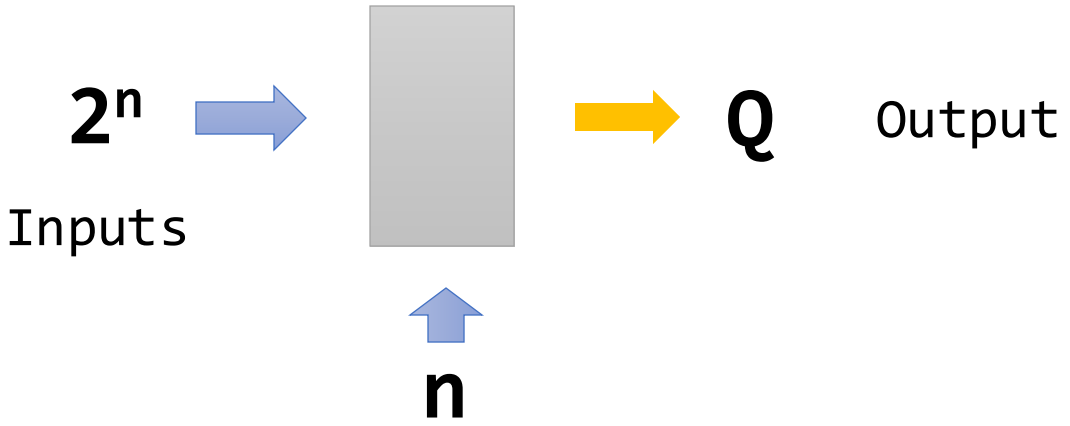
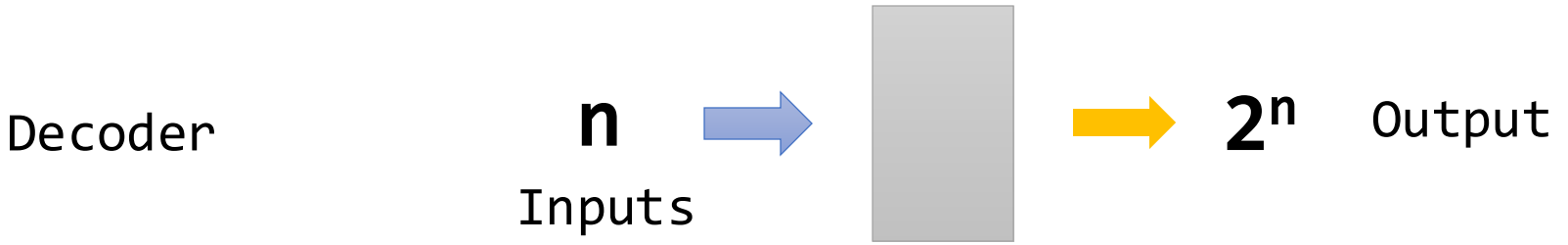
Summary

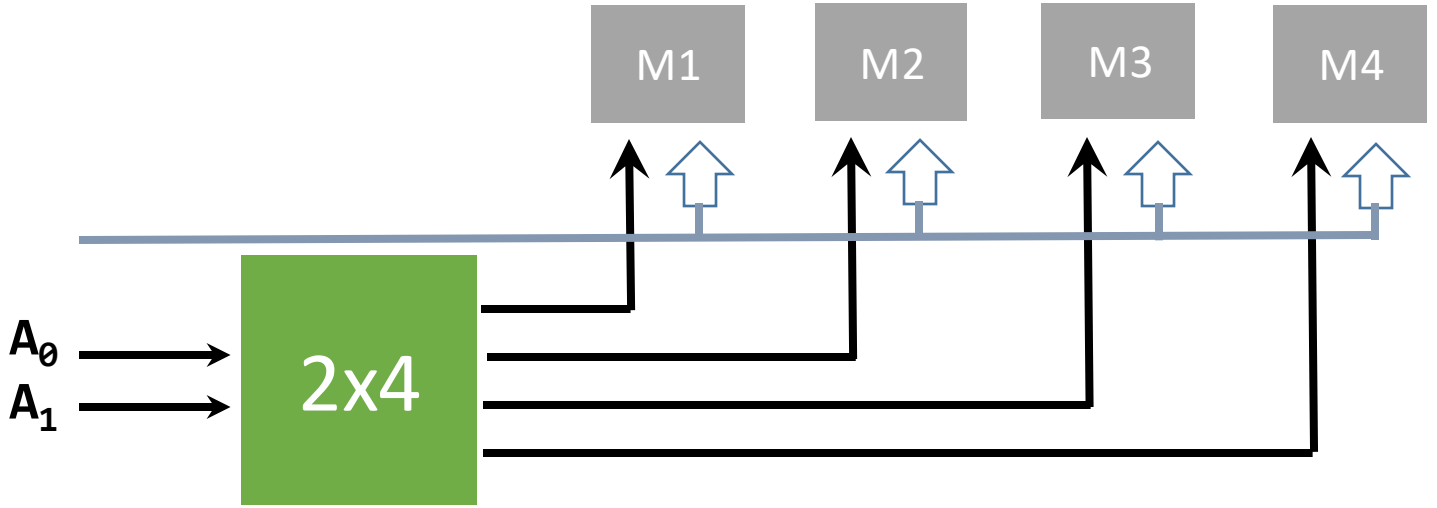
Multiplexer



Summary

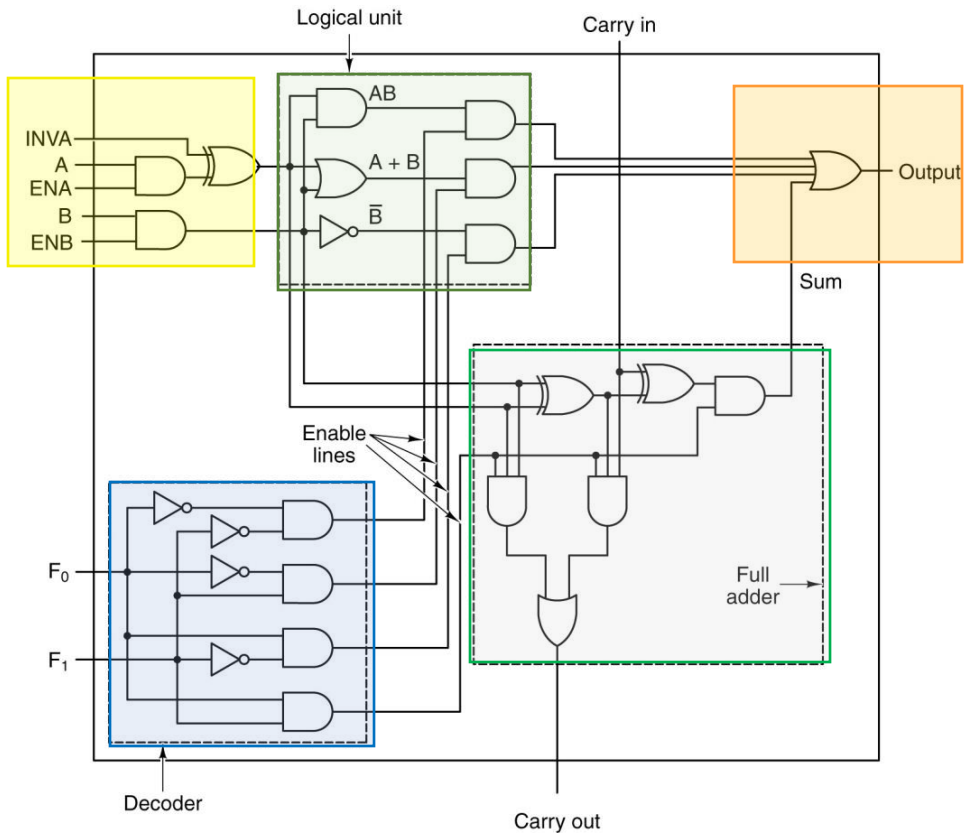


Example: Memory



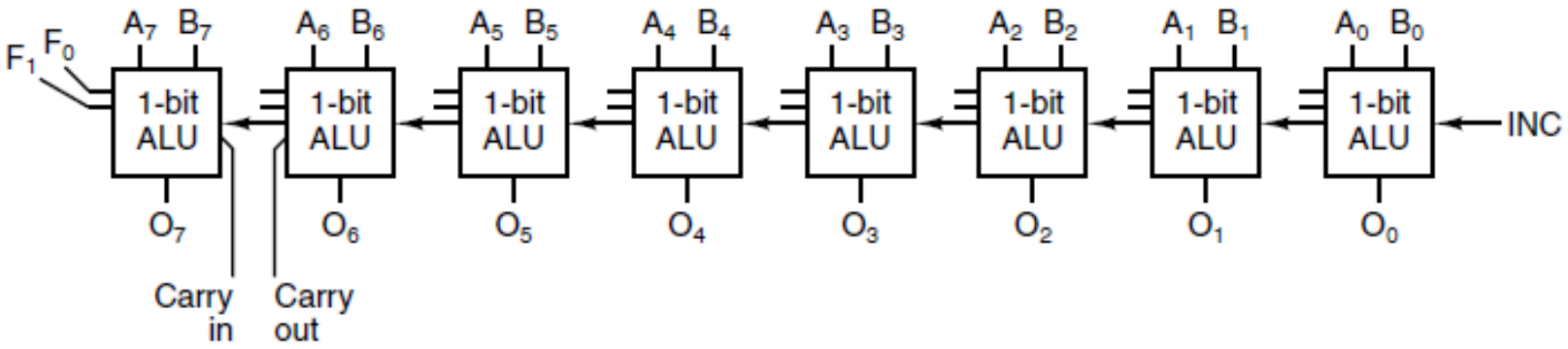
Summary

1-bit ALU(arithmetic logic unit)



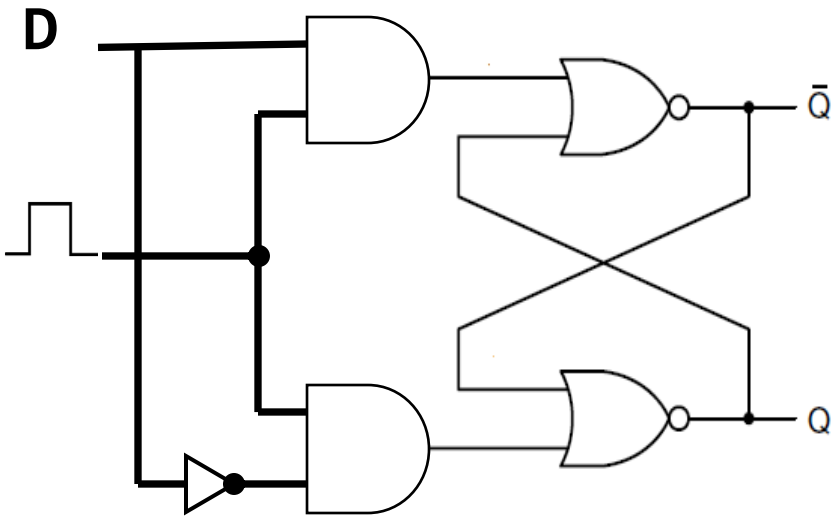
Summary

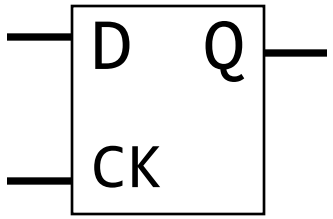
8-bit ALU(arithmetic logic unit)



Summary

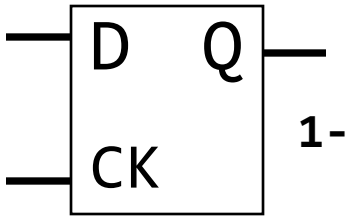
Clocked D Latches





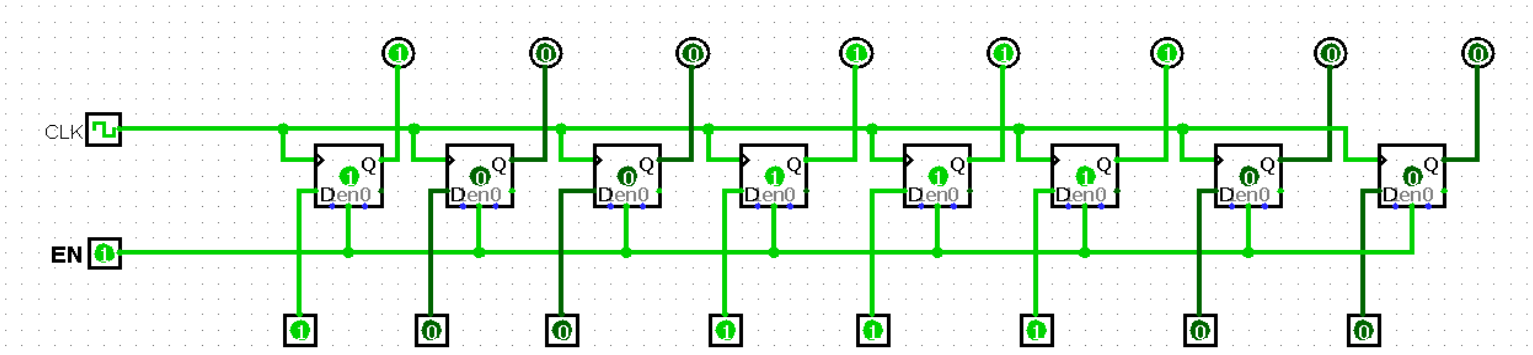
Summary

D Latches applications



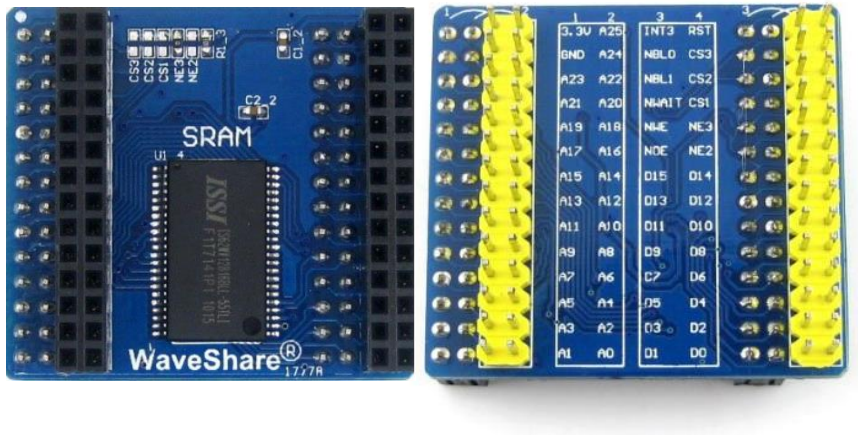
1-bit memory

8-bit memory(Register)

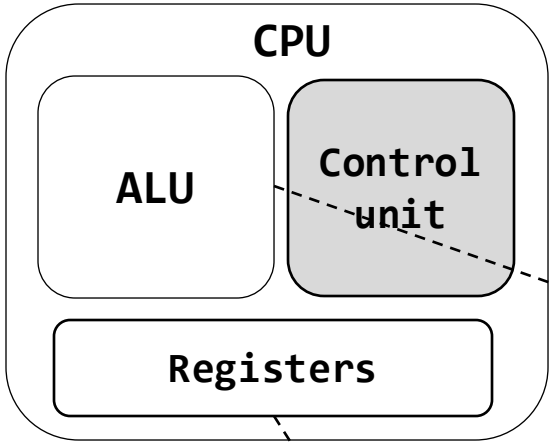


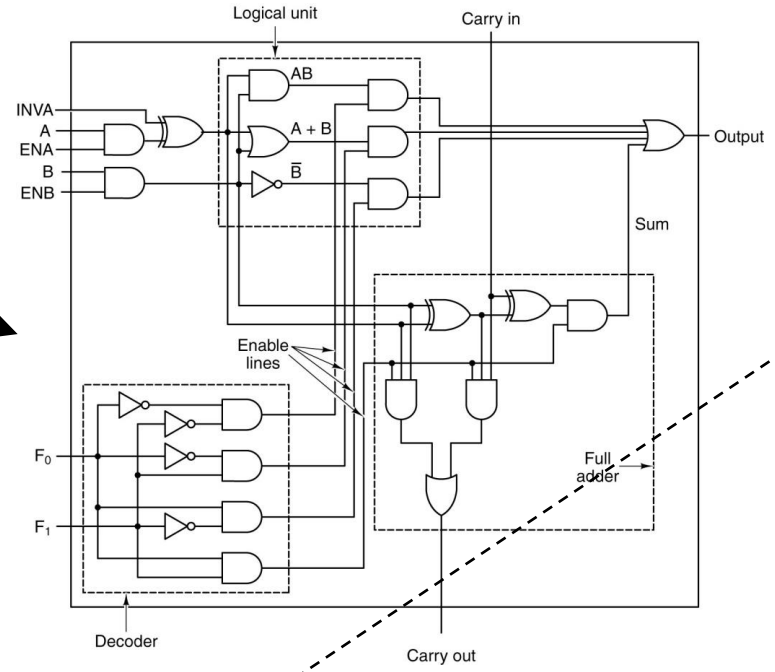
DRAM

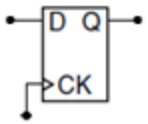
SRAM

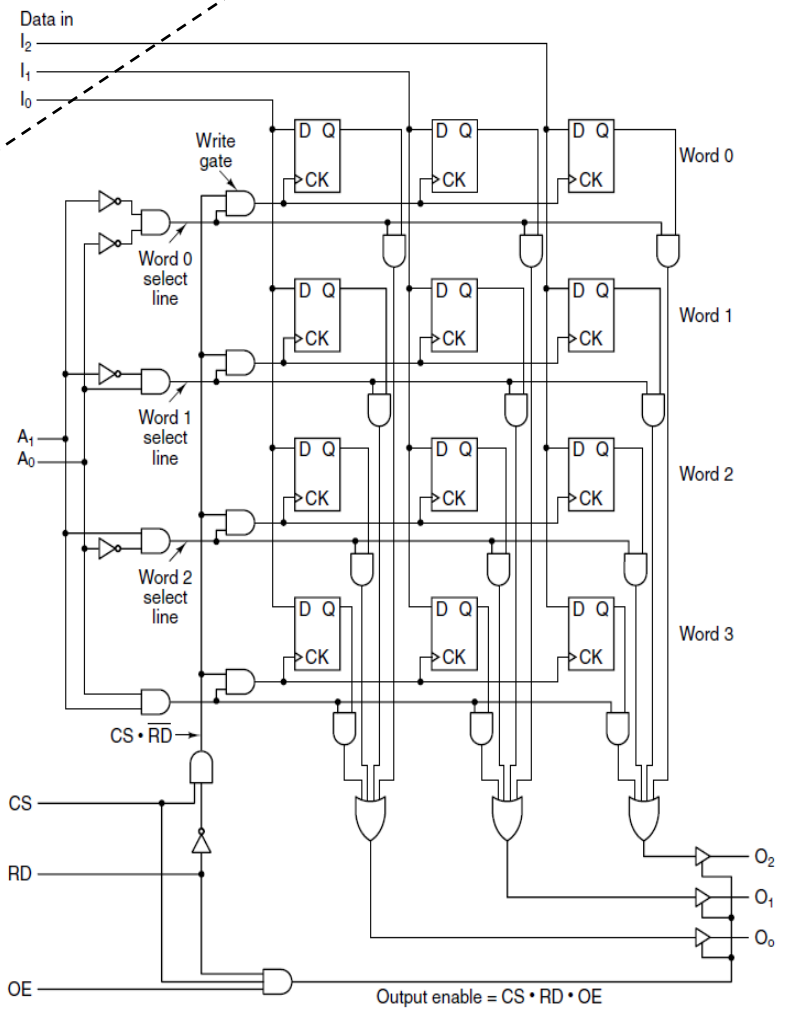


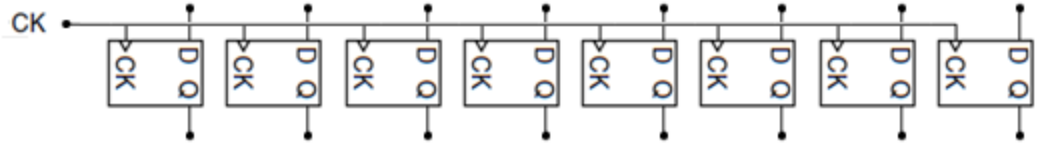
Summary


CPU
ALU
Control unit
Registers


Logical unit
Carry in
AB
A + B
$\overline{B}$
INVA
A
ENA
B
ENB
Output
Sum
Enable lines
F_0
F_1
Full adder
Decoder
Carry out

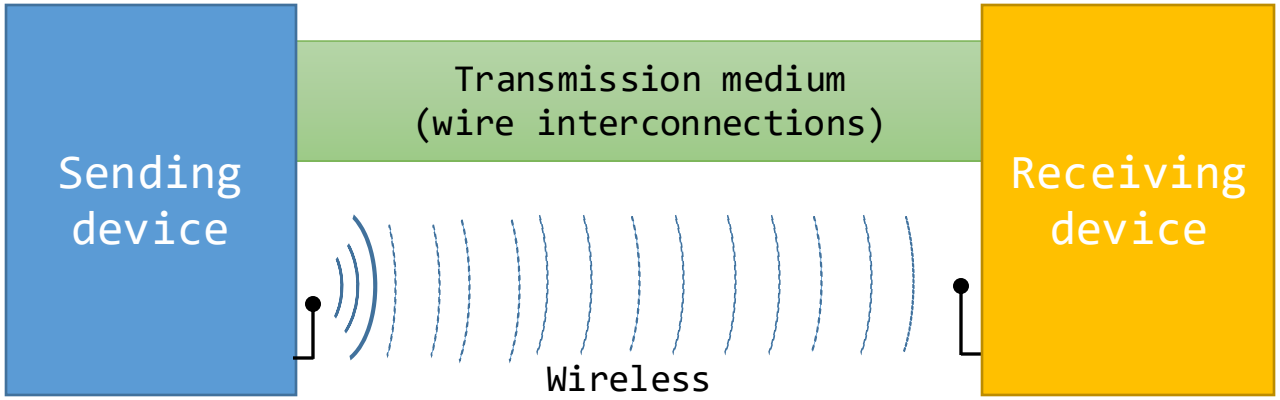

D Q
CK


Data in
I_2
I_1
I_0
Write gate
Word 0 select line
Word 1 select line
Word 2 select line
Word 0
Word 1
Word 2
Word 3
A_1
A_0
CS • $\overline{RD}$
CS
RD
OE
O_2
O_1
O_0
Output enable = CS • RD • OE


CK

Logic gates

Data transmission

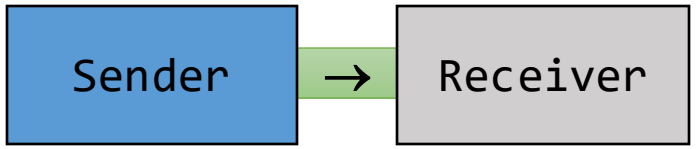


Data transmission systems between two devices

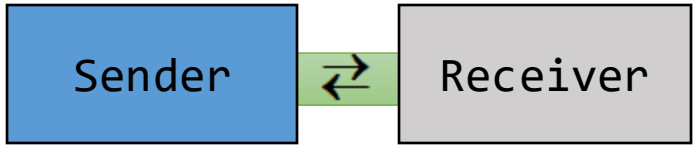
1) Transmission media such as coaxial cable, twisted pair cable, optical fiber cable, without modulation (baseband transmission)
2) Wireless
   Using space with modulation (broadband transmission)

Logic gates

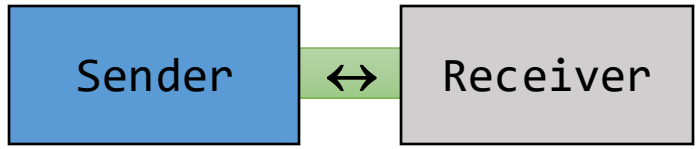
Data transmission



Simplex

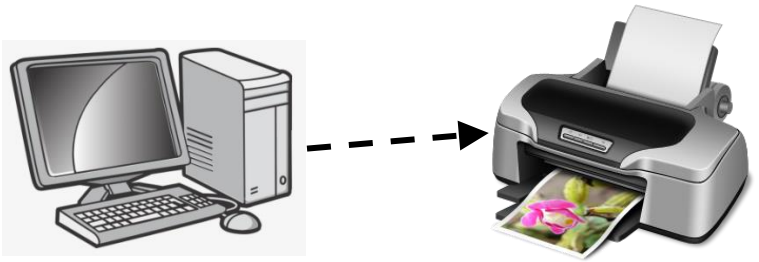


Half-duplex

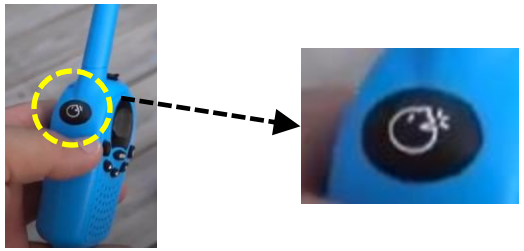


Full-duplex

Simplex: data flows in only one direction from the sender to the receiver e.g. computer to printer



Half-duplex: a data flow occurs in both ways but not simultaneously e.g. walkie talkies.



Full-duplex: data flow occurs in both ways simultaneously.

Logic gates

Data transmission (serial vs parallel)

Serial

One bit at a time is transmitted e.g. USB

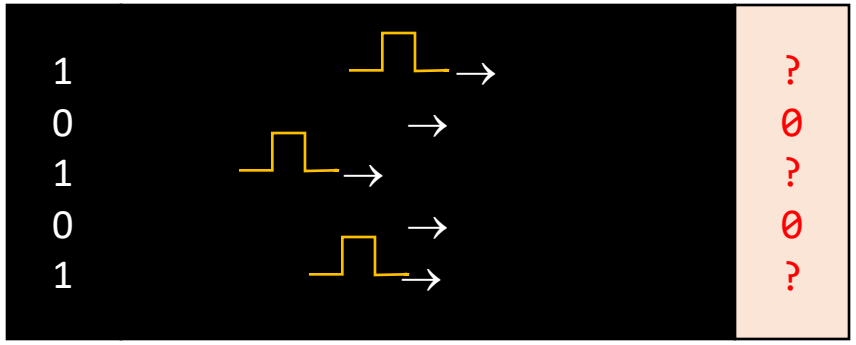
cheaper (fewer lines), complex hardware

10101 →

Parallel

Several bits at a time are transmitted e.g. ISA and PCI bus

1) data transmission is faster causing higher data rate
2) expansive (many lines)
3) signals may not traverse at the same speed causing skewing in the data and in turn causing the data to be more susceptible to error. To avoid the problem the parallel transmission is used in a very short distances.
4) Simpler controller hardware



Logic gates

Serial data transmission and data rate

In a serial data transmission, the speed of data transfer is called data rate. Two ways to define the data rate:

Bit rate : the number of bits per second (bps)
Baud rate : the symbol(or signal) rate or the number of data symbols (signals) per second

$$\text{Baud rate} = \frac{\text{Bit rate}}{\text{Number of bits per symbol}}$$

01010101010101010101010000101101010000110111100

Bit rate = 48 bps
or 48/4 = 12 symbol per second = baud rate
1 symbol = 4 bits

010101010101010101010100001011010100001101111100

Bit rate = 48 bps
or 48/12= 4 symbol per second = baud rate
1 symbol = 12 bits

Example:
A analog signal carries 8 bits in each signal. If 2000 signal units are sent per second, find the baud and bit rate.

Baud rate : 2000 (signal/s)
Bit rate = Baud rate x (# of bits per signal) = 2000 x 8 = 16000 bps = 16 kbps

Logic gates

Data transmission - Buses

A set of physical wires used for transmitting data or communication between various devices. A bus is either a parallel or serial bus, and either an internal or an external bus.

General Bus Characteristics:

Width : The number of bits that a bus can transmit at one time and can vary from 1 bit for a serial bus up to 64 bits for a parallel bus.

Frequency : The clock frequency at which a bus can operate

Transfer speed : The number of bytes per clock cycle

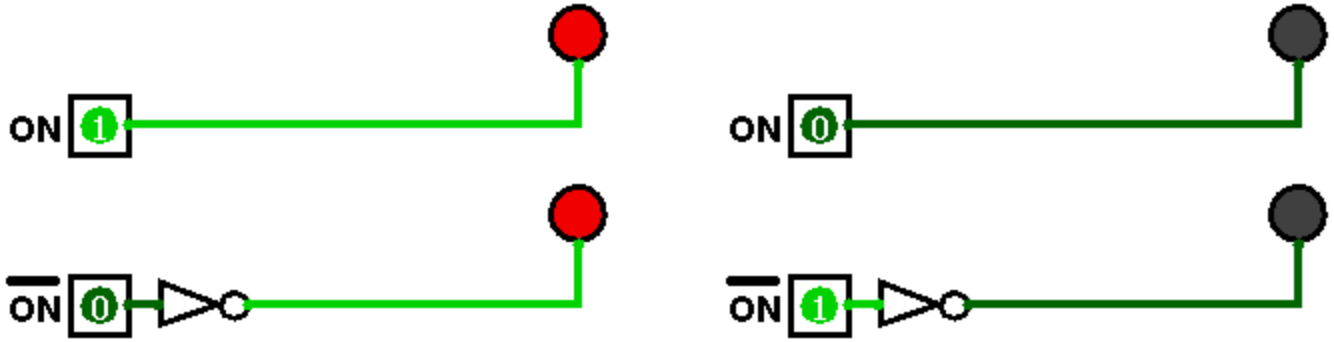
Bandwidth :
The number of bytes per clock cycle **times** the number of clock cycles per second

Logic gates

Asserted and Negated

- A signal is **asserted** (it goes high or goes low) means that it is set to cause some action. Thus, for some pins, asserting it means setting it high. For others, it means setting the pin low.

- The opposite of asserted is **negated**. When nothing special is happening, pins are negated.



Logic gates

CPU Chips and Buses

CPU Chips

CPU pins

Each CPU chip has a set of pins , for communication with other components e.g. memory and I/O devices by presenting signals on its pins and accepting signals on its pins.

The pins on a CPU chip

1) Power (usually +1.2 to +1.5 volts)
2) Ground
3) Clock signal (a square wave at some well-defined frequency)
4) Address
5) Data
6) Control
    A. Bus control
    B. Interrupts
    C. Bus arbitration
    D. Coprocessor signaling
    E. Status
    F. Miscellaneous

Logic gates

CPU Chips and Buses

CPU Chips

The pins on a CPU chip
Address, Data, Control

performance of a CPU

1. **m** address pins can address $\mathbf{2^m}$ memory locations, e.g. m = 16, 32, 64
2. a chip with **n** data pins can read or write an **n**-bit word in a single operation. e.g. n = 8, 32, and 64

   e.g. 8 data pins -> 32-bit word (4 operations)
   32 data pins -> 32-bit word (1 operations) faster

Logic gates

CPU Chips and Buses

CPU Chips

CPU control pins

1. **Bus control** - to control the rest of the system and tell it what it wants to do (e.g. whether the CPU wants to read or write memory or do something else)
2. **Interrupts** - tells an I/O device to start an operation and then go off and do some other activity, while the I/O device is doing its work, When the I/O has been completed, the I/O controller chip asserts a signal on one of these pins to interrupt the CPU and have it service the I/O device, for example to check whether if I/O errors occurred.
3. **Bus arbitration** - needed to regulate traffic on the bus
4. **Coprocessor signaling** - To facilitate communication between CPU and coprocessor (GPU), special pins are provided for making and granting various requests.
5. **Status** - running, halted, etc.
6. **Miscellaneous**

Logic gates

CPU Chips and Buses

CPU Chips

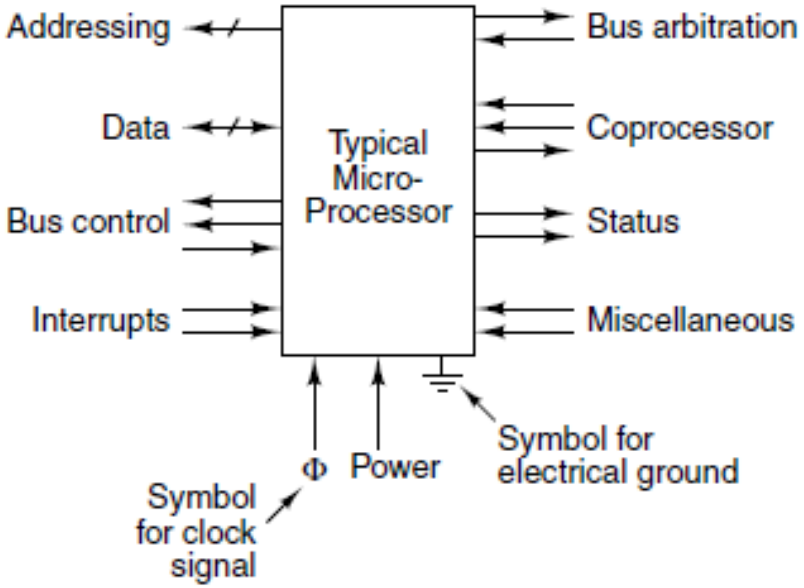


Figure 3-34. The logical pinout of a generic CPU. The arrows indicate input signals and output signals. The short diagonal lines indicate that multiple pins are used. For a specific CPU, a number will be given to tell how many.

Logic gates

CPU Chips and Buses

Computer Buses

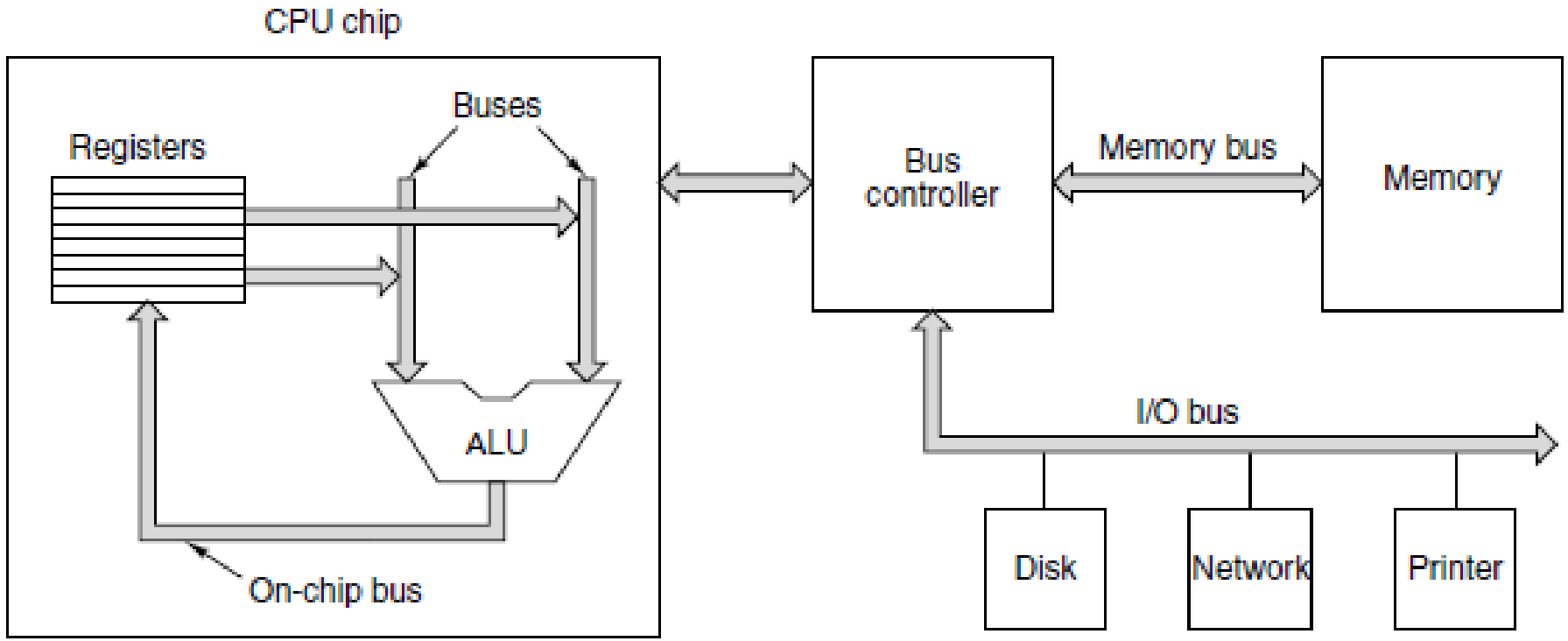
A bus is a common electrical pathway between multiple devices

Internal to CPU
transport data to and from the ALU

External to CPU
to connect it to memory or to I/O devices

Logic gates

Data transmission (personal computers)



A computer system with multiple buses.

Fat arrow: Address, data, and control lines

Logic gates

CPU Chips and Buses

Computer Buses

Internal BUS (CPU)
Free to use any kind of bus inside the chip

External BUS
In order to make it possible for boards designed by third parties to attach to the system bus, there must be well-defined rules about how the external bus works, which all devices attached to it must obey. These rules are called the bus protocol. In addition, there must be mechanical and electrical specifications, so that third-party boards will fit in the card cage and have connectors that match those on the motherboard mechanically and in terms of voltages, timing, etc.

Examples:
Omnibus (PDP-8), VME bus (physics lab equipment),
IBM PC bus (PC/XT), PCI bus (many PCs), SCSI bus (many PCs and workstations), Universal Serial Bus (modern PCs), and FireWire (consumer electronics)

Logic gates

CPU Chips and Buses

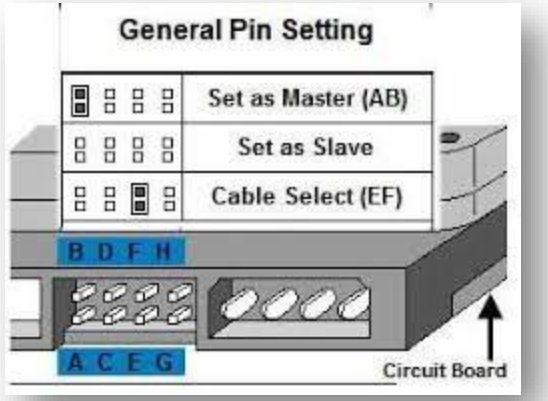
Computer Buses

How buses work?

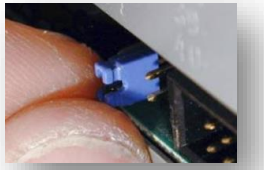
Masters: Some devices that attach to a bus are active and can initiate bus transfers
Slaves: others are passive and wait for requests

Master	Slave	Example
CPU	Memory	Fetching instructions and data
CPU	I/O device	Initiating data transfer
CPU	Coprocessor	CPU handing instruction off to coprocessor
I/O device	Memory	DMA (Direct Memory Access)
Coprocessor	CPU	Coprocessor fetching operands from CPU

Examples of bus masters and slaves.



Hard drive (master or slave setting)



Jumper shunt

A bus also has address, data, and control lines

Logic gates

CPU Chips and Buses

Not necessarily a one-to-one mapping between the CPU pins and the bus signals.

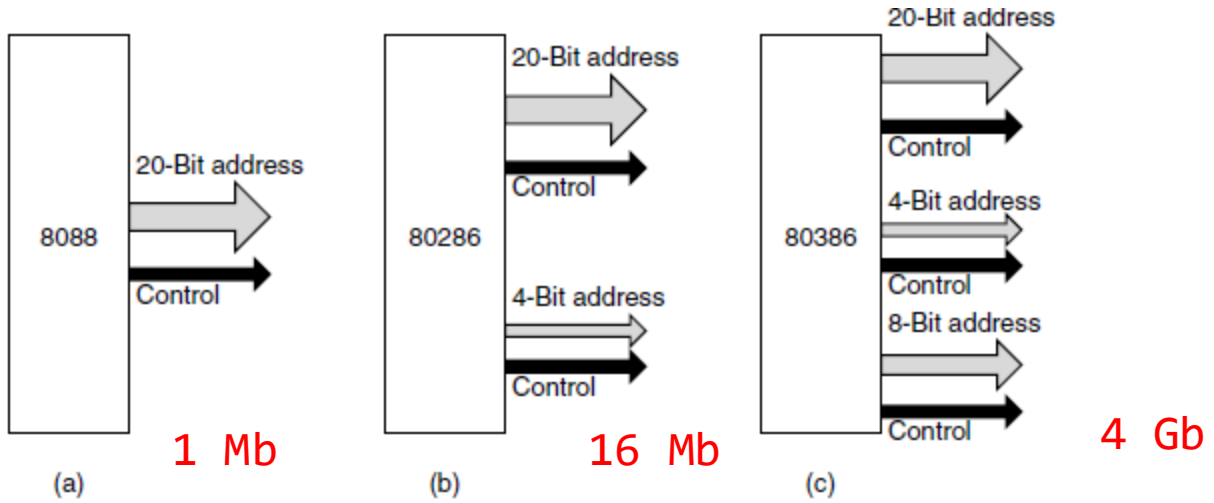
For example, some CPUs have three pins that encode whether the CPU is doing a memory read, memory write, I/O read, I/O write, or some other operation. A typical bus might have one line for memory read, a second for memory write, a third for I/O read, a fourth for I/O write, and so on (decoder).

Logic gates

CPU Chips and Buses

Bus Width

The more address lines a bus has, the more memory the CPU can address directly. Wide buses need more wires than narrow ones. They also take up more physical space (e.g., on the motherboard) and need bigger connectors. All of these factors make the bus more expensive.



Growth of an address bus over time.

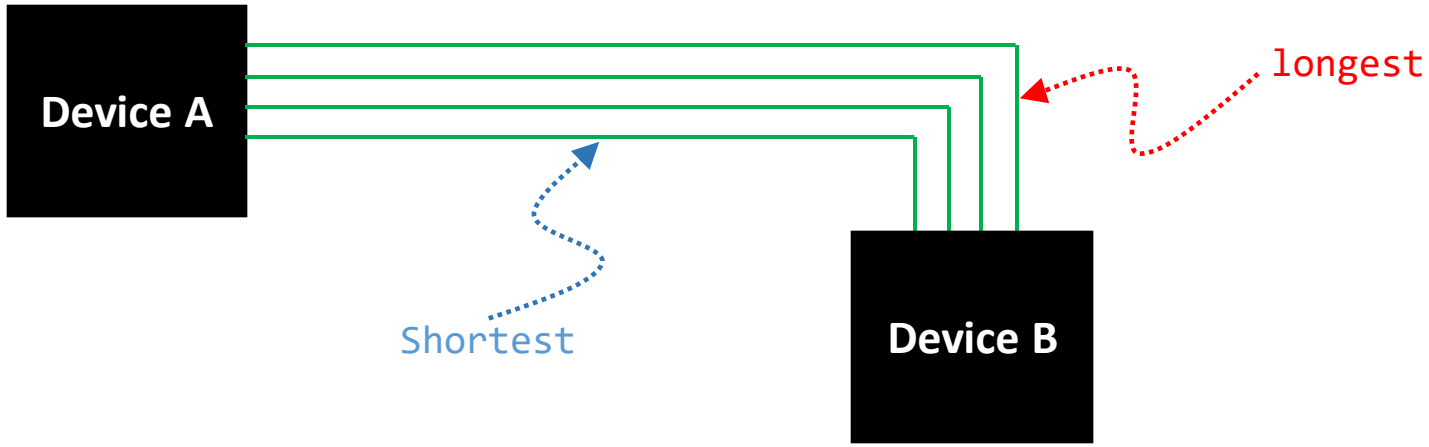
Logic gates

CPU Chips and Buses

Computer Buses

Two ways to increase the bandwidth of a bus

Decrease the bus cycle time (more transfers/sec) or increase the data bus width (more bits/transfer). Different lines travel at slightly different speeds (bus skew).



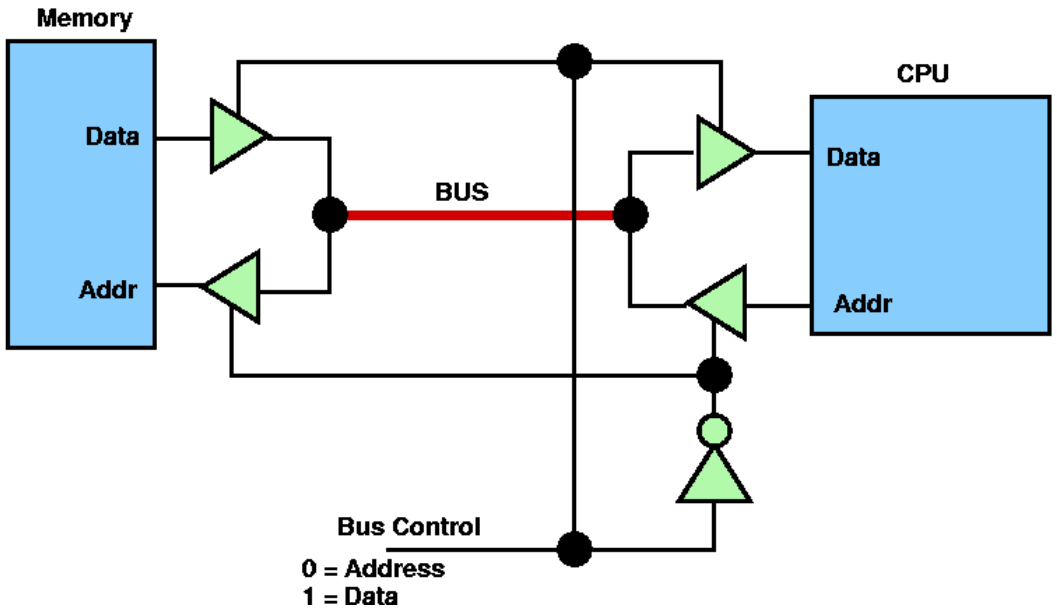
Logic gates

CPU Chips and Buses

Computer Buses

Multiplexed bus

Multiplexed bus, At the start of a bus operation, the lines are used for the address. Later on, they are used for data. Reduces bus width (and cost) but results in a slower system.



Logic gates

CPU Chips and Buses

Computer Buses

Two distinct categories depending on their clocking
Synchronous, has a master clock. All bus activities take an integral number of these cycles, called bus cycles.

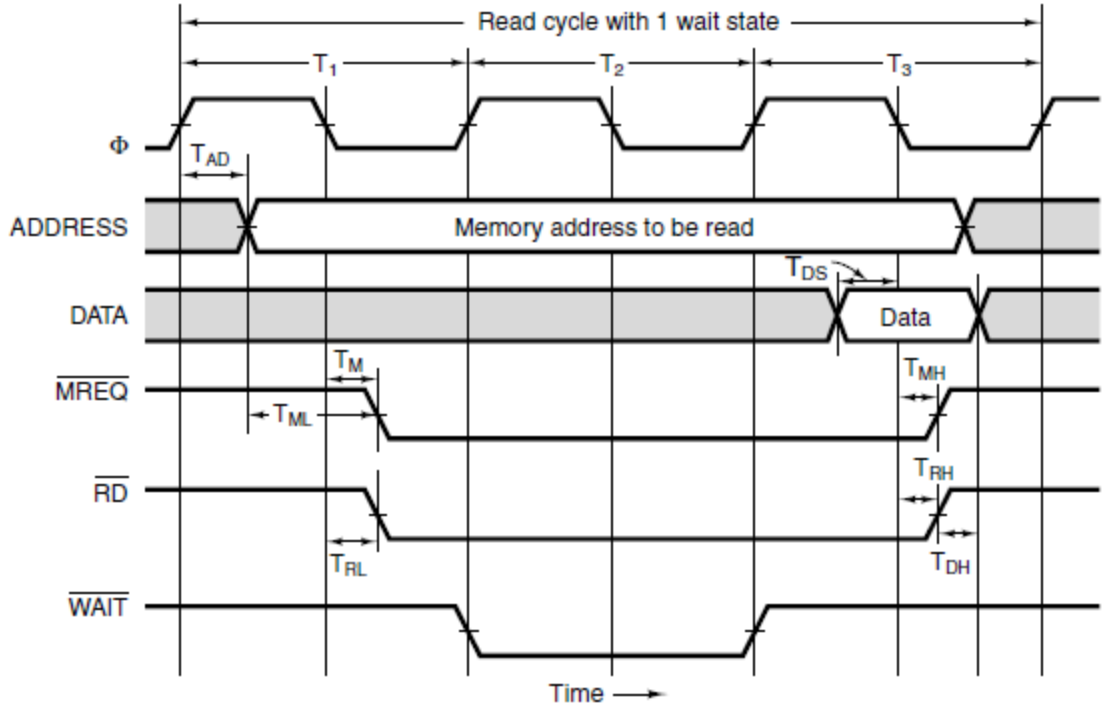
Asynchronous bus, does not have a master clock. Bus cycles can be of any length required and need not be the same between all pairs of devices.

Logic gates

CPU Chips and Buses

Computer Buses

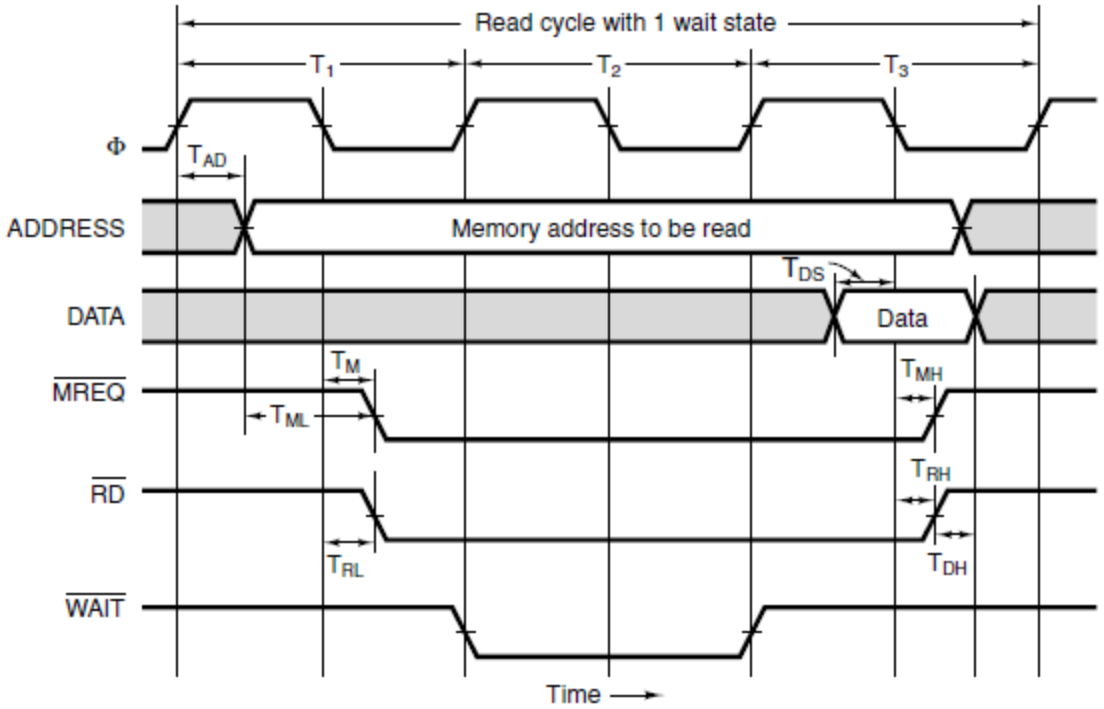
Synchronous



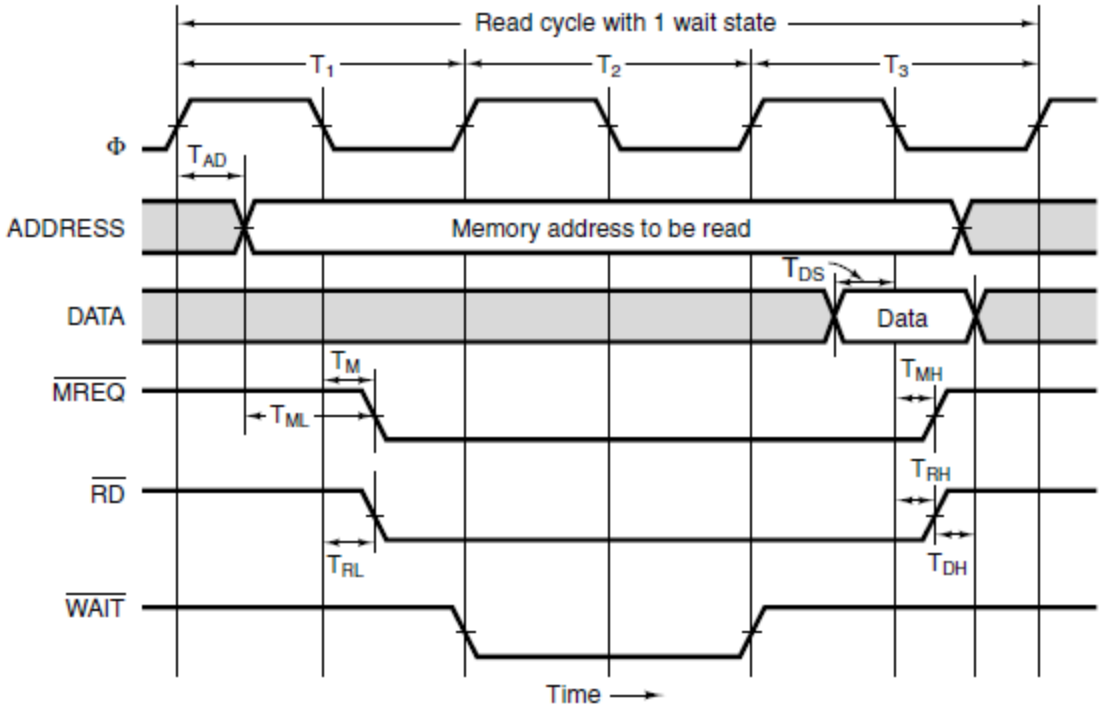
Example :

100-MHz clock or a bus cycle of 10 nsec

3 bus cycles to read a word



1) The first cycle starts at the rising edge of T1 and the third one ends at the rising edge of T4
2) The CPU puts the address of the word it wants on the address lines
3) MREQ(memory is being accessed) and RD are asserted (asserted for reads and negated for writes)
4) Since the memory takes 15 nsec after the address is stable it cannot provide the requested data during T2. To tell the CPU not to expect it, the memory asserts the WAIT line at the start of T2. This action will insert wait states



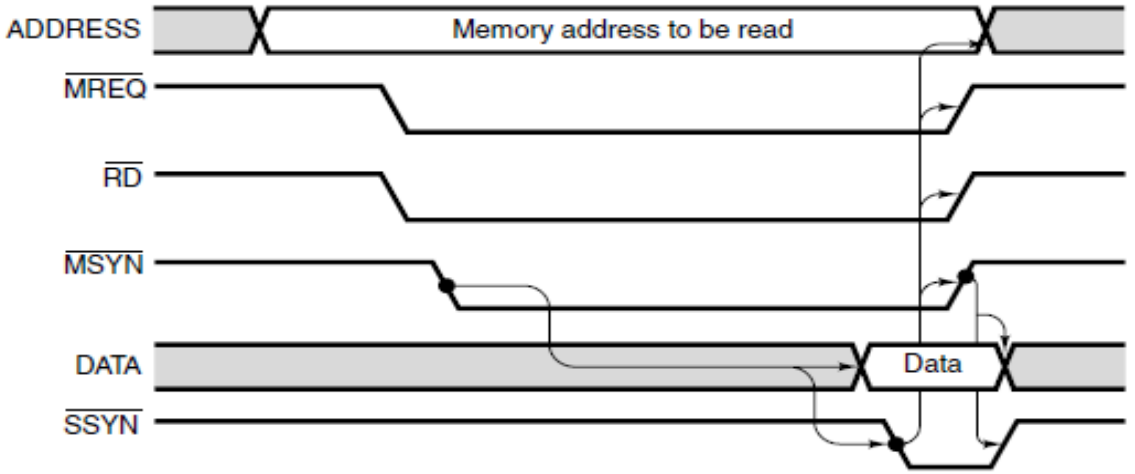
5) At the start of T3, when it is sure it will have the data during the current cycle, the memory negates WAIT
6) Having read the data, the CPU negates MREQ and RD. If need be, another memory cycle can begin at the next rising edge of the clock. This sequence can be repeatedly indefinitely.

Logic gates

CPU Chips and Buses

Computer Buses

Asynchronous



1. Bus master has asserted the address, MREQ, RD
2. Asserts a special signal that we will call MSYN (Master SYNchronization)
3. When the slave sees this, it performs the work as fast as it can and then asserts SSYN (Slave SYNchronization).
4. As soon as the master sees SSYN asserted, it knows that the data are available so it latches them and then negates the address lines, along with MREQ, RD, and MSYN.

Logic gates

CPU Chips and Buses

Bus Arbitration

Sometimes we have more than one bus master (e.g. Coprocessors and I/O). Bus arbitration decides who goes next. Many CPUs have the arbiter built into the CPU chip, but sometimes a separate chip is needed.

Bus Arbitration

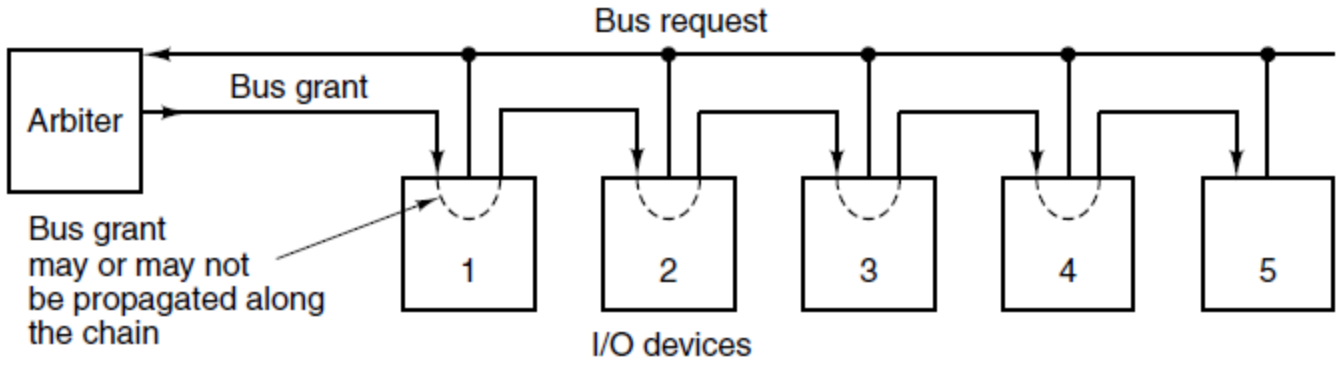
1) Centralized (one-level and two-levels)
    a) A bus arbiter is needed
    b) It has the property that devices are effectively assigned priorities depending on how close to the arbiter they are. The closest device wins

2) Decentralized
    a) No bus arbiter is needed

Logic gates

CPU Chips and Buses

Centralized Bus Arbitration



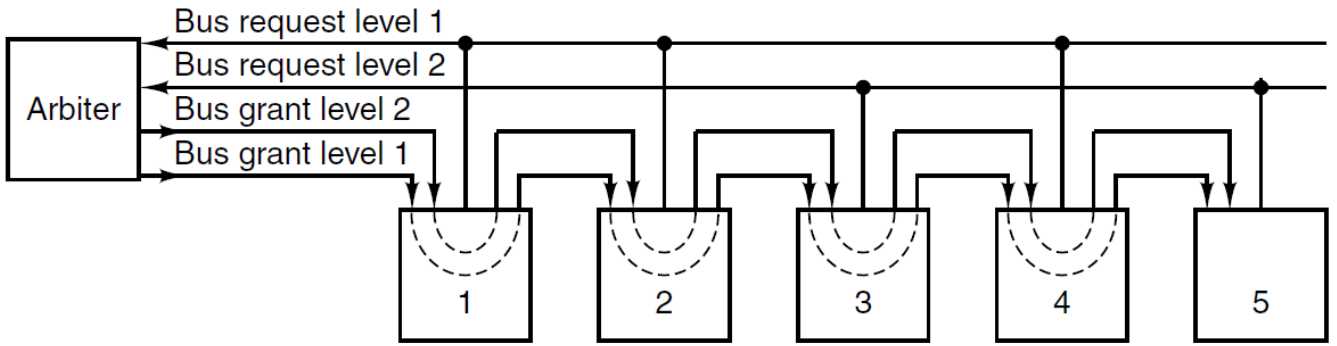
A centralized one-level bus arbiter using daisy chaining

1) If bus is not busy a bus request is made
2) The arbiter issues a grant by asserting the bus grant line
3) If device gets bus grant, marks bus busy otherwise it propagates the grant to the next device in line which behaves the same way, and so on until some device accepts the grant and takes the bus

Logic gates

CPU Chips and Buses

Centralized Bus Arbitration



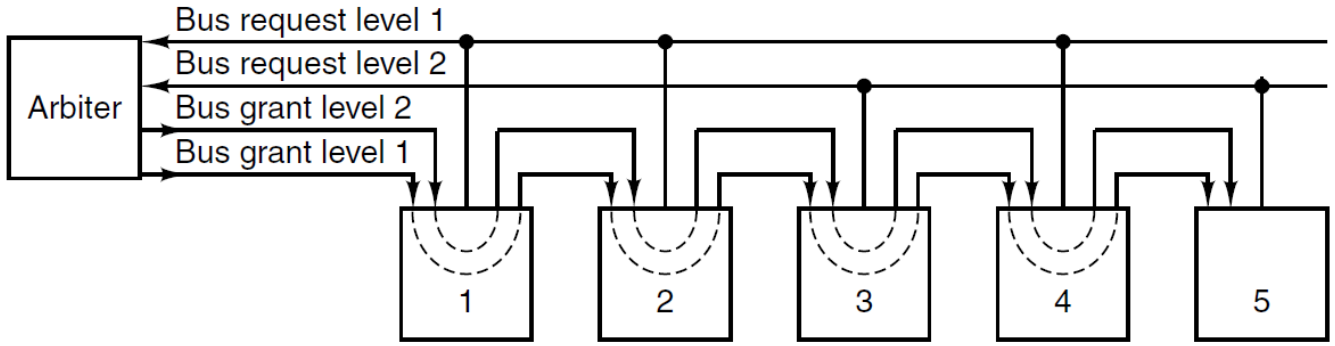
A centralized two-level bus arbiter using daisy chaining

1) Many buses have multiple priority levels. For each priority level there is a bus request line and a bus grant line
2) Each device attaches to one of the bus request levels, with more time-critical devices attaching to the higher-priority ones
3) Devices, 1, 2, and 4 use priority 1 while devices 3 and 5 use priority 2

Logic gates

CPU Chips and Buses

Centralized Bus Arbitration



A centralized two-level bus arbiter using daisy chaining

4) If multiple priority levels are requested at the same time, the arbiter issues a grant only on the highest-priority one. Among devices of the same priority, daisy chaining is used. In the event of conflicts, device 2 beats device 4, which beats 3. Device 5 has the lowest-priority because it is at the end of the lowest priority daisy chain

Logic gates

CPU Chips and Buses

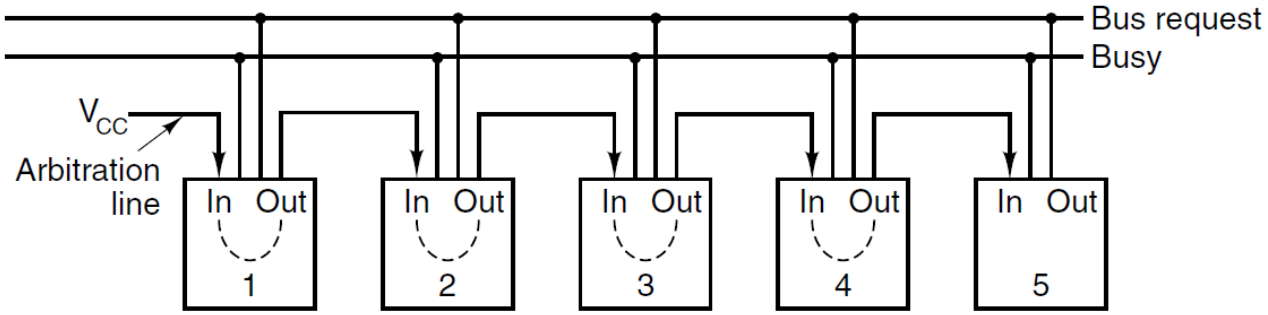
Decentralized Bus Arbitration

a) Eliminates bus arbiter
b) Each device on a separate request line
c) Each request line has a priority
d) All devices monitor all request lines
e) Requires a lot of bus lines
f) Number of devices limited to number of priority request lines in the bus

Logic gates

CPU Chips and Buses

Decentralized Bus Arbitration



Decentralized bus arbitration

a) Only need three lines, no matter how many devices are present
b) "busy" line asserted by current bus master
c) daisy-chained arbitration line wire to +Vcc
d) +Vcc is propagated through the devices until a device wants the bus
e) Device wanting bus, check "busy" line, and the incoming arbitration line if bus is idle and +Vcc on arbitration line, it negates the outgoing arbitration line, which propagates through all devices
f) Winner asserts "busy" and reasserts its outgoing arbitration line
g) Priority determined by "closeness" to +Vcc source

Logic gates

CPU Chips and Buses

Bus Operations

Various kinds of bus cycles:

1. A master (typically the CPU) reading from a slave (typically the memory) or writing to one.
2. Cashing or block reading When a block read is started, the bus master tells the slave how many words are to be transferred
3. On a multiprocessor system with two or more CPUs on the same bus, it is often necessary to make sure that only one CPU at a time uses some critical data structure in memory
4. Bus cycle used for handling interrupts

Logic gates

CPU Chips and Buses

Bus Operations

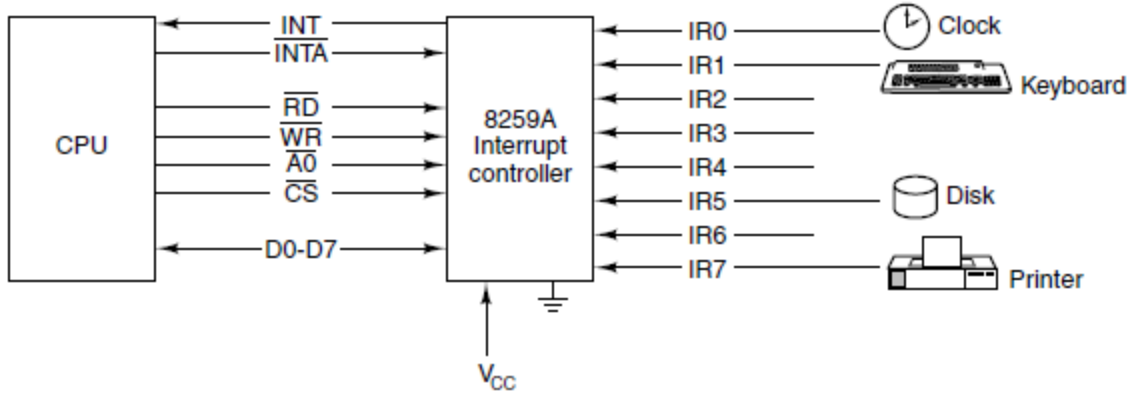


Figure 3-43. Use of the 8259A interrupt controller.

Multiple devices may want to cause an interrupt simultaneously. Solution is to assign priorities to devices and use a centralized arbiter to give priority to the most time-critical devices.

Logic gates

Example BUS

The **PCI bus:**

An internal synchronous bus for interconnecting chips, expansion boards, and processor/memory subsystems e.g. used for connecting adapters such as hard disks, sound cards, network cards and graphics cards.

Frequency and width:

32 bits and a frequency of 33 MHz bandwidths of up to 133 MBps

64 bits and a frequency of 66 MHz bandwidths of up to 1066 MBps

The PCI-X (eXtended) Bus

High-performance enhancement of the PCI

Backward compatible with the PCI bus

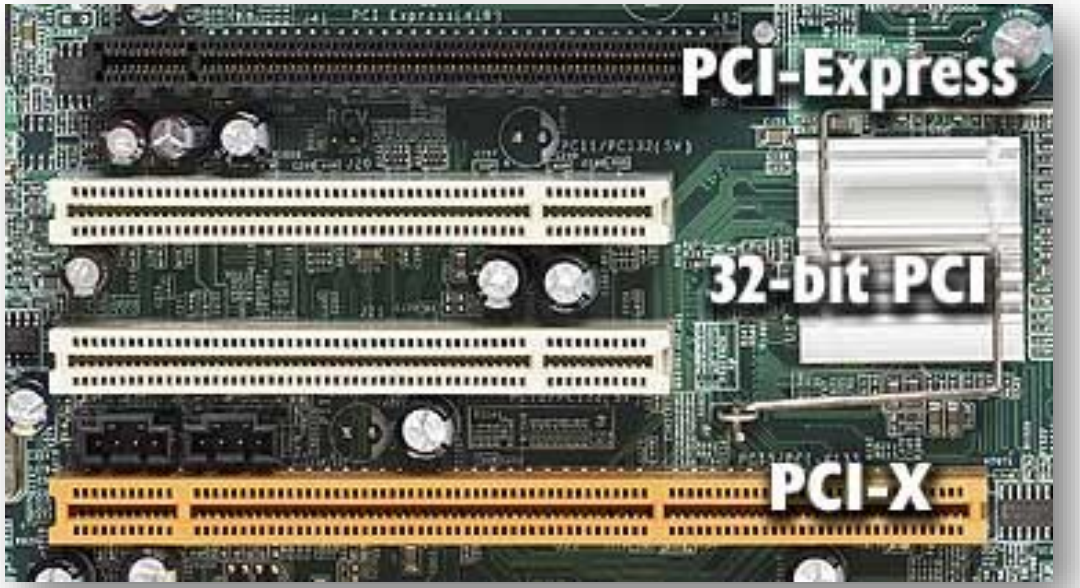
Frequency and bandwidth : 133 MHz and 1064 MBps

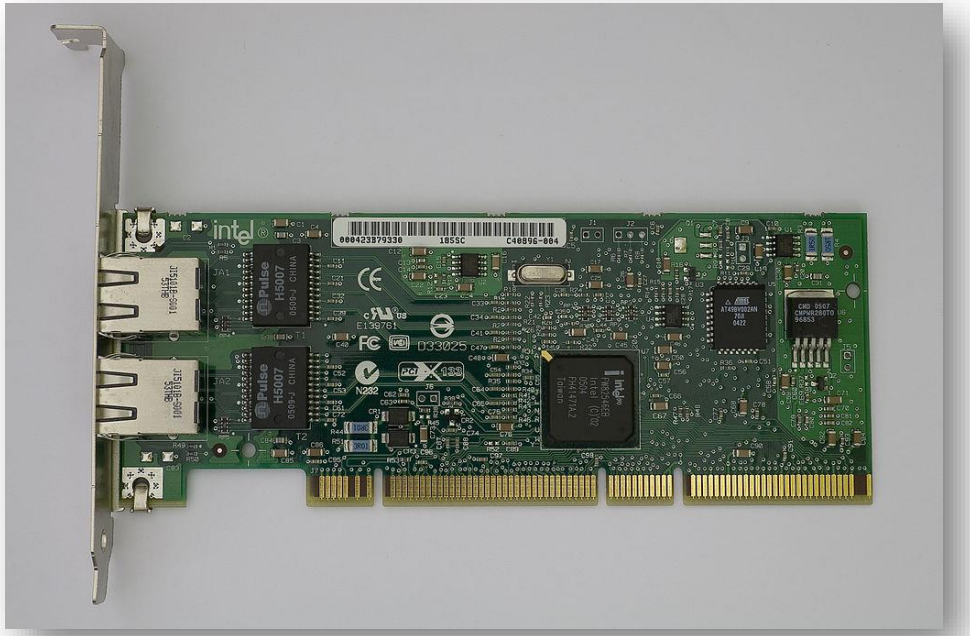
Width : 64 bits

The PCI-X 2.0 revision supports frequencies of 266 MHz and 533 MHz

Mostly Used in Servers

***PCI =** peripheral component interconnect

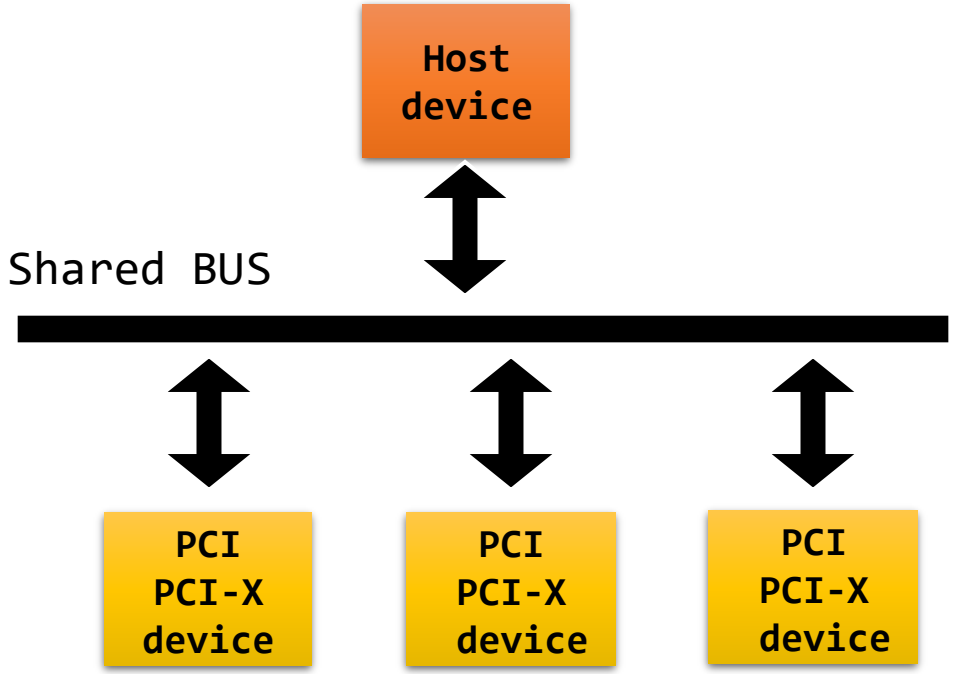


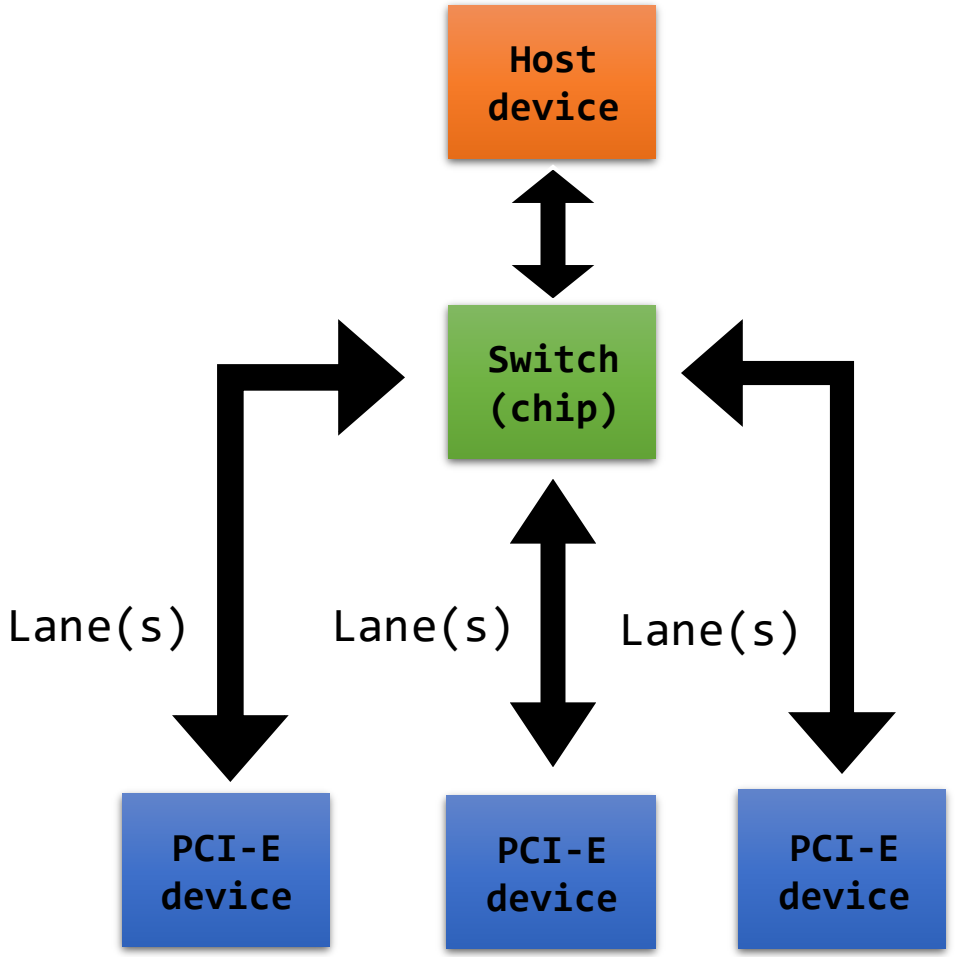


Dual Port Network Card for single PCI-X slot

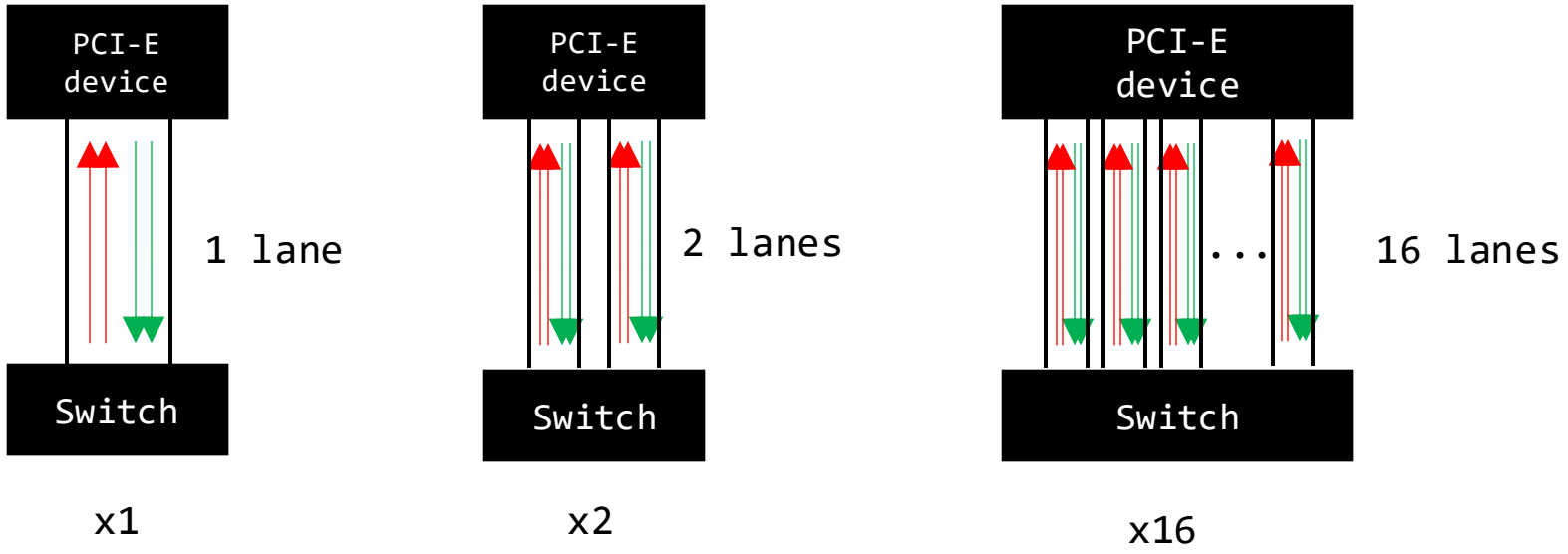
The PCI-Express Bus

Faster than PCI and PCI-X and ideal for video and graphics applications





Each lane (1x) has conductors for sending and receiving data (red and green colored arrows)



Bandwidth: 4 Gbps per lane

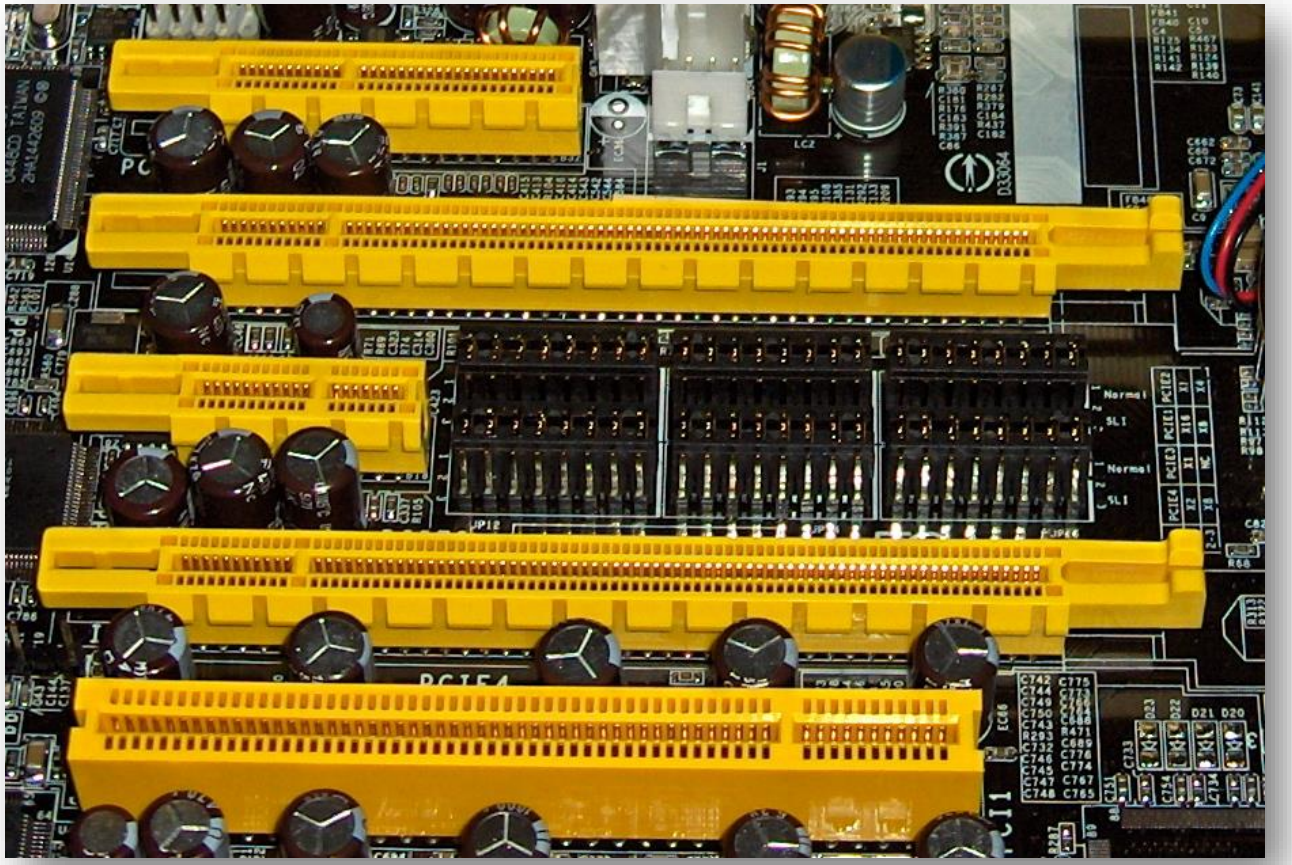
PCI Express x4

PCI Express x16

PCI Express x1

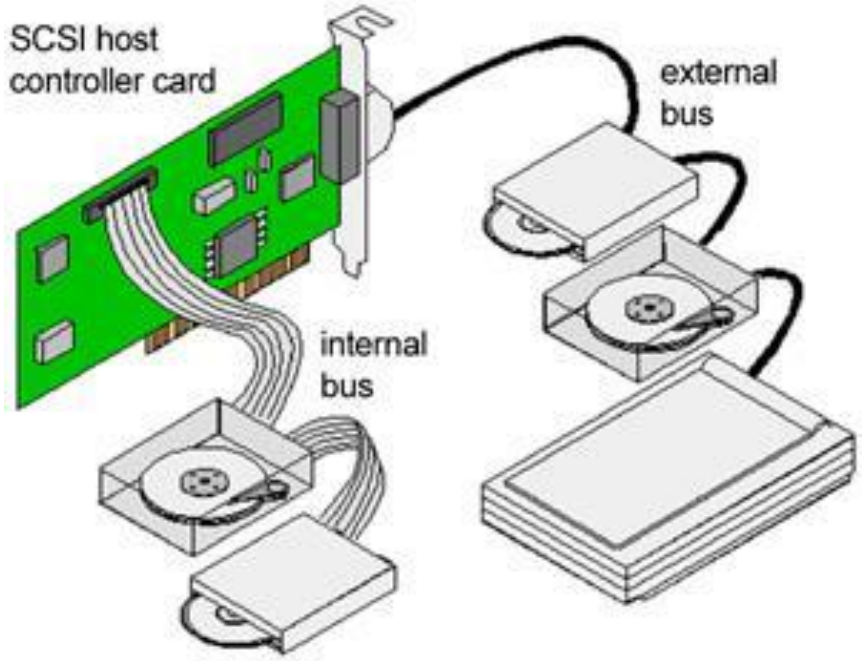
PCI Express x16

PCI



SCSI Bus

Versions	Variations	Maximum Target Devices Connected	Bus Width	Data Transfer Rates
SCSI-1	Asynchronous/Synchronous	7	8 bits	4 MBps/5 MBps
SCSI-2	Wide, Fast, Fast/Wide	7/15	8/16 bits	10 MBps/20 MBps
SCSI-3	Ultra, Ultra/Wide, Ultra2	7/15	8/16 bits	20 MBps/40 MBps
	Ultra2/Wide, Ultra 160			80 MBps/160 MBps
SCSI-4	Ultra320	15	16 bits	320 MBps
SCSI-5	Ultra640	15	8/16/32 bits	640 MBps



USB Bus

Some of the goals of the companies that originally conceived of the USB and started the project were as follows:

1. Users must not have to set switches or jumpers on boards or devices.
2. Users must not have to open the case to install new I/O devices.
3. There should be only one kind of cable to connect all devices.
4. I/O devices should get their power from the cable.
5. Up to 127 devices should be attachable to a single computer.
6. The system should support real-time devices (e.g., sound, telephone).
7. Devices should be installable while the computer is running.
8. No reboot should be needed after installing a new device.
9. The new bus and its I/O devices should be inexpensive to manufacture.

USB meets all these

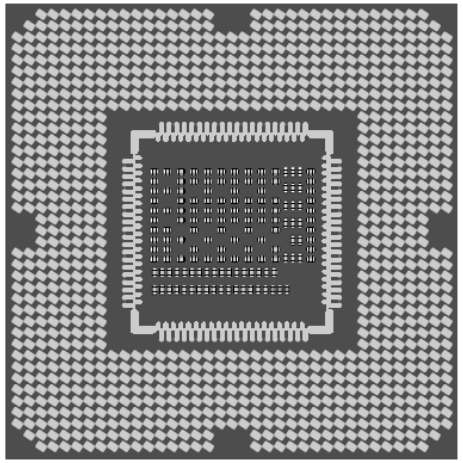
Versions	Low-Speed	Full-Speed	High Speed	Super-Speed
USB 1.0	✓	✓		
USB 1.1	✓	✓		
USB 2.0	✓	✓	✓	
USB 3.0	✓	✓	✓	✓

Data Rate	Max Value
Low-Speed	0.1875 MBps
Full-Speed	1.5 MBps
High Speed	60 MBps
Super-Speed	625 MBps

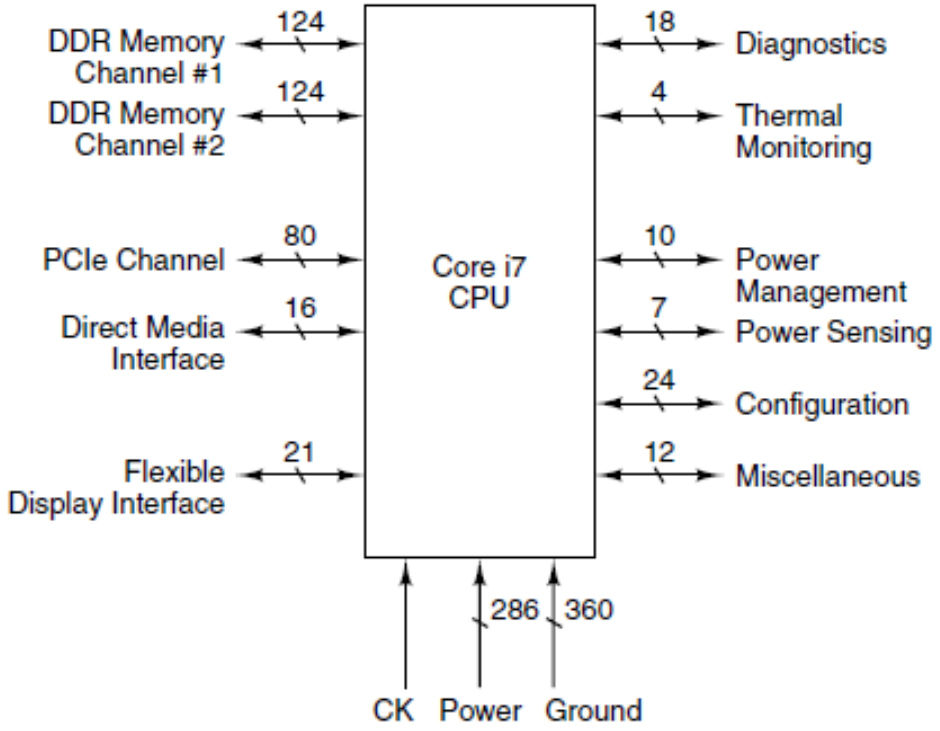
Logic gates

Example CPU Chips

The Intel Core i7



The Core i7 physical pinout.



Logical pinout of the Core i7.

Logic gates

Example CPU Chips

The Intel Core i7

The 1155 pins on the Core i7

A. 447 signals
B. 286 power connections (at several different voltages)
C. 360 grounds
D. 62 spares for future use

731-million transistor CPU running up to 3.2 GHz with a line width of 45 nanometers.

Three levels of cache. Each processor in a Core

A. L1 – 32-KB data and 32-KB instruction
B. L2 - 256-KB unified a mixture of instructions and data
C. L3 - All cores share a single level 3 unified cache, the size of which varies from 4 to 15 MB depending on the processor model