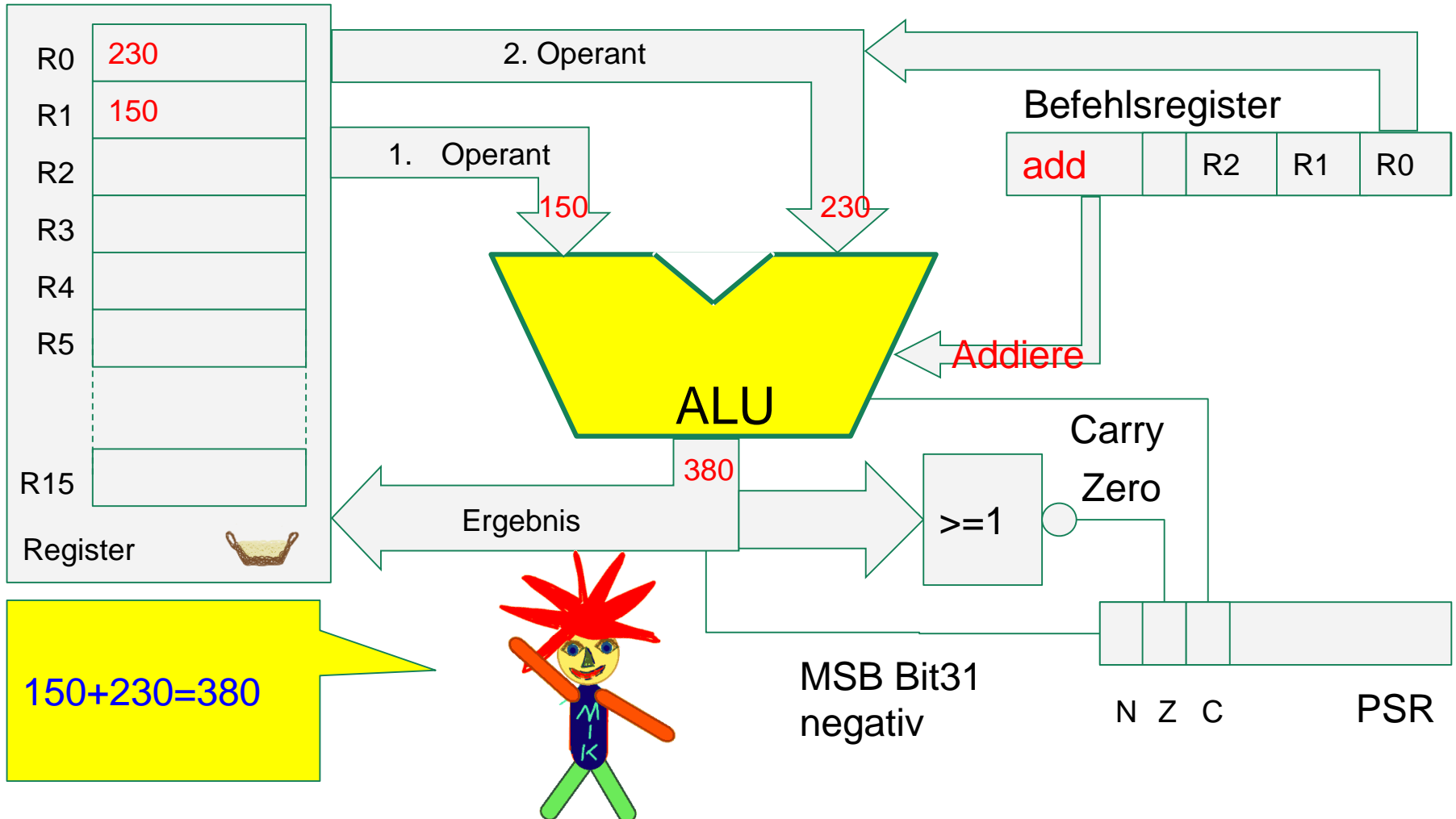


Zusatzelektronik zur  
Steuerung der ALU und zum  
Auswerten des Ergebnisses

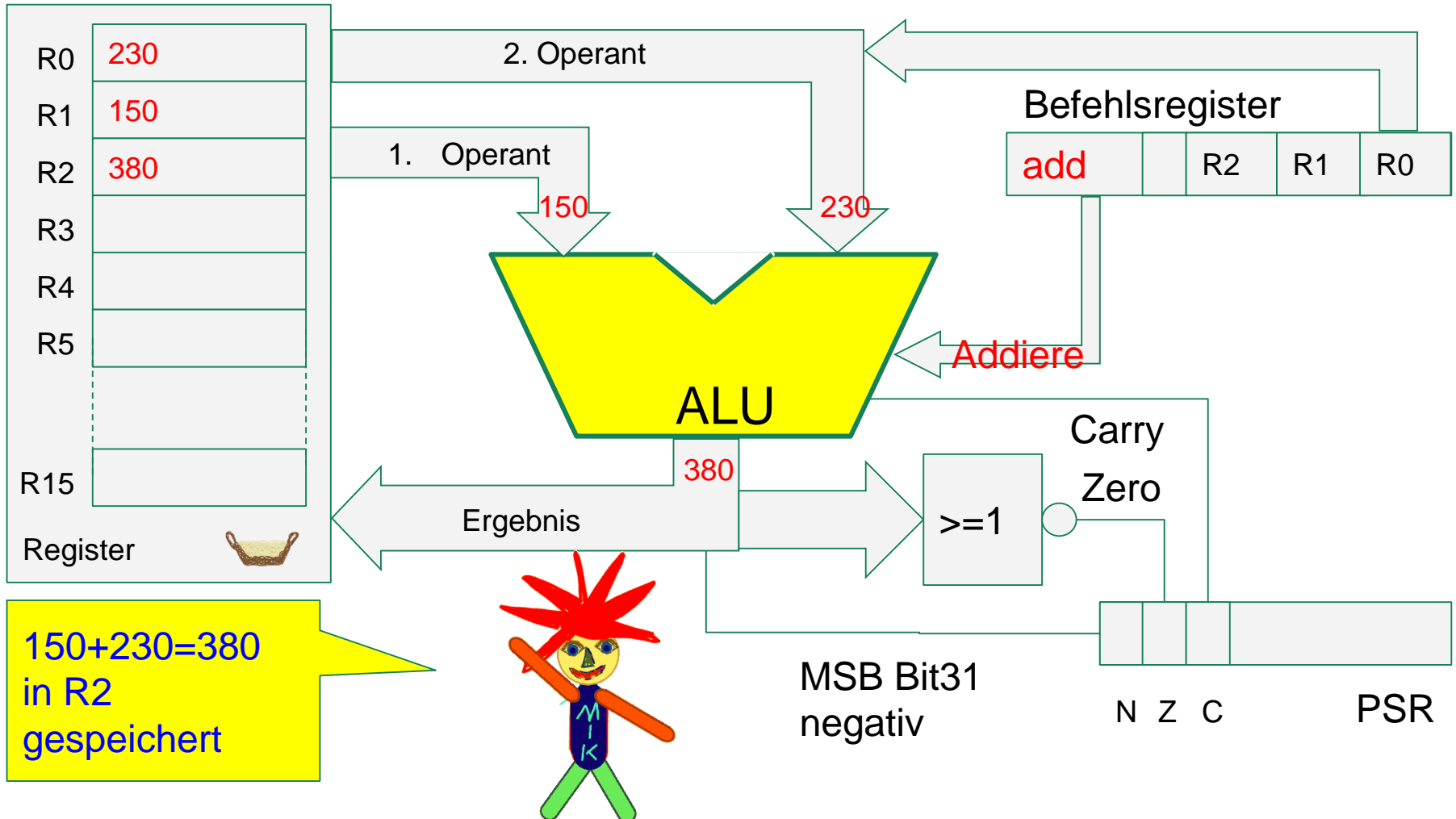


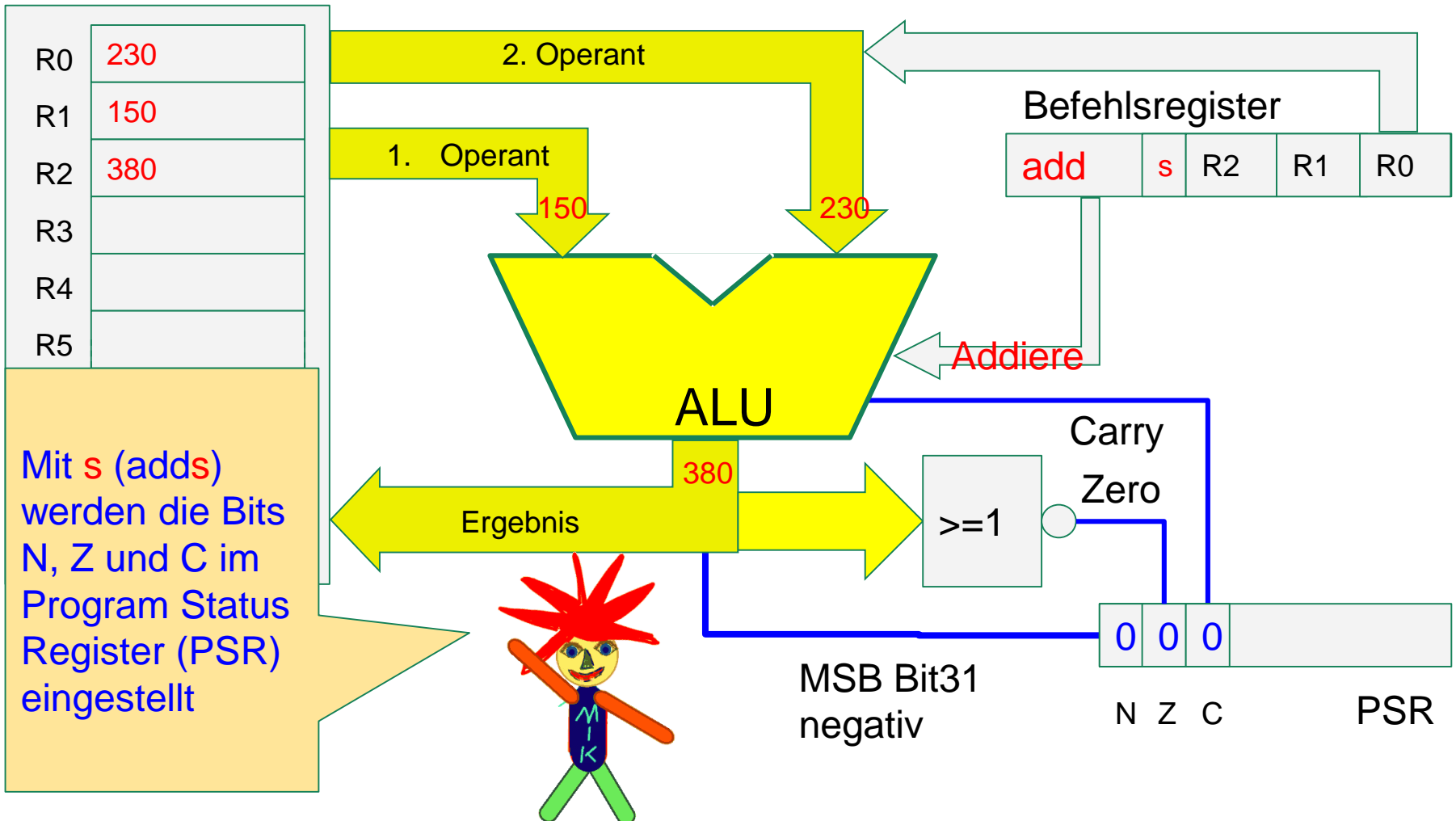


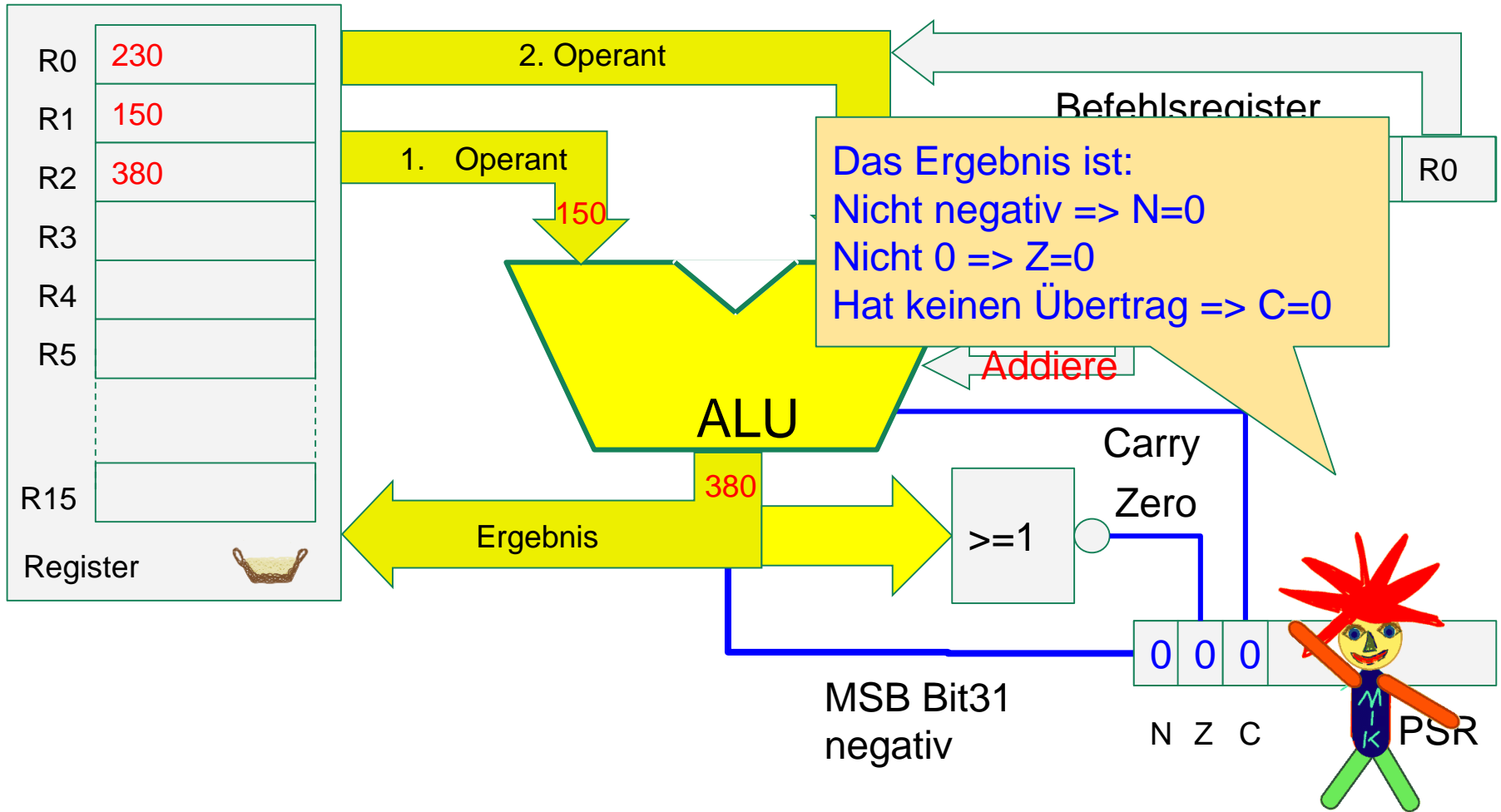


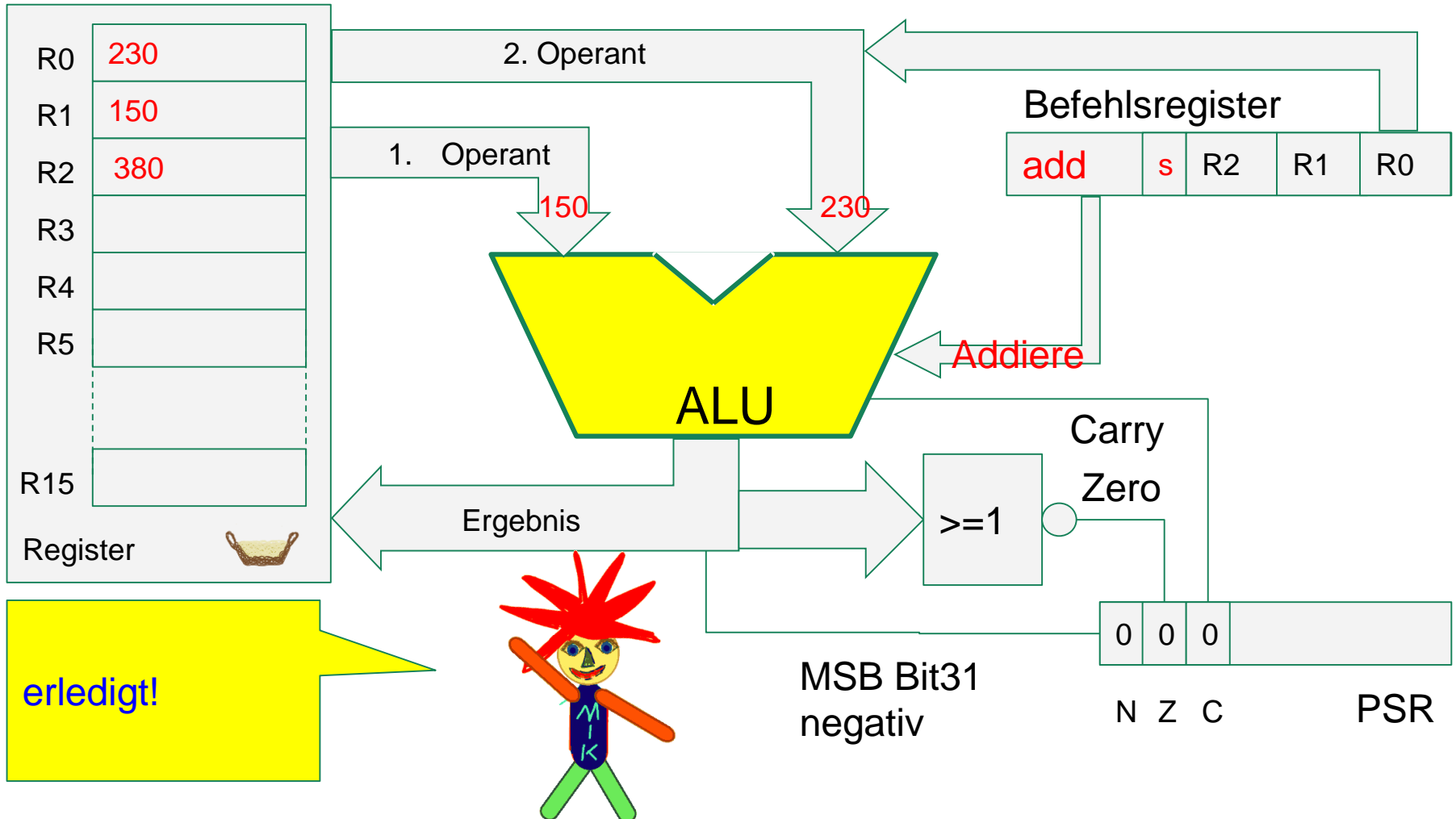


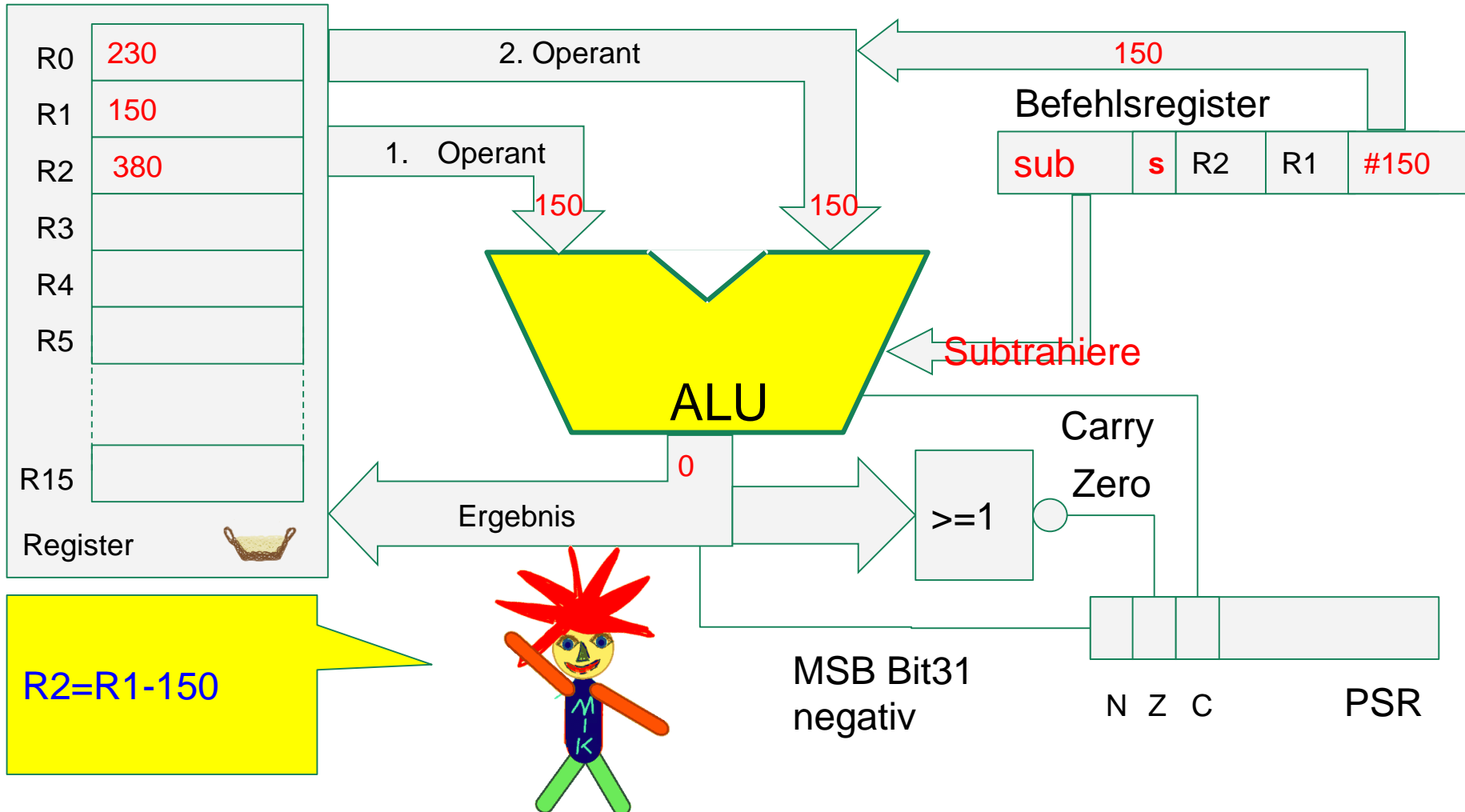


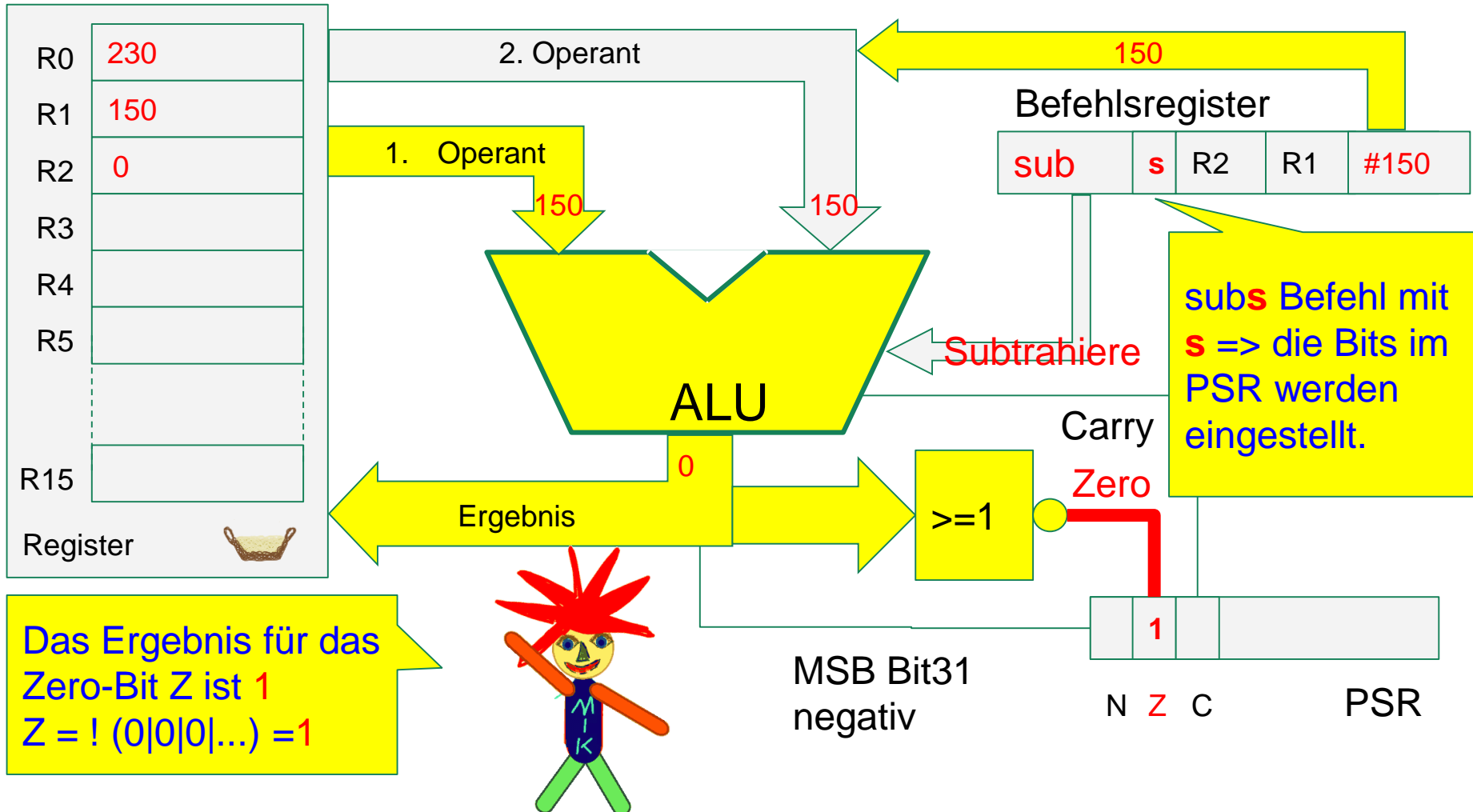


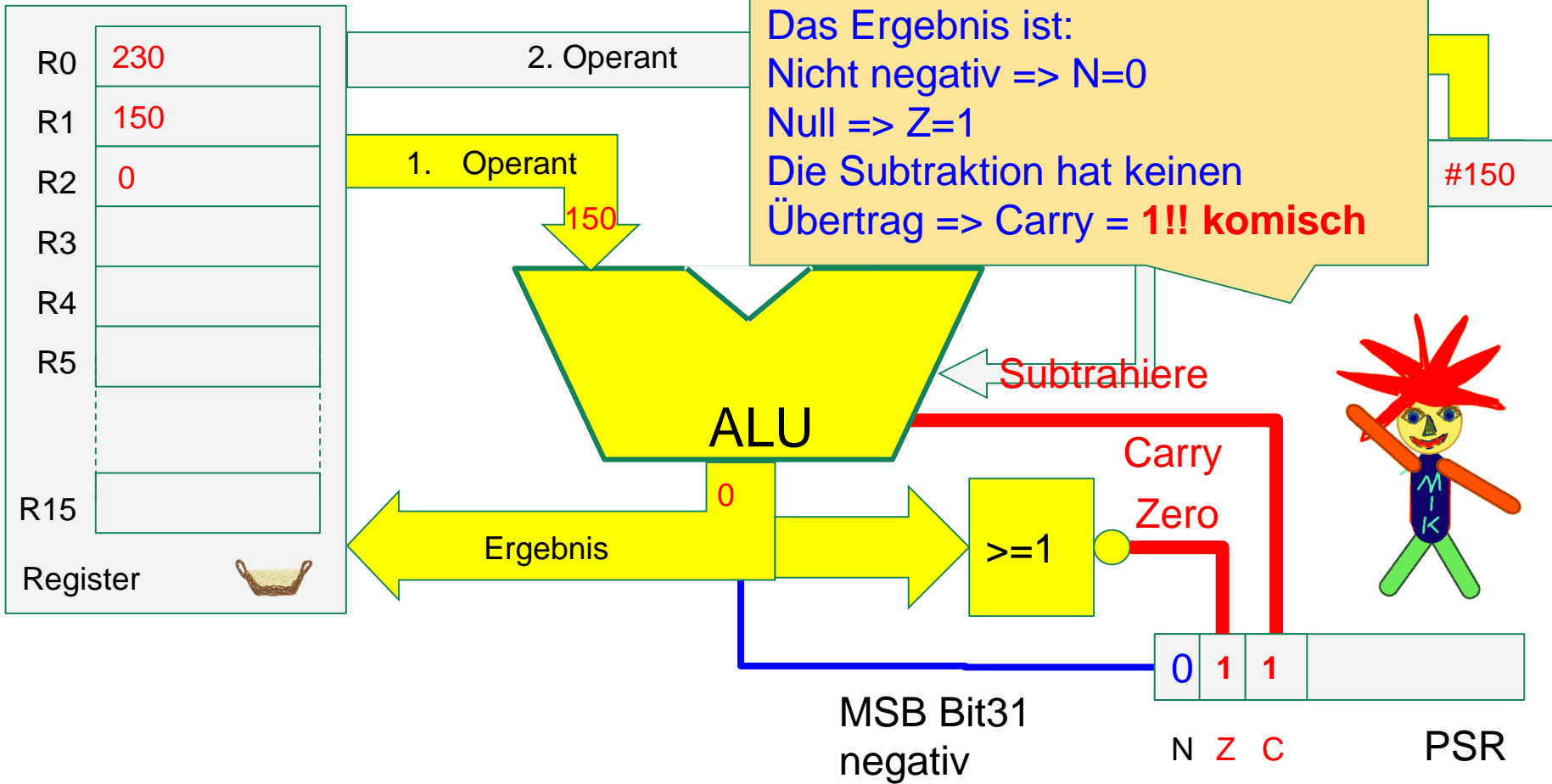


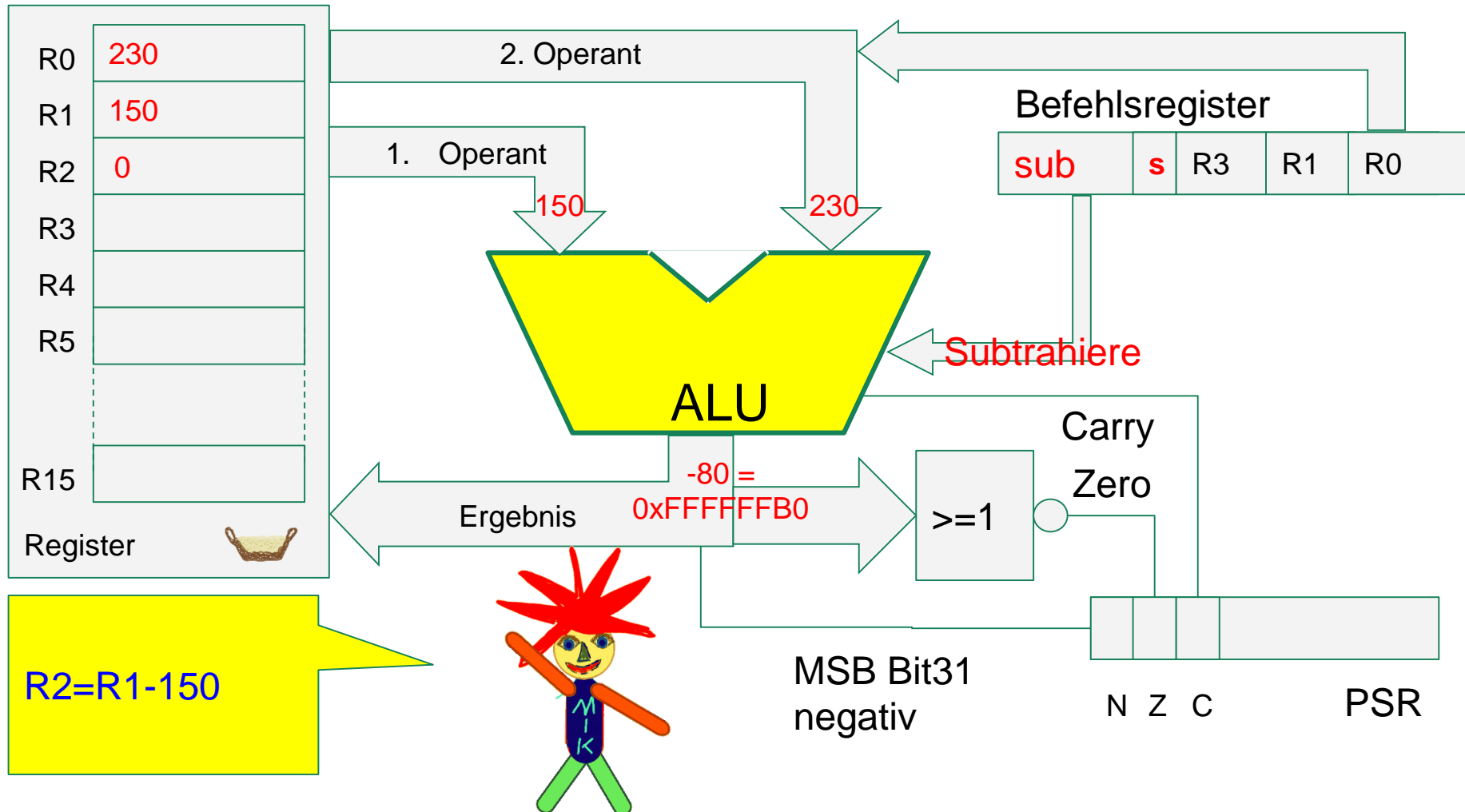




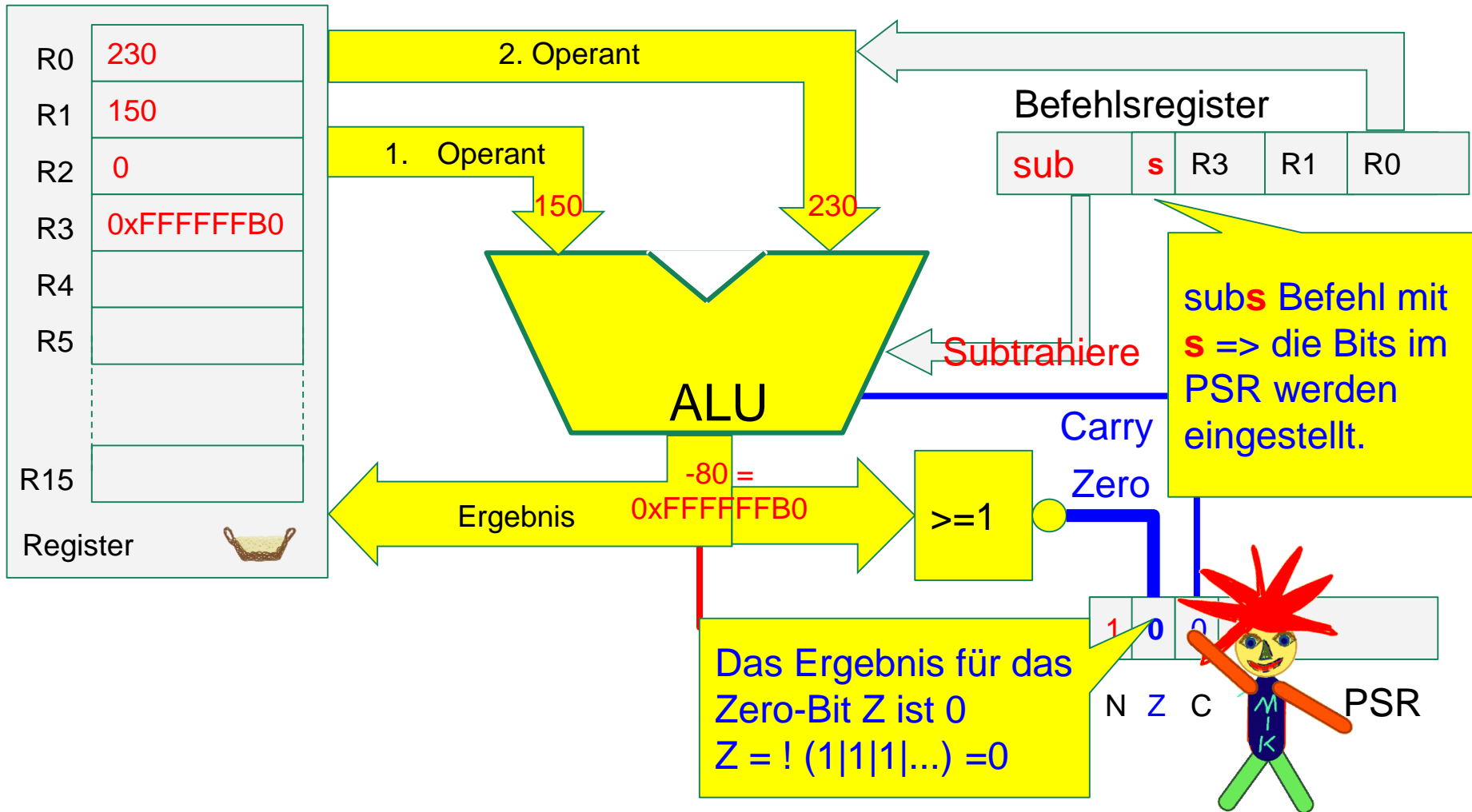


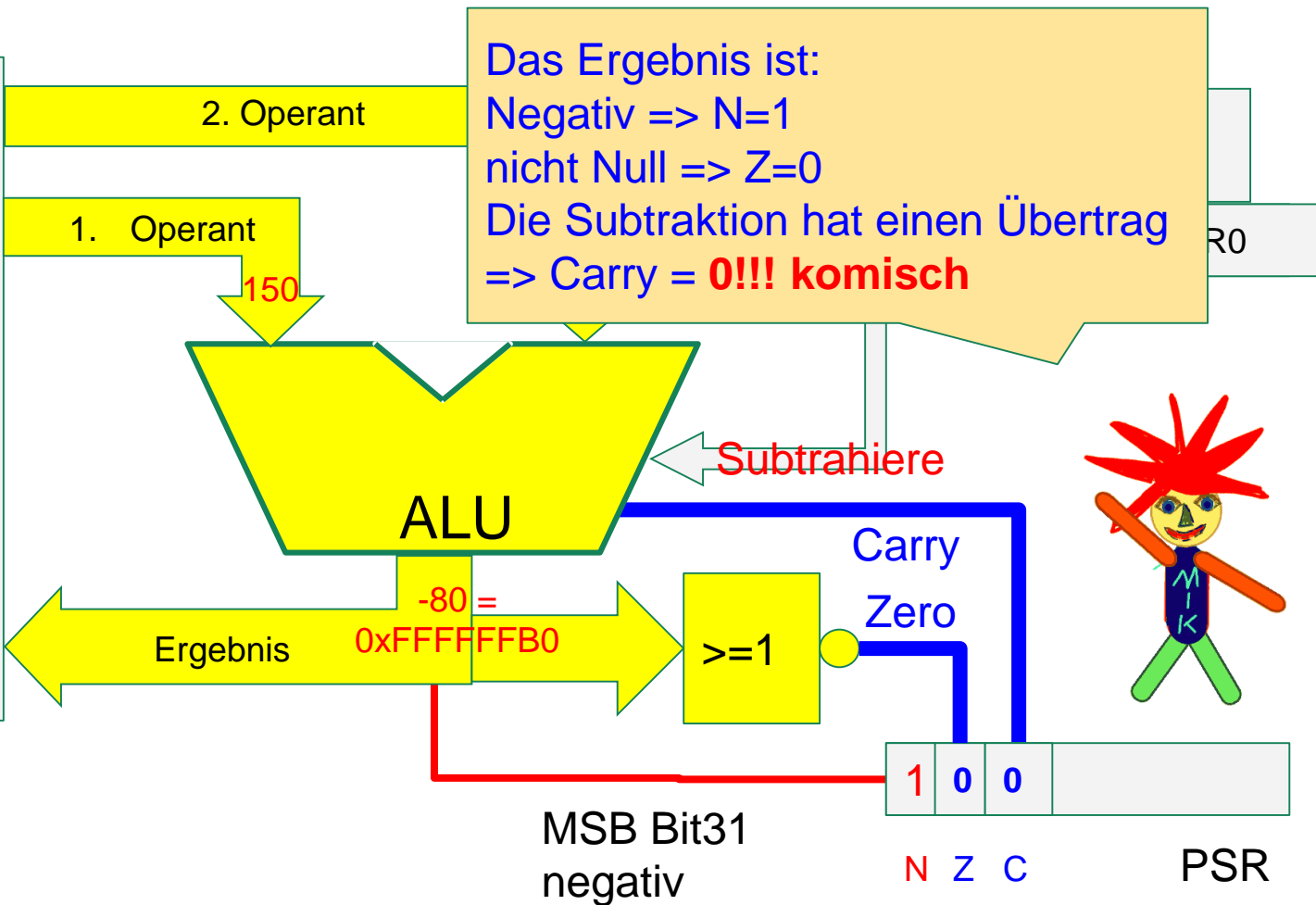
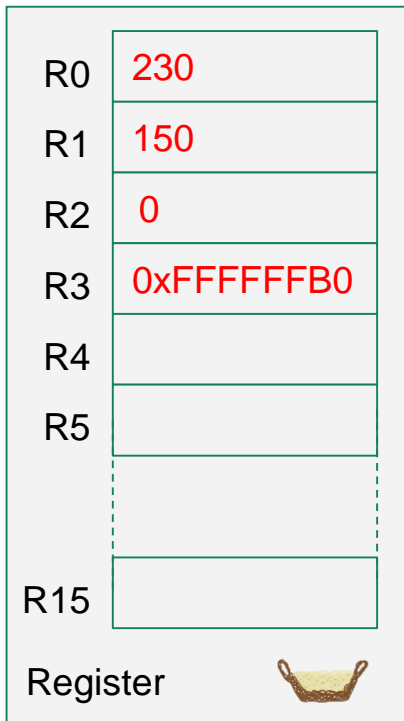


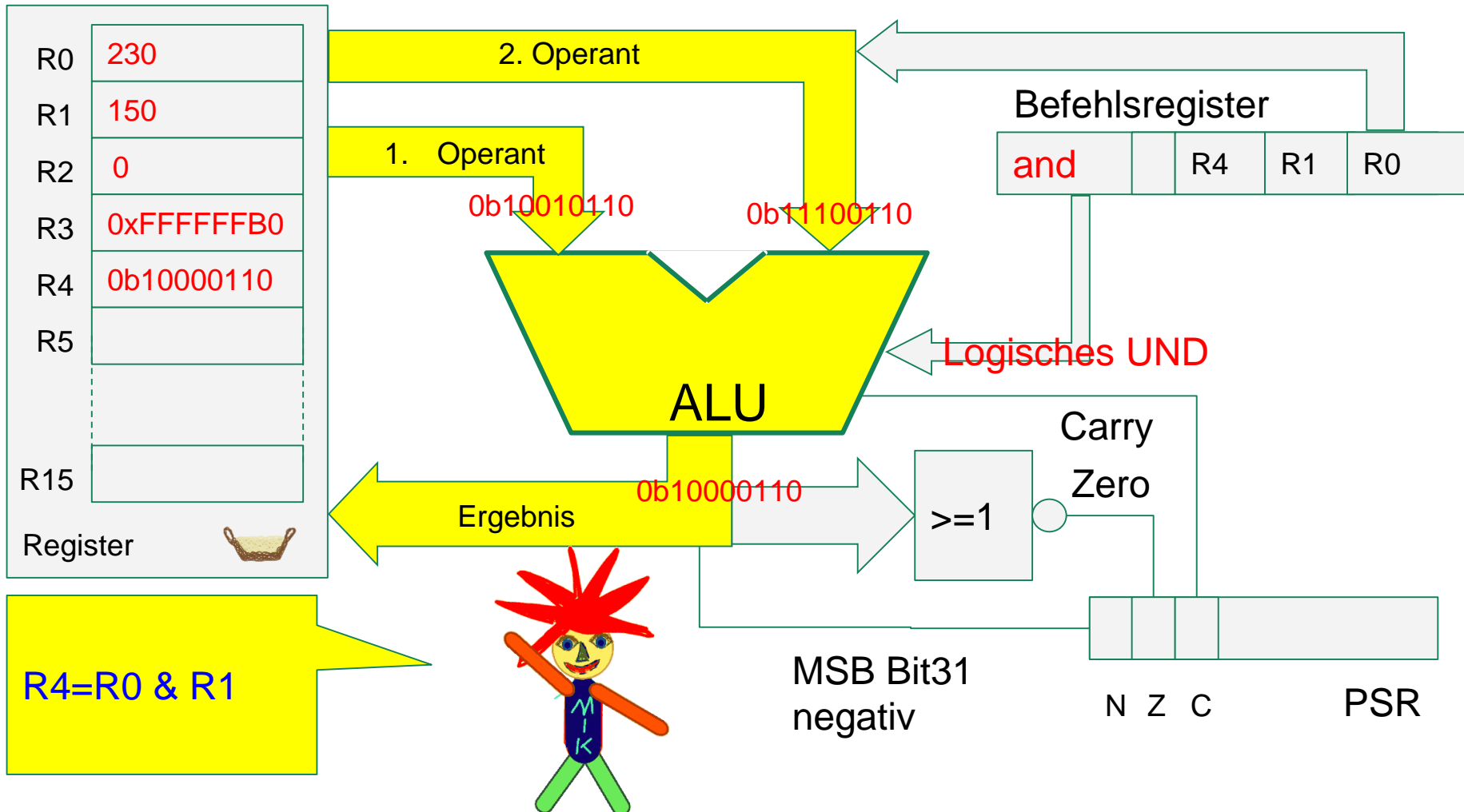


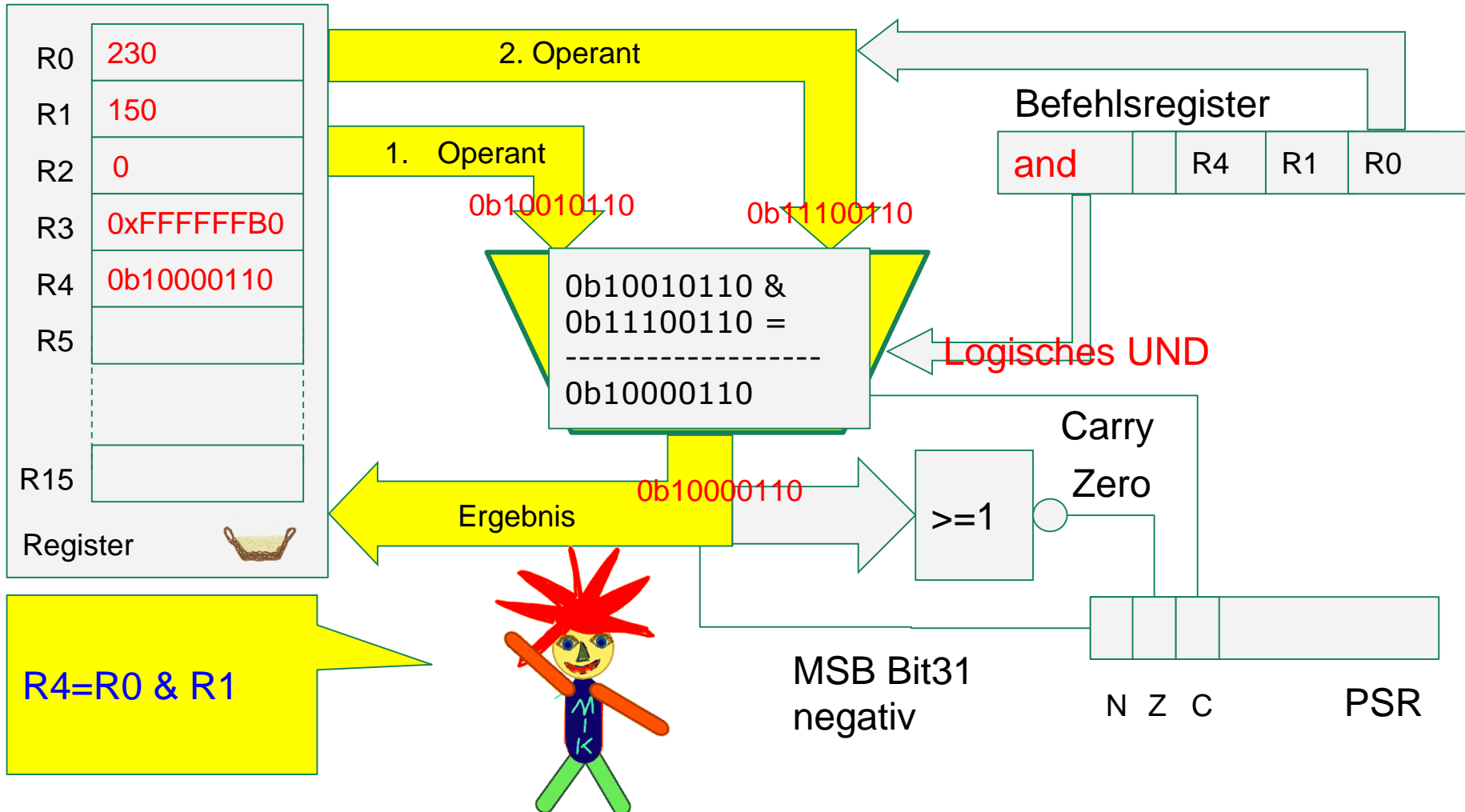












Das Ergebnis ist:  
nicht Negativ => N=0  
nicht Null => Z=0  
Das UND hat keinen Übertrag =>  
Carry = 0

