

# 1.7 Direct memory access

Tuesday, March 24, 2020 3:42 PM

Det er tre teknikker som er mulig for I/O-operasjoner: programmert I/O, avbruddsdrevet I/O, og direkte minneaksess (DMA).

**Programmert I/O:** I dette tilfelle utfører I/O-modulen den forespurte handlingen, for så å sette de nødvendige bits-ene i I/O-status registeret – merk at den ikke gjør noen videre handling for å informere prosessoren. I/O-modulen sender ikke et interrupt-signal. Derfor må prosessoren ta en aktiv rolle og sjekke for om I/O-instruksjonen er blitt utført i etterkant. Av denne grunn må da prosessoren sjekke statusen til I/O-modulen periodisk inntil den leser at operasjonen er utført. Ettersom prosessoren hele tiden må sjekke om operasjonen er utført har dette en negativ påvirkning på ytelsen til systemet.

**Avbruddsdrevet I/O (interrupt-driven):** Handler om at prosessoren sender en I/O-kommando til en modul for så å gå videre til annet nyttig arbeid. I/O-modulen sender et avbruddssignal til prosessoren og ettersørre tjeneste når den er klar for å overføre data med prosessoren. Deretter executer prosessoren dataoverføringen, og fortsetter deretter sin tidligere prosessering. Avbruddsdrevet I/O er mer effektivt enn enkel programmert I/O, men krever fortsatt aktiv undersøkning av prosessoren for om den kan overføre data mellom minne og I/O-modulen. I tillegg må all data som skal overføres traversere en sti gjennom prosessoren. Dermed lider begge (programmert- og avbruddsdrevet-I/O) av to iboende ulemper:

1. I/O-overføringsraten er begrenset av hastigheten til hvor fort prosessoren kan teste og tjenestegjøre en enhet.
2. Prosessoren blir begrenset av å måtte håndtere en I/O-transfer; en antall av instruksjoner må executes for hver I/O-overføring.

**Direkte minne aksess (DMA):** Når store volum av data skal flyttes er DMA, som er en mye mer effektiv teknikk, påkrevd. DMA-funksjonen kan utføres av en separat modul på systembussen, eller kan bli inkorporert inni en I/O-modul. Uansett virker teknikken som følgende:

Når en prosessor ønsker å lese av eller skrive til en blokk av data, sender den en kommando til DMA-modulen ved å sende følgende informasjon:

- Om det er en read-/write forespørsel.
- Adressen til I/O-enheten det gjelder
- Startlokasjonen i minne hvor det skal leses data fra, eller skrives data til
- Antall ord som skal leses eller skrives

*Hvorfor er DMA effektivt?*

Prosessoren fortsetter så med annet arbeid. Den har nå delegert denne I/O-operasjonen til en DMA-modul, som utfører arbeidet. DMA-modulen overfører hele blokken av data, ett ord om gangen, direkte til eller fra minnet uten å måtte gå gjennom minnet. Når overføringen er ferdig sender DMA-modulen et interrupt-signal til prosessoren. Dermed deltar prosessoren kun i starten og slutten av overføringen.

*Hvordan fungerer DMA?*

DMA-modulen må ta kontroll over bussen for å overføre data til og fra minne. På grunn av konkurranse om bussbruk kan det hende prosessoren noen ganger må vente på DMA-modulen når den trenger bussen. Merk at dette ikke er et avbrudd (interrupt); prosessoren verken lagrer en kontekst eller gjør noe annet. Istedenfor må prosessoren pause i en buss-sykel (tiden det tar å sende over et ord langs bussen). Den overordnede effekten er for å få prosessoren til å execute tregere under en DMA-overføringen når prosessor adgang til til bussen er påkrevd.

Uansett er DMA langt mer effektivt enn avbruddsdrevet- og programmert I/O for I/O-overføring av flere ord.