erasmus

HOGESCHOOL BRUSSEL

IT Essentials

Deel I: Getalrepresentaties en schakelingen

3: Rekenschakelingen

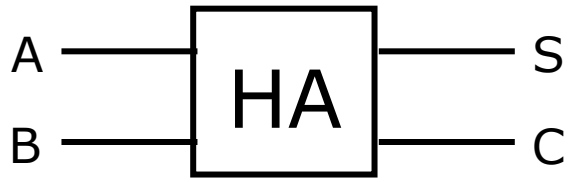
INHOUD

- Halve opteller (half adder)
- Volledige opteller (full adder)
- Intermezzo: maken van logische schakelingen
- Parallel-opteller
- Serie-opteller
- Simultaan-opteller
- Halve aftrekker (half subtractor)
- Volledige aftrekker (full subtractor)

erasmus
HOGESCHOOL BRUSSEL

HALVE OPTELLER (HALF ADDER)

- 2 bits ingang (A en B)
- S: sum (sombit)
- C: carry (overdracht)



- ! Logische bewerking vs. rekenkundige bewerking
- Logische bewerking: 1+1 = 1
- Rekenkundige bewerking: 1+1 =10

HALVE OPTELLER (HALF ADDER)

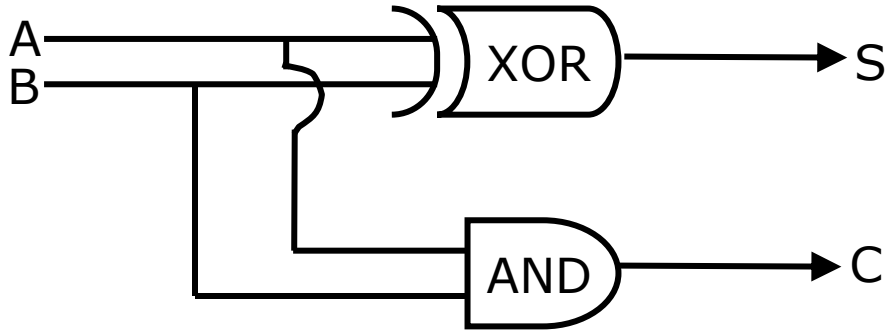
- We moeten volgende gedragstabel bekomen:

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

erasmus
HOGESCHOOL BRUSSEL

HALVE OPTELLER (HALF ADDER)

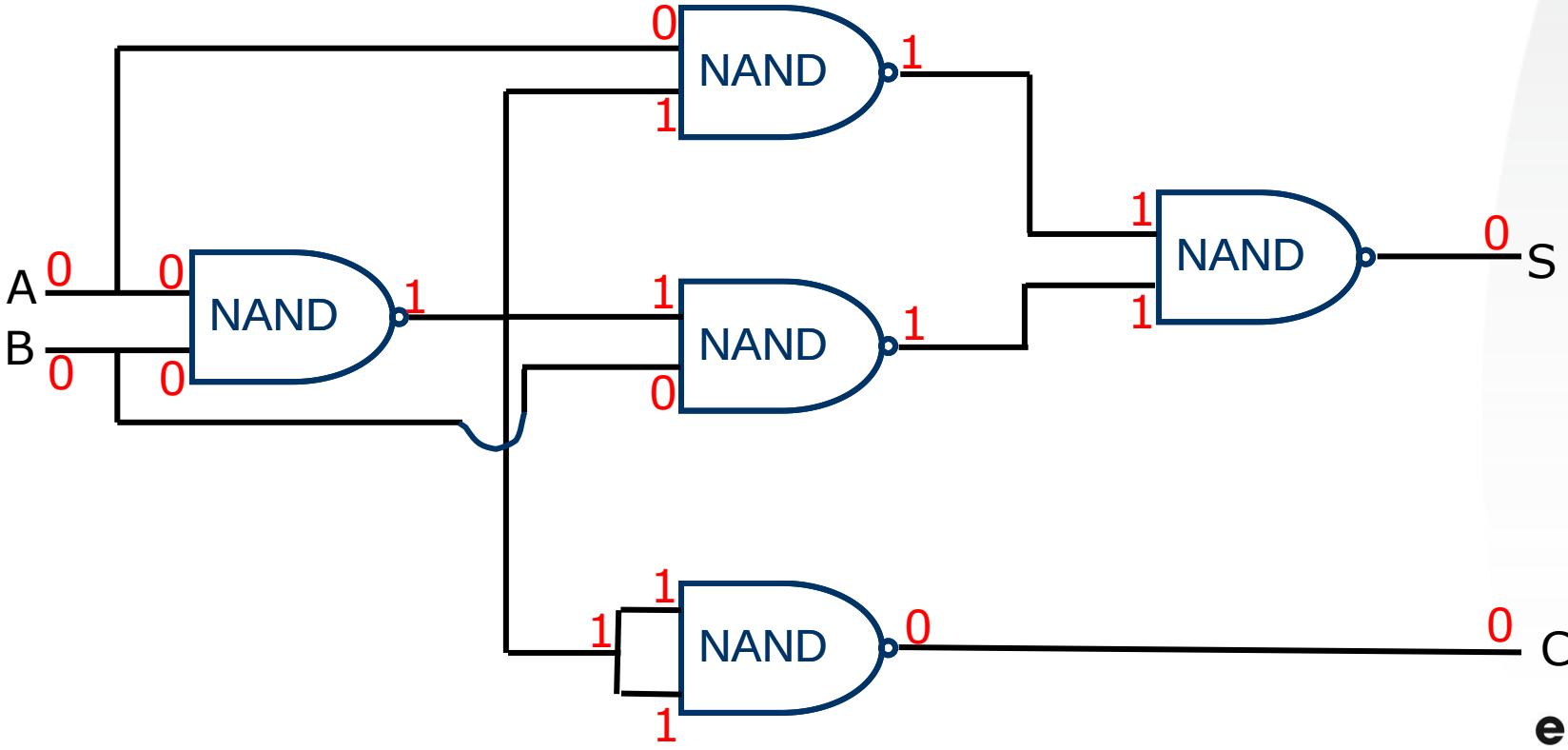
- XOR-schakeling voor Sombit
- AND-schakeling voor Carrybit (overdracht)



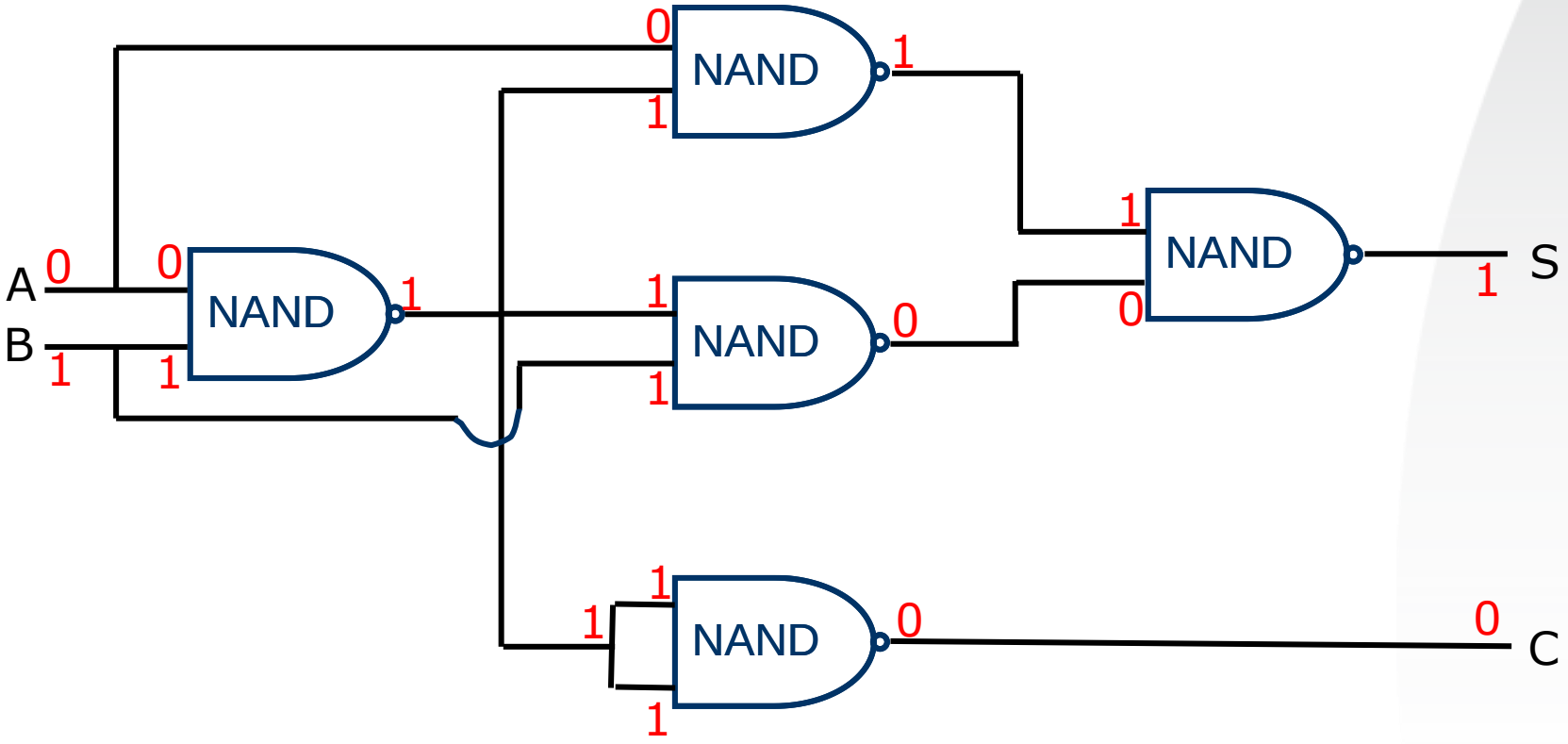
erasmus
HOGESCHOOL BRUSSEL

HALVE OPTELLER (HALF ADDER)

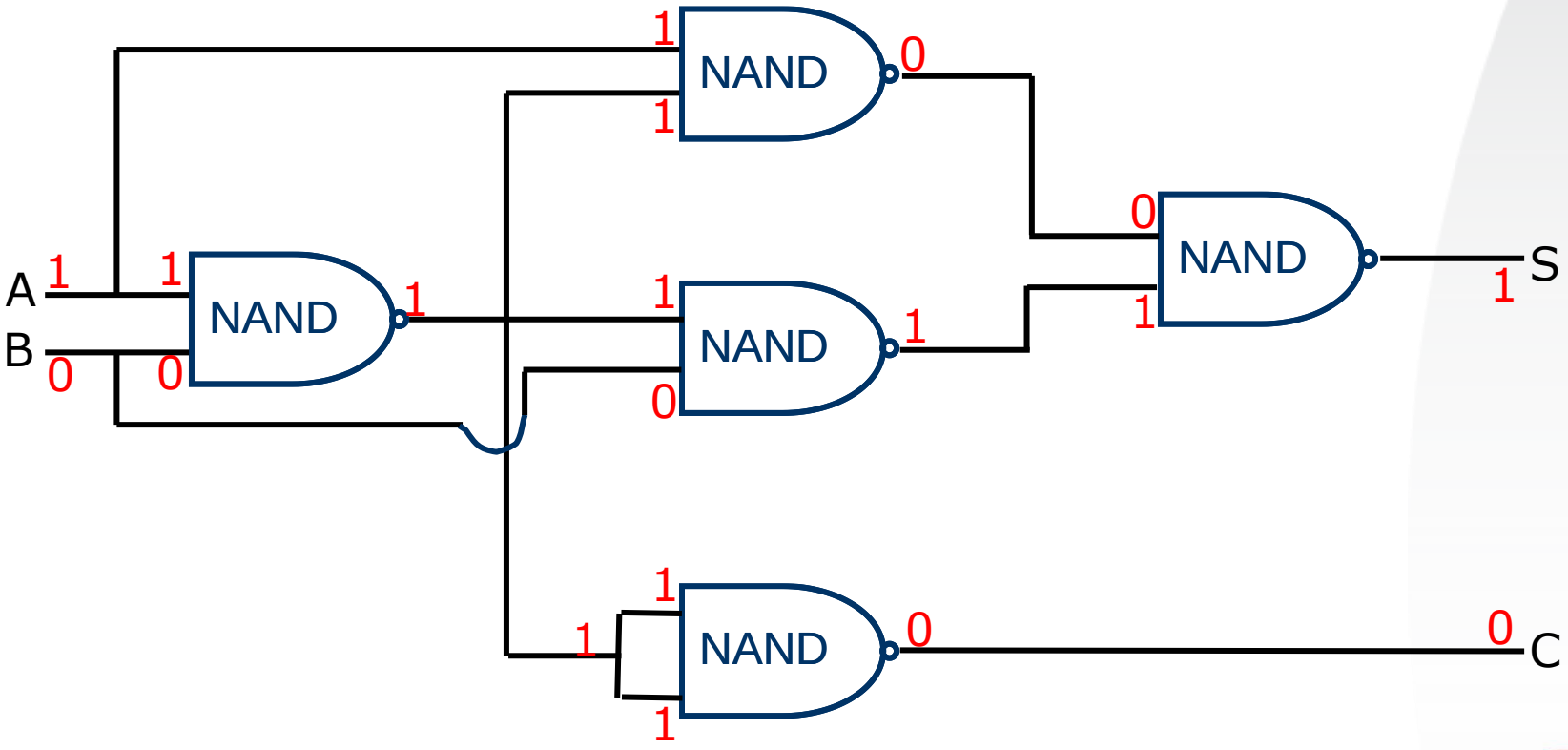
- Vaak wil men liever 1 soort schakeling gebruiken (bv. NOR of NAND) wegens lagere productiekost.
- NOR en NAND zijn computationeel compleet: elke combinatorische functie kan gerealiseerd worden met enkel NANDs of NOR's.
- De halve opteller kan ook op andere wijzen geconstrueerd worden.



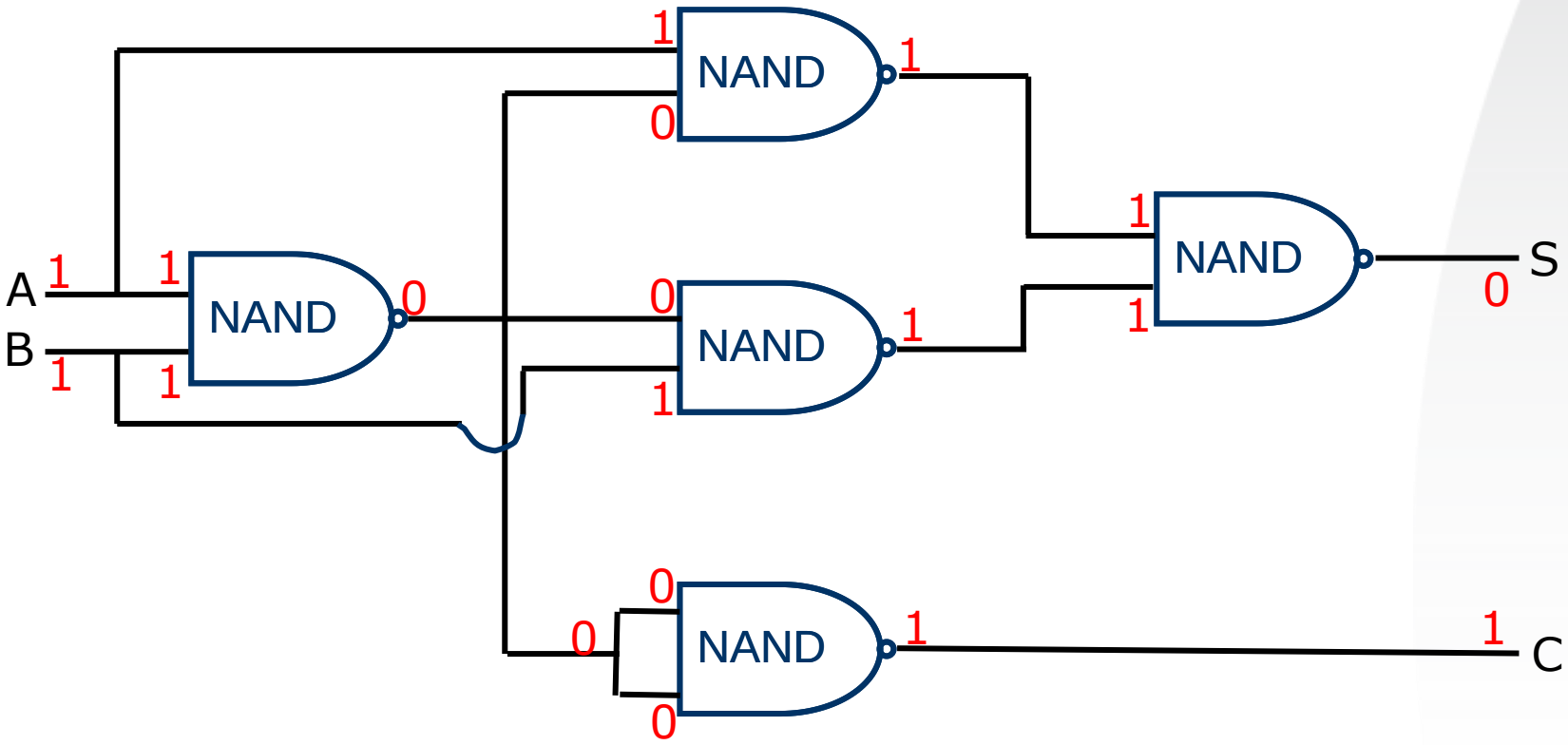
HALVE OPTELLER (HALF ADDER)



HALVE OPTELLER (HALF ADDER)

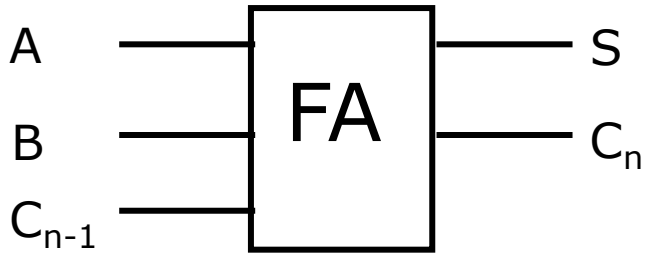


HALVE OPTELLER (HALF ADDER)



VOLLEDIGE OPTELLER (FULL ADDER)

- Halve opteller: we kunnen de overdracht of carry niet in rekening brengen
- Volledige opteller: overdracht in rekening brengen



erasmus
HOGESCHOOL BRUSSEL

VOLLEDIGE OPTELLER (FULL ADDER)

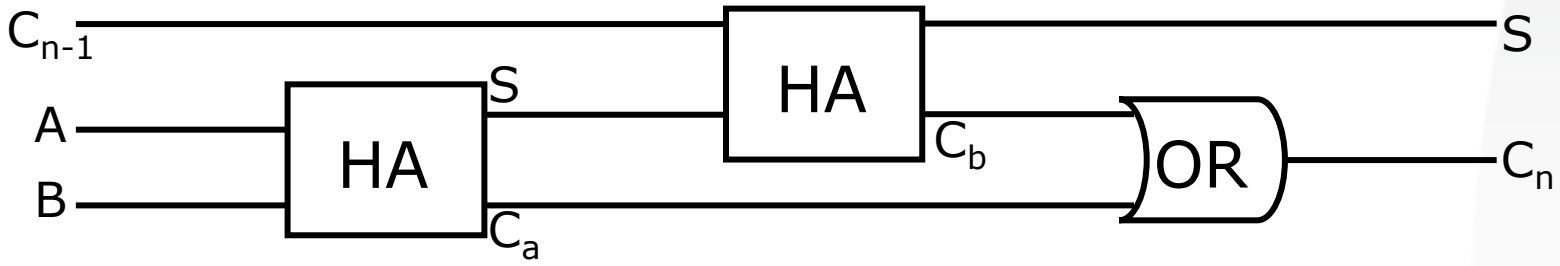
- We willen deze waarheidstabel bekomen

A	B	C_{n-1}	S	C_n
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

erasmus
HOGESCHOOL BRUSSEL

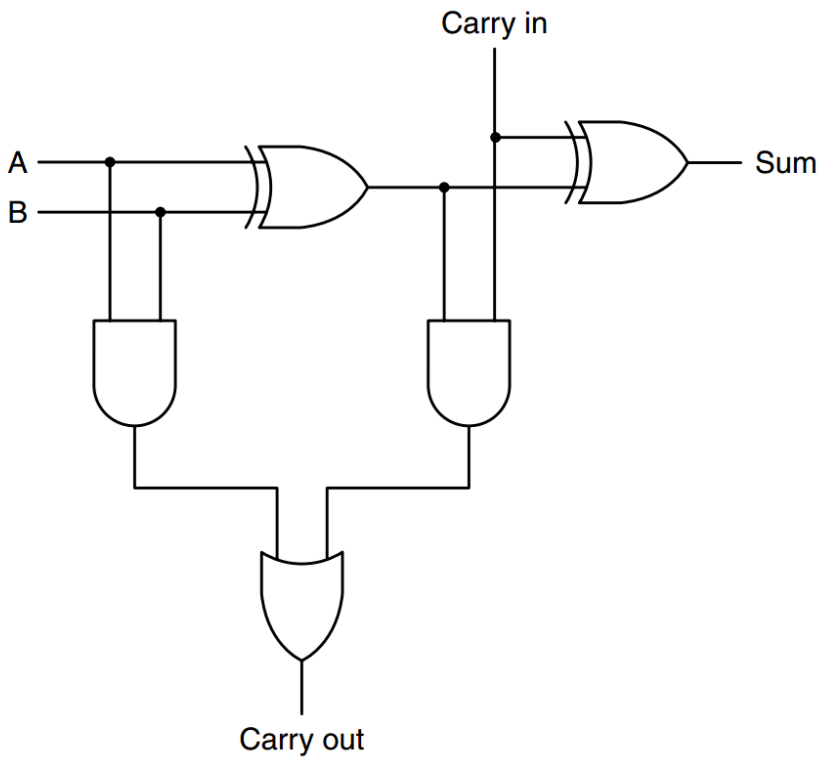
VOLLEDIGE OPTELLER (FULL ADDER)

- Ook voor de volledige opteller zijn er verschillende mogelijkheden
- Naargelang prijs, snelheid, … zal men een bepaalde schakeling kiezen



VOLLEDIGE OPTELLER (FULL ADDER)

- Een mogelijke optelschakeling



erasmus
HOGESCHOOL BRUSSEL

VOLLEDIGE OPTELLER (FULL ADDER)

A	B	C_{n-1}	S	C_n
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Interpretatie van de uitgangen als bewerking op de ingangen:

$$S = \bar{A}B\bar{C}_{n-1} + A\bar{B}\bar{C}_{n-1} + \bar{A}\bar{B}C_{n-1} + ABC_{n-1}$$

Deze vergelijkingen bepalen de schakelingen die we zullen construeren

erasmus
HOGESCHOOL BRUSSEL

VOLLEDIGE OPTELLER (FULL ADDER)

A	B	C_{n-1}	S	C_n
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Interpretatie van de uitgangen als bewerking op de ingangen:

$$C_n = AB\overline{C}_{n-1} + \bar{A}BC_{n-1} + A\bar{B}C_{n-1} + ABC_{n-1}$$

Deze vergelijkingen bepalen de schakelingen die we zullen construeren

erasmus HOGESCHOOL BRUSSEL

VOLLEDIGE OPTELLER (FULL ADDER)

- Vereenvoudiging kan leiden tot minder logische poorten → goedkopere schakeling

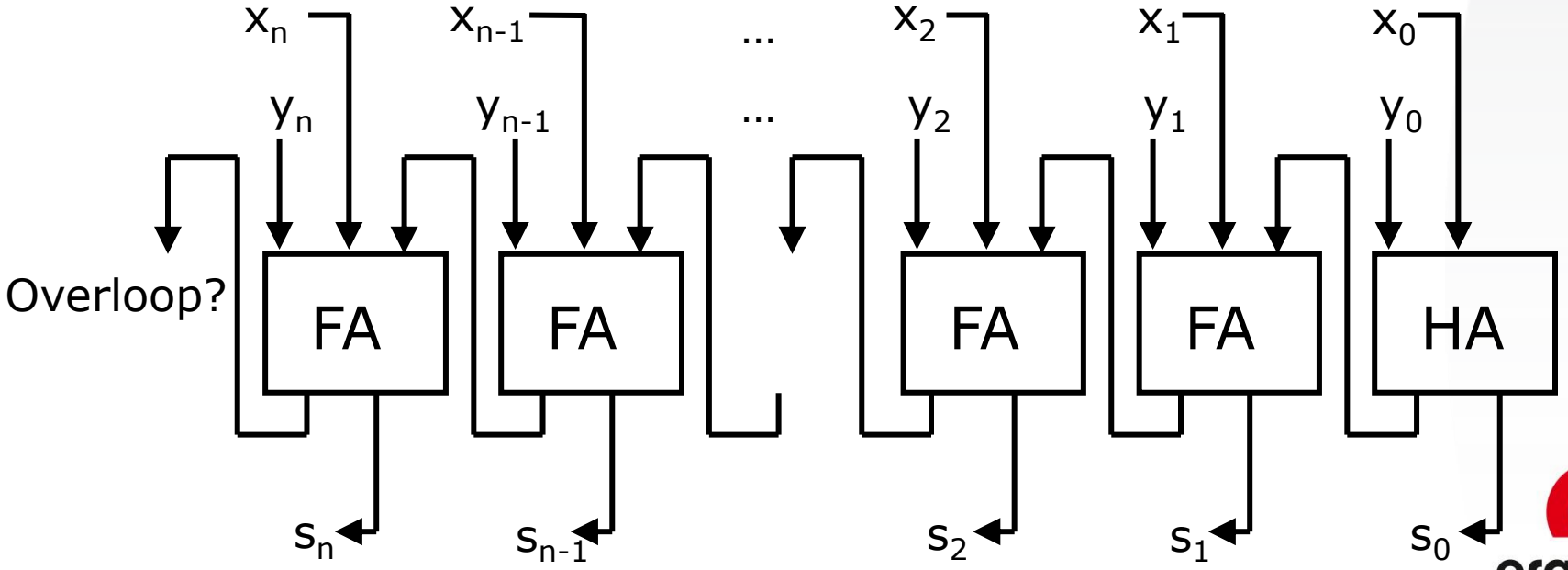
- Keuze van schakeling hangt af van prijs, snelheid, …

erasmus
HOGESCHOOL BRUSSEL

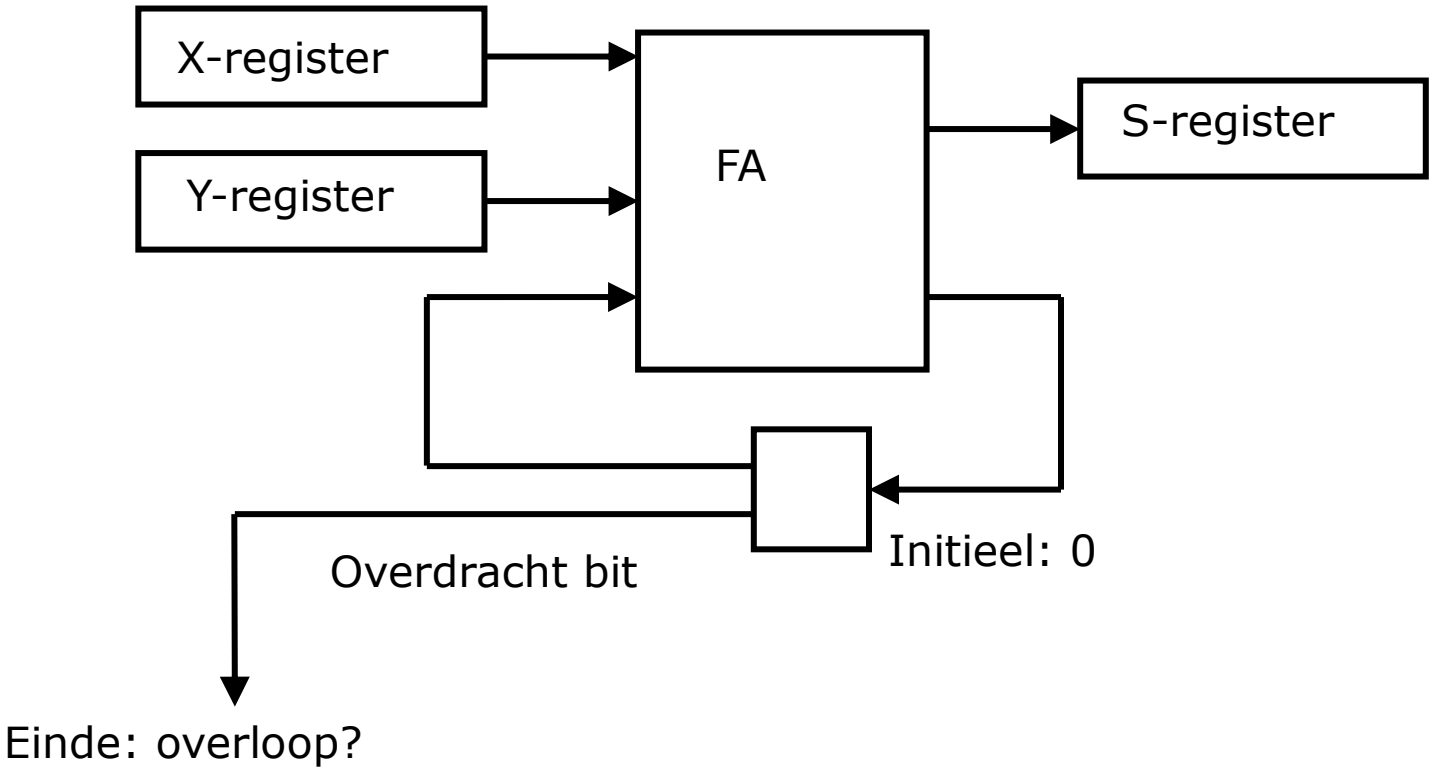
PARALLEL-OPTELLER (RIPPLE CARRY ADDER)

- Gegeven: halve opteller, volledige opteller
- Gevraagd: maak een schakeling zodat twee binaire getallen kunnen opgeteld worden

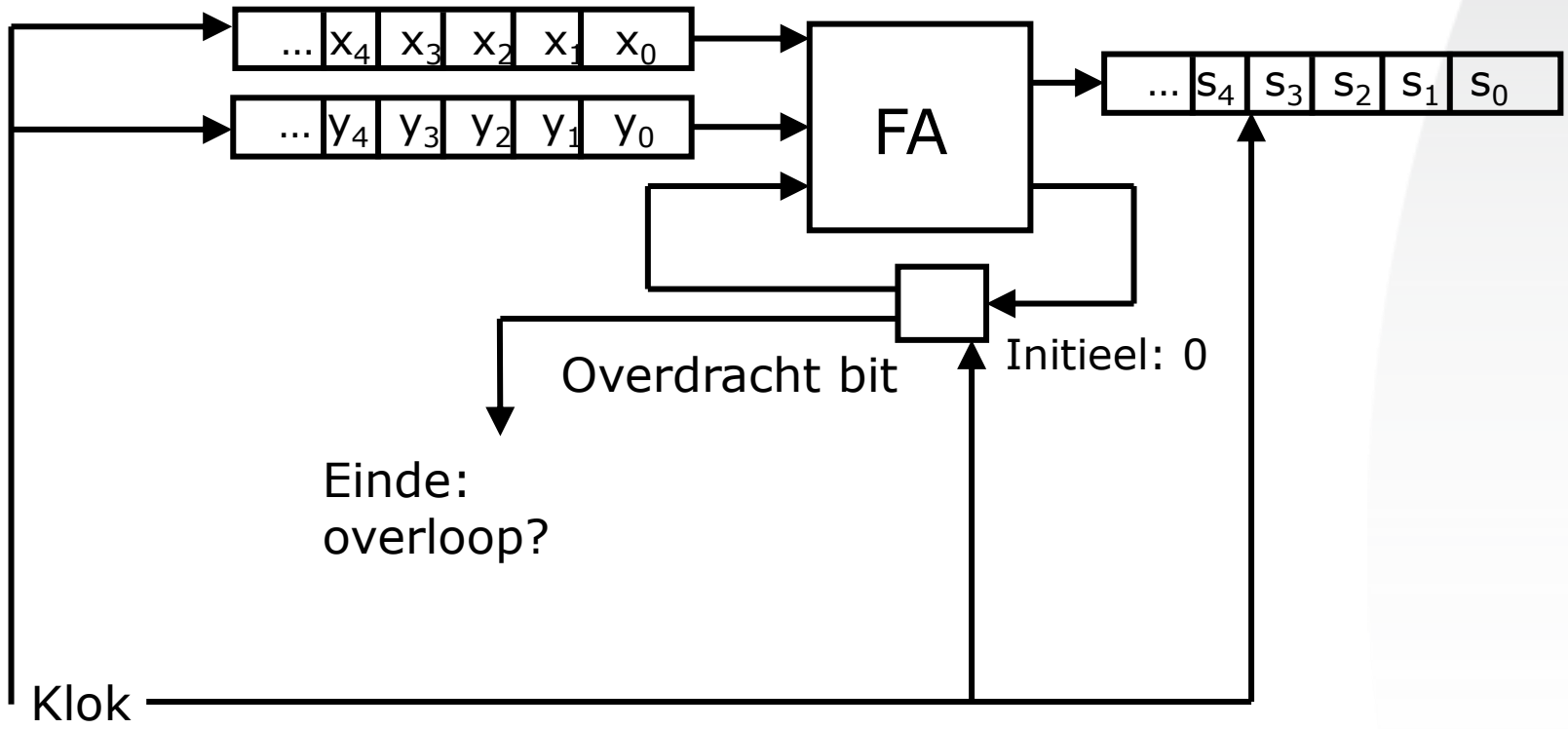
$X = x_n\ x_{n-1} \dots x_3\ x_2\ x_1\ x_0$
$Y = y_n\ y_{n-1} \dots y_3\ y_2\ y_1\ y_0$



SERIE-OPTELLER



SERIE-OPTELLER



SERIE-OPTELLER

- Slechts 1 FA-schakeling nodig
- Duurt langer, zeker wanneer er veel bits aan te pas komen

erasmus
HOGESCHOOL BRUSSEL

CARRY LOOKAHEAD ADDER

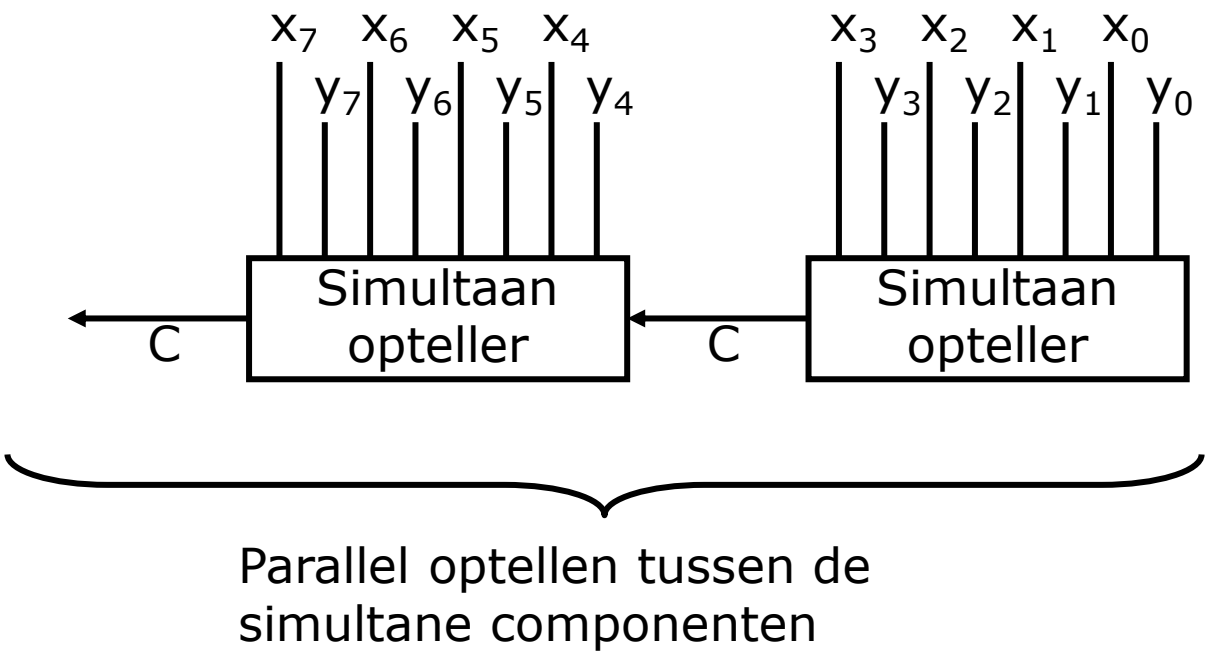
- Parallel-opteller heeft last van vertraging, doordat de overdracht-bit gekend moet zijn voor elke berekening in elke full adder

- Door te gaan kijken naar de input waarden in een full adder kan er een voorspelling worden gemaakt of er een carry zal worden gegenereerd, en moet er niet steeds gewacht worden.
- = snelheidswinst

erasmus
HOGESCHOOL BRUSSEL

LOOKAHEAD ADDER IN BLOCKS

- compromis tussen geldkost van lookahead opteller en tijdskost van parallel-opteller



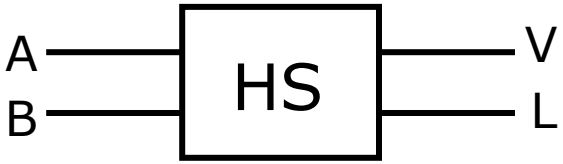
HALVE AFTREKKER (HALF SUBTRACTOR)

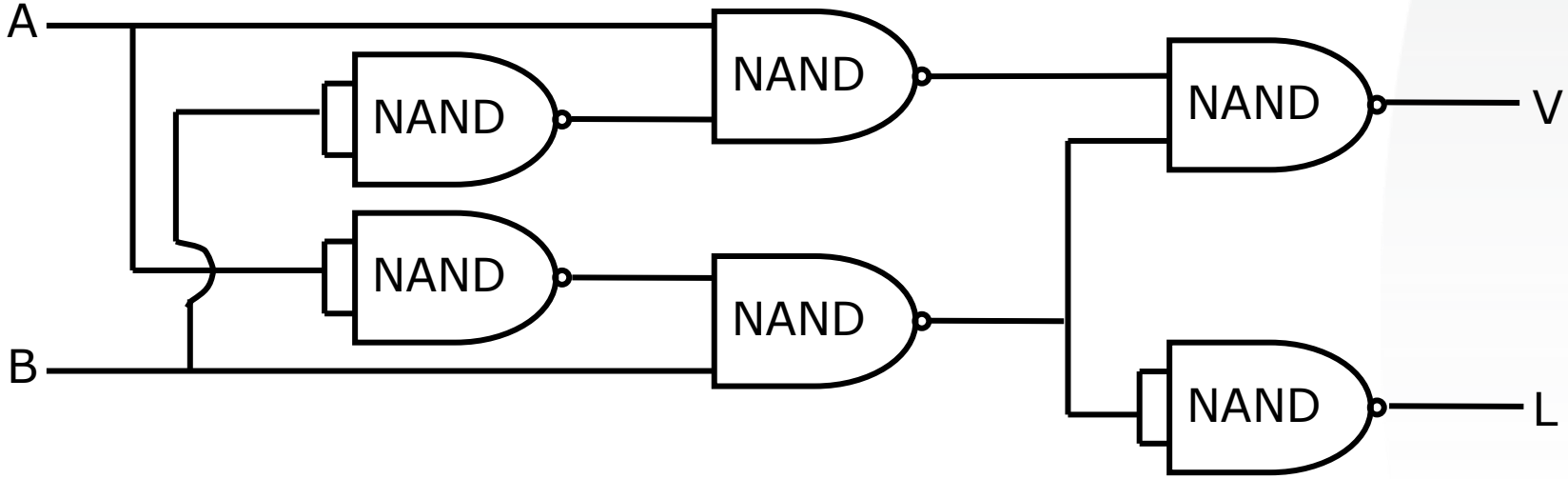
- Zelfde principes voor andere bewerkingen, zoals aftrekker (subtractor)
- Gedragstabel halve aftrekker: A-B
  - V = verschil
  - L = lenen

A	B	V	L
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

erasmus
HOGESCHOOL BRUSSEL

HALVE AFTREKKER (HALF SUBTRACTOR)

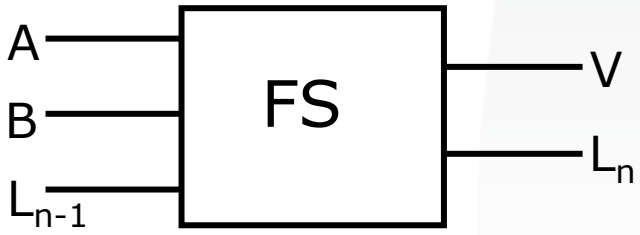




VOLLEDIGE AFTREKKER (FULL SUBTRACTOR)

- L_n: leenbit
- V: A-B-L_{n-1}
- Als L_{n-1} = 1, hebben we een bit geleend

A	B	L_{n-1}	V	L_n
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



VOLLEDIGE AFTREKKER (FULL SUBTRACTOR)