

Εργαστήριο 9: Τρικατάστατοι Οδηγητές, Λεωφόροι, Μνήμες SRAM

28 - 30 Νοεμβρίου 2018

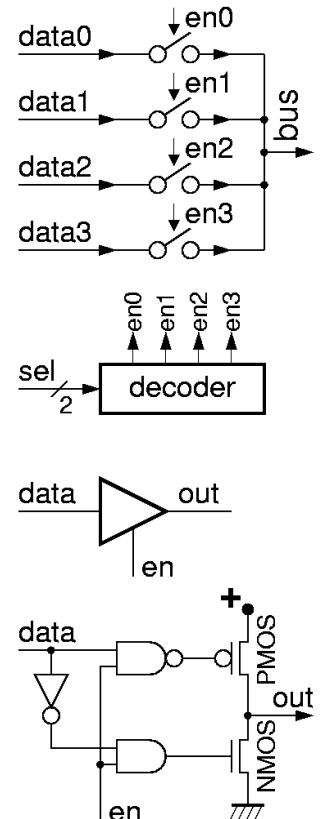
[Βιβλία: Προαιρετικά μπορείτε να διαβάσετε: Dally: §4.3.4 (σελ. 80-82), §8.3 (σελ. 170-175), §8.9 (σελ. 191-193), §24.2 (σελ. 535-537), και §25.2 (σελ. 551-553); Mano (5η έκδοση): §7.2, 7.3 (σελ. 302-311, εκτός HDL σελ. 305-306), και αν θέλετε και §9.14 (σελ. 471-473); Wakerly: από §10.2 έως και 10.3.4 (σελ. 1020-1030)].

9.1 Πολύπλεξη μέσω Τρικατάστατων Οδηγητών:

Στην §1.5 είδαμε ότι οι πολυπλέκτες αποτελούν βασικότατο δομικό λίθο των ψηφιακών συστημάτων. Στην §4.9 είδαμε πώς φτιάχνεται ένας πολυπλέκτης από πύλες, ενώ στο πείραμα 3.1 είχαμε δει έναν πολυπλέκτη φτιαγμένο με ηλεκτρονόμους. Στο σχήμα δεξιά φαίνεται μία άλλη υλοποίηση πολυπλέκτη με διακόπτες· η διαφορά έχει ως εξής. Γιά έναν πολυπλέκτη n -σε-1, το κύκλωμα του πειράματος 2.1 δέχεται $\log n$ εισόδους επιλογής, τις οποίες και αποκωδικοποιεί στους n συνδυασμούς τους προκειμένου να αποκαταστήσει έναν αγωγίμο δρόμο μεταξύ της εξόδου και μίας και μόνο εισόδου. Γιά έναν ίδιο πολυπλέκτη n -σε-1, το κύκλωμα δεξιά δέχεται n εισόδους επιλογής, $en_0, \dots, en_{(n-1)}$, οι οποίες πρέπει να είναι **ήδη** αποκωδικοποιημένες, δηλαδή ακριβώς μία από αυτές πρέπει να είναι ενεργή (1) και οι υπόλοιπες αδρανείς (0).

Όταν οι πηγές πληροφοριών, $data_0, data_1$, κλπ, βρίσκονται σχετικά μακριά ή μία από την άλλη, ή όταν είναι επιθυμητή η προσθαφαίρεση πηγών κατά τη λειτουργία του συστήματος (π.χ. προσθήκη νέας κάρτας σε υπολογιστή), τότε η υλοποίηση του πολυπλέκτη μέσω διακοπών είναι προτιμότερη της υλοποίησης μέσω πυλών. Η πρώτη αιτία είναι το πλήθος και το κόστος των απαιτούμενων συρμάτων: το κύκλωμα με πύλη OR απαιτεί να συλλεγούν οι πληροφορίες από όλες τις πηγές σε ένα κεντρικό σημείο, πράγμα που χρειάζεται τόσα σύρματα όσες και οι πηγές· αντίθετα, το κύκλωμα με διακόπτες λειτουργεί με ένα μόνο σύρμα που διατρέχει το σύστημα απ' άκρη σ' άκρη μεταφέροντας κάθε φορά την πληροφορία της όποιας πηγής μας ενδιαφέρει τη στιγμή εκείνη. (Ο αποκωδικοποιητής και τα σύρματα επιλογής είναι ένα επιπλέον κόστος, αλλά όταν υπάρχουν πολλαπλά σύρματα (bits) δεδομένων το κόστος της επιλογής αποσβέννεται μεταξύ όλων αυτών· επίσης, υπάρχει πλήθος εφαρμογών όπου η επιλογή γίνεται με άλλους, καταναμημένους τρόπους). Το δεύτερο πλεονέκτημα της πολύπλεξης μέσω διακοπών είναι η ευκολία προσθήκης νέων πηγών πληροφορίας: εάν κάθε πηγή έχει ενσωματωμένο και το δικό της διακόπτη εξόδου, τότε η προσθήκη αυτή γίνεται μέσω απλής σύνδεσης συρμάτων --π.χ. με την εισαγωγή μίας κάρτας σε μίαν υποδοχή-πρίζα.

Η παραπάνω λογική συμπεριφορά της πηγής πληροφορίας με τον διακόπτη στην έξοδο υλοποιείται στην τεχνολογία CMOS με το κύκλωμα που φαίνεται στο κάτω μέρος του σχήματος. Το κύκλωμα αυτό μπορεί να φέρει την έξοδο out σε μία από τρεις διαφορετικές καταστάσεις: μηδέν (0), ένα (1), ή "**υψηλή αντίσταση**" (**HiZ** - high impedance - υψηλή εμπέδηση). Λόγω των τριών καταστάσεων της εξόδου του, το κύκλωμα αυτό ονομάζεται **τρικατάστατος οδηγητής (tri-state driver)**. Η κατάσταση 0 αντιστοιχεί σε ενεργοποιημένο διακόπτη ($en=1$) και σε πληροφορία 0, οπότε η έξοδος πρέπει να οδηγηθεί σε χαμηλή τάση, ανάβοντας το transistor NMOS. Η κατάσταση 1 αντιστοιχεί σε ενεργοποιημένο διακόπτη ($en=1$) και σε πληροφορία 1, οπότε η έξοδος πρέπει να οδηγηθεί σε ψηλή τάση, ανάβοντας το transistor PMOS. Η κατάσταση HiZ αντιστοιχεί σε αδρανή (σβηστό) διακόπτη ($en=0$), οπότε η έξοδος πρέπει να μείνει ασύνδετη -



-να μην τοποθετηθεί καμία πληροφορία σε αυτήν, αφήνοντας άλλες πηγές να την οδηγήσουν· σε αυτή την περίπτωση πρέπει και τα δύο transistors, NMOS και PMOS, να παραμείνουν σβηστά. Οι πύλες NAND, NOT, και AND που φαίνονται στο σχήμα προφανώς επιτυγχάνουν ακριβώς τη λειτουργία αυτή· επίσης εξασφαλίζουν ότι σε καμία περίπτωση δεν ανάβουν ταυτόχρονα και τα δύο transistors, NMOS και PMOS, αφού κάτι τέτοιο και απροσδιόριστη τάση εξόδου θα έδινε, και υπερβολικό ρεύμα τροφοδοσίας θα ξόδευε, και υπερθέρμανση των transistors θα προκαλούσε.

Ένα σύρμα στο οποίο συνδέονται πολλαπλοί τρικατάστατοι οδηγητές προκειμένου να υλοποιηθεί πολύπλεξη των αντίστοιχων δεδομένων εισόδου ονομάζεται "λεωφόρος" (**bus**), ή "αρτηρία" ή "διάδρομος" (ή καμιά φορά "δίαυλος"). Ο κανόνας λειτουργίας μιάς λεωφόρου είναι ότι, ανά πάσα στιγμή, **το πολύ ένας** από τους οδηγητές που συνδέονται σε αυτήν επιτρέπεται να είναι ενεργοποιημένος· ο οδηγητής αυτός προσδιορίζει την τάση (λογική τιμή) της λεωφόρου. Όποτε δεν υπάρχει κανείς ενεργοποιημένος οδηγητής, η λογική τιμή της λεωφόρου είναι *απροσδιόριστη*, δηλαδή η πληροφορία πάνω της είναι "σκουπίδια" ή "θόρυβος" (σε ειδικές περιπτώσεις τεχνολογιών μπορεί να είναι προβλέψιμη η τιμή αυτή, όπως συμβαίνει με την "δυναμική αποθήκευση πληροφορίας" (δυναμική μνήμη) μέσα σε chips τεχνολογίας CMOS, όμως αυτά δεν αφορούν το παρόν μάθημα). Μία υδραυλική αναλογία της λεωφόρου είναι ένας σωλήνας που διατρέχει πολλά διαμερίσματα, και τον οποίο μπορούν να τροφοδοτούν βρύσες από κάθε διαμέρισμα. Ένας τρικατάστατος οδηγητής είναι μιά από αυτές τις βρύσες, αλλά με χωριστό ρομπινέτο ζεστού και κρύου νερού. Για να μεταδώσουμε πληροφορία "0" ανοίγουμε το κρύο ρομπινέτο· για να μεταδώσουμε "1" ανοίγουμε το ζεστό· για να μη μεταδώσουμε τίποτα, αφήνοντας κάποιον άλλον να μεταδώσει, κλείνουμε εντελώς και τα δύο ρομπινέτα μας. Εάν μεταδίδω ταυτόχρονα κι εγώ (π.χ. "1") και κάποιος άλλος (π.χ. "0"), τότε ο σωλήνας θα έχει "χλιαρό" νερό, που δεν είναι ούτε 0 ούτε 1· όποτε δεν μεταδίδει κανείς πληροφορία, η θερμοκρασία του σωλήνα είναι τυχαία, εξαρτόμενη από άλλους παράγοντες (π.χ. χειμώνας/καλοκαίρι).

Η "αρχιτεκτονική λεωφόρου" (bus) αποτελεί την απλούστερη μορφή δικτύου επικοινωνίας (communication network), και χρησιμοποιείται ευρύτατα όποτε η επικοινωνία γίνεται με ένα "κοινόχρηστο" μέσο (shared medium). Τέτοιο κοινόχρηστο μέσο μπορεί να είναι ένα σύρμα (όπως στην "κλασσική" μορφή του δικτύου "Ethernet" --όχι "switched ethernet"), μία ραδιοσυχνότητα σε δεδομένη περιοχή (π.χ. τα ραδιοταξί μιάς πόλης ή συνοικίας, ή μία από τις ραδιοσυχνότητες της κινητής τηλεφωνίας), ή παραδοσιακότερα μέσα όπως τα ηχητικά κύματα σε μιά αιθουσα συνεδρίασης.

Σε όλα αυτά, υπάρχουν ειδικοί κανόνες ("πρωτόκολλο επικοινωνίας") για το ποιός και πότε μπορεί να "πάρει το λόγο", δηλαδή με ποιόν τρόπο θα εξασφαλιστεί η τήρηση του κανόνα ότι το πολύ ένας οδηγητής (ομιλητής) μπορεί να είναι ενεργός κάθε φορά. Το κλασσικό Ethernet ακολουθεί ένα πολύ ενδιαφέρον πρωτόκολλο "κατανεμημένης διαιτησίας" για το ποιός και πότε μιλάει, το οποίο δεν απαιτεί πρόσθετο (χωριστό) μέσο μετάδοσης (σύρμα): (i) Ορίζεται μέγιστο μέγεθος πακέτου (1500 Bytes)· (ii) Καθένας που θέλει να μιλήσει περιμένει πρώτα να τελειώσει αυτός που τώρα μιλάει (εάν κάποιος μιλάει), και αμέσως μετά αρχίζει να μιλάει· (iii) Τη στιγμή που ο οιοσδήποτε αρχίζει να μιλάει, ακούει ταυτόχρονα τι υπάρχει πάνω στην αρτηρία (λεωφόρο): εάν αυτό που ακούει δεν συμπίπτει με αυτό που ο ίδιος μεταδίδει, σημαίνει ότι συνέβη "σύγκρουση" --υπήρχε και κάποιος άλλος που άρχισε να μιλάει την ίδια στιγμή.... (iv) Μόλις διαπιστωθεί ότι υπάρχει σύγκρουση (την διαπιστώνουν πάντα όλοι οι ομιλούντες), διακόπτουν όλοι οι ομιλούντες τη μετάδοσή τους, περιμένουν ένα τυχαίο χρονικό διάστημα καθένας, και μετά προσπαθούν να ξαναμεταδώσουν, εάν εν τω μεταξύ η αρτηρία είναι ελεύθερη: η πιθανότητα να ξαναγίνει σύγκρουση των ιδίων είναι μικρή, λόγω των τυχαίων διαστημάτων αναμονής, που συνήθως θα είναι διαφορετικά για τον καθένα από τους συγκρουσθέντες.

9.2 Πολύμπιτες Τρικατάστατες Λεωφόροι (Buses):

Λεωφόροι (buses) του ενός bit (ένα σύρμα) είναι κοινές όταν η απόσταση επικοινωνίας είναι σημαντική. Για μικρότερες αποστάσεις, π.χ. μέσα σε ένα κουτί ή πάνω σε μία πλακέτα ή μέσα σε ένα chip, συνήθως χρησιμοποιούμε λεωφόρους πολλαπλών bits, δηλαδή λεωφόρους που μεταφέρουν, κάθε φορά, μιάν ολόκληρη λέξη πληροφορίας αποτελούμενη από πολλά bits· όλα τα bits της λέξης αυτής προέρχονται από την ίδια πηγή. Το σχήμα (φανταστείτε το σαν τριδιάστατο), δείχνει ένα παράδειγμα τριών πηγών (τρεις πλακέτες), καθεμία από τις οποίες έχει μιά τετράμπιτη λέξη πληροφορίας να στείλει. Στο επάνω μέρος του σχήματος, η λέξη εξόδου,

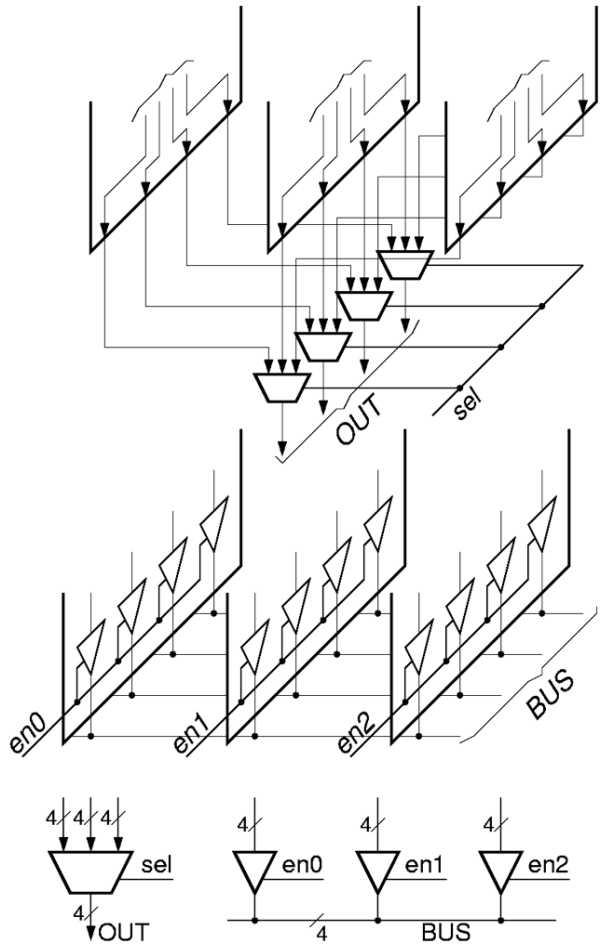
OUT, προκύπτει από έναν τετράμπιτο πολυπλέκτη 3-σε-1· αυτός αποτελείται, όπως είδαμε στην §8.1, από τέσσερις βασικούς (μονόμπιτους) πολυπλέκτες 3-σε-1, όπου όλοι τους έχουν κοινά σήματα επιλογής, sel.

Στη μέση του σχήματος φαίνεται η υλοποίηση του ίδιου συστήματος με τρικατάστατους οδηγητές· η λέξη εξόδου εμφανίζεται πάνω στην τετράμπιτη λεωφόρο BUS. Όλοι οι τρικατάστατοι οδηγητές της κάθε μιάς πηγής ελέγχονται (ενεργοποιούνται) από ένα κοινό σήμα ελέγχου, δηλαδή ανάβουν ή σβήνουν όλοι μαζί: όταν μιλάει μιά πηγή (πλακέτα)μιλάνε όλα της τα bits μαζί, το καθένα στο δικό του σύρμα του (πολύμπιτου) BUS, και κάθε φορά μιλάει μία και μόνο μία πηγή (πλακέτα). Έτσι, υπάρχουν τόσα σήματα ελέγχου, en0, en1, en2, όσες και οι πηγές (πλακέτες), και το πολύ ένα από αυτά μπορεί να είναι αναμένο ανά πάσα στιγμή.

Αυτός ο τρόπος σύνδεσης και επικοινωνίας "θυγατρικών πλακετών" (daughter boards) (οι 3 κατακόρυφες πλακέτες στο σχήμα) πάνω σε μία "μητρική πλακέτα" (mother board), ήταν πολύ συνηθισμένος στο πρόσφατο παρελθόν, και παραμένει ακόμα δημοφιλής στα συστήματα εκείνα που επιδιώκουν χαμηλό κόστος και όχι τις υψηλότερες δυνατές επιδόσεις. Οι θυγατρικές πλακέτες συχνά "κουμπώνουν" (μπαίνουν) σε ειδικές υποδοχές (sockets) (σαν πρίζες με πολλές επαφές) που υπάρχουν πάνω στη μητρική πλακέτα. Η μητρική πλακέτα έχει τη λεωφόρο (bus) --σύρματα δηλαδή-- και τις υποδοχές (sockets) --και συχνά και τον κεντρικό επεξεργαστή. Τις θυγατρικές πλακέτες μπορεί συχνά ο χρήστης να τις επιλέξει ανάλογα με τις ανάγκες του μεταξύ πολλαπλών επιλογών, και να τις αγοράσει χωριστά από την μητρική πλακέτα, σε κατοπινό χρόνο ή/και από άλλον κατασκευαστή, και να τις συνδέσει στο σύστημα ή να τις εναλλάξει όποτε το επιθυμεί. Τέτοια δημοφιλής λεωφόρος "περιφερειακών συσκευών" προσωπικών υπολογιστών ήταν π.χ. τα παλαιότερα μοντέλα "PCI" --μέχρι και το PCI-X.

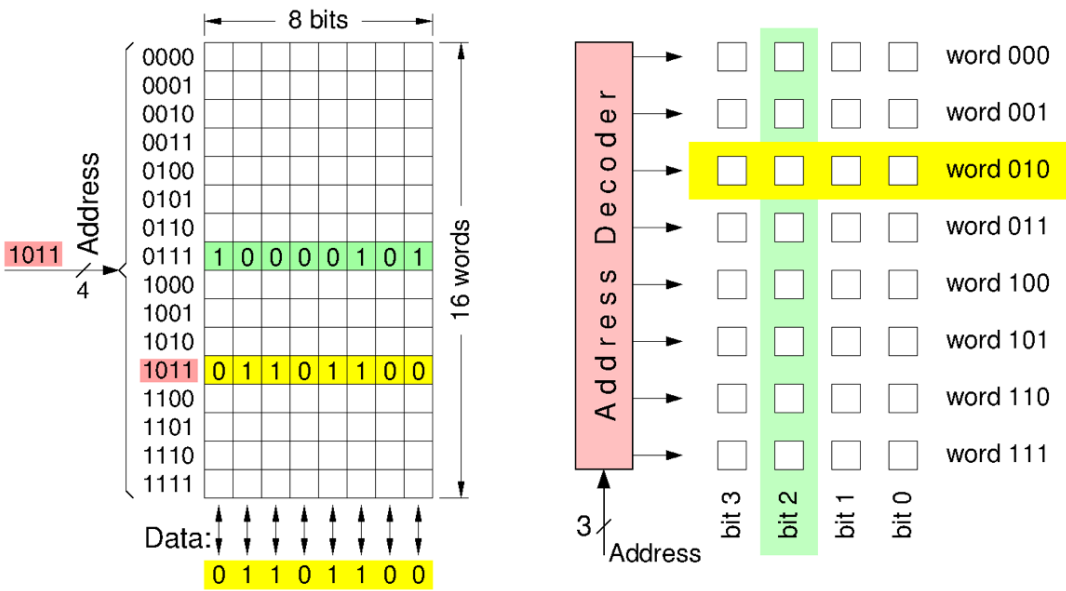
Στο κάτω μέρος του σχήματος φαίνονται τα συνεπτυγμένα σύμβολα των δύο υλοποιήσεων: κάθε τετράδα συρμάτων που μεταφέρει τα 4 bits της ίδιας λέξης παριστάνεται με ένα σύρμα και μία πλάγια γραμμή που υποδεικνύει το πλάτος της λέξης· το "πλάτος" του πολυπλέκτη ή των τρικατάστατων οδηγητών προκύπτει εμμέσως από το πλάτος των λέξεων που αυτοί χειρίζονται. Τα σήματα ελέγχου, en0, en1, en2, μπορεί π.χ. να τα δίνει ένας κεντρικός "Διαιτητής" (Arbiter), στον οποίο η κάθε πηγή που θέλει να μιλήσει στέλνει ένα σήμα "αίτησης" (request), και αυτός απαντά σε μία από τις αιτούσες πηγές, κάθε φορά, με ένα σήμα "grant" (enable).

[Προαιρετική, εγκυκλοπαιδική παράγραφος, εκτός ύλης μαθήματος:] Από την αρχή του 21ου αιώνα, οι λεωφόροι (bus) έχουν αρχίσει σταδιακά να χάνουν έδαφος, προς όφελος συνδέσμων μοναδικού οδηγητή και μοναδικού ακροατή (point-to-point links). Αυτό οφείλεται σε διάφορα ηλεκτρικά προβλήματα, καθώς και στη χαμηλή απόδοση των πρωτοκόλων, που εμφανίζονται όταν η συχνότητα λειτουργίας της λεωφόρου γίνεται πολύ ψηλή σε σχέση με το μήκος της λεωφόρου (η περίοδος ρολογιού της λεωφόρου γίνεται συγκρίσιμη με τον χρόνο που χρειάζονται τα ηλεκτρομαγνητικά κύματα (που ταξιδεύουν με την ταχύτητα του φωτός) να διανύσουν το μήκος της λεωφόρου). Π.χ., στο PCI, αυτό συνέβη όταν θελήσαμε η ταχύτητά του να ανέβει πολύ πάνω από τα 100 MHz --τότε αναπτύχθηκε η παραλλαγή του "PCI-Express (PCIe)", που δεν είναι πλέον λεωφόρος (bus) αλλά σύνδεσμος point-to-point, και σήμερα "τρέχει" συνήθως στα 2.5 ή 5.0 GHz. Ομοίως το σημερινό "Gigabit Ethernet (GigE)", που "τρέχει" στο 1 Gbit/s (ή και το νεώτερο

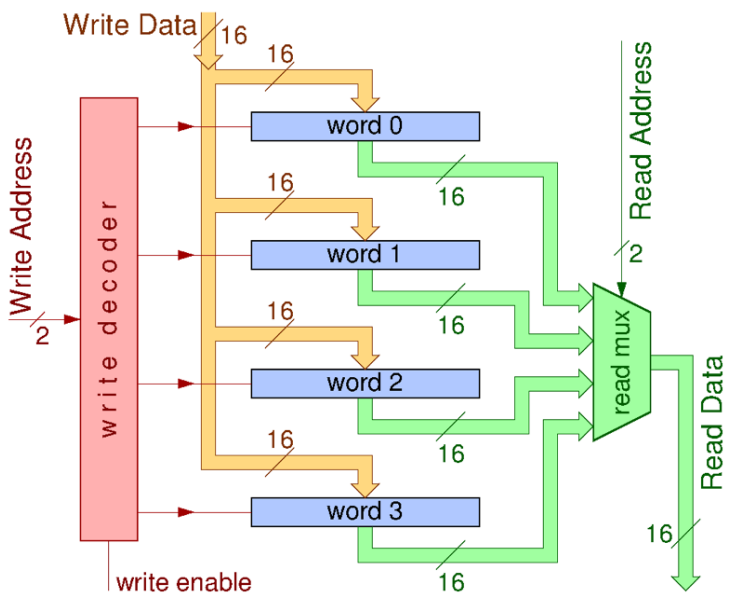


10GigE, στα 10 Gbits/s), χρησιμοποιούν πλέον συνδέσμους με έναν μόνο σταθμό που μιλάει κατευθείαν στον μεταγωγέα (switch) ή δρομολογητή (router) --δηλαδή point-to-point-- και όχι πιά πολλαπλούς σταθμούς συνδεδεμένους όλους στο ίδιο σύρμα (bus). Τα προβλήματα που οδηγούν σε αυτή την εξέλιξη είναι τα εξής (που απαιτούν πίο ειδικευμένες γνώσεις για την κατανόησή τους): (i) οι ανακλάσεις των ηλεκτρομαγνητικών κυμάτων στις ενδιάμεσες επαφές σε ένα σύρμα λεωφόρου· (ii) ο νεκρός χρόνος αναστροφής κατεύθυνσης μετάδοσης (turn-around delay) που πρέπει να περιμένουμε χωρίς κανέναν να μεταδίδει, μετά το τέλος της μετάδοσης από ένα πομπό, μέχρι να "αδειάσει" η λεωφόρος από τα ηλεκτρομαγνητικά κύματα της μετάδοσης αυτής, και πριν μπορέσει ένας νέος πομπός να μεταδώσει στην αντίθετη κατεύθυνση· (iii) το πρόβλημα "clock skew", δηλαδή η διαφοροποίηση στην ταχύτητα μετάδοσης των διαφόρων bits μιάς λέξης που ταξιδεύουν από διαφορετικούς δρόμους καθένα, από δεδομένη πηγή σε δεδομένο προορισμό· και (iv) η σταθερή καθυστέρηση της διατησίας, που καθορίζεται από τη φυσική απόσταση των συνομιλούντων (και το χρόνο που χρειάζονται να ηλεκτρομαγνητικά κύματα για να την διανύσουν), και άρα δεν επιταχύνεται όταν μεγαλώνει η συχνότητα του ρολογιού με το οποίο δουλεύει η λεωφόρος.

9.3 Εσωτερική Οργάνωση Μνημών SRAM

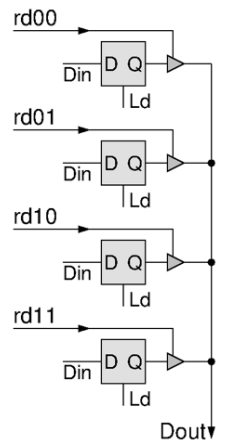


Στην §1.8 είδαμε την γενική ιδέα της μνήμης· εδώ επαναλαμβάνουμε τα δύο σχήματα από εκείνη την ενότητα. Θυμηθείτε ότι η μνήμη είναι μία σειρά από λέξεις (words), όπου η κάθε λέξη αποτελείται από κάμποσα bits. Αυτά είναι τα δεδομένα (data) που είναι αποθηκευμένα στη μνήμη. Κάθε φορά, διαβάζουμε ή γράφουμε μία και μόνο μία από τις λέξεις. Για να επιλέξουμε το ποιά λέξη θέλουμε να διαβάσουμε ή γράψουμε, δίνουμε τα bits διεύθυνσης (address), που φαίνονται με κόκκινο χρώμα στα σχήματα. Στην §4.12 είδαμε με περισσότερες λεπτομέρειες τη δομή μιάς μνήμης. Εδώ θα δούμε συγκεκριμένα πλέον την εσωτερική οργάνωση της Μνήμης Τυχαίας Προσπέλασης (RAM - Random Access Memory). Φυσικά, η μνήμη είναι κατασκευασμένη,



πρώτ' απ' όλα, από στοιχεία μνήμης --μανταλωτές (latches) τύπου D, όπως αυτοί που είδαμε στην § 7.3 (θα μπορούσε να κατασκευαστεί και από ακμοπυροδοτητα flip-flops, όπως της § 8.5, αλλά κάτι τέτοιο θα κόστιζε υπερδιπλάσια, άνευ λόγου). Η διαφορά από τους καταχωρητές είναι ότι η μνήμη περιέχει εσωτερικά *πολλές* λέξεις (πολλούς καταχωρητές), αλλά *εξωτερικά* εμείς θέλουμε να βλέπουμε, κάθε φορά, *μία μόνο* από αυτές τις λέξεις –διαφορετική κάθε φορά, και με τυχαία σειρά, εξ ου και ο όρος μνήμη *τυχαίας* προσπέλασης –διότι δεν θα μπορούσαμε να τις βλέπουμε όλες ταυτόχρονα: είναι πάρα πολλές. Για να μπορούμε να επιλέγουμε και να βλέπουμε (διαβάζουμε) μία οιαδήποτε από τις πολλές (εκατομμύρια!) λέξεις μιάς μνήμης, χρειαζόμαστε το γνωστό μας κύκλωμα επιλογής –τον *πολυπλέκτη*– όπως δείχνει το επόμενο σχήμα με πράσινο χρώμα. Βεβαίως, δεν ξεχνάμε ότι ο πολυπλέκτης έχει μέσα του κρυμμένον έναν αποκωδικοποιητή – τον αποκωδικοποιητή διευθύνσεων ανάγνωσης (read address), εδώ. Στο ίδιο σχήμα φαίνεται και το κύκλωμα εγγραφής, με πορτοκαλί χρώμα για τα δεδομένα εγγραφής και με κόκκινο χρώμα για τη διεύθυνση εγγραφής (write address) και τον αποκωδικοποιητή της. Εδώ είχαμε χωριστή διεύθυνση ανάγνωσης και εγγραφής –παρακάτω θα έχουμε μίαν ενιαία και για τις δύο λειτουργίες, για οικονομία στους αποκωδικοποιητές (μετρώντας και εκείνον που είναι κρυμμένος στον πολυπλέκτη ανάγνωσης).

Σε μία πραγματική μνήμη, με χιλιάδες ή εκατομμύρια λέξεις, εάν ο πολυπλέκτης ανάγνωσης ήταν κατασκευασμένος με τις κλασικές πύλες (όπως στην § 4.9), τότε θα χρειαζόμασταν εσωτερικά χιλιάδες ή εκατομμύρια σύρματα από τους μανταλωτές όλων των λέξεων μέχρι το σημείο πολύπλεξης, και αυτά τα τόσο πολλά και τόσο μακρυνά σύρματα θα κόστιζαν υπερβολικά. Αντ' αυτού, οι μνήμες φτιάχνονται εσωτερικά με το ισodύναμο των τρικατάστατων οδηγητών, όπως είδαμε στις προηγούμενες δύο παραγράφους, και όπως δείχνει το σχήμα δεξιά. Στο σχήμα αυτό υπάρχουν 4 μανταλωτές, και ο καθένας έχει έναν τρικατάστατο οδηγητή στην έξοδό του. Η πολύπλεξη γίνεται απλά ενώνοντας τις εξόδους των τρικατάστατων οδηγητών· όσο πολλοί και να είναι οι μανταλωτές με τους οδηγητές τους, η πολύπλεξη απαιτεί ένα μόνον (κατακόρυφο) σύρμα. (Στα πραγματικά κυκλώματα SRAM (Στατική RAM), ο μανταλωτής μαζί με τον τρικατάστατο οδηγητή υλοποιείται με 6 μόνον transistors, με ένα πολύ έξυπνο (ημ-αναλογικό) κύκλωμα που όμως ξεφεύγει από τα πλαίσια αυτού του μαθήματος. Στις μνήμες DRAM (Δυναμική RAM), αντί για μανταλωτή χρησιμοποιείται ένα μικρό φορτίο αποθηκευμένο πάνω σε ένα μικρό πυκνωτή (και που χρειάζεται αναζωογόνηση (refresh) κάθε λίγες δεκάδες millisecond) και αντί για τρικατάστατο οδηγητή χρησιμοποιείται ένα, μοναδικό transistor).



Στο σχήμα εδώ έχουμε έναν *μονόμπιτο* πολυπλέκτη, που επιλέγει στο 1 bit εξόδου του, Dout (Data out), το περιεχόμενο ενός από τους 4 μανταλωτές. Ο έλεγχος του ποιόν από τους 4 μανταλωτές (ποιάν από τις 4 μονόμπιτες λέξεις) θα διαβάζουμε κάθε φορά, γίνεται από τα 4 σήματα ελέγχου / ενεργοποίησης της πολύπλεξης / ανάγνωσης: rd00, rd01, rd10, rd11· τα δύο bits 00, 01, 10, 11 που εμφανίζονται σε αυτά τα ονόματα αντιστοιχούν στη *διεύθυνση* του bit (της λέξης) που επιλέγουμε και διαβάζουμε την κάθε φορά.

Ένα μοναδικό bit είναι πολύ "λίγη" πληροφορία για να μας ενδιαφέρει να διαβάζουμε μόνο τόσο κάθε φορά από μία μνήμη --συνήθως θέλουμε να διαβάζουμε περισσότερα bits κάθε φορά (όσα ποδαράκια ή σύρματα δεχόμαστε να έχει το chip της μνήμης). Φυσικά, δεν μπορούμε να πάμε ούτε και στην άλλη άκρη και να διαβάζουμε ταυτόχρονα *όλα* τα (εκατομμύρια!) bits που έχει μέσα της μία μνήμη --κάτι ενδιάμεσο θέλουμε, π.χ. 8, ή 16, ή 32, ή 64 bits κάθε φορά. Όπως είπαμε και στην § 1.7, τα bits που τα διαβάζουμε κάθε φορά όλα μαζί τα λέμε *μία λέξη* (word). Στο επόμενο σχήμα, δεξιά, βλέπουμε μία μνήμη αποτελούμενη από 8 λέξεις, όπου η κάθε μία από αυτές τις 8 λέξεις περιέχει 4 bits· είναι, όπως λέμε, μία μνήμη 8x4. Τις 8 λέξεις τις σχεδιάζουμε την μία κάτω από την άλλη, και τα 4 bits της κάθε λέξης τα σχεδιάζουμε το ένα δίπλα στο άλλο.

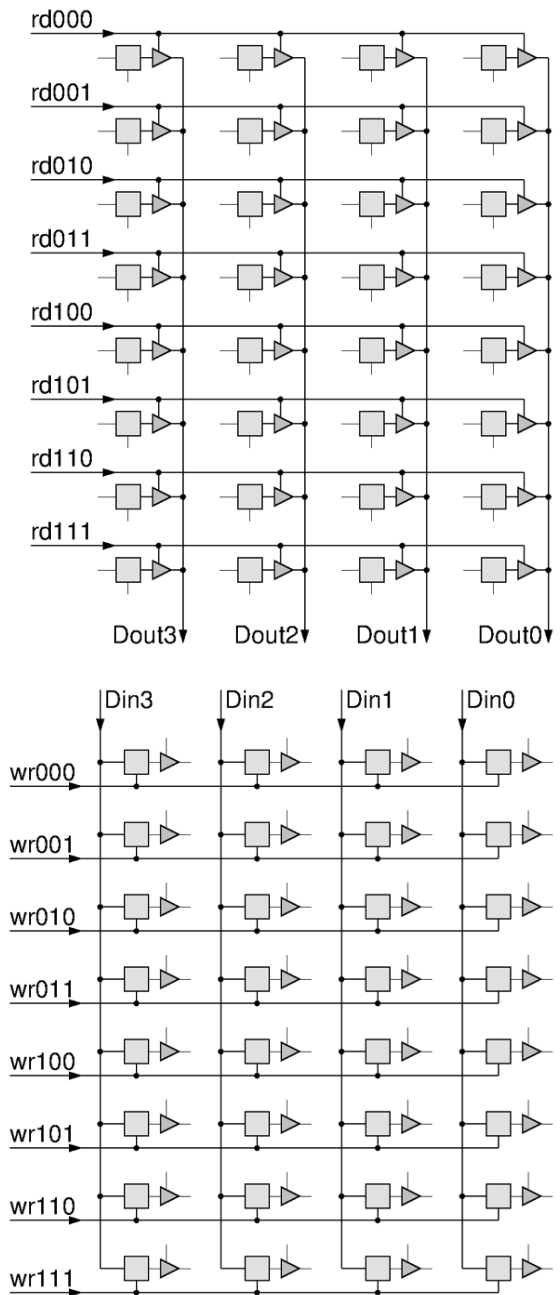
Σε αυτή τη μνήμη που περιέχει τετράμπιτες λέξεις, επιλέγουμε και διαβάζουμε μία από αυτές κάθε φορά χρησιμοποιώντας έναν τετράμπιτο πολυπλέκτη. Οι τετράμπιτοι πολυπλέκτες αποτελούνται από 4 μονόμπιτους πολυπλέκτες, όπως είδαμε στην § 8.1, και αυτός κατασκευάζεται με τρικατάστατους οδηγητές όπως είδαμε παραπάνω, στην § 9.2. Αυτή ακριβώς η κατασκευή φαίνεται και στο σχήμα εδώ: κάθε φορά, ένα και μόνον ένα από τα σήματα επιλογής, rd000, rd001, ..., rd111, είναι αναμένο, ενεργοποιώντας τους 4 αντίστοιχους τρικατάστατους οδηγητές, οι οποίοι αφήνουν τα 4 bits της αντίστοιχης λέξης να περάσουν έξω

και να οδηγήσουν τα 4 κατακόρυφα σύρματα - τα 4 bits της λεωφόρου δεδομένων, Dout3, Dout2, Dout1, Dout0, δηλαδή μία τετράμπιτη λεωφόρο. Αυτή είναι και η διαδικασία ανάγνωσης από αυτή τη μνήμη μεγέθους 8x4.

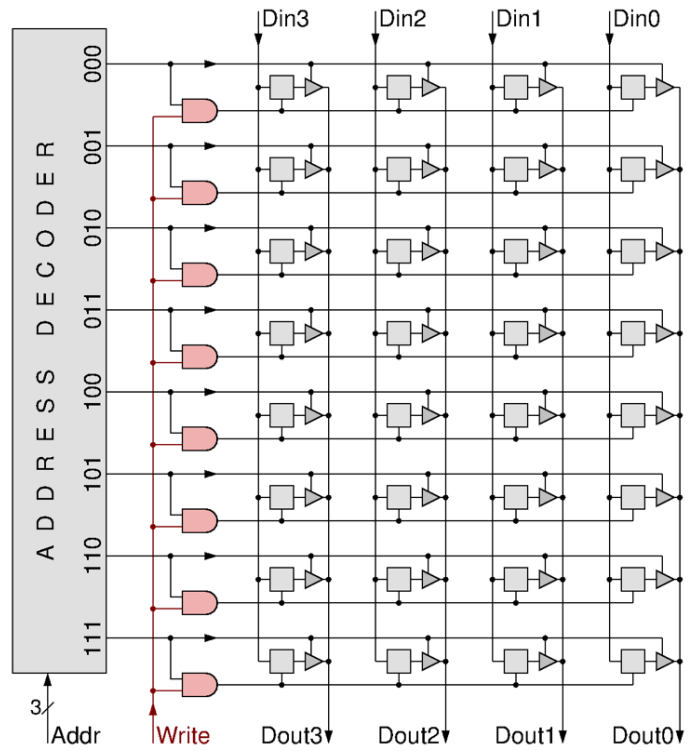
Το επόμενο σχήμα, δεξιά, δείχνει το κύκλωμα και τη διαδικασία εγγραφής στην ίδια αυτή μνήμη μεγέθους 8x4. Για να είναι καθαρότερο το σχήμα, εδώ δείχνουμε μόνο το κύκλωμα εγγραφής, χωρίς τα προηγούμενα σύρματα που χρησιμοποιεσαν για την ανάγνωση. Όπως θυμόμαστε από την § 7.3, για να γράψουμε σ' έναν μανταλωτή τύπου D ανάβουμε το σήμα load enable του (κάτω είσοδος στο τετράγωνο σύμβολο). Στις μνήμες, κάθε πράξη γίνεται πάνω σε μία, ολόκληρη λέξη, και μόνο: όπως όταν διαβάζουμε, διαβάζουμε πάντα *μία λέξη*, έτσι και όταν γράφουμε, γράφουμε πάντα *μία, ολόκληρη λέξη* (και μόνο). Άρα, όπως δείχνει και το σχήμα, τα σήματα load enable των 4 bits της κάθε λέξης (οριζόντια) ενώνονται μεταξύ τους, επομένως πάντα γράφουμε ή σε όλα ή σε κανένα από τα bits της κάθε λέξης. Παρ' ότι θα μπορούσαμε να γράφουμε ταυτόχρονα σε πολλαπλές λέξεις, κάτι τέτοιο δεν θεωρείται χρήσιμο, ακι δεν το κάνουν οι συνηθισμένες μνήμες. Επομένως, τα σήματα ελέγχου / επιλογής λέξης για εγγραφή, wr000, wr001, ..., wr111, ακολουθούν λειτουργία ανάλογη με εκείνη των σημάτων ελέγχου ανάγνωσης: το πολύ ένα από αυτά ανάβει κάθε φορά (εδώ όμως είναι συνηθισμένο και αναμενόμενο να υπάρχουν πολλές φορές όπου *κανένα* από τα σήματα αυτά δεν είναι αναμένο, όταν δεν θέλουμε να γράψουμε τίποτα και πουθενά, προκειμένου να διατηρήσουν όλες οι λέξεις μνήμης τα (χρήσιμα) περιεχόμενά τους).

Παραπάνω συζητήσαμε τον έλεγχο / επιλογή του πότε θα γράψουμε και σε ποιά λέξη θα γράψουμε. Το επόμενο που μένει να καθοριστεί είναι το *τι* θα γράψουμε. Αφού κάθε φορά που γράφουμε, γράφουμε μία (τετράμπιτη) λέξη, προφανώς πρέπει να υπάρχει μία (τετράμπιτη) εξωτερική είσοδος που να μας τροφοδοτεί μία (τετράμπιτη) λέξη δεδομένων η οποία και πρέπει να γραφτεί στη λέξη εκείνη της μνήμης όπου έχουμε επιλέξει (μέσω των wr000, ..., wr111) να κάνουμε την εγγραφή. Η τετράμπιτη αυτή είσοδος είναι τα 4 κατακόρυφα σύρματα "Data In" (Din): Din3, Din2, Din1, Din0. Την τετράμπιτη αυτή τιμή την βλέπουν *όλες* οι λέξεις της μνήμης στην είσοδό τους, αλλά *μόνον* η μία επιλεγμένη (μέσω των wr000, ..., wr111) λέξη την παραλαμβάνει και την εγγράφει μέσα της --οι υπόλοιπες λέξεις της μνήμης, που έχουν σβηστά τα load enable τους, απλώς την αγνοούν.

Τώρα που καταλάβαμε πώς λειτουργεί η ανάγνωση και πώς λειτουργεί η εγγραφή από και στην μνήμη μας μεγέθους 8x4 (8 λέξεις των τεσσάρων bits καθεμία), το σχήμα εδώ δεξιά βάζει μαζί τα δύο αυτά κυκλώματα, και δείχνει την ολοκληρωμένη εσωτερική δομή μιάς μνήμης. Στο δεξιό μέρος του σχήματος, πάνω από κάθε λέξη περνάει ένα σήμα ενεργοποίησης ανάγνωσης, που ελέγχει τους τρικατάστατους οδηγητές ανάγνωσης, και κάτω από κάθε λέξη περνάει ένα σήμα ενεργοποίησης εγγραφής, που ελέγχει τα σήματα load enable των μανταλωτών της λέξης. Δεξιά από κάθε στήλη bits περνάει ένα (τρικατάστατο) σύρμα εξόδου δεδομένων ανάγνωσης, και αριστερά από κάθε στήλη bits περνάει ένα σύρμα εισόδου που φέρνει δεδομένα για (πιθανή) εγγραφή σε μία λέξη.



Στο αριστερό μέρος του σχήματος υπάρχει ο Αποκωδικοποιητής Διευθύνσεων (Address Decoder). Προφανώς, τα χιλιάδες (ή εκατομμύρια) (οριζόντια) σύρματα ελέγχου για τις χιλιάδες (ή εκατομμύρια) λέξεις μιάς πραγματικής μνήμης, ούτε θέλουμε ούτε μπορούμε να τα οδηγούμε και ελέγχουμε όλα από έξω από το chip της μνήμης: προφανώς θέλουμε να τα δίνουμε κωδικοποιημένα απ' έξω, και να αποκωδικοποιούνται εσωτερικά --έτσι, και το πλήθος τους μειώνεται δραματικά, και εξασφαλίζουμε την ιδιότητα ποτέ να μην ανάβει πάνω από ένα από αυτά. Στο παράδειγμά μας εδώ, τρία σύρματα Διεύθυνσης, Addr, έρχονται από έξω, και αποκωδικοποιούνται για να μας δώσουν τα 8 σήματα που φαίνονται, με την ιδιότητα ένα και μόνον ένα από αυτά να είναι πάντα αναμένο.



Οι έξοδοι του αποκωδικοποιητή οδηγούν κατευθείαν τα σήματα ελέγχου ανάγνωσης (αυτά που είναι πάνω από την κάθε λέξη). Αυτό σημαίνει ότι πάντα ανάβει ένα (και μόνον ένα) από αυτά τα σήματα, άρα πάντα διαβάζουμε μία (και μόνο μία) λέξη. Αυτό είναι αποδεκτό: ακόμα κι όταν δεν χρειαζόμαστε να διαβάσουμε καμία λέξη, και ακόμα κι όταν η διεύθυνση είναι "σκουπίδια" (άγνωστη, άκυρη, λανθασμένη), το να διαβάζουμε μία τυχαία --πιθανότατα άχρηστη-- λέξη είναι αβλαβές, δηλαδή δεν καταστρέφει καμία χρήσιμη πληροφορία.

Από την άλλη μεριά, τα πράγματα είναι διαφορετικά για τις εγγραφές: εδώ, ΔEN είναι αποδεκτό να γράφουμε σε άγνωστες / άκυρες / λανθασμένες θέσεις (διευθύνσεις) της μνήμης, διότι η εγγραφή καταστρέφει τα παλαιά περιεχόμενα της λέξης όπου γίνεται. Επομένως, απαιτείται ένας μηχανισμός που θα αδρανοποιεί (disable) τη λειτουργία εγγραφής σε ολόκληρη τη μνήμη, όποτε δεν είμαστε σίγουροι ότι η διεύθυνση είναι σωστή, ή όποτε η διεύθυνση αλλάζει και ο αποκωδικοποιητής δεν πρόλαβε ακόμα να σταθεροποιήσει τις εξόδους του στη νέα, σωστή τιμή τους, ή όποτε η είσοδος δεδομένων (Din) δεν έχει σωστές / χρήσιμες τιμές. Τη λειτουργία αυτής της αδρανοποίησης την επιτελεί το (κόκκινο) σήμα Write και οι (κόκκινες) πύλες AND: όταν Write=0, τότε όλα τα σήματα ελέγχου εγγραφής είναι =0 (σβηστά), ανεξαρτήτως του τι κάνει η Διεύθυνση και ο αποκωδικοποιητής, άρα καμιά από τις πληροφορίες που βρίσκονται αποθηκευμένες στη μνήμη δεν καταστρέφεται. Η σωστή χρήση του σήματος Write απαιτεί πρώτα να δώσουμε σωστή διεύθυνση (και δεδομένα) εγγραφής, και μετά να ανάψουμε το Write: όταν Write=1, τότε ακριβώς ένα από τα σήματα ελέγχου εγγραφής ανάβει --εκείνο που αντιστοιχεί στη σωστή και σταθερή διεύθυνση εγγραφής-- και τότε εγγράφονται τα σωστά δεδομένα στη σωστή θέση (λέξη) της μνήμης. (Παρατηρήστε ότι η μνήμη αυτή (για απλότητα σχεδίασης) πάντα εκτελεί λειτουργία ανάγνωσης: όταν κάνουμε εγγραφή, τότε διαβάζει από την ίδια θέση (διεύθυνση) στην οποία κάνουμε την εγγραφή, επομένως διαβάζει τα νέα δεδομένα που γράφονται εκεί-- αυτό είναι πιθανότατα άχρηστο --αλλά και αβλαβές).




9.4 Μεγαλύτερες Μνήμες, κατασκευασμένες από πολλαπλά Chips

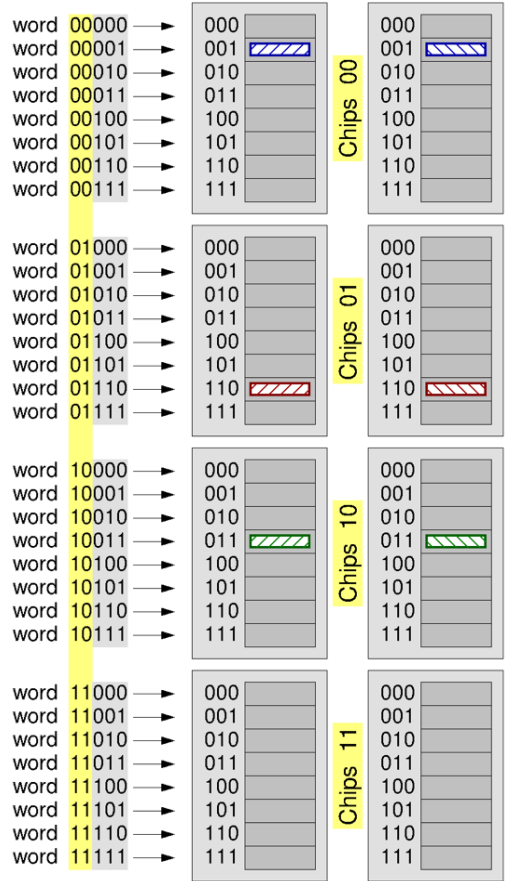
Πολύ συχνά, θέλουμε να φτιάξουμε μιά μνήμη πύο μεγάλη απ' όσο ένα chip μνήμης που βρίσκουμε έτοιμο να αγοράσουμε. Έτσι, την κατασκευάζουμε από πολλαπλά chips μνήμης. Η συνολική μνήμη έχει προφανώς χωρητικότητα σε bits ίση με το άθροισμα των χωρητικότητων, σε bits, των chips από τα οποία κατασκευάζεται: όμως, ποιά είναι η οργάνωση της συνολικής μνήμης --πόσες λέξεις, των πόσων bits καθεμία, και πώς πάνε οι συνολικές διευθύνσεις; Είχαμε μελετήσει το πρόβλημα αυτό θεωρητικά στην § 6.1, και το σχήμα εδώ το δείχνει με ένα παράδειγμα. Στο

σχήμα εδώ, ξεκινάμε με υποτιθέμενα chips μνήμης χωρητικότητας 8 λέξεων καθένα, και θέλουμε να φτιάξουμε μία μνήμη 32 λέξεων. Για το λόγο αυτό βάζουμε 4 chips "το ένα πάνω από το άλλο", αφού οι 32 λέξεις είναι τέσσερις οκτάδες λέξεων. Επιπλέον η κάθε λέξη στη μνήμη που θέλουμε να φτιάξουμε έχει διπλάσιο πλήθος bits απ' όσα έχει η κάθε λέξη των chips που μας δίνουν --π.χ. θέλουμε μνήμη 32x16 (32 λέξεις, καθεμία των 16 bits), και μας έχουν δώσει chips μεγέθους 8x8 (8 λέξεις, καθεμία των 8 bits). Για το λόγο αυτό χρειαζόμαστε "ζευγάρια από chips", δίπλα-δίπλα μεταξύ τους. Για να το πούμε πώ επίσημα, αν w το πλάτος του κάθε chip, θέλουμε συνολική μνήμη $32 \times 2w = 64w$ bits συνολικά. Για να την φτιάξουμε με chips των $8 \times w = 8w$ bits καθένα, θα χρειαστούμε 8 chips, διότι $8 * 8w = 64w$. Τα 8 αυτά chips πρέπει να οργανωθούν σαν 4 γραμμές καθ' ύψος επί 2 στήλες κατά πλάτος.

Στο σχήμα, το κάθε γκριζό ορθογώνιο παριστάνει ένα chip μνήμης. Εφ' όσον το κάθε chip έχει 8 λέξεις μέσα του, παίρνει διεύθυνση 3 bits· αυτές οι διευθύνσεις των 3 bits φαίνονται μέσα σε κάθε chip, δίπλα σε 8 κουτάκια που συμβολίζουν τις 8 λέξεις του κάθε chip. Αφού η συνολική μνήμη που θα φτιάξουμε θέλουμε να έχει 32 λέξεις, αυτή θα παίρνει διευθύνσεις των 5 bits· αυτές φαίνονται στο σχήμα αριστερά. Αντιστοιχίζουμε, όπως στο σχήμα, τις 32 διευθύνσεις της συνολικής μνήμης, 00000, 00001, ..., 11111, στα 4 chips και στις $4*8 = 32$ λέξεις τους, με τον προφανή τρόπο που φαίνεται στο σχήμα. Όπως παρατηρούμε, επειδή το πλήθος λέξεων σε κάθε chip είναι δύναμη του 2, και λόγω της διαιρέσης διά δύναμη του 2 που είχαμε πεί στην § 6.1, τα 2 αριστερά bits της συνολικής διεύθυνσης (σημειωμένα με κίτρινο στο σχήμα) αντιστοιχούν ακριβώς στην αρίθμηση των τεσσάρων chips, από 00 έως 11, και τα 3 δεξιά bits της συνολικής διεύθυνσης (σημειωμένα με γκριζό στο σχήμα) αντιστοιχούν ακριβώς στη διεύθυνση που θα έχει η αντίστοιχη λέξη μέσα στο chip στο οποίο θα βρεθεί.

Περαιτέρω, η κάθε λέξη της συνολικής μνήμης έχει διπλάσιο πλάτος (π.χ. 16 bits) από την κάθε λέξη ενός chip (π.χ. 8 bits). Για το λόγο αυτό, "κόβουμε" τις εξωτερικές λέξεις στο αριστερό μισό και το δεξιό μισό, και τις αποθηκεύουμε, καθεμία, μισή στο αριστερό chip και μισή στο δεξιό. Όποτε διαβάζουμε, θα διαβάζουμε από ένα ζευγάρι chips, το ένα δίπλα στο άλλο. Το ποιο από τα 4 ζευγάρια, μας το λένε τα 2 αριστερά bits της (εξωτερικής) διεύθυνσης. Στο κάθε chip απ' όπου διαβάζουμε, που περιμένει μία διεύθυνση 3 bits για να δουλέψει, θα δώσουμε τα 3 δεξιά bits της εξωτερικής διεύθυνσης, για να τα χρησιμοποιήσει σαν εσωτερική διεύθυνση. Το κάθε chip θα διαβάσει και θα μας δώσει τη ιση από τη λέξη που ψάχνουμε· εμείς, εξωτερικά, θα ανασυνθέσουμε τη λέξη που θέλουμε από τα δύο μισά της. Όταν γράφουμε, ομοίως θα γράψουμε σε ένα ζευγάρι chips, στο αριστερό την αριστερή μισή λέξη, και στο δεξιό την δεξιά μισή λέξη. Στο σχήμα φαίνονται επάνω τρεις "εξωτερικές" λέξεις, η μπλέ, η κόκκινη, και η πράσινη, κι τις οποίες θέλουμε να τοποθετήσουμε αντίστοιχα στις διευθύνσεις 00001, 01110, και 10011· το σχήμα δείχνει μέσα σε ποιά chips και σε ποιές θέσεις μέσα σε κάθε chip θα τοποθετηθεί ποιο κομάτι ποιās λέξης.

contents at address 00001: 
 contents at address 01110: 
 contents at address 10011: 



9.5 Αμφίδρομα Pins Δεδομένων σε Τρικατάστατες Λεωφόρους και σε SRAM:

Τα σύρματα που οδηγούνται από κλασσικές (όχι τρικατάστατες) πύλες έχουν μονόδρομη φορά ροής της πληροφορίας (unidirectional wire), από την έξοδο της οδηγήτριας πύλης προς τις εισόδους που συνδέονται στο σύρμα (§3.5). Αντίθετα, οι τρικατάστατες λεωφόροι επιτρέπουν

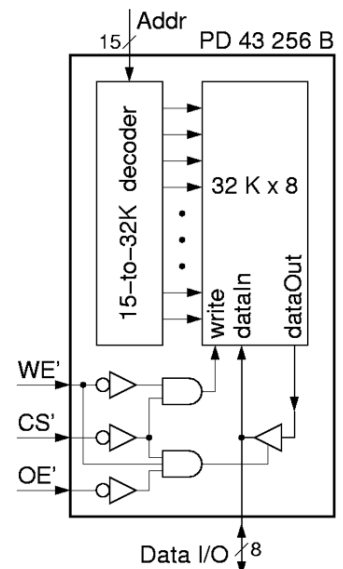
αμφίδρομη (bidirectional) μεταφορά πληροφορίας: η κατεύθυνση μπορεί να αλλάζει κάθε φορά που αλλάζει ο ενεργός οδηγητής. Την ιδιότητα αυτή την εκμεταλλεύονται οι συσκευές που επιθυμούν να επικοινωνούν αμφίδρομα (άλλοτε αποστολή και άλλοτε λήψη πληροφοριών) με άλλες συσκευές μέσω πολυπλέκτη-λεωφόρου. Εάν η επικοινωνία γίνει με κλασικές πύλες, χρειάζονται χωριστά σύρματα εισόδου και χωριστά σύρματα εξόδου, όπως στη μνήμη του σχήματος 7-27 του βιβλίου (σελ. 370). αν όμως η επικοινωνία γίνει μέσω τρικατάστατης λεωφόρου, τότε αρκεί ή συσκευή να επικοινωνεί με τον έξω κόσμο μέσω του ενός, μοναδικού σύρματος ανά bit της λεωφόρου. Ως αμφίδρομα, τα σύρματα αυτά συχνά ονομάζονται σύρματα **"εισόδου/εξόδου"** (input/output - **I/O**).

Ας χρησιμοποιήσουμε σαν παράδειγμα το chip στατικής RAM (SRAM) του παρόντος εργαστηρίου: πρόκειται για το chip "PD43256BCZ" (της εταιρείας NEC) που είναι μία SRAM μεγέθους 32Kx8· ένα λογικό διάγραμμα υψηλού επιπέδου του chip φαίνεται στο σχήμα, και περισσότερες πληροφορίες γι' αυτό μπορείτε να βρείτε στο: <http://www.downloads.reactivemicro.com/Public/Electronics/Static%20RAM/NEC%20uPD43256B%2032k%20x%208bit%20SRAM%20Datasheet.pdf> . Το chip αυτό έχει μία μονόδρομη, δεκαπεντάμπιτη είσοδο διεύθυνσης, τρεις μονόδρομες εισόδους ελέγχου, και μία αμφίδρομη, οκτάμπιτη είσοδο/εξόδο δεδομένων. Τα τελευταία αυτά 8 σύρματα προορίζονται να συνδεθούν σε μία λεωφόρο δεδομένων. Όταν θέλουμε να γράψουμε στη μνήμη, πρέπει κάποια άλλη συσκευή να οδηγήσει τη λεωφόρο, βάζοντας εκεί τα δεδομένα που θέλουμε να εγγραφούν· η μνήμη διαβάζει τα δεδομένα από τα σύρματα αυτά, και τα αποθηκεύει στην επιθυμητή λέξη. Όταν θέλουμε να διαβάσουμε από τη μνήμη, ενεργοποιούμε τους τρικατάστατους οδηγητές που έχει μέσα του το chip, οπότε αυτό γίνεται ο οδηγητής της λεωφόρου και τοποθετεί εκεί τη λέξη που του ζητήσαμε να διαβάσει· από εκεί, τη λέξη αυτή θα την πάρουν οι άλλοι "ακροατές" που συνδέονται πάνω στη λεωφόρο.

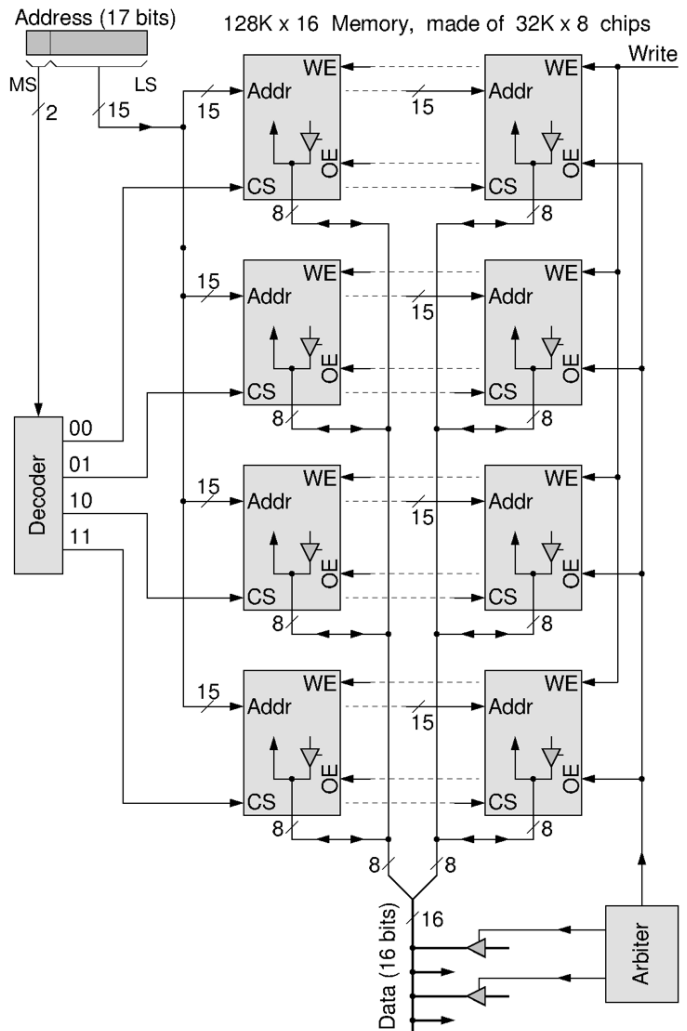
Από άποψη χρονισμού, η λειτουργία του PD43256B έχει ως εξής. Η ανάγνωση από το chip μπορεί να θεωρηθεί σαν καθαρά συνδυαστική πράξη: μπορούμε να φανταστούμε ότι ένας μεγάλος, οκτάμπιτος πολυπλέκτης 32K-σε-1 φέρνει στον εσωτερικό κόμβο "dataOut" το περιεχόμενο της λέξης εκείνης την οποία επιλέγουν τα 15 bits της διεύθυνσης· μόλις αλλάξει η διεύθυνση, μετά από λίγο (καθυστέρηση ανάγνωσης) αλλάζουν και τα dataOut και γίνονται ίσα με τα περιεχόμενα της νέας θέσης μνήμης (η πραγματική εσωτερική λειτουργία δεν είναι έτσι, αλλά αυτό δεν μας αφορά εδώ). Η εγγραφή, από την άλλη μεριά, πρέπει να γίνει προσεκτικότερα: πρώτα πρέπει να τοποθετηθεί η επιθυμητή διεύθυνση εγγραφής στην είσοδο διεύθυνσης, και **μετά** από αυτό και μόνο πρέπει να ανάψει το έσωτερικό σήμα "write". Όση ώρα είναι αναμένο το σήμα write πρέπει η διεύθυνση να παραμένει σταθερή· καθ' όλη αυτή την ώρα, η επιλεγόμενη λέξη μνήμης λειτουργεί σαν μανταλωτής: ό,τι δεδομένα υπάρχουν πάνω στις εισόδους "dataIn" εγγράφονται στη λέξη --προφανώς, τα τελευταία τέτοια δεδομένα πριν σβήσει το σήμα write θα παραμείνουν στην επιλεγμένη λέξη μνήμης. Η είσοδος διεύθυνσης επιτρέπεται να αλλάξει μόνον αφού σβήσει το σήμα write.

Η εξωτερική συμπεριφορά του chip προσδιορίζεται από τους παραπάνω κανόνες και από τα τρία σήματα ελέγχου, WE', CS', και OE', σύμφωνα με το λογικό διάγραμμα που φαίνεται στο σχήμα. Και τα τρία σήματα ελέγχου έχουν αρνητική πολικότητα: ενεργοποιούν τη λειτουργία τους, το καθένα, όταν είναι μηδέν (0). Το σήμα CS' (αρνητικό Chip Select) ενεργοποιεί ή αδρανοποιεί ολόκληρο το chip· προορίζεται για χρήση όταν φτιάχνουμε μία μεγάλη μνήμη από πολλά chips, για να επιλέγουμε σε ποίο chip απευθυνόμαστε κάθε φορά. Όταν CS'=0 (ενεργό chip), το σήμα WE' (αρνητικό Write Enable) ενεργοποιεί την εγγραφή, δηλαδή ανάβει το write, οπότε γίνεται μηδέν (0). Τέλος, όταν CS'=0 (ενεργό chip), το σήμα OE' (αρνητικό Output Enable) ανάβει τους τρικατάστατους οδηγητές της λεωφόρου εξόδου, όταν WE'=1 δηλαδή όταν δεν κάνουμε εγγραφή (άρα κάνουμε ανάγνωση), με άλλα λόγια όταν θέλουμε η λέξη που διαβάζουμε να βγει έξω από το chip, στη λεωφόρο δεδομένων.

Στο επόμενο σχήμα, δεξιά, φαίνεται ένα παράδειγμα, πώς με 8 chips σαν τα παραπάνω θα φτιάχναμε μία μεγαλύτερη μνήμη, μεγέθους 128 K x 16. Επειδή 128 K = 4 * 32 K, χρειαζόμαστε 4



σειρές ("ορόφους") από chips των 32 K λέξεων, η μία σειρά πάνω από την άλλη. Επειδή το επιθυμητό πλάτος 16 bits = $2 * 8$ bits που δίνει το κάθε chip, χρειαζόμαστε 2 στήλες από chips, για να φτιάχνουν ζευγάρια, το ένα δίπλα στο άλλο. Επειδή η συνολική μνήμη έχει 128 K λέξεις, οι διευθύνσεις για αυτήν θα έχουν 17 bits ($2^{17} = 128 \text{ K}$). Επειδή το κάθε chip έχει εσωτερικά 32 K λέξεις, χρειάζεται 15 bits διεύθυνση ($2^{15} = 32 \text{ K}$). Άρα, τα 2 αριστερά (MS) bits της εξωτερικής διεύθυνσης επιλέγουν ένα από τα 4 ζευγάρια (σειρές, ορόφους) chips· αυτό το κάνουν μέσω ενός αποκωδικοποιητή 2-σε-4 του οποίου οι έξοδοι οδηγούν τα Chip-Select (CS) των chips --καθεμία τα CS των δύο chips του ζευγαριού της (για απλότητα εδώ, θεωρούμε τα σήματα ελέγχου θετικής πολικότητας, δηλαδή ανάποδα από το τι κάνουν τα πραγματικά chips στο εργαστήριο). Τα 15 δεξιά bits της εξωτερικής διεύθυνσης μας λένε ποιά λέξη μέσα στο επιλεγμένο chip θέλουμε να διαβάσουμε ή να γράψουμε. Άρα, τα 15 αυτά bits διεύθυνσης θα τα δώσουμε και στα οκτώ chips σαν (εσωτερική τους) διεύθυνση: τα 6 από τα 8 chips που δεν επιλέγονται (CS=0) βλέπουν αυτή την (πλασματική) διεύθυνση, αλλά την αγνοούν επειδή είναι αδρνοποιημένα (CS=0)· τα 2 επιλεγμένα από τα 8 chips (που έχουν CS=1) χρησιμοποιούν σωστά αυτή τη διεύθυνση για να επιλέξουν μέσα τους την επιθυμητή ημι-λέξη (8 bits το καθένα, για να φτιάξουν την εξωτερική λέξη των 16 bits).



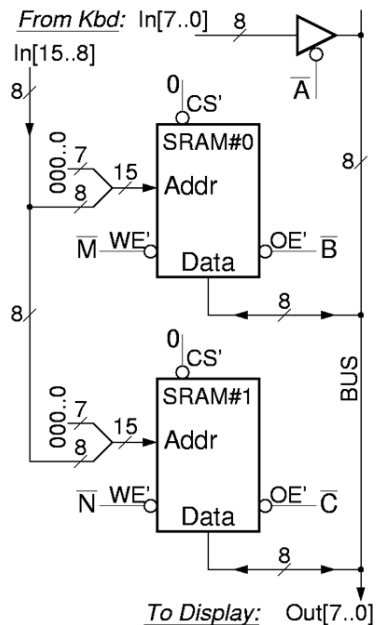
Τα 8 (αμφίδρομα) data pins των 4 αριστερών chips, τα συνδέουμε μεταξύ τους (κάθε τετράδα αντίστοιχων pins), ώστε να σχηματίσουν μιά "αριστερή" (ημί-) λεωφόρο. Ομοίως, τα 8 (αμφίδρομα) data pins των 4 δεξιών chips, τα συνδέουμε μεταξύ τους, ώστε να σχηματίσουν τη "δεξιά" (ημί-) λεωφόρο. Η αριστερή (οκτάμπιτη) ημι-λεωφόρος και η δεξιά (επίσης οκτάμπιτη) ημι-λεωφόρος, θεωρούμενες και οι δύο μαζί (βλ. κάτω μέρος του σχήματος) σχηματίζουν την δεκαξάμπιτη λεωφόρο δεδομένων που μας ζητήθηκε (αφού μας ζητήθηκε μιά μνήμη με οργάνωση "x16"). Άλλοι οδηγητές και ακροατές (π.χ. ο επεξεργαστής, περιφερειακές συσκευές) μπορεί να συνδέονται σε αυτή τη λεωφόρο (αφηρημένα σχεδιασμένοι στο κάτω μέρος του σχήματος). Η εξασφάλιση του κανόνα της λεωφόρου ("ένας και μόνον ένας μιλάει κάθε στιγμή") γίνεται ως εξής. Ο διαιτητής (arbiter, κάτω δεξιά) επιλέγει και ενεργοποιεί μία από τις 3 υποτιθέμενες συσκευές (π.χ. επεξεργαστής, περιφερειακό, ή τη μνήμη μας). Αν επιλέξει άλλη συσκευή (χι τη μνήμη μας), τότε δίνει μηδέν (0) στο OE (Output Enable) και των 8 chips της μνήμης μας, άρα κανένα από αυτά δεν μιλάει από τους τρικατάστατους οδηγητές του, αφού μιλάει άλλη συσκευή πάνω στη λεωφόρο. Όταν ο διαιτητής επιλέγει τη μνήμη μας για να μιλήσει (ανάγνωση), τότε ανάβει το σήμα που τροφοδοτεί τα 8 OE των 8 chips μας. Όλα τα chips μας βλέπουν OE=1, αλλά 6 από αυτά έχουν CS=0, άρα ΔΕΝ ανάβουν τους τρικατάστατους οδηγητές τους· δύο από τα chips μας (ένα αριστερά και ένα δεξιά) έχουν CS=1 (από τον αποκωδικοποιητή), άρα αυτά, βλέποντας και OE=1, ανάβουν τους τρικατάστατους οδηγητές τους, και τοποθετούν το καθένα τα 8 bits που διάβασε πάνω στην καθεμία ημι-λεωφόρο, επομένως οι άλλες συσκευές πάνω στη λεωφόρο (π.χ. ο επεξεργαστής) βλέπουν τη (16-μπιτη) λέξη που θέλαμε να διαβάσουμε.

Η εγγραφή λειτουργεί ανάλογα με τα προηγούμενα. Όταν δεν θέλουμε να γράψουμε, όλα τα chips βλέπουν $Write=0$ και δεν γράφουν. Όταν θέλουμε να γράψουμε, κάποια άλλη συσκευή (π.χ. ο επεξεργαστής) ανάβει τους οδηγητές του και βάζει πάνω στη λεωφόρο τη λέξη (data) που θέλουμε να γράψουμε (και δίνει και τη διεύθυνση εγγραφής). Μετά, ανάβουμε το $Write=1$, που το βλέπουν και τα 8 chips της μνήμης. Όμως, 6 από τα 8 chips είναι μη-επιλεγμένα ($CS=0$), άρα έχει αδρανοποιηθεί εσωτερικά τους η εγγραφή, κι επομένως δεν κάνουν τίποτα. Δύο από τα 8 chips (ένα αριστερό κι ένα δεξί) είναι επιλεγμένα ($CS=1$), και αυτά, βλέποντας και $Write=1$, θα γράψουν τα data που βλέπουν πάνω στη λεωφόρο (π.χ. από τον επεξεργαστή) στη διεύθυνση που τους δίδεται: το αριστερό επιλεγμένο chip θα γράψει τα 8 αριστερά bits των data, και το δεξιό επιλεγμένο chip θα γράψει τα 8 δεξιά bits των data.

Πείραμα 9.6: Τρικατάστατη Λεωφόρος με Μνήμες SRAM

Κατασκευάστε και ελέγξτε το κύκλωμα που φαίνεται στο σχήμα. Είναι οργανωμένο γύρω από μία οκτάμπιτη λεωφόρο, BUS. Σ' αυτήν συνδέονται τρεις πηγές πληροφορίας: δύο (οκτάμπιτα) SRAM chips (μέσον) σαν αυτά που μελετήσαμε παραπάνω, και μία πηγή οκτάμπιτων αριθμών (επάνω) που προορίζονται σαν δεδομένα εγγραφής στις μνήμες. Κάθε chip μνήμης έχει 32K (οκτάμπιτες) λέξεις, άρα 15 bits διεύθυνσης- όμως, λόγω περιορισμών στα σύρματα και στην πηγή διευθύνσεων, εμείς θα χρησιμοποιήσουμε μόνο τις πρώτες 256 λέξεις στο κάθε chip, μεταβάλλοντας μόνο τα 8 (LS) bits από τα 15 bits διεύθυνσης, και κρατώντας τα υπόλοιπα 7 bits σταθερά στο 0. Τροφοδοτούμε την ίδια διεύθυνση και στα δύο chips. Σαν πηγή τόσο διευθύνσεων όσο και δεδομένων θα χρησιμοποιήσετε το πληκτρολόγιο (§8.3): τα μεν 8 MS bits του ($In[15..8]$) θα χρησιμοποιήσετε σαν πηγή διεύθυνσης, τα δε 8 LS bits του ($In[7..0]$) θα χρησιμοποιήσετε σαν πηγή δεδομένων. Όταν πληκτρολογείτε ένα νέο τετραψήφιο δεκαεξαδικό αριθμό, τα δύο αριστερά ψηφία του θα προσδιορίζουν τη νέα διεύθυνση, και τα δύο δεξιά ψηφία του θα προσδιορίζουν τα νέα δεδομένα (για να αλλάξετε μόνο το ένα από τα δύο --δεδομένα ή διεύθυνση-- πρέπει δυστυχώς να ξαναπληκτρολογήσετε και το άλλο, αμετάβλητο).

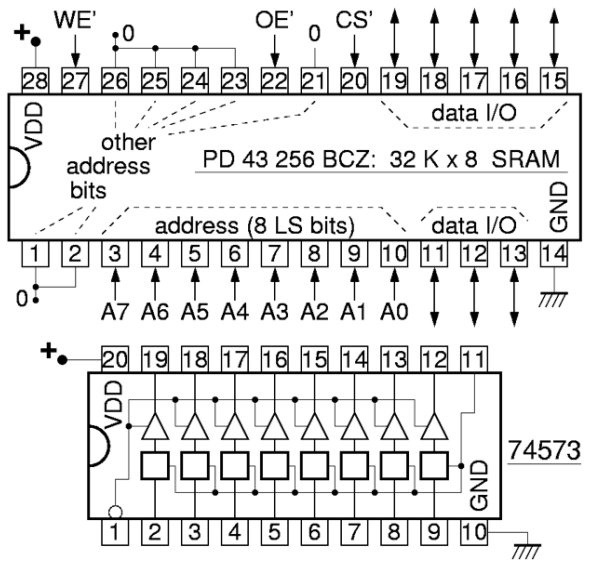
Στη λεωφόρο συνδέεται επίσης και ένας ακροατής, η οθόνη υγρών κρυστάλλων (LCD) (για την ακρίβεια μόνο τα 8 LS bits της), για να παρακολουθούμε τη λογική τιμή της λεωφόρου. Λάβετε υπ' όψη σας ότι το κύκλωμα εισόδου της οθόνης έχει ασθενείς αντιστάσεις καθέλκυσης: όταν ένα bit που του συνδέουμε είναι "αιωρούμενο" (floating - HiZ - ασύνδετο - ανοικτό κύκλωμα), τότε αυτή η αντίσταση καθέλκυσης το φέρνει σε χαμηλή τάση, κι έτσι η οθόνη δείχνει 0: όταν κάποιος (τρικατάστατος) οδηγητής οδηγήσει αυτό το bit με 0 ή 1, τότε ο οδηγητής αυτός υπερσχύει της (ασθενούς) αντίστασης καθέλκυσης, και φέρνει το bit στην τάση που αυτός επιθυμεί. Άρα, όποτε η λεωφόρος μένει ανοδηγήτη (δηλαδή όλοι οι τρικατάστατοι οδηγητές της είναι σβηστοί), στην οθόνη θα βλέπετε "00" (δεκαεξαδικό) στα δύο δεξιά ψηφία της. (Τα δύο αριστερά ψηφία της οθόνης αντιστοιχούν στα bits $Out[15..8]$, που είναι ασύνδετα, άρα, για τον ίδιο παραπάνω λόγο, αυτά θα δείχνουν πάντα "00").



Η λειτουργία του κυκλώματος έχει ως εξής. Επιλέγουμε μία διεύθυνση και μία λέξη δεδομένων μέσω του πληκτρολογίου. Στη συνέχεια, πατάμε το διακόπτη A για να ενεργοποιήσουμε τον επάνω τρικατάστατο οδηγητή: η λέξη δεδομένων πρέπει να εμφανιστεί πάνω στη λεωφόρο και την δούμε στην οθόνη. Με πατημένο το διακόπτη A, ανάβουμε και στη συνέχεια σβήνουμε έναν από τους διακόπτες M ή N, ώστε να ενεργοποιήσουμε την εγγραφή σε μία από τις δύο μνήμες: εάν το κύκλωμα λειτουργεί σωστά, τα δεδομένα από τη λεωφόρο θα πρέπει τώρα να έχουν γραφτεί στην επιλεγμένη διεύθυνση της μνήμης αυτής (γράψτε σ' ένα χαρτί τη διεύθυνση και τα δεδομένα για επαλήθευση αργότερα). Αλλάζουμε τα δεδομένα και ενεργοποιούμε το WE' της άλλης μνήμης: τα νέα δεδομένα θα πρέπει να γράφτηκαν στην επιλεγμένη διεύθυνση εκείνης της μνήμης. Αλλάζουμε και τη διεύθυνση και τα δεδομένα, και επαναλαμβάνουμε τη διαδικασία ούτως ώστε να γράψουμε προκαθορισμένες (και διαφορετικές!) λέξεις σε κάμποσες από τις 256 προσπελάσιμες λέξεις καθεμιάς από τις δύο μνήμες. Προσοχή: τη στιγμή της αλλαγής της

διεύθυνσης να μην είναι ενεργός κανένας από τους διακόπτες M ή N. Στη συνέχεια, αφού "γεμίσαμε" (αρχικοποιήσαμε) τις μνήμες, διαβάζουμε τα περιεχόμενά τους για να διαπιστώσουμε αν είναι τα σωστά: επιλέγουμε (μέσω του πληκτρολογίου) τις διευθύνσεις στις οποίες γράψαμε (τα δεδομένα από το πληκτρολόγιο τώρα δεν θα πρέπει να έχουν επίδραση), και για κάθε διεύθυνση πατάμε πρώτα το διακόπτη B για να διαβάσουμε από την πρώτη μνήμη, και μετά το διακόπτη C για να διαβάσουμε από τη δεύτερη μνήμη· κάθε φορά, πρέπει να εμφανίζονται στην οθόνη τα δεδομένα που είχαμε γράψει παλαιότερα σε εκείνη τη διεύθυνση εκείνης της μνήμης (που είναι σκόπιμο να είναι διαφορετικά μεταξύ τους, διαφορετικά από τα δεδομένα άλλων διευθύνσεων, και διαφορετικά από τα τωρινά δεδομένα του πληκτρολογίου). Φυσικά, ποτέ δεν πρέπει να είναι πατημένοι ταυτόχρονα περισσότεροι από ένας από τους διακόπτες A, B, C, οι οποίοι ενεργοποιούν τους τρεις διαφορετικούς οδηγητές της λεωφόρου.

Τα chips μνήμης είναι ακριβά, και γι' αυτό δεν σας έχουν δοθεί χωριστά σε καθέναν, στο "σακουλάκι" της §3.3· για τον ίδιο ακριβώς λόγο, να είστε **πολύ προσεκτικοί** να μην τα κάψετε/χαλάσετε! Θα τα βρείτε έτοιμα (αλλά ασύνδετα) πάνω στην πλακέτα συνδέσεων (breadboard), και όταν τελειώσετε αφήστε τα επίσης πάνω στην πλακέτα, χωρίς να τα βγάλετε, για να μην φθείρονται (στραβώνουν) τα ποδαράκια τους (βγάλτε μόνο τα σύρματά σας, και το δικό σας chip, 74573). Τα pins της SRAM φαίνονται στο σχήμα. Τα pins διεύθυνσης που θα χρησιμοποιήσουμε εμείς είναι τα 3 έως 10· τα υπόλοιπα 7 pins διεύθυνσης, που δεν τα χρησιμοποιούμε, πρέπει να συνδεθούν σε κάποια σταθερή τάση (π.χ. να γειωθούν (λογικό 0)), διότι αλλιώς, αν μείνουν ασύνδετα (αιωρούμενα - floating) μπορεί να παίρνουν τυχαίες (και μεταβαλλόμενες!) τιμές. Τα pins δεδομένων (εισόδου/εξόδου) είναι τα 11, 12, 13, και 15 έως και 19. Τέλος, οι 3 εισοδοί ελέγχου βρίσκονται στα pins 20, 22, και 27. Σαν **τρικατάστατους οδηγητές** για τα δεδομένα εγγραφής, χρησιμοποιήστε τα 8 bits ενός chip μανταλωτών "74573" όπως αυτά που χρησιμοποιήσατε στα 2 τελευταία εργαστήρια. Όπως είχαμε πεί, τα chips αυτά έχουν τρικατάστατες εξόδους. Για να χρησιμοποιήσουμε μόνο τους οδηγητές, χωρίς τους μανταλωτές, αρκεί να συνδέσουμε μονίμως στο 1 το σήμα φόρτωσης (pin 11): τότε, ό,τι υπάρχει στις εισόδους "εγγράφεται" συνεχώς και αμέσως και εμφανίζεται στις εξόδους. Η είσοδος ενεργοποίησης των τρικατάστατων οδηγητών είναι το pin 1, και είναι **αρνητικής** λογικής: ενεργοποιεί τους οδηγητές με 0, και τους αδρανοποιεί με 1.



Ποιν φτάσετε στο εργαστήριο κάντε ένα πλήρες σχεδιάγραμμα συνδεσμολογίας, δείχνοντας ακριβώς ποιά pin τίνος chip πρέπει να συνδεθεί πού, όπως έχουμε ήδη πεί πολλές φορές. Επίσης, γράψτε σ' ένα χαρτί το "πρόγραμμα ελέγχων" σας (test plan): σε ποιές διευθύνσεις θα γράψετε, και τι (διαφορετικά) δεδομένα σε καθεμία, στο κάθε chip. **Στο εργαστήριο**, κατασκευάστε το κύκλωμα και ελέγξτε τη λειτουργία του. Όταν σβήνεται την τάση τροφοδοσίας, τι γίνονται τα περιεχόμενα της μνήμης;

Άσκηση 9.7: Μνήμη Διπλασίου Ύψους (Λέξεων)

Το κύκλωμα του προηγούμενου εργαστηρίου μπορεί, με ελάχιστες αλλαγές στα σήματα ελέγχου, να χρησιμοποιηθεί σαν μία μεγάλη μνήμη με διπλάσιο αριθμό λέξεων απ' όσες λέξεις έχει το κάθε chip (ή χρησιμοποιούμε εμείς στο κάθε chip), δηλαδή μία μνήμη διπλασίου "ύψους" (θεωρώντας ότι το "ύψος" αντιστοιχεί στις λέξεις της μνήμης και το "πλάτος" στα bits της κάθε λέξης). Ας παραμείνουμε προς στιγμήν στις 256 μόνο λέξεις ανά chip του προηγούμενου περιγράμματος (μόνο 8 bits διεύθυνσης στα chips): τότε η "συνολική" μνήμη θα έχει 512 λέξεις, άρα 9 bits διεύθυνσης. Θεωρήστε ότι το ένατο bit διεύθυνσης είναι ο διακόπτης M· χρησιμοποιήστε τον για να ελέγξετε κατάλληλα τα σήματα CS' των chips, διότι έτσι πετυχαίνετε "μ' ένα σμπάρο δυό τρυγόνια": αδρανοποιήτε ταυτόχρονα και τις εγγραφές και τις αναγνώσεις στο εκάστοτε μη επιλεγμένο chip. Όταν το κάνετε σωστά αυτό, μπορείτε στη συνέχεια να οδηγήστε από μίαν ίδια, κοινή είσοδο τα

δύο σήματα WE' των δύο chips, και από μίαν άλλη, επίσης κοινή είσοδο τα δύο σήματα OE' και των δύο chips. Κάντε το σχεδιάγραμμα του κυκλώματος στην αναφορά του εργαστηρίου σας, και εξηγήστε εν συντομία το γιατί. Αν έχετε χρόνο, δοκιμάστε το και στο εργαστήριο (π.χ. οδηγήστε τα δύο WE' από τον διακόπτη B', και τα δύο OE' από τον διακόπτη C').

Εάν στη λεωφόρο δεν υπάρχει κανένας άλλος οδηγητής, τότε μπορούμε να αποφασίσουμε όποτε δεν κάνουμε εγγραφή να κάνουμε πάντοτε ανάγνωση. Πώς θα χρησιμοποιούσατε τότε τον διακόπτη A και μόνο (με τις δύο συμπληρωματικές του εξόδους, A και A') για να ελέγξετε κατάλληλα και τον επάνω τρικατάστατο οδηγητή και τα WE' των δύο chips, και τα OE' των δύο chips, όλα μόνο από τον A; (Ας πούμε ότι "A ελεύθερος" σημαίνει "ανάγνωση", και "A πατημένος" σημαίνει "εγγραφή"). Κάντε το σχεδιάγραμμα του κυκλώματος στην αναφορά του εργαστηρίου σας, και εξηγήστε εν συντομία πώς και γιατί δουλεύει σωστά.

Τέλος, ας αφήσουμε κατά μέρος τον περιορισμό για 8 μόνο bits διεύθυνσης, δηλαδή 256 μόνο λέξεις ανά chip. Θεωρήστε το πλήρες "ύψος" (βάθος) του κάθε chip, των 32K λέξεων. Τότε, η "συνολική" μας μνήμη είναι 64K λέξεις, και χρειάζεται 16 bits διεύθυνσης. Δείξτε σ' ένα σχεδιάγραμμα μιά δεκαεξάμπιτη διεύθυνση να μπαίνει, και πώς αυτή θα τροφοδοτήσει το κύκλωμά μας για να συμπεριφέρεται αυτό σαν μία μεγάλη μνήμη 64K x 8.

Άσκηση 9.8: Η Θέση των bits Δεδομένων και Διευθύνσεων

(1) Στο παραπάνω πείραμα 9.6 δεν δώσαμε σημασία στο ποιο ποδαράκι του chip ήταν το LS bit των δεδομένων, ποιο το επόμενο, κ.ο.κ. Σύμφωνα με το εγχειρίδιο, το pin 11 του chip φαίνεται να είναι το LS bit των δεδομένων, και το pin 12 το επόμενο.

- i. Σχεδιάστε τα 8 flip-flops (8 κουτάκια) μιάς λέξης της μνήμης (π.χ. της 13), και θεωρήστε ότι γράφουμε εκεί ως δεδομένα τον αριθμό F9, έχοντας συνδέσει σωστά το LS bit *BUS[0]* στο ποδαράκι 11 του chip, το επόμενο bit *BUS[1]* στο ποδαράκι 12, κ.ο.κ. Γράψτε μέσα σε κάθε flip-flop (κουτάκι) το σωστό bit, 1 ή 0.
- ii. Θεωρήστε ότι από το παραπάνω κύκλωμα (i) διαβάζουμε την παραπάνω λέξη (ας πούμε την 13). Ποιός αριθμός θα εμφανιστεί στο BUS;
- iii. Έστω τώρα ότι στο κύκλωμα (i) συνδέσαμε ανάποδα το LS bit *BUS[0]* στο ποδαράκι 12 και το bit *BUS[1]* στο ποδαράκι 11, ενώ τα υπόλοιπα bits παραμένουν όπως πριν. Έστω ότι πάλι γράφουμε στη λέξη 13, έχοντας τον αριθμό F9 πάνω στο BUS. Τώρα, τι bits θα μπουν μέσα σε κάθε κουτάκι της λέξης στη μνήμη; --σχεδιάστε όπως στο (i). Ποιός αριθμός θα νόμιζε κανείς ότι είναι αυτός, εάν μπορούσε να "δεί" μέσα στο chip;
- iv. Έστω ότι από το νέο ("ανάποδο") κύκλωμα (iii) διαβάζουμε την παραπάνω λέξη, 13. Ποιός αριθμός θα εμφανιστεί στο BUS; Είναι αυτός ίδιος με τον αριθμό που θα νόμιζε κανείς αν μπορούσε να "δεί" μέσα στο chip, ή ίδιος με τον αριθμό που "γράψαμε" από το BUS; Γιατί; Τι συμπέρασμα βγάζετε;

(2) Μία αντίστοιχη με την παραπάνω περίπτωση συμβαίνει και με τα bits διεύθυνσης! Θεωρήστε ότι γράφουμε στη λέξη 0 τον αριθμό 10, στην 1 τον 11, στην 2 τον 12, και στην 3 τον 13. Σχεδιάστε αυτές τις 4 πρώτες λέξεις της μνήμης, την μία κάτω από την άλλη, και γράψτε μέσα σε καθεμία τα περιεχόμενά της. Στη συνέχεια θεωρήστε ότι, κατά λάθος, κάποιος συνδέσει το LS bit *Addr[0]* της διεύθυνσης στο ποδαράκι 9 του chip που είναι το bit "A1", και το επόμενο bit *Addr[1]* της διεύθυνσης στο ποδαράκι 10 του chip που είναι το bit "A0" (ενώ όλα τα υπόλοιπα bits διεύθυνσης είναι σωστά συνδεδεμένα). Ξαναγράψουμε στη διεύθυνση 0 τον αριθμό 10, στην 1 τον 11, στην 2 τον 12, και στην 3 τον 13· οι διευθύνσεις αυτές αντιστοιχούν στο τι "σωστά" δίνει το *Addr*, ενώ το chip τα βλέπει αλλιώς στα pins A1, A0. Σχεδιάστε τώρα τις 4 πρώτες λέξεις της μνήμης, και γράψτε μέσα σε καθεμία τα περιεχόμενά της· είναι "ανακατεμένα"; Στο ίδιο αυτό κύκλωμα, με τα ανάποδα ποδαράκια 9 και 10, διαβάζουμε από τις διευθύνσεις 0, 1, 2, και 3· και πάλι, οι διευθύνσεις αυτές αντιστοιχούν στο τι δίνει το *Addr*, ενώ το chip τα βλέπει αλλιώς στα pins A1, A0. Τι δεδομένα διαβάσαμε από τη διεύθυνση 0, τι από την 1, τι από την 2, τι από την 3; Μοιάζουν λάθος ή σωστά; Γιατί; Τι συμπέρασμα βγάζετε;

