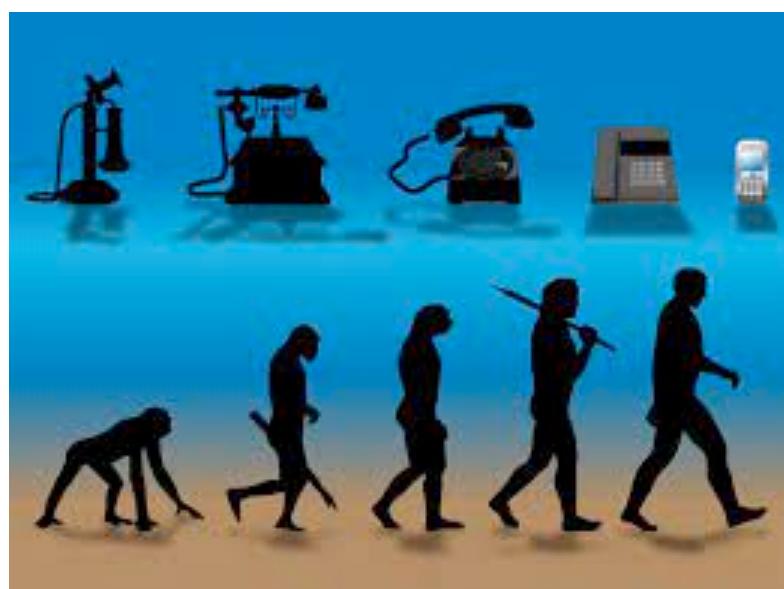


စဉ်	အကြောင်းအရာ	တမျက်နှာ
1	သိပ္ပံ့ဆိုတာဘာလဲ	2
2	Periodic Table and Structure of Atom	3
3	Units System	3
4	Analog System	4
5	Digital System	5
6	Analog System နှင့် Digital Systemတို့၏ ခြားနားချက်များ	5
7	Number System	5
8	Why computer necessary ?	7
9	Processor	9
10	Micorcontroller	11
11	Types of RAMs and ROMs	12
12	Firmware ဆိုတာဘာလဲ	13
13	Operating System	14
14	Difference between OS and Firmware	15
15	Types of Applications	15
16	Compiler	16
17	Interpreter	18
18	Compiler နှင့် Interpreter တို့၏ ခြားနားချက်များ	20

## ၁။ သိပ္ပံ့ဆိုတာဘာလ

သိပ္ပံ့ဟူသည် လောကကြီးတစ်ခုလုံး၏ ဖြစ်တည်မှုနှင့် လူသားတို့၏ ရှာဖွေတွေရှိမှုတို့ကို ပေါင်းစပ်ထားသော ဘာသာဖော်ပို့ရပေမည်။ လောကကြီးသည် သူသောသူဆောင်ပြီး ပြောင်းလဲတိုးတက်သော သဘောတရားရှိပေသည်။ ဥပမာ လူတို့၏ ယနေ့ခေတ် ဦးနောက်အရွယ်အစားသည် ရွှေနှစ်ပေါင်း (၂)သန်းကျော်ကထက် ၃ ဆန်းပို့ကြီးလာသည်ဟု သုတေသနတို့ပြောကြားချက်အရသိရသည်။ ခေတ်များတိုးတက်လာလေလေ သင်ယူစရာများ ပိုများလာလေလေဖြစ်သောပြောင့် လူဦးနောက်သည်လည်း ဆင့်ကဲတိုးတက်ဖြစ်ပေါ်လာရခြင်းဖြစ်ပေသည်။ လောကသာဘာဝ၏ ဆင့်ကဲပြောင်းလဲမှုဖြစ်စဉ်တိုင်း သည် တစ်တို့မှ ပြုပြင်ဖန်တီးထားခြင်းမျိုးမရှိဘဲ သူအလိုလို တိုးတက်ပြောင်းလဲလာခြင်းဖြစ်သည်။ လူသာဘာဝတွင် ပင့်သနေယူချိန် မိခင်ဝစ်းထဲတွင် (၉)လကြာနေရသည်။ မွေးဖားပြီးချိန်တွင်လည်း မိမိအသာ ရပ်တည်နိုင်ဖို့အတွက် မိဘများ၏ အကူအညီကို နှစ်ပေါင်းများစွာယူရသည်။ တချို့သတ္တဝါများကဲ့သို့ မွေးဖားပြီးချက်ချင်း မတ်တပ်ရပ်နိုင်ခြင်း၊ လမ်းလျောက်သွား နိုင်ခြင်းမရှိသောပြောင့် မိခင်၏စောင့်ရောက်မှုကို အပြည့်အဝယူရသည်။ ရင်သွေးငယ်များအပေါ်စောင့်ရောက်မှုအပြည့်အဝရအိုအတွက် မိခင်ဖြစ်သူသည်လည်း သူမ၏ယောကျုံးများထံမှ သို့တည်းမဟုတ် မိသားစုဝင်များထံမှ အကူအညီကို ယူရ ပြန်သည်။ ထို့ကြောင့် မိသားစုကိစ္စရပ်များဖြစ်ပေါ်လာရသည်။ မိသားစုအတွင်း relationship bonding ဖြစ်စေဖို့အတွက် ချစ် ခြင်းမေတ္တာများဖြစ်ပေါ်လာရသည်။ ချစ်ခြင်းမေတ္တာတို့ကြောင့် အောင်မြင်ခြင်း၊ ရှုံးနိမ့်ခြင်းပါယ်များဖြစ်ပေါ်လျက်ရှိပြီး ကမ္မာ လောကကြီးသည် သဘာဝတရား၏ အလိုလိုပြောင်းလဲတိုးတက်လျက်ရှိပေသည်။ ဤအခြင်းအရာအားလုံး တို့သည် Science & Evolution သဘောတရားများပောင်ဖြစ်သည်။ ပုံဌားငုတ်ယောက်များ မြေပြင်ဆင်းသက်နားနေလျှင် ဖိုးရွာမည့် လက္ခဏာဖြစ်ပြီး၊ မြေပြင်ရှိ ပုဂ္ဂက်ဆိတ်များ သစ်ပင်ပေါ် အစုလိုက် အပြုလိုက် တက်နေလျှင် ရေကြီးမည့်လက္ခဏာဖြစ်သည်ဟု လေးလေးခေါ်ခြင်းများအတွက် အကျိုးပြုလျက်ရှိနေပေပြီ။ ဤကဲ့သို့ သဘာဝတရား၏ဖြစ်စဉ်များကို လေ့လာဆန်းစစ်အဖြတ်တိုင်းမှုသည် ပိုမို ကောင်းမွန်သော လောကကြီးဆီ တဖြည့်ဖြည့်ချဉ်းကပ်လာကြသည်မှာ လက်ရှိမော်ဒန်လောကကြီးသို့တိုင် ရောက်ရှိလိုလာပေ သည်။ သိပ္ပံ့ပညာတွင် အကြမ်းဖျော်အားဖြင့် အပိုင်း ၂ ပိုင်းပါဝင်သည်။ Theorization နှင့် Invention အပိုင်းတို့ဖြစ်၏။ Theorization သည် နိုတ်တည်ရှိပြီးသား သဘာဝကို လေ့လာပြီး အတွေးသစ် အမြင်သစ်များအဖြစ် အဖြတ်ခုခုဖော်ထဲတိုင်း ဖြစ်ပြီး Invention သည် သစ်ဆန်းသော အတွေးအခေါ်နှင့် ရှာဖွေတွေရှိထားသော Theory တို့ကို အခြေခံပြီးတိတွင်ဖန်တီးခြင်းဖြစ်သည်။ ထို့ကြောင့်တိတွင်ဖန်တီးမှုများပြုလုပ်ရန်အတွက် Atom ဟုခေါ်သော ဖြပ်စင်များ၏ အခန်းကဏ္ဍအား သိရှိရန် လိုအပ်လာပေတွေ့သည်။



ပုံ(၁) Science & Evolution



## ၃၁။ Derived Units

Fundamental Units များအချင်းအချင်း တွက်ချက်ပြီး ထိုက်လာသော ယူနစ်ကို derived units ဟု ခေါ်သည်။ derived units စုစုပေါင်း (22) မျိုးရှိသည်။ ဥပမာအားဖြင့် သိသာထင်ရှားသော derived ယူနစ်ဖြစ်ပေါ်လာပုံမှာ  $W=FS$  တွင် F (force) ၏ ယူနစ်သည် N (နယူတန်) ဖြစ်ပြီး S (displacement) ၏ ယူနစ်သည် m (မီတာ) ဖြစ်သည်။ ထို့ကြောင့် W (work done) ၏ ယူနစ်မှာ Nm ဖြစ်သည်။

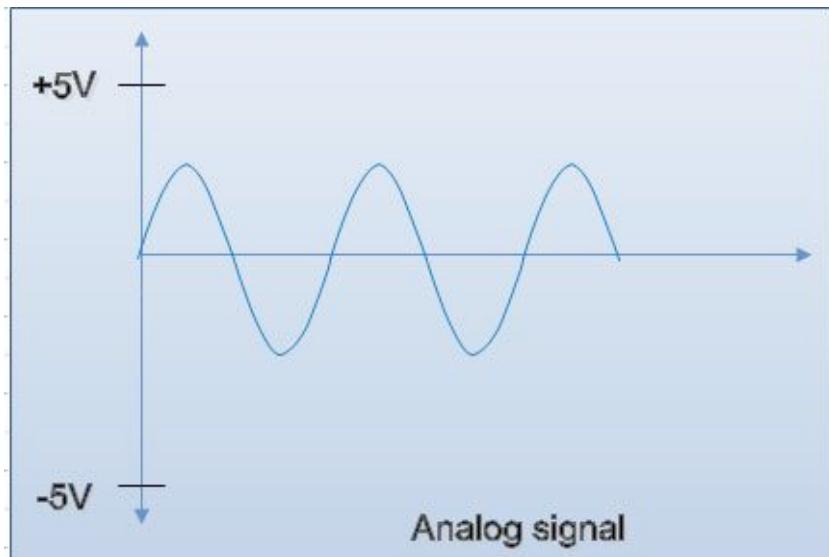
SI derived units with special names and symbols <sup>[2]-[5]</sup>				
Name	Symbol	Quantity	In SI base units	In other SI units
radian <sup>[N 1]</sup>	rad	plane angle	$m/m$	1
steradian <sup>[N 1]</sup>	sr	solid angle	$m^2/m^2$	1
hertz	Hz	frequency	$s^{-1}$	
newton	N	force, weight	$kg \cdot m \cdot s^{-2}$	
pascal	Pa	pressure, stress	$kg \cdot m^{-1} \cdot s^{-2}$	$N/m^2$
joule	J	energy, work, heat	$kg \cdot m^2 \cdot s^{-2}$	$N \cdot m = Pa \cdot m^3$
watt	W	power, radiant flux	$kg \cdot m^2 \cdot s^{-3}$	$J/s$
coulomb	C	electric charge	$s \cdot A$	
volt	V	electric potential, voltage, emf	$kg \cdot m^2 \cdot s^{-3} \cdot A^{-1}$	$W/A = J/C$
farad	F	capacitance	$kg^{-1} \cdot m^2 \cdot s^3 \cdot A^2$	$C/V = C^2/J$
ohm	$\Omega$	resistance, impedance, reactance	$kg \cdot m^2 \cdot s^{-3} \cdot A^{-2}$	$V/A = J \cdot s/C^2$
siemens	S	electrical conductance	$kg^{-1} \cdot m^2 \cdot s^3 \cdot A^2$	$\Omega^{-1}$
weber	Wb	magnetic flux	$kg \cdot m^2 \cdot s^{-2} \cdot A^{-1}$	$V \cdot s$
tesla	T	magnetic flux density	$kg \cdot s^{-2} \cdot A^{-1}$	$Wb/m^2$
henry	H	inductance	$kg \cdot m^2 \cdot s^{-2} \cdot A^{-2}$	$Wb/A$
degree Celsius	$^{\circ}C$	temperature relative to 273.15 K	K	
lumen	lm	luminous flux	$cd \cdot sr$	$cd \cdot sr$
lux	lx	illuminance	$cd \cdot sr \cdot m^{-2}$	$lm/m^2$
becquerel	Bq	activity referred to a radionuclide (decays per unit time)	$s^{-1}$	
gray	Gy	absorbed dose (of ionising radiation)	$m^2 \cdot s^{-2}$	$J/kg$
sievert	Sv	equivalent dose (of ionising radiation)	$m^2 \cdot s^{-2}$	$J/kg$
katal	kat	catalytic activity	$mol \cdot s^{-1}$	

Notes  
1. <sup>a</sup> & <sup>b</sup> The radian and steradian are defined as dimensionless derived units.

## ၄၁။ Derived Units

### ၅။ Analog System

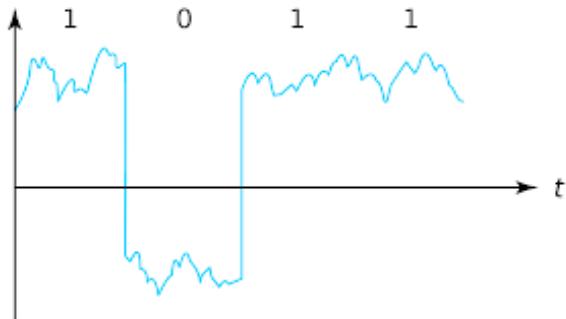
Analog စနစ်သည် လျှပ်စစ် voltageကို step-less manner ဖြင့် စီးဆင်းစေသော အီလက်ထောနစ် စနစ်ဖြစ်သည်။ Analog signal သည် continuous signal ဖြစ်သည်။ Continuous signal ၏ range အတွင်းမှာ analog တန်ဖိုးများဖြစ်ပေါ်ပါသည်။ လူတို့၏အသံ၊ သာမိမိတာ စသည်တို့သည် analog စနစ်များဖြစ်ပါသည်။



## ၅၁။ Analog System

## ၅။ Digital System

Analog နှင့်မတူသောအချက်မှာ Digital signal သည် continuous (စဉ်ဆက်မပြတ်စီးဆင်းမှု) မဟုတ်ဘဲ discrete values များဖြစ်နေခြင်းပင်ဖြစ်သည်။ တန်ဖိုးများကို binary number များဖြင့် ဖော်ပြပြီး မတူညီသော voltage များ ပါဝင်သည်။ 1 ဆိုပါက high voltage ဖြစ်ပြီး 0 ဆိုပါက low voltage ဖြစ်သည်။



ဗု(၆)Digital Signal

## ၆။ Analog System နှင့် Digital System တို့၏ ခြားနားချက်များ

- (c) Analog signal သည် continuous signal ဖြစ်ပြီး Digital signal သည် discrete signal ဖြစ်သည်။
- (j) Analog signal ၏ လိုင်းပံ့သဏ္ဌာန်ကို sine waves ဖြင့် ဖော်ပြပြီး Digital signal ၏ လိုင်းပံ့သဏ္ဌာန်ကို square waves ဖြင့် ဖော်ပြသည်။
- (r) လူတို့၏အသံ၊ သဘာဝအသံနှင့် analog အီလက်ထရောနစ်ပစ္စည်းများမှ ထုတ်ပေးသော signal များအားလုံး သည် analog signal များဖြစ်ပြီး computer, optical driver တို့မှ ထုတ်လွှတ်သော signal များသည် digital များ ဖြစ်ပါသည်။

## ၇။ Number System

Number System (၄) မျိုးရှိသည်။ ငါးတို့မှာ -

- (c) Decimal number system (Base- 10)
- (j) Binary number system (Base-2)
- (r) Octal number system (Base-8)
- (c) Hexadecimal number system (Base-16)

### ၇.၁။ Decimal number System

Decimal number system သည် 0 to 9 အကြော်ဖြစ်သောကြောင့် base 10 number system ဟု ခေါ်ပါသည်။ ဥပမာ 1475 ဟူသော ကိန်းကဏ္ဍးတစ်ခု၏ decimal တန်ဖိုးမှာ အောက်ပါအတိုင်းဖြစ်သည်-

$$= (1 \times 10^3) + (4 \times 10^2) + (5 \times 10^1) + (7 \times 10^0)$$

$$= (1 \times 1000) + (4 \times 100) + (5 \times 10) + (7 \times 1)$$

$$= 1000 + 400 + 50 + 7$$

$$= 1457$$

## ၇.၂။ Binary number system

Binary number system သည် 0 နှင့် 1 ကို အခြေပြုထားသော number system ဖြစ်သောကြောင့် base 2 Number system ဟု ခေါ်ပါသည်။ decimal number တွင် ကိန်းကဏ္ဍးတစ်ခု၏ ပေါင်းလဒ်တစ်ခုချင်းစီကို 10 နှင့် မြောက်ရသကဲ့သို့ binary number စနစ်တွင် 2 နှင့် မြောက်ပေးရသည်။ ထို့ကြောင့် 0 ကနေ 10 အထိ binary တန်ဖိုးအား ရေတွက်ရလျှင် 512, 256, 128, 64, 32, 16, 8, 4, 2, 1 ဖြစ်ပါသည်။

Binary to Decimal											
2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	
1024	512	256	128	64	32	16	8	4	2	1	
				1	0	1	1	0	0	1	
$1011001_2 = 1 \times 64 + 1 \times 16 + 1 \times 8 + 1 \times 1 = 89$											

ံ(၇) Binary to Decimal

## ၇.၃။ Octal number system(8 base)

Octal number system သည် 0 to 7 ပါဝင်သော 8 base number system ဖြစ်ပါသည်။ Octal number ကိုအောက်ပါအတိုင်း ဖော်ပြနိုင်ပါသည်-

$$\text{ဥပမာ (၁)} \quad (347)_8 = (3 \times 8^2) + (4 \times 8^1) + (7 \times 8^0)$$

$$\text{ဥပမာ (၂)} \quad (121)_8 = (1 \times 8^2) + (2 \times 8^1) + (1 \times 8^0)$$

$$= 1 \times 64 + 2 \times 8 + 1 \times 1$$

$$= 64 + 16 + 1$$

$$(121)_8 = (81)_{10}$$

## ၇.၄။ Hexadecimal number system (Base-16)

Hexadecimal number သည် 0 to 15 ပါဝင်သော base-16 number စနစ်ဖြစ်ပါသည်။

HEXADECIMAL NUMBER SYSTEM TABLE

Decimal Numbers	4-bit Binary Number	Hexadecimal Number
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

ံ(၈) Hexadecimal number system



### ၈.၁။ System of Computer Architecture

ကွန်ပူ။တာဟူသည် အခြေခံအားဖြင့် အီလက်ထရောနစ်ပစ္စ်းများဖြင့်ဖွဲ့စည်းထားသည်။ လူသားတို့အတွက် အတိုင်းအဆမဲ့လုပ်အားနှစ်းကို ဖြစ်ပေါ်စေရန်အတွက် ကွန်ပူ။တာသည် လောက်၍ မရှိမဖြစ်လိုအပ်ပေသည်။ ကွန်ပူ။တာတွင် Input Unit, Output Unit, Storage Unit , Arithmetic Logic Unit နှင့် Control Unit ဟူ၍ architectures ၅ လွှာဖြင့် စုစုပေါင်းသည်။

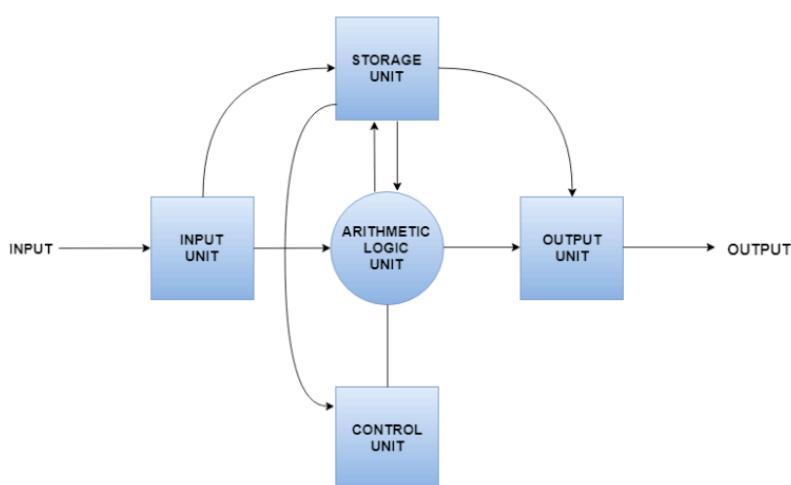
(၁) **Input Unit** ။ Input Unit သည် external state မှ ထည့်သွင်းရသောအပိုင်းဖြစ်သည်။ input data ကို machine language အဖြစ်ပြောင်းပြီး ကွန်ပူ။တာစနစ်ပေါ်သို့ တင်ဆောင်အလုပ်လုပ်စေသည်။ ကွန်ပူ။တာစနစ်၏ input devices များမှာ mouse နှင့် keyboards များဖြစ်သည်။

(၂) **Output Unit** ။ Output Unit သည်လည်း external state သို့ result ထုတ်ပြခြင်းဖြစ်သည်။ သိသာထင်ရှားသော Output data များမှာ video နှင့် audio များပင်ဖြစ်သည်။ printers, speakers, monitors နှင့် headphones တို့သည် output devices များဖြစ်သည်။

(၃) **Storage Unit** ။ Storage Unit သည် data များကို သို့လောင်သောအပိုင်းဖြစ်သည်။ Storage Unit တွင် primary storage unit နှင့် secondary storage unit ဟူ၍ အပိုင်း၂ ပိုင်းပါဝင်သည်။ Primary Storage သည် ကွန်ပူ။တာ၏ အဓိက သို့လောင်ရုံဖြစ်ပြီး data များကို CPU မှ တိုက်ရိုက်ခိုင်းစေပါသည်။ Secondary storage သို့တည်းမဟုတ် external storage ကိုမှ CPU မှ တိုက်ရိုက်ခိုင်းစေခြင်းမျိုးမဟုတ်ပေ။ အကယ်၍ CPU ပေါ်တွင် အလုပ်လုပ်စေလိုပါက main storage ဆီသို့ မဖြစ်မ နေ ရွှေပြောင်းပေးရန်လိုအပ်ပေသည်။ Secondary Storage တွင် ပမာဏကြီးမားသော data များစွာကို ရာသက်ပန်သိမ်းဆည်းထားနိုင်ပါသည်။

(၄) **Arithmetic Logic Unit** ။ ကွန်ပူ။တာစနစ်၏ ပေါင်း၊ နှုတ်၊ မြောက်၊ စား တွက်ချက်မှုအားလုံးကို arithmetic logic unit မှ လုပ်ဆောင်ပေးခြင်းဖြစ်သည်။ တွက်ချက်ရန် လိုအပ်လာသောအခါ control unit သည် storage unit မှ data များကို arithmetic logic unit ဆီသို့ ပြောင်းရွှေပေးသည်။ CPU တွင် Control Unit နှင့် Arithmetic Logic Unit တို့သည် အတူတကွဲဖွဲ့စည်းထားပေသည်။

(၅) **Control Unit** ။ ကွန်ပူ။တာစနစ်အားလုံးကိုထိန်းချုပ်သော Unit ဖြစ်သဖြင့် ကွန်ပူ။တာ၏ပင်မအာရုံးကြောစနစ်ဟု လူသိများလေသည်။ ကွန်ပူ။တာစနစ်မှ လိုအပ်သောအခါ Storage Unit မှ data များကို CPU သို့ ပြောင်းရွှေပေးပြီး ကွန်ပူ။တာ စနစ်တွင် သိမ်းဆည်းထားချင်သောအခါ data များကို CPU မှ Storage Unit သို့ ပြောင်းရွှေပေးသော တာဝန်ဝေါဘာများကို လုပ်ဆောင်ပေးသည်။



ပုံ(၁၀) System of Computer Architecture

## ၉။ Processor

Processor ၏ ပင်မတာဝန်မှာ input devices များမှ ထည့်သွင်းသော အချက်အလက်ကို တွက်ချက်ပြီး output devices များသို့ မှန်ကန်တိကျစွာ ထုတ်ပြန်ပေးခြင်းပင်ဖြစ်သည်။

### ၉.၁။ Types of Processor

Processor အမျိုးအစား ၅ မျိုးရှိသည်။ ငါးတို့မှာ အောက်ပါအတိုင်းဖြစ်ပါသည်-

(၉.၁.၁)။ **Single Core Processor** ။ Single Core Processor သည် microprocessor တမျိုးဖြစ်သည်။ နာရီလ က်တံ့တပတ်ပြည့်အခိုန်တွင် fetch decode တစ်ကြိမ်ပြုလုပ်နိုင်ပြီး တရာ့နှင့် Thread တစ်ခုကိုပဲ ပြုလုပ်နိုင်သည်။ Single Core Processor သည် Multi-Core Processor များထက် လုပ်ဆောင်မှုများစွာနေ့ကွွဲပါသည်။



ဥ (၁၁) Single core processor

(၉.၁.၂)။ **Dual Core Processor** ။ Processing Unit နှစ်ခု သို့မဟုတ် နှစ်ခုထက်ပိုပါဝင်သော micro-processor တမျိုးဖြစ်ပါသည်။ တရာ့နှင့်တွေ့တွင် Core နှစ်ခုကို တပြုပါသည်။ Single Core ထက် များစွာ မြန်ဆန်မှုရှိပါသည်။



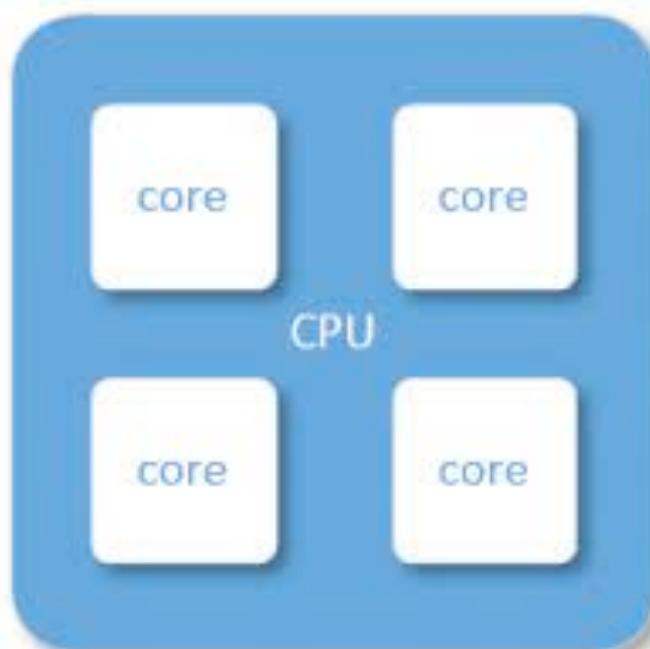
ဥ(၁၂) Dual Core Processor

(ပ.၁.၃)။ **Multi Core Processor**။ Multi core processor ဟူသည် IC တစ်ခုထဲတွင် Processing unit နှစ်ခု သို့မဟုတ် နှစ်ခုထက်ပိုပါဝင်သော cores ကို ခေါ်ဆိုသည်။ ငါး cores အသီးသီးတို့သည် CPU ၏ ညွှန်ကြားချက်များကို ကောင်းမွန်စွာ လုပ်ဆောင်ရှိနိုင်ကြသည်။



ပုံ(၁၂) Multi core processor

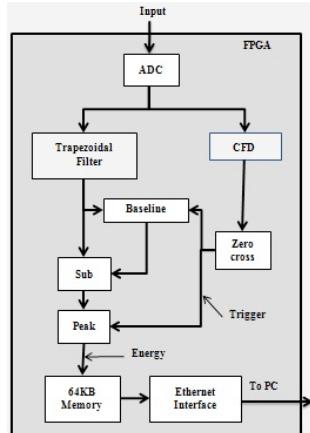
(ပ.၁.၄)။ **Quad Core Processor**။ Chip တစ်ခုထဲတွင် processor cores ၄ ခုပါဝင်ပြီး ငါးတို့အားလုံးတို့သည် CPU ၏ ညွှန်ကြားချက်ကို တဆိတ်တည်းတပြီးတည်းလုပ်ဆောင်ရှိနိုင်စွမ်းရှိသော ပရီဆက်ဆာကို Quad core processor ဟု ခေါ်သည်။



Quadruple-core CPU

ပုံ(၁၃)Diagram of Quad core Processor

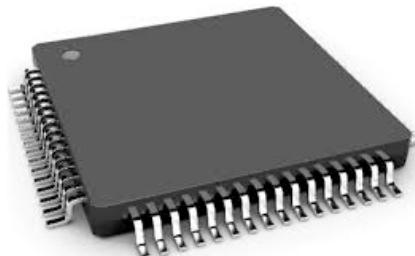
**(၁၁.၅) Octa Core Processor** || Octa Core Processor သည် Quad core processor နှစ်လုံးပေါင်းထားသည်ဟု အကြမ်းဖျင်းပြောနိုင်သည်။ Octa Core Processor သည် cores ငဲ ခုပါဝင်သော Chip နှစ်ခုကို ပြုပိတ်ဆင်ထားသည်။ Quad core Processor ထက် အဆပေါင်းများစွာ မြန်ဆန်ပြီး တဗြားသော မည်သည့် Processing systemနှင့်မဆို ယဉ်တိုးလုပ်ဆောင် နိုင်စွမ်းရှုပေါ်လေသည်။



ပုံ(၁၄) Octa Core Processor

## ၁၁။ Microcontroller

Microcontroller သည် VLSI (Very large scale Integration) IC (Integrated Circuit) ပေါ်က သေးငယ်သော ကွန်ပူးကာတစ်ခုပင်ဖြစ်သည်။ Microcontroller တစ်ခုတွင် CPUs တစ်ခု သို့မဟုတ် တစ်ခုထက်ပိုပြီး ပါဝင်နိုင်သည်။



ပုံ(၁၅) Microcontroller

## ၁၁.၁။ Difference between Microcontroller and microprocessor

(၁) Microprocessor တစ်ခုတွင် CPU တစ်မျိုးတည်းသာပါရှိပြီး Microcontroller တစ်ခုတွင် CPU, RAM, I/O devices များပါဝင်ပါသည်။

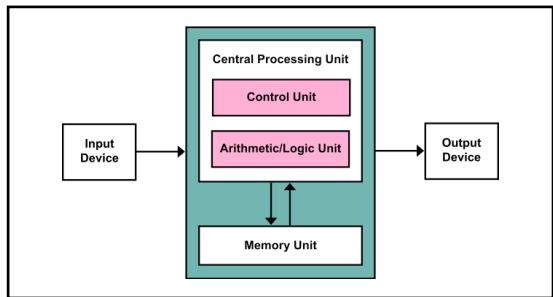
(၂) Microprocessor ကို Personal Computerများတွင် အသုံးပြုပြီး Microcontroller ကို embedded system များတွင် အသုံးပြုသည်။

(၃) Microprocessor တွင် RAM, ROMများကို external bus များဖြင့် အသုံးပြုပြီး Microcontroller တွင် RAM, ROMများကို chipထဲတွင် ထည့်သွင်းအသုံးပြုသည်။

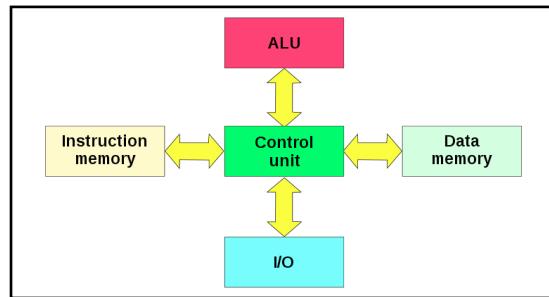
(၄) Microprocessor သည် Von Neumann model ကို အခြေခံထားပြီး၊ Microcontroller သည် Harvard architecture ကို အခြေခံထားသည်။ Von Neumann model သည် ၁၉၄၆ ခုနှစ်တွင် ဟန်ဂရာ-အမေရိကန်နယ်များ သံ့ချာဆရာ Jonh Von Neumann မှ ချမှတ်ပေးထားသော model ဖြစ်သည်။ ငါး model တွင် Control Unit, Arithmetic and Logical ,

Memory Unit (ALU) , I/O registers များ ဖြင့် ဖွဲ့စည်းထားလေသည်။ Harvard architecture သည်လည်း computer architecture ဖြစ်ပြီး Program instructionနှင့် storage တိုကို chip တစ်ခုထဲတွင် စုစည်းအသုံးပြုထားလေသည်။

(၂) Microprocessor သည် ရှုပ်ထွေးသော ညွှန်ကြားချက်များစွာကို ပြုလုပ်နိုင်ပြီး ကုန်ကျစရိတ်ကြီးမားလေသည်။ Microcontroller မှာ ညွှန်ကြားချက်များစွာမပြုလုပ်နိုင်ဘဲ instruction တစ်မျိုးကိုပဲ လွယ်ကူရှင်းလင်းစွာပြုလုပ်နိုင်သည်။ Microprocessor ထက် ကုန်ကျစရိတ် များစွာသက်သာသည်။



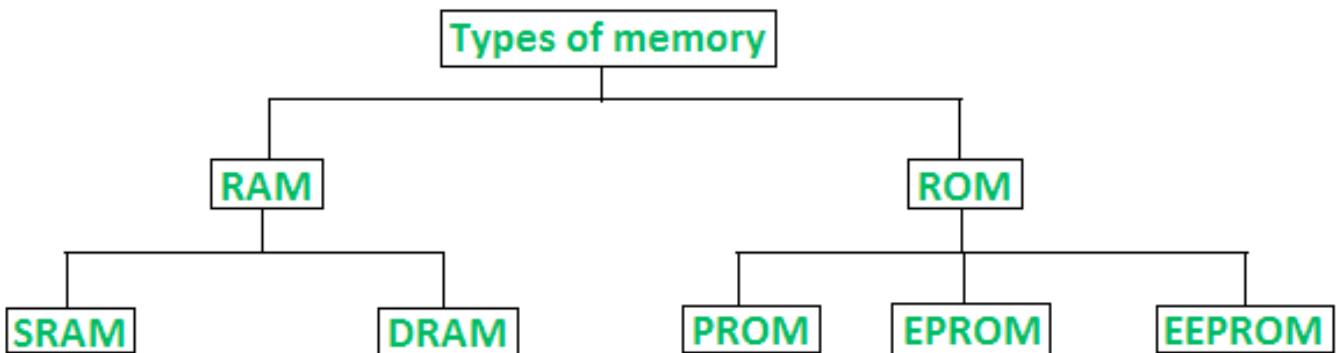
ပုံ(၁၆) Von Neumann model



ပုံ(၁၇) Harvard Architecture

## ၁၁။ Types of RAMs and ROMs

ကွန်ပျူောတွင် မှတ်ညာ၏(memory)သည် အကွဲရာကျသော အစိပ်အပိုင်းတစ်ခုဖြစ်သည်။ Memory တွင် main memory သို့မဟုတ် primary memory နှင့် secondary memory ဟူ၍ အပိုင်း (၂) ပိုင်းပါဝင်သည်။ RAM နှင့် ROM သည် primary memory တွင်ပါဝင်ပြီး secondary memory တွင် harddrive, CD တို့ပါဝင်ပါသည်။

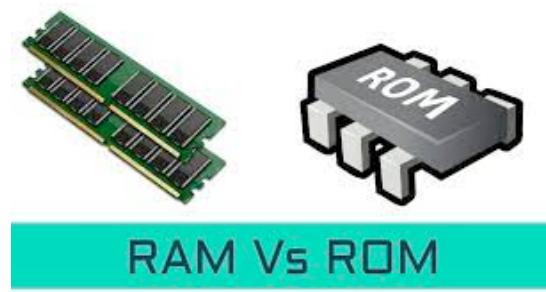


## Classification of computer memory

ပုံ(၁၈) Classification of Computer memory

### ၁၁.၁။ RAMs နှင့် ROMs တို့၏ ကွဲပြားခြားနားချက်များ

- (၁) RAM သည် ယာယိ storageဖြစ်ပြီး ROM သည် permanent storage ဖြစ်သည်။
- (၂) RAM တွင် ဒေတာများကို MBsအနေဖြင့် သိမ်းပြီး ROM တွင် GBs အနေဖြင့် သို့လောင်သည်။
- (၃) RAM တွင် ဒေတာများကို volatile memory (power shutdown ချလျင် ပျောက်ဆုံးနိုင်) အဖြစ်သိမ်းဆည်းပြီး ROM တွင် ဒေတာများကို non-volatile memory အဖြစ် သိမ်းထားသည်။
- (၄) RAM သည်ကွန်ပျူးတာ ဖွင့်ပြီး ပုံမှန်လုပ်ရှိလုပ်စဉ်အခြေအနေများတွင် အလုပ်လုပ်ပြီး ROM သည် ကွန်ပျူးတာဖွင့်ချင်း startup process တွင် အလုပ်လုပ်သည်။
- (၅) ဒေတာသိမ်းရာတွင် RAM က ROM ထက် ဂိုမြန်ဆန်သည်။



ပုံ(၁၉)RAM နှင့် ROM

### ၁၂။ Firmware ဆိုတာဘာလ

Firmware ဟူသော အသုံးအန္တားကို ၁၉၆၇ ခုနှစ်တွင် Ascher Opler ဆိုသူ သို့ပုံပညာရှင်မှ စတင်သုံးခဲ့သည်။ Firmware ဆိုသည်မှာ microcontroller/ microprocessor များပါဝင်သော Electronic devices များတွင် ထည့်သွင်းအသုံးပြုသော Computer program တစ်ခုပင်ဖြစ်သည်။ Computer program ဖြစ်သောကြောင့် CPU, Memory နှင့် Storage များ ပါဝင်ပါသည်။ Firmware ကို ခကေခက ပြင်ဆင်ဖို့ရန် မလိုအပ်ပေါ့၊ ထို့ကြောင့် Firmware ကို ROM, EEPROM နှင့် Flash Memoryပေါ်တွင် သိမ်းထားလေ့ရှုပါသည်။ Firmware သည် Hardwareနှင့် အနီးကပ်ဆုံးထိတွေ့ရသော program ဖြစ်သည့်အားလုံး၊ Machine language နှင့် အနီးကပ်ဆုံး programming language ဖြစ်သော Assembly language နှင့် ရေးသားခဲ့ကြပြီး ယခု အချိန်တွင် C language နှင့် ပိုမိုရေးသားကြပေသည်။



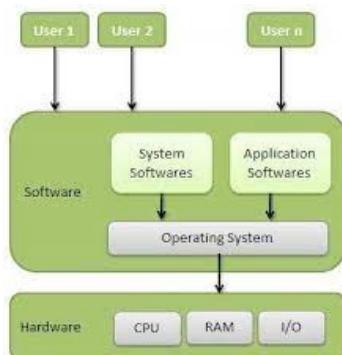
ပုံ(၂၀)Firmware

## ၁၃။ Operating System

Operating System သည် ကွန်ပျူတာ၏ မရှိမဖြစ် ဆောင်တစ်ခုဖြစ်သည်။ Windows, Linux, Mac စသော ဆောင်အားလုံးသည် OS များဖြစ်ပါသည်။ ငါး OS များသည် platforms များ ကဲပြားသော်လည်း အလုပ်လုပ်ပုံနှင့် ပေးရသော services များမှာ အားလုံးနှင့်ပါး ဆင်တူမှုရှိသည်ဟု ဆိုနိုင်ပါသည်။ OS ကို အကြမ်းဖျဉ်းအားဖြင့် User Programs နှင့် Hardwares များ အကြား Interface(ကြားခံဆောင်ရွက်သော) အနေဖြင့် အသုံးပြုသည်ဟု ဆိုရပေမည်။ User Programs ဆိုသည်မှာ OS ပေါ်တွင် run သမျှ ဆောင်များကို ဆိုလိုပါသည်။ Operating system ၏ အဓိက လုပ်ငန်းတာဝန်များမှာ-

- (၁) Process management
- (၂) Main memory management
- (၃) File management
- (၄) IO System management
- (၅) Secondary management
- (၆) Protection system
- (၇) Interprocess communication
- (၈) Networking စသည်များ ပါဝင်သည်။

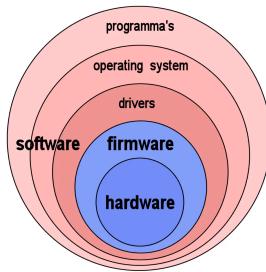
အထက်ပါ အချက်(၈)ချက်၏ အလုပ်လုပ်ပုံမှာ- OS သည် VLC Player တစ်ခုကို run တော့မည်ဆိုပါစိုး။ ငါး VLC player ကို run နိုင်အောင် vlc.exe ကို ဦးစွာပထမ memory ပေါ်သို့ တင်လိုက်ပါသည်။ ထိုသို့တင်နိုင်ရန် Harddisk အတွင်းရှိသော vlc.exe ကို ရှာဖွေဖို့လိုအပ်လာသည်။ **File Management Module** သည် Harddisk ၏ မည်သည့်နေရာ၊ မည်သည့်sector၊ မည်သည့် track ၏ vlc.exe တည်ရှိသည်ကိုသိပါသည်။ တွေ့ပြီဆိုပါက ငါး File ကို read လိုက်ပါသည်။ File read သော တာဝန်ကို **IO System Management Module** က လုပ်ပါသည်။ လူတွင်အတွေးမြန်ပြီး အလုပ်ကြာသကဲ့သို့ OS တွင်လည်း IO ကြာပြီး CPU က မြန်ပါသည်။ IO မှ ဖိုင်ဖတ်ခြင်းအလုပ်ကို CPU မှ ထိုင်စောင့်နေရလျှင် အလုပ်မတွင်ပဲ ရှိနေပေလိမ့်မည်။ IO ကြာနေချိန် CPU က တဗြား process များကို လုပ်ဆောင်ရန်လိုအပ်ပေသည်။ ဤ issue ကို ပြောလည်ဖို့ရန်အတွက် **Process Management Module** က တာဝန်ယူဖြေရှင်းပေးပါသည်။ Process တစ်ခုနှင့်တစ်ခုသည် memory ပေါ်တွင် ရောနောနရာမယူဘဲ အသီးသီးနေရာယူပါသည်။ အဘယ့်ကြောင့်ဆိုသော memory overitting မဖြစ်စေရန် **Security Protection System** လုပ်ထားခြင်းဖြစ်ပါသည်။ Process တစ်ခုမှ memory address ထုတ်ပေးပါက ငါး address သည် relative address အဆင့်ပဲ ရှိနေသေးသည့်အတွက် hardware physical address အဖြစ်ပြောင်းလဲပေးရသည်။ ဤတာဝန်ကို **Main Memory Management Module** က လုပ်ပေးပါသည်။ မည်သည့် Process ကို memory ပေါ်တင်ပြီး မည်သည့် Process ကို harddisk memory ပေါ်တင်မလဲဟုသော process များကို **Main Memory Management Module** က algorithm များဖြင့် တွက်ချက် လုပ်ဆောင်ပေးပါသည်။ ဥပမာ အသုံးများသော Process ကို Memory ပေါ်ထားပြီး အသုံးနည်းသော Process ကို Harddisk ပေါ် ပြောင်းပါမည်။ Process တစ်ခုက ဖိုင်တစ်ခုကို ဖတ်နေချိန် အခြားProcess တစ်ခုက ဤဖိုင်ကို ဖတ်မရအောင် ကာထားရသကဲ့သို့ ဖိုင်ဖတ်ပြီးပြုဆိုကြောင့်ကို အကြောင်းကြားပေးရသည်။ ထို့ကြောင့် Process အချင်းချင်း intercommunicate လုပ်ခြင်းကို **Interprocess Communication Module** က လုပ်ဆောင်ပေးပါသည်။ **Networking Module** မှာ TCP/IP Networking လုပ်ငန်းများ ဆောင်ရွက်ပေးပါသည်။



ပုံ(၂) Operating System

## ၁၄။ Difference between OS and Firmware

- OS နှင့် Firmware တို့၏ခြားများမှာ အောက်ပါအတိုင်းဖြစ်ပါသည်-
- OS ကို ကွန်ပျူဗာ၏ storage device တွင် သိလျှောင်ပြီး firmware ကို ROM တွင် သိလျှောင်ပါသည်။
  - OS သည် fixed codes များမဟုတ်သောကြောင့် update ခနေနဲ့လုပ်ပေးရန်လိုအပ်ပါသည်။ Firmware သည် OS ကဲ့သို့မဟုတ်ဘဲ ငှုံးအား fixed code အဖြစ် ရေးသားထားသောကြောင့် ခနေနဲ့update လုပ်ပေးရန်လည်း မလိုအပ်ပေ။
  - OS တွင် programming codes ပေါင်း မြောက်များစွာ ပါရှိသည့်အတွက် Programming codes အနည်းငယ်ပါရှိသော Firmware ထက် ဖိုင်ဆိုဒိုကြီးပါသည်။
  - OS သည် high-level taskများဖြေရှင်းနိုင်ပြီး Firmware သည် low-level task ဖြေရှင်းရန်အတွက် ဖြစ်သည်။
  - OS သည် ကွန်ပျူဗာ၏လုပ်ငန်းစဉ်အားလုံးကို တရာ့နိတည်းတပြုင်တည်း လုပ်ဆောင်နိုင်ပြီး Firmware သည် တရာ့နိတွင် task တစ်ခုတည်းကိုလုပ်ဆောင်ပါသည်။



၁၅။ Difference between OS and Firmware

## ၁၆။ Types of Applications

ကွန်ပျူဗာ၏ OS ပေါ်တွင် တင်run နိုင်သော ဆောင်မှန်သမျှကို Applications ဟု ခေါ်ပါသည်။ ဆောင်တွင် System software နှင့် Application software ဟူ၍ရှိပါသည်။ Applications Software ကို General Applications , Business Applications, Applications Based on Shareability ဟူ၍ အမျိုးအစား၏ မျိုးထပ်ခွဲနိုင်ပါသည်။

### (၁၆.၁) General Applications များ

- (a) Word Processing Software
- (b) Graphics Software
- (c) Spreadsheet Software
- (d) Presentation Software
- (e) Web Browsers
- (f) Multimedia Software
- (g) Education and Reference Software
- (h) Simulation Software

### (၁၆.၂) Business Applications များ

- (a) Customer Relationship Management Application Software
- (b) Enterprise Resource Planning Application Software
- (c) Project Management Application Software
- (d) Business Process Management Application Software
- (e) Educational Software

### (၁၆.၃) Applications Based on Shareability

- (a) Freeware

- (b) Shareware
- (c) Open source
- (d) Closed source

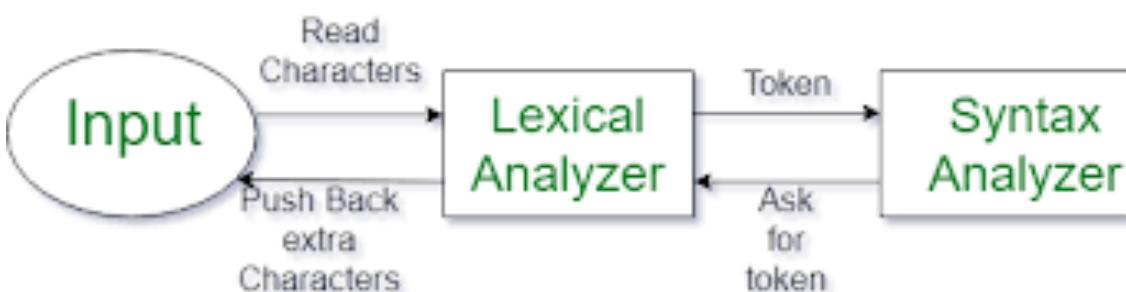
## ၁၆။ Compiler

Compiler ဆိုသည်မှာ High level programming language ဖြင့်ရေးသားထားသော source codes များကို အသစ် ပြင်ဆင်နှုန်းမရှိဘဲ ရှိရင်းအပိုပွယ်အတိုင်း ကွန်ပျူးတာနားလည်နိုင်သော Low level language အဖြစ်ပြောင်းလဲပေးသော program တစ်ခုဖြစ်ပါသည်။ Compiler အလုပ်တစ်ခု အောင်မြင်စွာ လုပ်ဆောင်နိုင်ရန်အတွက် အောက်ပါလုပ်ငန်းစဉ်များကို မဖြစ်မနေ လုပ်ဆောင်ရပါသည်-

- (က) Pre-processing
- (ဂ) Lexical Analysis
- (ဃ) Syntax Analysis
- (င) Semantic Analysis
- (စ) Code Optimization
- (ဆ) Code Generation

**(၁၆.၁) Preprocessing** ။ Compiler သည် source code တစ်ခုလုံးအား compile မလုပ်ခင် source codes အစအဆုံးကို ဖြင့် ကြည့်လိုက်သည်။ ဤများမည်ပုံစံကိုလုပ်ရမည်ဆုံးဖြတ်သည်။ ထိုနောက် လုပ်ငန်းတာဝန်များကို တစ်ခုချင်း ဆက်လုပ်သည်။ ဤအပိုင်းကို Preprocessing ဟု ခေါ်သည်။ ဥပမာ C/C++ code တွင် header ကြော်သော # သက်ာတသည် Processor statement ဖြစ်ပါသည်။ ငါး၏ အဓိကတာဝန်မှာ macro expansion နှင့် code replacement လုပ်ပေးရန်ဖြစ်သည်။

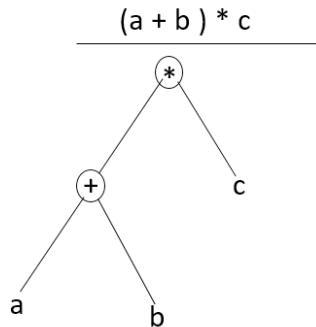
**(၁၆.၂) Lexical Analysis** ။ Compiler ၏ lexical analysis တာဝန်မှာ source codes များကို အပိုင်းလိုက်ခွဲထုတ်ခြင်း နေရာလွတ်များနှင့် comment lines နေရာများကို ဖယ်ထွတ်ပြီး token များအဖြစ် အတွဲလိုက် စီထားရန်ဖြစ်ပါသည်။ အကယ်၍ စနှင့်ကိုက်ညီမှုမရှိသော tokens များ တွေ့ရှုပါက error အဖြစ် ဖော်ပြပေးသည်။



ဗု(၂၃) Lexical Analysis

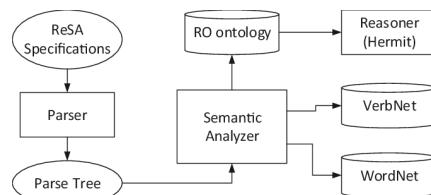
**(၁၆.၃) Syntax Analysis** ။ Lexical analysis လုပ်ပြီးသွားလျှင် ဒုတိယလုပ်ငန်းစဉ်အနေဖြင့် syntax analysis ကို ပြုလုပ်ပါသည်။ Syntax analysis ဆိုသည်မှာ source code များ၏ ဝါကျများကို စီစဉ်ခြင်းဟု ဆိုရပေမည်။ ဤ code ကို မည်သည့် programming language ဖြင့်ရေးထားသနည်း၊ မည်သည့် data structure များဖြင့် တည်ဆောက်ထားသနည်း စသဖြင့်

Syntax tree တနည်းအားဖြင့် Parse Tree နည်းဖြင့် ခွဲခြမ်းစိပ်ဖြောပြီး အလုပ်လုပ်ပါသည်။ Parse tree နည်းဆိုသည်မှာ input string (source codes) များကို grammar နည်းအရ ဖြောက်သတ်မှတ်ပြီး စကားလုံးများကို အသေးကယ်ဆုံးအစိတ်အပိုင်းထိ ခွဲခြမ်းစိပ်ဖြောသော နည်းလမ်းဖြစ်ပါသည်။



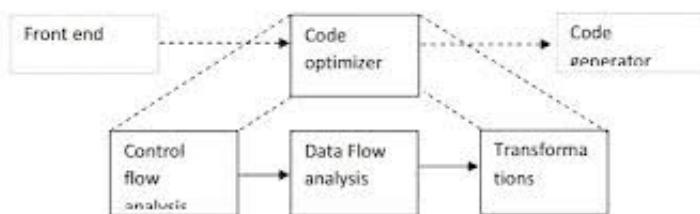
ပုံ(၂၄) Syntax analysis

(၁၆.၅) Semantic Analysis ။ ဤအပိုင်းသည် Syntax analysis လုပ်ပြီး ရရှိလာသော ဝါကျများကို သတ်မှတ်ထားသော grammar formats မှန်ကန်မှုရှိမရှိစစ်ဆေးခြင်းဖြစ်သည်။ Semantic analysis လုပ်ငန်းစဉ်တွင် စကားလုံးများ( words)၊ စကားစုများ(phrases)၊ ဝါကျအပိုဒ်ယေား( clauses) များပါဝင်ပြီး ငင်းတိုးအားလုံးသည် သတ်မှတ်ထားသော programming language ၏ စကားစပ်များ( contexts) နှင့် ကိုက်ညီမှုရှိမရှိ စစ်ဆေးသွားခြင်းပင်ဖြစ်သည်။



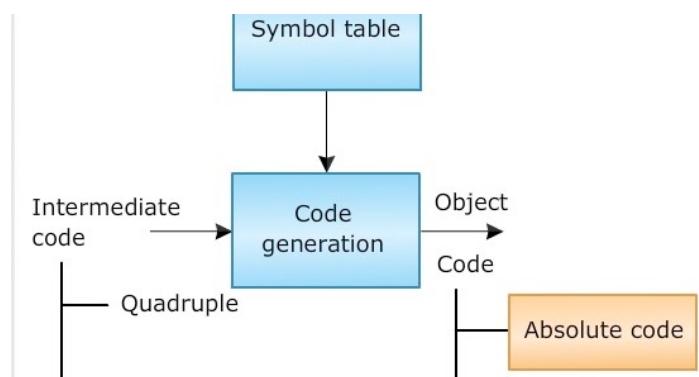
ပုံ(၂၅) Semantic analysis

(၁၆.၆) Code Optimization ။ Compiler ၏ Optimization ပြုလုပ်ခြင်းဆိုသည်မှာ high level source code များကို intermediate code အဖြစ် ပြောင်းလဲပေးခြင်းဖြစ်ပါသည်။ intermediate code ပြောင်းလိုက်သည်နှင့်တဗြိုင်နှင်း CPU ကို ပိုမို မြန်ဆန်စေပြီး RAM အသုံးပြုမှု သေးကယ်သွားစေပါသည်။ ထို့ကြောင့် run-time မြန်ဆန်သော machine code ကို ရရှိလာမည် ဖြစ်ပါသည်။



ပုံ(၂၆) Code Optimization

(၁၆.၆) **Code Generation** || code generationသည် compiler လုပ်ငန်းစဉ်၏ နောက်ဆုံးအဆင့်ဖြစ်ပါသည်။ code generation ပြုလုပ်ခြင်းဆိုသည်မှာ Code Optimization ပြုလုပ်၍ ရရှိလာသော intermediate code များကို ကွန်ပူ။တာ နားလည်နိုင်သော machine code အဖြစ် ပြောင်းလဲပေးသော အဆင့်ဖြစ်ပါသည်။ algorithms မြောက်များစွာကို အသုံးပြု၍ သောကြောင့် code generationပြုလုပ်ခြင်းအဆင့်သည် အခြားလုပ်ငန်းစဉ်များထက် ပိုမိုရှုပ်ထွေးသော လုပ်ငန်းစဉ်များ ရှိလေ သည်။ Code generation Algorithm ဆိုသည်မှာ compiler တစ်ခု၏ အသက်သွေးကြောပင်ဖြစ်ပါသည်။ ငြင်း Algorithmသည် register descriptors နှင့် Address descriptors တို့ကို စီစဉ်ညွှန်ကြားပြီး CPU မှ အဆင်သင့်စားသုံးနိုင်သော machine code များကို generate လုပ်ပေးပါသည်။ Register descriptors ဆိုသည်မှာ program ထဲ၌ အသုံးပြုထားသော register လုပ်ထားသော data structure များကို ဆိုလိုပြီး Address descriptor ဆိုသည်မှာ Program မှ သုံးစွဲထားသော Memory location များကို ဆိုလိုပါသည်။

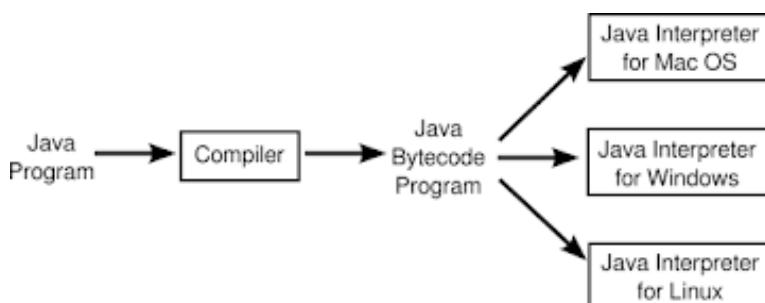


ပုံ(၂၇) Code Generation

## ၁၈။ Interpreter

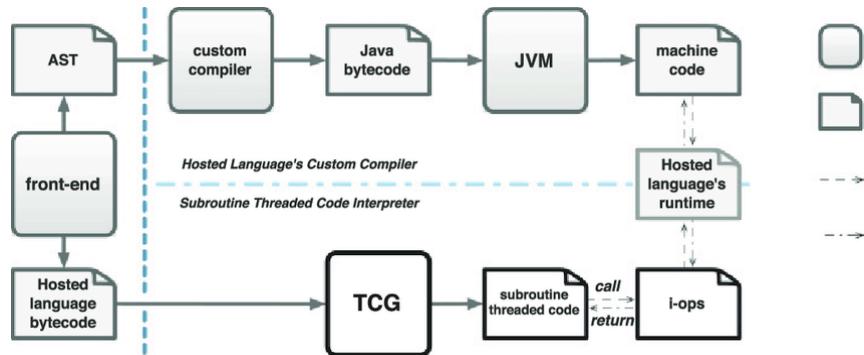
Interpreter ဆိုသည်မှာ High level programming language များကို ကြားခံ machine code အဖြစ်ပြောင်းစရာမလို ဘဲ output တန်းထွက်နိုင်အောင် source code များအား line by line စာ statement by statement တိုက်ရှိက်ဘာသာပြန်ပေး သော computer program တစ်ခုဖြစ်ပါသည်။ Interpreter အမျိုးအစား(၄)မျိုးရှိသည်။ ငြင်းတို့မှာ -

(၁၈.၁) **Bytecode Interpreters** || ဤ interpreter ၏ အလုပ်လုပ်ပုံမှာ Source code များကို byte code သို့ အရင်ပြောင်းပါသည်။ သို့သော် ငြင်း byte code များသည် machine code များမဟုတ်ပေါ့။ ထိုနောက် compile code သို့ထပ်ပြောင်းပါသည်။ Bytecode interpreter တွင် interpreter အဖြင့် compiler ပါ အလုပ်လုပ်ရသောကြောင့် compilers ဟူလည်း ခေါ်သည်။ Bytecode interpreter တွင် instructions တိုင်းကို 1 byte ဖြင့် အစပြုသောကြောင့် instructions ပေါင်း 256 ခု ရှိလေသည်။



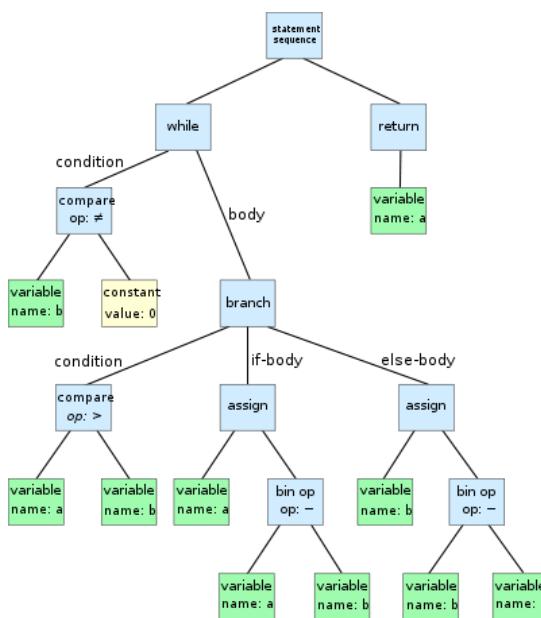
ပုံ(၂၈)Bytecodes and the Python Virtual Machine (VM)

(၁၇.၂) **Threaded Code Interpreters** || Byte code interpreter နှင့်လုပ်ဆောင်ပုံချင်း အလားသဏ္ဌာန်တူညီပါသည်။ သို့သော Threaded Code Interpreters သည် pointer ကို သုံးထားပါသည်။ Instructions တစ်ခုချင်းသည် pointer တစ်ခုဖြစ်ပြီး function တစ်ခု သို့မဟုတ် instruction များကို pointer များဖြင့် ညွှန်ပြပါသည်။ instructions အရေအတွက် ကန့်သတ်ချက်မလိုဘဲ memory နှင့် address နေရာလွှတ်ရှုမရှုကိုသာ ဆုံးဖြတ်ပြီး အလုပ်လုပ်ပါသည်။



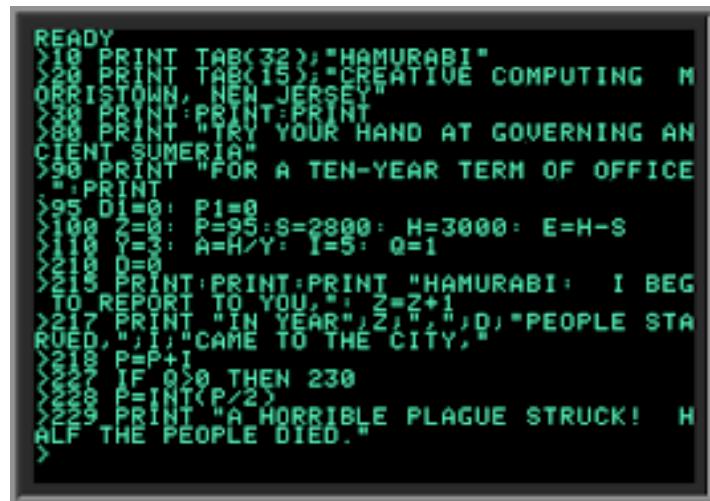
ပုံ(၂၉) Threaded Code Sysytem Overview

(၁၇.၃) **Abstract Syntax Tree Interpreters** || ဤ interpreter ၏ အလုပ်လုပ်ပုံမှာ source code ကို abstract syntax tree (AST) အဖြစ် ဦးစွာပြောင်းပြီး ငွေး အတိုင်း ဝါကျတစ်ကြောင်းလျင် တစ်ကြိမ်နှင့်ပြင် program ကို ဘာသာပြန်ပါသည်။ statements အချင်းချင်းကြားရှိ program structure နှင့် ဆက်သွယ်မှုမှာ ပြောင်းလဲမှုမရှိဘဲ နို့မူလအဓိပ္ပာယ်အတိုင် ဆက်ရှိနေသေစဉ်အပြင် ပိုမိုကောင်းမွန်သော analysis ကို ရရှိပေါ်ပါသည်။



ပုံ(၂၀) Abstract Syntax Tree

(၁၇.၄) **Self Interpreters** || ဤ interpreter အမျိုးအစားမှာ ထူးဆန်းသော interpreter ဖြစ်သည်ဟု ဆိုရပေမည်။ အဘယ့် ကြောင့်ဆိုသော် source code ကို ဘာသာပြန်သော interpreter ကိုယ်တိုင်အား ငင်း source code ကိုရေးသော programming language ဖြင့် ရေးသားထားသောကြောင့် itself interpret ၍ ရနိုင်သောကြောင့်ဖြစ်သည်။ ဥပမာအားဖြင့် BASIC interpreterကို BASIC ဟူသော Programming language ဖြင့် ရေးထားခြင်းဖစ်သည်။



```

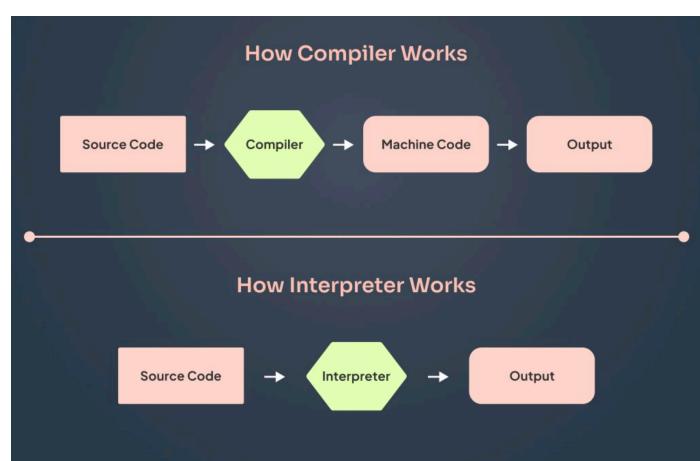
READY
>10 PRINT TAB(32); "HAMURABI"
>20 PRINT TAB(15); "CREATIVE COMPUTING M
  ORRISTOWN, NEW JERSEY"
>30 PRINT:PRINT:PRINT
>40 PRINT "TRY YOUR HAND AT GOVERNING AN
  CIENT SUMERIA"
>50 PRINT "FOR A TEN-YEAR TERM OF OFFICE
  =:PRINT
>60 D1=0: P1=0
>70 Z=0: P=95:S=2800: H=3000: E=H-S
>80 Y=3: A=H/Y: I=5: Q=1
>90 D=0
>100 PRINT:PRINT:PRINT "HAMURABI: I BEG
  TO REPORT TO YOU,"; Z=Z+1
>110 PRINT "IN YEAR"; Z;" ";D;"PEOPLE STA
  RUED,";I;"CAME TO THE CITY,";
>120 P=P+I
>130 IF Q>0 THEN 230
>140 P=INT(P/2)
>150 PRINT "A HORRIBLE PLAGUE STRUCK! H
  ALF THE PEOPLE DIED."
>

```

ပုံ(၃၁)BASIC interpreter

### ၁၈။ Compiler နှင့် Interpreter တို့၏ ခြားနားချက်များ

Programming နယ်ပယ်၌ လူနားလည်သောအရာမှာ high level language ဖြစ်၍ ကွန်ပျူးတာနားလည်သော ဘာသာစကားမှာ low level language ဟုခေါ်သော ၀ နှင့် ၁ ဖြစ်ပါသည်။ Programming language ရေးသားခြင်းဟူသည် လူ တို့အတွက် အနှစ်းမဲ့အကျိုးကော်ဇူးများဖြစ်ပေါ်စေရန်အလိုကြာ ရှုပ်ထွေးပေလီသော ပြဿနာပေါင်းသောင်းပြောင်းထွေလာများ ကို လူအစား ကွန်ပျူးတာအား ခိုင်းစေခြင်းဖြစ်သည်။ ထိုသို့ခိုင်စေရာတွင် ရေးသားလိုက်သော ဘာသာစကားကို ကွန်ပျူးတာ နားလည်နိုင်ရေးဟူသော အခန်းကဏ္ဍာသည် အစိကပြဿနာဖြစ်လာသည်။ ထိုပြဿနာအား ဖြေရှင်းပေးနိုင်သော အရာမှာ Compiler နှင့် Interpreter ဖြစ်၏။ Programming languages များ၏ အလုပ်လုပ်ပုံသည် compiler နှင့် interpreter အပေါ် တွင် ရာခိုင်နှုန်းပြည့် မိခိုအားထားမှုရှိပေရာ Compilerနှင့် Interpreter တို့၏ မတူကွဲပြားများချက်နှင့် ငင်းတို့၏ အားသာချက် အားနည်းချက်များကို သိရှိရန် မဖြစ်မနေလိုအပ်ပေတဲ့သည်။



ပုံ(၃၂) How do they work

ပုံ(၃၂)အရ Compiler သည် source code များကို machine code သို့အရင်ပြောင်းပြီးမှ Output ကိုထုတ်ပေးလေးရှိသော်လည်း Interpreter မှာမူ Source code အား machine code သို့ပြောင်းစရာမလိုဘဲ Output ကိုတန်းပြီးထုတ်ပေးနိုင်ပါသည်။

Compiler ၏သဘောတရားမှာ Source code ၏statements များမှန်ကန်မှုရှိမှုသာ machine code သို့မှန်ကန်စွာပြောင်းလဲပေးနိုင်မှာဖြစ်ပြီး statements များမှန်ကန်မှုမရှိပါက error တက်ပြီး result ကိုထုတ်ပြနိုင်မည်ဟုတ်ပေ။ သို့ရာတွင် interpreter မှာ statement တစ်ကြောင်းချင်းအလိုက် အလုပ်လုပ်သည့်စနစ်ဖြစ်သည့်အားလုံးဖြစ်ပေးနိုင်သော statement တိုင်းအတွက် result ထုတ်ပေးမည်ဖြစ်ပါသည်။

Compiler သည် Program တစ်ပုဒ်လုံးမှန်ကန်မှုရှိစေရန် Machine code သို့အရင်ပြောင်းပြီးအလုပ်လုပ်ရသောကြောင့် run-time မှာ interpreter ထက် ပိုကြန်ကြာမှုရှိပါသည်။ သို့ရာတွင် compile လုပ်ခြင်းကိုတစ်ကြိမ်ပဲ ပြုလုပ်ရသည့်အတွက် compile-time တွင် interpreter ထက် ပိုမြန်ဆန်ပါသည်။ Interpreter မှာ code line တစ်ကြောင်းချင်းစီတိုင်းကို output အဖြစ်ထုတ်နိုင်သောကြောင့် compiler ထက် run-time ပိုမြန်သော်လည်း code lines များများလာလျှင် Compiler ထက် compile-time နေးသွားပါလိမ့်မည်။

Compiler သည် သူ့သတ်မှတ်ချက်စံနှင့်မကိုက်ညီပါက လုံးဝအလုပ်ဆက်လုပ်မှုမဟုတ်တော့သကဲ့သို့ Interpreter သည် သက်ဆိုင်ရာ Language ၏interpreter ရှိမှုသာ အလုပ်ဆက်လုပ်နိုင်မည်ဖြစ်ပါသည်။ ဥပမာအနေဖြင့် compiled language များမှာ C,C++,C#, Java, Kotlin, Scala, Go, Ocami, Rust , etc.. စသည်ဖြစ်ကြပြီး Interpreted languages များမှာ JavaScript, Forth, PHP, PostScript, Ruby, Seed7 , etc.. စသည်တို့ဖြစ်ပါသည်။