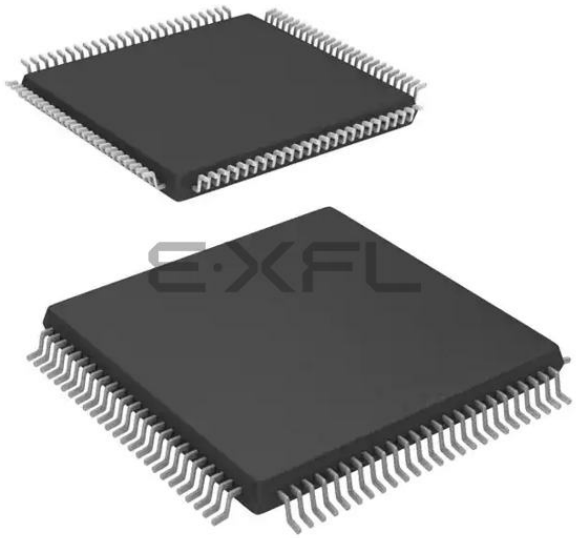


Welcome to E-XFL.COM

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	H8/300H
Core Size	16-Bit
Speed	25MHz
Connectivity	SCI, SmartCard
Peripherals	DMA, PWM, WDT
Number of I/O	70
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df3069rx25wv

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.5 shows the stack.

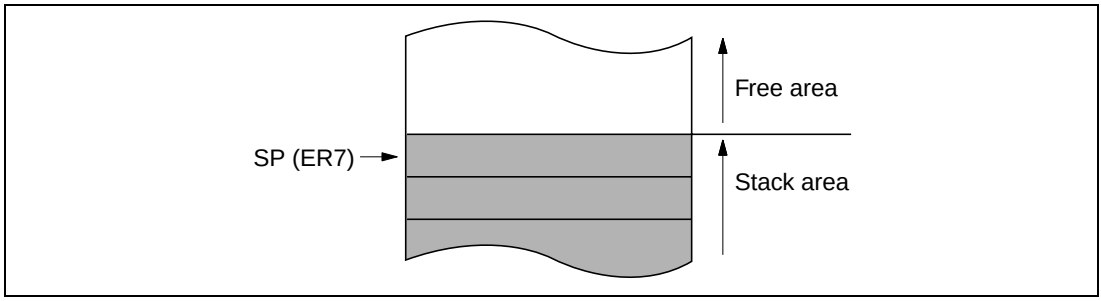


Figure 2.5 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC) and the 8-bit condition code register (CCR).

Program Counter (PC): This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. When an instruction is fetched, the least significant PC bit is regarded as 0.

Condition Code Register (CCR): This 8-bit register contains internal CPU status information, including the interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

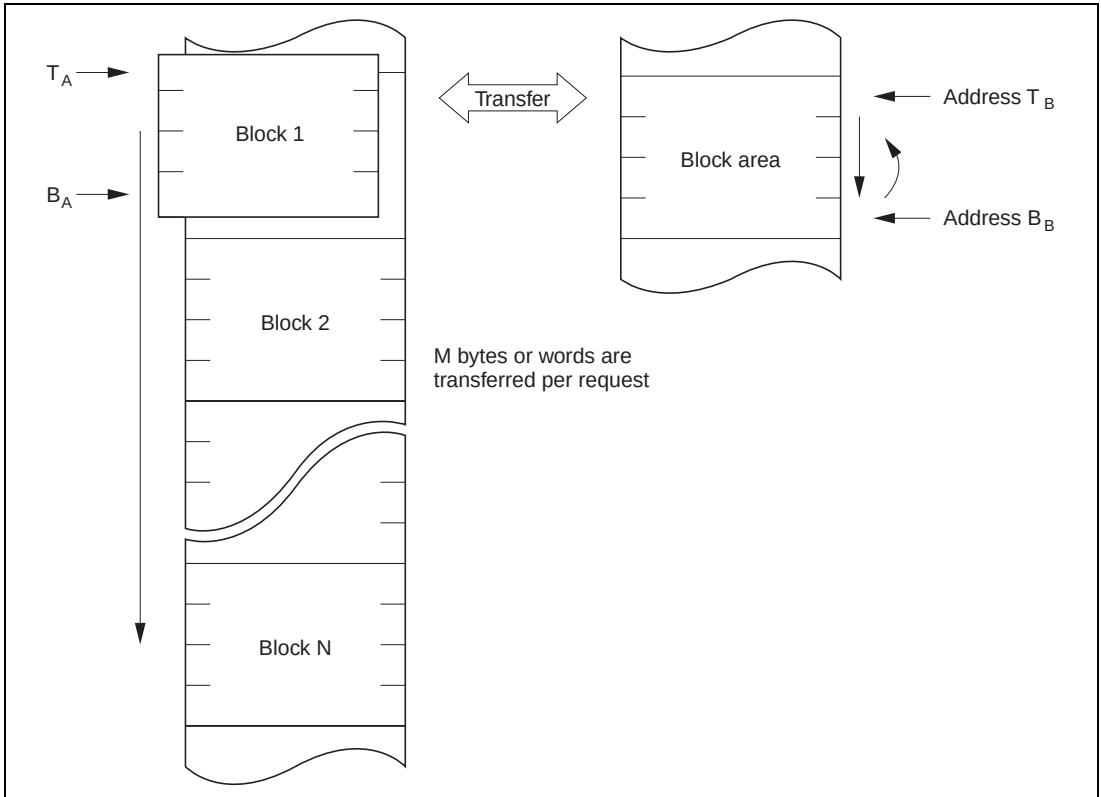
Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an interrupt mask bit. For details see section 5, Interrupt Controller.

Bit 5—Half-Carry Flag (H): When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.

If M (1 to 255) is the size of the block transferred at each request and N (1 to 65,536) is the number of blocks to be transferred, then ETCRAH and ETCRAL should initially be set to M and ETCRB should initially be set to N.

Figure 7.10 illustrates how block transfer mode operates. In this figure, bit TMS is cleared to 0, meaning the block area is the destination.



[Legend]
L_A = initial setting of MARA
L_B = initial setting of MARB
M = initial setting of ETCRAH and ETCRAL
N = initial setting of ETCRB
$T_A = L_A$
$B_A = L_A + SAIDE \bullet (-1)^{SAID} \bullet (2^{DTSZ} \bullet M - 1)$
$T_B = L_B$
$B_B = L_B + DAIDE \bullet (-1)^{DAID} \bullet (2^{DTSZ} \bullet M - 1)$

Figure 7.10 Operation in Block Transfer Mode

Figure 7.16 shows the timing when the DMAC is activated by the falling edge of $\overline{\text{DREQ}}$ in normal mode.

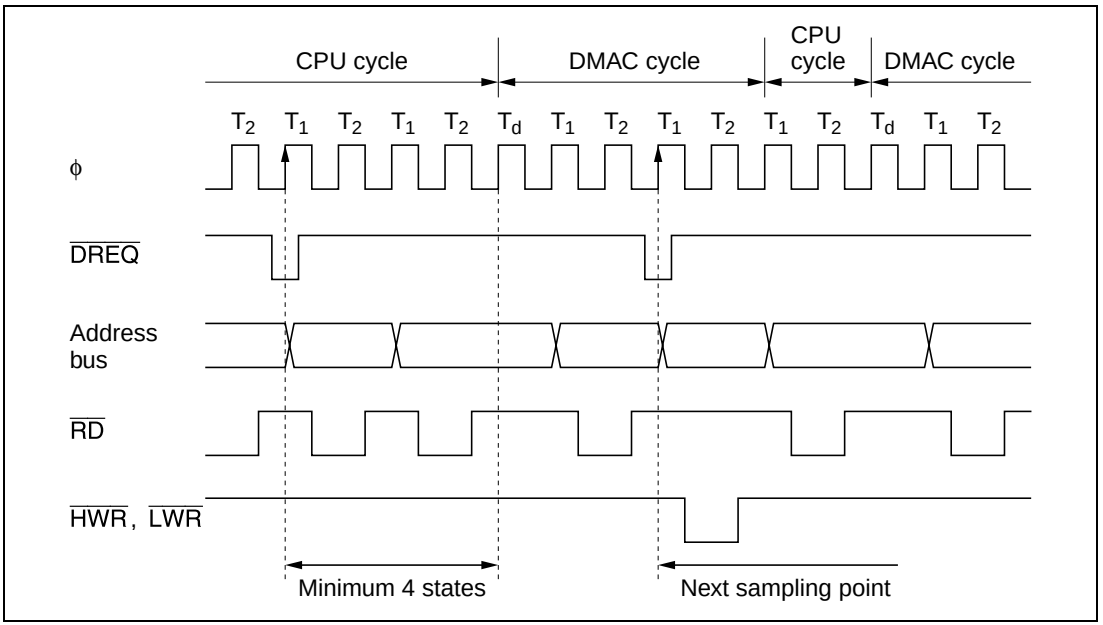


Figure 7.16 Timing of DMAC Activation by Falling Edge of $\overline{\text{DREQ}}$ in Normal Mode

Table 8.24 Port B Pin Functions (Mode 7)

Pin	Pin Functions and Selection Method

$PB_7/TP_{15}/RxD_2$

Bit RE in SCR of SCI2, bit SMIF in SCMR, bit NDER15 in NDERB, and bit PB_7DDR select the pin function as follows.

SMIF	0				1
RE	0			1	—
PB_7DDR	0	1	1	—	—
NDER15	—	0	1	—	—
Pin function	PB_7 input	PB_7 output	TP_{15} output	RxD_2 input	RxD_2 input

$PB_6/TP_{14}/TxD_2$

Bit TE in SCR of SCI2, bit SMIF in SCMR, bit NDER14 in NDERB, and bit PB_6DDR select the pin function as follows.

SMIF	0				1
TE	0			1	—
PB_6DDR	0	1	1	—	—
NDER14	—	0	1	—	—
Pin function	PB_6 input	PB_6 output	TP_{14} output	TxD_2 output	TxD_2 output*

Note: * Functions as the TxD2 output pin, but there are two states: one in which the pin is driven, and another in which the pin is at high-impedance.

$PB_5/TP_{13}/SCK_2$

Bit C/$\overline{A}$ in SMR of SCI2, bits CKE0 and CKE1 in SCR, bit NDER13 in NDERB, and bit PB_5DDR select the pin function as follows.

CKE1	0					1
C/$\overline{A}$	0				1	—
CKE0	0			1	—	—
PB_5DDR	0	1	1	—	—	—
NDER13	—	0	1	—	—	—
Pin function	PB_5 input	PB_5 output	TP_{13} output	SCK_2 output	SCK_2 output	SCK_2 input

PB_4/TP_{12}

Bit NDER12 in NDERB and bit PB_4DDR select the pin function as follows.

PB_4DDR	0	1	1
NDER12	—	0	1
Pin function	PB_4 input	PB_4 output	TP_{12} output