****

**MSPM’S**

**Deogiri Institute of Engineering and Management Studies, Aurangabad**

Report on

**Computer Architecture & organization**

Submitted By

**Vishal Gaikwad**

Under the Guidance of

**Prof. Pankaj Durole**

Department of Basic Sciences and Humanities

(Deogiri Institute of Engineering and Management Studies)

2018-2019



CERTIFICATE

This is to certify that Mr. **Vishal Gaikwad** Seat No.26119 has completed report writing on Computer Architecture & organization.

**Pankaj Durole**  **Dr.S.Kalyankar Dr.Ulhas Shiukar**

**Subject Teacher H.O.D. Director**



# Lenovo Legion Y540

# Laptop Information

|  |  |
| --- | --- |
| **Brand** | Lenovo |
| **Series** | Legion Y540 |
| **Colour** | BLACK |
| **Item Height** | 26 Millimeters |
| **Item Width** | 26 Centimeters |
| **Screen Size** | 15.6 Inches |
| **Maximum Display Resolution** | 1920 x 1080 (Full HD) |
| **Item Weight** | 2.3 Kg |
| **Product Dimensions** | 36.5 x 26 x 2.6 cm |
| **Batteries:** | 1 Lithium Polymer batteries required. (included) |
| **Item model number** | 81SY00C7IN |
| **Processor Brand** | Intel |
| **Processor Type** | Core i7 |
| **Memory Technology** | DDR4 |
| **Hard Disk Technology** | Flash Memory Solid State |
| **Speaker Description** | Harman speakers with Dolby Atmos |
| **Graphics Coprocessor** | NVIDIA GEFORCE GTX 1660ti (6GB GDDR5) |
| **Graphics Card Ram Size** | 6 GB |
| **Number of USB 3.0 Ports** | 3 |
| **Number of HDMI Ports** | 1 |
| **Number of Audio-out Ports** | 1 |
| **Number of Ethernet Ports** | 1 |
| **Number of Microphone Ports** | 1 |
| **Optical Drive Type** | None |
| **Operating System** | Windows 10 Home |
| **Lithium Battery Energy Content** | 52.5 Watt Hours |
| **Lithium battery Weight** | 100 Grams |
| **Number of Lithium Ion Cells** | 3 |
| **Included Components** | Laptop, Adapter, User Manual |

**RAM**

Double Data Rate 4 Synchronous Dynamic Random-Access Memory, officially abbreviated as **DDR4 SDRAM**, is a type of [synchronous dynamic random-access memory](https://en.wikipedia.org/wiki/Synchronous_dynamic_random-access_memory) with a high [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(computing)) ("[double data rate](https://en.wikipedia.org/wiki/Double_data_rate)") interface.

Released to the market in 2014, it is one of the latest variants of [dynamic random-access memory](https://en.wikipedia.org/wiki/Dynamic_random-access_memory) (DRAM), of which some have been in use since the early 1970s, and a higher-speed successor to the [DDR2](https://en.wikipedia.org/wiki/DDR2_SDRAM) and [DDR3](https://en.wikipedia.org/wiki/DDR3_SDRAM) technologies.

DDR4 is not compatible with any earlier type of random-access memory (RAM) due to different signaling voltage and physical interface, besides other factors.

DDR4 SDRAM was released to the public market in Q2 2014, focusing on [ECC memory](https://en.wikipedia.org/wiki/ECC_memory), while the non-ECC DDR4 modules became available in Q3 2014, accompanying the launch of [Has well-E](https://en.wikipedia.org/wiki/Haswell-E) processors that require DDR4 memory.

## Features

The primary advantages of DDR4 over its predecessor, DDR3, include higher module density and lower voltage requirements, coupled with higher [data rate transfer](https://en.wikipedia.org/wiki/Bit_rate#Goodput_(data_transfer_rate)) speeds. The DDR4 standard allows for [DIMMs](https://en.wikipedia.org/wiki/DIMM) of up to 64 [GB](https://en.wikipedia.org/wiki/Gibibyte) in capacity, compared to DDR3's maximum of 16 GB per DIMM.

Unlike previous generations of DDR memory, [prefetch](https://en.wikipedia.org/wiki/Prefetch_buffer" \o "Prefetch buffer) has *not* been increased above the 8n used in DDR3 the basic burst size is eight words, and higher bandwidths are achieved by sending more read/write commands per second. To allow this, the standard divides the DRAM banks into two or four selectable bank groups, where transfers to different bank groups may be done more rapidly.

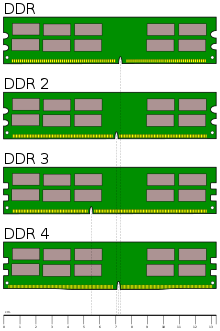
Because power consumption increases with speed, the reduced voltage allows higher speed operation without unreasonable power and cooling requirements.

DDR4 operates at a voltage between 1.2 V and 1.4 V with a frequency between 800 and 2133 MHz (DDR4-1600 through DDR4-4266), compared to frequencies between 400 and 1067 MHz and voltage requirements of 1.5 or 1.65 V of DDR3. Due to the nature of DDR, speeds are typically advertised as doubles of these numbers (DDR3-1600 and DDR4-2400 are common, with DDR4-3200 and DDR4-4800 available at high cost). Although a low-voltage standard has yet to be finalized (as of August 2014), it is anticipated that low-voltage DDR4 will run at a voltage of 1.05 V, compared to DDR3's low-voltage standard ([DDR3L](https://en.wikipedia.org/wiki/DDR3L)) which requires 1.35 V to operate.

## Timeline

[](https://en.wikipedia.org/wiki/File:Samsung_displays_first_DDR4_module.jpg)

The first DDR4 memory module prototype was manufactured by [Samsung](https://en.wikipedia.org/wiki/Samsung) and announced in January 2011.

[](https://en.wikipedia.org/wiki/File:Desktop_DDR_Memory_Comparison.svg)

* **2005:** standards body [JEDEC](https://en.wikipedia.org/wiki/JEDEC) began working on a successor to DDR3 around 2005, about 2 years before the launch of DDR3 in 2007. The high-level architecture of DDR4 was planned for completion in 2008.
* **2007:** some advance information was published in 2007, and a guest speaker from [Qimonda](https://en.wikipedia.org/wiki/Qimonda" \o "Qimonda) provided further public details in a presentation at the August 2008 [San Francisco](https://en.wikipedia.org/wiki/San_Francisco) [Intel Developer Forum](https://en.wikipedia.org/wiki/Intel_Developer_Forum) (IDF). DDR4 was described as involving a 30 nm process at 1.2 volts, with bus frequencies of 2133 [MT/s](https://en.wikipedia.org/wiki/MT/s) "regular" speed and 3200 MT/s "enthusiast" speed, and reaching market in 2012, before transitioning to 1 volt in 2013.
* **2009:** in February, [Samsung](https://en.wikipedia.org/wiki/Samsung) validated 40 nm DRAM chips, considered a "significant step" towards DDR4 development since in 2009, DRAM chips were only beginning to migrate to a 50 nm process.
* **2010:** subsequently, further details were revealed at MemCon 2010, [Tokyo](https://en.wikipedia.org/wiki/Tokyo) (a computer memory industry event), at which a presentation by a JEDEC director titled "Time to rethink DDR4" with a slide titled "New roadmap: More realistic roadmap is 2015" led some websites to report that the introduction of DDR4 was probably or definitely delayed until 2015. However, DDR4 [test samples](https://en.wikipedia.org/wiki/Engineering_sample)were announced in line with the original schedule in early 2011 at which time manufacturers began to advise that large scale commercial production and release to market was scheduled for 2012.
* **2011:** in January, [Samsung](https://en.wikipedia.org/wiki/Samsung) announced the completion and release for testing of a 2 GB DDR4 DRAM module based on a process between 30 and 39 [nm](https://en.wikipedia.org/wiki/Nanometer). It has a maximum data transfer rate of 2133 [MT/s](https://en.wikipedia.org/wiki/MT/s) at 1.2 V, uses [pseudo open drain](https://en.wikipedia.org/wiki/Open_drain) technology (adapted from [graphics DDR](https://en.wikipedia.org/wiki/GDDR) memory) and draws 40% less power than an equivalent DDR3 module.   
  In April, [Hynix](https://en.wikipedia.org/wiki/Hynix" \o "Hynix) announced the production of 2 GB DDR4 modules at 2400 MT/s, also running at 1.2 V on a process between 30 and 39 nm (exact process unspecified), adding that it anticipated commencing high volume production in the second half of 2012. Semiconductor processes for DDR4 are expected to transition to sub-30 nm at some point between late 2012 and 2014.
* **2012:** in May, [Micron](https://en.wikipedia.org/wiki/Micron_Technology) announced it is aiming at starting production in late 2012 of 30 nm modules.  
  In July, Samsung announced that it would begin sampling the industry's first 16 GB registered dual inline memory modules (RDIMMs) using DDR4 SDRAM for enterprise server systems.  
  In September, JEDEC released the final specification of DDR4.
* **2013:** DDR4 was expected to represent 5% of the DRAM market in 2013, and to reach [mass market](https://en.wikipedia.org/wiki/Mass_market) adoption and 50% [market penetration](https://en.wikipedia.org/wiki/Market_penetration) around 2015 as of 2013, however, adoption of DDR4 has been delayed and it is no longer expected to reach a majority of the market until 2016 or later. The transition from DDR3 to DDR4 is thus taking longer than the approximately five years taken for DDR3 to achieve mass market transition over DDR2. In part, this is because changes required to other components would affect all other parts of computer systems, which would need to be updated to work with DDR4.
* **2014:** in April, Hynix announced that it had developed the world's first highest-density 128 GB module based on 8 [Gb](https://en.wikipedia.org/wiki/Gibibit" \o "Gibibit) DDR4 using 20 nm technology. The module works at 2133 MHz, with a 64-bit I/O, and processes up to 17 GB of data per second.
* **2016:** in April, Samsung announced that they had begun to mass-produce DRAM on a "10 nm-class" process, by which they mean the 1x nm node regime of 16 nm to 19 nm, which supports a 30% faster data transfer rate of 3,200 [megabits](https://en.wikipedia.org/wiki/Megabit) per second. Previously, a size of 20 nm was used.

### Market perception and adoption

In April 2013, a news writer at [International Data Group](https://en.wikipedia.org/wiki/International_Data_Group) (IDG)‍—‌an American technology research business originally part of [IDC](https://en.wikipedia.org/wiki/International_Data_Corporation)‍—‌produced an analysis of their perceptions related to DDR4 SDRAM. The conclusions were that the increasing popularity of [mobile computing](https://en.wikipedia.org/wiki/Mobile_computing) and other devices using slower but low-powered memory, the slowing of growth in the traditional [desktop computing](https://en.wikipedia.org/wiki/Desktop_computing) sector, and the [consolidation](https://en.wikipedia.org/wiki/Consolidation_(economics)) of the memory manufacturing marketplace, meant that margins on RAM were tight.

As a result, the desired [premium pricing](https://en.wikipedia.org/wiki/Premium_pricing) for the new technology was harder to achieve, and capacity had shifted to other sectors. SDRAM manufacturers and chipset creators were, to an extent, "[stuck between a rock and a hard place](https://en.wikipedia.org/wiki/Stuck_between_a_rock_and_a_hard_place)" where "nobody wants to pay a premium for DDR4 products, and manufacturers don't want to make the memory if they are not going to get a premium", according to Mike Howard from iSuppli. A switch in [market sentiment](https://en.wikipedia.org/wiki/Market_sentiment) toward desktop computing and release of processors having DDR4 support by [Intel](https://en.wikipedia.org/wiki/Intel) and [AMD](https://en.wikipedia.org/wiki/AMD) could therefore potentially lead to "aggressive" growth.[[41]](https://en.wikipedia.org/wiki/DDR4_SDRAM#cite_note-IDG_2013-43)

Intel's 2014 [Haswell](https://en.wikipedia.org/wiki/Haswell_(microarchitecture)" \o "Haswell (microarchitecture)) roadmap revealed the company's first use of DDR4 SDRAM in [Haswell-EP](https://en.wikipedia.org/wiki/Haswell-EP" \o "Haswell-EP) processors.

AMD's [Ryzen](https://en.wikipedia.org/wiki/Ryzen" \o "Ryzen) processors, revealed in 2016 and shipped in 2017, use DDR4 SDRAM.

## Operation

DDR4 chips use a 1.2 [V](https://en.wikipedia.org/wiki/Volt) supply with a 2.5 V auxiliary supply for wordline boost called VPP, as compared with the standard 1.5 V of DDR3 chips, with lower voltage variants at 1.05 V appearing in 2013. DDR4 is expected to be introduced at transfer rates of 2133 MT/s, estimated to rise to a potential 4266 MT/sby 2013. The minimum transfer rate of 2133 MT/s was said to be due to progress made in DDR3 speeds which, being likely to reach 2133 MT/s, left little commercial benefit to specifying DDR4 below this speed. Techgage interpreted Samsung's January 2011 engineering sample as having [CAS latency](https://en.wikipedia.org/wiki/CAS_latency) of 13 clock cycles, described as being comparable to the move from DDR2 to DDR3.

Internal banks are increased to 16 (4 bank select bits), with up to 8 ranks per DIMM.

Protocol changes include.

* Parity on the command/address bus
* Data bus inversion (like [GDDR4](https://en.wikipedia.org/wiki/GDDR4))
* [CRC](https://en.wikipedia.org/wiki/Cyclic_redundancy_check) on the data bus
* Independent programming of individual DRAMs on a DIMM, to allow better control of [on-die termination](https://en.wikipedia.org/wiki/On-die_termination).

Increased memory density is anticipated, possibly using TSV ("[through-silicon via](https://en.wikipedia.org/wiki/Through-silicon_via)") or other [3D stacking processes](https://en.wikipedia.org/wiki/Three-dimensional_integrated_circuit). The DDR4 specification will include standardized [3D stacking](https://en.wikipedia.org/wiki/Three-dimensional_integrated_circuit) "from the start" according to JEDEC,  with provision for up to 8 stacked dies. X-bit Labs predicted that "as a result DDR4 memory chips with very high density will become relatively inexpensive".

Switched memory banks are also an anticipated option for servers.

In 2008 concerns were raised in the book *Wafer Level 3-D ICs Process Technology* that [non-scaling](https://en.wikipedia.org/wiki/Semiconductor_device_fabrication) analog elements such as [charge pumps](https://en.wikipedia.org/wiki/Charge_pump) and [voltage regulators](https://en.wikipedia.org/wiki/Bandgap_voltage_reference), and additional circuitry "have allowed significant increases in [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(signal_processing)) but they consume much more [die area](https://en.wikipedia.org/wiki/Silicon_die)". Examples include [CRC](https://en.wikipedia.org/wiki/Cyclic_redundancy_check) error-detection, [on-die termination](https://en.wikipedia.org/wiki/On-die_termination), burst hardware, programmable pipelines, low [impedance](https://en.wikipedia.org/wiki/Electrical_impedance), and increasing need for [sense amps](https://en.wikipedia.org/wiki/Current_sense_amplifier) (attributed to a decline in bits per bitline due to low voltage). The authors noted that, as a result, the amount of die used for the memory array itself has declined over time from 70–78% with SDRAM and DDR1, to 47% for DDR2, to 38% for DDR3 and potentially to less than 30% for DDR4.

The specification defined standards for ×4, ×8 and ×16 memory devices with capacities of 2, 4, 8 and 16 Gb.

**Command encoding** 🡪🡪🡪🡪🡪🡪🡪🡪🡪🡪🡪🡪🡪🡪🡪🡪🡪

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DDR4 command encoding** | | | | | | | | | | | | |
| **Command** | **CS** | **BG1–0, BA1–0** | **ACT** | **A17** | **A16 RAS** | **A15 CAS** | **A14 WE** | **A13** | **A12 BC** | **A11** | **A10 AP** | **A9–0** |
| Deselect (no operation) | H | X | | | | | | | | | | |
| Active (activate): open a row | L | Bank | L | Row address | | | | | | | | |
| No operation | L | V | H | V | H | H | H | V | | | | |
| ZQ calibration | L | V | H | V | H | H | L | V | | | Long | V |
| Read (BC, burst chop) | L | Bank | H | V | H | L | H | V | BC | V | AP | Column |
| Write (AP, auto-precharge) | L | Bank | H | V | H | L | L | V | BC | V | AP | Column |
| Unassigned, reserved | L | V | v | V | L | H | H | V | | | | |
| Precharge all banks | L | V | H | V | L | H | L | V | | | H | V |
| Precharge one bank | L | Bank | H | V | L | H | L | V | | | L | V |
| Refresh | L | V | H | V | L | L | H | V | | | | |
| Mode register set (MR0–MR6) | L | Register | H | L | L | L | L | L | Data | | | |
| * Signal level   + H,h   + L, low   + V, either low or high, a valid signal   + X, irrelevant * Logic level   + Active   + Inactive   + Not interpreted | | | | | | | | | | | | |

Although it still operates in fundamentally the same way, DDR4 makes one major change to the [command formats used by previous SDRAM generations](https://en.wikipedia.org/wiki/SDRAM#Commands). A new command signal, ACT, is low to indicate the activate (open row) command.

The activate command requires more address bits than any other (18 row address bits in an 16 Gb part), so the standard RAS, CAS, and WE [active low](https://en.wikipedia.org/wiki/Active_low) signals are shared with high-order address bits that are not used when ACT is high. The combination of RAS=L and CAS=WE=H that previously encoded an activate command is unused.

As in previous SDRAM encodings, A10 is used to select command variants: auto-precharge on read and write commands, and one bank vs. all banks for the precharge command. It also selects two variants of the ZQ calibration command.

As in DDR3, A12 is used to request *burst chop*: truncation of an 8-transfer burst after four transfers. Although the bank is still busy and unavailable for other commands until eight transfer times have elapsed, a different bank can be accessed.

Also, the number of bank addresses has been increased greatly. There are four bank select bits to select up to 16 banks within each DRAM: two bank address bits (BA0, BA1), and two bank group bits (BG0, BG1). There are additional timing restrictions when accessing banks within the same bank group; it is faster to access a bank in a different bank group.

In addition, there are three chip select signals (C0, C1, C2), allowing up to eight [stacked chips](https://en.wikipedia.org/wiki/Multi-Chip_Module#Chip_stack_MCMs) to be placed inside a single DRAM package. These effectively act as three more bank select bits, bringing the total to seven (128 possible banks).

Standard transfer rates are 1600, 1866, 2133, 2400, 2666, 2933, and 3200 MT/s (​12⁄15, ​14⁄15, ​16⁄15, ​18⁄15, ​20⁄15, ​22⁄15, and ​24⁄15 GHz clock frequencies, double data rate), with speeds up to DDR4-4800 (2400 MHz clock) commercially available.

**Processor**

A processor is the logic circuitry that responds to and processes the basic [instructions](https://whatis.techtarget.com/definition/instruction) that drive a computer. The four primary [functions](https://whatis.techtarget.com/definition/function) of a processor are [fetch](https://searchsqlserver.techtarget.com/definition/fetch), decode, execute and write back.

**The basic elements of a processor:**

The arithmetic logic unit ([ALU](https://whatis.techtarget.com/definition/arithmetic-logic-unit-ALU)), which carries out arithmetic and logic [operations](https://whatis.techtarget.com/definition/operator) on the [operands](https://whatis.techtarget.com/definition/operand) in [instructions](https://whatis.techtarget.com/definition/instruction).

The floating point unit ([FPU](https://searchwindowsserver.techtarget.com/definition/floating-point-unit-FPU)), also known as a math coprocessor or numeric coprocessor, a specialized [coprocessor](https://whatis.techtarget.com/definition/coprocessor) that manipulates numbers more quickly than the basic microprocessor circuitry can.

[Registers](https://whatis.techtarget.com/definition/register), which hold instructions and other data. Registers supply operands to the ALU and store the results of operations.

[L1 and L2](https://whatis.techtarget.com/definition/L1-and-L2) [cache memory](https://searchstorage.techtarget.com/definition/cache-memory). Their inclusion in the CPU saves time compared to having to get data from random access memory ([RAM](https://searchstorage.techtarget.com/definition/RAM-random-access-memory)).

Most processors today are [multi-core](https://searchdatacenter.techtarget.com/definition/multi-core-processor), which means that the IC contains two or more [processors](https://whatis.techtarget.com/definition/processor) for enhanced performance, reduced power consumption and more efficient simultaneous processing of multiple tasks (s*ee:* [parallel processing](https://searchdatacenter.techtarget.com/definition/parallel-processing)). Multi-core set-ups are similar to having multiple, separate processors installed in the same computer, but because the processors are actually plugged into the same socket, the connection between them is faster.

The term *processor* is used interchangeably with the term central processing unit ([CPU](https://whatis.techtarget.com/definition/processor)), although strictly speaking, the CPU is not the only processor in a computer. The [GPU](https://searchvirtualdesktop.techtarget.com/definition/GPU-graphics-processing-unit)(graphics processing unit) is the most notable example but the hard drive and other devices within a computer also perform some processing independently. Nevertheless, the term *processor* is generally understood to mean the CPU.

The processor in a personal computer or embedded in small devices is often called a [microprocessor](https://whatis.techtarget.com/definition/microprocessor-logic-chip). That term simply means that the processor's elements are contained on a single integrated circuitry ([IC](https://whatis.techtarget.com/definition/integrated-circuit-IC)) [chip](https://whatis.techtarget.com/definition/microchip).

## C:\Users\Admin\Desktop\media\image8.png

## X86 integer instructions

This is the full 8086/8088 instruction set of Intel. Most if not all of these instructions are available in 32-bit mode; they just operate on 32-bit registers (**eax**, **ebx**, etc.) and values instead of their 16-bit (**ax**, **bx**, etc.) counterparts. See also [x86 assembly language](https://en.wikipedia.org/wiki/X86_assembly_language) for a quick tutorial for this processor family. The updated instruction set is also grouped according to architecture ([i386](https://en.wikipedia.org/wiki/I386), [i486](https://en.wikipedia.org/wiki/I486), [i686](https://en.wikipedia.org/wiki/I686)) and more generally is referred to as [x86 32](https://en.wikipedia.org/wiki/X86_32) and [x86 64](https://en.wikipedia.org/wiki/X86_64) (also known as [AMD64](https://en.wikipedia.org/wiki/AMD64)).

**Original 8086/8088 instructions**

| **Original 8086/8088 instruction set** | | | |
| --- | --- | --- | --- |
| **Instruction** | **Meaning** | **Notes** | **Opcode** |
| [AAA](https://en.wikipedia.org/wiki/Intel_BCD_opcodes) | ASCII adjust AL after addition | used with unpacked [binary coded decimal](https://en.wikipedia.org/wiki/Binary_coded_decimal) | 0x37 |
| AAD | ASCII adjust AX before division | 8086/8088 datasheet documents only base 10 version of the AAD instruction ([opcode](https://en.wikipedia.org/wiki/Opcode" \o "Opcode) 0xD5 0x0A), but any other base will work. Later Intel's documentation has the generic form too. NEC V20 and V30 (and possibly other NEC V-series CPUs) always use base 10, and ignore the argument, causing a number of incompatibilities | 0xD5 |
| AAM | ASCII adjust AX after multiplication | Only base 10 version (Operand is 0xA) is documented, see notes for AAD | 0xD4 |
| AAS | ASCII adjust AL after subtraction |  | 0x3F |
| ADC | Add with carry | destination := destination + source + [carry\_flag](https://en.wikipedia.org/wiki/Carry_flag" \o "Carry flag) | 0x10…0x15, 0x80/2…0x83/2 |
| ADD | Add | (1) r/m += r/imm; (2) r += m/imm; | 0x00…0x05, 0x80/0…0x83/0 |
| AND | [Logical AND](https://en.wikipedia.org/wiki/Logical_conjunction) | (1) r/m &= r/imm; (2) r &= m/imm; | 0x20…0x25, 0x80/4…0x83/4 |
| CALL | Call procedure | push eip*; eip points to the instruction directly after the call* | 0x9A, 0xE8, 0xFF/2, 0xFF/3 |
| CBW | Convert byte to word |  | 0x98 |
| CLC | Clear [carry flag](https://en.wikipedia.org/wiki/Carry_flag) | CF = 0; | 0xF8 |
| CLD | Clear [direction flag](https://en.wikipedia.org/wiki/Direction_flag) | DF = 0; | 0xFC |
| [CLI](https://en.wikipedia.org/wiki/CLI_(x86_instruction)) | Clear [interrupt flag](https://en.wikipedia.org/wiki/IF_(x86_flag)) | IF = 0; | 0xFA |
| CMC | Complement carry flag |  | 0xF5 |
| CMP | Compare operands |  | 0x38…0x3D, 0x80/7…0x83/7 |
| CMPSB | Compare bytes in memory |  | 0xA6 |
| CMPSW | Compare words |  | 0xA7 |
| CWD | Convert word to double word |  | 0x99 |
| [DAA](https://en.wikipedia.org/wiki/Intel_BCD_opcodes) | Decimal adjust AL after addition | (used with packed [binary coded decimal](https://en.wikipedia.org/wiki/Binary_coded_decimal)) | 0x27 |
| [DAS](https://en.wikipedia.org/wiki/Intel_BCD_opcodes) | Decimal adjust AL after subtraction |  | 0x2F |
| DEC | Decrement by 1 |  | 0x48…0x4F, 0xFE/1, 0xFF/1 |
| DIV | Unsigned divide | DX:AX = DX:AX / r/m; resulting DX == remainder | 0xF6/6, 0xF7/6 |
| ESC | Used with [floating-point unit](https://en.wikipedia.org/wiki/Floating-point_unit) |  | 0xD8..0xDF |
| [HLT](https://en.wikipedia.org/wiki/HLT_(x86_instruction)) | Enter halt state |  | 0xF4 |
| IDIV | Signed divide | DX:AX = DX:AX / r/m; resulting DX == remainder | 0xF6/7, 0xF7/7 |
| IMUL | Signed multiply | (1) DX:AX = AX \* r/m; (2) AX = AL \* r/m | 0x69, 0x6B (both since 80186), 0xF6/5, 0xF7/5, 0x0FAF (since 80386) |
| IN | Input from port | (1) AL = port[imm]; (2) AL = port[DX]; (3) AX = port[imm]; (4) AX = port[DX]; | 0xE4, 0xE5, 0xEC, 0xED |
| INC | Increment by 1 |  | 0x40…0x47, 0xFE/0, 0xFF/0 |
| [INT](https://en.wikipedia.org/wiki/INT_(x86_instruction)) | Call to [interrupt](https://en.wikipedia.org/wiki/Interrupt) |  | 0xCC, 0xCD |
| INTO | Call to interrupt if overflow |  | 0xCE |
| IRET | Return from interrupt |  | 0xCF |
| Jcc | [Jump if condition](https://en.wikipedia.org/wiki/Branch_(computer_science)) | (JA, JAE, JB, JBE, JC, JE, JG, JGE, JL, JLE, JNA, JNAE, JNB, JNBE, JNC, JNE, JNG, JNGE, JNL, JNLE, JNO, JNP, JNS, JNZ, JO, JP, JPE, JPO, JS, JZ) | 0x70…0x7F, 0x0F80…0x0F8F (since 80386) |
| JCXZ | Jump if CX is zero |  | 0xE3 |
| [JMP](https://en.wikipedia.org/wiki/JMP_(x86_instruction)) | Jump |  | 0xE9…0xEB, 0xFF/4, 0xFF/5 |
| LAHF | Load FLAGS into AH register |  | 0x9F |
| LDS | Load pointer using DS |  | 0xC5 |
| LEA | [Load Effective Address](https://en.wikipedia.org/wiki/Load_Effective_Address) |  | 0x8D |
| LES | Load ES with pointer |  | 0xC4 |
| LOCK | Assert BUS LOCK# signal | (for multiprocessing) | 0xF0 |
| LODSB | Load string byte | **if** (DF==0) AL = \*SI++; **else** AL = \*SI--; | 0xAC |
| LODSW | Load string word | **if** (DF==0) AX = \*SI++; **else** AX = \*SI--; | 0xAD |
| LOOP/LOOPx | Loop control | (LOOPE, LOOPNE, LOOPNZ, LOOPZ) **if** (x && --CX) **goto** lbl; | 0xE0…0xE2 |
| MOV | Move | copies data from one location to another, (1) r/m = r; (2) r = r/m; | 0xA0...0xA3 |
| MOVSB | Move byte from string to string | **if** (DF==0)  \*(byte\*)DI++ = \*(byte\*)SI++;  **else**  \*(byte\*)DI-- = \*(byte\*)SI--; | 0xA4 |
| MOVSW | Move word from string to string | **if** (DF==0)  \*(word\*)DI++ = \*(word\*)SI++;  **else**  \*(word\*)DI-- = \*(word\*)SI--; | 0xA5 |
| MUL | Unsigned multiply | (1) DX:AX = AX \* r/m; (2) AX = AL \* r/m; | 0xF6/4…0xF7/4 |
| NEG | Two's complement negation | r/m \*= -1; | 0xF6/3…0xF7/3 |
| [NOP](https://en.wikipedia.org/wiki/NOP_(code)) | No operation | opcode equivalent to XCHG EAX, EAX | 0x90 |
| NOT | Negate the operand, [logical NOT](https://en.wikipedia.org/wiki/Bitwise_operation#NOT) | r/m ^= -1; | 0xF6/2…0xF7/2 |
| OR | [Logical OR](https://en.wikipedia.org/wiki/Logical_disjunction) | (1) r/m |= r/imm; (2) r |= m/imm; | 0x08…0x0D, 0x80…0x83/1 |
| OUT | Output to port | (1) port[imm] = AL; (2) port[DX] = AL; (3) port[imm] = AX; (4) port[DX] = AX; | 0xE6, 0xE7, 0xEE, 0xEF |
| POP | Pop data from [stack](https://en.wikipedia.org/wiki/Stack_(data_structure)) | r/m = \*SP++; POP CS (opcode 0x0F) works only on 8086/8088. Later CPUs use 0x0F as a prefix for newer instructions. | 0x07, 0x0F(8086/8088 only), 0x17, 0x1F, 0x58…0x5F, 0x8F/0 |
| POPF | Pop [FLAGS register](https://en.wikipedia.org/wiki/FLAGS_register_(computing)) from stack | FLAGS = \*SP++; | 0x9D |
| PUSH | Push data onto stack | \*--SP = r/m; | 0x06, 0x0E, 0x16, 0x1E, 0x50…0x57, 0x68, 0x6A (both since 80186), 0xFF/6 |
| PUSHF | Push FLAGS onto stack | \*--SP = FLAGS; | 0x9C |
| RCL | Rotate left (with carry) |  | 0xC0…0xC1/2 (since 80186), 0xD0…0xD3/2 |
| RCR | Rotate right (with carry) |  | 0xC0…0xC1/3 (since 80186), 0xD0…0xD3/3 |
| REPxx | Repeat MOVS/STOS/CMPS/LODS/SCAS | (REP, REPE, REPNE, REPNZ, REPZ) | 0xF2, 0xF3 |
| RET | Return from procedure | Not a real instruction. The assembler will translate these to a RETN or a RETF depending on the memory model of the target system. |  |
| RETN | Return from near procedure |  | 0xC2, 0xC3 |
| RETF | Return from far procedure |  | 0xCA, 0xCB |
| ROL | Rotate left |  | 0xC0…0xC1/0 (since 80186), 0xD0…0xD3/0 |
| ROR | Rotate right |  | 0xC0…0xC1/1 (since 80186), 0xD0…0xD3/1 |
| SAHF | Store AH into FLAGS |  | 0x9E |
| SAL | [Shift Arithmetically](https://en.wikipedia.org/wiki/Arithmetic_shift) left (signed shift left) | (1) r/m <<= 1; (2) r/m <<= CL; | 0xC0…0xC1/4 (since 80186), 0xD0…0xD3/4 |
| SAR | Shift Arithmetically right (signed shift right) | (1) (signed) r/m >>= 1; (2) (signed) r/m >>= CL; | 0xC0…0xC1/7 (since 80186), 0xD0…0xD3/7 |
| SBB | Subtraction with borrow | alternative 1-byte encoding of SBB AL, AL is available via [undocumented](https://en.wikipedia.org/wiki/X86_instruction_listings#Undocumented_instructions) SALC instruction | 0x18…0x1D, 0x80…0x83/3 |
| SCASB | Compare byte string |  | 0xAE |
| SCASW | Compare word string |  | 0xAF |
| SHL | [Shift](https://en.wikipedia.org/wiki/Logical_shift) left (unsigned shift left) |  | 0xC0…0xC1/4 (since 80186), 0xD0…0xD3/4 |
| SHR | Shift right (unsigned shift right) |  | 0xC0…0xC1/5 (since 80186), 0xD0…0xD3/5 |
| STC | Set carry flag | CF = 1; | 0xF9 |
| STD | Set direction flag | DF = 1; | 0xFD |
| [STI](https://en.wikipedia.org/wiki/STI_(x86_instruction)) | Set interrupt flag | IF = 1; | 0xFB |
| STOSB | Store byte in string | **if** (DF==0) \*ES:DI++ = AL; **else** \*ES:DI-- = AL; | 0xAA |
| STOSW | Store word in string | **if** (DF==0) \*ES:DI++ = AX; **else** \*ES:DI-- = AX; | 0xAB |
| SUB | Subtraction | (1) r/m -= r/imm; (2) r -= m/imm; | 0x28…0x2D, 0x80…0x83/5 |
| [TEST](https://en.wikipedia.org/wiki/TEST_(x86_instruction)) | Logical compare (AND) | (1) r/m & r/imm; (2) r & m/imm; | 0x84, 0x84, 0xA8, 0xA9, 0xF6/0, 0xF7/0 |
| WAIT | Wait until not busy | Waits until BUSY# pin is inactive (used with [floating-point unit](https://en.wikipedia.org/wiki/Floating-point_unit)) | 0x9B |
| XCHG | Exchange data | r :=: r/m*;* A [spinlock](https://en.wikipedia.org/wiki/Spinlock) typically uses xchg as an [atomic operation](https://en.wikipedia.org/wiki/Atomic_operation). ([coma bug](https://en.wikipedia.org/wiki/Coma_bug)). | 0x86, 0x87, 0x91…0x97 |
| XLAT | Table look-up translation | behaves like MOV AL, [BX+AL] | 0xD7 |
| XOR | [Exclusive OR](https://en.wikipedia.org/wiki/Exclusive_or) | (1) r/m ^= r/imm; (2) r ^= m/imm; | 0x30…0x35, 0x80…0x83/6 |

References:

* [www.google.in](http://www.google.in)
* <https://www.laptopmag.com/reviews/laptops/acer-aspire-5>
* [www.amazon.com](http://www.amazon.com)