Computer Architectures

I/O subsystem

Miroslav Šnorek, Pavel Píša, Michal Štepanovský

Czech Technical University in Prague, Faculty of Electrical Engineering

English version partially supported by:
European Social Fund Prague & EU: We invests in your future.

This lecture topics?

- Computer subsystem interconnection
  - Buses, examples of different platforms (VME)
- PC standard example
  - **PCI**
  - Why? Introduction for laboratory experiments
- PCI innovations
  - **PCIe**
  - Why? Introduction for laboratory experiments
  - **Further innovations**
  - Hypertransport, Infiniband, QPI
  - Why? For you to know

What is the main task of I/O subsystem?

- Interconnection of subsystems inside computer, connection of external peripherals and computers together
- Demands on I/O subsystem:

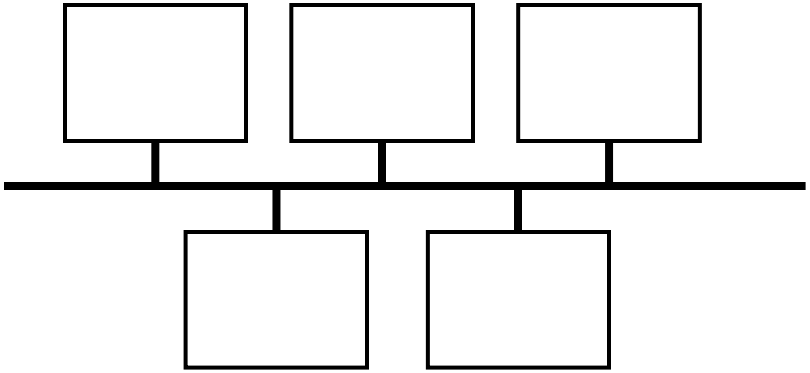
  Creating optimal data paths, especially critical for the most demanding peripherals (graphic cards, external memories)
- Possible solutions:

  There have to be compromises due to price/performance ratio when it is
  - possible to share data paths, or
  - advantageous to share them.

Interfacing terminology – Important terms:

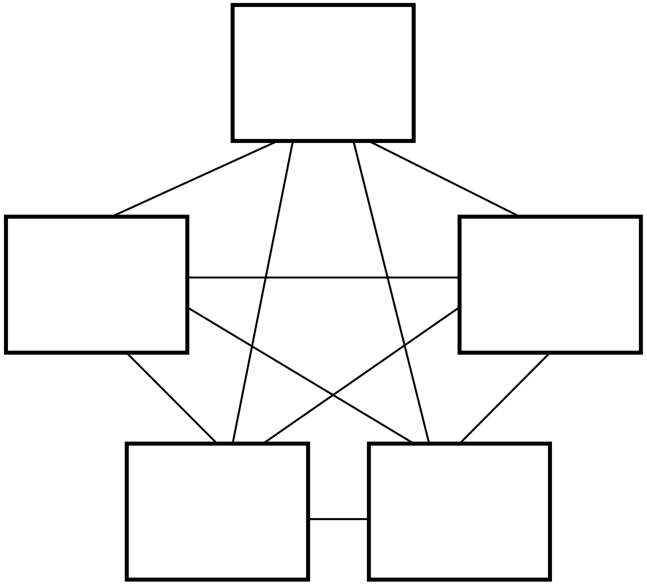
- Interface
  - Common communication part shared by two systems, equipment or programs.
  - Includes also boundary and supporting control elements necessary for their interconnection.
- Bus × point-to-point connection.
- Address, data, control bus.
- Multiplexed/separate bus.
- Processor, system, local, I/O bus.
- Bus cycle, bus transaction.
- Open collector, 3-state output, switched multiplexers

Reminder: bus x point-to-point connection

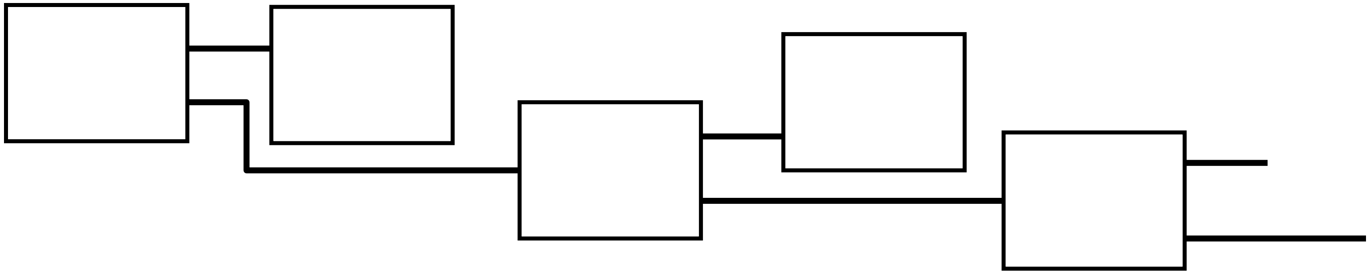


Bus – shared data path

Remark: logical topology as seen from computer system inside (OS, programs) can differ from the physical topology

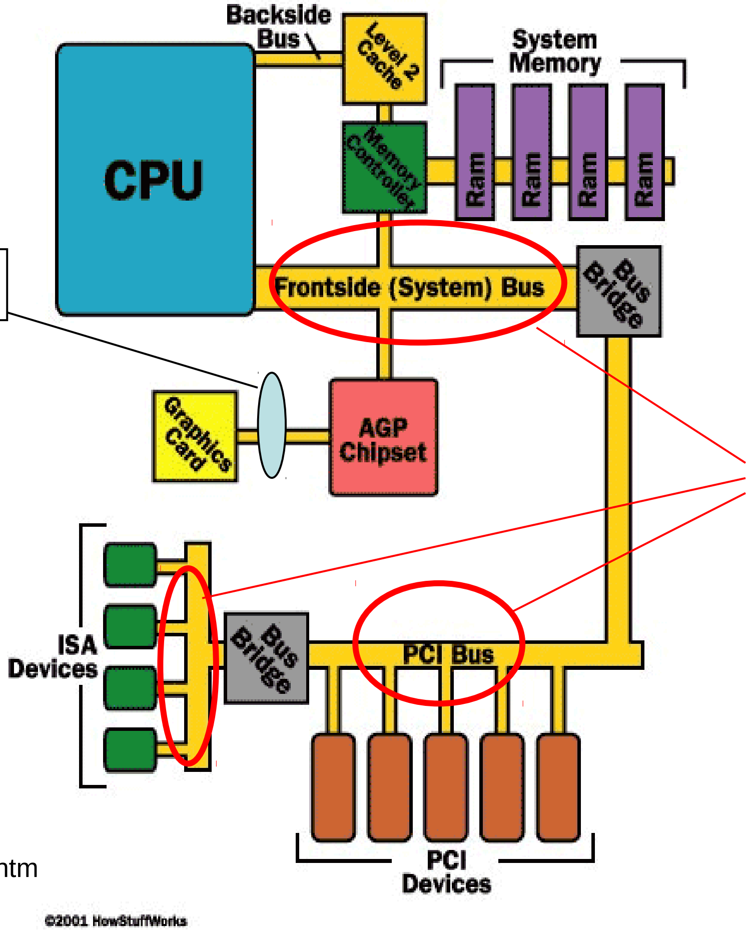


Point-to-point connection



Many different combinations in between in real systems

PC architecture (cca 2000+) … based on PCI

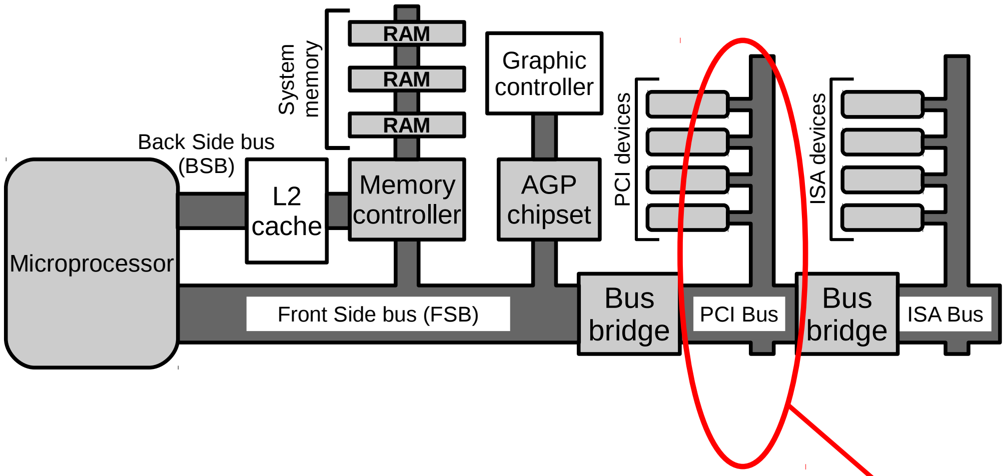


Point-to-point

Buses

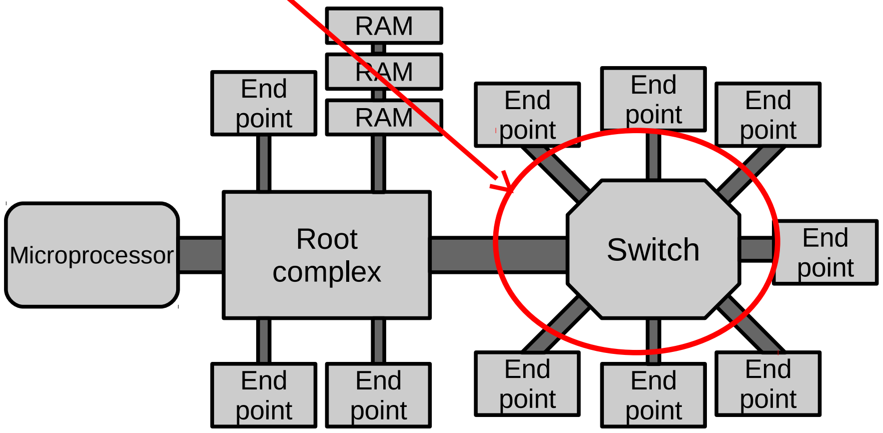
Source: http://computer.howstuffworks.com/pci.htm

PCIe architecture - bus is replaced by shared switch



More details later …

PCIe = PCI Express



Another bus example

Microcomputer control system VME

VME Bus (and system)

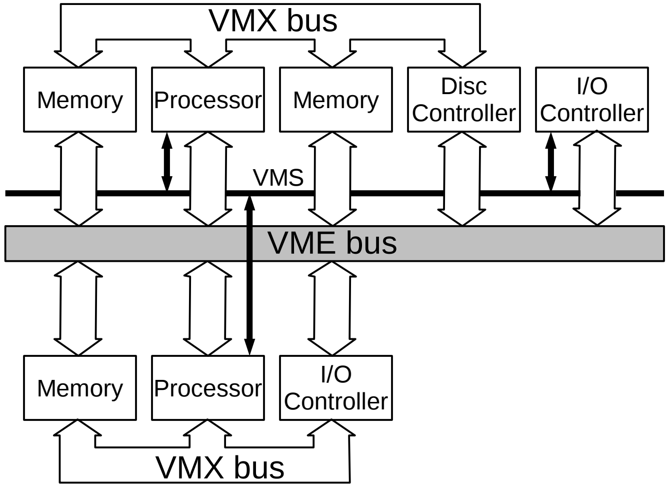
EUROCOM 148 VMEbus CPU boa
American ELTEC, Inc.

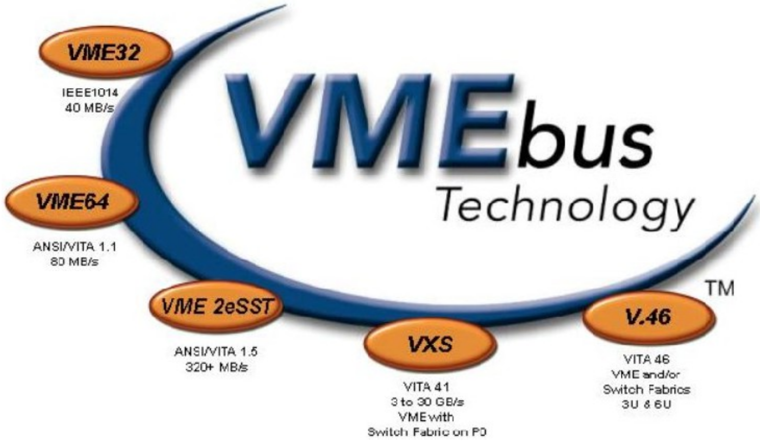
Radstone's PMC GA4, P10 graphics board

Students of KyR surely will meet soon (avionic, railway, industry)

VME milestones

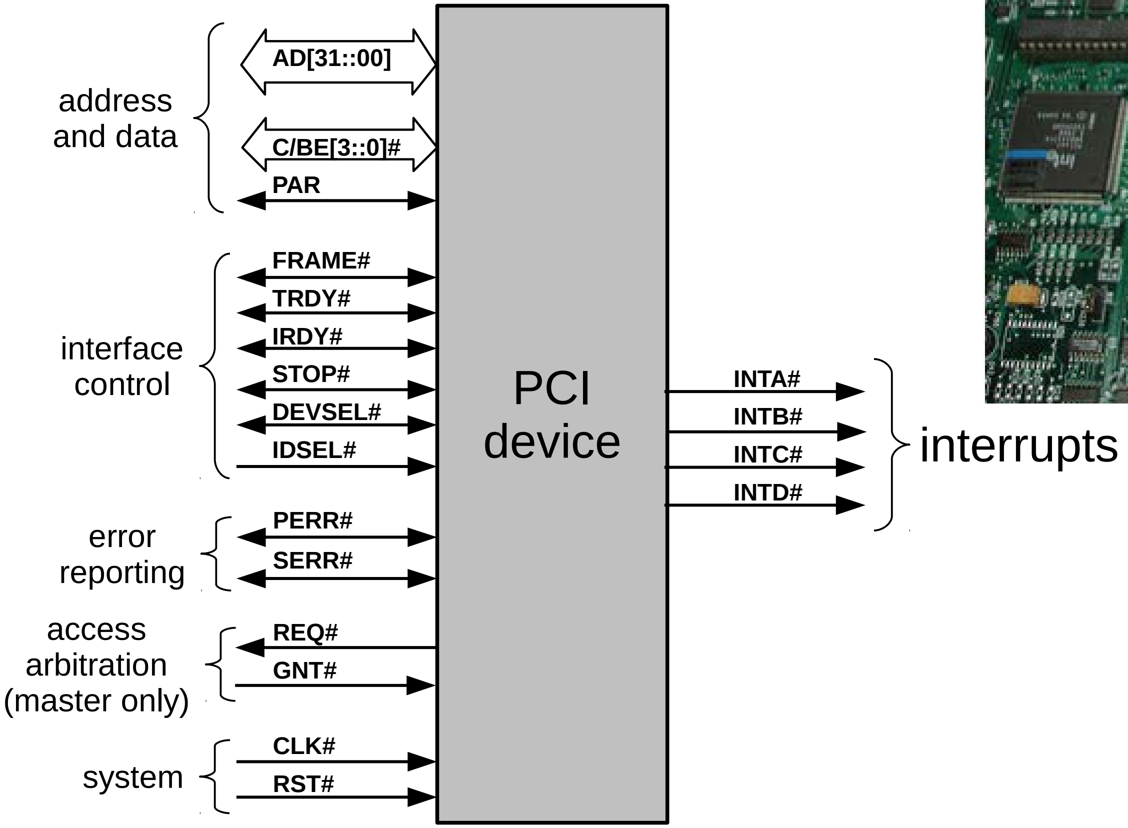
Version	Max. throughput
VME32 (IEEE-1014)	40 MB/s
VME64	80 MB/s
VME2eSST	320+ MB/s

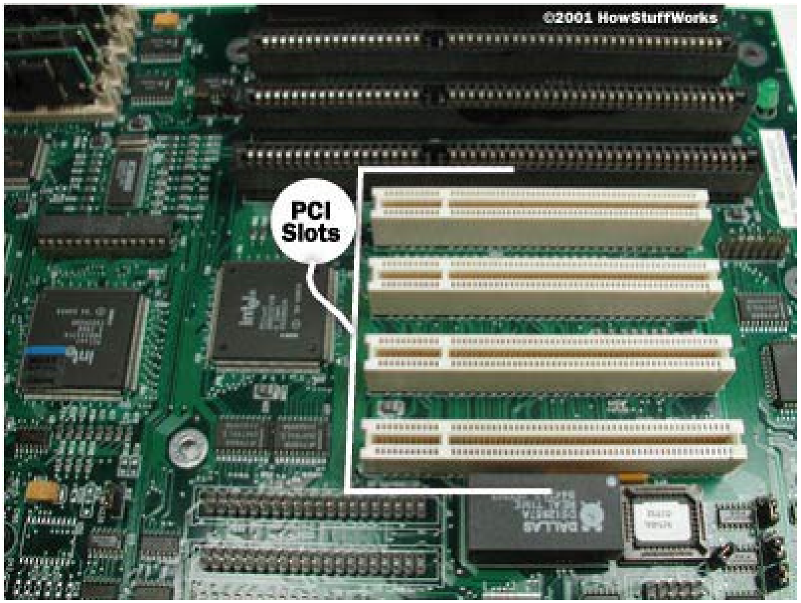




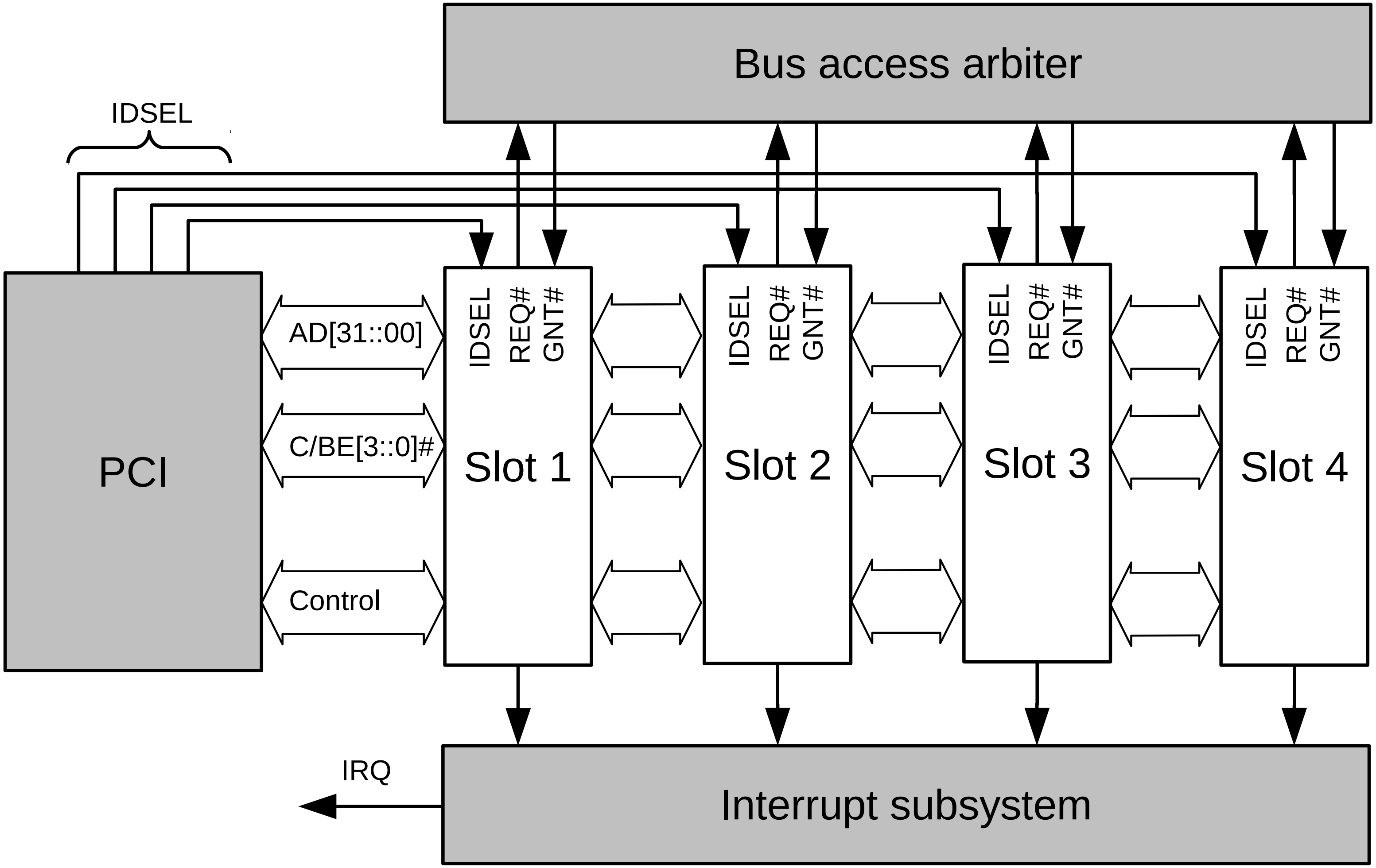
PC platform: some details concerning PCI

PCI signals – 32-bit version

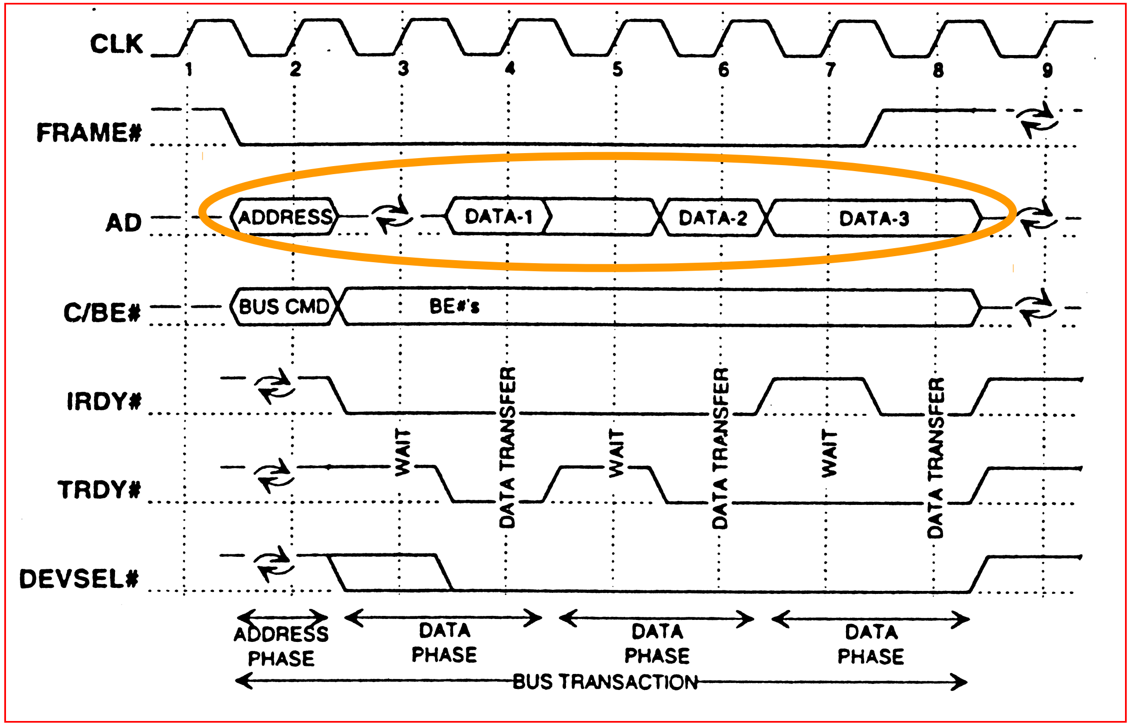




PCI - architecture



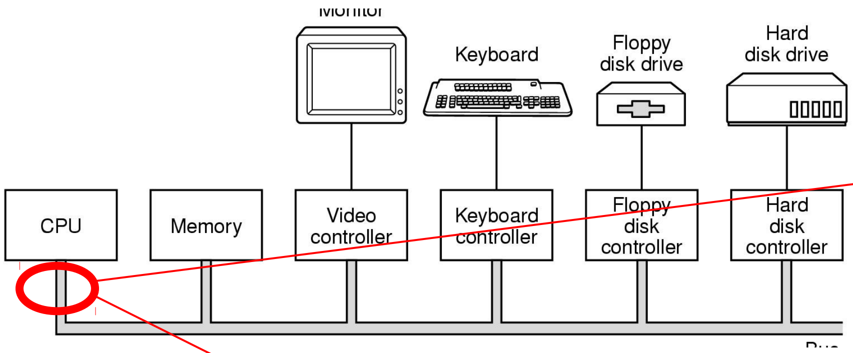
PCI bus cycle example

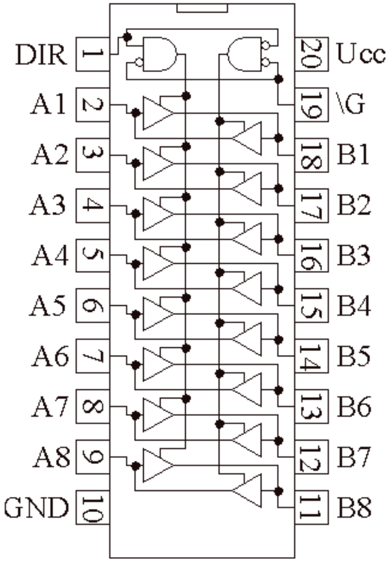


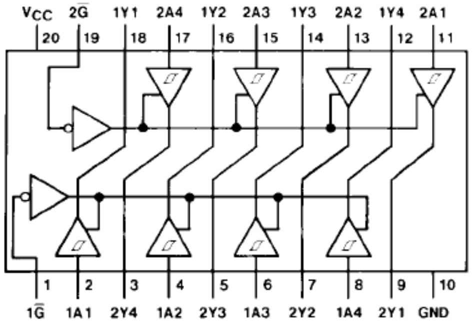
AD bus is - bidirectional and
- multiplexed (explanation later)

Bus-line electronics - buffer

bi-directional

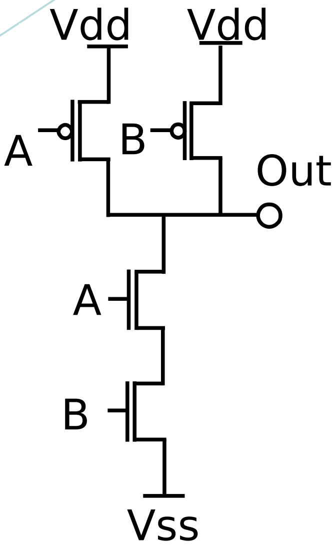




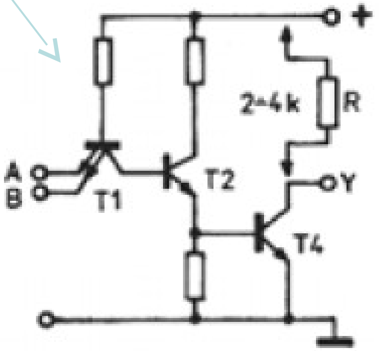


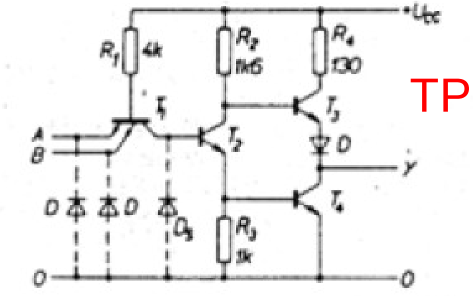
uni-directional

IO output: TP and OC? Practical examples:



OC – Open Collector



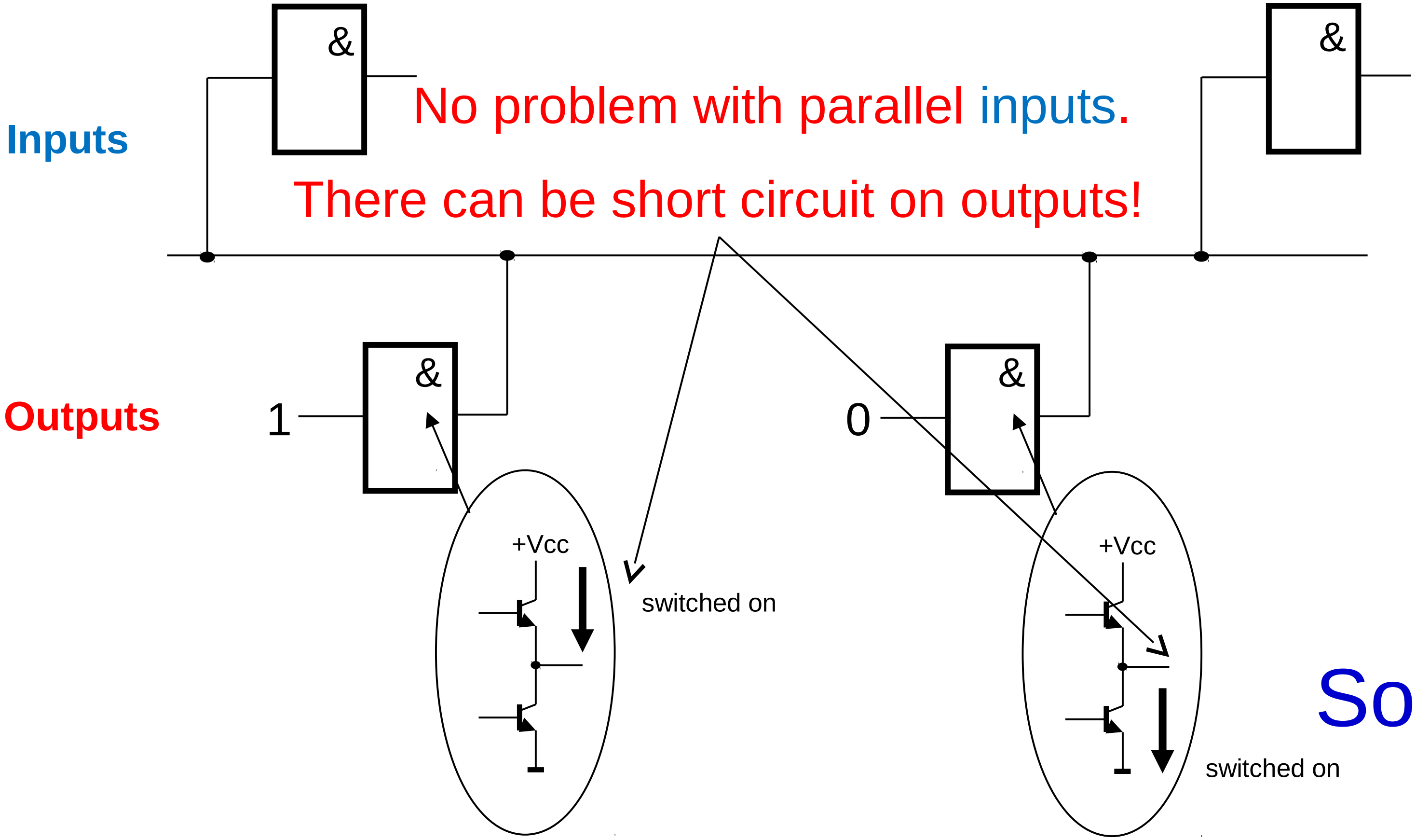


TP – Totem Pole

Internal structure NAND circuit equipped by open collector output.
Pull up resistor is connected externally.

Internal structure of two inputs NAND chip from TTL 7400 family

Study of conflict: bi-directional transfer on a single bus line

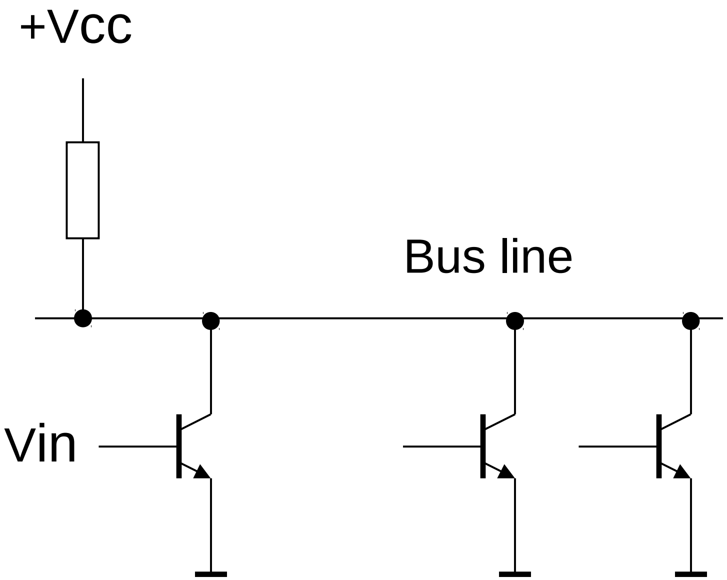


No problem with parallel inputs.

There can be short circuit on outputs!

Solution?

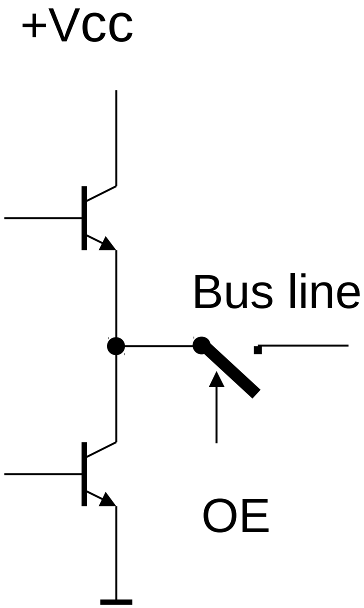
One solution: output circuit as OC x tri-state


+Vcc

Bus line

Vin

OC – Open collector


+Vcc

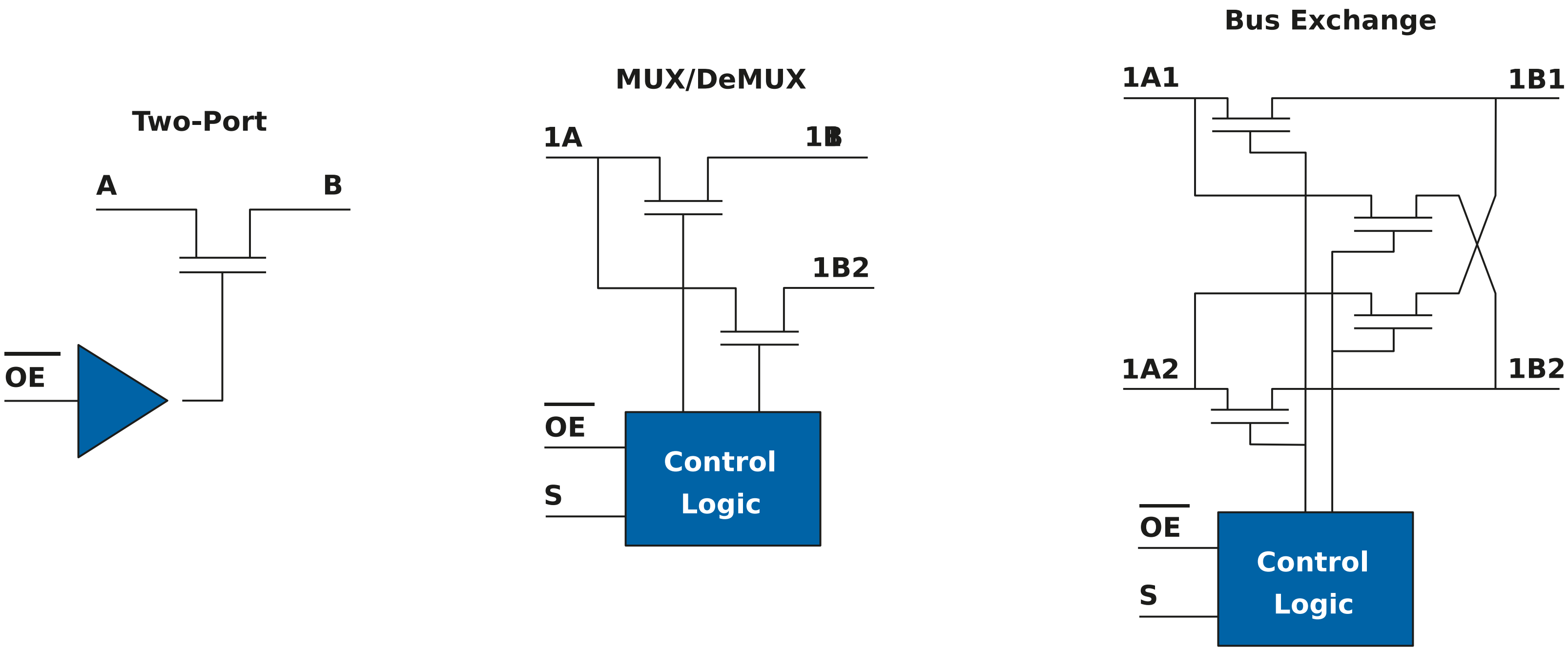
Bus line

OE

3S – Tri-state output

OE = Output Enable

Bus switches and multiplexers

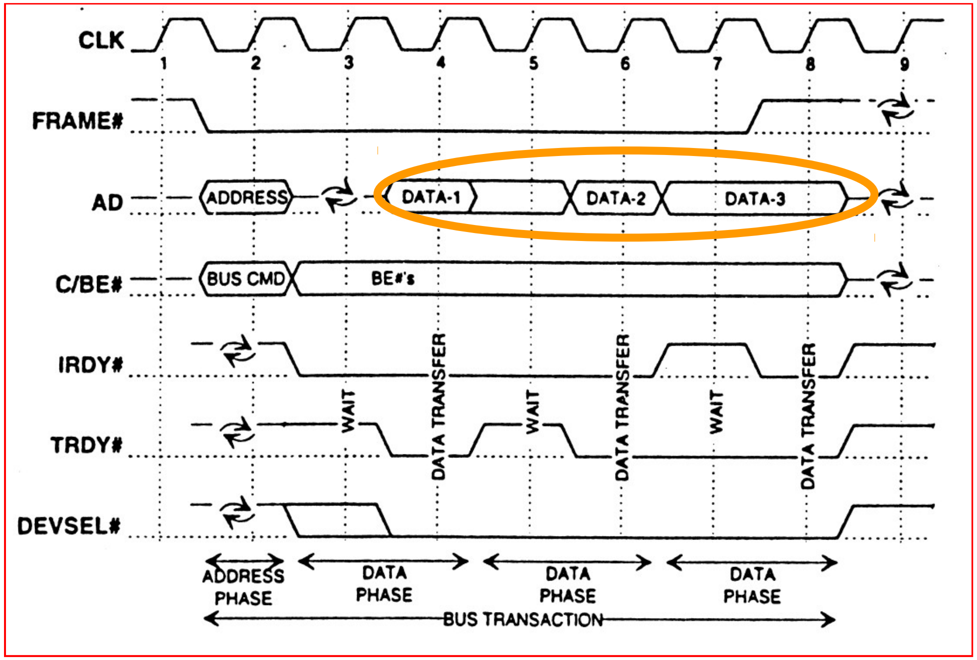


Source: Texas Instruments Digital Bus Switch Selection Guide
http://www.ti.com/signalswitches

Data transfer synchronization

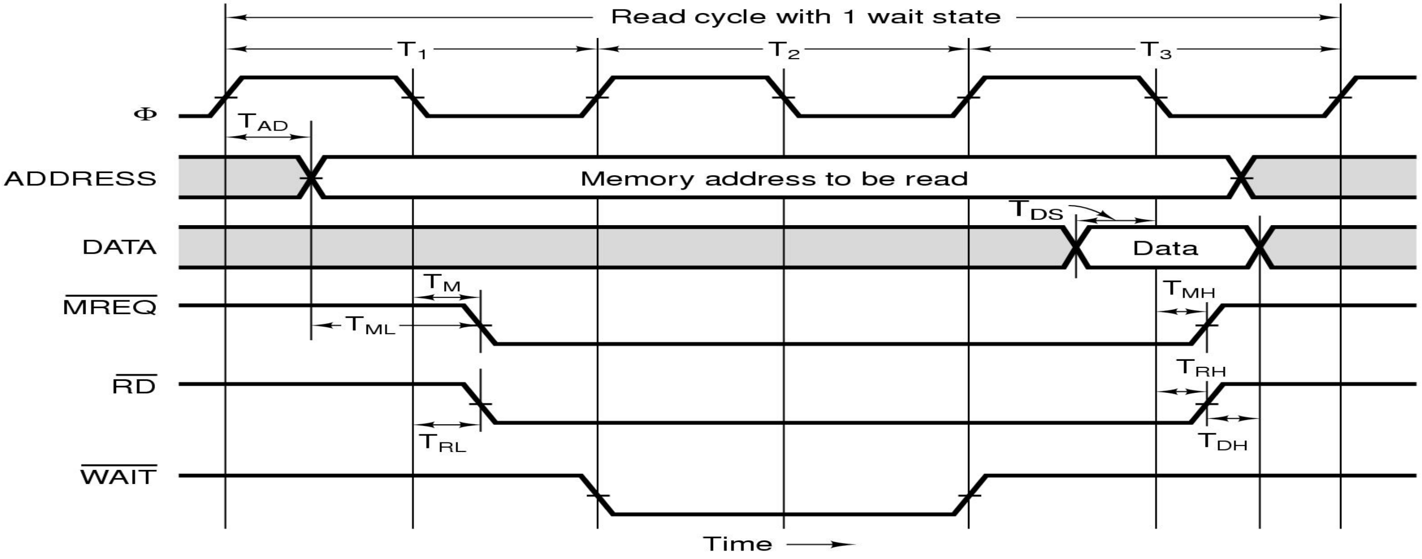
How to synchronize/recognize valid data on the bus?

- Possible transfer types:
  - asynchronous,
  - synchronous,
    - pseudo-synchronous,
  - isochronous (not focus of this lecture).

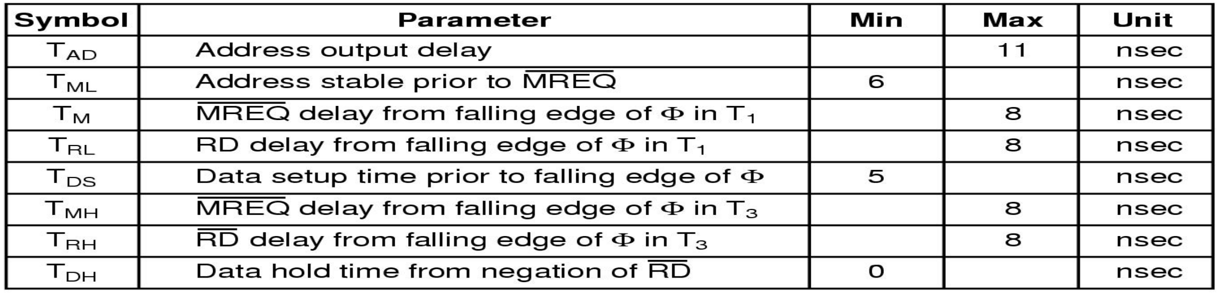


- Remark:
  - usually we call these timing diagrams as bus protocol
  - usually only one bus transfer cycle is analyzed

Synchronization options: synchronous transfer



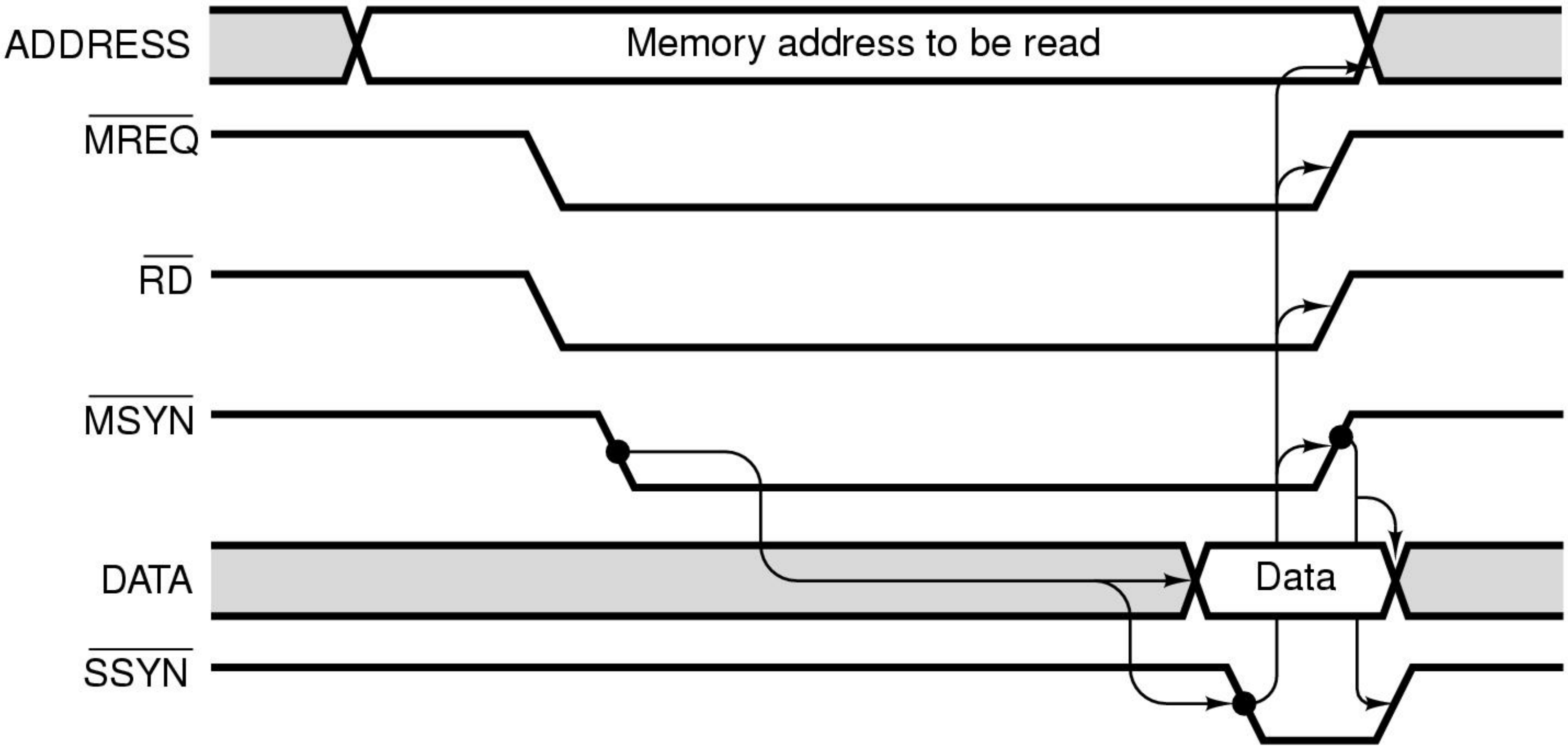
(a)



Symbol	Parameter	Min	Max	Unit
T_{AD}	Address output delay		11	nsec
T_{ML}	Address stable prior to $\overline{MREQ}$	6		nsec
T_M	$\overline{MREQ}$ delay from falling edge of Φ in T_1		8	nsec
T_{RL}	RD delay from falling edge of Φ in T_1		8	nsec
T_{DS}	Data setup time prior to falling edge of Φ	5		nsec
T_{MH}	$\overline{MREQ}$ delay from falling edge of Φ in T_3		8	nsec
T_{RH}	$\overline{RD}$ delay from falling edge of Φ in T_3		8	nsec
T_{DH}	Data hold time from negation of $\overline{RD}$	0		nsec

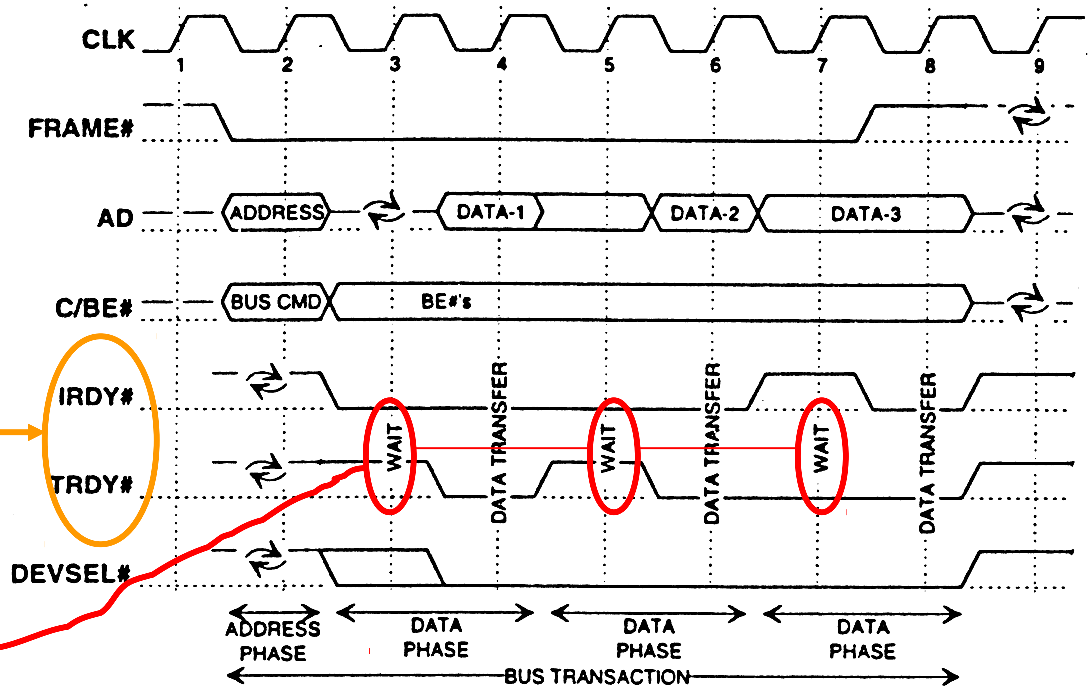
(b)

Synchronization options: asynchronous transfer



How to slow down synchronous variant

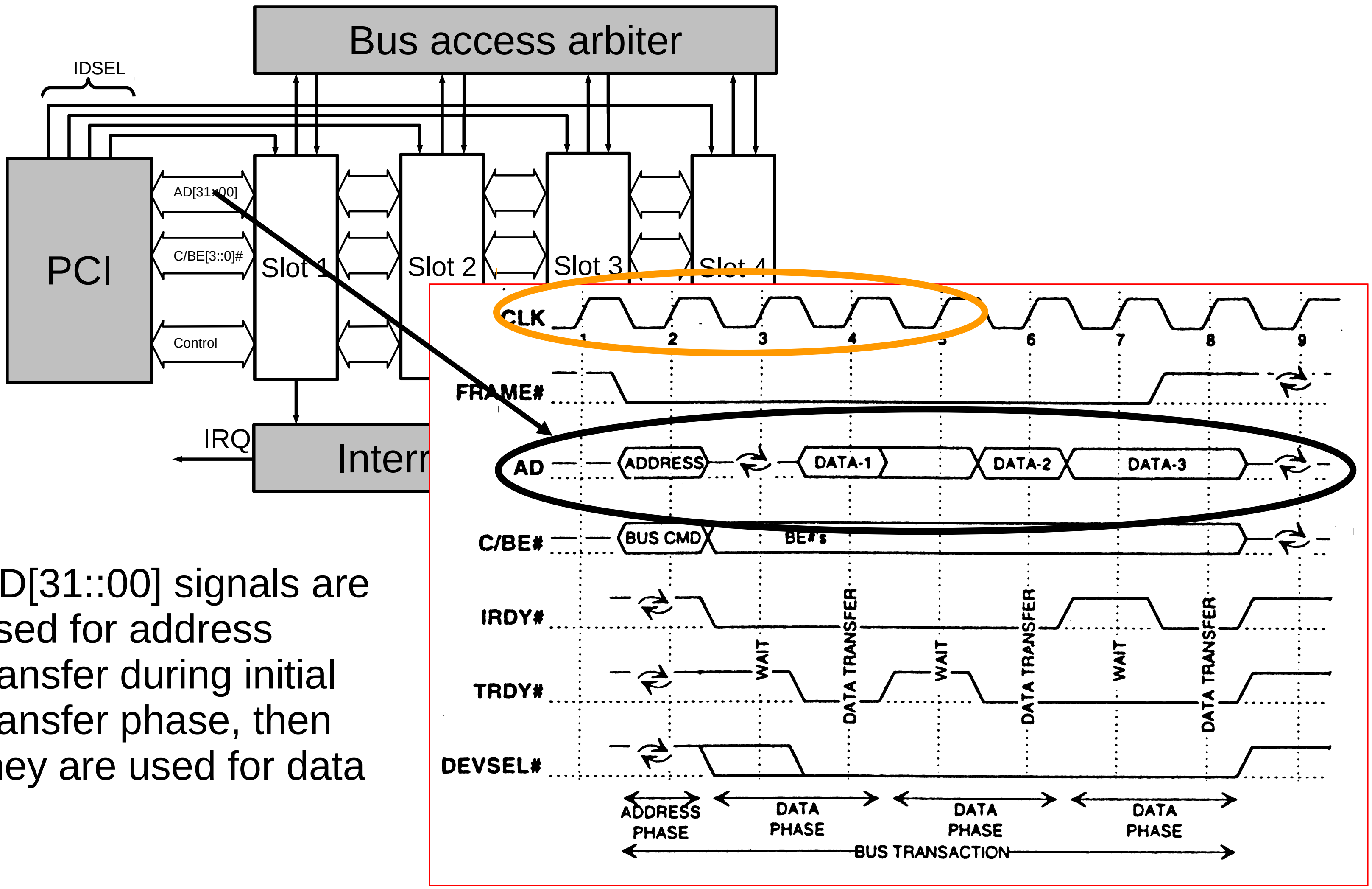
What to do when data source or target cannot process given data at clock rate



Status signals used by transfer participants (initiator or target of the transfer request) requests to insert WAIT-cycles

Pseudo-synchronous transfer variant

Bus/signal lines reuse/multiplexing



AD[31::00] signals are used for address transfer during initial transfer phase, then they are used for data

PCI terms and definitions I.

- Two devices participate in each bus transaction:
  - Initiator (starts the transaction) × Target (obeys request)
  - Initiator = Bus Master,
  - Target = Slave.
- Initiator and target role does not directly impose data source and receiver role!
- What does it mean that bus is multimaster?
  - More participants can act as Bus Master – initiate transfers!
- When bus signals are shared, then transactions need to be serialized and only one device can act as master at any given time instant
  - Bus master is responsible to grant bus to requesting participant
- Who takes care of bus arbitration?
  - One dedicated device or bus backplane
  - Or decentralized arbitration is possible for some buses (not PCI)

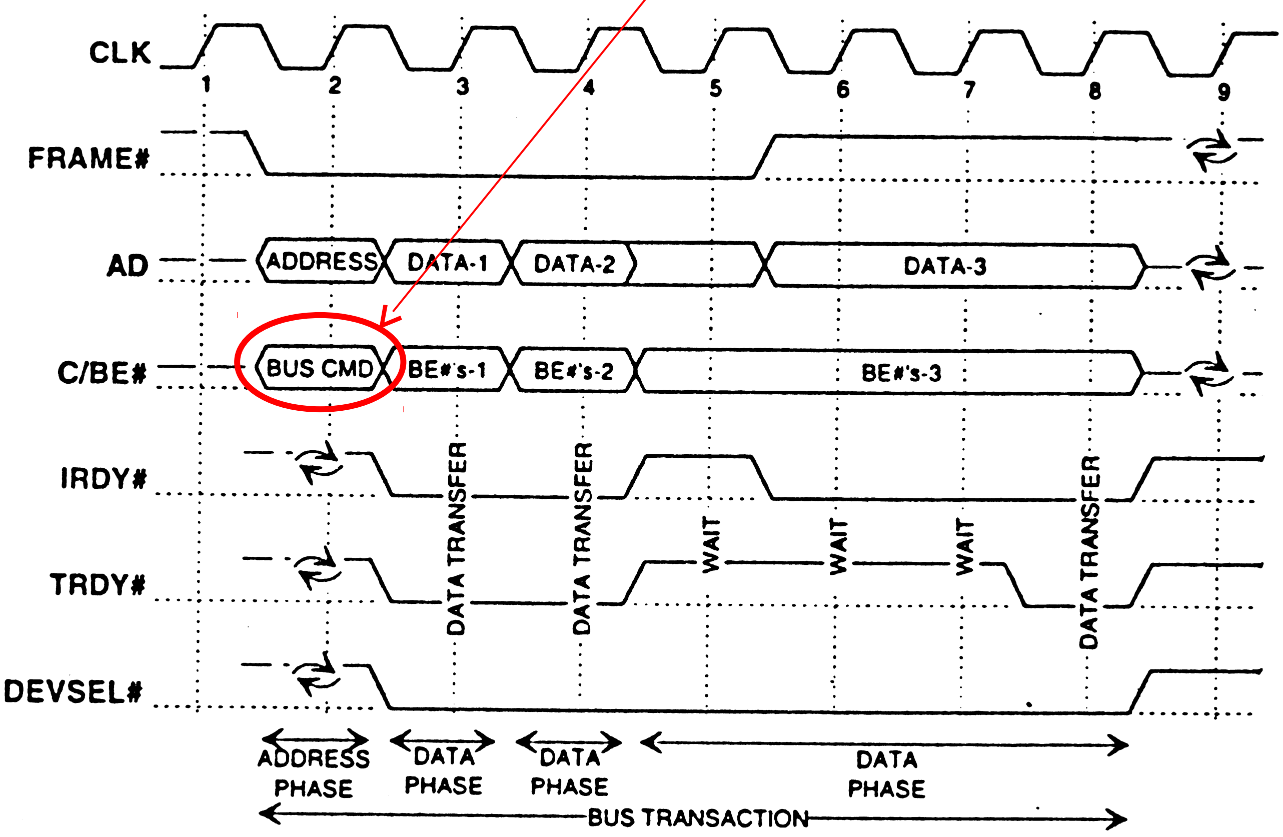
PCI terms and definitions II.

- The rising clock edge is reference/trigger point for all phase of the bus cycle/transaction timing
- Bus cycle is formed in most cases by
  - address phase
  - data phase
- Premature termination of data transfer is possible during bus cycle
- Transfer synchronization itself is pseudo-synchronous

Bus cycle kind/direction – command – specified by C/BE

C/BE[0::3]#	Bus command (BUS CMD)
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 (only 11 low addr bits for fnc and reg + IDSEL)
1011	Configuration Write (only 11 low addr bits for fnc and reg + IDSEL)
1100	Memory Read Multiple
1101	Dual Address Cycle (more than 32 bits for address – i.e. 64-bit)
1110	Memory Read Line
1111	Memory Write and Invalidate

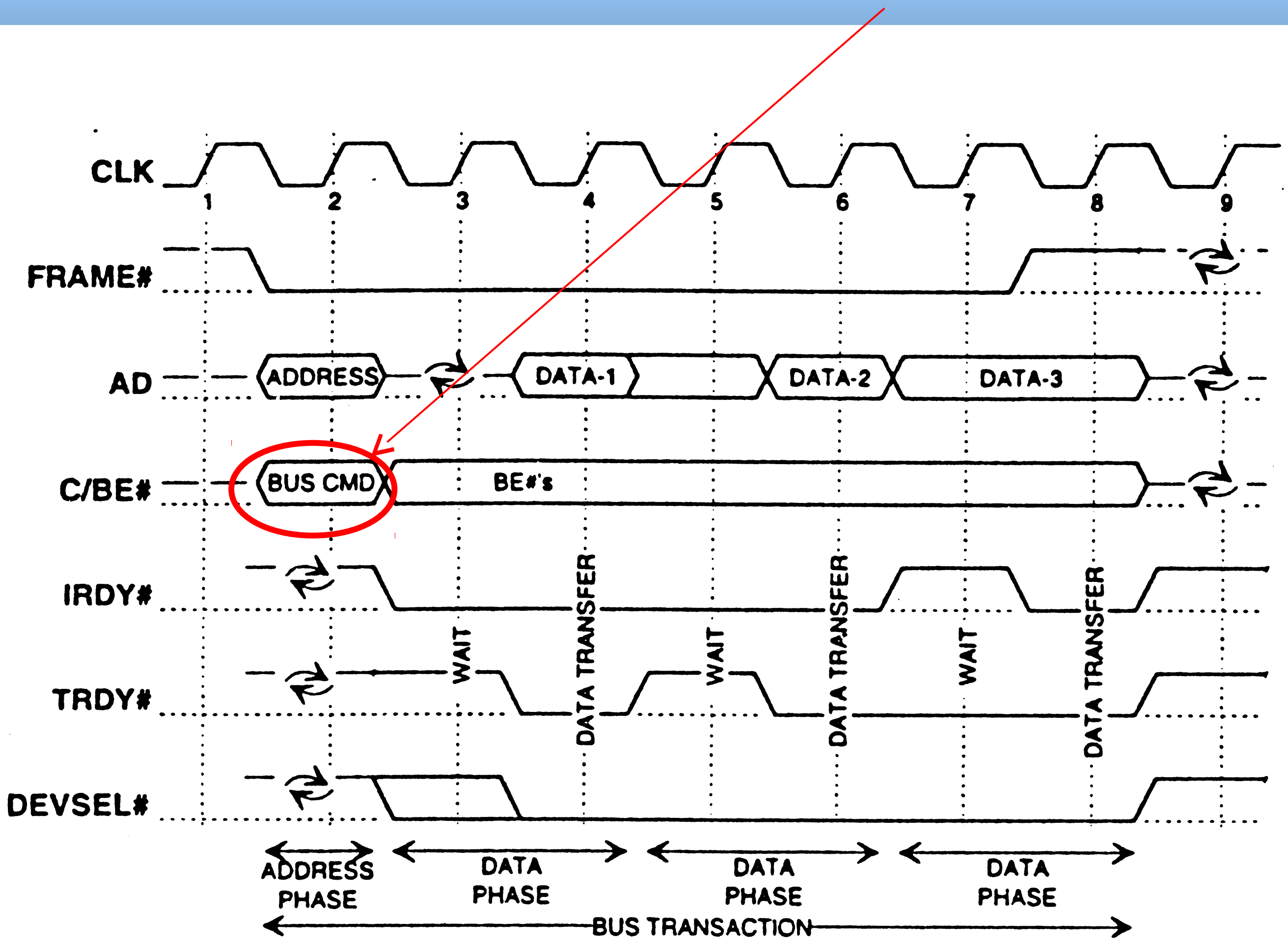
PCI bus memory **write** timing



Some remarks and observations

- The length of the transferred data block is controlled by the FRAME signal. It is negated (de-asserted) before last word transfer by initiator.
- Single world or **burst** transfer can be delayed by inserting wait cycles (pseudo-synchronous synchronization)!

PCI bus memory **read** timing

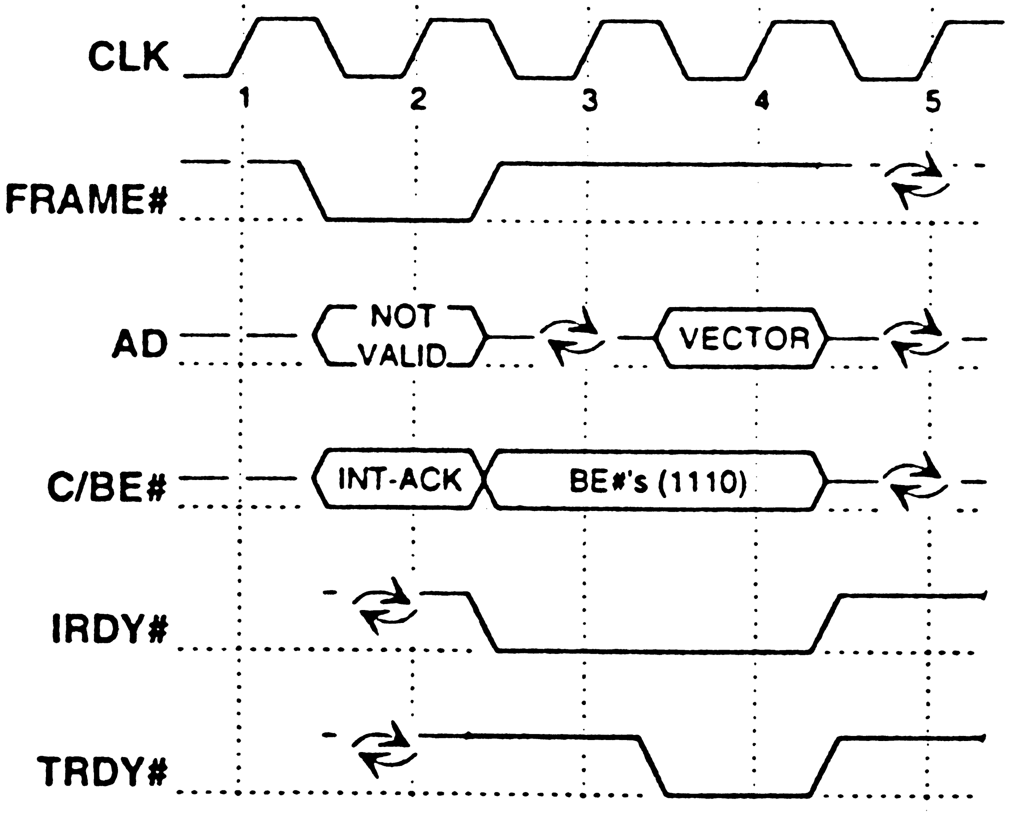


This transaction is going to be captured and shown by logic analyzer during labs

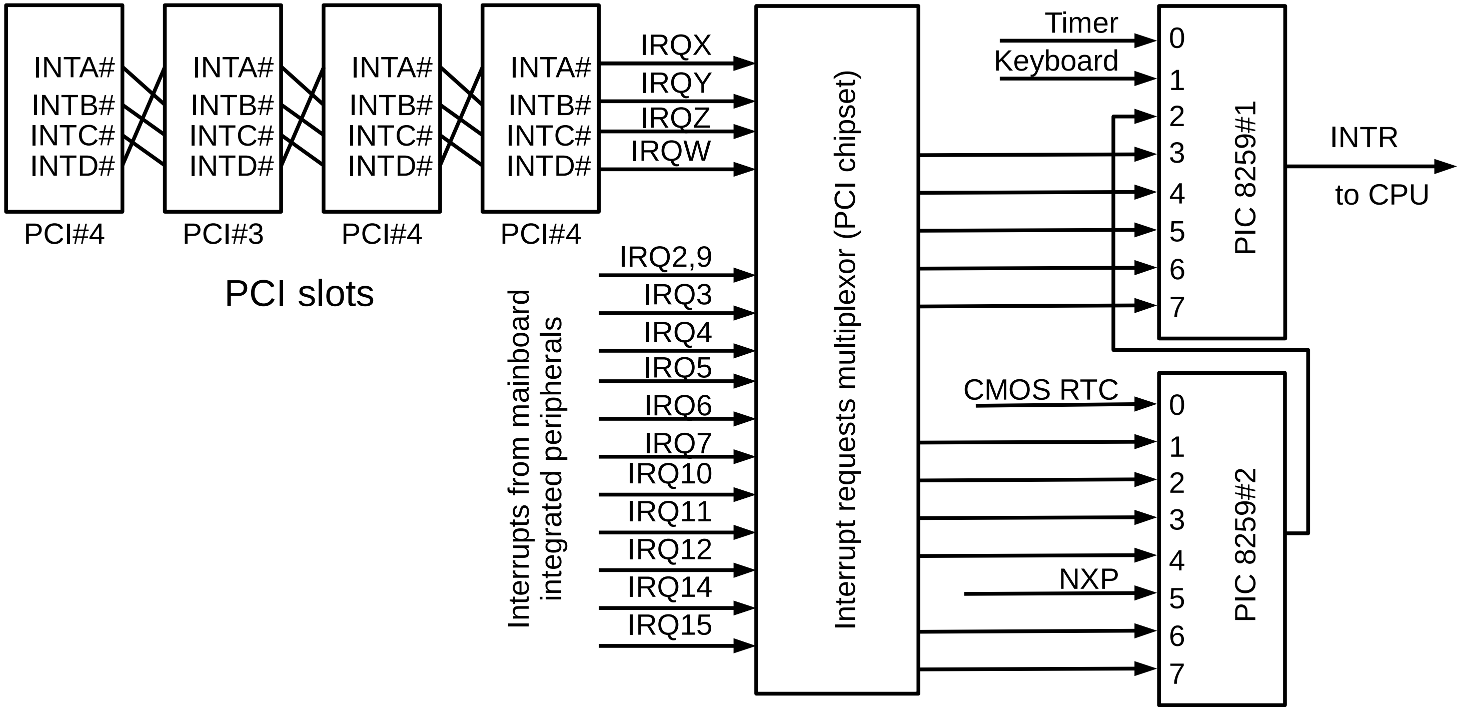
Interrupt acknowledge cycle

- Processor needs time to save interrupted execution state to allow state restoration after return from service routine
- Interrupt controller provides vector number (assigned to the asynchronous event) and CPU is required to translate it to the interrupt service routine start address
- All these activities require some time

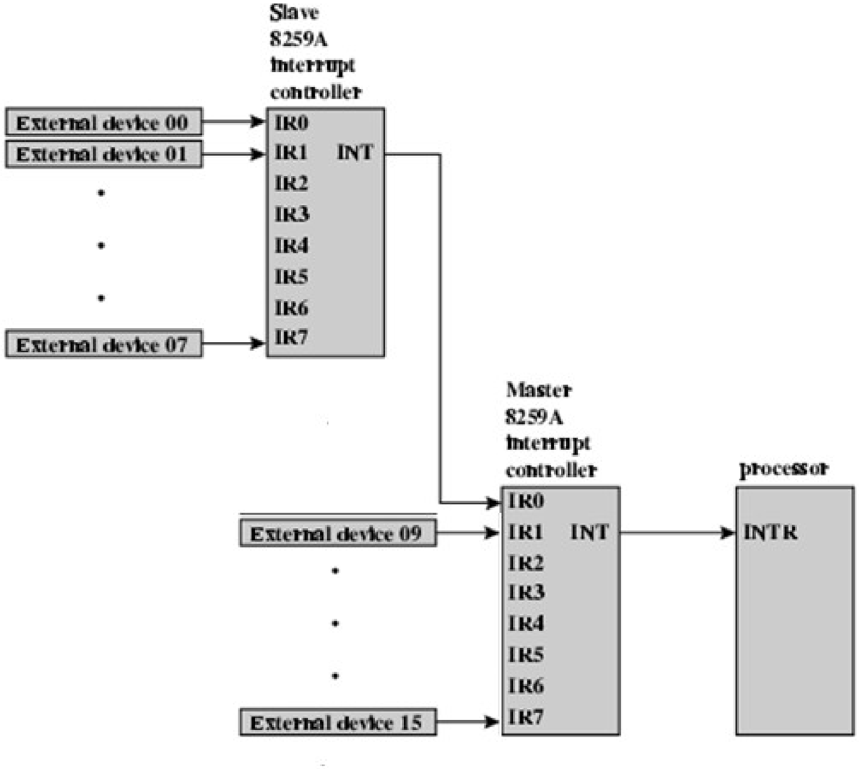
Interrupt acknowledge cycle timing



Some more details about standard PC PIC IRQ routing



Standard PC PIC interrupt vectors assignment



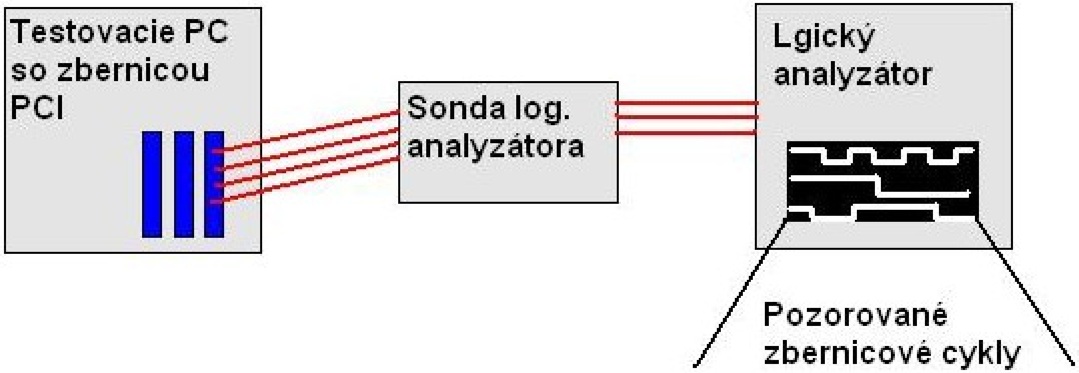
Common interrupt numbers for PC category computers:

IRQ	
0	Timer (for scheduler, timers)
1	Keyboard
2	i8259 cascade interrupt
8	Real-time clocks (CMOS wall time)
9	Available or SCSI controller
10,11	Available
12	Available or PS/2 mouse
13	Available or arithmetics co-processor
14	1-st IDE controller
15	2-nd IDE controller
3	COM2
4	COM1
5	LPT2 or available
6	Floppy disc controller
7	LPT1

Recapitulation of the bus description steps

- Notice: we started PCI description by bus topology analysis, PCI signals and then we moved to timing diagrams
  - simpler cases first (read and write)
  - then how bus access is arbitrated
  - the special functions (interrupt acknowledge) last
- Doc. Šnorek recommends: always follow these steps when trying to learn new bus technology.

PCI bus timing laboratory exercise

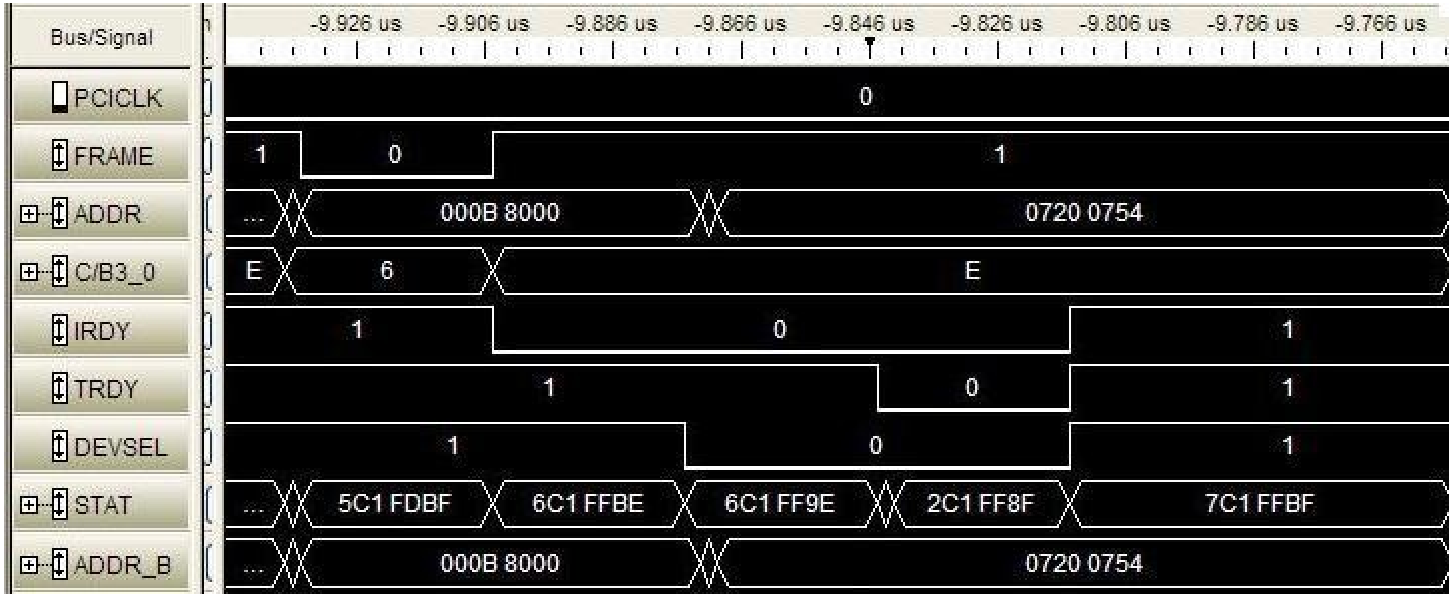


Testovacie PC so zbernicou PCI

Sonda log. analyzátora

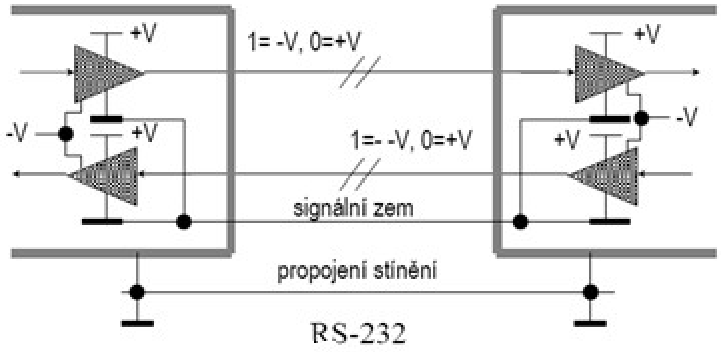
Lgický analyzátor

Pozorované zbernicové cykly



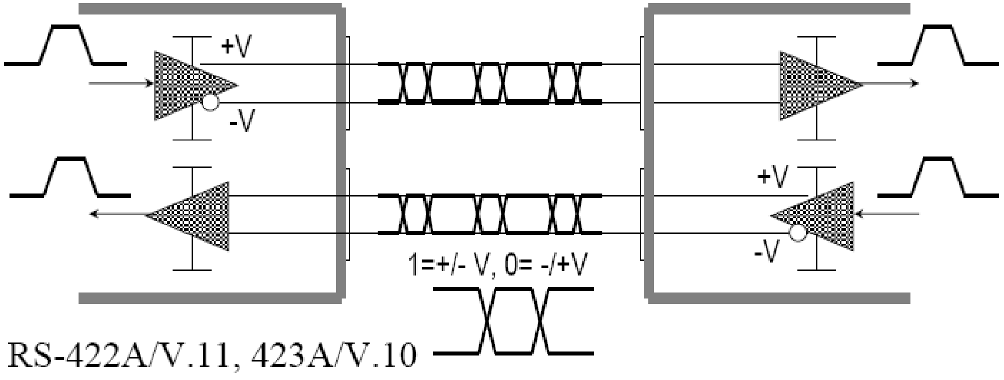
Some more notes regarding the bus hardware realization

How signals are transferred



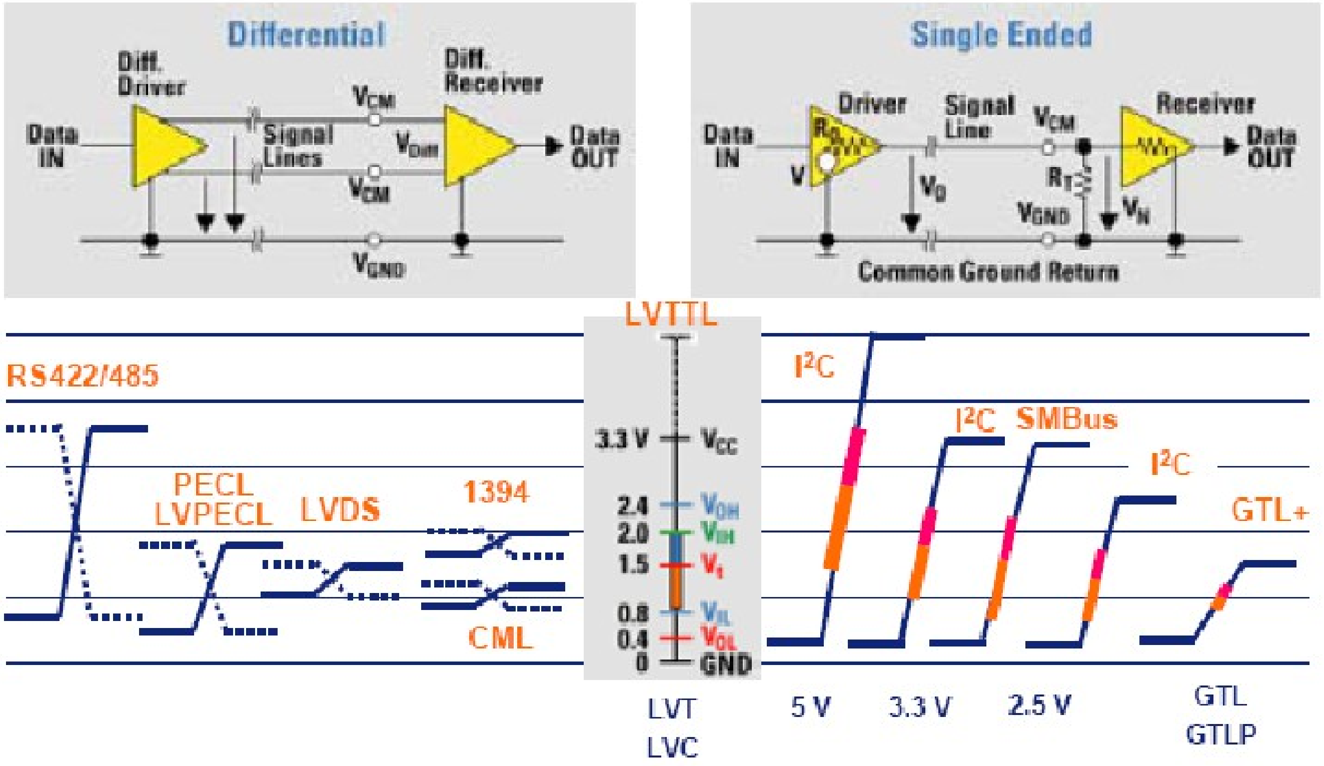
English term
Signalling

Single ended (asymmetric) versus.
differential (symmetric)



Typical signaling levels and some speed considerations

DesignCon 2003 TecForum I²C Bus Overview

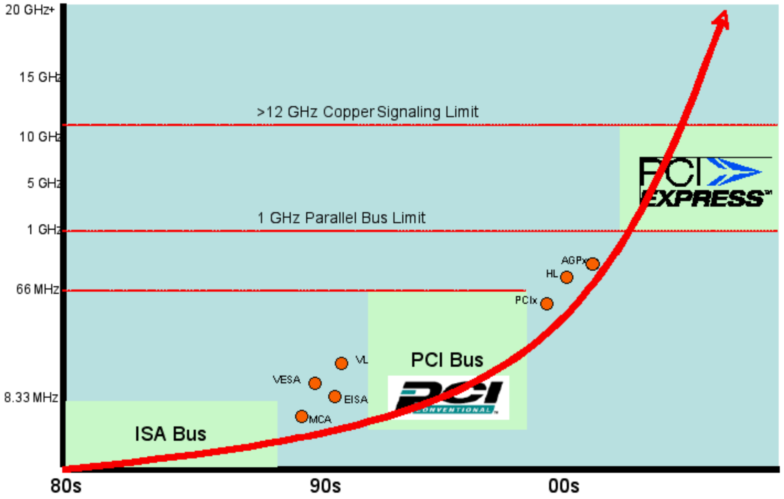


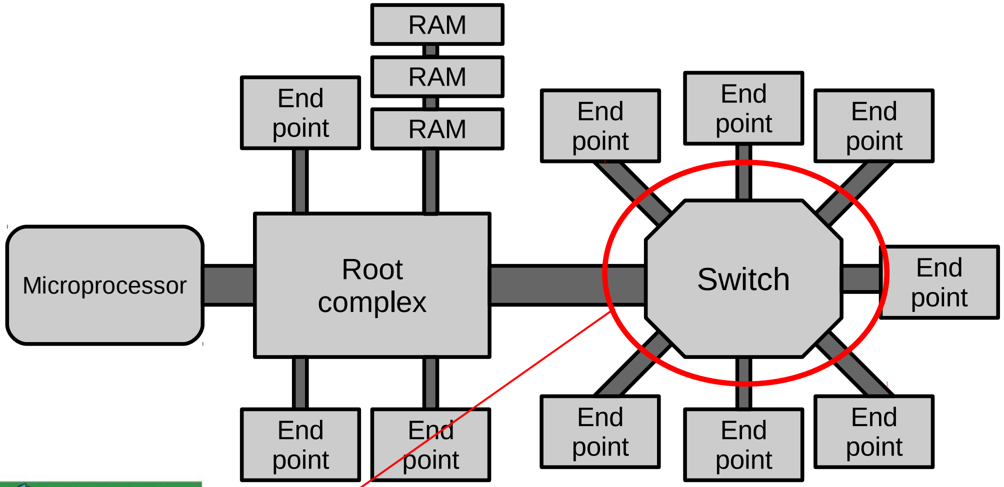
Differential signaling

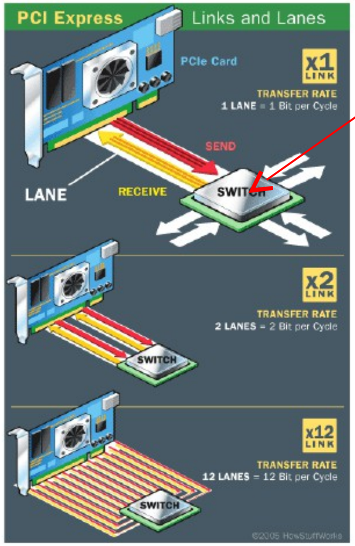
Single ended signaling

PCIe

PCIe architecture

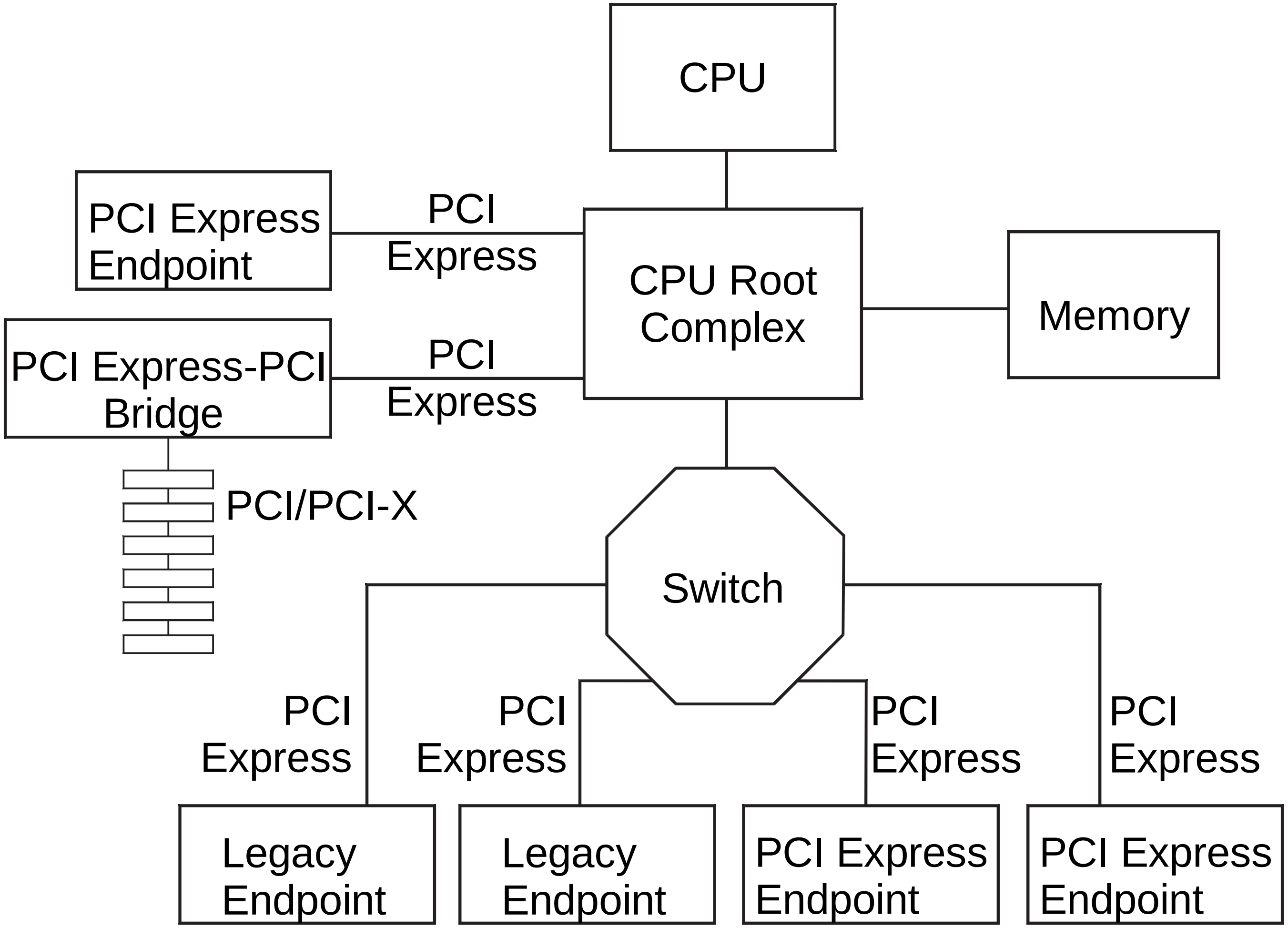



RAM
RAM
RAM
End point
Microprocessor
Root complex
Switch
End point
End point
End point
End point
End point
End point
End point
End point
End point

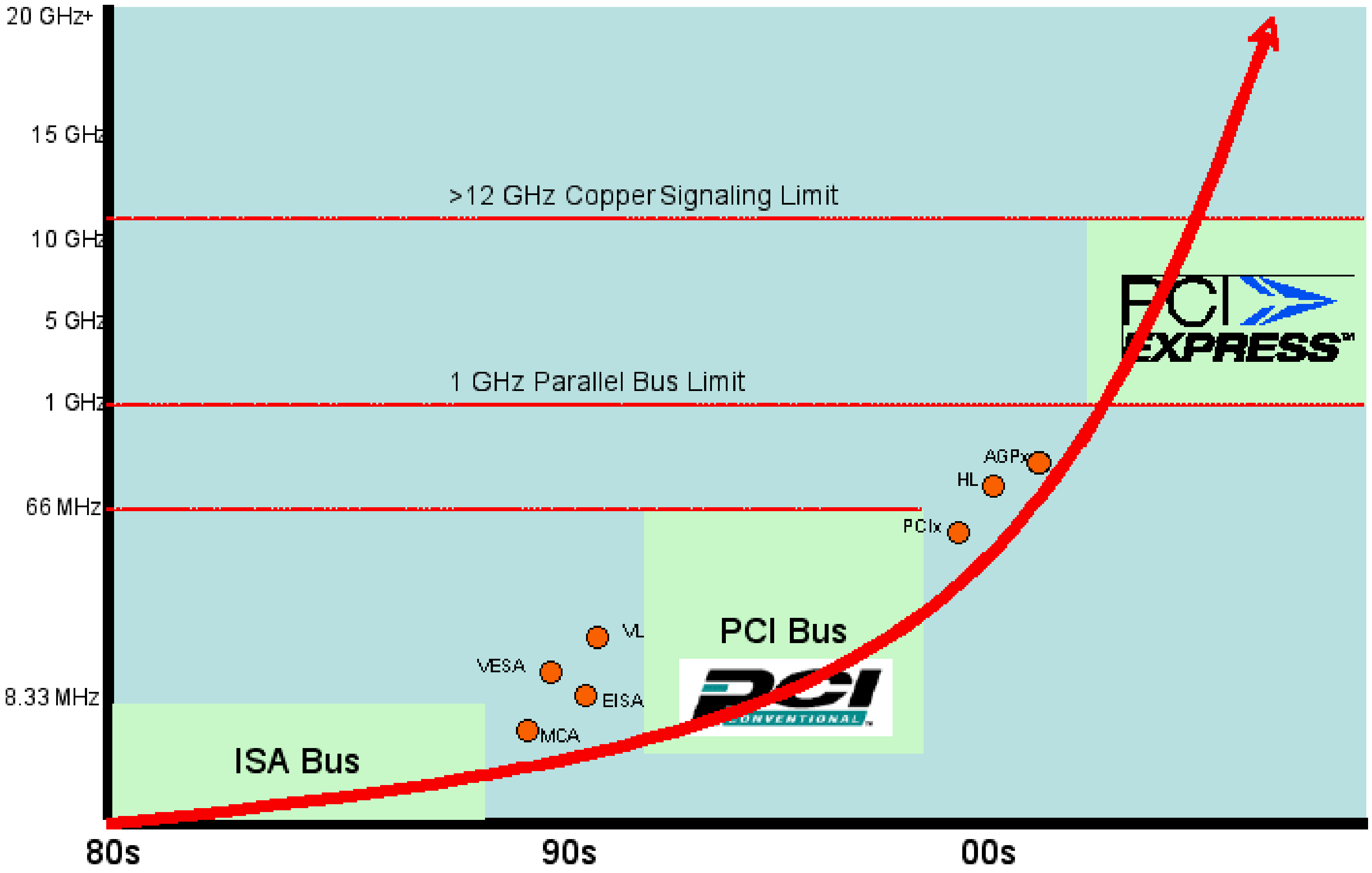


Source:
http://computer.howstuffworks.com/pci-express.htm

PCIe topology and components

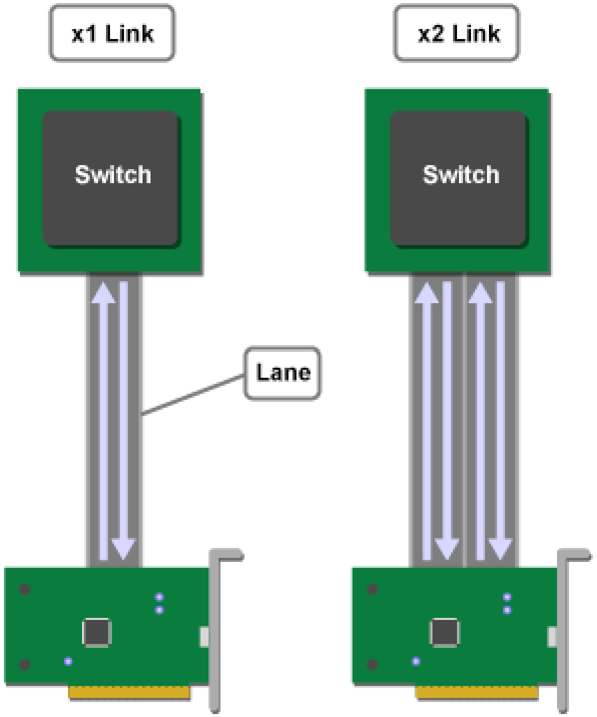


Why switch to serial data transfer



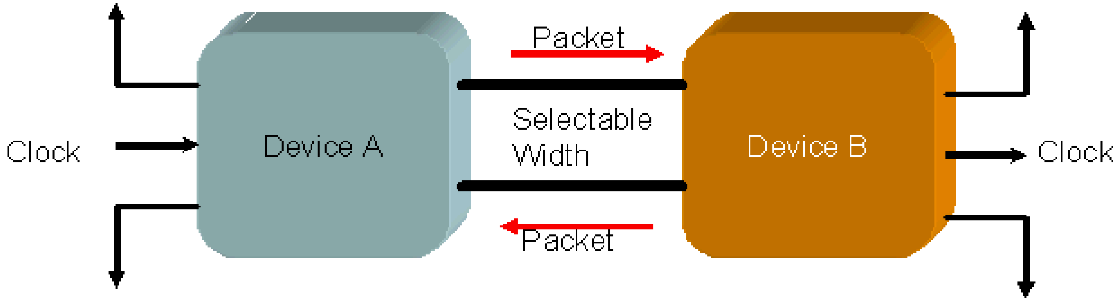
PCIe transfers signaling, PCIe lanes

- Link interconnect switch with exactly one device
- Differential AC signaling is used – two wires for single direction
- Each link consists of one or more lanes
- Lane consists of two pair of wires
- One pair for Tx and other one for Rx
- Data are serialized by 8/10 code
- The separate pairs allow full-duplex operation/transfers

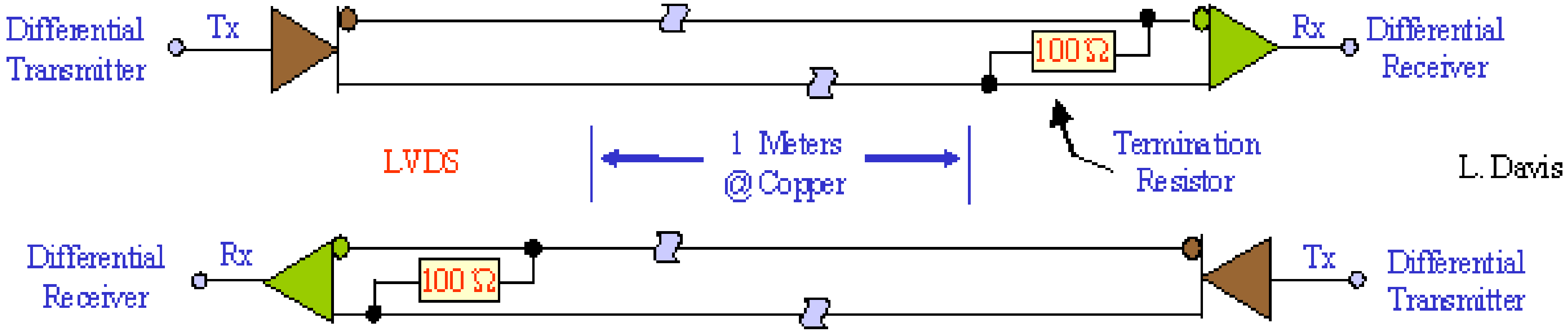


PCIe physical link layer

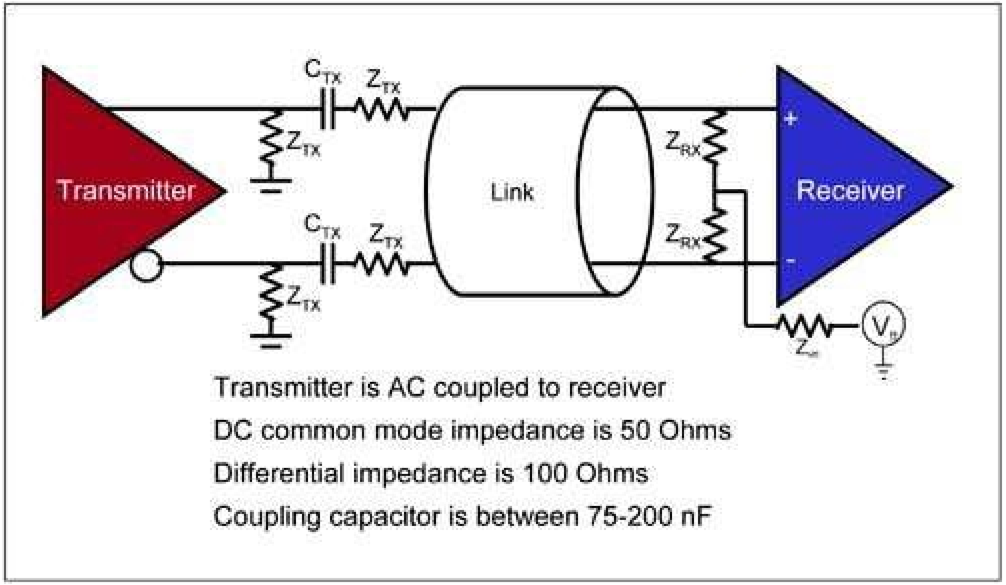
Differential full-duplex physical layer



8b/10b encoding provides enough edges for clock signal reconstruction/synchronization and balanced number of ones and zeros. This ensures zero common signal (DC) and AC (only) coupling is possible



PCIe physical layer model



Transmitter is AC coupled to receiver

DC common mode impedance is 50 Ohms

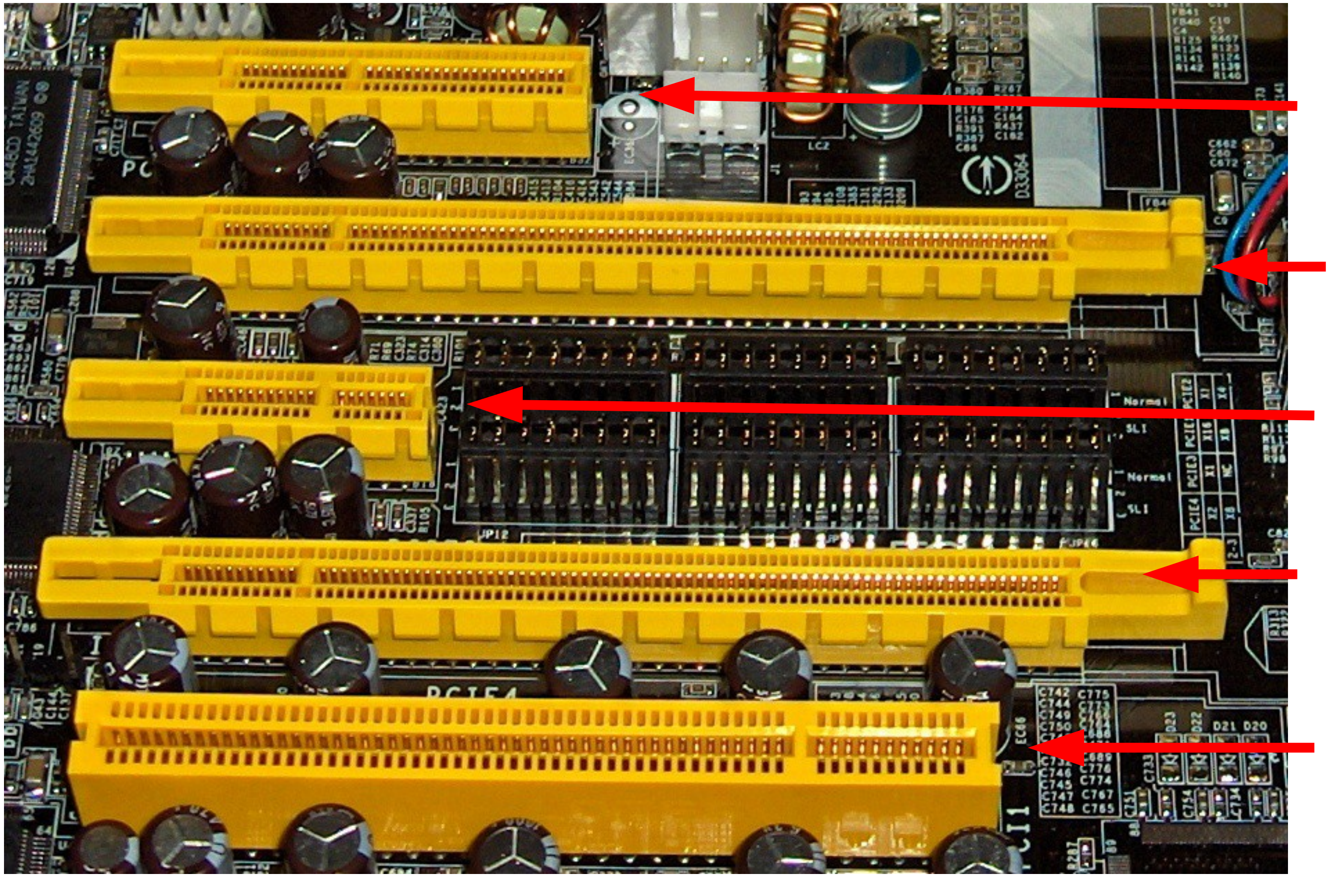
Differential impedance is 100 Ohms

Coupling capacitor is between 75-200 nF

Source: Budruk, R., at all: PCI Express System Architecture

PCIe characteristics

- 2.5 GHz clock frequency (1 GT/s), raw single link single direction bandwidth 250 MB/s (can be multiplied by parallel lanes – 2×, 4×, 8×)
- Single link efficient data rate is 200 MB/s, this is 2× … 4× more than for classic PCI
- The bandwidth is not shared, point to point interconnection
- Two pairs of wires, differential signaling
- Data are encoded (modulated) using 8b/10b code
- Expected up to 10 GHz clocks due technology advances
- PCI Express 2.x (2007) allows 5 GT/s (5 GHz clock)
- PCI Express 3.x (started at 2010) increases it to 8 GT/s
- Encoding changed from 8b/10b (20% bandwidth used by encoding) to "scrambling" and 128b/130b encoding (takes only 1.5% of the bandwidth)



PCIe x4

PCIe x16

PCIe x1

PCIe x16

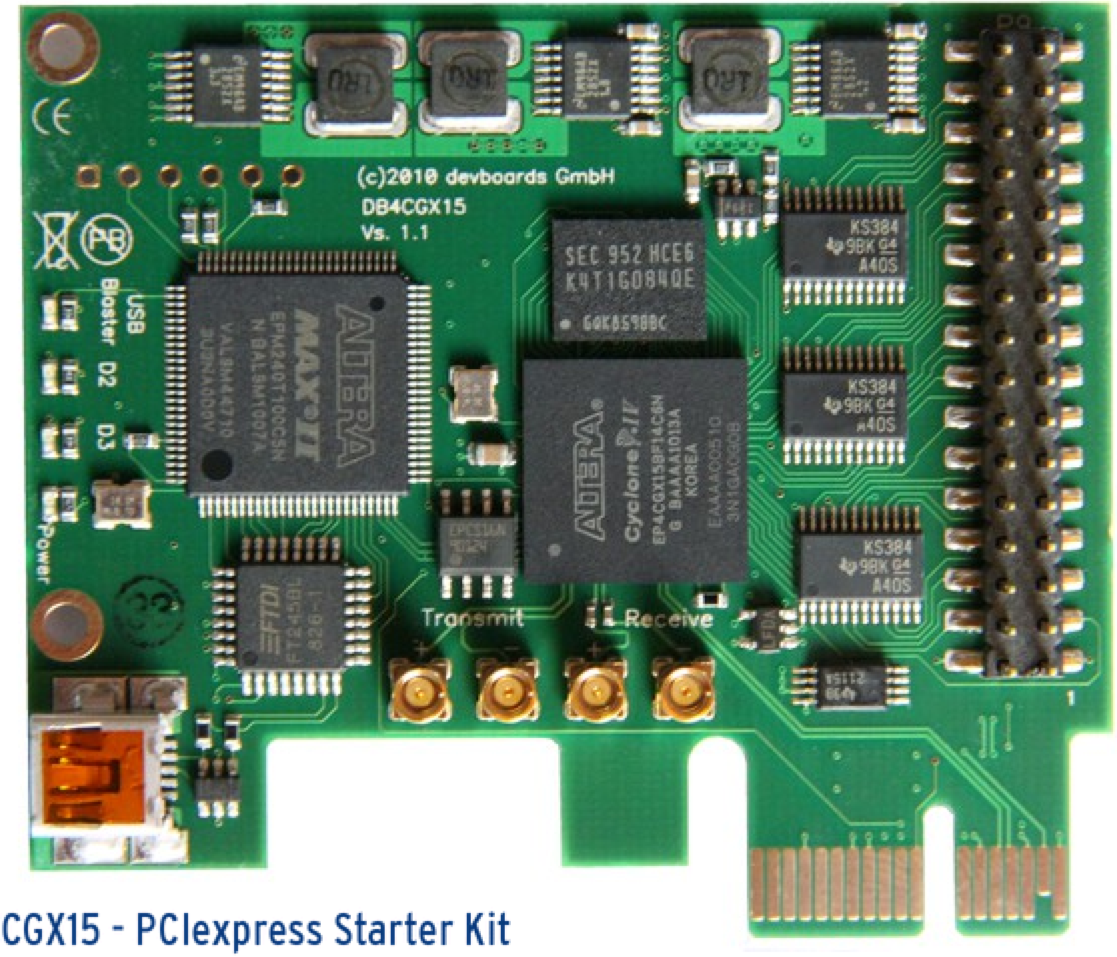
32-bit classic PCI slot

Picture source: Wikipedia

LanParty nF4 Ultra-D mainboard
from DFI

The PCIe board DB4CGX15 used in your semester work

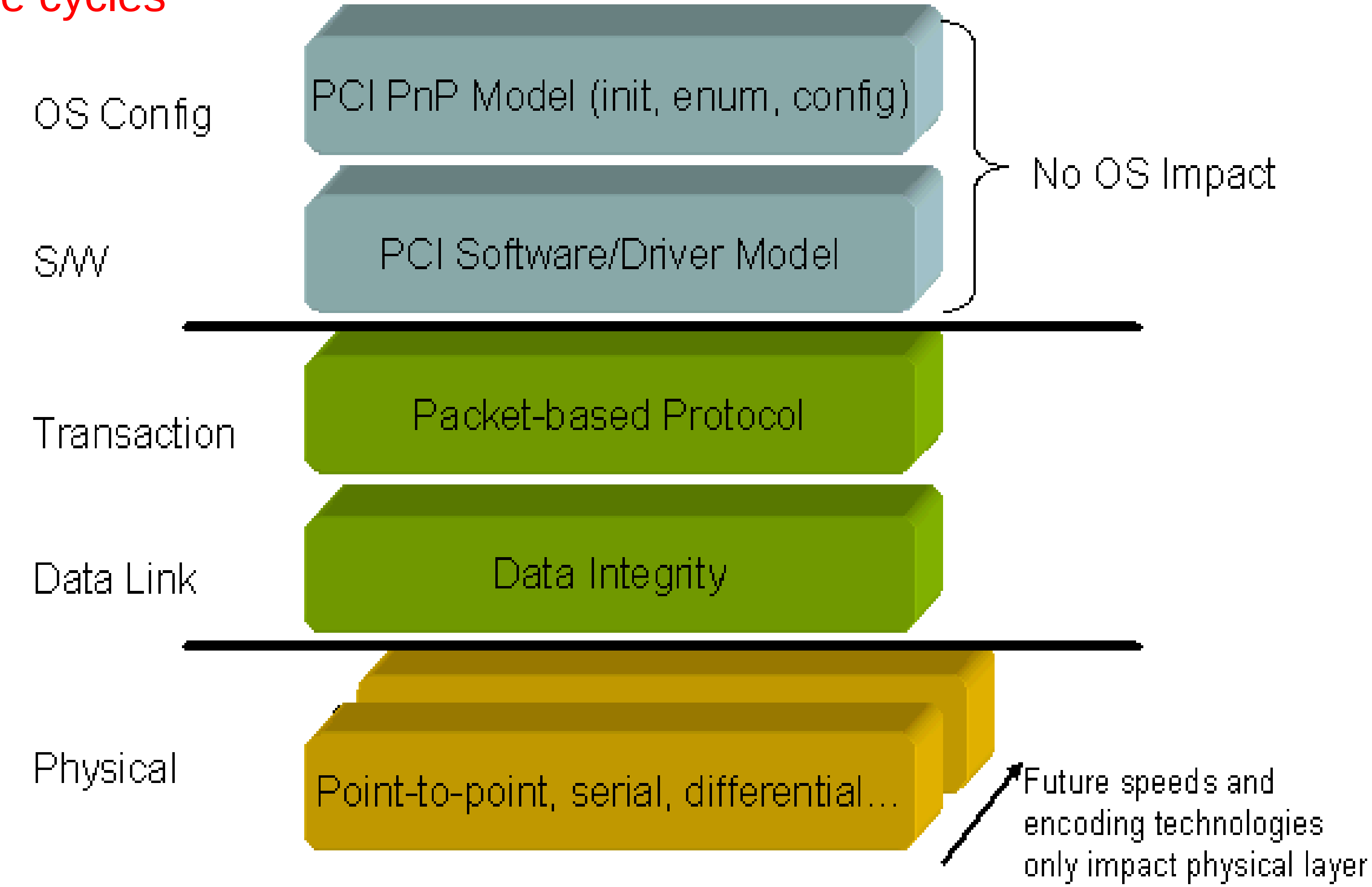
- PCIe x1
- Altera EP4CGX15 BF14C6N FPGA
- EPCS16 Configuration device
- 32Mbyte DDR2 SDRAM
- 20 I/O Pins
- 4 Input Pins
- 2 User LEDs



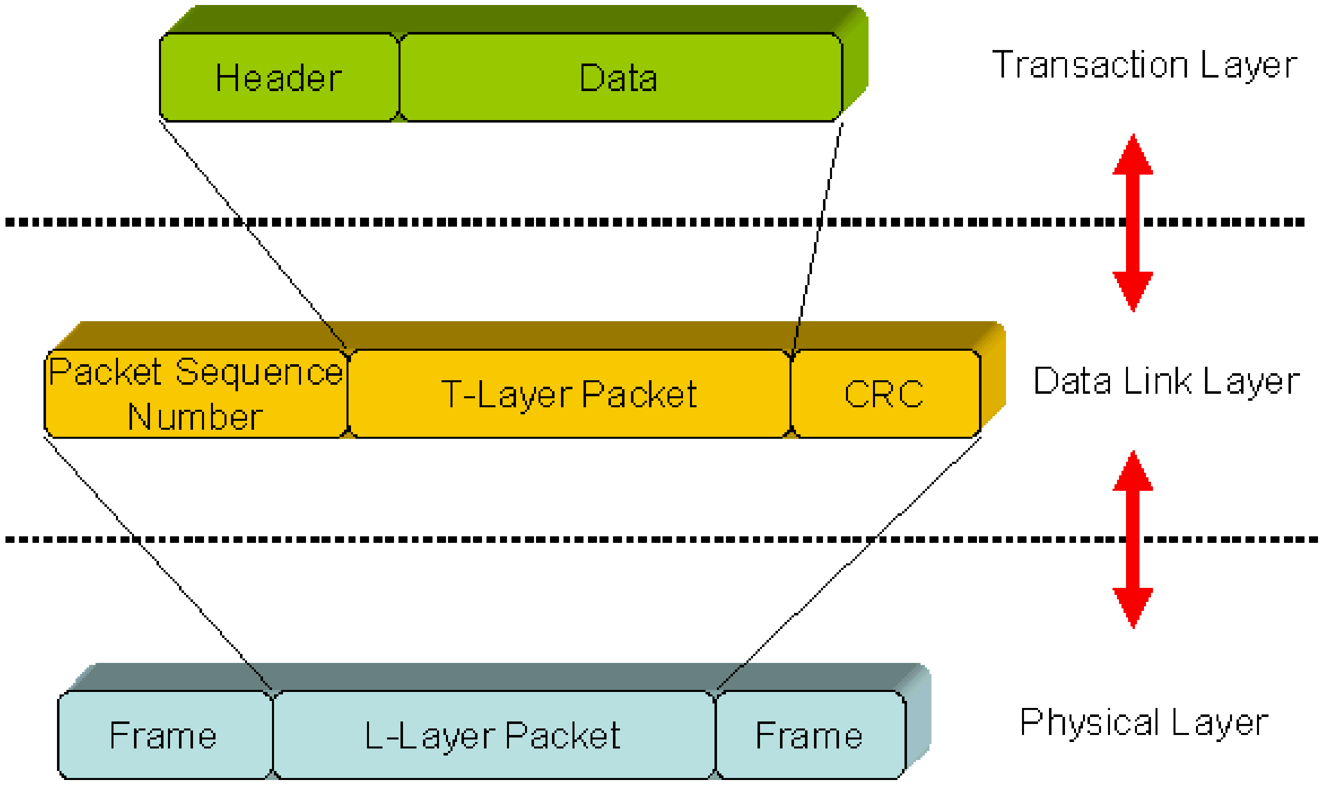
DB4CGX15 - PCIexpress Starter Kit
Development board for PCIexpress applications

PCIe communication protocol – hardware and software layers

PCIe physical topology is serial, point-to-point, packet oriented, but its logical view and behavior is the same as PCI – that is multi-master bus, same enumeration, read/write cycles


OS Config
PCI PnP Model (init, enum, config)
S/W
PCI Software/Driver Model
No OS Impact
Transaction
Packet-based Protocol
Data Link
Data Integrity
Physical
Point-to-point, serial, differential…
Future speeds and encoding technologies only impact physical layer

Data packet structure and transport layers



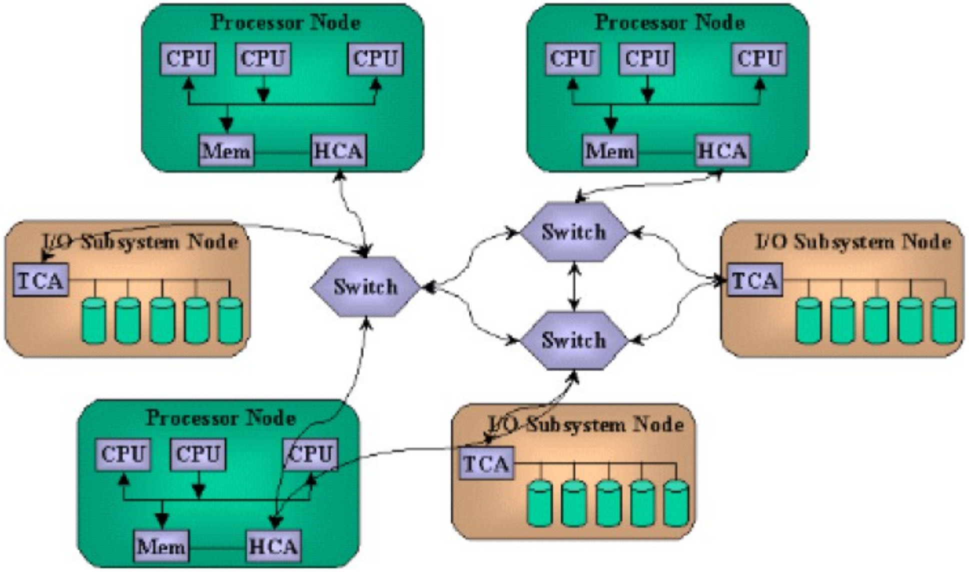
Source: http://zone.ni.com/devzone/cda/tut/p/id/3767#toc0

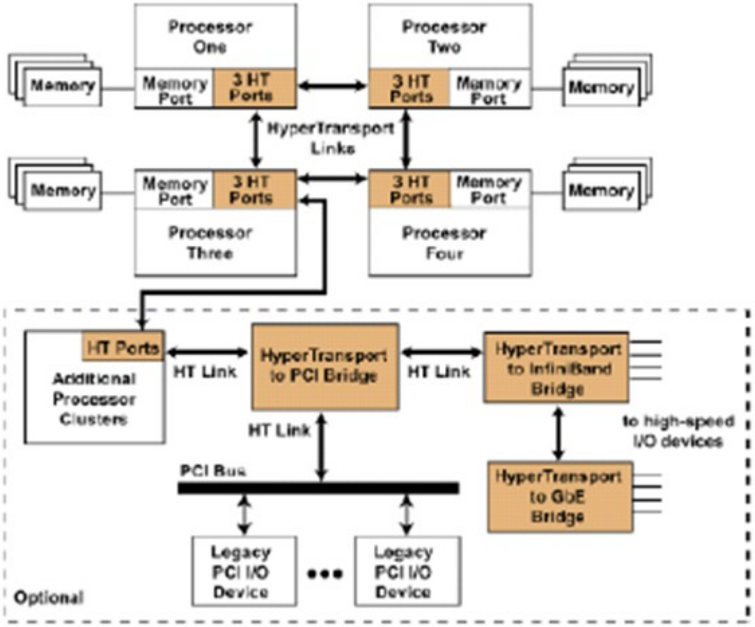
Other, more advanced, architectures and PC future

What are future options?

- In PC development, 15 years is quite a long time and there have been significant successes, but what about the future?
- There are already other technologies in use (some in PC systems already), many originating from supercomputers:
  - PCI-X,
  - HyperTransport
  - InfiniBand
  - QPI
- More:
  - Advanced Computer Architectures (AE4M36PAP) course

InfiniBand vs. HyperTransport

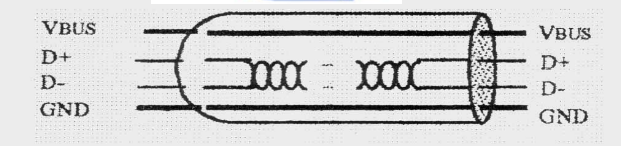


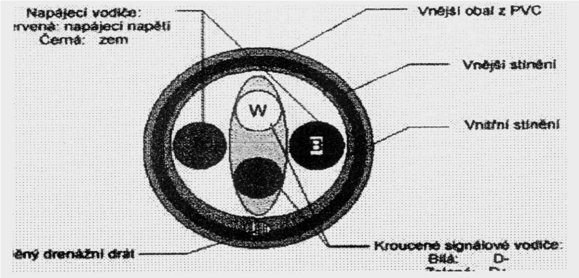


Miscellaneous information regarding buses

Nowadays, using high speed, serial bus is quite common.

HI-SPEED CERTIFIED USB



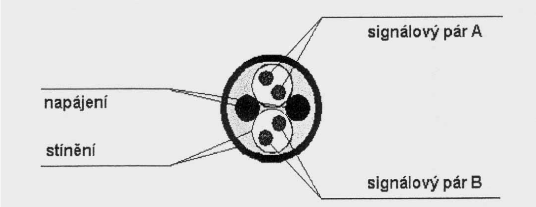


Replacement of slow RS-232 serial, PS/2. General purpose now: disc, video, etc.

SERIAL ATA

Switch from parallel ATA

FireWire



DVI-D, HDMI

switch from analog VGA/ RGB

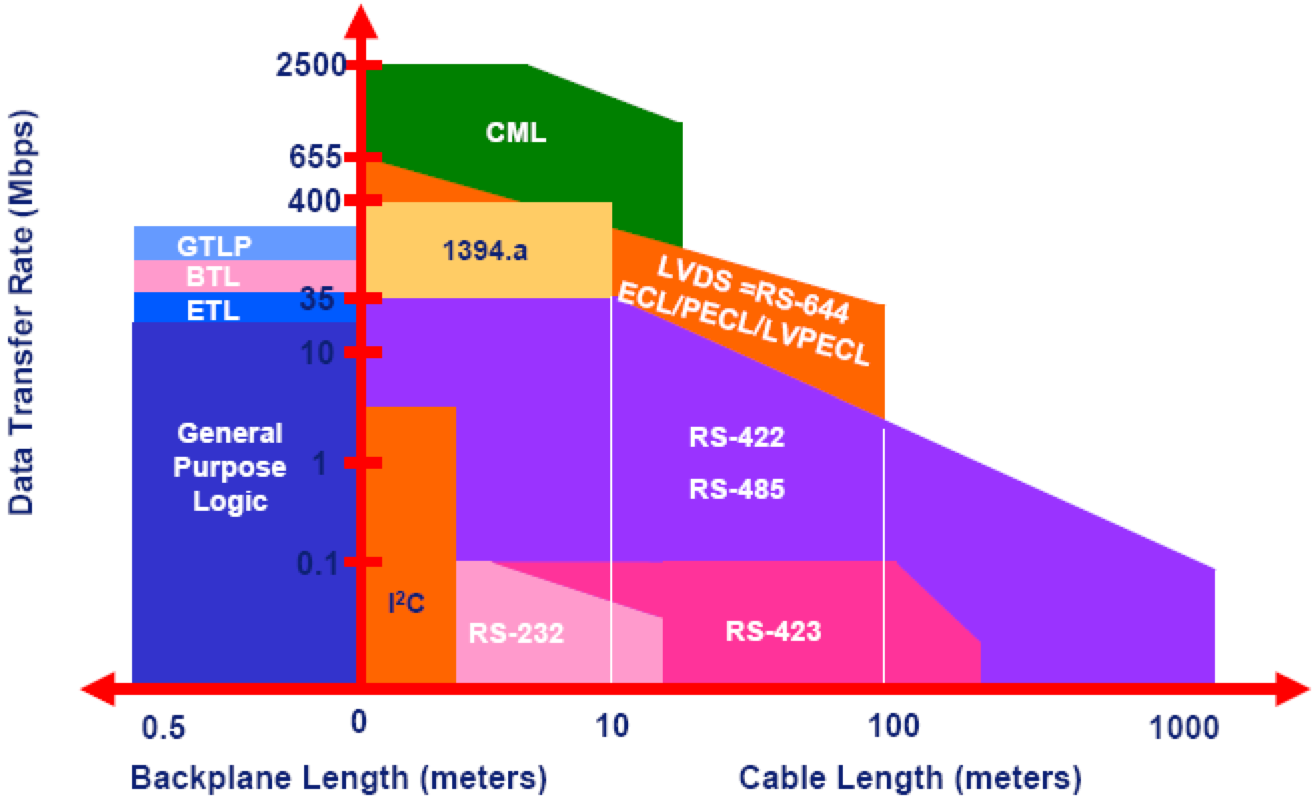
Serial attached SCSI

switch from parallel (wide, differential) SCSI

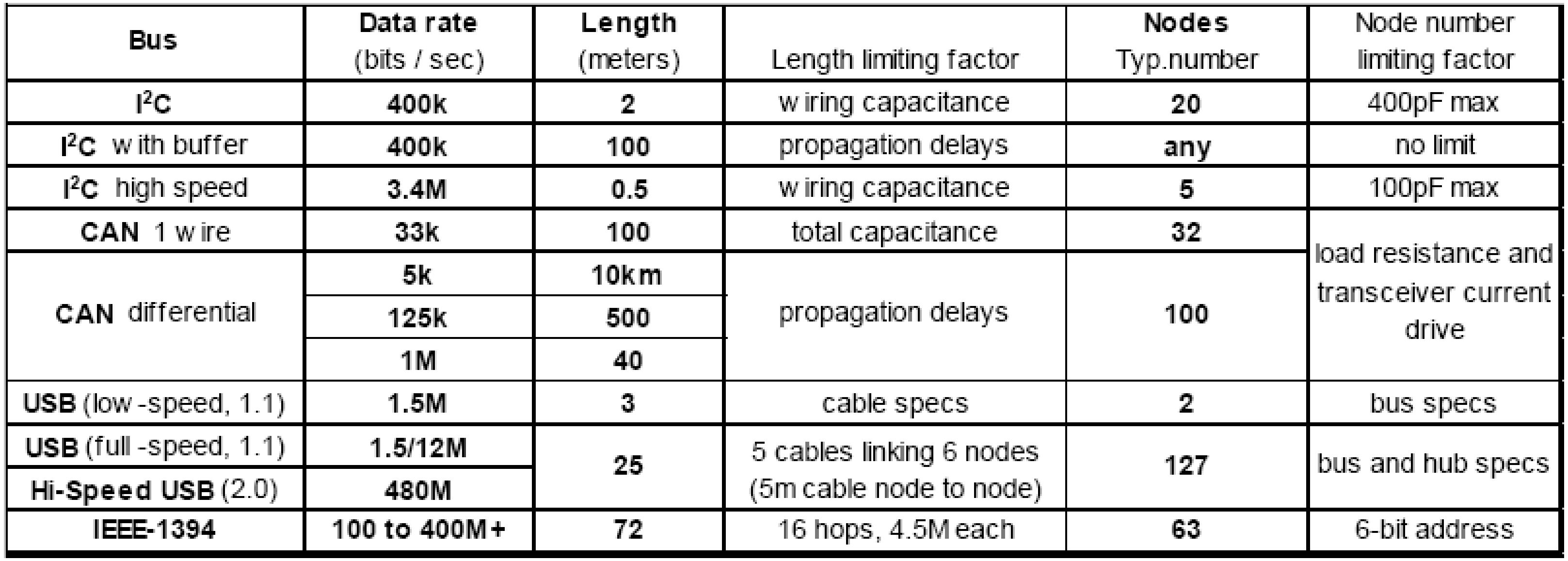
Key characteristics of the dominant bus technologies

Parameter	**Firewire**	**USB 2.0**	**PCI Express**	**SATA**	**Serial Attached SCSI**
Expected use	external	internal	internal	internal	both
Max. nodes number	63	127	1	1	4
No. of wires (fundamental only)	4	2	4	4	4
Raw bandwidth	50 MB/s 100 MB/s	0,2 MB/s 1,5 MB/s 60 MB/s	250 MB/s 1x	300 MB/s	300 MB/s
Hot-plug attach?	yes	yes	(yes)	yes	yes
Max. bus distance	4,5 m	5 m	0,5 m	1 m	8 m
Official standard designation	IEEE 1394	USB	PCI-SIG PCI	SATA-IO	T10 committee

Key characteristics – another view



Another comparison of bus standards



Bus	**Data rate** (bits / sec)	**Length** (meters)	Length limiting factor	**Nodes** Typ.number	Node number limiting factor
I^2C	**400k**	**2**	wiring capacitance	**20**	400pF max
I^2C with buffer	**400k**	**100**	propagation delays	**any**	no limit
I^2C high speed	**3.4M**	**0.5**	wiring capacitance	**5**	100pF max
CAN 1 wire	**33k**	**100**	total capacitance	**32**	load resistance and transceiver current drive
CAN differential	**5k**	**10km**	propagation delays	**100**	
	125k	**500**			
	1M	**40**			
USB (low-speed, 1.1)	**1.5M**	**3**	cable specs	**2**	bus specs
USB (full-speed, 1.1)	**1.5/12M**	**25**	5 cables linking 6 nodes (5m cable node to node)	**127**	bus and hub specs
Hi-Speed USB (2.0)	**480M**				
IEEE-1394	**100 to 400M+**	**72**	16 hops, 4.5M each	**63**	6-bit address