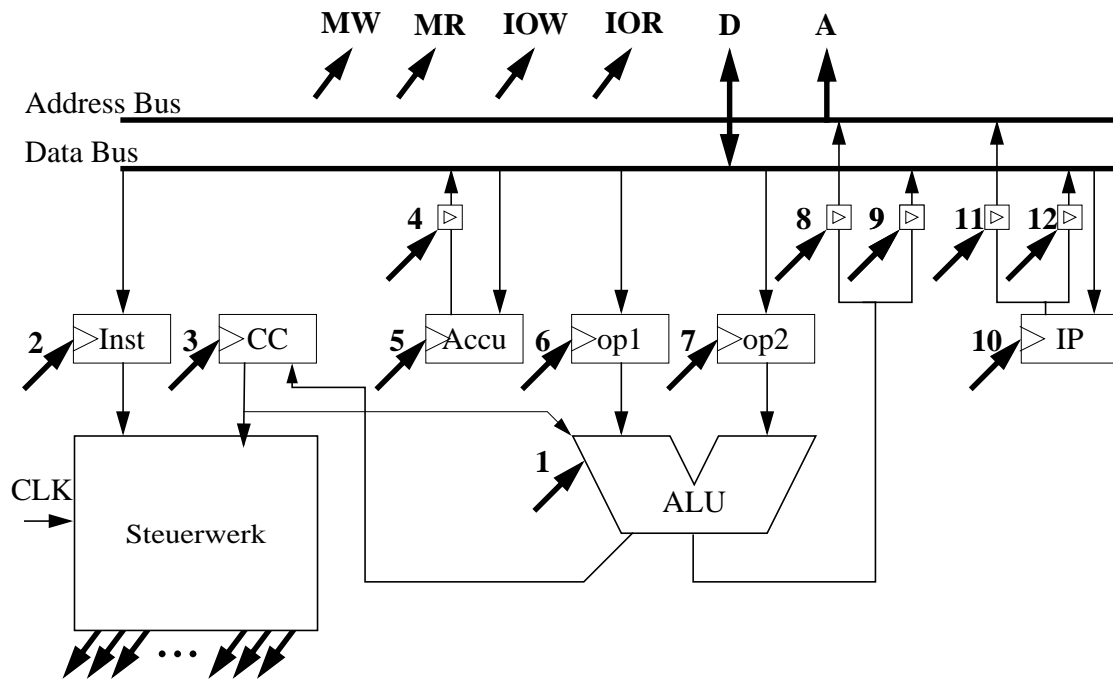


CPU (Struktur)



CPU-Steuerwerk (Beispiel)

Soll im Beispiel der Befehl ausgeführt werden, dessen Adresse im PC steht, laufen folgende Schritte ab:

1.) *Holen der Instruktion*

- Öffnen Tor 11 und Aktivieren des 'M'emory-'R'ead-Signals
- Übernahme der Daten des Datenbusses in das Instruktionsregister (2)
- Schliessen Tor 11 und Deaktivieren des 'M'emory-'R'ead-Signals

Für dieses Beispiel enthalte das Instruktionsregister jetzt die Instruktion ADD \$..., %ACCU

2.) *Inkrementieren des Program-Counters*

- Öffnen Tor 12
- Übernahme der Daten des Datenbusses in das Operandenregister op1 (6)
- Schliessen Tor 12, Signalisierung 'Increment' an 1, Öffnen Tor 9
- Übernahme der ALU-Daten in den Instruction-Pointer (10)
- Schliessen Tor 9

CPU-Steuerwerk (Beispiel)

3.) *Holen des Immediate-Operand*

- Öffnen Tor 11 und Aktivieren des 'M'emory-'R'ead-Signals
- Übernahme der Daten des Datenbusses in das Operandenregister op2 (7)
- Schliessen Tor 11 und Deaktivieren des 'M'emory-'R'ead-Signals

4.) *Nochmaliges Inkrementieren des Program-Counters*

- Öffnen Tor 12
- Übernahme der Daten des Datenbusses in das Operandenregister op1 (6)
- Schliessen Tor 12, Signalisierung 'Increment' an 1, Öffnen Tor 9
- Übernahme der ALU-Daten in den Program-Counter (10)
- Schliessen Tor 9

CPU-Steuerwerk (Beispiel)

5.) *Durchführen der Operation*

- Öffnen Tor 4
- Übernahme der Daten des Datenbusses in das Operandenregister op1 (6)
- Schliessen Tor 4, Signalisieren 'ADD' an 1, Öffnen Tor 9
- Übernahme der Daten des Datenbusses in den ACCU (5) und der Condition Codes (3)
- Schliessen Tor 9

6.) *nächsten Befehl holen und ausführen*

CPU-Steuerwerk

Aufgabe des Steuerwerkes ist es,

- entsprechend der Werte in 'Inst' bzw. 'CC'

in einer zeitlichen Abfolge (CLK)

- die Instruktionen an die ALU (1)
- die Signale für die Tore und Register (2-12),
- die Read- (MR) und Write-Signale (MW) an den Speicher
- die Read- (IOR) und Write-Signale (IOW) an die I/O-Geräte

zu generieren.

=> Steuerwerk ist programmierter Automat

CPU-Steuerwerk

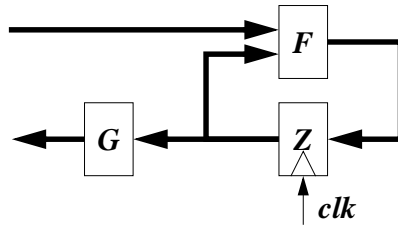
*Steuerwerk ist (binär) programmierter Automat, der in der Lage ist,
(binär kodierte) Hochsprachenprogramme abzuarbeiten.*

?!?

Steuerwerk ist allgemeingültig. Es kann alle Hochsprachenprogramme abarbeiten!

CPU-Steuerwerk

Moore-Automat:



Die Umsetzung

- Zustand => Nachfolgezustand (Funktion F)
- Zustand => Signale für Tore, Register, ALU und Speicher (Funktion G)

geschieht über

- Kombinatorik
- Tabelle (ROM)

CPU-Steuerwerk (1. Beispiel)

Inst	CC	State	Next	ALU	Signale f. Tore, Register, ...	
----	--	0	1	-	MR, 11	Holen der Instruktion
----	--	1	2	-	MR, 2, 11	
----	--	2	3	-	12	Inkrementieren des IP
----	--	3	4	-	6, 12	
----	--	4	5	Inc	9	
----	--	5	6	Inc	9, 10	
ADD\$	--	6	7	-	MR, 11	Holen Immediate Operand
ADD\$	--	7	8	-	MR, 7, 11	
ADD\$	--	8	9	-	12	Inkrementieren des IP
ADD\$	--	9	10	-	6, 12	
ADD\$	--	10	11	Inc	9	
ADD\$	--	11	12	Inc	9, 10	
ADD\$	--	12	13	-	4	Durchführen der Operation
ADD\$	--	13	14	-	4, 6	
ADD\$	--	14	15	Add	9	
ADD\$	--	15	0	Add	3, 5, 9	

CPU-Steuerwerk (2. Beispiel)

Inst	CC	State	Next	ALU	Signale f. Tore, Register, ...
...			...		Holen der Instruktion
...			...		Inkrementieren des PC
JE	--	6	7	-	R, 11
JE	--	7	8	-	R, 7, 11
JE	--	8	9	-	12
JE	--	9	10	-	6, 12
JE	0-	10	11	Inc	9
JE	0-	11	0	Inc	9, 10
JE	1-	10	11	Add	9
JE	1-	11	0	Add	9, 10

CPU (Ansteuerung der Tore, Register, ...)

Möglichkeiten:

die Tabelle zur Berechnung der Signale für die Tore, Register, ALU und den Speicher enthält

- für jedes Signal genau einen Eintrag (horizontales Steuerwort)

000000000010 => nur Tor 1 geöffnet; alle anderen geschlossen

000001101000 => Tor 3, 5, 6 gleichzeitig geöffnet

- $\log_2 N$ Einträge für alle N Signale (vertikales Steuerwort)

0010 => nur Tor 2 geöffnet

0111 => nur Tor 7 geöffnet

- Mischform: (quasi-horizontales Steuerwort)

Gruppen von Signalen jeweils zusammengefasst und vertikal codiert

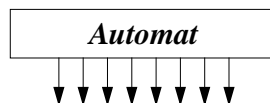
alle Gruppen zusammengefasst und horizontal codiert

100 010 => Tor 2 (Gruppe 0) und Tor 4 (Gruppe 1) gleichzeitig geöffnet

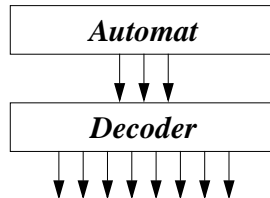
111 011 => Tor 3 (Gruppe 0) und Tor 7 (Gruppe 1) gleichzeitig geöffnet

CPU (Ansteuerung der Tore, Register, ...)

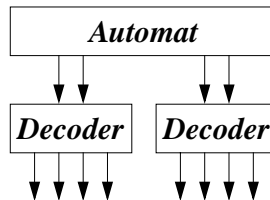
horizontales Steuerwort:



vertikales Steuerwort:



quasi-horizontales Steuerwort:



CPU (Ansteuerung der Tore, Register, ...)

horizontales Steuerwort

hoher Speicheraufwand; mehrere Tore können gleichzeitig geöffnet werden

vertikales Steuerwort

geringer Speicheraufwand; es kann jeweils nur ein einzelnes Tor geöffnet werden
teurer Dekodierer

quasi-horizontales Steuerwort

mittlerer Speicheraufwand; Tore können z.T. gleichzeitig geöffnet werden
im Beispiel dürfen Tor 2, 4, 9, 12 nicht gleichzeitig geöffnet werden
=> eine Gruppe
Tor 8, 11 unabhängig von Tor 2, 4, 9, 12
=> verschiedene Gruppen

CPU (Struktur)

Verbesserungsmöglichkeiten:

- mehr Register
- breitere Busse (z.B. 32-Bit => 64-Bit)
- Extra-Adress-Rechenwerk
(z.B. gleichzeitige Inkrementierung IP und Ausführung ADD)
- zusätzliche Busse
(z.B. gleichzeitige Ausführung ADD und Holen der nächsten Instruktion)
- zusätzliche Rechenwerke (Super-Scalar)
(gleichzeitige Ausführung mehrerer Berechnungen)
- ...

=> selbst ein (einfacher) Mikro-Controller (z.B. m68x05) schafft ADD \$. . . in 2 Takten!