

Datamaskiner Grunnkurs

Kristoffer Dalby

1 Forkortelser

Forkortelse	Betydning
PC	Program Counter
MPC	Micro Program Counter
MBR	Memory Buffer Register
IR	Instruction Register
MAR	Memory Address Register
MDR	Memory Data Register
LV	Local Variable
SP	Stack Pointer
CPP	Constant Pool Pointer
TOS	Top Of Stack
OPC	OpCode
H	Holding Register
ALU	Arithmetic Logic Unit
RAW	Read-After-Write
WAW	Write-After-Write
WAR	Write-After-Read
SRAM	Static RAM
SDRAM	Synchronous Dynamic RAM
DRAM	Dynamic RAM
CMP	Chip-level MultiProcessor
ILP	Instruction-Level Parallelism
RISC	Reduced Instruction Set Computer
CISC	Complex Instruction Set Computer

2 Registre

2.1 PC

Program counter peker på den instruksjonen i programmet som blir jobbet med.

2.2 MPC

Micro Program Counter peker på microinstruksjonen i 'control store'. Den peker på den første micro instruksjonen i den aktive instruksjonen. MPC holder orden på hva den neste micro instruksjonen er.

2.3 MBR

Memory Buffer Register er et bufferregister mellom minne og prosessor.

2.4 IR

Instruction Register er der kontrollenheten lagrer instruksjonen som blir gjennomført nå. Den ligger her mens instruksjonen blir dekodet, startet og gjennomført.

2.5 MAR

Memory Address Register inneholder adresse til neste minnelokasjon der vi finner neste instruksjon.

2.6 MDR

Memory Data Register inneholder data som skal bli lagret i hovedminne/RAM, eller data som har blitt hentet fra minne. Det fungerer som en buffer så data er klar for prosessoren.

2.7 LV

Local Variable inneholder pekerverdi.

2.8 SP

Stack Pointer inneholder pekerverdi mot element i stacken.

2.9 CPP

Constant Pool Pointer inneholder pekerverdi.

2.10 TOS

Top Of Stack skal alltid inneholde elementet på toppen av stakken

2.11 OPC

OpCode register kan fritt brukes.

2.12 H

Holding Register inneholder verdien som skal inn i A-inngangen til ALU.

3 ALU-flagg

Flagg	Årsak
N	Når svar er negativt
Z	Når svaret er 0
C	Carry, når vi legger sammen uten fortegnsbit
V	Overflow

4 Von Neumann

Data og instruksjoner er lagret i samme minne og beregninger skjer sekvensielt. Von Neumann-arkitektur gjør det mulig å skrive program som kan endre sin egen programkode.

- ALU - Aritmetisk og logisk enhet som utfører beregningene.
- Control Unit - Kontrollenhet som dekoder instruksjoner og gjennomfører dem.
Kontrollenheten kan enten være hard-wired eller inneholde kode styrt av en mikrokontroller.
- Memory - Primærminnet (RAM) inneholder data og instruksjoner.
- I/O - Enheter for inn- og utdata.

Moderne datamaskiner har ALU og kontrollenhet på prosessor (CPU), benytter seg av registre, hurtigbuffere, busser og millioner av transistorer, men konseptet er veldig likt. Overføring av data mellom minne og CPU blir i dag sett på som et av de største problemene med Von Neumann-arkitektur.

5 MicroInstruction Register

- Addr - Peker på neste mikroinstruksjon i instruksjonen.
- J - Sier ifra om ALU har flagget neste mikroinstruksjon eller om det kommer hopp (betinget hopp).
- ALU - bestemmer hvilken funksjon ALU skal gjennomføre.
- C - Inneholder adressen til C-bussen, som blir det samme som adressen til registeret det skal skrives til.
- Mem - Sier ifra om det skal gjøres noe med minne.
- B - inneholder adressen til B-bussen, som blir det samme som adressen til registeret det skal leses fra.

6 Superskalar CPU

En superskalar prosessor implementerer en form for parallelitet som kalles instruksjonsnivåparallelitet. Dette betyr at den kan utføre flere instruksjoner pr. klokkesyklus (dupliserer CPU-enheter).

7 Lokalitet

7.1 Tid

Om vi leste fra en minneadresse er det sannsynlig at vi snart vil lese fra den samme adressen igjen.

7.2 Rom

Om vi leste fra en minneadresse er det sannsynlig at vi snart vil lese fra naboadressen.

8 Dataavhengighet

8.1 RAW

Read-After-Write (sanne dataavhengigheter er når for eksempel instruksjon 1 skriver til et register og instruksjon 2 skal lese fra det samme registeret.

8.2 WAW

Write-After-Write (utavhengigheter) er når for eksempel instruksjon 3 skriver til register 1 og instruksjon 1 skriver til register 1.

8.3 WAR

Write-After-Read (antiavhengigheter er når for eksempel instruksjon 3 skriver til register 1 og instruksjon 2 leser fra register 1.

9 RAM

9.1 SRAM

Statisk RAM er raskt og trenger ikke oppdateres. Brukes ofte i hurtigbuffer.

9.2 DRAM

Dynamisk RAM må friskes opp jevnlig. Det tar mindre plass en SRAM (2 vs. 6 transistorer).

9.3 SDRAM

Synkront Dynamisk RAM betyr at data blir overført til/fra RAM synkront med klokka (og systembussen).

10 CMP

Chip-level Multiprocessor er flere prosessorer på samme brukke. Bruker samme hurtigbuffer.

- Homogene kjerner - alle kjerner er like
- Hetrogene kjerner - forskjellige kjerner til forskjellige oppgaver.

Fordeler med CMP er lavere effekt/varmeutvikling, bedre utnyttelse av prosessorkraft, mulighet for "ut av rekkefølge" og lettere å utnytte instruksjonsnivå-parallellitet.

10.1 ILP

Instruction-level parallelism er en måling av hvor mange av et dataprogram sine operasjoner kan bli utført samtidig.

11 Adressering

Måten instruksjonen angir hvor data skal hentes fra kalles en adresseringsmodus.

- Immediate - Operanden er innbakt i instruksjonen. Må være kjent når programmet lages.
- Direkte - Instruksjonen angir adressen til operand i RAM.
- Indirekte - Instruksjonen angir adresse til RAM-celle som igjen inneholder adressen til operand.
- Register - Instruksjonen har nummer på register som inneholder operand.
- Indirekte register - Instruksjon har nummer på register som inneholder adressen til operand i RAM.
- Stakk - Adressen er implisitt gitt av stakkpeker.

12 Branch Prediction

Forgreningspredikering

- Statisk - Forutsier hopp uavhengig av hvor hopp har forekommet før.
- Dynamisk - Forutsier hopp ut i fra hvor det har skjedd hopp før.

13 Prosessor

- RISC = Reduced instruction set computer
- CISC = Complex instruction set computer

- CISC = Hver instruksjon kan utføre flere lavnivåoperasjoner, som for eksempel lese fra minnet, en aritmetisk operasjon og skriving til minnet, alt i én instruksjon.
- RISC = I motsetning til CISC-prosessorer kan utføre relativt få instruksjoner. Til gjengjeld tar hver instruksjon kort tid å utføre.

Heterogenekjerner vil si at kjernene er ulike, dvs de har forskjellig instruksjonssett og/eller ytelse