PIOC User Manual

Manual

Version: V1

https://wch-ic.com

1. Overview

A programmable protocol I/O microcontroller PIOC, namely eMCU, is embedded in some WCH chips. The eMCU is based on the reduced instruction set RISC8B core with a single clock cycle, runs at the main frequency of the system, has a program ROM with 2K instructions, 49 SFR registers and PWM timers/counters, and supports the protocol control of 2 I/O pins.

2. Feature

- RISC8B core, optimized single-cycle bit operation instruction set, fully static design, supporting system main frequency.
- The system SRAM with 4K bytes is reused as a program ROM with 2K capacity, which supports program pause and dynamic loading.
- Provide a 33-byte bidirectional register and a unidirectional register, and provide a 6-level independent stack.
- 8-bit automatic reload timer 0 is used for timeout reset, programmable clock or 8-bit PWM output.
- Support two universal bidirectional I/O protocol controls and input level change detection.

3. Instruction, Program Space and Stack

3.1 Instruction Set

EMCU adopts RISC8B core with reduced instruction set, with data width of 8 bits and instruction width of 16 bits. There are 66 instructions, all of which are single clock cycles, except that the jump instruction is double clock cycle and the program space reading and writing instruction is double cycle. For more information, please refer to RISC8B core instruction set and assembly tool document CHRISC8B.PDF.

3.2 Program ROM

EMCU's program ROM comes from 4-kbyte system SRAM multiplexing. When RB_MST_CLK_GATE is 0, it is dedicated to the host-side system SRAM, and when RB_MST_CLK_GATE is 1, it is dedicated to eMCU's program ROM.

The program space of eMCU is 2048 words, and the address of the program space ranges from 0x0000 to 0x07FF. The instruction at address 0 in the program ROM has a special purpose and will not be executed.

EMCU chip supports reading program space. The steps are as follows: first, write the lower 8 bits of the target address into SFR_INDIR_ADDR, and the upper 3 bits of the target address into the A register, and then execute the RDCODE instruction, so that the 16-bit data in the program space can be read out at one time, which is divided into high and low bytes, with the low byte returned to the A register and the high byte returned to the SFR_INDIR_ADDR. This operation can be used for double-byte table lookup.

EMCU chip does not support writing to program space.

3.3 Stack

EMCU has a stack memory with a depth of 6 levels and a data width of 11 bits. It is used to save the return address of the program when the subroutine is called, and it can also be used for PUSHAS instruction to save data such as variables and states in the program running into the stack.

3.4 Sleep and Wakeup

When RB_MST_CLK_GATE is 0, the eMCU program will be suspended. When the SLEEP or SLEEPX instruction is executed, eMCU will go to sleep. EMCU is a fully static design, and program pause has the same effect as sleep. EMCU supports two wake-up modes after sleep: the level change wake-up of the pin with level change detection turned on, and the change wake-up of SB_MST_CLK_GATE from 0 to 1.

3.5 Event Wait Instruction

WAITB instruction supports eight kinds of event waiting, and when the event is invalid, it keeps executing WAITB waiting, and the program counter PC remains unchanged. When the target event is detected, exit the wait and execute the next instruction. Optional start timer timeout reset during waiting.

The instruction parameter WB_DATA_SW_MR_0 means to wait until SB_DATA_SW_MR is 0 and exit directly if it is already 0;

The instruction parameter WB_BIT_CYC_TAIL_1 indicates to wait until SB_BIT_CYC_TAIL is 1 and exit directly if it is already 1;

The instruction parameter WB_PORT_I0_FALL means to wait until SB_PORT_IN0 detects the falling edge to exit, and if it has been detected before, it will exit directly, and the falling edge detection record will be cleared when exiting;

The instruction parameter WB_PORT_I0_RISE means to wait until SB_PORT_IN0 detects the rising edge to exit. If it has been detected before, it will exit directly, and the rising edge detection record will be cleared when exiting.

The instruction parameter WB_DATA_MW_SR_1 means to wait until SB_DATA_MW_SR is 1 and exit directly if it is already 1;

The instruction parameter WB_PORT_XOR1_1 indicates to wait until SB_PORT_XOR1 is 1 and exit directly if it is already 1;

The instruction parameter WB_PORT_XOR0_0 means to wait until SB_PORT_XOR0 is 0, and exit directly if it is already 0;

The instruction parameter WB_PORT_XOR0_1 indicates to wait until SB_PORT_XOR0 is 1 and exit directly if it is already 1.

3.6 Bit Transfer Instruction

The bit transfer instruction supports 2-bit operation registers: register SFR _ Indian _ Addr corresponding to BP1F and BG1F, and register SFR_DATA_EXCH corresponding to BP2F and BG2F.

Bit transfer instructions BP1F and BP2F support four independent bit outputs: 0# parameter BIO_FLAG_C corresponds to SB_FLAG_C bit, 1# parameter BO_BIT_TX_O0 corresponds to SB_BIT_TX_O0, 2# parameter BO_PORT_OUT0 corresponds to SB_PORT_OUT0 and 3# parameter BO_PORT_OUT1.

The bit transfer instructions BG1F and BG2F support four independent bit inputs: 0# parameter BIO_FLAG_C corresponds to SB_FLAG_C bit, 1# parameter BI_BIT_RX_I0 corresponds to SB_BIT_RX_I0, 2# parameter

BI_PORT_IN0 corresponds to SB_PORT_IN0, and 3# parameter BI_PORT_IN1.

The bit transfer instruction BCTC supports four independent bit inputs: 0# parameter BI_C_XOR_IN0 updates the bit SB_FLAG_C to the XOR result of SB_FLAG_C and SB_PORT_IN0, 1# parameter BI_BIT_RX_I0 corresponds to SB_BIT_RX_I0, and 2# parameter BI_PORT_IN0 corresponds to sb.

4. Data space, Registers and Addressing

The data space of eMCU includes 49 special function registers SFR, which are addressed by 8-bit addresses and range from 0x00-0x3F. The data width of all registers is 1 byte, that is, 8 bits of data.

Some SFR can be read and written or read-only on the host side, and the host side supports 8-bit, 16-bit or 32-bit width for reading and writing. The host side has write priority. When the eMCU and the host write into the same register at the same time, the eMCU write operation is automatically discarded.

The addressing modes of eMCU include: immediate number addressing, immediate number fast addressing, ordinary direct addressing, extended direct addressing, indirect addressing and bit addressing, and the last four are used to address registers.

The addressing range of indirect addressing is 0x000-0x0FF, covering all the registers of eMCU. EMCU provides two groups of indirect addressing registers, each of which consists of an address register and a data read-write port. Write the address of the destination register to the address register first, and then read and write the destination register through the read-write port.

The addressing range of extended direct addressing is 0x000-0x1FF, and the 9-bit register address is directly provided by the instruction, which is only applicable to the following two instructions that directly read and write registers: MOVRREGISTER, A or F instruction and MOVA register instruction.

The addressing range of ordinary direct addressing is 0x000-0x0FF, and the 8-bit register address is directly provided by the instruction, which is applicable to all direct address instructions except the above extended direct addressing instruction, such as CLR register, ADD register, BS register, bit, etc. Indirect addressing should be adopted when a wider range of addressing is needed.

Immediate fast addressing is used to quickly write the operands in the instruction code into the target register without a transfer. MOVA1F is used to quickly set the 1# register SFR_PORT_DIR, and MOVA2F is used to quickly set the 2# register SFR_PORT_IO. MOVA1P and MOVA2P are suitable for fast setting of all registers through indirect addressing, which is equivalent to immediate number addressing before indirect addressing in a single cycle.

For bit addressing, instructions directly provide 3-bit addresses, while registers can be addressed directly or indirectly. The addressing range of 0x000-0x0FF is realized respectively, so that any bit of any register can be addressed.

5. Special Function Register SFR

Some SFR or some bits are not really realized, and they are reserved bits. Reading is 0, and writing must keep the original value or write 0.

Address	SFR name	SFR description	Host-side read/write	Default value after reset
00H	SFR_INDIR_PORT	Indirect addressed data read-write port	UUUUUUUU	XXXXXXXX
01H	SFR_INDIR_PORT2	Indirect addressed 2 data	UUUUUUUU	XXXXXXXX

		read-write port		
02H	SFR_PRG_COUNT	Low byte of program counter PC	UUUUUUUU	00000000
03H	SFR_STATUS_REG	Status register	UUUUSUSU	0000-0-0
04H	SFR_INDIR_ADDR	Indirectly addressed address register	RRRRRRRR	XXXXXXXX
05H	SFR_TMR0_COUNT	Count register of timer 0	RRSSRRRR	00000000
06H	SFR_TIMER_CTRL	Control register of timer	RRRRRRRR	00000000
07H	SFR_TMR0_INIT	Initial value register of timer 0	RRRRRRRR	00000000
08H	SFR_BIT_CYCLE	Coded bit period register	WWWWWWWW	00000000
09H	SFR_INDIR_ADDR2	Address register of indirect addressing 2	RRRRRRRR	00000000
0AH	SFR_PORT_DIR	Port direction setting register	RRRRRRRR	00000000
0BH	SFR_PORT_IO	Port input output register	RRRRRRRR	XXXXXX00
0CH	SFR_BIT_CONFIG	Coded bit configuration register	WWWRRRRR	00010000
1CH	SFR_SYS_CFG	System configuration register	RRRWWWWW	00000000
1DH	SFR_CTRL_RD	EMCU read-write and host read-only register	RRRRRRRR	00000000
1EH	SFR_CTRL_WR	Host reads and writes and eMCU read-only register	WWWWWWWW	00000000
1FH	SFR_DATA_EXCH	Data exchange register	WWWWWWWW	00000000
20H	SFR_DATA_REG0	Data register 0	WWWWWWWW	00000000
21H	SFR_DATA_REG1	Data register 1	WWWWWWWW	00000000
22H	SFR_DATA_REG2	Data register 2	WWWWWWWW	00000000
23H	SFR_DATA_REG3	Data register 3	WWWWWWWW	00000000
24H	SFR_DATA_REG4	Data register 4	WWWWWWWW	00000000
25H	SFR_DATA_REG5	Data register 5	WWWWWWWW	00000000
26H	SFR_DATA_REG6	Data register 6	WWWWWWWW	00000000
27H	SFR_DATA_REG7	Data register 7	WWWWWWWW	00000000
28H	SFR_DATA_REG8	Data register 8	WWWWWWWW	00000000
29H	SFR_DATA_REG9	Data register 9	WWWWWWWW	00000000
2AH	SFR_DATA_REG10	Data register 10	WWWWWWWW	00000000
2BH	SFR_DATA_REG11	Data register 11	WWWWWWWW	00000000

2CH	SFR_DATA_REG12	Data register 12	WWWWWWWW	00000000
2DH	SFR_DATA_REG13	Data register 13	WWWWWWWW	00000000
2EH	SFR_DATA_REG14	Data register 14	WWWWWWWW	00000000
2FH	SFR_DATA_REG15	Data register 15	WWWWWWWW	00000000
30H	SFR_DATA_REG16	Data register 16	WWWWWWWW	00000000
31H	SFR_DATA_REG17	Data register 17	WWWWWWWW	00000000
32H	SFR_DATA_REG18	Data register 18	WWWWWWWW	00000000
33H	SFR_DATA_REG19	Data register 19	WWWWWWWW	00000000
34H	SFR_DATA_REG20	Data register 20	WWWWWWWW	00000000
35H	SFR_DATA_REG21	Data register 21	WWWWWWWW	00000000
36H	SFR_DATA_REG22	Data register 22	WWWWWWWW	00000000
37H	SFR_DATA_REG23	Data register 23	WWWWWWWW	00000000
38H	SFR_DATA_REG24	Data register 24	WWWWWWWW	00000000
39H	SFR_DATA_REG25	Data register 25	WWWWWWWW	00000000
3AH	SFR_DATA_REG26	Data register 26	WWWWWWWW	00000000
3BH	SFR_DATA_REG27	Data register 27	WWWWWWWW	00000000
3CH	SFR_DATA_REG28	Data register 28	WWWWWWWW	00000000
3DH	SFR_DATA_REG29	Data register 29	WWWWWWWW	00000000
3EH	SFR_DATA_REG30	Data register 30	WWWWWWWW	00000000
3FH	SFR_DATA_REG31	Data register 31	WWWWWWWW	00000000

The default values after reset are all represented by binary numbers, and the bit values are described as follows:

0: Always 0 after reset;

1: Always 1 after reset;

X: Reset does not affect the data, and the initial value of the data is uncertain;

-: It is always cleared after power-on reset, and data is not affected by system reset or host forced reset.

The description of host side reading and writing is as follows:

W: Bits that can be read or written;

R: Read-only bit;

U: Invisible bits, which cannot be read or written;

S: Read-only swap bit. The SFR_TIMER_CTRL visible on the host side is different from that on the eMCU side, in which the two bits SB_TMR0_ENABLE and SB_TMR0_OUT_EN are replaced by SB_GP_BIT_Y and SB_GP_BIT_X respectively.

5.1 Indirect Addressed Data Read-write Port SFR_INDIR_PORT

5.2 Indirect Addressed Adress Register SFR_INDIR_ADDR

5.3 Indirect Addressed 2 Data Read-write Port SFR_INDIR_PORT2

5.4 Indirect Addressed 2 Adress Register SFR_INDIR_ADDR2

These are two groups of indirect addressing registers, each of which consists of an address register and a data read-write port. Write the address of the destination register to the address register first, and then read and write the destination register through the read-write port.

The data read/write port SFR_INDIR_PORT with indirect addressing is the destination register for reading and writing the address specified by the address register SFR_INDIR_ADDR with indirect addressing. The indirectly addressed data read-write port SFR_INDIR_PORT2 is the destination register for reading and writing the address specified by the indirectly addressed address register SFR_INDIR_ADDR2.

The dedicated instruction MOVIP is used to load an 8-bit target address into SFR_INDIR_ADDR, and SFR_INDIR_ADDR can also be independently modified by other non-dedicated instructions.

The dedicated instruction MOVIA is used to load an 8-bit target address into SFR_INDIR_ADDR2, and SFR_INDIR_ADDR2 can also be independently modified by other non-dedicated instructions.

The address register SFR_INDIR_ADDR2 of the indirect address 2 has the feature of automatic increment. When the "MOV SFR_INDIR_PORT2, A" reading instruction is executed or the "MOVA SFR_INDIR_PORT2" writing instruction is executed on the RAM, SFR_INDIR_ADDR2 will automatically increment by 1 after the instruction is executed.

The special instructions MOVA1P and MOVA2P are used to write the operands in the instruction code into the target register directly and quickly, which is equivalent to immediate address first and then indirect address in a single cycle.

5.5 Status Register SFR_STATUS_REG

The status register contains the result status of ALU, general bit variables and stack usage flags. When performing arithmetic or logical operations, eMCU will set corresponding status bits according to the results, so that the program can make further processing after judgment. General-purpose bit variables can be defined by applications.

Timer timeout reset is used to monitor the timeout of WAITB waiting instruction. When SB_TMR0_ENABLE is turned off and SB_EN_TOUT_RST is enabled, the timer counts during the execution of WAITB instruction, and when it overflows, SB_TMR0_CYCLE is 1, which will lead to timeout reset.

The initial value of the timer is automatically reloaded every time the WAITB instruction exits the WAITB waiting because it meets the conditions. When SB_TMR0_ENABLE is 0, the execution period of non-WAITB instructions is not counted, that is, the timer timeout reset is only for WAITB instructions.

Address	Bit name	Description	Default value
Bit 7		(Reserved)	0
Bit 6		(Reserved)	0
Bit 5	SB_STACK_USED	Current stack usage flag: 0: Not used or completely popped; 1: Stacked	0
Bit 4	SB_EN_TOUT_RST	Timer timeout reset enabled: 0: Disabled; 1: Allow to reset eMCU when timer timeout overflows.	0

Bit 3	SB_GP_BIT_Y	General bit variable y, defined and used by application programs, is 0 after power-on reset, and is not affected by system reset or host forced reset.	-
Bit 2	SB_FLAG_Z	ALU zero flag, whether the result is 00H: 0: The result is not 0; 1: The result is 0.	0
Bit 1	SB_GP_BIT_X	General bit variable X, defined and used by the application, is 0 after power-on reset, and is not affected by system reset or host forced reset.	-
Bit 0	SB_FLAG_C	ALU carry flag, whether the result is carried or generated by shift: 0: The result has no carry; 1: The result is a carry.	0

5.6 Low Byte of Program Counter PC SFR_PRG_COUNT

The width of the program counter PC of eMCU is 11 bits. By reading and writing SFR_PRG_COUNT, the lower 8 bits of PC can be modified, but the upper 3 bits of PC will not be affected, so that short jumps and table lookup can be made within the range of 256 bytes.

5.7 Port Direction Setting Register SFR_PORT_DIR

5.8 Port Input Output Register SFR_PORT_IO

The port contains two bidirectional input and output pins, which can be independently set as the input direction or output direction by their respective direction setting bits SB_PORT_DIR. If it is the output direction, the data in SB_PORT_OUT is output to the pins. The default direction after reset is input, and SB_PORT_IN is used to obtain the current pin input level (When SB_PORT_IN_EDGE=0) or the pin input level of the previous cycle (When SB_PORT_IN_EDGE=1), which is about half a clock period earlier than the former.

Both pins have the function of level change detection. When the input level is inconsistent with the data in the output data register SB_PORT_OUT, the corresponding SB_PORT_XOR becomes 1.

There are more available input instructions and methods for IO0 pin, which is more suitable for input or bidirectional data than IO1.

The following table is a description of the port direction setting register SFR_PORT_DIR.

Address	Bit name	Description	Default value
Bit 7	SB_PORT_MOD3	Pin mode control, the host side defines the purpose.	0
Bit 6	SB_PORT_MOD2		0
Bit 5	SB_PORT_MOD1		0
Bit 4	SB_PORT_MOD0		0
Bit 3	SB_PORT_PU1	IO1 pin pull-up enable: 0: Disabled; 1: Enabled	0

Bit 2	SB_PORT_PU0	IO0 pin pull-up enable: 0: Disabled; 1: Enabled	0
Bit 1	SB_PORT_DIR1	IO1 pin direction control: 0: Input; 1: Output	0
Bit 0	SB_PORT_DIR0	IO0 pin direction control: 0: Input; 1: Output	0

The following table is a description of the port input and output register SFR_PORT_IO.

Address	Bit name	Description	Default value
Bit 7	SB_PORT_IN_XOR	XOR result of input state of IO1 pin and input state of IO0 pin: 0: The level of IO1 and IO0 is the same; 1: IO1 and IO0 have different levels.	X
Bit 6	SB_BIT_RX_I0	The bit of the IO0 pin receives the decoded data.	X
Bit 5	SB_PORT_IN1	IO1 pin input status (Refer to SB_PORT_IN_EDGE)	X
Bit 4	SB_PORT_IN0	IO0 pin input status (Refer to SB_PORT_IN_EDGE)	X
Bit 3	SB_PORT_XOR1	XOR result of input state of IO1 pin and SB_PORT_OUT1: 0: IO1 level SB_PORT_IN1 is the same as SB_PORT_OUT1; 1: IO1 level is different from SB_PORT_OUT1, which is used to detect the level change.	X
Bit 2	SB_PORT_XOR0	XOR result of input state of IO0 pin and SB_PORT_OUT0: 0: IO0 level SB_PORT_IN0 is the same as SB_PORT_OUT0; 1: IO0 level is different from SB_PORT_OUT0, which is used to detect the level change.	X
Bit 1	SB_PORT_OUT1	IO1 pin output data: 0: Output low; 1: High output	0
Bit 0	SB_PORT_OUT0	IO0 pin output data: 0: Output low; 1: High output The timer output signal takes precedence, and when SB_TMR0_OUT_EN is 1, it outputs SB_TMR0_CYCLE or its frequency-divided signal.	0

5.9 Count Register of Timer 0 SFR_TMR0_COUNT

5.10 Initial Value Register of Timer 0 SFR_TMR0_INIT

5.11 Control Register of Timer SFR_TIMER_CTRL

Timer0 is an 8-bit wide counter. The count value is in the register SFR_TMR0_COUNT, and SFR_TMR0_COUNT supports direct writing of data to load new values. The clock frequency of Timer0 is selected by SB_TMR0_FREQ. Timer0 can operate in timer mode or pulse width modulator mode.

Timer0 is in timer mode, and the initial count value is stored in the initial value register SFR_TMR0_INIT in advance. Timer0 always counts from the initial count value to 0FFH. When it overflows from 0FFH, Timer0 automatically reloads the initial value from the initial value register SFR_TMR0_INIT, and then counts from the initial count value to 0FFH. When the count value is counted from 0FEH to 0FFH, Timer 0 automatically sets the counting period signal SB_TMR0_CYCLE to 1; When the counting value overflows from 0FFH counting and returns to the initial counting value, the timer 0 automatically clears the counting period signal SB_TMR0_CYCLE to 0. The frequency of SB_TMR0_CYCLE is (SB_TMR0_FREQ clock frequency) /(256-SFR_TMR0_INIT), and its duty ratio is 1/(256-SFR_TMR0_INIT), that is, there is one clock period in each cycle where SB_TMR0_CYCLE is 1.

Timer0 In the pulse width modulator mode, Timer0 always counts from 00H to 0FFH, then overflows from 0FFH and returns to 00H to start a new count, and its cycle is always 256 clocks. When the count value is greater than the value in the initial value register SFR_TMR0_INIT of timer 0, timer 0 automatically sets the count period signal SB_TMR0_CYCLE to 1; When the count value overflows from 0FFH count back to 00H, Timer 0 automatically clears the count period signal SB_TMR0_CYCLE to 0. The duty cycle of SB_TMR0_CYCLE is (255-SFR_TMR0_INIT)/256.

SB_TMR0_OUT_EN is used to set whether the internal signal SB_TMR0_CYCLE or its divided signal is output to the IO0 pin. If the internal signal is 1, the pin outputs a high level, otherwise it outputs a low level. If the output of counting cycle signal is allowed, then in timer mode, the frequency-divided signal of SB_TMR0_CYCLE (frequency is reduced to half and duty ratio is 50%) will be output through the IO0 pin; In the pulse width modulator mode, the SB_TMR0_CYCLE signal will be output through the IO0 pin.

The following table is a description of the control register SFR_TIMER_CTRL of the timer.

Address	Bit name	Description	Default value
Bit 7	SB_EN_LEVEL1	The level change of IO1 pin activates the interrupt flag and wake-up enable: 0: No activation, no wake-up; 1: Allow activation of interrupt flag and wake-up.	0
Bit 6	SB_EN_LEVEL0	The level change of IO0 pin activates the interrupt flag and wake-up enable: 0: No activation, no wake-up; 1: Allow activation of interrupt flag and wake-up.	0
Bit 5	SB_TMR0_ENABLE	The counting of timer 0 is enabled: 0: No counting; 1: Allow counting	0
Bit 4	SB_TMR0_OUT_EN	The counting period signal output of timer 0 is allowed: 0: output is prohibited; 1: Allow output to the IO0 pin.	0
Bit 3	SB_TMR0_MODE	Working mode of Timer 0: 0: Timer mode; 1: Pulse Width Modulator Mode	0
Bit 2 Bit 1 Bit 0	SB_TMR0_FREQ	Select the clock frequency of Timer 0 (Divided by the system main frequency): 000: 1024X; 001: 256X; 010: 64X; 011: 16X; 100: 8X; 101: 4X; 110: 2X; 111: 1X	000

5.12 System Configuration Register SFR_SYS_CFG

5.13 Host Reads/Writes and eMCU Read-only Register SFR_CTRL_WR

5.14 eMCU Reads/Writes and Host Read-only Register SFR_CTRL_WR

The system configuration register provides some control bits of the host and status bits of bidirectional data exchange.

SB_INT_REQ is completely controlled by eMCU, which is used to request the host to interrupt or cancel the request. However, the host can clear SB_INT_REQ by writing to the read-only register SFR_CTRL_RD falsely, and the host's writing to SFR_CTRL_RD has no actual effect on the data in it.

SB_DATA_SW_MR is automatically set to 1 after the eMCU writes SFR_CTRL_RD, and SB_DATA_SW_MR is automatically cleared to 0 after the host reads SFR_CTRL_RD.

SB_DATA_MW_SR is automatically set to 1 when SFR_CTRL_WR is written by host, and SB_DATA_MW_SR is automatically cleared to 0 when SFR_CTRL_WR is read by eMCU.

SB_MST_CLK_GATE also controls the multiplexing of eMCU program ROM. When RB_MST_CLK_GATE is 0, it is dedicated to the system SRAM on the host side, with a total of 4 bytes, which can dynamically load new programs for eMCU. When RB_MST_CLK_GATE is 1, the program ROM dedicated to eMCU has 2K instructions. Bits 0 to 4 of the system configuration register are completely controlled by the host, and the eMCU side is read-only.

The following table is a description of the system configuration register SFR_SYS_CFG.

Address	Bit name	Description	Default value
Bit 7	SB_INT_REQ	Interrupt request activation bit: 0: Cancel interrupt request; 1: Request an interrupt from the host	0
Bit 6	SB_DATA_SW_MR	SFR_CTRL_RD waiting to read status bits: 0: EMCU has not been written or the host side has read; 1: eMCU has been written but the host side has not yet read it.	0
Bit 5	SB_DATA_MW_SR	SFR_CTRL_WR waiting to read status bit: 0: The host side has not written or the eMCU side has read; 1: The host side has written but the eMCU side has not read.	0
Bit 4	SB_MST_CFG_B4	Configuration information bit, software-defined purpose	0
Bit 3	SB_MST_IO_EN1	Mode and output control switch of IO1 pin: 0: Controlled by the host; 1: Controlled by eMCU	0
Bit 2	SB_MST_IO_EN0	Mode and output control switch of IO1 pin: 0: Controlled by the host; 1: Controlled by eMCU	0
Bit 1	SB_MST_RESET	Forced eMCU reset: 0: No forced reset, eMCU only resets with the host; 1: The host additionally forces the eMCU to reset	0

		independently.	
Bit 0	SB_MST_CLK_GATE	Global clock control of eMCU: 0: Turn off the clock of eMCU, and its program is suspended, which is equivalent to sleep; 1: Turn on the clock of eMCU	0

5.15 Encode Bit Period Register SFR_BIT_CYCLE

5.16 Code Bit Configuration Register SFR_BIT_CONFIG

EMCU supports 2 bit modulation modes: PWM duty cycle modulation and Manchester modulation.

Set SB_BIT_CYCLE greater than 3, and then start encoding and decoding. When SB_BIT_TX_EN=0, the input state is received from IO0 according to the preset width, and the decoded data is put in SB_BIT_RX_I0. Send when SB_BIT_TX_EN=1, put the original data to be sent in SB_BIT_TX_O0, and automatically send it to SB_PORT_OUT0 for pin output after encoding.

SB_BIT_CYC_CNT and SB_BIT_CYC_TAIL identify the periodic state of the coded bits. When SB_BIT_CYC_TAIL=1, it means that it is in the last 25% of the bit period, indicating that SB_BIT_RX_I0 has finished receiving and decoding, and the transmission coding has been completed and reached SB_PORT_OUT0.

Setting SB_BIT_CYCLE=0 can force SB_BIT_CYC_TAIL to be set to 1, which is convenient for Manchester to start the first codec.

Setting SFR_BIT_CYCLE will clear the detection records of rising and falling edges of IO0, which is convenient for WAITB instruction to wait again.

The following table is a description of the encoded bit period register SFR_BIT_CYCLE.

Address	Bit name	Description	Default value
Bit 7	SB_BIT_TX_O0	The original bit data of the coded bit to be sent, which has a double buffer structure. When a new bit period starts, this bit can load the next bit data in advance.	0
Bit 6 ~ Bit 0	SB_BIT_CYCLE	Sets the width of the coded bit in clock period, which is the actual clock number of the bit minus 1. Its high 5 bits are 0 to turn off the codec, otherwise it is turned on.	000 0000

The following table is a description of the encoding bit configuration register SFR_BIT_CONFIG.

Address	Bit name	Description	Default value
Bit 7	SB_BIT_TX_EN	Enable the transmission of coded bits: 0: Prohibit transmission, and receive if SB_BIT_CYCLE is valid; 1: Start sending, encode SB_BIT_TX_O0 and send it to SB_PORT_OUT0.	0

Bit 6	SB_BIT_CODE_MOD	Modulation mode of coded bits: 0: PWM duty ratio modulation, 0 duty ratio 25%, 1 duty ratio 75%; 1: Manchester modulation, 0 does not flip in the period, and 1 flips in the period.	0
Bit 5	SB_PORT_IN_EDGE	Pin input level sampling point selection: 0: sampling at the middle point of this clock cycle, which is relatively more real-time; 1: Sampling at the edge of the previous cycle is equivalent to about half a cycle in advance.	0
Bit 4	SB_BIT_CYC_TAIL	Periodic status of coded bits: 0: In the first 75% of the bit period; 1: It is in the last 25% of the bit period, indicating that the bit encoding and decoding has been completed.	1
Bit 3	SB_BIT_CYC_CNT6	The periodic timing state of the coded bits, in which the bit period timing is a 7-bit counter with the clock period as the unit, and SB_BIT_CYC_CNT6~3 is the bit 6 ~ 3 of the counter.	0
Bit 2	SB_BIT_CYC_CNT5		0
Bit 1	SB_BIT_CYC_CNT4		0
Bit 0	SB_BIT_CYC_CNT3		0

5.17 Data Register SFR_DATA_*

SFR_DATA_* are both data registers that can be read and written on both sides, and the host-side write operation takes precedence, and the software defines the purpose.
Among them, SFR_DATA_EXCH also supports one-cycle bit transfer instructions.

6. Application

EMCU has the same main frequency as the host side, which is convenient for dynamic data exchange, and its bit operation instructions are not rich. It sets I/O in a single cycle, collects I/O status or copies bit data in a single cycle, and can encode and decode the bit data modulated by PWM or Manchester in hardware. It is suitable for I/O implementation of various medium and low speed communication protocols and I/O control that requires precise timing.