Penjumlahan Desimal pada Atmega8535

Diperuntukkan untuk memenuhi salah satu tugas praktikum Mata Kuliah Aplikasi Mikrokontroler



Praktikum : Aplikasi Mikrokontroler

Praktikum ke : 3

Tanggal Praktikum : Kamis, 15 Oktober 2020

Tanggal Pengumpulan Laporan : Jum’at, 16 Oktober 2020

Nama dan NIM : 1. Amir Husein (181344003)

Kelas : 3-TNK

Instruktur : 1. Ferry Satria, BSEE., M.T

2. Rahmawati Hasanah, S.ST., M.T

Politeknik Negeri Bandung

Tahun Ajaran 2020/2021

**I. TUJUAN**

* Mahasiswa dapat memahami prinsip penjumlahan desimal pada mikrokontroler Atmega8535
* Mahasiswa dapat memahami konsep faktor koreksi pada penjumlahan desimal

**II. LANDASAN TEORI**

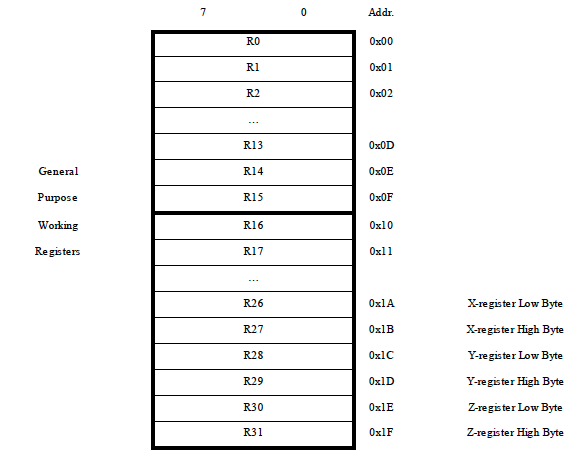
1. Mikrokontroler Atmega8535

Mikrokontroler merupakan suatu chip komputer mini, dimana di dalamnya sudah terdapat sebuah mikprosesor disertai memori, baik itu RAM, ROM, maupun EEPROM. Selain itu, mikrokontroler telah memiliki sistem integrasi Input dan Output (I/O) yang telah dikemas sedemikian rupa pada rangkaian Integrated Circuitnya, sehingga memudahkan dalam melakukan tugas atau operasi tertentu.

Atmega8535 merupakan sebuah mikrokontroler 8-bit yang dikeluarkan oleh perusahaan Atmel pada tahun 2006. Mikrokontroler ini pun memiliki flash memori sebesar 8kb serta EEPROM 512 byte. Selain itu, pada papan mikro ini pun sudah terdapat sebuah ADC dan 3 buah Timer sebagai pencacah waktu pemrosesan data.

1. General Purpose Register

Register ini merupakan register umum yang dapat digunakan sebagai kontrol pengisian dan penyesuaian data. Pada Atmega8535, terdapat 32 general purpose register, yang penamaannya berupa rentang angka dari R0 sampai dengan R31.



Gambar 1. General Purpose Register pada Atmega8535

Terlihat pada gambar diatas, diantara R15 dan R16 terdpat sebuah garis tebal, hal ini menandakan bahwa register R15 kebawah tidak berlaku instruksi *immediate* atau langsung, sedangkan register R16 keatas berlaku. Pada general purpose register ini juga terdapat register khusus sebagai pointer alamat 16-bit yaitu R2 hingga R21 yang terbagi menjadi 3 segmen yaitu pointer X, Y, dan Z.

1. Instruksi ADD dan ADC

Penjumlahan pada Atmega8535 dilakukan dengan memanggil instruksi ADD, serta ADC untuk turut serta menambahkan carry yang terdapat pada flag dalam operasinya. Operand yang dapat digunakan dari kedua intruksi ini adalah R0 sampai dengan R31.

Syntax = ADD Rd, Rr ; Operasi = Rd 🡨 Rd + Rr

Syntax = ADC Rd, Rr ; Operasi = Rd 🡨 Rd + Rr + C

1. Instruksi LPM

Instruksi LPM (Load Program Memory) digunakan untuk melakukan *load* dari memori dengan tujuannya adalah register pointer. LPM digunakan untuk mendapatkan nilai yang tepat dari setiap alamat memori yang dideklarasikan.

1. Faktor Koreksi

Dalam melakukan penjumlahan desimal, komputer akan memproses data tersebut sebagai data biner serta melakukan seluruh kegiatan operasi dengan sistem biner. Untuk melakukan faktor koreksi ini, terlebih dahulu tiap 4-bit dicek, apabila nilainya melebihi 9, maka faktor koreksi dilakukan dengan melakukan penjumlahan dengan 6 atau 0110.

**III. SOAL LAB**

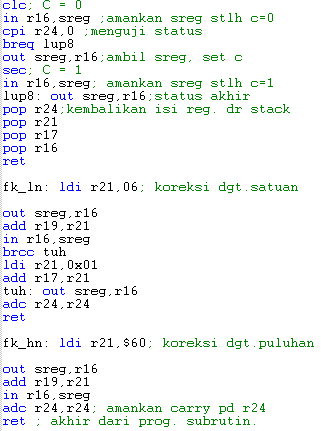
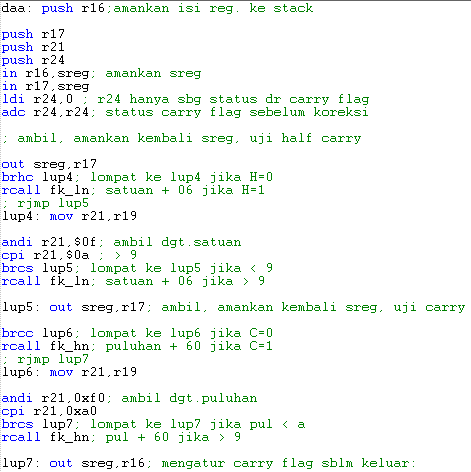
1. Buat program untuk menjumlahkan 2 data Desimal 2 DIGIT yang ada pada register R22 dan R23, simpan hasil penjumlahan pada Register R21:R20.
2. Buat program untuk menjumlahkan 10 data Desimal 2 DIGIT yang ada pada register R0 sampai R9, simpan hasil penjumlahan pada Register R21:R20.
3. Buat program untuk menjumlahkan 2 data Desimal 4 DIGIT yang ada pada register R21:R20 dan R23:R22, simpan hasil penjumlahan pada Register R26:R225:R24.
4. Buat program untuk menjumlahkan 10 data Desimal 4 DIGIT yang ada pada Program Area, simpan hasil penjumlahan pada Register R22:R21:R20.
5. Buat program untuk menjumlahkan 2 data Desimal 6 DIGIT yang ada pada Program area dengan Label var1 dan var2, simpan hasil penjumlahan pada Register R23:R22:R21:R20.
6. Buat program untuk menjumlahkan 2 data Desimal 8 DIGIT yang ada pada Program area dengan Label dat1 dan dat2, simpan hasil penjumlahan pada Register R24:R23:R22:R21:R20.
7. Buat program untuk menjumlahkan 2 data Desimal 10 DIGIT yang ada pada Program area dengan Label dat1 dan dat2, simpan hasil penjumlahan pada Register

R25:R24:R23:R22:R21:R20.

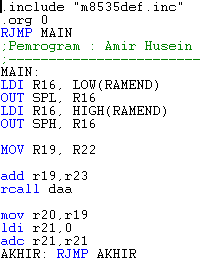
**IV. PROGRAM & HASIL**

* **SUBRUTIN DAA**

Pada Atmega8535, tidak terdapat instruksi DAA untuk melakukan koreksi seperti pada mikroprosesor 8086, karena DAA hanya ada pada mikroporsesor keluarga Intel saja. Maka dari itu, DAA ini perlu dirancang terlebih dahulu dan di *bundle* dalam bentuk subrutin. Subrutin DAA ini sudah disediakan oleh dosen penulis sekaligus pemrogram yaitu Pak Ferry, sehingga subrutin ini siap pakai dan diaplikasikan untuk melakukan koreksi. Program dari subrutin dapat dilihat dibawah ini:



1. Program: Hasil:

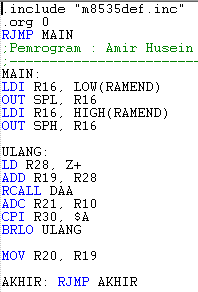
\* 99 + 99 = 198 

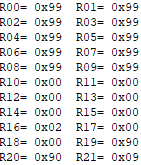


\* 84 + 39 = 123

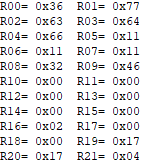


1. Program: Hasil:

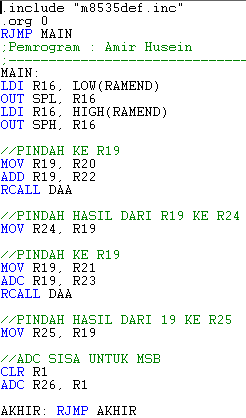
 \* 99+99+99+99+99+99+99+99+99+99 = 990



\* 36+77+63+64+66+11+11+11+32+46 = 417



1. Program: Hasil:

 \* 6355 + 7777 = 14132



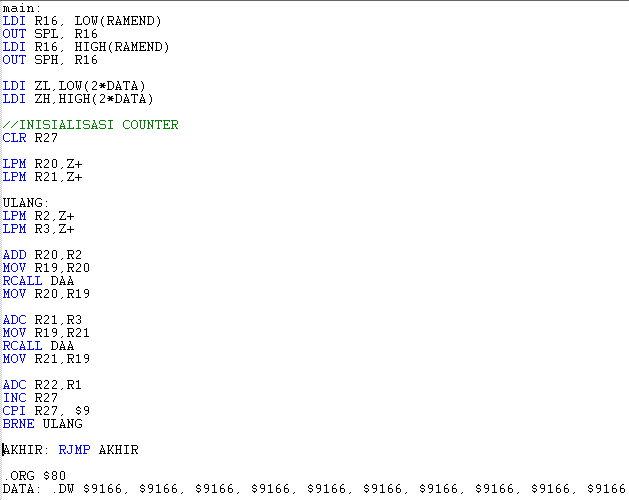
\* 9999 + 9999 = 19998



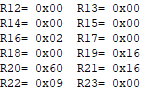
\* 1234 + 9876 = 11110

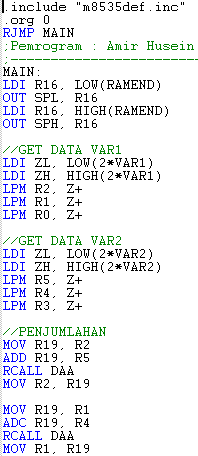


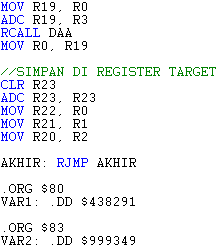
1. Program: Hasil:

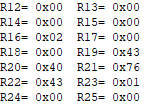
 \* 9166+9166+9166+9166+9166+9166+9166

9166+9166+9166 = 91660

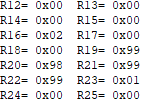


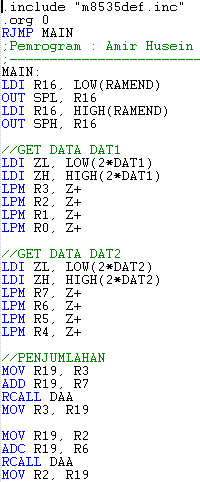
1. Program: Hasil:

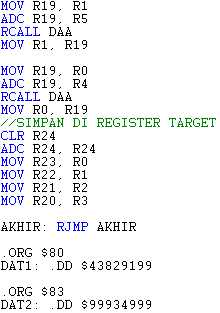
 \* 438291 + 999349 = 1437640



\* 999999 + 999999 = 1999998



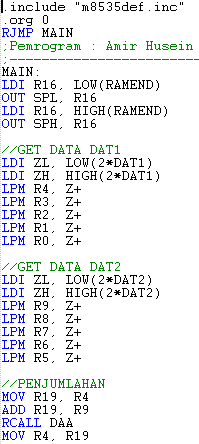
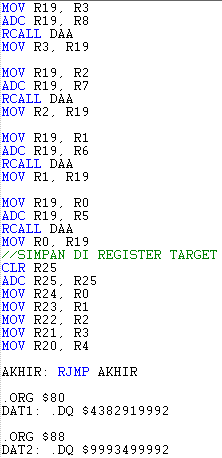
1. Program: Hasil:

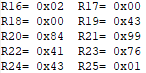
 \* 43829199 + 99934999 = 143764198 

\* 99999999 + 99999999 = 199999998

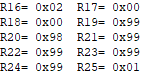


1. Program: Hasil:

 \* 4382919992+9993499992=14376419984



\* 9999999999+9999999999=19999999998



**V. ANALISIS**

1. Pada program ini, data pada R22 perlu dipindahkan dahulu ke register R19, dikarenakan subrutin DAA hanya bisa memproses data yang ada pada register R19. Setelah data dipindahkan, register R19 yang sudah berisi data 2 digit dijumlahkan dengan data 2 digit pada register R23. Hasil yang didapat perlu dicek dan dilakukan faktor pengkoreksian dengan subrutin DAA, maka subrutin tersebut perlu dipanggil dengan perintah RCALL. Setelah itu, hasil penjumlahan pada R19 yang tela dikoreksi dipindahkan ke R20, diikuti dengan instruksi ADC pada R21 untuk menambahkan carry (jika ada). Pada R21 ini tidak perlu dilakukan pengkoreksian dengan DAA dikarenakan angka maksimal yang akan muncul adalah 1 yang merupakan hasil penjumlahan maksimal dari 99 + 99 = 198.
2. Pada program nomor 2 ini berbeda dari sebelumnya, dikarenakan data yang dijumlahkan ada 10 terbentang dari R0 hingga R9. Untuk menyelesaikannya, diperlukan metode looping dengan pointer Z sebagai penunjuk ke alamat memori untuk R0 hingga R9. Untuk mengambil isi dari register R0 hingga R9 digunakan instruksi LD yang dimana akan disimpan di R28 untuk melakukan penjumlahan sementara. Program akan berhenti ketika nilai R30 sudah menyentuh $A, karena data yang dijumlahkan berjumlah 10 data.
3. Pada program ini, terdapat 2 data 4 digit R21:R20 dan R23:R22. R20 perlu dijumlahkan dengan R22, tapi operasi penjumlahan ini perlu dilakukan di R19, kemudian diberi faktor koreksi dengan memanggil subrutin DAA. Setelah itu hasilnya perlu dipindahkan ke R24. Langkah selanjutnya adalah menjumlahkan isi dari R21 dan R23 dengan instruksi ADC, serta tetap operasi ini dilakukan di R19, setelah itu hasilnya dipindahkan ke R25. R26 perlu diberi perintah ADC untuk menyimpan nilai carry yang ada.
4. Program pada soal ini sama seperti pada nomor 2, yang menjadi perbedaan hanya ada pada jumlah digit datanya dan disimpan di program area, bukan data area, sehingga dibutuhkan 2 register tambahan untuk menampung nilai data yang diambil dari program area. Register R27 bertindak sebagai counter dengan nilai maksimal 9, karena ketika nilai R27 telah mencapai 9, proses penjumlahan telah selesai. Data pada program area disimpan pada alamat awal $80.
5. Pada program ini, VAR1 dan VAR2 perlu dideklarasi dan diinisialisasi nilainya dahulu, bisa diawal program maupun diakhir program. Proses ini menggunakan directives .ORG untuk menentukan alamat pada program area, dengan tipe data yang digunakan adalah DD karena data yang dideklarasi merupakan data 6 digit. Langkah selanjutnya adalah melakukan pointing dengan register pointer Z serta melakukan pengisian atau pemindahan nilai pada program area ke GPR. Data 6 digit VAR1 akan disimpan di R0:R1:R2 dan data VAR2 disimpan di R3:R4:R5. Proses penjumlahan hanyalah menjumlahkan tiap register sesuai posisinya seperti halnya melakukan penjumlahan biasa, setelah itu diikuti oleh DAA untuk melakukan pengkoreksian. Perlu diingat bahwa setiap proses penjumlahan perlu dilakukan di R19, sehingga proses pemindahan nilai dari register ke register seringkali terjadi.
6. Program ini sama seperti program pada soal sebelumnya, yang menjadi perbedaan adalah jumlah digit masing-masing data yang akan dijumlahkan. Apabila pada soal sebelumnya 6 digit, maka pada soal ini merupakan 8 digit. Data pada DAT1 akan disimpan di register R0:R1:R2:R3 dan DAT2 di R4:R5:R6:R7, proses pengambilan data ini dilakukan dengan instruksi LPM pada pointer Z yang sudah tertuju pada alamat memori dari DAT1 dan DAT2. Proses penjumlahan hanyalah menjumlahkan tiap register sesuai posisinya seperti halnya melakukan penjumlahan biasa, setelah itu diikuti oleh DAA untuk melakukan pengkoreksian. Perlu diingat bahwa setiap proses penjumlahan perlu dilakukan di R19, sehingga proses pemindahan nilai dari register ke register seringkali terjadi.
7. Program ini sama seperti program pada soal sebelumnya, yang menjadi perbedaan adalah jumlah digit masing-masing data yang akan dijumlahkan. Apabila pada soal sebelumnya 8 digit, maka pada soal ini merupakan 10 digit. Data pada DAT1 akan disimpan di register R0:R1:R2:R3:R4 dan DAT2 di R5:R6:R7:R8:R9, proses pengambilan data ini dilakukan dengan instruksi LPM pada pointer Z yang sudah tertuju pada alamat memori dari DAT1 dan DAT2. Proses penjumlahan hanyalah menjumlahkan tiap register sesuai posisinya seperti halnya melakukan penjumlahan biasa, setelah itu diikuti oleh DAA untuk melakukan pengkoreksian. Perlu diingat bahwa setiap proses penjumlahan perlu dilakukan di R19, sehingga proses pemindahan nilai dari register ke register seringkali terjadi.

**VI. DAFTAR PUSTAKA**

Yusrizal. 2016. *Mikrokontroler Atmega8535.* Yusrizal Weblog. \_\_\_\_\_([https://yusrizalandeslubs.wordpress.com/dasar-elektronika](https://yusrizalandeslubs.wordpress.com/dasar-elektronika/)). Diakses 5 Oktober 2020