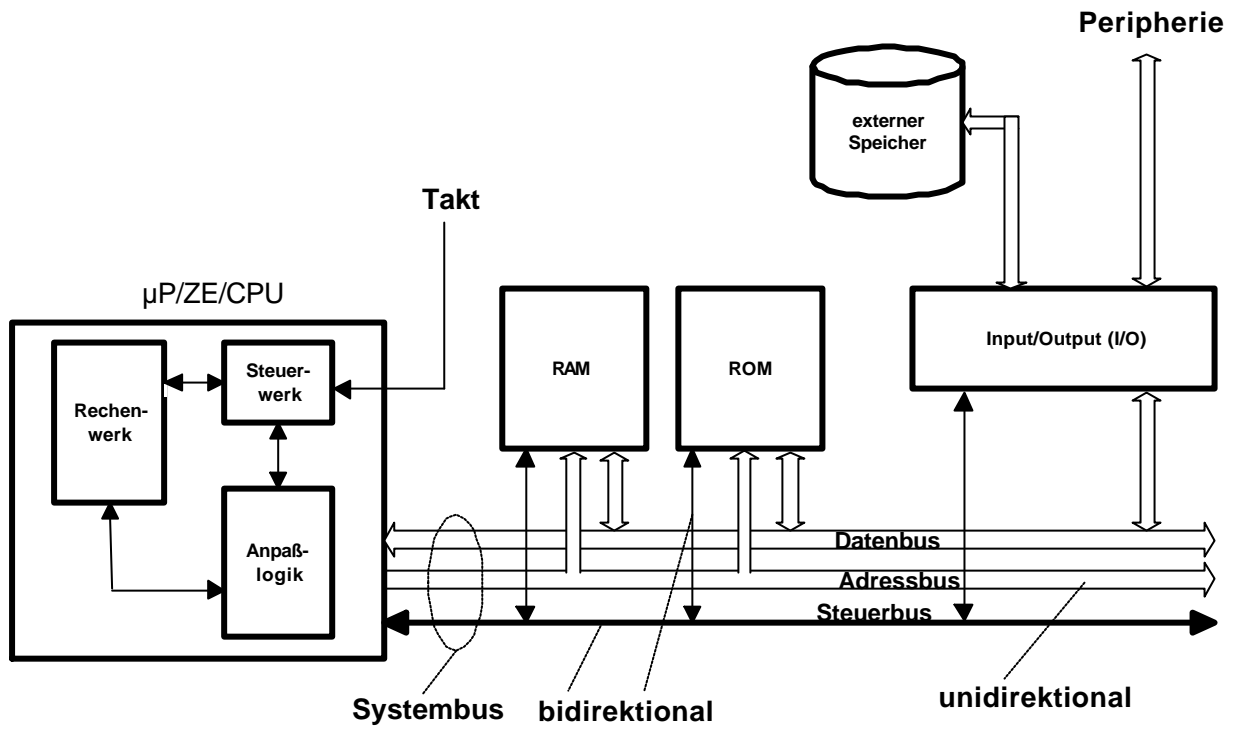


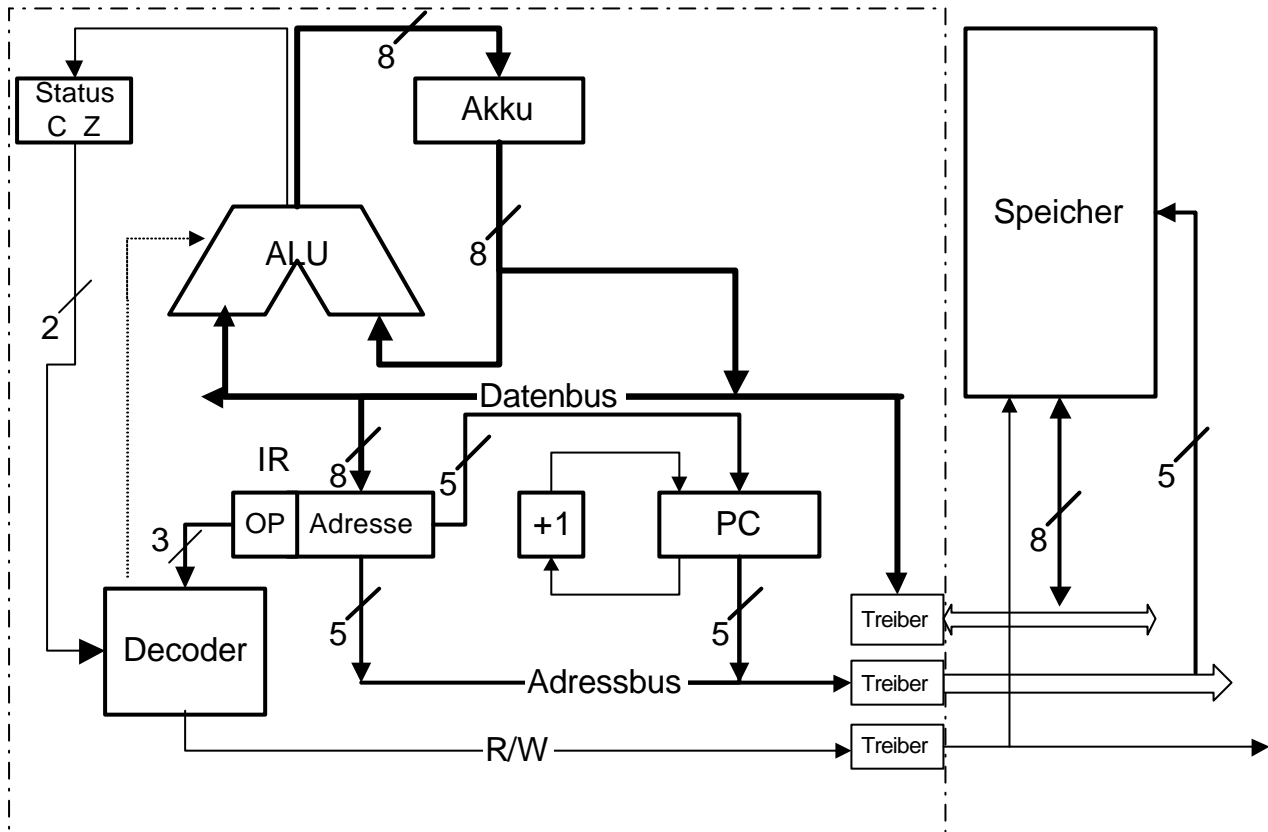
Entwicklungsgeschichte der Mikroprozessoren

Jahr	µP-Typ	
		1. Generation
1971	Intel 4004	4-Bit ALU, 16x4Bit Register, 12 Bit Adressbus, 45 Befehle, 2250 Transistoren
1972	Intel 8008	8-Bit ALU, 6x8 Bit Register, 14 Bit Adressbus, 30 µsec/Befehl,
		2. Generation
1972	Intel 8080	8-Bit Prozessor, 16 Bit Adressbus, 2µsec/Befehl, 75 Befehle, 5000 Transistoren, Interruptfähig
1974	Motorola 6800	
		3. Generation
1976	Intel 8085 Zilog Z80	Weiterentwicklung des 8080
1978	Intel 8086	16-Bit Prozessor, 29000 Transistoren, 20 Bit Adressbus
	Intel 8048	1. Ein-Chip Computer
1979	Motorola 68000	32 Bit Register, 16 MB Adressbereich, 68000 Transistoren
	Z8000	
	Intel 8088 Motorola 6809	8 Bit Datenbus, intern 16 Bit
1982	Intel 80286	
		4. Generation
1984	Intel 80386 Motorola 68020 NS 32332	32-Bit Prozessoren, 275000 Transistoren, virtuelle Speicherverwaltung integriert
1989 1990	Intel 80486 Motorola 68040	1,2 Mio Transistoren, FPU integriert
1993	Intel Pentium	3,1 Mio Transistoren, 16 Kbyte Cache, 64/32-Bit Prozessor
	DEC Alpha Chip	64 Bit RISC-Chip, 2,8 Mio Transistoren
	IBM/Motorola/ Apple: PowerPC 601	32/64 Bit RISC Chip, 2,8 Mio Transistoren

Blockschaltbild



Einfachste Prozessorarchitektur mit Akku



- PC Program Counter
- Akku Akkumulator
- OP Operation
- ALU Arithmetic Logic Unit
- IR Instruction Register
- C Carry
- Z Zero

Befehlsliste:

OpCode	Dezimal	Mnemonic
000	0	HALT
001	1	LOAD mem
010	2	STORE mem
011	3	ADD mem
100	4	SUB mem
101	5	JMP mem
110	6	JMPC mem
111	7	JMPZ mem

Realisierung des Prozessors als digitale Schaltung

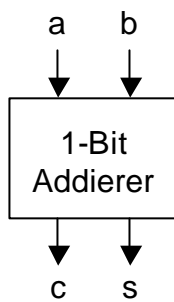
Realisierung der ALU

Befehlssatz der ALU besteht aus :

- Addition (ADD mem)
- Subtraktion (SUB mem)
- Laden (LOAD mem)

Die ALU wird als 1-Bit-ALU realisiert

Kernstück der 1-Bit –ALU ist ein Addierer

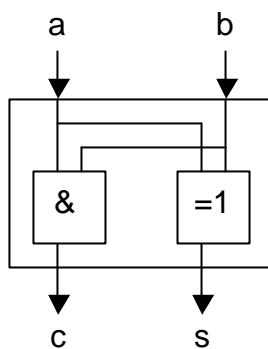


a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



$$S = a \text{ XOR } b$$

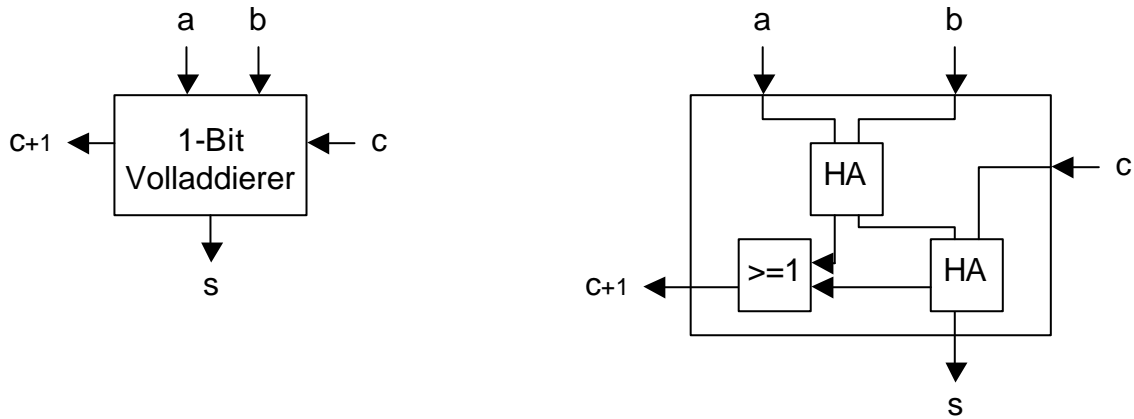
$$C = a \text{ AND } b$$



1-Bit
Halb-Addierer

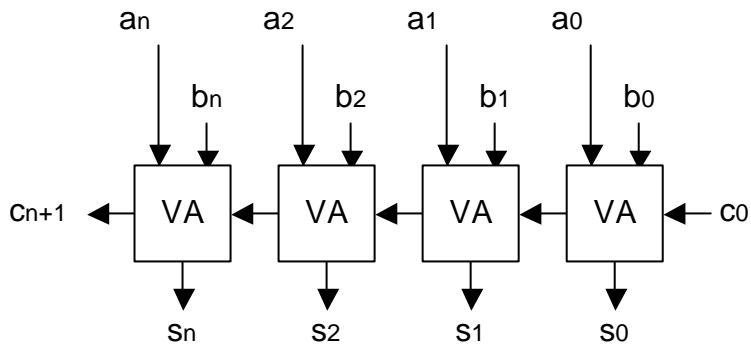


Volladdierer: Berücksichtigung des Übertrags aus der vorigen Stelle

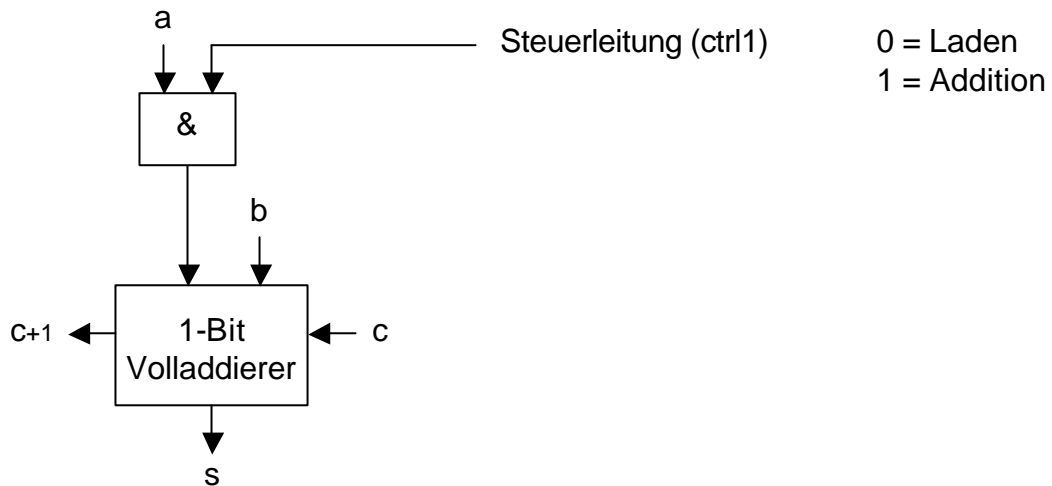


Wird aufgebaut aus 2 Halbaddierern, die Oderverknüpfung der carry`s ergibt das carry der nächsten Stelle

Addition mehrerer Bit (z.B. 1 Byte) wird durch Parallelschaltung mehrerer 1-Bit Addierer erreicht

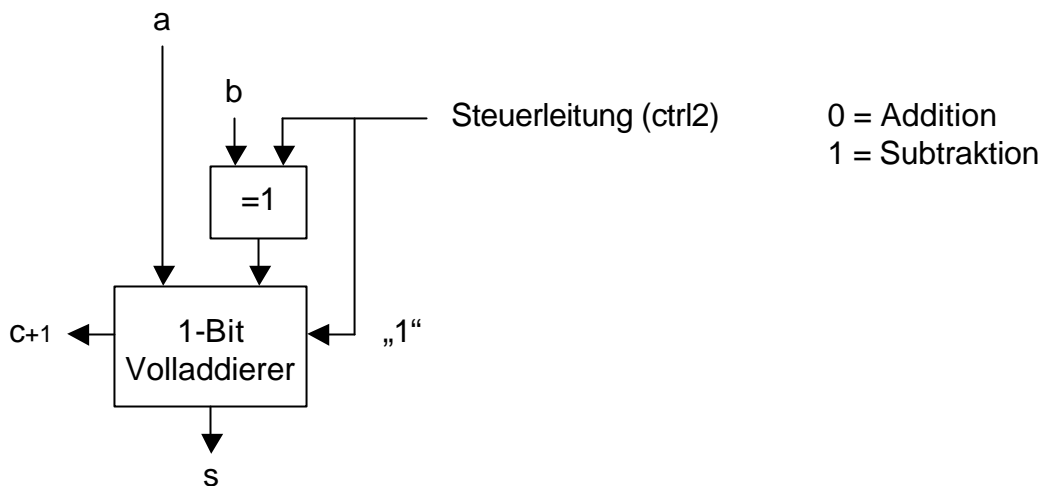


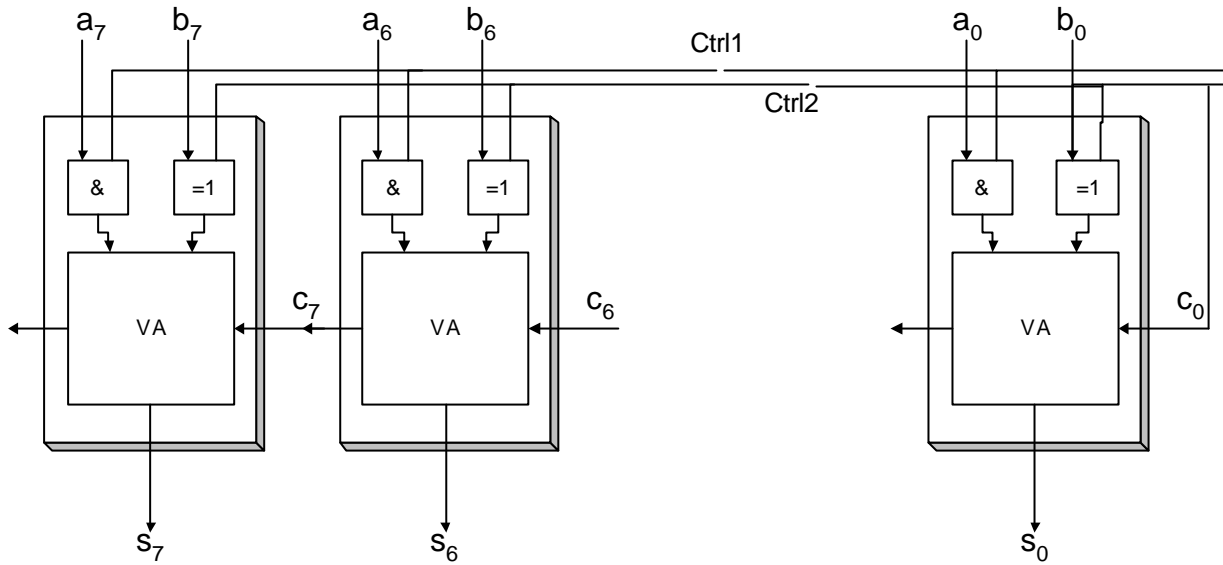
Load mem: Laden des Akku mit dem Inhalt einer Adresse
=> Addition einer 0



Sub mem: Subtraktion

=> Addition des 2er Komplements
2er Komplement : Invertieren (Exor) und Addition einer 1



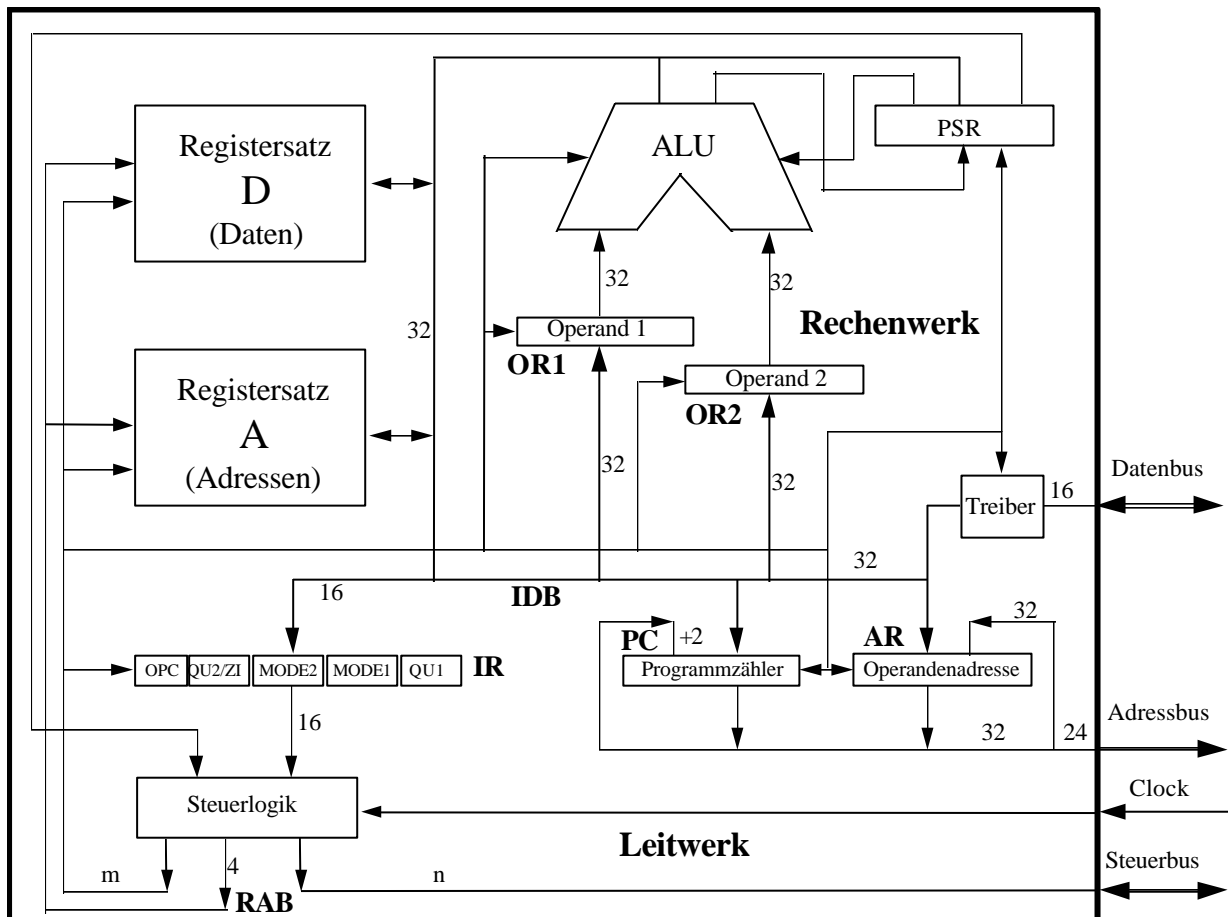


LOAD	Ctrl1 = 0	Ctrl2 = 0
ADD	Ctrl1 = 1	Ctrl2 = 0
SUB	Ctrl1 = 1	Ctrl2 = 1



Struktur eines 16/32 Bit μ P

Um den Datenfluß bei der Befehlsbearbeitung innerhalb eines μ P zu betrachten, gehen wir von einem Modell eines 16/32 Bit μ aus. Das Modell ist dem Prozessor MC 68000 von Motorola nachempfunden.



- | | |
|------------|---|
| PSR | Prozessorstatusregister |
| | <ul style="list-style-type: none"> • Zero-Flag Ergebnis = 0 • Negativ-Flag Ergebnis < 0 • Carry-Flag Übertrag in die nächste Stelle • Overflow-Flag Bereichsüberschreitung |
| PC | Programmzähler (Programcounter) |
| | enthält die Adresse des nächsten Befehls |
| IR | Instruktionsregister enthält das Befehlswort |
| DBT | Datenbustreiber |
| IDB | Interner Datenbus |
| IAR | Zwischenspeicher für Operanden |