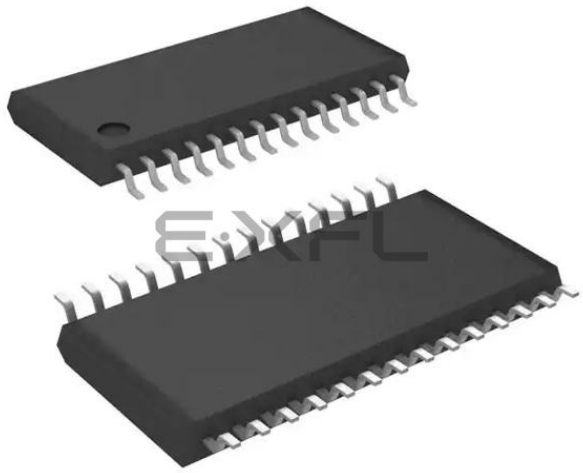

Welcome to E-XFL.COM

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	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, I²S, LCD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	18KB (18K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	A/D 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	28-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	28-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/n76e885at28

1. GENERAL DESCRIPTION

The N76E885 is an embedded flash type, 8-bit high performance 1T 8051-based microcontroller. The instruction set is fully compatible with the standard 80C51 and performance enhanced.

The N76E885 contains a up to 18K Bytes of main Flash called APROM, in which the contents of User Code resides. The N76E885 Flash supports In-Application-Programming (IAP) function, which enables on-chip firmware updates. IAP also makes it possible to configure any block of User Code array to be used as non-volatile data storage, which is written by IAP and read by IAP or MOVC instruction. There is an additional Flash called LDROM, in which the Boot Code normally resides for carrying out In-System-Programming (ISP). The LDROM size is configurable with a maximum of 4K Bytes. To facilitate programming and verification, the Flash allows to be programmed and read electronically by parallel Writer or In-Circuit-Programming (ICP). Once the code is confirmed, user can lock the code for security.

The N76E885 provides rich peripherals including 256 Bytes of SRAM, 256 Bytes of auxiliary RAM (XRAM), Up to 26 general purpose I/O, two 16-bit Timers/Counters 0/1, one 16-bit Timer2 with three-channel input capture module, one Watchdog Timer (WDT), one Self Wake-up Timer (WKT), one 16-bit auto-reload Timer3 for general purpose or baud rate generator, two UARTs with frame error detection and automatic address recognition, one SPI, one I^2C, four pairs of enhanced PWM output channels, eight-channel shared pin interrupt for all I/O, and one 10-bit ADC. The peripherals are equipped with 18 sources with 4-level-priority interrupts capability.

The N76E885 is equipped with five clock sources and supports switching on-the-fly via software. The four clock sources include 2 MHz to 25 MHz high-speed external crystal/resonator, 32.768 kHz low-speed external crystal/resonator, external clock input, 10 kHz internal oscillator, and one 22.118 MHz internal precise oscillator that is factory trimmed to ±1% at room temperature. The N76E885 provides additional power monitoring detection such as power-on reset and 8-level brown-out detection, which stabilizes the power-on/off sequence for a high reliability system design.

The N76E885 microcontroller operation consumes a very low power with two economic power modes to reduce power consumption – Idle and Power-down mode, which are software selectable. Idle mode turns off the CPU clock but allows continuing peripheral operation. Power-down mode stops the whole system clock for minimum power consumption. The system clock of the N76E885 can also be slowed down by software clock divider, which allows for a flexibility between execution performance and power consumption.

With high performance CPU core and rich well-designed peripherals, the N76E885 benefits to meet a general purpose, home appliances, or motor control system accomplishment.

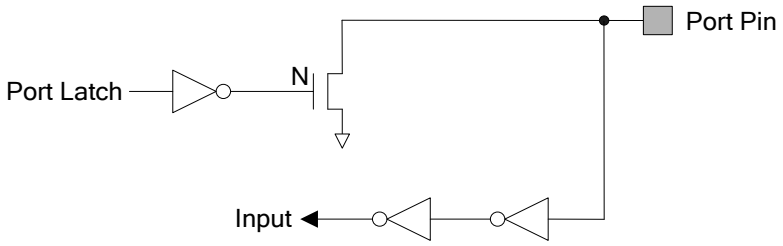

Figure 8-4. Open-Drain Mode Structure

8.5 Read-Modify-Write Instructions

Instructions that read a byte from SFR or internal RAM, modify it, and rewrite it back, are called “Read-Modify-Write” instructions. When the destination is an I/O port or a port bit, these instructions read the internal output latch rather than the external pin state. This kind of instructions read the port SFR value, modify it and write back to the port SFR. All “Read-Modify-Write” instructions are listed as follows.

Instruction		Description
ANL		Logical AND. (ANL direct, A and ANL direct, #data)
ORL		Logical OR. (ORL direct, A and ORL direct, #data)
XRL		Logical exclusive OR. (XRL direct, A and XRL direct, #data)
JBC		Jump if bit = 1 and clear it. (JBC bit, rel)
CPL		Complement bit. (CPL bit)
INC		Increment. (INC direct)
DEC		Decrement. (DEC direct)
DJNZ		Decrement and jump if not zero. (DJNZ direct, rel)
MOV	bit, C	Move carry to bit. (MOV bit, C)
CLR	bit	Clear bit. (CLR bit)
SETB	bit	Set bit. (SETB bit)

The last three seem not obviously “Read-Modify-Write” instructions but actually they are. They read the entire port latch value, modify the changed bit, and then write the new value back to the port latch.

8.6 Control Registers of I/O Ports

The N76E885 has a lot of I/O control registers to provide flexibility in all kinds of applications. The SFRs related with I/O ports can be categorized into four groups: input and output control, output mode control, input type and sink current control, and output slew rate control. All of SFRs are listed as follows.

TMOD – Timer 0 and 1 Mode

7	6	5	4	3	2	1	0
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: 89H Reset value: 0000 0000b

Bit	Name	Description
7	GATE	**Timer 1 gate control** 0 = Timer 1 will clock when TR1 is 1 regardless of $\overline{INT1}$ logic level. 1 = Timer 1 will clock only when TR1 is 1 and $\overline{INT1}$ is logic 1.
6	C/$\overline{T}$	**Timer 1 Counter/Timer select** 0 = Timer 1 is incremented by internal system clock. 1 = Timer 1 is incremented by the falling edge of the external pin T1.
5 4	M1 M0	**Timer 1 mode select** M1 M0 Timer 1 Mode 0 0 Mode 0: 13-bit Timer/Counter 0 1 Mode 1: 16-bit Timer/Counter 1 0 Mode 2: 8-bit Timer/Counter with auto-reload from TH1 1 1 Mode 3: Timer 1 halted
3	GATE	**Timer 0 gate control** 0 = Timer 0 will clock when TR0 is 1 regardless of $\overline{INT0}$ logic level. 1 = Timer 0 will clock only when TR0 is 1 and $\overline{INT0}$ is logic 1.
2	C/$\overline{T}$	**Timer 0 Counter/Timer select** 0 = Timer 0 is incremented by internal system clock. 1 = Timer 0 is incremented by the falling edge of the external pin T0.
1 0	M1 M0	**Timer 0 mode select** M1 M0 Timer 0 Mode 0 0 Mode 0: 13-bit Timer/Counter 0 1 Mode 1: 16-bit Timer/Counter 1 0 Mode 2: 8-bit Timer/Counter with auto-reload from TH0 1 1 Mode 3: TL0 as a 8-bit Timer/Counter and TH0 as a 8-bit Timer

TCON – Timer 0 and 1 Control (Bit-addressable)

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R (level) R/W (edge)	R/W	R (level) R/W (edge)	R/W

Address: 88H Reset value: 0000 0000b

Bit	Name	Description
7	TF1	**Timer 1 overflow flag** This bit is set when Timer 1 overflows. It is automatically cleared by hardware when the program executes the Timer 1 interrupt service routine. This bit can be set or cleared by software.
6	TR1	**Timer 1 run control** 0 = Timer 1 Disabled. Clearing this bit will halt Timer 1 and the current count will be preserved in TH1 and TL1. 1 = Timer 1 Enabled.

11. TIMER 3

Timer 3 is implemented simply as a 16-bit auto-reload, up-counting timer. The user can select the pre-scale with T3PS[2:0] (T3CON[2:0]) and fill the reload value into RH3 and RL3 registers to determine its overflow rate. User then can set TR3 (T3CON.3) to start counting. When the counter rolls over FFFFH, TF3 (T3CON.4) is set as 1 and a reload is generated and causes the contents of the RH3 and RL3 registers to be reloaded into the internal 16-bit counter. If ET3 (EIE1.1) is set as 1, Timer 3 interrupt service routine will be served. TF3 is auto-cleared by hardware after entering its interrupt service routine.

Timer 3 can also be the baud rate clock source of both UARTs. For details, please see Section 14.5 "Baud Rate" on page 70.

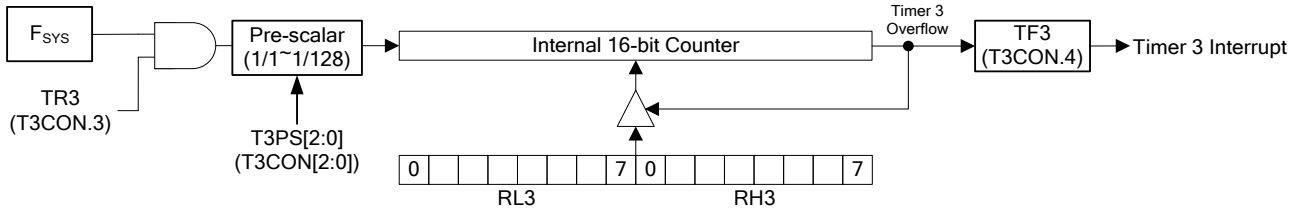

Figure 11-1. Timer 3 Block Diagram

T3CON – Timer 3 Control

7	6	5	4	3	2	1	0
SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W		

Address: C4H Reset value: 0000 0000b

Bit	Name	Description
4	TF3	**Timer 3 overflow flag** This bit is set when Timer 3 overflows. It is automatically cleared by hardware when the program executes the Timer 3 interrupt service routine. This bit can be set or cleared by software.
3	TR3	**Timer 3 run control** 0 = Timer 3 is halted. 1 = Timer 3 starts running. Note that the reload registers RH3 and RL3 can only be written when Timer 3 is halted (TR3 bit is 0). If any of RH3 or RL3 is written if TR3 is 1, result is unpredictable.

Bit	Name	Description
6	WDCLR	**WDT clear** Setting this bit will reset the WDT count to 00H. It puts the counter in a known state and prohibit the system from unpredictable reset. The meaning of writing and reading WDCLR bit is different. Writing: 0 = No effect. 1 = Clearing WDT counter. Reading: 0 = WDT counter is completely cleared. 1 = WDT counter is not yet cleared.
5	WDTF	**WDT time-out flag** This bit indicates an overflow of WDT counter. This flag should be cleared by software.
4	WIDPD	**WDT running in Idle or Power-down mode** This bit is valid only when control bits in WDTEN[3:0] (CONFIG4[7:4]) are all 1. It decides whether WDT runs in Idle or Power-down mode when WDT works as a general purpose timer. 0 = WDT stops running during Idle or Power-down mode. 1 = WDT keeps running during Idle or Power-down mode.
3	WDTRF	**WDT reset flag** When the CPU is reset by WDT time-out event, this bit will be set via hardware. This flag is recommended to be cleared via software after reset.
2:0	WDPS[2:0]	**WDT clock pre-scalar select** These bits determine the pre-scale of WDT clock from 1/1 through 1/256. See Table 12–1. The default is the maximum pre-scale value.

[1] WDTRF will be cleared after power-on reset, be set after WDT reset, and remains unchanged after any other resets.

[2] WDPS[2:0] are all set after power-on reset and keep unchanged after any reset other than power-on reset.

The Watchdog time-out interval is determined by the formula $\frac{1}{F_{LIRC} \times clockdividerscalar} \times 64$, where F_{LIRC} is the frequency of internal 10 kHz oscillator. The following table shows an example of the Watchdog time-out interval with different pre-scales.

T3CON – Timer 3 Control

7	6	5	4	3	2	1	0
SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W		

Address: C4H Reset value: 0000 0000b

Bit	Name	Description
5	BRCK	**Serial port 0 baud rate clock source** This bit selects which Timer is used as the baud rate clock source when serial port 0 is in Mode 1 or 3. 0 = Timer 1. 1 = Timer 3.

When using Timer 1 as the baud rate clock source, note that the Timer 1 interrupt should be disabled. Timer 1 itself can be configured for either “Timer” or “Counter” operation. It can be in any of its three running modes. However, in the most typical applications, it is configured for “Timer” operation, in the auto-reload mode (Mode 2). If using Timer 3 as the baud rate generator, its interrupt should also be disabled.

Table 14–3. UART Baud Rate Formulas

UART Mode	Baud Rate Clock Source	Baud Rate
0	System clock	$F_{SYS}/12$ or $F_{SYS}/2$ [1]
2	System clock	$F_{SYS}/64$ or $F_{SYS}/32$ [2]
1 or 3	Timer 1 (only for UART0)[3]	$\frac{2^{SMOD}}{32}\times\frac{F_{SYS}}{12\times(256-TH1)}$ or $\frac{2^{SMOD}}{32}\times\frac{F_{SYS}}{256-TH1}$ [4]
	Timer 3 (for UART0)	$\frac{2^{SMOD}}{32}\times\frac{F_{SYS}}{Pre-scale\times(65536-\{RH3,RL3\})}$ [5]
	Timer 3 (for UART1)	$\frac{1}{16}\times\frac{F_{SYS}}{Pre-scale\times(65536-\{RH3,RL3\})}$ [5]

[1] SM2 (SCON.5) or SM2_1(SCON_1.5) is set as logic 1.
[2] SMOD (PCON.7) or SMOD_1(T3CON.7) is set as logic 1.
[3] Timer 1 is configured as a timer in auto-reload mode (Mode 2).
[4] T1M (CKCON.4) is set as logic 1. While SMOD is 1, TH1 should not be FFH.
[5] {RH3,RL3} in the formula means $256\times RH3+RL3$. While SMOD is 1 and pre-scale is 1/1, {RH3,RL3} should not be FFFFH.

Table 14–4 lists various commonly used baud rates and how they can be obtained with Timer 1. In this mode, Timer 1 operates as an auto-reload Timer with SMOD (PCON.7) is 0 and T1M (CKCON.4) is 0. Table 14–5 is related to UART0 for Timer 3. This table illustrates that when SMOD is 0 the same setting doubles the baud rate for UART1.

Transmit data cannot be written to the shifter until the previous transfer is complete. Receiving logic consists of parallel read data buffer so the shift register is free to accept a second data, as the first received data will be transferred to the read data buffer.

The four pins of SPI interface are Master-In/Slave-Out (MISO), Master-Out/Slave-In (MOSI), Shift Clock (SPCLK), and Slave Select ($\overline{SS}$). The MOSI pin is used to transfer a 8-bit data in series from the Master to the Slave. Therefore, MOSI is an output pin for Master device and an input for Slave. Respectively, the MISO is used to receive a serial data from the Slave to the Master.

The SPCLK pin is the clock output in Master mode, but is the clock input in Slave mode. The shift clock is used to synchronize the data movement both in and out of the devices through their MOSI and MISO pins. The shift clock is driven by the Master mode device for eight clock cycles. Eight clocks exchange one byte data on the serial lines. For the shift clock is always produced out of the Master device, the system should never exist more than one device in Master mode for avoiding device conflict.

Each Slave peripheral is selected by one Slave Select pin ($\overline{SS}$). The signal should stay low for any Slave access. When $\overline{SS}$ is driven high, the Slave device will be inactivated. If the system is multi-slave, there should be only one Slave device selected at the same time. In the Master mode MCU, the $\overline{SS}$ pin does not function and it can be configured as a general purpose I/O. However, $\overline{SS}$ can be used as Master Mode Fault detection (see Section 15.5 “Mode Fault Detection” on page 86) via software setting if multi-master environment exists. The N76E885 also provides auto-activating function to toggle $\overline{SS}$ between each byte-transfer.

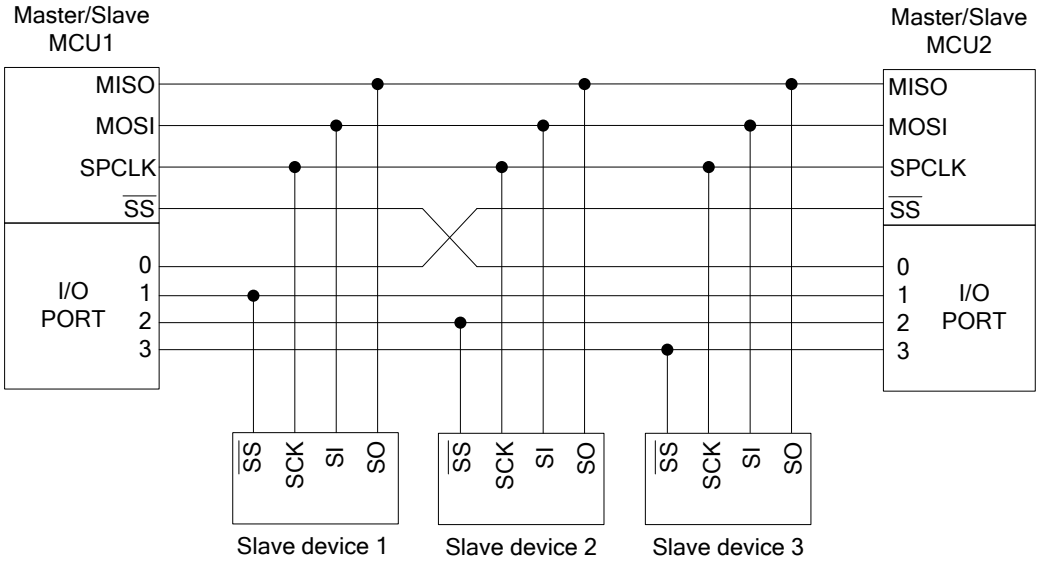

Figure 15-2. SPI Multi-Master, Multi-Slave Interconnection

continued by the winning master. However, the arbitration lost master continues generating clock pulses on SCL line until the end of the byte in which it loses the arbitration.

Arbitration is carried out by all masters continuously monitoring the SDA line after outputting data. If the value read from the SDA line does not match the value that the master has to output, it has lost the arbitration. Note that a master can only lose arbitration when it outputs a high SDA value while another master outputs a low value. Arbitration will continue until only one master remains, and this may take many bits. Its first stage is a comparison of address bits, and if both masters are trying to address the same device, arbitration continues on to the comparison of data bits or acknowledge bit.

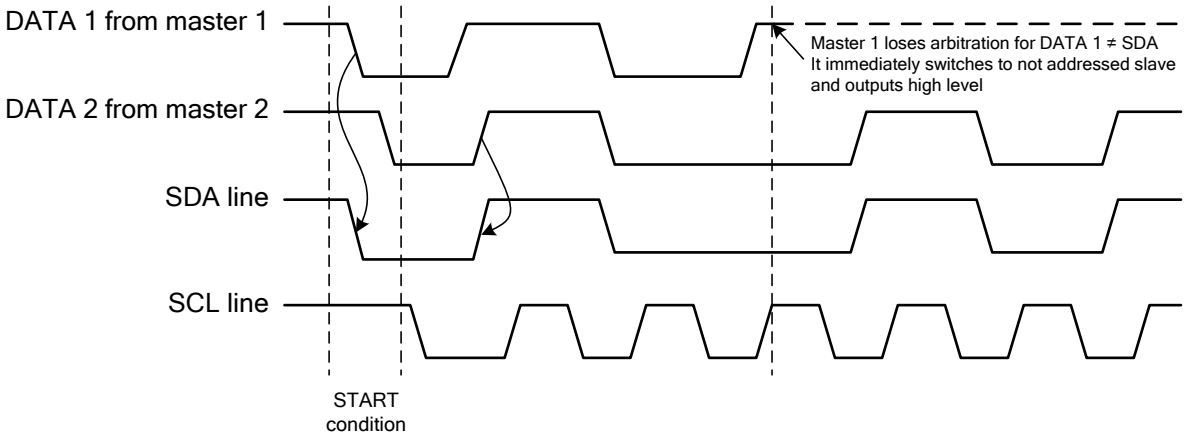

Figure 16-6. Arbitration Procedure of Two Masters

Since control of the I^{2}C bus is decided solely on the address or master code and data sent by competing masters, there is no central master, nor any order of priority on the bus.

Slaves are not involved in the arbitration procedure.

16.2 Control Registers of I^{2}C

There are five control registers to interface the I^{2}C bus including I2CON, I2STAT, I2DAT, I2ADDR, and I2CLK. These registers provide protocol control, status, data transmitting and receiving functions, and clock rate configuration. The following registers relate to I^{2}C function.

I2CON – I²C Control (Bit-addressable)

7	6	5	4	3	2	1	0
-	I2CEN	STA	STO	SI	AA	-	-
-	R/W	R/W	R/W	R/W	R/W	-	-

Address: C0H Reset value: 0000 0000b

Bit	Name	Description
7	-	**Reserved**
6	I2CEN	**I²C bus enable** 0 = I²C bus Disabled. 1 = I²C bus Enabled. Before enabling the I²C, P2.3 and P0.6 port latches should be set to logic 1.
5	STA	**START flag** When STA is set, the I²C generates a START condition if the bus is free. If the bus is busy, the I²C waits for a STOP condition and generates a START condition following. If STA is set while the I²C is already in the master mode and one or more bytes have been transmitted or received, the I²C generates a repeated START condition. Note that STA can be set anytime even in a slave mode, but STA is not hardware automatically cleared after START or repeated START condition has been detected. User should take care of it by clearing STA manually.
4	STO	**STOP flag** When STO is set if the I²C is in the master mode, a STOP condition is transmitted to the bus. STO is automatically cleared by hardware once the STOP condition has been detected on the bus. The STO flag setting is also used to recover the I²C device from the bus error state (I2STAT as 00H). In this case, no STOP condition is transmitted to the I²C bus. If the STA and STO bits are both set and the device is original in the master mode, the I²C bus will generate a STOP condition and immediately follow a START condition. If the device is in slave mode, STA and STO simultaneous setting should be avoid from issuing illegal I²C frames.
3	SI	**I²C interrupt flag** SI flag is set by hardware when one of 26 possible I²C status (besides F8H status) is entered. After SI is set, the software should read I2STAT register to determine which step has been passed and take actions for next step. SI is cleared by software. Before the SI is cleared, the low period of SCL line is stretched. The transaction is suspended. It is useful for the slave device to deal with previous data bytes until ready for receiving the next byte. The serial transaction is suspended until SI is cleared by software. After SI is cleared, I²C bus will continue to generate START or repeated START condition, STOP condition, 8-bit data, or so on depending on the software configuration of controlling byte or bits. Therefore, user should take care of it by preparing suitable setting of registers before SI is software cleared.

PWM45L – PWM4/5 Duty Low Byte

7	6	5	4	3	2	1	0
PWM45[7:0]							
R/W							

Address: DDH reset value: 0000 0000b

Bit	Name	Description
7:0	PWM45[7:0]	**PWM4/5 duty low byte** This byte with PWM45H controls the duty of the output signal PG45 from PWM generator.

PWM45H – PWM4/5 Duty High Byte

7	6	5	4	3	2	1	0
-	-	-	-	PWM45[11:8]			
-	-	-	-	R/W			

Address: D5H reset value: 0000 0000b

Bit	Name	Description
3:0	PWM45[11:8]	**PWM4/5 duty high byte** This byte with PWM45L controls the duty of the output signal PG45 from PWM generator.

PWM67L – PWM6/7 Duty Low Byte

7	6	5	4	3	2	1	0
PWM67[7:0]							
R/W							

Address: DCH reset value: 0000 0000b

Bit	Name	Description
7:0	PWM67[7:0]	**PWM6/7 duty low byte** This byte with PWM67H controls the duty of the output signal PG67 from PWM generator.

PWM67H – PWM6/7 Duty High Byte

7	6	5	4	3	2	1	0
-	-	-	-	PWM67[11:8]			
-	-	-	-	R/W			

Address: D4H reset value: 0000 0000b

Bit	Name	Description
3:0	PWM67[11:8]	**PWM6/7 duty high byte** This byte with PWM67L controls the duty of the output signal PG67 from PWM generator.

18.1.3.1 Independent Mode

Independent mode is enabled when PWMMOD[1:0] (PWMCON1[7:6]) is [0:0]. It is the default mode of PWM. PG0/2/4/6 output PWM signals and PG1/3/5/7 remains high state level.

18.1.3.2 Complementary Mode with Dead-Time Insertion

Complementary mode is enabled when PWMMOD[1:0] = [0:1]. In this mode, PG0/2/4/6 output PWM signals the same as the independent mode. However, PG1/3/5/7 output the out-phase PWM signals of PG0/2/4/6 correspondingly. This mode makes PG0/PG1 a PWM complementary pair and so on PG2/PG3, PG4/PG5, andPG6/PG7.

In a real motor application, a complementary PWM output always has a need of “dead-time” insertion to prevent damage of the power switching device like GPIBs due to being active on simultaneously of the upper and lower switches of the half bridge, even in a “μs” duration. For a power switch device physically cannot switch on/off instantly. For the N76E885 PWM, each PWM pair share a 9-bit dead-time down-counter PDTCNT used to produce the off time between two PWM signals in the same pair. On implementation, a 0-to-1 signal edge delays after PDTCNT timer underflows. The timing diagram illustrates the complementary mode with dead-time insertion of PG0/PG1 pair. Pairs of PG2/PG3, PG4/PG5, and PG6/PG7 have the same dead-time circuit. Each pair has its own dead-time enabling bit in the field of PDTEN[3:0].

Note that the PDTCNT and PDTEN registers are all TA write protection. The dead-time control are also valid only when the PWM is configured in its complementary mode.

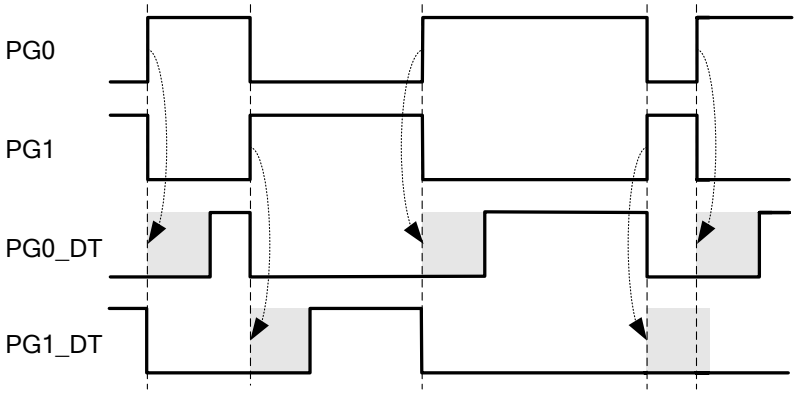

Figure 18-5. PWM Complementary Mode with Dead-time Insertion

Bit	Name	Description
1:0	INTSEL[1:0]	**PWM interrupt pair select** These bits select which PWM channel asserts PWM interrupt when PWM interrupt type is selected as falling or rising edge on PWM0/2/4 pin.. 00 = PWM0. 01 = PWM2. 10 = PWM4. 11 = PWM6.

The PWM interrupt related with PWM waveform is shown as figure below.

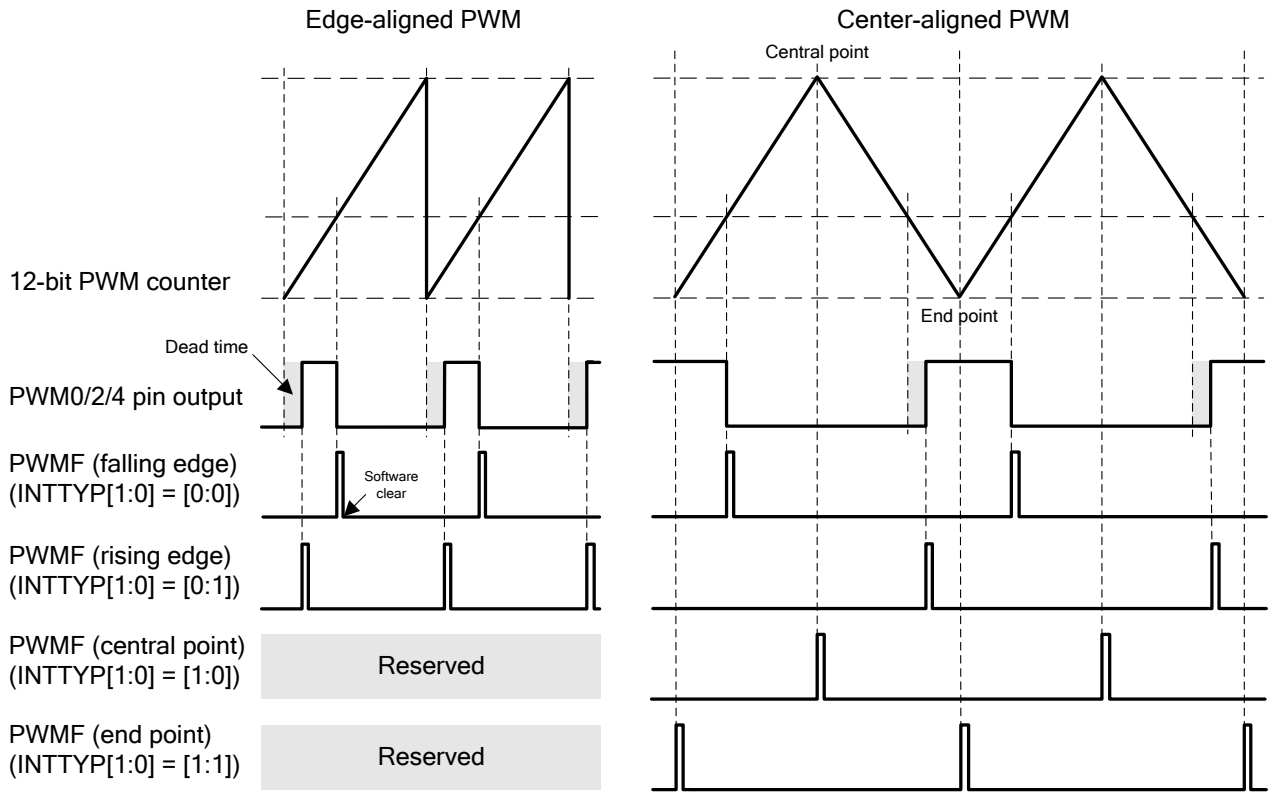

Figure 18-7. PWM Interrupt Type

Fault Brake event requests another interrupt, Fault Brake interrupt. It has different interrupt vector from PWM interrupt. When either Fault Brake pin input event or ADC compare event occurs, FBF (FBD.7) will be set by hardware. It generates Fault Brake interrupt if enabled. The Fault Brake interrupt enable bit is EFB (EIE.5). FBF Is cleared via software.

ADC circuit saves power. The VREF voltage input source can be selected from the internal V_{DD} or the external AIN0/VREF pin by VREFSEL (ADCCON1.7) bit.

The ADC analog input pin should be specially considered. ADCHS[3:0] are channel selection bits that control which channel is connected to the sample and hold circuit. User needs to configure selected ADC input pins as input-only (high impedance) mode via respective bits in PxMn registers. This configuration disconnects the digital output circuit of each selected ADC input pin. But the digital input circuit still works. Digital input may cause the input buffer to induce leakage current. To disable the digital input buffer, the respective bits in P0DIDS and ADCCON2 should be set. Configuration above makes selected ADC analog input pins pure analog inputs to allow external feeding of the analog voltage signals. Also, the ADC clock rate needs to be considered carefully. The ADC maximum clock frequency is listed in Table 32–10. Clock above the maximum clock frequency degrades ADC performance unpredictably.

An A/D conversion is initiated by setting the ADCS bit (ADCCON0.6). When the conversion is complete, the hardware will clear ADCS automatically, set ADCF (ADCCON0.7) and generate an interrupt if enabled. The new conversion result will also be stored in ADCRH (most significant 8 bits) and ADCRL (least significant 2 bits). The 10-bit ADC result value is $1023 \times \frac{V_{AIN}}{V_{REF}}$.

The ADC acquisition time is programmable, which provides a range of 6 (6 + 0) to 261 (6 + 255) ADC clock cycles, by configuring ADCAQT register. It is useful to preserve the accuracy of ADC result especially when the input impedance of the analog input source is not ideally low. The programmable acquisition time overcomes the high impedance of an analog input source.

By the way, digital circuitry inside and outside the device generates noise which might affect the accuracy of ADC measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. Keep analog signal paths as short as possible. Make sure to run analog signals tracks well away from high-speed digital tracks.

2. Place the device in Idle mode during a conversion.

3. If any AIN pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.

19.1.2 ADC Conversion Triggered by External Source

Besides setting ADCS via software, the N76E885 is enhanced by supporting hardware triggering method to start an A/D conversion. If ADCEX (ADCCON1.1) is set, edges or period points on selected

Bit	Name	Description
6	ESPI	**Enable SPI interrupt** 0 = SPI interrupt Disabled. 1 = Interrupt generated by SPIF (SPSR.7), SPIOVF (SPSR.5), or MODF (SPSR.4) Enable.
5	EFB	**Enable Fault Brake interrupt** 0 = Fault Brake interrupt Disabled. 1 = Interrupt generated by FBF (FBD.7) Enabled.
4	EWDT	**Enable WDT interrupt** 0 = WDT interrupt Disabled. 1 = Interrupt generated by WDTF (WDCON.5) Enabled.
3	EPWM	**Enable PWM interrupt** 0 = PWM interrupt Disabled. 1 = Interrupt generated by PWMF (PWMCON0.5) Enabled.
2	ECAP	**Enable input capture interrupt** 0 = Input capture interrupt Disabled. 1 = Interrupt generated by any flags of CAPF[2:0] (CAPCON0[2:0]) Enabled.
1	EPI	**Enable pin interrupt** 0 = Pin interrupt Disabled. 1 = Interrupt generated by any flags in PIF register Enabled.
0	EI2C	**Enable I^2C interrupt** 0 = I^2C interrupt Disabled. 1 = Interrupt generated by SI (I2CON.3) or I2TOF (I2TOC.0) Enabled.

EIE1 – Extensive Interrupt Enable 1

7	6	5	4	3	2	1	0
-	-	-	-	-	EWKT	ET3	ES_1
-	-	-	-	-	R/W	R/W	R/W

Address: 9CH Reset value: 0000 0000b

Bit	Name	Description
2	EWKT	**Enable WKT interrupt** 0 = WKT interrupt Disabled. 1 = Interrupt generated by WKTF (WKCON.4) Enabled.
1	ET3	**Enable Timer 3 interrupt** 0 = Timer 3 interrupt Disabled. 1 = Interrupt generated by TF3 (T3CON.4) Enabled.
0	ES_1	**Enable serial port 1 interrupt** 0 = Serial port 1 interrupt Disabled. 1 = Interrupt generated by TI_1 (SCON_1.1) or RI_1 (SCON_1.0) Enabled.

21.3 Interrupt Priorities

There are four priority levels for all interrupts. They are level highest, high, low, and lowest; and they are represented by level 3, level 2, level 1, and level 0. The interrupt sources can be individually set to one of four priority levels by setting their own priority bits. Table 21–2 lists four priority setting. Naturally, a low level priority interrupt can itself be interrupted by a high level priority interrupt, but not by any same level interrupt or lower level. In addition, there exists a pre-defined natural priority among

Bit	Name	Description
		condition is met: (1) The accessing address is oversize. (2) IAPCN command is invalid. (3) IAP erases or programs updating un-enabled block. (4) IAP erasing or programming operates under V_{BOD} while BOIAP (CONFIG2.5) remains un-programmed 1 with BODEN (BODCON0.7) as 1 and BORST (BODCON0.2) as 0. This bit should be cleared via software.
0	IAPEN	**IAP enable** 0 = IAP function Disabled. 1 = IAP function Enabled. Once enabling IAP function, the HIRC will be turned on for timing control. To clear IAPEN should always be the last instruction after IAP operation to stop internal oscillator if reducing power consumption is concerned.

IAPUEN – IAP Updating Enable (TA protected)

7	6	5	4	3	2	1	0
-	-	-	-	-	CFUEN	LDUEN	APUEN
-	-	-	-	-	R/W	R/W	R/W

Address: A5H Reset value: 0000 0000b

Bit	Name	Description
2	CFUEN	**CONFIG bytes updated enable** 0 = Inhibit erasing or programming CONFIG bytes by IAP. 1 = Allow erasing or programming CONFIG bytes by IAP.
1	LDUEN	**LDROM updated enable** 0 = Inhibit erasing or programming LDROM by IAP. 1 = Allow erasing or programming LDROM by IAP.
0	APUEN	**APROM updated enable** 0 = Inhibit erasing or programming APROM by IAP. 1 = Allow erasing or programming APROM by IAP.

IAPCN – IAP Control

7	6	5	4	3	2	1	0
IAPB[1:0]		FOEN	FCEN	FCTRL[3:0]			
R/W		R/W	R/W	R/W			

Address: AFH Reset value: 0011 0000b

Bit	Name	Description
7:6	IAPB[1:0]	**IAP control** This byte is used for IAP command. For details, see Table 22–1. IAP Modes and Command Codes.
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

CHPCON – Chip Control (TA protected)

7	6	5	4	3	2	1	0
SWRST	IAPFF	-	-	-	-	BS	IAPEN
W	R/W	-	-	-	-	R/W	R/W

Address: 9FH Reset value: see Table 6–2. SFR Definitions and Reset Values

Bit	Name	Description
7	SWRST	**Software reset** To set this bit as logic 1 will cause a software reset. It will automatically be cleared via hardware after reset is finished.

AUXR1 – Auxiliary Register 1

7	6	5	4	3	2	1	0
SWRF	RSTPINF	T1LXTM	T0LXTM	GF2	UART0PX	0	DPS
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Address: A2H Reset value: see Table 6–2. SFR Definitions and Reset Values

Bit	Name	Description
7	SWRF	**Software reset flag** When the MCU is reset via software reset, this bit will be set via hardware. It is recommended that the flag be cleared via software.

The software demo code is listed below.

```
ANL    AUXR1,#01111111b    ;software reset flag clear
...
...
CLR    EA
MOV    TA,#0AAh
MOV    TA,#55h
ORL    CHPCON,#10000000b   ;software reset
```

26.6 Boot Select

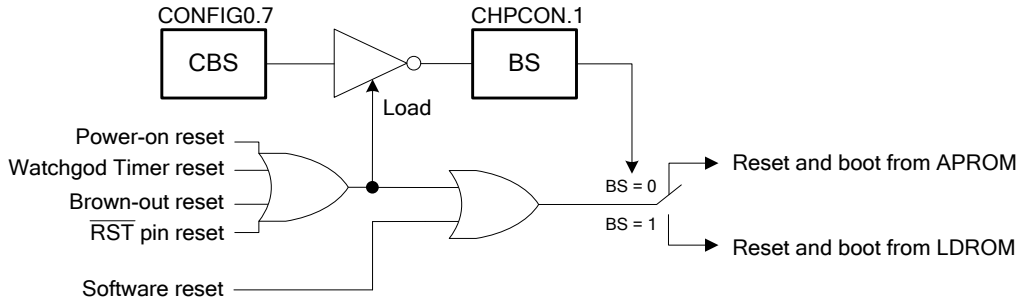

Figure 26-1. Boot Selecting Diagram

The N76E885 provides user a flexible boot selection for variant application. The SFR bit BS in CHPCON.1 determines MCU booting from APROM or LDROM after any source of reset. If reset occurs and BS is 0, MCU will reboot from address 0000H of APROM. Else, the CPU will reboot from

30. IN-CIRCUIT-PROGRAMMING (ICP)

The Flash Memory can be programmed by “In-Circuit-Programming” (ICP). In general, hardware programming mode uses gang-writers to reduce programming costs and time to market while the products enter the mass production state. However, if the product is just under development or the end product needs firmware updating in the hand of an end customer, the hardware programming mode will make repeated programming difficult and inconvenient. ICP method makes it easy and possible without removing the microcontroller from the system. ICP mode also allows customers to manufacture circuit boards with un-programmed devices. Programming can be done after the assembly process allowing the device to be programmed with the most recent firmware or a customized firmware.

There are three signal pins, $\overline{\text{RST}}$, ICPDA, and ICPCK, involved in ICP function. $\overline{\text{RST}}$ is used to enter or exit ICP mode. ICPDA is the data input and output pin. ICPCK is the clock input pin, which synchronizes the data shifted in to or out from MCU under programming. User should leave these three pins plus VDD and GND pins on the circuit board to make ICP possible.

Nuvoton provides ICP tool for N76E885, which enables user to easily perform ICP through Nuvoton ICP programmer. The ICP programmer developed by Nuvoton has been optimized according to the electric characteristics of MCU. It also satisfies the stability and efficiency during production progress. For more details, please visit Nuvoton 8-bit Microcontroller website: Nuvoton 80C51 Microcontroller Technical Support.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
TUE	Total un-adjust error		-	+3.5	+4	LSB
-	Monotonicity		Guaranteed			-
F_{ADC}	ADC clock frequency	V_{VREF} = 3.0V to 5.5V	0.01	-	6	MHz
		V_{VREF} = 2.4V to 5.5V	0.01	-	3	
T_S	Sampling time (software adjust)		6	-	261	$1/F_{ADC}$
T_{CONV}	Total conversion time		T_S + 12			$1/F_{ADC}$
T_{ADCEN}	ADC enable time		32			$1/F_{ADC}$
R_{IN}	ADC input equivalent resistor		-	-	7	kΩ
C_{IN}	ADC input equivalent capacitor			10	12	pF

34. DOCUMENT REVISION HISTORY

Revision	Date	Description
1.00	2015/2/26	Initial release.
1.01	2015/12/21	Modify ECLK 1.8v Domain Modify data retention guarantee years Modify TSSOP 28 package dimension Y value Modify RAM access description VIL and VIL1 description modify Remove AUXR Modify band-gap description with BODEN

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.