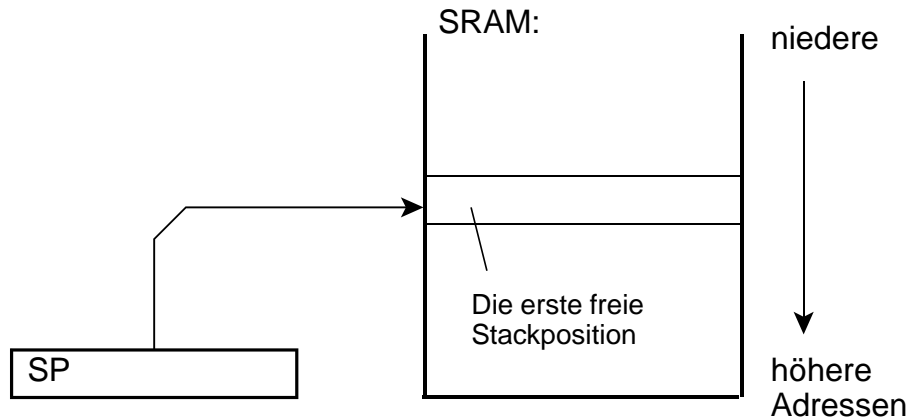
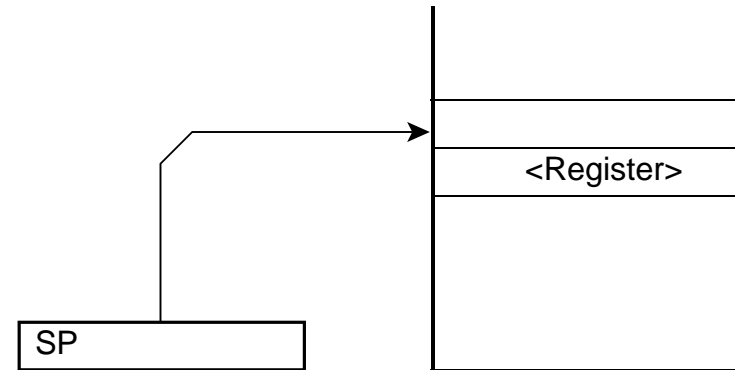


Anfangszustand:

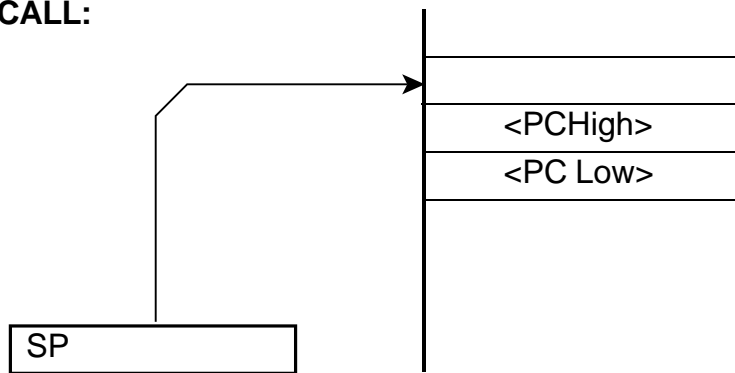


Anfangszustand:

PUSH Register:



CALL:

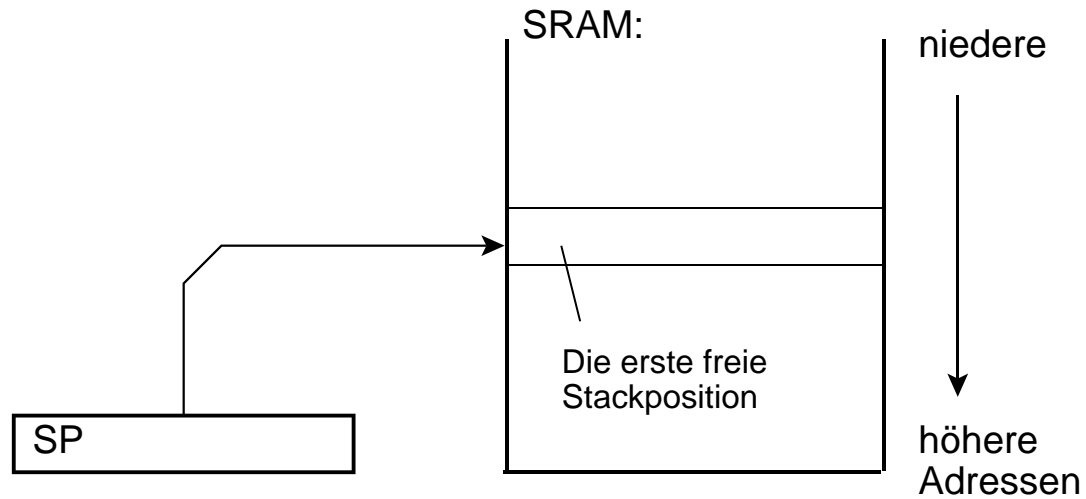


Achtung: Das höherwertige Adreßbyte des Befehlszählers befindet sich auf der niederen Adresse (obwohl man naiverweise erwartet, daß sich der AVR als Little-Endian-Maschine verhält...).

## Der Stack des Atmel AVR (1) Übersicht

12. 8. 2013

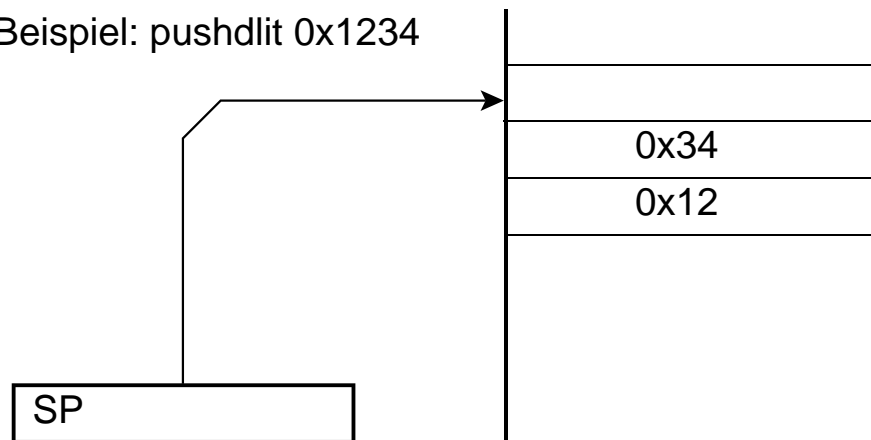
**Anfangszustand:**



**Anfangszustand:**

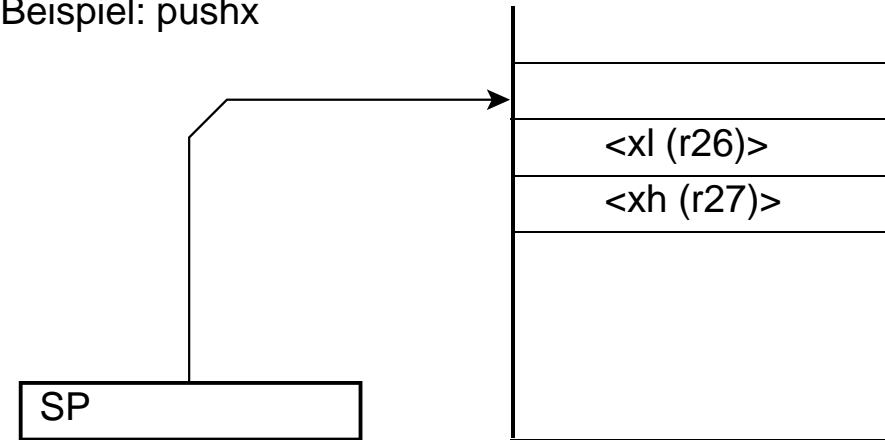
**PUSH Wort-Direktwert  
(Literal):**

Beispiel: pushdlit 0x1234



**PUSH Wortregister:**

Beispiel: pushx



Der Stack wird belegt wie bei einer  
Little-Endian-Maschine üblich  
(höwertige Bytes auf  
höherwertigen Adressen).

**Der Stack des Atmel AVR (2)  
Elementare Makros**

12. 8. 2013

Eine altbewährte Programmiertechnik: Die Parameter sollen vor dem Unterprogrammruf auf den Stack gelegt werden.

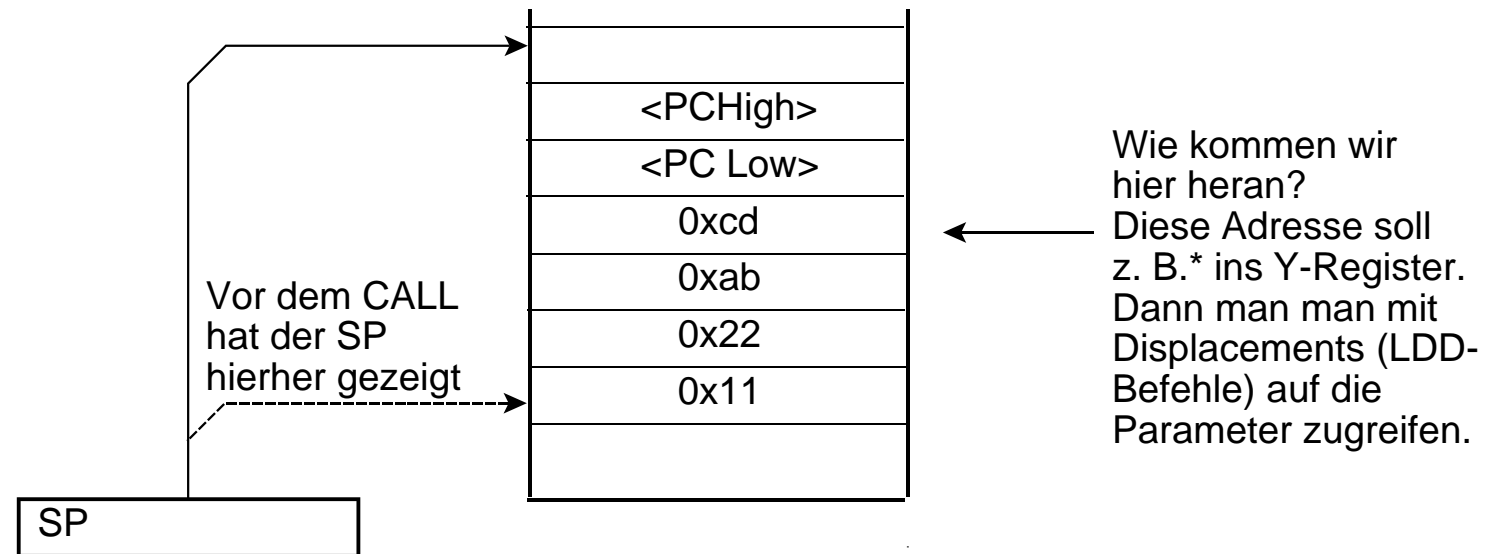
Beispiel einer Programmierabsicht:

```
pushlit 0x11  
pushlit 0x22  
pushdlit 0xabcd  
call subroutine
```

← Rückkehr hierher

Die Parameter müssen aus dem Stack raus.  
Das soll im Unterprogramm erledigt werden.

So sieht der Stack nach dem CALL aus:



\*: Es gibt eine Variante mit dem X-Register und eine mit dem Y-Register.

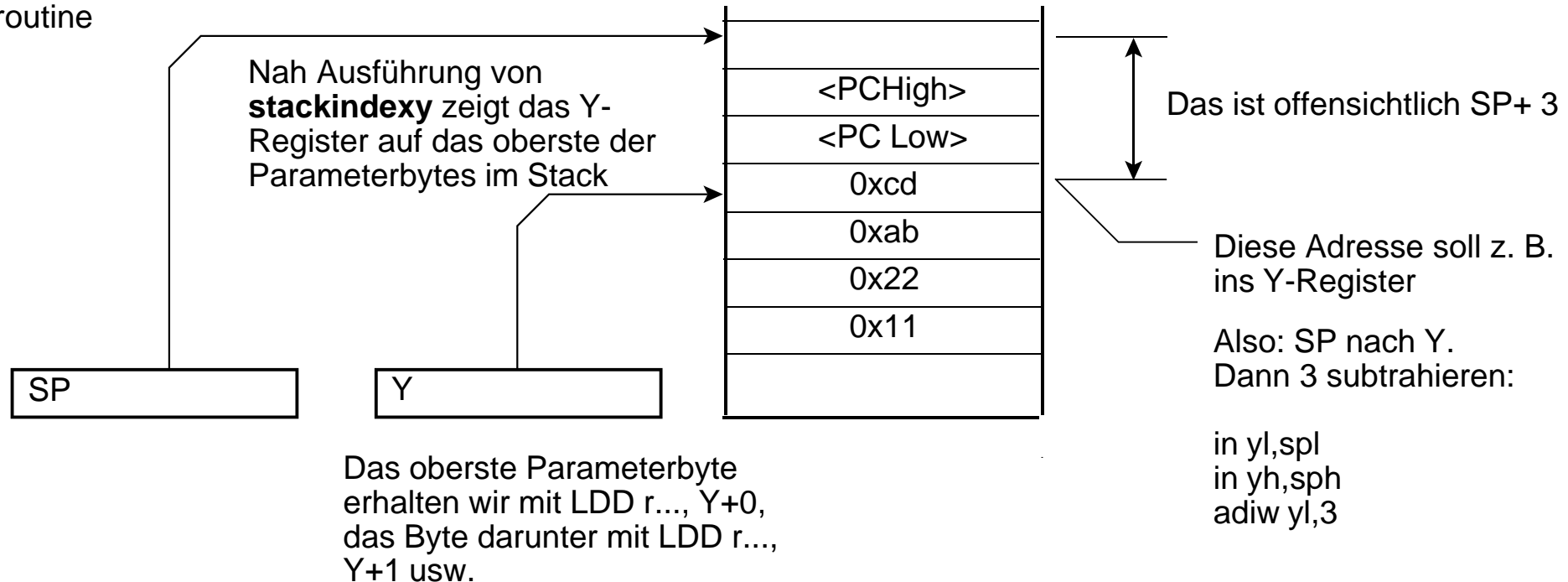
## Der Stack des Atmel AVR (3) Parameterübergabe im Stack

12. 8. 2013

Beispiel einer Programmierabsicht:

```
pushlit 0x11  
pushlit 0x22  
pushdlit 0xabcd  
call subroutine
```

Der Stack nach dem CALL:



**Der Stack des Atmel AVR (4)  
Parameterübergabe im Stack  
Die Makros stackindexx und  
stackindexy**

12. 8. 2013

Beispiel einer Programmierabsicht:

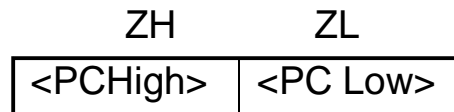
```
pushlit 0x11
pushlit 0x22
pushdlit 0xabcd
call subroutine
```

subroutine:  
**stackindexy**

... der auszuführende Code.....

**clrstackret 4** ; .. weil 4 Bytes als  
; Parameter übergeben wurden

```
pop zh
pop zl
```

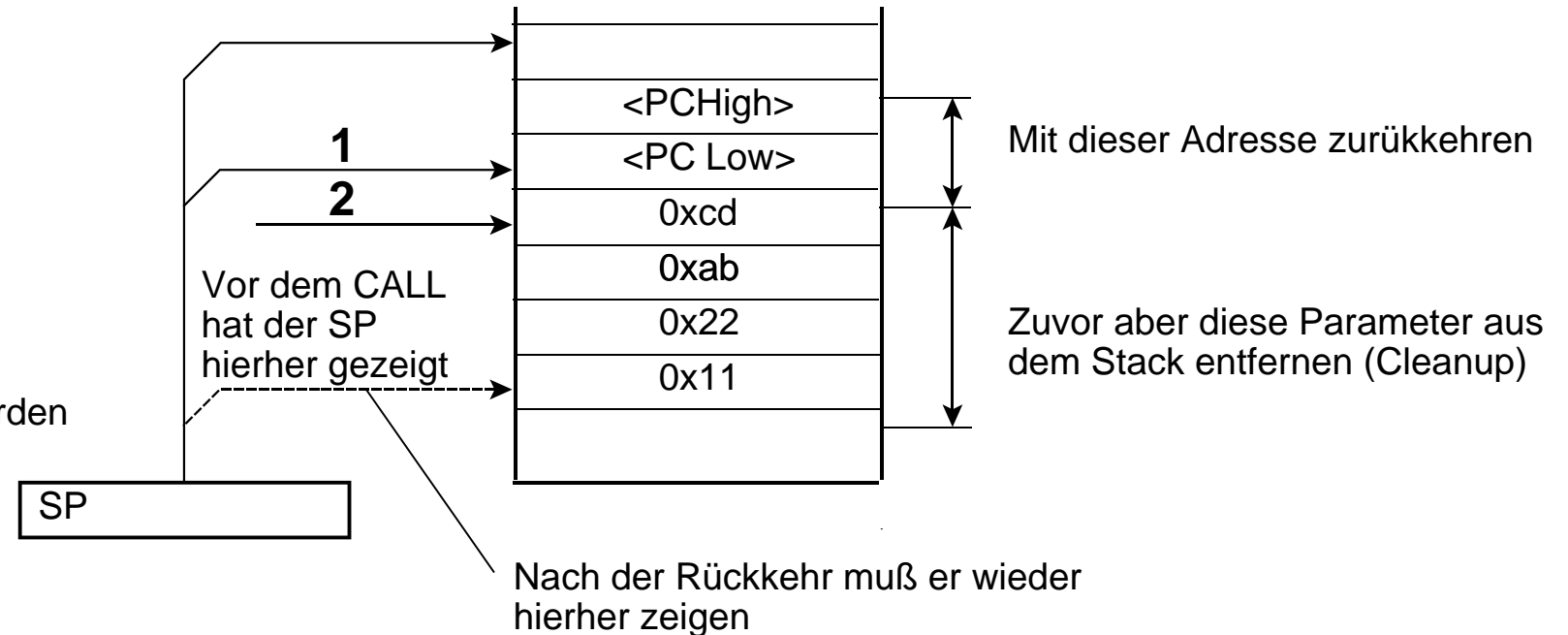


Zwei POPs holen die Rückkehradresse ins Z-Register. Danach zeigt der SP auf die **Position 1**

```
ldi temp,low(@0-1)
add yl,temp
ldi temp,high(@0-1)
adc yh,temp
out spl,yl
out sph,yh
ijmp
```

Das Y-Register zeigt auf das erste Parameterbyte (**Position 2**). Wir müssen die Anzahl der Parameterbytes -- 1 addieren, um auf die Stackposition vor dem Unterprogrammaufruf zu kommen. Dann kann mit dem Inhalt des Z-Registers indirekt verzweigt werden.

So sieht der Stack nach dem CALL aus:



**Der Stack des Atmel AVR (5)**  
**Parameterübergabe im Stack**  
**Die Makros clrstackretx und clrtstackrety**  
12. 8. 2013