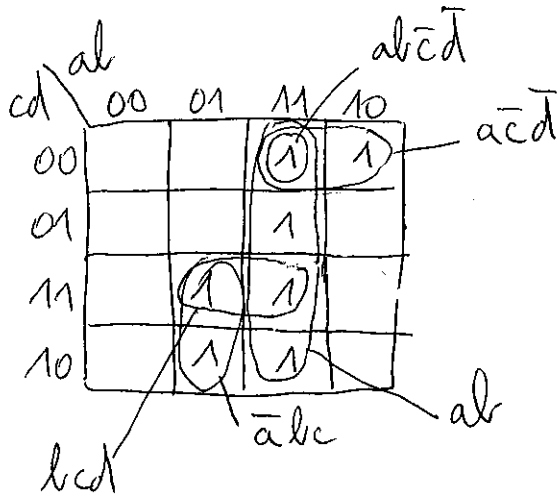


$$\begin{aligned}
 1 - \quad \Delta &= a\bar{c}\bar{d} + \bar{a}bc + ab\bar{c}\bar{d} + bcd + ab \\
 &= a\bar{c}\bar{d} + b(\bar{a}c + a) + bcd \\
 &= a\bar{c}\bar{d} + b(c + a) + bcd \\
 &= a\bar{c}\bar{d} + bc + ab + \cancel{bcd}
 \end{aligned}$$

$$\Delta = ab + bc + a\bar{c}\bar{d}$$



$$\Delta = ab + bc + a\bar{c}\bar{d}$$

2 - pas signé

$$\begin{array}{r}
 398 \\
 \oplus 215 \\
 \hline
 101
 \end{array}$$

débordement car
retenue C=1

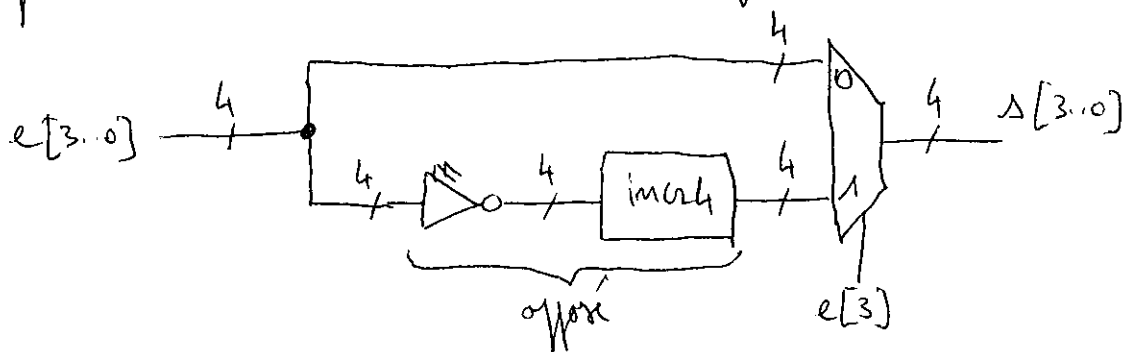
$$\begin{array}{r}
 \overset{1}{1}1000\overset{1}{1}\overset{1}{1}\overset{1}{1}10 \\
 \oplus 011010111 \\
 \hline
 001100101 \\
 \leftarrow C=1
 \end{array}$$

signé

$$\begin{array}{r}
 -114 \\
 \oplus +215 \\
 \hline
 +101
 \end{array}$$

pas de débordement car
signes cohérents (V=0)

3 - implémentation à l'aide de modules déjà vus :



Avec table de vérité

$e[3]$	$e[2]$	$e[1]$	$e[0]$	$\Delta[3]$	$\Delta[2]$	$\Delta[1]$	$\Delta[0]$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

← cas spécial : +8
min 4 bits codés,
résultat indéterminé

$$\Delta[0] = e[0]$$

$$\Delta[3] = 0$$

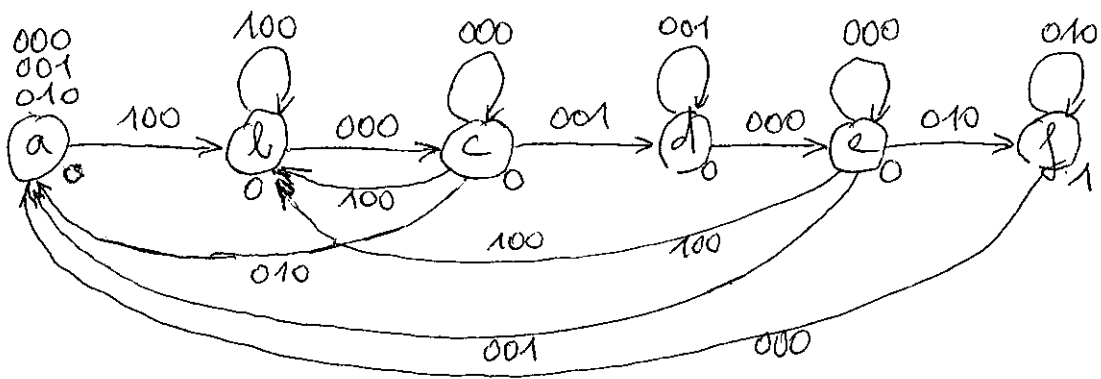
$$\Delta[1] = e[1] \cdot \overline{e[0]} + e[1] \cdot \overline{e[3]} + \overline{e[1]} \cdot e[0] \cdot e[3]$$

$e[1..0]$ \ $e[3..2]$	00	01	11	10
00				
01			1	1
11	1	1		
10	1	1	1	1

$$\Delta[2] = \overline{e[3]} \cdot e[2] + e[2] \cdot \overline{e[1]} \cdot \overline{e[0]} + e[3] \cdot \overline{e[2]} \cdot e[0] + e[3] \cdot e[2] \cdot \overline{e[1]}$$

$e[1..0]$ \ $e[3..2]$	00	01	11	10
00		1	1	
01		1		1
11		1		1
10		1		1

4- vecteur d'entris : (dieze, a, b) ; de sortie : (ouvert)



# a b	état	état
000	a	a
001	a	a
010	a	b
100	a	c
000	b	*
001	b	*
010	b	*
100	b	b
000	c	c
001	c	d
010	c	a
100	c	b
000	d	e
001	d	d
010	d	*
100	d	*
000	e	e
001	e	a
010	e	f
100	e	b
000	f	a
001	f	*
010	f	f
100	f	*

état	ouvert
a	0
b	0
c	0
d	0
e	0
f	1

- ps de simplification

6 états \rightarrow 3 bits X, Y, Z

- assignation

Z \ XY	00	01	11	10
0	a	b	c	d
1			f	e

# a b	xyz	xyz
000	000	000
001	000	000
010	000	000
100	000	010
000	010	110
001	010	***
010	010	***
100	010	010
000	110	110
001	110	100
010	110	000
100	110	010
000	100	101
001	100	100
010	100	***
100	100	***
000	101	101
001	101	000
010	101	111
100	101	010
000	111	000
001	111	***
010	111	111
100	111	***

xyz	ouvert
000	0
010	0
110	0
100	0
101	0
111	1
001	*
011	*

z \ xy	00	01	11	10
0				
1	*	*	1	

$$\text{ouvert} = yz$$

rien d'évident : on prend 3 bases
 Δ et on profite des exclusions mutuelles
entre diges, a, b

$$DX = \overline{\text{dige}} \cdot \bar{a} \cdot \bar{b} (\bar{x}y\bar{z} + x\bar{y}\bar{z} + x\bar{y}z + x\bar{y}z) + b \cdot x\bar{y}\bar{z} + b \cdot x\bar{y}z + a \cdot x\bar{y}z + a \cdot xy\bar{z}$$

$$DY = \overline{\text{dige}} \cdot \bar{a} \cdot \bar{b} (\bar{x}y\bar{z} + x\bar{y}\bar{z}) + \text{dige} \cdot \bar{x}y\bar{z} + \text{dige} \cdot x\bar{y}\bar{z} + a \cdot x\bar{y}z + \text{dige} \cdot x\bar{y}z + a \cdot xy\bar{z}$$

$$DZ = \overline{\text{dige}} \cdot \bar{a} \cdot \bar{b} (x\bar{y}\bar{z} + x\bar{y}z) + a \cdot x\bar{y}z + a \cdot xy\bar{z}$$

5- voir cours

6-

arg	brg	drug	oemum	cnd	wink	
30	0	31	mem	0	0	fetch
31	0	21	ual	right13	0	extract. cst
0	21	1	ual	add		addition
30	20	30	ual	add	0	pc += 1
30	0	31	mem	0	0	fetch
31	0	21	ual	right13	0	extract cst
0	21	1	ual	addce	0	addition
30	20	30	ual	add	0	pc += 1
30	0	31	mem	0	0	fetch
30	20	30	ual	add	0	pc += 1 (pos de branch)

7-

SW = 0x9000 0000

LD = 0xB000 0000

xt SW, %r1

xt LD, %r2

st %r0, [%r2] ; extinction led

a: ld [%r1], %r3 ; lecture switch

andce %r3, 0b100 ; test #

be z0 ; pos #

b: ld [%r1], %r3 ; lecture switch

andce %r3, 0b100, %r0 ; #?

brz b ; # : reste en b

andce %r3, 0b111, %r0 ; 000?

brz c ; 000 : va en c

---etc---