

Til udviklingsmødet

d 28/11-1982

Knud Arne

Multiprocessor System.

En processor består af CPU, MMU, lager med fejlkorrektion samt en businterface til en fællesbus. Til processoren hører endvidere enkelte ydre enheder, der kan anvendes ved test og diagnose. Se Fig. 1.

CPU'en adresserer sit lager gennem MMU'en. MMU'en oversætter CPU'ens logiske adresser til fysiske adresser, som adresserer lageret. Det er MMU'ens opgave at beskytte lageret og at relokere programmer og data.

Adressering af ydre enheder i MC68000 er memory mapped. Af hensyn til beskyttelse og relokering sker adressering af ydre enheder ved hjælp af fysiske adresser.

Data til og fra ydre enheder læses/skrives i en processors lager ved hjælp af fysiske adresser af hensyn til beskyttelse og relokering.

Fællesbussen forbinder processorerne med intelligente kontrollerere, som styrer de ydre enheder.

Interface til fællesbussen afhænger af dennes udseende, men der er to væsentligt forskellige muligheder:

1. Interface er blot en port til processorens lager. Adresser til adressering af lageret genereres af de intelligente kontrollere og sendes over fællesbussen. (Fordelt DMA). Når processoren adresserer intelligente kontrollere, genererer denne adressen, som overføres over fællesbussen. Set fra bussen er en processor identisk med en intelligent controller.

2. Interface til fællesbussen indeholder en DMA controller, som genererer adresser til adressering af processorens lager. Adressen overføres ikke over fællesbussen, som derimod fører et processor nummer og et kanal nummer. (Central DMA). Når processoren adresserer intelligente kontrollere, skal bussen

derimod føre adressen på kontrolleren. Bussen har da to funktioner:

A. Programmeret overførsel, d.v.s. processoren læser og skriver i kontrollere ved anvendelse af fysiske adresser, som overføres over fællesbussen.

B. DMA overførsel, d.v.s. bussen overfører data, processor nummer og kanal nummer, mens adressen til adressering af processorens lager genereres af dennes interface til fællebussen.

Fællesbussen.

Fællesbussen forbinder processorerne med de intelligente kontrollere. Fællesbussen består af en data bus på 16 bit og en adresse bus samt diverse styresignaler. Da flere enheder skal dele bussen, skal der ske en prioritering af adgangen til bussen.

Prioriteringen kan ske ved en central buskontroller, som prioriterer adgangen til bussen. Enhedernes prioritet ved denne fremgangsmåde kan være fast, hvilket er det simpleste, eller roterende.

Prioriteringen kan også ske uden anvendelse af en central buskontroller på følgende måde, se Fig. 2:

En enhed er den nuværende busmaster. Når en anden enhed ønsker bussen, sætter den signal på BR, bus request, som er et åbent collector signal, der er fælles for alle enheder på bussen. Når busmasteren ønsker at opgive bussen, og der er signal på BR, sætter busmasteren signal på BGO, bus grant out. BGO fra en enhed på bussen er forbundet til BGI, bus grant in, på den følgende enhed i kæden. En enhed, som modtager BGI, kan overtage bussen, hvis den ønsker det. Hvis en enhed med BGI ikke ønsker at overtage bussen, sætter den signal på BGO og giver således den næste enhed i kæden mulighed for at overtage bussen. Hvis en enhed med BGI ønsker at overtage bussen, sætter

den ikke signal på BGO, men derimod på BGACK, bus grant acknowledge, hvorved den tilkendegiver over for busmasteren, at en ny enhed er selekteret.

Denne fremgangsmåde giver roterende prioritet.

Uanset hvordan prioriteringen foretages kan den tidsmæssigt placeres på to forskellige måder i forhold til selve buscyclen:

1. Prioriteringen kan ske efter den foregående buscycle er afsluttet, hvilket giver den hurtigste adgang til bussen.
2. Prioriteringen kan ske parallelt med den foregående buscycle, hvilket giver den største bus båndbredde.

Tidsmultiplekset bussystem.

Bussystemet er vist på Fig. 3.

Bussystemet er multiplekset i tid. Tidspfasen angives af TDX(2:0). En buscycle består altså af 8 tidspfaser, hvilket muliggør 8 enheder på bussen. En enhed må drive ADR(7:0) og DATA(15:0), når den genhender sin tidspfase på TDX(2:0). Når en enhed driver ADR(7:0) og DATA(15:0), sætter den samtidigt signal på REQ som tegn på, at tidspfasen ikke er tom. En enhed, som genkender sin adresse på ADR(7:0), og som er i stand til at modtage data, aktiverer ACK og indlæser data. Hvis enheden ikke er i stand til at modtage data, aktiverer den ikke ACK, og det er da senderens ansvar at forsøge igen i næste buscycle. Bemærk at en enhed har en tidspfase som sender og en eller flere adresser som modtager.

I en fase skal der ske følgende:

1. En enhed genkender sin tidspfase og bliver sender.
2. Senderen sætter data og adresse på bussen.
3. Modtageren genkender sin adresse på adresse bussen.

4. Modtageren indlæser data, hvis den kan, og aktiverer ACK.
5. Senderen modtager ACK.

Ovenstående kan uden tvivl udføres på 125 ns, hvilket medfører, at en buscycle bliver på 1 us. Den maksimale overførsels-hastighed for en transport bliver 2 Mb/s, og der kan forløbe 8 sådanne transporter samtidigt.

Ulemperne ved ovenstående er at bussen er "write only", idet senderen altid er den aktive i en transport. Den største ulempe er imidlertid, at en enhed kun kan være i færd med en transport ad gangen. Hvis en processor sender en blok til en printer, kan den ikke samtidigt være i færd med at sende en blok til en disk.

Et bussystem.

I det følgende beskrives et bussystem, hvor processorernes interface til bussystemet blot er en port til processorens lager. Bussystemet er vist på Fig. 1.

Prioriteringen af bussystemet sker med fordelt prioritering som tidligere beskrevet. Prioriteringen af den følgende bus-cycle sker parallelt med den nuværende.

Bussystemet består af en databus på 16 bit og en adressebus. Adressebussen fører dels lageradresser dels adresser på enheder tilknyttet bussen.

Adressering.

Processorernes fysiske adresserum er delt, så en del af dette er lokalt lager, og resten er fælles lager og adresser i I/O kontrollerne. Adresseudgangene fra MMU'en dekodes, og når en reference ikke adresserer det lokale lager, aktiveres bus interfacen.

Læsning via fællesbussen.

Ved læsning i fælleslager eller I/O kontrollerer sender businterfacen busrequest og MC68000 standses ved hjælp af WAIT states. Når businterfacen får adgang til fællesbussen, sætter den adressen på adressebussen og aktiverer styresignalet læs. De intelligente kontrollerer dekode adressen, og i den adresserede enhed genereres et HOLD request til CPU'en. Når CPU'en i kontrolleren giver slip på den lokale bus, gates adressen ind på denne og læsningen i lageret udføres. Lageret i den intelligente kontroller udlæser data, som via den lokale bus i kontrolleren gates ud på fællesbussen og videre til den lokale bus i processoren, hvorefter MC68000 fortsætter sin cycle.

Læsningen består af følgende hændelser:

1. Adgang til fællesbussen.
2. Adgang til lokalbus i I/O kontroller.
3. Læsning af data i I/O kontrollers lager.

Skrivning via fællesbussen.

Skrivning i en I/O kontrollers lager kan ske hurtigere end læsning. Når processoren har fået adgang til fællesbussen, overføres adresse og data til I/O kontrollerens interface. Herefter er buscyclen afsluttet, og I/O kontrolleren kan selv skrive data i lageret enten programmeret eller ved hjælp af en sekvensmaskine, DMA.

I det foregående er beskrevet programmeret læsning/skrivning i I/O kontrollers lager, d.v.s. access til en I/O kontrollers lager udført af en processor. I/O kontrollerens interface vil også indeholde registre, som kan læses og skrives fra fællesbussen. Access til disse registre medfører ikke HOLD i I/O kontrolleren, men kan f.eks. generere et interrupt i denne.

Access til processors lager fra I/O kontroller.

I det følgende beskrives access til processors lager fra I/O kontroller. Normalt forløber datatransporter mellem processor og I/O kontroller på denne måde.

Set fra I/O kontrolleren kan datatransporten ske programmeret eller ved hjælp af DMA.

Programmeret overførsel.

Den programmerede overførsel kan f.eks. ske på denne måde: En del af CPU'ens adresserum tilknyttes fællesbussen, f.eks. 256 bytes. Businterfacen indeholder et adresseregister, som sammen med de 8 mindst betydende bit i CPU'ens adresse udgør adressen, som kan sættes ud på fællesbussen. CPU'en er i stand til at læse eller skrive 256 bytes i en processors lager med et givet indhold i adresseregisteret.

Selve datatransporten sker som tidligere. I/O kontrollerens adresse dekodes. Når fællesbussen adresseres, aktiveres interface, mens der genereres WAIT til CPU'en i kontrolleren. Når interface får adgang til fællesbussen, sætter den adressen ud på denne. Businterface i processorerne dekoder denne, og i den adresserede processor genereres "HOLD". Når processoren giver slip på den lokale bus, kan lageret accesses. Hele forløbet består af:

1. Adgang til fællesbussen.
2. Adgang til lokal bus i processor.
3. Læsning/skrivning af data.

DMA overførsel.

Interface i I/O kontrolleren er forsynet med en DMA enhed bestående af et 16 bit dataregister og adresseregister. Adresseregisteret er en tæller, der kan loades programmeret af I/O kontrolleren. Interfacen indeholder styrelogik, som kan sende bus request til fællesbussen og DMA request til en lokal DMA kontroller i I/O kontrolleren. Læsning i en processors lager sker på følgende måde: I/O kontrolleren initialiserer den lokale DMA kontroller og businterface herunder adresseregisteret. Businterface sender busrequest til fællesbussen og data læses i processorens lager som tidligere. Når data er indlæst

i interface i I/O kontrolleren, sendes DMA request til den lokale DMA kontroller, hvorved data overføres fra interface til I/O kontrollerens lager ved 2 DMA transporter. Interfacen sender igen bus request til fællesbussen for at læse et nyt ord.

Transporten omfatter:

1. Adgang til fællesbussen.
2. Adgang til lokal bus i processor.
3. Læsning/skrivning i lokalt lager i processor.
4. DMA transporter af 2 bytes mellem businterface og lokalt lager i I/O kontroller.

Fællesbussen er optaget i perioderne 2 og 3.

Tider.

Ventetiden for at få adgang til en bus sættes til den maksimale cycle tid for bussen. Dette er rigtigt, hvis der kun er to enheder på bussen eller bussen kun er svagt belastet. Cycle tiden for MC68000 sættes til 1 us. Cycletiden for 8085 er 1 us ved 5 MHz eller 2 us ved 2 MHz svarende til henholdsvis 5 og 4 states pr. cycle. I det følgende sættes cycle tiden for 8085 til 1 us, hvilket medfører, at adgangen til en lokal bus bliver 1 us, uanset om lokalbussen findes i en processor eller i en I/O kontroller. Lager "access" tider sættes til 0.5 us.

Fællesbussen er optaget i den tid det tager at accesse en lokal bus samt at lave en lager access, d.v.s. 1.5 us.

Processor læser/skriver i I/O kontrollers lager:

- | | |
|---------------------------|--------|
| 1. Adgang til fælles bus: | 1.5 us |
| 2. Adgang til lokal bus: | 1 us |
| 3. Lager access: | 0.5 us |

Total: 3 us

Fællesbussen anvendes i 1.5 us

I/O kontroller læser/skriver i processors lager:

Samme resultat som ovenstående: 3 us.

Et simpelt program i 8085, som læser i en processors lager og skriver i sit eget, tager 11/44 cycles/states.

2 MHz 8085 med 0 WAIT state: 22 us + 2.5 us = 24.5 us

5 MHz 8085 med 1 WAIT state: 11 us + 2.5 us = 13.5 us

Fællesbussen kan bære 9 transporter af denne art.

I/O kontroller læser/skriver i processors lager ved anvendelse af lokal DMA.

- | | |
|---------------------------|--------|
| 1. Adgang til fælles bus: | 1.5 us |
| 2. Adgang til lokal bus: | 1 us |
| 3. Lager access: | 0.5 us |

Totalt: 3 us/ord

Fællesbussen anvendes i: 1.5 us

DMA transporten med 2 mhz DMA kontroller:

- | | |
|--------------------------------|--------|
| 1. Adgang til lokal bus: | 1 us |
| 2. DMA cycle $5 \cdot 0.5$ us: | 2.5 us |

Totalt: 3.5 us/byte

Totalt for hele transporten: $3 + 2 \cdot 3.5$ us = 10 us/ord
= 5 us/byte

Maksimal transport hastighed: 5 us/byte

Maksimal bus hastighed: 0.75 us/byte

Bussen kan bære 6 sådanne transporter.

DMA delen af transporten kan forbedres, idet der kan anvendes en 4 MHz DMA kontroller, som anvender burst på 2 byte:

1. Adgang til lokal bus: 1 us
2. 2 DMA cycles $2 \cdot 5 \cdot 0.25$ us: 2.5 us

Total: 3.5 us/ord

Totalt for hele transporten: $3 + 3.5$ us = 6.5 us/ord
= 3.25 us/byte

Maksimal transport hastighed: 3.25 us/byte
Maksimal bus hastighed: 0.75 us/byte

Bussen kan bære 4 sådanne transportere.

Hvis interface forsynes med dobbelt buffer, så DMA cycles og cyclen på fællesbussen kan forløbe parallelt, bliver DMA tiden den begrænsende faktor i transporten. Herved opnår man

Maksimal transport hastighed: 1.75 us/byte
Maksimal bus hastighed: 0.75 us/byte

Bussen kan bære 2 sådanne transportere.

Når en DMA transport som den sidste af ovenstående forløber i en 8085, er denne stoppet i 2.5 us ud af 3.5 us, d.v.s. 8085's hastighed er reduceret til ca. 30 pct.

Vurdering af transport hastighed.

Da ovenstående tider ikke umiddelbart er imponerende, er en nøjere analyse nødvendig.

I det følgende regnes med:

Maksimal transport hastighed: 2 us/byte = 500 kb/sek.
Maksimal bus hastighed: 1 us/byte = 1 Mb/sek.

En hard disk har en overførselshastighed på ca. 1 us/byte. Transport hastigheden er altså for lille til at udnytte disken fuldt ud. Ved normal anvendelse vil en stor del af diskens tid anvendes til hoved flytninger. Hvis access tiden for disken sættes til 30 ms, kan der overføres 15 kbytes pr. disk access svarende til ca. 60 sektorer uden at transporthastigheden begrænser systemet. Hvis der overføres mere end 15 kbytes pr. disk access, eller hvis access tiden er mindre end 30 ms, vil transporthastigheden begrænse systemet.

Lad os antage, at systmet anvender 2 disk drev, og at der overføres 10 kbytes for hver disk access, og at disk access tiden er 30 ms. Der overføres da 20 kbytes/30 ms svarende til 666 kbytes/sek. Bus båndbredden er 1 Mb/sek, hvoraf der rester 333 kb/sek.

En asynkron linie på 9600 baud svarer til ca. 1 kb/sek. Hvis fællesbussen til disse transportere kun anvendes til byte transportere, bliver belastningen fra en linie 2 kb/sek. Kommandoer til styring af en asynkron linie overføres over fællesbussen. Denne belastning kan sættes til 4 kb/sek., hvilket er dobbelt så meget som belastningen hidrørende fra data. Den resulterende belastning fra en linie er da 6 kb/sek.

Maskinen kan altså drive de to omtalte disktransportere og 50 asynkron linier på 9600 baud.

Transport hastigheder i SPC/1.

Datatransporten i SPC/1 sker fra disken til det faste lager ved hjælp af DMA og dernæst fra det faste lager til brugerens lager ved hjælp af lager til lager DMA. Bufferstørrelsen i det faste lager vil normalt begrænse transport hastigheden, så denne bliver 4 sektorer pr. disk rotation.

Ved en access til en hard disk bliver transport hastigheden derfor 1 kb/16.6 ms, svarende til 60 kb/sek.

Ved en acces til en floppy disk bliver transport hastigheden 6 kb/sek.

I de nuværende back up programmer sker data transporten med en hastighed på 16 us/byte svarende til 62 kb/sek.

Konklusion.

Det skitserede bussystem tillader ikke fuld udnyttelse af en hard disk med f.eks SMD interface. På den anden side er I/O kapaciteten med sikkerhed 10 gange så stor som i SPC/1.

Multiprocessor system

Fig. 1

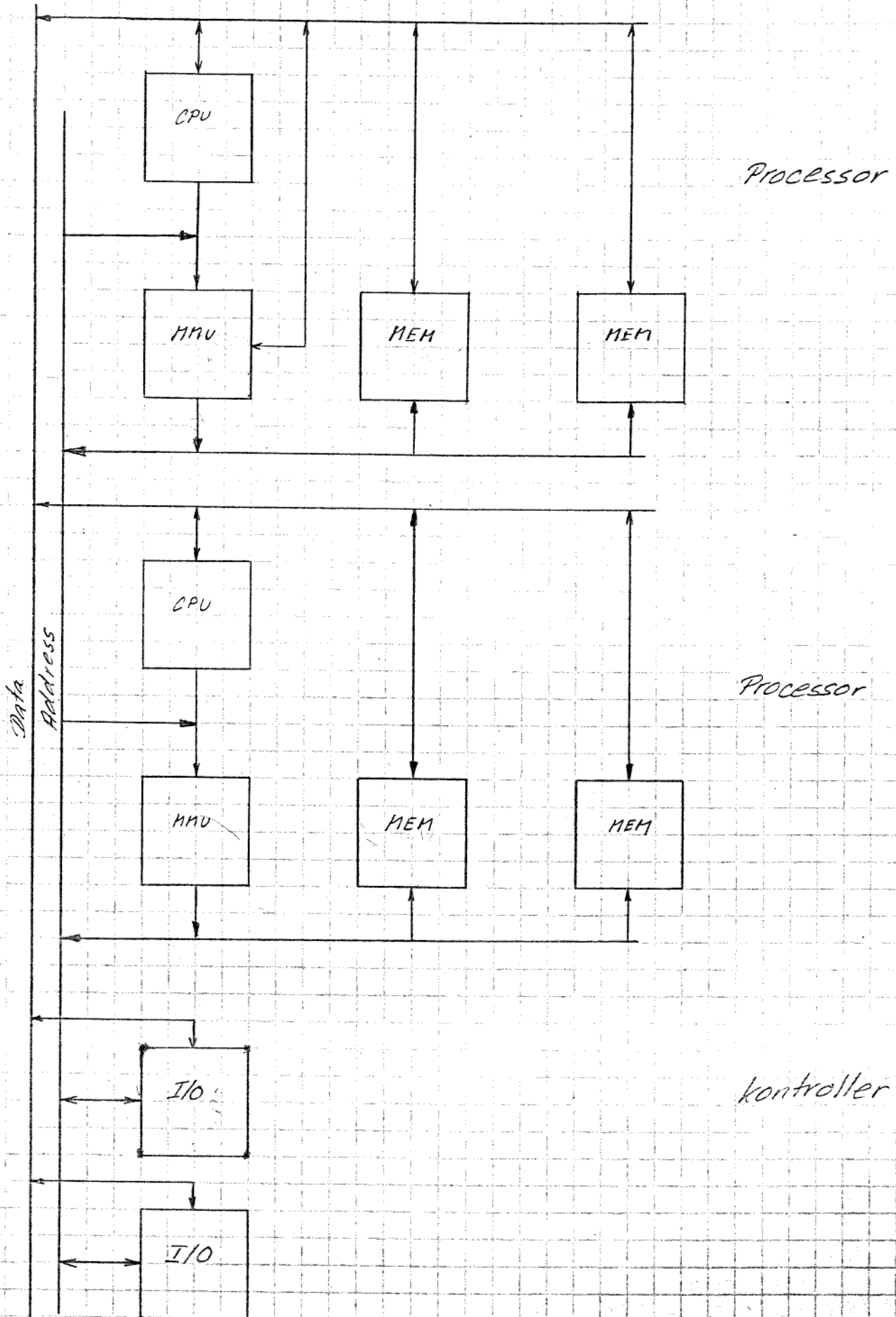


Fig. 2

Cyklisk prioritering

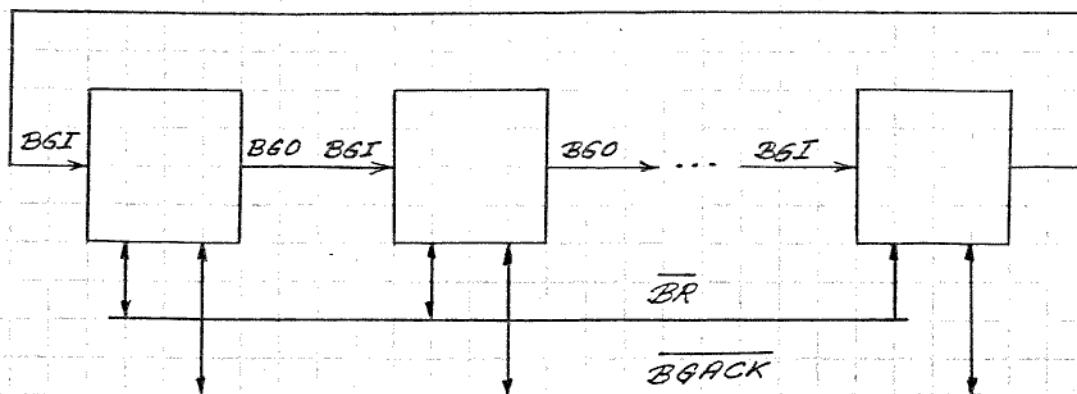


Fig. 3

Tidsmultiplexset bussystem

