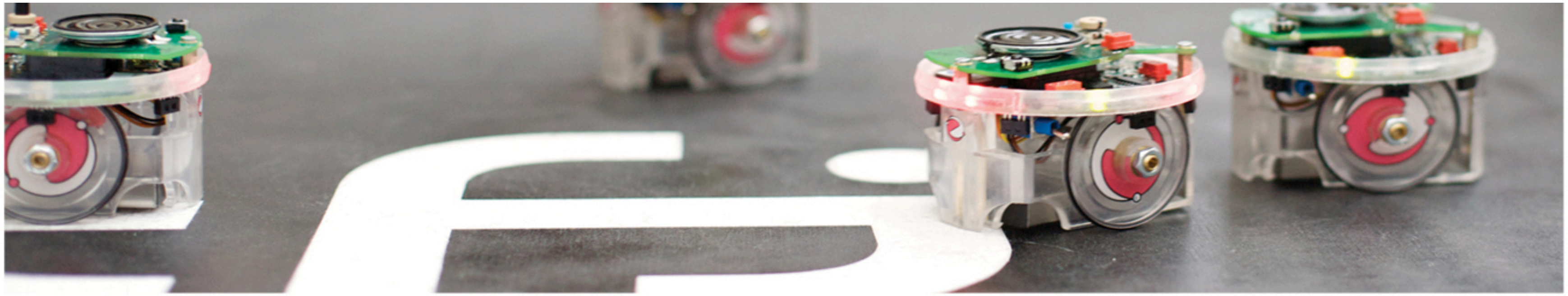


UiO : Institutt for informatikk
Det matematisk-naturvitenskapelige fakultet

INF1400

Kombinatorisk Logikk

Oversikt

- Binær addisjon
- Negative binære tall - 2'er komplement
- Binær subtraksjon
- Binær adder
  - Halvadder
  - Fulladder
  - Flerbitsadder
  - Carry propagation / carry lookahead
- Generell analyseprosedyre

Binær addisjon

Prosedyren for binær addisjon er identisk med prosedyren for desimal addisjon

Eksempel

Adder 5 og 13:

$$\begin{array}{rccccc} & 1 & 1 & & 1 & \\ & 0 & 0 & 1 & 0 & 1 \\ + & 0 & 1 & 1 & 0 & 1 \\ \hline = & 1 & 0 & 0 & \mathbf{1} & 0 \end{array} \qquad \begin{array}{rcc} & 0 & 5 \\ + & 1 & 3 \\ \hline = & 1 & 8 \end{array}$$

Negative binære tall

Mest vanlig representasjon: 2'er komplement

Lar mest signifikante bit være 1 for negative tall

Dette må være "avtalt" på forhånd

Eksempel:

4 bit kan representere tallene –8 til +7

7	0 1 1 1
6	0 1 1 0
5	0 1 0 1
4	0 1 0 0
3	0 0 1 1
2	0 0 1 0
1	0 0 0 1
0	0 0 0 0
-1	1 1 1 1
-2	1 1 1 0
-3	1 1 0 1
-4	1 1 0 0
-5	1 0 1 1
-6	1 0 1 0
-7	1 0 0 1
-8	1 0 0 0

2’er komplement

Setter minus foran et binært tall ved å invertere alle bittene og plusse på 1

Eksempel:

Finner -5:

invertert 5:		1 0 1 0
	+	0 0 0 1
-5:	=	1 0 1 1

7	0 1 1 1
6	0 1 1 0
5	0 1 0 1
4	0 1 0 0
3	0 0 1 1
2	0 0 1 0
1	0 0 0 1
0	0 0 0 0
-1	1 1 1 1
-2	1 1 1 0
-3	1 1 0 1
-4	1 1 0 0
-5	1 0 1 1
-6	1 0 1 0
-7	1 0 0 1
-8	1 0 0 0

Binær subtraksjon

Fremgangsmåte for tall representert ved 2'er komplement:

Adder tallene på vanlig måte.

Eksempel:

$$
\begin{array}{rrr}
 & & \scriptstyle 1\ 1 \\
 & & 0\ 1\ 1\ 0 \\
+ & & 1\ 1\ 1\ 0 \\
\hline
= & (1) & 0\ 1\ 0\ 0
\end{array}
\qquad\qquad
\begin{array}{rr}
 & 6 \\
+ & -2 \\
\hline
= & 4
\end{array}
$$

Går ut

Betyr positivt tall

Binær subtraksjon

Eksempel:

$$
\begin{array}{lr}
 & 0\ 0\ 1\ 1 \\
+ & 1\ 0\ 0\ 0 \\
\hline
= & 1\ 0\ 1\ 1
\end{array}
\qquad\qquad
\begin{array}{lr}
 & 3 \\
+ & -8 \\
\hline
= & -5
\end{array}
$$

Betyr negativt tall

Binær adder

En av de mest brukte digitale kretser

Vanlige anvendelser:

Mikroprosessor ALU / Xbox / mikserbord / digitalt kommunikasjonsutstyr / AD-DA omformere osv...

- Basis for addisjon / subtraksjon / multiplikasjon / divisjon og mange andre matematiske operasjoner
- All form for filtrering / signalbehandling

Binær adder

Ønsker å designe en generell binær adder

Funksjonelt eksempel

Adder to tall A=5 og B=13:

A_1 A_0 B_1 B_0

		1	1	0	1	
A		0	0	1	0	1
B	+	0	1	1	0	1
S	=	1	0	0	1	0

C_2 C_1 S_0 S_1

Halvadder (ingen mente inn)

Adderer sammen de to minst signifikante bittene A_0 og B_0.

Elementet har 2 innganger og 2 utganger

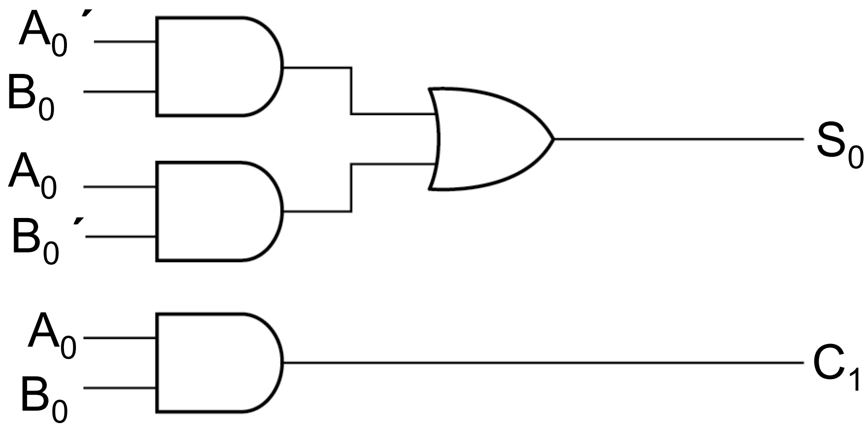
Sannhetstabell

A_0	B_0	S_0	C_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

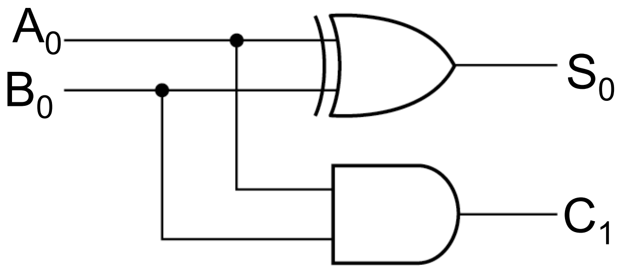
$$S_0 = A_0'B_0 + A_0B_0' = A_0 \oplus B_0$$

$$C_1 = A_0B_0$$

Halvadder implementasjon



$$S_0 = A_0'B_0 + A_0B_0'$$
$$C_1 = A_0B_0$$



$$S_0 = A_0 \oplus B_0$$
$$C_1 = A_0B_0$$

Fulladder (mente inn)

Adderer sammen bit A_n , B_n med evt. mente inn

Elementet har 3 innganger og 2 utganger

Sannhetstabell

A_n	B_n	C_n	S_n	C_{n+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S_n = A_n \oplus B_n \oplus C_n$ (oddefunksjon)

$C_{n+1} = A_n{'}B_nC_n + A_nB_n{'}C_n + A_nB_nC_n{'} + A_nB_nC_n$

Forenkling

Forenkler C_{n+1} ved Karnaughdiagram

A_n \ B_nC_n	00	01	11	10
0			1	
1		1	1	1

$$C_{n+1} = A_n'B_nC_n + A_nB_n'C_n + A_nB_nC_n' + A_nB_nC_n$$

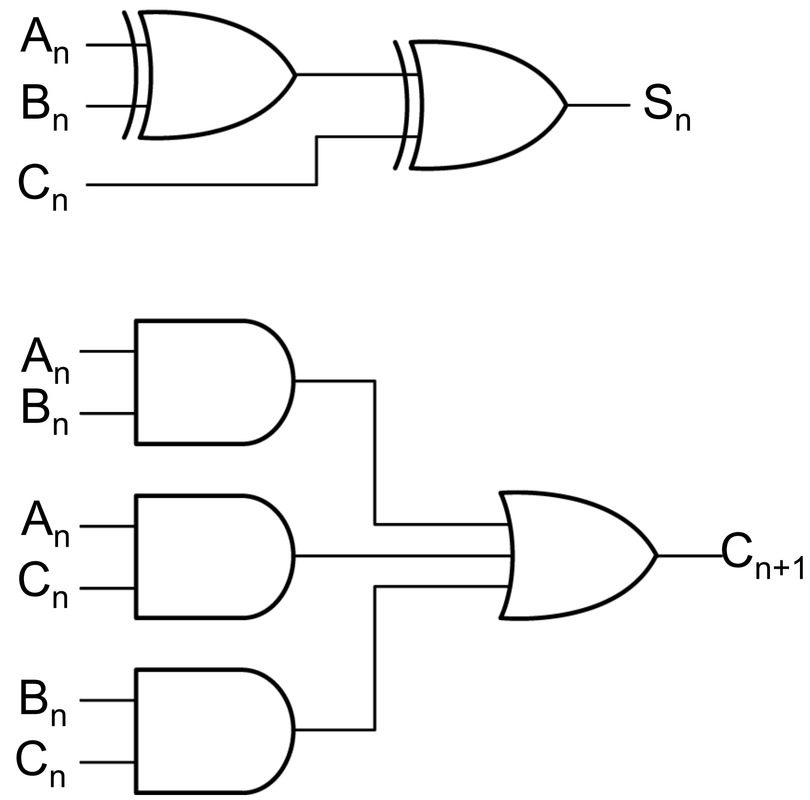
$$C_{n+1} = A_nB_n + A_nC_n + B_nC_n$$

Implementasjon I

Rett fram implementasjon

$$S_n = A_n \oplus B_n \oplus C_n$$

$$C_{n+1} = A_nB_n + A_nC_n + B_nC_n$$



Implementasjon II

Forenklet implementasjon av C_{n+1} basert på gjenbruk av porter fra S_n

$$S_n = (A_n \oplus B_n) \oplus C_n$$

Leser ut C_{n+1} fra karnaughdiagram på nytt

A_n \ B_nC_n	00	01	11	10
0			1	
1		1	1	1

$$C_{n+1} = A_nB_n + A_nB_n{'}C_n + A_n{'}B_nC_n$$

$$C_{n+1} = A_nB_n + (A_nB_n{'} + A_n{'}B_n)\, C_n$$

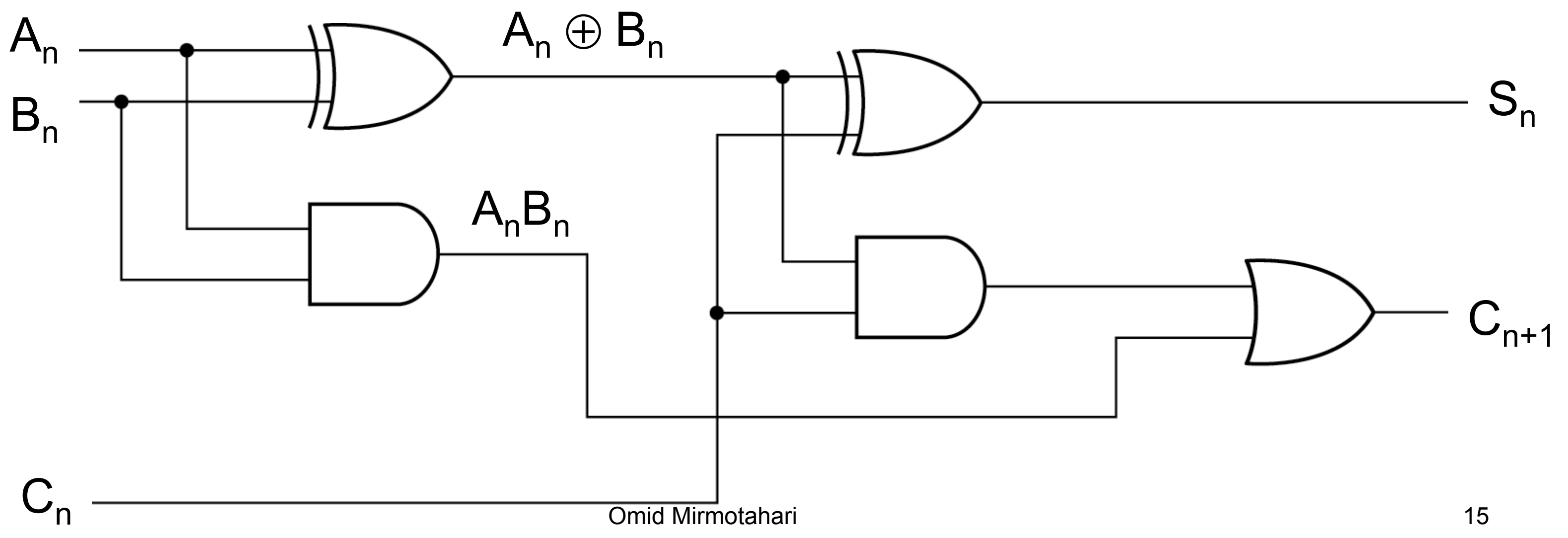
$$C_{n+1} = A_nB_n + (A_n \oplus B_n)\, C_n$$

Implementasjon II

Vanlig implementasjon av en-bits fulladder

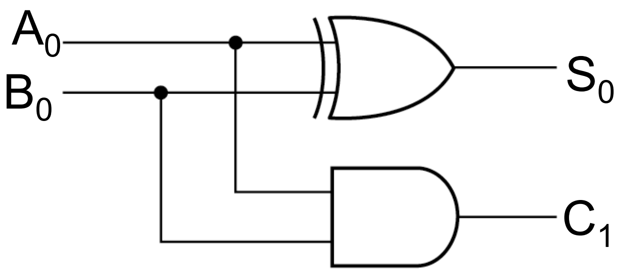
$$S_n = (A_n \oplus B_n) \oplus C_n$$

$$C_{n+1} = A_nB_n + (A_n \oplus B_n)\, C_n$$

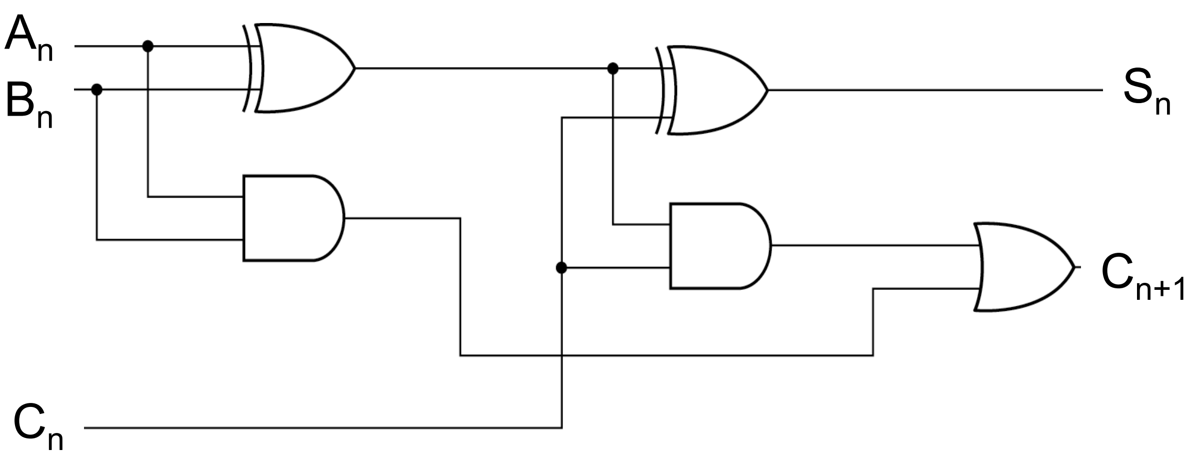


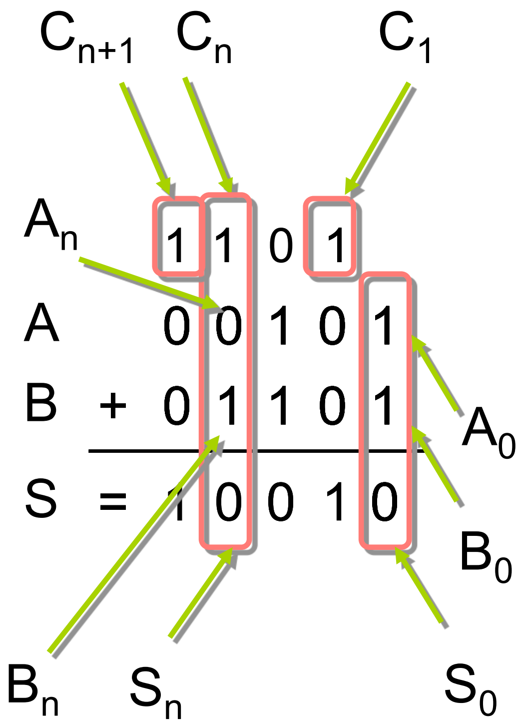
Binær adder

Halvadder (ikke mente inn)



Fulladder (evt. mente inn)



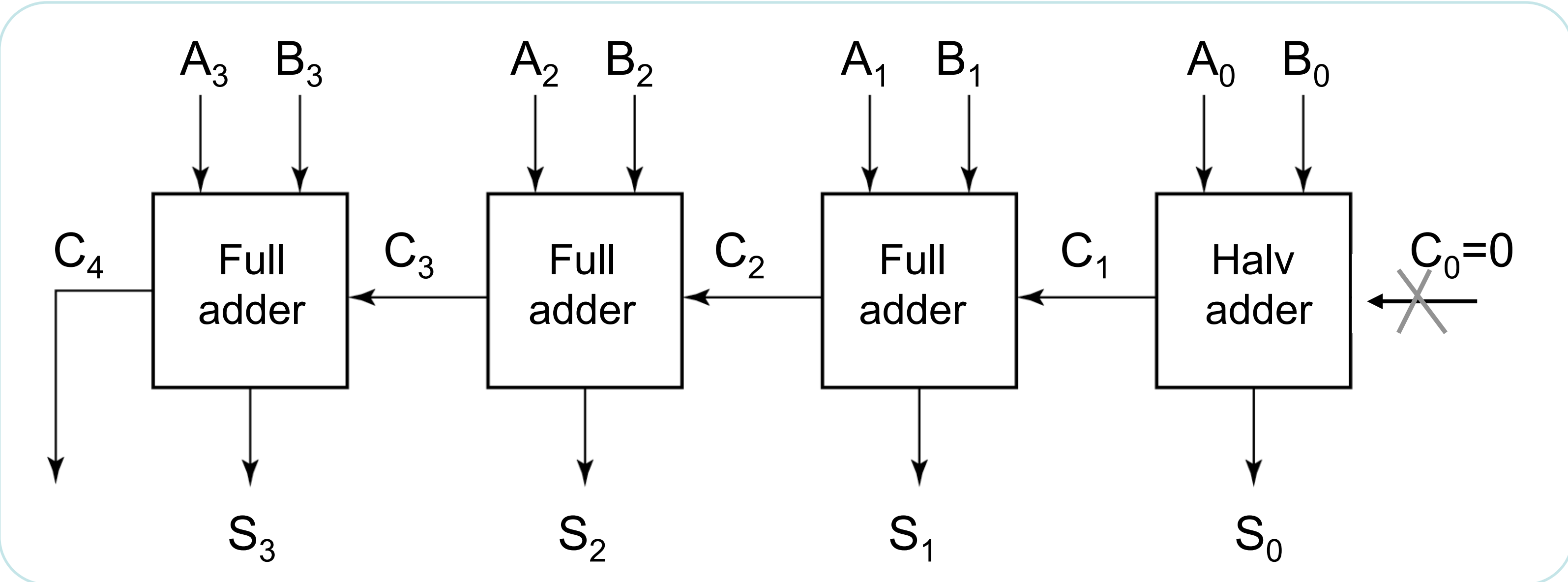


Et adder system

Systemelementer:

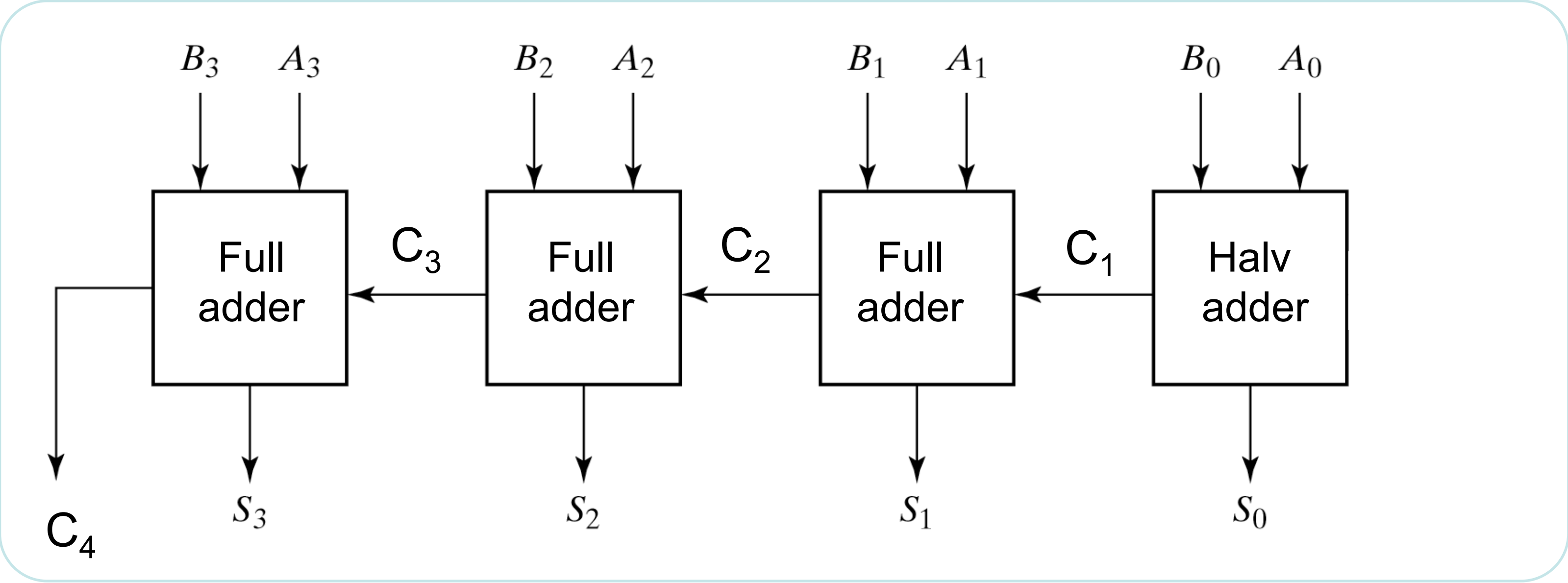
Halvadder: Tar ikke mente inn

Fulladder: Tar mente inn



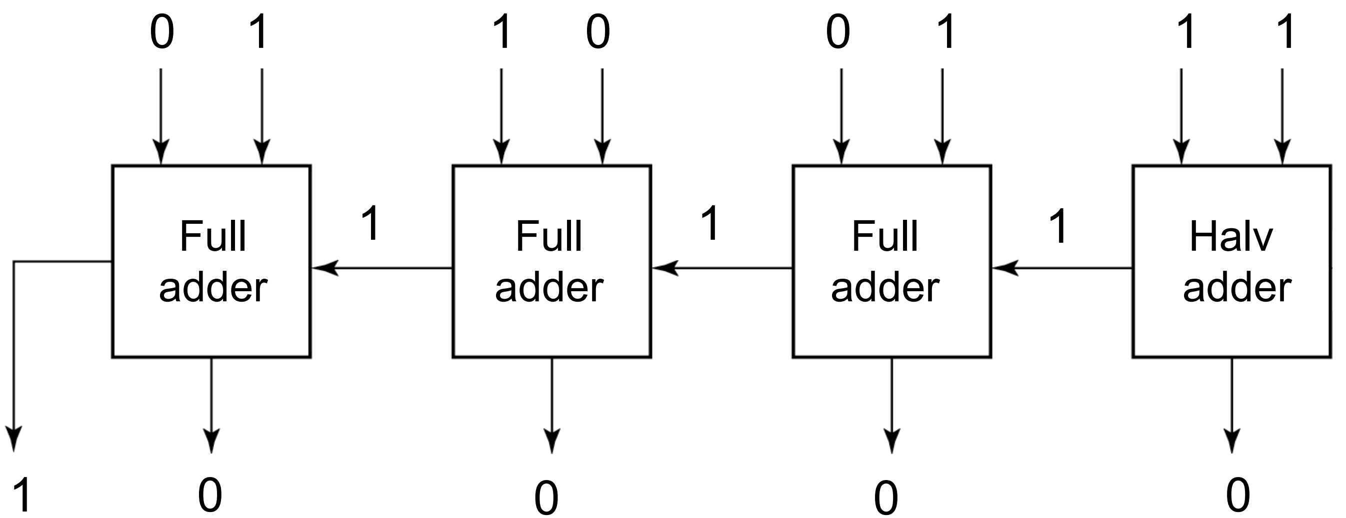
Menteforplantning

4-bits binær adder



Menteforplantning

- Portforsinkelse gir menteforplantning (rippeladder)
  - Eksempel
    - Adderer 0101 og 1011

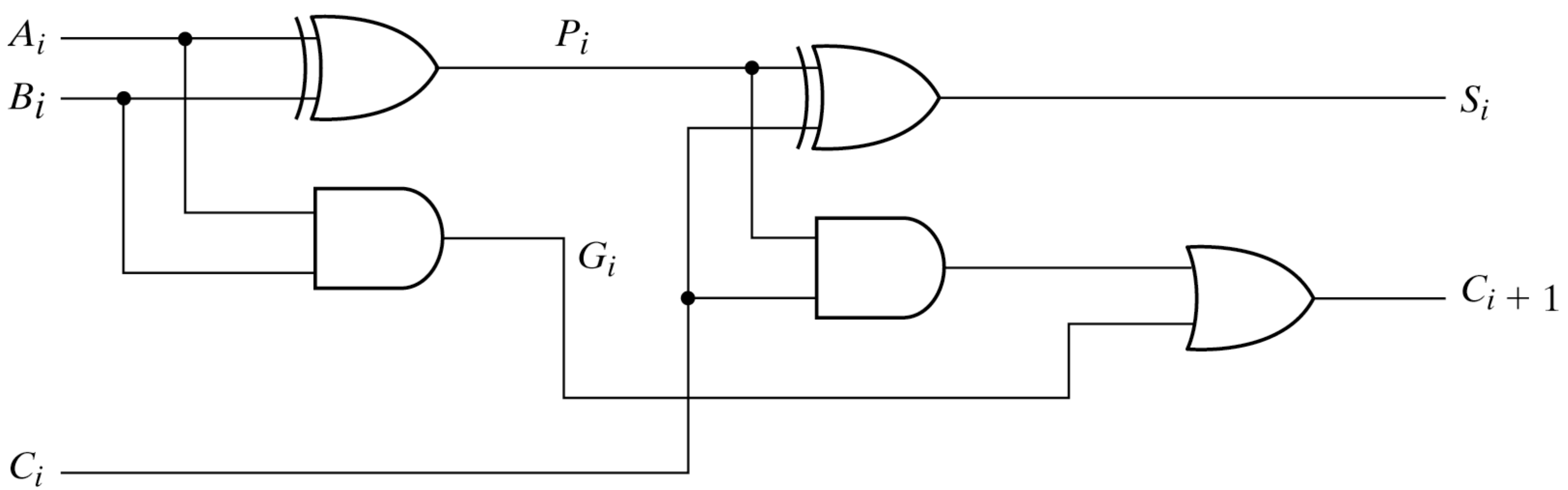


Subtraksjon ?

"Carry Lookahead"

Ønsker å unngå menteforplantning – gir økt hastighet

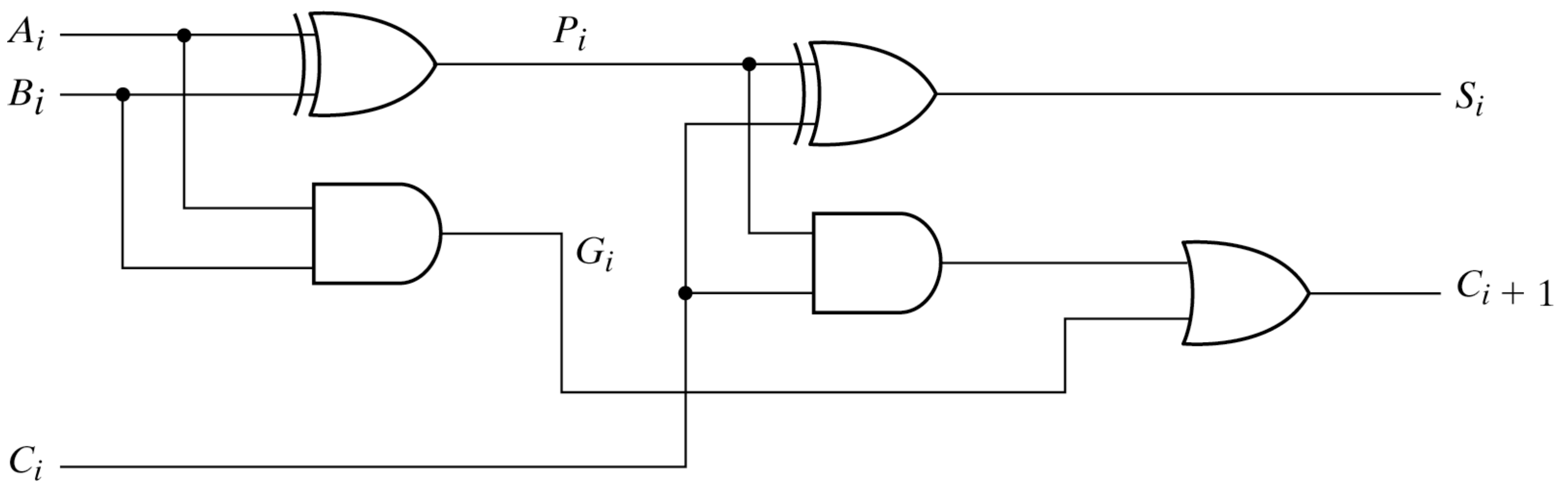
- G_i – generate: brukes i menteforplantningen
- P_i – propagate: påvirker menteforplantningen



"Carry Lookahead"

$S_i = P_i \oplus C_i$

$C_{i+1} = G_i + P_iC_i$



"Carry Lookahead"

For en 4-bits adder bestående av 4 fulladdertrinn har vi:

$S_i = P_i \oplus C_i$ $\quad\quad C_{i+1} = G_i + P_iC_i$

Uttrykker C_1, C_2 og C_3 rekursivt

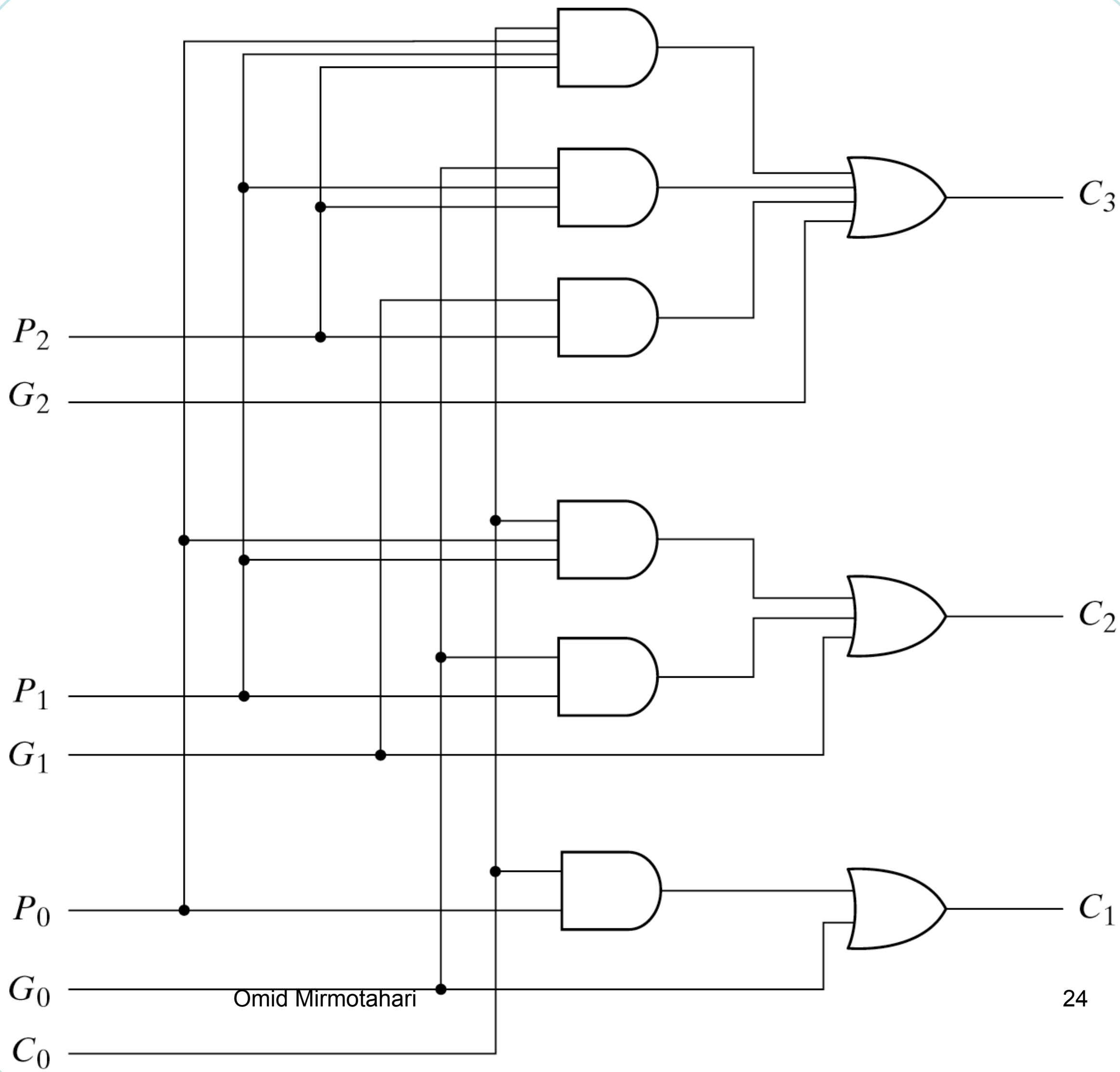
$$C_1 = G_0 + P_0C_0$$

$$C_2 = G_1 + P_1C_1 = G_1 + P_1(G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0$$

$$C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$$

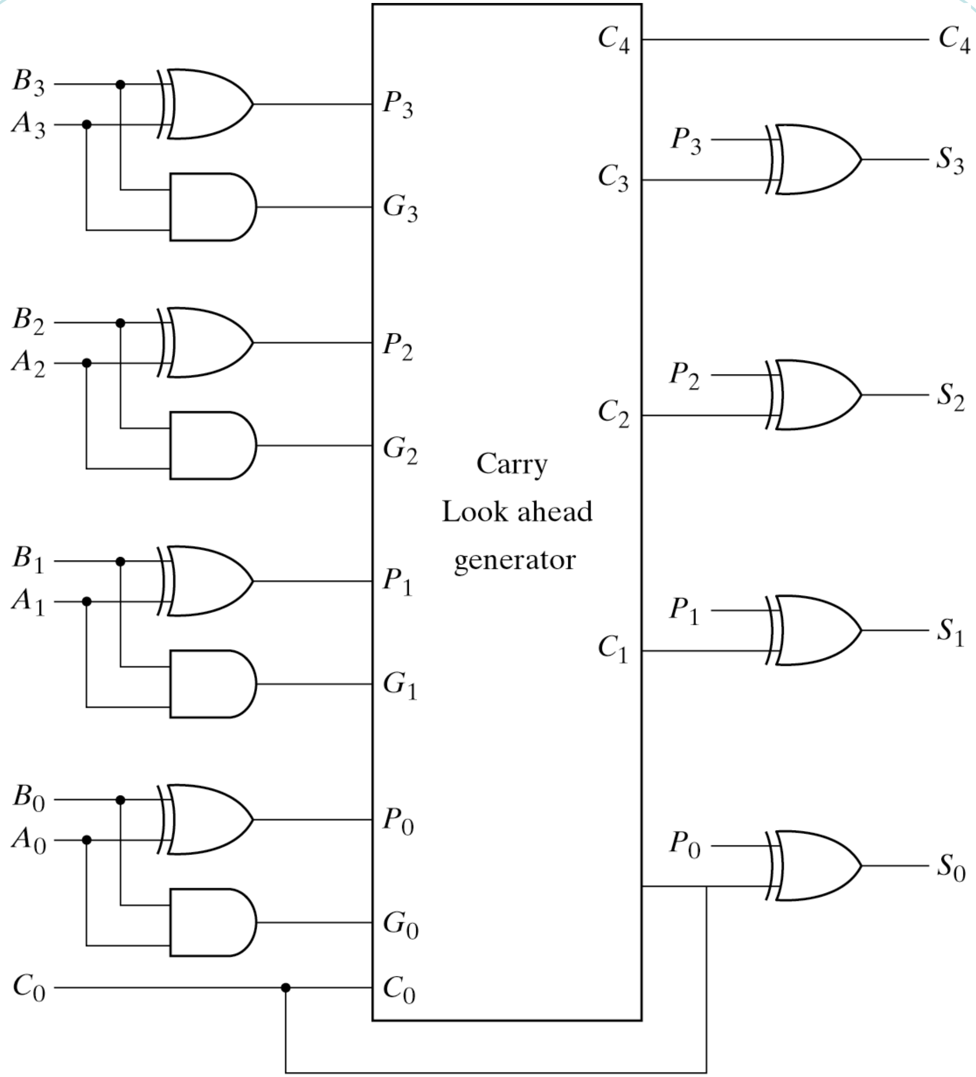
"Carry Lookahead" generator

Rett fram implementasjon av C_1, C_2, C_3



"Carry Lookahead" adder

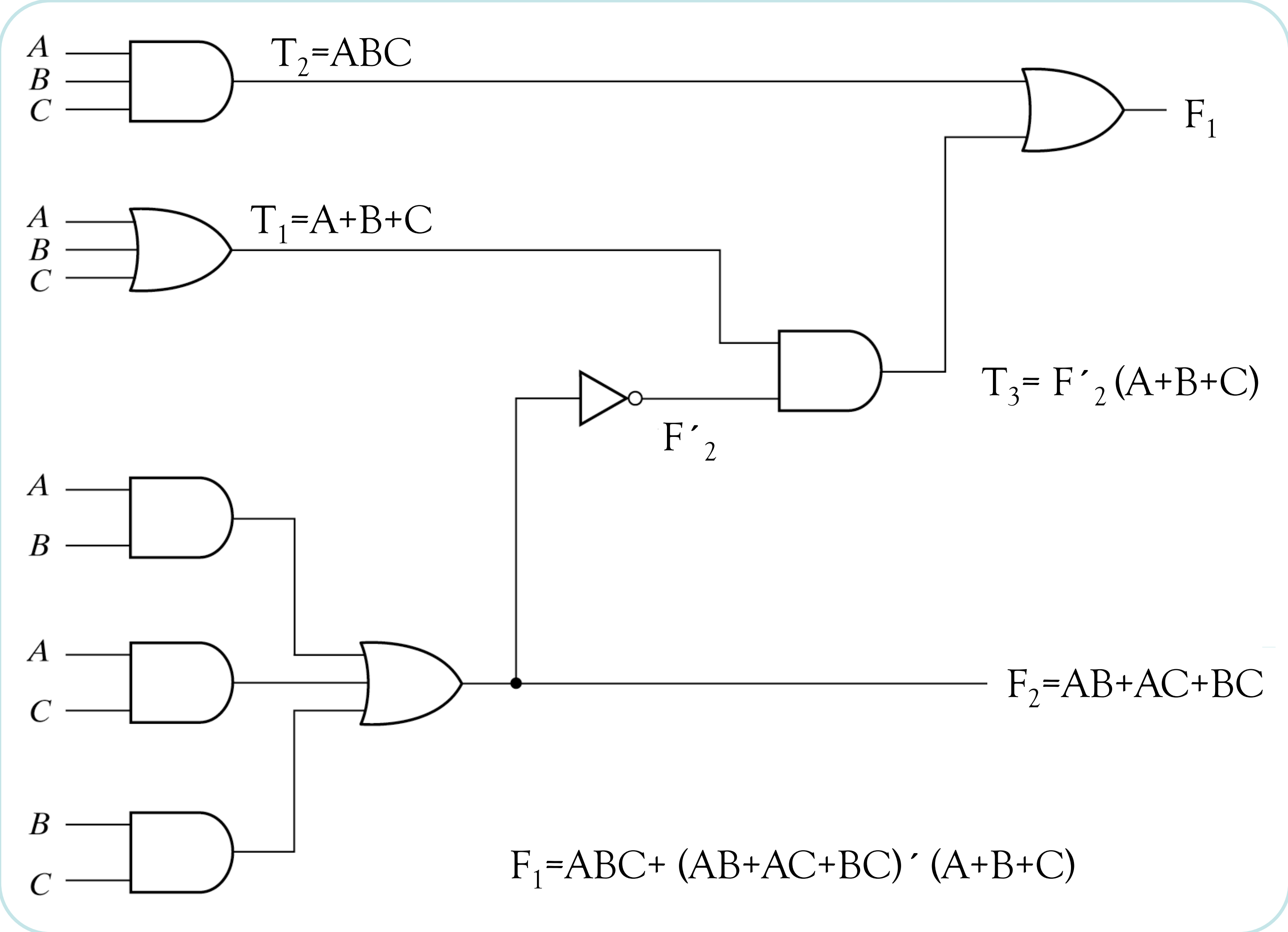
4-bits Carry Lookahead adder med input carry C_0



Generell analyseprosedyre for digitale kretser

1) Sett funksjonsnavn på ledningene
2) Finn funksjonene
3) Kombiner funksjonsuttrykkene

Eksempel


A
B
C
T2=ABC
F1
A
B
C
T1=A+B+C
T3= F´2 (A+B+C)
F´2
A
B
A
C
F2=AB+AC+BC
B
C
F1=ABC+ (AB+AC+BC)´ (A+B+C)

Oppsummering

- Binær addisjon
- Negative binære tall - 2'er komplement
- Binær subtraksjon
- Binær adder
  – Halvadder
  – Fulladder
  – Flerbitsadder
  – Carry propagation / carry lookahead
- Generell analyseprosedyre