Penggunaan Software Multimedia Logic Untuk Mengecek Kebenaran Rangkaian Logika Berdasarkan Peta Karnough

Oleh : Akik Hidayat

Jurusan Matematika FMIPA UNPAD

Abstrak

Arithmetic and Logic Unit (ALU) merupakan bagian komputer yang berfungsi membentuk operasi-operasi aritmatik dan logik terhadap data. Semua elemen lain sistem komputer (control unit, register, memori, I/O) berfungsi terutama untuk membawa data ke ALU untuk selanjutnya diproses dan kemudian mengambil kembali hasilnya.

Selain itu, ALU dan seluruh komponen elektronik di dalam komputer didasarkan pada penggunaan perangkat logik digital sederhana yang dapat menyimpan digit-digit biner dan membentuk operasi logik Boolean sederhana.
\

Pendahuluan

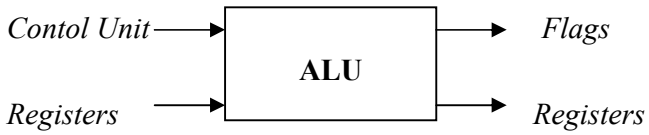
Untuk menyelesaikan permasalahan peta karnough secara matematik digunakan suatu sistem bilangan biner, dalam hal ini akan dicoba dengan komputerisasi yaitu dengan menggunakan software Multimedia Logic, dengan perumusan masalah sebagai berikut :

- Menentukan ALU dalam software.
- Menggunakan ALU untuk mengamati N, Z, V, C.
- Menggunakan Peta Karnough

LANDASAN TEORI

1 Konsep Dasar ALU

Sebagaiman yang telah dijelaskan dalam BAB I bahwa Arithmetic and Logic Unit merupakan bagian komputer yang berfungsi membentuk operasi-operasi aritmatik dan logik terhadap data. Selain itu, ALU dan seluruh komponen elektronik di dalam komputer didasarkan pada penggunaan perangkat logik digital sederhana yang dapat menyimpan digit-digit biner dan membentuk operasi logik Boolean sederhana.



Gambar diatas menjelaskan gambaran secara umum tentang interkoneksi ALU dengan elemen-elemen CPU lainnya. Data diberikan ke ALU di dalam register, dan hasil operasinya disimpan di dalam register. Register-register ini lokasi penyimpanan sementara di dalam CPU yang dihubungkan ke ALU dengan menggunakan lintasan sinyal. ALU juga akan menyetel flag sebagai hasil dari suatu operasi. Misalnya, overflow flag disetel 1 bila hasil komputasi melampaui panjang register tempat flag disimpan. Control unit menghasilkan sinyal yang akan mengontrol operasi ALU, dan pemindahan data ke ALU atau dari ALU.

2 Register Isyarat

Sebelumnya kita akan bahas mengenai register isyarat yang akan mempengaruhi kinerja ALU.

1. C(Carry=pindahan)

Bit pindahan C melakukan dua fungsi yang berbeda dan saling tak bergantungan. Pada prosesor besar kedua fungsi ini akan dilakukan oleh bit-bit yang terpisah:

- *Bit pindahan menyimpan pindahan aritmatika*, yaitu bit kesembilan yang bisa dihasilkan sewaktu operasi aritmatika. Ini terjadi pada hasil 8 bit yang melimpah (*overflow*). Akan tetapi, kata "*Overflow*" mempunyai arti khusus yang akan dijelaskan dibawah.
- *Pindahan digunakan sebagai tumpahan(spill-out)* sewaktu operasi penggeseran dan rotasi. Bila digunakan sebagai tumpahan, pindahan berlaku kembali sebagai bit kesembilan hasil operasi, yang membenarkan penggabungan kedua fungsi ini dalam bit yang sama. Penggabungan ini memudahkan dan memperbaiki kecepatan operasi perkalian dan pembagian aritmatika.

2. V (Overflow = pelimpahan)

Pelimpah menunjukkan bahwa sebuah pindahan aritmatik dalam sebuah kata telah mengubah harga bit paling berarti. Ini mengakibatkan tandakesalahan bila dipakai notasi komplementer dua-dua. Bit 7 dalam komplementer dua-dua (bit paling berarti, atau MSB) menunjukkan tanda harga = 1 untuk negatif, 0 untuk positif. Bila 2 bilangan komplementer dua-dua dijumlahkan, pembawa yang dihasilkan sewaktu penjumlahan atau pengurangan bisa melimpah ke bit tanda. Bila hal ini terjadi, sebuah bilangan negatifdapat berubah menjadi bilangan positif. Bit pelimpahdipakai untuk menunjukkan kejadian ini. Secara matematis pelimpahanadalah OR eksklusif antara bit pindahan (dari bit 7) dan pindahan yang dihasilkan dari bit 6 ke bit 7. Pelimpahan biasanya digunakan hanya bila melakukan aritmetik komplementer dua-dua.

3. N (Negatif) atau S (Sign = tanda)

Bit N dihubungkan langsung ke posisi bit 7 sebuah hasil. Ingatlah bahwa dalamnotasi komplementer dua-dua, harga 1 pada posisibit 7 menunjukkan bilangan negatif, dari sini berasal nama bit status tersebut. Sayang, dalam

kebanyakan mikroprosesor umumnya tak mungkin mengetest bit tertentu dalam register tertentu atau bahkan dalam akumulator. Biasanya satu-satunya bit yang dapat langsung ditest di dalam akumulator adalah bit 7, yang mana efektifnya adalah isyarat N register status. Bilamana bit lain akumulator harus ditest, pemrogram harus melaukan serangkaian pergeseran. Satu penggeseran menempatkan sebuah bit akumulator ke bit pindahan dimana ia kemudian ditest.

Jadi, bit 7 dalam byte manapun siap sedia untuk ditest (karena tersedianya N di dalam register status) dan karena itu merupakan posisi yang disukai untuk menyimpan status bagi penahan masukan/keluaran (input/output latch) atau register manapun. Bila melakukan operasi aritmatika, bit n dipakai untuk menetapkan apakah sebuah bilangan atau hasil operasi, positif atau negatif.

4. H atau AC (Half-Carry =Setengah-Pindahan)

Bit ini dipakai sewaktu operasi *desimal yang dikode biner* (Binary Code Decimal, BCD). BCD adalah notasi yang sering digunakan dalam pemakaian komersil yang membutuhkan hasil yang tepat tanpa adanya kesalahan pembulatan (*round-off- error*) yang disebabkan oleh notasi biasa komplementer dua-dua. BCD menggunakan sebuah kode 4 bit untuk mewakili tiap digit desimal.

5. Z (Zero = nol)

Bit Z berharga 1 bilamana hasil sebuah operasi 0. Ini digunakan oleh instruksi aritmatika dalam menentukan apakah hasil 0 atau tidak, dan digunakan oleh operasi logika seperti COMPARE(bandingkan). Yang terakhir ini mengimplementasi logika XOR antara kata yang ditest dan pola dengan mana bit Z dibandingkan. Bilamana hasil dari suatu perbandingan memuaskan, bit z dibuat berharga 1.

Bit Z sering digunakan oleh instruksi masukkan dan keluaran untuk menentukan apakah isi sebuah bit dalam register telah berubah atau tidak. Hal ini dilakukan secara sederhana dengan meng-XOR harga register dengan harga sebelumnya. Bila tidak ada bit yang berubah, hasil adalah 0. bila sebuah bit telah berubah, hasil operasi XOR bukan 0 dan akan dideteksi oleh bit Z.

6. P (Parity = Paritas)

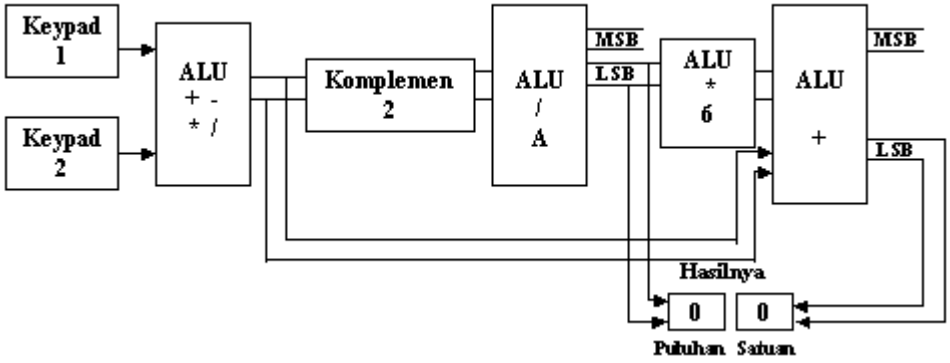
Bit P biasanya tidak disediakan dalam kebanyakan mikroprosesor, tetapi tersedia dalam 8080 yang sebelumnya. Paritas digunakan untuk mendeteksi apakah data telah dikirim dengan betul atau tidak. *Prinsip Paritas* adalah menghitung jumlah satu yang terdapat pada kedelapan bit. Pola paritas genap melengkapi jumlah 1 pada kata 7 bit dengan menambahkan sebuah harga 0 atau harga 1 sehingga jumlah seluruhnya adalah genap. Sebaliknya pola paritas ganjil membuat kedelapan bit sedemikian sehingga jumlah 1 seluruhnya yang terdapat dalam kedelapan bit adalah ganjil.

7. Bit-bit Status Lainnya.

Bit-bit status lainnya dapat disediakan didalam register isyarat. Khususnya, sebuah bit interupsi, yang biasanya berfungsi sebagai *mampu interupsi*, dapat disediakan. Bila bit interupsi disiapkan, interupsi luar akan diterima. Bila bit interupsi tak disiapkan (0), interupsi tidak diizinkan, mereka disebut *dihalangi (masked)*. Informasi status tambahan bisa juga digabungkan dengan register isyarat, jadi memudahkan pengetesan flip-flop lainnya yang menyimpan kejadian intern prosesor. Itulah kenapa register ini juga disebut *PSW (Program Status Word = kata status program)*. PSW menyimpan informasi status prosesor secara lengkap untuk program yang sedang dilaksanakannya.

8.ancangan Arsitektur

Arsitektur yang mau dirancang nantinya bisa melakukan program aritmatik terutama untuk penjumlahan, pengurangan, perkalian, dan pembagian. Serta hasilnya bisa berupa negatif, overflow, carry, maupun zero. Adapun untuk rancangan arsitekturnya adalah sebagai berikut :



Untuk nilai yang nantinya dimasukkan adalah berupa bilangan hexadesimal. Untuk cara kerjanya pertama kita ketikan angka yang ada pada keypad satu dan dua, kemudian kita atur aritmatik yang mau kita cari yaitu penjumlahan, pengurangan, perkalian, maupun pembagian. Setelah itu hasilnya akan masuk dalam komplemen 2 serta masuk ke dalam ALU lagi untuk dijumlahkan. Untuk di komplemen 2, hasil hexanya akan dibagi dengan A, kemudian akan keluar melalui LSB (*Least Significant Bit*). Kemudian hasilnya akan digunakan sebagai keluaran dan juga bisa digunakan untuk melaukan aritmatik lagi. Hasil yang digunakan sebagai keluaran akan digunakan sebagai **puluhan**, dan yang kedua melakukan aritmatik dengan dikalihkan 6. Hasil perkalian tersebut akan dijumlahkan lagi dengan hasil dari ALU yang sebelum masuk ke dalam komplemen dua. Kemudian hasilnya akan keluar dalam bentuk LSB (*Least Significationt Bit*) dalam bentuk **satuan**. Itu merupakan cara kerja dari rancangan arsitektur yang akan dibangun.

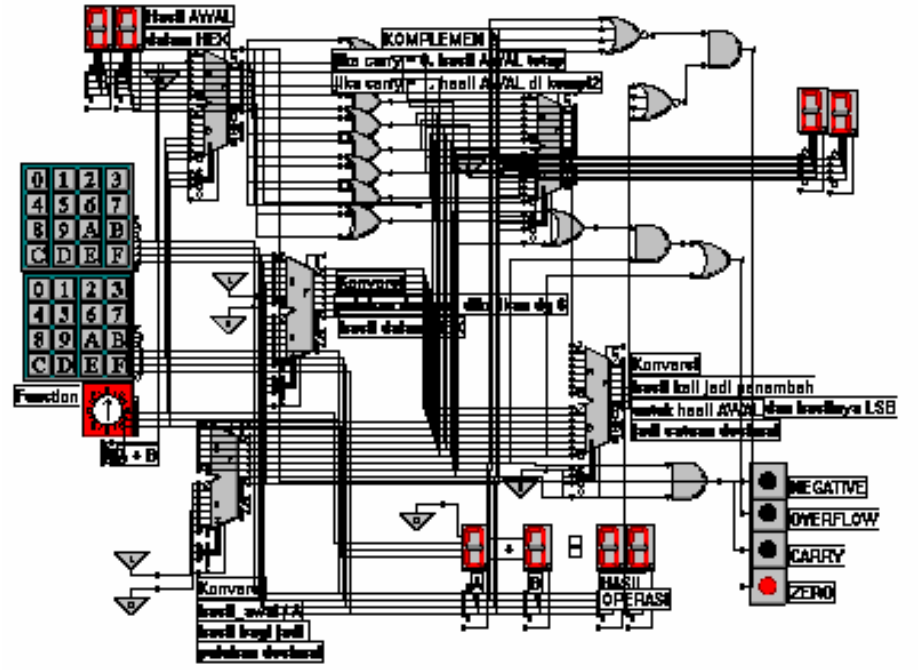
HASIL DAN PEMBAHASAN

1. Langkah percobaan

Untuk mengimplementasikan rancangan arsitektur yang ada pada BAB II, kita terlebih dahulu harus mengikuti langkah-langkah sebagai berikut :

1. Untuk langkah awal kita harus menginstall program **Multimedia Logic**. Sebenarnya masih banyak program yang mendukung untuk membuat rancangan arsitektur yang seperti ada pada BAB II diatas, tapi untuk aplikasi ini cukup menggunakan multimedia logic, selain muda untruk dibangun rancangannya juga kapasitasnya tidak terlalu banyak yaitu kurang lebih 2,13 MB, sehingga tidak terlalu banyak membebankan bagi harddisk.
2. Dan selanjutnya kita bisa langsung bekerja untuk membangun program yang sesuai dengan rancanagn arsitektur yang telah dibangun pada BAB II sebelumnya.

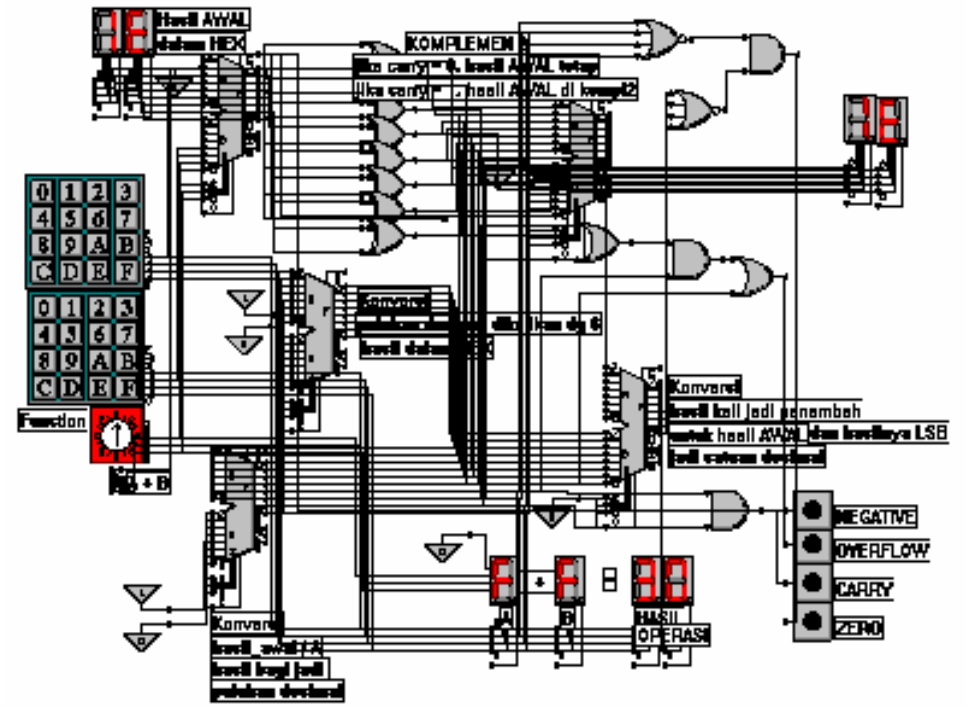
Dan dengan melihat rancangan arsitekturnya tampilan program yang telah dibuat adalah sebagai berikut :



2. Hasil Percobaan

Untuk lebih jelasnya dari cara kerja program diatas, kita akan mengeksekusi program tersebut, yaitu :

1. Penjumlahan



Dari contoh percobaan diatas kita dapat melihat penjumlahan antara dua buah bilangan hexa yaitu **F + F** hasil operasinya adalah **30** setelah dikonversi kedalam desimal. Sedangkan hasil hexanya yaitu **1E**. Untuk lebih jelasnya bisa dilihat seperti dibawah ini :

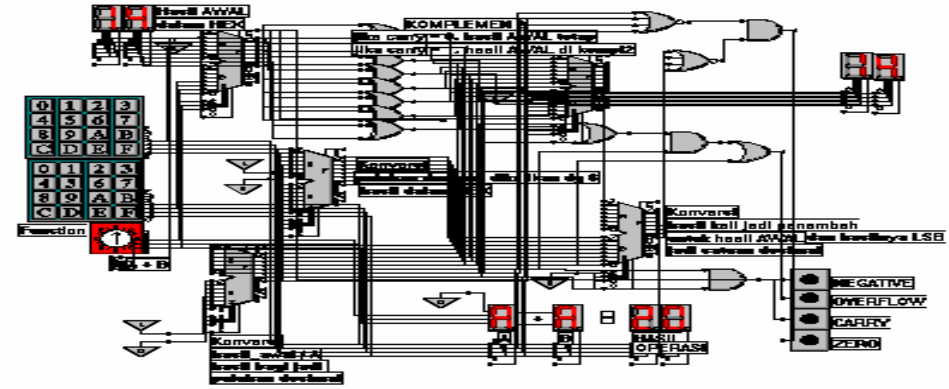
F + F = 1E : positif (+) **1E / A = 3** → diambil sebagai puluhan desimal (**3** **0**)

3 * 6 = ~~12~~ → **1E**

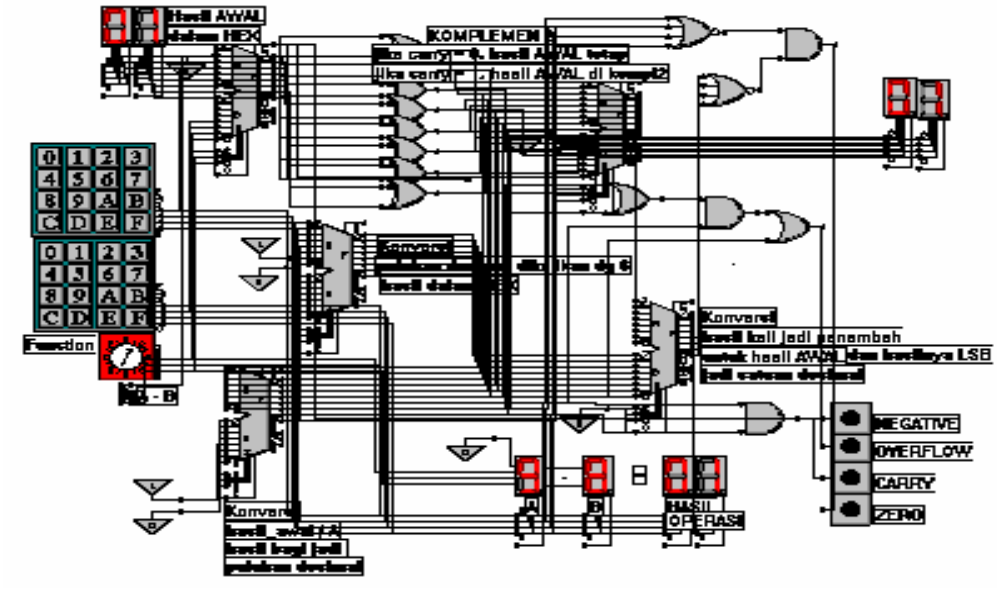
❖

12 +

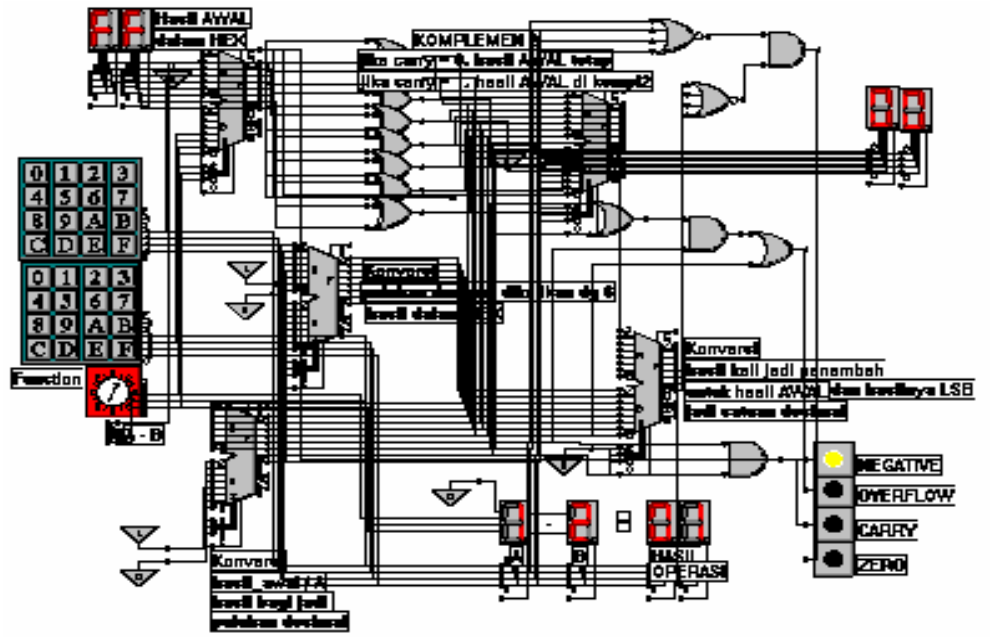
30 , **0** sebagai satuan



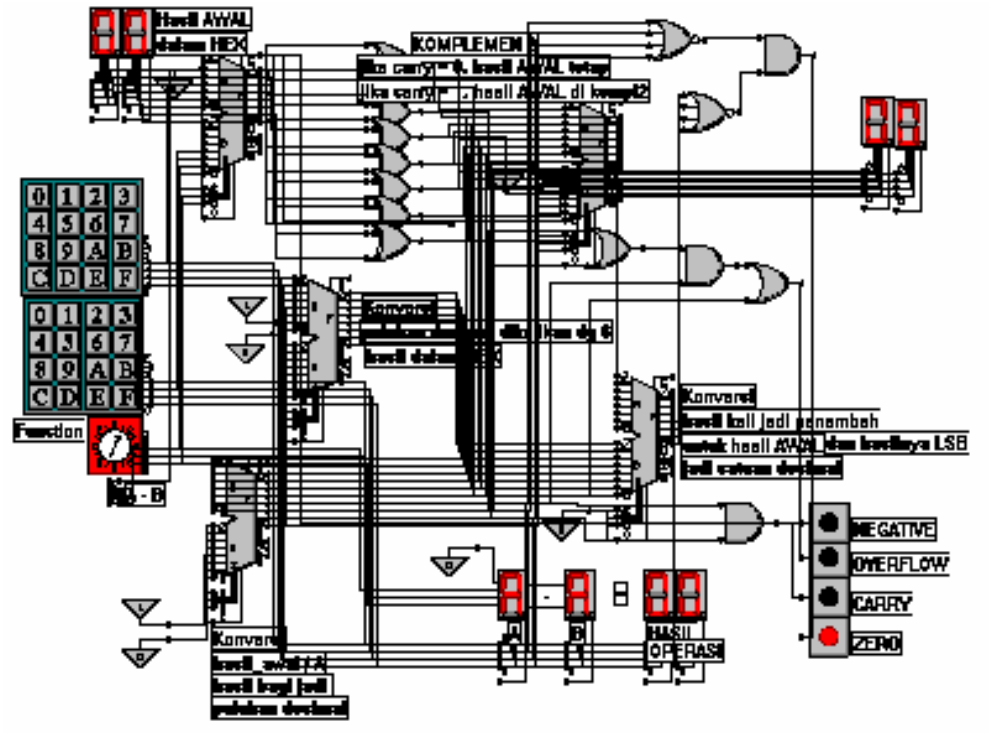
Pengurangan hasilnya positif:



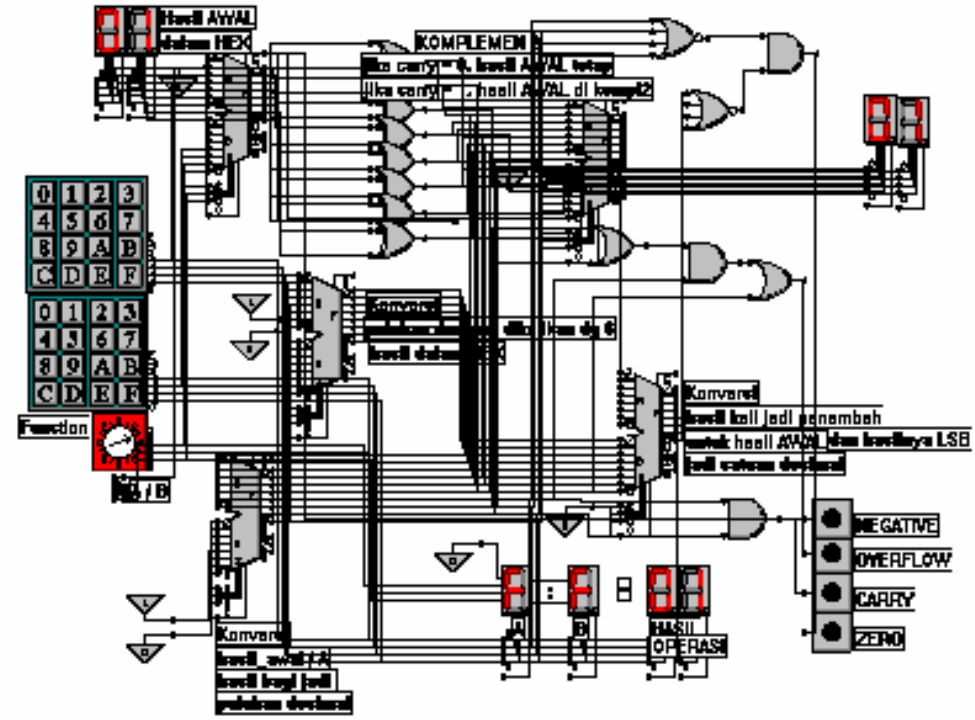
Pengurangan hasilnya negatif:



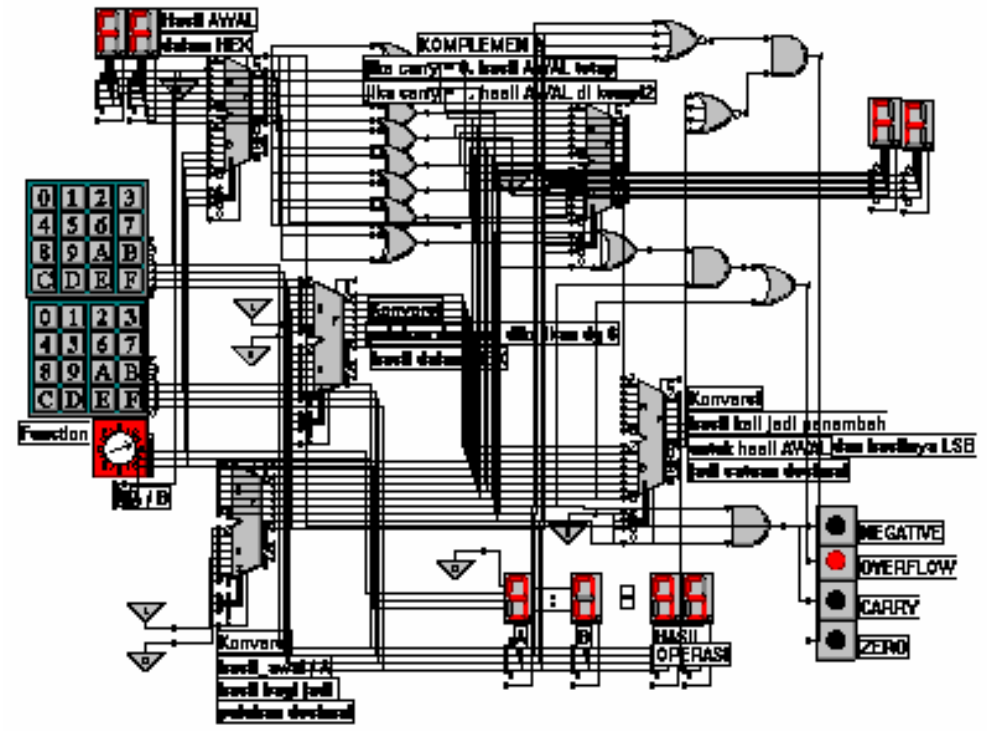
Contoh Zero :



Pengurangan



Pengurangan yang hasilnya Overflow



Kesimpulan.

Dari Percobaan diatas dapat disimpulkan sbb:

engan menggunakan alata bantu komputer dengan software multimedia logic ternyata akan lebih mudah,cepat serta dapat di rubah-rubah suatu fungsinya dan akan mendapatkan hasil yang akurat dibandingkan dengan menggunakan konsep matematika.

DAFTAR PUSTAKA

Alxanis, Nikitas., *Design Of Micoprocessor Based Systems*, Prentic Hall, Engleood Cliffs, New Jersey, 1993, pp 246-258

Hill, Frederick J., and Peterson, Gerald R., *Introduction to Switching Theory and Logical Design*, University of Arizona, 1981.

Stalling, Wiliam., *Computer Oganization and architecture Dsigning for performanc*, sixth dition, 2003.

Susanto, *Belajar Sendiri Pemrograman dengan Bahasa Assembly*, Elex Media Komputindo,1995