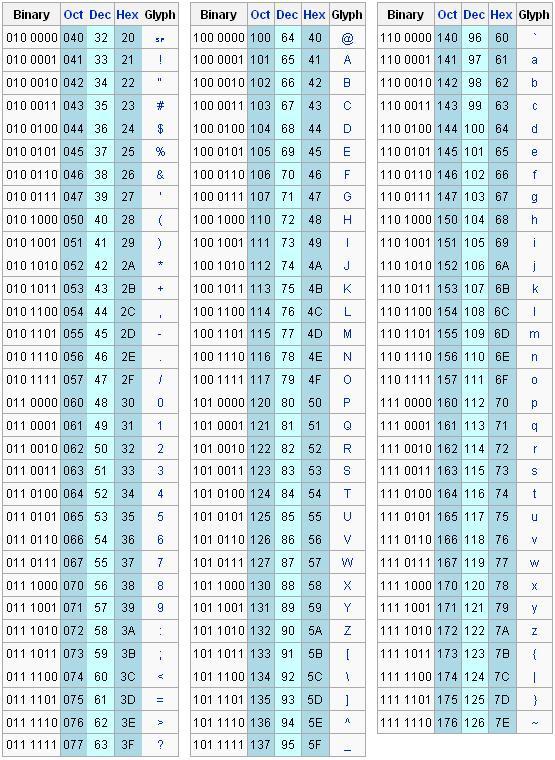
**Nama : Nur Fitria Melani  
NIM : L200180012**

**TUGAS MODUL 1**

1. Apa yang dimaksud dengan kode “ASCII”, buatlah tabel kode ASCII lengkap dengan kode ASCII yang standar tidak perlu extended, tuliskan kode ASCII dalam format angka desimal, binary, hexadesimal serta karakter dan simbol yang dikodekan.
2. Carilah daftar perintah bahasa assembly untuk mesin intel keluarga x86 lengkap(dari buku referensi atau internet). Daftar perintah ini dapat digunakan sebagai pedoman untuk memahami program ‘boot.asm’ dan ‘kernel.asm’.

**JAWAB**

1. Kode **ASCII (**American Standard Codes for International Interchange) adalah kumpulan kode-kode yang dipergunakan untuk mempermudah interaksi antara user dan komputer.



1. Assembly Directive merupakan kode yang menjadi arahan bagi assembler/compiler untuk menata program. Instruksi yaitu kode yang harus dieksekusi oleh CPU mikrokontroler dengan melakukan operasi tertentu sesuai dengan daftar yang sudah tertanam dalam CPU.

**Daftar Assembly Directive**

|  |  |
| --- | --- |
| **Assembly Directive** | **Keterangan** |
| EQU | Pendefinisian konstanta |
| DB | Pendefinisian data dengan ukuran satuan 1 byte |
| DW | Pendefinisian data dengan ukuran satuan 1 word |
| DBIT | Pendefinisian data dengan ukuran satuan 1 bit |
| DS | Pemesanan tempat penyimpanan data di RAM |
| ORG | Inisialisasi alamat mulai program |
| END | Penanda akhir program |
| CSEG | Penanda penempatan di code segment |
| XSEG | Penanda penempatan di external data segment |
| DSEG | Penanda penempatan di internal direct data segment |
| ISEG | Penanda penempatan di internal indirect data segment |
| BSEG | Penanda penempatan di bit data segment |
| CODE | Penanda mulai pendefinisian program |
| XDATA | Pendefinisian external data |
| DATA | Pendefinisian internal direct data |
| IDATA | Pendefinisian internal indirect data |
| BIT | Pendefinisian data bit |
| #INCLUDE | Mengikutsertakan file program lain |

**Daftar Instruksi**

|  |  |
| --- | --- |
| **Instruksi** | **Keterangan Singkatan** |
| ACALL | Absolute Call |
| ADD | Add |
| ADDC | Add with Carry |
| AJMP | Absolute Jump |
| ANL | AND Logic |
| CJNE | Compare and Jump if Not Equal |
| CLR | Clear |
| CPL | Complement |
| DA | Decimal Adjust |
| DEC | Decrement |
| DIV | Divide |
| DJNZ | Decrement and Jump if Not Zero |
| INC | Increment |
| JB | Jump if Bit Set |
| JBC | Jump if Bit Set and Clear Bit |
| JC | Jump if Carry Set |
| JMP | Jump to Address |
| JNB | Jump if Not Bit Set |
| JNC | Jump if Carry Not Set |
| JNZ | Jump if Accumulator Not Zero |
| JZ | Jump if Accumulator Zero |
| LCALL | Long Call |
| LJMP | Long Jump |
| MOV | Move from Memory |
| MOVC | Move from Code Memory |
| MOVX | Move from Extended Memory |
| MUL | Multiply |
| NOP | No Operation |
| ORL | OR Logic |
| POP | Pop Value From Stack |
| PUSH | Push Value Onto Stack |
| RET | Return From Subroutine |
| RETI | Return From Interrupt |
| RL | Rotate Left |
| RLC | Rotate Left through Carry |
| RR | Rotate Right |
| RRC | Rotate Right through Carry |
| SETB | Set Bit |
| SJMP | Short Jump |
| SUBB | Subtract With Borrow |
| SWAP | Swap Nibbles |
| XCH | Exchange Bytes |
| XCHD | Exchange Digits |
| XRL | Exclusive OR Logic |