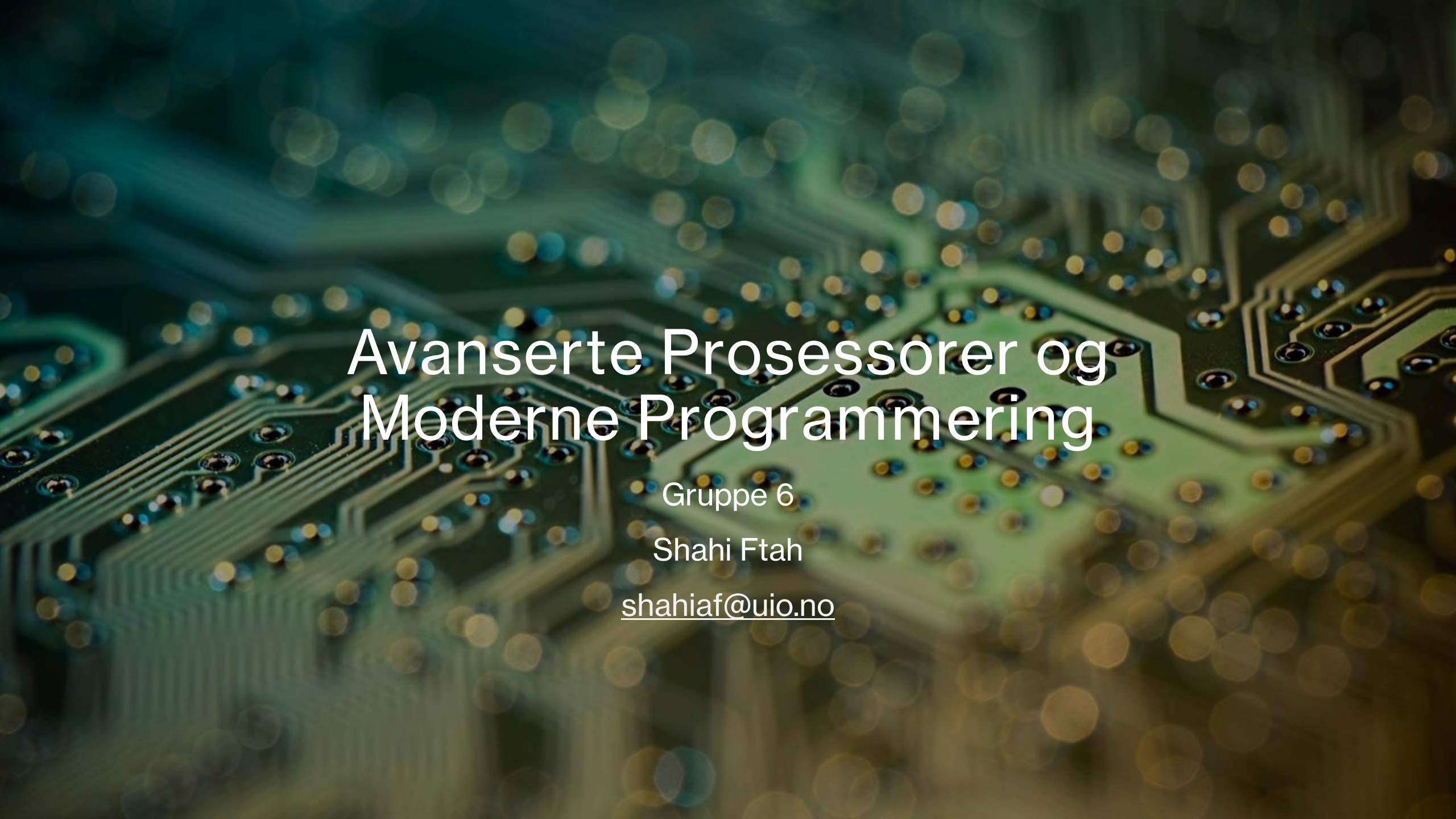


Avanserte Prosessorer og Moderne Programmering

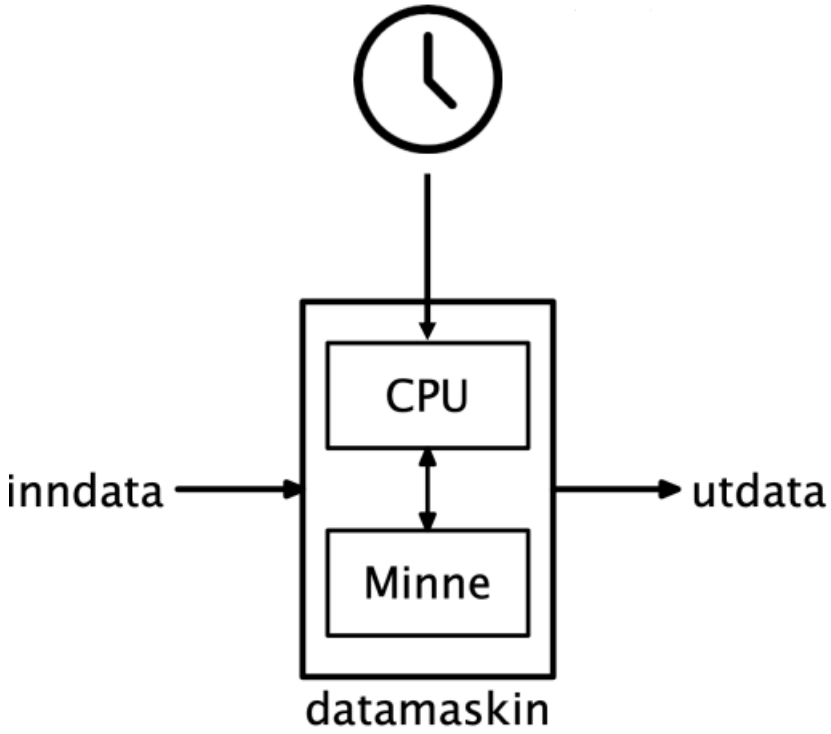
Gruppe 6

Shahi Ftah

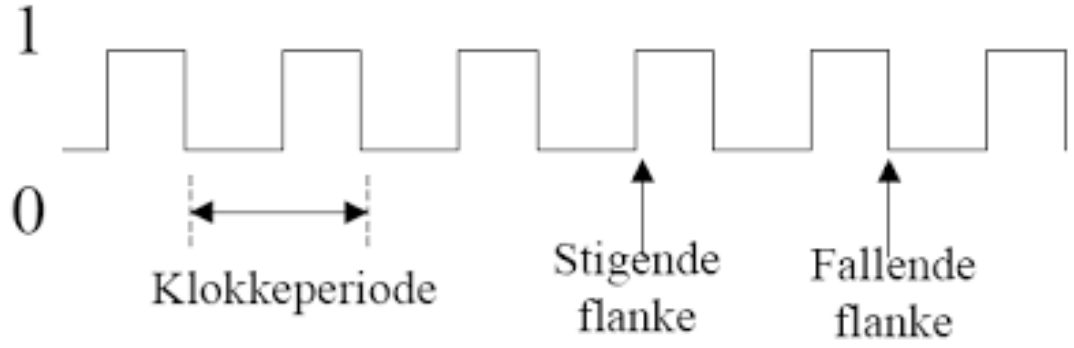
shahiaf@uio.no

Klokkefrekvens

- Sier noe om hvor raskt CPU arbeider
- Høyere klokkefrekvens = raskere CPU



- Eks. 20 Hz – Klokka slår 20 ganger per sek

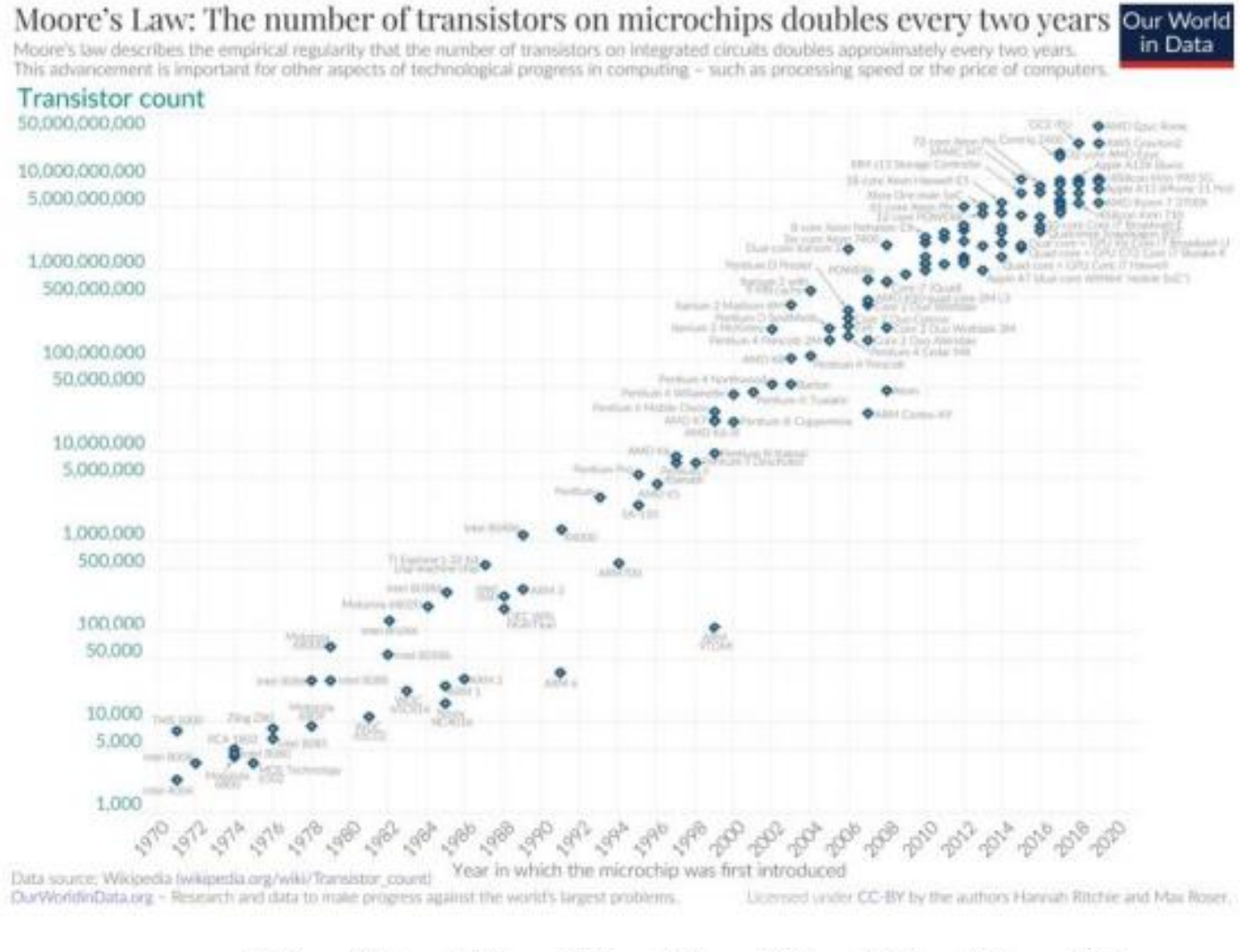


Prosessoren x86

- Intel sin x86 er dominerende
- Kan brukes med eldre systemer

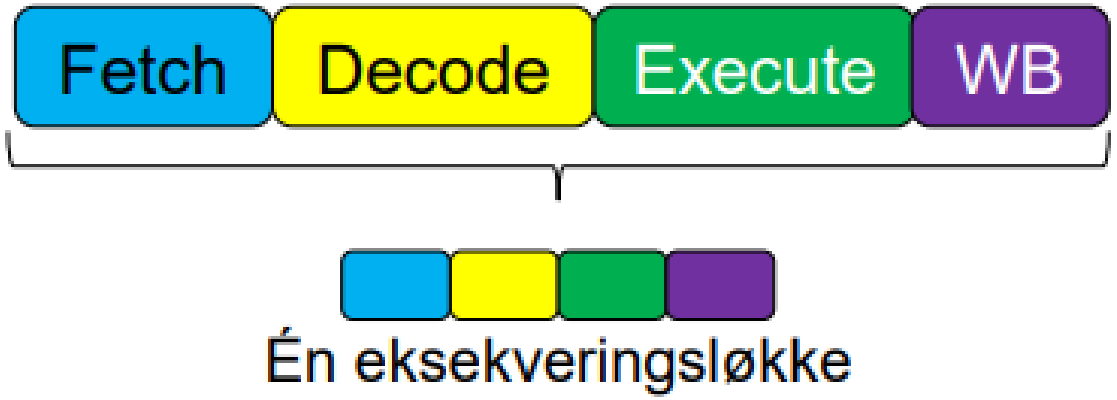
- ARM og AMD

- Gordon Moore Intel gründer
- Antall transistorer i en krets dobler seg annethvert år



Prosessoren x86-64 Og LMC

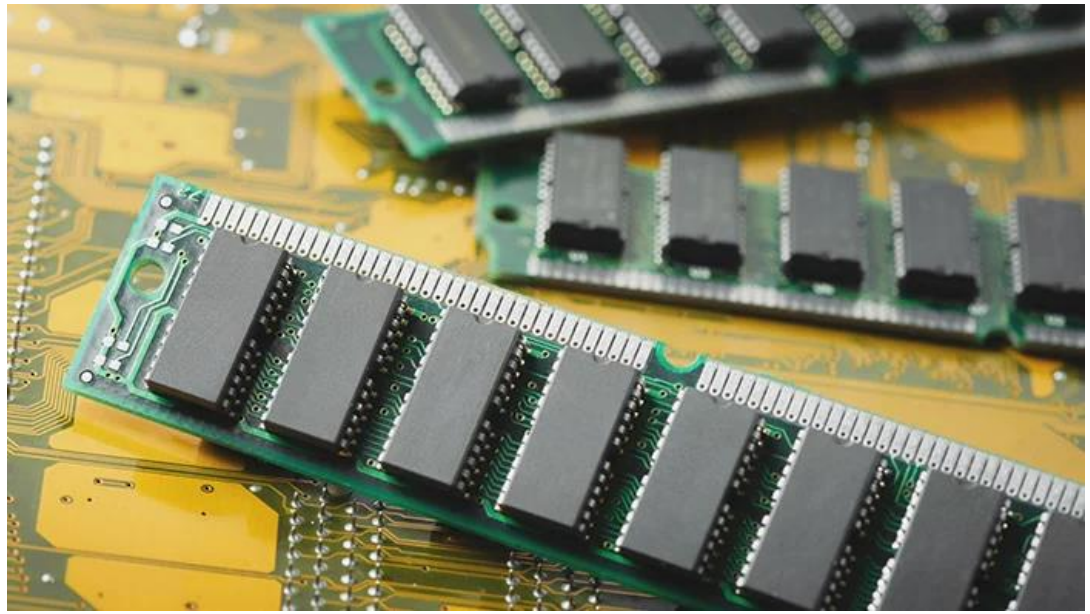
1. Hent instruksjonen i minnet; programtelleren angir hvor.
2. Dekod instruksjonen i feks. operasjon og adresse.
3. Utfør instruksjonen.
4. Om aktuelt, skriv svaret i minnet/register.



Raskere = bruk pipeline, se foiler fra forrige uke

RAM – RANDOM ACCESS MEMORY

- LMC har 100 celler ca. 138byte (11bit pr celle)
- X86-64 har 2^64 celler, kan inneholde opptil = 17 179 869 184Gb = 17 179 869 TB
- Hvis det blir tomt? Flytte til harddisk med virtuell minne
- X86_64 - Virtuell minne – Kan bruke mer minne



CPU og LMC

- LMC er RISC («Reduced instruction set computer») maskiner – begrepet antall enkle instruksjoner
- X86-64 er CISC («Complex instruction set computer») maskiner – komplekse instruksjoner

```
a[i] += 1
```

x86-64:

```
leaq    a, %r8
movq    i, %r9
incq    (%r8, %r9, 8)
```

LMC:

```
        LDA     LDA_A_i
        ADD     i
        STA     INCR_i
        LDA     STA_A_i
        ADD     i
        STA     INCR_i2
INCR_i  DAT     0
        ADD     v1
INCR_i2 DAT     0
        :
LDA_A_i LDA     A
STA_A_i STA     A
v1      DAT     1
i       DAT     0
```

1.5 LMC-1

```
          INP
          STA a
          INP
          BRZ print
          LDA a
          SUB b
          STA a
print     LDA a
          OTC
          HLT
a         DAT
b         DAT 32
```

Hva skrives ut når denne koden kjøres og bruker gir følgende inndata: **114** og **999**?

Velg ett eller flere alternativer

- [ ] r
- [ ] T
- [ ] 999
- [ ] R
- [ ] q

- R = 82
- r = 114
- q = 113
- T = 84

- Hva skjer hvis brukeren gir 7 og 11 som input?

```
start   INP
        STA     x
        INP
        STA     y
        LDA     x
        SUB     y
        OUT
slutt   HLT

x       DAT     0
y       DAT     0
```



- [ ] Ingenting skrives ut
- [ ] -4
- [ ] 11
- [ ] 0
- [ ] 4
- [ ] 18

- Hvis vi sender inn tallet 7 og 3, hva skjer i programmet?
- Programmet skriver ut 4

- Hva skjer hvis vi sender inn tallet 10 og 10?
- Programmet vil ikke skrive ut noe

```
INP
STA NUM
INP
SUB NUM
BRZ ZERO
OUT
HLT
NUM DAT 0
ZERO DAT 0
```