

Εργαστηριακή Άσκηση 1

ΧΡΗΣΤΟΣ ΚΟΥΤΟΥΛΗΣ ΑΜ:5064

ΜΑΡΙΑ ΚΑΤΩΛΗ, ΑΜ:5083

ΟΜΑΔΑ 18

Βασικά Συνδυαστικά Κυκλώματα

Ερώτημα 1^ο

Ο αποκωδικοποιητής παίρνει σήμα από δύο εισόδους και το μεταφράζει στην αντίστοιχη λειτουργία.

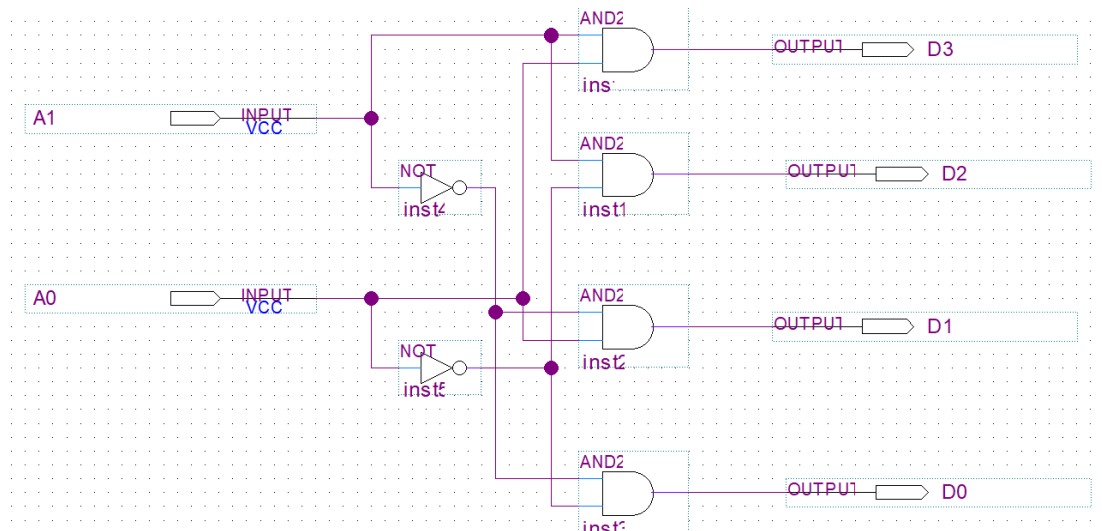
Πχ το 00->D0, 01->D1 κ.ο.κ

Από τον πίνακα αληθείας παρατηρούμε ότι στην πρώτη περίπτωση εμφανίζεται 1 μόνο στο D0, άρα η έξοδος για το D0 θα περνάει από πύλη AND2 η οποία θα πρέπει να βγάζει 1. Για να έχει αποτέλεσμα 1, πρέπει να συμπληρώσουμε με μια πύλη NOT και το A1 και το A0.

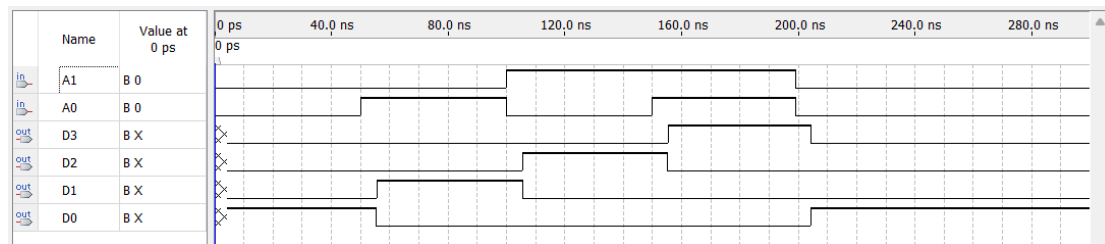
Αντίστοιχα, στη δεύτερη περίπτωση, παρατηρούμε ότι ανάβει μόνο το λαμπάκι D1 και με το ίδιο σκεπτικό θα βάλουμε το A1' και το A0' σε ένα AND2 το οποίο θα καταλήγει στο D1.

Οι υπόλοιπες δύο περιπτώσεις ακολουθούν την ίδια μεθοδολογία.

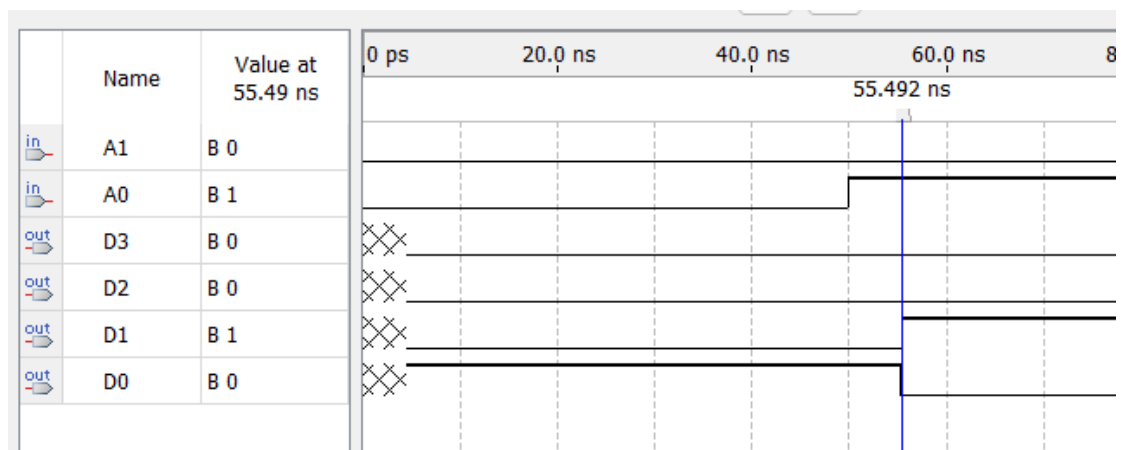
Το σχηματικό:



Η εξομοίωση:



Ελέγχουμε την χρονοκαθυστέρηση όταν το A0->1 και το D1->1. Με τη διαφορά υπολογίζουμε ότι καθυστερεί 5.492ns.



Propagation Delay						
	Input Port	Output Port	RR	RF	FR	FF
1	A0	D0		5.245	5.245	
2	A0	D1	5.492			5.492
3	A0	D2		5.272	5.272	
4	A0	D3	5.510			5.510
5	A1	D0		5.456	5.456	
6	A1	D1		5.633	5.633	
7	A1	D2	5.456			5.456
8	A1	D3	5.690			5.690

Στη στατική ανάλυση ελέγχουμε την 2^η γραμμή και τη στήλη RR και εντοπίζουμε ότι η καθυστέρηση είναι 5.492ns. Άρα το αποτέλεσμα είναι και στα δύο ίδιο.

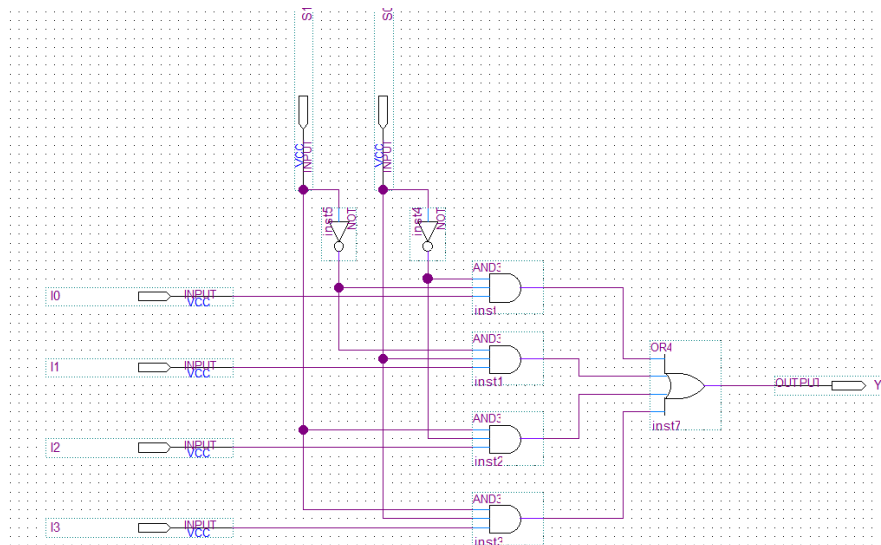
Ερώτημα 2°

Ο πολυπλέκτης χρησιμοποιεί εισόδους επιλογής οι οποίες αντιστοιχούν σε μία είσοδο δεδομένων. Δηλαδή, το $A1A0=00$ επιτρέπει την είσοδο δεδομένων $I0$ η οποία στην συνέχεια θα βγει στην έξοδο Y . Άρα, εφόσον έχουμε τέσσερις λειτουργίες θα χρησιμοποιήσουμε τέσσερα AND3 και ένα OR4. Στο κάθε AND3 θα μπου οι κατάλληλες δύο εισόδους επιλογής $A1A0$ και η αντίστοιχη λειτουργία τους I .

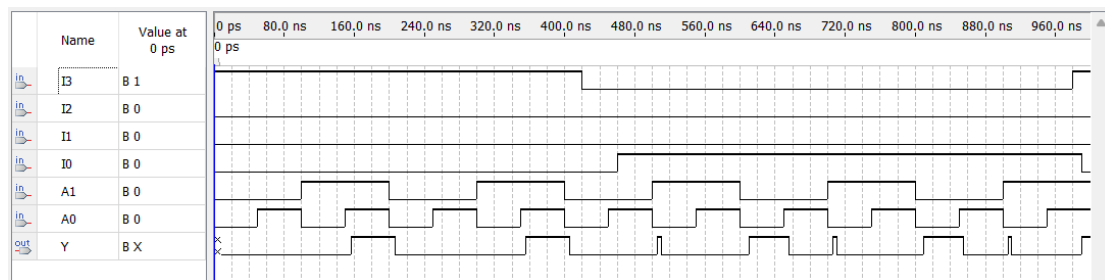
Πχ. Στην δεύτερη λειτουργία $A1A0=01$ θα μπει το $A1'$, το $A0$ και η είσοδος $I1$.

Οι έξοδοι των τεσσάρων AND3 θα καταλήξουν στο OR4 το οποίο δείχνει στο Y ποια είσοδος δεδομένων επιλέγεται σε κάθε περίπτωση.

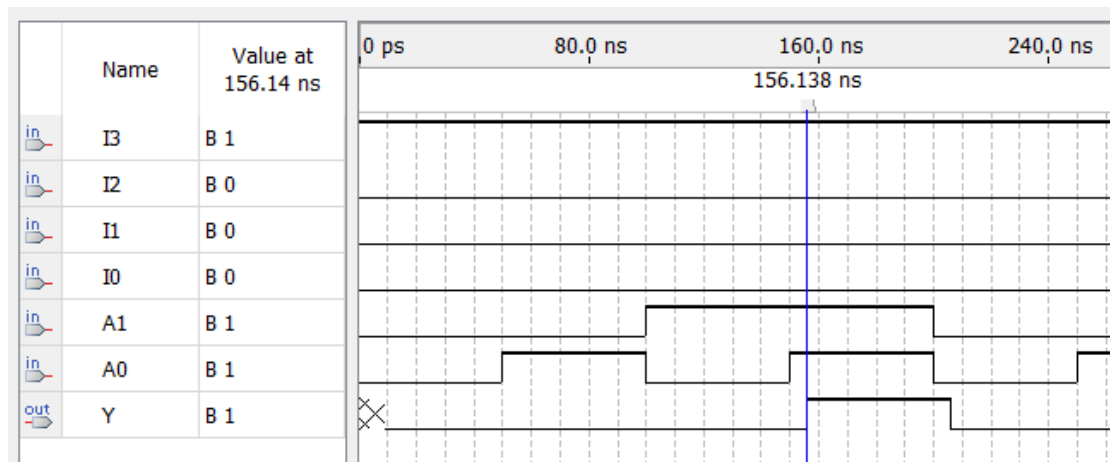
Το σχηματικό:



Η εξομοίωση:



Ελέγχουμε την χρονοκαθυστέρηση όταν το $A0 \rightarrow 1$ και το $Y \rightarrow 1$. Με τη διαφορά υπολογίζουμε ότι καθυστερεί 6.138ns .



	Input Port	Output Port	RR	RF	FR	FF
1	A0	Y	6.138	6.138	6.138	6.138
2	A1	Y	10.091	10.091	10.091	10.091
3	I0	Y	10.081	10.081	10.081	10.081
4	I1	Y	5.318			5.318
5	I2	Y	10.033	10.033	10.033	10.033
6	I3	Y	9.405			9.405

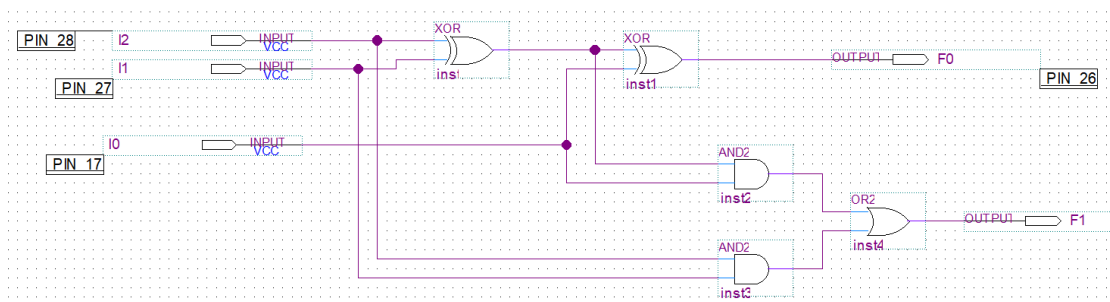
Στη στατική ανάλυση ελέγχουμε την 1^η γραμμή και τη στήλη RR και εντοπίζουμε ότι η καθυστέρηση είναι 6.138ns. Άρα το αποτέλεσμα είναι και στα δύο ίδιο.

Ερώτημα 3^ο

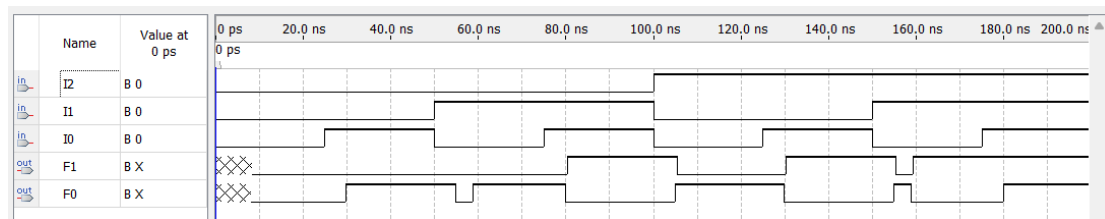
Στην κάθετη πράξη πρόσθεσης, για την πρόσθεση δύο ψηφίων εφαρμόζεται ο πίνακας αληθείας του XOR.

Το αποτέλεσμα του XOR είναι το άθροισμα (F0) και στην περίπτωση κρατούμενου (δηλαδή $1+1=0$, $C=1$) το κρατούμενο προστίθεται με XOR στην επόμενη πράξη μέχρι τη στιγμή που δεν θα υπάρχει κρατούμενο.

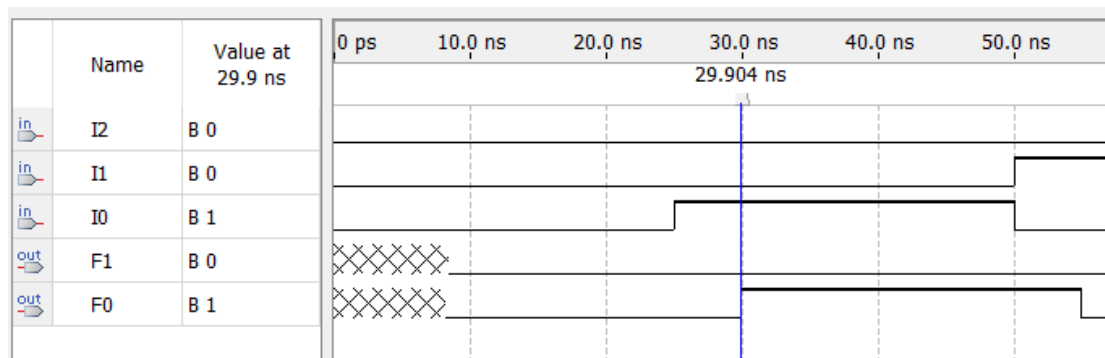
Το σχηματικό:



Η εξομοίωση:



Ελέγχουμε την χρονοκαθυστέρηση όταν το I0->1 και το F0->1. Με τη διαφορά υπολογίζουμε ότι καθυστερεί 4.904ns.



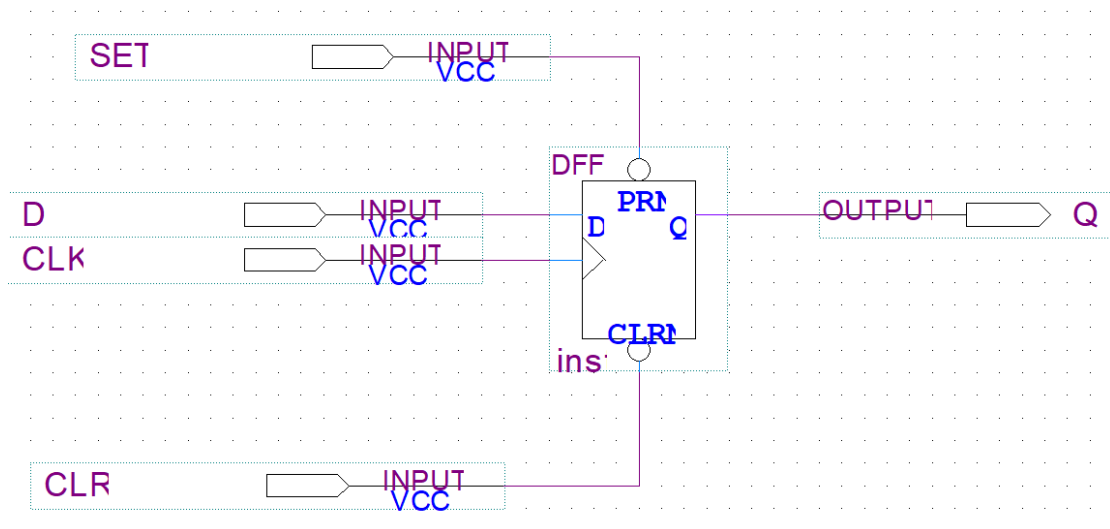
Propagation Delay						
	Input Port	Output Port	RR	RF	FR	FF
1	I0	F0	4.904	4.904	4.904	4.904
2	I0	F1	5.358			5.358
3	I1	F0	8.889	8.889	8.889	8.889
4	I1	F1	9.346			9.346
5	I2	F0	9.179	9.179	9.179	9.179
6	I2	F1	9.634			9.634

Στη στατική ανάλυση ελέγχουμε την 1^η γραμμή και τη στήλη RR και εντοπίζουμε ότι η καθυστέρηση είναι 4.904ns. Άρα το αποτέλεσμα είναι και στα δύο ίδιο.

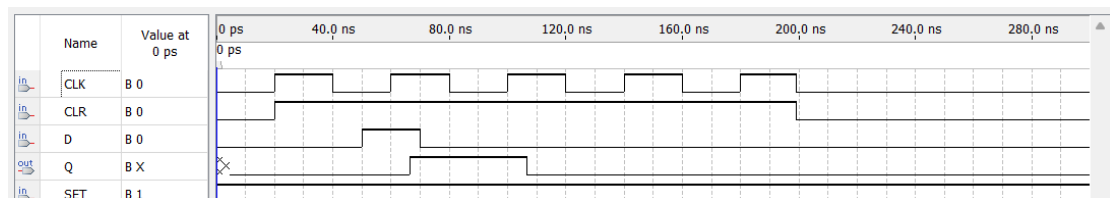
Βασικά ακολουθιακά κυκλώματα

Ερώτημα 1^ο

Το σχηματικό:



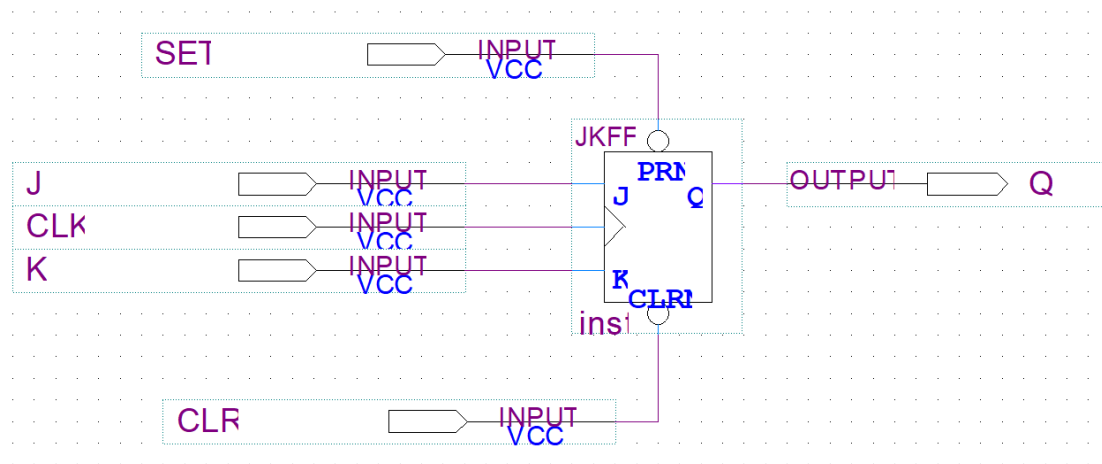
Η εξομίωση:



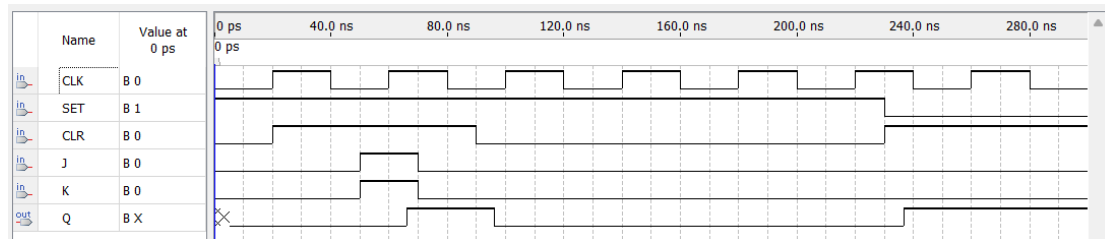
Ερώτημα 2^ο

JK Flip-Flop

Το σχηματικό:



Η εξομίωση:



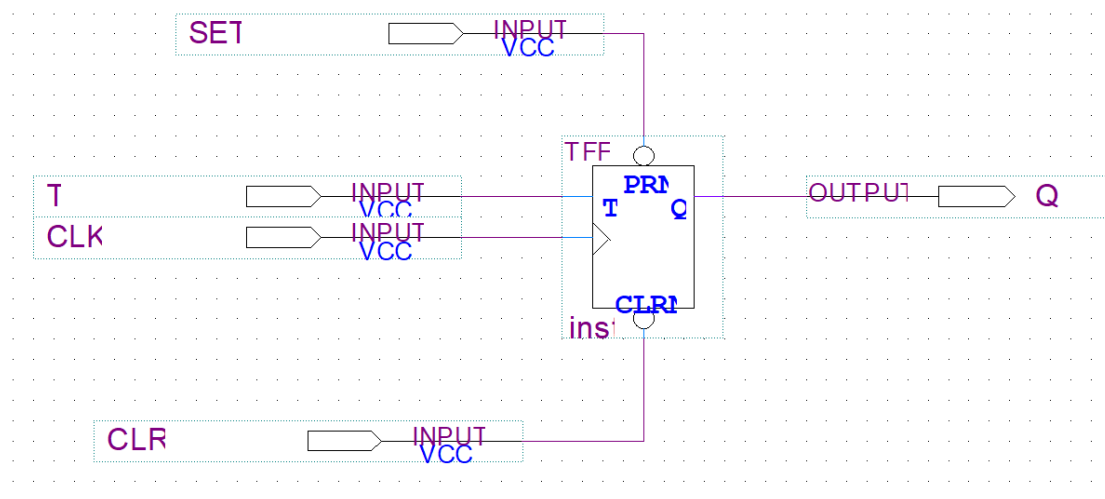
20-90ns: Οι ασύγχρονη θέση και μηδένιση είναι απενεργοποιημένες οπότε το JKFF παίρνει τα δεδομένα εισόδου.

90-220ns: Ενεργοποιήσαμε την ασύγχρονη μηδένιση και παρατηρούμε ότι τα δεδομένα που μας δίνει το JKFF μηδενίζονται.

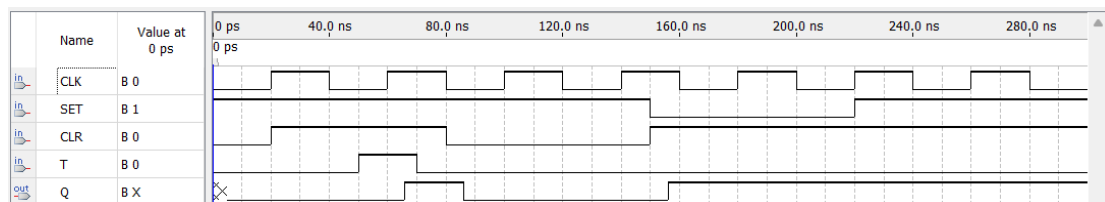
220ns-: Ενεργοποιήσαμε την ασύγχρονη θέση και απενεργοποιήσαμε την ασύγχρονη μηδένιση και έτσι θέσαμε τα δεδομένα στη μονάδα.

T Flip-Flop

Το σχηματικό:



Η εξομοίωση:



Παρατηρούμε ότι λειτουργεί όπως το JKFF.

Ερώτημα 3°

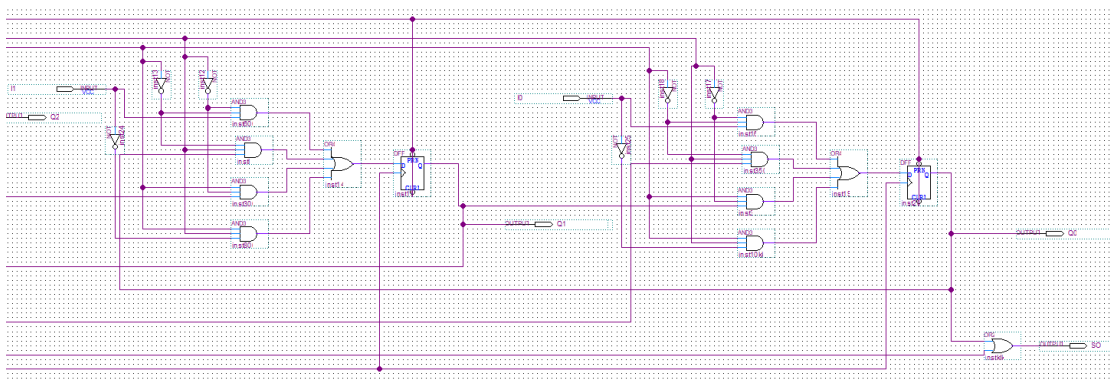
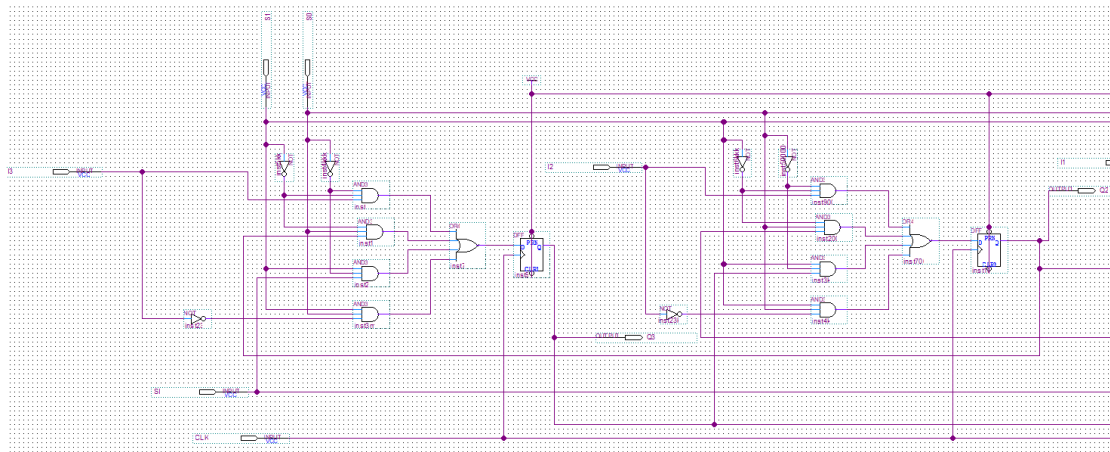
Στην πρώτη περίπτωση έχουμε παράλληλη φόρτωση, δηλαδή τα δεδομένα I θα περάσουν από το DFF και θα βγουν αυτούσια στην έξοδο Q. Άρα στο D0 κάθε πολυπλέκτη βάζουμε την είσοδο I_i (i=0,1,2,3).

Στη δεύτερη περίπτωση έχουμε αριστερή ολίσθηση, δηλαδή τα ψηφία μετακινούνται μια θέση από τα δεξιά προς τα αριστερά. Στο πιο δεξί πολυπλέκτη θα βάλουμε την είσοδο SI (serial input), η οποία θα προσθέτει ένα ψηφίο κάθε ακμή του ρολογιού. Το Q output του FF θα μεταφέρεται στο αμέσως επόμενο (αριστερό) πολυπλέκτη στην είσοδο για τη λειτουργία D1. Όταν φτάσουμε στο πιο αριστερό πολυπλέκτη, βάζουμε στην είσοδο D1 το προηγούμενο Q και την έξοδο Q αυτού του FF τη βάζουμε στο SO (serial output).

Στην τρίτη περίπτωση γίνεται η ίδια διαδικασία με την προηγούμενη, όμως από τα αριστερά προς τα δεξιά. Το πιο δεξιό ψηφίο θα πάει στο SO. Πριν καταλήξουμε στην έξοδο SO χρησιμοποιούμε μια πύλη OR2 για να συμπεριλάβουμε και το ψηφίο της δεξιάς ολίσθησης και το ψηφίο της αριστερής ολίσθησης (ανάλογα, βέβαια, με την λειτουργία που επιλέγεται από τα S1S0).

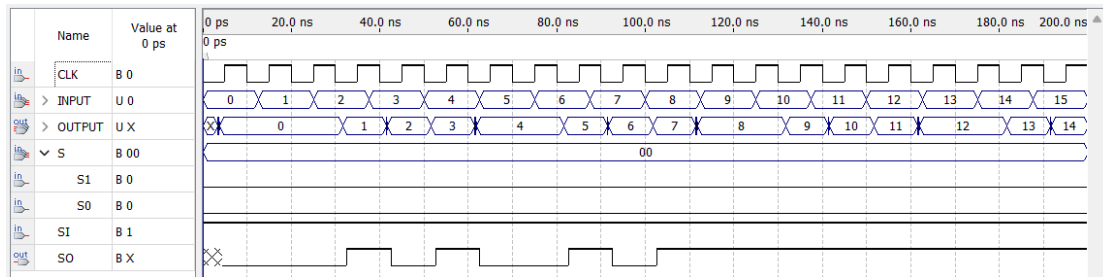
Στην τέταρτη περίπτωση γίνεται συμπλήρωση οπότε στην είσοδο D3 του κάθε πολυπλέκτη θα μπει το output του ίδιου FF, όμως αντεστραμμένο με μια πύλη NOT.

Το σχηματικό (στα δύο επόμενα screenshots):

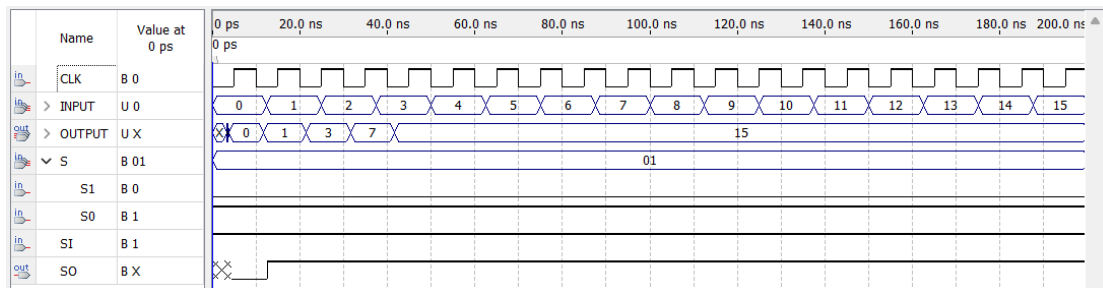


Η εξομοίωση:

S1S0=00 (Παράλληλη φόρτωση):

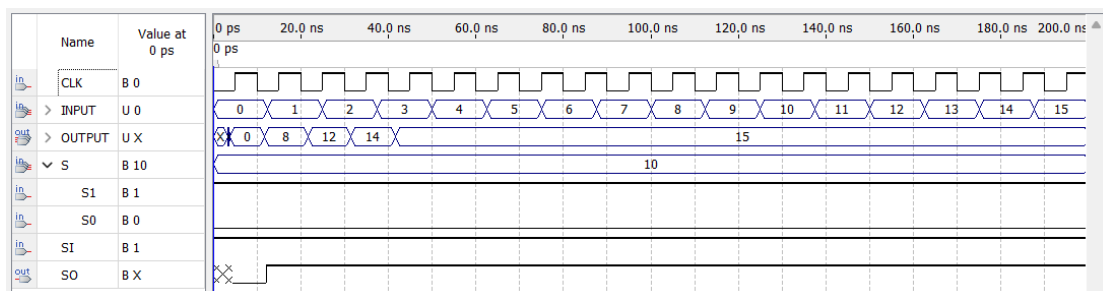


S1S0=01 (Αριστερή ολίσθηση):



Χρησιμοποιούμε μια είσοδο serial input (SI) για να προσθέτουμε σε κάθε ακμή του ρολογιού το ψηφίο 1 και μια έξοδο serial output (SO) για να παρακολουθούμε την έξοδο των τελευταίων ψηφίων της ολίσθησης. Αυτό το κάνουμε για να δείξουμε την μετακίνηση των ψηφίων κατά την ολίσθηση.

S1S0=10 (Δεξιά ολίσθηση):



S1S0=11 (Συμπλήρωση):

