Problem 1

a) Combinational. Suppose both numbers are 4 bit wide, we can use 4 bit comparators to compare each of the 4 bits. A bit comparator can be implemented as a combinational logic using XNOR, which is combinational. To compare whether one number is greater than another, we can use a combinational logic as well.

b) Sequential. By implementing the circuit using feedback, this circuit is sequential. We feed the previous computed value back into the input and add one to calculate the next output. Clearly, since it requires previous input, it requires memories; therefore, it must be sequential.

c) Combinational. To control the luminosity of the light, we only need the input of the position of the switch. Since it requires no previous states, it is combinational.

d) Sequential. Because the circuit needs to remember the previously (whether it is dimly on or brightly on) to determine the next output.

e) Combinational. There is not requirement on the knowledge of previous states.

f) Combinational. The output depends entirely on the inputs but never the previous states.

Problem 2

a)

Input ABCD	Output F3
0000	1
0001	0
0010	0
0011	1
0100	0
0101	0
0110	1
0111	0

1000	0
1001	1
1010	0
1011	0
1100	1
1101	0
1110	0
1111	1

b) F3 = A'B'C'D' + A'B'CD + A'BCD' + AB'C'D + ABC'D' + ABCD

c) 4 INVERTERs, 1 6-input OR gates, 6 4-input AND gates

Problem 3

a)

Input ABCD	Output F6
0000	1
0001	0
0010	0
0011	0
0100	0
0101	0
0110	1
0111	0
1000	0
1001	0
1010	0
1011	0
1100	1
1101	0
1110	0
1111	0

b) F6 = A'B'C'D' + A'BCD' + ABC'D'

c) 4 INVERTERs, 1 3-input OR gate, 3 4-input AND gates

b) F3 = F6 + A'B'CD + AB'C'D + ABCD

Problem 4

a)

Input: 4 4-bit binary number, close lock
Output: boolean flag, 0 to represent lock, 1 to represent unlock

b) On separate page

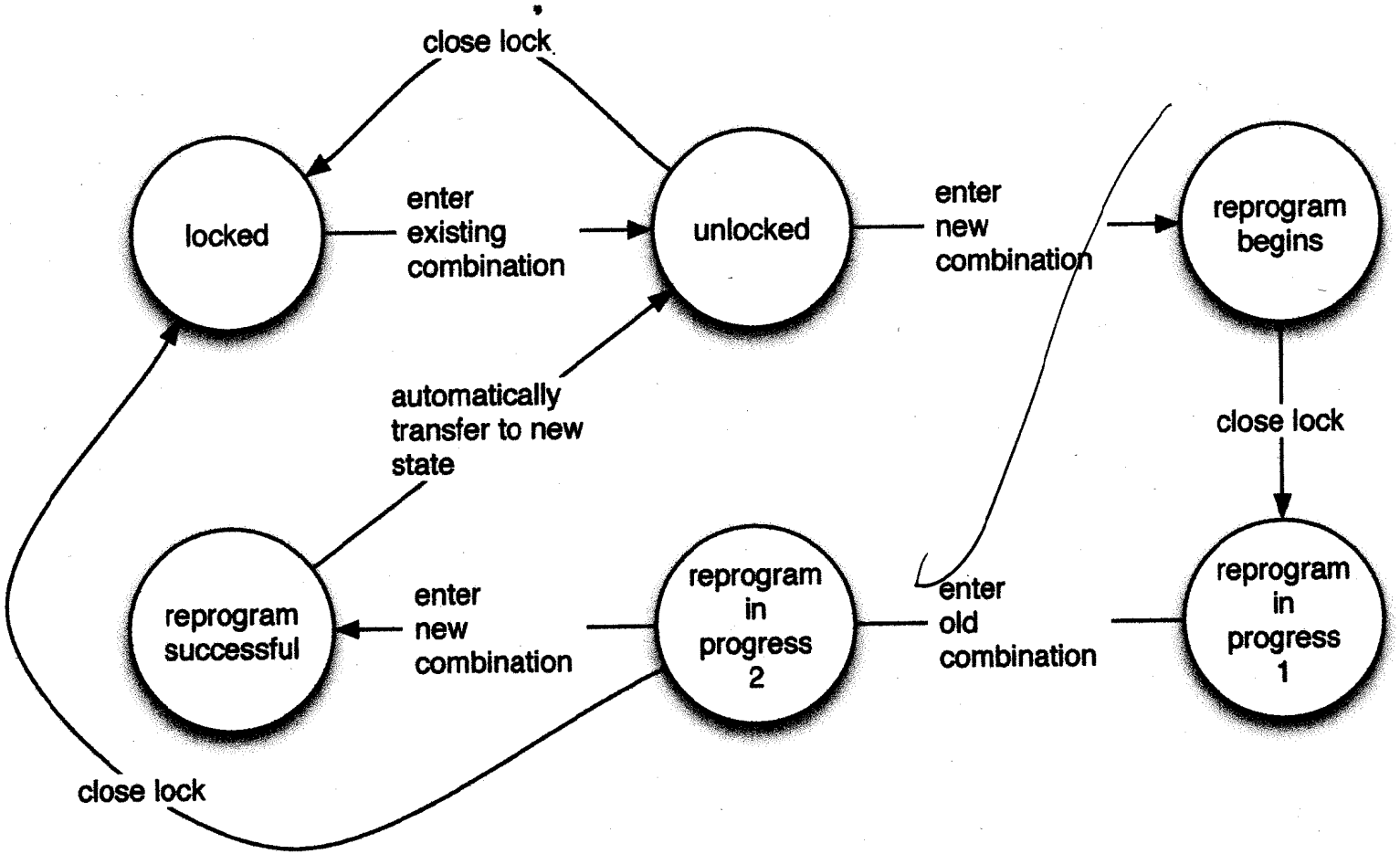
Problem 5

a)

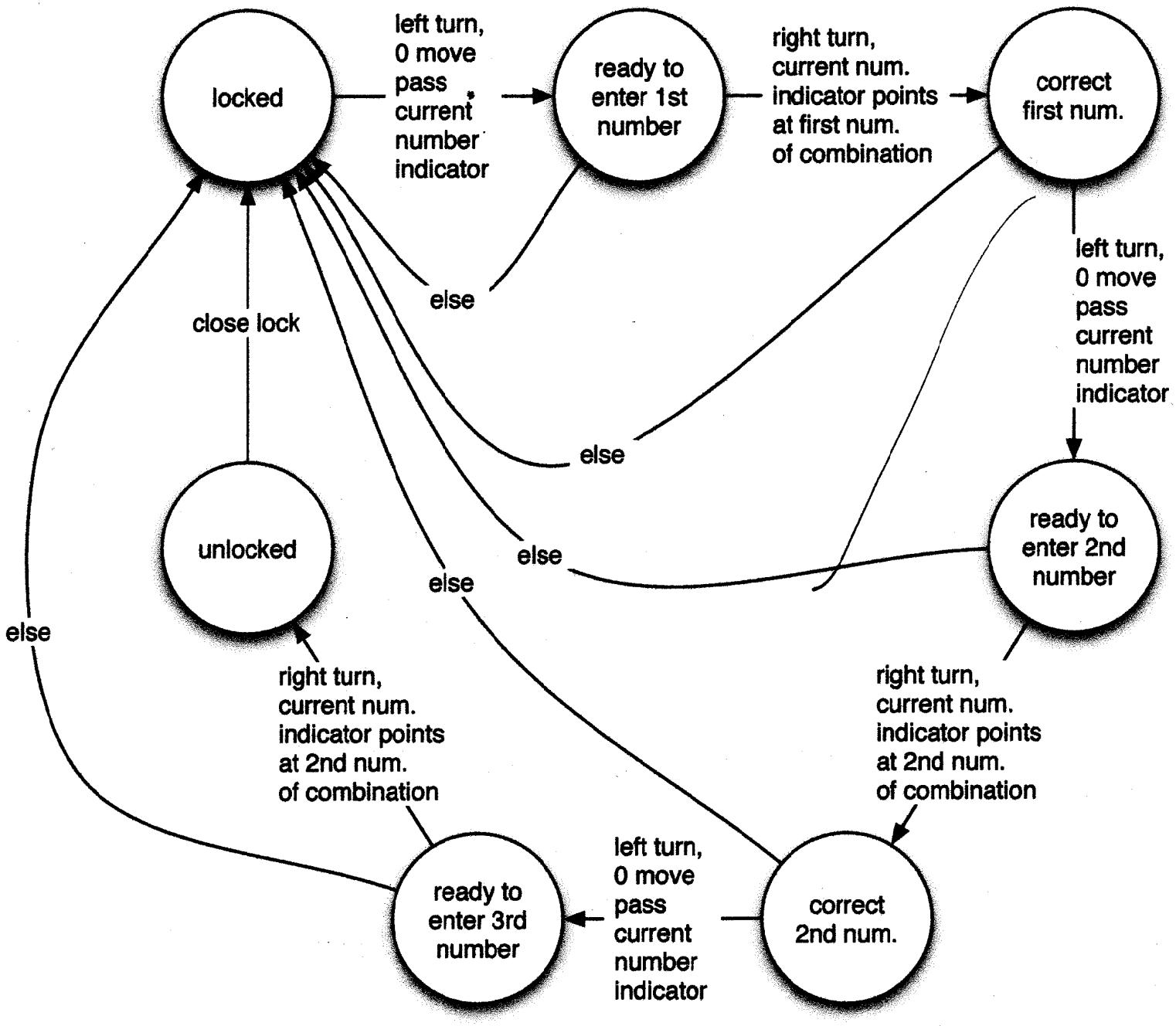
Input: 6-bit binary number & 1-bit directional, 0 represent left turn, 1 represent right turn
Output: boolean flat, 0 to represent lock, 1 to represent unlock

b) On separate page

Problem 4b


close lock
locked
enter
existing
combination
unlocked
enter
new
combination
reprogram
begins
automatically
transfer to new
state
close lock
reprogram
successful
enter
new
combination
reprogram
in
progress
2
enter
old
combination
reprogram
in
progress
1
close lock

Problem 5b


locked
left turn,
0 move
pass
current
number
indicator
ready to
enter 1st
number
right turn,
current num.
indicator points
at first num.
of combination
correct
first num.
left turn,
0 move
pass
current
number
indicator
else
close lock
else
else
unlocked
else
else
ready to
enter 2nd
number
right turn,
current num.
indicator points
at 2nd num.
of combination
right turn,
current num.
indicator points
at 2nd num.
of combination
ready to
enter 3rd
number
left turn,
0 move
pass
current
number
indicator
correct
2nd num.