Fundusze Europejskie
Wiedza Edukacja Rozwój

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Społeczny

Politechnika Śląska jako Centrum Nowoczesnego Kształcenia
opartego o badania i innowacje

POWR.03.05.00-IP.08-00-PZ1/17

Projekt współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego

Microprocessor and Embedded Systems

Faculty of Automatic Control, Electronics and Computer Science, Informatics, Bachelor Degree

Lecture 7

From EISA to PCI

Bartłomiej Zieliński, PhD, DSc

From EISA to PCI

Program:

- EISA bus
- MCA architecture
- VLB architecture
- PCI architecture
- PCI signals

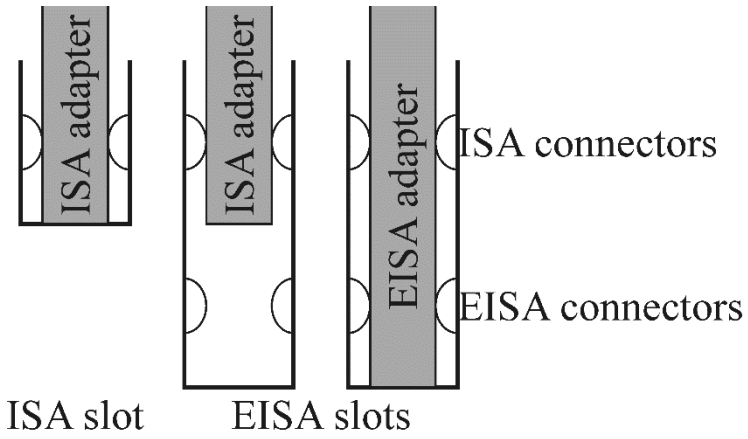
From EISA to PCI

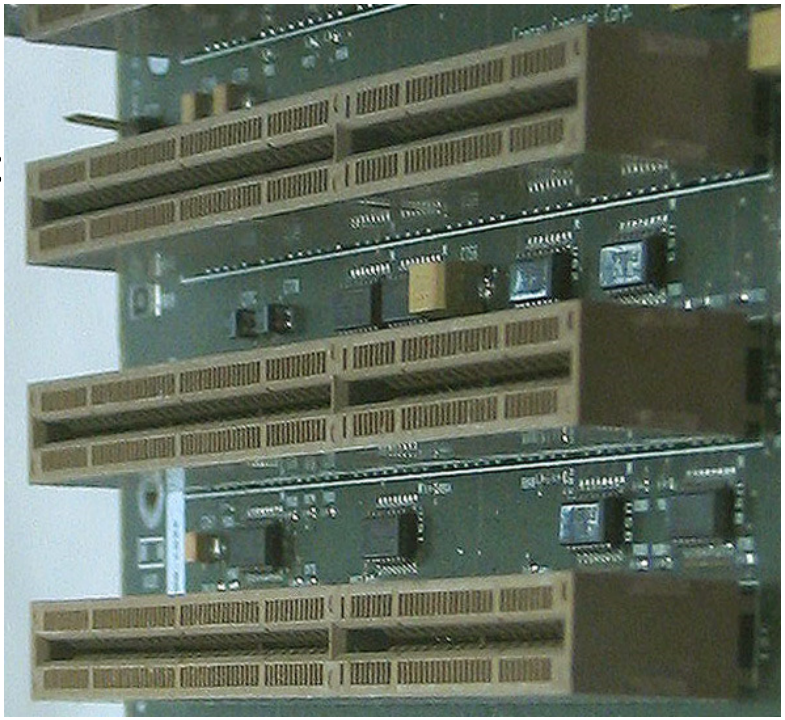
- PC architecture deployment

Bus	Data bits	Frequency [MHz]	Throughput [MBps]
ISA	8 or 16	8.33	8.33
EISA	32	8.33	33
MCA	32	10	20
VLB	32	40	120
PCI 2.0	32	33	132
PCI 2.0	64	33	264
PCI 2.1	32	66	264
AGP	32	66	264
AGP	32	133	532
AGP	32	266	1064

From EISA to PCI

- EISA
  - **E**xtended **I**ndustry **S**tandard **A**rchitecture
  - Backward compatible with ISA
    - 16/32-b recognition
  - ISA cards work in EISA slots
    - 2-level contact slot
    - 98 ISA lines + 90 EISA-specific





From EISA to PCI

- EISA
  - Additional signals
    - $LA_{2..16}$, $LA_{24..31}$
      - $LA_{17..23}$ present in ISA
      - Not buffered – smaller delays
    - $D_{16..31}$, $BE_{0..3}$
    - EXE16, EXE32
      - Force 16- or 32-b transfer
      - Default = 8-b for compatibility

From EISA to PCI

- EISA
  - 32-b DMA, 8/16/32-b transfers, incl. 8237-like
    - 7 channels
  - 15 IRQ
  - F_{clk}=8.33 MHz (ISA)
    - Memory access = CPU clock
    - 1 clk / cycle

From EISA to PCI

- EISA
  - Configuration memory
    - 4 KB
    - Software-only device installation
    - Configuration program for a card
    - Numbered sockets
      - MoBo: 0
      - Slots: 1-15
      - MoBo resources: MaxSlot+1..63
    - BIOS tries to resolve resources conflicts

From EISA to PCI

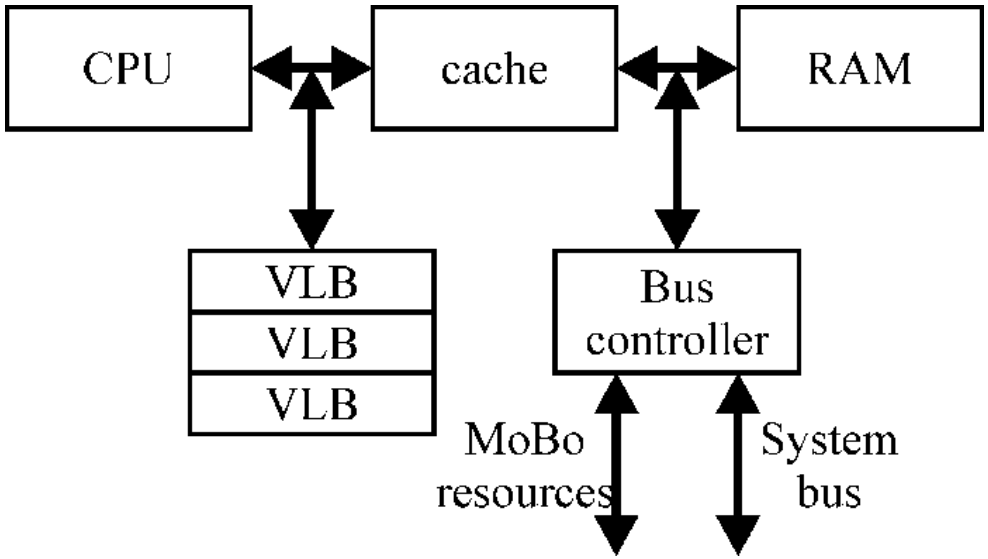
- MCA
  - MicroChannel Architecture (IBM PS/2)
  - 10 MHz, 2 clk/cycle
  - 32-b DMA (no 8237-like)
  - CPU + up to 16 µp on expansion cards
    - 4 arbitration bits
  - Software-only device configuration
    - Card numbers
    - Socket numbers
    - Software card disconnection possible

From EISA to PCI

- VLB
  - VESA Local Bus
    - Video Electronics Standards Association
  - Devices connected directly to µp's local bus
    - 80386, 80486, (also Pentium?)
  - Fclk=CPU clk
  - 1-3 slots
    - Number of slots depends on frequency (?)
      - 50 MHz – 1 slot, 40 MHz – 2 slots, 33 MHz – 3 slots
  - Card types
    - Graphics card, HDD/FDD controllers, network cards

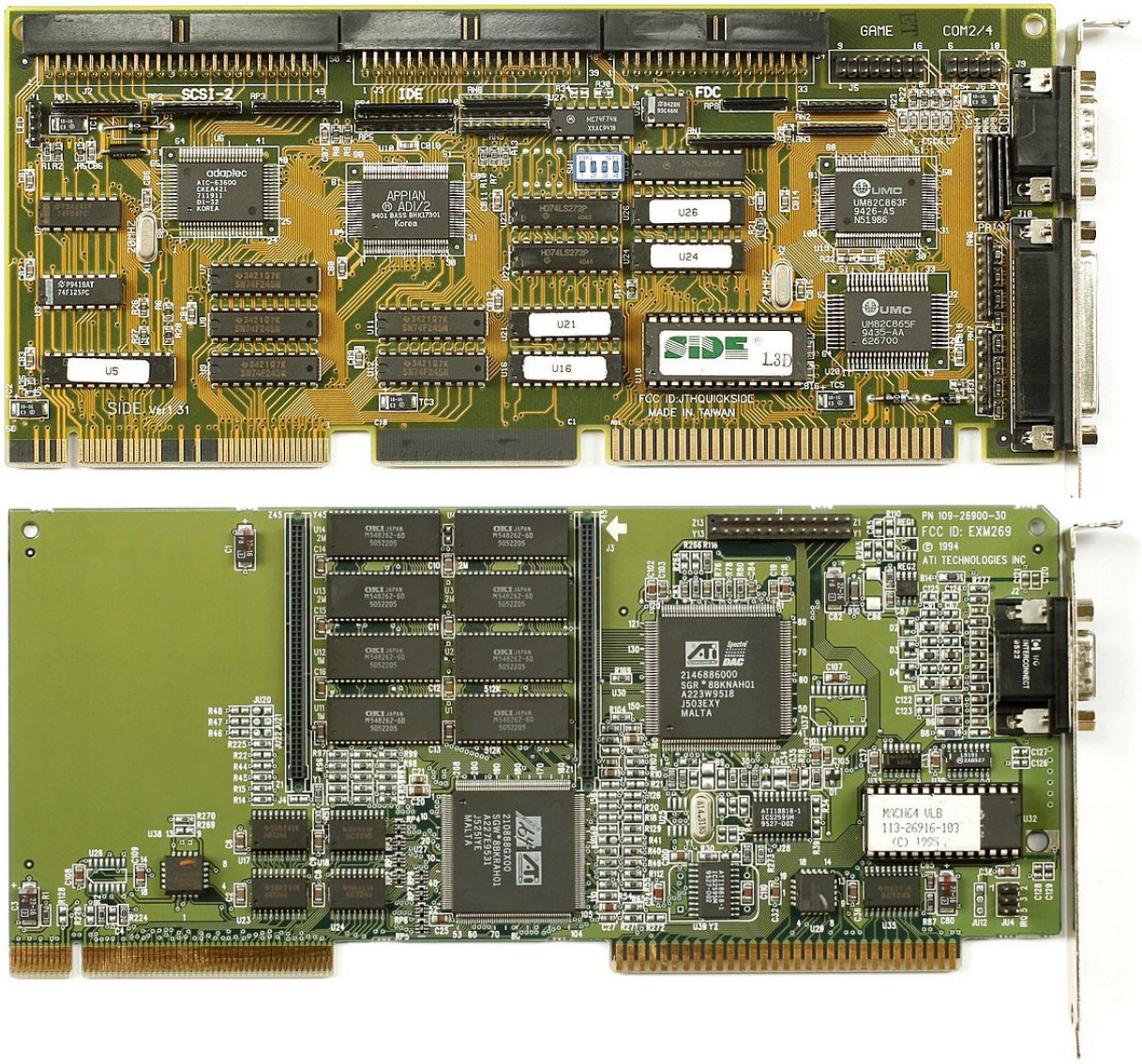
From EISA to PCI

- VLB



From EISA to PCI

- VLB



From EISA to PCI

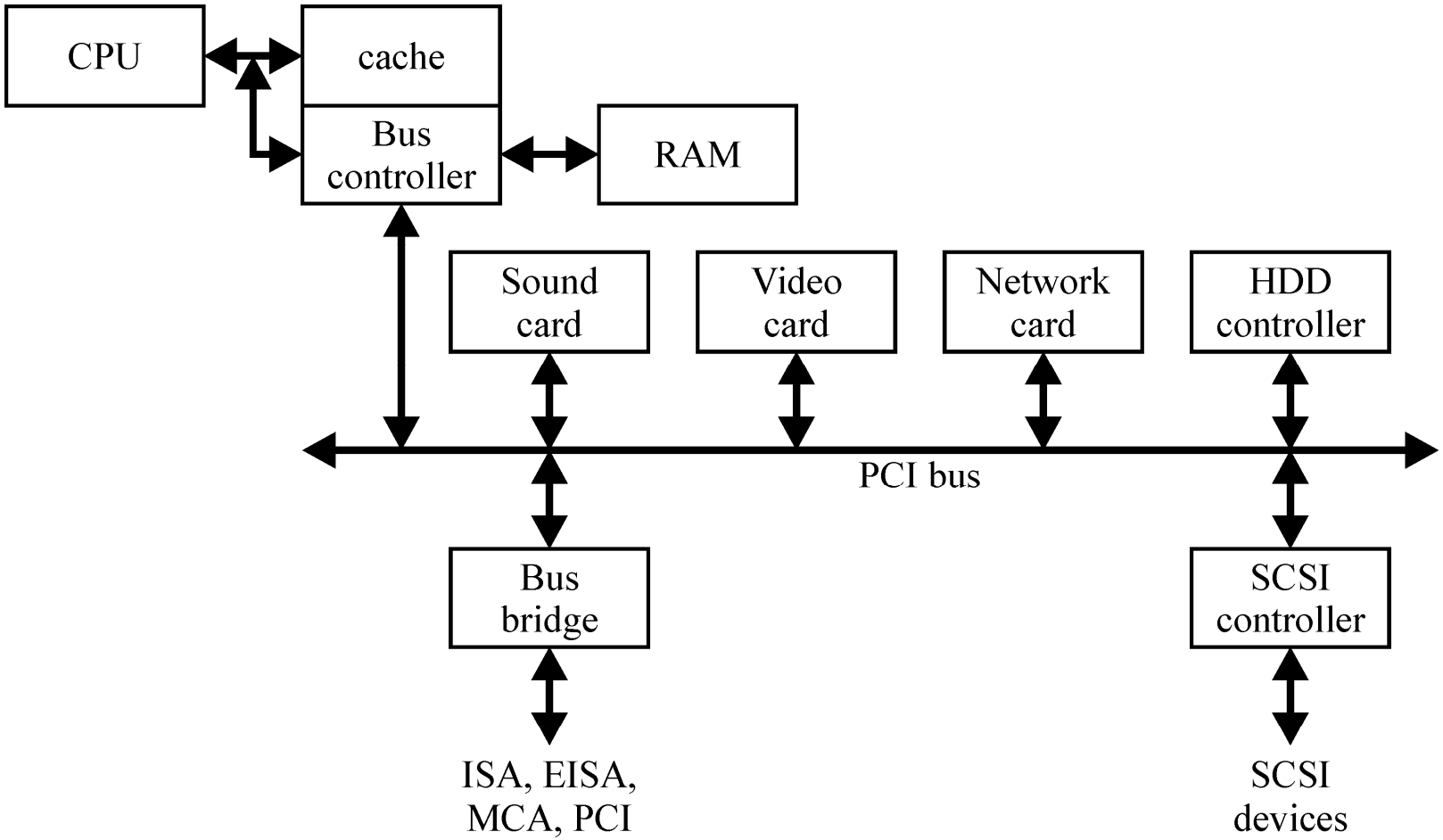
- VLB
  - Advantages
    - Simplicity, low cost
    - High throughput (up to 200 MBps @ 50 MHz)
  - Disadvantages
    - No auto-configuration
      - Hardware-only configuration
    - 486-signals dependence
      - PCI-VLB bridges not very popular
    - Limited number of slots
    - Low reliability & scalability
      - Problems with high frequencies

From EISA to PCI

- PCI
  - Peripherial Component Interconnect
  - Very complex systems:
    - 256 buses × 32 devices × 8 functions
  - 33/66 MHz, 1 clk/cycle, 32/64 bits
  - Devices:
    - Initiator – master
    - Target – slave
  - Software configuration
    - 256 B configuration memory
    - Socket numbers
    - Vendors & devices ID’s

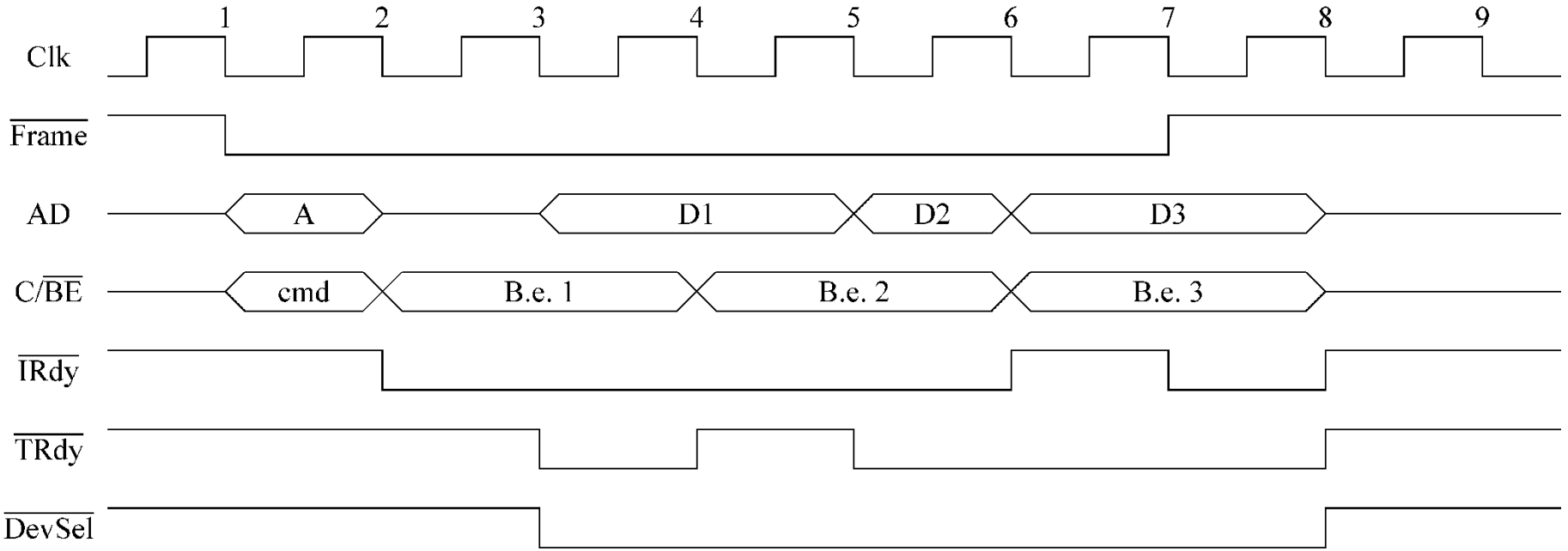
From EISA to PCI

- PCI



From EISA to PCI

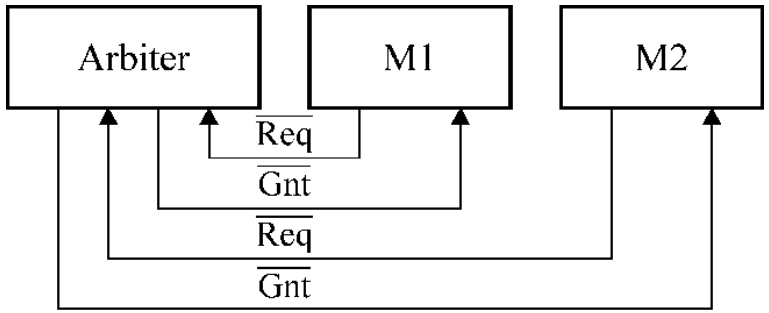
- PCI



- Frame=0 → read addr and cmd
- Addr OK → devsel
- Data exchange
  - 1 clk/exchange
  - Irdy, Trdy=1 → wait
- Frame=1 → end
- Devsel=1 → bus free

From EISA to PCI

- PCI – arbitration
  - Performed when ≥1 master active
  - Usually in Host/PCI, but not standarized



  - Arbitration algorithm not defined
    - E.g., round-robin, priority, etc.
  - Possible during data transfer
    - „hidden arbitration”

From EISA to PCI

- PCI signals (basic – 32b and auxilliary – 64b)

Master	Signal	Slave
↔	$AD_{0..31}$	↔
→	$C/\overline{BE}_{0..3}$	→
↔	PAR	↔
↔	$\overline{Frame}$	→
←	$\overline{TRdy}$	←
→	$\overline{IRdy}$	→
←	$\overline{Stop}$	←
←	$\overline{DevSel}$	←
←	IDSel	→
→	$\overline{Req}$	×
←	$\overline{Gnt}$	×
←	Clk	→
←	RST	→
←	$\overline{PErr}$	←
→	$\overline{SErr}$	←

Master	Signal	Slave
↔	$AD_{32..63}$	↔
→	$C/\overline{BE}_{4..7}$	→
↔	PAR64	↔
↔	$\overline{Req64}$	→
←	$\overline{Ack64}$	←
↔	$\overline{Lock}$	→
↔	$\overline{ClkRun}$	↔
×	$\overline{SBO}$	→
×	Sdone	→
←	TDI	→
→	TDO	←
←	TCK	→
←	TMS	→
←	$\overline{TRst}$	→
→	$\overline{IntA}$.. $\overline{IntD}$	←

From EISA to PCI

- PCI signals detailed
  - AD – address/data
    - No bus sizing – only 32- or 64-b transfers
  - $\mathrm{C}/\overline{\mathrm{BE}}$ – command/byte enable
    - Address phase – command
    - Data phase – byte enable
  - PAR – parity check for AD and $\mathrm{C}/\overline{\mathrm{BE}}$
    - Parity error – action depends on settings
      - From ignore to NMI

From EISA to PCI

- PCI signals detailed

Code	Command	Comment
0	IntAck	Read of interrupt controller; only for PC compat.
1	Special Cycle	Broadcast to many targets
2	I/O Read	
3	I/O Write	
4	®	
5	®	
6	Mem Read	
7	Mem Write	
8	®	
9	®	
A	Conf Read	
B	Conf Write	
C	Mem Read Multiple	Long read beyond cache line
D	Dual Address cycle	64-bit address from initiator
E	Mem Read Line	Read to the end of cache line
F	Mem Write & Invalidate	

From EISA to PCI

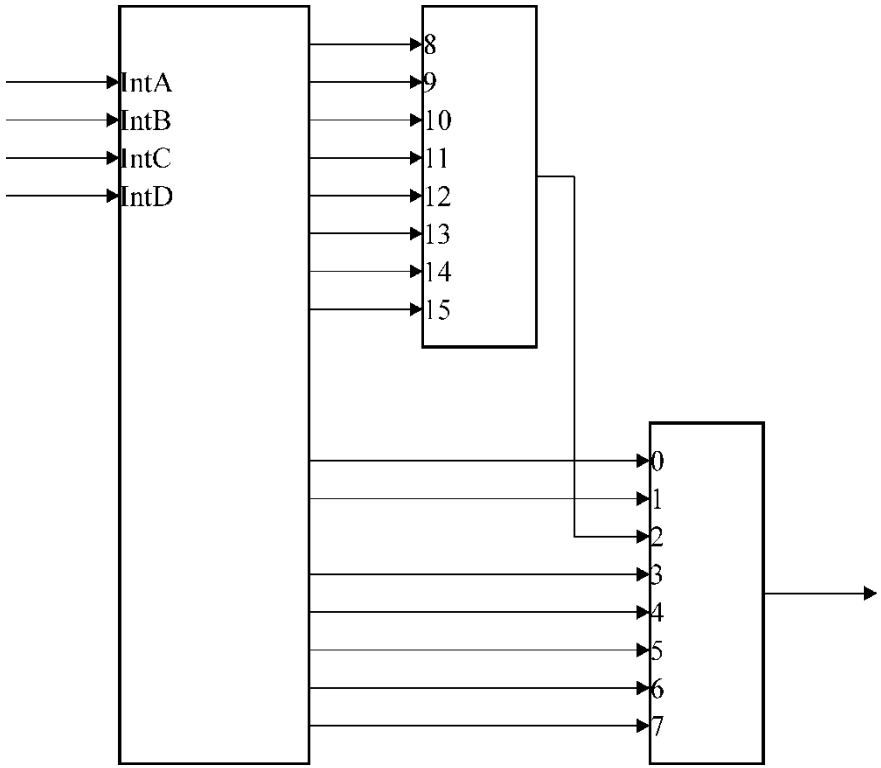
- PCI signals detailed
  - $\overline{\text{Frame}}$ – activated by granted initiator
    - Inactive during last transfer
  - $\overline{\text{TRdy}}$ – target ready; inactive → wait state
  - $\overline{\text{IRdy}}$ – initiator ready
  - $\overline{\text{Stop}}$ – immediate transmission end by target
    - Reaction depends on $\overline{\text{TRdy}}$ and $\overline{\text{DevSel}}$
  - $\overline{\text{DevSel}}$ – address recognized by target
    - No later than 2 clk after address sent
  - IDSel – used for Configuration rd/wr

From EISA to PCI

- PCI signals detailed
  - $\overline{\text{Req}}$, $\overline{\text{Gnt}}$ – for initiator arbitration
  - $\overline{\text{PErr}}$ – parity error
  - $\overline{\text{SErr}}$ – system error other than parity; → NMI
  - PAR64 – parity bits for 64-b extension
  - $\overline{\text{Req64}}$ – attempt to perform 64-b transfer
  - $\overline{\text{Ack64}}$ – 64-b transfer allowed; timing like $\overline{\text{DevSel}}$
  - $\overline{\text{Lock}}$ – target locked
  - $\overline{\text{ClkRun}}$ – lower fclk; for power saving
  - $\overline{\text{SBO}}$ – don’t write to not update WriteBack cache

From EISA to PCI

- PCI signals detailed
  - $\overline{\text{IntA}}$.. $\overline{\text{IntD}}$ – interrupt lines
    - Can be shared among PCI devices
    - In PC translated to IRQ lines



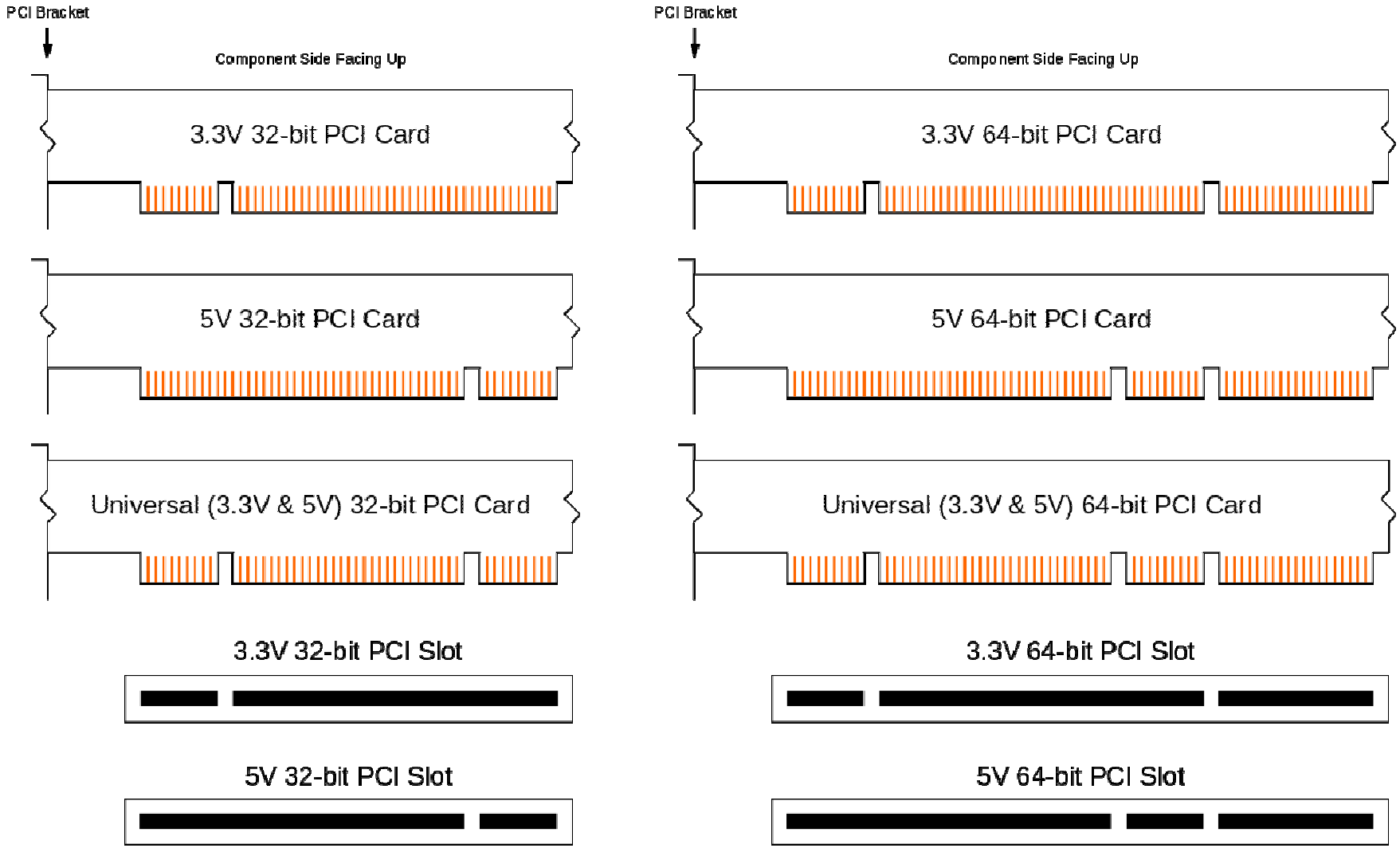
From EISA to PCI

- PCI signals detailed
  - In PC, various INT assignment to various slots
  - PCI devices can only use A6 pin
  - INT conflits avoidance → slot change
  - Example assignment (from some real MoBo)

Slot	A6	A7	B7	B8
1	**A**	C	B	D
2	**B**	D	A	C
3	**C**	A	B	D
4	**A**	B	D	C
5	**D**	B	C	A

From EISA to PCI

- PCI – slot & card types



From EISA to PCI

- PCI example photos