The K Project

LSE Team

EPITA

mars 10, 2017

I/O Ports

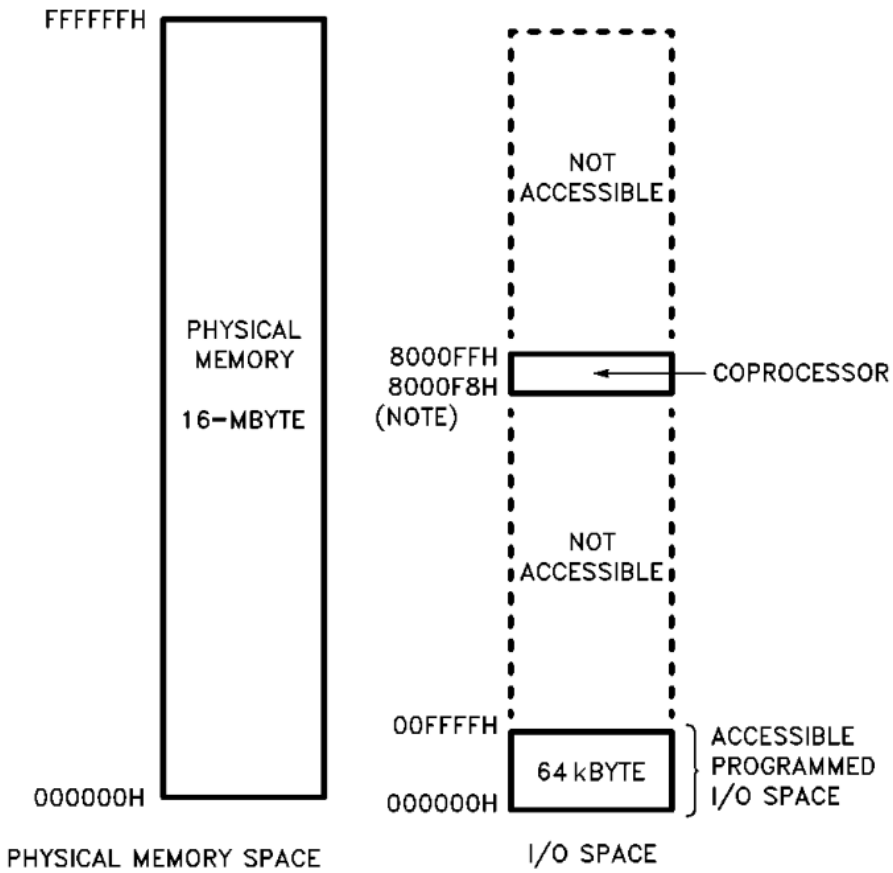


Figure:

I/O Ports


```
asm volatile("outb %0, %1\n\t"
             : /* No output */
             : "a" (val), "d" (port));

asm volatile("inb %1, %0\n\t"
             : "=&a" (res)
             : "d" (port));
```

8259 Overview

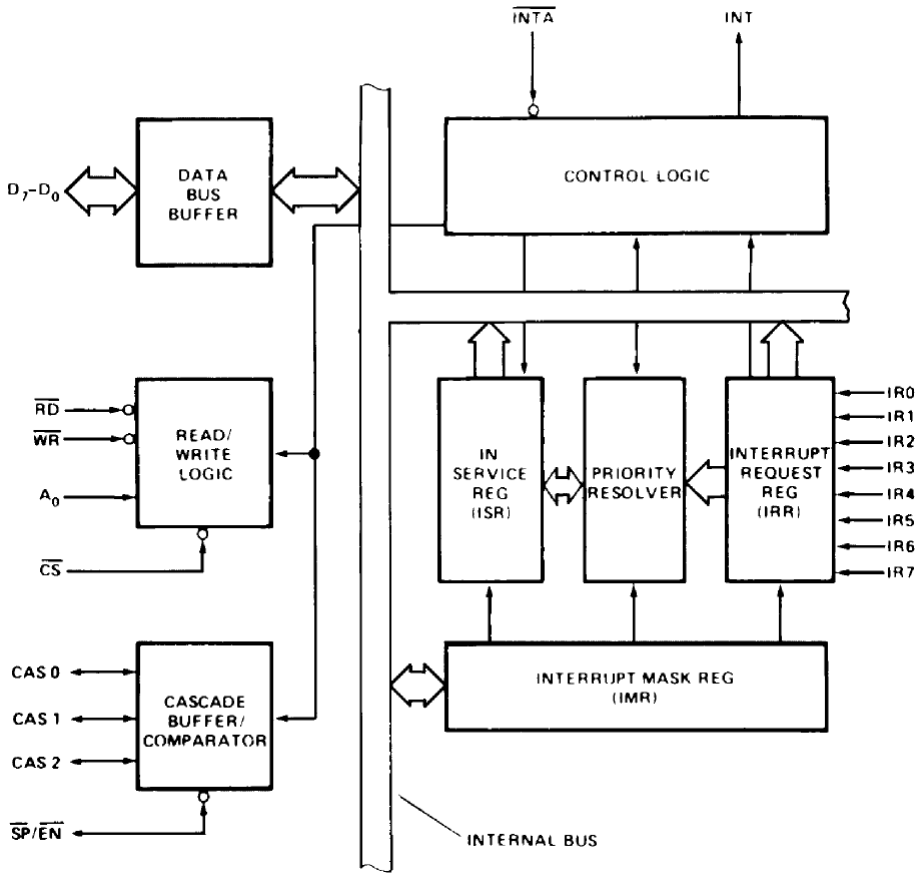


Figure:

PIC Wiring

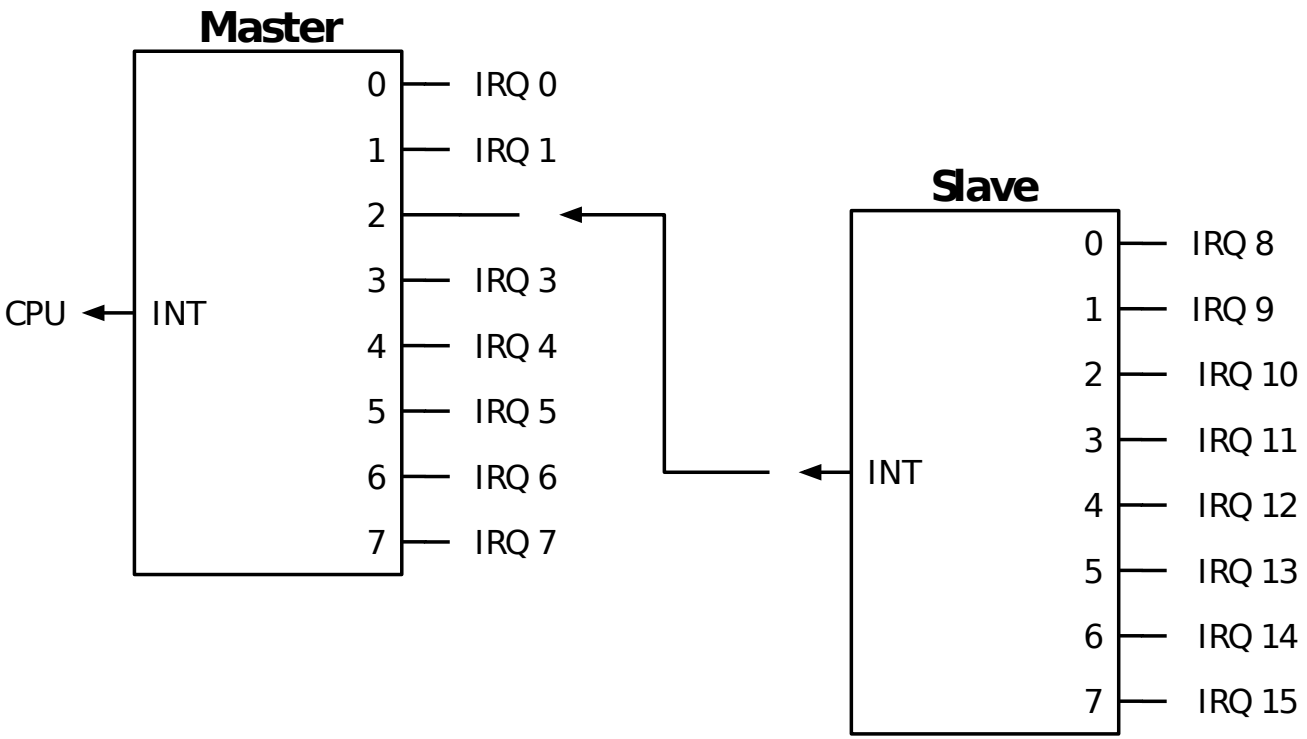


Figure:

Typical wiring of the PIC (Master)

- IRQ0 - PIT
- IRQ1 - Keyboard
- IRQ2 - Not assigned in PC/XT; cascaded to slave 8256
- IRQ3 - UART (COM2 and COM4)
- IRQ4 - UART (COM1 and COM3)
- IRQ5 - Hard disk in PC/XT; Parallel port LPT2 in PC/AT
- IRQ6 - Floppy disk controller
- IRQ7 - Parallel port LPT1

Typical wiring of the PIC (Slave)

- IRQ8 - RTC
- IRQ9 -
- IRQ10 -
- IRQ11 -
- IRQ12 - PS/2 mouse controller
- IRQ13 - Math coprocessor
- IRQ14 - Hard disk controller 1
- IRQ15 - Hard disk controller 2

Interrupt acknowledge (8086/8088)

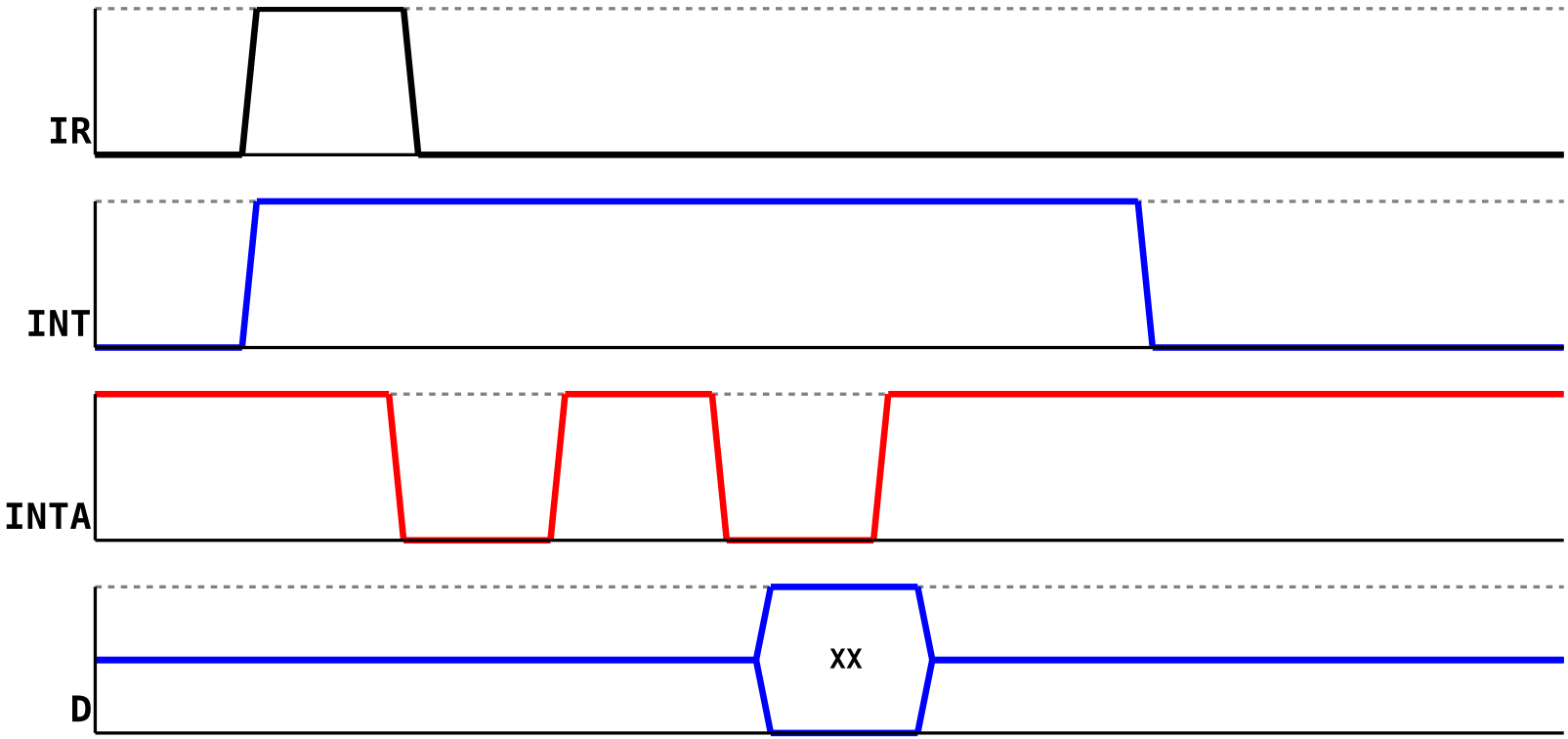


Figure:

PIC ports

- `0x20`, the master PIC's port A
- `0x21`, the master PIC's port B
- `0xA0`, the slave PIC's port A
- `0xA1`, the slave PIC's port B

PIC Initialization

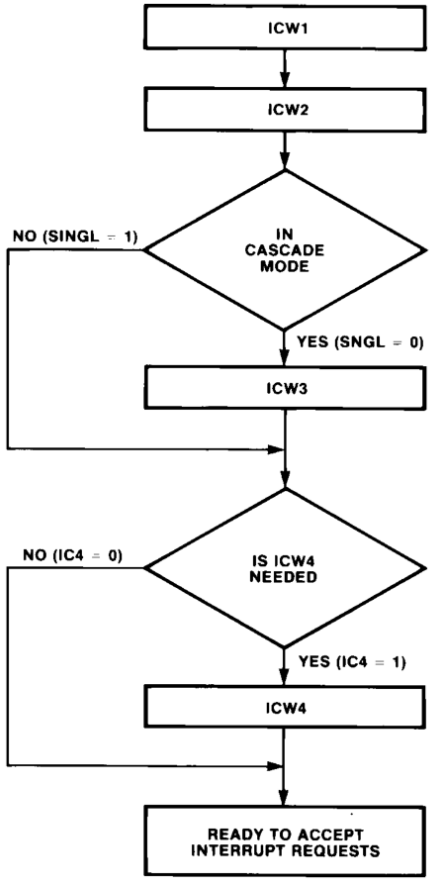


Figure:

ICW1

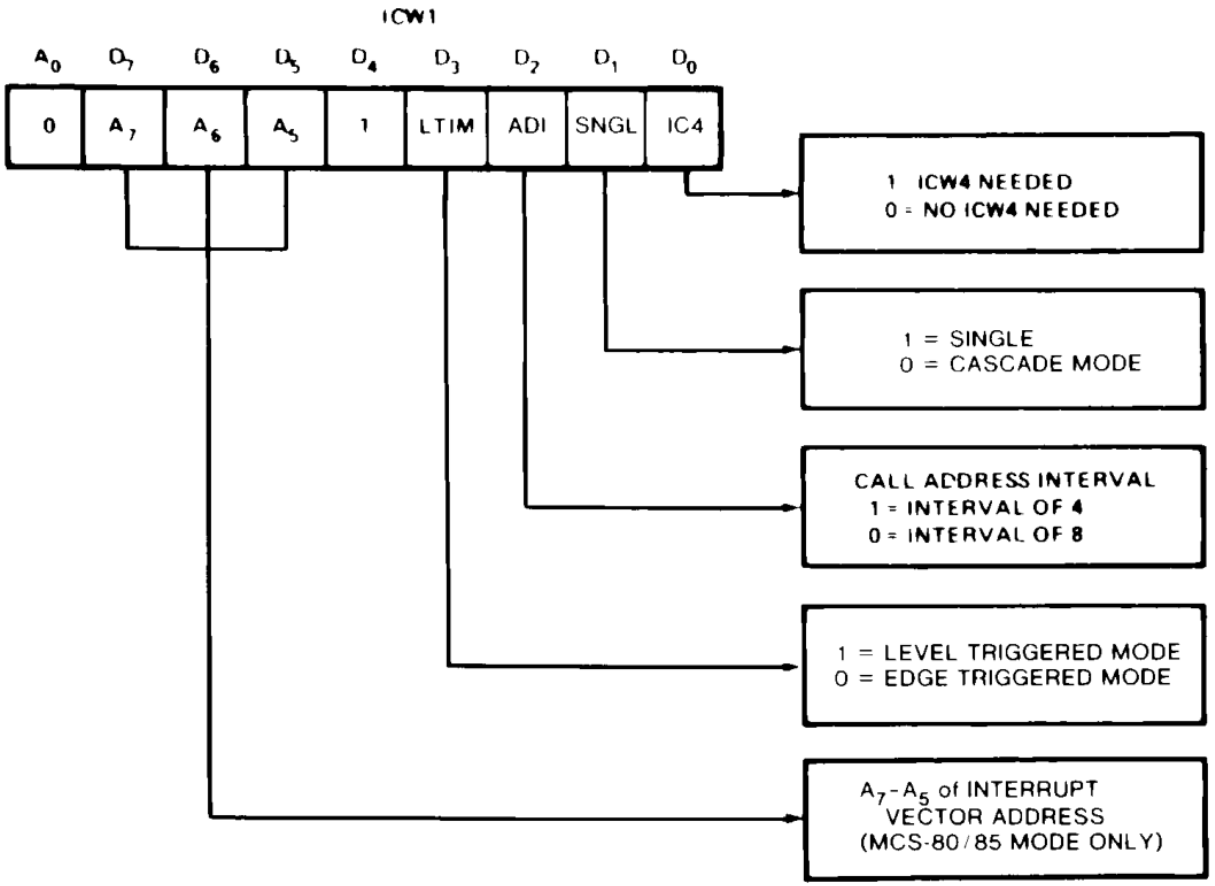


Figure:

ICW2

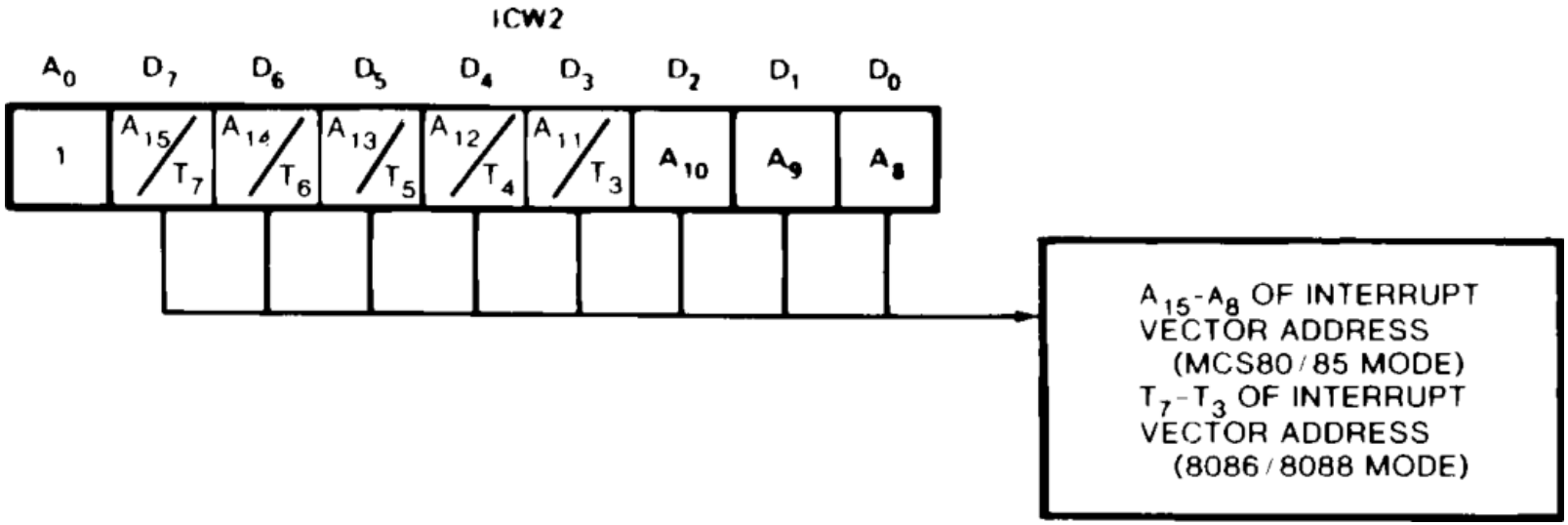


Figure:

ICW3

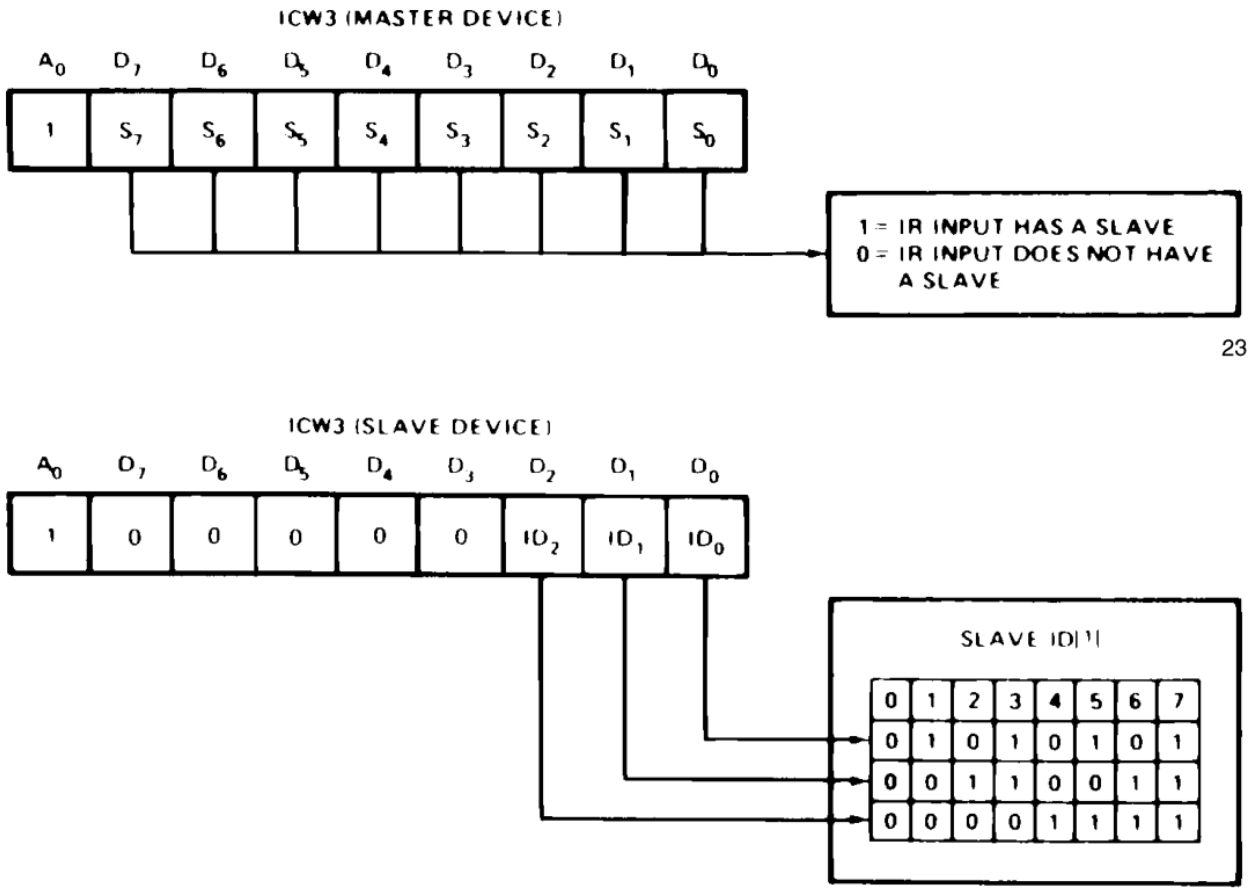


Figure:

ICW4

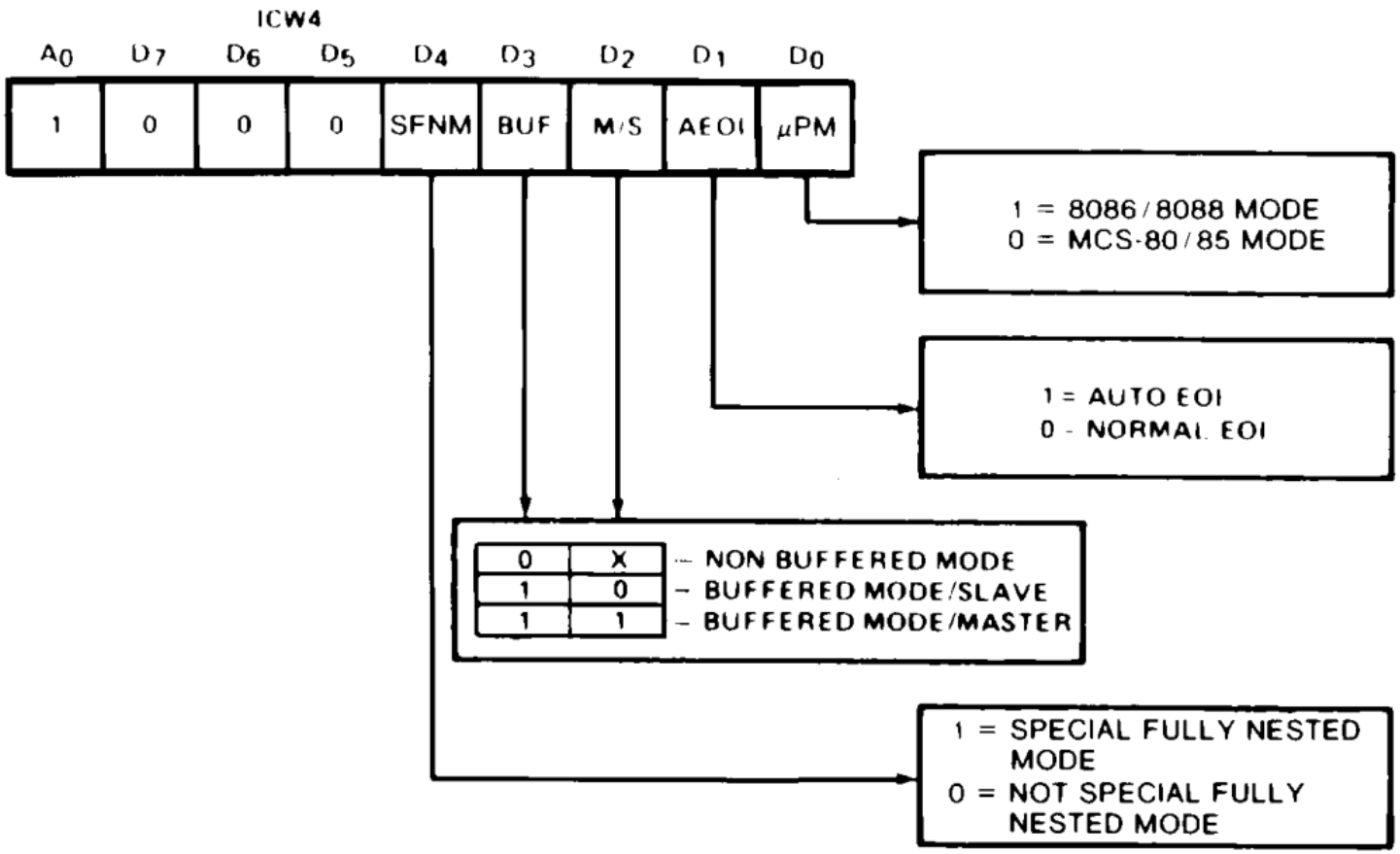


Figure:

OCW1

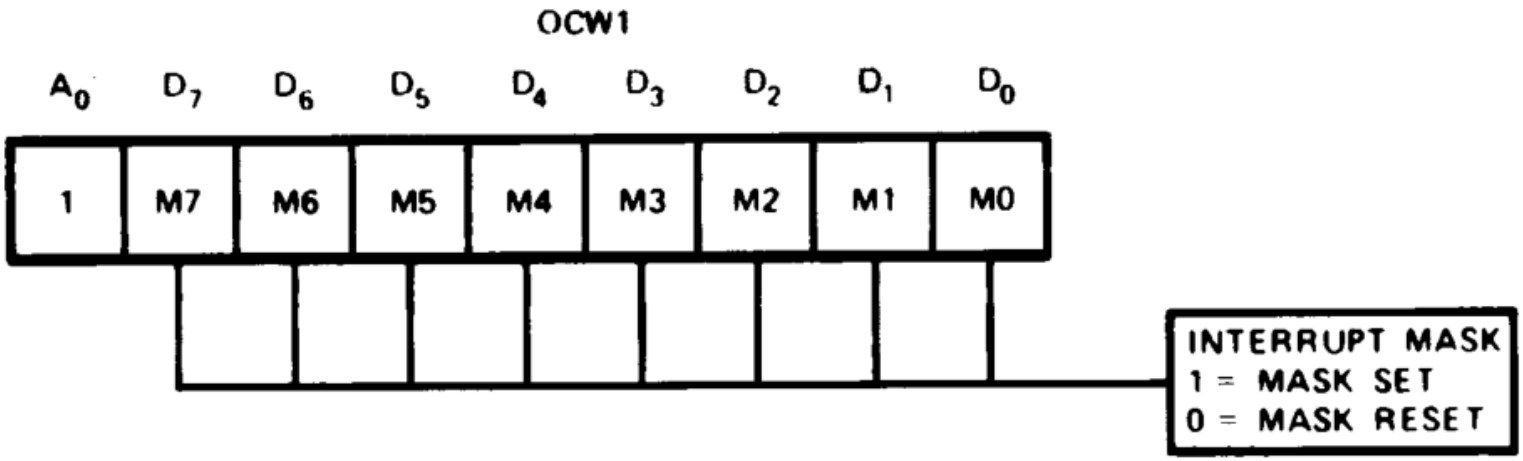


Figure:

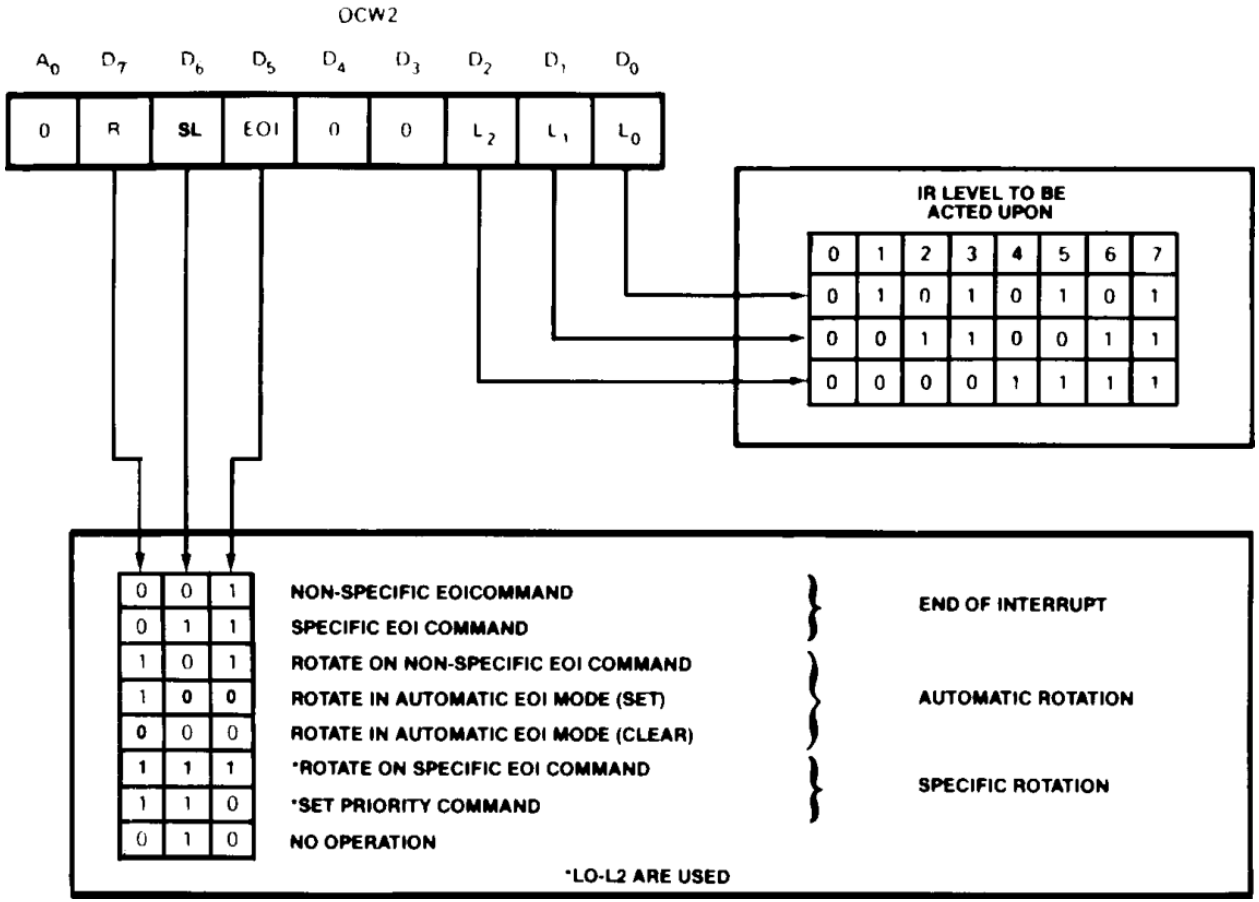

OCW2
A0 D7 D6 D5 D4 D3 D2 D1 D0
0 R SL EOI 0 0 L2 L1 L0
IR LEVEL TO BE ACTED UPON
0 1 2 3 4 5 6 7
0 1 0 1 0 1 0 1
0 0 1 1 0 0 1 1
0 0 0 0 1 1 1 1
0 0 1 NON-SPECIFIC EOICOMMAND
0 1 1 SPECIFIC EOI COMMAND
1 0 1 ROTATE ON NON-SPECIFIC EOI COMMAND
1 0 0 ROTATE IN AUTOMATIC EOI MODE (SET)
0 0 0 ROTATE IN AUTOMATIC EOI MODE (CLEAR)
1 1 1 *ROTATE ON SPECIFIC EOI COMMAND
1 1 0 *SET PRIORITY COMMAND
0 1 0 NO OPERATION
END OF INTERRUPT
AUTOMATIC ROTATION
SPECIFIC ROTATION
*L0-L2 ARE USED

Figure:

8042 Controller

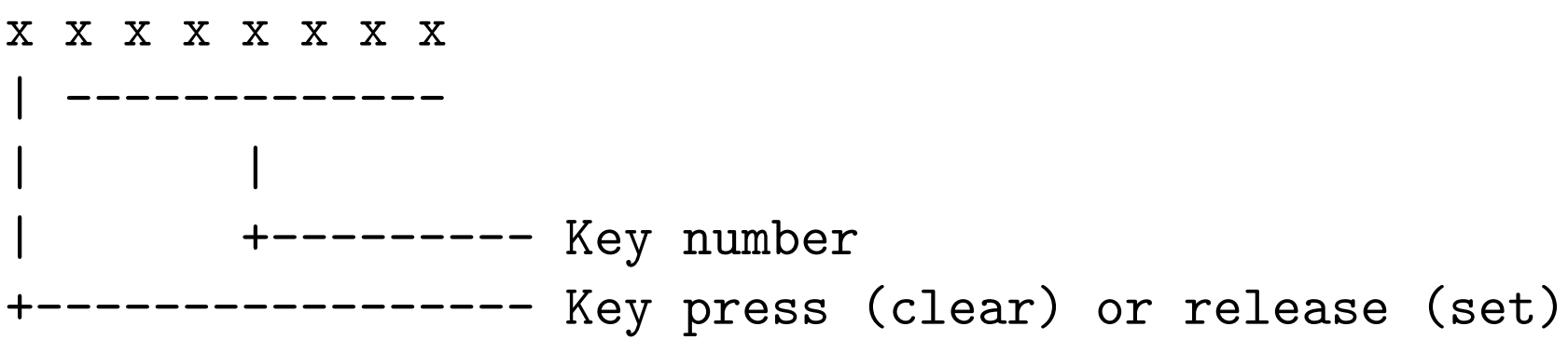
- 0x60: I/0 buffer
- 0x64: Status register

8042 Status Register

BIT	BIT DESCRIPTION	FUNCTION
0	Output Buffer Full	0: Output Buffer Empty 1: Output Buffer Full
1	Input Buffer Full	0: Input Buffer Empty 1: Input Buffer Full
2	System Flag	This bit may be set to 0 or 1 by writing to the system flag bit in the command byte of the keyboard controller. It is set to 0 after a power-on reset
3	Command/data	0: Data Byte 1: Command Byte
4	Inhibit Switch	0: Keyboard is Inhibited 1: Keyboard is Not Inhibited
5	Transmit Time Out	0: No Transmit Time Out Error 1: Transmit Time Out Error
6	Receive Time Out	0: No Receive Time Out Error 1: Receive Time Out Error
7	Parity Error	0: Odd Parity (No Error) 1: Even Parity (Error)

Figure:

Keyboard scancode



8254 Overview

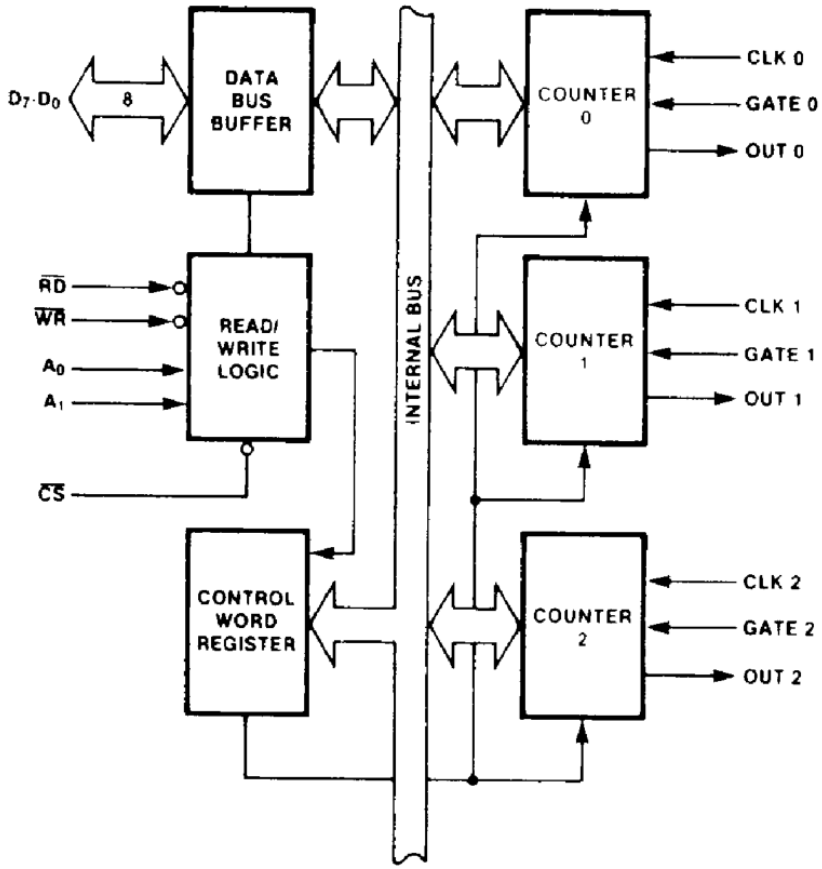


Figure:

Programmable Interval Timer

- Counter 0: fire an interrupt at a user-defined frequency.
- Counter 1: historically used in order to periodically refresh the RAM, but it not used anymore.
- Counter 2: linked with the PC speaker, so you can use it in order to generate sound

PIT Registers

- 0x40 : Counter 0
- 0x41 : Counter 1
- 0x42 : Counter 2
- 0x43 : Control Register

PIT Configuration

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)

Figure:

PIT Operation Modes

- Mode 0: Interrupt on terminal count
- Mode 1: hardware retriggerable one-shot
- Mode 2: rate generator
- Mode 3: square generator
- Mode 4: Software Triggered Strobe
- Mode 5: Hardware Triggered Strobe

PIT - Mode 2

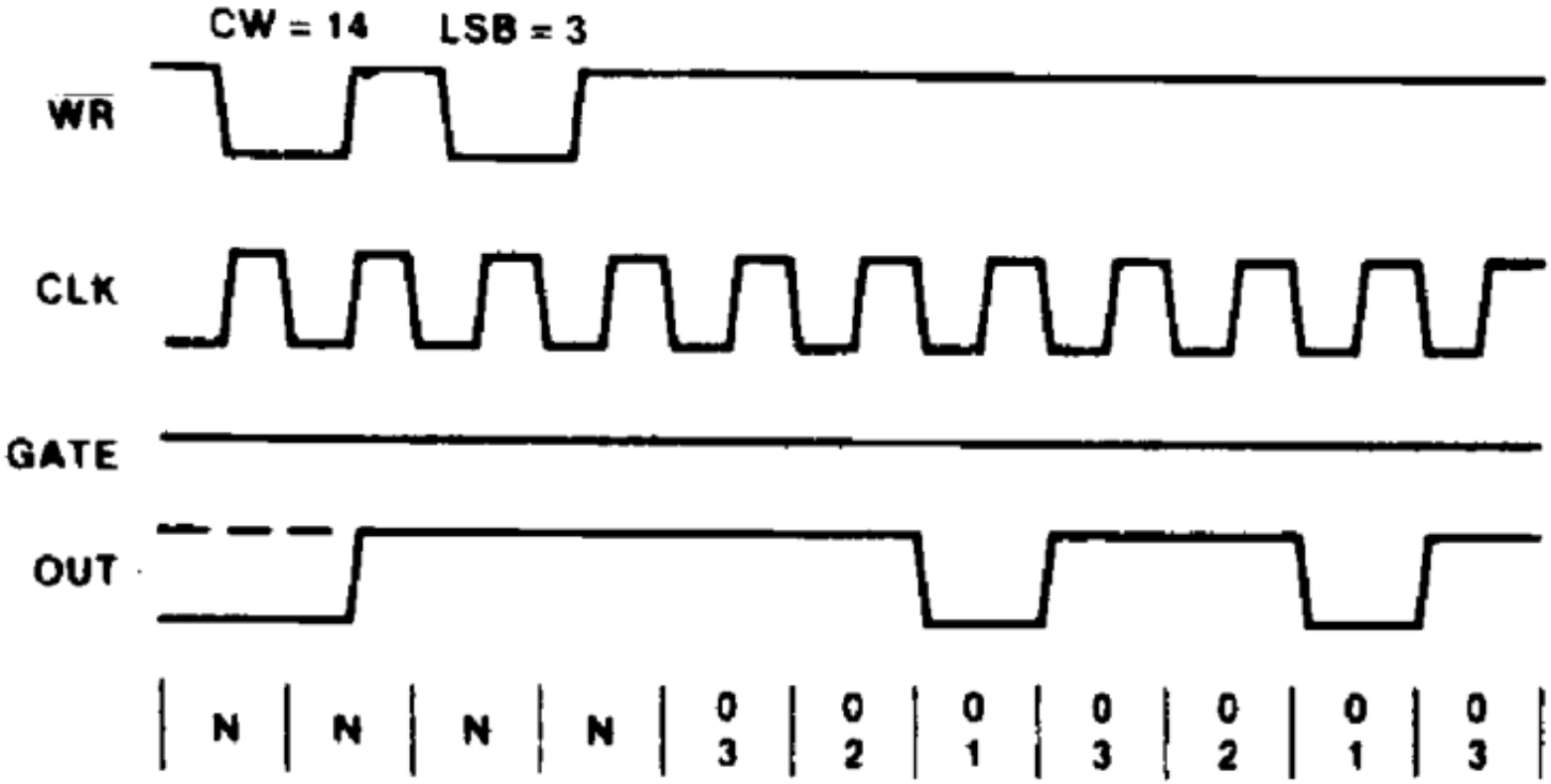


Figure:

PIT Programming


```
unsigned long gettick(void);
```

- Counter 0
- Mode 2
- Interrupt rate : 100 Hz (Input clock frequency = 1193182 Hz)

Summary

- Build IDT
- Write context saving/restoring in assembly code
- Implement exceptions and interrupt wrappers
- Load IDT
- Initialize PIC
  - send ICWs to both master and slave
  - mask all interrupts
- Set keyboard interrupt handler
- Initialize PIT
  - Send CW
  - Set PIT interrupt handler

Contact

- #k (irc.rezosup.org)
- epita.cours.k
- k[at]lse.epita.fr
- naam[at]lse.epita.fr
- nurelin[at]lse.epita.fr