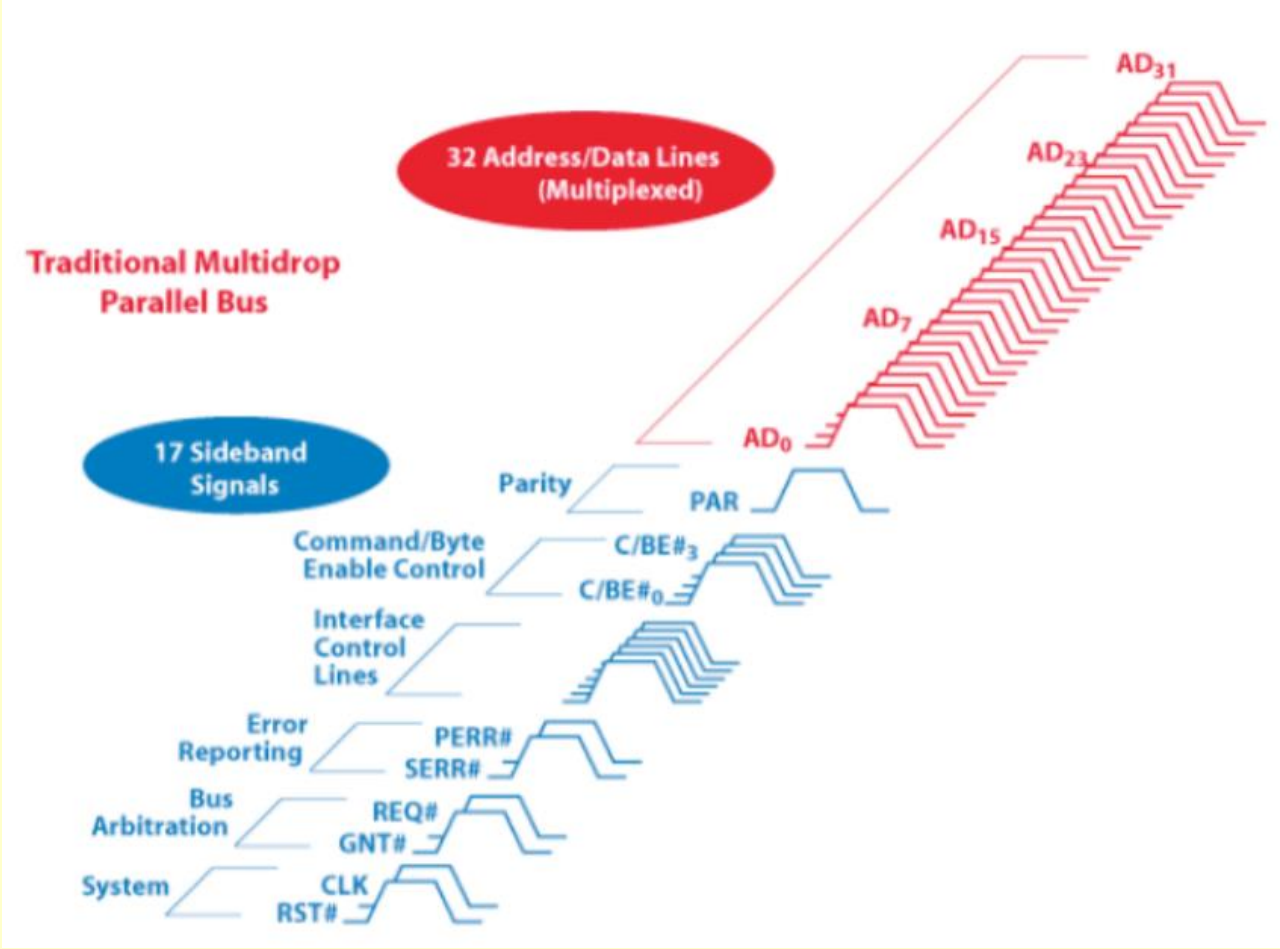
C11. MAGISTRALE IN PC

1. PCI, PCI-X
2. Tema



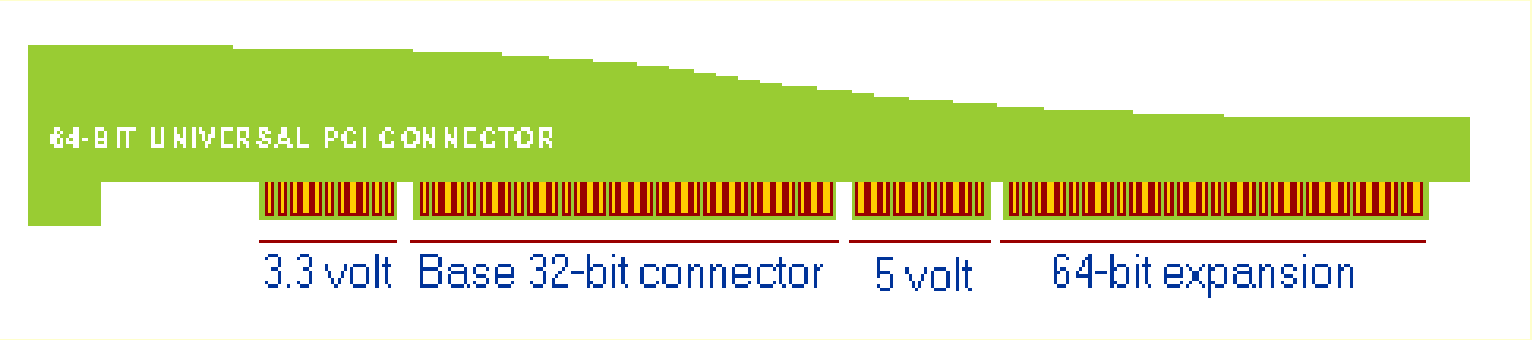
1. PCI - Peripheral Component Interconnection

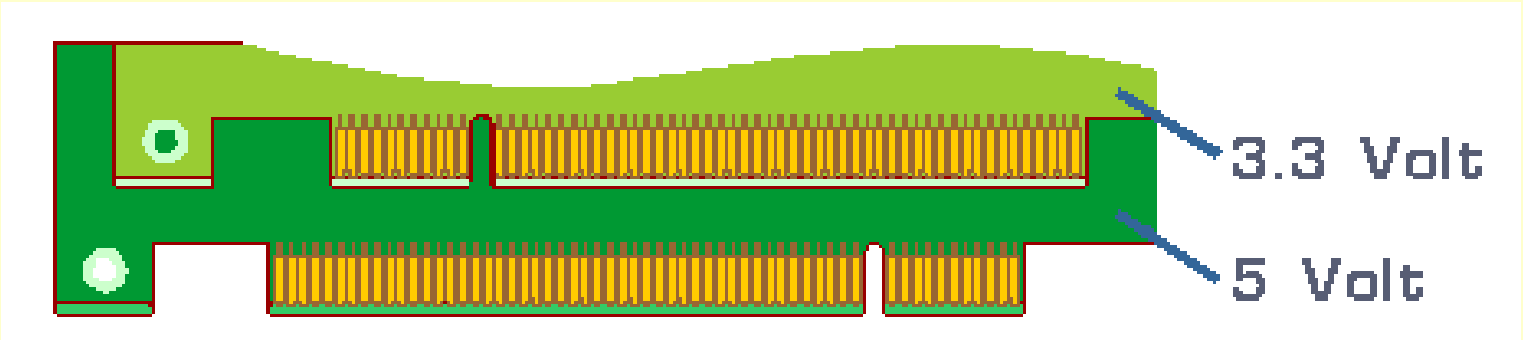
- Intel- Pentium (1993)

-IC Interface -82430 PCI chip set

- fast access memory, HD, video

-http://computer.howstuffworks.com/pci2.htm





http://what-when-how.com/Tutorial/topic-736imk5dv/Computer-Busses-bookmark.html

Bus Type	MB/sec
VL-bus	100 MBps
VL-bus	132 MBps
32-Bit PCI	132 MBps
PCI-X 66	512 MBps
PCI-X 133	1 GBps
AGP x1	264 MB/s
AGP x2	528 MB/s
AGP x4	1056 MB/s
AGP x8	2112 MB/s
PCI Express x1	500 MB/s
PCI Express x2	1000 MB/s
PCI Express x4	2000 MB/s
PCI Express x8	4000 MB/s
PCI Express x12	6000 MB/s
PCI Express x16	8000 MB/s

1.1. Bus-ul PCI - istoric

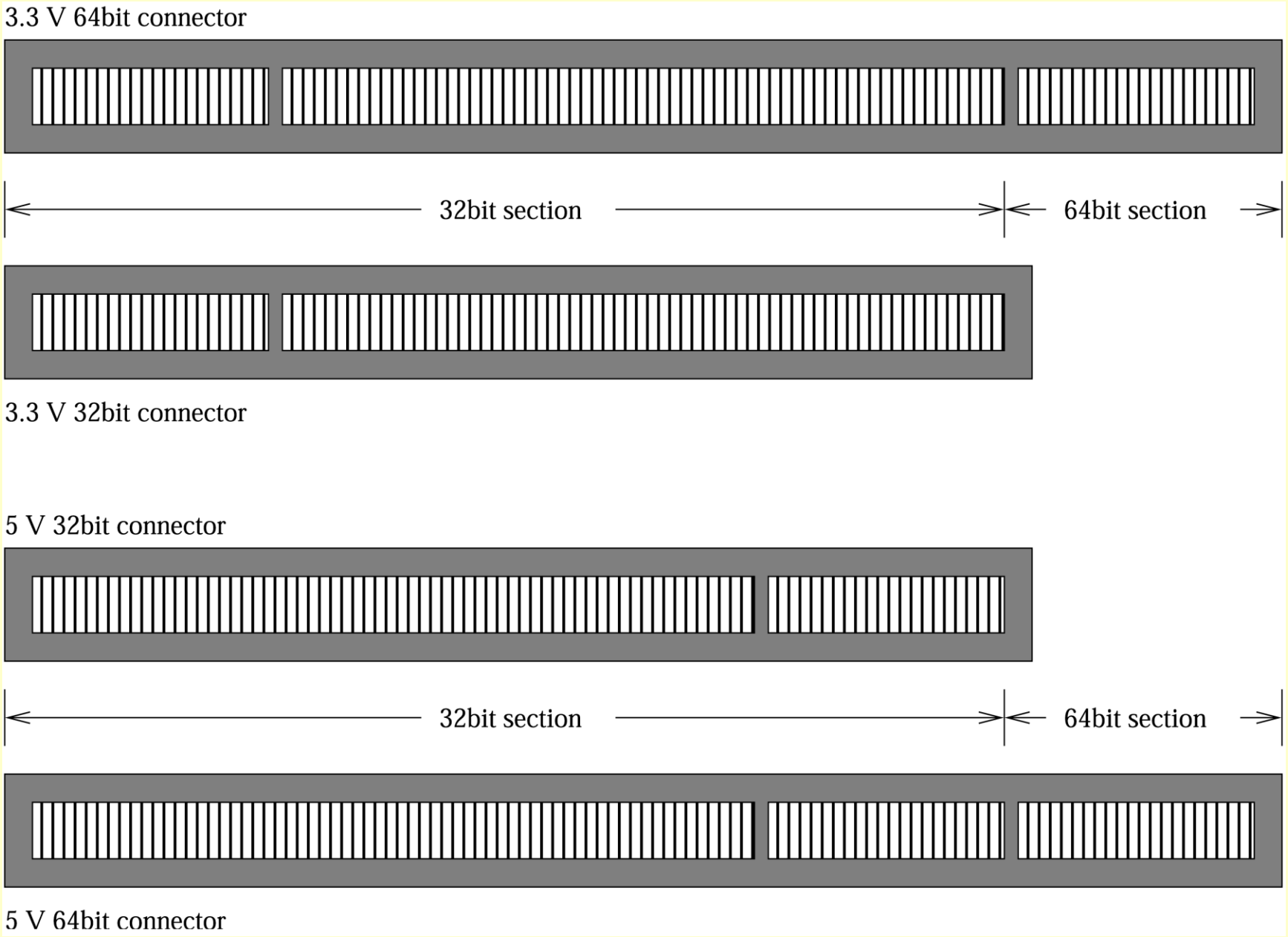
- Proiectarea a început în **1990**
  - Intel a inceput sa lucreze la un nou bus pentru sistemele lor Pentium, independent de processor, pentru a sprijini cerințele de lățime de bandă mai mari ptr. sistemele de operare multitasking (bazate pe ferestre)

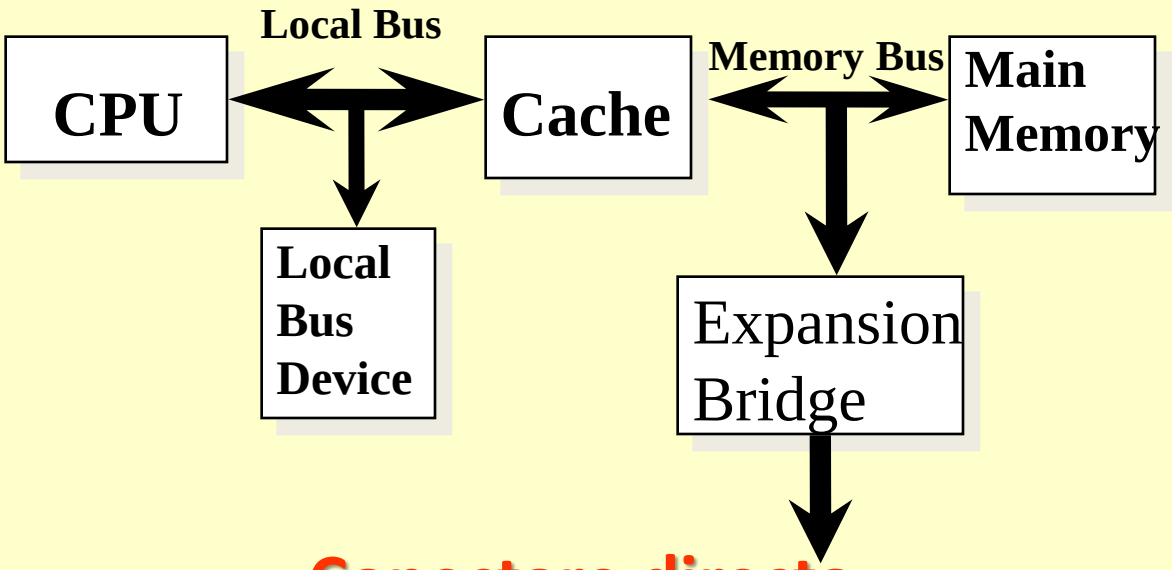
  - Versiunea originala (1.0), dezvoltata de Intel în 1990
  - Versiunea 2.0 în 1993
  - Versiunea 2.1 în 1995
  - Versiunea 2.2 de gestionare a energiei s-a introdus pentru sistemele mobile
  - Versiunea 3.0 (în aprilie 2004)

PCI Bus (cont.)

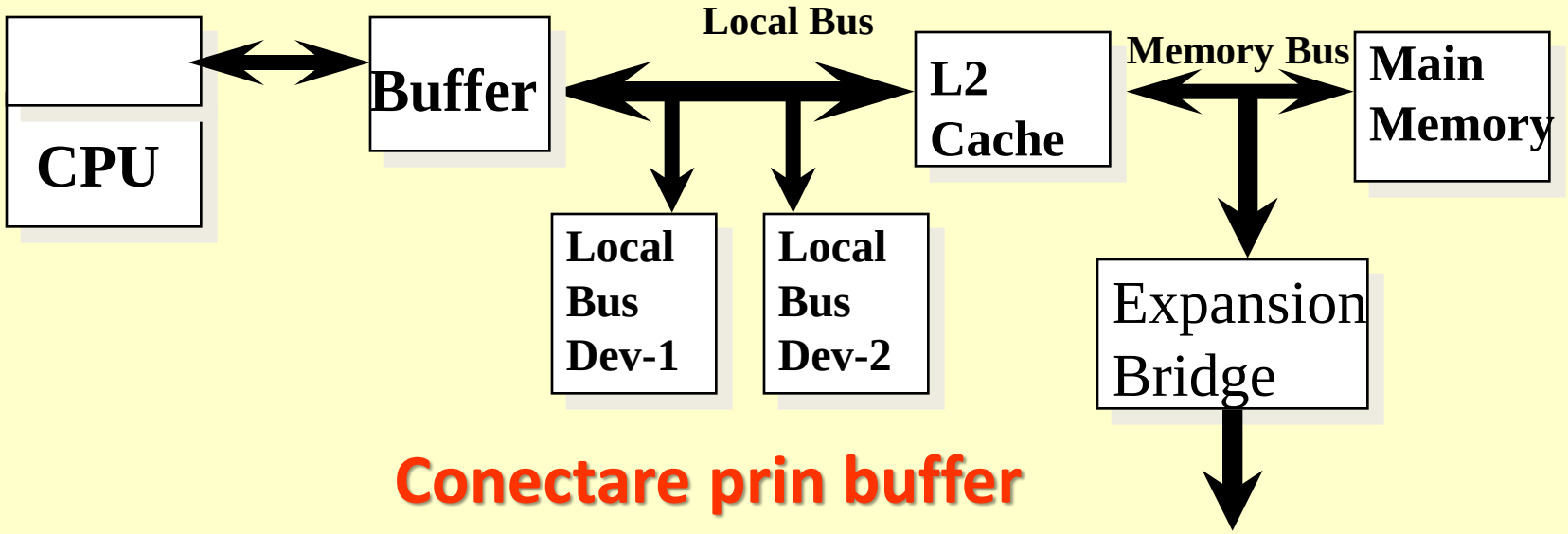
- PCI este folosit pe 32-biti/64-biți
- Funcționeaza la 33 MHz/66 MHz
- Alimentare la 5 V (cartele mai vechi) sau 3,3 V (cele mai noi)
- Pe 32-biti PCI de opereaza la 33 MHz, oferă lățime de bandă de vârf de 133 MB/s, iar pe 64-biti PCI operand la 66 MHz oferă lățime de bandă de vârf de 528 MB/s

PCI Bus (cont.)


3.3 V 64bit connector
32bit section
64bit section
3.3 V 32bit connector
5 V 32bit connector
32bit section
64bit section
5 V 64bit connector



Conectare directa



Conectare prin buffer

1.2. PCI - terminologie

- **PCI Inițiator** – Master-ul actual al Bus-ului, care inițiază un transfer
  - Poate controla bus-ul
  - Inițiază transferul
  - poate acționa, de asemenea, ca o țintă (destinatie)
  - Nu include în mod necesar un arbitru

- **PCI Tinta** - Slave - adresat de către inițiator
  - Nu poate controla bus-ul
  - Doar răspunde la inițiator (citire/scriere)

- **Agenți PCI** - Inițiatorul și ținta sunt menționati ca agenți

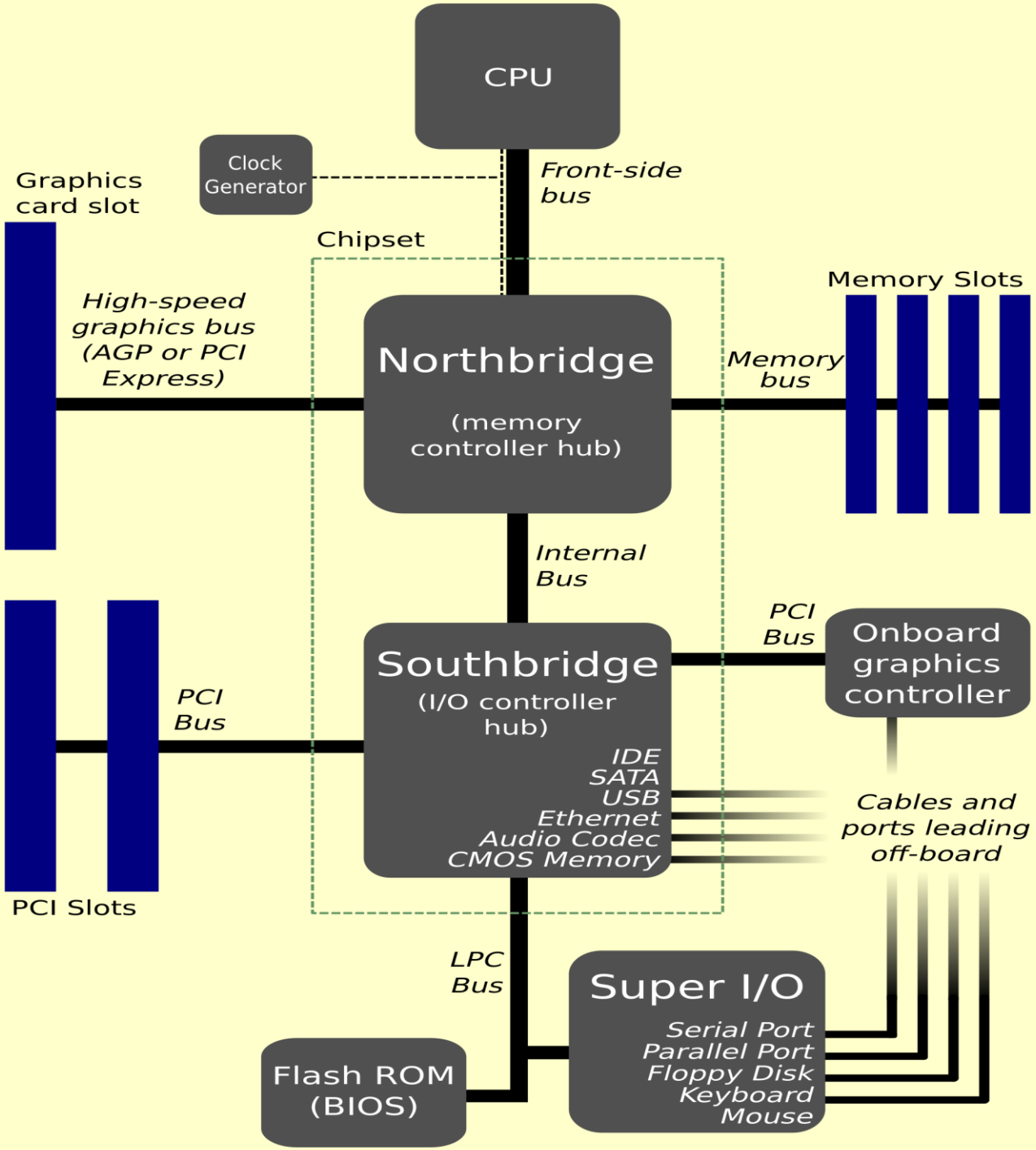
- **Multiplexarea bus-ului adrese/date** reduce numărul de pini ai bus-ului

- Penalizarea este de mai multi cicli de ceas/transfer (2 cicluri/scriere sau 3 cicluri/citire)
    - *Primul ciclu: Faza Adresa*
    - *Al doilea ciclu: de scriere a datelor*
    - *Al treilea ciclu: Citire date*

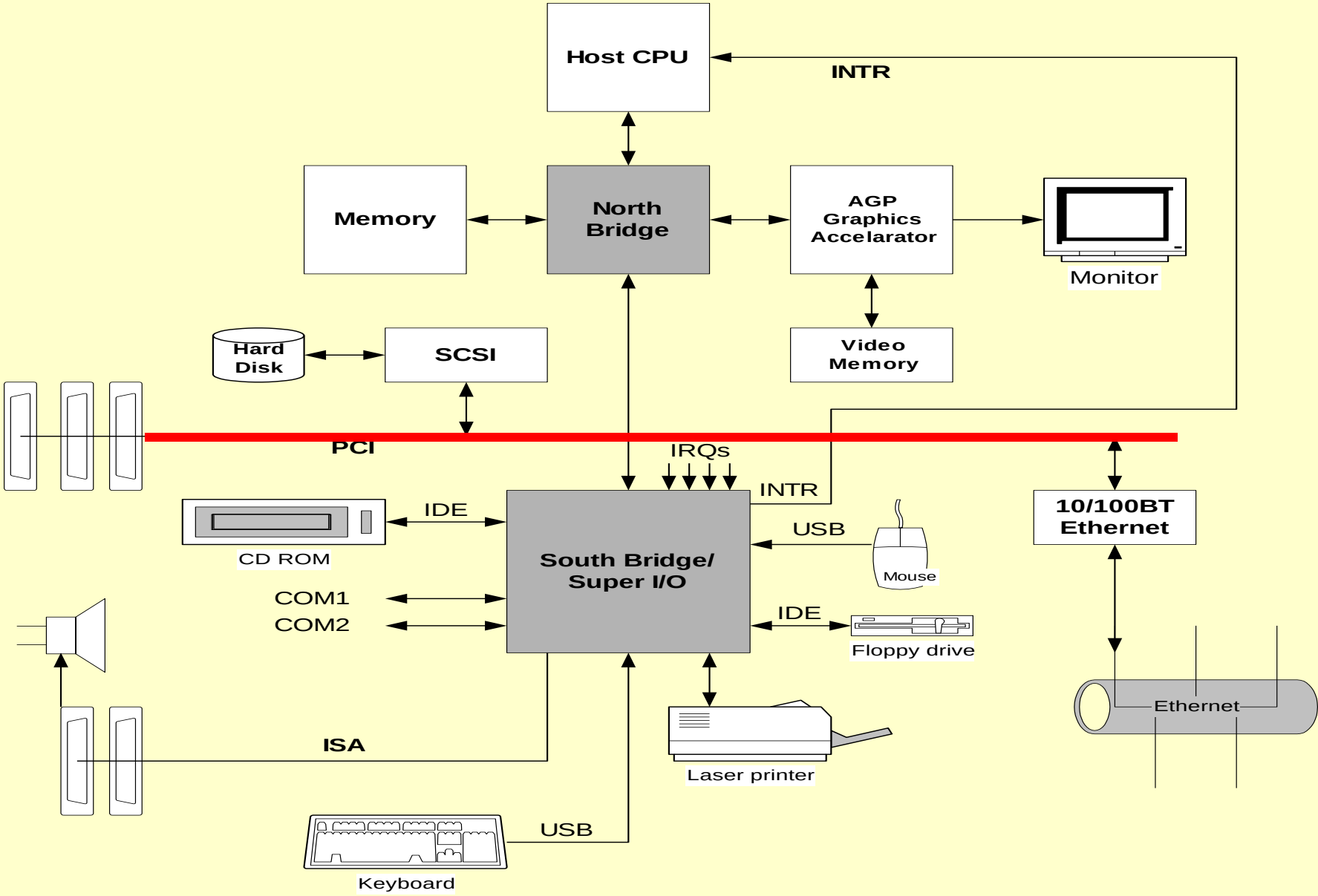
- **Conceptul Burst**

- un ciclu de adresă + mai multe cicluri de date
- adresele sunt secvențiale
- tinta are un contor intern de adresă

- La un ceas de 33MHz și 32-biti magistrala de date, asigură rată maxima de scriere date = 66MB/s, iar rata maximă de citire date = 44MB / s


CPU
Clock Generator
Front-side bus
Graphics card slot
Chipset
High-speed graphics bus (AGP or PCI Express)
Northbridge
(memory controller hub)
Memory Slots
Memory bus
Internal Bus
PCI Bus
Southbridge
(I/O controller hub)
IDE
SATA
USB
Ethernet
Audio Codec
CMOS Memory
PCI Bus
Onboard graphics controller
Cables and ports leading off-board
PCI Slots
LPC Bus
Super I/O
Serial Port
Parallel Port
Floppy Disk
Keyboard
Mouse
Flash ROM (BIOS)

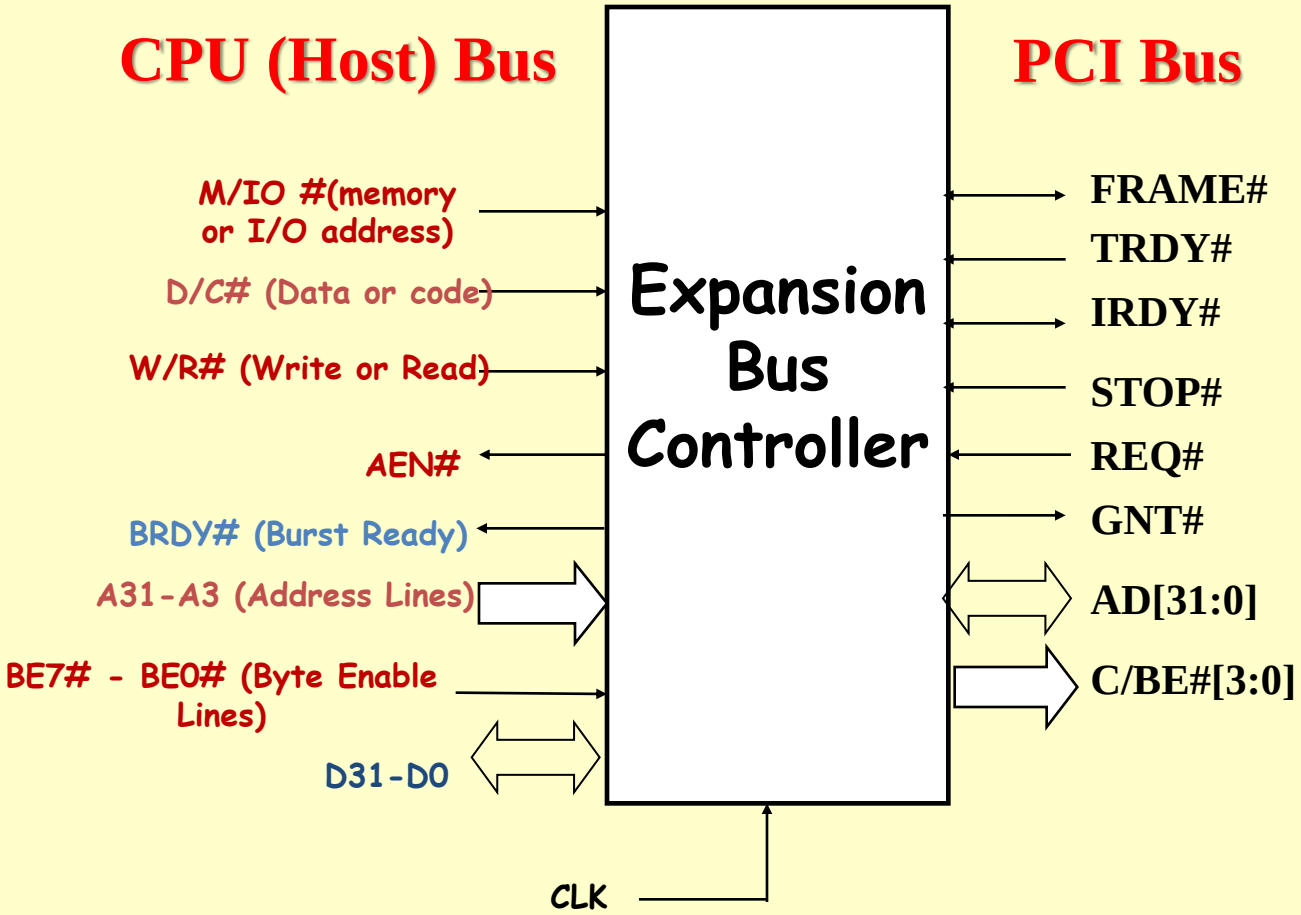
1.3. Structura Bus-ului PCI



Schema unui PC raportata la bus-ul PCI

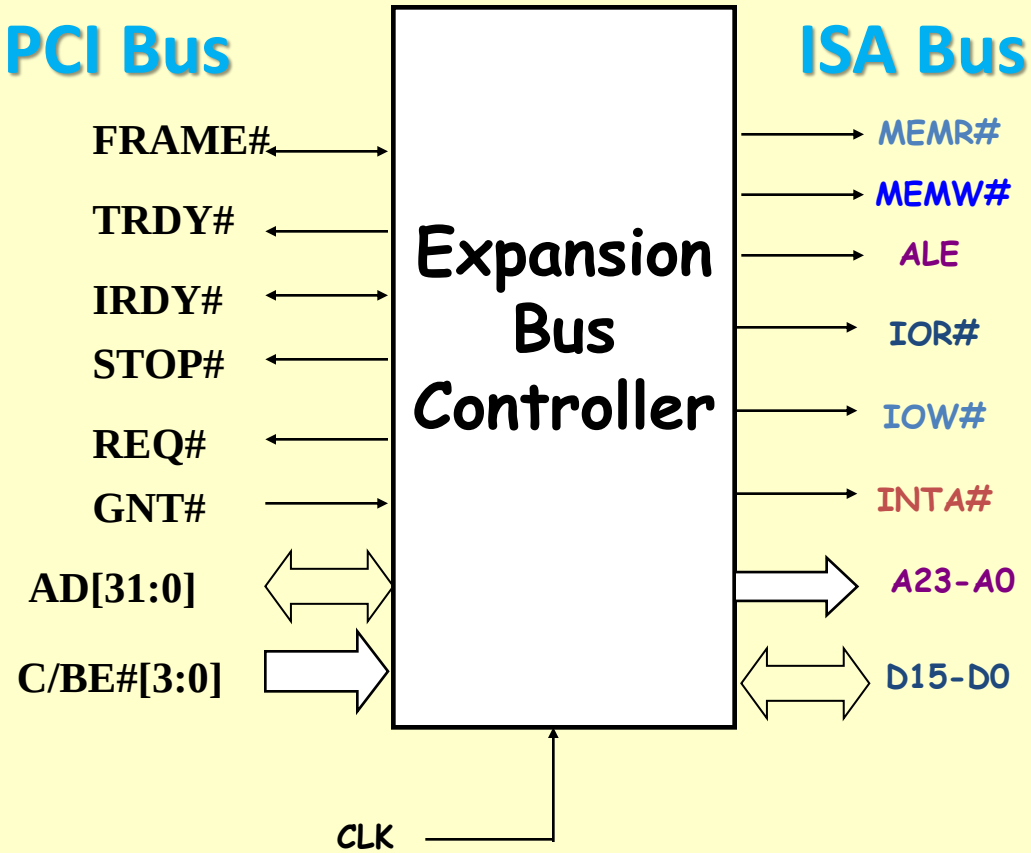
Structura Bus-ului

NORTH BRIDGE

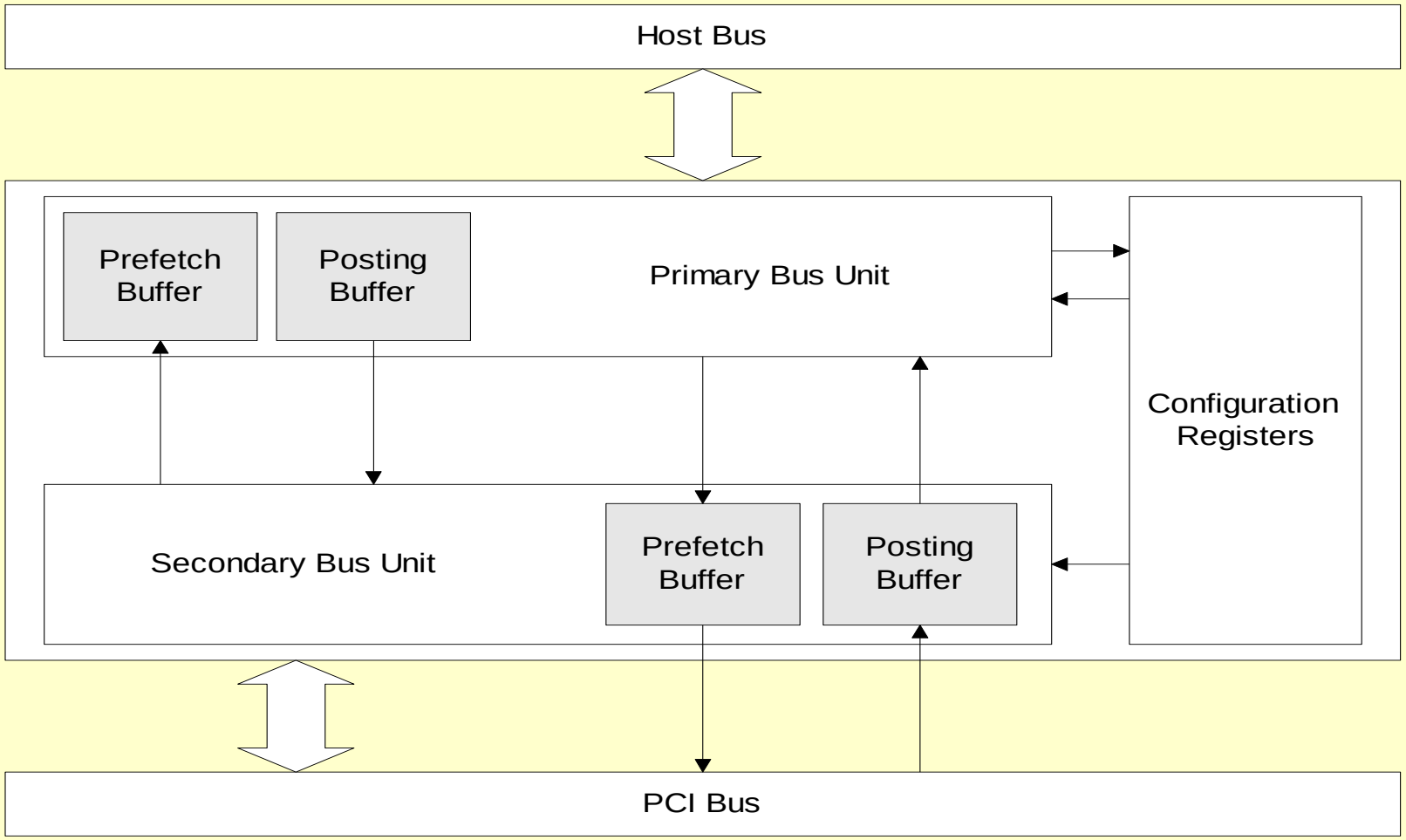


Structura Bus-ului

SOUTH BRIDGE



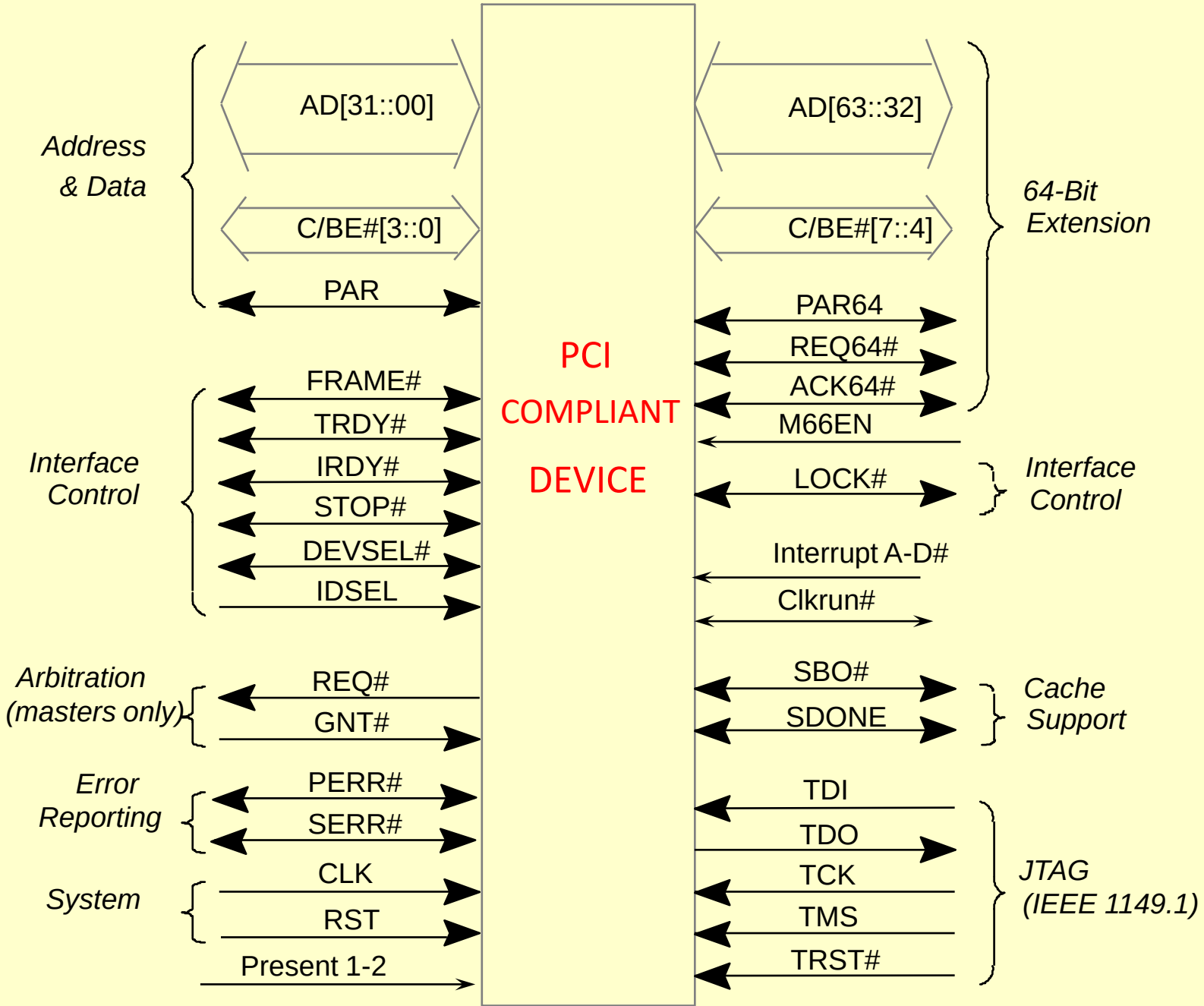
Schema bloc a puntii PCI



Typical Layout of a PCI Bridge

- Prefetch Buffers se folosesc ptr. citire
- Posting Buffers stocheaza datele ptr. scriere ptr. a le posta pe magistrala adresata mai tarziu

1.4 Semnale Bus PCI



Semnalele Bus-ului PCI

- System
  - CLK
  - RST#
- Address Data
  - AD[31:00]
  - C/BE[3:0]#
  - PAR
- Error Reporting
  - SERR#
  - PERR#
- Interface Control
  - FRAME#
  - IRDY#
  - TRDY#
  - STOP#
  - LOCK#
  - IDSEL#
  - DEVSEL#
- Arbitration
  - REQ#
  - GNT#

- *PCI este Bus Multimaster*
- *Tranzactiile initiate de master*
- *Tranzactiile se fac la/de la target*

Semnale Sistem

- **CLK (in)**
  - Intrare la toate dispozitivele PCI
  - Semnalele eșantionate pe *front pozitiv,* cu excepția RST # și INTx#
  - 0 - 33MHz, permite - extinderea la 66MHz

- **RST# (in)**
  - intrare asincrona
  - Trece ieșirile in 3-state
  - Aduce registrele de configurare și mașina de stare in stari cunoscute

Adrese si Date

- **AD[31:0] (t/s)**
  - Bus Multiplexat, contine adr. de start
  - Pe double-word (cicluri de Memorie/Configurare)
  - Byte aligned (I/O cycles)
  - Adresele si datele ***scrise*** sunt controlate de catre Initiator
  - Datele ***citite*** sunt controlate de Target
  - Little Endian

- **C/BE#[3:0] (t/s) – Command/Byte enable**
  - Faza de Adresare:
    - *Tipul tranzactiei – PCI command*
    - Controlat de Initiator
  - Faza de Date:
    - *Byte Enable* – eg. BE#[3] corespunde la AD[31:24]-MSB….
    - Controlat de Initiator (write/read)

Interfata Control – Initiator

- **FRAME# (s/t/s)**
  - Controlat de Initiator
  - Indica inceputul, durata si sfarsitul transferului

- **IRDY# (s/t/s)**
  - Controlat de Initiator (“I-Ready”)
  - Indica, ca Initiatorul este gata pentru transfer de date
  - Poate fi folosit de catre Initiator pentru a introduce wait-states

- **“Idle” Bus State**
  - FRAME# si IRDY# sunt inactive/ “de-asserted”

Interfata Control – Target

- **DEVSEL# (s/t/s)**
  - Device Select
  - Controlat de Target, dupa decodarea cu succes a adr. curente
- **TRDY# (s/t/s)**
  - Controlat de Target (“T-Ready”)
  - Indica faptul ca Target este gata ptr. un transfer date
- **STOP# (s/t/s)**
  - Indica intentia Target de terminare a tranzactiei

Configurare

- **IDSEL (in)**
  - Initializare Device Selectat
  - Ciclii PCI de configurare
  - Permit sistemului host configurarea, inainte ca agentii PCI sa fie programati

Arbitrare

- **REQ# (t/s)**
  - INITIATORUL cere controlul asupra bus-ului de la arbitru
  - Linia REQ# dedicata ptr. conexiune punct - punct la arbitru
  - 3-state cand RST# e activ
- **GNT# (t/s)**
  - Arbitrul cedeaza bus-ul initiatorului
  - Linia GNT# dedicata ptr. conexiunea punct - punct la arbitru
  - Ignorat cand RST# este activ

Parity Checking

- **PAR (t/s)**
  - Cerut de toti agentii PCI
  - Transmisie corecta a AD[31:0] si C/BE[3:0]#
  - Even parity (paritate para)*: XOR {AD[31:0],C/BE[3:0],PAR} = 0*
  - Controlat de Initiator ptr. *adrese si scriere date*
  - Controlat de Target ptr. *citire date*

Error Reporting

- **PERR# (s/t/s)**
  - Indica o “data” cu eroare de paritate
  - Controlata de Initiator ptr ciclii read
  - Controlat de Target ptr. Ciclii write
- **SERR# (o/d)**
  - Indica o eroare “catastrofala” in sistem
    - ex. eroare de paritate adresa
  - Uzual rezulta un NMI (restart system)

Semnale extensie/optionale

- Intreruperi
- LOCK#
- CLKRUN#
- Semnale de Power Management
- Semnale interfata JTAG – IEEE 1149.1
- Semnale de extensie pe 64 de biți

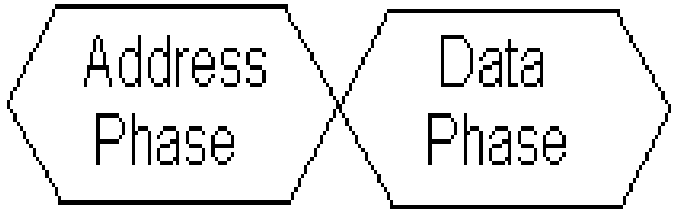
- Linii de extindere la 64 de biți Adresa /date (AD [32-63])
- Comandă /bus Enable (C/BE# [4-7])
  -extensie cu 4 linii ca să opereze cu 8 octeți
- Cerere de transfer pe 64-Bit (REQ64 #)
  - indică slave-ului că inițiatorul cere transferuri pe 64 de biți
- Acceptare transfer pe 64-Biti (ACK64 #)
  - slave indică faptul că este capabil de transferuri pe 64 de biți
- Bit de paritate de date superioare (PAR64)
  - paritate para pr. cei 32 de biti de sus și patru linii de comandă

- *Linii Cerere Întrerupere*
  - Patru linii de întrerupere: INTA #, INTB #, INTC #, INTD # nepartajate

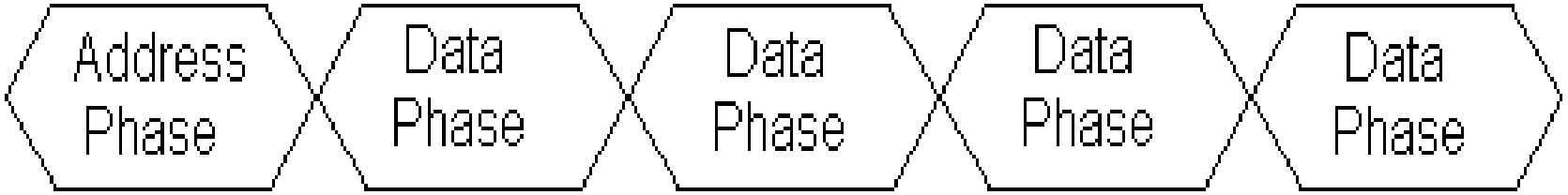
- *Semnalele suplimentare*
  - Pentru a sprijini cache Snoopy protocol
  - semnale Boundary Scan JTAG- IEEE 1149.1
    - Permite testare in-circuit a dispozitivelor PCI
  - M66En semnal pentru a indica frecvența de lucru a bus-ului
    - 33MHz/66 MHz

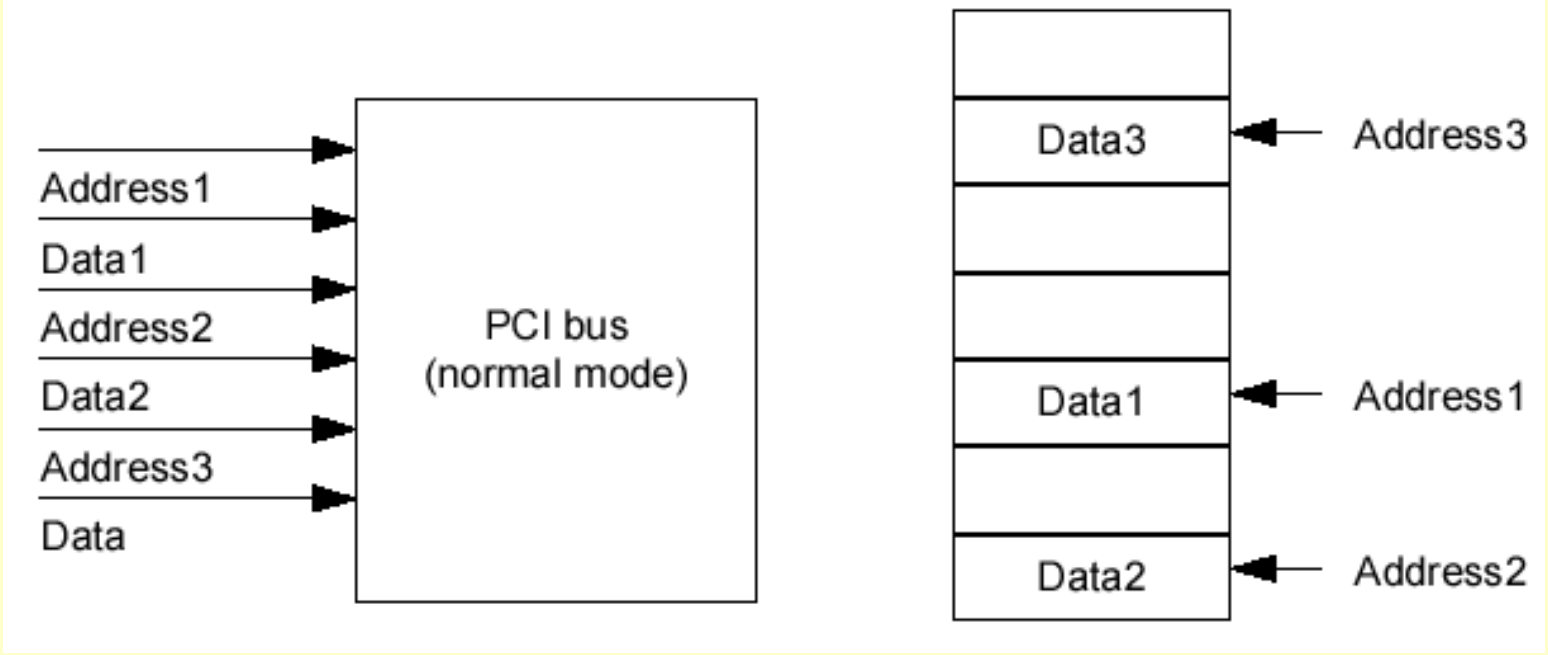
1.5. Transferuri de date pe PCI

Single data phase transaction



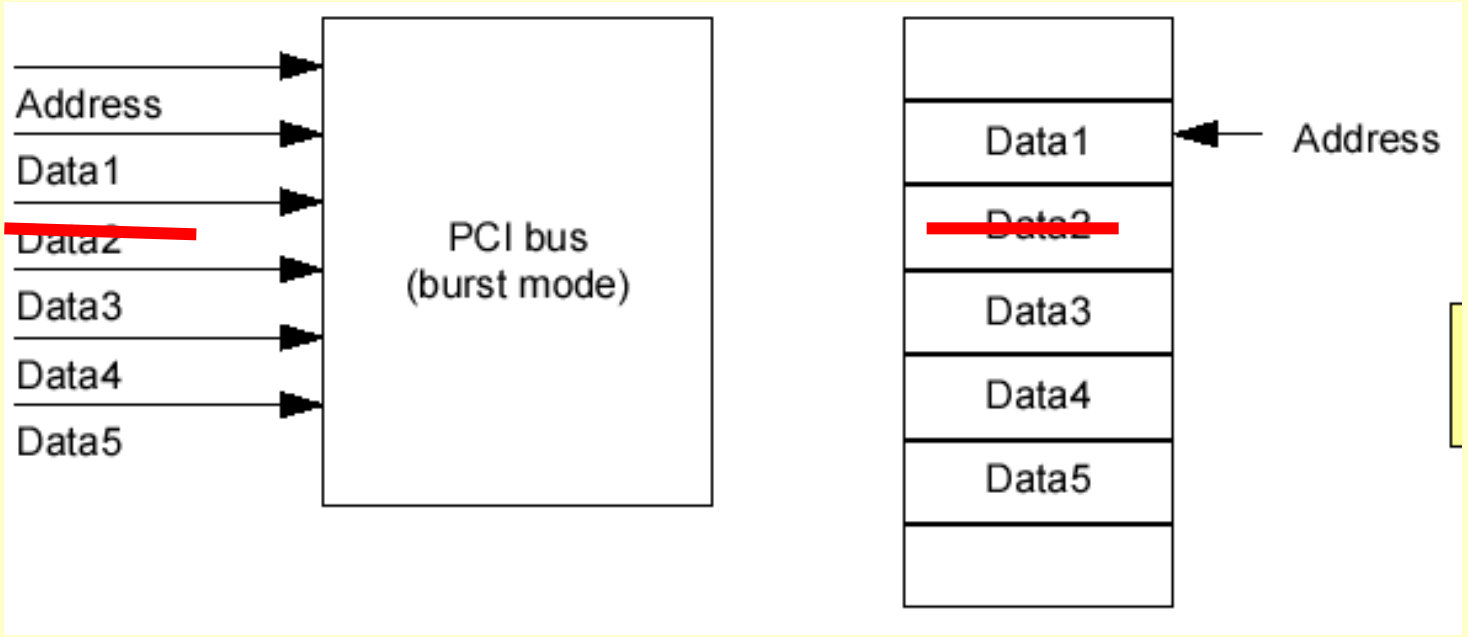
Burst transfer





- **Multiplexarea bus adrese/date** reduce numărul de pini ai bus-ului
  - Penalizarea este de mai multi cicluri de ceas pe transfer (2 cicluri /scriere sau 3 cicluri/citire)
    - Primul ciclu: Faza Adresa
    - Al doilea ciclu: de scriere a datelor
    - Al treilea ciclu: Citire date

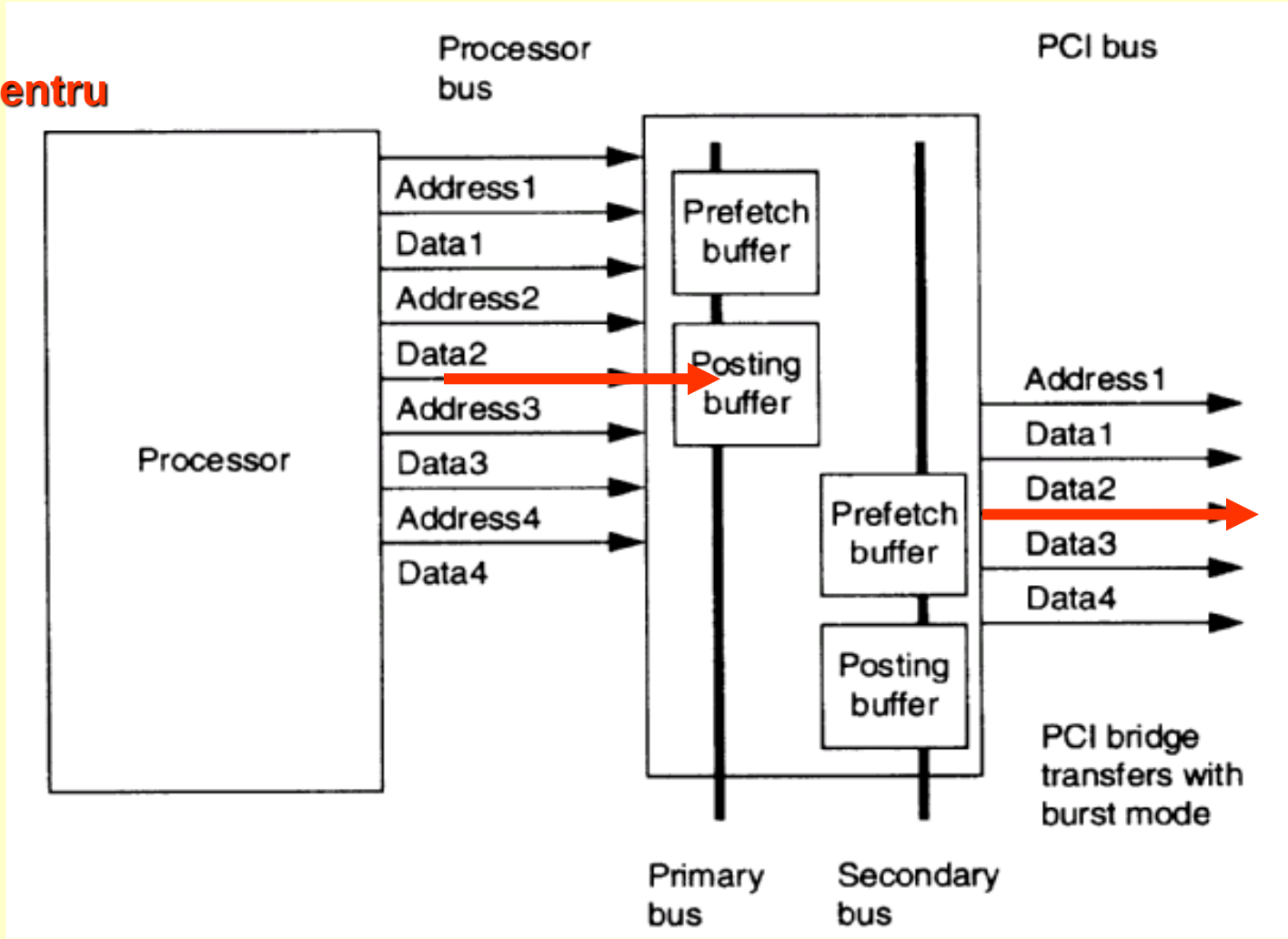
•La un ceas de 33MHz și magistrala de date pe 32 de biți, rata maximă de date scriere = 66MB/s, rata maximă de date citire = 44MB/s



PCI Bridge – buffer-ele sunt utilizate ptr. modul BURST

- Se accelereaza rata de transfer date ptr. ca adresa se trimite doar o data
- Emitatorul si receptorul incrementeaza adresele intern, astfel se transfera doar date – nu avem faze de adresare.
- Pot fi executati orice numar de ciclii de transfer
- Rata max. de tr. date (Burst Mode): 133MB/s (32-bit bus, 33MHz) resp. 266MB/s (64-bit bus, 33MHz)

PCI Bridge – folosirea bufferelor pentru modul BURST



- Puntea PCI formează independent accesele burst (avalansa)
- Puntea PCI *reuneste transferurile de citire singulare* și le scrie ptr. a forma accese in avalansa, dacă adresele de acces individuale sunt secvențiale

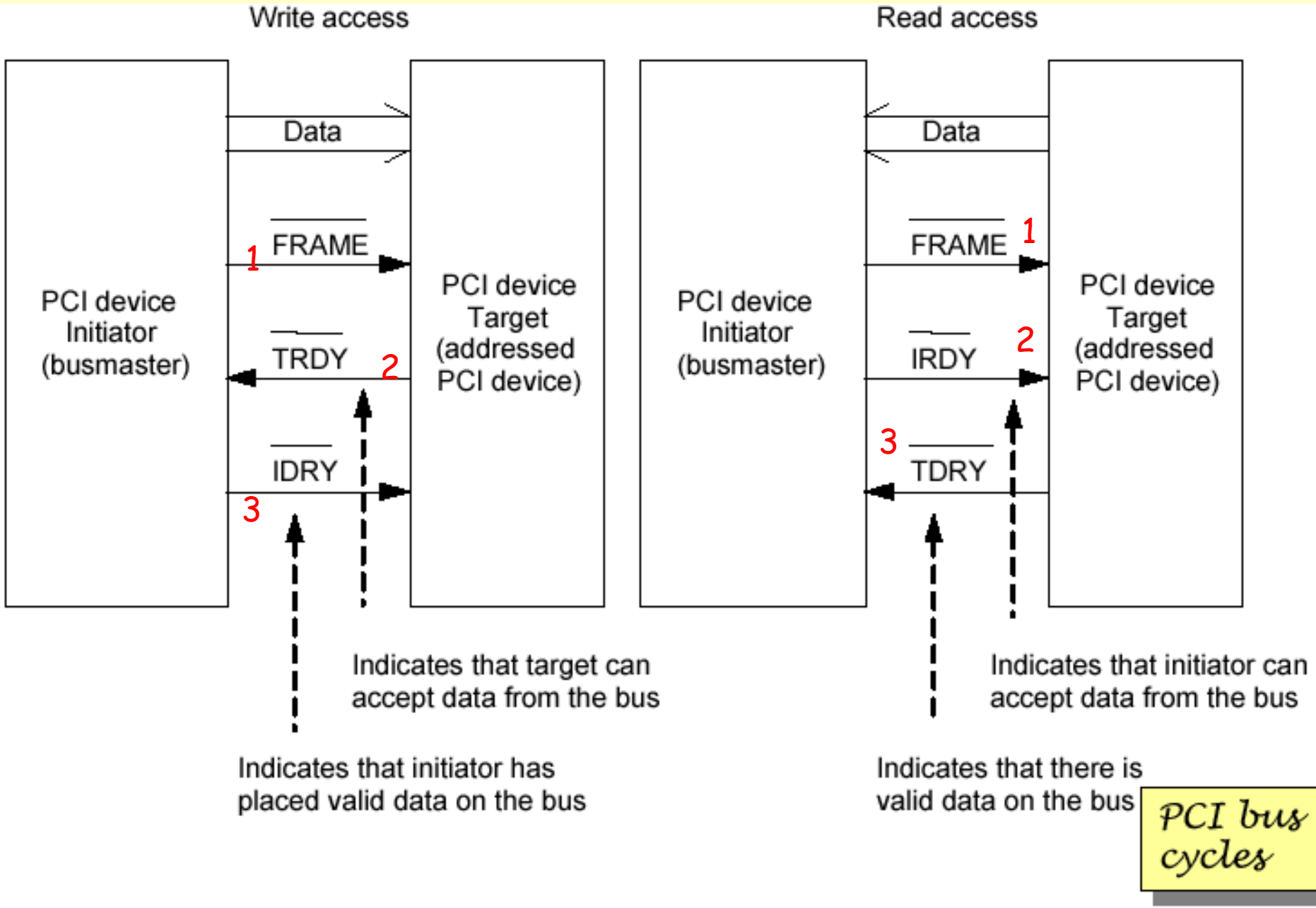
- Pot face transferuri burst, chiar dacă o adresă este sărita dintr-o posibila secvență
- de ex. secventa DW0-DW1-DW3-DW4-DW5. Puntea PCI efectuează DW0-DW1-**DW2**-DW3-DW4-DW5, dar dezactiveaza toate semnalele **# BEx** pentru DW2 pentru a asigura că datele respective nu sunt transferate

1.6. Comenzi PCI

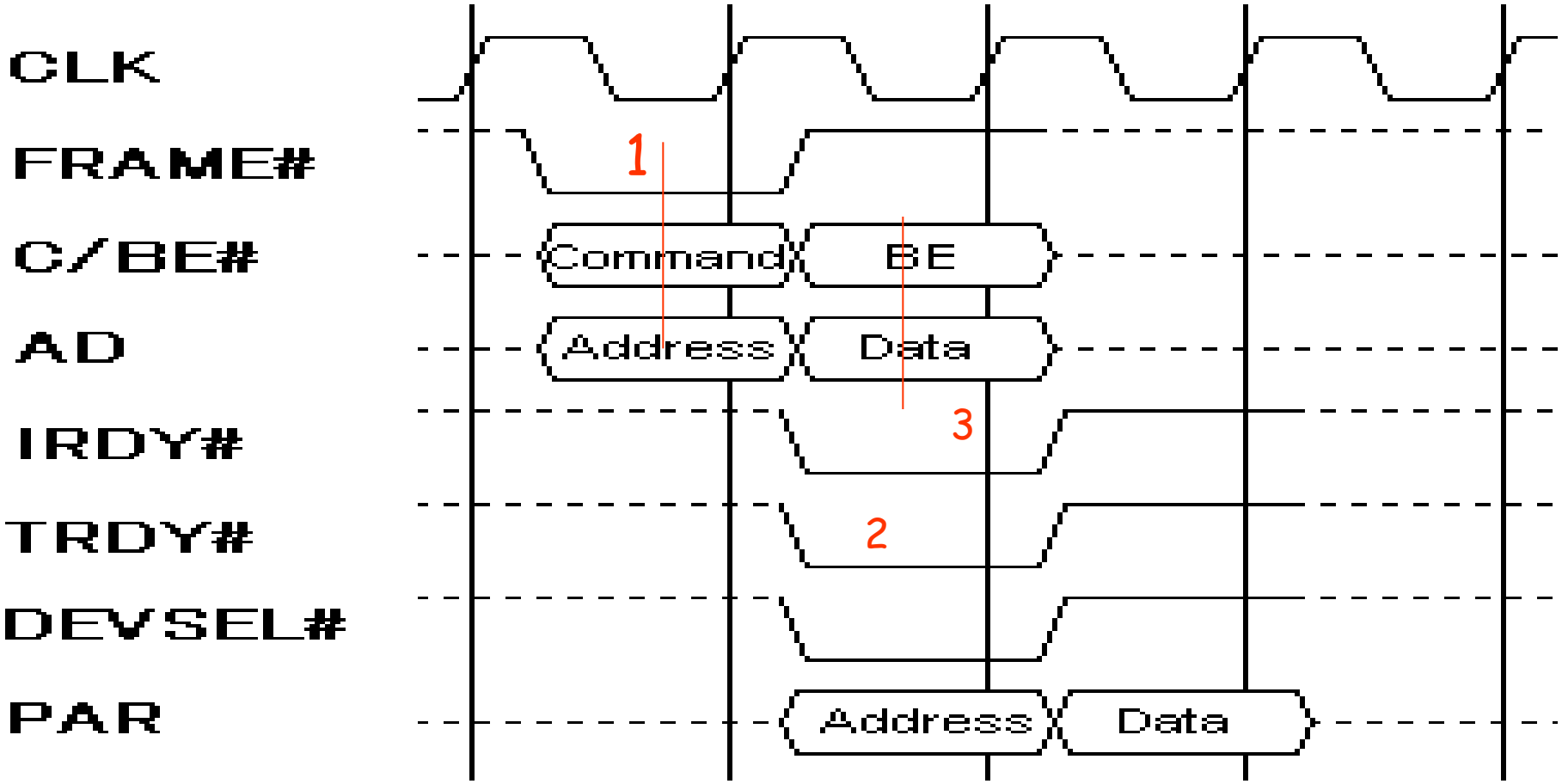
- Tipul comenzii este pus pe liniile C/BE în timpul fazei de adrese
  - **Operațiile de I / O**
    citire/scriere I/O
  - **Operațiile cu memoria**
    Operațiile cu memorie standard citește/scrie memoria
  - **Operațiile cu memoria Bulk**
    Citire linie Memorie
    Citire multipla Memorie
    Scriere-Invalidare Memorie
  - **Operații de configurare**
    Fiecare dispozitiv PCI dispune de un spatiu de configurare de 256-byte
    Două comenzi: Citire/scriere configurație
  - **Operații diverse :** Ciclul special de comanda folosit pentru a transmite un mesaj la *toti sclavii* PCI – eg. Shutdown și Halt
    Ciclul de comandă *Adresa Duala*:
    - Permite inițiatorului pe 32 de biți de a utiliza adrese pe 64 de biți
    - Adresa pe 64 de biți este trimisa ca două valori pe 32 de biți

C/BE[3::0]#	Command Type
0000	Interrupt Acknowledge
0001	Special Cycle
0010	I/O Read
0011	I/O Write
0100	Reserved
0101	Reserved
0110	Memory Read
0111	Memory Write
1000	Reserved
1001	Reserved
1010	Configuration Read
1011	Configuration Write
1100	Memory Read Multiple
1101	Dual Address Cycle
1110	Memory Read Line
1111	Memory Write and Invalidate

Ciclii BUS-ului PCI (C/BE3-C/BE0)

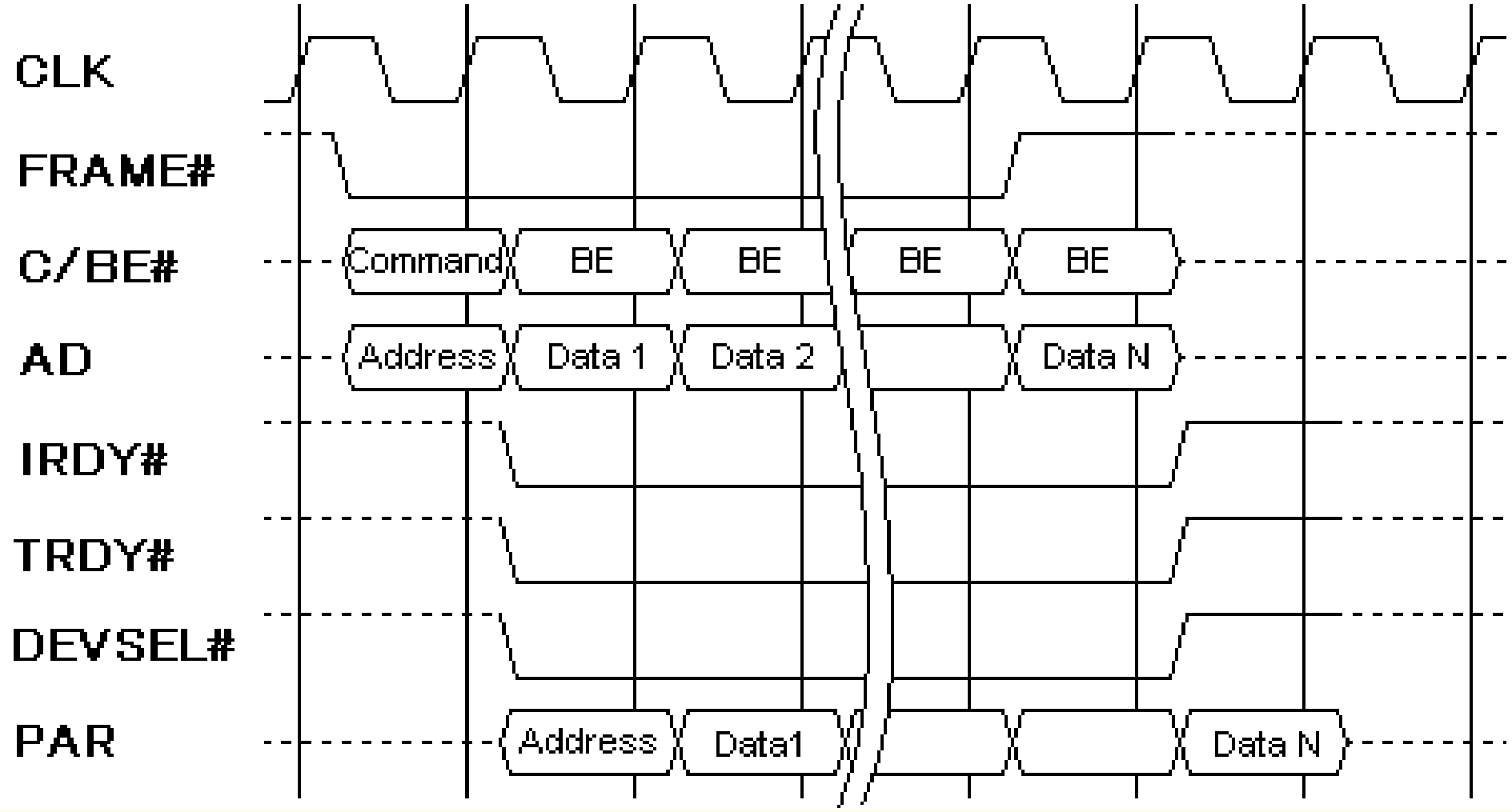


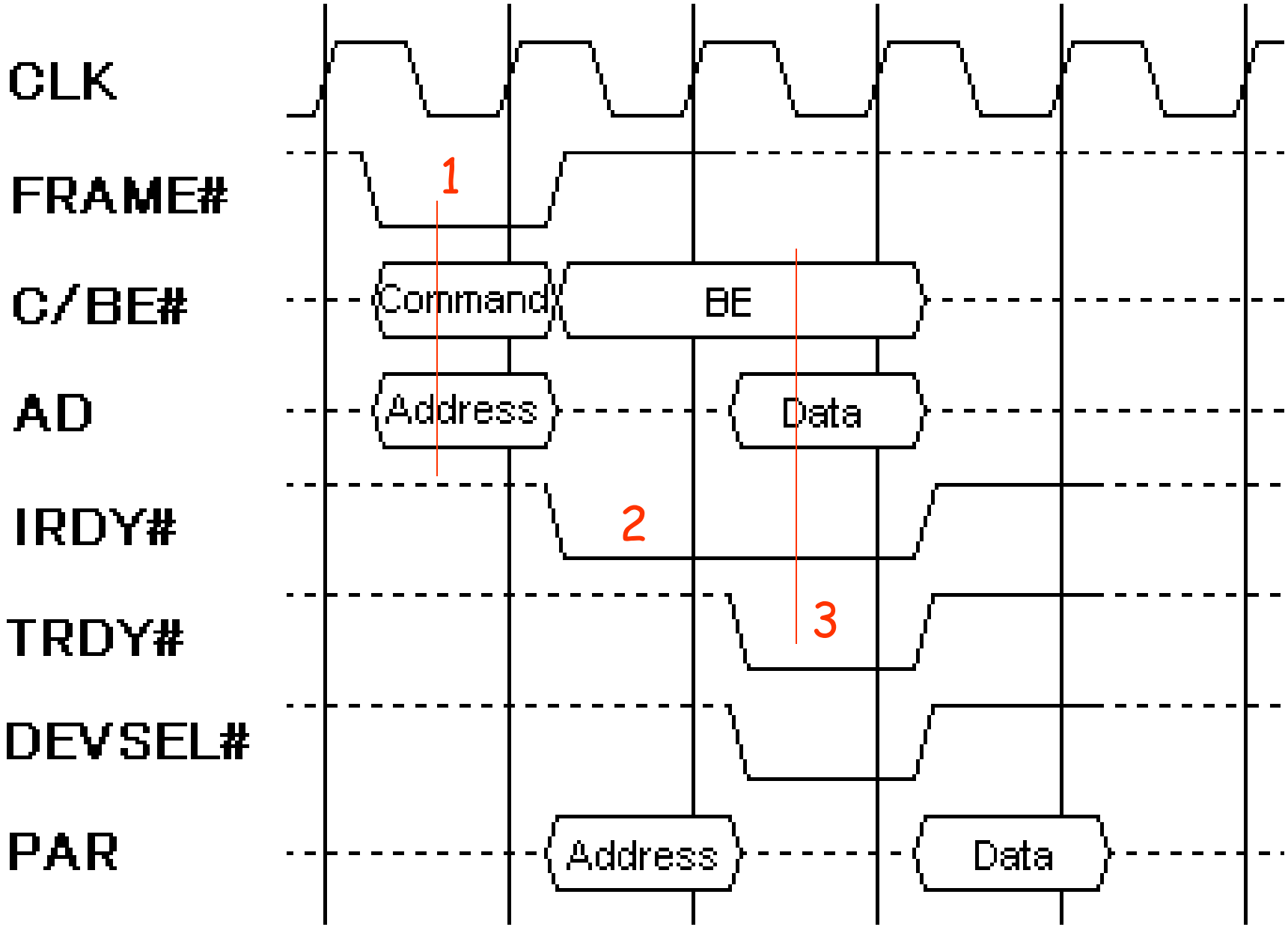
PCI Handshaking



Ciclu WRITE normal

Ciclu WRITE burst ↓




CLK
FRAME#
C/BE#
AD
IRDY#
TRDY#
DEVSEL#
PAR
1
2
3
Command
BE
Address
Data
Address
Data

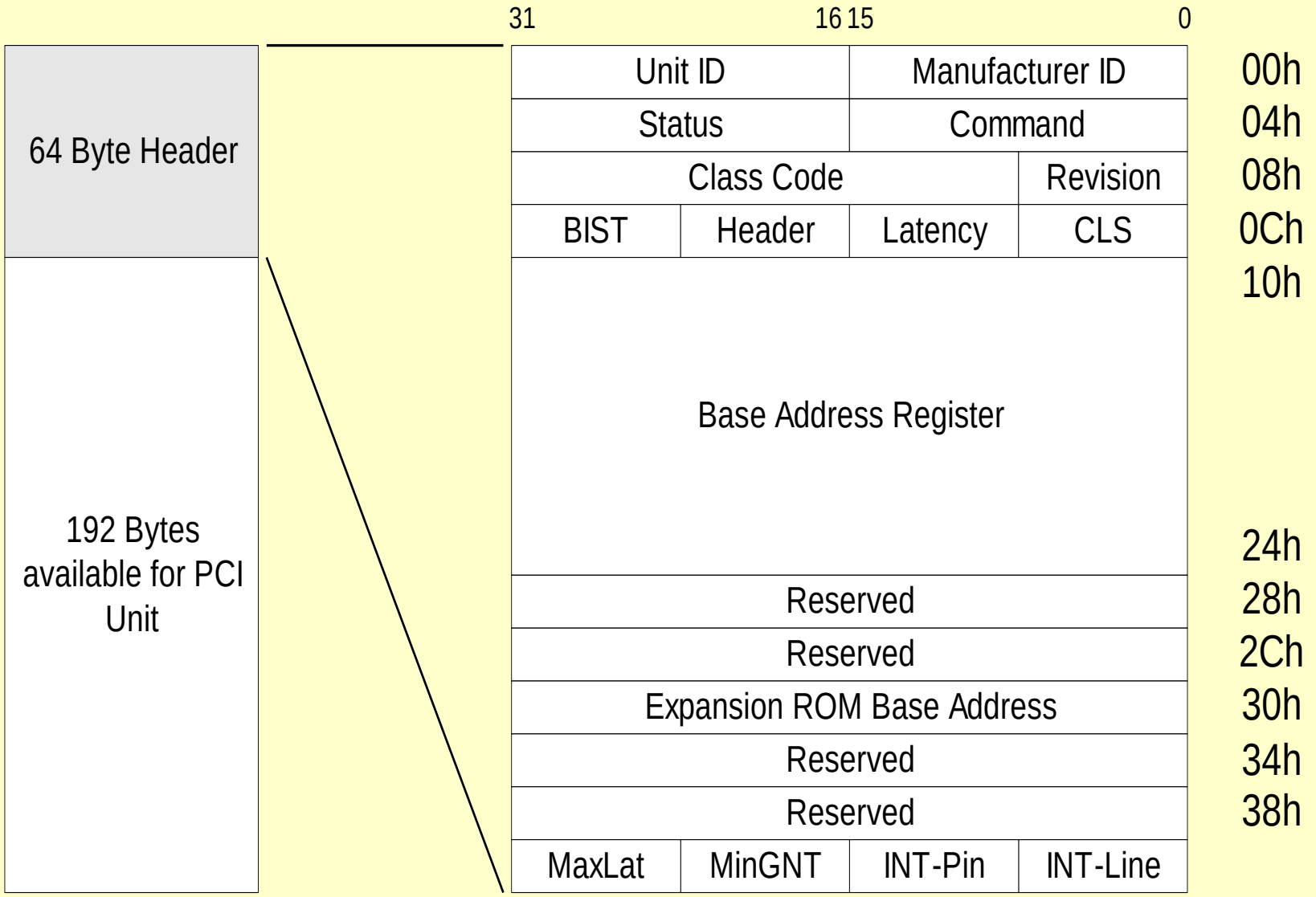
Ciclu READ

1.7. Configurarea bus PCI

- Conceptul Plug-and-Play (PNP)

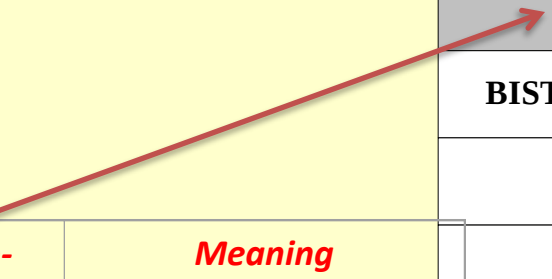
- Permite adaugare de carduri la orice slot fara a schimba jumperii sau switch-urile
- Adresele de selectare, IRQs, porturi COM etc, sunt alocate dinamic la startul sistemul

- Pentru ca PnP sa functioneze, cardurile trebuie să conțină *informații* de bază pentru *BIOS și/sau SO*, de ex.:
  - Tipul de card și dispozitivul folosit
    - Cerințele de memorie-spațiu
    - Întreruperile folosite ...
- Fiecare unitate PCI/funcție separată într-o unitate multi-funcționala are o zona de configurare de 256-bytes care corespunde la 64 registre x 32 de biți
- Primii 64 bytes sunt o zonă de antet fixa. Ceilalti 192 bytes rămasi sunt specifici diferitelor unități



256-byte Configuration Area and 64-byte Header

Basic Code	Meaning	Sub-code	Meaning
00h	Unit was produced before Class Codes were defined	00h	All previous units except VGA
		01h	VGA
01h	Controller for Mass Storage	00h	SCSI controller
		01h	IDE Controller
		02h	Floppy Controller
		03h	IPI Controller
		80h	Other Controller



Device ID			Vendor ID	0x00
Status Register			Command Register	0x04
Class Code			Revision ID	0x08
BIST	Header Type	Latency Timer	Cache Line Size	0x0C
Base Addres 0				0x10
Base Addres 1				0x14
Base Addres 2				0x18
Base Addres 3				0x1C
Base Addres 4				0x20
Base Addres 5				0x24
CardBus CIS Pointer				0x28
Subsystem ID			Subsystem Vendor ID	0x2C
Expansion ROM Base Address				0x30
Reserved			Capability Pointer	0x34
Reserved				0x38
Min_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	0x3C

Vendor Identification	A unique number describing the originator of the PCI device. Digital's PCI Vendor Identification is *0x1011* and **Intel's is *0x8086*.**
Device Identification	A unique number describing the device itself. For example, ***Digital's 21141 fast- ethernet device*** has a device identification of *0x0009*.
Status	This field gives the status of the device with the meaning of the bits of this field set by the standard.
Command	By writing to this field, the system controls the device, for example, allowing the device to access PCI I/O memory
Class Code	This identifies the type of device that this is. There are standard classes for every sort of device; video, SCSI and so on. The *class code for SCSI is 0x0100.*
Base Address Registers	These registers are used to ***determine and allocate the type, amount and location of PCI I/O and PCI memory space*** that the device can use.
Interrupt Pin	Four of the physical pins on the PCI card carry interrupts from the card to the PCI bus. The standard labels these as A, B, C and D. The *Interrupt Pin* field describes which of these pins this PCI device uses. Generally, it is hardwired for a particular device. That is, every time the system boots, the device uses the same interrupt pin. This information allows the interrupt handling subsystem to manage interrupts from this device
Interrupt Line	The *Interrupt Line* field of the device's PCI Configuration header is used to pass an interrupt handle between the PCI initialisation code, the device's driver and the OS's interrupt handling subsystem. The number written there is meaningless to the device driver, but it allows the interrupt handler to correctly route an interrupt from the PCI device to the correct device driver's interrupt handling code within the operating system.

Description		Network Controller
Location		bus 5 (0x05), device 0 (0x00), function 0 (0x00)
Common header		
	Vendor ID	0x8086
	Model ID	0x4222
	Revision ID	0x02
	PI	0x00
	Sub Class	0x80
	Base Class	0x02
	Cache Line	0x10
	Latency	0x00
	Header	0x00
PCI header		
	Address 0 (memory)	0xFE3FF000
	Sub-vendor ID 0x8086	
	Subsystem ID 0x1001	
	Int. Line	0x00
	Int. Pin	0x01
PCI capability		
	Caps class	Power Management
	Caps offset	0xC8
	Caps version	1.1
PCI capability		
	Caps class	Message Signaled Interrupts
	Caps offset	0xD0
PCI capability		
	Caps class	PCI Express
	Caps offset	0xE0
	Device type	Legacy PCI-E Endpoint Device
	Port	0
	Version	1.0
	Link width	1x (max 1x)
Extended capabilities		
	Caps class	Advanced Error Reporting
	Caps offset	0x100
	Caps class	Device Serial Number
	Caps offset	0x140

PCI registers **Vendor ID = INTEL**

	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00	**86 80** 22 42 06 04 10 00 02 00 80 02 10 00 00 00
10	**00 F0 3F FE** 00 00 00 00 00 00 00 00 00 00 00 00
20	00 00 00 00 00 00 00 00 00 00 00 00 86 80 01 10
30	00 00 00 00 C8 00 00 00 00 00 00 00 00 01 00 00
40	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
90	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0	00 00 00 00 00 00 00 00 01 D0 22 C8 00 00 00 0D
D0	05 E0 81 00 0C 30 E0 FE 00 00 00 00 B0 49 00 00
E0	10 00 11 00 C0 0E 00 00 10 08 1B 00 11 1C 07 00
F0	43 01 11 10 00 00 00 00 00 00 00 00 00 00 00 00

1.8 Servicii BIOS ptr. PCI : INT 1Ah

- continutul reg.de configurare pot fi cititi folosind functii BIOS ale INT 1Ah
- *BIOS-ul nu permite transfer de date intre CPU si device-urile PCI (numai cu driverele furnizate de producatorul de PCI device)!!!*

Ex. Detectare servicii BIOS PCI bus

```
.386
        mov ax, 0b101h          ; interrupt 1ah function b101h
        int 1ah                 ; will tell us if there is a PCI
        cmp edx," ICP"          ; bus on the board.
        jz yupi                 ; EDX=20494350h =“  ICP”
                                ; nop

yupi:
```

Ex. Gasirea unui device pe bus-ul PCI

```
INTEL_VENDOR_ID EQU 8086h ; intel's unique sig #
INTEL_EXP_NIC       EQU 1227h ; PCI device etherexpress 10/100 NIC

.386
    mov ax, 0b102h              ; interrupt 1ah function 0b102h
    mov dx, INTEL_VENDOR_ID
    mov cx, INTEL_EXP_NIC
    xor si, si                  ; 0=1st device, 1=2nd etc.
    int 1ah
    jc nope ;   once returned from this call,   BH=bus number, BL=device/function #
nope:
```

Category: **expansion bus BIOSes**

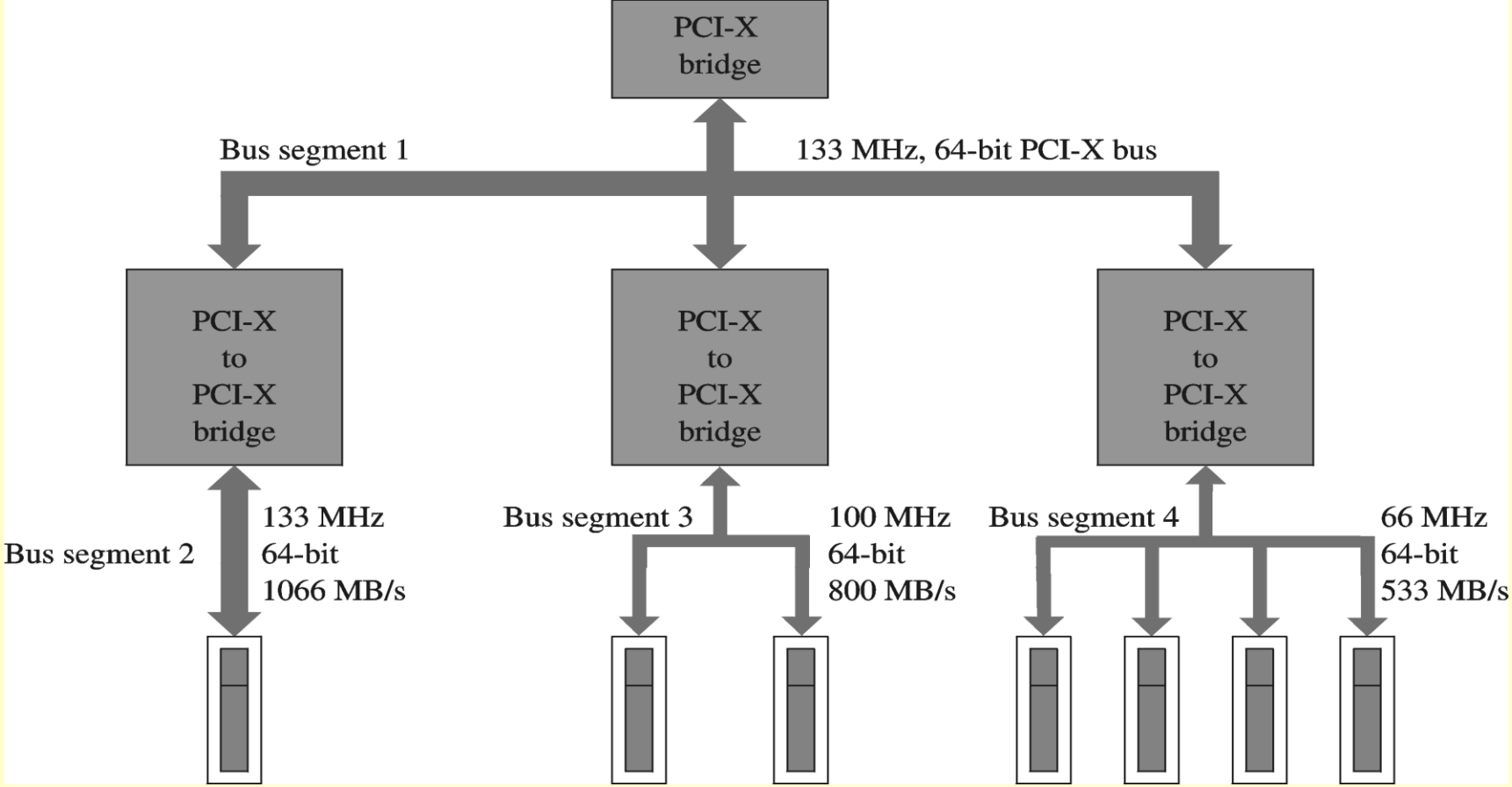
```
INT 1A - PCI BIOS v2.0c+ - FIND PCI DEVICE

        AX = B102h
        CX = device ID (see #00735,#00742,#00743,#00873,#00875)
        DX = vendor ID (see #00732)
        SI = device index (0-n)
Return: CF clear if successful
        CF set on error
        AH = status (00h,83h,86h) (see #00729)
            00h successful
                BH = bus number
                BL = device/function number (bits 7-3 device, bits 2-0 func)
        EAX, EBX, ECX, and EDX may be modified
        all other flags (except IF) may be modified
Notes:  this function may require up to 1024 byte of stack; it will not enable
          interrupts if they were disabled before making the call
        device ID FFFFh may be reserved as a wildcard in future implementations
        the meanings of BL and BH on return were exchanged between the initial
          drafts of the specification and final implementation
        all devices sharing a single vendor ID and device ID may be enumerated
          by incrementing SI from 0 until error 86h is returned
SeeAlso: AX=B182h
```

Table 9.2 BIOS Functions for the PCI Bus (INT 1AH)

Functions	*Call With*	*Returns*
01H – BIOS available?	AH = 0B1H, AL = 01H	AH = 0 if PCI BIOS extension is available BX = version number, EDX = ASCII string 'PCI' CY = 1 if no PCI extension present
02H – PCI unit search	AH = 0B1H, AL= 02H, CX = unit, SI = index DX = manufacturer	AH = result code* BX = bus and unit number CY = 1 for error
03H – class code search	AH = 0B1H, AL = 03H, ECX = class code SI = index	AH = result code* BX = bus and unit number CY = 1 for error
06H – start special cycle	AH = 0B1H, AL = 06H BX = bus, unit number EDX = data	AH = result code* CY = 1 for error *Note*: The value passed in EDX is sent to the PCI bus during the address phase.
08H – configuration byte read	AH = 0B1H, AL = 08H BX = bus, unit number DI = register number	AH = result code* CL = data from configuration register CY = 1 for error
09H – configuration word read	AH = 0B1H, AL = 09H BX = Bus, unit number DI = register number	AH = result code* CX = data from configuration register CY = 1 for error
0AH – configuration double-word read	AH = 0B1H, AL = 0AH BX = bus, unit number DI = register number	AH = result code* ECX = data from configuration register CY = 1 for error
0BH – configuration byte write	AH = 0B1H, AL = 0BH BX = bus, unit number CL = data to be written DI = register number	AH = result code* CY = 1 for error
0CH – configuration word write	AH = 0B1H, AL = 0CH BX = bus, unit number CX = data to be written DI = register number	AH = result code* CY = 1 for error
0DH – configuration double-word write	AH = 0B1H, AL = 0DH BX = bus, unit number CX = data to be written DI = register number	AH = result code* CY = 1 for error

1.9 PCI-X Bus

- **PCI-X = Peripheral Component Interconnect eXtended**
- Destinata nevoilor de banda mai mare: bus-urilor si retelelor mai rapide
- Poate furniza o banda peste 1 GB/s
  - Realizat cu ajutorul bus-ului pe 64 de biți care operează la 133MHz
  - Foloseste protocol registru-registru
- Poate opera la trei frecvente diferite
  - 66 MHz
  - 100 MHz
  - 133 MHz



Year created	1998
Created by	IBM, HP, and Compaq
Superseded by	PCI Express (2004)
Width in bits	64
Capacity	1064 MB/s
Style	*Parallel*
Hotplugging interface	yes

Extensiile PCI-X

Caracteristici:

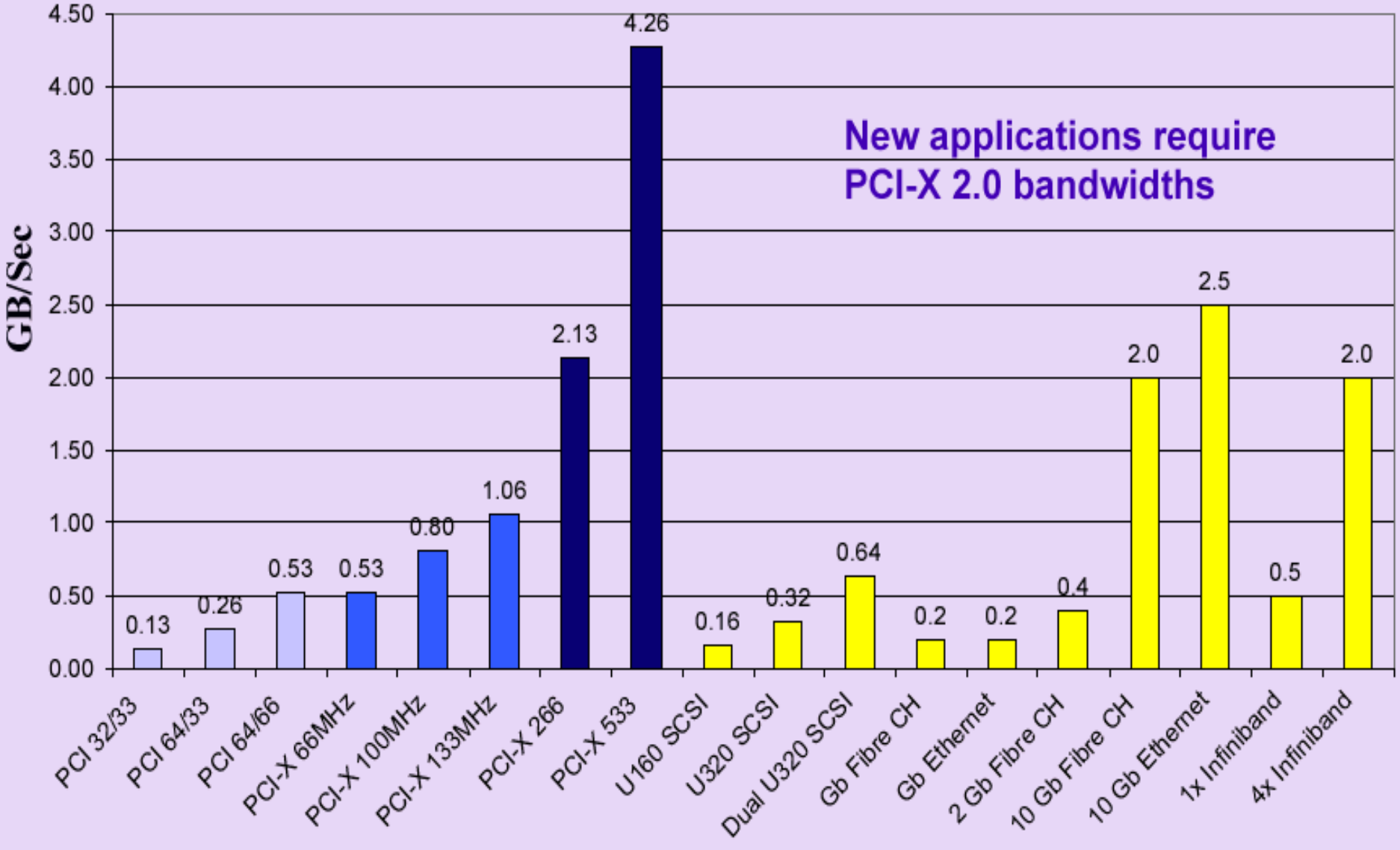
- **Faza atribut** (faza noua de tranzacție)
  - Descrie tranzacția cu mai multe detalii decât PCI (marimea tranzactiei, identitatea inițiatorului tranzacției)

- **Divizarea tranzacției**
  PCI: Tratează cererea-răspuns ca o singură tranzacție
  PCI-X: le împarte în două operațiuni

- **Stările de așteptare optimizate**
  Bus-ul poate fi eliberat ca urmare a împărțirii tranzacției

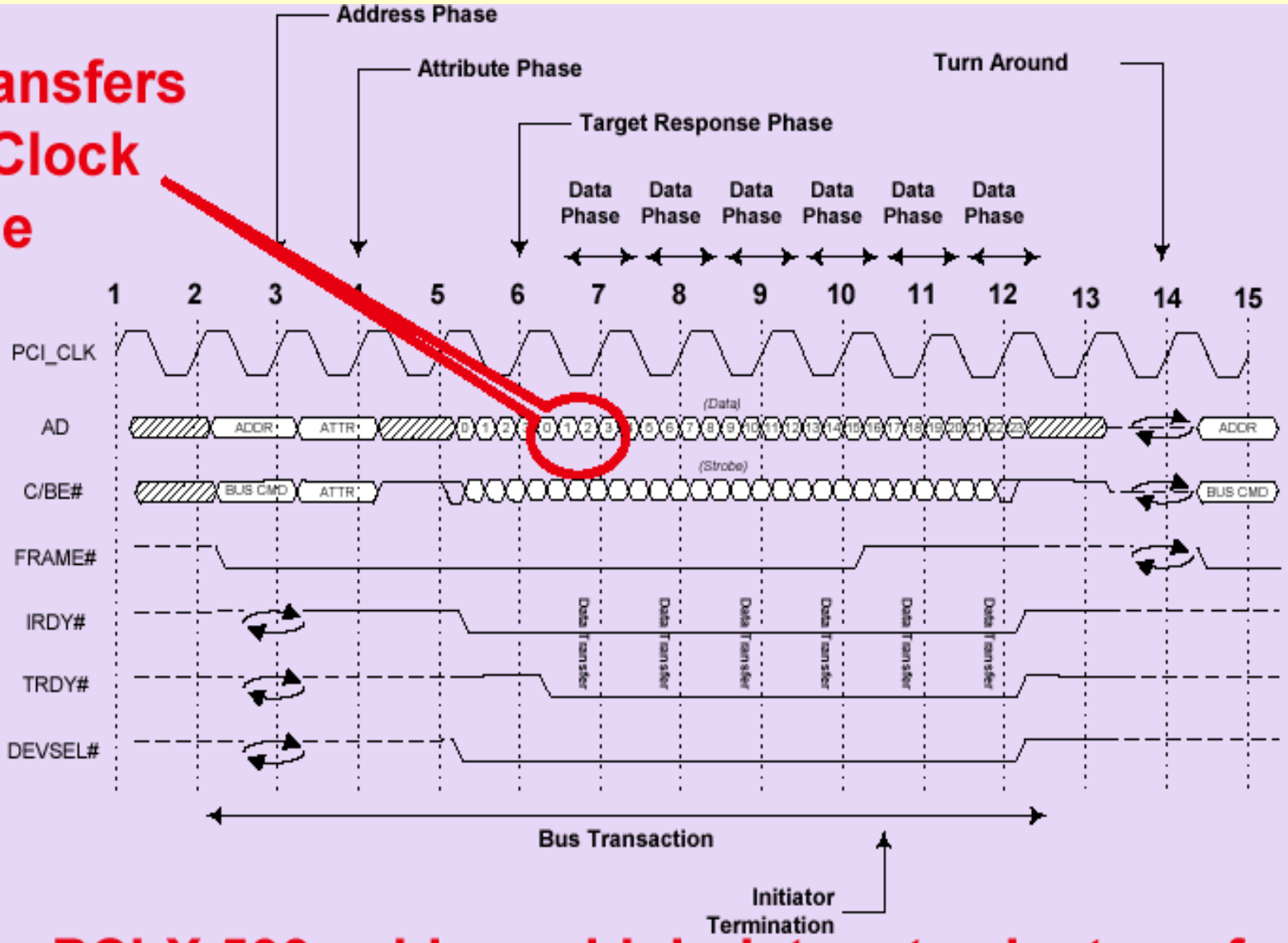
- **Transfer de blocuri de dimensiune standard**

Benzi comparative ale Bus-urilor



PCI-X 2.0 Write - Example

4 Transfers per Clock Cycle



PCI-X 533 achieves high data rates by transferring 4 QWORDS of data per clock cycle.

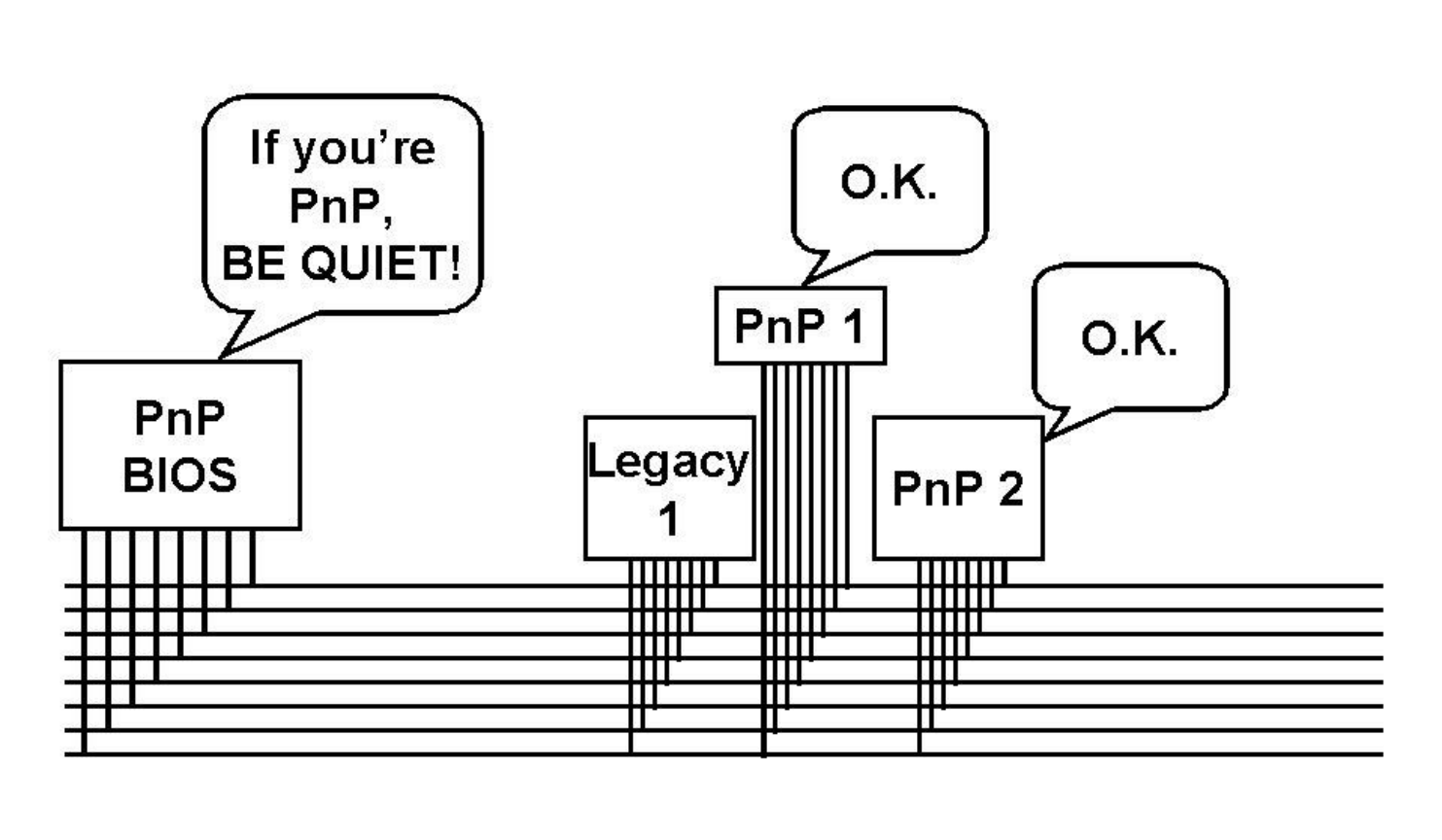
Tema:

a. Cum functioneaza mecanismul PnP?

b. Studiati bus-ul AGP.

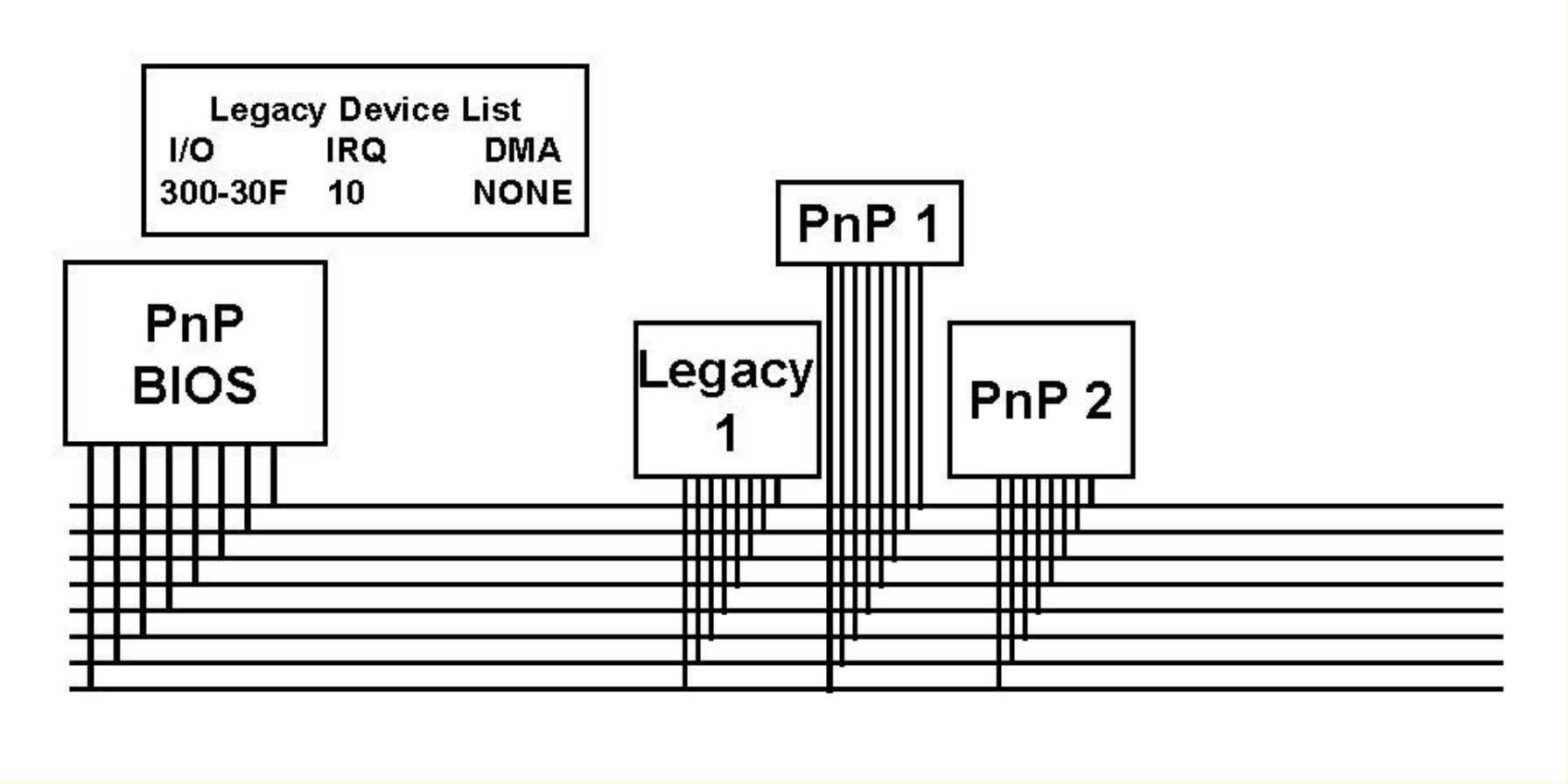
How PnP Works?

- Initially PnP devices remain quiet while the BIOS determines resources required by legacy devices



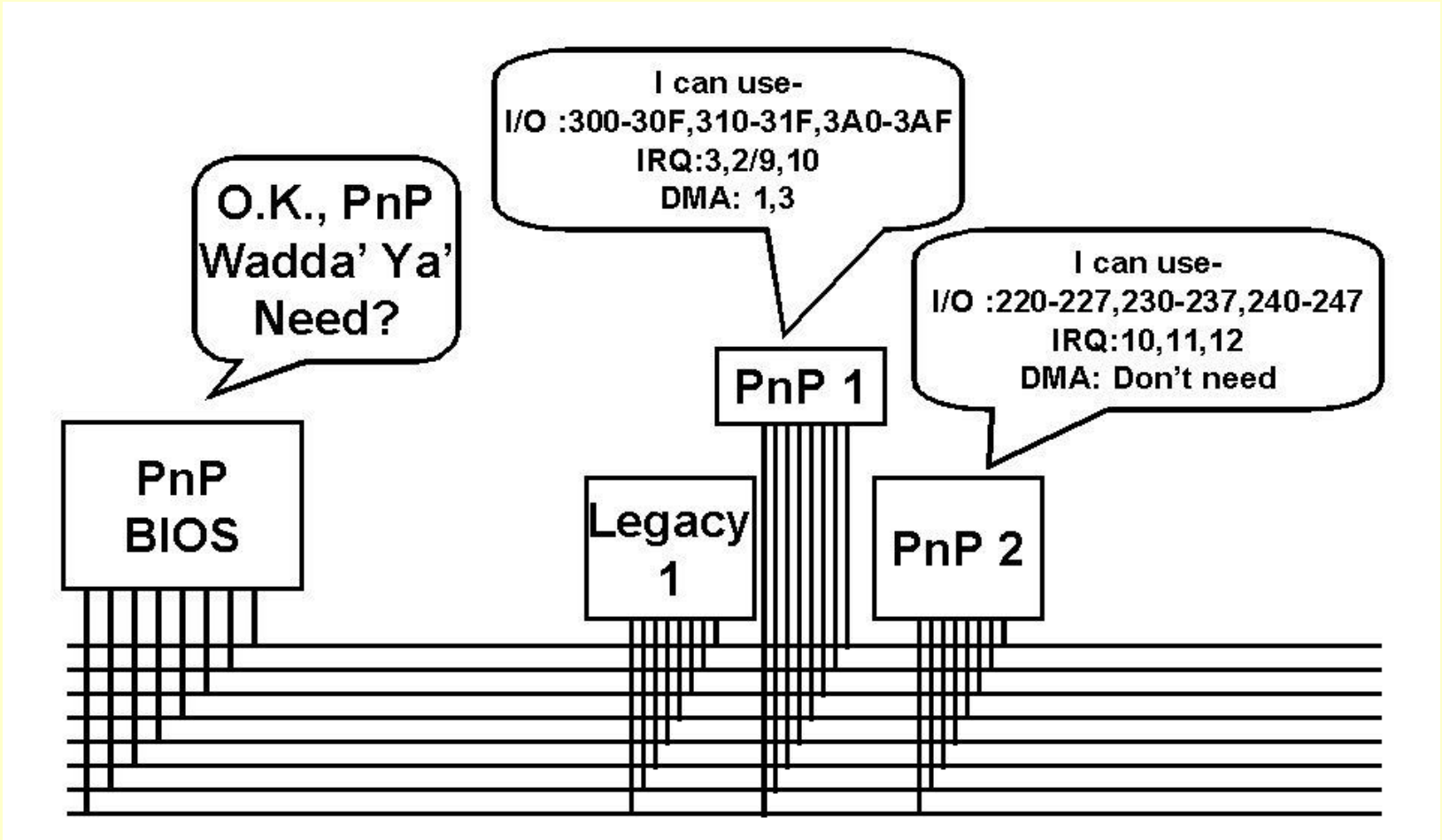
How PnP Works?

- A legacy device list is created to reserve those system resources



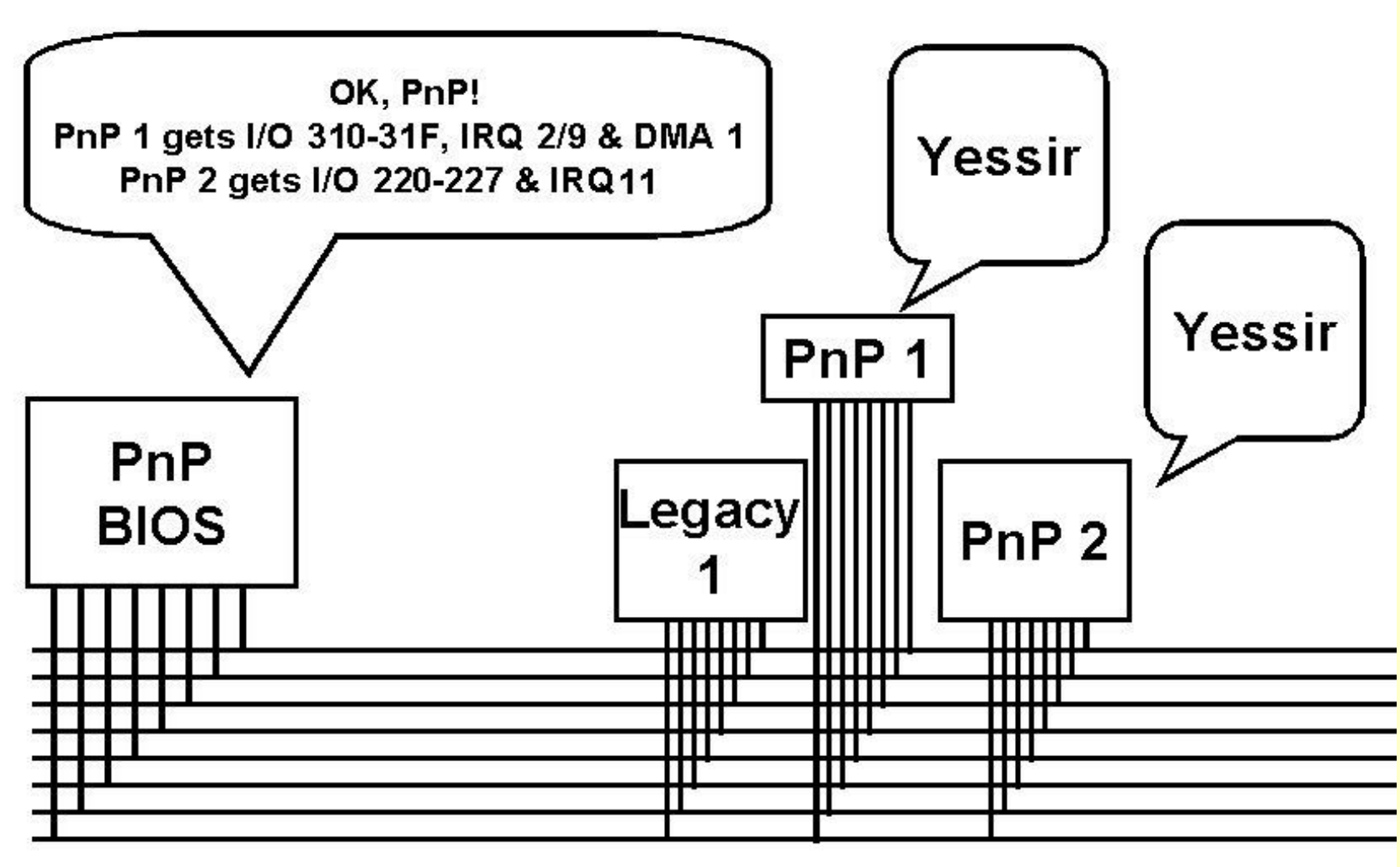
How PnP Works?

- Next the PnP BIOS checks with the PnP devices to see which system resources are options for each device



How PnP Works?

- The PnP BIOS then assigns system resources based on that information



Year created	**2004**
Created by	**Intel · Dell · HP · IBM**
Supersedes	**AGP · PCI · PCI-X**
Width in bits	**1–32**
Number of devices	**One device each on each endpoint of each connection.** **PCI Express switches can create multiple endpoints out of one endpoint to allow sharing one endpoint with multiple devices.**
Capacity	**Per lane (each direction):** **•v1.x: 250 MB/s (2.5 GT/s)** **•v2.x: 500 MB/s (5 GT/s)** **•v3.0: 985 MB/s (8 GT/s)** **•v4.0: 1969 MB/s (16 GT/s)** **So, a 16-lane slot (each direction):** **•v1.x: 4 GB/s (40 GT/s)** **•v2.x: 8 GB/s (80 GT/s)** **•v3.0: 15.75 GB/s (128 GT/s)** **•v4.0: 31.51 GB/s (256 GT/s)**
Style	**Serial**
Hotplugging interface	**Yes, if Express Card, Mobile PCI Express Module or XQD card**
External interface	**Yes, with PCI Express External Cabling, such as Thunderbolt**

http://www.ni.com/white-paper/3767/en/#toc2

Understanding PCI Express
PCI Express: An Overview: Page 1
PCI Express for Graphics
IBM Redbooks | Introduction to PCI Express
Intel® Developer Network for PCI Express Architecture

PCI EXPRESS®

- PCI Express are cele mai bune performante fiind un set paralel de conexiuni seriale (portul 16x de pe placa de baza are 16 conexiuni seriale). Făcând acest lucru, fiecare linie seriala nu trebuie să se sincronizeze perfect cu celelalte linii, atâta timp cât controlerul de la celălalt capăt poate reordona "pachetele" de date pe măsură ce apar, utilizând ordinea corectă.