

# 8

## REGISTER

Register adalah sekumpulan sel biner yang dipakai untuk menyimpan informasi yang disajikan dalam kode-kode biner. Penulisan (pemuatan) informasi itu tidak lain daripada penyetelan keadaan kumpulan flip-flop dalam register itu secara serentak sebagai satu kesatuan. Setiap flip-flop dalam register membentuk satu sel dan dapat menyimpan 1 angka biner (binary digit, bit). Satu register yang tersusun atas  $n$  sel dapat menyimpan  $n$  bit data yang dapat menyatakan salah satu dari  $2^n$  macam kode yang dapat dibentuk dari  $n$  bit tersebut, yang untuk data desimal dapat berharga dari 0 sampai dengan  $2^n - 1$ . Register 8 bit, misalnya, dapat menyimpan salah satu dari 256 macam kode atau harga desimal 0 sampai dengan 255. Register dapat menyimpan informasi dalam kode biner dan menampilkannya kembali dan dikatakan dapat melakukan operasi baca dan tulis.

Dalam lingkungan komputer digital, register menjadi bagian yang sangat penting. Dalam lingkungan ini, istilah register digunakan khusus bagi register dalam prosesor yang mempunyai fungsi khusus dengan kemampuan tambahan di samping kemampuan baca/tulis. Register yang hanya mempunyai kemampuan baca/tulis disebut memory (pengingat) atau storage (penyimpan). Penyimpanan data dalam memori bersifat jauh lebih permanen dibanding penyimpanan dalam register. Pada umumnya, dalam satu prosesor disediakan register dalam jumlah yang sangat terbatas sedangkan memori disediakan dalam ukuran yang sangat besar, dalam ukuran KB (Kilo Byte) sampai MB (Mega Byte) yang masing-masing byte terdiri atas 8 sel. Dalam pandangan rangkaian logika, memori dan register khusus tetap sama dan disebut register. Bab ini akan menguraikan register pemalang, memori dan register geser.

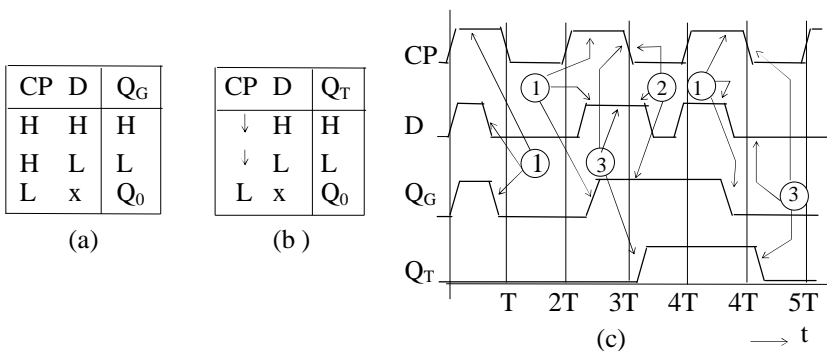
### 8.1 Register Pemalang

Register yang paling sederhana adalah register yang dapat ditulis (dimuat) dengan data dan data itu tetap tersedia pada keluarannya. Register demikian banyak digunakan sebagai pengantara dalam menghubungkan satu bagian yang berkecepatan tinggi dengan bagian yang berkecepatan yang lebih rendah dalam sistem digital, umpamanya sebagai pengantara pusat pengolahan dengan peranti masukan/keluaran (I/O device). Dalam hal ini, register itu berfungsi sebagai penyangga (buffer). Data yang diberikan pada masukan disimpan dan dipalang di dalam register. Setelah pemalangan terjadi, keadaan keluaran register tidak akan berubah walaupun masukannya berubah. Jadi, satu data yang telah dipalang akan tetap tersedia bagi bagian yang membutuhkan data itu dalam waktu yang lebih

lama tanpa sumber data harus mempertahankan keadaannya di bagian masukan. Kejadian ini dapat dibandingkan dengan sebarisan manusia yang memasuki suatu ruangan tertutup melalui pintu masuk. Jumlah dan perilaku manusia yang sudah di ruangan tidak akan terpengaruh oleh barisan yang di luar ruangan setelah pintu masuk dipalang. Karena sifat penyanggaan (buffering) dan proses penulisan secara pemalangan itu, register demikian dinamakan penyangga pemalang (Latch Buffer). Jadi, register pemalang pada dasarnya merupakan sekumpulan flip-flop yang ditabuh secara serentak. Penabuh berfungsi sebagai kendali pemalangan.

Ditinjau dari cara pelaksanaan pemalangan, register pemalang dapat dibedakan atas 2 macam, yaitu: *transparan* (*transparent*) dan *terpicu* (*triggered*). Kedua jenis pemalang ini pada umumnya disusun dari flip-flop D yang telah diuraikan dalam Bab 6 dan berbeda hanya pada sifat pemalangannya.

Dalam pemalang transparan yang disebut juga pemalang peka aras (level sensitive), keadaan keluaran mengikuti keadaan masukan selama penabuh berkeadaan aktif dan keadaan keluaran dipalang pada keadaan masukan tepat sebelum penabuh berubah ke keadaan tak aktif dan tetap pada keadaan tersebut sampai penabuh berkeadaan aktif kembali. Jadi pemalangan terjadi pada saat perubahan penabuh dari keadaan aktif, misalnya logika 1 (aras tegangan tinggi), ke keadaan tak aktif, misalnya logika 0 (aras tegangan rendah). Kerja register ini biasanya digambarkan dalam bentuk tabel keadaan seperti pada Gambar 8.1(a) dengan  $Q_0$  sebagai keadaan sebelumnya dan diagram waktu seperti pada Gambar 8.1(c). Titik nomor 1 pada Gambar 8.1(c) menunjukkan perubahan keluaran  $Q_G$  yang diakibatkan oleh perubahan masukan pada saat penabuh berlogika 1. Pada nomor 2 ditunjukkan bahwa keluaran  $Q_G$  terpalang pada logika 1 setelah terjadinya pemalangan pada perubahan penabuh dari 1 ke 0 sehingga keluaran  $Q_G$  tetap berlogika 1 walaupun masukan D berubah menjadi 0.



Gambar 8.1. Tabel Keadaan masing-masing flip-flop D dalam pemalang:  
(a) transparan, (b) terpicu (c) diagram waktu

Dalam pemalang terpicu, keadaan keluaran hanya berubah pada saat munculnya pulsa pemicu pada perubahan penabuh dari aras tinggi ke aras rendah. Keadaan keluaran dipalng pada keadaan masukan tepat pada saat munculnya pulsa pemicu, dan tetap pada keadaan tersebut sampai munculnya pulsa pemicu berikutnya. Dalam tabel keadaan Gambar 8.1(b) penabuh digambarkan sebagai panah turun untuk menunjukkan bahwa pemucuan terjadi pada saat terjadinya perubahan sinyal pemalang (latch) dari aras tinggi (H) ke aras rendah (L). Dalam diagram waktu Gambar 8.1(c) pemucuan ini ditunjukkan oleh nomor 3.

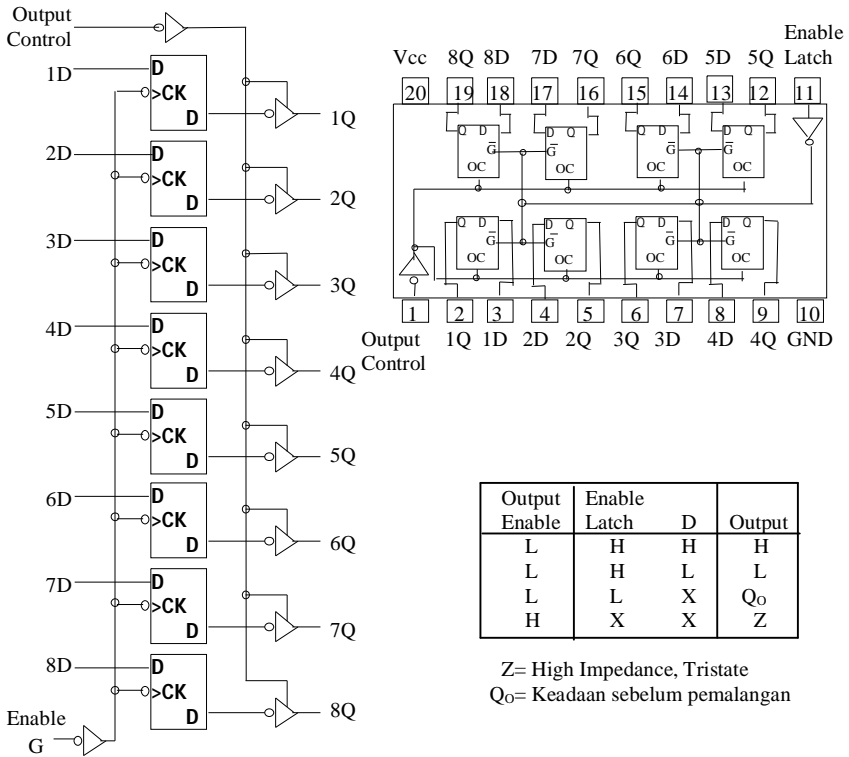
Untuk penggunaan dalam sistem komputer banyak dibutuhkan pengendalian keluaran supaya keluaran muncul hanya pada waktu yang diinginkan dan di luar waktu tersebut keluaran itu berada dalam keadaan “**bukan 0 dan juga bukan 1**” supaya tidak mempengaruhi dan tidak dipengaruhi oleh rangkaian di luarnya. Keadaan keluaran yang bukan 0 dan juga bukan 1 itu disebut sebagai tri-state yang dipandang sebagai akibat dari keluaran yang mempunyai impedansi tinggi (Hi-Z) sehingga tidak dilalui arus listrik. Keadaan tri-state ini memungkinkan beberapa sumber data dihubungkan dengan perkawatan secara langsung (hard-wired) dalam sistem bus. Register pemalang dalam bentuk rangkaian terpadu dibuat dalam kemasan MSI dua diantaranya yang banyak digunakan adalah 74LS373 (transparan) dan 74LS374 (terpicu). Pada Gambar 8.2 ditunjukkan pemalang 74LS373 yang merupakan register pemalang 8-bit yang disusun dari flip-flop D. Perhatikan bahwa rangkaian terpadu ini mempunyai 2 kendali: Output Control (Output Enable) dan Enable Latch.

Bila Output Enable dibuat aktif ( $OC=0$ ) maka keadaan flip-flop D akan muncul pada keluaran. Sebaliknya, bila  $OC=1$ , keadaan flip-flop D tidak muncul pada keluaran. Dalam keadaan ini, keluaran berada dalam keadaan tristate (berimpedansi tinggi) sehingga keluaran ini tidak mempengaruhi rangkaian luar yang dihubungkan kepadanya dan juga tidak dipengaruhi oleh rangkaian luar tersebut. Sifat ini sangat dibutuhkan dalam sistem bus. Seperti ditunjukkan dalam bagan rangkaiannya, sifat ini diberikan dengan memasang buffer/line driver pada semua keluaran flip-flopnya.

Enable Latch merupakan penabuh bersama bagi semua flip-flop D dengan fungsi sama dengan penabuh bagi pemalang transparan yang telah diuraikan sebelumnya, yaitu keluaran akan mengikuti keadaan masukan D selama sinyal Enable ini berlogika 0 dan akan tetap mempertahankan keadaannya saat pemalangan setelah sinyal ini berubah ke logika 1.

## 8.2 Memori

Dalam lingkungan komputer digital, secara garis besar dikenal dua macam memori: RAM (Random Access Memory) dan ROM (Read Only Memory). Dalam Bab 5 telah diterangkan bahwa ROM adalah rangkaian kombinasi tanpa adanya sel pengingat. RAM-lah yang merupakan memori dalam arti mempunyai



Gambar 8.2 Rangkaian Terpadu Pemalang tipe 74LS373.

sel pengingat. Nama RAM sebenarnya kurang tepat karena kalau ditinjau dari cara aksesnya, ROM juga dapat diakses secara acak (random). Tetapi karena nama RAM lebih luas dikenal di masyarakat, nama tersebut tetap dipertahankan. Nama yang tepat sebenarnya adalah memori baca-tulis (Read-Write Memory). Pada dasarnya, RAM adalah sekumpulan flip-flop. Jadi, sesuai dengan uraian di atas, RAM tidak lain dari pada register. Dasar dari RAM adalah flip-flop D yang telah diuraikan dalam Bab 6.

Seperti telah disebutkan di atas, memori selalu terdiri dari sejumlah sel (flip-flop) serupa. Masing-masing flip-flop dalam RAM dikenal berdasarkan nomor alamat (address)-nya. Flip-flop juga harus dapat ditulis dengan data baru dan data yang disimpannya harus dapat dibaca. Karena itu diperlukan sinyal pemilih alamat untuk menentukan sel yang akan ditulis atau dibaca, dan sinyal kendali untuk menulis (sinyal tulis) dan sinyal kendali untuk membaca (sinyal baca). Hanya sel

terpilih sajalah yang boleh diberikan sinyal baca atau tulis yang aktif. Jadi pulsa penabuh untuk masing-masing flip-flop dapat diperoleh dari peng-AND-an sinyal tulis dan pemilih alamat. Operasi baca tidak perlu dilakukan pada saat bersamaan dengan operasi tulis. Karena itu, data masukan juga perlu dihalang (disabled) masuk ke flip-flop pada saat operasi bukan operasi tulis. Ini dapat dilakukan dengan meng-AND-kan data masukan dengan sinyal penabuh flip-flop. Dalam operasi baca juga, hanya keluaran sel terpilih sajalah yang boleh ditampilkan pada keluaran RAM. Jadi keluaran masing-masing sel harus di-AND-kan dengan sinyal baca dan sinyal pemilih alamatnya.

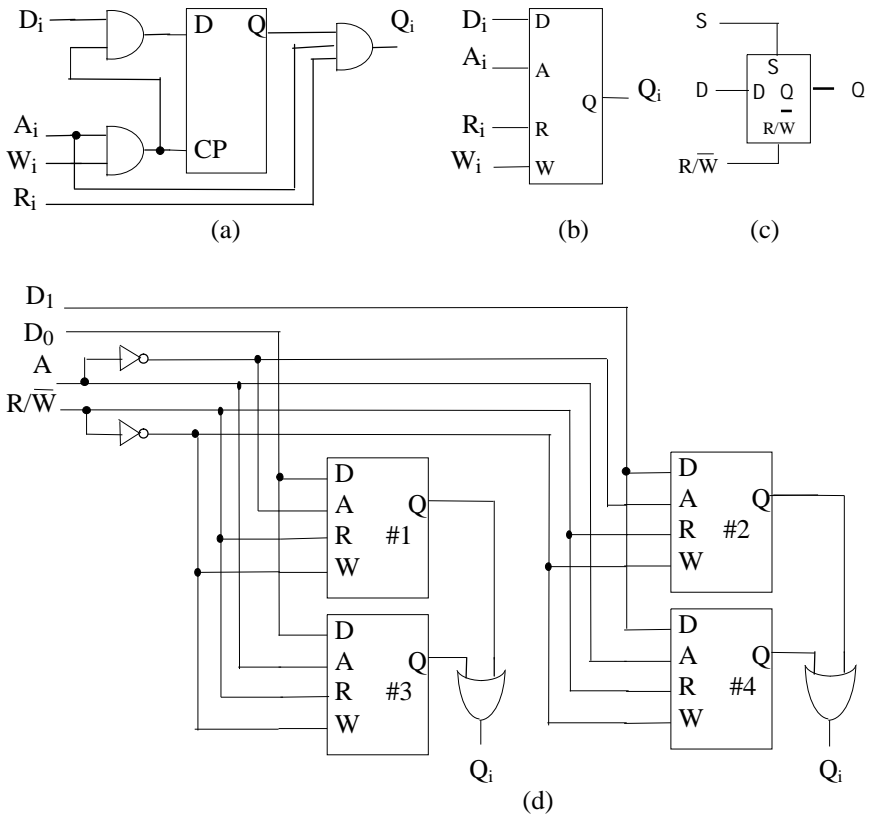
Bila data masukan dan keluaran untuk elemen ke  $i$  kita sebut  $D_i$  dan  $Q_i$ , sinyal alamatnya kita sebut  $A_i$ , sinyal baca  $R_i$  dan sinyal tulis  $W_i$ , maka untuk masing-masing flip-flop  $D$  dengan masukan  $D$  dan keluaran  $Q$ , dapat ditulis persamaan logika sebagai berikut:

$$\begin{aligned} \text{Penabuh:} & \quad CP = A_i W_i \\ \text{Data masukan:} & \quad D = D_i CP \\ \text{Data Keluaran:} & \quad Q_i = A_i R_i Q \end{aligned}$$

Rangkaian logika masing-masing sel dapat digambarkan seperti pada Gambar 8.2(a) dan dapat disimbolkan seperti pada Gambar 8.3(b). Dalam banyak pemakaian memori, kendali baca-tulis digabung menjadi satu sinyal bersama, biasanya diberi nama  $R/W$ , yang berlogika 1 untuk operasi baca dan berlogika 0 untuk operasi tulis. Dengan demikian, setial sel dalam RAM dapat digambarkan seperti ditunjukkan dalam Gambar 8.3(c). Sinyal  $S$  (Select) dalam simbol ini mewakili sinyal alamat.

Dalam kebanyakan sistem digital, operasi baca-tulis atas memori dilakukan serentak untuk sekumpulan bit (sel) yang dipandang sebagai satu kesatuan data yang disebut kata ("word"). Mengakses satu lokasi berarti membaca atau menulis satu kata. Ukuran kata yang paling banyak digunakan adalah 4 dan 8 bit. Untuk contoh, pada Gambar 8.3(d) digambarkan rangkaian logika suatu RAM  $2 \times 2$  bit, yaitu RAM 2 kata dengan panjang kata 2 bit.

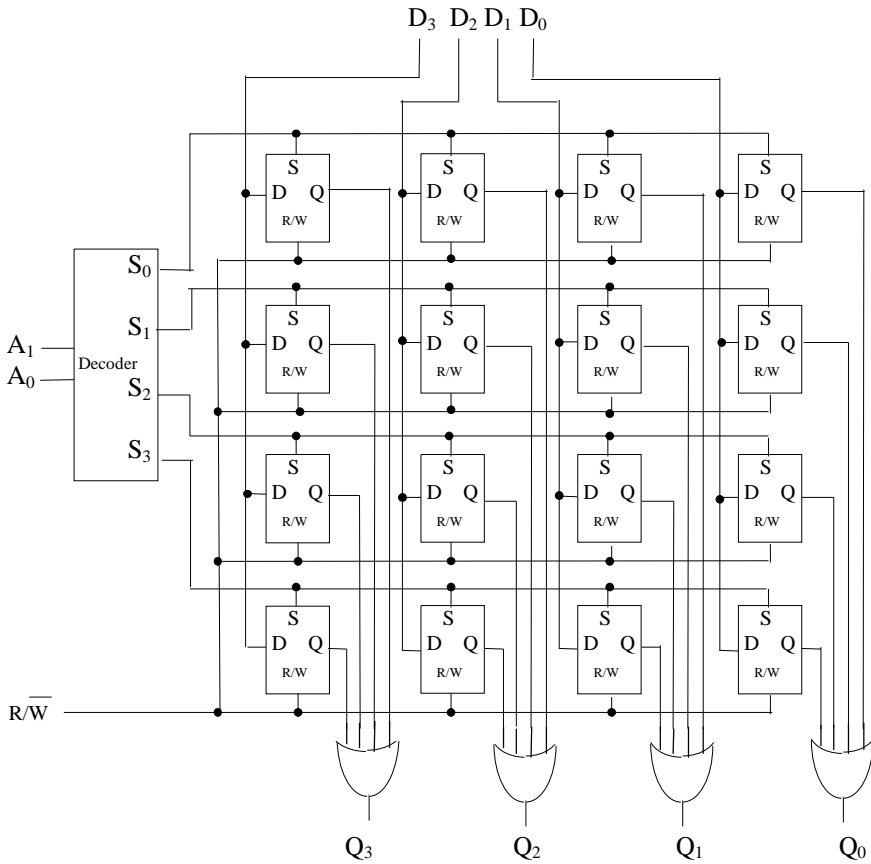
Dalam Gambar 8.3(d), data keluaran  $Q_1 Q_0$  diambil dari sel #1 atau sel #3 sebagai bit 1 ( $= Q_1$ ) dan dari sel #2 atau sel #4 sebagai bit 0 ( $= Q_0$ ). Perhatikan bahwa dalam diagram tersebut, sel digambarkan dengan simbol pada bagian (b) dan keluaran hanya di-OR-kan tanpa menunjukkan hubungan antara sinyal kendali baca dan alamat dengan keluaran. Sel #1 dan #2 mempunyai nomor alamat (lokasi) yang sama, yaitu lokasi 0 dan karena itu membentuk kata 0. Sel #3 dan #4 mempunyai nomor lokasi yang sama, yaitu lokasi 1 dan membentuk kata 1. Pengaksesan dilakukan atas lokasi 0 bila logika  $A = 0$  dan atas lokasi 1 bila logika  $A = 1$ . Dikatakan kata 0 disimpan di lokasi atau alamat 0 dan kata 1 disimpan di lokasi atau alamat 1. Untuk memilih salah satu kata dari suatu memori dengan  $2^n$  kata, dibutuhkan  $n$  bit alamat:  $A_0, A_1, A_2, \dots, A_{n-1}$ . Untuk pemilihan ini digunakan dekoder yang memilih sel/kata yang akan diakses.



Gambar 8.3. Rangkaian memori baca-tulis (RAM)

Dalam rangkaian terpadu RAM, pemilihan kata memori yang di akses dilakukan dengan menggunakan dekoder yang sudah dipadukan dalam serpih yang sama dengan sel memorinya. Dalam Gambar 8.4 ditunjukkan rangkaian dalam suatu rangkaian terpadu RAM 4 kata (word) dengan panjang kata (word length) 4 bit. Dari gambar ini dapat dilihat bahwa:

- setiap data masukan diumpungkan kepada 4 sel (flip-flop)
- setiap keluaran dekoder alamat  $S_0, S_1, S_2, S_3$  memilih 4 sel yang diperlakukan sebagai satu kata 4 bit (dipandang sebagai satu alamat).
- Seluruh sel dikendalikan oleh satu sinyal baca/tulis bersama.
- Setiap bit data keluaran diperoleh dari gerbang OR yang dicatu dari 4 sel.



Gambar 8.4. Rangkaian RAM 4x4.

Walaupun data masukan diumpunkan kepada 4 sel, hanya sel yang dipilih (diaktifkan) oleh dekoder alamat yang dipengaruhi oleh operasi baca/tulis. Dalam RAM dengan kapasitas yang besar, pemilihan alamat tidak dari hanya satu sisi seperti dalam gambar di atas, tetapi dilakukan dari dua sisi, yaitu sisi baris dan sisi kolom. Sebagai contoh, RAM tipe 2112 yang berkapasitas 1024 bit yang diorganisasikan sebagai 256 x 4 bit. Ini berarti bahwa setiap lokasi mengandung 4 bit sebagai satu kata. Serpih RAM ini mempunyai 8 bit alamat, 5 bit pemilih baris (untuk 32 baris) dan 3 bit pemilih kolom (untuk 8 kolom). Contoh lain, RAM tipe 2114 yang berkapasitas 4096 bit (1024 x 4 bit) mempunyai 10 bit alamat yang terdiri atas 6 pemilih baris dan 4 pemilih kolom.

Perhatikan bahwa data keluaran rangkaian di atas akan tersedia setiap saat. Untuk memungkinkan pemanfaatan RAM dalam sistem bus, maka dalam banyak hal, pada bagian keluaran RAM sudah ditambahkan buffer (line driver) tristate seperti yang dilakukan pada rangkaian keluaran register pemalang Gambar 8.2 di bagian depan. Dalam banyak serpih RAM yang digunakan sekarang ini, terutama yang digunakan pada mikrokomputer, data masuk dan keluar disalurkan pada saluran dua arah dengan tujuan mengurangi cacah pen Kemasan. Dalam hal ini, saluran (bus) data dilengkapi bus driver dua arah dengan arah yang ditentukan oleh sinyal baca/tulis (R/W).

### 8.3 Register Geser Masukan Seri

Register geser (Shift Register), disamping dapat menyimpan data biner, juga dapat melakukan proses penggeseran data. Penggeseran data diperlukan baik dalam pengiriman data secara berderet (serial) maupun dalam perhitungan aljabar perkalian dan pembagian.

Dalam komunikasi data (umumnya secara seri), data beberapa bit (umumnya 8 bit) dikirim melalui saluran komunikasi bit demi bit. Data yang disimpan/diolah dalam komputer selalu bersifat paralel. Agar dapat dikirim melalui satu saluran komunikasi, maka data beberapa bit itu harus digeser keluar satu demi satu. Begitu juga di sisi penerima, bit demi bit data yang diterima dari saluran komunikasi harus digeser sampai membentuk satu satuan data paralel agar dapat disimpan/diolah dalam register komputer. Jadi dalam komunikasi data, register geser memegang peranan yang sangat penting.

Dalam perhitungan aritmatika (aljabar), komputer selalu melaksanakan operasi perkalian dan pembagian dengan melakukan penambahan/pengurangan disertai penggeseran ke kiri/kanan secara berulang-ulang. Perlu dicatat bahwa bila suatu data bilangan yang digeser ke kiri, maka harga bilangan itu akan digandakan menjadi dua kali harga semula dan bila suatu bilangan biner digeser ke kanan, maka harganya menjadi setengah dari harga sebelum digeser. Sebagai contoh, kalau biner 0110, yang setaranya dalam desimal adalah 6, digeser ke kanan satu kedudukan maka harganya menjadi 0011 ( $= 3_{10}$ ), sedangkan bila digeser ke kiri, harganya menjadi 1100 ( $12_{10}$ ). Perhatikan bahwa penggeseran yang normal dilakukan dengan menambahkan bit 0 pada posisi paling kanan pada penggeseran ke kiri dan pada kedudukan paling kiri pada penggeseran ke kanan. Karena cacah bit yang dapat ditampung oleh suatu register sudah tertentu, maka bit di ujung lain akan hilang, yaitu bit paling kanan pada penggeseran ke kanan dan bit paling kiri pada penggeseran ke kiri.

Pengertian penggeseran akan lebih mudah dimengerti dengan memperhatikan contoh-contoh berikut ini. Andaikan register kita dapat menampung 8 bit data, artinya terdiri atas 8 flip-flop, dan andaikan pula bahwa sebelum melakukan penggeseran, register itu berisi 1001 1010, maka untuk operasi geser kanan dan kiri,



data yang disimpan register tersebut berubah sebagai berikut:

Geser kanan: Sebelum penggeseran: 1 0 0 1 1 0 1 0  
 Geser 1 x : 0 1 0 0 1 1 0 1  
 Geser 2 x : 0 0 1 0 0 1 1 0

Geser kiri: Sebelum penggeseran: 1 0 0 1 1 0 1 0  
 Geser 1 x : 0 0 1 1 0 1 0 0  
 Geser 2 x : 0 1 1 0 1 0 0 0

Masukan 0 di sisi awal pergeseran yang diuraikan di atas dapat diberikan sebagai data masukan dari luar. Jelas dari sini bahwa untuk merealisasikan rangkaian penggeser sebagai register geser, menuntut keluaran satu flip-flop diberikan kepada masukan flip-flop berikutnya dalam urutan penggeseran. Jadi, dengan menggunakan flip-flop JK, persamaan masukan masing-masing flip-flop dapat ditulis sebagai berikut:

$$\begin{array}{llll} J_A = D_{in} & K_A = \bar{J}_A & J_C = Q_B & K_C = \bar{J}_C \\ J_B = Q_A & K_B = \bar{J}_B & J_D = Q_C & K_D = \bar{J}_D \end{array}$$

dengan  $D_{in}$  sebagai data masukan luar.

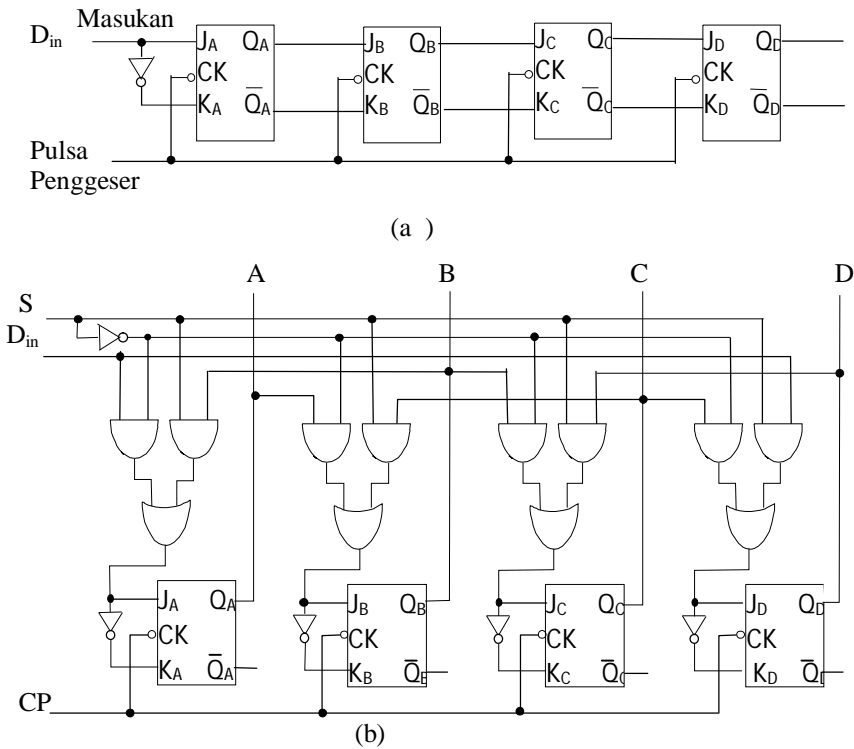
Pada Gambar 8.5(a) ditunjukkan rangkaian logika dari suatu register geser kanan 4 bit. Perhatikan bahwa keluaran flip-flop di kiri menjadi masukan bagi yang di kanannya. Untuk memuat register ini dengan sesuatu data (menyetel keadaan seluruh flip-flop), data harus dimasukkan bit per bit dari masukan yang di ujung kiri. Isi register pada setiap saat dapat dibaca dari keluaran masing-masing flip-flop. Karena itu, register geser ini disebut juga sebagai register geser masukan seri keluar-paralel (Serial-in Parallel-out Shift Register).

Untuk membuat register ini menjadi register geser kiri, maka keluaran dari setiap flip-flop (kecuali yang paling kanan) dihubungkan kepada masukan flip-flop di kirinya dan masukan luar diberikan ke masukan flip-flop ujung kanan.

Suatu register geser yang dapat menggeser baik ke kiri maupun ke kanan dapat disusun dengan menambahkan gerbang-gerbang pemilih masukan untuk setiap flip-flop untuk modus geser kanan atau kiri beserta gerbang pemilih modus gesernya. Masukan luar harus dapat diberikan ke masukan flip-flop ujung kiri maupun ujung kanan. Dengan membuat sinyal pemilih  $S=0$  untuk geser kanan dan  $S=1$  untuk geser kiri dan  $D_{in}$  sebagai masukan dari luar, maka untuk Gambar 8.5 (a) dapat diperoleh persamaan masukan sebagai berikut:

$$\begin{array}{llll} J_A = \bar{S} D_{in} + S B & K_A = \bar{J}_A & J_C = \bar{S} B + S D & K_C = \bar{J}_C \\ J_B = \bar{S} A + S C & K_B = \bar{J}_B & J_D = \bar{S} C + S D_{in} & K_D = \bar{J}_D \end{array}$$

Rangkaian logika register geser kiri-kanan ini dapat digambarkan seperti ditunjukkan pada Gambar 8.5(b).



Gambar 8.5. Rangkaian Register Geser.  
 (a) Geser Kanan (b) Geser Kanan/Kiri

Perhatikan bahwa untuk geser kanan ( $S=0$ ),  $J_A$  yang menerima masukan  $D_{in}$  dan untuk geser kiri ( $S=1$ ),  $J_D$  yang menerima  $D_{in}$ .

Di samping register geser kiri dan kanan, dapat juga dibuat register geser rotasi atau *cyclic* atau geser daur dimana bit di satu ujung digeser ke ujung lainnya. Contoh operasi geser daur ini dapat ditunjukkan sebagai berikut:

|                      |                 |
|----------------------|-----------------|
| Sebelum pergeseran : | 1 0 0 1 1 0 1 0 |
| Rotasi Kanan 1 x :   | 0 1 0 0 1 1 0 1 |
| 2 x :                | 1 0 1 0 0 1 1 0 |
| ...                  | .....           |
| 8 x :                | 1 0 0 1 1 0 1 0 |

Sebelum pergeseran : 1 0 0 1 1 0 1 0  
 Rotasi Kiri 1 x : 0 1 0 0 1 1 0 1  
               2 x : 1 0 0 1 1 0 1 0  
               ... : .....  
               8 x : 1 0 0 1 1 0 1 0

Register geser rotasi ini dapat diperoleh dengan menghubungkan keluaran flip-flop D ke masukan pada Gambar 8.5(a) untuk rotasi kanan. Untuk rotasi kiri, urutan hubungan masukan-keluaran flip-flop satu dengan yang lain harus dipertukarkan. Perhatikan bahwa pergeseran daur/rotasi ke kanan yang diikuti pergeseran daur ke kiri sejumlah yang sama akan mengembalikan register kepada keadaannya semula.

#### 8.4 Register Geser masukan paralel

Register geser (Shift register) yang diuraikan di atas menerima masukan secara seri. Dalam komunikasi data antara prosesor dalam komputer dengan peralatan masukan/ keluaran (I/O devices) disekelilingnya atau antar komputer, sering dibutuhkan register geser yang dapat dimuati data 1 kata (umumnya 8 bit) secara lengkap. Untuk itu dibutuhkan kemampuan menyetel keadaan masing-masing flip-flop menurut bit data yang sesuai. Kemampuan ini biasanya disebut sebagai kemampuan PRESET. Untuk membuat register geser demikian, maka kepada rangkaian register geser yang diuraikan di depan [Gambar 8.5(b)] perlu ditambahkan kendali pemilih masukan bagi masing-masing flip-flop apakah dari data luar atau dari keluaran flip-flop tetangganya. Jadi, untuk masing-masing flip-flop ada 3 masukan yang dapat dipilih: keluaran flip-flop di kiri (geser kanan), keluaran flip-flop di kanan (geser kiri) dan masukan paralel dari luar. Karena itu dibutuhkan paling tidak 2 bit kendali pemilihan. Untuk 2 bit pemilih dapat dipilih 4 macam operasi, sehingga masih tersedia 1 macam operasi tambahan yang dapat dipilih. Modus operasi tambahan ini dapat digunakan untuk operasi diam (tanpa operasi) dengan menghalangi (inhibit) penabuh. Modus operasi tersebut dikodekan seperti dalam tabel berikut:

| Pemilih |    | Operasi     |
|---------|----|-------------|
| S1      | S0 |             |
| 0       | 0  | geser kanan |
| 0       | 1  | geser kiri  |
| 1       | 0  | muat        |
| 1       | 1  | diam        |

Dengan pemilihan modus seperti pada tabel di atas, maka persamaan masukan masing-masing flip-flop untuk register geser yang menggunakan flip-flop RS

dapat diperoleh sebagai berikut:

$$S_A = \overline{S_1} \overline{S_0} D_{in} + \overline{S_1} S_0 Q_B + S_1 \overline{S_0} A \quad R_A = \overline{S_A}$$

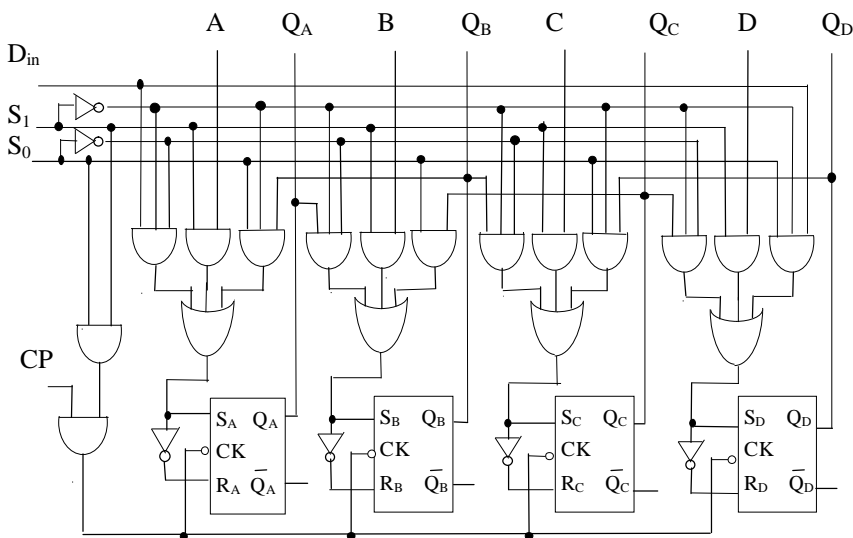
$$S_B = \overline{S_1} \overline{S_0} Q_A + \overline{S_1} S_0 Q_C + S_1 \overline{S_0} B \quad R_B = \overline{S_B}$$

$$S_C = \overline{S_1} \overline{S_0} Q_B + \overline{S_1} S_0 Q_D + S_1 \overline{S_0} C \quad R_C = \overline{S_C}$$

$$S_D = \overline{S_1} \overline{S_0} Q_C + \overline{S_1} S_0 D_{in} + S_1 \overline{S_0} D \quad R_D = \overline{S_D}$$

$$\text{Penabuh} = S_1 S_0 CP$$

dengan A, B, C, dan D adalah data masukan paralel dan  $Q_A$ ,  $Q_B$ ,  $Q_C$ , dan  $Q_D$  adalah keluaran paralel (keadaan flip-flop). Rangkaian register geser ini ditunjukkan pada Gambar 8.6.



Gambar 8.6. Register Geser dengan masukan paralel

Dapat dilihat bahwa operasi geser dilakukan bila kendali pemilih  $S_1=0$  dan operasi muat (load) dilakukan bila kendali pemilih  $S_1=1$  dan  $S_0=0$  ( $S_1 S_0=10$ ). Bila kendali pemilih  $S_1 S_0=11$  maka register tidak melakukan apa-apa sebab penabuhnya dihalangi (inhibit).

Bila  $S_1 S_0=00$  maka flip-flop A mendapat masukan dari masukan seri  $D_{in}$  sedangkan bila  $S_1 S_0=01$  maka flip-flop D mendapat masukan dari masukan seri  $D_{in}$

Register Geser dalam bentuk rangkaian terpadu skala menengah (MSI) seperti diuraikan di atas dapat diperoleh di pasaran, misalnya yang disebut "4-bit Bidirectional Universal Shift Register" tipe 74194 sudah dilengkapi dengan semua kemudahan: geser kanan, geser kiri, bersihkan (clear), muati (load), dan diam (tidak berubah).

### 8.5 Soal Latihan

1. Gambarkanlah suatu kumpulan register (register file) yang terdiri dari 8 register 4 byte, yang beri nomor 0 sampai dengan 7. Gunakan rangkaian dekoder untuk memilih nomor alamat (lokasi)nya.
2. Melalui satu bus data 4 bit dikirimkan data 8 bit dalam dua kali pengiriman. Data ini akan disimpan di memori 8 bit dengan penulisan 8 bit sekaligus. Gambarkanlah diagram blok hubungan pemalang, yang digunakan untuk memalang data kriman pertama, dengan memorinya.
3. Rencanakanlah suatu penggeser 4 bit yang dapat melakukan penggeseran baik ke kiri maupun ke kanan. Setiap dilakukan penggeseran ke kanan, keadaan flip-flop yang paling kiri dibuat berharga 0 dan pada penggeseran ke kiri, keadaan flip-flop yang paling kanan dibuat berharga 0. Pergunakanlah :
  - a. Flip-flop RS
  - b. Flip-flop D
4. Rencanakanlah register 4 bit dengan menggunakan flip-flop JK yang setiap diberikan pulsa penabuh kepadanya harga yang dikandung oleh ke-4 flip-floponya akan ditambah 5 dengan catatan bahwa harga yang dapat disimpangnya tidak lebih dari 9, bila lebih dari 9 keadaannya kembali menjadi 0. Jadi, kalau saat sebelum dikenakan pulsa keadaannya berharga  $N$ ,  $0 (0000) \leq N \leq 9 (1001)$ , setelah penabuhan keadaannya menjadi  $N + 5$ .
5. Rencanakanlah register geser "cyclic" 4 bit dengan masukan paralel yang menggeser isi flip-floponya sebanyak 2 posisi setiap muncul penabuh.