Experiment #10

8259 Programmable Interrupt Controller Application

Birzeit University
Faculty of Engineering and Technology
Department of Electrical and Computer Engineering

Abstract

This experiment aims at understanding and expanding 8086 Interrupt capabilities using Intel 8259 PIC that includes reviewing Intel 8259 control, its initialization and operational modes. Also, in this experiment we will use PPI 8255, and PIT 8253/4.

PART I Theoretical Introduction

The 8259A programmable interrupt controller chip accepts interrupts from up to eight different devices. If any one of the devices requests service, the 8259 will toggle an interrupt output line (connected to the CPU) and pass a programmable interrupt vector to the CPU. We can cascade the device to support up to 64 devices by connecting nine 8259s together: eight of the devices with eight inputs each whose outputs become the eight inputs of the ninth device. A typical PC uses two of these devices to provide 15 interrupt inputs. In this experiment we will focus on a single PIC (no cascading). **Figure 2** shows the pin-out of the 8259A. [**Refer to the datasheet for more info**]

Interrupt request inputs (IR_0-IR_7) are used to request an interrupt and to connect to a slave in a system with multiple 8259As.

The **interrupt output (INT)** connects to the INTR pin on the microprocessor from the master and is connected to a master IR pin on a slave.

Interrupt acknowledge (INTA) is an input that connects to the signal on the system. In a system with a master and slaves, only the master signal is connected.

The **A0 address input** selects different command words within the 8259A.

Chip select (CS) enables the 8259A for programming and control.

Slave program/enable buffer (SP/EN) is a dual-function pin. When the 8259A is in buffered mode, this is an output that controls the data bus transceivers in a large microprocessor-based system. When the 8259A is not in the buffered mode, this pin programs the device as a master (1) or a slave (0).

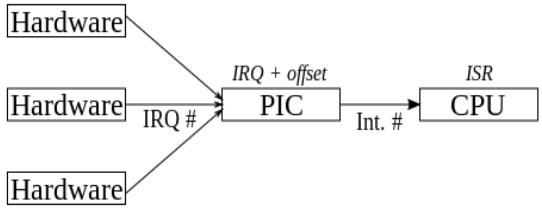


Figure 1: Path of an interrupt, from hardware to CPU

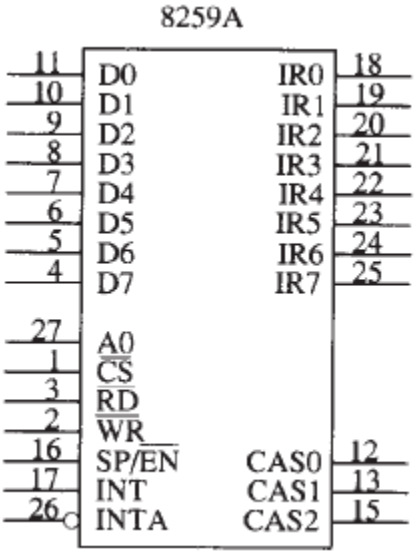


Figure 2: The pin-out of the 8259A programmable interrupt controller (PIC).

The 8259A is programmed by initialization and operation command words. **Initialization command words** (ICWs) are programmed before the 8259A is able to function in the system and dictate the basic operation of the 8259A. **Operation command words** (OCWs) are programmed during the normal course of operation. The OCWs control the operation of the 8259A.

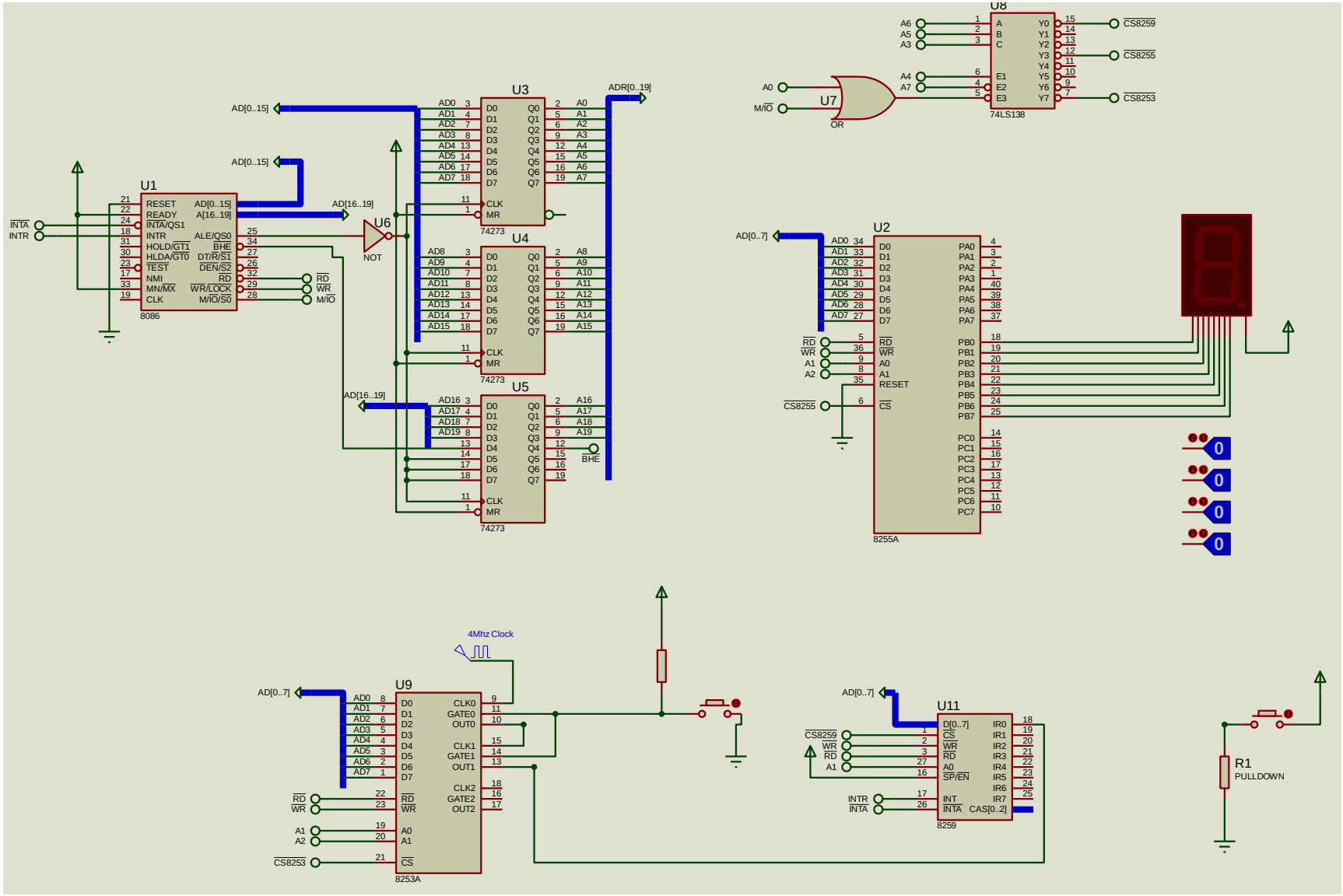
PART II Pre-Lab

(This part should be handed on to the teaching assistant in your Lab)

1. Review Intel 8259 Programmable Interrupt Controller and its modes of operation. Make sure you read the datasheet.
2. What would be the I/O ports for the 8259 if direct addressing mode is used with only 8086 A4 being "1" and 8086 A1 being connected to A0 of 8259?
3. Study the TO-DO Practices and write down the values for ICW1, ICW2, and ICW4?
4. What values of OCWs are needed?

PART III Practices

Study the following Figure carefully and Find the addresses for all Peripherals inside, then study the code and understand the configuration for each Component.



```
CODE    SEGMENT PARA 'CODE'
       ASSUME CS:CODE, DS:DATA, SS:STAK

STAK    SEGMENT PARA STACK 'STACK'
       DW 20 DUP(?)
STAK    ENDS

DATA    SEGMENT PARA 'DATA'
; 0 for lighting, 1 otherwise
DIGITS  DB 0C0h, 0F9h, 0A4h, 0B0h, 99h, 92h, 82h, 0F8h, 80h, 90h, 088h, 083h, 0C6h, 0A1h, 86h, 8Eh
          , 00     ; 0 to F
DATA    ENDS



START PROC
  MOV AX, DATA
  MOV DS, AX

CALL INSERT_ISR_INTO_VECTOR_TABLE
     CALL SETUP_8259
     ;; Address of 8253: 78h
     ;; Addrss of  8255: 70h
     ;; Address of 8259: 10h
     MOV AL, 80h                        ; 1000 0000b
     OUT 76h, AL                        ; Setup 8255 as all ports mo0 output
     CALL SETUP_8253

     STI
     XOR AX,AX
ENDLESS:
     CALL SHOW_AX

     CMP AX, 10h
     JNZ ENDLESS
     MOV AX, 0h                 ; MAke AX 0 after reaching 16

     JMP ENDLESS
RET
START ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;          Common Anode Seven Segmeent        ; 0 for lighting, 1 otherwise
SHOW_AX PROC NEAR
  PUSH AX

  XOR BX, BX
  MOV BL, AL
  MOV AL, DIGITS[BX]
  OUT 72h, AL                           ; 8255's Port B at 72h

  POP AX
RET
SHOW_AX ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SETUP_8253 PROC NEAR
        ;; SETUP 8253

        ; --> Used to Divide 4Mhz Clock to 2000 so that we get 2Khz Output
      MOV AL, 00110111b       ; 00 [counter 0], 11[ 2 byte data], 010 [ with mod 3], 1 [BCD]
        OUT 7Eh, AL           ; configure the counter 0 of 8253

        ; --> Pulse Wave  :: MOD 2 of 8253
        MOV AL, 01110111b     ; 01 [counter 1], 11[ 2 byte data], 010 [ with mod 3], 1 [BCD]
        OUT 7Eh, AL           ; configure the counter 1 of 8253

        MOV AL, 00h
        OUT 78h, AL           ; Send first byte of Counter 0
        MOV AL, 20h
        OUT 78h, AL           ; Send second byte of Counter 0
        ;;--> Counter0 : 2000h; send BCD 2000 to counter 0

        MOV AL, 00h
        OUT 7Ah, AL           ; Send first byte of Counter 1
        MOV AL, 5h
        OUT 7Ah, AL           ; Send second byte of Counter 1
        ;;--> Counter1 : 5h ; send BCD 5 to counter 1
RET
SETUP_8253 ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SETUP_8259 PROC NEAR

MOV AL, 00010011b             ; ICW1 : Edge Triggered - 1 Master - ICW4 Needed
OUT 10h, AL                   ; SET InputControlWord1

MOV AL, 40h                   ; ICW2 : 40h
OUT 12h, AL                   ; SET InputControlWord2

MOV AL, 03h                   ; AEOI(Automatic Interrupt) = 1, 8086= 1
OUT 12h, AL                   ; SET InputControlWord4

RET
SETUP_8259 ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
INSERT_ISR_INTO_VECTOR_TABLE PROC NEAR
  XOR AX, AX
  MOV ES, AX
  MOV AL, 40H
  MOV AH, 4
  MUL AH
  MOV BX, AX
  LEA AX, INC_AX_PER_15_SEC
  MOV WORD PTR ES:[BX], AX
  MOV AX, CS
  MOV WORD PTR ES:[BX+2], AX
RET
```

```
INSERT_ISR_INTO_VECTOR_TABLE ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; Interrupt Routine
INC_AX_PER_15_SEC PROC FAR
   INC AX
IRET
INC_AX_PER_15_SEC ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

CODE    ENDS
        END START
```

DATA SHEETS

PIT Control Register:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC—Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (see Read Operations)

M—Mode

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW—Read/Write

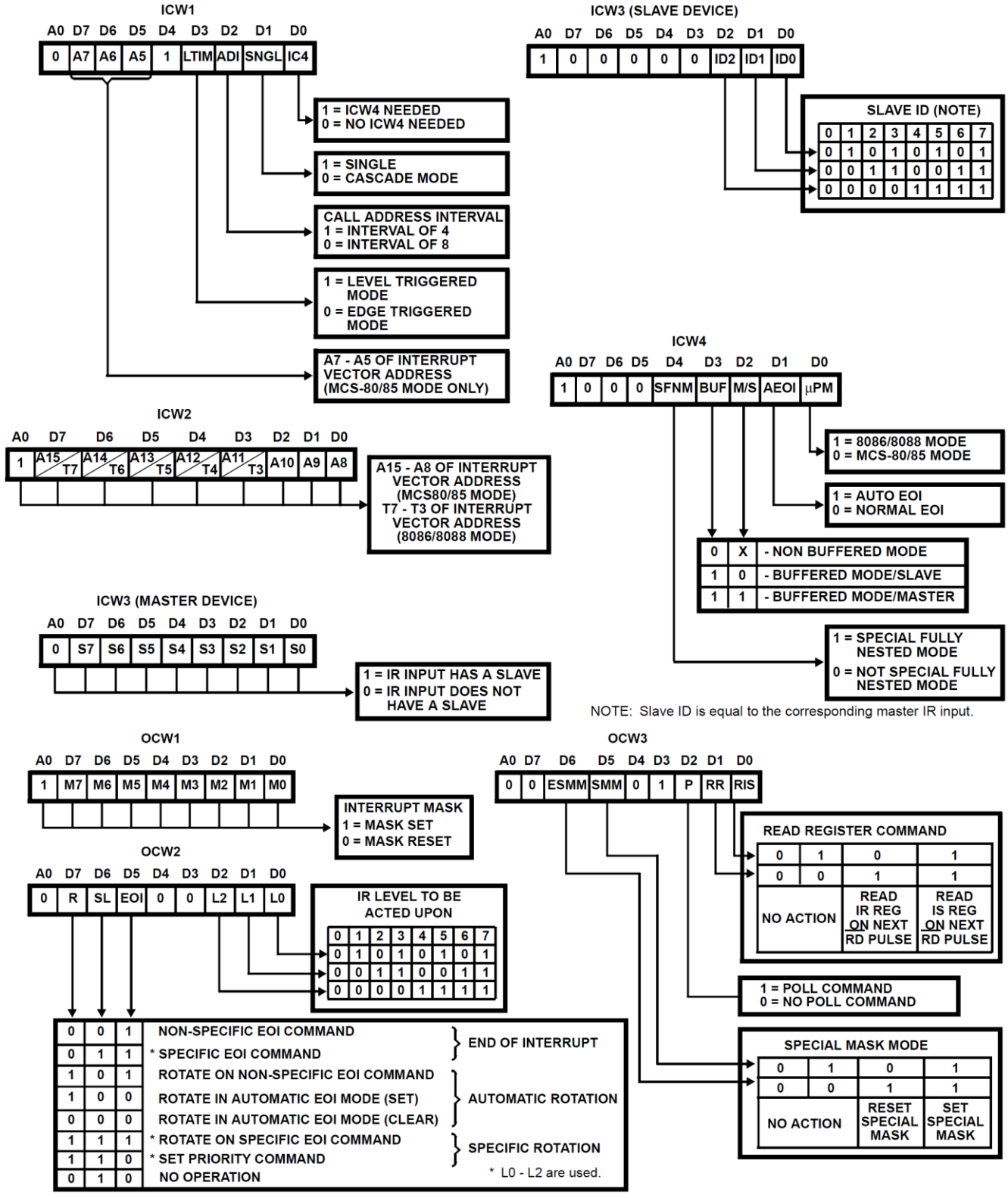
RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only
1	0	Read/Write most significant byte only
1	1	Read/Write least significant byte first, then most significant byte

BCD

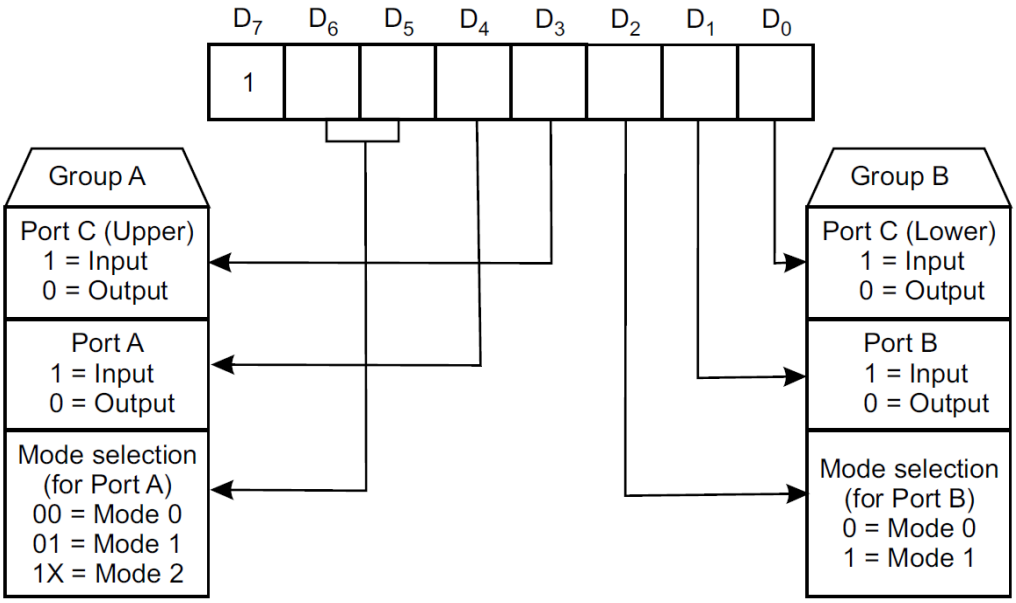
0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

PIC Command Words:



PPI:


D7 D6 D5 D4 D3 D2 D1 D0
1
Group A
Port C (Upper)
1 = Input
0 = Output
Port A
1 = Input
0 = Output
Mode selection
(for Port A)
00 = Mode 0
01 = Mode 1
1X = Mode 2
Group B
Port C (Lower)
1 = Input
0 = Output
Port B
1 = Input
0 = Output
Mode selection
(for Port B)
0 = Mode 0
1 = Mode 1