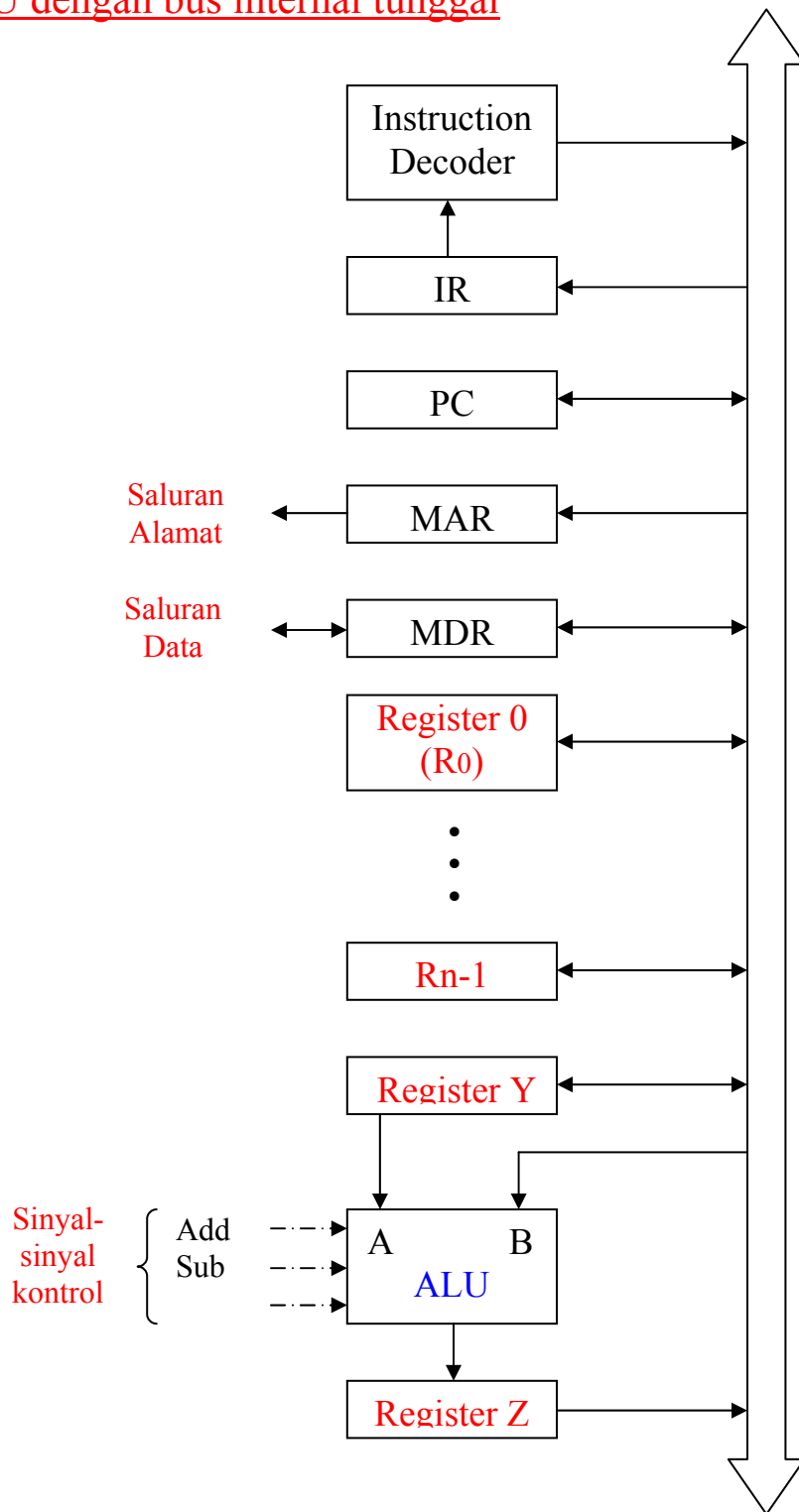


STRUKTUR INTERNAL CPU

CPU dengan bus internal tunggal



- Gambar di atas adalah struktur internal **CPU dengan bus internal tunggal**. Pada gambar di atas terdapat n buah register (R0 sampai R_{n-1}) yang dapat digunakan untuk menyimpan data. Karena hanya ada **satu bus**, bus akan **digunakan secara bergantian** dan **tidak diperbolehkan dua komponen mengoutputkan data ke bus secara bersamaan**.

- Berikut ini akan ditunjukkan pemindahan data antar komponen pada saat CPU membaca sebuah data dari memori.
- Misalkan alamat *data tersebut disimpan di register R1, dan nantinya data tersebut harus disimpan di register R2*.
- Urutan operasinya adalah sebagai berikut :
 1. $MAR \leftarrow [R1]$
 2. Read
 3. Tunggu sinyal MFC (Memory Function Completed)
 4. $R2 \leftarrow [MDR]$
 5. END

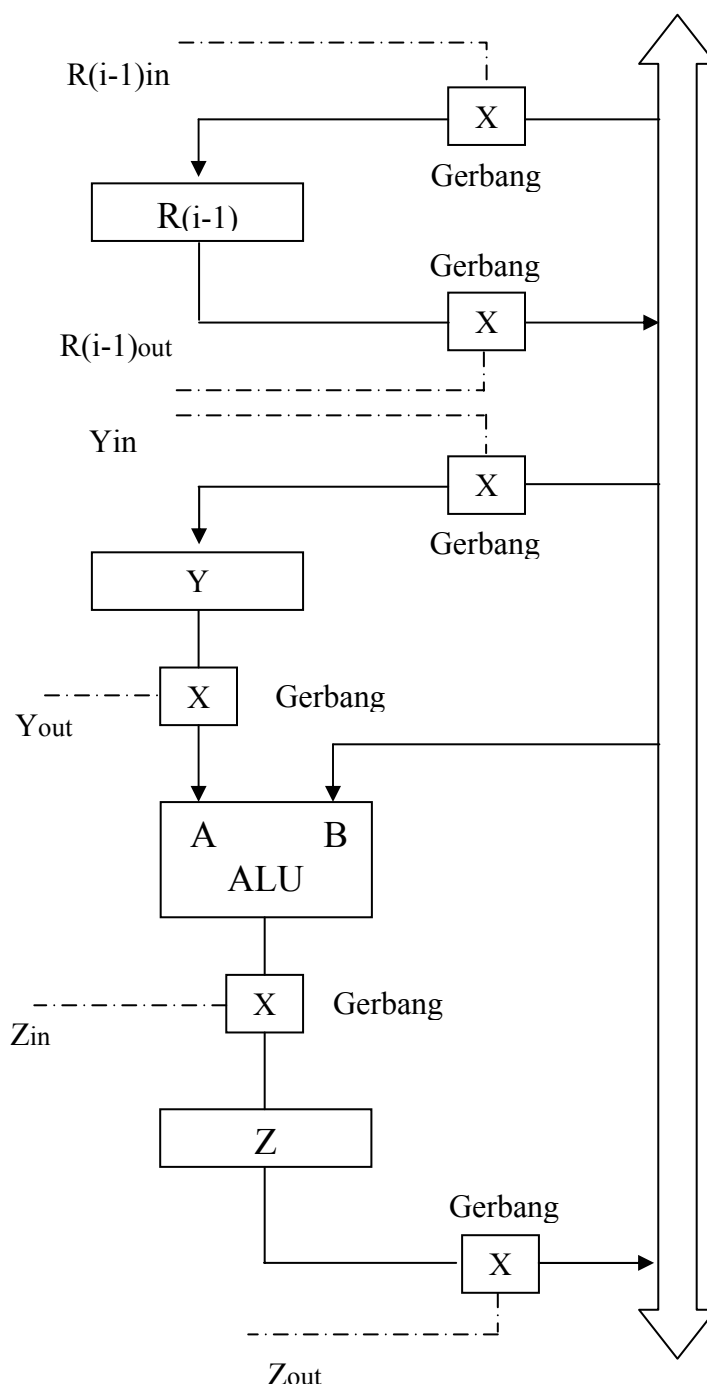
Pada operasi di atas alamat data terlebih dahulu diisikan ke register MAR. Setelah itu CPU akan mengirimkan sinyal kontrol (perintah) READ ke memori untuk membaca data. **Memori membutuhkan sejumlah waktu untuk memproses pembacaan data** tersebut dan nantinya akan mengirimkan data tersebut ke register MDR. Untuk itu, **CPU harus menunggu sampai memori mengirimkan sinyal MFC** (sebagai tanda bahwa proses pembacaan telah selesai dikerjakan dan data telah ada di MDR) sebelum memindahkan data dari MDR ke register R2.

Proses penulisan data ke memori

- Misalkan saat ini data tersimpan di register R2, dan data tersebut akan dituliskan ke suatu lokasi memori yang alamatnya tersimpan pada register R3.
- Urutan operasinya adalah sebagai berikut :
 1. $MAR \leftarrow [R3]$
 2. $MDR \leftarrow [R2]$
 3. WRITE
 4. Tunggu sinyal MFC
 5. END

Transfer data antar komponen

- ❑ Pada saat melakukan suatu operasi tertentu, CPU harus memindahkan data dari satu komponen ke komponen yang lain.
- ❑ Tidak diperbolehkan dua buah komponen mengoutputkan data ke bus secara bersamaan.
- ❑ Untuk itu CPU harus mengatur komponen mana yang mengoutputkan data pada suatu saat tertentu.
- ❑ Untuk melakukan fungsi tersebut CPU menggunakan sinyal-sinyal kontrol yang akan mengaktifkan gerbang-gerbang tertentu.
- ❑ Untuk lebih jelasnya perhatikan gambar berikut. Pada gambar digambarkan sejumlah gerbang. Ada gerbang yang berfungsi sebagai gerbang input dan ada gerbang yang berfungsi sebagai gerbang output.
- ❑ Gerbang-gerbang tersebut akan terbuka jika padanya diberikan sinyal kontrol.



- Untuk menginputkan data ke Register $R(i-1)$, pada gerbang input diberikan sinyal kontrol $R(i-1)_{in}$
- Untuk mengoutputkan data dari Register $R(i-1)$, pada gerbang output diberikan sinyal kontrol $R(i-1)_{out}$
- Untuk memindahkan data dari register $R(i-1)$ ke Register Y, pada gerbang output register $R(i-1)$ diberikan sinyal kontrol $R(i-1)_{out}$ dan kemudian pada gerbang input register Y diberi sinyal Y_{in}

Contoh :

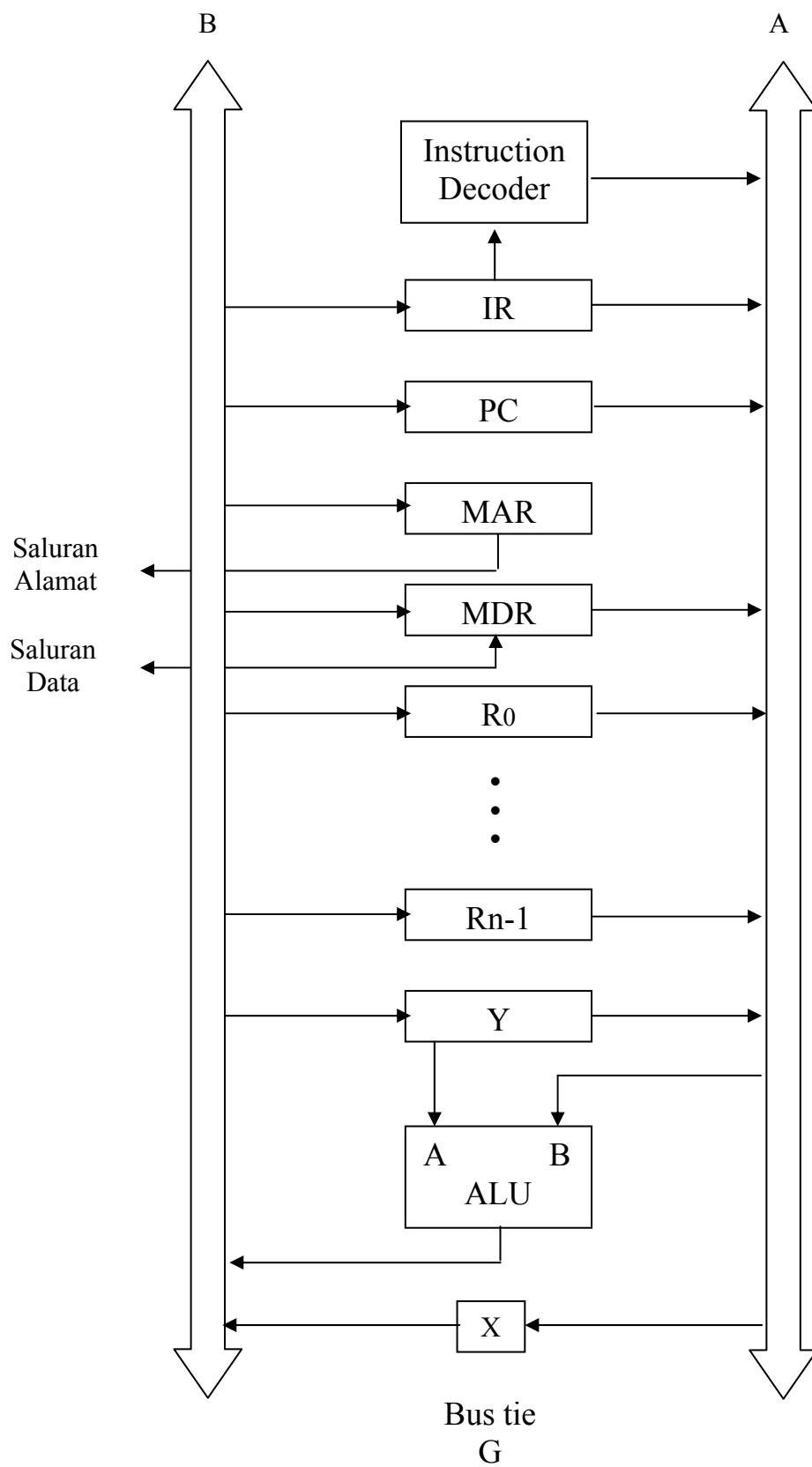
- Tentukan sinyal kontrol yang diperlukan untuk melakukan operasi penjumlahan 2 buah operan.
- Misalkan operan-operan tersimpan di register R1 dan R2
- Register Y digunakan sebagai penyimpan sementara salah satu operan pada saat operan kedua dipindahkan ke ALU.
- Register Z digunakan untuk menyimpan sementara hasil operasi.
- Selanjutnya hasil operasi tersebut akan disimpan di register R3

Urutan sinyal kontrol yang dihasilkan untuk melakukan operasi tersebut adalah :

1. $R1_{out}, Y_{in}$
2. $R2_{out}, Add, Z_{in}$
3. $Z_{out}, R3_{in}$

- Sinyal kontrol Add diberikan pada ALU agar ALU melakukan operasi penjumlahan pada kedua operan.

Struktur 2 bus



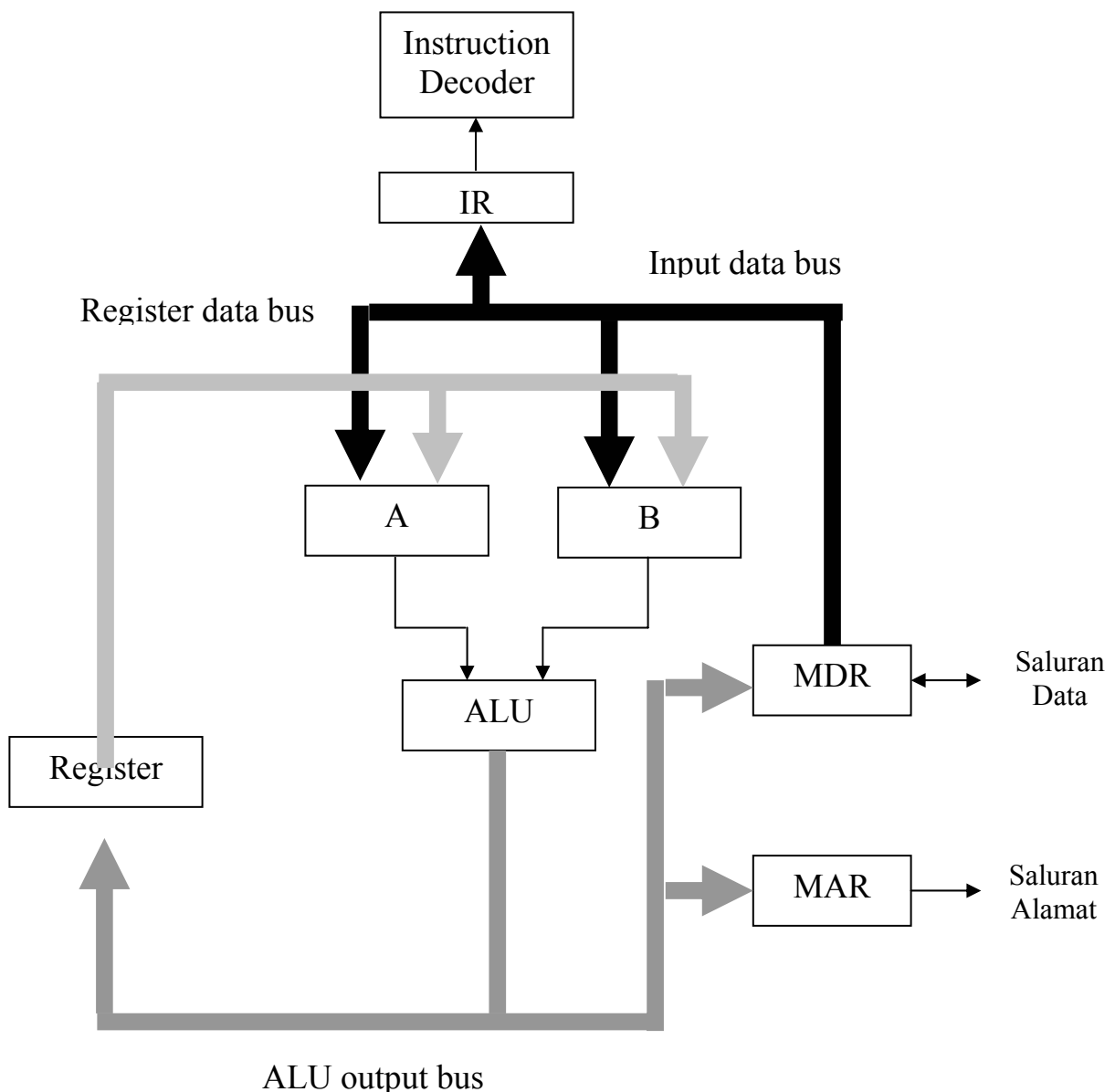
- Pada struktur di atas terdapat 2 buah bus yang dihubungkan oleh sebuah gerbang (bus tie G).
- Agar kedua bus terhubung, pada bus tie harus diberikan sinyal G_{enable} . Agar kedua bus terpisah pada bus tie diberikan sinyal kontrol $G_{disable}$

Contoh :

- Untuk melakukan operasi penjumlahan seperti pada contoh sebelumnya $R3 \leftarrow [R1] + [R2]$ dihasilkan sinyal kontrol berikut :

1. $R1_{out}$, G_{enable} , Y_{in}
2. $R2_{out}$, Add , ALU_{out} , $R3_{in}$

Struktur 3 bus



- Pada struktur di atas terdapat 3 buah bus, yaitu : Input data bus, register data bus, ALU output bus.
- Karena jumlah bus lebih banyak pemindahan data antar blok dapat dilakukan secara bersamaan.