



b)  $d4 = d8 = d16 = 1$

$$d2 = (m8 + m4 + m2' + m1 + L') * (m8 + m4 + m2' + m1 + L) \\ = (m8 + m4 + m2' + m1)$$

$$d1 = (m8' * m4' * m2' * m1) + (m8' * m4' * m2 * m1' * L) + \\ (m8' * m4' * m2 * m1) + (m8' * m4 * m2' * m1) + \\ (m8' * m4 * m2 * m1) + (m8 * m4' * m2' * m1') + \\ (m8 * m4' * m2 * m1') + (m8 * m4 * m2' * m1')$$

c)  $d4, d8, d16$ : none

$d2$ : 1 OR4

1 INV

$d1$ : 7 AND4

1 AND5

1 OR8

18 INV

3a)

d28	d29	d30	d31	d16	d8	d4	d2	d1	
1	0	0	0	1	1	1	0	0	28
0	1	0	0	1	1	1	0	1	29
0	0	1	0	1	1	1	1	0	30
0	0	0	1	1	1	1	1	1	31

b)  $d4 = d8 = d16 = 1$

$$d2 = (d28' * d29' * d30 * d31') + (d28' * d29' * d30' * d31)$$

$$d1 = (d28' * d29 * d30' * d31') + (d28' * d29' * d30' * d31)$$

c)  $d4, d8, d16$ : none

$d2$ : 2 AND4

1 OR2

6 INV

$d1$ : 2 AND4

1 OR2

6 INV

d) The one presented in class uses more gates than the design in Problem 2. Using the subsystem in this problem will only add to the complexity if the original design.

4a)  $d4 = d8 = d16 = 1$

$$d2 = d30 + d31$$

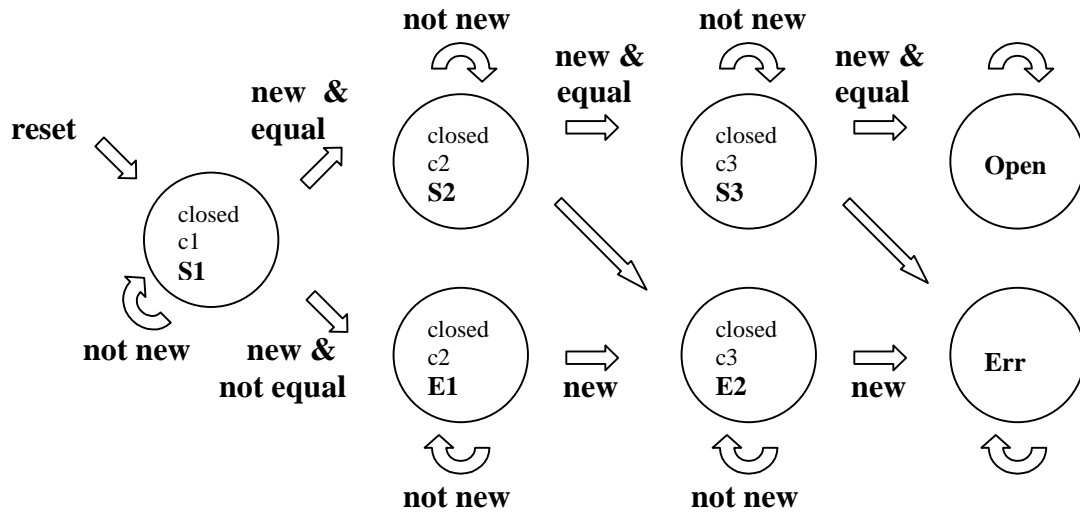
$$d1 = d29 + d31$$

b)  $d4, d8, d16$ : none

$d2$ : 1 OR2

$d1$ : 1 OR2

5a)



b) 7 states:

reset	new	equal	state	next state	mux	open/closed
1	-	-	-	S1	c1	closed
0	0	-	S1	S1	c1	closed
0	1	0	S1	E1	-	closed
0	1	1	S1	S2	c2	closed
0	0	-	S2	S2	c2	closed
0	1	0	S2	E2	-	closed
0	1	1	S2	S3	c3	closed
0	0	-	S3	S3	c3	closed
0	1	0	S3	Error	-	closed
0	1	1	S3	Open	-	open
0	-	-	Open	Open	-	open
0	0	-	E1	E1	-	closed
0	1	-	E1	E2	-	closed
0	0	-	E2	E2	-	closed
0	1	-	E2	Error	-	closed
0	-	-	Error	Error	-	closed

5c)

States: 4 bits

Mux: 3 bits

Open/closed: 1 bit

States							Mux			open/ closed	
S1	S2	S3	Open	E1	E2	Error	c1	c2	c3	open	closed
0001	0010	0100	1000	0011	0110	0000	001	010	100	1	0

reset	new	equal	state	next state	mux	open/closed
1	-	-	-	0001	001	0
0	0	-	0001	0001	001	0
0	1	0	0001	0011	-	0
0	1	1	0001	0010	010	0
0	0	-	0010	0010	010	0
0	1	0	0010	0110	-	0
0	1	1	0010	0100	100	0
0	0	-	0100	0100	100	0
0	1	0	0100	0000	-	0
0	1	1	0100	1000	-	1
0	-	-	1000	1000	-	1
0	0	-	0011	0011	-	0
0	1	-	0011	0110	-	0
0	0	-	0110	0110	-	0
0	1	-	0110	0000	-	0
0	-	-	0000	0000	-	0

5d)

Original implementation:  $\text{open/closed} = (\text{reset}' * \text{new} * \text{equal} * S3) + (\text{reset}' * \text{Open})$

New implementation:  $\text{open/closed} = (\text{reset}' * \text{new} * \text{equal} * S3) + (\text{reset}' * \text{Open})$

Therefore, same complexity.