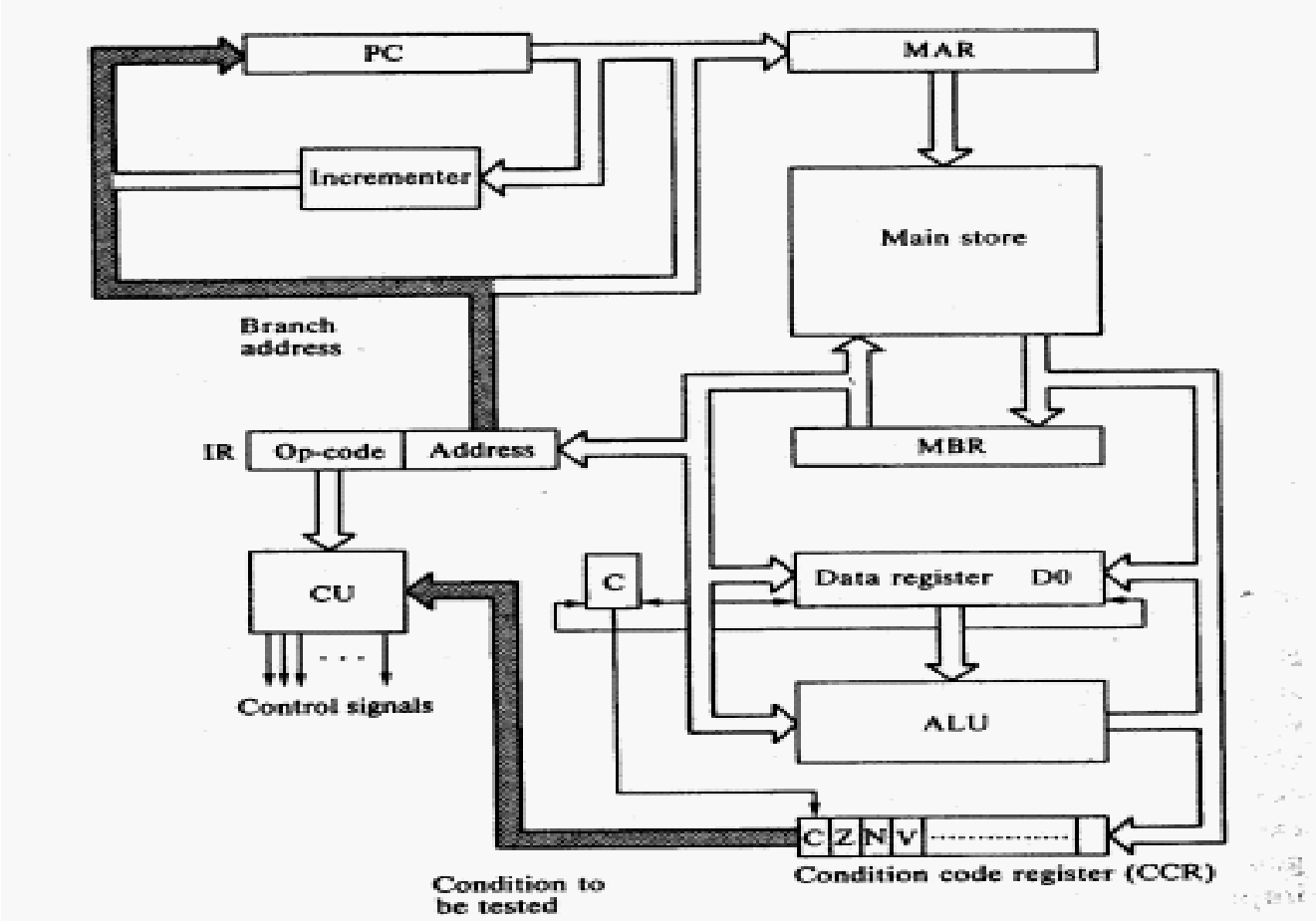
Prinzipielle Struktur der CPU



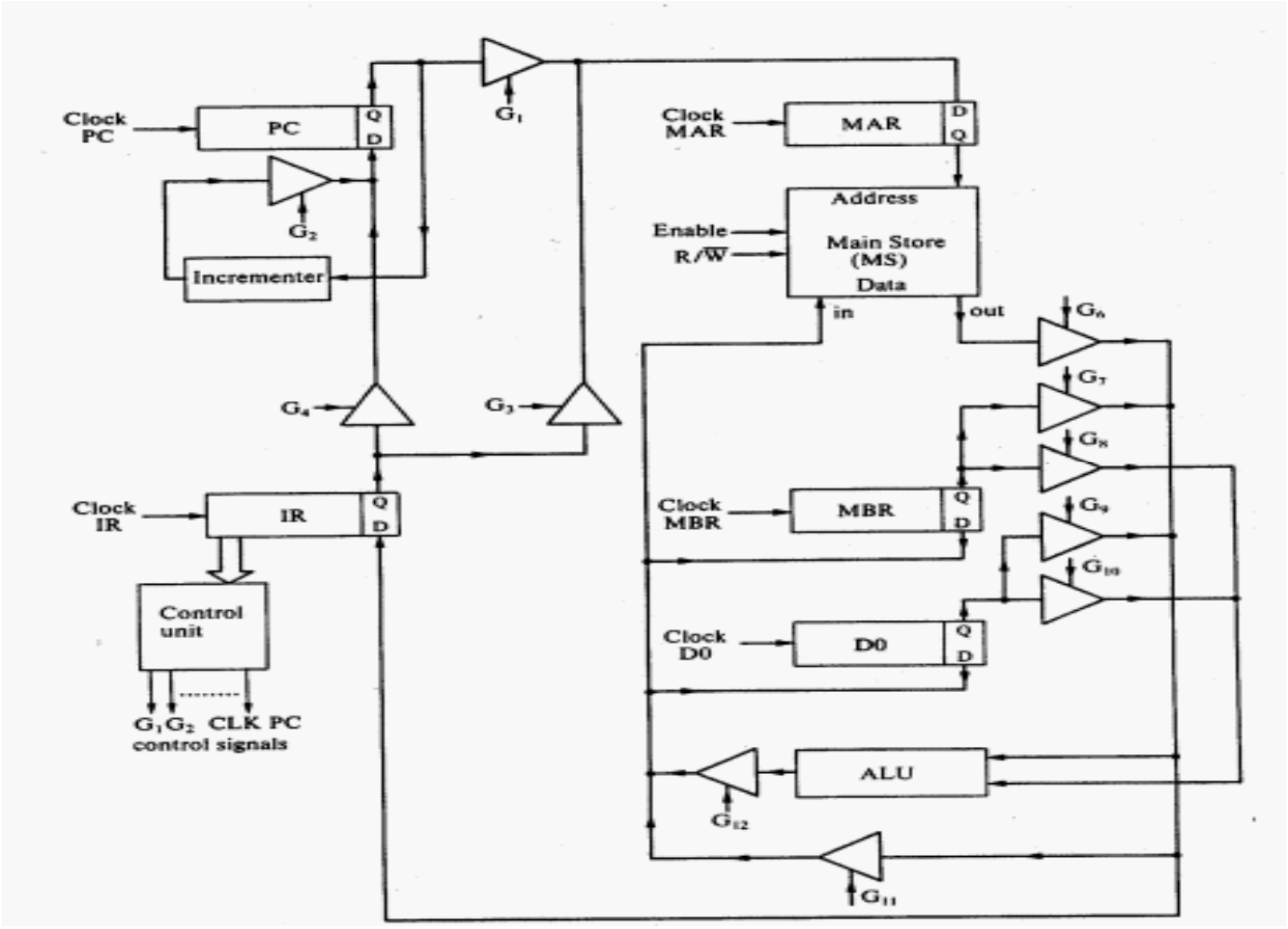
Operand 1 + Operand 2 = Result	CCR status bits
00000011 + 00000100 = 00000111	C=0, Z=0, N=0, V=0
11111111 + 00000001 = 00000000	C=1, Z=1, N=0, V=0
01100110 + 00110010 = 10011000	C=0, Z=0, N=1, V=1
11001001 + 10100000 = 01101001	C=1, Z=0, N=0, V=1

Struktur der CPU (6)

Die Kontrolleinheit (CU):

- Mikroprogrammsteuerwerk
- Steuerschaltnetz

Funktionsweise der CU am Beispiel einer einfachen CPU:



Struktur der CPU (7)

Übersetzung eines HOLEN-AUSFÜHREN-Zyklus in RTL:

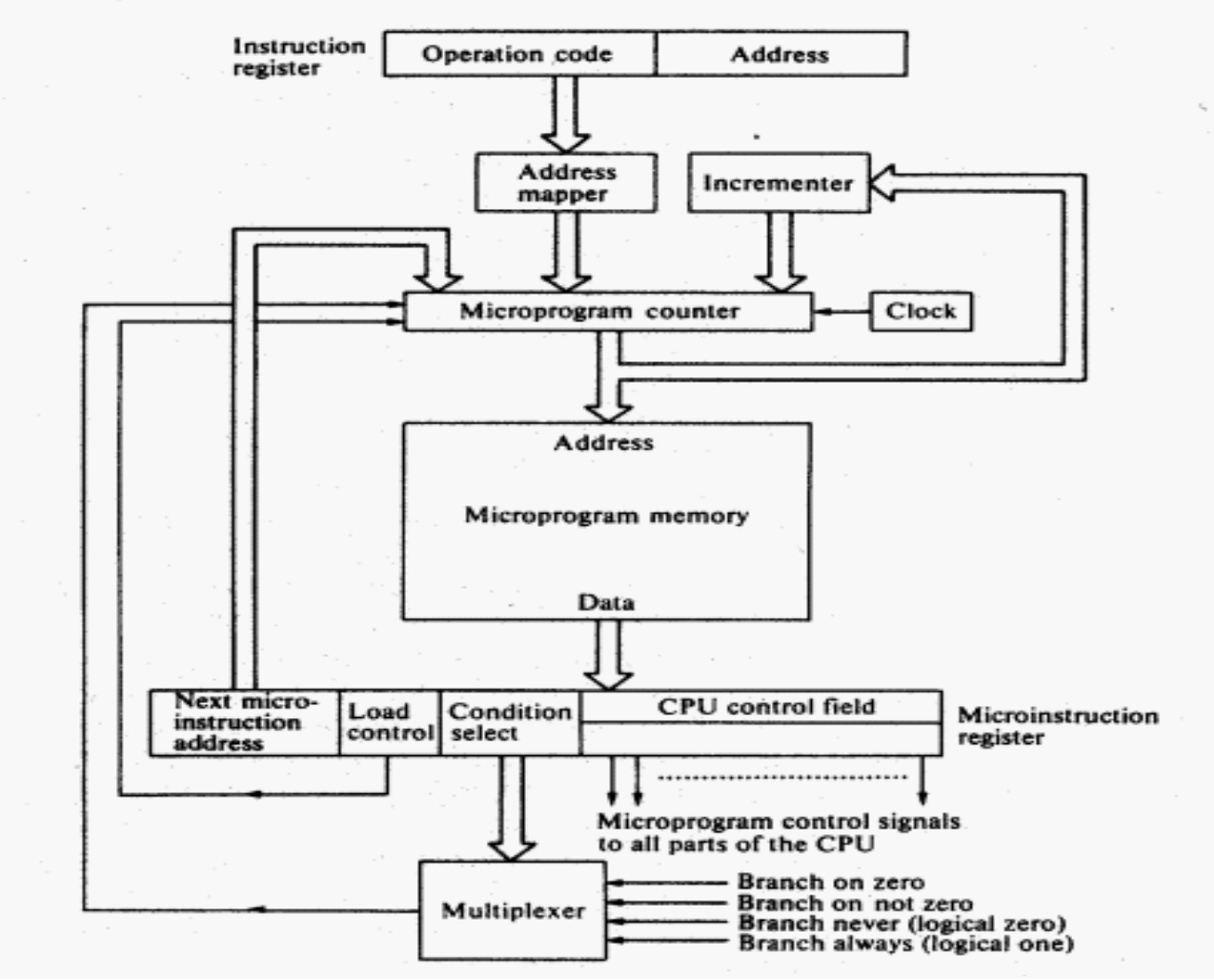
Step	Register-transfer language	Operations required
1	[MAR] ← [PC]	enable G_1, clock MAR
1a	INC ← [PC]	
2	[PC] ← INC	enable G_2, clock PC
3	[MBR] ← [MS([MAR])]	enable MS, R/$\overline{W}$=1, enable G_6 enable G_{11}, clock MBR
4	[IR] ← [MBR]	enable G_7, clock IR
4a	CU ← [IR(op-code)]	
5	[MAR] ← [IR(address)]	enable G_3, clock MAR
6	[MBR] ← [MS([MAR])]	enable MS, R/$\overline{W}$=1, enable G_6, enable G_{11}, clock MBR
7	ALU ← [MBR]	enable G_7
7a	ALU ← [D0]	enable G_{10}
8	[D0] ← ALU	enable G_{12}, clock data register

Steuersignale während der HOLEN und AUSFÜHREN Phase eines arithm. Befehls:

Step	Gate control signals												MS control		Register clocks				
	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	ENABLE	R/$\overline{W}$	PC	MAR	MBR	D0	IR
1	1	0	0	0	0	0	0	0	0	0	0	0	0	X	0	1	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	X	1	0	0	0	0
3	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	1	0	0
4	0	0	0	0	0	0	1	0	0	0	0	0	0	X	0	0	0	0	1
5	0	0	1	0	0	0	0	0	0	0	0	0	0	X	0	1	0	0	0
6	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	1	0	0
7	0	0	0	0	0	0	1	0	0	1	0	0	0	X	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	1	0	X	0	0	0	1	0

Struktur der CPU (8)

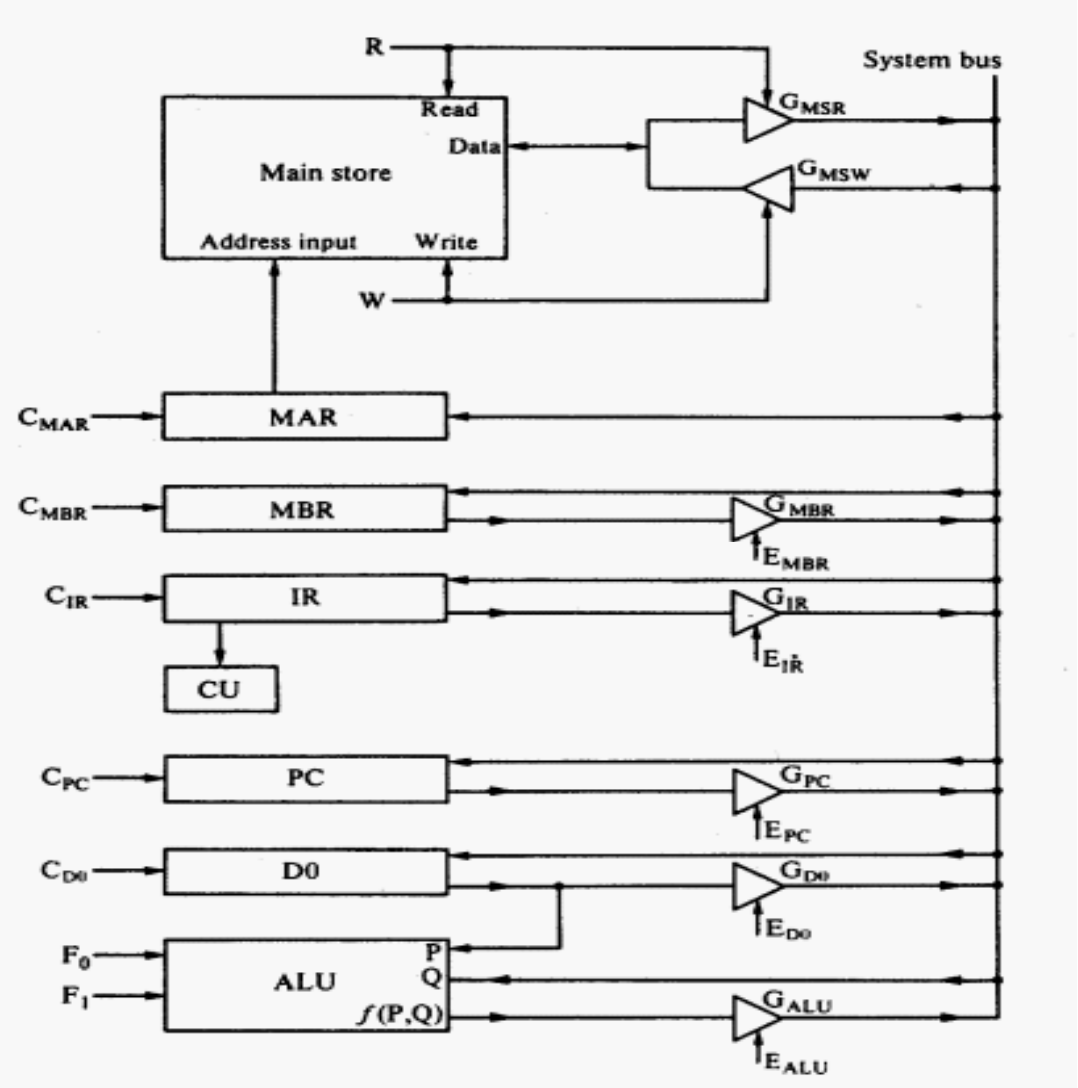
Die mikroprogrammierbare Kontrolleinheit:



Struktur der CPU (9)

Die fest verdrahtete Kontrolleinheit (random logic CU)

Funktionsweise an Hand der Struktur einer noch einfacheren CPU:



Struktur der CPU (10)

Dekodierung des Kontrollcodes der ALU:

trol code, F_0,F_1

F_1	F_0	Function
0	0	add P to Q
0	1	subtract Q from P
1	0	increment Q
1	1	decrement Q

Ein einfacher Befehlssatz für die CPU:

Op-code	Mnemonic		Operation
000	LOAD	M	[D0]←[MS(M)]
001	STORE	M	[MS(M)]←[D0]
010	ADD	M	[D0]←[D0]+[MS(M)]
011	SUB	M	[D0]←[D0]−[MS(M)]
100	INC	M	[MS(M)]←[MS(M)]+1
101	DEC	M	[MS(M)]←[MS(M)]−1
110	BRA	M	[PC]←M
111	BEQ	M	IF Z=1 THEN [PC]←M

Note that M defines the memory location used by the instruction.

Struktur der CPU (11)

Interpretation des Befehlssatzes:

Instruction	Op-code	Operations (RTL)	Control actions
Fetch	—	[MAR]←[PC] [IR]←[MS([MAR])] ALU←[PC] [PC]←ALU	$E_{PC}=1$, C_{MAR} $R=1$, C_{IR} $E_{PC}=1$, $F_1,F_0=1,0$ $E_{ALU}=1$, C_{PC}
LOAD	000	[MAR]←[IR] [D0]←[MS([MAR])]	$E_{IR}=1$, C_{MAR} $R=1$, C_{D0}
STORE	001	[MAR]←[IR] [MS([MAR])]←[D0]	$E_{IR}=1$, C_{MAR} $E_{D0}=1$, $W=1$
ADD	010	[MAR]←[IR] [MBR]←[MS([MAR])] ALU←[MBR] [D0]←ALU	$E_{IR}=1$, C_{MAR} $R=1$, C_{MBR} $E_{MBR}=1$, $F_1,F_0=0,0$ $E_{ALU}=1$, C_{D0}
SUB	011	[MAR]←[IR] [MBR]←[MS([MAR])] ALU←[MBR] [D0]←ALU	$E_{IR}=1$, C_{MAR} $R=1$, C_{MBR} $E_{MBR}=1$, $F_1,F_0=0,1$ $E_{ALU}=1$, C_{D0}
INC	100	[MAR]←[IR] [MBR]←[MS([MAR])] [ALU]←[MBR] [MBR]←ALU [MS([MAR])]←[MBR]	$E_{IR}=1$, C_{MAR} $R=1$, C_{MBR} $E_{MBR}=1$, $F_1,F_0=1,0$ $E_{ALU}=1$, C_{MBR} $E_{MBR}=1$, $W=1$
DEC	101	[MAR]←[IR] [MBR]←[MS([MAR])] ALU←[MBR] [MBR]←ALU [MS([MAR])]←[MBR]	$E_{IR}=1$, C_{MAR} $R=1$, C_{MBR} $E_{MBR}=1$, $F_1,F_0=1,1$ $E_{ALU}=1$, C_{MBR} $E_{MBR}=1$, $W=1$
BRA	110	[PC]←[IR]	$E_{IR}=1$, C_{PC}
BEQ	111	IF Z=1 THEN [PC]←[IR]	IF Z=1 THEN $E_{IR}=1$, C_{PC}