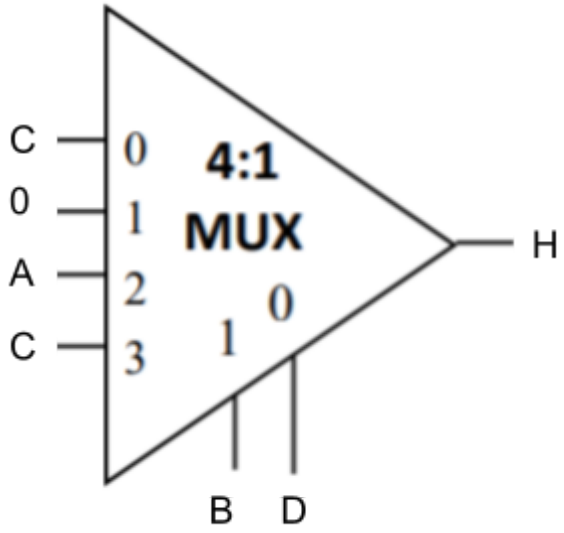
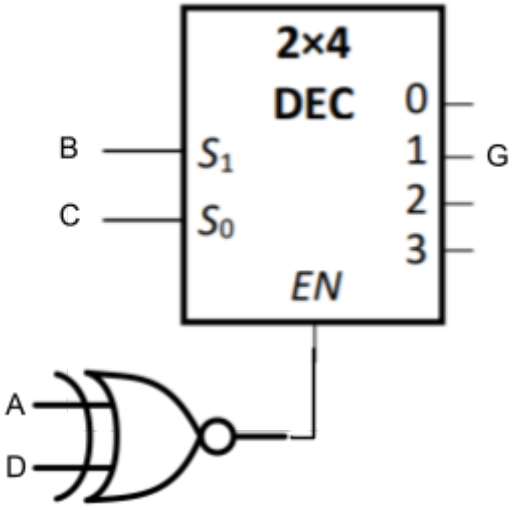
Tutorial T21
Toh Zhen Yu, Nicholas
A0201406Y

1.
a)
B+C.D

b)
Selector lines BD
Inputs C0AC



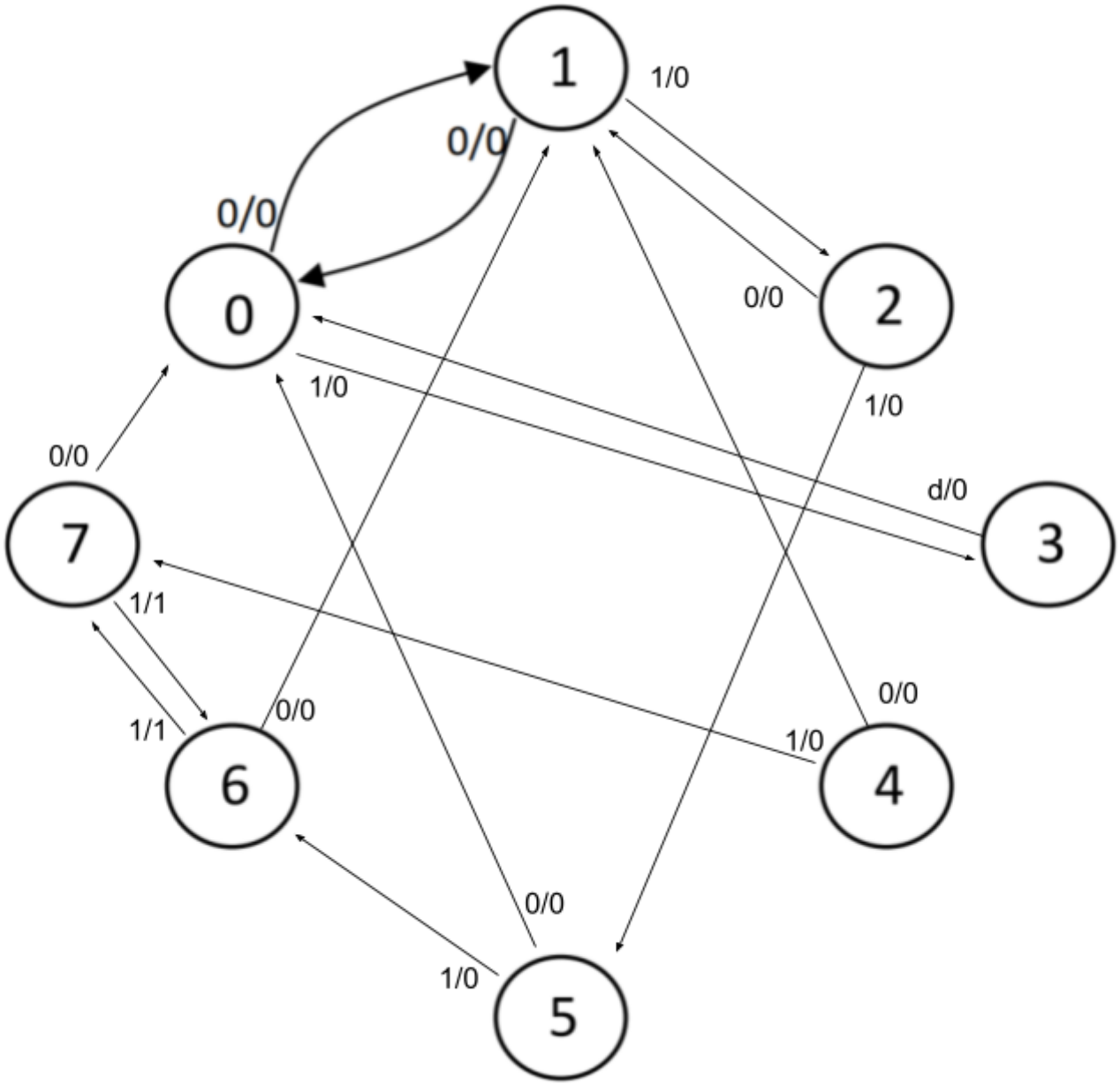
c)
SOP is A.B'.C.D + A'.B'.C.D'
G is given by C and B' and (A xnor D)
EN = (A xnor D)
S1 = B
S0 = C
G = output 1



d)
$F_2 = A_7 + A_6 + A_5 + A_4$

$F_1 = A_7 + A_6 + A_5'.A_4'.A_3 + A_5'.A_4'.A_2$
$F_0 = A_7 + A_6'.A_5 + A_6'.A_4'.A_3 + A_6'.A_4'.A_2'.A_1$

2.
JA = X.B.C'
KA = X'
JB = X
KB = A' + X'
JC = 1
KC = 1
Z = X.A.B
Because the inputs for flip flop C are always 1, its state always changes.



3.
a) 13
b) 27
c) 17
d) the branch will be taken if the array value is not smaller (greater or equal) than cutoff. In the initial case where the values are all smaller, then the branch will not be taken, hence it is better to predict branch not

taken, which will not cause any delays (saves 1 cycle). If we predict that the branch is not taken but it is taken (because the value is not smaller than cutoff), then this will cause the same delay of 1 cycle as in part c because we have executed incorrected instructions. In the opposite case, if we assume branch taken, but the branch is not taken (because the value is smaller than cutoff), then this will also cause a similar delay. It is best to choose the case with higher probability of occurrence, i.e. if it is more likely that the values are smaller, then we should assume branch not taken, and vice versa.