Pertemuan ke-8

Programming Memory & Numbering System

Objektif

- Mengidentifikasikan dan mendefinisikan fungsi dari pemetaan memori di PLC
- Menggambarkan input dan output dalam image table files
- Mendefinisikan dan mengidentifikasi fungsi instruksi relay

Pendahuluan

- Fungsi ladder logic memungkinkan controller melakukan kalkulasi, membuat keputusan dan melakukan perintah yang kompleks.
- Fungsi ladder logic disimpan pada memory PLC.
- Memory PLC digunakan untuk menyimpan tipe-tipe program dan data.

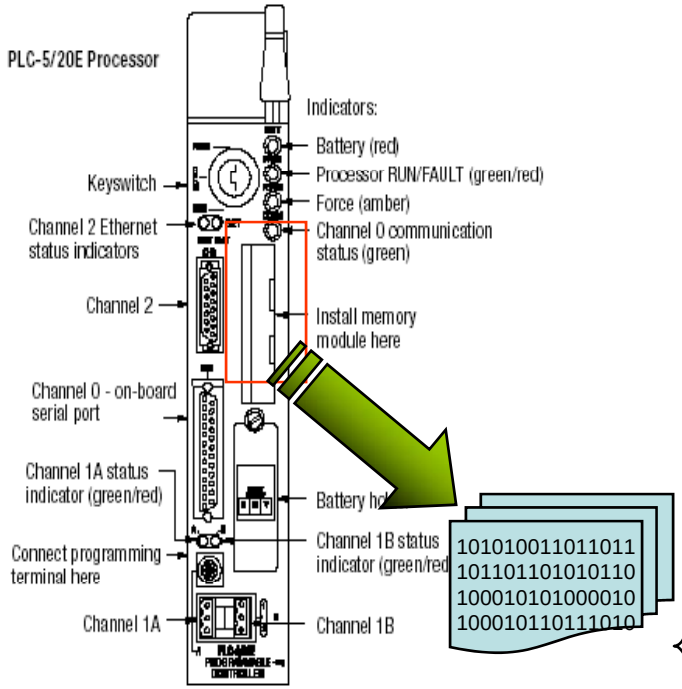
Faktor-faktor Penentu Memory PLC

- Jumlah I/O yang digunakan
- Ukuran program
- Kebutuhan pengumpulan data
- Kebutuhan fungsi supervisory
- Keperluan ekspansi masa mendatang

Memory Address

- Terdapat dua tipe dasar memory yang digunakan pada PLC Allen-Bradley :
  - Program memory
  - Data memory.
- Memory disusun dalam blok-blok hingga 256 element pada suatu susunan yang disebut file.
- Terdapat delapan Data files yang telah didefinisikan secara default, tapi data files tambahan dapat didefinisikan jika diperlukan.

Memory Space



Data Table :

I/O Locations

Timer/Counter Locations

User Program :

Ladder Logic, SFC, IL, ST, FBD

Housekeeping Memory :

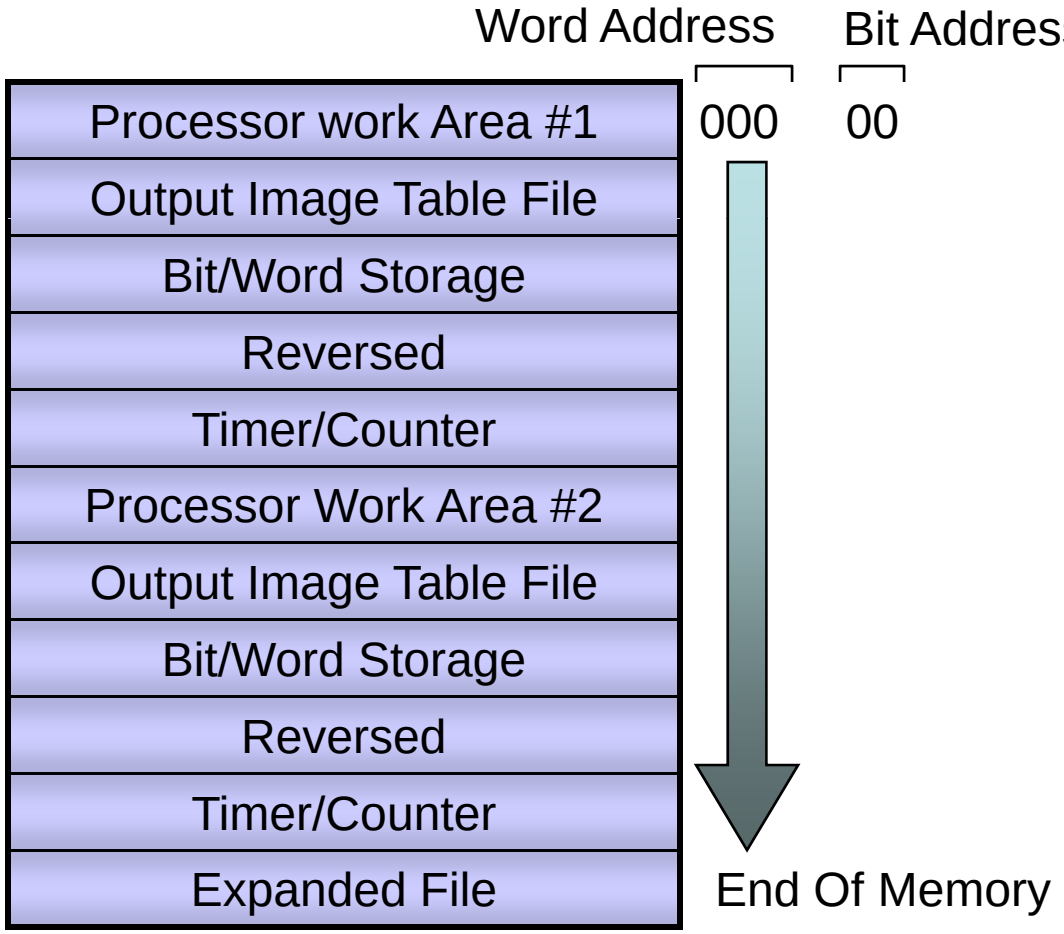
Functions

Processor Files

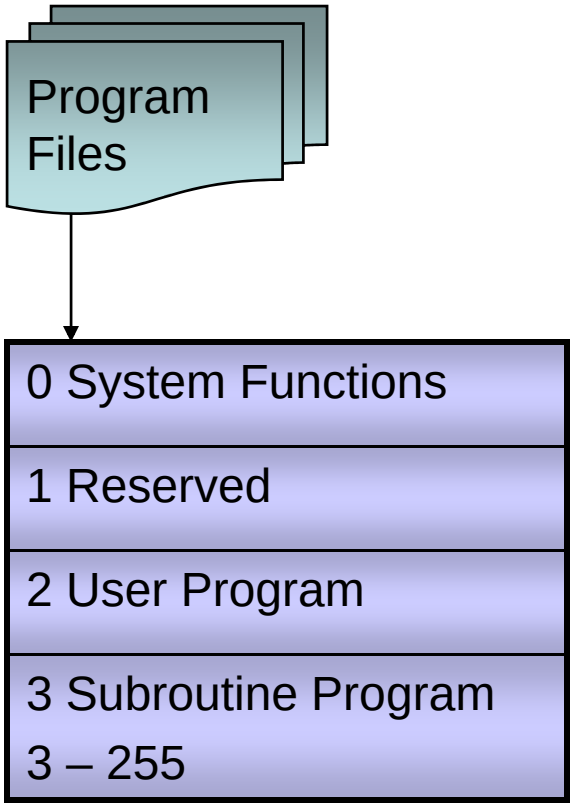
- System Program (file 0) - file ini selalu ada dalam setiap sistem processor, mengandung informasi yang berkaitan dengan tipe processor, konfigurasi I/O, file name processor, dan password
- Reserved (file 1) – file temporari space untuk processor
- Main Ladder Program (file 2) - program yang dibuat User untuk mendefinisikan bagaimana processor bekerja dengan baik
- Subroutine Ladder Program (file 3) – subroutine yang hanya bisa diaktivasi oleh instruksi subroutine dalam ladder

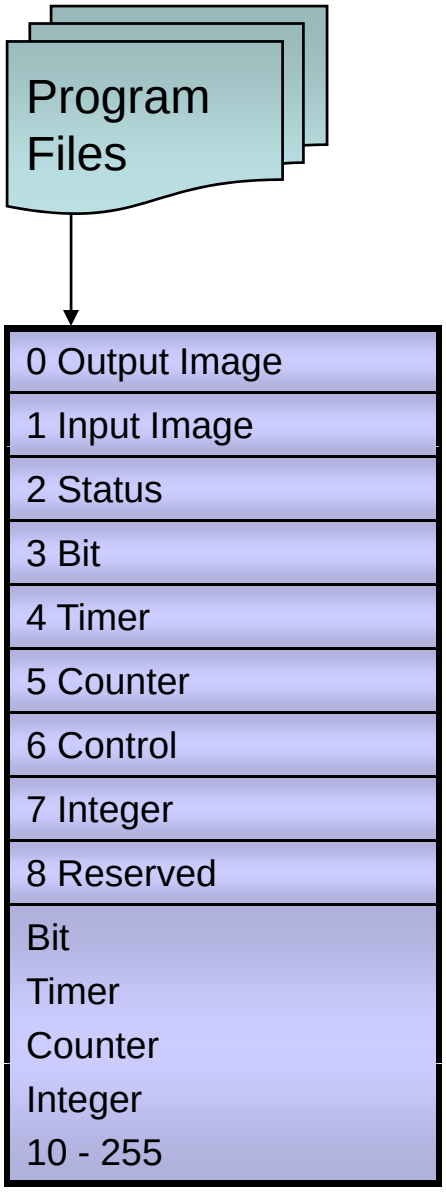
Processor Files

- Processor Files adalah kumpulan dari kedua file data dan file program. Terdiri dari semua instruksi, data dan informasi konfigurasi

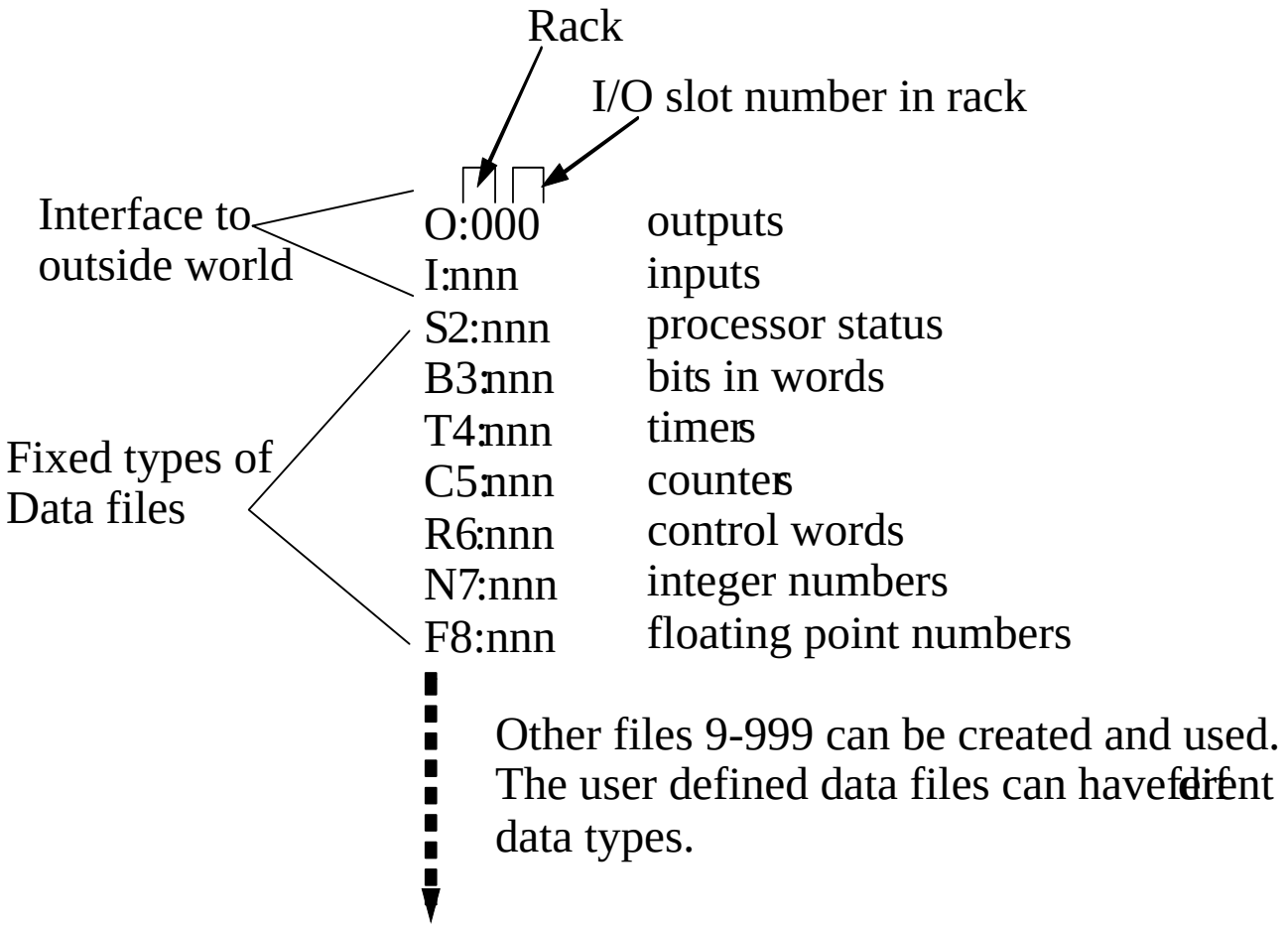


Data Files


Program Files
0 System Functions
1 Reserved
2 User Program
3 Subroutine Program
3 – 255


Program Files
0 Output Image
1 Input Image
2 Status
3 Bit
4 Timer
5 Counter
6 Control
7 Integer
8 Reserved
Bit
Timer
Counter
Integer
10 - 255

Data Files AB PLC 5

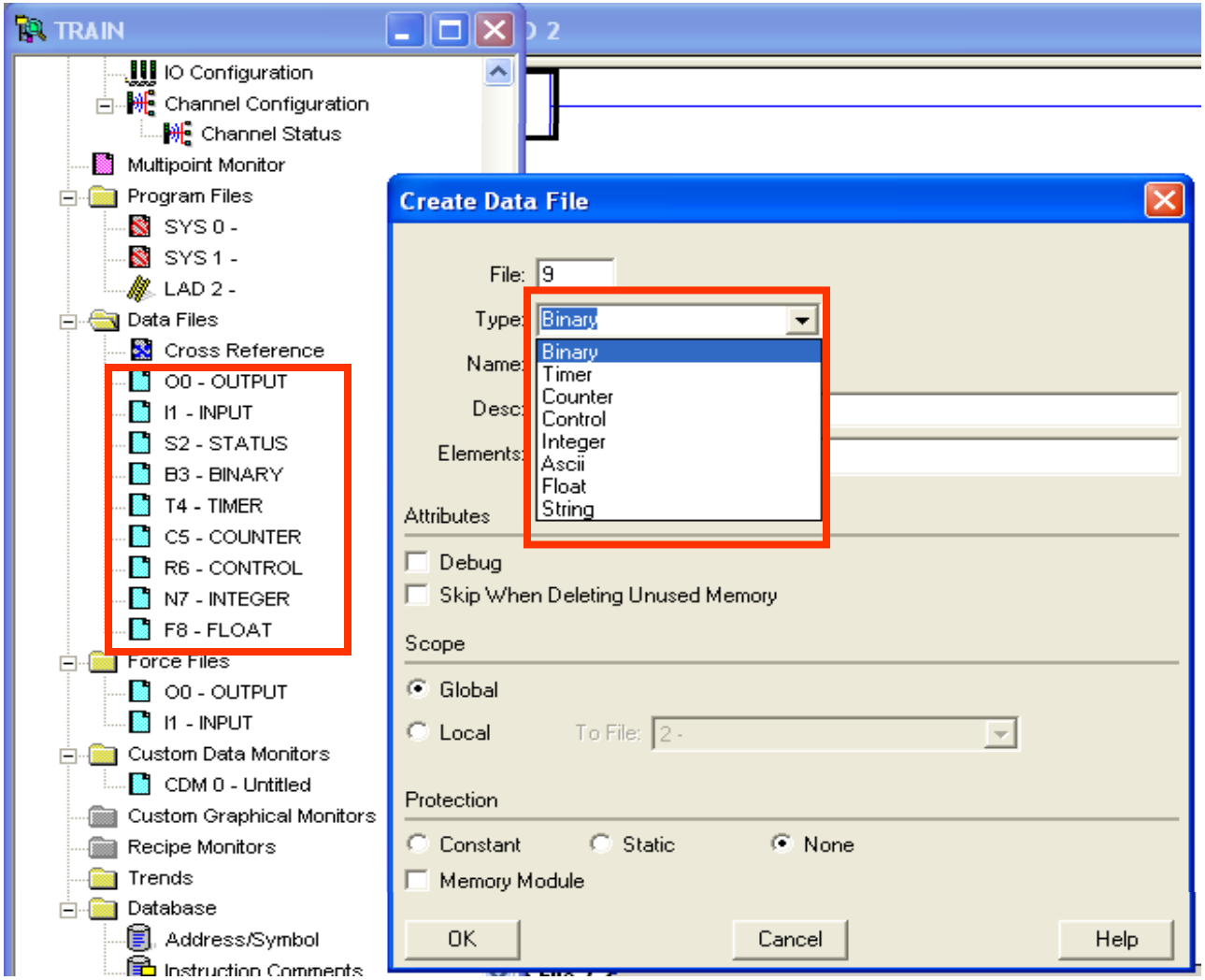


Tipe Data AB PLC 5

Tipe	Ukuran (word)
A - ASCII	1/2
B - bit	1/16
BT - block transfer	6
C - counter	3
D - BCD	1
F - floating point	2
MG - message	56
N - integer (signed, unsigned, 2s compliment, BCD)	1
PD - PID controller	82
R - control	3
SC - SFC status	3
ST - ASCII string	42
T - timer	3

Catatan: Memory menunjukkan file maupun lokasi. Pengaturan file adalah spesifik bagi manufaktur PLC tertentu dan tidak dikenal pada tipe lainnya

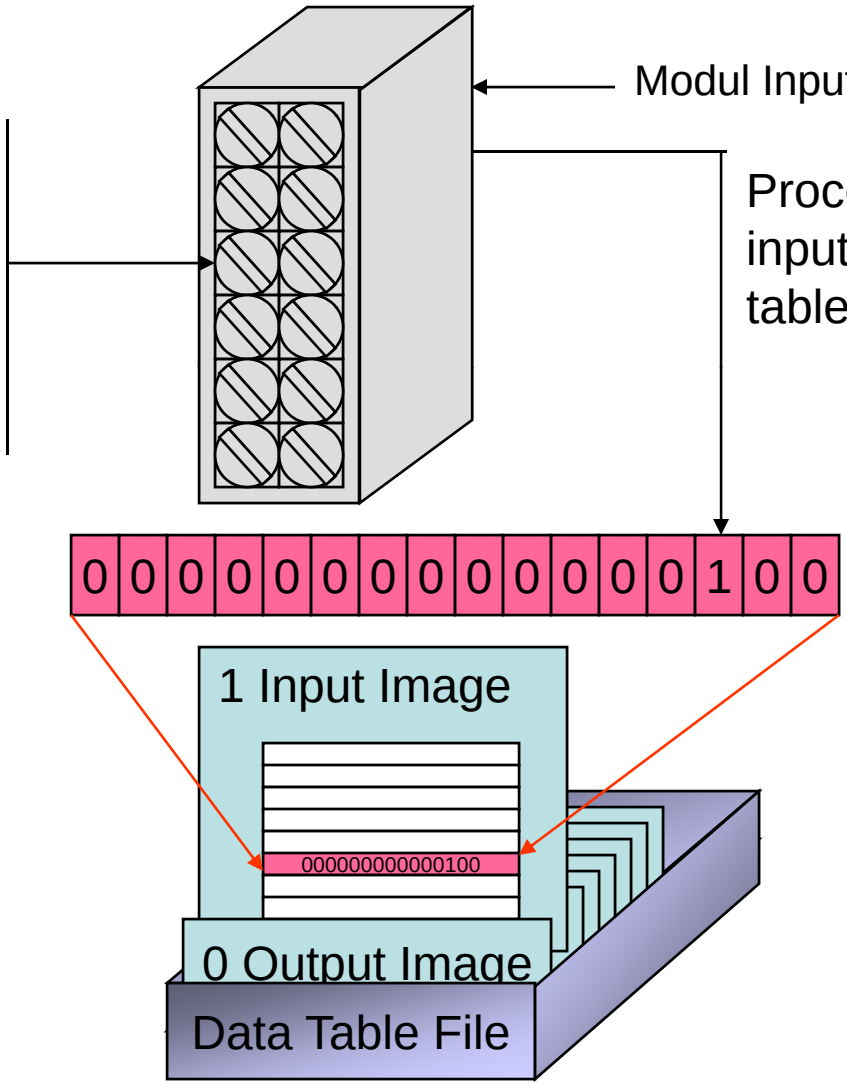
Tipe Data AB SLC 500



Typical Address Format

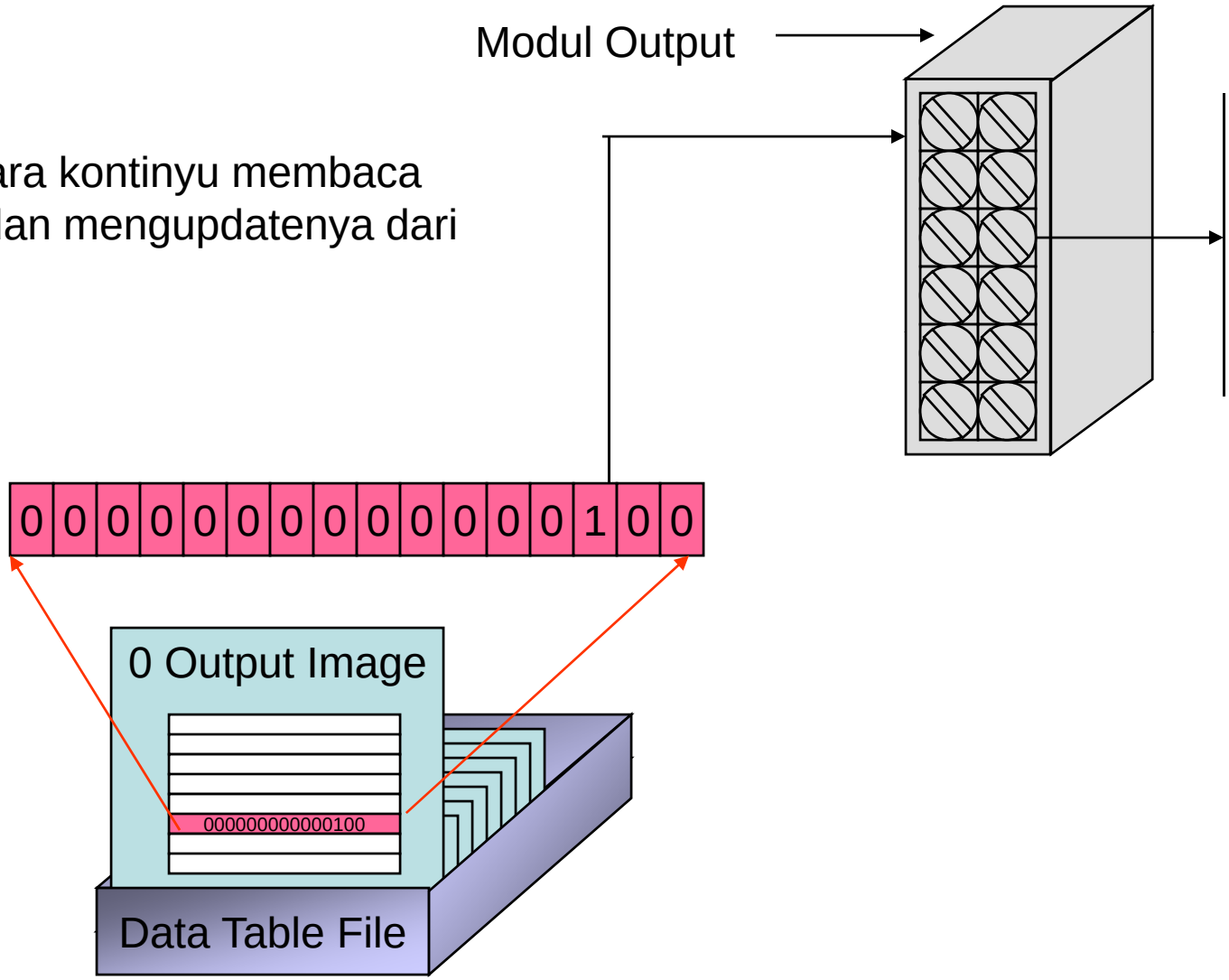
- Output data file dan Input data file merepresentasikan lokasi dari channel dan modul dalam chassis PLC
- Status data file mengandung informasi mengenai status prosesor
- Bit data file menyimpan status bit. Bisa juga digunakan untuk internal output, sequencer, instruksi bit shift dan intruksi logika :
- Timer file menyimpan status timer dan data timer. Elemen timer terdiri dari 3 word, yaitu : control word, preset word dan accumulate word.
- Counter file menyimpan counter status dan data counter. Elemen counter terdiri dari 3 word, yaitu : control word, preset word dan accumulate word
- Control file menyimpan status elemen control dan data, digunakan untuk mengontrol berbagai instruksi file. Elemen control terdiri dari 3 word, yaitu : control word, length word dan position word.
- Integer file dapat menyimpan besaran data integer dengan rentang -32.768 sampai 32.767. disimpan dalam bentuk desimal. Elemen integer adalah elemen tunggal word 16 bits.
- Floating-point file menyimpan besaran dengan rentang $\pm 1.1754944e^{-38} sampai \pm 3.402837e^{+38}$

Koneksi Input Image dengan Table File



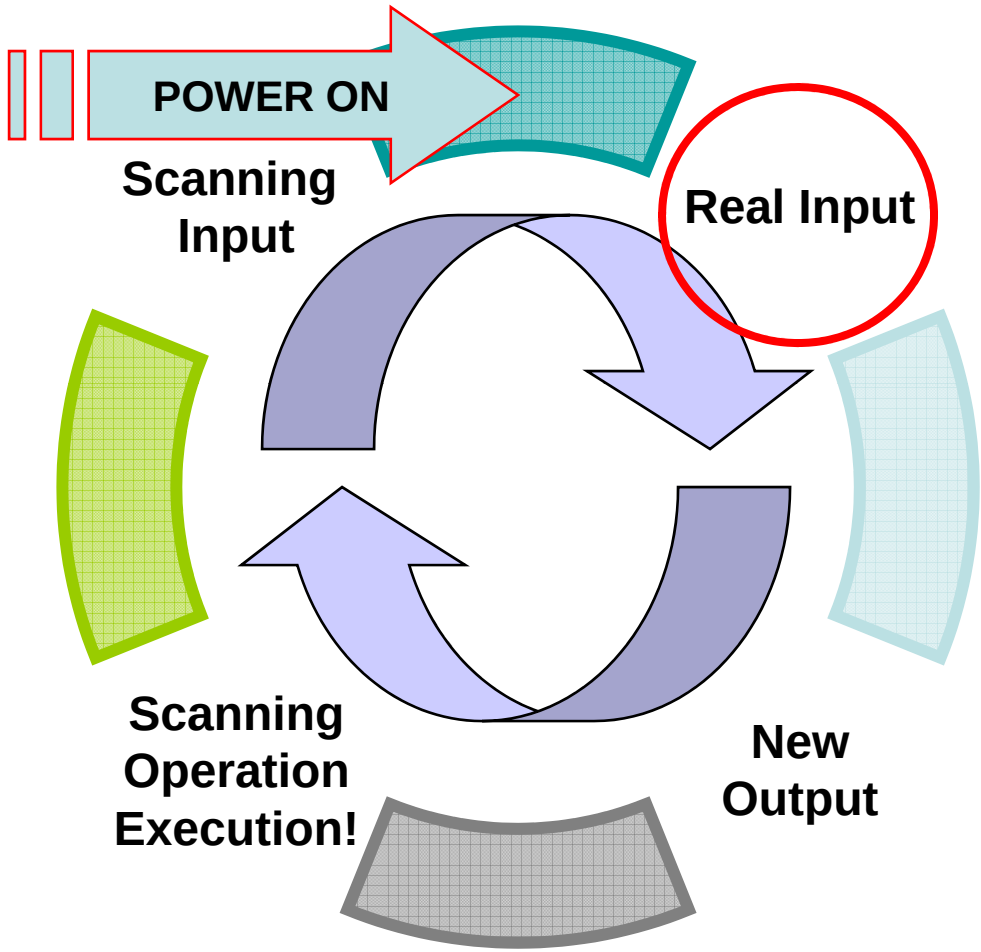
Koneksi Output Image dengan Table File

Processor secara kontinyu membaca Output status dan mengupdatenya dari image table file

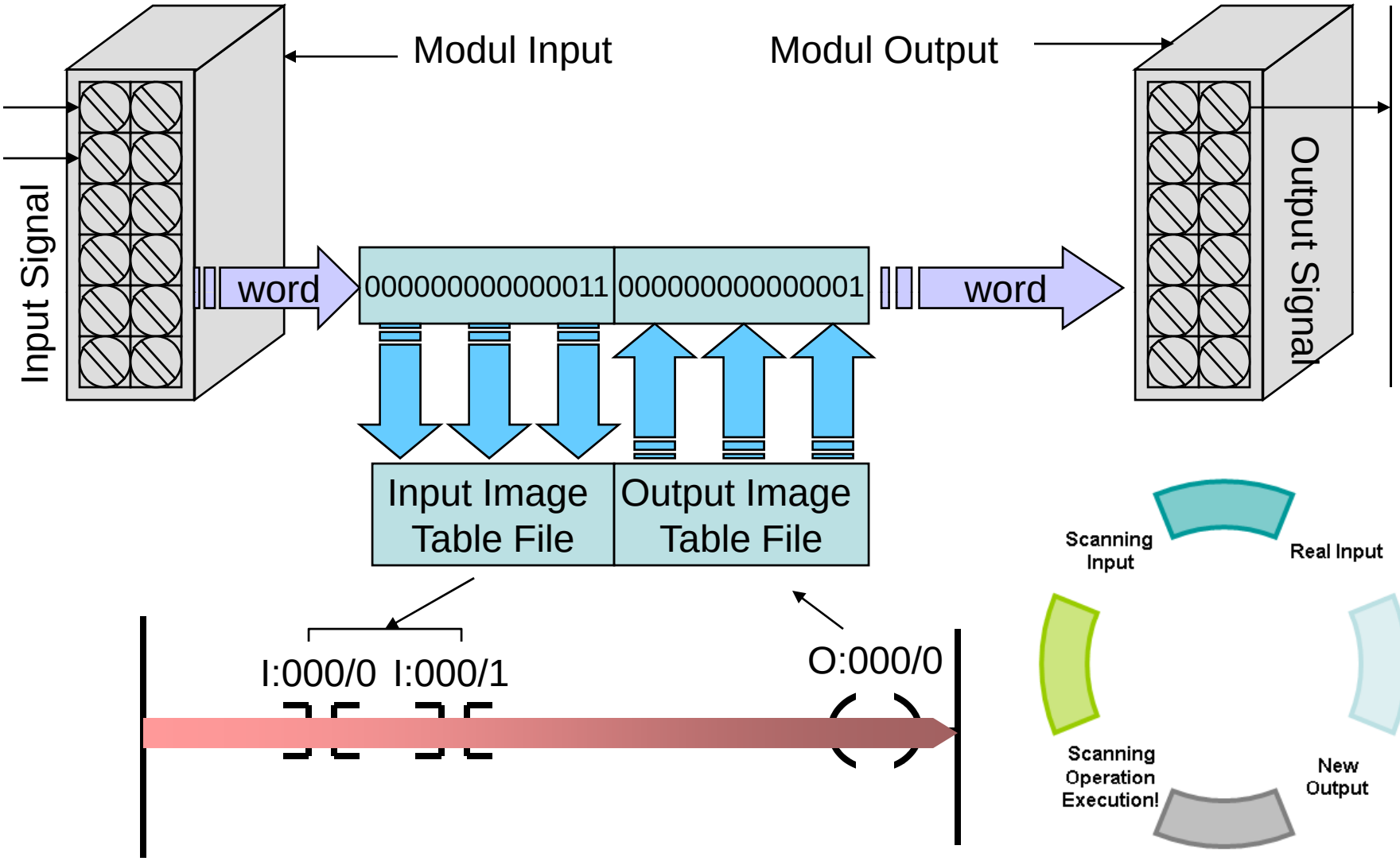


Program Scan

Program Scan adalah siklus kontinyu dari pembacaan input PLC, memecahkan logika LLD, kemudian mengubah output



Scan Process



Address

- Penggunaan data file dan functions selalu terkait dengan informasi berupa address.
- Data address yang paling sederhana adalah data bits
- Memory bits ditandai dengan tanda garis miring diikuti oleh bit number */n*.

Addressing Bit Level

- Addressing bit menyerupai addressing sebuah output sebagai suatu data bit.
- Contohnya:

  I:000/02 → input bit ketiga dari input card I:000

  C5:4/DN → bit DN dari suatu counter

  B3/4 → bit keempat pada memory bit

Catatan:
- Beberapa bit address, terutama input dan output menggunakan sistem bilangan oktal.
- Hal ini sering menimbulkan kecerobohan. Contohnya jika menggunakan output bit kesebelas, atau bit 10, maka address oktalnya adalah 12.

1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th	15th	16th
00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17

Addressing Integer Word

- Integer word – 16 bit dapat dianggap sebagai suatu integer.
- Contohnya:

  N7:8 → nilai kesembilan dari memori integer

  I:000 → suatu integer dengan seluruh nilai dari input card

  T4:7.ACC → nilai accumulator dari suatu timer

  T4:7.PRE → nilai preset dari suatu timer

Nilai Data Literal

- Nilai data tidak selalu perlu disimpan dalam memory, melaikan dapat juga didefinisikan secara literal/harfiah.
- Contohnya:

8	→ suatu integer
8.5	→ suatu floating point number
08FH	→ nilai heksadesimal dari 8F
01101101B	→ nilai biner dari 01101101

Addressing File

- File terkadang digunakan untuk menyatakan suatu deret nilai.
- File menyatakan lokasi awal dari deret nilai-nilai data.
- Contohnya:

  #F8:5 → menyatakan grup nilai dimulai dari F8:5

  #N7:0 → menyatakan grup nilai dimulai dari N7:0

Indirect Addressing

- Indirect addressing merupakan metoda untuk memungkinkan suatu variabel dinyatakan dalam suatu address data
- Lokasi memory lainnya dapat digunakan pada deskripsi suatu lokasi
- Contohnya:

  I:000/[N7:2] → jika lokasi N7:2 bernilai 5 maka akan menjadi I:000/05

  I:[N7:1]/03 → jika lokasi memori integer bernilai 2 maka menjadi I:002/03

  #I:[N7:1] → jika lokasi memori integer bernilai 2 maka file akan dimulai dari I:002

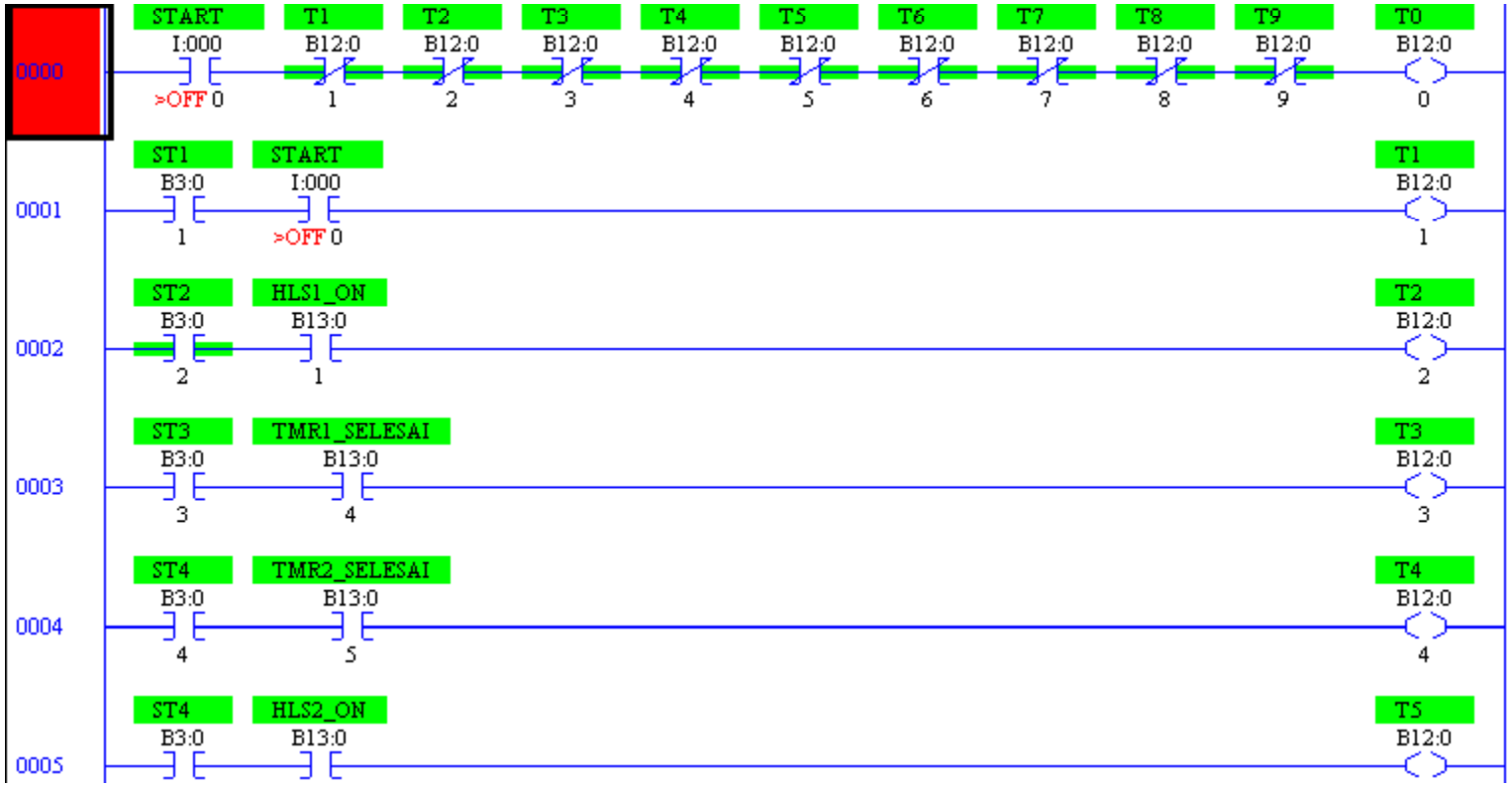
  N[N7:0]:8 → jika N7:1 bernilai 10 maka data address menjadi N10:8

Nilai Data Expression

- Expression memungkinkan address dan function ditulis dan diinterpretasikan saat program berjalan (run).
- Expression merupakan suatu text string yang menjabarkan operasi yang kompleks.
- Contohnya:

  “sin(F8:3) + 1.3” → merupakan suatu kalkulasi sederhana

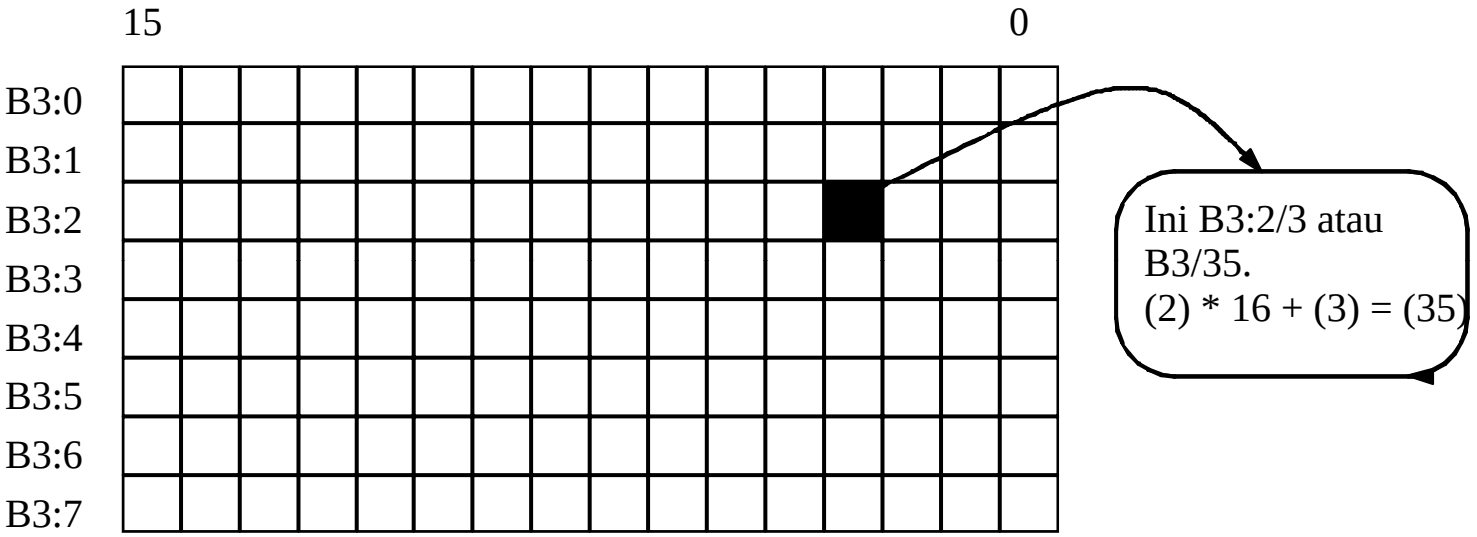
Contoh Ladder Logic Functions



User Bit Memory

- Individual data bits dapat diakses pada memory bit.
- Bit memory dapat diakses dengan individual bit atau integer word.

Bit Memory



Contoh lainnya : B3:0/0 = B3/0
B3:0/10 = B3/10
B3:1/0 = B3/16
B3:1/5 = B3/21
B3:2/0 = B3/32
dll...

Timer Memory

- Default timer disimpan pada T4
- Bit dan word untuk timer:
  - EN → timer enabled bit (bit 15)
  - TT → timer timing bit (bit 14)
  - DN → timer done bit (bit 13)
  - PRE → preset word
  - ACC → accumulated time word

Counter Memory

- Counter disimpan pada C5.
- Bit dan word untuk counter:
  - CU → count up bit (bit 15)
  - CD → count down bit (bit 14)
  - DN → counter done bit (bit 13)
  - OV → overflow bit (bit 12)
  - UN → underflow bit (bit 11)
  - PRE → preset word
  - ACC → accumulated count word

Bit dan Words Status PLC

- Status memory memungkinkan program memeriksa operasi PLC, dan melakukan perubahan.

S2:0/0 carry in math operation
S2:0/1 overflow in math operation
S2:0/2 zero in math operation
S2:0/3 sign in math operation
S2:1/15 first scan of program file
S2:8 the scan time (ms)
S2:18 year
S2:19 month
S2:20 day
S2:21 hour
S2:22 minute
S2:23 second
S2:28 watchdog setpoint
S2:29 fault routine file number
S2:30 STI (selectable timed interrupt) setpoint
S2:31 STI file number
S2:46-S2:54,S2:55-S2:56 PII (Programmable Input Interrupt) settings
S2:55 STI last scan time (ms)
S2:77 communication scan time (ms)

User Function Control Memory

- Ladder logic function sederhana dapat menyelesaikan operasi dalam sekali scan ladder logic.
- Fungsi lainnya seperti timer dan counter akan memerlukan beberapa kali ladder scan logic hingga selesai.
- Bit dan word control memory:

  EN → enable bit (bit 15)
  EU → enable unload (bit 14)
  DN → done bit (bit 13)
  EM → empty bit (bit 12)
  ER → error bit (bit 1)
  UL → unload bit (bit 10)
  IN → inhibit bit (bit 9)
  FD → found bit (bit 8)
  LEN → length word
  POS → position word

Integer Memory

- Integer memory merupakan 16 bit words yang biasanya digunakan untuk menyimpan nilai data dari -32768 sampai dengan +32767.
- Pecahan desimal akan dibulatkan ke bilangan terdekat.
- Nilai ini secara default disimpan pada N7:xx, namun dapat juga dibuat blok integer memory pada lokasi lainnya contohnya *N9:xx*
- Integer memory dapat juga digunakan untuk bit.

Floating Point Memory

- Memory yang menyimpan bilangan real pada 4 word, dengan 7 digit akurasi pada selang +/-1.1754944e-38 sampai dengan +/-3.4028237e38.
- Floating point memory secara default disimpain pada F8:xx, namun dapat juga disimpan pada lokasi lainnya.
- Bit level access tidak dapat digunakan pada tipe floating point.

Numerical System and Code

Sasaran

- Mendefinisikan sistem penomoran desimal, binari, oktal, dan heksadesimal
- Mampu untuk mengkonversi dari satu sistem ke sistem berikutnya
- Mendefinisikan bit, byte, word, LSB, MSB, aplikasinya ke lokasi memori

Sistem bilangan

- Data internal PLC dipresentasikan dalam format biner
- Beberapa PLC menggunakan sistem bilangan desimal dan oktal untuk addressing
- Data analog input dikonversikan ke data biner ketika dimasukan ke PLC
- Data analog output dikonversikan dari biner sebagai sinyal ke alat output analog
- Sistem bilangan yang umum dipakai pada PLC adalah:
  - Desimal
  - Biner
  - Oktal
  - Heksadesimal

Sistem Desimal

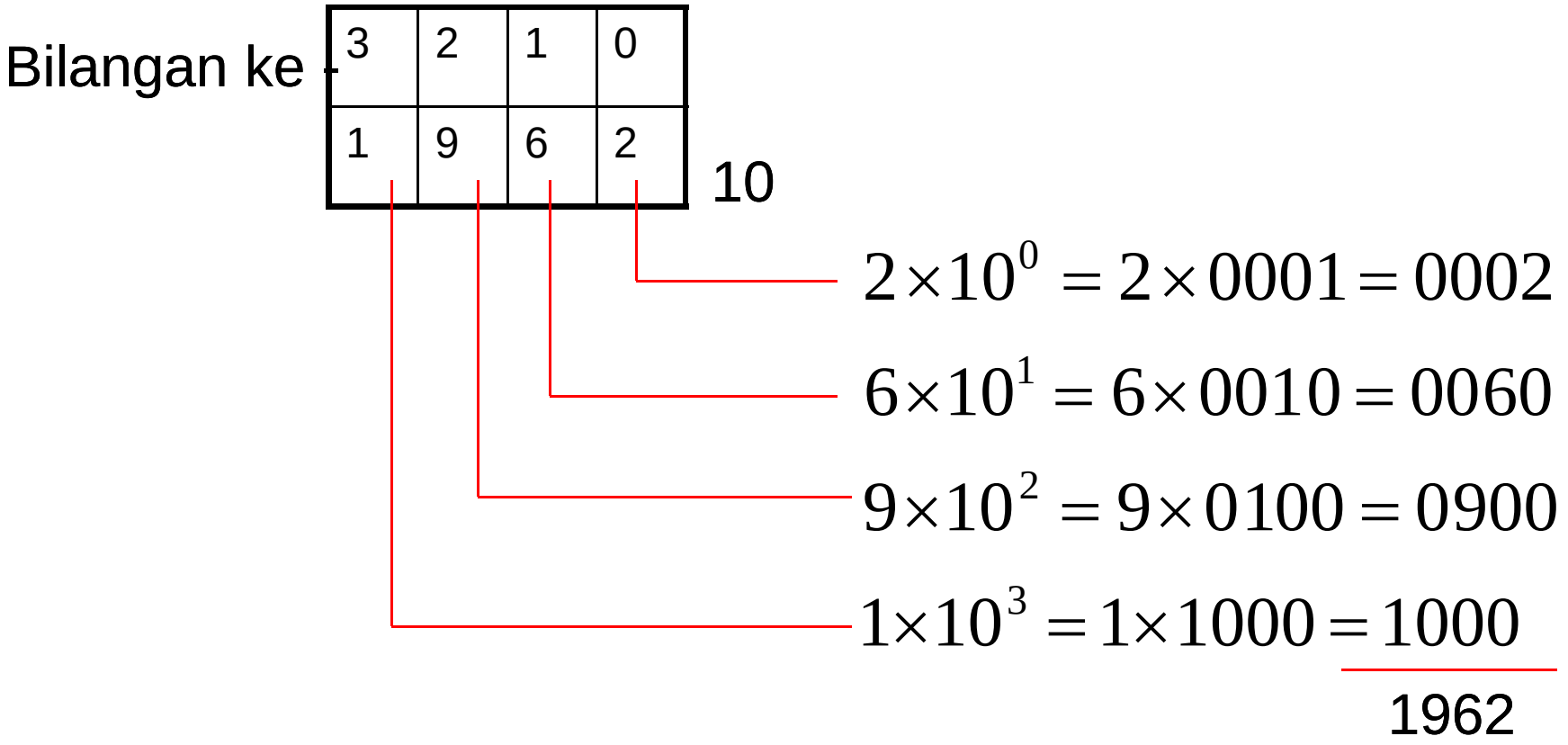
- Sistem desimal mempunyai basis 10
- Radix atau base adalah jumlah total dari simbol berbeda atau digit yang digunakan
- Untuk sistem desimal, digunakan 10 nomor unik, darir 0 sampai 9

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

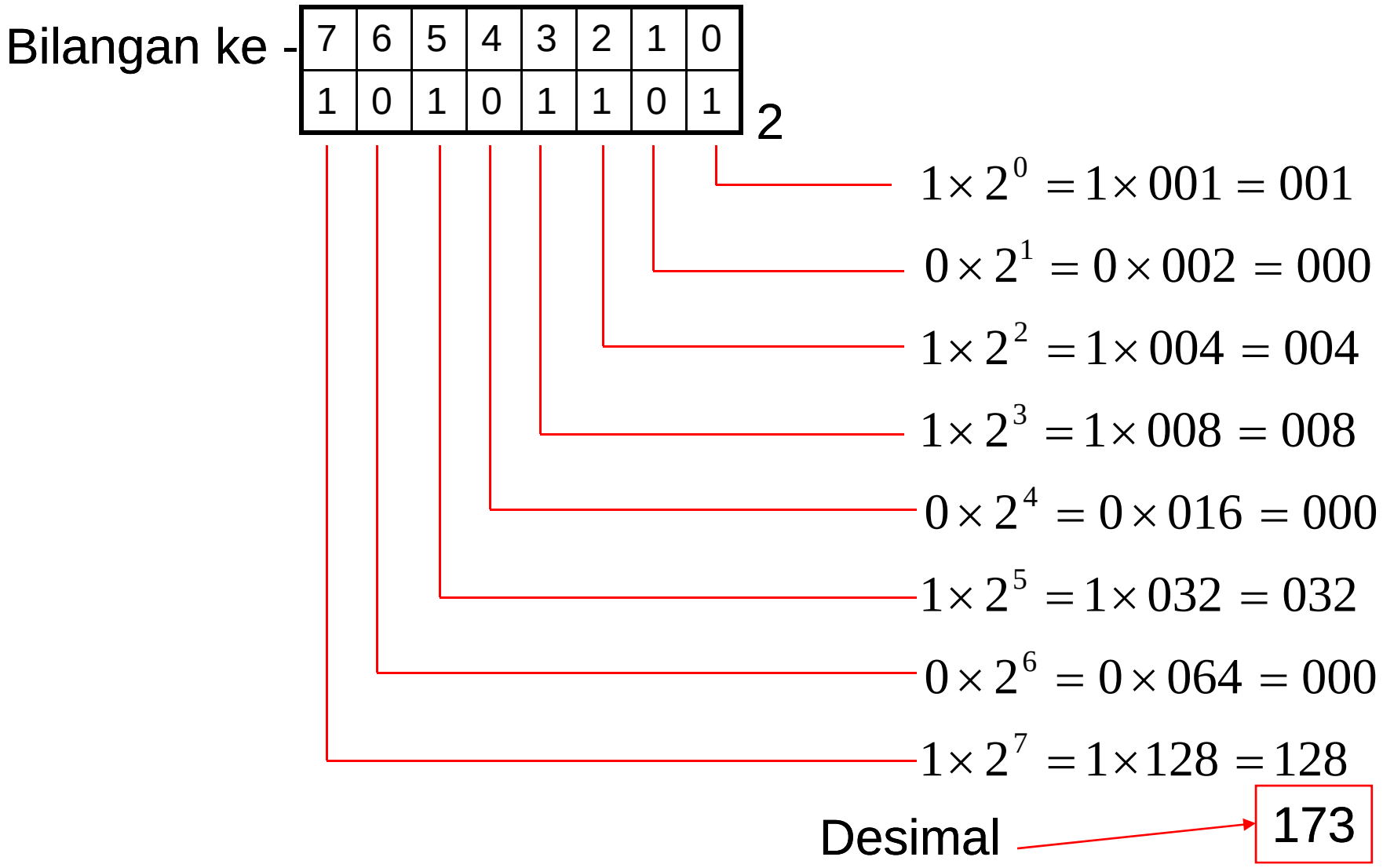
Sistem Biner

- Sistem biner mempunyai basis 2
- Digit yang digunakan hanya 0 dan 1
- Dengan rangkain elektrik sistem biner dapat dengan mudah diimplementasikan dengan tegangan 0 dan 5 Volt

Besarnya Nilai Pada Sistem Desimal



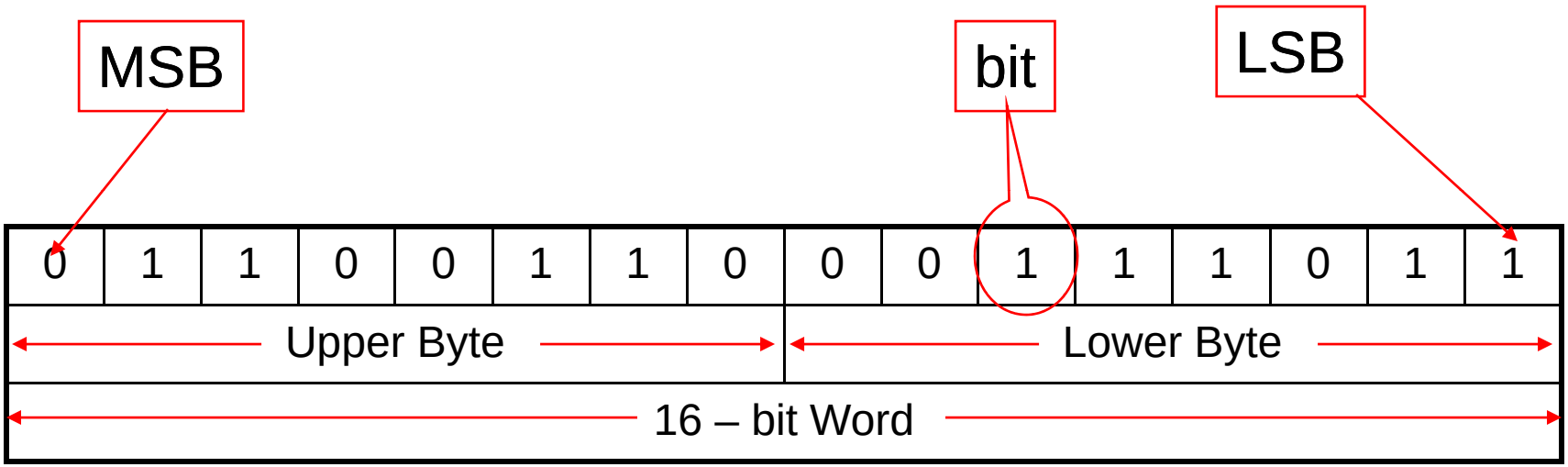
Konversi Sistem Biner ke Desimal



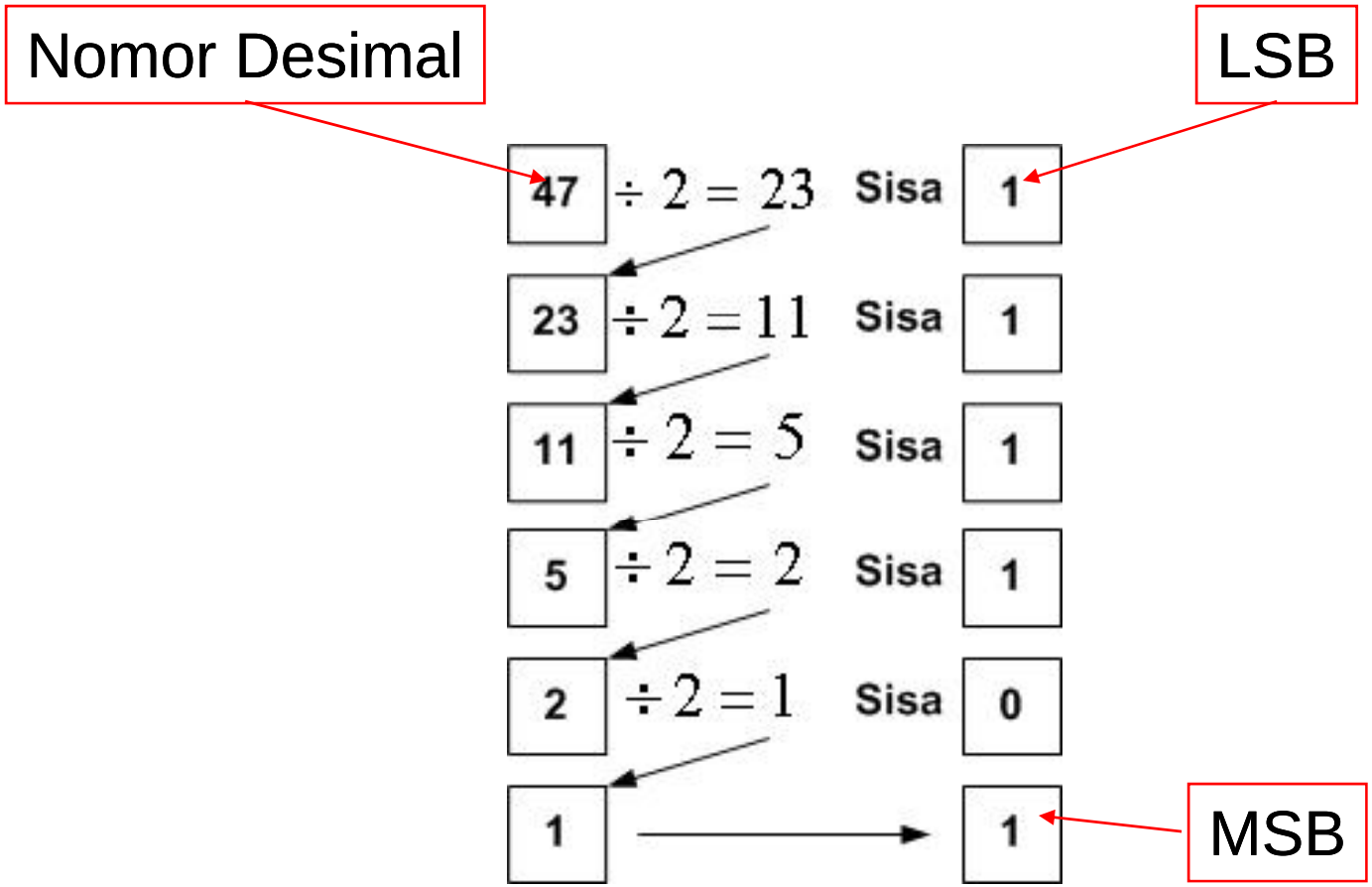
Perbandingan Sistem Penomeran

Desimal	Oktal	Hexadesimal	Biner
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	10	8	1000
9	11	9	1001
10	12	A	1010
11	13	B	1011
12	14	C	1100
13	15	D	1101
14	16	E	1110
15	17	F	1111
16	20	10	10000
17	21	11	10001
18	22	12	10010
19	23	13	10011
20	24	14	10101

16-bit Word


MSB
bit
LSB
0 1 1 0 0 1 1 0 0 0 1 1 1 0 1 1
Upper Byte
Lower Byte
16 – bit Word

Mengkonversi Nilai Desimal ke Biner



Nilai Negasi

- Jika nilai desimal positif, maka mempunyai tanda +
- Jika nilai desimal negatif, maka mempunyai tanda –
- Di sistem biner, metode untuk penendaan +/- digunakan ekstra digit/sign bit

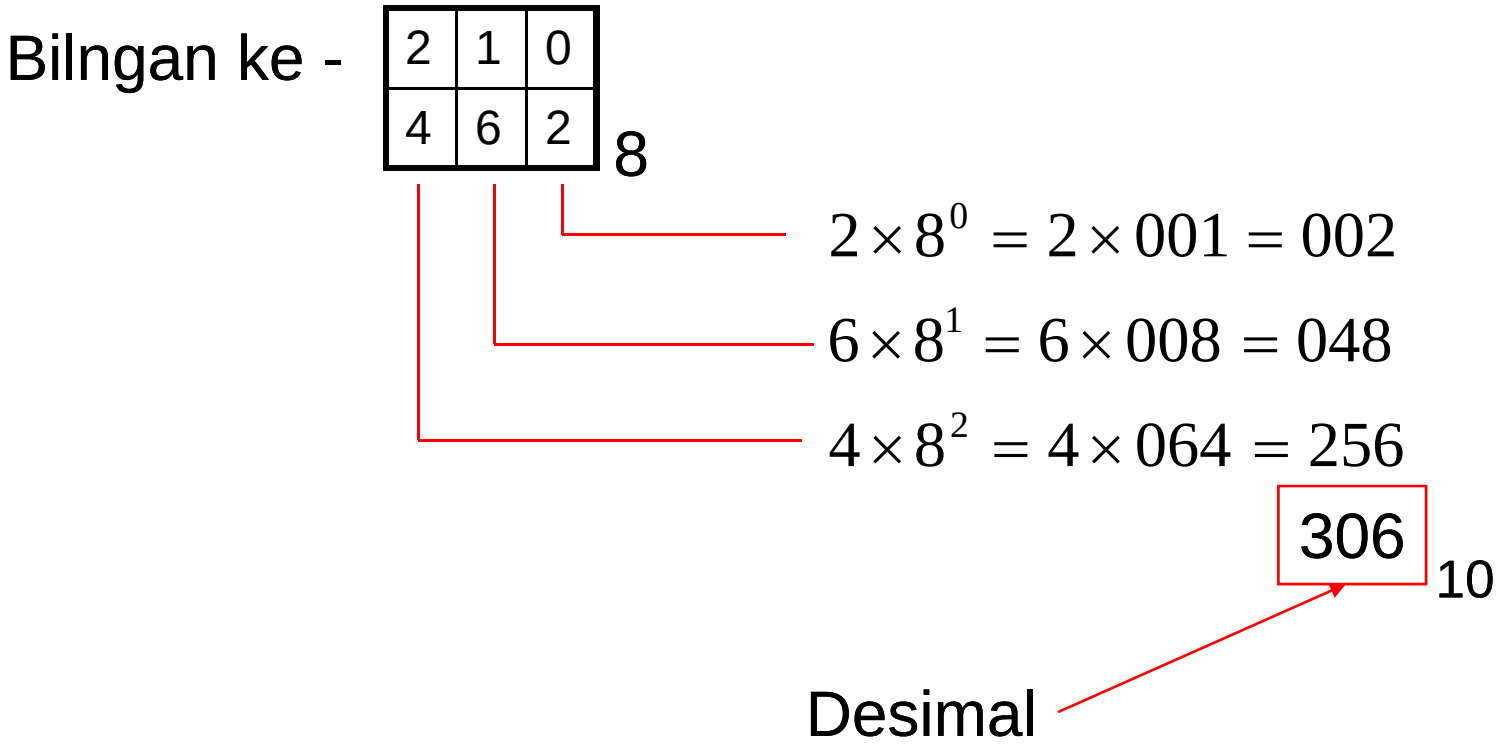
Signed Decimal	1's Complement	2's Complement
7	0111	0111
6	0110	0110
5	0101	0101
4	0100	0100
3	0011	0011
2	0010	0010
1	0001	0001
0	0000	0000
-1	1110	1111
-2	1101	1110
-3	1100	1101
-4	1011	1100
-5	1010	1011
-6	1001	1010
-7	1000	1001

Sistem Oktal

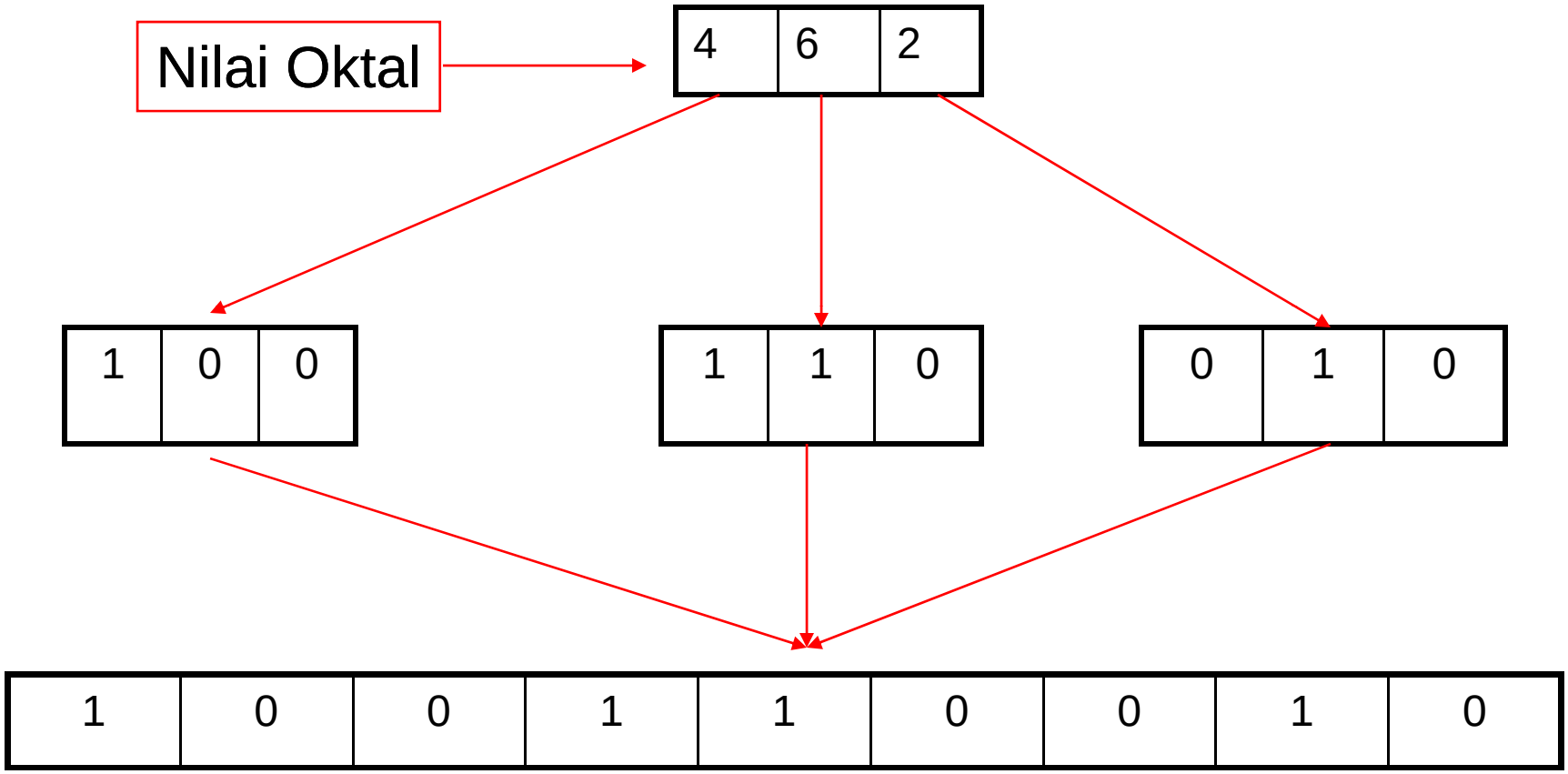
- Sistem Oktal berbasis angka 8 sering digunakan dalam processor PLC
- Bit 8 data sejumlah byte informasi dapat dialamatkan dalam pemrograman

Pengalamatan I/O menggunakan sistem Oktal			
Input Module 0	**Input Module 1**	**Input Module 2**	**Input Module 3**
I00	**I10**	**I20**	**I30**
I01	**I11**	**I21**	**I31**
I02	**I12**	**I22**	**I32**
I03	**I13**	**I23**	**I33**
I04	**I14**	**I24**	**I34**
I05	**I15**	**I25**	**I35**
I06	**I16**	**I26**	**I36**
I07	**I17**	**I27**	**I37**

Mengkonversi Nilai Oktal ke Desimal



Mengkonversi Nilai Oktal ke Biner



BCD System

- Sistem BCD menyediakan cara mudah dalam menangani sejumlah besar I/O PLC.
- Sangatlah tidak mudah dalam mengkonversi nilai dari desimal ke biner atau sebaliknya.

Perbandingan BCD dengan Biner

Decimal	BCD	Biner
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	0001 0000	1010
11	0001 0001	1011
12	0001 0010	1100
13	0001 0011	1101
14	0001 0100	1110
15	0001 0101	1111

Representasi BCD dari nilai Desimal

