MSI™
Link to the Future

PCI Configuration Register Programming

jasonlin@msi.com.tw

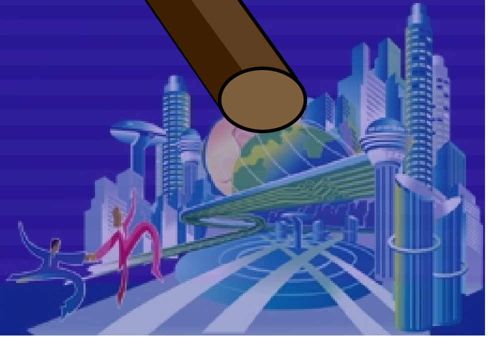
MSI
Link to the Future

Agenda

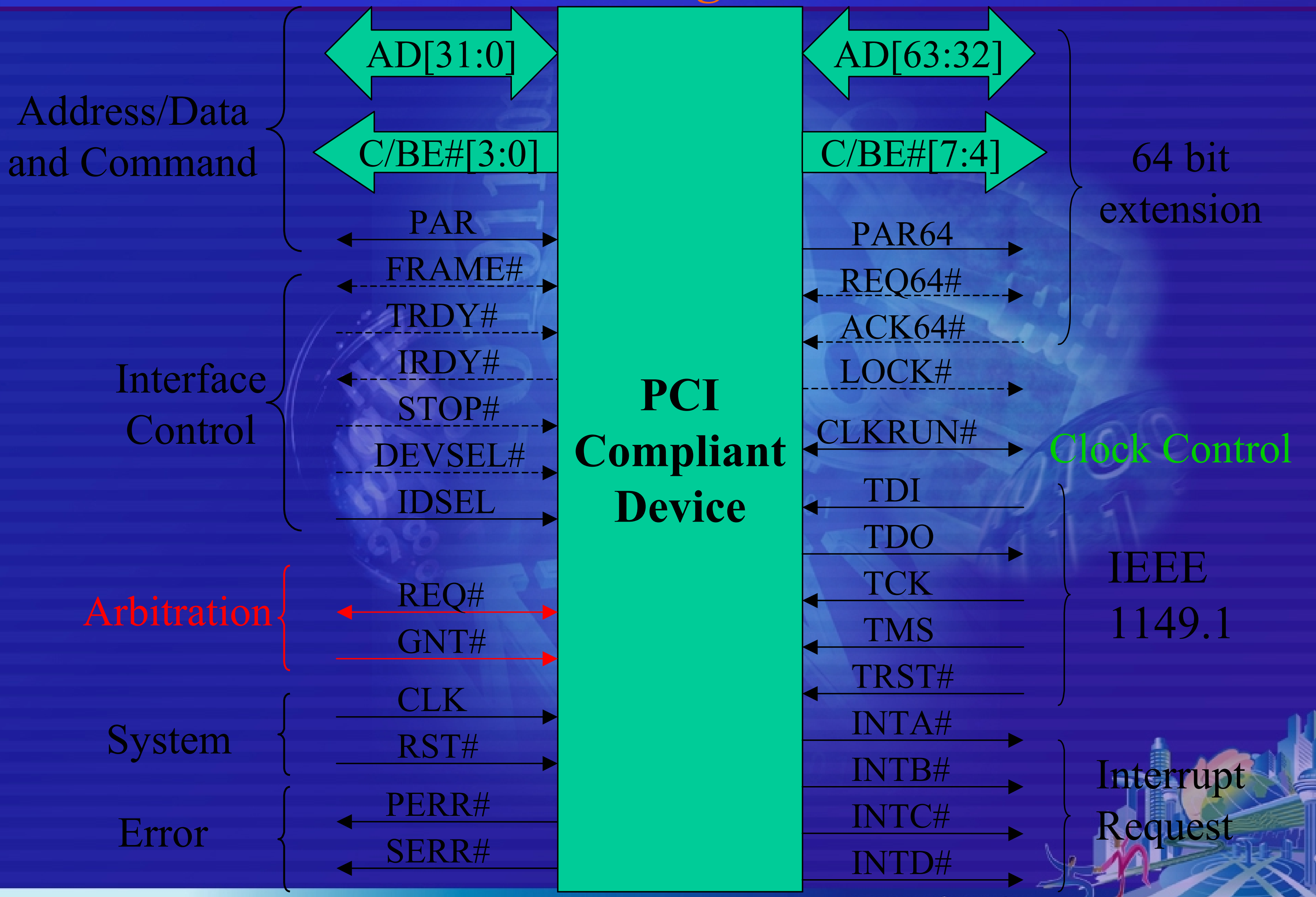
MSI™ Link to the Future

The PCI Bus

- *Address and Data multiplex together*
- *Address Phase*
  - **Command**
- *Data Phase*
  - **Byte Enable**



PCI Signals


Require Signal
Optional Signal
Address/Data and Command
AD[31:0]
C/BE#[3:0]
PAR
Interface Control
FRAME#
TRDY#
IRDY#
STOP#
DEVSEL#
IDSEL
Arbitration
REQ#
GNT#
System
CLK
RST#
Error
PERR#
SERR#
PCI Compliant Device
AD[63:32]
C/BE#[7:4]
PAR64
REQ64#
ACK64#
64 bit extension
LOCK#
CLKRUN#
Clock Control
TDI
TDO
TCK
TMS
TRST#
IEEE 1149.1
INTA#
INTB#
INTC#
INTD#
Interrupt Request

Command/Byte Enable

- C/BE#[0:3]
- During the address phase
  - Define the *transaction type* (like I/O read, I/O write, etc)
- During the data phase
  - Define the *bytes* to be used to transfer the data

MSI™
Link to the Future

PCI Command Type

C/BE3#	C/BE2#	C/BE1#	C/BE0#	Command Type
0	0	0	0	Interrupt Acknowledge
0	0	0	1	Special Cycle
0	0	1	0	I/O Read
0	0	1	1	I/O Write
0	1	1	0	Memory Read
0	1	1	1	Memory Write
1	0	1	0	Configuration Read
1	0	1	1	Configuration Write
1	1	0	0	Memory Read Multiple
1	1	0	1	Dual-Access Cycle
1	1	1	0	Memory Read Line
1	1	1	1	Mem. Write and Invalid

MSI™
Link to the Future

PCI Byte Enable Mapping

Byte Enable Signal	Map To
C/BE3#	Data Path 3, AD[31:24]
C/BE2#	Data Path 2, AD[23:16]
C/BE1#	Data Path 1, AD[15:8]
C/BE0#	Data Path 0, AD[7:0]

Typical PCI Translation

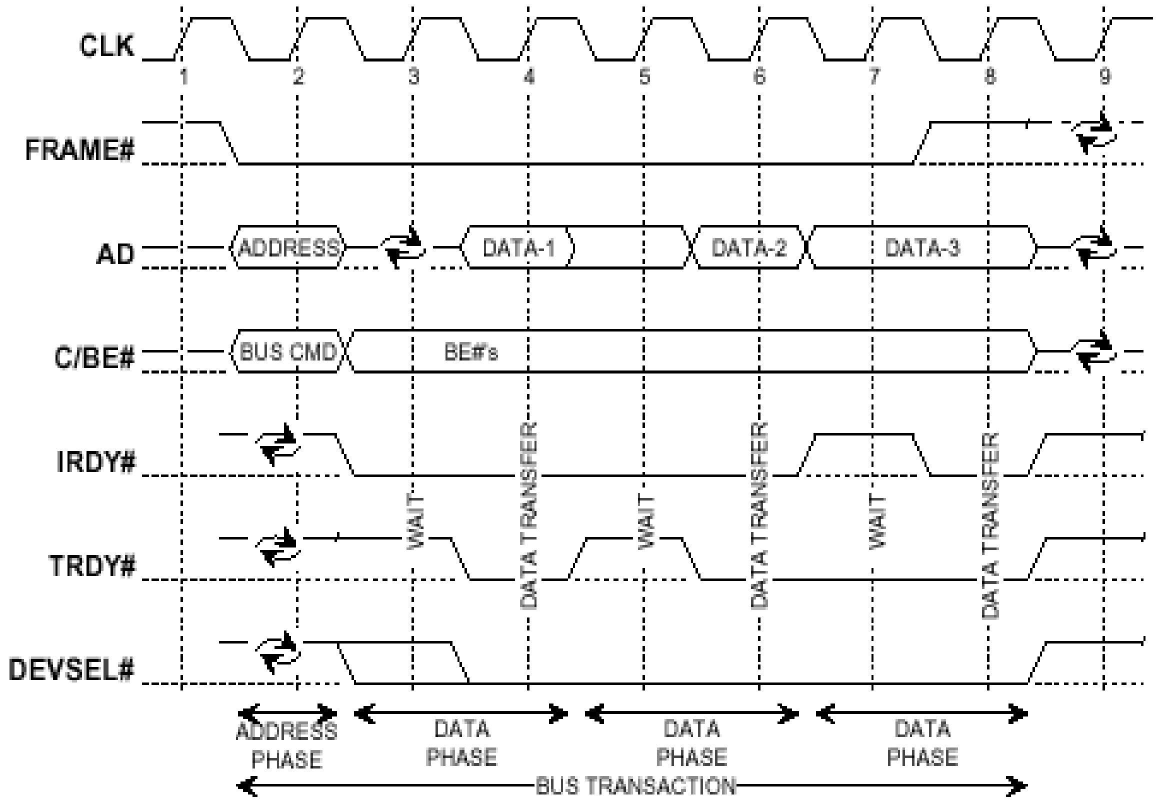


Figure 3-5: Basic Read Operation

MSI™ Link to the Future

PCI Configuration Mechanism #1

1.Configuration Port :

a.Address Port : 0CF8h (Access as DWORD)

b.Data Port : 0CFCh

2.Type 0 Configuration Access Translation

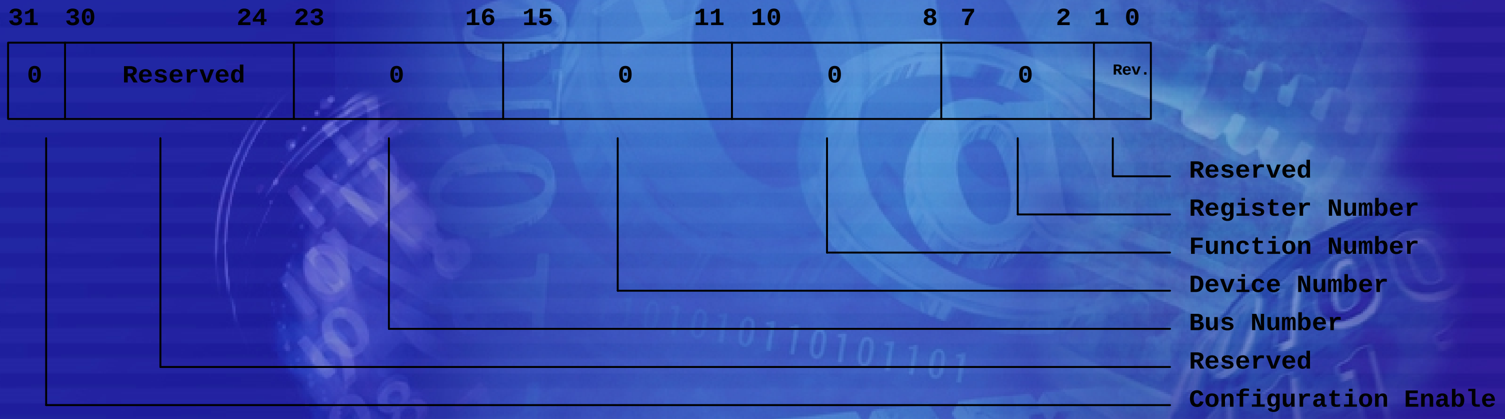
3.Type 1 Configuration Access Translation

4.Type 1 to 0 Configuration Access Translation

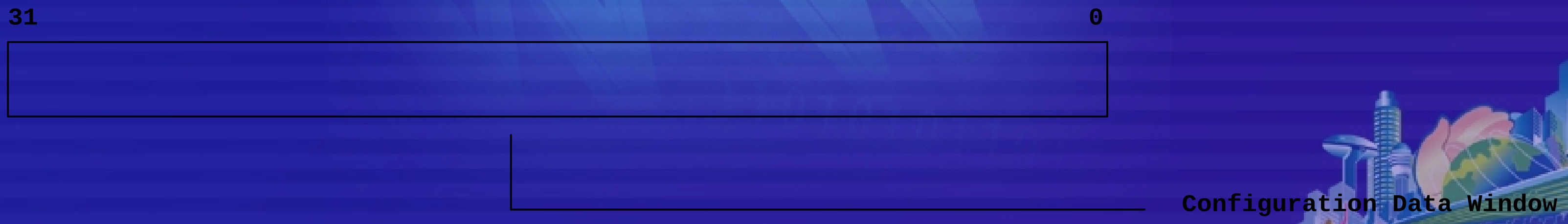
5.Type 0 to 0 Configuration Access Translation

PCI Configuration Register Format

Configuration Address Register :



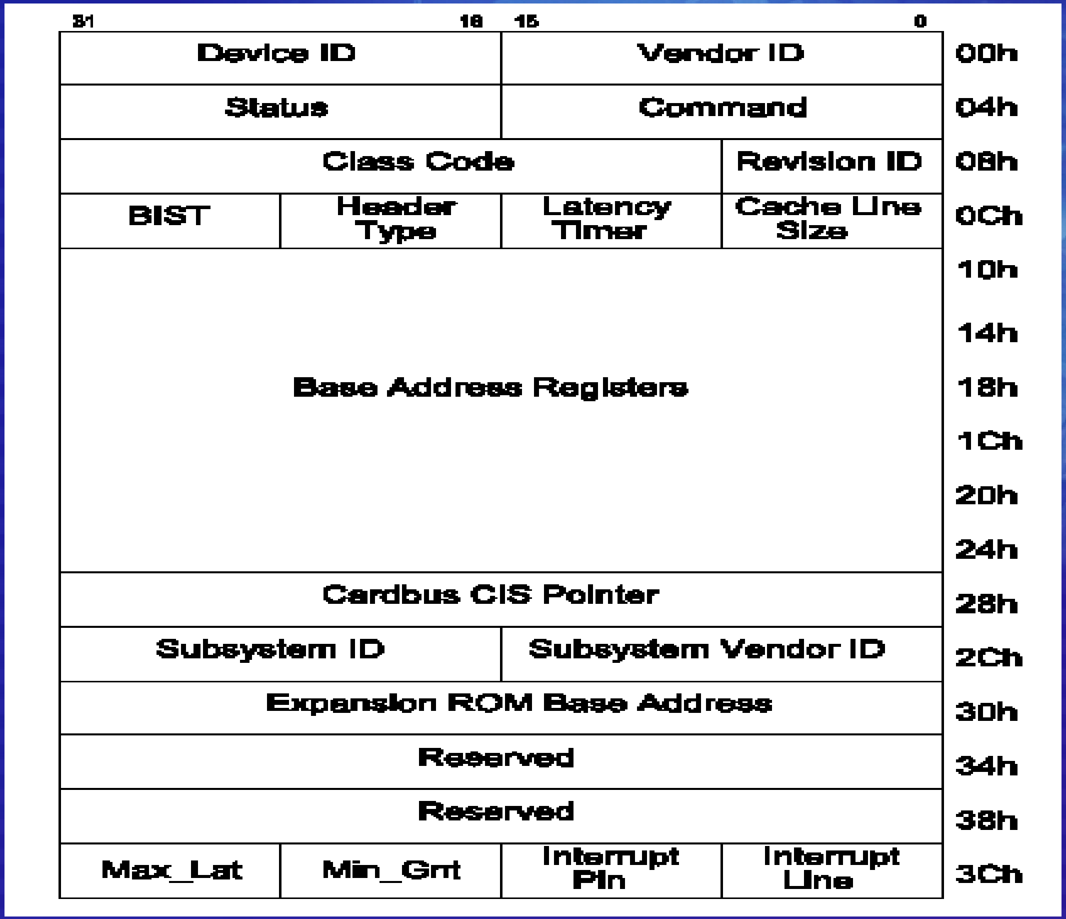
Configuration Date Register :



PCI Configuration Space

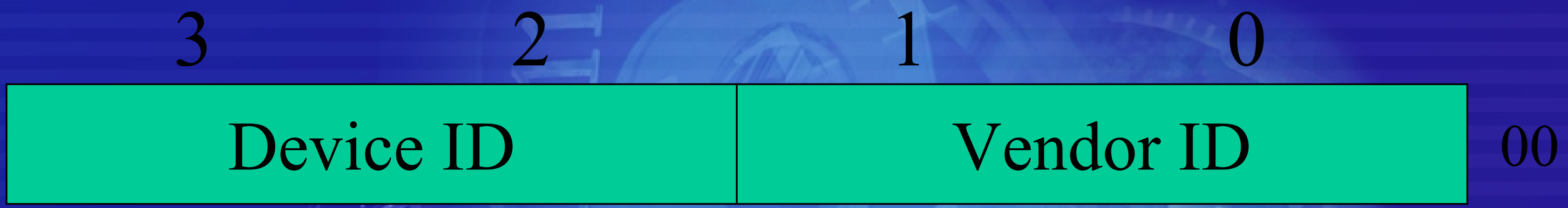
- I/O address space
  - *4GB (2^{32}) in size*
- Memory address space
  - *4GB (2^{32}) in size*
- Configuration address space
  - *First 16 double words is device configuration header space*
  - *Last 48 double words is device-specific configuration register*

PCI Configuration Space



31	16	15	0	
Device ID		Vendor ID		00h
Status		Command		04h
Class Code			Revision ID	08h
BIST	Header Type	Latency Timer	Cache Line Size	0Ch
Base Address Registers				10h
				14h
				18h
				1Ch
				20h
				24h
Cardbus CIS Pointer				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Expansion ROM Base Address				30h
Reserved				34h
Reserved				38h
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	3Ch

Vendor ID & Device ID



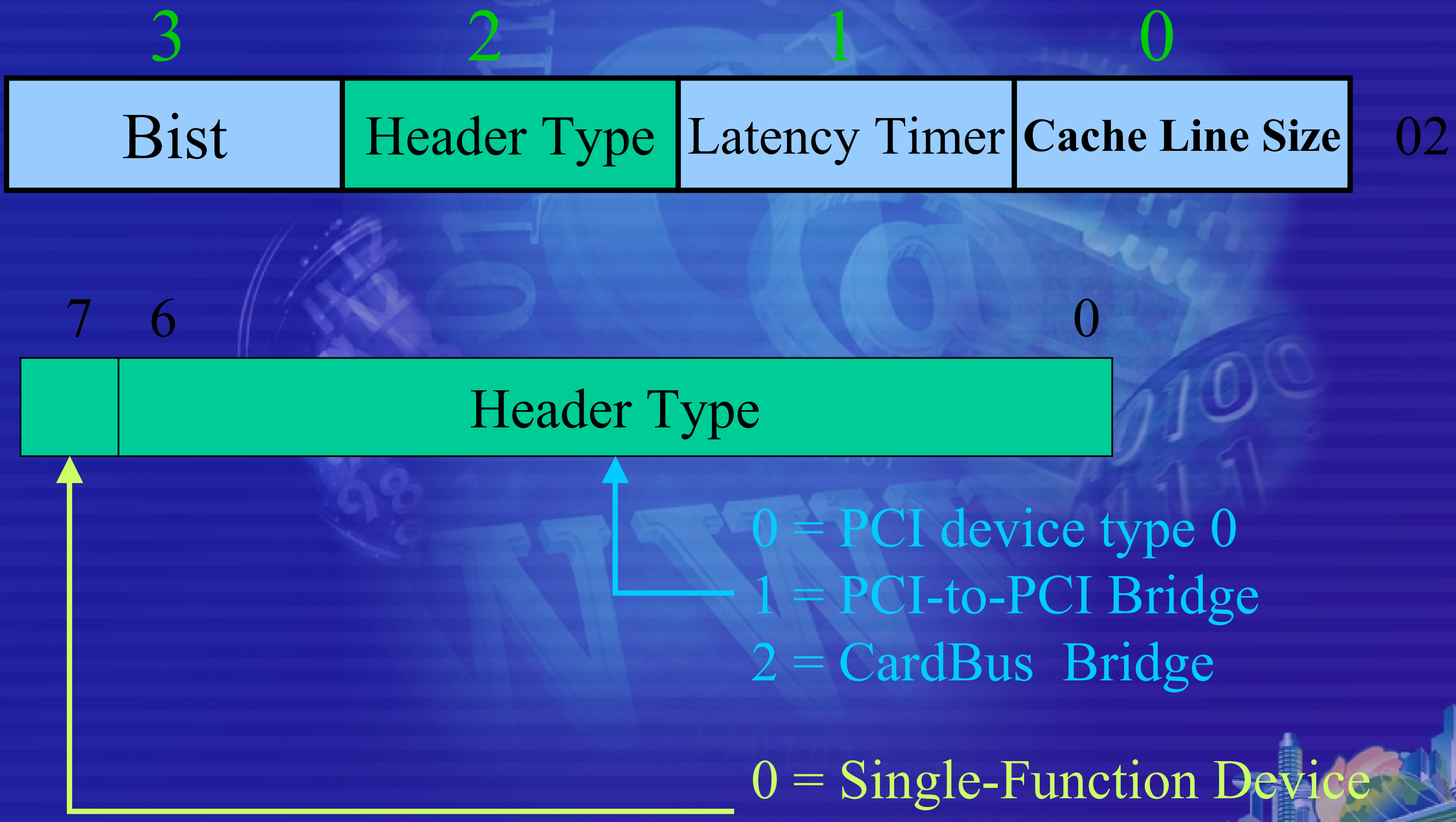
- **Vendor ID**
  - *Return "FFFFh" when read from a non-existent device*
- **Device ID**
  - *Assign by the device manufacturer and identifies the type of device*

Class Code

3	2	1	0	
Class Code	Sub-Class Code	Prog. I/F	Revision ID	02

Class	Description	Class	Description
00h	Before PCI 2.0	08h	Base system peripheral
01h	Mass storage controller	09h	Input device
02h	Network controller	0Ah	Docking station
03h	Display controller	0Bh	Processors
04h	Multimedia controller	0Ch	Serial bus controller
05h	Memory controller	0D-FEh	Reserved
06h	Bridge device	FFh	Device not fit above
07h	Simple communication controller		

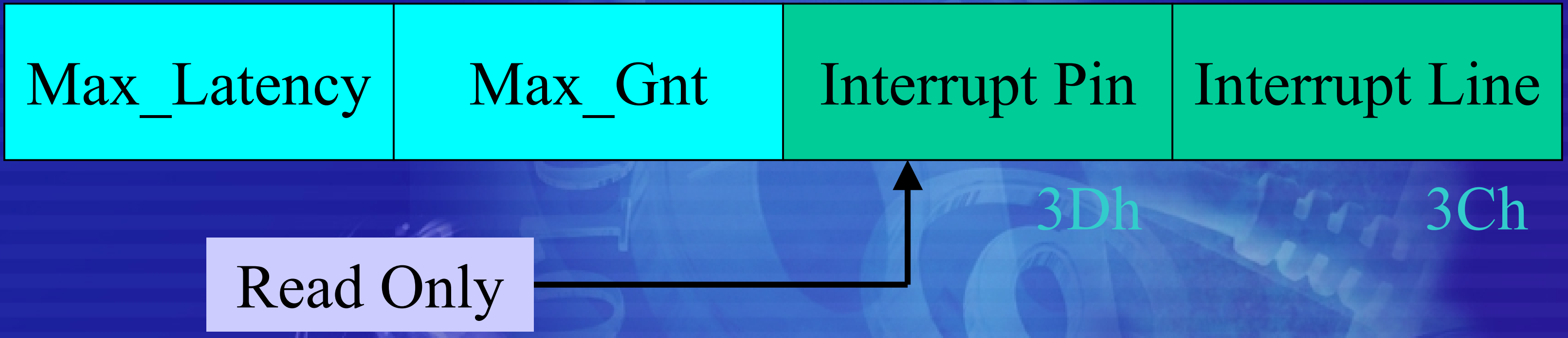
Header Type Register



0 = PCI device type 0
1 = PCI-to-PCI Bridge
2 = CardBus Bridge

0 = Single-Function Device
1 = Multi-Function Device

PCI Interrupt Pin Register

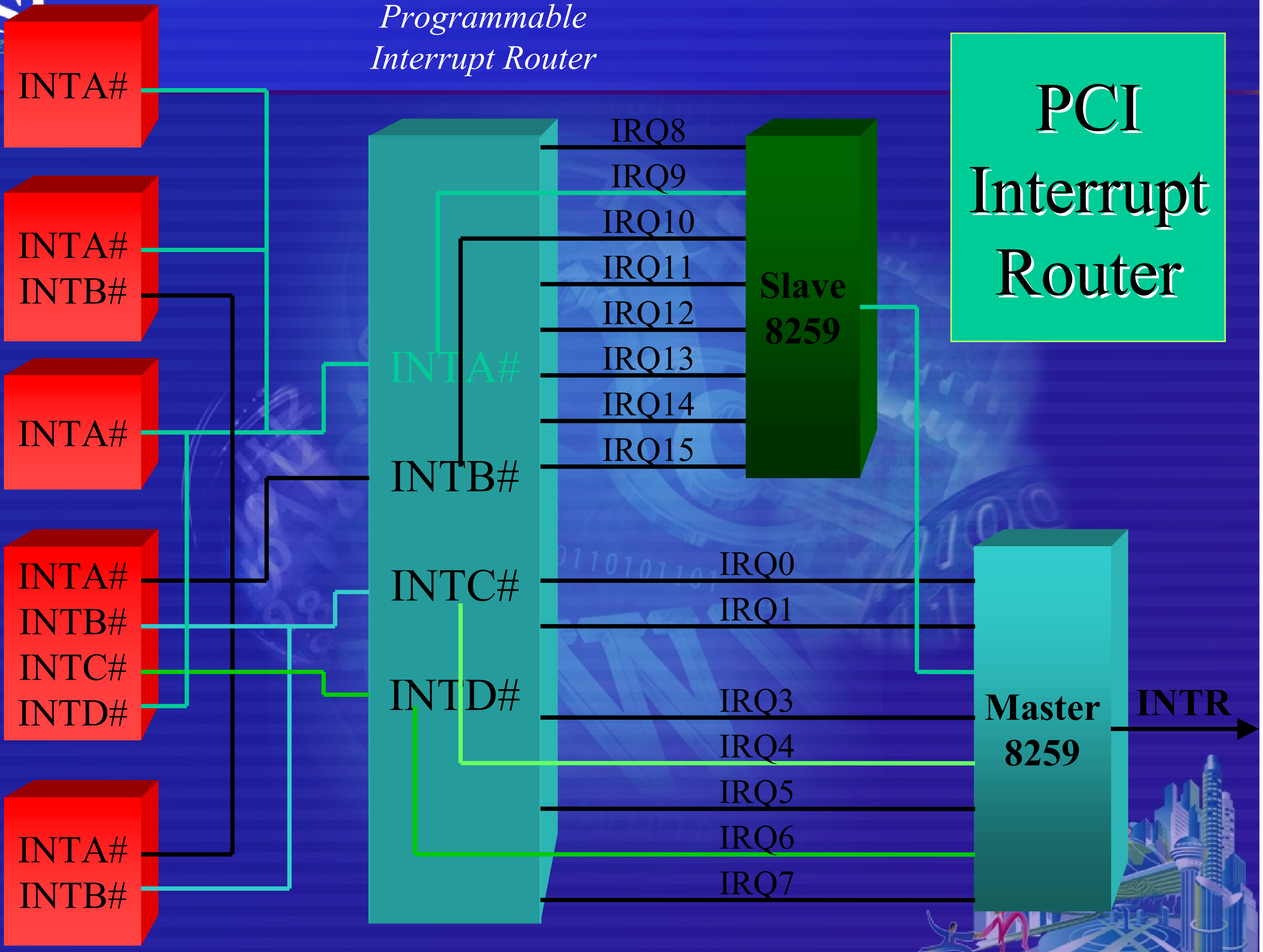


Interrupt Signal Bounded to	Value in Pin Register
INTA# Pin	01h
INTB# Pin	02h
INTC# Pin	03h
INTD# Pin	04h

PCI Interrupt Line Register

System IRQ line Routed to	Value in Line Register
IRQ0	00h
IRQ1	01h
IRQ2	02h
IRQ3	03h
IRQ12	0Ch
IRQ13	0Dh
IRQ14	0Eh
IRQ15	0Fh

PCI Interrupt Router

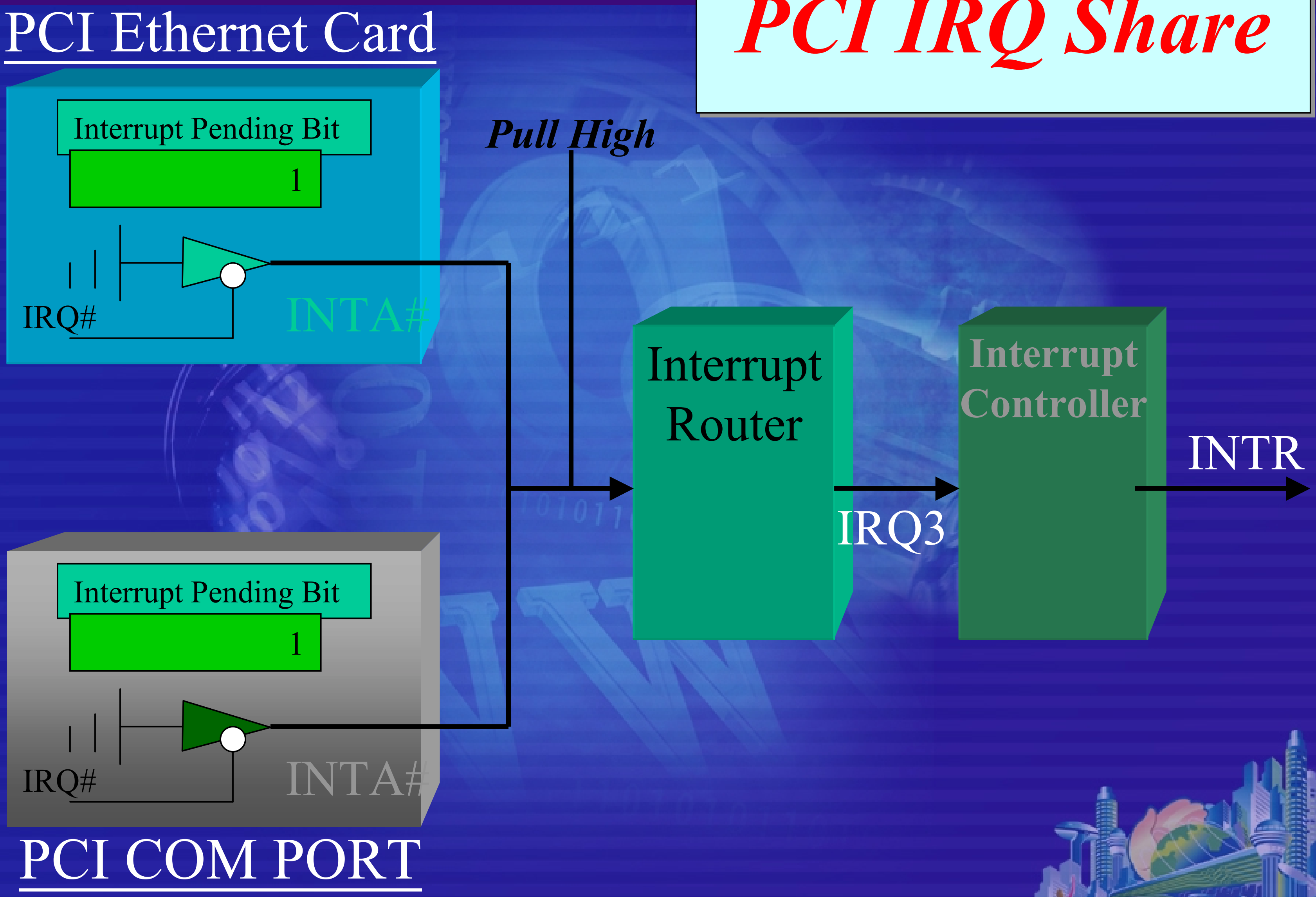


MSI™
Link to the Future

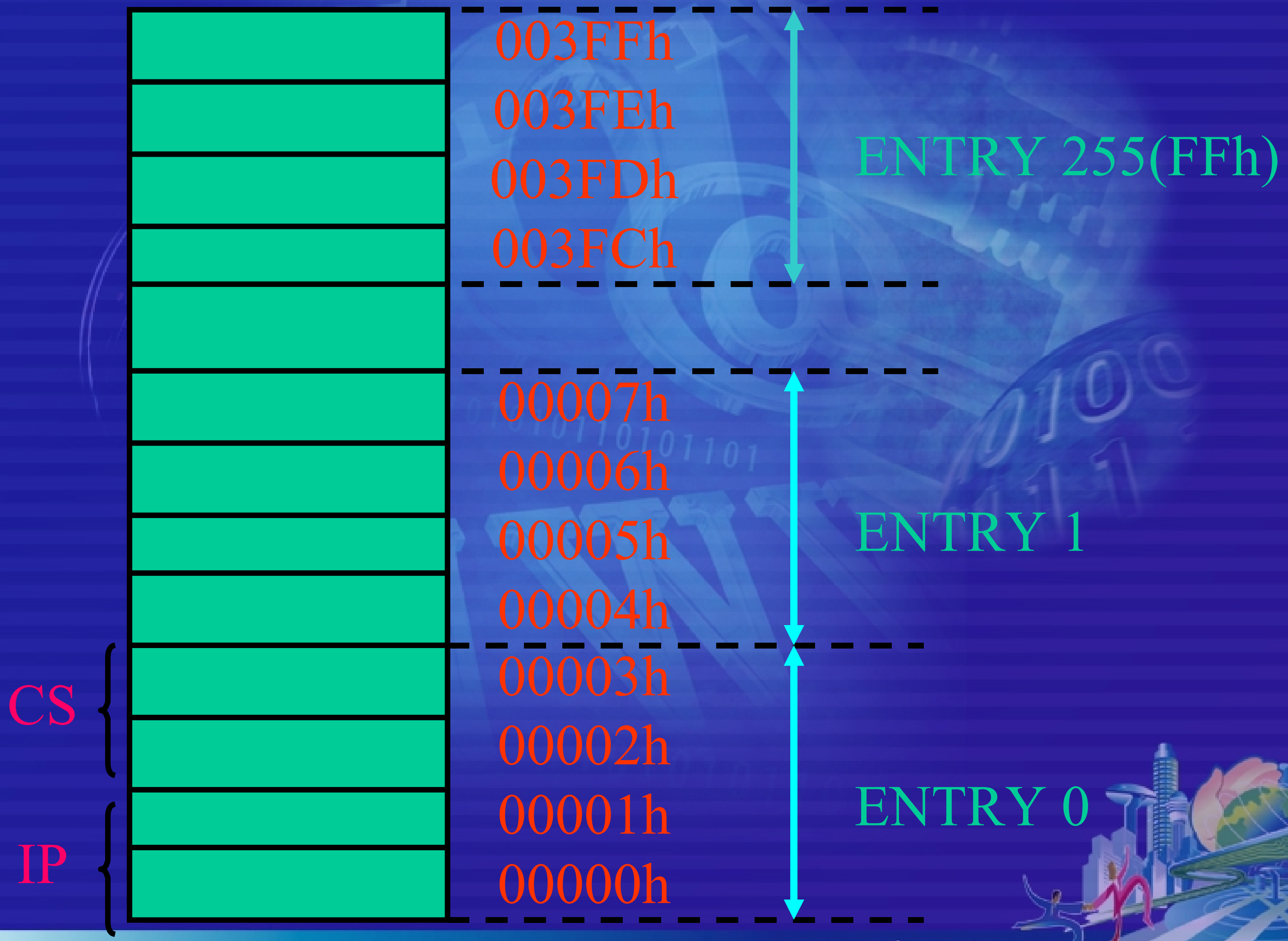
Programmable Interrupt Router

- *Software programmable for startup configuration*
- *Program the router to route each PCI interrupt to unused ISA IRQ lines*
- *PCI interrupt is active low, level sensitive, shareable inputs*

PCI IRQ Share



Interrupt Table

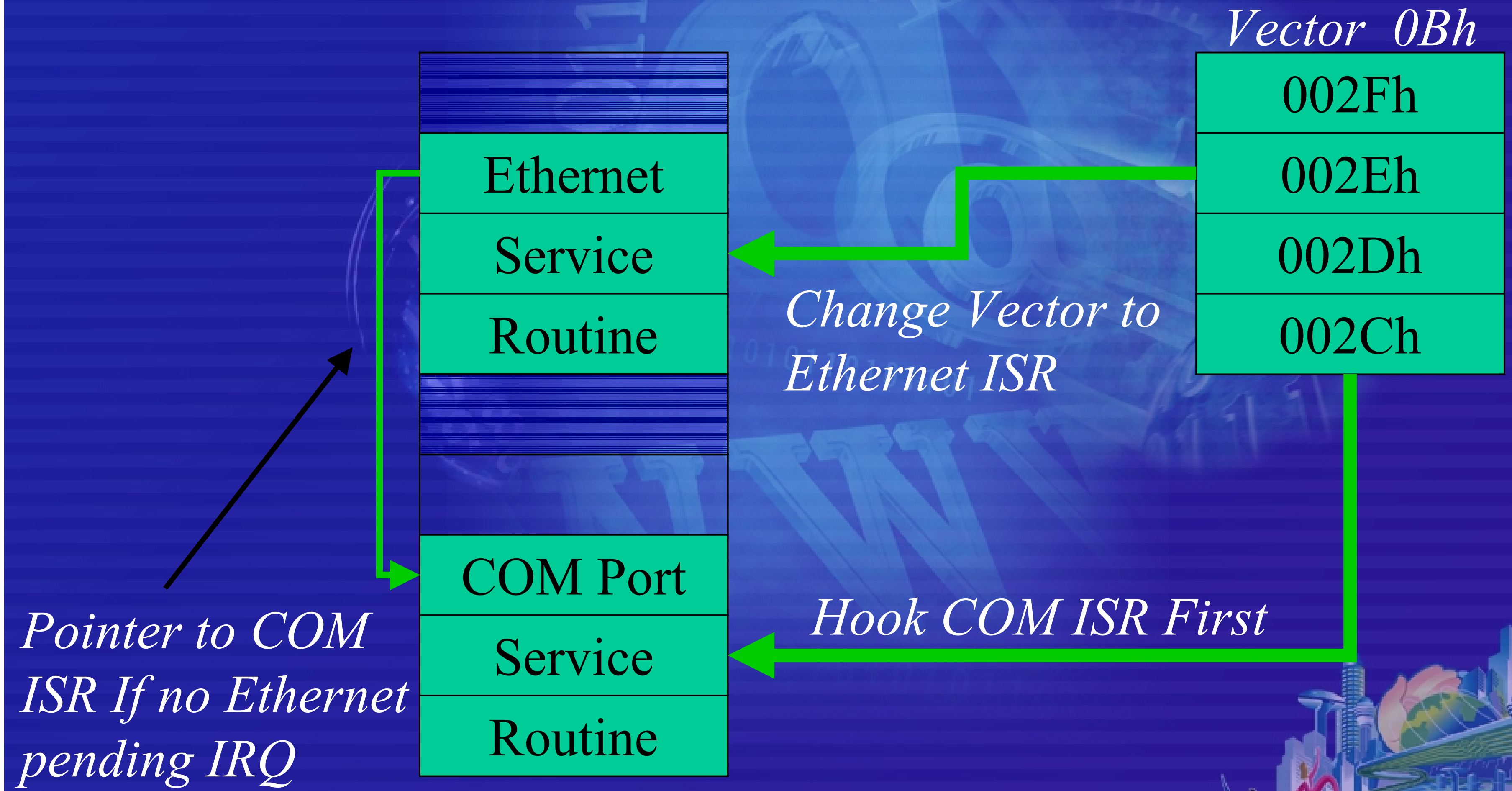


MSI™
Link to the Future

ISA Interrupt Table Entry Number

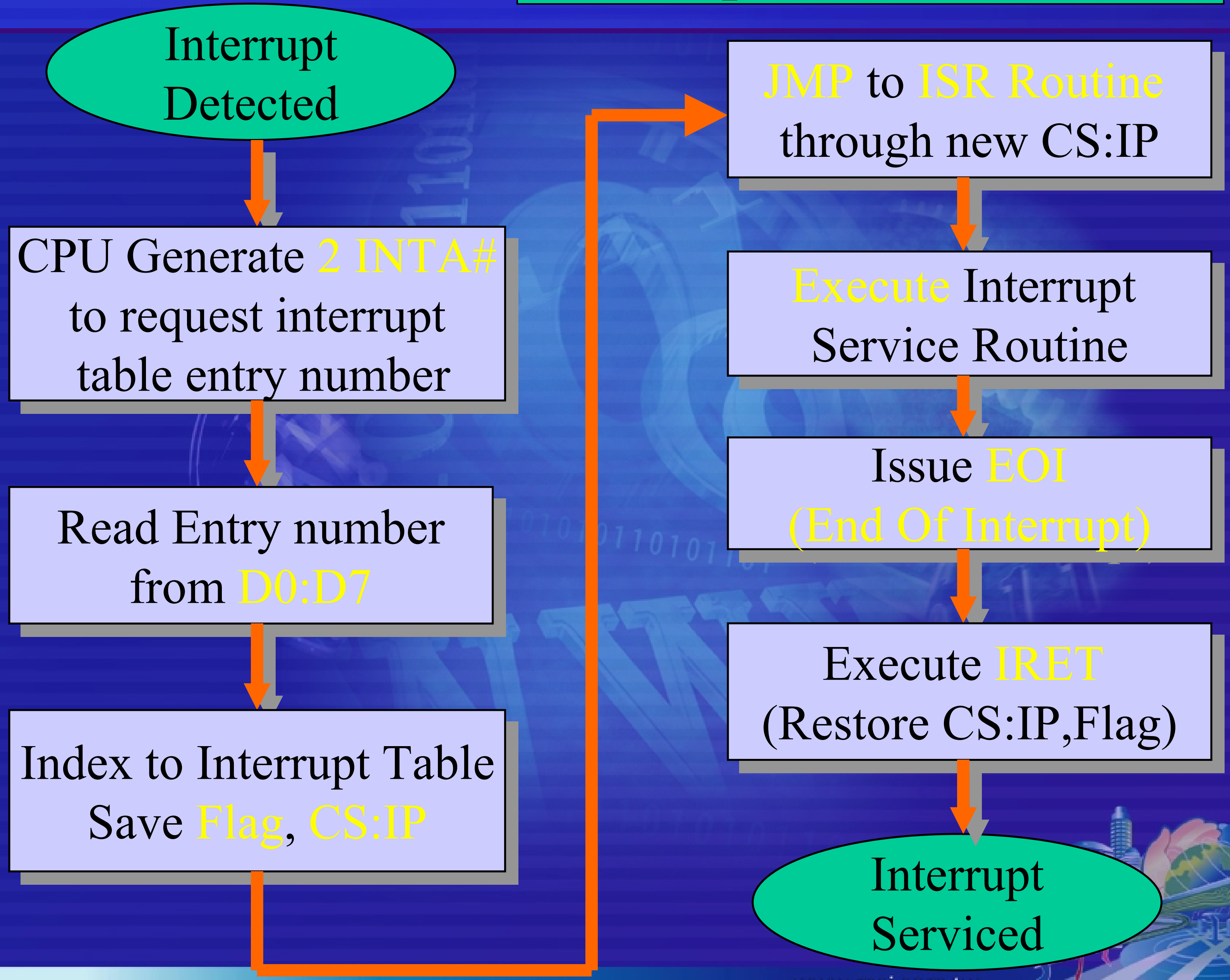
IRQ Line	Interrupt Entry No	IRQ Line	Interrupt Entry No
0	08h	8	70h
1	09h	9(or2)	71h
3	0Bh	10	72h
4	0Ch	11	73h
5	0Dh	12	74h
6	0Eh	13	75h
7	0Fh	14	76h
		15	77h

Linking List for IRQ Sharing



MSI
Link to the Future

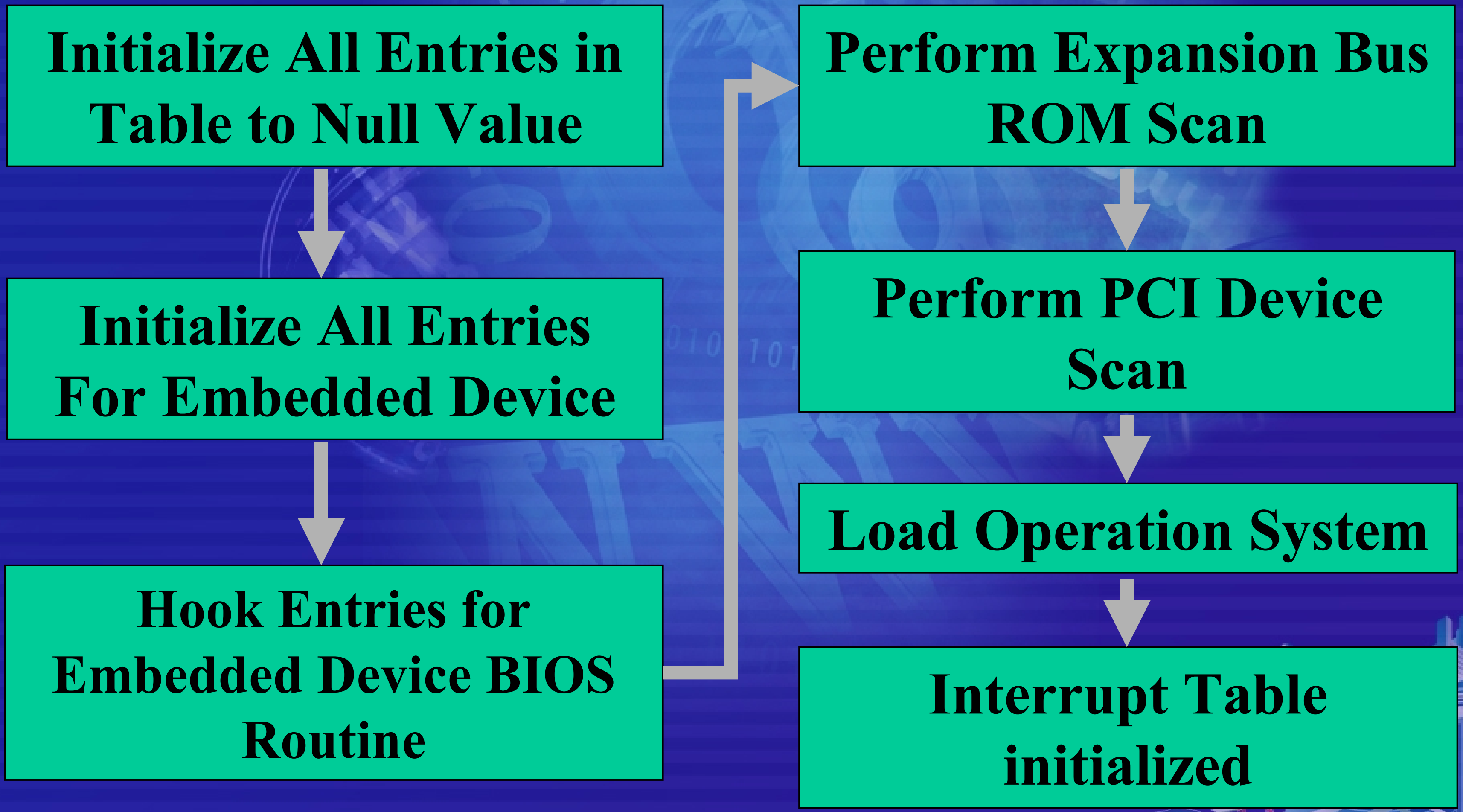
Interrupt Handler Process



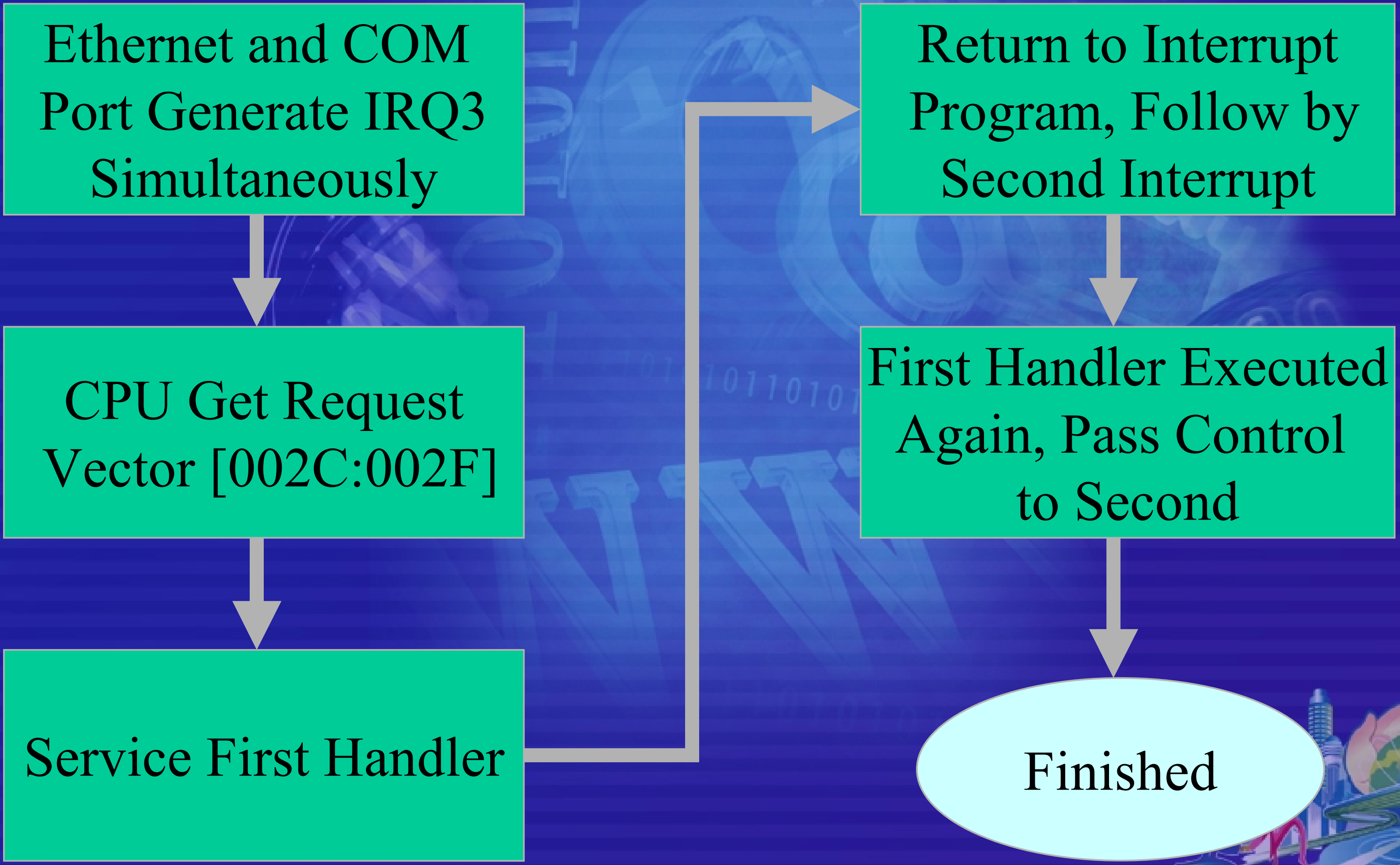
Interrupt Detected
CPU Generate 2 INTA# to request interrupt table entry number
Read Entry number from D0:D7
Index to Interrupt Table Save Flag, CS:IP
JMP to ISR Routine through new CS:IP
Execute Interrupt Service Routine
Issue EOI (End Of Interrupt)
Execute IRET (Restore CS:IP,Flag)
Interrupt Serviced

MSI™
Link to the Future

Build Interrupt Table


Initialize All Entries in Table to Null Value
Initialize All Entries For Embedded Device
Hook Entries for Embedded Device BIOS Routine
Perform Expansion Bus ROM Scan
Perform PCI Device Scan
Load Operation System
Interrupt Table initialized

Service Share Interrupt



Ethernet and COM Port Generate IRQ3 Simultaneously
CPU Get Request Vector [002C:002F]
Service First Handler
Return to Interrupt Program, Follow by Second Interrupt
First Handler Executed Again, Pass Control to Second
Finished

PCI Configuration Read Cycle

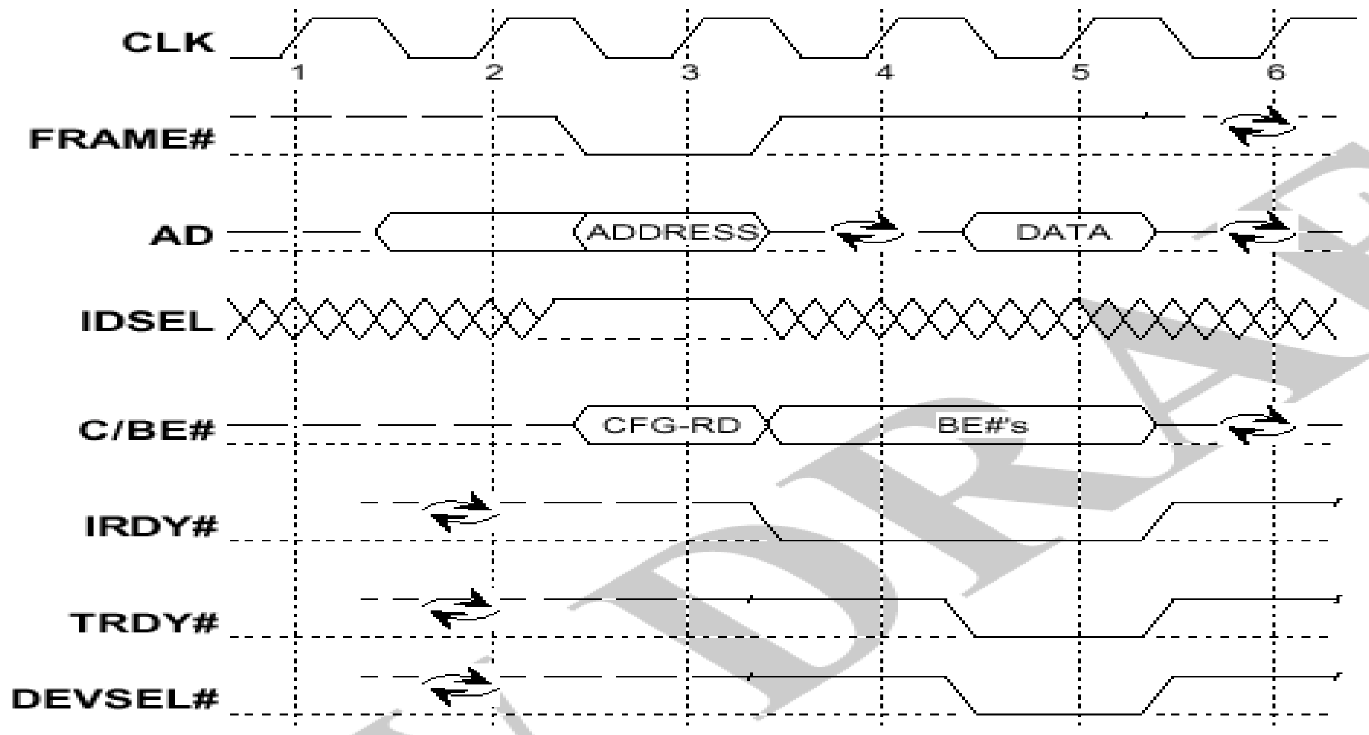


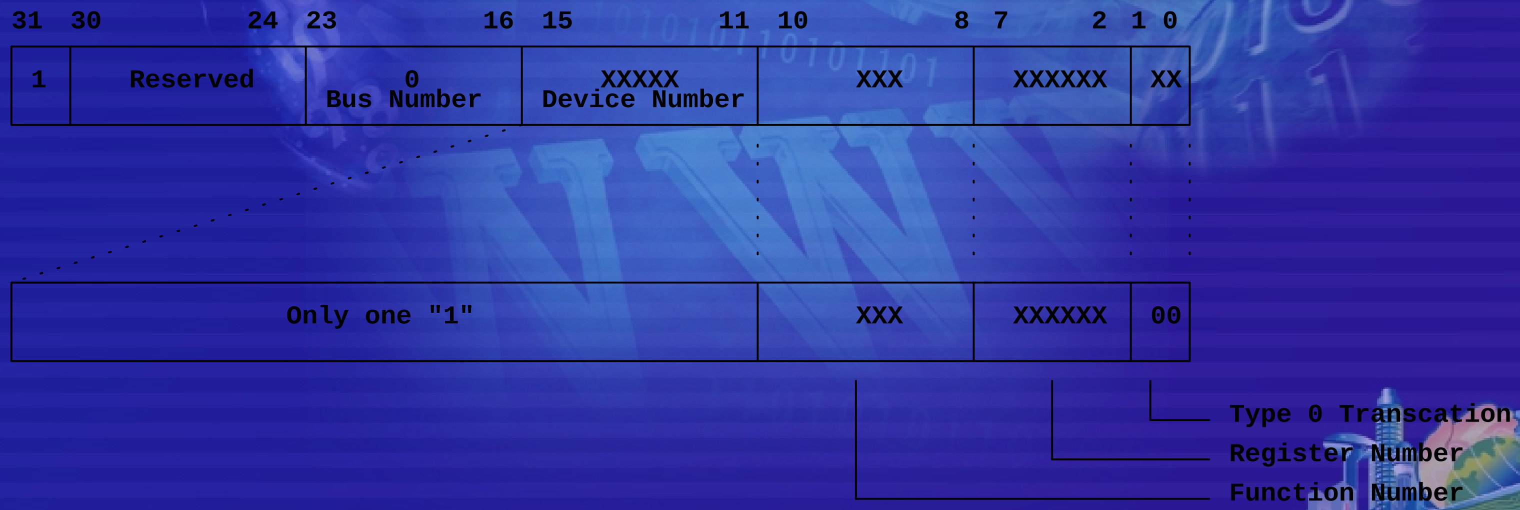
Figure 3-4: Configuration Read

Type 0 Translation

1.Access the PCI device on the Bus 0

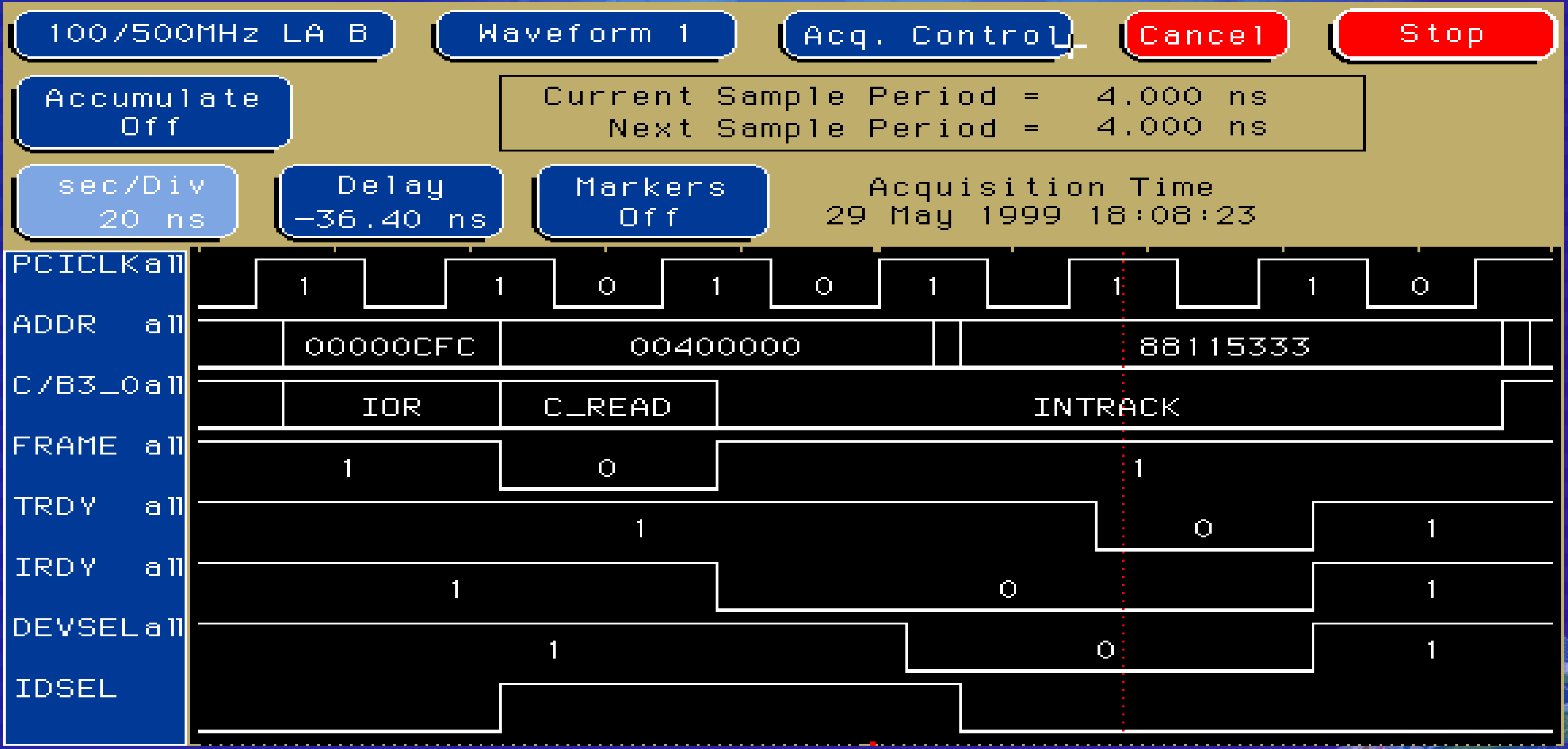
2.Translation Format :

Type 0 Configuration Translation :



Type 0 Translation – PCI Cycle Capture

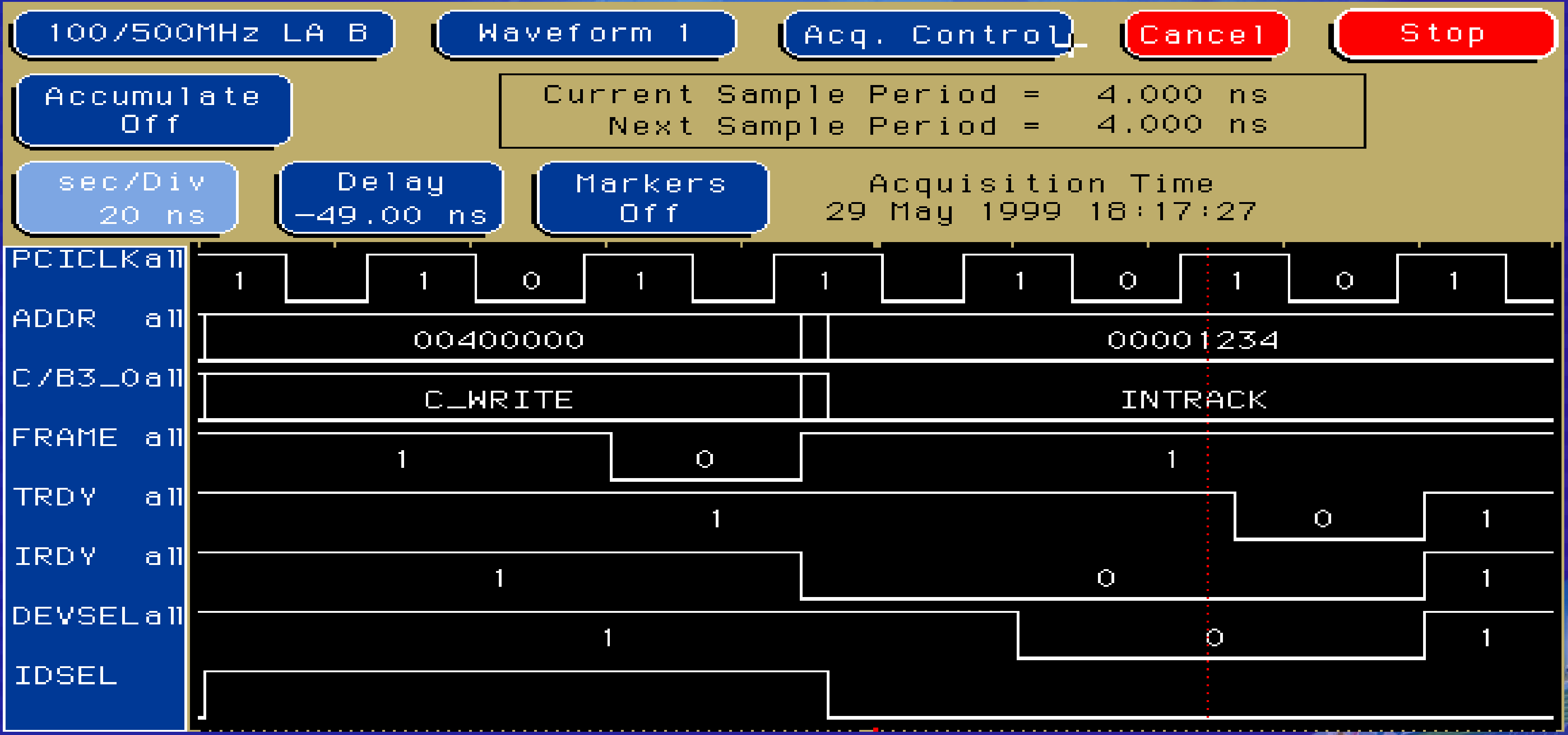
Target device = Bus 0h , Device 0Bh, Function 0h. Read S3 Trio64V+ PCI VGA Card. Configuration Double Word Read Cycle (Register 00-03h) on 82443EX Platform



```
Current Sample Period =   4.000 ns
   Next Sample Period =   4.000 ns

Acquisition Time
29 May 1999 18:08:23
```

Type 0 Translation – PCI Cycle Capture

Target device = Bus 0h , Device 0Bh, Function 0h. Read S3 Trio64V+ PCI VGA Card. Configuration Double Word Write Cycle (Register 00-03h,1234H) on 82443EX Platform

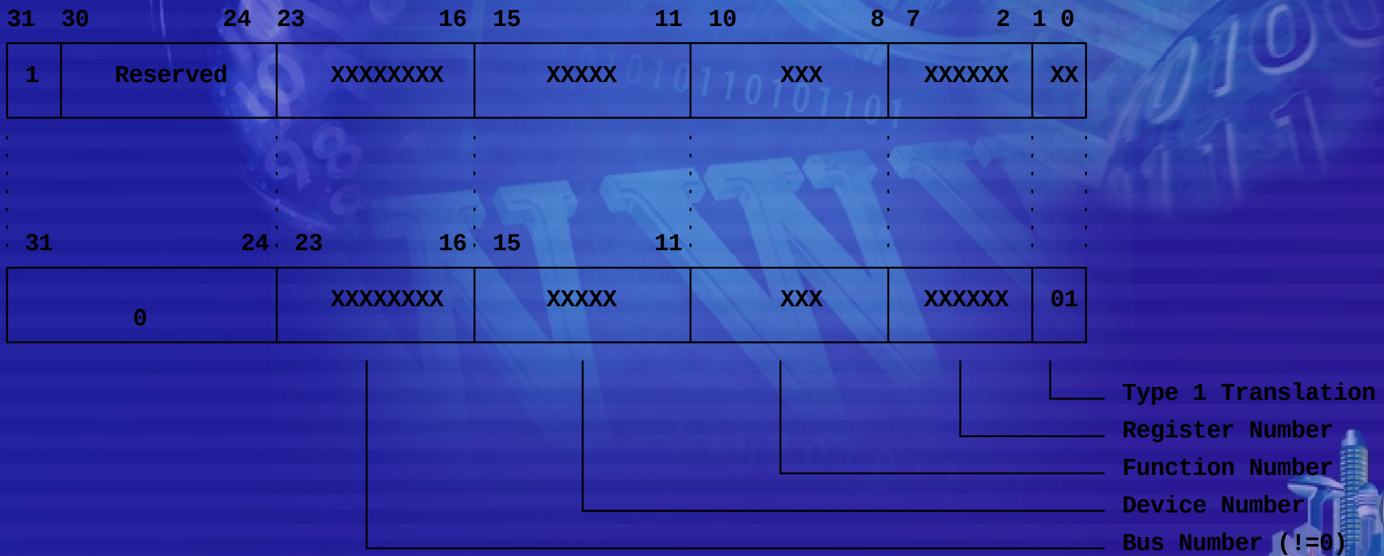


```
Current Sample Period =   4.000 ns
   Next Sample Period =   4.000 ns

    Acquisition Time
29 May 1999 18:17:27
```

Type 1 Translation

1. Access when the PCI bus number !=0
2. Translation Format :

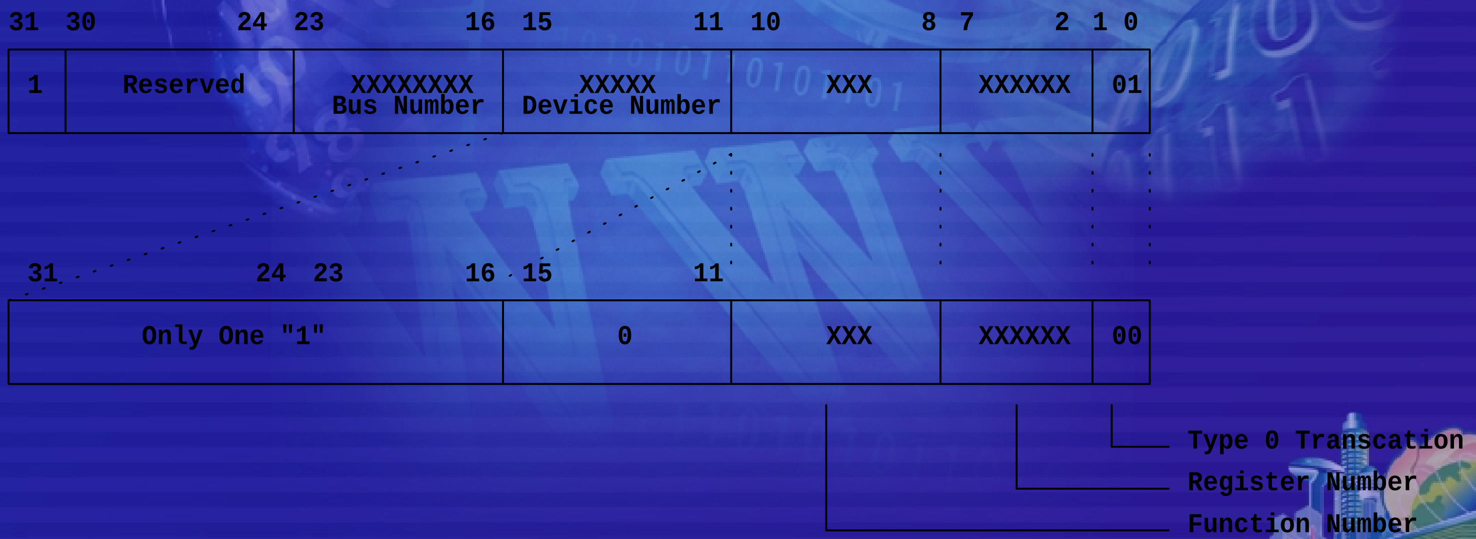
Type 1 Configuration Translation :



Type 1 to 0 Translation

1. *Translation behind the PCI to PCI Bridge*
2. *Translation Format :*

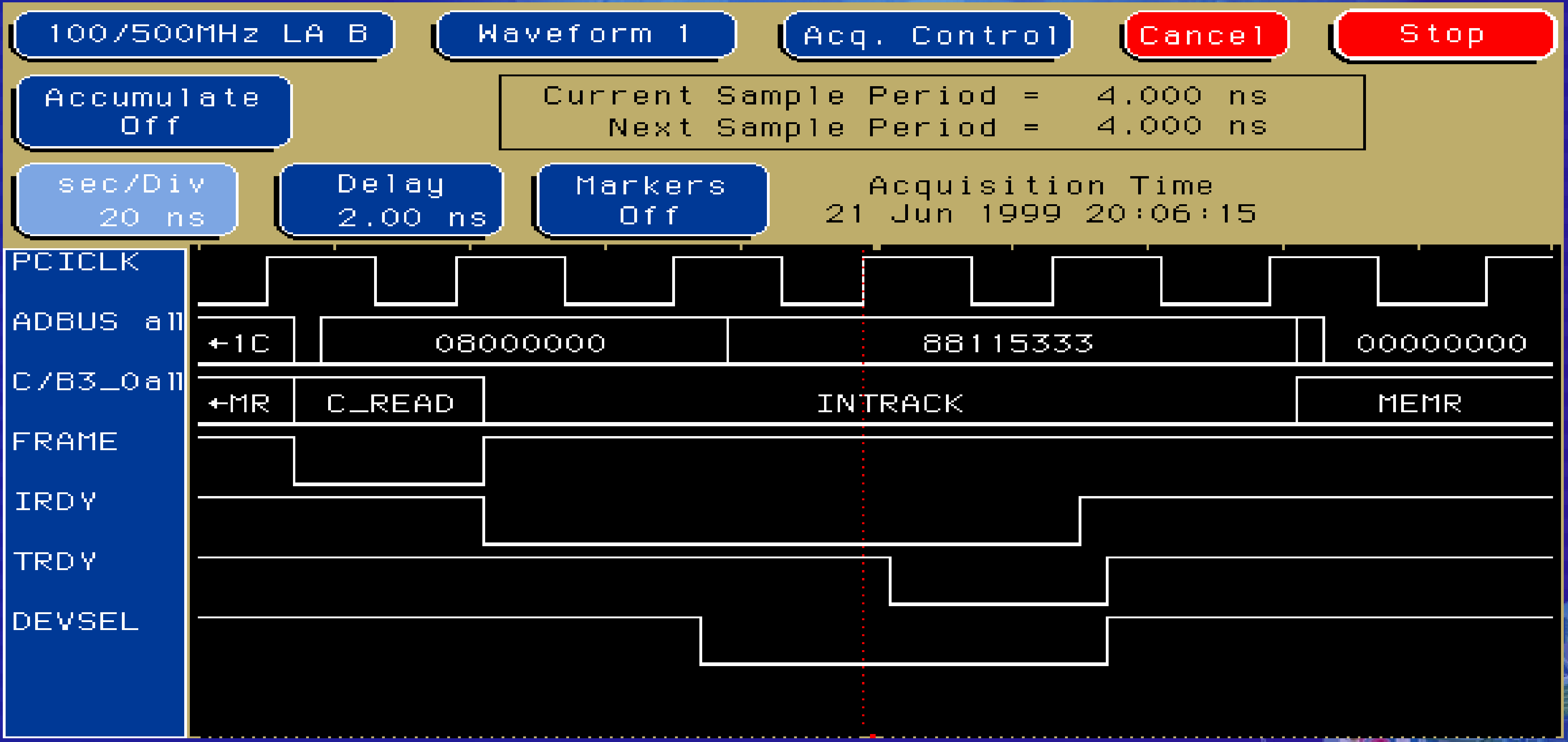
Type 1 to 0 Configuration Translation :



MSI™ Link to the Future

Type 1 to 0 Translation – PCI Cycle Capture

Target device = Bus 1h, Device 0Bh, Function 0h, Read S3 Trio64V+ PCI VGA Card. Double Word Configuration Read Cycle (Register 00-03h) on i810 Platform



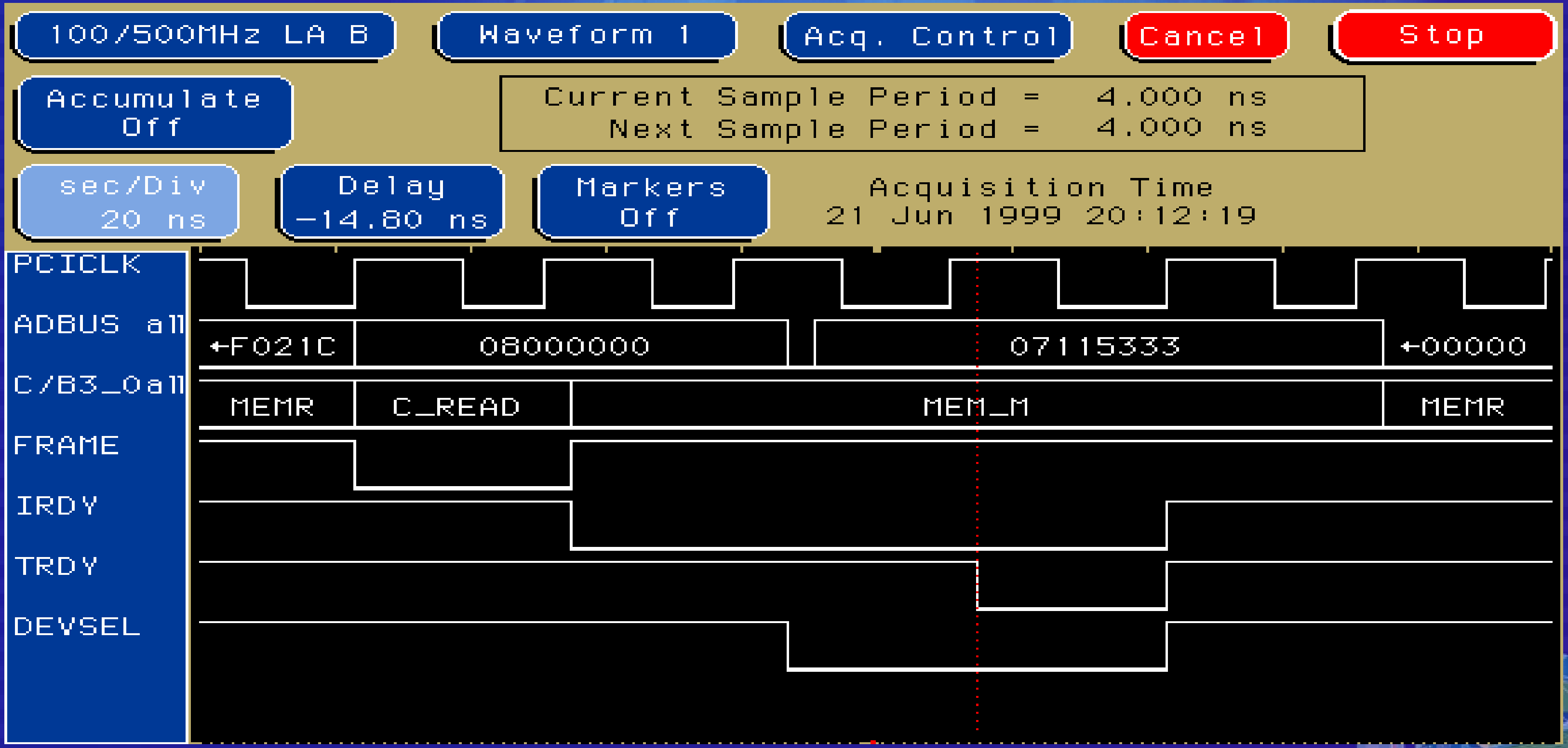
```
Current Sample Period =   4.000 ns
   Next Sample Period =   4.000 ns
```

Acquisition Time
21 Jun 1999 20:06:15

MSI
Link to the Future

Type 1 to 0 Translation – PCI Cycle Capture

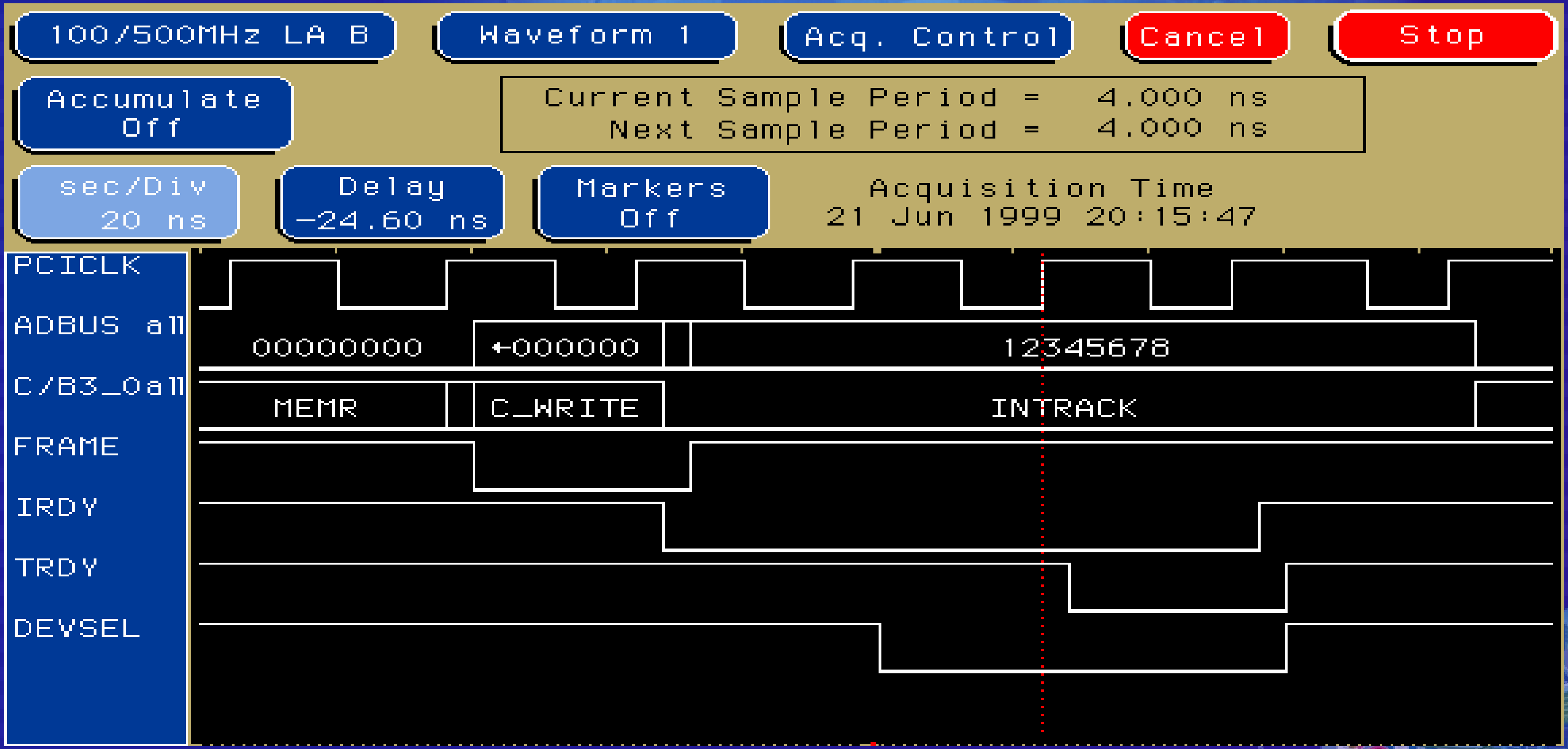
Target device = Bus 1h , Device 0Bh, Function 0h. Read S3 Trio64V+ PCI VGA Card. Word Configuration Read Cycle (Register 00-01h) on i810 platform



```
Current Sample Period =   4.000 ns
   Next Sample Period =   4.000 ns

    Acquisition Time
21 Jun 1999 20:12:19
```

Type 1 to 0 Translation – PCI Cycle Capture

Target device = Bus 1h , Device 0Bh, Function 0h. Read S3 Trio64V+ PCI VGA Card. Double Word Configuration Write Cycle (Register 00-03h) on i810 platform



```
Current Sample Period =   4.000 ns
   Next Sample Period =   4.000 ns
```

Acquisition Time
21 Jun 1999 20:15:47

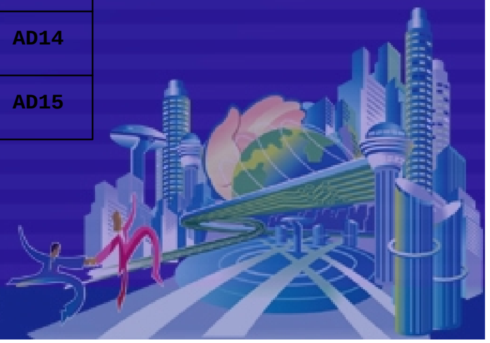
MSI
Link to the Future

Type 0 to 0 Translation (ICHx)

When a Type 0 configuration cycle is received on hub link,the ICH forwards these cycle on PCI bus and then reclaims them. The PCI ICHx uses address bits AD[15:14] to communicate the ICHx device number other than 30 or 31. The ICHx will not set any address bits in the range AD[31..11] during the corresponding translation on the PCI.

Type 0 to 0 Configuration Translation : (ICHx)

Device # in Hub Link Type 0 Cycle	AD[31:11] During Address Phase of Type 0 Cycle on PCI	
0 through 29	0000 0000 0000 0000 00000b	
30	0000 0000 0000 0000 01000b (Device Number = 01000b)	AD14
31	0000 0000 0000 0000 10000b (Device Number = 10000b)	AD15



Coding Example – PCI Configuration Read

```
ReadPCIReg    proc   bus:word,dev:word,func:word,reg:word
              xor       eax,eax
              mov       ax,bus
              or        ax,8000h      ; Bit31 Must Enable
              shl       eax,5
              or        ax,dev
              shl       eax,3
              or        ax,func
              shl       eax,8
              or        ax,reg
              mov       dx,PCI_ADDR_PORT
              out       dx,eax
              mov       dx,PCI_DATA_PORT
              in        eax,dx
              ov        dx,ax
              shr       eax,16
              xchg      dx,ax
              ret
ReadPCIReg    endp
```

Coding Example – PCI Configuration Write

```
WritePCIReg    proc    bus:word,dev:word,func:word,reg:word,ldata:dword
               xor         eax,eax
               mov         ax,bus
               or          ax,8000h       ;  Bit31 Must Enable
               shl         eax,5
               or          ax,dev
               shl         eax,3
               or          ax,func
               shl         eax,8
               or          ax,reg
               mov         dx,PCI_ADDR_PORT
               out         dx,eax
               mov         eax,ldata
               mov         dx,PCI_DATA_PORT
               out         dx,eax
               ret
WritePCIReg    endp
```

PCI Configuration Mechanism #2

1.PCI specification : "Configuration mechanism # 1 is the preferred implementation and must be provide by all future host bridge (and exiting bridges should convert if possible). Configuration mechanism # 2 is defined for backward compatibility and must not be used by new design. Host bridge implemented in PC-AT compatible systems must implement at least one of these mechanisms“

2.Basic Configuration Mechanism

A.Configuration space enable,or CSE,register at I/O port 0CF8h

B.Forward register I/O port at I/O port 0CFAh

3. Program Configuration Register :

A.Write the target bus number (00h to ffh) into forward register 0CFAh

B.Write a one byte value to CSE register at 0CF8h

4.Configuration Space Enable, or CSE , Register

	Bit 7 4	3 1	0
CSE	Key Filed	Function Number	SCE

Bit [7..4] = Key filed : non-zero = allow configuration
Bit [3..1] = Target Function Number
Bit [0]   = Special Cycle Enable : 1 = Enable

5.Perform a one,two,or four byte I/O read or write translation within

the I/O range C000h through CFFFFh

Cxyz -> C : mean PCI Configuration space

x : physical package use by bridge to select an IDSEL signal

line to activate

yz : target configuration double word

6. I/O Sub Range within C000h through CFFFh I/O range :

I/O Sub-Range	Target Phayscal PCI package
C000h - C0FFh	0d
C100h - C1FFh	1d
C200h - C2FFh	2d
C300h - C3FFh	3d
C400h - C4FFh	4d
C500h - C5FFh	5d
C600h - C6FFh	6d
C700h - C7FFh	7d
C890h - C8FFh	8d
C900h - C9FFh	9d
CA00h - CAFFh	10d
CB00h - CBFFh	11d
CC00h - CCFFh	12d
CD00h - CDFFh	13d
CE00h - CEFFh	14d
CF00h - CFFFh	15d

PCI BIOS

- *Purpose of PCI BIOS*
  - *–An access between OS/AP and H/W platform*
  - *–Interface to access configuration register*
  - *–Interface to access interrupt routing logic*
  - *–Interface to get the ability of the H/W platform, like PCI chipset*
  - *–INT 1Ah for real mode entry*

MSI™
Link to the Future

PCI BIOS Function Call

Function Request	AH	AL
Test for PCI Present	B1h	01h
Find PCI Device		02h
Find PCI Class Code		03h
Generate Special Cycle		06h
Read Configuration Byte		08h
Read Configuration Word		09h
Read Configuration Double word		0Ah
Write Configuration Byte		0Bh
Write Configuration Word		0Ch
Write Configuration Double word		0Dh
Get Interrupt routing options		0Eh
Set PCI Interrupt		0Fh

PCI BIOS Function Call

Read Double word :

This function allows reading individual dwords from the configuration Space of a specific device. The Register Number parameter must be a Multiple of four (i.e., bits 0 and 1 must be set to 0).

Entry :

[AH] PCI Function ID
[AL] Read Config DWord
[BH] Bus Number (0...255)
[BL] Device Number in upper 5 bits,Function Number in lower 3 bits
[DI] Register Number (0,4,8,...252)

Exit :

[ECX] Dword read.
[AH] Return Code:
Successful
Bad Register Number
[CF] Complption Status, set = error, reset = success

Return Codes [AH] :

00h Successful	81h Func not support
83h Bad VendorID	86h Device NotFound
87h Bad Register Number	88h Set failed
89h Buffer too small	

Coding Example – PCI Configuration Read by PCI BIOS call

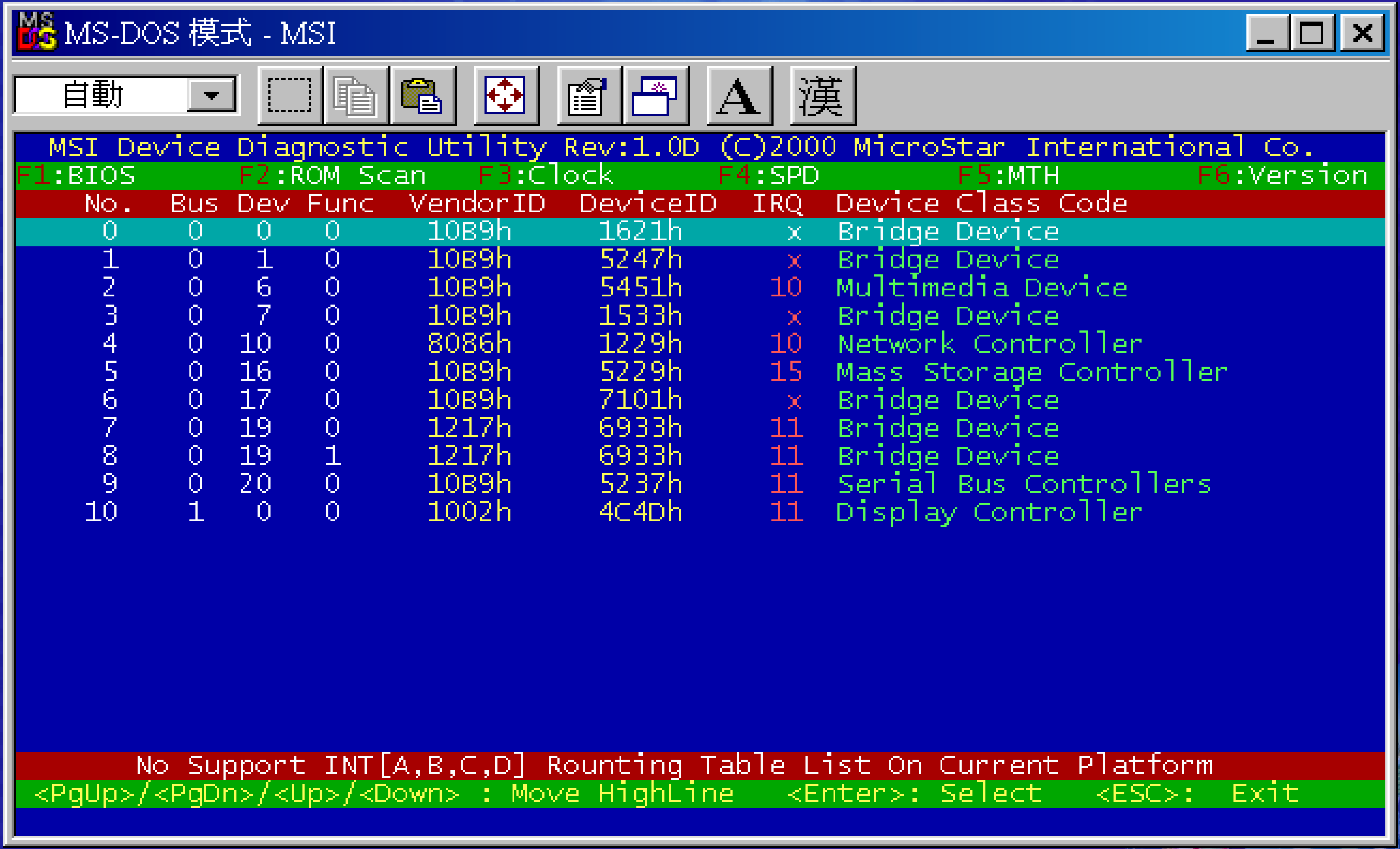
```
ReadPCIReg      proc    bus:word,dev:word,func:word,reg:word

        cli

        mov ah,0b1h             ;; Set ah = b1h
        mov al,0ah              ;; Set al = 0ah
        mov dx,bus
        mov bh,dl               ;; Set bh = bus number
        mov dx,dev
        mov bl,dl               ;; Set bl = device + function number
        shl bl,3h               ;;
        mov dx,func
        add bl,dl
        mov di,reg              ;; Set di = register number
        int 01ah                ;; Call PCI BIOS function call 1Ah
        mov ax,cx               ;; Return value on ECX and adjust to
        shr ecx,16              ;; AX and DX
        mov dx,cx               ;;
        sti
        ret

    ReadPCIReg      endp
```

MSI™
Link to the Future

PCI Configuration Tools

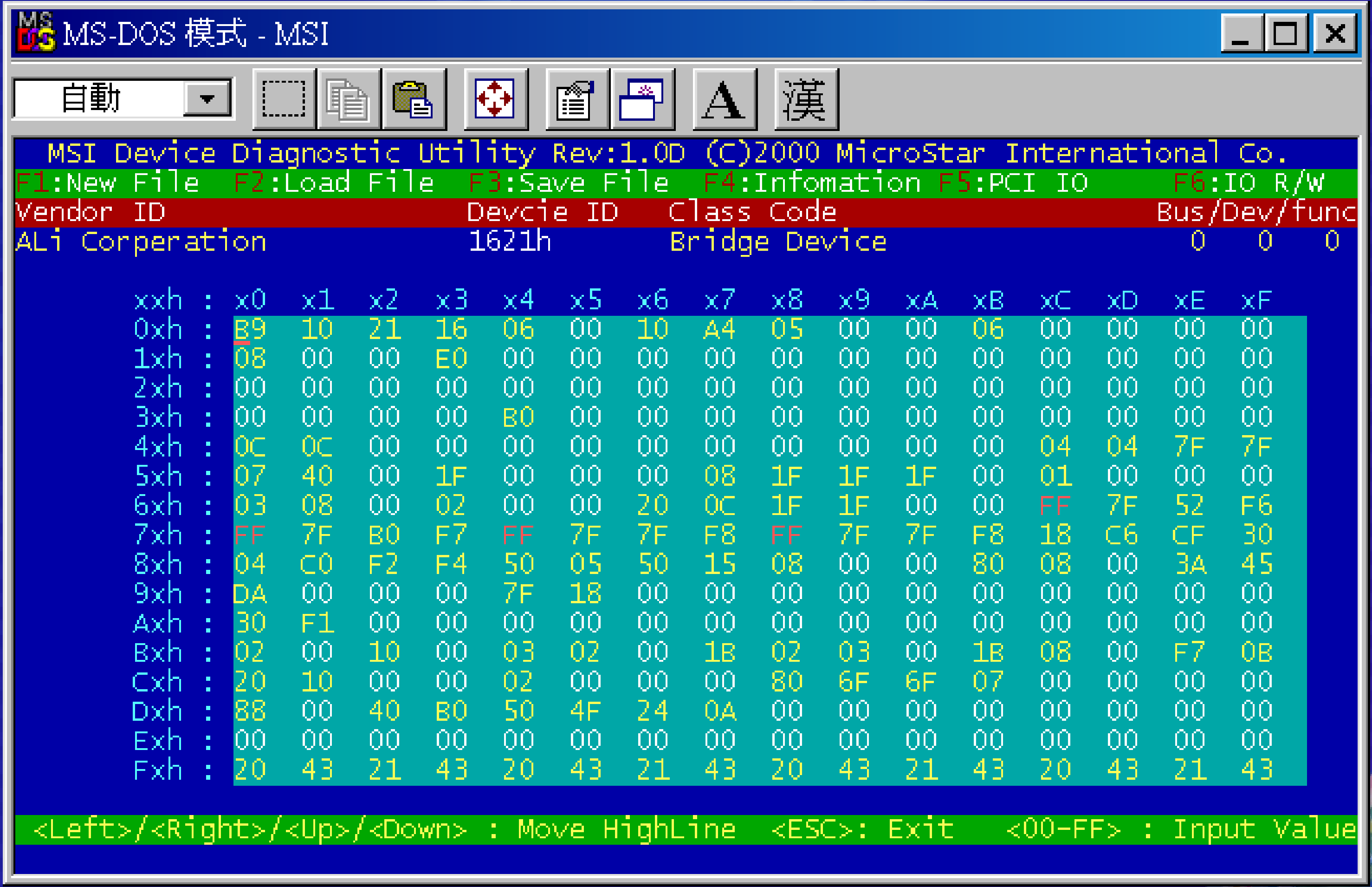
1. Intel – PCI31.EXE
2. AMI – RU.EXE
3. Award – PCI.EXE
4. MSI – MSI.EXE

PCI Configuration Tools - Demo



```
 MSI Device Diagnostic Utility Rev:1.0D (C)2000 MicroStar International Co.
F1:BIOS      F2:ROM Scan    F3:Clock      F4:SPD        F5:MTH       F6:Version
    No.   Bus Dev Func  VendorID  DeviceID  IRQ  Device Class Code
     0     0   0   0     10B9h     1621h     x   Bridge Device
     1     0   1   0     10B9h     5247h     x   Bridge Device
     2     0   6   0     10B9h     5451h    10   Multimedia Device
     3     0   7   0     10B9h     1533h     x   Bridge Device
     4     0  10   0     8086h     1229h    10   Network Controller
     5     0  16   0     10B9h     5229h    15   Mass Storage Controller
     6     0  17   0     10B9h     7101h     x   Bridge Device
     7     0  19   0     1217h     6933h    11   Bridge Device
     8     0  19   1     1217h     6933h    11   Bridge Device
     9     0  20   0     10B9h     5237h    11   Serial Bus Controllers
    10     1   0   0     1002h     4C4Dh    11   Display Controller

       No Support INT[A,B,C,D] Rounting Table List On Current Platform
 <PgUp>/<PgDn>/<Up>/<Down> : Move HighLine   <Enter>: Select   <ESC>:  Exit
```

PCI Configuration Tools - Demo



```
 MSI Device Diagnostic Utility Rev:1.0D (C)2000 MicroStar International Co.
F1:New File  F2:Load File  F3:Save File  F4:Infomation F5:PCI IO    F6:IO R/W
Vendor ID                             Devcie ID    Class Code             Bus/Dev/func
ALi Corperation                       1621h        Bridge Device             0    0    0

          xxh : x0  x1  x2  x3  x4  x5  x6  x7  x8  x9  xA  xB  xC  xD  xE  xF
          0xh : B9  10  21  16  06  00  10  A4  05  00  00  06  00  00  00  00
          1xh : 08  00  00  E0  00  00  00  00  00  00  00  00  00  00  00  00
          2xh : 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
          3xh : 00  00  00  00  B0  00  00  00  00  00  00  00  00  00  00  00
          4xh : 0C  0C  00  00  00  00  00  00  00  00  00  00  04  04  7F  7F
          5xh : 07  40  00  1F  00  00  00  08  1F  1F  1F  00  01  00  00  00
          6xh : 03  08  00  02  00  00  20  0C  1F  1F  00  00  FF  7F  52  F6
          7xh : FF  7F  B0  F7  FF  7F  7F  F8  FF  7F  7F  F8  18  C6  CF  30
          8xh : 04  C0  F2  F4  50  05  50  15  08  00  00  80  08  00  3A  45
          9xh : DA  00  00  00  7F  18  00  00  00  00  00  00  00  00  00  00
          Axh : 30  F1  00  00  00  00  00  00  00  00  00  00  00  00  00  00
          Bxh : 02  00  10  00  03  02  00  1B  02  03  00  1B  08  00  F7  0B
          Cxh : 20  10  00  00  02  00  00  00  80  6F  6F  07  00  00  00  00
          Dxh : 88  00  40  B0  50  4F  24  0A  00  00  00  00  00  00  00  00
          Exh : 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
          Fxh : 20  43  21  43  20  43  21  43  20  43  21  43  20  43  21  43

 <Left>/<Right>/<Up>/<Down> : Move HighLine  <ESC>: Exit   <00-FF> : Input Value
```

MSI™ Link to the Future

Reference Document

1.PCI System Architecture Fourth Edition

2.PCI BIOS Specification 2.1

3.PCI Local Bus Specification 2.2

4.Intel 82802BA IO Controller Hub 2 (ICH2) Datasheet

Q & A

MSI™
Link to the Future

Thank You!!

Wish you have a nice day!!