Computer Architecture and Programming Languages
Jacobs University Bremen
Fjolla Dedaj

Course: CO20-320241
30 September, 2019

Homework 3

Problem 3.1

Solution:

Distributive Law: DL
Identity Law: IL
Annulment Law: AL
Complement Law: CL
De Morgan's Theorem: DM
Double Negation Law: DN

a)

$x = (M + N)(\overline{M} + P)(\overline{N} + \overline{P}) =$

$DL := (M\overline{M} + MP + N\overline{M} + NP)(\overline{N} + \overline{P}) =$

$IL := (MP + N\overline{M} + NP)(\overline{N} + \overline{P}) =$

$DL := MP\overline{N} + MP\overline{P} + N\overline{M}\overline{N} + N\overline{M}\overline{P} + NP\overline{N} + NP\overline{P}$

$IL := M\overline{N}P + \overline{M}N\overline{P}$

b)

$z = \overline{A}B\overline{C} + AB\overline{C} + B\overline{C}D =$

$DL := B\overline{C}(\overline{A} + A + D) =$

$CL := B\overline{C}(1 + D) =$

$AL := B\overline{C}$

c)

$x = \overline{(M + N + P)Q} =$

$DM := \overline{M + N + P} + \overline{Q} =$

$DM := \overline{M} * \overline{N} * \overline{P} + \overline{Q}$

d)

$z = \overline{ABC + DEF} =$

$DM := \overline{ABC} * \overline{DEF} =$

$DM := (\overline{A} + \overline{B} + \overline{C}) * (\overline{D} + \overline{E} + \overline{F})$

e)

$z = \overline{A\overline{B} + C\overline{D} + EF} =$

$DM := \overline{A\overline{B}} * \overline{C\overline{D}} * \overline{EF} =$

$DM := (\overline{A} + \overline{\overline{B}}) * (\overline{C} + \overline{\overline{D}}) * (\overline{E} + \overline{F}) =$

$DN := (\overline{A} + B) * (\overline{C} + D) * (\overline{E} + \overline{F})$

f)
Throughout this problem, we use De Morgan's Theorem in every step and DN in some of the steps where we have double negation:

$$z = \overline{\overline{A + B\overline{C}} + \overline{D(E + \overline{F})}} =$$

$$= \overline{\overline{A + B\overline{C}}} * \overline{D * (E + \overline{F})} =$$

$$= \overline{\overline{A} * \overline{B * \overline{C}}} * \overline{D * \overline{E} * \overline{\overline{F}}} =$$

$$= \overline{\overline{A} * (\overline{B} + \overline{\overline{C}})} * \overline{D\overline{E}F} =$$

$$= \overline{\overline{A} * (\overline{B} + C)} * (\overline{D\overline{E}F}) =$$

$$= (\overline{A} + \overline{\overline{B} + C}) * (\overline{D} + \overline{\overline{E}} + \overline{F}) =$$

$$= (A + \overline{\overline{B}} * \overline{C}) * (\overline{D} + E + \overline{F}) =$$

$$= (A + B * \overline{C}) * (\overline{D} + E + \overline{F})$$

Problem 3.2
Solution:

Firstly, we obtain the following boolean expression from analyzing the circuit:

$$x = (\overline{A} \wedge \overline{B} \wedge D) \vee (A \wedge \overline{B} \wedge \overline{C}) \vee (\overline{A} \wedge \overline{B} \wedge \overline{C})$$

Now, from the expression above, we can create the Karnaugh Map by using gray encoding and plotting the 1s correspondingly.

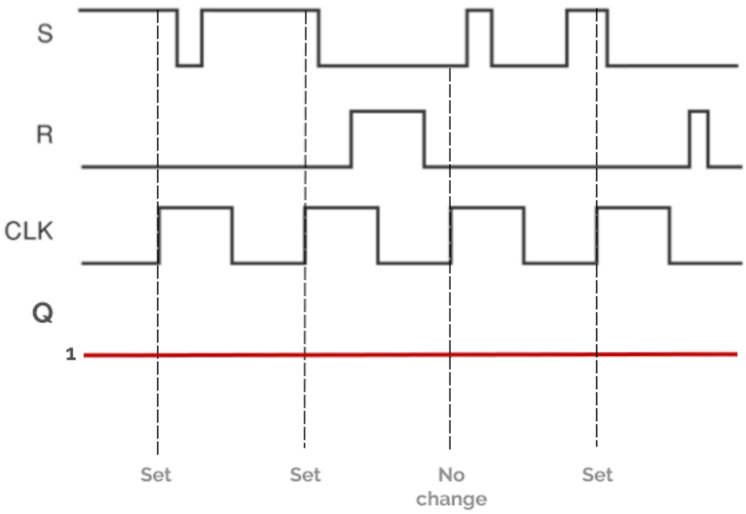
Karnaugh map

	$\overline{cd}$	$\overline{c}d$	cd	$c\overline{d}$
$\overline{ab}$	1	1	1	0
$\overline{a}b$	0	0	0	0
ab	0	0	0	0
$a\overline{b}$	1	1	0	0

After grouping the 1s in three different groups (two groups with two 1s and a group with a single 1) and analyzing the Karnaugh Map, we get the following simplified boolean expression:
$\overline{A}\overline{B}D + \overline{B}\overline{C}$

Problem 3.3

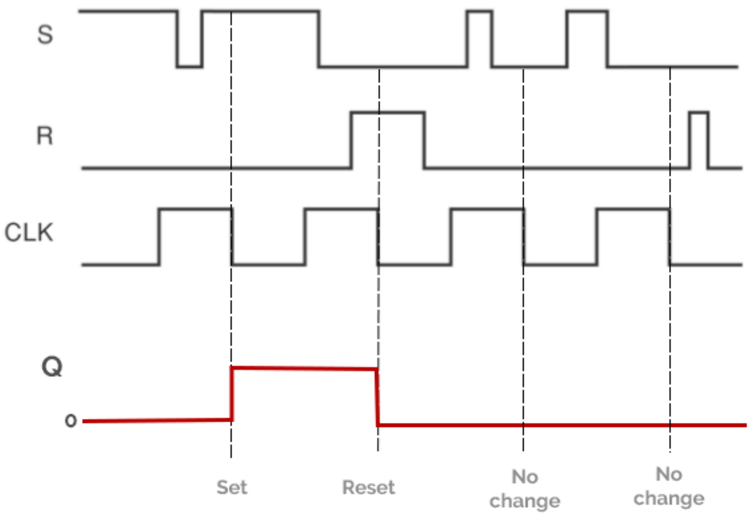
Solution:



Inputs			Output
S	R	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambiguous

Problem 3.4

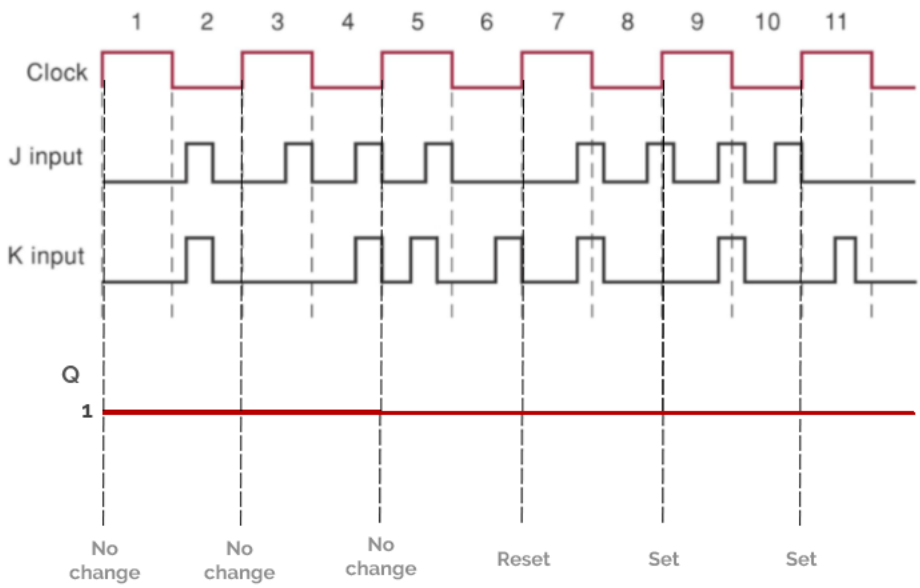
Solution:



Inputs			Output
S	R	CLK	Q
0	0	↓	Q_0 (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	Ambiguous

Problem 3.5

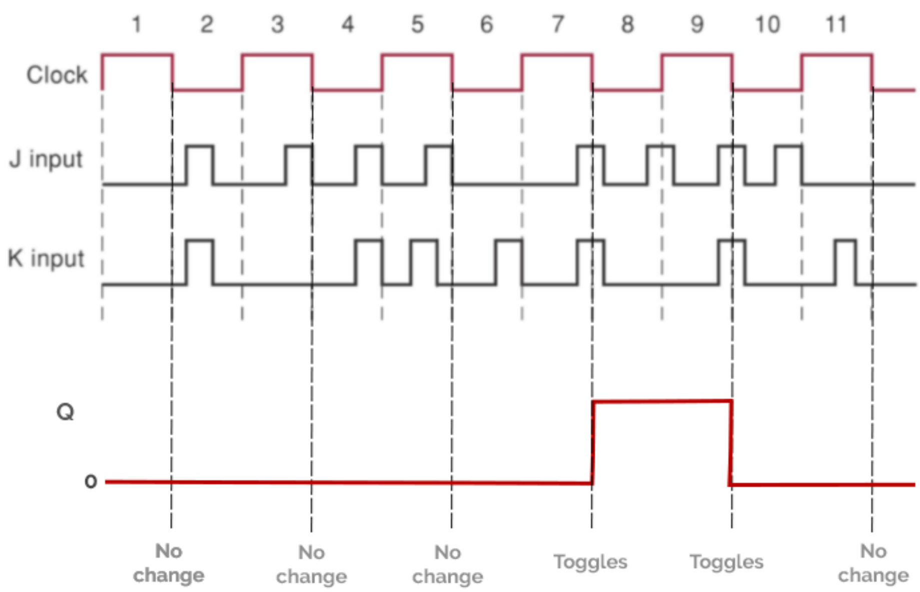
Solution:



J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q_0}$ (toggles)

Problem 3.6

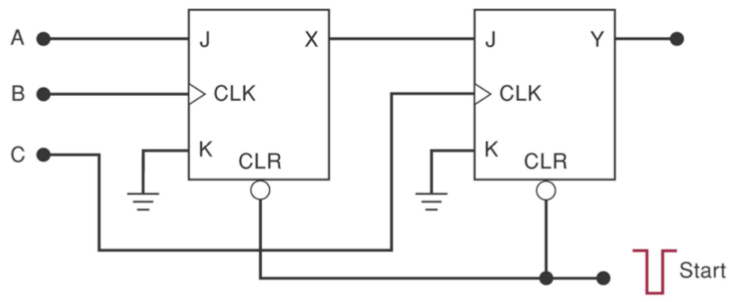
Solution:



J	K	CLK	Q
0	0	↓	Q_0 (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	$\overline{Q_0}$ (toggles)

Problem 3.7

Solution:



a)

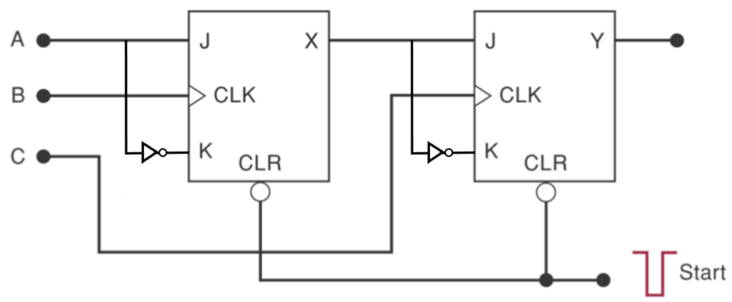
The sequence that makes Y go HIGH is A, B, C because Y can go HIGH only when C goes HIGH while X is already HIGH. X can go HIGH only if B goes HIGH while A is HIGH.

b) Need for Start Pulse

We know that the outputs X and Y need to be cleared to 0 before applying the A, B and C signals. To clear the outputs, we need a negative going Start pulse at the R input. R of JK flip-flop is active LOW.

c)

A D Flip-Flop may be implemented with a J-K Flip-Flop by tying the J input to the K input through an inverter, as it is shown below:



which is equivalent to: