LAPORAN RESMI 2

PRAKTIKUM ARSITEKTUR KOMPUTER

"MIKROPROSESOR DAN INSTRUCTION SET SERTA PENAMBAHAN DELAY PADA INSTRUKSI"

Disusun Oleh :

Izzuddin Ahmad Afif (2421600011)

Dosen :

Mohamad Ridwan S.T., M.T.

PROGRAM STUDI SARJANA TERAPAN TEKNOLOGI REKAYASA INTERNET

DEPARTEMEN TEKNIK ELEKTRO

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2021/2022

Praktikum 1
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 mnemorik, yang mengindikasikan operasi, contohnya adalah: ADD : Add (Menambahkan) SUB : Substract (Pengurangan) MPY : Multiply (Perkalian) DIV : Divide (Pembagian0 LOAD : Muatkan data data dari memori STOR : Simpan data ke memori .

B. Percobaan

Tools:

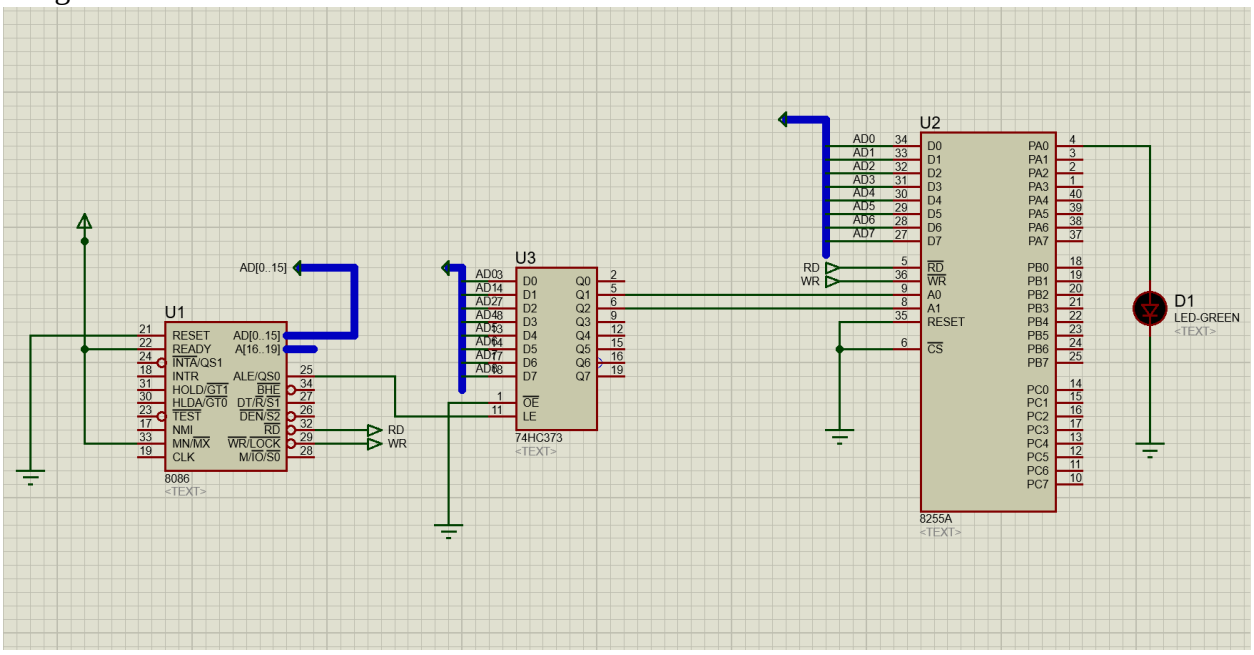
1. Proteus Professional
2. EMU8086

Bahan Percobaan:

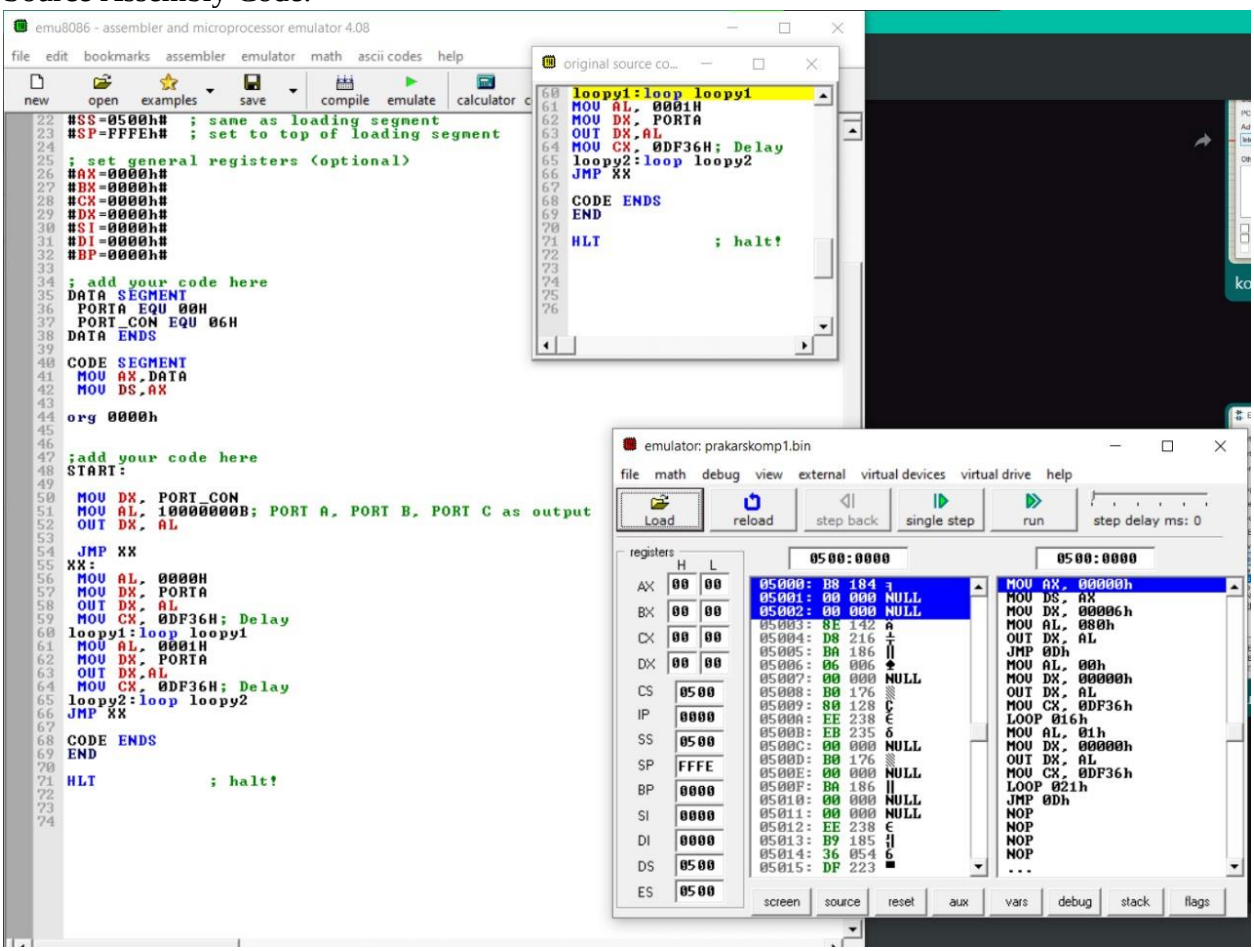
1. Datasheet Intel 8086
2. Instruction set for Intel 8086
3. Datasheet IC 8255 PPI
4. Datasheet IC 74HC373

Hasil Percobaan:

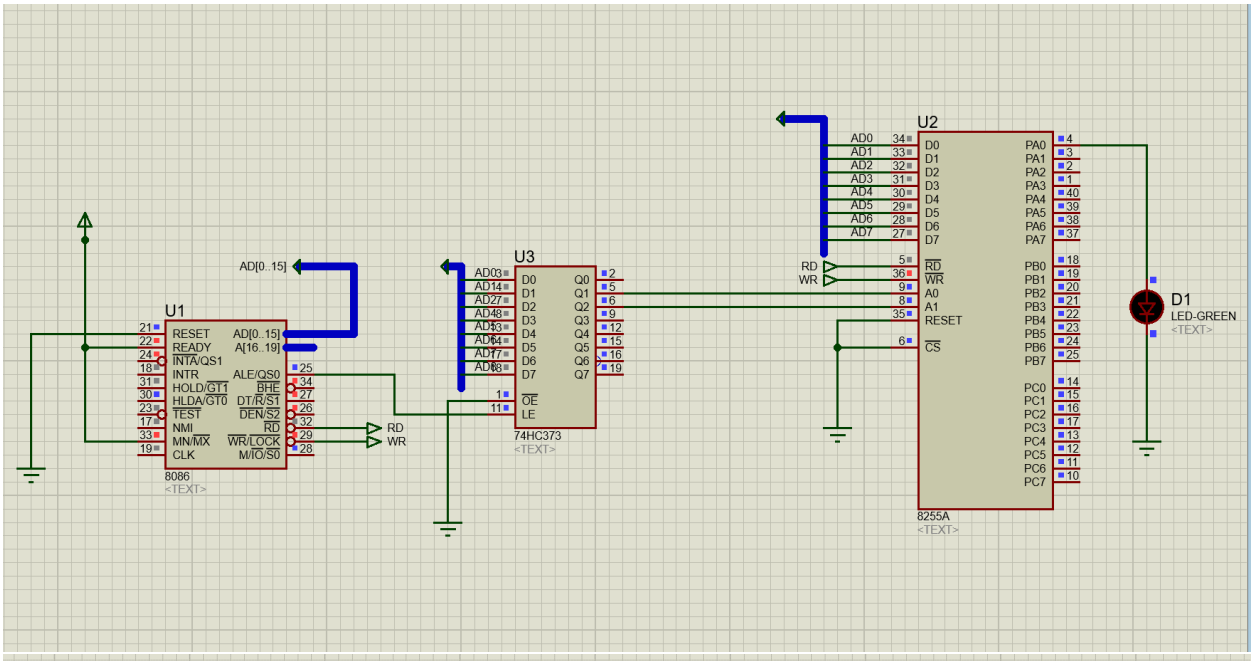
- Rangkaian:

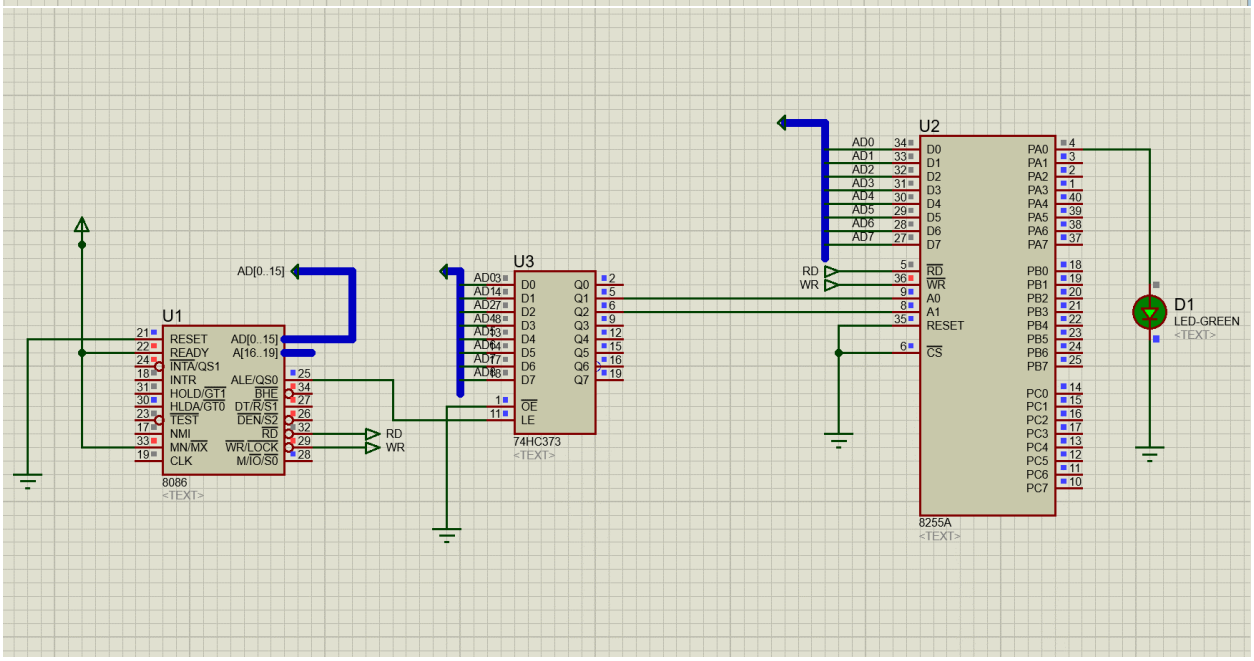


Source Assembly Code:

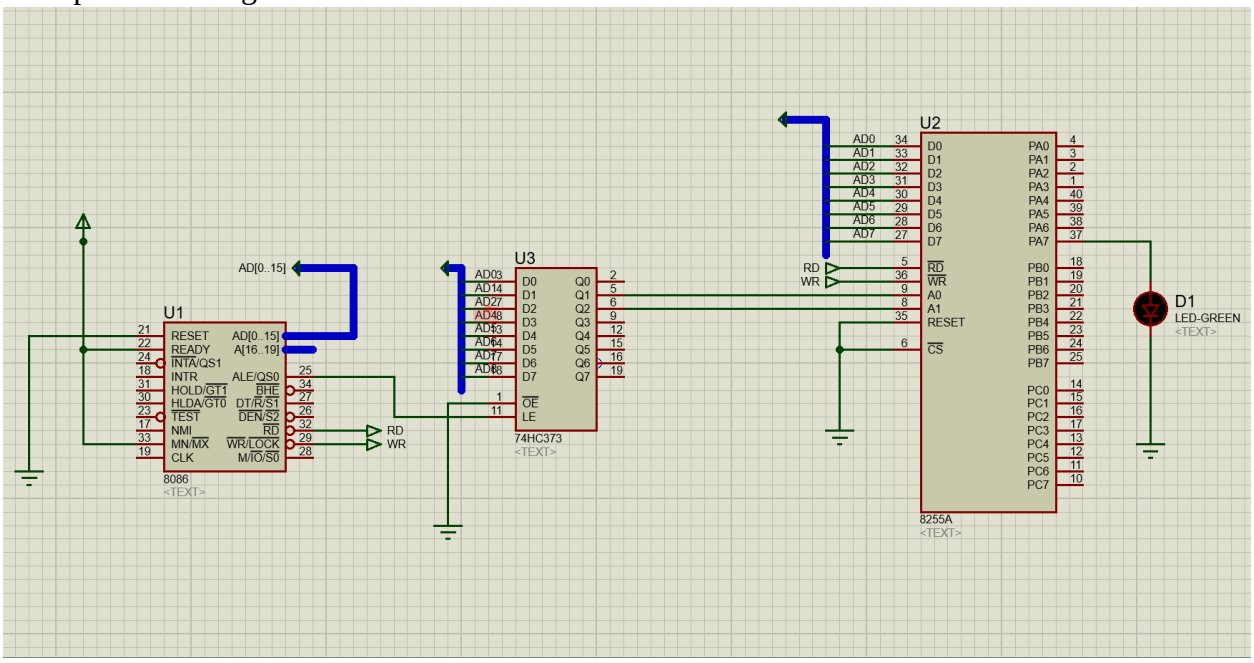


- Hasil Run:

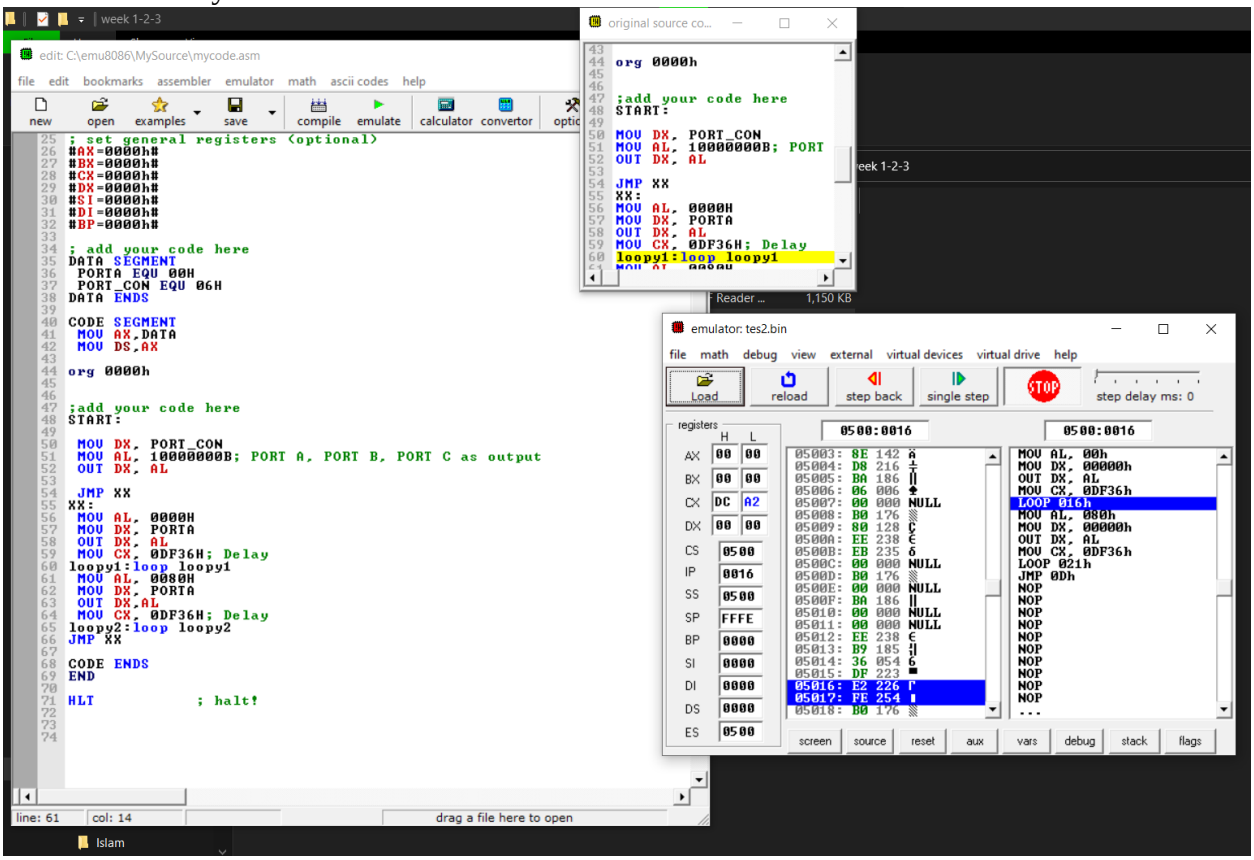

U1
8086
<TEXT>
U3
74HC373
<TEXT>
U2
8255A
<TEXT>
D1
LED-GREEN
<TEXT>
AD[0..15]
RD
WR


U1
8086
<TEXT>
U3
74HC373
<TEXT>
U2
8255A
<TEXT>
D1
LED-GREEN
<TEXT>
AD[0..15]
RD
WR

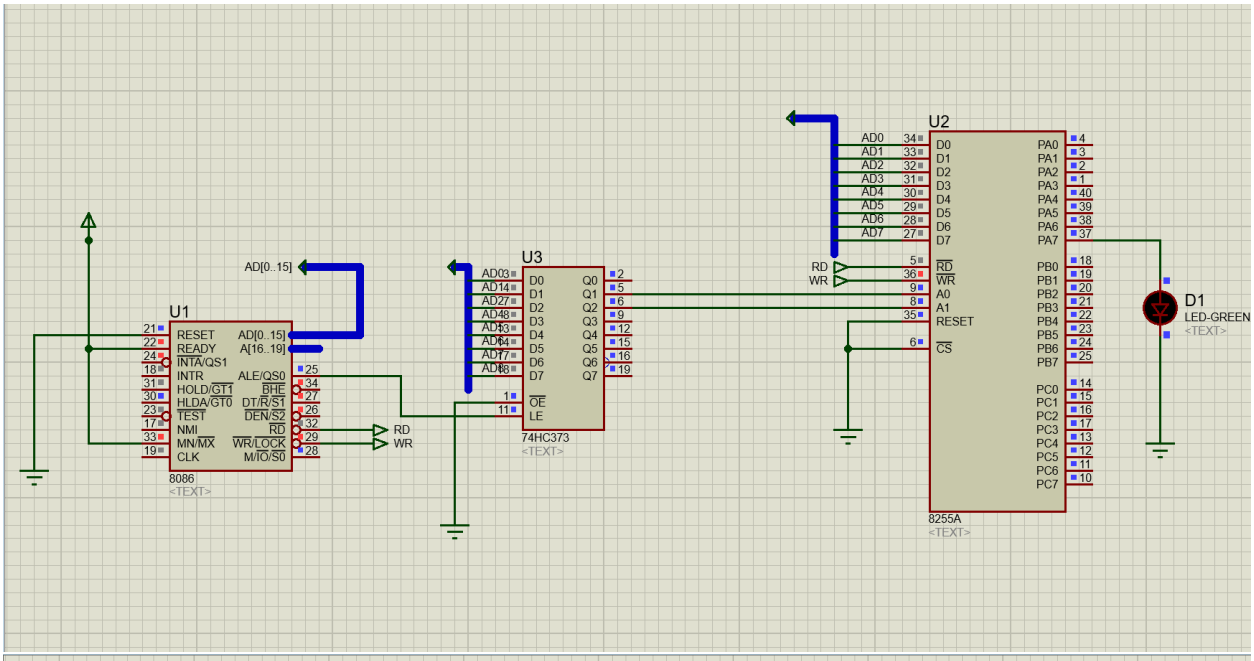
- LED pada sambungan PA7:

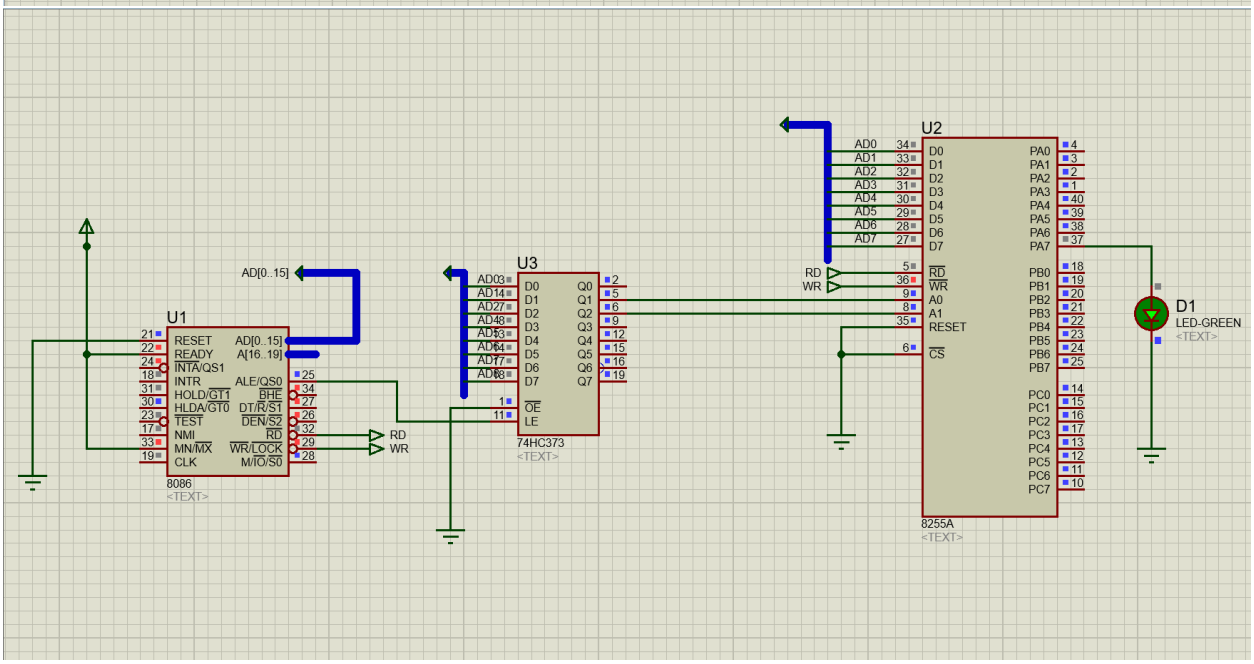


- Source Assembly Code:



- Hasil Run:


U1
8086
<TEXT>
AD[0..15]
A[16..19]
RESET
READY
INTA/QS1
INTR
HOLD/GT1
HLDA/GT0
TEST
NMI
MN/MX
CLK
ALE/QS0
BHE
DT/R/S1
DEN/S2
RD
WR/LOCK
M/IO/S0
U3
74HC373
<TEXT>
D0
D1
D2
D3
D4
D5
D6
D7
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
OE
LE
U2
8255A
<TEXT>
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
RD
WR
A0
A1
RESET
CS
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
D1
LED-GREEN
<TEXT>


U1
8086
<TEXT>
AD[0..15]
A[16..19]
RESET
READY
INTA/QS1
INTR
HOLD/GT1
HLDA/GT0
TEST
NMI
MN/MX
CLK
ALE/QS0
BHE
DT/R/S1
DEN/S2
RD
WR/LOCK
M/IO/S0
U3
74HC373
<TEXT>
D0
D1
D2
D3
D4
D5
D6
D7
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
OE
LE
U2
8255A
<TEXT>
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
RD
WR
A0
A1
RESET
CS
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
D1
LED-GREEN
<TEXT>

ANALISA PENAMBAHAN DELAY:

Program default sudah memiliki delay sebesar kurang lebih 1 detik, sehingga bila kita ingin mengatur delay dengan mudah, kita dapat membagi value yang akan kita assignkan ke register CX (register untuk loop counter) dengan 2 untuk mendapatkan hasil 500 ms serta mengalikan dengan 3/10 untuk menghasilkan 300 ms. Kalua value awal adalah DF36 menghasilkan 1 detik, maka value yang menghasilkan 500 ms adalah DF36/2 = 6F9B dan value yang menghasilkan 300 ms adalah DF36 * 3/10 = 29DA. Namun bila kita ingin presisi, kita bisa menggunakan perhitungan clock cycle setiap perintah dimana loop adalah 17 saat loop dan 5 saat keluar loop, dan mov reg, imm adalah 4, dll (dapat dilihat di manual intel 8086 untuk selengkapnya) dimana kita kalikan jumlah loop dengan value yang ada di register CX sehingga kita mendapatkan total cycle delay, lalu kita tambahkan cycle dari instruksi lain, sehingga kita dapatkan total cycle dalam sekali iterasi keadaan on/off, kemudian kita kalikan dengan periode clock cycle (dalam kasus kali ini, periode adalah 1 karena frekuensi CPU adalah 1 mhz) sehingga kita dapatkan besaran delay dalam satuan mikro second. Sebenarnya, kita bisa menambahkan atau mengurangi delay dengan memvariasikan frekuensi CPU. Semakin besar frekuensi, maka semakin sedikit delay, vice versa. Namun hal ini sangatlah tidak praktikal, karena dalam dunia nyata, hampir tidak mungkin seorang engineer mengganti frekuensi CPU hanya untuk menambah maupun mengurangi delay.

```
; add your code here
DATA SEGMENT
PORTA EQU 00H
PORT_CON EQU 06H
DATA ENDS

CODE SEGMENT
MOV AX,DATA
MOV DS,AX

org 0000h


;add your code here
START:

MOV DX, PORT_CON; moving control port adress to dx
MOV AL, 10000000B; PORT A, PORT B, PORT C as output
OUT DX, AL; transfer byte from al to control port in dx
;4+4+10 clocks

JMP XX;15 clocks

XX:
MOV AL, 0000H; OFF STATE coz 0h points to nothing
MOV DX, PORTA; move port a to register dx
OUT DX, AL    ; move 0h to register dx
MOV CX, 0DF36H; Delay by moving h value to loop counter register
;4+4+10+4 clocks sisa 999,978
loopy1:loop loopy1;17*cx-12

MOV AL, 0081H; ON STATE coz 81h points to both 1st and 1st and 8th x port
MOV DX, PORTA; move port a to register dx
OUT DX, AL    ; move 81h to register dx, so its ap0 and ap7 that are gonna be lit.
MOV CX, 0DF36H; Delay by moving h value to loop counter register
;4+4+10+4 clocks
loopy2:loop loopy2;17*cx-12

JMP XX; by jmping back, we create an infinite loop. 15 clocks.

CODE ENDS
END

HLT            ; halt!
```

Gambar: Program Assembly dengan delay 1 detik beserta komentar penjelasan.

Analisa:

Percobaan kali membahas tentang Mikroprosesor dan set-set instruksi. Pertama kita membuat rangkaian seperti di modul, lalu kita mengetikkan code assembly seperti yang ada di modul dan dicompile dalam bentuk .bin. Lalu kita import code kita dari EMU8086 ke perangkat 8086 yang ada di rangkaian serta mengganti internal memory size nya menjadi 0x10000 dan kita run percobaannya. Setelah itu kita memindah LED pada sambungan PA7 dan mengubah program assembly supaya dihasilkan input yang diinginkan.

Di program assembly, ada kode untuk mengalirkan listrik menuju LED yang menggunakan akhiran h yang menandakan heksadesimal atau dapat juga diisi dengan biner dengan symbol b, di situ bila kita isi dengan 0001h/00000001b maka akan menyalakan sambungan dengan label PA0 sedangkan 0080h/10000000b akan menyalakan sambungan label PA7.

Kesimpulan:

Dari percobaan kali ini, dapat disimpulkan bahwa mikroprosesor dapat menerima set-set instruksi dari code assembly dan mengeksekusinya sesuai code, lalu menghasilkan sebuah input yang sesuai dengan code juga.

TUGAS:

1. Register adalah merupakan memori pada bagian komputer (mikroprosessor) yang berkapasitas kecil namun memiliki kecepatan baca yang sangat tinggi. Register pada mikroprosessor dapat diibaratkan sebagai kaki dan tangan dari mikroprosessor, karena dalam setiap pekerjaan, mikroprosessor selalu mengandalkan dan melibatkan register sebagai perantara dan sebagai komponen yang paling banyak melakukan pekerjaan mikroprosessor. Secara umum, fungsi sebuah register pada mikroprosessor adalah sebagai tempat penyimpanan sementara untuk menyimpan hasil dari operasi aritmatika ataupun operasi logika yang dilakukan oleh mikroprosessor.

2. - Register AX (AH + AL) / Accumulator Register

   Register AX adalah register yang berfungsi untuk menyimpan dan membaca data yang berhubungan dengan operasi aritmatika, yang meliputi perkalian, pembagian, penjumlahan dan pengurangan (KABATAKU). Karena setiap general purpose register memiliki register High dan Low, maka untuk Register AX, register Low nya adalah AL dan Highnya adalah AH.

   - Register CX (CH + CL) / Counter Register

   Register CX merupakan memori yang berfungsi untuk menyimpan jumlah lompatan pada loop yang dilakukan oleh mikroprosessor. Secara umum, terdapat beberapa fungsi dari register CX, yaitu:

   - Melakukan operasi pencacahan untuk operasi loop
   - Melakukan operasi pencacahan untuk operasi shift dan rotate
   - Melakukan operasi pencacahan untuk operasi string

3. Assembly Language

   a. MOV is an X86 assembly language instruction, it is meant to move data between registers and memory.

   b. JMP instruction performs an unconditional jump. Such an instruction transfers the flow of execution by changing the instruction pointer register.

   c. OUT instruction transfers a byte, word, or long from the AL, AX, or EAX registers respectively to a port (0 to 65535), specified by the DX register.