How to Perform Plug and Play Initiation Key with PCnet-ISA II

Application Note

TITLE: How to perform Plug and Play Initiation Key with PCnet-ISA II.

KEYWORDS: Am79C961A, PnP, plugplay, auto-config, wakeup, software, ISA.

PRODUCT: Am79C961A PCnet-ISA II Jumperless, Full Duplex Single-Chip Ethernet Controller for ISA.

VERSION: All silicon revisions.

CROSS REF: AN012001.TXT (PCnet-ISA+)

SYNOPSIS: ISA Plug and Play (PnP) devices are naturally quiescent following system power-on and the system reset sequence. In order for your system software to gain control of a Plug and Play device, to perform activities such as dynamic resource allocation by Configuration Management or EEPROM programming by a manufacturing utility, your software must first perform two basic operations: you must wake up the PnP device; you must allocate system resources to the device.

This Application Note takes you through the steps necessary to wake up the PnP Device with the standard Plug and Plug Initiation Key, or optionally, with the special AMD Initiation Key.

Then we'll cover basic Plug and Play Register initialization for the PCnet-ISA II Ethernet controller.

Publication# **21617** Rev. **A** Amendment/0
Issue Date: **March 1995**

Plug and Play

In a Plug and Play system, all plug and Play devices are quiescent on power up and following a hardware reset, therefore, each PnP device must be identified and configured by software before it can function.

The PnP Specification defines an access method, using standard I/O access ports and commands, that allows your software to find PnP devices, determine resources required by each PnP device, and reconfigure PnP devices to eliminate conflicts.

When your software issues a special sequence of data writes to the standard I/O access ports, the PnP logic in each PnP device is enabled. Once a PnP device is enabled, you can determine its resource requirements and capabilities and configure the device as needed. The special sequence of data, which is written to the standard I/O access port, is referred to as the Initiation Key.

The tasks of identifying all PnP devices, determining all of their resource requirements, and reconfiguring PnP devices are covered in great detail in the PnP Specification and will not be covered here.

The goal of this Application Note is to describe for you the data contained in the Initiation Key and the procedure for issuing the Initiation Key.

Auto-Configuration Ports

Three 8-bit I/O ports are used by the Plug and Play configuration software on each Plug and Play device to communicate with the Plug and Play registers. The ports are listed in the table below. The software configuration space is defined as a set of 8-bit registers. These registers are used by the Plug and Play software configuration manager to issue commands, access the resource information, check status, and configure the PCnet-ISA II controller hardware.

Name	Location	Type
ADDRESS	0x0279 (printer status port)	Write Only
WRITE_DATA	0x0A79 (printer status port + 0x0800)	Write Only
READ_DATA	Relocatable: 0x0203 to 0x03FF	Read Only

Table 1. Auto-Configuration Ports.

The ADDRESS and WRITE_DATA ports are located at fixed I/O addresses. The WRITE_DATA port is located at an address alias of the ADDRESS port. All three auto-configuration ports use a 12-bit ISA address decode.

The READ_DATA port is relocatable within the I/O range of 0x0203 to 0x03FF.

ADDRESS Port

The internal Plug and Play registers are accessed by writing the address to the ADDRESS port and then either reading the READ_DATA port or writing the WRITE_DATE port. Once the ADDRESS port has been written, any number of reads or writes can occur without having to rewrite the ADDRESS port.

The ADDRESS port is also the address to which the Initiation Key is written to, which is described later.

WRITE_DATA Port

The WRITE_DATA port is the adress to which all writes to the internal Plug and Play registers occur. The destination of the data written to the WRITE_DATA port is determined by the last value written to the ADDRESS port.

READ_DATA Port

The READ_DATA port is used to the read information from the internal Plug and Play registers. The address of the register to be read is determined by the last value written to the ADDRESS port.

The I/O address of the READ_DATA port is set by writing the proper value to Plug and Play register 0. The 8-bit value you write in Plug and Play register 0 is used to match I/O address bits 1 through 8.

Initiation Key

The PCnet-ISA II controller is disabled at power-on or hardware reset when operating in Plug and Play mode. It will not respond to any memory or I/O accesses, nor will the PCnet-ISA II controller drive any interrupts or DMA channels.

The Initiation Key places the PCnet-ISA II Plug and Play logic into the configuration mode. This is accomplished by writing a predefined sequence of data to the ADDRESS port. If the proper sequence of data are detected by the PCnet-ISA II controller, then the Plug and Play auto-configuration ports are enabled.

The exact sequence of data in the Initiation Key is illustrated in the following table of hexadecimal values.

6A, B5, DA, ED, F6, FB, 7D, BE, DF, 6F, 37, 1B, 0D, 86, C3, 61,

B0, 58, 2C, 16, 8B, 45, A2, D1, E8, 74, 3A, 9D, CE, E7, 73, 39

Table 2. Plug and Play Initiation Key.

AMD Initiation Key

The PCnet-ISA II controller relies heavily on the presence of a correctly programmed serial configuration EEPROM to establish run time operational parameters and to initialize the Plug and Play registers.

When the EEPROM does not exist, or in the case where there's an EEPROM checksum error, PCnet-ISA II waits in the Plug and Play WAIT_FOR_KEY state. Your software can gain access to the PCnet-ISA II I/O resources by writing a special AMD Initialization Key to the Plug and Play ADDRESS port at 0x0279.

The exact sequence of data in the AMD Initiation Key is illustrated in the following table of hexadecimal values.

6B, 35, 9A, CD, E6, F3, 79, BC, 5E, AF, 57, 2B, 15, 8A, C5, E2,

F1, F8, 7C, 3E, 9F, 4F, 27, 13, 09, 84, 42, A1, D0, 68, 34, 1A

Table 3. AMD Initiation Key.

Initiation Key Insider Tips

Your software should reset the (internal Linear Feedback Shift Register) hardware to its initial value by writing two 8-bit values of 0x00 to the ADDRESS port before the Initiation Key is sent.

The I/O writes of the Initiation Key must be contiguous with no other I/O cycles to the controller. I recommend that you disable system interrupts while issuing the Initation Key in order to avoid any extraneous I/O cycles.

In order to avoid the possibility of having the PCnet-ISA II controller miss the Initiation Key because it happened to be in the wrong state, I recommend that you send the Initiation Key to the controller twice in succession.

Plug and Play Register Initialization

Plug and Play Register initialization is the next step after getting access to the PCnet-ISA II controller. This activity serves two major purposes.

Generally, it is the point in controller initialization where you can override run time operational parameters read from the EEPROM or setup initial values when the EEPROM is missing or corrupted. By default, when the serial configuration EEPROM is missing or contains corrupted data, the PCnet-ISA II controller clears its internal Plug and Play Registers.

Specifically, it is the point in controller initialization where you can assign an ISA I/O address to the Plug and Play device. Once you have assigned the I/O address, your software can begin accessing the PCnet-ISA II controller resources: IEEE Address, Register Address Port, Register Data Port, and ISACSR Data Port.

The Plug and Play Register initialization function provided (ip_cfg_init()), illustrates a typical code sequence to initialize the PnP Registers to known working values for the reference adapter card design supplied by AMD. Your adapter card design may require different values.

The power on default values of the Plug and Play Registers in the PCnet-ISA II controller are shown in Table 4.

PnP Register	Default
0x60	0000001x
0x61	xxx00000
0x70	00000011
0x71	00000010
0x74	00000011
0x40	00000000
0x41	00000000
0x42	00000010
0x43	00000000
0x44	xxx00000
0x48	00000000
0x49	00000000
0x4A	00000010
0x4B	00000000
0x4C	xxx00000
0xF0	00000000

Table 4. Plug and Play Register Default Values

```
/*****************************************************************************
NAME
   ip_esrm - Exit software relocatable mode.

 DESCRIPTION
   ip_esrm() will wake up the PCnet-ISA II controller and initialize the
    Plug and Play Registers.

    First, the Plug and Play Initiation Key is used to get the controller
   to exit software relocatable mode. If this attempt fails, the AMD
   Initiation Key is used to get the controller to exit software
   relocatable mode.

   If either method successful in waking the controller, the Plug and Play
   Registers are initialized.

 CALLS
   inport() - Gets a word or byte from a hardware port. Function in the
   (Borland) run-time library.

   AROME - #define AROME Iobase+0xE

 RETURN
   00 = PCnet-ISA II Plug and Play device awake and PnP registers initialized.
   01 = Abort - Adapter not found.

*****************************************************************************/
int     ip_esrm(void)
{
    int     cc = 0;

    /* Send the standard Plug and Play Initiation Key. */
    ip_wake1();

    /* Initialize the Plug and Play Registers. */
    ip_cfg_init();

    /* If the target PnP device was not awakened, then we'll have to
       attempt the special AMD Initiation Key. */
    if ( inport(AROME) != 0x5757 ) {
        ip_wake2();                    /* ...try the secret AMD handshake. */
        ip_cfg_init();                 /* Init Plug and Play Registers. */

       /* If the PnP device is awake now, we're finished. */
       if ( inport(AROME) == 0x5757 ) {
          printf("ip_esrm: Info - AMD controller has been activated.\n");
       }
       /* Otherwise, the lights are on, but nobody's home. */
       else {
           cc = 1;
          printf("ip_esrm: Abort - Unable to activate AMD controller.\n");
       }
    }
    else {
          printf("ip_esrm: Info - AMD controller has been activated.\n");
    }

   return(cc);
}
```

```
/*****************************************************************************
NAME
   ip_wake1 - Perform controller wake up sequence 1.

 DESCRIPTION
    This function writes the standard Plug and Play Initiation Key to
    the Plug and Play Address Port at I/O address 0x0279.

    This operation is required to bring a Plug and Play device out of
    inactive mode so it can be configured.

    Note: The standard 6A sequence works on a PCnet-ISA II controller with a
    configuration EEPROM that contains a good Checksum 2.
    If the configuration EEPROM is missing or the EEPROM Checksum 2 is
    incorrect, then you must use the special AMD 6B sequence.

 CALLS
    outportb() - Outputs a byte to a hardware port. Function in the
    (Borland) run-time library.
*****************************************************************************/
void    ip_wake1(void)
{
    int     i;
    for ( i = 0; i < 2; i++ ) {
        outportb(0x279, 0x00);          /* Initialize LFSR. */
        outportb(0x279, 0x00);          /* Initialize LFSR. */
        outportb(0x279, 0x6A);          /* 0x6A is the PnP sequence. */
        outportb(0x279, 0xB5);
        outportb(0x279, 0xDA);
        outportb(0x279, 0xED);
        outportb(0x279, 0xF6);
        outportb(0x279, 0xFB);
        outportb(0x279, 0x7D);
        outportb(0x279, 0xBE);
        outportb(0x279, 0xDF);
        outportb(0x279, 0x6F);
        outportb(0x279, 0x37);
        outportb(0x279, 0x1B);
        outportb(0x279, 0x0D);
        outportb(0x279, 0x86);
        outportb(0x279, 0xC3);
        outportb(0x279, 0x61);
        outportb(0x279, 0xB0);
        outportb(0x279, 0x58);
        outportb(0x279, 0x2C);
        outportb(0x279, 0x16);
        outportb(0x279, 0x8B);
        outportb(0x279, 0x45);
        outportb(0x279, 0xA2);
        outportb(0x279, 0xD1);
        outportb(0x279, 0xE8);
        outportb(0x279, 0x74);
        outportb(0x279, 0x3A);
        outportb(0x279, 0x9D);
        outportb(0x279, 0xCE);
        outportb(0x279, 0xE7);
        outportb(0x279, 0x73);
        outportb(0x279, 0x39);
    }
}
```

```
/*****************************************************************************
NAME
   ip_wake2 - Perform controller wake up sequence 2.

 DESCRIPTION
    This function writes the special AMD Initiation Key to the Plug
    and Play Address Port at I/O address 0x0279.

    This operation is required to bring a Plug and Play device out of
    inactive mode so it can be configured.

    Note: The standard 6A sequence works on a PCnet-ISA II controller with a
    configuration EEPROM that contains a good Checksum 2.
    If the configuration EEPROM is missing or the EEPROM Checksum 2 is
    incorrect, then you must use the special AMD 6B sequence.

 CALLS
    outportb() - Outputs a byte to a hardware port. Function in the
    (Borland) run-time library.
*****************************************************************************/
void    ip_wake2(void)
{
    int     i;
    for ( i = 0; i < 2; i++ ) {
        outportb(0x279, 0x00);          /* Initialize LFSR. */
        outportb(0x279, 0x00);          /* Initialize LFSR. */
        outportb(0x279, 0x6B);          /* 0x6B is the AMD sequence! */
        outportb(0x279, 0x35);
        outportb(0x279, 0x9A);
        outportb(0x279, 0xCD);
        outportb(0x279, 0xE6);
        outportb(0x279, 0xF3);
        outportb(0x279, 0x79);
        outportb(0x279, 0xBC);
        outportb(0x279, 0x5E);
        outportb(0x279, 0xAF);
        outportb(0x279, 0x57);
        outportb(0x279, 0x2B);
        outportb(0x279, 0x15);
        outportb(0x279, 0x8A);
        outportb(0x279, 0xC5);
        outportb(0x279, 0xE2);
        outportb(0x279, 0xF1);
        outportb(0x279, 0xF8);
        outportb(0x279, 0x7C);
        outportb(0x279, 0x3E);
        outportb(0x279, 0x9F);
        outportb(0x279, 0x4F);
        outportb(0x279, 0x27);
        outportb(0x279, 0x13);
        outportb(0x279, 0x09);
        outportb(0x279, 0x84);
        outportb(0x279, 0x42);
        outportb(0x279, 0xA1);
        outportb(0x279, 0xD0);
        outportb(0x279, 0x68);
        outportb(0x279, 0x34);
        outportb(0x279, 0x1A);
    }
}
```

```
/*****************************************************************************
NAME
    ip_cfg_init - Plug and Play Register initialization.

 DESCRIPTION
    This function initializes the PCnet-ISA II Plug and Play registers with a
   predefined set of values.

    In order to provide compatability with AMD device drivers, this function
    writes the ASCII WW pattern in EEPROM word location 7.

 INPUT
    Global variable Iobase is assumed to contain the 16-bit I/O address that
    will locate the target Plug and Play device in the I/O memory space.

 CALLS
    outport() - Outputs a word or byte to a hardware port. Function in the
    (Borland) run-time library.

    AROME - #define AROME Iobase+0xE

*****************************************************************************/
void    ip_cfg_init(void)
{
    int temp;

    ip_cfg_w(0x02, 0x05);              /* RESET command. */
    ip_cfg_w(0x03, 0x00);              /* WAKE[0] command. */
    ip_cfg_w(0x06, 0x01);              /* SET_CSN[1] command. */

    ip_cfg_w(0x60, (BYTE)((Iobase & 0xFF00) >> 8));    /* I/O addr: 15:8 */
    ip_cfg_w(0x61, (BYTE)(Iobase & 0x00FF));           /* I/O addr: 7:0 */
    ip_cfg_w(0x70, 0x00);              /* IRQ level: 0 = no IRQ selection. */
    ip_cfg_w(0x71, 0);                 /* IRQ type: Edge, active low. */
    ip_cfg_w(0x74, 0x00);              /* DMA 0: Channel 0. */
    ip_cfg_w(0x43, 0xFE);              /* Mem Desc 0: bit0 = 0 = disable. */
    ip_cfg_w(0x4B, 0xFE);              /* Mem Desc 1: bit0 = 0 = disable. */
    ip_cfg_w(0xF0, 0x00);              /* Vendor (AMD) Defined Byte. */
    ip_cfg_w(0x31, 0x00);              /* Disable I/O range check. */
    ip_cfg_w(0x30, 0x01);              /* Activate Reg: bit0 = 1 = active. */

    ip_cfg_w(0x02, 0x02);              /* Cfg Ctl: bit1 = 1 = WAIT_FOR_KEY */

    /* Write ASCII WW to PCnet-ISA II internal address PROM. */
    temp = ip_bcr_r(2);                /* Set APWEN bit... */
    ip_bcr_w(2, (temp | 0x0100));      /* ...in ISACSR2. */

    outport(AROME, 0x5757);            /* Stuff the "WW". */

    temp = ip_bcr_r(2);                /* Clear APWEN bit... */
    ip_bcr_w(2, (temp & 0xFEFF));      /* ...in ISACSR2. */
}
```

```
/*****************************************************************************
NAME
    ip_cfg_w - Plug and Play Register write.

 DESCRIPTION
    This function performs a single 8-bit write to the specified Plug and
     Play Register.

    The Plug and Play Register number is written to the Auto-configuration
    Port (ACP) at I/O address 0x0279 then the data is written to the ACP
    data port at I/O address 0x0A79.

    Note that per the Plug and Play Specification, all Plug and Play register
    accesses are 8-bit.

 INPUT
    Argument <reg> specifies the Plug and Play register you want to access.
    Argument <data> specifies the data you want to write.

 CALLS
    outportb() - Outputs a byte to a hardware port. Function in the
    (Borland) run-time library.

*****************************************************************************/
void    ip_cfg_w(int reg, BYTE data)
{
    outportb(0x279, reg);               /* Write ACP address. */
    outportb(0xA79, data);              /* Write ACP data. */
}
```

```
/******************************************************************************
NAME
   ip_bcr_r - Bus Configuration Register (BCR) read.

 DESCRIPTION
   This function performs a single 16-bit read of the specified Bus
   Configuration Register. You must specify the BCR you want access.

   In the data sheet for the PCnet-ISA II controller, the bus configuration
   registers are called ISA Bus Configuration Registers (ISACSRs). The new
   standardized name for these registers for the PCnet family is Bus
   Configuration Register (BCR).

 INPUT
   Argument <reg> specifies the BCR you want to read.

 CALLS
   outport() - Outputs a word or byte to a hardware port. Function in the
   (Borland) run-time library.

   inport() - Gets a word or byte from a hardware port. Function in the
   (Borland) run-time library.

   RAP - #define RAP Iobase+0x12
   IDP - #define IDP Iobase+0x16

 RETURN
   The data read from the specified BCR is returned to you.
   If the specified register is not a legal number, the returned value
   is 0xDEAD.

******************************************************************************/
WORD16  ip_bcr_r(int reg)
{
   if ( reg <= 8 ) {
      outport(RAP, reg);            /* Write Register Address Port. */
      return( inport(IDP) );        /* Read BCR Data Port. */
   }
   else
      return(0xDEAD);
}
```

```
/*****************************************************************************
NAME
    ip_bcr_w - Bus Configuration Register (BCR) write.

 DESCRIPTION
    This function performs a single 16-bit write to the specified Bus
    Configuration Register. You must specify the BCR you want to access and
    the data to be written.

    In the data sheet for the PCnet-ISA II controller, the bus configuration
    registers are called ISA Bus Configuration Registers (ISACSRs). The new
    standardized name for these registers for the PCnet family is Bus
    Configuration Register (BCR).

 INPUT
    Argument <reg> specifies the BCR you want to access.
    Argument <data> specifies the data you want to write.

 OUTPUT
    The specified data is written to the specified BCR.

 CALLS
    outport() - Outputs a word or byte to a hardware port. Function in the
    (Borland) run-time library.

    RAP - #define RAP Iobase+0x12
    IDP - #define IDP Iobase+0x16

 RETURN
    0   Function successfully completed.
    1   Error, Unknown Register.

*****************************************************************************/
int     ip_bcr_w(int reg, WORD16 data)
{
    if ( reg <= 8 ) {
        outport(RAP, reg);          /* Write Register Address Port. */
        outport(IDP, data);         /* Write BCR Data Port. */
        return(0);
    }
    else
        return(1);
}
```

REFERENCES:
Ethernet/IEEE 802.3 Family
1994 World Network Data Book/Handbook
Publication ID. 14287C
Advanced Micro Devices, Inc.
AMD Literature: 1-800-222-9323

Am79C961A PCnet-ISA II Jumperless, Full Duplex Single-Chip Ethernet
Controller for ISA data sheet.
Publication ID. 19364 Rev.A October 1994
Advanced Micro Devices, Inc.

Plug and Play ISA Specification
Version 1.0a May 5, 1994
CompuServe: GO PLUGPL   Forum Library File: ISA10A.ZIP

Clarifications to the Plug and Play ISA Specification, Version 1.0a
August 9, 1994
CompuServe: GO PLUGPL   Forum Library File: CLAR_I.ZIP

/* End of Application Note: AN013001.TXT */