TOSHIBA

32bit TX System RISC
TX19 family

TMP1942FDU/XBG

Rev1.0 March 29, 2007

32-Bit RISC Microprocessor TX19 Family

TMP1942FDU/FDXBG

1. Features

The TX19 is a family of high-performance 32-bit microprocessors that offers the speed of a 32-bit RISC solution with the added advantage of a significantly reduced code size of a 16-bit architecture. The instruction set of the TX19 includes as a subset the 32-bit instructions of the TX39, which is based on the MIPS R3000A™ architecture. Additionally, the TX19 supports the MIPS16 Application-Specific Extensions (ASE) for improved code density.

The TMP1942 is built on a TX19 core processor and a selection of intelligent peripherals. The TMP1942 is suitable for low-voltage and low-power applications.

Features of the TMP1942 include the following:

RESTRICTIONS ON PRODUCT USE

070122EBP

- The information contained herein is subject to change without notice. 021023_D
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
  In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the “Handling Guide for Semiconductor Devices,” or “TOSHIBA Semiconductor Reliability Handbook” etc. 021023_A
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury (“Unintended Usage”). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer’s own risk. 021023_B
- The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations. 060106_Q
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patents or other rights of TOSHIBA or the third parties. 070122_C
- The products described in this document are subject to foreign exchange and foreign trade control laws. 060925_E
- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions. 030619_S

(1) TX19 core processor

1) Two instruction set architecture (ISA) modes: 16-bit ISA for code efficiency and 32-bit ISA for performance.
   - The 16-bit ISA is object-code compatible with the code-efficient MIPS16™ ASE.
   - The 32-bit ISA is object-code compatible with the high-performance TX39 family.
2) High performance combined with low power consumption

   - High performance
   - Single clock cycle execution for most instructions
   - 3-operand arithmetic instructions for high instruction throughput
   - 5-stage pipeline
   - On-chip high-speed memory
   - DSP function: Executes 32-bit x 32-bit multiply operations in a single clock cycle.

   - Low power consumption
   - Optimized design using a low-power cell library
   - Programmable standby modes in which processor clocks are stopped
3) Fast interrupt response suitable for real-time control
   - Distinct starting locations for each interrupt service routine
   - Automatically generated vectors for each interrupt source
   - Automatic updates of the interrupt mask level

(2) On-chip RAM:

TMP1942FDU/FDXB 20 KB

On-chip FROM

TMP1942FDU/FDXB 512 KB

ROM correction function (8 words x 4 blocks)

(In the flash-version product, only the registers exist and no correction is performed.)

(3) External memory expansion
- 16-Mbyte off-chip address space for code and data
- External bus interface with dynamic bus sizing for 8-bit and 16-bit data ports

(4) 4-channel DMA controller
- Interrupt- or software-triggered

(5) 6-channel 8-bit PWM timer
- 12-channel 8-bit interval timer
- 6-channel 16-bit interval timer
- 6-channel 8-bit PPG output

(6) 14-channel 16-bit timer
- Two channels support two-phase input pulse counter mode.

(7) 1-channel RTC timer

(8) 5-channel general-purpose serial interface
- Supports both UART and synchronous transfer modes.

(9) 1-channel serial bus interface

- Either I^2C bus mode or clock-synchronous mode can be selected.

(10) 16-channel 10-bit A/D converter (with internal sample/hold)

- Conversion time: 2 µs (throughput), 4 µs to 5 µs (latency)

(11) 3-channel 10-bit D/A converter

(12) Watchdog timer

(13) 4-channel chip select/wait controller

(14) Interrupt sources

- 4 CPU interrupts: software interrupt instruction
- 45 internal interrupts: 7 priority levels, with the exception of the watchdog timer interrupt
- 29 external interrupts: 7 priority levels, with the exception of the NMI interrupt
  Of the 29, 14 are used for key-on wakeup and share an interrupt vector. INTB, INTC, INTD, and INTE are extended interrupts which share an interrupt vector and are distinguished by flags. Therefore, a total of 13 interrupt sources are available.

(15) 108-pin input/output ports

(16) Three standby (HALT) modes

- IDLE, SLEEP, STOP

(17) Dual clocks

- RTC clock: Low-speed clock (32.768 kHz)

(18) Clock generator

- On-chip PLL (x4)
- Clock gear: Divides the high-speed clock to 1/2, 1/4, or 1/8.

(19) Operating voltage range: 2.7 V to 3.6 V

2.7 V to 3.6 V or 4.5 V to 5.25 V for Port C and Port F supporting 5-V operation

(20) Operating frequency

- 32 MHz ($Vcc \geq 3.0$ V) (Interleave mode)
- 28 ($Vcc \geq 2.7$ V)

(21) Packages

- 144-pin QFP (16 mm ×16 mm × 1.4 mm thick, 0.4 mm pitch): TMP1942FDU
- 177-pin CSP (13 mm × 13 mm × 1.4 mm thick, 0.8 mm pitch): TMP1942FDXBG

Note: The TMP1942FDXBG packaged in a 177-pin CSP is under development.

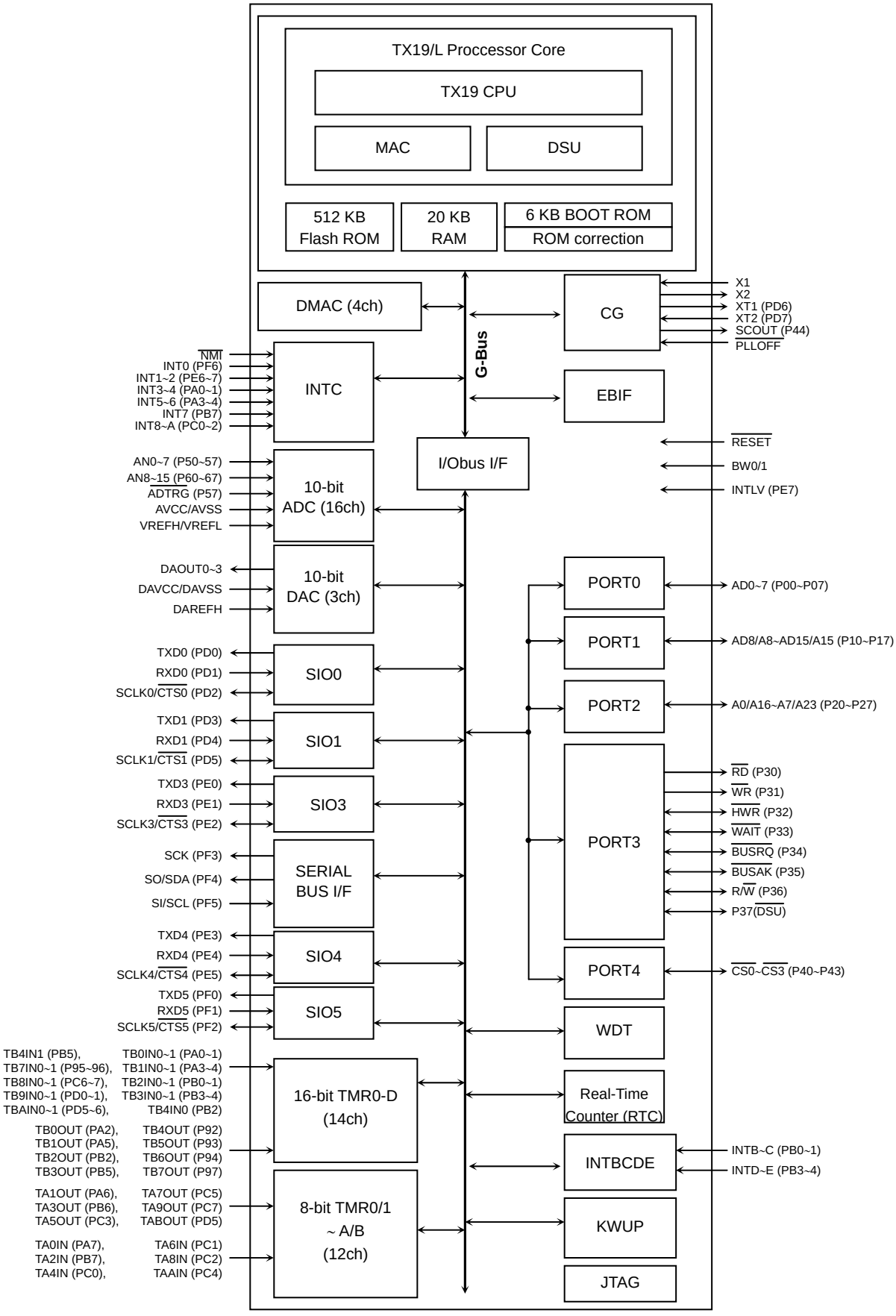


Figure1.1.1 TMP1942 Block Diagram

2. Signal Descriptions

This section contains pin assignments for the TMP1942FD as well as brief descriptions of the TMP1942FD input and output signals.

2.1 Pin Assignments

Table 2.1.1 lists the pin assignments for the TMP1942FD in a 144-pin package.

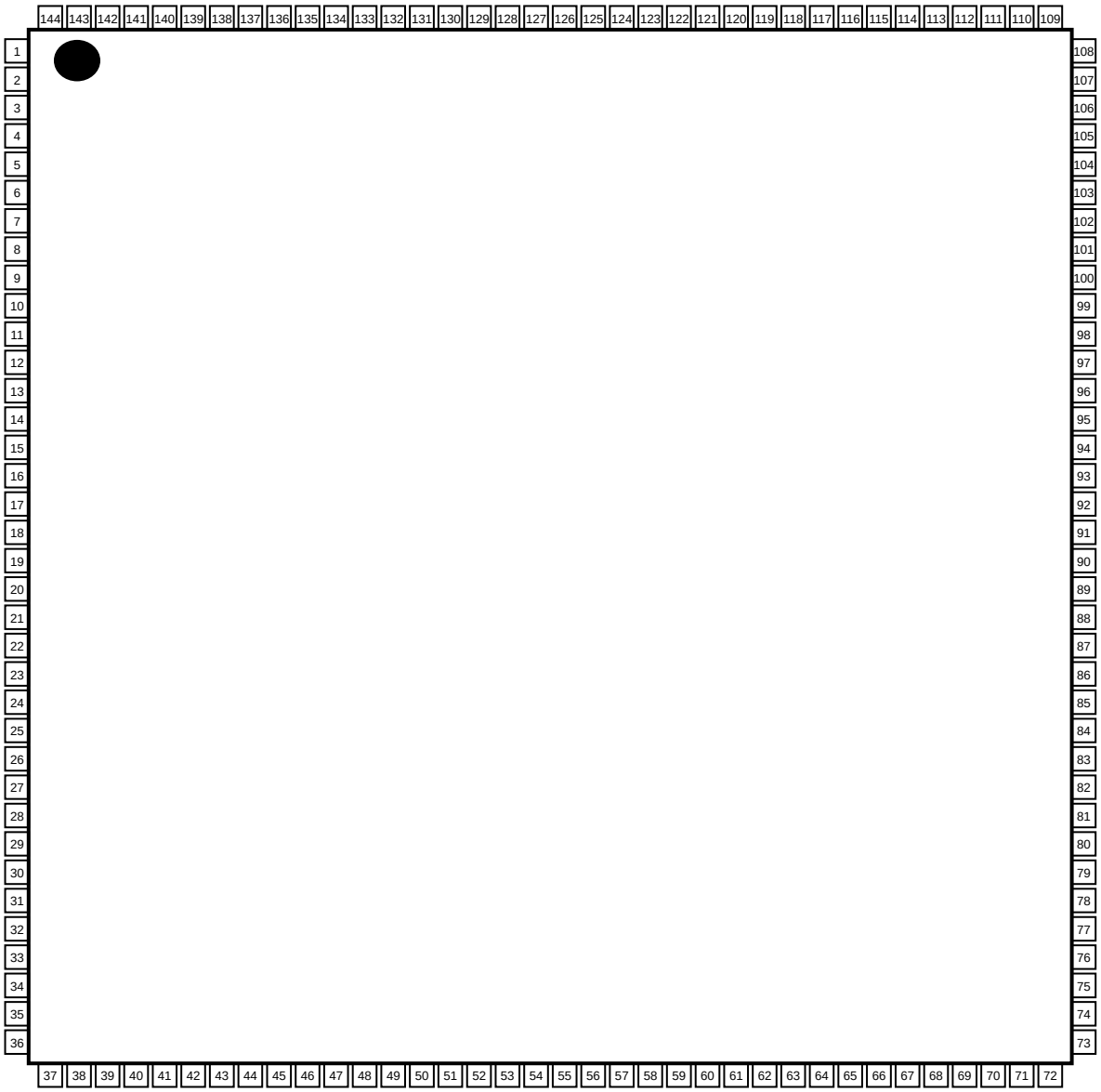


Table 2.1.1 144-Pin LQFP Pin Assignments

Pin #	Pin Function	Pin #.	Pin Function	Pin #	Pin Function	Pin #	Pin Function
1	VREFH	37	P11/AD9/A9	73	P90/KEY8/DCLK	109	CVCC
2	VREFL	38	P12/AD10/A10	74	P91/KEY9/PCST2	110	X2
3	P50/AN0	39	P13/AD11/A11	75	P92/TB4OUT/PCST1	111	CVSS
4	P51/AN1	40	P14/AD12/A12	76	P93/TB5OUT/PCST0	112	X1
5	P52/AN2	41	P15/AD13/A13	77	P94/TB6OUT/SDAO/TPC	113	TEST1
6	P53/AN3	42	P16/AD14/A14	78	P95/TB7IN0/ $\overline{\text{DBGE}}$	114	$\overline{\text{RESET}}$
7	DAVCC	43	P17/AD15/A15	79	P96/TB7IN1/ $\overline{\text{DINT}}$	115	PD6/XT1
8	DAVSS	44	P20/A0/A16	80	P97/TB7OUT/ $\overline{\text{DRESET}}$	116	PD7/XT2
9	DAREH	45	P21/A1/A17	81	DVCC3	117	$\overline{\text{NMI}}$
10	DAOUT0	46	P22/A2/A18	82	PA0/TB0IN0/INT3	118	BW0
11	DAOUT1	47	P23/A3/A19	83	PA1/TB0IN1/INT4	119	PB0/TB2IN0/INTB
12	DAOUT2	48	P24/A4/A20	84	PA2/TB0OUT	120	PB1/TB2IN1/INTC
13	P54/AN4	49	P25/A5/A21	85	PA3/TB1IN0/INT5	121	PB2/TB2OUT/TB4IN0
14	P55/AN5	50	P26/A6/A22	86	PA4/TB1IN1/INT6	122	PB3/TB3IN0/INTD
15	P56/AN6	51	P27/A7/A23	87	PA5/TB1OUT	123	PB4/TB3IN1/INTE
16	P57/AN7/ $\overline{\text{ADTRG}}$	52	TEST0	88	PA6/TA1OUT	124	PB5/TB3OUT/TB4IN1
17	P60/AN8/KEY0	53	$\overline{\text{PLLOFF}}$	89	PA7/TA0IN/KEYA	125	PB6/TA3OUT
18	DVSS	54	FVSS	90	DVSS	126	DVSS
19	P61/AN9/KEY1	55	ALE	91	RSTPUP	127	DVCC3
20	P62/AN10/KEY2	56	DVCC3	92	PC0/TA4IN/INT8	128	PB7/TA2IN/INT7/KEYB
21	P63/AN11/KEY3	57	BW1	93	PC1/TA6IN/INT9	129	PD0/TXD0/TB9IN0
22	P64/AN12/KEY4	58	P30/ $\overline{\text{RD}}$	94	PC2/TA8IN/INTA	130	PD1/RXD0/TB9IN1
23	P65/AN13/KEY5	59	P31/ $\overline{\text{WR}}$	95	PC3/TA5OUT	131	PD2/SCLK0/ $\overline{\text{CTS0}}$
24	P66/AN14/KEY6	60	P32/ $\overline{\text{HWR}}$	96	PC4/TAAIN	132	PD3/TXD1/TBAIN0
25	P67/AN15/KEY7	61	P33/ $\overline{\text{WAIT}}$	97	PC5/TA7OUT	133	PD4/RXD1/TBAIN1
26	DVCC3	62	P34/ $\overline{\text{BUSRQ}}$	98	PC6/TB8IN0/KEYC	134	PD5/SCLK1/ $\overline{\text{CTS1}}$ /TABOUT
27	P00/AD0	63	P35/ $\overline{\text{BUSAK}}$	99	PC7/TB8IN1/TA9OUT	135	PE0/TXD3
28	P01/AD1	64	P36/ R / $\overline{\text{W}}$	100	DVCC52	136	PE1/RXD3
29	P02/AD2	65	P37/ $\overline{\text{DSU}}$	101	PF0/TXD5	137	PE2/SCLK3/ $\overline{\text{CTS3}}$
30	P03/AD3	66	DVSS	102	PF1/RXD5/KEYD	138	PE3/TXD4
31	P04/AD4	67	FVCC	103	PF2/SCLK5/ $\overline{\text{CTS5}}$	139	PE4/RXD4
32	P05/AD5	68	P40/ $\overline{\text{CS0}}$	104	PF3/SCK	140	PE5/SCLK4/ $\overline{\text{CTS4}}$
33	P06/AD6	69	P41/ $\overline{\text{CS1}}$	105	PF4/SO/SDA	141	PE6/INT1/ $\overline{\text{BOOT}}$
34	P07/AD7	70	P42/ $\overline{\text{CS2}}$	106	PF5/SI/SCL	142	PE7/INT2/INTLV
35	DVSS	71	P43/ $\overline{\text{CS3}}$	107	PF6/INT0	143	AVCC
36	P10/AD8/A8	72	P44/SCOUT	108	DVCC51	144	AVSS

Table 2.2.2 lists the pin assignments for the TMP1942FD in a 177-pin package.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
E1	E2	E3	E4								E12	E13	E14	E15
F1	F2	F3	F4								F12	F13	F14	F15
G1	G2	G3	G4								G12	G13	G14	G15
H1	H2	H3	H4								H12	H13	H14	H15
J1	J2	J3	J4								J12	J13	J14	J15
K1	K2	K3	K4								K12	K13	K14	K15
L1	L2	L3	L4								L12	L13	L14	L15
M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14	N15
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15

Table 2.1.2 177-Pin CSP Pin Assignments

Pin #	Pin Function	Pin #	Pin Function	Pin #	Pin Function	Pin #	Pin Function
A1	VREFL	D1	P50/AN0	H13	NC	N4	P16/AD14/A14
A2	AVSS	D2	DAVSS	H14	NC	N5	P21/A1/A17
A3	AVCC	D3	P52/AN2	H15	DVSS	N6	P25/A5/A21
A4	PE7/INT2/INTLV	D4	P51/AN1	J1	P67/AN15/KEY7	N7	DVSS
A5	PE3/TXD4	D5	PE0/TXD3	J2	P65/AN13/KEY5	N8	TEST0
A6	TCK (JTAG)	D6	PD3/TXD1/TBAIN0	J3	P66/AN14/KEY6	N9	P30/ $\overline{RD}$
A7	PD2/SCLK0/ $\overline{CTS0}$	D7	PB7/TA2IN/INT7/ KEYB	J4	P64/AN12/KEY4	N10	P32/ $\overline{HWR}$
A8	PB5/TB3OUT/ TB4IN1	D8	DVSS	J12	PA6/TA1OUT	N11	P37/ $\overline{DSU}$
A9	PB1/TB2IN1/INTC	D9	PB2/TB2OUT/ TB4IN0	J13	PA7/TA0IN/KEYA	N12	DVSS
A10	PD7/TX2	D10	$\overline{NMI}$	J14	NC	N13	P41/ $\overline{CS1}$
A11	PD6/TX1	D11	NC	J15	PA5/TB1OUT	N14	P91/KEY9
A12	X1	D12	NC	K1	P01/AD1	N15	NC
A13	X2	D13	PF1/RXD5/KEYD	K2	DVCC3	P1	NC
A14	CVCC	D14	PF3/SCK	K3	NC	P2	P10/AD8/A8
A15	NC	D15	PF6/INT0	K4	NC	P3	P12/AD10/A10
B1	NC	E1	DAVCC	K12	PA2/TB0OUT	P4	P20/A0/A16
B2	NC	E2	DAOUT0	K13	PA3/TB1IN0/INT5	P5	P22/A2/A18
B3	PE6/INT1	E3	DAREFH	K14	PA4/TB1IN1/INT6	P6	P26/A6/A22
B4	PE4/RXD4	E4	P53/AN3	K15	PA1/TB0IN1/INT4	P7	TDO (JTAG)
B5	$\overline{TRST}$ (JTAG)	E5	NC(not bonded)	L1	P04/AD4	P8	ALE
B6	PD5/SCLK1/ $\overline{CTS1}$ / TABOUT	E12	PC6/TB8IN0/KEYC	L2	P02/AD2	P9	BW1
B7	PD0/TXD0/TB9IN0	E13	DVCC52	L3	TMS (JTAG)	P10	P33/ $\overline{WAIT}$
B8	DVCC3	E14	PF0/TXD5	L4	P00/AD0	P11	TDI (JTAG)
B9	PB4/TB3IN1/INTE	E15	PF2/SCLK5/ $\overline{CTS5}$	L12	P97/TB7OUT	P12	P40/ $\overline{CS0}$
B10	PB0/TB2IN0/INTB	F1	DAOUT1	L13	DVCC3	P13	P42/ $\overline{CS2}$
B11	NC	F2	P55/AN5	L14	PA0/TB0IN0/INT3	P14	P44/SCOUT
B12	$\overline{RESET}$	F3	P54/AN4	L15	P96/TB7IN1	P15	NC
B13	CVSS	F4	DAOUT2	M1	P07/AD7	R1	P11/AD9/A9
B14	DVCC51	F12	PC2/TA8IN/INTA	M2	P05/AD5	R2	NC
B15	NC	F13	PC4/TAAIN	M3	P03/AD3	R3	NC
C1	VREFH	F14	PC5/TA7OUT	M4	P14/AD12/A12	R4	P13/AD11/A11
C2	NC	F15	PC7/TB8IN1/ TA9OUT	M5	P15/AD13/A13	R5	P17/AD15/A15
C3	PE5/SCLK4/ $\overline{CTS4}$	G1	P56/AN6	M6	P24/A4/A20	R6	P23/A3/A19
C4	PE2/SCLK3/ $\overline{CTS3}$	G2	P61/AN9/KEY1	M7	$\overline{PLLOFF}$	R7	P27/A7/A23
C5	PE1/RXD3	G3	NC	M8	NC	R8	NC
C6	PD4/RXD1/TBAIN1	G4	P60/AN8/KEY0	M9	DVCC3	R9	P31/ $\overline{WR}$
C7	PD1/RXD0/TB9IN1	G12	PC0/TA4IN/INT8	M10	P34/ $\overline{BUSRQ}$	R10	P35/ $\overline{BUSAK}$
C8	PB6/TA3OUT	G13	PC1/TA6IN/INT9	M11	P36/ R/ $\overline{W}$	R11	DVCC3
C9	PB3/TB3IN0/INTD	G14	NC	M12	P93/TB5OUT	R12	P43/ $\overline{CS3}$
C10	BW0	G15	PC3/TA5OUT	M13	P94/TB6OUT	R13	NC
C11	NC	H1	DVSS	M14	P95/TB7IN0	R14	NC
C12	TEST1	H2	P63/AN11/KEY3	M15	P92/TB4OUT	R15	P90/KEY8
C13	PF4/SO/SDA	H3	P57/AN7/ $\overline{ADTRG}$	N1	NC		
C14	PF5/SI/SCL	H4	P62/AN10/KEY2	N2	DVSS		
C15	NC	H12	RSTPUP	N3	P06/AD6		

2.2 Pin Usage Information

Table 2.2.1 lists the input and output pins of the TMP1942FD, including alternate pin names and functions for multi-function pins.

Table 2.2.1 Pin Names and Functions

Pin Name	# of Pins	Type	Function
P00-P07 AD0-AD7	8	Input/Output Input/Output	Port 0: Individually programmable as input or output Address (Lower): Bits0-7 of the address/data bus
P10-P17 AD8-AD15 A8-A15	8	Input/Output Input/Output Output	Port 1: Individually programmable as input or output Address/Data (Upper): Bits 8-15 of the address/data bus Address: Bits 8-15 of the address bus
P20-P27 A0-A7 A16-A23	8	Input/Output Output Output	Port 2: Individually programmable as input or output Address: Bits 0-7 of the address bus Address: Bits 16-23 of the address bus
P30 $\overline{RD}$	1	Output Output	Port 30: Output-only Read Strobe: Asserted during a read operation from an external memory device
P31 $\overline{WR}$	1	Output Output	Port 31: Output-only Write Strobe: Asserted during a write operation on D0-D7
P32 $\overline{HWR}$	1	Input/Output Output	Port 32: Programmable as input or output (with internal pull-up resistor) Higher Write Strobe: Asserted during a write operation on D8-D15
P33 $\overline{WAIT}$	1	Input/Output Input	Port 33: Programmable as input or output (with internal pull-up resistor) Wait: Causes the CPU to suspend external bus activity
P34 $\overline{BUSRQ}$	1	Input/Output Input	Port 34: Programmable as input or output (with internal pull-up resistor) Bus Request: Asserted by an external bus master to request bus mastership
P35 $\overline{BUSAK}$	1	Input/Output Output	Port 35: Programmable as input or output (with internal pull-up resistor) Bus Acknowledge: Indicates that the CPU has relinquished the bus in response to $\overline{BUSRQ}$
P36 R / $\overline{W}$	1	Input/Output Output	Port 36: Programmable as input or output (with internal pull-up resistor) Read/Write: Indicates the direction of data transfer on the bus: 1 = read or dummy cycle, 0 = write cycle
P37 $\overline{DSU}$	1	Input/Output Input	Port 37: Programmable as input or output (with internal pull-up resistor) DSU enable signal: DSU is enabled when this pin is sampled low at the rising edge of $\overline{RESET}$. To enable DSU, this pin should be pulled down. (Flash version only)
P40 $\overline{CS0}$	1	Input/Output Output	Port 40: Programmable as input or output (with internal pull-up resistor) Chip Select 0: Asserted low to enable external devices at programmed addresses
P41 $\overline{CS1}$	1	Input/Output Output	Port 41: Programmable as input or output (with internal pull-up resistor) Chip Select 1: Asserted low to enable external devices at programmed addresses
P42 $\overline{CS2}$	1	Input/Output Output	Port 42: Programmable as input or output (with internal pull-up resistor) Chip Select 2: Asserted low to enable external devices at programmed addresses
P43 $\overline{CS3}$	1	Input/Output Output	Port 43: Programmable as input or output (with internal pull-up resistor) Chip Select 3: Asserted low to enable external devices at programmed addresses
P44 SCOUT	1	Input/Output Output	Port 44: Programmable as input or output System Clock Output: Drives out a clock signal at the same frequency as the CPU clock (high-speed or low-speed)
P50-P57 AN0-AN7 $\overline{ADTRG}$	8	Input Input Input	Port 5: Input-only Analog Input: Input to the on-chip A/D converter External A/D trigger pin for starting an A/D conversion (multiplexed with P57)
P60-P67 AN8-AN15 KEY0-KEY7	1	Input/Output Input Input	Port 6: Input-only Analog Input: Input to the on-chip A/D converter Key-On Wakeup Input (dynamic pull-up selectable) (with internal pull-up resistor)
P90 DSU (DCLK) KEY8	1	Input/Output Output Input	Port 90: Programmable as input or output DSU pin Key-On Wakeup Input (dynamic pull-up selectable) (with internal pull-up resistor)
P91 DSU (PCST2) KEY9	1	Input/Output Output Input	Port 91: Programmable as input or output DSU pin Key-On Wakeup Input (dynamic pull-up selectable) (with internal pull-up resistor)

Pin Name	# of Pins	Type	Function
P92 DSU (PCST1) TB40UT	1	Input/Output Output Output	Port 92: Programmable as input or output DSU pin 16-Bit Timer 4 Output: Output from 16-bit Timer 4
P93 DSU (PCST0) TB5OUT	1	Input/Output Output Output	Port 93: Programmable as input or output DSU pin 16-Bit Timer 5 Output: Output from 16-bit Timer 5
P94 DSU (SDAO/TPC) TB6OUT	1	Input/Output Output Output	Port 94: Programmable as input or output DSU pin 16-Bit Timer 6 Output: Output from 16-bit Timer 6
P95 DSU ($\overline{\text{DBGE}}$) TB7IN0	1	Input/Output Input	Port 95: Programmable as input or output DSU pin 16-Bit Timer 7 Input 0: Count/capture trigger input to 16-bit Timer 7
P96 DSU ($\overline{\text{DINT}}$) TB7IN1	1	Input/Output Input	Port 96: Programmable as input or output DSU pin 16-Bit Timer 7 Input 0: Capture trigger input to 16-bit Timer 7
P97 DSU($\overline{\text{DRESET}}$) TB7OUT	1	Input/Output Input Output	Port 97: Programmable as input or output DSU pin 16-Bit Timer 7 Output: Output from 16-bit Timer 7
PA0 TB0IN0 INT3	1	Input/Output Input Input	Port A0: Programmable as input or output 16-Bit Timer 0 Input 0: Count/capture trigger input to 16-bit Timer 0 Interrupt Request 3: Programmable as high-level, low-level, rising-edge or falling-edge sensitive
PA1 TB0IN1 INT4	1	Input/Output Input Input	Port A1: Programmable as input or output 16-Bit Timer 0 Input 1: Capture trigger input to 16-bit Timer 0 Interrupt Request 4: Programmable to be high-level, low-level, rising-edge or falling edge sensitive
PA2 TB0OUT	1	Input/Output Output	Port A2: Programmable as input or output 16-Bit Timer 0 Output: Output from 16-bit Timer 0
PA3 TB1IN0 INT5	1	Input/Output Input Input	Port A3: Programmable as input or output 16-Bit Timer 1 Input 0: Count/capture trigger input to 16-bit Timer 1 Interrupt Request 5: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PA4 TB1IN1 INT6	1	Input/Output Input Input	Port A4: Programmable as input or output 16-Bit Timer 1 Input 1: Capture trigger input to 16-bit Timer 1 Interrupt Request 6: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PA5 TB1OUT	1	Input/Output Output	Port A5: Programmable as input or output 16-Bit Timer 1 Output: Output from 16-bit Timer 1
PA6 TA1OUT	1	Input/Output Output	Port A6: Programmable as input or output 8-Bit Timer 01 Output: Output from 8-bit Timer 0 or Timer 1
PA7 TA0IN KEYA	1	Input/Output Input Input	Port A7: Programmable as input or output 8-Bit Timer 0 Input: Input to 8-bit Timer 0 Key-On Wakeup Input (dynamic pull-up selectable) (with internal pull-up resistor)
PB0 TB2IN0 INTB	1	Input/Output Input Input	Port B0: Programmable as input or output 16-Bit Timer 2 Input 0: Count/capture trigger input to 16-bit Timer 2 or two-phase input pulse counter input Interrupt Request 7: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PB1 TB2IN1 INTC	1	Input/Output Input Input	Port B1: Programmable as input or output 16-Bit Timer 2 Input 1: Capture trigger input to 16-bit Timer 2 or two-phase input pulse counter input Interrupt Request 8: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PB2 TB2OUT TB4IN0	1	Input/Output Output Input	Port B2: Programmable as input or output 16-Bit Timer 2 Output: Output from 16-bit Timer 2 16-Bit Timer 4 Input 0: Count/capture trigger input to 16-bit Timer 4
PB3 TB3IN0 INTD	1	Input/Output Input Input	Port B3: Programmable as input or output 16-Bit Timer 3 Input 0: Count/capture trigger input to 16-bit Timer 3 or two-phase input pulse counter input Interrupt Request B: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PB4 TB3IN1 INTE	1	Input/Output Input Input	Port B4: Programmable as input or output 16-Bit Timer 3 Input 1: Capture trigger input to 16-bit Timer 3 or two-phase input pulse counter input Interrupt Request C: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive

Pin Name	# of Pins	Type	Function
PB5	1	Input/Output	Port B5: Programmable as input or output
TB3OUT		Output	16-Bit Timer 3 Output: Output from 16-Bit Timer 3
TB4IN1		Input	16-Bit Timer 4 Input 1: Capture trigger input to 16-bit Timer 4
PB6	1	Input/Output	Port B6: Programmable as input or output
TA3OUT		Output	8-Bit Timer 23 Output: Output from 8-bit Timer 2 or Timer 3
PB7	1	Input/Output	Port B7: Programmable as input or output
TA2IN		Input	8-Bit Time 2 Input: Input to 8-bit Timer 2
INT7		Input	Interrupt Request 7: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
KEYB		Input	Key-On Wakeup Input (dynamic pull-up selectable) (with internal pull-up resistor)
PC0	1	Input/Output	Port C0: Programmable as input or output
TA4IN		Input	8-Bit Timer 4 Input: Input to 8-bit Timer 4
INT8		Input	Interrupt Request 8: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PC1	1	Input/Output	Port C1: Programmable as input or output
TA6IN		Input	8-Bit Timer 6 Input: Input to 8-bit Timer 6
INT9		Input	Interrupt Request 9: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PC2	1	Input/Output	Port C2: Programmable as input or output
TA8IN		Input	8-Bit Timer 8 Input: Input to 8-bit Timer 8
INTA		Input	Interrupt Request A: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PC3	1	Input/Output	Port C3: Programmable as input or output
TA5OUT		Output	8-Bit Timer 45 Output: Output from 8-bit Timer 4 or Timer 5
PC4	1	Input/Output	Port B7: Programmable as input or output
TAAIN		Input	8-Bit Timer A Input: Input to 8-bit Timer A
PC5	1	Input/Output	Port C5: Programmable as input or output
TA7OUT		Output	8-Bit Timer 67 Output: Output to 8-bit Timer 6 or Timer 7
PC6	1	Input/Output	Port C6: Programmable as input or output
TB8IN0		Input	16-Bit Timer 8 Input 0: Count/capture trigger input to 16-bit Timer 8
KEYC		Input	Key-On Wakeup Input (dynamic pull up selectable) (with internal pull-up resistor)
PC7	1	Input/Output	Port C7: Programmable as input or output
TB8IN1		Input	16-Bit Timer 8 Input: Capture trigger input to 16-bit Timer 8
TA9OUT		Output	8-Bit Timer 89 Output: Output from 8-bit Timer 8 or Timer 9
PD0	1	Input/Output	Port D0: Programmable as input or output
TXD0		Output	Serial Transmit Data 0 Programmable as push-pull or open-drain output
TB9IN0		Input	16-Bit Timer 9 Input 0: Count/capture trigger input to 16-bit Timer 9
PD1	1	Input/Output	Port D1: Programmable as input or output
RXD0		Input	Serial Receive Data 0
TB9IN1		Input	16-Bit Timer 9 Input 1: Capture trigger input to 16-bit Timer 9
PD2	1	Input/Output	Port D2: Programmable as input or output
SCLK0		Input/Output	Serial Clock Input/Output 0
$\overline{CTS0}$		Input	Serial Clear-to-Send 0 Programmable as a push-pull or open-drain output
PD3	1	Input/Output	Port d3: Programmable as input or output
TXD1		Output	Serial Transmit Data 1 Programmable as a push-pull or open-drain output
TBAIN0		Input	16-Bit Timer A Input 0: Count/capture trigger input to 16-bit Timer A
PD4	1	Input/Output	Port D4: Programmable as input or output
RXD1		Input	Serial Receive Data 1
TBAIN1		Input	16-Bit Timer A Input 1: Capture trigger input to 16-bit Timer A
PD5	1	Input/Output	Port D5: Programmable as input or output
SCLK1		Input/Output	Serial Clock Input/Output 1
$\overline{CTS1}$		Input	Serial Clear-to-Send 1 Programmable as a push-pull or open-drain output
TABOUT		Output	8-Bit Timer AB Output: Output from 8-bit Timer A or Timer B
PD6	1	Input/Output	Port D6: Programmable as input or open-drain output
XT1		Input	Connection pin for a low-speed crystal
PD7	1	Input/Output	Port D7: Programmable as input or open-drain output
XT2		Output	Connection pin for a low-speed crystal
PE0	1	Input/Output	Port E0: Programmable as input or output
TXD3		Output	Serial Transmit Data 3 Programmable as push-pull or open-drain output

Pin Name	# of Pins	Type	Function
PE1 RXD3	1	Input/Output Input	Port E1: Programmable as input or output Serial Receive Data 3
PE2 SCLK3 $\overline{CTS3}$	1	Input/Output Input/Output Input	Port E2: Programmable as input or output Serial Clock Input/Output 3 Serial Clear-to-Send 3 Programmable as push-pull or open-drain output
PE3 TXD4	1	Input/Output Output	Port E3: Programmable as input or output Serial Transmit Data 4 Programmable as push-pull or open-drain output
PE4 RXD4	1	Input/Output Input	Port E4: Programmable as input or output Serial Receive Data 4
PE5 SCLK4 $\overline{CTS4}$	1	Input/Output Input/Output Input	Port E5: Programmable as input or output Serial Clock Input/Output 4 Serial Clear-to-Send 4 Programmable as push-pull or open-drain output
PE6 INT1 $\overline{BOOT}$	1	Input/Output Input	Port E6: Programmable as input or output Interrupt Request 1: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive Single Boot Mode: The on-chip flash memory enters Single Boot mode to allow re-programming when this pin is sampled low. This pin should be pulled up to a logic 1 to put the flash memory in Normal mode or when the device contains a mask ROM.
PE7 INT2 INTLV	1	Input/Output Input	Port E7: Programmable as input or output Interrupt Request 2: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive Interleave Mode: This pin should be pulled up to a logic 1 when Interleave mode is used, and pulled down to a logic 0 when Interleave mode is not used.
PF0 TXD5	1	Input/Output Output	Port F0: Programmable as input or output Serial Transmit Data 5 Programmable as push-pull or open-drain output
PF1 RXD5 KEYD	1	Input/Output Input Input	Port F1: Programmable as input or output Serial Receive Data 5 Key-On wakeup Input (dynamic pull-up selectable) (with internal pull-up resistor)
PF2 SCLK5 $\overline{CTS5}$	1	Input/Output Input/Output Input	Port F2: Programmable as input or output Serial Clock Input/Output 5 Serial Clear-to-Send 5 Programmable as push-pull or open-drain output
PF3 SCK	1	Input/Output Input/Output	Port F3: Programmable as input or output Clock input/output pin when the Serial Bus Interface is in SIO mode
PF4 SO SDA	1	Input/Output Output Input/Output	Port F4: Programmable as input or output Data transmit pin when the Serial Bus Interface is in SIO mode Data transmit/receive pin when the Serial Bus Interface is in I^2C mode Programmable as push-pull or open-drain output
PF5 SI SCL	1	Input/Output Input Input/Output	Port F5: Programmable as input or output Data receive pin when the Serial Bus Interface is in SIO mode Clock input/output pin when the Serial Bus Interface is in I^2C mode Programmable as push-pull or open-drain output
PF6 INT0		Input/Output Input	Port F6: Programmable as input or output Interrupt Request 0: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
ALE	1	Output	Address Latch Enable (This signal is driven out only when external memory is accessed.)
TEST0	1	Input	Test pin
TEST1	1	Input	Test pin
RSTPUP	1	Input	The pull-up resistors for Port 3 and Port 4 are enabled when this signal is high at reset, and they are disabled when this signal is low at reset.
DAOUT0-2	3	Output	DA Output
$\overline{NMI}$	1	Input	Nonmaskable Interrupt Request: Causes an NMI interrupt on the falling edge
BW0-1	2	Input	Both BW0 and BW1 should be tied to logic 1.
$\overline{PLLOFF}$	1	Input	This pin should be tied to logic 1 when the frequency multiplied clock from the PLL is used; otherwise, it should be tied to logic 0.
$\overline{RESET}$	1	Input	Reset (with internal pull-up resistor): Initializes the whole TMP1942FD
VREFH	1	Input	Input pin for high reference voltage for the A/D Converter
VREFL	1	Input	Input pin for low reference voltage for the A/D Converter

Pin Name	# of Pins	Type	Function
AVCC	1	—	Power supply pin for the A/D Converter This pin should always be connected to power supply even when the A/D Converter is not used.
AVSS	1	—	Ground pin for the A/D Converter (0V) This pin should always be connected to ground even when the A/D Converter is not used.
DAVCC	1	—	Power supply pin for the D/A Converter This pin should always be connected to power supply even when the D/A Converter is not used.
DAVSS	1	—	Ground pin for the D/A Converter (0V) This pin should always be connected to ground even when the D/A Converter is not used.
DAREFH	1	—	Poser supply input pin for the D/A converter
X1/X2	2	Input/Output	Connection pins for high-speed crystal
FVCC	1	—	Power supply pin for the on-chip flash memory (DVCC in the mask-version product)
FVSS	1	—	Ground pin for the on-chip flash memory (DVSS in the mask-version product)
CVCC	1	—	Power supply pin for an oscillator
CVSS	1	—	Ground pin for an oscillator (0V)
DVCC3	4	—	Power supply pins
DVCC51	1	—	Power supply pin (Port F)
DVCC52	1	—	Power supply pin (Port C)
DVSS	5	—	Ground pins (0V)

Port C functions as a 5-V port when DVCC52 is connected to a 5-V power supply.

Port F functions as a 5-V port when DVCC51 is connected to a 5-V power supply.

Note: When the DSU is enabled, Port 9 functions as an interface to a processor probe regardless of the setting of the Port 9 Control Register (P9CR).

The following JTAG pins are available in the TMP1942 FD in a CSP package.

Pin Name	# of Pins	Type	Function
$\overline{\text{TRST}}$	1	Input	JTAG reset pin (with internal pull-up resistor)
TCK	1	Input	JTAG clock pin
TDI	1	Input	JTAG data input pin (with internal pull-up resistor)
TDO	1	Output	JTAG data output pin
TMS	1	Input	JTAG mode switch input pin (with internal pull-up resistor)

3. Flash Memory

This chapter describes the hardware configuration and functionality of the flash memory contained in the TMP1942FD.

3.1 Features

(1) Organization

The TMP1942FD contains 4 Mbits (512 Kbytes) of flash memory, which is divided into a total of 16 blocks (32 Kbytes x 16) to allow independent protection from program and erase for each block. While the CPU can access information in the flash memory through a full 32-bit data bus, an external flash programmer can only perform 16-bit data bus writes to the flash memory.

(2) Access Types

The flash memory of the TMP1942FD provides two selectable access types: one-clock access and interleaved access.

(3) Program/Erase Times

Chip program time: 6 seconds (typ.), including program verify operations (20 μs per long word)

Chip erase time: 30 seconds (typ.), including erase verify operations

Note: These program and erase times are typical values and do not include data transfer overhead. The actual chip program and erase times depend on the programming method used.

(4) Programming Modes

On-Board Programming modes allow re-programming of the flash memory while the chip is soldered on a printed circuit board. Programmer mode utilizes an EPROM programmer to re-program the flash memory.

- On-Board Programming modes
  1) User Boot mode

     Supports use of a user-written programming algorithm.

  2) Single Boot mode

     Downloads new program code using a Toshiba-defined serial interface protocol.
- Programmer mode

  Supports use of a general-purpose EPROM programmer. (Planned)

(5) Re-programming

The flash memory contained in the TMP1942FD is compatible with the JEDEC standards, except a few unique functions. Thus, it is easy to migrate from a discrete flash memory device to the TMP1942FD on-chip flash memory. The TMP1942FD contains hardware to perform programming and erase operations automatically. This eliminates the need for the user to code complex program and erase sequences.

The security feature of the TMP1942FD flash memory prevents the stored data from being read while it is being re-programmed with programming equipment. The TMP1942FD also allows the user to protect individual blocks of the flash memory against program or erase through software commands; however, 12-V VPP programming does not support data protection on a block-by-block basis.

JEDEC Standards	Changes and Enhancements
•Auto Program • Auto Chip Erase • Auto Block Erase • Auto Multi-Block Erase •Data Polling/Toggle Bit	Added feature: Security Auto Program Changed feature: Block protection is available only under software control. Removed feature:Erase Resume/Suspend mode

3.1.1 Block Diagram

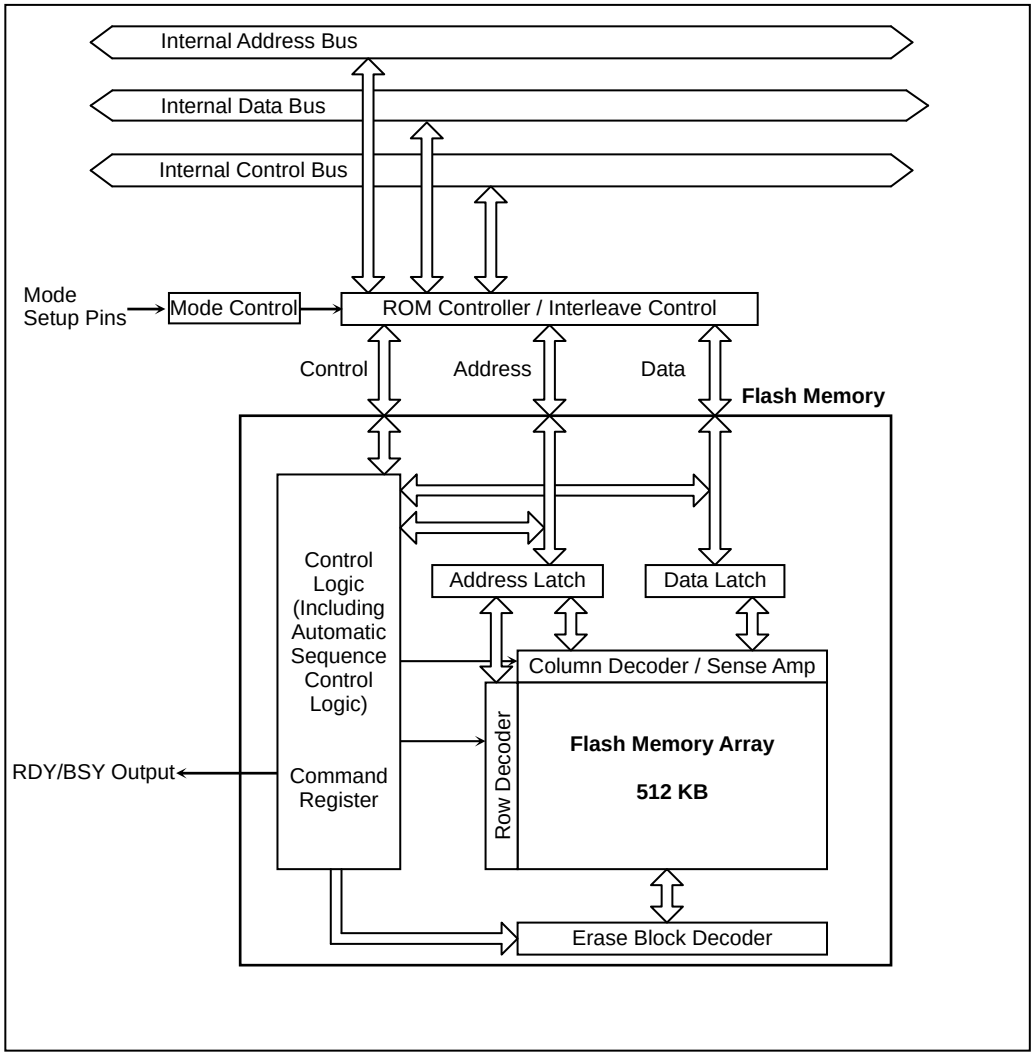


Figure 3.1.1 Flash Memory Block Diagram

3.2 Operating Modes

The TMP1942FD offers a total of five operating modes, including the one in which the flash memory is unused.

Table 3.2.1 Operating Modes

Operating Mode	Description
Single-Chip Mode	After a reset, the TX19 core processor executes out of the on-chip flash memory. Either fast (one-clock) or interleave mode operation is selected through the INTLV pin when the reset state is released. Single-Chip mode is further divided into Normal mode in which the user application executes and User Boot mode which allows re-programming of the flash memory while the TMP1942FD is installed on a printed circuit board. The user can freely define how to switch between Normal mode and User Boot mode. For example, the logic state on, for example, Port 00 can be used to determine whether to put the flash memory in Normal mode or User Boot mode. The user must include a routine in the application program to test the state of that port.
Normal Mode	
User Boot Mode	
Single Boot Mode	After a reset, the TX19 core processor executes out of the on-chip boot ROM (which is a mask ROM).The boot ROM contains a routine to aid users in performing on-board programming of the flash memory via a serial port of the TMP1942FD. The serial port is connected to an external host which transfers new data according to a prescribed protocol.
Writer Mode	This mode allows re-programming of the flash memory with a general-purpose EPROM programmer. Use the programmer and programming adaptor recommended by Toshiba.

The on-chip flash memory can be re-programmed in one of the following three modes: User Boot mode, Single Boot mode and Writer mode. Of these modes, User Boot mode and Single Boot mode are collectively referred to as on-board programming modes.

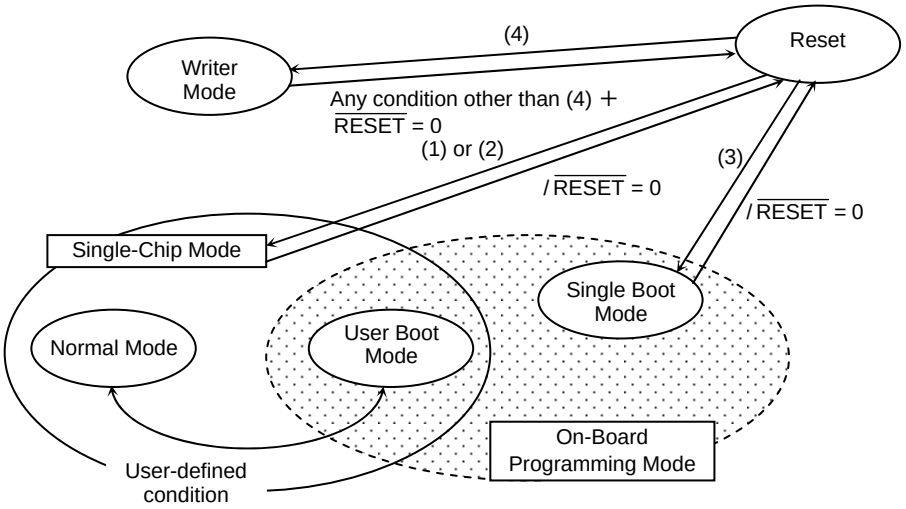
The logic states on the BW0, BW1, $\overline{\text{BOOT}}$ and INTLV pins during a reset sequence determine the mode of operation for the flash memory, as shown in Table 3.2.2. After $\overline{\text{RESET}}$ is released, PE6 ($\overline{\text{BOOT}}$) and PE7 (INTLV) can be configured as general-purpose I/O or timer output pins.

After $\overline{\text{RESET}}$ is released, the CPU starts operating in the selected mode, except for Writer mode. When Writer mode is selected, $\overline{\text{RESET}}$ must be held at logic 0. The input pins listed in Table 3.2.2 must remain stable once the flash memory is put in a given mode of operation.

Table 3.2.2 Modes of Operation

	Operating Mode	Input Pins				
		$\overline{\text{RESET}}$	BW0	BW1	$\overline{\text{BOOT}}$	INTLV
(1)	Single-Chip Mode (Interleave)	0 → 1	1	1	1	1
(2)	Single-Chip Mode (Single-Clock)	0 → 1	1	1	1	0
(3)	Single Boot Mode	0 → 1	1	1	0	(Note 1)
(4)	Writer Mode (Note 2)	0	0	1	(Note 1)	(Note 1)

Note 1: Don't care; however, the pins must be held at 0 or 1.
Note 2: In Writer mode, P40 must be held at 1, and P41 and P42 must be held at 0. For how to set other pins, refer to Section 3.4.2.3, *Pin Functions and Settings*.



Parenthesized numbers indicate that the relevant pins are at the logic states shown in Table 3.2.2.

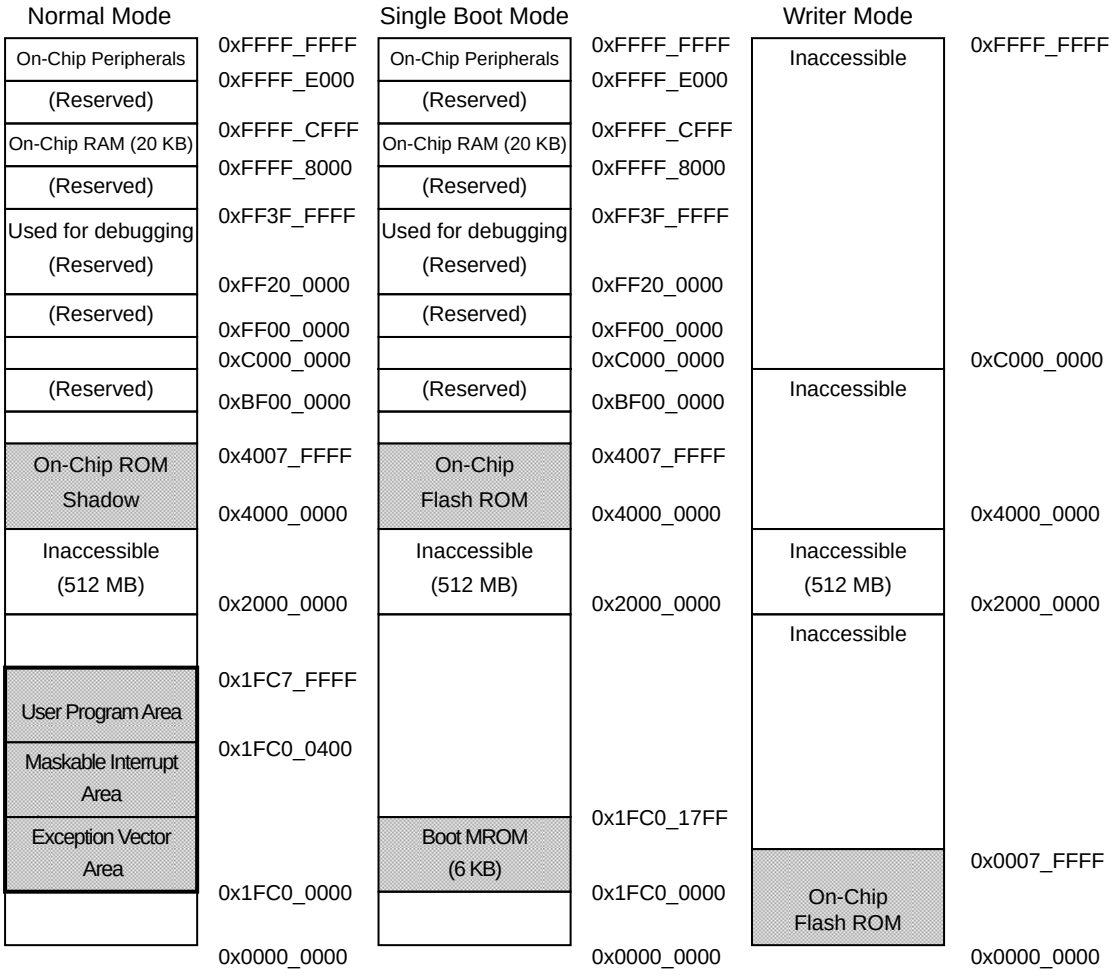
Figure 3.2.1 Mode Transitions

3.2.1 Reset Operation

To reset the TMP1942FD, $\overline{\text{RESET}}$ must be asserted for at least 12 system clock periods after the power supply voltage and the internal high-frequency oscillator have stabilized. This time is typically 3 μs at 32 MHz when the on-chip PLL is utilized.

3.2.2 Memory Maps

The memory map for the TMP1942FD varies according to the operation mode selected for the on-chip flash memory. Following are the memory maps in each operation mode.



Note: The addresses shown above are physical addresses.

Figure 3.2.2 TMP1942FD Memory Maps

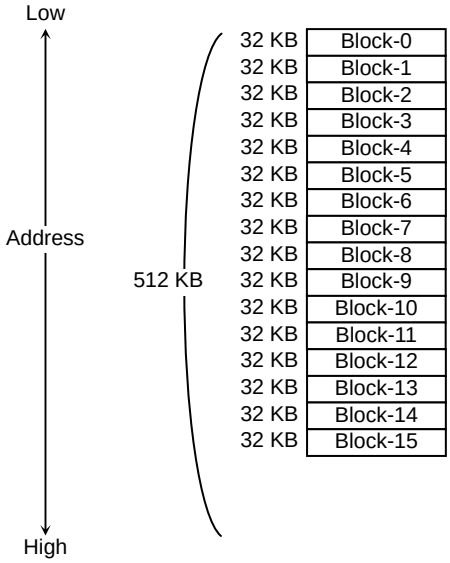


Figure 3.2.3 Flash Memory Block Architecture

Table 3.2.3 Block Addresses

	User Boot Mode	Boot Mode	Writer Mode
Block-0	0x1FC0_0000 - 0x1FC0_7FFF (or 0x4000_0000 - 0x4000_7FFF)	0x1FC0_0000 - 0x1FC0_7FFF	0x0000_0000 - 0x0000_7FFF
Block-1	0x1FC0_8000 - 0x1FC0_FFFF (or 0x4000_8000 - 0x4000_FFFF)	0x1FC0_8000 - 0x1FC0_FFFF	0x0000_8000 - 0x0000_FFFF
Block-2	0x1FC1_0000 - 0x1FC1_7FFF (or 0x40010000 - 0x4001_7FFF)	0x1FC1_0000 - 0x1FC1_7FFF	0x0001_0000 - 0x0001_7FFF
Block-3	0x1FC1_8000 - 0x1FC1_FFFF (or 0x4001_8000 - 0x4001_FFFF)	0x1FC1_8000 - 0x1FC1_FFFF	0x0001_8000 - 0x0001_FFFF
Block-4	0x1FC2_0000 - 0x1FC2_7FFF (or 0x4002_0000 - 0x4002_7FFF)	0x1FC2_0000 - 0x1FC2_7FFF	0x0002_0000 - 0x0002_7FFF
Block-5	0x1FC2_8000 - 0x1FC2_FFFF (or 0x4002_8000 - 0x4002_FFFF)	0x1FC2_8000 - 0x1FC2_FFFF	0x0002_8000 - 0x0002_FFFF
Block-6	0x1FC3_0000 - 0x1FC3_7FFF (or 0x4003_0000 - 0x4003_7FFF)	0x1FC3_0000 - 0x1FC3_7FFF	0x0003_0000 - 0x0003_7FFF
Block-7	0x1FC3_8000 - 0x1FC3_FFFF (or 0x4003_8000 - 0x4003_FFFF)	0x1FC3_8000 - 0x1FC3_FFFF	0x0003_8000 - 0x0003_FFFF
Block-8	0x1FC4_0000 - 0x1FC4_7FFF (or 0x4004_0000 - 0x4004_7FFF)	0x1FC4_0000 - 0x1FC4_7FFF	0x0004_0000 - 0x0004_7FFF
Block-9	0x1FC4_8000 - 0x1FC4_FFFF (or 0x4004_8000 - 0x4004_FFFF)	0x1FC4_8000 - 0x1FC4_FFFF	0x0004_8000 - 0x000_4FFFF
Block-10	0x1FC5_0000 - 0x1FC5_7FFF (or 0x4005_0000 - 0x4005_7FFF)	0x1FC5_0000 - 0x1FC5_7FFF	0x0005_0000 - 0x0005_7FFF
Block-11	0x1FC5_8000 - 0x1FC5_FFFF (or 0x4005_8000 - 0x4005_FFFF)	0x1FC5_8000 - 0x1FC5_FFFF	0x0005_8000 - 0x0005_FFFF
Block-12	0x1FC6_0000 - 0x1FC6_7FFF (or 0x4006_0000 - 0x4006_7FFF)	0x1FC6_0000 - 0x1FC6_7FFF	0x0006_0000 - 0x0006_7FFF
Block-13	0x1FC6_8000 - 0x1FC6_FFFF (or 0x4006_8000 - 0x4006_FFFF)	0x1FC6_8000 - 0x1FC6_FFFF	0x0006_8000 - 0x0006_FFFF
Block-14	0x1FC7_0000 - 0x1FC7_7FFF (or 0x4007_0000 - 0x4007_7FFF)	0x1FC7_0000 - 0x1FC7_7FFF	0x0007_0000 - 0x0007_7FFF
Block-15	0x1FC7_8000 - 0x1FC7_FFFF (or 0x4007_8000 - 0x4007_FFFF)	0x1FC7_8000 - 0x1FC7_FFFF	0x0007_8000 - 0x0007_FFFF

3.2.3 Interleave Mode

If Port E7 (PE7) is sampled high at the rising edge of $\overline{\text{RESET}}$, the flash memory enters Interleave mode. When the system clock (fsys) operates at 16 MHz or faster, the flash memory must be configured into Interleave mode.

The TMP1942FD flash memory is comprised of two banks which are distinguished by the second lowest bit of an address. The memory controller of the TMP1942FD core processor provides addresses for each bank. Each bank can be accessed in two system clock cycles. Switching between the two banks enables each address to be accessed in a single system clock cycle, unless a branch occurs (see Figure 3.2.4).

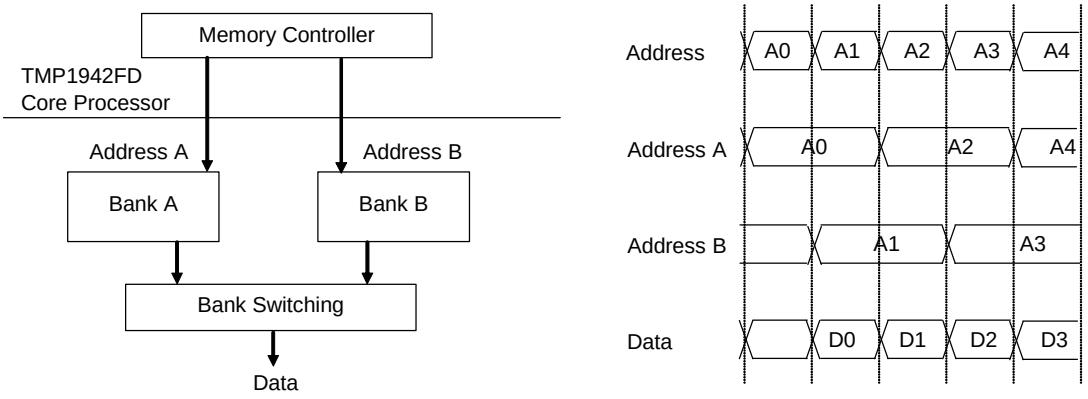


Figure 3.2.4 Interleave Configuration

3.2.4 Block Protection

The TMP1942FD flash memory is organized into a total of 16 blocks (32 Kbytes x 16). To protect stored data from any program and erase operations, each block has a protect bit, which can be set by executing the Block Protect command sequence. Blocks in protection mode are protected from even the Chip Erase and Multi-Block Erase commands; these commands erase only unprotected blocks. Since protection status is stored in flash memory cells, it is retained even if the chip is powered off.

3.2.5 DSU-ICE Interface

If Port 37 (P37) is sampled high at the rising edge of $\overline{\text{RESET}}$, the TMP1942FD enters DSU mode, which is used for software debugging using an external DSU-ICE unit. In DSU mode, Port A serves as an interface to the DSU-ICE, and cannot be used for other functions. Consult the DSU-ICE operation manual for a description of debugging using the DSU-ICE. When the TMP1942FD is in DSU mode, the on-chip flash memory provides a security feature.

(1) Flash security feature

The TMP1942FD supports on-board debugging using a DSU-ICE while it is installed on a printed circuit board. The TMP1942FD provides a security feature to prevent intrusive access to the flash memory. When the flash memory is in the secure state, a DSU-ICE is denied access to the entirety of the flash memory.

(2) Securing the flash memory (Disabling debugging with a DSU-ICE)

Once program debugging is completed, set the FSE bit in the Flash Control/Status Register and write the Auto Security On command. This turns on the security feature. While the flash memory is in the secure state, a DSU-ICE cannot read its contents. When the chip is powered off and powered on again, the SEQON bit in the SEQMOD register is automatically set, which disables debugging using a DSU-ICE until the flash memory is unsecured.

(3) Unsecuring the flash memory (Enabling debugging with a DSU-ICE)

The flash memory may only be unsecured by clearing the SEQON bit in the SEQMOD register and then writing a special code (0x0000_00C5) to the Security Control (SEQCNT) register. This prevents runaway software from inadvertently turning off the security feature. Unsecuring the flash memory enables the DSU interface. The flash memory can be secured again by setting the SEQON bit in the SEQMOD and writing 0x0000_00C5 to the SEQCNT while the chip is powered.

SEQMOD (0xFFFF_E510)

	7	6	5	4	3	2	1	0
Bit Symbol								SEQON
Read/Write								R/W
Reset Value								1
Function								1: Security on 0: Security off

Note: This register must be read as a 32-bit quantity. Bits 1 to 31 are read as 0s.

SEQCNT (0xFFFF_E514)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							

Note: The security feature of the TMP1942FD flash memory is not intended to guarantee rigid security protection. In cases where security protection is of utmost importance, use the mask ROM version of this product.

(4) Application example

The following flowchart shows an example of how to use the security feature with a DSU-ICE.

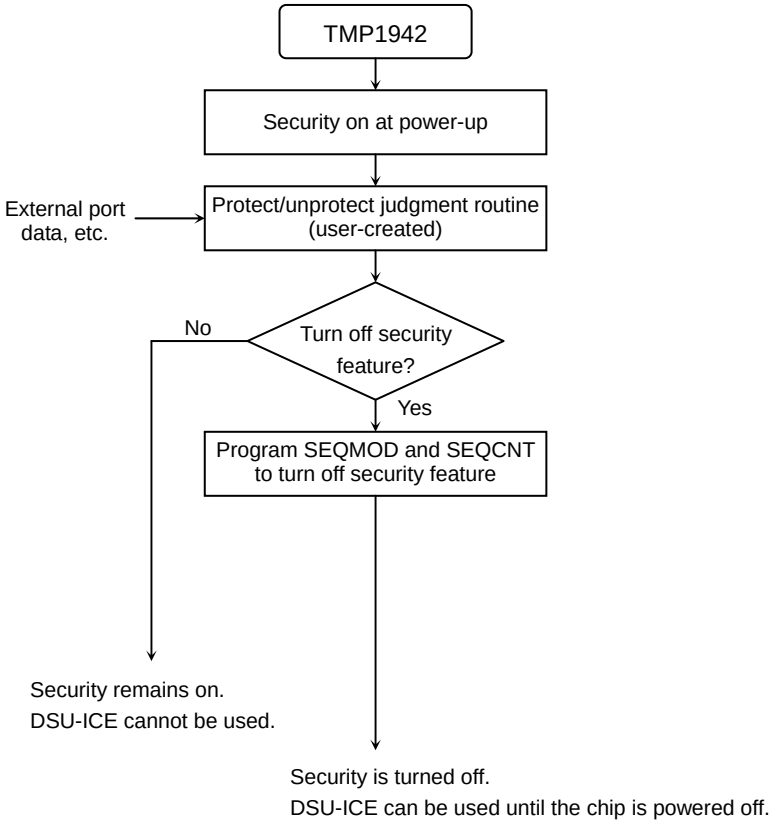


Figure 3.2.5 Using the Security Feature

3.3 On-Board Programming Mode

On-board programming modes allow re-programming of the flash memory while the TMP1942FD is soldered on a printed circuit board. In Single Boot mode, new data comes from a serial port under control of a Toshiba-provided routine in the boot ROM. User Boot mode allows you to create an algorithm of your own for flash memory erase and program operations.

The TMP1942FD flash memory provides a security feature to prevent intrusive access to the flash memory while in Programmer mode. This security feature can be enabled upon completion of on-board programming to reduce the potential risk of software leaks to third parties.

3.3.1 User Boot Mode (Single-Chip Mode)

User Boot mode allows you to create a programming algorithm of your own. This mode supports situations where the flash memory is to be re-programmed via a bus other than serial I/O. User Boot mode is one of the two submodes in Single-Chip mode; the other submode is Normal mode in which the CPU executes the user application. To re-program the flash memory, the mode of operation must be switched from Normal mode to User Boot mode. The user application code must include a mode judgment routine as part of the reset procedure.

The user must define the conditions for mode switching, based on the logic states on I/O ports of the TMP1942FD. Additionally, the user must incorporate a programming algorithm into the user application code that is to be executed after User Boot mode is entered.

It is not possible to read from the flash memory while it is being erased or programmed; therefore, the programming algorithm must be placed and executed outside of the flash memory.

Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption.

All interrupts including the non-maskable (NMI) interrupt must be globally disabled while the flash memory is being erased or programmed.

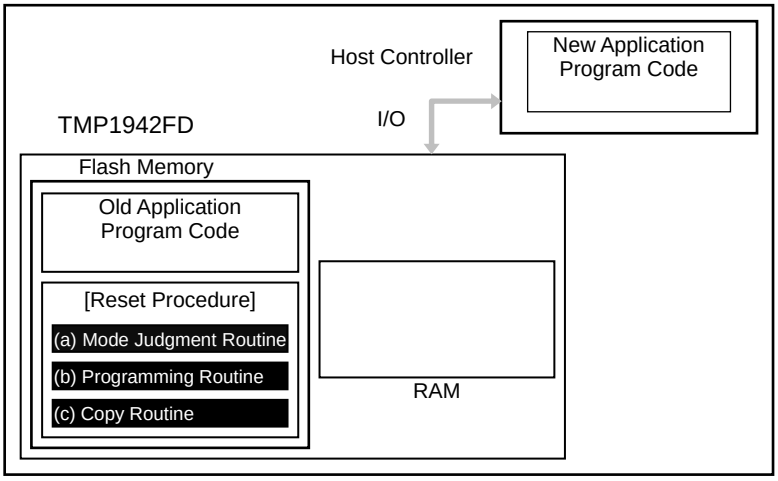
The pages that follow describe the general procedures for two cases where the programming routine is: a) stored within the TMP1942FD flash memory, and b) loaded from an external controller. For a detailed description of the erase and program sequence, refer to Section 3.4. *On-Board Programming and Erasure*.

User Boot Mode

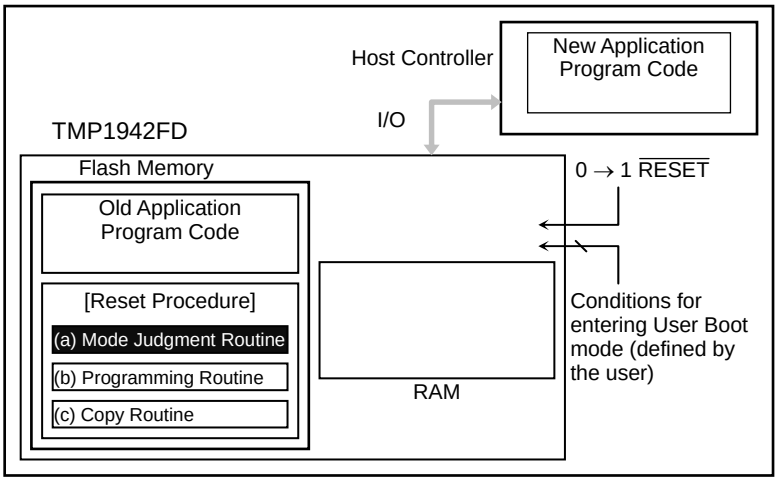
(1-A) Method 1: Storing a Programming Routine in the Flash Memory

(1) Determine the conditions (e.g. pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMP1942FD on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

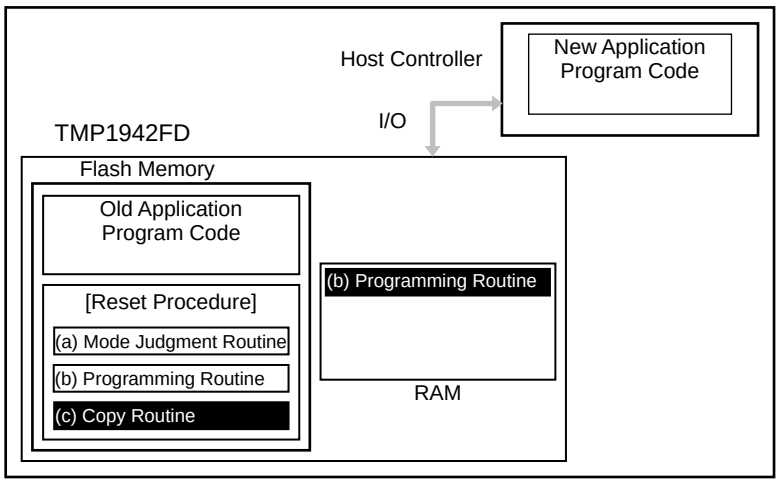
- Mode judgment routine: Code to determine whether or not to switch to User Boot mode
- Programming routine: Code to download new program code from a host controller and re-program the flash memory
- Copy routine: Code to copy the flash programming routine from the TMP1942FD flash memory to either the TMP1942FD on-chip RAM or an external memory device.



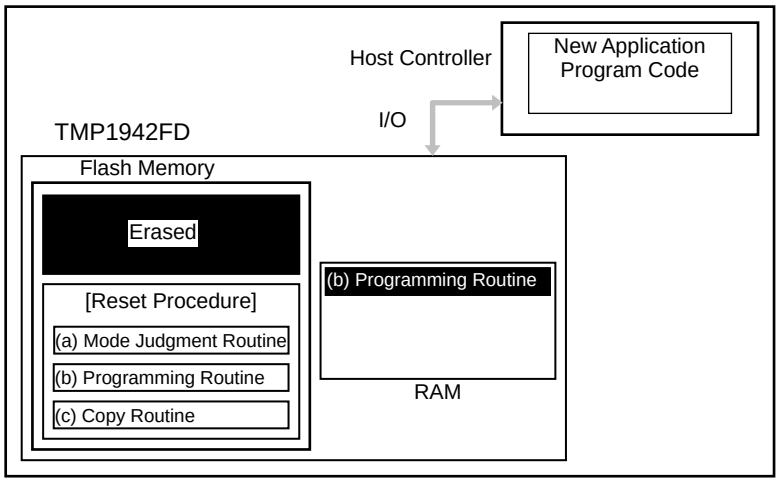
(2) After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMP1942FD flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be globally disabled while in User Boot mode.)



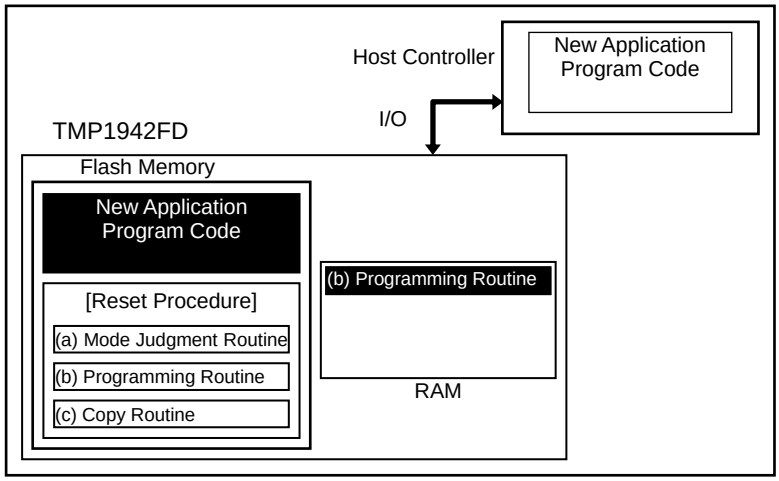
(3) Once User Boot mode is entered, execute the copy routine to copy the flash programming routine to either the TMP1942FD on-chip RAM or an external memory device. (In the following figure, the on-chip RAM is used.)



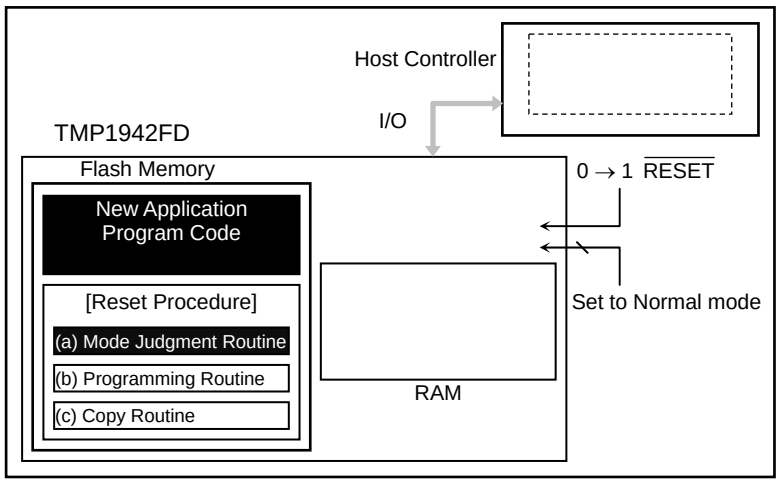
(4) Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



(5) Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.



(6) Drive $\overline{\text{RESET}}$ low to reset the TMP1942FD. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



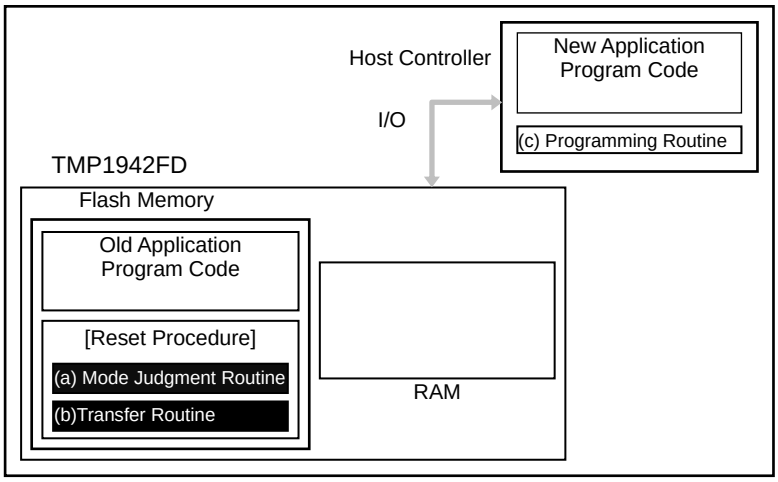
(1-B) Method 2: Transferring a Programming Routine from an External Host

(1) Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMP1942FD on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

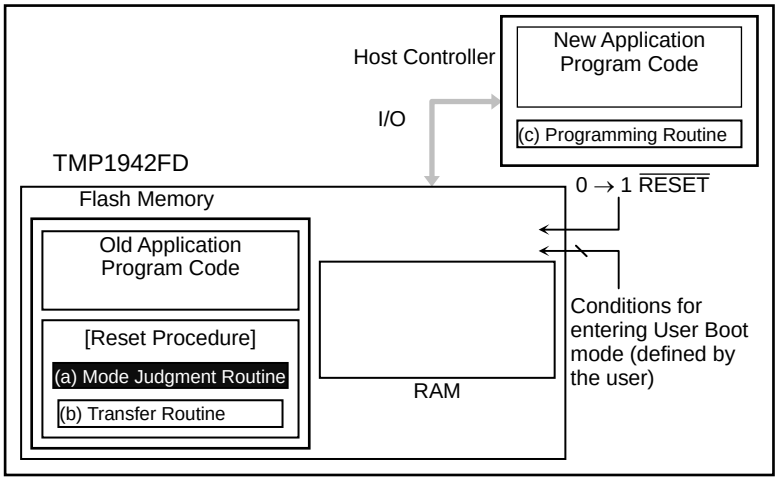
- Mode judgment routine: Code to determine whether or not to switch to User Boot mode
- Transfer routine: Code to download new program code from a host controller

Also, prepare a programming routine on the host controller

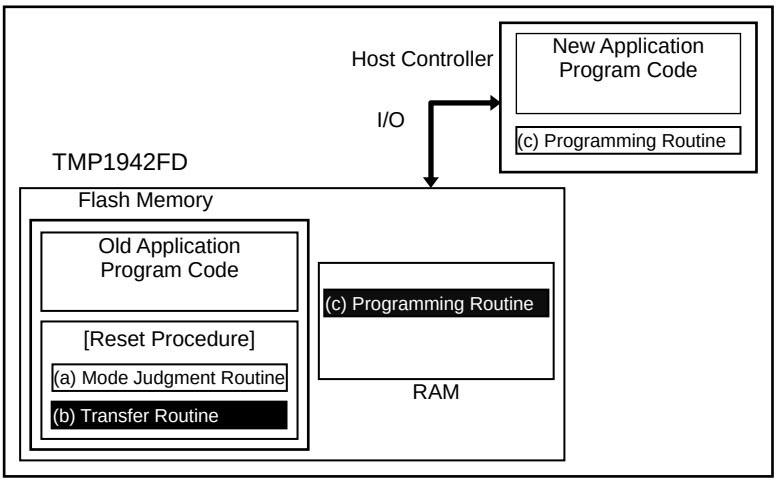
- Programming routine: Code to download new program code from an external host controller and re-program the flash memory



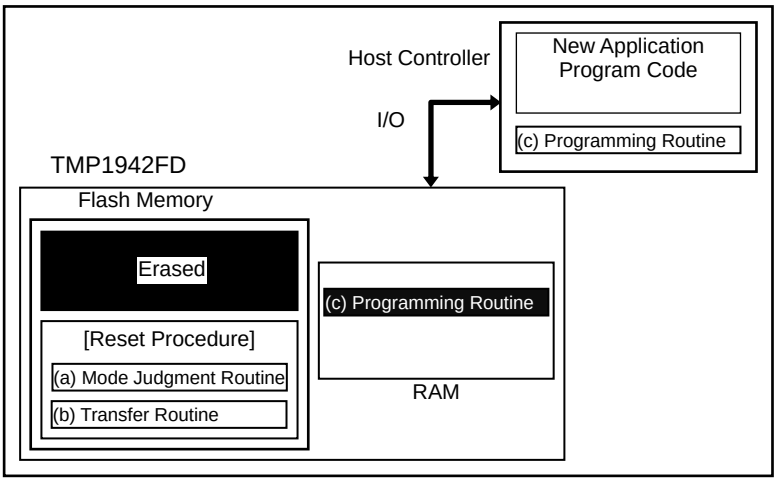
(2) After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMP1942FD flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be globally disabled while in User Boot mode.)



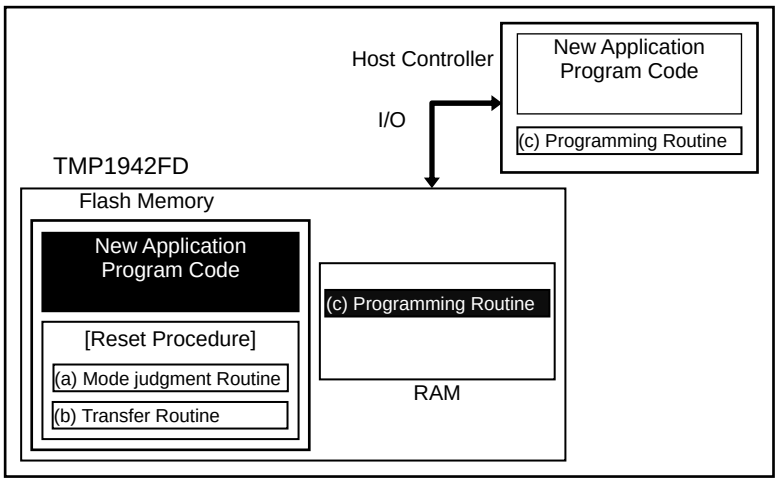
(3) Once User Boot mode is entered, execute the transfer routine to download the flash programming routine from the host controller to either the TMP1942FD on-chip RAM or an external memory device. (In the following figure, the on-chip RAM is used.)



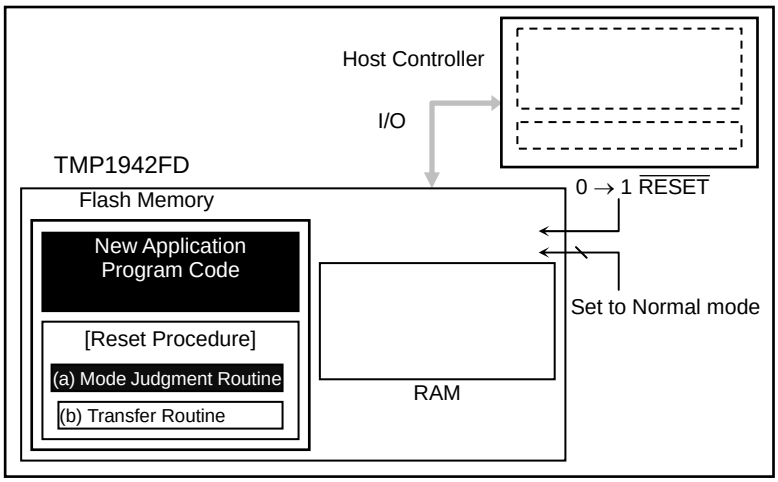
(4) Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



(5) Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.



(6) Drive $\overline{\text{RESET}}$ low to reset the TMP1942FD. Upon reset, the on-chip flash memory is put in Normal mode. After is $\overline{\text{RESET}}$ released, the CPU will start executing the new application program code.



3.3.2 Single Boot Mode

In Single Boot mode, the flash memory can be re-programmed by using a program contained in the TMP1942FD on-chip boot ROM. This boot ROM is a masked ROM. When Single Boot mode is selected upon reset, the boot ROM is mapped to the address region including the interrupt vector table while the flash memory is mapped to an address region different from it (See Figure 3.2.2).

Single Boot mode allows serial programming of the flash memory. Channel 0 of the SIO (SIO0) of the TMP1942FD is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMP1942FD on-chip RAM. Then, the flash memory is re-programmed by executing the programming routine. The host sends out both commands and programming data to re-program the flash memory.

Communications between the SIO0 and the host must follow the protocol described later. To secure the contents of the flash memory, the validity of the application's password is checked before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted.

As in the case of User Boot mode, all interrupts including non-maskable (NMI) interrupt must be globally disabled in Single Boot mode while the flash memory is being erased or programmed.

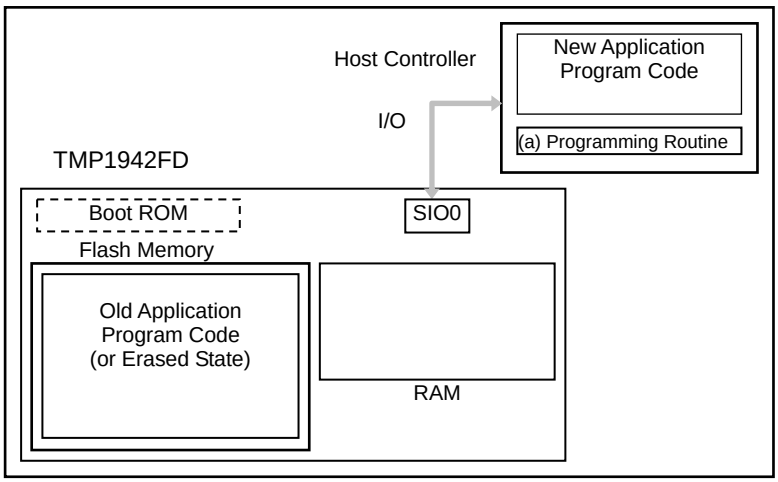
Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations. For a detailed description of the erase and program sequences, refer to section 3.4. *On-Board Programming and Erasure*.

Note: In Single Boot mode, the boot ROM programs are executed in Normal mode. Do not switch to any other mode in the programming routine.

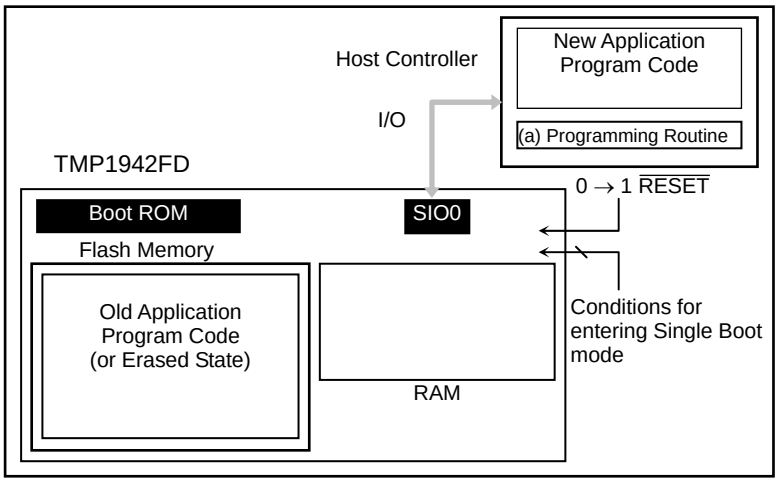
Boot Mode

(2-A) General Procedure: Using the Program in the On-Chip Boot ROM

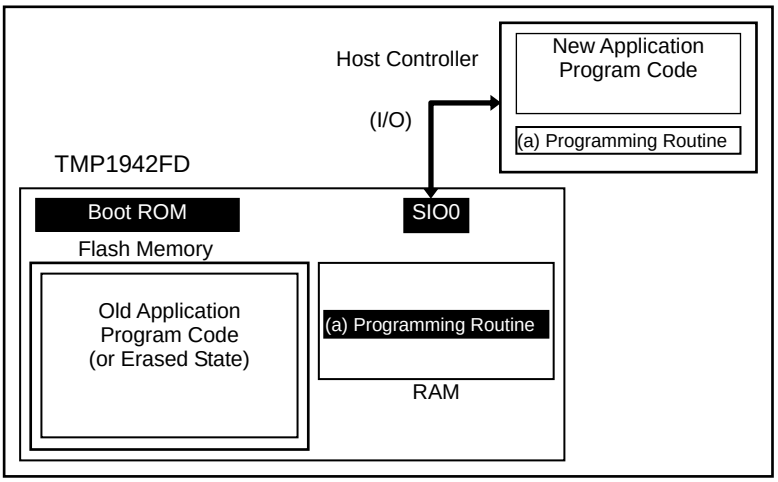
(1) The flash block containing the older version of the program code need not be erased before executing the programming routine. Since a programming routine and programming data are transferred via the SIO0, the SIO0 must be connected to a host controller. Prepare a programming routine on the host controller.



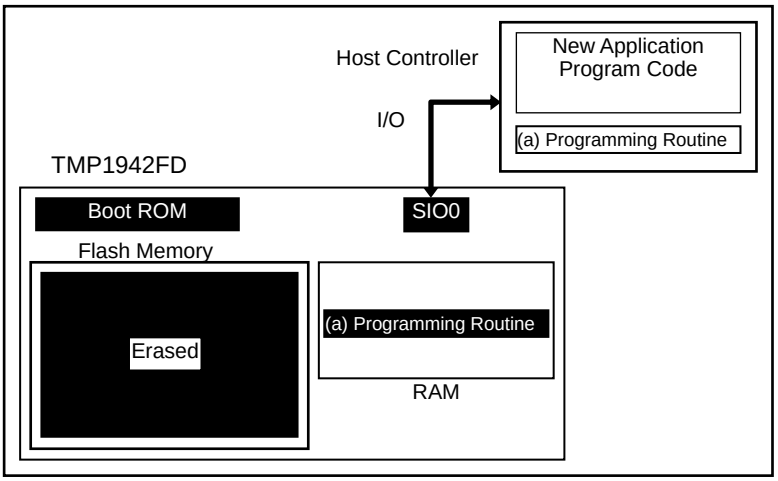
(2) Reset the TMP1942FD with the mode setting pins held at appropriate logic values, so that the CPU re-boots from the on-chip boot ROM. The 12-byte password transferred from the host controller is first compared to the contents of the special flash memory locations. (If the flash block has already been erased, the password is 0xFFFF).



(3) If the password was correct, the boot program downloads, via the SIO0, the programming routine from the host controller into the on-chip RAM of the TMP1942FD. The programming routine must be stored in the address range 0xFFFF_8000 - 0xFFFF_BFFF.

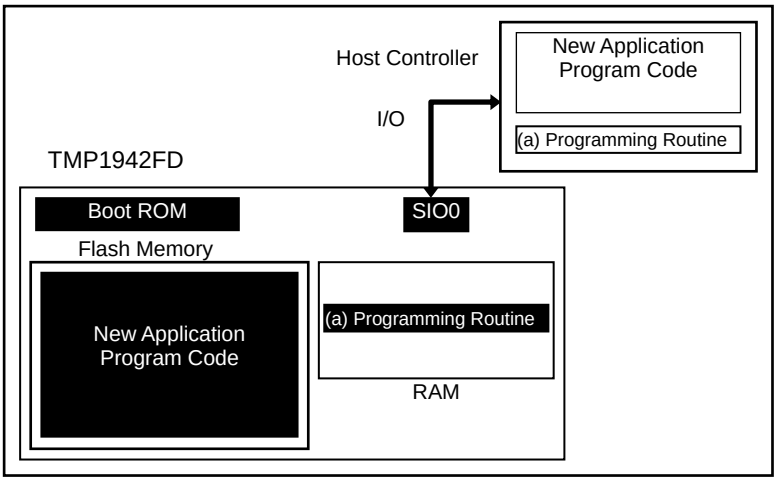


(4) The CPU jumps to the programming routine in the on-chip RAM to erase the flash block containing the old application program code. The Block Erase or Chip Erase command may be used.

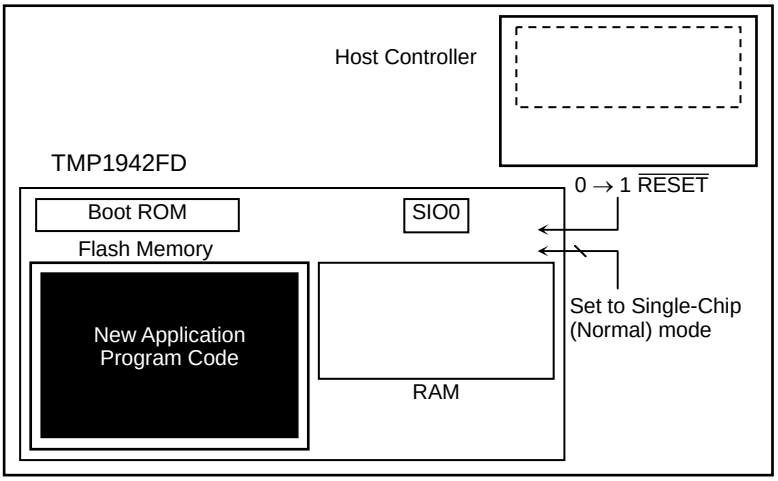


Note: At this time, r29 (sp) points to 0xFFFF_9100.

(5) Next, the programming routine (a) downloads new application program code from the host controller and programs it into the erased flash block. Once programming is complete, protection of that flash block is turned on.
It is not allowed to move program control from the programming routine (a) back to the boot ROM.
In the example below, new program code comes from the same host controller via the same SIO channel as for the programming routine. However, once the programming routine has begun to execute, it is free to change the transfer path and the source of the transfer. Create board hardware and a programming routine to suit your particular needs.



(6) When programming of the flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the TMP1942FD re-boots in Single-Chip (Normal) mode to execute the new program.



3.3.3 Host-to-Target Connection Examples

In Single Boot mode, serial transfer is used to re-program the flash memory while the TMP1942FD is installed on the board. In this mode, channel 0 of the SIO (SIO0) of the TMP1942FD is connected to a host controller, which is to issue commands to the target board. Figure 3.3.1 and Figure 3.3.2 show examples of host-to-target connections.

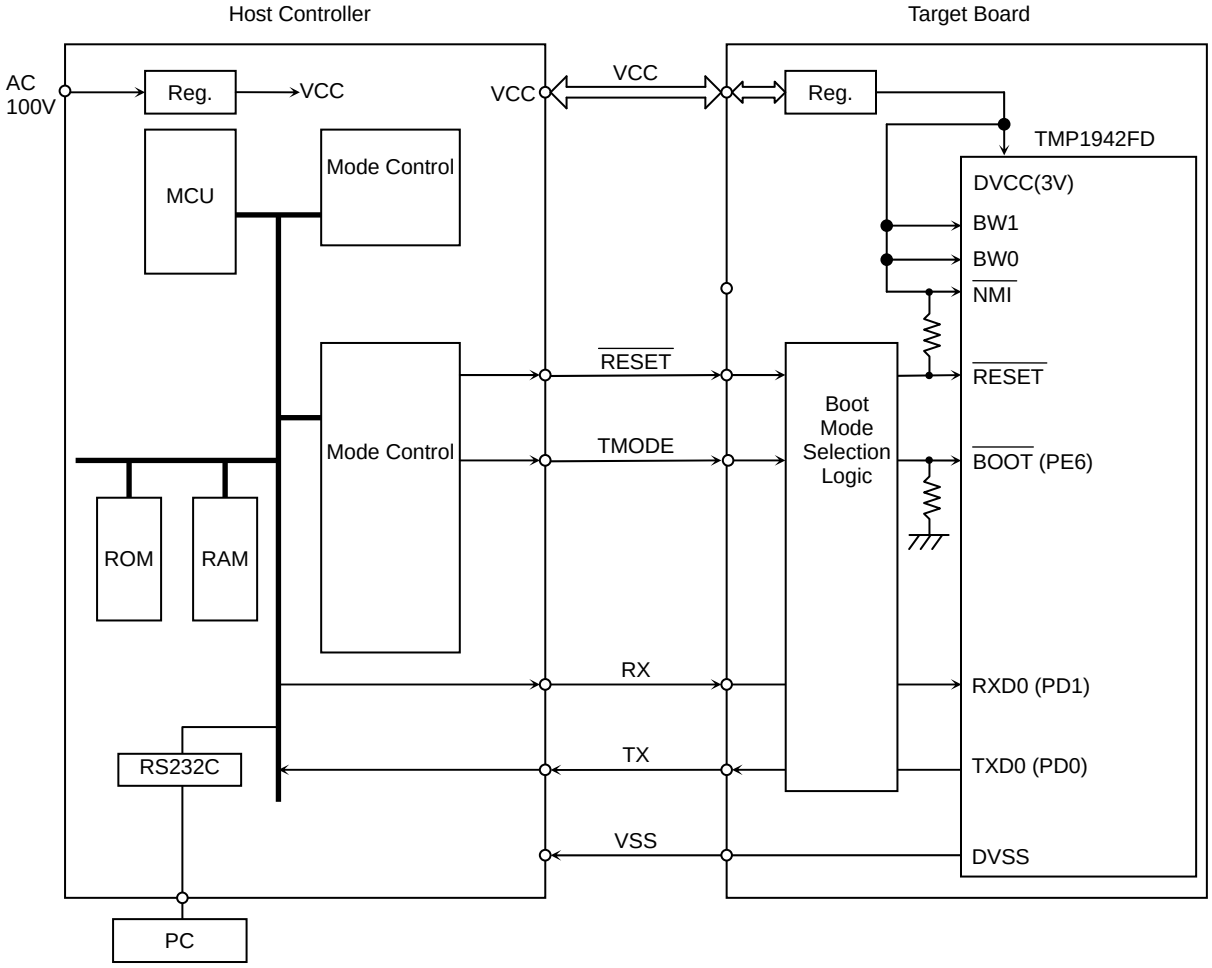


Figure 3.3.1 Example of a Connection between a Host Controller and a Target Board (When the SIO0 is Configured for UART Mode)

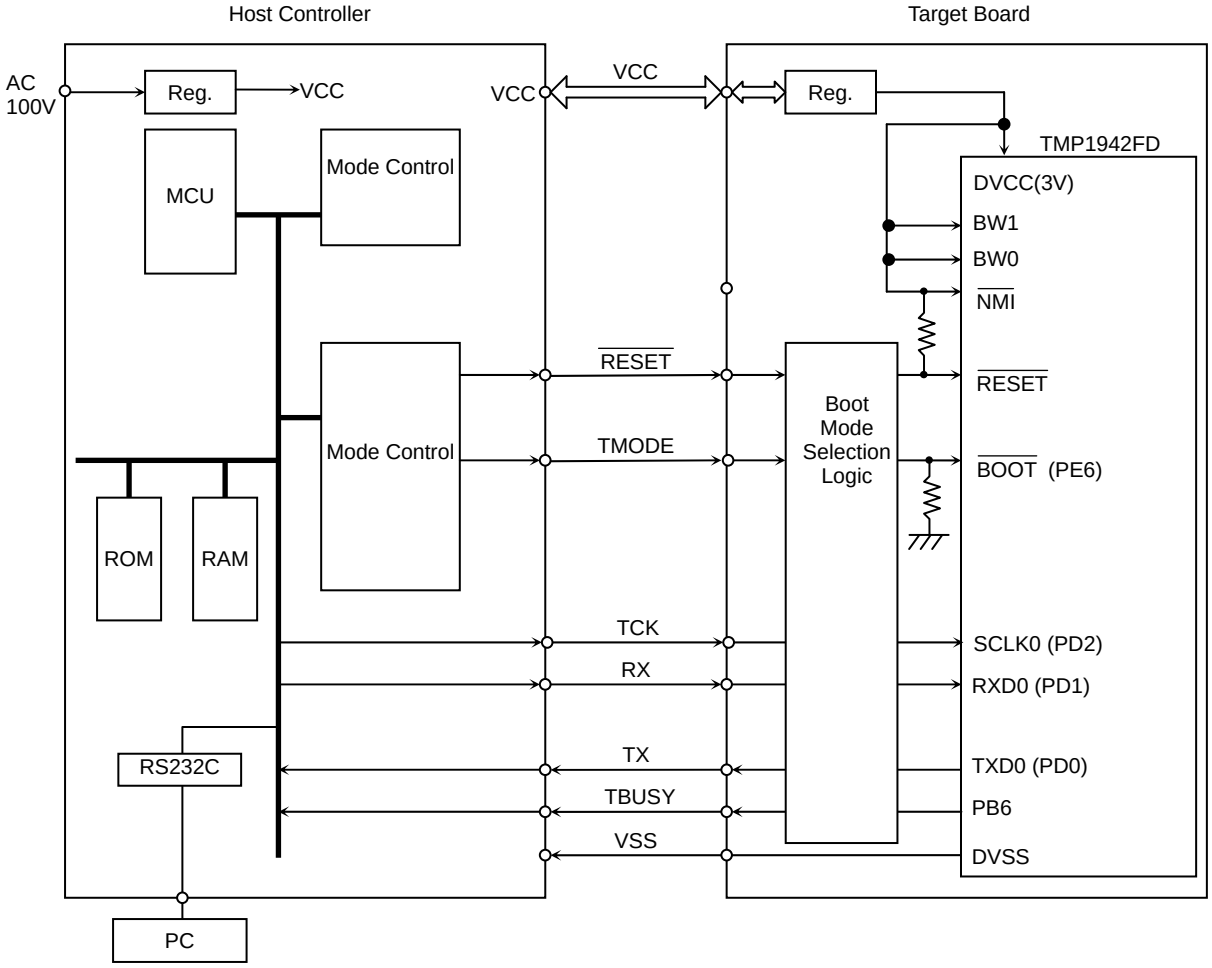


Figure 3.3.2 Example of a Connection between a Host Controller and a Target Board
(When the SIO0 is Configured for I/O Interface Mode)

The NET IMPRESS (flash programmer) from Yokogawa Digital Computer Corporation is being planned as our recommended programming controller.

3.3.4 Configuring for Single Boot Mode

For on-board programming, boot the TMP1942FD in Single Boot mode, as follows:

BW0	= 1
BW1	= 1
$\overline{\text{BOOT}}$ (PE6)	= 0
$\overline{\text{RESET}}$	= 0 → 1

Set the $\overline{\text{RESET}}$ input at logic 0, and the BW0, BW1 and $\overline{\text{BOOT}}$ (PE6) inputs at the logic values shown above, and then release $\overline{\text{RESET}}$ (high).

3.3.5 Memory Map

Figure 3.3.3 shows a comparison of the memory maps in Normal and Single Boot modes. In Single Boot mode, the on-chip flash memory is mapped to physical addresses 0x4000_0000 through 0x4007_FFFF, virtual addresses 0x0000_0000 through 0x0007_FFFF, and the on-chip boot ROM is mapped to physical addresses 0x1FC0_0000 through 0x1FC0_17FF.

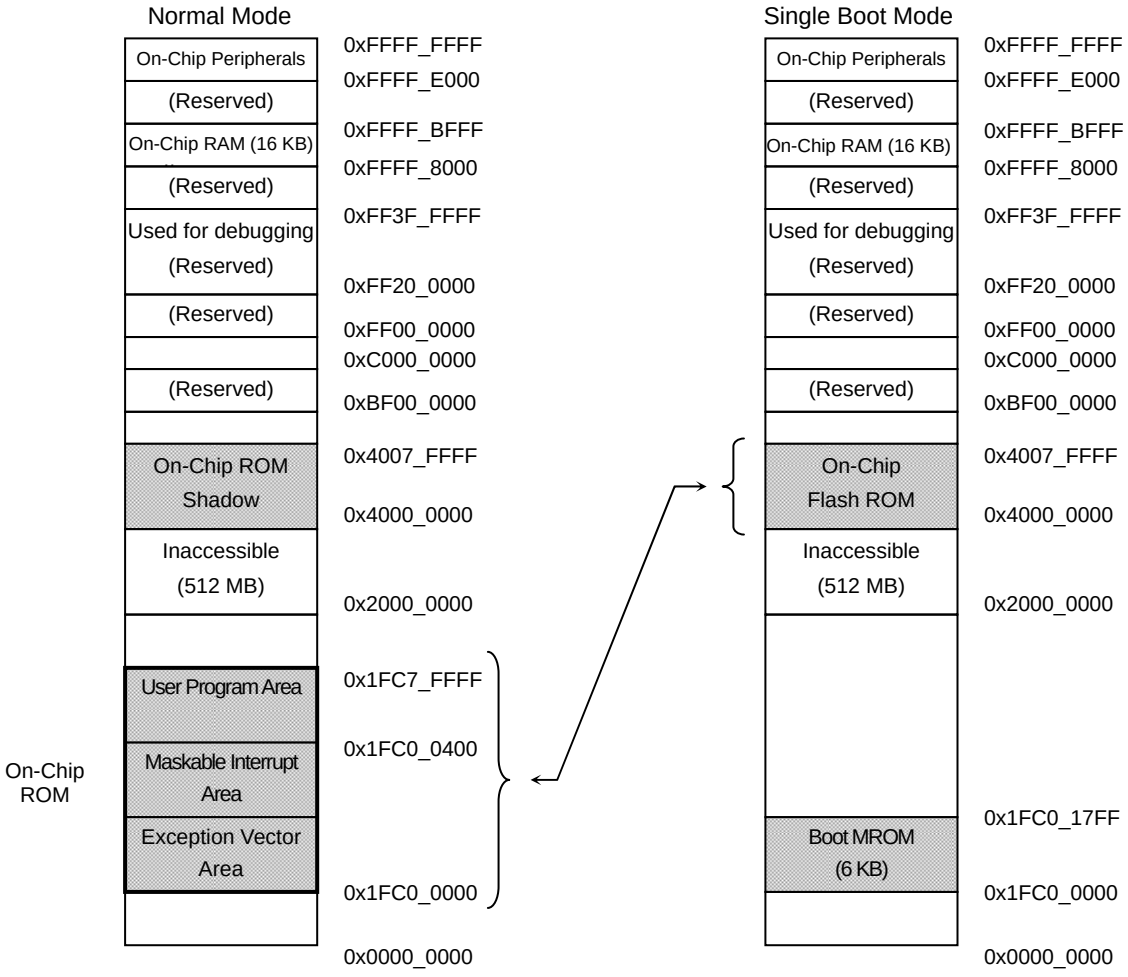


Figure 3.3.3 Memory Maps for Normal and Single Boot Modes (Physical Addresses)

3.3.6 Interface Specifications

In Single Boot mode, an SIO channel is used for communications with a programming controller. Both UART (asynchronous) and I/O Interface (synchronous) modes are supported. The communication formats are shown below. To perform on-board programming, the programming controller must also be set for the same communication format.

- UART mode
  - Communication channel: SIO channel 0 (SIO0)
  - Transfer mode: UART (asynchronous) mode, full-duplex
  - Data length: 8 bits
  - Parity bits: None
  - STOP bits: 1 bit
  - Baud rate: Arbitrary baud rate

- I/O Interface mode
  - Communication channel: SIO channel 0 (SIO0)
  - Transfer mode: I/O Interface mode, half-duplex
  - Synchronization clock (SCLK0): Input
  - Handshaking signal: PB6 configured as an output
  - Baud rate: Arbitrary baud rate

Table 3.3.1 Required Pin Connections

Pin		Interface	
		UART Mode	I/O Interface Mode
Power Supply Pins	DVCC (3V)	Required	Required
	DVSS	Required	Required
Mode Setting Pin	$\overline{\text{BOOT}}$	Required	Required
Reset Pin	$\overline{\text{RESET}}$	Required	Required
Communication Pins	TXD0	Required	Required
	RXD0	Required	Required
	SCLK0	Not required	Required (Input mode)
	PB6	Not required	Required (Output mode)

3.3.7 Data Transfer Format

The host controller is to issue one of the commands listed in Table 3.3.2 to the target board. Table 3.3.3 to Table 3.3.6 show the sequence of two-way communications that should occur in response to each command.

Table 3.3.2 Single Boot Mode Commands

Code	Command
10H	RAM Transfer
20H	Show Flash Memory Sum
30H	Show Product Information

Table 3.3.3 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMP1942FD	Baud Rate	Data Transferred from the TMP1942FD to the Controller
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H For I/O Interface mode 30H	Desired baud rate (Note 1)	-
	2nd byte	-		ACK for the serial operation mode byte For UART mode Normal acknowledge 86H (The boot program aborts if the baud rate cannot be set correctly.) For I/O Interface mode Normal acknowledge 30H
	3rd byte	Command code (10H)		-
	4th byte	-		ACK for the command code byte (Note 2) Normal acknowledge 10H Negative acknowledge × 1H Communication error × 8H
	5th byte to 16th byte	Password sequence (12 bytes) (0x4000_03F4 - 0x4000_03FF)		-
	17th byte	Checksum value for the 5th to 16th bytes		-
	18th byte	-		ACK for the checksum byte (Note 2) Normal acknowledge 10H Negative acknowledge 11H Communication error 18H
	19th byte	RAM storage start address (bits 31-24)		-
	20th byte	RAM storage start address (bits 23-16)		-
	21st byte	RAM storage start address (bits 15-8)		-
	22nd byte	RAM storage start address (bits 7-0)		-
	23rd byte	RAM storage byte count (bits 15-8)		-
	24th byte	RAM storage byte count (bits 7-0)		-
	25th byte	Checksum value for the 19th to 24th bytes		-
	26th byte	-		ACK for the checksum byte (Note 2) Normal acknowledge 10H Negative acknowledge 11H Communication error 18H
	27th byte to mth byte	RAM storage data		-
	(m + 1)th byte	Checksum value for the 27th to mth bytes		-
	(m + 2)th byte	-		ACK for the checksum byte (Note 2) Normal acknowledge 10H Negative acknowledge 11H Communication error 18H
RAM	(m + 3)th byte	-		Jump to RAM storage start address

Note 1: In I/O Interface mode, the baud rate for the transfers of the 1st and 2nd bytes must be 1/16 of the desired baud rate.
Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.
Note 3: The 19th to 25th bytes must be within the RAM address range of 0xFFFF_8000 to 0xFFFF_BFFF.

Table 3.3.4 Transfer Format for the Show Flash Memory Sum Command

	Byte	Data Transferred from the Controller to the TMP1942FD	Baud Rate	Data Transferred from the Controller to the TMP1942FD
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H For I/O Interface mode 30H	Desired baud rate (Note 1)	-
	2nd byte	-		ACK for the serial operation mode byte For UART mode Normal acknowledge 86H (The boot program aborts if the baud rate cannot be set correctly.) For I/O Interface mode Normal acknowledge 30H
	3rd byte	Command data (20H)		-
	4th byte	-		AK for the command code byte (Note 2) Normal acknowledge 20H Negative acknowledge x1H Communication error x8H
	5th byte	-		SUM (upper byte)
	6th byte	-		SUM (lower byte)
	7th byte	-		Checksum value for the 5th and 6th bytes
	8th byte	(Wait for the next command code)		-

Note 1: In I/O Interface mode, the baud rate for the transfers of the 1st and 2nd bytes must be 1/16 of the desired baud rate.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte.) In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.

Table 3.3.5 Transfer Format for the Show Product Information Command (1/2)

	Byte	Data Transferred from the Controller to the TMP1942FD	Baud Rate	Data Transferred from the TMP1942FD to the Controller
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H For I/O Interface mode 30H	Desired baud rate (Note 1)	-
	2nd byte	-		ACK for the serial operation mode byte For UART mode Normal acknowledge (The boot program aborts if the baud rate cannot be set correctly.) 86H For I/O Interface mode Normal acknowledge 30H
	3rd byte	Command code (30H)		-
	4th byte	-		ACK for the command code byte (Note 2) Normal acknowledge 10H Negative acknowledge × 1H Communication error × 8H
	5th byte	-		Flash memory data (at address 0x4000_03F0)
	6th byte	-		Flash memory data (at address 0x4000_03F1)
	7th byte	-		Flash memory data (at address 0x4000_03F2)
	8th byte	-		Flash memory data (at address 0x4000_03F3)
	9th byte to 20th byte	-		Product name (12-byte ASCII code) “TX1942” from the 9th byte
	21st byte to 24th byte	-		Password comparison start address (4 bytes) F4H, 03H, 00H, 00H from the 21st byte
	25th byte to~ 28th byte	-		RAM start address (4 bytes) 00H, 80H, FFH, FFH from the 25th byte
	29th byte to 32nd byte	-		Dummy data (4 bytes) FFH, 8FH, FFH, FFH from the 29th byte
	33rd byte to 36th byte	-		RAM end address (4 bytes) FFH, BFH, FFH, FFH from the 33rd byte
	37th byte to 40th byte	-		Dummy data (4 bytes) 00H, 91H, FFH, FFH from the 37th byte
	41st byte to 44th byte	-		Dummy data (4 bytes) FFH, AFH, FFH, FFH from the 41st byte
	45th byte to 46th byte	-		Fuse information (2 bytes) 00H, 00H from the 45th byte
	47th byte to 50th byte	-		Flash memory start address (4 bytes) 00H, 00H, 00H, 00H from the 47th byte
	51st byte to 54th byte	-		Flash memory end address (4 bytes) FFH, FFH, 07H, 00H from the 51st byte
	55th byte to 56th byte	-		Flash memory block count (2 bytes) 13H, 00H from the 55th byte
	57th byte to 60th byte	-		Start address of a group of the same-size flash blocks (4 bytes) 00H, 00H, 00H, 00H from the 57th byte

Table 3.3.6 Transfer Format for the Show Product Information Command (2/2)

	Byte	Data Transferred from the Controller to the TMP1942FD	Baud Rate	Data Transferred from the TMP1942FD to the Controller
Boot ROM	61st byte to 64th byte	-		Size (in words) of the same-size flash blocks (4 bytes) 00H, 40H, 00H, 00H from the 61st byte
	65th byte	-		Number of flash blocks of the same size (1byte) 0FH
	66th byte to 69th byte	-		Start address of a group of the same-size flash blocks (4 bytes) 00H, 80H, 07H, 00H from the 66th byte
	70th byte to 73rd byte	-		Size (in words) of the same-size flash blocks (4 bytes) 00H, 20H, 00H, 00H from the 70th byte
	74th byte	-		Number of flash blocks of the same size (1 byte) 01H
	75th byte to 78th byte	-		Start address of a group of the same-size flash blocks (4 bytes) 00H, C0H, 07H, 00H from the 75th byte
	79th byte to 82nd byte	-		Size (in words) of the same-size flash blocks (4 bytes) 00H, 10H, 00H, 00H from the 79th byte
	83rd byte	-		Number of flash blocks of the same size (1byte) 01H
	84th byte to 87th byte	-		Start address of a group of the same-size flash blocks (4 bytes) 00H, E0H, 07H, 00H from the 84th byte
	88th byte to 91st byte	-		Size (in words) of the same-size flash blocks (4 bytes) 00H, 08H, 00H, 00H from the 88th byte
	92nd byte	-		Number of flash blocks of the same size (1 byte) 02H
	93rd byte	-		Checksum value for the 5th to 92nd bytes
	94th byte	(Wait for the next command data)		-

Note 1: In I/O Interface mode, the baud rate for the transfers of the 1st and 2nd bytes must be 1/16 of the desired baud rate.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.

3.3.8 Overview of the Boot Program Commands

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers the following three commands, the details of which are provided in the following subsections.

- RAM Transfer command
  The RAM Transfer command stores program code transferred from a host controller to the on-chip RAM and executes the program once the transfer is successfully completed. The maximum program size is 16 Kbytes. The RAM storage start address must be within the range.
  The RAM Transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The programming routine must utilize the flash memory command sequences described in Section 3.4.
  Before initiating a transfer, the RAM Transfer command checks a password sequence coming from the controller against that stored in the flash memory. If they do not match, the RAM Transfer command aborts.

- Show Flash Memory Sum command
  The Show Flash Memory Sum command adds the contents of the 512 Kbytes of the flash memory together. The boot program does not provide a command to read out the contents of the flash memory. Instead, the Flash Memory Sum command can be used for software revision management.

- Show Product Information command
  The Show Product Information command provides the product name, on-chip memory configuration and the like. This command also reads out the contents of the flash memory locations at addresses 0x0000_03F0 through 0x0000_03F3. In addition to the Show Flash Memory Sum command, these locations can be used for software revision management.

3.3.9 RAM Transfer Command

See Table 3.3.3.

(1) The 1st byte specifies which one of the two serial operation modes is used. For a detailed description of how the serial operation mode is determined, see 3.3.13. If it is determined as UART mode, the boot program then checks if the SIO0 is programmable to the baud rate at which the 1st byte was transferred. During the first-byte interval, the RXE bit in the SC0MOD register is cleared.

- To communicate in UART mode
  Send, from the controller to the target board, 86H in UART data format at the desired baud rate. If the serial operation mode is determined as UART, then the boot program checks if the SIO0 can be programmed to the baud rate at which the 1st byte was transferred. If that baud rate is not possible, the boot program aborts, disabling any subsequent communications.

- To communicate in I/O Interface mode
  Send, from the controller to the target board, 30H in I/O Interface data format at 1/16 of the desired baud rate. Also send the 2nd byte at the same baud rate. Then send all subsequent bytes at a rate equal to the desired baud rate.

  In I/O Interface mode, the CPU sees the serial receive pin as if it were a general input port in monitoring its logic transitions. If the baud rate of the incoming data is high or the chip's operating frequency is high, the CPU may not be able to keep up with the speed of logic transitions. To prevent such situations, the 1st and 2nd bytes must be transferred at 1/16 of the desired baud rate; then the boot program calculates 16 times that as the desired baud rate.

  When the serial operation mode is determined as I/O Interface mode, the SIO0 is configured for SCLK Input mode. Beginning with the 3rd byte, the controller must ensure that its AC timing restrictions are satisfied at the selected baud rate. In the case of I/O Interface mode, the boot program does not check the receive error flag; thus there is no such thing as error acknowledge (x8H).

(2) The 2nd byte, transmitted from the target board to the controller, is an acknowledge response to the 1st byte. The boot program echoes back the 1st byte: 86H for UART mode and 30H for I/O Interface mode.

- UART mode
  If the SIO0 can be programmed to the baud rate at which the 1st byte was transferred, the boot program programs the BR0CR and sends back 86H to the controller as an acknowledge. If the SIO0 is not programmable at that baud rate, the boot program simply aborts with no error indication.

  Following the 1st byte, the controller should allow for a time-out period of five seconds. If it does not receive 86H within the allotted time-out period, the controller should give up the communication.

  The boot program sets the RXE bit in the SC0MOD register to enable reception before loading the SIO transmit buffer with 86H.

- I/O Interface mode
  The boot program programs the SC0MOD and SC0CR registers to configure the SIO0 in I/O Interface mode (clocked by the rising edge of SCLK0), and writes 30H

to the SC0BUF. Then, the SIO0 waits for the SCLK0 signal to come from the controller. Following the transmission of the 1st byte, the controller should send the SCLK clock to the target board after a certain idle time (several microseconds). This must be done at 1/16 the desired baud rate. If the 2nd byte, which is from the target board to the controller, is 30H, then the controller should take it as a go-ahead. The controller must then deliver the 3rd byte to the target board at a rate equal to the desired baud rate. The boot program sets the RXE bit in the SC0MOD register to enable reception before loading the SIO transmit buffer with 30H.

(3) The 3rd byte, which the target board receives from the controller, is a command. The code for the RAM Transfer command is 10H.

(4) The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined – they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

If the 3rd byte is equal to any of the command codes listed in Table 3.3.2, the boot program echoes it back to the controller. When the RAM Transfer command was received, the boot program echoes back a value of 10H and then branches to the RAM Transfer routine. Once this branch is taken, a password check is done. Password checking is detailed in Section 3.3.14.

If the 3rd byte is not a valid command, the boot program sends back x1H to the controller and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined – they hold the same values as the upper four bits of the previously issued command.

(5) The 5th to 16th bytes, which the target board receives from the controller, are a 12-byte password. The 5th byte is compared to the contents of address 0x0000_03F4 in the flash memory; the 6th byte is compared to the contents of address 0x0000_03F5 in the flash memory; likewise, the 16th byte is compared to the contents of address 0x0000_03FF in the flash memory. If the password checking fails, the RAM transfer routine sets the password error flag.

(6) The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in detail in Section 3.3.16.

(7) The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes.

First, the RAM Transfer routine checks for a receive error in the 5th to 17th bytes. If there was a receive error, the boot program sends back 18H and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer

routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 5th to 17th bytes must result in zero (with the carries dropped). If it is not zero, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

Finally, the RAM Transfer routine examines the result of the password check. The following two cases are treated as a password error. In these cases, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

- Irrespective of the result of the password comparison, all of the 12 bytes of a password in the flash memory are the same value other than FFH.
- Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

When all the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (10H) to the controller.

(8) The 19th to 22nd bytes, which the target board receives from the controller, indicate the start address of the RAM region where subsequent data (e.g., a flash programming routine) should be stored. The 19th byte corresponds to bits 31-24 of the address, and the 22nd byte corresponds to bits 7-0 of the address.

(9) The 23rd and 24th bytes, which the target board receives from the controller, indicate the number of bytes that will be transferred from the controller to be stored in the RAM. The 23rd byte corresponds to bits 15-8 of the number of bytes to be transferred, and the 24th byte corresponds to bits 7-0 of the number of bytes.

(10) The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in detail in Section 3.3.16.

(11) The 26th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th to 25th bytes of data.

First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there was a receive error, the RAM Transfer routine sends back 18H and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 19th to 25th bytes must result in zero (with the carries dropped). If it is not zero, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

- The RAM storage start address must be within the range of 0xFFFF_8000 to 0xFFFF_BFFF.

When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (10H) to the controller.

(12) The 27th to mth bytes from the controller are stored in the on-chip RAM of the TMP1942FD. Storage begins at the address specified by the 19th to 22nd bytes and continues for the number of bytes specified by the 23rd and 24th bytes.

(13) The (m+1)th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in detail in Section 3.3.16.

(14) The (m+2) byte is an acknowledge response to the 27th to (m+1)th bytes.

First, the RAM Transfer routine checks for a receive error in the 27th to (m+1)th bytes. If there was a receive error, the RAM Transfer routine sends back 18H and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m+1)th bytes must result in zero (with the carries dropped). If it is not zero, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again. When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (10H) to the controller.

(15)If the (m+2)th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes in 32-bit ISA mode.

Note: At this time, r29 (sp) points to 0xFFFF_9100. Do not transfer program control from the RAM back to the boot ROM.

3.3.10 Show Flash Memory Sum Command

See Table 3.3.4.

(1) The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

(2) The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Flash Memory Sum command is 20H.

(3) The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined – they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

If the 3rd byte is equal to any of the command codes listed in Table 3.3.2, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 20H and then branches to the Show Flash Memory Sum routine.

If the 3rd byte is not a valid command, the boot program sends back x1H to the controller and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined – they hold the same values as the upper four bits of the previously issued command.

(4) The Show Flash Memory Sum routine adds all the bytes of the flash memory together. The 5th and 6th bytes, transmitted from the target board to the controller, indicate the upper and lower bytes of this total sum, respectively. For details on sum calculation, see Section 3.3.15.

(5) The 7th byte is a checksum value for the 5th and 6th bytes. To calculate the checksum value, add the 5th and 6th bytes together, drop the carry and take the two's complement of the sum. Transmit this checksum value from the controller to the target board.

(6) The 8th byte is the next command code.

3.3.11 Show Product Information Command

See Table 3.3.5.

(1) The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

(2) The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Product Information command is 30H.

(3) The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined – they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

If the 3rd byte is equal to any of the command codes listed in Table 3.3.2, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 30H and then branches to the Show Flash Memory Sum routine.

If the 3rd byte is not a valid command, the boot program sends back x1H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are undefined – they hold the same values as the upper four bits of the previously issued command.

(4) The 5th to 8th bytes, transmitted from the target board to the controller, are the data read from addresses 0x4000_03F0 to 0x4000_03F3 in the flash memory. Software version management is possible by storing a software ID in these locations.

(5) The 9th to 20th bytes, transmitted from the target board to the controller, indicate the product name, which is "TX1942" in ASCII code.

(6) The 21st to 24th bytes, transmitted from the target board to the controller, indicate the start address of the flash memory area containing the password, i.e., F4H, 03H, 00H, and 00H.

(7) The 25th to 28th bytes, transmitted from the target board to the controller, indicate the start address of the on-chip RAM, i.e., 00H, 80H, FFH, FFH.

(8) The 29th to 32nd bytes, transmitted from the target board to the controller, are dummy data, i.e., FFH, 8FH, FFH, FFH.

(9) The 33rd to 36th bytes, transmitted from the target board to the controller, indicate the end address of the on-chip RAM, i.e., FFH, BFH, FFH, FFH.

(10) The 37th to 44th bytes, transmitted from the target board to the controller, are dummy data.

(11) The 45th and 46th bytes, transmitted from the target board to the controller, indicate the presence or absence of the security and protect bits and whether the flash memory is divided into blocks. Bit 0 indicates the presence or absence of the security bit; it is 0 if the security bit is available. Bit 1 indicates the presence or absence of the protect bits; it is 0 if the protect bits are available. If bit 2 is 0, it indicates that the flash memory is divided into blocks. The remaining bits are undefined. The 45th and 46th bytes are 00H, 00H.

(12) The 47th to 50th bytes, transmitted from the target board to the controller, indicate the start address of the on-chip flash memory, i.e., 00H, 00H, 00H, and 00H.

(13) The 51st to 54th bytes, transmitted from the target board to the controller, indicate the end address of the on-chip flash memory, i.e., FFH, FFH, 07H, and 00H.

(14) The 55th and 56th bytes, transmitted from the target board to the controller, indicate the number of flash blocks available, i.e., 13H, 00H.

(15) The 57th to 92nd bytes, transmitted from the target board to the controller, contain information about the flash blocks.
Flash blocks of the same size are treated as a group. Information about the flash blocks indicate the start address of a group, the size of the blocks in that group (in words) and the number of the blocks in that group.

The 57th to 65th bytes are the information about the 32-Kbyte blocks (Block 0 to Block 14). The 66th to 74th bytes are the information about the 16-Kbyte block (Block 15). The 75th to 83rd bytes are the information about the 8-Kbyte block (Block 16). The 84th to 92nd bytes are the information about the 4-Kbyte blocks (Block 17 and Block 18). See Table 3.3.5 for the values of bytes transmitted.

(16) The 93rd byte, transmitted from the target board to the controller, is a checksum value for the 5th to 92nd bytes. The checksum value is calculated by adding all these bytes together, dropping the carries and taking the two's complement of the total sum.

(17) The 94th byte is the next command code.

3.3.12 Acknowledge Responses

The boot program represents processing states with specific codes. Table 3.3.7 to Table 3.3.9 show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. Bit 3 of the code indicates a receive error. Bit 0 indicates an invalid command error, a checksum error or a password error. Bit 1 and bit 2 are always 0. Receive error checking is not done in I/O Interface mode.

Table 3.3.7 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
86H	The SIO can be configured to operate in UART mode. (See Note)
30H	The SIO can be configured to operate in I/O Interface mode.

Note: If the serial operation mode is determined as UART, the boot program checks if the SIO can be programmed to the baud rate at which the operation mode byte was transferred. If that baud rate is not possible, the boot program aborts without sending back any response.

Table 3.3.8 ACK Response to the Command Byte

Return Value	Meaning
x8H (See Note)	A receive error occurred while getting a command code.
x1H (See Note)	An undefined command code was received.
10H	The RAM Transfer command was received.
20H	The Show Flash Memory Sum command was received.
30H	The Show Product Information command was received.

Note: The upper four bits of the ACK response are the same as those of the previous command code.

Table 3.3.9 ACK Response to the Checksum Byte

Return Value	Meaning
18H	A receive error occurred.
11H	A checksum or password error occurred.
10H	The checksum was correct.

3.3.13 Determination of a Serial Operation Mode

The first byte from the controller determines the serial operation mode. To use UART mode for communications between the controller and the target board, the controller must first send a value of 86H at a desired baud rate to the target board. To use I/O Interface mode, the controller must send a value of 30H at 1/16 the desired baud rate. Figure shows the waveforms of the first byte.

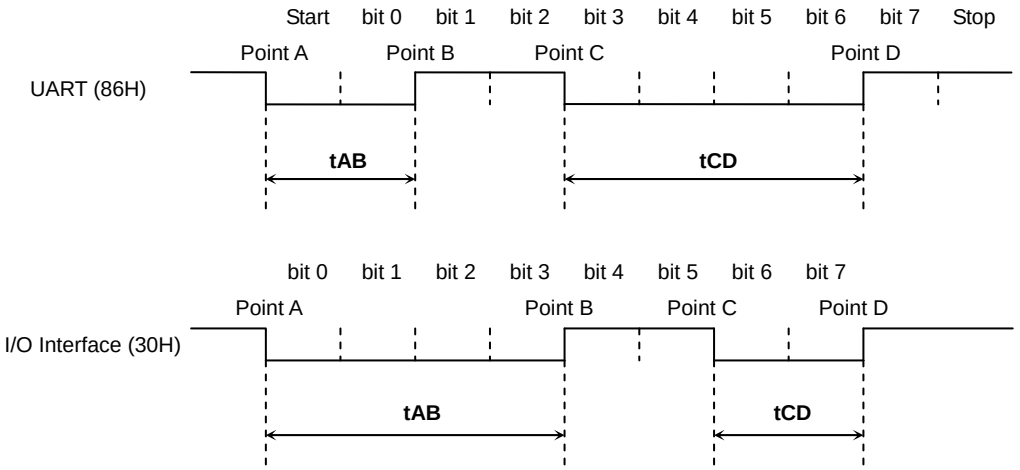


Figure 3.3.4 Serial Operation Mode Byte

After $\overline{\text{RESET}}$ is released, the boot program monitors the first serial byte from the controller with the SIO reception disabled, and calculates the intervals of tAB, tAC and tAD. Figure 3.3.5 shows a flowchart describing the steps to determine the intervals of tAB, tAC and tAD. As shown in the flowchart, the boot program captures timer counts each time a logic transition occurs in the first serial byte. Consequently, the calculated tAB, tAC and tAD intervals are bound to have slight errors. If the transfer goes at a high baud rate, the CPU might not be able to keep up with the speed of logic transitions at the serial receive pin. In particular, I/O Interface mode is more prone to this problem since its baud rate is generally much higher than that for UART mode. To avoid such a situation, the controller should send the first serial byte at 1/16 the desired baud rate.

The flowchart in Figure 3.3.6 shows how the boot program distinguishes between UART and I/O Interface modes. If the length of tAB is equal to or less than the length of tCD, the serial operation mode is determined as UART mode. If the length of tAB is greater than the length of tCD, the serial operation mode is determined as I/O Interface mode. Bear in mind that if the baud rate is too high or the timer operating frequency is too low, the timer resolution will be coarse, relative to the intervals between logic transitions. This becomes a problem due to inherent errors caused by the way in which timer counts are captured by software; consequently the boot program might not be able to determine the serial operation mode correctly.

For example, the serial operation mode may be determined to be I/O Interface mode when the intended mode is UART mode. To avoid such a situation, when UART mode is utilized, the controller should allow for a time-out period within which it expects to receive an echo-back (86H) from the target board. The controller should give up the communication if it fails to get that echo-back within the allotted time. When I/O Interface mode is utilized, once the first serial byte has been transmitted, the controller should send the SCLK clock after a certain idle time to get an acknowledge response. If the received acknowledge response is not 30H, the controller should give up further communications.

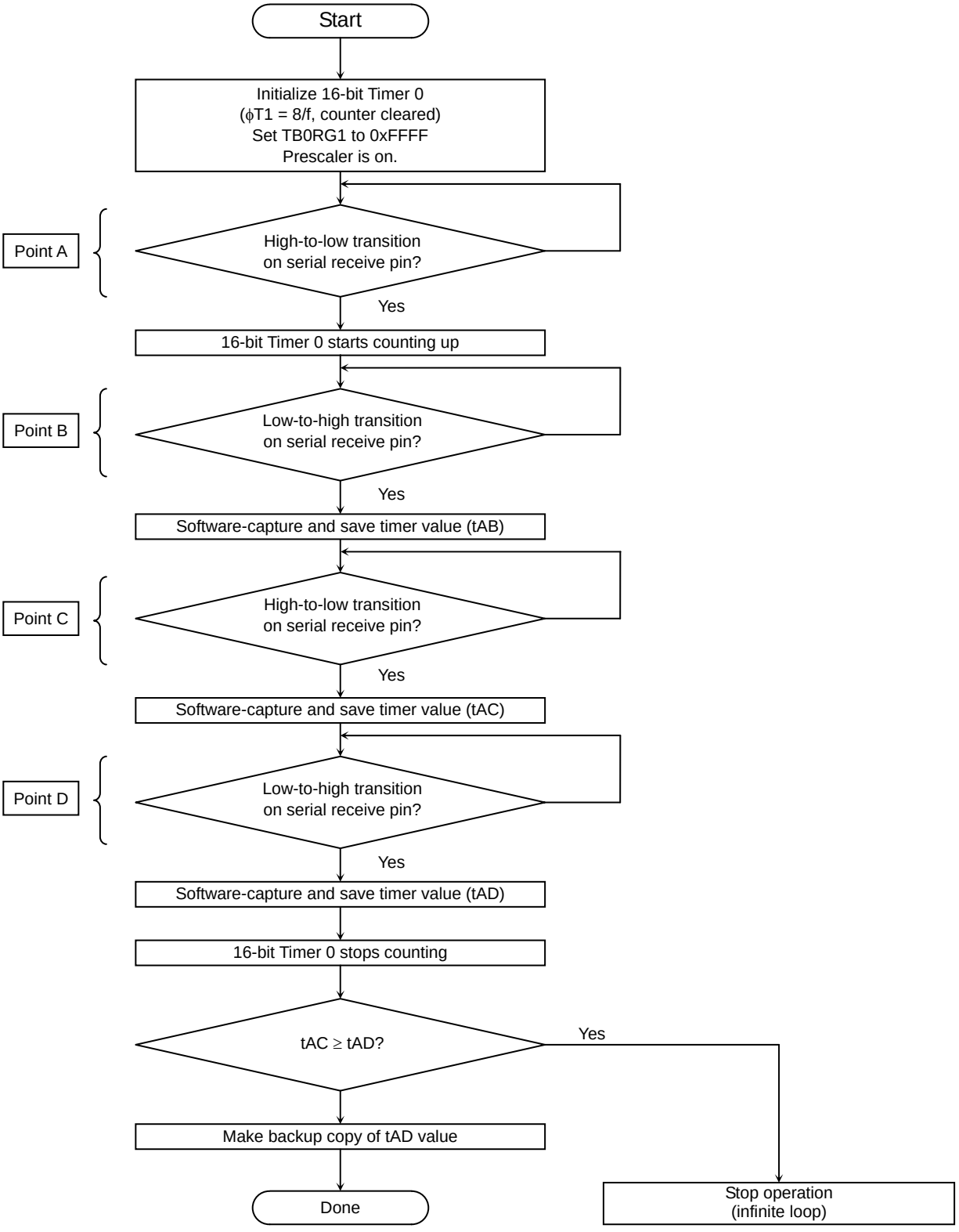


Figure 3.3.5 Serial Operation Mode Byte Reception Flow

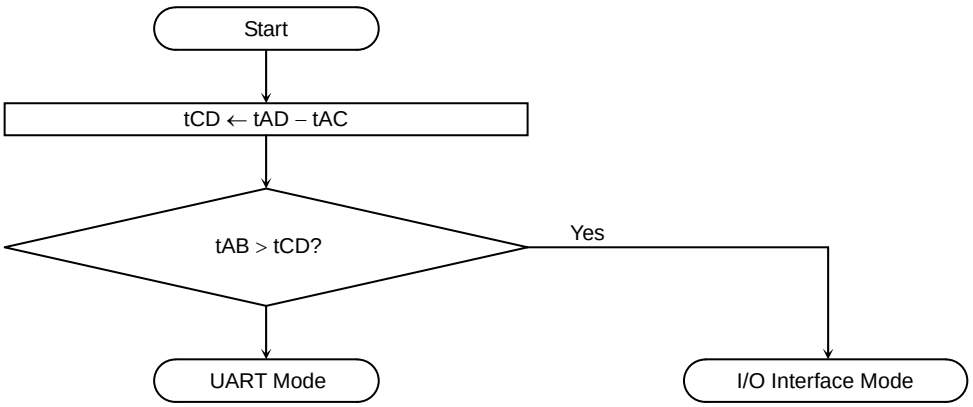


Figure 3.3.6 Serial Operation Mode Determination Flow

3.3.14 Password

The RAM Transfer command (10H) causes the boot program to perform a password check. Following an echo-back of the command code, the boot program checks the contents of the 12-byte password area (0x4000_03F4 to 0x4000_03FF) within the flash memory. If all these address locations contain the same bytes of data other than FFH, a password area error occurs. In this case, the boot program returns an error acknowledge (11H) in response to the checksum byte (the 17th byte), regardless of whether the password sequence sent from the controller is all FFHs.

The password sequence received from the controller (the 5th to 16th bytes) is compared to the password stored in the flash memory. Table 3.3.10 show how they are compared byte by byte. All of the 12 bytes must match to pass the password check. Otherwise, a password error occurs, which causes the boot program to return an error acknowledge in response to the checksum byte (the 17th byte).

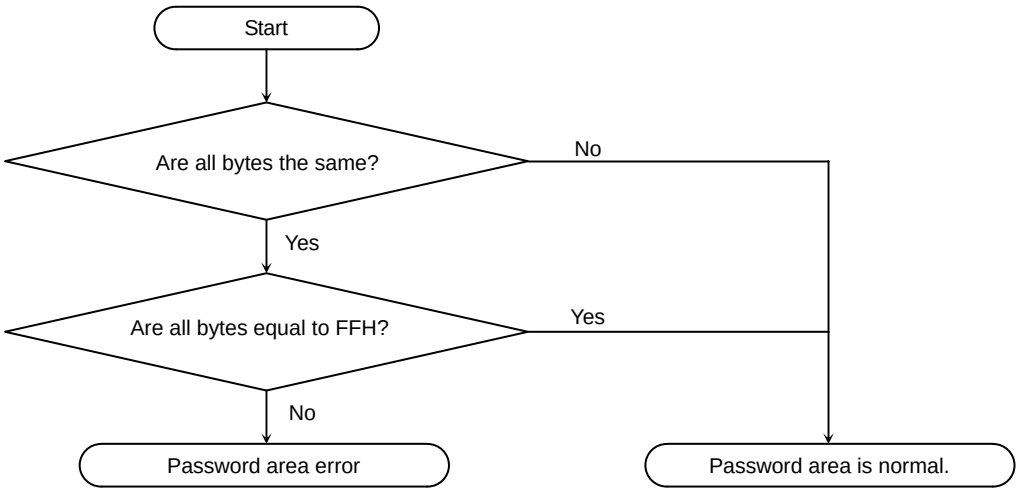


Figure 3.3.7 Password Area Check Flow

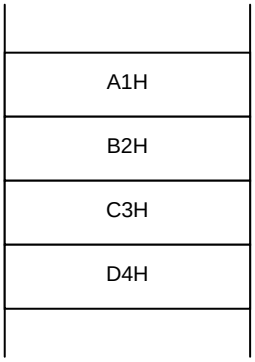
Table 3.3.10 Relationship between Received Bytes and Flash Memory Locations

Received Byte	Compared Flash Memory Data
5th byte	Address 0x4000_03F4
6th byte	Address 0x4000_03F5
7th byte	Address 0x4000_03F6
8th byte	Address 0x4000_03F7
9th byte	Address 0x4000_03F8
10th byte	Address 0x4000_03F9
11th byte	Address 0x4000_03FA
12th byte	Address 0x4000_03FB
13th byte	Address 0x4000_03FC
14th byte	Address 0x4000_03FD
15th byte	Address0x4000_03FE
16th byte	Address 0x4000_03FF

3.3.15 Calculation of the Show Flash Memory Sum Command

The Show Flash Memory Sum command adds all 512 Kbytes of the flash memory together and provides the total sum as a word quantity. The sum is sent to the controller with the upper eight bits first, followed by the lower eight bits.

Example:



For the interest of simplicity, assume the depth of the flash memory is four locations. Then the sum of the four bytes is calculated as:

A1H + B2H + C3H + D4H = 02EAH

Hence, 02H is first sent to the controller, followed by

3.3.16 Checksum Calculation

The checksum byte for a series of bytes of data is calculated by adding the bytes together, dropping the carries, and taking the two's complement of the total sum. The Show Flash Memory Sum command and the Show Product Information command perform the checksum calculation. The controller must perform the same checksum operation in transmitting checksum bytes.

Example:

Assume the Show Flash Memory Sum command provides the upper and lower bytes of the sum as E5H and F6H. To calculate the checksum for a series of E5H and F6H:

(1) Add the bytes together.
E5H + F6H = 1DBH

(2) Drop the carry.

(3) Take the two's complement of the sum, and that is the checksum byte.
0 – DBH = 25H

3.3.17 General Boot Program Flowchart

Figure 3.3.8 shows an overall flowchart of the boot program.

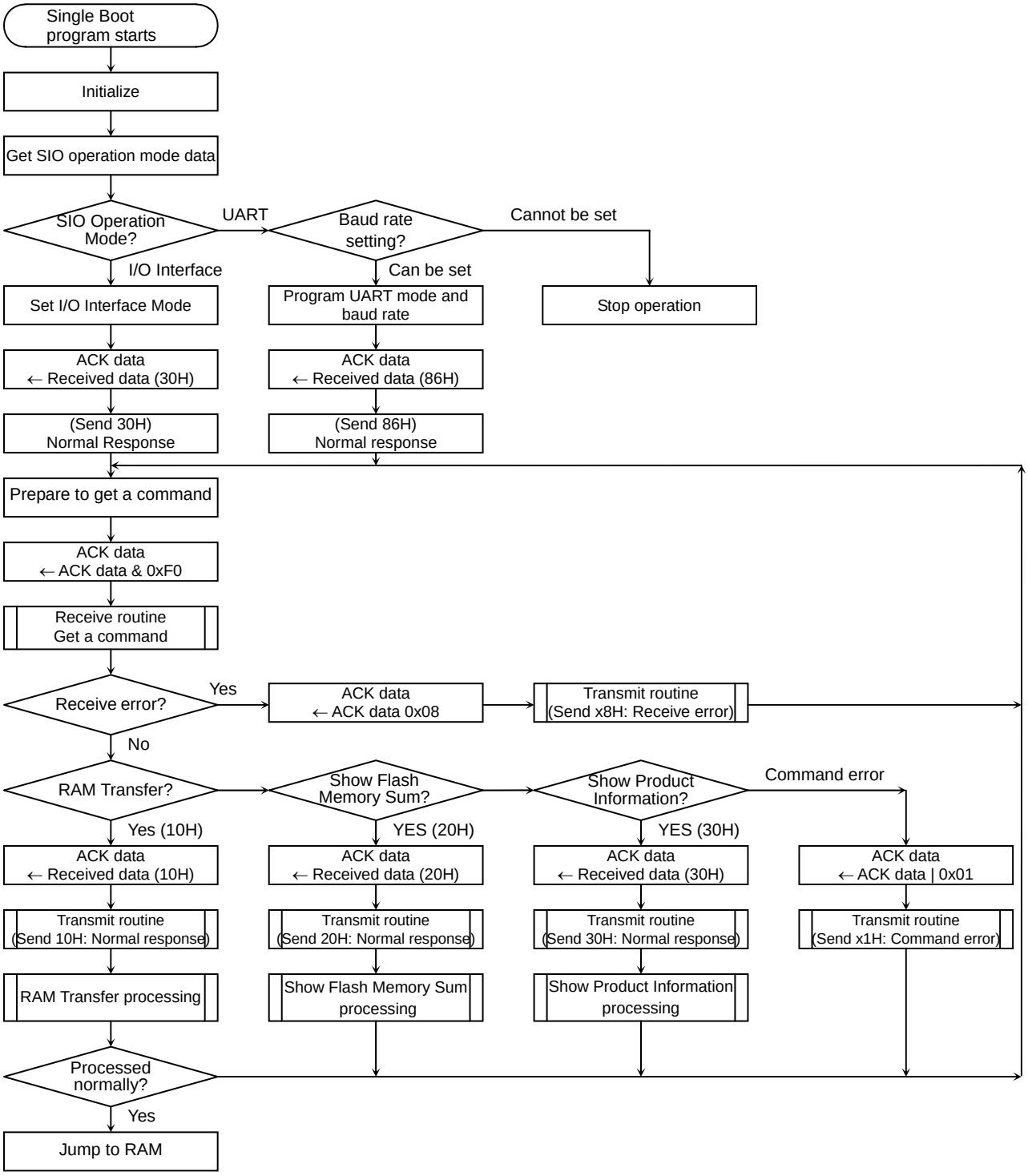


Figure 3.3.8 Overall Boot Program Flow

3.4 On-Board Programming and Erasure

The TMP1942FD flash memory is command set compatible with the JEDEC EEPROM standard, with a few exceptions. In User Boot mode and Single Boot mode (the RAM Transfer command), the flash memory can be programmed and erased by the CPU executing software commands. It is the user's responsibility to create a program/erase routine. Because the flash memory cannot be read while it is being programmed or erased, the program/erase routine must be executed out of the on-chip RAM or an external memory device.

3.4.1 Key Features

The TMP1942FD flash memory commands are in principle compatible with the standard JEDEC commands. For program/erase operations, the system can issue a command sequence to the flash memory by using CPU instructions such as LD. After the command sequence is written, the flash memory does not require the system to provide further controls or timings. The flash memory initiates the embedded program or erase algorithm automatically. The entire flash memory or one or more flash blocks can be erased at a time.

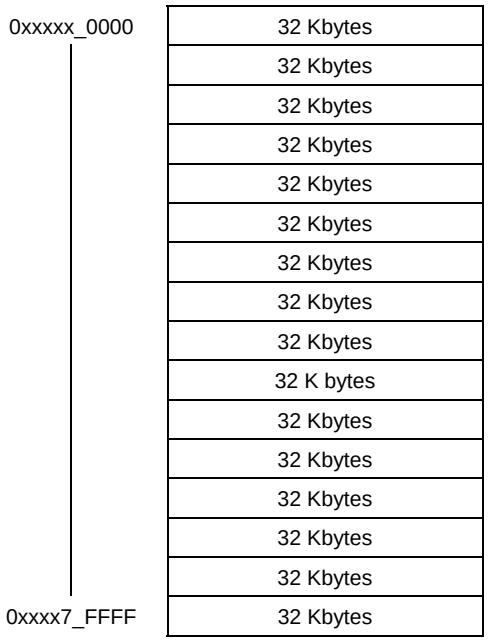
Table 3.4.1 Flash Memory Features

Feature	Description
Auto Program	Programs and verifies the desired addresses in longword units automatically.
Auto Chip Erase	Erases and verifies the entire memory array automatically.
Auto Block Erase	Erases and verifies all memory locations in the selected block automatically.
Auto Multi-Block Erase	Erases and verifies all memory locations in multiple selected blocks automatically.
Write operation status	Provides several status bits such as the Data Polling bit and Toggle bit, which can be used to determine whether a program or erase operation is complete or in progress.
Security feature	Prevents intrusive access to the flash memory while in Programmer mode. When the security feature is turned off, the entire memory array is erased and verified automatically, regardless of whether a given block is protected or not.
Block protection	Disables both program and erase operations in any block.

Bear in mind that, due to the on-chip CPU interface, the TMP1942FD uses addresses different from those of the standard flash command sequences. Unless otherwise noted, programming is done in longword (32-bit) units; thus the longword (32-bit) load instruction should be used to write to the flash array. The byte load instruction can be used to issue commands to the flash memory.

The program/erase operations in Programmer mode are very similar to those of the on-board programming modes, with a few exceptions such as the data bus width. Refer to Section 3.4.2 for a description of the program and erase operations in Programmer mode.

3.4.1.1 Block Architecture



x: Depends on the TMP1942FD operation mode.

Figure 3.4.1 Flash Memory Block Architecture

3.4.1.2 CPU-to-Flash Interface

Figure 3.4.2 illustrates the internal iterface between the CPU and the flash memory in on-board programming modes. The diagram does not show the actual logic network; instead it is only a conceptual depiction of the CPU-to-flash interface.

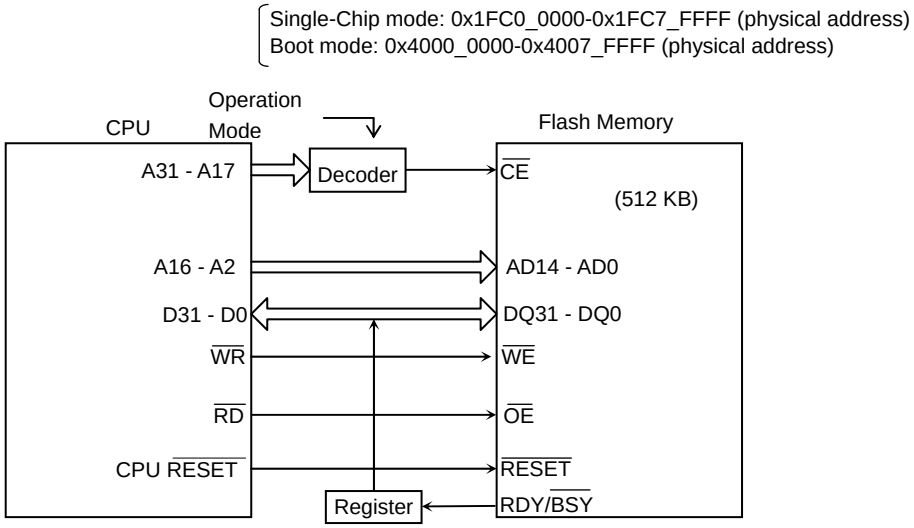


Figure 3.4.2 Internal CPU-to-Flash Interface

3.4.1.3 Read Mode and Embedded Operation Mode

The flash memory of the TMP1942FD has the following two modes of operation:

- Read mode in which array data is read
- Embedded Operation mode in which the flash array is programmed or erased

The flash memory enters Embedded Operation mode when a valid command is executed in Read mode. In Embedded Operation mode, array data cannot be read.

3.4.1.4 Reading Array Data

The flash memory is automatically set to reading array data upon CPU reset after device power-up and after an embedded operation is successfully completed. If an embedded operation terminated abnormally or the flash memory is required to return to Read mode, the Read/Reset command (software reset) or hardware reset is used.

3.4.1.5 Writing Commands

The operations of the flash memory are selected by commands or command sequences written into the internal command register. This uses the same mechanism as for JEDEC-standard EEPROMs. Commands are made up of data sequences written at specific addresses via the command register (see Table 3.4.3 and Table 3.4.4).

Commands are written via DQ0-DQ7 except the fourth (read) cycle in the Read/Reset command sequence, the fourth (write) cycle in the Auto Program command sequence and the fourth (write) cycle in the Verify Block Protect command sequence. Thus, commands can be provided byte by byte.

The command sequence being written can be canceled by issuing the Read/Reset command between sequence cycles. The Read/Reset command clears the command register and resets the flash memory to Read mode. Invalid command sequences also cause the flash memory to clear the command register and return to Read mode.

3.4.1.6 Reset

- Read/Reset command (software reset)
  The flash memory does not return to Read mode if an embedded operation terminated abnormally. In this case, the Read/Reset command must be issued to put the flash memory back in Read mode. The Read/Reset command may also be written between sequence cycles of the command being written to clear the command register.

- Hardware reset
  As shown in Figure 3.4.2, the flash memory has a reset pin, which is connected to the reset signal of the CPU. When the system drives the $\overline{\text{RESET}}$ pin to V_{IL} or when certain events such as a watchdog timer time-out causes a CPU reset, the flash memory immediately terminates any operation in progress and is reset to Read mode.

The Read/Reset command is also tied to the $\overline{\text{RESET}}$ pin to reset the flash memory to Read mode. The embedded operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

For a description of the hardware reset operation, see Section 3.2.1, *Reset Operation*. When a valid reset is achieved, the CPU reads the Reset exception vector from the flash memory and services the Reset exception.

3.4.1.7 Auto Program Command

A bit must be programmed to change its state from a 1 to a 0. A bit cannot be programmed from a 0 back to a 1. Only an erase operation can change a 0 back to a 1.

In User Boot mode and the RAM Transfer command of Single Boot mode, the Auto Program command programs the desired addresses in 32-bit (longword) units. The Auto Program command requires four bus cycles; the program address and data are written in the fourth cycle, upon completion of which the program operation will commence. As programming is performed in 32-bit units, the program address must be a multiple of four.

Writing data shorter than a 32-bit longword requires special considerations for the bits that are not to be altered. The longword in the memory does not need to be in the erased state prior to programming. If the longword is in the erased state, a 32-bit write must be performed, with all the bits not to be altered set to 1. If the longword is not in the erased state, it must be loaded into the CPU first to modify necessary bits, and the modified word must be written to the flash memory.

Examples:

1) When a longword location is in the erased state
   To program the least-significant byte to 55H, 0xFFFF_FF55 must be written to the longword address.
2) When a longword location is not in the erased state and contains 0x8888_88FF
   To program the least-significant byte to AAH, 0x8888_88AA must be written to the longword address.

The Auto Program command executes a sequence of internally timed events to program the desired bits of the addressed memory location and verify that the desired bits are sufficiently programmed. The system can determine the status of the programming operation by using write status flags (See Table 3.4.6).

Any commands written during the programming operation are ignored. A hardware reset immediately terminates the programming operation. The programming operation that was interrupted should be re-initiated once the flash memory is ready to accept another command because data may be corrupted.

The block protection feature disables programming operations in any block. If an attempt is made to program a protected block, the Auto Program command does nothing; the flash memory returns to Read mode in approximately 3 μs after the completion of the fourth bus cycle of the command sequence.

When the embedded Auto Program algorithm is complete, the flash memory returns to Read mode.

If any failure occurs during the programming operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to

reset the flash memory or a hardware reset to reset the whole chip. In case of a programming failure, it is recommended to replace the chip or discontinue the use of the failing flash block.

3.4.1.8 Auto Chip Erase Command

The Auto Chip Erase command requires six bus cycles. After completion of the sixth bus cycle, the Auto Chip Erase operation will commence immediately. The embedded Auto Chip Erase algorithm automatically preprograms the entire memory for an all-0 data pattern prior to the erase; then it automatically erases and verifies the entire memory for an all-1 data pattern. The system can determine the status of the chip erase operation by using write status flags (see Table 3.4.6).

Any commands written during the chip erase operation are ignored. A hardware reset immediately terminates the chip erase operation. The chip erase operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables erase operations in any block. The Auto Chip Erase algorithm erases the unprotected blocks and ignores the protected blocks. If all the blocks are protected, the Auto Chip Erase command does nothing; the flash memory returns to Read mode in approximately 100 μs after the completion of the sixth bus cycle of the command sequence.

When the embedded Auto Chip Erase algorithm is complete, the flash memory returns to Read mode.

If any failure occurs during the erase operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block. The failing block can be identified by means of the Block Erase command.

3.4.1.9 Auto Block Erase and Auto Multi-Block Erase Commands

The Auto Block Erase command requires six bus cycles. A time-out begins from the completion of the command sequence. After a time-out, the erase operation will commence. The embedded Auto Block Erase algorithm automatically preprograms the selected block for an all-0 data pattern, and then erases and verifies that block for an all-1 data pattern. During the time-out period, additional block addresses and Auto Block Erase commands may be written (see Section 3.4.1.16).

Any command other than Auto Block Erase during the time-out period resets the flash memory to Read mode. The block erase time-out period is 50 μs. The system may read DQ3 to determine whether the time-out period has expired. The block erase timer begins counting upon completion of the sixth bus cycle of the Auto Block Erase command sequence. The system can determine the status of the erase operation by using write status flags (see Table 3.4.6).

Any commands written during the block erase operation are ignored. A hardware reset immediately terminates the block erase operation. The block erase operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables erase operations in any block. The Auto Block Erase algorithm erases the unprotected blocks and ignores the protected blocks. If all the

selected blocks are protected, the Auto Block Erase algorithm does nothing; the flash memory returns to Read mode in approximately 100 μs after the final bus cycle of the command sequence. When the embedded Auto Block Erase algorithm is complete, the flash memory returns to Read mode.

If any failure occurs during the erase operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block. If any failure occurred during the multi-block erase operation, the failing block can be identified by running Auto Block Erase on each of the blocks selected for multi-block erasure.

3.4.1.10 Block Protect Command

The block protection feature disables both program and erase operations in any block. The effects of the program and erase commands on the protected blocks are summarized below.

Table 3.4.2 Effects of the Program and Erase Commands on the Protected Blocks

Command	Operation
Program command on a protected block	No programming operation is performed, and the flash memory automatically returns to Read mode.
Block Erase command on a protected block	No erase operation is performed, and the flash memory automatically returns to Read mode.
Chip Erase command when all the blocks are protected	No erase operation is performed, and the flash memory automatically returns to Read mode.
Chip Erase command when any blocks are protected	Only the unprotected blocks are erased. Upon completion, the flash memory automatically returns to Read mode.
Multi-Block Erase command when any blocks are protected	Only the unprotected blocks are erased. Upon completion, the flash memory automatically returns to Read mode.

The Block Protect command requires 10 bus cycles. The address of the block to be protected is internally latched in the seventh cycle. Then, allow an interval of 4 μs to elapse before providing data for the eighth cycle, which enables writing to the protection control circuitry. Next, allow an interval of at least 100 μs to elapse before providing data for the ninth cycle. This terminates writing to the protection control circuitry. Finally, allow an interval of 8 μs to elapse and provide data for the tenth cycle to complete the command.

Note that the block protect operation is not verified automatically. The Verify Block Protect command must be written to verify the protect status after executing Block Protect. If the desired block is not in the protected state, the Block Protect command sequence must be re-initiated. Figure 3.4.8 illustrates the algorithm for the Block Protect command.

Any commands written during the Block Protect algorithm are ignored. A hardware reset immediately terminates the block protect operation. The Block Protect command that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence.

3.4.1.11 Block Unprotect Command

The Block Unprotect command unprotects all blocks simultaneously. All blocks must be protected before executing the Block Unprotect command.

The Block Unprotect command requires 10 bus cycles. After the seventh bus write cycle is completed (where the address is don't care), allow an interval of 4 μs to elapse before

providing data for the eighth cycle, which enables the block unprotect operation. Next, allow an interval of at least 10 μs to elapse before providing data for the ninth cycle. This terminates the block unprotect operation. Finally, allow an interval of 8 μs to elapse and provide data for the tenth cycle to complete the command.

Note that the block unprotect operation is not verified automatically. The Verify Block Protect command must be written to verify the protect status after executing Block Unprotect. If the desired block is not unprotected, the Block Unprotect command sequence must be re-initiated.

Any commands written during the block unprotect operation are ignored. The hardware reset immediately terminates the block unprotect operation. The Block Protect command that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence. Use the Verify Block Protect command to verify the protect status of a block.

3.4.1.12 Verify Block Protect Command

The Verify Block Protect command is used to verify the protect status of a block. Verify Block Protect is a four-bus-cycle operation. The address of the block to be verified is given in the fourth cycle. Any address within the block range will suffice, provided A0 = A1 = A2 = A3 = 0, A4 = 1 and A6 = 0. To get correct data, a 32-bit read must be performed. Use the last read as valid data. If the selected block is protected, a value of 0x0000_0001 is returned. If the selected block is not protected, a value of 0x0000_0000 is returned. Additional blocks may be verified by repeating the fourth bus read cycle.

The Verify Block Protect command does not return the flash memory to Read mode. Either the Read/Reset command or a hardware reset is required to reset the flash memory to Read mode or to write the next command.

3.4.1.13 Write Operation Status

As shown in Table 3.4.6 , the flash memory provides several flag bits to determine the status of an embedded operation: DQ7, DQ5 and DQ3. These status bits can be read during an embedded operation using the same timing as for Read mode. The flash memory automatically returns to Read mode when an embedded operation completes.

During the embedded program operation, the system must provide the program address (with A0 = 0 and A1 = 0) to read valid status information. During the embedded erase operation, the system must provide an address (with A0 =0 and A1 = 0) within any of the blocks selected for erasure to read valid status information.

- DQ7 (Data Polling)

  The Data Polling bit, DQ7, indicates to the host system the status of the embedded operation. Data Polling is valid after the final bus write cycle of an embedded command sequence.

  When the embedded Program algorithm is in progress, an attempt to read the flash memory will produce the complement of the data last written to DQ7. Upon completion of the embedded Program algorithm, an attempt to read the flash memory will produce the true data last written to DQ7. Therefore, the system can use DQ7 to determine whether the embedded Program algorithm is in progress or complete.

  When the embedded Erase algorithm is in progress, an attempt to read the flash memory will produce a 0 at the DQ7 output. Upon completion of the embedded Erase algorithm, the flash memory will produce a 1 at the DQ7 output.

  If there is a failure during an embedded operation, DQ7 continues to output the

same value. Thus, DQ7 must always be polled in conjunction with the Exceeded Timing Limits (DQ5) flag. Figure 3.4.6 shows the DQ7 polling algorithm.
The flash memory disables address latching when an embedded operation is complete. Data polling must be performed with a valid programmed address or an address within any of the non-protected blocks selected for erasure.

- DQ5 (Exceeded Timing Limits)
  DQ5 produces a 0 while the program or erase operation is in progress normally. DQ5 produces a 1 to indicate that the program or erase time has exceeded the specified internal limit. This is a failure condition that indicates the program or erase cycle was not successfully completed.
  The DQ5 failure condition also appears if the system tries to program a 1 to a location that was previously programmed to a 0. Only an erase operation can change a 1 back to a 0. In this case, the embedded Program algorithm halts the operation. Once the operation has exceeded the timing limits, DQ5 will indicate a 1. Note that this is not a device failure condition since the flash memory was used incorrectly.
  Under both these conditions, the flash memory remains locked in Embedded Operation mode. The system must issue the Read/Reset command to return the flash memory to Read mode.

- DQ3 (Block Erase Timer)
  After the completion of the sixth bus cycle of the Auto Block Erase command sequence, the block erase time-out window of 50 μs begins. The erase operation will begin after the time-out has expired. When the time-out is complete and the erase operation has begun, DQ3 switches from 0 to 1. If DQ3 is 0, the flash memory will accept additional Auto Block Erase commands. Each time an Auto Block Erase command is written, the time-out window is reset. To ensure that the command has been accepted, the system should check DQ3 prior to and following each Auto Block Erase command. If DQ3 is 1 on the second status check, the command might not have been accepted.

3.4.1.14 Flash Control/Status Register

This is an 8-bit register that indicates the Ready/Busy status of an embedded algorithm and controls the security feature.

FLCS (0xFFFF_E520)

	7	6	5	4	3	2	1	0
Bit Symbol					—	RDY/BSY	—	FSE
Read/Write					R/W	R	R/W	R/W
Reset Value					0	1	0	0
Function					Must be written as “0”.	Ready/Busy 0: Embedded algorithm is in progress. 1: Embedded algorithm is complete	Must be written as “0”.	Security control 0: Access flash memory array 1: Access security logic

Bit 2: Ready/Busy Flag Bit
In Programmer mode, the ALE pin functions as the RDY/BSY pin. The host system can monitor the state of this pin to determine whether an embedded algorithm is in progress or complete. The CPU can poll the RDY/BSY bit in the FLCS register for the same purpose. The RDY/BSY bit is cleared to 0 when the flash memory is

actively erasing or programming. The RDY/BSY bit is set to 1 when an embedded operation has completed and the flash memory is ready to accept the next command. If any failure occurs during the program or erase operation, this bit remains cleared. A hardeare reset sets this bit.
The RDY/BSY bit is cleared upon completion of the final bus write cycle of an embedded operation command, with one exception. In the case of the Auto Block Erase command, this bit is cleared after the time-out has expired. Any command is ignored while the RDY/BSY bit is cleared.

Bit 0: Flash Security Enable (FSE)
The FSE bit is used to enable and disable the security feature. After a reset, this bit is cleared. Under this condition, the program and erase commands access the memory array. To turn on the security feature, set the FSE bit and write the Auto Security On command. Thereafter, the FSE bit must be cleared to enable access to the memory array. To turn off the security feature, set the FSE bit and write the Auto Security Off command.

Note: The Flash Control/Status register must be accessed as a 32-bit quantity.

Figure 3.4.3 Flash Control/Status Register

3.4.1.15 Flash Security

The TMP1942FD flash memory supports not only on-board programming but also programming using a general-purpose programmer. Therefore, the TMP1942FD flash memory provides a security feature to prevent intrusive access to the flash memory while in Programmer mode.

The TMP1942FD has a security bit apart from the flash array. Programming this security bit disables access to the flash array. The paragraphs that follow describe the methods to secure and unsecure the flash memory. As in the case with a flash programming routine, the security control routine must also be placed and executed outside of the flash memory—either the on-chip RAM or an external memory device.

- Securing the flash memory (Disabling read accesses)
  Securing the flash memory disables a general-purpose programmer to read its contents. To turn on the security feature, once programming is complete, set the FSE bit in the FLCS register and write the Auto Security On command. After the completion of the fourth bus cycle of that command sequence, the embedded Security On algorithm automatically programs and verifies the security bit.
  Any commands written during the embedded operation are ignored. A hardware reset immediately terminates the embedded operation. The FSE bit must not be altered throughout the embedded operation.
  When the embedded algorithm completes, the flash memory automatically returns to Read mode. In on-board operating modes, the CPU can read the flash memory even if the security is on; clear the FSE bit to 0 to enable access to the flash array.
  If any failure occurs during the embedded operation, the flash memory remains locked in Embedded Operation mode and does not return to Read mode. The system can determine the status of the embedded operation by using write status flags. Note that this is a security bit failure. If the flash memory needs to be secured, the chip should be replaced. When the security is on, any reads by programming equipment will always returns a value of 0x0098.

- Unsecuring the flash memory (Enabling read accesses)
  The security feature is designed to disable reads of the flash memory by programming equipment. While the TMP1942FD is soldered on a board, the CPU can always read the flash memory regardless of whether or not the security is on.

Since the flash memory is placed under control of a user's application program in on-board operating modes, it is not easy for third parties to perform intrusive access to the flash memory. Therefore, within the confines of a board, the flash memory does not need to be secured.

To turn off the security feature, set the FSE bit in the FLCS register to 1 and write the Auto Security Off command. After the completion of the sixth bus cycle of that command sequence, the embedded Security Off algorithm automatically erases and verifies the entire flash array, and then erases and verifies the security bit.

Any commands written during the embedded operation are ignored. A hardware reset immediately terminates the embedded operation. In this case, if any erase operation is in progress, data may be corrupted. The FSE bit must not be altered throughout the embedded operation.

When the embedded algorithm completes, the flash memory automatically returns to Read mode. If any on-board operation is subsequently required, clear the FSE bit to 0 to enable access to the flash array.

If any failure occurs during the embedded operation, the flash memory remains locked in Embedded Operation mode and does not return to Read mode. The system can determine the status of the embedded operation by using write status flags. If a failure occurs in the memory array, the security bit is not erased. In this case, the security is left on. The chip should be replaced if a memory array or security bit failure occurs.

The Auto Security Off command erases the flash array prior to turning off the security feature. Even if a given block is protected, it is unconditionally erased, but the protect status of that block remains unchanged. The Auto Security Off and Auto Chip Erase command sequences are the same. The only difference is that the Auto Security Off command requires the FSE bit to be set to 1 before the command is written. The Auto Block Erase command cannot turn off the security feature even when the FSE bit is set. If the Auto Block Erase command is written when the security is on, no block will be erased and the operation is immediately terminated.

3.4.1.16 Command Definitions

Table 3.4.3 On-Board Programming Mode Command Definitions

Command Sequence	Cycles Required	Bus Cycles									
		1st Cycle (Write)		2nd Cycle (Write)		3rd Cycle (Write)		4th Cycle (Read/Write)		5th Cycle (Write)	
		Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data
Read/Reset	1	0xXXX0	F0H								
Read/Reset	3	0xAAA8	AAH	0x5554	55H	0xAAA8	F0H	RA	RD		
Auto Program	4	0xAAA8	AAH	0x5554	55H	0xAAA8	A0H	PA	PD		
Auto Chip Erase	6	0xAAA8	AAH	0x5554	55H	0xAAA8	80H	0xAAA8	AAH	0x5554	55H
Auto Block Erase	6	0xAAA8	AAH	0x5554	55H	0xAAA8	80H	0xAAA8	AAH	0x5554	55H
Block Protect	10	0xAAA8	AAH	0x5554	55H	0xAAA8	9AH	0xAAA8	AAH	0x5554	55H
Verify Block Protect	4	0xAAA8	AAH	0x5554	55H	0xAAA8	90H	BPA	BD		
Auto Security On (Note 1)	4	0xAAA8	AAH	0x5554	55H	0xAAA8	A0H	0x0000	98H		

(Continued from above)

Command Sequence	Cycles Required	Bus Cycles									
		6th Cycle (Write)		7th Cycle (Write)		8th Cycle (Write)		9th Cycle (Write)		10th Cycle (Write)	
		Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data
Read/Reset	1										
Read/Reset	3										
Auto Program	4										
Auto Chip Erase	6	0xAAA8	10H								
Auto Block Erase	6	BA	30H								
Block Protect	10	0xAAA8	9AH	BA	00H	0xXXX0	00H	0xXXX0	00H	0xXXX0	00H
Verify Block Protect	4										
Auto Security On (Note 1)	4										

Note 1: Before executing the command sequence, set the FSE bit in the Flash Control/Status (FLCS) register to enable access to the security bit.

Note 2: There must be an interval of at least two instructions between writing one command sequence and another.

The addresses to be provided by the CPU are shown below.

Table 3.4.4 Addresses Provided by the CPU

Command Address	CPU Addresses: A23-A0																
Addr.	A23~A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0xXXX0	Flash memory block	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0
0x0000		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xAAA8		1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
0x5554		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0

- F0H, AAH, 55H, A0H, 80H, 10H, 30H:
  Command data. Write command data as a byte quantity.
- RA: Read Address
  RD: Read Data
- PA: Program Address
  PD: Program Data
  The address must be a multiple of four. Write data on a word-by-word basis.
- BA: Block Address (BA0-BA6)
  Refer to Table 3.4.5.
- BPA: Verify Block Protect Address
  BD: Block Protect Data
  The address of the block to be verified can be any of the addresses within the block, with A6=0, A4=1, A3=0, A1=0 and A0=0. If a block is protected, a value of 0x0000_0001 will be returned. If a block is not protected, a value of 0x0000_0000 will be returned.

Table 3.4.5 Block Erase Addresses

Block	Address Range		Size
	User Boot Mode	Boot Mode	
BA0	0x1FC0_0000-0x1FC0_7FFF (or x4000_0000-0x4000_7FFF)	0x1FC0_0000-0x1FC0_7FFF	32 Kbytes
BA1	0x1FC0_8000-0x1FC0_FFFF (or 0x4000_8000-0x4000_FFFF)	0x1FC0_8000-0x1FC0_FFFF	32 Kbytes
BA2	0x1FC1_0000-0x1FC1_7FFF (or 0x4001_0000-0x4001_7FFF)	0x1FC1_0000-0x1FC1_7FFF	32 Kbytes
BA3	0x1FC1_8000-0x1FC1_FFFF (or 0x4001_8000-0x4001_FFFF)	0x1FC1_8000-0x1FC1_FFFF	32 Kbytes
BA4	0x1FC2_0000-0x1FC2_7FFF (or 0x4002_0000-0x4002_7FFF)	0x1FC2_0000-0x1FC2_7FFF	32 Kbytes
BA5	0x1FC2_8000-0x1FC2_FFFF (or 0x4002_8000-0x4002_FFFF)	0x1FC2_8000-0x1FC2_FFFF	32 Kbytes
BA6	0x1FC3_0000-0x1FC3_7FFF (or 0x4003_0000~0x4003_7FFF)	0x1FC3_0000-~0x1FC3_7FFF	32 Kbytes
BA7	0x1FC3_8000-0x1FC3_FFFF (or0x4003_8000~0x4003_FFFF)	0x1FC3_8000-0x1FC3_FFFF	32 Kbytes
BA8	0x1FC4_0000-0x1FC4_7FFF (or 0x4004_0000-0x4004_7FFF)	0x1FC4_0000-0x1FC4_7FFF	32 Kbytes
BA9	0x1FC4_8000-0x1FC4_FFFF (or0x4004_8000-0x4004_FFFF)	0x1FC4_8000-0x1FC4_FFFF	32 Kbytes
BA10	0x1FC5_0000-0x1FC5_7FFF (or0x4005_0000-0x4005_7FFF)	0x1FC5_0000-0x1FC5_7FFF	32 Kbytes
BA11	0x1FC5_8000-0x1FC5_FFFF (or 0x4005_8000-0x4005_FFFF)	0x1FC5_8000-0x1FC5_FFFF	32 Kbytes
BA12	0x1FC6_0000-0x1FC6_7FFF (or 0x4006_0000~0x4006_7FFF)	0x1FC6_0000-0x1FC6_7FFF	32 Kbytes
BA13	0x1FC6_8000-0x1FC6_FFFF (or 0x4006_8000-0x4006_FFFF)	0x1FC6_8000-0x1FC6_FFFF	32 Kbytes
BA14	0x1FC7_0000-0x1FC7_7FFF (or 0x4007_0000~0x4007_7FFF)	0x1FC7_0000-0x1FC7_7FFF	32 Kbytes
BA15	0x1FC7_8000-0x1FC7_FFFF (or 0x4007_8000-0x4007_FFFF)	0x1FC7_8000-0x1FC7_FFFF	32 Kbytes

The address of the block to be erased can be any of the addresses within that block with A0=0 and A1=0. For example, to select BA0 in User Boot mode, provide any address in the range of 0x1FC0_0000 to 0x1FC0_7FFF.

Table 3.4.6 Write Status Flags

Status		D7 (DQ7)	D5 (DQ5)	D3 (DQ3)
Embedded operation in progress	Auto Program	$\overline{DQ7}$	0	0
	Auto Erase (during the time-out window)	0	0	0
	Auto Erase	0	0	1
Time-out in embedded operation	Auto Program	$\overline{DQ7}$	1	1
	Auto Erase	0	1	1

Note: D31 to D8, D4 and D2 to D0 are don't cares.

3.4.1.17 Embedded Algorithms

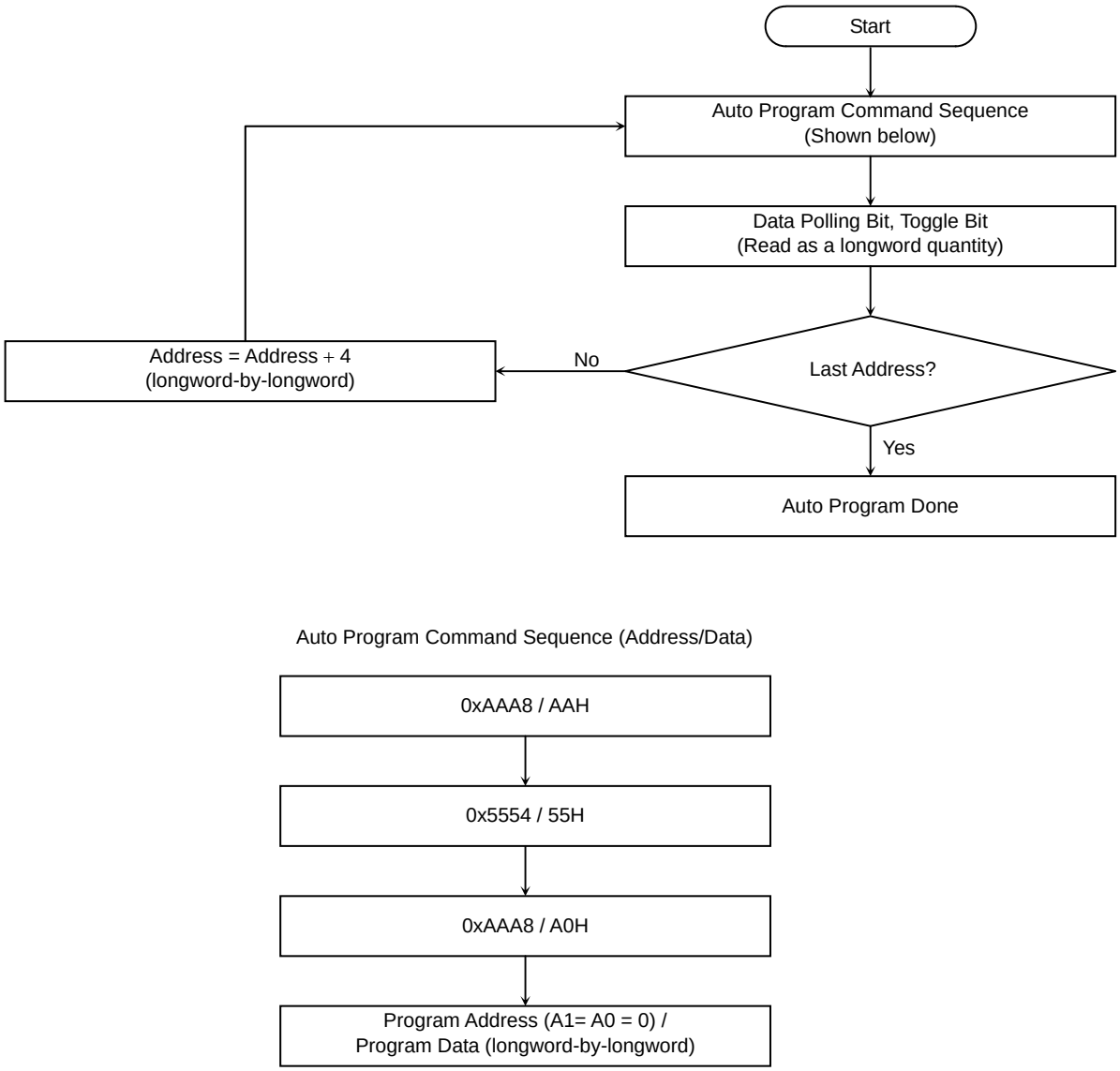


Figure 3.4.4 Auto Program Operation

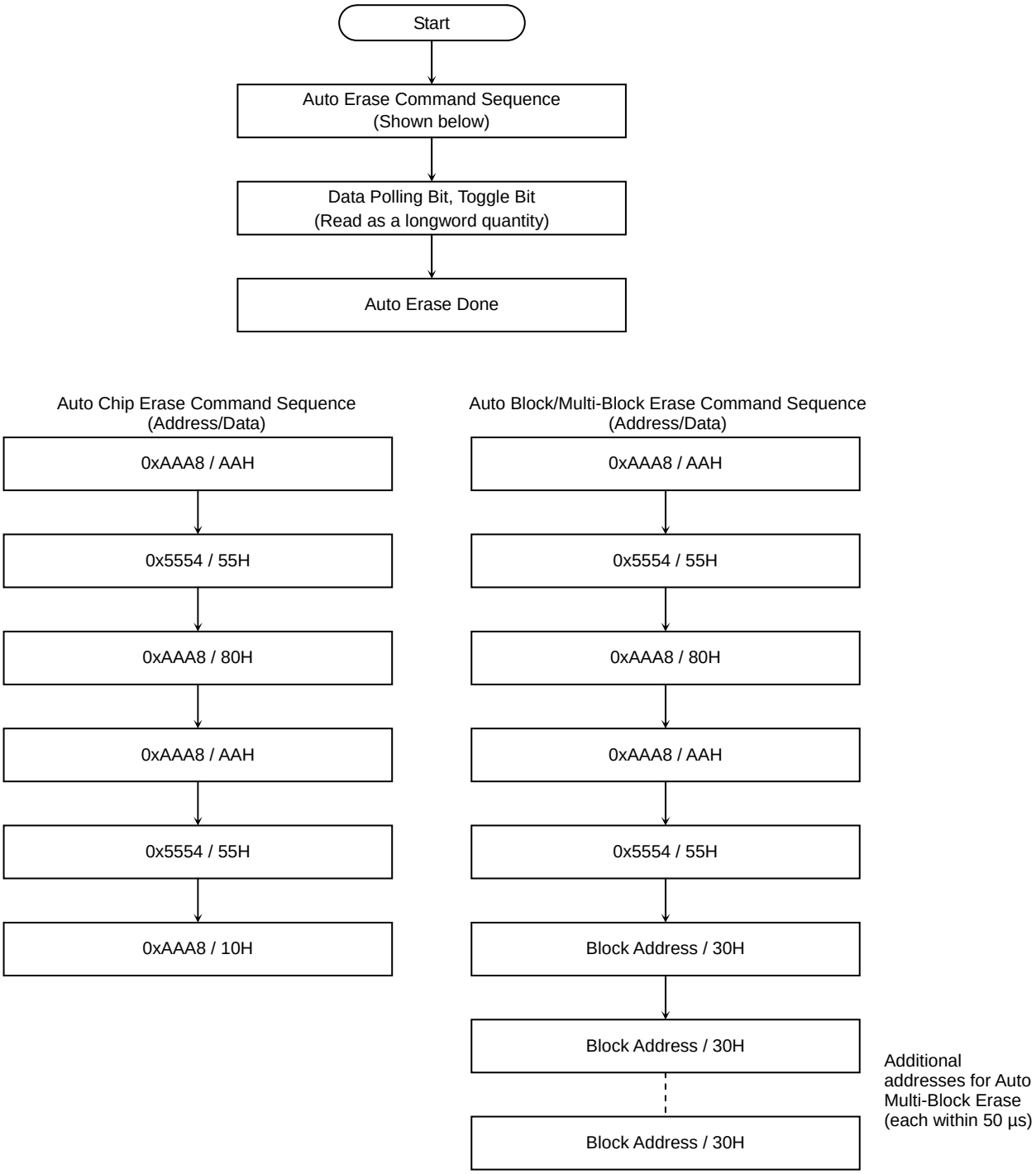


Figure 3.4.5 Auto Erase Operation

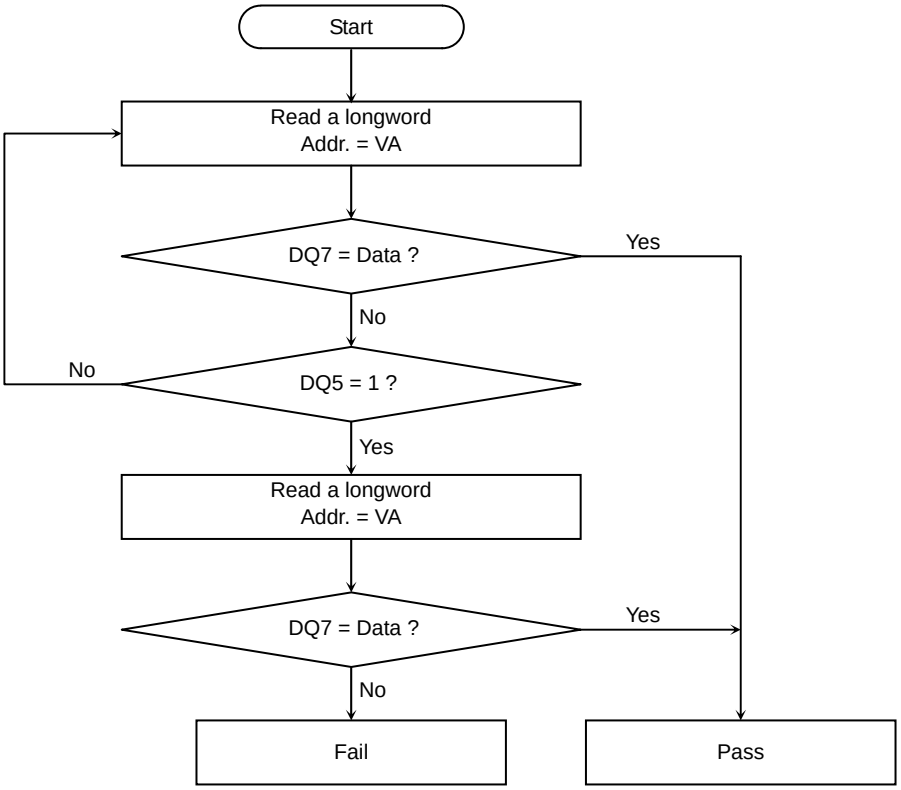
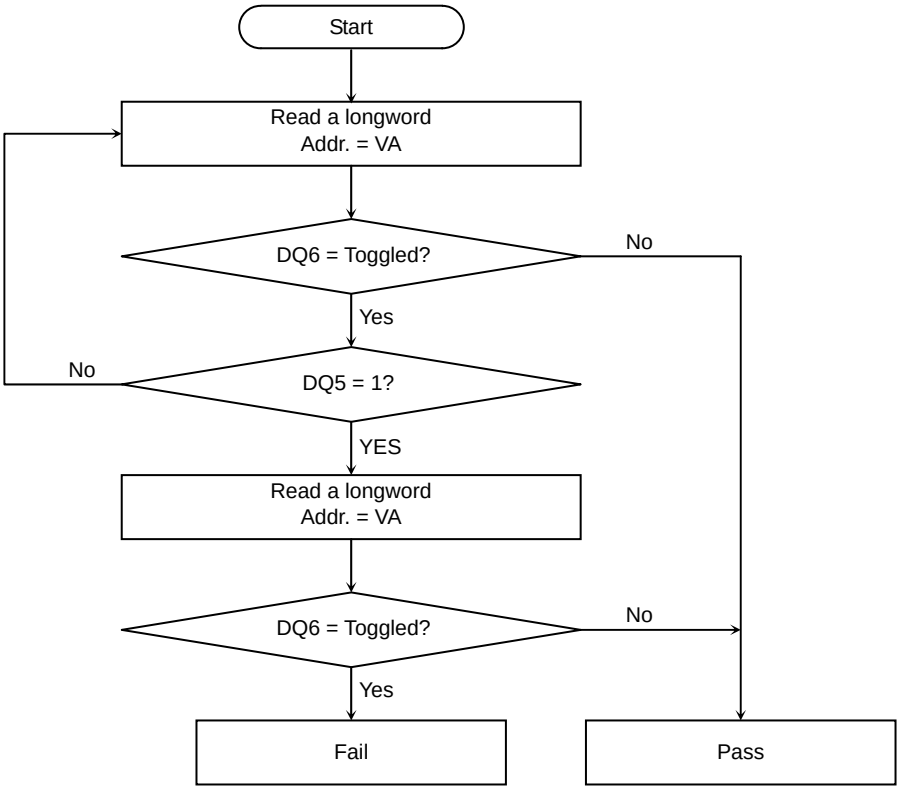


Figure 3.4.6 Data Polling (DQ7) Algorithm



VA: Address being programmed in Auto Program
Any address in the flash memory in Auto Chip Erase
Any address in the selected block in Auto Block Erase

Figure 3.4.7 Toggle Bit (DQ6) Algorithm

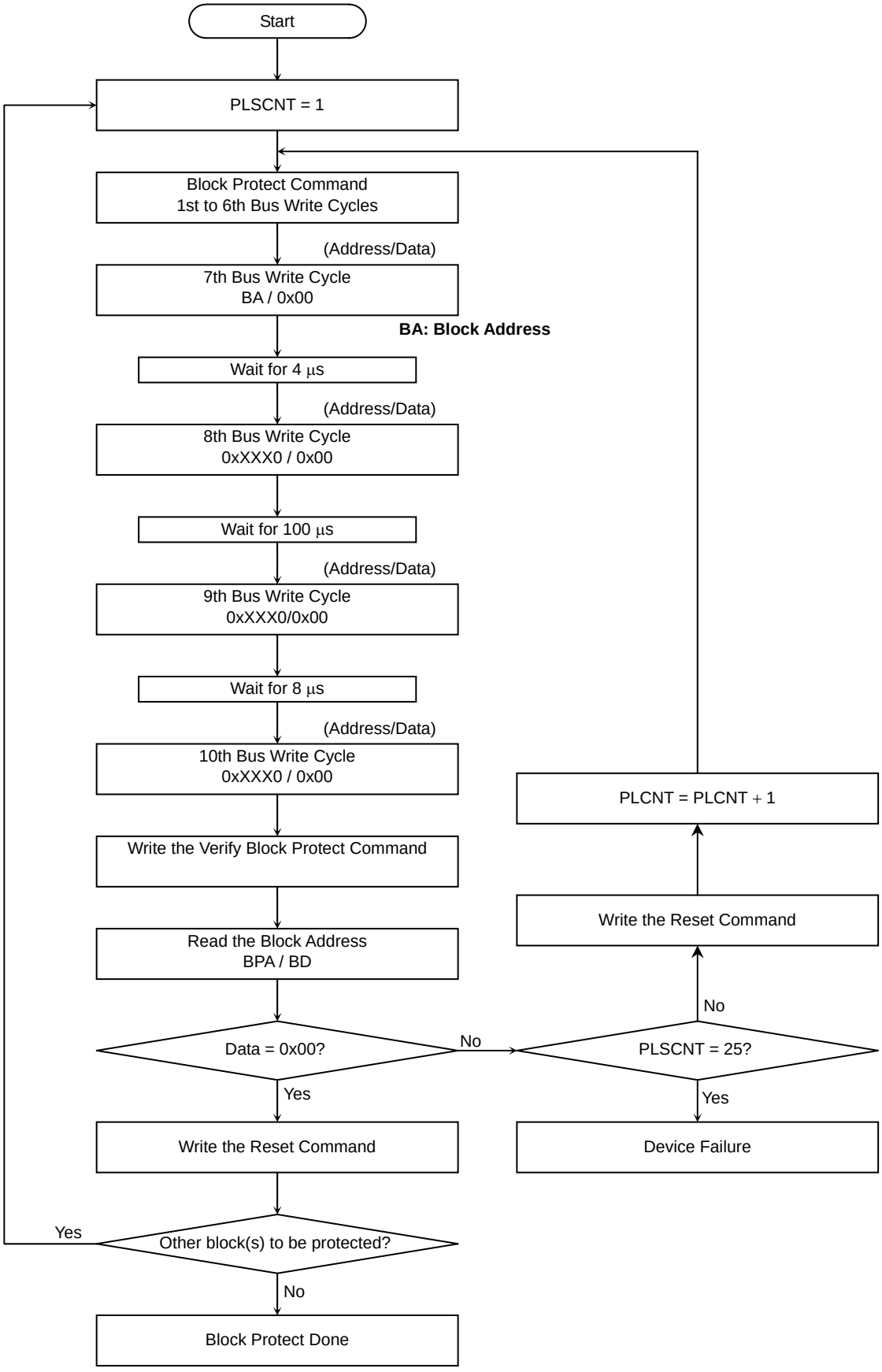


Figure 3.4.8 Block Protect Operation

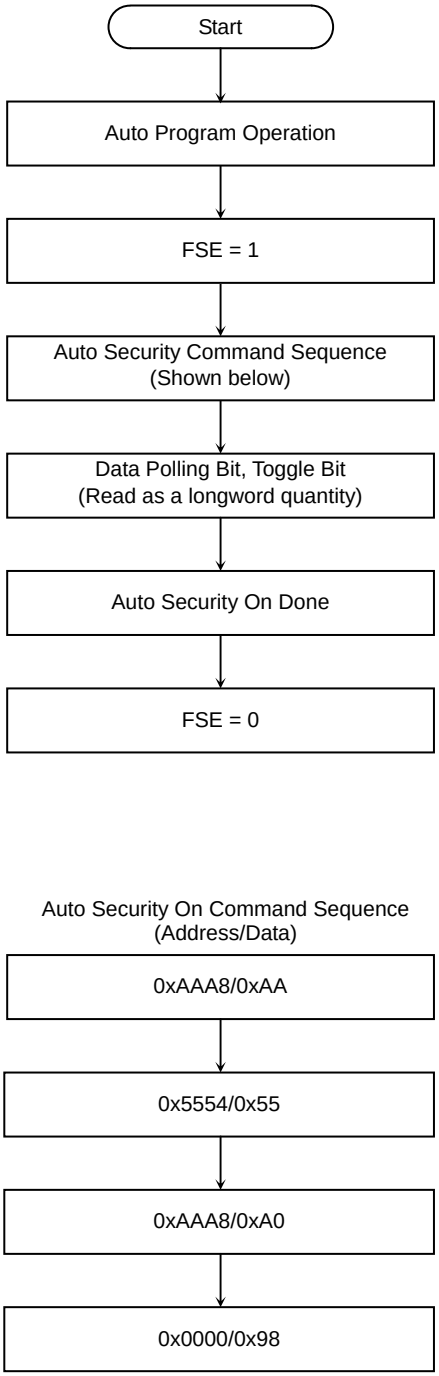


Figure 3.4.9 Auto Security On Operation

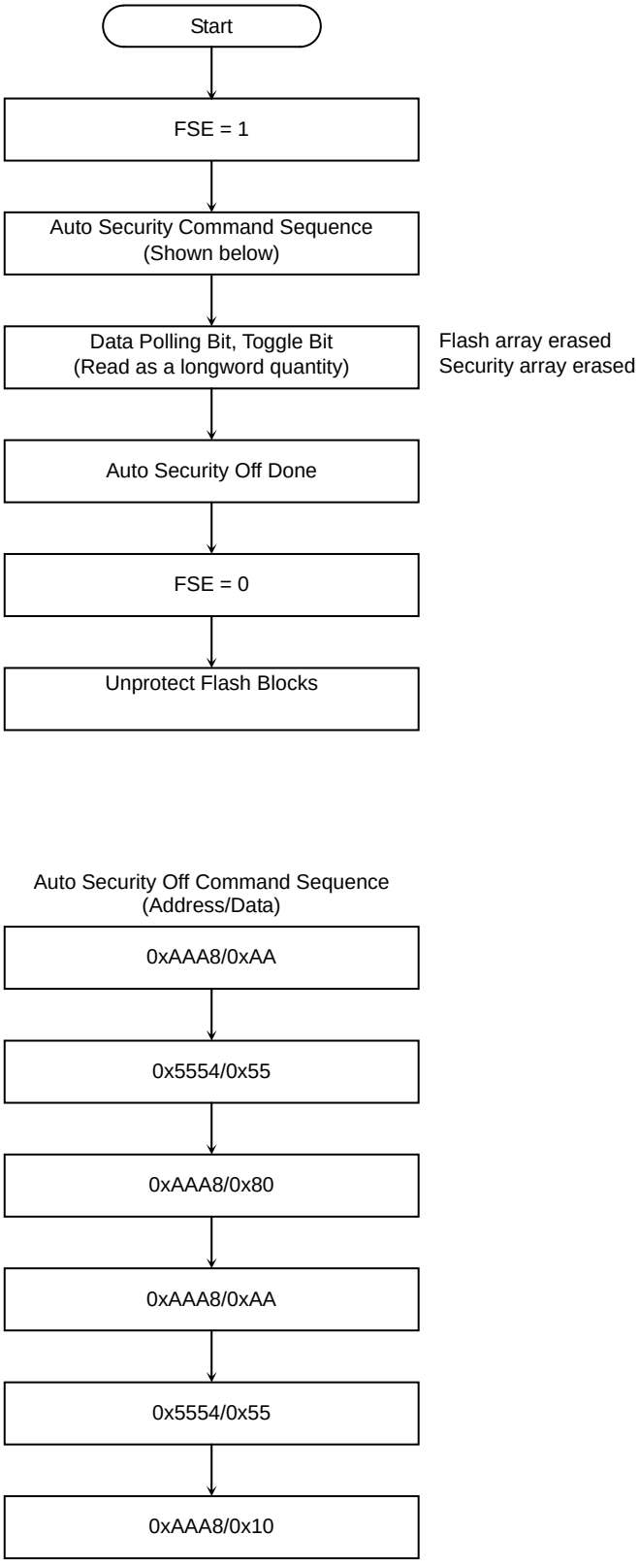


Figure 3.4.10 Auto Security Off Operation

3.4.2 Writer Mode

3.4.2.1 Pin Settings

The TMP1942FD is placed in Writer mode by holding the $\overline{\text{RESET}}$, BW0, P41, P42 and P43 pins at logic 0 and the BW1 and P40 pins at logic 1. In Writer mode, the flash memory can be read, erased and programmed using a general-purpose EPROM programmer. For instructions about the settings of the remaining pis, see section 3.4.2.3, *Pin Functions and Settings.* Figure 3.4.11 below shows the pin settings for Writer mode.

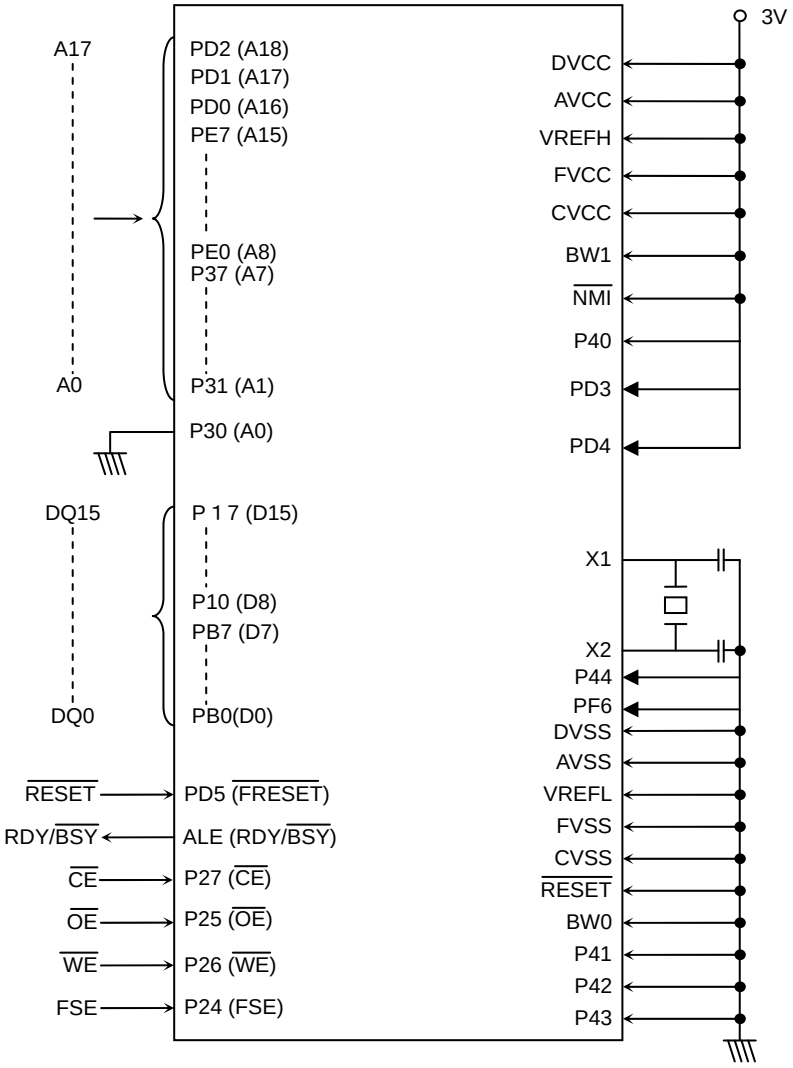


Figure 3.4.11 Pin Settings for Programmer Mode

3.4.2.2 Memory Maps

Figure 3.4.12 shows a comparison of memory maps in Single-Chip (Normal) mode and Writer mode. In Writer mode, the on-chip flash memory is mapped to physical addresses 0x0000_0000 through 0x0007_FFFF. In Writer mode, all reads and writes use 16-bit accesses aligned on an even-byte boundary.

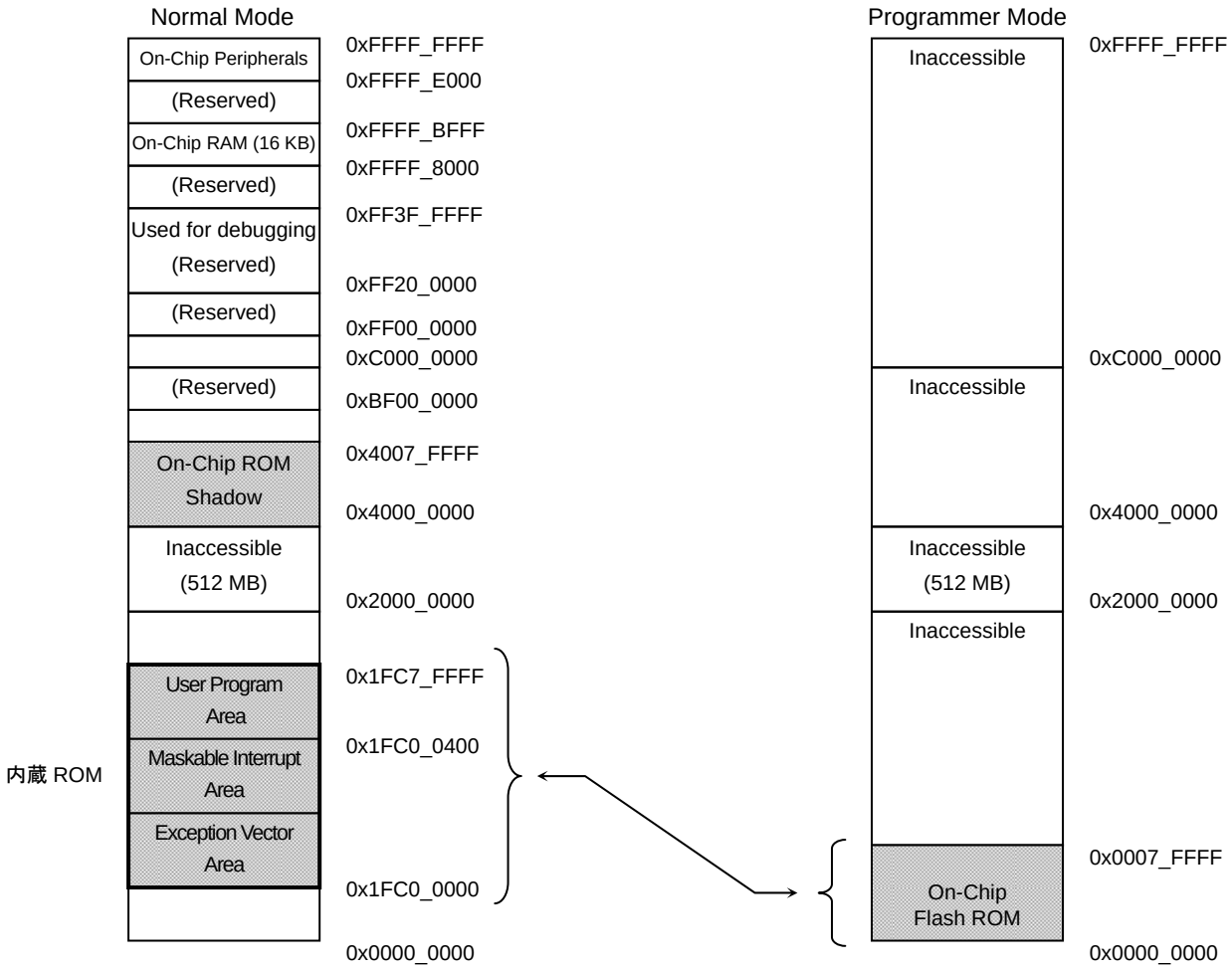


Figure 3.4.12 Memory Maps in Normal and Programmer Modes

3.4.2.3 Pin Functions and Settings

Table 3.4.7 EPROM Programmer Connections

Programmer	TMP1942	Function	Programmer	TMP1942	Function
GND	P30		DQ0	PB0	
A0	P31		DQ1	PB1	
A1	P32		DQ2	PB2	
A2	P33		DQ3	PB3	
A3	P34		DQ4	PB4	
A4	P35		DQ5	PB5	
A5	P36		DQ6	PB6	
A6	P37		DQ7	PB7	Data bus (input/output)
A7	PE0		DQ8	P10	
A8	PE1	Address bus (input)	DQ9	P11	
A9	PE2		DQ10	P12	
A10	PE3		DQ11	P13	
A11	PE4		DQ12	P14	
A12	PE5		DQ13	P15	
A13	PE6		DQ14	P16	
A14	PE7		DQ15	P17	
A15	PD0		$\overline{CE}$	P27	Chip Enable input
A16	PD1		$\overline{WE}$	P26	Write Enable input
A17	PD2		$\overline{OE}$	P25	Output Enable input
$\overline{RESET}$	PD5	Hardware reset input	FSE	P24	Flash Security Enable input
RDY/BSY	ALE	Ready/Busy output			
GND	FVSS	Ground	VCC (3 V)	FVCC (3 V)	Power supply (+3 V)

Table 3.4.8 Settings of the Other Pins

Pin Name	# of Pins	Type	Setting
$\overline{\text{RESET}}$	1	Input	Tie to logic 0 (0V). (Programmer mode setting)
BW0	1	Input	Tie to logic 0 (0 V). (Programmer mode setting)
BW1	1	Input	Tie to logic 1 (3 V). (Programmer mode setting)
P40	1	Input	Tie to logic 1 (3 V). (Programmer mode setting)
P41, P42, P43	3	Input	Tie to logic 0 (0 V). (Programmer mode setting)
$\overline{\text{NMI}}$	1	Input	Tie to logic 1 (3 V).
X1	1	Input	Connect a 20-MHz crystal for self-oscillation.
X2	1	Output	
P44	1	Input	Tie to logic 0 (0 V).
P00-P07	8	Input	Tie to logic 1 (3 V).
P20-P23	4	Input	Tie to logic 1 (3 V).
P50- P57	8	Input	Tie to logic 1 (3 V).
P60- P67	8	Input	Tie to logic 1 (3 V).
P90-P97	8	Input	Tie to logic 1 (3 V).
PA0-PA7	8	Input	Tie to logic 1 (3 V).
PC0-PC7	8	Input	Tie to logic 1 (3 V).
PD3, PD4	2	Input	Tie to logic 1 (3 V).
PF0- PF7	8	Input	Tie to logic 0 (0 V).
$\overline{\text{PLLOFF}}$	1	Input	Tie to logic 0 (0 V).
TEST	1	Input	Tie to logic 0 (0 V).
DVCC3, DVCC51, 52	6	Input	Tie to logic 1 (3 V).
DVSS	5	Input	0 V
CVCC	1	Input	Tie to logic 1 (3 V).
CVSS	1	Input	0 V
AVCC, DAVCC	2	Input	Tie to logic 1 (3 V).
AVSS, DAVSS	2	Input	0 V
VREFH	1	Input	Tie to logic 1 (3V).
VREFL	1	Input	0 V

3.4.2.4 Key Features

The TMP1942FD flash memory commands are in principle compatible with the standard JEDEC commands. After a command sequence is written, the flash memory does not require the system to provide further controls or timings. The flash memory initiates the embedded program or erase algorithm automatically. The entire flash memory or one or more flash blocks can be erased at a time.

Table 3.4.9 Flash Memory Features

Feature	Description
Auto Program	Programs and verifies the desired addresses word by word automatically.
Auto Chip Erase	Erases and verifies the entire memory array automatically.
Auto Block Erase	Erases and verifies all memory locations in the selected block automatically.
Auto Multi-Block Erase	Erases and verifies all memory locations in multiple selected blocks automatically.
Write operation status	Provides several status bits such as the Data Polling bit and Toggle bit, which can be used to determine whether a program or erase operation is complete or in progress.
Security feature	Prevents intrusive access to the flash memory while in Programmer mode. When the security feature is turned off, the entire memory array is erased and verified automatically, regardless of whether a given block is protected or not.
Block protection	Disables both program and erase operations in any block.

All accesses to the flash memory are performed word by word (16 bits), including the writing of commands. Unless otherwise noted, the subsections that follow indicate addresses as seen from the programmer.

The program/erase operations of on-board programming modes are very similar to those

of Programmer mode, with a few exceptions such as the data bus width. For details, refer to the descriptions of the program and erase operations in on-board programming modes earlier in this section.

3.4.2.5 Block Architecture

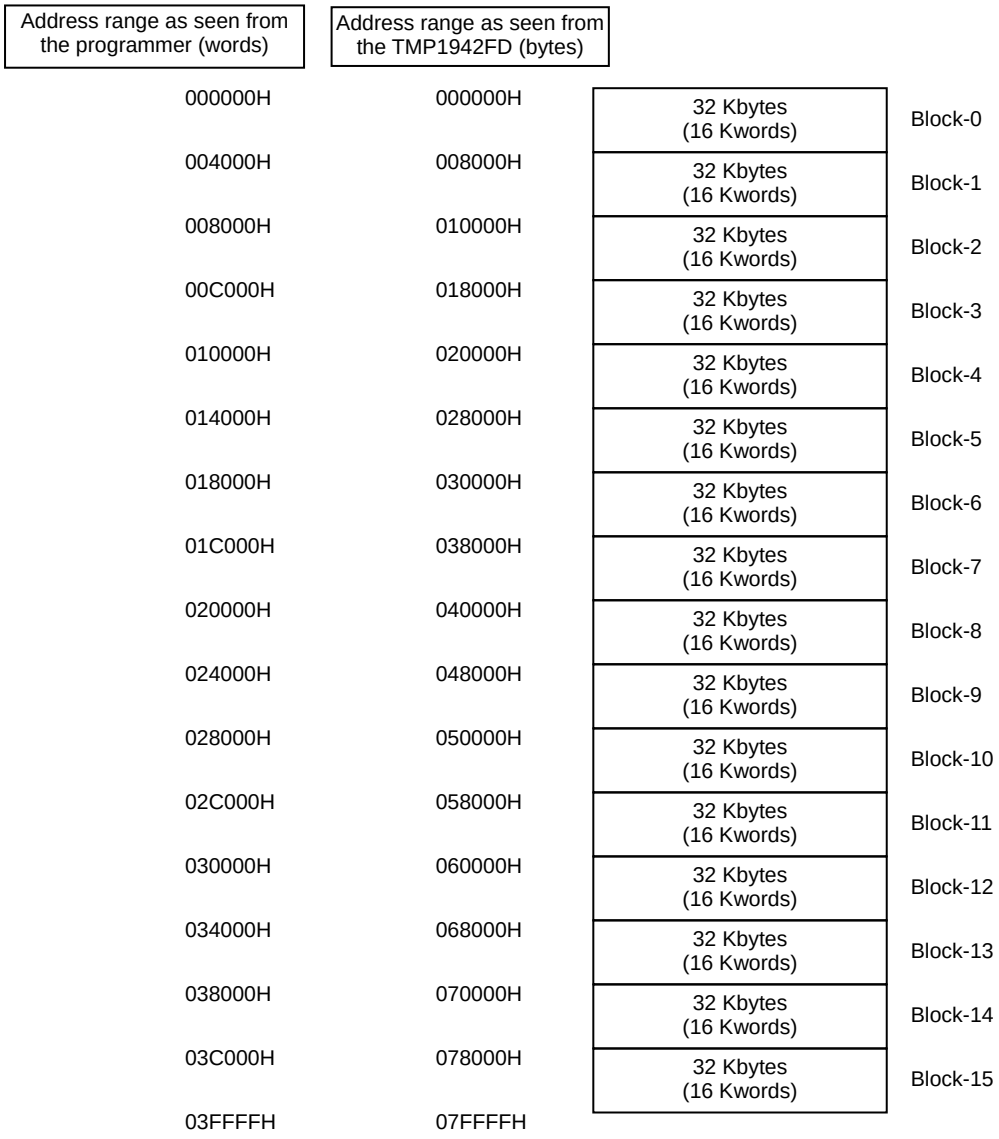


Figure 3.4.13 Flash Memory Block Architecture and Address Ranges in Writer Mode

3.4.2.6 Read Mode and Embedded Operation Mode

The TMP1942FD flash memory has the following two modes of operation:

- Read mode in which array data is read
- Embedded Operation mode in which the flash array is programmed or erased

The flash memory enters Embedded Operation mode when a valid command is executed in Read mode. In Embedded Operation mode, array data cannot be read. In Writer mode, all bus cycles such as the writing of commands and the reading of data are performed as a 16-bit quantity.

The flash memory has a security bit apart from the flash array. The reading of the flash array can be disabled in Writer mode by programming this bit. In Writer mode, the FSE pin is used for this purpose. In Normal operation mode, the FSE pin must be held at the V_{IL} level to access the flash array. During any operation, the FSE pin must remain stable.

3.4.2.7 Reading Array Data

The flash memory is automatically set to Read mode upon CPU reset after device power-up and after an embedded operation is successfully completed. If an embedded operation terminated abnormally or the flash memory is required to return to Read mode, the Read/Reset command (software reset) or hardware reset is used.

3.4.2.8 Writing Commands

The operations of the flash memory are selected by commands or command sequences written into the internal command register. This uses the same mechanism as for JEDEC-standard EEPROMs. Commands are made up of data sequences written at specific addresses via the command register. (See Table 3.4.12 and Table 3.4.13.)

The command sequence being written can be canceled by issuing the Read/Reset command between sequence cycles. The Read/Reset command clears the command register and resets the flash memory to Read mode. Invalid command sequences also cause the flash memory to clear the command register and return to Read mode.

3.4.2.9 Reset

- Read/Reset command (software reset)
  The flash memory does not return to Read mode if an embedded operation terminated abnormally. In this case, the Read/Reset command must be issued to put the flash memory back in Read mode. The Read/Reset command may also be written between sequence cycles of the command being written to clear the command register.

- Hardware reset
  The $\overline{\text{RESET}}$ pin provides a hardware method of terminating an embedded operation or clearing the internal command register being written. A reset is performed when the $\overline{\text{RESET}}$ pin is set to V_{IL} and kept low at least 500 ns. It takes 20 μs for a reset to complete and put the flash memory in Read mode. An embedded operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.
  After a reset, the flash memory is set to Read mode if $\overline{\text{RESET}}$ is at the V_{IH} level

and to Standby mode if $\overline{\text{RESET}}$ is at the V_{IL} level. While $\overline{\text{RESET}}$ is at the V_{IL} level, D0 to D15 are held at the high-impedance state. Any command sequence must be written after the flash memory is put back in Read mode.

3.4.2.10 Auto Program Command

In Writer mode, the programming of the flash array is performed on a word-by -word (16-bit) basis. In the fourth bus cycle of the Auto Program command sequence, the program address is latched on the falling edge of $\overline{\text{WE}}$, and data is latched on the rising edge of $\overline{\text{WE}}$. The latching of the program data initiates the embedded Auto Program algorithm. The Auto Program command executes a sequence of internally timed events to program the desired bits of the addressed memory location and verify that the desired bits are sufficiently programmed. The system can determine the status of the programming operation by using write status flags (see Table 3.4.15).

Any commands written during the programming operation are ignored. A hardware reset immediately terminates the programming operation. The programming operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables programming operations in any block. If an attempt is made to program a protected block, the Auto Program command does nothing; the flash memory returns to Read mode in approximately 3 μs after the rising edge of $\overline{\text{WE}}$ in the fourth bus cycle of the command sequence.

A bit must be programmed to change its state from a 1 to a 0. A bit cannot be programmed from a 0 back to a 1. Only an erase operation can change a 0 back to a 1. A programming failure condition is indicated if the system tries to program a 1 to a location that was previously programmed to a 0. Note that this is not a device failure condition since the flash memory was used incorrectly.

When the embedded Auto Program algorithm is complete, the flash memory returns to Read mode.

If any failure occurs during the programming operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of a programming failure, it is recommended to replace the chip or discontinue the use of the failing flash block.

3.4.2.11 Auto Chip Erase Command

The embedded Auto Chip Erase algorithm is initiated on the rising edge of $\overline{\text{WE}}$ in the sixth bus cycle of the command sequence. The embedded Auto Chip Erase algorithm automatically preprograms the entire memory for an all-0 data pattern prior to the erase; then it automatically erases and verifies the entire memory for an all-1 data pattern. The system can determine the status of the chip erase operation by using write status flags (see Table 3.4.15).

Any commands written during the chip erase operation are ignored. A hardware reset immediately terminates the chip erase operation. The chip erase operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables erase operations in any block. If all the blocks are protected, the Auto Chip Erase command does nothing; the flash memory returns to Read mode in approximately 100 μs after the rising edge of $\overline{\text{WE}}$ in the sixth bus write cycle of

the command sequence.

When the embedded Auto Chip Erase algorithm is complete, the flash memory returns to Read mode.

If any failure occurs during the erase operation, the flash memory remains locked in Embedded Operation mode. The system can determine this state by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block. The failing block can be identified by means of the Block Erase command.

3.4.2.12 Auto Block Erase and Auto Multi-Block Erase Commands

The address of the block to be erased is latched on the falling edge of $\overline{WE}$ in the sixth bus write cycle of the command sequence. A time-out begins at the rising edge of that $\overline{WE}$ pulse. After a time-out, the erase operation will commence. The embedded Auto Block Erase algorithm automatically preprograms the selected block for an all-0 data pattern, and then erases and verifies that block for an all-1 data pattern.

During the time-out period, additional block addresses and Auto Block Erase commands may be written (see Table 3.4.14).

Any command other than Auto Block Erase during the time-out period resets the flash memory to Read mode. The block erase time-out period is 50 μs. The time-out window is reset on each rising edge of $\overline{WE}$. The system may determine the status of the erase operation by using write status flags (see Table 3.4.15).

Any commands written during the block erase operation are ignored. A hardware reset immediately terminates the block erase operation. The block erase operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables erase operations in any block. If all the selected blocks are protected, the Auto Block Erase algorithm does nothing; the flash memory returns to Read mode in approximately 100 μs after the rising edge of $\overline{WE}$ in the final bus cycle of the command sequence.

If any failure occurs during the erase operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block. If any failure occurred during the multi-block erase operation, the failing block can be identified by running Auto Block Erase on each of the blocks selected for multi-block erasure.

3.4.2.13 Block Protect Command

The block protection feature disables both program and erase operations in any block. The effects of the program and erase commands on the protected blocks are summarized below.

Table 3.4.10 Effects of the Program and Erase Commands on the Protected Blocks

Command	Operation
Program command on a protected block	No programming operation is performed, and the flash memory automatically returns to Read mode.
Block Erase command on a protected block	No erase operation is performed, and the flash memory automatically returns to Read mode.
Chip Erase command when all the blocks are protected	No erase operation is performed, and the flash memory automatically returns to Read mode.
Chip Erase command when any blocks are protected	Only the unprotected blocks are erased. Upon completion, the flash memory automatically returns to Read mode.
Multi-Block Erase command when any blocks are protected	Only the unprotected blocks are erased. Upon completion, the flash memory automatically returns to Read mode.

After the command sequence is complete, writing to the protect control logic is performed by pulsing $\overline{WE}$ for t_{PPLH} while $\overline{CE}$ is set to V_{IL} and the block address is placed on P70 (A16) to P54 (A12).

Any commands written during the Block Protect algorithm are ignored. A hardware reset immediately terminates the block protect operation. The Block Protect command that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence.

Note that the block protect operation is not verified automatically. The Verify Block Protect command must be written to verify the protect status after executing Bock Protect. If the selected block is not in the protected state, the Block Protect command sequence must be re-initiated (see Figure 3.4.18).

3.4.2.14 Block Unprotect Command

The Block Unprotect command unprotects all blocks simultaneously. All blocks must be protected before executing the Block Unprotect command. After the Block Unprotect command sequence is complete, block unprotection is performed by pulsing $\overline{WE}$ for t_{PULH} with $\overline{CE}$ set to V_{IL} .

Any commands written during the Block Unprotect algorithm are ignored. A hardware reset immediately terminates the block unprotect operation. The Block Unprotect command that was interrupted should be re-initiated from protecting all blocks. The Verify Block Protect command must be written to verify the protect status after executing Block Unprotect.

3.4.2.15 Verify Block Protect Command

The Verify Block Protect command is used to verify the protect status of a block. The address of the block to be verified is given in the fourth bus read cycle of the command sequence. Any address within the block range will suffice, provided A0 = 1 and A5 = 0. If the selected block is protected, a value of 0x0001 is returned. If the selected block is not protected, a value of 0x0000 is returned. Following the fourth bus cycle, an additional block address may be provided.

The Verify Block Protect command does not return the flash memory to Read mode. Either the Read/Reset command or a hardware reset is required to reset the flash memory to Read mode or to write the next command.

3.4.2.16 Write Operation Status

As shown in Table 3.4.15, the flash memory provides several flag bits to determine the status of an embedded operation: DQ7, DQ5, DQ3 and RDY/BSY. These status bits can be read during an embedded operation using the same timing as for Read mode by setting $\overline{CE}$ and $\overline{OE}$ to V_{IL}. The RDY/BSY status is valid after the rising edge of the final $\overline{WE}$ pulse in the command sequence, regardless of $\overline{CE}$ and $\overline{OE}$. The flash memory automatically returns to Read mode when an embedded operation completes.

- DQ7 (Data Polling)
  The Data Polling bit, DQ7, indicates to the host system the status of an embedded operation. Data Polling is valid after the rising edge of the final $\overline{WE}$ pulse in the command sequence.
  When the embedded Program algorithm is in progress, an attempt to read the flash memory will produce the complement of the data last written to DQ7. Upon completion of the embedded Program algorithm, an attempt to read the flash memory will produce the true data last written to DQ7. Therefore, the system can use DQ7 to determine whether the embedded Program algorithm is in progress or completed.
  When the embedded Erase algorithm is in progress, an attempt to read the flash memory will produce a 0 at the DQ7 output. Upon completion of the embedded Erase algorithm, the flash memory will produce a 1 at the DQ7 output.
  If there is a failure during an embedded operation, DQ7 continues to output the same value. Thus, DQ7 must always be polled in conjunction with the Exceeded Timing Limits (DQ5) flag. Figure 3.4.16 shows the DQ7 polling algorithm.
  The flash memory disables address latching when an embedded operation is complete. Data polling must be performed with a valid programmed address or an address within any of the non-protected blocks selected for erasure. DQ7 may change asynchronously while $\overline{OE}$ is asserted low.

- DQ5 (Exceeded Timing Limits)
  DQ5 produces a 0 while the program or erase operation is in progress normally. DQ5 produces a 1 to indicate that the program or erase time has exceeded the specified internal limit. This is a failure condition that indicates the program or erase cycle was not successfully completed.
  The DQ5 failure condition also appears if the system tries to program a 1 to a location that was previously programmed to a 0. In this case, the embedded Program algorithm halts the operation. Once the operation exceeded the timing limits, DQ5 will indicate a 1. Note that this is not a device failure condition since the flash memory was used incorrectly.
  Under both these conditions, the flash memory remains locked in Embedded Operation mode. The Read/Reset command must be issued to return the flash memory to Read mode.

- DQ3 (Block Erase Timer)
  The block erase time-out window of 50 μs begins from the rising edge of $\overline{WE}$ in the sixth bus cycle of the command sequence. The erase operation will begin after the time-out has expired. When the time-out is complete and the erase operation has begun, DQ3 switches from 0 to 1. If DQ3 is 0, the flash memory will accept additional Auto Block Erase commands. Each time an Auto Block Erase command is written, the time-out window is reset. To ensure that the command has been accepted, the system should check DQ3 prior to and following each Auto Block Erase command. If DQ3 is 1 on the second status check, the command might not

have been accepted.

- RDY/BSY (Ready/Busy)
  In Programmer mode, the ALE pin functions as the RDY/BSY pin. The programming equipment can monitor the state of this pin to determine whether an embedded algorithm is in progress or complete. RDY/BSY produces a 0 when the flash memory is actively erasing or programming. RDY/BSY produces a 1 when an embedded operation has completed and the flash memory is ready to accept the next command. If any failure occurs during the program or erase operation, this flag remains at the 0 logic state. RDY/BSY is not an open-drain output, but a normal CMOS output pin.

3.4.2.17 Flash Security

The TMP1942FD flash memory has a security bit apart from the flash array. Programming this security bit disables access to the flash array. This prevents intrusive access to the flash memory by third parties while in Programmer mode.

- Securing the flash memory (Disabling read accesses)
  Securing the flash memory disables a general-purpose programmer to read its contents. To turn on the security feature, once programming is complete, write the Auto Security On command, with the FSE pin set to V_{IL}. In the fourth bus cycle of the command sequence, program 0x0098 at address 0x0000. After the rising edge of $\overline{WE}$ in the fourth bus cycle, the embedded Security On algorithm automatically programs and verifies the security bit.
  Any commands written during the embedded operation are ignored. A hardware reset immediately terminates the embedded operation. The FSE bit must be held stable throughout the embedded operation.
  When the embedded algorithm completes, the flash memory automatically returns to Read mode.
  If any failure occurs during the embedded operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. Note that this is a security bit failure. If the flash memory needs to be secured, the chip should be replaced. When the security is on, any reads by programming equipment always return a word-length value of 0x0098.

- Unsecuring the flash memory (Enabling read accesses)
  To turn off the security feature, with the Auto Security Off command, with the FSE pin set to V_{IH}. After the rising edge of $\overline{WE}$ in the sixth bus cycle of the command sequence, the embedded Security Off algorithm automatically erases and verifies the entire flash array, and then erases and verifies the security bit.
  Any commands written during the embedded operation are ignored. A hardware reset immediately terminates the embedded operation. In this case, if any erase operation is in progress, data may be corrupted. The FSE pin must be held stable throughout the embedded operation.
  When the embedded algorithm completes, the flash memory automatically returns to Read mode.
  If any failure occurs during the embedded operation, the flash memory remains locked in Embedded Operation mode. The system can determine the status of the embedded operation by using write status flags. If a failure occurs in the memory array, the security bit is not erased. In this case, the security bit is left on. The chip should be replaced if a memory array or security bit failure occurs.
  The Auto Security Off command erases the flash array prior to turning off the security feature. Even if a given block is protected, it is unconditionally erased, but

the protect status of that block remains unchanged. The Auto Security Off and Auto Chip Erase command sequences are the same. The only difference is that the Auto Security Off command requires the FSE pin to be set to the V_{IH} level before the command is written. The Auto Block Erase command does not turn off the security feature even when the FSE pin is set to V_{IH}. If the Auto Block Erase command is written with the FSE input pin set to V_{IH}, no block will be erased and the operation is immediately terminated.

Table 3.4.11 Basic Operation Modes (with addresses as seen from the programmer)

Mode	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A5	A0	$\overline{RESET}$	DQ0-DQ15
Read	0	0	1	A5	A0	1	Dout
Standby	1	X	X	X	X	1	Hi-Z
Output Disable	X	1	1	X	X	X	Hi-Z
Write	0	1	0	A5	A0	1	Din
Hardware Reset/Standby	X	X	X	X	X	0	Hi-Z

Table 3.4.12 Programmer Mode Command Definitions (with addresses as seen from the programmer)

Command Sequence	Cycles Required	Bus Cycles											
		1st Cycle (Write)		2nd Cycle (Write)		3rd Cycle (Write)		4th Cycle (Read/Write)		5th Cycle (Write)		6th Cycle (Write)	
		Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data
Read/Reset	1	0xXXXX	F0H										
Read/Reset	3	0x5555	AAH	0xAAAA	55H	0x5555	F0H	RA	RD				
Auto Program	4	0x5555	AAH	0xAAAA	55H	0x5555	A0H	PA	PD				
Auto Chip Erase	6	0x5555	AAH	0xAAAA	55H	0x5555	80H	0x5555	AAH	0xAAAA	55H	0x5555	10H
Auto Block Erase	6	0x5555	AAH	0xAAAA	55H	0x5555	80H	0x5555	AAH	0xAAAA	55H	BA	30H
Block Protect	6	0x5555	AAH	0xAAAA	55H	0x5555	9AH	0x5555	AAH	0xAAAA	55H	0x5555	9AH
Verify Block Protect	4	0x5555	AAH	0xAAAA	55H	0x5555	90H	BPA	BD				
Auto Security On (Note 1)	4	0x5555	AAH	0xAAAA	55H	0x5555	A0H	0x0000	98H				

Note: Write the command sequence with the FSE input pin set to V_{IH}. This enables access to the security bit. Write the other command sequences with the FSE input pin set to V_{IL}.

- F0H, AAH, 55H, A0H, 80H, 10H, 30H:
  Command data. Write command as a word quantity with the upper byte 00H.
- RA: Read Address
  RD: Read Data
- PA: Program Address
  PD: Program Data
  Write data as a word quantity.
- BA: Block Address (BA0-BA6)
  Refer to Table 3.4.14.
- BPA: Verify Block Protect Address
- BD: Block Protect Data
  The address of the block to be verified can be any of the addresses within the block with A5 = 0 and A0 = 1. If a block is protected, a value of 0x0001 will be returned. If a block is not protected, a value of 0x0000 will be returned.

Table 3.4.13 shows the relationships between the addresses as seen from the programmer and the TMP1942FD.

Table 3.4.13 Relationships between Addresses

	Command Address																			
Programmer Address	—	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	—
TMP1942 Address	P73 ~ P77	P72 / A18	P71 / A17	P70 / A16	P57 / A15	P56 / A14	P55 / A13	P54 / A12	P53 / A11	P52 / A10	P51 / A9	P50 / A8	P37 / A7	P36 / A6	P35 / A5	P34 / A4	P33 / A3	P32 / A2	P31 / A1	P30 / A0
0xXXXX	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
0x0000		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xAAAA		1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0
0x5555		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Table 3.4.14 Block Erase Addresses in Writer Mode (as seen from the programmer)

Block	Address Range	Size
BA0	0x0000-0x3FFF	16 Kwords
BA1	0x4000-0x7FFF	16 Kwords
BA2	0x8000-0xBFFF	16 Kwords
BA3	0xC000-0xFFFF	16 Kwords
BA4	0x1000-0x13FFF	16 Kwords
BA5	0x14000-0x17FFF	16 Kwords
BA6	0x18000-0x1BFFF	
BA7		
BA8		
BA9		
BA10		
BA11		
BA12		
BA13		
BA14	0x38000-0x3BFFF	16 Kwords
BA15	0x3C000-0x3FFFF	16 Kwords

The address of the block to be erased can by any of the addresses within that block. For example, to select BA0, provide any address in the range between 0x0000 and 0x3FFF.

Table 3.4.15 Write Status Flags

Status		D7	D5	D3
Embedded operation in progress	Auto Program	$\overline{D7}$	0	0
	Auto Erase (during the time-out window)	0	0	0
	Auto Erase	0	0	1
Time-out in embedded operation	Auto Program	$\overline{D7}$	1	1
	Auto Erase	0	1	1

Note: D4 and D2-D0 are don't cares.

3.4.2.18 Embedded Algorithms

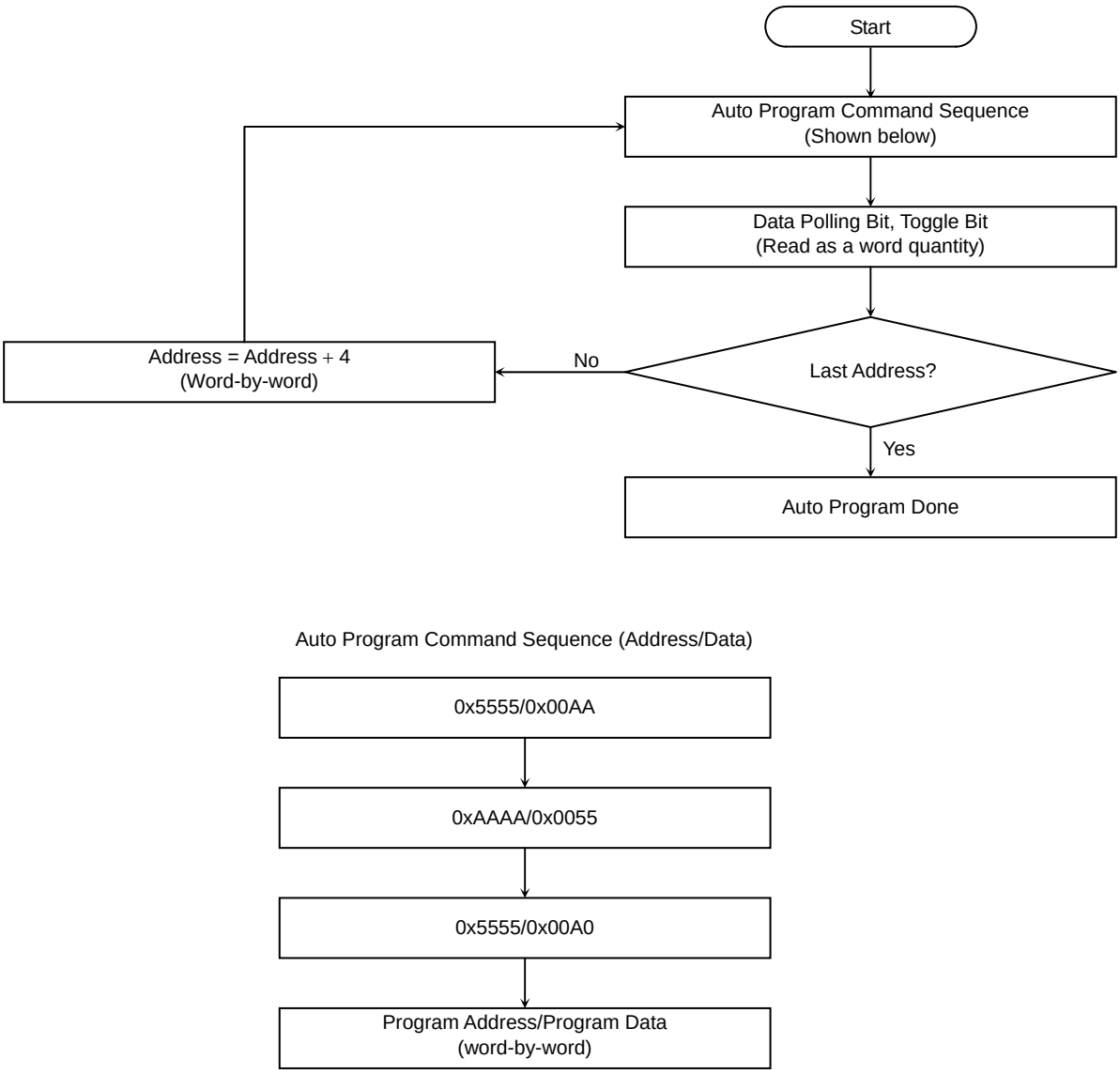


Figure 3.4.14 Auto Program Operation

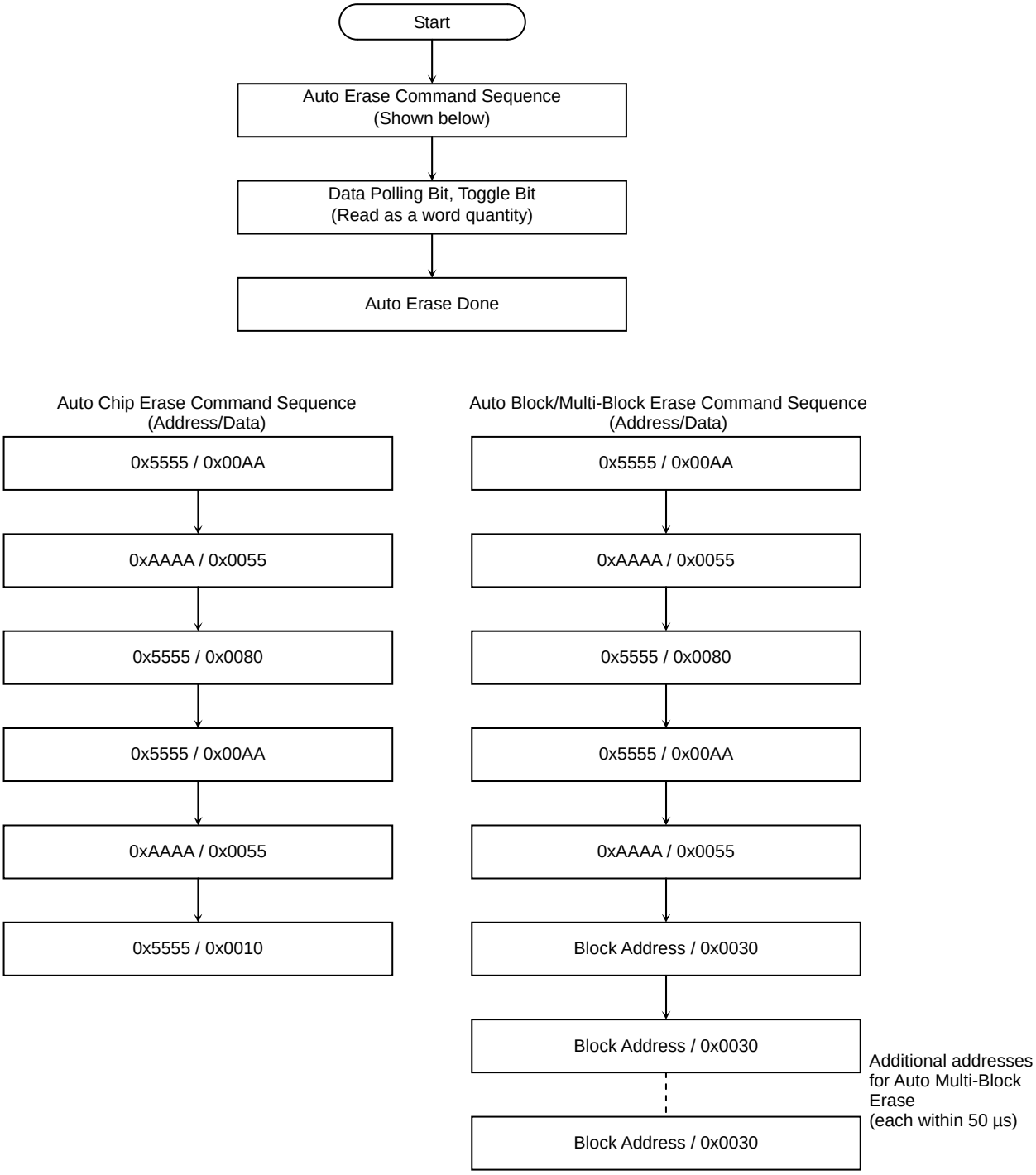


Figure 3.4.15 Auto Erase Operations

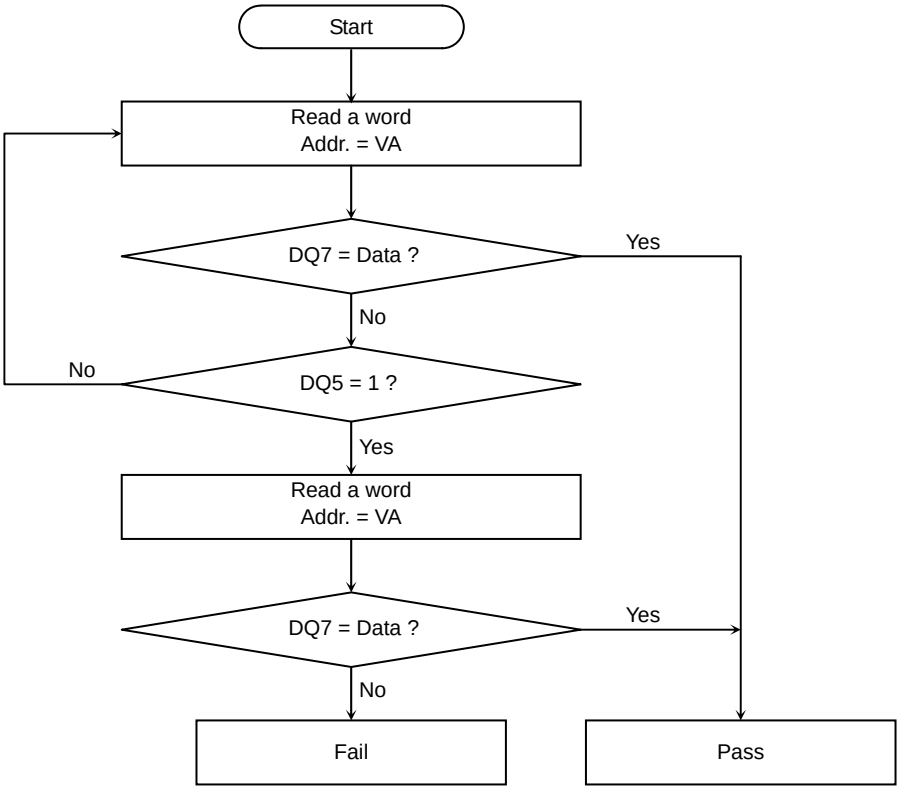
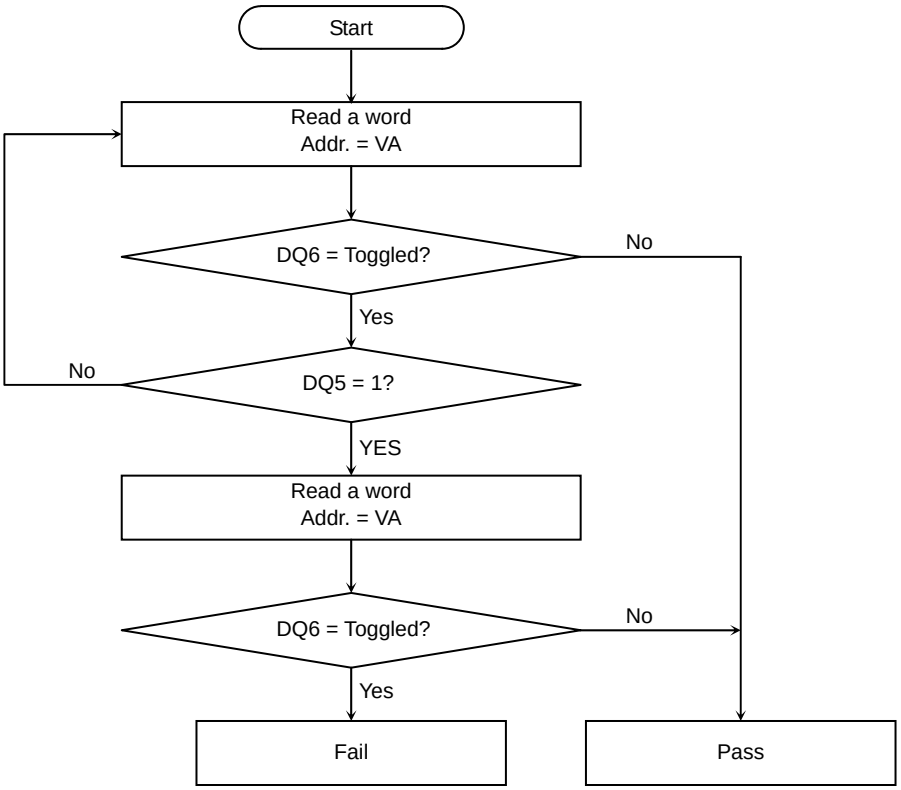


Figure 3.4.16 Data Polling (DQ7) Algorithm



VA: Address being programmed in Auto Program
Any address in the flash memory in Auto Chip Erase
Any address in the selected block in Auto Block Erase

Figure 3.4.17 Toggle Bit (DQ6) Algorithm

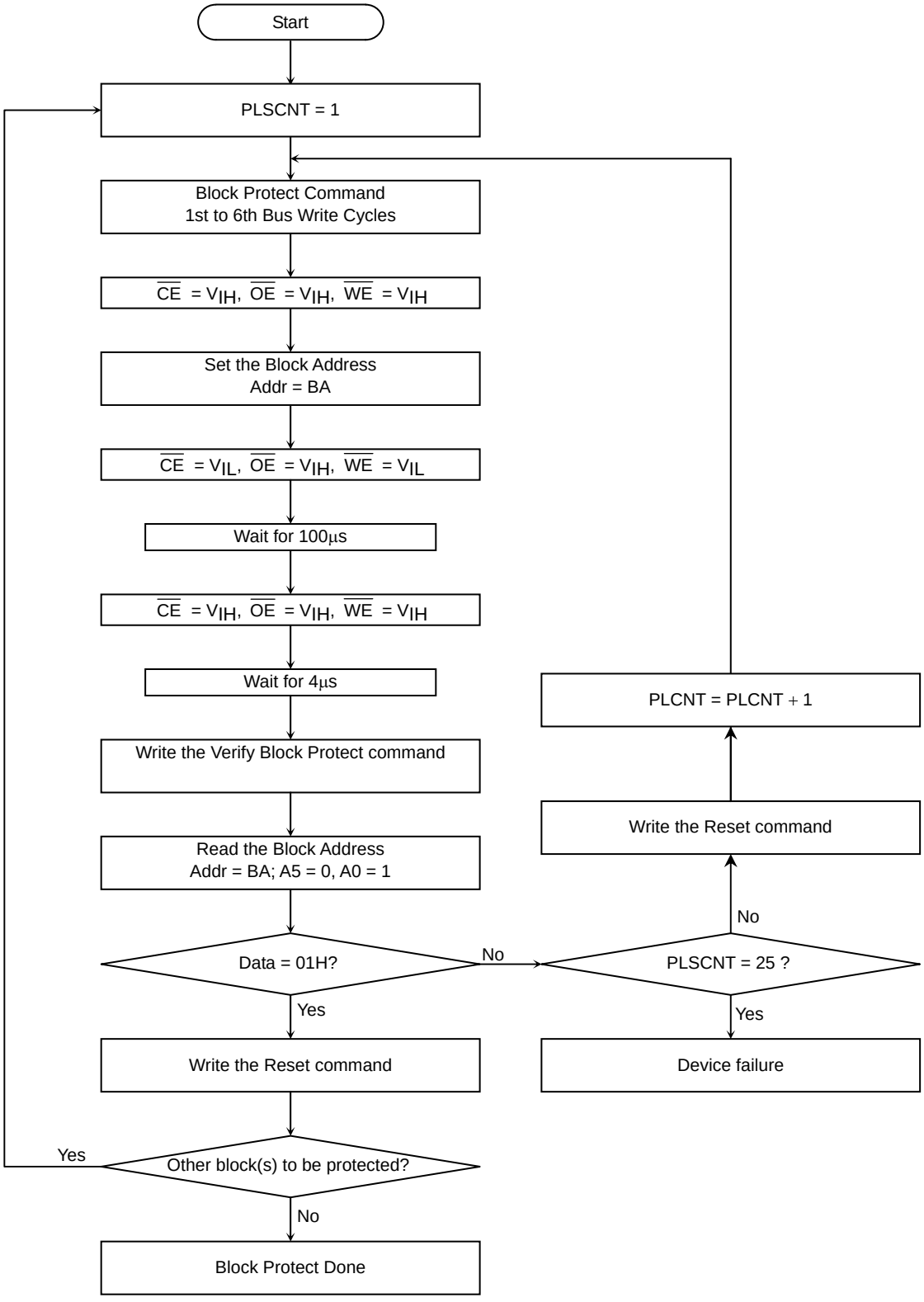


Figure 3.4.18 Block Protect Operation

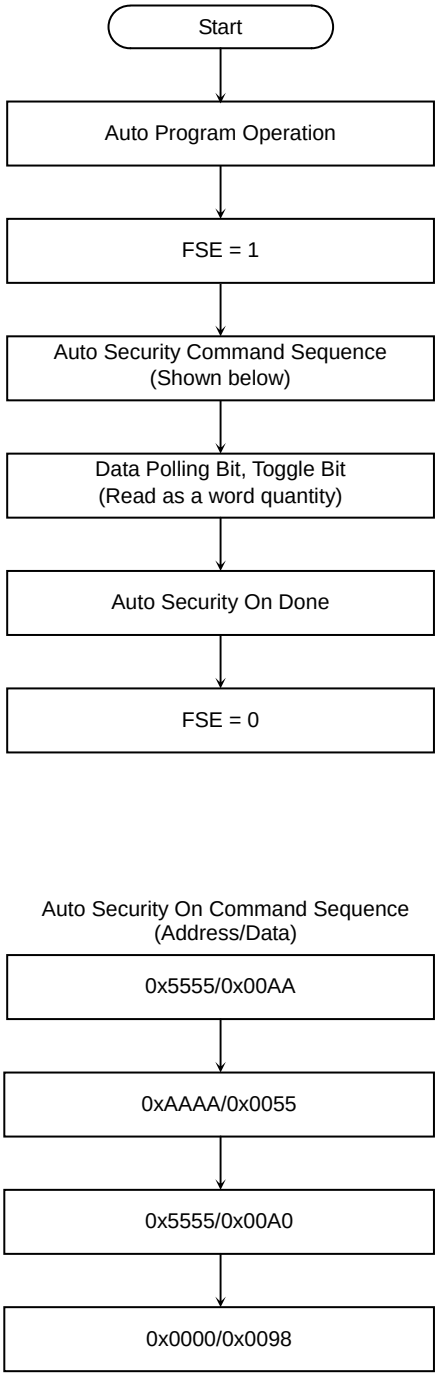


Figure 3.4.19 Auto Security On Operation

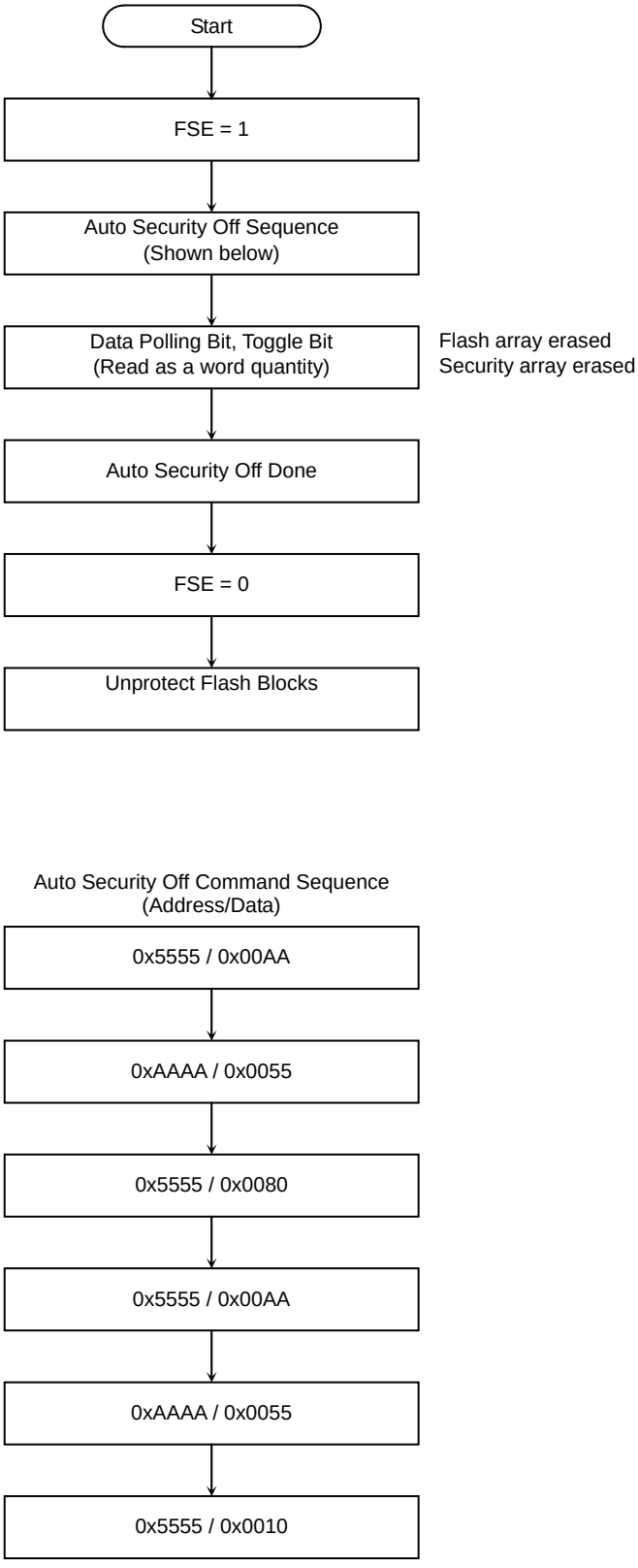


Figure 3.4.20 Auto Security Off Operation

4. Electrical Characteristics

The letter x in equations presented in this chapter represents the cycle period of the fsys clock selected through the programming of the SYSCR1.SYSCK bit. The fsys clock may be derived from either the high-speed or low-speed crystal oscillator. The programming of the clock gear function also affects the fsys frequency. All relevant values in this chapter are calculated with the high-speed (fc) system clock (SYSCR1.SYSCK = 0) and a clock gear factor of 1/fc (SYSCR1.GEAR[1:0] = 00).

4.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		V_{CC3}	− 0.5~4.0	V
		V_{CC5}	− 0.5~6.0	V
Input voltage		V_{IN3}	− 0.5~V_{CC3} + 0.5	V
		V_{IN5}(*)	− 0.5~ V_{CC5}+ 0.5	V
Analog input voltage		V_{AIN}	− 0.5~ AVCC+ 0.5	V
Analog reference input voltage		VAREFH	− 0.5~ AVCC+ 0.5	V
		DAREFH	− 0.5~ DAVCC+ 0.5	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	80	
High-level output current	Per pin	I_{OH}	−5	
	Total	ΣI_{OH}	−80	
Power dissipation (Ta = 85°C)		PD	600	mW
Soldering temperature (10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−65~150	°C
Operating temperature	Except during flash write/erase	T_{OPR}	−40~85	°C
	During flash write/erase		0~70	
Write/erase cycles		N_{EW}	100	cycle

V_{CC3} = DV_{CC3} = AV_{CC} = DAV_{CC} = CV_{CC}, V_{CC5} = DV_{CC51} = DV_{CC52}, V_{SS} =DV_{SS} = AV_{SS} = DAV_{SS} = CV_{SS}

(*) Applies only to Port C and Port F.

Note: Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute Maximum Ratings value is exceeded with respect to current, voltage, power dissipation, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.

4.2 DC Electrical Characteristics (1/5)

Ta = −40 to 85°C

Parameter		Symbol	Conditions		Min.	Typ. (Note 1)	Max.	Unit
Supply voltage DAVCC=AVCC =CVCC=DVCC3 DAVSS=AVSS =CVSS= 0V		DVCC3	PLLON (INTLV="H")	fosc = 5 to 8 MHz fsys = 2.5 to 32 MHz fs = 30 to 34 kHz	3.0		3.6	V
				fosc = 5 to 7 MHz fsys = 5 to 28 MHz fs = 30 to 34 kHz	2.7			
			PLLOFF (Crystal)	(INTLV = "H") fosc = 10 to 20 MHz fsys = 1 to 20 MHz fs = 30 to 34 kHz (INTLV = "L") fosc = 10 to 16 MHz fsys = 1 to 16 MHz fs = 30 to 34 kHz	2.7			
			PLLOFF (External clock)	fosc = 20 to 32 MHz fsys = 1.25 to 16 MHz fs = 30 to 34 kHz <DFOSC> = "0"	3.0			
				fosc = 10 to 16 MHz fsys = 1 to 16 MHz fs = 30 to 34 kHz	2.7			
		DVCC5* (Note 2)	fsys = 1 to 32 MHz fs = 30 to 34 kHz		4.5		5.25	
Low-level input voltage	P00 - P17 (AD0 - AD15)	V_{IL}			− 0.3		0.6	V
	P20 - PB7, PD0 - PE7	V_{IL1}					0.3 DVCC3	
	$\overline{\text{PLLOFF}}$, BW0, BW1, RSTPUP, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$	V_{IL2}					0.2 DVCC3	
	PC0 - PC7,PF0 - PF6	V_{IL3}					0.3 DVCC5	
	X1	V_{IL4}					0.2 DVCC3	
High-level input voltage	P00 - P17 (AD0 - AD15)	V_{IH}	DVCC3 ≥ 2.7 V DVCC5 ≥ 4.5 V		2.0		DVCC3 + 0.3	
	P20 - PB7,PD0 - PE7	V_{IH1}			0.7DVCC3			
	$\overline{\text{PLLOFF}}$, BW0, BW1, RSTPUP, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$	V_{IH2}			0.8DVCC3			
	PC0 - PC7,PF0 - PF6	V_{IH3}			0.7DVCC5		DVCC5 + 0.3	
	X1	V_{IH4}			0.8DVCC3		DVCC3 + 0.3	
Low-level output voltage		V_{OL}	I_{OL} = 1.6 mA	DVCC3 ≥ 2.7 V DVCC5 ≥ 4.5 V			0.45	V
High-level output voltage		V_{OH1}	I_{OH} = − 400 μA	DVCC3 ≥ 2.7 V	2.4			
		V_{OH2} (Note 3)		DVCC5 ≥ 2.7 V	2.4			
				DVCC5 ≥ 4.5 V	4.2			

Note 1: Ta = 25 °C, DVCC3 = 3.3 V, DVCC5 = 5.0 V, unless otherwise noted.
Note 2: DVCC5*: DVCC51, DVCC52
DVCC5* can also be used as 2.7 V ≤ DVCC5* ≤ 3.6 V.
Note 3: Applies only to Port C and Port F.

4.3 DC Electrical Characteristics (2/5)

Ta = −40 to 85°C

Parameter	Symbol	Conditions	Min.	Typ. (Note)	Max.	Unit
Input leakage current	I_{LI}	$0.0 \leq V_{IN} \leq DVCCn$ (n = 3, 5)		0.02	± 5	μA
Output leakage current	I_{LO}	$0.2 \leq V_{IN} \leq DVCCn - 0.2$ (n = 3, 5)		0.05	± 10	
Power-down voltage (STOP mode, RAM backup)	V_{STOP1}	V_{IL2} = 0.2DVCC3 V_{IH2} = 0.8DVCC3	2.2		3.6	V
	V_{STOP2}	V_{IL2} = 0.2DVCC5 V_{IH2} = 0.8DVCC5	V_{STOP1}		5.25	
Pull-up resistor at reset	RRST	V_{CC} = 3.3 V ± 0.3 V	100		550	kΩ
Programmable pull-up resistor P32-P37,P40-P43 KEY0-KEYD	PKH1	DVCC3 = 3.3 V ± 0.3 V	30	45	100	kΩ
	PKH2	DVCC5 = 4.5 V to 5.25 V	30	55	100	
Pin capacitance (Except power supply pins)	C_{IO}	fc = 1 MHz			10	pF

Note: Ta = 25 °C, DVCC3 = 3.3 V, DVCC5 = 5.0 V, unless otherwise noted.

4.4 DC Electrical Characteristics (3/5)

DVCC3 = 3.3 V±0.3 V, DVCC51 = DVCC52 = 3.3 V±0.3 V, Ta = −40 to 85°C

Parameter	Symbol	Conditions	Min.	Typ. (Note 1)	Max.	Unit
NORMAL (Note 2): Gear = 1/1	I_{CC}	f_{sys} = 32 MHz (f_{OSC} = 8 MHz, PLLON) INTLV = "H"		110	135	mA
IDLE (Doze)				35	50	
IDLE (Halt)				30	44	
NORMAL (Note 2): Gear = 1/1		f_{sys} = 16 MHz (f_{OSC} = 16 MHz, PLLOFF) INTLV = "L"		70	85	mA
IDLE (Doze)				17.5	32	
IDLE (Halt)				15	28	
SLOW		fs = 32.768 kHz		210	280	μA
SLEEP		fs = 32.768 kHz		10	60	μA
STOP		DVCC3 = 2.7 to 3.6 V DVCC5 = 2.7 to 3.6 V		1	50	μA

Note 1: Ta = 25 °C, DVCC3 = 3.3 V, DVCC5 = 3.3 V, unless otherwise noted.

Note 2: I_{CC} NORMAL measurement conditions

CPU: Dhrystone (Ver. 2.1) execution (including external memory access)
8-bit timer: 500 kHz/50% output × 3 channels, 50 kHz/50% output x 3 channels
16-bit timer: 500 kHz/50% output x 3 channels, 50 kHz/50% output x 3 channels,
2-ms interval timer x 6 channels, two-phase pulse input counter x 2 channel
SIO: UART (11.5 kbps) transmission x 1 channel, clock synchronization (50 kHz) × 4 channels
ADC: Channel-fixed continuous conversion mode
DAC: Output (0x200) × 3 channels

Note 3: I_{CC} SLOW and I_{CC} SLEEP measurement conditions

CPU: Same as for NORMAL mode
RTC timer, two-phase pulse input counter, dynamic pull-up logic operating (16-ms cycle, 250-μs sampling)

Note 4: I_{CC} includes the supply current flowing through the DVCC3, DVCC5, CVCC, AVCC and DAVCC pins.

Note 5: I_{CC} NORMAL includes the reference current for the A/D and D/A converters.

Note 6: I_{CC} IDLE indicates the current values when on-chip peripherals are not operating.

4.5 DC Electrical Characteristics (4/5)

DVCC3 = 3.3 V±0.3 V, DVCC51 = DVCC52 = 5.0 V±0.25 V, Ta = −40 to 85°C

Parameter	Symbol	Conditions	Min.	Typ. (Note 1)	Max.	Unit
NORMAL: Gear =1/1	I_{CC}	f_{sys} = 32 MHz (f_{OSC} = 8 MHz, PLLON) INTLV = "H"		110	135	mA
		f_{sys} = 16 MHz (f_{OSC} = 16 MHz,PLLOFF) INTLV = "L"		70	85	mA
SLOW		fs = 32.768 kHz		210	280	μA
SLEEP		fs = 32.768 kHz		10	60	
STOP		DVCC3 = 2.7 to 3.6 V DVCC5 = 4.75 to 5.25 V		1	50	μA

Note 1: Ta = 25 °C, DVCC3 = 3.3 V, DVCC5 = 5.0 V, unless otherwise noted.

Note 2: I_{CC} measurement conditions

See Note 2 and Note 3 in 4.4, *DC Electrical Characteristics*.

Note 3: I_{CC} for DVCC3 includes the current consumed by CVCC, AVCC and DAVCC.

4.6 DC Electrical Characteristics (5/5)

Table 4.6.1 DC Electrical Characteristics in Modes Other Than Programmer Mode

Ta = −40 to 85°C (0 to70°C during program and erase of the flash memory), DVCC = 2.7 to 3.6 V

Symbol	Parameter	Condition	Min	Max	Unit
$I_{DDO}1$	Active write current	f_{SYS} = 32 MHz	-	185	mA

Table 4.6.2 DC Electrical Characteristics In Programmer Mode

Ta = 25 ± 5°C, DVCC = 2.7 to 3.6 V

Symbol	Parameter	Condition	Min	Max	Unit
V_{IH}	High-level input voltage	-	0.7 × D_{VCC}	D_{VCC} + 0.5	V
V_{IL}	Low-level input voltage	-	−0.3	0.8	V
I_{LI}	Input leakage current	0 V ≤ V_{IN} ≤ D_{VCC}	-	±1	μA
I_{LO}	Output leakage current	0 V ≤ V_{OUT} ≤ D_{VCC}	-	±1	μA
V_{OH}	High-level output voltage	I_{OH} = −0.1 mA	D_{VCC} − 0.4	-	V
		I_{OH} = −2.5 mA	0.85 × D_{VCC}	-	
V_{OL}	Low-level output voltage	I_{OL} = 4.0 mA	-	0.4	V
$I_{DDO}2$	Active write current	t_{CYC} = t_{RC} (Min)	-	50	mA

4.6.1 Precautions for Programming and Erasing the Flash Memory

- In on-board programming modes (Single Boot mode and User Boot mode), the flash program and erase operations must be given the highest priority. All interrupts including NMI must be disabled.
- An auto erase operation is required before performing an auto program operation on addresses that have already been programmed.
- To use Programmer mode to re-program the flash memory that has been programmed/erased in an on-board programming mode, it is recommended to perform an auto erase operation before executing an auto program operation.

Table 4.6.3 AC Electrical Characteristics in Programmer Mode

Symbol	Parameter	Min	Max	Unit
t_{RC}	Read cycle time	120	-	ns
t_{ACC}	Address access time	-	120	ns
t_{CE}	$\overline{CE}$ access time	-	120	ns
t_{OE}	$\overline{OE}$ access time	-	50	ns
t_{CEE}	$\overline{CE}$ to output low-Z	0	-	ns
t_{OEE}	$\overline{OE}$ to output low-Z	0	-	ns
t_{OEH}	$\overline{OE}$ hold time (read)	0	-	ns
t_{OH}	Output data hold time	0	-	ns
t_{DF1}	$\overline{CE}$ to output high-Z	-	30	ns
t_{DF2}	$\overline{OE}$ to output high-Z	-	30	ns
t_{CMD}	Command cycle time	120	-	ns
t_{AS}	Address setup time	0	-	ns
t_{AH}	Address hold time	50	-	ns
t_{DS}	Data setup time	60	-	ns
t_{DH}	Data hold time	0	-	ns
t_{WELH}	$\overline{WE}$ pulse width	50	-	ns
t_{WEHH}	$\overline{WE}$ pulse width high	20	-	ns
t_{CES}	$\overline{CE}$ setup time	0	-	ns
t_{CEH}	$\overline{CE}$ hold time	0	-	ns
t_{OES}	$\overline{OE}$ setup time	0	-	ns
t_{OEHP}	$\overline{OE}$ hold time (data polling and toggle)	10	-	ns
t_{OEHT}	$\overline{OE}$ high-level hold time (toggle)	20	-	ns
t_{PPW}	Auto Program time	16 (Note 1)	-	μs
t_{PCEW}	Auto Chip Erase time	30 (Note 1)	-	s
t_{PBEW}	Auto Block Erase time	3 (Note 1)	-	s
t_{VDS}	DVCC (3V) setup	500	-	μs
t_{BUSY}	RDY/BSY delay to program/erase valid	20	-	ns
t_{RP}	$\overline{RESET}$ pulse width	6	-	μs
t_{READY}	$\overline{RESET}$ low to Read mode	-	20	μs
t_{RB}	RDY/BSY recovery time	0	-	ns
t_{RH}	$\overline{RESET}$ recovery time	500	-	ns
t_{PPLH}	$\overline{WE}$ low-level hold time (Block Protect)	100	-	μs
t_{PAS}	Protect address setup time	0	-	ns
t_{PAH}	Protect address hold time	0	-	ns
t_{CESP}	$\overline{CE}$ setup time (Block Protect)	4	-	μs
t_{CEHP}	$\overline{CE}$ hold time (Block Protect)	8	-	μs

Note 1: Typical values

Note 2: AC test conditions:

/Input pulse levels: 2.4 V/0.4 V

/Input pulse rise/fall time (10% to 90%) : 5 ns

/Input timing measurement reference levels: 1.5 V/1.5 V

/Output timing measurement reference levels: 1.5 V/1.5 V

/Output load capacitance (CL): 100 pF

Note 3: Other AC characteristics are the same as for the TX1942CY/CZ.

4.6.2 Timing Charts

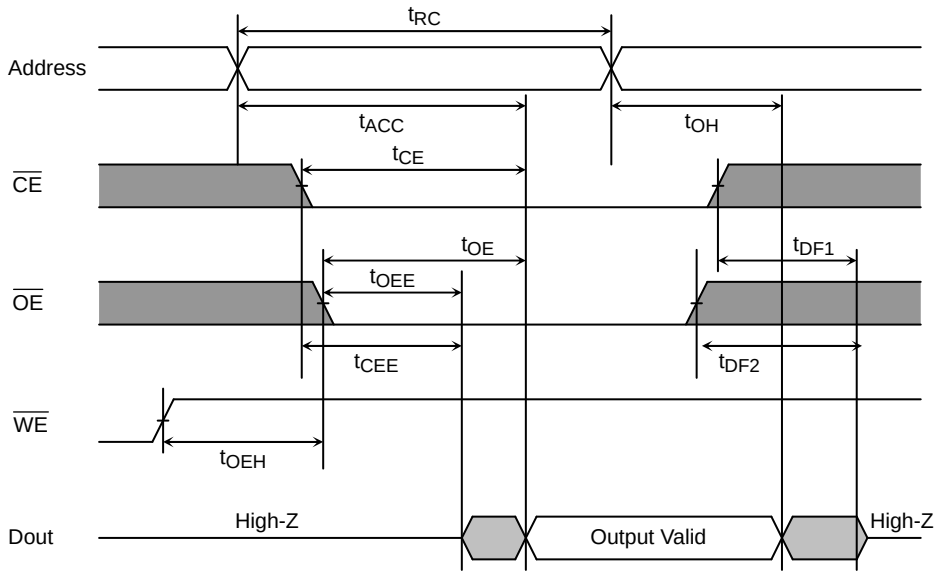
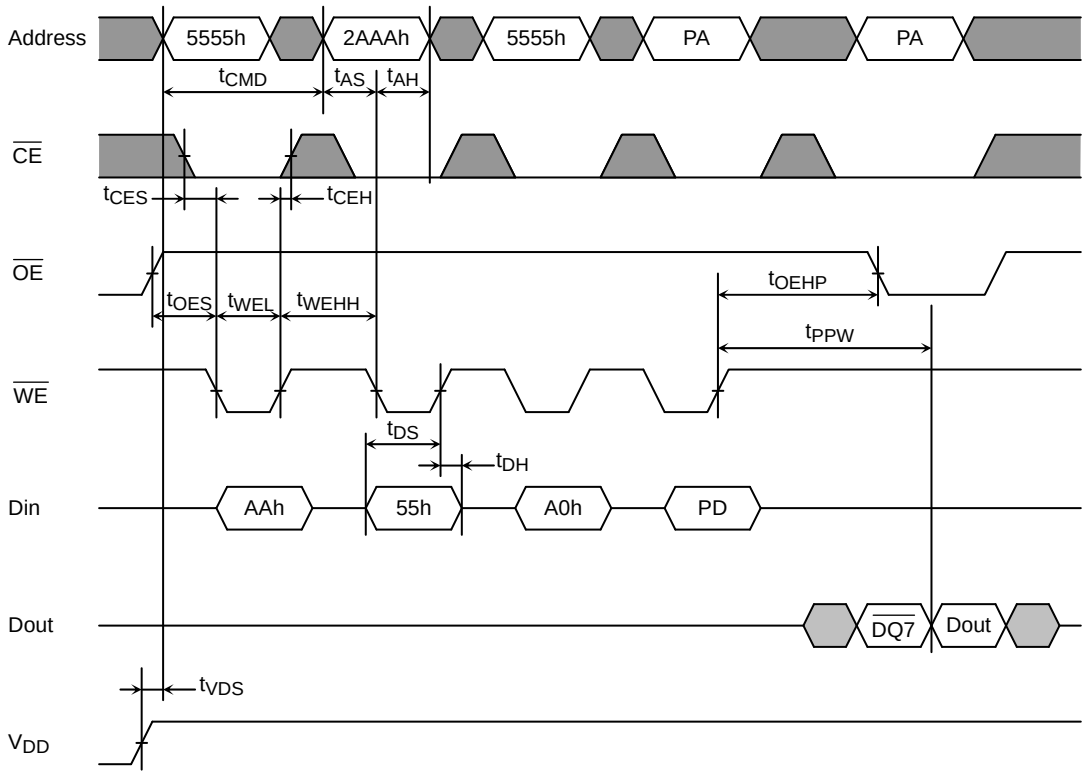
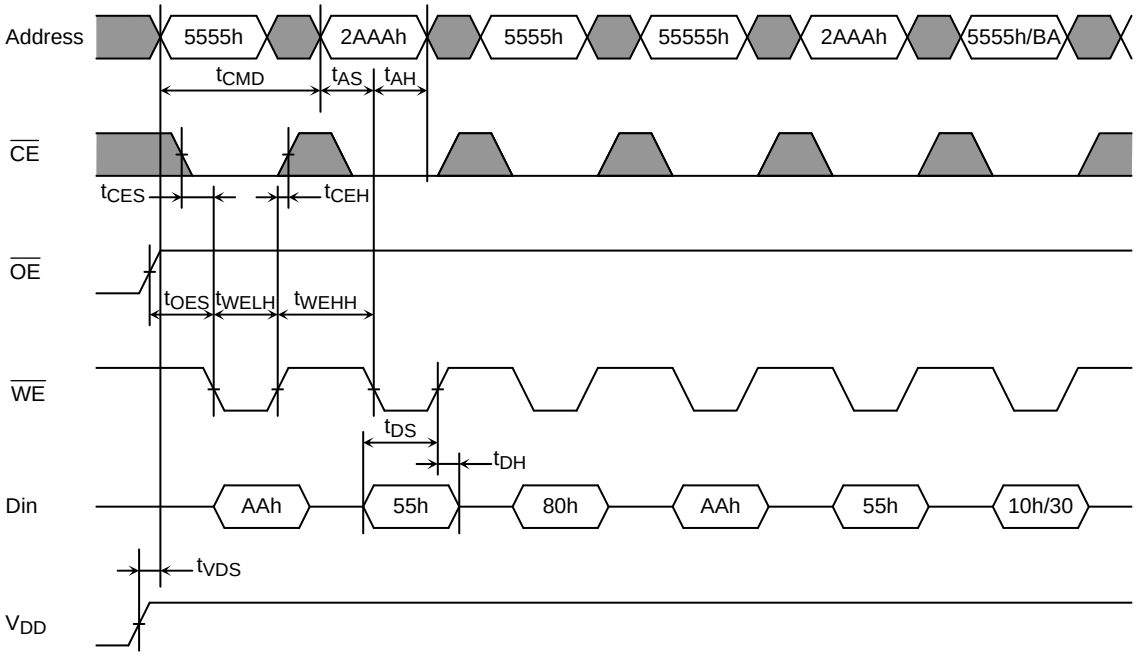


Figure 4.6.1 Read/ID Read Operation Timings



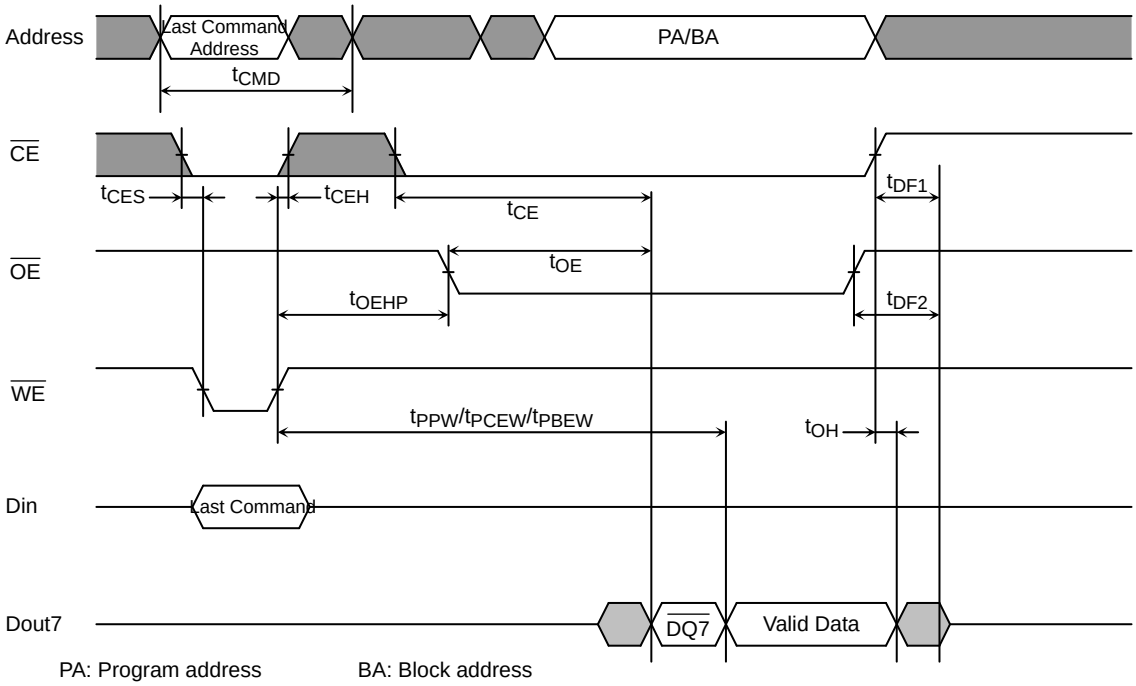
PA: Program address PD: Program data

Figure 4.6.2 Auto Program Operation Timings



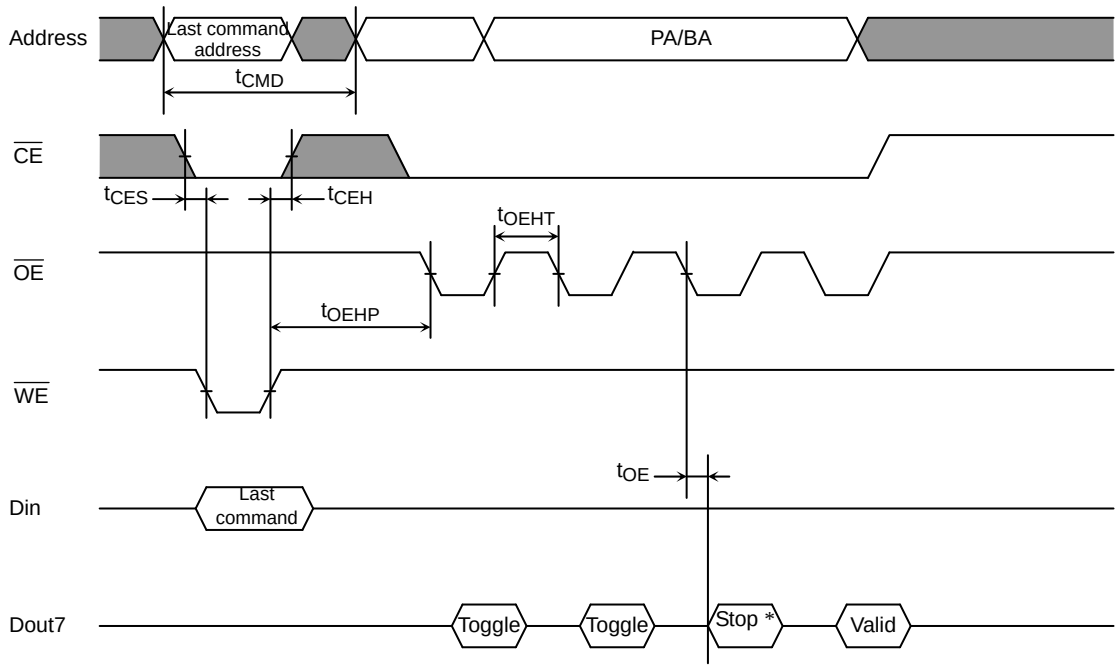
BA: Block address for Auto Block Erase

Figure 4.6.3 Auto Chip Erase /Auto Block Erase Operation Timings



PA: Program address BA: Block address

Figure 4.6.4 Data Polling Timings During Embedded Algorithms



* The output on the toggle bit stops when an embedded algorithm completes.

PA: Program address BA: Block address

Figure 4.6.5 Toggle Bit Operation Timings During Embedded Algorithms

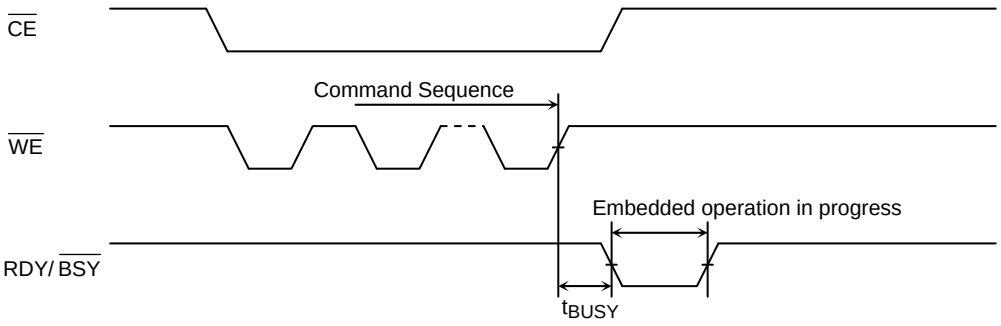


Figure 4.6.6 RDY/$\overline{\text{BSY}}$ Status Timings During Embedded Operations

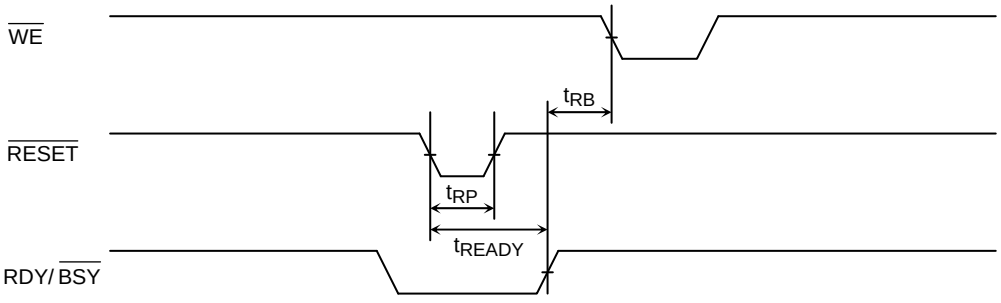


Figure 4.6.7 Hardware Reset Timings

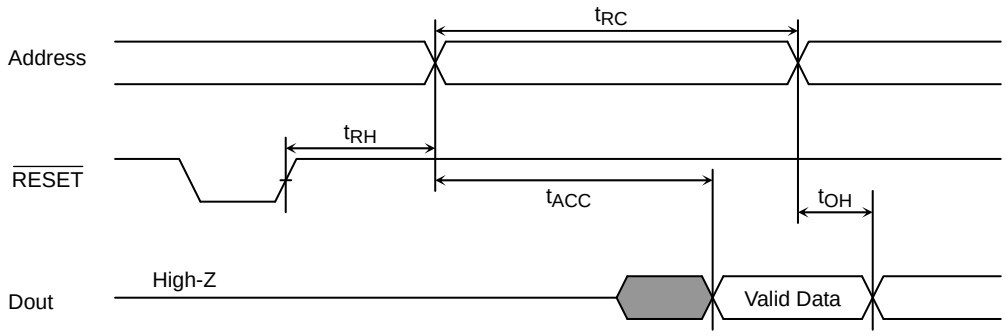


Figure 4.6.8 Read Timings After $\overline{\text{RESET}}$

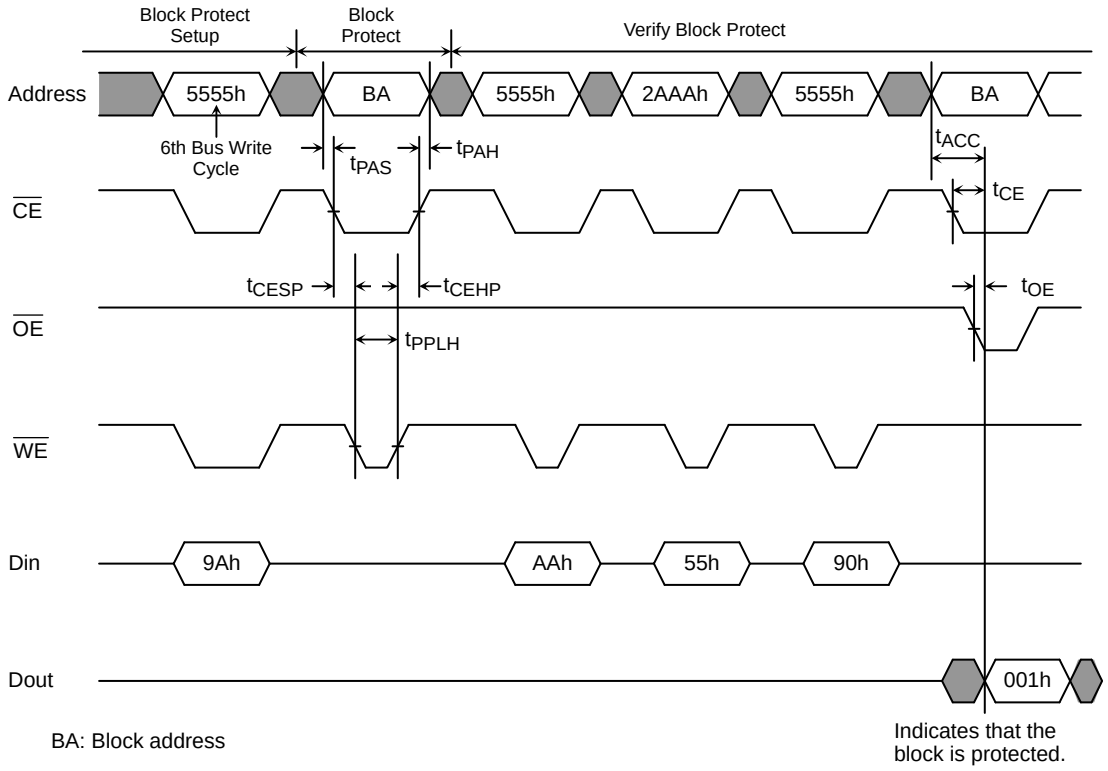


Figure 4.6.9 Block Protect Operation