Judul: Hamming Error Correcting Code

Deskripsi Masalah:

Dalam Arsitektur dan Organisasi Komputer, kita akan mempelajari pengoreksian error dengan cara (kode) yang ditemukan oleh Hamming (atau disebut juga *Hamming Error Correcting Code*). Pengoreksian error menggunakan kode Hamming merupakan cara pengoreksian yang cukup sederhana. Setelah dipelajari lebih jauh, ternyata cara yang dilakukan Hamming menerapkan Finite State Machine.

Apa itu Hamming Error Correcting Code?

*Hamming Error Correcting Code* adalah cara pengoreksian error yang dikirimkan dalam transmisi data yang dilakukan oleh komputer, khususnya bagian memori dan penyimpanan data. Cara pengoreksian ini dapat mengecek error hingga x-1 bit error dan dapat mengoreksi hingga (x-1)/2 kesalahan. Variabel x tersebut adalah jarak Hamming (Hamming distance) antara setiap pasangan dalam satuan kode.

*Hamming distance* atau jarak Hamming adalah jumlah bit yang harus diubah. Jarak Hamming biasa digambarkan dengan sebuah kubus n-dimensi. Untuk mendeteksi d-bit error, diperlukan Hamming distance yang lebih kecil dari d, sedangkan untuk mengoreksi d-bit error, diperlukan Hamming distance yang lebih kecil dari 2d.

(gambar Hamming Distance)

Hamming distance berukuran 2k + 1 diperlukan untuk dapat mengoreksi k error di setiap susunan data (yang disebut juga kata).

Kriteria yang diperlukan untuk Hamming Error Correction:

1. Jika semua angka dalam sindrom adalah angka 0, berarti tidak ada error yang terdeteksi.
2. Jika terdapat satu bit dengan angka 1 dalam sindrom, maka sebuah error terjadi dalam salah satu check bit. Tidak perlu pengoreksian.
3. Jika terdapat lebih dari satu bit dengan angka 1 dalam sindrom, maka urutan sindrom tersebut yang menjadi posisi error dalam data bit.

Metode Pengerjaan:

1. Tentukan panjang kodenya

Misalkan kita memiliki data words sepanjang m = 8 dan 2k -1 > 8 + k, maka k (check bit) adalah 4. Tambah 4 check bit ke dalam code words (code words menjadi 12-bit).

1. Tentukan posisi penempatan check bit

Jika diperlukan, buatlah kumpulan angka yang terdiri dari pangkat dua.

Contoh:

1 = 20, 2 = 21, 3 = 21 + 20, dst.

1 (20) menyusun setiap angka ganjil. 2 (21) menyusun angka 2, 3, 6, 7, 10, dan 11.

Parity bit akan diletakkan pada tempat dari data words yang selanjutnya akan bertambah sebesar 2n (20, 21, 22, dst.)

1. Melakukan operasi XOR

Setiap parity bit akan mengecek posisi data bit yang bersangkutan (contoh: bit 1 mengecek posisi ke-3, ke-5, ke-7, ke-9, dan ke-11).

Bilangan-bilangan tersebut dilakukan operasi XOR satu per satu.

Solusi:

Finite State Machine akan diterapkan saat mencari error pada check bit dan data bit.

(gambar FSM khusus HECC)

Finite State Machine di atas adalah jenis FSM yang tidak mengeluarkan output (Finite State Automata). Proses yang dilakukan dalam FSM tersebut adalah membuat acceptive area berdasarkan check bit dari data yang direquest dari memori oleh processor dan memasukan input yang berupa check bit dari data yang diterima oleh processor. Jika salah, maka akan diarahkan pada S5 (trap) dan menunjukkan bahwa code words yang dimasukkan memiliki error.

Kesimpulan:

Hamming distance menunjukkan banyaknya error dalam program. Error tersebut ditemukan dengan cara Hamming yang menggunakan Finite State Automata dalam membuat acceptive area dari data yang diminta oleh program (request bit).