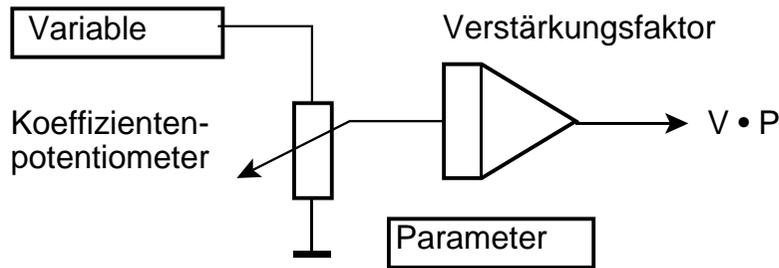


Aus 4 Eingangsvariablen V1 bis V4 wird ein Ergebnis gebildet. Zu jeder Eingangsvariable gehört ein Skalierungsparameter (P1 bis P4). Der Skalierungsparameter ist eine rationale Zahl, dargestellt als Binärbruch (Zähler / Nenner). Er wirkt wie ein Verbund von Koeffizientenpotentiometer und Verstärkungsfaktoreinstellung.



\*: Vorzeichen / Festwert  
 \*\*: Vorzeichen / Begrenzung

### Variante 1: Skalierung einer einzelnen Variablen (V1 bis V4)



Parameter = Koeffizient • Verstärkungsfaktor

$$P = K \cdot A$$

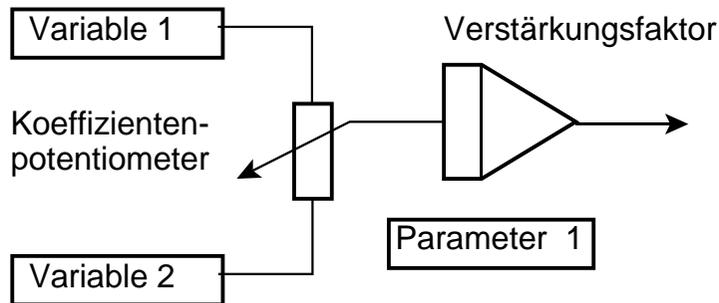
$$K < 1; A > 1$$

Darstellung als Bruch mit Nenner N:

$$P = K \cdot A; K < 1; A > 1$$

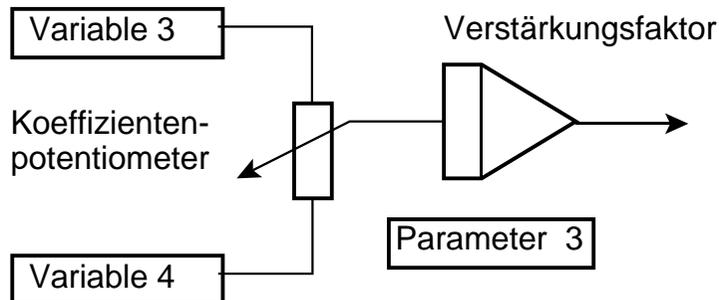
$$P = \frac{A \cdot N + K}{N}$$

### Variante 2: Skalierter Differenz / "Spannungsteiler" für die Variablenpaare V1, V2 und V3, V4



Skalierte Differenz:  $(V1 - V2) \cdot P1$

Spannungsteiler:  $(V1 - V2) \cdot P1 + V2$



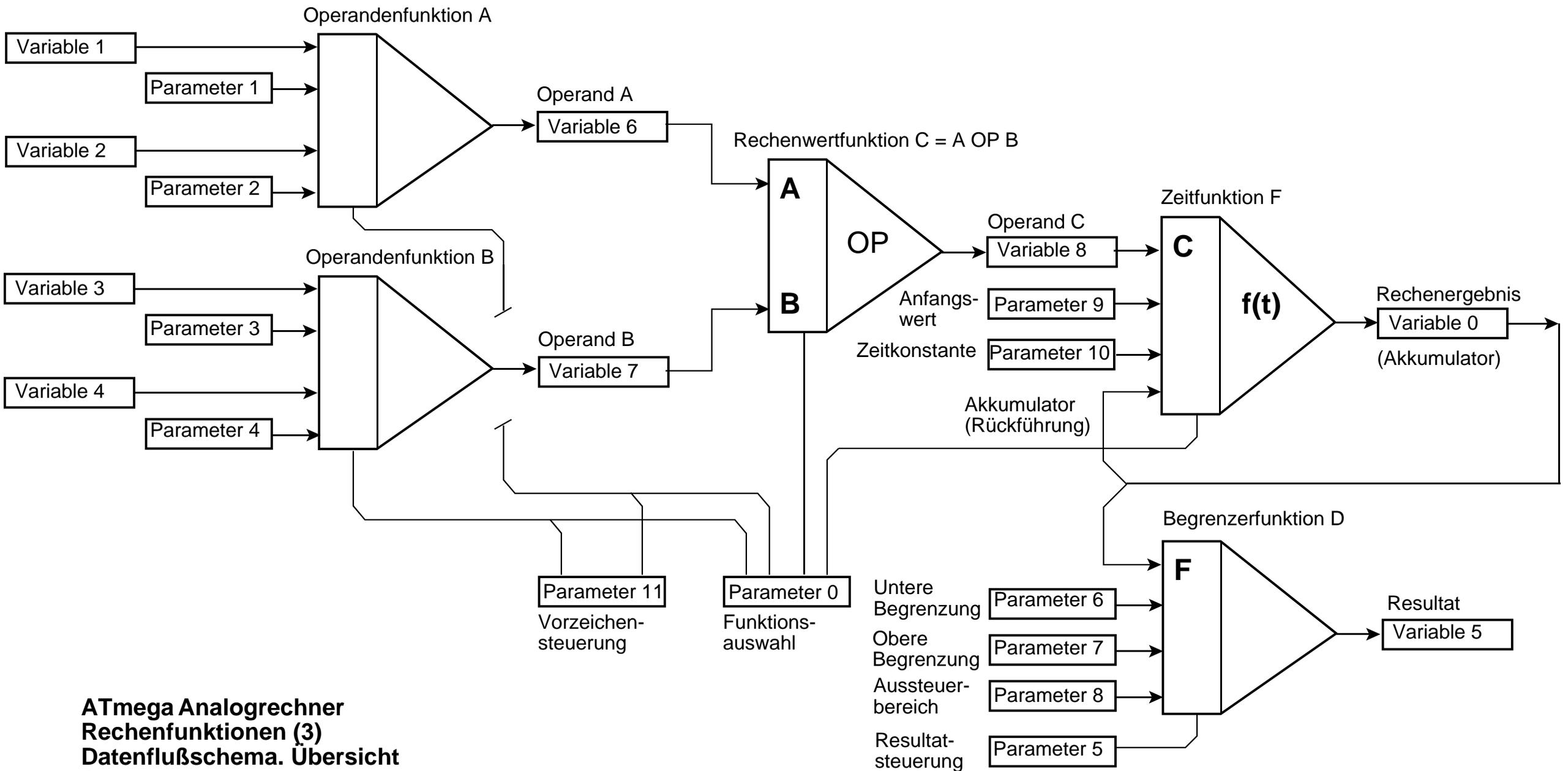
Skalierte Differenz:  $(V3 - V4) \cdot P3$

Spannungsteiler:  $(V3 - V4) \cdot P3 + V4$

**ATmega Analogrechner  
Rechenfunktionen (2)**

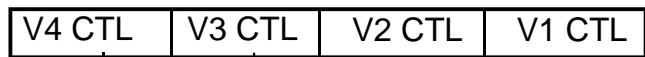
**Die Wirkung der  
Skalierungsparameter**

Stand: 1.4 vom 1. 8. 2016

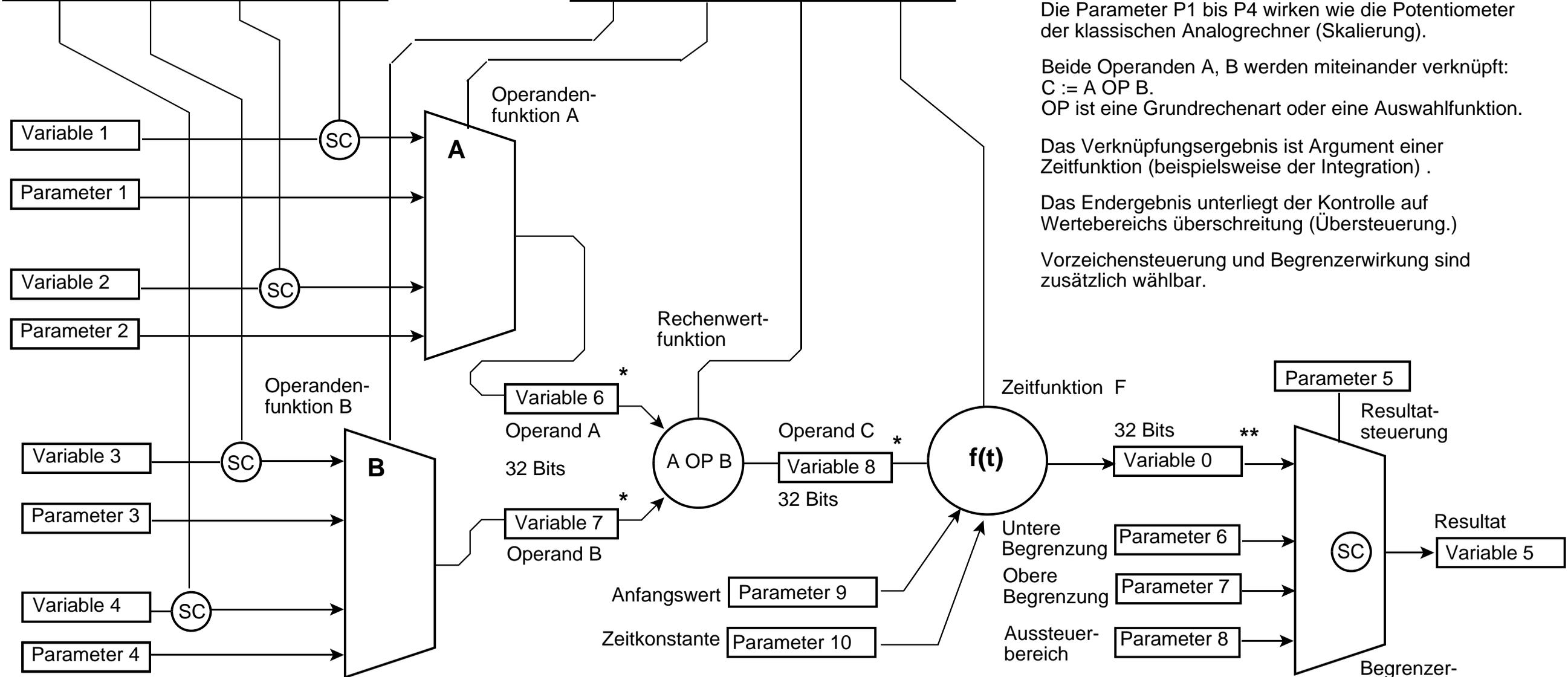


**ATmega Analogrechner**  
**Rechenfunktionen (3)**  
**Datenflußschema. Übersicht**  
 Stand: 1.4 vom 1. 8. 2016

Parameter 11 Festwert-und Vorzeichensteuerung



Parameter 0 Funktionsauswahl

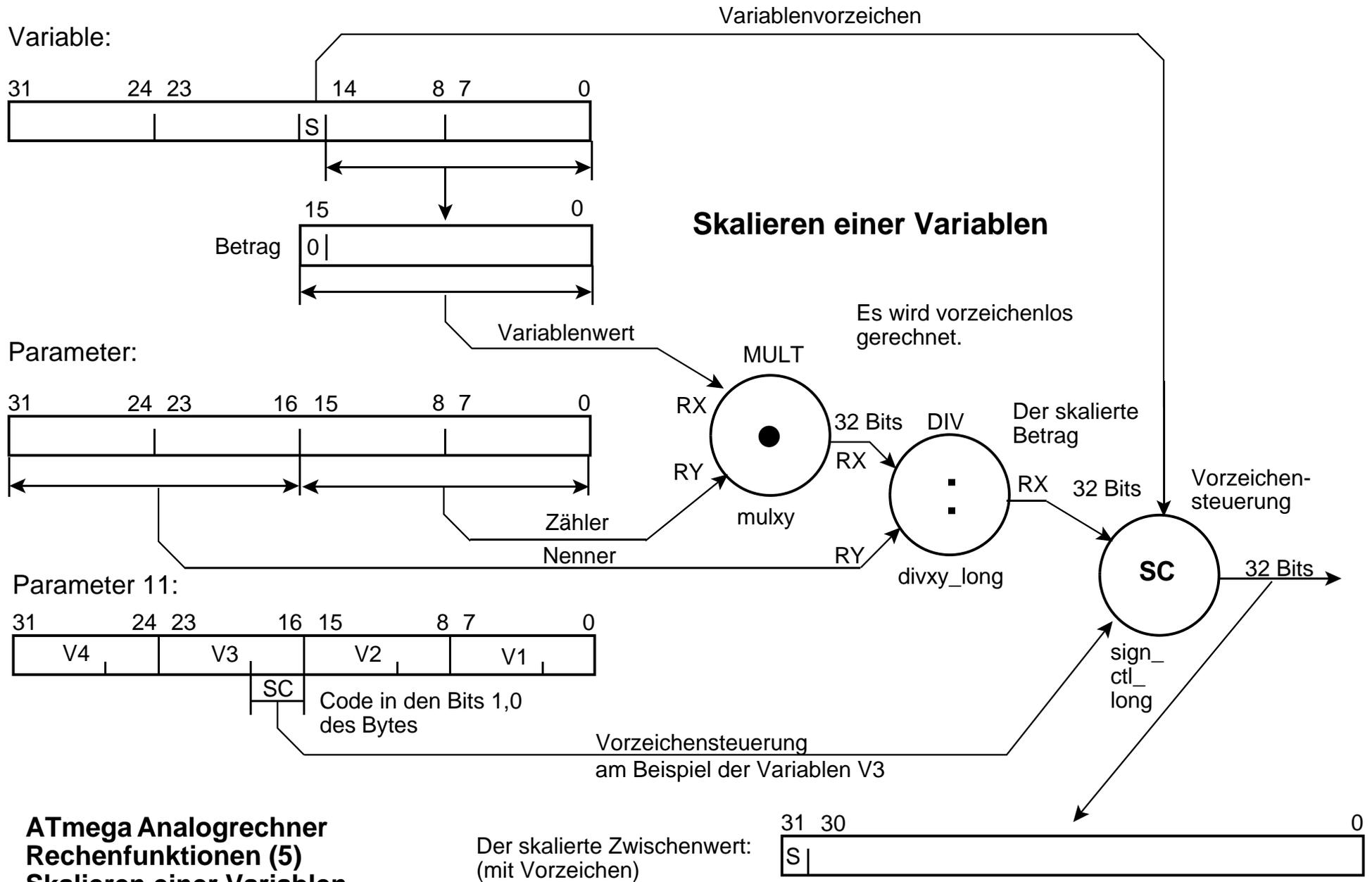


Aus den Variablen V1 und V2 entsteht der Operand A.  
 Aus den Variablen V3 und V4 entsteht der Operand B.  
 Die Parameter P1 bis P4 wirken wie die Potentiometer der klassischen Analogrechner (Skalierung).  
 Beide Operanden A, B werden miteinander verknüpft:  
 $C := A \text{ OP } B$ .  
 OP ist eine Grundrechenart oder eine Auswahlfunktion.  
 Das Verknüpfungsergebnis ist Argument einer Zeitfunktion (beispielsweise der Integration) .  
 Das Endergebnis unterliegt der Kontrolle auf Wertebereichs überschreitung (Übersteuerung.)  
 Vorzeichensteuerung und Begrenzerwirkung sind zusätzlich wählbar.

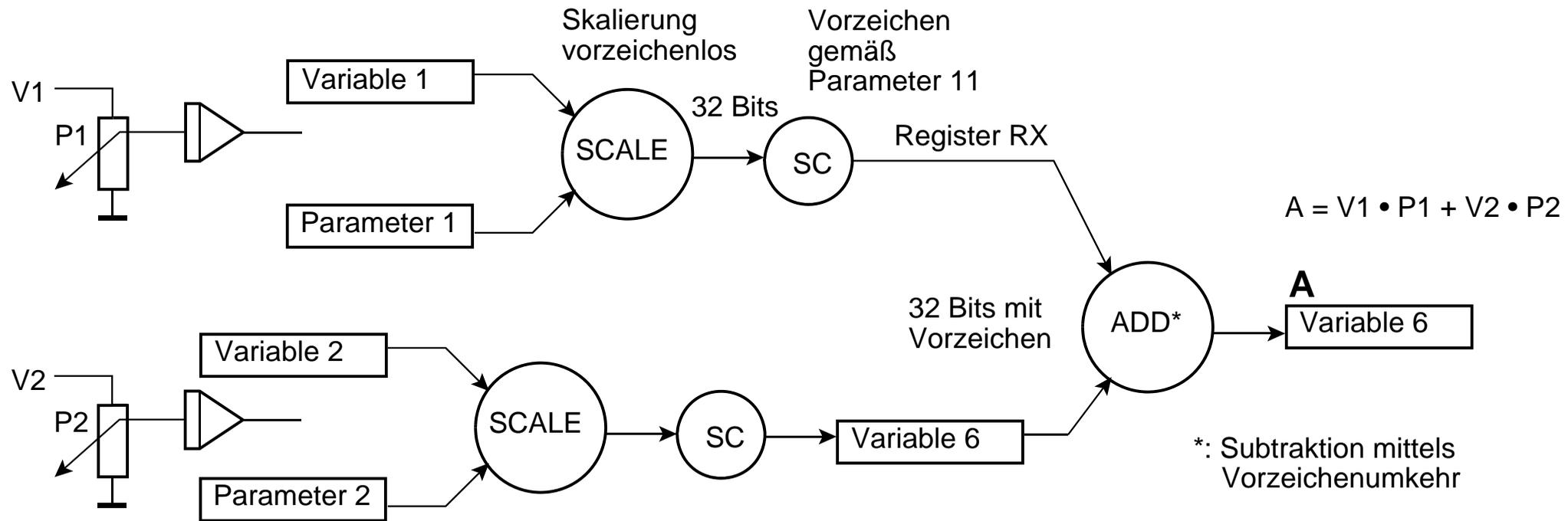
SC = Vorzeichensteuerung (Sign Control). Die Vorzeichensteuerung gehört zur Operandenfunktion (A oder B). Sie ist hier nur der Übersichtlichkeit wegen so dargestellt, als sei sie vorgeschaltet.

\* : Zwischenspeicherung in den Variablen zwecks Unterstützung der Fehlersuche (Sichtbarkeit).  
 \*\*: Akkumulator

**ATmega Analogrechner**  
**Rechenfunktionen (4)**  
**Datenflußschema. Einzelheiten**  
 Stand: 1.4 vom 1. 8. 2016

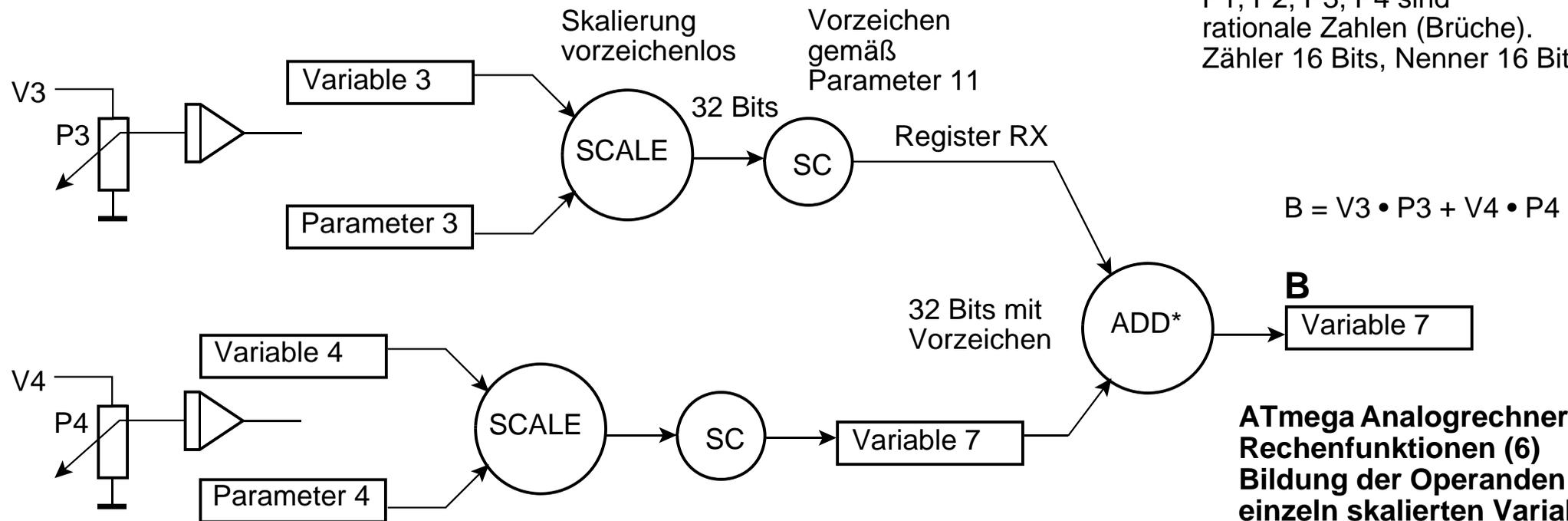


**ATmega Analogrechner**  
**Rechenfunktionen (5)**  
**Skalieren einer Variablen**  
 Stand: 1.4 vom 1. 8. 2016

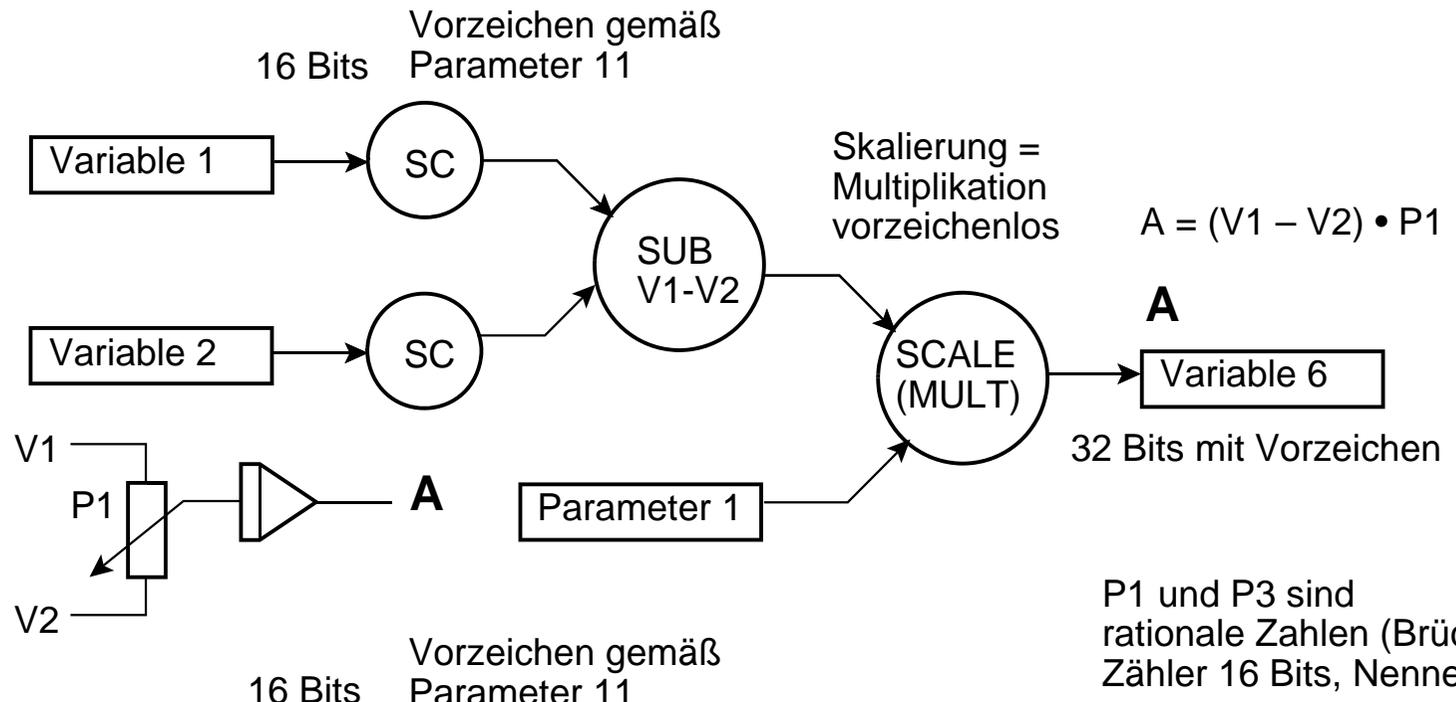


\*: Subtraktion mittels Vorzeichenumkehr

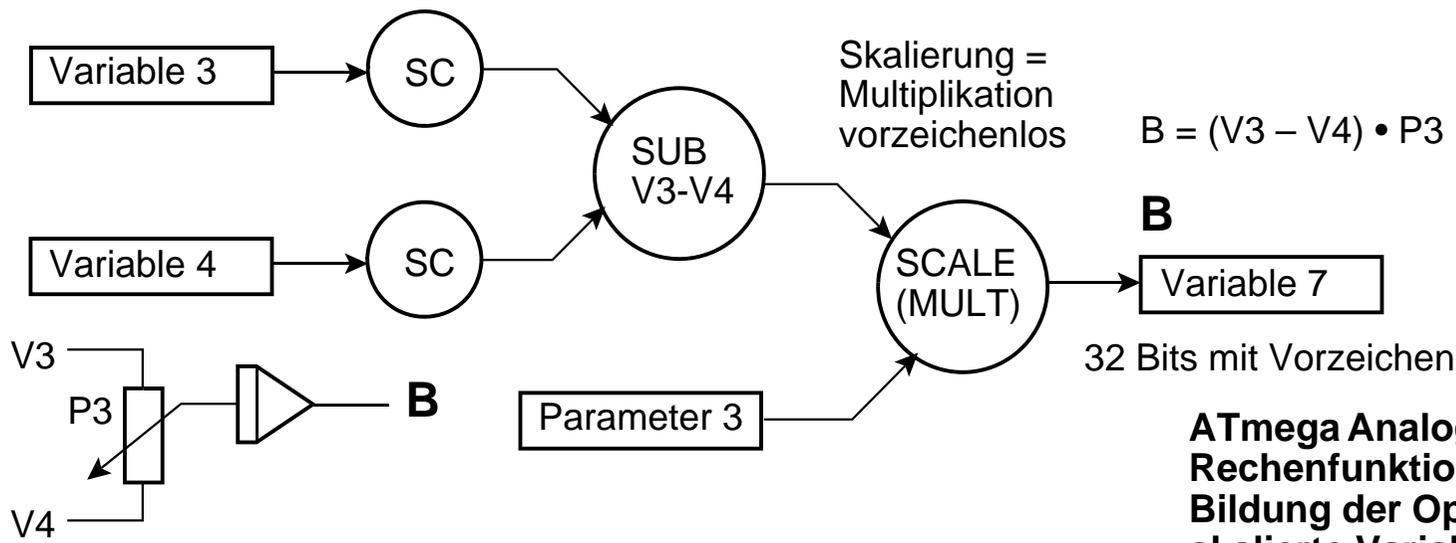
P1, P2, P3, P4 sind rationale Zahlen (Brüche).  
Zähler 16 Bits, Nenner 16 Bits.



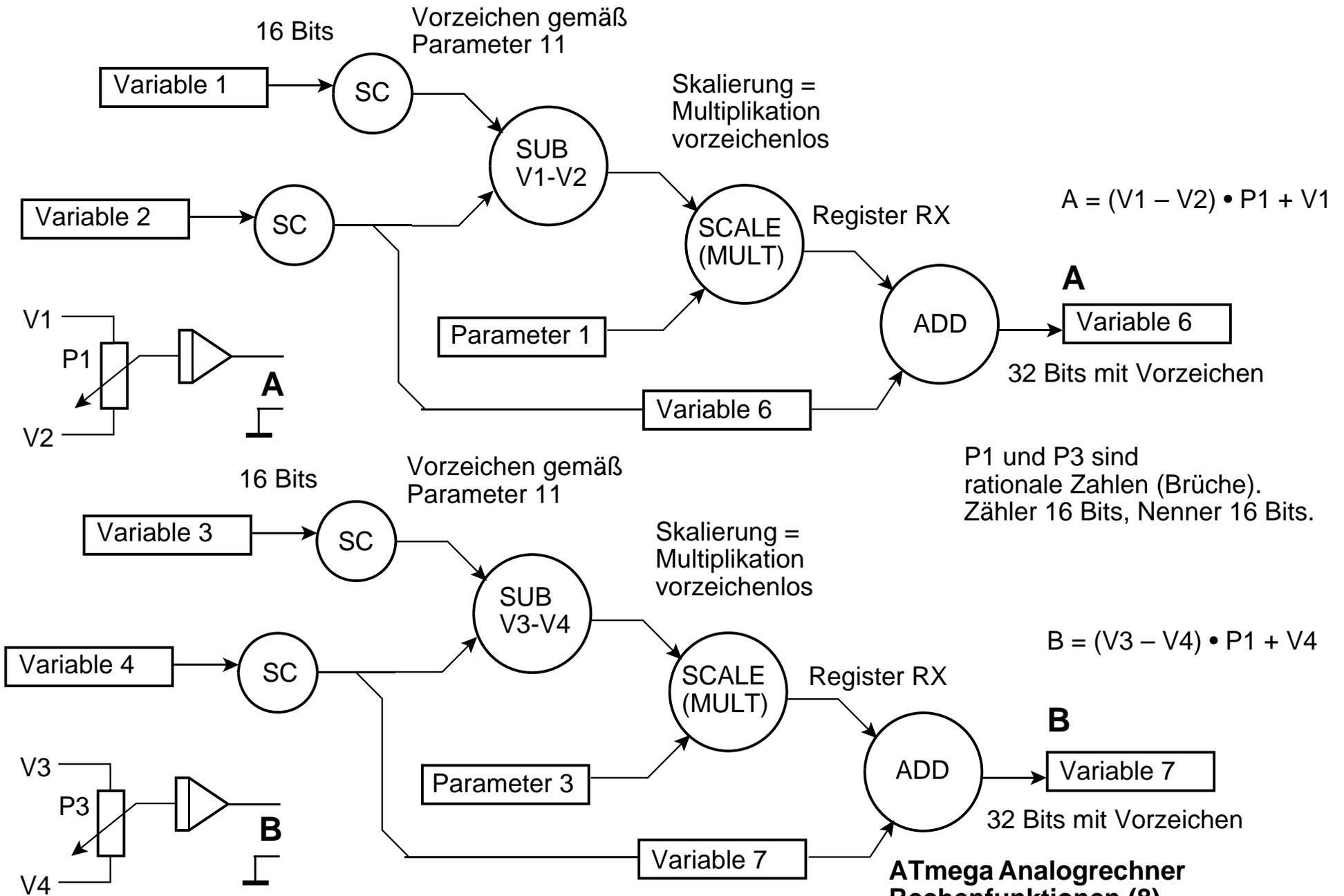
**ATmega Analogrechner**  
**Rechenfunktionen (6)**  
**Bildung der Operanden A, B aus**  
**einzelnen skalierten Variablen**  
Stand: 1.4 vom 1. 8. 2016



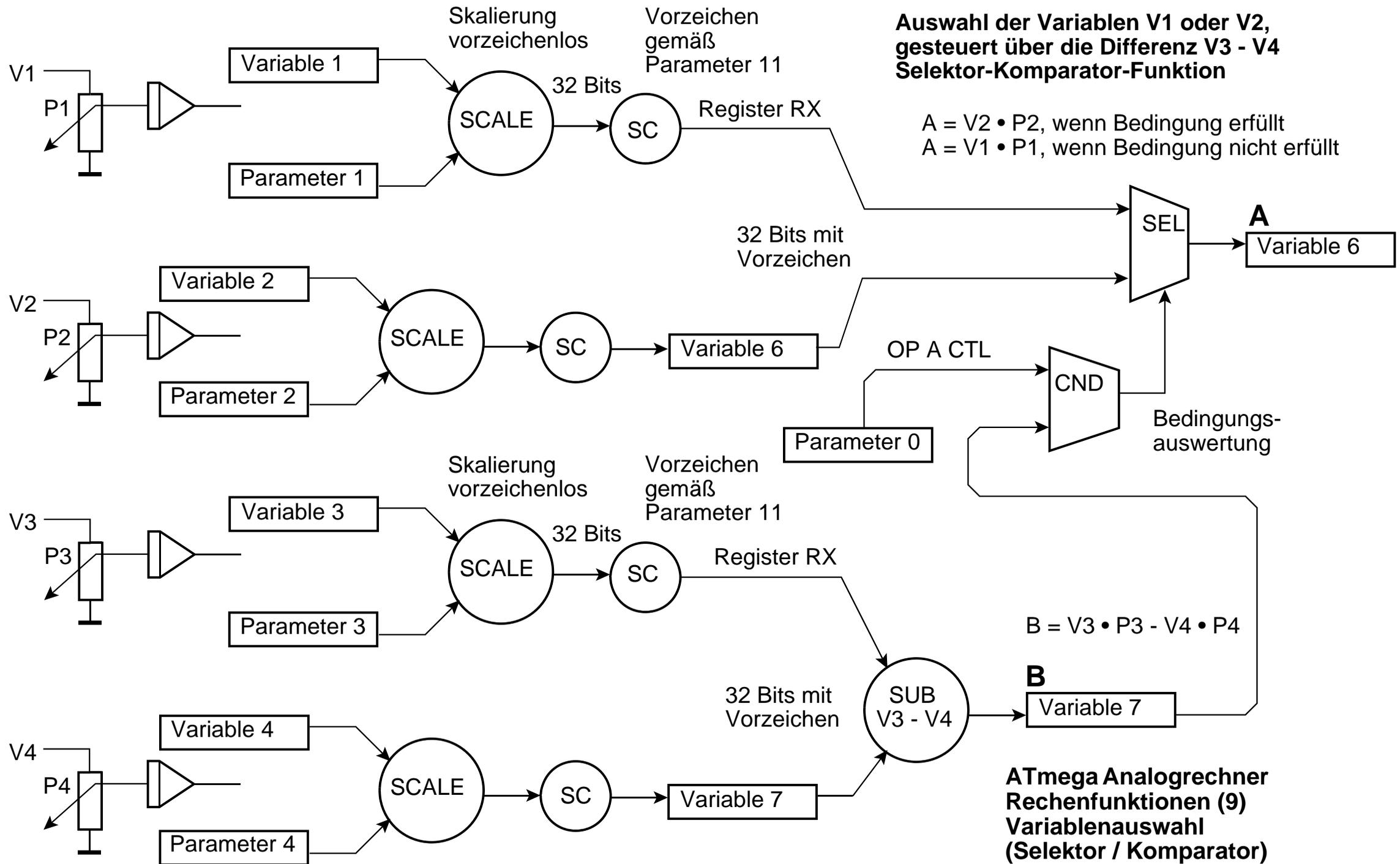
P1 und P3 sind rationale Zahlen (Brüche).  
Zähler 16 Bits, Nenner 16 Bits.



**ATmega Analogrechner  
Rechenfunktionen (7)  
Bildung der Operanden A, B als  
skalierte Variablendifferenzen**  
Stand: 1.4 vom 1. 8. 2016

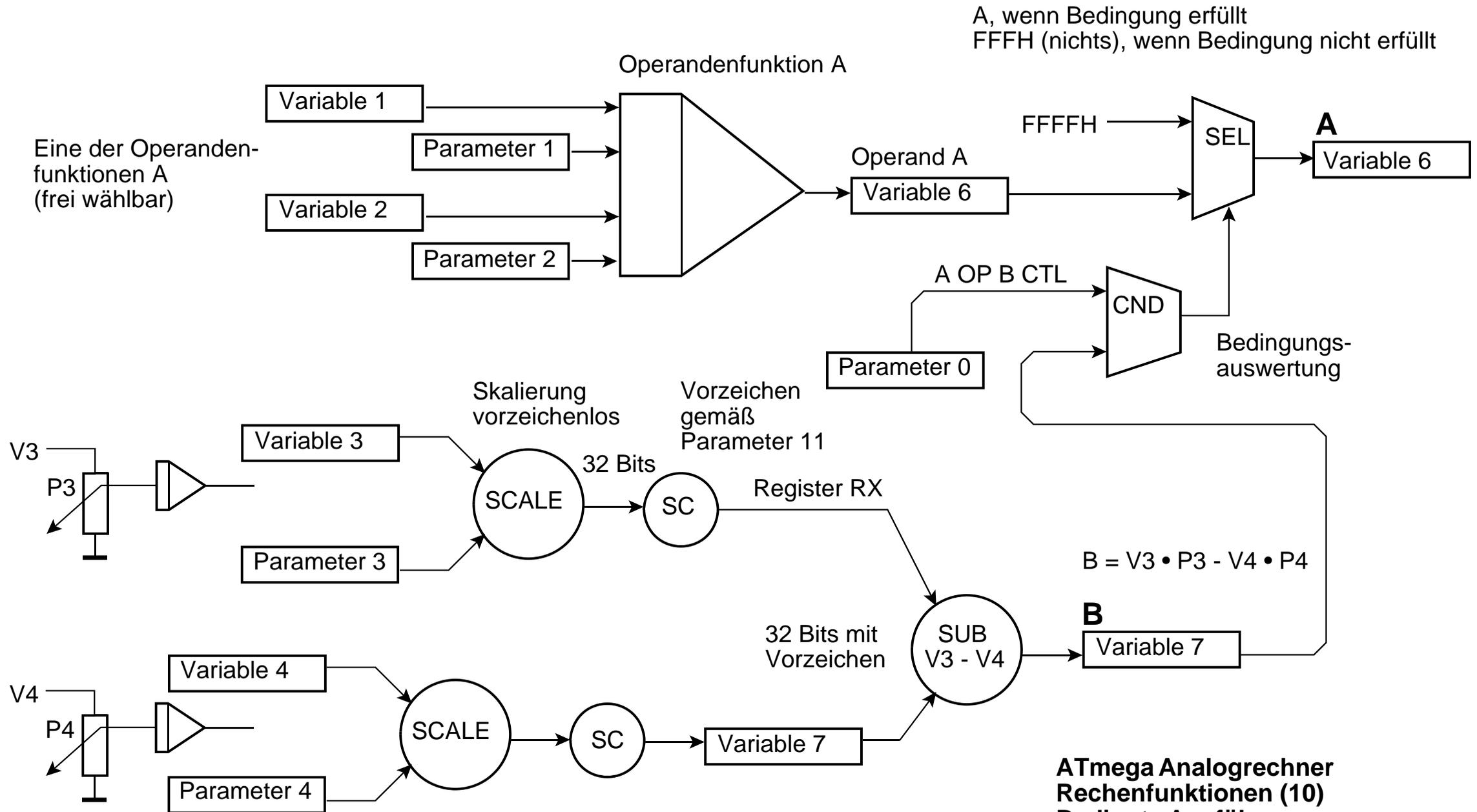


**ATmega Analogrechner  
Rechenfunktionen (8)  
Bildung der Operanden A, B aus  
Variablendifferenzen.  
Prinzip Spannungsteiler  
Stand: 1.3 vom 11. 7. 2016**



**ATmega Analogrechner**  
**Rechenfunktionen (9)**  
**Variablenauswahl**  
**(Selektor / Komparator)**  
 Stand: 1.3 vom 11. 7. 2016

## Bedingte Ausführung der Operandenfunktion A, gesteuert über die Differenz V3 - V4



Die Rechenregister:

Rechenregister RX  
Akkumulator

RX0
RX1
RX2
RX3

Rechenregister RY  
2. Operand

RY0
RY1

2. Operand 32 Bits

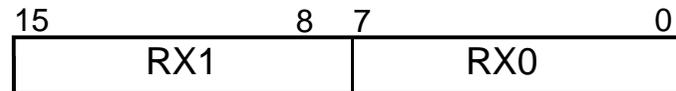
RY0
RY1
AUX0
AUX1

Hilfsregister AUX

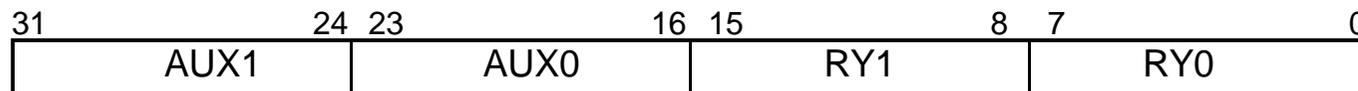
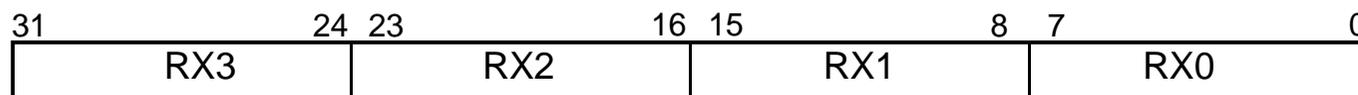
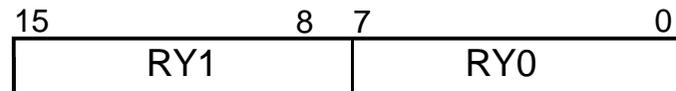
AUX0
AUX1
AUX2
AUX3
AUX4
AUX5
AUX6
AUX7

Verlängerung RY auf 32 Bits / Multiplikation / Verlängerung RX auf 48 Bits / Divisionsrest
Multiplikation
Vorzeichensteuerung
Parameterpuffer höherwertige 16 Bits

16-Bit-Operanden:



32-Bit-Operanden:



**ATmega Analogrechner  
Rechenfunktionen (11)  
Rechenregister und  
Operandenformate**

Stand: 1.3 vom 11. 7. 2016

## Statischer Speicher

(Inhalte bleiben erhalten und sind von außen zugänglich (Dienstfunktionen / Bedienininterface))

Operanden (zum Rechnen)

Ergebnisse

Rechenregister:

16 Variable:

V0
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15

16 Parameter:

P0
P1
P2
P3
P4
P5
P6
P7
P8
P9
P10
P11
P12
P13
P14
P5

Rechenregister RX  
Akkumulator

RX0
RX1
RX2
RX3

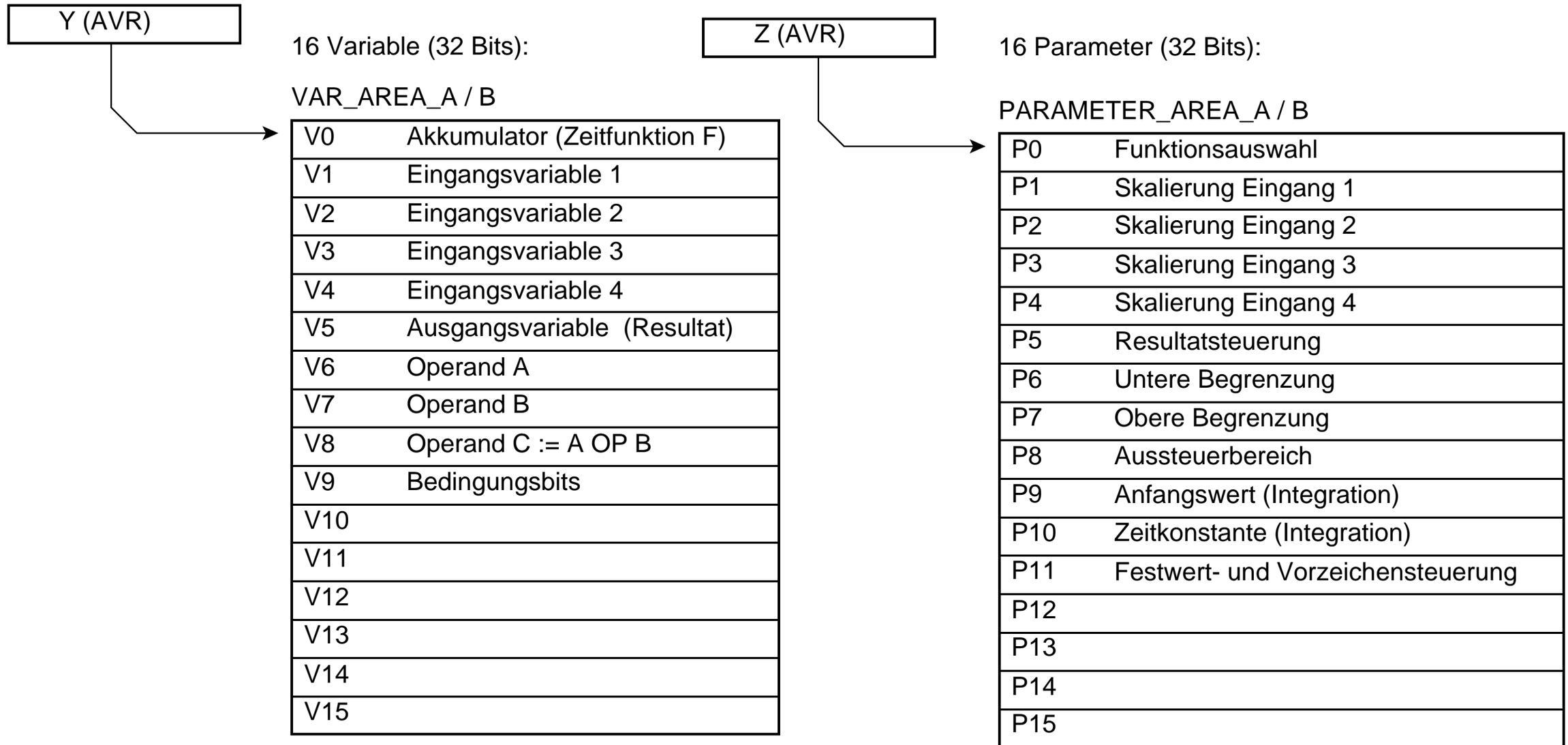
Rechenregister RY  
2. Operand

RY0
RY1

Hilfsregister AUX

AUX0
AUX1
AUX2
AUX3
AUX4
AUX5
AUX6
AUX7

**ATmega Analogrechner  
Rechenfunktionen (12)  
Variable, Parameter und  
Rechenregister**  
Stand: 1.3 vom 11. 7. 2016



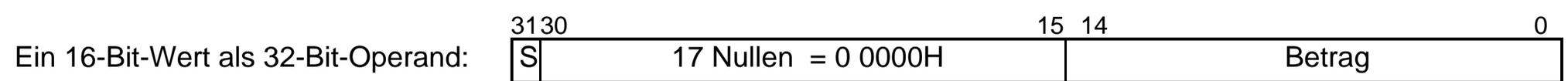
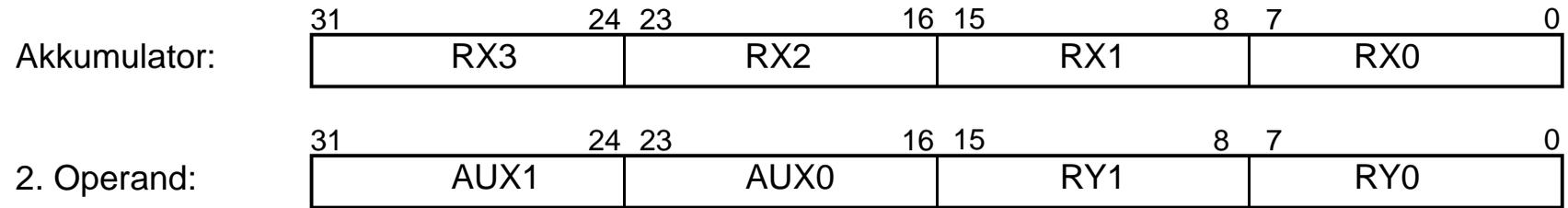
Ein Mikrocontroller unterstützt zwei Rechenfunktionen, Rechenfunktion A und Rechenfunktion B. Jede hat 16 Variable und 16 Parameter.

Beim Rechnen zeigen

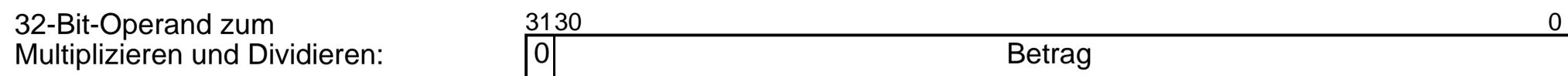
- das AVR-Register Y auf den Anfang des Variablenbereichs,
- das AVR-Register Z auf den Anfang des Parameterbereichs.

Alle Bytes der beiden Bereiche sind somit über Offset-Adressen erreichbar.

Die Rechenregister:

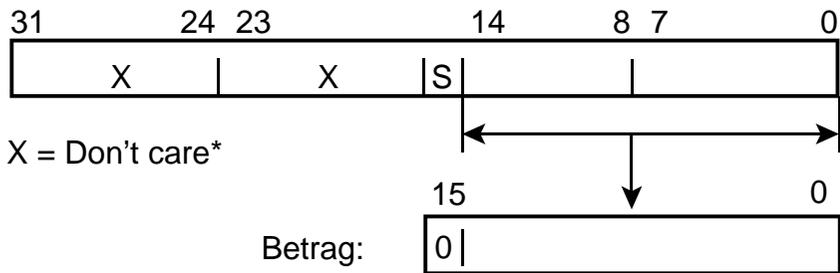


S = Vorzeichen (Sign). Der Betrag ist 15 oder 31 Bits lang.

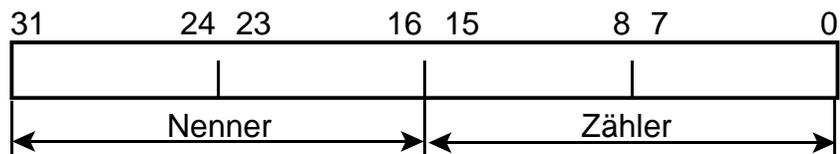


Es wird vorzeichenlos multipliziert und dividiert. Das Vorzeichen wird später aus den Variablenvorzeichen bestimmt und eingefügt.

16 Bits mit Vorzeichen:



Zähler und Nenner (je 16 Bits vorzeichenlos):



\*: Programmierhinweis: Mit diesen 16 Bits nichts tun (also nicht z. B. als Hilfsspeicher ausnutzen).

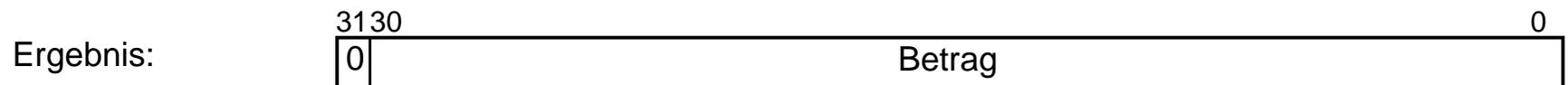
32 Bits mit Vorzeichen:



32 Bits vorzeichenlos:



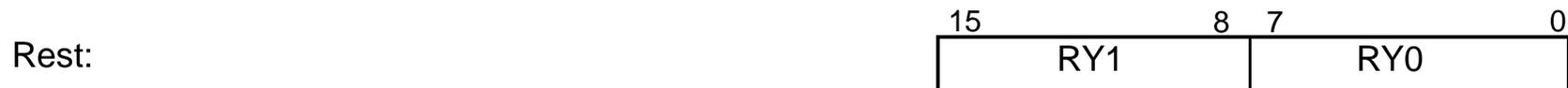
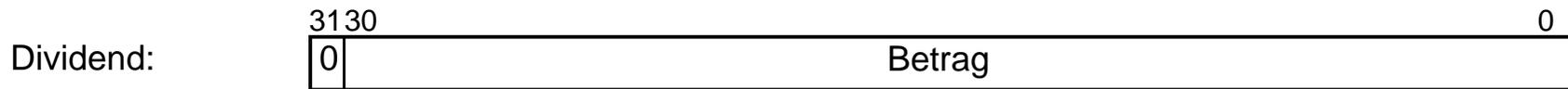
Multiplizieren. Operanden 16 Bits, Ergebnis 32 Bits.



Es wird vorzeichenlos multipliziert. Das Vorzeichen wird später aus den Variablenvorzeichen bestimmt und eingefügt.

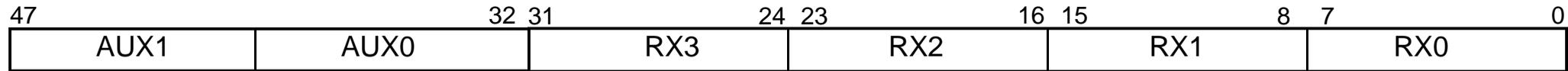
Dividieren. Dividend 32 Bits, Divisor 16 Bits, Quotient 16 Bits, Rest 16 Bits.

Die Rechenregister:

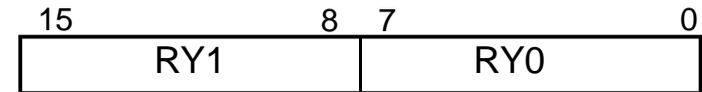


Diese Division ist die Umkehrung der Multiplikation. Quotient darf nicht länger werden als 15 Bits (Betrag). Dividieren durch 1 nicht möglich. Es wird vorzeichenlos dividiert. Das Vorzeichen wird später aus den Variablenvorzeichen bestimmt und eingefügt.

Dividieren. Dividend 32 Bits, Divisor 16 Bits, Quotient 32 Bits, Rest 16 Bits.

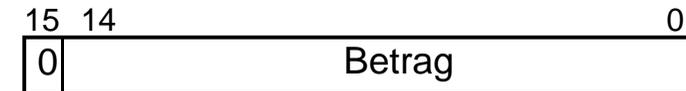


2. Operand:

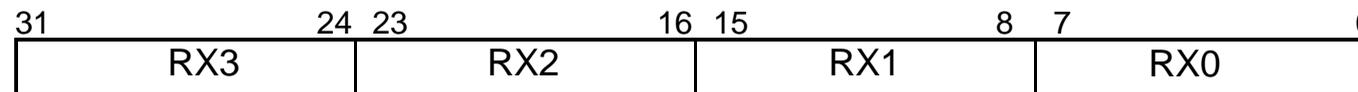


Nullen werden automatisch angefügt.

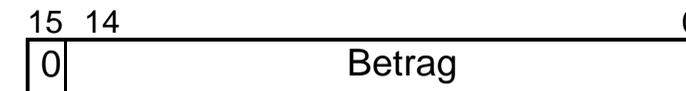
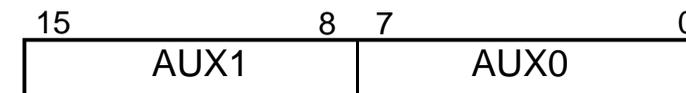
Divisor:



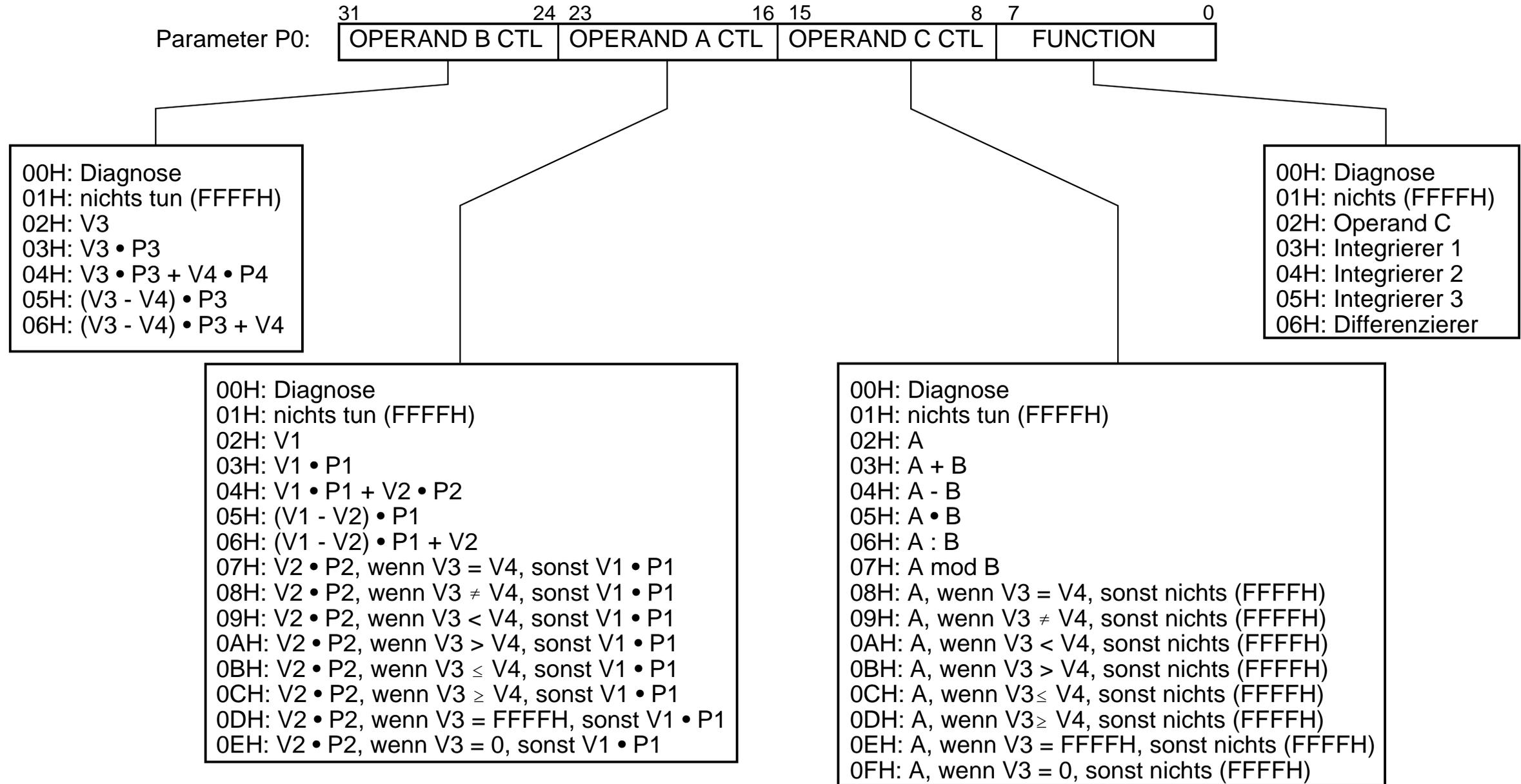
Quotient:



Rest:



Dividend wird automatisch auf 48 Bits verlängert.  
 Quotient darf 31 Bits lang werden (Betrag).  
 Dividieren durch 1 möglich.  
 Diese Division wirkt auch als Umkehrung der Multiplikation.  
 Es wird vorzeichenlos dividiert. Das Vorzeichen wird  
 später aus den Variablenvorzeichen bestimmt und eingefügt.



## Festwert- und Vorzeichensteuerung:



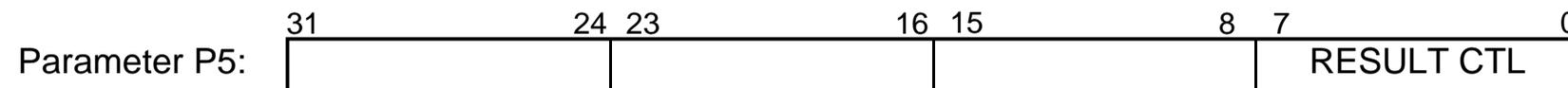
Jeder der Funktionscodes FC:



0: Variable. Der am Eingang angekommene Wert wird übernommen.  
1: Festwert. Der am Eingang angekommene Wert wird ignoriert.

0H: nichts tun  
1H: invertieren  
2H: immer positiv (Betrag)  
3H: immer negativ

## Resultatsteuerung:



← MARGINS →

Bit 2: untere Begrenzung (P6)  
Bit 3: obere Begrenzung (P7)

0H: nichts tun  
1H: invertieren  
2H: immer positiv (Betrag)  
3H: immer negativ