**MINGGU 2**

Kalau misalnya kita melakukan instruksi ADD, kalau signed terdapat negative maka sign nya akan menjadi 1, terus kalau misalnya hasilnya nol maka zero flag akan ada. Flag kerjanya independent.

Org 0100h artinya aplikasi yang kita buat ditempatkan pada Lokasi 100 pada memory, pada assembly memiliki format tiap baris ada 4 field, pertama adalah label, kedua adalah yg dilakukan / instruksi, keempat adalah comment, ketiga adalah seperti REG, memory atau memory, REG dll yaitu operand. Label selalu ditandai dengan diakhiri “:”, misalnya “awal :”. Instruksi ada 2 jenis yaitu instruksi yang hasilkan operation code yaitu code hexa, selain itu ada assembler directive, ini tidak menghasilkan code hexa (OPCODE), karena tidak dikenal pada microprocessor, ini merupakan sebuah perintah yang dipaham oleh assembler untuk membantu user, misalnya “org” pada 8086 tidak ada karena merupakan assembler directive, “org” merupakan origin yang berarti merupakan awal dari program kita diletakan di sana.

Assembly code :

“jmp start” -> berarti akan jump start, jump ada code yaitu EB dan OE maksudnya adalah 14. Instruksi count (processor akan mengarah ke contoh 102 letaknya var count berada), val. Db berupakan define byte. OA akan memberikan baris yang baru, OD akan digeser ke paling kiri.

mov cx, word ptr count. Ptr itu menganggap count menjadi word yang yaitu melihat data processer, kalau missal processor 64 bit maka 64 bit itu merupakan word. Mov kalau bit tidak sesuai tidak bisa.

Mnemonic maksudnya adalah instruksi yang ada pada microprocessor, db untuk memabntu menentukan jenis datanya.

Setiap baris bisa terdiri dari 1 ato 2 ato 3 bytes.

“0100 : EB 0E jmp start”, pada contoh ini maka jmp start teridir dari 2 bytes, dengan instruksi sama misalnya mov tapi bisa saja bytenya beda. Yang diawal itu merupakan instruction code misalnya “011F: B4 02 mov ah, 02h”, pada contoh ini maka instruction codenya meruapakan B4. Kalau misalnya 00 bukna instruction.

Define byte didefinisikan per 1 byte, jika defined word maka 2 byte dan lainnya.kita menggunakan define byte supaya tidak banyak yang tidak terpakai tetapi disiapkan yang elbih besar untuk yang lebih besar.

“list db 10h, 20h, 41h,2” kalau misalnya ditulis ada h nyam aka dijelaskan bahwa 10 itu hexa dan seterusnya kalau misalnya tapa h misalnya hanya “10,20,41” maka akan menjadi “OA, 14, 29”

DUP, untuk duplicate misalnya “db 20 sup(0)” maka akan pada Lokasi ini allocate 20 byte yang akan diisi oleh data o. misalnya lagi “db 20 dup(“stack”)” maka kita akan membuat 20 variable, dan isinya akan “stack”. Kalau misalnya kita simpan hexa langsung huruf maka tidak bisa seperti “db AGH” ini salah sehingga harus diberikan 0 pada depannya seperti.

“value1 dw 2AB6h” -> maka akan disimpan 2 byte dan akan diurutkan dari yang terkecil sehingga menjadi ada 2 yaitu low byte adalah B6 dan high byte yaitu 2A.

MOV :

“mov al, count” operand awalnya sebagai destination, dan kedua adalah source, sehingga data dari variable count akan dipindahkan atau dicpy ke register AL. sehingga count harus merupakan 1 byte, karena AL atau AH hanya 1 byte (8 bit), kalau AX 2 byte. Bisa dengan seperti ini “mov AX, word ptr count” ini bisa berjalan karena count akan dijadikan seperti word.

Register berguna untuk menyimpan data secara sementara pada microprocessor.

“mov al, count” ini merupakan direct jadinya langsung, kalau indirect maka seperti “mov AX, CX” cs akan dilakukan secara indirect karena misalnya CX menyimpan 0120 merupakan address dan isi dari 0120 bernilai FF00 maka AX akan menjadi 00FF diurutkan dari lowbite

MINGGU 4

Interrupt adalha sebuah fasilitas pada microprocessor, ada hardware interrupt, software interrupt, cara kerjanya sama tapi dibedakan, hardware akan dibedakan jadi beberapa tingkatan juga.

Pada Hardware interrupt

Reset , ketika computer awal dinyalakan atau di tekan tombol reset sampai muncul operating systemnya, Tingkat paling tinggi. Kemudian ada non maskable interrupt, saat processor mendapat sinyal ini maka processor harus mau tidak mau melaksanakan non maskable interrupt, harus meninggalkan apa yang dikerjakan. Tapi kalau maskable interrupt diberikan pada microprocessor maka bisa tidak dijalankan perintah maskable interrupt.

Processor bisa melihat dari interrupt flag pada status register, kalau interrupt flag misalnya set maka dia bisa menolak apa yang duperintahkan user dari maskable interrupt. Tapi kalau non maskable interrupt akan harus dilakukan.

Pada software interrupt berupa instruksi.

Misalnya ada INT21, INT20 itu merupakan software interrupt. Interrupt tersebut berupa instruksi.

Cara kerja interrupt : ada pada pdf / slide

Misalnya MP menerima interrupt baik software atau hardware dan dia check flag interrupt, maka dia akan Bersiap siap register yang perlu saat itu akan dititipkan sehingga dia bisa Kembali nanti, karena kalau dia lagi kerjakan sesuatu maka akan diberhentikan, jadi register yang diperlukan (ditentukan pengguna), akan dititip di stack, baru akan masuk ke rutin interrupt yang ditunjuk oleh interrupt yang bersangkutan sampai mendapatkan Return From Interrupt, biasa sebelum eksekusi RTI, ada proses pengembalian isi register yang dititipkan di stack. Jadi diawal rutin ketika ada interrupt maka dia tidak melaksanakan rutin tapi menyimpan register yang diperlukan pada stack.

Sehingga pada saat RTI akan dikembalikan register yang dititipkan. Processor pada saat melanjutkan lagi dia akan melihat isinya lagi, kalau isinya ngawur maka tidak bisa.

Setiap interrupt ada Lokasi memorynya, Namanya interrupt Vector Table, sehingga ketika terjadi interrupt A maka dia akan ke mana. Akan ada daftarknya. Menggunakan vector karena ada arahnya, sehingga merupakan table yang akan mengarahkan.

Akan disediakan beberapa byte untuk simpan interrupt, misalnya 4 byte untuk setiap interrupt, kalau tidak dipakai maka akan dikosongkan biasanya.

INPUT – OUTPUT SERVICE

0B untuk dapet status console input, 0C untuk clear input buffer, jadi interrupt semua ada penomeran sendiri sendiri.

DOS service, diberikan nomer 21, untuk video service nomer 10. Jadi input output berada pada DOS Service dengan nomer 21, terus yang beberapa input dan output itu merupakan functionnya, ada beberapa fungsi misalnya ada fungsi 1, 10, dll.jadi misalnya input buffered nomer 10 pada DOS Service, maka yang ditunjuk adalah interrupt 21 dan dipanggil nomer 10. Cara dia tau nomer 10 itu sudah ada registernya.

Mov ah, 1 -> menggunakan function 1

Int 21h -> interrupt DOS Service, hasil input akan disimpan dalam code ascii

Mov char, AL - > dipindah ke char karena register AL bisa terganti jadi supaya data tidak hilang.

Mov ah,6

Mov dl, 0FFh -> karena fungsi 6 input output, maka jika ingin menggunakan input dl isi dengan ff, dan jika output dl isi dengan value yang ingin di ouput.

Kalau misalnya menggunakan fungsi nomer 7 pada interrupt 21 maka tidak akan menggunakan echo, sehingga ketika input tidak akan dicetak.

Bisa dibaca di buku chapter 5

0ah (control character) untuk ganti baris, tapi hanya turun sekali, 0dh (control character, carriage return, sehingga akan memberikan baris sebelah kiri, kalau hanya enter maka tidak ke bagian paling kiri) carriage return ini merupakan ascii. Kalau misalnya ingin menambahkan blank space / white space maka gunakan 20h

Jadi nomer function itu akan seperti tersimpan pada arsip interrupt tersebut, maka misalnya untuk IO menggunakan interrupt 21 dan fungsi yang digunakan aka nada beberapa yang bisa kita gunakan.

**MINGGU 5**

Membuat segitiga terbalik, dengan memberikan input 9 yang disimpan adalah asciinya, ascii angka mulai dari 30 sampai dengan 39. 00 sampai 1F adlaha control character bisa untuk komunikasi dan lain sebagainya, misalnya untuk kirim data dengan serial atau lainnya itu ada di sana.

Mov ah, 1 - pakai fungsi input 1

Int 21h – ketika kita input missal 9 maka akan masuk ke dalam al

And al, 0fh – and adalah sifat and gate yaitu semuanya harus 1, kalau misalnya salah satu 0 maka hasilnya pasti 0. Sehingga maksudnya adalah isi dari al di and kan dengan 0fh dan hasilnya akan disimpan di al. ketika misalnya al sudah tersimpan sebuah angka 9 misalnya, maka jika ascii 39 di and dengan 0F maka akan menjadi 0011 1001 and 0000 1111 maka akan menjadi 0000 1001, sehingga akan mendapat 9.

Mov cl, al –

Call cetak\_cr – memanggil sebuah procedure.

Cetak\_cr :

…

Ret – untuk return ke pemanggilnya

Untuk memberhentikan program bisa dengan code berikut

Mov ax, 04c00h

Int 21h

Atau bisa juga dengan ah, 04ch. atau interrupt 20h hanya saja ada sedikit perbedaan.

Pencetakan hasil

Iterasi :

Mov bil, cl – sehingga angka 9 akan dipindahkan ke variable bil

Or bil, 30h – untuk menjadikan 39 yaitu ascii dari angka 9

Dec bil – berguna untuk decrement bil.

Loop cetak – maka akan Kembali ke label cetak:, dia akan terpanggil ke label cetak selama cl masih belum 0, jika selesai habis maka akan lanjut ke step lanjtunya.

Inc cl – increment cl, increment tersebut berguna untuk loop iterasi pada akhir, karena setiap loop akan mengurangi cl sebanyak 1.

Pada program ini tidak bis ajika digunakan untuk 2 angka, karena hanya dibataskan 1 character input saja. Sehingga kitab isa menginputkan character, misalnya H maka akan mendapatkan ascii 68 sehingga kalau di and akan menjadi 08, sehingga sama dengan kita input angka 8.

Loop akan mengambil dari cx.

MEMBUAT PROGRAM UNTUK MEMBALIK KALIMAT

Mov ah, 0ah – kita menggunakan input buffer. Membaca input string, akan menyimpan hexa, penyimpanan pada byte petama merupakan maximum number of character allowed (akan disimpan dalam bentuk hexa, maksimalnya 255 character yatu FF, bukan 256 karena terpakai carries return) termasuk carried return . byte kedua adalah jumlah character yang kita. Setelah itu maka adalah data yang kita masukan.

Mov dx, offset max\_key –

Buffer db 75 dup (0) – maka 75 isi dengan 0 duplicate semua.

**MINGGU 6**

Ketika membuat program dan menggunakan call harus terdapat pada main program.

Untuk melakukan video display, maka character yang harus ditampilkan ada dan terdapat attribute seperti background dan foreground colornya dan sebagainya, semua itu termasuk video attribute.

Pada layer standard terdapat 25 baris dan columnnya 80, mulai dari 0-24 dan 0-79 sehingga terdapat total 2000. 1 character disimpan dalam 2 byte yaitu character dan attributenya. Sehingga 2000 x 2 byte yaitu 4000 byte. Sehingga 1 halaman layer disimpan 4000 byte. Selain itu enaknya kita juga bisa menuju pada Lokasi tertentu karena apa yang ditampilkan di layer di simpan di memory. Sehingga kita mengirimkan pada posisi tertentu mengenai character dan attributenya. Setiap attribute memiliki nilai hexanya masing masing. Bit 4 5 6 untuk background dan 0 1 2 3 untuk foreground.

Pada adapter monochrome kita cuman bisa ada 1 halaman saja.

Jika misalnya cga mode 00h 01h bisa sampai 8 halaman

Tujuan jumlah halaman adalah jika kadang kadang sebuah layer perlu disimpan dan ganti layer untuk input sesuatu yang lain, tetapi beberapa saat kita perlu layer yang sebelumnya maka seperti pop up yang bisa kita simpan.

Untuk monitor menggunakan interrupt 10, sehingga jika kita ingin set video Mode maka kita menggunakan function 0, selain itu set cursor lain bisa dengan menggunakan function 1. Kita juga bisa mendapatkan posisi cursor saat ini.

Move ah, 0 -> set video mode

Move al, 6 -> menggunaan mode 6 yaitu 640 x 200 color graphics

Int 10h

Dari code tersebut maka mode video akan di set pada mode 6, layer Ketika kita menjalankan code itu akan dijalankan pada mode monitor 6

Ah -> menempatkan function yang mau digunakan

CGA dan EGA terdiri dari 8 scanline. Untuk monochrome lebih rapat karena terdapat 11 scanline untuk character sedangkan CGA dan EGA hanya 6 scanline.

Savecursor pada chapter 05 halaman 19 untuk Ketika kita misalnya ingin Kembali ke cursor / mungkin page sebelumnya.

Mov ax,@data

Mov ds,ax

Digunakan untuk inisialisasi registernya.

Mov ah, 9

Mov dx, offset page1 -> menunjukan dx ditunjukan dengan Alamat yang menampung isi dari page1.

Int 21h

Offset merupakan assembler directive, merupakan code juga tapi tidak menghasilkan code hexa sehingga membantu meletakan varible pada register.

Kalau misalnya variable numeric bisa tanpa offset kalau misalnya string maka harus menggunakan bisa.

Ketika perpindahan page maka page yang sebelumnya akan tersimpan pada memory.

Untuk manipulasi video menggunakna interrupt 10

Pada interrupt 20 tidak ada register yang perlu dimasukan.

Setiap halaman aka nada ukurannya seperti yang dijelaskan sebelumnya, dan setiap halamna akan disimpan pada memory sehingga Ketika kita berpindah halaman, maka page lainnya akna tersimpan.

**MINGGU 7**

Microprocessor, akan terdapat control unit, ALU dan register. Pada 8086, registernya akan terdiri dari 16 bit tetapi ada juga register khusus yang 8 bit. Pada sebuah microprocessor terdapat juga data bus, address bus, control bus. Data bus dan addres bus akan keluar secara bersamaan. Data bus bisa terdapat 2 arah. Jika misalnya address bus hanya 1 arah. Data bus terdapat 16 bit dan address bus 20 bit. Makanya registernya tidak ada yang lebih dari 16 bit.

Register CS dan IP keduanya 16 bit susunannya

x x x x

x x x x

makanya bisa sebagai 20 bit.

Semua register akan kerja bersamaan.

Terdapat register khusus yaitu PSR/PSW/Flag Register. Setiap processor akan mempunyai, bisa berguna untuk mendeteksi bilangan positif negative, khusus register tersebut bitnya independent, akan kerja sendiri sendiri.

Control unit, yang akan control misalnya memastikan data bus dan address bus siap dan melakukan hal selanjutnya.

SETIAP CODINGAN PADA UTS akan terdapat remark nrp dan nama. Pada awal Tengah dan akhir program.

Pada uts akan terdapat 2 jenis soal, untuk dikirim dari email dan juga tulis kertas.

REGISTER untuk menampung data, sebenarnya isinya adalah flip flop

ALU untuk melakukan aritmatik dan logika

ASSEMBLER

Assembly Lang akan diterjemahkan menjadi machine code oleh assembly.

Pada instruksi assembly language terdapat 4 field, field pertama label, field kedua ada mnemonic, field ketiga adlaha operand, dan field ke 4 merupakan remark.

Org (origin) tidak ada pada code 8086, db juga tidak ada, sehingga itu disebut sebagai assembler directive yang maksudnya adalah instruksi yang tidak menghasilkan code hexa, bukan merupakan code microprocessor, tapi akan membantu process assembly, akan membantu user untuk memproses assembly.

Assembly adalah Bahasayang digunakan untuk memahami Bahasa machine, misalnya memahami mov lebih mudah dibandingkan dengan menghafal codenya.

Machine code -> code hexa atau operation code, dalam bentuk biner tapi disimplekan menjadi hexa sehingga lebih mudah dipahami.

Terdapat istilah assembly directive sebagai pseudocode (kode semu).

Lokasi memory

0100 : 0A 3C 41 20 7A 56

Sehingga pada 0100 adalah 0A, selanjtunya akan terurut 0101 adalah 3C, 0102 adalah 41 sampai seterusnya.

Nilai (0104/DW) maka variable tersebut akan bernilai maka nilainya adlah 567A karena mendapatkan 2 byte yaitu 7A dan 56 dan diurutkan dari lowbyte

2 byte untuk 1 word

Stack memory -> LIFO.

Loop akan memperhatikan register cx untuk counter.