

Arithmetic Logic Unit (ALU 9+8 Bit)

---

Bit 1	Bit			
Vorzeichenerweiterung	123456789			
Operand 1	AL1	<u>00000000</u>	<u>00000000</u>	ALBA Faktor für Addition
Operand 1	AL2	<u>00000000</u>	<u>00000000</u>	ALBS Faktor für Subtraktion
Übertrag	ALC	<u>00000000</u>		
Resultat	ALR	<u>00000000</u>	<u>00000000</u>	0 ALx extra Bit
		1+8 Bit	ARL 8 Bit	
		Resultat: High Byte	Low Byte	

---

Die Ablaufsteuerung erfolgt nach dem rechten Bit von ARL (Resultat Low Byte) und dem extra Bit ALx.

>> Die Ergebniszeile (Resultat) ALR ARL ALx wird 1 Bit nach rechts geschoben, das Hilfsbit fällt dabei raus, links wird als Vorzeichenerweiterung das zum weggeschobenen Bit identische ergänzt.

ARL ALx	Kurzform	Bedeutung
...0 0	>>	nur schieben, shift >>
...0 1	add >>	Faktor wird addiert, dann >>
...1 0	sub >>	-Faktor wird addiert, dann >>
...1 1	>>	nur schieben, shift >>

ALR wird als Zwischensumme jeweils nach AL1 kopiert.

Der Multiplikations-Algorithmus (nach Booth, für vorzeichenbehaftete Faktoren)

```

-----
set ALBA Faktor 1           // Faktor für Addition
set ALBS -Faktor 1          // -Faktor (vorab zu berechnen, Zweierkomplement für Subtraktion)
set ALR  00000000           // 9 Bit: Anfangswert für fortlaufende Summenbildung
set ARL  Faktor 2           // 8 Bit ins Resultat Low Byte setzen
set ALx  0                   // 1 extra Bit (Anfangswert 0)
//-----
loop 8                       // Schleife 8-mal durchlaufen
  b1 = right(ARL)            // rechtes Bit von ARL
  b2 = get(ALx)              // extra Bit
  if b1=b2                    // 0 0 und 1 1
    shift >>                 // nur rechts schieben
  else
    if b1=0 ^ b2=1
      set AL2 ALBA           // Faktor 1 zur Addition von ALBA in AL2 setzen
    endif
    if b1=1 ^ b2=0
      set AL2 ALBS           // -Faktor 1 zur Addition von ALBS in AL2 setzen
    endif
    set AL1 ALR              // laufende Summe nach AL1 umspeichern
    // Addition in der ALU ausführen, das Vorzeichenbit von AL2 wird vorab dupliziert.
    //
    add // <-- Additions-Algorithmus add_ALU_9 (AL1+AL2) im 9 Bit-Rechenwerk mit Vorzeichenerweiterung
    //
    shift >>                 // rechts schieben
  endif
end_loop                     // Schleifenende

```

Wird der Zahlenbereich -128..127 nicht überschritten, befindet sich das 8 Bit Produkt im Low Byte ARL.