

ΕΡΓΑΣΤΗΡΙΟ 6

ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ

kprousalis@csd.auth.gr
ntalaperas@csd.auth.gr
gkeramidas@csd.auth.gr

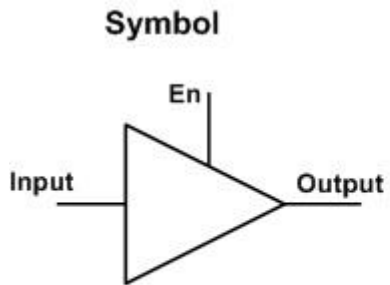


ΠΕΡΙΕΧΟΜΕΝΑ

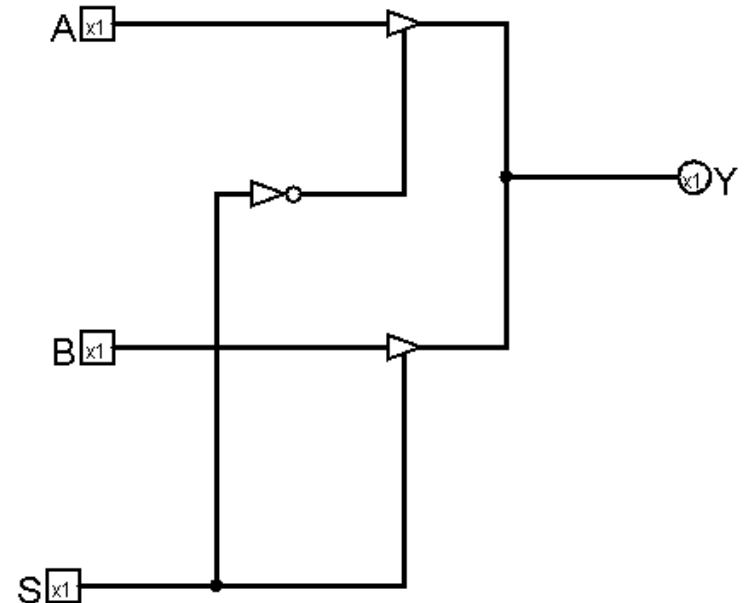
- Πύλες τριών καταστάσεων (3-state Gates)
- Λειτουργία RAM
- Παράδειγμα
- Ζητούμενα

Πύλες Τριών Καταστάσεων (3-state Gates) 2-1 MUX

Οι πύλες τριών καταστάσεων μπορούν να εξομοιώσουν τους πολυπλέκτες.

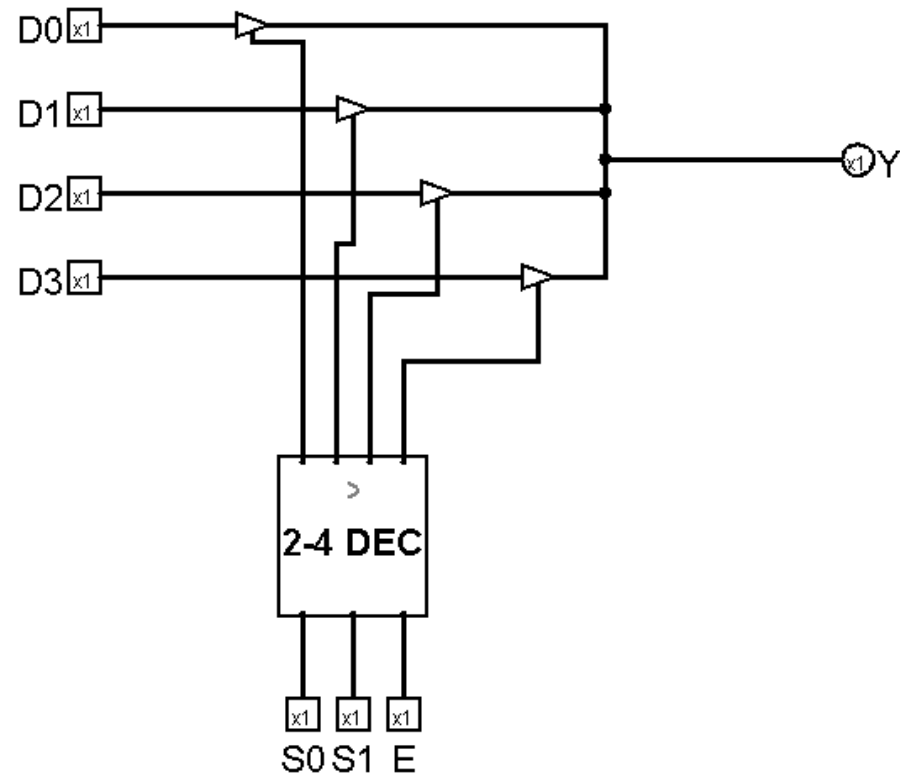


En	Input	Output
0	X	Hi-Z
1	0	0
1	1	1



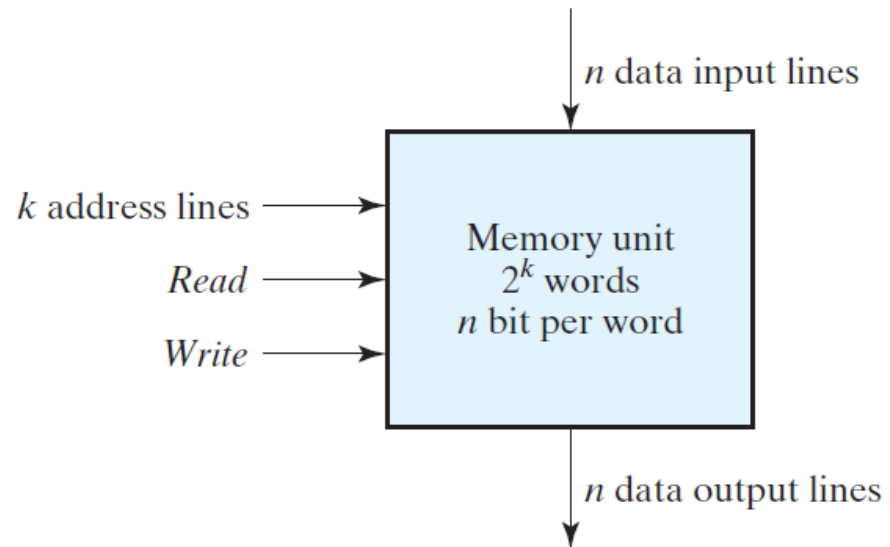
Πύλες Τριών Καταστάσεων (3-state Gates) 4-1 MUX

Πολυπλέκτης 4-1:
Οι πύλες εδώ
ελέγχονται από έναν 2-4
αποκωδικοποιητή.

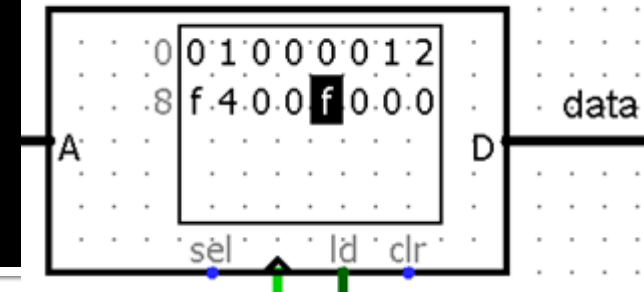


Μνήμη RAM

Διάγραμμα μονάδας μνήμης RAM



Η Μνήμη RAM στο Logisim



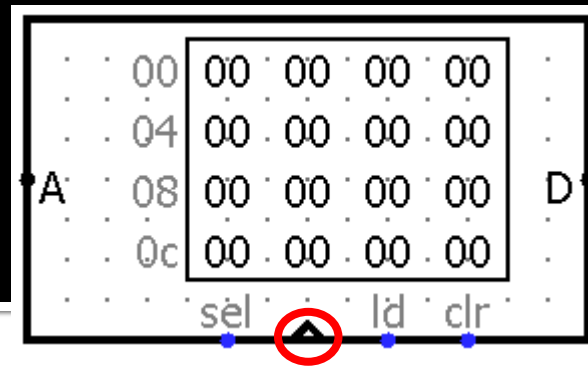
- Εύρος αποθήκευσης διευθύνσεων σε bit: 0-16,777,216 (24bit) (Address Bit Width)
- Σε κάθε διεύθυνση αντιστοιχεί χώρος: 32 bits (Data Bit Width)
- Μπορούμε να φορτώσουμε και να αποθηκεύσουμε τιμές
- Ο χρήστης μπορεί να τροποποιήσει τιμές διαδραστικά μέσω του Poke Tool, ή να τροποποιήσει όλα τα περιεχόμενα μέσω του Menu Tool
- Στο πλαίσιο φαίνονται οι τρέχουσες τιμές
- Αριστερά του πλαισίου φαίνονται οι διευθύνσεις με γκρι
- Η αναπαράσταση των τιμών είναι στο δεκαεξαδικό σύστημα. Οι επιλεγμένες "μαρκάρονται" ως άσπρο μέσα σε μαύρο.

Η Μνήμη RAM του Logisim

Υποστηρίζονται τρεις διαφορετικές διεπαφές:
(ρυθμίζεται από Object's Properties: Data Interface)

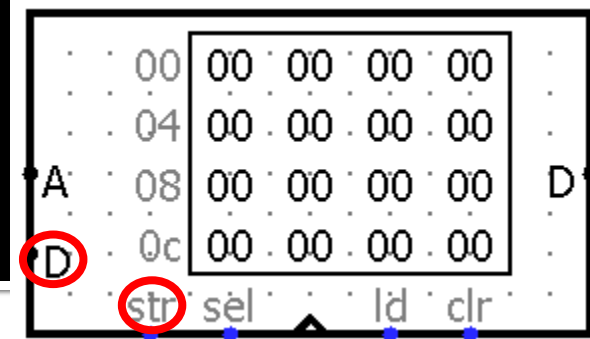
1. One synchronous load/store port (default)
2. One asynchronous load/store port
3. Separate load and store ports

One a/synchronous load/store port



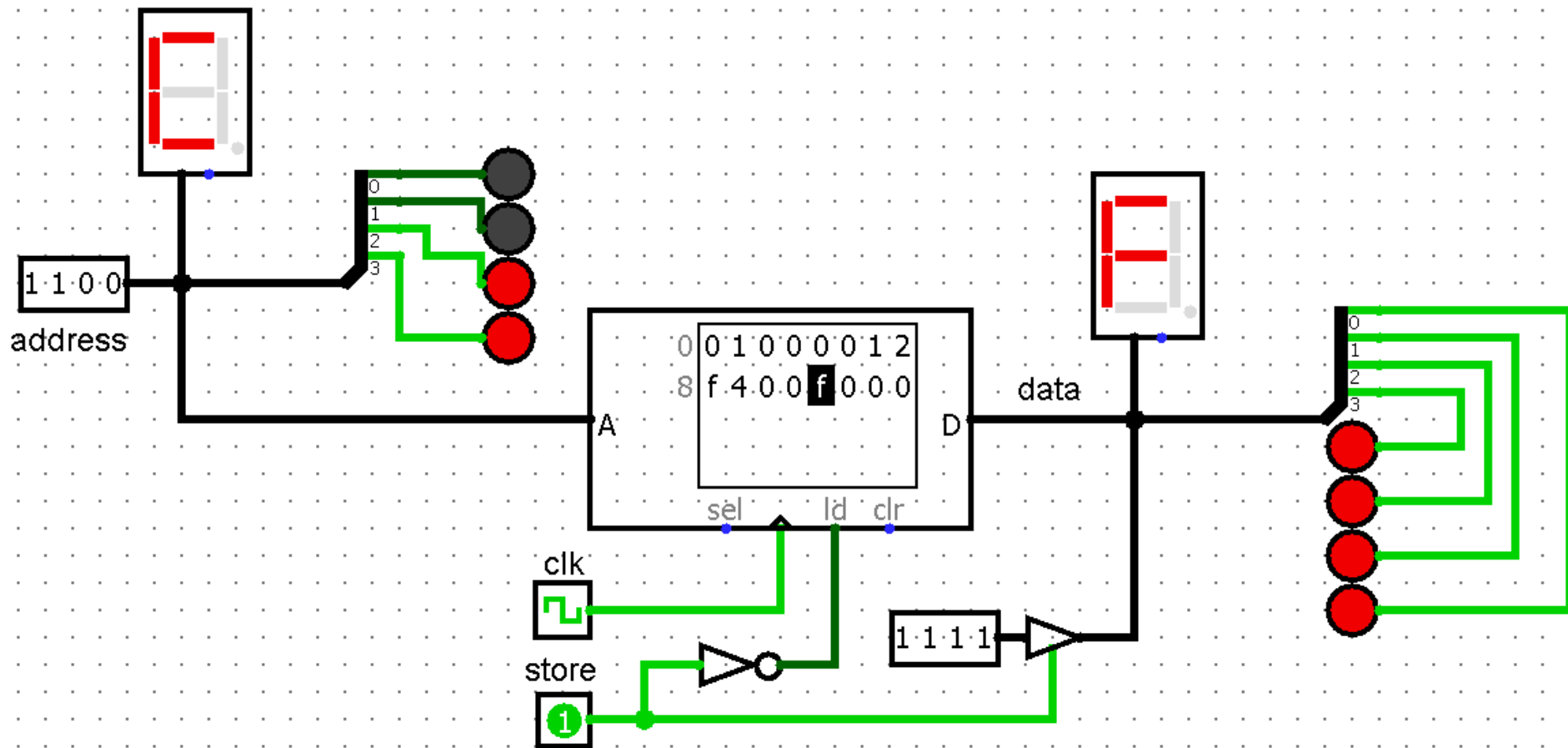
- *A* (input, συμβατό με Address Bit Width): Επιλέγει τη διεύθυνση που θα προσπελάσει το κύκλωμα.
- *D* *east* (input/output, συμβατό με Data Bit Width): Η τιμή του *D* αποθηκεύεται στην τρέχουσα θέση μνήμης *A*, εφόσον αυτό ζητηθεί.
- *ld* ή Load (input, bit width 1): Επιλέγει αν θα εξάγει στην πόρτα *D* την τιμή που υπάρχει στη τρέχουσα θέση *A* της μνήμης.
- *sel* ή Chip Select (input, bit width 1): Ενεργοποιεί και απενεργοποιεί το σχετικό άρθρωμα μνήμης.

Separate load and store ports

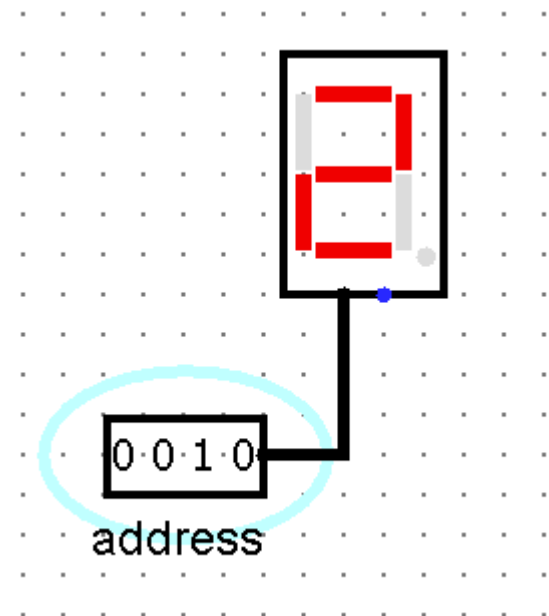
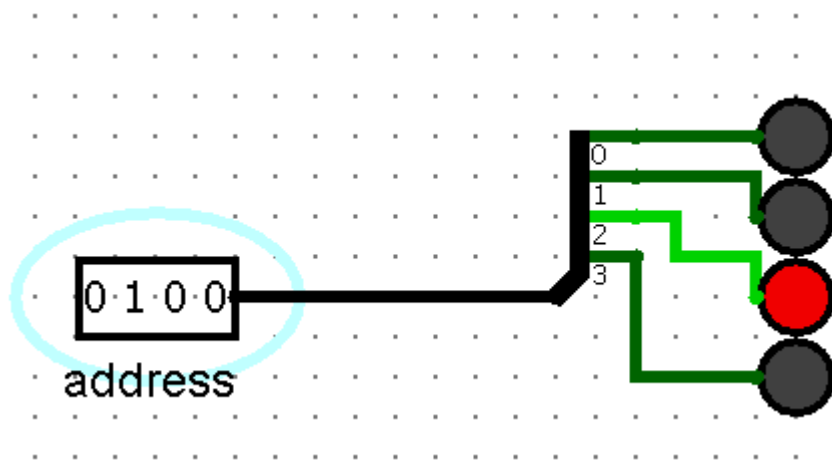


- **A** (input, συμβατό με Address Bit Width): Επιλέγει τη διεύθυνση που θα προσπελάσει το κύκλωμα.
- **D west** (input, συμβατό με Data Bit Width): Όταν απαιτείται store (μέσω της θετικής ακμής του ρολογιού, ενώ *sel* και *str* είναι 1 ή ασταθές), η τιμή αυτή αποθηκεύεται στη τρέχουσα διεύθυνση μνήμης A.
- **D east** (output, συμβατό με Data Bit Width): Όταν *sel* και *ld* είναι 1 ή ασταθής, τότε ανακτάται στην έξοδο αυτή η τιμή της τρέχουσας διεύθυνσης.
- **str** (input, bit width 1): Store: Μόνο όταν έχουμε mode: separate load and store ports: Όταν 1 ή ασταθές, ο παλμός θα επιτρέψει την αποθήκευση της τιμής του Δυτικού D, εφόσον το *sel* είναι 1 ή ασταθές.
- **ld** ή Load (input, bit width 1): Επιλέγει αν θα εξάγει στην πόρτα D την τιμή που υπάρχει στη τρέχουσα θέση A της μνήμης.
- **sel** ή Chip Select (input, bit width 1): Ενεργοποιεί και απενεργοποιεί ολόκληρο το άρθρωμα μνήμης.

Παράδειγμα



Splitter και Hex Digit Display

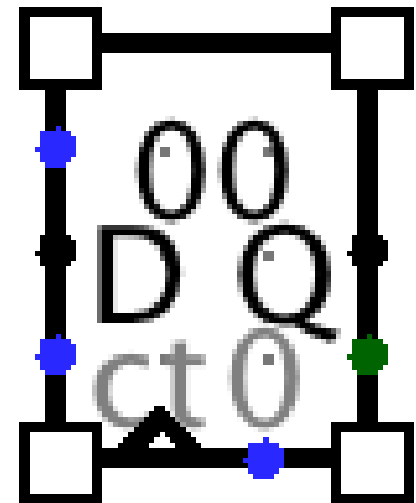


Σχεδιάστε

- Τροποποιήστε κατάλληλα το προηγούμενο κύκλωμα ώστε η μνήμη να έχει μέγεθος 32 (θέσεις) x 5 bits

Παρουσίαση Μετρητή με Clk

- D: Δεδομένα που θα φορτωθούν στο μετρητή
- Load: Αν 1 και count = 0, ο μετρητής φορτώνει δεδομένα από το D
- Counter: Αν 1 και load = 0, ο μετρητής αυξάνει τις τιμές. Αν load = 1, ο μετρητής μειώνει τις τιμές
- Carry: 1 όταν ο μετρητής φτάνει τη μέγιστη τιμή (ή την ελάχιστη όταν μειώνει τις τιμές).



Memory Fill & Memory Test

Να υλοποιήσετε τις ακόλουθες λειτουργίες:

- **Memory fill** → αποθήκευση διαδοχικών αριθμών σε διαδοχικές θέσεις της μνήμης π.χ. ο αριθμός 0 στην θέση μνήμης 0, ο αριθμός 1 στην θέση μνήμης 1 κτλ.
- **Memory test** → διάβασμα διαδοχικά (μία-μία θέση μνήμης) και έλεγχος αν η αποθηκευμένη τιμή είναι η επιθυμητή.

Ασκήσεις για υποβολή

6^ο Εργαστήριο

Παραδοτέα 2 αρχεία:
1. αρχείο αναφοράς .pdf
2. αρχείο .circ

Να σχεδιάσετε σύστημα για τον έλεγχο μια μνήμης με μέγεθος 32 (θέσεις) x 5 bits χρησιμοποιώντας τον αυτοματοποιημένο τρόπο παραγωγής clk του logisim

- Για την παραγωγή των addresses θα πρέπει να χρησιμοποιηθεί 5bit σύγχρονο, προσθετικό δυαδικό μετρητή με D ff
- Για την παραγωγή των data θα πρέπει να χρησιμοποιηθεί 5bit ασύγχρονο, αφαιρετικό, δυαδικό μετρητή με JK ff
- Με ένα σήμα εισόδου (με όνομα mode) θα πρέπει να ελέγχεται αν το σύστημα κάνει memory fill ή memory test.

Υποχρεωτική υποβολή με αξιολόγηση

ΤΕΛΟΣ

6^{ου} εργαστηρίου

Ερωτήσεις;

Εργασία Εξαμήνου

- Εργασία Εξαμήνου!!!!

Διαδικαστικά (1/2) ...

1. Η διεκπεραίωση της εργασίας εξαμήνου είναι προϋπόθεση απαραίτητη για τη συμμετοχή σας στην τελική εξέταση.
2. Οι φοιτητές παλαιότερων ετών που παρακολούθησαν ολοκληρωμένα το εργαστήριο αλλά **δεν είχαν διεκπεραιώσει την εργασία** εξαμήνου, θα πρέπει να την περατώσουν προκειμένου να λάβουν το δικαίωμα συμμετοχής στην τελική εξέταση.
3. Οι φοιτητές παλαιότερων ετών που παρακολούθησαν ολοκληρωμένα το εργαστήριο και είχαν παραδώσει την εργασία εξαμήνου **δε χρειάζεται να την εκπονήσουν ξανά!**
4. Ο φοιτητής που δε θα εκπονήσει την εργασία στο τρέχον εξάμηνο, δε θα είναι υποχρεωμένος να παρακολουθήσει ξανά το εργαστήριο το επόμενο έτος.
5. Η εκφώνηση της εργασίας μαζί με σχετικές πληροφορίες υπάρχει στη σελίδα του μαθήματος (e-learning) και η υλοποίησή της θα πρέπει να γίνεται ατομικά από τον καθένα.

Διαδικαστικά (2/2) ...

1. Υπενθύμιση: οι εργαστηριακές ασκήσεις και η εργασία εξαμήνου συμβάλουν μαζί στο 30% του τελικού βαθμού, ενώ το υπόλοιπο 70% διαμορφώνεται από την τελική εξέταση.
2. Η εργασία εξαμήνου και οι εργαστηριακές ασκήσεις συμβάλουν ισομερώς, δηλαδή: 15% οι ασκήσεις και 15% η εργασία εξαμήνου.
3. Θα εξετασθούν προφορικά οι ομάδες για τις οποίες υπάρχουν υπόνοιες αντιγραφής. Μηδενίζονται οι εργασίες που αποδεδειγμένα είναι αντιγραφές.
4. Δεν ισχύει η βάση του 5 στην εργασία αλλά ισχύει στην τελική εξέταση.
5. Η εργασία εξαμήνου και οι εργαστηριακές ασκήσεις συμβάλουν ισομερώς, δηλαδή: 15% οι ασκήσεις και 15% η εργασία εξαμήνου.

Τελική εξέταση: 70%

Εργασία εξαμήνου: 15%

Εργαστηριακές ασκήσεις: 15%

Παραδοτέα Εργασίας Εξαμήνου

- Η προθεσμία για την εκπόνηση της εργασίας **ξεκινά από τις 12/05 και λήγει στις 20/06.**
- Παραδοτέα:
 - Δύο αρχεία .circ και δύο αρχεία εικόνας των κυκλωμάτων (.png, .jpeg, κ.τ.λ.)
 - Ένα αρχείο κειμένου (pdf κατά προτίμηση) ως αναφορά, όπου θα αναπτύσσετε μια σύντομη και συνοπτική περιγραφή περίπου 5 σειρών για τη λειτουργία του κάθε κυκλώματος.
- Στην αναφορά θα πρέπει να αναφέρονται τα προσωπικά σας στοιχεία: ονοματεπώνυμο, ΑΕΜ, email, έτος.

Υποχρεωτική Εργασία Εξαμήνου

Ζήτημα Α΄

Να σχεδιάσετε και να υλοποιήσετε μια αριθμητική-λογική μονάδα (ALU) η οποία να δέχεται εισόδους A, B των 2-bit και να εκτελεί τις ακόλουθες πράξεις: NOT_A, ADD, XOR, NAND και A+1. Η έξοδος είναι 2-bits. Η πύλη NOT να χρησιμοποιείται μόνο για την είσοδο A. Εκτός από τις εισόδους A, B, η ALU θα δέχεται ως είσοδο ένα αριθμό από σήματα επιλογής (selection bits) τα οποία θα προσδιορίζουν την πράξη που θα εκτελείται. Τα σήματα επιλογής θα χρησιμοποιηθούν ως σήματα επιλογής στους πολυπλέκτες του κυκλώματος. Ένα επιπλέον σήμα εισόδου στην ALU θα αποτελεί το αρχικό κρατούμενο του αθροιστή. Ο αριθμός των σημάτων επιλογής θα πρέπει να προσδιοριστεί από εσάς.

Υποχρεωτική Εργασία Εξαμήνου

Ζήτημα Β'

Να υλοποιήσετε σύγχρονο προσθετικό (count-up) μη-δυναμικό μετρητή με negative edge-triggered JK-FF και κύκλο απαρίθμησης 9 (mod-9), δηλαδή 0, 1, ..., 7, 8, 0, 1, ...