F7: I/O hantering

- **Periferikretsar**
  - **ADC, DAC, UART, etc.**
- **Databussar**
  - **Seriella bussar**
    - **I²C**
    - **USB**
    - **CAN**
  - **Systembussar**
    - **PCI**
    - **VME**
- **Asynkron och synkron busscykel 68000**
- **Bussfördelning**

Periferikretsar

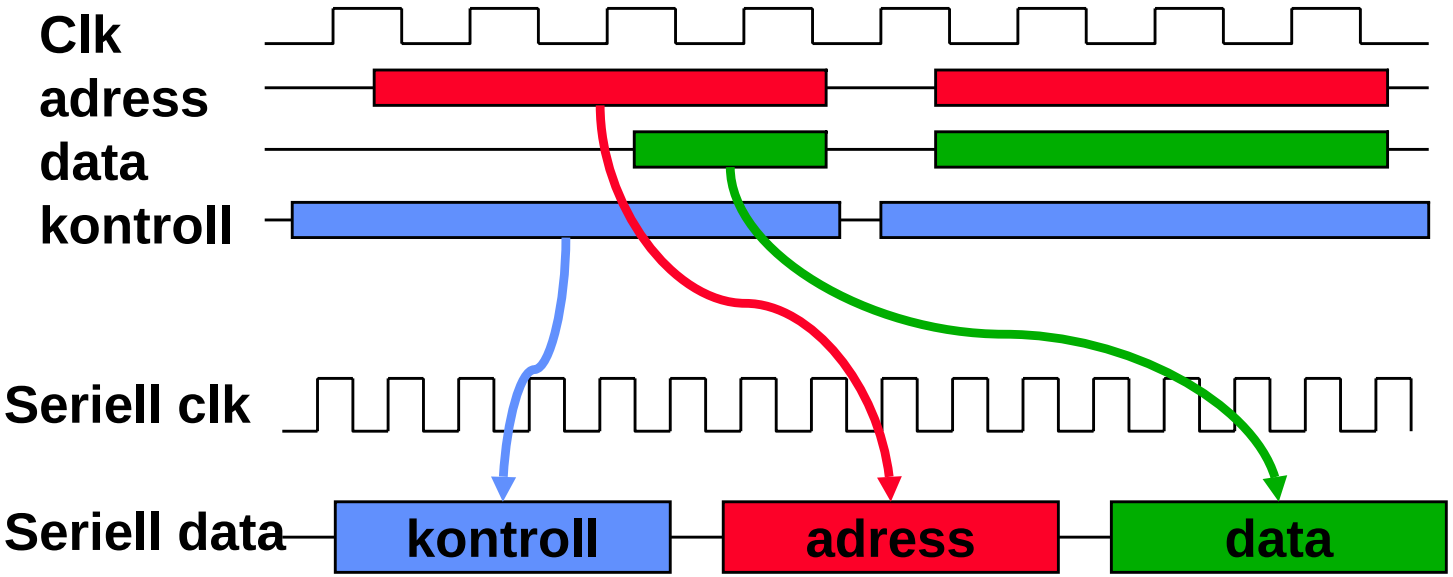
- ADC/DAC
- Räknare
- Realtidsklocka
- Timer
- Interrupthanterare
- UART (Universal Asynchronous Reciever and Transmitter)
  - Seriell dataöverföring
- Parallellport
- PWM (Pulse Width Modulation)
- Kommunikationskretsar (I2C, CAN, Ethernet, etc.)

Systembussar

- **En databuss karakteriseras av**
  - **Överföringshastighet (bytes/s)**
  - **Klockhastighet (Hz)**
  - **Bitbredd (bits)**
  - **Överföringstyp (parallell/seriell)**
  - **Adresseringsområde**
- **Databussar är ofta standardiserade**
  - **Systemkomponenter kan återanvändas**
  - **Ett väldefinierat och känt system är lättare att använda**

Seriellabussar

- **En seriell buss är en seriell kommunikations-kanal med kontroll-, adress- och databuss på samma seriella buss (tidsmultiplexad).**
  - **I^2C, USB, CAN**

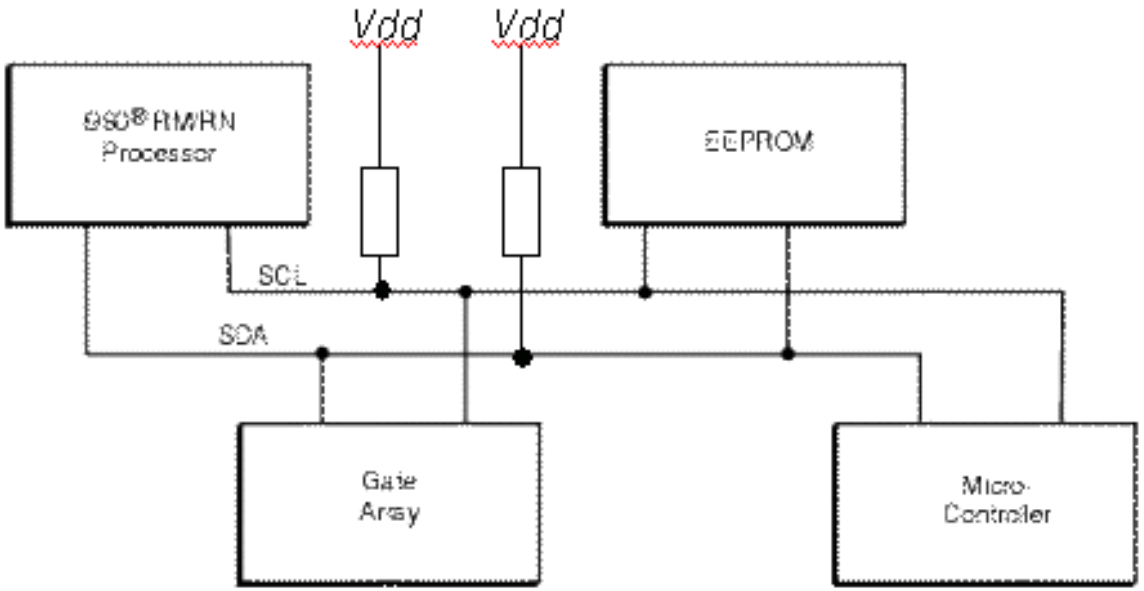


I²C

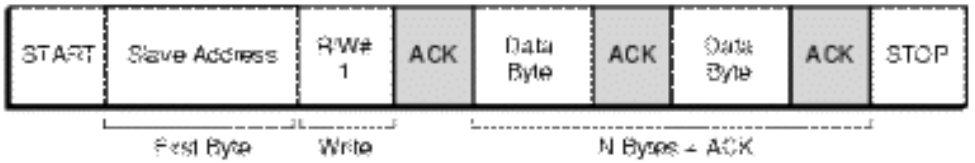
- **I2C, Inter Integrated Circuit bus.**
  - **I2C bussen utveckaldes av Philips (1980) för att enkelt och billigt kunna koppla samman komponenter i TV apparater.**
  - **Signaler, I2C bussen består av 2 fysiska ledningar (signaler) + en jordledning.**
    - **SDA and SCL.**
    - **SDA är Serial DAta line och SCL är Serial CLock line.**
  - **Varje komponent som sitter på bussen har en unik adress**
  - **Buss MASTER är komponenten som skickar kommandon på I2C bussen som initierar bussaccesser (ofta en processor)**
  - **I2C bussen är en multimaster buss.**
  - **Max. hastighet, ca 400 kbit/s => 40kb/s**

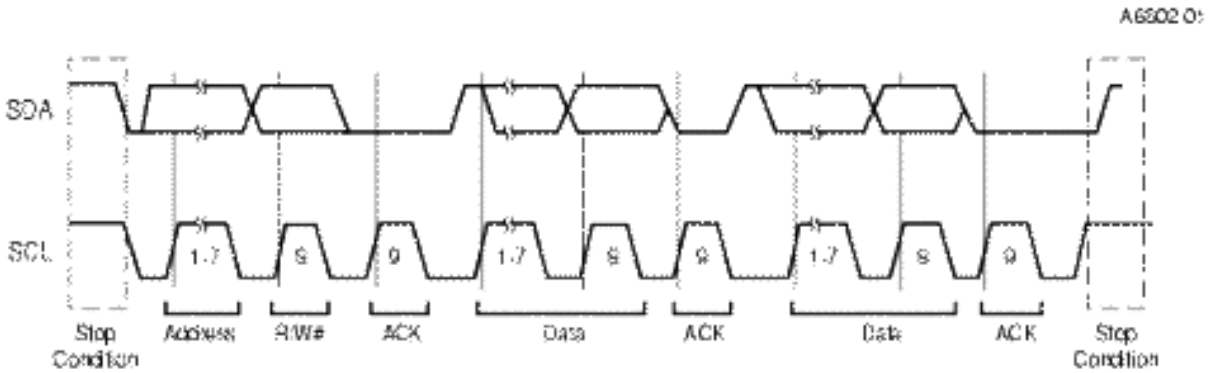
I²C

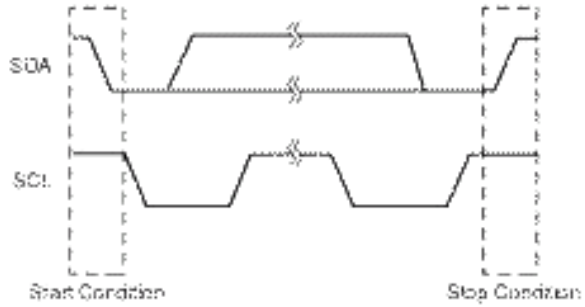
- **Endast 2 signaler används för att kommunicera mellan alla komponenter**
  - **Mycket enkelt och billigt att koppla ihop komponenter**



I²C

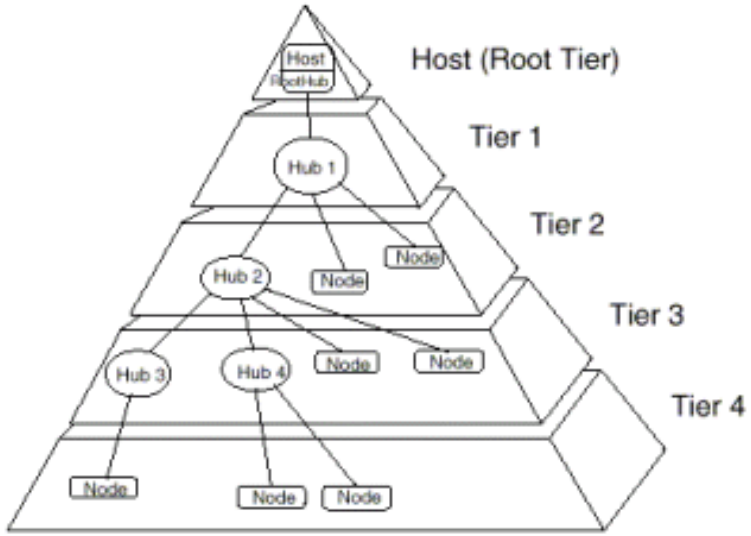




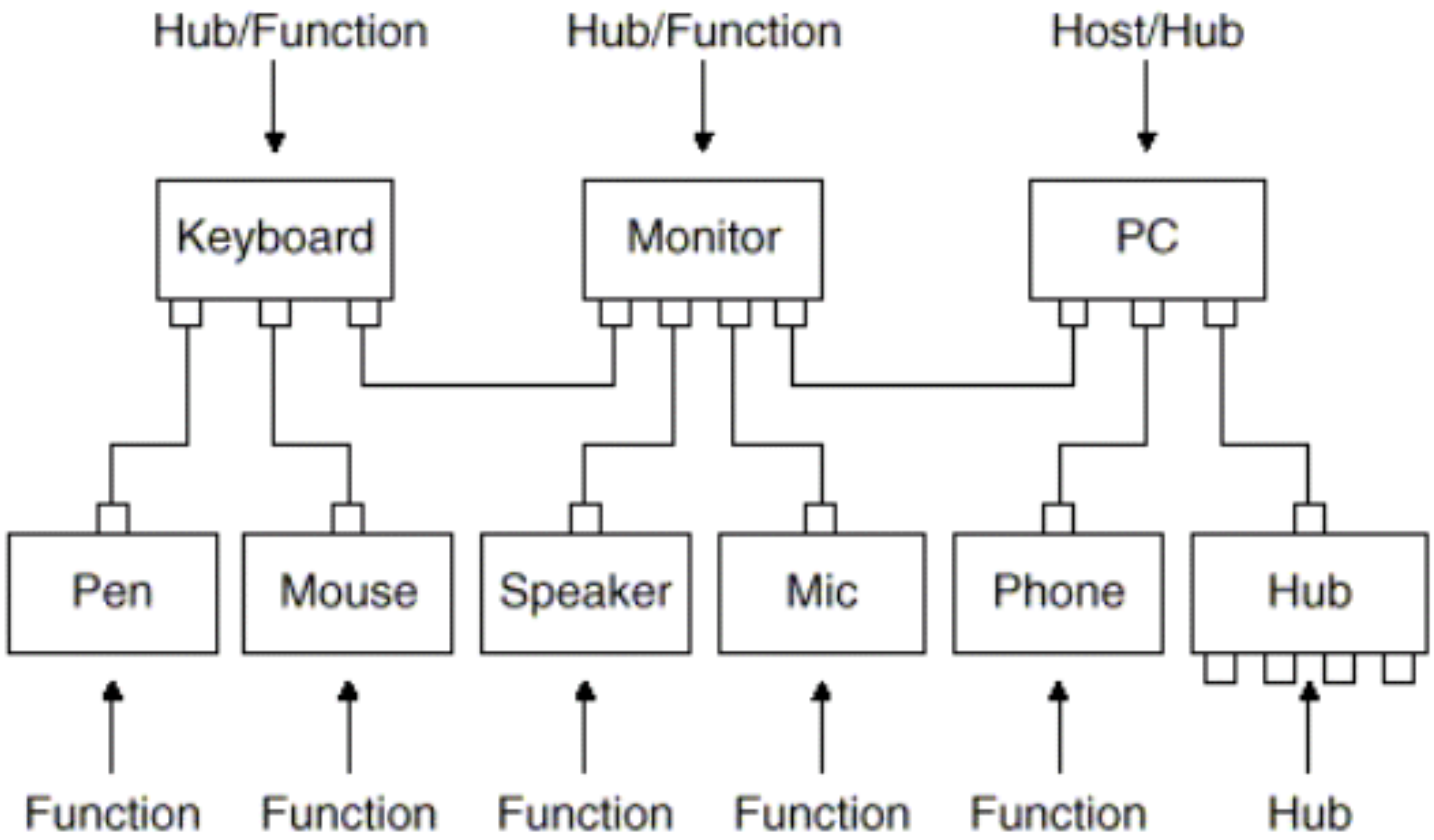


USB

- **Universal Serial Bus (USB)**
- **Utvecklad av dator- och telekommunikations-industrin för att kommunicera mellan datorer och datorkomponenter, t ex, tangent bord, mus, etc**
  - **Maximal bandbredd för USB 1.0 är 12 Mbps**
  - **-”- USB 2.0 är 480 Mbps**

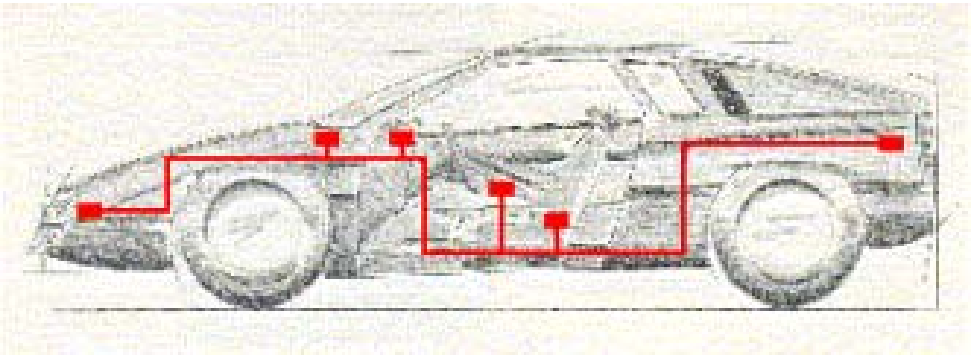


USB

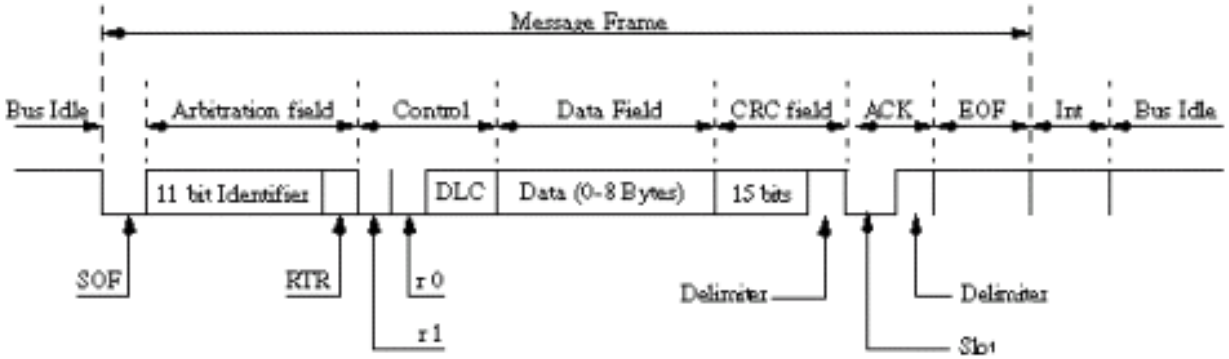


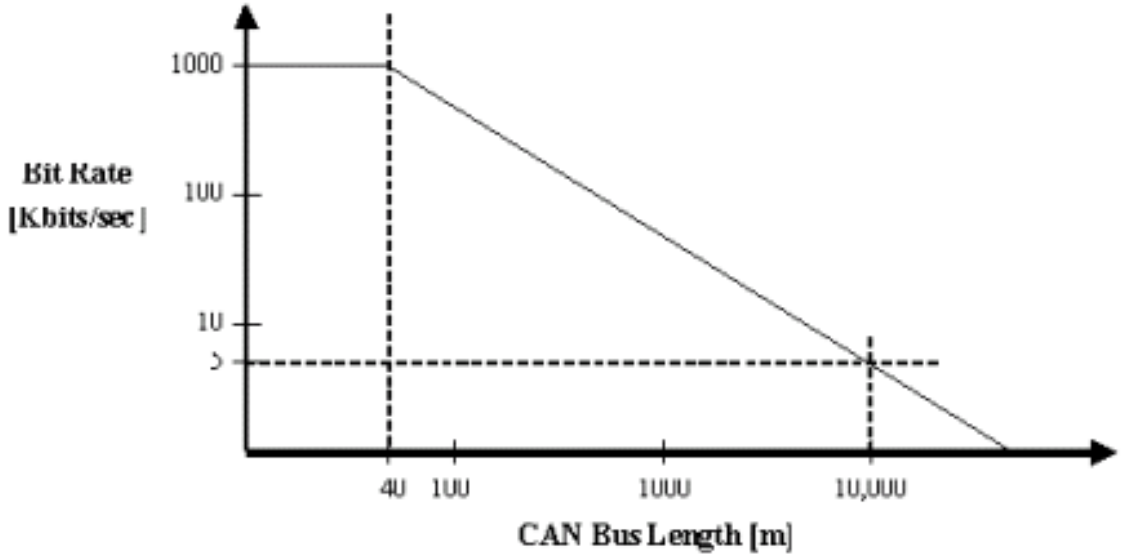
CAN

- **CAN, Controller Area Network (Bosch)**
  - **Är konstruerad för att klara störningar och realtidstillämpningar**
  - **Datahastigheter upptill 1 Mbits/s**
  - **Standarden stöder felupptäckt**
  - **Utvecklades av Bosch för att användas i bilar**
    - **Används i många andra industriella tillämpningar.**



CAN





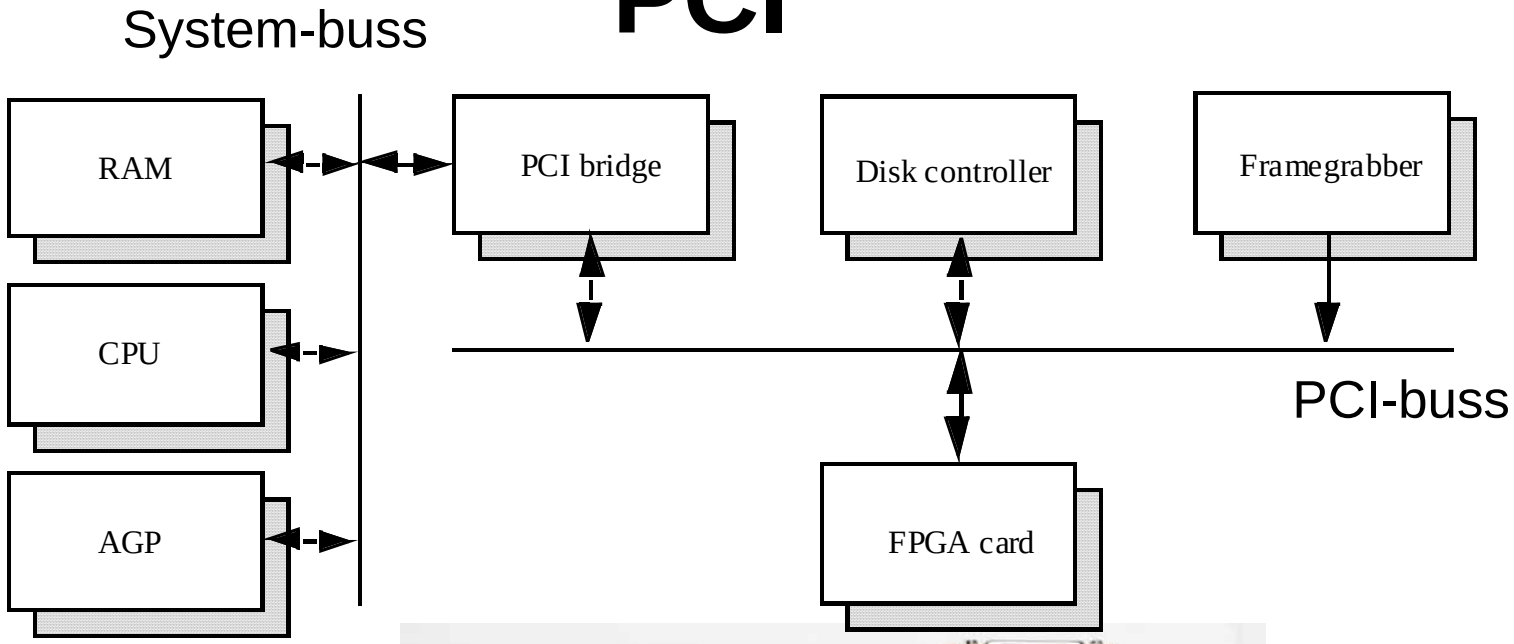
Parallella datorbussar

- **Används som anslutningsmedium för tilläggskort i datorsystem.**
  - **PCI**
  - **VME**
  - **PCMCIA**
  - **ISA**
  - **etc...**

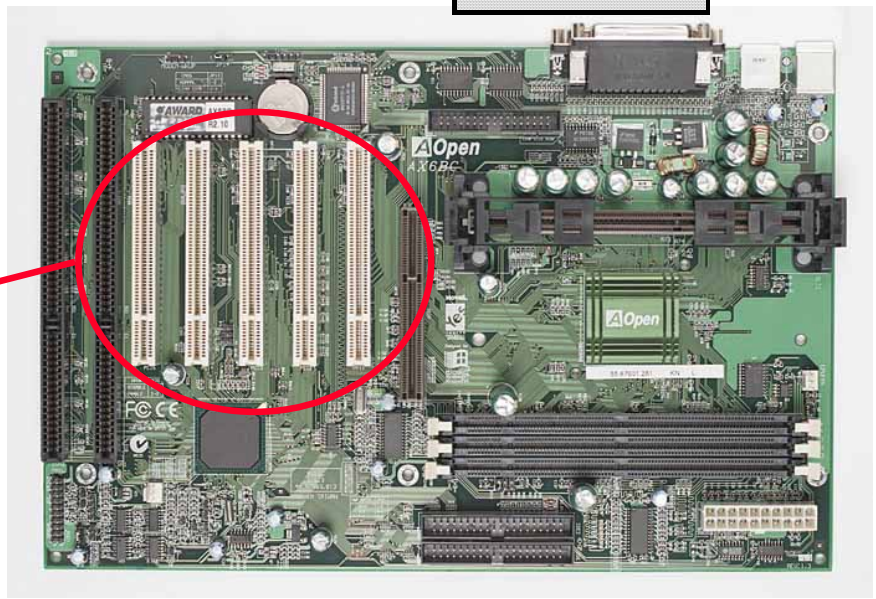
PCI

- PCI, Peripheral Component Interconnect bus
  - Initierad 1991 av Intel
- Bitbredden är 32 eller 64 bitar
- Frekvensen är 33 eller 66 MHz
- Hastighet
  - 32 bit, 33 MHz ger 120 Mbytes/s
  - 32 bit, 66 MHz ger 240 Mbytes/s
  - 64 bit, 66 MHz ger 480 Mbytes/s

PCI

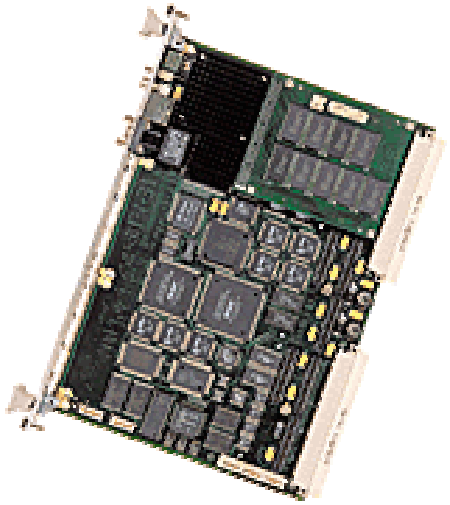


PCI anslutningar

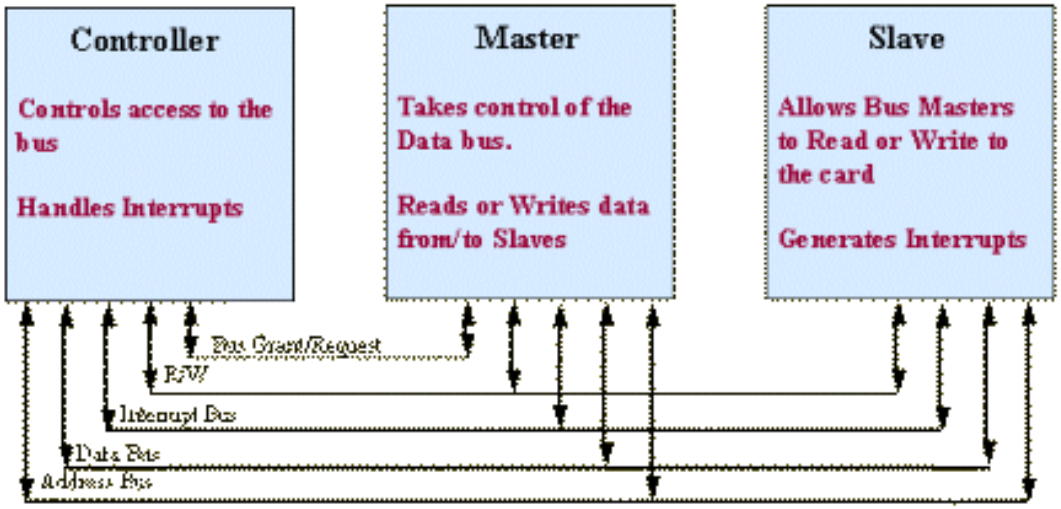


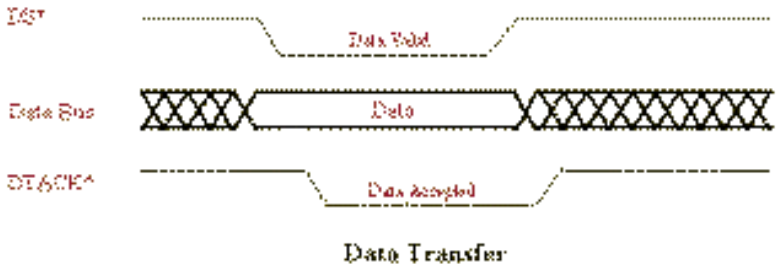
VME

- **Initierades av Motorola (1979)**
  - **Baserad på Motorolas 68000 processorbuss**
  - **Standardiserad av IEEE 1984 (IEEE 1014)**
- **Bitbredd = 8/16/32 bitar**
- **Asynkron**
- **Hastighet = 20 Mbytes/s**



VME



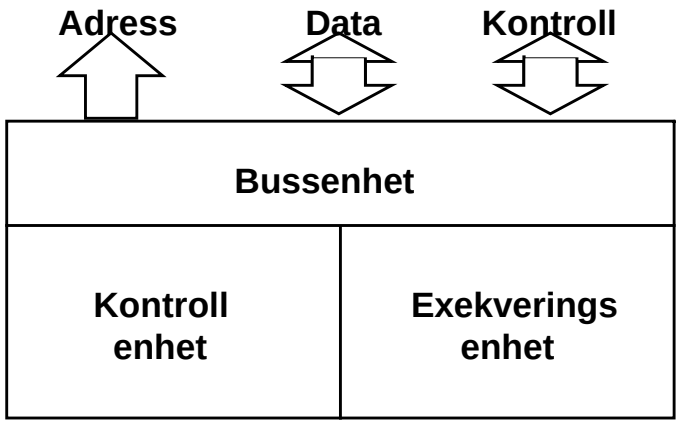


Data Transfer

Address Transfer

68000

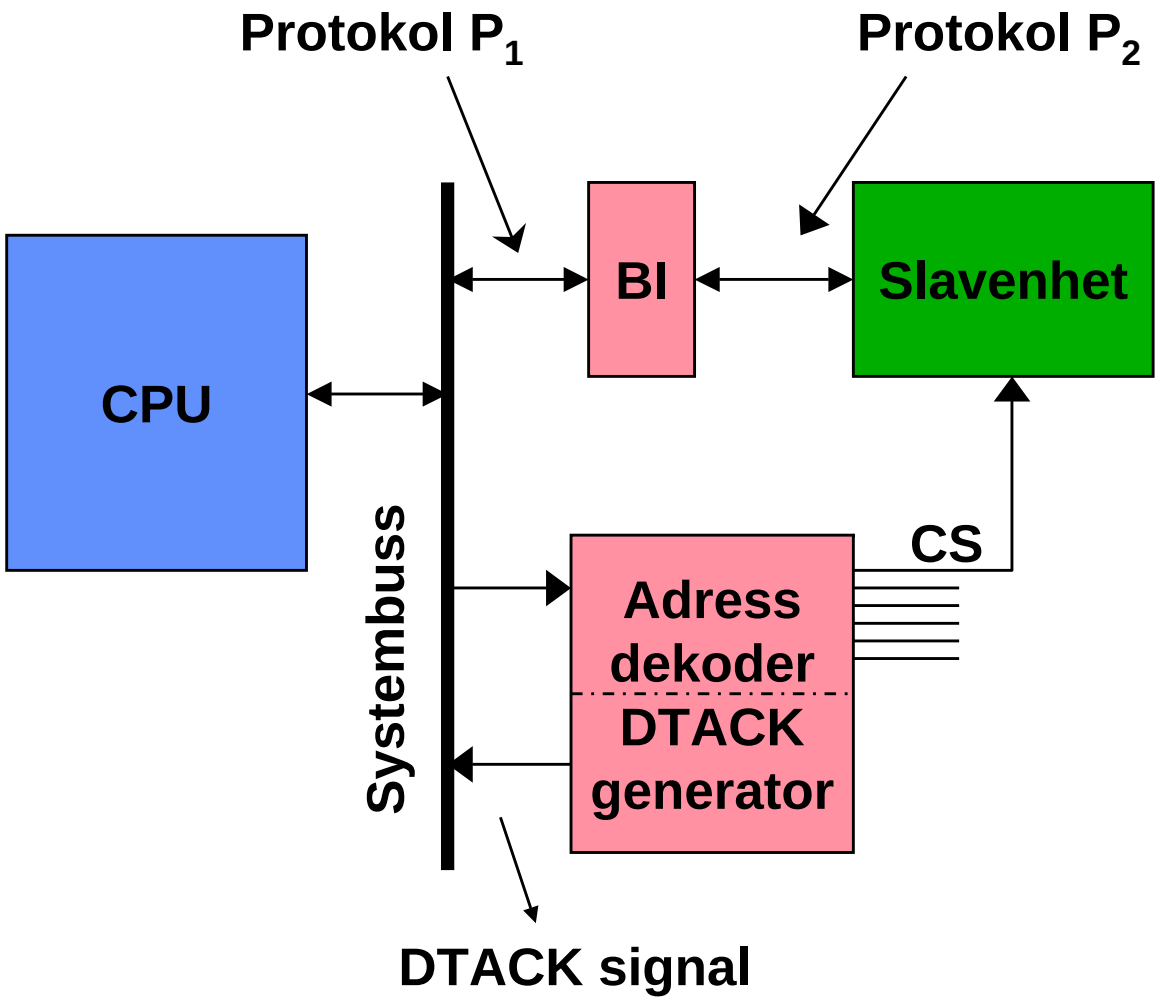
- **Exekveringsenhet**
  - **Utför logiska/aritmetiska operationer**
- **Kontrollenhet**
  - **Översätter instruktionrna till handling**
- **Bussenhet**
  - **Kontrollerar läs/skriv data operationer**
  - **Hämtar instruktioner från minnet**



Bussenhet

- **Läsa/Skriva data till/från yttre enheter.**
  - **Interna bussystemet används för att transportera data mellan de olika enheterna.**
- **Buffertenheten**
  - **För att kontrollenheten ska kontinuerligt få instruktioner att exekvera utan att behöva vänta på att instruktionen ska laddas från minne.**
    - **Förhämtningskö: Hämtar instruktioner som är på tur att exekveras vilket ökar utnyttjande graden.**
  - **24 - bitars bredd på adressbussen**
  - **16 - bitars bredd på databussen**
  - **2 kontrollbussar**
    - **Asynkron kontroll (med handskakning)**
    - **Synkron kontroll (MC6800 buss)**

Bussinterface



Asynkron busskontroll, 1(2)

- **Använder full handskakning**
  - **När processorn har adresserat en slavenhet väntar den tills den fått ett klartecken från enheten**
- **En del slavenheter har inte denna typ av full handskakning**
  - **Detta genereras av en speciell enhet, (sekvensnät)**
- **Handskakning sköts med följande signaler**
  - **DTACK*, Data transfer acknowledge**
    - **Använd för att indikera att dataöverföringen är klar.**
  - **BERR*, Bus Error**
    - **Indikera att något fel uppstod vid dataöverföringen.**
  - **AS*, Address Strobe**
    - **Startar en dataöverföring, indikera att adressen på adressbussen har ett giltigt värde.**

Asynkron busskontroll, 2(2)

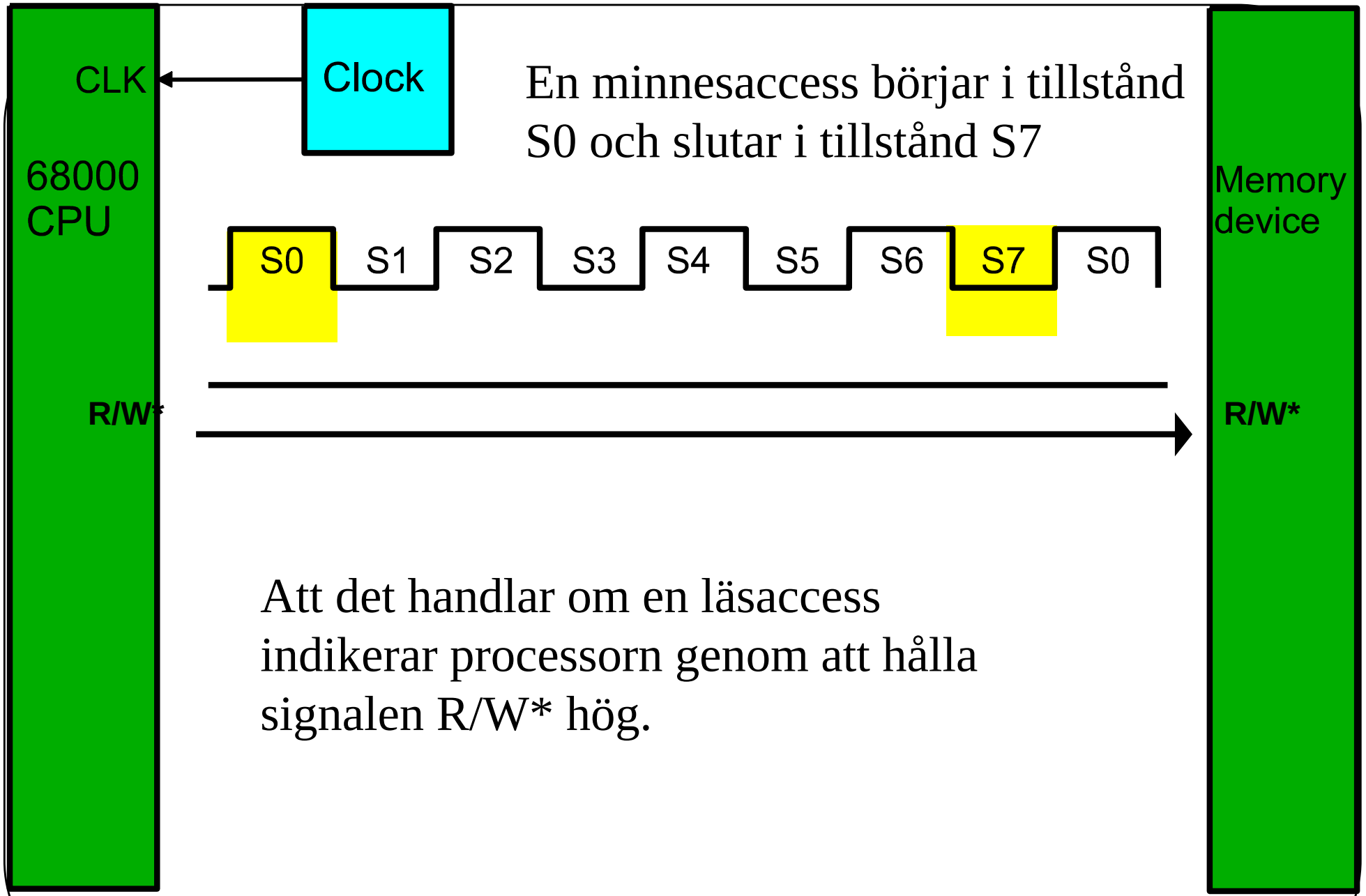
- **LDS*, Lower Data Strobe**
  - Indikera om data ska läsas/skrivas på bit d0-d7
  - Inträffar om:
    - a0 = 0 och läs/skriv ord (16 bit)
    - a0 = 1 och läs/skriv byte (8 bit)

- **UDS*, Upper Data Strobe**
  - Indikera om data ska läsa/skrivas på bit d8-d15
  - Inträffar om:
    - a0 = 0 och läs/skriv ord (16 bit)
    - a0 = 0 och läs/skriv byte (8 bit)

- **R/W*, Read/Write**
  - Indikera om processorn ska läsa eller skriva data
  - '1' = Read, '0' = Write

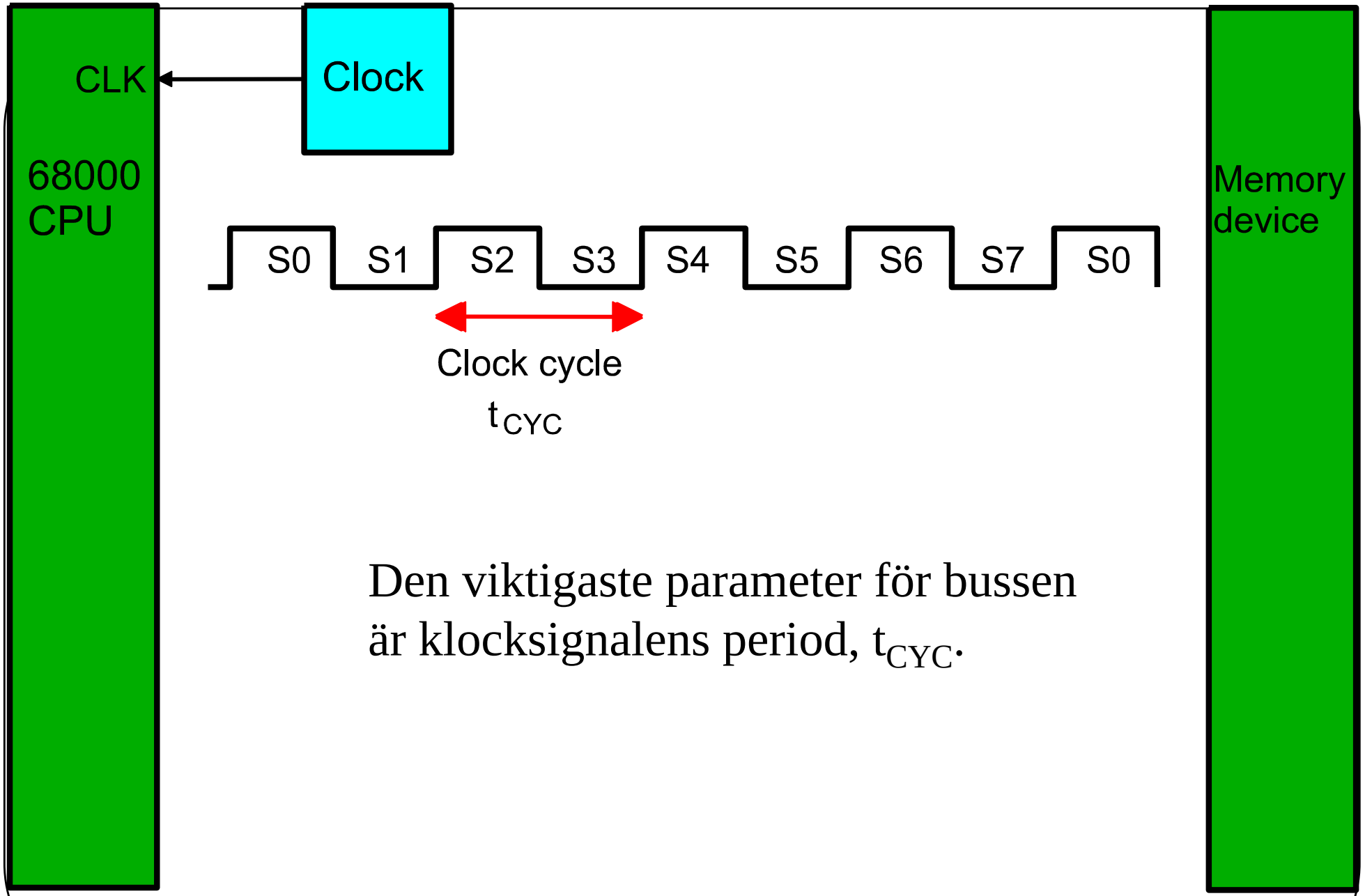
En läscykel på 68000-bussen

- **Klocksignalen**
  - **Klocksignalen används för att dela in operationer i steg (synkronisering)**
  - **Klocksignalen används också för att kontrollera bussaccesser**
  - **68000's minnesaccess tar minst 8 klocktillstånd att genomföras (S0 till S7)**
    - **En klockcykel är 2 klocktillstånd för en bussaccess utan väntcykler**

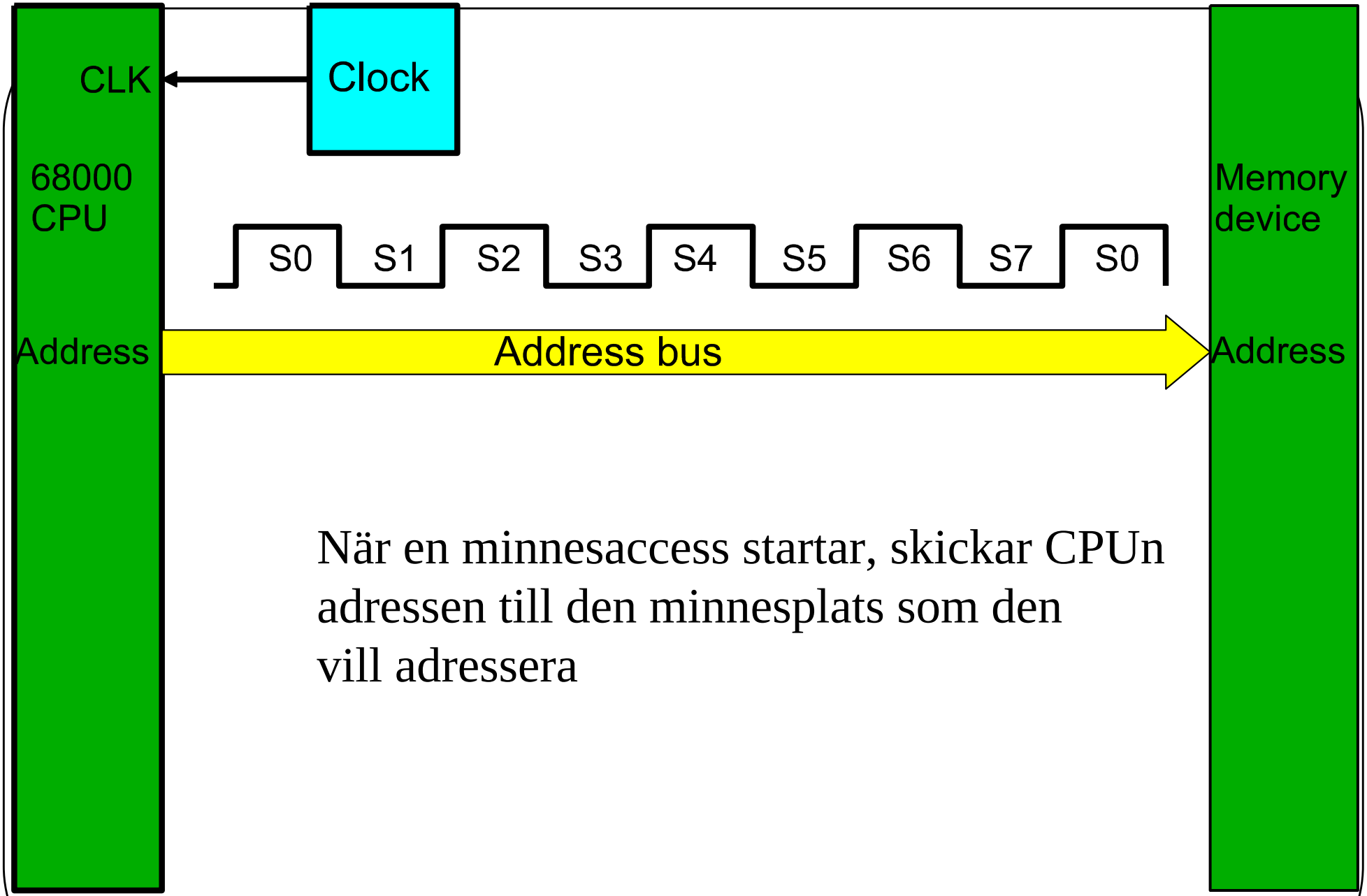


En minnesaccess börjar i tillstånd S0 och slutar i tillstånd S7

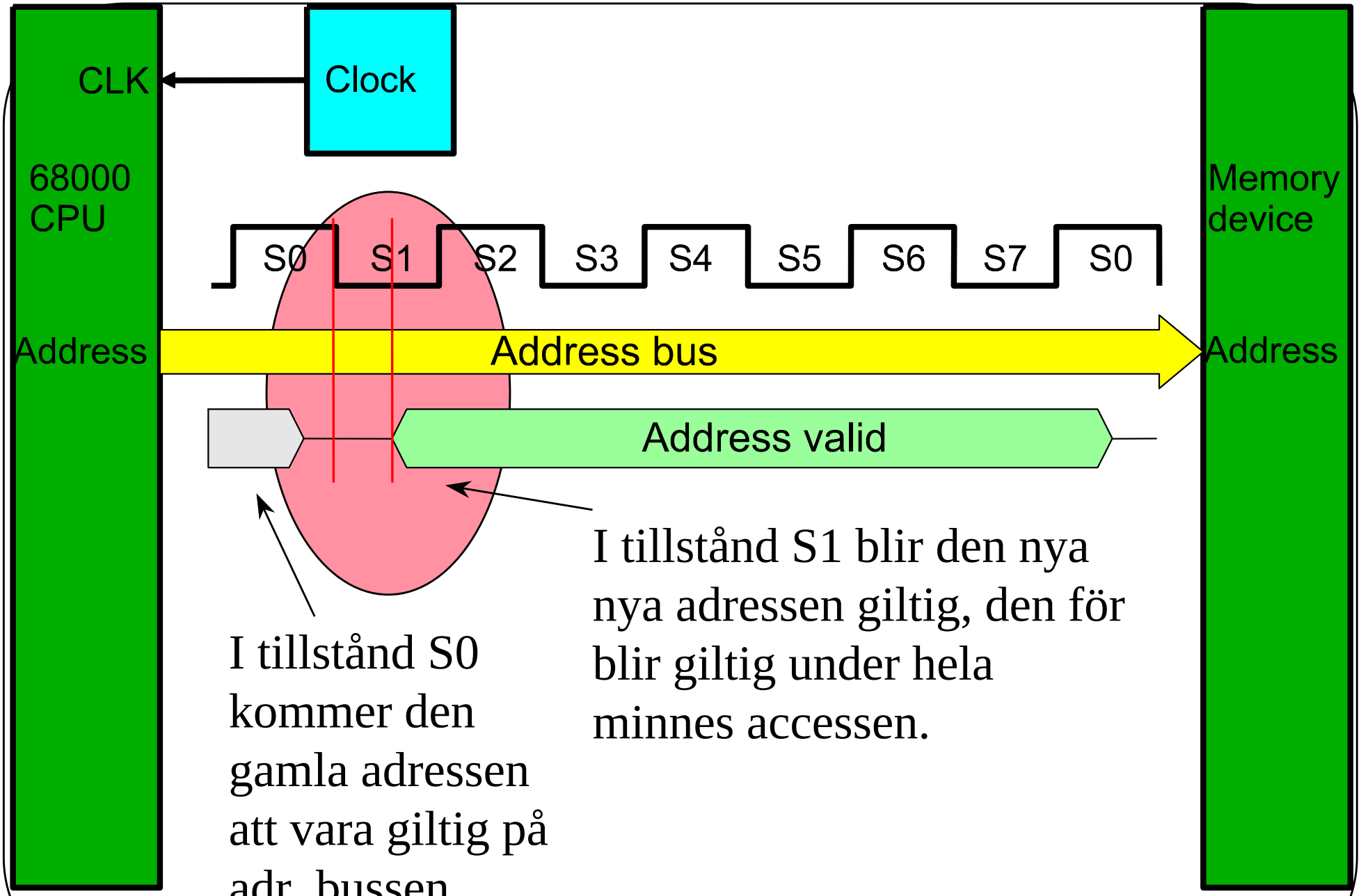
Att det handlar om en läsaccess indikerar processorn genom att hålla signalen R/W* hög.



Den viktigaste parameter för bussen är klocksignalens period, t_{CYC}.

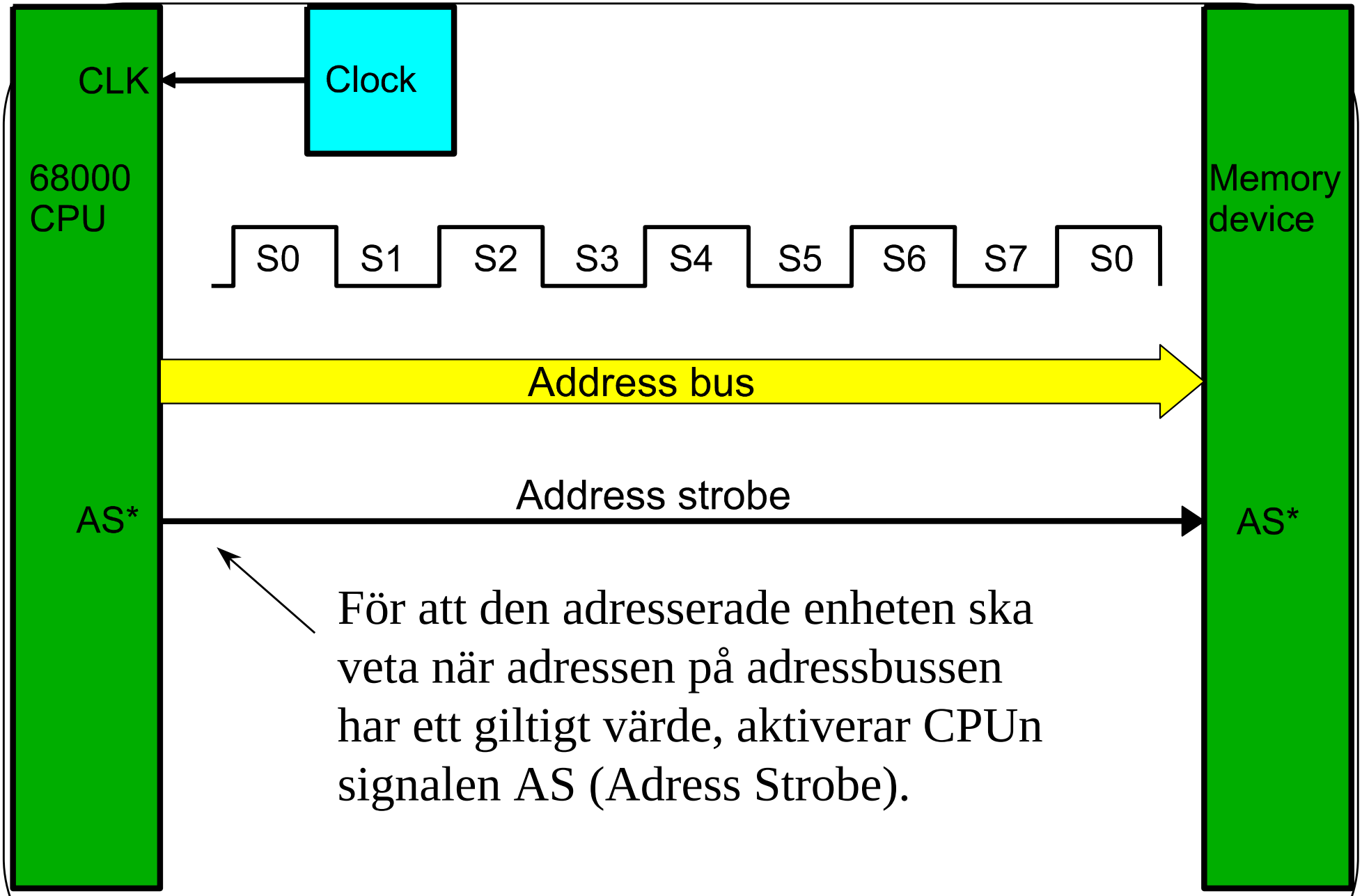


När en minnesaccess startar, skickar CPUn adressen till den minnesplats som den vill adressera

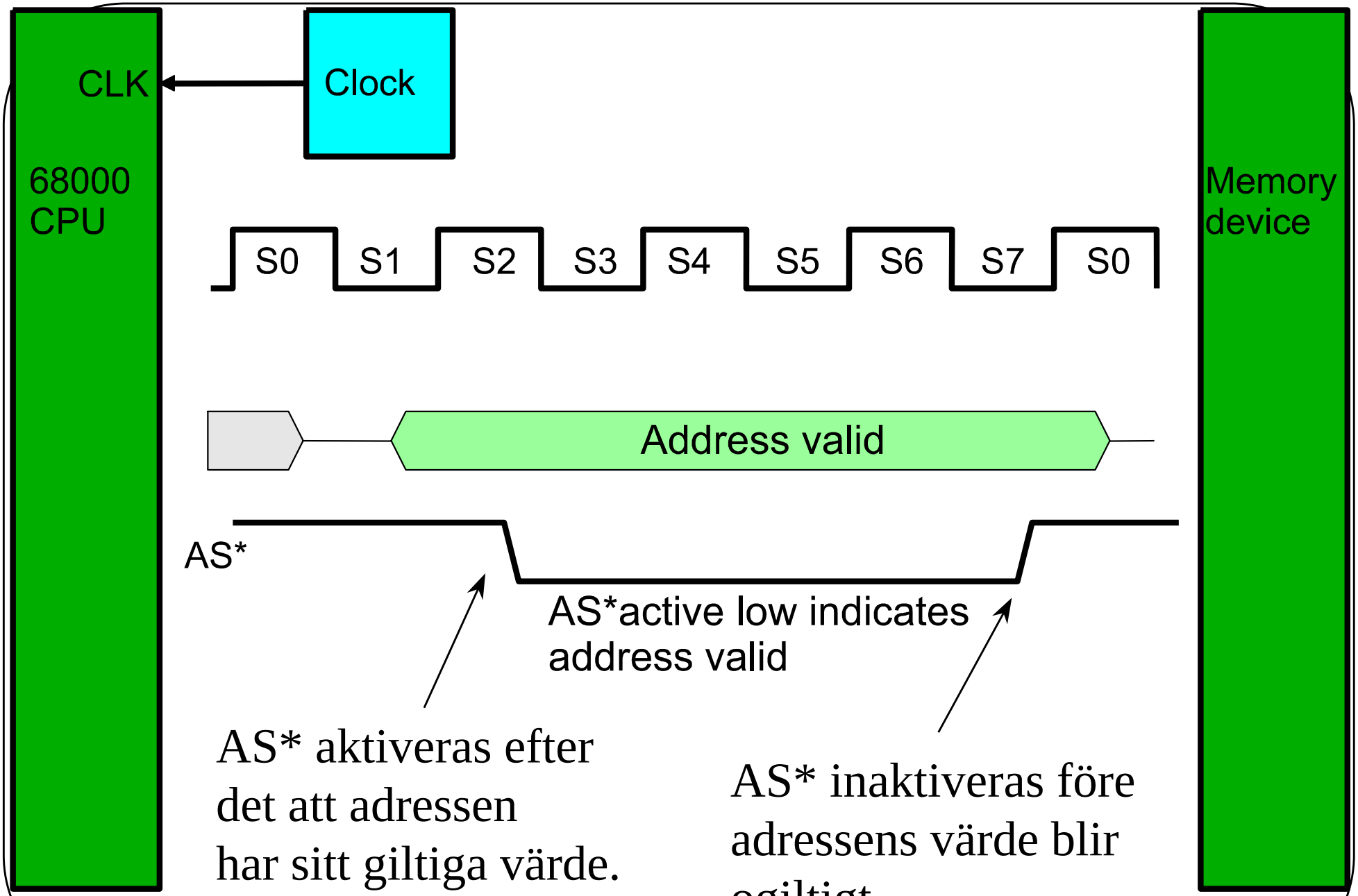


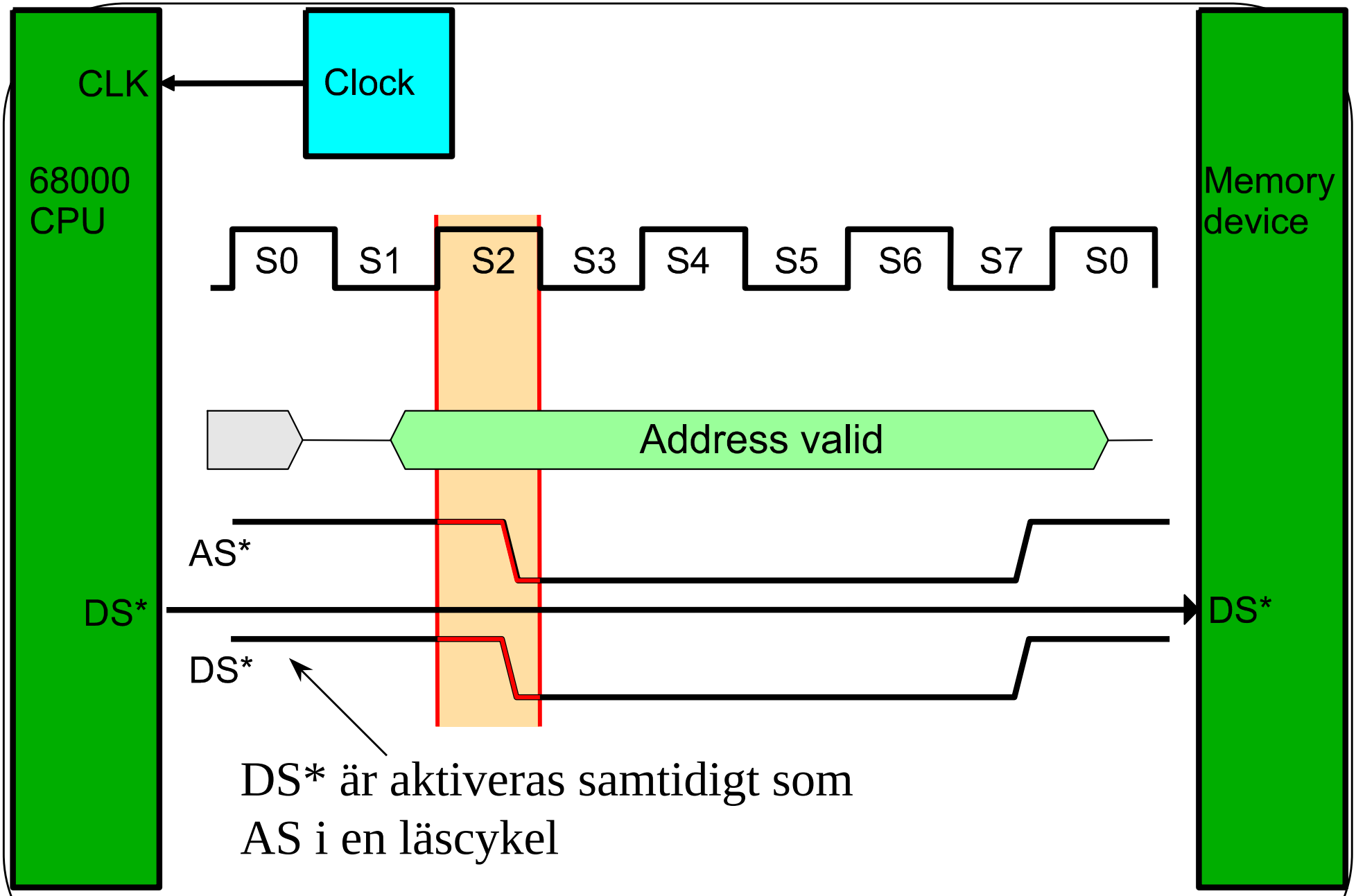
I tillstånd S1 blir den nya nya adressen giltig, den för blir giltig under hela minnes accessen.

I tillstånd S0 kommer den gamla adressen att vara giltig på adr. bussen

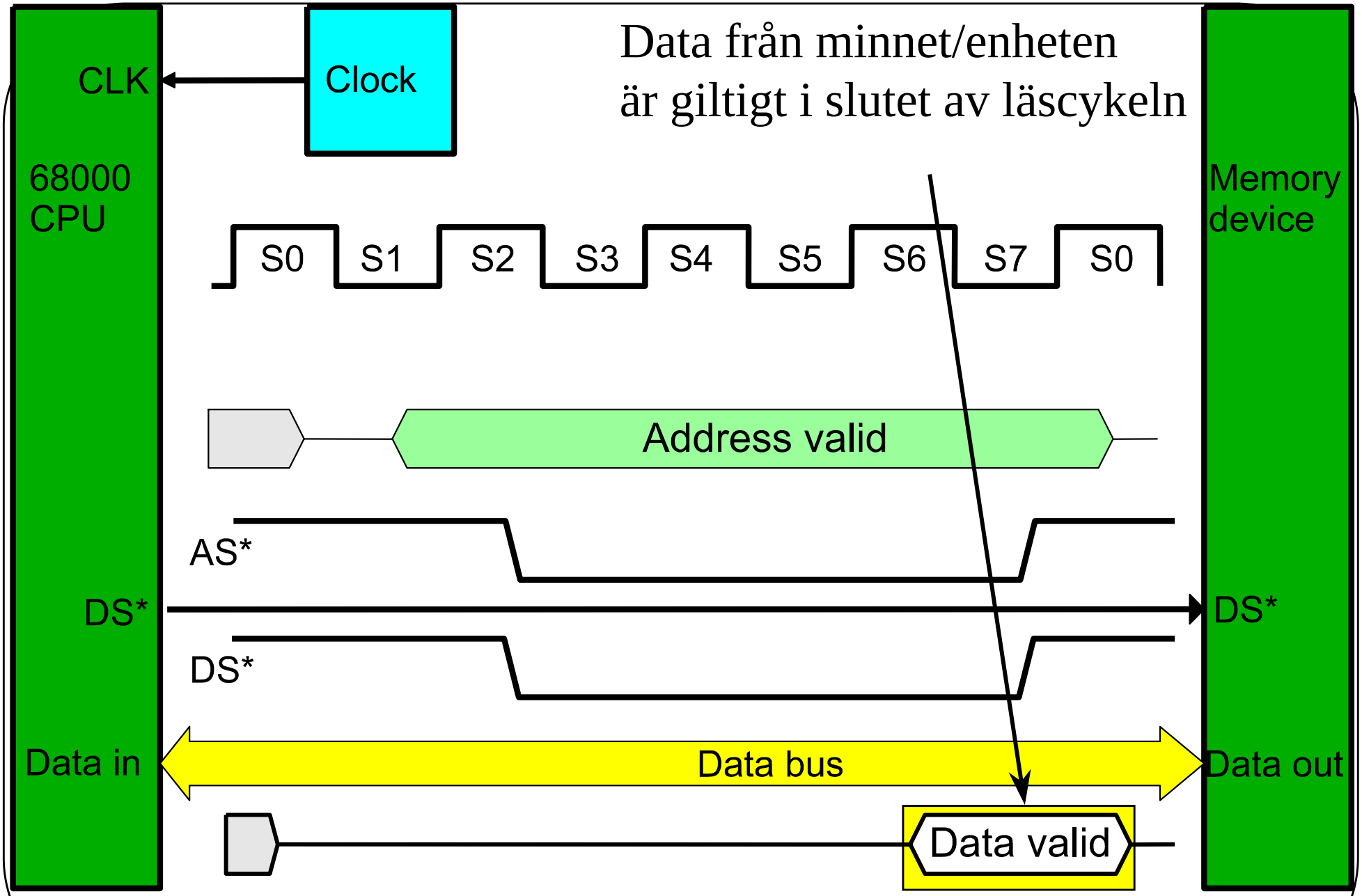


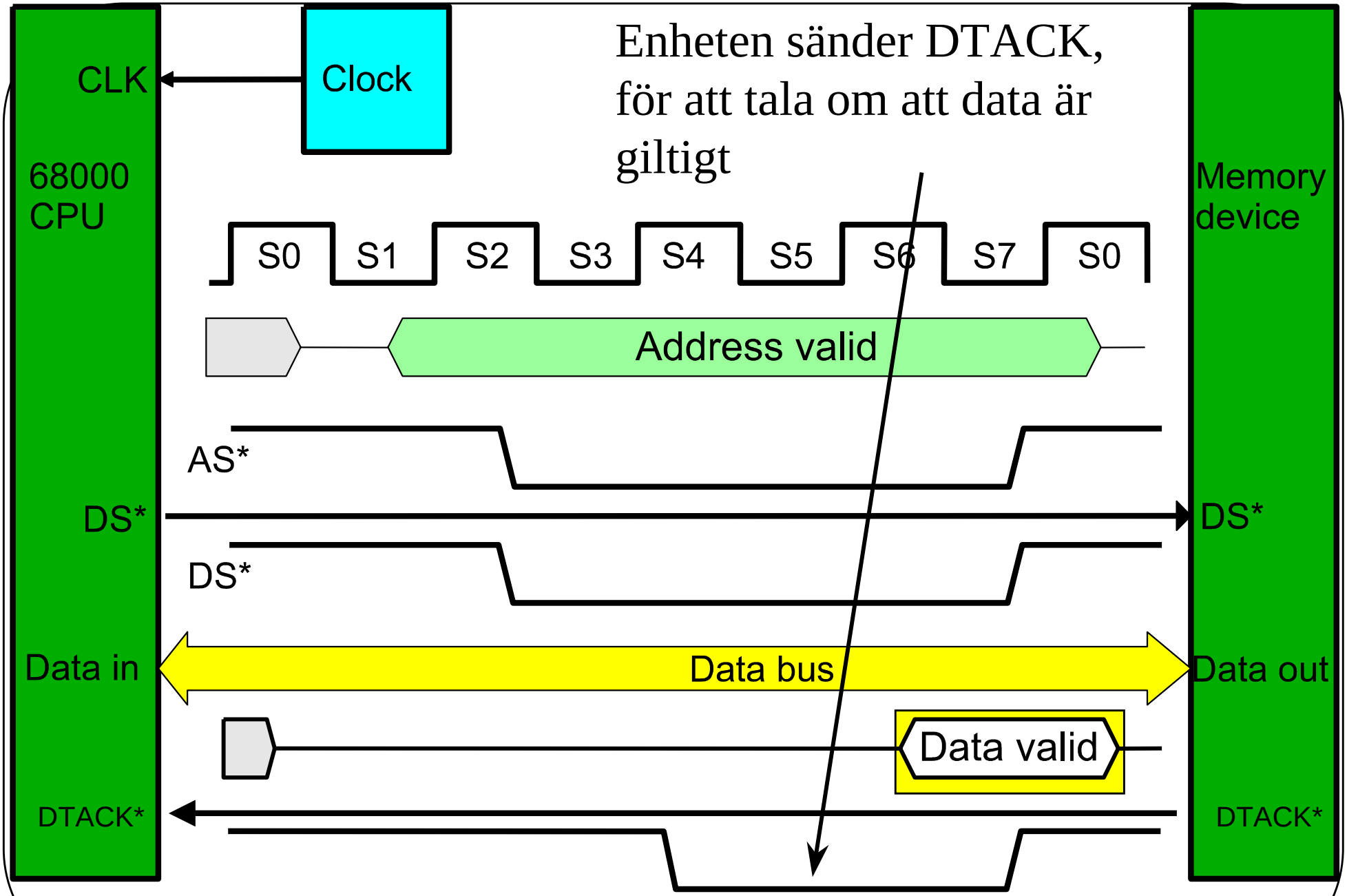
För att den adresserade enheten ska veta när adressen på adressbussen har ett giltigt värde, aktiverar CPUn signalen AS (Adress Strobe).

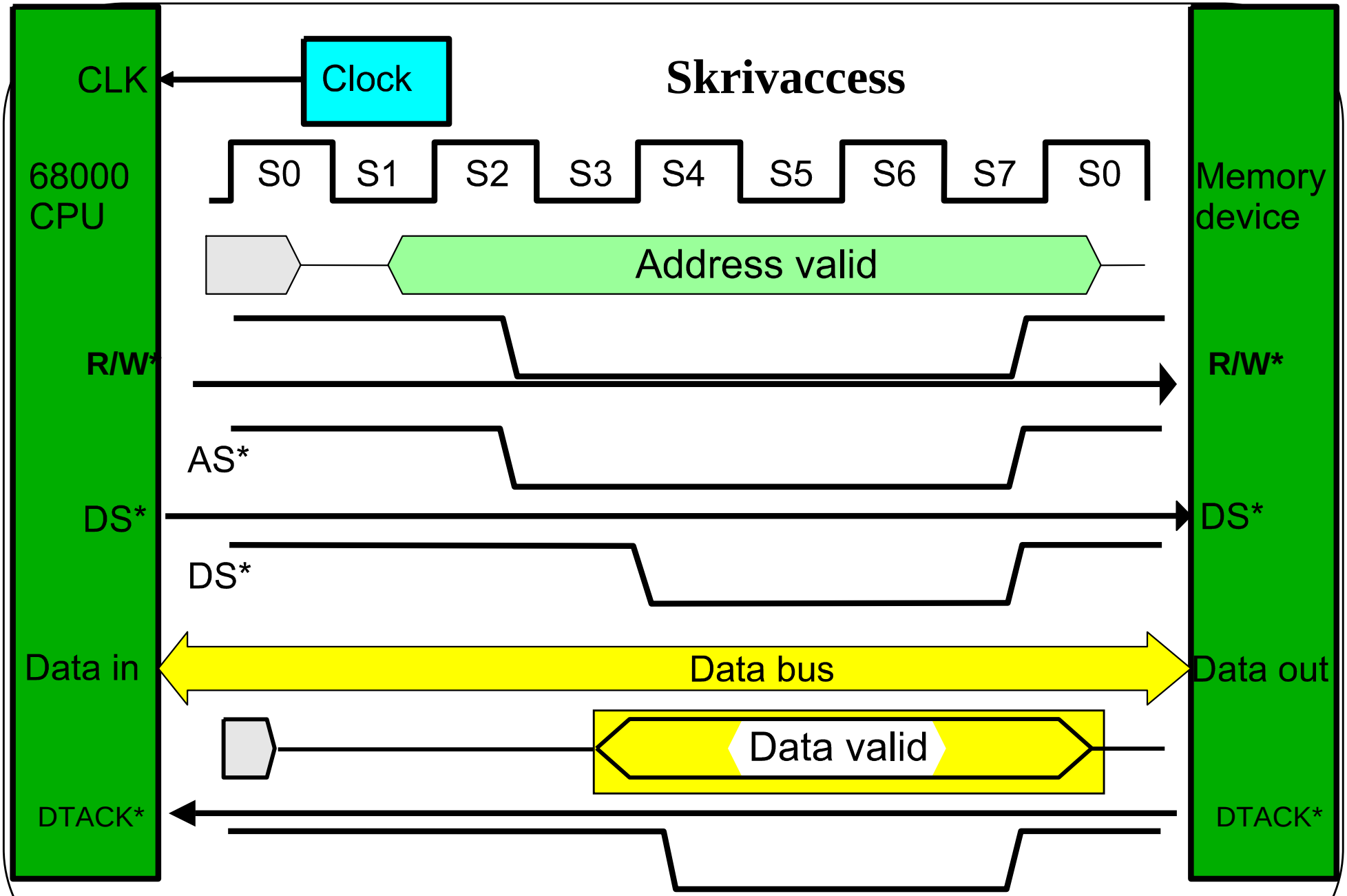




DS* är aktiveras samtidigt som AS i en läscykel







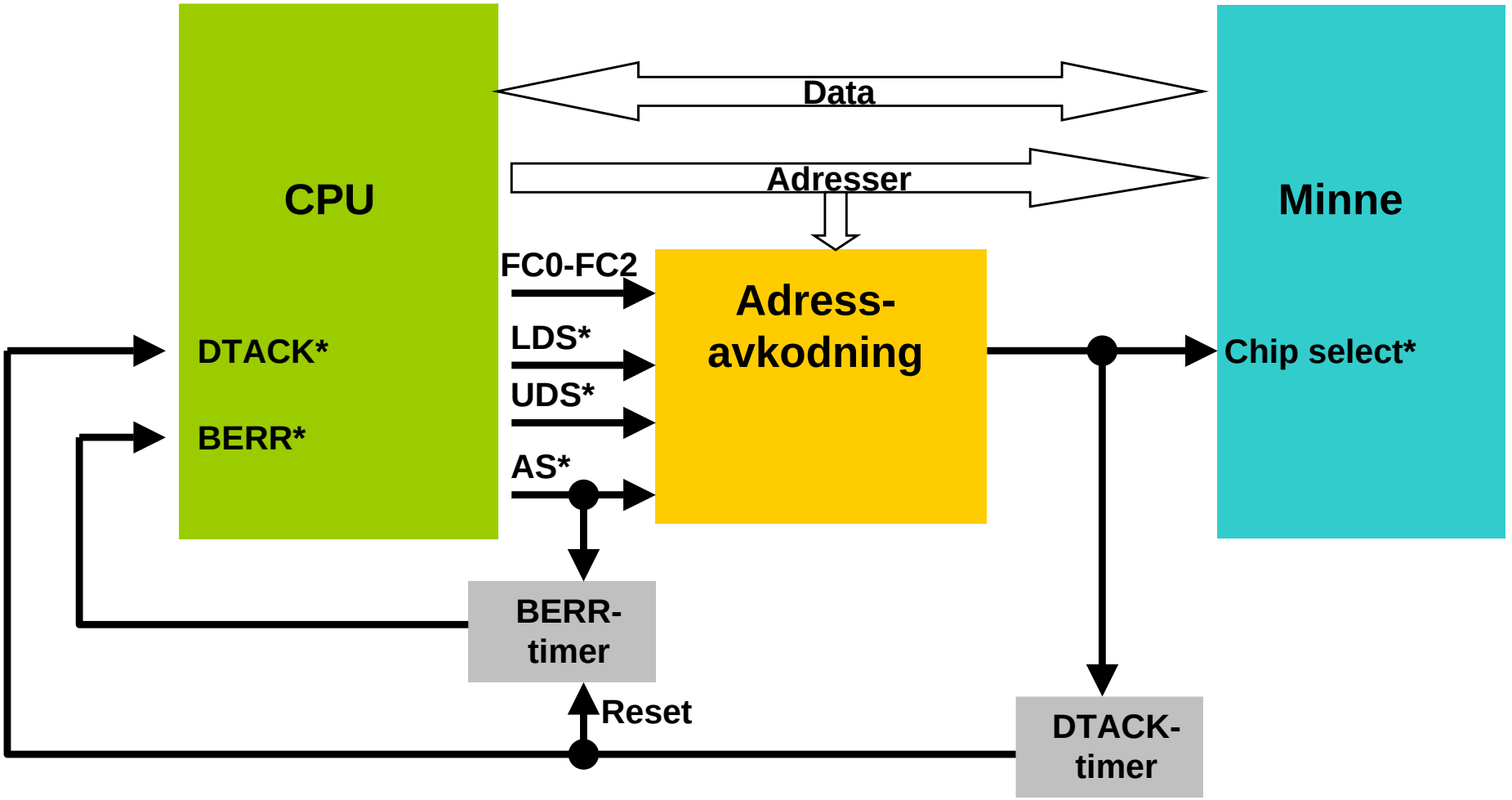
DTACK* generering

- **Med hjälp av DTACK signalen kan man dynamiskt bestämma hur lång tid en slavenhet behöver för att genomföra accessen.**
  - **Genom att förlänga tiden till dess DTACK signalen aktiveras, kan slavenheten införa wait-states**
  - **Hur många wait-states som behövs bestäms av timingen för den enheten som accessas kombinerat med klockcykel tiden.**
- **Dvs. olika snabba enheter kan kopplas till samma buss.**

BERR* generering

- **Om minne adresseras som ej är fysiskt närvarande i systemet så kommer processorn att låsas.**
  - **Ingen DTACK ger oändligt många wait-states.**
- **För att ex. upptäcka ett sådant kritiskt fel så har Motorola infört en timeout-signal, BERR***
- **Tiden innan BERR* aktiveras skall vara längre än minnet med längst accesstid.**

DTACK* och BERR*



Synkron busscykel

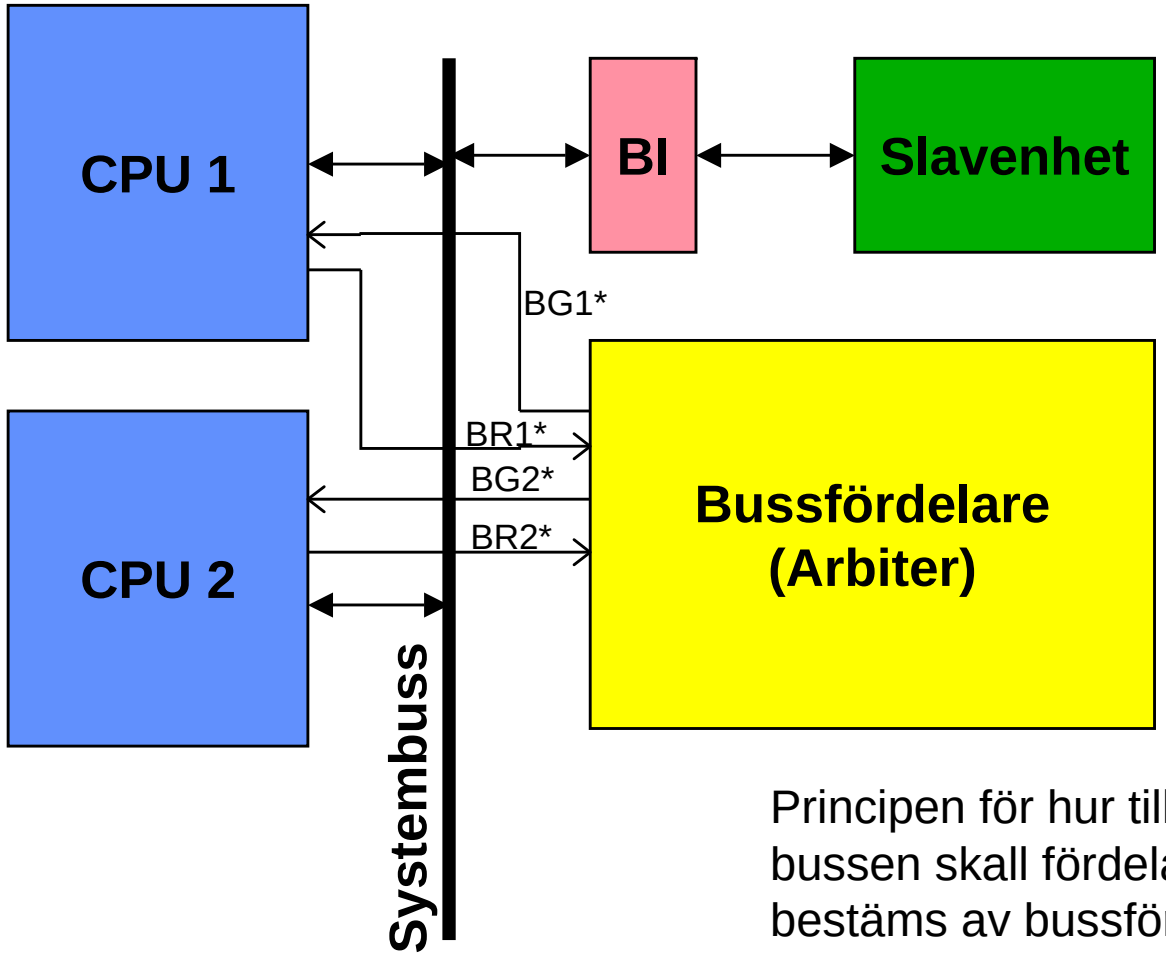
- **VPA* Om den periferikrets som adresserats har ett synkront interface så begär periferikretsen en synkron överföring genom att aktivera VPA***
- **Processorn aktiverar då VMA* för att indikera att adressen är giltig och att överföringen kan synkroniseras med E-signalen.**
- **E är en klocksignal som har 10 ggr så stor periodtid som systemklockan. E är gemensam för alla synkrona periferienheter.**

Bussfördelningskontroll 68000

- **När vill man fördela bussen ?**
  - **Om datorsystemet innehåller två CPU**
  - **Någon I/O enhet vill ha access till minnet utan att gå omvägen via CPU:n, t.ex. vid DMA.**

- **Detta sker med signalerna**
  - **BR*, Bus Request**
    - **Ansökan om att få kontroll över bussen (Bli Master)**
  - **BG*, Bus Grant**
    - **Tillstånd att få kontroll över bussen (Enheten har fått tillstånd att bli Master)**
  - **BGACK*, Bus Grant Acknowlege**
    - **Enheten accepterar att ta kontroll över bussen (Enheten är Master)**

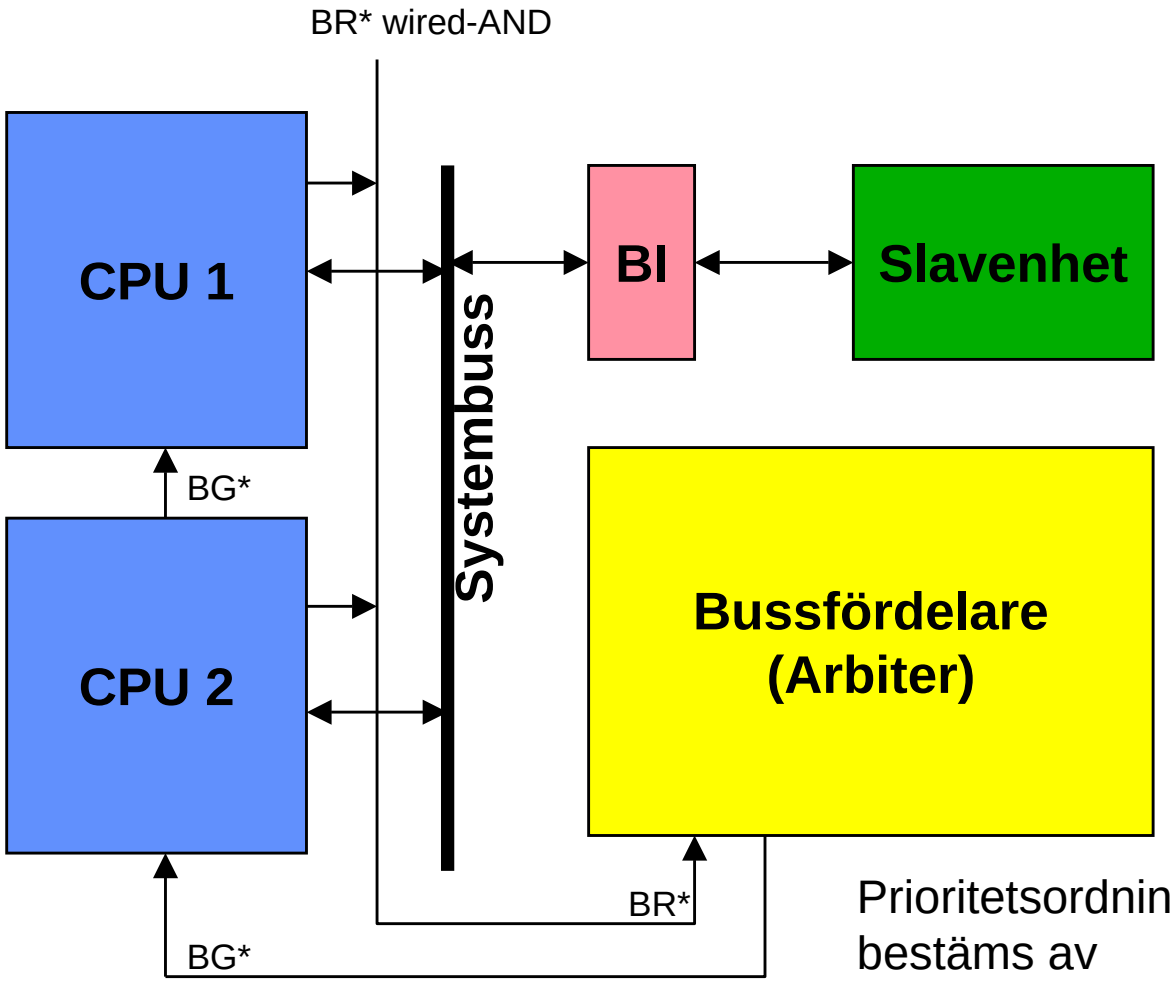
Centraliserad bussfördelning



Principen för hur tillgång till bussen skall fördelas bestäms av bussfördelaren.

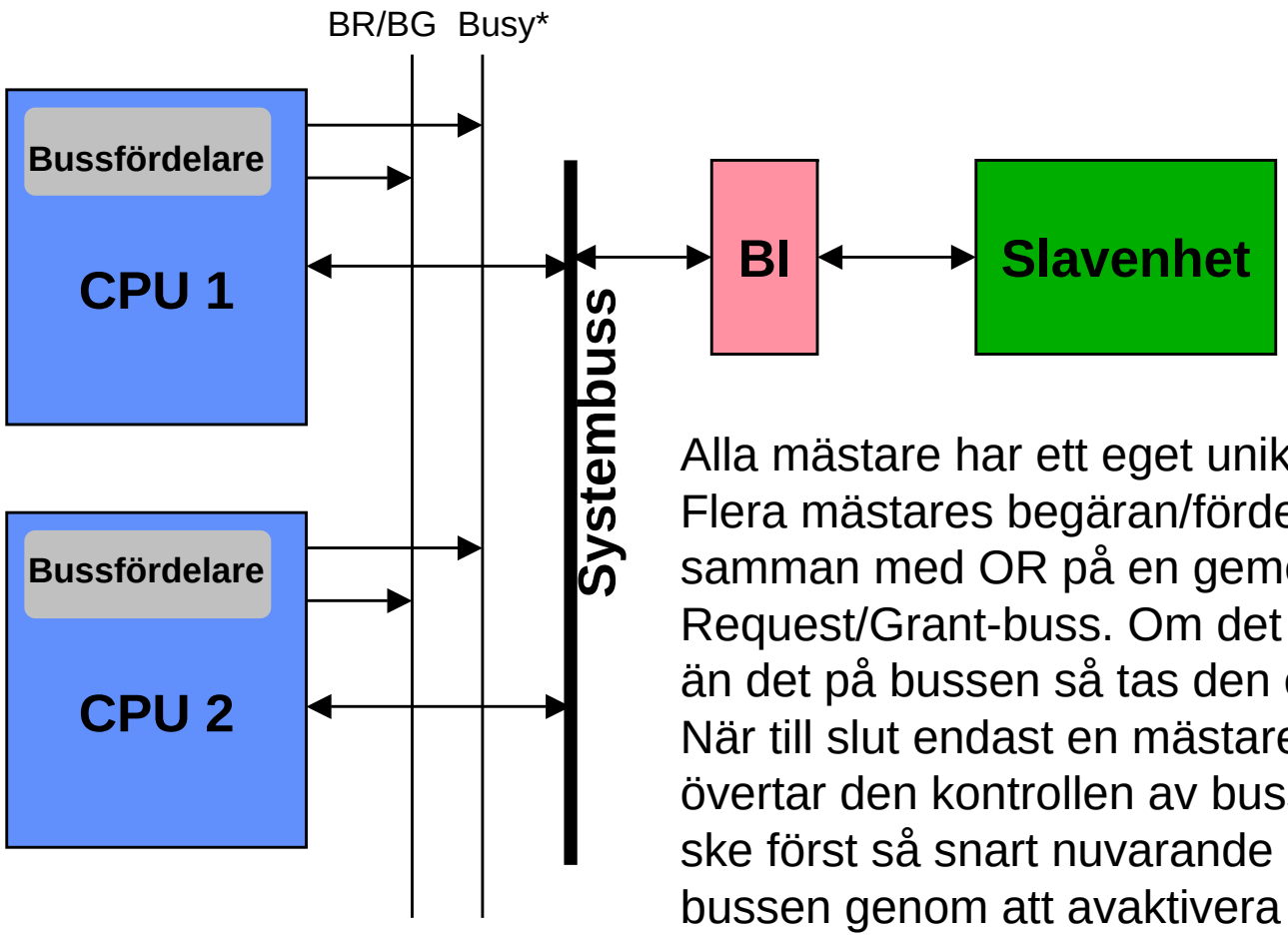
Centraliserad bussfördelning – Daisy chain

Daisy chain innebär att den första i kedjan som begärt tillgång till bussen bryter kedjan.



Prioritetsordningen bestäms av kedjan.

Distribuerad bussfördelning



Alla mästare har ett eget unikt fördelningsnummer. Flera mästares begäran/fördelningsnummer sätts samman med OR på en gemensam Request/Grant-buss. Om det egna numret är lägre än det på bussen så tas den egna begäran bort. När till slut endast en mästare begär bussen, så övertar den kontrollen av bussen. Detta kan dock ske först så snart nuvarande bussmästare släpper bussen genom att avaktivera Busy*.

Statisk bussfördelning

- **Oavsätt om hårdvaran för bussfördelning distribuerats på alla bussmästare eller centraliserats, så måste systemet anta någon princip för hur potentiella bussmästare skall få tillgång till resp. släppa kontrollen av systembussen.**
- **Statisk bussfördelning innebär att ett schema för när och hur länge respektive bussmästare skall få tillgång till systembussen har förutbestämts.**
- **En nackdel med detta är att även om en mästare ej behöver bussen så får den ändå tillgång till den.**

Principer för dynamisk bussfördelning

- **Allokering av bussen**
  - **Prioritet**

    Varje bussmästare har en unik prioritet. De med lägre prioritet kan bli utan tillgång till bussen.
  - **Rättvisa**

    Alla bussmästare måste ha fått tillgång till bussen innan en mästare kan få tillgång till bussen igen.
- **Deallokering av bussen**
  - **Släpp på begäran**

    Behåller bussen tills någon annan bussmästare begär tillgång.
  - **Släpp när klar**

    Släpper bussen efter varje transaction.
  - **Pre-emption**

    En bussmästare med högre prioritet kan avbryta en mästare med lägre.