

Tutorial: I/O Controllers

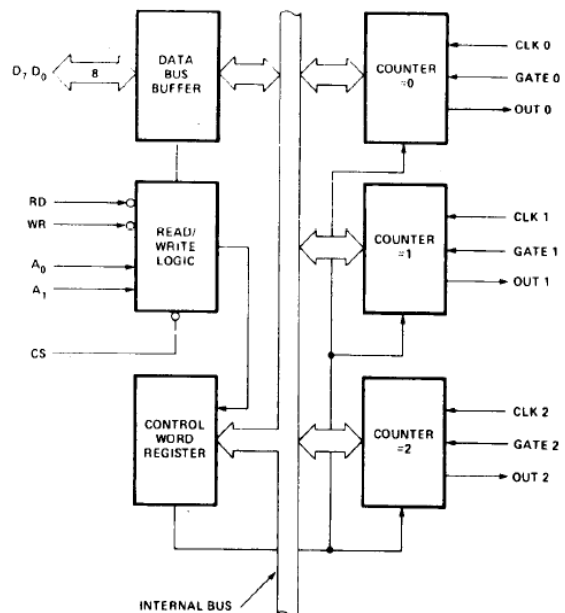
- In this tutorial, we'll look at some example I/O components and discuss how they software can program them
- Topics covered:
 - Intel 8253 PIT
 - Port Numbers and \overline{CS} (Chip Select)
 - Linux 8253 Driver Excerpts
 - Intel 8259 PIC

I/O Example (1)

8253 Programmable Interval Timer (PIT)

- Connected to 2 address lines
 - Supports up to 4 registers
- In x86 PC's, the base address of the PIT is 0x40 (port mapped address)
- Counter 0: 0 1 0 0 0 0 0 0 = 0x40
- Counter 1: 0 1 0 0 0 0 0 1 = 0x41
- Counter 2: 0 1 0 0 0 0 1 0 = 0x42
- CWR: 0 1 0 0 0 0 1 1 = 0x43

Example: out value, 0x40 ; resets the value of counter 0



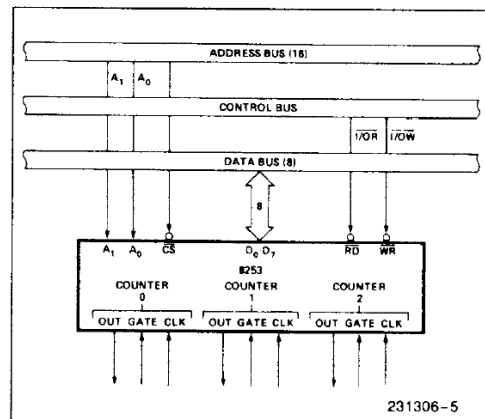
Control Word Register

Example: When A0,A1 are 11, then the 8253 accepts information from the data bus and stores it in the Control Word Register

CS	RD	WR	A ₁	A ₀	
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation 3-State
1	X	X	X	X	Disable 3-State
0	1	1	X	X	No-Operation 3-State

8253 System Interface

- 8253's A0, A1 inputs connect to the A0, A1 address bus signals of the CPU
- $\overline{\text{CS}}$ can be derived either directly from the address bus, or it can be connected to the output of a decoder (such as the 8205)



Linux 8253 Driver Code Excerpts

```

28 /*
29  * Initialize the PIT timer.
30  *
31  * This is also called after resume to bring the PIT into operation again
32  */
33 static void init_pit_timer(enum clock_event_mode mode,
34                          struct clock_event_device *evt)
35 {
36     unsigned long flags;
37
38     spin_lock_irqsave(&i8253_lock, flags);
39
40     switch(mode) {
41     case CLOCK_EVT_PERIODIC:
42         /* binary, mode 2, LSB/MSB, ch 0 */
43         outb_p(0x34, PIT_MODE);
44         udelay(10);
45         outb_p(LATCH & 0xff, PIT_CH0); /* LSB */
46         udelay(10);
47         outb(LATCH >> 8, PIT_CH0); /* MSB */
48         break;
49
50     case CLOCK_EVT_ONESHOT:
51     case CLOCK_EVT_SHUTDOWN:
52         /* One shot setup */
53         outb_p(0x38, PIT_MODE);
54         udelay(10);
55         break;
56     }
57     spin_unlock_irqrestore(&i8253_lock, flags);
58 }

```

```

1 /*
2  * arch/i386/mach-generic/io_ports.h
3  *
4  * Machine specific IO port address definition
5  * Written by Osamu Tomita <tomita@cinet.co.jp>
6  */
7 #ifndef MACH_IO_PORTS_H
8 #define MACH_IO_PORTS_H
9
10 /* i8253A PIT registers */
11 #define PIT_MODE 0x43
12 #define PIT_CH0 0x40
13 #define PIT_CH2 0x42
14
15 /* i8259A PIC registers */
16 #define PIC_MASTER_CMD 0x20
17 #define PIC_MASTER_IMR 0x21
18 #define PIC_MASTER_ISR PIC_MASTER_CMD
19 #define PIC_MASTER_POLL PIC_MASTER_ISR
20 #define PIC_MASTER_OCW3 PIC_MASTER_ISR
21 #define PIC_SLAVE_CMD 0xa0
22 #define PIC_SLAVE_IMR 0xa1
23
24 /* i8259A PIC related value */
25 #define PIC_CASCADE_IR 2
26 #define MASTER_ICW4_DEFAULT 0x01
27 #define SLAVE_ICW4_DEFAULT 0x01
28 #define PIC_ICW4_AEOI 2
29
30 #endif /* !MACH_IO_PORTS_H */
31
36 /* LATCH is used in the interval timer and ftape setup. */
37 #define LATCH ((CLOCK_TICK_RATE + HZ/2) / HZ) /* For divider */

```

Programming the 8253 (1)

Control Word Format

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC1	SC0	RL1	RL0	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	Illegal

Control Word: 0x34 = 0011 0100

RL—READ/LOAD:

RL1	RL0	
0	0	Counter Latching operation (see READ/WRITE Procedure Section).
1	0	Read/Load most significant byte only.
0	1	Read/Load least significant byte only.
1	1	Read/Load least significant byte first, then most significant byte.

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

BCD:

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

Programming the 8253 (2)

MODE Control Word Counter n	
LSB	Counter Register byte Counter n
MSB	Counter Register byte Counter n

		A1	A0
No. 1	MODE Control Word Counter 0	1	1
No. 2	MODE Control Word Counter 1	1	1
No. 3	MODE Control Word Counter 2	1	1
No. 4	LSB Count Register Byte Counter 1	0	1
No. 5	MSB Count Register Byte Counter 1	0	1
No. 6	LSB Count Register Byte Counter 2	1	0
No. 7	MSB Count Register Byte Counter 2	1	0
No. 8	LSB Count Register Byte Counter 0	0	0
No. 9	MSB Count Register Byte Counter 0	0	0

I/O Example (2)

8259A Programmable Interrupt Controller (PIC)

- Connected to only 1 address line
 - Supports up to 2 registers
- In x86 PC's, the base address of the PIC is 0x20
 - 0010 0000 = 0x20
 - 0010 0001 = 0x21
- No real one-to-one mapping between addresses and PIC registers in this case
 - Used to decipher and differentiate between various command words
 - One of the I/O registers (ISR, IRR, IMR) can be addressed by a sequence of command words (i.e., send register address in data command word, followed by another read/write command to read/write register)

