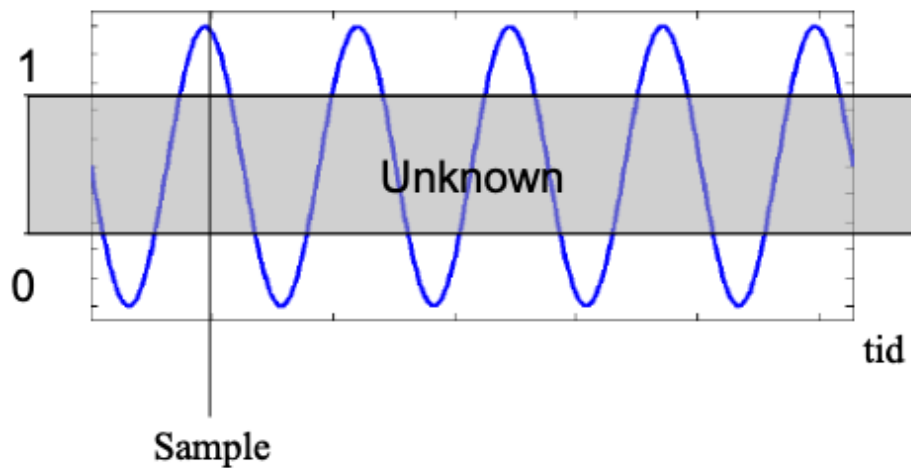


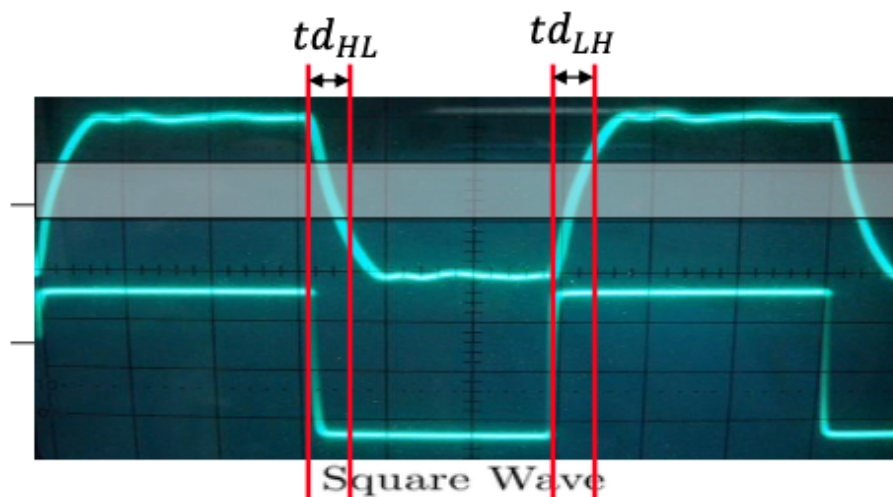
Week 5

Analog/Digital

- Analog: verdi- og tidskontinuerlig
- Digital diskreteverdi og -tid



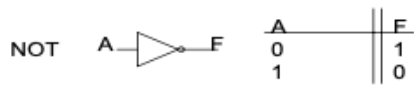
Det finnes tidsforsinkelser i elektriske pulser. Begrenser hvor fort et signal kan skifte verdi.



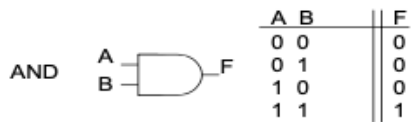
Digitale maskiner (bygget av porter)

Logiske porter:

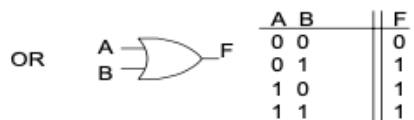
Symbol Sanningstabell Logisk (boolsk) uttrykk



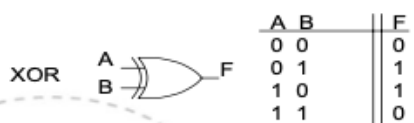
$$F = \overline{A} \quad (F = \text{not } A, F = \sim A)$$



$$F = A * B \quad (F = AB, F = A \text{ and } B)$$



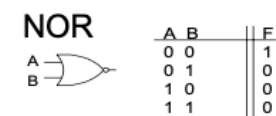
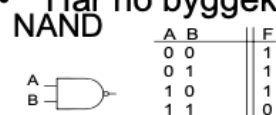
$$F = A + B \quad (F = A \text{ or } B)$$



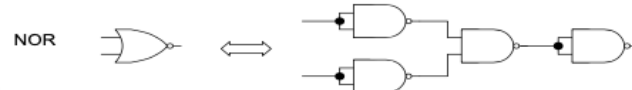
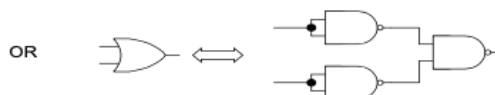
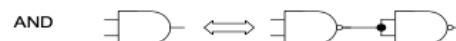
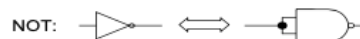
$$F = A \oplus B \quad (F = A \text{ xor } B)$$



- Har no byggeklossar til å lage alle bolskefunksjonar



- Kan no lage alle dei andre (Boolsk algebra) med NAND (eksempel)



Vi kan lage alle andre logiske funksjoner av kun NAND og NOR! Dette ser vi ovenfor

Three state buffer

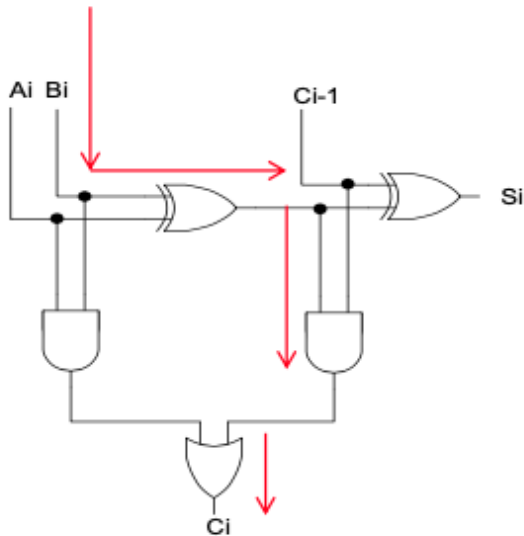
- Buffer som gjør at vi kan koble fra f.eks busslinjer
- Tre pinner:
 - Inn
 - Ut
 - Kontroll
- To tilstander
 - Tilkoblet

- Høy impedans (three state)

Dette er viktig dersom vi deler buss med andre komponenter. Vi kan skru på buffer på komponenter som ikke skal ha tilgang til bussen og skru av buffer på komponenter som skal ha tilgang til bussen

Timing

- Lengste sti definerer tiden

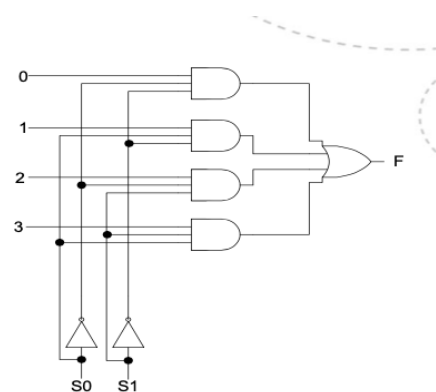
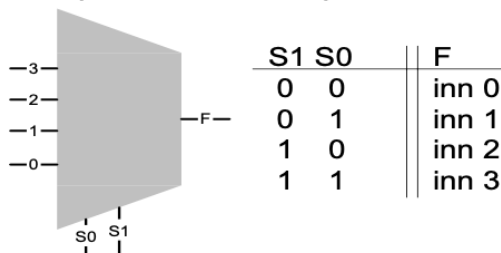


Her har vi en forsinkelse på tre porter. Den lengste stien må være kortere enn en klokkeperiode

Komponenter

Multiplexer

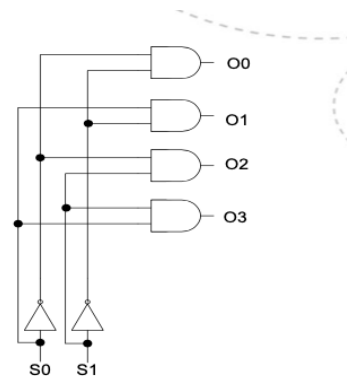
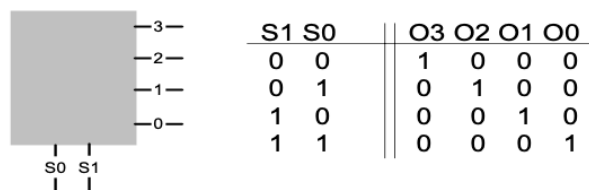
- Eksempel kretsar: Multiplexer



S-verdiene i en multiplexer sier hvilke innganger som kommer ut hos F. Dersom $S_0=0$ og $S_1=1$ så vil inngang 1 være F

Decoder

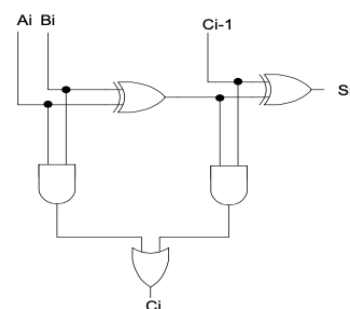
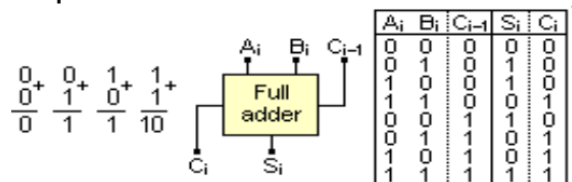
- Eksempel kretsar: Decoder



Ganske likt en multiplexer, men har kun utganger. S-verdiene bestemmer kun hvilken av utgangene som skal være aktive

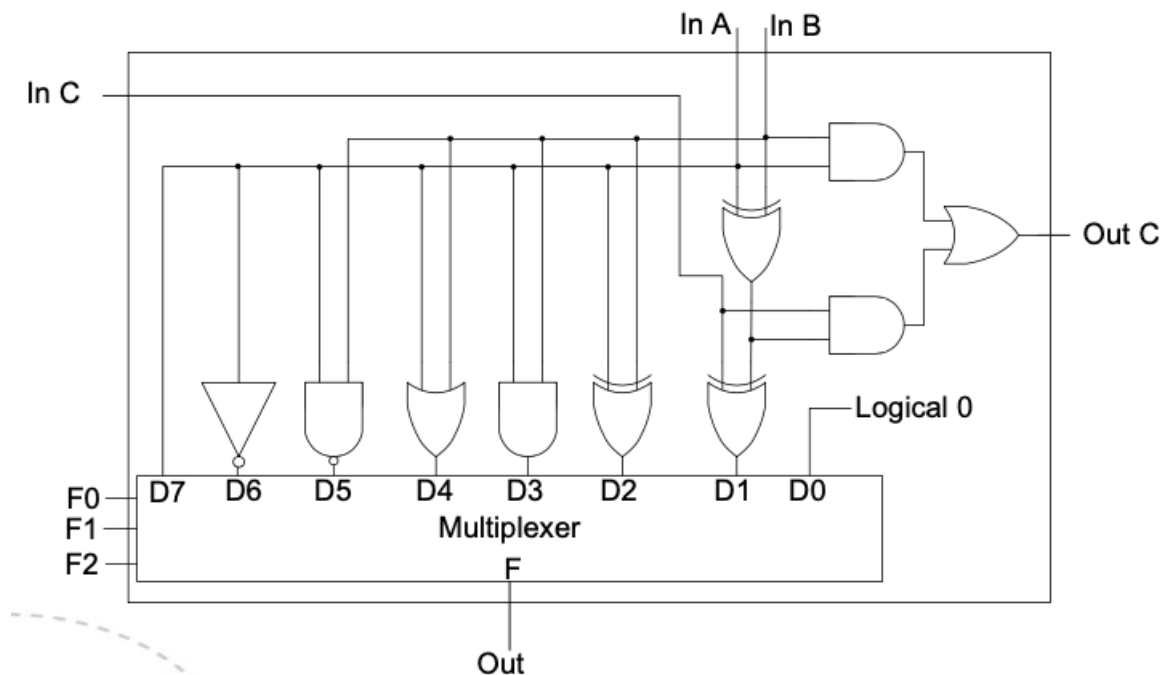
Full adder

- Eksempel kretsar: Full adder



Legger sammen to en-bitt tall. Vi har C som en carrier. Ci er hva som kommer ut. Dersom vi får inn en carrier inn i adderen så får vi det via C(i-1).

Enkel 1-bit ALU

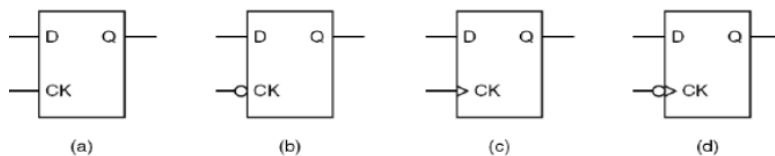
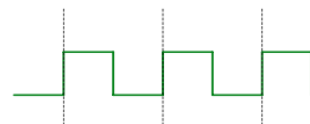


Her kan vi gjøre NOT, NAND, OR AND, NOR og vi har en en-bit adder. Alle kretsene vil gjøre operasjonen sin. Multiplexeren velger hva som skal komme ut av ALUen.

D-vippe (Dlatch) og Flip-flop

Lagrer en bit, basis byggekloss for minne

- a) D-vippe, lagrer D når CK = 1
- b) D-vippe, lagrer D når CK = 0
- c) D-flip-flop, lagrer D på stigende klokkeflanke (0 → 1)
- d) D-flip-flop, lagrer D på synkende klokkeflanke (1 → 0)



I a) vil inngangen bli overført til utgangen når CK er høy.

c) er en flip-flop. D går over til Q kun når CK går fra 0 → 1. De samme gjelder for d) bare fra 1 → 0

Register

Her ser vi hvordan alt fungerer sammen. Vi har forskjellige registre med fire bit. Disse er koblet opp til to multiplexere som hver for seg velger hva som skal inn som A og B inngang i ALU-en. ALU-en er bygget opp av full-adder, AND, NOR... som vist tidligere. Hva ALU-en gjør velges igjen av en multiplexer i ALU-en. Når vi har fått resultatet kan dette skrives i kun et register. Vi kan velge hvilket register vi ønsker å skrive til ved hjelp av en decoder.