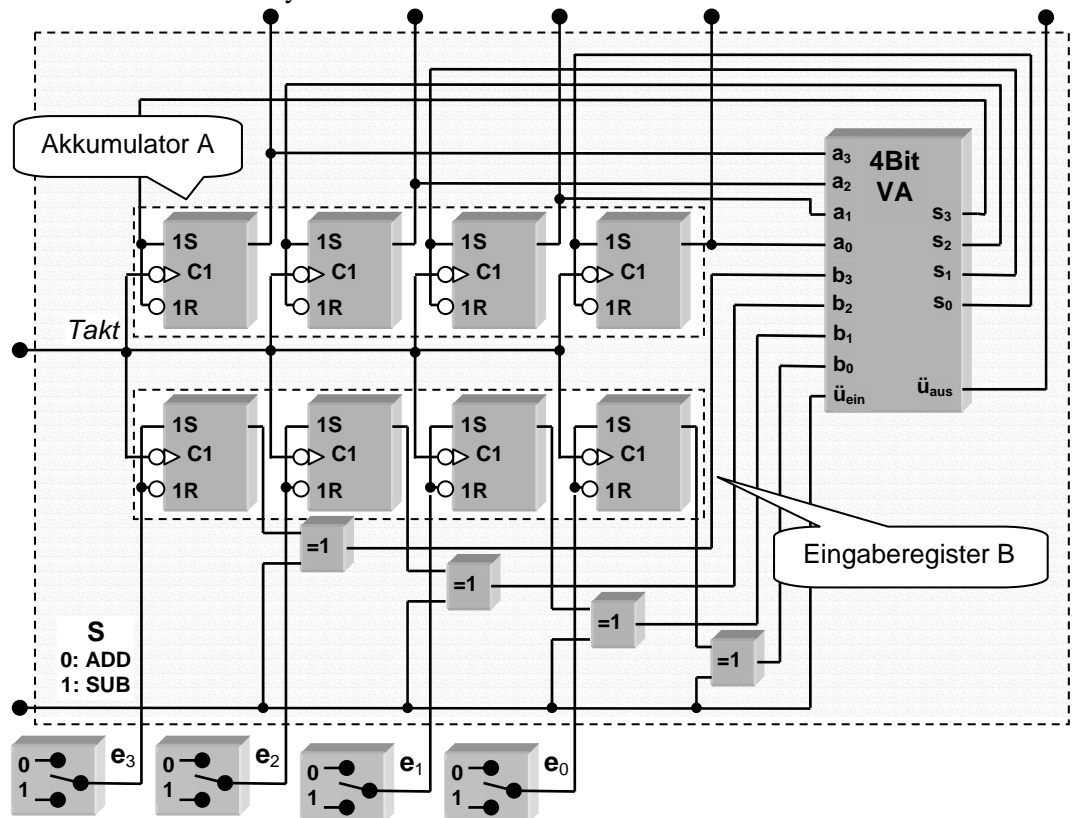


3.4 Ein programmierbares Rechenwerk

Etwas einfacher als serielle Schaltwerke sind parallele Rechenwerke zu realisieren, weil diese ihre Ergebnisse in einem Takt berechnen. Wir wollen ein *umschaltbares paralleles 4-Bit-Addier-/Subtrahierwerk* entwickeln, das neben dem Akkumulator *A* über ein zweites Eingaberegister *B* verfügt. Es arbeitet ansonsten genauso wie das schon bekannte – nur werden die Schalterreihen durch Register ersetzt. Als Addierer wollen wir einen 4-Bit-Volladdierer wählen, wie er z. B. auf dem IC 7486 realisiert ist. Aus Platzgründen benutzen wir hier die Symbole für einfache JK-MS-FFs.



Denken wir uns die ganze Schaltung auf einer Platine untergebracht, dann verfügt diese über sechs Eingabeleitungen:

- 4 Eingabeleitungen für das Register *B*, die hier von den Schaltern belegt werden
- 1 Steuerleitung *S*
- 1 Taktleitung *T*

Entsprechend gibt es fünf Ausgangsleitungen:

- 4 Datenleitungen, die den Inhalt des Akkumulators repräsentieren
- 1 Übertrags-(Carry-)-Bit

Wir können uns die Schaltung symbolisch als einen „Chip“ vorstellen. Lassen wir den Chip jetzt eine Rechnung durchführen, die aus mehreren Operationen besteht, dann können wir die Belegungen der Eingangsleitungen nacheinander notieren.

Beispiel: $3 + 4 - 5 + 1 =$

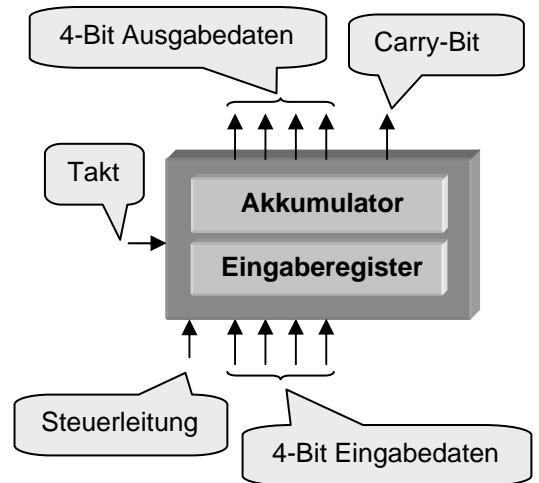
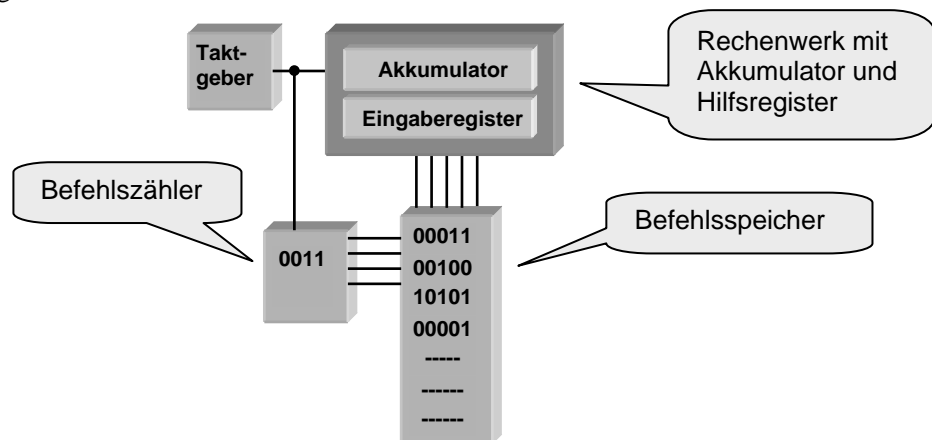
S	Eingabe	Kommentar
0	0011	3 laden
0	0100	4 addieren
1	0101	5 subtrahieren
0	0001	1 addieren

Wir können die Bitfolge

00011
00100
10101
00001

als „Maschinenprogramm“ auffassen, das die Arbeitsweise der Maschine festlegt.

Legen wir dieses Maschinenprogramm in einem Speicher ab, der seine Inhalte taktweise an die Eingänge der Schaltung bringt, dann haben wir ein programmierbares Rechenwerk gefunden. Um festzulegen, welcher der gespeicherten Befehle als nächstes ausgeführt werden soll, benötigen wir noch zusätzlich einen Programmschrittzähler, der ebenfalls taktgesteuert arbeitet.



Statten wir so ein Rechenwerk mit zusätzlichen Steuerleitungen aus, z. B.

- um den Akkumulator zu löschen,
- um mit oder ohne Carrybit zu addieren oder zu subtrahieren,
- um den Akkumulatorinhalt um ein Bit nach rechts zu verschieben,
- ...

dann erhalten wir zusätzliche Möglichkeiten, aber auch zusätzlichen Bedarf, das Rechenwerk zu programmieren. Die Belegung der Steuerleitungen bei einer bestimmten Operation ergibt dann die Codierung des entsprechenden Befehls.

*In diesem Sinne ist **Programmierung** die Festlegung der zeitlichen Reihenfolge mehrerer Operationen. Die Auswahl der jeweiligen Operation (Addieren, Laden, ...) erfolgt durch die Belegung geeigneter Steuerleitungen, und eben diese Belegung kann als Codierung des entsprechenden Befehls aufgefasst und in einem Speicher abgelegt werden. Ein **Programm** ist dann eine bestimmte Folge solcher Steuerleitungsbelegungen. Die Reihenfolge, in der die Befehle ausgeführt, also die Steuerleitungen auf die gespeicherten Werte gesetzt werden, wird durch einen Programmschrittzähler bestimmt.*