**NAMA : VENY FITRIANA ISNAINI**

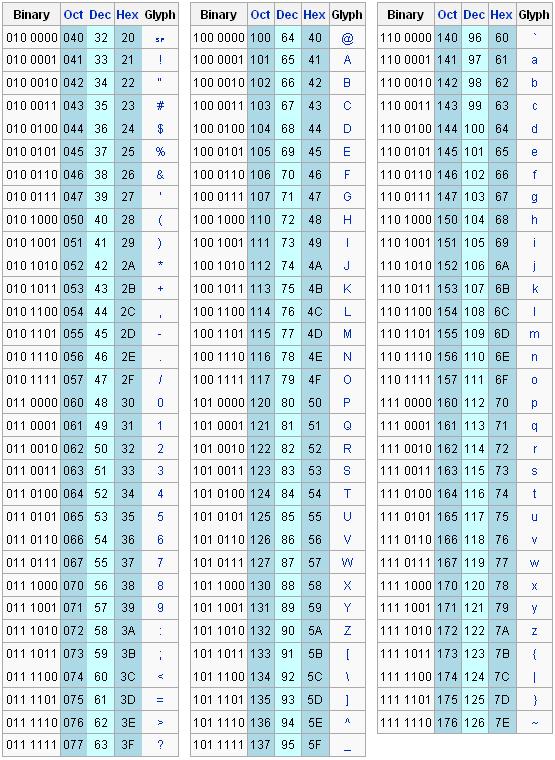
**NIM : L200180045**

**TUGAS!**

1. Apa yang dimaksud dengan kode ‘ASCII’, buatlah tabel kode ASCII lengkap cukup kode ASCII yang standar tidak perlu extended, tuliskan kode ASCII dalam format angka desimal, binary dan 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. ASCII singkatan dari American Standard Code for Information Interchange (Kode Standar Amerika untuk Pertukaran Informasi). ASCII adalah kode-kode yang semuanya terdiri dari 256 kode (kode 0 sampai kode 255). Misalnya, spasi dilambangkan oleh kode 32, angka nol oleh kode 48, huruf A besar (kapital) oleh kode 65, dst.



1. Dalam program bahasa assembly terdapat 2 jenis yang kita tulis dalam program:  
   **1. Assembly Directive** (yaitu merupakan kode yang menjadi arahan bagi assembler/compiler untuk menata program)  
   **2. 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 |

Untuk yang lebih jelas dan detil:  
 **a. MOV**  
 Perintah MOV adalah perintah untuk mengisi, memindahkan,memperbaruhi isi suatu register, variable ataupun lokasi memory, Adapun tata penulisan perintah MOV adalah :  
 MOV [operand A], [Operand B]  
 Contoh :  
 MOV AH,02  
 Operand A adalah Register AH  
 Operand B adalah bilangan 02  
 Hal yang dilakukan oleh komputer untuk perintah diatas adalahmemasukan 02 ke register AH.  
  
 **b. INT (Interrupt)** Bila anda pernah belajar BASIC, maka pasti anda tidak asing lagi dengan perintah GOSUB. Perintah INT juga mempunyai cara kerja yang sama dengan GOSUB, hanya saja subroutine yang dipanggil telah disediakan oleh memory komputer yang terdiri 2 jenis yaitu :  
 - Bios Interrupt ( interput yang disediakan oleh BIOS (INT 0 – INT 1F))  
 - Dos Interrupt ( Interrupt yang disediakan oleh DOS (INT 1F – keatas))  
  
 **c. Push**  
 Adalah perintah untuk memasukan isi register pada stack, dengan tata penulisannya:POP [operand 16 bit]  
  
 **d. Pop** perintah yang berguna untuk mengeluarkan isi dari register/variable dari stack,dengan tata penulisannya adalah : POP [operand 16 bit]  
  
 **e. RIP (Register IP)**  
 Perintah ini digunakan untuk memberitahu komputer untuk memulai memproses program dari titik tertentu.  
  
 **f. A (Assembler)** Perintah Assembler berguna untuk tempat menulis program Assembler.  
 -A100  
 0FD8:100  
  
 **g. RCX (Register CX)**  
 Perintah ini digunakan untuk mengetahui dan memperbaruhi isi register CX yang merupakan tempat penampungan panjang program yang sedan aktif