M65762FP

QM-CODER

DESCRIPTION

The M65762FP is a compression and decompression LSI conforming to the high efficiency encoding system (QM-Coder) in the International Standard, the JBIG/JPEG (ITU-T Recommendations T.81 and T.82) for coding still images. It also conforms to the International Standard (ITU-T Recommendation T.85) for facsimile. The QM-Coder is an information dependent type which is capable of completely restoring original image data, and is equipped with the learning function to always optimize parameters according to the statistical characteristics of images. The QM-Coder is therefore superior in compression ratio compared with the existing binary coding system (MH/MR/MMR) and can greatly improve the half toning image (dithered half toning image) whose compression ratio is especially poor.

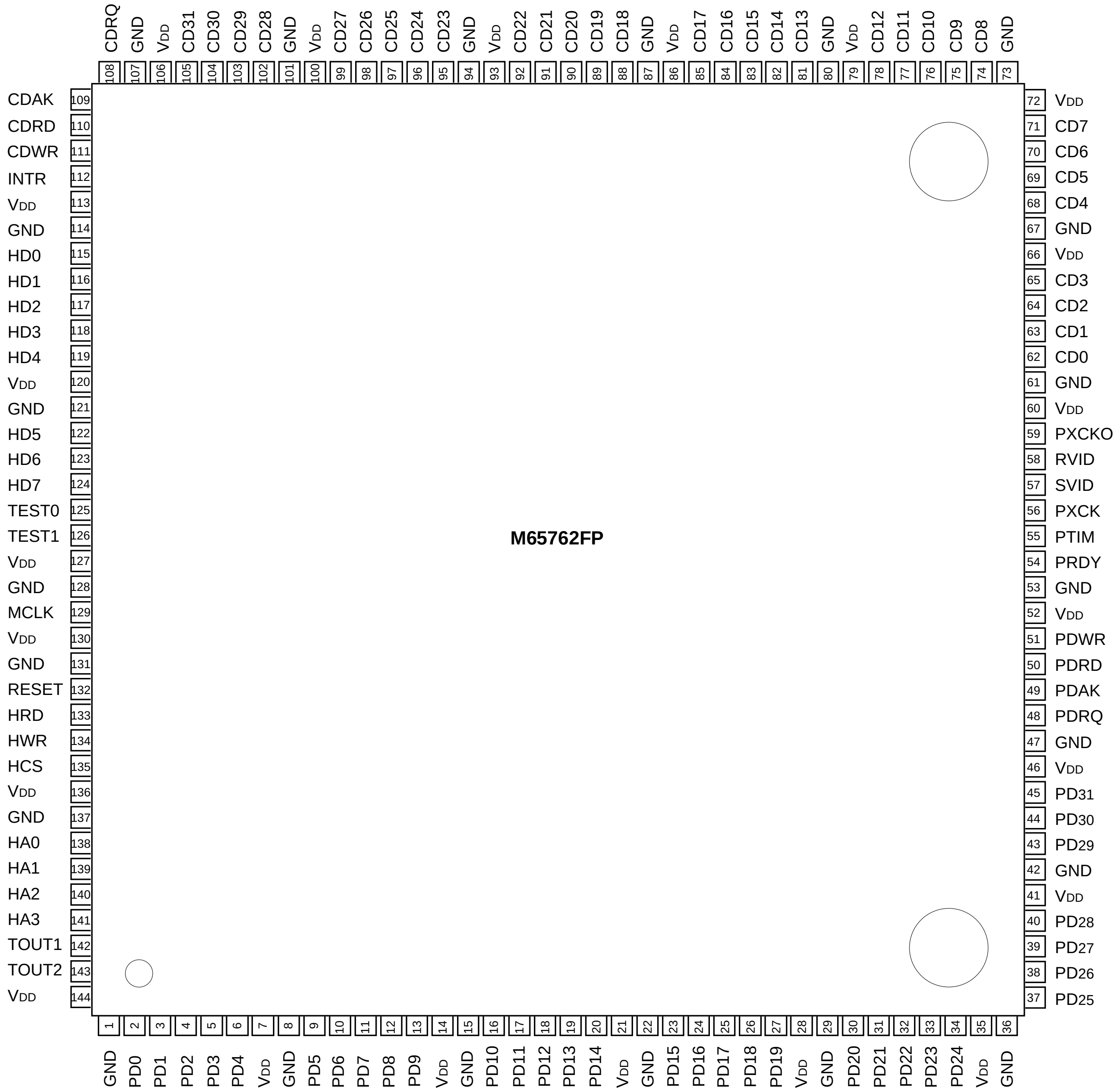
FEATURES

- Completely conforms to the International Standard (ITU-T T.85) for facsimile.
- Achieves encoding/decoding with the arithmetic coder (QM-Coder) conforming to the recommendation of the International Standard JBIG/JPEG.
- Is expected to conform to the International Standard for color facsimile (T.Pallete-colour).
- High speed processing that puts into effect coding and decoding at 40 million pixels per sec maximum.
- Is possible data-through processing without coding and decodin.
- Can select context
  - Provides 10 pixel template model for minimum resolution conforming to JBIG and can select 2-line or 3-line template model.
- Built-in typical prediction function
  - Capable of coding and decoding by using the typical prediction.
  - Since use of the typical prediction does not require the processing of the line (TP line) which is matched the previous line's data, is capable of reducing data and processing time.
- Built-in adaptive template (AT) function
  - Is capable of setting AT pixels before 127 pixels on the coding line.
  - Since It is possible to change the position of AT pixel in a specified line, is capable of improving compression characteristics even when image characteristic is changed in the middle of the screen.
- Supporting multi-stripe
  - When a page consists of more than one stripe, is capable of repeating encoding/decoding process in stripes.
- Built-in load/store function of line memory → Supporting multiple planes and multi-stripe function
  - Is capable of loading image data for reference line from outside to line memory of the LSI and storing image data from line memory to outside.
- Number of processing lines
  - Is capable of issuing the start of processing (temporary stop command) several times to encode/decode any lines more than or equal to 65535 lines.
- Supporting 3-bus interface
  - An 8-bit host bus corresponds to the MPU is available to load and store of context table RAM.
  - For input/output of binary image data, is capable of performing 32-bit or 16-bit parallel or serial input/output.
  - For input/output of coding data, is capable of selecting 32-bit/16-bit/8-bit bus to perform DMA transfer of coding data.
- Is capable of making scale-down for coding and scale-up for decoding.
- Is capable of setting marker code for coding and detecting marker code for decoding.
- Built-in RAM for 4096 bytes for line memory, built-in context table RAM and built-in probability estimation table ROM of 113 status
- +5V single power supply

APPLICATION

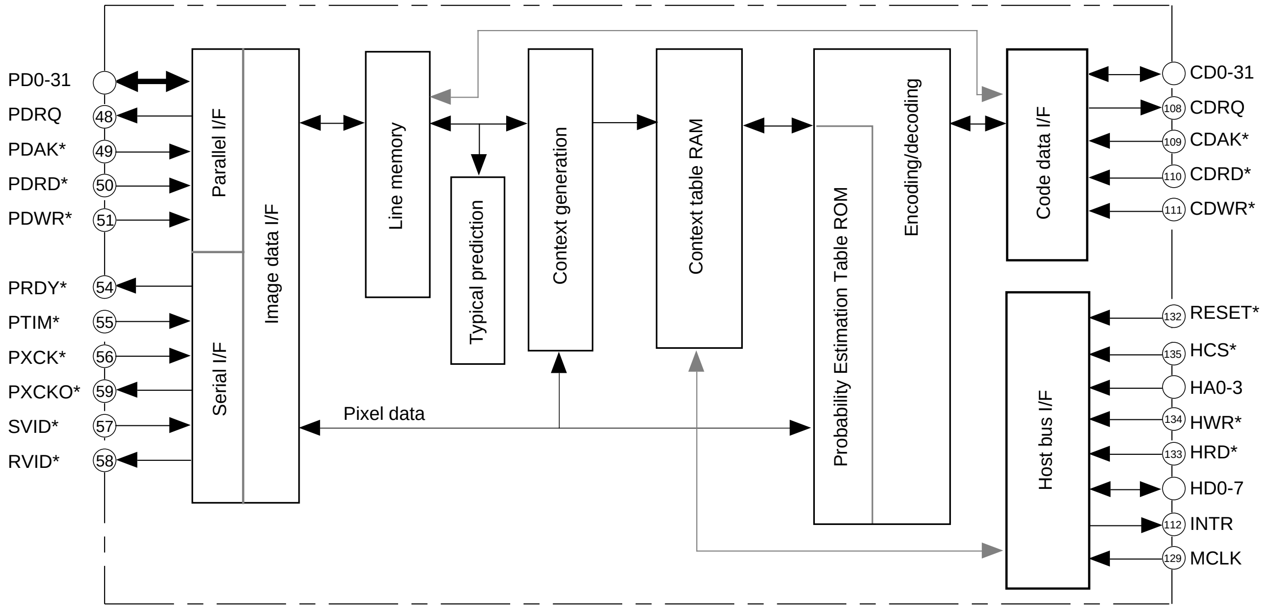
- OA equipment including facsimile, copier and printer
- Digital and amusement equipment for the purpose of reducing memory

PIN CONFIGURATION (TOP VIEW)



Outline144P6Q-A

BLOCK DIAGRAM



(Asterisk "*" indicates negative logic.)

Description on Block Functions

(1) Host bus I/F block
This bus is used to set command parameters and load the status between the MPU and this block. It is 8-bit bus, This block is also available to load and store of context table RAM via the host bus.

(2) Code data I/F block
Bus for input/output of coding data. For the bus width, 32-bits, 16-bits or 8-bits can be selected.
Image data can also be transferred (in through mode) between the Image data I/F and this block via built-in line memory. FIFO buffer for 16 bytes are provided in the code data I/F block.

(3) Image data I/F block
The Image data I/F is used for input/output of binary image data. The 32-/16-bit parallel I/F or serial I/F can be selected. Selection of the serial I/F transfers data in units of 1 pixel in synchronization with the line, using the handshake signal (PRDY*, PTIM*).
Selection of parallel I/F uses an external DMA controller for DMA transfer (in units of stripe).
The image data I/F provides a function for scale-down of length and breadth by 1/2 in coding and a function for scale-up of length and breadth by twice in decoding.

(4) Line memory block
4K-byte memory. This block can be set to a maximum of 8192 pixels/line for 3-line template and can be set to a maximum of 10240 pixels/line for 2-line template. A line is used for input/output processing of image data to/from outside and the other lines (2 or 3 lines) are used for encoding/decoding processing. These two processes can be independently carried out in synchronization with each line.
The contents of line memory can be loaded or stored via the image data I/F or coding data I/F.

(5) Typical prediction block
In the typical prediction mode, compares the encoding/decoding process line agree with the immediately preceding line and generates pseudo-pixel (SLNTP).

(6) Context generator
By using the 10 pixel template of 2-lines or 3-lines.(including AT pixel) the standard context minimum of JBIG is generated with the resolution.

(7) Context table RAM block
Corresponds to the 10-bit standard context. This block can initialize, load and store the context table RAM.

(8) Coding/decoding block
This block performs arithmetic coding and decoding.
It contains a ROM which contains a table capable of estimating 113 states and is capable of byte stuffing function ('OO' byte insertion/rejection) and is capable of end marker code control (Marker insertion/detection).

DESCRIPTION PIN

Pin No.	I/O	Pin name
1	Power supply	GND
2	I/O	PD0
3	I/O	PD1
4	I/O	PD2
5	I/O	PD3
6	I/O	PD4
7	Power supply	VDD
8	Power supply	GND
9	I/O	PD5
10	I/O	PD6
11	I/O	PD7
12	I/O	PD8
13	I/O	PD9
14	Power supply	VDD
15	Power supply	GND
16	I/O	PD10
17	I/O	PD11
18	I/O	PD12
19	I/O	PD13
20	I/O	PD14
21	Power supply	VDD
22	Power supply	GND
23	I/O	PD15
24	I/O	PD16
25	I/O	PD17
26	I/O	PD18
27	I/O	PD19
28	Power supply	VDD
29	Power supply	GND
30	I/O	PD20
31	I/O	PD21
32	I/O	PD22
33	I/O	PD23
34	I/O	PD24
35	Power supply	VDD
36	Power supply	GND
37	I/O	PD25
38	I/O	PD26
39	I/O	PD27
40	I/O	PD28
41	Power supply	VDD
42	Power supply	GND
43	I/O	PD29
44	I/O	PD30
45	I/O	PD31
46	Power supply	VDD
47	Power supply	GND
48	O	PDRQ
49	I	PDAK
50	I	PDRD

Pin No.	I/O	Pin name
51	I	PDWR
52	Power supply	VDD
53	Power supply	GND
54	O	PRDY
55	I	PTIM
56	I	PXCK
57	I	SVID
58	O	RVID
59	O	PXCKO
60	Power supply	VDD
61	Power supply	GND
62	I/O	CD0
63	I/O	CD1
64	I/O	CD2
65	I/O	CD3
66	Power supply	VDD
67	Power supply	GND
68	I/O	CD4
69	I/O	CD5
70	I/O	CD6
71	I/O	CD7
72	Power supply	VDD
73	Power supply	GND
74	I/O	CD8
75	I/O	CD9
76	I/O	CD10
77	I/O	CD11
78	I/O	CD12
79	Power supply	VDD
80	Power supply	GND
81	I/O	CD13
82	I/O	CD14
83	I/O	CD15
84	I/O	CD16
85	I/O	CD17
86	Power supply	VDD
87	Power supply	GND
88	I/O	CD18
89	I/O	CD19
90	I/O	CD20
91	I/O	CD21
92	I/O	CD22
93	Power supply	VDD
94	Power supply	GND
95	I/O	CD23
96	I/O	CD24
97	I/O	CD25
98	I/O	CD26
99	I/O	CD27
100	Power supply	VDD

Pin No.	I/O	Pin name
101	Power supply	GND
102	I/O	CD28
103	I/O	CD29
104	I/O	CD30
105	I/O	CD31
106	Power supply	VDD
107	Power supply	GND
108	O	CDRQ
109	I	CDAK
110	I	CDRD
111	I	CDWR
112	O	INTR
113	Power supply	VDD
114	Power supply	GND
115	I/O	HD0
116	I/O	HD1
117	I/O	HD2
118	I/O	HD3
119	I/O	HD4
120	Power supply	VDD
121	Power supply	GND
122	I/O	HD5
123	I/O	HD6
124	I/O	HD7
125	I	TEST0
126	I	TEST1
127	Power supply	VDD
128	Power supply	GND
129	I	MCLK
130	Power supply	VDD
131	Power supply	GND
132	I	RESET
133	I	HRD
134	I	HWR
135	I	HCS
136	Power supply	VDD
137	Power supply	GND
138	I	HA0
139	I	HA1
140	I	HA2
141	I	HA3
142	O	TOUT1
143	O	TOUT2
144	Power supply	VDD

(Notes)
- Directly connect the input pin having pull-up (see Section 3.3.2 "Pin Function") to Vcc when the pin is not used.
- Directly connect the input pin having pull-down (see Section 3.3.2 "Pin Function" to GND when the pin is not used.
- Connect test input pin TEST 0/1 to GND.
- Leave test output pin TOUT 1/2 open.

Description on Pin Functions

(Asterisk "*" in signal name indicates negative logic.)

I/F		Pin name	I/O	BUF	Function
Host bus I/F		RESET*	I	S	H/W reset signal
		HCS*	I		Chip select signal
		HA0-3	I		Address select signal of internal register
		HWR*	I	S	Write strobe signal
		HRD*	I	S	Read strobe signal
		HD0-7	I	R8	Input/output data bus signal
		INTR	O	4	Interrupt request signal
Code data I/F		CD0-31	I/O	UR8	Coding data input/output bus signal
					(CD0-15 is used in 16-bit bus and CD0-7 is used in 8-bit bus.)
		CDRQ	O	4	DMA request signal for coding data (image data)
		CDAK*	I	US	DMA acknowledge signal for coding data (image data)
		CDRD*	I	US	Read strobe signal for coding data (image data)
		CDWR	I	US	Write strobe signal for coding data (image data)
Image data I/F	Parallel	PD0-31	I/O	UR8	Parallel image data input/output bus (PD0-15 is used in 16-bit bus.)
		PDRQ	O	4	DMA request signal for image data
		PDAK*	I	US	DMA acknowledge signal for image data
		PDRD*	I	US	Read strobe signal for image data
		PDWR*	I	US	Strobe signal for image data
	Serial	PRDY*	O	4	1-line input/output start ready signal for image data
		PTIM*	I	US	1-line transfer sector signal for image data
		PXCK*	I	US	Transfer clock signal for image data
		PXCKO*	O	4	Transfer clock signal for image data (LSI internal loopback output signal of PXCK*)
		SVID*	I	U	Image data input signal
		RVID*	O	4	Image data output signal
Others		MCLK	I		Master clock input signal
		TEST0, 1	I	DS	Test input signal 0/1 (Should be connected to GND when used normally.)
		V_{DD}	–	–	Power supply (+5V)
		GND	–	–	Ground

- Input buffer for the input pins ("I" and "IO") are set at the TTL level and the options are as follows.
  (U: Having pull-up resistance, D: Having pull-down resistance, S: Schmitt trigger, R: Through rate control)
- Numbers (4, 8) in the BUF column for the output pins ('O' and 'IO') indicate Io (= 4 or 8 mA).

Specifications

(1) Package
Plastic QFP 144 pins (20 mm*20 mm)

(2) Power consumption
5V 120mA (600mW)

(3) Maximum clock frequency
40MHz

Specifications of Coding Functions

(1) Coding algorithm
- QM-Coder (JBIG standard arithmetic coding system)

(2) Context

a) Template model
- 2- or 3-line of 10 pixel template (See Figure 1.)
  (Conforming to the template for JBIG minimum resolution)
  (Note) The coding efficiency of the 3-line template is better than that of 2-line template by several %.

b) Adaptive template (AT)
- It is possible to move up to 127 pixels on the coding line.
  (AT position is indicated by MPU.)
  (Note) AT is available to improve the coding efficiency for dither image.
- Even in the middle of coding/decoding , the position of AT line can be changed for a line (ATmove)
  (Note) When the position the AT pixel of is changed, the template model cannot be changed concurrently.

(3) Typical Prediction
- Agreement with the typical prediction of the minimum resolution of JBIG.
  The psedo-pixel (SLNTP) is generated by the symbol LNTP which shows whether the coding/decoding process lines agree with the immediately preceding line. If they agree, the pesudo-pixel only is coded. This makes it possible to shorten the time of process and rejection of the code data.
  SLNTPy = !(LNTPy⊕LNTPy-1) (where: y indicates a line No., y = 1 indicates that lines do not match each other, and initial value LNTP for head line is given with y - 1 = 1)

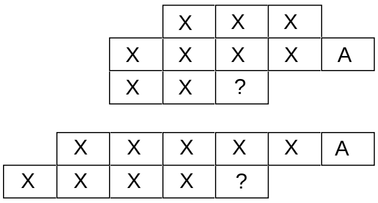


Figure 1 Template (X, A)
(Upper: 3 lines, Lower: 2 lines)

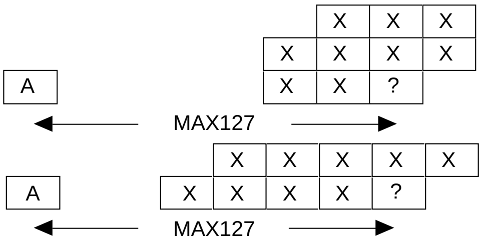


Figure 2. Adaptive Template (A)

(4) Coding data format
- The stripe data entity (SDE = stripe coded data with byte stuffing (PSCD) + end marker (SDNORM/SDRST)). Performs coding and decoding of one stripe (See Attached Figure A.1.)
  In the case of multi-striped (multi-stripes), can be supported by activation for each stripe.

(5) Marker code
- Supports the SDE end marker (During coding, the marker code previously set in the register is outputted. During decoding, the marker code byte detected by requesting on interrupt to MPU when the maker is detected is read out of the register.)

(6) Estimation of coding/decoding speed

Figure 3 compares the estimation of coding/decoding speed between the M65762FP and the existing product type (M65760/1FP). Polygonal lines in the diagram are processing speeds of images theoretically generated assuming the unmatched estimation ratio as a parameter. In addition, , ○□ △ indicate processing speeds of real image (without TP function).

As shown in this diagram, the M65762FP has been largely improved in the processing speed compared with existing product types. If the compression ratio is reduced, the reduction ratio of processing speed is moderated.

When a theoretical image is used to compare processing speeds in the worst case, the processing speed of existing product type is about 9.4M pixels/sec (1/compression ratio≒is about 1), while the processing speed of the M65762FP is about 27.5M pixels/sec (1/compression ratio ≒0.9) for coding and is about 31.2M pixels/sec (1/compression ratio≒0.75) for decoding.

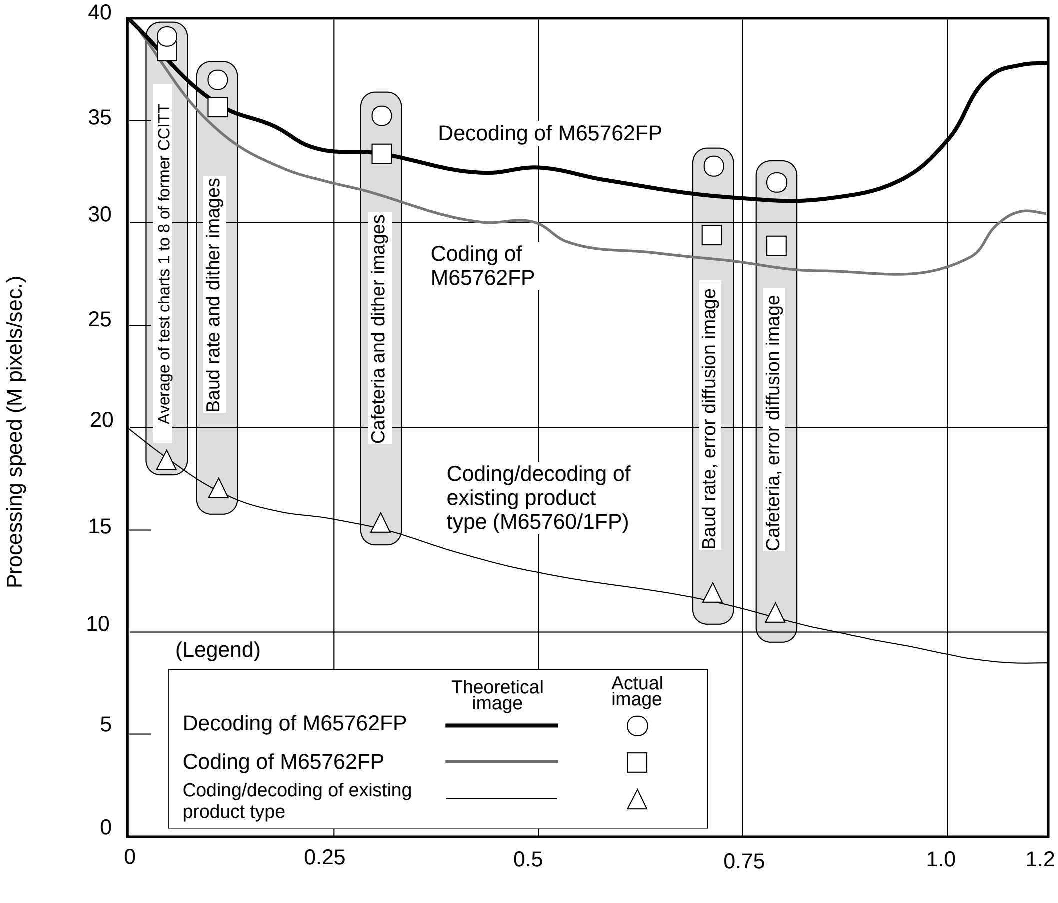


Figure 3 Estimated Processing Speed

Register Configuration

1. List of Registers

Table 1 List of Registers

Address	Register name	R/W	Content
0	System setting	W/R	- LSI H/W reset - Selects bit width of code data bus (32 bits/16 bits/8 bits). - Selects coding (image) data byte swap on code data bus. - Selects coding (image) data bit swap on code data bus. - Selects image data bit swap on image data bus. - Selects image data I/F (parallel I/F and serial I/F). - Selects bit width of image data bus (32 bits/16 bits).
1	Parameter setting	W/R	- Template selection (3-line template/2-line template). - Sets up the AT pixel position (127 max). (When set to 0, selects non-AT (default position).)
2	Command	W	- Context table RAM initializing processing command - Start/stop command (Coding/decoding, image data through, load/store of the line memory) - Start/stop command of load/store of context table RAM - Selects temporary stop/termination end mode.
2	Status	R	- Processing status (in process/end of process) - Ready for reading/writing coding (image) data on code data bus - Detects marker code (SDNORM, SDRST, ABORT, etc.). - Interrupt request status - SC counter overflow error - Processing mode (temporary stop/end of termination)
3	Interrupt enable setting	W/R	- Interrupt enable setting corresponding to each bit position of status register - Indicates pause/restart with marker code detected (at time of decoding)
4, 5	Setting number of pixels	W/R	- Sets the number of pixels per line. (a maximum of 10240 pixels with 2-line template selected)
6, 7	Setting number of lines	W/R	- Sets the number of lines to be coded/decoded (1 line or more, a maximum of 65535 lines)
8, 9	Number of processing lines	R	- Number of setting the coded/decoded lines (a maximum of 65535 lines)
A	Load/store buffer	W/R	- Buffer register that loads/stores context table RAM data from the MPU. (RAM address is automatically incremented each time data is written/read.)
B	Operation mode setting	W/R	- Sets the operation mode. (Coding/decoding, image data through, and load/store of line memory) - Selects read-through of head coding data in decoding (0 ~ 3 bytes). - Selects the typical prediction function. - Selects prohibition of line memory initialization.
C	Marker code setting	W	- Sets the terminal marker code in encoding (SDNORM/SDRST)
C	Marker code reading	R	- Reads a marker code in decoding. (SDNORM, SDRST, ABORT, others)
D	Scale-up/ scale-down setting	W/R	- Scale down in coding (1/2 scale-down of horizontal and vertical, horizontal OR processing) - Scale-up at time of decoding (scale-up of horizontal and vertical by twice)

2. Description on Register

(1) System setting register (W/R)
(Address: 0)

	d7(MSB)						d0(LSB)
SYS_REG:	PB	PI	BX	BS	DS	CB	HR

d0 (HR): H/W reset (0: Active status, 1: Reset status)
To reset H/W, set this bit to 1 then to 0. The entire LSI including register group and line memory is initialized by writing in this reset. However, context table RAM is not initialized.

d1-2 (CB): Selects the bit width of code data bus (d2 = 0, d1 = 0: 8-bit bus (CD0-7), d2 = 0, d1 = 1: 16-bit bus (CD0-15), d2 = 1, d1 = 0: 32-bit bus (CD0-31))
(Note1)Prohibition of setting for d2 = 1, d1 = 1
(Note2)For encoding in 16-/32-bit bus, the last encoding data is output followed by bit byte of '00' (3 bytes maximum) for word alignment of encoding data at the end.

d3 (DS): Selects data bit swap of image data bus (0: MSB first, 1: LSB first) → See Table 3.

d4 (BS): Selection of data bit swap of code data bus (0: MSB first, 1: LSB first) → See Table 2.

d5 (BX): Selection of data byte swap of code data bus (0: low order byte first, 1: high order byte first) → See Table 2.
(Note) BX is effective only when the host bus selects 16-bit/32-bit bus.

d6 (PI): Selection of image data input/output I/F (0: serial I/F, 1: parallel IF)

d7 (PB): Selection of bit width of image data bus (0: 32-bit bus (PD0-31), 1: 16-bit bus (PD0-15) → See Table 3.

Note) PB and DS are effective only when PI = 1.

Table 2 Line up of Coded Data/Image Data in Code Data Bus

Bus width (CB)	Swap (BX, BS)	Order of data in code data bus (CD)			
d2 d1	d5 d4	CD31 • • CD24	CD23 • • CD16	CD15 • • CD8	CD7 • • CD0
1 0	0 0	b24 • • b31	b16 • • b23	b8 • • b15	b0 • • b7
	0 1	b31 • • b24	b23 • • b16	b15 • • b8	b7 • • b0
	1 0	b0 • • b7	b8 • • b15	b16 • • b23	b24 • • b31
(32-bits)	1 1	b7 • • b0	b15 • • b8	b23 • • b16	b31 • • b24
0 1	0 0	–	–	b8 • • b15	b0 • • b7
	0 1	–	–	b15 • • b8	b7 • • b0
	1 0	–	–	b0 • • b7	b8 • • b15
(16-bits)	1 1	–	–	b7 • • b0	b15 • • b8
0 0	– 0	–	–	–	b0 • • b7
(8-bits)	– 1	–	–	–	b7 • • b0

(Note) b0 is image data, given in time series, on the left side of the first encoding data/screen. b31 is image data, given in time series, on the right side of the last encoding data/screen.

Table 3 Order of Image Data on Image Data Parallel Bus

Bit width	Swap	PD31 • • • • PD16	PD15 • • • • PD0
PB=0	DS=0	p0 • • • • p15	p16 • • • • p31
	DS=1	p31 • • • • p16	p15 • • • • p0
PB=1	DS=0	–	p0 • • • • p15
	DS=1		p15 • • • • p0

p0 is image data on the left side of the screen. p31 is image data on the right side of the screen.

(2) Parameter setting register (W/R)

(Address: 1)

	d7 d6	d5	d4 d0
PARA_REG :	AT	TM	AT

d0-4 (AT<0>-AT<4>): Low order 5-bits of AT pixel position (See Figure 2.)

d5 (TM): Selection of template (0: 3-line template, 1: 2-line template)

d6-7 (AT<5>-AT<6>): High-order 2-bits of AT pixel position (6th/7th bit)

(Example)	d7			d4				d0
3-line template, AT = 4	0	0	0	0	0	1	0	0
2-line template, AT = 48	0	1	1	1	0	0	0	0

(Note) AT pixel position is set (0 to 127) with AT <6:0>.
At the default position (AT pixel is not used), set AT = 0.
The 2-line template, prohibits AT = 1 to 4 from being set. The 3-line template prohibits AT = 1 to 2 from being set.

(3) Command Register (W)

(Address: 2)

	d7		d3			d0
CMD_REG:	0	JP	RC	JC	IC	

d0 (IC) :Context table RAM initialization start command (1: Start initialization)
Setting this bit to 1 starts to initialize context table RAM. When the initialization is completed automatically returns this bit to 0.

d1 (JC) :Processing (coding/decoding/through) start/end command (1: Start of processing, 0: End of processing)
Setting this bit to 1 starts processing (coding/decoding, image data through and lead/store of line memory). Before the issuance of this command, concrete operation mode must be set in the operation mode setup register.
When the processing for the number of setup lines ends with the end of termination selected this bit automatically returns to 0.
(Note)When this JC bit is set to 0 during the coding process (is in progress,) and input of image data is stopped, the coding is stopped (flashed) even if the set lines are not filled. When this bit is set to 0 auring decoding process, and input of encoding data ceases, processing for the number of setup lines is carried out assuming coding data "00" to have been input. In the case of multi-stripe coding, however, process must not be stopped by setting this bit to 0 except for the final stripe.

d2 (RC) :Load/store start/end command of context table RAM (1: Start of load/store, 0: End of load/store)
Setting this bit to 1 can load context data into context table RAM from outside via a buffer register or can store context data in outside. (See the section for buffer register.)
When load/store processing is completed, this bit must be set to 0.

d3 (JP) :Temporary stop mode of processing(coding/decoding/through)/termination end mode selection (1: Selection of temporary stop, 0: Selection of terminationend) Issuance of processing start command d1 (JC) with this JP bit set to 1 temporarily stops performing the process operation at the completion of processing for the number of setup lines. After that, reissuance of processing start command d1 (JC) restarts processing. (See Section 4.(3).)

(4) Status register (R)
(Address: 2)

	d7	d5					d0
STAT_REG :	0	PS	SC	IS	MS	DS	JS

d0 (JS) :Processing (initialization/coding/decoding/through) status (0: Processing in progress (temporary stop or initial), 1: Completion of processing)
This JS bit is set to 1 in the following cases: when the initialization is complete with the RAM initialization command issued (IC = 1), when all coding data is read completely at time of coding with the start command of termination end processing issued (JC=1, JP=0), and when all image data is read completely at time of image data through and at time of decoding. When the temporary stop processing start command is issued (JC = 1, JP = 1), this JS bit remains to be 0, even if the process for the number of setup lines ends. (However, an interruption occurs at time of temporary stop.)

d1 (DS) :Ready for reading/writing coding data (image data case of the through mode) on the code data bus (1: Ready, 0: Read/write disabled)
When this bit is set to 1, data can be read/written on the code data bus. (This bit is equivalent to the CDRQ pin.)

d2 (MS) :Detects marker code at time of decoding (0: Not detected, 1: Detected)
This bit is set to 1 when some marker code is detected at time of decoding.

d3 (IS) :Status of interrupt request (INTR pin) (0: Not requested, 1: Requested)

d4 (SC) :SC count-over error at time of coding (0: Normal, 1: Occurrence of SC counter overflow)
(Note)The SC counter is a counter for consecutive "FF" data bytes generated in the coding process. Though coding process continues if the SC counter overflows, normal coding data is not output (encoding error).

d5 (PS) :Processing (temporary stop/termination end) mode (1: Temporary stop processing mode, 0: Termination end processing mode)
This PS bit corresponds to the selection of process temporary stop/termination end of the d3 (JP) bit of command register.

(5) Interrupt enable register (W/R)

(Address: 3)

	d7		d3			d0
IENB_REG:	MP	0	SE	ME	DE	JE

d0 (JE) :Processing (initialization/coding/decoding/through) Temporary stop/termination end interrupt (0: Interrupt mask, 1: Interrupt enable)

d1 (DE) :Coding data (image data) read/write ready interrupt (0: Interrupt mask, 1: Interrupt enable)

d2 (ME) :Marker code detection interrupt at time of decoding (0: Interrupt mask, 1: Interrupt enable)

d3 (SE) :SC count-over error interrupt at time of coding (0: Interrupt mask, 1: Interrupt enable)
(Note)Bits d0 to d3 are interrupt enable of bits d0 to d2 and d4 corresponding to the status register. When one of the status bits set to interrupt enable is set to 1, the interrupt request signal (INTR) is asserted (for d0 (JE), an interrupt occurs even at the time of temporary stop).
When the status is set to 0 by H/W reset etc., or when interrupt factor is eliminated by interruption masking, INTR is negated. The status register is not cleared by occurrence of interruption or by R/W of interruption enable register.

d7 (MP) :Indication of pause at time of marker code detection (0: Indication of continuation/restart, 1: Indication of temporary pause)
If this MP bit is in advance set to 1 in decoding, the decoding temporarily pauses at the time of marker code detected.
(When the ME bit is set to 1, an interruption occurs when marker code is detected.)
When decoding process is not completed at time of temporary pause of marker detection, the register for setting the number of lines can be respecified (See Item (7).) Afterwards, setting this MP bit to 0 restarts the decoding process (the decoding process is carried out for the number of set lines).

(6) Register for setting the number of pixels (W/R)

	d7	d0
(Address: 4) PEL_REG_L:	PEL_L	
(Address: 5) PEL_REG_H:	0	PEL_H
	d7	d5 d0

d0-7 (PEL_L) :Sets the number of pixels in a line. (Low byte)

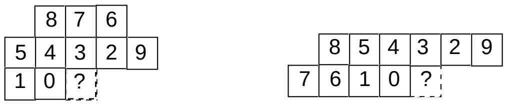
d0-5 (PEL_H) :Sets the number of pixels in a line. (Upper byte)
A maximum of 8192 pixels can be set at the 3-line template. A maximum of 10240 pixels can be set at the 2-line template.
Set the number of pixels to be actually coded (decoded) at time of scale-up (scale-down).
When the image data bus is 16-bits (32-bits) with the parallel I/F selected, set the number of pixels to multiples of 16 (multiples of 32).
With the serial I/F selected, set the number of pixels to multiples of 8.

(7) Register for setting the number of lines (W/R)

	d7 d0
(Address: 6) LSET_REG_L:	LSET_L
(Address: 7) LSET_REG_H:	LSET_H

d0-7 (LSET_L) :Sets the number of lines to be processed. (Low order byte) (1 to 65535: 0 line is not allowed.)

d0-7 (LSET_H) :Sets the number of lines to be processed. (High order byte)
At time of scale-down (scale-up), set the number of lines to be actually coded (decoded).
Set the number of lines (number of relative lines) ranging from the processing start command to be issued next to the temporary stop/termination end just after. This register must be set to a specific value before the issuance of the process start command.
As far as the following conditions are satisfied, this register can be rewritten in the course of processing.
•When the maximum value (65535) is set before issuance of the processing start command, an arbitrary value can be set once in the course of processing.
•When a value except for the maximum value (65535) is set before issuance of the processing start command, and the value requires to be respecified in the course, respecify the maximum value (65535) once and then respecify a desired value.



3-line template 2-line template

Figure 4. Address Assignment of Context Table RAM
(Number for address bit (LSB: 0, MSB: 9), MSB: 9 for AT pixel)

(8) Processing line count register (R)

	d7 d0
(Address: 8) LIN_REG_L:	LINE_L
(Address: 9) LIN_REG_H:	LINE_H

d0-7 (LINE_L) :Read out the number of lines actually processed (low byte) (0 to 65535)

d0-7 (LINE_H) :Read out the number of lines actually processed (upper byte)
The number of processed lines ≥ number of set lines, coding/decoding/through processing stop temporary/end of processing.
(Note)The number of lines in this process is cleared to 0 with the processing start command issued.

(9) Buffer register (W/R)

	d7 d0
(Address: A) DWR_BUF:	DWR

d0-7 (DWR) :Data for loading/storing context table RAM
This register is a buffer for loading data into the context table RAM via the host bus or for storing data outside. After issuance of load/store start command of the context table RAM (command register d3 = 1), this register is available to start loading or storing data. Prediction value (MPS) and prediction unmatched probability (LSZ) can be stored in context table RAM for a unit of 1024 contexts in total. Figure 4 and Table 4 provide the address assignment of context table RAM and the data bit array.
Since context table RAM is 2-byte data, access is gained alternately in order from low byte to upper byte. Each time two-byte access is gained, the RAM address is automatically incremented (sequential access from address 0).
(Note1)Data is not allowed to be loaded and stored at a time. Random access to RAM is not allowed.
(Note2)Only 133 types specified by the JBIG international standard (See attached Figure A.2) are allowed to be specified for the LSZ value.
(For example, load '5a1d' for initialization.)

Table 4. Data Bit Array of Context Table RAM

High order byte		Low order byte
d15	d14 • • • • • d8	d7 • • • • • d0
MPS	L14 • • • • • L8	L7 • • • • • L0

MPS :Prediction value MPS (0/1)
L14-0 :Low 15-bits of prediction unmatched probability LSZ ('0001' to '5b12')

(10) Operation mode setting register (W/R)

(Address: B)

MOD_REG:

d7				d0
TP	LI	OB	LIO	MOD

This register is used to set the LSI operation mode and requires to be set before issuance of the processing start command (command register d1 (JC) = 1).

d0-1 (MOD) :Operation mode setting (d1 = 0, d0 = 0: Coding, d1 = 1, d0 = 0: Image data through (image data I/F→Code data I/F) load/store, d1 = 0, d0 = 1: Decoding, d1 = 1, d0 = 1: Image data through (code data I/F→Image data I/F) load/store)

d2- 3 (LIO) :Load/store selection of image data of line memory (d2 = selection of load, d3 = selection of store)
In the case of multi-stripe, this LIO bit is set according to the following table, to load image data for reference line from outside into line memory before coding/decoding of stripes or to store image data stored in line memory into outside after encoding/decoding of stripes. This LIO bit is effective only in the image data through mode (d1 = 1).

(Notes)

- LIO (d3, d2) = (1, 1) not allowed being set.
- When selection of load/store of image data of line memory, temporary stop (d3 (JP) = 1 of command register) is not allowed to be set.
- When load/store mode of image data is selected, the number of lines to be transferred must be set in the register setting the number of lines.
- The number of lines for image data load to line memory must be 2-line either case of 2-line template or 3-line template.
(This is because typical prediction (LNTP) cannot be judged correctly with only a line.)

Table 5. Operation Mode List

Operation mode (d1, d0)	Load/store LIO (d3, d2)	Operation mode	Remarks
0 , 0	X , X	Coding mode	Normal coding mode
0 , 1	X , X	Decoding mode	Normal decoding mode
1 , 0	0 , 0	Image data through (image data I/F→ code data I/F)	For inter-IF transfer of image data
	0 , 1	Image data load to line memory (Input from image data I/F)	For loading of reference line to LSI
	1 , 0	Image data store of line memory (output to code data I/F)	For storing line memory to outside
1 , 1	0 , 0	Image data through (code data I/F→ image data I/F)	For inter-I/F transfer of image data
	0 , 1	Image data load to line memory (input from code data I/F)	For loading of reference line to LSI
	1 , 0	Image data store of line memory (output to image data I/F)	For storing line memory to outside

d4-5 (OB) :Sets head of the coding data read-through at time of decoding (0 to 3: Sets the number of read-through bytes. For example, with d4 = 0 and d5 = 1, read-through of 2 bytes)
When OB is set to 1 to 3 at time of decoding, and the first stripe decoding processing start command is issued, the head data for the number of set bytes is to be read through (not used for decoding process). With OB set to 0, no data is read through (normal decoding process).
For example, if the code data bus is 32/16-bits, and the head of coding data does not contact the word boundary, this function is used.
(Note)When the code data bus is 8-bits, this function is effective.

d6 (LI) :Prohibition of line memory initialization (0: Indication of initialization, 1: Prohibition of initialization)
When first stripe coding/decoding process start command is issued, and LI = 1, initialization of built-in line memory is prohibited.
(The final image data, coded/decoded just before, that is left in line memory is used as the reference line data at the head of next coding/decoding operation.)With LI = 0, built-in line memory is initialized.(Full white (0) data is used as the reference line data at the head of next coding/decoding operation.)
When the previous stripe is terminated at the SDNORM marker with coding/decoding of the multi-stripe configuration, this bit is set to initialization prohibition (1) to make the data of previous stripe left in line memory available as the coding reference line data of the next stripe. (For details, see 4. (6) Sequence.)
(Note)With LI = 1, this LI bit is cleared (to 0) by H/W reset writing to an external reset pin or system setup register. At the same time, built-in line memory is also initialized.

d7 (TP) :Selection of typical prediction at time of coding/decoding (0: Sets typical prediction function to OFF, 1: Sets typical prediction function to ON.)
This bit is set to 1 when encoding/decoding process is carried out using the typical prediction function.

(11) Marker code set up register (W)

(Address: C) MSET_REG:

d7	d0
MSET	

d0-7(MSET) :The End marker code used during coding is set (SDNORM = 02h, SDRST = 03h, etc.)
The Byte set to this register is output attached to coding data as the end marker during coding.

(12) Marker code read out register (R)

(Address: C) MDET_REG:

d7	d0
MDET	

d0-7(MDET) :Reads out the marker code detected during decoding (SDNORM = 02h, SDRST = 03h, ABORT = 04h, etc.)
Marker code bytes detected at time of decoding can be read directly.

(13) Scale-up/scale-down set register (W/R)

	d7	d4				d0
(Address: D) CONV_REG:	0	HO	HR	VR	HE	VE

d0 (VE) :Selection of scale-up in vertical direction during decoding (0: Equal size, Scale-up by twice)
d1 (HE) :Selection of scale-up in horizontal direction during decoding (0: Equal size, Scale-up by twice)

Scale-up function is effective only in decoding (Scale-up enabled)

d2 (VR) :Selection of scale-down in vertical direction (0: Equal size, Scale-down by 1/2)
d3 (HR) :Selection of scale-down in horizontal direction (0: Equal size, Scale-down by 1/2)
d4 (HO) :Selection of thinned-out processing in horizontal direction (0: Simple thinned-out, 1: OR processing)

Scale-down function is effective only in encoding (Scale-down enabled)

(Note1) During coding, simple thinned-out is applied to 1/2 scale-down in vertical direction (Odd lines are skipped in reading.)
(Note2) With VR = 1 during coding, the number of lines on input image data must be larger by twice than the set value of line count setup register.
(Note3) With VE = 1 during decoding, the number of lines on output image data must be larger by twice than the set value of line count setup register.

3. Register Initial Value

Registers are initialized as provided in the following table by writing H/W reset into the external reset pin or system setup register.

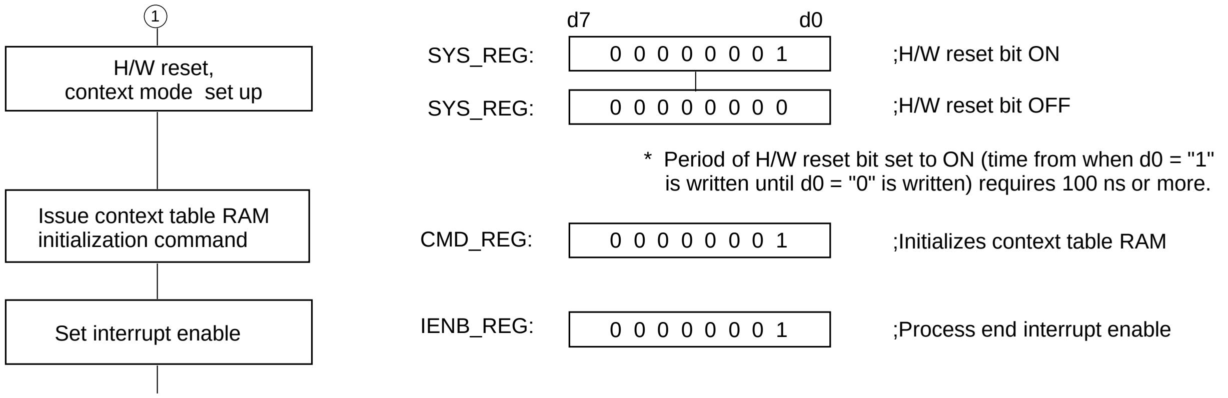
Table 6. Initial Values of Registers

Register	Initial value	Register	Initial value
System setting	0 0 h (Note)	Number of processed lines	0 0 h
Parameter setting	0 0 h	Buffer register	Indefinite
Command	0 0 h	Operation mode setting	0 0 h
Status	0 0 h	Marker code setting	0 0 h
Interrupt enable	0 0 h	Marker code reading	0 0 h
Pixel setting	0 0 h	Scale-up/scale-down setting	0 0 h
Line count setting	0 0 h		

(Note) When H/W reset is written into the system setting register, written value is set in the system setting register.

4. Register Setting Sequence

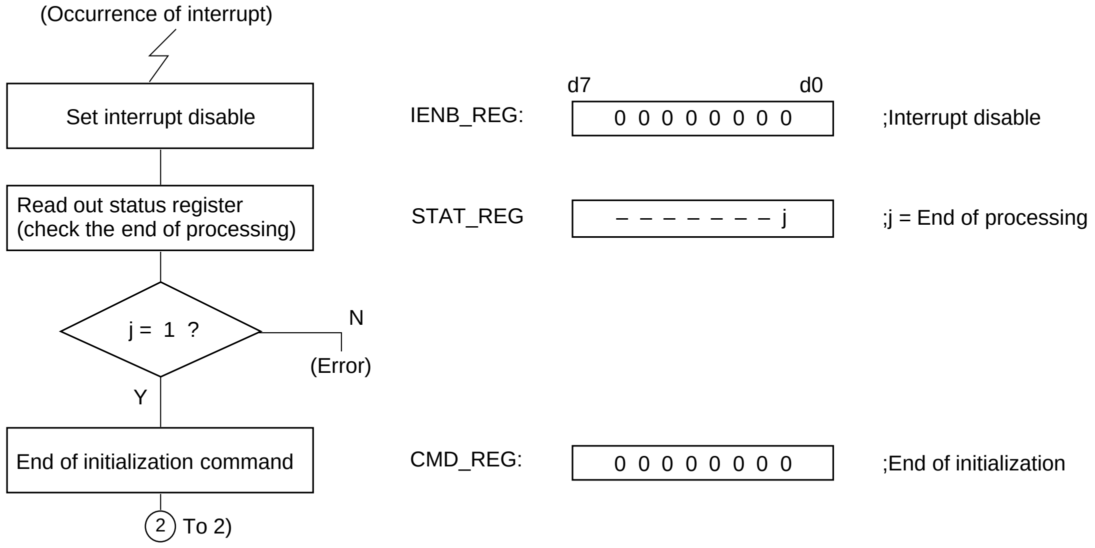
(1) Initialization sequence of built-in line memory and context table RAM
This sequence is used to carry out initialization sequence (0 clear) of context table RAM after the initialization (Note) of the built-in line memory by H/W reset.
When the initialization is unnecessary (the contents of the current status table are directly used), this sequence is unnecessary.



* Period of H/W reset bit set to ON (time from when d0 = "1" is written until d0 = "0" is written) requires 100 ns or more.

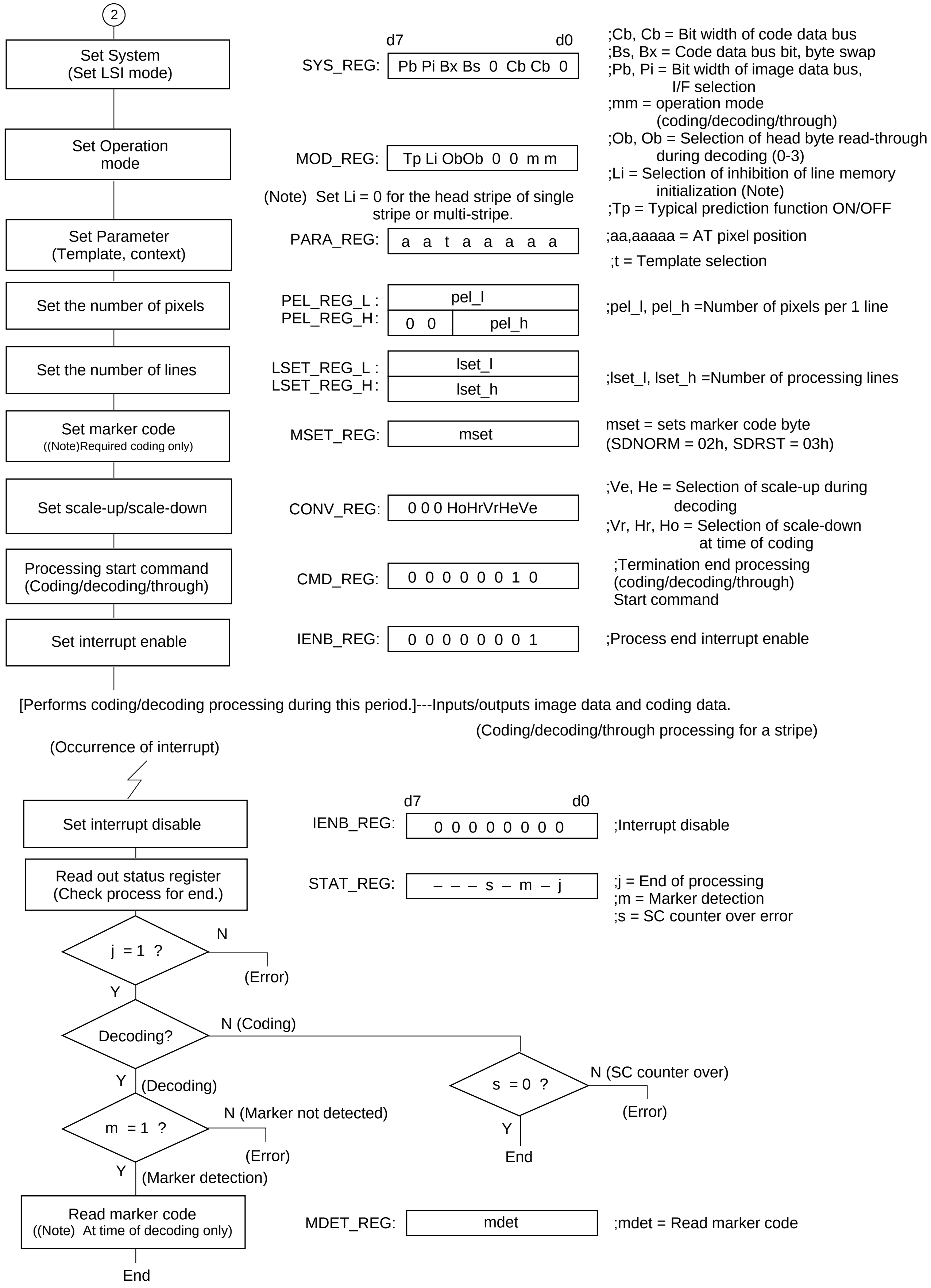
Context table RAM is initialized (0 clear) in this period.

The number of clocks required for initialization is as follows:
1024 +a[Clock]

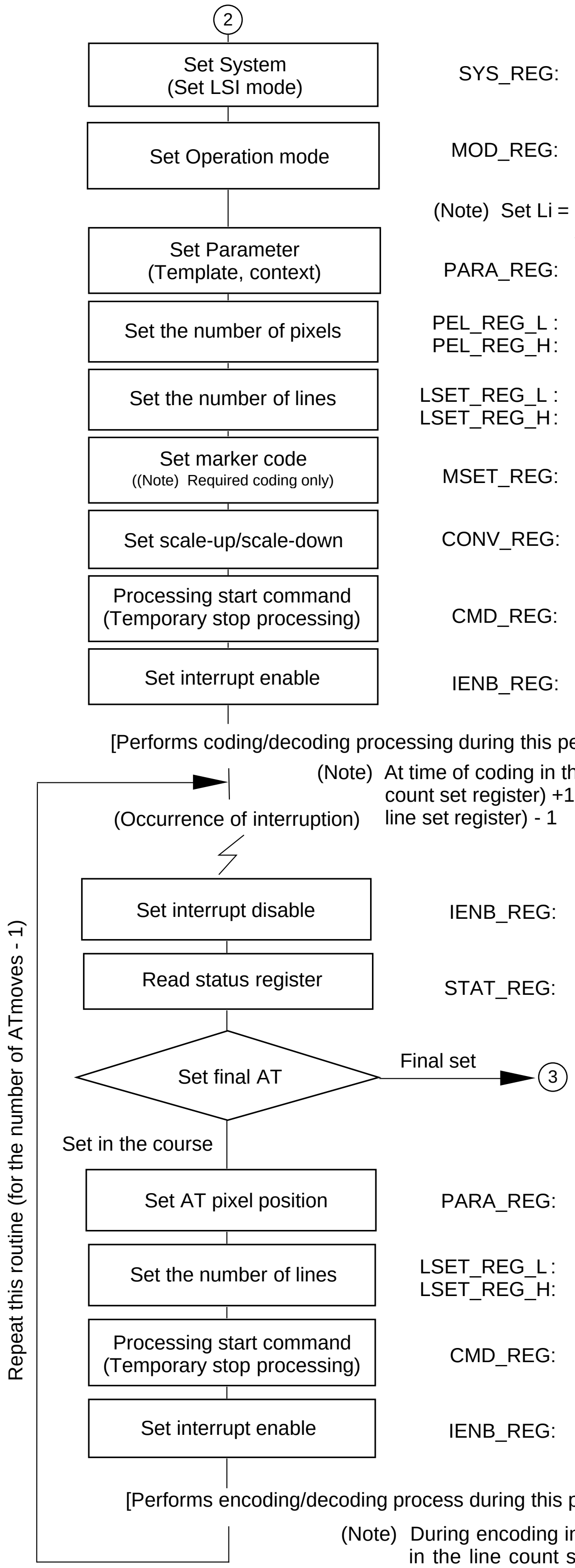


(Note) Line memory is initialized by H/W reset to prepare the all white (0) data as a reference line to provide for the start of coding/decoding process and to initialize LNTP bit (LNTP = 1) for typical prediction.

(2) Stripe coding/decoding (without change in AT pixel position)/image data through processing sequence


2
Set System (Set LSI mode)
d7
d0
SYS_REG: Pb Pi Bx Bs 0 Cb Cb 0
;Cb, Cb = Bit width of code data bus
;Bs, Bx = Code data bus bit, byte swap
;Pb, Pi = Bit width of image data bus, I/F selection
;mm = operation mode (coding/decoding/through)
;Ob, Ob = Selection of head byte read-through during decoding (0-3)
;Li = Selection of inhibition of line memory initialization (Note)
;Tp = Typical prediction function ON/OFF
Set Operation mode
MOD_REG: Tp Li ObOb 0 0 m m
(Note) Set Li = 0 for the head stripe of single stripe or multi-stripe.
Set Parameter (Template, context)
PARA_REG: a a t a a a a a
;aa,aaaaa = AT pixel position
;t = Template selection
Set the number of pixels
PEL_REG_L : pel_l
PEL_REG_H: 0 0 pel_h
;pel_l, pel_h =Number of pixels per 1 line
Set the number of lines
LSET_REG_L : lset_l
LSET_REG_H: lset_h
;lset_l, lset_h =Number of processing lines
Set marker code ((Note)Required coding only)
MSET_REG: mset
mset = sets marker code byte (SDNORM = 02h, SDRST = 03h)
Set scale-up/scale-down
CONV_REG: 0 0 0 HoHrVrHeVe
;Ve, He = Selection of scale-up during decoding
;Vr, Hr, Ho = Selection of scale-down at time of coding
Processing start command (Coding/decoding/through)
CMD_REG: 0 0 0 0 0 0 1 0
;Termination end processing (coding/decoding/through) Start command
Set interrupt enable
IENB_REG: 0 0 0 0 0 0 0 1
;Process end interrupt enable
[Performs coding/decoding processing during this period.]---Inputs/outputs image data and coding data.
(Coding/decoding/through processing for a stripe)
(Occurrence of interrupt)
Set interrupt disable
d7
d0
IENB_REG: 0 0 0 0 0 0 0 0
;Interrupt disable
Read out status register (Check process for end.)
STAT_REG: – – – s – m – j
;j = End of processing
;m = Marker detection
;s = SC counter over error
j = 1 ?
N
(Error)
Y
Decoding?
N (Coding)
Y
(Decoding)
s = 0 ?
N (SC counter over)
(Error)
Y
End
m = 1 ?
N (Marker not detected)
(Error)
Y
(Marker detection)
Read marker code ((Note) At time of decoding only)
MDET_REG: mdet
;mdet = Read marker code
End

(3) Stripe encoding/decoding (with change in AT pixel position) processing sequence


2
Set System (Set LSI mode)
Set Operation mode
Set Parameter (Template, context)
Set the number of pixels
Set the number of lines
Set marker code ((Note) Required coding only)
Set scale-up/scale-down
Processing start command (Temporary stop processing)
Set interrupt enable
(Occurrence of interruption)
Set interrupt disable
Read status register
Set final AT
Final set
3
Set in the course
Set AT pixel position
Set the number of lines
Processing start command (Temporary stop processing)
Set interrupt enable
Repeat this routine (for the number of ATmoves - 1)

Step	Register	d7 … d0	Comment
Set System (Set LSI mode)	SYS_REG:	Pb Pi Bx Bs 0 Cb Cb 0	;Cb, Cb = Bit width of code data bus ;Bs, Bx = Code data bus bit, byte swap ;Pb, Pi = Bit width of image data bus, I/F selection ;m = operation mode (encoding/decoding)
Set Operation mode	MOD_REG:	Tp Li Ob Ob 0 0 0 m	;Ob, Ob = Selection of head byte read-through during decoding (0-3) ;Li = Selection of inhibition of line memory initialization (Note) ;Tp = Typical prediction function ON/OFF
Set Parameter (Template, context)	PARA_REG:	a a t a a a a a	;aa,aaaaa = AT pixel position ;t = Template selection
Set the number of pixels	PEL_REG_L : PEL_REG_H:	pel_l 0 0 pel_h	;pel_l, pel_h = Number of pixels per 1 line
Set the number of lines	LSET_REG_L : LSET_REG_H:	lset_l lset_h	;lset_l, lset_h = Number of processing lines (Note) Set the number of processing lines to position change of AT pixel.
Set marker code ((Note) Required coding only)	MSET_REG:	mset	mset = sets marker code byte (SDNORM = 02h, SDRST = 03h)
Set scale-up/scale-down	CONV_REG:	0 0 0 HoHrVrHeVe	;Ve, He = Selection of scale-up at time of decoding ;Vr, Hr, Ho = Selection of scale-down at time of coding
Processing start command (Temporary stop processing)	CMD_REG:	0 0 0 0 1 0 1 0	;Temporary stop processing (coding/decoding) Start command
Set interrupt enable	IENB_REG:	0 0 0 0 0 0 0 1	;Process end interrupt enable

(Note) Set Li = 0 for single stripe or the head stripe of multi-stripe.

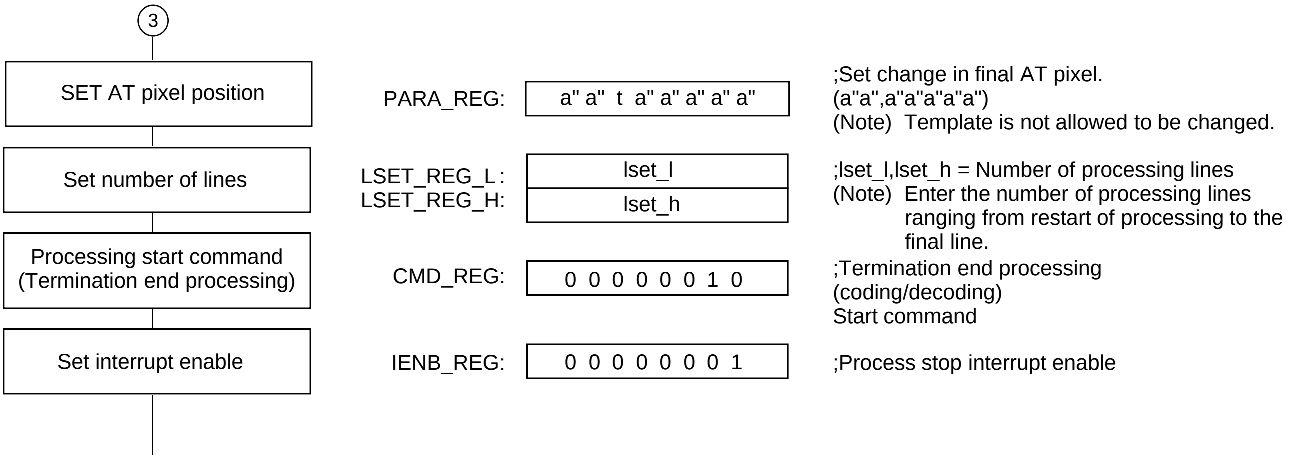
[Performs coding/decoding processing during this period.]---Input/output first image data and coding data.

(Note) At time of coding in the first processing, (number of lines of input image data) = (value set in the line count set register) +1. During decoding, (number of lines in output image data) = (value set in the line set register) - 1

Step	Register	d7 … d0	Comment
Set interrupt disable	IENB_REG:	0 0 0 0 0 0 0 0	;Interrupt disable
Read status register	STAT_REG:	– – p – – – – j	;Status check j=0, p=1; Temporary stop status
Set AT pixel position	PARA_REG:	a' a' t' a' a' a' a' a'	;Set change of AT pixel (a'a',a'a'a'a'a') (Note) Template is not allowed to be changed.
Set the number of lines	LSET_REG_L: LSET_REG_H:	lset_l lset_h	;lset_l,lset_h = Number of processing lines (Note) Set the number of processing lines ranging from processing restart to change of AT pixel position
Processing start command (Temporary stop processing)	CMD_REG:	0 0 0 0 1 0 1 0	;Temporary stop processing (encoding/decoding) Start command
Set interrupt enable	IENB_REG:	0 0 0 0 0 0 0 1	;Process stop interrupt enable

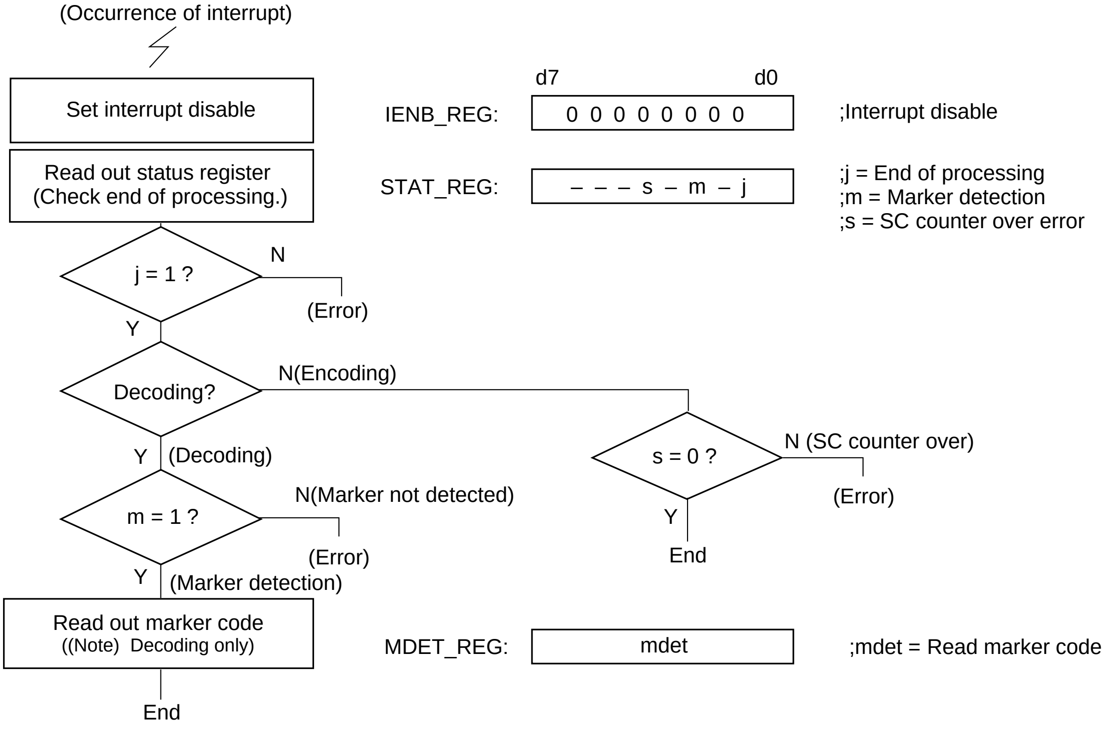
[Performs encoding/decoding process during this period.]---Inputs/outputs image data and coding data in the course.

(Note) During encoding in the course of processing, (number of lines in input image data) = (value set in the line count set register). At time of decoding, (number of lines in output image data) = (value set in the line count set register).


3
SET AT pixel position
PARA_REG:
a" a" t a" a" a" a" a"
;Set change in final AT pixel.
(a"a",a"a"a"a"a")
(Note) Template is not allowed to be changed.
Set number of lines
LSET_REG_L :
LSET_REG_H:
lset_l
lset_h
;lset_l,lset_h = Number of processing lines
(Note) Enter the number of processing lines ranging from restart of processing to the final line.
Processing start command
(Termination end processing)
CMD_REG:
0 0 0 0 0 0 1 0
;Termination end processing
(coding/decoding)
Start command
Set interrupt enable
IENB_REG:
0 0 0 0 0 0 0 1
;Process stop interrupt enable

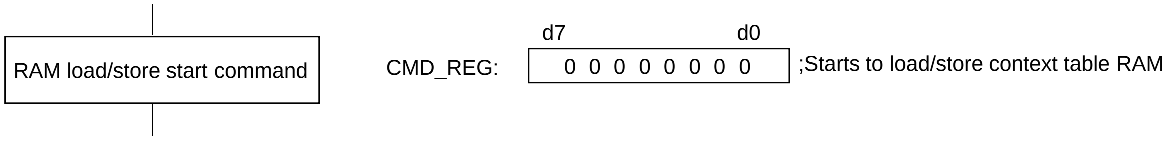
[Performs coding/decoding processing during this period.] --- Inputs/outputs final image data and coding data.

(Note) During coding in the final processing, (number of lines in input image data) = (value set in the line count set register) – 1. During decoding, (number of lines in output image data) = (value set in the line count set register) + 1.

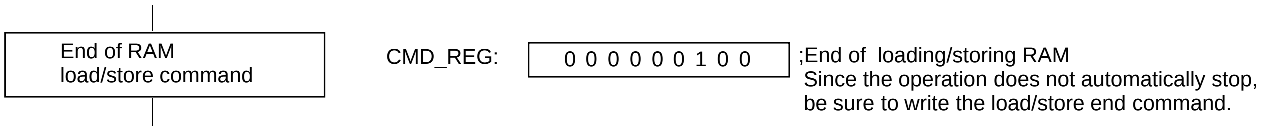


(4) Load/store processing sequence of the context table RAM

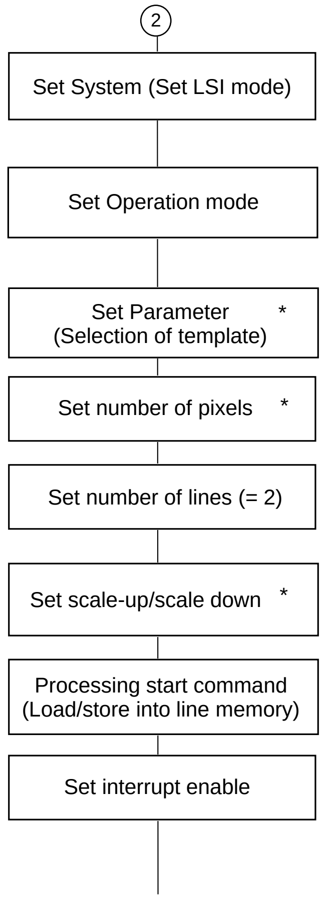
This sequence is used to load or store context table RAM.


RAM load/store start command
d7
d0
CMD_REG:
0 0 0 0 0 0 0 0
;Starts to load/store context table RAM

[Stores (loads) the context table RAM during this period.
Context RAM data is stored (loaded) via buffer register.
Reading (writing) 2 bytes automatically increments the RAM address.
(Note) Reading (storing) operation and writing (loading) operation are not allowed to be done at a time.


End of RAM
load/store command
CMD_REG:
0 0 0 0 0 0 1 0 0
;End of loading/storing RAM
Since the operation does not automatically stop, be sure to write the load/store end command.

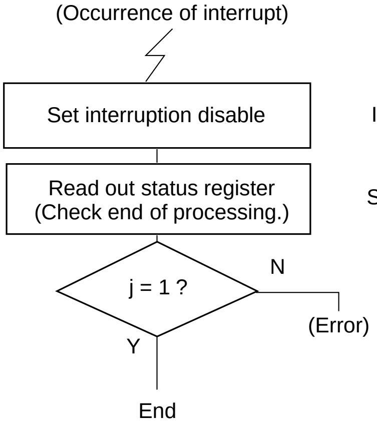
(5) Load/store processing sequence of line memory image data



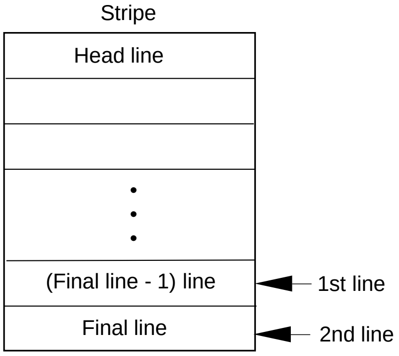
Register	d7 … d0	Comment
SYS_REG:	Pb Pi Bx Bs Ds Cb Cb 0	;Cb, Cb = Bit width of code data bus ;Ds = Bit swap of image data bus ;Bs, Bx = Code data bus bit, byte swap ;Pb, Pi = Bit width of image data bus, I/F selection
MOD_REG:	Tp Li 0 0 LioLio 1 m	;m = Operation mode (selection of through mode) ;Lio, Lio = 01 or 10 (selection of load or store) ;Li = 1 (selection of prohibition of line memory initialization) ;Tp = Typical prediction function ON/OFF (Note 1)
PARA_REG:	– – t – – – – –	;t = Selection of template
PEL_REG_L : PEL_REG_H:	pel_l 0 0 | pel_h	;pel_l, pel_h = Number of pixels per line
LSET_REG_L : LSET_REG_H:	lset_l lset_h	lset_l, lset_h = 2 (Number of processed lines) (Note 2)
CONV_REG:	0 0 0 HoHrVrHeVe	;Ve, He = Selection of scale-up during decoding ;Vr, Hr, Ho = Selection of scale-down during coding
CMD_REG:	0 0 0 0 0 0 1 0	;Load/store processing start command of image data
IENB_REG:	0 0 0 0 0 0 0 1	;Process end interrupt enable

*Settings of template selection, number of pixels per line, selection of scale-up/scale-down and typical prediction function must meet the settings at time of stripe coding/decoding to be carried out after this.

[Performs loading/storing process during this period] --- Inputs (outputs) image data. (Transfer processing of image data for 2 lines)



Register	d7 … d0	Comment
IENB_REG:	0 0 0 0 0 0 0 0	;Interrupt disable
STAT_REG:	– – – – – – – j	;j = End of processing

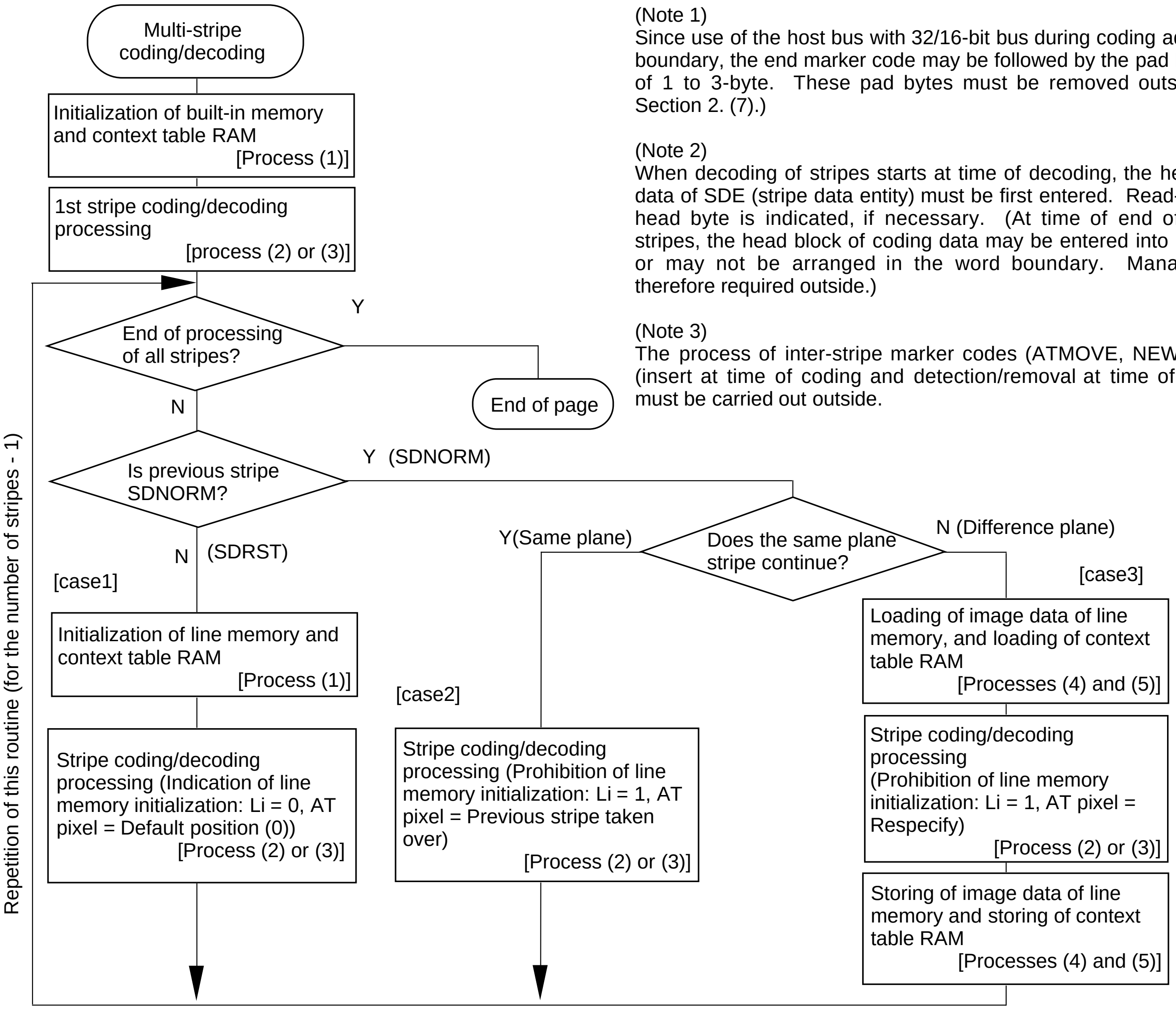


Note 1) For ON/OFF bit of TP function in the image data processing, the ON/OFF bit of the TP function just before coding/decoding shall be kept.

Note 2) In the image data load/store processing, be sure to set the number of transfer lines to "2".
(The 1st line is data on the line (final line - 1) of the stripe. The 2nd line is data on the last line of stripe.)
When a line stripe is adopted for the first stripe of the page in the image data store processing, and read out line of the first line is outside data of stripe, the all white data must used for replacement or the image data load function must be used in advance to clear line memory.

(6) Total sequence of multi-stripe coding/decoding

For an image with a page consisting of more than one stripe or plane, coding or decoding process must be carried out in units of stripe after initialization.


Multi-stripe coding/decoding
Initialization of built-in memory and context table RAM [Process (1)]
1st stripe coding/decoding processing [process (2) or (3)]
End of processing of all stripes?
Y
End of page
N
Is previous stripe SDNORM?
Y (SDNORM)
N (SDRST)
[case1]
Initialization of line memory and context table RAM [Process (1)]
Stripe coding/decoding processing (Indication of line memory initialization: Li = 0, AT pixel = Default position (0)) [Process (2) or (3)]
Does the same plane stripe continue?
Y(Same plane)
[case2]
Stripe coding/decoding processing (Prohibition of line memory initialization: Li = 1, AT pixel = Previous stripe taken over) [Process (2) or (3)]
N (Difference plane)
[case3]
Loading of image data of line memory, and loading of context table RAM [Processes (4) and (5)]
Stripe coding/decoding processing (Prohibition of line memory initialization: Li = 1, AT pixel = Respecify) [Process (2) or (3)]
Storing of image data of line memory and storing of context table RAM [Processes (4) and (5)]
Repetition of this routine (for the number of stripes - 1)

(Note 1)
Since use of the host bus with 32/16-bit bus during coding adopts word boundary, the end marker code may be followed by the pad bytes ('00') of 1 to 3-byte. These pad bytes must be removed outside. (See Section 2. (7).)

(Note 2)
When decoding of stripes starts at time of decoding, the head coding data of SDE (stripe data entity) must be first entered. Read-through of head byte is indicated, if necessary. (At time of end of decoding stripes, the head block of coding data may be entered into LSI (FIFO) or may not be arranged in the word boundary. Management is therefore required outside.)

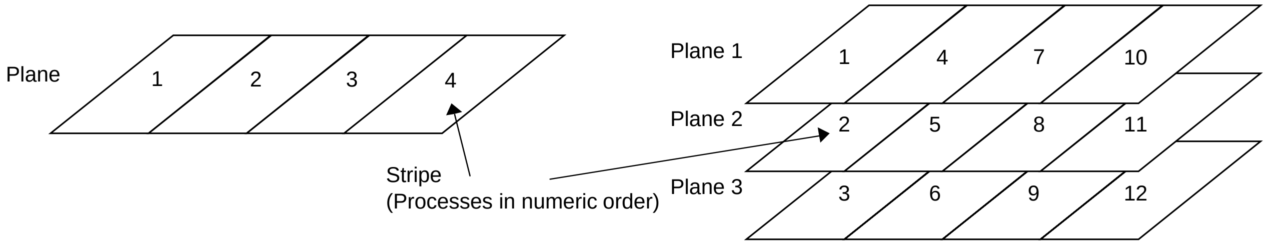
(Note 3)
The process of inter-stripe marker codes (ATMOVE, NEWLEN, etc.) (insert at time of coding and detection/removal at time of decoding) must be carried out outside.

(Description)
If the end marker of the previous stripe is SDRST, the status must be initialized for coding/decoding the next stripe. Start to carry out the process of next stripe by returning the AT pixel position to the default position after the initialization of built-in line memory and context table RAM. [case 1]
If the termination marker of the previous stripe is SDNORM, the status of the previous stripe must be taken over for coding/decoding the next stripe. If the stripe of the same plane is continuously coded/decoded, the AT pixel position takes over the final value of the previous stripe and the process of the next stripe is to start without initializing line memory and context table RAM to use the status of line memory and context table RAM at the end of previous stripe for the next stripe. [case 2]
On the other hand, since the status at the end of pre-stripe status of the same plane must be respecified for the status of line memory and context table RAM, line memory and context table RAM are to be loaded into LSI to respecify the AT pixel position and to start processing the next stripe when alternately coding/decoding stripes of different planes. After coding/decoding of stripe, save line memory and context table RAM for next stripe. [case 3]

(Example)

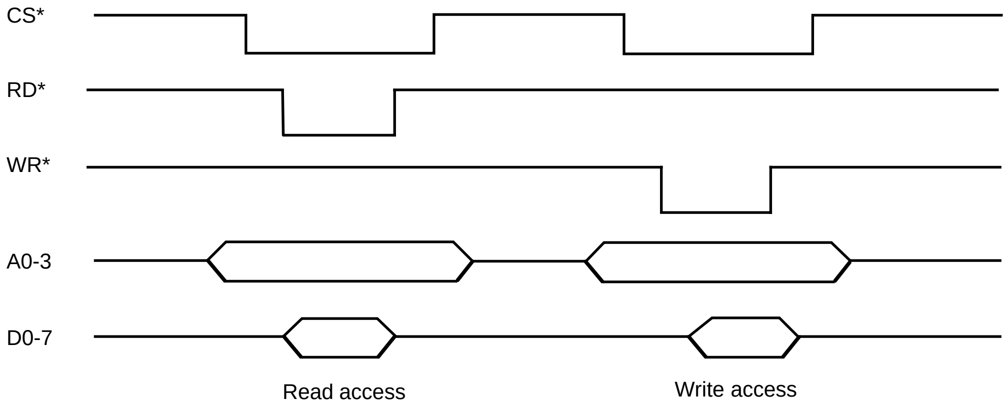
• Single plane, multi-stripe

• Multiple planes and multi-stripe



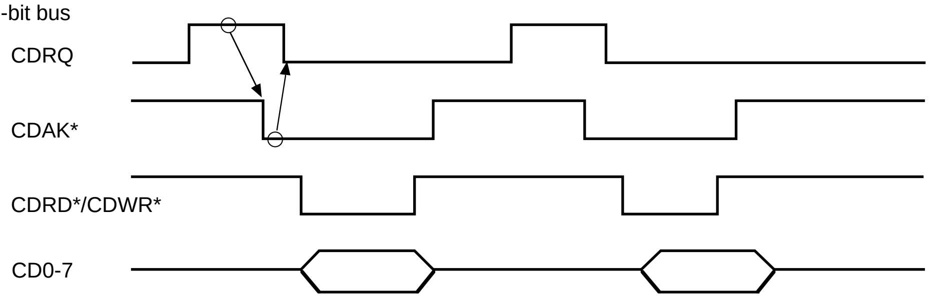
Timing Chart

1. Host bus I/F

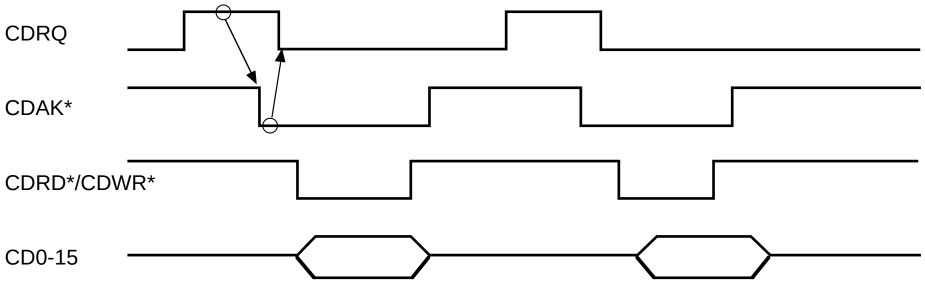


2. Code data I/F

(a) For 8-bit bus

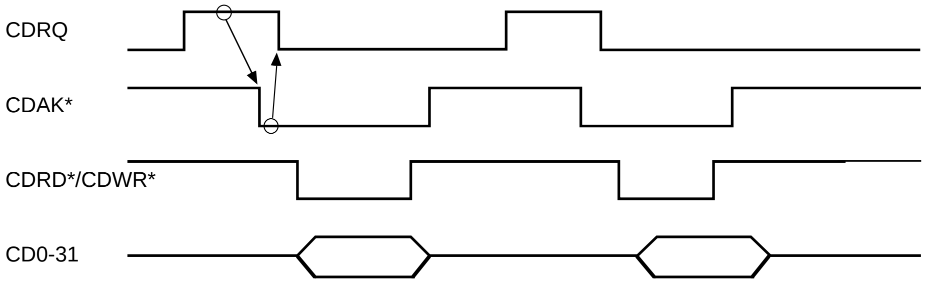


(b) For 16-bit bus



(Note) For 16-bit bus, only the word access (CD0-15) is allowed.

(c) For 32-bit bus



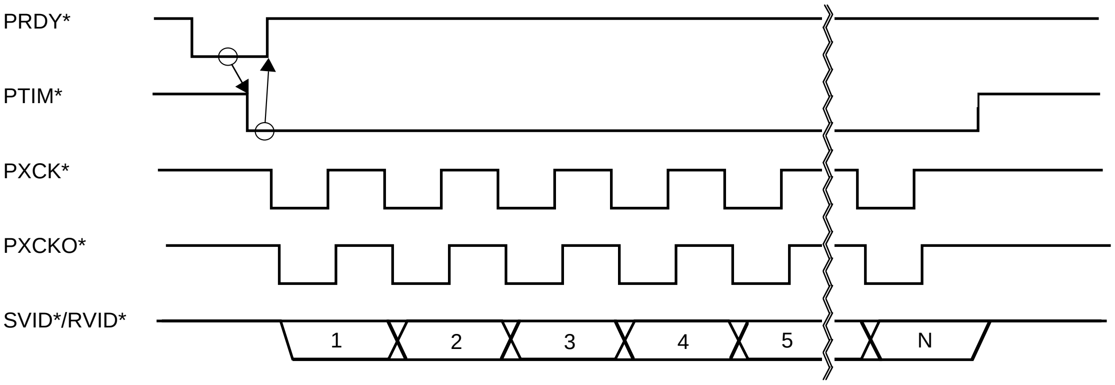
(Note) For 32-bit bus, only the long word access (CD0-31) is allowed.

(Description)

CDRQ can be checked for being asserted (H) to assert (L) CDAK*.
Asserting (L) CDAK* negates (L) CDRQ.
Asserting (L) section of CDRD*/CDWR* must be included in the CDAK* asserting section (L).

3. Image Data I/F

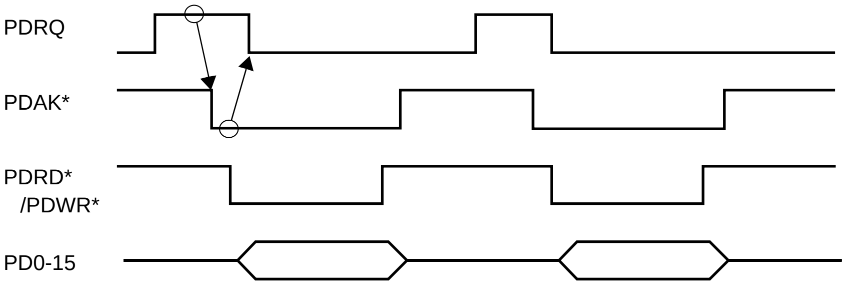
(1) Serial image data I/F



(Note) The above chart shows a timing for a line (N pixel/line).

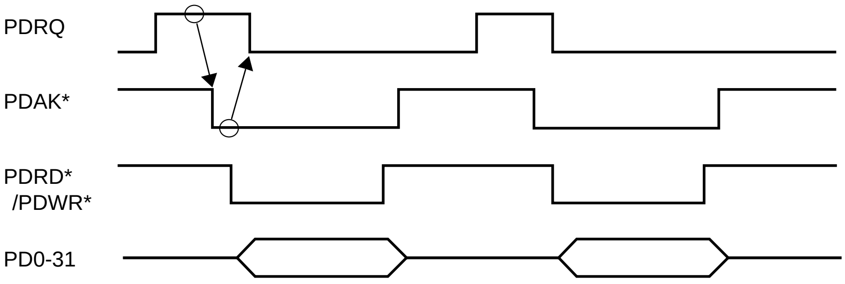
(Description)
PRDY* can be checked for being asserted (L) to assert (L) PTIM*.
Asserting (L) PTIM* negates (H) PRDY*.
PXCKO* is an output of having gated PXCK* input with PTIM*.
The image data (SVID*/RVID*) is input/output in synchronization with PXCK* or PXCKO*.

(2) Parallel image data I/F
(a) 16-bit bus



(Note) For 16-bit bus, only the word access (PD0-15) is allowed.

(b) 32-bit bus



(Note) For 32-bit bus, only the long word access (PD0-31) is allowed.

(Description)
PDRQ can be checked for being asserted (H) to assert (L) PDAK*.
Asserting (L) PDAK* negates (H) PDRQ.
Asserting (L) section of PDRD*/PDWR* must be included in the asserting section (L) of PDAK*.

System Configuration Example

1. Application Examples to Digital PPC and FAX Hybrid Machine

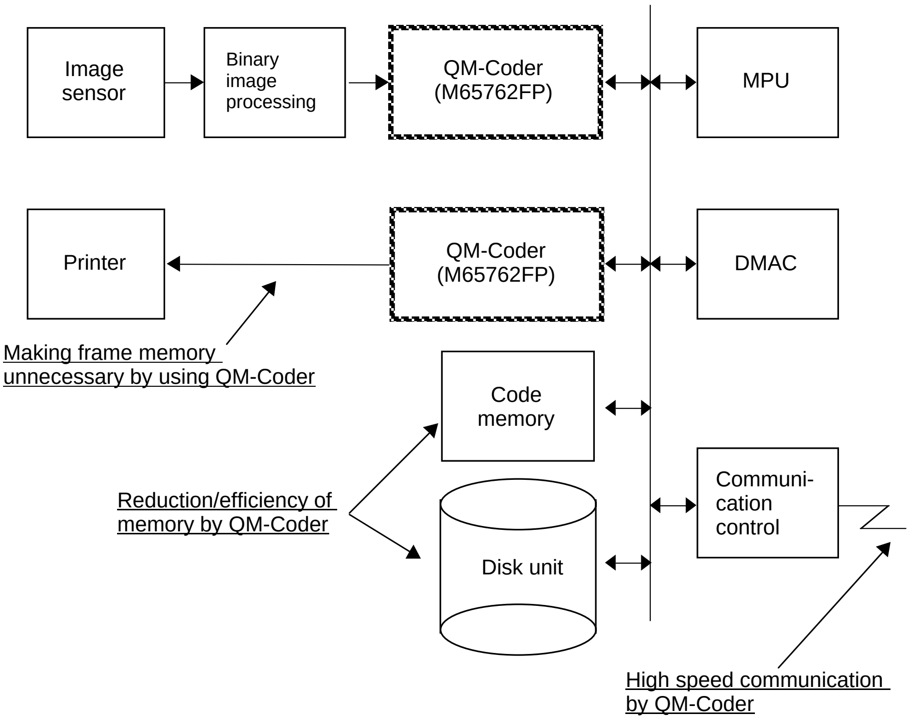


Figure 5. Application Examples to Digital PPC and Fax Hybrid Machine

2. Application Example to Printer

High Speed Transfer from PC/WS to Printer (LBP/UP), and Reduction of Memory

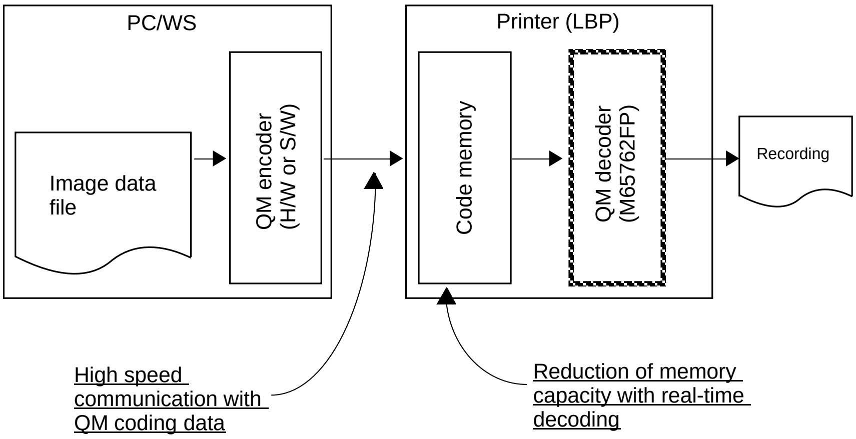


Figure 6 Application Example to Printer

[Appendix A.1] JBIG Data Structure

```
B I E          ;Bi-level  Image  Entity
  B I H          ;Bi-level  Image  Header
     DL       1       ;lowest  resolution  layer
     D        1       ;finel    resolution  layer
     P        1       ;number  of  bit-planes
     -        1       ;dummy  0
     XD       4       ;horizontal  dimmension  at  highest  resolution
     YD       4       ;vertical    dimmension  at  highest  resolution
     LD       4       ;number  of  lines  per  stripe  at  lowest  resolution
     MX       1       ;maximum  horizontal  offsets  allowed  for  AT  pixel
     MY       1       ;maximum  vertical    offsets  allowed  for  AT  pixel
     Order    1       ;order  byte
       -              b7-4;dummy   0
       HITOLO         b3  ;resolution-order  distinction
       SEQ            b2  ;progressive-versus-seqential  distinction
       ILEAVE         b1  ;interleaving  of  multiple  bit-planes
       SMID           b0  ;indexed  over  stripe  is  in  middle
    Options  1      ;option  byte
       -                b7  ;dummy  0
       LRLTWO           b6  ;lowest  resolution-layer  two  line  template
       VLENGTH          b5  ;NEWLEN(new  vertical  dimmension)marker  enable
       TPDON            b4  ;differential-layer  TP  enable
       TPBON            b3  ;lowest-resolution-layer  TP  enable
       DPON             b2  ;DP  enable
       DPPRIV           b1  ;private  DP  table
       DPLAST           b0  ;DP  table  last  is  to  be  reused
    DPTABLE   0/1728  ;private  DP  table
                                   (it  is  present only  if  DPON=1,  DPPRIV=1,  DPLAST=0)
  B I D          ;bi-level  Image  Data((① ②) x N)
      ① Flloating  Marker  Segments(ⓐ~ⓒ)
       ⓐ AT  move  marker
          ESC          1        ;FFh
          ATMOVE       1        ;06h
          YAT          4        ;line  in  which  an  AT  switch  is  to  be  made
          τX           1        ;horizontal  offset  of  the  AT  pixel
          τY           1        ;vertical    offset  of  the  AT  pixel

       ⓑ new-length  marker
          ESC          1        ;FFh
          NEWLEN       1        ;05h
          YD           4        ;new  YD

       ⓒ comment  marker
          ESC          1        ;FFh
          COMMENT      1        ;07h
          LC           4        ;length  in  bytes  of  private  comment
          comment      LC       ;contents  of  comment
      ② SDE     ;Stripe  Data  Entry                (Within the frame:  LSI support range)
          +--------------------------------------------------------------------+
          | PSCD                        ;Protected  Stripe  Coded  Data        |
          |                                  =byte  stuffed  SCD(Stripe  Code  Data) |
          | ESC                    1    ;FFh                                   |
          | SDNORM/SDRST           1    ;normal  terminate(02h)                |
          |                             ;/reset  "state"  for  next  SDE(03h)  |
          +--------------------------------------------------------------------+
     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     abort  BID  marker
          ESC          1        ;FFh
          ABORT        1        ;04h

     reserved  marker
          ESC          1        ;FFh
          RESERVE      1        ;01h
```

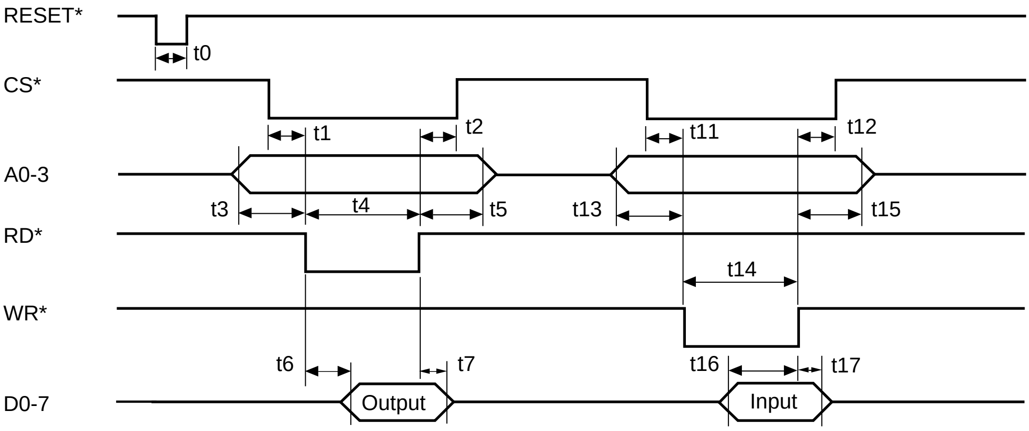
[Appendix A.2] JBIG Probability Estimation Table

ST	LSZ	NLPS	NMPS	SWTCH	ST	LSZ	NLPS	NMPS	SWTCH
0	0x5ald	1	1	1	57	0x01a4	55	58	0
1	0x2586	14	2	0	58	0x0160	56	59	0
2	0x1114	16	3	0	59	0x0125	57	60	0
3	0x080b	18	4	0	60	0x00f6	58	61	0
4	0x03d8	20	5	0	61	0x00cb	59	62	0
5	0x01da	23	6	0	62	0x00ab	61	63	0
6	0x00e5	25	7	0	63	0x008f	61	32	0
7	0x006f	28	8	0	64	0x5b12	65	65	1
8	0x0036	30	9	0	65	0x4d04	80	66	0
9	0x001a	33	10	0	66	0x412c	81	67	0
10	0x000d	35	11	0	67	0x37d8	82	68	0
11	0x0006	9	12	0	68	0x2fe8	83	69	0
12	0x0003	10	13	0	69	0x293c	84	70	0
13	0x0001	12	13	0	70	0x2379	86	71	0
14	0x5a7f	15	15	1	71	0x1edf	87	72	0
15	0x3f25	36	16	0	72	0x1aa9	87	73	0
16	0x2cf2	38	17	0	73	0x174e	72	74	0
17	0x207c	39	18	0	74	0x1424	72	75	0
18	0x17b9	40	19	0	75	0x119c	74	76	0
19	0x1182	42	20	0	76	0x0f6b	74	77	0
20	0x0cef	43	21	0	77	0x0d51	75	78	0
21	0x09a1	45	22	0	78	0x0bb6	77	79	0
22	0x072f	46	23	0	79	0x0a40	77	48	0
23	0x055c	48	24	0	80	0x5832	80	81	1
24	0x0406	49	25	0	81	0x4d1c	88	82	0
25	0x0303	51	26	0	82	0x438e	89	83	0
26	0x0240	52	27	0	83	0x3bdd	90	84	0
27	0x01b1	54	28	0	84	0x34ee	91	85	0
28	0x0144	56	29	0	85	0x2eae	92	86	0
29	0x00f5	57	30	0	86	0x299a	93	87	0
30	0x00b7	59	31	0	87	0x2516	86	71	0
31	0x008a	60	32	0	88	0x5570	88	89	1
32	0x0068	62	33	0	89	0x4ca9	95	90	0
33	0x004e	63	34	0	90	0x44d9	96	91	0
34	0x003b	32	35	0	91	0x3e22	97	92	0
35	0x002c	33	9	0	92	0x3824	99	93	0
36	0x5ae1	37	37	1	93	0x32b4	99	94	0
37	0x484c	64	38	0	94	0x2e17	93	86	0
38	0x3a0d	65	39	0	95	0x56a8	95	96	1
39	0x2ef1	67	40	0	96	0x4f46	101	97	0
40	0x261f	68	41	0	97	0x47e5	102	98	0
41	0x1f33	69	42	0	98	0x41cf	103	99	0
42	0x19a8	70	43	0	99	0x3c3d	104	100	0
43	0x1518	72	44	0	100	0x375e	99	93	0
44	0x1177	73	45	0	101	0x5231	105	102	0
45	0x0e74	74	46	0	102	0x4c0f	106	103	0
46	0x0bfb	75	47	0	103	0x4639	107	104	0
47	0x09f8	77	48	0	104	0x415e	103	99	0
48	0x0861	78	49	0	105	0x5627	105	106	1
49	0x0706	79	50	0	106	0x50e7	108	107	0
50	0x05cd	48	51	0	107	0x4b85	109	103	0
51	0x04de	50	52	0	108	0x5597	110	109	0
52	0x040f	50	53	0	109	0x504f	111	107	0
53	0x0363	51	54	0	110	0x5a10	110	111	1
54	0x02d4	52	55	0	111	0x5522	112	109	0
55	0x025c	53	56	0	112	0x59eb	112	111	1
56	0x01f8	54	57	0					

[Appendix B] Timing Characteristics

Conditions:V_{DD}=5V±5%
C=50pF
Ta=0-70°C

1. Host bus I/F



2. Code data I/F

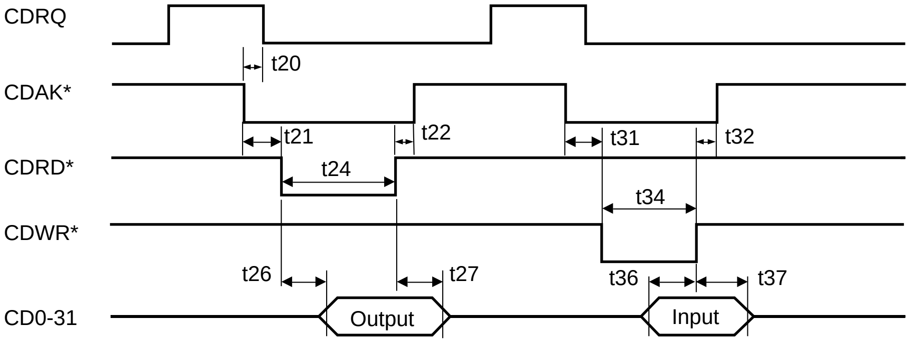


Table B. 1 Host Bus I/F Timing Characteristics

(Unit: ns)

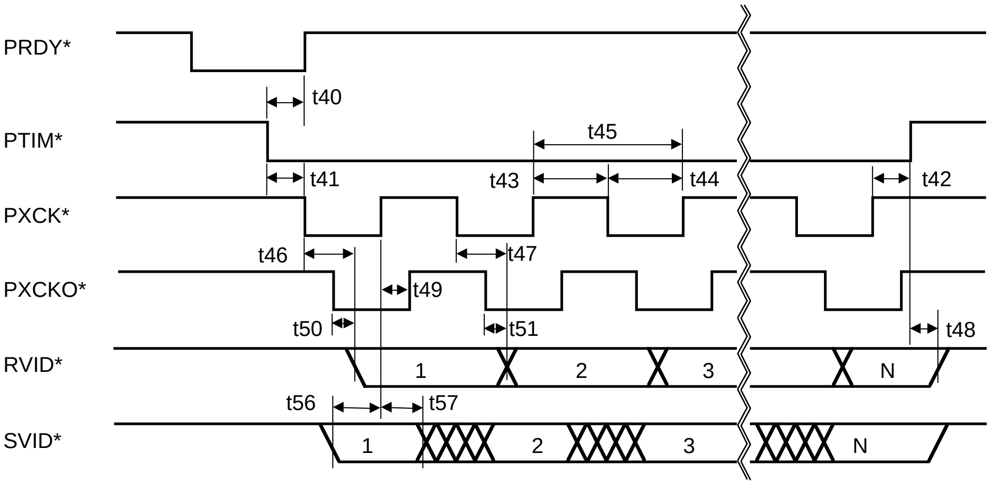
Abbreviation	Parameter	Timing conditions		
		Min	Typ	Max
t0	RESET* assert time	100	-	-
t1	CS* setup time to RD* assert	15	-	-
t2	CS* hold time to RD* negate	15	-	-
t3	A0-3 setup time to RD* assert	15	-	-
t4	RD* assert time	20	-	-
t5	A0-3 hold time to RD* negate	15	-	-
t6	D0-7 output determination time to RD* assert	0	-	20
t7	D0-7 output hold time to RD* negate	0	-	20
t11	CS* setup time to WR* assert	15	-	-
t12	CS* hold time to WR* negate	15	-	-
t13	A0-3 setup time to WR* assert	15	-	-
t14	WR* assert time	15	-	-
t15	A0-3 hold time to WR* negate	15	-	-
t16	D0-7 input setup time to WR* negate	20	-	-
t17	D0-7 input hold time to WR* negate	5	-	-

Table B. 2 Timing Characteristics of Code Data Bus I/F

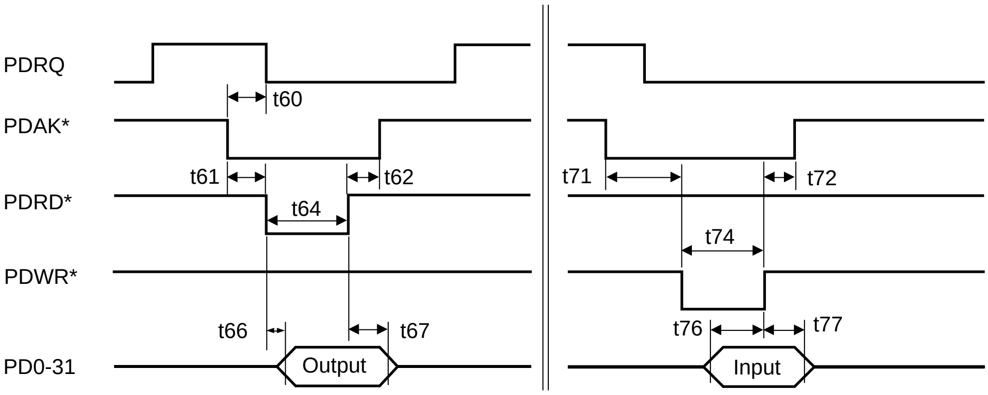
Abbreviation	Parameter	Timing conditions		
		Min	Typ	Max
t20	CDRQ negate time to CDAK* assert	-	-	15
t21	CDAK* setup time to CDRD* assert	15	-	-
t22	CDAK* hold time to CDRD* negate	15	-	-
t24	CDRD* assert time	20	-	-
t26	CD0-31 output determination time to CDRD* assert	0	-	20
t27	CD0-31 output hold time to CDRD* negate	0	-	20
t31	CDAK* setup time to CDWR* assert	15	-	-
t32	CDAK* hold time to CDWR* negate	15	-	-
t34	CDWR* assert time	15	-	-
t36	CD0-31 input setup time to CDWR* negate	15	-	-
t37	CD0-31 input hold time to CDWR* negate	5	-	-

3. Image data I/F

(1) Serial image data I/F



(2) Parallel image data I/F



4. Master clock input frequency (LSI operating frequency)

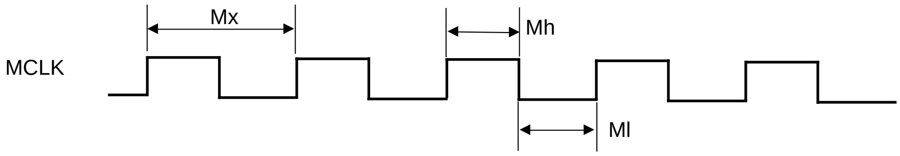


Table B. 3 Timing Characteristics of Image Data I/F

(Unit: ns)

Abbreviation	Parameter	Timing conditions		
		Min	Typ	Max
t40	PRDY* negate time to PTIM* assert	-	-	20
t41	PTIM* setup time to PXCK* fall	15	-	-
t42	PTIM* hold time to PXCK* rise	15	-	-
t43	PXCK* high time	10	-	-
t44	PXCK* low time	10	-	-
t45	PXCK* cycle	25	-	-
t46	RVID* output determination time to PXCK* fall	-	-	20
t47	RVID* output change time to PXCK* fall	-	-	20
t48	RVID* negate time to PTIM* negate	0	-	-
t49	PXCKO* delay time to PXCK*	-	-	10
t50	RVID* output determination time to PXCKO* fall	-	-	12
t51	RVID* output change time to PXCKO* fall	-	-	12
t56	SVID* setup time to PXCK* rise	10	-	-
t57	SVID* hold time to PXCK* rise	10	-	-
t60	PDRQ negate time to PDAK* assert	-	-	15
t61	PDAK* setup time to PDRD* assert	15	-	-
t62	PDAK* hold time to PDRD* negate	15	-	-
t64	PDRD* assert time	20	-	-
t66	PD0-31 output determination time to PDRD* assert	0	-	20
t67	PD0-31 output hold time to PDRD* negate	0	-	20
t71	PDAK* setup time to PDWR* assert	15	-	-
t72	PDAK* hold time to PDWR* negate	15	-	-
t74	PDWR* assert time	15	-	-
t76	PD0-31 input setup time to PDWR* negate	15	-	-
t77	PD0-31 input hold time to PDWR* negate	5	-	-

Table B. 4 Master Clock Frequencies

(Unit: ns)

Parameter	Timing conditions			Max frequency
	Min	Typ	Max	
MCLK cycle (Mx)	25	-	-	40MHz
MCLK high level time (Mh)	10	-	-	
MCLK low level time (Ml)	10	-	-	