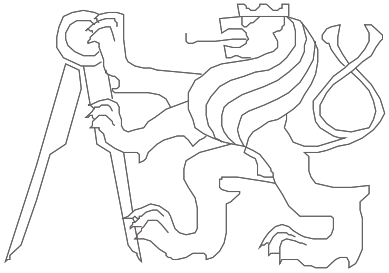
Computer Architectures

I/O Subsystem Part 2

Pavel Píša, Richard Šusta,
Michal Štepanovský, Miroslav Šnorek



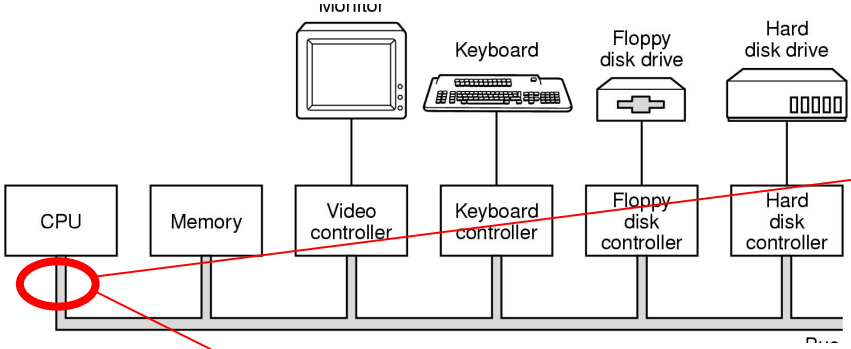
Czech Technical University in Prague, Faculty of Electrical Engineering

Lecture outline

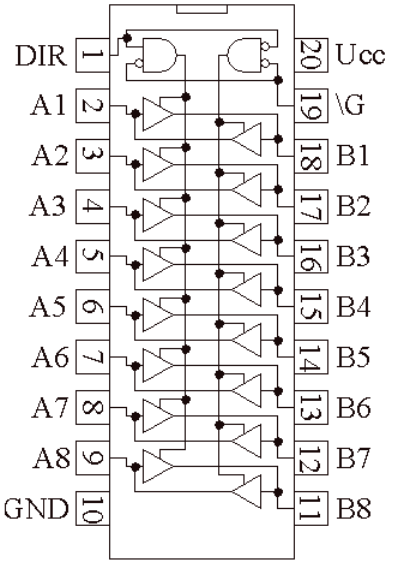
- Computer Buses Hardware Components
- Design of addon PCI cards
- Design of PCIe with use of FPGA chip

Computer Buses Hardware Components

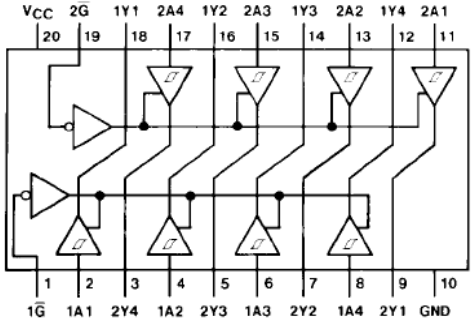
Bus-line Electronics - Buffers



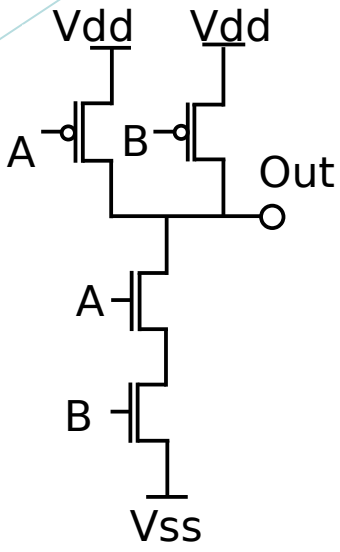
bi-directional

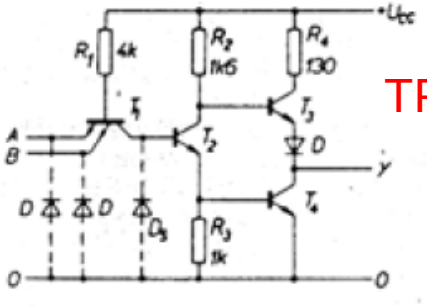


uni-directional



IO Output: TP and OC? Practical Examples:

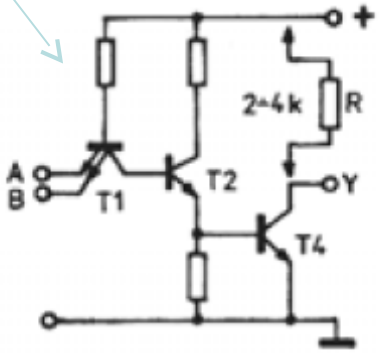




TP – Totem Pole

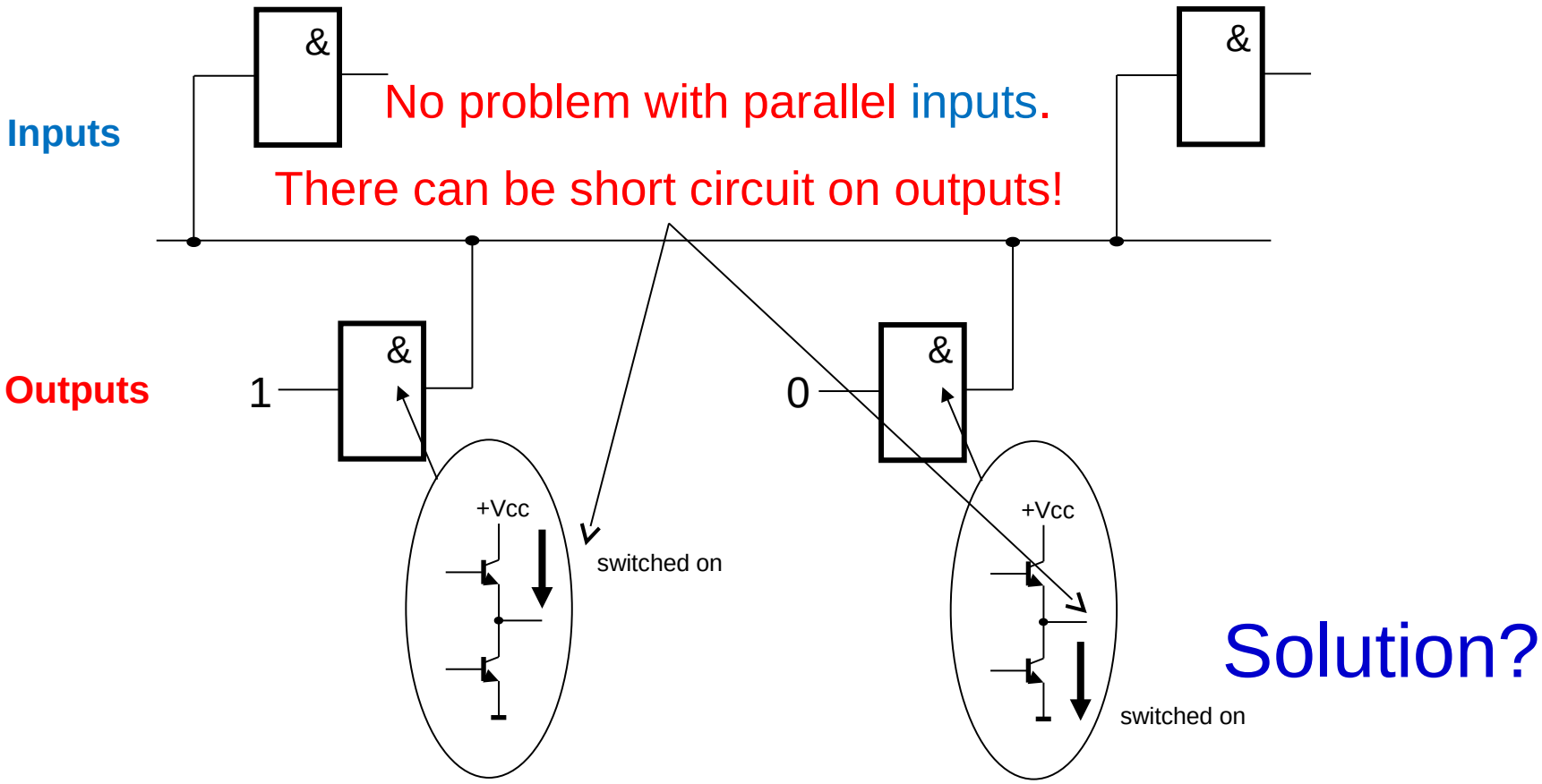
Internal structure of two inputs
NAND chip from TTL 7400 family

OC – Open Collector

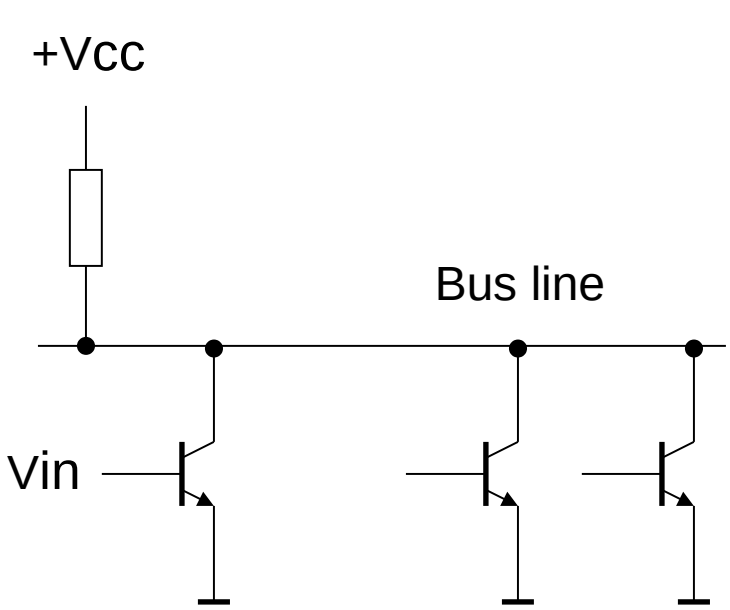


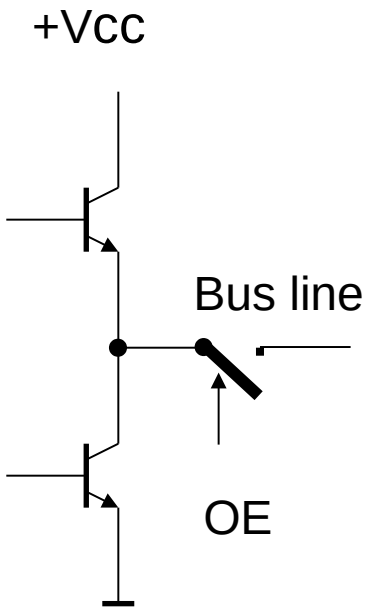
Internal structure NAND circuit equipped by open collector output.
Pull up resistor is connected externally.

Study of Conflict: Bi-directional Transfer on a Single Bus Line



One Solution: Output Circuit as OC x Tri-state



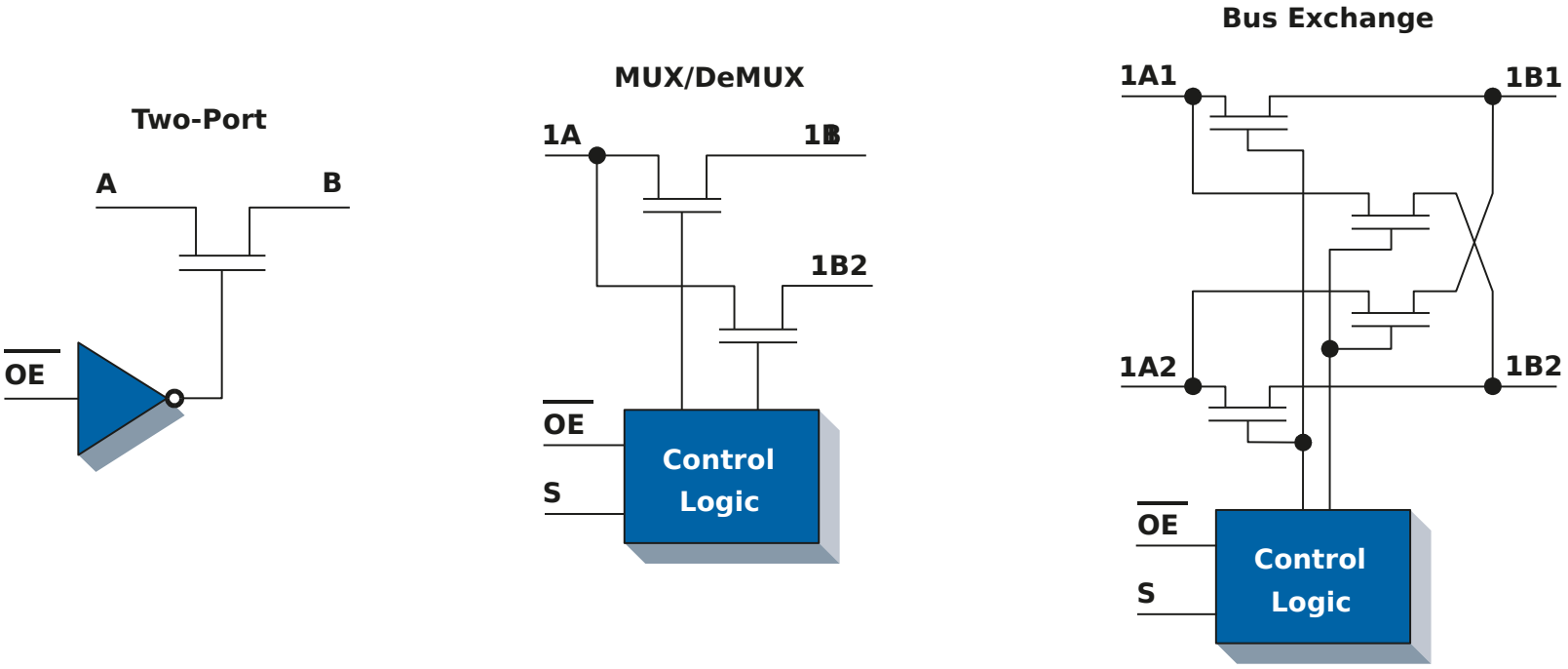


OC – Open collector

3S – Tri-state output

OE = Output Enable

Bus Switches and Multiplexers



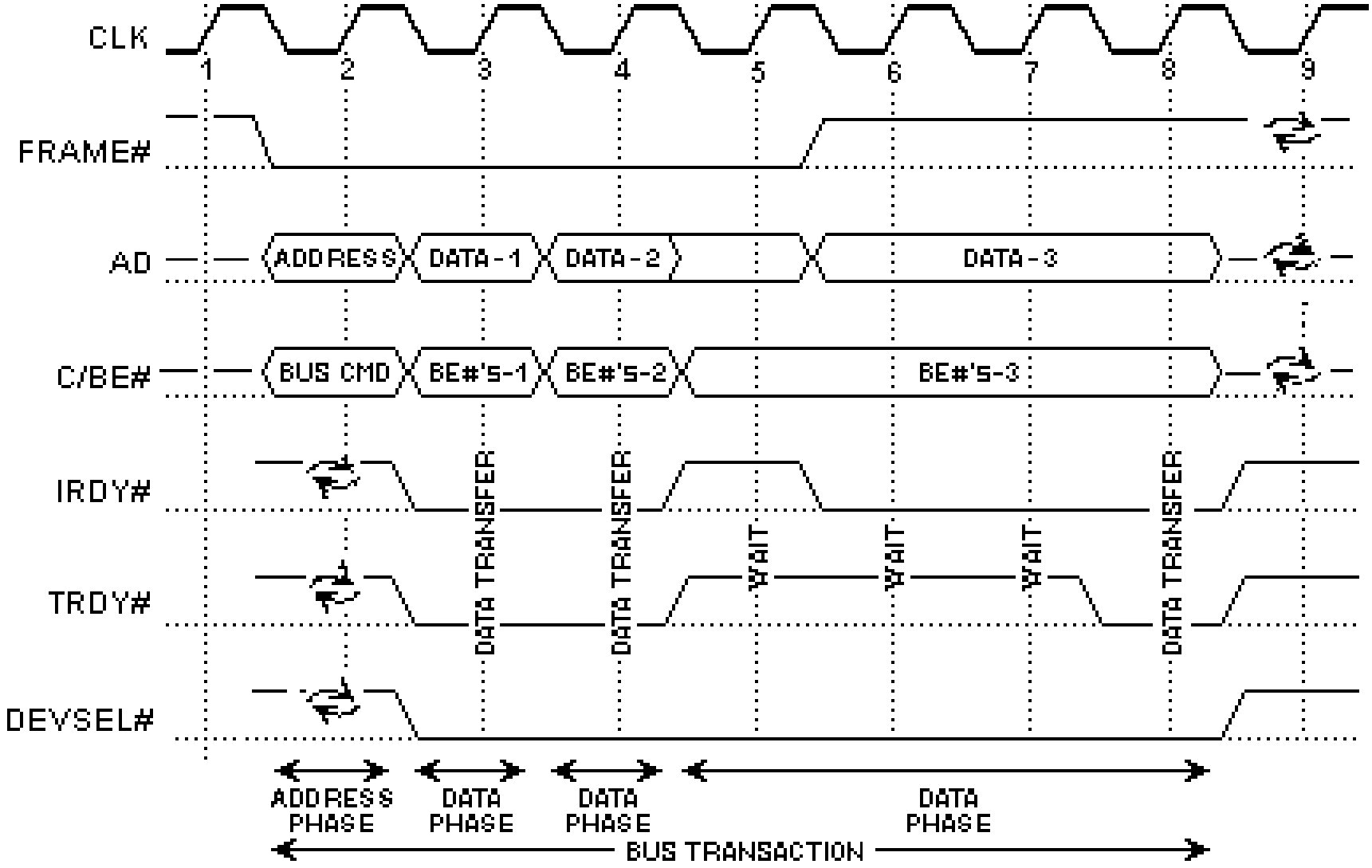
Source: Texas Instruments Digital Bus Switch Selection Guide
http://www.ti.com/signalswitches

PCI Device Design Example

PCI Device Card Interface Design Example

- The card requires three address ranges
  - Two memory mapped, 4kiB each
  - One I/O space mapped, size 16B
- Design steps
  - Analyze bus cycles sequences that should be recognized
  - Remember electronics (CPU, bus) building blocks
  - Define interface structure
  - Implement address decoder
  - Implement control logic
  - Implement data path
  - And then think what the card should be used for
    - **NO**, regular design starts from function and its needs

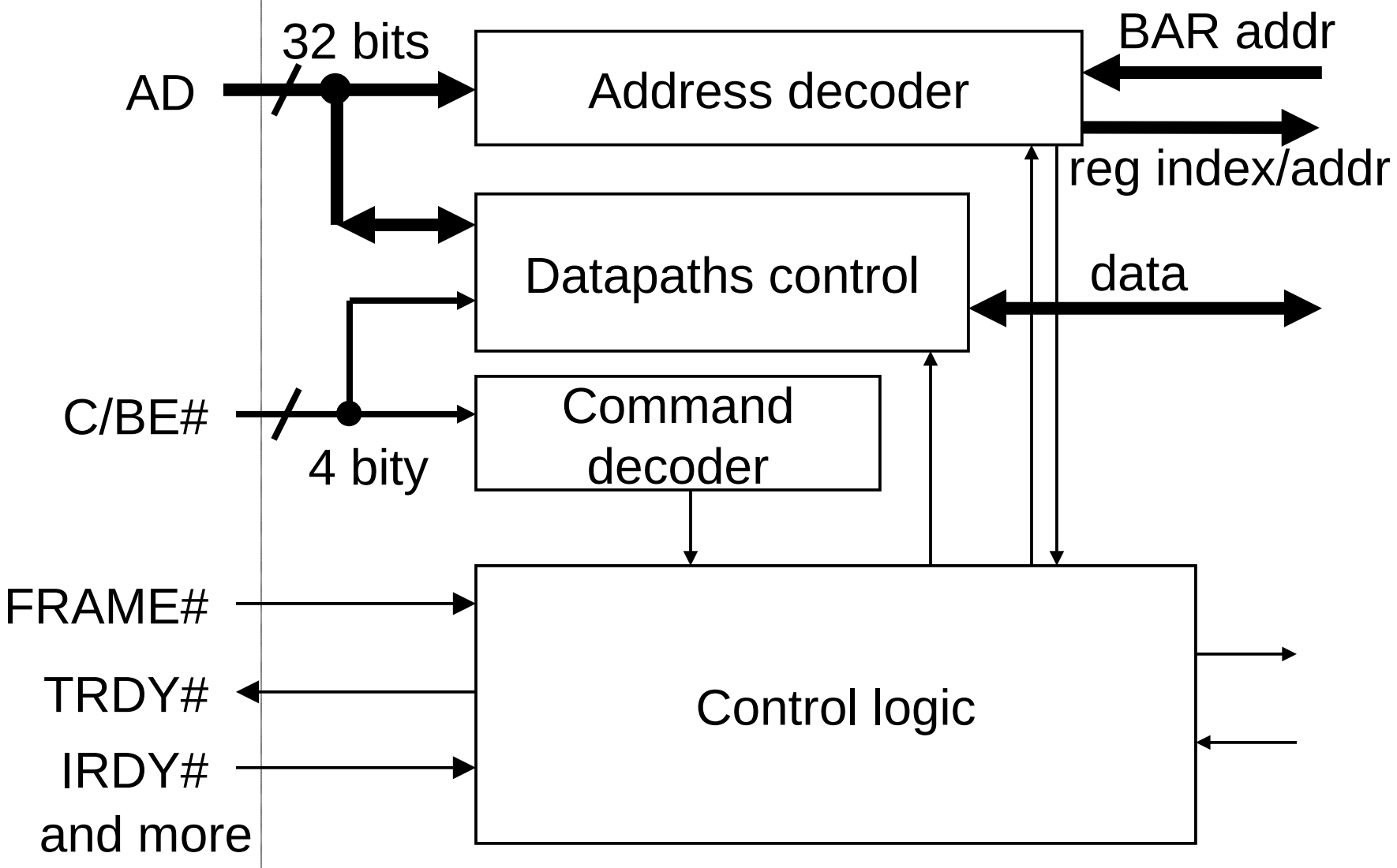
Bus Cycle (Transaction)



Interface Building Blocks

- Data bus and block to control datapath (enable, direction)
- Address signals, address decoder
- Command decoder
- Control logic
- (Interrupt signal generator – INT#)
  - Only when card uses interrupt – but highly desirable
- Logic to request bus control (initiator/master) role from the bus arbiter
  - Only if card is/can act as master (bus master DMA etc.)

PCI Device/Card Interface



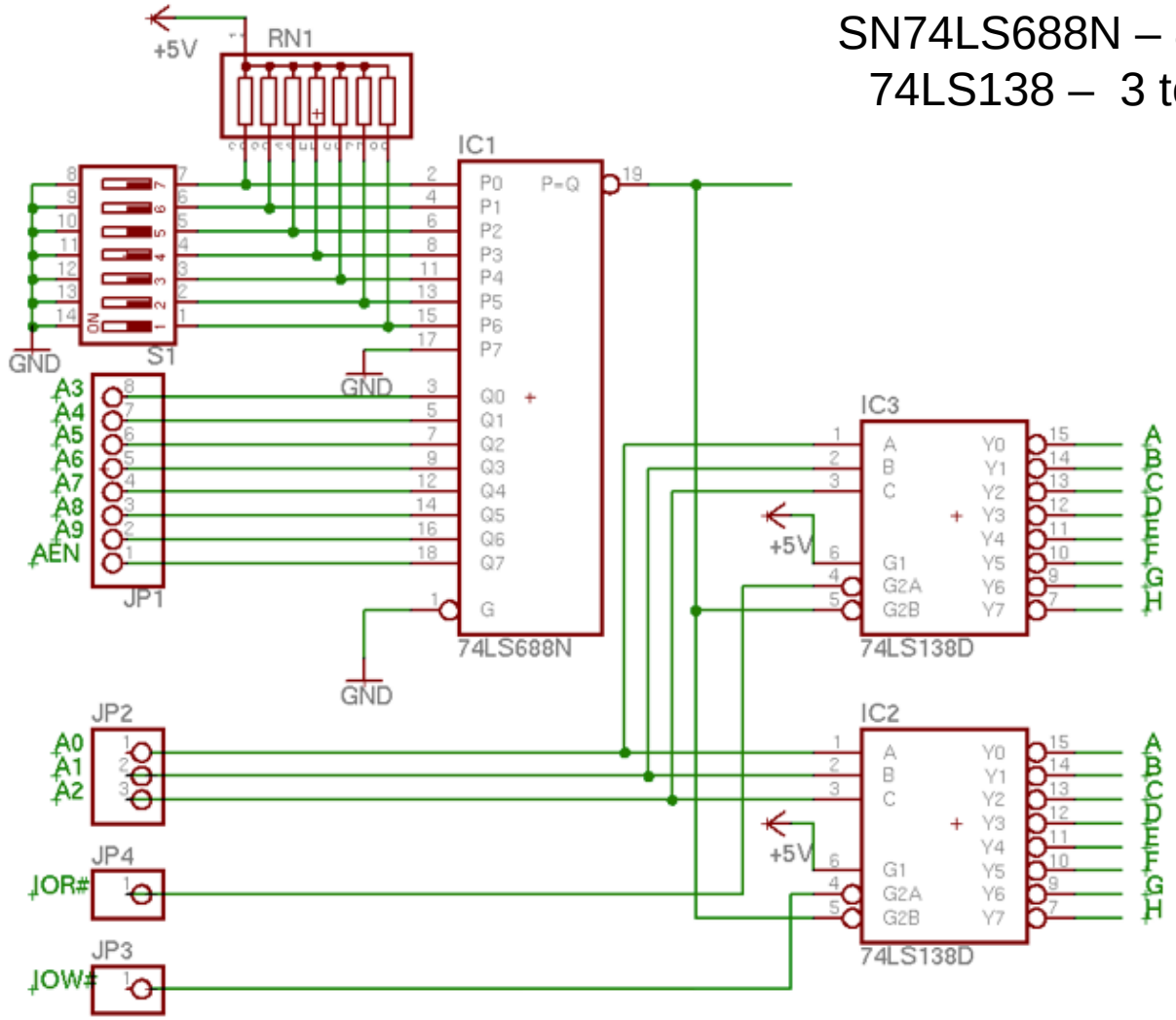
Address Decoder

- The basic block is address comparator
  - It compares significant bits (according to the region size) of the address sent on bus with the address stored in one of the base address registers (one comparator for each BAR)
- Address is present on AD signals only in the first phase of the bus cycle ⇒ the address has to be latched (stored) in card's **address register**
- If block transfers are supported then address register has to provide autoincrement function – it is realized by up counter with parallel preset (LOAD)

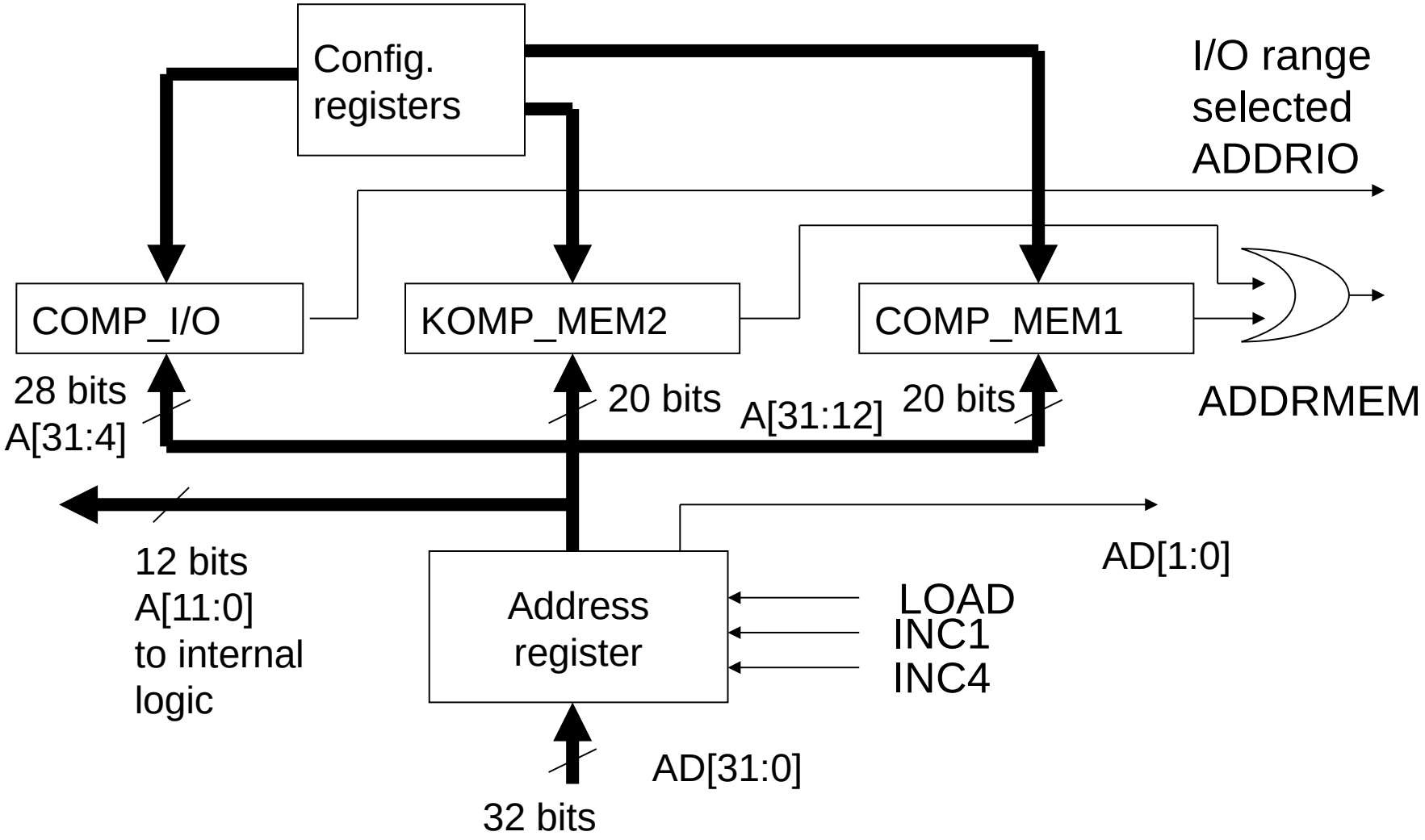
- Consider relocable address decoder. Consider reduced comparator – mirroring.

Example of DIP Switch Programmed Address Decoder

SN74LS688N – 8-bit comparator
74LS138 – 3 to 1of8 decoder



Configurable Address Decoder (i.e. BAR Based)



Configurable Address Decoder Signals

- ADDRIO
  - Address matches I/O range
- ADDRMEM
  - Address matches one of two memory mapped ranges
- Address register is a synchronous counter with parallel synchronous preset
  - LOAD – synchronous address load on the next rising edge of the clocks
  - INC1 – increment stored value/address by 1
  - INT4 – increment stored value/address by 4
- AD[1:0] – informs internal logic about burst mode type

Other Required Blocks

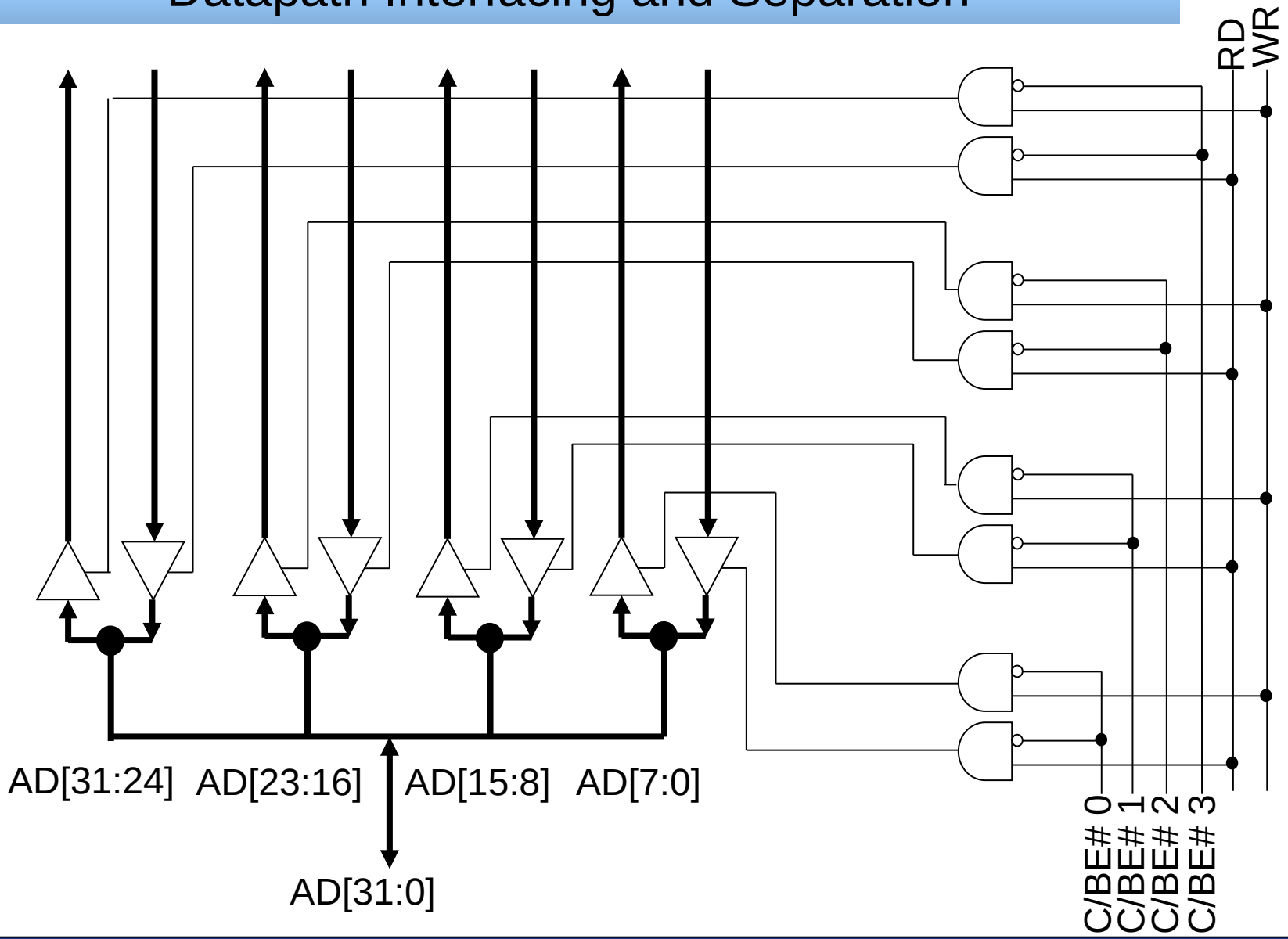
- Configuration space
  - Register array (size 256B). All cells can be read, writes to some registers/bits are ignored (i.e. BAR's low order bits)
- Parity check (generates PERR# signal)
- Error control, i.e. check for the address register/counter overflow during continuous/burst transfer (generates SERR# signal)

- Consider wait cycles logic TRDY# assignment etc.
- Address for memory reads/writes has to be 4 bytes aligned (partial bus use/data validity can be controlled by C/BE[3:0] signals) ⇒ address increment is 4 for memory accesses
- but for I/O byte wide accesses INC1 required as well

Datapaths and their Control

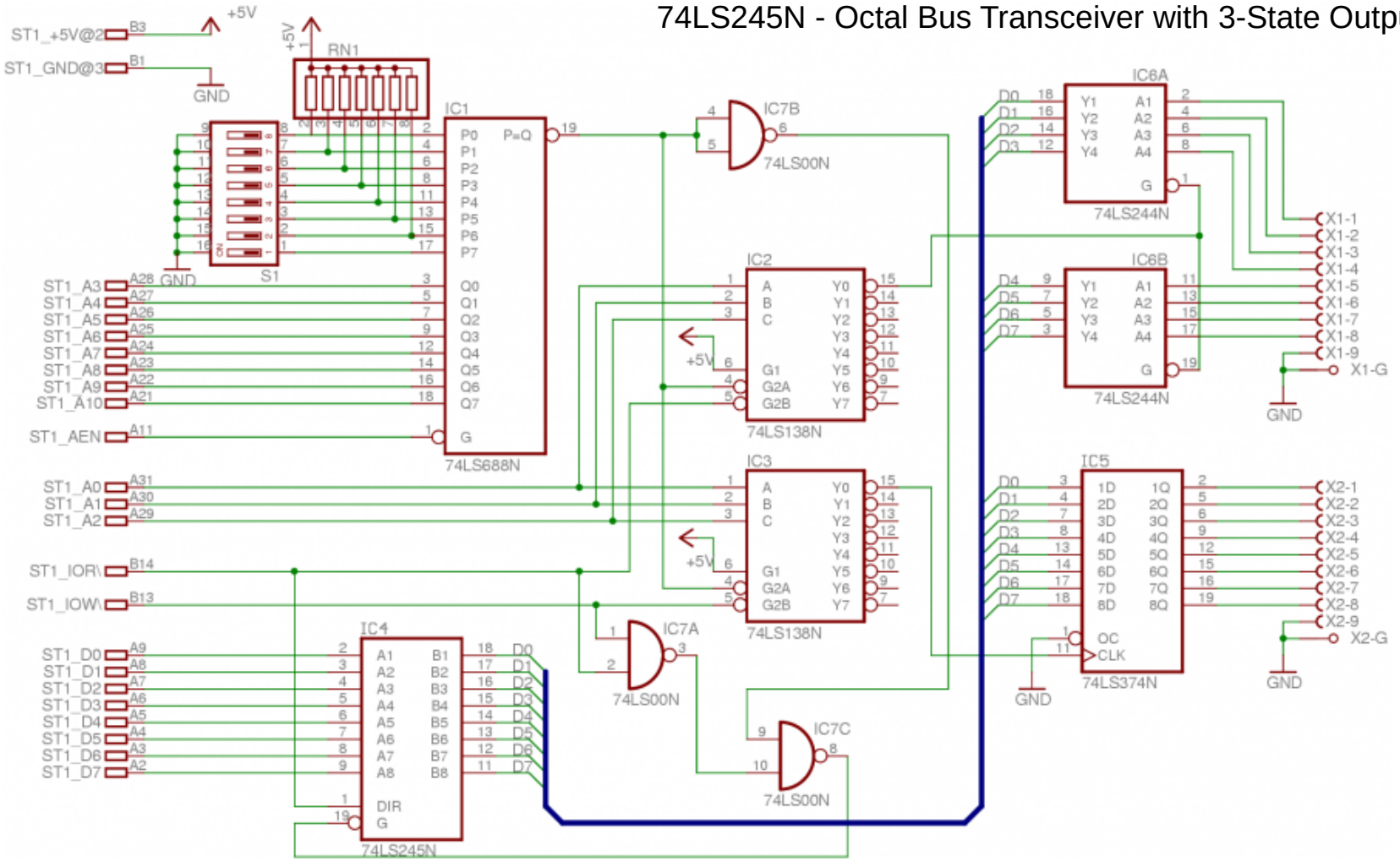
- Data bus is bidirectional ⇒ the interface requires (bidirectional) transceiver with three-state outputs
- 8/16/32-bit data transfers the direction and the high impedance state control is based on command type (read/write) and on mask selecting valid octets of bits on the bus C/BE#

Datapath Interfacing and Separation



Pre-PCI Style I/O Port Realization Example

74LS244N - Edge Triggered Flip-Flop
74LS245N - Octal Bus Transceiver with 3-State Outputs

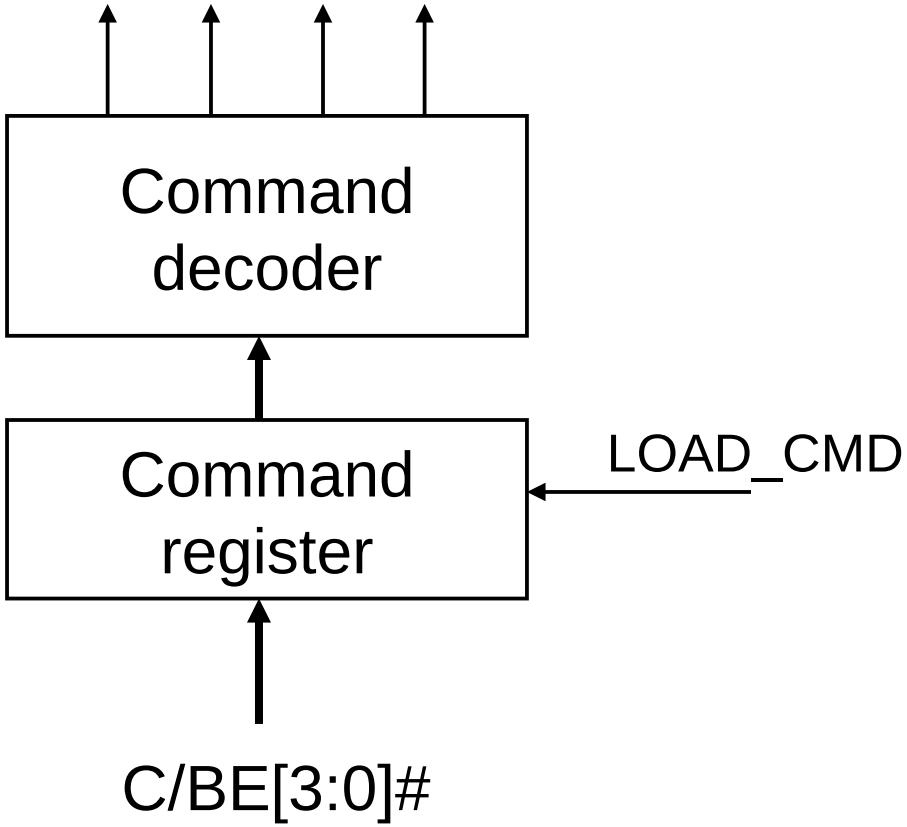


PCI Command Decoder

- Command is latched into command register
- Command decoder is then realized as combinatorial logic
- Outputs are control signals which specify:
  - data transaction direction
  - transaction type

    I/O operation, memory space operation, configuration access/ cycle, interrupt request/acknowledge
- Use of combinatorial decoder simplifies control logic design
  - Compare with opcodes decode and arithmetic operation specifications described in lecture “Processor”

Command Decoder



Command Decode – C/BE[3..0]# Meaning

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

Command Decoder Output Signals

- RD – read operation
- WR – write operation
- IO – operation targets I/O space
- MEM – operation targets memory space
- CONF – read/write from/to configuration space
- INT – command Interrupt Acknowledge

Interface Control

- Detect start and end of a cycle
- Generates DEVSEL# if address recognized by device/card, controls address register, command register and decoder, monitors IRDY# signal (wait cycle inserted by initiator – master) to inform card logic that given transaction phase is prolonged
- Input signals are
  - FRAME# – controls transaction start and transaction last transfer phase
  - IRDY# – initiator ready/wait request
  - ADDRIO, ADDRMEM, MEM, IO

Interface Control Realization

- Sequential circuit can be described/realized as finite state machine
- PCI clock signal is used as clock input for designed FSM, synchronous bus and control design
- **Quiz**:
- Should be design based on Moore FSM or Meally FSM or it is not important?

Quiz Answer

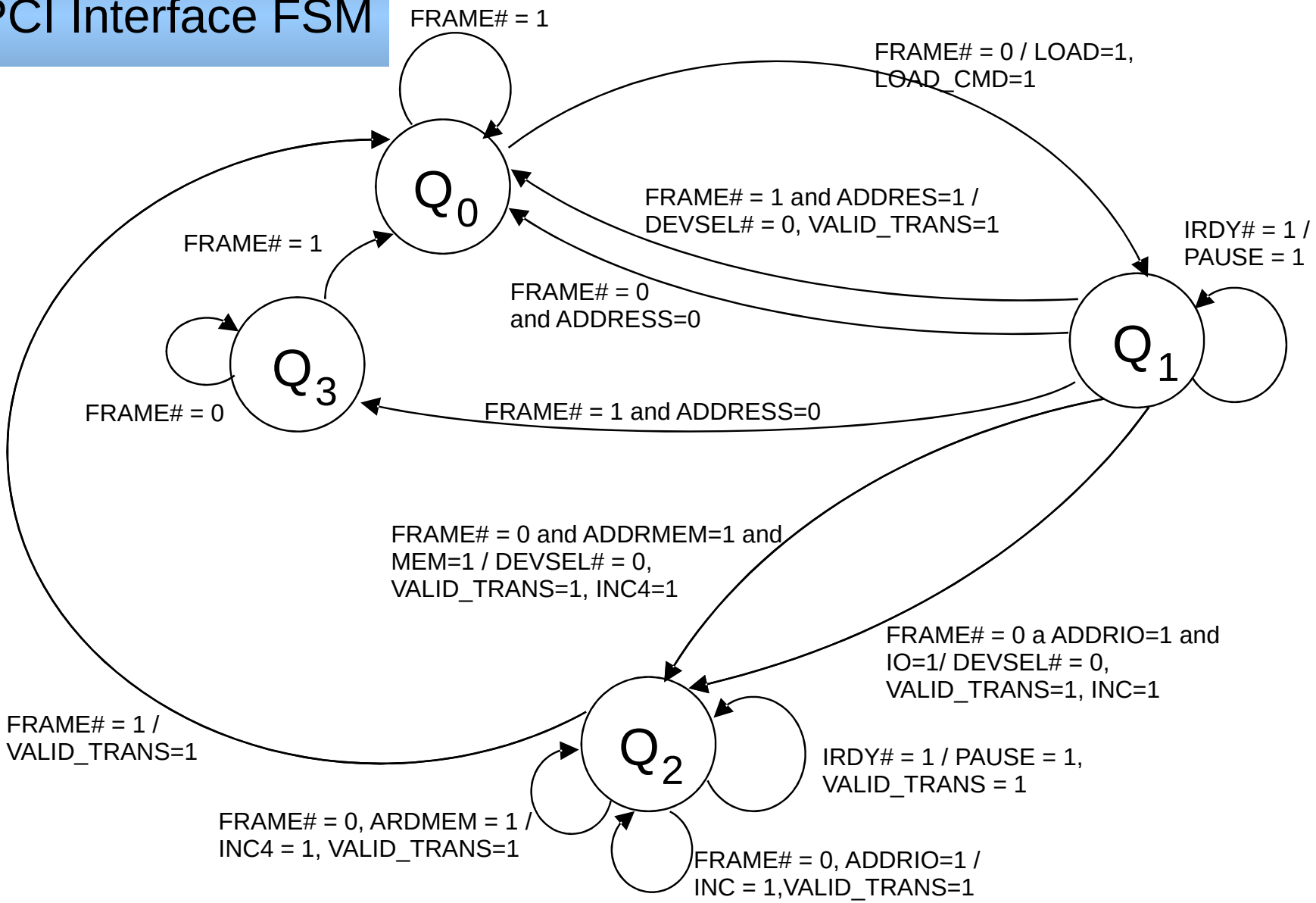
- The control logic design has to be Meally FSM, because control signals have to be prepared even before first rising clock of the PCI clock to select the right function of address latch register and other components
- Design choice
  - We consider all control signals in positive logic for simplicity

Interface Controller/FSM Signals

- Output signals
  - LOAD
  - LOAD_CMD
  - DEVSEL#
  - VALID_TRANS
  - INC1, INC4, PAUSE (wait/phase hold for internal logic)

- Design choices
  - only active output signals are shown in the transition graph
  - ADDRESS = ((ADDRIO==1 and IO==1) or (ADDRMEM==1 and MEM==1))

PCI Interface FSM



Data Path Direction and HiZ Control

- The data path transceiver direction and high impedance state control can be derived from signals
  - VALID_TRANS and WR
  - VALID_TRANS and RD

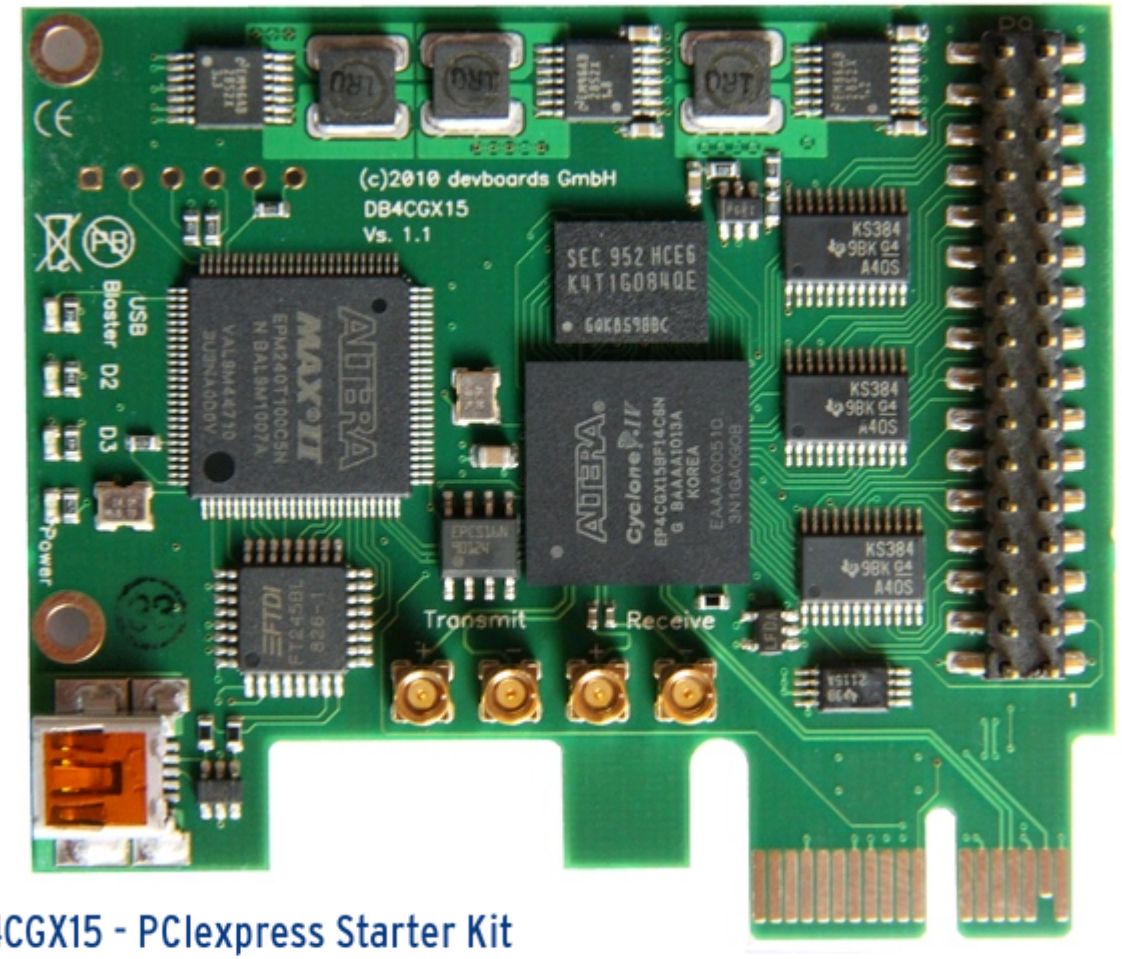
  generated by command decoder

PCI Express Design based on FPGA Chip

The PCIe board DB4CGX15 example

- 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