

**MSPM’S**

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

**Aurangabad**

**Department of Computer Science and Engineering**

Survey Based Project on

Hp pavilion 14-ce1001tx

Asus Vivobook s15 s530un

Subject: **Computer Architecture and Organisation**

Submitted By:

**Ansari Mohd Akef Bilal (26143)**

Under Guidance of

**Prof. P.H. Durole**

Asst. Prof. Department of CSE

(Deogiri Institute of Engineering and Management Studies)

2019-2020



**CERTIFICATE**

This is to certify that Mr **Ansari Mohd Akef Bilal** had successfully completed his Survey Based Project on Hp Pavilion 14 ce1001tx and Asus Vivobook s15 on the date 05/09/2019

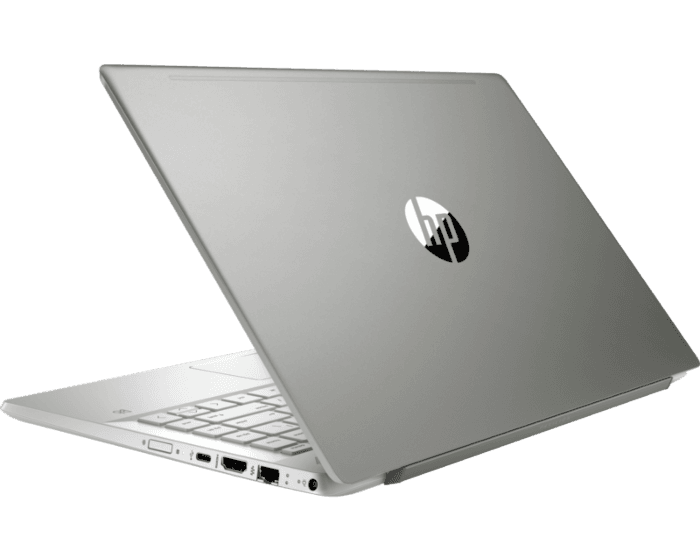
Name of Guide

Prof.**P.H. Durole**

Asst. Prof. Department of CSE

Hp pavilion 14-ce1001tx

Photo



Tech Specification

|  |  |
| --- | --- |
| **Operating system** | **Windows 10 Home Single Language 64** |
| **Processor Name** | **Intel® Core™ i5-8265U (1.6 GHz base frequency, up to 3.9 GHz with Intel® Turbo Boost Technology, 6 MB cache, 4 cores)** |
| **Processor footnote** | **[6] Multi-core is designed to improve performance of certain software products. Not all customers or software applications will necessarily benefit from use of this technology. Performance and clock frequency will vary depending on application workload and your hardware and software configurations. Intel’s numbering is not a measurement of higher performance. Intel, Pentium, Intel Core, Celeron, Intel logo and the Intel Inside logo are trademarks of Intel Corporation in the U.S. and other countries. [7] Intel® Turbo Boost performance varies depending on hardware, software and overall system configuration. See http://www.intel.com/technology/turboboost/ for more information.** |
| **Processor family** | **8th Generation Intel® Core™ i5 processor** |
| **Form factor** | **Standard laptop** |
| MEMORY | |
| **Memory** | **8 GB DDR4-2400 SDRAM (1 x 8 GB)** |
| **Memory Note** | **Transfer rates up to 2400 MT/s.** |
| STORAGE | |
| **Hard drive description** | **1 TB 5400 rpm SATA + 128 GB M.2 SSD** |
| **Hard drive (2nd)** | **128 GB M.2 SSD** |
| **Storage type** | **HDD; SSD** |
| **Cloud service** | **Dropbox** |
| DISPLAY AND GRAPHICS | |
| **Display HP** | **35.56cm(14) diagonal FHD IPS BrightView WLED-backlit (1920 x 1080)** |
| **Graphics** | **NVIDIA® GeForce® MX150 (2 GB GDDR5 dedicated)** |
| EXPANSION FEATURES | |
| **Ports** | **1 USB Type-C™ 3.1 Gen 1 (Data transfer only, up to 5Gb/s); 2 USB 3.1 Gen 1 (1 HP Sleep and Charge); 1 HDMI 1.4; 1 RJ-45; 1 headphone/microphone combo** |
| **Expansion slots** | **1 multi-format SD media card reader** |
| MEDIA DEVICES | |
| **Audio features** | **B&O PLAY, dual speakers, HP Audio Boost** |
| **Webcam** | **HP Wide Vision HD Camera with integrated dual array digital microphone** |
| INPUT DEVICES | |
| **Pointing device** | **HP Imagepad with multi-touch gesture support** |
| **Keyboard** | **Full-size island-style backlit keyboard with numeric keypad** |
| COMMUNICATIONS | |
| **Network interface** | **Integrated 10/100/1000 GbE LAN** |
| **Wireless** | **Intel® Wireless-AC 9560 802.11a/b/g/n/ac (2x2) Wi-Fi® and Bluetooth® 5 Combo** |
| **Wireless note** | **MU-MIMO supported; Miracast compatible** |
| POWER AND OPERATING REQUIREMENTS | |
| **Power supply type** | **65 W EM AC power adapter** |
| **Battery type** | **3-cell, 41 Wh Li-ion** |
| **Battery weight** | **210 g** |
| **Energy Efficiency Compliance** | **ENERGY STAR® certified; EPEAT® Silver registered** |
| DIMENSIONS AND WEIGHT | |
| **Dimensions without stand (W x D x H)** | **32.68 x 22.17 x 1.79 cm** |
| **Weight** | **1.6 kg** |
| DESIGN | |
| **Product color** | **Mineral silver** |
| SECURITY MANAGEMENT | |
| **Security management** | **Kensington Nano Security Slot™; Fingerprint reader** |
| SOFTWARE | |
| **HP apps** | **HP Audio Switch; HP CoolSense; HP Documentation; HP ePrint; HP JumpStart; HP Support Assistant; HP Connection Optimizer** |
| **Software included** | **McAfee LiveSafe™** |
| **Software footnote** | **[1] McAfee LiveSafe 30-day free trial offer (Internet access required. First 30 days included. Subscription required for live updates afterwards.)** |
| WARRANTY AND SERVICES | |
| **Warranty** | **1 year limited parts and labour** |

Processor

i-5 8th generation-8265u

* Code Name Products formerly Whiskey Lake
* Vertical Segment Mobile
* Processor Numberi5-8265U
* Status Launched
* Launch DateQ3'18
* Lithography14 nm
* Recommended Customer Price $297.00

## Performance

* No of Cores 4
* No of Threads 8
* Processor Base Frequency 1.60 GHz
* Max Turbo Frequency 3.90 GHz
* Cache 6 MB Smar tCache
* Bus Speed4 GT/s OPI
* TDP15 W
* Configurable TDP-up Frequency 1.80 GHz
* Configurable TDP-up 25 W
* Configurable TDP-down Frequency 800 MHz
* Configurable TDP-down 10 W

## Memory Specifications

* Max Memory Size (dependent on memory type) 64 GB
* Memory Types DDR4-2400, LPDDR3-2133
* Max no of Memory Channels 2
* Max Memory Bandwidth 37.5 GB/s
* ECC Memory Supported No

## Processor Graphics

* Processor GraphicsIntel® UHD Graphics 620
* Graphics Base Frequency 300 MHz
* Graphics Max Dynamic Frequency 1.10 GHz
* Graphics Video Max Memory 32 GB
* Graphics Output DP/DP/HDMI/DVI
* 4K Support Yes, at 60Hz
* Max Resolution (HDMI 1.4) 4096x2304@24Hz
* Max Resolution (DP) 4096x2304@60Hz
* Max Resolution (eDP - Integrated Flat Panel) 4096x2304@60Hz
* DirectX\* Support12
* OpenGL\* Support4.5
* Intel® Quick Sync Video Yes
* Intel® Clear Video HD Technology Yes
* Intel® Clear Video Technology Yes
* No of Displays Supported 3
* Device ID0x3EA0

## Expansion Options

* PCI Express Revision 3.0
* PCI Express Configurations 1x4, 2x2, 1x2+2x1 and 4x1
* Max # of PCI Express Lanes 16

## Package Specifications

* Sockets Supported FCBGA1528
* Max CPU Configuration1
* TJUNCTION 100°C
* Package Size 46x24

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.m.wikipedia.org/wiki/Synchronous_dynamic_random-access_memory) with a high bandwidth ("double data rate") interface.

|  |  |
| --- | --- |
| **DDR4 SDRAM Double Data Rate 4 Synchronous Dynamic Random-Access Memory** | |
| Type of RAM | |
| 8 GiB DDR4-2133 ECC 1.2 V RDIMM | |
| **Developer** | JEDEC |
| **Type** | Synchronous dynamic random-access memory (SDRAM) |
| **Generation** | 4th generation |
| **Release date** | 2014 |
| **Standards** | * DDR4-1600 (PC4-12800) * DDR4-1866 (PC4-14900) * DDR4-2133 (PC4-17000) * DDR4-2400 (PC4-19200) * DDR4-2666 (PC4-21333) * DDR4-2933 (PC4-23466) * DDR4-3200 (PC4-25600) |
| **Clock rate** | 800–2133 MHz |
| **Voltage** | 1.2~1.4 V |
| **Predecessor** | DDR3 SDRAM |
| **Successor** | DDR5 SDRAM |

Graphics

Photo



## **GeForce® MX150 Dedicated Graphics for Laptops**

NVIDIA® GeForce® MX150 supercharges your laptop for work and play. Get up to 4X faster performance over integrated graphics for photo and video-editing applications, as well as faster, smoother gaming

Up to 4X1 GeForce Performance Score

**Memory Specs:**

GDDR5Standard Memory Interface

**Technology Support:**

Yes NVIDIA® Optimus™ Support

3. NVIDIA GPU Boost™

Yes NVIDIA GameWorks™ support

12 API with feature level 12\_1Microsoft DirectX

Yes Vulcan API

4.5OpenGL Support

Yes Windows 7, 8, and 10 support

Yes PCI Express 3.0 support

Instruction set

# x86-64 Instructions Set

## CPU instructions

The general-purpose instructions perform basic data movement, arithmetic, logic, program flow, and string operations which programmers commonly use to write application and system software to run on Intel 64 and IA-32 processors. They operate on data contained in memory, in the general-purpose registers (EAX, EBX, ECX, EDX, EDI, ESI, EBP, and ESP) and in the EFLAGS register. They also operate on address information contained in memory, the general-purpose registers, and the segment registers (CS, DS, SS, ES, FS, and GS).

It performers the following instructions

* Data Transfer Instructions
* Arithmetic Instructions
* Bit Manipulation Instructions
* String Instructions
* Program Execution Transfer Instructions (Branch & Loop Instructions)
* Processor Control Instructions
* Iteration Control Instructions
* Interrupt Instructions

Let us now discuss these instruction sets in detail.

## **Data Transfer Instructions**

These instructions are used to transfer the data from the source operand to the destination operand. Following are the list of instructions under this group −

### **Instruction to transfer a word**

* **MOV** − Used to copy the byte or word from the provided source to the provided destination.
* **PPUSH** − Used to put a word at the top of the stack.
* **POP** − Used to get a word from the top of the stack to the provided location.
* **PUSHA** − Used to put all the registers into the stack.
* **POPA** − Used to get words from the stack to all registers.
* **XCHG** − Used to exchange the data from two locations.
* **XLAT** − Used to translate a byte in AL using a table in the memory.

### **Instructions for input and output port transfer**

* **IN** − Used to read a byte or word from the provided port to the accumulator.
* **OUT** − Used to send out a byte or word from the accumulator to the provided port.

### **Instructions to transfer the address**

* **LEA** − Used to load the address of operand into the provided register.
* **LDS** − Used to load DS register and other provided register from the memory
* **LES** − Used to load ES register and other provided register from the memory.

### **Instructions to transfer flag registers**

* **LAHF** − Used to load AH with the low byte of the flag register.
* **SAHF** − Used to store AH register to low byte of the flag register.
* **PUSHF** − Used to copy the flag register at the top of the stack.
* **POPF** − Used to copy a word at the top of the stack to the flag register.

## **Arithmetic Instructions**

These instructions are used to perform arithmetic operations like addition, subtraction, multiplication, division, etc.

Following is the list of instructions under this group −

### **Instructions to perform addition**

* **ADD** − Used to add the provided byte to byte/word to word.
* **ADC** − Used to add with carry.
* **INC** − Used to increment the provided byte/word by 1.
* **AAA** − Used to adjust ASCII after addition.
* **DAA** − Used to adjust the decimal after the addition/subtraction operation.

### **Instructions to perform subtraction**

* **SUB** − Used to subtract the byte from byte/word from word.
* **SBB** − Used to perform subtraction with borrow.
* **DEC** − Used to decrement the provided byte/word by 1.
* **NPG** − Used to negate each bit of the provided byte/word and add 1/2’s complement.
* **CMP** − Used to compare 2 provided byte/word.
* **AAS** − Used to adjust ASCII codes after subtraction.
* **DAS** − Used to adjust decimal after subtraction.

### **Instruction to perform multiplication**

* **MUL** − Used to multiply unsigned byte by byte/word by word.
* **IMUL** − Used to multiply signed byte by byte/word by word.
* **AAM** − Used to adjust ASCII codes after multiplication.

### **Instructions to perform division**

* **DIV** − Used to divide the unsigned word by byte or unsigned double word by word.
* **IDIV** − Used to divide the signed word by byte or signed double word by word.
* **AAD** − Used to adjust ASCII codes after division.
* **CBW** − Used to fill the upper byte of the word with the copies of sign bit of the lower byte.
* **CWD** − Used to fill the upper word of the double word with the sign bit of the lower word.

## **Bit Manipulation Instructions**

These instructions are used to perform operations where data bits are involved, i.e. operations like logical, shift, etc.

Following is the list of instructions under this group −

### **Instructions to perform logical operation**

* **NOT** − Used to invert each bit of a byte or word.
* **AND** − Used for adding each bit in a byte/word with the corresponding bit in another byte/word.
* **OR** − Used to multiply each bit in a byte/word with the corresponding bit in another byte/word.
* **XOR** − Used to perform Exclusive-OR operation over each bit in a byte/word with the corresponding bit in another byte/word.
* **TEST** − Used to add operands to update flags, without affecting operands.

### **Instructions to perform shift operations**

* **SHL/SAL** − Used to shift bits of a byte/word towards left and put zero(S) in LSBs.
* **SHR** − Used to shift bits of a byte/word towards the right and put zero(S) in MSBs.
* **SAR** − Used to shift bits of a byte/word towards the right and copy the old MSB into the new MSB.

### **Instructions to perform rotate operations**

* **ROL** − Used to rotate bits of byte/word towards the left, i.e. MSB to LSB and to Carry Flag [CF].
* **ROR** − Used to rotate bits of byte/word towards the right, i.e. LSB to MSB and to Carry Flag [CF].
* **RCR** − Used to rotate bits of byte/word towards the right, i.e. LSB to CF and CF to MSB.
* **RCL** − Used to rotate bits of byte/word towards the left, i.e. MSB to CF and CF to LSB.

## **String Instructions**

String is a group of bytes/words and their memory is always allocated in a sequential order.

Following is the list of instructions under this group −

* **REP** − Used to repeat the given instruction till CX ≠ 0.
* **REPE/REPZ** − Used to repeat the given instruction until CX = 0 or zero flag ZF = 1.
* **REPNE/REPNZ** − Used to repeat the given instruction until CX = 0 or zero flag ZF = 1.
* **MOVS/MOVSB/MOVSW** − Used to move the byte/word from one string to another.
* **COMS/COMPSB/COMPSW** − Used to compare two string bytes/words.
* **INS/INSB/INSW** − Used as an input string/byte/word from the I/O port to the provided memory location.
* **OUTS/OUTSB/OUTSW** − Used as an output string/byte/word from the provided memory location to the I/O port.
* **SCAS/SCASB/SCASW** − Used to scan a string and compare its byte with a byte in AL or string word with a word in AX.
* **LODS/LODSB/LODSW** − Used to store the string byte into AL or string word into AX.

## **Program Execution Transfer Instructions (Branch and Loop Instructions)**

These instructions are used to transfer/branch the instructions during an execution. It includes the following instructions −

Instructions to transfer the instruction during an execution without any condition −

* **CALL** − Used to call a procedure and save their return address to the stack.
* **RET** − Used to return from the procedure to the main program.
* **JMP** − Used to jump to the provided address to proceed to the next instruction.

Instructions to transfer the instruction during an execution with some conditions −

* **JA/JNBE** − Used to jump if above/not below/equal instruction satisfies.
* **JAE/JNB** − Used to jump if above/not below instruction satisfies.
* **JBE/JNA** − Used to jump if below/equal/ not above instruction satisfies.
* **JC** − Used to jump if carry flag CF = 1
* **JE/JZ** − Used to jump if equal/zero flag ZF = 1
* **JG/JNLE** − Used to jump if greater/not less than/equal instruction satisfies.
* **JGE/JNL** − Used to jump if greater than/equal/not less than instruction satisfies.
* **JL/JNGE** − Used to jump if less than/not greater than/equal instruction satisfies.
* **JLE/JNG** − Used to jump if less than/equal/if not greater than instruction satisfies.
* **JNC** − Used to jump if no carry flag (CF = 0)
* **JNE/JNZ** − Used to jump if not equal/zero flag ZF = 0
* **JNO** − Used to jump if no overflow flag OF = 0
* **JNP/JPO** − Used to jump if not parity/parity odd PF = 0
* **JNS** − Used to jump if not sign SF = 0
* **JO** − Used to jump if overflow flag OF = 1
* **JP/JPE** − Used to jump if parity/parity even PF = 1
* **JS** − Used to jump if sign flag SF = 1

## **Processor Control Instructions**

These instructions are used to control the processor action by setting/resetting the flag values.

Following are the instructions under this group −

* **STC** − Used to set carry flag CF to 1
* **CLC** − Used to clear/reset carry flag CF to 0
* **CMC** − Used to put complement at the state of carry flag CF.
* **STD** − Used to set the direction flag DF to 1
* **CLD** − Used to clear/reset the direction flag DF to 0
* **STI** − Used to set the interrupt enable flag to 1, i.e., enable INTR input.
* **CLI** − Used to clear the interrupt enable flag to 0, i.e., disable INTR input.

## **Iteration Control Instructions**

These instructions are used to execute the given instructions for number of times. Following is the list of instructions under this group −

* **LOOP** − Used to loop a group of instructions until the condition satisfies, i.e., CX = 0
* **LOOPE/LOOPZ** − Used to loop a group of instructions till it satisfies ZF = 1 & CX = 0
* **LOOPNE/LOOPNZ** − Used to loop a group of instructions till it satisfies ZF = 0 & CX = 0
* **JCXZ** − Used to jump to the provided address if CX = 0

**Interrupt Instructions**

These instructions are used to call the interrupt during program execution.

* **INT** − Used to interrupt the program during execution and calling service specified.
* **INTO** − Used to interrupt the program during execution if OF = 1
* **IRET** − Used to return from interrupt service to the main program

Asus Vivobook S15

Photo



Finish / Color

VivoBook S15 S530UN

Icicle Gold, Gun Metal, Silver Blue, Star Grey, Firmament Green

Operation System

Windows 10 Home  
(ASUS recommends Windows 10 Pro)

Processor

Intel® Core™ i7-8550U processor

1.8GHz quad-core with Turbo boost (up to 4.0GHz) and 8MB cache

Graphics

NVIDIA® GeForce® MX250

Video memory size: 2GB GDDR5 VRAM

Display

39.62cm (15.6”) LED-backlit Full HD (1920 x 1080) 16:9 matte display

Advanced three-sided Nano Edge display with 86% screen-to-body ratio

178° wide-view technology

ASUS Eye Care technology for up to 30% blue-light reduction

Memory

8GB 2400MHz DDR4, expandable up to 16GB

Storage

Dual Storage: SATA3 256GB M.2 SSD + SATA 1TB 5400RPM HDD

Interfaces

1 x Type-C™ USB 3.1 Gen 1

1 x Type-A USB 3.1 Gen 1

2 x USB 2.0

1 x HDMI

1 x Combo audio jack

1 x MicroSD card reader

Keyboard and Touchpad

Keyboard

Full-size backlit, with 1.4mm key travel

Touchpad

Glass-covered with integrated fingerprint sensor supporting Windows Hello; intelligent palm-rejection

Precision touchpad (PTP) technology supports up to four-finger smart gestures

Audio

ASUS Sonic Master stereo audio system with surround-sound; smart amplifier for maximum audio performance

Array microphone with Cortana voice-recognition support

3.5mm headphone jack

Camera

HD webcam

Wireless Connectivity

Wi-Fi Master

Dual-band 802.11ac Wi-Fi with advanced Wi-Fi Master technology

Bluetooth®

Bluetooth V4.2

Battery and Power

Fast charging to 60% in 49 minutes

42Wh 3-cell lithium-prismatic battery

65W power adapter

Plug Type: ø4 (mm)  
(Output: 19V DC, 3.42A, 65W)  
(Input: 100-240V AC, 50/60Hz universal)

Weight and Dimensions

Height: 1.8cm (0.7 inches)

Width: 36.14cm (14.22 inches)

Depth: 24.35cm (9.58 inches)

Weight: 1.8kg (3.97 pounds)

36.14cm  
(14.22 inches)24.35cm  
(9.58 inches)1.8cm  
(0.7 inches)

Included in the Box

VivoBook S15

Mouse (optional)

Included Software

ASUS Splendid

ASUS Eye Care

ASUS Tru2life Video

ASUS AudioWizard

Processor

i7-8550u 8th generation processor

* Product Collection [8th Generation Intel® Core™ i7 Processors](https://ark.intel.com/content/www/us/en/ark/products/series/122593/8th-generation-intel-core-i7-processors.html)
* Code Name Products formerly Kaby Lake R
* Vertical Segment Mobile
* Processor Number i7-8550U
* Status Launched
* Launch Date Q3'17
* Lithography 14 nm
* Recommended Customer Price $409.00

Performance

* No of Cores 4
* No of Threads 8
* Processor Base Frequency 1.80 GHz
* Max Turbo Frequency 4.00 GHz
* Cache 8 MB Smart Cache
* Bus Speed 4 GT/s OPI
* TDP 15 W
* Configurable TDP-up Frequency 2.00 GHz
* Configurable TDP-up 25 W
* Configurable TDP-down Frequency 800 MHz
* Configurable TDP-down 10 W

## Memory Specifications

* Max Memory Size (dependent on memory type) 32 GB
* Memory Types DDR4-2400, LPDDR3-2133
* Max No. of Memory Channels 2
* Max Memory Bandwidth 37.5 GB/s
* ECC Memory Supported No

## Processor Graphics

* Processor GraphicsIntel® UHD Graphics 620
* Graphics Base Frequency 300 MHz
* Graphics Max Dynamic Frequency 1.15 GHz
* Graphics Video Max Memory 32 GB
* Graphics Output eDP/DP/HDMI/DVI
* 4K Support Yes, at 60Hz
* Max Resolution (HDMI 1.4) 4096x2304@24Hz
* Max Resolution (DP) 4096x2304@60Hz
* Max Resolution (eDP - Integrated Flat Panel) 4096x2304@60Hz
* DirectX\* Support 12
* OpenGL\* Support 4.4
* Intel® Quick Sync Video Yes
* Intel® Clear Video HD Technology Yes
* Intel® Clear Video Technology Yes
* No. of Displays Supported 3
* Device ID0 x5917

## Expansion Options

* PCI Express Revision 3.0
* PCI Express Configurations 1x4, 2x2, 1x2+2x1 and 4x1
* Max No. of PCI Express Lane s12

## Package Specifications

* Sockets Supported FC-BGA1356
* Max CPU Configuration 1
* TJUNCTION 100°C
* Package Size 42mm X 24mm

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.m.wikipedia.org/wiki/Synchronous_dynamic_random-access_memory) with a high bandwidth ("double data rate") interface.

|  |  |
| --- | --- |
| **DDR4 SDRAM Double Data Rate 4 Synchronous Dynamic Random-Access Memory** | |
| Type of RAM | |
| 8 GiB DDR4-2133 ECC 1.2 V RDIMM | |
| **Developer** | JEDEC |
| **Type** | Synchronous dynamic random-access memory (SDRAM) |
| **Generation** | 4th generation |
| **Release date** | 2014 |
| **Standards** | * DDR4-1600 (PC4-12800) * DDR4-1866 (PC4-14900) * DDR4-2133 (PC4-17000) * DDR4-2400 (PC4-19200) * DDR4-2666 (PC4-21333) * DDR4-2933 (PC4-23466) * DDR4-3200 (PC4-25600) |
| **Clock rate** | 800–2133 MHz |
| **Voltage** | 1.2~1.4 V |
| **Predecessor** | DDR3 SDRAM |
| **Successor** | DDR5 SDRAM |

Graphics



The NVIDIA® GeForce® MX250 supercharges your laptop for work and play. Now, you can expect up to 3.5X faster performance over integrated graphics for photo and video-editing applications, as well as faster, smoother gaming. It also works seamlessly with NVIDIA Optimus® technology to give you the perfect balance between long battery life and performance.

**GPU Engine Specs:**

Up to 3.5X1 GeForce Performance Score

**Memory Specs:**

GDDR5 Standard Memory Interface

**Technology Support:**

Yes NVIDIA® Optimus™ Support

3.NVIDIA GPU Boost™

Yes NVIDIA GameWorks™ support

No NVIDIA® SLI®-Ready

12 API with feature level 12\_1Microsoft DirectX

1.1Vulcan API

4.6 OpenGL Support

1.2OpenCL Support

Yes Windows 7, 8, and 10 support

Instruction set

# x86-64 Instructions Set

## CPU instructions

The general-purpose instructions perform basic data movement, arithmetic, logic, program flow, and string operations which programmers commonly use to write application and system software to run on Intel 64 and IA-32 processors. They operate on data contained in memory, in the general-purpose registers (EAX, EBX, ECX, EDX, EDI, ESI, EBP, and ESP) and in the EFLAGS register. They also operate on address information contained in memory, the general-purpose registers, and the segment registers (CS, DS, SS, ES, FS, and GS).

It performers the following instructions

* Data Transfer Instructions
* Arithmetic Instructions
* Bit Manipulation Instructions
* String Instructions
* Program Execution Transfer Instructions (Branch & Loop Instructions)
* Processor Control Instructions
* Iteration Control Instructions
* Interrupt Instructions

Let us now discuss these instruction sets in detail.

## **Data Transfer Instructions**

These instructions are used to transfer the data from the source operand to the destination operand. Following are the list of instructions under this group −

### **Instruction to transfer a word**

* **MOV** − Used to copy the byte or word from the provided source to the provided destination.
* **PPUSH** − Used to put a word at the top of the stack.
* **POP** − Used to get a word from the top of the stack to the provided location.
* **PUSHA** − Used to put all the registers into the stack.
* **POPA** − Used to get words from the stack to all registers.
* **XCHG** − Used to exchange the data from two locations.
* **XLAT** − Used to translate a byte in AL using a table in the memory.

### **Instructions for input and output port transfer**

* **IN** − Used to read a byte or word from the provided port to the accumulator.
* **OUT** − Used to send out a byte or word from the accumulator to the provided port.

### **Instructions to transfer the address**

* **LEA** − Used to load the address of operand into the provided register.
* **LDS** − Used to load DS register and other provided register from the memory
* **LES** − Used to load ES register and other provided register from the memory.

### **Instructions to transfer flag registers**

* **LAHF** − Used to load AH with the low byte of the flag register.
* **SAHF** − Used to store AH register to low byte of the flag register.
* **PUSHF** − Used to copy the flag register at the top of the stack.
* **POPF** − Used to copy a word at the top of the stack to the flag register.

## **Arithmetic Instructions**

These instructions are used to perform arithmetic operations like addition, subtraction, multiplication, division, etc.

Following is the list of instructions under this group −

### **Instructions to perform addition**

* **ADD** − Used to add the provided byte to byte/word to word.
* **ADC** − Used to add with carry.
* **INC** − Used to increment the provided byte/word by 1.
* **AAA** − Used to adjust ASCII after addition.
* **DAA** − Used to adjust the decimal after the addition/subtraction operation.

### **Instructions to perform subtraction**

* **SUB** − Used to subtract the byte from byte/word from word.
* **SBB** − Used to perform subtraction with borrow.
* **DEC** − Used to decrement the provided byte/word by 1.
* **NPG** − Used to negate each bit of the provided byte/word and add 1/2’s complement.
* **CMP** − Used to compare 2 provided byte/word.
* **AAS** − Used to adjust ASCII codes after subtraction.
* **DAS** − Used to adjust decimal after subtraction.

### **Instruction to perform multiplication**

* **MUL** − Used to multiply unsigned byte by byte/word by word.
* **IMUL** − Used to multiply signed byte by byte/word by word.
* **AAM** − Used to adjust ASCII codes after multiplication.

### **Instructions to perform division**

* **DIV** − Used to divide the unsigned word by byte or unsigned double word by word.
* **IDIV** − Used to divide the signed word by byte or signed double word by word.
* **AAD** − Used to adjust ASCII codes after division.
* **CBW** − Used to fill the upper byte of the word with the copies of sign bit of the lower byte.
* **CWD** − Used to fill the upper word of the double word with the sign bit of the lower word.

## **Bit Manipulation Instructions**

These instructions are used to perform operations where data bits are involved, i.e. operations like logical, shift, etc.

Following is the list of instructions under this group −

### **Instructions to perform logical operation**

* **NOT** − Used to invert each bit of a byte or word.
* **AND** − Used for adding each bit in a byte/word with the corresponding bit in another byte/word.
* **OR** − Used to multiply each bit in a byte/word with the corresponding bit in another byte/word.
* **XOR** − Used to perform Exclusive-OR operation over each bit in a byte/word with the corresponding bit in another byte/word.
* **TEST** − Used to add operands to update flags, without affecting operands.

### **Instructions to perform shift operations**

* **SHL/SAL** − Used to shift bits of a byte/word towards left and put zero(S) in LSBs.
* **SHR** − Used to shift bits of a byte/word towards the right and put zero(S) in MSBs.
* **SAR** − Used to shift bits of a byte/word towards the right and copy the old MSB into the new MSB.

### **Instructions to perform rotate operations**

* **ROL** − Used to rotate bits of byte/word towards the left, i.e. MSB to LSB and to Carry Flag [CF].
* **ROR** − Used to rotate bits of byte/word towards the right, i.e. LSB to MSB and to Carry Flag [CF].
* **RCR** − Used to rotate bits of byte/word towards the right, i.e. LSB to CF and CF to MSB.
* **RCL** − Used to rotate bits of byte/word towards the left, i.e. MSB to CF and CF to LSB.

## **String Instructions**

String is a group of bytes/words and their memory is always allocated in a sequential order.

Following is the list of instructions under this group −

* **REP** − Used to repeat the given instruction till CX ≠ 0.
* **REPE/REPZ** − Used to repeat the given instruction until CX = 0 or zero flag ZF = 1.
* **REPNE/REPNZ** − Used to repeat the given instruction until CX = 0 or zero flag ZF = 1.
* **MOVS/MOVSB/MOVSW** − Used to move the byte/word from one string to another.
* **COMS/COMPSB/COMPSW** − Used to compare two string bytes/words.
* **INS/INSB/INSW** − Used as an input string/byte/word from the I/O port to the provided memory location.
* **OUTS/OUTSB/OUTSW** − Used as an output string/byte/word from the provided memory location to the I/O port.
* **SCAS/SCASB/SCASW** − Used to scan a string and compare its byte with a byte in AL or string word with a word in AX.
* **LODS/LODSB/LODSW** − Used to store the string byte into AL or string word into AX.

## **Program Execution Transfer Instructions (Branch and Loop Instructions)**

These instructions are used to transfer/branch the instructions during an execution. It includes the following instructions −

Instructions to transfer the instruction during an execution without any condition −

* **CALL** − Used to call a procedure and save their return address to the stack.
* **RET** − Used to return from the procedure to the main program.
* **JMP** − Used to jump to the provided address to proceed to the next instruction.

Instructions to transfer the instruction during an execution with some conditions −

* **JA/JNBE** − Used to jump if above/not below/equal instruction satisfies.
* **JAE/JNB** − Used to jump if above/not below instruction satisfies.
* **JBE/JNA** − Used to jump if below/equal/ not above instruction satisfies.
* **JC** − Used to jump if carry flag CF = 1
* **JE/JZ** − Used to jump if equal/zero flag ZF = 1
* **JG/JNLE** − Used to jump if greater/not less than/equal instruction satisfies.
* **JGE/JNL** − Used to jump if greater than/equal/not less than instruction satisfies.
* **JL/JNGE** − Used to jump if less than/not greater than/equal instruction satisfies.
* **JLE/JNG** − Used to jump if less than/equal/if not greater than instruction satisfies.
* **JNC** − Used to jump if no carry flag (CF = 0)
* **JNE/JNZ** − Used to jump if not equal/zero flag ZF = 0
* **JNO** − Used to jump if no overflow flag OF = 0
* **JNP/JPO** − Used to jump if not parity/parity odd PF = 0
* **JNS** − Used to jump if not sign SF = 0
* **JO** − Used to jump if overflow flag OF = 1
* **JP/JPE** − Used to jump if parity/parity even PF = 1
* **JS** − Used to jump if sign flag SF = 1

## **Processor Control Instructions**

These instructions are used to control the processor action by setting/resetting the flag values.

Following are the instructions under this group −

* **STC** − Used to set carry flag CF to 1
* **CLC** − Used to clear/reset carry flag CF to 0
* **CMC** − Used to put complement at the state of carry flag CF.
* **STD** − Used to set the direction flag DF to 1
* **CLD** − Used to clear/reset the direction flag DF to 0
* **STI** − Used to set the interrupt enable flag to 1, i.e., enable INTR input.
* **CLI** − Used to clear the interrupt enable flag to 0, i.e., disable INTR input.

## **Iteration Control Instructions**

These instructions are used to execute the given instructions for number of times. Following is the list of instructions under this group −

* **LOOP** − Used to loop a group of instructions until the condition satisfies, i.e., CX = 0
* **LOOPE/LOOPZ** − Used to loop a group of instructions till it satisfies ZF = 1 & CX = 0
* **LOOPNE/LOOPNZ** − Used to loop a group of instructions till it satisfies ZF = 0 & CX = 0
* **JCXZ** − Used to jump to the provided address if CX = 0

**Interrupt Instructions**

These instructions are used to call the interrupt during program execution.

* **INT** − Used to interrupt the program during execution and calling service specified.
* **INTO** − Used to interrupt the program during execution if OF = 1

**IRET** − Used to return from interrupt service to the main program