

Tetris

Mikhail Peshkov
Vyacheslav Chelondaev
Sayan Radnaev

2022

Contents

1	Overview	2
2	Hardware	3
2.1	LCD	3
2.2	IOreg	4
2.3	fieldCell chip	4
2.4	Main chip	4
2.4.1	IO addresses	4
2.4.2	Control register	5
2.4.3	Keyboard	5
2.4.4	Game over	5
3	Software	6

Chapter 1

Overview

Lorem ipsum

Chapter 2

Hardware

2.1 LCD

		west	
		0	1
e a s t	0		
	1		

Figure 2.1: Color scheme

This project uses custom LCD library written in Java. Resolution of the LCD is 17x20 and it has 4-bit color scheme. In the library implementation, memory is statically allocated so that you can feed data to one LCD and display it on another. There are 32-bit inputs on the east and west sides. Each display input only processes the upper 17 bits. To draw each row, the corresponding inputs are taken and matched against each other. If 0 is stored at the same bit position, then a white pixel is displayed. If the bit from the east input is 1 and the bit from the west input is 0 then a green pixel is drawn. If the bit from the west input is 1 and from the east input is 0, then a black pixel is drawn. If the same bit position in the east and west input is 1, then a blue pixel is displayed. Coordinates are counted from the top left.

2.2 IOreg

2.3 fieldCell chip

The fieldCell chip has 9 inputs: xRow - the sprite row shifted to the X coordinate, y - the current y value, id - the id value for this chip, clrMap - a bit string containing the id of the chips in the format of descending high bits, lastIn - the previous line of the field, clk - clock frequency, check - command to check xRow for intersection with field, union - command to merge xRow with field, full - command to clear rows specified in clrMap. Also this chip has 10 outputs. Most of them just send the same data that the inputs receive. The exception is idNext - it increments the current id by 1 and the full and fail outputs. full produces a logical one if the value of the register holding the string becomes 0xffc00000 which corresponds to 10 binary ones. Due to the structure of the **field** chip, if this output is raised in at least one chip, then its value will come to all the others. The output of fail is a logical one if there is an overlapping xRow intersection with the current row.

When the chip id matches the y value, the cs tunnel is activated. If cs is enabled and a union command is issued, and full is omitted, then the register value and xRow are combined. If full is raised and the chip id is less than or equal to y, then the lastIn value is written to the register. Thus, the field is cleared of filled lines in 8 cycles. Since there are only 4 lines in each sprite, 2 cycles are allocated for each line: the first for writing the value to the register, the second for clearing the lines if full is raised.

2.4 Main chip

Main chip has 3 1-bit inputs for the keyboard, 1 1-bit output for the end of the game, and 6 4-bit outputs to display the score.

2.4.1 IO addresses

Name	Mode	Address(hex)
Control register	w	f0
Write data	w	f1
Read data	r	f2
Figure X coordinate	r/w	f3
Figure Y coordinate	r/w	f4
Score	r/w	f5
Read keyboard	r	f6
Read status	r	f7

Addresses 0xf0 to 0xf7 are reserved for I/O. Addresses from 0xf8 to 0xff are reserved for the stack. Working with addresses that have the ability to read and write is described in the **ioreg** chip.

2.4.2 Control register

This register activates field and figure chips and controls them.

Name	Receiver	Purpose
csFig	figure	enable figure chip
csField	field	enable field chip
getNew	figure	return next figure sprite
rotate	figure	return next turn sprite
check	field	command to check the coordinates of a figure and its position on the field
union	field	command to union a figure with a field
draw	field	command to draw a figure on the LCD
back	figure	return previous turn sprite

2.4.3 Keyboard

This part of the chip has 3 inputs, each of which is responsible for the movement of the figure: left, turn, right. When one of the inputs receives a logical one, the value corresponding to the purpose of the input is written to the register in accordance with the table. The register value is cleared after the Cdm-8 reads its value.

Bit mask		
left	rotate	right
0b00000010	0b00000011	0b00000001

2.4.4 Game over

When it is no longer possible to take the next figure, the program ends. The address of the last command is 0x54. When Cdm-8 reaches it, a logical one appears at the output of gameOver.

Chapter 3

Software