



98

NAMA : Tirta Aurelia hijaya..... Mata Kuliah : ..OAK.....  
NIM : 24060122130041..... Dosen : Bu. Rismiyah.....  
No. Absen / Kursi : 53..... Hari / Tanggal : Rabu, 5 April 2023.....  
Prodi SI / D3 : Informatika..... Tanda Tangan :

1) a) Komponen komputer:

1. CPU

untuk mengatur operasi yang berjalan di komputer

2. Main Memory

untuk menyimpan (tempat menyimpan) data dan instruksi

3. I/O (input output)

untuk mengirim data dari komputer ke lingkungan luar (environmentnya)

4. Interconnection

untuk menghubungkan masing-masing komponen komputer

b) Instruction:

\* Register

523 5755

524 2756

525 1757

Memory lama\*

PC 523

AC 0003

IR 2754

Memori baru

755 0001

756 0002

757 0005

PC 523

AC 0004 (0003 + 0001)

IR 5755

Penjelasan: di PC ada instruksi 523, setelah kita melihat di register ternyata 523 berisikan 5755 yang kemudian dimasukan di IR. Instruksi di IR kini menjadi 5755, setelahnya kita lihat perintah 5 adalah untuk menambahkan yang ada di AC dengan angka yang berada di 755, yakni 0001, sehingga pada AC sekarang terdapat 0004 (0003 + 0001). setelahnya di PC + 1 untuk mengeksekusi instruksi selanjutnya.

\* Memori lama\*

PC 523

AC 0004

IR 5755

Memori baru

PC 524

AC 0004

IR 2756

Penjelasan: setelah instruksi sebelumnya, di PC sekarang terdapat angka 524, setelah melihat register ternyata 524 berisikan 2756 yang kemudian dimasukan di IR. Instruksi di IR kini menjadi 2756, setelahnya kita lihat perintah 2 ialah untuk menyimpan data dari AC ke memory dengan angka yang berada di 756, yakni 0002, sehingga pada AC sekarang terdapat 0004 dan di memory dengan alamat 756 berisikan 0004. setelahnya di PC + 1 untuk eksekusi instruksi berikutnya.

Register baru

755 0001

756 0004

757 0005



* Register	Memory lama*
523 5755	PC 524
524 2756	AC 0004
525 1757	IR 2756

.... Memory baru

755 0001	PC 525
756 0004	AC 0005
757 0005	IR 1757

Penjelasan: setelah instruksi sebelumnya, di PC sekarang terdapat angka 525. setelah kita melihat angka 525 di register, ternyata bernilai 1757 yang kemudian masuk di IR sehingga IR bernilai 1757. perintah 1 ialah untuk me-load AC dari memori dengan alamat 757, yang bernilai 0005. kemudian 0005 disimpan dalam AC. dan di PC +1 untuk eksekusi selanjutnya. inilah kondisi akhir setelah instruksi 525 dieksekusi (lihat diagram di atas).

2) Raid ialah salah satu jenis external memory yang digunakan untuk back-up data saat terjadi kerusakan sehingga data tidak serta merta menghilang. OS memandang raid sebagai satu kesatuan dan membaginya per jumlah individu Raid. Raid memiliki beberapa level, yakni:

- |                  |   |
|------------------|---|
| Tidak Independen | Raid level 0 : data masih dalam bentuk stripe, jumlahnya banyak dengan 1st yang berbeda sehingga tidak ada redundansi, hanya mengenal konsep paralel. |
|                  | 1: data masih dalam bentuk stripe, sudah ada redundansi, ada 1 yang mirip dengan 1st data sama, sama seperti raid 0 (ini masih mahal).                |
|                  | 2: data dlm bentuk word, jumlahnya banyak, yang disimpan data partnyanya  |
| Ada downmenya    | 3: data dalam bentuk word, jumlahnya 1, yang disimpan data partnyanya.  |
|                  | 4: data dalam bentuk block-block, sudah independen  |
|                  | 5: data partnyanya di distribusikan (tidak pada 1 tempat), data disimpan dalam bentuk word.   |
|                  | 6: fungsi data paritas ada 2, sehingga bisa handle lebih dari 1 raid yang rusak (maksimal 2)  |

3) Data pada memory (sudah di translate)\*

a) Alamat → Direct Mapping (8 tags, 14 line, 2 word).

00	0000	11111111	FF	3FFC	12345678
00	0001	22222222	FF	3FFD	13245678
01	3C03	44444444	FF	3FFE	14235678
01	3C04	55555555	FF	3FFF	15234678





sehingga	tag	isi data	Line number
	00	22222222	0001
	01	55555555	3C04
	FF	15234678	3FFF

b) Asotatif Mapping (22 tag, 2 word).

isi data	Alamat	word	Tag.
22222222	0000 0000 0000 0000 0000 0100		0000001
55555555	0000 0001 1111 0000 0001 0000		007C04
	0 0 7 C 0 4		
15234678	1111 1111 1111 1111 1111 1100		3FFFFF
	3 F F F F F F		

f) a) Teknologi semikonduktor yang cocok:

- 1) cache (SRAM) karena SRAM aksesnya lebih cepat daripada DRAM sehingga bisa digunakan dengan clock prosesor yang cepat
- 2) Main Memory (DRAM) karena DRAM cocok untuk kapasitas data yang besar. Selain itu lebih murah
- 3) SSD (NAND flash memory) karena NAND bernilai logika 1 jika seluruh transistornya mati. NAND menggunakan bus serial dan lebih padat dibandingkan dengan NOR flash memory dan kecepatan tulis NAND flash memory lebih baik
- 4) Internal memory untuk embedded system (NOR flash memory) karena dengan hubungan paralel, nilai logika 0 bisa dicapai selama salah satu transistor mati. Keunggulannya bisa langsung menuju word yang dimaksud bukan bloknya saja seperti NAND flash memory.

b) Data 10100101

Parity 0101

\* Buat tabel

12	11	10	9	8	7	6	5	4	3	2	1
D1	D7	D6	D5	C4	D4	D3	D2	C3	D1	C2	C1
1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001

$$C_1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$C_2 = D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$C_3 = D2 \oplus D3 \oplus D4 \oplus D8 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$C_4 = D5 \oplus D6 \oplus D7 \oplus D8 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

sehingga = 0011

0101 XOR

0110 → data yang sudah ada di posisi no 6 yaitu D3.

Data awal = 10100101

Data akhir = 10100001

5)

a) 5 state:

→ New : data baru, disasong tag disini, masih di hardisk.

→ Admit : setelah new, di load ke memory, nunggu antrian eksekusi.

long term.

(load ke memory dr hardisk)

→ Running : keadaan ketika data diproses.

short term.

→ Interrupt : ketika di I/O, biasanya ke medium term (jika menggunakan swapping)

(memutuskan program mana yang dieksekusi data) medium term.

→ Release : data dikeluarkan

(pada swapping, di load ke medium term sehingga komputer ada ruang kosong short term.)

(mengadwalkan kapan program dikeluarkan)

untuk eksekusi program lain)

b) efek positif demand paging: Mendapat virtual memori (user merasa seolah-olah memiliki kapasitas memori lebih besar dibandingkan seharusnya) karena metode paging + swapping dimana tidak semua page disimpan dalam memory. Sebagian bisa disimpan ke hardisk yang nanti ketika dibutuhkan dapat dianggil kembali. sehingga bisa menyimpan banyak data.

efek negatif demand paging : thrashing

⇒ ketika terlalu banyak menggunakan demand paging, prosesor akan sibuk swapping saja data yang kita butuhkan akibatnya, clock prosesor habis untuk swapping dan butanya mengeksekusi program kita. keadaan inilah yang disebut dengan thrashing. (swapping untuk mereplace data di memori dengan data di hardisk).

NB: saya tidak pakai kertas corel"an. kertas corat-coratnya kasar.