Building a better future
Global Leader

| Technical Support Document for Inverter | |
|---|---|
| Doc. No. | TSD-N700-COM-001E(00) |
| Model | All N700 model |
| Rev. Date | June, 2009 |

# N700 Inverter RS232/485 Interface Instruction Manual

We build a better future
HYUNDAI
HEAVY INDUSTRIES CO.,LTD.
www.hhi.co.kr

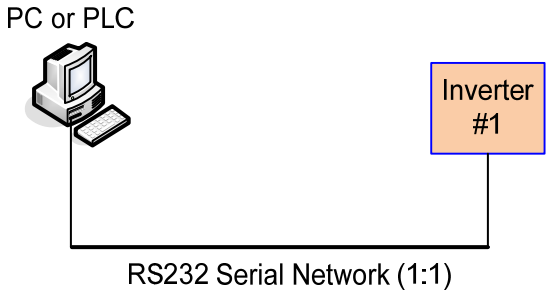
# Contents

# 1. Serial communication

## 1.1 RS232 communication protocol

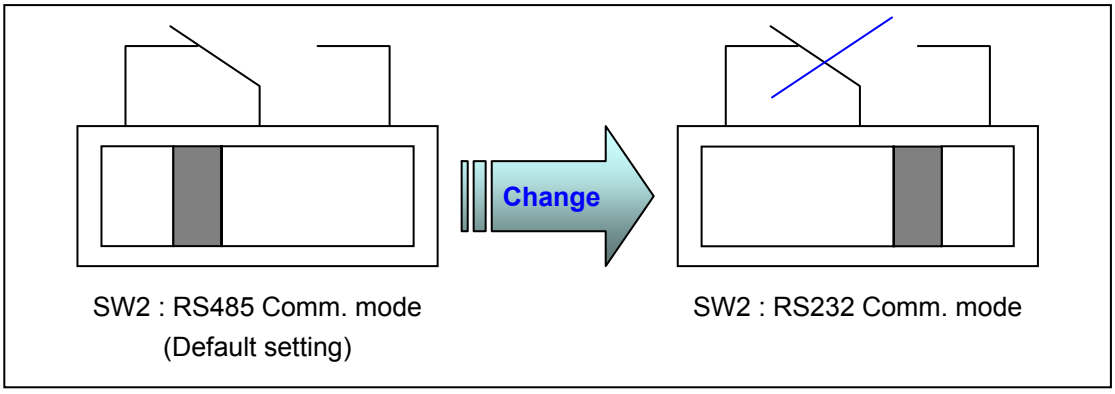
Communication circuit for RS232 is built in N700 Inverter.

It is possible to control 1 Inverters (Slave) from a main control device (Master) by using RS232 serial communication is supported.

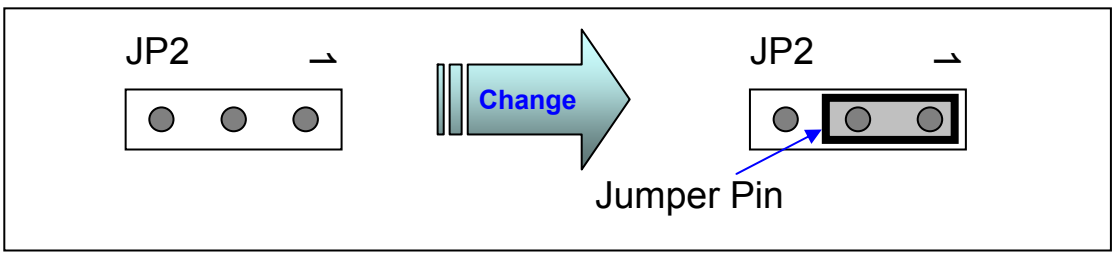



### 1.1.1 RS232 setting in a inverter

(1) Change SW2 switch into the RS232 mode in the control terminal PCB.




(2) Short the end inverter by using JP2 jumper in the control terminal PCB.
(In case of RS232 network, there is no necessity for JP2 jumper setting)



(3) Function code setting in Inverter

① b037 : Data command
Change a default value 0(operator) into 4(RS232), and save the value.

② b038 : Communicating transmission speed
Confirm a default value 2(9600bps) or change a parameter to reasonable setting.

③ b039 : Communication code
Confirm a default value 1(address No. 1) or change a parameter to reasonable setting.

④ b040 : Communication bit
Confirm a default value 8(8bit) or change a parameter to reasonable setting.

⑤ b041 : Communication parity
Confirm a default value 0(No parity) or change a parameter to reasonable setting.

⑥ b042 : Communication stop bit
Confirm a default value 1(1bit) or change a parameter to reasonable setting.

### 1.1.2 Cable connection between PC(or PLC) and Inverter

**PC**

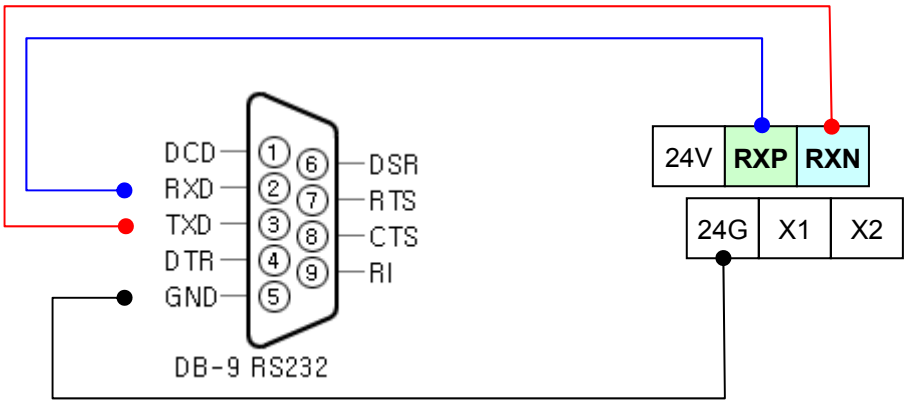
| Name | Pin No. |
|---|---|
| DCD | 1 |
| **RXD** | **2** |
| **TXD** | **3** |
| DTR | 4 |
| SG(GND) | 5 |
| DSR | 6 |
| RTS | 7 |
| CTS | 8 |
| RI | 9 |

<PC COM port (DB-9 male)>

**N700 Inverter**

| Terminal |
|---|
| **RXP** |
| **RXN** |
| **24G** |

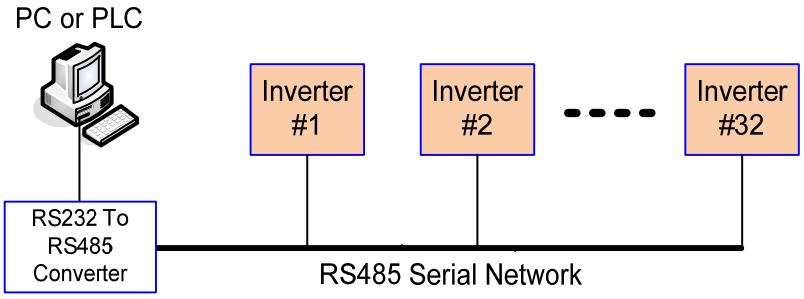
<Control Terminal PCB>



## 1.2 RS485 Communication Function

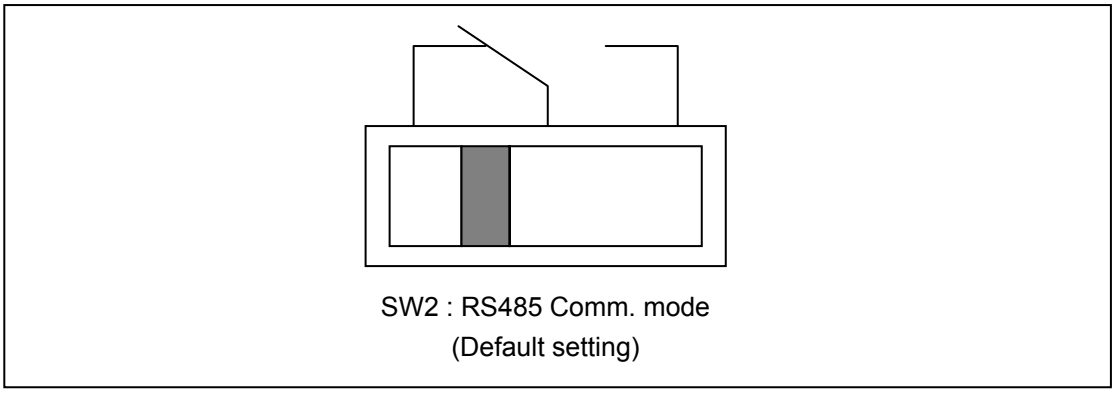
Communication circuit for RS485 is built in N700 Inverter.

It is possible to control 1~32 Inverters (Slave) from a main control device (Master) by using RS485 serial communication is supported.



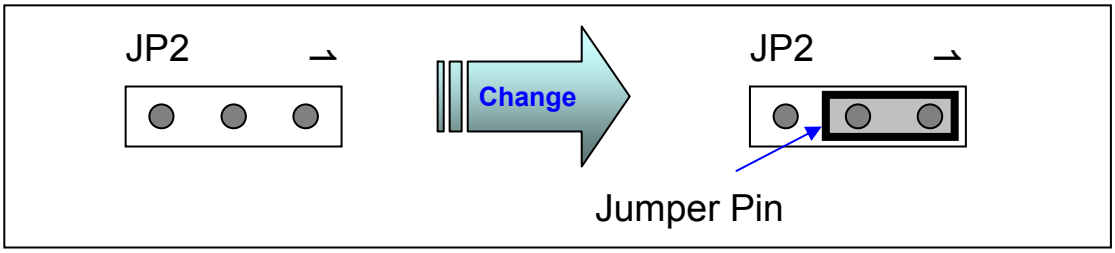

### 1.2.1 RS485 setting in a inverter

(1) Confirm SW2 switch status for RS485 mode in the control terminal PCB.




(2) Short the end inverter by using JP2 jumper in the control terminal PCB. (Although RS485 communication is used to one inverter, short JP2 jumper as an equal method.)

RS485 communication requires a terminaltion resistor. The N700 has built-in terminal resistor that by shorting JP2 jumper, the termination resistor of the control terminal is valid and enables to control the reflection of a signal.



(3) Function code setting in Inverter

① b037 : Data command
Change a default value 0(operator) into 1(RS-485), and save the value.

② b038 : Communicating transmission speed
Confirm a default value 2(9600bps) or change a parameter to reasonable setting.

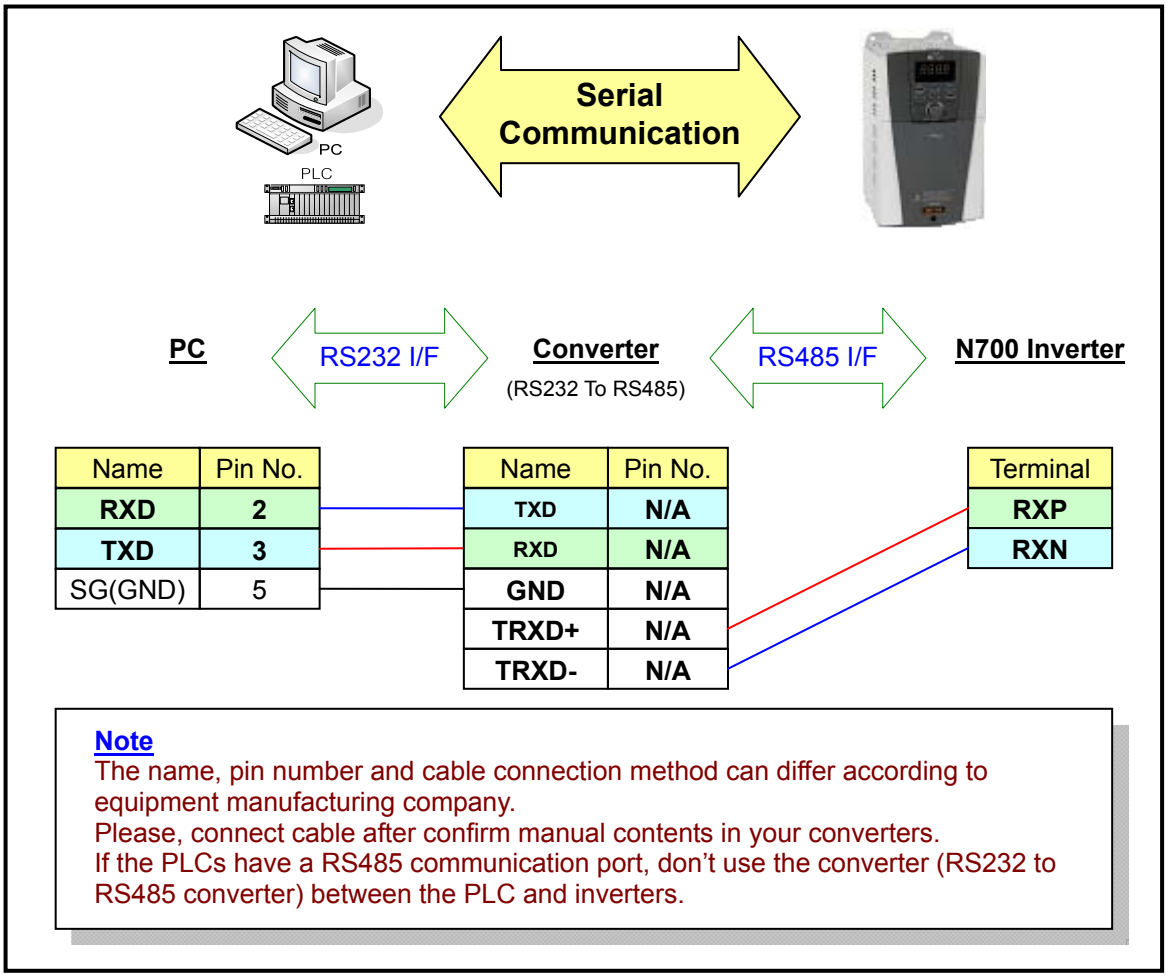
③ b039 : Communication code
Confirm a default value 1(address No. 1) or change a parameter to reasonable setting.

④ b040 : Communication bit
Confirm a default value 8(8bit) or change a parameter to reasonable setting.

⑤ b041 : Communication parity
Confirm a default value 0(No parity) or change a parameter to reasonable setting.

⑥ b042 : Communication stop bit
Confirm a default value 1(1bit) or change a parameter to reasonable setting.

### 1.2.2 Cable connection between PC(or PLC) and Inverter




| Name | Pin No. |
|---|---|
| RXD | 2 |
| TXD | 3 |
| SG(GND) | 5 |

| Name | Pin No. |
|---|---|
| TXD | N/A |
| RXD | N/A |
| GND | N/A |
| TRXD+ | N/A |
| TRXD- | N/A |

| Terminal |
|---|
| RXP |
| RXN |

**Note**
The name, pin number and cable connection method can differ according to equipment manufacturing company.
Please, connect cable after confirm manual contents in your converters.
If the PLCs have a RS485 communication port, don't use the converter (RS232 to RS485 converter) between the PLC and inverters.

# 2. Modbus communication protocol

## 2.1 Function code in Inverter

N700 Inverter provides external control equipment with Modbus RTU protocol.

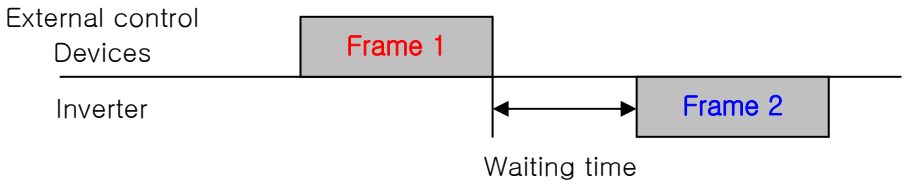
Set the inverter function code using the below table in order to communicate with external control equipment by RS485 communication.

| Function code | Set item | Initial value | Data | Description |
|---|---|---|---|---|
| b037 | Data command | 0 | 0(OPERATOR) | OPERATOR |
| | | | 1(RS485) | RS485 |
| | | | 2(OPT1) | Option 1 |
| | | | 3(OPT2) | Option 2 |
| | | | 4(RS232) | RS232 |
| b038 | Communication transmission speed | 2 | 0(2400BPS) | 2,400 BPS |
| | | | 1(4800BPS) | 4,800 BPS |
| | | | 2(9600BPS) | 9,600 BPS |
| | | | 3(19200BPS) | 19,200 BPS |
| | | | 4(38400BPS) | 38,400 BPS |
| b039 | Communication code | 1 | 1~32 | This assigns the station number of the inverter. This is used when you control more than one simultaneously. |
| b040 | Communication code | 8 | 7(7BIT) | 7bits |
| | | | 8(8BIT) | 8bits |
| b041 | Communication parity | 0 | 0(NO PARITY) | No parity |
| | | | 1(EVEN PARITY) | Even parity |
| | | | 2(ODD PARITY) | Odd parity |
| b042 | Communication stop bit | 1 | 1(1BIT) | 1bits |
| | | | 2(2BIT) | 2bits |

※ Among above code, other code alteration is incomprehensible using using operator after change b037 code by communication.
Please change b037 code value on last certainly.

## 2.2 Communication order

The flow of the communication protocol between an external control device and an inverter is shown below in the time diagram.




Frame start : Frame start is recognized by signal line data transmitted.

Frame completion: Frame completion is recognized by no data during correspond 4, 5-character time.

Frame 1: Transmit from external controller to inverter.

Frame 2: Indication reflects from inverter to external controller.
Frame 2 in inverter displays as the signal that inverter receiving Frame 1 and recognizes a suitable frame and responds, and don't output actively.

## 2.3 Read command

### 2.3.1 Request frame

| 1Byte | 1Byte | 2Byte | 2Byte | 1Byte | 1Byte |
|---|---|---|---|---|---|
| **Comm. number** | **Command** | **Parameter** | **Parameter count** | **CRC Hi** | **CRC Lo** |

| Clause | Description | Data size | Specifications |
|---|---|---|---|
| Communication number | Inverter communication number | 1 byte | 1~32 |
| Command | Frame type | 1 byte | 0x03 ➔ Read command |
| Parameter | Parameter | 2 byte | 1st byte : Group<br>D(0x01) F(0x02) U(0x03)<br>A(0x04) B(0x05) I(0x06)<br>O(0x07) C(0x08) H(0x09)<br>P(0x0a)<br>2nd byte : index (Index) |
| Parameter count | Request parameter number | 2 byte | 1st byte : 0x00<br>2nd byte : N(0x01~0x08) |
| CRC Hi | - | 1 byte | Higher 8bit of 16bit CRC |
| CRC Lo | - | 1 byte | Lower 8bit of 16bit CRC |

* (예) 010301010001D436

① 01 : Inverter communication number is 01.

② 03 : Read command

③ 0101 : D001(Output frequency)

④ 0001 : Parameter count is 1.

⑤ D436 : D4 ➔ Higher 8bit of 16bit CRC  36 ➔ Lower 8bit of 16bit CRC

### 2.3.2 Response frame

| 1Byte | 1Byte | 1Byte | 2Byte | | 2Byte | 1Byte | 1Byte |
|---|---|---|---|---|---|---|---|
| **Comm. number** | **Order** | **Byte number** | **Data1** | …… | **Data N** | **CRC Hi** | **CRC Lo** |

| Clause | Description | Data size | Specifications |
|---|---|---|---|
| Communication number | Inverter communication number | 1 byte | 1~32 |
| Command | Frame type | 1 byte | 0x03 |
| Request Byte | Data byte number | 1 byte | Request parameter number × 2 |
| Data 1 | Parameter 1 | 2 byte | Parameter value |
| Data N | Parameter N | 2 byte | Nth parameter value |
| CRC Hi | - | 1 byte | Higher 8bit of 16bit CRC |
| CRC Lo | - | 1 byte | Lower 8bit of 16bit CRC |

※ Frame Size = 5byte (Communication number + Command + Request Byte + CRC H + CRC L)
+ Request parameter number × 2byte (Data1 + Data2 + ….. + Data N)

### 2.3.3 Transmit frame [Example]

(1) A prior condition

① Communication number : 01 (Inverter address is 1)

② Command : 03 (Read Request)

③ Parameter count : 0001 (Parameter count is 1)

(2) Output frequency monitoring (Function code : D001)

① Request frame

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Parameter Count | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010301010001D436 | 01 | 03 | 01 | 01 | 00 | 01 | D4 | 36 |

② Response frame (In case of D001 = 0.00Hz)

| Response (7Byte) | Com. Num. | Com-mand | Byte Q'ty | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|
| 0103020000B844 | 01 | 03 | 02 | 00 | 00 | B8 | 44 |
| Data Description | $0x16^3 + 0x16^2 + 0x16^1 + 0x16^0 = 0$(Communication Data) ➔ divide by 100 ➔ **0.00Hz (Parameter value)** | | | | | | |

③ Response frame (In case of D001 = 60.00Hz)

| Response (7Byte) | Com. Num. | Com-mand | Byte Q'ty | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|
| 0103021770B650 | 01 | 03 | 02 | 17 | 70 | B6 | 50 |
| Data Description | $1x16^3 + 7x16^2 + 7x16^1 + 0x16^0 = 6000$(Communication Data) ➔ divide by 100 ➔ **60.00Hz (Parameter value)** | | | | | | |

(3) Output frequency (Function code : F001)

① Request frame

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Parameter Count | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010302010001D472 | 01 | 03 | 02 | 01 | 00 | 01 | D4 | 72 |

② Response frame (In case of F001 = 60.0Hz)

| Response (7Byte) | Com. Num. | Com-mand | Byte Q'ty | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|
| 0103021770B650 | 01 | 03 | 02 | 17 | 70 | B6 | 50 |
| Data Description | $1x16^3 + 7x16^2 + 7x16^1 + 0x16^0 = 6000$(Communication Data) ➔ divide by 100 ➔ **60.00Hz (Parameter value)** | | | | | | |

③ Response frame (In case of F001 = 50.00Hz)

| Response (7Byte) | Com. Num. | Com-mand | Byte Q'ty | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|
| 0103021388B512 | 01 | 03 | 02 | 13 | 88 | B5 | 12 |
| Data Description | $1x16^3 + 3x16^2 + 8x16^1 + 8x16^0 = 5000$(Communication Data) ➔ divide by 100 ➔ **50.00Hz (Parameter value)** | | | | | | |

(4) Acceleration time (Function code : F007)

① Request frame

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Parameter Count | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010302010001D472 | 01 | 03 | 02 | 0C | 00 | 01 | 45 | B1 |

② Response frame (In case of F007 = 10.0sec)

| Response (7Byte) | Com. Num. | Com-mand | Byte Q'ty | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|
| 0103020064B9AF | 01 | 03 | 02 | 00 | 64 | B9 | AF |
| Data Description | $0x16^3 + 0x16^2 + 6x16^1 + 4x16^0 = 100$(Communication Data) ➔ divide by 10 ➔ **10.0sec (Parameter value)** | | | | | | |

## 2.4 Write command

User can change a value of function code using communication.

Refer to both **2.5 Frequency command** and **2.6 RUN command** about the Frequency and/or RUN command control.

### 2.4.1 Request frame

| 1Byte | 1Byte | 2Byte | 2Byte | 1Byte | 1Byte |
|---|---|---|---|---|---|
| **Comm. number** | **Command** | **Parameter** | **Data** | **CRC Hi** | **CRC Lo** |

| Clause | Description | Data size | Specifications |
|---|---|---|---|
| Communication number | Inverter communication number | 1 byte | 1~32, FF(Broadcasting) |
| Command | Frame type | 1 byte | 0x06 ➔ Write command |
| Parameter | Parameter | 2 byte | 1st byte : Group<br>D(0x01) F(0x02) U(0x03)<br>A(0x04) B(0x05) I(0x06)<br>O(0x07) C(0x08) H(0x09)<br>P(0x0a)<br>2nd byte : index (Index) |
| Data | Parameter Data | 2 byte | Setting value of Parameter |
| CRC Hi | - | 1 byte | Higher 8bit of 16bit CRC |
| CRC Lo | - | 1 byte | Lower 8bit of 16bit CRC |

※ Broadcasting can be used with command 06, and there is no response from inverter.

### 2.4.2 Response frame

| 1Byte | 1Byte | 2Byte | 2Byte | 1Byte | 1Byte |
|---|---|---|---|---|---|
| **Comm. number** | **Command** | **Parameter** | **Data** | **CRC Hi** | **CRC Lo** |

| Clause | Description | Data size | Specifications |
|---|---|---|---|
| Communication number | Inverter communication number | 1 byte | 1~32 |
| Command | Frame type | 1 byte | 0x06 ➔ Write command |
| Parameter | Parameter | 2 byte | 1st byte : Group<br>D(0x01) F(0x02) U(0x03)<br>A(0x04) B(0x05) I(0x06)<br>O(0x07) C(0x08) H(0x09)<br>P(0x0a)<br>2nd byte : index (Index) |
| Data | Parameter Data | 2 byte | Setting value of Parameter |
| CRC Hi | - | 1 byte | Higher 8bit of 16bit CRC |
| CRC Lo | - | 1 byte | Lower 8bit of 16bit CRC |

### 2.4.3 Transmit frame [Example]

(1) A prior condition

① Communication number : 01 (Inverter address is 1)

② Command : 06 (Write Request)

(2) Output frequency setting (Function code : F001, F201)

Refer to 2.5 Frequency command in details.

(3) Acceleration time setting (F007)

① Request frame (In case of 10sec)

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 0106020200642859 | 01 | 06 | 02 | 0C | 00 | 64 | 49 | 9A |
| Data Description | $0x16^3 + 0x16^2 + 6x16^1 + 4x16^0 = 100$ (Communication Data)<br>➔ divide by 10 ➔ **10.0sec (Parameter value)** | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 0106020200642859 | 01 | 06 | 02 | 0C | 00 | 64 | 49 | 9A |

(4) Deceleration time setting (F008)

① Request frame (In case of 30sec)

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 01060203012C783F | 01 | 06 | 02 | 0E | 01 | 2C | E9 | FC |
| Data Description | $0x16^3 + 1x16^2 + 2x16^1 + 12x16^0 = 300$ (Communication Data)<br>➔ divide by 10 ➔ **30.0sec (Parameter value)** | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 01060203012C783F | 01 | 06 | 02 | 0E | 01 | 2C | E9 | FC |

## 2.5 Frequency command

### 2.5.1 Frequency command source setting (F010)

Set F010 parameter for 3 in order to control the output frequency by RS232/RS485 communication.

(1) Function code

| Function code | Set item | Initial value | Data | Description |
|---|---|---|---|---|
| F010 | Frequency source setting | 0 | 0 | OPE VOL (Hex. value : 0000) |
| | | | 1 | Terminal (Hex. value : 0001) |
| | | | 2 | OPE Keypad (Hex. value: 0002) |
| | | | 3 | COM (Hex. Value : 0003) |
| | | | 4 | OPT1 (Hex. value : 0004) |
| | | | 5 | OPT2 (Hex. value : 0005) |

(2) Change F010 parameter to 3(COM) by RS232/RS485 communication.

① Request frame (In case of the inverter communication number = 1)

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 0106021100039876 | 01 | 06 | 02 | 11 | 00 | 03 | 98 | 76 |
| Data Description | 0003 ➔ **Change the frequency source setting to COM.** | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 0106021100039876 | 01 | 06 | 02 | 11 | 00 | 03 | 98 | 76 |

### 2.5.2 Transmit frame [Example]

(1) A prior condition

① Communication number : 01 (Inverter address is 1)

② Command : 06 (Write Request)

③ Parameter : 0004 (Special parameter for Frequency command)

(2) Frequency command (In case of 60Hz)

① Request frame

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600041770C61F | 01 | 06 | 00 | 04 | 17 | 70 | C6 | 1F |
| Data Description | $1x16^3 + 7x16^2 + 7x16^1 + 0x16^0 = 6000$ (Communication Data) ➔ divide by 100 ➔ **60.00Hz (Parameter value)** | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600041770C61F | 01 | 06 | 00 | 04 | 17 | 70 | C6 | 1F |

(3) Frequency command (In case of 50Hz)

① Request frame

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600041388C55D | 01 | 06 | 00 | 04 | 13 | 88 | C5 | 5D |
| Data Description | $1x16^3 + 3x16^2 + 8x16^1 + 8x16^0 = 5000$ (Communication Data) ➔ divide by 100 ➔ **50.00Hz (Parameter value)** | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600041388C55D | 01 | 06 | 00 | 04 | 13 | 88 | C5 | 5D |

## 2.6 RUN command

### 2.6.1 RUN command source setting (F011)

Set F011 parameter for 3 in order to control inverter RUN/STOP/RST by RS232/RS485 communication.

(1) Function code

| Function code | Set item | Initial value | Data | Description |
|---|---|---|---|---|
| F011 | RUN command source setting | 0 | 1 | Terminal (Hex. value : 0001) |
| | | | 2 | OPE (Hex. value : 0002) |
| | | | 3 | COM (Hex. value : 0003) |
| | | | 4 | OPT1 (Hex. value : 0004) |
| | | | 5 | OPT2 (Hex. value : 0005) |

(2) Change F011 parameter to 3(COM) by RS232/RS485 communication.

① Request frame (In case of the inverter communication number = 1)

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 0106021100039876 | 01 | 06 | 02 | 12 | 00 | 03 | 68 | 76 |
| Data Description | 0002 ➔ **Change the RUN command source setting to COM.** | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 0106021100039876 | 01 | 06 | 02 | 12 | 00 | 03 | 68 | 76 |

### 2.6.2 Transmit frame [Example]

(1) A prior condition

① Communication number : 01 (Inverter address is 1)

② Command : 06 (Write Request)

③ Parameter : 0002 (Special parameter for RUN command)

(2) STOP command

① Request frame

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600020000280A | 01 | 06 | 00 | 02 | 00 | 00 | 28 | 0A |
| Data Description | 0000 : STOP 0001 : FWD, 0002 : REV, ,0003 : Cutoff 0004 : RST | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600020000280A | 01 | 06 | 00 | 02 | 00 | 00 | 28 | 0A |

(3) FWD(Forward) RUN command

① Request frame

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600020001E9CA | 01 | 06 | 00 | 02 | 00 | 01 | E9 | CA |
| Data Description | 0000 : STOP 0001 : FWD, 0002 : REV, ,0003 : Cutoff 0004 : RST | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600020001E9CA | 01 | 06 | 00 | 02 | 00 | 01 | E9 | CA |

(4) REV(Reverse) RUN command

① Request frame

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600020002A9CB | 01 | 06 | 00 | 02 | 00 | 02 | A9 | CB |
| Data Description | 0000 : STOP 0001 : FWD, 0002 : REV, ,0003 : Cutoff 0004 : RST | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600020002A9CB | 01 | 06 | 00 | 02 | 00 | 02 | A9 | CB |

(5) Cut-off command for output

① Request frame

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600020003680B | 01 | 06 | 00 | 02 | 00 | 03 | 68 | 0B |
| Data Description | 0000 : STOP 0001 : FWD, 0002 : REV, ,0003 : Cutoff 0004 : RST | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 010600020003680B | 01 | 06 | 00 | 02 | 00 | 03 | 68 | 0B |

(6) RST(Reset) command

① Request frame

| Transmit (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 01060002000429C9 | 01 | 06 | 00 | 02 | 00 | 04 | 29 | C9 |
| Data Description | 0000 : STOP 0001 : FWD, 0002 : REV, ,0003 : Cutoff 0004 : RST | | | | | | | |

② Response frame

| Response (8Byte) | Com. Num. | Com-mand | Parameter | | Data | | CRC 16 Hi | CRC 16 Lo |
|---|---|---|---|---|---|---|---|---|
| 01060002000429C9 | 01 | 06 | 00 | 02 | 00 | 04 | 29 | C9 |

## 2.7 CRC-16 generation

### 2.7.1 CRC-16 generation procedure

(1) All of 16-bit register is 1. 0xFFFF

(2) The exclusive OR of 16-bit register and 8-bit register.

(3) Shift right side 1bit a 16-bit register.

(4) If the result of step 3 is 1, exclusive OR 16-bit register and 0xa001.

(5) Execute 8 times step 3 and step 4.

(6) Execute step 2 ~ 6 until data completion

(7) Exchange the step 6 result of higher 8bit and lower 8bit.

### 2.7.2 Example : The case of D001 output frequency reading.

| Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 |
|---|---|---|---|---|---|
| Communication code | Command | Parameter | | Parameter count | |
| **0x01** | **0x03** | **0x01** | **0x01** | **0x00** | **0x01** |

## CRC-16 Generation

**1. Transmit frame(1~6 Byte)** **2. Generation**

| | Hex. | Binary | CRC-16 |
|---|---|---|---|
| Byte | Data value | Data value | Value |
| Byte 1 | 0x01 | 00000001 | 0x807e |
| Byte 2 | 0x03 | 00000011 | 0x2140 |
| Byte 3 | 0x01 | 00000001 | 0x30e1 |
| Byte 4 | 0x01 | 00000001 | 0x8831 |
| Byte 5 | 0x00 | 00000000 | 0xd449 |
| Byte 6 | 0x01 | 00000001 | 0x36d4 |

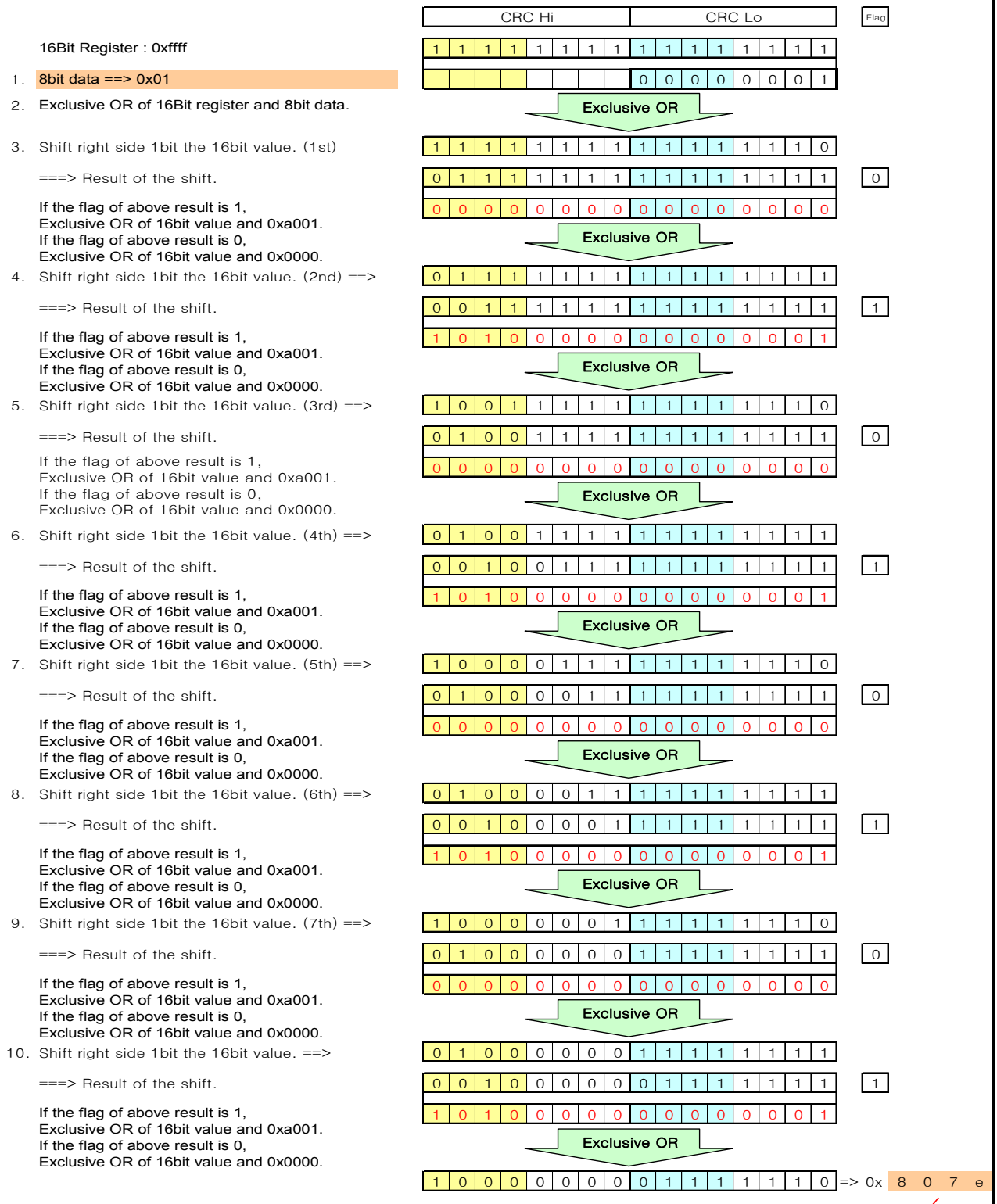
Binary Conversion → CRC Conversion →

**3. Exchange the result of higher 8bit and lower 8bit.**

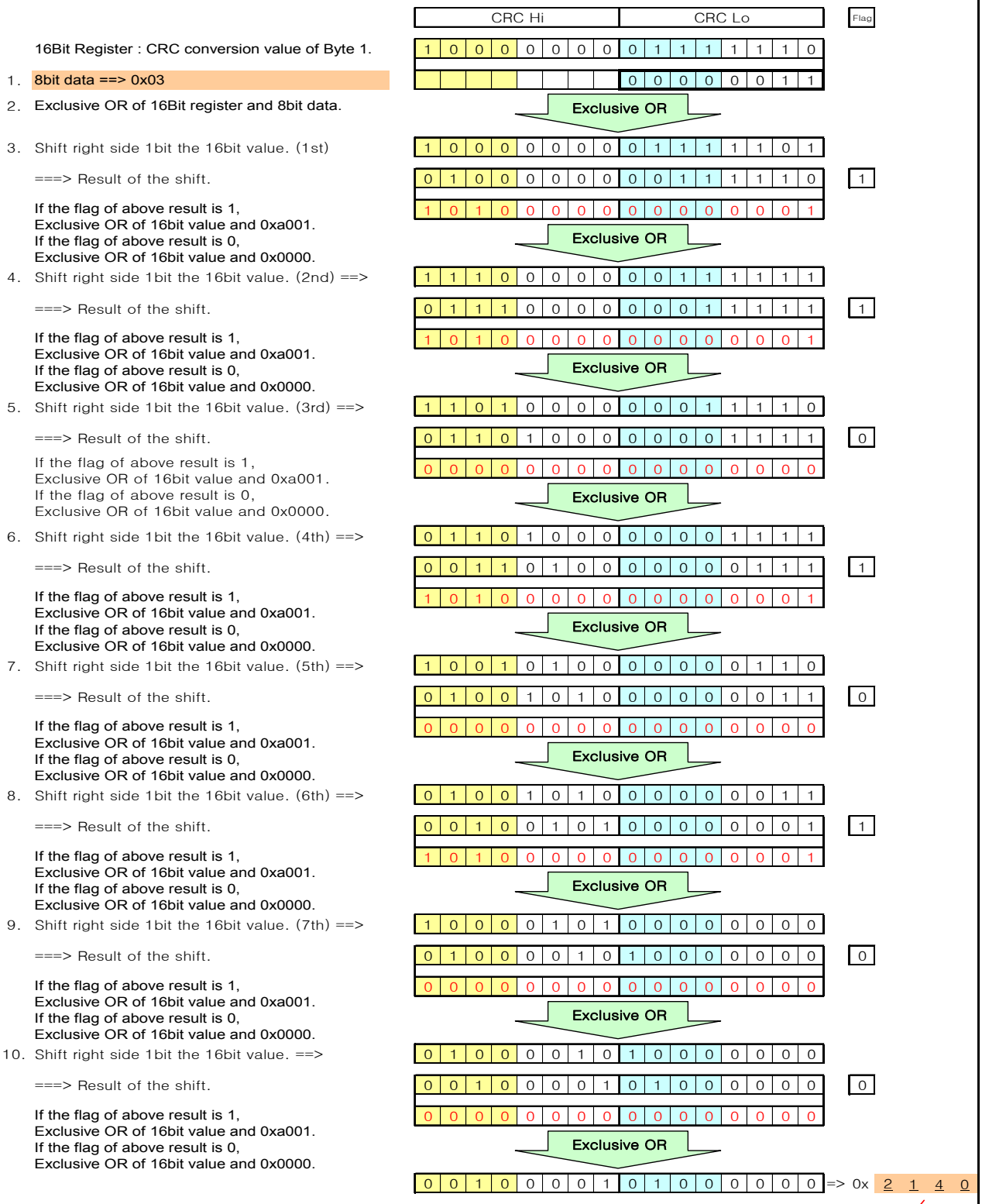
**0x36d4** → **0xd436**

**4. Byte 7 = 0xd4**
**Byte 8 = 0x36**

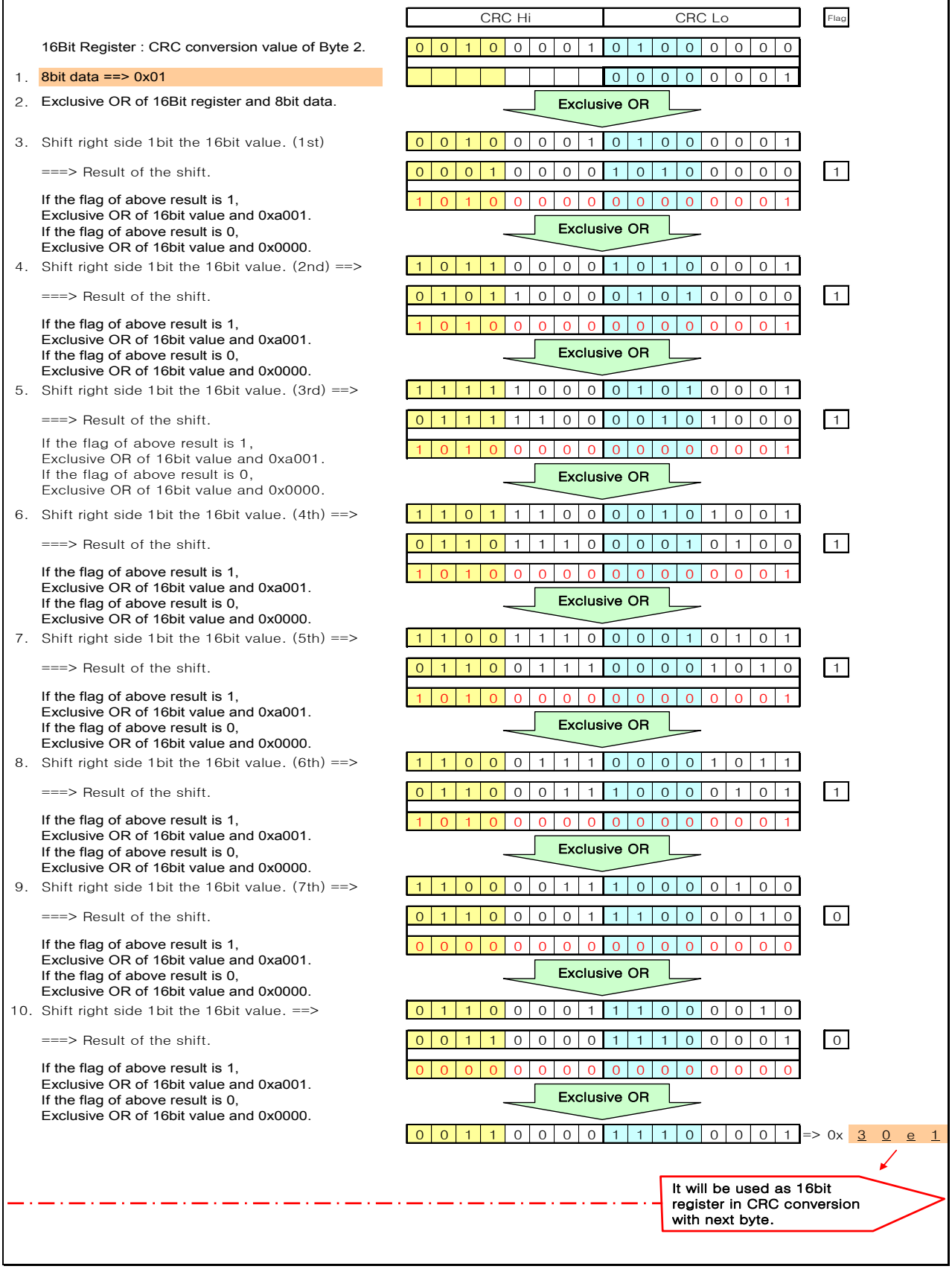
## CRC-16 Generation Procedure (1/6 step)



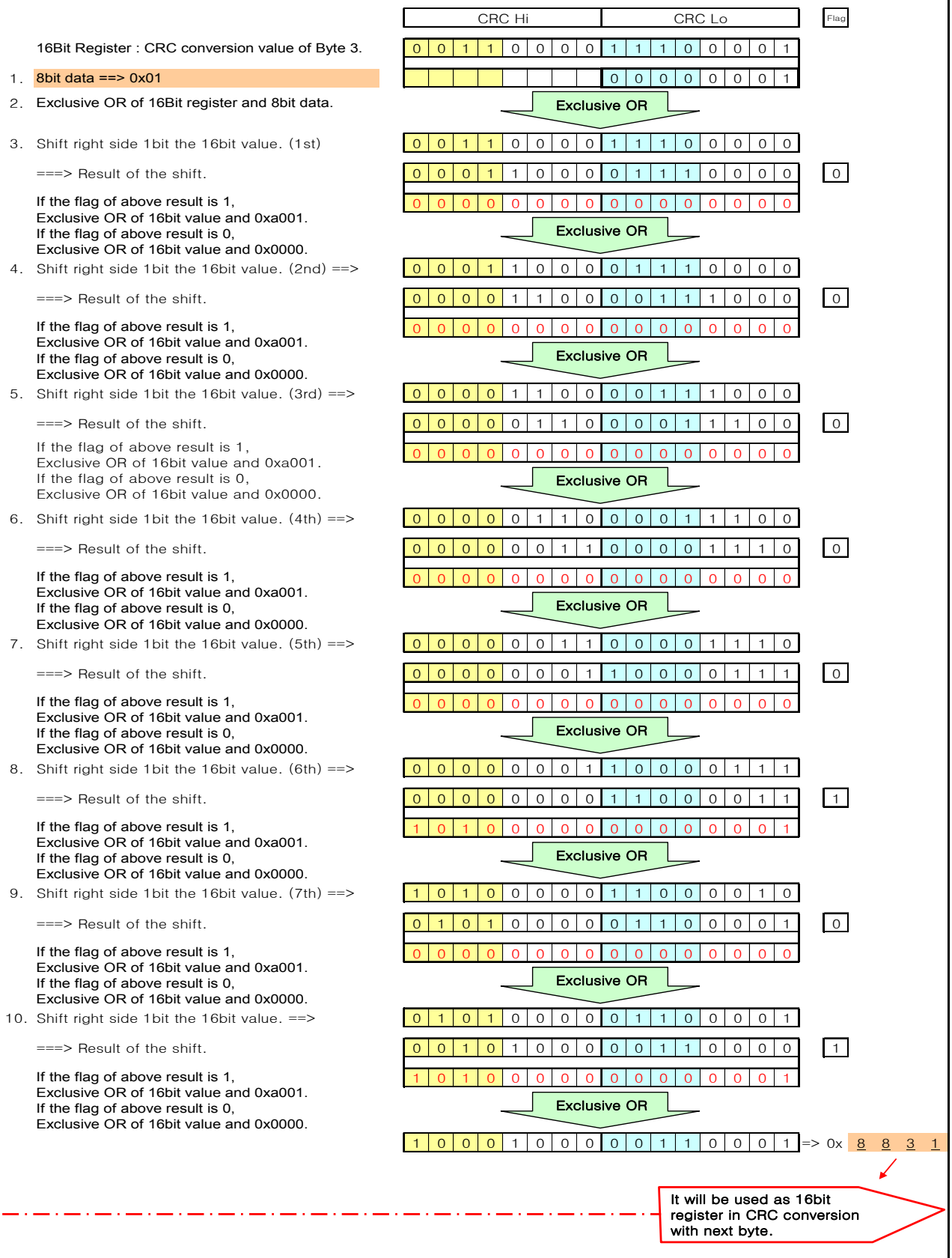
## CRC-16 Generation Procedure (2/6 step)



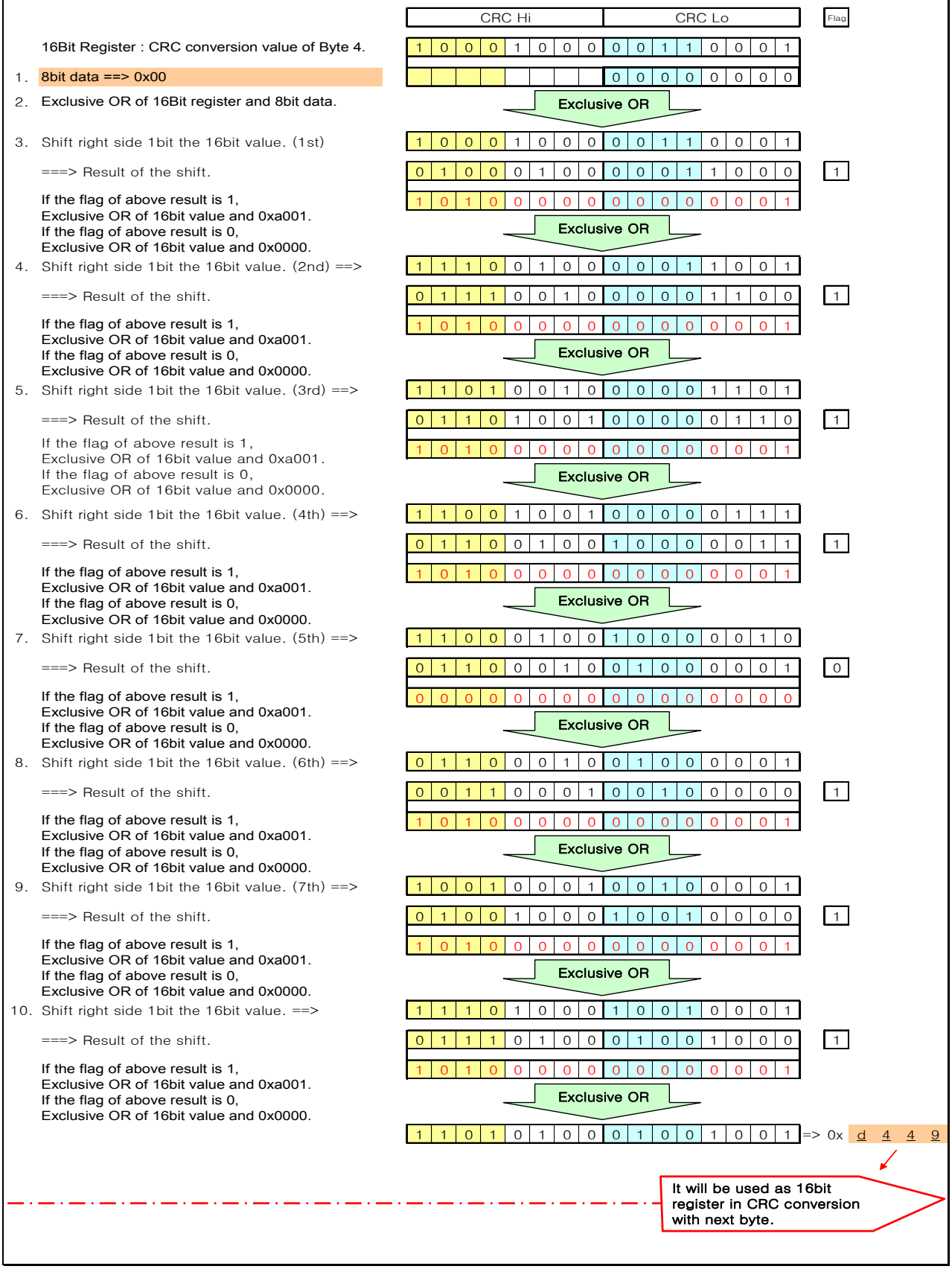
## CRC-16 Generation Procedure (3/6 step)



## CRC-16 Generation Procedure (4/6 step)



## CRC-16 Generation Procedure (5/6 step)



## CRC-16 Generation Procedure (6/6 step)