

# Week 13 Instruksjon og operander

## Instruksjon og operander

**Korleis instruksjonar er bygd opp (type)**

**Korleis handteras operandar (adrModi)**

**Korleis utføres instruksjonen (operandar)**

**Mot minne**

**Interne register**

## Bit i instruksjon (lengde og koding)

**Hvordan er instruksjoner oppbygget (type instruksjoner)**

Måten instruksjonen er bygget opp på av oPcode og operander. Dette kaller vi **adresseinstruksjoner**. Hvor mange operander blir adressert i en instruksjon?

### **0-adresseinstruksjoner**

Har ingen operander. Kun oPcode.

### Operation: NOP

Function: None, waste time

Syntax: No Operation

Instructions	OpCode	Bytes	Flags
NOP	0x00	1	None

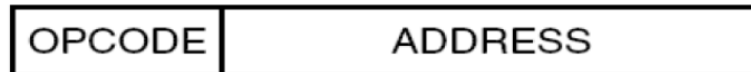
Description: NOP, as it's name suggests, causes No Operation to take place for one machine cycle. NOP is generally used only for timing purposes. Absolutely no flags or registers are affected.



- PUSH A
- PUSH B
- ADD
- PUSH C
- PUSH D
- ADD
- MUL
- POP X

Alle disse er 0-adresseinstruksjoner. Vi har ingen operander. Har bare en funksjon.

### 1-adresseinstruksjon



# 1-adresseinstruksjoner

**ADD A,direct**

Bytes: 2

Cycles: 1

Encoding: 

0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---

direct address
----------------

Operation: ADD  
 $(A) \leftarrow (A) + (\text{direct})$

**ADD A,@R<sub>i</sub>**

Bytes: 1

Cycles: 1

Encoding: 

0	0	1	0	0	1	1	i
---	---	---	---	---	---	---	---

Operation: ADD  
 $(A) \leftarrow (A) + ((R_i))$

Vi har en oPcode og en operandfelt.

Vi har en ADD A, direct. Vi har en verdi i adressefeltet.

ADD A, @R<sub>i</sub> vi adresserer et regiser. Vi får verdien i et register. Registeret i er gitt i det siste bitet (0/1)

## 2-adresseinstruksjoner



# 2-adresseinstruksjoner

**MOV <dest-byte>,<src-byte>**

Function: Move byte variable

Description: The byte variable indicated by the second operand is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.

En instruksjon av oPcode og to adresseinstruksjoner. Vi har to operander.

Operandene over i MOV kopierer en byte fra en plass til en annen.

## 3-adresseinstruksjoner

OPCODE	ADDR1	ADDR2	ADDR3
--------	-------	-------	-------

## 3-adresseinstruksjoner

- $X = (A + B) * (C + D)$**
- T1 og T2 brukes til mellomlagring (kan være i hovedlager eller registre)
  - ADD T1, A, B
  - ADD T2, C, D
  - MUL X, T1, T2
  - Hvis ADDR1-3 er hovedlageradr., trenger vi ikke registre

Vi har tre adressefelt. Vi kan ha minneadresse eller register i registerfeltene våre.