**UTS EL2008 – Pemecahan Masalah dengan C**

Hari / Tanggal : Sabtu / 19 Maret 2022

Dosen Pengampu : Arif Sasongko

Reza Darmakusuma

M. Ogin Hasanuddin

Waktu : 19 Maret 2022 (12.00) – 20 Maret 2022 (12.00)

Sifat : Openbook

“Saya menyatakan bahwa saya melaksanakan ujian ini dengan jujur. Apabila nanti ditemukan kecurangan dalam pelaksanaan ujian ini, saya bersedia menerima sanksi yang diberikan.”

Nama : Muhammad Hanif Hibatullah

NIM : 13220051

1.

a. Representasi integer dengan **unsigned** akan dimulai dari angka 0 sampai 2n-1 dengan n merupakan banyak bit yang direpresentasikan, hal ini mirip dengan range dari unsigned int pada bahasa C. Sehingga jika kita memiliki 16 bit digit yang akan direpresentasikan dalam **unsigned** maka nilai tertinggi yang didapatkan adalah 216-1 = 65535 dan nilai terendah adalah 0. Untuk representasi integer dengan **2’s-complement** akan dimulai dari angka -2n-1 sampai dengan 2n-1-1 hal ini mirip dengan range integer pada bahasa C. Sehingga jika kita memiliki 16 bit digit yang akan direpresentasikan dalam **2’s-complement** maka nilai tertingginya adalah 215-1 = 32767 dan nilai terendahnya adalah -32768.

b. Bilangan yang saya dapatkan: 32051

Dalam desimal dapat dinyatakan sebagai:

32051 =

Sehingga dalam desimal 32051 dapat direpresentasikan sebagai 3205110.

Dalam biner dapat dinyatakan sebagai:

Sehingga dalam biner 32051 dapat direpresentasikan sebagai 1111101001100112.

c. least significant byte dari jawaban b adalah A = 0011 0011 (karena byte = 8 bits).

d. A = 0011 0011

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **x** | **y** | **Operasi** | **Binary (least significant byte)** | **Hexa** | **Unsigned** | **Signed 2C** |
| 1010 0101 | 1110 1111 | (A & y) ^ x | 1000 0110 | 86 | 134 | -122 |
| 1101 1011 | 1111 0011 | (~(y >> 4) && x) + A | 1001 1101 | 9D | 157 | -99 |
| 0010 1111 | 1010 0101 | ((x >> 2) | y) \* A | 0101 1101 | 5D | 93 | 93 |

Diasumsikan x, y, A merupakan **signed binary** (bukan **signed 2’s complement**) dan diasumsikan pula hasilnya hanya diambil least significant bytenya.

Operasi operasi = & (and), ^ (xor), >> (shift right), ~(negation), +(add), | (or), \*(multiplication)

Soal pertama:

(A & y) = (0011 0011 & 1110 1111) = (0010 0011)

(A & y) ^ x = (0010 0011) ^ (1010 0101) = 1000 0110

Soal kedua:

(~(y >> 4) && x) \*diasumsikan hanya satu tanda & = (~(1111 0011 >> 4) & 1101 1011)

(~(y >> 4) & x) = (~(0000 1111) & 1101 1011) = (1111 0000 & 1101 1011) = 1101 0000

(~(y >> 4) & x) + A = 1101 0000 + 0011 0011 = -80 + 51 = -29 = 1001 1101

Soal ketiga:

((x >> 2) | y) = ((0010 1111 >> 2) | 1010 0101) = (0000 1011 | 1010 0101) = 1010 1111

((x >> 2) | y) \* A = 1010 1111 \* 0011 0011 = (-47) \* 51 = -2397 = 1 1001 0101 1101

Dimana ketika diambil least significant bytenya didapatkan hasil dari soal ketiga yaitu 0101 1101

e. Untuk program yang dibuat dengan bahasa C dapat diakses pada link berikut:

<https://github.com/HanifHBK/SolusiUTSPMC> dengan nama file 1e.c

f. Hasil yang didapatkan dari program:

Text

Description automatically generated

Dimana dapat dilihat bahwa tidak terdapat perbedaan hasil untuk NIM saya (tidak dilakukan pengecekan dengan NIM lain).

2. Terlebih dahulu rumus untuk mencari m dan v disederhanakan

a. Flowchart yang digunakan (dibuat dengan draw.io):

A picture containing diagram

Description automatically generated

Program dalam bahasa C (kode yang lebih jelas dapat diakses pada link:

<https://github.com/HanifHBK/SolusiUTSPMC> berupa file 2a.c

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define C 2.9979

#define E 1.602

#define m0 9.109

#define MAXLEN 255

int main**(){**

char namafile**[**MAXLEN**];**

printf**(**"Masukkan nama file berisi Tegangan: "**);**

scanf**(**"%s"**,** **&**namafile**);**

FILE**\*** vfile **=** fopen**(**namafile**,** "r"**);**

char line**[**MAXLEN**];**

float Tegangan**[**MAXLEN**];**

int count **=** 0**;**

**while(**fgets**(**line**,** MAXLEN**,** vfile**)){**

Tegangan**[**count**]** **=** atof**(**line**);**

count**+=**1**;**

**}**

int i**;**

float m**[**count**],** mpow**[**count**],** v**[**count**],** vpow**[**count**];**

**for(**i **=** 0**;** i **<** count**;** i**++){**

m**[**i**]** **=** **(**Tegangan**[**i**]\***E**\***pow**(**10**,-**4**)/(**C**\***C**))** **+** m0**;**

mpow**[**i**]** **=** **-**31**;**

**while(**m**[**i**]** **>=** 10**){**

mpow**[**i**]** **+=** 1**;**

m**[**i**]** **/=** 10**;**

**}**

v**[**i**]** **=** C **\*** pow**((**1**-**pow**((**m0**\***pow**(**10**,-**31**-**mpow**[**i**])/**m**[**i**]),**2**)),** 0.5**);**

vpow**[**i**]** **=** 8**;**

**while(**v**[**i**]** **<=** 1**){**

vpow**[**i**]** **-=** 1**;**

v**[**i**]** **\*=** 10**;**

**}**

printf**(**"Untuk tegangan %.0f, didapatkan m = %.3f x 10^(%.0f) dan v = %.3f x 10^(%.0f)\n"**,** Tegangan**[**i**],** m**[**i**],** mpow**[**i**],** v**[**i**],** vpow**[**i**]);**

**}**

**}**

b. Plot v/c terhadap V (dibuat dengan excel)

c. Isi file tegangan.txt:

tegangan.txt :

1000000

1100000

1200000

1300000

1400000

1500000

1600000

1700000

1800000

1900000

Output dari input berupa tegangan.txt:

A screen shot of a computer

Description automatically generated with low confidence

3.

Diagram, schematic

Description automatically generated

R = 20 k

C = 132051 pF

Vin = 5 Volt

a. Dengan KVL didapatkan

Dengan menganggap solusi umum dari adalah

Solusi particular:

Solusi homogen:

Dengan menganggap solusi umum dari vout adalah didapatkan

maka didapatkan

Dengan asumsi kondisi awal (t = 0-) kapasitor sudah lama kosong sehingga v(0) = 0 didapatkan A = -5

Saat kondisi tunak atau t maka didapatkan

Dalam domain s dengan menggunakan transformasi laplace didapatkan fungsinya adalah

b.

c.

d. Program untuk menghitung Vout pada setiap saat dan saat kondisi tunak dengan masukan berupa NIM dan t yang akan dicari. Output yang dihasilkan berupa tegangan saat t input dan nilai t saat kondisi tunak.

e. Grafik Vout terhadap waktu (t) (dibuat dengan excel):

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

int main**(){**

char NIM**[**8**];**

printf**(**"Program untuk menghitung nilai Vout rangkaian RC dengan spesifikasi sebagai berikut.\n"**);**

printf**(**"Nilai R adalah digit keempat dan kelima NIM yang akan anda input (dalam kOhm)\n"**);**

printf**(**"Nilai C adalah digit pertama, kedua, ketiga, keenam, ketujuh dan kedelapan NIM anda (dalam pF)\n"**);**

printf**(**"Nilai Vin adalah tetap yaitu 5V.\n"**);**

printf**(**"Masukkan NIM anda : "**);**

scanf**(**"%s"**,** **&**NIM**);**

char Rs**[**2**];**

char Cs1**[**2**];**

char Cs2**[**4**];**

strncpy**(**Rs**,** NIM **+** 3**,** 2**);**

strncpy**(**Cs1**,** NIM**,** 3**);**

strncpy**(**Cs2**,** NIM **+** 5**,** 3**);**

char Cs**[**6**];**

strncpy**(**Cs**,** Cs1**,** 3**);**

strcat**(**Cs**,** Cs2**);**

double R**,** C**;**

R **=** atof**(**Rs**);**

C **=** atof**(**Cs**);**

printf**(**"Dari data NIM anda, maka diperoleh nilai R dan C berturut turut adalah %.0f kOhm dan %.0f pF\n"**,** R**,** C**);**

double time**;**

printf**(**"Masukkan nilai waktu (dalam s) yang ingin anda cari tegangan Voutnya (gunakan . sebagai pemisah desimal) : "**);**

double result**;**

scanf**(**" %lf"**,** **&**time**);**

double time\_constant **=** R **\*** C **\*** pow**(**10**,** **-**9**);**

result **=** 5 **-** **(**5 **\*** exp**(-**time**/**time\_constant**));**

printf**(**"Nilai time constant berdasarkan NIM anda : %lf\n"**,** time\_constant**);**

printf**(**"Pada saat t = %lf s nilai Tegangan Vout adalah = %lf V\n"**,** time**,** result**);**

double tunak **=** 0**;**

result **=** 5 **-** **(**5 **\*** exp**(-**tunak**/**time\_constant**));**

**while(**result **!=** 5**){**

tunak **+=** 0.000001**;**

result **=** 5 **-** **(**5 **\*** exp**(-**tunak**/**time\_constant**));**

**}**

printf**(**"Rangkaian tunak pada saat t = %lf s"**,** tunak**);**

**}**

f. Dari jawaban bagian (a), konstanta waktu yang didapatkan adalah

4.

a.

b.

c. Aplikasi dengan bahasa C tersebut dapat diakses pada link berikut :

<https://github.com/HanifHBK/SolusiUTSPMC>

dengan nama file 4c.c