AUTOCONFIG™

THE AUTOCONFIG MECHANISM

The AUTOCONFIG mechanism used for the Zorro III bus is an extension of the original Zorro II configuration mechanism. The main reason for this is that the Zorro II mechanism works so well, there was little need to change anything. The changes are simply support for new hardware features on the Zorro III bus.

Amiga autoconfiguration is surprisingly simple. When an Amiga powers up or resets, every card in the system goes to its unconfigured state. At this point, the most important signals in the system are /CFGINN and /CFGOUTN. As long as a card's /CFGINN line is negated, that card sits quietly and does nothing on the bus (though memory cards should continue to refresh even through reset, and any local board activities that don't concern the bus may take place after /RESET is negated). As part of the unconfigured state, /CFGOUTN is negated by the PIC immediately on reset.

The configuration process begins when a card's /CFGINN line is asserted, either by the backplane, if it's the first slot, or via the configuration chain, if it's a later card. The configuration chain simply ensures that only one unconfigured card will see an asserted /CFGINN at one time. An unconfigured card that sees its /CFGINN line asserted will respond to a block of memory called *configuration space.* In this block, the PIC will assert a set of read-only registers, followed by a set of write-only registers (the read-only registers are also known as AUTOCONFIG ROM). Starting at the base of this block, the read registers describe the device's size, type, and other requirements. The operating system reads these, and based on them, decides what should be written to the board. Some write information is optional, but a board will always be assigned a base address or be told to shut up. The act of writing the final bit of base address, or writing anything to a shutup address, will cause the PIC to assert its /CFGOUTN, enabling the next board in the configuration chain.

The Zorro II configuration space is the 64K memory block $00E8xxxx, which of course is driven with 16-bit Zorro II cycles; all Zorro II cards configure there. The Zorro III configuration space is the 64K memory block beginning at $FF00xxxx, which is always driven with 32-bit Zorro III cycles (PICs need only decode A31-A24 during configuration). A Zorro III PIC can configure in Zorro II or Zorro III configuration space, at the designer's discretion, but not both at once. All read registers physically return only the top 4 bits of data, on D31-D28 for either bus mode. Write registers are written to support nybble, byte, and word registers for the same register, again based on what works best in hardware. This design attempts to map into real hardware as simply as possible. Every AUTOCONFIG register is logically considered to be 8 bits wide; the 8 bits actually being nybbles from two paired addresses.

The register mappings for the two different blocks are shown in *Figure 9-10.* All the bit patterns mentioned in the following sections are logical values. To avoid ambiguity, all registers are referred to by the number of the first register in the pair, since the first pair member is the same for both mapping schemes. In the actual implementation of these registers, all read registers except for the 00 register are physically complemented; eg, the logical value of register 3C is always 0, which means in hardware, the upper nybbles of locations $00E8003C and $00E8003E, or $FF00003C and $FF00013C, both return all 1s.

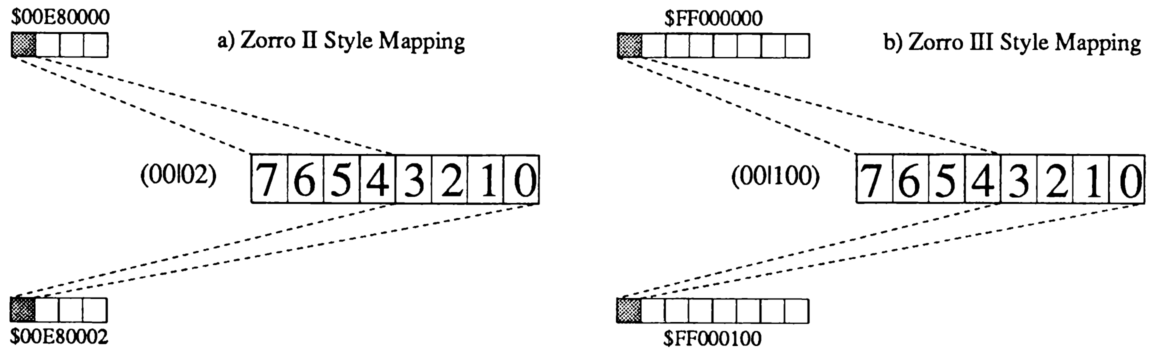

Figure K-14: Configuration Register Mapping

REGISTER BIT ASSIGNMENTS

The actual register assignments are below. Most of the registers are the same as for the Zorro II bus, and are included here for completeness. The Amiga OS software names for these registers in the ExpansionRom or ExpansionControl structures are included.

Reg	ZII	ZIII	Bit	
00	02 (er_Type)	100	7,6	These bits encode the PIC type:

00	Reserved
01	Reserved
10	Zorro III
11	Zorro II

Bit	
5	If this bit is set, the PIC's memory will be linked into the system free pool. The Zorro III register 08 may modify the size of the linked memory.

4 Setting this bit tells the OS to read an autoboot ROM.

3 This bit is set to indicate that the next board is related to this one; often logically separate PICs are physically located on the same card.

2-0 These bits indicate the configuration size of the PIC. This size can be modified for the Zorro III cards by the size extension bit, which is the new meaning of bit 5 in register 08.

Bits	Unextended	Extended
000	8 megabytes	16 megabytes
001	64 kilobytes	32 megabytes
010	128 kilobytes	64 megabytes
011	256 kilobytes	128 megabytes
100	512 kilobytes	256 megabytes
101	1 megabyte	512 megabytes
110	2 megabytes	1 gigabyte
111	4 megabytes	RESERVED

04 06 104 7-0 **(er_Product)** The device's product number, which is completely up to the manufacturer. This is generally unique between different products, to help in identification of system cards, and it must be unique between devices using the automatic driver binding features.

08 0A 108 7 **(er_Flags)** This was originally an indicator to place the card in the 8 megabyte Zorro II space, when set, or anywhere it'll fit, if cleared. Under the Zorro III spec, this is set to indicate that the board is basically a memory device, cleared to indicate that the board is basically an I/O device.

6 This bit is set to indicate that the board can't be shut up by software, cleared to indicate that the board can be shut up.

5 This is the size extension bit. If cleared, the size bits in register 00 mean the same as under Zorro II, if set, the size bits indicate a new size. The most common new Zorro III sizes are the smaller ones; all new sized cards get aligned on their natural boundaries.

4 Reserved, must be 1 for all Zorro III cards.

3-0 These bits indicate a board's sub-size; the amount of memory actually required by a PIC. For memory boards that auto-link, this is the actual amount of memory that will be linked into the system free memory pool. A memory card, with memory starting at the base address, can be automatically sized by the Operating System. This sub-size option is intended to support cards with variable setups without requiring variable physical configuration capability on such cards. It also may greatly simplify a Zorro III design, since 16-megabyte cards and up can be designed with a single latch and comparator for base address matching, while 8 megabyte and smaller PICs require large latch/comparator circuits not available in standard TTL packages.

Bits	Encoding
0000	Logical size matches physical size
0001	Automatically sized by the Operating System
0010	64 kilobytes
0011	128 kilobytes
0100	256 kilobytes
0101	512 kilobytes
0110	1 megabyte
0111	2 megabytes
1000	4 megabytes
1001	6 megabytes
1010	8 megabytes
1011	10 megabytes
1100	12 megabytes
1101	14 megabytes
1110	Reserved
1111	Reserved

For boards that wish to be automatically sized by the operating system, a few rules apply. The memory is sized in 512K increments, and grows from the base address upward. Memory wraps are detected, but the design must insure that its data bus doesn't float when the sizing routine addresses memory locations that aren't physically present on the board; data bus pullups or pulldowns are recommended. This feature is designed to allow boards to be easily upgraded with additional or increased density memoried without the need for memory configuration jumpers.

0C	0E	10C	7-0	Reserved, must be 0.
	(er_Reserved03)			
10	12	110	7-0	Manufacturer's number, high byte.
14	16	114	7-0	Manufacturer's number, low bytes. These are unique, and can only be assigned by Commodore (CATS).
	(er_Manufacturer)			
18	1A	118	7-0	Optional serial number, byte 0 (msb)
1C	1E	11C	7-0	Optional serial number, byte 1
20	22	120	7-0	Optional serial number, byte 2
24	26	124	7-0	Optional serial number, byte 3 (lsb)
	(er_SerialNumber)			This is for the manufacturer's use and can contain anything at all. The main intent is to allow a manufacturer to uniquely identify individual cards, but it can certainly be used for revision information or other data.

28	2A 128 7-0	Optional ROM vector, high byte.
2C	2E 12C 7-0 **(er_InitDiagVec)**	Optional ROM vector, low byte. If the ROM address valid bit (bit 4 of register (00|02)) is set, these two registers provide the sixteen bit offset from the board's base at which the start of the ROM code is located. If the ROM address valid bit is cleared, these registers are ignored.
30	32 130 7-0 **(er_Reserved0c)**	Reserved, must be 0. Unsupported base register reset register under Zorro II[6]
34	36 134 7-0 **(er_Reserved0d)**	Reserved, must be 0.
38	3A 138 7-0 **(er_Reserved0e)**	Reserved, must be 0.
3C	3E 13C 7-0 **(er_Reserved0f)**	Reserved, must be 0.
40	42 140 7-0 **(ec_Interrupt)**	Reserved, must be 0. Unsupported control state register under Zorro II[7]
44	46 144 7-0	High order base address register, write only.
48	4A 148 7-0 **(ec_Z3_HighByte)** **(ec_BaseAddress)**	Low order base address register, write only. The high order register takes bits 31-24 of the board's configured address, the low order register takes bits 23-16. For Zorro III boards configured in the Zorro II space, the configuration address is written both nybble and byte wide, with the ordering:

Reg	Nybble	Byte
46	A27-A24	N/A
44	A31-A28	A31-A24
4A	A19-A16	N/A
48	**A23-A20**	**A23-A16**

[6] The original Zorro specifications called for a few registers, like these, that remained active after configuration. Support for this is impossible, since the configuration registers generally disappear when a board is configured, and absolutely must move out of the $00E8xxxx space. So since these couldn't really be implemented in hardware, system software has never supported them. They're included here for historical purposes.

[7] IBID

Note that writing to register 48 actually configures the board for both Zorro II and Zorro III boards in the Zorro II configuration block. For Zorro III PICs in the Zorro III configuration block, the action is slightly different. The software will actually write the configuration as byte and word wide accesses:

Reg	Byte	Word
48	A23-A16	N/A
44	A31-A24	A31-A16

The actual configuration takes place when register 44 is written, thus supporting any physical size of configuration register.

4C	4E	14C	7-0	Shut up register, write only. Anything written to 4C will cause a board that supports shut-up to completely disappear until the next reset.
	(ec_Shutup)			
50	52	150	7-0	Reserved, must be 0.
54	56	154	7-0	Reserved, must be 0.
58	5A	158	7-0	Reserved, must be 0.
5C	5E	15C	7-0	Reserved, must be 0.
60	62	160	7-0	Reserved, must be 0.
64	66	164	7-0	Reserved, must be 0.
68	6A	168	7-0	Reserved, must be 0.
6C	6E	16C	7-0	Reserved, must be 0.
70	72	170	7-0	Reserved, must be 0.
74	76	174	7-0	Reserved, must be 0.
78	7A	178	7-0	Reserved, must be 0.
7C	7E	17C	7-0	Reserved, must be 0.