

# 1 Shifts

## 1.1 SHL / SHR

Als mathematische Operation funktioniert folgendes nur fuer Zahlen welche als vorzeichenlose Zahl interpretiert werden sollen.

Als einfacher Bitshift ist dies zu vernachlaessigen.

### 1.1.1 SHL

"SHL r/imm n" ( $n := N$ )  $\rightarrow$  linksschieben um n Stellen:

n: 1  $\Rightarrow$  1010 0110  $\rightarrow$  0100 1100

$\rightarrow$  SHL wirkt wie eine schnelle und rechenguenstige Multiplikation mit  $basis^n$   
(basis := 2, da wir Zahlen binaer darstellen)

### 1.1.2 SHR

"SHR r/imm n" ( $n := N$ )  $\rightarrow$  rechtsschieben um n Stellen:

n: 1  $\Rightarrow$  1010 0110  $\rightarrow$  0101 0011

$\rightarrow$  SHR wirkt wie eine schnelle und rechenguenstige Division mit  $basis^n$

## 1.2 SAL / SAR

Als mathematische Operation funktioniert folgendes SAR nur fuer Zahlen des Zweierkomplements aka vorzeichenbehaftete Zahlen.

Dies waere ein Bitshift mit beachtung des letzten / fuehrenden Bits.

### 1.2.1 SAL

SAL verhaelt sich genauso wie SHL

### 1.2.2 SAR

"SAR r/imm n" ( $n := N$ )  $\rightarrow$  arithmetisches rechsschieben um n Stellen (hierbei erhaelt man fuer alle geshifteten Stellen das hoechste Bit  $\rightarrow$  Bewahrung des Vorzeichens):

n: 2  $\Rightarrow$  1001 1100  $\rightarrow$  1110 0111

$\rightarrow$  Division mit  $basis^n$  bewahrt das Vorzeichen

$\rightarrow$  rundet nach unten ab fuer Divisionen mit negativen Ergebnis

## 2 Rotationen

### 2.1 ROL / ROR

#### 2.1.1 ROL

"ROL r/imm n" ( $n := N$ )  $\rightarrow$  linksrotieren um n Stellen:

n: 2  $\Rightarrow$  1110 0011  $\rightarrow$  1000 1111

- die vorderen n Bits werden abgeschnitten und "nach hinten rotiert"
- ROL hat keine arithmetische Bedeutung

#### 2.1.2 ROR

"ROR r/imm n" ( $n := N$ )  $\rightarrow$  rechtsrotieren um n Stellen:

n: 2  $\Rightarrow$  1110 0011  $\rightarrow$  1111 1000

- die hinteren n Bits werden abgeschnitten und "nach vorne rotiert"
- ROR hat keine arithmetische Bedeutung