Federal State Budgetary Educational Institution of Higher Education. "National Research University "Moscow Power Engineering Institute"

Department of VMSS

Laboratory work No. 3

CONSTRUCTION OF A MICROPROGRAM CONTROL DEVICE

Course: OEVM

Completed by: Team 6

Selyukov K

Task:

1. Build a circuit containing a DMC8-master microcontroller and 3 DMC8-slave microcontrollers. Connect these microcontrollers using a serial interface.

2. Set up the operation of the DMC8-slave circuit according to the option.

3. Configure the operation of the DMC8-master circuit to receive/transmit information to three DMC8-slaves.

1.

Watch. Create a clock diagram (hours, minutes, seconds) that meets the following requirements:

the ability to set an alarm for hours, minutes, seconds (via master);

When the alarm goes off, the indicator starts flashing.

2

Indicator. Create a logical diagram of the indicator that meets the following requirements:

output of a two-byte number to two indicators. The first indicator shows the number in the binary number system, the second in the hexadecimal number system;

master can pass a two-byte value for display.

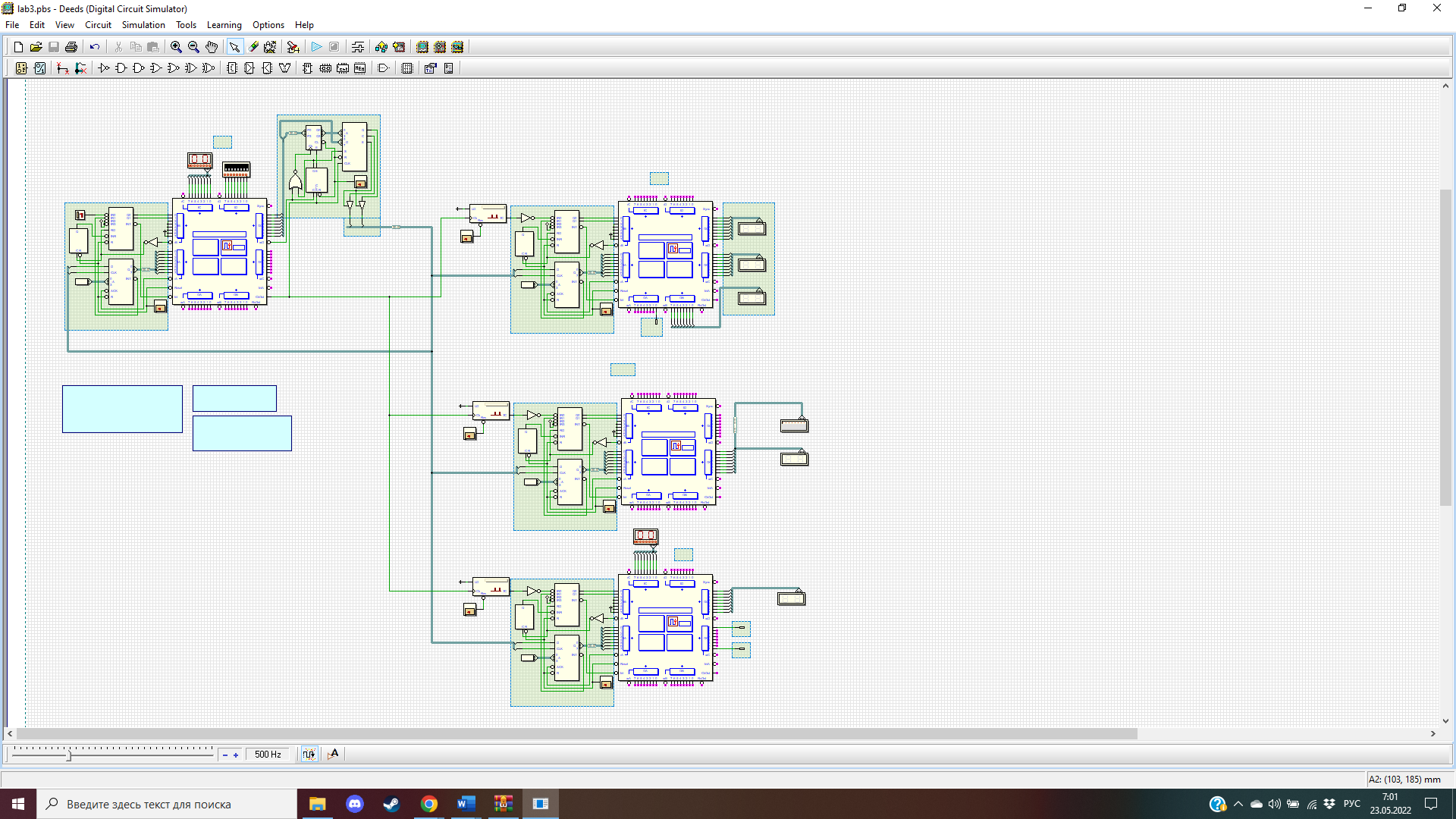
3\*

Code lock. Create a combination lock diagram that meets the following requirements:

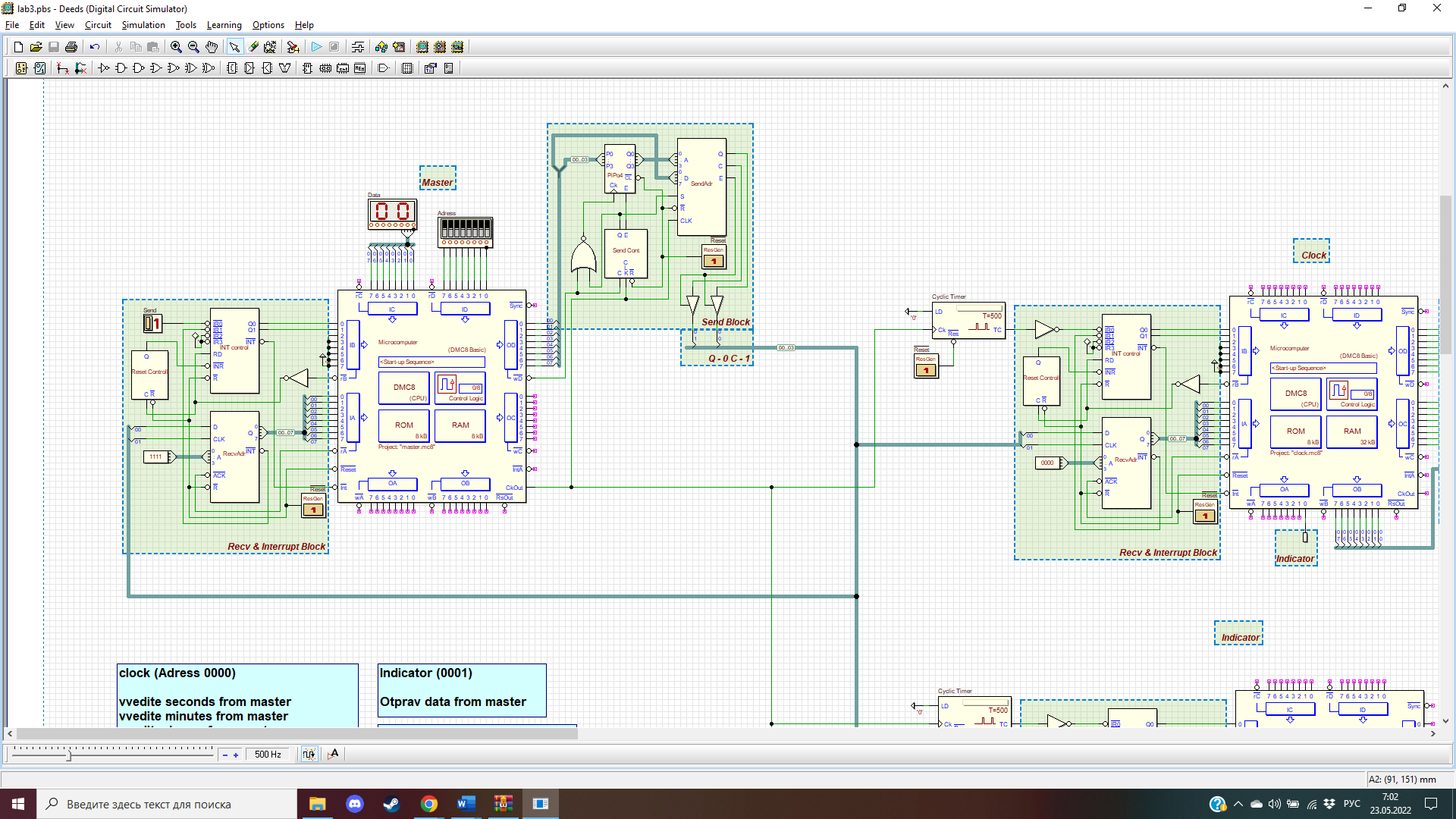
the code combination consists of four bits;

if the first code combination is successful, we generate a signal “Good 1”, otherwise “Alert 1>”;

possibility of receiving signal values ​​or resetting (via master);

General scheme:

Master:



pA EQU 00

pB EQU 01

pC EQU 02

pD EQU 03

ORG 0

JP START

ORG 38h

JP INT

ORG 100h

START:

EI

INF:

JP INF

INT:

IN A, (pB)

CP 0

JP Z, INT\_0

CP 1

JP Z, INT\_1

CP 2

JP Z, INT\_2

CP 3

JP Z, INT\_3

JP INT\_EXIT

INT\_0:

IN A, (pD)

OUT (pD), A

IN A, (pC)

OUT (pD), A

JP INT\_EXIT

INT\_1:

IN A, (pA)

OUT (pB), A

JP INT\_EXIT

INT\_2:

JP INT\_EXIT

INT\_3:

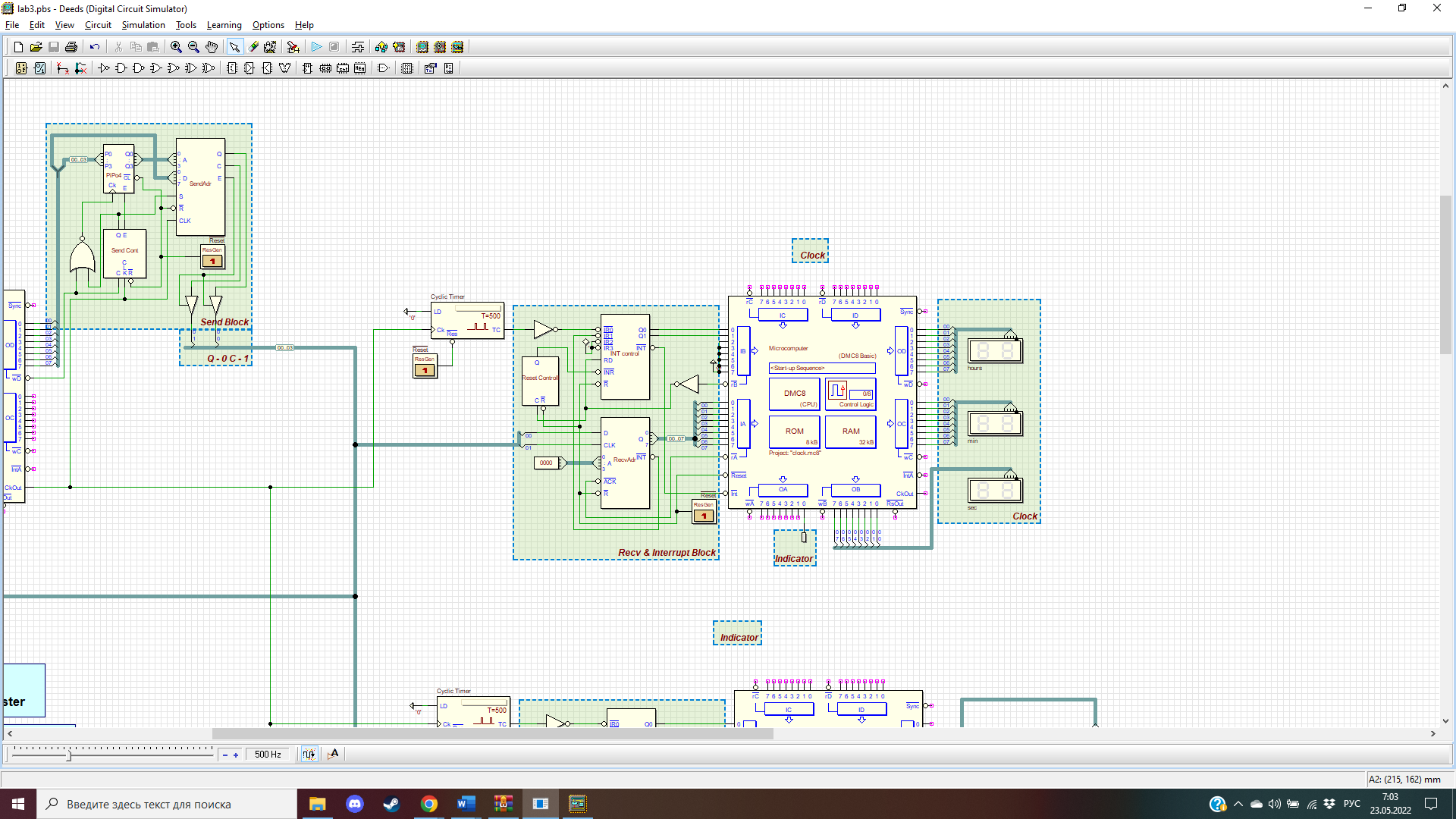
INT\_EXIT:

IN A, (pB)

EI

RET

Watch:



pA EQU 00

pB EQU 01

pC EQU 02

pD EQU 03

ORG 0

JP START

ORG 38h

JP INT

ORG 100h

START:

LD A, 0

LD (8000h), A ; Identifying what is already loaded into the alarm

LD (8001h), A ; Checking if the alarm has been activated

LD A, 59h

LD (8002h), A ; By default, we load 59 secs into this cell.

LD (8003h), A ; By default, we load 59 minutes into this cell.

LD A, 23h

LD (8004h), A ; By default we load 23 hours into this cell

LD A, 1

LD (8005h), A ; With this we will change the signal when the alarm goes off

EI

INF:

JP INF

INT:

IN A, (pB)

; Determine the interrupt number

CP 0

JP Z, INT\_0

CP 1

JP Z, INT\_1

CP 2

JP Z, INT\_2

CP 3

JP Z, INT\_3

JP INT\_EXIT

; Interrupt 0

; Starting the clock

INT\_0:

SEC:

INC B ; add sec

LD A, B

CP 5Ah ; Checking if we have moved to the next minute

JP NZ, TENSEC ; If not, then we check for the need to increase the senior part of the seconds

INC C ; Otherwise, increase by one minute

LD B, 0 ; Reset seconds

JP MIN ; go to minutes

TENSEC:

AND 0Fh ; get the minor part of seconds

CP 0Ah ; Let's check if we jumped over in 9 seconds

JP NZ, MIN ; If not, then we move on to minutes.

LD A, B ; Return to A seconds

ADD A, 10h ; We increase the senior part by 1

AND 0F0h ; leave only the older part

LD B, A ; Load the received seconds into B

MIN:

LD A, C

CP 5Ah ; check if we have moved to the next hour

JP NZ, TENMIN ; If not, check for the need to increase the senior part of the minutes

INC D ; Otherwise, increase by one hour

LD C, 0 ; reset mins

JP HOURS ; go to hours

TENMIN:

AND 0Fh ; get the minor part of minutes

CP 0Ah ; check if we jumped over in 9 minutes

JP NZ, HOURS ; If not, move on to the clock.

LD A, C ; Return to A minutes

ADD A, 10h ; Increase the senior part by 1

AND 0F0h ; leave only the older part

LD C, A ; Load the received minutes into C

HOURS:

LD A, D

CP 24h ; check if we have moved to the next day

JP NZ, TENHOUR ; If not, then we check for the need to increase the senior part of the clock

LD D, 0 ; Otherwise, reset the clock

JP OUTTIME ; move on to the time output

TENHOUR:

AND 0Fh ; get the younger part of the clock

CP 0Ah ; check if we've jumped over 9 hours

JP NZ, OUTTIME ; If not, move on to the time output

LD A, D ; Return to A hours

ADD A, 10h ; increase the senior part by 1

AND 0F0h ; leave only the older part

LD D, A ; Load the received hours into D

OUTTIME:

LD A, B

OUT (pB), A ; display secs

LD A, C

OUT (pC), A ; display mins

LD A, D

OUT (pD), A ; display hours

CHECKALARM: ; Проверка будильника

LD A, (8001h) ; if the alarm was set, here will be 1

CP 1 ; Check if the alarm was set

JP NZ, INT\_EXIT ; if no exit

LD A, (8005h) ; check whats in 8005h

CP 2 ; if 2, change signal from 1 to 0

JP Z, CHANGESIGNAL1

CP 3 ; if 3, change signal from 0 to 1

JP Z, CHANGESIGNAL0

LD A, (8002h) ; ; load in A secs, sent by the master

CP B ; compare

JP NZ, INT\_EXIT ; exit, if !=

LD A, (8003h) ; load in A mins, sent by the master

CP C ; compare

JP NZ, INT\_EXIT ; exit, if !=

LD A, (8004h) ; load in A hours, sent by the master

CP D ; compare

JP NZ, INT\_EXIT ; exit, if !=

CHANGESIGNAL0:

LD A, 2

LD (8005h), A ; load in 8005h 2 for future signal change

LD A, 1

OUT (pA), A ; on the lamp

JP INT\_EXIT ;

CHANGESIGNAL1:

LD A, 3

LD (8005h), A ; load in 8005h 3 for future signal change

LD A, 0

OUT (pA), A ; off the lamp

JP INT\_EXIT ;

;intettupt 1

INT\_1:

LD A, 0

OUT (pA), A ; off the lamp

LD A, 1

LD (8005h), A ; change 8005, so that the light bulb doesn't have to be changed anymore

LD (8001h), A ; say, alarm is on

LD A, (8000h) ; in 8000h alarm loading stage

CP 0 ; if 0, set secs

JP Z, SETSEC

CP 1 ; if 1, set mins

JP Z, SETMIN

CP 2 ; if 2, set hours

JP Z, SETHOURS

SETSEC:

IN A, (pA) ; get secs

LD (8002h), A ; load in 8002h

LD A, (8000h)

INC A

LD (8000h), A ; say that the secs are loaded

JP INT\_EXIT ;

SETMIN:

IN A, (pA) ; get mins

LD (8003h), A ; load in 8003h

LD A, (8000h)

INC A

LD (8000h), A ; say that the minuts are loaded

JP INT\_EXIT ; exit

SETHOURS:

IN A, (pA) ; receive the hours

LD (8004h), A ; load in 8004h

LD A, (8000h)

INC A

LD (8000h), A ; say that the hours are loaded

JP INT\_EXIT ; exit

;Interrupt 2

INT\_2:

JP INT\_EXIT

;Interrupt 3

INT\_3:

;End of interrupt

INT\_EXIT:

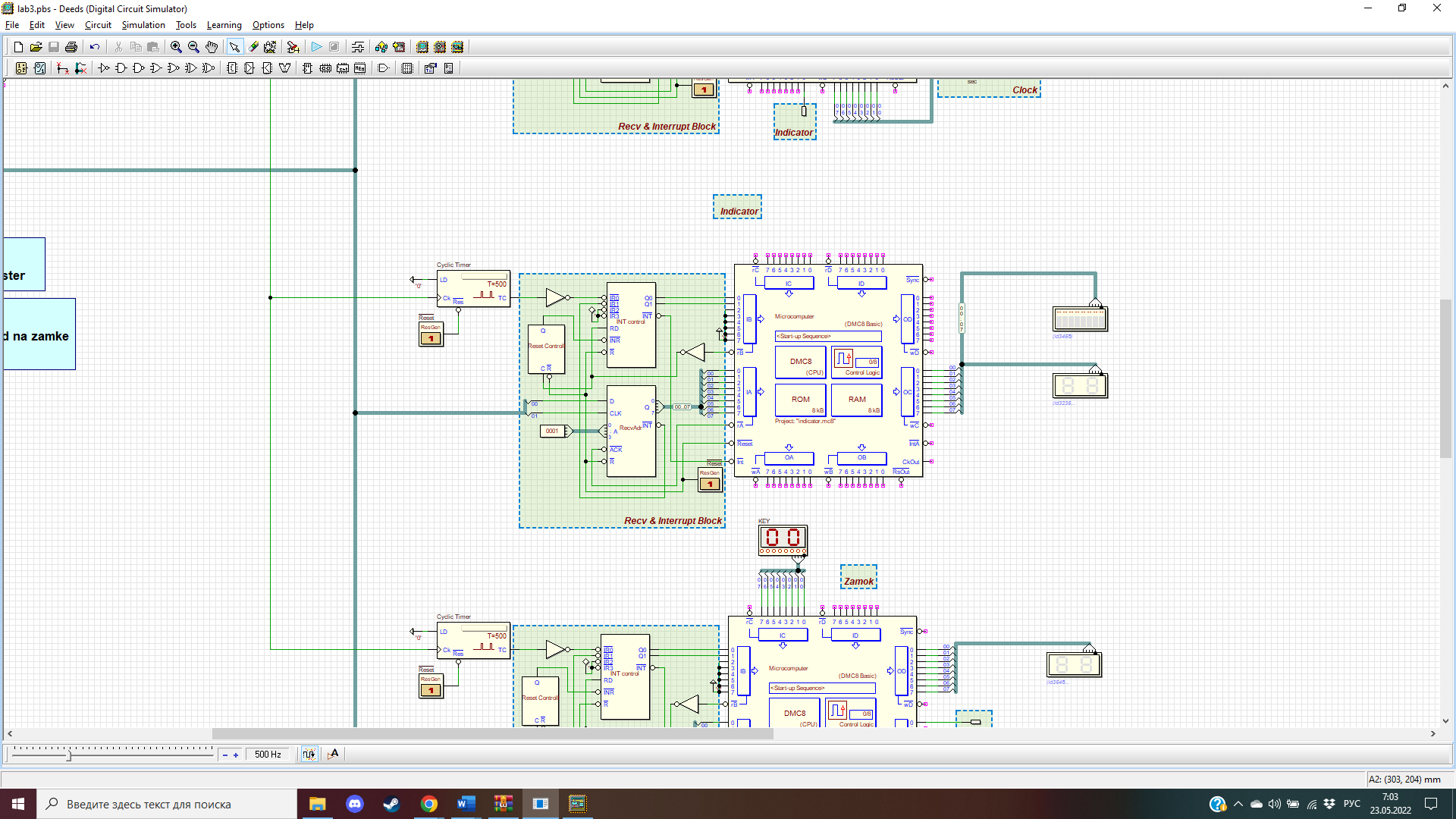
;We tell the interrupt controller that we are ready to accept the following

IN A, (pB)

EI

RET

Indicator:



pA EQU 00

pB EQU 01

pC EQU 02

pD EQU 03

ORG 0

JP START

org 38h

JP INT

org 100h

START:

EI

INF:

JP INF

INT:

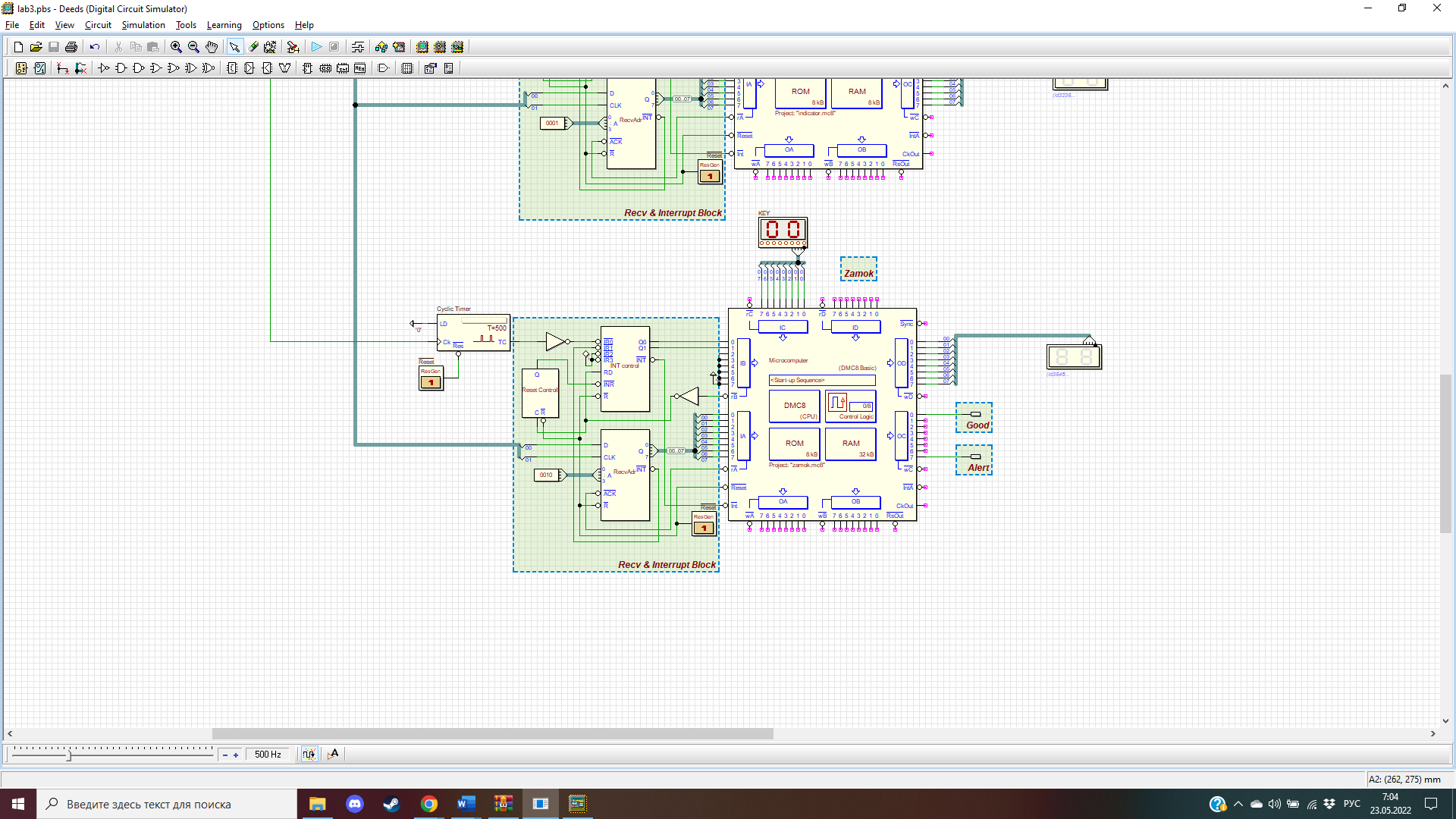
IN A, (pA)

OUT (pC), A

EI

RET

Lock:



pA EQU 00

pB EQU 01

pC EQU 02

pD EQU 03

ORG 0

JP START

ORG 38h

JP INT

ORG 100h

START:

EI

INF:

JP INF

INT:

LD A, 0

OUT (pC), A

IN A, (pB)

CP 0

JP Z, INT\_0

CP 1

JP Z, INT\_1

CP 2

JP Z, INT\_2

CP 3

JP Z, INT\_3

JP INT\_EXIT

INT\_0:

IN A, (pC)

LD (8000h), A

JP INT\_EXIT

INT\_1:

LD A, (8000h)

LD B, A

IN A, (pA)

OUT (pD), A

CP B

jp z, Good

jp nz, Alert

Good:

LD A, 01b

OUT (pC), A

JP INT\_EXIT

Alert:

LD A, 10000000b

OUT (pC), A

JP INT\_EXIT

INT\_2:

JP INT\_EXIT

INT\_3:

INT\_EXIT:

IN A, (pB)

EI

RET