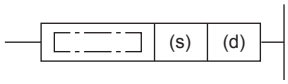
Converting 32-bit binary data to decimal ASCII

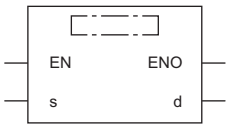
DBINDA(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert 32-bit binary data specified by (s) into decimal ASCII codes, and store the converted data in the device specified by (d) and later.

Ladder diagram	Structured text	
	ENO:=DBINDA(EN,s,d); ENO:=DBINDAP(EN,s,d);	ENO:=DBINDA_U(EN,s,d); ENO:=DBINDAP_U(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)	DBINDA(P)	Binary data to be converted into ASCII codes	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DBINDA(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)		Head device number storing conversion result	—	Character string	ANYSTRING_SINGLE
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions convert 32-bit binary data specified by (s) into decimal ASCII codes, and store the converted data in the device specified by (d) and later.
- The format of the decimal ASCII data to be stored in (d) depends on the status of SM705 (Number of conversion digits selection).

Status of SM705*1	Storage format of (d)	Reference
OFF	Data is stored in a fixed number of digits (a sign + 10 digits).	Page 607 Operation of when SM705 (Number of conversion digits selection) is off
ON	Each digit is stored left-justified depending on the value of (s).	Page 608 Operation of when SM705 (Number of conversion digits selection) is on

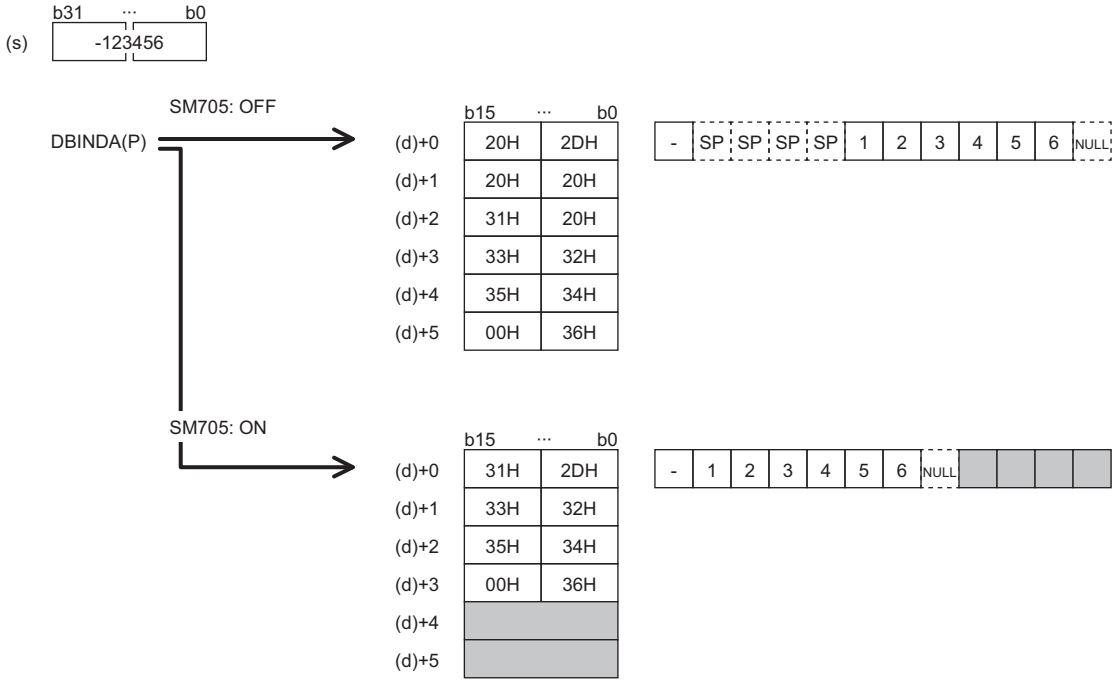
*1 For the firmware version of the CPU module supporting SM705, refer to the following.
MELSEC iQ-F FX5 User's Manual (Application)
A CPU module which does not support SM705 operates in the same way as SM705 is off even if it is turned on.

■Operation overview

The following figure shows the operation when SM705 (Number of conversion digits selection) is off and on.

Ex.

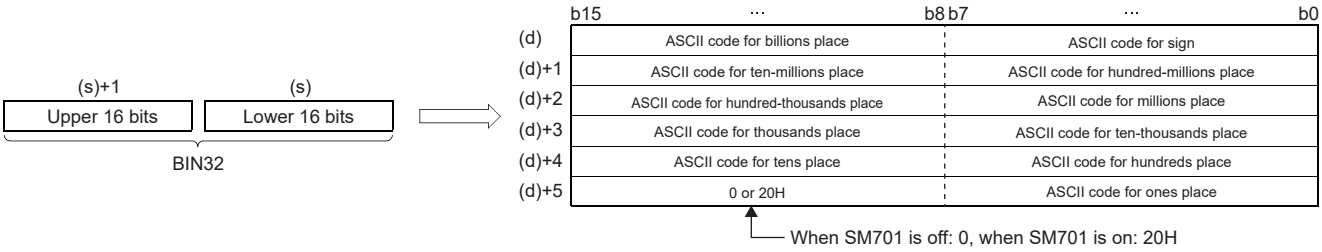
When the DBINDA(P) instruction is executed with a numeric value "-123456" stored in (s)



- When SM705 is off, the number of digits is fixed. The first character is a sign and it is 2DH(-) in the above example. (If (s) is 0 or positive, the first character is 20H (space).) The numeric part is right-justified. When the length of the numeric part is less than 10 digits, the ASCII code 20H (space) is stored for the ASCII code of the upper digit(s).
- When SM705 is on, data is left-justified. When the length of the numeric part is less than 10 digits, 00H is stored in the end.

■Operation of when SM705 (Number of conversion digits selection) is off

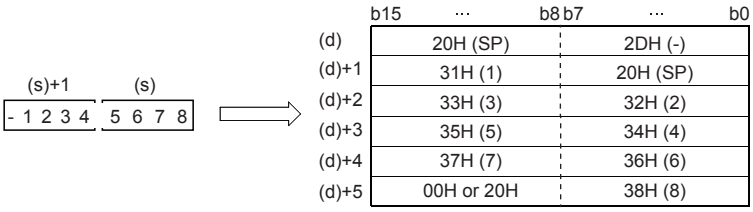
Decimal ASCII data is stored in a fixed number of digits in (d) to (d)+5.



- The following shows the operation result to be stored in (d).
  - As sign data, "20H" is stored if the binary data is positive, and "2DH" is stored if the data is negative.
  - "20H" is stored for "0" on the left side of the valid digits (zero suppression). For "0012034560", 20H is stored for "00", and the number of digits is 8 based on "12034560".
  - In the upper 8 bits of the device specified by (d)+5, 0 is stored when SM701 (Output characters selection) is off, and 20H is stored when SM701 is on.

Ex.

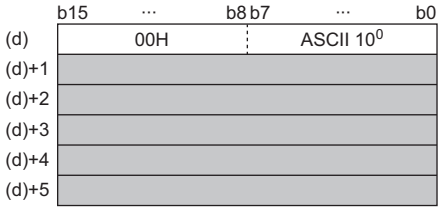
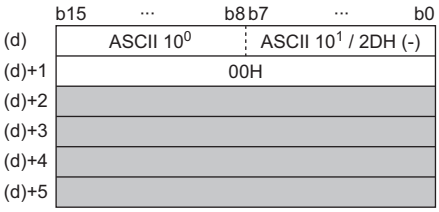
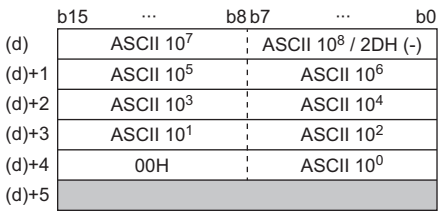
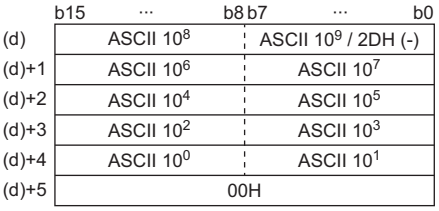
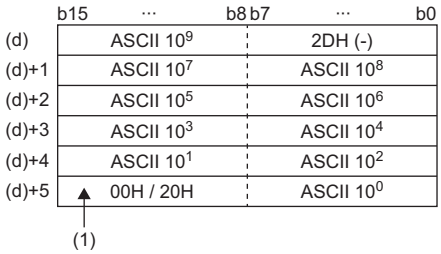
When -12345678 is specified in (s) (when signed data is specified)



■Operation of when SM705 (Number of conversion digits selection) is on

Decimal ASCII data is stored right-justified in (d).

The following figures show an example of a value of (s) and a value stored in (d).

<table>
<tr><th>Value of (s)</th><th>Data of (d) to (d)+5</th><th>Value of (s)</th><th>Data of (d) to (d)+5</th></tr>
<tr><td>• 0
• Positive value (1 digit in numeric part)</td><td>• The upper byte of (d) is filled with 00H.
• (d)+1 and later remain unchanged.
 
</td><td>• Positive value (2 digits in numeric part)
• Negative value (1 digit in numeric part)</td><td>• (d)+1 is filled with 00H.
• (d)+2 and later remain unchanged.
 
</td></tr>
<tr><td colspan="4">⋮</td></tr>
<tr><td>• Positive value (9 digits in numeric part)
• Negative value (8 digits in numeric part)</td><td>• The upper byte of (d)+4 is filled with 00H.
• (d)+5 and later remain unchanged.
 
</td><td>• Positive value (10 digits in numeric part)
• Negative value (9 digits in numeric part)</td><td>• (d)+5 is filled with 00H.
 
</td></tr>
<tr><td>• Negative value (10 digits in numeric part)</td><td>• 00H is stored in the upper byte (1) of (d)+5 when SM701 (Output characters selection) is off, and 20H (space) is stored when it is on.
 
</td><td colspan="2">ASCII 10^9: ASCII code for the billions place
ASCII 10^8: ASCII code for the hundred-millions place
⋮
ASCII 10^1: ASCII code for the tens place
ASCII 10^0: ASCII code for the ones place</td></tr>
</table>

- When the number of operation digits is less than the maximum number of digits (a sign + 10 digits), 00H is stored in the end of the string regardless of the status (on/off) of SM701 (Output characters selection). If the end of the string is the lower byte, 00H is also stored in the upper byte.
- When the number of operation digits is equal to the maximum number of digits (a sign + 10 digits), 00H is stored in the upper byte of (d)+5 if SM701 is off. 20H (space) is stored in the upper byte of (d)+5 if SM701 is on.

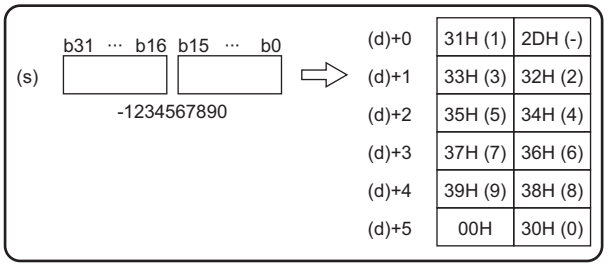
Precautions

- (d) occupies six points.
- When SM705 (Number of conversion digits selection) is on, the operation result is stored in (d) for the effective number of digits. Therefore, when the DBINDA(P)(_U) instruction is executed successively and the operation result for each execution is stored in the same device, a part of the previous operation result may not be overwritten by the succeeding result and can remain in (d).

Ex.

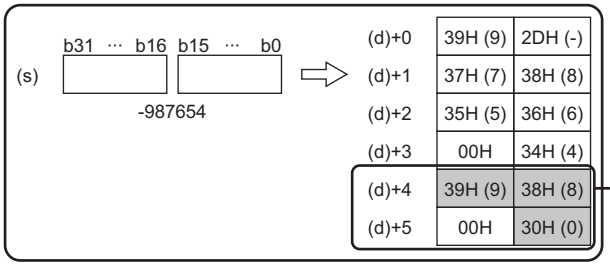
Executing the DBINDA(P) instruction when (s) is "-1234567890" and then executing another DBINDA(P) instruction when (s) is "-987654"

(1)



(d)+0	31H (1)	2DH (-)
(d)+1	33H (3)	32H (2)
(d)+2	35H (5)	34H (4)
(d)+3	37H (7)	36H (6)
(d)+4	39H (9)	38H (8)
(d)+5	00H	30H (0)

(2)



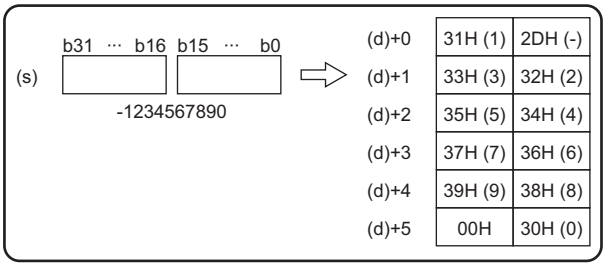
(d)+0	39H (9)	2DH (-)
(d)+1	37H (7)	38H (8)
(d)+2	35H (5)	36H (6)
(d)+3	00H	34H (4)
(d)+4	39H (9)	38H (8)
(d)+5	00H	30H (0)

(3)

(1) "-1234567890" is converted into a string.
(2) "-987654" is converted into a string.
(3) A part of the previous conversion result remains in (d)+4 and (d)+5.

To avoid this, create a program to clear the entire data storage areas (d)+0 to (d)+5 before executing the DBINDA(P)(_U) instruction.

(4)

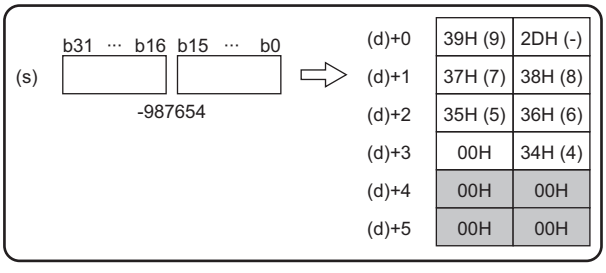


(d)+0	31H (1)	2DH (-)
(d)+1	33H (3)	32H (2)
(d)+2	35H (5)	34H (4)
(d)+3	37H (7)	36H (6)
(d)+4	39H (9)	38H (8)
(d)+5	00H	30H (0)

(5)

(d)+0	00H	00H
(d)+1	00H	00H
(d)+2	00H	00H
(d)+3	00H	00H
(d)+4	00H	00H
(d)+5	00H	00H

(6)



(d)+0	39H (9)	2DH (-)
(d)+1	37H (7)	38H (8)
(d)+2	35H (5)	36H (6)
(d)+3	00H	34H (4)
(d)+4	00H	00H
(d)+5	00H	00H

(4) "-1234567890" is converted into a string.
(5) (d)+0 to (d)+5 are cleared.
(6) "-987654" is converted into a string.

Operation error

Error code (SD0/SD8067)	Description
2820H	The device specified by (d) exceeds the corresponding device range.