# Toshiba Semiconductor and Storage - TMPM366FDXBG Datasheet



**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | ARM® Cortex®-M3 |
| Core Size | 32-Bit Single-Core |
| Speed | 48MHz |
| Connectivity | I²C, SIO, UART/USART |
| Peripherals | DMA, PWM, WDT |
| Number of I/O | 74 |
| Program Memory Size | 512KB (512K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 64K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | A/D 12x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 109-TFBGA |
| Supplier Device Package | 109-TFBGA (9x9) |
| Purchase URL | https://www.e-xfl.com/product-detail/toshiba-semiconductor-and-storage/tmpm366fdxbg |

# TMPM366FDXBG/FYXBG/FWXBG

The TMPM366FDXBG/FYXBG/FWXBG is a 32-bit RISC microprocessor series with an ARM Cortex™-M3 microprocessor core.

| Product Name | ROM (FLASH) | RAM | Package |
|---|---|---|---|
| TMPM366FDXBG | 512 Kbyte | 64 Kbyte | P-TFBGA109-0909-0.65-002 |
| TMPM366FYXBG | 256 Kbyte | 48 Kbyte | |
| TMPM366FWXBG | 128 Kbyte | 32 Kbyte | |

Features of the TMPM366FDXBG/FYXBG/FWXBG are as follows:

## 1.1 Features

1. ARM Cortex-M3 microprocessor core
   a. Improved code efficiency has been realized through the use of Thumb® -2 instruction.
      - New 16-bit Thumb instructions for improved program flow
      - New 32-bit Thumb instructions for improved performance
      - New Thumb mixed 16-/32-bit instruction set can produce faster, more efficient code.
   b. Both high performance and low power consumption have been achieved.

      [High performance]
      - A 32-bit multiplication (32×32=32 bit) can be executed with one clock.
      - Division takes between 2 and 12 cycles depending on dividend and devisor

      [Low power consumption]
      - Optimized design using a low power consumption library
      - Standby function that stops the operation of the micro controller core
   c. High-speed interrupt response suitable for real-time control
      - An interruptible long instruction.
      - Stack push automatically handled by hardware.

2. On chip program memory and data memory
   - On chip SRAM : 64 Kbyte / 48 Kbyte / 32 Kbyte
   - On chip Flash ROM : 512 Kbyte / 256 Kbyte / 128 Kbyte

3. External bus interface (EBIF)
   - Up to 16Mbytes access area (Program / Data)
   - External data bus (Separate / Multiplex): 8 /16bit bus width
   - Chip select / Wait controller: 2 channels

4. DMA controller (DMAC) : 2 units 4 channels

The example of warm-up function setup.

Table 8-1 <example> from STOP mode to NORMAL mode transition (internal high-speed oscillator is selected)

| | | |
|---|---|---|
| | CGOSCCR<WUODR[11:0]> = "0x9C4" | : Specify the warm-up time |
| ↻ | CGOSCCR<WUODR[11:0]> read | : Confirm warm-up time reflecting<br>Repaet until the read data is "0x9C4". |
| | CGOSCCR<XEN2> = "1" | : Internal high-speed oscillator (fosc) enable |
| | CGOSCCR<WUEON> = "1" | : Start the warm-up timer (WUP) |
| ↻ | CGOSCCR<WUEF> read | : Wait until the state becomes "0" (warm-up is finished) |

Note 1: It is not required the warm-up time in using the external clock to be stabled.

Note 2: The warm-up timer operates according to the oscillation clock, and it may contain errors if there is any fluctuation in the oscillation frequency. Therefore, the warm-up time should be taken as approximate time.

Note 3: After setting warm-up count value to OSCCR<WUDOR>, wait until confirming of the value to be reflected, then change to the standby mode by WFI instruction.

Note 4: When returning from STOP1/STOP2 mode, related bits CGPLLSEL<PLLSEL>, CGOSCCR<HWUPSEL>, <OSCSEL>, <XEN2>, <XEN1> and <PLLON> are initialized in order to start internal high-speed oscillator and CGOSCCR<WUDOR[11:0]> is not initialized.

### 10.2.6.2 PFDATA (Port F data register)

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| bit symbol | PF7 | PF6 | PF5 | PF4 | PF3 | PF2 | PF1 | PF0 |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Bit Symbol | Type | Function |
|---|---|---|---|
| 31-8 | − | R | Read as "0". |
| 7-0 | PF7-PF0 | R/W | Port F data register |

### 10.2.6.3 PFCR (Port F output control register)

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| bit symbol | PF7C | PF6C | PF5C | PF4C | PF3C | PF2C | PF1C | PF0C |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Bit Symbol | Type | Function |
|---|---|---|---|
| 31-8 | − | R | Read as "0". |
| 7-0 | PF7C-PF0C | R/W | Output<br>0: Disable<br>1: Enable |

### 10.2.9.4 PIFR1(Port I function register 1)

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| bit symbol | PI7F1 | PI6F1 | PI5F1 | PI4F1 | PI3F1 | PI2F1 | PI1F1 | PI0F1 |
| After reset | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

| Bit | Bit Symbol | Type | Function |
|---|---|---|---|
| 31-8 | − | R | Read as "0". |
| 7 | PI7F1 | R/W | 0: PORT<br>1: $\overline{\text{TRST}}$ |
| 6 | PI6F1 | R/W | 0: PORT<br>1: TDI |
| 5 | PI5F1 | R/W | 0: PORT<br>1: TDO/SWV |
| 4 | PI4F1 | R/W | 0: PORT<br>1: TMS/SWDIO |
| 3 | PI3F1 | R/W | 0: PORT<br>1: TCK/SWCLK |
| 2 | PI2F1 | R/W | 0: PORT<br>1: TRACECLK |
| 1 | PI1F1 | R/W | 0: PORT<br>1: TRACEDATA0 |
| 0 | PI0F1 | R/W | 0: PORT<br>1: TRACEDATA1 |

### 10.4.7 Port G Setting

Table 10-12 Port Setting List (Port G)

| Pin | Port Type | Function | After reset | PGCR | PGFR1 | PGFR2 | PGFR3 | PGFR4 | PGFR5 | PGOD | PGPUP | PGIE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PG0 | FT1 | Input Port | | 0 | 0 | 0 | 0 | 0 | 0 | x | x | 1 |
| | | Output Port | | 1 | 0 | 0 | 0 | 0 | 0 | x | x | 0 |
| | FT1 | SDA0 (Input/Output) | | 1 | 1 | 0 | 0 | 0 | 0 | 1 | x | 1 |
| | | SO0 (Output) | | 1 | 1 | 0 | 0 | 0 | 0 | x | x | 0 |
| | FT9 | A3 (Output) | | 1 | 0 | 1 | 0 | 0 | 0 | x | x | 0 |
| | FT1 | TX02 (Output) | | 1 | 0 | 0 | 0 | 1 | 0 | x | x | 0 |
| | | IROUT (Output) | | 1 | 0 | 0 | 0 | 0 | 1 | x | x | 0 |
| PG1 | FT1 | Input Port | | 0 | 0 | 0 | 0 | 0 | 0 | x | x | 1 |
| | | Output Port | | 1 | 0 | 0 | 0 | 0 | 0 | x | x | 0 |
| | FT1 | SCL0 (Input/Output) | | 1 | 1 | 0 | 0 | 0 | 0 | 1 | x | 1 |
| | | SI0 (Input) | | 0 | 1 | 0 | 0 | 0 | 0 | x | x | 1 |
| | FT9 | A4 (Output) | | 1 | 0 | 1 | 0 | 0 | 0 | x | x | 0 |
| | FT1 | TB3IN0 (Input) | | 0 | 0 | 0 | 1 | 0 | 0 | x | x | 1 |
| | | RX02 (Input) | | 0 | 0 | 0 | 0 | 1 | 0 | x | x | 1 |
| | | IRIN (Input) | | 0 | 0 | 0 | 0 | 0 | 1 | x | x | 1 |
| PG2 | FT1 | Input Port | | 0 | 0 | 0 | 0 | 0 | 0 | x | x | 1 |
| | | Output Port | | 1 | 0 | 0 | 0 | 0 | 0 | x | x | 0 |
| | FT1 | SCK0 (Input) | | 0 | 1 | 0 | 0 | 0 | 0 | x | x | 1 |
| | | SCK0 (Output) | | 1 | 1 | 0 | 0 | 0 | 0 | x | x | 0 |
| | FT9 | A5 (Output) | | 1 | 0 | 1 | 0 | 0 | 0 | x | x | 0 |
| | FT1 | TB3IN1 (Input) | | 0 | 0 | 0 | 1 | 0 | 0 | x | x | 1 |
| | | $\overline{\text{CTS2}}$ (Input) | | 0 | 0 | 0 | 0 | 1 | 0 | x | x | 1 |
| PG3 | FT1 | Input Port | | 0 | 0 | 0 | 0 | 0 | 0 | x | x | 1 |
| | | Output Port | | 1 | 0 | 0 | 0 | 0 | 0 | x | x | 0 |
| | FT4 | INT0 (Input) | | 0 | 1 | 0 | 0 | 0 | 0 | x | x | 1 |
| | FT9 | A6 (Output) | | 1 | 0 | 1 | 0 | 0 | 0 | x | x | 0 |
| | FT1 | TB4IN0 (Input) | | 0 | 0 | 0 | 1 | 0 | 0 | x | x | 1 |
| | FT1 | RIN2 (Input) | | 0 | 0 | 0 | 0 | 1 | 0 | x | x | 1 |
| PG4 | FT1 | Input Port | | 0 | 0 | 0 | 0 | 0 | 0 | x | x | 1 |
| | | Output Port | | 1 | 0 | 0 | 0 | 0 | 0 | x | x | 0 |
| | FT9 | A7 (Output) | | 0 | 0 | 1 | 0 | 0 | 0 | x | x | 0 |
| | FT1 | TB4IN1 (Input) | | 0 | 0 | 0 | 1 | 0 | 0 | x | x | 1 |
| | | $\overline{\text{RTS2}}$ (Input) | | 1 | 0 | 0 | 0 | 1 | 0 | x | x | 0 |
| PG5 | FT1 | Input Port | | 0 | 0 | 0 | 0 | 0 | 0 | x | x | 1 |
| | | Output Port | | 1 | 0 | 0 | 0 | 0 | 0 | x | x | 0 |
| | FT4 | INT1 (Input) | | 0 | 1 | 0 | 0 | 0 | 0 | x | x | 1 |
| | FT4 | USBPON (Input) | | 0 | 0 | 0 | 0 | 1 | 0 | x | x | 1 |

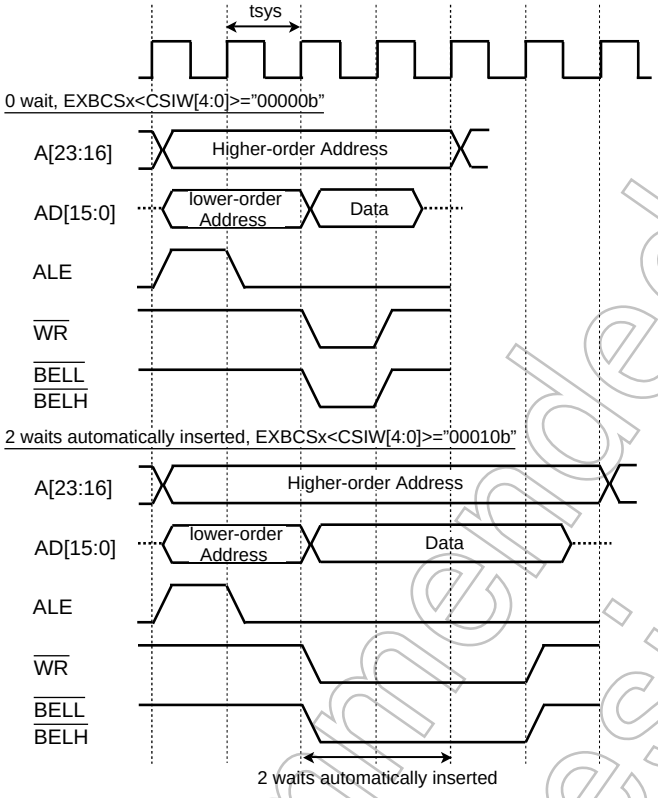



Figure 12-13 Write Operation Timing

## 12.6.4 EXBCSx (External Bus Chip Select Control Register)

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| bit symbol | CSR | | WRR | | | RDR | | |
| After reset | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| bit symbol | - | - | ALEW | | WRS | | RDS | |
| After reset | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| bit symbol | - | - | - | CSIW | | | | |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| bit symbol | - | - | - | - | - | CSW | | CSW0 |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bit | Bit Symbol | Type | Function |
|---|---|---|---|
| 31-30 | CSR[1:0] | R/W | Chip select ($\overline{CSx}$) Recovery cycle<br>00: None, 01: 1 cycle, 10: 2 cycles, 11: 4 cycles |
| 29-27 | WRR[2:0] | R/W | Write ($\overline{WR}$) Recovery cycle<br>000: None, 001: 1 cycle, 010: 2 cycles, 011: 3 cycles,<br>100: 4 cycles, 101: 5 cycles, 110: 6 cycles, 111: 8 cycles |
| 26-24 | RDR[2:0] | R/W | Read ($\overline{RD}$) Recovery cycle<br>000: None, 001: 1 cycle, 010: 2 cycles, 011: 3 cycles,<br>100: 4 cycles, 101: 5 cycles, 110: 6 cycles, 111: 8 cycles |
| 23-22 | − | R | Read as 0. |
| 21-20 | ALEW[1:0] | R/W | ALE wait cycle for multiplex bus<br>000: None, 001: 1 cycle, 010: 2 cycles, 011: 4 cycles |
| 19-18 | WRS[1:0] | R/W | Write ($\overline{WR}$) Setup cycle<br>000: None, 001: 1 cycle, 010: 2 cycles, 011: 4 cycles |
| 17-16 | RDS[1:0] | R/W | Read ($\overline{RD}$) Setup cycle<br>000: None, 001: 1 cycle, 010: 2 cycles, 011: 4 cycles |
| 15-13 | − | R | Read as 0. |
| 12-8 | CSIW[4:0] | R/W | Internal Wait (Automatically insertion)<br>0000: 0 wait, 0001: 1 wait, 0010: 2 waits, 0011: 3 waits,<br>0100: 4 waits, 0101: 5 waits, 0110: 6 waits, 0111: 7 waits,<br>1000: 8 waits, 1001: 9 waits, 1010: 10 waits, 1011: 11 waits,<br>1100: 12 waits, 1101: 13 waits, 1110: 14 waits, 1111: 15 waits |
| 7-4 | − | R | Read as 0. |
| 3 | − | R/W | Always write to "0" |
| 2-1 | CSW[2:1] | R/W | Data bus width<br>00: 8-bit, 01: 16-bit, Others: Prohibited |
| 0 | CSW0 | R/W | CS Enable<br>0: Disable, 1: Enable |

### 12.7.2 Connection Example for external 16-bit SRAM and NOR-Flash (Multiplex bus)

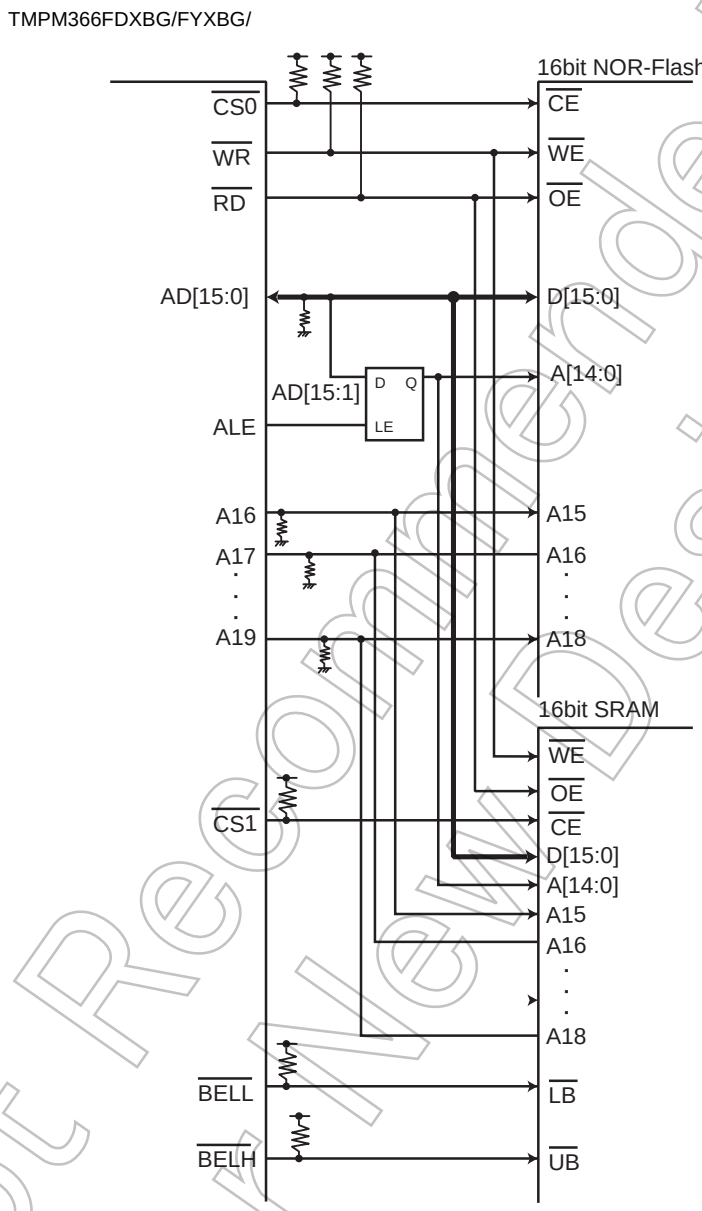



Figure 12-20 Connection Example for external 16-bit SRAM and NOR-Flash (Multiplex bus)

Table 13-2 Prescaler Output Clock Resolutions(fc = 48MHz)

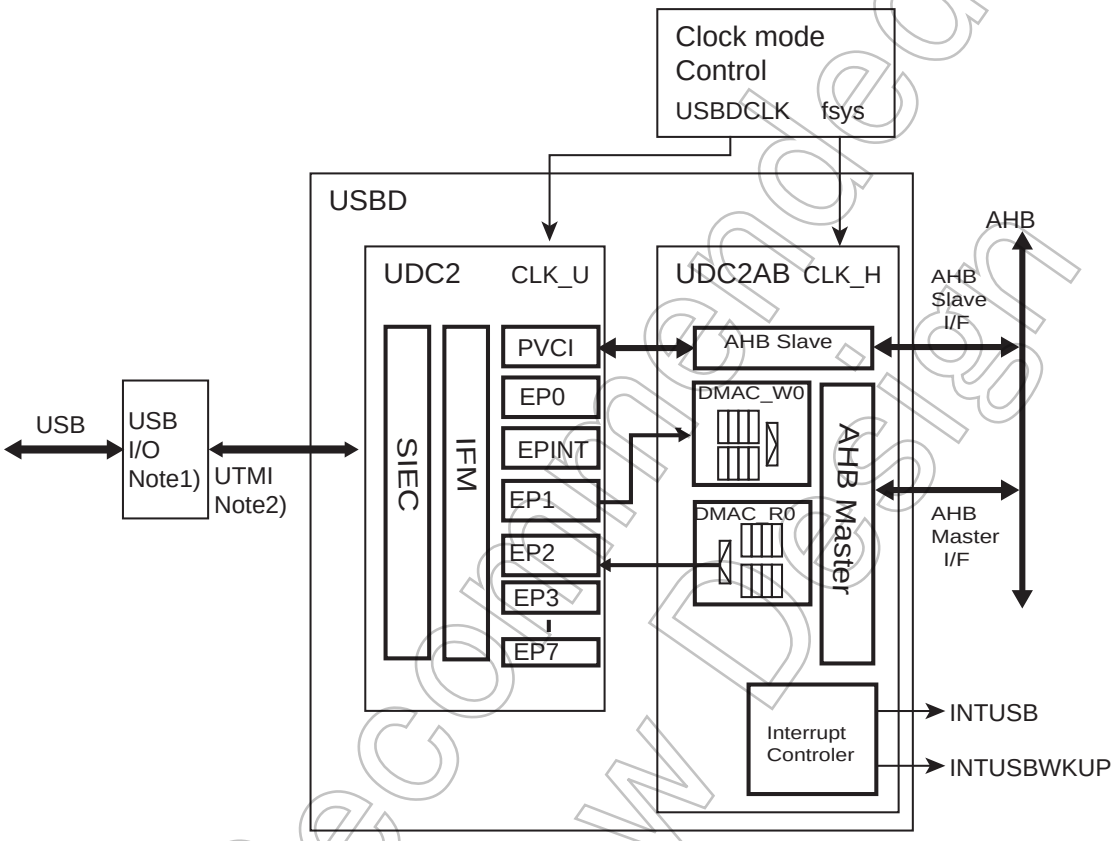
| Select peripheral clock CGSYSCR <FPSEL> | Clock gear value CGSYSCR <GEAR[2:0]> | Select prescaler clock CGSYSCR <PRCK[2:0]> | Prescaler output clock function | | |
|---|---|---|---|---|---|
| | | | φT1 | φT4 | φT16 |
| 1 (fc) | 000 (fc) | 000 (fperiph/1) | fc/$2^1$ (0.04 μs) | fc/$2^3$ (0.17 μs) | fc/$2^5$ (0.67 μs) |
| | | 001 (fperiph/2) | fc/$2^2$ (0.08 μs) | fc/$2^4$ (0.33 μs) | fc/$2^6$ (1.33 μs) |
| | | 010 (fperiph/4) | fc/$2^3$ (0.17 μs) | fc/$2^5$ (0.67 μs) | fc/$2^7$ (2.67 μs) |
| | | 011 (fperiph/8) | fc/$2^4$ (0.33 μs) | fc/$2^6$ (1.33 μs) | fc/$2^8$ (5.33 μs) |
| | | 100 (fperiph/16) | fc/$2^5$ (0.67 μs) | fc/$2^7$ (2.67 μs) | fc/$2^9$ (10.67 μs) |
| | | 101 (fperiph/32) | fc/$2^6$ (1.33 μs) | fc/$2^8$ (5.33 μs) | fc/$2^{10}$ (21.33 μs) |
| | 100 (fc/2) | 000 (fperiph/1) | − | fc/$2^3$ (0.17 μs) | fc/$2^5$ (0.67 μs) |
| | | 001 (fperiph/2) | fc/$2^2$ (0.08 μs) | fc/$2^4$ (0.33 μs) | fc/$2^6$ (1.33 μs) |
| | | 010 (fperiph/4) | fc/$2^3$ (0.17 μs) | fc/$2^5$ (0.67 μs) | fc/$2^7$ (2.67 μs) |
| | | 011 (fperiph/8) | fc/$2^4$ (0.33 μs) | fc/$2^6$ (1.33 μs) | fc/$2^8$ (5.33 μs) |
| | | 100 (fperiph/16) | fc/$2^5$ (0.67 μs) | fc/$2^7$ (2.67 μs) | fc/$2^9$ (10.67 μs) |
| | | 101 (fperiph/32) | fc/$2^6$ (1.33 μs) | fc/$2^8$ (5.33 μs) | fc/$2^{10}$ (21.33 μs) |
| | 101 (fc/4) | 000 (fperiph/1) | − | fc/$2^3$ (0.17 μs) | fc/$2^5$ (0.67 μs) |
| | | 001 (fperiph/2) | − | fc/$2^4$ (0.33 μs) | fc/$2^6$ (1.33 μs) |
| | | 010 (fperiph/4) | fc/$2^3$ (0.17 μs) | fc/$2^5$ (0.67 μs) | fc/$2^7$ (2.67 μs) |
| | | 011 (fperiph/8) | fc/$2^4$ (0.33 μs) | fc/$2^6$ (1.33 μs) | fc/$2^8$ (5.33 μs) |
| | | 100 (fperiph/16) | fc/$2^5$ (0.67 μs) | fc/$2^7$ (2.67 μs) | fc/$2^9$ (10.67 μs) |
| | | 101 (fperiph/32) | fc/$2^6$ (1.33 μs) | fc/$2^8$ (5.33 μs) | fc/$2^{10}$ (21.33 μs) |
| | 110 (fc/8) | 000 (fperiph/1) | − | − | fc/$2^5$ (0.67 μs) |
| | | 001 (fperiph/2) | − | fc/$2^4$ (0.33 μs) | fc/$2^6$ (1.33 μs) |
| | | 010 (fperiph/4) | − | fc/$2^5$ (0.67 μs) | fc/$2^7$ (2.67 μs) |
| | | 011 (fperiph/8) | fc/$2^4$ (0.33 μs) | fc/$2^6$ (1.33 μs) | fc/$2^8$ (5.33 μs) |
| | | 100 (fperiph/16) | fc/$2^5$ (0.67 μs) | fc/$2^7$ (2.67 μs) | fc/$2^9$ (10.67 μs) |
| | | 101 (fperiph/32) | fc/$2^6$ (1.33 μs) | fc/$2^8$ (5.33 μs) | fc/$2^{10}$ (21.33 μs) |
| | 111 (fc/16) | 000 (fperiph/1) | − | − | fc/$2^5$ (0.67 μs) |
| | | 001 (fperiph/2) | − | − | fc/$2^6$ (1.33 μs) |
| | | 010 (fperiph/4) | − | fc/$2^5$ (0.67 μs) | fc/$2^7$ (2.67 μs) |
| | | 011 (fperiph/8) | − | fc/$2^6$ (1.33 μs) | fc/$2^8$ (5.33 μs) |
| | | 100 (fperiph/16) | fc/$2^5$ (0.67 μs) | fc/$2^7$ (2.67 μs) | fc/$2^9$ (10.67 μs) |
| | | 101 (fperiph/32) | fc/$2^6$ (1.33 μs) | fc/$2^8$ (5.33 μs) | fc/$2^{10}$ (21.33 μs) |

Note 1: The prescaler output clock φTn must be selected so that φTn < fsys is satisfied (so that φTn is slower than fsys).
Note 2: Do not change the clock gear while the timer is operating.
Note 3: "−" denotes a setting prohibited.

## 14.2 System Structure

The USB device controller consists of the USB-Spec2.0 device controller (hereinafter called UDC2) and the bus bridge (hereinafter called UDC2AB) which connects the UDC2 and the AHB bus.

In this section, "14.2.1 AHB Bus Bridge (UDC2AB)" describes the configuration of the UDC2AB, and "14.2.2 Toshiba USB-Spec2.0 Device Controller (UDC2)" describes that of the UDC2.




Note1) TMPM366FDXBG/FYXBG/FWXBG has USB I/O which can be used for Full Speed mode not Low speed mode.
The word "PHY" in this section should be read as USB I/O.

Figure 14-1 Block diagram of the USB device controller

f. By detecting Resume on the USB bus, the $\overline{\text{WAKEUP}}$ output signal will be asserted to 0 asynchronously.By $\overline{\text{WAKEUP}}$ output signal, INTUSBWKUP occurs and the low-power consumption mode is cancelled. Then, supply of CLK_H starts.

g. With the supply of CLK_H, $\overline{\text{PHYSUSPEND}}$ output signal is automatically asserted to "1", and <phy_suspend> is zero-cleared.

Set CGUSBCTL<USBCLKEN> of the clock/mode control circuit to "1" to activate the CLK_U.

h. 2.5 s after the interrupt is asserted (time required for the signal to stabilize when VBUS is disconnected) and check UDFSPWCTL<pw_detect>. If the UDFSPWCTL<pw_detect> is "1", $\overline{\text{WAKEUP}}$ is asserted by Resume. If UDFSPWCTL<pw_detect> is "0", $\overline{\text{WAKEUP}}$ is asserted by disconnection of the VBUS.

i. To resume, perform the sequences below.To disconnect, perform the sequences of the "14.5.7.3 Resuming from the suspend state (disconnect)".

j. Clears the interrupt factor and <wakeup_en> to deassert the $\overline{\text{WAKEUP}}$ output signal. Set <suspend_resume_en> to "1".

k. Resumes from the suspended state.

## 15.4.12 SCxRST (RX FIFO Status Register)

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| bit symbol | - | - | - | - | - | - | - | - |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| bit symbol | ROR | - | - | - | - | RLVL | | |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Bit Symbol | Type | Function |
|---|---|---|---|
| 31-8 | − | R | Read as 0. |
| 7 | ROR | R | RX FIFO Overrun (Note)<br>0: Not generated<br>1: Generated |
| 6-3 | − | R | Read as 0. |
| 2-0 | RLVL[2:0] | R | Status of RX FIFO fill level.<br>000: Empty<br>001: 1 byte<br>010: 2 byte<br>011: 3 byte<br>100: 4 byte |

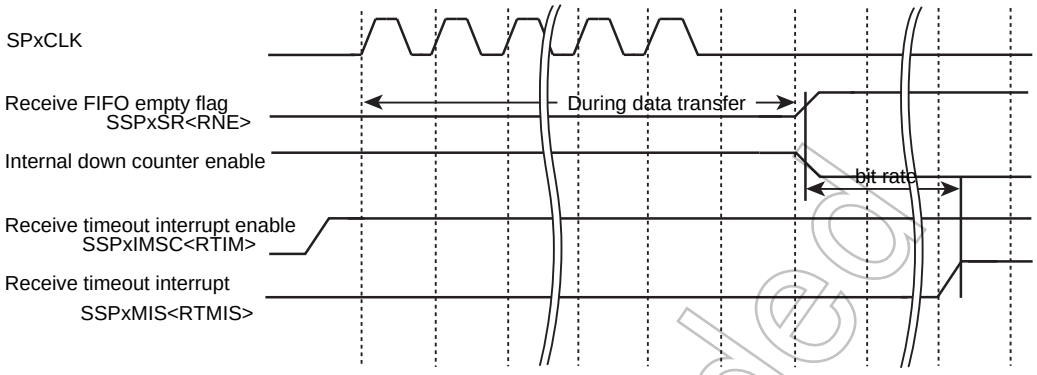
Note: **The <ROR> bit is cleared to "0" when receive data is read from the SCxBUF register.**

(1) UART transmit / receive data format.

| Transmit / receive data format | | | |
|---|---|---|---|
| START | DATA (LSB → MSB) | PARITY | STOP |

(2)Receive FIFO data format

| | Receive data (LSB → MSB) | | | | | | | | Framing error flag | Parity error flag | Break error flag | Overrun error flag |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit Number | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | |
| Receive 8-bit data | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | |
| Receive 7-bi data | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | |
| Receive 6-bit data | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | | | | |
| Receive 5-bit data | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | | | | |



### 18.5.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBIxCR2<TRX> to "1" configures the SBI as a transmitter. Setting <TRX> to "0" configures the SBI as a receiver.

At the slave mode:

- when data is transmitted in the addressing format.
- when the received slave address matches the value specified at SBIxI2CAR.
- when a general-call address is received; i.e., the eight bits following the start condition are all zeros.

If the value of the direction bit (R/$\overline{W}$) is "1", <TRX> is set to "1" by the hardware. If the bit is "0", <TRX> is set to "0".

As a master device, the SBI receives acknowledgement from a slave device. If the direction bit of "1" is transmitted, <TRX> is set to "0" by the hardware. If the direction bit is "0", <TRX> changes to "1". If the SBI does not receive acknowledgement, <TRX> retains the previous value.

<TRX> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

If SBI is used in free data format, <TRX> is not changed by the hardware.

### 18.5.7 Configuring the SBI as a Master or a Slave

Setting SBIxCR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device. <MST> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

### 18.5.8 Generating Start and Stop Conditions

When SBIxSR<BB> is "0", writing "1" to SBIxCR2<MST, TRX, BB, PIN> causes the SBI to start a sequence for generating the start condition and to output the slave address and the direction bit prospectively written in the data buffer register. <ACK> must be set to "1" in advance.

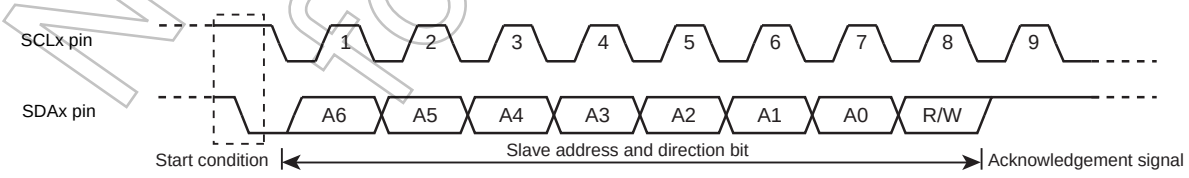



Figure 18-5 Generating the Start Condition and a Slave Address

When <BB> is "1", writing "1" to <MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. The contents of <MST, TRX, BB, PIN> should not be altered until the stop condition appears on the bus.

## 19.2 Configuration

Figure 19-1 shows the block diagram of the AD converter.

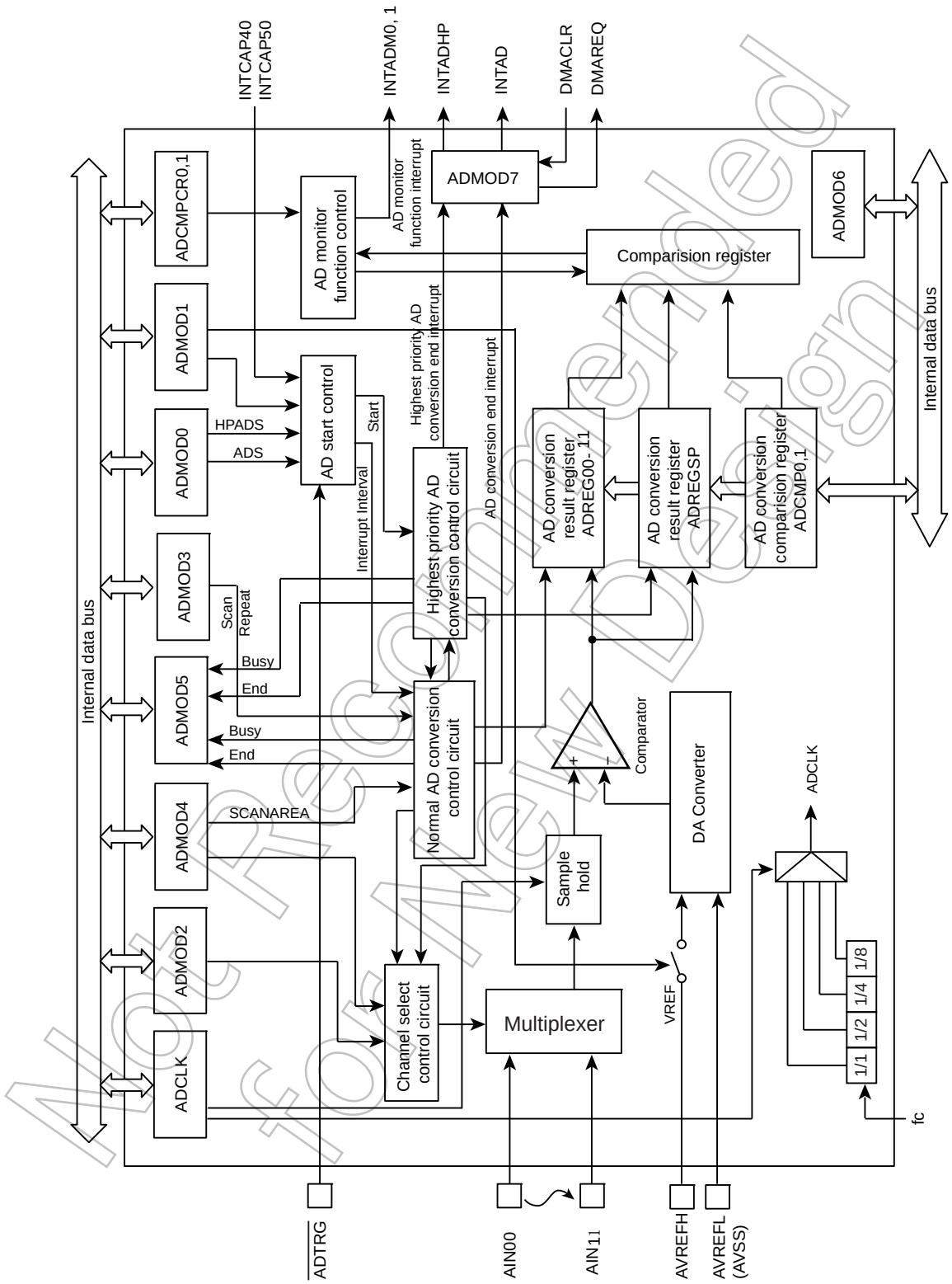
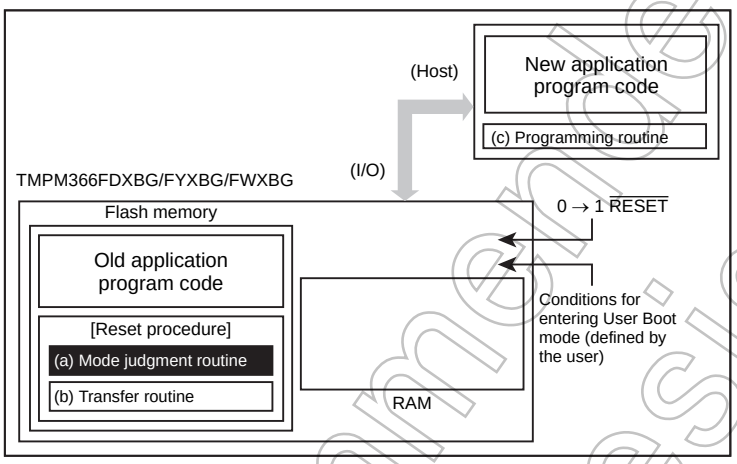



Figure 19-1 AD Converter Block Diagram

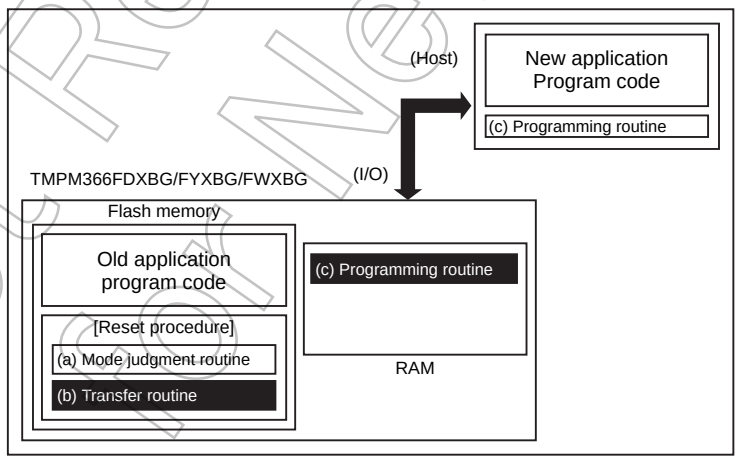
(2) Step-2

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMPM366FDXBG/FYXBG/FWXBG 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 disabled while in User Boot mode).



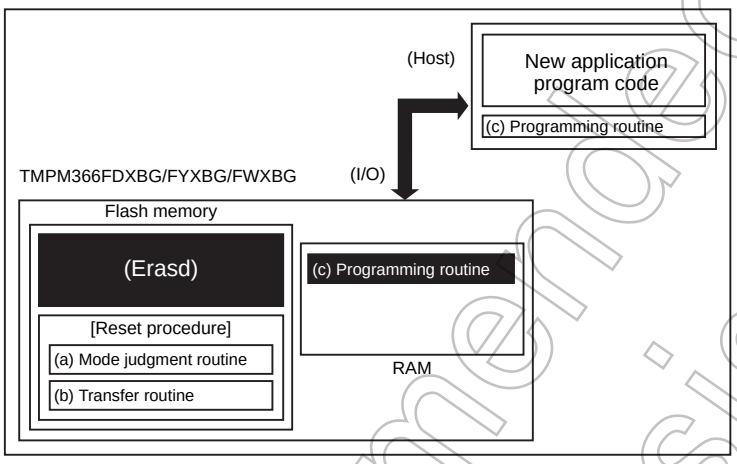

(3) Step-3

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to the TMPM366FDXBG/FYXBG/FWXBG on-chip RAM.



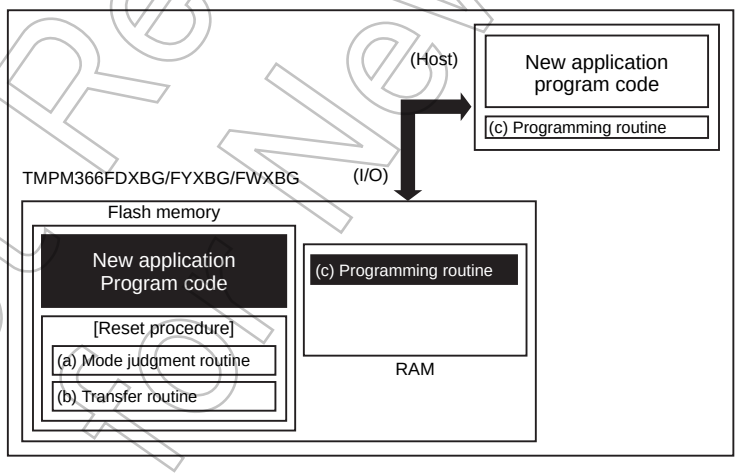
(4) Step-4

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) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user program area must be set.



# 24. Package Dimensions

Type: P-TFBGA109-0909-0.65-002 Unit:mm