

The ISA Bus - Supplemental Course Notes

The Industry Standard Architecture (ISA) is the most prolific bus in IBM PC's and compatibles. This almost 20-year-old architecture prescribes the peripheral-to-CPU (central processing unit) protocol. Understanding this architecture serves the building of dedicated ISA-bus based cards for PC-based control and data acquisition.

The ISA Bus

A "bus" is physically a collection of lines that run between two points. Typically these points are the computer's brain, the central processing unit (CPU) and a peripheral. A peripheral is a hardware device attached to a computer, familiar examples being devices printers, modems and joysticks. For control applications, peripherals include actuators and sensors; relays, motors, solenoids and switches are examples of the former and strain gages, pressure transducers, thermocouples are examples of the latter. For many applications, speed is important and that bus acts to "carry" data between the CPU and peripheral using dedicated lines.

The ISA bus, found on PC's, is quite typical of most computer and microcomputer buses. Thus, although ISA is being phased out by PCI (peripheral component interconnect) bus in latest PC's, it still serves for a solid and basic understanding of any computer bus. A bus is typically found on a computer's motherboard (see Figure 1) or as headers off an embedded microcomputer.

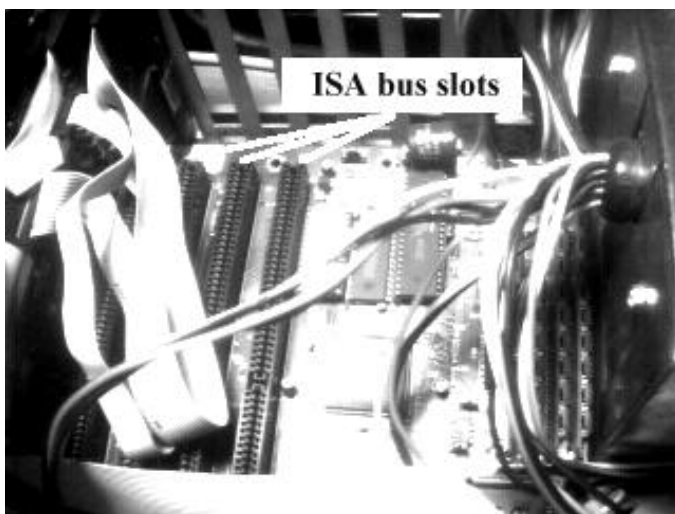


Figure 1 - With the PC cover off, the ISA slots are visible

SOLDER	ISA SLOT		COMPONENT
B SIDE	PC REAR	A SIDE	
1	GND	~IO CHCK	1
2	RESET DRV	D7	2
3	+5 V	D6	3
4	IRQ2	D5	4
5	-5 V	D4	5
6	DRQ2	D3	6
7	-12 V	D2	7
8	RESERVED	D1	8
9	+12 V	D0	9
10	GND	I/O CH RDY	10
11	~MEMW	AEN	11
12	~MEMR	A19	12
13	~IOW	A18	13
14	~IOR	A17	14
15	~DACK3	A16	15
16	DRQ3	A15	16
17	~DACK1	A14	17
18	DRQ1	A13	18
19	~DACK0	A12	19
20	CLOCK	A11	20
21	IRQ7	A10	21
22	IRQ8	A9	22
23	IRQ5	A8	23
24	IRQ4	A7	24
25	IRQ3	A6	25
26	~DACK2	A5	26
27	T/C	A4	27
28	ALE	A3	28
29	+5 V	A2	29
30	OSC	A1	30
31	GND	A0	31

Figure 2 - The ISA bus pin out and labels

ISA cards like modem and sounds cards physically install into an ISA slot much like a tongue-and-groove joint. Figure 2 is the ISA bus pin out which illustrate 31 different pins on both sides of

the slot. The key features of any bus are lines to handle memory and data and on the ISA bus these are labeled A12-A31 and A2-A9 respectively. You'll note that ISA is an 8-bit bus, since there are only eight data lines (D0-D7). The remaining lines exist for other features like interrupts (IRQ's), voltage lines (+12, -12, +5, -5 and GND), timing (OSC) and lines to control data direction of data (i.e. reading to and writing from memory).

ISA Bus Communications

Recalling Turbo C, `outportb(address, byte)` and `byte = inportb(address)`, 8-bit data is written to and read from a card installed in the ISA slot. These high-level C functions implement low-level bus handling by port-mapped I/O, that is, read/writes to a 16-bit value `address`, accesses the card. IBM technical manuals give a detailed I/O map. Addresses ranging from 300-31F Hex (768-799 Decimal) are specifically reserved for developers of prototype cards.

Just like every building has an address that uniquely distinguishes it from other buildings, each device attached to the PC has a memory address. For example, read/writes to memory address 378 Hex allows one to communicate to a PC-attached printer. Likewise, accessing a PC hard disk is achieved by memory address 320 Hex. Custom designing a card thus requires assigning an address that is different from other devices already attached to the PC. Attempts to use an address that is already taken by another card will confuse the CPU and lead to *address conflicts*, also known as *bus contention*.

Third-party card developers often use 300 Hex for their cards (e.g. sound cards). Thus to avoid bus contention, the card designer should provide alternative card addresses. This is the reason why many ISA cards have a jumper over a double-row header (see Figure 3).

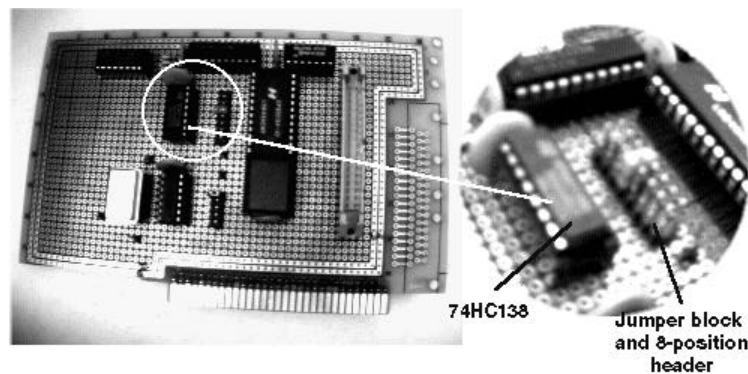


Figure 3 – Unique card address assignment is achieved by moving a jumper block

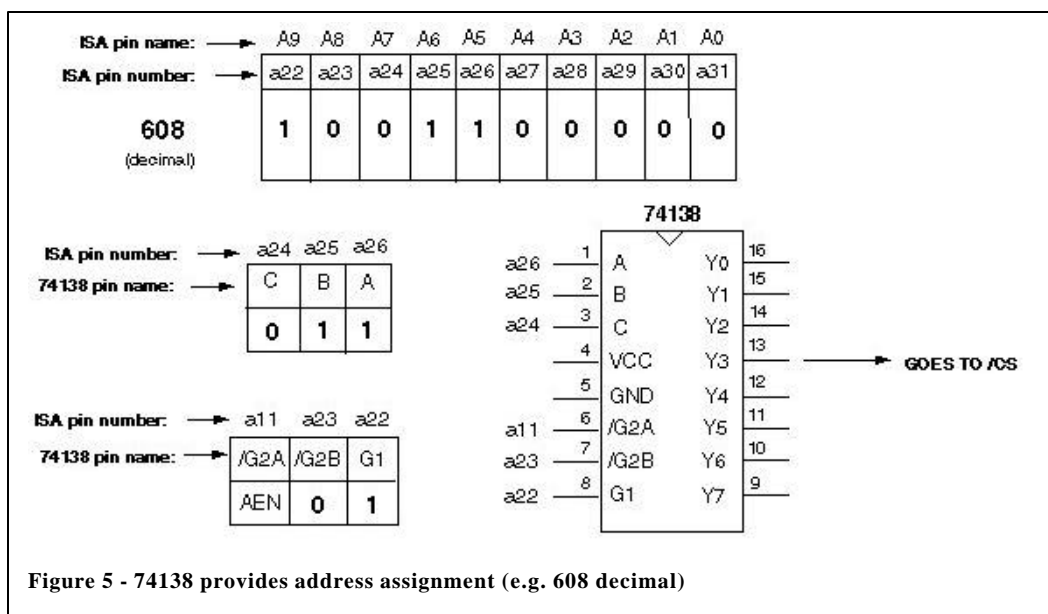
Address Decoding

Circuitry in addition to the jumper block is required for memory address assignment. From Figure 2, A0-A9 provides 10 bits for address assignment, or $2^{10} = 1024$ port addresses. As mentioned above, the IBM I/O map defines only some of these addresses (e.g. 300 Hex) for third party or custom-designed cards.

One often-free address is 260 Hex (608 decimal). An `outportb(608, byte)` will thus write `byte` to address 608 decimal. Figures 4 and 5 suggest how a 3-to-8 multiplexer (74138) can provide the physical connection to the ISA-installed card. The procedure follows.

G1	G2	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	L	H	H
H	L	H	H	H	H	H	H	H	H	H	H	L

Figure 4 – 74138 Truth Table



Procedure:

1. `outportb(608, byte)` or `byte = inportb(608)` will place 1001100000 binary on the bus lines pins **a22** to **a31**. This will also force pin **a11** (AEN) low. AEN stands for "address enable".
2. The 74138 truth table (Figure 4) defines that 74138's Y3 will be low if its pins A, B and C are high, high and low respectively and /G2A, /G2B and G1 are low, low and high respectively. These are all achieved using `outportb` or `inportb` with 608 decimal.

Typically an integrated chip (IC) that readily interfaces to microprocessors (e.g. DAC, ADC, 8255, timer chips) has a chip select (/CS) pin. Thus connecting Y3 above to /CS will activate that chip. A jumper provides such a connection. The 74138 has eight possible outputs (Y0-Y7) and hence provides the possibility of assigning one of eight possible addresses.

Lastly, writing or reading is physically achieved by bringing ISA pins /IOW (**a13**: I/O write) or /IOR (**a14**: I/O read) low and `outportb (address, byte)` and `byte = inportb(address)` achieve this respectively.