



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Διερευνώντας την αρχιτεκτονική του Game Boy: Τεχνική ανάλυση και
ανάπτυξη εξομοιωτή Game Boy για συσκευές Android**

**Ρόμπολας Διονύσιος-Αντώνιος
Α.Μ. 71345636**

Εισηγητής: Χρήστος Τρούσσας, Επίκουρος Καθηγητής

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Διερευνώντας την αρχιτεκτονική του Game Boy: Τεχνική ανάλυση και ανάπτυξη εξομοιωτή
Game Boy για συσκευές Android**

**Ρόμπολας Διονύσιος-Αντώνιος
Α.Μ. 71345636**

Εισηγητής: Χρήστος Τρούσσας, Επ. Καθηγητής

Εξεταστική Επιτροπή:

Χρήστος Τρούσσας, Επ. Καθηγητής

Παναγιώτα Τσελέντη, ΕΔΙΠ

Ακριβή Κρούσκα, Μεταδιδακτορική Ερευνήτρια

Ημερομηνία εξέτασης: 13/07/2023

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Ρόμπολας Διονύσιος-Αντώνιος του Ιωάννη, με αριθμό μητρώου 71345636 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε δύο ενδιαφέροντα γνωστικά αντικείμενα, όπως αυτά της ανάπτυξης εφαρμογών και της αρχιτεκτονικής υπολογιστών. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου κ. Χρήστος Τρούσσας, τον οποίο θα ήθελα να ευχαριστήσω.

Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συμπαράσταση κατά τη διάρκεια των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Το Game Boy είναι μια φορητή συσκευή παιχνιδιών που κυκλοφόρησε από τη Nintendo το 1989 και πλέον θεωρείται από τις πιο δημοφιλείς και σημαντικές κονσόλες βιντεοπαιχνιδιών όλων των εποχών. Η επιτυχία του Game Boy οφειλόταν σε μεγάλο βαθμό στον απλό αλλά στιβαρό σχεδιασμό του υλικού του, τη μεγάλη διάρκεια ζωής της μπαταρίας του, την καινοτομία της φορητότητάς του, και την τεράστια βιβλιοθήκη παιχνιδιών του που εκτείνεται σε διάφορα είδη και απευθύνεται σε όλες τις ηλικίες. Ωστόσο, παρά τη δημοτικότητα και τη μακροζωία του, οι τεχνικές προδιαγραφές και οι εσωτερικές λειτουργίες του Game Boy δεν είναι ευρέως γνωστές ή κατανοητές από το ευρύ κοινό ή ακόμα και από πολλούς προγραμματιστές παιχνιδιών Game Boy. Η παρούσα διπλωματική εργασία αποτελείται από δύο μέρη. Το πρώτο μέρος περιέχει ένα θεωρητικό πλαίσιο αφιερωμένο στην ολοκληρωμένη και λεπτομερή ανάλυση της αρχιτεκτονικής του Game Boy, συμπεριλαμβανομένων των δυνατοτήτων του επεξεργαστή και της μονάδας επεξεργασίας εικόνας, τη δομή της μνήμης του, του τρόπου αλληλεπίδρασης του χρήστη με το υλικό και τον τρόπο επικοινωνίας του με άλλες συσκευές Game Boy. Το δεύτερο μέρος χρησιμοποιεί τις γνώσεις που αποκτήθηκαν από το πρώτο μέρος της εργασίας, εφαρμόζοντάς τις στα πλαίσια ανάπτυξης μίας μοντέρνας εφαρμογής εξομοιωτή Game Boy για έξυπνες συσκευές Android, υλοποιημένη σε γλώσσα προγραμματισμού Kotlin. Η συγκεκριμένη εφαρμογή επιτρέπει στους χρήστες να φορτώσουν και να εκτελέσουν αρχεία παιχνιδιών Game Boy (ROM files) στην εφαρμογή, και υποστηρίζει επιπλέον λειτουργίες όπως αποθήκευση κατάστασης παιχνιδιού (save states), κλιμάκωση της ανάλυσης (resolution scaling), και παραμετροποίηση της χαρτογράφησης της εισόδου του χειριστηρίου (controller mapping). Με αυτές τις συνεισφορές, η παρούσα διπλωματική εργασία αποσκοπεί στη συμβολή της διατήρησης και της εκτίμησης της κληρονομιάς του Game Boy προσφέροντας στους χρήστες έναν άμεσο τρόπο εκτέλεσης των παιχνιδιών του, καθώς και να αποτελέσει μια αξιοσημείωτη αναφορά ως οδηγός για μελλοντική έρευνα και ανάπτυξη στον τομέα της εξομοίωσης υπολογιστικών συστημάτων.

ABSTRACT

The Game Boy is a portable gaming device released by Nintendo in 1989 and is now considered one of the most popular and influential video game consoles of all time. The Game Boy's success was largely due to its simple yet robust hardware design, long battery life, portability, and a vast genre-spanning game library for all ages. However, despite its popularity and longevity, the technical specifications and inner workings of the Game Boy are not widely known or understood by the general public or even by many Game Boy game developers. This thesis consists of two parts. The first part contains a theoretical framework dedicated to the comprehensive and detailed analysis of the Game Boy architecture, including the capabilities of its processor and picture processing unit, its memory structure, the way the user interacts with the hardware, and how it communicates with other Game Boy devices. The second part uses the knowledge gained from the thesis' first part, applying it in the context of developing a modern Game Boy emulator application for smart Android devices, implemented using the Kotlin programming language. This application allows users to load and run Game Boy game files (ROM files) in the application, and supports additional features such as save states, resolution scaling and controller mapping. With the aforementioned parts, this thesis aims to contribute to the preservation and appreciation of the Game Boy's legacy by offering users a convenient way to play its games, as well as being a notable reference for future research and development in the field of computer system emulation.

Πίνακας περιεχομένων

Κατάλογος σχημάτων	10
Κατάλογος πινάκων	11
1 Εισαγωγή.....	12
1.1 Ιστορική αναδρομή	12
1.2 Κίνητρα επιλογής του θέματος	13
1.3 Συνεισφορά διπλωματικής εργασίας.....	14
1.4 Περιγραφή θέματος διπλωματικής.....	15
2 Θεωρητικό υπόβαθρο.....	16
3 Μεθοδολογία έρευνας.....	18
4 Ανάλυση τεχνικών χαρακτηριστικών του Game Boy.....	20
4.1 Αναλυτική παρουσίαση μνήμης του Game Boy	20
4.1.1 Παρουσίαση χάρτη μνήμης.....	21
4.1.2 Ανάλυση περιοχών τράπεζας ROM (Read Only Memory) και εναλλασσόμενης τράπεζας ROM (Switchable ROM)	21
4.1.3 Ανάλυση περιεχομένων Video RAM.....	24
4.1.4 Περίπτωση χρήσης εξωτερικής RAM (External RAM)	26
4.1.5 Περίπτωση χρήσης RAM εργασίας (WRAM).....	27
4.1.6 Λόγος ύπαρξης αντικατοπτρισμού RAM εργασίας (WRAM Echo)	27
4.1.7 Ανάλυση μνήμης χαρακτηριστικών γραφικών στοιχείων (Sprite Attribute Memory - OAM)	27
4.1.8 Παρουσίαση καταχωρητών εισόδου/εξόδου (Input/Output)	29
4.1.9 Περίπτωση χρήσης υψηλής RAM (HRAM).....	29
4.1.10 Παρουσίαση τρόπου λειτουργίας του καταχωρητή ενεργοποίησης διακοπών (IE).....	29
4.2 Ανάλυση λειτουργίας επεξεργαστή (CPU).....	30
4.2.1 Τεχνικά χαρακτηριστικά επεξεργαστή.....	30
4.2.2 Παρουσίαση των βασικών καταχωρητών του επεξεργαστή	30
4.2.3 Αναλυτική παρουσίαση του σετ εντολών του Game Boy (Instruction Set)	34

4.2.4	Παρουσίαση του τρόπου λειτουργίας των διακοπών στο Game Boy (Interrupts).....	37
4.2.5	Αναφορά στη λειτουργία Game Boy Color (CGB)	39
4.3	Ανάλυση λειτουργίας υποσυστήματος μονάδας επεξεργασίας εικόνας (Picture Processing Unit - PPU)	39
4.3.1	Συνοπτική περιγραφή λειτουργιών της μονάδας επεξεργασίας εικόνας.....	39
4.3.2	Ανάλυση λειτουργιών καταχωρητών της μονάδας επεξεργασίας εικόνας	40
4.3.3	Ανάλυση κύκλου λειτουργίας της μονάδας επεξεργασίας εικόνας	45
4.4	Ανάλυση λειτουργίας της εισόδου χρήστη	49
5	Υλοποίηση της εφαρμογής εξομοιωτή Game Boy σε περιβάλλον Android Studio	51
5.1	Εγκατάσταση απαραίτητων εργαλείων για την ανάπτυξη του εξομοιωτή	51
5.2	Δημιουργία διεπαφής χρήστη (User Interface)	52
5.2.1	Ανάλυση απαιτήσεων διεπαφής χρήστη	52
5.2.2	Παρουσίαση τρόπου δημιουργίας της διεπαφής χρήστη στο Android Studio.....	54
5.2.3	Παρουσίαση προβλημάτων κατά τη δημιουργία της διεπαφής χρήστη.....	55
5.3	Σύνοψη δομής κώδικα Kotlin έργου.....	57
5.3.1	Σύνοψη κώδικα υποσυστήματος επεξεργαστή	58
5.3.2	Σύνοψη κώδικα υποσυστήματος μονάδας επεξεργασίας εικόνας.....	60
5.3.3	Σύνοψη κώδικα βοηθητικών συναρτήσεων	63
5.3.4	Σύνοψη κώδικα υλοποίησης εφαρμογής Android	64
5.4	Δυσκολίες ανάπτυξης εφαρμογής και τρόποι αντιμετώπισής τους	66
6	Αξιολόγηση εφαρμογής	69
6.1	Αξιολόγηση απόδοσης εξομοιωτή με χρήση δοκιμαστικών προγραμμάτων Blargg (Blargg's Test ROMs).....	69
7	Συμπεράσματα και μελλοντικές επεκτάσεις	71
7.1	Συμπεράσματα	71
7.2	Μελλοντικές επεκτάσεις	71
7.2.1	Προσθήκη ήχου στον εξομοιωτή	72
7.2.2	Προσθήκη λειτουργίας Game Boy Color (CGB).....	72

7.2.3	Αναδόμηση διεπαφής χρήστη για συμβατότητα με μεγαλύτερο αριθμό συσκευών Android	73
7.2.4	Επίλυση περιπτώσεων ασύμβατων παιχνιδιών	73
8	Βιβλιογραφία	74
9	Παράρτημα.....	77
9.1	Σετ εντολών του επεξεργαστή του Game Boy (Game Boy Instruction Set).....	77

Κατάλογος σχημάτων

Σχήμα 1 - Χάρτης Μνήμης Game Boy	21
Σχήμα 2 – Παράδειγμα περιεχομένων VRAM [21].....	25
Σχήμα 3 - Παράδειγμα περιεχομένων χαρτη παρασκηνίου. Η οθόνη του Game Boy συμβολίζεται από το κίτρινο τετράγωνο το οποίο κινείται με βάση τις τιμές των καταχωρητών SCY και SCX. [21]	26
Σχήμα 4 - Αναπαράσταση πρόσθεσης που ενεργοποιεί τη σημαία Half Carry	32
Σχήμα 5 - Παράδειγμα εντολής του Game Boy [32]	34
Σχήμα 6 - Τοποθέτηση οθόνης στον χάρτη παρασκηνίου για SCY=5, SCX=50	43
Σχήμα 7 - Παράδειγμα μετατροπής τιμών του χάρτη παρασκηνίου σε χρώματα.....	45
Σχήμα 8 - Σειρά καταστάσεων λειτουργίας μονάδας επεξεργασίας εικόνας (PPU) για την εμφάνιση ενός καρέ.....	46
Σχήμα 9 - Βήματα κατάστασης λειτουργίας εμφάνισης εικονοστοιχείων [40]	47
Σχήμα 10 - Η κεντρική οθόνη της εφαρμογής.....	53
Σχήμα 11 - Παρουσίαση τμημάτων της διεπαφής χρήστη της εφαρμογής.....	55
Σχήμα 12 - Οι εικόνες των κουμπιών της εφαρμογής του εξομοιωτή	56
Σχήμα 13 - Η γραμματοσειρά Luckiest Guy που χρησιμοποιήθηκε στην εφαρμογή.....	57
Σχήμα 14 - Δομή των αρχείων κώδικα του εξομοιωτή.....	58
Σχήμα 15 - Διάγραμμα UML της κλάσης Cpu	59
Σχήμα 16 - Διάγραμμα UML της κλάσης Ppu.....	61
Σχήμα 17- Παράδειγμα επιτυχημένης δοκιμής.....	69
Σχήμα 18 - Παράδειγμα αποτυχημένης δοκιμής.....	70
Σχήμα 19 - Το ολοκληρωμένο σετ εντολών του Game Boy [44]	77

Κατάλογος πινάκων

Πίνακας 1 - Συνδυασμοί Memory Bank Controllers και οι κωδικοί τους [20].....	24
Πίνακας 2 - Αποκωδικοποίηση τέταρτου byte κάθε γραφικού στοιχείου στη μνήμη χαρακτηριστικών γραφικών στοιχείων	28
Πίνακας 3 - Τύποι διακοπών στον καταχωρητή IE [29].....	30
Πίνακας 4 - Περιεχόμενα καταχωρητή F.....	30
Πίνακας 5 - Διακοπές που χρησιμοποιούνται στο Game Boy [33]	38
Πίνακας 6 - Ρυθμίσεις καταχωρητή ελέγχου LCD [35].....	41
Πίνακας 7 - Ρυθμίσεις καταχωρητή ελέγχου LCD [37].....	42
Πίνακας 8 - Τρόπος ανάθεσης χρώματος στα εικονοστοιχεία από καταχωρητή δεδομένων παλέτας παρασκηνίου [38].....	44
Πίνακας 9 - Χρώματα που κατανέμει ο καταχωρητής δεδομένων παλέτας παρασκηνίου [38]	44
Πίνακας 10 - Καταστάσεις λειτουργίας της μονάδας επεξεργασίας εικόνας [37].....	46
Πίνακας 11 - Τιμές bits καταχωρητή εισόδου χρήστη [26].....	50
Πίνακας 12 - Αξιολόγηση λειτουργίας υποσυστήματος επεξεργαστή	70

1 Εισαγωγή

1.1 Ιστορική αναδρομή

Το 1979, ο σχεδιαστής του Game Boy και εργαζόμενος της Nintendo, Gunpei Yokoi, ταξιδεύοντας με τρένο, παρατήρησε έναν επιχειρηματία να περνάει την ώρα του ασχολούμενος με μία αριθμομηχανή. Η συγκεκριμένη σκηνή του έδωσε την ιδέα για τη δημιουργία μιας συσκευής ρολογιού, η οποία θα υποστήριζε παράλληλα και λειτουργίες παιχνιδιών. Έναν χρόνο αργότερα, το 1980, η Nintendo κυκλοφόρησε στην αγορά τη σειρά ηλεκτρονικών παιχνιδιών Game & Watch [1], η οποία χρησιμοποιούσε hardware από αριθμομηχανές, και σε συνδυασμό με μία LCD οθόνη και τα απαραίτητα πλήκτρα, επέτρεπε στο χρήστη να παίζει ένα ηλεκτρονικό παιχνίδι, δείχνοντάς του παράλληλα την ώρα στη γωνία της οθόνης. Κατά το σχεδιασμό της επόμενης φορητής κονσόλας της Nintendo, ο Yokoi, λαμβάνοντας υπόψιν τη φορητότητα των συσκευών Game & Watch, και σε συνδυασμό με τα χαρακτηριστικά της πρωτοποριακής κονσόλας NES (Nintendo Entertainment System) της Nintendo, εφηύρε το Game Boy [2].

Στις 21 Απριλίου του 1989, το Game Boy κυκλοφόρησε στην αγορά [3]. Παρά το γεγονός πως το Game Boy ήταν τεχνολογικά υποδεέστερο από τις υπόλοιπες φορητές κονσόλες της περιόδου, συγκεκριμένα το Lynx της Atari και το Game Gear της Sega, κατάφερε να κυριαρχήσει στις πωλήσεις [4], με 118,69 εκατομμύρια πωλήσεις παγκοσμίως [5]. Παράγοντες οι οποίοι συνέβαλαν στην κυριαρχία του στην αγορά των φορητών κονσολών, ήταν η χαμηλή τιμή του, η οποία ήταν η μισή αυτής του ανταγωνιστικού Atari Lynx, το μεγάλο κεφάλαιο που επενδύθηκε για σκοπούς μάρκετινγκ από τη Nintendo, και το πλήθος συμφωνιών που είχε συνάψει η Nintendo με εξωτερικούς συνεργάτες, χάρη στις οποίες αναπτύχθηκαν ηλεκτρονικά παιχνίδια σε όλο το εύρος των ειδών [6].

Δημοφιλή παιχνίδια για το Game Boy υπήρξαν η μεταφορά του ήδη δημοφιλούς Tetris στην κονσόλα, οι σειρές Pokemon, The Legend of Zelda, Kirby, καθώς και η γνωστή σειρά Super Mario. Συγκεκριμένα, τα παιχνίδια Pokemon Red/Green/Blue/Yellow τα οποία κυκλοφόρησαν στις 27 Φεβρουαρίου του 1996, πούλησαν 31,38 εκατομμύρια αντίτυπα [7] και το Tetris 35 εκατομμύρια αντίτυπα [8]. Συγκριτικά, το Sonic the Hedgehog 2 το οποίο ήταν ο κορυφαίος τίτλος για το Game Gear της Sega, πραγματοποίησε μόλις 400,000 πωλήσεις [9].

Ο επεξεργαστής του Game Boy είναι ο Sharp SM83, ο οποίος ήταν βασισμένος στον 8 Bit επεξεργαστή Zilog Z80 με συχνότητα λειτουργίας 4.19 MHz, που κυκλοφόρησε το 1976 και είχε παρόμοια αρχιτεκτονική με τον 8080 της Intel [10]. Η οθόνη του ήταν τεχνολογίας dot-matrix LCD (liquid-crystal display) 2,6 ιντσών και ανάλυσης 160x144 pixels [11], και υποστήριζε μονοχρωματική παλέτα 4 αποχρώσεων του πράσινου χρώματος (ανοιχτό έως σκούρο πράσινο), των οποίων η επιλογή γινόταν με χρήση 2 bits. Είχε ένα μονό ηχείο του οποίου η ένταση ρυθμιζόταν από μία ροδέλα στο δεξί μέρος της συσκευής, και διέθετε 5 κουμπιά για τον έλεγχο των παιχνιδιών, τα οποία ήταν τα A, B, Start,

Select, και τα βελάκια τεσσάρων κατευθύνσεων. Εξωτερικά, είχε το σχήμα ενός ορθογωνίου παραλληλογράμμου, με μόνη διαφορά την κάτω δεξιά γωνία, η οποία ήταν στρογγυλή. Τα παιχνίδια του Game Boy ερχόντουσαν σε μορφή κασετών (cartridges), οι οποίες τοποθετούνταν στο πίσω μέρος της συσκευής.

Επίσης, το Game Boy υποστήριζε τη χρήση διάφορων περιφερειακών, όπως κάμερα η οποία πραγματοποιούσε λήψη φωτογραφιών απεικονισμένων στην παλέτα του Game Boy, εκτυπωτή ο οποίος εκτύπωνε εικόνες με χρήση ειδικού χαρτιού ευαίσθητου στη θερμότητα, καλωδίου σύνδεσης το οποίο επέτρεπε την εκτέλεση παιχνιδιών πολλών παικτών (multiplayer), και ακόμα και σόναρ, το οποίο χρησιμοποιούταν στα πλαίσια ψαρέματος για τον εντοπισμό ψαριών από την οθόνη του Game Boy [12].

Το Game Boy κυκλοφόρησε στην τιμή των 89,99 δολαρίων στην Αμερική, και η υψηλή δημοτικότητά του οδήγησε στη συνέχιση της παραγωγής του μέχρι και το έτος 2003, 14 χρόνια μετά από την κυκλοφορία του. Ο διάδοχός του ήταν το Game Boy Color, το οποίο κυκλοφόρησε στις 21 Οκτωβρίου του 1998, και ήταν ουσιαστικά μία αναβάθμιση του αρχικού Game Boy, με υποστήριξη μεγαλύτερης παλέτας 16 χρωμάτων, και υψηλότερης συχνότητας επεξεργαστή. [13]

1.2 Κίνητρα επιλογής του θέματος

Η επιλογή του θέματος της παρούσας διπλωματικής εργασίας πραγματοποιήθηκε για δύο λόγους. Ο πρώτος ήταν το αντίκτυπο το οποίο είχαν οι κονσόλες της Nintendo στην παιδική μου ηλικία, όπως και σε εκατομμύρια άλλους ανθρώπους. Η συχνή ασχολία μου με τα βιντεοπαιχνίδια, τα οποία προσέφεραν ατελείωτες ώρες διασκέδασης, μου προκάλεσαν με την πάροδο των χρόνων την απορία σχετικά με την πολυπλοκότητα του εγχειρήματος της δημιουργίας ενός υπολογιστικού συστήματος, όπως μία κονσόλα, από το μηδέν. Οι προγραμματιστικές γνώσεις αλλά και οι γνώσεις αρχιτεκτονικής υπολογιστών που έλαβα κατά τη διάρκεια της ακαδημαϊκής μου πορείας, μου έδωσαν τα εφόδια τα οποία ήταν απαραίτητα για την επίλυση της συγκεκριμένης απορίας.

Ο δεύτερος λόγος σχετίζεται με την υψηλή δυσκολία ανάπτυξης ενός εξομοιωτή, η οποία είναι ένα απαιτητικό προγραμματιστικό εγχείρημα του οποίου η ολοκλήρωση θα επιφέρει σημαντικές προγραμματιστικές γνώσεις, αλλά και εξειδικευμένες γνώσεις υπολογιστικών συστημάτων. Παράλληλα, πέρα από την ανάπτυξη των υπαρχόντων δεξιοτήτων και την εκμάθηση νέων, η ολοκλήρωση ενός τέτοιου έργου προσφέρει μεγάλο βαθμό ικανοποίησης σε προσωπικό επίπεδο.

Όσον αφορά την απόκτηση νέων δεξιοτήτων, η επιλογή του Android ως πλατφόρμα εκτέλεσης του εξομοιωτή έγινε με σκοπό την εκμάθηση της μοντέρνας γλώσσας προγραμματισμού Kotlin, και την επέκταση των γνώσεών μου σχετικά με τον σχεδιασμό διεπαφής χρήστη. Επιπρόσθετα, η φορητότητα που παρέχουν οι κινητές συσκευές Android προσφέρει στους χρήστες μία εμπειρία αντίστοιχη με εκείνη του αυθεντικού Game Boy.

Η επιλογή του Game Boy έγινε λόγω του μεσαίου επιπέδου δυσκολίας ανάλυσης λειτουργίας της συγκεκριμένης συσκευής, καθώς είναι κονσόλα η οποία έχει δημιουργηθεί εδώ και δεκαετίες, πριν την ταχεία αύξηση της πολυπλοκότητας των επεξεργαστών και των γραφικών υπολογιστών, το οποίο καθιστά εφικτή τη δημιουργία ενός εξομοιωτή από έναν μόνο άνθρωπο. Ταυτόχρονα όμως δεν υπάρχουν επίσημα έγγραφα από τη Nintendo που να αναλύουν τη δομή της συγκεκριμένης συσκευής, οπότε η εύρεση σχετικών πληροφοριών είναι δύσκολη και χρονοβόρα, καθώς περιλαμβάνει την αναζήτηση εγγράφων και δημοσιεύσεων σε χώρους συζητήσεων που έχουν γραφτεί με πληροφορίες που προήλθαν από διαδικασία αντίστροφης μηχανικής (reverse engineering) από προγραμματιστές που έχουν υλοποιήσει παρόμοια εγχειρήματα. Συνεπώς, υπάρχει και εδώ το κίνητρο της προσωπικής ικανοποίησης από τη συνεισφορά που προκύπτει από τη συγκέντρωση όλων αυτών των χρήσιμων, αλλά διάσπαρτων πληροφοριών σε ένα μέρος.

1.3 Συνεισφορά διπλωματικής εργασίας

Η συνεισφορά της παρούσας διπλωματικής εργασίας είναι πολύπλευρη. Αρχικά, η ολοκλήρωση του έργου αποτελεί συνεισφορά στον τομέα της διασκέδασης, καθώς ο τελικός σκοπός του είναι η δημιουργία μίας εφαρμογής Android η οποία παρέχει στον χρήστη μία πιστή αναπαράσταση της εμπειρίας που προσέφερε η φορητή κονσόλα Game Boy. Καθώς υπάρχουν ήδη δημοσιευμένες εφαρμογές εξομοιωτών Game Boy στην πλατφόρμα Android, η συγκεκριμένη εφαρμογή διαφοροποιείται από τις υπόλοιπες εστιάζοντας στην αμεσότητα, την παροχή λειτουργιών οι οποίες διευκολύνουν τον χρήστη, και στον καλαίσθητο σχεδιασμό εμπνευσμένο από παιχνίδια της εποχής.

Αναφορικά με την παροχή λειτουργιών, από την εφαρμογή υποστηρίζονται περαιτέρω λειτουργίες όπως η υποστήριξη δημιουργίας καταστάσεων αποθήκευσης και φόρτωσης (save/load states) για άμεση συνέχεια παιχνιδιού από το τελευταίο σημείο όπου σταμάτησε ο χρήστης. Επιπρόσθετα, δίνεται έμφαση στην ανάπτυξη μοντέρνας διεπαφής χρήστη, η οποία είναι παραμετροποιήσιμη από τον χρήστη ανάλογα με τις προτιμήσεις του.

Πέρα από τον τομέα της διασκέδασης, η συγκεκριμένη διπλωματική αποσκοπεί επίσης και στη συνεισφορά στη γνώση που υπάρχει όσον αφορά τα τεχνικά χαρακτηριστικά και την εσωτερική λειτουργία του Game Boy, συλλέγοντας τις διάσπαρτες πληροφορίες που βρίσκονται στο διαδίκτυο και παραθέτοντάς τις συγκεντρωμένες σε έναν οδηγό, ο οποίος καταγράφει αναλυτικά τα εσωτερικά συστήματα του Game Boy αλλά και τη διαδικασία ανάπτυξης ενός εξομοιωτή Game Boy. Επομένως, τα περιεχόμενα της συγκεκριμένης διπλωματικής είναι πιθανό να αποδειχτούν χρήσιμα σε προγραμματιστές που αναλαμβάνουν την ανάπτυξη ενός παρόμοιου έργου σχετικού με εξομοιωτές, ή σε άτομα τα οποία θέλουν να εμβαθύνουν τις γνώσεις τους σχετικά με τον τρόπο λειτουργίας του Game Boy.

Τέλος, αξιοσημείωτη είναι η συνεισφορά η οποία δύναται να παρέχει η παρούσα διπλωματική εργασία στην εκμάθηση των βασικών εννοιών της επιστήμης των υπολογιστών από μαθητές ή

ανθρώπους οι οποίοι επιθυμούν να αποκτήσουν γνώσεις πάνω στο αντικείμενο. Μέσω της μελέτης του κεφαλαίου της τεχνικής ανάλυσης αλλά και του κώδικα της εφαρμογής ενός εξομοιωτή, ο αναγνώστης έχει τη δυνατότητα να κατανοήσει πλήρως τη λειτουργία ενός επεξεργαστή και τον τρόπο αλληλοεπίδρασής του με τα υπόλοιπα μέρη του υπολογιστικού συστήματος, τον τρόπο με τον οποίο αποτυπώνονται τα γραφικά στην οθόνη, καθώς και να αποκτήσει προγραμματιστικές γνώσεις σε γλώσσα μηχανής.

1.4 Περιγραφή θέματος διπλωματικής

Η παρούσα διπλωματική εργασία αποσκοπεί στην ανάλυση των λειτουργιών των υποσυστημάτων που απαρτίζουν το Game Boy, η γνώση των οποίων είναι απαραίτητη για την ανάπτυξη του εξομοιωτή. Επίσης, παρουσιάζεται αναλυτικά η διαδικασία ανάπτυξης του εξομοιωτή με χρήση της γλώσσας προγραμματισμού Kotlin σε περιβάλλον Android Studio, με όλα τα βήματα που πραγματοποιήθηκαν για την ολοκλήρωση του έργου, καθώς και δυσκολίες που παρουσιάστηκαν στην πορεία, μαζί με τρόπους αντιμετώπισής τους. Τέλος, παρατίθεται αξιολόγηση της εφαρμογής από χρήστες αλλά και πιθανές μελλοντικές επεκτάσεις της.

2 Θεωρητικό υπόβαθρο

Για την καλύτερη κατανόηση από τον αναγνώστη, του θέματος με το οποίο ασχολείται η παρούσα διπλωματική εργασία, κρίνεται απαραίτητος ο ορισμός ενός σχετικού θεωρητικού υποβάθρου. Η παρούσα διπλωματική εργασία σε ορισμένα σημεία της τείνει να εμβαθύνει σε τεχνικά μέρη της μηχανικής υπολογιστών, τα οποία είναι απαιτούμενα για την κατανόηση της λειτουργίας του συστήματος το οποίο μελετάται και την επιτυχή ολοκλήρωση του έργου του εξομοιωτή. Για αυτό τον λόγο, στο παρόν κεφάλαιο θα παρουσιαστούν ορισμένες έννοιες με τις οποίες ο αναγνώστης είναι επιθυμητό να έχει ήδη μια επαφή, με σκοπό την υποβοήθηση της κατανόησης των τεχνικών κομματιών της παρούσας διπλωματικής εργασίας.

Ειδικότερα, κρίνεται απαραίτητο να αναλυθεί η έννοια της εξομοίωσης (emulation) ενός συστήματος. Η εξομοίωση ορίζεται ως η αναπαράσταση των διαδικασιών ενός συστήματος ή φαινομένου, με χρήση ενός υπολογιστικού συστήματος. Με απλά λόγια, μέσω της διαδικασίας της εξομοίωσης ενός συστήματος, επιχειρούμε να εκτελέσουμε επιτυχώς τις λειτουργίες του σε ένα άλλο σύστημα.

Μία παρεμφερή έννοια με αυτήν της εξομοίωσης, είναι η έννοια της προσομοίωσης (simulation). Οι δύο αυτές έννοιες τείνουν να χρησιμοποιούνται αλληλένδετα, φαινόμενο το οποίο είναι λανθασμένο, καθώς έχουν διαφορετικό ορισμό. Ενώ ο ορισμός της προσομοίωσης είναι η εκτέλεση ενός μοντέλου το οποίο εκπροσωπεί ένα σύστημα, ο ορισμός της εξομοίωσης είναι η εκτέλεση των πραγματικών διαδικασιών οι οποίες αποτελούν το σύστημα. [14] Ουσιαστικά, με την προσομοίωση δεν μας ενδιαφέρει ο τρόπος με τον οποίο παρέχεται η έξοδος του συστήματος, αρκεί αυτή να είναι πιστή στην έξοδο του αρχικού συστήματος. Στην περίπτωση της εξομοίωσης, ο τρόπος με τον οποίο παρέχεται η έξοδος πρέπει να είναι ίδιος με τον τρόπο όπου το πραγματικό σύστημα παρέχει την έξοδό του, και σαφώς η συγκεκριμένη έξοδος να είναι ίδια με αυτή του αρχικού συστήματος.

Επομένως, για την επιτυχή εξομοίωση του υπολογιστικού συστήματος Game Boy, κρίνεται απαραίτητη η έρευνα των υποσυστημάτων τα οποία το απαρτίζουν, καθώς ήταν σημαντικό τα υποσυστήματα αυτά να δημιουργηθούν με υψηλή ακρίβεια για την ανάπτυξη του εξομοιωτή. Από αυτή την ανάγκη πηγάζει η δημιουργία του πρώτου μέρους της παρούσας διπλωματικής εργασίας, το οποίο αφορά την τεχνική ανάλυση των υποσυστημάτων του Game Boy.

Τα υποσυστήματα με των οποίων την ανάπτυξη θα ασχοληθεί η παρούσα εργασία, είναι αυτά του επεξεργαστή, της μονάδας επεξεργασίας, αλλά και της μνήμης που χρησιμοποιείται από το Game Boy για την εκτέλεση προγραμμάτων. Τα υποσυστήματα αυτά βρίσκονται σε κάθε υπολογιστικό σύστημα και είναι κρίσιμα για τη λειτουργία του. Στο κεφάλαιο της ανάλυσης των τεχνικών χαρακτηριστικών του Game Boy, θα αναλυθούν πλήρως οι λειτουργίες τις οποίες εκτελεί το κάθε υποσύστημα, καθώς και ο τρόπος με τον οποίο τις εκτελεί.

Η μνήμη ενός υπολογιστικού συστήματος αποτελεί το χώρο στον οποίο αποθηκεύονται όλες οι εντολές και τα δεδομένα τα οποία χρησιμοποιούνται από το σύστημα για τη λειτουργία του. Συνήθως, η μνήμη αναπαρίσταται ως ένας πίνακας κελιών (χάρτης μνήμης), και στο κάθε κελί περιέχονται οι εντολές και τα δεδομένα σε μορφή bits, τα οποία μπορούν να διαβαστούν με χρήση της διεύθυνσης μνήμης του κελιού. Το Game Boy είναι σύστημα 8-bit, δηλαδή σε κάθε κελί της μνήμης μπορούν να αποθηκευτούν 8 bits.

Ο επεξεργαστής (CPU – Central Processing Unit), είναι το υποσύστημα της συσκευής το οποίο είναι υπεύθυνο για την εκτέλεση όλων των εντολών οι οποίες περιέχονται στη μνήμη. Αναλυτικότερα, κατά τη λειτουργία του, ο επεξεργαστής διαβάζει τα περιεχόμενα της διεύθυνσης μνήμης στην οποία βρίσκεται το πρόγραμμα, και ανάλογα με την τιμή που περιέχεται εκτελεί την ανάλογη πράξη. Τέτοιες πράξεις μπορεί να είναι πρόσθεση, αφαίρεση και μετακίνηση δεδομένων μεταξύ άλλων. Για την εκτέλεση αυτών των πράξεων, ο επεξεργαστής χρησιμοποιεί ορισμένες θέσεις μνήμης για την προσωρινή αποθήκευση δεδομένων στο υποσύστημά του, οι οποίοι βρίσκονται εκτός της μνήμης. Αυτές οι θέσεις μνήμης ονομάζονται καταχωρητές.

Τέλος, η μονάδα επεξεργασίας εικόνας είναι υπεύθυνη για την εμφάνιση της τελικής εικόνας στην οθόνη. Εκτελείται παράλληλα με τον επεξεργαστή, με τον οποίο συγχρονίζονται με τη χρήση διακοπών, κατά τις οποίες συνοπτικά, η μονάδα επεξεργασίας εικόνας αλλάζει τα δεδομένα σε ορισμένες θέσεις μνήμης, τις οποίες ο επεξεργαστής ελέγχει συνεχώς για αλλαγές. Όταν πραγματοποιείται μία διακοπή, ο επεξεργαστής παγώνει την κανονική ροή του προγράμματος, και εκτελεί συγκεκριμένες εντολές ανάλογα με τον τύπο διακοπής.

Εν κατακλείδι, γνωρίζοντας την έννοια της εξομοίωσης, αλλά και τις γενικότερες λειτουργίες οι οποίες εκτελούνται από ένα υπολογιστικό σύστημα, στην περίπτωση της παρούσας διπλωματικής εργασίας, το Game Boy, ο αναγνώστης της παρούσας εργασίας κατέχει τα απαραίτητα εφόδια για να εμβαθύνει τις γνώσεις του μέσω του κεφαλαίου της ανάλυσης των τεχνικών χαρακτηριστικών του Game Boy.

3 Μεθοδολογία έρευνας

Η ανάπτυξη ενός εξομοιωτή αποτελεί ένα απαιτητικό έργο το οποίο απαιτεί οργάνωση μέσω της εκτέλεσης συγκεκριμένων βημάτων. Τα βήματα αυτά αποτελούνται από την αναζήτηση όλων των τεχνικών πληροφοριών σχετικών με τη λειτουργία του Game Boy, τον ορισμό των απαιτήσεων της εφαρμογής, τον σχεδιασμό της εφαρμογής, και τέλος, την αξιολόγηση της εφαρμογής.

Η αναζήτηση των τεχνικών χαρακτηριστικών είναι το πρώτο βήμα το οποίο πραγματοποιήθηκε για την ανάπτυξη του εξομοιωτή. Αποτελεί ένα χρονοβόρο βήμα που επαναλαμβάνεται συχνά και κατά τη διάρκεια των επόμενων βημάτων, καθώς είναι σύνηθες να προκύπτουν καινούργιες απορίες κατά την ανάπτυξη της εφαρμογής. Επιπλέον, η έλλειψη επίσημων εγγράφων σχετικά με τη λειτουργία και τα τεχνικά χαρακτηριστικά του Game Boy από τη Nintendo αυξάνει το βαθμό δυσκολίας του συγκεκριμένου βήματος. Η αναζήτηση πραγματοποιήθηκε μέσω του διαδικτύου και αντικείμενό της είναι όλα τα μέσα τα οποία περιέχουν πληροφορίες σχετικές με τη λειτουργία των υποσυστημάτων του Game Boy. Τέτοια μέσα μπορεί να είναι οδηγοί που έχουν συγγράψει προγραμματιστές οι οποίοι έχουν αναλάβει παρόμοια εγχειρήματα, δημοσιεύσεις σε χώρους συζητήσεων, βίντεο παρουσιάσεων σε συνέδρια αλλά και δημοσιευμένες έρευνες. Αξιοσημείωτη πηγή αποδείχτηκε η ιστοσελίδα Pan Docs [15] στην οποία φιλοξενούνται τεχνικά έγγραφα σχετικά με τη λειτουργία του Game Boy, τα οποία έχουν συγγραφεί από κοινότητα προγραμματιστών και μηχανικών ηλεκτρονικής.

Έπειτα από τη συλλογή πληροφοριών σχετικών με τη λειτουργία του Game Boy, επόμενο βήμα είναι ο ορισμός των απαιτήσεων της εφαρμογής. Σε αυτό το βήμα ορίστηκαν απαιτήσεις οι οποίες καθορίζουν τη γενική πορεία της ανάπτυξης. Παραδείγματα τέτοιων απαιτήσεων είναι η απαίτηση της ευχρηστίας της εφαρμογής, ο μοντέρνος σχεδιασμός της, η ύπαρξη πρόσθετων λειτουργιών όπως η αποθήκευση καταστάσεων παιχνιδιού (save states), αλλά και η αναπροσαρμογή της διεπαφής χρήστη από τον χρήστη. Μέσω σαφώς καθορισμένων απαιτήσεων, τα επόμενα βήματα του σχεδιασμού της εφαρμογής και της ανάπτυξής της, καθίστανται απλούστερα καθώς πραγματοποιείται αποφυγή της χρονοβόρας αναδιοργάνωσης συστημάτων κατά τη διάρκεια της συγγραφής του κώδικα.

Στο στάδιο σχεδιασμού της εφαρμογής πραγματοποιήθηκε ο διαχωρισμός σε κλάσεις των υποσυστημάτων του εξομοιωτή, όπως αυτά του επεξεργαστή, γραφικών και ήχου, και ορίστηκε ο τρόπος με τον οποίο επικοινωνούν μεταξύ τους οι συγκεκριμένες κλάσεις. Επιπρόσθετα, δημιουργήθηκε το προσχέδιο της διεπαφής χρήστη στο Android Studio, λαμβάνοντας υπόψιν τις προκαθορισμένες απαιτήσεις.

Το στάδιο της ανάπτυξης είναι το πιο χρονοβόρο και περίπλοκο στάδιο του έργου, καθώς η ολοκλήρωσή του διήρκεσε μήνες. Το συγκεκριμένο στάδιο περιέχει την υλοποίηση των υποσυστημάτων του εξομοιωτή σε γλώσσα Kotlin. Αρχικά, με τη βοήθεια των τεχνικών εγγράφων, πραγματοποιήθηκε η υλοποίηση του υποσυστήματος του επεξεργαστή. Πιο συγκεκριμένα, υλοποιήθηκαν οι 512 εντολές που αποτελούν το συνολικό σετ εντολών του επεξεργαστή του Game

Boy. Με την ολοκλήρωση της υλοποίησης των εντολών, πραγματοποιήθηκε έλεγχος του υποσυστήματος του επεξεργαστή με χρήση δοκιμαστικής κασέτας και σύγκρισης των αποτελεσμάτων με αποτελέσματα ολοκληρωμένων εξομοιωτών. Όταν η δημιουργία του υποσυστήματος του επεξεργαστή και ο έλεγχός του ολοκληρώθηκαν, ξεκίνησε η ανάπτυξη του υποσυστήματος των γραφικών. Η ανάπτυξη του υποσυστήματος των γραφικών αποδείχθηκε εξαιρετικά δύσκολη, καθώς οι διαθέσιμες πληροφορίες σχετικά με τη λειτουργία του είναι ελλιπείς. Για αυτό το λόγο, κατά τη διάρκεια της συγγραφής του κώδικα, ήταν απαραίτητες αρκετές δοκιμές για την επίτευξη ικανοποιητικού αποτελέσματος. Σε αυτό το σημείο πραγματοποιήθηκε και η σύνδεση μεταξύ του κώδικα Kotlin και της διεπαφής χρήστη, έτσι ώστε να δημιουργηθεί μία αρχική έκδοση της εφαρμογής στην οποία παρουσιάζονται τα γραφικά στην οθόνη της συσκευής Android. Παράλληλα, συνδέθηκαν τα υποσυστήματα επεξεργαστή και γραφικών μέσω συστήματος χρονισμού, έτσι ώστε να επιτευχθεί η παράλληλη λειτουργία τους, με τον ίδιο τρόπο όπως εκτελούνταν στο πραγματικό Game Boy. Τέλος, προστέθηκαν οι περαιτέρω απαιτήσεις της εφαρμογής, πέραν της βασικής που είναι η εξομοίωση.

Το τελικό στάδιο του έργου είναι το στάδιο της αξιολόγησης της εφαρμογής του εξομοιωτή. Η διαδικασία αξιολόγησης είναι ιδιαίτερα σημαντική, καθώς μέσω αυτής ορίζονται οι μελλοντικοί στόχοι και οι επεκτάσεις του έργου, και προσδιορίζεται ο βαθμός στον οποίο ικανοποιήθηκαν οι απαιτήσεις του έργου. Στο τελικό στάδιο, πραγματοποιήθηκε αξιολόγηση της ακρίβειας του εξομοιωτή, μέσω εκτέλεσης δοκιμαστικών προγραμμάτων που έχουν δημιουργηθεί για αυτόν τον σκοπό.

4 Ανάλυση τεχνικών χαρακτηριστικών του Game Boy

Στα πλαίσια της ανάπτυξης του εξομοιωτή Game Boy, το αρχικό βήμα είναι η κατανόηση του τρόπου λειτουργίας των υποσυστημάτων του, όπως αυτά της μνήμης ή του επεξεργαστή. Το παρόν κεφάλαιο αποσκοπεί στην ανάλυση του τρόπου λειτουργίας του κάθε υποσυστήματος, εμβαθύνοντας σε τεχνικές λεπτομέρειες στα σημεία στα οποία κρίνεται απαραίτητο, έτσι ώστε ο αναγνώστης να αποκτήσει την απαραίτητη γνώση για την υλοποίηση του εξομοιωτή. Με την ολοκλήρωση του κεφαλαίου, ο αναγνώστης λαμβάνει εξειδικευμένες γνώσεις όσον αφορά την εσωτερική λειτουργία των υποσυστημάτων της μνήμης, του επεξεργαστή (CPU), της μονάδας επεξεργασίας εικόνας (PPU), του ήχου, και της εισόδου χρήστη.

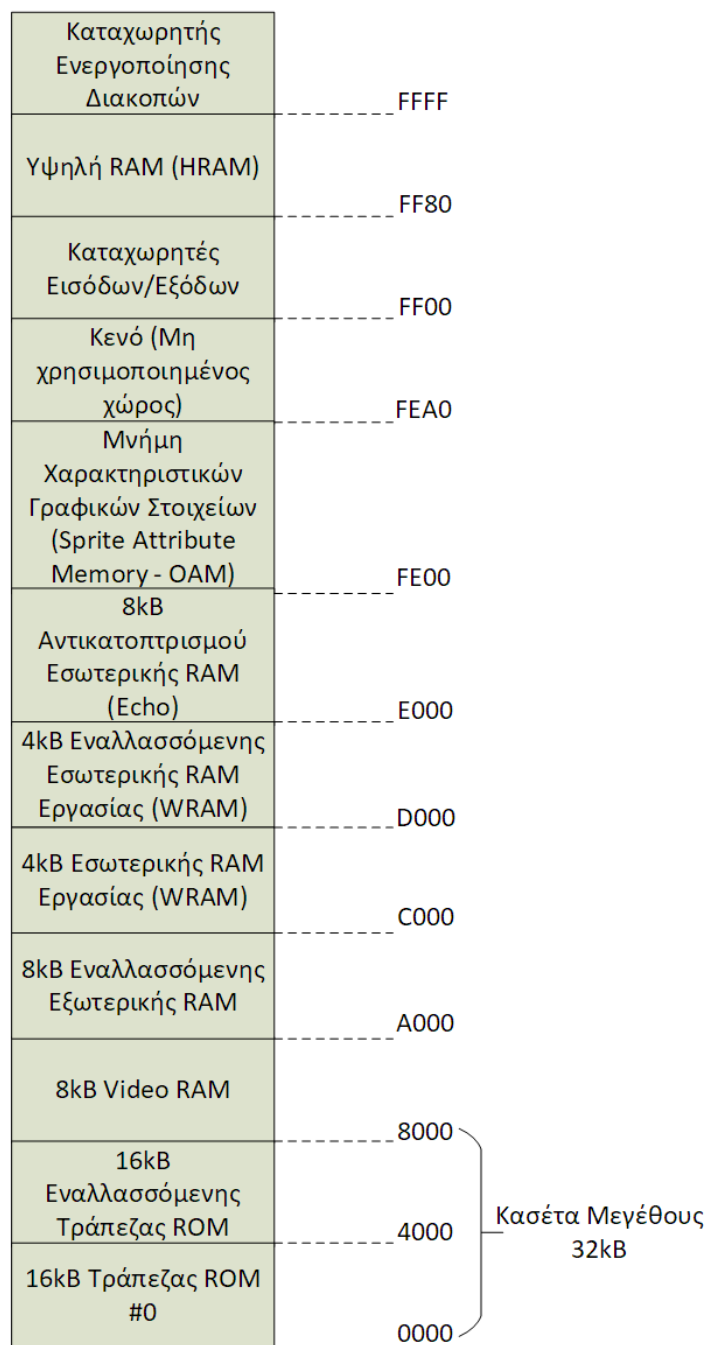
4.1 Αναλυτική παρουσίαση μνήμης του Game Boy

Αρχικά, στο παρόν κεφάλαιο το οποίο αφορά τη μνήμη του Game Boy παρουσιάζεται ο χάρτης μνήμης, με σκοπό να οπτικοποιηθεί η δομή της μνήμης του Game Boy από τον αναγνώστη. Επιπρόσθετα, παρουσιάζεται ο λόγος ύπαρξης κάθε περιοχής μνήμης, οι οποίες έπειτα αναλύονται σε τεχνικό επίπεδο. Η τεχνική ανάλυση τους παρέχει όλες τις απαραίτητες πληροφορίες για τη δημιουργία ορισμένων λειτουργιών που αναπτύχθηκαν στα πλαίσια του εξομοιωτή, καθώς και για την αποσφαλμάτωσή του.

Για παράδειγμα, είναι απαραίτητη η γνώση του εύρους περιοχών μνήμης στις οποίες αντιστοιχούν τα περιεχόμενα της κασέτας που φορτώνεται στο Game Boy. Γνωρίζοντας το συγκεκριμένο εύρος, μας δίνεται η δυνατότητα να μεταφέρουμε τα δεδομένα που περιέχει η κασέτα που διαβάζεται από τον εξομοιωτή μας στις κατάλληλες θέσεις του πίνακα που έχουμε δημιουργήσει στο πρόγραμμά μας, ο οποίος περιέχει τη μνήμη του εξομοιωτή.

Από την άλλη, ένα παράδειγμα χρήσης της γνώσης του συγκεκριμένου κεφαλαίου για λόγους αποσφαλμάτωσης, είναι πως γνωρίζοντας το εύρος μνήμης της περιοχής μνήμης της Video RAM, θα μπορούσαμε να εξάγουμε τα γραφικά στοιχεία της κάθε κασέτας σε κάποιο αρχείο με χρήση βοηθητικής συνάρτησης και να τα συγκρίνουμε με αυτά τα οποία αποτυπώνονται στην οθόνη, για να εξακριβώσουμε πως τα γραφικά αποτυπώνονται στην οθόνη με ακρίβεια.

4.1.1 Παρουσίαση χάρτη μνήμης



Σχήμα 1 - Χάρτης Μνήμης Game Boy

4.1.2 Ανάλυση περιοχών τράπεζας ROM (Read Only Memory) και εναλλασσόμενης τράπεζας ROM (Switchable ROM)

Στο παρόν υποκεφάλαιο αναλύεται η δομή και η λειτουργία των περιοχών τράπεζας ROM και εναλλασσόμενης ROM της μνήμης του Game Boy. Η κατανόηση της λειτουργίας των περιοχών αυτών καθίσταται απαραίτητη, καθώς αποτελούν βασικά δομικά στοιχεία της μνήμης του Game Boy. Πιο συγκεκριμένα, στην περιοχή ROM υπάρχει ο κώδικας ο οποίος εκτελείται κατά την εκκίνηση κάθε

προγράμματος και είναι αποθηκευμένος στην κασέτα που εισάγεται στο Game Boy, ενώ η περιοχή της εναλλασσόμενης ROM χρησιμοποιείται από προγράμματα τα οποία χρειάζονται μεγαλύτερο μέγεθος μνήμης από τα 16kB που προσφέρει η περιοχή ROM για να λειτουργήσουν, περίπτωση αρκετά συνηθισμένη. Η δεύτερη λειτουργία της περιοχής της εναλλασσόμενης ROM είναι η εναλλαγή μεταξύ τραπεζών μνήμης, στις περιπτώσεις όπου το πρόγραμμα που εκτελείται χρειάζεται ακόμα μεγαλύτερο χώρο στη μνήμη από εκείνον που παρέχουν οι περιοχές μνήμης ROM και εναλλασσόμενης ROM.

Για παράδειγμα, μία κασέτα η οποία χρησιμοποιεί 64kB για τη λειτουργία της περιέχει τα πρώτα 16kB στην περιοχή ROM, και η περιοχή της εναλλασσόμενης ROM εναλλάσσεται στα επόμενα 48kB, χωρισμένα σε 3 τράπεζες μνήμης χωρητικότητας 16kB η κάθε μία, ανάλογα με τις ανάγκες του προγράμματος. Η αλλαγή αυτή γίνεται μέσω εγγραφής σε καταχωρητές των μικροκυκλωμάτων ελεγκτών τράπεζας μνήμης (Memory Bank Controllers ή MBC) [16], η λειτουργία των οποίων αναλύεται στο υποκεφάλαιο 4.1.2.1.

4.1.2.1 Ανάλυση τρόπου λειτουργίας και είδη των ελεγκτών τραπεζών μνήμης (Memory Bank Controllers - MBC)

Υπεύθυνοι για την εναλλαγή μεταξύ των τραπεζών μνήμης στην περιοχή της εναλλασσόμενης ROM, αλλά και για την υποστήριξη της λειτουργίας περαιτέρω περιφερειακών συσκευών τα οποία είναι ενσωματωμένα στην κασέτα (όπως κάμερας και χρονομέτρου), είναι οι ελεγκτές τράπεζας μνήμης (Memory Bank Controllers - MBC). Η κατανόηση του τρόπου λειτουργίας των ελεγκτών τραπεζών μνήμης είναι απαραίτητη για την ενσωμάτωση των δυνατοτήτων που προσφέρουν στον εξομοιωτή. Πολλά από τα πιο δημοφιλή παιχνίδια του Game Boy δεν ξεκινούν εάν δεν παρέχεται η υποστήριξη των ελεγκτών τραπεζών μνήμης που περιέχουν. Για αυτό το λόγο, η έλλειψη της ενσωμάτωσής τους στον εξομοιωτή τον καθιστά μη λειτουργικό. Σκοπός του παρόντος υποκεφαλαίου είναι να περιγράψει τους τύπους ελεγκτών τραπεζών μνήμης που χρησιμοποιούνται στις κασέτες Game Boy με απώτερο σκοπό την επιτυχή προσθήκη της συγκεκριμένης λειτουργίας στην εφαρμογή του εξομοιωτή.

Πιο συγκεκριμένα, οι ελεγκτές τράπεζας μνήμης είναι μικροκυκλώματα υπεύθυνα για την εναλλαγή των δεδομένων της εναλλασσόμενης ROM (θέσεις μνήμης 0x4000 έως 0x8000), τα οποία περιέχονται μέσα στις κασέτες. Υπάρχουν διάφοροι τύποι ελεγκτών τραπεζών μνήμης, οι οποίοι επιλέγονται από τους προγραμματιστές του κάθε παιχνιδιού. Ο τύπος ελεγκτή ο οποίος περιέχεται στην κάθε κασέτα, αναφέρεται κωδικοποιημένος ως byte στη θέση μνήμης 0x147 της ROM.

Με το byte 0x00 στη θέση μνήμης 0x147 προσδιορίζεται η έλλειψη χρήσης ελεγκτή. Ορισμένα παιχνίδια δεν πραγματοποιούν εναλλαγές στην περιοχή της εναλλασσόμενης ROM, καθώς τα δεδομένα που χρειάζονται περιέχονται σε λιγότερα από 32kB. Παράδειγμα παιχνιδιού που δεν χρησιμοποιεί ελεγκτή είναι το Tetris, το οποίο έχει μέγεθος ROM 32kB [17].

Με το byte 0x01 στη θέση μνήμης 0x147 προσδιορίζεται ο τύπος μικροκυκλώματος MBC1. Ο MBC1 ήταν το πρώτος ελεγκτής που χρησιμοποιήθηκε στο Game Boy και μπορεί να υποστηρίξει έως

2MB ROM και 32kB RAM, αλλά στην προεπιλεγμένη λειτουργία του υποστηρίζει έως 512kB ROM και 32kB RAM. Οι νεότεροι τύποι ελεγκτών λειτουργούν παρόμοια με τον MBC1, γεγονός που επιτρέπει στους προγραμματιστές την εύκολη εναλλαγή μεταξύ τύπων ελεγκτών στις απαραίτητες περιπτώσεις. Ο MBC1 χρησιμοποιείται από μεγάλο πλήθος παιχνιδιών, ιδιαίτερα σε παιχνίδια των πρώτων ετών κυκλοφορίας του Game Boy, συμπεριλαμβανομένων των *The Legend of Zelda - Link's Awakening* και *Kirby's Dream Land* [18].

Με το byte 0x02 προσδιορίζεται ο συνδυασμός MBC1 και εξωτερικής RAM, δηλαδή RAM που δεν περιέχεται μέσα στον ελεγκτή αλλά βρίσκεται εκτός αυτού. Ο MBC1 ως προεπιλογή δε χρησιμοποιεί την εξωτερική RAM, αλλά έχει τη δυνατότητα να την χρησιμοποιήσει γράφοντας το byte 0x0A (ή κάποιο που να έχει ως λιγότερο σημαντικό δεκαεξαδικό ψηφίο το 0xA) σε μία θέση στο εύρος μνήμης 0x0000-0x1FFF [19].

Παρομοίως, με το byte 0x03 προσδιορίζεται ο συνδυασμός MBC1, εξωτερικής RAM και μπαταρίας. Ο συγκεκριμένος συνδυασμός χρησιμοποιείται για περιπτώσεις προγραμμάτων τα οποία απαιτούν την αποθήκευση των περιεχομένων της RAM όταν η κασέτα αφαιρείται από το Game Boy, όπως στην περίπτωση αποθήκευσης προόδου ενός παιχνιδιού. Με τη χρήση μπαταρίας τα αποθηκευμένα δεδομένα στη RAM της κασέτας διατηρούνται, παρά την διακοπή παροχής ενέργειας λόγω αφαίρεσης της κασέτας.

Εκτός από τον MBC1 υπάρχουν και άλλοι ελεγκτές τράπεζας μνήμης, όπως οι MBC2, MBC3, MBC5, MBC6 και MBC7. Οι κυριότερες διαφορές τους είναι τα μεγέθη ROM και RAM τα οποία υποστηρίζουν, καθώς και η ύπαρξη ρολογιού πραγματικού χρόνου (Real Time Clock - RTC) το οποίο τροφοδοτείται από μπαταρία στο εσωτερικό της κασέτας. Για λόγους πληρότητας, στον πίνακα 1 παρουσιάζονται όλοι οι ελεγκτές τράπεζας μνήμης και οι συνδυασμοί στους οποίους χρησιμοποιούνται:

Συνδυασμός	Κωδικός
ROM (Έλλειψη ελεγκτή τράπεζας μνήμης)	0x00
MBC1	0x01
MBC1-RAM	0x02
MBC1-RAM-Μπαταρία	0x03
MBC2	0x05
MBC2-Μπαταρία	0x06
ROM-RAM	0x08
ROM-RAM-Μπαταρία	0x09
MMM01	0x0B
MMM01-RAM	0x0C

MMM01-RAM-Μπαταρία	0x0D
MBC3-Χρονόμετρο-Μπαταρία	0x0F
MBC3-Χρονόμετρο-RAM-Μπαταρία	0x10
MBC3	0x11
MBC3-RAM	0x12
MBC3-RAM-Μπαταρία	0x13
MBC5	0x19
MBC5-RAM	0x1A
MBC5-RAM-Μπαταρία	0x1B
MBC5-Δόνηση	0x1C
MBC5-Δόνηση-RAM	0x1D
MBC5-Δόνηση-RAM-Μπαταρία	0x1E
MBC6	0x20
MBC7-Αισθητήρας-Δόνηση-RAM-Μπαταρία	0x22
Pocket Camera	0xFC
BANDAI TAMA5	0xFD
HuC3	0xFE
HuC1-RAM- Μπαταρία	0xFF

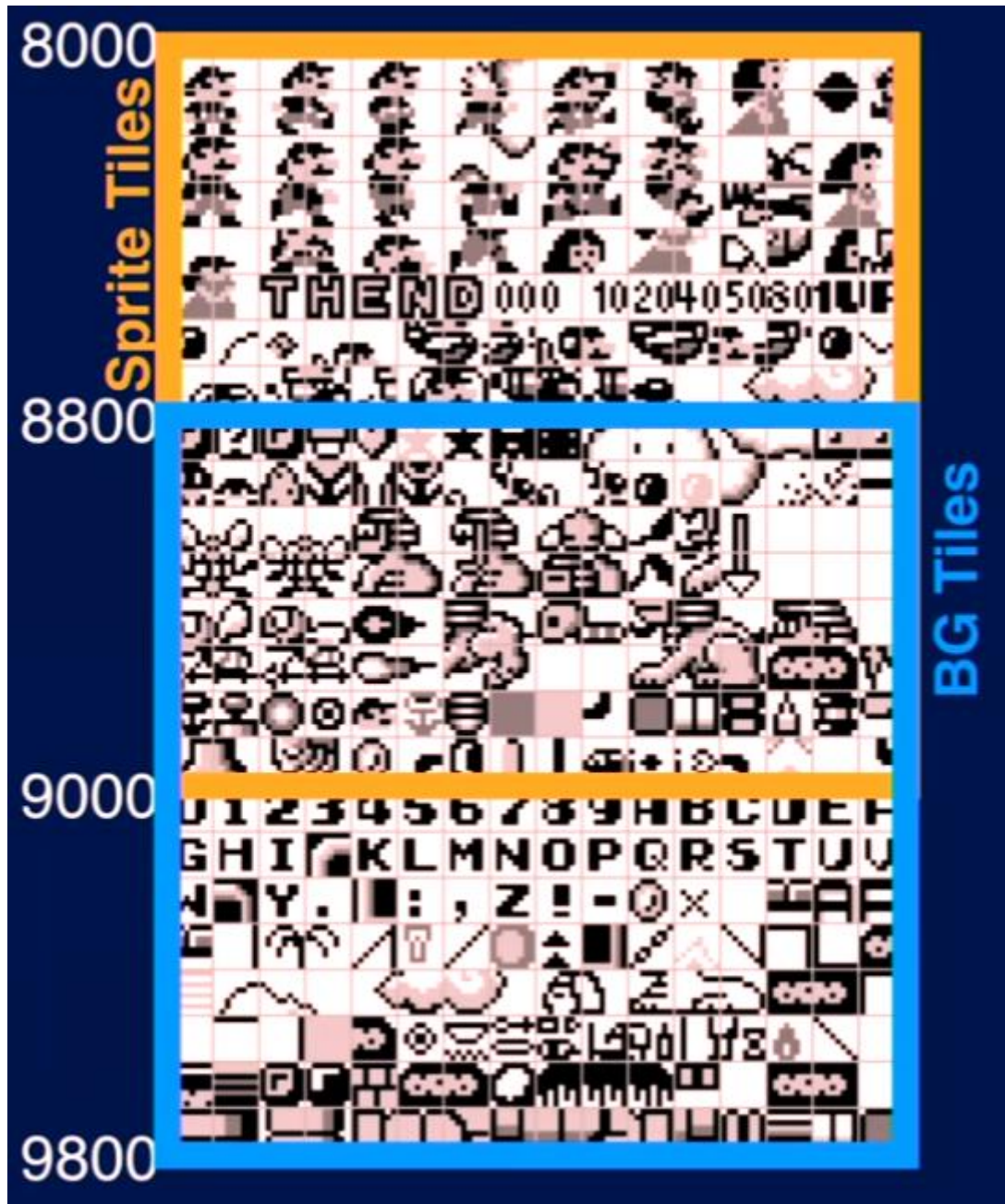
Πίνακας 1 - Συνδυασμοί Memory Bank Controllers και οι κωδικοί τους [20]

4.1.3 Ανάλυση περιεχομένων Video RAM

Για την κατανόηση της διαδικασίας εμφάνισης των γραφικών στην οθόνη του Game Boy, απαραίτητη προϋπόθεση είναι η γνώση των ιδιοτήτων των στοιχείων που είναι αποθηκευμένα στη Video RAM. Η γνώση της περιοχής μνήμης της Video RAM, αλλά και των περιεχομένων της, αποδεικνύεται ιδιαίτερα σημαντικά κατά την ανάπτυξη του υποσυστήματος της μονάδας επεξεργασίας εικόνας, αλλά και τυχόν εργαλείων τα οποία θα αναπτυχθούν για σκοπούς αποσφαλμάτωσης του υποσυστήματος.

Ως Video RAM ορίζεται το εύρος θέσεων μνήμης το οποίο χρησιμοποιείται από τα προγράμματα του Game Boy για τον σκοπό της αποθήκευσης των γραφικών που αποτυπώνονται στην οθόνη. Η Video RAM βρίσκεται στις θέσεις μνήμης 0x8000 έως 0x9FFF και αποτελείται από 8kB δεδομένων. Τα γραφικά αποθηκεύονται στη μνήμη σε πλακίδια (tiles) τα οποία έχουν μέγεθος 8x8 pixels το καθένα, για λόγους βελτιστοποίησης χρήσης της διαθέσιμης μνήμης.

Στο εύρος διευθύνσεων 0x8000 έως 0x97FF αποθηκεύονται τα πλακίδια που απαρτίζουν τα sprites (γραφικά δύο διαστάσεων που ενσωματώνονται στην εικόνα, όπως για παράδειγμα ο παίκτης ή ένας εχθρός) και το παρασκήνιο (background) της εικόνας.

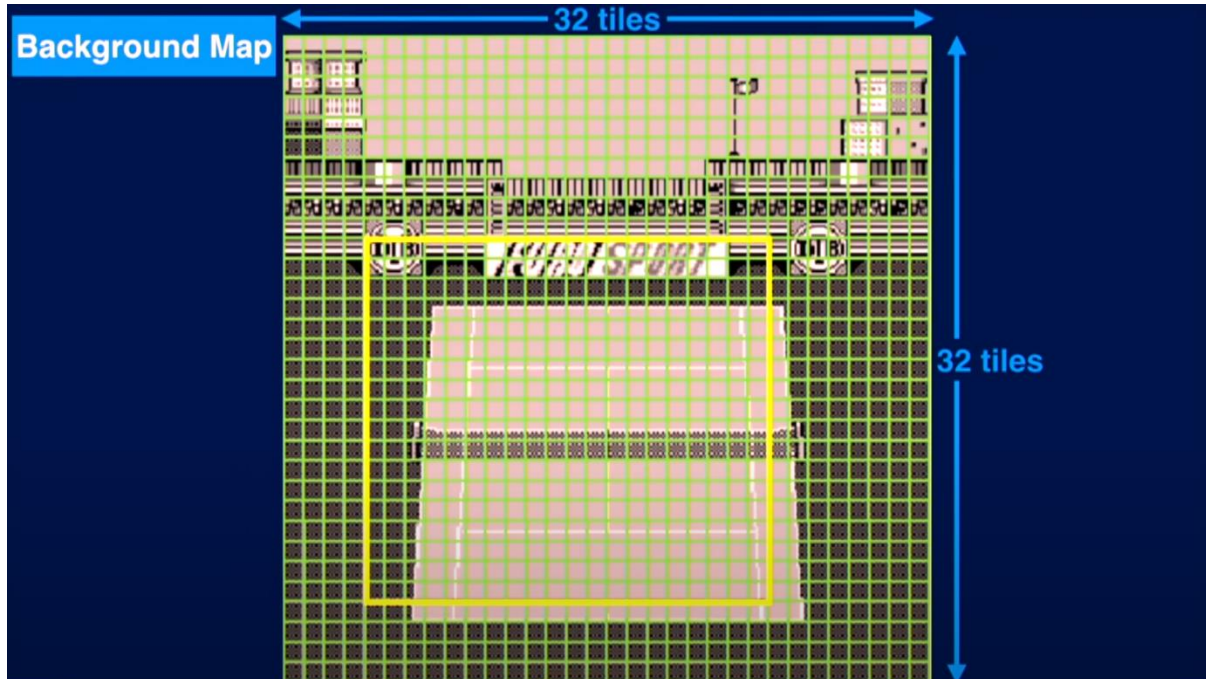


Σχήμα 2 – Παράδειγμα περιεχομένων VRAM [21]

Στο εύρος διευθύνσεων 0x9800 έως 0x9FFF αποθηκεύονται δύο χάρτες παρασκηνίων (background maps) μεγέθους 32x32 πλακιδίων το καθένα. Οι χάρτες παρασκηνίων χρησιμοποιούνται ως ο καμβάς πάνω στον οποίο αποτυπώνονται τα πλακίδια τα οποία βρίσκονται στο εύρος διευθύνσεων 0x8000-0x97FF και δημιουργούν την τελική εικόνα που εμφανίζεται στην οθόνη του Game Boy.

Καθώς ο κάθε χάρτης παρασκηνίου έχει μέγεθος 32x32 πλακίδια, το ανάλογο μέγεθος του σε εικονοστοιχεία είναι 256x256 εικονοστοιχεία, ενώ η οθόνη του Game Boy έχει μέγεθος 160x144 εικονοστοιχεία. Για την εμφάνιση των χαρτών παρασκηνίων στη μικρότερη οθόνη χρησιμοποιούνται δύο καταχωρητές SCY (κύλιση στον άξονα Y) και SCX (κύλιση στον άξονα X), οι οποίοι

πραγματοποιούν κάθετη και οριζόντια κύλιση στον χάρτη παρασκηνίου, έτσι ώστε να εμφανίζεται κάθε φορά το επιθυμητό μέρος του στην οθόνη. Όταν ο χάρτης παρασκηνίου ξεπεράσει τα τελευταία εικονοστοιχεία κατά τη διαδικασία της κύλισης, επανέρχεται ξανά στα πρώτα εικονοστοιχεία, πραγματοποιείται δηλαδή τεχνική wraparound [22].



Σχήμα 3 - Παράδειγμα περιεχομένων χάρτη παρασκηνίου. Η οθόνη του Game Boy συμβολίζεται από το κίτρινο τετράγωνο το οποίο κινείται με βάση τις τιμές των καταχωρητών SCY και SCX. [21]

Σε λειτουργία CGB (η οποία παρουσιάζεται στην ενότητα 4.2.3), η Video RAM έχει τη δυνατότητα να εναλλάσσεται μεταξύ δύο τραπεζών ROM μεγέθους 8kB το καθένα, διπλασιάζοντας με αυτόν τον τρόπο τη διαθέσιμη μνήμη γραφικών.

4.1.4 Περίπτωση χρήσης εξωτερικής RAM (External RAM)

Τα περισσότερα παιχνίδια του Game Boy χρειάζονται έναν τρόπο για να διατηρούν αποθηκευμένα δεδομένα ακόμα και όταν κόβεται η παροχή ρεύματος, έτσι ώστε ο παίκτης να μπορέσει να συνεχίσει την πρόοδό του στο παιχνίδι. Το πρόβλημα αυτό λύνει η εξωτερική RAM, στην οποία υπάρχει συνεχής παροχή ρεύματος με χρήση μπαταρίας στο εσωτερικό της κασέτας. Συνεπώς, γνωρίζοντας το εύρος μνήμης της εξωτερικής RAM, μπορούμε να δημιουργήσουμε τα αρχεία αποθήκευσης των παιχνιδιών στην εφαρμογή του εξομοιωτή.

Η εξωτερική RAM καταλαμβάνει το εύρος διευθύνσεων 0xA000 έως 0xBFFF. Ο συγκεκριμένος τύπος μνήμης δεν υπάρχει σε όλες τις κασέτες και συνεπώς πρέπει να ενεργοποιηθεί στις περιπτώσεις που χρησιμοποιείται, με την πραγματοποίηση μίας εγγραφής σε οποιαδήποτε θέση μνήμης στο εύρος διευθύνσεων 0x0000-0x1FFF.

4.1.5 Περίπτωση χρήσης RAM εργασίας (WRAM)

Η RAM εργασίας (Work RAM - WRAM) είναι γενικής χρήσης RAM μεγέθους 8kB, η οποία βρίσκεται στο εσωτερικό του Game Boy και αντιστοιχεί στο εύρος θέσεων μνήμης 0xC000 έως 0xDFFF. Σε αυτή αποθηκεύονται συνήθως μεταβλητές οι οποίες χρησιμοποιούνται συχνά από το πρόγραμμα. Το συγκεκριμένο εύρος μνήμης δεν αποτελεί κάποιο σημαντικό κομμάτι στο οποίο πρέπει να δοθεί σημασία κατά την ανάπτυξη του εξομοιωτή, καθώς η χρήση του αφορά τους προγραμματιστές των παιχνιδιών Game Boy, αλλά αναφέρεται για λόγους πληρότητας της τεχνικής ανάλυσης.

Τα πρώτα 4kB τα οποία αντιστοιχούν στο εύρος 0xC000 έως 0xCFFF (WRAM Τράπεζα 0) είναι σταθερά, σε αντίθεση με τα επόμενα 4kB του εύρους 0xD000 έως 0xDFFF (WRAM Τράπεζα 1), τα οποία έχουν τη δυνατότητα να εναλλάσσονται μεταξύ 7 διαφορετικών RAM τραπεζών χωρητικότητας 4kB, μόνο όμως όταν το πρόγραμμα εκτελείται σε λειτουργία CGB.

4.1.6 Λόγος ύπαρξης αντικατοπτρισμού RAM εργασίας (WRAM Echo)

Στο εύρος θέσεων μνήμης 0xE000 έως 0xFDFF, βρίσκεται ένα ακριβές αντίγραφο του εύρους 0xC000-0xDDFF της RAM εργασίας. Κάθε εγγραφή στις διευθύνσεις της RAM εργασίας αντικατοπτρίζεται στο συγκεκριμένο εύρος, στο οποίο συνιστάται από τον κατασκευαστή να μη πραγματοποιούνται εγγραφές από τους προγραμματιστές. Δεν υπάρχει υλικό στο εσωτερικό του Game Boy για αυτές τις περιοχές μνήμης, καθώς η ύπαρξή του συγκεκριμένου εύρους οφείλεται στον τρόπο με τον οποίο είναι συνδεδεμένη η RAM εργασίας, και αλλαγές σε αυτό το εύρος στην πραγματικότητα γίνονται στη RAM εργασίας. [23] Η συγκεκριμένη περιοχή μνήμης αποτελεί μία ιδιαιτερότητα η οποία πρέπει να ληφθεί υπόψιν από τον αναγνώστη έτσι ώστε εκείνος να έχει την πλήρη εικόνα της λειτουργίας της κάθε περιοχής του χάρτη μνήμης.

4.1.7 Ανάλυση μνήμης χαρακτηριστικών γραφικών στοιχείων (Sprite Attribute Memory - OAM)

Το εύρος μνήμης 0xFE00-0xFEAF αποτελείται από 160 bytes, τα οποία χρησιμοποιούνται για την αποθήκευση πληροφοριών 40 γραφικών στοιχείων (4 bytes για κάθε στοιχείο). Το συγκεκριμένο εύρος μνήμης αποτελεί σημαντικό κομμάτι του τομέα εμφάνισης των γραφικών, καθώς περιέχει ρυθμίσεις για την εμφάνιση σημαντικών γραφικών στοιχείων όπως ο παίκτης του παιχνιδιού ή οι εχθροί. Συνεπώς, είναι απαραίτητη η γνώση των συγκεκριμένων ρυθμίσεων για την υλοποίηση του κύκλου καταστάσεων της μονάδας επεξεργασίας εικόνας του εξομοιωτή, και πιο συγκεκριμένα, για την κατάσταση λειτουργίας 2 (Αναζήτηση OAM).

Το πρώτο από τα 4 bytes περιέχει τη θέση στον άξονα Y στην οποία πρέπει να εμφανιστεί η πρώτη σειρά εικονοστοιχείων ενός γραφικού στοιχείου 8x16 εικονοστοιχείων συν 16. Για τιμή 0 το γραφικό στοιχείο εξαφανίζεται, ενώ για τιμή 2 εμφανίζονται οι δύο τελευταίες σειρές του γραφικού στοιχείου. Εάν το γραφικό στοιχείο έχει μέγεθος 8x8, τότε για τιμή 2 το γραφικό στοιχείο θα εξαφανιστεί. Για τιμή 144, το γραφικό στοιχείο 8x16 ευθυγραμμίζεται με το κάτω μέρος της οθόνης,

ενώ στην περίπτωση ενός γραφικού στοιχείου 8x8, αυτό θα ευθυγραμμιστεί με το κάτω μέρος της οθόνης για τιμή 152.

Το δεύτερο byte περιέχει τη θέση του γραφικού στοιχείου στον άξονα X, και λειτουργεί με παρόμοιο τρόπο, όμως προστίθεται η τιμή 8 αντί για 16, καθώς όλα τα γραφικά στοιχεία έχουν πλάτος 8. Για παράδειγμα, για τιμή 0 το γραφικό στοιχείο κρύβεται αριστερά της οθόνης, ενώ για τιμή 160 ευθυγραμμίζεται με το δεξί μέρος της οθόνης, καθώς το πλάτος της οθόνης του Game Boy αποτελείται από 160 εικονοστοιχεία.

Το τρίτο byte περιέχει τον δείκτη ως προς τη Video RAM του πλακιδίου το οποίο χρειαζόμαστε. Η τιμή του byte επιλέγει ένα πλακίδιο (8x8 pixels) από το εύρος θέσεων μνήμης 0x8000-0x8FFF (Video RAM).

Το τέταρτο byte περιέχει πληροφορίες σχετικά με τον τρόπο τοποθέτησης του γραφικού στοιχείου, οι οποίες περιγράφονται στον πίνακα 2:

Bit	Λειτουργία
7	Επικάλυψη γραφικού από background (0=Όχι, 1=Επικάλυψη)
6	Αναστροφή στον άξονα Y (0=Κανονικό, 1=Ανεστραμμένο)
5	Αναστροφή στον άξονα X (0=Κανονικό, 1=Ανεστραμμένο)
4	Αριθμός παλέτας (0= Object Palette 0, 1=Object Palette 1)
3	Επιλογή τράπεζας Video Ram (0=Τράπεζα 0, 1=Τράπεζα 1)
2-0	Αριθμός παλέτας 0-7 (Λειτουργία CGB)

Πίνακας 2 - Αποκωδικοποίηση τέταρτου byte κάθε γραφικού στοιχείου στη μνήμη χαρακτηριστικών γραφικών στοιχείων

4.1.7.1 Ανάλυση λειτουργίας της μεταφοράς OAM DMA (OAM DMA Transfer)

Στις περιπτώσεις παιχνιδιών τα οποία απαιτούν γρήγορες εναλλαγές των γραφικών στοιχείων στην οθόνη, είναι κρίσιμη η γρήγορη μεταφορά δεδομένων από την RAM/ROM στην μνήμη χαρακτηριστικών γραφικών στοιχείων (OAM). Με εγγραφή στον καταχωρητή DMA (Direct Memory Access – άμεση προσπέλαση μνήμης) ο οποίος βρίσκεται στη διεύθυνση μνήμης 0xFF46, πραγματοποιείται εκκίνηση της λειτουργίας μεταφοράς OAM DMA. Η αναφορά στην συγκεκριμένη λειτουργία κρίνεται απαραίτητη, καθώς για τη σωστή λειτουργία του εξομοιωτή, θα πρέπει αυτή να υλοποιείται όταν πραγματοποιηθεί εγγραφή στην συγκεκριμένη διεύθυνση μνήμης.

Κατά τη διαδικασία μεταφοράς OAM DMA, πραγματοποιείται αντιγραφή των γραφικών στοιχείων, από τις θέσεις μνήμης ROM/RAM, στις θέσεις μνήμης της μνήμης χαρακτηριστικών γραφικών στοιχείων (OAM). Η συγκεκριμένη αντιγραφή ολοκληρώνεται σε 160 κύκλους μηχανής, οι οποίοι πραγματοποιούνται σε 152 microseconds. Κατά τη διάρκεια της αντιγραφής, ο επεξεργαστής μπορεί να προσπελάσει μόνο την περιοχή μνήμης υψηλής RAM (0xFF80-0xFFFF). Για αυτό το λόγο, για την εκκίνηση της λειτουργίας μεταφοράς OAM DMA πραγματοποιείται αντιγραφή μίας

υπορουτίνας στη συγκεκριμένη περιοχή μνήμης από τους προγραμματιστές των παιχνιδιών, η οποία ξεκινάει τη διαδικασία μεταφοράς OAM DMA, και εκτελείται για 160 κύκλους μηχανής ώσπου αυτή να ολοκληρωθεί [24].

4.1.8 Παρουσίαση καταχωρητών εισόδου/εξόδου (Input/Output)

Το εύρος διευθύνσεων από 0xFF00 έως 0xFF7F περιέχει τους καταχωρητές οι οποίοι είναι υπεύθυνοι για τον έλεγχο του υλικού του Game Boy, όπως για παράδειγμα της οθόνης, του ήχου ή της εισόδου χρήστη. Οι συγκεκριμένοι καταχωρητές είναι κρίσιμοι για την υλοποίηση σημαντικών κομματιών του εξομοιωτή, όπως του υποσυστήματος της μονάδας επεξεργασίας εικόνας, αλλά και της εισόδου χρήστη, και για αυτό το λόγο κρίνεται σημαντική μία συνοπτική αναφορά τους στο παρόν υποκεφάλαιο, προτού αναλυθούν περαιτέρω στα κεφάλαια της ανάπτυξης των υποσυστημάτων τους.

Οι λειτουργίες που πραγματοποιούνται από τους συγκεκριμένους καταχωρητές συμπεριλαμβάνουν την κύλιση του χάρτη παρασκήνιου στην οθόνη (καταχωρητές κύλισης Y και X στις θέσεις μνήμης 0xFF42, 0xFF43) [25], τον έλεγχο για πάτημα κουμπιών από τον χρήστη (καταχωρητής P1/JOYP στη θέση μνήμης 0xFF00) [26], τον έλεγχο της έντασης ήχου σε κάθε ένα από τα δύο ηχεία (καταχωρητής NR50 στη θέση μνήμης 0xFF24) και την επιλογή καναλιού ήχου από τα 4 διαθέσιμα (καταχωρητής NR51 στη θέση μνήμης 0xFF25) [27].

4.1.9 Περίπτωση χρήσης υψηλής RAM (HRAM)

Οι διευθύνσεις μνήμης 0xFF80 έως 0xFFFF περιέχουν την περιοχή της υψηλής RAM (High Ram - HRAM). Η συγκεκριμένη περιοχή μνήμης βρίσκεται στο εσωτερικό του Game Boy και χρησιμοποιείται κατά τη διαδικασία DMA transfer, καθώς σε αυτή την περιοχή δεν κόβεται η πρόσβαση όσο διαρκεί η διαδικασία DMA transfer [28]. Επίσης, ο προεπιλεγμένος δείκτης στοίβας βρίσκεται μέσα σε αυτό το εύρος, οπότε χρησιμοποιείται και από τη στοίβα εάν ο δείκτης δεν αλλαχθεί από τον προγραμματιστή της κασέτας.

4.1.10 Παρουσίαση τρόπου λειτουργίας του καταχωρητή ενεργοποίησης διακοπών (IE)

Ο καταχωρητής ενεργοποίησης διακοπών (Interrupt Enable - IE) είναι ένας σημαντικός καταχωρητής, καθώς είναι υπεύθυνος για την ενεργοποίηση και την απενεργοποίηση των διακοπών του συστήματος. Ο συγκεκριμένος καταχωρητής βρίσκεται στη θέση μνήμης 0xFFFF και περιέχει τους τύπους διακοπών καθώς και την ενεργοποίηση και απενεργοποίησή τους. Για παράδειγμα, στην περίπτωση όπου έχει ξεκινήσει μία διαδικασία διακοπής, και το bit που ελέγχει τη διακοπή για την είσοδο χρήστη (Joyrad Interrupt) έχει τιμή 1, τότε εκτελείται η διαδικασία για την αναγνώριση του κουμπιού που πάτησε ο χρήστης. Η ενεργοποίησή τους ελέγχεται από τις τιμές που περιέχουν τα bits του καταχωρητή ενεργοποίησης διακοπών, σύμφωνα με τον πίνακα 3:

Bit	0	1	2	3	4
Διακοπή (1=Ενεργοποίηση, 0=Απενεργοποίηση)	VBlank	LCD STAT	Timer	Serial	Joypad

Πίνακας 3 - Τύποι διακοπών στον καταχωρητή IE [29]

4.2 Ανάλυση λειτουργίας επεξεργαστή (CPU)

4.2.1 Τεχνικά χαρακτηριστικά επεξεργαστή

Ο επεξεργαστής που χρησιμοποιείται στο Game Boy είναι 8 bit, καθώς αυτή είναι η χωρητικότητα των καταχωρητών του, έχει συχνότητα λειτουργίας 4,194304 MHz και ήταν γνωστό πως χρησιμοποιεί παρόμοιο, αλλά όχι ακριβώς το ίδιο σετ εντολών με τους επεξεργαστές Zilog Z80 και Intel 8080 [30]. Ωστόσο, το 2019 παρατηρήθηκε πως τεχνικά έγγραφα από σειρά ενσωματωμένων συστημάτων (System on Chip - SoC) της Sharp χρονολογίας 1996 περιείχαν τον επεξεργαστή SM83, ο οποίος προοριζόταν για κλιματιστικά και είχε απaráλλακτο σετ εντολών σε σχέση με τον επεξεργαστή του Game Boy [31]. Με αυτή την ανακάλυψη, άρχισε να πιστεύεται πως το Game Boy χρησιμοποιούσε τον συγκεκριμένο επεξεργαστή, καθώς δεν είχε βρεθεί μέχρι τότε άλλος επεξεργαστής με ολόιδιο σετ εντολών.

4.2.2 Παρουσίαση των βασικών καταχωρητών του επεξεργαστή

Ο επεξεργαστής του Game Boy αποτελείται από οκτώ καταχωρητές χωρητικότητας 8 bit (A, B, C, D, E, F, H, L) και δύο 16 bit καταχωρητές (SP, PC). Οι 8 bit καταχωρητές μπορούν να συνδυαστούν σε ζεύγη (AF, BC, DE, HL) για να χρησιμοποιηθούν και ως 16 bit καταχωρητές σε ορισμένες εντολές. Οι συγκεκριμένοι καταχωρητές αποτελούν σημαντικά κομμάτια του υποσυστήματος του επεξεργαστή. Για αυτό τον λόγο, στο παρόν κεφάλαιο παρουσιάζεται αναλυτικά η λειτουργία τους, ως οδηγός για την σωστή ενσωμάτωση των λειτουργιών τους στον εξομοιωτή.

4.2.2.1 Παρουσίαση καταχωρητή σημαίας F (Flag Register)

Ο καταχωρητής F ονομάζεται καταχωρητής σημαίας (Flag Register), επειδή σε αυτόν αποθηκεύονται οι σημαίες (flags) οι οποίες χρησιμοποιούνται στις αριθμητικές εντολές. Ειδικότερα, τα 4 περισσότερο σημαντικά bit (most significant bit) του καταχωρητή F αποτελούνται από τα Zero Flag (Z), Subtraction Flag (N), Half Carry Flag (H) και Carry Flag (C) με τη συγκεκριμένη σειρά. Ο καταχωρητής F είναι ένας ιδιαίτερα σημαντικός καταχωρητής, καθώς χρησιμοποιείται από μεγάλο μέρος εντολών του επεξεργαστή, και συνεπώς ήταν απαραίτητη η σωστή ενσωμάτωσή του για την ακρίβεια του εξομοιωτή.

Θέση Bit	7	6	5	4	3	2	1	0
Περιεχόμενα	Z	N	H	C	0	0	0	0

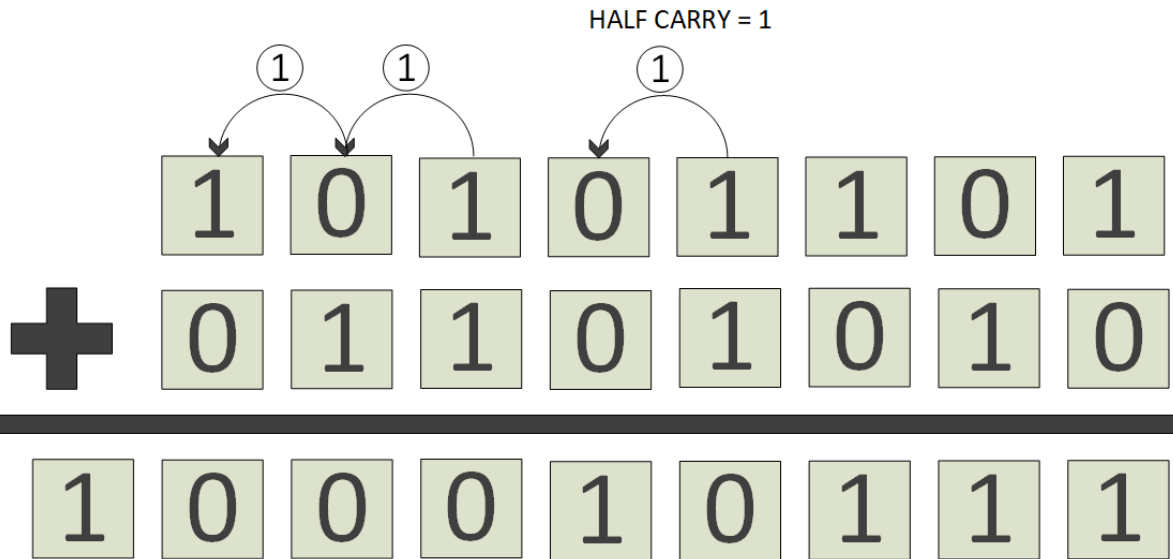
Πίνακας 4 - Περιεχόμενα καταχωρητή F

Το Zero Flag λαμβάνει τιμή 1 όταν το αποτέλεσμα μίας αριθμητικής εντολής είναι ίσο με 0, και τιμή 0 όταν αυτό δεν είναι ίσο με 0. Παράδειγμα μίας τέτοιας εντολής είναι η εντολή <SUB B>, η οποία αφαιρεί τα περιεχόμενα του καταχωρητή B από τον καταχωρητή A και αποθηκεύει το αποτέλεσμα στον καταχωρητή A. Όταν οι καταχωρητές A και B έχουν ίσα περιεχόμενα, η αφαίρεσή τους έχει αποτέλεσμα 0, και το Zero Flag λαμβάνει την τιμή 1. Στην περίπτωση όπου το αποτέλεσμα είναι διαφορετικό του 0, το Zero Flag λαμβάνει την τιμή 0.

Το Subtraction Flag λαμβάνει τιμή 1 όταν εκτελεστεί μία αριθμητική εντολή αφαίρεσης, και τιμή 0 στις υπόλοιπες περιπτώσεις αριθμητικών εντολών. Ειδικότερα, οι αριθμητικές εντολές οι οποίες θέτουν την τιμή 1 στο Subtraction Flag είναι οι εντολές SUB (Subtract), SBC (Subtract with Carry), DEC (Decrease) και CP (Compare). Όλες οι υπόλοιπες αριθμητικές εντολές θέτουν την τιμή 0 στο Subtraction Flag αφού εκτελεστούν.

Το Carry Flag λαμβάνει τιμή 1 όταν το αποτέλεσμα μίας αριθμητικής εντολής παράγει κρατούμενο, και 0 όταν δεν παράγει κρατούμενο. Έστω ότι οι καταχωρητές A και B περιέχουν τις δεκαεξαδικές τιμές 0xC6 (δεκαδική τιμή 198) και 0x91 (δεκαδική τιμή 145) αντίστοιχα. Με την εκτέλεση της εντολής <ADD A, B>, η οποία προσθέτει τα περιεχόμενα των καταχωρητών A και B και αποθηκεύει το αποτέλεσμα στον καταχωρητή A, το αποτέλεσμα είναι 0x157 (δεκαδική τιμή 343). Όμως ο καταχωρητής A, καθώς έχει χωρητικότητα ενός byte, μπορεί να λάβει τιμές στο εύρος 0x00 έως 0xFF (δεκαδικές τιμές 0 έως 255). Σε αυτή την περίπτωση υπάρχει υπερχείλιση (overflow), δηλαδή για τιμές υψηλότερες από 0xFF, το άθροισμα συνεχίζει να υπολογίζεται από την τιμή 0x00 και το Carry Flag λαμβάνει την τιμή 1. Συνεπώς, από την πρόσθεση 0xC6+0x91, το αποτέλεσμα που θα αποθηκευτεί στον καταχωρητή A θα είναι ο δεκαεξαδικός αριθμός 0x58 (δεκαδική τιμή 88), και το Carry Flag λαμβάνει την τιμή 1.

Το Half Carry Flag λαμβάνει τιμή 1 όταν παράγεται κρατούμενο κατά την εκτέλεση πράξης μεταξύ των τέταρτων λιγότερο σημαντικών bit δύο αριθμών. Για παράδειγμα, έστω πως πρέπει να εκτελεστεί η εντολή <ADD A, B>, και τα περιεχόμενα των καταχωρητών A και B είναι οι δυαδικοί αριθμοί 10101101 και 01101010 αντίστοιχα (δεκαεξαδικές τιμές 0xAD και 0x6A). Κατά τη διάρκεια της πρόσθεσης των τέταρτων λιγότερων σημαντικών bit (1+1) δημιουργείται κρατούμενο, το οποίο έπειτα προστίθεται στην πράξη μεταξύ των πέμπτων λιγότερο σημαντικότερων bit (0+0).



Σχήμα 4 - Αναπαράσταση πρόσθεσης που ενεργοποιεί τη σημαία Half Carry

Στη συγκεκριμένη περίπτωση, το Half Carry Flag λαμβάνει την τιμή 1. Αντιθέτως, εάν στη συγκεκριμένη θέση δεν υπάρξει κρατούμενο, το Half Carry Flag λαμβάνει την τιμή 0. Συνοπτικά, το Half Carry Flag ενεργοποιείται όταν υπάρξει κρατούμενο κατά την εκτέλεση πράξης μεταξύ των δύο λιγότερο σημαντικών δεκαεξαδικών ψηφίων δύο bytes (Παράδειγμα: $0xAD+0x6A \Rightarrow 0xD0+0xA \Rightarrow$ Παράγεται κρατούμενο \Rightarrow Half Carry Flag λαμβάνει τιμή 1).

4.2.2.2 Περιγραφή μετρητή προγράμματος (Program Counter - PC)

Ο μετρητής προγράμματος είναι ίσως ο πιο σημαντικός καταχωρητής, και από τους πρώτους που δημιουργήθηκαν κατά την συγγραφή του κώδικα της εφαρμογής, καθώς είναι απαραίτητος για την εκτέλεση όλων των εντολών και το διάβασμα της μνήμης του Game Boy. Στο παρόν υποκεφάλαιο θα πραγματοποιηθεί η περιγραφή του τρόπου λειτουργίας του μετρητή προγράμματος, έτσι ώστε αυτός να γίνει κατανοητός από τον αναγνώστη, αλλά και να επεξηγηθεί η σημασία του καταχωρητή για τη λειτουργία του επεξεργαστή.

Ο μετρητής προγράμματος (Program Counter - PC) είναι ένας 16 bit καταχωρητής στον οποίο περιέχεται η διεύθυνση μνήμης της επόμενης εντολής που θα εκτελεστεί στο πρόγραμμα. Καθώς το εύρος των θέσεων μνήμης του Game Boy κυμαίνεται από τις δεκαεξαδικές τιμές $0x0000$ έως $0xFFFF$, τα 16 bit είναι αρκετά για να τις αναπαραστήσουν όλες. Κατά την εκκίνηση του Game Boy ο μετρητής προγράμματος αρχικοποιείται με την τιμή $0x0100$, συνεπώς η πρώτη εντολή που εκτελείται σε κάθε πρόγραμμα βρίσκεται στη θέση μνήμης $0x0100$. Στη συνέχεια η τιμή του μετρητή προγράμματος μεταβάλλεται μέσω των εντολών του προγράμματος.

Κάποιες εντολές, όπως η εντολή $\langle ADD A, d8 \rangle$, η οποία προσθέτει το επόμενο byte της μνήμης στα περιεχόμενα του καταχωρητή A και αποθηκεύει το αποτέλεσμα σε αυτόν, χρησιμοποιούν την

επόμενη θέση μνήμης για τα ορίσματα που απαιτούνται για την εκτέλεσή τους. Σε αυτές τις περιπτώσεις ο μετρητής προγράμματος λαμβάνει την τιμή της επόμενης θέσης μνήμης η οποία περιέχει εκτελέσιμη εντολή (για το παράδειγμά μας ο μετρητής θα λάβει την τιμή της μεθεπόμενης θέσης μνήμης από αυτήν της εντολής ADD).

Ένα ακόμα αξιοσημείωτο πλήθος εντολών οι οποίες μεταβάλλουν τον μετρητή προγράμματος, είναι οι εντολές άλματος JUMP. Ένα παράδειγμα είναι η εντολή <JP a16>, η οποία μεταβάλλει τον μετρητή αλλάζοντας την τιμή του με τα bytes τα οποία βρίσκονται στις δύο επόμενες θέσεις μνήμης από αυτήν της εντολής. Επομένως, η επόμενη εντολή που θα εκτελεστεί βρίσκεται στη θέση μνήμης η οποία ορίζεται από τα επόμενα δύο bytes.

4.2.2.3 Περιγραφή δείκτη στοίβας (Stack Pointer)

Για την σωστή λειτουργία του εξομοιωτή, ήταν απαραίτητη η ενσωμάτωση της στοίβας που χρησιμοποιείται από το Game Boy για την αποθήκευση μεταβλητών. Αυτή πραγματοποιείται με τον ορισμό μίας μεταβλητής στον κώδικα του εξομοιωτή, η οποία προσομοιώνει τον δείκτη στοίβας (Stack Pointer – SP) που περιέχει την διεύθυνση μνήμης της κορυφής της. Για τους σκοπούς της σωστής λειτουργίας του εξομοιωτή, είναι απαραίτητο να αναφερθούν όλοι οι τρόποι με τους οποίους μπορεί να μεταβληθεί ο δείκτης στοίβας.

Ο δείκτης στοίβας είναι ένας 16 bit καταχωρητής, ο οποίος περιέχει τη διεύθυνση μνήμης η οποία δείχνει στην κορυφή της στοίβας που χρησιμοποιείται από το πρόγραμμα για την αποθήκευση απαραίτητων μεταβλητών. Τέτοιες μεταβλητές μπορεί να είναι αποθηκευμένες μεταβλητές για μετέπειτα χρήση, διευθύνσεις μνήμης ή ορίσματα υπορουτινών.

Η μεταβολή του δείκτη στοίβας γίνεται συνήθως με τη χρήση των εντολών CALL, PUSH, RST οι οποίες προσθέτουν δεδομένα στη στοίβα, και με τη χρήση εντολών POP, RET και RETI, οι οποίες αφαιρούν δεδομένα από τη στοίβα. Η στοίβα μεγαλώνει από τις μεγαλύτερες θέσεις μνήμης προς τις μικρότερες και ο δείκτης στοίβας αυξομειώνεται αυτόματα προτού προστεθεί ή αφαιρεθεί ένα καινούργιο δεδομένο από την μνήμη. Είναι αποδεκτός λοιπόν, ο ορισμός μίας αρχικής διεύθυνσης στον δείκτη στοίβας, η οποία είναι μία θέση εκτός του εύρους της διαθέσιμης μνήμης της στοίβας, αφού εκείνη θα μειωθεί πριν την είσοδο του πρώτου δεδομένου, και συνεπώς το δεδομένο θα εισέλθει μέσα στα όρια που έχουν κατανεμηθεί για τη στοίβα.

Κατά την εκκίνηση του Game Boy ο δείκτης στοίβας περιέχει την τιμή 0xFFFFE, άρα τα περιεχόμενα της στοίβας θα ξεκινήσουν από την αμέσως προηγούμενη θέση μνήμης. Ωστόσο, είναι συνηθισμένη πρακτική να αλλάζει η διεύθυνση της στοίβας χειροκίνητα από τον προγραμματιστή ανάλογα με τις ανάγκες του προγράμματος. Για παράδειγμα, μέσω της εκτέλεσης της εντολής <LD SP, 0xD000>, η κορυφή της στοίβας μετακινείται στη θέση μνήμης 0xD000. Στο συγκεκριμένο παράδειγμα, το πρώτο byte που θα αποθηκευτεί στη στοίβα, θα τοποθετηθεί στη θέση μνήμης 0xCFFF.

4.2.3 Αναλυτική παρουσίαση του σετ εντολών του Game Boy (Instruction Set)

Το σετ εντολών του Game Boy αποτελείται από πλήθος εντολών οι οποίες κατηγοριοποιούνται σε μεταφορά δεδομένων, αριθμητικές/λογικές, εντολές κύλισης/αντικατάστασης bit, άλματα και διάφορες εντολές ελέγχου του επεξεργαστή. Το παρόν υποκεφάλαιο αποσκοπεί σε μία ολοκληρωμένη αναφορά σε όλες τις ομάδες εντολών του σετ εντολών οι οποίες υλοποιήθηκαν στα πλαίσια ανάπτυξης του υποσυστήματος του επεξεργαστή. Επίσης, αναφέρονται και τυχόν ιδιαιτερότητες οι οποίες χρειάζονται προσοχή, καθώς επηρεάζουν τη σωστή λειτουργία του εξομοιωτή.

Για την σωστή λειτουργία του εξομοιωτή και την εξασφάλιση της συμβατότητας με το μέγιστο δυνατό αριθμό παιχνιδιών, είναι απαραίτητο να ενσωματωθεί το πλήρες σετ εντολών στον εξομοιωτή. Η έλλειψη έστω και μίας εντολής είναι ικανή να θέσει τον εξομοιωτή μη λειτουργικό. Οι εντολές αυτές καταλαμβάνουν από 8 έως και 24 bit η κάθε μία, ανάλογα με τον τύπο της εντολής.

LD r, r'	
Load to the 8-bit register r, data from the 8-bit register r'.	
Opcode	0b01xxxyyy / various
Length	1 byte
Flags	-
Pseudocode	opcode = read(PC++) # example: LD B, C if opcode == 0x41: B = C

Σχήμα 5 - Παράδειγμα εντολής του Game Boy [32]

Η κάθε εντολή χωρίζεται σε δύο μέρη. Το πρώτο μέρος είναι ο κωδικός της εντολής (opcode), ο οποίος είναι ένας δεκαεξαδικός αριθμός που αναπαριστά την συγκεκριμένη εντολή η οποία θα εκτελεστεί και καταλαμβάνει 8 bit (1 byte) μνήμης. Το δεύτερο μέρος της εντολής είναι οι τελεστικοί (operands), οι οποίοι είναι οι απαραίτητες τιμές που χρειάζεται μία εντολή για να εκτελεστεί επιτυχώς, και καταλαμβάνουν είτε 8 είτε 16 bit ανάλογα με τον τύπο της εντολής. Παραδείγματα τελεστικών μπορεί να είναι ένας αριθμός που θα προστεθεί σε κάποιον καταχωρητή, ή μία διεύθυνση μνήμης στην οποία θα μεταβεί ο μετρητής προγράμματος μετά την εκτέλεση εντολής άλματος.

Για τον προγραμματισμό στο Game Boy γίνεται η χρήση μνημονικών βοηθητικών φράσεων (mnemonics). Οι φράσεις αυτές χρησιμοποιούνται για να κάνουν τον κώδικα πιο ευανάγνωστο και κατανοητό, καθώς αντικαθιστούν τους πιο δύσκολους στην απομνημόνευση δεκαεξαδικούς κωδικούς εντολών που χρησιμοποιούνται στη γλώσσα μηχανής του Game Boy. Ορισμένα κοινά μνημονικά που χρησιμοποιούνται κατά τον προγραμματισμό περιλαμβάνουν τα LD (φόρτωση τιμής σε καταχωρητή), ADD (πρόσθεση), SUB (αφαίρεση), JP (άλμα) και CALL (κλήση υπορουτίνας).

4.2.3.1 Αναφορά στις εντολές μεταφοράς δεδομένων

Οι εντολές μεταφοράς δεδομένων επιτρέπουν στον επεξεργαστή του Game Boy να μετακινεί δεδομένα μεταξύ καταχωρητών και θέσεων μνήμης. Πιο συγκεκριμένα, οι εντολές αυτές είναι οι LD (Load), PUSH και POP. Οι οδηγίες LD επιτρέπουν στον επεξεργαστή να φορτώσει δεδομένα από τη μνήμη σε έναν από τους καταχωρητές του. Οι οδηγίες PUSH και POP χρησιμοποιούνται για τη μετακίνηση δεδομένων εντός και εκτός της στοίβας, αντίστοιχα.

4.2.3.2 Αναφορά στις αριθμητικές εντολές

Οι αριθμητικές εντολές χρησιμοποιούνται από τον επεξεργαστή για την εκτέλεση πράξεων με δεδομένα που είναι αποθηκευμένα στη μνήμη ή στους καταχωρητές του επεξεργαστή. Αυτές οι εντολές περιλαμβάνουν τις ADD, SUB, ADC (πρόσθεση με κρατούμενο), SBC (αφαίρεση με κρατούμενο), INC (αύξηση), DEC (μείωση) και CP (σύγκριση).

Οι εντολές ADD και SUB χρησιμοποιούνται για την πρόσθεση και την αφαίρεση τιμών. Αντίστοιχα, οι εντολές ADC και SBC χρησιμοποιούνται για πρόσθεση και αφαίρεση, συμπεριλαμβανόμενου και του κρατούμενου που ενδέχεται να υπάρχει στον καταχωρητή F (Flag Register). Οι εντολές INC και DEC χρησιμοποιούνται για την αύξηση ή τη μείωση μιας τιμής, αντίστοιχα. Το CP χρησιμοποιείται για να συγκρίνει δύο τιμές και να καθορίσει ποια είναι μεγαλύτερη ή μικρότερη. Εάν οι τιμές είναι ισάξιες, τότε το Zero Flag λαμβάνει την τιμή 1.

4.2.3.3 Αναφορά στις λογικές εντολές

Οι λογικές εντολές χρησιμοποιούνται για την εκτέλεση λογικών πράξεων σε δεδομένα που είναι αποθηκευμένα στη μνήμη ή στους καταχωρητές του επεξεργαστή. Αυτές οι εντολές περιλαμβάνουν τις AND, OR, XOR, NOT, RLA (Rotate Left Accumulator), RLCA (Rotate Left Circular Accumulator), RRA (Rotate Right Accumulator) και RRCA (Rotate Right Circular Accumulator).

Οι εντολές AND, OR και XOR χρησιμοποιούνται για την εκτέλεση των λογικών πράξεων AND, OR, XOR σε δύο τιμές. Το NOT χρησιμοποιείται για την αναστροφή μιας τιμής. Οι εντολές RLA και RRA χρησιμοποιούνται για την μετατόπιση αριστερά και δεξιά, αντίστοιχα, των bit που περιέχονται στον καταχωρητή A κατά ένα bit. Η μετατόπιση αυτή περνάει μέσα από το Carry Flag, δηλαδή στην περίπτωση της αριστερής μετατόπισης, το Carry Flag θα λάβει την τιμή του πρώτου περισσότερο σημαντικού ψηφίου των bit που περιέχονται στον καταχωρητή A, ενώ το όγδοο περισσότερο σημαντικό ψηφίο του καταχωρητή A θα λάβει την τιμή την οποία περιείχε το Carry Flag.

Τέλος, οι εντολές RLCA και RRCA λειτουργούν με παρόμοιο τρόπο με τις εντολές RLA και RRA, με μόνη διαφορά ότι το Carry Flag δεν δίνει την τιμή του στην κύλιση, αλλά απλά λαμβάνει την τιμή του περισσότερο σημαντικού ψηφίου ή του λιγότερο σημαντικού ψηφίου του καταχωρητή A, ανάλογα με το εάν πραγματοποιείται αριστερή ή δεξιά κύλιση. Η ίδια τιμή λαμβάνεται και από το λιγότερο σημαντικό ψηφίο ή το περισσότερο σημαντικό ψηφίο, ανάλογα και πάλι με το εάν πραγματοποιείται αριστερή ή δεξιά κύλιση.

4.2.3.4 Αναφορά στις εντολές κύλισης και αντικατάστασης Bit

Οι εντολές κύλισης και αντικατάστασης bit χρησιμοποιούνται για την κύλιση και την αντικατάσταση bit σε ένα byte δεδομένων. Οι εντολές κύλισης είναι οι RL (Rotate Left – αριστερή κύλιση), RLC (Rotate Left Circular – αριστερή κυκλική κύλιση), RR (Rotate Right – δεξιά κύλιση), RRC (Rotate Right Circular – δεξιά κυκλική κύλιση), SL (Shift Left – αριστερή μετατόπιση), SR (Shift Right – δεξιά μετατόπιση), ενώ οι εντολές αντικατάστασης bit είναι οι SWAP, BIT, RES και SET.

Οι εντολές RL και RR χρησιμοποιούνται για την μετατόπιση αριστερά και δεξιά, αντίστοιχα, των bit που δίνονται στον τελεσταίο κατά ένα bit. Η μετατόπιση αυτή περνάει μέσα από το Carry Flag, δηλαδή στην περίπτωση της αριστερής μετατόπισης, το Carry Flag λαμβάνει την τιμή του πρώτου περισσότερο σημαντικού ψηφίου των bit που περιέχονται στον τελεσταίο, ενώ το όγδοο περισσότερο σημαντικό ψηφίο του τελεσταίου λαμβάνει την τιμή την οποία περιείχε το Carry Flag.

Οι εντολές RLC και RRC λειτουργούν με παρόμοιο τρόπο με τις εντολές RL και RR, με μόνη διαφορά ότι το Carry Flag δεν δίνει την τιμή του στην κύλιση, αλλά απλά λαμβάνει την τιμή του περισσότερο σημαντικού ψηφίου ή του λιγότερο σημαντικού ψηφίου του τελεσταίου, ανάλογα με το εάν πραγματοποιείται αριστερή ή δεξιά κύλιση. Η ίδια τιμή λαμβάνεται και από το λιγότερο σημαντικό ψηφίο ή το περισσότερο σημαντικό ψηφίο του τελεσταίου, ανάλογα και πάλι με το εάν πραγματοποιείται αριστερή ή δεξιά κύλιση.

Οι εντολές SL και SR λειτουργούν και αυτές με παρόμοιο τρόπο με τις εντολές RL και RR, με τη διαφορά ότι στην αριστερή κύλιση το λιγότερο σημαντικό ψηφίο αντικαθίσταται από 0, ενώ στη δεξιά κύλιση το ίδιο συμβαίνει στο περισσότερο σημαντικό ψηφίο.

Η εντολή αντικατάστασης bit SWAP αντικαθιστά τα 4 περισσότερο σημαντικά ψηφία του τελεσταίου με τα 4 λιγότερο σημαντικά ψηφία και το αντίστροφο, η εντολή BIT αναστρέφει ένα συγκεκριμένο bit του οποίου η θέση δίνεται στον έναν τελεσταίο, στον καταχωρητή που δίνεται στον δεύτερο τελεσταίο, και οι εντολές SET και RES θέτουν ως 1 και 0 αντίστοιχα, ένα bit στον καταχωρητή, των οποίων οι τιμές της θέσης και του καταχωρητή δίνονται από τους δύο τελεσταίους της εντολής.

4.2.3.5 Αναφορά στις εντολές μετάβασης

Οι εντολές μετάβασης JUMP, CALL, RET και RST χρησιμοποιούνται για τον έλεγχο της ροής εκτέλεσης ενός προγράμματος, αλλάζοντας τον μετρητή προγράμματος σε μια νέα διεύθυνση. Η εντολή JUMP θα κάνει τον επεξεργαστή να μεταβεί αμέσως στην καθορισμένη από τον τελεσταίο διεύθυνση, εάν ικανοποιείται η προϋπόθεση που θέτεται ανάλογα με τον κωδικό εντολής (όπως για παράδειγμα η τιμή του Zero Flag να είναι 0) και να αρχίσει να εκτελεί εντολές από αυτήν τη διεύθυνση.

Η εντολή CALL πραγματοποιεί κλήση υπορουτίνας, δηλαδή ωθεί πρώτα τη διεύθυνση της εντολής η οποία ακολουθεί την εντολή CALL στη στοίβα και στη συνέχεια μεταβαίνει στην καθορισμένη από τον τελεσταίο διεύθυνση και εκκινεί την εκτέλεση εντολών από αυτήν τη διεύθυνση.

Η εντολή RET χρησιμοποιείται στο τέλος μίας υπορουτίνας για επιστροφή στο κυρίως πρόγραμμα, λαμβάνοντας την προηγούμενη τιμή του Program Counter από τη στοίβα και θέτοντάς την σε αυτόν, εφόσον ικανοποιείται η προϋπόθεση η οποία θέτεται από τον κωδικό της εντολής.

Η εντολή RST είναι παρόμοια με την CALL, με τη διαφορά πως αντί να δοθεί η τιμή του Program Counter κατευθείαν από τον τελεσταίο, δίνεται η διεύθυνση μνήμης στην οποία βρίσκεται η τιμή του Program Counter από τον τελεσταίο.

4.2.3.6 Αναφορά στις εντολές ελέγχου επεξεργαστή

Οι διάφορες εντολές ελέγχου επεξεργαστή (DI, EI, HALT) περιλαμβάνουν λειτουργίες όπως ενεργοποίηση και απενεργοποίηση διακοπών (Interrupts) και διακοπή λειτουργίας του επεξεργαστή. Στα πλαίσια της εφαρμογής του εξομοιωτή, θέτουν κάποιες μεταβλητές στον κώδικα οι οποίες διαβάζονται σε κάθε κύκλο λειτουργίας του επεξεργαστή, και ανάλογα με την τιμή τους, πραγματοποιούνται οι αντίστοιχες λειτουργίες.

4.2.3.7 Σύνοψη του σετ εντολών του Game Boy

Συμπερασματικά, το σετ εντολών του Game Boy περιλαμβάνει μια ποικιλία εντολών που επιτρέπουν στον επεξεργαστή να μετακινεί δεδομένα μεταξύ καταχωρητών και μνήμης, να χειρίζεται δεδομένα προκειμένου να εκτελεί υπολογισμούς και λειτουργίες και να ελέγχει τη ροή της εκτέλεσης στη μνήμη του συστήματος. Αυτές οι εντολές παρέχουν στον προγραμματιστή ενός παιχνιδιού τα απαραίτητα εργαλεία για την επεξεργασία και τον χειρισμό δεδομένων για την ανάπτυξη ενός προγράμματος Game Boy. Οι προγραμματιστές εξομοιωτών πρέπει να μεριμνούν για την σωστή υλοποίηση των συγκεκριμένων εντολών στους εξομοιωτές τους, με σκοπό την επίτευξη του μεγαλύτερου βαθμού ακριβείας του εξομοιωτή.

4.2.4 Παρουσίαση του τρόπου λειτουργίας των διακοπών στο Game Boy (Interrupts)

Η ανάλυση της λειτουργίας του επεξεργαστή του Game Boy θα ήταν ελλιπής δίχως αναφορά στις διακοπές (interrupts) που πραγματοποιούνται κατά τη λειτουργία του συστήματος, καθώς και την επεξήγηση της σημασίας τους. Το παρόν υποκεφάλαιο έχει ως στόχο την ανάλυση της σημασίας των διακοπών από τον αναγνώστη, καθώς και την αναφορά στις πιο συχνές διακοπές, έτσι ώστε να αποκτηθούν οι κατάλληλες γνώσεις για την ενσωμάτωσή τους στον εξομοιωτή.

Οι διακοπές επιτρέπουν στο Game Boy να ολοκληρώνει σημαντικές λειτουργίες, όπως είναι η αλλαγή καρέ στην οθόνη, ή το πάτημα ενός κουμπιού από τον χρήστη. Ο τρόπος με τον οποίο πραγματοποιούνται αυτές οι λειτουργίες επιτυγχάνεται με την εγγραφή στον καταχωρητή επισήμανσης διακοπών (IF – Interrupt Flag Register) στη θέση μνήμης 0xFF0F του Game Boy, ο οποίος ελέγχεται διαρκώς από τον επεξεργαστή. Όταν υπάρξει εγγραφή της τιμής 1 σε κάποια θέση bit του συγκεκριμένου επεξεργαστή, τότε στον επόμενο έλεγχο για διακοπή ο επεξεργαστής θα αποθηκεύσει τη θέση μνήμης στην οποία βρίσκεται ο μετρητής προγράμματος (Program Counter) στη στοίβα, και θα ανατρέξει σε συγκεκριμένη θέση μνήμης στην οποία είναι γραμμένη η υπορουτίνα που θα

εκτελεστεί. Η θέση μνήμης της υπορουτίνας διαφέρει ανάλογα με τον τύπο διακοπής που εκτελείται. Μετά το πέρας της εκτέλεσης της υπορουτίνας, ο επεξεργαστής φορτώνει ξανά το μετρητή προγράμματος από τη στοίβα και συνεχίζει κανονικά την εκτέλεσή του. Οι διάφοροι τύποι διακοπών, οι συνθήκες εκτέλεσής τους, και οι θέσεις μνήμης των υπορουτινών τους, παρουσιάζονται στον πίνακα 5.

Θέση Bit στον καταχωρητή επισήμανσης διακοπών (IF Register)	Τύπος διακοπής	Συνθήκη εκτέλεσης	Θέση μνήμης υπορουτίνας
Bit 0	Κατακόρυφου διαστήματος κενού (VBlank)	Η μονάδα επεξεργασίας εικόνας (PPU) εισέρχεται σε λειτουργία κατακόρυφου διαστήματος κενού	0x0040
Bit 1	Καταχωρητή κατάστασης (STAT)	Τιμή 1 σε μία από τις θέσεις bit 3-6 του καταχωρητή κατάστασης LCD (LCD STAT)	0x0048
Bit 2	Χρονομέτρου (Timer)	Υπερχείλιση καταχωρητή χρονομέτρου (TIMA) στη θέση μνήμης 0xFF05 (Τιμή μεγαλύτερη του 0xFF)	0x0050
Bit 3	Σειριακής επικοινωνίας (Serial)	Ολοκλήρωση σειριακής μεταφοράς (αποστολή ή λήψη 8 bits στον καταχωρητή SB στη θέση μνήμης 0xFF01)	0x0058
Bit 4	Εισόδου χρήστη (Joypad)	Τιμή 1 σε μία από τις θέσεις bit 0-3 του καταχωρητή εισόδου χρήστη (Joypad) στη θέση μνήμης 0xFF00	0x0060

Πίνακας 5 - Διακοπές που χρησιμοποιούνται στο Game Boy [33]

Οι διακοπές μπορούν να ενεργοποιηθούν ή να απενεργοποιηθούν μέσω του κύριου καταχωρητή ενεργοποίησης διακοπών (IME – Interrupt Master Enable Register), ο οποίος βρίσκεται στη θέση μνήμης. Όταν ο συγκεκριμένος καταχωρητής έχει την τιμή 0, όλες οι διακοπές απενεργοποιούνται. Συνεπώς, πριν από την εκτέλεση κάποιας διακοπής, πρέπει να ελέγχεται η τιμή του συγκεκριμένου καταχωρητή από τον εξομοιωτή.

Τέλος, εκτός από τον καταχωρητή ενεργοποίησης διακοπών, ένας τρόπος με τον οποίο επιτυγχάνεται η απενεργοποίηση συγκεκριμένων διακοπών, είναι με χρήση του καταχωρητή ενεργοποίησης διακοπών (IE – Interrupt Enable Register) στη θέση μνήμης 0xFFFF. Όταν υπάρχει η

τιμή 1 στις θέσεις bit 0-4, οι διακοπές ενεργοποιούνται, ενώ για την τιμή 0 απενεργοποιούνται. Οι θέσεις και τύποι των διακοπών αντιστοιχούν στις ίδιες θέσεις και τύπους διακοπών, με των δύο πρώτων στηλών του πίνακα 5.

4.2.5 Αναφορά στη λειτουργία Game Boy Color (CGB)

Η λειτουργία Game Boy Color (CGB) είναι μια σημαντική αναβάθμιση του αρχικού Game Boy (DMG), η οποία εισήγαγε αρκετές νέες δυνατότητες. Αυτές περιλαμβάνουν έγχρωμη οθόνη με δυνατότητα ταυτόχρονης εμφάνισης έως και 56 χρωμάτων ταυτόχρονα (από μια παλέτα 32.768 χρωμάτων), 3 φορές περισσότερη μνήμη (32 KB σε σύγκριση με τα 8 KB του αρχικού Game Boy) και ταχύτερο επεξεργαστή, ο οποίος έχει τη δυνατότητα να εκτελείται σε χρονισμό 8.38MHz (διπλάσιος σε σύγκριση με του αρχικού Game Boy, ο οποίος εκτελείται σε χρονισμό 4.19MHz), για τα παιχνίδια τα οποία τον υποστηρίζουν. [13]

Για να χρησιμοποιηθούν οι δυνατότητες CGB από ένα πρόγραμμα, πρέπει πρώτα να υπάρχει μία συγκεκριμένη τιμή στη θέση μνήμης 0x0143. Συνήθως, η τιμή 0x80 χρησιμοποιείται για παιχνίδια που υποστηρίζουν την εκτέλεση και σε λειτουργία CGB, αλλά και σε DMG, ενώ η τιμή 0xC0 χρησιμοποιείται για παιχνίδια που εκτελούνται μόνο σε λειτουργία CGB. Σε περίπτωση διαφορετικής τιμής, τα παιχνίδια εκτελούνται αποκλειστικά σε λειτουργία του αρχικού Game Boy (DMG). [34]

Ενώ η λειτουργία CGB προσφέρει μία αναβαθμισμένη εμπειρία παιχνιδιού μέσα από τις ανεπτυγμένες δυνατότητές της, δεν θα αναπτυχθεί στα πλαίσια της παρούσας εργασίας, αλλά συμπεριλαμβάνεται αναφορικά για λόγους ολοκληρωμένης περιγραφής των πιθανών μελλοντικών δυνατοτήτων του εξομοιωτή. Συνεπώς, αποτελεί μία σημαντική πιθανή μελλοντική προσθήκη στον εξομοιωτή.

4.3 Ανάλυση λειτουργίας υποσυστήματος μονάδας επεξεργασίας εικόνας (Picture Processing Unit - PPU)

Στο παρόν κεφάλαιο θα παρουσιαστεί σε αναλυτική μορφή η λειτουργία του υποσυστήματος μονάδας επεξεργασίας εικόνας (Picture Processing Unit - PPU) του Game Boy. Η κατανόηση της λειτουργίας της μονάδας επεξεργασίας εικόνας είναι απαραίτητη για την ανάπτυξη ενός εξομοιωτή Game Boy που έχει τη δυνατότητα να αναπαράγει με ακρίβεια την εμπειρία παιχνιδιού της αυθεντικής συσκευής. Επιπλέον, καθώς η μονάδα επεξεργασίας εικόνας είναι ένα πολύπλοκο υποσύστημα, η κατανόηση των λειτουργιών που αναλύονται στο παρόν κεφάλαιο μπορεί επίσης να ωφελήσει τους προγραμματιστές στη βελτιστοποίηση της απόδοσης των εξομοιωτών τους, όσον αφορά την ταχύτητά τους αλλά και την ακρίβειά τους.

4.3.1 Συνοπτική περιγραφή λειτουργιών της μονάδας επεξεργασίας εικόνας

Η μονάδα επεξεργασίας εικόνας είναι ένα από τα πιο σημαντικά μέρη της αρχιτεκτονικής του Game Boy, καθώς είναι υπεύθυνη για την εμφάνιση της τελικής εικόνας στην οθόνη. Η κύρια λειτουργία της μονάδας επεξεργασίας εικόνας είναι η λήψη δεδομένων από την περιοχή μνήμης Video Ram του Game

Boy (θέσεις μνήμης 0x8000 έως 0xA000), με χρήση των οποίων δημιουργούνται τα εικονοστοιχεία (pixels) που αποτυπώνονται στην οθόνη LCD της συσκευής.

Η μονάδα επεξεργασίας εικόνας είναι επίσης υπεύθυνη για την εκτέλεση λειτουργιών, όπως η κύλιση του οπτικού πεδίου της οθόνης πάνω στον χάρτη πλακιδίων (tile map), η αναστροφή των γραφικών στοιχείων και η αλλαγή της παλέτας των χρωμάτων που χρησιμοποιούνται στην οθόνη.

4.3.2 Ανάλυση λειτουργιών καταχωρητών της μονάδας επεξεργασίας εικόνας

Η επικοινωνία μεταξύ ενός προγράμματος και του υποσυστήματος μονάδας επεξεργασίας εικόνας, γίνεται με χρήση ορισμένων καταχωρητών. Ένας προγραμματιστής μπορεί να αλληλοεπιδράσει με το υποσύστημα καταχωρώντας ή διαβάζοντας συγκεκριμένες τιμές από αυτούς τους καταχωρητές. Για παράδειγμα, μπορεί να αλλάξει την εμφανιζόμενη εικόνα, να αλλάξει τον χάρτη πλακιδίων που χρησιμοποιείται, να δημιουργήσει διάφορα οπτικά εφέ εν μέσω ενός καρέ, ή ακόμα και να διακόψει τη λειτουργία της μονάδας επεξεργασίας εικόνας, κλείνοντας την οθόνη LCD.

Το παρόν κεφάλαιο αποσκοπεί στην ανάλυσή τους, για λόγους σωστής ενσωμάτωσης όλων των λειτουργιών που προσφέρουν στον εξομοιωτή, και συνεπώς να υπάρξει μεγαλύτερη ακρίβεια και συμβατότητα με το μέγιστο δυνατό αριθμό παιχνιδιών.

4.3.2.1 Παρουσίαση καταχωρητή ελέγχου LCD (LCD Control - LCDC)

Ο καταχωρητής ελέγχου LCD (LCDC) βρίσκεται στη θέση μνήμης 0xFF40, και είναι ο κύριος τρόπος ελέγχου των λειτουργιών του υποσυστήματος της μονάδας επεξεργασίας εικόνας. Παρέχει στον προγραμματιστή την δυνατότητα αλλαγής ορισμένων ρυθμίσεων σχετικών με τον έλεγχο του υποσυστήματος, οι οποίες παρουσιάζονται στο παρόν υποκεφάλαιο.

Καθώς πρόκειται για έναν πολύ σημαντικό καταχωρητή, οι ρυθμίσεις του οποίου επηρεάζουν άμεσα το γραφικό αποτέλεσμα στην οθόνη του εξομοιωτή, και κάθε πρόγραμμα μπορεί να τον μεταβάλλει πολλές φορές, ήταν απαραίτητη η σωστή ενσωμάτωσή στον εξομοιωτή όλων των λειτουργιών που παρέχει. Σε περίπτωση που υπάρξει λανθασμένη ενσωμάτωση, η τελική εικόνα που παρουσιάζεται στην οθόνη του εξομοιωτή θα διαφέρει από τη σωστή, καθιστώντας τον εξομοιωτή μη λειτουργικό.

Οι ρυθμίσεις του καταχωρητή ελέγχου LCD σχετίζονται με την ενεργοποίηση και απενεργοποίηση της οθόνης LCD, την εναλλαγή μεταξύ των διαθέσιμων χαρτών πλακιδίων, την απόκρυψη ή την εμφάνιση του παρασκηνίου, του παραθύρου και των γραφικών στοιχείων, και το μέγεθος των γραφικών στοιχείων (8x8 ή 16x16). Ο καταχωρητής ελέγχου LCD αποτελείται από 8 bits, και το κάθε ένα από αυτά είναι υπεύθυνο για μία ρύθμιση, όπως φαίνονται στον πίνακα 6:

Bit	Λειτουργία
7	Ενεργοποίηση οθόνης LCD (0=Όχι, 1=Ναι)
6	Περιοχή μνήμης χάρτη πλακιδίων παραθύρου (0=9800-9BFF, 1=9C00-9FFF)
5	Ενεργοποίηση παραθύρου (0=Όχι, 1=Ναι)
4	Περιοχή μνήμης δεδομένων πλακιδίων παρασκηνίου και παραθύρου (0=8800-97FF, 1=8000-8FFF)
3	Περιοχή μνήμης χάρτη πλακιδίων παρασκηνίου (0=9800-9BFF, 1=9C00-9FFF)
2	Μέγεθος γραφικών στοιχείων (0=8x8, 1=16x16)
1	Ενεργοποίηση γραφικών στοιχείων (0=Όχι, 1=Ναι)
0	Ενεργοποίηση παρασκηνίου και παραθύρου (0=Όχι, 1=Ναι)

Πίνακας 6 - Ρυθμίσεις καταχωρητή ελέγχου LCD [35]

4.3.2.2 Παρουσίαση καταχωρητή κατάστασης LCD (LCD Status - STAT)

Ο καταχωρητής κατάστασης LCD (LCD Status - STAT), είναι άλλος ένας σημαντικός καταχωρητής που σχετίζεται με το υποσύστημα της μονάδας επεξεργασίας εικόνας. Επομένως, η ανάλυση της λειτουργίας του καθίσταται κρίσιμη για τους σκοπούς δημιουργίας της εφαρμογής. Ο καταχωρητής κατάστασης LCD βρίσκεται στη θέση μνήμης 0xFF41, και πραγματοποιεί δύο σημαντικές λειτουργίες. Η πρώτη λειτουργία είναι η ένδειξη της τρέχουσας κατάστασης λειτουργίας της μονάδας επεξεργασίας εικόνας, η οποία παρέχεται από τις τιμές των τριών λιγότερο σημαντικών bit του καταχωρητή. Οι τέσσερις καταστάσεις λειτουργίας της μονάδας επεξεργασίας εικόνας θα αναλυθούν λεπτομερώς στο κεφάλαιο 4.3.3. Ανάλογα με την τρέχουσα κατάσταση λειτουργίας της μονάδας επεξεργασίας εικόνας, ορισμένα κομμάτια της μνήμης δεν είναι διαθέσιμα στον επεξεργαστή, καθώς χρησιμοποιούνται από την μονάδα επεξεργασίας εικόνας. Συνεπώς, για την αποφυγή λανθασμένων εικόνων, είναι σημαντική η σωστή υλοποίηση της ένδειξης της τρέχουσας κατάστασης λειτουργίας της μονάδας επεξεργασίας εικόνας στον εξομοιωτή.

Η δεύτερη λειτουργία του καταχωρητή κατάστασης LCD σχετίζεται με τις διακοπές που προκαλούνται από την κεντρική μονάδα επεξεργασίας. Κατά την μετάβαση κατάστασης λειτουργίας της μονάδας επεξεργασίας εικόνας σε μία νέα, η θέση του bit που αντιστοιχεί στη νέα κατάσταση λειτουργίας λαμβάνει τιμή 1, ζητώντας διακοπή από τον επεξεργαστή, η οποία θα ολοκληρωθεί όταν πληρούνται και όλες οι υπόλοιπες προϋποθέσεις για διακοπή όπως αναφέρονται στο κεφάλαιο 4.2.4. Με την εκτέλεση της διακοπής, ο επεξεργαστής θα αποθηκεύσει τον τρέχον μετρητή προγράμματος στη στοιβά και θα τον αντικαταστήσει με τη νέα τιμή που δόθηκε ανάλογα με τον τύπο διακοπής για να συνεχίσει την εκτέλεση του προγράμματος από εκείνη τη διεύθυνση μνήμης.

Τέλος, πρέπει να αναφερθεί πως η μονάδα επεξεργασίας εικόνας συγκρίνει διαρκώς την τιμή της τρέχουσας σειράς εικονοστοιχείων στο καρτέ (καταχωρητής LY στη θέση μνήμης 0xFF44), με την τιμή που περιέχεται στη θέση μνήμης 0xFF45 (καταχωρητής LYC – LY Compare). Όταν οι δύο αυτές

τιμές είναι ίσες, τα bit στις θέσεις 2 και 6 του καταχωρητή κατάστασης LCD λαμβάνουν την τιμή 1 και ζητείται διακοπή από τον επεξεργαστή. Η συγκεκριμένη λειτουργία παρέχει στους προγραμματιστές τη δυνατότητα να εκτελέσουν διακοπή σε μία συγκεκριμένη σειρά με σκοπό τη δημιουργία διάφορων γραφικών εφέ. [36] Στα πλαίσια της δημιουργίας του εξομοιωτή είναι απαραίτητη μόνο η αλλαγή των bits του καταχωρητή και η πυροδότηση των διακοπών όταν πληρούνται οι παραπάνω προϋποθέσεις. Στον πίνακα 7 παρουσιάζεται η λειτουργία που παρέχει το κάθε bit. Αξίζει να σημειωθεί πως το περισσότερο σημαντικό ψηφίο (bit 7) δε χρησιμοποιείται.

Bit	Λειτουργία
6	Κλήση διακοπής LYC=LY (0=Όχι, 1=Ναι)
5	Κλήση διακοπής λειτουργίας 2 – OAM STAT (0=Όχι, 1=Ναι)
4	Κλήση διακοπής λειτουργίας 1 – VBlank (0=Όχι, 1=Ναι)
3	Κλήση διακοπής λειτουργίας 0 – HBlank (0=Όχι, 1=Ναι)
2	Έλεγχος LY=LYC (0=Όχι, 1=Ναι)
1-0	Τρέχουσα κατάσταση λειτουργίας μονάδας επεξεργασίας εικόνας: 00 - Λειτουργία 0 (HBlank) 01 - Λειτουργία 1 (VBlank) 10 - Λειτουργία 2 (Αναζήτηση OAM) 11 - Λειτουργία 3 (Εμφάνιση εικονοστοιχείων στην οθόνη)

Πίνακας 7 - Ρυθμίσεις καταχωρητή ελέγχου LCD [37]

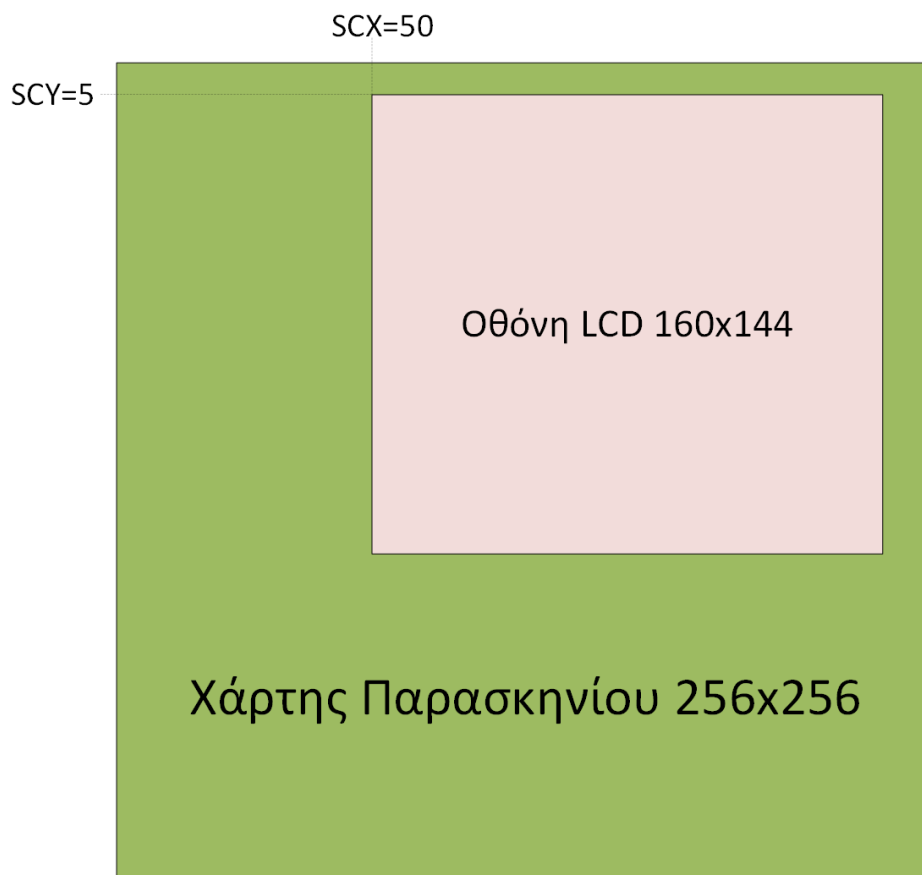
4.3.2.3 Παρουσίαση καταχωρητών κύλισης (SCY, SCX, WY, WX)

Για την καλύτερη κατανόηση της λειτουργίας της μονάδας επεξεργασίας εικόνας, είναι απαραίτητη η αναφορά στους καταχωρητές κύλισης οι οποίοι είναι υπεύθυνοι για τη θέση της οθόνης πάνω στους χάρτες παρασκηνίου και παραθύρου. Οι καταχωρητές κύλισης είναι σημαντικοί για την διαδικασία εμφάνισης της εικόνας στην οθόνη, καθώς διαβάζονται συνεχώς από την μονάδα επεξεργασίας εικόνας. Το παρόν υποκεφάλαιο παρουσιάζει τους καταχωρητές κύλισης που χρησιμοποιούνται για την εμφάνιση των γραφικών στην οθόνη και τον τρόπο λειτουργίας τους.

Οι καταχωρητές κύλισης Y και X (SCY, SCX) βρίσκονται στις θέσεις μνήμης 0xFF42 και 0xFF43 αντίστοιχα. Είναι υπεύθυνοι για τη θέση της οθόνης (μεγέθους 160x144 εικονοστοιχείων) πάνω στον χάρτη παρασκηνίου (μεγέθους 256x256 εικονοστοιχείων). Οι συντεταγμένες που προσδιορίζονται από αυτούς τους καταχωρητές αντιστοιχούν στην πάνω αριστερή γωνία της οθόνης και λαμβάνουν τιμές από 0 έως 255. Εάν κάποιο μέρος του τετραγώνου της οθόνης έχει συντεταγμένες μεγαλύτερες από 255, τότε πραγματοποιείται περιτύλιξη του χάρτη παρασκηνίου, δηλαδή η εμφάνιση συνεχίζεται από την αντίθετη μεριά του χάρτη, άρα εκείνες οι συντεταγμένες θα λάβουν τιμή X-256.

Για παράδειγμα, έστω ότι οι καταχωρητές κύλισης SCX και SCY έχουν τις τιμές 50 και 5. Σε αυτή την περίπτωση, η πάνω αριστερή γωνία του τετραγώνου 160x144 της οθόνης που παρουσιάζεται

στον παίκτη θα βρίσκεται στις συντεταγμένες (50,5) του χάρτη παρασκηνίου. Η τοποθέτηση της οθόνης πάνω στον χάρτη παρασκηνίου εάν οι τιμές των καταχωρητών SCY και SCX είναι 5 και 50 αντίστοιχα, φαίνεται στο σχήμα 6.



Σχήμα 6 - Τοποθέτηση οθόνης στον χάρτη παρασκηνίου για SCY=5, SCX=50

Εκτός από τους καταχωρητές κύλισης SCY και SCX, εξίσου σημαντικοί είναι και οι καταχωρητές συντεταγμένων παραθύρου WY και WX. Οι καταχωρητές συντεταγμένων παραθύρου ορίζουν τις συντεταγμένες της πάνω αριστερής γωνίας του χάρτη παραθύρου με παρόμοιο τρόπο με αυτόν των καταχωρητών κύλισης. Η διαφορά σε αυτή την περίπτωση είναι πως ο χάρτης παραθύρου έχει μέγεθος 256x256, και επικαλύπτει τον χάρτη παρασκηνίου. Το παράθυρο δεν εμφανίζεται, εκτός εάν το bit στη θέση 5 του καταχωρητή ελέγχου LCD έχει την τιμή 1, όπου θα εμφανιστεί πάνω από τον χάρτη παρασκηνίου.

Οι τιμές των καταχωρητών WY και WX υποδηλώνουν τις συντεταγμένες στην οθόνη LCD όπου εμφανίζεται η πάνω αριστερή γωνία του χάρτη παραθύρου. Ο καταχωρητής WY λαμβάνει τιμές στο εύρος 0-143, ενώ ο καταχωρητής WX έχει την ιδιαιτερότητα ότι λαμβάνει τιμές στο εύρος 0-166. Ο καταχωρητής WX λαμβάνει τιμές μεγαλύτερες του 159 (μεγαλύτερες του πλάτους της οθόνης LCD), καθώς για WX=7 το παράθυρο εφάπτεται στην αριστερή πλευρά της οθόνης, ενώ για τιμές WX μικρότερες του 7 το παράθυρο εμφανίζεται αριστερά από την οθόνη, αποκρύπτοντας μέρος του. [25]

Ο χάρτης παραθύρου χρησιμοποιείται συνήθως για τα μενού των παιχνιδιών ή τα γραφικά που έχουν αμετάβλητη θέση στο παιχνίδι, όπως για παράδειγμα η ζωή του παίκτη ή το σκορ του.

4.3.2.4 Ανάλυση τρόπου επιλογής παλέτας χρωμάτων

Ο τελευταίος καταχωρητής του οποίου η γνώση είναι απαραίτητη για τη δημιουργία της μονάδας επεξεργασίας εικόνας στον εξομοιωτή βρίσκεται στη θέση μνήμης 0xFF47, και είναι ο καταχωρητής δεδομένων παλέτας παρασκηνίου (Background Palette Data - BGP).

Ο καταχωρητής δεδομένων παλέτας παρασκηνίου είναι υπεύθυνος για την κατανομή χρωμάτων στα πλακίδια που βρίσκονται στους χάρτες παρασκηνίου και παραθύρου. Κάθε πλακίδιο σε αυτούς τους χάρτες αποτελείται από 8x8 εικονοστοιχεία, όπου το κάθε ένα από αυτά συμβολίζεται στον χάρτη με μία τιμή στο εύρος τιμών 0-3. Ανάλογα με την τιμή του εικονοστοιχείου, η μονάδα επεξεργασίας εικόνας θα του αναθέσει ένα από τα τέσσερα διαθέσιμα χρώματα (λευκό, ανοιχτό γκρι, σκούρο γκρι, μαύρο) σύμφωνα με τις τιμές των bit του καταχωρητή δεδομένων παλέτας παρασκηνίου.

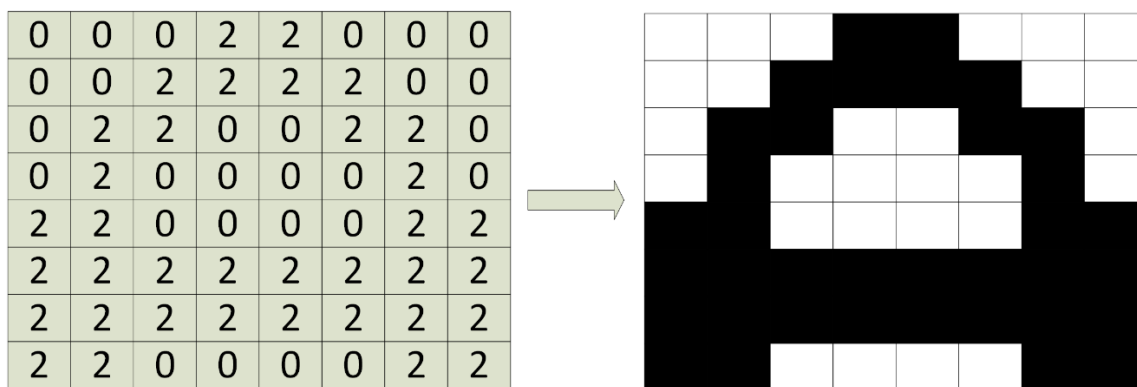
Θέσεις Bit καταχωρητή δεδομένων παλέτας παρασκηνίου	Λειτουργία
7-6	Χρώμα για εικονοστοιχεία με τιμή 3
5-4	Χρώμα για εικονοστοιχεία με τιμή 2
3-2	Χρώμα για εικονοστοιχεία με τιμή 1
1-0	Χρώμα για εικονοστοιχεία με τιμή 0

Πίνακας 8 - Τρόπος ανάθεσης χρώματος στα εικονοστοιχεία από καταχωρητή δεδομένων παλέτας παρασκηνίου [38]

Τιμή Bit καταχωρητή δεδομένων παλέτας παρασκηνίου	Χρώμα
0	Λευκό
1	Ανοιχτό γκρι
2	Σκούρο γκρι
3	Μαύρο

Πίνακας 9 - Χρώματα που κατανέμει ο καταχωρητής δεδομένων παλέτας παρασκηνίου [38]

Για παράδειγμα, έστω ότι ο καταχωρητής δεδομένων παλέτας παρασκηνίου περιέχει τον δυαδικό αριθμό 10110100 (δεκαεξαδικός: 0xB4). Ο συγκεκριμένος δυαδικός αριθμός θα χωριστεί σε «10», «11», «01» και «00» (δεκαδικοί: 2,3,1,0). Επομένως, σύμφωνα με τον πίνακα 8, τα εικονοστοιχεία που έχουν τιμή 3 θα αποκτήσουν σκούρο γκρι χρώμα, ενώ τα εικονοστοιχεία που έχουν τιμή 2 θα αποκτήσουν μαύρο χρώμα. Τέλος, τα εικονοστοιχεία με τιμές 1 και 0 θα λάβουν ανοιχτό γκρι και λευκό χρώμα αντίστοιχα. Στο σχήμα 7 παρουσιάζεται μετατροπή από τις τιμές του χάρτη παρασκηνίου σε εικονοστοιχεία, τα οποία εμφανίζονται στην οθόνη σύμφωνα με την παλέτα του παραδείγματος.



Σχήμα 7 - Παράδειγμα μετατροπής τιμών του χάρτη παρασκηνίου σε χρώματα

4.3.3 Ανάλυση κύκλου λειτουργίας της μονάδας επεξεργασίας εικόνας

Ο κύκλος που εκτελεί η μονάδα επεξεργασίας εικόνας είναι σχετικά περίπλοκος. Το συγκεκριμένο υποκεφάλαιο αποσκοπεί στην πλήρη ανάλυση του με σκοπό τη διευκόλυνση της κατανόησης του κύκλου, έτσι ώστε να είναι δυνατή η δημιουργία μιας λειτουργικής μονάδας επεξεργασίας εικόνας στον εξομοιωτή. Αρχικά, θα παρουσιαστεί μια συνοπτική περίληψη του κύκλου της λειτουργίας της μονάδας επεξεργασίας εικόνας, και στη συνέχεια θα αναλυθούν περαιτέρω τα βήματα τα οποία πραγματοποιεί σε κάθε κατάσταση λειτουργίας της.

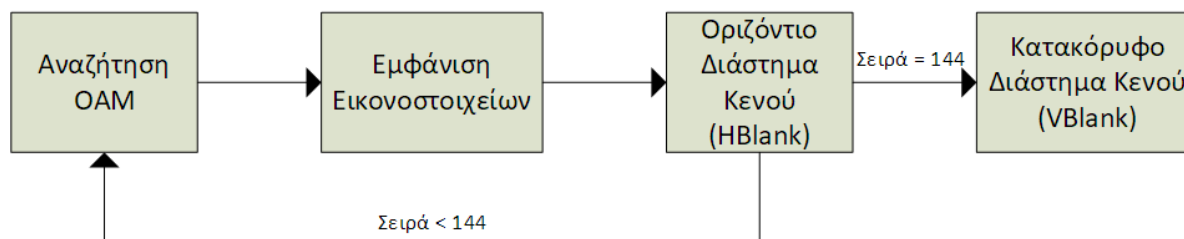
Η εικόνα που παράγεται από την μονάδα επεξεργασίας εικόνας δημιουργείται ανά σειρά εικονοστοιχείων, ξεκινώντας από την κορυφή και συνεχίζοντας προς τα κάτω. Αυτό σημαίνει πως με την εκκίνησή της, η μονάδα επεξεργασίας εικόνας θα ξεκινήσει να σχεδιάζει την πρώτη σειρά της οθόνης, και έπειτα θα προχωρήσει στις επόμενες με τη σειρά, ώσπου να έχει σχεδιάσει και τις 144 σειρές, δημιουργώντας ένα καρέ (frame). Αυτή η διαδικασία επαναλαμβάνεται 59.7 φορές ανά δευτερόλεπτο, δηλαδή το Game Boy έχει συχνότητα εικόνων 59.7 καρέ/δευτερόλεπτο (Frames Per Second – FPS). Από αυτή την τιμή προκύπτει πως ο χρόνος που χρειάζεται η μονάδα επεξεργασίας εικόνας για να εμφανίσει ένα καρέ είναι 0,0167 δευτερόλεπτα. Ο συγκεκριμένος χρόνος λαμβάνεται υπόψιν κατά την ανάπτυξη του εξομοιωτή, καθώς εάν ο εξομοιωτής απαιτήσει περισσότερο χρόνο για το σχεδιασμό ενός καρέ, τότε θα είναι πιο αργός από το αυθεντικό Game Boy. Αντίστοιχα, εάν είναι πιο γρήγορος, τότε τα παιχνίδια θα εκτελούνται σε πιο γρήγορη ταχύτητα.

Η μονάδα επεξεργασίας εικόνας κατά τη διάρκεια ενός κύκλου της μπορεί να βρίσκεται σε 1 από 4 καταστάσεις λειτουργίας, οι οποίες παρουσιάζονται στον πίνακα 10:

Αριθμός Λειτουργίας	Όνομα Κατάστασης	Διάρκεια Κατάστασης (dots)
0	Οριζόντιο διάστημα κενού (Horizontal Blanking - HBlank)	85 έως 208
1	Κατακόρυφο διάστημα κενού (Vertical Blanking - VBlank)	4560
2	Αναζήτηση OAM	80
3	Εμφάνιση εικονοστοιχείων	168 έως 291

Πίνακας 10 - Καταστάσεις λειτουργίας της μονάδας επεξεργασίας εικόνας [37]

Για να εμφανίσει 1 σειρά εικονοστοιχείων, η μονάδα επεξεργασίας εικόνας θα εναλλαχθεί ανάμεσα στις καταστάσεις αναζήτησης OAM, εμφάνισης εικονοστοιχείων και οριζοντίου διαστήματος κενού (HBlank), με αυτή τη σειρά. Έπειτα, θα προχωρήσει στην επόμενη σειρά επαναλαμβάνοντας αυτές τις καταστάσεις. Για την ολοκλήρωση μίας σειράς, η μονάδα επεξεργασίας εικόνας χρειάζεται χρονικό διάστημα 456 dots. Ως dot ορίζουμε το ελάχιστο χρονικό διάστημα μέσα στο οποίο η μονάδα επεξεργασίας εικόνας δύναται να εμφανίσει 1 εικονοστοιχείο. [37] Όταν η μονάδα επεξεργασίας εικόνας ολοκληρώσει την εμφάνιση της 144^{ης} σειράς, θα προχωρήσει στην κατάσταση λειτουργίας κατακόρυφου διαστήματος κενού (VBlank), στην οποία θα παραμείνει για χρονικό διάστημα ίσο με 10 σειρές (4560 dots). Με το πέρας της κατάστασης λειτουργίας κατακόρυφου διαστήματος κενού, η μονάδα επεξεργασίας εικόνας θα προχωρήσει στην εμφάνιση του επόμενου καρέ.



Σχήμα 8 - Σειρά καταστάσεων λειτουργίας μονάδας επεξεργασίας εικόνας (PPU) για την εμφάνιση ενός καρέ

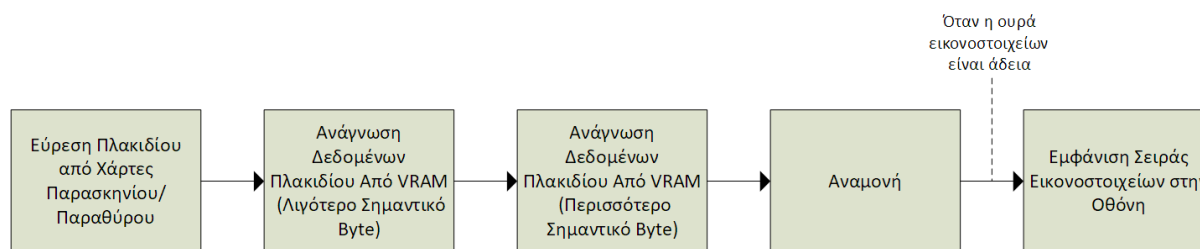
4.3.3.1 Ανάλυση κατάστασης λειτουργίας 2: Αναζήτηση OAM (OAM Search)

Η κατάσταση λειτουργίας αναζήτησης OAM είναι η πρώτη κατάσταση λειτουργίας στην οποία μεταβαίνει η μονάδα επεξεργασίας εικόνας κατά την εμφάνιση μίας καινούργιας σειράς εικονοστοιχείων. Σε αυτή την κατάσταση λειτουργίας, η μονάδα επεξεργασίας εικόνας αναζητάει στη μνήμη χαρακτηριστικών γραφικών στοιχείων (OAM) τα γραφικά στοιχεία (παίκτης, εχθροί, κλπ.) των οποίων οι συντεταγμένες Y (όπως αναφέρεται στο υποκεφάλαιο 4.1.7) περιέχουν τη σειρά την οποία εμφανίζει η μονάδα επεξεργασίας εικόνας σύμφωνα με τον καταχωρητή τρέχουσας σειράς (LY), και επιλέγει για εμφάνιση τα πρώτα 10 γραφικά στοιχεία από τη μνήμη των οποίων οι παραπάνω προϋποθέσεις πληρούνται, ενώ τα επόμενα αγνοούνται. Η αναζήτηση ολοκληρώνεται όταν σαρωθούν και τα 40 γραφικά στοιχεία που περιέχονται στη μνήμη. [39]

Για παράδειγμα, εάν ένα γραφικό στοιχείο στη μνήμη έχει μέγεθος 8×16 , και η συντεταγμένη Y στη μνήμη χαρακτηριστικών γραφικών στοιχείων έχει την τιμή 16, τότε εφόσον η παρούσα σειρά εικονοστοιχείων που εμφανίζεται από την μονάδα επεξεργασίας εικόνας είναι μεταξύ των γραμμών 1 και 16 αυτό το στοιχείο θα επιλεγεί για εμφάνιση, εκτός εάν έχουν ήδη επιλεγεί 10 γραφικά στοιχεία στην παρούσα σειρά.

4.3.3.2 Ανάλυση κατάστασης λειτουργίας 3: Εμφάνιση εικονοστοιχείων

Η κατάσταση λειτουργίας εμφάνισης εικονοστοιχείων είναι αρκετά πιο περίπλοκη από όλες τις υπόλοιπες καταστάσεις λειτουργίας της μονάδας επεξεργασίας εικόνας. Κατά τη διάρκεια αυτής της κατάστασης λειτουργίας, πραγματοποιείται η εμφάνιση των εικονοστοιχείων της σειράς στην οθόνη LCD, μέσω ενός κύκλου βημάτων. Για αυτό το λόγο είναι σημαντική η σωστή υλοποίησή της, καθώς ακόμη και με μικρές παρεκκλίσεις από τη σωστή, θα εμφανίζεται λάθος εικόνα στην οθόνη του εξομοιωτή. Το συγκεκριμένο υποκεφάλαιο αποσκοπεί στην ανάλυση των βημάτων που πραγματοποιεί η συγκεκριμένη λειτουργία, έτσι ώστε να πραγματοποιηθεί η σωστή υλοποίησή της στον εξομοιωτή. Η κατάσταση λειτουργίας εμφάνισης εικονοστοιχείων αποτελείται από τα 5 βήματα που παρουσιάζονται στο σχήμα 9.



Σχήμα 9 - Βήματα κατάστασης λειτουργίας εμφάνισης εικονοστοιχείων [40]

Το πρώτο βήμα είναι η προσπέλαση του χάρτη παρασκηνίου ή παραθύρου, ανάλογα με τις τιμές που περιέχει ο καταχωρητής ελέγχου LCD, όπως αυτές παρουσιάζονται στο υποκεφάλαιο 4.3.2.1. Από τις τιμές του καταχωρητή, θα δοθεί και το εύρος θέσεων μνήμης που φιλοξενούν το χάρτη που χρειαζόμαστε. Μόλις οριστούν οι θέσεις μνήμης και η επιλογή ανάμεσα σε χάρτη παρασκηνίου ή παραθύρου, τότε η μονάδα επεξεργασίας εικόνας θα λάβει από αυτόν την τιμή που περιέχεται στη θέση πλακιδίου το οποίο πρέπει να εμφανιστεί. Η συγκεκριμένη θέση πλακιδίου θα υπολογιστεί σύμφωνα με τις τιμές των καταχωρητών κύλισης, αλλά και τα πλακίδια που έχουν εμφανιστεί ήδη στο συγκεκριμένο καρέ. Η τιμή που περιέχεται στη θέση μνήμης του πλακιδίου που θα εμφανιστεί, θα μας δώσει τη θέση μνήμης στη VRAM η οποία περιέχει τα απαραίτητα δεδομένα για την εμφάνιση του πλακιδίου.

Στο δεύτερο βήμα, θα γίνει προσπέλαση της θέσης μνήμης της VRAM που βρέθηκε από το πρώτο βήμα, προσθέτοντας την τιμή που περιείχε η θέση μνήμης του πλακιδίου στην τιμή της διεύθυνσης μνήμης από την οποία ξεκινάει η VRAM, η οποία συνήθως είναι η $0x8000$. Σε αυτή τη θέση μνήμης στη VRAM, θα περιέχεται το λιγότερο σημαντικό byte των δεδομένων πλακιδίου που

πρέπει να εμφανιστεί. Σε αυτό το σημείο πρέπει να σημειωθεί πως η μονάδα επεξεργασίας εικόνας γνωρίζει ποια γραμμή εικονοστοιχείων εμφανίζει, από τις 8 του πλακιδίου. Με αυτόν τον τρόπο, θα αναζητήσει το byte στην κατάλληλη θέση. Για παράδειγμα, εάν εμφανίζεται η τρίτη σειρά ενός πλακιδίου, τότε η θέση μνήμης VRAM που θα προσπελαστεί θα υπολογιστεί ως $0x8000 + \text{θέση μνήμης πλακιδίου στη VRAM} + 4$, καθώς κάθε σειρά εικονοστοιχείων καταλαμβάνει 2 θέσεις μνήμης στη VRAM.

Το τρίτο βήμα αποτελεί συνέχεια του δεύτερου βήματος, καθώς θα αναζητήσουμε στην επόμενη θέση μνήμης της VRAM το περισσότερο σημαντικό byte των δεδομένων πλακιδίου που πρέπει να εμφανιστεί. Έχοντας πλέον και τα δύο bytes των δεδομένων πλακιδίου, μπορούμε να τα αποκωδικοποιήσουμε για να εξάγουμε το χρώμα του κάθε ενός από τα 8 εικονοστοιχεία της σειράς που πρέπει να εμφανιστεί, σύμφωνα με τις τιμές του πίνακα 8. Η αποκωδικοποίηση γίνεται μετατρέποντας το υψηλό και το χαμηλό byte στο δυαδικό σύστημα, και ενθέτοντας τα bits των δύο αριθμών. Δηλαδή το πρώτο bit του υψηλού byte θα ακολουθηθεί από το πρώτο bit του χαμηλού byte, το οποίο θα ακολουθηθεί από το δεύτερο bit του υψηλού byte, και θα συνεχιστεί αντιστοίχως μέχρι να δημιουργηθεί η νέα σειρά των 16 bits. Μέσω αυτής της διαδικασίας θα δημιουργηθούν οι τιμές των χρωμάτων του κάθε εικονοστοιχείου της σειράς. Για παράδειγμα, το πρώτο εικονοστοιχείο θα λάβει τα πρώτα δύο bits, το δεύτερο τα επόμενα, κλπ.

Το τέταρτο βήμα είναι η αναμονή μέχρι να αδειάσει η ουρά εικονοστοιχείων τα οποία αναμένουν την εμφάνισή τους στην οθόνη. Σε αυτό το βήμα, δεν πραγματοποιείται κάποια ενέργεια, και αυτό επαναλαμβάνεται ώσπου η ουρά 8 θέσεων να είναι κενή, έτσι ώστε αυτή να λάβει τα 8 νέα bytes που δημιουργήθηκαν από τη σειρά των 16 bits στο προηγούμενο βήμα.

4.3.3.3 *Ανάλυση κατάστασης λειτουργίας 1: Οριζόντιο διάστημα κενού (HBlank)*

Η κατάσταση λειτουργίας οριζοντίου διαστήματος κενού είναι μία κατάσταση λειτουργίας στην οποία η μονάδα επεξεργασίας εικόνας παραμένει αδρανής μετά το πέρας της εμφάνισης της σειράς εικονοστοιχείων. Όσο διαρκεί η κατάσταση λειτουργίας διαστήματος κενού, ο επεξεργαστής αποκτάει πρόσβαση στις περιοχές μνήμης OAM και Video Ram. Αυτό δίνει τη δυνατότητα στους προγραμματιστές παιχνιδιών να πραγματοποιήσουν αλλαγές στην εικόνα κατά τη διάρκεια εμφάνισης ενός καρέ, χωρίς να δημιουργηθούν σφάλματα στα γραφικά που εμφανίζονται στην εικόνα (glitches). Για αυτό το λόγο, η ενσωμάτωση της συγκεκριμένης αδράνειας στον εξομοιωτή είναι απαραίτητη.

Κατά τη διάρκεια του οριζοντίου διαστήματος κενού, η μονάδα επεξεργασίας εικόνας παραμένει αδρανής για χρόνο ίσο με 376 dots μείον το χρόνο σε dots που χρειάστηκε για να ολοκληρωθεί η κατάσταση λειτουργίας εμφάνισης εικονοστοιχείων. Για παράδειγμα, εάν η κατάσταση λειτουργίας αναζήτησης OAM χρειαστεί 291 dots για την ολοκλήρωσή της, τότε η κατάσταση οριζοντίου διαστήματος κενού θα διαρκέσει 85 dots.

4.3.3.4 Ανάλυση κατάστασης λειτουργίας 2: Κατακόρυφο διάστημα κενού (VBlank)

Η κατάσταση λειτουργίας κατακόρυφου διαστήματος κενού είναι παρόμοια με την κατάσταση λειτουργίας οριζοντίου διαστήματος κενού, καθώς και σε αυτή την κατάσταση λειτουργίας η μονάδα επεξεργασίας εικόνας παραμένει αδρανής. Η συγκεκριμένη κατάσταση ξεκινάει μετά το πέρας του σχεδιασμού ολόκληρου του καρέ, και η διάρκειά της είναι 4560 dots (1.1 milliseconds), τα οποία αντιστοιχούν στο 10% της διάρκειας εμφάνισης ενός καρέ.

Η μεγαλύτερη διάρκειά της σε σύγκριση με την κατάσταση λειτουργίας οριζοντίου διαστήματος κενού, την καθιστά ιδιαίτερα σημαντική για τους προγραμματιστές παιχνιδιών, καθώς σε αυτή την κατάσταση μπορούν να γίνουν περισσότερες αλλαγές στις περιοχές μνήμης που σχετίζονται με τα γραφικά.

Ένα παραπάνω χαρακτηριστικό της συγκεκριμένης κατάστασης, είναι ότι εκτελείται διακοπή όταν η μονάδα επεξεργασίας εικόνας εισέρχεται σε αυτή. Αυτή η διακοπή επιτρέπει στους προγραμματιστές να ορίσουν συγκεκριμένα βήματα τα οποία θα εκτελούνται κατά τη διάρκεια της κατάστασης λειτουργίας κατακόρυφου διαστήματος κενού, όπως η φόρτωση νέων γραφικών στοιχείων, η αλλαγή των πλακιδίων παρασκηνίου στον χάρτη παρασκηνίου και η αλλαγή της παλέτας χρώματος. Ωστόσο, είναι σημαντικό να ολοκληρωθούν όλες οι εντολές μέσα στη διάρκεια της συγκεκριμένης κατάστασης, καθώς μόλις αυτή ολοκληρωθεί, η προσπέλαση του επεξεργαστή στις απαραίτητες περιοχές μνήμης θα διακοπεί.

4.4 Ανάλυση λειτουργίας της εισόδου χρήστη

Ένα βασικό υποσύστημα το οποίο πρέπει να ενσωματωθεί στον εξομοιωτή για την επίτευξη των βασικών λειτουργιών του Game Boy, είναι το υποσύστημα της εισόδου χρήστη. Δίχως την ύπαρξη ενός τέτοιου συστήματος, ο χρήστης δε θα έχει τη δυνατότητα αλληλεπίδρασης με τα παιχνίδια, καθιστώντας τον εξομοιωτή ελλιπή και μη λειτουργικό. Το παρόν κεφάλαιο περιέχει μία αναλυτική περιγραφή του υποσυστήματος το οποίο είναι υπεύθυνο για την αλληλεπίδραση του χρήστη με το Game Boy, καθώς η γνώση της λειτουργίας του υποσυστήματος καθίσταται απαραίτητη για τη δημιουργία του κομματιού της εισόδου χρήστη του εξομοιωτή.

Η είσοδος χρήστη του Game Boy αποτελείται από 8 βασικά πλήκτρα. Αυτά είναι τα: A, B, Start, Select, καθώς και 4 βελάκια κατεύθυνσης (πάνω, κάτω, δεξιά, αριστερά) τα οποία επιτρέπουν την κίνηση ενός παίκτη προς 8 κατευθύνσεις. Με το πάτημα ενός πλήκτρου από τον χρήστη, η τιμή του καταχωρητή εισόδου χρήστη (Joypad) στη θέση μνήμης 0xFF00 θα αλλάξει. Η νέα τιμή του καταχωρητή εξαρτάται από τον συνδυασμό των πλήκτρων που πατιούνται από τον χρήστη, σύμφωνα με τον πίνακα 11:

Θέση Bit	Τιμή Bit
7	Δε χρησιμοποιείται
6	Δε χρησιμοποιείται
5	Διάβασμα πλήκτρων δράσης (0 = Επιλογή)
4	Διάβασμα πλήκτρων κατεύθυνσης (0 = Επιλογή)
3	Πλήκτρα κάτω/Start (0 = Πατημένο)
2	Πλήκτρα πάνω/Select (0 = Πατημένο)
1	Πλήκτρα αριστερά/B (0 = Πατημένο)
0	Πλήκτρα δεξιά/A (0 = Πατημένο)

Πίνακας 11 - Τιμές bits καταχωρητή εισόδου χρήση [26]

Ο πίνακας 10 μας παρουσιάζει το σκοπό του κάθε bit στον καταχωρητή εισόδου. Τα bits του καταχωρητή έχουν την τιμή 1, εκτός εάν πραγματοποιηθούν συγκεκριμένες ενέργειες οι οποίες θέτουν στα bits την τιμή 0. Τα bits στις θέσεις 6 και 7 δε χρησιμοποιούνται, ενώ με τα bits στις θέσεις 4 και 5 γίνεται η επιλογή διαβάσματος ανάμεσα στα πλήκτρα δράσης (A, B, Start, Select) και στα 4 πλήκτρα κατεύθυνσης, θέτοντας την τιμή 0. Όταν γραφτεί η τιμή 0 σε κάποια από αυτές τις θέσεις, θα εκτελεστεί διακοπή εισόδου χρήστη, και θα διαβαστούν οι τιμές των bits στις θέσεις 0 έως 3, οι οποίες λαμβάνουν την τιμή 1 ανάλογα με τα πατημένα πλήκτρα. Για παράδειγμα, εάν γραφτεί η τιμή 0 στη θέση bit 5, τότε θα διαβαστούν οι τιμές των θέσεων 0 έως 3, και εάν είναι πατημένο κάποιο πλήκτρο δράσης, η ανάλογη τιμή στην κατάλληλη θέση θα έχει την τιμή 0. Αντιθέτως, εάν γραφτεί η τιμή 0 στη θέση bit 4, τότε θα διαβαστούν οι τιμές των θέσεων 0 έως 3, και εάν είναι πατημένο κάποιο πλήκτρο κατεύθυνσης, η θέση του στις θέσεις 0 έως 3 θα έχει την τιμή 0.

5 Υλοποίηση της εφαρμογής εξομοιωτή Game Boy σε περιβάλλον Android Studio

Το παρόν κεφάλαιο περιέχει μία αναλυτική περιγραφή της διαδικασίας ανάπτυξης της εφαρμογής του εξομοιωτή Game Boy για συσκευές Android. Σκοπός του κεφαλαίου είναι η παρουσίαση της συγκεκριμένης διαδικασίας με σαφή τρόπο και η παροχή ενός ολοκληρωμένου οδηγού για την υλοποίηση παρόμοιων εγχειρημάτων.

Αρχικά, παρουσιάζονται τα εργαλεία και οι εκδόσεις που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Έπειτα, αναλύεται η διαδικασία δημιουργίας της διεπαφής χρήστη (User Interface) μέσω παρουσίασης της διαδικασίας της υλοποίησης των γραφικών στοιχείων που χρησιμοποιούνται στην εφαρμογή, καθώς και αιτιολόγησης των σχεδιαστικών επιλογών της. Στη συνέχεια, πραγματοποιείται ανάλυση της δομής του κώδικα Kotlin στην οποία παρουσιάζεται η υλοποίηση των λειτουργιών των υποσυστημάτων που αναλύθηκαν στο κεφάλαιο της ανάλυσης των τεχνικών χαρακτηριστικών του Game Boy. Τέλος, παρατίθενται οι διαδικασίες ελέγχου και αποσφαλμάτωσης του εξομοιωτή, οι δυσκολίες που προέκυψαν κατά τη διάρκεια της ανάπτυξης της εφαρμογής, καθώς και οι τρόποι αντιμετώπισής τους.

Σε αυτό το σημείο είναι σημαντικό να σημειωθεί πως η παροχή των αρχείων των παιχνιδιών πρέπει να γίνει από τους ίδιους τους χρήστες, με νόμιμη απόκτησή τους, καθώς η παρούσα εφαρμογή δεν περιέχει κανένα αρχείο παιχνιδιού.

5.1 Εγκατάσταση απαραίτητων εργαλείων για την ανάπτυξη του εξομοιωτή

Για την ανάπτυξη του εξομοιωτή χρησιμοποιήθηκε το λογισμικό Android Studio με έκδοση Dolphin 2021.3.1. Απαραίτητη προϋπόθεση για την ανάπτυξη της εφαρμογής με χρήση γλώσσας προγραμματισμού Kotlin ήταν η εγκατάσταση του Java Development Kit (JDK). Η εγκατεστημένη έκδοση JDK κατά τη διάρκεια ανάπτυξης της εφαρμογής είναι η JDK 11.0.13.

Για το χτίσιμο (build) της εφαρμογής χρησιμοποιήθηκε το εργαλείο Gradle, το οποίο παρέχεται στην εφαρμογή Android Studio. Η έκδοση Gradle που χρησιμοποιήθηκε κατά την ανάπτυξη της εφαρμογής είναι η έκδοση 7.4. Για την παροχή λειτουργιών σχετικών με το χτίσιμο και τον έλεγχο εφαρμογών Android από το εργαλείο Gradle, είναι απαραίτητη η χρήση του πρόσθετου Gradle Android (Gradle Android Plugin). Στην ανάπτυξη του εξομοιωτή χρησιμοποιήθηκε η έκδοση 7.3.0.

Τέλος, η εφαρμογή στοχεύει σε SDK (Software Development Kit) με έκδοση 33, και η χαμηλότερη έκδοση SDK που υποστηρίζεται από την εφαρμογή είναι η 21. Το SDK ορίζει την έκδοση Android για την οποία έχει χτιστεί η εφαρμογή. Για νεότερες εκδόσεις, προσφέρονται νέες λειτουργίες στην πλατφόρμα Android ή καταργούνται απαρχαιωμένες λειτουργίες. Η στοχευμένη έκδοση (Target SDK Version) αναφέρεται στην έκδοση πάνω στην οποία χτίστηκε η εφαρμογή, άρα και αυτή με τη μεγαλύτερη συμβατότητα, ενώ παρέχεται και η λειτουργία υποστήριξης χαμηλότερων εκδόσεων SDK

(άρα και παλαιότερων συσκευών Android) μέσω της επιλογής χαμηλότερης έκδοσης SDK (Min SDK Version).

5.2 Δημιουργία διεπαφής χρήστη (User Interface)

Σκοπός του παρόντος κεφαλαίου είναι η παρουσίαση του τρόπου δημιουργίας της διεπαφής χρήστη της εφαρμογής στο Android Studio. Επίσης, θα αιτιολογηθούν οι επιλογές που έγιναν κατά το σχεδιασμό της διεπαφής χρήστη και θα αναφερθούν οι δυσκολίες που προέκυψαν κατά τη δημιουργία της.

5.2.1 Ανάλυση απαιτήσεων διεπαφής χρήστη

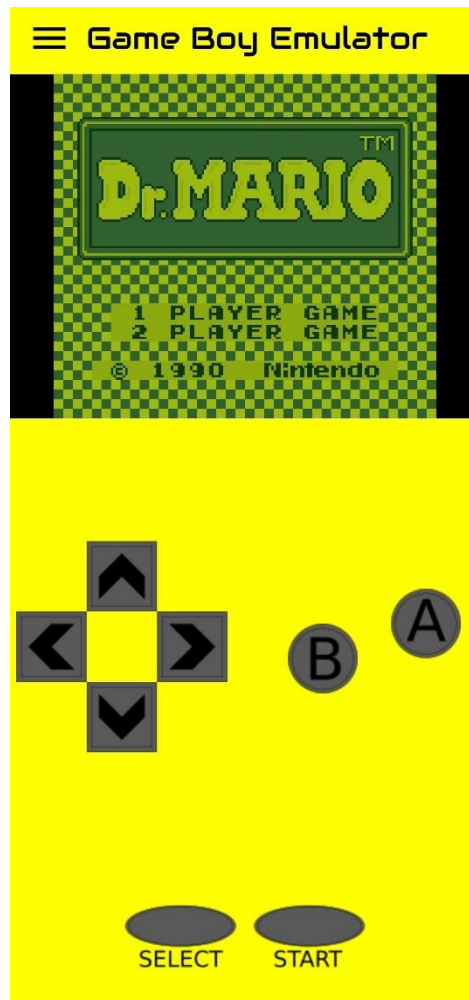
Στο παρόν υποκεφάλαιο θα πραγματοποιηθεί η παρουσίαση των απαιτήσεων οι οποίες λήφθηκαν υπόψη για το σχεδιασμό της διεπαφής χρήστη. Η εμφάνιση της διεπαφής χρήστη κατέχει σημαντικό ρόλο στην εφαρμογή, καθώς καθιστά το πρώτο πράγμα με το οποίο αλληλοεπιδρά ο χρήστης, και συνεπώς δημιουργεί την σημαντική πρώτη εντύπωση. Επίσης, εκτός από την πρώτη εντύπωση, ο χρήστης βρίσκεται σε διαδικασία αλληλεπίδρασης με την διεπαφή χρήστη καθ' όλη τη διάρκεια της χρήσης του εξομοιωτή. Επομένως, η παροχή μίας καλαίσθητης και ευχάριστης εμπειρίας χρήσης θα προκαλέσει θετικές εντυπώσεις στον χρήστη όσον αφορά τον εξομοιωτή, οδηγώντας τον σε περισσότερο χρόνο χρήσης της εφαρμογής.

Αρχικά, μία σημαντική σχεδιαστική απαίτηση είναι αυτή του μοντέρνου σχεδιασμού, ο οποίος παράλληλα πρέπει να παραπέμπει στην παιχνομηχανή Game Boy, δημιουργώντας στον χρήστη μία εμπειρία εφάμιλλη με αυτήν της αυθεντικής κονσόλας. Για αυτό το λόγο κρίθηκε απαραίτητος ένας μιμησιαστικός σχεδιασμός της βασικής οθόνης της εφαρμογής, στην οποία ο χρήστης καταναλώνει τον περισσότερο χρόνο χρήσης στην εφαρμογή. Συνεπώς, τα στοιχεία τα οποία παρουσιάζονται μέσω της κεντρικής οθόνης στο χρήστη είναι η οθόνη του Game Boy, τα απαραίτητα πλήκτρα (A, B, Start, Select και βελάκια κατεύθυνσης), και η μπάρα τίτλου η οποία περιέχει το όνομα της εφαρμογής. Τέλος, προστέθηκε και το εικονίδιο του πλάγιου μενού, μέσω του οποίου ο χρήστης έχει διάφορες δυνατότητες, μεταξύ άλλων, το να φορτώσει ένα παιχνίδι και να παραμετροποιήσει την εμφάνιση του εξομοιωτή μέσα από το μενού των ρυθμίσεων.

Όσον αφορά την εμφάνιση της διεπαφής χρήστη, ένα άλλο απαιτούμενο χαρακτηριστικό είναι η δυνατότητα παραμετροποίησής της για λόγους ευχρηστίας και προτίμησης του κάθε χρήστη. Δηλαδή, είναι απαραίτητο ο χρήστης να έχει τη δυνατότητα να αλλάζει την εμφάνιση της εφαρμογής, συμπεριλαμβανομένων των θέσεων των πλήκτρων, αλλά και του χρώματος του παρασκηνίου της εφαρμογής. Λόγοι για τους οποίους ο χρήστης είναι πιθανό να αλλάξει τη θέση των πλήκτρων μπορεί να πηγάζουν από το μέγεθος της οθόνης της συσκευής του, στην οποία οι προκαθορισμένες θέσεις μπορεί να αποβούν άβολες ως προς τη συνεχή χρήση της εφαρμογής, αλλά και για λόγους προσβασιμότητας (accessibility) προς όλους τους χρήστες, απαίτηση η οποία είναι βασική πλέον για όλες τις μοντέρνες εφαρμογές.

Το μεγάλο εύρος του μεγέθους των οθονών που τρέχουν το λειτουργικό σύστημα Android, μπορεί να δημιουργήσει προβλήματα συμβατότητας σε κάποιες από αυτές. Για παράδειγμα, κάποια πλήκτρα μπορεί να μην είναι στις κατάλληλες θέσεις λόγω του μικρού μεγέθους μίας οθόνης, ή αντιθέτως, να βρίσκονται πολύ μακριά μεταξύ τους. Για την αντιμετώπιση αυτού του προβλήματος, υπάρχει η απαίτηση του μεταβαλλόμενου μεγέθους των κουμπιών, ανάλογα με το μέγεθος της κάθε οθόνης, και οι αποστάσεις μεταξύ τους να καθορίζονται με βάση τα άκρα αυτής. Με αυτόν τον τρόπο, ελαχιστοποιούνται τα προβλήματα συμβατότητας που δύνανται να προκύψουν λόγω του εύρους του μεγέθους των οθονών.

Τέλος, είναι σημαντική η προσθήκη ανάδρασης (feedback) στα πλήκτρα της εφαρμογής, με την εκτέλεση δόνησης από τη συσκευή όταν ο χρήστης πατάει κάποιο πλήκτρο. Μέσω αυτής της λειτουργίας, επιτυγχάνεται η προσομοίωση του πατήματος ενός πραγματικού πλήκτρου, όπως αυτά του αυθεντικού Game Boy, καθώς κάτι τέτοιο είναι αδύνατο στις οθόνες αφής.



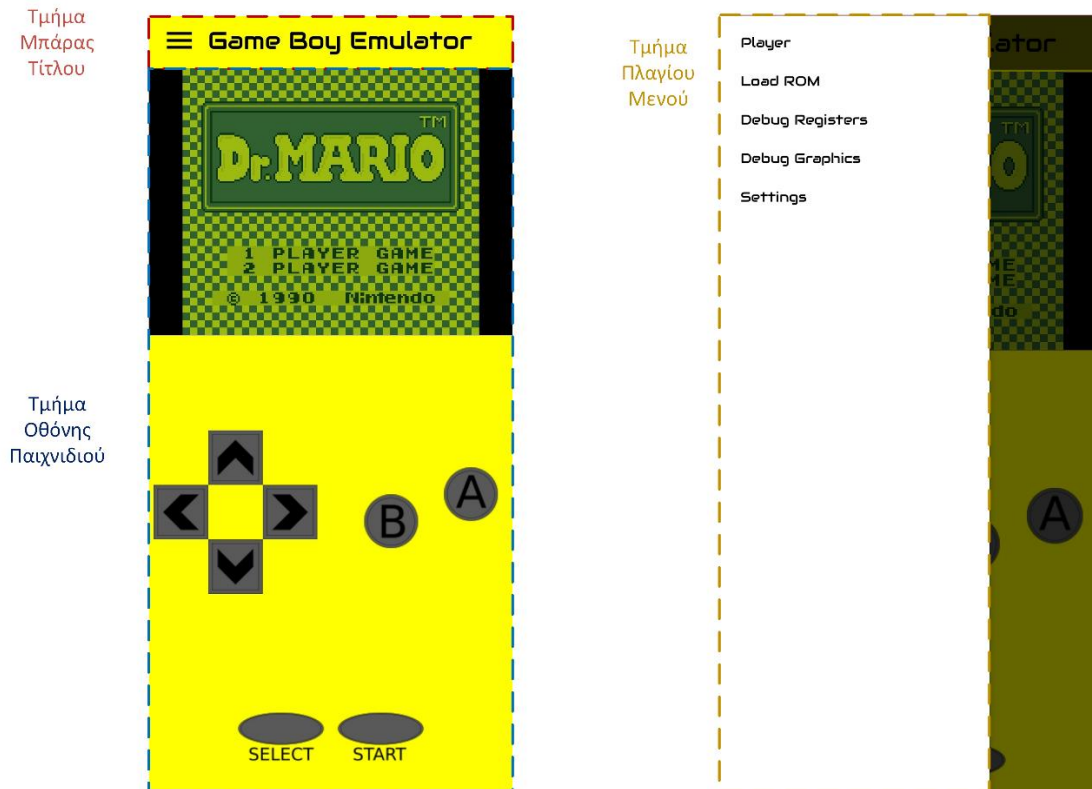
Σχήμα 10 - Η κεντρική οθόνη της εφαρμογής

5.2.2 Παρουσίαση τρόπου δημιουργίας της διεπαφής χρήστη στο Android Studio

Έπειτα από τον καθορισμό των απαιτήσεων της διεπαφής χρήστη της εφαρμογής του εξομοιωτή Game Boy, το επόμενο στάδιο είναι η δημιουργία της. Ο σκοπός του παρόντος υποκεφαλαίου είναι η παρουσίαση της διαδικασίας ανάπτυξης της διεπαφής χρήστη στο περιβάλλον Android Studio. Με την ανάγνωση του παρόντος υποκεφαλαίου, ο αναγνώστης θα αποκτήσει τις απαραίτητες γνώσεις σχετικά με τη δομή και τον τρόπο λειτουργίας της διεπαφής χρήστη μίας εφαρμογής Android.

Η δημιουργία των οθονών που εμφανίζονται στον χρήστη μίας εφαρμογής Android, πραγματοποιείται κατά κύριο λόγο με τη χρήση αρχείων XML (Extensible Markup Language). Τα συγκεκριμένα αρχεία περιέχουν όλες τις ιδιότητες που κατέχουν τα στοιχεία τα οποία εμφανίζονται στο χρήστη, όπως για παράδειγμα το μέγεθος των εικόνων των πλήκτρων, το μέγεθος της κεντρικής οθόνης του Game Boy, αλλά και τις γραμματοσειρές και τα χρώματα που εμφανίζονται στις οθόνες και στα μενού του εξομοιωτή. Στα πλαίσια της ανάπτυξης μίας εφαρμογής Android, οι προαναφερθείσες οθόνες αποκαλούνται «δραστηριότητες» (activities). Η κάθε δραστηριότητα συνήθως περιέχει μέσα της διάφορα επιπλέον στοιχεία. Τα στοιχεία αυτά αποκαλούνται «τμήματα» (fragments). Συνεπώς, για τη δημιουργία της διεπαφής του εξομοιωτή, πραγματοποιήθηκε η δημιουργία αρχείων XML, το κάθε ένα από τα οποία είναι υπεύθυνο για τον ορισμό των ιδιοτήτων της κάθε οθόνης και τμημάτων που πρέπει να εμφανίσει η εφαρμογή.

Η κύρια οθόνη η οποία παρουσιάζεται στον χρήστη της εφαρμογής, αποτελεί μία δραστηριότητα. Η συγκεκριμένη δραστηριότητα περιέχει δοχεία (containers), στα οποία φιλοξενούνται διάφορα τμήματα τα οποία εναλλάσσονται μεταξύ τους ανάλογα με τις επιλογές του χρήστη. Πιο συγκεκριμένα, στη δραστηριότητα της κύριας οθόνης περιέχονται 3 δοχεία. Με την έναρξη της εφαρμογής, τα δοχεία αυτά φιλοξενούν τα τμήματα της μπάρας τίτλου, της οθόνης παιχνιδιού, αλλά και του πλαγίου μενού. Τα τμήματα της μπάρας τίτλου και του πλαγίου μενού παραμένουν αμετάβλητα κατά τη διάρκεια της εφαρμογής.



Σχήμα 11 - Παρουσίαση τμημάτων της διεπαφής χρήστη της εφαρμογής

Το τμήμα της οθόνης παιχνιδιού μπορεί να αντικατασταθεί από άλλα τμήματα, όπως το τμήμα οθόνης ρυθμίσεων ή επιλογής φόρτωσης παιχνιδιού, ανάλογα με τις επιλογές που πραγματοποιεί ο χρήστης κατά τη χρήση της εφαρμογής. Δηλαδή, εάν ο χρήστης κατά τη διάρκεια του παιχνιδιού επιλέξει μέσα από το πλάγιο μενού να εμφανίσει το μενού των ρυθμίσεων, θα πρέπει το τμήμα το οποίο είναι υπεύθυνο για την οθόνη ρυθμίσεων να αντικαταστήσει το τμήμα της οθόνης παιχνιδιού, συνεπώς και ο χρήστης να παύει να βλέπει την οθόνη του παιχνιδιού και να εισέλθει στο μενού ρυθμίσεων. Με την εναλλαγή αυτών των τμημάτων, το τμήμα της οθόνης παιχνιδιού «παγώνει» όση ώρα ο χρήστης χρησιμοποιεί την οθόνη ρυθμίσεων. Με την έξοδό του από την οθόνη ρυθμίσεων, αυτή αντικαθίσταται από την οθόνη παιχνιδιού, όπου και ο χρήστης μπορεί να συνεχίσει το παιχνίδι.

5.2.3 Παρουσίαση προβλημάτων κατά τη δημιουργία της διεπαφής χρήστη

Κατά την ανάπτυξη της διεπαφής χρήστη της εφαρμογής του εξομοιωτή, παρουσιάστηκαν ορισμένα προβλήματα και προκλήσεις οι οποίες έπρεπε να ξεπεραστούν. Το συγκεκριμένο υποκεφάλαιο αποσκοπεί στην παρουσίαση αυτών των προβλημάτων και των προκλήσεων, παραθέτοντας παράλληλα τις λύσεις που χρησιμοποιήθηκαν.

Κατά τον σχεδιασμό της διεπαφής χρήστη του εξομοιωτή, παρουσιάστηκε η πρόκληση του διαχωρισμού των μερών του εξομοιωτή. Πιο συγκεκριμένα, λόγω των διαφορετικών στόχων που επιτυγχάνουν τα διάφορα μέρη της εφαρμογής του εξομοιωτή, όπως η εμφάνιση ρυθμίσεων, η φόρτωση νέας κασέτας, ή η διαδικασία εκτέλεσης μίας κασέτας, καθώς και για λόγους ευκολίας ανάπτυξης της

διεπαφής χρήστη της εφαρμογής, ήταν θεμιτό αυτές οι λειτουργίες να χωριστούν σε διαφορετικά μέρη η κάθε μία, τα οποία θα είναι ανεξάρτητα μεταξύ τους. Η επιλογή του διαχωρισμού των μερών, πραγματοποιήθηκε με βάση το στόχο του κάθε μέρους, καθώς ο χρήστης της εφαρμογής δε θα πραγματοποιεί δύο στόχους ταυτόχρονα. Για παράδειγμα, όταν ο χρήστης επιλέξει να φορτώσει μία νέα κασέτα, δε θα πραγματοποιεί αλλαγή ρυθμίσεων του εξομοιωτή παράλληλα. Έτσι, το τμήμα το οποίο είναι υπεύθυνο για το φόρτωμα μιας νέας κασέτας θα μπορεί να λαμβάνει όλο το χώρο της οθόνης, αντί για το τμήμα των ρυθμίσεων της εφαρμογής. Για την επίλυση αυτού του προβλήματος, έγινε η διαίρεση των λειτουργιών του εξομοιωτή στα τμήματα της μπάρας τίτλου, της οθόνης παιχνιδιού, του πλαγίου μενού, της οθόνης ρυθμίσεων, αλλά και σε δύο οθόνες αποσφαλμάτωσης (εντολών επεξεργαστή και γραφικών), οι οποίες χρησιμοποιήθηκαν για σκοπούς αποσφαλμάτωσης του εξομοιωτή.

Ένα σημαντικό πρόβλημα το οποίο παρουσιάστηκε κατά την ανάπτυξη της διεπαφής χρήστη της εφαρμογής, ήταν το πρόβλημα της κλιμάκωσης (scaling) των εικόνων των κουμπιών του εξομοιωτή. Αρχικά, ο εξομοιωτής αναπτύχθηκε για λειτουργία σε έξυπνη συσκευή με υψηλή ανάλυση. Όταν όμως η εφαρμογή φορτώθηκε σε παλαιότερη συσκευή της οποίας η ανάλυση ήταν μικρότερη, παρατηρήθηκε πως τα κουμπιά του εξομοιωτή (A, B, βελάκια κατεύθυνσης), ήταν πολύ μεγάλα σε σχέση με την οθόνη. Αυτό είχε σαν αποτέλεσμα να μην είναι στις σωστές θέσεις και να επικαλύπτουν την οθόνη του παιχνιδιού, καθιστώντας την εφαρμογή μη λειτουργική. Για αυτό το λόγο, κρίθηκε απαραίτητη η δημιουργία εικόνων για κουμπιά των οποίων το μέγεθος να είναι μεταβαλλόμενο, ανάλογα με την ανάλυση της οθόνης της συσκευής στην οποία τρέχει η εφαρμογή. Για να επιτευχθεί αυτό, δημιουργήθηκαν τρεις εικόνες κουμπιών για το κάθε κουμπί, μία σε χαμηλή ανάλυση, μία σε μεσαία ανάλυση, και μία σε υψηλή ανάλυση. Ανάλογα με τον τύπο οθόνης της κάθε συσκευής που εκτελεί την εφαρμογή, χρησιμοποιούνται οι εικόνες με την κατάλληλη ανάλυση.



Σχήμα 12 - Οι εικόνες των κουμπιών της εφαρμογής του εξομοιωτή

Ο σχεδιασμός της διεπαφής χρήστη περιέχει επίσης το πρόβλημα της επιλογής των χρωμάτων της εφαρμογής. Η εμφάνιση, για να είναι σύμφωνη με την απαίτηση του μοντέρνου σχεδιασμού ο οποίος θα παραπέμπει ταυτόχρονα στην αυθεντική κονσόλα Game Boy, πρέπει να αποτελείται από χρώματα τα οποία είναι καλαίσθητα, αλλά και ταυτόχρονα γνώριμα στους χρήστες της εφαρμογής. Λαμβάνοντας υπόψιν αυτές τις απαιτήσεις, έγινε η επιλογή του κίτρινου χρώματος ως βασικό χρώμα παρασκήνιου της εφαρμογής, καθώς παραπέμπει στην κίτρινη έκδοση του Game Boy, η οποία κυκλοφόρησε το 1995. Για την ικανοποίηση και της απαίτησης της δυνατότητας παραμετροποίησης της διεπαφής χρήστη, δίνεται η επιλογή στους χρήστες μέσω της οθόνης ρυθμίσεων να επιλέξουν οι

ίδιοι το χρώμα του παρασκηνίου της εφαρμογής, μέσα από πλήθος χρωμάτων τα οποία αντιστοιχούν σε χρώματα στα οποία κυκλοφόρησε η αυθεντική κονσόλα.

Τέλος, πέρα από την επιλογή των χρωμάτων, παρουσιάστηκε και το πρόβλημα της επιλογής γραμματοσειράς της εφαρμογής. Η γραμματοσειρά της εφαρμογής θα πρέπει να παραμένει σταθερή σε όλες τις οθόνες της εφαρμογής, καθώς εάν διέφερε από οθόνη σε οθόνη, θα επηρεαζόταν αρνητικά η εμπειρία χρήσης της εφαρμογής. Ταυτόχρονα, έπρεπε να γίνει επιλογή γραμματοσειράς η οποία να έχει ύψος το οποίο παραπέμπει σε κονσόλα παιχνιδιών, αλλά να είναι και φιλική προς το διάβασμα. Λαμβάνοντας υπόψιν αυτές τις απαιτήσεις, έγινε η επιλογή της γραμματοσειράς «Luckiest Guy», της οποίας η άδεια επιτρέπει τη χρήση της στην εφαρμογή.

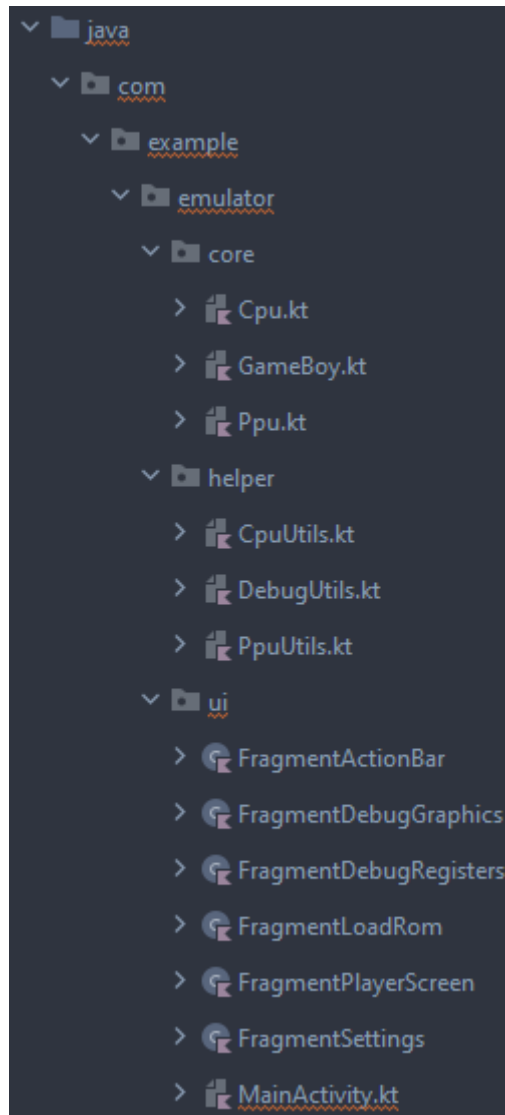
GAME BOY

Σχήμα 13 - Η γραμματοσειρά Luckiest Guy που χρησιμοποιήθηκε στην εφαρμογή

5.3 Σύνοψη δομής κώδικα Kotlin έργου

Το παρόν υποκεφάλαιο αφορά την περιγραφή του κώδικα Kotlin που υλοποιήθηκε στα πλαίσια ανάπτυξης του εξομοιωτή Game Boy. Αποσκοπεί σε μία επεξήγηση της υλοποίησης της λογικής της εφαρμογής του εξομοιωτή, παρουσιάζοντας τον κώδικα που δημιουργήθηκε με βάση τη θεωρία του προηγούμενου κεφαλαίου. Πιο συγκεκριμένα, στα πλαίσια του παρόντος υποκεφαλαίου θα παρουσιαστεί ο κώδικας ο οποίος αφορά τόσο τα υποσυστήματα του επεξεργαστή και της μονάδας επεξεργασίας εικόνας τα οποία είναι υπεύθυνα για τις εσωτερικές λειτουργίες του εξομοιωτή του Game Boy, όσο και τις βοηθητικές συναρτήσεις που χρησιμοποιήθηκαν για την υλοποίηση αυτών των συστημάτων, αλλά και τις περαιτέρω λειτουργίες που απαιτήθηκαν στα πλαίσια ανάπτυξης της Android εφαρμογής. Επομένως, παρέχονται στον αναγνώστη κρίσιμες πληροφορίες για την ανάληψη παρόμοιων εγχειρημάτων, και ταυτόχρονα παρουσιάζονται έμπρακτα τα αποτελέσματα της τεχνικής ανάλυσης του Game Boy.

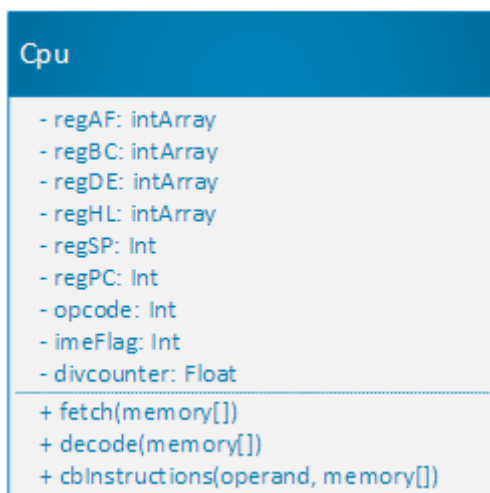
Καθώς δεν είναι δυνατό να αποτυπωθεί ολόκληρος ο κώδικας της εφαρμογής στο παρόν υποκεφάλαιο, θα πραγματοποιηθεί μία γενική παρουσίαση της δομής του κώδικα, με εμβάθυνση στην περιγραφή των πιο σημαντικών εννοιών για καλύτερη κατανόηση από τον αναγνώστη. Πρέπει να αναφερθεί πως η δομή του κώδικα του κάθε προγραμματιστικού έργου είναι διαφορετική, και στο συγκεκριμένο κεφάλαιο παρουσιάζεται απλά μία προσέγγιση της οργάνωσης του κώδικα ενός τέτοιου έργου από τις πολλές που μπορούν να χρησιμοποιηθούν σε παρόμοιες περιπτώσεις.



Σχήμα 14 - Δομή των αρχείων κώδικα του εξομοιωτή

5.3.1 Σύνοψη κώδικα υποσυστήματος επεξεργαστή

Ο κώδικας Kotlin ο οποίος είναι υπεύθυνος για την εκτέλεση της λογικής του επεξεργαστή του Game Boy, περιέχεται μέσα στην κλάση Cpu. Στο παρόν υποκεφάλαιο θα αναλυθούν τα περιεχόμενα της συγκεκριμένης κλάσης, η σημασία τους στη λειτουργία του εξομοιωτή, καθώς και ο τρόπος με τον οποίο αλληλοεπιδρούν με τις υπόλοιπες κλάσεις.



Σχήμα 15 - Διάγραμμα UML της κλάσης Cpu

5.3.1.1 Επεξήγηση μεταβλητών της κλάσης Cpu

Στην κλάση Cpu περιέχονται οι σημαντικότερες μεταβλητές για τη λειτουργία του εξομοιωτή. Αυτές οι μεταβλητές είναι πίνακες ακεραίων 2 θέσεων που περιέχουν τους καταχωρητές που παρουσιάστηκαν στο κεφάλαιο της ανάλυσης των τεχνικών χαρακτηριστικών του Game Boy, regAF, regBC, regDE, και regHL. Στην πρώτη θέση του πίνακα φιλοξενείται η τιμή του πρώτου καταχωρητή, ενώ στη δεύτερη θέση φιλοξενείται η τιμή του δεύτερου καταχωρητή. Για παράδειγμα, στον πίνακα regAF, η πρώτη θέση θα φιλοξενεί τον καταχωρητή A (καταχωρητής συσσώρευσης - Accumulator Register), ενώ η δεύτερη θέση τον καταχωρητή F (καταχωρητής σημαίας - Flag Register). Οι τιμές των καταχωρητών αυτών, αλλάζουν συνεχώς κατά τη λειτουργία του εξομοιωτή.

Πέρα από τις προαναφερθείσες μεταβλητές των καταχωρητών, στην κλάση Cpu φιλοξενούνται επίσης και οι μεταβλητές του μετρητή προγράμματος (Program Counter), του δείκτη στοίβας (Stack Pointer), καθώς και του τρέχοντος κωδικού εντολής (opcode). Αυτές οι μεταβλητές είναι εξίσου σημαντικές με αυτές των καταχωρητών και αποτελούν αναπόσπαστο κομμάτι της λειτουργίας του επεξεργαστή, καθώς είναι απαραίτητες για την εκτέλεση των σωστών εντολών από τις μεθόδους της κλάσης Cpu.

Η κλάση Cpu περιέχει επίσης μεταβλητές οι οποίες είναι υπεύθυνες για την εκτέλεση διακοπών, αλλά και μεταβλητές σχετικές με άλλες λειτουργίες, όπως αυτή του χρονομέτρου. Για παράδειγμα, τέτοιες μεταβλητές είναι οι imeflag (σημαία καταχωρητή ενεργοποίησης διακοπών – Interrupt Master Enable Flag) και divcounter (μετρητής χρονομέτρου).

5.3.1.2 Επεξήγηση μεθόδων της κλάσης Cpu

Η κλάση Cpu, εκτός των μεταβλητών που σχετίζονται με τη λειτουργία του επεξεργαστή, φιλοξενεί και δύο μεθόδους οι οποίες είναι ύψιστης σημασίας για τη σωστή λειτουργία του εξομοιωτή. Οι μέθοδοι αυτές είναι οι fetch() (φέρε) και decode() (αποκωδικοποίηση). Με την εναλλαγή των δύο αυτών

μεθόδων κατά τη λειτουργία του επεξεργαστή, διαβάζεται η μνήμη του Game Boy, και έπειτα εκτελείται η εντολή η οποία διαβάστηκε.

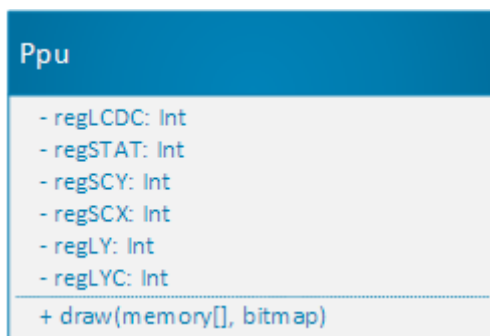
Η μέθοδος `fetch()`, της οποίας η εκτέλεση πραγματοποιείται πρώτη, λαμβάνει ως είσοδο τον πίνακα ακεραίων ο οποίος αποτελεί τη μνήμη του Game Boy (πίνακας ακεραίων 65536 θέσεων). Έχοντας πρόσβαση στη μεταβλητή του μετρητή προγράμματος του επεξεργαστή, τη χρησιμοποιεί για να βρει την θέση της επόμενης εντολής η οποία θα εκτελεστεί. Ως παράδειγμα, κατά την εκκίνηση του Game Boy, ο μετρητής προγράμματος περιέχει την τιμή `0x100`. Αυτό έχει ως αποτέλεσμα, κατά την πρώτη εκτέλεση της `fetch()`, η μέθοδος αυτή να διαβάσει τα περιεχόμενα της θέσης μνήμης `0x100` (δηλαδή το δεκαδικό αριθμό 256) του πίνακα μνήμης. Έπειτα, αποθηκεύει την εντολή που διαβάστηκε από τη θέση μνήμης στη μεταβλητή `opcode` (κωδικός εντολής) η οποία θα χρησιμοποιηθεί από τη μέθοδο `decode()`, και αυξάνει τον μετρητή προγράμματος κατά 1.

Η μέθοδος `decode()` καλείται μετά την ολοκλήρωση της μεθόδου `fetch()`. Λαμβάνει ως είσοδο τον πίνακα της μνήμης του Game Boy, παρομοίως με την εντολή `fetch()`, έτσι ώστε να πραγματοποιήσει τις απαιτούμενες αλλαγές στη μνήμη, εφόσον κάτι τέτοιο είναι απαραίτητο. Η μέθοδος `decode()` διαβάζει τη μεταβλητή κωδικού εντολής η οποία αποθηκεύτηκε από τη μέθοδο `fetch`, και ανάλογα με την τιμή της ανατρέχει στο σετ εντολών του Game Boy και πραγματοποιεί τις απαραίτητες ενέργειες. Ουσιαστικά η μέθοδος `decode()` είναι ένας διακόπτης (`switch`), ο οποίος έχει ως συνθήκη τη μεταβλητή `opcode`, και πιθανά αποτελέσματα ολόκληρο το σετ εντολών του Game Boy. Η δημιουργία της μεθόδου `decode()` αποτέλεσε ένα χρονοβόρο κομμάτι της ανάπτυξης του εξομοιωτή, καθώς έπρεπε να δημιουργηθούν σε αυτή και οι 245 εντολές του σετ εντολών του Game Boy.

Εκτός από τις μεθόδους `fetch()` και `decode()`, η κλάση `Cpu` περιέχει και την μέθοδο `cbInstructions()` (εντολές αλλαγής bits), η οποία αποτελεί επέκταση της μεθόδου `decode()`, και καλείται όταν ο κωδικός εντολής είναι ο `0xCB`. Σκοπός της συγκεκριμένης μεθόδου είναι να παρέχει εντολές εκτός του σετ των 245 εντολών, οι οποίες πραγματοποιούν ενέργειες στους τελεσταίους (`operands`) βασισμένες στα bits, όπως κύλιση ή αντικατάσταση bits. Το σετ εντολών `CB` αποτελείται από 256 εντολές bits, οι οποίες εκτελούν τις ενέργειες που παρουσιάστηκαν στο κεφάλαιο 4.2.3.4 (εντολές κύλισης και αντικατάστασης Bit).

5.3.2 Σύνοψη κώδικα υποσυστήματος μονάδας επεξεργασίας εικόνας

Η κλάση `Rpu` περιέχει τον κώδικα ο οποίος υλοποιεί τις λειτουργίες της μονάδας επεξεργασίας εικόνας. Αυτές οι λειτουργίες περιέχουν την εκτέλεση του κύκλου λειτουργίας της μονάδας επεξεργασίας εικόνας, σύμφωνα με το κεφάλαιο 4.3.3, αλλά και την εγγραφή των παραχθέντων εικονοστοιχείων σε μία μεταβλητή `bitmap` (δυσιοαπεικόνιση), η οποία δίνεται στη μέθοδο `draw()` (σχεδίασε) της κλάσης `Rpu` από την κλάση `MainActivity` (από την οποία πραγματοποιείται η εκκίνηση της εφαρμογής), για σκοπούς εμφάνισης της τελικής εικόνας στον χρήστη.



Σχήμα 16 - Διάγραμμα UML της κλάσης Ppu

5.3.2.1 Επεξήγηση μεταβλητών της κλάσης Ppu

Η κλάση Ppu, η οποία ενσωματώνει όλη τη λογική λειτουργίας της μονάδας επεξεργασίας εικόνας, περιέχει όλες τις μεταβλητές οι οποίες σχετίζονται με τους καταχωρητές της μονάδας επεξεργασίας εικόνας. Το παρόν υποκεφάλαιο παρουσιάζει αυτές τις μεταβλητές και περιγράφει συνοπτικά η λειτουργία της κάθε μίας. Η αναλυτική περιγραφή της κάθε μίας από αυτές τις μεταβλητές πραγματοποιείται στο κεφάλαιο της ανάλυσης των λειτουργιών των καταχωρητών της μονάδας επεξεργασίας εικόνας (κεφάλαιο 4.3.2).

Όλες οι μεταβλητές των καταχωρητών της μονάδας επεξεργασίας εικόνας αποθηκεύονται ως ακέραιοι αριθμοί (σε δεκαεξαδική μορφή). Για παράδειγμα, καθώς ο καταχωρητής ελέγχου LCD (regLCDC) κατά την εκκίνηση του εξομοιωτή έχει ενεργοποιημένες τις λειτουργίες ενεργοποίηση οθόνης LCD, επιλογή περιοχής 0x8000-0x8FFF ως χάρτη παρασκηνίου και παραθύρου και ενεργοποίηση γραφικών στοιχείων, σύμφωνα με τον πίνακα 5, θα περιέχει τη δυαδική τιμή 1001 0001, η οποία ισούται με τον δεκαεξαδικό αριθμό 0x91. Άρα, η μεταβλητή του καταχωρητή ελέγχου LCD θα αρχικοποιηθεί με την τιμή 0x91. [41]

Κατά τη λειτουργία του εξομοιωτή, είναι φυσικό οι τιμές των καταχωρητών αυτών να αλλάξουν. Παραδείγματος χάριν, στις περιπτώσεις στις οποίες το παιχνίδι που εκτελείται χρησιμοποιεί την περιοχή μνήμης 0x8800-0x97FF για τον χάρτη παρασκηνίου και παραθύρου αντί της προαναφερθείσας περιοχής 0x8000-0x8FFF, το πέμπτο bit του καταχωρητή ελέγχου LCD θα λάβει την τιμή 0. Για αυτές τις περιπτώσεις αλλαγής συγκεκριμένων bits, έχουν δημιουργηθεί βοηθητικές συναρτήσεις οι οποίες καλούνται όταν κρίνεται απαραίτητο από την μονάδα επεξεργασίας εικόνας. Αυτές οι βοηθητικές συναρτήσεις θα αναλυθούν περαιτέρω στο κεφάλαιο της σύνοψης κώδικα βοηθητικών συναρτήσεων (κεφάλαιο 5.3.3).

5.3.2.2 Επεξήγηση της μεθόδου draw() της κλάσης Ppu

Η μέθοδος draw() αποτελεί τη βασική μέθοδο για τη δημιουργία της τελικής εικόνας του παιχνιδιού, η οποία αποτυπώνεται στην οθόνη της συσκευής του χρήστη. Καλείται σε κάθε κύκλο λειτουργίας της μονάδας επεξεργασίας εικόνας, και είναι υπεύθυνη για την εκτέλεση πολλαπλών λειτουργιών. Το

παρόν υποκεφάλαιο αναλύει αυτές τις λειτουργίες, μαζί με τον τρόπο που εκτελούνται από τη μέθοδο `draw()` της κλάσης `Ppu`, έτσι ώστε αυτές να γίνουν κατανοητές από τον αναγνώστη.

Κύρια λειτουργία της μεθόδου `draw()` είναι η εκτέλεση του κύκλου λειτουργίας της μονάδας επεξεργασίας εικόνας, όπως αυτή αναλύθηκε στο κεφάλαιο 4.3.3. Για την επιτυχή ολοκλήρωση του συγκεκριμένου κύκλου, η μέθοδος `draw()` εκτελεί αρκετά βήματα. Αρχικά, ελέγχει εάν η μονάδα επεξεργασίας εικόνας έχει αρχικοποιηθεί ήδη. Κατά την αρχικοποίησή της, η μονάδα επεξεργασίας εικόνας βρίσκεται σε κατάσταση κατακόρυφου κενού διαστήματος. Εάν δεν έχει αρχικοποιηθεί, τότε παραμένει σε κατάσταση κατακόρυφου κενού λειτουργίας για όσους κύκλους γραφικών χρειάζεται, μέχρι να εισαχθεί σε κατάσταση λειτουργίας αναζήτησης OAM.

Μετά την αρχικοποίηση της μονάδας επεξεργασίας εικόνας, η μέθοδος `draw()` εισέρχεται κανονικά στον κύκλο λειτουργίας της. Κάθε φορά που καλείται, αρχικά ελέγχει για τη συνθήκη διακοπής $LYC=LY$, και έπειτα συνεχίζει με την εκτέλεση της κατάστασης λειτουργίας στην οποία βρίσκεται κάθε φορά.

Η επιλογή της κατάστασης λειτουργίας την οποία θα εκτελέσει η μέθοδος `draw()` γίνεται με τη χρήση ενός διακόπτη. Η μέθοδος `draw()`, με χρήση της μεταβλητής-σημιαίας (flag) `modeflag`, εισέρχεται κάθε φορά στην κατάλληλη κατάσταση λειτουργίας, στην οποία παραμένει μέχρι να ολοκληρωθούν οι κύκλοι γραφικών όπου πρέπει να καταναλώσει η συγκεκριμένη κατάσταση λειτουργίας, και έπειτα θέτει τη μεταβλητή `modeflag` στην επόμενη κατάσταση λειτουργίας.

Κατά την κατάσταση λειτουργίας 0 (κατάσταση οριζοντίου διαστήματος κενού), η μέθοδος `draw()` αρχικοποιεί τις μεταβλητές της προηγούμενης κατάστασης λειτουργίας (3), και έπειτα ελέγχει για τυχόν πιθανή διακοπή που προκαλείται από τον καταχωρητή κατάστασης. Έπειτα, ανάλογα με τη γραμμή την οποία εμφανίζει η μονάδα επεξεργασίας εικόνας, θέτει τη μεταβλητή `modeflag` στην κατάσταση λειτουργίας κατακόρυφου διαστήματος κενού (εάν η γραμμή είναι η 144^η) ή στην κατάσταση λειτουργίας αναζήτησης OAM (για όλες τις υπόλοιπες γραμμές). Η κατάσταση λειτουργίας οριζοντίου διαστήματος κενού εκτελείται για 87 έως 204 κύκλους γραφικών, ανάλογα με τη διάρκεια της κατάστασης λειτουργίας εμφάνισης εικονοστοιχείων.

Κατά την κατάσταση λειτουργίας 1 (κατάσταση κατακόρυφου διαστήματος κενού), η μέθοδος `draw()` θέτει τη σημαία διακοπής κατακόρυφου διαστήματος κενού, και κάθε 456 κύκλους γραφικών αυξάνει τη γραμμή. Η συγκεκριμένη κατάσταση λειτουργίας εκτελείται για 4560 κύκλους γραφικών, ή 10 γραμμές. Οι γραμμές αυτές δεν εμφανίζονται στην οθόνη, και αποτελούν τις γραμμές 145 έως 154. Κατά το πέρας της κατάστασης λειτουργίας κατακόρυφου διαστήματος κενού, η μεταβλητή `modeflag` θέτεται στην κατάσταση λειτουργίας αναζήτησης OAM.

Κατά την κατάσταση λειτουργίας 2 (αναζήτηση OAM), η μέθοδος `draw()` ελέγχει την περιοχή μνήμης χαρακτηριστικών γραφικών στοιχείων (Object Attribute Memory) για γραφικά στοιχεία τα οποία πρέπει να εμφανιστούν στην τρέχουσα γραμμή, και τα αποθηκεύει στην ουρά των γραφικών

στοιχείων. Η κατάσταση λειτουργίας OAM καταναλώνει 80 κύκλους γραφικών, και στο τέλος της, θέτει τη μεταβλητή `modeflag` στην κατάσταση λειτουργίας εμφάνισης εικονοστοιχείων.

Η τελευταία κατάσταση λειτουργίας την οποία εκτελεί η μέθοδος `draw()`, είναι η κατάσταση λειτουργίας 3 (εμφάνιση εικονοστοιχείων). Η υλοποίηση της συγκεκριμένης κατάστασης λειτουργίας είναι η πιο πολύπλοκη από όλες, καθώς περιέχει έναν διακόπτη (μεταβλητή `mode3step`) ο οποίος καθορίζει σε ποιο σημείο της εμφάνισης εικονοστοιχείων βρίσκεται. Όταν ολοκληρωθεί ένα βήμα, ο μετρητής αυξάνεται, και στον επόμενο κύκλο γραφικών εκτελείται το επόμενο βήμα. Τα βήματα τα οποία υλοποιούνται μέσα από την συγκεκριμένο διακόπτη παρουσιάστηκαν στο υποκεφάλαιο ανάλυσης κατάστασης λειτουργίας εμφάνισης εικονοστοιχείων (υποκεφάλαιο 4.3.3.2). Συνοπτικά, ελέγχεται ο χάρτης πλακιδίων ο οποίος χρησιμοποιείται, και λαμβάνονται τα δεδομένα των πλακιδίων από τη Video RAM. Έπειτα, προωθούνται τα εικονοστοιχεία τα οποία πρόκειται να εμφανιστούν στην ουρά εικονοστοιχείων, και όταν αυτή η ουρά περιέχει πάνω από 8 εικονοστοιχεία, αυτά γράφονται στη μεταβλητή δυφιοαπεικόνισης (`bitmap`) η οποία χρησιμοποιείται για την εμφάνιση της εικόνας στην οθόνη της εφαρμογής. Η κατάσταση λειτουργίας εμφάνισης εικονοστοιχείων διαρκεί από 179 έως 289 κύκλους γραφικών, ανάλογα με το πλήθος των εικονοστοιχείων τα οποία πρέπει να εμφανιστούν. Όσο μεγαλύτερη είναι η διάρκεια της συγκεκριμένης κατάστασης, τόσο μικρότερη είναι η διάρκεια της κατάστασης λειτουργίας οριζόντιου διαστήματος κενού. Το άθροισμα κύκλων γραφικών των δύο αυτών καταστάσεων είναι 383 κύκλοι γραφικών.

Σε αυτό το υποκεφάλαιο, παρουσιάστηκε η μέθοδος `draw()` της κλάσης `Ppu`, η οποία είναι υπεύθυνη για τη δημιουργία της τελικής εικόνας του παιχνιδιού. Η μέθοδος αυτή καλείται σε κάθε κύκλο λειτουργίας της μονάδας επεξεργασίας εικόνας, και εκτελεί μια σειρά από λειτουργίες που αφορούν τη διαχείριση των καταστάσεων της μονάδας επεξεργασίας εικόνας και την απόδοση των εικονοστοιχείων τα οποία αποτελούν την εικόνα. Μέσω αυτής της μεθόδου, επιτυγχάνεται η πιστή εξομοίωση των εικόνων των παιχνιδιών Game Boy στην οθόνη της εφαρμογής.

5.3.3 Σύνοψη κώδικα βοηθητικών συναρτήσεων

Σκοπός του παρόντος κεφαλαίου είναι η πραγματοποίηση μιας συνοπτικής περιγραφής των βοηθητικών συναρτήσεων οι οποίες χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής του εξομοιωτή. Η χρήση τέτοιων συναρτήσεων απαιτείται για λόγους καλύτερης δομής του κώδικα, καθώς σε πολλά σημεία των προαναφερθέντων κλάσεων, απαιτούνταν η επαναλαμβανόμενη χρήση παρόμοιων κομματιών κώδικα (`boilerplate code`). Οι βοηθητικές συναρτήσεις οι οποίες χρησιμοποιήθηκαν, είναι συγκεντρωμένες σε έναν φάκελο με όνομα `helper`, ο οποίος περιλαμβάνει τρία αρχεία.

Το αρχείο `CpuUtils.kt` περιλαμβάνει συναρτήσεις οι οποίες χρησιμοποιούνται για την υλοποίηση του κώδικα του υποσυστήματος του επεξεργαστή. Ο κώδικας του επεξεργαστή υλοποιεί όλο το σετ εντολών του Game Boy. Επομένως, πολλές εντολές εκτελούν τον ίδιο κώδικα για

διαφορετικούς καταχωρητές, θέσεις μνήμης, κλπ. Στο αρχείο `CpuUtils.kt` περιέχονται συναρτήσεις χωρισμένες σε διάφορες κατηγορίες, ανάλογα με την εντολή η οποία εκτελείται. Μερικές από αυτές τις κατηγορίες είναι οι εξής: συναρτήσεις για τη διαχείριση των σημαίων (flags), όπως οι `setFlag()` και `getFlag()`, συναρτήσεις για την αλλαγή δυαδικών αριθμών σε συγκεκριμένες θέσεις ενός αριθμού, όπως η `setBit()`, και συναρτήσεις για πράξεις μεταξύ 2 δεκαεξαδικών αριθμών οι οποίες λαμβάνουν υπόψη την υπερχείλιση (overflow) η οποία συμβαίνει για αριθμούς μεγαλύτερους από `0xFF`.

Το αρχείο `PpuUtils.kt` περιλαμβάνει συναρτήσεις οι οποίες χρησιμοποιούνται για την υλοποίηση του κώδικα του υποσυστήματος της μονάδας επεξεργασίας εικόνας. Οι συναρτήσεις αυτές υλοποιούν βασικές λειτουργίες του υποσυστήματος, με σκοπό την καλύτερη δομή της ήδη πολύπλοκης συνάρτησης `draw()`, η οποία τις καλεί. Παραδείγματα τέτοιων συναρτήσεων είναι οι: `initRegisters()`, συνάρτηση η οποία αρχικοποιεί τις μεταβλητές των καταχωρητών της μονάδας επεξεργασίας εικόνας, διαβάζοντας τις τιμές τους από τις ανάλογες θέσεις μνήμης, `getTileDataLow()` και `getTileDataHigh()`, συναρτήσεις οι οποίες διαβάζουν τα δεδομένα του ζητούμενου πλακιδίου από τη Video Ram, και η συνάρτηση `PixelSelector()`, η οποία εισάγει την τιμή του εικονοστοιχείου, το οποίο πρόκειται να εμφανιστεί, στη μεταβλητή `bitmap`.

Τέλος, το αρχείο `DebugUtils.kt` περιλαμβάνει συναρτήσεις οι οποίες ήταν αρκετά βοηθητικές στην αποσφαλμάτωση της εφαρμογής, κυρίως κατά τη διάρκεια των πρώιμων σταδίων ανάπτυξής της. Οι συναρτήσεις που περιέχονται στο συγκεκριμένο αρχείο δεν απαιτούνται για την επιτυχημένη λειτουργία του εξομοιωτή. Παραδείγματα περιπτώσεων χρήσης τέτοιων εφαρμογών είναι η εκτύπωση τιμών καταχωρητών ή θέσεων μνήμης σε αρχείο κειμένου σε συγκεκριμένα στάδια εκτέλεσης της εφαρμογής για τη σύγκρισή τους με τις σωστές τιμές, η αποθήκευση του πίνακα της μνήμης του GameBoy σε αρχείο για τη δημιουργία καταστάσεων αποθήκευσης (save states), ή η ανάγνωση των στοιχείων της κασέτας η οποία φορτώνεται στην εφαρμογή, όπως τίτλος κασέτας, τύπος τράπεζας μνήμης ή χώρα προέλευσης της συγκεκριμένης κασέτας (γιαπωνέζικα ή αγγλικά).

5.3.4 Σύνοψη κώδικα υλοποίησης εφαρμογής Android

Για την επιτυχή εκτέλεση της εφαρμογής Android σε έξυπνες συσκευές, απαιτείται η δημιουργία του κώδικα ο οποίος είναι υπεύθυνος για την αρχικοποίηση των τμημάτων της διεπαφής χρήστη της εφαρμογής, του κώδικα που είναι υπεύθυνος για την είσοδο χρήστη, αλλά και για τη δημιουργία των αντικειμένων των κλάσεων που αναλύθηκαν στα προηγούμενα υποκεφάλαια. Στο παρόν υποκεφάλαιο θα αναλυθούν τα περιεχόμενα των αρχείων Kotlin τα οποία δημιουργήθηκαν για τους συγκεκριμένους σκοπούς. Με την ανάγνωση του παρόντος υποκεφαλαίου, ο αναγνώστης θα λάβει τις κατάλληλες γνώσεις σχετικά με τη δομή του κώδικα ο οποίος καλείται για του σκοπούς εκκίνησης μίας εφαρμογής Android, αλλά και τη σύνδεση του συγκεκριμένου κώδικα με τον κώδικα ο οποίος πραγματοποιεί την υλοποίηση της λογικής του εξομοιωτή που αναλύθηκε στα προηγούμενα υποκεφάλαια.

Η σημαντικότερη κλάση του κώδικα της υλοποίησης της εφαρμογής Android, είναι η κλάση `MainActivity`, η οποία περιέχεται στο αρχείο `MainActivity.kt` που βρίσκεται στον φάκελο `ui` του κώδικα της εφαρμογής. Η κλάση `MainActivity` κατέχει ιδιαίτερη σημασία, καθώς ο κώδικας ο οποίος περιέχει εκτελείται κατά την εκκίνηση της εφαρμογής, και δημιουργεί την κύρια δραστηριότητα της.

Με την εκτέλεση του κώδικα της συγκεκριμένης κλάσης, δημιουργείται το αντικείμενο `GameBoy`, καθώς και η μεταβλητή `bitmap` η οποία περιέχει τον πίνακα εικονοστοιχείων που εμφανίζονται στο τμήμα της οθόνης παιχνιδιού της εφαρμογής. Έπειτα, δημιουργείται ο πίνακας 5 θέσεων `fragmentsarray`, που περιέχει τα τμήματα τα οποία αποτελούν τη διεπαφή χρήστη, τα οποία εναλλάσσονται μεταξύ τους. Τέλος, καλείται η μέθοδος `onCreate()`, η οποία πραγματοποιεί την εκτέλεση της εφαρμογής.

5.3.4.1 Ανάλυση σκοπού και περιεχομένων μεθόδου `onCreate()` της κλάσης `MainActivity`

Η μέθοδος `onCreate()` της κλάσης `MainActivity` είναι η πρώτη μέθοδος που εκτελείται κατά την εκκίνηση μίας εφαρμογής Android. Έχει σημαντικό ρόλο στην εφαρμογή, καθώς προσθέτει τα τμήματα τα οποία θα εμφανιστούν με την εκκίνηση της εφαρμογής στα κατάλληλα δοχεία (υποκεφάλαιο 5.2.2), και καλεί τις αρχικές μεθόδους της λογικής του εξομοιωτή.

Στην περίπτωση της εφαρμογής του εξομοιωτή, με την εκκίνηση της εκτέλεσης της μεθόδου `onCreate()`, πραγματοποιούνται συγκεκριμένα βήματα αρχικοποίησης της κατάστασης του εξομοιωτή, προτού εκτελεστεί ο κώδικας της λογικής του. Το πρώτο βήμα αποτελεί την εμφάνιση των τμημάτων της μπάρας τίτλου και της οθόνης παιχνιδιού. Στη συνέχεια, καλείται η μέθοδος `initMemory()` της κλάσης `GameBoy` η οποία θέτει τις αρχικές τιμές των καταχωρητών του `GameBoy` στη μνήμη. Μετά την αρχικοποίηση, καλείται ο κώδικας που φορτώνει τα δεδομένα της κασέτας στη μνήμη του `Game Boy`, μέσω της μεθόδου `loadCartridge()` του αντικειμένου `GameBoy`.

Έπειτα από τη διαδικασία αρχικοποίησης του εξομοιωτή, πραγματοποιείται η εκτέλεση μίας υπορουτίνας η οποία εκτελείται παράλληλα με το κύριο νήμα της εφαρμογής. Η συγκεκριμένη υπορουτίνα περιέχει την εκτέλεση της μεθόδου `executionCycle()` του αντικειμένου `GameBoy`, η οποία είναι υπεύθυνη για την εκτέλεση του βρόχου εξομοίωσης του `Game Boy`. Ο λόγος χρήσης υπορουτίνας για την εκτέλεση της συγκεκριμένης μεθόδου, είναι η αποφυγή της παρεμπόδισης του νήματος το οποίο είναι υπεύθυνο για την εμφάνιση της διεπαφής χρήστη. Εάν το κύριο νήμα ασχολείται μόνιμα με την εκτέλεση του βρόχου, τότε δεν θα υπάρχει διαθέσιμο νήμα για να χειρίζεται δράσεις όπως οι αλλαγές των τμημάτων της διεπαφής χρήστη, ή η διαχείριση της εισόδου από την πλευρά του χρήστη. Τέλος, στη μέθοδο `executionCycle()` θα δοθεί ως όρισμα η μεταβλητή `bitmap`, η οποία θα χρησιμοποιηθεί από την μονάδα επεξεργασίας εικόνας για την εμφάνιση των καρτέ του παιχνιδιού.

5.3.4.2 Παρουσίαση υπόλοιπων μεθόδων της κλάσης `MainActivity`

Πέρα από τη μέθοδο `onCreate()` της κλάσης `MainActivity`, έχει πραγματοποιηθεί και η υλοποίηση κάποιων άλλων μεθόδων στη συγκεκριμένη κλάση, οι οποίες είναι απαραίτητες για την λειτουργία του

εξομοιωτή. Στο συγκεκριμένο υποκεφάλαιο θα παρουσιαστούν συνοπτικά αυτές οι μεθόδους και οι λόγοι υλοποίησής τους, για σκοπούς ολοκληρωμένης περιγραφής του κώδικα.

Μια σημαντική μέθοδος για την αλληλοεπίδραση του χρήστη με την εφαρμογή του εξομοιωτή, είναι η μέθοδος `onNavigationItemSelected()`. Η συγκεκριμένη μέθοδος εκτελείται όταν ο χρήστης πραγματοποιεί μία επιλογή στο τμήμα του πλαγίου μενού. Με τη χρήση ενός διακόπτη μέσα στη μέθοδο `onNavigationItemSelected()`, πραγματοποιείται αλλαγή του τμήματος στο δοχείο το οποίο φιλοξενεί το τμήμα της οθόνης παιχνιδιού, με κάποιο άλλο τμήμα, ανάλογα με την επιλογή του χρήστη. Για παράδειγμα, με την επιλογή της εμφάνισης των ρυθμίσεων από το πλαϊνό μενού, το τμήμα της οθόνης παιχνιδιού αντικαθίσταται από το τμήμα της οθόνης των ρυθμίσεων της εφαρμογής. Το τμήμα το οποίο αντικαταστάθηκε, προστίθεται σε μία στοίβα (back stack), από την οποία μπορεί να κληθεί ξανά στην περίπτωση όπου ο χρήστης επιλέξει να μεταβεί στην προηγούμενη οθόνη μέσω του κουμπιού «Back» της συσκευής.

Η μέθοδος `onBackPressed()` είναι υπεύθυνη για την εμφάνιση ενός συγκεκριμένου τμήματος, στην περίπτωση όπου ο χρήστης πατήσει το κουμπί «Back» με σκοπό την επιστροφή στην προηγούμενη οθόνη. Στην περίπτωση όπου το πλάγιο μενού είναι ανοιχτό, με την εκτέλεση της συγκεκριμένης μεθόδου θα κλείσει. Σε περίπτωση όπου το πλάγιο μενού δεν είναι ανοιχτό και κάποιο τμήμα πέρα από το τμήμα της οθόνης παιχνιδιού είναι φορτωμένο στο δοχείο, τότε το τρέχον τμήμα θα αντικατασταθεί από το προηγούμενο τμήμα σύμφωνα με τη στοίβα (back stack). Τέλος, στην περίπτωση όπου το τμήμα της οθόνης παιχνιδιού είναι φορτωμένο στο δοχείο (η αρχική οθόνη της εφαρμογής), τότε θα εμφανιστεί ένα μήνυμα στο χρήστη το οποίο θα τον προειδοποιεί ότι η εφαρμογή θα κλείσει εάν αυτός πατήσει ξανά το κουμπί «Back». Εάν ο χρήστης ξαναπατήσει το κουμπί μέσα στα επόμενα 2 δευτερόλεπτα, η εφαρμογή θα τερματιστεί.

Τέλος, οι μέθοδοι `openDrawer()` και `closeDrawer()` πραγματοποιούν το άνοιγμα και το κλείσιμο του πλαγίου μενού, ενώ οι μέθοδοι `pauseGameboyExecution()` και `resumeGameboyExecution()` θέτουν μία σημαία η οποία πραγματοποιεί παύση και συνέχιση του βρόχου εκτέλεσης του εξομοιωτή, έτσι ώστε αυτός να μην εκτελείται ενώ ο χρήστης βρίσκεται σε κάποιο τμήμα το οποίο δεν είναι το τμήμα της οθόνης παιχνιδιού.

5.4 Δυσκολίες ανάπτυξης εφαρμογής και τρόποι αντιμετώπισής τους

Κατά την υλοποίηση του έργου με το οποίο ασχολείται η παρούσα διπλωματική εργασία, όπως είναι φυσικό, παρουσιάστηκαν αρκετές δυσκολίες. Οι δυσκολίες αυτές εμφανίστηκαν τόσο στο θεωρητικό κομμάτι της τεχνικής ανάλυσης των υποσυστημάτων του Game Boy, όσο και στο τεχνικό κομμάτι της υλοποίησης της εφαρμογής του εξομοιωτή. Στο παρόν κεφάλαιο θα πραγματοποιηθεί παρουσίαση των εμποδίων τα οποία ήταν απαραίτητα να ξεπεραστούν για την επιτυχή ολοκλήρωση του έργου, καθώς και οι τρόποι με τους οποίους πραγματοποιήθηκε η προσέγγιση της αντιμετώπισής τους.

Μία αξιοσημείωτη δυσκολία η οποία παρουσιαζόταν σε όλη τη διάρκεια του έργου, ήταν αυτή της εύρεσης των απαραίτητων πληροφοριών σχετικά με την αναλυτική λειτουργία των υποσυστημάτων του Game Boy. Οι πληροφορίες αυτές αποτελούσαν πληροφορίες ύψιστης σημασίας για την περάτωση του έργου, καθώς στα έργα ανάπτυξης εξομοιωτών είναι σημαντικό να υλοποιηθούν σωστά όλες οι λεπτομέρειες των υποσυστημάτων για την επίτευξη της εξομοίωσης. Αξίζει να σημειωθεί πως δεν υπάρχουν επίσημα έγγραφα από την κατασκευάστρια εταιρεία για τη λειτουργία της συσκευής. Ο συγκεκριμένος παράγοντας έπαιξε καθοριστικό ρόλο στην αυξημένη διάρκεια της έρευνας, καθώς για την αντιμετώπιση της συγκεκριμένης δυσκολίας οι πληροφορίες αντλήθηκαν από μεγάλο εύρος διαφορετικών πηγών του διαδικτύου, σε μορφή ιστοσελίδων ή βίντεο από συνέδρια, οι οποίες είχαν δημιουργηθεί μέσω πολυετούς μελέτης του αυθεντικού συστήματος.

Επίσης, με την ολοκλήρωση της ανάπτυξης των υποσυστημάτων του επεξεργαστή και των γραφικών, παρουσιάστηκε η ανάγκη για τον έλεγχο της σωστής λειτουργίας του εξομοιωτή. Για τον επιτυχή έλεγχο της εφαρμογής, επιλέχθηκε η διαδικασία της σύγκρισης της κατάστασης της μνήμης του συστήματος με την κατάσταση άλλων εξομοιωτών οι οποίοι είναι γνωστό ότι δουλεύουν με ακρίβεια. Για αυτό τον λόγο, δημιουργήθηκε η απαίτηση της ανάπτυξης εργαλείων αποσφαλμάτωσης της εφαρμογής, τα οποία αποδείχθηκαν ιδιαίτερα χρήσιμα για την παροχή πληροφοριών σχετικά με τα δεδομένα που περιέχονται σε συγκεκριμένες θέσεις μνήμης κατά τη λειτουργία του εξομοιωτή. Επομένως, δημιουργήθηκαν δύο λειτουργίες για σκοπούς αποσφαλμάτωσης, οι οποίες μπορούν να εκτελεστούν από το πλάγιο μενού του εξομοιωτή, η λειτουργία Debug Registers και η λειτουργία Debug Graphics.

Κατά το στάδιο της αποσφαλμάτωσης του εξομοιωτή, παρατηρήθηκε πως η απόδοσή του δεν ήταν ικανοποιητική. Χαρακτηριστικό παράδειγμα το οποίο παρουσιάστηκε, ήταν η εκτέλεση του προγράμματος εκκίνησης του Game Boy. Κατά την εκτέλεση του προγράμματος εκκίνησης, το Game Boy πραγματοποιεί κάποιους ελέγχους για την εγκυρότητα της κασέτας η οποία φορτώνεται, θέτει τις κατάλληλες τιμές στους καταχωρητές, και παράλληλα πραγματοποιεί κύλιση του λογότυπου της Nintendo στην οθόνη. Η εκτέλεση του συγκεκριμένου προγράμματος στο αυθεντικό υλικό πραγματοποιείται περίπου σε 5 δευτερόλεπτα. Αντιθέτως, στον εξομοιωτή αρχικά πραγματοποιούνταν σε πολύ μεγαλύτερο χρονικό διάστημα, στο εύρος από 20 έως 30 δευτερόλεπτα. Σε εκείνο το σημείο ήταν προφανές πως έπρεπε να πραγματοποιηθεί αύξηση της απόδοσης του εξομοιωτή. Για να αυξηθεί η απόδοση, έπρεπε να εντοπιστούν οι εντολές οι οποίες καθυστερούσαν την εκτέλεση του προγράμματος εκκίνησης. Ο εντοπισμός πραγματοποιήθηκε μέσω μέτρησης του χρόνου (benchmarking) των εντολών όπου εκτελούνταν, έτσι ώστε να βρεθούν οι πιο χρονοβόρες. Πράγματι, οι πιο χρονοβόρες εντολές αποδείχθηκαν οι εντολές οι οποίες πραγματοποιούσαν μετατροπή των bits σε συμβολοσειρές (strings), με σκοπό την αντικατάσταση bits σε συγκεκριμένες θέσεις. Η συγκεκριμένη μετατροπή προκαλούσε αρκετά μεγάλο χτύπημα στην απόδοση, και διορθώθηκε με

χρήση πράξεων της άλγεβρας Boole. Έπειτα από την αλλαγή όλων των εντολών οι οποίες αρχικά χρησιμοποιούσαν μετατροπή σε συμβολοσειρά, η απόδοση αυξήθηκε σημαντικά, οδηγώντας στην εκτέλεση του προγράμματος εκκίνησης σε χρόνο μικρότερο των 5 δευτερολέπτων.

Τέλος, παραμένοντας στο πλαίσιο της αποσφαλμάτωσης της εφαρμογής, ήταν εξίσου σημαντικό να γίνει η επιλογή των κατάλληλων εξωτερικών εργαλείων, έτσι ώστε να πραγματοποιηθεί ο σωστός έλεγχος της ακρίβειας του εξομοιωτή. Μετά από έρευνα επιλέχθηκε ο ήδη υπάρχων εξομοιωτής BGB, ο οποίος παρέχει την εξαιρετικά χρήσιμη λειτουργία αποσφαλμάτωσης, μέσω της οποίας έγινε η σύγκριση των τιμών μνήμης με τα προαναφερθέντα εργαλεία τα οποία δημιουργήθηκαν στα πλαίσια της αποσφαλμάτωσης της εφαρμογής.

6 Αξιολόγηση εφαρμογής

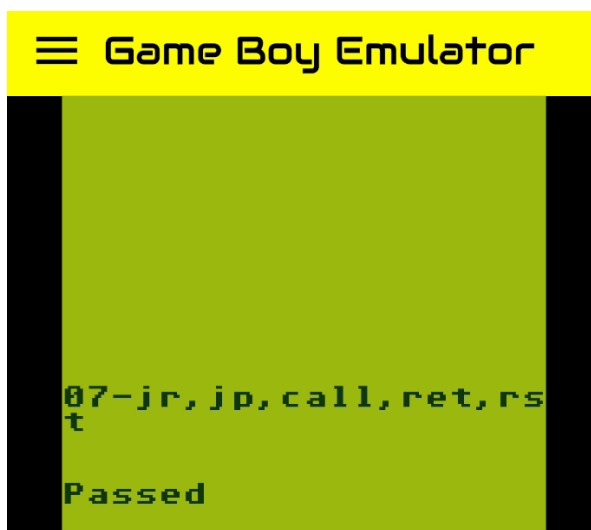
Η αξιολόγηση του έργου της παρούσας διπλωματικής εργασίας είναι ένα κεφάλαιο το οποίο κατέχει ιδιαίτερη σημασία. Μέσω του παρόντος κεφαλαίου, εξασφαλίζεται η επιτυχία της ανάπτυξης του εξομοιωτή, παρουσιάζονται τα σημεία του έργου τα οποία χρήζουν βελτίωσης, και τελικώς, διαμορφώνεται η τελική εικόνα της απόδοσης του εξομοιωτή.

Η αξιολόγηση πραγματοποιήθηκε με χρήση δοκιμαστικών προγραμμάτων (Test ROMs), τα οποία έχουν αναπτυχθεί από την κοινότητα, και αποσκοπούν στη δοκιμή της απόδοσης όλων των υποσυστημάτων του εξομοιωτή. Μέσω της διαδικασίας αξιολόγησης, παρουσιάζεται επίσης και οποιαδήποτε ανάγκη υπάρχει για προσθήκη μελλοντικών επεκτάσεων στον εξομοιωτή, οι οποίες θα αναλυθούν στο επόμενο κεφάλαιο.

6.1 Αξιολόγηση απόδοσης εξομοιωτή με χρήση δοκιμαστικών προγραμμάτων Blargg (Blargg's Test ROMs)

Για την αξιολόγηση της απόδοσης του εξομοιωτή, έγινε χρήση της ομάδα δοκιμαστικών προγραμμάτων που αναπτύχθηκε από τον Blargg, και είναι ιδιαίτερα γνωστή στην κοινότητα ανάπτυξης προγραμμάτων εξομοιωτών. Εξετάζει όλα τα βασικά μέρη του εξομοιωτή, καθώς και ακραίες περιπτώσεις, και παρουσιάζει τα αποτελέσματα στην οθόνη, δίνοντας τον αριθμό της κάθε δοκιμής και τον κωδικό κατάστασής της.

Ο κωδικός κατάστασης περιέχει την τιμή «Passed» σε περίπτωση όπου η δοκιμή εκτελέστηκε με επιτυχία, διαφορετικά περιέχει έναν κωδικό αποτυχίας, μέσω του οποίου ο προγραμματιστής του εξομοιωτή μπορεί να εντοπίσει το πρόβλημα που παρουσιάστηκε στη συγκεκριμένη δοκιμή.



Σχήμα 17- Παράδειγμα επιτυχημένης δοκιμής



Σχήμα 18 - Παράδειγμα αποτυχημένης δοκιμής

Τα δοκιμαστικά προγράμματα του επεξεργαστή αποτελούν τις σημαντικότερες δοκιμές οι οποίες πρέπει να εκτελεστούν στον εξομοιωτή. Η επιτυχής εκτέλεσή τους εξασφαλίζει πως ο πυρήνας του εξομοιωτή εκτελείται δίχως σφάλματα, και παρέχει μία ισχυρή βάση για τη δημιουργία των υπόλοιπων υποσυστημάτων.

Όνομα Δοκιμαστικού Προγράμματος	Τύπος Ελέγχου	Επιτυχές;
01-special	Διάφορες εντολές	Ναι
02-interrupts	Διακοπές	Όχι
03-op sp,hl	Πράξεις μεταξύ καταχωρητών SP και HL	Ναι
04-op r, imm	Πράξεις καταχωρητών και δεδομένων μνήμης	Ναι
05-op rp	Πράξεις καταχωρητών BC, DE, HL	Ναι
06-ld r,r	Εντολή LD	Ναι
07-jr, jp, call, ret, rst	Εντολές αλλαγής ροής προγράμματος	Ναι
08-misc instrs	Διάφορες εντολές όπως PUSH, POP	Ναι
09-op r,r	Πράξεις μεταξύ καταχωρητών	Ναι
10-bit ops	Πράξεις δυαδικών	Ναι
11-op a, (hl)	Πράξεις μεταξύ καταχωρητή A και μνήμης	Ναι

Πίνακας 12 - Αξιολόγηση λειτουργίας υποσυστήματος επεξεργαστή

Από τα αποτελέσματα των δοκιμαστικών προγραμμάτων του επεξεργαστή, προκύπτει πως έχουν ενσωματωθεί σωστά όλες οι λειτουργίες του, πλην των διακοπών οι οποίες χρήζουν περαιτέρω αποσφαλμάτωσης. Πιο συγκεκριμένα, το δοκιμαστικό πρόγραμμα μας ενημερώνει πως ο τύπος διακοπής που δε δουλεύει σωστά είναι η διακοπή χρονομέτρου, της οποίας η αποσφαλμάτωση είναι απαραίτητη. Πέραν της διακοπής χρονομέτρου, από την αξιολόγηση με χρήση δοκιμαστικών προγραμμάτων, προκύπτει πως το υπόλοιπο σύστημα του επεξεργαστή δουλεύει δίχως σφάλματα.

7 Συμπεράσματα και μελλοντικές επεκτάσεις

7.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκε η διερεύνηση της αρχιτεκτονικής του υπολογιστικού συστήματος Game Boy, της δημοφιλούς φορητής παιχνιδομηχανής η οποία κυκλοφόρησε από τη Nintendo το 1989. Η συγκεκριμένη διερεύνηση ολοκληρώθηκε με δύο τρόπους: αρχικά, πραγματοποιήθηκε μία τεχνική ανάλυση των υποσυστημάτων τα οποία αποτελούν το υπολογιστικό σύστημα, με σκοπό την καθιέρωση του θεωρητικού υπόβαθρου πάνω στο οποίο χτίστηκε το δεύτερο μέρος, στο οποίο υλοποιήθηκε ένας εξομοιωτής (emulator) του υπολογιστικού συστήματος για έξυπνες συσκευές Android, γραμμένος σε γλώσσα προγραμματισμού Kotlin.

Κατά την τεχνική ανάλυση της συσκευής παρουσιάστηκε λεπτομερώς η δομή της μνήμης της, αλλά και τα υποσυστήματα του επεξεργαστή, της μονάδας επεξεργασίας εικόνας, και της εισόδου χρήστη. Μέσω της ανάλυσης των προαναφερθέντων υποσυστημάτων του Game Boy, αποκτήθηκε η απαιτούμενη γνώση για την επιτυχή εξομοίωσή τους, η οποία πραγματοποιήθηκε στα πλαίσια της ανάπτυξης της εφαρμογής Android. Πέραν της εξομοίωσης, η εφαρμογή αποσκοπεί μεταξύ άλλων και στην ευκολία χρήσης της, την παροχή περαιτέρω λειτουργιών οι οποίες διευκολύνουν τον χρήστη, αλλά και στη δημιουργία μίας καλαίσθητης διεπαφής χρήστη. Επιπρόσθετα, στην παρούσα διπλωματική εργασία παρουσιάζονται τα βήματα της δημιουργίας της εφαρμογής, παρέχοντας στους αναγνώστες καθοδήγηση για την ανάληψη παρόμοιων εγχειρημάτων.

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, εκπληρώθηκαν συγκεκριμένοι στόχοι. Αρχικά, ολοκληρώθηκε η δημιουργία μίας εφαρμογής η οποία συνεισφέρει στον τομέα της διασκέδασης, παρέχοντας στους χρήστες της έναν άμεσο τρόπο να αναπαράγουν παιχνίδια Game Boy μέσω της συσκευής Android τους. Περαιτέρω, πραγματοποιήθηκε η συγκέντρωση πληροφοριών σχετικά με τα τεχνικά χαρακτηριστικά της συσκευής, την αναλυτική λειτουργία των υποσυστημάτων της, και την υλοποίησή τους στα πλαίσια ανάπτυξης μίας εφαρμογής, παρέχοντας με αυτόν τον τρόπο στους αναγνώστες ένα σημείο αναφοράς για μελλοντικές έρευνες. Τέλος, μέσω της επίτευξης αυτών των στόχων, η παρούσα διπλωματική εργασία επιχειρεί να συνδράμει στη διατήρηση και την εκτίμηση μίας ιστορικής συσκευής, η οποία συνέβαλλε στην ριζική αλλαγή του τρόπου με τον οποίο οι άνθρωποι περνούν τον ελεύθερο χρόνο τους τις τελευταίες δεκαετίες.

7.2 Μελλοντικές επεκτάσεις

Η ανάπτυξη ενός εξομοιωτή αποτελεί ένα πολύπλοκο και συνεχές έργο, καθώς η πιστή εξομοίωση ενός συστήματος απαιτεί μεγάλο αριθμό ανθρωποωρών, αφιερωμένο στην ανάπτυξη της εφαρμογής αλλά και στον έλεγχο της εφαρμογής. Τα πλαίσια της παρούσας εργασίας περιορίζουν την περάτωση του έργου, τόσο στα πλαίσια της πιστής εξομοίωσης, όσο και στην προσθήκη χρήσιμων χαρακτηριστικών αλλά και συμβατότητας με το μέγιστο δυνατό αριθμό συσκευών. Το παρόν υποκεφάλαιο είναι

αφιερωμένο στην ανάλυση των μελλοντικών επεκτάσεων της εφαρμογής, η προσθήκη των οποίων θα την καταστήσει ολοκληρωμένη.

7.2.1 Προσθήκη ήχου στον εξομοιωτή

Η σημαντικότερη μελλοντική επέκταση της εφαρμογής του εξομοιωτή είναι η προσθήκη του ολοκληρωμένου υποσυστήματος του ήχου. Δίνεται ιδιαίτερη έμφαση στην συγκεκριμένη επέκταση, καθώς η ύπαρξη ήχου κατά τη χρήση της εφαρμογής κατέχει εξέχουσα σημασία για την αφοσίωση του χρήστη της εφαρμογής σε αυτήν, διότι αναβαθμίζει σημαντικά την εμπειρία παιχνιδιού.

Η ανάπτυξη του υποσυστήματος ήχου αποτελεί ένα εξαιρετικά χρονοβόρο έργο, καθώς στα πλαίσια της ανάπτυξής του, πρέπει να συγχρονιστεί με τα ήδη υπάρχοντα υποσυστήματα του επεξεργαστή και της μονάδας επεξεργασίας εικόνας. Η διαδικασία συγχρονισμού αυξάνει την πολυπλοκότητα του έργου, επειδή είναι πολύ πιθανό να κριθεί απαραίτητη η αναδιοργάνωση ή και η ολική αλλαγή μεγάλων κομματιών του κώδικα των υπολοίπων υποσυστημάτων.

Επιπρόσθετα, ως ιδιαίτερα χρονοβόρα παρουσιάζεται η διαδικασία ελέγχου πιστής εξομοίωσης του ήχου, καθώς πέρα από την σύγκριση των τιμών των καταχωρητών που σχετίζονται με τον ήχο, είναι απαραίτητος και ο έλεγχος της σωστής αναπαραγωγής ήχου σε μεγάλο εύρος παιχνιδιών, πολλές φορές σε σημεία τα οποία βρίσκονται μακριά από την αρχή του παιχνιδιού. Παρά την χρήση αποθηκευμένων καταστάσεων παιχνιδιού (save states), είναι αναμενόμενο λοιπόν να απαιτηθούν ώρες εκτέλεσης συγκεκριμένων παιχνιδιών με σκοπό την αποσφαλμάτωση του υποσυστήματος και την επίτευξη της πιστής εξομοίωσης από την εφαρμογή.

7.2.2 Προσθήκη λειτουργίας Game Boy Color (CGB)

Μία ακόμη σημαντική επέκταση στην εφαρμογή του εξομοιωτή, είναι η υλοποίηση των εντολών και υποσυστημάτων τα οποία θα επιτρέψουν στον εξομοιωτή να εκτελεί παιχνίδια τα οποία υποστηρίζουν την λειτουργία Game Boy Color. Τα συγκεκριμένα παιχνίδια εκτελούνται με χρήση αυξημένου αριθμού χρωμάτων παλέτας και ήταν διαθέσιμα για την αναβαθμισμένη έκδοση του Game Boy, το Game Boy Color, το οποίο κυκλοφόρησε το 1998, σχεδόν μία δεκαετία αργότερα. Η υλοποίηση της συγκεκριμένης επέκτασης θα επιτρέψει στον εξομοιωτή να εκτελέσει 918 επιπλέον παιχνίδια, τα οποία κυκλοφόρησαν την περίοδο 1998 έως 2003 για τη συγκεκριμένη κονσόλα. [42]

Μερικά από τα χαρακτηριστικά τα οποία θα πρέπει να υλοποιηθούν για την προσθήκη της λειτουργίας Game Boy Color στον εξομοιωτή, πέρα από την υποστήριξη της μεγαλύτερης παλέτας χρωμάτων, είναι η δυνατότητα εκτέλεσης παιχνιδιών τα οποία απαιτούν τη διπλάσια ταχύτητα εκτέλεσης από τον επεξεργαστή από αυτή του αρχικού Game Boy (8.38MHz), αλλαγές αρκετών λεπτομερειών στον κώδικα των υπάρχοντων υποσυστημάτων επεξεργαστή και γραφικών, αλλά και δυνατότητα ανίχνευσης δυνατοτήτων παιχνιδιού από την εφαρμογή του εξομοιωτή, δηλαδή εάν το παιχνίδι υποστηρίζει λειτουργία Game Boy Color, αρχικού Game Boy ή και τις 2.

Η προσθήκη αυτών των χαρακτηριστικών θα αποτελέσει χρονοβόρο έργο, όμως θα επιτρέψει στην εφαρμογή να παρέχει την επιλογή της εξομοίωσης 2 υπολογιστικών συστημάτων, αυξάνοντας με αυτόν τον τρόπο τον αριθμό των χρηστών που θα την επιλέξουν.

7.2.3 Αναδόμηση διεπαφής χρήστη για συμβατότητα με μεγαλύτερο αριθμό συσκευών Android

Η εφαρμογή στο παρόν στάδιο ανάπτυξης της εκτελείται επιτυχώς στις σύγχρονες συσκευές Android οι οποίες χρησιμοποιήθηκαν για τη δοκιμή της, καλύπτοντας ικανοποιητικό αριθμό χρηστών, δίχως να παρουσιαστούν σφάλματα στην απεικόνιση της διεπαφής χρήστη. Παρόλα αυτά, κρίνεται σημαντική η δοκιμή της σε ακόμα μεγαλύτερο εύρος συσκευών Android, με σκοπό τη μεγιστοποίηση της συμβατότητάς της.

Για παράδειγμα, σε παλαιότερες συσκευές οι οποίες έχουν χαμηλή ανάλυση για τα σύγχρονα δεδομένα, τα γραφικά στοιχεία τα οποία αποτελούν τη διεπαφή χρήστη είναι πιθανό να καλύπτουν τεράστιο μέρος της οθόνης, καθιστώντας την εφαρμογή δύσχρηστη. Επίσης, υπάρχει και το ενδεχόμενο εκτέλεσης της εφαρμογής σε συσκευές πέρα από έξυπνα κινητά ή τάμπλετ, όπως τηλεοράσεις. Πρέπει λοιπόν να πραγματοποιηθούν οι απαραίτητοι έλεγχοι και η προσαρμογή της διεπαφής χρήστη σε αυτές τις ακραίες περιπτώσεις χρήσης, για λόγους προσέγγισης μεγαλύτερου αριθμού χρηστών.

7.2.4 Επίλυση περιπτώσεων ασύμβατων παιχνιδιών

Υπάρχουν ορισμένες περιπτώσεις παιχνιδιών τα οποία απαιτούν τον υψηλότερο βαθμό ακρίβειας εξομοίωσης του αυθεντικού υλικού του Game Boy. Υπάρχουν επίσης και περιπτώσεις παιχνιδιών τα οποία χρησιμοποιούν σφάλματα τα οποία υπάρχουν στο αυθεντικό Game Boy για την εκτέλεση λειτουργιών.

Γνωστό παράδειγμα τέτοιου παιχνιδιού, αποτελεί το παιχνίδι Pinball Fantasies, στου οποίου την εκτέλεση αποτυγχάνουν ακόμα και οι πιο γνωστοί εξομοιωτές που κυκλοφορούν. [43] Ο λόγος αποτυχίας εκτέλεσης του συγκεκριμένου παιχνιδιού δεν έχει προσδιοριστεί ακόμα, αλλά πιθανολογείται πως η εκτέλεση του βασίζεται σε συνδυασμό χρήσης λανθασμένων περιπτώσεων χρονισμού της μονάδας επεξεργασίας εικόνας οι οποίες υπάρχουν στο αυθεντικό Game Boy, μαζί με διάφορα άλλα σφάλματα που υπάρχουν σε αυτό. Για αυτό το λόγο, οι εξομοιωτές οι οποίοι έχουν διορθώσει κάποιο από αυτά τα σφάλματα που επικρατούν στο αυθεντικό Game Boy, αποτυγχάνουν στην εκτέλεση του συγκεκριμένου παιχνιδιού.

8 Βιβλιογραφία

- [1] L. Crigger, «The Escapist: Searching for Gunpei Yokoi,» 3 March 2007.
- [2] D. Sheff, *Game Over: Press Start to Continue*, GamePress, 1999, p. 294.
- [3] Retro Gamer No. 88, «retrodiary: 1 April – 28 April,» *Retro Gamer No. 88*, αρ. 88, p. 17, 2011.
- [4] AtariAge, «AtariAge - Lynx History,» [Ηλεκτρονικό]. Available: <https://web.archive.org/web/20161201065331/http://www.atariage.com/Lynx/history.html>.
- [5] Nintendo, «Consolidated Sales Transition by Region,» [Ηλεκτρονικό]. Available: https://www.nintendo.co.jp/ir/library/historical_data/pdf/consolidated_sales_e1603.pdf.
- [6] E. C. Nobles, «Game Boy — Nintendo dominates the portable market,» 22 January 2011.
- [7] The Huffington Post, «'Pokken Tournament' and Pokemon's \$1.5 Billion Brand,» 19 March 2017.
- [8] J. Wynne-Jones, «Computer game Tetris celebrates 25 years,» 6 June 2009.
- [9] Guinness World Records, *Guinness World Records 2016 Gamer's Edition*, Jim Pattison Group, 2015, p. 149.
- [10] S. Goss και F. Rodriguez, «Processors of the Game Boy Series,» [Ηλεκτρονικό]. Available: <http://meseec.ce.rit.edu/551-projects/winter2011/1-2.pdf>.
- [11] Nintendo, «Technical Data,» [Ηλεκτρονικό]. Available: <https://www.nintendo.co.uk/Support/Game-Boy-Pocket-Color/Product-information/Technical-data/Technical-data-619585.html>.
- [12] Wikipedia, «Game Boy accessories,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Game_Boy_accessories.
- [13] Wikipedia, «Game Boy Color,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Game_Boy_Color.
- [14] V. Fulber-Garcia, «Differences Between Simulation and Emulation,» [Ηλεκτρονικό]. Available: <https://www.baeldung.com/cs/simulation-vs-emulation>.
- [15] Pan Docs, «Pan Docs,» [Ηλεκτρονικό]. Available: <https://gbdev.io/pandocs/>.
- [16] R. Stähler, «Memory and Memory-Mapped I/O of the Gameboy — Part 3 of a Series,» 2020. [Ηλεκτρονικό]. Available: <https://raphaelstaebler.medium.com/memory-and-memory-mapped-i-o-of-the-gameboy-part-3-of-a-series-37025b40d89b>.
- [17] Data Crystal, «Tetris (Game Boy),» [Ηλεκτρονικό]. Available: [https://datacrystal.romhacking.net/wiki/Tetris_\(Game_Boy\)](https://datacrystal.romhacking.net/wiki/Tetris_(Game_Boy)).

- [18] Game Boy hardware database, «Cartridges by mapper: MBC1,» [Ηλεκτρονικό]. Available: <https://gbhwdb.gekkio.fi/cartridges/mbc1.html>.
- [19] Gameboy Development Wiki, «Memory Bank Controllers,» [Ηλεκτρονικό]. Available: https://gbdev.gg8.se/wiki/articles/Memory_Bank_Controllers.
- [20] B. Byers, «Exploring the Gameboy Memory Bank Controller,» [Ηλεκτρονικό]. Available: <https://b13rg.github.io/Gameboy-MBC-Analysis/#the-mb>.
- [21] M. Steil, «The Ultimate Game Boy Talk,» [Ηλεκτρονικό]. Available: https://media.ccc.de/v/33c3-8029-the_ultimate_game_boy_talk.
- [22] Gameboy Development Wiki, «Video Display,» [Ηλεκτρονικό]. Available: https://gbdev.gg8.se/wiki/articles/Video_Display#VRAM_Background_Maps.
- [23] «Stack Exchange,» [Ηλεκτρονικό]. Available: <https://retrocomputing.stackexchange.com/a/1197>.
- [24] Pan Docs, «OAM DMA Transfer,» [Ηλεκτρονικό]. Available: https://gbdev.io/pandocs/OAM_DMA_Transfer.html#oam-dma-transfer.
- [25] Pan Docs, «LCD Position and Scrolling,» [Ηλεκτρονικό]. Available: <https://gbdev.io/pandocs/Scrolling.html>.
- [26] Pan Docs, «Joypad Input,» [Ηλεκτρονικό]. Available: https://gbdev.io/pandocs/Joypad_Input.html.
- [27] Pan Docs, «Sound Controller,» [Ηλεκτρονικό]. Available: https://gbdev.io/pandocs/Sound_Controller.html.
- [28] Pan Docs, «OAM DMA Transfer,» [Ηλεκτρονικό]. Available: https://gbdev.io/pandocs/OAM_DMA_Transfer.html#oam-dma-transfer.
- [29] Pan Docs, «Interrupts,» [Ηλεκτρονικό]. Available: <https://gbdev.io/pandocs/Interrupts.html>.
- [30] M. Rawer, «What is inside the Game Boy,» [Ηλεκτρονικό]. Available: <http://marc.rawer.de/Gameboy/index.html>.
- [31] Sharp, «Microcomputer Databook,» 1996. [Ηλεκτρονικό]. Available: https://web.archive.org/web/20220129113513/http://bitsavers.informatik.uni-stuttgart.de/components/sharp/_dataBooks/1996_Sharp_Microcomputer_Data_Book.pdf.
- [32] R. Vaccari, «A journey into GameBoy emulation,» [Ηλεκτρονικό]. Available: https://robertovaccari.com/blog/2020_09_26_gameboy/.
- [33] Pan Docs, «Interrupt Sources,» [Ηλεκτρονικό]. Available: https://gbdev.io/pandocs/Interrupt_Sources.html.

- [34] Pan Docs, «CGB Registers,» [Ηλεκτρονικό]. Available: https://gbdev.io/pandocs/CGB_Registers.html.
- [35] Pan Docs, «LCD Control,» [Ηλεκτρονικό]. Available: <https://gbdev.io/pandocs/LCDC.html>.
- [36] «The Timing of LYC STAT Handlers,» [Ηλεκτρονικό]. Available: https://gbdev.io/guides/lyc_timing.html.
- [37] Pan Docs, «LCD Status Registers,» [Ηλεκτρονικό]. Available: <https://gbdev.io/pandocs/STAT.html>.
- [38] Pan Docs, «Palettes,» [Ηλεκτρονικό]. Available: <https://gbdev.io/pandocs/Palettes.html>.
- [39] Pan Docs, «OAM,» [Ηλεκτρονικό]. Available: <https://gbdev.io/pandocs/OAM.html>.
- [40] Pan Docs, «Pixel FIFO,» [Ηλεκτρονικό]. Available: https://gbdev.io/pandocs/pixel_fifo.html.
- [41] Pan Docs, «Power Up Sequence,» [Ηλεκτρονικό]. Available: https://gbdev.io/pandocs/Power_Up_Sequence.html.
- [42] Wikipedia, «List of Game Boy Color games,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/List_of_Game_Boy_Color_games.
- [43] endrift, «"Holy Grail" Bugs in Emulation, Part 1,» 29 5 2017. [Ηλεκτρονικό]. Available: <https://mgba.io/2017/05/29/holy-grail-bugs/>.
- [44] M. Sullivan, «Meet the Game Boy Instruction Set,» [Ηλεκτρονικό]. Available: <https://meganesulli.com/blog/game-boy-opcodes/>.

9 Παράρτημα

9.1 Σετ εντολών του επεξεργαστή του Game Boy (Game Boy Instruction Set)

Game Boy CPU Instructions

8-bit opcodes

Game Boy CPU instructions, organized by opcode

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
8x	NOP 1 1 3 3 --	LD BC, d16 3 3 --	LD (BC), A 1 2 --	INC BC 1 2 --	INC B 1 1 Z 0 H -	DEC B 1 1 Z 1 H -	LD B, dB 2 2 --	RLCA 1 1 0 0 0 A7	LD (a16), SP 3 5 --	ADD HL, BC 1 2 - 0 H CY	LD A, (BC) 1 2 --	DEC BC 1 2 --	INC C 1 1 Z 0 H -	DEC C 1 1 Z 1 H -	LD C, dB 2 2 --	RRCA 1 1 0 0 0 A0
1x	STOP 2 1 --	LD DE, d16 3 3 --	LD (DE), A 1 2 --	INC DE 1 2 Z 0 H -	INC D 1 1 Z 0 H -	DEC D 1 1 Z 1 H -	LD D, dB 2 2 --	RLA 1 1 0 0 0 A7	JR s8 1 2 - 0 H CY	ADD HL, DE 1 2 - 0 H CY	LD A, (DE) 1 2 --	DEC DE 1 2 --	INC E 1 1 Z 0 H -	DEC E 1 1 Z 1 H -	LD E, dB 2 2 --	RRD 1 1 0 0 0 A0
2x	JR NZ, s8 2 3/2 --	LD HL, d16 3 3 --	LD (HL+), A 1 2 --	INC HL 1 2 Z 0 H -	INC H 1 1 Z 0 H -	DEC H 1 1 Z 1 H -	LD H, dB 2 2 --	DAA 1 1 Z - 0 CY	JR Z, s8 1 2 - 0 H CY	ADD HL, HL 1 2 - 0 H CY	LD A, (HL+) 1 2 --	DEC HL 1 2 --	INC L 1 1 Z 0 H -	DEC L 1 1 Z 1 H -	LD L, dB 2 2 --	CPL 1 1 - 1 1 -
3x	JR NC, s8 2 3/2 --	LD SP, d16 3 3 --	LD (HL-), A 1 2 --	INC SP 1 2 Z 0 H -	INC (HL) 1 3 Z 0 H -	DEC (HL) 1 3 Z 1 H -	LD (HL), dB 2 3 --	SCF 1 1 0 0 0 1	JR C, s8 1 2 - 0 H CY	ADD HL, SP 1 2 - 0 H CY	LD A, (HL-) 1 2 --	DEC SP 1 2 --	INC A 1 1 Z 0 H -	DEC A 1 1 Z 1 H -	LD A, dB 2 2 --	CCF 1 1 0 0 0 1CY
4x	LD B, B 1 1 --	LD B, C 1 1 --	LD B, D 1 1 --	LD B, E 1 1 --	LD B, H 1 1 --	LD B, L 1 1 --	LD B, (HL) 1 1 --	LD B, A 1 1 --	LD C, B 1 1 --	LD C, C 1 1 --	LD C, D 1 1 --	LD C, E 1 1 --	LD C, H 1 1 --	LD C, L 1 1 --	LD C, A 1 1 --	LD C, (HL) 1 1 --
5x	LD D, B 1 1 --	LD D, C 1 1 --	LD D, D 1 1 --	LD D, E 1 1 --	LD D, H 1 1 --	LD D, L 1 1 --	LD D, (HL) 1 1 --	LD D, A 1 1 --	LD E, B 1 1 --	LD E, C 1 1 --	LD E, D 1 1 --	LD E, E 1 1 --	LD E, H 1 1 --	LD E, L 1 1 --	LD E, A 1 1 --	LD E, (HL) 1 1 --
6x	LD H, B 1 1 --	LD H, C 1 1 --	LD H, D 1 1 --	LD H, E 1 1 --	LD H, H 1 1 --	LD H, L 1 1 --	LD H, (HL) 1 1 --	LD H, A 1 1 --	LD L, B 1 1 --	LD L, C 1 1 --	LD L, D 1 1 --	LD L, E 1 1 --	LD L, H 1 1 --	LD L, L 1 1 --	LD L, A 1 1 --	LD L, (HL) 1 1 --
7x	LD (HL), B 1 2 --	LD (HL), C 1 2 --	LD (HL), D 1 2 --	LD (HL), E 1 2 --	LD (HL), H 1 2 --	LD (HL), L 1 2 --	HALT 1 1 --	LD (HL), A 1 2 --	LD A, B 1 1 --	LD A, C 1 1 --	LD A, D 1 1 --	LD A, E 1 1 --	LD A, H 1 1 --	LD A, L 1 1 --	LD A, (HL) 1 2 --	LD A, A 1 1 --
8x	ADD A, B 1 1 Z 0 H CY	ADD A, C 1 1 Z 0 H CY	ADD A, D 1 1 Z 0 H CY	ADD A, E 1 1 Z 0 H CY	ADD A, H 1 1 Z 0 H CY	ADD A, L 1 1 Z 0 H CY	ADD A, (HL) 1 2 Z 0 H CY	ADD A, A 1 1 Z 0 H CY	ADC A, B 1 1 Z 0 H CY	ADC A, C 1 1 Z 0 H CY	ADC A, D 1 1 Z 0 H CY	ADC A, E 1 1 Z 0 H CY	ADC A, H 1 1 Z 0 H CY	ADC A, L 1 1 Z 0 H CY	ADC A, (HL) 1 2 Z 0 H CY	ADC A, A 1 1 Z 0 H CY
9x	SUB B 1 1 Z 1 H CY	SUB C 1 1 Z 1 H CY	SUB D 1 1 Z 1 H CY	SUB E 1 1 Z 1 H CY	SUB H 1 1 Z 1 H CY	SUB L 1 1 Z 1 H CY	SUB (HL) 1 2 Z 1 H CY	SUB A 1 1 Z 1 H CY	SBC A, B 1 1 Z 1 H CY	SBC A, C 1 1 Z 1 H CY	SBC A, D 1 1 Z 1 H CY	SBC A, E 1 1 Z 1 H CY	SBC A, H 1 1 Z 1 H CY	SBC A, L 1 1 Z 1 H CY	SBC A, (HL) 1 2 Z 1 H CY	SBC A, A 1 1 Z 1 H CY
Ax	AND B 1 1 Z 0 1 0	AND C 1 1 Z 0 1 0	AND D 1 1 Z 0 1 0	AND E 1 1 Z 0 1 0	AND H 1 1 Z 0 1 0	AND L 1 1 Z 0 1 0	AND (HL) 1 2 Z 0 1 0	AND A 1 1 Z 0 1 0	XOR B 1 1 Z 0 0 0	XOR C 1 1 Z 0 0 0	XOR D 1 1 Z 0 0 0	XOR E 1 1 Z 0 0 0	XOR H 1 1 Z 0 0 0	XOR L 1 1 Z 0 0 0	XOR (HL) 1 2 Z 0 0 0	XOR A 1 1 Z 0 0 0
Bx	OR B 1 1 Z 0 0 0	OR C 1 1 Z 0 0 0	OR D 1 1 Z 0 0 0	OR E 1 1 Z 0 0 0	OR H 1 1 Z 0 0 0	OR L 1 1 Z 0 0 0	OR (HL) 1 2 Z 0 0 0	OR A 1 1 Z 0 0 0	CP B 1 1 Z 1 H CY	CP C 1 1 Z 1 H CY	CP D 1 1 Z 1 H CY	CP E 1 1 Z 1 H CY	CP H 1 1 Z 1 H CY	CP L 1 1 Z 1 H CY	CP (HL) 1 2 Z 1 H CY	CP A 1 1 Z 1 H CY
Cx	RET NZ 1 5/2 --	POP BC 1 3 --	JP NZ, a16 3 4/3 --	JP a16 3 4 --	CALL NZ, a16 3 6/3 --	ADD A, dB 1 4 --	RST 0 2 2 --	RET Z 1 4 5/2	RET 1 4 5/2	RET JP Z, a16 3 4/3 --	CALL Z, a16 3 6/3 --	CALL Z, a16 3 6/3 --	CALL Z, a16 3 6/3 --	CALL Z, a16 3 6/3 --	ADD A, dB 2 2 --	RST 1 1 4 --
Dx	LD (a8), A 1 2 --	POP HL 1 3 --	LD (C), A 1 2 --	LD (C), A 1 2 --	PUSH HL 1 4 --	AND dB 2 2 Z 0 1 0	RST 4 2 2 --	ADD SP, s8 1 4 --	JP HL 1 4 --	LD (a16), A 3 4 --					XOR dB 2 2 Z 0 0 0	RST 5 1 4 --
Fx	LD A, (a8) 1 2 --	POP AF 1 3 --	LD A, (C) 1 2 --	DI 1 1 --	PUSH AF 1 4 --	OR dB 2 2 Z 0 0 0	RST 6 2 2 --	LD HL, SP+s8 2 3 --	LD SP, HL 1 2 --	LD A, (a16) 3 4 --	EI 1 1 --				CP dB 2 2 Z 1 H CY	RST 7 1 4 --

16-bit opcodes, where the first 8 bits are 0xCB

Game Boy CPU instructions for opcodes prefixed by 'CB'

	x6	x7	x2	x3	x4	x5	xA	xB	xC	xD	xE	xF				
8x	RLC B 2 2 Z 0 0 B7	RLC C 2 2 Z 0 0 C7	RLC D 2 2 Z 0 0 D7	RLC E 2 2 Z 0 0 E7	RLC H 2 2 Z 0 0 H7	RLC L 2 2 Z 0 0 L7	RLC (HL) 2 2 Z 0 0 (HL)7	RLC A 2 2 Z 0 0 A7	RRC B 2 2 Z 0 0 B0	RRC C 2 2 Z 0 0 C0	RRC D 2 2 Z 0 0 D0	RRC E 2 2 Z 0 0 E0	RRC H 2 2 Z 0 0 H0	RRC L 2 2 Z 0 0 L0	RRC (HL) 2 2 Z 0 0 (HL)0	RRC A 2 2 Z 0 0 A0
1x	RL B 2 2 Z 0 0 B7	RL C 2 2 Z 0 0 C7	RL D 2 2 Z 0 0 D7	RL E 2 2 Z 0 0 E7	RL H 2 2 Z 0 0 H7	RL L 2 2 Z 0 0 L7	RL (HL) 2 2 Z 0 0 (HL)7	RL A 2 2 Z 0 0 A7	RR B 2 2 Z 0 0 B0	RR C 2 2 Z 0 0 C0	RR D 2 2 Z 0 0 D0	RR E 2 2 Z 0 0 E0	RR H 2 2 Z 0 0 H0	RR L 2 2 Z 0 0 L0	RR (HL) 2 2 Z 0 0 (HL)0	RR A 2 2 Z 0 0 A0
2x	SLA B 2 2 Z 0 0 B7	SLA C 2 2 Z 0 0 C7	SLA D 2 2 Z 0 0 D7	SLA E 2 2 Z 0 0 E7	SLA H 2 2 Z 0 0 H7	SLA L 2 2 Z 0 0 L7	SLA (HL) 2 2 Z 0 0 (HL)7	SLA A 2 2 Z 0 0 A7	SRA B 2 2 Z 0 0 B0	SRA C 2 2 Z 0 0 C0	SRA D 2 2 Z 0 0 D0	SRA E 2 2 Z 0 0 E0	SRA H 2 2 Z 0 0 H0	SRA L 2 2 Z 0 0 L0	SRA (HL) 2 2 Z 0 0 (HL)0	SRA A 2 2 Z 0 0 A0
3x	SWAP B 2 2 Z 0 0 B	SWAP C 2 2 Z 0 0 C	SWAP D 2 2 Z 0 0 D	SWAP E 2 2 Z 0 0 E	SWAP H 2 2 Z 0 0 H	SWAP L 2 2 Z 0 0 L	SWAP (HL) 2 2 Z 0 0 (HL)	SWAP A 2 2 Z 0 0 A	SRL B 2 2 Z 0 0 B0	SRL C 2 2 Z 0 0 C0	SRL D 2 2 Z 0 0 D0	SRL E 2 2 Z 0 0 E0	SRL H 2 2 Z 0 0 H0	SRL L 2 2 Z 0 0 L0	SRL (HL) 2 2 Z 0 0 (HL)0	SRL A 2 2 Z 0 0 A0
4x	BIT 0, B 2 2 1r0 0 1 -	BIT 0, C 2 2 1r0 0 1 -	BIT 0, D 2 2 1r0 0 1 -	BIT 0, E 2 2 1r0 0 1 -	BIT 0, H 2 2 1r0 0 1 -	BIT 0, L 2 2 1r0 0 1 -	BIT 0, (HL) 2 2 1(HL)0 0 1 -	BIT 0, A 2 2 1r0 0 1 -	BIT 1, B 2 2 1r1 0 1 -	BIT 1, C 2 2 1r1 0 1 -	BIT 1, D 2 2 1r1 0 1 -	BIT 1, E 2 2 1r1 0 1 -	BIT 1, H 2 2 1r1 0 1 -	BIT 1, L 2 2 1r1 0 1 -	BIT 1, (HL) 2 2 1(HL)1 0 1 -	BIT 1, A 2 2 1r1 0 1 -
5x	BIT 2, B 2 2 1r2 0 1 -	BIT 2, C 2 2 1r2 0 1 -	BIT 2, D 2 2 1r2 0 1 -	BIT 2, E 2 2 1r2 0 1 -	BIT 2, H 2 2 1r2 0 1 -	BIT 2, L 2 2 1r2 0 1 -	BIT 2, (HL) 2 2 1(HL)2 0 1 -	BIT 2, A 2 2 1r2 0 1 -	BIT 3, B 2 2 1r3 0 1 -	BIT 3, C 2 2 1r3 0 1 -	BIT 3, D 2 2 1r3 0 1 -	BIT 3, E 2 2 1r3 0 1 -	BIT 3, H 2 2 1r3 0 1 -	BIT 3, L 2 2 1r3 0 1 -	BIT 3, (HL) 2 2 1(HL)3 0 1 -	BIT 3, A 2 2 1r3 0 1 -
6x	BIT 4, B 2 2 1r4 0 1 -	BIT 4, C 2 2 1r4 0 1 -	BIT 4, D 2 2 1r4 0 1 -	BIT 4, E 2 2 1r4 0 1 -	BIT 4, H 2 2 1r4 0 1 -	BIT 4, L 2 2 1r4 0 1 -	BIT 4, (HL) 2 2 1(HL)4 0 1 -	BIT 4, A 2 2 1r4 0 1 -	BIT 5, B 2 2 1r5 0 1 -	BIT 5, C 2 2 1r5 0 1 -	BIT 5, D 2 2 1r5 0 1 -	BIT 5, E 2 2 1r5 0 1 -	BIT 5, H 2 2 1r5 0 1 -	BIT 5, L 2 2 1r5 0 1 -	BIT 5, (HL) 2 2 1(HL)5 0 1 -	BIT 5, A 2 2 1r5 0 1 -
7x	BIT 6, B 2 2 1r6 0 1 -	BIT 6, C 2 2 1r6 0 1 -	BIT 6, D 2 2 1r6 0 1 -	BIT 6, E 2 2 1r6 0 1 -	BIT 6, H 2 2 1r6 0 1 -	BIT 6, L 2 2 1r6 0 1 -	BIT 6, (HL) 2 2 1(HL)6 0 1 -	BIT 6, A 2 2 1r6 0 1 -	BIT 7, B 2 2 1r7 0 1 -	BIT 7, C 2 2 1r7 0 1 -	BIT 7, D 2 2 1r7 0 1 -	BIT 7, E 2 2 1r7 0 1 -	BIT 7, H 2 2 1r7 0 1 -	BIT 7, L 2 2 1r7 0 1 -	BIT 7, (HL) 2 2 1(HL)7 0 1 -	BIT 7, A 2 2 1r7 0 1 -
8x	RES 0, B 2 2 --	RES 0, C 2 2 --	RES 0, D 2 2 --	RES 0, E 2 2 --	RES 0, H 2 2 --	RES 0, L 2 2 --	RES 0, (HL) 2 2 --	RES 0, A 2 2 --	RES 1, B 2 2 --	RES 1, C 2 2 --	RES 1, D 2 2 --	RES 1, E 2 2 --	RES 1, H 2 2 --	RES 1, L 2 2 --	RES 1, (HL) 2 2 --	RES 1, A 2 2 --
9x	RES 2, B 2 2 --	RES 2, C 2 2 --	RES 2, D 2 2 --	RES 2, E 2 2 --	RES 2, H 2 2 --	RES 2, L 2 2 --	RES 2, (HL) 2 2 --	RES 2, A 2 2 --	RES 3, B 2 2 --	RES 3, C 2 2 --	RES 3, D 2 2 --	RES 3, E 2 2 --	RES 3, H 2 2 --	RES 3, L 2 2 --	RES 3, (HL) 2 2 --	RES 3, A 2 2 --
Ax	RES 4, B 2 2 --	RES 4, C 2 2 --	RES 4, D 2 2 --	RES 4, E 2 2 --	RES 4, H 2 2 --	RES 4, L 2 2 --	RES 4, (HL) 2 2 --	RES 4, A 2 2 --	RES 5, B 2 2 --	RES 5, C 2 2 --	RES 5, D 2 2 --	RES 5, E 2 2 --	RES 5, H 2 2 --	RES 5, L 2 2 --	RES 5, (HL) 2 2 --	RES 5, A 2 2 --
Bx	RES 6, B 2 2 --	RES 6, C 2 2 --	RES 6, D 2 2 --	RES 6, E 2 2 --	RES 6, H 2 2 --	RES 6, L 2 2 --	RES 6, (HL) 2 2 --	RES 6, A 2 2 --	RES 7, B 2 2 --	RES 7, C 2 2 --	RES 7, D 2 2 --	RES 7, E 2 2 --	RES 7, H 2 2 --	RES 7, L 2 2 --	RES 7, (HL) 2 2 --	RES 7, A 2 2 --
Cx	SET 0, B 2 2 --	SET 0, C 2 2 --	SET 0, D 2 2 --	SET 0, E 2 2 --	SET 0, H 2 2 --	SET 0, L 2 2 --	SET 0, (HL) 2 2 --	SET 0, A 2 2 --	SET 1, B 2 2 --	SET 1, C 2 2 --	SET 1, D 2 2 --	SET 1, E 2 2 --	SET 1, H 2 2 --	SET 1, L 2 2 --	SET 1, (HL) 2 2 --	SET 1, A 2 2 --
Dx	SET 2, B 2 2 --	SET 2, C 2 2 --	SET 2, D 2 2 --	SET 2, E 2 2 --	SET 2, H 2 2 --	SET 2, L 2 2 --	SET 2, (HL) 2 2 --	SET 2, A 2 2 --	SET 3, B 2 2 --	SET 3, C 2 2 --	SET 3, D 2 2 --	SET 3, E 2 2 --	SET 3, H 2 2 --	SET 3, L 2 2 --	SET 3, (HL) 2 2 --	SET 3, A 2 2 --
Ex	SET 4, B 2 2 --	SET 4, C 2 2 --	SET 4, D 2 2 --	SET 4, E 2 2 --	SET 4, H 2 2 --	SET 4, L 2 2 --	SET 4, (HL) 2 2 --	SET 4, A 2 2 --	SET 5, B 2 2 --	SET 5, C 2 2 --	SET 5, D 2 2 --	SET 5, E 2 2 --	SET 5, H 2 2 --	SET 5, L 2 2 --	SET 5, (HL) 2 2 --	SET 5, A 2 2 --
Fx	SET 6, B 2 2 --	SET 6, C 2 2 --	SET 6, D 2 2 --	SET 6, E 2 2 --	SET 6, H 2 2 --	SET 6, L 2 2 --	SET 6, (HL) 2 2 --	SET 6, A 2 2 --	SET 7, B 2 2 --	SET 7, C 2 2 --	SET 7, D 2 2 --	SET 7, E 2 2 --	SET 7, H 2 2 --	SET 7, L 2 2 --	SET 7, (HL) 2 2 --	SET 7, A 2 2 --

Σχήμα 19 - Το ολοκληρωμένο σετ εντολών του Game Boy [44]