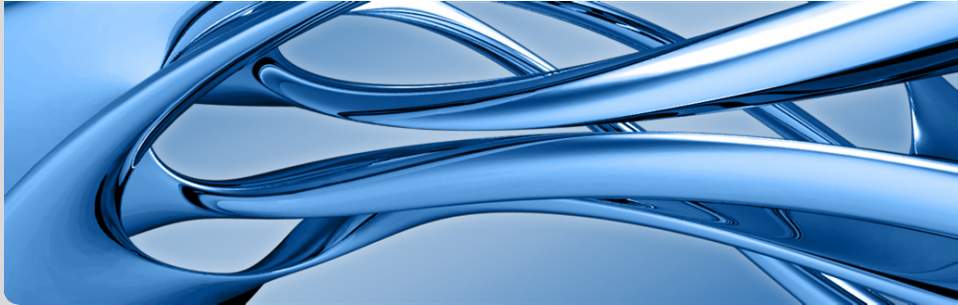


2. Tutorium Rechnerorganisation

Tutorium 7 | WS19/20

Grégoire Mercier | 17. November 2019

INSTITUT FÜR TECHNISCHE INFORMATIK



- 1 Komponenten eines Von-Neumann-Rechners
- 2 Einführung in MIMA
- 3 Mikroprogrammierung

- Von-Neumann-Architektur bezieht sich auf Rechner, wir beschäftigen uns hier aber zuerst einmal nur mit der CPU
- Besteht aus verschiedenen Komponenten:
 - Steuerwerk
 - Rechenwerk
 - Speicherwerk
 - Registersatz
 - Systembusschnittstelle
- Programme und Daten liegen im selben Speicher
 - Müssen über den gleichen Speicherbus angesprochen werden (Von-Neumann-Flaschenhals)
 - Alternative: Harvard-Architektur mit getrennten Speicher

- Für die Abarbeitung des Programms zuständig
 - Lädt den auszuführenden Befehl in das Befehlsregister (Holphase)
 - Dekodiert den Befehl im Befehlsregister (Dekodierphase)
 - Führt den dekodierten Befehl unter Verwendung der anderen CPU-Komponenten aus (Ausführungsphase)
- Steuerregister beeinflusst Verhalten des Steuerwerks
- Fest verdrahtete Logik vs. Mikroprogrammierung

- ALU = „arithmetic logic unit“
- Führt auf Anweisung des Steuerwerks arithmetisch-logische Operationen aus
- Statusregister speichert zusätzliche Informationen über die letzte Berechnung, z.B.
 - Negative Flag
 - Carry Flag
 - Overflow Flag

- Beinhaltet in Von-Neumann-Architektur eigentlich auch den Speicher selbst, wir betrachten erstmal nur die CPU
- Regelt den Zugriff auf den Hauptspeicher, der Programme und Daten enthält
- Bei vielen modernen Prozessoren zusätzliche Aufgabenbereiche für MMU ("memory management unit") des Prozessors, z.B.
 - Zugriffsschutz
 - Segmentierung
 - Paging (Kachelverwaltung)

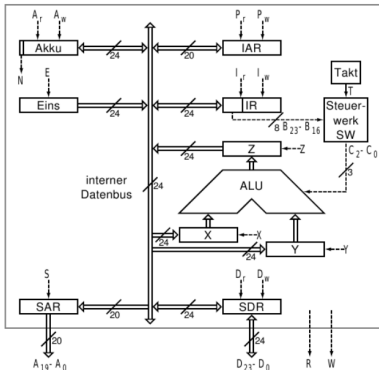
- Enthält die Register des Prozessors
- Spezialregister vs. allgemein verwendbare (general purpose) Register

- Erlaubt dem Prozessor, über den Systembus mit anderen Rechnerkomponenten zu kommunizieren
- Bestehend aus:
 - Steuerbus (unidirektional): Wahl des Betriebsmodus
 - Adressbus (unidirektional): Wahl der Adresse
 - Datenbus (bidirektional): Übergabe des Datenworts
- Beispiele:
 - „Schreibe den Wert 10 bei Adresse 20“ (CPU → RAM)
 - „Lies den Wert bei Adresse 20“ (CPU → RAM) - „10“ (RAM → CPU)

- Mikroprogrammierte Minimalmaschine
- Sehr einfaches Modell für eine CPU mit Von-Neumann-Architektur
- Wurde im Rahmen der Rechnerorganisation-Vorlesung entwickelt und wird dort vorgestellt
- Folgende Spezifikationen gibt es als „Merkblatt“ auch nochmal an das 4. Übungsblatt angehängt und auf der Vorlesungshomepage zum Download (als mima.pdf)

Die Architektur der MIMA

Architektur der MIMA



Register

Akku: Akkumulator
X: 1. ALU Operand
Y: 2. ALU Operand
Z: ALU Ergebnis
Eins: Konstante 1
IAR: Instruktionsadreibregister
IR: Instruktionsregister
SAR: Speicheradreibregister
SDR: Speicherdatenregister

Steuersignale vom SW

– für den internen Datenbus

A_r : Akku liest
 A_w : Akku schreibt
 X_r : X-Register liest
 Y_r : Y-Register liest
 Z_r : Z-Register schreibt
 E : Eins-Register schreibt
 I_r : IAR liest
 I_w : IAR schreibt
 I_r : IR liest
 I_w : IR schreibt
 D_r : SDR liest
 D_w : SDR schreibt
 S_r : SAR liest

– für die ALU

C_2 - C_0 : Operation auswählen

– für den Speicher

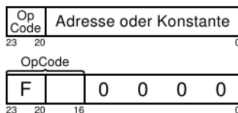
R : Leseanforderung
 W : Schreib Anforderung

Meldesignale zum SW

T : Takteingang
 N : Vorzeichen des Akku
 $B_{23} \dots B_{16}$: OpCode-Feld im IR

OpCode	Mnemonic	Beschreibung
0	LDC c	c -> Akku
1	LDV a	<a> -> Akku
2	STV a	Akku -> <a>
3	ADD a	Akku + <a> -> Akku
4	AND a	Akku AND <a> -> Akku
5	OR a	Akku OR <a> -> Akku
6	XOR a	Akku XOR <a> -> Akku
7	EQL a	falls Akku = <a>: -1 -> Akku sonst : 0 -> Akku
8	JMP a	a -> IAR
9	JMN a	falls Akku < 0 : a -> IAR
F0	HALT	stoppt die MIMA
F1	NOT	bilde Eins-Komplement von Akku -> Akku
F2	RAR	rotiere Akku eins nach rechts -> Akku

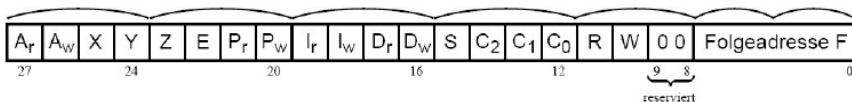
Befehlsformate



ALU-Operationen der MIMA

$c_2 c_1 c_0$	ALU Operation
0 0 0	tue nichts (d.h. $Z \rightarrow Z$)
0 0 1	$X + Y \rightarrow Z$
0 1 0	rotiere X nach rechts $\rightarrow Z$
0 1 1	$X \text{ AND } Y \rightarrow Z$
1 0 0	$X \text{ OR } Y \rightarrow Z$
1 0 1	$X \text{ XOR } Y \rightarrow Z$
1 1 0	Eins-Komplement von X $\rightarrow Z$
1 1 1	falls $X = Y$, $-1 \rightarrow Z$, sonst $0 \rightarrow Z$

Mikrobefehlsformat der MIMA



- Das MIMA-Steuerwerk ist mikroprogrammiert
- \Rightarrow Ein Maschinenbefehl wird durch ein kleines Programm von Mikrobefehlen ausgeführt
- Jeder dieser Mikrobefehle gibt an, welche der Steuerleitungen in einem Taktzyklus aktiv sind
- Register-Transfer-Anweisungen als alternative Schreibweise:
 - $A_w = 1$, $P_r = 1$ (Steuersignale)
 - Akku \rightarrow IAR (Register-Transfer)

Übungsaufgabe 1: Mikrocode 1

Geben Sie das Mikroprogramm für die Holphase und Ausführungsphase des MIMA-Befehls

OR a # akku OR <a> => Akku

- in Register-Transfer-Anweisung Schreibweise
- in binärer Schreibweise (nur Takte 1 und 7)

an.

Das bisher reservierte 8. Bit im Mikrobefehlsformat der MIMA sei mit D bezeichnet und wird als Kennzeichen dafür verwendet, dass die Adresse des nächsten Mikrobefehls aus dem Befehlsteil B 23 - B 16 ermittelt werden muss. Wie sieht dann der Mikrobefehl für die Dekodierung (6. Takt) aus?

Übungsaufgabe 3: Mikrocode 2

Schreiben Sie ein Programm in Register-Transfer-Schreibweise, das den Inhalt des Akkumulators als Zweierkomplement wieder im Akkumulator speichert. Beginnen Sie dabei ab Takt 7 der Befehlsabarbeitung.