Nama : Muhammad Rasyid Ridho

Kelas : A 2016

NIM : 1601731

PEMAHAMAN MENGENAI POINT 12 SERTA SIMULASI

Datum adalah satuan data yang paling kecil.

Comments adalah penjelasan bagaimana proses tersebut dijalankan.

AC atau Accumulator adalah tempat penyimpanan data sementara yang berguna dimana data tersebut adalah hasil operasi/operan.

M atau Memory adalah tempat penyimpanan data yang sebelumnya sudah diproses dan berasal dari AC.

IAS memiliki 1000 tempat penyimpanan dimana setiap penyimpanan memiliki 40 bit words. 2 x 20 bit instruksi (8 bit opcode & 12 bit address).

|  |  |  |
| --- | --- | --- |
| **TABLE 1.3 VECTOR ADD PROGRAM** | | |
| **Location** | **Datum/Instruction** | **Comments** |
| 0 | 999 | Count |
| 1 | 1 | Constant |
| 2 | 1000 | Constant |
| **Inner Loop for Each Add** | | |
| 3L | AC <− M(3000) | Load **B***i* |
| 3R | AC <− AC+M(2000) | **B***i* + **A***i* |
| 4L | M(4000) <− AC | Store AC |
| **Loop Test and Continue/Terminate** | | |
| 4R | AC <− M(0) | Load count |
| 5L | AC <− AC ‑ M(1) | Decrement count |
| 5R | If AC >= 0, go to M(6,0:19) | Test count |
| 6L | Go to M(6,20:39) | Halt |
| 6R | M(0) <− AC | Store count |
| **Address Adjustment (Decrement)** | | |
| 7L | AC <− AC+ M(1) | Increment count |
| 7R | AC <− AC + M(2) | Add constant |
| 8L | M(3,8:19) <− AC(28:39) | Store modified address in 3R |
| 8R | AC <− AC+ M(2) | Add constant |
| 9L | M(3,28:39) <− AC | Store modified address in 3L |
| 9R | AC <− AC + M(2) | Add constant |
| 10L | M(4,28:39) <− AC | Store modified address in 4L |
| 10R | Go to M(3,20:39) | Unconditional branch to 3L |

**Table 1.3 vector Add Program**

Lokasi 0 adalah lokasi memori ke 0 pada sebuah word yang memiliki nilai data/instruksi senilai 999 dan bersifat berubah-ubah.

Lokasi 1 adalah lokasi memori ke 1 pada sebuah word yang memiliki nilai data/instruksi senilai 1 dan bersifat konstan/tetap.

Lokasi 2 adalah lokasi memori ke 2 pada sebuah word yang memiliki nilai data/instruksi senilai 1000 dan bersifat konstan/tetap.

3L dan 3R itu merupakan lokasi memori ke 3 pada sebuah instruksi sebelah kiri (L = Left) dan pada sebelah kanan (R = right), masing-masing memiliki kapasitas 20 bit. Begitu juga untuk setiap lokasi, hingga lokasi ke 10 (3L - 10L) dan (3R - 10R).

**Inner loop for each add**

Tahap ini menjelaskan proses penjumlahan berulang antara AC dengan M.

AC <- M(3000) adalah proses pemberian isi yang berada pada alamat ke 3000 kedalam AC dari M(3000) dalam vektor Bi.

AC <- AC + M(2000) adalah proses penjumlahan nilai antara AC da nisi memori yang berada pada alamat ke 2000 kedalam AC. AC berada pada vektor Bi dan M(2000) berada pada vektor Ai.

M(4000) <- AC adalah proses transfer nilai dari AC ke memori yang berada pada alamat ke 4000. M(4000) berada pada vektor Ci (Ci hasil dari penjumlahan Ai + Bi).

**Loop test and continue/terminate**

AC <- M(0) itu merupakan proses pemberian nilai dari isi memori yang berada pada alamat ke 0 kedalam AC dimana M(0) itu bernilai 999.

AC <- AC – M(1) merupakan proses decrement atau pengurangan antara AC dengan M(1) yang memiliki nilai konstan yaitu 1. Pada proses ini nilai pada AC itu akan dikurangi sebesar nilai M(1) secara berulang hingga data berakhir/ habis. Misal 999 – 1 = 998, 998 – 1 = 997, dan seterusnya.

If AC >= 0 goto M(6, 0:19) merupakan salah satu proses pengkondisian dimana keterangan tersebut menjelaskan bahwa jika nilai AC lebih dari atau sama dengan 0 maka pergi/masuk ke lokasi memori ke 6 yang berada pada bit ke 0 sampai ke 19. Bit 0 – 19 itu berada tempat penyimpanan untuk opcode.

M(0) <- AC merupakan proses pemberian nilai (assignment) dari AC ke M(0). Store count ini merupakan proses penyimpanan data Ac ke M(0).

**Adress Adjustment (decrement)**

AC <- AC + M(1) merupakan penambahan nilai AC dengan (M1), proses ini dinamakan dengan proses increment atau meningkat.

AC <- AC + M(2) merupakan penambahan nilai AC dengan M(2) yaitu dengan data konstan (1000 data) dan setelah itu berikan nilai penjumlahan tadi ke AC yang baru (update data AC).

M(3, 8:19) <- AC(28:39) merupakan proses pemberian nilai dari AC yang berada pada bit ke 28 hingga 39 (instruksi kanan dan pada bit address) ke lokasi memori yang berada pada memori ke 3 yang berada pada bit ke 8 hingga ke 19 (instruksi kiri dan pada bit adress). Proses ini merupakan penyimpanan alamat yang telah dimodifikasi dari AC (tempat penyimpanan sementara) ke M (memori).

AC <- AC + M(2) merupakan penambahan nilai AC yang terbaru dengan M(2) yaitu dengan data konstan (1000 data).

M(3, 28:39) <- AC merupakan pemberian nilai dari AC (penyimpanan sementara) ke memori (tempat penyimpanan asli) pada lokasi ke 3 yang dimana terletak pada bit ke 28 hingga 39 (instruksi kanan dan pada bit address). Proses ini merupakan penyimpanan alamat dari AC (tempat penyimpanan sementara) ke M (memori).

AC <- AC + M(2) merupakan penambahan nilai AC yang terbaru dengan M(2) yaitu dengan data konstan (1000 data). Data AC akan terus ditambah/diupdate.

M(4, 28 - 39) <- AC merupakan pemberian nilai dari AC (penyimpanan sementara) ke memori (tempat penyimpanan asli) pada lokasi ke 4 yang dimana terletak pada bit ke 28 hingga 39 (instruksi kanan dan pada bit address). Proses ini merupakan penyimpanan alamat dari AC (tempat penyimpanan sementara) ke M (memori).

Go to M(3, 20 : 39) merupakan proses pindah atau pergi ke lokasi memori ke 3 yang dimana terletak pada bit ke 20 hingga ke 39 (instruksi kanan dan pada bit address).

Contoh Simulasi :

2 + 3 = 5

|  |  |  |  |
| --- | --- | --- | --- |
| Alamat Memori | **0** | **1** | **2** |
| Nilai Memori | 999 | 1 | 1000 |

Alamat memori yang bernilai 1 dan 2 itu adalah sebuah variable batas, awal(alamat memori bernilai 1) dan akhir(alamat memori yang bernilai 1000).

PROCESSOR

AC (Accumulator)

= Nilai Memory

= Alamat memori

Dikirim ke RAM (M)

Dikirim ke CPU (AC)

Dikirim ke CPU (AC)

2

5

3

2000

5000

3000