

PERCOBAAN 01

MIKROPROSESOR DAN INSTRUCTION SET

A. Pendahuluan

Arsitektur komputer adalah konsep perencanaan dan struktur pengoperasian dasar dari suatu sistem komputer. Arsitektur komputer ini merupakan rencana cetak-biru dan deskripsi fungsional dari kebutuhan bagian perangkat keras yang didesain (kecepatan proses dan sistem interkoneksinya). Dalam hal ini, implementasi perencanaan dari masing-masing bagian akan lebih difokuskan terutama, mengenai bagaimana CPU akan bekerja, dan mengenai cara pengaksesan data dan alamat dari dan ke memori cache, RAM, ROM, cakram keras, dll). Beberapa contoh dari arsitektur komputer ini adalah arsitektur von Neumann, CISC, RISC, blue Gene, dll.

Mikroprosesor adalah sebuah chip (IC=Integrated Circuits) yang di dalamnya terkandung rangkaian ALU (Arithmetic-Logic Unit), rangkaian CU (Control Unit) dan kumpulan register-register. Mikroprosesor disebut juga dengan CPU (Central Processing Unit) yang digunakan sebagai otak/pengolah utama dalam sebuah sistem komputer. Mikroprosesor pertama yang diproduksi adalah mikroprosesor 4bit dari intel yang diberi nama Intel 4004, lalu dikembangkan menjadi Intel 4008, lalu dikembangkan lagi ,menjadi 8 bit dengan diproduksinya seri 8008 dan 8085. Agar CPU dapat bekerja sesuai dengan yang diinginkan, maka diperlukan suatu instruksi. Setiap mikroprosesor atau CPU mempunyai satu set kode instruksi (instruction set) yang spesifik, dan berbeda antara instruction set antara CPU satu dengan yang lainnya.

Selama berlangsungnya eksekusi instruksi, instruksi dibaca ke dalam register instruksi yang terdapat dalam CPU. Untuk melakukan operasi yang diperlukan, CPU harus dapat mengeluarkan data dari berbagai bidang instruksi. Opcode direpresentasikan dengan singkatan-singkatan, yang disebut mnemonik, yang mengindikasikan operasi, contohnya adalah:

ADD	: Add (Menambahkan)
SUB	: Subtract (Pengurangan)
MPY	: Multiply (Perkalian)
DIV	: Divide (Pembagian)
LOAD	: Muatkan data data dari memori
STOR	: Simpan data ke memori

B. Percobaan

Tools :

1. Proteus Professional :

<https://downloadly.net/2020/13/3175/03/proteus/03/?#/3175-proteus-032127081430.html>

2. EMU8086 :

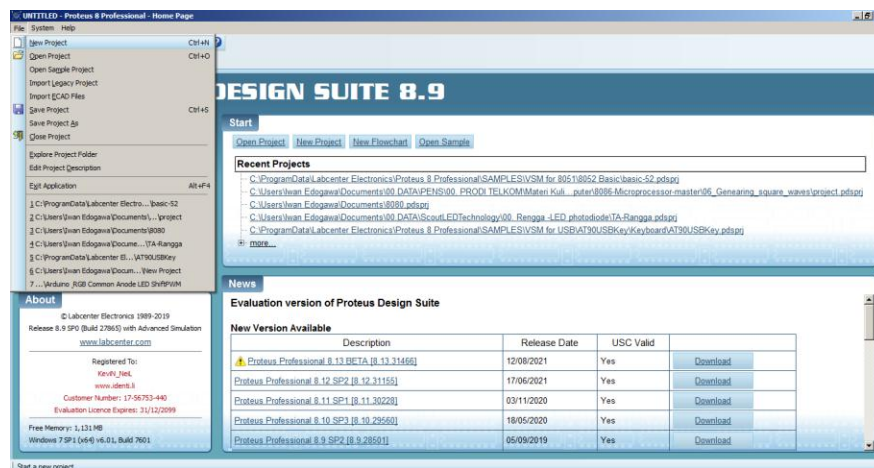
https://drive.google.com/drive/folders/1OPVhsYiHJm3_rfvUWiqL9yJW5Wn7S3LU

Bahan percobaan :

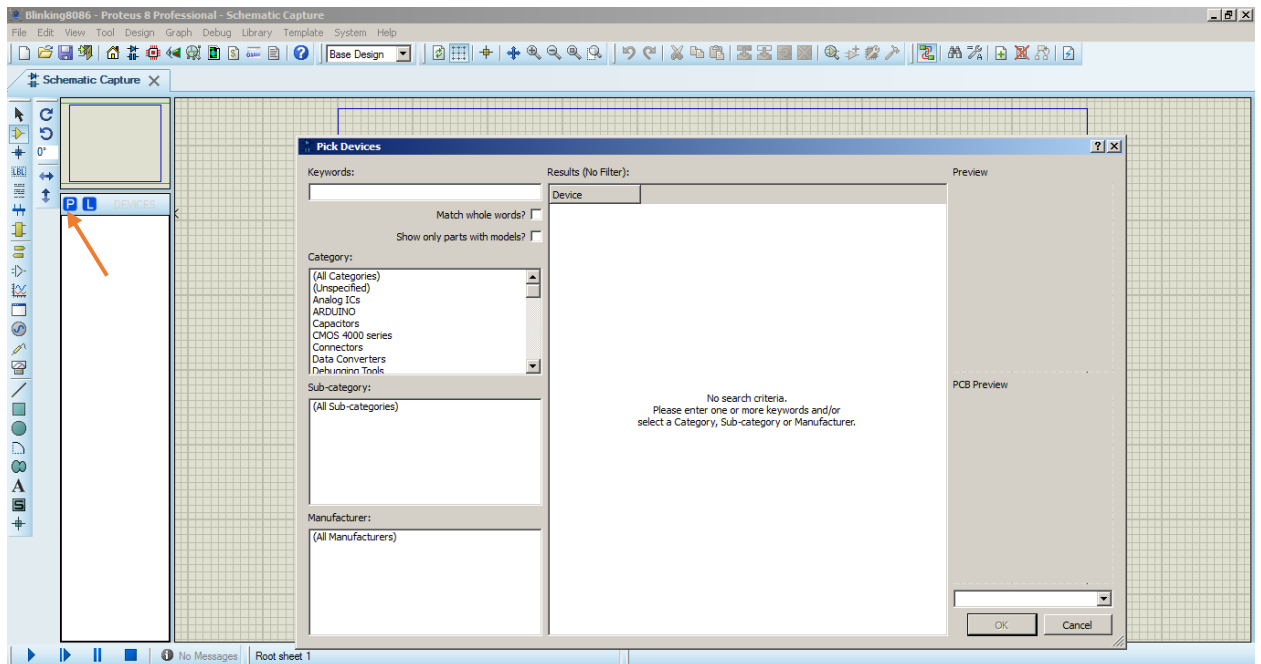
1. Datasheet Intel 8086
2. Instruction set for Intel 8086
3. Datasheet IC 8255 PPI
4. Datasheet IC 74HC373

Langkah percobaan :

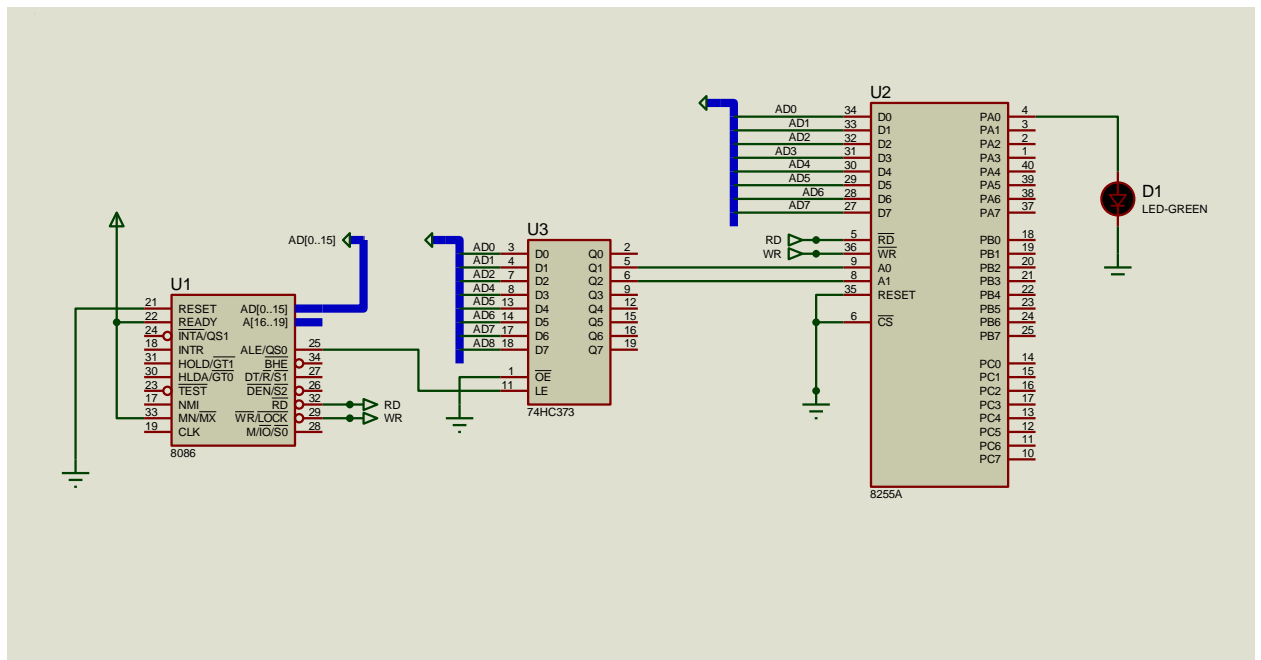
1. Instal Proteus dan EMU8086
2. Buat project baru di proteus



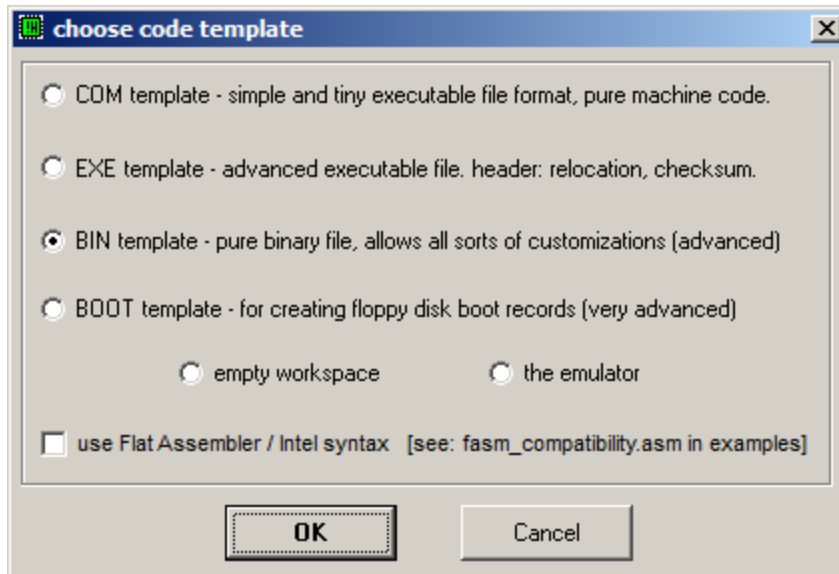
3. Setelah project baru sudah jadi, pilih komponen yang akan digunakan, klik tanda sebagaimana yang ditunjukkan oleh gambar berikut



4. Buat rangkaian sebagaimana yang ditunjukkan oleh gambar berikut :



5. Open EMU8086, buat file baru, pilih BIN Template



6. Ketikkan kode assembly berikut :

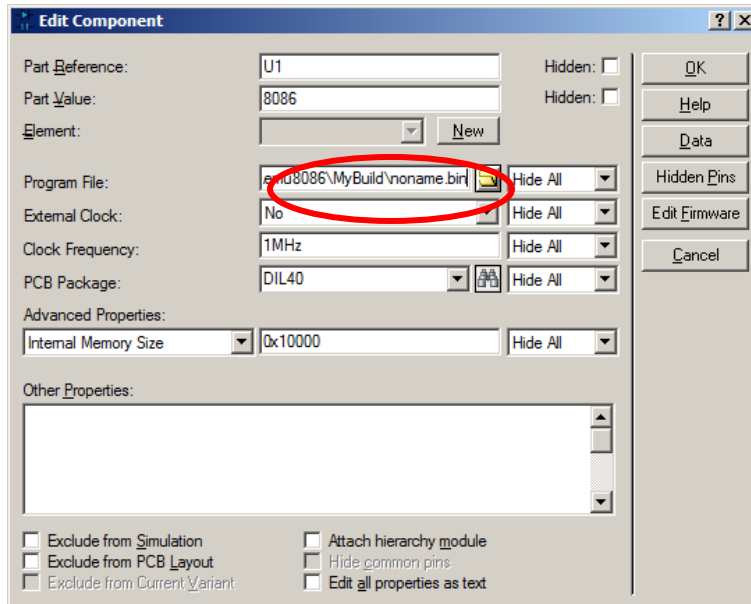
```

34 ; add your code here
35 DATA SEGMENT
36     PORTA EQU 00H
37     PORT_CON EQU 06H
38 DATA ENDS
39
40 CODE SEGMENT
41     MOV AX, DATA
42     MOV DS, AX
43
44     org 0000h
45
46 ; add your code here
47 START:
48
49     MOV DX, PORT_CON
50     MOV AL, 10000000B; Port A, Port B, Port C as output
51     OUT DX, AL
52
53
54     JMP XX
55 XX:
56     MOV AL, 0000H
57     MOV DX, PORTA
58     OUT DX, AL
59     MOV CX, 0DF36H; Delay
60     loopy1: loop loopy1
61     MOV AL, 0001H
62     MOV DX, PORTA
63     OUT DX, AL
64     MOV CX, 0DF36H; Delay
65     loopy2: loop loopy2
66     JMP XX
67
68
69 CODE ENDS
70 END
71
72
73 HLT ; halt!
74

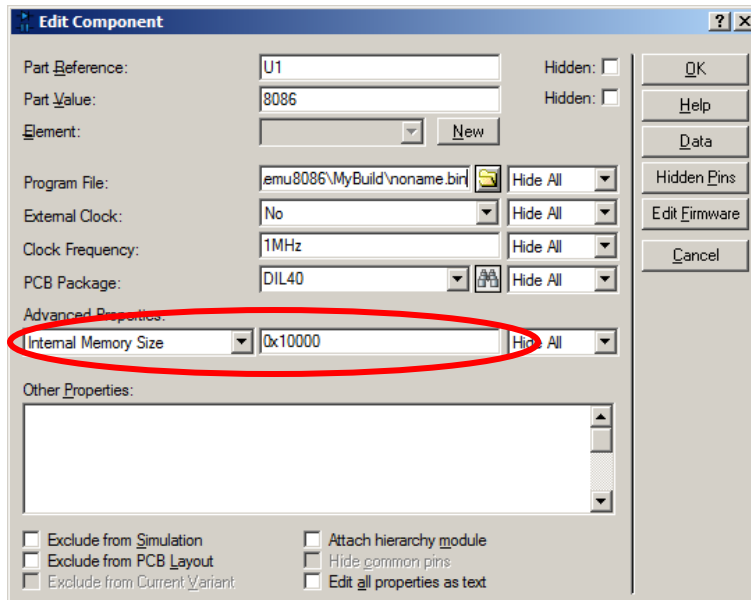
```

7. Compile program yang telah dibuat dan simpan file *.bin pada directory yang diinginkan.

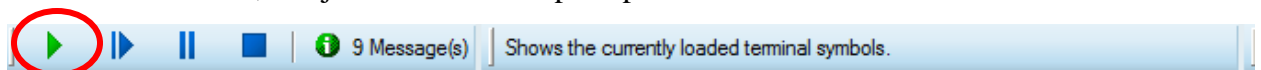
8. Masukkan file *.bin yang telah digenerate pada simulator proteus, arahkan ke directory dimana file *.bin tersimpan



9. Ubah memory size seperti berikut:



10. Jika sudah klik OK, dan jalankan simulasi pada proteus



11. Amati yang terjadi pada LED

12. Sekarang coba pindahkan LED pada sambunan PA7, modifikasi program assembly sehingga menghasilkan output yang sama.

C. Tugas

1. Apa yang dimaksud dengan register pada sistem mikrokontroler?
2. Jelaskan tentang register-register berikut :
AL, AX, CX
3. Jelaskan perintah MOV, JMP dan OUT!