

## 3. Busser

### Hvad er en bus?

---

- En forbindelse mellem komponenter
- Både interne og eksterne
  - **Internt mellem system komponenter**
    - \* SPI, I<sup>2</sup>C - Seriel
  - **Eksterne komponenter**
    - \* eSATA
    - \* USB(Universal Serial Bus)
- Kan også benyttes internt i en chip
  - Mange parallelle forbindelser kan indgå
  - Høj båndbredde
  - F.eks. internt i en CPU
- Busser bliver ofte delt mellem flere komponenter, med kun en CPU er delte busser lige så effektive som separate busser

### Hvilke signaler benytter man typisk på en parallel bus?

---

- På en typisk microprocessor:
  - Select Address on slave device
  - Data to/from slave device
  - Select, read, write and wait signals
  - Slave interrupts
  - Multi-master Control signals(Daisy Chaining, device tættest på får adgang først)
  - Special Handshaking signals
  - Misc. signals

### Hvordan adresserer man på en bus?

---

- Der benyttes en adresse bus
- For at kunne adressere 64K ( $2^{16}$ ) memory, skal der benyttes 16 adresse linjer
- READ/ fra host, er forbundet til OE/(output enable) på modtageren
- WRITE/ til WE/(write enable)
- Hosten ved reelt ikke hvad der sidder på hvilke adresser
  - Derfor vil "address spaces" som regel være delt op til hver enhed
  - F.eks. en opdeling mellem RAM - mest betydende bit altid 0, og ROM - mest betydende bit altid 1
- (Byte-enable(NBE0, NBE1) kan benyttes til at læse en halv atomic memory block)

### Hvilke ting skal konfigureres på en host for at kunne tale med et device på en bus?

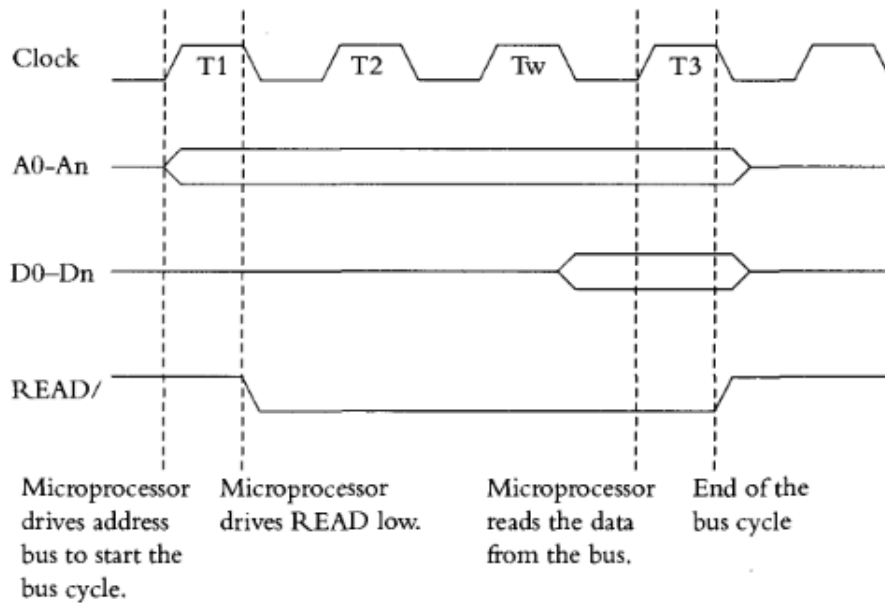
---

- Adresse
  - Adressen på bussen
  - Chipselect til den specifikke enhed på bussen
- Read/Write - host skal vide om der ønskes at læse fra eller skrive til devicen
- Timing, Clockhastigheden skal være den samme på host og device

### Beskriv et læse/skrive scenarie og de timing parametre som indgår

---

- Wait states til langsomme eksterne enheder



*Note: Der kan være et wait signal sendt fra devicet til CPU'en*