

CMS89F52x

Benutzerhandbuch

Verbessertes Flash 8 -Bit

CMOS -MCU

V1.3

Bitte beachten Sie Folgendes in Bezug auf die CMS-Richtlinie zum geistigen Eigentum

China Micro Semiconductor hat ein Patent angemeldet und genießt absolute gesetzliche Rechte. Die Verwendung der Patentrechte in Bezug auf die MCU oder andere Produkte der China Micro Semiconductor Company wurde nicht autorisiert. Jedes Unternehmen, jede Organisation oder Einzelperson, die die Patentrechte der China Micro Semiconductor Company auf unangemessene Weise verletzt, wird die China Micro Semiconductor Company alle Rechte übernehmen. Mögliche rechtliche Schritte Stoppen Sie die unangemessene Verletzung des Rechtsverletzers und erstatten Sie die Verluste, die AMEC aufgrund der Rechtsverletzung erlitten hat, oder die rechtswidrigen Vorteile, die der Rechtsverletzer erhalten hat.

Der Name und das Logo von China Microelectronics sind eingetragene Warenzeichen der China Micro Semiconductor Corporation.

China Micro-Semiconductor behält sich das Recht vor, die Verbesserung des Produkts in der Spezifikation in Bezug auf Zuverlässigkeit, Funktion und Design weiter zu erläutern. AMEC ist jedoch nicht verantwortlich für die Verwendung des Inhalts der Spezifikation. Die in diesem Artikel erwähnten Anwendungen dienen lediglich der Veranschaulichung. AMEC übernimmt keine Garantie und gibt keine Zusicherung, dass diese Anwendungen ohne weitere Modifikationen anwendbar sind, und empfiehlt auch nicht, dass seine Produkte in Anwendungen verwendet werden, die aufgrund von Fehlern oder anderen Gründen ausfallen könnten. Ein Ort, der eine persönliche Gefahr darstellt. AMEC-Produkte dürfen nicht als Schlüsselkomponenten in lebensrettenden und lebenserhaltenden Geräten oder Systemen verwendet werden. MCU behält sich das Recht vor, Produkte ohne vorherige Ankündigung zu ändern. Die neuesten Informationen finden Sie auf unserer Website <http://www.mcu.com.cn>

1. Produktübersicht	1
1.1 Merkmale.....	1
1.2 Blockschaltbild der Systemstruktur.....	2
1.3 Pinbelegung	3
1.3.1 Pinbelegung CMS89F5231.....	3
1.3.2 Pinbelegung CMS89F5232.....	4
1.3.3 Pinbelegung CMS89F5233.....	5
1.3.4 CMS89F526-Pin-Diagramm.....	6
1.4 Systemkonfigurationsregister.....	7
1.5 Serielle In-Circuit-Programmierung	8
2. Zentraleinheit (CPU)	9
2.1 Speicher.....	9
2.1.1 Programmspeicher.....	9
2.1.2 Datenspeicher.....	14
2.2 Adressierungsmodus.....	19
2.2.1 Direkte Adressierung.....	19
2.2.2 Sofortige Adressierung.....	19
2.2.3 Indirekte Adressierung.....	19
2.3 stapeln.....	20
2.4 Arbeitsregister (ACC)	einundzwanzig
2.4.1 Übersicht.....	einundzwanzig
2.4.2 ACC-Anwendung.....	einundzwanzig
2.5 Programmstatusregister (STATUS)	zweiundzwanzig
2.6 Prescaler (OPTION_REG).....	dreiundzwanzig
2.7 Programmzähler (PC)	25
2.8 Watchdog-Zähler (WDT)	26
2.8.1 WDT-Periode.....	26
2.8.2 Watchdog-Timer-Steuerregister WDTCON	26
3. Systemuhr	27
3.1 Übersicht.....	27
3.2 Systemoszillator	28
3.2.1 Interner RC-Oszillator.....	28
3.3 Startzeit.....	28
3.4 Oszillator-Steuerregister	29
4. Zurücksetzen	30
4.1 Power-On-Reset	30
4.2 Brownout-Reset	31
4.2.1 Übersicht zum Brownout-Reset.....	31

4.2.2	Verbesserungsmethode des Power-Down-Resets	32
4.3	Watchdog-Reset.....	32
5.	Ruhemodus	33
5.1	Wechseln in den Schlafmodus.....	33
5.2	Aufwachen aus dem Winterschlaf	33
5.3	Aufwecken mit Interrupt.....	33
5.4	Anwendungsbeispiele für den Sleep-Modus.....	34
5.5	Weckzeit für Schlafmodus.....	34
6.	E/A -Anschlüsse.....	35
6.1	E/A-Port-Strukturdiagramm.....	36
6.2	PORTA	37
6.2.1	PORTA-Daten und Richtungssteuerung.....	37
6.2.2	PORTA-Analog-Select-Regler.....	38
6.2.3	PORTA-Pull-up-Widerstand.....	38
6.3	PORTB.....	39
6.3.1	PORTB-Daten und -Richtung.....	39
6.3.2	PORTB-Analogauswahl.....	40
6.3.3	PORTB-Pullup-Widerstand.....	40
6.4	PORTE.....	41
6.4.1	PORTE-Daten und -Richtung.....	41
6.4.2	PORTE-Pull-Up-Widerstand.....	41
6.4.3	PORTE-ANALOG-AUSWAHL.....	42
6.5	E/A-Nutzung	43
6.5.1	E/A-Port schreiben.....	43
6.5.2	I/O-Port lesen.....	43
6.6	Vorsichtsmaßnahmen für die Verwendung von E/A-Anschlüssen.....	44
7.	Unterbrechung.....	45
7.1	Unterbrechungsübersicht.....	45
7.2	Unterbrechungssteuerregister	46
7.2.1	Interrupt Control Register.....	46
7.2.2	Peripheral Interrupt Enable Register.....	47
7.2.3	Peripheral Interrupt Request Register.....	49
7.3	Methoden zum Schutz von Unterbrechungsstellen	51
7.4	Priorität von Interrupts und Verschachtelung mehrerer Interrupts.....	51
8.	Zeitähler TIMERO	52
8.1	Übersicht Timerzähler TIMERO.....	52
8.2	Funktionsweise von TIMERO.....	53
8.2.1	8-Bit-Timer-Modus	53
8.2.2	8-Bit-Zählermodus.....	53

8.2.3 Softwareprogrammierbarer Prescaler	53
8.2.4 UMSCHALTUNG DES VORTEILERS ZWISCHEN TIMER0- UND WDT-MODULEN	53
8.2.5 TIMER0-Unterbrechung.....	54
8.3 Register im Zusammenhang mit TIMER0.....	55
9. Zeitzähler TIMER1	56
9.1 Übersicht TIMER1.....	56
9.2 Funktionsweise von TIMER1.....	57
9.3 TIMER1 Prescaler.....	57
9.4 TIMER1-Unterbrechung.....	57
9.5 TIMER1-bezogene Register.....	58
10. Zeitzähler TIMER2	59
10.1 Übersicht TIMER2.....	59
10.2 Funktionsweise von TIMER2.....	60
10.3 TIMER2-bezogene Register.....	61
11. Analog-Digital-Umwandlung (ADC)	62
11.1 ADC-Übersicht.....	62
11.2 ADC-Konfiguration.....	63
11.2.1 Portkonfiguration.....	63
11.2.2 Kanalauswahl.....	63
11.2.3 ADC-Referenzspannung.....	63
11.2.4 Schaltuhren.....	63
11.2.5 ADC-Unterbrechung.....	64
11.3 ADC-Arbeitsprinzip.....	65
11.3.1 Boot-Übergang.....	65
11.3.2 Abschluss der Konvertierung.....	65
11.3.3 Konvertierung abbrechen.....	65
11.3.4 Funktionsweise von ADC im Sleep-Modus.....	65
11.3.5 A/D-Umwandlungsverfahren.....	66
11.4 ADC-bezogener RAM.....	67
12. PWM -Modul.....	69
12.1 PWM-Eigenschaften.....	69
12.2 PWM-bezogene Register	69
13. Erfassungsmodul CCP.....	72
13.1 Erfassung des CCP-Registers	72
13.2 Aufnahmemodus.....	73
13.2.1 CCP-Pin-Konfiguration.....	73
13.2.2 Auswahl des TIMER1-Modus.....	73
13.2.3 Software-Interrupts.....	73
14. MSSP-Modul (Master Synchronous Serial Port)	74

14.1 Übersicht über das Main Control SSP (MSSP) Modul.....	74
14.2 SPI-Modus.....	74
14.2.1 SPI-bezogene Register.....	75
14.2.2 SPI-Arbeitsprinzip.....	77
14.2.3 Aktivieren von SPI I/O	78
14.2.4 Mastermodus.....	78
14.2.5 Slave-Modus.....	79
14.2.6 Slave-Select-Synchronisation.....	80
14.2.7 Ruhezustand	82
14.2.8 Auswirkungen des Zurücksetzens.....	82
14.3 I2C-Modul.....	83
14.3.1 Beschreibung zugehöriger Register.....	84
14.3.2 Mastermodus.....	87
14.3.3 Unterstützung des I2C-Mastermodus.....	87
14.3.4 Baudratengenerator.....	89
14.3.5 I2C-Mastermode-Übertragung.....	90
14.3.6 I2C-Mastermodus-Empfang.....	91
14.3.7 I2C-Master-Modus-Startbedingungs-Timing.....	93
14.3.8 Zeitsteuerung für wiederholte Startbedingungen im I2C-Master-Modus	94
14.3.9 Bestätigungssequenz-Timing.....	95
14.3.10 Abfolge der Stoppbedingungen.....	96
14.3.11 Clock-Arbitrierung	97
14.3.12 Multi-Host-Modus.....	97
14.3.13 Multi-Master-Kommunikation, Buskonflikt und Bus-Arbitrierung	98
14.3.14 Slave-Modus.....	98
14.3.15 SSP-Maskenregister.....	101
14.3.16 Betrieb im Schlafmodus.....	101
14.3.17 Auswirkungen des Resets	101
15. Programmierbarer Impulsgenerator PPG	102
15.1 Wie PPG funktioniert.....	102
15.2 PPG-bezogene Pins.....	103
15.3 PPG-Arbeitsmodus.....	104
15.3.1 Einzelausgangsmodus.....	105
15.3.2 Synchroner Ausgabemodus.....	105
15.4 Komparator.....	106
15.4.1 Synchronkomparator COMP1.....	106
15.4.2 Überspannungskomparator COMP2 und Surge-Komparator COMP4/COMP5	108
15.4.3 Überspannungskomparator 1- COMP3	111
15.4.4 Komparatornullung.....	113
15.4.5 Komparator- und PPG-internes Strukturdiagramm.....	116

16. Daten- EEPROM - Steuerung	117
16.1 Übersicht Daten-EEPROM.....	117
16.2 Zugehörige Register.....	118
16.2.1 EEADR-Register.....	118
16.2.2 EECON1- und EECON2-Register.....	118
16.3 Daten EEPROM-Speicher auslesen.....	120
16.4 Schreiben von Daten in den EEPROM-Speicher.....	121
16.5 Vorsichtsmaßnahmen für den Umgang mit EEPROM.....	122
16.5.1 Schreibbestätigung.....	122
16.5.2 Schutz vor versehentlichem Schreiben.....	122
17. Operationsverstärker.....	123
17.1 Einführung in Operationsverstärker.....	123
17.2 Register im Zusammenhang mit Operationsverstärkern.....	124
18. Elektrische Parameter	126
18.1 DC-Eigenschaften.....	126
18.2 AC-Kennlinien.....	126
18.3 Eigenschaften des internen RC-Oszillators.....	127
18.3.1 Spannungscharakteristik des internen RC-Oszillators.....	127
18.3.2 Temperaturverhalten des internen RC-Oszillators.....	127
19. Anweisungen.....	128
19.1 Befehlsliste	128
19.2 Befehlsbeschreibung.....	130
20. Paket.....	146
20.1 DIP16	146
20.2 SOP16.....	147
20.3 DIP20	148
20.4 SOP20.....	149
21. Hinweise zur Versionsrevision	150

1. Produktübersicht

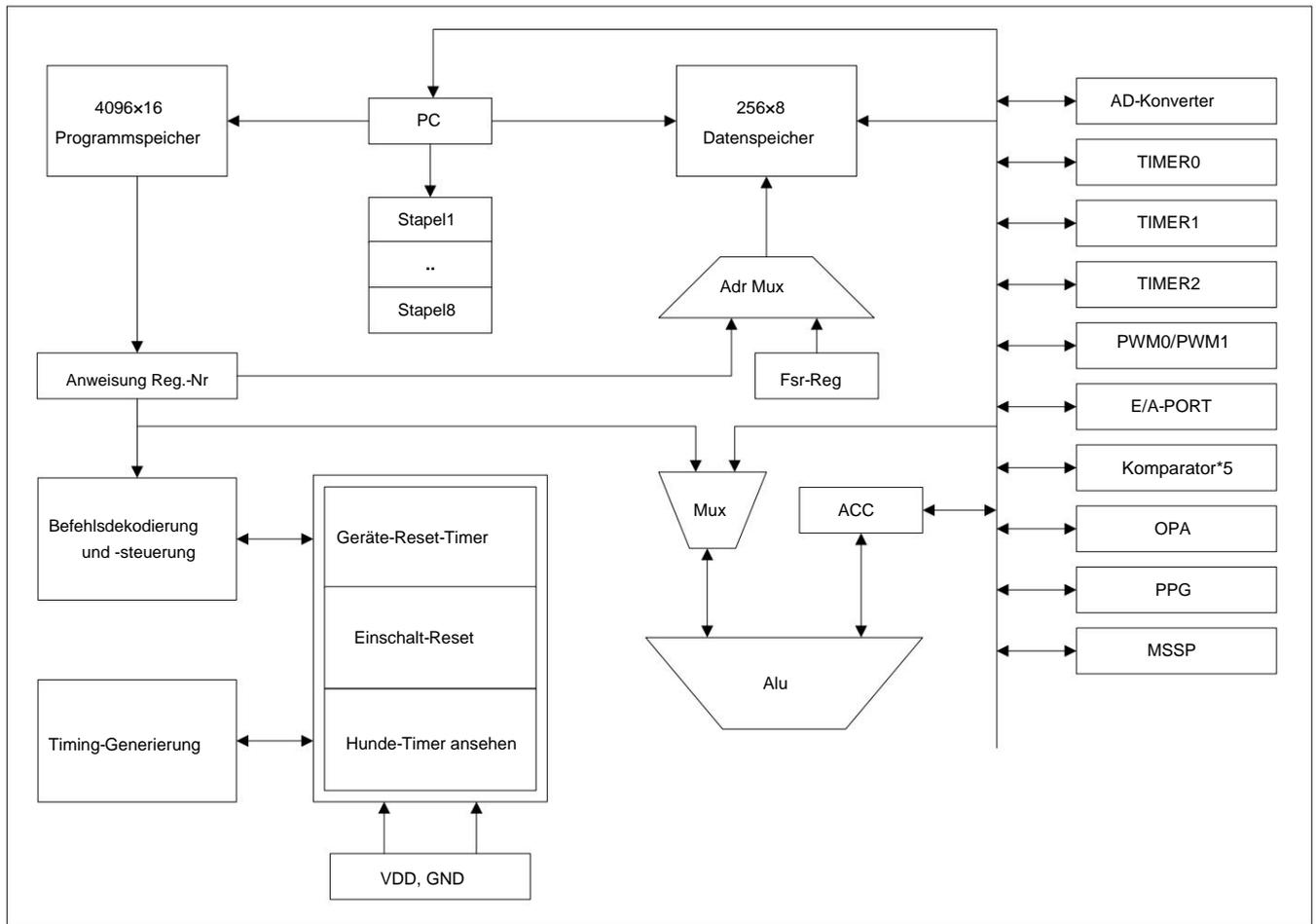
1.1 Funktionen

- Speicher
 - BLITZ: 4Kx16Bit
 - Allgemeiner Arbeitsspeicher: 256x8Bit
 - EEPROM: 32x16Bit
- Arbeitsspannungsbereich: 3,5 V~5,5 V bei 8 MHz 3,5 V~5,5 V bei 4 MHz
- Arbeitstemperaturbereich: -40~85°
- Oszillationsmodus
 - Interner RC-Oszillator: Designfrequenz 8MHz/16MHz
- Stapelpuffer mit 8 Ebenen
- Einfaches und praktisches Befehlssystem (68 Befehle) • Befehlszyklus (Einzelbefehl oder Doppelbefehl)
- Eingebauter Niederspannungs-Erkennungsschaltkreis
- Eingebauter WDT-Timer •
- Unterbrechungsquelle:
 - 3 Timer-Interrupts,
 - externe Interrupts,
 - andere periphere Interrupts
- Hochpräziser 10-Bit-ADC • MSSP-Kommunikationsmodul (SPI/I 2C)
- PPG-Steuermodul:
 - 1 Operationsverstärker OPA mit dreipoligem Anschluss, das positive/negative Ende kann so ausgewählt werden, dass es intern geerdet ist
 - 5 Komparatoren CMP, Synchron-/Überspannungs-/Spannungstoß-/Stromstoßkomparator 10-Bit-PPG-Timer PPGWDT-Timer • 2-Wege-8Bit-PWM, Periode und Tastverhältnis einstellbar. Überspannung selbstverringend
- Timer:
 - 8-Bit-Timer TIMER0, TIMER2 16-Bit-Timer
 - TIMER1
 - PPG_TIMER
- Tabellensuchfunktion

Modellbeschreibung

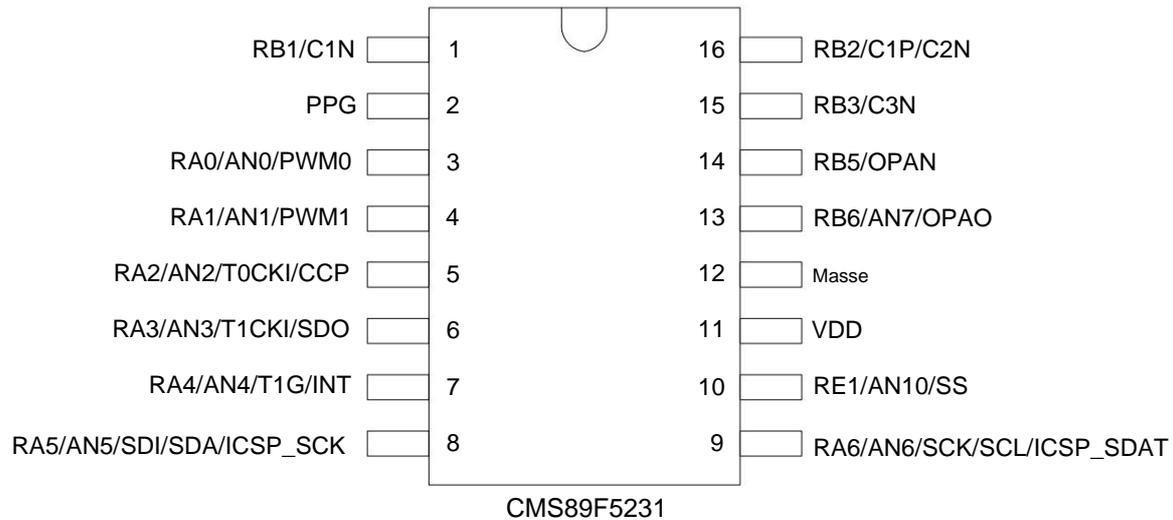
PRODUKT	BLINKEN	RAM	EEPROM E/A		ADC	COMP	OPA	PAKET
CMS89F5231	4Kx16Bit	256	32x16Bit	14	10Bitx9	3	1	DIP16/SOP16
CMS89F5232	4Kx16Bit	256	32x16Bit	14	10Bitx9	3	1	DIP16/SOP16
CMS89F5233	4Kx16Bit	256	32x16Bit	14	10Bitx9	3	1	DIP16/SOP16
CMS89F526	4Kx16Bit	256	32x16Bit	18	10Bitx11	5	1	DIP20/SOP20

1.2 Blockdiagramm der Systemstruktur



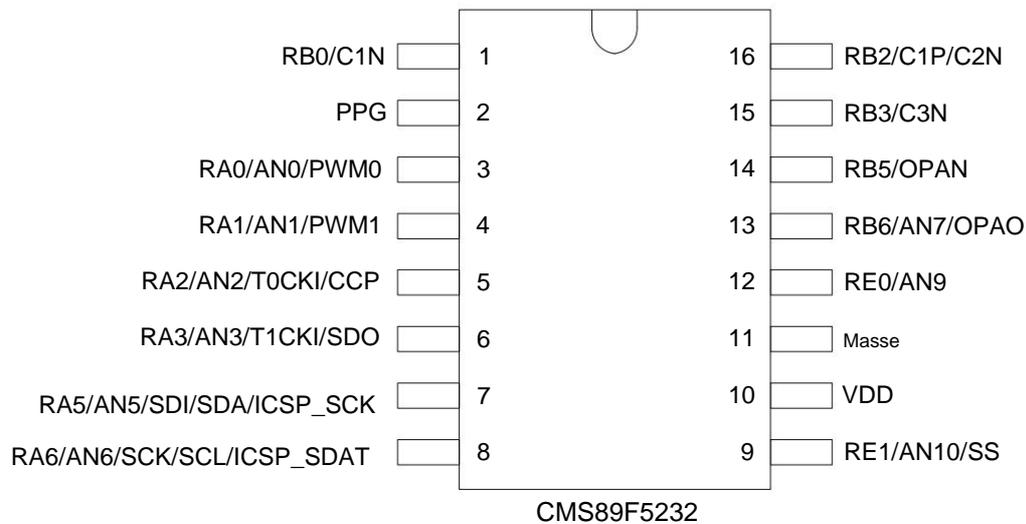
1.3 Pin-Verteilung

1.3.1 Pin-Diagramm CMS89F5231

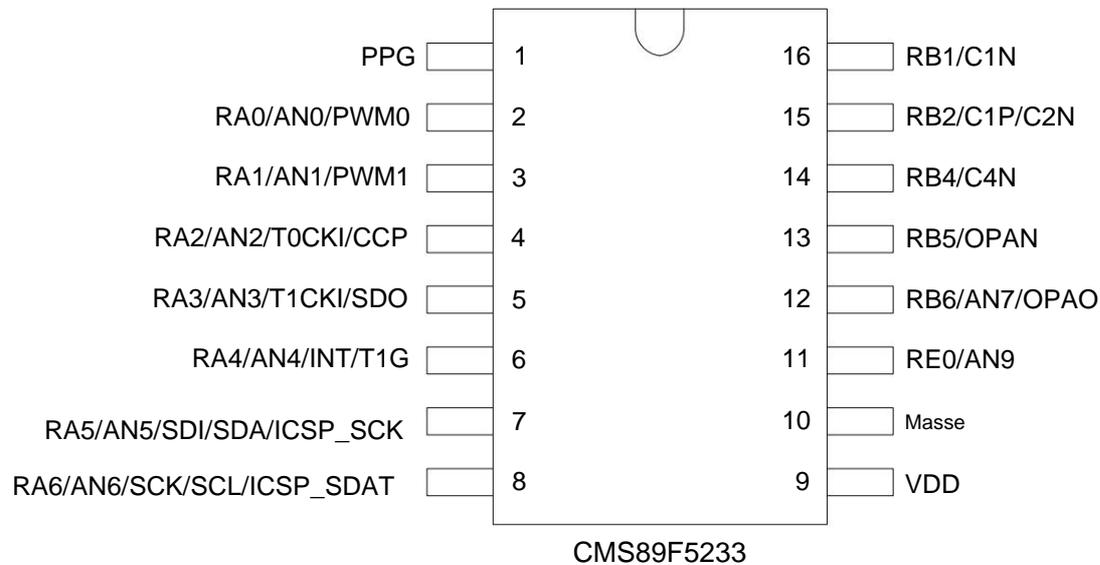


Pin-Beschreibung CMS89F5231:

Pin-Name	Pin-Beschreibung des IO-Typs	
VDD, GND	P Stromversorgungsspannungseingangsstift, Erdungsstift	
RA0-RA6	I/O kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden	
RB1-RB3/ RB5-RB6	I/O kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden	
RE1	I/O kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden	
ICSP_SCK	I Programmierung des Takteingangspins	
ICSP_SDAT	E/A-Programmierungsdaten-Eingangs-/Ausgangs-Pin	
INT	I Externer Interrupt-Eingangspin	
AN0-AN7/AN10	- Eingangspin des AD-Kanals	
T0CKI/T1CKI	- TIMER0/1 externer Takteingangsstift	
T1G	- TIMER1 Gate-Eingangspin	
KPCh	I Capture-Modus-Eingangsstift	
SDI/SDA/SCK/SCL/SDO/SS	E/A	I2C /SPI-Daten/Uhr/Steuerstift
C1P	I Positiver Eingang des Synchronkomparators	
C1N	I Negativer Eingang des Synchronkomparators	
C2N	I Negativer Eingang des Überspannungskomparators	
C3	I Optionale Überspannungsfunktion oder Überspannungsfunktion	
OPAN/OPAO	Negativer Eingangs-/Ausgangspin des I/O-Operationsverstärkers	
PPG	ö	IGBT-Ausgangssteuerstift (Open-Drain-Ausgang)

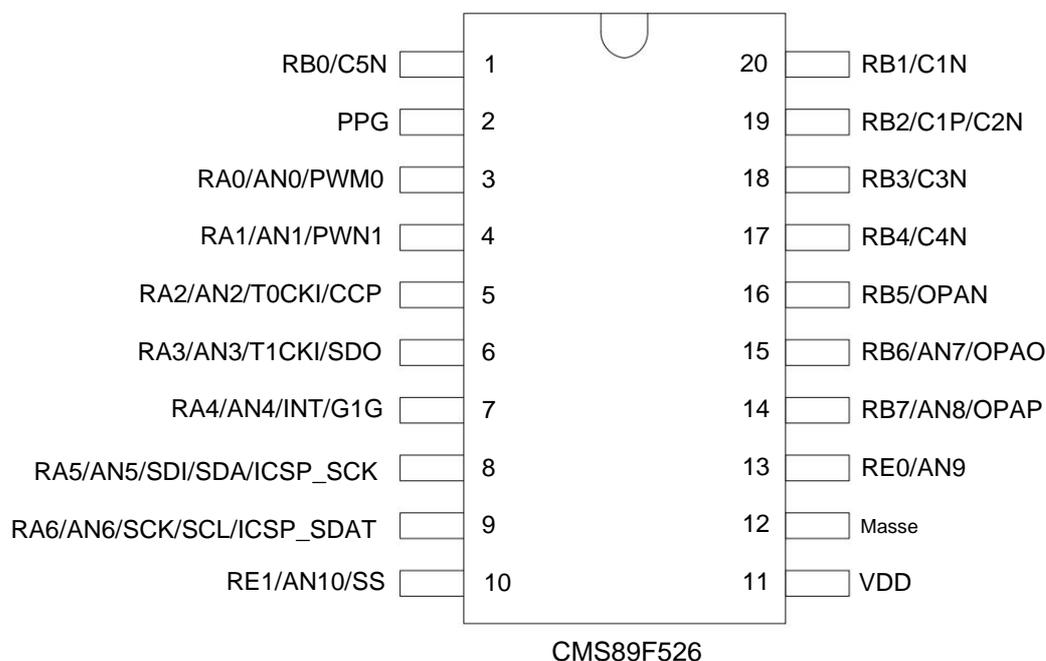
1.3.2 Pin-Diagramm CMS89F5232**Pin-Beschreibung CMS89F5232:**

Pin-Name	Pin-Beschreibung des IO-Typs
VDD, GND	P Stromversorgungsspannungseingangsstift, Erdungsstift
RA0-RA3/ RA5-RA6	I/O kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden
RB0-RB2/ RB5-RB6	I/O kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden
RE0/RE1	I/O kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden
ICSP_SCK	I Programmierung des Takteingangspins
ICSP_SDAT	E/A-Programmierdaten-Eingangs-/Ausgangs-Pin
AN0-AN3/AN5-AN7/AN9-AN10	- Eingangspin des AD-Kanals
T0CKI/T1CKI	- TIMER0/1 externer Takteingangsstift
KPCh	I Capture-Modus-Eingangsstift
SDI/SDA/SCK/SCL/SDO/SS	E/A I2C /SPI-Daten/Uhr/Steuerstift
C1P	I Positiver Eingangspin des Synchronkomparators
C1N	I Negativer Eingangspin des Synchronkomparators
C2N	I Negativer Eingangspin des Überspannungskomparators
C5N	I Minuspol-Eingangsstift des Spannungsstoß-Leistungskomparators
OPAN/OPAO	Negativer Eingangs-/Ausgangspin des I/O-Operationsverstärkers
PPG	O IGBT-Ausgangssteuerstift (Open-Drain-Ausgang)

1.3.3 Pin-Diagramm CMS89F5233

CMS89F5233-Pin-Beschreibung:

Pin-Name	Pin-Beschreibung des IO-Typs	
VDD, GND	P	Stromversorgungsspannungseingangsstift, Erdungsstift
RA0-RA6	I/O	kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden
RB1-RB2/ RB4-RB6	I/O	kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden
RE0	I/O	kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden
ICSP_SCK	I	Programmierung des Takteingangspins
ICSP_SDAT	E/A	Programmierdaten-Eingangs-/Ausgangs-Pin
AN0-AN3/AN5-AN7/AN9-AN10	-	Eingangspin des AD-Kanals
INT	I	Externer Interrupt-Eingangspin
T1G	-	TIMER1 Gate-Eingangspin
T0CKI/T1CKI	-	TIMER0/1 externer Takteingangsstift
KPCh	I	Capture-Modus-Eingangsstift
SDI/SDA/SCK/SCL/SDO/SS	E/A	I2C /SPI-Daten/Uhr/Steuerstift
C1P	I	Positiver Eingangspin des Synchronkomparators
C1N	I	Negativer Eingangspin des Synchronkomparators
C2N	I	Negativer Eingangspin des Überspannungskomparators
C4	I	Minuspoleingangsstift des Stromstoßleistungskomparators
OPAN/OPAO	-	Negativer Eingangs-/Ausgangspin des I/O-Operationsverstärkers
PPG	Ö	IGBT-Ausgangssteuerstift (Open-Drain-Ausgang)

1.3.4 Pin-Diagramm CMS89F526

Pin-Beschreibung CMS89F526:

Pin-Name	Pin-Beschreibung des IO-Typs
VDD, GND	P Stromversorgungsspannungseingangsstift, Erdungsstift
RA0-RA6	I/O kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden
RB0-RB6	I/O kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden
RE0/RE1	I/O kann als Eingangspin, Push-Pull-Ausgangspin, mit Pull-up-Widerstandsfunktion programmiert werden
ICSP_SCK	I Programmierung des Takteingangspins
ICSP_SDAT	E/A-Programmierdaten-Eingangs-/Ausgangs-Pin
INT	I Externer Interrupt-Eingangspin
AN0-AN10	- Eingangspin des AD-Kanals
T0CKI/T1CKI	- TIMER0/1 externer Takteingangsstift
T1G	- TIMER1 Gate-Eingangspin
KPCh	I Capture-Modus-Eingangsstift
SDI/SDA/SCK/SCL/SDO/SS	E/A I2C /SPI-Daten/Uhr/Steuerstift
C1P	I Positiver Eingang des Synchronkomparators
C1N	I Negativer Eingang des Synchronkomparators
C2N	I Negativer Eingang des Überspannungskomparators
C3	I Optionale Überspannungsfunktion oder Überspannungsfunktion
C4	I Minuspol-Eingangsstift des Stromstoßleistungskomparators
C5N	I Minuspol-Eingangsstift des Spannungsstoß-Leistungskomparators
OPAN/OPAO	Negativer Eingangs-/Ausgangspin des I/O-Operationsverstärkers
PPG	O IGBT-Ausgangssteuerstift (Open-Drain-Ausgang)

1.4 Systemkonfigurationsregister

Das Systemkonfigurationsregister (CONFIG) ist die FLASH-Option für den Anfangszustand der MCU. Es kann nur vom CMS-Writer programmiert werden und kann nicht von Benutzern aufgerufen und betrieben werden. Es umfasst die folgenden Inhalte: 1. OSC (Oszillationsmodusauswahl)

- INTRC Interner RC-Oszillator
2. WDT (Watchdog-Auswahl)
- AKTIVIEREN Watchdog-Timer einschalten
 - DEAKTIVIEREN Watchdog-Timer ausschalten
3. SCHÜTZEN (Verschlüsselung)
- DEAKTIVIEREN FLASH-Code ist nicht verschlüsselt
 - AKTIVIEREN Der FLASH-Code ist verschlüsselt, und der nach der Verschlüsselung gelesene Wert ist ungewiss

1.5 Serielle In-Circuit-Programmierung

Der Mikrocontroller kann in der endgültigen Anwendungsschaltung seriell programmiert werden. Die Programmierung kann einfach mit den folgenden 4 Drähten erfolgen: ÿ Stromkabel ÿ Erdungskabel ÿ Datenkabel ÿ Taktkabel

Dies ermöglicht es Benutzern, Boards mit unprogrammierten Geräten zu bauen und nur den Mikrocontroller vor der Produktauslieferung zu programmieren, damit die neueste Version der Firmware oder kundenspezifische Firmware in die MCU programmiert.

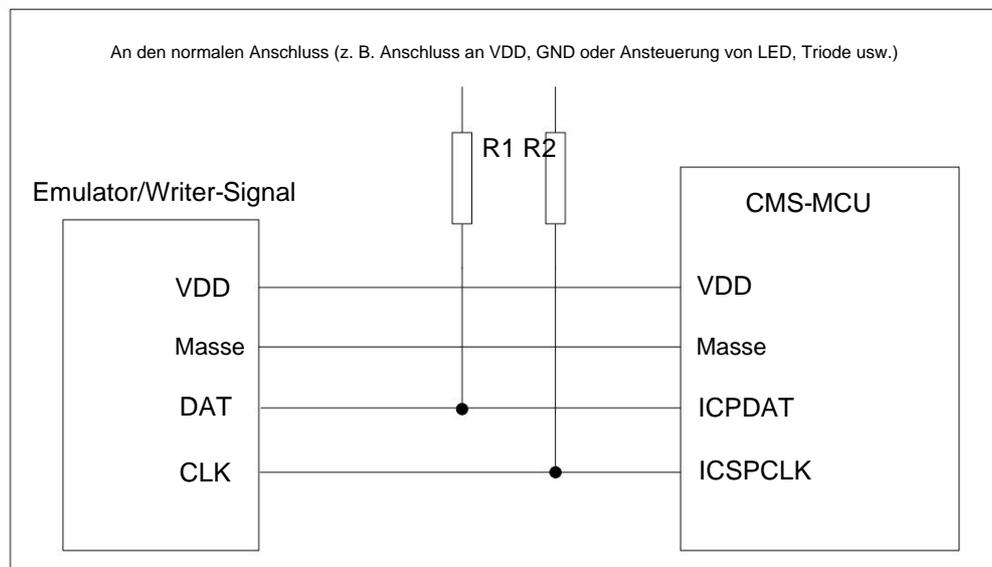


Abbildung 1-1: Typische Verbindungsmethode für die serielle In-Circuit-Programmierung

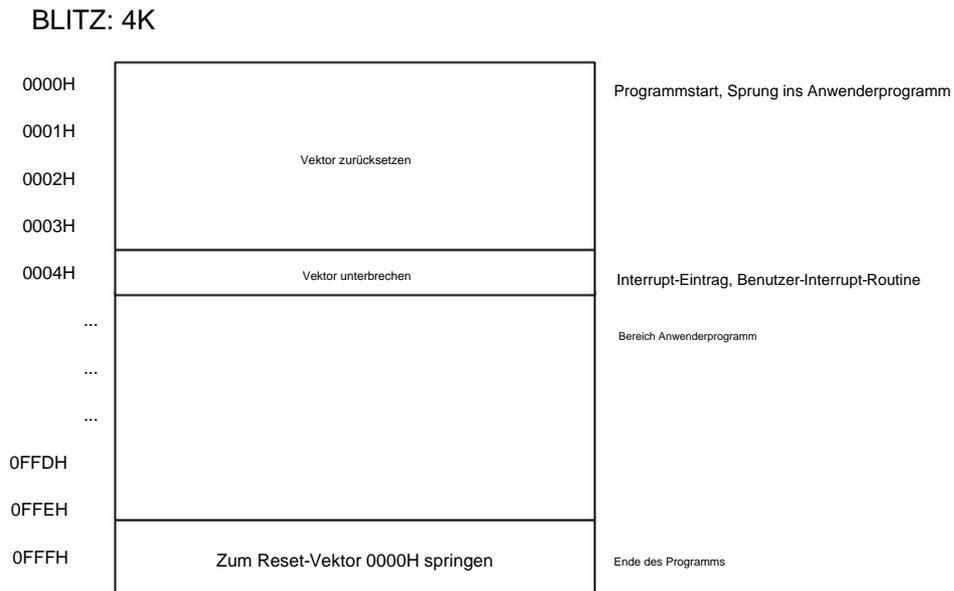
In der obigen Abbildung sind R1 und R2 elektrische Trennvorrichtungen, die häufig durch Widerstände ersetzt werden, und ihre Widerstandswerte sind wie folgt: R1 \geq 4,7K, R2 \geq 4,7K.

2. Zentraleinheit (CPU)

2.1 Speicher

2.1.1 Programmspeicher

CMS89F5231/5232/5233/526 Programmspeicherplatz



2.1.1.1 Reset-Vektor (0000H)

Die MCU hat einen System-Reset-Vektor (0000H) mit einer Wortlänge. Es gibt drei Reset-Methoden:

- Power-On-Reset
- Watchdog-Reset
- Niederspannungs-Reset (LVR)

Nach einem der oben genannten Resets startet das Programm bei 0000H neu und die Systemregister werden ebenfalls auf ihre Standardwerte zurückgesetzt. Entsprechend den Inhalten der PD- und TO-Flag-Bits im STATUS-Register kann der Systemrücksetzmodus beurteilt werden. Das folgende Programm demonstriert, wie der Reset-Vektor in FLASH definiert wird. Beispiel: Definition eines Reset-Vektors

	ORG	0000H	;System-Reset-Vektor
	JP	START	
	ORG	0010H	;Start des Anwenderprogramms
START:	...		; Benutzerprogramm
	...		
	ENDE		; Ende des Programms

2.1.1.2 Unterbrechungsvektor

Die Unterbrechungsvektoradresse ist 0004H. Sobald ein Interrupt bedient wird, wird der aktuelle Wert des Programmzählers PC im Stack-Puffer gespeichert und springt auf 0004H, um mit der Ausführung der Interrupt-Service-Routine zu beginnen. Alle Interrupts werden in den Interrupt-Vektor 0004H eingetragen, wobei der Interrupt vom Benutzer gemäß dem Bit des Interrupt-Request-Flag-Registers ausgeführt wird. Das folgende Beispielprogramm veranschaulicht, wie eine Interrupt-Service-Routine geschrieben wird.

Beispiel: Definieren Sie den Interrupt-Vektor, das Interrupt-Programm wird nach dem Anwenderprogramm platziert

```

                ORG          0000H                ;System-Reset-Vektor
                JP          START
                ORG          0004H                ;Start des Anwenderprogramms
INT_START:
                FORDERUNG   DRÜCKEN             ;ACC und STATUS sichern ;Benutzer-
                ...                                     Interrupt-Programm
                ...
INT_BACK:
                FORDERUNG   POP                 ;Rückgabe ACC und STATUS ;Rückgabe
                RETI                                     unterbrechen
START:
                ...                                     ; Benutzerprogramm
                ...
                ENDE                               ; Ende des Programms

```

Hinweis: Da die MCU keine speziellen Anweisungen zum Knallen und Schieben des Stacks bereitstellt, muss der Benutzer die Interrupt-Site selbst schützen.

Beispiel: Eintrittsschutzstelle unterbrechen

```

DRÜCKEN:
                LD          ACC_BAK,A           ;ACC in Custom-Register ACC_BAK speichern ;High-
                SWAPA      STATUS              und Low-Nibbles des Statusregisters STATUS tauschen
                LD          STATUS_BAK,A       ;Im benutzerdefinierten Register STATUS_BAK speichern

```

Beispiel: Szene beenden verlassen wiederherstellen

```

POP:
                SWAPA      STATUS_BAK         ; Vertausche die hohen und niedrigen Nibbles der in STATUS_BAK gespeicherten Daten mit ACC ;
                LD          STATUS,A          gebe den Wert von A an das Statusregister STATUS ; vertausche die hohen und niedrigen Nibbles
                SWAPR      ACC_BAK           der in ACC_BAK gespeicherten Daten ; vertausche die hohen und niedrigen Nibbles der
                SWAPA      ACC_BAK           gespeicherten Daten zu ACC_BAK zu ACC

```

2.1.1.3 Lookup- Tabelle

Der Chip hat eine Nachschlagetabellenfunktion, und jede Adresse im FLASH-Raum kann als

Nachschlagetabelle verwendet werden. Verwandte Befehle: `ȳ TABLE [R]` Sendet das Low-Byte des

Tabelleninhalts an das Register R und das High-Byte an das Register `TABLE_DATAH`. `ȳ TABLEA` sendet das Low-Byte des Tabelleninhalts an den Akkumulator ACC und das High-Byte an das Register `TABLE_DATAH`.

Zugehörige

Register: `ȳ TABLE_SPH(110H)` Löschbares Register, das verwendet wird, um die hohe 5-Bit-Adresse der Tabelle

anzuzeigen. `ȳ TABLE_SPL(111H)` Löschbares Register, das verwendet wird, um die untere 8-Bit-Adresse der

Tabelle anzuzeigen. `ȳ TABLE_DATAH(112H)` Nur-Lese-Register, das den High-Byte-Inhalt der Tabelle speichert.

Hinweis: Bevor Sie die Tabelle nachschlagen, schreiben Sie die Tabellenadresse in `TABLE_SPH` und `TABLE_SPL`. Wenn sowohl das Hauptprogramm als auch das Unterbrechungsdienstprogramm den Tabellensuchbefehl verwenden, kann sich der Wert von `TABLE_SPH` im Hauptprogramm aufgrund des im Interrupt ausgeführten Tabellensuchbefehls ändern, was zu einem Fehler führt. Das heißt, vermeiden Sie die Verwendung von Tabellensuchbefehlen sowohl im Hauptprogramm als auch in der Interrupt-Service-Routine. Aber wenn es notwendig ist, können wir den Interrupt vor der Tabellensuchanweisung deaktivieren und den Interrupt aktivieren, nachdem die Tabellensuche beendet ist, um Fehler zu vermeiden.

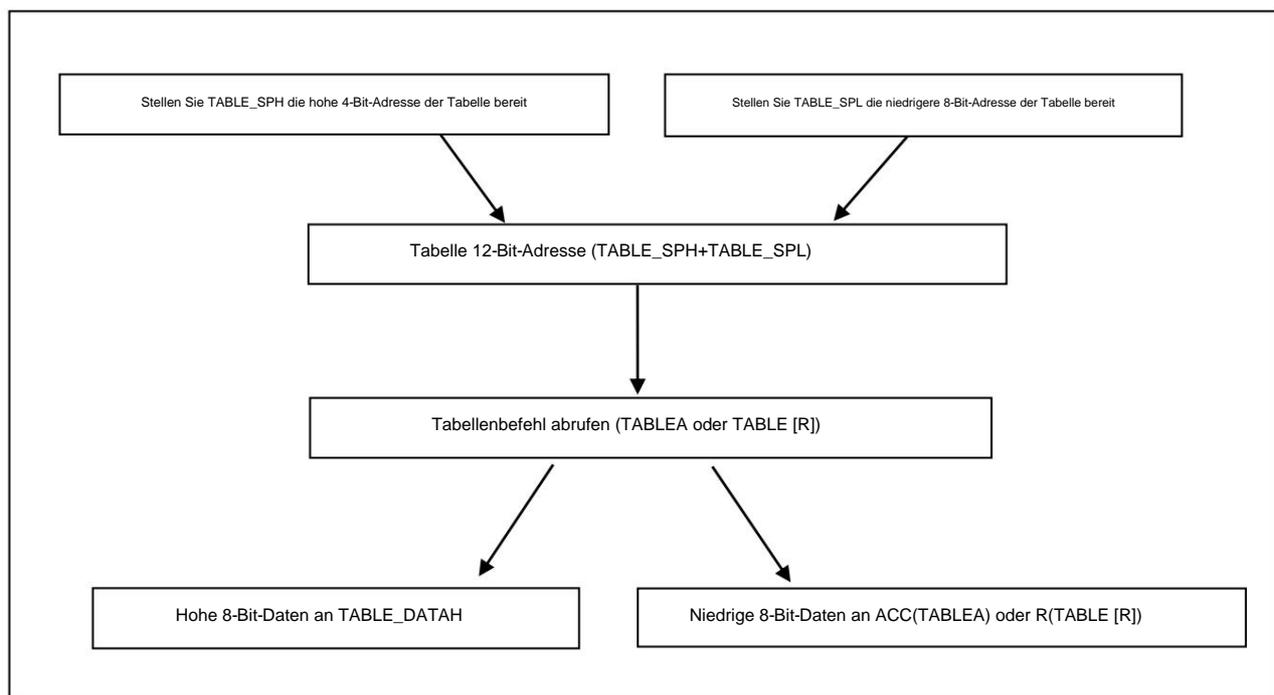


Abbildung 2-1: Flussdiagramm des Tabellenaufrufs

Das folgende Beispiel zeigt, wie die Tabelle im Programm aufgerufen wird.

...			;Uplink-Anwenderprogramm;
LDIA	02H		Tabellen-Low-Adresse
LD	TABLE_SPL,A		
LDIA	06H		;Tabellenhöhe Adresse
LD	TABLE_SPH,A		
TISCH	R01		;Tabellenbefehl, weise die unteren 8 Bits (56H) der Tabelle dem benutzerdefinierten Register R01 zu;
LD	A, TABLE_DATAH		weise die oberen 8 Bits (34H) des Look-Up-Table-Ergebnisses dem Akkumulator ACC zu; weise den
LD	R02,A		ACC-Wert zu (34H) in das benutzerdefinierte Register R02; Anwenderprogramm
...			
ORG	0600H		; Startadresse der Tabelle
DW	1234H		;0600H Inhalt des Adressformulars
DW	2345H		;0601H Inhalt des Adressformulars
DW	3456H		;0602H Inhalt des Adressformulars
DW	0000H		;0603H Inhalt des Adressformulars

2.1.1.4 Sprungtabelle

Die Sprungtabelle kann eine Mehrfachadressen-Sprungfunktion realisieren. Da die neue PCL durch Addieren der Werte von PCL und ACC erhalten werden kann, kann die PCL sein Fügen Sie verschiedene ACC-Werte hinzu, um einen Sprung mit mehreren Adressen zu realisieren. Wenn der ACC-Wert n ist, bedeutet PCL+ACC ein Überlauf auftritt, wird der PC nicht automatisch übertragen, achten Sie also beim Schreiben des Programms um 1, nachdem die aktuelle Anweisung ausgeführt wurde. Bitte beachten Sie das folgende Beispiel. Wenn nach PCL+ACC ein Überlauf auftritt, wird der PC nicht automatisch übertragen, achten Sie also beim Schreiben des Programms darauf. Auf diese Weise kann der Benutzer leicht einen Mehrfachadressensprung realisieren, indem er den Wert von ACC modifiziert.

PCLATH ist das PC-Pufferregister höherer Ordnung. Beim Betrieb von PCL müssen Sie PCLATH zuerst einen Wert zuweisen. Beispiel:

Korrigieren Sie das Multiadressen-Sprungprogrammbeispiel FLASH-Adresse

	LDIA	01H	
	LD	PCLATH, A	; PCLATH muss ein Wert zugewiesen werden
	...		
0110H:	ADDR	PCL	;ACC+PCL
0111H:	JP	SCHLEIFE1	;ACC=0, zu LOOP1 springen
0112H:	JP	SCHLEIFE2	;ACC=1, Sprung zu LOOP2
0113H:	JP	SCHLEIFE3	;ACC=2, zu LOOP3 springen
0114H:	JP	SCHLEIFE4	;ACC=3, Sprung zu LOOP4
0115H:	JP	SCHLEIFE5	;ACC=4, Sprung zu LOOP5
0116H:	JP	SCHLEIFE6	;ACC=5, zu LOOP6 springen

Beispiel: Beispiel für ein falsches Sprungprogramm mit mehreren Adressen

FLASH-Adresse	CLR	PCLATH	
	...		
00FCH:	ADDR	PCL	;ACC+PCL
00FDH:	JP	SCHLEIFE1	;ACC=0, zu LOOP1 springen
00FEH:	JP	SCHLEIFE2	;ACC=1, Sprung zu LOOP2
00FFH:	JP	SCHLEIFE3	;ACC=2, zu LOOP3 springen
0100H:	JP	SCHLEIFE4	;ACC=3, Sprung auf Adresse 0000H
0101H:	JP	SCHLEIFE5	;ACC=4, Sprung auf Adresse 0001H
0102H:	JP	SCHLEIFE6	;ACC=5, Sprung auf Adresse 0002H

Hinweis: Da der PCL-Überlauf nicht automatisch auf das High-Bit übertragen wird, müssen Sie bei der Verwendung von PCL für Multi-Adress-Sprung also auf diesen Abschnitt des Programms nicht achten

Es kann im Seitenumbruch des FLASH-Bereichs platziert werden.

CMS89F52x Special Function Register Summary Bank0

Adresse	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Wert zurücksetzen	
00h	INDF	Das Adressieren dieser Stelle adressiert den Datenspeicher (keine physikalischen Register) unter Verwendung des Inhalts des FSR								XXXX XXXX	
01h	TMR0	Tmr0-Datenregister								xxxx xxxx	
02h	PCL	Niedriges Byte des Programmzählers								0000 0000	
03h	STATUS	IRPs	----	----	ZU	PD	Z	-----	C	0--1 1xxx	
04h	FSR	Indirekter Datenspeicheradressenzeiger								xxxx xxxx	
05h	PORTA	----	RA6	RA5	RA4	RA3	RA2	RA1	RA0	-xxx xxxx	
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	
09h	PORTE	----	----	----	----	----	----	RE1	RE0	---- -xx	
0Ah	PCLATH	----	---	---- Schreibpuffer für die oberen 5 Bits des Programmzählers						---	0000
0Bh	INTCON	GIE	PEIE	TOIE	INTE	----	TOIF	INTF	----	0000-00-	
0Ch	PIR1	EEIF	ADIF	SSPIF	BCLIF	CCPIF	----	TMR2IF	TMR1IF	0000 0-00	
0Dh	PIR2	----	----	C5IF	C4IF	C3IF	C2IF	C1IF	PPGWDTIF	--0 0000	
0Eh	TMR1L	16-Bit-TMR1-Register Low-Byte-Datenregister								xxxx xxxx	
0Fh	TMR1H	16-Bit-TMR1-Register High-Byte-Datenregister								xxxx xxxx	
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	----	----	TMR1CS	TMR1ON	0000 --00	
11h	TMR2	TIMER2-Modulregister								0000 0000	
12h	T2CON	----	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	--000 0000	
14h	PPGTMRL	PPGTMR untere 8 Bit								0000 0000	
15h	PPGTM RH	----	----	----	----	----	----	PPGTMR hoch 2 Bit		---- --00	
16h	PPGDLY	----	----	----	----	PPGDLY				---- 0000	
17h	PPGCON	DETC5F	DETC4F	----	RELOAD_EN	----	----	PPGMD_PPG_ON		11-0 --00	
18h	PWM0DR	PWM0 Arbeitszyklus-Datenregister								xxxx xxxx	
19h	PWM0PR	PWM0-Periodendatenregister								xxxx xxxx	
1Ah	PWM0CR	PWM0EN	PWM0MOD	----	PWM0POL	PWM0CKS[3:0]				00-0 0000	
1Bh	PWM1DR	PWM1 Arbeitszyklus-Datenregister								xxxx xxxx	
1Kanal	PWM1PR	PWM1-Periodendatenregister								xxxx xxxx	
1Dh	PWM1CR	PWM1EN	PWM1MOD	----	PWM1POL	PWM1CKS[3:0]				00-0 0000	
1Fh	ADCON0	ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/FERTIG	ADON	0000 0000	

CMS89F52x Sonderfunktionsregisterzusammenfassung Bank1

Adressname	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 Rücksetzwert		
80h	INDF	Beim Adressieren dieses Speicherorts wird der Inhalt des FSR verwendet, um den Datenspeicher (kein physisches Register) zu adressieren.							xxxx xxxx	
81h	OPTION_REG	----	INTEGR	T0CS	T0SE	PSA	PS2	PS1	PS0	-111 1111
82h	PCL	Das Low-Byte des Programmzählers (PC)							0000 0000	
83h	STATUS	IRPs	----	----	ZU	PD	Z	-----	C	0—1 1xxx
84h	FSR	Indirekter Datenspeicheradressenzeiger							xxxx xxxx	
85h	TRISA	----	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0-111	111
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0 1111	1111
89h	TRISE	----	----	----	----	----	----	TRISE1	TRISE0 ----	---1
8Ah	PCLATH	----	----	----	Schreibpuffer für die oberen 5 Bits des Programmzählers				---	0 0000
8Bh	INTCON	GIE	PEIE	T0IE	INTE	----	T0IF	INTF	----	0000-00-
8Ch	TORTE1	EIE	ADIE	SSPIE	BCLIE	CCPIE	----	TMR2IE	TMR1IE 0000	0-00
8Dh	TORTE2	----	----	C5IE	C4IE	C3IE	C2IE	C1IE PPGWDTIE	--00 0000	
8Fh	OSCCON	----	IRCF2	IRCF1	IRCF0	----	----	----	----	-110----
90h	OSKTUNE	----	----	----	TUN4	TUN3	TUN2	TUN1	TUN0---	0 0000
92h	PR2	TIMER2 Periodenregister							1111 1111	
93h	CM1CNT	CM1OF	CM1CNT[6:0]						0000 0000	
94h	WPUA	----	WPUA6	WPUA5	WPUA4	WPUA3 WPUA2 WPUA1		WPUA0 -000	0000	
95h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3 WPUB2 WPUB1		WPUB0 0000	0000	
96h	WPUE	----	----	----	----	----	----	WPUE1	WPUE0 ----	---00
97h	CM1CON	CM1EN CM1COFM CM1CEN		CM1CLR	CM1NSL	----	----	----	0000 0---	
98h	CM2CON	CM2EN CM2COFM	CM2DBSEL[1:0]			CM2PVSL[3:0]			0000 0000	
99h	CM2CON1	ATPEN	----	----	----	CM2COF	CM2COS[2:0]		0 --- 0000	
9Ah	CM3CON	CM3EN CM3COFM	CM3DBSEL[1:0]			CM3PVSL[3:0]			0000 0000	
9Bh	CM3CON1	CM3M1	CM3M0	----	CM3CIS	CM3COF	CM3COS[2:0]		00-0 0000	
9Ch	CM4CON	CM4EN CM4COFM	CM4DBSEL[1:0]			CM4PVSL[3:0]			0000 0000	
9Dh	CM5CON	CM5EN CM5COFM	CM5DBSEL[1:0]			CM5PVSL[3:0]			0000 0000	
9Ah	ADRESL	Die unteren 8 Bits des A/D-Ergebnisregisters							xxxx xxxx	
9Fh	ADRESSE	Die oberen 2 Bits des A/D-Ergebnisregisters							---- --XX	

CMS89F52x Sonderfunktionsregisterzusammenfassung Bank2

Adresse	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 Rücksetzwert		
100h INDF	Das Adressieren dieses Speicherorts verwendet den Inhalt des FSR, um den Datenspeicher (kein physisches Register) zu adressieren.								xxxx xxxx	
101h TMR0	TIMER0-Modulregister								xxxx xxxx	
102h PCL	Das Low-Byte des Programmzählers (PC)								0000 0000	
103h STATUS	IRPs	----	----	ZU	PD	Z	-----	C	0--1 1xxx	
104h FSR	Indirekter Datenspeicheradressenzeiger								xxxx xxxx	
105h WDTCN	----	----	----	----	----	----	----	SWDTEN ----	---0	
106h PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	
107h OPADJ	OPAOUT	OPARS	--	OPAADJS[4:0]					00-1 0000	
108h OPACON	GEÖFFNET	OPAFM	OPAFE	----	OPAPS1	OPAPS0	OPANS1	OPANS0	0010000	
109h OPACON1	----	----	----	----	OPO2ADE	-----	ANRS1	ANRS0	----0-00	
10Ah PCLATH	----	----	--- Schreibpuffer für die oberen 5 Bits des Programmzählers					----	----	---0 0000
10Bh INTCON	GIE	PEIE	T01E	INTE	----	TOIF	INTF	----	0000-00-	
10Ch EEDATA	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	
10Dh EEADR	----	----	----	EEADR4	EEADR3	EEADR2	EEADR1	----	EEADR0 ----0 0000	
10Äh EEDATH	EEDATH7	EEDATH6	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	---0 0000	
110h TABLE_SPH	Tabelle hoher Zeiger								---X XXXX	
111h TABLE_SPL	niedriger Zeiger der Tabelle								xxxx xxxx	
112h TABLE_DATAH	tabellenhohe Daten								xxxx xxxx	
113h CM1ADJ	CM1AUS	CM1CRS	CM1ADJ[5:0]					0010 0000		
114h CM2ADJ	CM2AUS	CM2CRS	CM2ADJ[5:0]					0010 0000		
115h CM3ADJ	CM3AUS	CM3CRS	CM3ADJ[5:0]					0010 0000		
116h CM4ADJ	CM4AUS	CM4CRS	CM4ADJ[5:0]					0010 0000		
117h CM5ADJ	CM5AUS	CM5CRS	CM5ADJ[5:0]					0010 0000		

CMS89F52x Sonderfunktionsregisterzusammenfassung Bank3

Adressname	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 Rücksetzwert			
180h	INDF	Beim Adressieren dieses Speicherorts wird der Inhalt des FSR verwendet, um den Datenspeicher (kein physisches Register) zu adressieren.							xxxx xxxx		
181h	OPTION_REG	----	INTEGR	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	
182h	PCL	Programmierer (PC) niedriges Byte							0000 0000		
183h	STATUS	IRPs	----	----	ZU	PD	Z	-----	C	0001 1xxx	
184h	FSR	Indirekter Datenspeicheradressenzeiger							xxxx xxxx		
186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	
187h	PANSEL	----	PAANS6 PAANS5	PAANS4	PAANS3	PAANS2	PAANS1 PAANS0	0000 0000			
188h	PBANSEL	PBANS7 PBANS6 PBANS5	PBANS4	PBANS3	PBANS2	PBANS1 PBANS0	0000 0000				
189h	PERLEN	----	----	----	----	----	----	PEANS1 PEANS0	---- --00		
18Ah	PCLATH	----	----	---- Schreibpuffer für die oberen 5 Bits des Programmzählers					----	0000	
18Bh	INKON	GIE	PEIE	T01E	INTE	----	T01F	INTF	----	0000 0000	
18Ch	EECON1	EPPGD	----	----	----	WRERR WREN	WR	RD		0 --- x000	
18Dh	EECON2	EEPROM-Steueregister 2 (kein physikalisches Register)							---		
18Äh	CCPRL	Das Low-Byte des Capture-Registers							xxxx xxxx		
18Fh	CCPRH	High-Byte des Capture-Registers							xxxx xxxx		
190h	CCPCON	CCPEN	----	----	CCPIS	CCPES CPTM2	CPTM1	CPTM0 0--00000			
191H	SSPMSK	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	11111111	
191h	SSPAD	Adressregister für synchrone serielle Schnittstelle (I2C-Modus)							0000 0000		
192h	SSPBUF	Empfangspuffer/Senderegister des synchronen seriellen Ports							xxxx xxxx		
193h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	
194h	SSPCON	WCOL SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0 0000 0000			
195h	SSPCON2	GCEN ACKSTAT ACKDT	ACKEN	RCEN	STIFT	RSEN	Sen	0000 0000			

2.2 Adressierungsmodus

2.2.1 Direkte Adressierung

RAM wird über Arbeitsregister (ACC) betrieben. Beispiel: Der Wert von ACC

wird an das Register 30H gesendet

LD	30H,A
----	-------

Beispiel: Der Wert des 30H-Registers wird an ACC gesendet

LD	A,30H
----	-------

2.2.2 Sofortige Adressierung

Übergeben Sie die unmittelbaren Daten an das Arbeitsregister (ACC).

Beispiel: Direktrufnummer 12H wird an ACC gesendet

LDIA	12H
------	-----

2.2.3 Indirekte Adressierung

Datenspeicher können direkt oder indirekt angesprochen werden. Es ist indirekt über das INDF-Register adressierbar, das kein physisches Register ist. Beim Zugriff auf INDF verwendet es den Wert im FSR-Register (untere 8 Bits) und das IRP-Bit (9. Bit) des STATUS-Registers als Adresse und zeigt auf das Adressregister, wenn also das FSR-Register und STATUS gesetzt sind. Nach dem Setzen des IRP-Bits des Registers kann auf das INDF-Register als Zielregister zugegriffen werden. Das indirekte Lesen von INDF (FSR=0) erzeugt 00H. Das Schreiben in das INDF-Register führt indirekt zu einem No-Op. Die folgenden Beispiele veranschaulichen die Verwendung der indirekten Adressierung in Programmen.

Beispiel: Anwendung von FSR und INDF

LDIA	30H	
LD	FSR,A	;Der indirekte Adressierungszeiger zeigt auf
CLRB	STATUS, IRP	30H; das 9. Bit des Zeigers wird gelöscht; das
CLR	INDF	Löschen von INDF bedeutet tatsächlich das Löschen des 30H-Adress-RAM, auf das FSR zeigt

Beispiel: RAM löschen (20H-7FH) durch indirekte Adressierung:

LDIA	1FH	
LD	FSR,A	;Der Zeiger der indirekten Adressierung zeigt auf 1FH
CLRB	STATUS, IRP	
SCHLEIFE:		
ERHÖHUNG	FSR	;Füge 1 zur Adresse hinzu, die Anfangsadresse
CLR	INDF	ist 30H ;Lösche die Adresse, auf die FSR zeigt
LDIA	7FH	
SUBA	FSR	
SNZB	STATUS,C	;Auf 0 löschen, bis die FSR-Adresse 7FH ist
JP	SCHLEIFE	

2.3 Stapel

Im Chip befinden sich 8 Lagen Stapelpuffer, die weder Teil des Datenspeichers noch Teil des Programmspeichers sind und weder gelesen noch beschrieben werden können. Die Operation darauf wird durch den Stapelzeiger (SP) realisiert, und der Stapelzeiger (SP) kann nicht gelesen oder geschrieben werden. Wenn das System zurückgesetzt wird, zeigt der Stapelzeiger auf die Spitze des Stapels. Wenn ein Unterprogrammaufruf oder Interrupt auftritt, wird der Wert des Programmzählers (PC) in den Stack-Puffer geschoben. Bei der Rückkehr von einem Interrupt oder Unterprogramm wird der Wert an den Programmzähler (PC) zurückgegeben. Die folgende Abbildung zeigt seine Funktionsweise Prinzip.

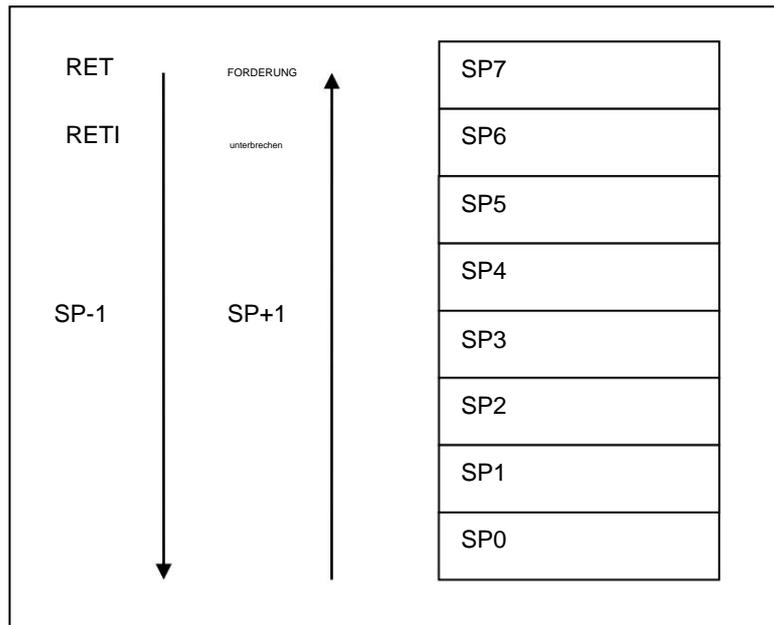


Abbildung 2-2: Funktionsweise des Stapelpuffers

Die Verwendung des Stapelpuffers folgt einem Prinzip von "first in, last out".

Hinweis: Der Stack-Puffer hat nur 8 Schichten. Wenn der Stack voll ist und ein unmaskierbarer Interrupt auftritt, wird nur das Interrupt-Flag aufgezeichnet und die Interrupt-Antwort wird unterdrückt. Der Interrupt wird nicht beantwortet, bis der Stack-Zeiger dekrementiert ist. , kann diese Funktion verhindern, dass Interrupts den Stack überlaufen. Wenn der Stack voll ist und ein Subrutinenaufruf auftritt, wird der Stack ähnlich überlaufen, und der Inhalt, der zuerst in den Stack gelangt, geht verloren, und nur die letzten 8 Rücksprungadressen werden reserviert. Diesen Punkt sollten Sie beim Schreiben des Programms beachten, um das Programm nicht wegfliegen zu lassen.

2.4 Arbeitsregister (ACC)

2.4.1 Übersicht

ALU ist ein 8Bit breites Rechenwerk, durch das alle mathematischen und logischen Operationen der MCU abgewickelt werden. Es kann addieren, subtrahieren,

Schiebe- und Logikoperationen; ALU steuert auch das Statusbit (im STATUS-Statusregister), das verwendet wird, um den Status des Operationsergebnisses anzuzeigen.

Das ACC-Register ist ein 8-Bit-Register, in dem die Operationsergebnisse der ALU gespeichert werden können, es ist aber nicht Teil des Datenspeichers

Befindet sich in der CPU zur Verwendung durch die ALU im Betrieb und kann daher nicht adressiert werden, sondern nur durch die bereitgestellten Anweisungen.

2.4.2 ACC- Anwendung

Beispiel: Verwenden Sie ACC für die Datenübertragung

LD	A, R01	;Wert von Register R01 ACC
LD	R02,A	zuweisen ;Wert von ACC Register R02 zuweisen

Beispiel: Verwenden Sie ACC als Zieloperanden für die unmittelbare Adressierung

LDIA	30H	;30H ACC zuweisen ;"UND"-
ANDIA	30H	Verknüpfung mit aktuellem ACC-Wert und Sofortwert 30H durchführen, ;Ergebnis
XORIA	30H	in ACC einfügen ;"Exklusiv-ODER"-Verknüpfung zwischen aktuellem ACC-Wert
		und Sofortwert 30H ausführen, ;Ergebnis in ACC einfügen

Beispiel: Verwenden Sie ACC als ersten Operanden einer Doppeloperandenanweisung

HSUBA	R01	;ACC-R01, lege das Ergebnis in ACC
HSUBR	R01	;ACC-R01, lege das Ergebnis in R01

Beispiel: Verwenden Sie ACC als zweiten Operanden einer Doppeloperandenanweisung

SUBA	R01	;R01-ACC, lege das Ergebnis in ACC
SUBR	R01	; R01-ACC, trage das Ergebnis in R01 ein

2.5 Programmstatusregister (STATUS)

Das STATUS-Register ist in der folgenden Tabelle dargestellt und

enthält: • Den arithmetischen Status der ALU. • Status zurücksetzen.

• Bankauswahlbits für Datenspeicher (GPR und SFR).

Wie andere Register kann das STATUS-Register das Ziel jeder Anweisung sein. Wenn das STATUS-Register das Ziel eines Befehls ist, der die Z-, DC- oder C-Bits betrifft, können diese 3 Statusbits nicht geschrieben werden. Diese Bits werden entsprechend der Gerätelegik gesetzt oder gelöscht. Auch die TO- und PD-Bits können nicht geschrieben werden. Daher haben Anweisungen, die STATUS als Zielregister verwenden, möglicherweise nicht die erwarteten Ergebnisse.

Beispielsweise löscht CLRSTATUS die oberen 3 Bits und setzt das Z-Bit. Der Wert von STATUS ist dann 000u u1uu (wobei u = unverändert). Daher wird empfohlen, nur die Befehle CLRb, SETb, SWAPA, SWAPR zu verwenden, um das STATUS-Register zu ändern, da diese Befehle keine Statusbits beeinflussen. Programmstatusregister STATUS(03H)

03H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	IRPs	---	---	ZU	PD	Z	<small>Gleichstrom</small>	C
Rücksetzwert	R/W	---	---	R/W	R/W	R/W	R/W	R/W
lesen und schreiben	0	---	---	1	1	X	X	X

Bit7 IRP: Speicherauswahlbits registrieren (für indirekte
 1= Adressierung); Bank2 und Bank3 (100h-1FFh);
 0= Bank0 und Bank1 (00h-FFh).

Bit6~Bit5 deaktiviert

Bit4 ZU:
 1= Zeitüberschreitungsbit; Einschalten oder Ausführung des CLRWDT-Befehls oder
 0= des STOP-Befehls; WDT-Zeitüberschreitung aufgetreten. Ausschalten; Einschalten

Bit3 PD:
 oder CLRWDT-Befehl ausführen; STOP-Befehl ausführen. Das Ergebnis eines
 1= Nullbits; das Ergebnis einer arithmetischen oder logischen Operation ist null; das
 0= Ergebnis einer arithmetischen oder logischen Operation ist nicht null. Half Carry/

Bit2 Z:
 Borrow Bit; Das 4. Low-Bit des Ergebnisses wird zum High-Bit übertragen; Das 4.
 1= Low-Bit des Ergebnisses wird nicht zum High-Bit übertragen. Carry/Borrow-Bit;
 0= Carry trat im höchsten Bit des Ergebnisses auf; Carry trat nicht im höchsten Bit

Bit1 Gleichstrom: des Ergebnisses auf.
 1=
 0=

Bit0 C:
 1=
 0=

Die TO- und PD-Flags können den Grund für den Chip-Reset widerspiegeln. Die Ereignisse, die TO und PD betreffen, und die Zustände von TO und PD nach verschiedenen Resets sind unten aufgelistet.

	AN	PD
Ereignis einschalten	1	1
WDT-Überlauf	0	X
STOP-Befehl	1	0
CLRWDT-Befehl 1 Sleep wirkt sich		1
auf die Ereignistabelle von	1	0

PD und TO aus

AN	PD	Grund zurücksetzen
0	0	WDT-Überlauf weckt schlafende MCU auf
0	1	WDT-Überlauf Nicht-Schlafzustand Drücken
1	0	Sie die Taste, um den Ruhemodus
1	1	aufzuwecken. MCU Der Zustand

von TO/PD nach dem Zurücksetzen beim Einschalten

2.6 Prescaler (OPTION_REG)

Das OPTION_REG-Register ist ein lesbares und beschreibbares Register, das verschiedene Steuerbits für die Konfiguration enthält: • TIMER0/WDT-Prescaler. • TIMER0.

Prescaler OPTION_REG(181H)

181H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPTION_REG Reset-Wert	---	INTEGR	T0CS	T0SE	PSA	PS2	PS1	PS0
lesen und	---	R/W	R/W	R/W	R/W	R/W	R/W	R/W
schreiben	---	1	1	1	1	1	1	1

Bit7 Disable

Bit6 INTEDG: Flankenauswahlbit zum Auslösen des Interrupts.

1 = Steigende Flanke am INT-Pin löst Interrupt aus.

0 = Interrupt ausgelöst durch fallende Flanke am INT-Pin.

Bit5 T0CS: TIMER0 Taktquellen-

0= Auswahlbit. Interner Befehlszyklus (FOSC/4).

1= Kante am T0CKI-Stift.

Bit4 T0SE: TIMER0 Taktquellenflanken-Auswahlbit.

0= Inkrementiert beim Übergang des T0CKI-Pins von niedrig nach hoch.

1= Inkrementiert beim Übergang des Pins T0CKI von High auf Low.

Bit3 PSA: Prescaler-Zuordnungsbits. Der Prescaler ist dem TIMER0-Modul

0= zugeordnet. Der Prescaler ist WDT zugeordnet.

1=

Bit2~Bit0 PS2-PS0: Vorgelegte Parameterkonfigurationsbits.

PS2	PS1	PS0	TMR0-Teilverhältnis	WDT-Teilverhältnis
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

Das Prescaler-Register ist eigentlich ein 8-Bit-Zähler. Bei der Überwachung des Registers WDT wird es als Postscaler verwendet, bei der Verwendung für einen Timer/Zähler wird es als Prescaler verwendet, was im Allgemeinen bezeichnet wird als Prescaler. Es gibt nur einen physikalischen Frequenzteiler im Chip, der nur für WDT oder TIMER0 verwendet werden kann, und die beiden können nicht gleichzeitig verwendet werden. Das heißt, wenn es für TIMER0 verwendet wird, kann WDT den Vorteiler nicht verwenden und umgekehrt. Bei Verwendung mit WDT löscht der CLRWDT-Befehl sowohl den Vorteiler als auch den WDT-Timer. Bei Verwendung für TIMER0 löschen alle Befehle zum Schreiben in TIMER0 (wie: CLR TMR0, SETB TMR0,1 usw.) den Prescaler.

Ob TIMER0 oder der WDT den Prescaler verwendet, unterliegt vollständig der Softwaresteuerung. Sie kann dynamisch geändert werden. Um einen ungewollten Chip-Reset zu vermeiden, sollten beim Umschalten von TIMER0 auf WDT folgende Anweisungen ausgeführt werden.

CLR	TMR0	;TMR0 löschen
CLRWDT		;WDT wird gelöscht
LDIA	B'00xx1111'	Notwendige Schritte müssen ausgeführt
LD	OPTION_REG,A	werden Notwendige Schritte müssen
LDIA	B'00xx1xxx'	ausgeführt werden Neuen Prescaler setzen
LD	OPTION_REG,A	

Um den Vorteiler von der Zuweisung zum WDT auf die Zuweisung zum TIMER0-Modul umzuschalten, sollte der folgende Befehl ausgeführt werden

CLRWDT		;WDT
LDIA	B'00xx0xxx'	löschen ;Neuen Prescaler setzen
LD	OPTION_REG,A	

Hinweis: Um eine 1:1-Prescaler-Konfiguration für TIMER0 zu erhalten, weisen Sie den Prescaler zu zu WDT.

2.7 Programmzähler (PC)

Der Programmzähler (PC) steuert die Ausführungsreihenfolge von Anweisungen im Programmspeicher FLASH, er kann den gesamten Bereich von FLASH ansprechen und Anweisungen erhalten. Nachdem der Code ausgeführt wurde, erhöht sich der Programmzähler (PC) automatisch um eins und zeigt auf die Adresse des nächsten Befehls. Wenn jedoch Sprung, bedingter Sprung, Zuweisung an PCL, Unterprogrammaufruf, Initialisierungsrücksetzung, Unterbrechung, Unterbrechungsrückkehr, Unterprogrammrückkehr und andere Operationen ausgeführt werden, lädt der PC die Adresse, die sich auf die Anweisung bezieht, anstelle der Adresse der nächsten Anweisung.

Wenn ein bedingter Sprungbefehl angetroffen wird und die Sprungbedingung erfüllt ist, wird der nächste während der Ausführung des aktuellen Befehls gelesene Befehl verworfen und a Dummy-Befehlsoperationszyklen, bevor der richtige Befehl erhalten werden kann. Andernfalls wird die nächste Anweisung sequentiell ausgeführt.

Der Programmzähler (PC) ist 12 Bit breit, der Benutzer kann über das PCL-Register (02H) auf die unteren 8 Bit zugreifen, und der Benutzer kann nicht auf die oberen 4 Bit zugreifen. Es kann 4Kx16Bit Programmadressen speichern. Das Zuweisen eines Werts zu PCL erzeugt eine kurze Sprungaktion, und der Sprungbereich beträgt 256 Adressen auf der aktuellen Seite.

Hinweis: Wenn der Programmierer PCL verwendet, um einen kurzen Sprung zu machen, muss er zuerst dem PC-Hochpufferregister PCLATH einen Wert zuweisen.

Die PC-Werte einiger Spezialfälle sind unten angegeben.

Bei Unterbrechung	PC=0000;
beim Zurücksetzen	PC=0004 (das ursprüngliche PC+1 wird automatisch in den Stack geschoben);
Beim Anrufen	PC=programmspezifische Adresse (der ursprüngliche PC+1 wird automatisch in den Stack geschoben);
Beim Betrieb von PCL bei RET, RETI, RETI	PC = der Wert aus dem Stack;
	PC[11:8] bleibt unverändert, PC[7:0] = vom Benutzer festgelegter Wert;
JP-Stunden	PC=der vom Programm vorgegebene Wert;
andere Anweisungen	PC=PC+1;

2.8 Watchdog-Zähler (WDT)

Der Watch Dog Timer (Watch Dog Timer) ist ein selbstschwingender RC-Oszillator-Timer auf dem Chip ohne externe Komponenten, auch wenn der Chip

Die Hauptuhr des Geräts funktioniert nicht mehr und der WDT kann auch weiterzählen. Ein WDT-Timeout erzeugt einen Reset.

2.8.1 WDT -Zyklus

WDT teilt den 8-Bit-Prescaler mit TIMER0. Nach allen Resets beträgt die WDT-Überlaufperiode 18 ms, falls Sie die WDT-Periode ändern müssen

Zeitraum kann das Register OPTION_REG gesetzt werden. Die Überlaufzeit von WDT wird durch Parameter wie Umgebungstemperatur und Versorgungsspannung beeinflusst.

Die Befehle „CLRWDT“ und „STOP“ löschen den WDT-Timer und den Zählwert im Vorteiler (wenn der Vorteiler WDT zugewiesen ist). WDT wird im Allgemeinen verwendet, um zu verhindern, dass das System außer Kontrolle gerät, oder es kann gesagt werden, dass es verhindert, dass das Mikrocontrollerprogramm außer Kontrolle gerät. Unter normalen Bedingungen sollte der WDT durch die Anweisung „CLRWDT“ gelöscht werden, bevor er überläuft, um ein Zurücksetzen zu verhindern. Wenn das Programm aufgrund einer Störung außer Kontrolle gerät, kann die "CLRWDT"-Anweisung nicht ausgeführt werden, bevor der WDT überläuft, was dazu führt, dass der WDT überläuft und einen Reset erzeugt. Lassen Sie das System neu starten, ohne die Kontrolle zu verlieren. Wenn das Zurücksetzen durch einen WDT-Überlauf verursacht wird, wird das "TO"-Bit des Statusregisters (STATUS) auf 0 gelöscht, und der Benutzer kann anhand dieses Bits beurteilen, ob das Zurücksetzen durch einen WDT-Überlauf verursacht wurde oder nicht. hinein.

Hinweis: 1. Wenn Sie die WDT-Funktion verwenden, müssen Sie die „CLRWDT“-Anweisung irgendwo im Programm platzieren, um sicherzustellen, dass sie gelöscht werden kann, bevor der WDT überläuft. null. Andernfalls wird der Chip kontinuierlich zurückgesetzt, was dazu führt, dass das System nicht normal funktioniert.

2. Der WDT kann im Interrupt-Programm nicht gelöscht werden, da sonst das "Durchgehen" des Hauptprogramms nicht erkannt werden kann. 3. Im Programm sollte es einen Vorgang zum Löschen von WDT im Hauptprogramm geben. Versuchen Sie, WDT nicht in mehreren Zweigen zu löschen. Diese Struktur kann maximiert werden Ausübung der Schutzfunktion des Watchdog-Zählers.

4. Die Überlaufzeit des Watchdog-Zählers ist für verschiedene Chips unterschiedlich, daher sollte beim Einstellen der WDT-Löschzeit eine große Redundanz mit der WDT-Überlaufzeit vorhanden sein, um ein unnötiges Zurücksetzen des WDT zu vermeiden.

2.8.2 Watchdog-Timer -Steuerregister WDTCON

Watchdog-Timer-Steuerregister WDTCON(105H)

105H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTCON	---	---	---	---	---	---	---	SWDTEN
R/W	---	---	---	---	---	---	---	R/W
Wert zurücksetzen	---	---	---	---	---	---	---	0

Bit7~Bit5

Nicht implementiert, als 0

Bit5~Bit4

gelesen. Halten (nicht betätigen).

Bit0

SWDTEN: Software aktiviert oder deaktiviert das Watchdog-Timer-

Bit. 1 = WDT aktiviert. 0 = WDT deaktivieren

(Rücksetzwert).

Hinweis: Wenn das WDT-Konfigurationsbit in CONFIG = 1 ist, ist der WDT immer aktiviert, unabhängig vom Zustand des SWDTEN-Steuerbits. Wenn

Wenn das WDT-Konfigurationsbit in CONFIG = 0 ist, können Sie das SWDTEN-Steuerbit verwenden, um den WDT zu aktivieren oder zu deaktivieren.

3. Systemuhr

3.1 Übersicht

Das Taktsignal wird vom internen Oszillator erzeugt, und nach der CONFIG-Frequenzteilung und der Registerfrequenzteilung zur Erzeugung des Systemarbeitstakts (Fsys) werden auf dem Chip vier nicht überlappende Quadratur-Taktsignale erzeugt, die als Q1, Q2, Q3 und bezeichnet werden Q4 bzw. Innerhalb des IC erhöht jeder Q1 den Programmzähler (PC) um eins, und Q4 holt die Anweisung aus der Programmspeichereinheit und speichert sie im Anweisungsregister. Der abgerufene Befehl wird decodiert und zwischen den nächsten Q1 und Q4 ausgeführt, das heißt, ein Befehl wird in 4 Taktzyklen ausgeführt. Die folgende Abbildung zeigt das Zeitdiagramm für die Takt- und Befehlszyklusausführung.

Ein Befehlszyklus enthält 4 Q-Zyklen. Die Ausführung und Erfassung von Befehlen nimmt eine Pipeline-Struktur an. Das Abrufen von Befehlen dauert einen Befehlszyklus, während das Decodieren und Ausführen einen weiteren Befehlszyklus dauert. Aufgrund der Pipeline-Struktur wird jedoch aus Makroperspektive jeder Befehl ausgeführt. Die effektive Ausführungszeit beträgt einen Befehlszyklus. Wenn ein Befehl bewirkt, dass sich die Programmzähleradresse ändert (z. B. JP), ist der vorab abgerufene Befehls-Opcode ungültig, und es dauert zwei Befehlszyklen, um den Befehl abzuschließen, weshalb es zwei Taktzyklen für PC-Operationsbefehle dauert.

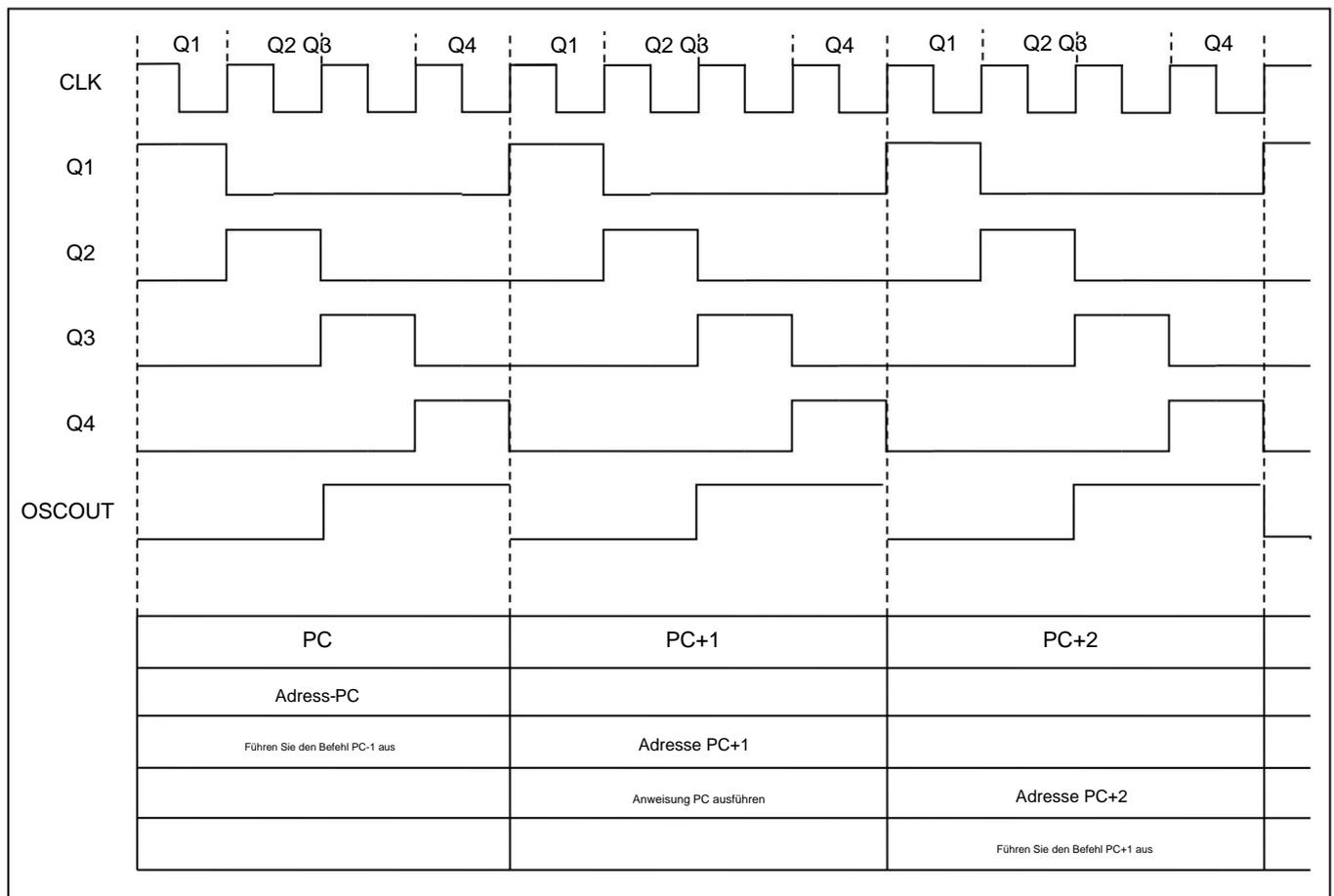


Abbildung 3-1: Takt- und Befehlszyklus-Zeitdiagramm

Die Beziehung zwischen der Systembetriebsfrequenz und der Befehlsgewindigkeit ist unten

aufgeführt: Systembetriebsfrequenz (Fsys)	Dualer	Einzelner
1MHz	Befehlszyklus	Befehlszyklus
2MHz	8ÿs 4ÿs	4ÿs 2ÿs
4MHz	2ÿs	1ÿs
8MHz	1ÿs	500ns
16MHz	500ns	250ns

3.2 Systemoszillator

Der Chip hat nur einen Oszillationsmodus, die interne RC-Oszillation.

3.2.1 Interner RC - Oszillator

Der voreingestellte Oszillationsmodus des Chips ist die interne RC-Oszillation und seine Oszillationsfrequenz F_{osc} beträgt 8 M/16 M. Der Chipbetrieb kann über das OSCCON-Register eingestellt werden.

Arbeitsfrequenz. Die Oszillationsfrequenz wird im Werk kalibriert und ihr Fehler liegt innerhalb von $\pm 3\%$.

3.3 Anlaufzeit

Die Reset-Zeit bezieht sich auf den Zeitraum vom Chip-Reset bis zur Stabilisierung der Chip-Oszillation und ihr Designwert beträgt etwa 18 ms bei 5 V.

Hinweis: Unabhängig davon, ob der Chip durch Einschalten oder aus anderen Gründen zurückgesetzt wird, gibt es diese Startzeit.

3.4 Oszillator-Steuerregister

Das Oszillatorsteuerregister (OSCCON) steuert den Systemtakt und die Frequenzauswahl, und das Oszillatorabstimmregister OSCTUNE kann die interne Oszillatorfrequenz per Software einstellen. Oszillator-Steuerregister OSCCON(8FH)

8FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCON	---	IRCF2	IRCF1	IRCF0	---	---	---	---
R/W	---	R/W	R/W	R/W	---	---	---	---
Wert zurücksetzen	---	1	1	0	---	---	---	---

Bit7 wird nicht verwendet, als 0 gelesen.

Bit6~Bit4

IRCF<2:0>: Auswahlbits für die Frequenz des internen Oszillators.

111 = 8MHz.

110 = 4 MHz (Standard).

101 = 2 MHz.

100 = 1 MHz.

011 = 500 KHz.

010 = 250 KHz.

001 = 125 KHz.

000 = 31 KHz (LFINTOSC).

Bit3~Bit0

Behinderte.

Oszillator-Einstellregister OSCTUNE(90H)

90H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSKTUNE	---	---	---	TUN4	TUN3	TUN2	TUN1	TUN0
R/W	---	---	---	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	---	---	---	0	0	0	0	0

Bit7~Bit5

Nicht benutzt.

Bit4~Bit0

TUN<4:0>: Frequenzanpassungsbits.

01111= Höchste Frequenz.

01110=

.

.

.

00001=

00000 = Oszillatormodul läuft mit werkseitig kalibrierter Frequenz.

11111=

.

.

.

10000= Mindestfrequenz.

Hinweis: Die Oszillationssteuerregister wirken sich alle auf den Systemtakt SystemClock aus (Tsys ist die Systemtaktperiode = 1/Fsys).

4. Zurücksetzen

Der Chip kann auf die folgenden vier Arten zurückgesetzt

werden:

- Power-On-Reset;

- Zurücksetzen auf niedrige

Spannung

- Zurücksetzen des Watchdog-Überlaufs im

Normalbetrieb

- Zurücksetzen des Watchdog-Überlaufs im Schlafmodus

Wenn einer der oben genannten Resets auftritt, stellen alle Systemregister den Standardzustand wieder her, das Programm stoppt die Ausführung und gleichzeitig wird der Programmzähler PC gelöscht, und das Programm beginnt nach dem Reset ab dem Reset-Vektor 0000H zu laufen. Die TO- und PD-Flag-Bits von STATUS können Informationen über den Status des System-Resets geben (siehe die Beschreibung von STATUS für Einzelheiten), und der Benutzer kann den Programmlaufpfad entsprechend dem Status von PD und TO steuern.

Jede Reset-Situation erfordert eine bestimmte Reaktionszeit, und das System bietet einen vollständigen Reset-Prozess, um den reibungslosen Ablauf der Reset-Aktion sicherzustellen.

4.1 Einschalt-Reset

Das Zurücksetzen beim Einschalten ist eng mit dem LVR-Betrieb verbunden. Der Einschaltvorgang des Systems hat die Form einer allmählich ansteigenden Kurve und es dauert eine gewisse Zeit, bis das normale Niveau erreicht ist Wert. Die normale Abfolge des Power-on-Resets ist unten

angegeben:

- Power-on: das System erkennt den Anstieg der Versorgungsspannung und

- wartet auf deren Stabilisierung;
- Systeminitialisierung: alle Systemregister werden auf die

- Anfangswerte gesetzt;
- Oszillator startet : der Oszillator beginnt, den Systemtakt bereitzustellen;

- Ausführen des Programms: Nach dem Einschalten beginnt das Programm zu laufen.

4.2 Power-down-Reset

4.2.1 Überblick über Brownout-Reset

Brown-Out-Reset ist für Systemspannungseinbrüche, die durch externe Faktoren (z. B. Störungen oder Änderungen in externen Lasten) verursacht werden.

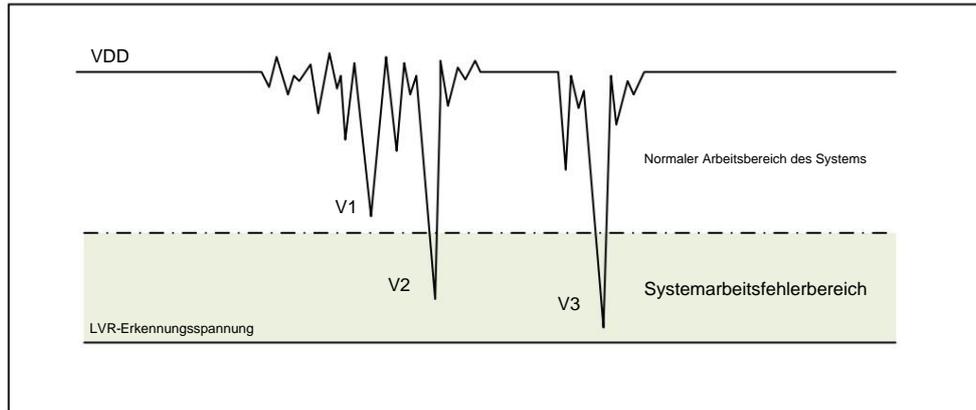


Abbildung 4-1: Schematische Darstellung des Power-Down-Resets

Die obige Abbildung ist ein typisches Power-Down-Reset-Diagramm. In der Figur ist VDD stark gestört und die Spannung fällt sehr niedrig ab. Zonensystem über der gepunkteten Linie

Normalbetrieb, im Bereich unterhalb der gepunkteten Linie tritt das System in einen unbekanntem Arbeitszustand ein, dieser Bereich wird Totzone genannt. Beim Absinken von VDD auf V1 befindet sich das System noch im Normalzustand, beim Absinken von VDD auf V2 und V3 tritt das System in eine fehleranfällige Totzone ein.

Das System kann in den folgenden Situationen in die Totzone eintreten:

• DC im Einsatz:

- DC-Anwendungen werden im Allgemeinen von Batterien versorgt. Wenn die Batteriespannung zu niedrig ist oder der Mikrocontroller die Last steuert, kann die Systemspannung abfallen und in die Todeszone. Zu diesem Zeitpunkt fällt die Stromversorgung nicht weiter auf die LVD-Erkennungsspannung ab, sodass das System in der Totzone bleibt.

• Bei AC-Anwendung: - Wenn

das System mit AC betrieben wird, wird der DC-Spannungswert durch das Rauschen in der AC-Stromversorgung beeinflusst. Wenn die externe Last zu hoch ist, z. B. beim Antreiben eines Motors,

Durch Lastaktionen erzeugte Störungen wirken sich auch auf die Gleichstromversorgung aus. Wenn VDD durch Störungen unter die minimale Betriebsspannung fällt, dann

Das System kann in einen instabilen Betriebszustand übergehen.

- Bei AC-Anwendung ist die Ein- und Ausschaltzeit des Systems relativ lang. Unter ihnen lässt der Einschaltsequenzschutz das System normal einschalten, aber den Ausschaltvorgang

Ähnlich wie bei DC-Anwendungen neigt die VDD-Spannung nach dem Abschalten der AC-Stromversorgung jedoch dazu, während des langsamen Abfalls in die Totzone einzutreten.

Wie in der obigen Abbildung gezeigt, ist der normale Betriebsspannungsbereich des Systems im Allgemeinen höher als die Systemrücksetzspannung, und die Rücksetzspannung wird durch den Niederspannungserkennungspegel (LVR) bestimmt. Wenn die Systemausführungsgeschwindigkeit zunimmt, erhöht sich auch die minimale Betriebsspannung des Systems entsprechend, aber weil die Systemrücksetzspannung fest ist, die minimale Betriebsspannung des Systems Zwischen der Betriebsspannung und der Rücksetzspannung des Systems befindet sich ein Spannungsbereich. Das System kann nicht normal arbeiten und wird nicht zurückgesetzt. Dieser Bereich ist die Totzone.

4.2.2 Verbesserungsmethode des Power-Down-Resets

Hier sind einige Vorschläge, wie Sie die Power-Off-Reset-Leistung des Systems verbessern können:

- Schalten Sie die Unterspannungserkennungsfunktion der MCU ein; • Schalten Sie den Watchdog-Timer ein; • Reduzieren Sie die Betriebsfrequenz des Systems; • Erhöhen Sie die Steigung des Spannungsabfalls.

Schalten Sie die Unterspannungserkennungsfunktion der

MCU ein. Die Unterspannungserkennungsfunktion (LVR) ist in den Chip integriert und kann durch die Programmierung von CONFIG gesteuert werden. Einzelheiten finden Sie in Kapitel 1.5 über die Programmierung der CONFIG-Auswahlweisungen. Wenn die LVR-Funktion aktiviert ist und die Systemspannung unter die LVR-Spannung fällt, wird der LVR ausgelöst und das System zurückgesetzt.

Watchdog-Timer Der

Watchdog-Timer wird verwendet, um den normalen Betrieb des Programms zu gewährleisten. Wenn das System in die Arbeitstotzone eintritt oder das Programm falsch läuft, läuft der Watchdog-Timer über und das System wird zurückgesetzt.

Reduzieren Sie die Betriebsgeschwindigkeit des

Systems Je schneller die Betriebsfrequenz des Systems ist, desto höher ist die minimale Betriebsspannung des Systems. Daher wird der Umfang der Arbeitstotzone erhöht und die minimale Arbeitsspannung kann durch Verringern der Systemarbeitsgeschwindigkeit verringert werden, wodurch die Wahrscheinlichkeit, dass das System in der Totzonenspannung arbeitet, effektiv verringert wird.

Erhöhen Sie die Spannungsabfallsteilheit

Diese Methode kann in einer Umgebung verwendet werden, in der das System mit Wechselstromversorgung arbeitet. In einem allgemeinen Wechselstromversorgungssystem fällt die Systemspannung während des Ausschaltvorgangs sehr langsam ab, was dazu führt, dass der Chip die Totzonenspannung für eine lange Zeit. Wenn das System wieder eingeschaltet wird, kann der Arbeitszustand des Chips falsch sein. Es wird empfohlen, einen Entladewiderstand zwischen der Chip-Stromversorgung und dem Erdungskabel hinzuzufügen, damit die MCU schnell durchqueren Sie die Totzone und betreten Sie die Rücksetzzone, um die Möglichkeit von Chip-Einschaltfehlern zu vermeiden.

4.3 Watchdog-Reset

Watchdog-Reset ist eine Schutzeinstellung für das System. Unter normalen Bedingungen wird der Watchdog-Timer vom Programm gelöscht. Wenn ein Fehler auftritt, befindet sich das System in einem unbekanntem Zustand, der Watchdog-Timer läuft über und das System wird zu diesem Zeitpunkt zurückgesetzt. Nach dem Watchdog-Reset startet das System im Normalzustand neu.

Der Zeitpunkt des Watchdog-Resets ist wie folgt:

- Watchdog-Timer-Status: Das System erkennt, ob der Watchdog-Timer überläuft, und wenn er überläuft, wird das System zurückgesetzt;
- Initialisierung: Alle Systemregister werden auf den Standardzustand gesetzt;
- Oszillator beginnt zu arbeiten: Der Oszillator beginnt mit der Bereitstellung des Systemtakts; -
- Programm: Der Reset ist abgeschlossen und das Programm beginnt zu laufen.

Zur Anwendung des Watchdog-Timers siehe Kapitel 2.8 WDT-Anwendung.

5. Schlafmodus

5.1 Wechseln Sie in den Schlafmodus

Führen Sie die STOP-Anweisung aus, um in den Abschaltmodus zu wechseln. Wenn der WDT aktiviert ist, dann:

- Der WDT wird gelöscht und läuft weiter.
- Das PD-Bit im STATUS-Register wird gelöscht.
- Das TO-Bit wird auf 1 gesetzt.
- Schalten Sie den Oszillatortreiber aus.
- Der E/A-Anschluss behält den Zustand bei, bevor der STOP-Befehl ausgeführt wird (High-Pegel-, Low-Pegel- oder Hochimpedanzzustand des Laufwerks).

Im Schlafmodus sollten zur Minimierung des Stromverbrauchs alle I/O-Pins auf VDD oder GND gehalten werden, ohne dass externe Schaltungen von den I/O abgeleitet werden. Stift verbraucht Strom. Um zu verhindern, dass der Eingangspin schwebt und Schaltstrom einführt, sollte der I/O-Pin des hochohmigen Eingangs extern auf High-Pegel oder Low-Pegel gezogen werden. Um den Stromverbrauch zu minimieren, sollte auch die Wirkung von Pull-up-Widerständen im Inneren des Chips berücksichtigt werden.

5.2 Aufwachen aus dem Winterschlaf

Das Gerät kann durch eines der folgenden Ereignisse aus dem Ruhezustand aufgeweckt werden:

1. Aufwecken des Watchdog-Timers (WDT erzwungene Aktivierung)
2. Peripherer Interrupt.

Die TO- und PD-Bits im STATUS-Register werden verwendet, um die Ursache eines Geräte-Resets zu bestimmen. Das PD-Bit wird beim Einschalten auf 1 gesetzt, während die STOP-Anweisung ausgeführt wird. Die Befehlszeit wird gelöscht. Das TO-Bit wird gelöscht, wenn ein WDT-Aufwecken auftritt.

Wenn die STOP-Anweisung ausgeführt wird, wird die nächste Anweisung (PC+1) vorab abgerufen. Wenn ein Interrupt-Ereignis erwünscht ist, um das Gerät aufzuwecken, muss das entsprechende Interrupt-Freigabe-Bit gesetzt (freigegeben) werden. Das Aufwecken ist unabhängig vom Zustand des GIE-Bits. Wenn das GIE-Bit gelöscht (deaktiviert) ist, fährt das Gerät mit der Ausführung der Anweisung nach der STOP-Anweisung fort. Wenn das GIE-Bit gesetzt (aktiviert) ist, führt das Gerät die Anweisung nach der STOP-Anweisung aus und springt dann zur Interrupt-Adresse (0004h), um Code auszuführen. Wenn die Anweisung nach der STOP-Anweisung nicht ausgeführt werden soll, sollte der Benutzer eine NOP-Anweisung nach der STOP-Anweisung platzieren. Der WDT wird gelöscht, wann immer das Gerät aus dem Energiesparmodus aufwacht, unabhängig von der Ursache des Aufwachens.

5.3 Aufwachen mit Interrupt

Wenn globale Interrupts deaktiviert sind (GIE wird gelöscht) und das Interrupt-Aktivierungsbit und das Interrupt-Flag-Bit einer Interrupt-Quelle gesetzt sind, tritt eines der folgenden Ereignisse auf:

- Wenn der Interrupt generiert wurde, bevor der STOP-Befehl ausgeführt wurde, dann STOP-Anweisung wird als NOP-Anweisung ausgeführt. Daher werden der WDT und sein Prescaler und Postscaler (falls aktiviert) nicht gelöscht, das TO-Bit wird nicht gesetzt und PD wird nicht gelöscht.

- Wenn während oder nach einer STOP-Anweisung ein Interrupt auftritt, wird das Gerät sofort aus dem Sleep-Modus geweckt. Die STOP-Anweisung wird vor dem Aufwachen abgeschlossen. Daher werden der WDT und sein Prescaler und Postscaler (falls aktiviert) gelöscht, das TO-Bit wird gesetzt und PD wird gelöscht. Selbst wenn das Flag-Bit vor der Ausführung des STOP-Befehls auf 0 geprüft wird, kann es auf 1 gesetzt werden, bevor die Ausführung des STOP-Befehls abgeschlossen ist. Um festzustellen, ob ein STOP-Befehl ausgeführt wurde, kann das PD-Bit getestet werden. Wenn das PD-Bit gesetzt ist, wurde der STOP-Befehl als NOP-Befehl ausgeführt. Vor der Ausführung des STOP-Befehls muss ein CLRWDT-Befehl ausgeführt werden, um sicherzustellen, dass der WDT gelöscht wird.

5.4 Anwendungsbeispiel Schlafmodus

Bevor das System in den Ruhemodus wechselt, bestätigen Sie bitte zuerst den Status aller I/Os, wenn der Benutzer einen kleinen Ruhestrom erhalten muss. Zu gewährleistende Ports Jeder Eingangsport hat einen festen Zustand, um eine Erhöhung des Schlafstroms aufgrund des unbestimmten Zustands des Portleitungspegels zu vermeiden, wenn sich die E / A im Eingangszustand befindet; andere Peripheriemodule wie AD ausschalten; je nach Funktion Anforderungen des aktuellen Programms, kann die WDT-Funktion deaktiviert werden, um den Schlafstrom zu reduzieren.

Beispiel: Ein Handler, der in den Ruhezustand übergeht

SCHLAFMODUS:		
CLR	INKON	;Shutdown-Interrupt freigeben
LDIA	B'00000000'	
LD	TRISA, A	
LD	TRISB, A	;Alle I/O als Ausgangsports setzen
LD	TRISC, A	
LD	TRISE, A	
...		; andere Funktionen ausschalten
LDIA	0A5H	
LD	SP_FLAG, A	;Sleep-State-Speicherregister setzen
CLRWDT		(benutzerdefiniert) ;WDT löschen ;STOP-Befehl
STOPPEN		ausführen

5.5 Weckzeit für Schlafmodus

Wenn die MCU aus dem Ruhezustand aufwacht, muss sie auf eine Oszillationsstabilisierungszeit (ResetTime) warten, und der Nennwert dieser Zeit beträgt 18 ms.

6. E/A- Ports

Der Chip hat 3 I/O-Ports: PORTA, PORTB, PORTE (bis zu 17 I/Os). Die lesbaren und beschreibbaren Portdatenregister bieten direkten Zugriff auf diese Ports.

Port-Bit	Pin-Beschreibung	E/A
PORTA	0 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, AN0, PWM0	E/A
	1 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, AN1, PWM1	E/A
	2 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, AN2, TMR0-Takteingang, CCP-Eingang	E/A
	3 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, AN3, TMR1-Takteingang, SPI-Datenausgangsport	E/A
	4 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, AN4, externer Interrupt-Eingang, TMR1-Gate-Eingang	E/A
	5 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, AN5, Online-Programmierung und Emulationsdatenport, I2C-Kommunikationsdatenport I/O	I/O
	6 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, AN6, Online-Programmierung und Emulations-Taktport, I2C-Kommunikations-Taktport I/O	I/O
PORTB	0 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, Spannungsstoßkomparator negativer Eingang	E/A
	1 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, synchroner Komparator-Negativeingang	E/A
	2 Schmitt-Trigger-Eingang, Gegentaktausgang, interner schwacher Pull-up, positiver Eingang des Synchronkomparators, negativer Eingang des Überspannungskomparators I/O	I/O
	3 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, optionaler Spannungsstoß- oder Überspannungskomparator-Minuseingang	E/A
	4 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, Stromstoßkomparator-Negativeingang	E/A
	5 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, negativer Eingang des Operationsverstärkers	E/A
	6 Schmitt-Trigger-Eingang, Gegentaktausgang, interner schwacher Pull-up, AN7, Operationsverstärkerausgang	E/A
	7 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, AN8, positiver Eingang des Operationsverstärkers	E/A
PORTE	0 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, AN9	E/A
	1 Schmitt-Trigger-Eingang, Push-Pull-Ausgang, interner schwacher Pull-up, AN10	E/A

<Tabelle 6-1: Übersicht Portkonfiguration>

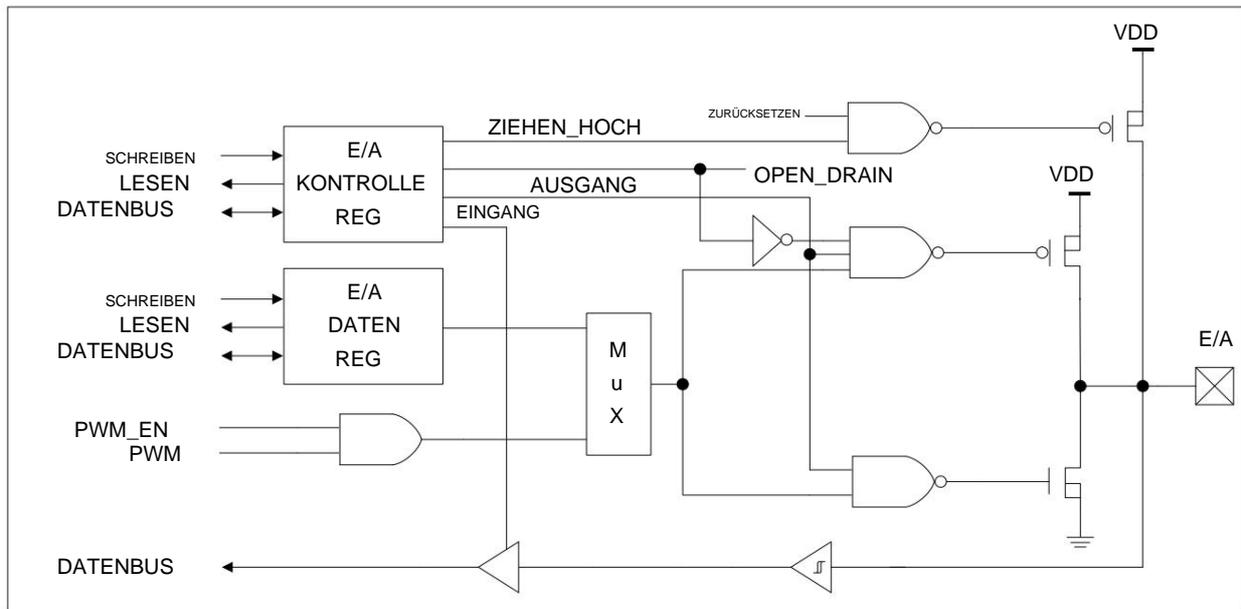
6.1 E/A -Port-Strukturdiagramm

Abbildung 6-1: E/A-Port-Strukturdiagramm (1)

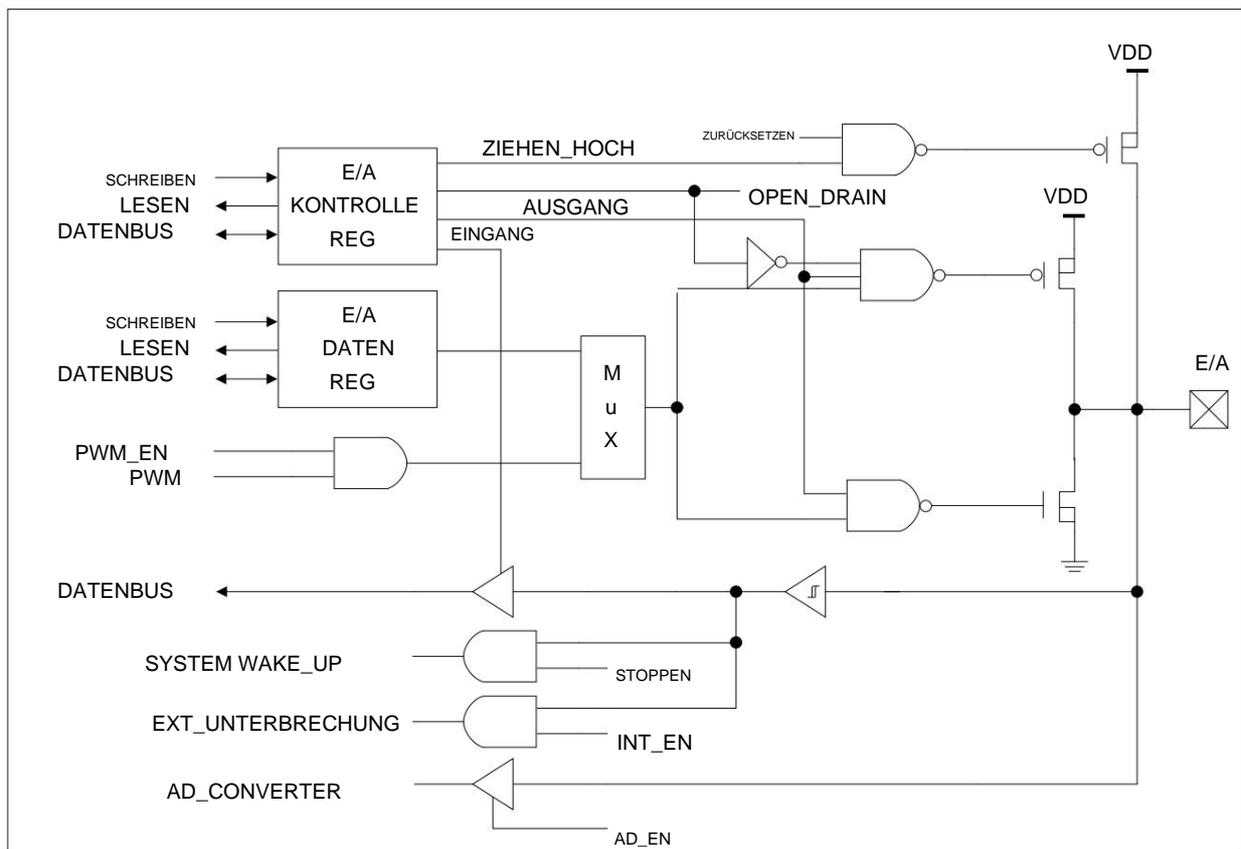


Abbildung 6-2: E/A-Port-Strukturdiagramm (2)

6.2 PORTA

6.2.1 PORTA -Daten und Richtungssteuerung

PORTA ist ein 7Bit breiter bidirektionaler Port. Sein entsprechendes Datenrichtungsregister ist TRISA. Setzen eines Bits von TRISA auf 1 (=1) kann Konfigurieren Sie den entsprechenden Pin als Eingang. Das Löschen eines TRISA-Bits (= 0) konfiguriert den entsprechenden PORTA-Pin als Ausgang.

Das Lesen des PORTA-Registers liest den Zustand der Pins und das Schreiben in dieses Register schreibt in den PORT-Latch. Alle Schreiboperationen sind Lese-Modifizier-Schreib-Operationen. Daher bedeutet das Schreiben auf einen Port, zuerst den Pin-Pegel des Ports zu lesen, den gelesenen Wert zu modifizieren und dann den modifizierten Wert in den Port-Daten-Latch zu schreiben. Das TRISA-Register steuert die Richtung des PORTA-Pins, selbst wenn der PORTA-Pin als analoger Eingang verwendet wird. Bei Verwendung der PORTA-Pins als analoge Eingänge muss der Anwender dafür sorgen, dass die Bits im TRISA-Register gesetzt bleiben. Als analoge Eingänge konfigurierte I/O-Pins lesen immer '0'.

Hinweis: Das ANSEL-Register muss initialisiert werden, um einen analogen Kanal als digitalen Eingang zu konfigurieren. Pins, die als analoge Eingänge konfiguriert sind, lesen 0.

Zu PORTA gehörige Register umfassen PORTA, TRISA, WPUA, PAANSEL usw.

PORTA-Datenregister PORTA (05H)

05H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA	---	RA6	RA5	RA4	RA3	RA2	RA1	RA0
R/W	---	R/W						
Wert zurücksetzen	---	X	X	X	X	X	X	X

Bit6~Bit0 PORTA<6:0>: PORTA/I/O-Pin-Bits; 1 = Port-Pin-Spannungspegel > VIH; 0 = Port-Pin-Spannungspegel < VIL.

PORTA Richtungsregister TRISA(85H)

85H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TRISA	---	TRISA6 TRISA5	TRISA4	TRISA3 TRISA2 TRISA1				TRISA0
R/W	---	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	---	1	1	1	1	1	1	1

Bit6~Bit0 TRISA<6:0>: PORTA-Tristate-Steuersbits;
1 = PORTA-Pin ist als Eingang konfiguriert (Tri-State);
0 = PORTA-Pin ist als Ausgang konfiguriert.

Beispiel: Port-Handler PORTA

CLR	PANSEL	:Stellen Sie alle PORTA-Ports als digitale I/O-Ports ein
LDIA	B'11110000'	:Setze PORTA<3:0> als Ausgangsport und PORTA<6:4> als Eingangsport
LD	TRISA, A	
LDIA	03H	:PORTA<1:0> gibt hohen Pegel aus, PORTA<3:2> gibt niedrigen Pegel aus Da
LD	PORTA,A	PORTA<6:4> ein Eingangsport ist, hat die Zuweisung von 0 oder 1 keine Auswirkung

6.2.2 PORTA- Analog-Auswahlsteuerung

Das PAANSEL-Register wird verwendet, um den Eingangsmodus der I/O-Pins auf den analogen Modus zu konfigurieren. Das Setzen des entsprechenden Bits in PAANSEL bewirkt, dass alle digitalen Lesevorgänge des entsprechenden Pins eine '0' zurückgeben und die analogen Funktionen des Pins normal funktionieren. Der Zustand des PAANSEL-Bits hat keine Auswirkung auf die digitale Ausgangsfunktion. Pins mit gelöschtem TRIS und auf 1 gesetztem PAANSEL funktionieren weiterhin als digitale Ausgänge, aber der Eingangsmodus wird analog. Dies kann zu unvorhersehbaren Ergebnissen führen, wenn Lese-, Änderungs- und Schreibvorgänge an den betroffenen Ports ausgeführt werden.

PORTA analoges Auswahlregister PAANSEL(187H)

187H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PANSEL	---	PAANS6	PAANS5	PAANS4	PAANS3	PAANS2	PAANS1	PAANS0
R/W	---	R/W						
Wert zurücksetzen	---	0	0	0	0	0	0	0

Bit6–Bit0 PAANS<6:0>: Analoge Auswahlbits, wählen jeweils die analoge oder digitale Funktion der Pins PORTA<6:0> aus.
 1 = Analogeingang, der Pin ist als Analogeingang belegt.
 0 = Digital I/O, Pin ist Port oder Sonderfunktion zugeordnet.

6.2.3 PORTA -Pullup -Widerstand

Jeder PORTA-Stift hat einen individuell konfigurierbaren internen schwachen Pull-up. Steuerbits WPUA<6:0> aktivieren oder deaktivieren jeden schwachen Pull-up, wenn der Hafen Wenn ein Pin als Ausgang konfiguriert ist, wird sein schwacher Pullup automatisch abgeschnitten.

PORTA Pull-up-Widerstandsregister WPUA(94H)

94H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WPUA	---	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
R/W	---	R/W						
Wert zurücksetzen	---	0	0	0	0	0	0	0

Bit6–Bit0 WPUA<6:0>: Schwache Pull-up-Registerbits.
 1 = Pullup aktivieren.
 0 = Pullup deaktivieren.

Hinweis: Wenn der Pin als Ausgang konfiguriert ist, wird der schwache Pull-up automatisch deaktiviert.

6.3 PORTB

6.3.1 PORTB -Daten und Richtung

PORTB ist ein 8Bit breiter bidirektionaler Port. Das entsprechende Datenrichtungsregister ist TRISB. Das Setzen eines Bits in TRISB (=1) macht den entsprechenden PORTB-Pin zu einem Eingang. Das Löschen (=0) eines Bits in TRISB macht den entsprechenden PORTB-Pin zu einem Ausgang. Das Lesen des PORTB-Registers liest den Zustand der Pins und das Schreiben in dieses Register schreibt in das PORT-Latch. Alle Schreiboperationen sind Lese-Modifizier-Schreib-Operationen.

Daher bedeutet das Schreiben auf einen Port, zuerst den Pin-Pegel des Ports zu lesen, den gelesenen Wert zu modifizieren und dann den modifizierten Wert in den Port-Daten-Latch zu schreiben. Das TRISB-Register steuert die Richtung des PORTB-Pins, selbst wenn der PORTB-Pin als analoger Eingang verwendet wird. Bei Verwendung des PORTB-Pins als analoger Eingang muss der Anwender dafür sorgen, dass das Bit im TRISB-Register gesetzt bleibt. Als analoge Eingänge konfigurierte I/O-Pins lesen immer '0'.

Zu PORTB gehörige Register umfassen PORTB, TRISB, PBANSEL, WPUB usw.

PORTB-Datenregister PORTB(06H)

06H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	X	X	X	X	X	X	X	X

Bit7~Bit0 PORTB<7:0>: PORTB I/O-Pin-Bits 1 =
 Port-Pin-Pegel > VIH 0 =
 Port-Pin-Pegel < VIL

PORTB Richtungsregister TRISB (86H)

86H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	1	1	1	1	1	1	1	1

Bit7~Bit0 TRISB<7:0>: PORTB Tri-State-Steuerbits
 1 = PORTB-Pin ist als Eingang konfiguriert (Tri-State)
 0 = PORTB-Pin ist als Ausgang konfiguriert

Beispiel: PORTB-Port-Handler

CLR	PORTB	:Datenregister löschen;
LDIA	B'00110000'	PORTB<5:4> als Eingangsport und den Rest als Ausgangsport setzen
LD	TRISB, A	

6.3.2 PORTB- Analogauswahl

Das PBANSEL-Register wird verwendet, um den Eingangsmodus der I/O-Pins auf den analogen Modus zu konfigurieren. Das Setzen des entsprechenden Bits in PBANSEL bewirkt, dass alle digitalen Lesevorgänge des entsprechenden Pins 0 zurückgeben und ermöglichen, dass die analogen Funktionen des Pins normal funktionieren. Der Zustand des PBANSEL-Bits hat keine Auswirkung auf die digitale Ausgangsfunktionalität. Pins mit gelöschtem Richtungsregister und auf 1 gesetztem PBANSEL funktionieren weiterhin als digitale Ausgänge, aber der Eingangsmodus wird analog. Dies kann zu unvorhersehbaren Ergebnissen führen, wenn Lese-, Änderungs- und Schreibvorgänge an den betroffenen Ports ausgeführt werden.

PORTB analoges Auswahlregister PBANSEL(188H)

188H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PBANSEL	PBANS7	PBANS6	PBANS5	PBANS4	PBANS3	PBANS2	PBANS1	PBANS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7~Bit0

PBANS<7:0>: Analoge Auswahlbits, wählen jeweils die analoge oder digitale Funktion von Pin PORTB<7:0> aus.

1 = Analogeingang, der Pin ist als Analogeingang

belegt. 0 = Digital I/O, Pin ist Port oder Sonderfunktion zugeordnet.

6.3.3 PORTB -Pullup -Widerstand

Jeder PORTB-Pin hat einen internen schwachen Pull-up, der individuell konfigurierbar ist. Steuerbits WPUB<7:0> aktivieren oder deaktivieren jeden schwachen Pull-up. Wenn ein Port-Pin als Ausgang konfiguriert ist, wird sein schwacher Pullup automatisch abgeschnitten.

PORTB-Pullup-Widerstandsregister WPUB(95H)

95H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7~Bit0

WPUB<7:0>: Schwache Pull-up-Registerbits.

1 = Pullup aktivieren.

0 = Pullup deaktivieren.

Hinweis: Wenn der Pin als Ausgang konfiguriert ist, wird der schwache Pull-up automatisch deaktiviert.

6.4 PORTE

6.4.1 PORTE -Daten und Richtung

PORTE ist ein 2Bit breiter bidirektionaler Port. Das entsprechende Datenrichtungsregister ist TRISE. Das Setzen (=1) eines Bits in TRISE macht den entsprechenden PORTE-Pin zu einem Eingang. Das Löschen (=0) eines Bits in TRISE macht den entsprechenden PORTE-Pin zu einem Ausgang. Das Lesen des PORTE-Registers liest den Zustand der Pins, und das Schreiben in dieses Register schreibt in den Port-Latch. Alle Schreiboperationen sind Lese-Modifizier-Schreib-Operationen. Daher bedeutet das Schreiben eines Ports, zuerst den Pin-Pegel des Ports zu lesen, den gelesenen Wert zu modifizieren und dann den modifizierten Wert in den Port-Daten-Latch zu schreiben

Gerät.

PORTE-Datenregister PORTE(09H)

09H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTE	---	---	---	---	---	---	RE1	RE0
R/W	---	---	---	---	---	---	R/W	R/W
Wert zurücksetzen	---	---	---	---	---	---	X	X

Bit1~Bit0 PORTE<1:0>: PORTEI/O-Pin-Bits.
 1 = Port-Pin ist hoch. 0
 = Port-Pin ist niedrig.

PORTE Richtungsregister TRISE(89H)

89H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TRISE	---	---	---	---	---	---	TRISE1	TRISE0
R/W	---	---	---	---	---	---	R/W	R/W
Wert zurücksetzen	---	---	---	---	---	---	1	1

Bit1~Bit0 TRISE<1:0>: PORTE-Steuerbits mit drei Zuständen.
 1 = PORTE-Pin ist als Eingang konfiguriert (Tri-State).
 0 = PORTE-Pin ist als Ausgang konfiguriert.

6.4.2 PORTE -Pull-up-Widerstand

Jeder PORTE-Pin hat einen internen schwachen Pull-up, der individuell konfigurierbar ist. Steuerbits WPUE<1:0> aktivieren oder deaktivieren jeden schwachen Pull-up. Wenn ein Port-Pin als Ausgang konfiguriert ist, wird sein schwacher Pullup automatisch abgeschnitten.

PORTE-Pullup-Widerstandsregister WPUE(96H)

96H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WPUD	---	---	---	---	---	---	WPUE1	WPUE0
R/W	---	---	---	---	---	---	R/W	R/W
Wert zurücksetzen	---	---	---	---	---	---	0	0

Bit1~Bit0 WPUE<1:0>: Schwache Pull-up-Registerbits.
 1 = Pullup aktivieren.
 0 = Pullup deaktivieren.

Hinweis: Wenn der Pin als Ausgang konfiguriert ist, wird der schwache Pull-up automatisch deaktiviert.

6.4.3 PORTE- Analogauswahl

Das PEANSEL-Register wird verwendet, um den Eingangsmodus der I/O-Pins auf den analogen Modus zu konfigurieren. Das Setzen des entsprechenden Bits in PEANSEL bewirkt, dass alle digitalen Lesevorgänge des entsprechenden Pins eine '0' zurückgeben und die analogen Funktionen des Pins normal funktionieren. Der Zustand des PEANSEL-Bits hat keine Auswirkung auf die digitale Ausgangsfunktion. Pins mit gelöschtem Richtungsregister und auf 1 gesetztem PEANSEL funktionieren weiterhin als digitale Ausgänge, aber der Eingangsmodus wird analog. Dies kann zu unvorhersehbaren Ergebnissen führen, wenn Lese-, Änderungs- und Schreibvorgänge an den betroffenen Ports ausgeführt werden.

PORTE analoges Auswahlregister PEANSEL(189H)

189H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PERLEN	---	---	---	---	---	---	PEANS1	PEANS0
R/W	---	---	---	---	---	---	R/W	R/W
Wert zurücksetzen	---	---	---	---	---	---	0	0

Bit1~Bit0

PEANS<1:0> Analoge Auswahlbits, wählen jeweils die analoge oder digitale Funktion von Pin PORTE<1:0> aus.

1 = Analogeingang. Pins sind als analoge Eingänge belegt.

0 = digitale E/A. Pins werden Ports oder Sonderfunktionen zugeordnet.

6.5 E/A -Nutzung

6.5.1 E/A - Port schreiben

Die E/A-Port-Register des Chips können wie Mehrzweckregister durch Datenübertragungsbefehle, Bitoperationsbefehle usw. geschrieben werden. Beispiel: E/

A-Port-Programm schreiben

LD	PORTA,A	;Zuweisen des ACC-Werts zum Port
CLRB	PORTB,1	PORTA;Setzen von PORTB.1 auf
CLR	PORTA	Null;Löschen von PORTA auf Null;Setzen
SATZ	PORTA	aller Ausgangsports von PORTA auf
SETB	PORTB,1	1;Setzen von PORTB.1 auf 1

6.5.2 E/A - Port lesen

Beispiel: E/A-Port-Programm lesen

LD	A, PORTA	; Weise ACC den Wert von PORTA
SNZB	PORTA,1	zu; beurteile, ob PORTA,1 1 ist, überspringe die nächste Anweisung, wenn sie 1
SZB	PORTA,1	ist; beurteile, ob PORTA,1 0 ist, überspringe die nächste Anweisung, wenn sie 0 ist

Hinweis: Wenn der Benutzer den Status eines E/A-Ports liest und der E/A-Port ein Eingangsport ist, sind die vom Benutzer zurückgelesenen Daten der Status der externen Ebene der Portleitung.

Wenn der E/A-Port ein Ausgangsport ist, sind die gelesenen Werte die Daten des internen Ausgangsregisters dieser Portleitung.

6.6 Vorsichtsmaßnahmen für die Verwendung von E/A- Ports

Beim Betrieb des E/A-Ports sollten Sie auf folgende Aspekte achten: 1. Wenn

der E/A-Port von Ausgang auf Eingang umgestellt wird, warten Sie mehrere Befehlszyklen, um den Zustand des E/A-Ports zu stabilisieren. 2. Wenn der interne Pull-up-

Widerstand verwendet wird, wenn der I/O von Ausgang zu Eingang konvertiert wird, hängt die Stabilisierungszeit des internen Pegels von dem an den I/O-Port angeschlossenen Kondensator ab. Der Benutzer

sollte die Wartezeit einstellen Zeit entsprechend der tatsächlichen Situation, um sicherzustellen, dass der E / A-Anschluss den Pegel nicht versehentlich scannt. 3. Wenn der E/A-Port ein Eingangsport ist,

sollte sein Eingangspegel zwischen „VDD+0,7 V“ und „GND-0,7 V“ liegen. Wenn die Spannung des Eingangsports nicht innerhalb dieses Bereichs liegt, kann die in der Abbildung unten gezeigte Methode

verwendet werden.

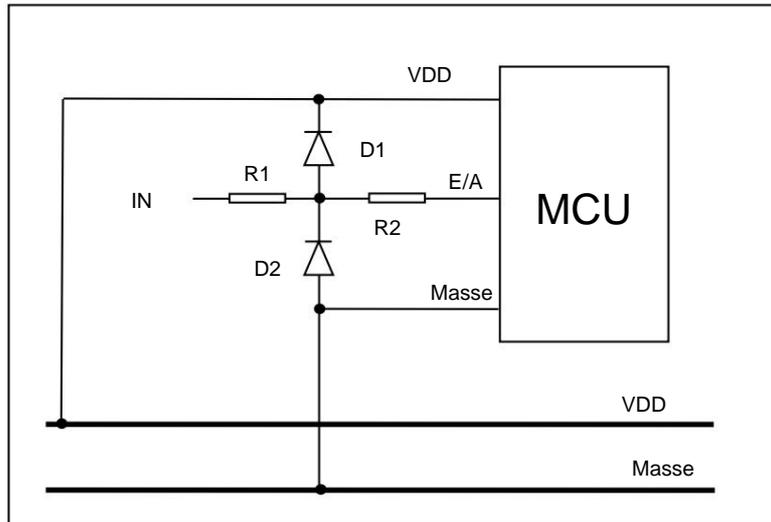


Abbildung 6-3: Die Eingangsspannung liegt bei Verwendung der Schaltung nicht innerhalb des angegebenen Bereichs

4. Wenn eine lange Verbindungsleitung mit dem I/O-Port in Reihe geschaltet ist, fügen Sie bitte einen Strombegrenzungswiderstand in der Nähe des Chip-I/O hinzu, um die EMV-Festigkeit der MCU zu verbessern.

Gewalt.

7. Unterbrechung

7.1 Unterbrechungsübersicht

Der Chip hat die folgenden Interrupt-

- | | |
|--|------------------------------------|
| • TIMER0-Überlauf-Interrupt, | • TIMER1-Überlaufunterbrechung; |
| • TIMER2-Übereinstimmungs-Interrupt, | • INT-Unterbrechung; • CCP- |
| • A/D-Interrupt, • Komparator-Interrupt, • | • Unterbrechung; • PPGWDT- |
| • MSSP-Interrupt, • Interrupt. | • Überlaufunterbrechung; • EEPROM- |
| | • Schreiboperationsunterbrechung. |

Das Unterbrechungssteuerregister (INTCON) und die peripheren Unterbrechungsanforderungsregister (PIR1, PIR2) zeichnen verschiedene Unterbrechungsanforderungen in ihren jeweiligen Flag-Bits auf.

Das INTCON-Register enthält auch einzelne Interrupt-Freigabebits und globale Interrupt-Freigabebits.

Das Global-Interrupt-Enable-Bit GIE (INTCON<7>) gibt, wenn es gesetzt ist, alle unmaskierten Interrupts frei, und wenn es gelöscht ist, sperrt es alle Interrupts. Kann

Einzelne Interrupts können durch die entsprechenden Freigabebits in den Registern INTCON, PIE1, PIE2 deaktiviert werden. GIE wird beim Zurücksetzen gelöscht.

Die Ausführung des „Return from Interrupt“-Befehls RETI verlässt die Interrupt-Service-Routine und setzt das GIE-Bit, wodurch unmaskierte Interrupts wieder aktiviert werden.

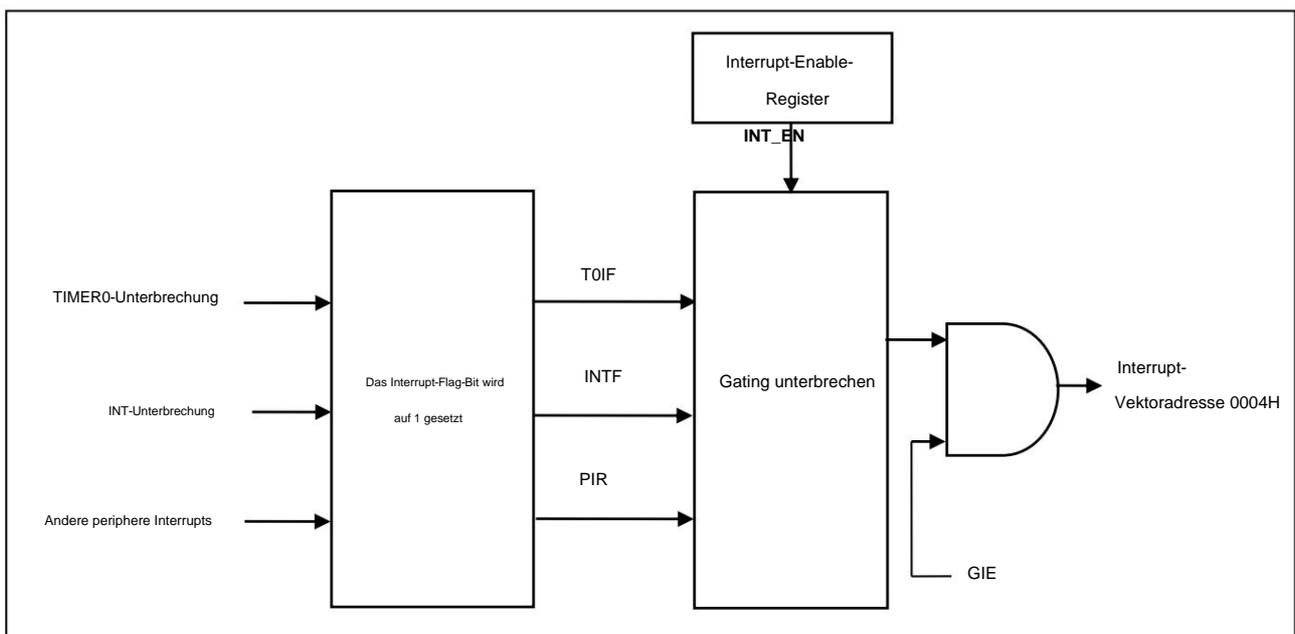


Abbildung 7-1: Schematische Darstellung des Interrupt-Prinzips

7.2 Unterbrechungssteuerregister

7.2.1 Unterbrechungssteuerregister

Das Interrupt-Steuerregister INTCON ist ein lesbares und beschreibbares Register, das die Aktivierung des TMR0-Registerüberlaufs, des PORTB-Port-Level-Change-Interrupts usw. enthält.

Hsu und Zeichenbit.

Wenn eine Unterbrechungsbedingung auftritt, wird unabhängig vom Zustand des entsprechenden Unterbrechungsfreigabebits oder des globalen Freigabebits GIE (im INTCON-Register) die Alle Break-Flags werden auf 1 gesetzt. Die Benutzersoftware sollte sicherstellen, dass das entsprechende Interrupt-Flag-Bit gelöscht wird, bevor ein

Interrupt aktiviert wird. Interrupt-Steuerregister INTCON (0BH)

0BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INKON	GIE	PEIE	TOIE	INTE	---	TOIF	INTF	---
R/W	R/W	R/W	R/W	R/W	---	R/W	R/W	---
Wert 0 zurücksetzen		0	0	0	---	0	0	---

Bit7	GIE: Globales Interrupt-Aktivierungsbit; 1 = alle unmaskierten Interrupts aktivieren; 0 = alle Interrupts deaktivieren.
Bit6	PEIE: Peripheral Interrupt Enable Bit; 1 = alle unmaskierten peripheren Interrupts freigeben; 0 = alle peripheren Interrupts deaktivieren.
Bit5	TOIE: TIMER0-Überlauf-Interrupt-Aktivierungsbit; 1 = TIMER0-Interrupt aktivieren; 0 = TIMER0-Interrupt deaktivieren.
Bit4	INTE: INT-Extern-Interrupt-Freigabebit; 1 = INT-Extern-Interrupt aktivieren; 0 = INT-Extern-Interrupt deaktivieren.
Bit3	Unbenutzt
Bit2	TOIF: TIMER0-Überlauf-Interrupt-Flag (1); 1 = TMR0-Register ist übergelaufen (muss per Software gelöscht werden); 0 = TMR0-Register ist nicht übergelaufen.
Bit1	INTF: INT externes Interrupt-Flag; 1 = INT externer Interrupt tritt auf (muss per Software gelöscht werden); 0 = INT externer Interrupt tritt nicht auf. ungebraucht
Bit0	

Hinweis: Das TOIF-Bit wird gesetzt, wenn TMR0 auf 0 wechselt. Das Zurücksetzen ändert TMR0 nicht, es sollte initialisiert werden, bevor das TOIF-Bit gelöscht wird.

Initialisierung.

7.2.2 Peripheral Interrupt Enable Register

Die peripheren Interrupt-Freigaberegister sind PIE1 und PIE2. Vor der Freigabe eines peripheren Interrupts muss das PEIE-Bit des INTCON-Registers auf 1 gesetzt werden. Peripherie-Interrupt-Freigaberegister PIE1(8CH)

8CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TORTE1	EIE	ADIE	SSPIE	BCLIE	CCPIE	---	TMR2IE TMR1IE	
R/W	R/W	R/W	R/W	R/W	R/W	---	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	---	0	0

Bit7	EEIE: EEPROM-Schreiboperationsunterbrechungs-Freigabebit; 1 = EEPROM-Schreiboperationsunterbrechung aktivieren; 0 = EEPROM-Schreiboperationsunterbrechung deaktivieren.
Bit6	ADIE: Unterbrechungsfreigabebit für A/D-Wandler (ADC); 1 = ADC-Interrupt aktivieren; 0 = ADC-Interrupt deaktivieren.
Bit5	SSPIE: Master Synchronous Serial Port (MSSP) Interrupt-Aktivierungsbit; 1 = MSSP-Interrupt aktivieren; 0 = MSSP-Interrupt deaktivieren.
Bit4	BCLIE: Buskollisionsunterbrechungs-Freigabebit; 1 = Buskollisionsunterbrechung freigeben; 0 = Buskollisionsunterbrechung sperren.
Bit3	CCPIE: CCP-Interrupt-Aktivierungsbit; 1 = CCP-Interrupt aktivieren; 0 = CCP-Interrupt deaktivieren. Nicht benutzt.
Bit2	
Bit1	TMR2IE: TIMER2 und PR2 stimmen mit dem Interrupt-Freigabebit überein; 1 = TMR2-zu-PR2-Übereinstimmungsunterbrechung aktivieren; 0 = TMR2-zu-PR2-Übereinstimmungsunterbrechung deaktivieren.
Bit0	TMR1IE: TIMER1-Überlauf-Interrupt-Aktivierungsbit; 1 = TIMER1-Überlauf-Interrupt aktivieren; 0 = TIMER1-Überlauf-Interrupt deaktivieren.

Peripherie-Interrupt-Aktivierungsregister PIE2(8DH)

8DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TORTE2	---	---	C5IE	C4IE	C3IE	C2IE	C1IE	PPGWDTIE
R/W	---	---	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	---	---	0	0	0	0	0	0

Bit7~Bit6

ungebraucht

Bit5

C5IE: Komparator-C5-Interrupt-Aktivierungsbit; 1 =
= Komparator-C5-Interrupt aktivieren; 0 =
Komparator-C5-Interrupt deaktivieren.

Bit4

C4IE: Komparator-C4-Interrupt-Aktivierungsbit; 1 =
= Komparator-C4-Interrupt aktivieren; 0 =
Komparator-C4-Interrupt deaktivieren.

Bit3

C3IE: Komparator-C3-Interrupt-Aktivierungsbit; 1 =
= Komparator-C3-Interrupt aktivieren; 0 =
Komparator-C3-Interrupt deaktivieren.

Bit2

C2IE: Komparator-C2-Interrupt-Aktivierungsbit; 1 =
= Komparator-C2-Interrupt aktivieren; 0 =
Komparator-C2-Interrupt deaktivieren.

Bit1

C1IE: Komparator-C1-Interrupt-Aktivierungsbit; 1 =
= Komparator-C1-Interrupt aktivieren; 0 =
Komparator-C1-Interrupt deaktivieren.

Bit0 PPGWDTIE: PPGWDT-Überlauf-Interrupt-Aktivierungsbit; 1 =
PPGWDT-Überlauf-Interrupt aktivieren; 0 =
PPGWDT-Überlauf-Interrupt deaktivieren.

7.2.3 Peripheral Interrupt Request Register

Die peripheren Unterbrechungsanforderungsregister sind PIR1 und PIR2. Wenn eine Unterbrechungsbedingung auftritt, unabhängig von dem entsprechenden Unterbrechungsfreigabebit oder globalen Freigabebit GIE

Unabhängig vom Status wird das Interrupt-Flag-Bit auf 1 gesetzt. Die Benutzersoftware sollte sicherstellen, dass das entsprechende Interrupt-Flag-Bit gelöscht wird, bevor ein Interrupt aktiviert wird.

Peripherie-Interrupt-Request-Register PIR1(0CH)

0CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR1	EEIF	ADIF	SSPIF	BCLIF	CCPIF	---	TMR2IF TMR1IF	
R/W	R/W	R/W	R/W	R/W	R/W	---	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	---	0	0

Bit7	EEIF: EE-Schreibvorgang-Unterbrechungsflag; 1 = Schreibvorgang abgeschlossen (muss per Software gelöscht werden); 0 = Schreibvorgang nicht abgeschlossen oder noch nicht gestartet.
Bit6	ADIF: A/D-Wandler-Interrupt-Flag; 1 = A/D-Wandlung ist abgeschlossen (muss per Software gelöscht werden); 0 = A/D-Wandlung ist nicht abgeschlossen oder hat noch nicht begonnen.
Bit5	SSPIF: Interrupt-Flag des Master Synchronous Serial Port (MSSP). 1 = Eine MSSP-Unterbrechungsbedingung ist aufgetreten und muss durch Software gelöscht werden, bevor von der Unterbrechungsdienstroutine zurückgekehrt wird. Die Bedingungen, die dieses Bit auf 1 setzen, sind: - SPI; - Übertragung/Empfang - erfolgt; I2C Slave/ Master; - Übertragung/ - Empfang erfolgt; I2C Master; - Startbedingung tritt durch MSSP-Modul auf; - Stoppbedingung tritt durch MSSP-Modul auf; - Neustartbedingung wird durch MSSP-Modul erfüllt; - Bestätigungsbedingung tritt auf durch das MSSP-Modul; - Die Startbedingung tritt auf, wenn das MSSP-Modul im Leerlauf ist (Multi-Master-System); - Die Stoppbedingung tritt auf, wenn das MSSP-Modul im Leerlauf ist (Multi-Master-System). 0 = Es ist keine MSSP-Unterbrechungsbedingung aufgetreten.
Bit4	BCLIF: Buskollisions-Interrupt-Flag; 1 = Buskollision in MSSP aufgetreten, wenn als I2C-Master-Modus konfiguriert; 0 = Keine Buskollision aufgetreten.
Bit3	CCP1IF: CCP1-Interrupt-Flag-Bit. Erfassungsmodus: 1 = Eine Erfassung des TMR1-Registers ist aufgetreten (muss durch Software gelöscht werden); 0 = Es ist keine Erfassung des TMR1-Registers aufgetreten. Vergleichsmodus: 1 = Eine Vergleichsübereinstimmung des TMR1-Registers ist aufgetreten (muss per Software gelöscht werden); 0 = Eine Vergleichsübereinstimmung des TMR1-Registers ist nicht aufgetreten. PWM-Modus: Wird in diesem Modus nicht verwendet.
Bit2	Behinderte.
Bit1	TMR2IF: TIMER2 und PR2 stimmen mit dem Interrupt-Flag überein. 1 = Eine Übereinstimmung zwischen TIMER2 und PR2 ist aufgetreten (muss in der Software gelöscht werden); 0 = Fehlanpassung zwischen TIMER2 und PR2.
Bit0	TMR1IF: TIMER1-Überlauf-Interrupt-Flag. 1 = TMR1-Register übergelaufen (muss per Software gelöscht werden); 0 = TMR1-Register nicht übergelaufen.

Peripherie-Interrupt-Request-Register PIR2 (ODH)

ODH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR2	---	---	C5IF	C4IF	C3IF	C2IF	C1IF PPGWDTIF	
R/W	---	---	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	---	---	0	0	0	0	0	0

Bit7~Bit6

Behinderte.

Bit5

C5IF: Interrupt-Flag des Komparators C5;

1 = Komparatorausgang (C5OUT-Bit) geändert (muss per Software gelöscht werden)

0 = Komparatorausgang (C5OUT-Bit) unverändert.

Bit4

C4IF: Interrupt-Flag des Komparators C4;

1 = Komparatorausgang (C4OUT-Bit) geändert (muss per Software gelöscht werden)

0 = Komparatorausgang (C4OUT-Bit) unverändert.

Bit3

C3IF: Interrupt-Flag des Komparators C3;

1 = Komparatorausgang (C3OUT-Bit) geändert (muss per Software gelöscht werden)

0 = Komparatorausgang (C3OUT-Bit) unverändert.

Bit2

C2IF: Interrupt-Flag des Komparators C2;

1 = Komparatorausgang (C2OUT-Bit) geändert (muss per Software gelöscht werden)

0 = Komparatorausgang (C2OUT-Bit) unverändert.

Bit1

C1IF: Interrupt-Flag des Komparators C1;

1 = Komparatorausgang (C1OUT-Bit) geändert (muss per Software gelöscht werden)

0 = Komparatorausgang (C1OUT-Bit) unverändert.

Bit0 PPGWDTIF: PPGWDT-Überlauf-Interrupt-Flag;

1 = PPGWDT-Überlauf aufgetreten (muss per Software gelöscht werden);

0 = PPGWDT-Überlauf nicht aufgetreten.

7.3 Schutzmethoden für unterbrochene Sites

Nachdem eine Unterbrechungsanforderung auftritt und beantwortet wird, geht das Programm zu 0004H über, um das Unterbrechungsunterprogramm auszuführen. Bevor auf den Interrupt geantwortet wird, müssen die Inhalte von ACC und STATUS gesichert werden. Der Chip bietet keine speziellen Push-to-Save- und Pop-to-Restore-Anweisungen, und Benutzer müssen den Inhalt von ACC und STATUS schützen, um mögliche Fehler beim Ausführen des Programms nach dem Ende des Interrupts zu vermeiden.

Beispiel: Stapelschutz für ACC und STATUS

	ORG	0000H	
	JP	START	; Anfangsadresse des Anwenderprogramms
	ORG	0004H	
	JP	INT_SERVICE	; Serviceroutine unterbrechen
	ORG	0008H	
START:			
	...		
	...		
INT_SERVICE:			
DRÜCKEN:			;Serviceroutineneintrag unterbrechen, ACC und STATUS
	LD	ACC_BAK,A	speichern ;Wert von ACC speichern, (ACC_BAK muss angepasst werden)
	SWAPA	STATUS	
	LD	STATUS_BAK,A	;Save the value of STATUS, (STATUS_BAK muss angepasst werden)
	...		
	...		
POP:			;Exit der Serviceroutine unterbrechen, ACC und STATUS wiederherstellen
	SWAPA	STATUS_BAK	
	LD	STATUS,A	;den Wert von STATUS wiederherstellen;den
	SWAPR	ACC_BAK	Wert von ACC wiederherstellen
	SWAPA	ACC_BAK	
	RETI		

7.4 Unterbrechungspriorität und mehrfache Unterbrechungsverschachtelung

Die Priorität jedes Interrupts des Chips ist gleich. Wenn ein Interrupt im Gange ist, wird auf einen anderen Interrupt nicht geantwortet. Auf den nächsten Interrupt kann nur nach Ausführung des "RETI"-Befehls geantwortet werden. Wenn mehrere Interrupts gleichzeitig auftreten, hat die MCU keine voreingestellte Interrupt-Priorität. Erstens muss die Priorität jedes Interrupts voreingestellt werden, zweitens steuern Sie mit dem Interrupt-Enable-Bit und dem Interrupt-Control-Bit, ob das System auf den Interrupt antwortet. Im Programm müssen das Interrupt-Steuerbit und das Interrupt-Request-Flag erkannt werden.

8. Zeitzähler TIMER0

8.1 Übersicht Zeitzähler TIMER0

TIMER0 besteht aus den folgenden

Funktionen: • 8-Bit-Timer/Zählerregister (TMR0); • 8-Bit-

Vorteiler (gemeinsam mit dem Watchdog-Timer); •

Programmierbare interne oder externe Taktquelle; •

Programmierbare externe Taktflankenauswahl; • Überlauf-

Interrupt .

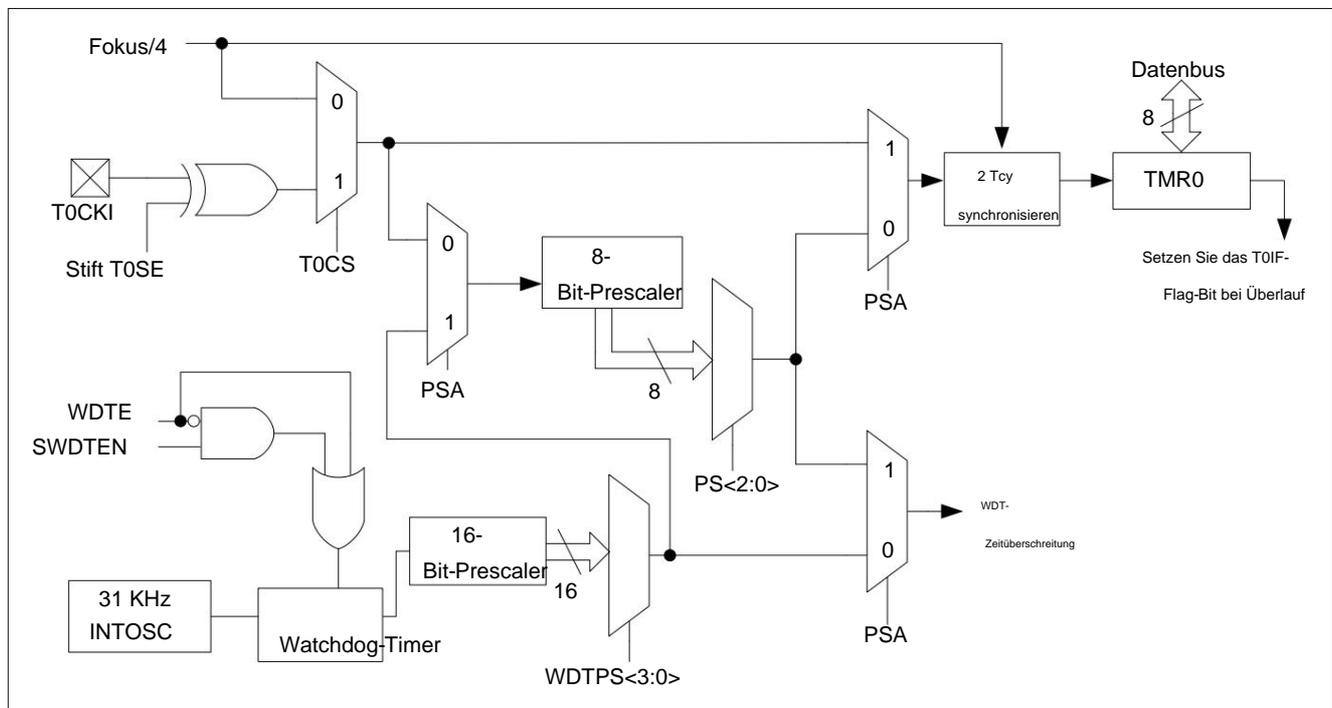


Abbildung 8-1: Blockdiagramm des TIMER0/WDT-Moduls

Hinweis: 1. T0SE, T0CS, PSA, PS<2:0> sind Bits im OPTION_REG-Register.

2. SWDTEN und WDTPS<3:0> sind Bits im WDTCN-Register.

3. Das WDTE-Bit befindet sich in Konfigurationswort 1.

8.2 Arbeitsprinzip von TIMER0

Das TIMER0-Modul kann entweder als 8-Bit-Timer oder als 8-Bit-Zähler verwendet werden.

8.2.1 8 -Bit-Timer-Modus

Wenn es als Timer verwendet wird, erhöht das TIMER0-Modul jeden Befehlszyklus (ohne Prescaler). Der Timer-Modus wird ausgewählt, indem das T0CS-Bit des OPTION_REG-Registers auf 0 gelöscht wird. Wenn ein Schreibvorgang in das TMR0-Register ausgeführt wird, werden Inkremente für die nächsten zwei Befehlszyklen gesperrt. Der in das TMR0-Register geschriebene Wert kann angepasst werden, um eine Latenzzeit von zwei Befehlszyklen beim Schreiben in TMR0 zu berücksichtigen.

8.2.2 8 -Bit-Zählermodus

Beim Betrieb als Zähler erhöht das TIMER0-Modul bei jeder steigenden oder fallenden Flanke des T0CKI-Pins. Die inkrementierende Flanke hängt von ab T0SE-Bit des OPTION_REG-Registers. Der Zählermodus wird ausgewählt, indem das T0CS-Bit des OPTION_REG-Registers gesetzt wird.

8.2.3 Softwareprogrammierbarer Prescaler

TIMER0 und Watchdog Timer (WDT) teilen sich einen softwareprogrammierbaren Prescaler, aber nicht gleichzeitig. Der Vorteiler wird zugewiesen durch Gesteuert durch das PSA-Bit des OPTION_REG-Registers. Um den Prescaler TIMER0 zuzuweisen, muss das PSA-Bit auf 0 gelöscht werden.

Das TIMER0-Modul hat 8 Prescaler-Auswahlmöglichkeiten von 1:2 bis 1:256. Es kann durch die PS<2:0>-Bits des OPTION_REG-Registers gesteuert werden. Wählen Sie das Prescaler-Verhältnis aus. Damit das TIMER0-Modul einen 1:1 Vorteiler hat, muss der Vorteiler dem WDT-Modul zugewiesen werden.

Der Vorteiler ist nicht lesbar oder beschreibbar. Wenn der Vorteiler dem TIMER0-Modul zugewiesen ist, löschen alle Befehle, die in das TMR0-Register schreiben, den Vorteiler null. Wenn der Vorteiler dem WDT zugewiesen ist, löscht der CLRWDT-Befehl sowohl den Vorteiler als auch den WDT.

8.2.4 UMSCHALTEN DES VORTEILERS ZWISCHEN TIMER0- UND WDT- MODULEN

Wenn der Vorteiler TIMER0 oder dem WDT zugewiesen ist, kann es beim Umschalten der Vorteiler zu einem unbeabsichtigten Geräte-Reset kommen. So ändern Sie den Prescaler von Beim Wechsel von TIMER0 auf WDT-Modul muss die unten gezeigte Anweisungssequenz ausgeführt werden.

Vorteiler ändern (TMR0-WDT)

CLRWDT			
CLR	TMR0		
SETB	OPTION_REG,PSA		; Wählen Sie WDT
CLRWDT			
LDIA	B'11111000'		
UND EIN	OPTION_REG		;Die untere 3-Position ist
ORIA	B'00000101'		0, die untere 3-Position ist 101, und der Rest bleibt unverändert
LD	OPTION_REG,A		

Um die Prescaler-Zuordnung vom WDT- zum TIMER0-Modul zu ändern, muss die folgende Anweisungssequenz ausgeführt werden.

Vorteiler ändern (WDT-TMR0)

CLRWDT			
LDIA	B'11110000'		
UND EIN	OPTION_REG		;Die unteren 4 Stellen
ORIA	B'00000101'		sind 0 ;Die unteren 4 Stellen sind 0101, der Rest bleibt unverändert
LD	OPTION_REG,A		

8.2.5 TIMER0- Unterbrechung

Ein TIMER0-Interrupt wird generiert, wenn das TMR0-Register von FFh auf 00h überläuft. Das T0IF-Interrupt-Flag-Bit des INTCON-Registers wird jedes Mal gesetzt, wenn das TMR0-Register überläuft, unabhängig davon, ob der TIMER0-Interrupt aktiviert ist oder nicht. Das T0IF-Bit muss in der Software gelöscht werden. Das Interrupt-Freigabebit TIMER0 ist das T0IE-Bit des INTCON-Registers.

Hinweis: Der TIMER0-Interrupt kann den Prozessor nicht aufwecken, da der Timer im SLEEP-Zustand deaktiviert ist.

8.3 Register bezogen auf TIMER0

Dem TMR0 sind zwei Register zugeordnet, der 8-Bit-Zeitgeber/Zähler (TMR0) und das programmierbare 8-Bit-Steuerregister (OPTION_REG).

TMR0 ist ein lesbarer und beschreibbarer 8-Bit-Timer/Zähler und OPTION_REG ist ein schreibgeschütztes 8-Bit-Register. Benutzer können den Wert von OPTION_REG ändern, um den Arbeitsmodus von TMR0 usw. zu ändern. Siehe 2.6 Anwendung des Vorteilerregisters

(OPTION_REG). 8-Bit-Zeitgeber/Zähler TMR0(01H)

01H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR0								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	X	X	X	X	X	X	X	X

OPTION_REG-Register (181H)

181H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPTION_REG lesen und	---	INTEGR	T0CS	T0SE	PSA	PS2	PS1	PS0
schreiben	---	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	---	1	1	1	1	1	1	1

Bit7 INTEDG

Bit6 deaktivieren: Flankenwahlbit unterbrechen.

1 = Steigende Flanke am INT-Pin löst Interrupt aus.

0 = Interrupt ausgelöst durch fallende Flanke am INT-Pin.

Bit5 T0CS: TMR0-Taktquellen-Auswahlbit.

1= Kante am T0CKI-Stift.

0= Interner Befehlszyklus (FOSC/4).

Bit4 T0SE: TIMER0 Taktquellenflanken-Auswahlbit.

1= Inkrementiert beim Übergang des Pins T0CKI von High auf Low.

0= Inkrementiert beim Übergang des T0CKI-Pins von niedrig nach hoch.

Bit3 PSA: Prescaler-Zuordnungsbits. Der Prescaler ist WDT zugeordnet. Der

1= Prescaler ist dem TIMER0-Modul zugeordnet.

0=

Bit2~Bit0 PS2-PS0: Vorbelegte Parameterkonfigurationsbits.

PS2	PS1	PS0	TMR0-Teilverhältnis	WDT-Teilverhältnis
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

9. Zeitzähler TIMER1

9.1 Überblick über TIMER1

Das TIMER1-Modul ist ein 16-Bit-Timer/Zähler mit den folgenden Merkmalen:

- 16-Bit-Timer-/Zählerregister (TMR1H:TMR1L) • Gate TIMER1 über T1G-Pin (Zählen aktivieren) • 3-Bit-Pre-scaler • Aufwachen bei Überlauf (nur im asynchronen Modus mit externer Uhr) • Triggerfunktion für spezielle Ereignisse (mit ECCP)
- Überlaufunterbrechung
- Zeitbasis für Capture/Compare-Funktion

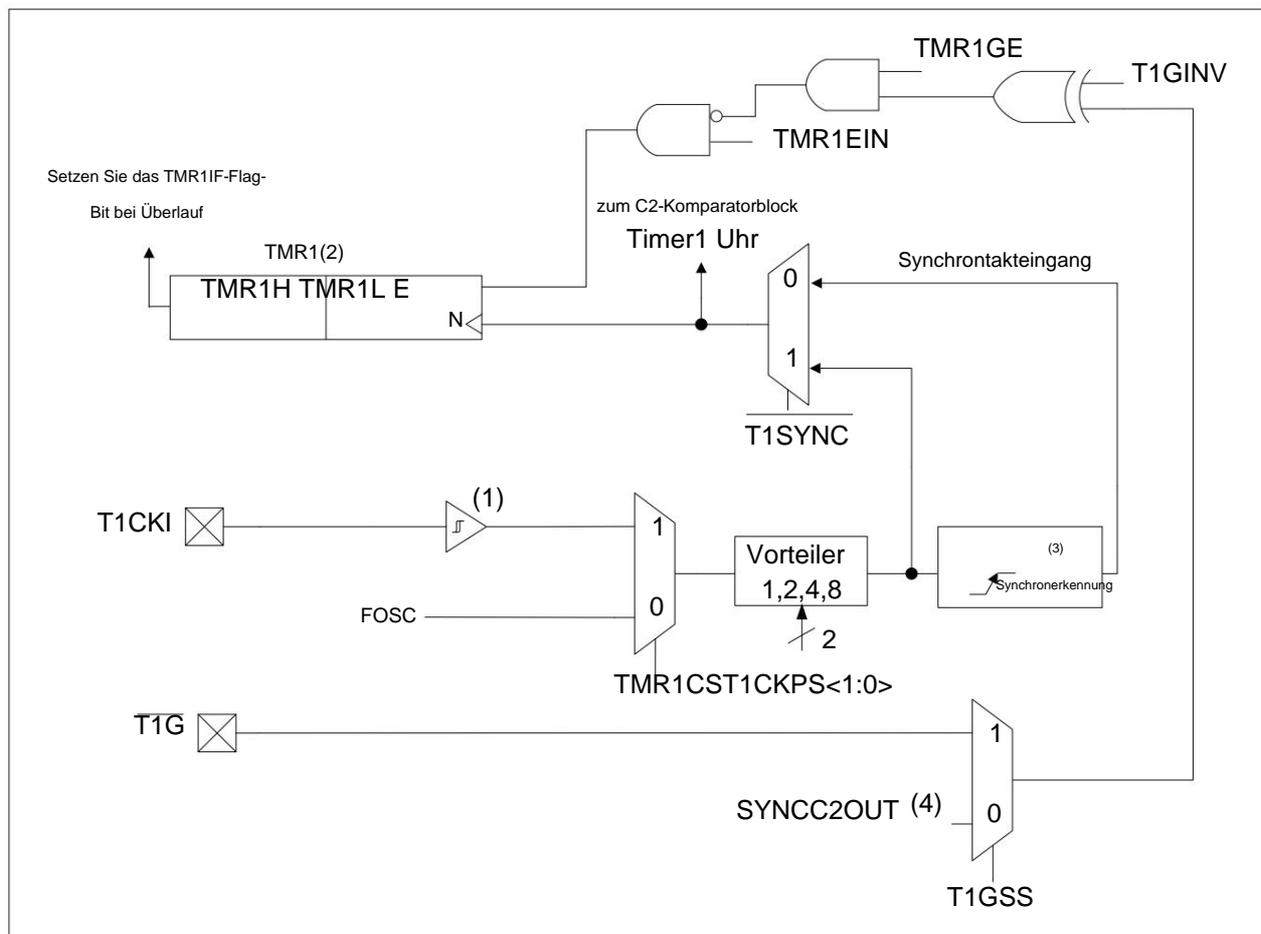


Abbildung 9-1: TIMER1-Strukturdiagramm

Notiz:

1. Der ST-Puffer befindet sich im Low-Power-Modus, wenn der LP-Oszillator verwendet wird, und im Hochgeschwindigkeitsmodus, wenn T1CKI verwendet wird.
2. Timer1-Register erhöht sich bei steigender Flanke. 3.
- Keine Synchronisation während des Ruhezustands. 4.
- SYNCC2OUT wird synchronisiert, wenn das C2SYNC-Bit des CM2CON1-Registers gesetzt ist.

9.2 Arbeitsprinzip von TIMER1

Das TIMER1-Modul ist ein 16-Bit-Aufwärtszähler, auf den über ein Registerpaar TMR1H:TMR1L zugegriffen wird. Beim Schreiben in TMR1L wird nur in das interne Pufferregister geschrieben, und beim Schreiben in TMR1H wird das interne Pufferregister in den TIMER1-Zähler geladen. Wenn Sie also in TMR1L und TMR1H schreiben, müssen Sie zuerst in das TMR1L-Register und dann in das TMR1H-Register schreiben. Wenn TIMER1 aktiv ist, werden die TMR1H:TMR1L-Register um ein Vielfaches von FOSC inkrementiert, das spezifische Vielfache wird durch den TIMER1-Prescaler bestimmt.

9.3 TIMER1 PRESCALER

TIMER1 verfügt über vier Prescaler-Auswahlmöglichkeiten, mit denen der Takteingang durch 1, 2, 4 oder 8 geteilt werden kann. Das T1CKPS-Bit des T1CON-Registers steuert den Prescaler-Zähler. Der Verteilerzähler kann nicht direkt gelesen oder geschrieben werden.

9.4 TIMER1- Unterbrechung

Ein Paar von TIMER1-Registern (TMR1H:TMR1L) wird auf 0000 h überlaufen, nachdem es auf FFFFh inkrementiert wurde. Wenn TIMER1 überläuft, Das TIMER1-Interrupt-Flag-Bit des PIR1-Registers wird gesetzt. Um diesen Überlauf-Interrupt zu aktivieren, sollte der Benutzer das folgende Bit setzen:

• TIMER1-Interrupt-Aktivierungsbit im PIE1-Register; • PEIE-Bit im INTCON-Register; • GIE-Bit im INTCON-Register; • Lösche das TMR1IF-Bit in der Interrupt-Service-Routine, um den Interrupt zu löschen.

Hinweis: Das TMR1H:TMR1L-Registerpaar und das TMR1IF-Bit sollten gelöscht werden, bevor dieser Interrupt erneut aktiviert wird. Aufgrund des Timings während des Schlafs

Der Timer ist ausgeschaltet, sodass der TIMER1-Interrupt den Prozessor nicht aufwecken kann.

9.5 TIMER1- bezogene Register

TIMER1 wird hauptsächlich von drei RAMs gesteuert: TMR1-Steuerregister T1CON, Datenregister TMR1L, TMR1H. Datenregister

Wenn Sie einen Wert zuweisen, müssen Sie zuerst dem niedrigen Bit TMR1L einen Wert zuweisen und dann TMR1H einen Wert zuweisen.

TIMER1-Daten-Low-Register TMR1L(0EH)

0EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR1L								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	X	X	X	X	X	X	X	X

TIMER1-Daten-High-Register TMR1H(0FH)

0FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR1H								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	X	X	X	X	X	X	X	X

TIMER1 Steuerregister T1CON(10H)

10 Uhr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CON T1GINV TMR1GE T1CKPS1 T1CKPS0					---	---	TMR1CS TMR1EIN	
Rücksetzwert	R/W	R/W	R/W	R/W	---	---	R/W	R/W
lesen und schreiben	0	0	0	0	---	---	0	0

Bit7 T1GINV: TIMER1 Torsignalpolaritätsbit (1);
 1 = Gate-Signal von TIMER1 ist aktiv hoch (TIMER1 zählt, wenn das Gate-Signal hoch ist);
 0 = Gate-Signal von TIMER1 ist aktiv niedrig (TIMER1 zählt, wenn das Gate-Signal niedrig ist).

Bit6 TMR1GE: TIMER1-Gate-Freigabebit (2). Wenn
 TMR1ON = 0: Dieses Bit wird ignoriert.
 Wenn TMR1ON = 1: 1 = TIMER1-Zählung
 wird durch TIMER1-Torfunktion gesteuert; 0 = TIMER1
 zählt immer.

Bit5–Bit4 T1CKPS<1:0>: TIMER1-Eingangstakt-Prescaler-Auswahlbits; 11
 = 1:8-Prescaler; 10 = 1:4-Prescaler; 01 = 1:2-
 Prescaler; 00 = 1:1-Prescaler. TMR1CS
 deaktivieren: TIMER1-Taktquellen-Auswahlbit;

Bit3–Bit2

Bit1
 1 = Externe Taktquelle vom T1CKI-Pin (ansteigende Flanke
 getriggert); 0 = Interne Taktquelle (FOSC/4).

Bit0 TMR1ON: TIMER1-Aktivierungsbit;
 1 = TIMER1 aktivieren; 0
 = TIMER1 deaktivieren.

Anmerkungen: 1. Das T1GINV-Bit invertiert den Logikpegel des Gate-Signals TIMER1, unabhängig von der Quelle des Gate-Signals.
 2. Das TMR1GE-Bit muss auf 1 gesetzt werden, um den T1G-Pin als TIMER1-Gate-Quelle zu verwenden.

10. Zeitzähler TIMER2

10.1 Überblick über TIMER2

Das TIMER2-Modul ist ein 8-Bit-Timer/Zähler mit den folgenden Merkmalen:

- 8-Bit-Timer-Register (TMR2);
- 8-Bit-Periodenregister (PR2);
- Interrupt bei TMR2-PR2-Übereinstimmung;
- Software-programmierbarer Prescaler (1 : 1, 1:4 und 1:16);
- Per Software programmierbares Postscaler-Verhältnis (1:1 bis 1:16).

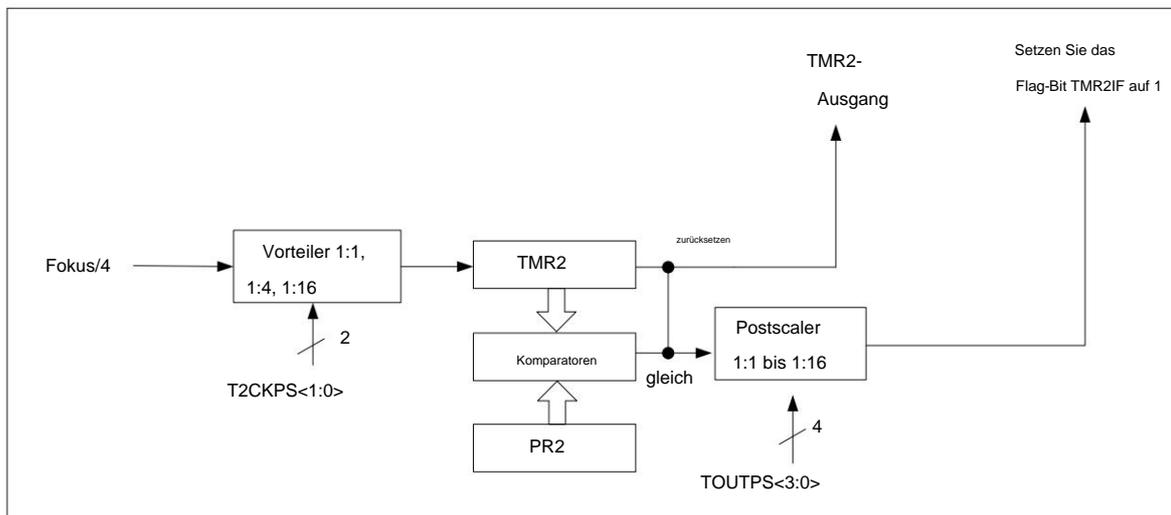


ABBILDUNG 10-1: TIMER2 BLOCKDIAGRAMM

10.2 Wie TIMER2 funktioniert

Der Takteingang zum TIMER2-Modul ist der Systemanweisungstakt (FOSC/4). Der Takt wird in den Vorteiler TIMER2 eingegeben, der die folgenden Frequenzunterteilungen hat
Verfügbares Verhältnis: 1:1, 1:4 oder 1:16. Die Ausgabe des Prescalers wird dann verwendet, um das TMR2-Register zu inkrementieren.

Die Werte von TMR2 und PR2 werden kontinuierlich verglichen, um festzustellen, wann sie übereinstimmen. TMR2 erhöht sich von 00h, bis es mit dem Wert in PR2 übereinstimmt.
Wenn eine Übereinstimmung auftritt, treten die folgenden beiden

Ereignisse auf: \bar{y} TMR2 wird beim nächsten Inkrementierungszyklus auf 00h

zurückgesetzt; \bar{y} TIMER2-Postscaler wird inkrementiert.

Die Übereinstimmungsausgabe der TIMER2- und PR2-Komparatoren wird dann in den TIMER2-Postscaler eingegeben. Postscaler mit 1:1 bis 1:16 Prescaler
Als verfügbar. Der Ausgang des TIMER2-Postscalers wird verwendet, um das TMR2IF-Interrupt-Flag-Bit des PIR1-Registers zu setzen.

Die TMR2- und PR2-Register sind sowohl lesbar als auch schreibbar. Bei jedem Zurücksetzen wird das TMR2-Register auf 00h und das PR2-Register auf FFh gesetzt. TIMER2
wird aktiviert, indem das TMR2ON-Bit des T2CON-Registers gesetzt wird; TIMER2 wird deaktiviert, indem das TMR2ON-Bit gelöscht wird.

Der TIMER2-Prescaler wird durch das T2CKPS-Bit des T2CON-Registers gesteuert, der TIMER2-Postscaler wird durch das TOUTPS-Bit des T2CON-Registers gesteuert
Bit-Steuerung.

Die Prescaler- und Postscaler-Zähler werden gelöscht durch: \bar{y} Ein Schreiben in das

TMR2-Register \bar{y} Ein Schreiben in das T2CON-Register \bar{y} Ein beliebiges

Zurücksetzen des Geräts (Einschalt-Reset, Watchdog-Press-Reset).

Hinweis: Das Schreiben von T2CON löscht TMR2 nicht.

10.3 TIMER2- bezogene Register

Es gibt 2 Register, die sich auf TIMER2 beziehen, nämlich den Datenspeicher TMR2 und das Steuerregister T2CON.

TIMER2 Datenregister TMR2(11H)

11H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR2								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	X	X	X	X	X	X	X	X

TIMER2 Steuerregister T2CON(12H)

12H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON ----	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	
lesen und Schreiben ----		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen ----		0	0	0	0	0	0	0

Bit7

Nicht implementiert.

Bit6~Bit3

gelesen als 0 TOUTPS<3:0>: TIMER2 gibt Postscaler-Auswahlbits aus.

0000= 1:1 Post-Skalierungsverhältnis

0001= 1:2 Post-Skalierungsverhältnis

0010= 1:3 Post-Skalierungsverhältnis

0011= 1:4 Post-Skalierungsverhältnis

0100= 1:5 Post-Skalierungsverhältnis

0101= 1:6 Post-Scale-Verhältnis; 0110=

1:7 Post-Scale-Frequenzverhältnis;

0111= 1:8 Post-Scale-Frequenzverhältnis;

1000= 1:9 Post-Scale-Frequenzverhältnis;

1001= 1:10 Post-Scale-Verhältnis.

Skalierungsfrequenzverhältnis; 1010 =

1:11 Post-Skalierungsverhältnis; 1011 =

1:12 Post-Skalierungsverhältnis; 1100 =

1:13 Post-Skalierungsverhältnis; 1101 =

1:14 Post-Skalierungsverhältnis; 1110 =

1:15 Post -Skalierungsverhältnis; 1111= 1:16 Postscaler.

Bit2

TMR2ON: TIMER2-Aktivierungsbit;

1 = TIMER2 aktivieren; 0

= TIMER2 deaktivieren.

Bit1~Bit0

T2CKPS<1:0>: TIMER2-Takt-Prescaler-Auswahlbits; 00 =

Prescaler-Wert ist 1; 01 = Prescaler-Wert ist

4; 1x = Prescaler-Wert ist 16.

11. Analog-Digital-Wandlung (ADC)

11.1 ADC- Übersicht

Ein Analog-Digital-Wandler (ADC) wandelt ein analoges Eingangssignal in eine 10-Bit-Binärzahl um, die das Signal darstellt. Die vom Gerät verwendeten analogen Eingangskanäle teilen sich eine Sample-and-Hold-Schaltung. Der Ausgang der Sample-and-Hold-Schaltung ist mit dem Eingang des Analog-Digital-Wandlers verbunden. Der Analog-Digital-Wandler erzeugt ein binäres 10-Bit-Ergebnis unter Verwendung des sukzessiven Approximationsverfahrens und speichert das Ergebnis in den ADC-Ergebnisregistern (ADRESH und ADRESL).

Die ADC-Referenzspannung wird immer intern erzeugt. Der ADC kann einen Interrupt erzeugen, nachdem eine Konvertierung abgeschlossen ist.

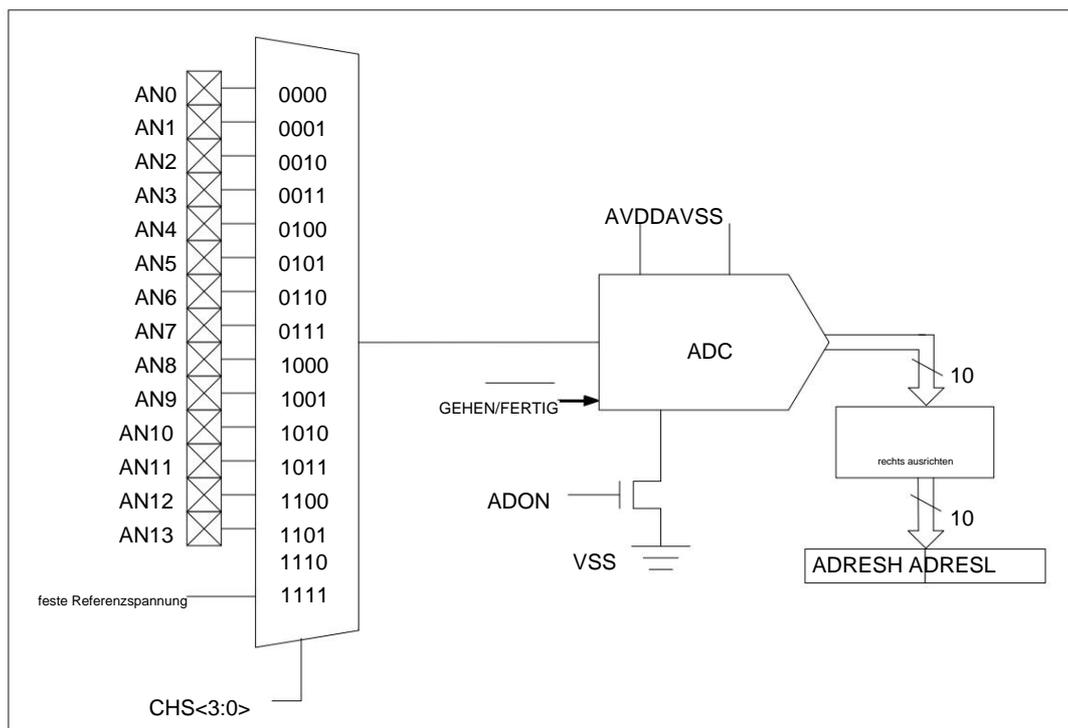


Abbildung 11-1: ADC-Blockdiagramm

11.2 ADC- Konfiguration

Bei der Konfiguration und Verwendung von ADC müssen die folgenden

Faktoren berücksichtigt werden: • Portkonfiguration, • Kanalauswahl, • ADC-

Umwandlungstaktquelle, • Unterbrechungssteuerung.

11.2.1 Portkonfiguration

ADCs können sowohl analoge als auch digitale Signale umwandeln. Beim Umwandeln analoger Signale sollten die I/O-Pins als analoge Eingänge konfiguriert werden, indem die entsprechenden TRIS- und ANSEL-Bits gesetzt werden. Weitere Informationen finden Sie im entsprechenden Anschlusskapitel.

Hinweis: Das Anlegen analoger Spannungen an Pins, die als digitale Eingänge definiert sind, kann zu Überstrom in den Eingangspuffern führen.

11.2.2 Kanalauswahl

Welcher Kanal mit der Sample-and-Hold-Schaltung verbunden ist, wird durch das CHS-Bit des ADCON0-Registers

bestimmt. Wenn der Kanal geändert wird, ist eine Verzögerung erforderlich, bevor die nächste Umwandlung beginnt. Weitere Informationen finden Sie in Abschnitt 11.3 „ADC-Betrieb“.

11.2.3 ADC -Referenzspannung

Die Referenzspannung des ADC wird immer von VDD und GND des Chips bereitgestellt.

11.2.4 Wandlungsuhr

Die Konvertierungstaktquelle kann durch Software-Setzen des ADCS-Bits im ADCON0-Register ausgewählt werden. Es stehen 4 mögliche Taktfrequenzen zur Auswahl

wählen:

- FOSC/8
- FOSC/16
- FOSC/32 • FRC
(dedizierter interner Oszillator)

Die Zeit zum Abschließen einer Ein-Bit-Umwandlung wird als TAD definiert. Eine vollständige 10-Bit-Umwandlung erfordert 41 TAD-Zyklen. Um das

richtige Wandlungsergebnis zu erhalten, muss die entsprechende TAD-Spezifikation eingehalten werden, die folgende Tabelle ist ein Beispiel für die richtige Auswahl des ADC-Taktes.

Hinweis: Sofern nicht FRC verwendet wird, ändert jede Änderung der Systemtaktfrequenz die Frequenz des ADC-Taktes, wodurch das Ergebnis der ADC-Konvertierung beeinflusst wird.

Negative Auswirkung.

Die Beziehung zwischen der ADC-Taktperiode (TAD) und der Gerätebetriebsfrequenz (VDD = 5,0 V) ADC-

Taktperiode		Gerätefrequenz		
ADC-Taktquelle	ADCS<1:0>	8MHz	4MHz	1MHz
Fokus/8	00	49,0 μ s	98,0 μ s	392,0 μ s
Fokus/16	01	98,0 μ s	196,0 μ s	784,0 μ s
Fokus/32	10	196,0 μ s	392,0 μ s	1,5 ms
FRC	11	1-3ms	1-3ms	1-3ms

Legende: Werte in schattierten Zellen werden nicht empfohlen.

11.2.5 ADC- Unterbrechung

Das ADC-Modul ermöglicht die Erzeugung eines Interrupts, nachdem die Analog-Digital-Wandlung abgeschlossen ist. Das ADC-Interrupt-Flag-Bit ist das ADIF-Bit im PIR1-Register. Das ADC-Interrupt-Freigabebit ist das ADIE-Bit im PIE1-Register. Das ADIF-Bit muss per Software gelöscht werden. Das ADIF-Bit wird nach jeder Konvertierung auf 1 gesetzt, unabhängig davon, ob der ADC-Interrupt aktiviert ist oder nicht.

Unterbrechungen können unabhängig davon generiert werden, ob sich das Gerät im Aktiv- oder Schlafmodus befindet. Wenn sich das Gerät im Schlafmodus befindet, kann dieser Interrupt das Gerät aufwecken. Beim Aufwecken des Geräts aus dem Sleep-Modus wird immer die nächste Anweisung nach der STOP-Anweisung ausgeführt. Globale Interrupts müssen deaktiviert werden, wenn der Benutzer versucht, das Gerät aus dem Ruhezustand zu wecken und die Codeausführung sequenziell fortzusetzen. Wenn der globale Interrupt aktiviert ist, springt das Programm zur Interrupt-Service-Routine.

11.3 ADC- Arbeitsprinzip

11.3.1 Boot-Übergang

Um das ADC-Modul zu aktivieren, muss das ADON-Bit des ADCON0-Registers gesetzt werden und das GO/DONE-Bit des ADCON0-Registers muss auf gesetzt werden

Starten Sie die Analog-Digital-Wandlung.

Hinweis: Das GO/DONE-Bit kann nicht mit demselben Befehl gesetzt werden, der das A/D-Modul einschaltet.

11.3.2 Vollständige Konvertierung

Wenn die Konvertierung abgeschlossen ist, wird das ADC-Modul: \bar{y} das GO/DONE-Bit löschen; \bar{y} das ADIF-Flag-Bit setzen; \bar{y} die ADRESH:ADRESL-Register mit dem neuen Ergebnis der Konvertierung aktualisieren.

11.3.3 Beendigungskonvertierung

Wenn es notwendig ist, die Konvertierung abubrechen, bevor sie abgeschlossen ist, kann das GO/DONE-Bit in der Software gelöscht werden. wird nicht mit den Ergebnissen von Analog-Digital-Wandlungen aktualisiert, die noch nicht abgeschlossen sind ADRESH:ADRESL-Register. Daher halten die ADRESH:ADRESL-Register den Wert von der letzten Konvertierung. Außerdem die A/D wiederum Nach Beendigung des Austauschs kann die nächste Erfassung erst nach einer Verzögerung von 2 TADs gestartet werden. Nach der Verzögerung beginnt das Eingangssignal des ausgewählten Kanals automatisch zu sammeln.

Hinweis: Ein Geräte-Reset zwingt alle Register in ihren Reset-Zustand. Daher schaltet ein Reset das ADC-Modul aus und beendet alle anstehenden Konvertierungen.

11.3.4 Funktionsweise von ADC im Schlafmodus

Das ADC-Modul kann im Schlafmodus arbeiten. Dieser Vorgang erfordert die Einstellung der ADC-Taktquelle auf die FRC - Option. Wenn die FRC-Taktquelle ausgewählt ist, wartet der ADC einen weiteren Befehlszyklus, bevor er mit der Umwandlung beginnt. Dadurch kann eine STOP-Anweisung ausgeführt werden, um das Systemrauschen während Übergängen zu reduzieren. Wenn aktiviert, bewirkt der ADC-Interrupt, dass das Gerät aus dem Schlafmodus aufwacht, wenn die Konvertierung abgeschlossen ist. Wenn der ADC-Interrupt deaktiviert ist, wird das ADC-Modul nach der Konvertierung abgeschaltet, selbst wenn das ADON-Bit gesetzt bleibt. Wenn die ADC-Taktquelle nicht FRC ist, wird die Ausführung des STOP-Befehls die aktuelle Umwandlung abbrechen und das A/D-Modul ausschalten, obwohl das ADON-Bit auf 1 gesetzt bleibt.

11.3.5 A/D- Wandlungsschritte

Die folgenden Schritte geben ein Beispiel für die Analog-Digital-Wandlung mit

ADC: 1. Portkonfiguration:

• den Pin-Ausgangstreiber deaktivieren (siehe TRIS-Register) • den Pin als analogen Eingangspin konfigurieren. 2. Konfigurieren Sie das ADC-Modul:

• Wählen Sie den ADC-Wandlungstakt aus; • Wählen Sie den ADC-Eingangskanal aus; • Wählen Sie das Format des Ergebnisses aus; • Starten Sie das ADC-Modul.

3. Konfigurieren Sie den ADC-Interrupt (optional):

• ADC-Interrupt-Flag löschen; • ADC-Interrupt aktivieren; • Peripheral-Interrupt aktivieren; • Globalen Interrupt aktivieren. 4. Warten Sie

die erforderliche Erfassungszeit ab. 5. Stellen

Sie GO/DONE ein, um die Konvertierung zu starten.

6. Warten Sie mit einer der folgenden Methoden auf das

Ende der ADC-Wandlung: • GO/DONE-Bit abfragen • auf ADC-

Interrupt warten (Interrupt aktivieren). 7. Lesen Sie das ADC-

Ergebnis ab. 8. Lösche das ADC-Interrupt-Flag-Bit

(erforderlich, wenn Interrupts aktiviert sind).

Hinweis: Globale Interrupts müssen deaktiviert werden, wenn der Benutzer versucht, die sequenzielle Codeausführung fortzusetzen, nachdem das Gerät aus dem Schlafmodus geweckt wurde.

Beispiel: AD-Wandlung

LDIA	B'10000000'	
LD	ADCON1,A	
SETB	TRISA,0	;PORTA.0 als Eingangsport einstellen;
SETB	ANSEL, 0	PORTA.0 als analogen Port einstellen
LDIA	B'11000001'	
LD	ADCON0,A	
FORDERUNG	VERZÖGERUNG	;Eine Weile verzögern
SETB	ADCON0,GO	
SZB	ADCON0,GO	;Warten Sie auf das Ende der AD-Konvertierung
JP	\$_-1	
LD	A, Adresse	;Das hohe Bit des AD-Umwandlungsergebnisses speichern
LD	ERGBNIS, A	
LD	A, ADRESL	;Niedriges Bit des AD-Umwandlungsergebnisses speichern
LD	ERGBNIS,A	

11.4 ADC- bezogener RAM

Es gibt hauptsächlich 3 RAMs, die sich auf die AD-Umwandlung beziehen, nämlich das Steuerregister ADCON, das Datenregister ADRESH und ADRESL.

AD-Steuerregister ADCON(1FH)

1FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADCS1	ADCS0		CHS3	CHS2	CHS1	CHS0 GO/FERTIG	ADON
Rücksetzwert	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
lesen und schreiben	0	0	0	0	0	0	0	0

Bit7~Bit6 ADCS<1:0>: A/D-Wandlungstaktauswahlbits.
 00 = FOSC/8
 01 = FOSC/16
 10 = FOSC/32
 11 = FRC (Die Taktfrequenz beträgt bis zu 500 KHz, erzeugt durch den dedizierten internen

Bit5~Bit2 Oszillator) CHS<3:0>: Analoge Kanalauswahlbits.

0000=AN0
 0001=AN1
 0010=AN2
 0011=AN3
 0100=AN4
 0101=AN5
 0110= AN6
 0111= AN7
 1000=AN8
 1001= AN9
 1010=AN10
 1011= ----
 1100= ----
 1101= ----
 1110 = CVREF

1111= Feste Referenzspannung (0,6 V feste Referenzspannung)

Bit1 GO/DONE: A/D-Wandlungsstatusbit.

1 = A/D-Wandlung läuft. Das Setzen dieses Bits auf 1 startet eine A/D-Wandlung. Dieses Bit wird von der Hardware automatisch auf 0 gelöscht, wenn die A/D-Wandlung abgeschlossen ist.

0 = A/D-Wandlung abgeschlossen oder nicht im Gange.

Bit0 ADON: ADC-Aktivierungsbit. 1 =

ADC aktivieren; 0 = ADC

deaktivieren, verbraucht keinen Betriebsstrom.

AD-Datenregister hoch ADRESH (9FH)

9FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESSE	----	----	----	----	----	----	ADRES9 ADRES8	
lesen und Schreiben	----	----	----	----	----	----	R	R
Wert zurücksetzen	----	----	----	----	----	----	X	X

Bit7~Bit0

ADRES<9:8>: ADC-Ergebnisregisterbits.

Die oberen 2 Bits des 10-Bit-Umwandlungsergebnisses.

AD-Datenregister niedrig ADRESL (9EH)

9EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESL ADRES7 ADRES6 ADRES5 ADRES4 ADRES3 ADRES2 ADRES1 ADRES0								
Rücksetzvent	R	R	R	R	R	R	R	R
lesen und schreiben	X	X	X	X	X	X	X	X

Bit7~Bit0

ADRES<7:0>: ADC-Ergebnisregisterbits.

Die unteren 8 Bits des 10-Bit-Umwandlungsergebnisses.

12. PWM -Modul

12.1 PWM- Eigenschaften

- 8-Bit-Präzision •
- Ausgangspolarität wählbar •
- Zählfrequenz wählbar • Mitten-
oder kantenbündiger Ausgangsmodus wählbar

CMS89F52x hat zwei eingebaute 8-Bit-PWM-Module. Das PWM-Modul kann Pulsweitenmodulations-Wellenformen erzeugen, deren Periode und Arbeitszyklus jeweils angepasst werden können.

Die von den beiden PWMs erzeugten Wellenformen werden jeweils von RA0 und RA1 ausgegeben.

12.2 PWM- bezogene Register

Die mit der PWM-Funktion verbundenen Register sind: Steuerregister PWM0CR, PWM1CR, Periodeneinstellregister PWM0PR, PWM1PR; Arbeitszyklus-Einstellregister PWM0DR, PWM1DR.

PWM0 Tastverhältnis Einstellregister PWM0DR(18H)

18 Uhr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0DR	PWM0DR[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

PWM0-Periodeneinstellregister PWM0PR(19H)

19H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0PR	PWM0PR[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

PWM0-Steuerregister PWM0CR(1AH)

1AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0CR	PWM0EN	PWM0MOD	PWM0POL			PWM0CKS[3:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7 PWM0EN: PWM0-Aktivierungsbit 0:
 Deaktivieren 1: Aktivieren
 (PWM0-Port wird als Ausgang eingestellt und PWM0-Wellenform wird ausgegeben)

Bit6 PWM0MOD: PWM0-Modusauswahlbit 0:
 Normalmodus 1: Back-to-Back-Modus deaktiviert

Bit5

Bit4 PWM0POL: PWM0-Ausgangspolaritätsauswahlbit
 0: Normale Ausgabe
 1: Invertierte Ausgabe

Bit3~Bit0 PWM0CKS[3:0]: PWM0-Taktfrequenzauswahlbits

0000: Fosc	1000: Fosc/256
0001: Fosc/2	1001: Fosc/512
0010: Fosc/4	1010: Fosc/1024
0011: Fosc/8	1011: Fosc/2048
0100: Fosc/16	1100: Fosc/4096
0101: Fosc/32	1101: Fosc/8192
0110: Fosc/64	111x: Fosc/8192
0111: Fosc/128	

PWM1 Tastverhältnis Einstellregister PWM1DR(1BH)

1BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DR	PWM1DR[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

PWM1-Periodeneinstellungsregister PWM1PR(1CH)

1CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1PR	PWM1PR[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

PWM1-Steuerregister PWM1CR(1DH)

1DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1CR	PWM1EN	PWM1MOD	--- PWM1POL		PWM1CKS[3:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7	PWM1EN: PWM1 aktivieren Bit 0: Deaktivieren 1: Aktivieren (PWM1-Port ist als Ausgang eingestellt und gibt PWM1-Wellenform aus)
Bit6	PWM1MOD: PWM1-Modusauswahlbit 0: Normalmodus 1: Back-to-Back-Modus deaktiviert
Bit5	
Bit4	PWM1POL: PWM1-Ausgangspolaritätsauswahlbit 0: Normale Ausgabe 1: Invertierte Ausgabe
Bit3~Bit0	PWM1CKS[3:0]: PWM1-Taktfrequenzauswahlbits 0000: Fosc 0001: Fosc/2 0010: Fosc/4 0011: Fosc/8 0100: Fosc/16 0101: Fosc/32 0110: Fosc/64 0111: Fosc/128 1000: Fosc/256 1001: Fosc/512 1010: Fosc/1024 1011: Fosc/2048 1100: Fosc/4096 1101: Fosc/8192 111x: Fosc/8192

- Zugehörige Parameter von PWM0/PWM1: n=0, 1
- PWMn-Periode: $T_{pwmp} = (PWMnPR[7:0] + 1) \times 2 \times PWMnCKS[3:0] \times T_{sys}$ (die längste Periode: 262Ms@Fosc=8MHz)
- PWMn-Hochpulszeit: $T_{pwmh} = PWMnDR[7:0] \times 2 \times PWMnCKS[3:0] \times T_{sys}$
- PWMn-Tastverhältnis: $T_{pwmh} / T_{pwmp} = PWMnDR[7:0] / (PWMnPR[7:0] + 1)$ (Einstellbereich: 0%-100%, max. Genauigkeit 1/256)
- Arbeitszyklus im PWMn-Back-to-Back-Modus: $(2 \times PWMnDR[7:0] + 1) / (PWMnPR[7:0] + 1)$ - Wenn der Wert von PWMnDR oder PWMnPR 0 ist, das Tastverhältnis : 0 %; wenn PWMnDR[7:0] \neq PWMnPR[7:0] + 1, dann berücksichtigen
Die Leerquote beträgt: 100 %.

13. Erfassungsmodul CCP

13.1 Erfassung des CCP- Registers

Captures sind Peripheriegeräte, die es dem Benutzer ermöglichen, verschiedene Ereignisse zu timen und zu steuern. Im Erfassungsmodus kann das Peripheriegerät die Dauer eines Ereignisses messen. Bei KPCh

Bei Verwendung im Aufnahmemodus ist der Timer TIMER1 erforderlich.

CCP-Steuerregister CCPCON(190H)

190H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCPCON	CCPEN	----	----	CCPIS	CCPES	CCPM2	CCPM1	CCPM0
lesen und Schreiben	R/W	----	----		R/W	R/W	R/W	R/W
Wert zurücksetzen	0	----	----	0	0	0	0	0

Bit7 CCPEN: Bit zum Aktivieren der Erfassungsfunktion;

0 = Erfassungsfunktion deaktivieren; 1

= Erfassungsfunktion aktivieren. CCPIS

Bit6–Bit5 deaktivieren: Bit zur Auswahl der Taktquelle

Bit4 erfassen;

0 = Taktquelle ist vom Eingang des RA2-Anschlusses;

1 = Taktquelle ist vom Ausgang des synchronen Komparators.

Bit3 CCPES: Taktflankenauswahlbit erfassen;

0 = Erfassung erfolgt bei fallender Taktflanke,

1 = Erfassung erfolgt bei steigender Taktflanke.

Bit2–Bit0 CCPM<2:0>:

Erfassungsmodusauswahlbits;

000 = Erfassung alle 1 Takt; 001 =

Erfassung alle 2 Takte; 010 =

Erfassung alle 4 Takte; 011 =

Erfassung alle 8 Takte; 100 =

Erfassung tritt alle 16 Takte auf;

101 = Erfassung erfolgt alle 32

Takte; 110 = Erfassung erfolgt alle 64 Takte; 111 = Erfassung erfolgt alle 128 Takte.

13.2 Aufnahmemodus

Wenn im Erfassungsmodus ein Ereignis am entsprechenden CCP-Pin auftritt, erfasst das CCPRH:CCPRL-Registerpaar den 16-Bit-Wert des TMR1-Registers.

Wert.

Der Ereignistyp wird durch die Modusauswahlbits CCPM2:CCPM0 (CCPCON<2:0>) ausgewählt. Das Unterbrechungsanforderungs-Flagbit CCPIF im PIR1-Register wird gesetzt, wenn eine Erfassung auftritt, es muss in der Software gelöscht werden. Wenn eine weitere Erfassung erfolgt, bevor der Wert im CCPRH- und CCPRL-Paar gelesen wird, wird der zuvor erfasste Wert durch den neuen erfassten Wert überschrieben (siehe Abbildung 13-1).

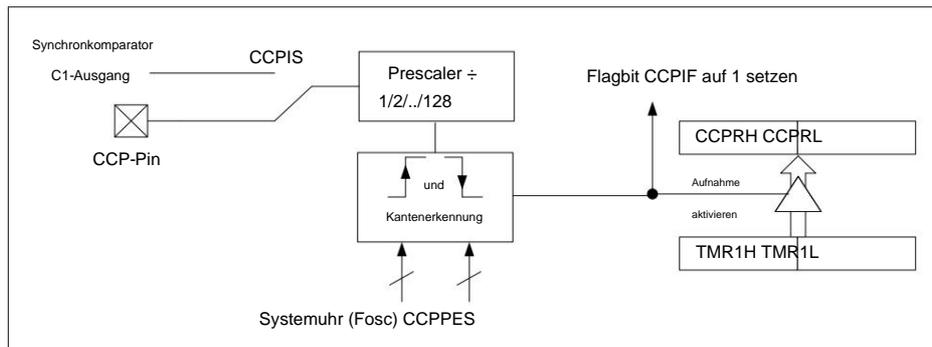


Abbildung 13-1: Arbeitsblockdiagramm des Capture-Modus

13.2.1 CCP- Pin-Konfiguration

Im Capture-Modus sollte der entsprechende CCP-Pin als Eingang konfiguriert werden, indem das entsprechende TRIS-Steuersbit gesetzt wird.

Hinweis: Wenn der CCP-Pin als Ausgang konfiguriert ist, kann ein Schreiben auf den Port ein Capture-Ereignis verursachen.

13.2.2 Auswahl des TIMER1-Modus

TIMER1 muss im Timer-Modus oder Synchronzähler-Modus laufen, damit das CCP-Modul die Erfassungsfunktion verwenden kann. Erfassungsvorgänge sind im asynchronen Zählermodus nicht verfügbar.

13.2.3 Software-Interrupts

Falsche Erfassungsunterbrechungen können erzeugt werden, wenn der Erfassungsmodus geändert wird. Der Benutzer sollte das CCPIE-Interrupt-Freigabebit im PIE1-Register gelöscht halten um falsche Interrupts zu vermeiden. Das Interrupt-Flag-Bit CCPIF im PIR1-Register sollte auch nach jeder Änderung des Betriebsmodus gelöscht werden.

14. MSSP-Modul (Master Synchronous Serial Port).

14.1 Überblick über das Master Control SSP (MSSP)-Modul

Das Modul Master Synchronous Serial Port (Master Synchronous Serial Port, MSSP) wird zur Kommunikation mit anderen Peripheriegeräten oder Mikrocontrollern verwendet serielle Schnittstelle. Diese Peripheriegeräte können serielles EEPROM, Schieberegister, Anzeigetreiber oder A/D-Wandler usw. sein.

Das MSSP-Modul hat die folgenden zwei

Betriebsmodi: • Serial Peripheral Interface (SPI).

• I2C.

- Vollständiger Master-Modus.
- Slave-Modus (unterstützt Aufrufe von Broadcast-Adressen).

Die I2C -Schnittstelle unterstützt die folgenden

Modi in Hardware: • Master-Modus. • Multi-Host-

Modus. • Slave-Modus.

14.2 SPI- Modus

Der SPI-Modus ermöglicht das gleichzeitige Senden und Empfangen von 8-Bit-Daten gleichzeitig. Alle 4 Modi von SPI werden unterstützt. Die Kommunikation erfolgt normalerweise über die folgenden drei Pins: • Serial Data Out (SDO) – RA3/SDO • Serial Data In (SDI) – RA5/SDI/SDA • Serial Clock (SCK) – RA6/SDO SCK/SCL
Zusätzlich a Der vierte Pin kann in beiden Slave-Betriebsarten verwendet werden: • Slave Select (SS) - RE1/AN10/SS

14.2.1 SPI- bezogene Register**SSPSTAT: SSP-Statusregister (193H)**

193H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPSTAT	SMP	CKE	---	---	---	---	---	---
Rücksetzwert	R/W	R/W	---	---	---	---	---	---
lesen und schreiben	0	0	---	---	---	---	---	---

Bit7

SMP: Abtastbit.

SPI-Mastermodus:

1 = Eingangsdaten werden am Ende der Datenausgabezeit abgetastet,

0 = Eingangsdaten werden in der Mitte der Datenausgabezeit abgetastet.

SPI-Slave-Modus: Bei Verwendung des SPI-Slave-Modus muss SMP gelöscht werden.

Bit 6

CKE: SPI-Taktflankenauswahlbit.

CKP=0

1 = Daten bei steigender Flanke des SCK-Pins übertragen

0 = Daten bei fallender Flanke des SCK-Pins übertragen.

KP = 1

1 = Daten werden bei der fallenden Flanke des SCK-Pins übertragen;

0 = Daten werden bei der steigenden Flanke des SCK-Pins übertragen.

Bit5–Bit0 werden im SPI-Modus nicht verwendet.

SSPCON: SSP-Steuerregister 1 (194H)

194H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPCON	---	SSPOV	SSPEN	CKP	SSPM3 SSPM2 SSPM1 SSPM0			
Rücksetzwert	---	R/W	R/W	R/W	R/W	R/W	R/W	R/W
lesen und schreiben	---	0	0	0	0	0	0	0

Bit7

Bit6

Unbenutztes SSPOV: Überlaufanzeigebit empfangen.

1 = Ein neues Byte wurde empfangen, während SSPBUF noch die vorherigen Daten enthält. Daten im SSPSR gehen verloren, wenn ein Überlauf auftritt. Überlauf tritt nur im Slave-Modus auf. Im Slave-Modus muss der Benutzer SSPBUF lesen, um einen Überlauf zu vermeiden, selbst wenn nur Daten übertragen werden. Im Master-Modus ist das Overflow-Bit nicht auf 1 gesetzt, da jedes Mal, wenn neue Daten empfangen oder gesendet werden, dies durch Schreiben in das SSPBUF-Register initiiert werden muss (dieses Bit muss per Software gelöscht werden). 0 = kein Überlauf.

Bit5

SSPEN: Aktivierungsbit für synchrone serielle Schnittstelle.

1 = Aktiviert den seriellen Port und konfiguriert SCK, SDO, SDI und SS als serielle Port-Pins. 0 = Deaktiviert den seriellen Port und konfiguriert diese Pins als E/A-Port-Pins.

Bit4

CKP: Taktpolaritätsauswahlbit.

1 = Ruhezustand der Uhr ist hoch.
0 = Ruhezustand der Uhr ist niedrig.

Bit3~Bit0

SSPM<3:0>: Auswahlbits für den Modus des synchronen seriellen Anschlusses;

0000 = SPI-Master-Modus, Uhr = FOSC/4; 0001 = SPI-Master-Modus, Uhr = FOSC/16; 0010 = SPI-Master-Modus, Uhr = FOSC/64; 0011 = SPI-Master-Modus, Uhr = TMR2-Ausgang /2; 0100 = SPI-Slave-Modus, Uhr = SCK-Pin, SS-Pin-Steuerung aktivieren; 0101 = SPI-Slave-Modus, Uhr = SCK-Pin, SS-Pin-Steuerung deaktivieren, SS kann als I/O-Pin-Pin verwendet werden; 0110 = I2C-Slave-Modus, 7-Bit-Adresse; 0111 = I2C-Slave-Modus, 10-Bit-Adresse; 1000 = I2C-Master-Modus, Takt = FOSC/(4 * (SSPADD+1)); 1001 = Ladefunktion deaktivieren; 1010 = reserviert; 1011 = reserviert; 1100 = reserviert; 1101 = reserviert; 1110 = I2C Slave-Modus, 7-Bit-Adresse und Startbit- und Stoppbit-Interrupt aktivieren; 1111 = I2C Slave-Modus, 10-Bit-Adresse und aktiviert Start- und Stoppbit-Interrupts.

14.2.2 SPI- Arbeitsprinzip

Bei der Initialisierung des SPI müssen mehrere Optionen angegeben werden. Sie kann durch Programmierung der entsprechenden Steuerbits (SSPCON<5:0> und SSPSTAT<7:6>) vorgegeben werden. Diese Steuerbits werden verwendet, um die folgenden Optionen festzulegen:

- Master-Modus (SCK als Taktausgang) •
- Slave-Modus (SCK als Takteingang) •
- Taktpolarität (Ruhezustand von SCK) • Taktrate
- Abtastphase der Eingangsdaten (Mitte oder Ende der Datenausgabezeit)
- (nur Master-Modus) • Slave-Auswahlmodus
- Taktflanke (Datenausgabe bei steigender/fallender Flanke von SCK)
- (nur Slave-Modus)

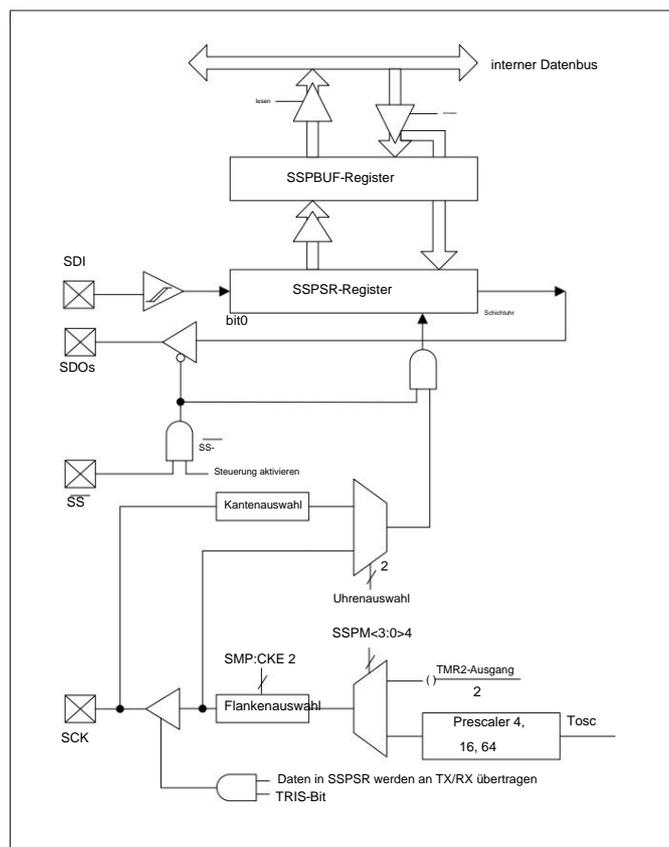


ABBILDUNG 14-1: BLOCKDIAGRAMM DES MSSP-MODULS IM SPI-MODUS

Hinweis: I/O-Pins haben einen Diodenschutz für VDD und VSS.

Das MSSP-Modul besteht aus einem Send-/Empfangsschieberegister (SSPSR) und einem Pufferregister (SSPBUF). Das SSPSR verschiebt Daten in das und aus dem Gerät, das höchstwertige Bit zuerst. SSPBUF hält die zuletzt in das SSPSR geschriebenen Daten, bis neue empfangene Daten bereit sind. Sobald die 8-Bit-Daten empfangen wurden, wird das Byte in das SSPBUF-Register geschoben. Dann wird das Interrupt-Flag-Bit SSPIF des PIR1-Registers gesetzt. Dieses doppelt gepufferte Datenempfangsverfahren (SSPBUF) ermöglicht es, mit dem Empfang des nächsten Bytes zu beginnen, bevor die gerade empfangenen Daten gelesen werden. Während der Datenübertragung/Empfang wird jeder Versuch, in das SSPBUF-Register zu schreiben, ignoriert und das Schreibkollisionserkennungsbit WCOL des SSPCON-Registers wird gesetzt. Zu diesem Zeitpunkt muss der Benutzer das WCOL-Bit per Software löschen, andernfalls ist es unmöglich zu beurteilen, ob die nächste Schreiboperation in SSPBUF erfolgreich abgeschlossen ist.

Wenn die Anwendungssoftware auf den Empfang gültiger Daten wartet, sollte das vorherige Datenbyte im SSPBUF gelesen werden, bevor das nächste zu übertragende Datenbyte in den SSPBUF geschrieben wird. Das Puffer-Voll-Flag-Bit BF (SSPSTAT-Register) wird verwendet, um anzuzeigen, wann der SSPBUF mit empfangenen Daten geladen wurde (Übertragung abgeschlossen). Wird der SPI nur als Sender verwendet, werden die empfangenen Daten ignoriert. MSSP-Interrupts werden häufig verwendet, um festzustellen, wann ein Send- oder Empfangsvorgang abgeschlossen ist. Wenn Sie keine Interrupts verwenden, um das Senden und Empfangen von Daten zu verarbeiten, kann die Verwendung der Software-Abfragemethode auch sicherstellen, dass keine Schreibkonflikte auftreten.

14.2.3 Aktivieren von SPI-E/A

Um den seriellen Port freizugeben, muss das MSSP-Freigabebit SSPEN des SSPCON-Registers gesetzt werden. Um den SPI-Modus zurückzusetzen oder neu zu konfigurieren, löschen Sie zuerst das SSPEN-Bit, initialisieren Sie das SSPCON-Register neu und setzen Sie dann das SSPEN-Bit. Dadurch werden die SDI-, SDO-, SCK- und SS-Pins als serielle Port-Pins konfiguriert. Um diese Pins als serielle Ports zu verwenden, müssen ihre Datenrichtungsbits (im TRIS-Register) auch wie folgt korrekt programmiert werden:

• SDI wird automatisch vom SPI-Modul gesteuert;
 • TRISA<3> von SDO muss gelöscht werden; • TRISA<6>
 von SCK (Master-Modus) muss gelöscht werden; • TRISA< von SCK (Slave-Modus)
 muss gelöscht werden 6> Bit 1; • TRISE<1> von SS muss auf 1 gesetzt werden.

Jede unerwünschte Funktionalität der seriellen Schnittstelle kann übersprungen werden, indem das entsprechende Datenrichtungsregister (TRIS) auf den entgegengesetzten Wert gesetzt wird.

14.2.4 Master-Modus

Der Master steuert SCK, sodass Datenübertragungen jederzeit initiiert werden können. Der Master bestimmt basierend auf einem Softwareprotokoll, wann der Slave Daten senden soll.

Im Master-Modus beginnt die Übertragung oder der Empfang, sobald Daten in das SSPBUF-Register geschrieben werden. Wenn der SPI nur ein Empfänger ist, kann er deaktiviert werden SDO-Ausgang (als Eingang programmieren). Das SSPSR-Register verschiebt kontinuierlich das Signal auf dem SDI-Pin mit der programmierten Taktrate. Wenn jedes Byte empfangen wird, wird es in das SSPBUF-Register als ein normales empfangenes Byte geladen (mit entsprechenden gesetzten Unterbrechungs- und Statusbits). Dies kann in Empfängeranwendungen als "Line Activity Monitor"-Modus nützlich sein.

Die Taktpolarität wird durch entsprechendes Programmieren der CKP-Bits des SSPCON-Registers ausgewählt. Abbildung 14-2, Abbildung 14-4 und Abbildung 14-5 zeigen zeigt das Wellenformdiagramm der PI-Kommunikation, wobei MSb zuerst gesendet wird. Im Master-Modus ist die SPI-Taktrate (Bitrate) vom Benutzer wie folgt programmierbar

Einer der Spaltenpreise:

• FOSC/4 (oder TCY) • FOSC/16
 (oder 4.TCY) • FOSC/64 (oder 16.TCY)

• Ausgang TIMER2/2

Abbildung 14-2 zeigt das Wellenformdiagramm des Master-Modus. Wenn das CKE-Bit des SSPSTAT-Registers gesetzt ist, sind die SDO-Daten vor der Taktflanke auf SCK gültig. Die Variation der in der Figur gezeigten Eingangsabtastwerte wird durch den Zustand des SMP-Bits des SSPSTAT-Registers bestimmt. Die Figur zeigt an, wann die empfangenen Daten in den SSPBUF geladen werden.

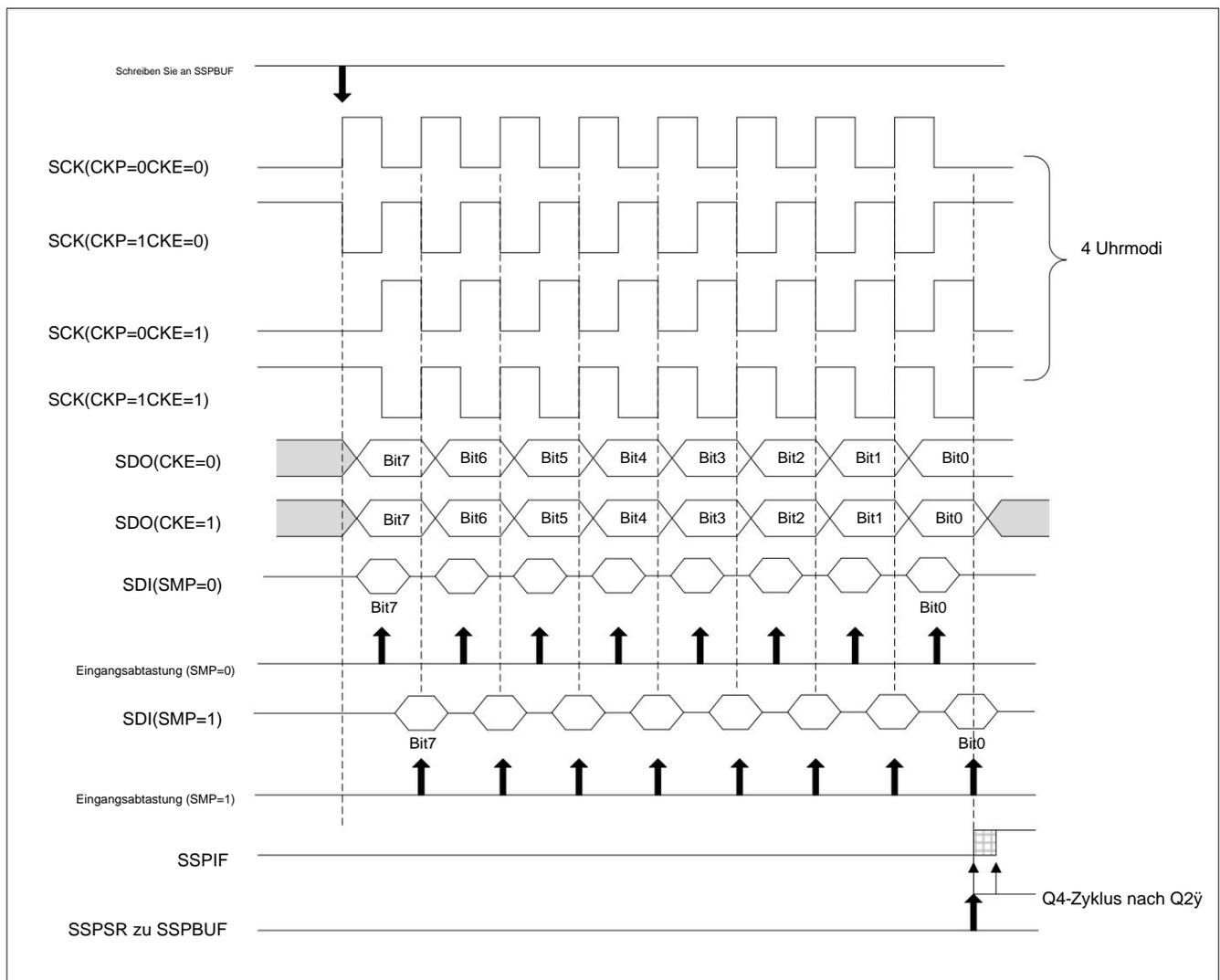


Abbildung 14-2: Wellenformen im SPI-Modus (Master-Modus)

14.2.5 Slave-Modus

Im Slave-Modus werden Daten gesendet und empfangen, wenn ein externer Taktimpuls am SCK-Pin erscheint. Wenn das letzte Datenbit zwischengespeichert ist, wird PIR1

Das SSPIF-Interrupt-Flag-Bit des Registers wird gesetzt.

Im Slave-Modus wird der Takt von einer externen Taktquelle am SCK-Pin bereitgestellt. Der externe Takt muss die in der elektrischen Spezifikation angegebenen hohen und niedrigen Pegel erfüllen

Die Mindestzeitanforderung für das Level.

Im Ruhezustand kann das Slave-Gerät weiterhin Daten senden/empfangen. Das Gerät wacht aus dem Ruhezustand auf, wenn ein Byte empfangen wird.

14.2.6 Slave-Auswahl-Synchronisation

Der SS-Pin ermöglicht dem Gerät, im synchronen Slave-Modus zu arbeiten. Der SPI muss im Slave-Modus mit aktivierter SS-Pin-Steuerung arbeiten. Damit der SS-Pin als Eingang fungiert, darf er nicht auf Low getrieben werden. Wenn der SS-Pin niedrig ist, werden Datenübertragung und -empfang aktiviert, während der SDO-Pin angesteuert wird. Wenn der SS-Pin auf hohem Pegel ist, wird der SDO-Pin selbst während des Sendens von Daten nicht mehr angesteuert, sondern wird zu einem schwebenden Ausgang. Je nach Bedarf der Anwendung kann ein externer Pull-up-/Pull-down-Widerstand angeschlossen werden. Wenn das SPI-Modul zurückgesetzt wird, wird der Bitzähler auf Null gesetzt. Dies

kann durch Forcieren des SS-Pins auf High oder durch Löschen des SSPEN-Bits erfolgen. Das Verbinden des SDO-Pins mit dem SDI-Pin emuliert eine Zweidrahtkommunikation. Wenn der SPI als Empfänger arbeiten muss, kann der SDO-Pin als Eingang konfiguriert werden. Dadurch wird das Senden von Daten von SDO deaktiviert. Da SDI keine Buskonflikte verursacht, kann es immer als Eingang belassen werden (SDI-Funktion).

Notiz:

1. Wenn der SPI im Slave-Modus arbeitet und die SS-Pin-Steuerung aktiviert ist und der SS-Pin auf VDD-Pegel eingestellt ist, wird das SPI-Modul zurückgesetzt Bit.
2. Wenn der SPI im Slave-Modus mit gesetztem CKE (SSPSTAT-Register) verwendet wird, muss die SS-Pin-Steuerung aktiviert werden.

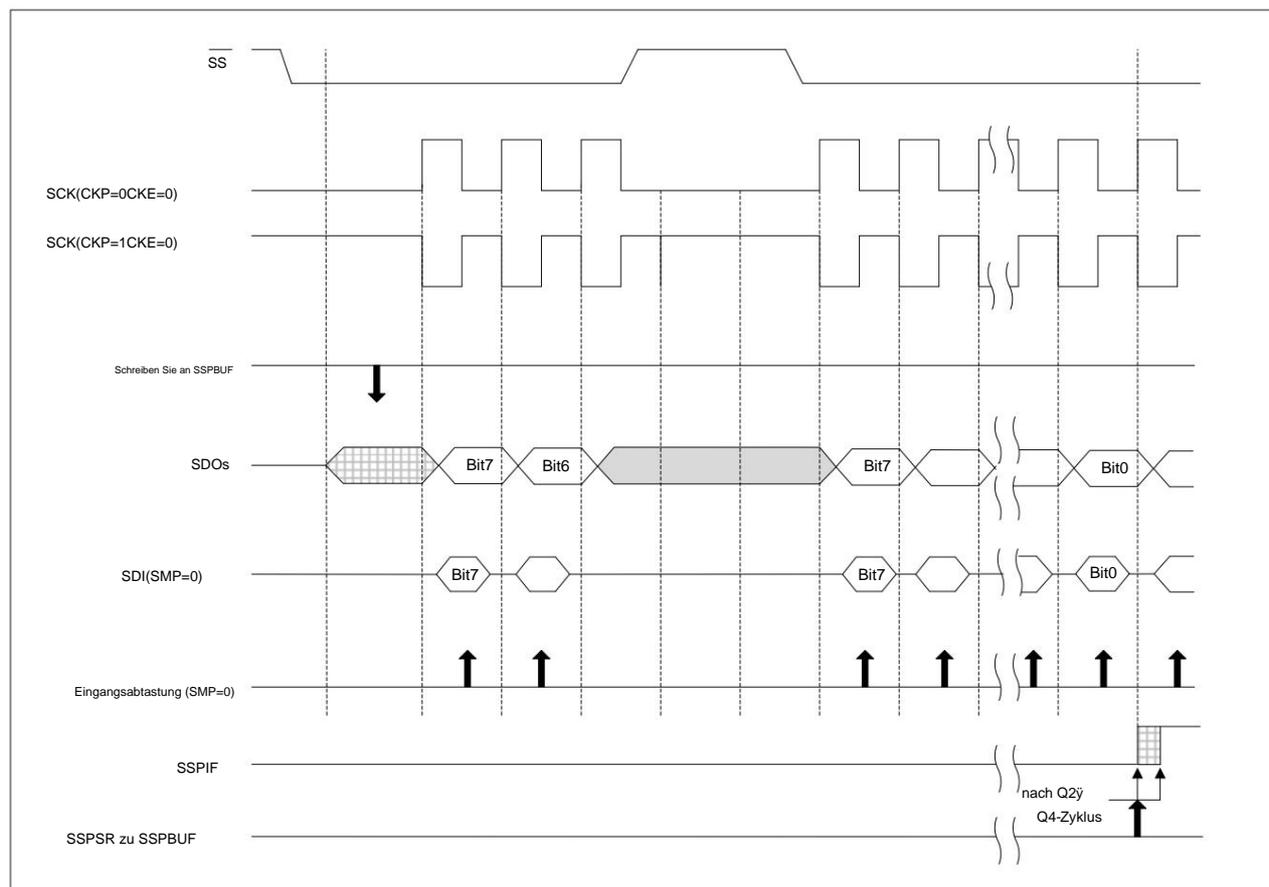


Abbildung 14-3: Slave-Synchronisationswellenform

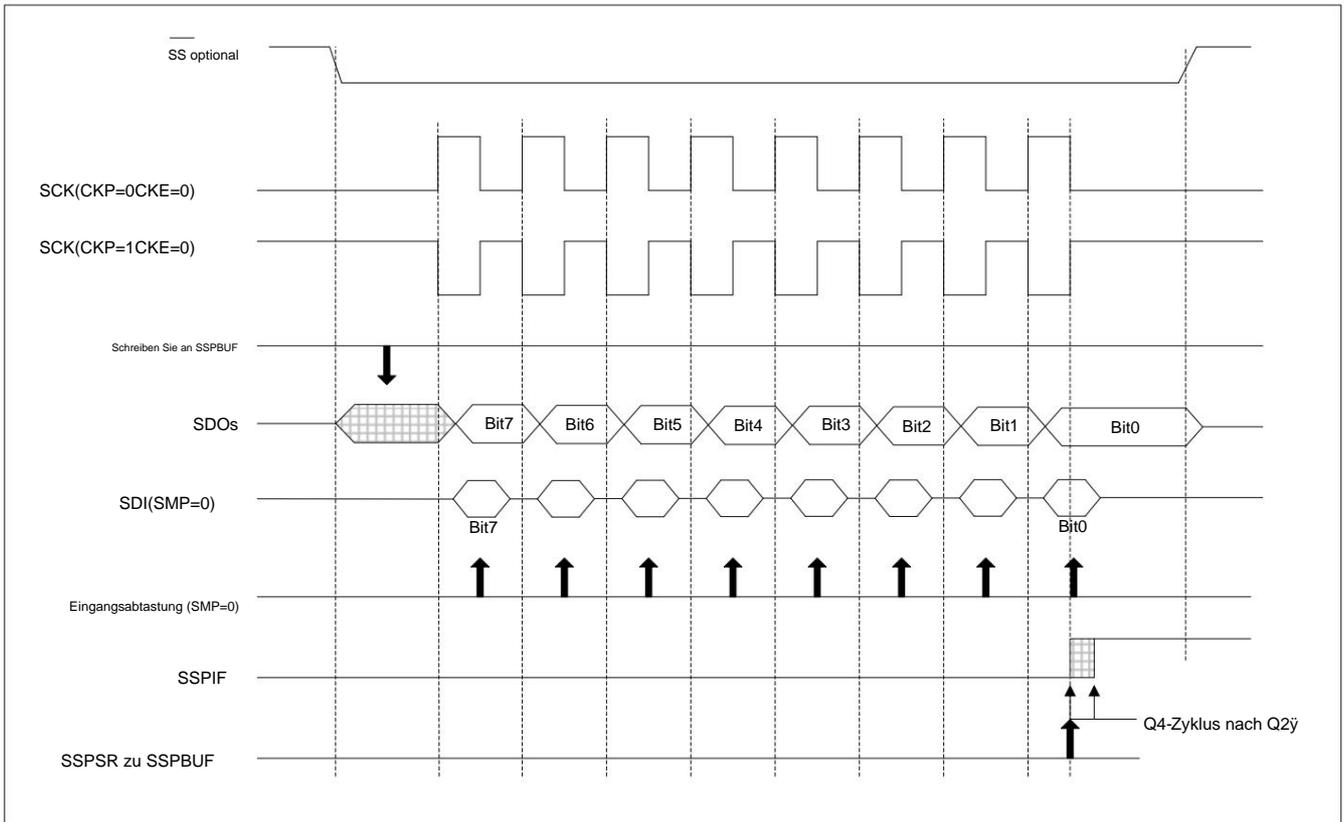


Abbildung 14-4: Wellenform im SPI-Modus (Slave-Modus, CKE=0)

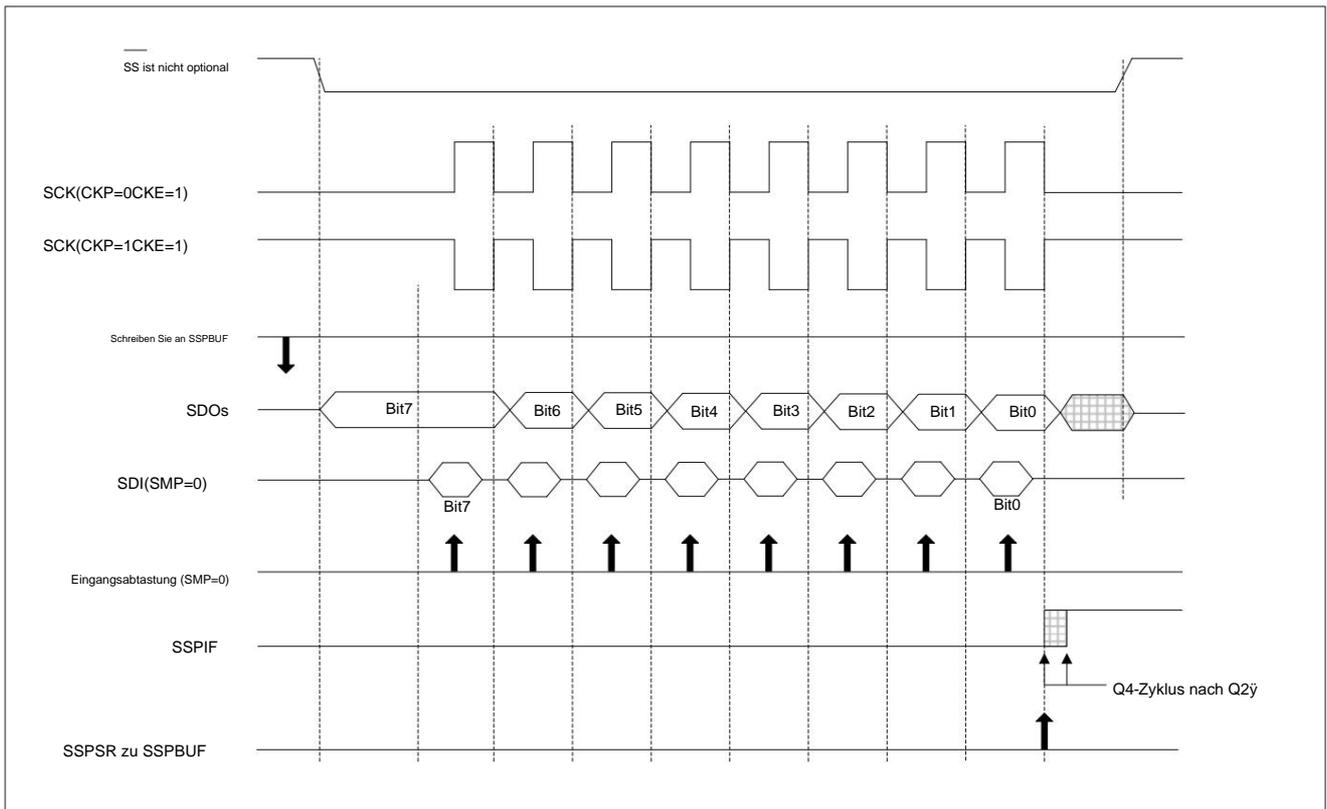


Abbildung 14-5: Wellenform im SPI-Modus (Slave-Modus, CKE=1)

14.2.7 Sleep-Betrieb

Wenn im Master-Modus der Schlafmodus ausgewählt ist, werden die Uhren zu allen Modulen angehalten und das Senden/Empfangen bleibt aktiv, bis das Gerät aufwacht.

Dieser Stillstand. Wenn das Gerät in den Run-Modus zurückkehrt, nimmt das Modul das Senden und Empfangen von Daten wieder auf.

Im Slave-Modus arbeitet das SPI-Sende-/Empfangs-Schieberegister asynchron zum Gerät. Dadurch wird das Gerät in den Schlafmodus versetzt und Daten können weiterhin in das SPI-Sende-/Empfangs-Schieberegister verschoben werden. Wenn alle 8 Datenbits empfangen wurden, wird das MSSP-Interrupt-Flag-Bit gesetzt und weckt das Gerät auf, wenn es aktiviert ist.

14.2.8 Auswirkung des Zurücksetzens

Ein Reset deaktiviert das MSSP-Modul und beendet die aktuelle Übertragung.

14.3 I2C -Modul

Wenn das MSSP-Modul im I2C-Modus arbeitet, kann es alle Master- und Slave-Funktionen implementieren (einschließlich allgemeiner Anruferunterstützung) und Hardware verwenden, um Startbit- und Stoppbit-Interrupts bereitzustellen, um festzustellen, wann der Bus frei ist (Multi-Master-Funktion). Das MSSP-Modul implementiert die Standardmoduspezifikation sowie 7-Bit- und 10-Bit-Adressierung.

Es gibt zwei Pins für die Datenübertragung. Sie sind Taktstift (SCL) - RA6/SCK/SCL-Stift und Datenstift (SDA) - RA5/SDI/SDA-Stift. Der Benutzer muss diese Pins über die TRISA<6:5>-Bits als Eingangs- oder Ausgangspins konfigurieren. Die MSSP-Modulfunktionalität wird durch Setzen des MSSP-Aktivierungsbits SSPEN des SSPCON-Registers aktiviert.

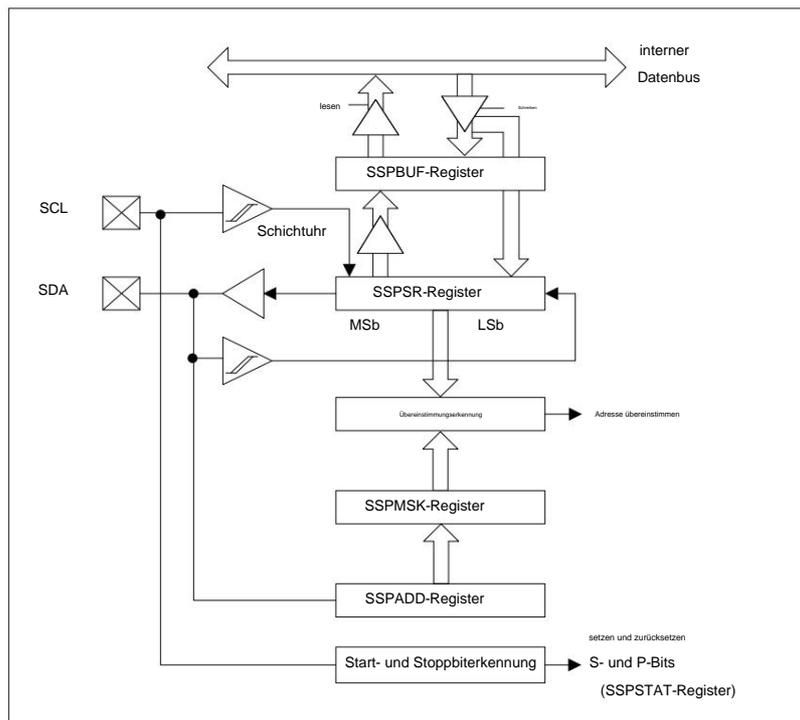


ABBILDUNG 14-6: MSSP-BLOCKDIAGRAM (I2C-MODUS)

Hinweis: I/O-Pins haben Schutzdioden zu VDD und VSS.

Das MSSP-Modul hat sieben Register für den I2C-Betrieb. Sie sind:

- MSSP-Steuerregister 1 (SSPCON) • MSSP-Statusregister (SSPSTAT) • MSSP-Schieberegister (SSPSR): nicht direkt zugänglich • MSSP-Adressregister (SSPADD) • MSSP-Maskenregister (SSPMSK)
- MSSP-Steuerregister 2 (SSPCON2) • Serielles Empfangs-/Sendepufferregister (SSPBUF)

Der I2C-Betrieb kann über das SSPCON-Register gesteuert werden. Folgendes kann unter Verwendung der Modusauswahlbits SSPM<3:0> (SSPCON-Register) ausgewählt werden.

Einer der I2C- Modi:

- I2C-Slave-Modus (7-Bit-Adresse) • I2C-Slave-Modus (10-Bit-Adresse) • I2C-Firmware-gesteuerter Master-Betrieb, Slave-Leerlauf • I2C-Slave-Modus, 10-Bit-Adresse, Startbit aktiviert und Stoppbit-Unterbrechung
- I2C-Master-Modus, Takt = OSC/4 (SSPADD+1) • I2C-Slave-Modus, 7-Bit-Adresse, Startbit aktivieren und Stoppbit unterbrechen

Wenn die SCL- und SDA-Pins als Eingänge programmiert wurden (setzen Sie das entsprechende TRISA-Bit), wird die Auswahl eines beliebigen I2C-Modus mit gesetztem SSPEN-Bit die SCL- und SDA-Pins zum Open-Drain zwingen.

14.3.1 Beschreibung verwandter Register

SSPSTAT: SSP-Statusregister (193H)

193H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF
Rücksetzwert	R/W	R/W	R	R	R	R	R	R
lesen und schreiben	0	0	0	0	0	0	0	0

Bit7

SMP:

1 = Ratensteuerung deaktivieren, Standardgeschwindigkeitsmodus (100 kHz und 1 MHz).

0 = Slew-Steuerung aktivieren, Hochgeschwindigkeitsmodus (400 KHz).

Bit6

CKE: SPI-Taktflankenauswahlbit.

CKP=0

1 = Daten bei steigender Flanke des SCK-Pins übertragen

0 = Daten bei fallender Flanke des SCK-Pins übertragen.

KP = 1

1 = Daten werden bei der fallenden Flanke des SCK-Pins übertragen;

0 = Daten werden bei der steigenden Flanke des SCK-Pins übertragen.

Bit5

D/A: Daten/Adressbit.

1= Zeigt an, dass das letzte empfangene oder gesendete Byte Daten waren.

0= Zeigt an, dass das letzte empfangene oder gesendete Byte eine Adresse war.

Bit4

P: Stoppbit (dieses Bit wird gelöscht, wenn das MSSP-Modul deaktiviert wird (SSPEN wird gelöscht)). 1

= Zeigt an, dass zuletzt ein Stoppbit erkannt wurde (Bit ist 0 beim Zurücksetzen). 0= Zeigt an, dass am

Ende kein Stoppbit erkannt wurde.

Bit3

S: Startbit (dieses Bit wird gelöscht, wenn das MSSP-Modul deaktiviert wird (SSPEN wird gelöscht)).

1 = Zeigt an, dass zuletzt ein Startbit erkannt wurde (Bit ist 0 beim Zurücksetzen). 0 = Kein Startbit

am Ende erkannt.

Bit2

R/W: Bitinformationen lesen/schreiben.

Dieses Bit wird verwendet, um die R/W-Bit-Informationen nach der letzten Adressübereinstimmung zu speichern. Dieses Bit ist nur von der Adressübereinstimmung bis zum nächsten Startbit, Stoppbit oder Nicht-ACK-Bit gültig.

Im I2C-Slave-Modus:

1 = lesen. 0

= Schreiben.

Im I2C-Master-Modus:

1= Senden läuft. 0 =

Es findet keine Übertragung

statt. Das Ergebnis der logischen ODER-Verknüpfung dieses Bits mit SEN, RSEN, PEN, RCEN oder ACKEN zeigt an, ob sich der MSSP im Ruhemodus befindet.

Bit1

UA: Adressbits aktualisieren (nur 10-Bit-I2C-Modus).

1 = Zeigt an, dass der Benutzer die Adresse im SSPADD-Register aktualisieren muss. 0 =

Adresse muss nicht aktualisiert werden.

Bit0

Statusbit BF-Puffer voll.

übernehmen:

1 = Empfang abgeschlossen, SSPBUF ist voll.

0 = Empfang nicht abgeschlossen, SSPBUF ist leer.

schicken:

1 = Daten werden übertragen (ohne ACK- und Stoppbits), SSPBUF ist voll. 0 =

Datenübertragung abgeschlossen (ohne ACK- und Stoppbits), SSPBUF ist leer.

SSPCON: SSP-Steuerregister 1 (194H)

194H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPCON	WCOL SSPOV	SSPEN CKP			SSPM3	SSPM2 SSPM1	SSPM0	
Rücksetzwert	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
lesen und schreiben	0	0	0	0	0	0	0	0

Bit7

WCOL: Kollisionserkennungsbit schreiben.

Master-Modus: 1 = Es wurde versucht, in das SSPBUF-Register zu schreiben, obwohl die I2C-Bedingungen nicht erfüllt waren, um mit dem Senden von Daten zu beginnen.

0 = Kein Konflikt aufgetreten.

Slave-Modus:

1 = SSPBUF-Register wurde geschrieben, während das vorherige Wort übertragen wurde (muss in der Software gelöscht werden). 0

= Kein Konflikt aufgetreten.

Bit6

SSPOV: Überlaufanzeigebit empfangen.

1 = Ein neues Byte wurde empfangen, während das SSPBUF-Register noch die vorherigen Daten enthielt. Das SSPOV-Bit kann im Sendemodus einen beliebigen Wert haben (das Bit muss per Software gelöscht werden).

0 = kein Überlauf.

Bit5

SSPEN: Synchronous Serial Port Enable Bit (diese Pins müssen richtig als Eingangs- oder Ausgangspins konfiguriert sein).

1 = Aktiviert den seriellen Port und konfiguriert die SDA- und SCL-Pins als serielle Port-Pins.

0 = Deaktiviert den seriellen Port und konfiguriert diese Pins als E/A-Port-Pins.

Bit4

CKP: Taktpolaritätsauswahlbit. Im

I2C-Slave-Modus: SCK-Freigabesteuerung. 1 = Uhr ist

aktiviert. 0 = Taktleitung

niedrig halten (Taktdehnung) (wird verwendet, um die Dateneinrichtungszeit sicherzustellen). Im I2C-

Master-Modus: Wird in diesem Modus nicht verwendet.

Bit3~Bit0

SSPM<3:0>: Auswahlbits für den Modus des synchronen seriellen Anschlusses.

0000 = SPI-Mastermodus, Uhr = FOSC/4.

0001 = SPI-Mastermodus, Uhr = FOSC/16.

0010 = SPI-Mastermodus, Uhr = FOSC/64.

0011 = SPI-Mastermodus, Takt = TMR2-Ausgang/2.

0100 = SPI-Slave-Modus, Uhr = SCK-Pin, SS-Pin-Steuerung aktivieren.

0101 = SPI-Slave-Modus, Uhr = SCK-Pin, SS-Pin-Steuerung deaktiviert, SS kann als I/O-Pin verwendet werden.

0110 = I2C-Slave-Modus, 7-Bit-Adresse.

0111 = I2C-Slave-Modus, 10-Bit-Adresse.

1000 = I2C-Mastermodus, Takt = FOSC/(4 * (SSPADD+1)). 1001 =

Ladefunktion deaktivieren. 1010 = Reserviert. 1011 = Reserviert. 1100 =

Reserviert. 1101 = Reserviert.

1110 = I2C-Slave-Modus, 7-Bit-Adresse und Startbit- und Stoppbit-Interrupt aktivieren.

1111 = I2C-Slave-Modus, 10-Bit-Adresse und Startbit- und Stoppbit-Interrupt aktivieren.

Alle seriellen Taktimpulse und Start/Stop-Bedingungen werden vom Mastergerät generiert. Eine Stoppbedingung oder eine wiederholte Startbedingung kann die Übertragung beenden. Wegen wiederholt Der aktive Zustand ist auch der Beginn der nächsten seriellen Übertragung, sodass der I2C-Bus nicht freigegeben wird. Im Master-Transmitter-Modus werden serielle Daten auf SDA und der serielle Takt auf SCL ausgegeben. Das erste übertragene Byte besteht aus der Adresse des empfangenden Geräts (7 Bits) und einem Lese-/Schreibbit (R/W). In diesem Fall ist das R/W-Bit logisch 0. Serielle Daten werden 8 Bits gleichzeitig gesendet. Jedes Mal, wenn ein Byte gesendet wird, wird ein Bestätigungsbit empfangen. Ausgänge für Start- und Stoppbedingungen zeigen den Beginn und das Ende einer seriellen Übertragung an.

Im Master-Empfangsmodus besteht das erste übertragene Byte aus der Adresse des sendenden Geräts (7 Bit) und dem R/W-Bit. In diesem Fall das R/W-Bit wird logisch 1 sein. Daher ist das erste übertragene Byte eine 7-Bit-Slave-Adresse, gefolgt von einer 1 für den Empfang. Serielle Daten werden auf SDA empfangen, während der serielle Takt auf SCL ausgegeben wird. Empfangen Sie jedes Mal serielle 8-Bit-Daten. Jedes Mal, wenn ein Byte empfangen wird, wird ein Bestätigungsbit gesendet. Die Start- und Stoppbedingungen geben den Beginn bzw. das Ende einer Übertragung an. Im I2C-Modus wird der im SPI-Modus verwendete Baudratengenerator verwendet, um die SCL-Taktfrequenz auf 100 kHz, 400 kHz oder 1 MHz einzustellen.

Der Neuladewert des Baudratengenerators befindet sich in den unteren 7 Bits des SSPADD-Registers. Wenn in SSPBUF geschrieben wird, beginnt der Baudratengenerator automatisch mit dem Zählen. Wenn die angegebene Operation abgeschlossen ist (dh auf das letzte gesendete Datenbit folgt ein ACK), hört die interne Uhr automatisch auf zu zählen und der SCL-Pin verbleibt in seinem letzten Zustand. Das Folgende ist eine typische Abfolge von Übertragungsereignissen: - Der Benutzer erzeugt eine Startbedingung, indem er das Startaktivierungsbit SEN (SSPCON2-Register) setzt.

- SSPIF-Bit ist gesetzt. Das MSSP-Modul wartet die erforderliche Startzeit ab, bevor es irgendetwas anderes tut. - Benutzer lädt SSPBUF mit Slave-Adresse für die Übertragung. - Die Adresse wird auf dem SDA-Pin herausgeschoben, bis alle 8 Bit übertragen wurden.

- Das MSSP-Modul schiebt das ACK-Bit vom Slave ein und schreibt seinen Wert in das ACKSTAT-Bit des SSPCON2-Registers.

- Das MSSP-Modul erzeugt einen Interrupt, indem es am Ende des neunten Taktzyklus das SSPIF-Bit setzt. - Benutzer lädt 8-Bit-Daten in SSPBUF. - Daten werden auf dem SDA-Pin herausgeschoben, bis alle 8 Bits übertragen wurden.

- Das MSSP-Modul schiebt das ACK-Bit vom Slave ein und schreibt seinen Wert in das ACKSTAT-Bit des SSPCON2-Registers.

- Das MSSP-Modul erzeugt einen Interrupt durch Setzen des SSPIF-Bits am Ende des 9. Taktes. - Der Benutzer erzeugt eine Stopp-Bedingung, indem er das Stopp-Aktivierungsbit (PEN) (SSPCON2-Register) setzt. - Ein Interrupt wird generiert, sobald die STOP-Bedingung abgeschlossen ist.

14.3.4 Baudratengenerator

Im I2C-Master-Modus befindet sich der Reload-Wert des Baudratengenerators in den unteren sieben Bits des SSPADD-Registers (Abbildung 14-8). Wenn dieser Wert geladen ist, beginnt der Baudratengenerator automatisch mit dem Zählen und Dekrementieren auf 0 und stoppt dann bis zum nächsten Neuladen. Der BRG zählt zweimal bei Q2- und Q4-Taktzyklen pro Befehlszyklus (TCY) herunter. Im I2C-Master-Modus wird BRG automatisch neu geladen. Wenn beispielsweise eine Clock-Arbitrierung auftritt, lädt der BRG neu, wenn der SCL-Pin hoch abgetastet wird (Abbildung 14-9).

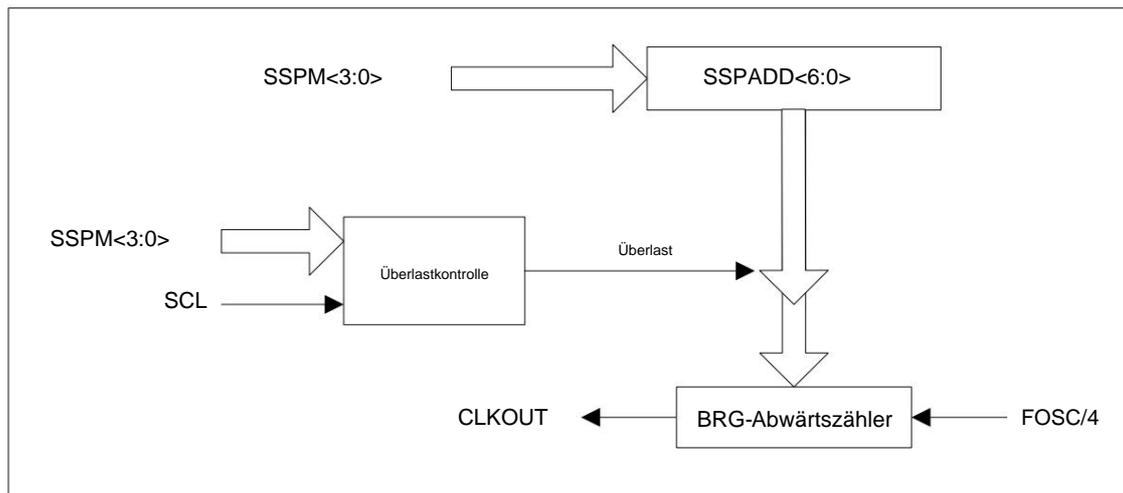


Abbildung 14-8: Blockdiagramm des Baudratengenerators

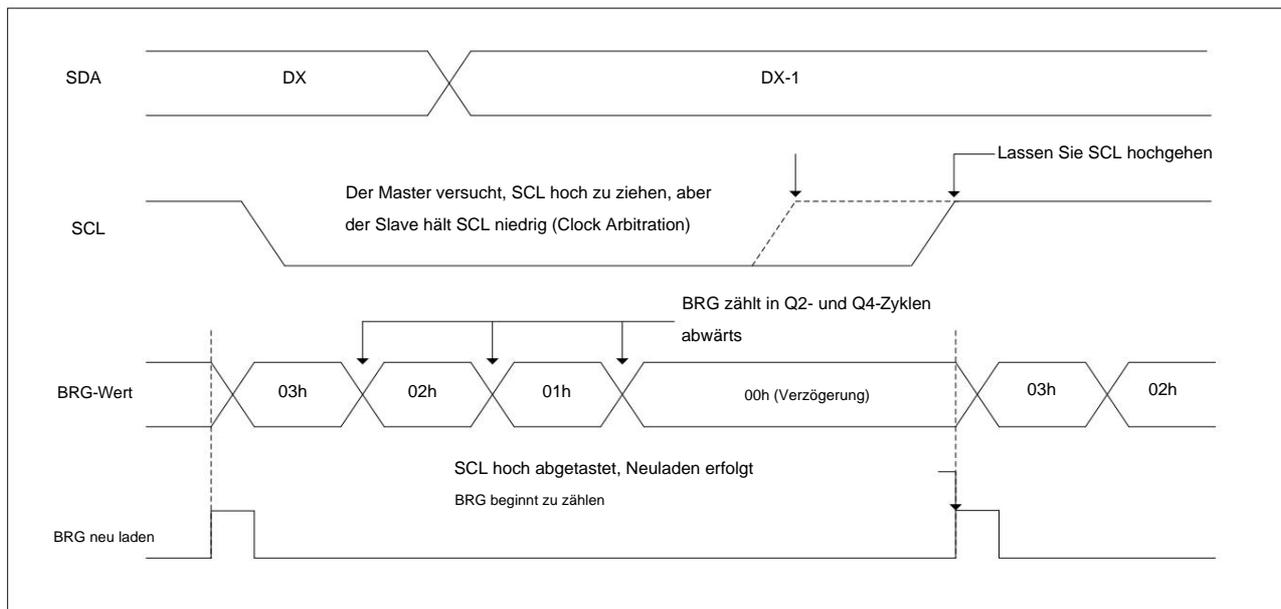


Abbildung 14-9: Timing des Baudratengenerators mit Clock Arbitration

14.3.5 Übertragung im I2C -Mastermodus

Das Senden eines Datenbytes, einer 7-Bit-Adresse oder der anderen Hälfte einer 10-Bit-Adresse kann direkt erfolgen, indem ein Wert in das SSPBUF-Register geschrieben wird. Diese Aktion setzt das Buffer Full Flag Bit, BF, und der Baudratengenerator beginnt mit dem Zählen und startet die nächste Übertragung. Jedes Adress-/Datenbit wird auf dem SDA-Pin herausgeschoben, nachdem die fallende Flanke von SCL gültig ist. SCL wird für einen Rollover-Zählwert des Baudratengenerators (TBRG) niedrig gehalten. Die Daten sollten gültig bleiben, bis SCL hoch freigegeben wird. Wenn der SCL-Stift hoch freigegeben wird, bleibt er während TBRG hoch. Die Daten auf dem SDA-Pin müssen während dieses Zeitraums und für einen Zeitraum nach der nächsten fallenden SCL-Flanke stabil bleiben. Nachdem das 8. Bit herausgeschoben ist (abfallende Flanke des 8. Taktzyklus), wird das BF-Flag-Bit gelöscht und der Master gibt SDA frei.

Wenn zu diesem Zeitpunkt eine Adressübereinstimmung auftritt oder Daten richtig empfangen werden, antwortet die adressierte Slave-Vorrichtung mit einem ACK-Bit zur neunten Bitzeit. ACK Der Status wird an der fallenden Flanke des neunten Taktzyklus in das ACKDT-Bit geschrieben. Das Bestätigungsstatusbit ACKSTAT wird auf 0 gelöscht, nachdem der Master eine Bestätigung empfangen hat; es wird auf 1 gesetzt, wenn keine Bestätigung empfangen wurde. Nach dem 9. Takt wird das SSPIF-Bit gesetzt und der Haupttakt (Baudratengenerator) ausgesetzt, bis das nächste Datenbyte in den SSPBUF geladen wird, der SCL-Pin bleibt niedrig und SDA bleibt unverändert. Nach dem Schreiben in SSPBUF wird jedes Bit der Adresse bei der fallenden Flanke von SCL herausgeschoben, bis alle 7 Bits der Adresse und das R/W-Bit herausgeschoben sind. An der fallenden Flanke des achten Takts zieht der Master den SDA-Pin hoch, damit der Slave mit einer Bestätigung antworten kann. An der abfallenden Flanke des neunten Takts tastet die Master-Vorrichtung den SDA-Pin ab, um zu bestimmen, ob die Adresse von der Slave-Vorrichtung erkannt wird. Der Status des ACK-Bits wird in das ACKSTAT-Statusbit (SSPCON2-Register) geladen. Nach der fallenden Flanke des neunten Takts zum Senden der Adresse wird SSPIF auf 1 gesetzt, das BF-Flag wird auf 0 gelöscht, der Baudratengenerator wird bis zum nächsten Schreiben auf SSPBUF ausgeschaltet und der SCL-Pin bleibt niedrig, wodurch die SDA-Pin zum Schweben.

14.3.5.1 BF- Status-Flag

Im Sendemodus wird das BF-Bit (SSPSTAT-Register) gesetzt, wenn die CPU in SSPBUF schreibt, und gelöscht, nachdem alle 8 Datenbits hinausgeschoben wurden.

14.3.5.2 WCOL- Status-Flag

Wenn der Benutzer während einer Übertragung in SSPBUF schreibt (d. h. während das SSPSR immer noch Datenbytes verschiebt), wird WCOL gesetzt und der Inhalt des Puffers Der Inhalt bleibt unverändert (es findet kein Schreibvorgang statt). WCOL muss per Software gelöscht werden.

14.3.5.3 ACKSTAT- STATUS-FLAG

Im Sendemodus wird das ACKSTAT-Bit (SSPCON2-Register) gelöscht, wenn der Slave eine Bestätigungsantwort sendet (ACK=0), es wird auf 1 gesetzt, wenn der Slave nicht bestätigt (ACK=1). Ein Slave sendet ein Acknowledge, wenn er seine Adresse (einschließlich einer allgemeinen Rufadresse) erkennt oder wenn er Daten korrekt empfangen hat.

14.3.6 Empfang im I2C-Mastermodus

Der Empfang im Master-Modus wird durch Programmieren des Receive-Enable-Bits RCEN (SSPCON2-Register) aktiviert. Der Baudratengenerator beginnt zu zählen und bei jedem Rollover ändert sich der Zustand des SCL-Pins (hoch zu niedrig oder niedrig zu hoch) und Daten werden in den SSPSR verschoben. Nach der fallenden Flanke des achten Takts wird das Empfangsaktivierungs-Flag automatisch gelöscht, der Inhalt von SSPSR wird in SSPBUF geladen, das BF-Flag wird auf 1 gesetzt, das SSPIF-Flag wird auf 1 gesetzt, der Baudratengenerator unterbricht das Zählen und SCL bleibt niedrig. An diesem Punkt befindet sich MSSP im Ruhezustand und wartet auf den nächsten Befehl. Wenn die CPU den Puffer liest, wird das BF-Flag automatisch gelöscht. Der Benutzer kann ein Acknowledge-Bit senden, nachdem der Empfang abgeschlossen ist, indem er das Acknowledge Sequence Enable-Bit ACKEN (SSPCON2-Register) setzt.

14.3.6.1 BF- Status-Flag

Beim Empfang wird das BF-Bit gesetzt, wenn ein Adress- oder Datenbyte vom SSPSR in den SSPBUF geladen wird; das BF-Bit wird gelöscht, wenn das SSPBUF-Register gelesen wird null.

14.3.6.2 SSPOV- Statusflags

Beim Empfang, wenn das SSPSR 8-Bit-Daten empfängt, wird das SSPOV-Bit auf 1 gesetzt, und das BF-Flag wurde beim vorherigen Empfang auf 1 gesetzt.

14.3.6.3 WCOL- Statusflags

Wenn der Benutzer während des Empfangs in SSPBUF schreibt (d. h. während das SSPSR immer noch Datenbytes verschiebt), wird das WCOL-Bit gesetzt und der Pufferinhalt

Keine Änderung (kein Schreibvorgang aufgetreten).

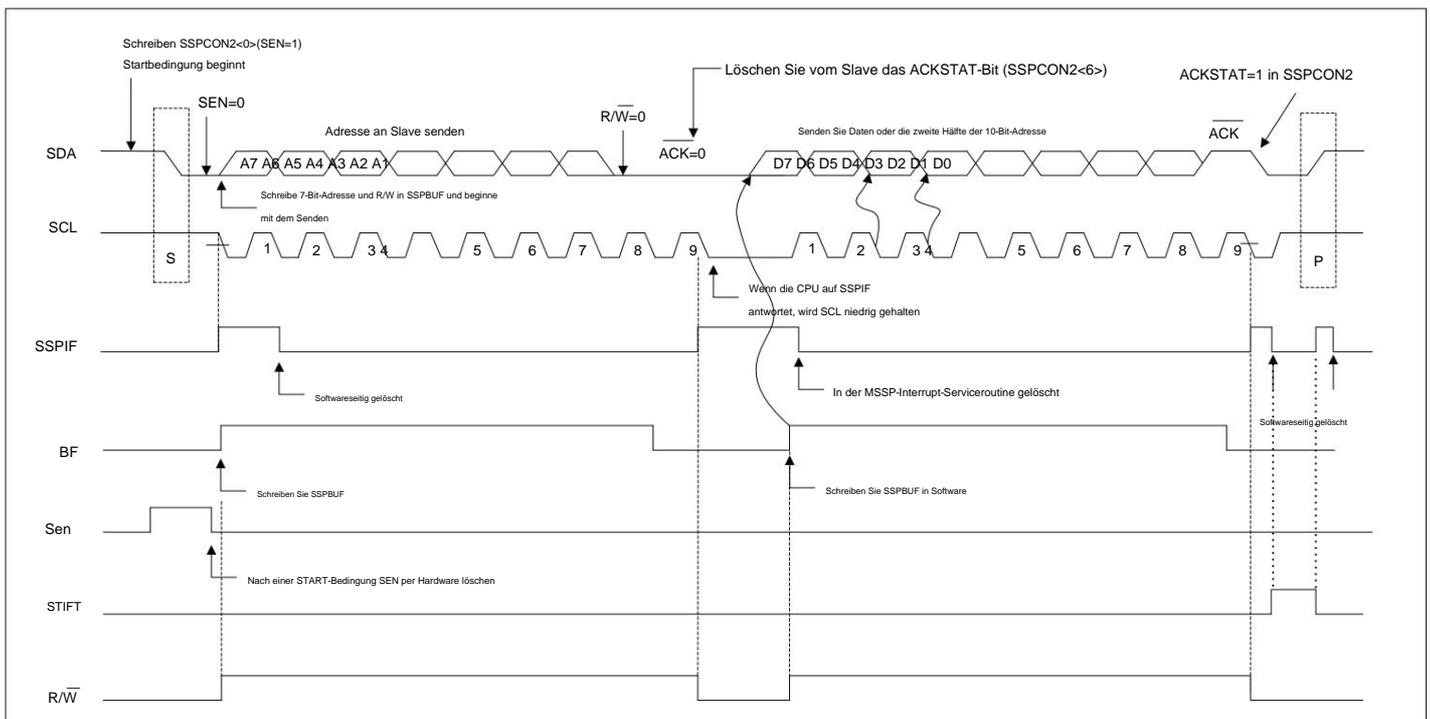


Abbildung 14-10: I2CTM-Mastermodus-Übertragungstiming (7-Bit- oder 10-Bit-Adresse)

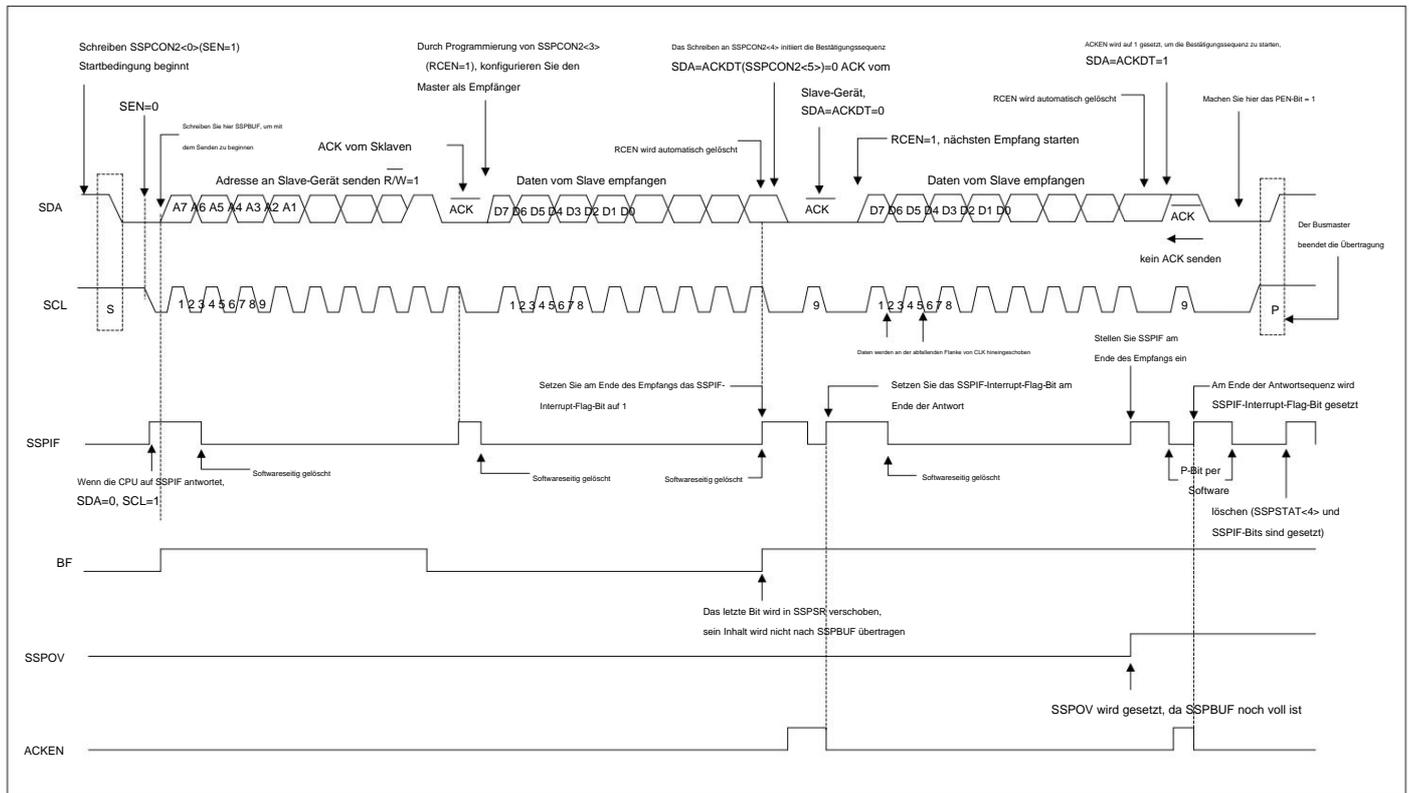


Abbildung 14-11: I2CTM-Mastermodus-Empfangstimung (7-Bit-Adresse)

14.3.7 Startbedingungs-Timing für den I2C-Master -Modus

Um eine Startbedingung einzuleiten, sollte der Benutzer das Startbedingungsflag SEN des SSPCON2-Registers setzen. Wenn sowohl die SDA- als auch die SCL-Pins abgetastet werden. Wenn er hoch ist, wird der Baudratengenerator mit den Inhalten von SSPADD<6:0> neu geladen und beginnt mit dem Zählen. Wenn der Baudratengenerator abläuft (TBRG), wenn sowohl SCL als auch SDA hoch abgetastet werden, wird der SDA-Stift niedrig getrieben. Das Treiben von SDA auf Low, während SCL auf High ist, ist eine Startbedingung und setzt das S-Bit (SSPSTAT-Register). Der Baudratengenerator wird dann mit den Inhalten von SSPADD<6:0> neu geladen und fährt mit dem Zählen fort. Wenn der Baudratengenerator abläuft (TBRG), wird das SEN-Bit des SSPCON2-Registers automatisch durch die Hardware gelöscht. Der Baudratengenerator wird ausgesetzt, die SDA-Leitung wird niedrig gehalten und die START-Bedingung endet.

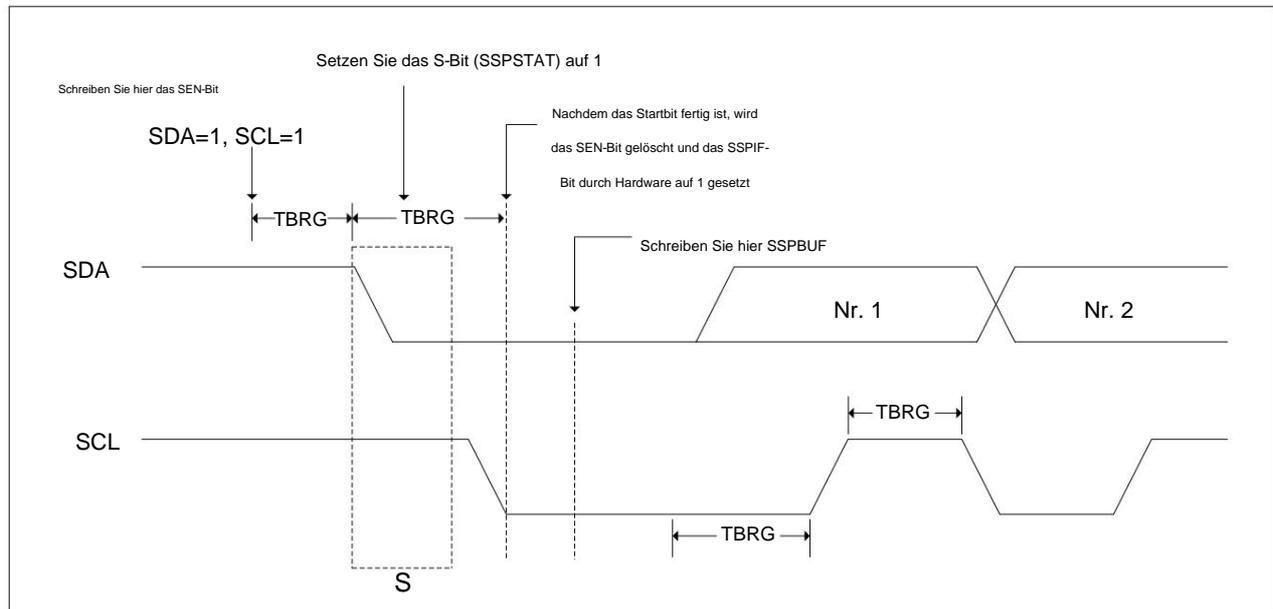


Abbildung 14-12: Timing des ersten Startbits

14.3.7.1 WCOL- Statusflags

Wenn die Startsequenz im Gange ist und der Benutzer in SSPBUF schreibt, wird WCOL gesetzt und der Pufferinhalt bleibt unverändert (es findet kein Schreibvorgang statt).

Hinweis: Da keine Warteschlange von Ereignissen erlaubt ist, können die unteren 5 Bits von SSPCON2 nicht beschrieben werden, bis die START-Bedingung abgeschlossen ist.

14.3.9 Bestätigungssequenz-Timing

Die Acknowledge Sequence wird freigegeben, indem das Acknowledge Sequence Enable Bit ACKEN (SSPCON2-Register) gesetzt wird. Wenn dieses Bit auf 1 gesetzt ist, ist der SCL-Pin Niedrig gezogen, erscheint der Inhalt des Bestätigungsdatenbits auf dem SDA-Stift. Wenn der Benutzer eine Bestätigung erzeugen möchte, sollte das ACKDT-Bit gelöscht werden, andernfalls sollte der Benutzer das ACKDT-Bit setzen, bevor er mit der Bestätigungssequenz beginnt. Dann zählt der Baudratengenerator für eine Rollover-Periode (TBRG), und dann wird der SCL-Pin hochgezogen. Wenn der SCL-Pin hoch abgetastet wird (Taktarbitrierung), zählt der Baudratengenerator für einen weiteren TBRG -Zyklus. Dann wird der SCL-Pin auf Low gezogen. Danach wird das ACKEN-Bit automatisch gelöscht, der Baudratengenerator ausgeschaltet und das MSSP-Modul in den Ruhemodus versetzt.

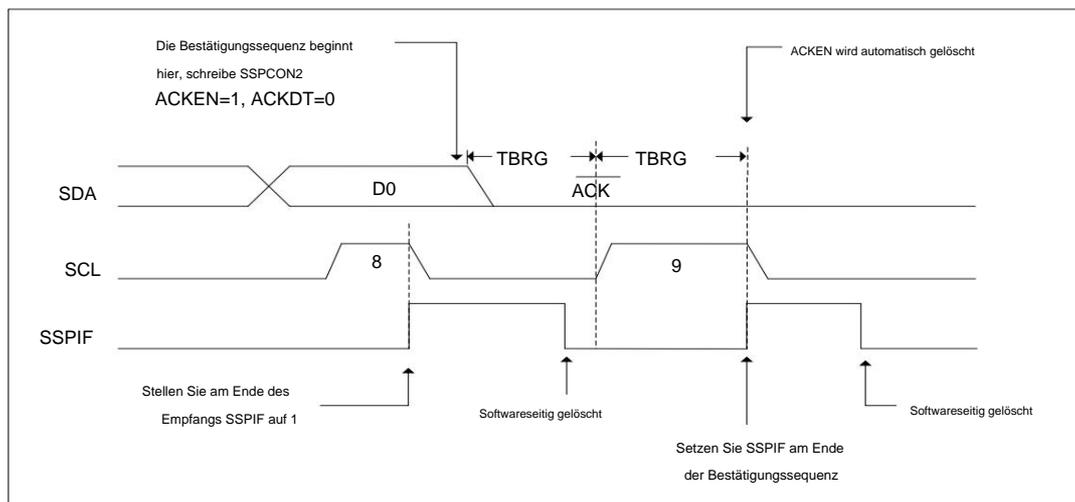


Abbildung 14-14: Bestätigungssequenz-Timing-Wellenform

Hinweis: TBRG = eine Periode des Baudratengenerators.

14.3.9.1 WCOL- Status-Flag-Bits

Wenn der Benutzer in den SSPBUF schreibt, während eine Bestätigungssequenz abläuft, wird WCOL gesetzt und der Inhalt des Puffers bleibt unverändert (es ist kein Schreibvorgang aufgetreten).

14.3.10 STOP-BEDINGUNGSFOLGE

Am Ende eines Empfangs/Sendevorgangs wird ein Stopbit auf dem SDA-Pin erzeugt, indem das Aktivierungsbit für die Stopsequenz PEN (SSPCON2-Register) gesetzt wird. Am Ende des Empfangs/der Übertragung wird der SCL-Stift nach der abfallenden Flanke des neunten Takts niedrig gehalten. Wenn das PEN-Bit gesetzt ist, macht die Master-Vorrichtung SDA niedrig. Wenn die SDA-Leitung niedrig abgetastet wird, wird der Baudratengenerator neu geladen und zählt auf Null herunter. Wenn der Baudratengenerator abläuft, wird der SCL-Stift hochgezogen und ein TBRG (Baudratengenerator-Rollover) später wird der SDA-Stift wieder hochgezogen. Das P-Bit (SSPSTAT-Register) wird gesetzt, wenn der SDA-Stift hoch abgetastet wird und SCL ebenfalls hoch ist. Nach einer TBRG-Periode wird das PEN-Bit gelöscht und das SSPIF-Bit gesetzt.

14.3.10.1 WCOL- Statusflags

Wenn der Benutzer versucht, in den SSPBUF zu schreiben, während die Stop-Sequenz abläuft, wird das WCOL-Bit gesetzt und der Inhalt des Puffers wird nicht geändert (es ist kein Schreibvorgang aufgetreten).

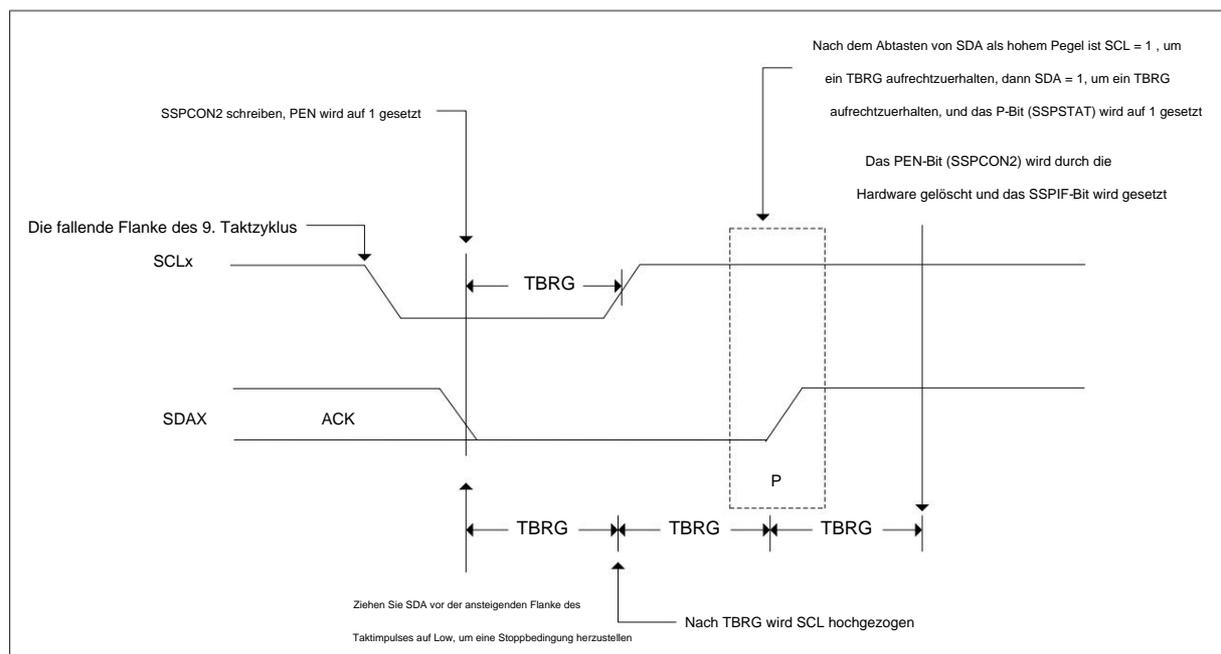


Abbildung 14-15: Stopbedingung Empfangs- oder Sendemodus

Hinweis: TBRG = eine Periode des Baudratengenerators.

14.3.11 Uhr-Schlichtung

Die Taktarbitrierung tritt auf, wenn der Master den SCL-Pin während eines beliebigen Empfangs-, Sende- oder wiederholten Start-/Stopp-Zustands deaktiviert (ermöglicht, dass der SCL-Pin hoch schwebt). Wenn es dem SCL-Pin erlaubt wird, hoch zu schweben, unterbricht der Baudratengenerator (BRG) das Zählen, bis der SCL-Pin tatsächlich hoch abgetastet wird. Wenn der SCL-Pin hoch abgetastet wird, wird der Baudratengenerator mit dem Inhalt von SSPADD<6:0> neu geladen und beginnt mit dem Zählen. Dadurch wird sichergestellt, dass SCL für mindestens einen BRG-Rollover-Zyklus hoch bleibt, wenn ein externes Gerät den Takt nach unten zieht.

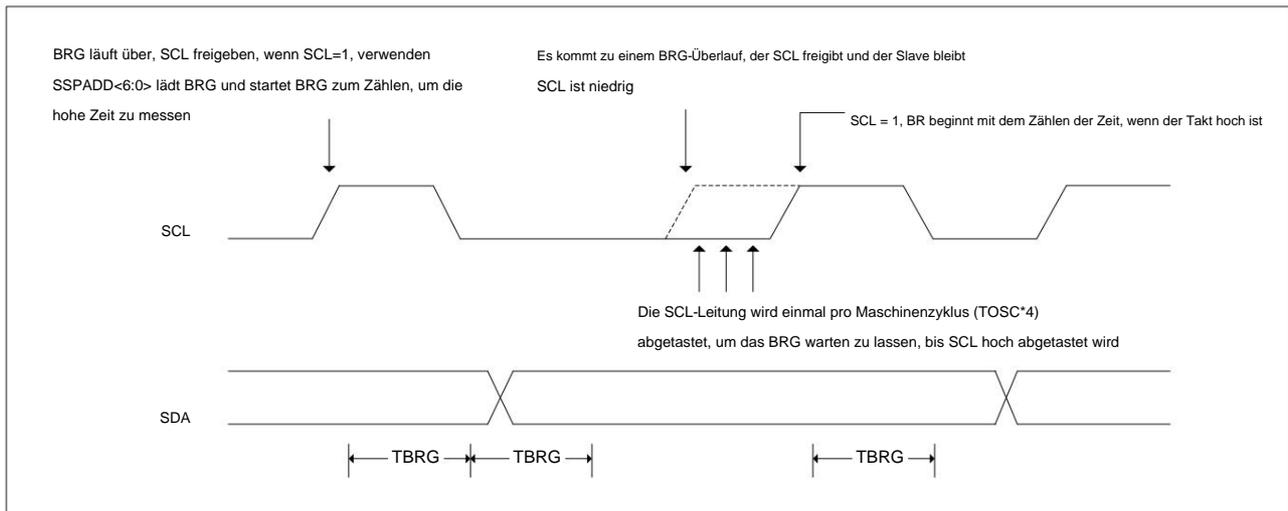


Abbildung 14-16: Clock Arbitration Timing im Master-Sendemodus

14.3.12 Multi-Host-Modus

Im Multi-Master-Modus ist es möglich, festzustellen, wann der Bus frei ist, indem bei der Erkennung von Start- und Stoppbedingungen Interrupts generiert werden. Die Stopp- (P) und Start- (S) Bits werden beim Zurücksetzen gelöscht oder wenn das MSSP-Modul deaktiviert wird. Wenn das P-Bit gesetzt ist, kann die Kontrolle über den I2C-Bus erlangt werden, andernfalls ist der Bus im Leerlauf Zustand, und die P- und S-Bits werden gelöscht. Wenn der STOP-Zustand eintritt, während der Bus belegt ist, wird ein Interrupt generiert (wenn MSSP-Interrupts aktiviert sind).

Beim Betrieb im Multi-Master-Modus muss die SDA-Leitung auf Arbitrierung überwacht werden, um festzustellen, ob der Signalpegel dem gewünschten Ausgangspegel entspricht. Diese Überprüfung erfolgt durch die Hardware schließt ab und das Ergebnis wird in die BCLF-Bits

gesetzt. Quorum kann in den folgenden Zuständen fehlschlagen:

- Adressübertragung •
- Datenübertragung •
- Startbedingung •
- Wiederholte Startbedingung
- Antwortbedingung

14.3.13 Multi-Master-Kommunikation, Buskonflikt und Busarbitrierung

Der Multi-Master-Modus wird durch Busarbitrierung unterstützt. Wenn Master Adress-/Datenbits auf dem SDA-Pin ausgegeben, tritt eine Busarbitrierung auf, wenn ein Master eine 1 auf SDA ausgibt, indem er den SDA-Pin hoch schweben lässt, während der andere Master eine 0 ausgibt. Wenn am SDA-Pin eine 1 erwartet wird und am SDA-Pin tatsächlich eine 0 abgetastet wird, ist eine Buskollision aufgetreten. Der Master setzt das Buskollisions-Interrupt-Flag-Bit BCL1F und setzt den I2C-Port in den Ruhezustand zurück.

Wenn während der Übertragung ein Buskonflikt auftritt, stoppt die Übertragung, das BF-Flag wird gelöscht, die SDA- und SCL-Leitungen werden hochgezogen und der SSPBUF darf geschrieben werden. Nach dem Ausführen der Buskollisionsunterbrechungsdienstroutine kann der Benutzer, wenn der I2C-Bus frei ist, die Kommunikation wieder aufnehmen, indem er eine Startbedingung ausgibt. Wenn während eines Start-, wiederholten Start-, Stopp- oder Bestätigungszustands eine Buskollision auftritt, wird der Zustand abgebrochen, die SDA- und SCL-Leitungen werden deaktiviert und die entsprechenden Steuerbits im SSPCON2-Register werden gelöscht. Nach dem Ausführen der Buskollisionsunterbrechungsdienstroutine kann der Benutzer, wenn der I2C-Bus frei ist, die Kommunikation wieder aufnehmen, indem er eine Startbedingung ausgibt. Der Master überwacht weiterhin die SDA- und SCL-Pins. Wenn eine STOP-Bedingung eintritt, wird das SSPIF-Bit gesetzt. Das Schreiben in SSPBUF beginnt mit dem Übertragen von Daten mit dem ersten Datenbit, unabhängig vom Fortschritt der Übertragung, wenn eine Buskollision auftritt.

Im Multi-Master-Modus ist es möglich, festzustellen, wann der Bus frei ist, indem bei der Erkennung von Start- und Stoppbedingungen Interrupts generiert werden. Wenn das P-Bit auf 1 gesetzt ist, wird die Übernahme der Kontrolle über den I2C-Bus, ansonsten ist der Bus frei und die S- und P-Bits sind gelöscht.

14.3.14 Slave-Modus

Im Slave-Modus müssen der SCL-Pin und der SDA-Pin als Eingänge konfiguriert werden (TRISA<6:5> gesetzt). MSSP bei Bedarf (z. B. vom Sender)

Der Baustein überschreibt den Eingangszustand mit den Ausgangsdaten.

Wenn die Adresse übereinstimmt oder wenn die nach der Adressübereinstimmung gesendeten Daten empfangen werden, generiert die Hardware automatisch einen Bestätigungsimpuls (ACK) und sendet die Im SSPSR-Register empfangene Daten werden in das SSPBUF-Register geladen.

Das MSSP-Modul generiert diesen ACK-Impuls nicht, solange eine der folgenden Bedingungen erfüllt ist:

- Puffer-Voll-Flag-Bit BF (SSPCON-Register) wird auf 1 gesetzt, bevor die Datenübertragung empfangen wird.
- Das Overflow-Flag-Bit SSPOV (SSPCON-Register) wurde gesetzt, bevor die gesendeten Daten empfangen wurden.

In diesem Fall wird nicht der Wert des SSPSR-Registers in den SSPBUF geladen, sondern das SSPIF-Bit des PIR1-Registers gesetzt. Das BF-Bit ist eingeschaltet wird durch Lesen des SSPBUF-Registers gelöscht, während das SSPOV-Bit durch Software gelöscht wird.

Für einen ordnungsgemäßen Betrieb muss der SCL-Takteingang die Anforderungen an die Mindest-High-Zeit und die Mindest-Low-Zeit erfüllen.

14.3.14.1 Adressierung

Sobald das MSSP-Modul aktiviert ist, wartet es auf das Eintreten einer Startbedingung. Nachdem eine START-Bedingung auftritt, werden 8 Datenbits in das SSPSR-Register geschoben. Alle Eingangsbits werden an der ansteigenden Flanke der Taktleitung (SCL) abgetastet. Der Wert des Registers SSPSR<7:1> wird mit dem Wert des Registers SSPADD verglichen, und der Vergleich wird an der abfallenden Flanke des achten Taktimpulses (SCL) durchgeführt. Wenn die Adressen übereinstimmen und die BF- und SSPOV-Bits Null sind, treten die folgenden Ereignisse auf:

- Der Wert des SSPSR-Registers wird in das SSPBUF-Register geladen.
- Das Puffer-Voll-Flag-Bit BF wird auf 1 gesetzt. • ACK-Impuls erzeugen.
- Auf der fallenden Flanke des neunten SCL-Impulses wird das MSSP-Interrupt-Flag-Bit SSPIF des PIR1-Registers gesetzt (Interrupt, falls aktiviert). Im 10-Bit-Adressmodus muss das Slave-Gerät zwei Adressbytes empfangen.

Die oberen 5 Bit des ersten Adressbytes geben an, ob es sich um eine 10-Bit-Adresse handelt. R/W (SSPSTAT-Register) muss eine Schreiboperation spezifizieren, damit das Slave-Gerät das zweite Adressbyte empfangen kann. Bei einer 10-Bit-Adresse sollte das erste Byte 11110A9 A8 0 sein, wobei A9 und A8 die beiden höchstwertigen Bits der Adresse sind.

Das Timing der 10-Bit-Adressoperation ist wie folgt, wobei die Schritte 7–9 für den Slave-Sender gelten: 1. Empfangen des ersten (hohen) Bytes der Adresse (das SSPIF-Bit des PIR1-Registers und die BF- und UA-Bits des SSPSTAT-Registers auf 1) gesetzt sind). 2. Aktualisiere das SSPADD-Register mit dem zweiten (niedrigen) Byte der Adresse (löscht das UA-Bit und gibt die SCL-Leitung frei). 3. Lies das SSPBUF-Register (löscht das Bit BF) und löscht das Flag-Bit SSPIF. 4. Erhalte das zweite (niedrige) Byte der Adresse (SSPIF-Bit, BF-Bit und UA-Bit gesetzt). 5. Aktualisieren Sie das SSPADD-Register mit dem ersten (hohen) Byte der Adresse. Bei Übereinstimmung wird die SCL-Leitung freigegeben und das UA-Bit gelöscht. 6. Lies das SSPBUF-Register (löscht das Bit BF) und löscht das Flag-Bit SSPIF. 7. Empfangen Sie die Bedingung „Wiederholter Start“. 8. Erhalte das erste (hohe) Byte der Adresse (SSPIF-Bit und BF-Bit gesetzt). 9. Lies das SSPBUF-Register (löscht das Bit BF) und löscht das Flag-Bit SSPIF.

14.3.14.2 Empfangen

Wenn das R/W-Bit des Adressbytes gelöscht wird und eine Adressübereinstimmung auftritt, wird das R/W-Bit des SSPSTAT-Registers gelöscht. Die empfangene Adresse wird in das SSPBUF-Register geladen. Ein Bestätigungsimpuls (ACK) wird nicht erzeugt, wenn ein Adressbyte-Überlaufzustand besteht. Ein Überlaufzustand liegt vor, wenn das

BF-Bit (SSPSTAT-Register) gesetzt ist oder das SSPOV-Bit (SSPCON-Register) gesetzt ist. Jedes Datenübertragungsbyte erzeugt einen MSSP-Interrupt. Das Interrupt-Flag-Bit SSPIF des PIR1-Registers muss per Software gelöscht werden. Das SSPSTAT-Register wird verwendet, um den Status dieses Bytes zu bestimmen.

14.3.14.3 Senden

Das R/W-Bit des SSPSTAT-Registers wird gesetzt, wenn das R/W-Bit des empfangenen Adressbytes gesetzt ist und eine Adressübereinstimmung auftritt. Die empfangene Adresse wird in das SSPBUF-Register geladen. Ein ACK-Impuls wird auf dem 9. Bit gesendet, während der SDA-Stift niedrig gehalten wird. Zu übertragende Daten müssen in das SSPBUF-Register geladen werden, das auch in das SSPSR-Register geladen wird. Der SCL-Pin sollte dann durch Setzen des CKP-Bits (SSPCON-Register) freigegeben werden. Das Master-Gerät muss den SCL-Pin überwachen, bevor es einen weiteren Taktimpuls sendet. Das Slave-Gerät kann die Datenübertragung mit dem Master-Gerät pausieren, indem es den Takt verlängert. Bei der fallenden Flanke des SCL-Eingangs werden acht Datenbits herausgeschoben. Dies stellt sicher, dass das SDA-Signal gültig ist, während SCL hoch ist.

Jedes Datenübertragungsbyte erzeugt einen MSSP-Interrupt. Das SSPIF-Flagbit muss per Software gelöscht werden, und das SSPSTAT-Register wird verwendet, um den Status des Bytes zu bestimmen. Das SSPIF-Bit wird an der fallenden Flanke des neunten Taktimpulses gesetzt. Der ACK-Impuls vom Master-Empfänger wird an der ansteigenden Flanke des 9. Impulses am SCL-Eingang zwischengespeichert. Wenn die SDA-Leitung hoch ist (kein ACK), dann ist die Datenübertragung abgeschlossen. Wenn der Slave in diesem Fall ACK zwischengespeichert hat, wird die Slave-Logik zurückgesetzt (Zurücksetzen des SSPSTAT-Registers), während der Slave auf das nächste Startbit überwacht. Wenn die SDA-Leitung niedrig ist (ACK), muss das SSPBUF-Register mit den nächsten zu übertragenden Daten geladen werden, was auch das SSPSR-Register laden wird. CKP sollte auf 1 gesetzt werden, um RB1/SCK/SCL zu aktivieren.

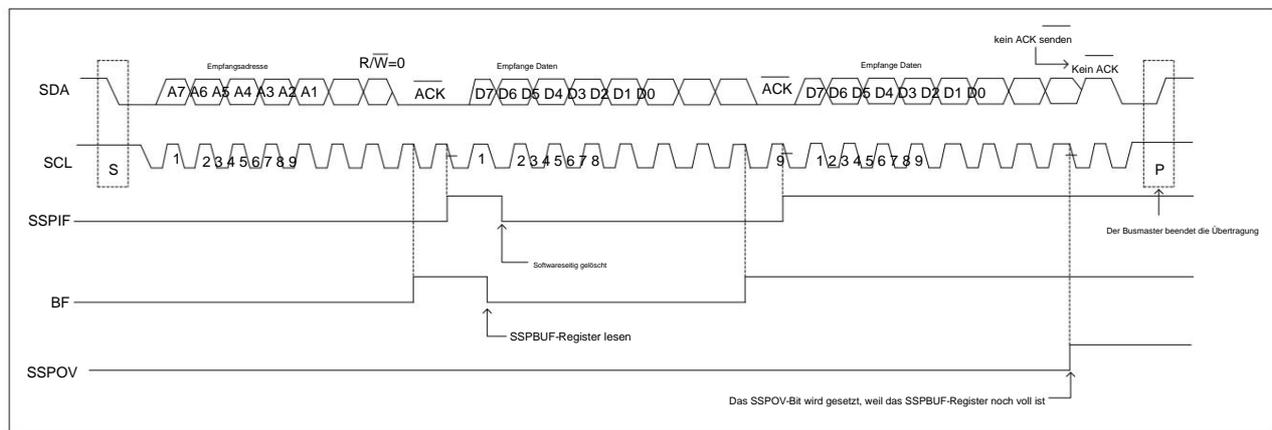


Abbildung 14-17: I2CTM Slave-Modus Empfangstiming (7-Bit-Adresse)

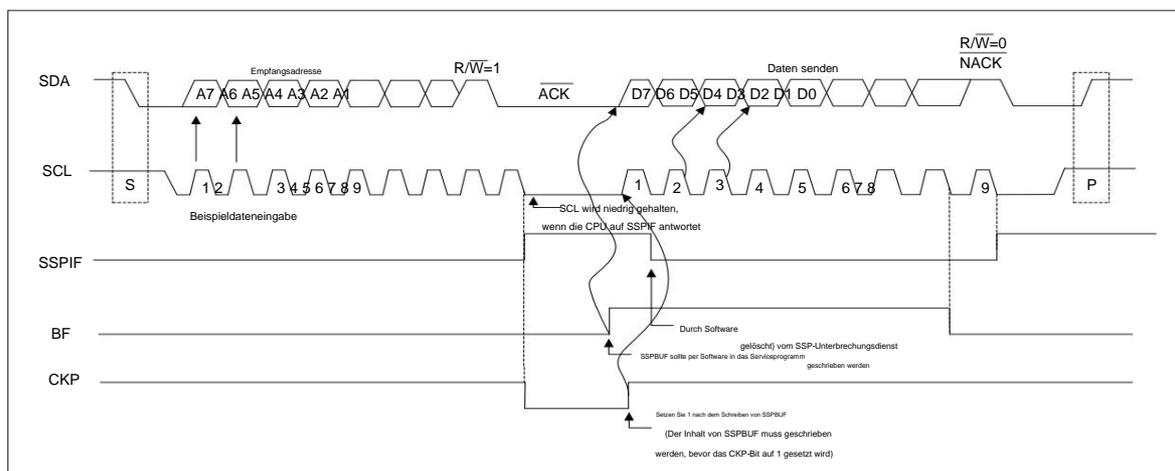


Abbildung 14-18: I2CTM-Slave-Modus-Übertragungstiming (7-Bit-Adresse)

14.3.15 SSP- Maskenregister

Im I2C-Slave-Modus wird das SSP-Maskenregister (SSPMSK) verwendet, um den Wert im SSPSR-Register während Adressvergleichsoperationen zu maskieren.

Eine '0' im SSPMSK-Register macht das entsprechende Bit im SSPSR-Register zu einem "don't care". Dieses Register wird

bei jeder Reset-Bedingung vollständig auf 1 zurückgesetzt, sodass es keine Auswirkung auf den Standard-SSP-Betrieb hat, bis der Maskenwert geschrieben wird. Dieses Register muss initialisiert werden, bevor der I2C-Slave-Modus (7-Bit- oder 10-Bit-Adresse) ausgewählt wird, indem die SSPM<3:0>-Bits gesetzt werden. Auf dieses Register kann nur zugegriffen werden, wenn der geeignete Modus über die SSPM<3:0>-Bits von SSPCON ausgewählt wird.

Das SSP-Maskenregister ist in folgenden Fällen gültig: - 7-

Bit-Adressmodus: Adressvergleich mit A<7:1>. - 10-Bit-Adressmodus:

Adressvergleich nur mit A<7:0>.

Die SSP-Maske ist während des Empfangs des ersten (hohen) Bytes der Adresse inaktiv.

SSPMSK: SSP-Maskenregister (191H) (1)

191H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPMSK	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0(2)
Rücksetzwert	R/W							
lesen und schreiben	1	1	1	1	1	1	1	1

Bit7–Bit1

MSK<7:1>: Bits maskieren.

1 = Bit n der empfangenen Adresse wird mit SSPADD<n> verglichen, um I2C-Adressenübereinstimmung zu erkennen. 0 = Bit n der empfangenen Adresse wird nicht verwendet, um I2C-Adressenübereinstimmung zu erkennen.

Bit 0

MSK<0>: I2C-Slave-Modus 10-Bit-Adressmaskenbit(2).

I2C -Slave-Modus, 10-Bit-Adresse
(SSPM<3:0> = 0111):

1 = Bit0 der empfangenen Adresse wird mit SSPADD<0> verglichen, um eine Übereinstimmung der I2C-Adresse zu erkennen. 0 = Bit0 der empfangenen Adresse wird nicht verwendet, um I2C-Adressenübereinstimmung zu erkennen.

Anmerkungen: 1. Wenn die SSPCON-Bits SSPM<3:0> = 1001 sind, wird jede Lese- oder Schreiboperation an der SSPADDSFR-Adresse durch SSPMSK registriert

Behalten. 2. In allen anderen SSP-Modi hat dieses Bit keine Wirkung.

14.3.16 Betrieb im Schlafmodus

Im Schlafmodus ist der I2C-Block in der Lage, Adressen oder Daten zu empfangen. Und nachdem eine Adressübereinstimmung oder eine Byteübertragung abgeschlossen ist, wacht der Prozessor auf (wenn der MSSP-Interrupt aktiviert ist).

14.3.17 Auswirkung des Zurücksetzens

Ein Reset deaktiviert das MSSP-Modul und beendet die aktuelle Übertragung.

15. Programmierbarer Impulsgenerator PPG

15.1 Arbeitsprinzip von PPG

Für die Anwendung der Induktionsherdlösung integriert CMS89F52x einen programmierbaren Impulsgenerator (Programmable Pulse Generator, im Folgenden als PPG bezeichnet), der aus einem 10-Bit-Timer PPGTMR, 5 hochpräzisen Komparatoren besteht: Synchronkomparator (COMP1), Überspannung Komparator (COMP2), Überspannungskomparator 1 (COMP3), Stromstoßkomparator (COMP4), Stromspannungsstoßkomparator (COMP5) und einen unabhängigen Watchdog-Zähler PPGWDT.

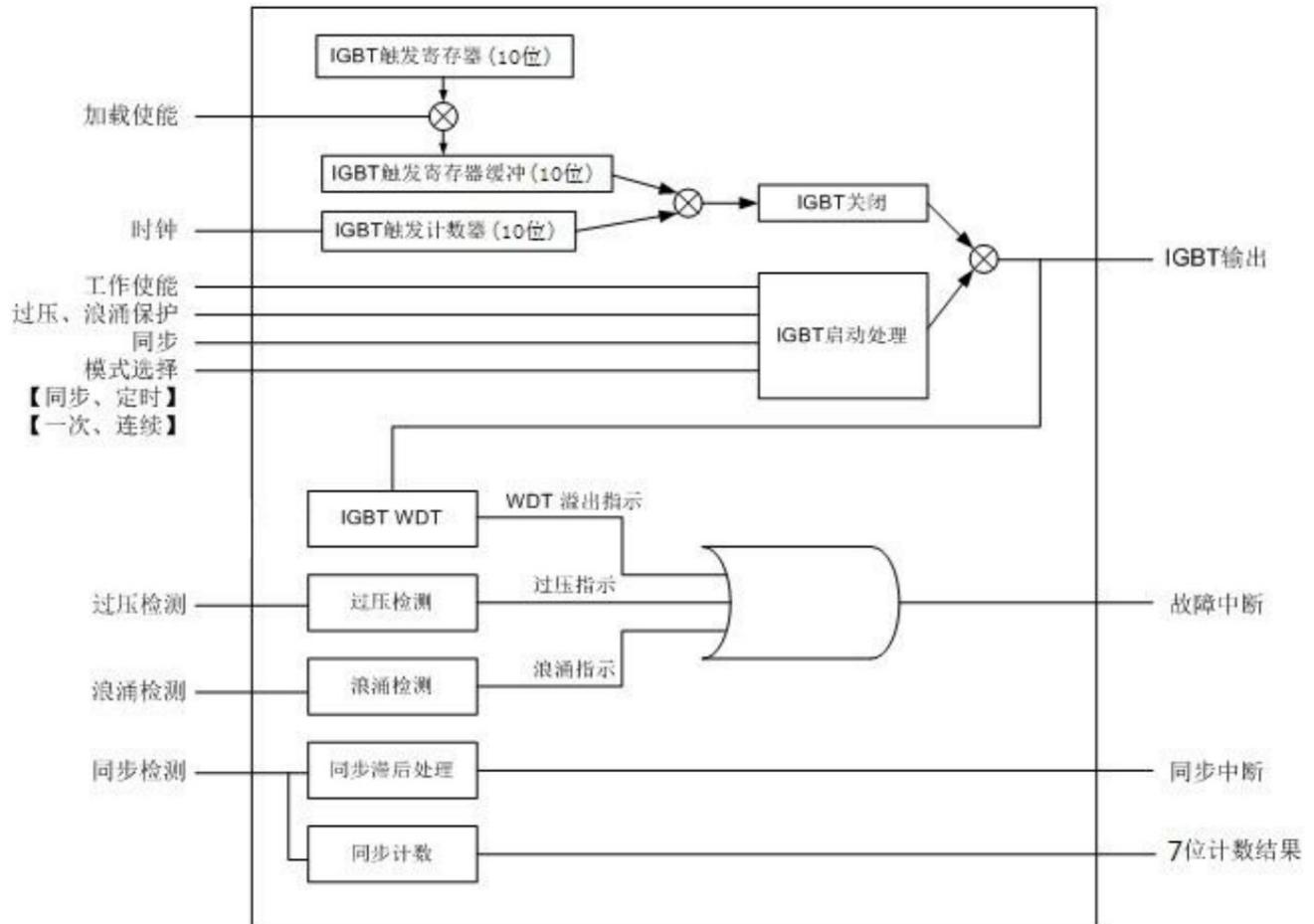


Abbildung 15-1: PPG-Arbeitsprinzip

Das PPG-Ausgangssignal ist ein Signal, das nur einen Zustand mit niedrigem Pegel oder hoher Impedanz ausgeben kann. Wenn die PPG-Funktion ausgeschaltet ist, verhält es sich wie ein Zustand mit hoher Impedanz.

PPGTMR ist ein 10-Bit-Zeitgeber, die unteren 8 Bits werden im Register PPGTMRL (14H) gespeichert, und die oberen 2 Bits werden im Register PPGTMRH (15H) gespeichert. Wenn der PPG-Ausgang ausgeschaltet wird, wird der interne 10-Bit-Zähler gelöscht. Wenn der PPG-Ausgang eingeschaltet wird, beginnt der Zähler zu zählen und erhöht sich automatisch um 1 für jeden Oszillationszyklus. Wenn der Zähler gleich dem PPGTMR-Wert ist, wird der Der PPG-Ausgang wird automatisch ausgeschaltet und der Zähler wird automatisch gelöscht.

PPGTMR niedriges 8-Bit-Register PPGTMRL (14H)

14H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPGTMRL	PPGTMR untere 8 Bit							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

PPGTMR hohes 2-Bit-Register PPGTMRH(15H)

15 Uhr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPGTMRH	----	----	----	----	----	----	PPGTMR hoch 2 Bit	
R/W-	----	----	----	----	----	----	R/W	R/W
Reset-Wert	----	----	----	----	----	----	0	0

15.2 PPG- bezogene Pins

Es gibt 6 PPG-Pins, wie in der folgenden Tabelle

gezeigt: Pin-Name	E/A-Typ	Pin-Beschreibung
C1N	I	Negativeingang Komparator 1
C1P C2N	I	Komparator 1 positiver Eingang Komparator 2 negativer Eingang
C3	I	Komparator 3 negativer Eingang
C4	I	Komparator 4 negativer Eingang
C5N	I	Komparator 5 negativer Eingang
PPG_OUT	O	PPG-Ausgang

15.3 PPG- Arbeitsmodus

Das PPG-Modul des CMS89F52x hat 2 Arbeitsmodi, nämlich: ÿ

Einzelausgabemodus, ÿ synchroner Ausgabemodus;

Die Steuerregister im Zusammenhang mit dem PPG-Status sind wie folgt:

PPG-Steuerregister PPGCON(17H)

17H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPGCON	DETC5F	DETC4F	RELOAD_EN	DETC5EN	DETC4EN	PPGMD	PPG_ON	
R/W	R/W	R/W	----	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	1	1	----	0	0	0	0	0

- Bit7 DETC5F: Statusbit des Komparators 5 (PPG-Statusbit);
0: Löschen des 0->1-Flip-Flags von Komparator 5 (wenn DETC5EN=1, PPG wird wieder eingeschaltet); 1: Es gibt 0->1-Flip von Komparator 5, Schreiben von 1 ist ungültig (wenn DETC5EN=1, dann schaltet sich PPG aus).
- Bit6 DETC4F: Statusbit des Komparators 4 (PPG-Statusbit);
0: Lösche das 1->0-Flip-Flag von Komparator 4 (wenn DETC4EN=1, wird PPG wieder eingeschaltet); 1: Es gibt 1->0-Flip von Komparator 4, Schreiben von 1 ist ungültig (wenn DETC4EN=1, dann schaltet sich PPG aus).
- Bit5 RELOAD_EN deaktivieren: PPG TMR-Laden aktivieren; 0: Laden zulassen (im automatischen Abschaltmodus
- Bit4 PPG TMR gleichzeitig von -1 gehen lassen); 1: Laden deaktivieren (im automatischen Abschaltmodus auch PPG TMR verbieten).
-1).
- Bit3 DETC5EN: Komparator 5 schließt PPG-Aktivierungsbit;
0: deaktiviert; 1: aktiviert.
- Bit2 DETC4EN: Komparator 4 schließt PPG-Freigabebit;
0: deaktivieren;
1: aktivieren.
- Bit1 PPGMD: PPG-Ausgangsmodus; 0:
Synchrone Ausgabe gemäß Komparator
1; 1: Einzelausgabe.
- Bit0 PPG_ON: PPG-Ausgangsaktivierungsbit;
0: deaktiviert; 1: aktiviert.

15.3.1 Einzelausgangsmodus

Wenn das erste Bit des Systemregisters PPGCON auf "1" gesetzt ist, befindet sich das PPG im Einzelausgabemodus. Setzen Sie zu diesem Zeitpunkt das PPG-Aktivierungsbit (PPGCON.0) auf 1, und der PPG wechselt nach Ausgabe eines niedrigen Pegels für 1 PPGTMR-Zeit wieder in einen hochohmigen Zustand, und das PPG-Aktivierungsbit wird automatisch gelöscht, und das PPG hört auf zu arbeiten. Der Einzelausgangsmodus wird im Allgemeinen verwendet, um zu erkennen, ob sich ein Topf im Induktionsherd befindet.

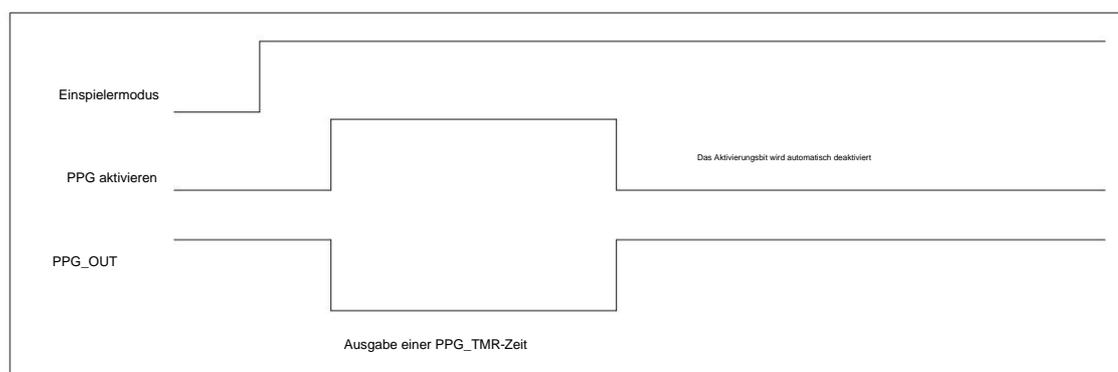


Abbildung 15-2: Timing im PPG-Einzelausgabemodus

15.3.2 Synchroner Ausgabemodus

Synchrone Ausgabe bedeutet, dass die PPG-Ausgabe mit der Invertierung von Komparator 1 synchronisiert ist. Wenn das erste Bit des Systemregisters PPGCON auf "0" gesetzt ist, befindet sich das PPG-Modul im synchronen Ausgangsmodus. In diesem Modus gibt das PPG ein niedriges Signal aus, wenn das Freigabesignal von "0" auf "1" wechselt. Schalten Sie den Ausgang nach dem Pegel aus und geben Sie dann jedes Mal automatisch und kontinuierlich aus, wenn der Ausgang von Komparator 1 von "1" auf "0" wechselt.

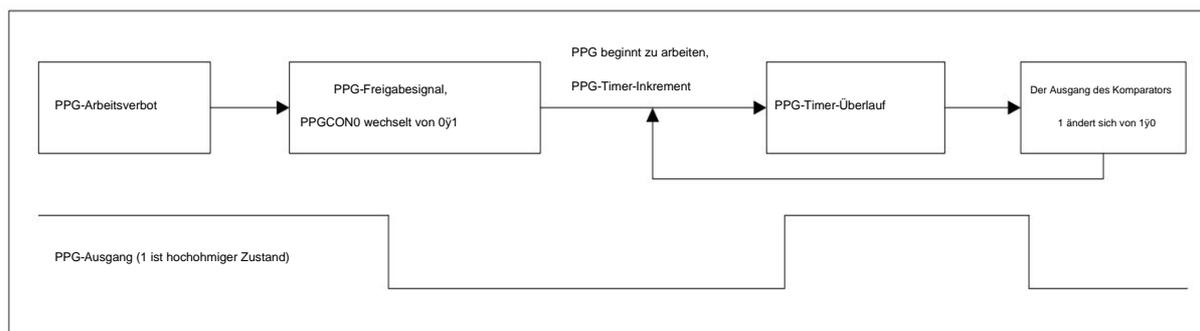


Abbildung 15-3: Synchroner PPG-Ausgabemodus

15.4 Komparatoren

Das PPG-Modul hat 5 Komparatoren: Synchronkomparator (COMP1), Überspannungskomparator (COMP2), Überspannungskomparator 1 (COMP3), Stromstoßkomparator (COMP4), Spannungstoßkomparator (COMP5).

15.4.1 Synchronkomparator COMP1

Die Funktion des Synchronkomparators besteht darin, dem PPG ein Synchronsignal bereitzustellen. Wenn PPG im Synchronmodus arbeitet, kann PPG_OUT nur dann einen niedrigen Pegel ausgeben, wenn der Ausgang des Komparators von "1" auf "0" wechselt, flach. Sie können den Synchronkomparator so einstellen, dass er umkehrt und dann eine gewisse Zeit verzögert, bevor Sie den PPG einschalten. Die kürzeste Zeit ist $0 \cdot T_{sys}$ und die längste ist $64 \cdot T_{sys}$.

Gesteuert durch die unteren 4 Bits von PPGDLY.

Vergleicher 1 Steuerregister CM1CON (97H)

97H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM1CON	CM1EN	CM1COFM	CM1CEN	CM1CLR	CM1NSL	----	----	----
R/W	R/W	R/W	R/W	R/W	R/W	----	----	----
Wert zurücksetzen	0	0	0	0	0	----	----	----

- Bit7** CM1EN: Freigabebit Komparator 1; 0: Deaktiviert, Komparator funktioniert nicht, Ausgang
0; 1: Aktiviert, COMP1+, COMP1- sind Komparatoreingangsanschlüsse.
- Bit6** CM1COFM: Auswahl des Einstellmodus für Komparator
1; 0: normaler Betriebsmodus; 1: Einstellmodus.
- Bit5** CM1CEN: Toggle-Zähler von Komparator 1 aktivieren;
0: Zählung deaktivieren;
1: Zählung aktivieren.
- Bit4** CM1CLR: Komparator 1 Zählung gelöscht;
0: gelöscht; 1: normale Zählung.
- Bit3** CM1NSL: Komparator 1 negative interne Masseauswahl, nur gültig im Regelungsmodus;
0: COMP1 – mit E/A-Port verbinden (wenn CM1EN = 1, C1N-Kanalport als negativer Eingang des Komparators 1 aktivieren);
1: COMP1 – mit GND verbinden (C1N-Kanalport ist gemeinsamer E/A-Port). Behinderte

Bit2~Bit0

PPG-Verzögerungszeit-Steuerregister PPGDLY (16H)

16H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPGDLY	----	----	----	----	PPGDLY			
R/W	----	----	----	----	R/W	R/W	R/W	R/W
Wert zurücksetzen	----	----	----	----	0	0	0	0

Bit7~Bit4

Behindert

Bit3~Bit0

PPGDLY: PPG-Verzögerungsausgabe;

0000: keine Verzögerung;

0001: 4-5* T_{sys} ;0010: 8-9* T_{sys} ;

....

1111: 64-65* T_{sys} .

Ein synchroner Komparator hat eine Toggle-Count-Funktion, die verfolgt, wie oft sein Ausgang umschaltet. Um diese Funktion gültig zu machen, muss die Funktion Bit 5 von CM1CON auf "1" setzen. Wenn die PPG-Zählfunktion aktiviert ist und der Ausgang des Synchronkomparators von "1" auf "0" wechselt, erhöht sich der Zähler automatisch um 1 bis 128. Das Zählergebnis wird im CM1CNT-Register gespeichert, die unteren 7 Bits geben den Zählwert an, das höchste Bit zeigt an, ob die Zählung überläuft, wenn die Zählung 128 überschreitet, ist das höchste Bit 1 und die Zählung stoppt. Der Zähler kann durch Schreiben von „0“ auf Bit 4 von CM1CON gelöscht werden.

Flip-Count-Register des Komparators 1 CM1CNT (93H)

93H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM1CNT	CM1OF	CM1COUNT[6:0]						
R/W	R	R	R	R	R	R	R	R
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7

CM1OF: Rollover-Zähler-Überlauf-Flag des Komparators 1; 0: kein Überlauf; 1: Überlauf.

Bit6~Bit0

CM1COUNT[6:0]: 1->0 Rollover-Zähler für Komparator 1, nur lesbar.

15.4.2 Überspannungskomparator COMP2 und Surge-Komparator COMP4/COMP5

Sowohl der Überspannungskomparator als auch der Überspannungskomparator werden verwendet, um den Ausgang des PPG zu begrenzen, um den IGBT zu schützen. Der negative Anschluss des Überspannungskomparators wird vom Port RB2 eingegeben, der negative Anschluss des Spannungsstoßkomparators wird vom Port RB0 eingegeben und der negative Anschluss des Stromstoßkomparators wird vom Port RB4 eingegeben. Ihre positiven Enden befinden sich im Inneren des Chips, und der Widerstandsteiler wird per Software eingestellt. Wenn sich die negative Klemmenspannung des Überspannungskomparators von niedriger als die positive Klemmenspannung auf höher als die positive Klemmenspannung ändert, nennen wir dies die Überspannung des IGBT. Wenn die Anzahl der Überspannungen die von der Software eingestellte Anzahl erreicht, deaktiviert das PPG die Ausgabe oder reduziert die Ausgabzeit (von der Software eingestellt) und generiert ein Unterbrechungs-Flag, das gelöscht werden muss, bevor das PPG wieder normal werden kann.

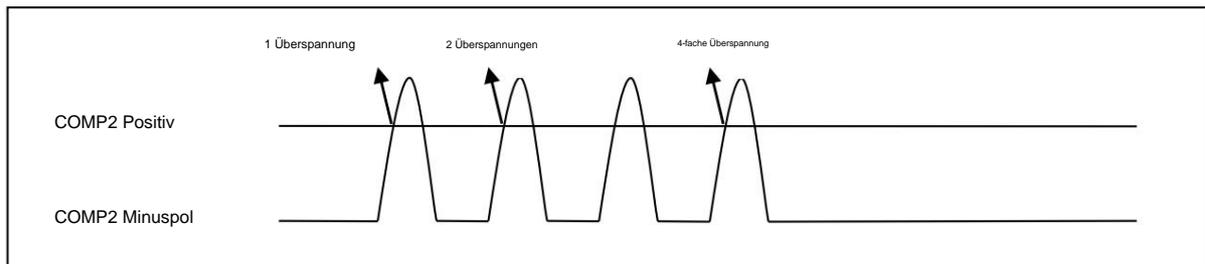


ABBILDUNG 15-4: ÜBERSpannungskomparator COMP2

Wenn der Überspannungskomparator den eingestellten effektiven Pegel erreicht und die Zeit der von der Software eingestellten Zeit entspricht, deaktiviert der PPG den Ausgang und erzeugt eine Interrupt-Flag-Bit muss auf Null gelöscht werden, bevor PPG zum Normalzustand zurückkehren kann. Die positive Anschlussspannung des Überspannungskomparators kann optional geerdet werden.

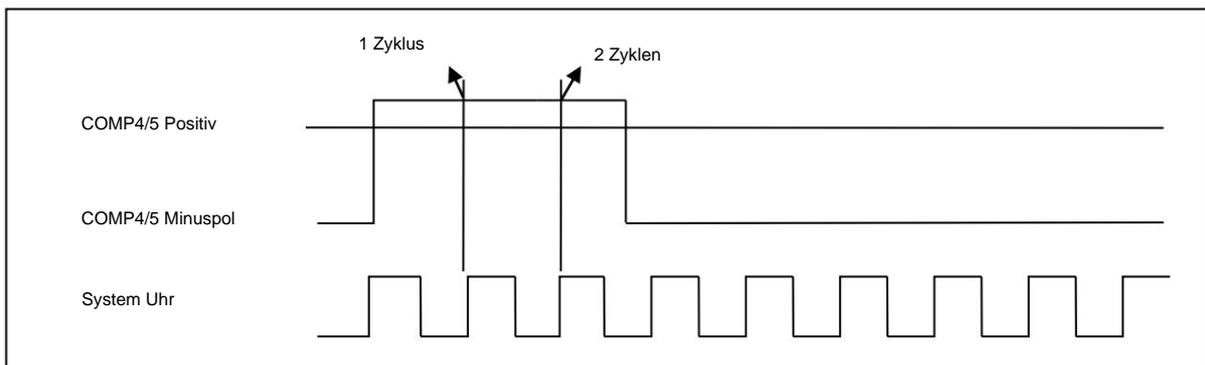


Abbildung 15-5: Schematische Darstellung des aktiven Low-Ausgangs

Zu COMP2, COMP4, COMP5 gehörige Register:

Vergleicher 2 Steuerregister CM2CON (98H)

98H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM2CON	CM2EN CM2COFM	CM2DBSEL			CM2PVSL			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7 CM2EN: Aktivierungsbit Komparator 2; 0:

Deaktiviert, Komparator 2 funktioniert nicht, Ausgang 0; 1:

Aktiviert, Zählerüberlauf von Komparator 2 wirkt sich auf PPG aus oder tritt in Unterbrechung ein.

Bit6 CM2COFM: Auswahl des Einstellmodus für Komparator 2;

0: normaler Betriebsmodus; 1:

Einstellmodus.

Bit5~Bit4 CM2DBSEL: Auswahl Filterzeit Komparator 2;

00: $\leq 1T_{sys}$ (ausfilterbare Impulsbreite); 01: $\leq 4T_{sys}$; 10: $\leq 8T_{sys}$; 11: $\leq 16T_{sys}$.

Bit3~Bit0 CM2PVSL: Auswahl der Spannung an der internen positiven Klemme des Komparators 2.

0000~1111: 0,4 VDD~0,775 VDD (16 Stufen insgesamt, jede Stufe ist 0,025 VDD).

Vergleicher 2 Steuerregister 1CM2CON1 (99H)

99H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM2CON1 ATPEN		----	----	----	CM2COF	CM2CNT		
R/W	R/W	----	----	----	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	----	----	----	1	0	0	0

Bit7 ATPEN: Der Zählerüberlauf des Komparators 2 senkt automatisch das Freigabebit PPG_TMR;

0: Deaktiviert (Sie müssen CM2COF löschen, um mit dem Zählen zu beginnen, nachdem der Zähler von Komparator 2 gelöscht wurde);

1: Aktiviert, der Wert von PPT_TMR wird jedes Mal automatisch um 1 verringert, wenn ein Überlauf erkannt wird

(Der Zähler des Komparators 2 beginnt automatisch mit dem Zählen, nachdem er gelöscht wurde).

Bit6~Bit4

Behinderte

Bit3

CM2COF: Zählerüberlauf-Flag des Vergleichers 2, kann per Software gelöscht werden; 0: Zähler

des Vergleichers 2 hat keinen Überlauf, schreiben Sie 0, um ihn zu löschen (wenn ATPEN=0, Zähler des Vergleichers 2 aktiviert das Zählen);

1: Zähler des Vergleichers 2 ist übergelaufen Zähler läuft über, Schreiben von 1 ist ungültig (wenn ATPEN = 0, bleibt der Zähler des Komparators 2 gelöscht).

Bit2~Bit0

CM2COS[2:0]: Wählen Sie die Anzahl der Impulse aus, die für den Zählerüberlauf des Komparators 2 erforderlich sind (Ausgang 1->0 des Komparators 2 springt zum Zähltriggereingang);

000: 1 Mal Überlauf oder ein Interrupt wird generiert und der Zähler

wird gelöscht 001: 2 Mal 010: 4 Mal;

...

111: 128 Mal.

Komparator 4 Steuerregister CM4CON (Stromstoß) (9CH)

9CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM4CON CM4EN CM4COFM			CM4DBSEL		CM4PVSL			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7 CM4EN: Freigabebit Komparator 4 (Ausgang hoch aktiv);

0: deaktiviert, Komparator 4 funktioniert nicht, Ausgang 0; 1:

aktiviert, die 0->1-Umkehr von Komparator 4 wirkt sich auf PPG aus oder tritt in einen Interrupt ein.

Bit6 CM4COFM: Auswahl des Einstellmodus für Komparator 4;

0: normaler Betriebsmodus; 1:

Einstellmodus.

Bit5~Bit4 CM4DBSEL: Filterzeitauswahl des Komparators 4;

00: $\leq 1T_{sys}$ (ausfilterbare Impulsbreite); 01:

$\leq 4T_{sys}$; 10: $\leq 8T_{sys}$; 11: $\leq 16T_{sys}$.

Bit3~Bit0 CM4PVSL: Interne Spannungsauswahl des positiven Anschlusses des Komparators 4;

0000: Intern mit GND verbinden;

0001-1111: 0,050 VDD-0,400 VDD (16 Stufen insgesamt, jede Stufe ist 0,025 VDD).

Komparator 5 Steuerregister CM5CON (Spannungsstoß) (9DH)

9DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM5CON CM5EN CM5COFM			CM5DBSEL		CM5PVSL			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7 CM5EN: Komparator 5 Aktivierungsbit (Ausgang hoch aktiv);

0: deaktiviert, Komparator 5 funktioniert nicht, Ausgang 0; 1:

aktiviert, die 0->1-Umkehr von Komparator 5 wirkt sich auf PPG aus oder tritt in einen Interrupt ein.

Bit6 CM5COFM: Auswahl des Einstellmodus für Komparator 5;

0: normaler Betriebsmodus; 1:

Einstellmodus.

Bit5~Bit4 CM5DBSEL: Filterzeitauswahl des Komparators 5;

00: $\leq 1T_{sys}$ (ausfilterbare Impulsbreite); 01:

$\leq 4T_{sys}$; 10: $\leq 8T_{sys}$; 11: $\leq 16T_{sys}$.

Bit3~Bit0 CM5PVSL: Interne Spannungsauswahl des positiven Anschlusses des

Komparators 5; 0000-1111: 0,4 VDD-0,775 VDD (insgesamt 16 Stufen, jede Stufe ist 0,025 VDD).

15.4.3 Überspannungskomparator 1 - COMP3

Der negative Anschluss des Überspannungskomparators 1 wird vom Anschluss RB3 eingegeben, und der positive Anschluss befindet sich im Inneren des Chips, und der

Widerstandsteiler kann per Software eingestellt werden. Zu COMP3 gehörige Register: Steuerregister des Komparators 3 CM3CON(9AH)

9AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM3CON	CM3EN	CM3COFM	CM3DBSEL		CM3PVSL			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7 CM3EN: Komparator 3 Aktivierungsbit (Ausgang hoch aktiv); 0:

Deaktivieren, Komparator 3 funktioniert nicht, Ausgang 0; 1:

Aktivieren, Komparator 3 Zählerüberlauf wirkt sich auf PPG aus oder tritt in Unterbrechung ein.

Bit6 CM3COFM: Auswahl des Einstellmodus für Komparator 3;

0: normaler Betriebsmodus; 1:

Einstellmodus.

Bit5~Bit4 CM3DBSEL Komparator 3 Filterzeitauswahl;

00: $\leq 1T_{sys}$ (ausfilterbare Impulsbreite); 01:

$\leq 4T_{sys}$; 10: $\leq 8T_{sys}$; 11: $\leq 16T_{sys}$.

Bit3~Bit0 CM3PVSL: Interne Spannungsauswahl des positiven Anschlusses des Komparators 3;

00001-1111: 0,4 VDD-0,775 VDD (16 Stufen insgesamt, jede Stufe ist 0,025 VDD).

Vergleicher 3 Steuerregister CM3CON1(9BH)

9BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM3CON1 CM3M1 CM3M0			----	CM3CIS	CM3COF	CM3COS[2:0]		
R/W	R/W	R/W	----	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	----	0	1	0	0	0

Bit7~Bit6

CM3M1-CM3M0: Funktionsauswahl Komparator 3;

00: Komparator 3 beeinflusst PPG nicht;

01: Komparator 3 deaktiviert PPG (Ausgang Low ist aktiv);

10: Komparator 3 reduziert PPG_TMR (reduziert üblicherweise PPG_TMR-Modul mit Komparator 2); 11:

Komparator 3 beeinflusst PPG nicht. Behinderte

Bit5

Bit4

CM3CIS: Zählertriggerflanken-Auswahlbit für Vergleicher 3;

0: Komparatorausgang 1->0 geht auf Zähltriggereingang über

1: Komparatorausgang 0->1 geht auf Zähltriggereingang über.

Bit3

CM3COF: Zählerüberlauf-Flag von Vergleicher 3, kann per Software gelöscht werden; 0: Zähler

von Vergleicher 3 hat keinen Überlauf, schreiben Sie 0, um ihn zu löschen (wenn ATPEN = 0, beginnt Zähler von Vergleicher 3 zu zählen); 1:

Zähler von Vergleicher 3 Es gibt einen Überlauf, das Schreiben von 1 ist ungültig (wenn ATPEN = 0, bleibt der Zähler des Komparators 3 gelöscht

Zustand).

Bit2~Bit0

CM3COS[2:0]: Wählen Sie die Anzahl der Impulse aus, die für den Zählerüberlauf des Komparators 3 erforderlich sind

(Die Trigger-Eingangsfanke des Zählers wird von CM3CIS

bestimmt); 000: 1 Mal Überlauf oder ein Interrupt wird generiert und der

Zähler wird gelöscht; 001: 2 Mal; 010: 4 Mal;

...

111: 128 Mal.

15.4.4 Nullstellung des Komparators

Aufgrund des Fehlers des Produktionsprozesses kann der Chip im tatsächlichen Gebrauch eine große Komparator-Offsetspannung haben. Um dieses Problem zu lösen

CMS69F52x integriert die Komparator-Nulleinstellungsfunktion im Inneren. Das relevante RAM ist

wie folgt: Komparator 1 Nullabgleichsregister CM1ADJ(113H)

113H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM1ADJ CM1OUT CM1CRS			CM1ADJ[5:0]					
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	1	0	0	0	0	0

Bit7 CM1OUT: Ausgang Komparator 1, nur lesbar, Schreibvorgang hat keine
Auswirkung; 0: Ausgang Komparator 0; 1: Ausgang Komparator 1.

Bit6 CM1CRS: Auswahl des Eingangsanschlusses im Anpassungsmodus; 0: Eingang
des negativen Anschlusses; 1: Eingang des positiven
Anschlusses.

Bit5~Bit0 CM1ADJ[5:0]: Komparator 1 Offset-Spannungseinstellung;
000000: Minuspol auf Minimum einstellen (normalerweise ist die Offsetspannung positiv)
111111: Pluspol auf Minimum einstellen (normalerweise ist die Offsetspannung negativ).

Vergleicher 2 Nullsetzregister CM2ADJ(114H)

114H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM2ADJ CM2OUT CM2CRS			CM2ADJ[5:0]					
R/W-	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset-Wert	0	0	1	0	0	0	0	0

Bit7 CM2OUT: Ausgang Komparator 2, nur lesbar, Schreibvorgang hat keine Wirkung;
0: Ausgang Komparator 0; 1: Ausgang Komparator 1.

Bit6 CM2CRS: Auswahl des Eingangsanschlusses im Anpassungsmodus; 0: Eingang
des negativen Anschlusses; 1: Eingang des positiven
Anschlusses.

Bit5~Bit0 CM2ADJ[5:0]: Komparator 2 Offset-Spannungseinstellung;
000000: Minuspol auf Minimum einstellen (normalerweise ist die Offsetspannung positiv)
111111: Pluspol auf Minimum einstellen (normalerweise ist die Offsetspannung negativ).

Vergleicher 3 Nullregister CM3ADJ(115H)

115H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM3ADJ CM3OUT CM3CRS	CM3ADJ[5:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	1	0	0	0	0	0

Bit7 CM3OUT: Ausgang Komparator 3, nur lesbar, Schreibvorgang hat keine
Auswirkung; 0: Ausgang Komparator 0; 1: Ausgang Komparator 1.

Bit6 CM3CRS: Auswahl des Eingangsanschlusses im Anpassungsmodus; 0: Eingang
des negativen Anschlusses; 1: Eingang des positiven
Anschlusses.

Bit5–Bit0 CM3ADJ[5:0]: Komparator 3 Offset-Spannungseinstellung;
000000: Minuspol auf Minimum einstellen (normalerweise ist die Offsetspannung positiv)
111111: Pluspol auf Minimum einstellen (normalerweise ist die Offsetspannung negativ).

Vergleicher 4 Nullsetzregister CM4ADJ(116H)

116H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM4ADJ CM4OUT CM4CRS	CM4ADJ[5:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	1	0	0	0	0	0

Bit7 CM4OUT: Ausgang Komparator 4, nur lesbar, Schreibvorgang hat keine
Auswirkung; 0: Ausgang Komparator 0; 1: Ausgang Komparator 1.

Bit6 CM4CRS: Auswahl des Eingangsanschlusses im Anpassungsmodus; 0: Eingang
des negativen Anschlusses; 1: Eingang des positiven
Anschlusses.

Bit5–Bit0 CM4ADJ[5:0]: Komparator 4 Offset-Spannungseinstellung;
000000: Minuspol auf Minimum einstellen (normalerweise ist die Offsetspannung positiv)
111111: Pluspol auf Minimum einstellen (normalerweise ist die Offsetspannung negativ).

Vergleicher 5 Nullstellungsregister CM5ADJ(117H)

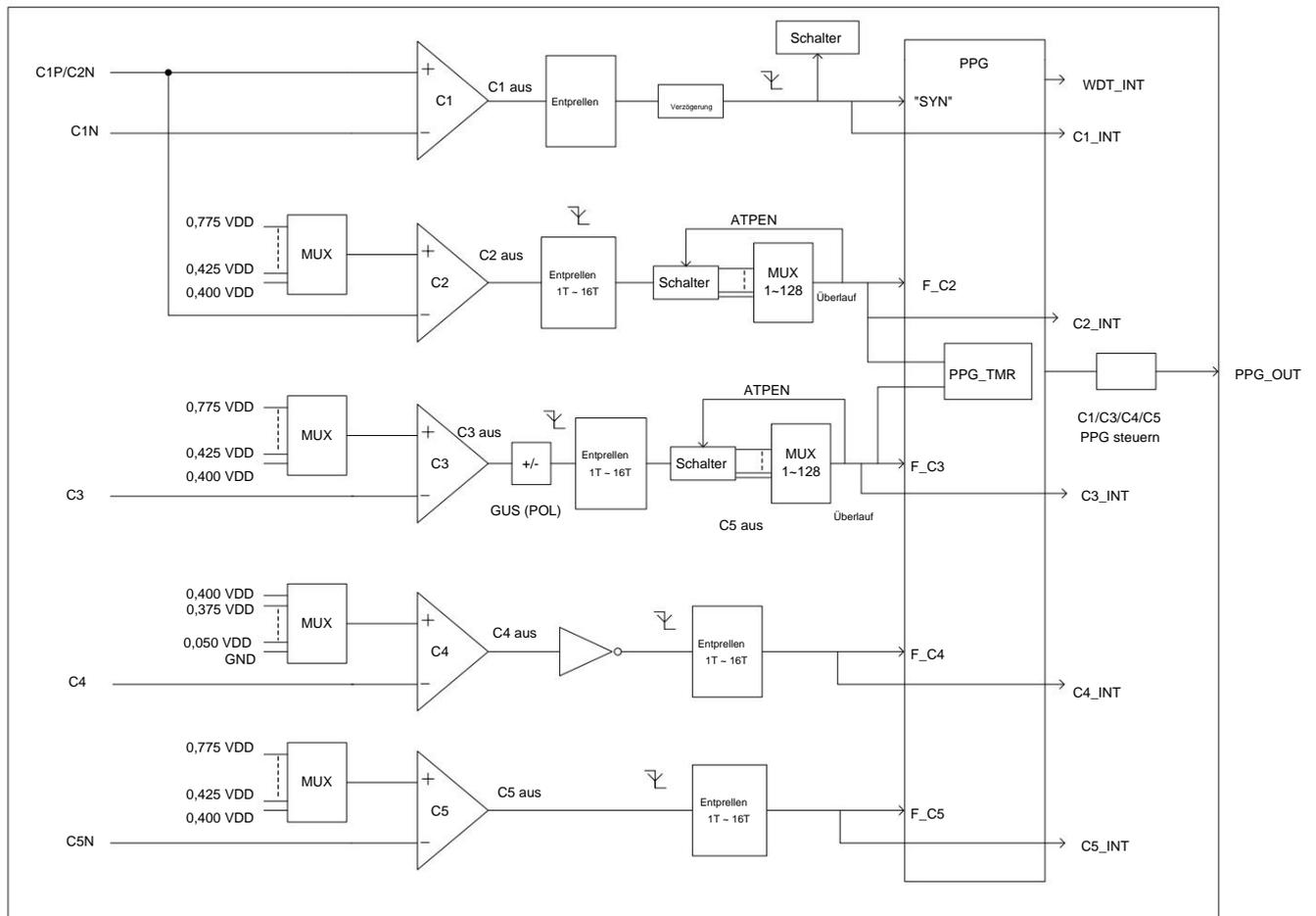
117H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CM5ADJ	CM5OUT	CM5CRS	CM5ADJ[5:0]					
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	1	0	0	0	0	0

Bit7 CM5OUT: Ausgang Komparator 5, nur lesbar, Schreibvorgang hat keine
Auswirkung; 0: Ausgang Komparator 0; 1: Ausgang Komparator 1.

Bit6 CM5CRS: Auswahl des Eingangsanschlusses im Anpassungsmodus; 0: Eingang
des negativen Anschlusses; 1: Eingang des positiven
Anschlusses.

Bit5–Bit0 CM5ADJ[5:0]: Offset-Spannungseinstellung des Komparators 5;
000000: Minuspol auf Minimum einstellen (normalerweise ist die Offsetspannung positiv)
111111: Pluspol auf Minimum einstellen (normalerweise ist die Offsetspannung negativ).

15.4.5 Komparator und internes Strukturdiagramm des PPG



16. Daten-EEPROM-Steuerung

16.1 Daten- EEPROM - Übersicht

Das Daten-EEPROM ist unter normalen Betriebsbedingungen lesbar und beschreibbar. Diese Speicher werden nicht direkt in den Registerdateiraum abgebildet, sind es aber Es wird indirekt über Spezialfunktionsregister (SFRs) adressiert. Es gibt fünf SFR-Register, die für den Zugriff auf diese Speicher verwendet werden:

• EECON1

• EECON2

• EEDAT

• EEDATH

• EEADR

Beim Verbinden mit dem Datenspeicherblock bilden die EEDAT- und EEDATH-Register ein Zwei-Byte-Wort, das verwendet wird, um die zu lesenden/schreibenden 16-Bit-Daten zu halten, während das EEADR-Register die Adresse der EEDAT-Stelle hält, auf die zugegriffen wird. Geräte dieser Familie verfügen über 32 Worte Daten-EEPROM mit einem Adressbereich von 0H bis 01FH.

Der EEPROM-Datenspeicher ermöglicht das Lesen und Schreiben von Bytes. Eine Byte-Schreiboperation löscht automatisch die Zielzelle und schreibt neue Daten (Löschen vor Schreiben). Die Schreibzeit wird durch einen On-Chip-Timer gesteuert. Die Schreib- und Löschspannungen werden von einer On-Chip-Ladungspumpe erzeugt, die für den Betrieb innerhalb des Spannungsbereichs des Geräts für Byte- oder Wortoperationen ausgelegt ist. Wenn das Gerät codegeschützt ist, kann die CPU weiterhin Daten im EEPROM lesen und schreiben. Wenn der Code geschützt ist, kann der Geräteprogrammierer nicht mehr auf die Daten zugreifen.

16.2 Zugehörige Register

16.2.1 EEADR- Register

Das EEADR-Register kann bis zu 32 Bytes Daten-EEPROM adressieren.

16.2.2 EECON1- und EECON2- Register

EECON1 ist das Steuerregister für den Zugriff auf den EE-Speicher.

Zum Betrieb des Datenspeichers muss das Steuerbit EEPGD auf 1 gesetzt werden. Wenn dieses Bit wie beim Zurücksetzen gelöscht wird, ist jede nachfolgende Operation ungültig von.

Die Steuerbits RD und WR geben das Lesen bzw. Schreiben frei. Diese Bits können nur per Software auf 1 gesetzt und nicht gelöscht werden. Nachdem der Lese- oder Schreibvorgang abgeschlossen ist, wird die Hardware sie auf Null zu setzen. Da das WR-Bit in Software nicht gelöscht werden kann, wird ein versehentliches vorzeitiges Beenden von Schreiboperationen verhindert.

- Wenn WREN gesetzt ist, sind Schreibvorgänge in das Daten-EEPROM aktiviert. Beim Einschalten wird das WREN-Bit gelöscht. Das WRERR-Bit wird gesetzt, wenn eine normale Schreiboperation durch ein LVR-Reset oder ein WDT-Timeout-Reset unterbrochen wird. In diesen Fällen kann der Benutzer das WRERR-Bit überprüfen und die entsprechende Stelle nach dem Zurücksetzen neu schreiben.
- Das Interrupt-Flag-Bit EEIF im PIR2-Register wird auf 1 gesetzt, wenn die Schreiboperation abgeschlossen ist. Dieses Flag muss per Software gelöscht werden.

EECON2 ist kein physisches Register. Beim Lesen von EECON2 werden nur Nullen ausgegeben.

Das EECON2-Register wird nur verwendet, wenn eine Daten-EEPROM-Schreibsequenz durchgeführt wird.

EEPROM-Datenregister EEDAT(10CH)

10CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0
R/W lesen und schreiben		R/W						
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7–Bit0 EEDAT<7:0>: Die unteren 8 Bits der Daten zum Lesen oder Schreiben in den Daten-EEPROM.

EEPROM-Adressregister EEADR(10DH)

10DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEADR	----	----	----	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0
lesen und schreiben	----	----	----	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	----	----	----	0	0	0	0	0

Bit7–Bit5 ----(egal)

Bit4–Bit0 EEADR<4:0>: Gibt die Adresse für den EEPROM-Lese-/Schreibvorgang an.

EEPROM-Datenregister EEDATH(10EH)

10EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEDATH	EEDATH7	EEDATH6	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0
Lesen und Schreiben	R/W							
Wert zurücksetzen	0	0	0	0	0	0	0	0

Bit7~Bit0 EEDATH<7:0>: Die oberen 8 Bits der Daten, die in das Daten-EEPROM gelesen oder geschrieben werden.

EEPROM-Steuerregister EECON1(18CH)

18CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EECON1	EEPGD	----	----	----	WRERR	WREN	WR	RD
Lesen und Schreiben	R/W	----	----	----	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	----	----	----	X	0	0	0

Bit7 EEPGD: Daten-EEPROM-Aktivierungsbit;

1 = Daten-EEPROM-Betrieb aktivieren; 0 = Daten-EEPROM-Betrieb deaktivieren.

Bit6~Bit4 Nicht implementiert, als 0 gelesen WRERR: EEPROM-

Bit3 Fehler-Flag;

1 = Schreibvorgang vorzeitig abgebrochen (beliebiger WDT-Reset oder Brown-Out-Reset im Normalbetrieb) 0 = Schreibvorgang abgeschlossen.

Bit2 WREN: EEPROM-Schreibaktivierungsbit; 1 =

Schreibzyklus aktivieren; 0 = Schreiben von Daten EEPROM verbieten.

Bit1 WR: Steuerbit schreiben;

1 = Schreibzyklus einleiten (dieses Bit wird von der Hardware gelöscht, sobald die Schreiboperation abgeschlossen ist, das WR-Bit kann nur von Software auf 1 gesetzt, aber nicht gelöscht werden); 0 = Daten-EEPROM-Schreibzyklus ist abgeschlossen.

Bit0 RD: Steuerbit lesen;

1 = Speicherleseoperation starten (RD wird durch Hardware gelöscht, nur das RD-Bit kann durch Software auf 1 gesetzt, aber nicht gelöscht werden); 0 = Speicherleseoperation nicht starten.

EEPROM-Steuerregister EECON1(18DH)

18DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EECON2	---							
Lesen und Schreiben	W							

EECON2 ist kein physisches Register. Beim Lesen von EECON2 werden nur Nullen ausgegeben.

Das EECON2-Register wird nur verwendet, wenn eine Daten-EEPROM-Schreibsequenz durchgeführt wird.

16.3 Lesen von Daten EEPROM - Speicher

Um eine Datenspeicherstelle zu lesen, muss der Benutzer die Adresse in das EEADR-Register schreiben, das EEPGD-Steuerbit des EECON1-Registers setzen und dann das Steuerbit RD setzen. Sobald das Lesesteuerbit gesetzt ist, verwendet die Datenspeichersteuerung den zweiten Befehlszyklus, um die Daten zu lesen. Dies bewirkt, dass die zweite Anweisung nach der Anweisung "SETB EECON1,RD" ignoriert wird (1). Beim nächsten Zyklus befinden sich Daten im EEDAT-Register. EEDAT hält diesen Wert, bis der Benutzer das nächste Mal Daten liest oder auf das Gerät schreibt.

Hinweis: Die zwei Anweisungen, die einem Lesen des Programmspeichers folgen, müssen NOPs sein. Dadurch wird verhindert, dass der Benutzer die nächste Anweisung im Doppelzyklus ausführt, nachdem das RD-Bit gesetzt wurde

Anweisung.

Daten-EEPROM lesen

LD	A,EE_ADD	; Die zu lesende Adresse in das EEADR-Register eintragen
LD	EEADR,A	
SETB	EECON1,EEPGD	;Daten-EEPROM
SETB	EECON1,RD	aktivieren;Lesesignal aktivieren;Um
NOP		hier Daten zu lesen, müssen Sie den NOP-Befehl hinzufügen
NOP		
LD	A, EEDATA	; Daten in ACC lesen

16.4 SCHREIBEN VON DATEN IN EEPROM - SPEICHER

Um in eine EEPROM-Datenspeicherstelle zu schreiben, sollte der Benutzer zuerst die Adresse der Stelle in das EEADR-Register schreiben und die Daten in das EEDATA-Register schreiben.

Speicher. Der Benutzer muss dann beginnen, jedes Byte in einer bestimmten Reihenfolge zu schreiben.

Wenn die folgende Anweisungsfolge nicht genau befolgt wird (d. h. zuerst 55h in EECON2 schreiben, dann Aah in EECON2 schreiben und schließlich

WR-Bit gesetzt) schreibt jedes Byte, der Schreibvorgang wird nicht initiiert. Interrupts sollten in diesem Codesegment deaktiviert werden.

Außerdem muss das WREN-Bit in EECON1 gesetzt werden, um Schreiboperationen zu ermöglichen. Dieser Mechanismus verhindert, dass das EEPROM versehentlich aufgrund von Codeausführungsfehlern (Anomalien) geschrieben wird (d. h. Programmablauf). Der Benutzer sollte das WREN-Bit immer frei lassen, wenn er das EEPROM nicht aktualisiert. Das WREN-Bit kann nicht durch Hardware gelöscht werden. Nachdem ein Schreibprozess initiiert wurde, wird das Löschen des WREN-Bits den Schreibzyklus nicht beeinflussen. Das WR-Bit kann nicht gesetzt werden, wenn das WREN-Bit nicht gesetzt ist. Wenn der Schreibzyklus abgeschlossen ist, wird das WR-Bit durch die Hardware gelöscht und das EE-Write-Complete-Interrupt-Flag-Bit (EEIF) gesetzt. Der Benutzer kann diesen Interrupt aktivieren oder dieses Bit abfragen.

EEIF muss in der Software gelöscht werden.

Nach Ausführung des Befehls SETB EECON1,WR benötigt der Prozessor 2 Befehlszyklen, um die Lösch-/Schreiboperation einzurichten. Der Benutzer muss die eingeben

Platzieren Sie zwei NOP-Befehle nach dem Befehl mit dem gesetzten WR-Bit. Nach dem Ausführen des Schreiboperationsbefehls setzt der Prozessor die interne Operation für eine Zeit von 4 ms

(optional) aus. Dies ist kein Schlafmodus, da Uhren und Peripheriegeräte weiterhin funktionieren. Nachdem der Schreibzyklus abgeschlossen ist, nimmt der Prozessor den Betrieb ab dem dritten Befehl nach dem EECON1-Schreibbefehl wieder auf. Daten EEPROM-Speicher schreiben

LD	A,ADDR	; Adresse schreiben
LD	EEADR,A	
LD	A, DATEN	; Daten schreiben
LD	EEDAT,A	
LD	A, DATAH	
LD	EEDAT, A	
SETB	EECON1, EEPGD ;	Betrieb des EEPROM zulassen; Schreibsignal aktivieren;
SETB	EECON1, Zaunkönig	Interrupt deaktivieren; Bestätigen,
CLRB	INTCON, GIE	dass Interrupt deaktiviert ist
SZB	INTCON, GIE	
JP	\$_2	
LDIA	055H	; Schreiben Sie 55H und 0AAH in das EECON2-Register
LD	EECON2,A	
LDIA	0AAH	
LD	EECON2,A	
SETB	EECON1, WR	;Schreiben des Programmspeichers
NOP		starten; Schreibpuffer benötigt Verzögerung
NOP		
CLRB	EECON1, Zaunkönig	
SETB	INTCON, GIE	;Interrupt aktivieren;
SZB	EECON1, WR	beurteilen, ob der Schreibvorgang abgeschlossen ist, das WREN-Bit muss während des Schreibvorgangs 1 bleiben
JP	\$_1	
CLRB	EECON1, Zaunkönig	;Schreibende, Write-Enable-Bit ausschalten

16.5 Hinweise zum EEPROM- Betrieb

16.5.1 Schreibbestätigung

Abhängig von der Anwendung schreibt eine gute Programmierpraxis im Allgemeinen vor, dass der in das Daten-EEPROM geschriebene Wert mit dem erwarteten Wert verglichen wird.

16.5.2 Schutz vor versehentlichem Schreiben

In einigen Fällen möchte der Benutzer möglicherweise keine Daten in den Daten-EEPROM-Speicher schreiben. Um ein versehentliches Beschreiben des EEPROMs zu verhindern, sind verschiedene Schutzmechanismen in den Chip eingebettet. Das WREN-Bit wird beim Einschalten gelöscht. Außerdem verhindert ein Power-up-Timer (18 ms Verzögerung) das Schreiben auf das EEPROM.

Die Schreibinitiationssequenz verhindert zusammen mit dem WREN-Bit unbeabsichtigte Schreibvorgänge unter den folgenden

Bedingungen: • Brown-out • Stromstörung • Softwarestörung

17. Operationsverstärker

Der Chip hat einen eingebauten Operationsverstärker.

17.1 Einführung in Operationsverstärker

Der CMS89F52x verfügt über einen eingebauten Operationsverstärker und sein positiver oder negativer Anschluss kann direkt intern geerdet oder durch Softwareeinstellungen über einen Widerstand geerdet werden.

Der Ausgang kann so programmiert werden, dass er durch interne RC-Filtrung mit AN9 verbunden wird oder zur Erkennung direkt mit dem entsprechenden AD-Wandlungskanal von AN7 verbunden wird. Sein Prinzip ist wie folgt:

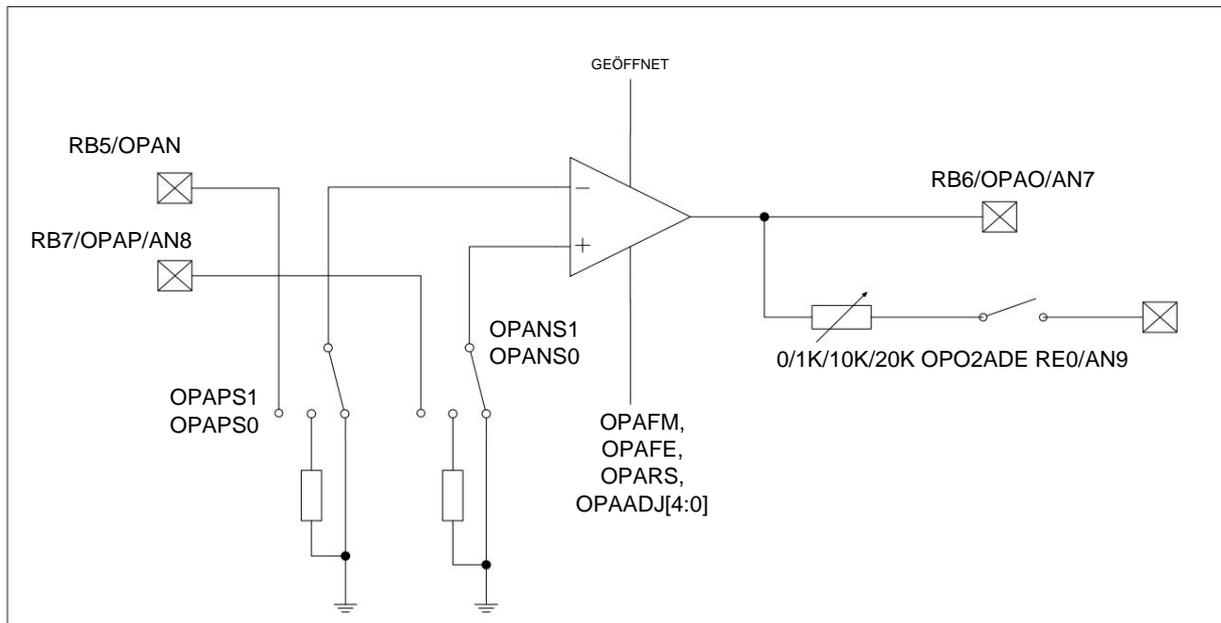


Abbildung 17-1: Funktionsprinzip des Operationsverstärkers

Zugehörige Pin-

Beschreibung Pin-Name	IO-Typ	Pin Beschreibung	Negativer Eingang
OPAN			des Operationsverstärkers
OPAP		Positiver Eingang des Operationsverstärkers	
OPA0		Operationsverstärkerausgang	
AN7		- Der Operationsverstärker kann intern mit dem AD-Kanal dieses Ports verbunden werden. Der Operationsverstärker kann	
AN9		- intern mit dem AD-Kanal dieses Ports verbunden werden und das externe Kondensatorfilter kann mit diesem Port verbunden werden	

17.2 Register im Zusammenhang mit Operationsverstärkern

Es gibt 3 Register, die sich auf den Operationsverstärker beziehen, nämlich OPAON, OPAON1 und OPAADJ.

Operationsverstärker-Steuerregister OPAON(108H)

108H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPAON OPAEN OPAFM OPAFE				----	OPAPS1 OPAPS0	OPANS1	OPANS0	
R/W	R/W	R/W	R/W	----	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	1	----	0	0	0	0

Bit7

OPAEN: Operationsverstärker-Aktivierungsbit;
0: Operationsverstärker aus; 1:
Operationsverstärker aktiviert.

Bit6

OPAFM: Betriebsverstärker-Anpassungsmodus
aktiviert; 0: Normalmodus; 1:
Anpassungsmodus.

Bit5

OPAFE: OPA-Ausgangsfilter aktivieren; 0:
deaktiviert; 1: aktiviert. Behinderte

Bit4**Bit3~Bit2**

OPAPS1-OPAPS0: Auswahlbits für positive Eingänge des Operationsverstärkers;
00: Mit GND verbinden;
01: Mit 1K-Pulldown-Widerstand
verbinden; 1x: Mit OPP-Port verbinden (wenn OPAEN=1, OPP-Port als positiver Eingang des Operationsverstärkers aktivieren).

Bit1~Bit0

OPANS1-OPANS0: Eingangsauswahlbits für den negativen Anschluss des Operationsverstärkers;
00: Mit GND verbinden;
01: Mit 1K-Pulldown-Widerstand
verbinden; 1x: Mit OPN-Port verbinden (wenn OPAEN=1, OPN-Port-Operationsverstärker-Minuspoleingang aktivieren).

Operationsverstärker-Steuerregister OPAON1(109H)

109H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPAON1	----	----	----	----	OPO2ADE	----	ANRS1 ANRS0	
R/W	----	----	----	----	R/W	----	R/W	R/W
Wert zurücksetzen	----	----	----	----	0	----	0	0

Bit7~Bit4

Behinderte

Bit3

OPO2ADE: Der Operationsverstärker-Ausgangsport OPAO ist mit dem Aktivierungsbit des CAP*-Ports verbunden;
0: deaktivieren;
1: aktivieren.

Bit2

Behinderte

Bit1~Bit0

ANRS1-ANRS0: Operationsverstärker-Ausgangsport OPAO verbunden mit CAP*-Port-Widerstandsauswahlbit (wirksam, wenn OPO2ADE = 1); 00: Operationsverstärker-Ausgang direkt mit CAP-Port verbunden; 01: Operationsverstärker-Ausgang mit 1K-Widerstand mit CAP-Port verbunden; 10: Operationsverstärker-Ausgang Verbinden Sie den 10K-Widerstand mit dem CAP-Port für den Verstärker-Ausgang; 11: Verbinden Sie den 20K-Widerstand für den Operationsverstärker-Ausgang mit dem CAP-Port.



Operationsverstärker-Einstellregister OPAADJ(107H)

107H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPAADJ	OPADOUT	OPARS	----	OPAADJS[4:0]				
R/W	R	R/W	----	R/W	R/W	R/W	R/W	R/W
Wert zurücksetzen	0	0	----	1	0	0	0	0

Bit7

OPADOUT: Ausgabe im Operationsverstärker-Einstellmodus, nur lesbar.

Bit6

OPARS: Operationsverstärker-Einstellmodus Eingangsauswahl; 0:

Negativer Eingang; 1: Positiver Eingang. OPAADJ[4:0]

deaktivieren: Bits zum Anpassen der Offset-Spannung

Bit5

des Operationsverstärkers anpassen.

Bit4~Bit0

18. Elektrische Parameter**18.1 DC- Eigenschaften**

Symbol	Parameter	Test-		Min.	Typische	Max.	Einheit
		VDD	Bedingungen				
VDD	Betriebsspannung	-	8M	3.5	-	5.5	v
		-	4M	3.5	-	5.5	v
IDD	Arbeitsstrom	5V	ADC aktivieren	-	3	-	mA
		3V	ADC aktivieren	-	2	-	mA
ISTB	Ruhestrom	5V	----		0,1		mA
		3V	----		0,1		mA
VIL	Niedrige Eingangsspannung	-	----	-	-	0,3 VDD	v
VIH	Hohe Eingangsspannung	-	----	0,7 VDD	-	-	v
VOH	Hohe Ausgangsspannung	-	ohne Last	0,9 VDD	-	-	v
VOL	Niedrige Ausgangsspannung	-	ohne Last	-	-	0,1 VDD	v
VADI	Eingangsspannung des AD-Anschlusses	-	----	0	-	VDD	v
VAD	Betriebsspannung des AD-Moduls	-	----	2.7	-	5.5	v
VEEPROM EEPROM Modul Betriebsspannung-				3.0	-	5.5	v
EAD	AD-Konvertierungsfehler	-	----	-	±2	-	-
RPH	Pullup-Widerstandswert	5V	----	-	35	-	K
		3V	----	-	65	-	K
IOL	Senkenstrom des Ausgangsanschlusses	5V	VOL = 0,3 VDD	-	60	-	mA
		3V	VOL = 0,3 VDD	-	25	-	mA
IOH	Ausgangsport-Quellenstrom	5V	VOH = 0,7 VDD	-	15	-	mA
		3V	VOH = 0,7 VDD	-	10	-	mA

18.2 AC- Eigenschaften

Symbol	Parameter	Test-		Min.	Typische	Max.	Einheit
		VDD	Bedingungen				
TWDT	WDT-Resetzeit	5V	---	-	18	-	MS
		3V	---	-	36	-	MS
BISCHEN	AD-Konvertierungszeit	5V	---	-	41	-	CLK
		3V	---	-	41	-	CLK
TEEPROM	EEPROM-Schreibzeit	5V	---	-	2.5	-	MS
		3V	---	-	2.5	-	MS

18.3 Eigenschaften des internen RC- Oszillators**18.3.1** Spannungscharakteristik des internen RC - Oszillators

Test-Bedingungen	Oszillationsfrequenz (typisch) (Hz)
2,5 V	8,0 Mio
2,6 V	8,1 Mio
2,8 V	8,2 Mio
3,0 V	8,3 Mio
3,2 V	8,3 Mio
3,4 V	8,2 Mio
3,6 V	8,2 Mio
3,8 V	8,2 Mio
4,0 V	8,1 Mio
4,2 V	8,1 Mio
4,4 V	8,1 Mio
4,6 V	8,0 Mio
4,8 V	8,0 Mio
5,0 V	8,0 Mio
5,2 V	8,0 Mio
5,4 V	7,9 Mio
5,5 V	7,8 Mio

18.3.2 Temperatureigenschaften des internen RC - Oszillators

Prüfbedingung	-20 °C	25 Grad	40y	60y	85 °C
Schwingfrequenz (typischer Wert) (Hz)	7,9 Mio	8,0 Mio	8,0 Mio	8,1 Mio	8,1 Mio

19. Anweisungen**19.1 Befehlsliste**

Gedächtnisstütze	arbeiten	Befehlszyklus	Flag
Kontrollklasse			
NOP	keine Operation	1	keiner
STOPPEN	in den Schlafmodus wechseln	1	AN, PD
CLRWDT	Löschen Sie den Watchdog-Zähler	1	AN, PD
Datenübertragung			
LD	[R], A überträgt ACC-Inhalte an R	1	KEINER
LD	A,[R] R-Inhalt an ACC übertragen	1	Z
TESTZ [R]	Datenspeicherinhalt in Datenspeicher übertragen	1	Z
LDIA	i Sofortiger Wert, den ich an ACC gesendet habe	1	KEINER
logische Operation			
CLRA	ACC löschen	1	Z
SATZ	[R] Datenspeicher R einstellen	1	KEINER
CLR	[R] Datenspeicher löschen R	1	Z
ODER EIN	[R] Führen Sie eine "ODER"-Operation mit R- und ACC-Inhalt durch und speichern Sie das Ergebnis in ACC	1	Z
ORR	[R] Führen Sie eine „oder“-Operation mit R- und ACC-Inhalten durch und speichern Sie das Ergebnis in R	1	Z
ANDA [R]	Führen Sie eine "UND"-Operation mit R- und ACC-Inhalt durch und speichern Sie das Ergebnis in ACC	1	Z
ANDR [R]	Führen Sie eine "AND"-Operation mit R- und ACC-Inhalten durch und speichern Sie das Ergebnis in R	1	Z
XORA [R]	Führen Sie eine "XOR"-Operation mit R- und ACC-Inhalt durch und speichern Sie das Ergebnis in ACC	1	Z
XORR [R]	Führen Sie eine "XOR"-Operation mit R- und ACC-Inhalten durch und speichern Sie das Ergebnis in R	1	Z
SWAPA [R]	Wandeln Sie die hohen und niedrigen Nibbles des Inhalts des R-Registers um und speichern Sie das Ergebnis in ACC	1	KEINER
SWAPR [R]	High- und Low-Nibble-Konvertierung des Inhalts des R-Registers, und das Ergebnis wird in R gespeichert	1	KEINER
Koma [R]	Invertieren Sie den Inhalt des R-Registers und speichern Sie das Ergebnis in ACC	1	Z
KOMR [R]	Invertieren Sie den Inhalt des R-Registers und speichern Sie das Ergebnis in R	1	Z
XORIA i	ACC und die unmittelbare Zahl i führen eine "XOR"-Operation durch, und das Ergebnis wird in ACC gespeichert	1	Z
ANDIA	Führen Sie eine "UND"-Operation mit ACC und dem unmittelbaren Wert i durch und speichern Sie das Ergebnis in ACC	1	Z
ORIA	Führen Sie eine "ODER"-Operation mit ACC und dem unmittelbaren Wert i durch und speichern Sie das Ergebnis in ACC	1	Z
Schichtbetrieb			
RRCA [R]	Der Datenspeicher wird mit einem Übertrag nach rechts rotiert und das Ergebnis in ACC gespeichert	1	C
RRCR [R]	Der Datenspeicher wird mit einem Übertrag um ein Bit nach rechts rotiert und das Ergebnis in R gespeichert	1	C
RLCA	[R] Der Datenspeicher wird mit Übertrag um ein Bit nach links rotiert und das Ergebnis in ACC gespeichert	1	C
RLCR	[R] Der Datenspeicher wird mit Übertrag um ein Bit nach links rotiert und das Ergebnis in R gespeichert	1	C
RLA	[R] Der Datenspeicher rotiert ohne Übertrag um ein Bit nach links, und das Ergebnis wird in ACC gespeichert	1	KEINER
RLR	[R] Der Datenspeicher wird ohne Übertrag um ein Bit nach links rotiert und das Ergebnis in R gespeichert	1	KEINER
RRA	[R] Der Datenspeicher rotiert ohne Übertrag um ein Bit nach rechts, und das Ergebnis wird in ACC gespeichert	1	KEINER
RRR	[R] Der Datenspeicher rotiert ohne Übertrag um ein Bit nach rechts, und das Ergebnis wird in R gespeichert	1	KEINER
Inkrement Dekrement			
INCA	[R] Datenspeicher R inkrementieren, Ergebnis in ACC stellen	1	Z
ERHÖHUNG	[R] Datenspeicher R inkrementieren, Ergebnis in R setzen	1	Z
DECA [R]	Dekrementiere den Datenspeicher R und lege das Ergebnis in ACC	1	Z

Gedächtnisstütze	arbeiten	Befehlszyklus	Flag
DECR [R]	Datenspeicher R dekrementieren, Ergebnis in R-Bit-	1	Z
Operation setzen			
CLRB [R],b	Lösche ein Bit im Datenspeicher R	1	KEINER
SETB [R],b	Setzen Sie ein Bit im Datenspeicher R auf 1	1	KEINER
Tabelle überprüfen			
TISCH [R]	Lesen Sie den FLASH-Inhalt und fügen Sie das Ergebnis in TABLE_DATAH und R ein	2	KEINER
TABLEA	Lesen Sie den FLASH-Inhalt und fügen Sie das Ergebnis in TABLE_DATAH und ACC ein	2	KEINER
Berechnung			
ADDA[R]	$ACC+[R] \dot{y} ACC$	1	C,DC,Z,OV
ADR[R]	$ACC+[R] \dot{y} R$	1	C,DC,Z,OV
ADDCA [R]	$ACC+[R]+C \dot{y} ACC$	1	Z,C,DC,OV
ADDCR [R]	$ACC+[R]+C \dot{y} R$	1	Z,C,DC,OV
ADDIA	$ACC+i \dot{y} ACC$	1	Z,C,DC,OV
SUBA [R]	$[R]-ACC \dot{y} ACC$	1	C,DC,Z,OV
SUBR [R]	$[R]-ACC \dot{y} R$	1	C,DC,Z,OV
SUBCA[R]	$[R]-ACC-C \dot{y} ACC$	1	Z,C,DC,OV
SUBCR [R]	$[R]-ACC-C \dot{y} R$	1	Z,C,DC,OV
SUBIA	$i-ACC \dot{y} ACC$	1	Z,C,DC,OV
HSUBA[R]	$ACC-[R] \dot{y} ACC$	1	Z,C,DC,OV
HSUBR[R]	$ACC-[R] \dot{y} R$	1	Z,C,DC,OV
HSUBCA[R]	$ACC-[R]-C \dot{y} ACC$	1	Z,C,DC,OV
HSUBCR[R]	$ACC-[R]-C \dot{y} R$	1	Z,C,DC,OV
HSUBIA i	$ACC-i \dot{y} ACC$	1	Z,C,DC,OV
bedingungslose Übergabe			
RET	Rückkehr vom Unterprogramm	2	KEINER
RET i	kehrt von der Unteroutine zurück und speichert den unmittelbaren Wert I in ACC	2	KEINER
RETI	Rückkehr vom Interrupt	2	KEINER
FORDERUNG	ADD-Unterprogrammaufruf	2	KEINER
JP	Bedingungslos Sprung HINZUFÜGEN	2	KEINER
Bedingte Verzweigung			
SZB [R],b	Wenn das Bit b des Datenspeichers R "0" ist, überspringe die nächste Anweisung	1 oder 2	KEINER
SNZB [R],b	Wenn das Bit b des Datenspeichers R "1" ist, überspringe die nächste Anweisung	1 oder 2	KEINER
SZA [R]	Datenspeicher R wird an ACC gesendet, wenn der Inhalt "0" ist, überspringe den nächsten Befehl	1 oder 2	KEINER
ZR [R]	Der Inhalt des Datenspeichers R ist "0", dann den nächsten Befehl überspringen	1 oder 2	KEINER
SZINCA [R]	Addiere "1" zum Datenspeicher R, lege das Ergebnis in ACC, wenn das Ergebnis "0" ist, überspringe die nächste Anweisung	1 oder 2	KEINER
SZINCR [R]	Addiere "1" zum Datenspeicher R, lege das Ergebnis in R, wenn das Ergebnis "0" ist, überspringe die nächste Anweisung	1 oder 2	KEINER
SZDECA [R]	Subtrahiere "1" vom Datenspeicher R, lege das Ergebnis in ACC, wenn das Ergebnis "0" ist, überspringe die nächste Anweisung	1 oder 2	KEINER
SZDECR [R]	Subtrahiere "1" vom Datenspeicher R, lege das Ergebnis in R, wenn das Ergebnis "0" ist, überspringe die nächste Anweisung	1 oder 2	KEINER

19.2 Befehlsbeschreibung**FÜGE HINZU EIN [R]**

arbeiten: Addiere R zu ACC und trage das Ergebnis in ACC ein

Zyklus: 1

Betroffene Flags: C, DC, Z, OV

Beispiel:

LDIA	09H	;Zuweisung von 09H zu
LD	R01,A	ACC ;Zuweisung des Werts von ACC (09H) zu Kundenregister R01
LDIA	077H	;Ordnen Sie 77H dem ACC zu
FÜGE HINZU EIN	R01	;Ausführungsergebnis: ACC=09H + 77H =80H

ADDR [R]

arbeiten: Addiere ACC zu R und trage das Ergebnis in R ein

Zyklus: 1

Betroffene Flags: C, DC, Z, OV

Beispiel:

LDIA	09H	;Zuweisung von 09H zu
LD	R01,A	ACC ;Zuweisung von ACC-Wert (09H) zu benutzerdefiniertem
LDIA	077H	Register R01 ;Zuweisung von 77H zu ACC ;Ausführungsergebnis:
ADDR	R01	R01=09H + 77H =80H

ADDCA [R]

arbeiten: Addiere R plus ACC plus C und trage das Ergebnis in ACC ein

Zyklus: 1

Betroffene Flags: C, DC, Z, OV

Beispiel:

LDIA	09H	;Zuweisung von 09H zu
LD	R01,A	ACC ;Zuweisung von ACC-Wert (09H) zu benutzerdefiniertem
LDIA	077H	Register R01 ;Zuweisung von 77H zu ACC ;Ausführungsergebnis:
ADDCA	R01	ACC= 09H + 77H + C=80H (C=0) ACC= 09H + 77H + C=81H (C=1)

ADDCR [R]

arbeiten: R plus ACC plus C-Bit addieren und das Ergebnis in R eingeben

Zyklus: 1

Betroffene Flags: C, DC, Z, OV

Beispiel:

LDIA	09H	;Zuweisung von 09H zu
LD	R01,A	ACC ;Zuweisung von ACC-Wert (09H) zu benutzerdefiniertem
LDIA	077H	Register R01 ;Zuweisung von 77H zu ACC ;Ausführungsergebnis:
ADDCR	R01	R01 = 09H + 77H + C=80H (C=0) R01 = 09H + 77H + C=81H (C=1)

ADDIA

-

Betrieb: Addieren Sie den unmittelbaren Wert i zu ACC und geben Sie das Ergebnis in ACC ein

Zeitraum: 1

Betroffene Flags: C, DC, Z, OV

Beispiel:

LDIA	09H	;09H ACC
ADDIA	077H	zuweisen ;Ausführungsergebnis: ACC = ACC(09H) + i(77H)=80H

UND EIN

[R]

arbeiten: Register R und ACC führen eine logische UND-Operation durch und geben das Ergebnis in ACC ein

Zyklus: 1

Betroffene Flaggen: Z

Beispiel:

LDIA	0FH	;0FH an ACC
LD	R01,A	zuweisen ;ACC-Wert (0FH) an Register R01
LDIA	77H	zuweisen ;77H an ACC
UND EIN	R01	zuweisen ;Ausführungsergebnis: ACC=(0FH und 77H)=07H

ANDR

[R]

arbeiten: Register R und ACC führen eine logische UND-Operation durch, und das Ergebnis wird in R abgelegt

Zyklus: 1

Betroffenes Flag: Z Beispiel:

LDIA	0FH	;0FH an ACC
LD	R01,A	zuweisen ;ACC-Wert (0FH) an Register R01
LDIA	77H	zuweisen ;77H an ACC
ANDR	R01	zuweisen ;Ausführungsergebnis: R01=(0FH und 77H)=07H

ANDIA

-

arbeiten: Führen Sie eine logische UND-Operation mit der unmittelbaren Zahl i und ACC durch und geben Sie das Ergebnis in ACC ein

Zyklus: 1

Betroffene Flaggen: Z

Beispiel:

LDIA	0FH	;0FH an ACC
ANDIA	77H	zuweisen ;Ausführungsergebnis: ACC =(0FH und 77H)=07H

FORDERUNG

hinzufügen

arbeiten: Unterprogramm aufrufen

Zyklus: 2

Betroffene Flags: Keine

Beispiel:

FORDERUNG	SCHLEIFE	;Unterprogrammadresse aufrufen, deren Name als "LOOP" definiert ist
-----------	----------	---

CLRA

Betrieb: ACC-Reset

Zeitraum: 1

Betroffene Flaggen: Z

Beispiel:

CLRA ;Ausführungsergebnis: ACC=0

CLR**[R]**

arbeiten: Register R wird gelöscht

Zyklus: 1

Betroffenes Flag: Z Beispiel:

CLR R01 ;Ausführungsergebnis: R01=0

CLRB**[R], Bit**

arbeiten: b des b-Registers R wird gelöscht

Zyklus: 1

Betroffene Flags: Keine

Beispiel:

CLRB R01,3 ;Ausführungsergebnis: Das dritte Bit von R01 ist Null

CLRWDT

Betrieb: Löschen Sie den Watchdog-Zähler

Zeitraum: 1

Betroffene Flags: TO, PD

Beispiel:

CLRWDT ; Watchdog-Zähler löschen

KOMA**[R]**

arbeiten: Register R wird umgekehrt, und das Ergebnis wird in ACC eingegeben

Zyklus: 1

Betroffene Flaggen: Z

Beispiel:

```
LDIA    0AH           ;Zuweisung von 0AH zu
LD      R01,A        ACC ;Zuweisung des Werts von ACC (0AH) zu Register R01
KOMA   R01           ;Ausführungsergebnis: ACC=0F5H
```

KOMR [R]

Betrieb: Das Register R wird invertiert und das Ergebnis in R abgelegt

Zeitraum: 1

Betroffene Flaggen: Z

Beispiel:

LDIA	0AH	;0AH an ACC
LD	R01,A	zuweisen ;Wert von ACC (0AH) an Register R01
KOMR	R01	zuweisen ;Ausführungsergebnis: R01=0F5H

DECA [R]

arbeiten: Register R wird um 1 dekrementiert und das Ergebnis in ACC abgelegt

Zyklus: 1

Betroffene Flaggen: Z

Beispiel:

LDIA	0AH	;0AH an ACC
LD	R01,A	zuweisen ;Wert von ACC (0AH) an Register R01
DECA	R01	zuweisen ;Ausführungsergebnis: ACC=(0AH-1)=09H

DEKR [R]

arbeiten: Das Register R wird um 1 dekrementiert und das Ergebnis in R abgelegt

Zyklus: 1

Betroffenes Flag: Z Beispiel:

LDIA	0AH	;0AH an ACC
LD	R01,A	zuweisen ;Wert von ACC (0AH) an Register R01
DEKR	R01	zuweisen ;Ausführungsergebnis: R01=(0AH-1)=09H

HSUBA [R]

arbeiten: Subtrahieren Sie R von ACC, geben Sie das Ergebnis in ACC ein

Zyklus: 1

Betroffene Flags: C,DC,Z,OV

Beispiel:

LDIA	077H	;ACC-Zuweisung 077H
LD	R01,A	; Weisen Sie Register R01 den Wert von ACC (077H) zu
LDIA	080H	;ACC-Zuweisung
HSUBA	R01	080H ;Ausführungsergebnis: ACC=(80H-77H)=09H

HSUBR**[R]**

Subtrahiere R von ACC und setze das Ergebnis in R ein

1

Operation: Zyklus: Betroffene Flags:

C,DC,Z,OV Beispiel:

LDIA	077H	;077H ACC
LD	R01,A	zuweisen ;Wert von ACC (077H) Register R01 zuweisen
LDIA	080H	;ACC-Zuweisung
HSUBR	R01	080H ;Ausführungsergebnis: R01=(80H-77H)=09H

HSUBCA**[R]**

arbeiten: ACC minus R minus C, trage das Ergebnis in ACC ein

Zyklus: 1

Betroffene Flags: C,DC,Z,OV

Beispiel:

LDIA	077H	;077H ACC
LD	R01,A	zuweisen ;Wert von ACC (077H) Register R01 zuweisen
LDIA	080H	;ACC-Zuweisung
HSUBCA R01		080H ;Ausführungsergebnis: ACC=(80H-77H-C)=09H(C=0) ACC=(80H-77H-C)=08H(C=1)

HSUBCR**[R]**

arbeiten: ACC minus R minus C, trage das Ergebnis in R ein

Zyklus: 1

Betroffene Flags: C,DC,Z,OV

Beispiel:

LDIA	077H	;077H ACC
LD	R01,A	zuweisen ;Wert von ACC (077H) Register R01 zuweisen
LDIA	080H	;ACC-Zuweisung
HSUBC R	R01	080H ;Ausführungsergebnis: R01=(80H-77H-C)=09H(C=0) R01=(80H-77H-C)=08H(C=1)

INCA**[R]**

Register R addiert 1, und das Ergebnis wird in ACC eingegeben

1

Operation: Zeitraum:

Betroffene Flags: Z Beispiel:

LDIA	0AH	;0AH an ACC
LD	R01,A	zuweisen ;Wert von ACC (0AH) an Register R01
INCA	R01	zuweisen ;Ausführungsergebnis: ACC=(0AH+1)=0BH

ERHÖHUNG**[R]**

Das Register R wird um 1 erhöht und das Ergebnis in R abgelegt

1

Operation: Zeitraum:

Betroffene Flags: Z Beispiel:

LDIA	0AH	;Zuweisung von 0AH
LD	R01,A	zu ACC ;Zuweisung von ACC-Wert (0AH) zu Register
ERHÖHUNG	R01	R01 ;Ausführungsergebnis: R01=(0AH+1)=0BH

JP

hinzufügen

Zum Hinzufügen einer Adresse springen

2

Operation: Zeitraum: Betroffene

Flags: Keine Beispiel:

JP	SCHLEIFE	;Zur Unterprogrammadresse springen, deren Name als "LOOP" definiert ist
----	----------	---

LD**A,[R]**

Weist ACC den Wert von R zu

1

Operation: Zeitraum:

Betroffene Flags: Z Beispiel:

LD	A, R01	;Ordnen Sie den Wert von Register R0 ACC
LD	R02,A	zu. ;Ordnen Sie den Wert von ACC Register R02 zu, wodurch die Übertragung von Daten von R01 zu R02 realisiert wird

LD**[R], A**

weist R den Wert von ACC zu

1

Operation: Zeitraum: Betroffene

Flags: Keine Beispiel:

LDIA	09H	;09H ACC
LD	R01,A	zuweisen ;Ausführungsergebnis: R01=09H

LDIA

-

Der unmittelbare Wert i wird ACC zugewiesen

1

Operation: Zeitraum: Betroffene

Flags: Keine Beispiel:

LDIA	0AH	;ACC-Zuweisung 0AH
------	-----	--------------------

NOP

leere Anweisung

1

Operation: Zeitraum: Betroffene

Flags: Keine Beispiel:

NOP

NOP

ORIA

-

Die logische OR-Operation wird mit dem unmittelbaren Wert und ACC durchgeführt, und das Ergebnis wird ACC zugewiesen

1

Operation: Zeitraum:

Betroffene Flags: Z Beispiel:

LDIA	0AH	;ACC-Zuweisung
ORIA	030H	0AH ;Ausführungsergebnis: ACC =(0AH oder 30H)=3AH

ODER EIN

[R]

Register R führt eine logische ODER-Operation mit ACC durch, und das Ergebnis wird in ACC eingegeben

1

Operation: Zeitraum:

Betroffene Flags: Z Beispiel:

LDIA	0AH	;0AH zu ACC
LD	R01,A	zuweisen ;ACC(0AH) zu Register R01
LDIA	30H	zuweisen ;30H zu ACC
ODER EIN	R01	zuweisen ;Ausführungsergebnis: ACC=(0AH oder 30H)=3AH

ORR

[R]

Register R führt eine logische ODER-Operation mit ACC durch, und das Ergebnis wird in R abgelegt

1

Operation: Zeitraum:

Betroffene Flags: Z Beispiel:

LDIA	0AH	;Zuordnen von 0AH zu
LD	R01,A	ACC ;Zuordnen von ACC(0AH) zu Register
LDIA	30H	R01 ;Zuordnen von 30H zu
ORR	R01	ACC ;Ausführungsergebnis: R01=(0AH oder 30H)=3AH

RET

Rückkehr vom Unterprogramm

2

Operation: Zeitraum: Betroffene

Flags: Keine Beispiel:

FORDERUNG	SCHLEIFE	;Unterprogramm LOOP aufrufen
NOP		; Diese Anweisung wird ausgeführt, nachdem die RET-Anweisung zurückgegeben
...		wird; andere Programme
SCHLEIFE:		
...		
RET		;Unterprogramm;Unterprogrammrücksprung

RET

-

Kehre von der Unterroutine mit Parametern zurück und setze die Parameter in ACC

2

Operation: Zeitraum: Betroffene

Flags: Keine Beispiel:

FORDERUNG	SCHLEIFE	;Unterprogramm LOOP aufrufen
NOP		; Diese Anweisung wird ausgeführt, nachdem die RET-Anweisung zurückgegeben
...		wird; andere Programme
SCHLEIFE:		
...		
RET	35H	;Unterprogramm; Unterprogrammrücksprung, ACC=35H

RETI

Rückkehr unterbrechen

2

Operation: Zeitraum: Betroffene

Flags: Keine Beispiel:

INT_START		;Eintritt in die Interrupt-
...		Routine;Interrupt-
RETI		Handler;Interrupt-Rückkehr

RLCA**[R]**

arbeiten: Register R mit C rotiert um ein Bit nach links und das Ergebnis wird in ACC abgelegt

Zyklus: 1

Betroffene Flaggen: C

Beispiel:

LDIA	03H	;ACC-Zuweisung 03H
LD	R01,A	;ACC-Wert R01
RLCA	R01	zuweisen,R01=03H ;Operationsergebnis: ACC=06H(C=0); ACC=07H(C=1) C=0

RLCR [R]

Betrieb: Register R mit C dreht ein Bit nach links, und das Ergebnis wird in R abgelegt

Zeitraum: 1

Betroffene Flaggen: C

Beispiel:

```
LDIA      03H          ;ACC-Zuordnung 03H
LD        R01,A       ;ACC-Wert R01
RLCR      R01         zuweisen,R01=03H ;Operationsergebnis: R01=06H(C=0);
                                   R01=07H(C=1);
                                   C=0
```

RLA [R]

arbeiten: Register R ohne C rotiert um ein Bit nach links, und das Ergebnis wird in ACC abgelegt

Zyklus: 1

Betroffene Flags: Keine

Beispiel:

```
LDIA      03H          ;ACC-Zuweisung 03H
LD        R01,A       ;ACC-Wert R01
RLA       R01         zuweisen,R01=03H ;Operationsergebnis: ACC=06H
```

RLR [R]

arbeiten: Register R ohne C dreht ein Bit nach links und legt das Ergebnis in R ab

Zyklus: 1

Betroffene Flags: Keine

Beispiel:

```
LDIA      03H          ;ACC-Zuordnung 03H
LD        R01,A       ;ACC-Wert R01 zuweisen,
RLR       R01         R01=03H ;Operationsergebnis: R01=06H
```

RRCA [R]

arbeiten: Register R mit C rotiert um ein Bit nach rechts und das Ergebnis wird in ACC abgelegt

Zyklus: 1

Betroffene Flaggen: C

Beispiel:

```
LDIA      03H          ;ACC-Zuordnung 03H
LD        R01,A       ;ACC-Wert R01
RRCA     R01         zuweisen,R01=03H ;Operationsergebnis: ACC=01H(C=0);
                                   ACC=081H(C=1);
                                   C=1
```

RRCR**[R]**

Betrieb: Register R mit C rotiert um ein Bit nach rechts, und das Ergebnis wird in R abgelegt

Zeitraum: 1

Betroffene Flaggen: C

Beispiel:

```
LDIA      03H          ;ACC-Zuweisung 03H
LD        R01,A       ;ACC-Wert R01
RRCR      R01         zuordnen,R01=03H ;Operationsergebnis: R01=01H(C=0);
                                   R01=81H(C=1);
                                   C=1
```

RRA**[R]**

arbeiten: Register R ohne C rotiert um ein Bit nach rechts und legt das Ergebnis in ACC ab

Zyklus: 1

Betroffene Flags: Keine

Beispiel:

```
LDIA      03H          ;ACC-Zuweisung 03H
LD        R01,A       ;ACC-Wert R01 zuweisen,
RRA       R01         R01=03H ;Operationsergebnis: ACC=81H
```

RRR**[R]**

arbeiten: Register R ohne C dreht ein Bit nach rechts und legt das Ergebnis in R ab

Zyklus: 1

Betroffene Flags: Keine

Beispiel:

```
LDIA      03H          ;ACC-Zuweisung 03H
LD        R01,A       ;ACC-Wert R01 zuweisen,
RRR       R01         R01=03H ;Operationsergebnis: R01=81H
```

SATZ**[R]**

arbeiten: Register R alle Bits 1

Zyklus: 1

Betroffene Flags: Keine

Beispiel:

```
SATZ      R01         ;Operationsergebnis: R01=0FFH
```

SETB**[R], die**

arbeiten: b-Position des b-Registers R ist 1

Zyklus: 1

Betroffene Flags: Keine

Beispiel:

```
CLR       R01
SETB      R01,3       ;R01=0 ;Operationsergebnis: R01=08H
```

STOPPEN

Betrieb: Geh schlafen

Zeitraum: 1

Betroffene Flags: TO, PD

Beispiel:

STOPPEN

;Der Chip wechselt in den Energiesparmodus, die CPU und der Oszillator funktionieren nicht mehr und der IO-Port bleibt im ursprünglichen Zustand

SUBIA

-

arbeiten: Subtrahieren Sie ACC vom unmittelbaren Wert I und geben Sie das Ergebnis in ACC ein

Zyklus: 1

Betroffene Flags: C,DC,Z,OV Beispiel:

LDIA 077H

;ACC-Zuordnung 77H

SUBIA 80H

;Operationsergebnis: ACC=80H-77H=09H

SUBA

[R]

Register R minus ACC, das Ergebnis wird in ACC abgelegt

1

Operation: Zyklus: Betroffene Flags:

C,DC,Z,OV Beispiel:

LDIA 080H

;ACC-Zuordnung 80H

LD R01,A

;R01 den Wert von ACC zuweisen, R01=80H

LDIA 77H

;ACC-Zuweisung

SUBA R01

77H ;Operationsergebnis: ACC=80H-77H=09H

SUBR

[R]

arbeiten: Register R minus ACC, das Ergebnis wird in R abgelegt

Zyklus: 1

Betroffene Flags: C,DC,Z,OV

Beispiel:

LDIA 080H

;ACC-Zuordnung 80H

LD R01,A

;R01 den Wert von ACC zuweisen, R01=80H

LDIA 77H

;ACC-Zuweisung 77H

SUBR R01

;Operationsergebnis: R01=80H-77H=09H

SUBCA

[R]

Register R minus ACC minus C, das Ergebnis wird in ACC abgelegt

1

Operation: Zyklus: Betroffene Flags:

C,DC,Z,OV Beispiel:

LDIA	080H	;ACC-Zuweisung 80H
LD	R01,A	;R01 den Wert von ACC zuweisen, R01=80H
LDIA	77H	;ACC-Zuweisung
SUBCA	R01	77H ;Operationsergebnis: ACC=80H-77H-C=09H(C=0); ACC=80H-77H-C=08H(C=1);

SUBCR

[R]

Register R minus ACC minus C, das Ergebnis wird in R abgelegt

1

Operation: Zyklus: Betroffene Flags:

C,DC,Z,OV Beispiel:

LDIA	080H	;ACC-Zuordnung 80H
LD	R01,A	;R01 den Wert von ACC zuweisen, R01=80H
LDIA	77H	;ACC-Zuordnung
SUBCR	R01	77H ;Operationsergebnis: R01=80H-77H-C=09H(C=0) R01=80H-77H-C=08H(C=1)

SWAPA

[R]

arbeiten: Die High- und Low-Nibbles des Registers R werden ausgetauscht und das Ergebnis in ACC abgelegt

Zyklus: 1

Betroffene Flags: Keine

Beispiel:

LDIA	035H	;ACC-Zuordnung 35H
LD	R01,A	;R01 den Wert von ACC zuweisen,
SWAPA	R01	R01=35H ;Operationsergebnis: ACC=53H

SWAPR

[R]

arbeiten: Register R High und Low Nibble Austausch, das Ergebnis wird in R abgelegt

Zyklus: 1

Betroffene Flags: Keine

Beispiel:

LDIA	035H	;ACC-Zuweisung 35H
LD	R01,A	;R01 den Wert von ACC zuweisen,
SWAPR R01		R01=35H ;Operationsergebnis: R01=53H

SZB**[R],b**

Betrieb: beurteilt das b-te Bit des Registers R, springt auf 0, wird andernfalls sequentiell ausgeführt

Zeitraum: 1 oder 2

Betroffene Flags: Keine

Beispiel:

SZB	R01,3	;Das dritte Bit des Registers R01 beurteilen
JP	SCHLEIFE	;Das dritte Bit von R01 ist 1, um diese Anweisung auszuführen, zu LOOP springen
JP	SCHLEIFE1	;Das dritte Bit von R01 ist 0 Zeitsprung, diese Anweisung ausführen, zu LOOP1 springen

SNZB**[R],b**

arbeiten: beurteilt das b-te Bit des Registers R, es ist 1 Sprung, sonst wird sequentiell ausgeführt

Zyklus: 1 oder 2

Betroffene Flags: Keine

Beispiel:

SNZB	R01,3	;Das dritte Bit des Registers R01 beurteilen
JP	SCHLEIFE	;Das dritte Bit von R01 ist 0, um diese Anweisung auszuführen, zu LOOP springen
JP	SCHLEIFE1	;Das dritte Bit von R01 ist ein 1-facher Sprung, führen Sie diese Anweisung aus und springen Sie zu LOOP1

SZA**[R]**

arbeiten: Weist ACC den Wert von Register R zu, überspringt, wenn R 0 ist, andernfalls sequenziell ausführen

Zyklus: 1 oder 2

Betroffene Flags: Keine

Beispiel:

SZA	R01	;R01ýACC
JP	SCHLEIFE	;Diese Anweisung ausführen, wenn R01 nicht 0 ist, zu LOOP springen
JP	SCHLEIFE1	;R01 ist 0 Zeitsprung, diese Anweisung ausführen, zu LOOP1 springen

ZR**[R]**

arbeiten: Weist R den Wert von Register R zu, wenn R 0 ist, überspringen, andernfalls sequentiell ausführen

Zyklus: 1 oder 2

Betroffene Flags: Keine

Beispiel:

ZR	R01	;R01ýR01
JP	SCHLEIFE	;Diese Anweisung ausführen, wenn R01 nicht 0 ist, zu LOOP springen
JP	SCHLEIFE1	;R01 ist 0 Zeitsprung, um diese Anweisung auszuführen, springe zu LOOP1

Szinca**[R]**

Betrieb: Inkrementiere das Register R um 1, setze das Ergebnis in ACC, wenn das Ergebnis 0 ist, überspringe die nächste Anweisung, andernfalls sequenziell ausführen

Zeitraum: 1 oder 2

Betroffene Flags: Keine

Beispiel:

Szinca	R01	;R01+1ÿACC
JP	SCHLEIFE	;Diese Anweisung ausführen, wenn ACC nicht 0 ist, zu LOOP springen
JP	SCHLEIFE1	;Diese Anweisung ausführen, wenn ACC 0 ist, zu LOOP1 springen

SZINCR**[R]**

arbeiten: Register R um 1 erhöhen, Ergebnis in R schreiben, wenn das Ergebnis 0 ist, nächste Anweisung überspringen, andernfalls sequenziell ausführen

Zyklus: 1 oder 2

Betroffene Flags: Keine

Beispiel:

SZINCR	R01	;R01+1ÿR01
JP	SCHLEIFE	; Führen Sie diese Anweisung aus, wenn R01 nicht 0 ist, springen Sie zu LOOP
JP	SCHLEIFE1	; Führen Sie diese Anweisung aus, wenn R01 0 ist, springen Sie zu LOOP1

SZDECA**[R]**

arbeiten: Dekrementiere das Register R um 1, setze das Ergebnis in ACC, wenn das Ergebnis 0 ist, überspringe die nächste Anweisung, andernfalls sequenziell ausführen

Zyklus: 1 oder 2

Betroffene Flags: Keine

Beispiel:

SZDECA R01		;R01-1ÿACC
JP	SCHLEIFE	;Diese Anweisung ausführen, wenn ACC nicht 0 ist, zu LOOP springen
JP	SCHLEIFE1	;Diese Anweisung ausführen, wenn ACC 0 ist, zu LOOP1 springen

SZDECR**[R]**

arbeiten: Dekrementiere das Register R um 1, setze das Ergebnis in R, wenn das Ergebnis 0 ist, überspringe die nächste Anweisung, andernfalls sequenziell ausführen

Zyklus: 1 oder 2

Betroffene Flags: Keine

Beispiel:

SZDECR R01		;R01-1ÿR01
JP	SCHLEIFE	; Führen Sie diese Anweisung aus, wenn R01 nicht 0 ist, springen Sie zu LOOP
JP	SCHLEIFE1	; Führen Sie diese Anweisung aus, wenn R01 0 ist, springen Sie zu LOOP1

TISCH

[R] Die

Betrieb: Tabelle nachschlagen, die unteren 8 Bits des Nachschlageergebnisses werden in R abgelegt und die höheren Bits werden in das Spezialregister TABLE_SPH abgelegt

Zeitraum: 2

Betroffene Flags: Keine

Beispiel:

```

LDIA      01H          ;ACC-Zuweisung 01H
LD        TABLE_SPH,A ;Der ACC-Wert wird der hohen Adresse der Tabelle zugewiesen, TABLE_SPH=1
LDIA      015H        ;ACC-Zuweisung 15H
LD        TABLE_SPL,A ;ACC-Wert der Tabellenstatusadresse zuweisen,
TISCH     R01         TABLE_SPL=15H ;Adresse der Tabelle 0115H nachschlagen, Ergebnis der Operation: TABLE_DATAH=12H, R01=34H
...
ORG       0115H
DW        1234H

```

TABLEA

arbeiten: Schlagen Sie in der Tabelle nach, die unteren 8 Bits des Nachschlageergebnisses werden in ACC abgelegt, und die höheren Bits werden in das Spezialregister TABLE_SPH abgelegt

Zyklus: 2

Betroffene Flags: Keine

Beispiel:

```

LDIA      01H          ;ACC-Zuordnung 01H
LD        TABLE_SPH,A ;Der ACC-Wert wird der hohen Adresse der Tabelle zugewiesen, TABLE_SPH=1
LDIA      015H        ;ACC-Zuweisung 15H
LD        TABLE_SPL,A ;Ordnen Sie den ACC-Wert der Tabellenstatusadresse zu,
TABLEA    TABLE_SPL=15H ;Schauen Sie die Adresse der Tabelle 0115H nach, das Ergebnis der Operation: TABLE_DATAH=12H, ACC=34H
...
ORG       0115H
DW        1234H

```

TESTZ

[R]

arbeiten: Weisen Sie R den Wert von R zu, um das Z-Flag zu beeinflussen

Zyklus: 1

Betroffenes Flag: Z Beispiel:

```

TESTZ     R0          ; Weisen Sie R0 den Wert von Register R0 zu, um das Z-Flag zu
SZB       STATUS,Z   beeinflussen; beurteilen Sie das Z-Flag, springen Sie auf 0; springen Sie
JP        Hinzufügen1 zu Adresse Add1, wenn Register R0 0 ist
JP        Hinzufügen2 ;Zur Adresse Add2 springen, wenn Register R0 nicht 0 ist

```

XORIA

-

Betrieb: Sofortwert und ACC führen eine logische XOR-Operation durch, und das Ergebnis wird in ACC eingegeben

Zeitraum: 1

Betroffene Flaggen: Z

Beispiel:

```
LDIA      0AH          ;ACC-Zuweisung 0AH
XORIA    0FH          ;Ausführungsergebnis: ACC=05H
```

XORA

[R]

arbeiten: Register R und ACC führen eine logische XOR-Operation durch, und das Ergebnis wird in ACC eingegeben

Zyklus: 1

Betroffene Flaggen: Z

Beispiel:

```
LDIA      0AH          ;ACC-Zuweisung 0AH
LD        R01,A        ;R01 ACC-Wert zuweisen, R01=0AH
LDIA      0FH          ;ACC-Zuweisung
XORA     R01           0FH ;Ausführungsergebnis: ACC=05H
```

XORR

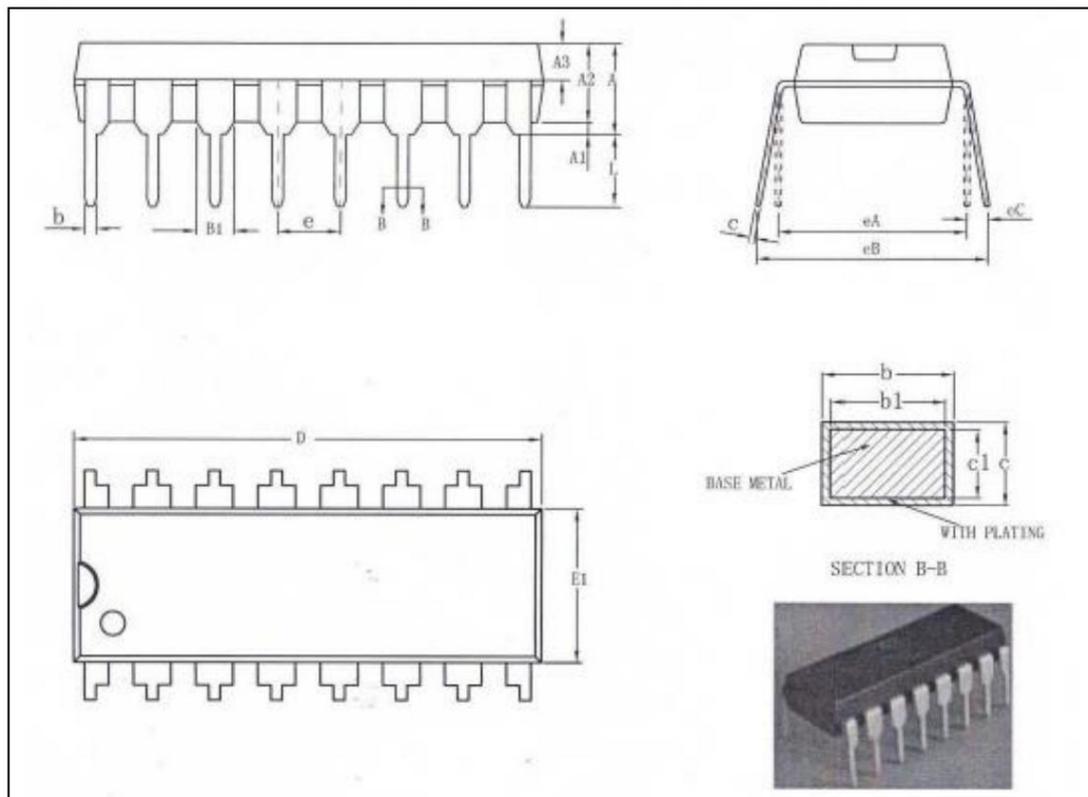
[R]

arbeiten: Register R und ACC führen eine logische XOR-Operation durch, und das Ergebnis wird in R abgelegt

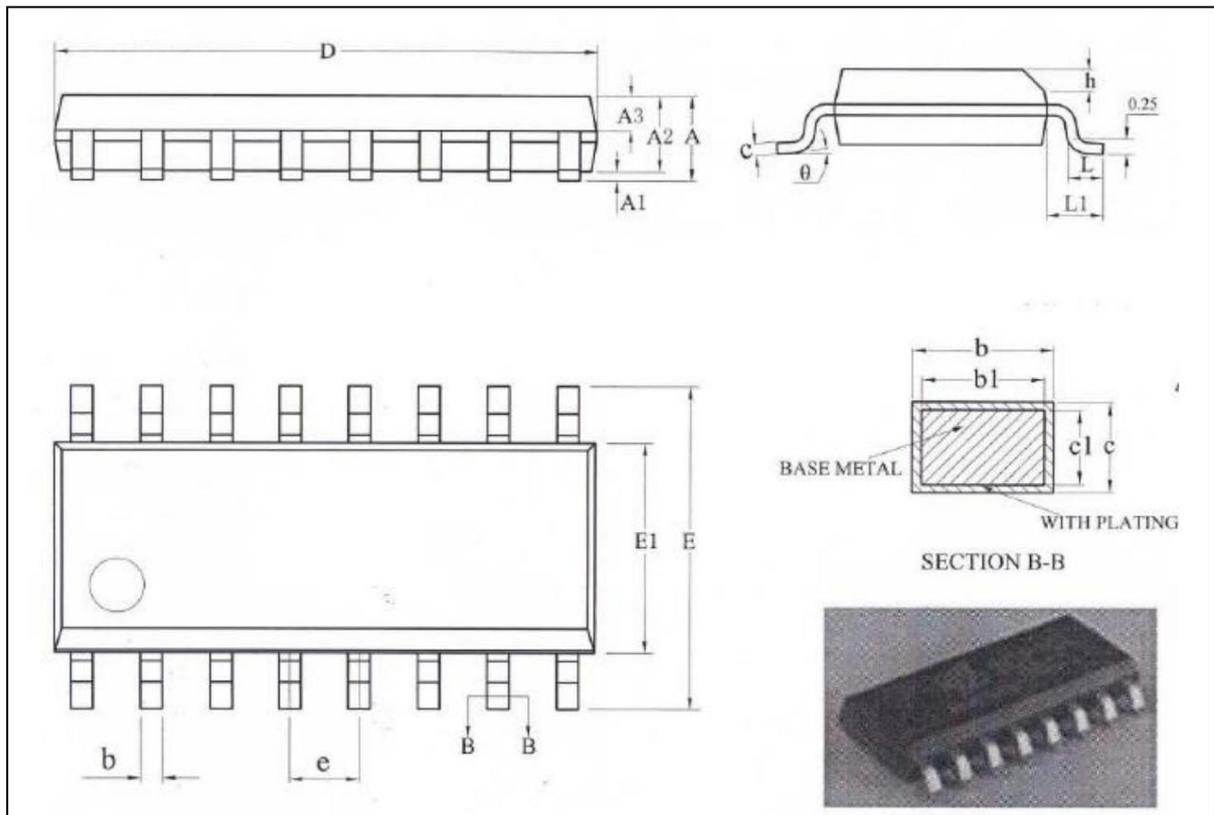
Zyklus: 1

Betroffenes Flag: Z Beispiel:

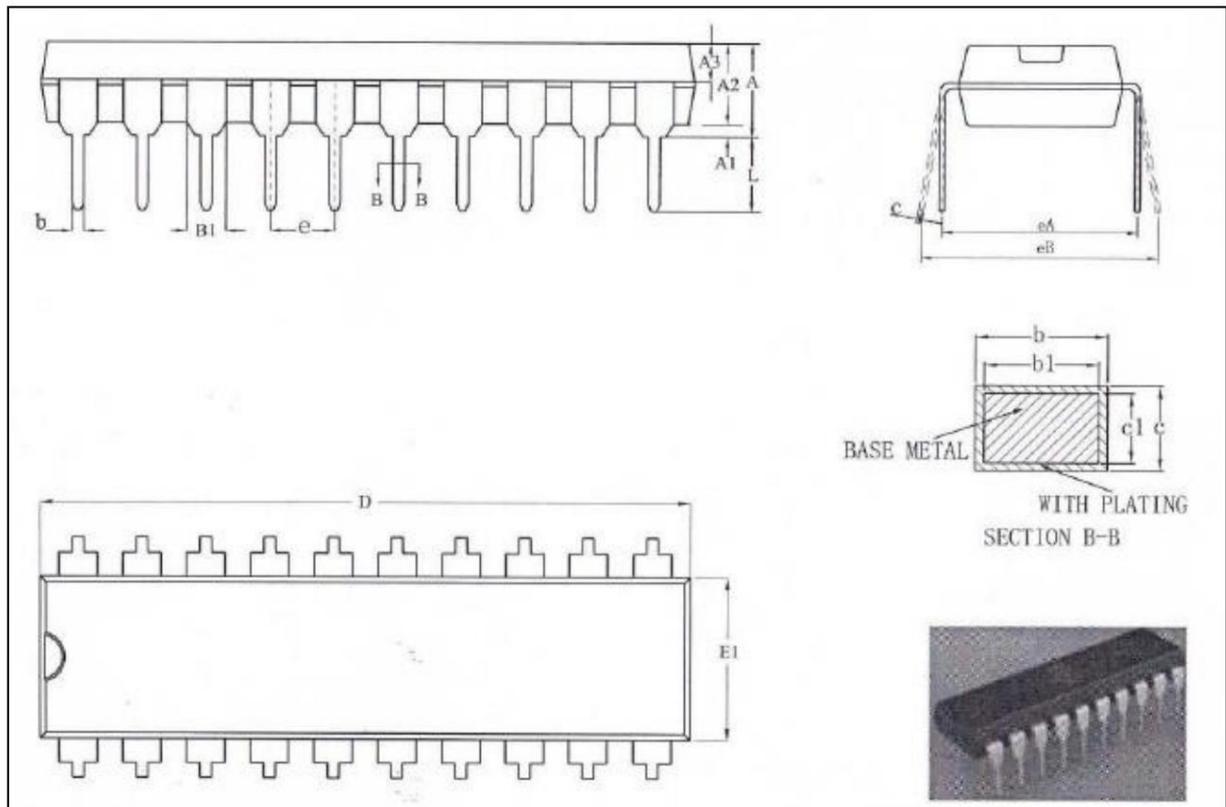
```
LDIA      0AH          ;ACC-Zuweisung 0AH
LD        R01,A        ;R01 ACC-Wert zuweisen, R01=0AH
LDIA      0FH          ;ACC-Zuweisung
XORR     R01           0FH ;Ausführungsergebnis: R01=05H
```

20. Einkapselung**20.1 DIP16**

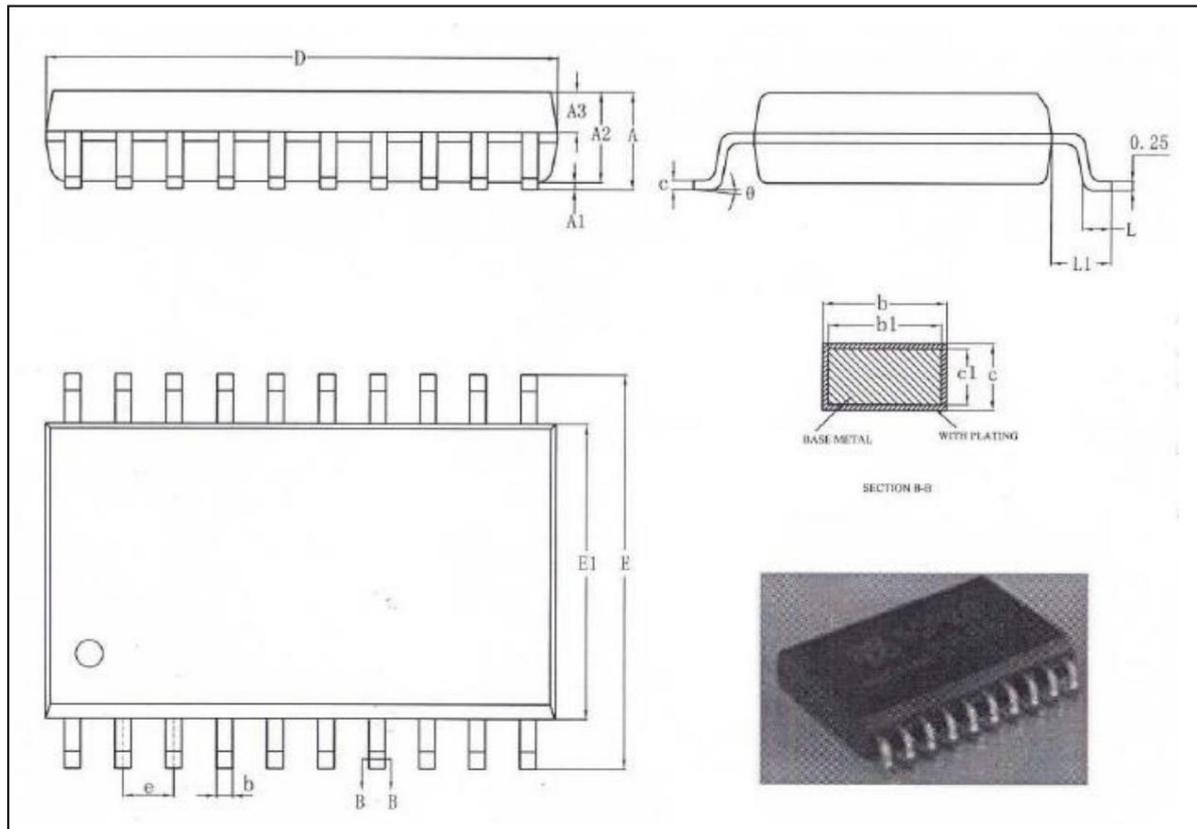
Symbol	Millimeter		
	Mindest	Nom	max
A	3,60	3,80	4,00
A1	0,51	-	-
A2	3,20	3,30	3,40
A3	1,47	1,52	1,57
B	0,44	-	0,52
b1	0,43	0,46	0,49
B1	1,52 REF		
c	0,25	-	0,29
c1	0,24	0,25	0,26
D	19,00	19,10	19,20
E1	6,25	6,35	6,45
e	2,54 BSC		
E	7,62 REF		
eB	7,62	-	9,30
eC	0	-	0,84
L	3,00	-	-

20.2 SOP16

Symbol	Millimeter		
	Mindest	Nom	max
A	-	-	1,75
A1	1.10	-	0,225
A2	1.30	1.40	1,50
A3	0,60	0,65	0,70
B	0,39	-	0,47
b1	0,38	0,41	0,44
c	0,20	-	0,24
c1	0,19	0,20	0,21
D.	9.80	9,90	10.00
E.	5,80	6.00	6.20
E1	3,80	3,90	4.00
e	1,27 BSC		
H	0,25	-	0,50
L	0,5	-	0,80
L1	1,05 REF		
γ	0	-	8°

20.3 DIP20

Symbol	Millimeter		
	Mindest	Nom	max
A	3,60	3,80	4,00
A1	0,51	-	-
A2	3,20	3,30	3,40
A3	1,47	1,52	1,57
B	0,44	-	0,52
b1	0,43	0,46	0,49
B1	1,52 REF		
c	0,25	-	0,29
c1	0,24	0,25	0,26
D	25,80	25,90	26,00
E1	6,45	6,55	6,65
e	2,54 BSC		
E	7,62 REF		
eB	7,62	-	9,30
eC	0	-	0,84
L	3,00	-	-

20.4 SOP20

Symbol	Millimeter		
	Mindest	Nom	max
A	-	-	2.65
A1	1.10	-	0.30
A2	2.25	2.30	2.35
A3	0.97	1.02	1.07
B	0.35	-	0.43
b1	0.34	0.37	0.40
c	0.25	-	0.29
c1	0.24	0.25	0.26
D.	12.70	12.80	12.90
E.	10.10	10.30	10.50
E1	7.40	7.50	7.60
e	1,27 BSC		
L	0,70	-	1.00
L1	1,40 REF		
γ	0	-	8°



Anweisungen zur Versionsrevision

Versionsnummer	Zeit	Änderungen																																																																																																												
V1.0	Juli 2016	CCP2IN2 im gesamten Dokument wurde geändert zu CCP2IN0 (P26, P95)																																																																																																												
V1.1	Juli 2016	<p>Die Prescaler-Anwendung fügt 2 Zeilen mit Anweisungen hinzu, um sicherzustellen, dass kein Zurücksetzen erfolgt</p> <pre> CLR TMR0 ;TMR0 löschen CLRWDT ;WDT gelöscht LDIA B'00xx1111' ; LD OPTION_REG,A LDIA B'00xx1xxx' ; Neuen Prescaler setzen LD OPTION_REG,A </pre>																																																																																																												
	SSPMSK-Adresse im Juli 2016 hinzugefügt. 1.																																																																																																													
V1.2	August 2016	<p>Kapitel 16.4 Originaldokument: 例 16-2 写数据 EEPROM 存储器</p> <pre> LD A,ADDRL ; 写地址 LD EEADR,A LD A,ADDRH LD EEADRH,A LD A,DATAL ; 写数据 LD EEDAT,A LD A,DATAH LD EEDATH,A SETB EECON1,EEPGD ; 选择程序存储器 SETB EECON1,WREN ; 使能写信号 CLR INTCON,GIE ; 关闭中断 </pre> <p>Änderungsschema: 4 Zeilen auswählen, die ersten beiden Zeilen ändern in: LD A, ADDR ; Adresse schreiben LD EEADR,A Löschen Sie die letzten beiden Zeilen.</p> <p>2. Kapitel 16.2.2</p> <p>EEPROM 地址寄存器 EEADR(10DH)</p> <table border="1"> <tr> <th>10DH</th> <th>Bit7</th> <th>Bit6</th> <th>Bit5</th> <th>Bit4</th> <th>Bit3</th> <th>Bit2</th> <th>Bit1</th> <th>Bit0</th> </tr> <tr> <td>EEADR</td> <td>---</td> <td>---</td> <td>---</td> <td>EEADR4</td> <td>EEADR3</td> <td>EEADR2</td> <td>EEADR1</td> <td>EEADR0</td> </tr> <tr> <td>读写</td> <td>---</td> <td>---</td> <td>---</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> </tr> <tr> <td>复位值</td> <td>---</td> <td>---</td> <td>---</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table> <p>Bit 7-6 --- (无关) Bit 4-0 EEADR<4:0>: 指定EEPROM读取/写入操作的地址</p> <p>EEPROM 数据寄存器 EEDATH(10EH)</p> <table border="1"> <tr> <th>10EH</th> <th>Bit7</th> <th>Bit6</th> <th>Bit5</th> <th>Bit4</th> <th>Bit3</th> <th>Bit2</th> <th>Bit1</th> <th>Bit0</th> </tr> <tr> <td>EEDATH</td> <td>EEDATH7</td> <td>EEDATH6</td> <td>EEDATH5</td> <td>EEDATH4</td> <td>EEDATH3</td> <td>EEDATH2</td> <td>EEDATH1</td> <td>EEDATH0</td> </tr> <tr> <td>读写</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> </tr> <tr> <td>复位值</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table> <p>Bit 7-0 EEDATH<7:0>: 对数据EEPROM 进行读取或写入数据的高8 位</p> <p>EEPROM 控制寄存器 EECON1 (18CH)</p> <table border="1"> <tr> <th>18CH</th> <th>Bit7</th> <th>Bit6</th> <th>Bit5</th> <th>Bit4</th> <th>Bit3</th> <th>Bit2</th> <th>Bit1</th> <th>Bit0</th> </tr> <tr> <td>EECON1</td> <td>EEPGD</td> <td>---</td> <td>---</td> <td>---</td> <td>WRERR</td> <td>WREN</td> <td>WR</td> <td>RD</td> </tr> <tr> <td>读写</td> <td>R/W</td> <td>---</td> <td>---</td> <td>---</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> <td>R/W</td> </tr> <tr> <td>复位值</td> <td>0</td> <td>---</td> <td>---</td> <td>---</td> <td>x</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table> <p>Bit 7-6 EEPCG: 数据EEPROM 使能位 1 = 允许操作数据EEPROM 0 = 禁止操作数据 EEPROM Bit 6-4 未用, 读为 0 Bit 3 WRERR: EEPROM 错误标志位 1 = 写操作过早终止 (正常工作期间的任何WDT 复位或欠压复位) 0 = 写操作完成</p> <p>1. Stelle: EEDATA1 auf EEADR1 geändert, EEDATA0 auf EEADR0 geändert, 2. Stelle: Bit7-6 auf Bit7 geändert</p>	10DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	EEADR	---	---	---	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	读写	---	---	---	R/W	R/W	R/W	R/W	R/W	复位值	---	---	---	0	0	0	0	0	10EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	EEDATH	EEDATH7	EEDATH6	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	读写	R/W	复位值	0	0	0	0	0	0	0	0	18CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	EECON1	EEPGD	---	---	---	WRERR	WREN	WR	RD	读写	R/W	---	---	---	R/W	R/W	R/W	R/W	复位值	0	---	---	---	x	0	0	0							
10DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0																																																																																																						
EEADR	---	---	---	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0																																																																																																						
读写	---	---	---	R/W	R/W	R/W	R/W	R/W																																																																																																						
复位值	---	---	---	0	0	0	0	0																																																																																																						
10EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0																																																																																																						
EEDATH	EEDATH7	EEDATH6	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0																																																																																																						
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W																																																																																																						
复位值	0	0	0	0	0	0	0	0																																																																																																						
18CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0																																																																																																						
EECON1	EEPGD	---	---	---	WRERR	WREN	WR	RD																																																																																																						
读写	R/W	---	---	---	R/W	R/W	R/W	R/W																																																																																																						
复位值	0	---	---	---	x	0	0	0																																																																																																						

3. Kapitel 2.1.2

表 2-2 CMS89F52X 特殊功能寄存器汇总 Bank0

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
00h	INDF	寻址该单元会使用FSR的内容寻址数据存储器 (不是物理寄存器)								xxxx xxxx
01h	TMR0	Tmr0数据寄存器								xxxx xxxx
02h	PCL	程序计数器低字节								0000 0000
03h	STATUS	IRP	---	---	TO	PD	Z	DC	C	0001 1xxx
04h	FSR	间接寄存器地址指针								xxxx xxxx
05h	PORTA	---	RA6	RA5	RA4	RA3	RA2	RA1	RA0	-xxx xxxx
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx
09h	PORTE	---	---	---	---	---	RE1	RE0	---	--- -0x
0Ah	PCLATH	---	---	---	程序计数器高5位的写缓冲器			---	---	-0 0000
0Bh	INTCON	GIE	PEIE	TOIE	INTE	---	TOIF	INTF	---	0000 -0-
0Ch	PIR1	EEIF	ADIF	SSPIF	BCLIF	CCPIF	---	TMR2IF	TMR1IF	0000 0-00
0Dh	PIR2	---	---	CSIF	CAIF	C3IF	C2IF	C1IF	PPGIF	-00 0000
0Eh	TMR1L	16位TMR1寄存器低字节的寄存器								xxxx xxxx
0Fh	TMR1H	16位TMR1寄存器高字节的寄存器								xxxx xxxx
10h	T1CON	T1GNV	TMR1GE	T1CKPS1	T1CKPS0	TOOSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000
11h	TMR2	TIMER2寄存器								0000 0000
12h	T2CON	---	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000
14h	PPGTMRL									
15h	PPGTMRH									
16h	PPGDLY									

1 Stelle: geändert in „PPGWDTIF“ 2
 Stellen: geändert in „---“ und „---“

4. Kapitel 2.1.2

表 2-3 CMS89F52X 特殊功能寄存器汇总 Bank1

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
80h	INDF	寻址该单元会使用FSR的内容寻址数据存储器 (不是物理寄存器)								xxxx xxxx
81h	OPTION_REG	RBPUL	INTEG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111
82h	PCL	程序计数器 (PC) 的低字节								0000 0000
83h	STATUS	IRP	---	---	TO	PD	Z	DC	C	0-1 1xxx
84h	FSR	间接寄存器地址指针								xxxx xxxx
85h	TRISA	---	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	-111 1111
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	---	1111 1111
89h	TRISE	---	---	---	TRISE3	TRISE2	TRISE1	TRISE0	---	--- 1111
8Ah	PCLATH	---	---	---	程序计数器高5位的写缓冲器			---	---	-0 0000
8Bh	INTCON	GIE	PEIE	TOIE	INTE	---	TOIF	INTF	---	0000 -0-
8Ch	PIE1	EEIE	ADIE	SSPIE	BCLIE	CCPIE	---	TMR2IF	TMR1IE	0000 0-00
8Dh	PIE2	---	---	CSIE	CAIE	C3IE	C2IE	C1IE	PPGIE	-00 0000
8Fh	OSCCON	---	IRCF2	IRCF1	IRCF0	---	---	---	---	-1 10 ---
90h	OSCTUNE	---	---	---	TUN4	TUN3	TUN2	TUN1	TUN0	-0 0000
92h	PR2	TIMER2寄存器								1111 1111
93h	CMICNT	CMICF	CMICNT[6:0]							0000 0000
94h	WPUA	---	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	-000 0000
95h	WPUB	---	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	0000 0000

1 Stelle: ändern auf
 „---“ 2 Stellen: ändern auf
 „TRISB0“ 3 Stellen: ändern auf
 „---“ und „---“ 4 Stellen: ändern
 auf „ADIE“ 5 Stellen: ändern auf "PPGWDTIE"

5. Kapitel 2.1.2

表 2-5 CMS89F52X 特殊功能寄存器汇总 Bank3

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
100h	INDF	寻址该单元会使用FSR的内容寻址数据存储器 (不是物理寄存器)								xxxx xxxx
101h	OPTION_REG	RBPUL	INTEG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111
102h	PCL	程序计数器 (PC) 的低字节								0000 0000
103h	STATUS	IRP	---	---	TO	PD	Z	DC	C	0001 1xxx
104h	FSR	间接寄存器地址指针								xxxx xxxx
106h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111
107h	PAANSEL	---	PAANS6	PAANS5	PAANS4	PAANS3	PAANS2	PAANS1	PAANS0	-000 0000
108h	PBANSSEL	PBANS7	PBANS6	PBANS5	PBANS4	PBANS3	PBANS2	PBANS1	PBANS0	0000 0000
109h	PEANSEL	---	---	---	---	---	PEANS1	PEANS0	---	--- 00
10Ah	PCLATH	---	---	---	程序计数器高5位的写缓冲器			---	---	-0 0000
10Bh	INTCON	GIE	PEIE	TOIE	INTE	---	TOIF	INTF	---	0000 0000
10Ch	EEDCON1	EEPGRD	---	---	WRERR	WRREN	WR	RD	---	0- -000
10Dh	EEDCON2	EEPROM控制寄存器2 (不是物理寄存器)								---
10Eh	CCPRL	捕捉寄存器的低字节								xxxx xxxx
10Fh	CCPRH	捕捉寄存器的高字节								xxxx xxxx
110h	CCPCON	CCPEN	---	---	CCPS1	CCPS0	CPTCS2	CPTCS1	CPTCS0	0-0000
111h	SSPMASK	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	11111111
112h	SSPADD	同步串行接口 (I ² C) 模式地址寄存器								0000 0000
113h	SSPBUF	同步串行接口接收缓冲器/发送寄存器								xxxx xxxx

1. Stelle: Wechsel zu
 „---“ 2. Stelle: „CPTCS2“ geändert zu „CCPM2“, „CPTCS1“ geändert zu „CCPM1“,
 "CPTCS0" geändert in "CCPM0"

6. Kapitel 15.3

PPG 控制寄存器 PPGCON(17H)

17H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPGCON	DETC5F	DETC4F	----	RELOAD_EN	DETC4EN	DETC3EN	PPGMD	PPG_ON
R/W	R/W	R/W	----	R/W	R/W	R/W	R/W	R/W
复位值	1	1	----	0	0	0	0	0

- Bit7 **DETC5F**: 比较器 5 状态位 (PPG 状态位)
 0: 清零比较器 5 的 0->1 翻转标志, (若 DETC4EN=1, 则 PPG 重新打开)
 1: 有比较器 5 的 0->1 翻转, 写 1 无效, (若 DETC4EN=1, 则 PPG 关闭)
- Bit6 **DETC4F**: 比较器 4 状态位 (PPG 状态位)
 0: 清零比较器 4 的 1->0 翻转标志, (若 DETC3EN=1, 则 PPG 重新打开)
 1: 有比较器 4 的 1->0 翻转, 写 1 无效, (若 DETC3EN=1, 则 PPG 关闭)
- Bit5 **禁用**
- Bit4 **RELOAD_EN**: PPG TMR 加载使能
 0: 允许加载 (自动降功率模式时, 同时允许 PPG TMR 自-1)
 1: 禁止加载 (自动降功率模式时, 同时禁止 PPG TMR 自-1)
- Bit3 **DETC5EN**: 比较器 5 关闭 PPG 使能位
 0: 禁止
 1: 使能
- Bit2 **DETC4EN**: 比较器 4 关闭 PPG 使能位

1. Stelle: „DETC4EN“ wurde in „DETC5EN“ , „DETC3EN“ geändert in "DETC4EN"

V1.3

geändert Im November 2017 waren viele Ausdrücke im Text falsch