Velkommen!

Oppsummering datamaskinarkitektur, Historien om datanettverk

Gruppetime 9 – 24.10.2021

(Hanne-)Kristin Schrøder Leiros

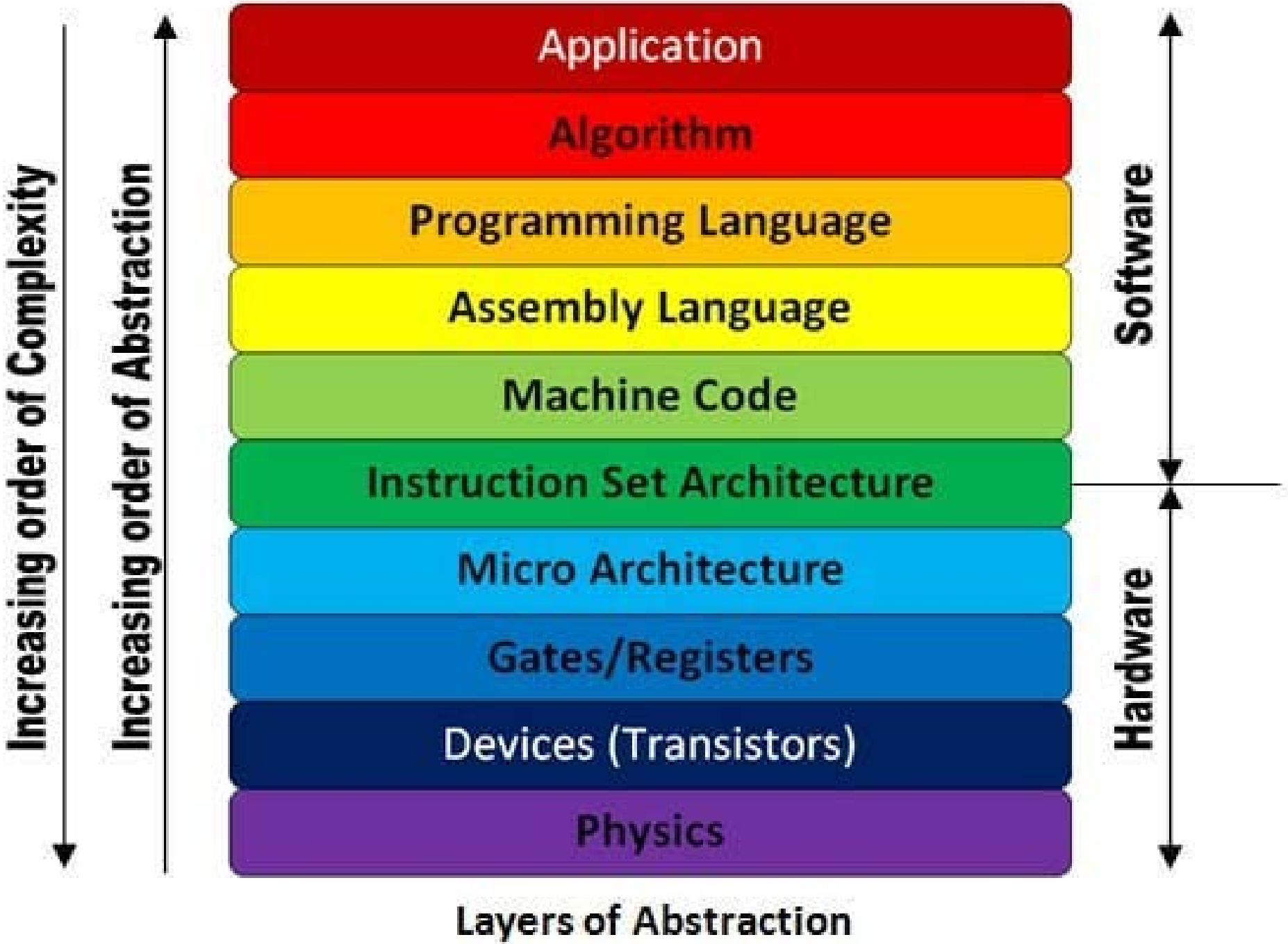
hkleiros@uio.no

Plan for dagen

- Oppsummering datamaskinarkitektur
- Tidligere eksamensoppgaver fra 2019 om datamaskinarkitektur
- Repetisjon av ukas pensum
- Ukesoppgaver 5 stykk

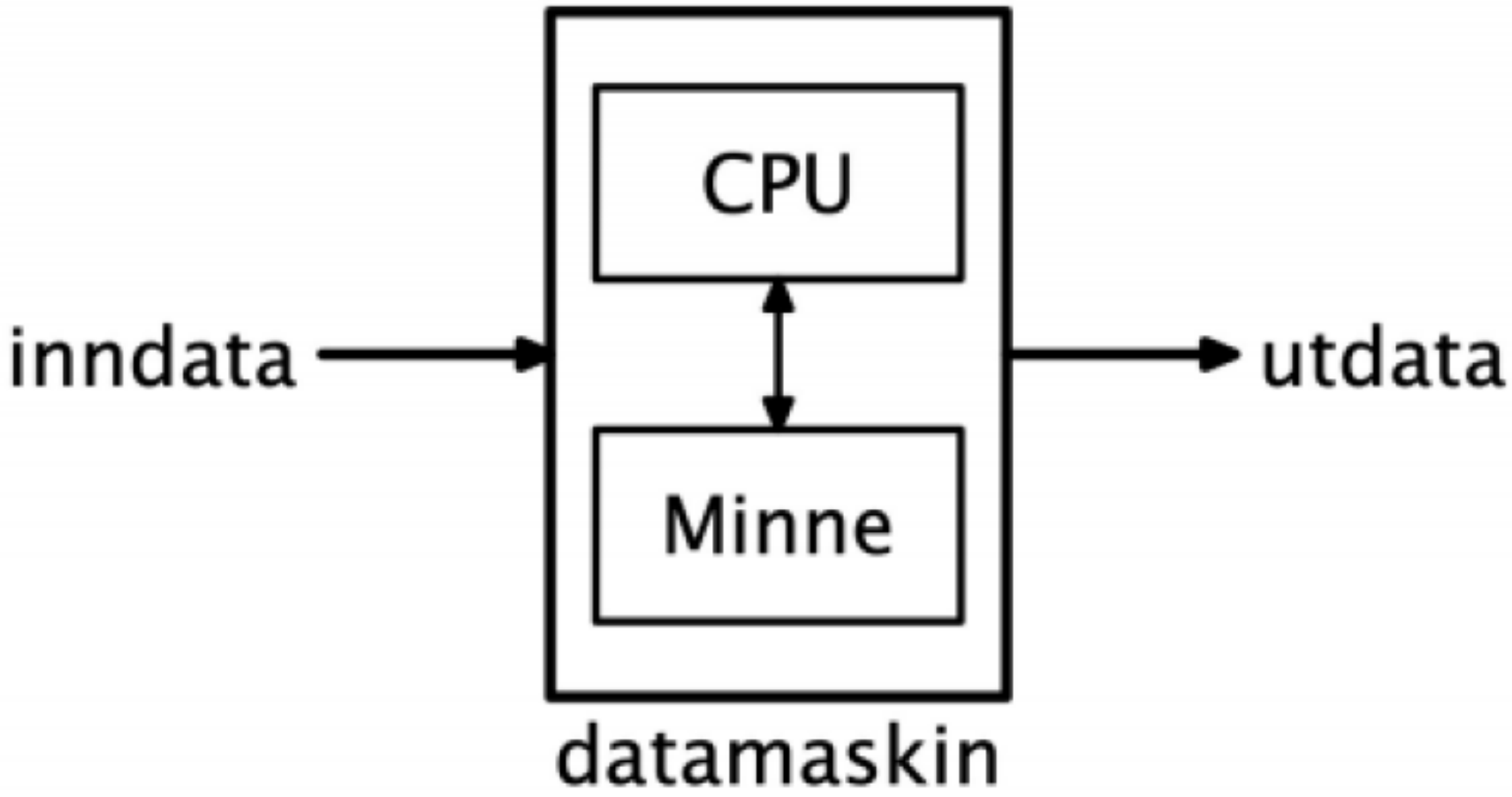
Datamaskinarkitektur

- Maskinvare og Abstraksjonsnivå
- Datamaskinarkitektur og ALU
- Porter og funksjoner
- Fysiske begrensinger og synkronisering
- Pipeline og minnehierarki


Increasing order of Complexity
Increasing order of Abstraction
Application
Algorithm
Programming Language
Assembly Language
Machine Code
Instruction Set Architecture
Micro Architecture
Gates/Registers
Devices (Transistors)
Physics
Software
Hardware
Layers of Abstraction

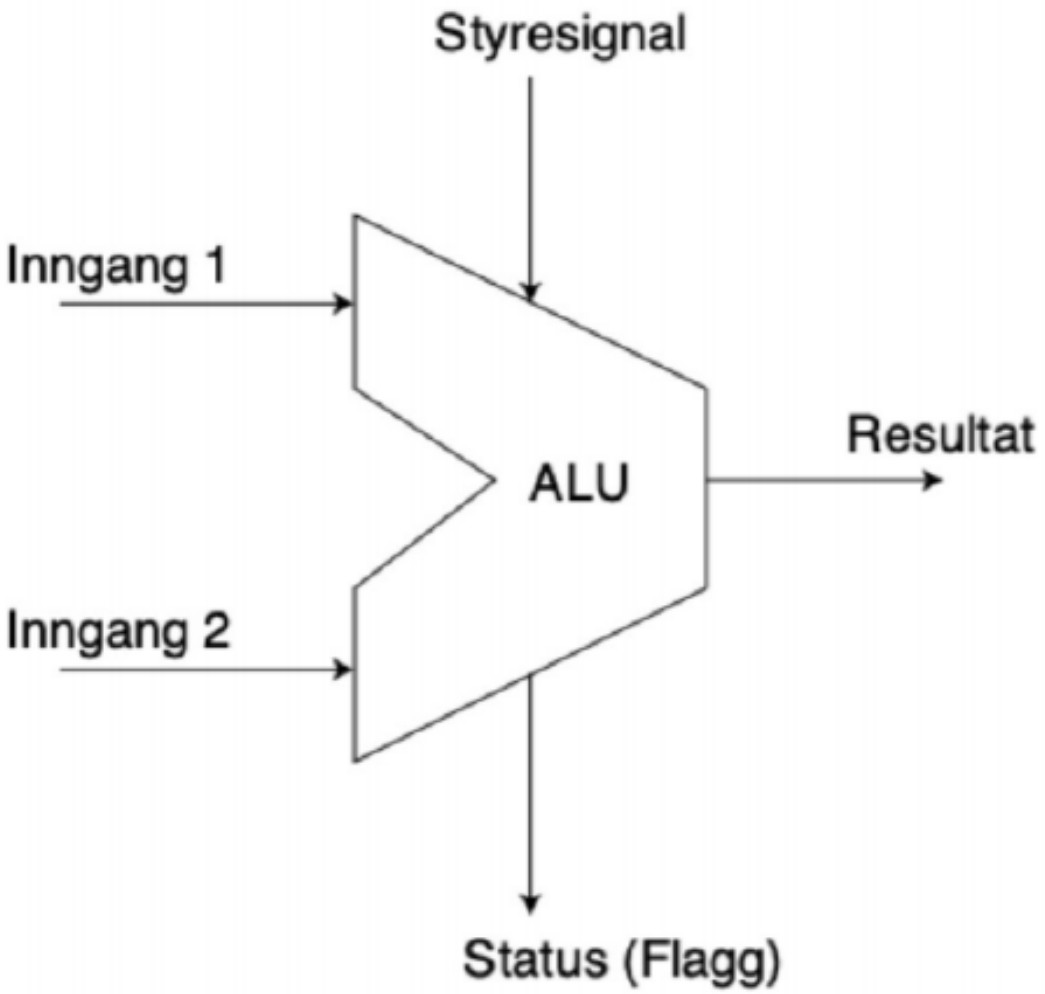
Applikasjoner	3D-Rendering program, Word, Zoom, etc. Operativsystem er det nederste «laget» av applikasjons-laget.	Software
Algoritmer	Hvordan et applikasjon skal kjøres elns.	
Programmeringsspråk	Java, Python, etc.	
Assembly kode	Simpleste koden som ikke består av kun binære tall. Varierer basert på hvilken maskin som brukes.	
Maskin kode	Binære tall som maskinen kan «forstå». Antall binære tall er avhengig av antall bits som er avhengig av maskin-arkitektur.	Fra SW Til HW
Instruction Set Architecture	En abstrakt modell av en datamaskin som beskriver hva maskinen består av. data typer, registre, etc.	
Micro Architecture	Kretsløpet i en datamaskin	Hardware
Gates/Registers	Enheter som utfører boolske funksjoner. AND, NOT, OR, XOR,	
Devices (Transistors)	Spenning, ikke spenning	
Fysikk	HvA eR sPeNiNg	

Von Neumann arkitektur



ALU – Aritmetic Logic Unit

- Brukes til aritmetiske og logiske beregninger
- Input: to for data, en instruksjon
- Output: resultat og status
- Viktigste delen av CPU-en



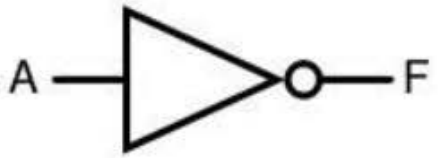
Sannhetsverditabeller og porter

- En sannhetsverditabell beskriver de mulige resultatene av alle kombinasjoner i en port eller krets
- Én sannhetsverditabell kan ha mange funksjonsbeskrivelser
  - (sammensetninger av porter)
- Én funksjon kan bare ha én sannhetsverditabell

INV - port

$F = A'$

A	F
0	1
1	0



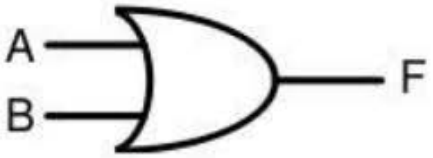
Invert - snur om inputet

OR - et eller begge tall er 1

OR - port

$F = A+B$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

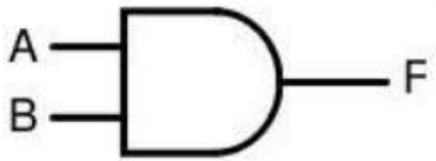


AND - begge tallene er 1

AND - port

$F = AB$

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

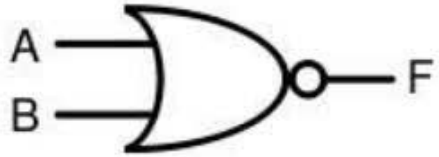


Not OR - ingen av tallene er 1

NOR - port

$F = (A + B)'$

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

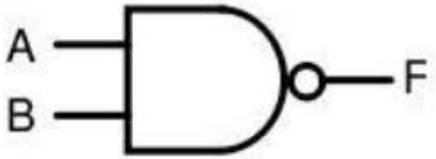


Not AND - minst ett tall er ikke 1

NAND - port

$F = (AB)'$

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

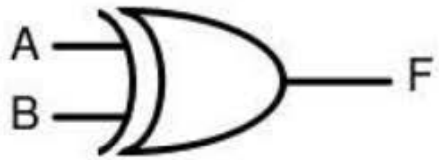


eXclusive OR - ett av tallene er 1, men ikke begge

XOR - port

$F = A \oplus B$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

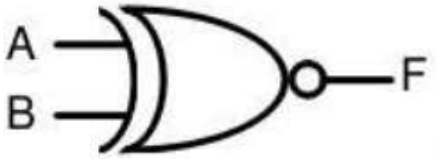


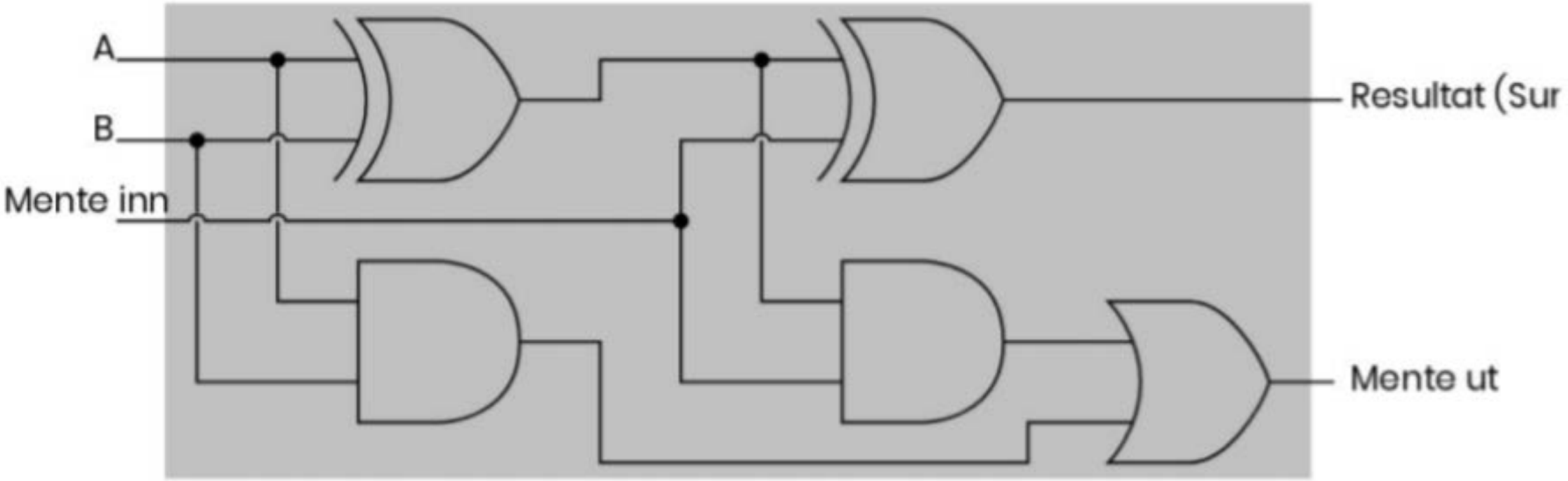
eXclusive Not OR - begge tall har samme verdi

XNOR - port

$F = (A \oplus B)'$

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1





FULL-ADDER (FA)

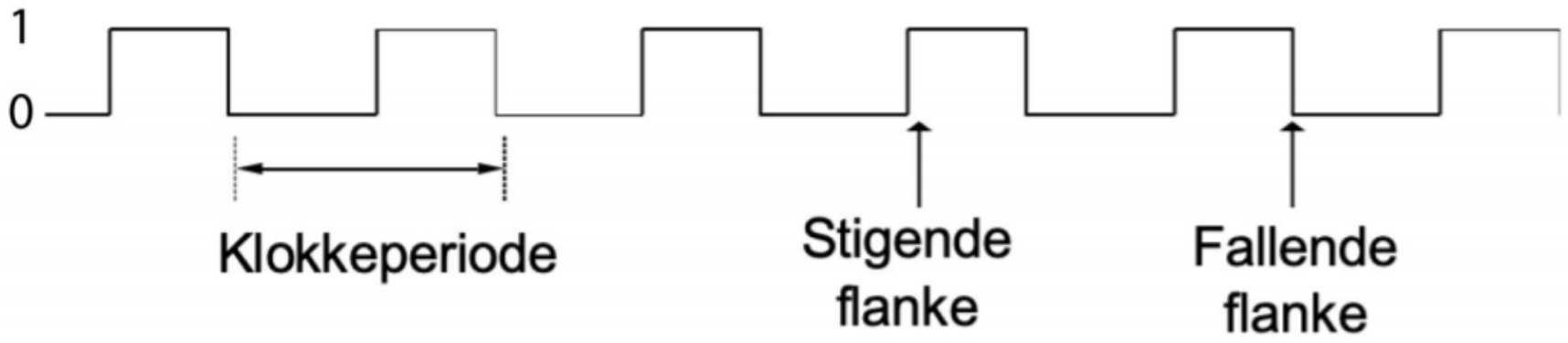
Addere

- Brukes for å kunne utføre binær addisjon
- Halv-adder kan legge sammen to tall , men gir ikke om vi har mente inn også
- Derfor setter vi sammen flere Logiske Porter og får ne Full-Adder, som også tar mente inn
- Hvis vi kombinerer fire Full-Addere kan lage en Seriell Adder, som kan legge samme 4-bit tall.

Synkronisering og klokkesignal

- Synkronisering brukes for å hindre at forskjellige komponenter er ferdig til forskjellig tid
- Stilles etter den tregeste komponenten
- Globalt klokkesignal i CPU-en
- Lokalt klokkesignal for mindre deler

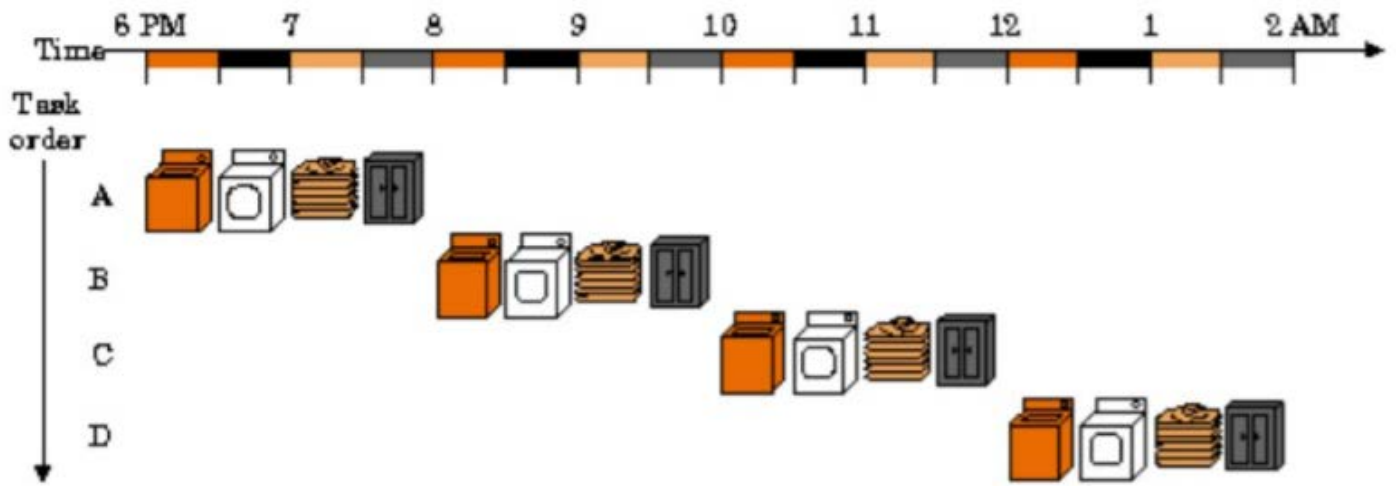
Et klokkesignal er et digitalt signal som veksler mellom ´0´ og ´1´ med fast takt

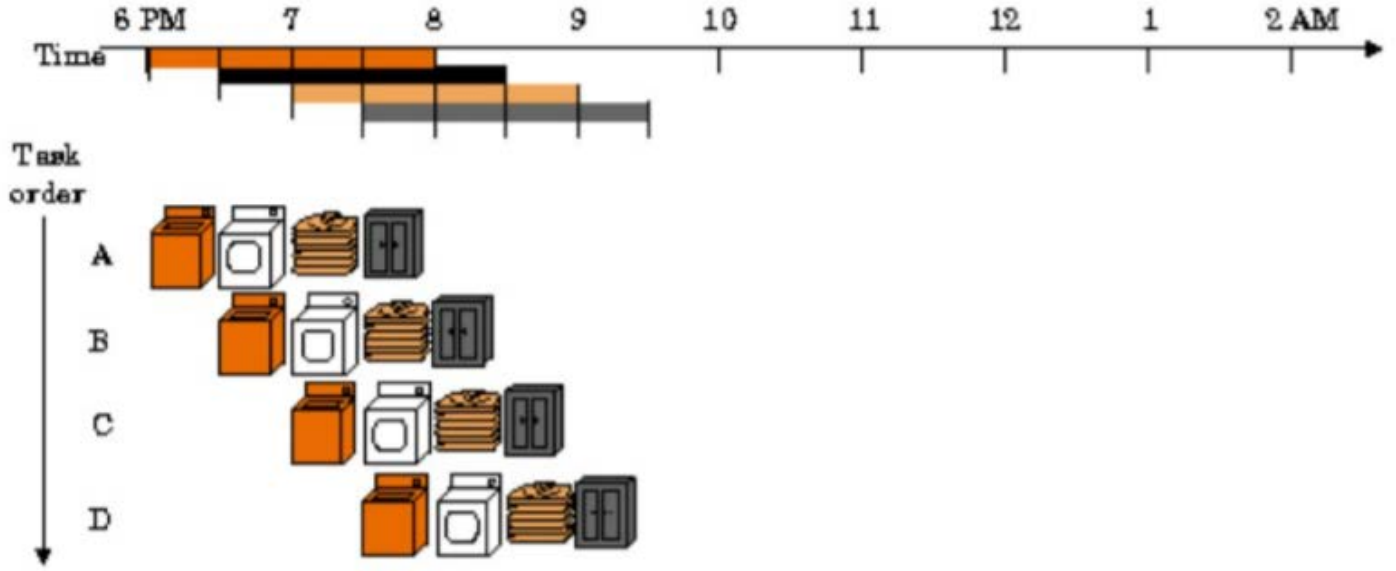


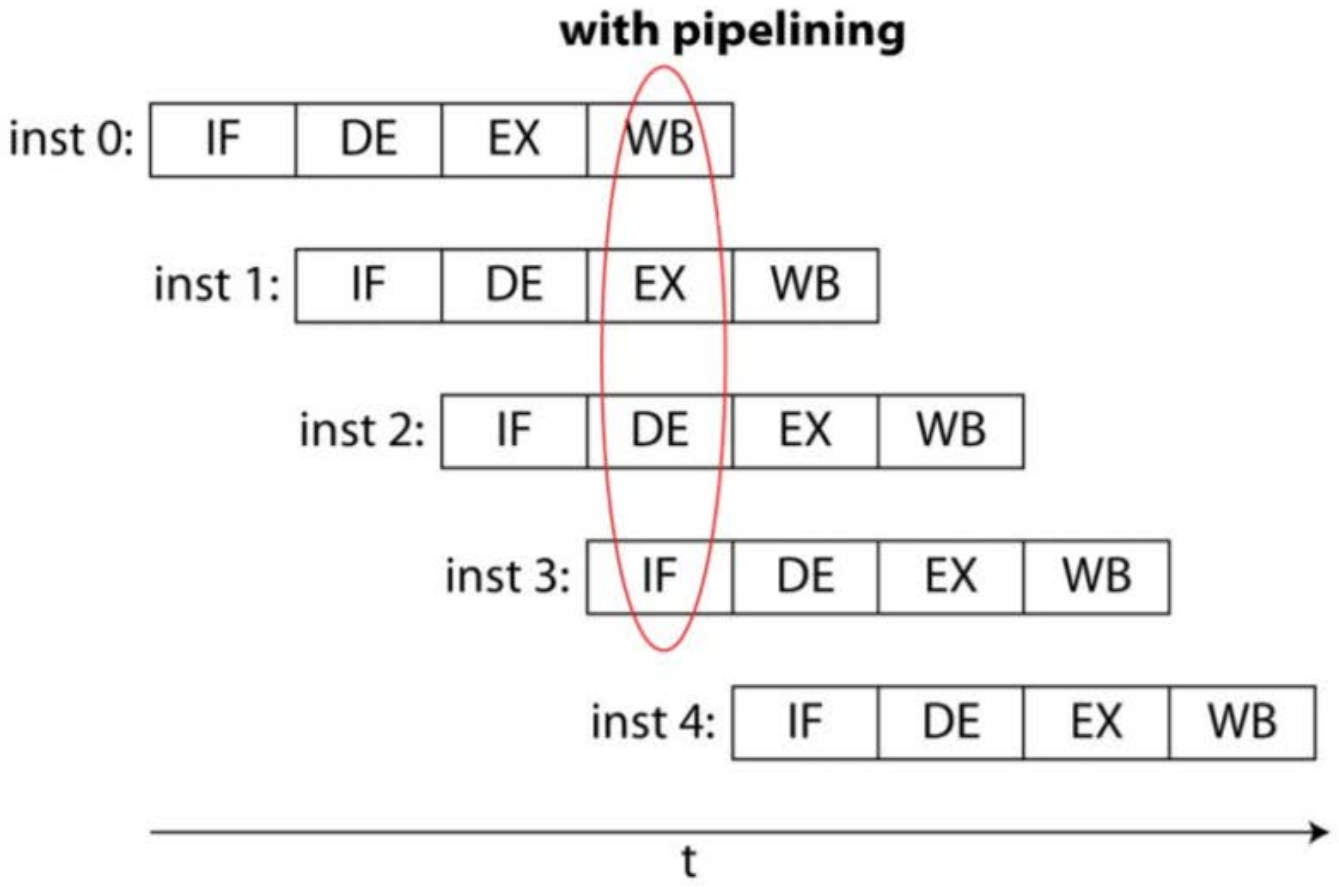
Pipeline

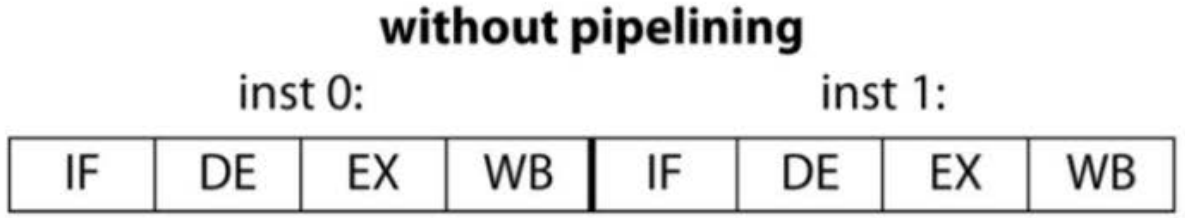
- Pipeline er et samlebåndsprinspipp som skal effektivisere gjennomføringen av instruksjoner
- Dele opp instruksjoner i uavhengige deler
- Krever bestemt rekkefølge på instruksjonene
- Sette like lang tid til hver del

- Hvis vi deler en instruksjon i 4 betyr det ikke at det går 4 ganger fortere, men det vil fortsatt være mye mer effektivt

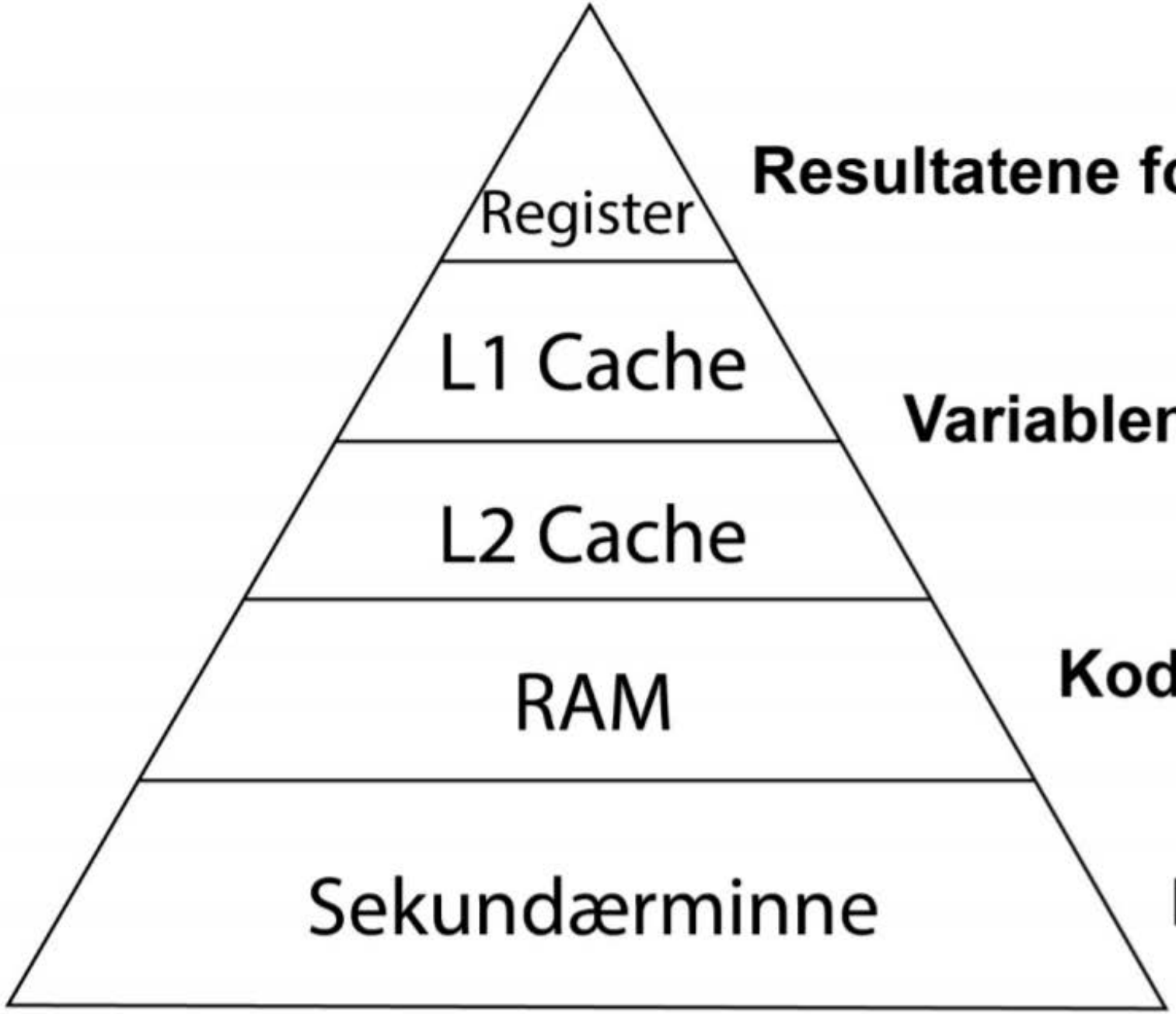

6 PM
7
8
9
10
11
12
1
2 AM
Time
Task order
A
B
C
D


6 PM
7
8
9
10
11
12
1
2 AM
Time
Task order
A
B
C
D


with pipelining
inst 0:
IF
DE
EX
WB
inst 1:
IF
DE
EX
WB
inst 2:
IF
DE
EX
WB
inst 3:
IF
DE
EX
WB
inst 4:
IF
DE
EX
WB
t


without pipelining
inst 0:
inst 1:
IF
DE
EX
WB
IF
DE
EX
WB

Mentimeter



Resultatene for hver iterasjon

Variablene og telleren

Koden til For-løkke

Hele programmet

Typer minne

- Register
  - Internminnet / intern kladdeblokk i CPUen med rask aksess til innholdet (som i LMC)
  - Ca. 100 B
- Cache
  - Liten mengde hurtig mellomlager for både instruksjoner og data
  - Gjør at man ikke må skrive og lese så ofte fra RAM og man unngår flaskehalser
  - Mellomlager internt (L1) eller i nærheten (L2,L3) av CPU-en
    - Typisk kapasitet fra 10 KiloByte (L1) til 1 eller flere MegaByte (L2, L3)
- RAM
  - Buffer mellom eksternt lagringsmedium og CPU-en med rask skrive og leseaksess
  - Internt minne på hovedkortet i nærheten av CPU-en
  - Opptil flere GigaByte med plass
- SSD/Flash/Harddisk
  - Høykapasitetsmedium for program / data
  - Ekstern eller intern lagringsenhet i maskinen med kapasitet opptil flere TeraByte

Cache hit og cache miss

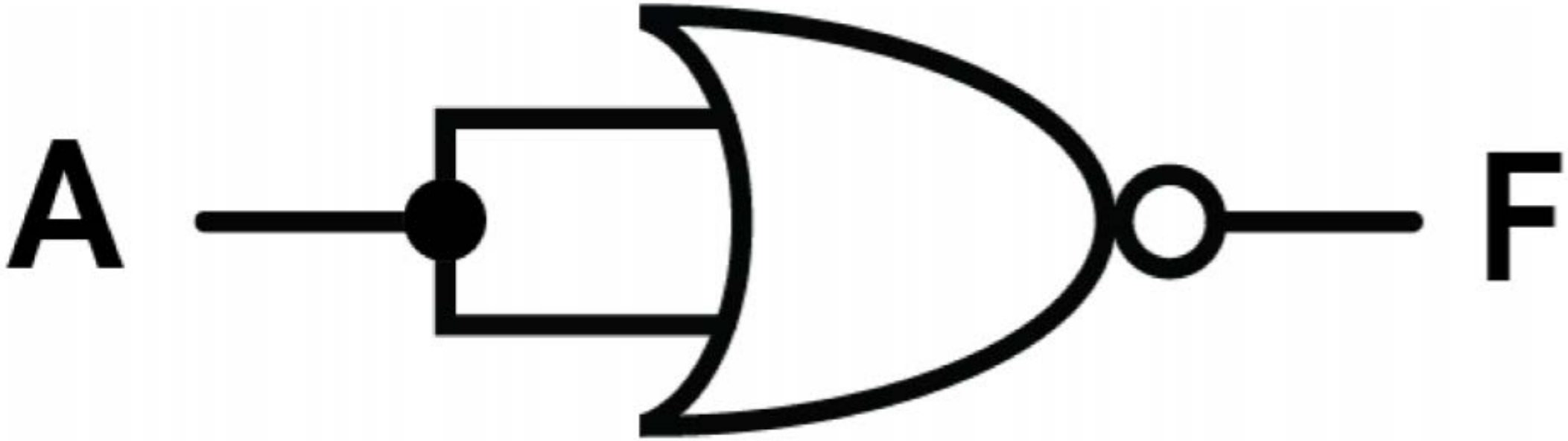
- Cache miss er når et forsøk på å lese eller skrive til cache feiler
  - Man må da hente dataen i annet minne f.eks. RAM
- Cache hit er når man finner dataen i cachen
- De fleste CPU-er har opptil 90% cache hit

- Cache minne er delt i lag eller cache hierarki
  - Nærmeste er level-one cache eller L1 som er direkte på prosessoren
  - Neste er L2 og L3 som ikke er på prosessoren men veldig nært og har mer plass

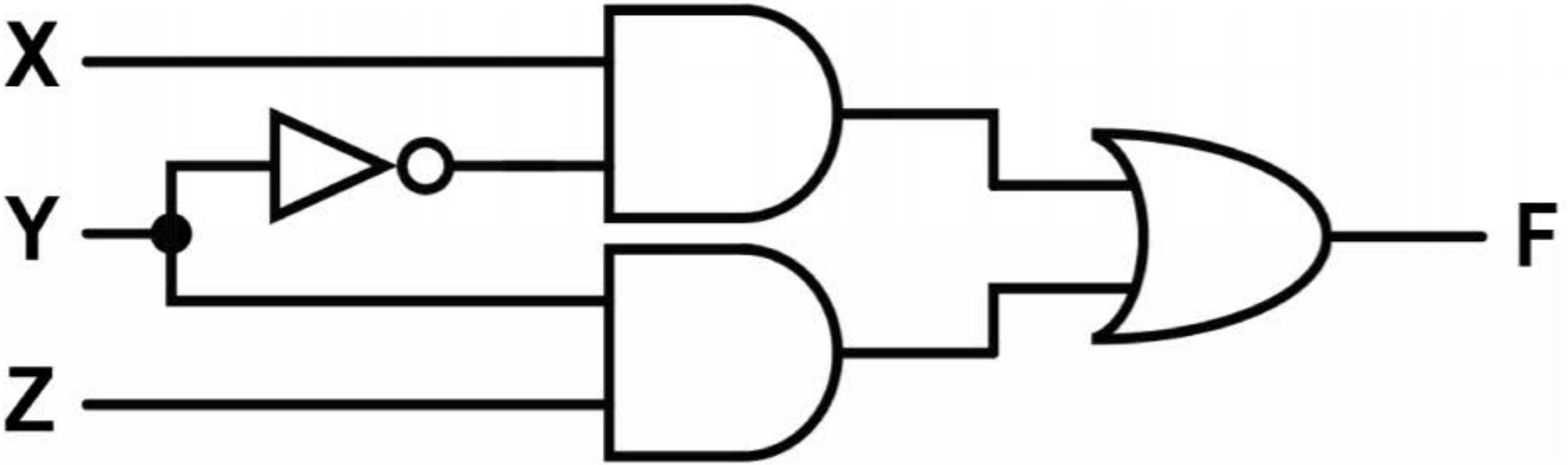
Tidligere eksamensoppgaver om datamaskinarkitektur

Samme oppgaver som når forelesingen ble avlyst men nå på zoom

H2019: 4.1 Portanalyse



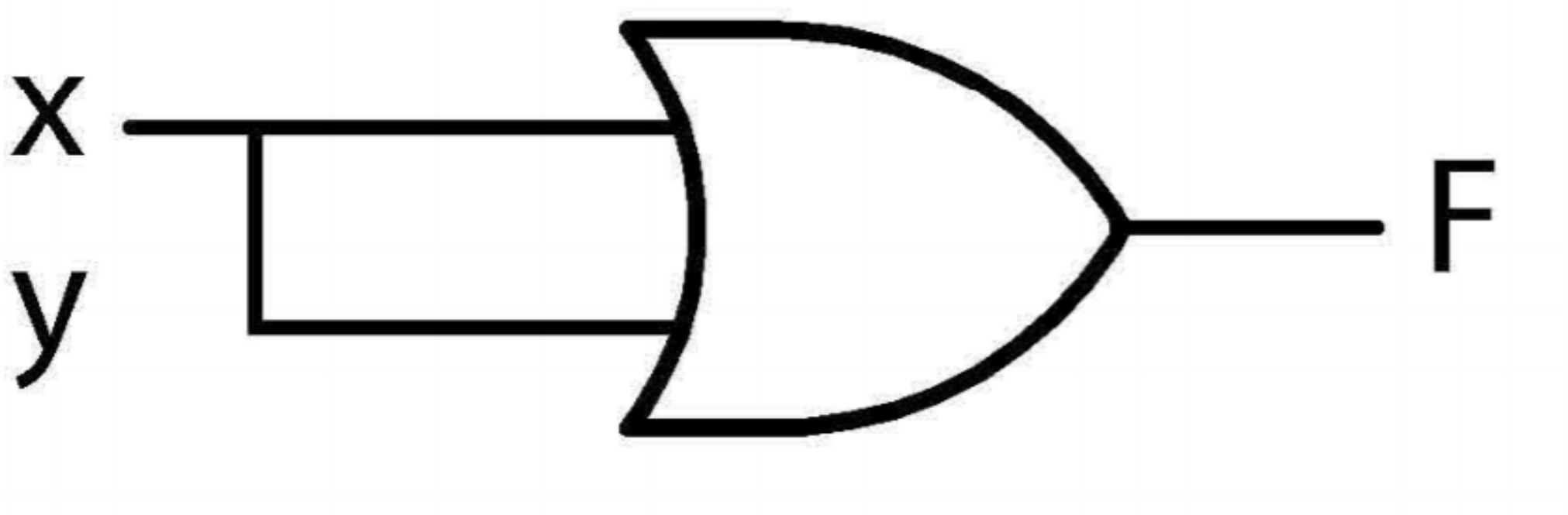
H2019: 4.2: Kretsanalyse



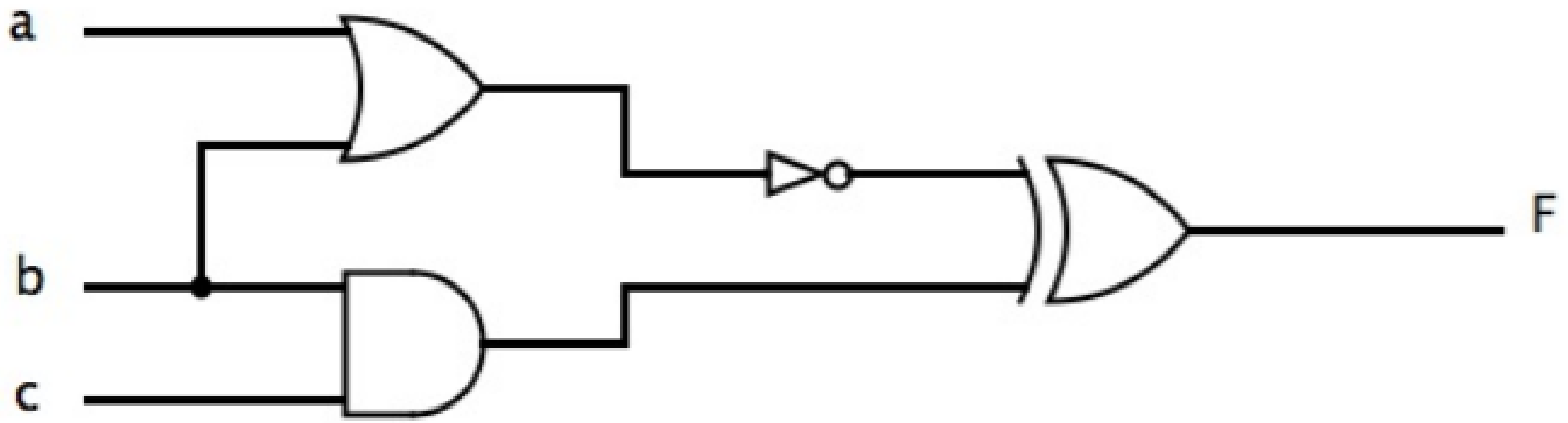
H2019: 4.3: Cache-miss

- Anta at prosessoren har 100 instruksjoner igjen å utføre.
- Det tar én klokkesykel per instruksjon, og man regner med å ha minneaksessering på 80% og med cache-miss på 50%.
- En cache-miss fører til en total tidsbruk på 20 klokkeyskler.
- Cache-hit bruker totalt én klokkesykel.

H2017: 2a Porter



H2017: 2d Kretsanalyse



(utvalgt) Historie om datanett

Veldig kort oppsummert

Overblikk nettverkshistorie

- Telegraf – Morse
- Telefoni
- Internett
  - ARPANET: tjeneste for utveksling av data mellom universiteter i USA
  - Standardisering
- Andre typer datanettverk
  - Interconnected nettverk
  - Trådløse nettverk
- Ekstreme datanettverk
  - NASA Deep Space Network – konstant kommunikasjon med satellitter og rovere sendt til fjerne planeter og ut av solsystemet

Telefon

- Slutten av 1870-tallet i Norge
- Solgt parvis med dedikert linje
- Ble fort veldig komplisert at alle skulle ha egne linjer
- Switch / sentralbord
  - Grunnlag til switcher i internett
  - Linje fra hver telefon til et sentralbord
  - Lange avstander ble komplisert

Telegraf vs Telefoni

- Message switching
  - Telegrammet som en beskjed blir videresend fra sender til mottaker
  - Ingen dedikert linje
- Tilkoblingsløs tjeneste
  - Forskjellige telegrammer fra sender til mottaker kan følge forskjellige stier
- Ligner på Pakkeswitching i dag
  - Pakker kan ta forskjellige stier for å komme til endepunkter selv om de er del av samme beskjed

- Circuit switching
  - Dedikert linje mellom sender og mottaker
  - Reservert «eksklusivt» for hele samtalen / forbindelsen
- Tilkoblingsorientert tjeneste
  - Kommunikasjonen følger alltid samme vei
  - Tilkobling, samtale, nedkobling
- Konseptene bak brukes i dag, men
  - Ingen dedikert linje, kun ressurser reserveres

Standardisering

- Utfordring: Utviklingen av kommunikasjonsprotokoller trenger standardisering
  - Forskjellige lokasjoner, institusjoner, produsenter, operatører
  - Vitenskapelig publikasjoner: For sent
  - Industriell standardisering: For sent og for kostbart

- Løsning: RFC (Request for Comments)
  - Memos, møtereferater sirkulert med post
  - Senere publisert elektronisk: HTTP, FTP etc.

Internetworking (ca. 1972 og utover)

- Foreslått i 1973 med mål om å koble sammen flere forskjellige nettverk

- Grunnreglene fremdeles aktuelle i dag
  - Ingen interne endringer trengs for å koble nett til internett
  - «Best effort» kommunikasjon: tilstandsløs
  - «Stateless roturers» brukt for å koble sammen nettverkene
  - Skulle ikke være kontrollert av ett land

Lagdeling i datakommunikasjon

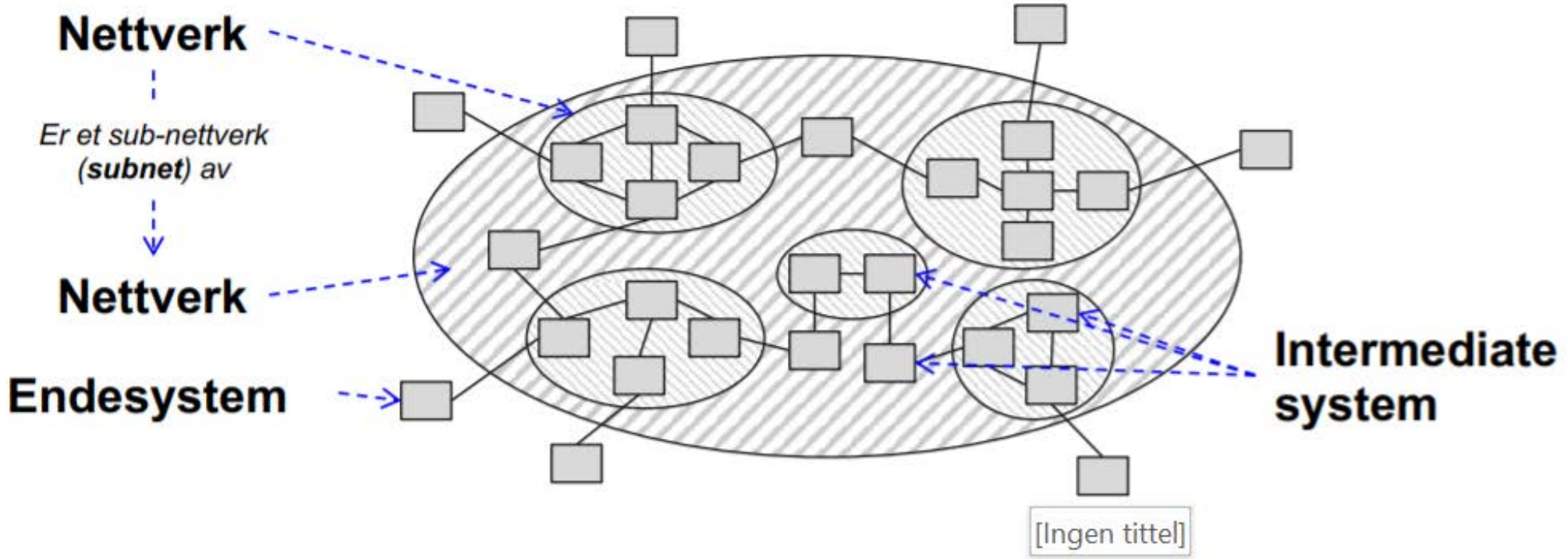
- Komponenter i nettverket
- Nettverksstrukturer
- Aksessmodeller i datanettverk
- Nettverksprotokoller
- Lagdelingskonseptet
- TCP/IP-modellen
- Lagene i TCP/IP-modellen

Rep: Komponenter, aksess modeller

- Tjenere – server
- Klienter

- Switcher – dirigerer signalene i kablet nett
- Routere – oversetter mellom kablet og trådløst nett

Nettverkskomponenter



Endesystemer (tjenere og klienter):

- *For eksempler:*
  - Datamaskiner
  - Mobiltelefoner
  - «Duppedingser» (IoT), skrivere, etc.

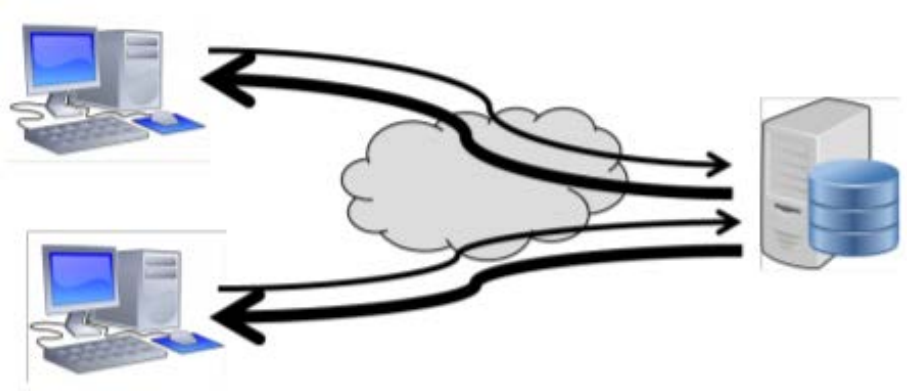
Intermediate system (rutere og switcher):

- *For eksempel:*
  - Ruter
  - Switch

Aksessmodeller

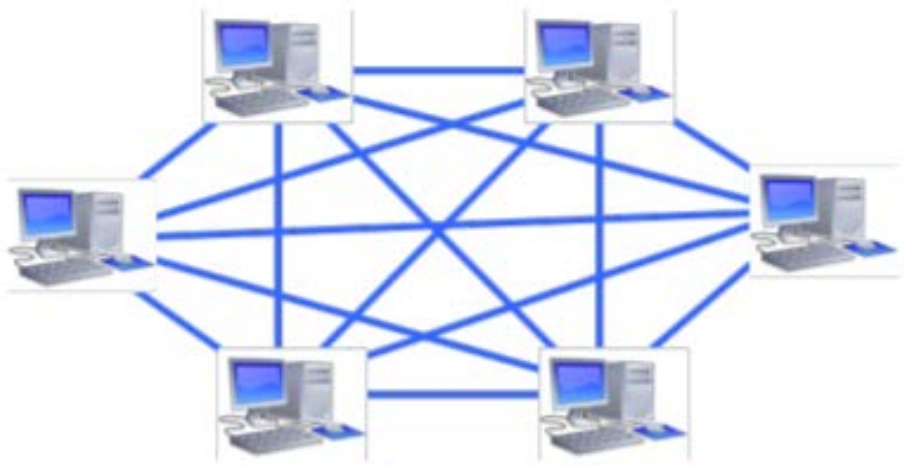
Klient-tjener:

- Klienter ber om en tjeneste
  - Oppretter en forbindelse
- Tjenere leverer tjenesten
  - Svarer på forespørselen



Peer-to-Peer (P2P)

- Alle noder er likeverdige
- Alle noder kan nå hverandre
- Eierskapet er distribuert

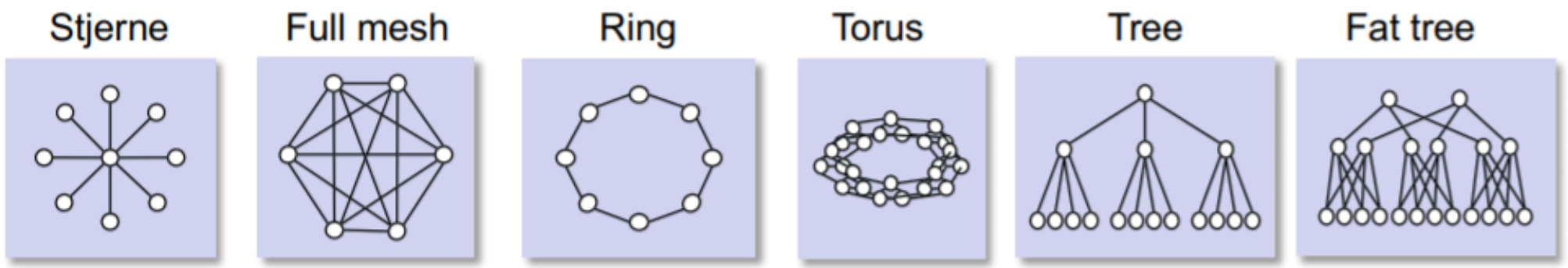


Nettverksstrukturer

Punkt-til-punkt nettverk:

- Flere forskjellige kabler, kabeltyper eller radiolinker som kommuniserer fra punkt til punkt.
- Kabel eller radiolink kobler alltid sammen to noder.
- En-til-en overføring.

Topologieksempler:



Nettverksstrukturer

Broadcast-nettverk:

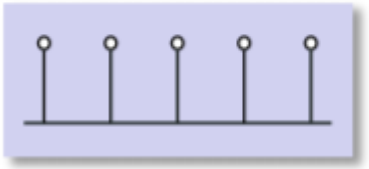
- Nettverk som deler kommunikasjonsmedium.
- En sender, alle lytter (en-til-mange).

Bruk:

- *Trådløs:* Eneste mulighet (mobiltelefoner, satellitter, radio, NFC, ...)
- *Kablet:* Gamle nettverk (Coax, Token ring)

Eksempler

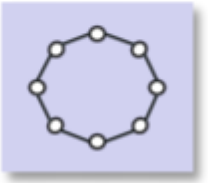
Lagdeling i datakommunikasjon



Bus



Broadcast



Ring

Protokoller og lagdeling

Faktisk pensum for uka

Protokoll

- En protokoll definerer strukturen på beskjeder sendt over et nettverk
- Skal løse problemet med å få felles kommunikasjon mellom forskjellige typer system og/eller nettverk
- Introdusere standardiserte abstraksjonsnivåer
  - Gjøre de modulære og utskiftbare
- Disse er lagdelt for å gjøre dem uavhengige og på forskjellige abstraksjonsnivåer

- Laget vi er på bestemmer abstraksjonsnivå

Hva bestemmer en protokoll?

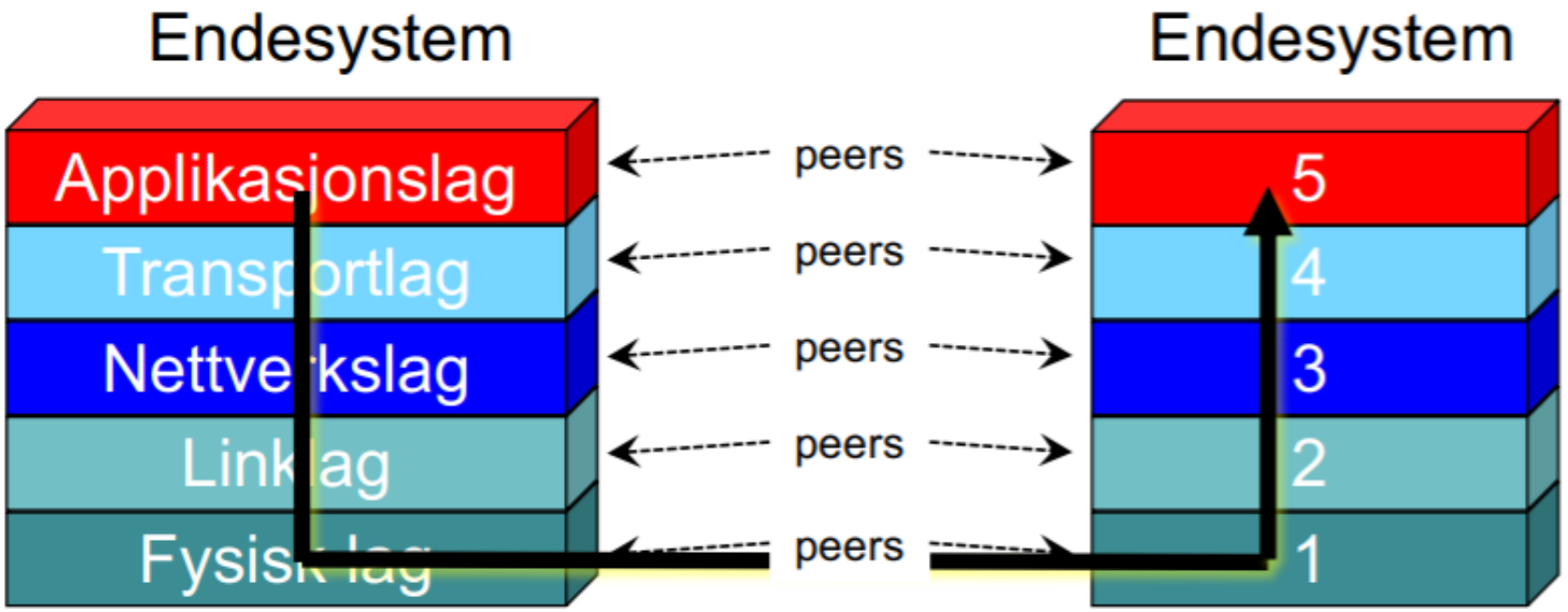
- Formatet på beskjeden og header / konvolutt
- Rekkefølgen på beskjedene
- Utvekslingen av beskjeder mellom to eller flere kommunikasjonssystemer
- Hva skal skje ved mottak eller sending av en beskjed

TCP/IP modellen

- Femlags referansemodell
- Presentasjon, sesjon og applikasjonslagene fra OSI slå sammen til ett lag

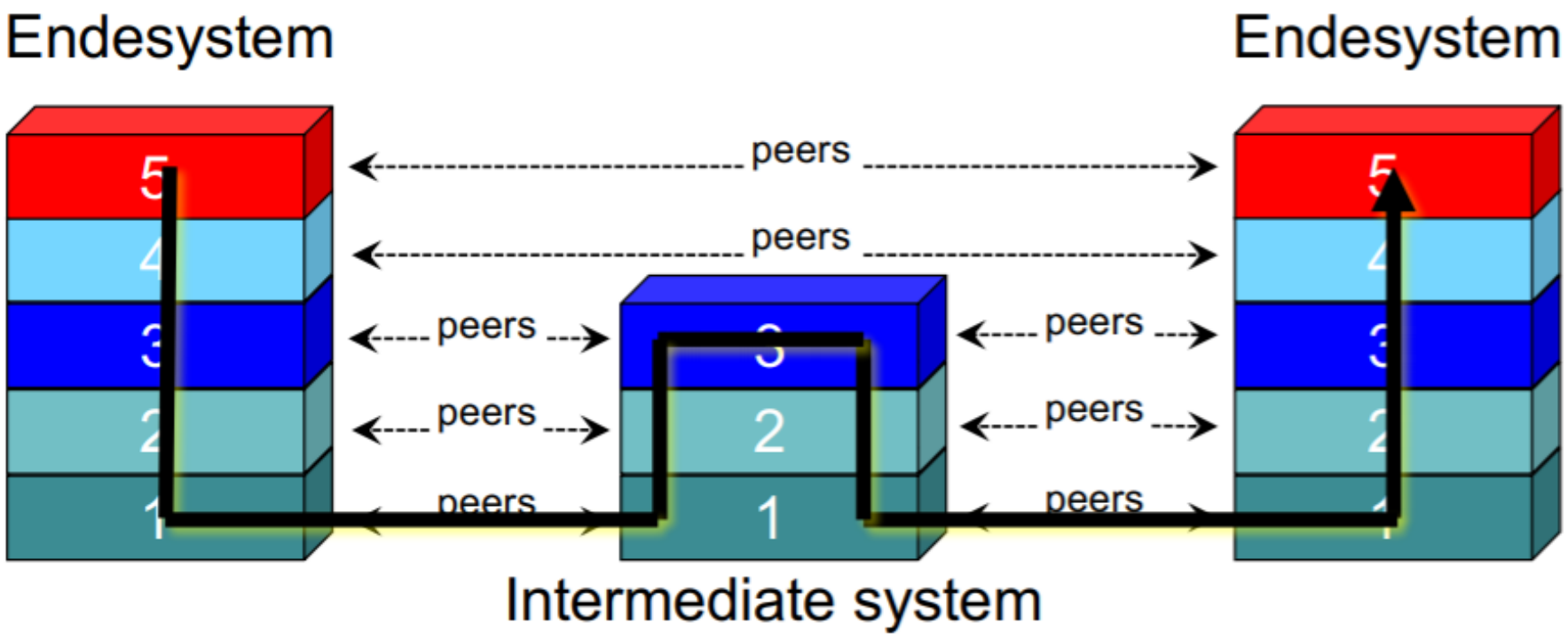
Lag		Funksjon
5	**Applikasjon**	Applikasjonsrelaterte tjenester (**HTTP**, **Mail**)
4	**Transport**	Kobler sammen systemene ende-til-ende (**TCP/UDP**)
3	**Nettverk**	Sende data fra ende-til-ende systemer (**IP**)
2	**Link**	Pålitelig overføring mellom to noder (**LAN/WiFi**)
1	**Fysisk**	Sender bit ut på mediet (kablet eller trådløst)

Datastrøm mellom to systemer med direkte tilkobling



- Forespørselen går nedover lagene på "*avsender*"-siden og oppover hos mottaker.
- Det fysiske laget håndterer selve sendingen til neste (nabo) node.

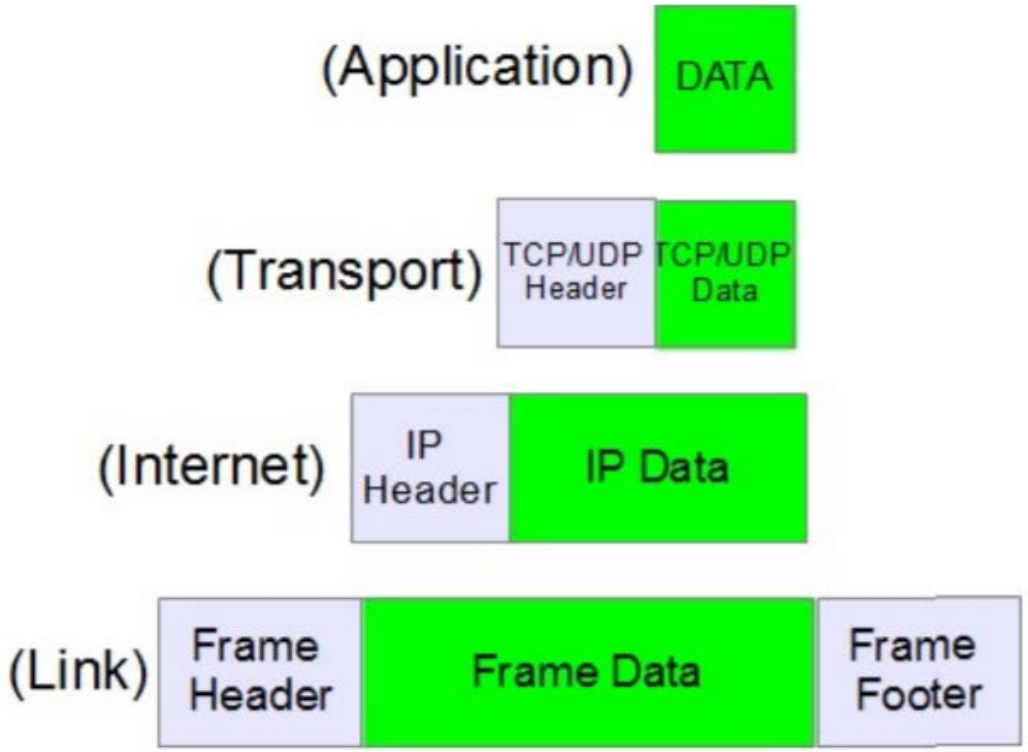
Datastrøm mellom to systemer uten direkte tilkobling



- Nettverksprotokollen (IP) tar hånd om rutingen til endelig destinasjon.
  - På hver ruter så går pakken opp til nettverkslaget for prosessering.
- Transport og applikasjonslaget kommuniserer ende-til-ende.

Lagene i Bruk

- Hva skjer med en melding i de forskjellige lagene når vi prøver å åpne en nettside?
- På Internett blir ofte nettverkslaget og transportlaget blandet
  - Transportprotokoll TPC og nettverksprotokoll IP



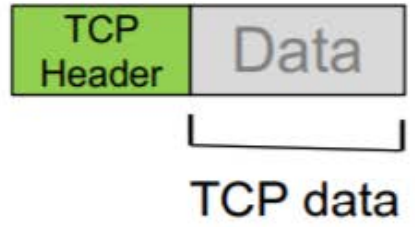
Data

Kan også inneholde en header, men det bestemmer applikasjonen

Lag 5 - Applikasjonslaget

- Lag med tjenester for applikasjoner:
- Eksempler:
  - Nettlesere (WWW)
  - E-post
  - Filoverføring
  - P2P

- *Mer om dette 2. november (Tjenester i Internett)!*



Lag 4 - Transportlaget

- TCP: HTTP, E-post, filoverføring, etc.
- UDP: Strømming av video og lyd

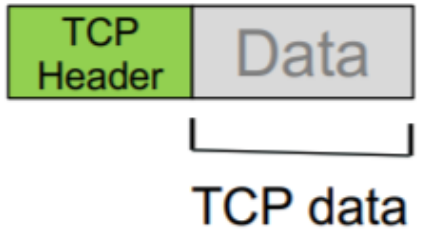
- Bruker «port» som en unik identifikator.
  - Representeres med et 16-bit heltall

TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Res	Flags	Window Size
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

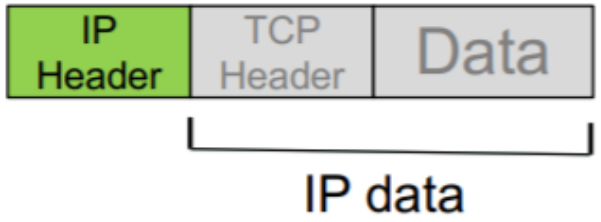
UDP Datagram Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	



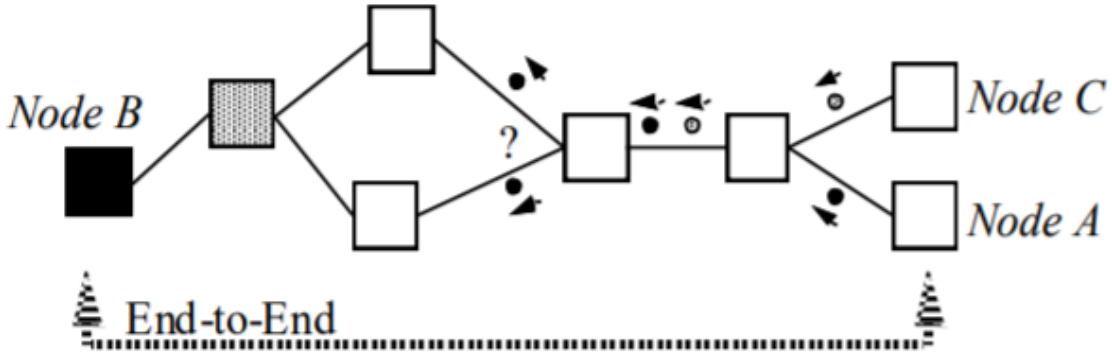
Lag 4 - Transportlaget

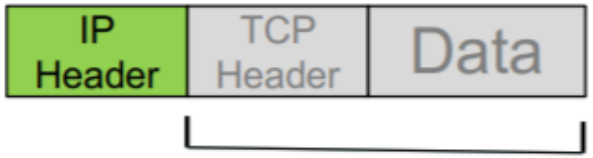
- **TCP:**
  - Oppsett av forbindelse (3-way handshake)
  - Garanterer at pakkene leveres i riktig rekkefølge
  - *Pålitelighet* – Pakker sendes på nytt hvis kvitteringen (ACK) ikke kommer frem
  - Flytkontroll og meningskontroll
- **UDP:**
  - Tilkoblingsløs forbindelse
  - Ingen garantier
  - «Best-effort» levering av data



Lag 3 - Nettverkslaget

- **Kobler sammen systemene ende-til-ende**
- **Ruting**
  - Statisk, definert under tilkobling eller dynamisk
  - Meningskontroll (for mange pakker på en sti)
  - Tjenestekvalitet (QoS)
- En «ruter» jobber på lag 3
- Eksempler:
  - IP (tilkoblingsløst)

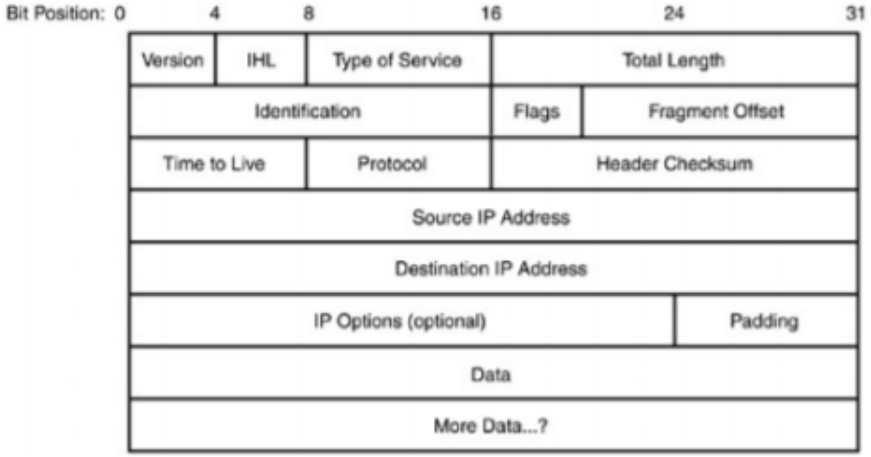


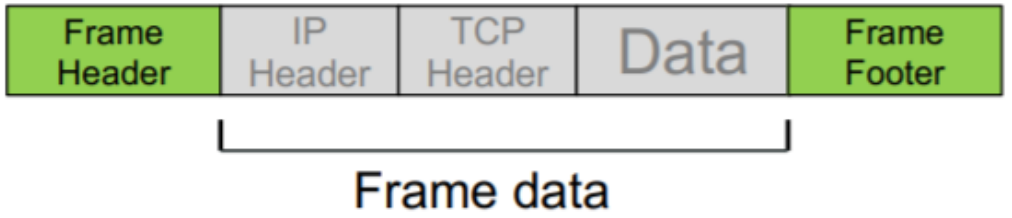


Lag 3 - Nettverkslaget

- Den mest brukte nettverkslagsprotokollen i dag er Internet Protocol (IP). Den mest brukte versjonen er IPv4.

- IPv4 bruker en 32-bit adresse, (4.3×10^9)
- Den nye versjonen, IPv6 har 128-bit adresser (3.4×10^{38})
  - Representeres med fire 8-bit heltall: 192.168.1.101

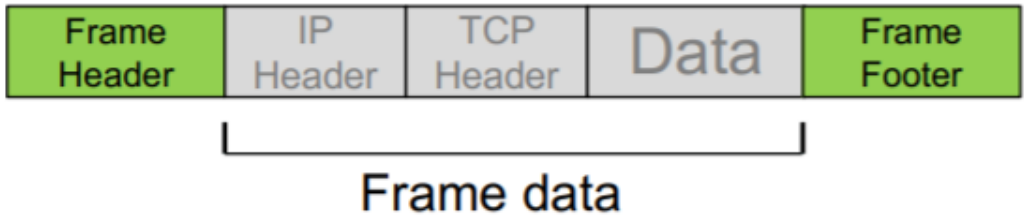




Lag 2 - Linklaget

- **Pålitelig overføring mellom to enheter.**
  - Pakker som overføres i linklaget kalles «frames»
  - Feildeteksjon or retting innenfor en «frame»
- En «switch» vil kun jobbe på lag 2
- Lag 2 vil kunne ha enkel flytkontroll
  - Rask sender, treg mottaker
- Medium Access Control (MAC)

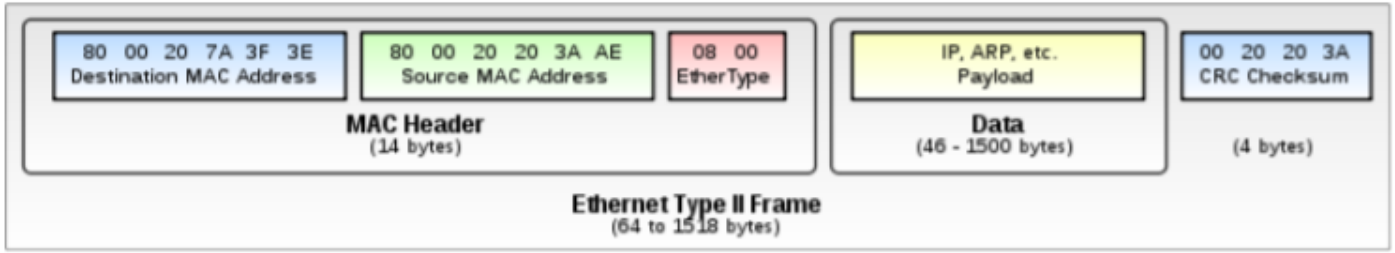




Lag 2 - Linklaget

- Det vanligste linklagene er ”Ethernet”, og ”WiFi”. Disse er ganske like, men har noen forskjeller.

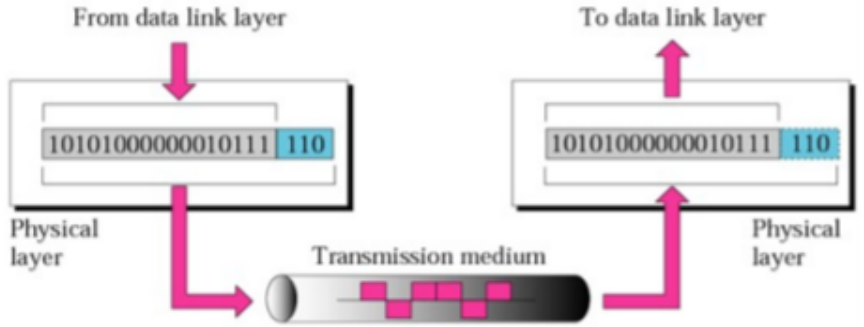
- Bruker en 6-byte adresse (48-bit) som ofte er lagret i nettverkskortet
  - MAC-adresse, brukes både på WiFi og Ethernet.
  - Hver *byte* representeres med en *heksadesimal* verdi: 07:01:02:01:2C:4B



Lag 1 – Det fysiske laget

- **Signalrepresentasjonen av bits:**
- Sørger for at 1-bit også blir mottatt som 1-bit (og ikke et 0-bit)
- Mekanikk: Koblingstype, kabler/medium,..
- Elektronikk: Spenning, bit-lengde,..
- Formelle regler for kommunikasjon:
  - Enveis (unidirectional) – half-duplex
  - Toveis (bidirectional) – full-duplex
  - *Hva skal markere starten og slutten på overføringer*

– Eksempler: RS-232-C, 1000BASE-X



Repetisjon:

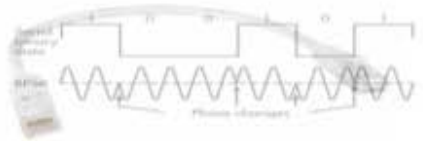
Lagene i Internett (TCP/IP referansemodellen)





Applikasjonslag	**http://www.uio.no**		Applikasjonslag
Transportlag	**192.168.1.5:80**		Transportlag
Nettverkslag	**192.168.1.5**	Nettverkslag	Nettverkslag
Linklag	**A1:B2:C3:D4:E5:F6**	Linklag	Linklag
Fysiske lag		Fysiske lag	Fysiske lag







Ukesoppgaver