# TAKE AWAYS

- *CIRCUITS / SWITCHES*
- *CODING SCHEMES*
- *BINARY DIGITS*
- $2^n$
- *5 generations of computers*
- *Moore's Law*
- *Bits*
- *Bits, Bytes, KB, MB, GB, TB*
- *Machine Language*

# Computers Run On ELECTRICITY

# Electricity Has
# Two States: On & Off

# 1 Light
# How Many Messages
# can be stored / conveyed?

# Todd's Coding Scheme

| | |
|---|---|
| *on* | Come in |
| *off* | Go away |

# 2 Lights
# How Many Messages
# can be stored / conveyed?

# Todd's Coding Scheme

| | |
|---|---|
| *off* \| *off* | Go away |
| *off* \| *on* | Come in |
| *on* \| *off* | Bring pizza |
| *on* \| *on* | Bring beer |

# 3 Lights
# How Many Messages
# can be stored / conveyed?

# Todd's Coding Scheme

| | |
|---|---|
| *off* \| *off* \| *off* | Run away |
| *off* \| *off* \| *on* | Come in |
| *off* \| *on* \| *off* | Bring pizza |
| *on* \| *off* \| *off* | Bring beer |
| *on* \| *on* \| *off* | By land |
| *on* \| *off* \| *on* | By sea |
| *off* \| *on* \| *on* | Aerosmith |
| *on* \| *on* \| *on* | Jack Daniel's |

# Todd's Coding Scheme

| | |
|---|---|
| *off \| off \| off ( 0 \| 0 \| 0)* | Run away |
| *off \| off \| on ( 0 \| 0 \| 1)* | Come in |
| *off \| on \| off ( 0 \| 1 \| 0)* | Bring pizza |
| *on \| off \| off ( 1 \| 0 \| 0)* | Bring beer |
| *on \| on \| off (1 \| 1 \| 0)* | By land |
| *on \| off \| on (1 \| 0 \| 1)* | By sea |
| *off \| on \| on (0 \| 1 \| 1)* | Aerosmith |
| *on \| on \| on (1 \| 1 \| 1)* | Jack Daniel's |

# Todd's Coding Scheme

| | |
|---|---|
| *0 \| 0 \| 0* | Run away |
| *0 \| 0 \| 1* | Come in |
| *0 \| 1 \| 0* | Bring pizza |
| *1 \| 0 \| 0* | Bring beer |
| *1 \| 1 \| 0* | By land |
| *1 \| 0 \| 1* | By sea |
| *0 \| 1 \| 1* | Aerosmith |
| *1 \| 1 \| 1* | Jack Daniel's |

# Todd's Coding Scheme

| | |
|---|---|
| *0 0 0* | Run away |
| *0 0 1* | Come in |
| *0 1 0* | Bring pizza |
| *1 0 0* | Bring beer |
| *1 1 0* | By land |
| *1 0 1* | By sea |
| *0 1 1* | Aerosmith |
| *1 1 1* | Jack Daniel's |

# Todd's Coding Scheme

| | |
|---|---|
| *0 0 0* | A |
| *0 0 1* | B |
| *0 1 0* | C |
| *1 0 0* | D |
| *1 1 0* | E |
| *1 0 1* | F |
| *0 1 1* | G |
| *1 1 1* | H |

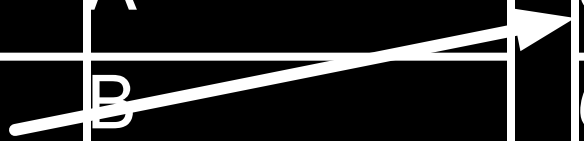
# Todd's Coding Scheme

| | |
|---|---|
| *0 0 0* | A |
| *0 0 1* | B |
| *0 1 0* | C |
| *1 0 0* | D |
| *1 1 0* | E |
| *1 0 1* | F |
| *0 1 1* | G |
| *1 1 1* | H |

| | |
|---|---|
| *0 0 1* | |
| *0 0 0* | |
| *1 0 0* | |
| *0 1 0* | |
| *0 0 0* | |
| *0 0 1* | |

# Todd's Coding Scheme

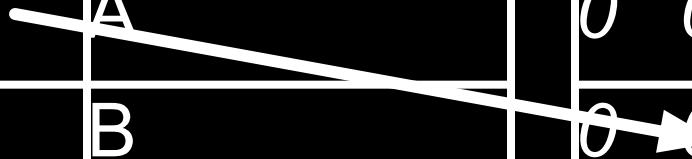
| | |
|---|---|
| *0 0 0* | A |
| *0 0 1* | B |
| *0 1 0* | C |
| *1 0 0* | D |
| *1 1 0* | E |
| *1 0 1* | F |
| *0 1 1* | G |
| *1 1 1* | H |



| | |
|---|---|
| *0 0 1* | B |
| *0 0 0* | |
| *1 0 0* | |
| *0 1 0* | |
| *0 0 0* | |
| *0 0 1* | |

# Todd's Coding Scheme

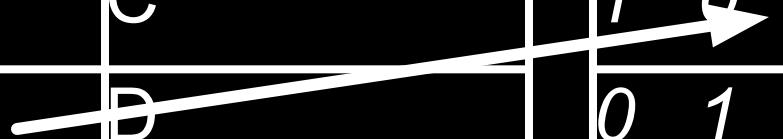
| | |
|---|---|
| *0 0 0* | A |
| *0 0 1* | B |
| *0 1 0* | C |
| *1 0 0* | D |
| *1 1 0* | E |
| *1 0 1* | F |
| *0 1 1* | G |
| *1 1 1* | H |



| | |
|---|---|
| *0 0 1* | B |
| *0 0 0* | A |
| *1 0 0* | |
| *0 1 0* | |
| *0 0 0* | |
| *0 0 1* | |

# Todd's Coding Scheme

| | |
|---|---|
| *0 0 0* | A |
| *0 0 1* | B |
| *0 1 0* | C |
| *1 0 0* | D |
| *1 1 0* | E |
| *1 0 1* | F |
| *0 1 1* | G |
| *1 1 1* | H |



| | |
|---|---|
| *0 0 1* | B |
| *0 0 0* | A |
| *1 0 0* | D |
| *0 1 0* | |
| *0 0 0* | |
| *0 0 1* | |

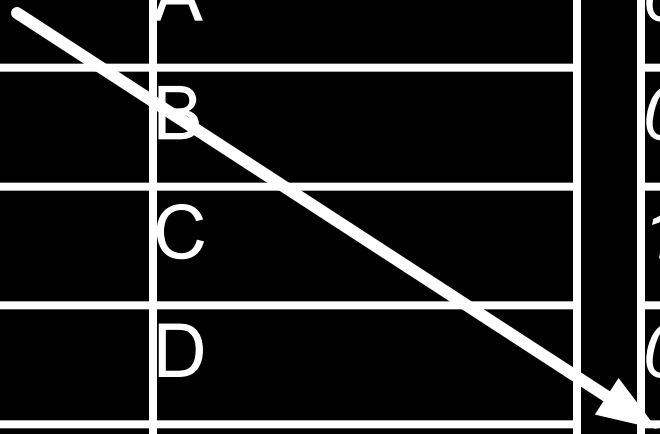
# Todd's Coding Scheme

| | |
|---|---|
| *0 0 0* | A |
| *0 0 1* | B |
| *0 1 0* | C |
| *1 0 0* | D |
| *1 1 0* | E |
| *1 0 1* | F |
| *0 1 1* | G |
| *1 1 1* | H |

| | |
|---|---|
| *0 0 1* | B |
| *0 0 0* | A |
| *1 0 0* | D |
| *0 1 0* | C |
| *0 0 0* | |
| *0 0 1* | |

# Todd's Coding Scheme

| | |
|---|---|
| *0 0 0* | A |
| *0 0 1* | B |
| *0 1 0* | C |
| *1 0 0* | D |
| *1 1 0* | E |
| *1 0 1* | F |
| *0 1 1* | G |
| *1 1 1* | H |



| | |
|---|---|
| *0 0 1* | B |
| *0 0 0* | A |
| *1 0 0* | D |
| *0 1 0* | C |
| *0 0 0* | A |
| *0 0 1* | |

# Todd's Coding Scheme

| | |
|---|---|
| *0 0 0* | A |
| *0 0 1* | B |
| *0 1 0* | C |
| *1 0 0* | D |
| *1 1 0* | E |
| *1 0 1* | F |
| *0 1 1* | G |
| *1 1 1* | H |

| | |
|---|---|
| *0 0 1* | B |
| *0 0 0* | A |
| *1 0 0* | D |
| *0 1 0* | C |
| *0 0 0* | A |
| *0 0 1* | B |

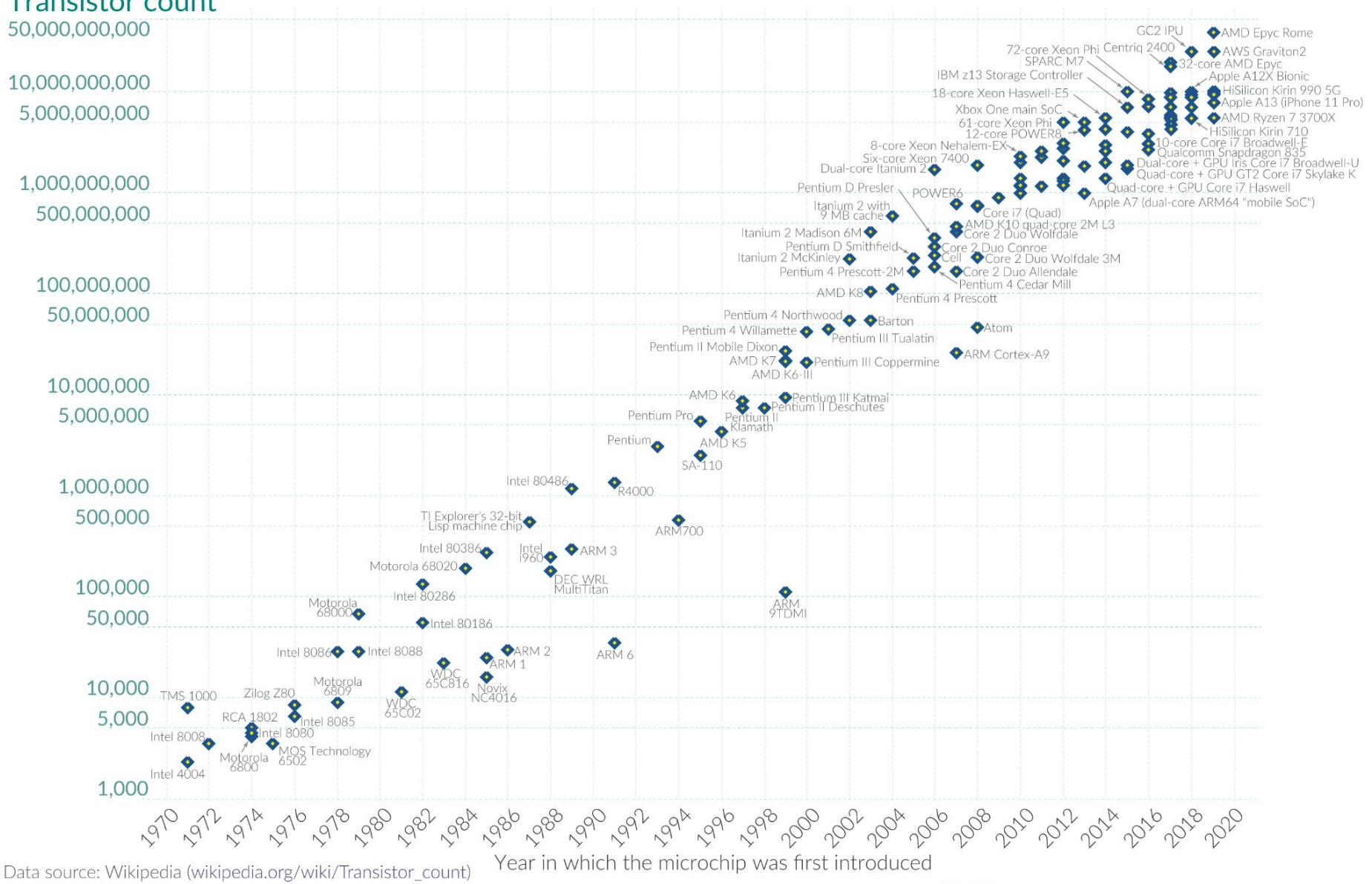
# Generations of Computers

1. *Vacuum tubes*
2. *Transistors*
3. *Integrated circuits (chips)*
4. *Microprocessors (cpu's)*
5. *AI (?)*

# Moore's Law: The number of transistors on microchips doubles every two years

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

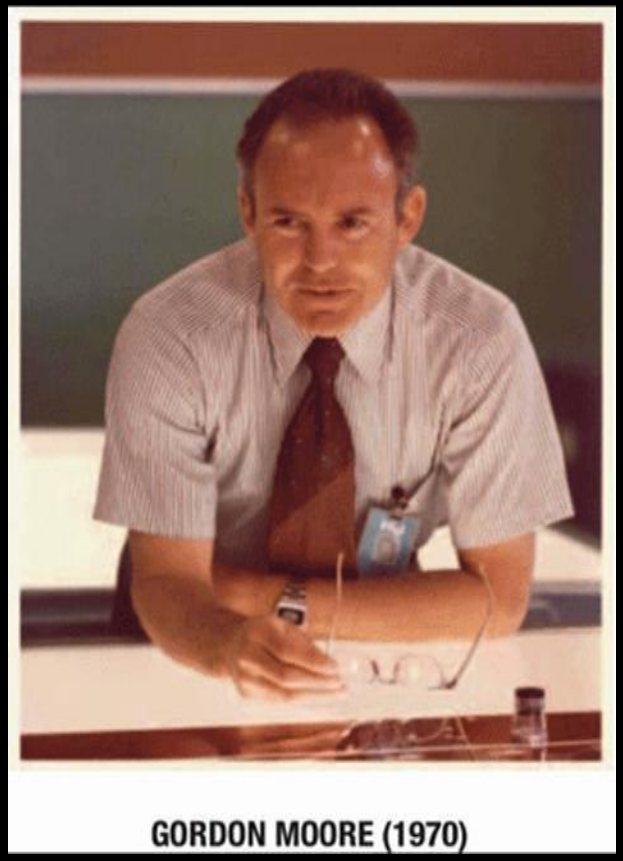



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

**Moore's Law**

**18 - 20 months**

**processing power doubles**



GORDON MOORE (1970)

# *Terminology Disambiguation - Circuits, Switches, Transistors, and Gates*

*circuits, switches, transistors, and even "gates" are all words used to refer to this thing within a computer that can either be ON or OFF. It's a circuit, it's a switch, it's a gate that can either be OPENED or CLOSED, it's a transistor - you will learn that people use all of those words to talk about this same thing, this ability of computers to store ON / OFF states.*

| | | |
|---|---|---|
| 1 light | $2^1$ | Represent 2 things |
| 2 light | $2^2$ | Represent 4 things |
| 3 light | $2^3$ | Represent 8 things |
| 4 light | $2^4$ | Represent 16 things |
| 5 light | $2^5$ | Represent 32 things |
| 6 light | $2^6$ | Represent 64 things |
| 7 light | $2^7$ | Represent 128 things |
| 8 light | $2^8$ | Represent 256 things |

| | | |
|---|---|---|
| 1 light | $2^1$ | Represent 2 things |
| 2 light | $2^2$ | Represent 4 things |
| 3 light | $2^3$ | Represent 8 things |
| 4 light | $2^4$ | Represent 16 things |
| 5 light | $2^5$ | Represent 32 things |
| 6 light | $2^6$ | Represent 64 things |
| 7 light | $2^7$ | Represent 128 things |
| 8 light | $2^8$ | Represent 256 things |

$2^n$

# 16 Lights
# How Many Messages
# can be stored / conveyed?

$$2^{16} = 65{,}536$$

# 32 Lights
# How Many Messages
# can be stored / conveyed?

$$2^{32} = 4,294,967,296$$

# Bits = Binary Digits

**Bits = Binary Digits**

# Measuring Bits

| | |
|---|---|
| 1 bit | |
| 8 bits | 1 byte |
| 1024 bytes | 1 KiloByte |
| 1024 KiloBytes | 1 MegaByte |
| 1024 MegaBytes | 1 GigaByte |
| 1024 GigaBytes | 1 TeraByte |

| | |
|---|---|
| 1 bit | |
| 8 bits | 1 byte |
| 1000 bytes | 1 KiloByte |
| 1000 KiloBytes | 1 MegaByte |
| 1000 MegaBytes | 1 GigaByte |
| 1000 GigaBytes | 1 TeraByte |

| |
|---|
| 1 bit |
| 8 bits |
| 8,000 bits |
| 8,000,000 bits |
| 8,000,000,000 bits |
| 8,000,000,000,000 bits |

# How many bits are in 2 bytes?

| | |
|---|---|
| 1 bit | |
| 8 bits | 1 byte |
| 1000 bytes | 1 KiloByte |
| 1000 KiloBytes | 1 MegaByte |
| 1000 MegaBytes | 1 GigaByte |
| 1000 GigaBytes | 1 TeraByte |

| |
|---|
| 1 bit |
| 8 bits |
| 8,000 bits |
| 8,000,000 bits |
| 8,000,000,000 bits |
| 8,000,000,000,000 bits |

# How many bits are in 2 kilobytes?

| | |
|---|---|
| 1 bit | |
| 8 bits | 1 byte |
| 1000 bytes | 1 KiloByte |
| 1000 KiloBytes | 1 MegaByte |
| 1000 MegaBytes | 1 GigaByte |
| 1000 GigaBytes | 1 TeraByte |

| |
|---|
| 1 bit |
| 8 bits |
| 8,000 bits |
| 8,000,000 bits |
| 8,000,000,000 bits |
| 8,000,000,000,000 bits |

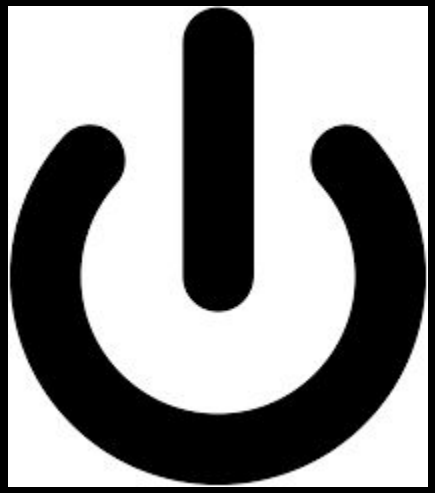
# How many bits are in 2 TB?

*Terminology Disambiguation*

*ON & OFF, 1 & 0, Binary Digits, Bits, Machine Language*

*ON & OFF, 1 & 0, Binary Digits, Bits, and Machine Language are all words used to refer to this idea that, within a computer, it's all nothing but a bunch of ZERO's and ONE's, or switches that are ON or OFF, it's all just a bunch of Binary Digits, or Bits, that's the language which computers speak, it's machine language.*

*The POWER SYMBOL on many devices is a combination of ZERO and ONE, where ONE means ON, and ZERO means OFF.*



# Machine Language

*All programs are written in a programming language and then translated to machine language.*

**I**nput **P**rocess **O**utput **S**torage

# IPOS

# I P O S

(processing & memory)

2 +

2 + 3

$$2 + 3 =$$

# Registers

# Cache

# RAM
# (memory)


C300515-001
REV A
256X3X3

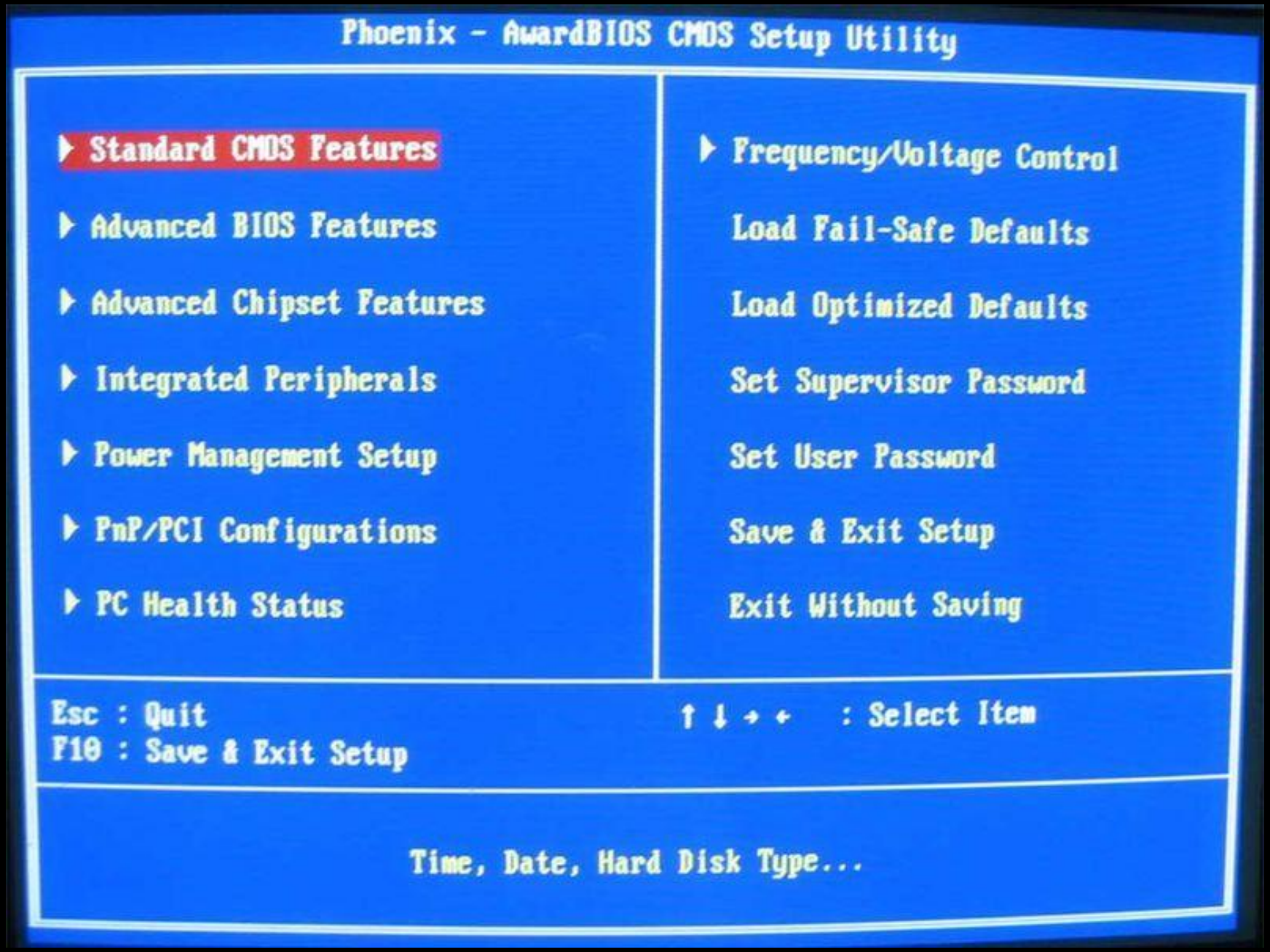

FPM

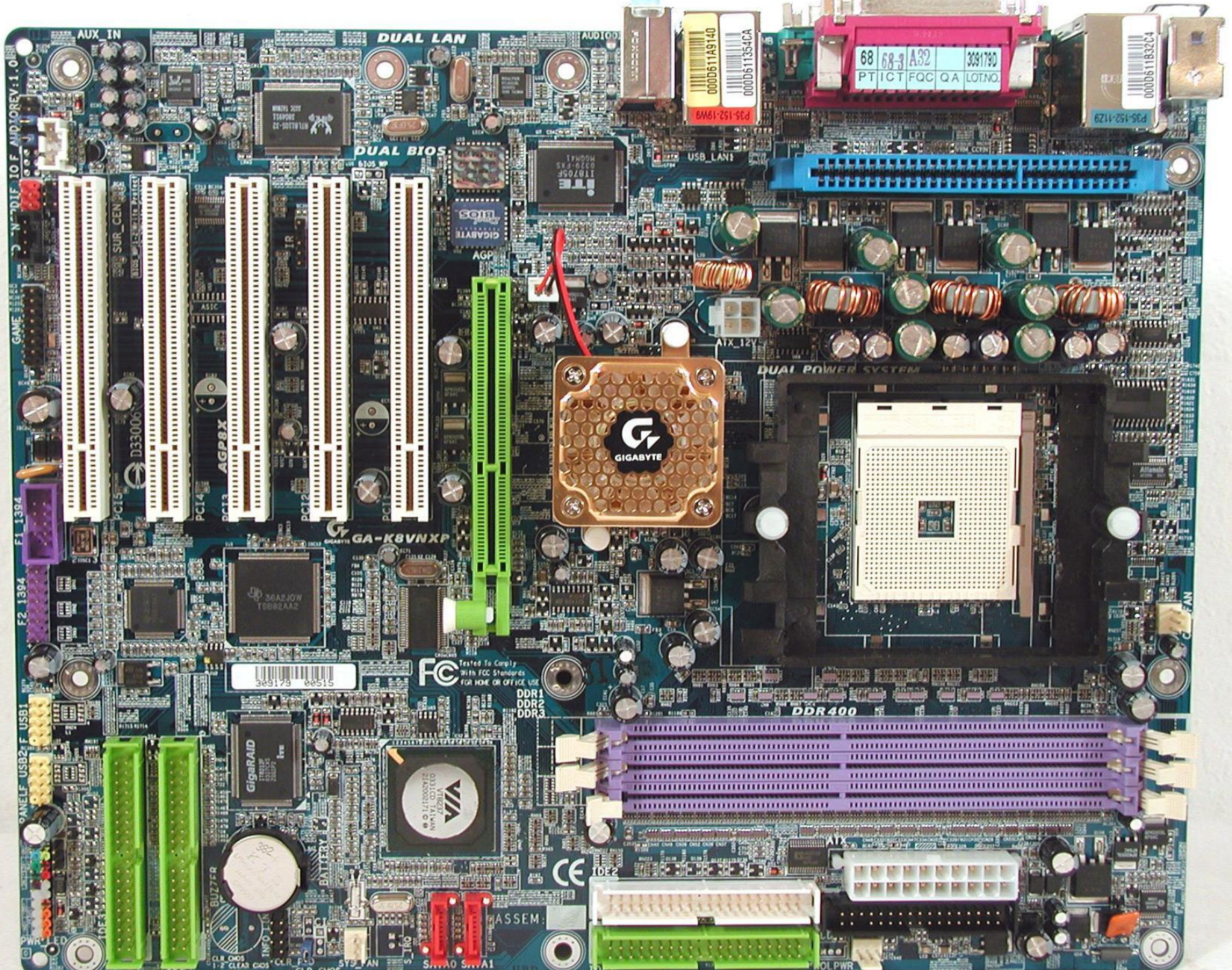

4X64


NCP
NC5942
256MB
PC2700
DDR
W E
256MB
333DDR

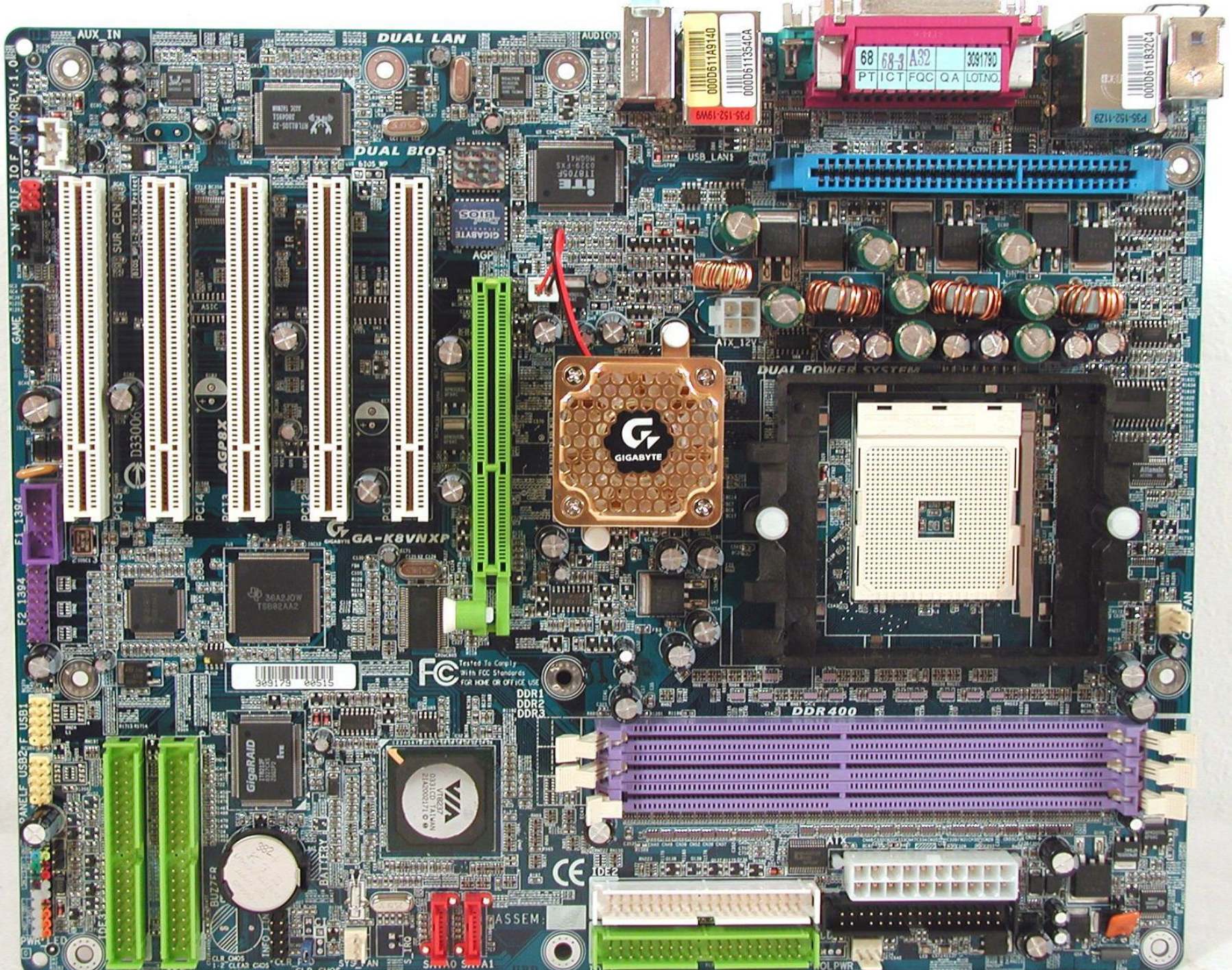
# ROM

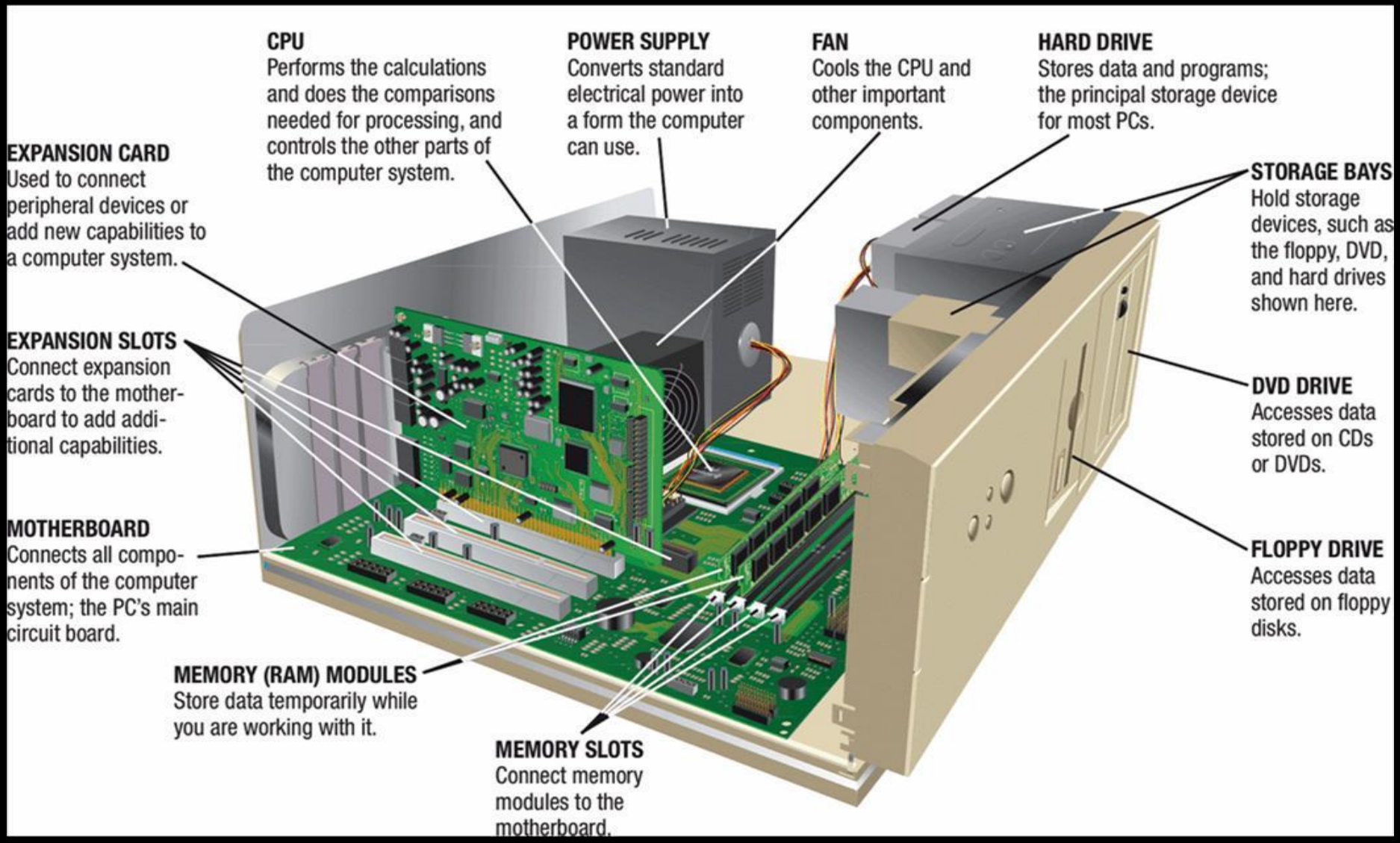
# Startup Memory (BIOS)


Phoenix - AwardBIOS CMOS Setup Utility
Standard CMOS Features
Advanced BIOS Features
Advanced Chipset Features
Integrated Peripherals
Power Management Setup
PnP/PCI Configurations
PC Health Status
Frequency/Voltage Control
Load Fail-Safe Defaults
Load Optimized Defaults
Set Supervisor Password
Set User Password
Save & Exit Setup
Exit Without Saving
Esc : Quit
F10 : Save & Exit Setup
↑ ↓ → ← : Select Item
Time, Date, Hard Disk Type...


AUX_IN
DUAL LAN
AUDIO
DUAL BIOS
USB_LAN
LPT
AGP
ATX_12V
DUAL POWER SYSTEM
AGP8X
GA-K8VNXP
GIGABYTE
Tested To Comply
With FCC Standards
FOR HOME OR OFFICE USE
DDR1
DDR2
DDR3
DDR400
GigaRAID
IDE2
ATX
BUZZER
BATTERY
CLR_CMOS
SYS_FAN
F_USB1
F_USB2
F1_1394
F2_1394
GAME
IR
SUR_CEN
PCI1
PCI2
PCI3
PCI4
PCI5

# Motherboard


AUX_IN
DUAL LAN
AUDIO
DUAL BIOS
USB_LAN
LPT
AGP
ATX_12V
DUAL POWER SYSTEM
AGP8X
GA-K8VNXP
GIGABYTE
Tested To Comply
With FCC Standards
FOR HOME OR OFFICE USE
DDR1
DDR2
DDR3
DDR400
GigaRAID
IDE2
ATX
BUZZER
BATTERY
CLR_CMOS
SYS_FAN
F_USB1
F_USB2
F1_1394
F2_1394
GAME
IR
SUR_CEN
PCI1
PCI2
PCI3
PCI4
PCI5


CPU
Performs the calculations and does the comparisons needed for processing, and controls the other parts of the computer system.
POWER SUPPLY
Converts standard electrical power into a form the computer can use.
FAN
Cools the CPU and other important components.
HARD DRIVE
Stores data and programs; the principal storage device for most PCs.
EXPANSION CARD
Used to connect peripheral devices or add new capabilities to a computer system.
STORAGE BAYS
Hold storage devices, such as the floppy, DVD, and hard drives shown here.
EXPANSION SLOTS
Connect expansion cards to the motherboard to add additional capabilities.
DVD DRIVE
Accesses data stored on CDs or DVDs.
MOTHERBOARD
Connects all components of the computer system; the PC's main circuit board.
FLOPPY DRIVE
Accesses data stored on floppy disks.
MEMORY (RAM) MODULES
Store data temporarily while you are working with it.
MEMORY SLOTS
Connect memory modules to the motherboard.

# Bus

# AGP / USB / Thunderbolt

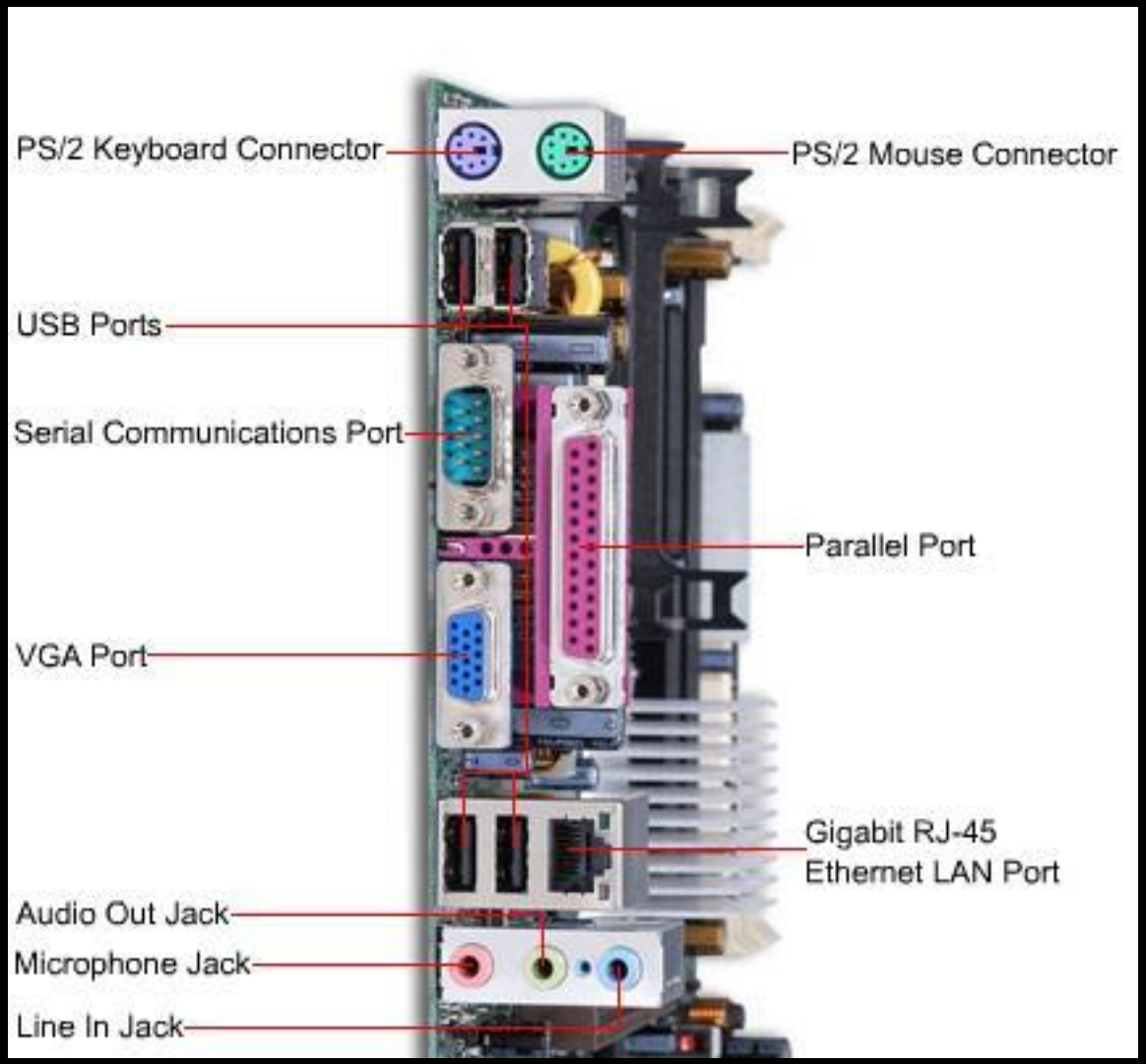
**BusWidth + Speed = Bandwidth (throughput)**

# Expansion slots & cards


REV: 1.0
GV-R96X128D
GIGABYTE


BELKIN
USB2
2.0 2-Port
CardBus
BELKIN

# Ports & Connectors


PS/2 Keyboard Connector
PS/2 Mouse Connector
USB Ports
Serial Communications Port
Parallel Port
VGA Port
Gigabit RJ-45
Ethernet LAN Port
Audio Out Jack
Microphone Jack
Line In Jack

**MOUSE PORT**
Used to connect a mouse.

**POWER CONNECTOR**
Connects PC to a power outlet.

**KEYBOARD PORT**
Used to connect a keyboard.

**SERIAL PORT**
Usually used for a scanner or mouse.

**USB PORTS**
Used to connect a keyboard, mouse, scanner, flash memory drive, or other USB devices.

**PARALLEL PORT**
Usually used for a printer.

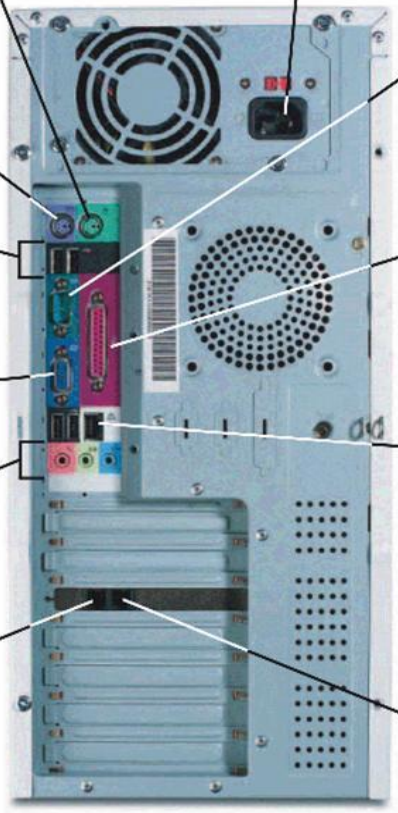
**MONITOR PORT**
Used to connect a monitor.

**NETWORK PORT**
Used to connect the PC to a network.

**SOUND PORTS**
Used to connect speakers, headphones, and a microphone.

**PHONE PORT**
Used to connect a telephone so you don't lose the use of your phone jack.

**MODEM PORT**
Used to connect the PC to a phone jack.



## CONNECTORS


Power plug


USB plug


FireWire plug


PS/2 plug for mouse or keyboard


Serial plug


Monitor plug


Parallel plug


Telephone plug for modem and telephone


Network (RJ-45) plug


Network (Fiber-optic) plug