

# The Digital Logic Level

Wolfgang Schreiner  
Research Institute for Symbolic Computation (RISC-Linz)  
Johannes Kepler University

Wolfgang.Schreiner@risc.uni-linz.ac.at  
<http://www.risc.uni-linz.ac.at/people/schreine>

## The Digital Logic Level

The computer's real hardware.

- Basic elements: gates.
- Basic logic: Boolean algebra.
- Combinatorial Circuits.
- Arithmetic Circuits.
- Memory.
- CPUs and buses.

Boundary between computer science and electrical engineering.

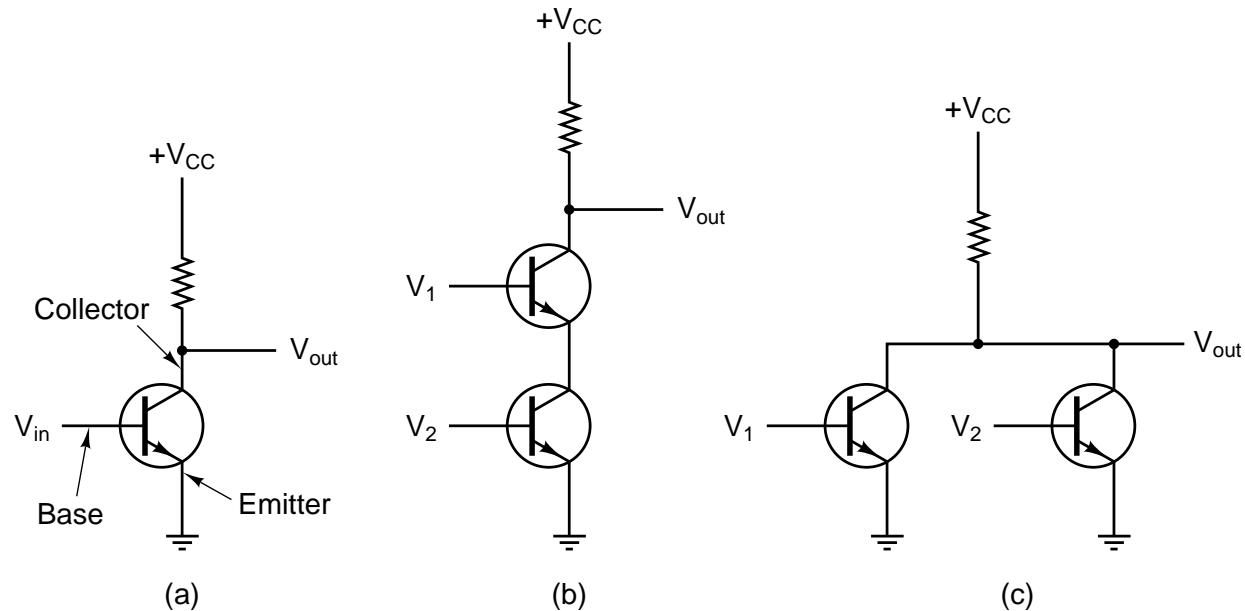
## Gates

A gate is a device that computes a function on a two-valued signal.

- **Fundament:** **transistor** can operate as a binary switch.
  - Three connections to the outside: **collector**, **base**, **emitter**.
  - Input voltage  $V_{in} <$  critical value: transistor becomes infinite resistance.
    - \* Output voltage  $V_{out}$  becomes externally regulated voltage  $V_{cc}$  (5V).
  - Input voltage  $V_{in} >$  critical value: transistor becomes a wire.
    - \* Output voltage  $V_{out}$  is pulled to ground (0V).
- **Interpret voltages as logical values.**
  - “High” voltage ( $V_{cc}$ ) is a logical 1.
  - “Low” voltage (ground) is a logical 0.

**Transistor acts like a logical inverter (NOT).**

## Basic Gates: Construction



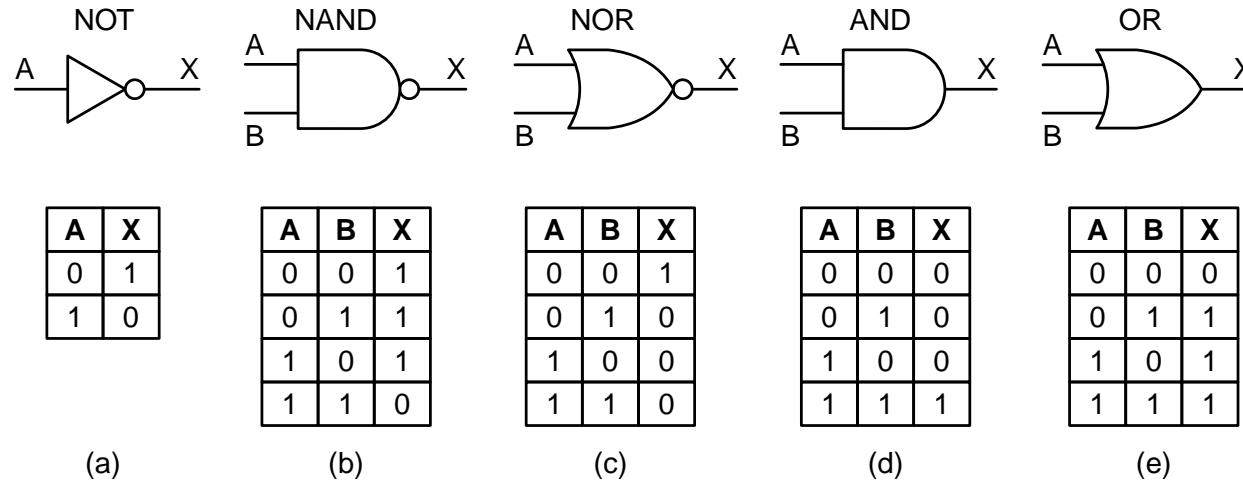
NOT

NAND

NOR

NAND and NOR gates can be constructed by wiring two transistors in series respectively in parallel.

## Basic Gates: Logic



Most computers are based on NAND and NOR gates.

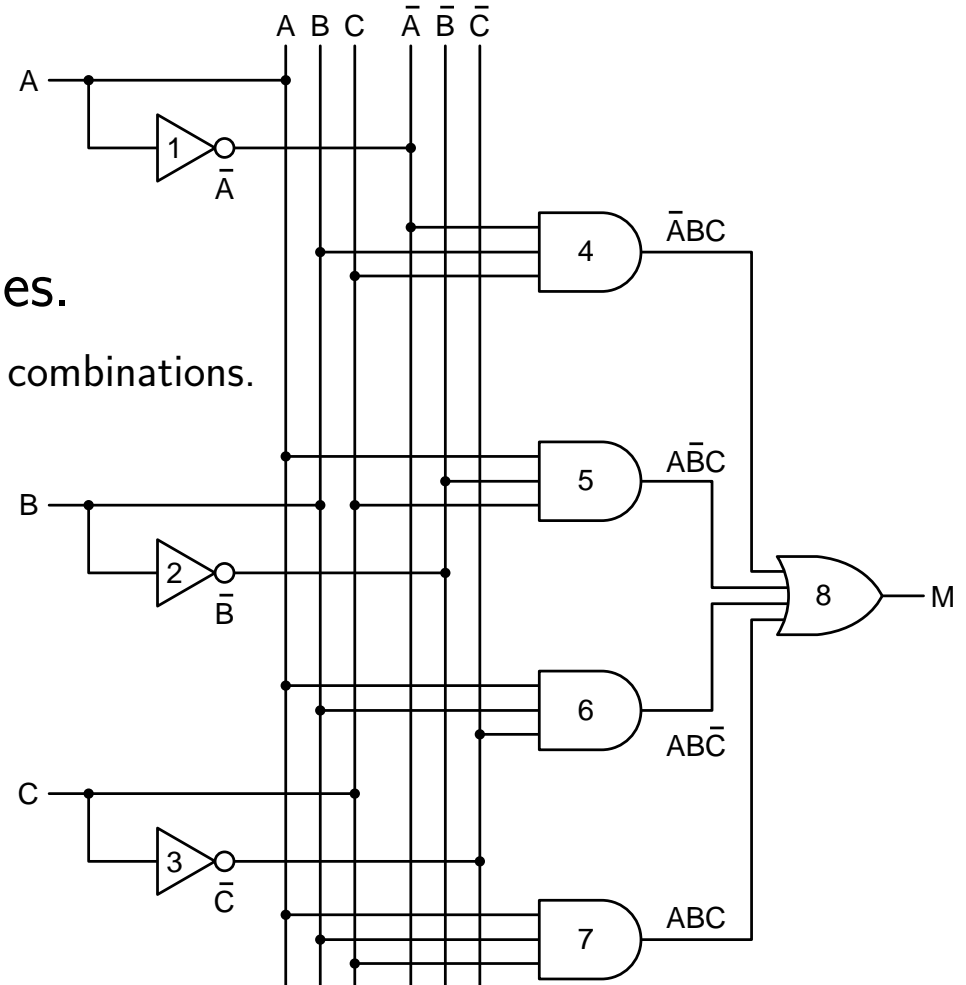
# Boolean Algebra

Algebra of boolean functions.

- Inputs and results are logical values.
  - Boolean function of  $n$  variables has  $2^n$  input combinations.
  - Representation by **truth table** with  $2^n$  rows.
  - $2^{2^n}$  Boolean functions with  $n$  variables exist.

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(a)



(b)

## Other Notation

Truth tables are too clumsy to handle.

- Suffices to specify which combinations of inputs gives output 1.
  - Let  $\bar{A}$  denote negation,  $AB$  denote conjunction,  $A + B$  denote disjunction.
  - $M = \bar{A}BC + A\bar{B}C + ABC\bar{C} + ABC$ .
  - A function of  $n$  variables can be described by a sum of at most  $2^n$  product terms of  $n$  variables.

Linear representation of Boolean functions.

## Implementation of Boolean Functions

Construct circuit for a given Boolean function.

- Systematic process:

1. Write down the truth table for the function.
2. Provide inverters to generate the complement of each input.
3. Draw an AND gate for each term with a 1 in the result column.
4. Wire the AND gates to the appropriate inputs.
5. Feed the output of all AND gates into an OR gate.

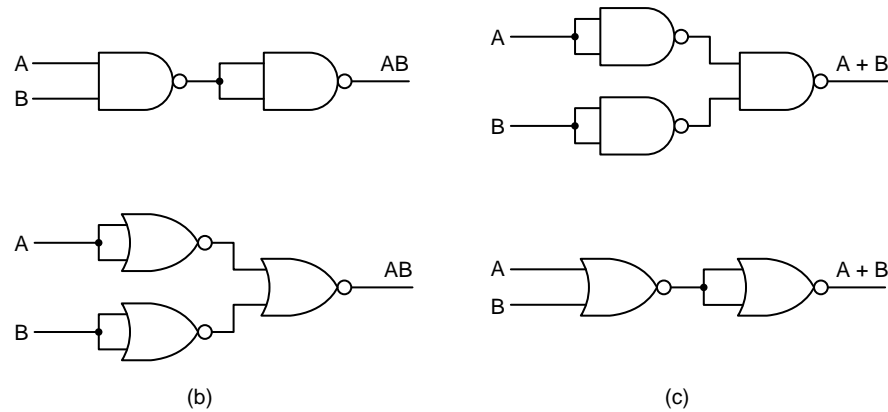
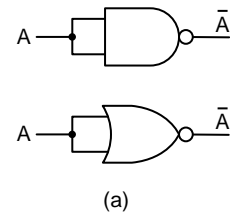
- Further transformations possible:

1. Replace multi-input gates by two-input gates ( $A + B + C + D = (A + B) + (C + D)$ ).
2. Replace NOT, AND, OR gates by NAND gates (or by NOR gates).

Circuit is not necessarily the simplest one.

## Construction of NOT, AND, OR

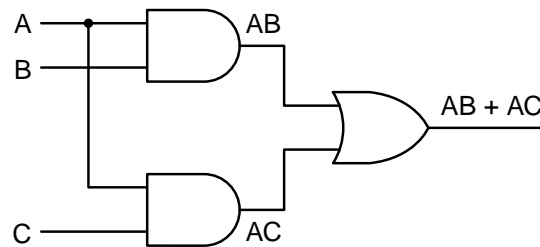
Any Boolean function can be constructed from NAND or NOR only.



The gate set {NAND, NOR} is complete.

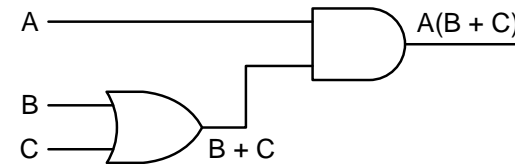
## Circuit Equivalence

Try to reduce the number of gates in a circuit.



A	B	C	AB	AC	AB + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

(a)



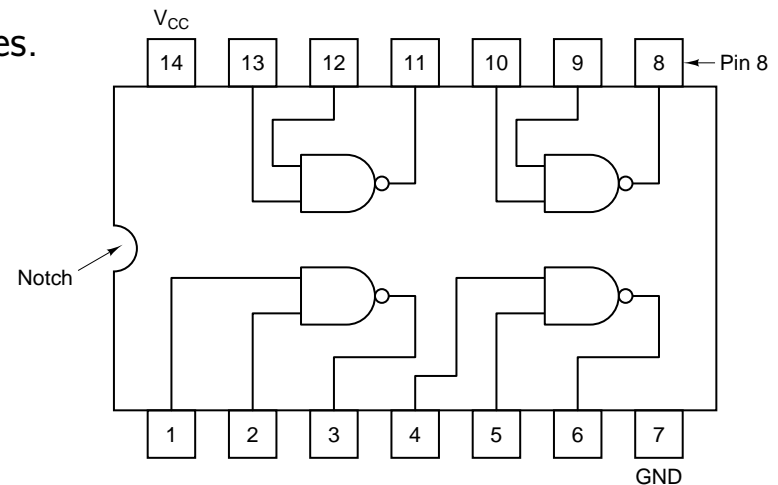
A	B	C	A	B + C	A(B + C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

(b)

## Integrated Circuits

Gates are manufactured in units called Integrated Circuits (ICs).

- Square piece of silicon ( $5\text{mm} \times 5\text{mm}$ ).
  - Gates are deposited on these “chips”.
  - Multiple chips are mounted in packages of e.g.  $15\text{ mm} \times 50\text{mm}$ .
  - Two parallel rows of pins are placed on long edges.
- Various integration scales.
  - SSI (Small Scale Integrated): 1–10.
  - MSI (Medium Scale Integrated): 10–100.
  - LSI (Large Scale Integrated): 100–100.000.
  - VLSI (Very Large Scale Integrated):  $>100.100$ .



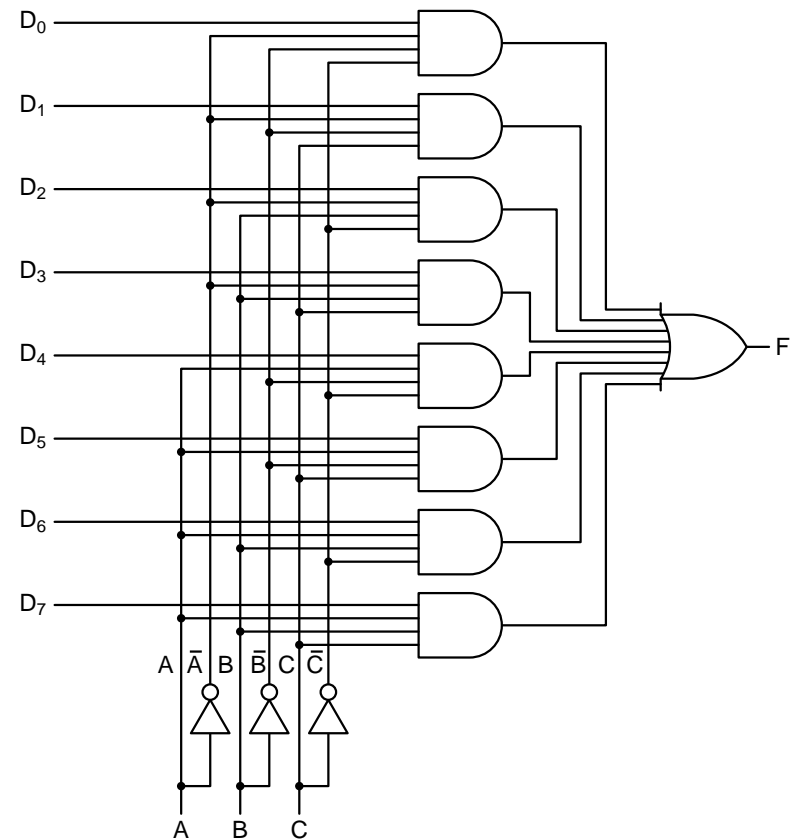
Today: up to 10 million transistors per chip.

# Combinatorial Circuits

## Multiplexers

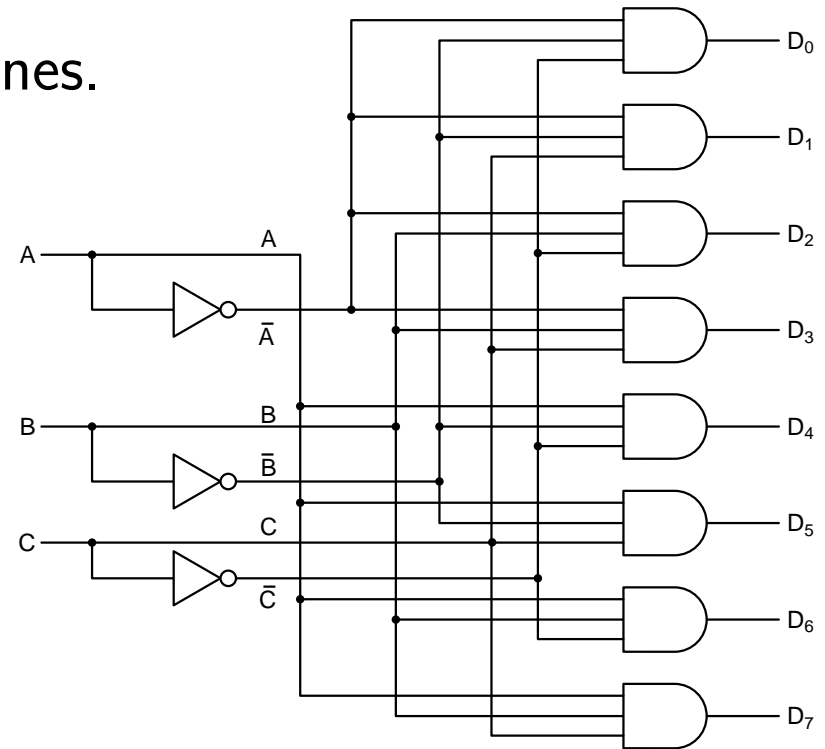
- $2^n$  data inputs, one data outputs, 1 control input.
  - Control input selects one of the data inputs.
  - Selected input is routed to the output.
- Inverse is **demultiplexer**.
  - 1 data inputs,  $2^n$  outputs, 1 control input.
  - Input is routed to the selected output.

Fundamental routing operations.



## Decoders

- $n$ -bit number as input,  $2^n$  output lines.
  - Input selects output line which is set to 1.
- Example application:
  - Memory of eight 1MB chips.
  - 0-1MB, 1-2MB, ...
  - Address is presented to memory.
  - High-order 3 bits are used to select one chip.



Fundamental control operations.

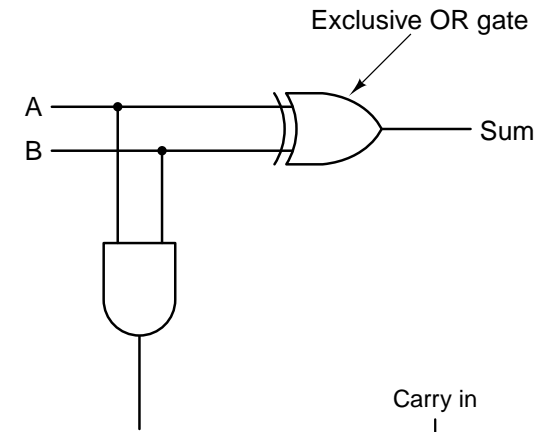
# Arithmetic Circuits

## Adders

- Half adder.

- Two inputs, two outputs.
- Sum of inputs in one output.
- Carry in other output.

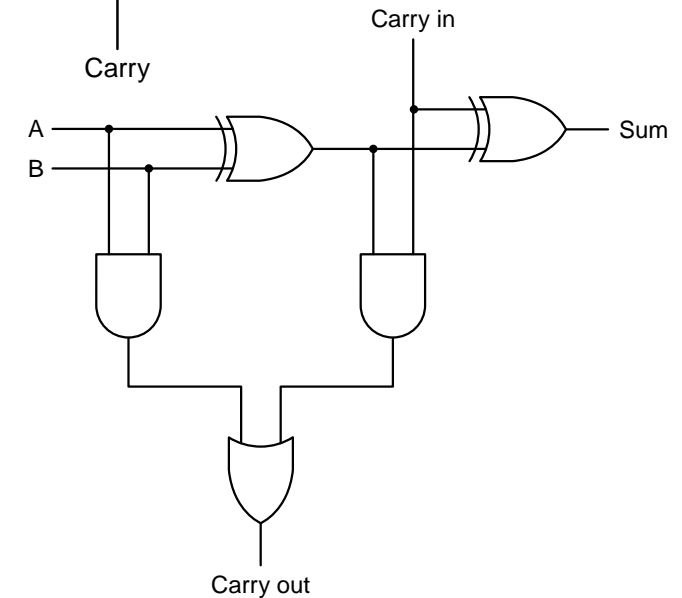
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



- Full adder.

- Three inputs, two outputs.
- Sum of inputs in one output.
- Carry but in other output.

A	B	Carry in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



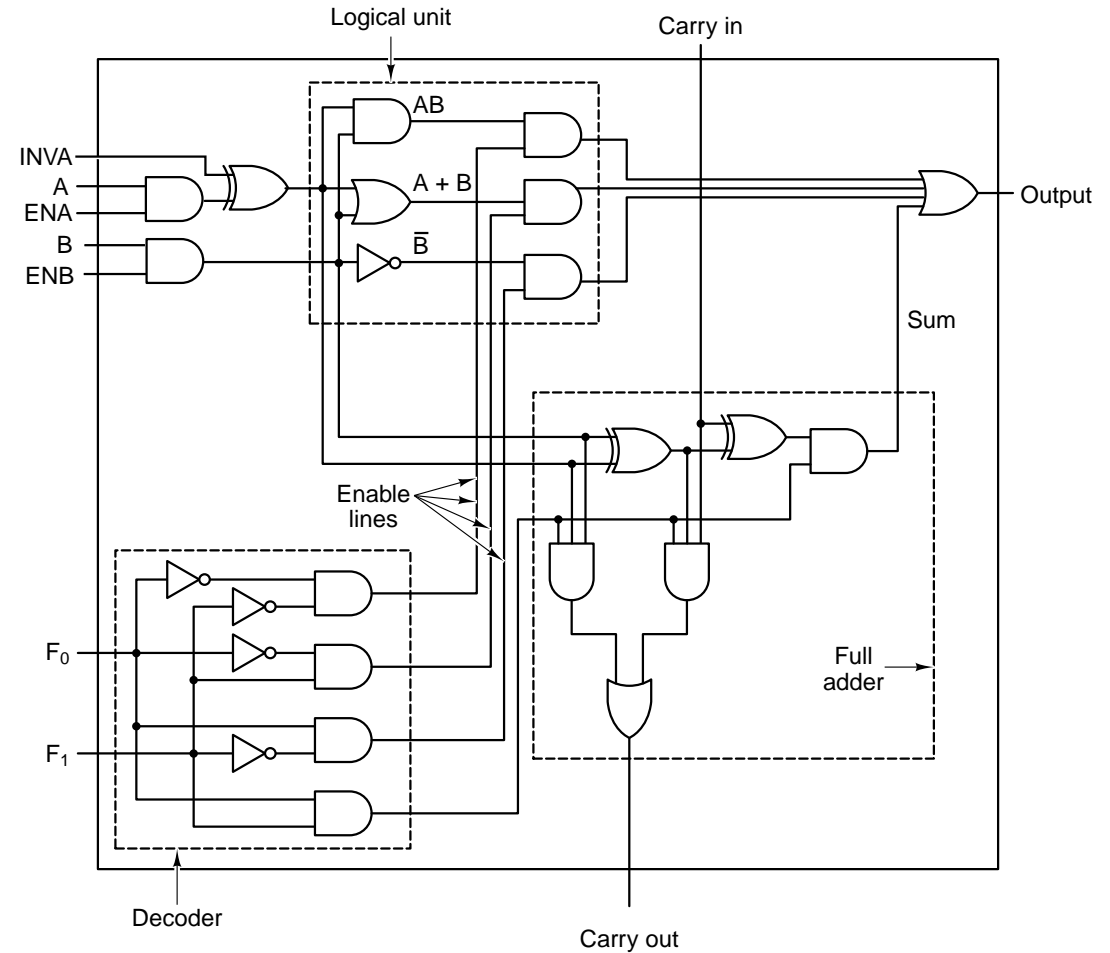
## Basis of 1 bit ALU.

# Arithmetic Logic Units

- 1 bit ALU.

- Inputs enabled or not (set to 0).
- Control input selects operation.
- AND, OR, NOT, Addition.

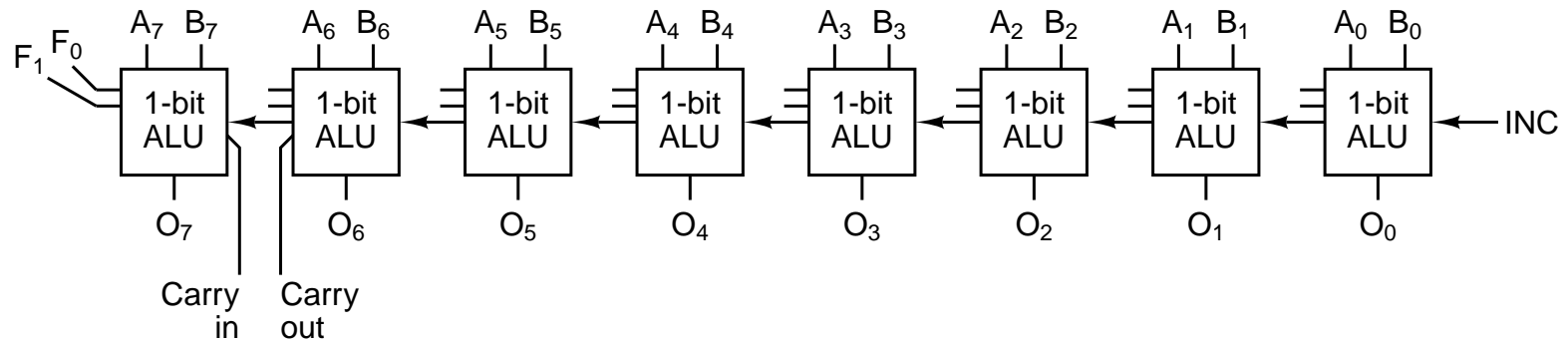
## Basis of $n$ bit ALU.



## Arithmetic Logic Units

- 8 bit ALU.

- Connection of 1-bit ALU slices.



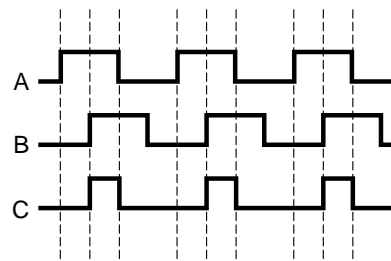
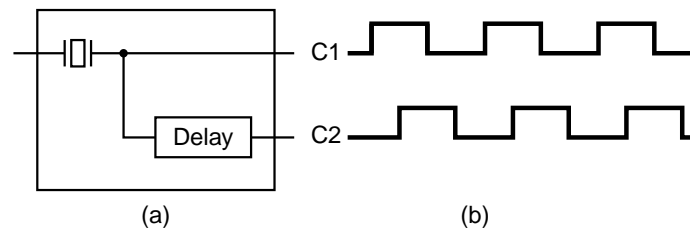
$n$ -bit ALUs can be constructed from 1-bit slices.

# Memory

## Clocks

In digital circuits, timing relations must be controlled.

- **Clock:** circuit that emits sequence of pulses (crystal oscillator).
  - Precise pulse width; precise interval between pulses (**clock cycle time**).
- **Derived clock signals can be constructed by delays.**
  - By combination, clock cycle can be divided in subcycles.

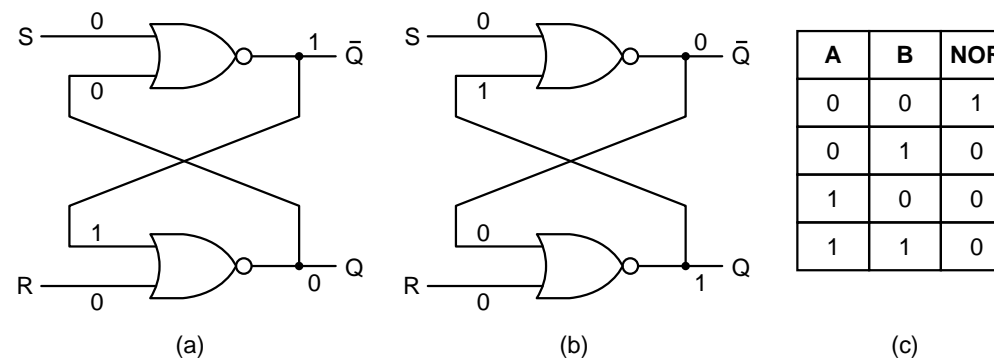


## Latches

Circuits that remember “previous” input values.

- SR latch.

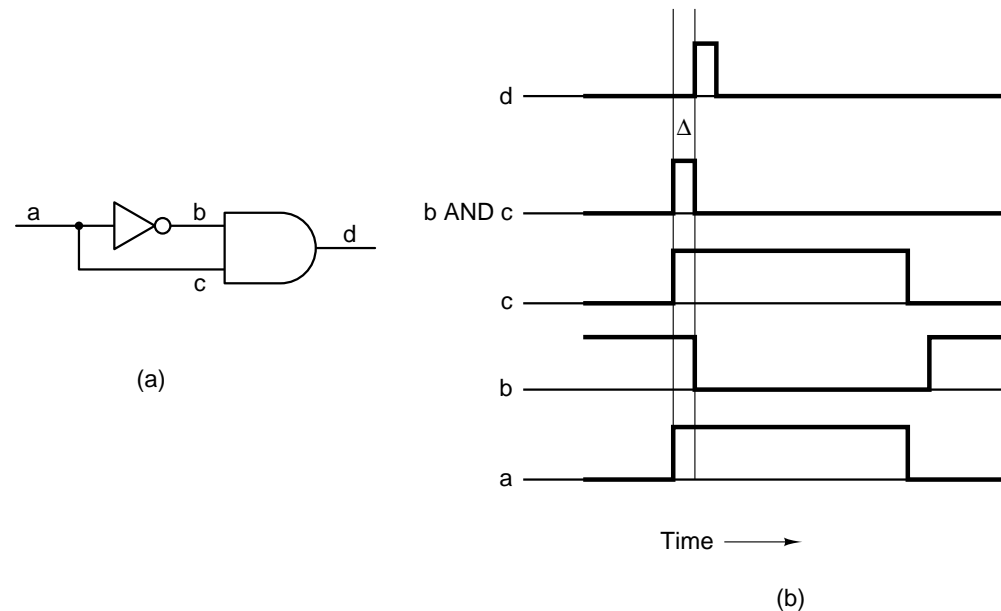
- $S$  input: sets the latch;  $R$  input: resets the latch.
- If  $S$  is 1 and  $R$  is 0,  $Q$  gets 1.
- If  $R$  is 1 and  $S$  is 0,  $Q$  gets 0.
- If  $R$  and  $S$  are 0,  $Q$  remains unchanged.
- $\bar{Q}$  is inverse of  $Q$ .



## Pulse Generators

Circuits which generates very short pulses.

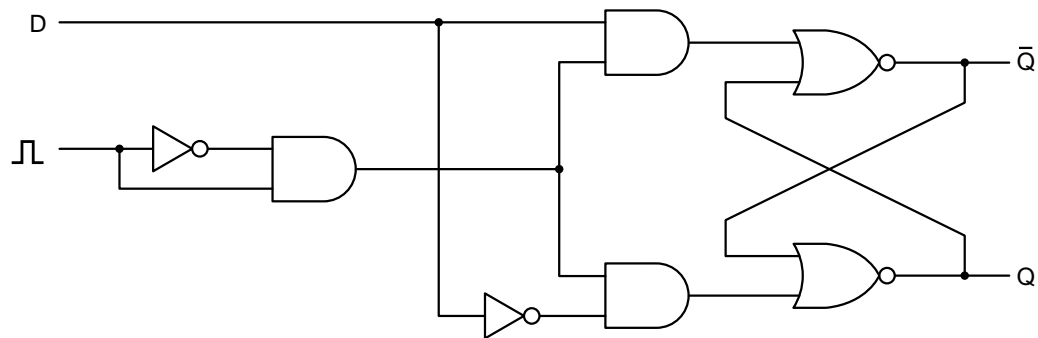
- A signal  $a$  and its negation  $b$  are fed into an AND gate.
  - When signal  $a$  is set, negation  $b$  is slightly delayed.
  - For a short period, there is a signal on output  $d$ .



## Flip-Flops

Circuit which stores a data value at a precise time.

- Combination of a pulse generator and a latch.
  - Inputs of latch are  $D$  AND  $\bar{D}$  (no inconsistency may occur between  $R$  and  $S$ ).
  - Inputs are conjoined with output of pulse generator (input is read at well-defined time).



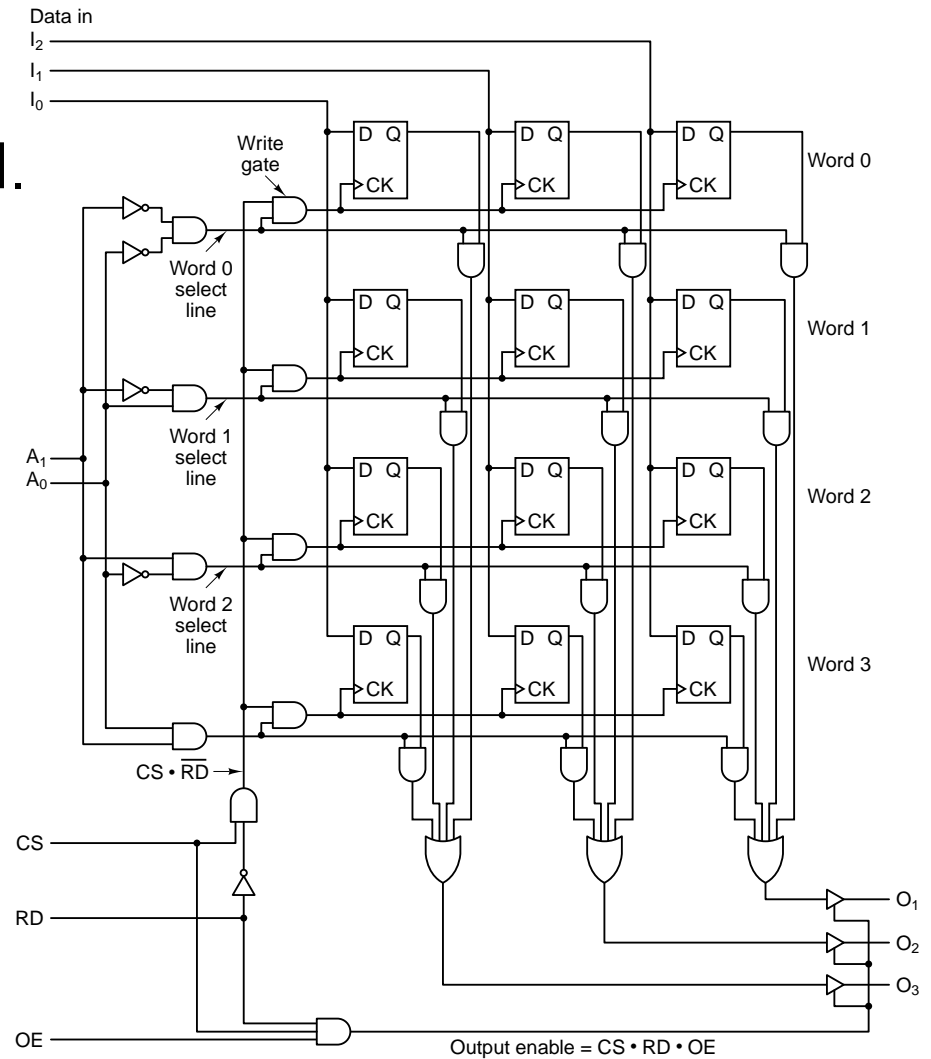
Current value of  $D$  is read and stored a fixed time after clock signal.

# Memory Organization

Individual words must be addressed.

- $4 \times 3$  memory.
- Input lines  $I_i$ .
- Address lines  $A_j$ .
- Chip select signal  $CS$ .
- $RD$  signal for read/write.
- $OE$  signal for output enable.

Simple regular structure.



## RAMs: Random Access Memories

- **SRAM:** Static RAM.
  - Constructed from flip-flops.
  - Content is retained as long as power is kept on.
  - Very fast (few nanoseconds access time), used for caches.
- **DRAM:** Dynamic RAM.
  - Each cell consists of transistor and capacitor only.
  - Capacitor can be charged or discharged (0 or 1).
  - Charge leaks out, bit needs to be **refreshed** every few milliseconds.
  - Rather slow (tens of nanoseconds access time), used for main memory.
- **SDRAM:** Synchronous DRAM.
  - Hybrid of SRAM and DRAM.
  - Access driven by synchronous clock.
  - Used for main memory today.

## ROMs: Read Only Memories

- Content is inserted during manufacture.
  - Content cannot be changed or erased, is retained even if power is switched off.
  - Data are etched via mask into silicon surface.
- **PROM**: Programmable ROM.
  - Content can be written **once**.
  - Contains array of tiny fuses that can be blown out by high voltage.
- **EPROM**: Erasable PROM.
  - Data can be erased by exposure to ultraviolet light.
- **EEPROM**: Electric EPROM.
  - Data can be erased by electric pulses.
- **Flash Memory**: memory is block erasable and rewritable.
  - Compact Flash card, Smartmedia card, ...

## **CPU Chips and Buses**

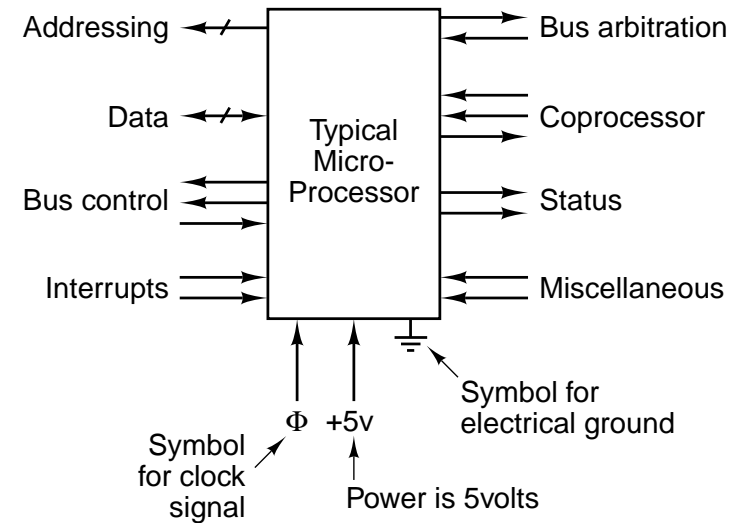
## CPU Chips

All modern CPUs are contained on a single chip.

- Interaction with outside world through set of pins.
  - Input signals, output signals, bidirectional signals.
  - Connected to similar pins on memory chips and I/O chips via bus.
- **Address pins:**
  - CPU puts memory address on its address pins to load a memory cell.
- **Data pins:**
  - Memory replies by putting requested word on the CPU's data pins.
- **Control pins:**
  - CPU asserts via some control lines when it wants to read data.
  - Memory asserts via some control lines when data are available.

## Control Pins

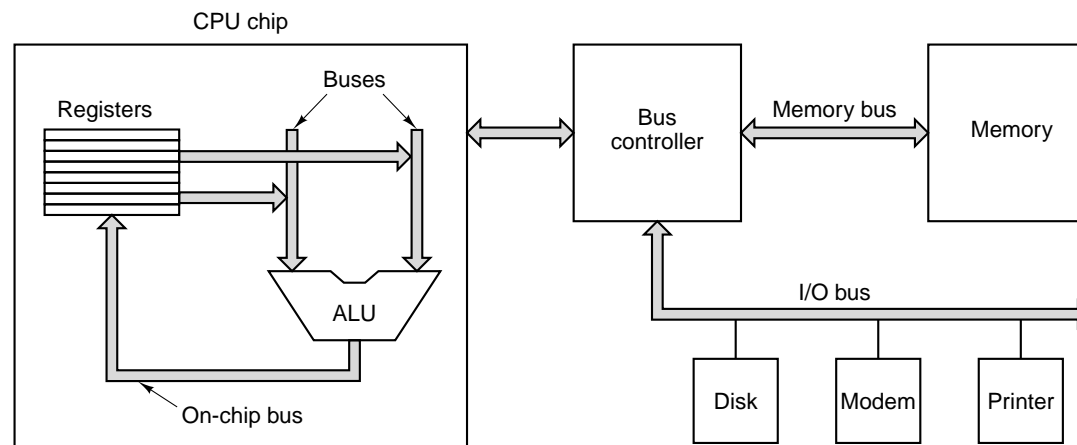
- Bus control.
  - CPU tells bus whether it wants to use it.
- Interrupts.
  - I/O devices tell CPU to interrupt current program.
- Bus arbitration.
  - Used for regulating traffic on the bus.
- Coprocessor signaling.
  - Used for making/granting requests to auxiliary processors.
- Status.
  - Accept or provide status information.



## Computer Buses

Electrical pathways shared between multiple devices.

- Various functions.
  - Internal to CPU: transport data to and from ALU.
  - External to CPU: connect it to memory or to I/O devices.
- Multiple external buses with special properties.
  - Memory bus, I/O bus, graphics bus, ...



## Computer Buses

- Various types of buses:

- PCI bus (PCs), SCSI bus (PCs and workstations), Universal Serial Bus (USB, PCs), FireWire (consumer electronics), ...

- Bus Protocols:

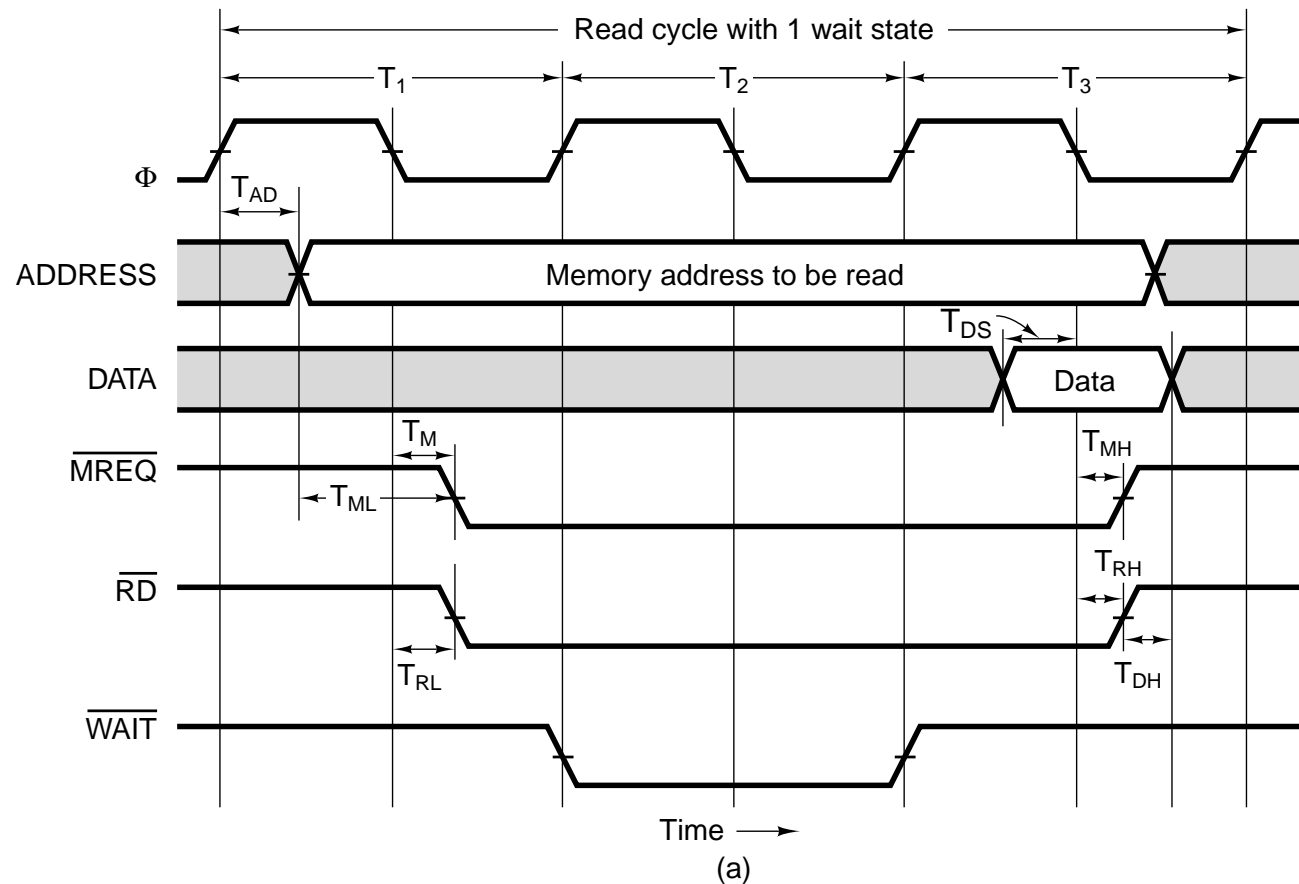
- Sets of rules that devices must obey to use the bus.
- **Masters**: active devices that can initiate bus transfers.
- **Slaves**: passive devices that wait for requests.
  - \* CPU master, I/O device slave: initiate data transfer.
  - \* I/O device master, memory slave: DMA (Direct Memory Access).

- Design parameters:

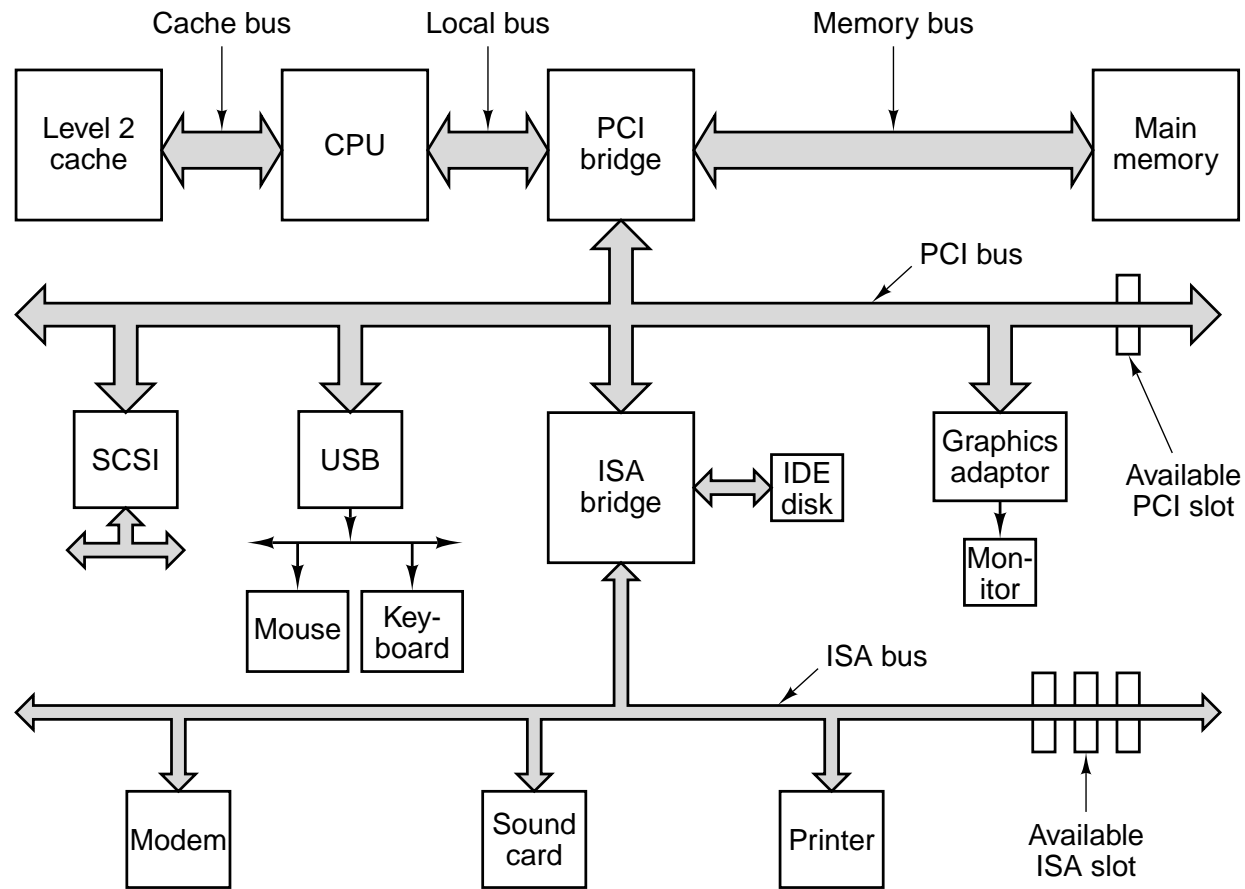
- Bus width: number of address and data lines (e.g. 64 bits).
- Bus cycle time: number of transfers per second (e.g. 100 MHz).
- Bus bandwidth = data width \* cycle time (781 MB/s).

## Synchronous Buses

All activities take a fixed number of bus cycles.

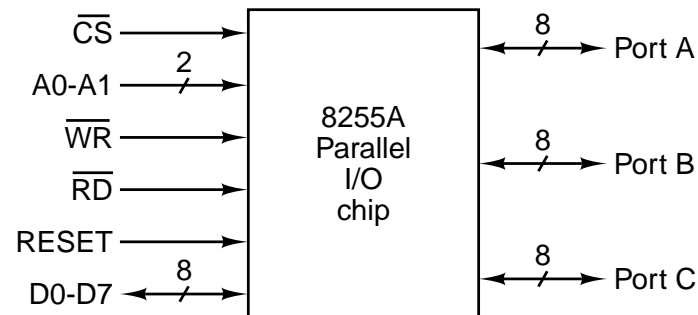


# Example: Pentium PC



## I/O Controllers

- **UART:** Universal Asynchronous Receiver Transmitter.
  - Can read a byte from data bus and output it bit by bit on a serial line.
  - Can read a byte bit by bit from a serial line and put it on the data bus.
- **PIO:** Parallel Input/Output chip.
  - Chip that connects to the parallel interface of a computer.
  - Computer writes 8 bit number into a register of the chip.
  - Chip puts 8 bit number on the output lines until register is rewritten.



## Memory Mapped I/O

I/O registers are assigned part of the memory address space.

- CPU reads/writes corresponding memory locations.
  - Chip Select (CS) pin of PIO chip is wired to bus address lines.
  - If corresponding address is issued, data pins of PIO chip take value from bus data lines.

