PIC-Grundlagen Teil 2

Control-Register und Timer der PIC 16C5x-Familie beschreibt der zweite Teil dieser Artikelserie.

Indirekte Adressierung (INDF- und FSR-Register)

Das INDF-Register an Andresse 00h ermöglicht eine indirekte Adressierung von File-Registern. Dabei ist das INDF-Register nicht wirklich hardwaremäßig vorhanden. Alle Befehle, die auf das INDF-Register zugreifen, arbeiten nicht mit diesem Register, sondern mit dem Register, das über das FSR-Register adressiert ist. Das FSR-Register ist somit ein Zeiger auf ein File-Register, auf das dann über das INDF-Register zugegriffen werden kann.

Das FSR-Register befindet sich an der Adresse 04h (siehe Abbildung 8). Das Bit 7 ist immer „1" und hat keine Bedeutung. Die unteren 5 Bits D 0 bis D 4 dienen zur Auswahl der Registeradresse. Mit ihnen sind insgesamt 32 Register (Adresse 00h bis 1Fh) adressierbar. Die Auswahl der Bank erfolgt mit den Bits D 5 und D 6, mit denen die Bänke 0 bis 3 ausgewählt werden können.

Zu beachten ist hierbei, daß die Register 00h bis 0Fh der Bänke 1 bis 3 gespiegelt sind und über diese auf die Register der Bank 0 zugreifen.

Zur indirekten Adressierung wird zum Beispiel das FSR-Register mit 10111010 (binär) beschrieben, um das File-Register 1Ah (= 11010) der Bank 1 anzusprechen. Alle Befehle, die jetzt auf das INDF-Register zugreifen, verändern das adressierte File-Register.

Programm Counter

Der Programmzähler zeigt auf die Speicherstelle im Programmspeicher, die gerade ausgeführt wird. Nach jedem Befehl wird dieser um eine Stelle erhöht. Bei Sprungbefehlen wird der Programmzähler direkt geladen, wodurch das Programm ab dieser Position fortgesetzt wird.

Die Bitbreite des Programmzählers ist abhängig von der Größe des Programmspeichers. So verfügten der PIC 16C54 A und der PIC 16C55 über 512 x 12 Bit Programmspeicher, der mit einem 9 Bit breiten Programmzähler adressiert wird. Für den PIC 16C58A und PIC 16C57 ist ein 11 Bit breiter Programmzähler erforderlich, um deren Programmspeicher mit 2048 x 12 Bit adressieren zu können.

Die unteren 8 Bit des Programmzählers sind über das PCL-Register an der Adresse 02h zu erreichen und können ausgelesen oder gesetzt werden.

Für den PIC 16C54A und PIC 16C55 ist

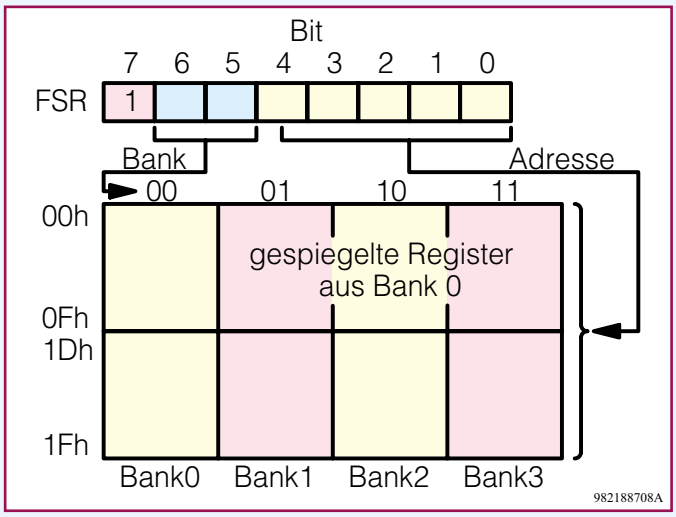


Bild 8: Auswahl der Register und Bänke mit dem FSR-Register

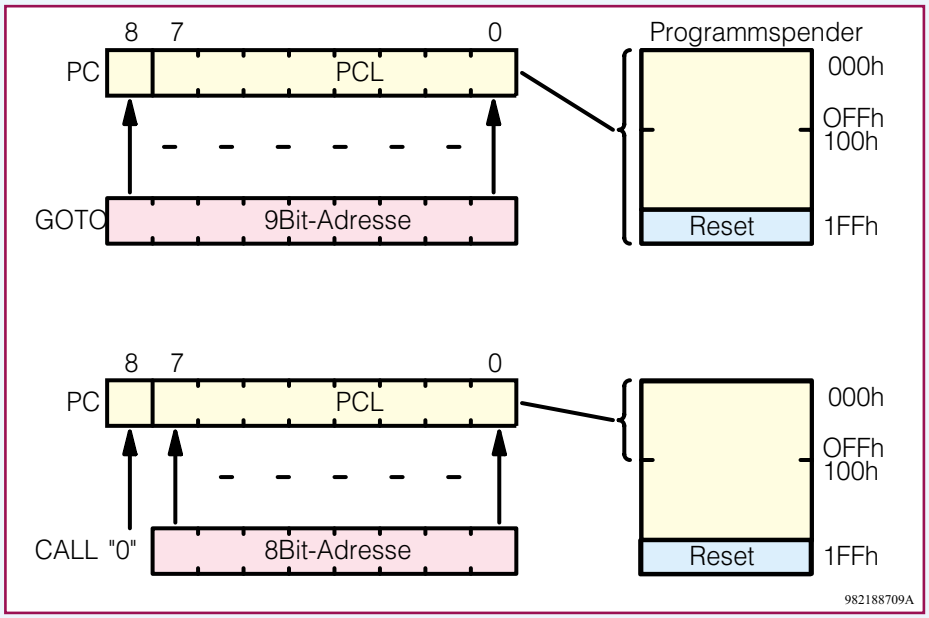


Bild 9: Laden des Programmzählers beim PIC 16C54A und PIC 16C55

der Programmzähler in Abbildung 9 dargestellt.

Mit einer GOTO-Anweisung ist ein Sprung zu einer bestimmten Position im Programm realisierbar.

Dabei wird die 9 Bit breite Adresse aus dem Programmcode in den Programmzähler geladen, womit jede Speicherstelle angesprungen werden kann.

Um im Programmablauf eine Unterfunktion aufzurufen und danach das Programm weiter fortzusetzen, dient die CALL-Anweisung. Diese lädt nur die unteren 8 Bit des Programmzählers und setzt das obere Bit auf Null.

Ebenso kann ein Sprungziel berechnet und die Zieladresse direkt in das PCL-Register geschrieben werden. Auch hierbei werden nur die unteren 8 Bit des Programmzählers gesetzt und das neunte Bit gelöscht. Bei CALL-Anweisungen und berechneten Sprungzielen darf das Ziel also nur im unteren Adressbereich von 000h bis 0FFh liegen, da diese mit 8 Bit adressierbar sind.

Der Programmzähler für den PIC 16C57 und PIC 16C58A ist in Abbildung 10 dargestellt.

Bei diesen Typen ist der Programmspeicher in 4 Seiten zu je 512 x 12 Bit aufgeteilt. Die Auswahl der Seiten erfolgt über die Bits PA 0 und PA 1 des STATUS-Registers.

Bei einer GOTO-Anweisung werden dabei die unteren 9 Bit vom Sprungziel übernommen und die oberen zwei Bit aus dem STATUS-Register übertragen. Bei einer CALL-Funktion oder dem Springen über das PCL-Register werden hier die unteren 8 Bit gesetzt, das neunte Bit gelöscht und die oberen zwei Bit aus dem STATUS-Register übertragen.

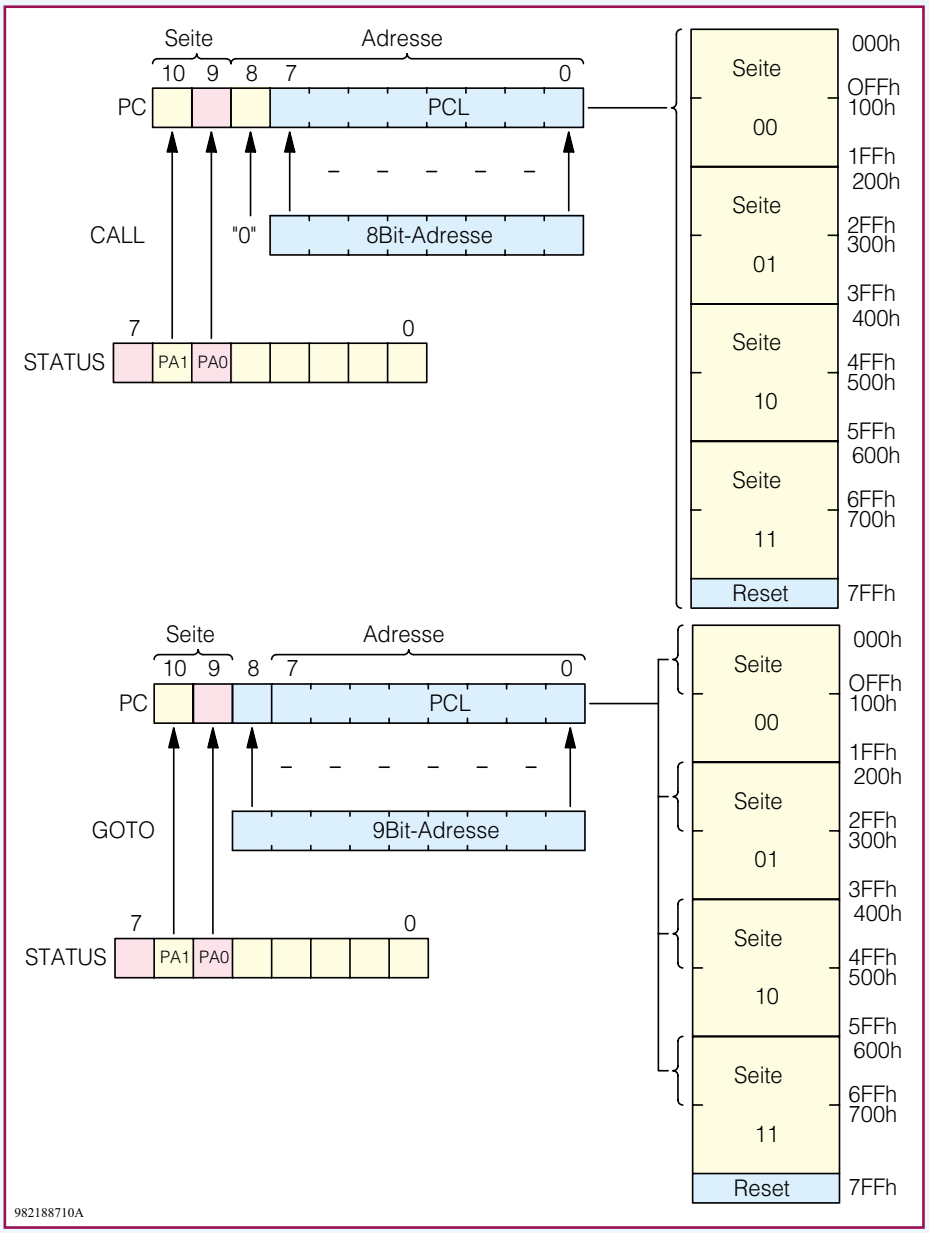
Bild 10: Laden des Programmzählers beim PIC 16C57 und PIC 16C58A

Tabelle 2:
Teilerfaktoren für den Vorteiler

PS2..PS0	Timer-Teiler	WDT-Teiler
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

Bei Sprüngen ist daher immer darauf zu achten, daß die richtige Seite des Ziels über das STATUS-Register angegeben ist und daß mit CALL-Anweisungen nur Ziele in der unteren Hälfte der Seiten erreichbar sind.

Wird bei einem linearen Programmablauf auf die nächste Seite gewechselt, so erhöht sich der Programmzähler. So wird der Programmzähler zum Beispiel von 1FFh auf 200h erhöht und dabei auf die zweite Bank gewechselt. Zu beachten ist hierbei, daß sich die Bits PA 0 und PA 1



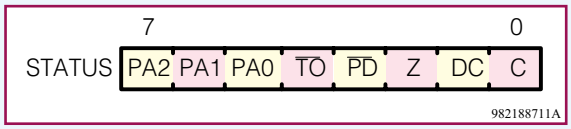


Bild 11: STATUS-Register

dabei nicht ändern. Vor jedem Sprung ist daher zu prüfen, ob die PA-Bits auf die aktuelle Seite eingestellt sind.

STATUS-Register

Das STATUS-Register befindet sich an der Adresse 03h und ist in Abbildung 11 dargestellt.

Das Carry-Bit (Bit 0) dient zum Übertrag bei allen Rechen- und Schiebeoperationen. Es wird gesetzt, wenn bei einer Addition ein Übertrag aufgetreten ist. Bei einer Subtraktion sollte das Carry-Flag zuvor gesetzt werden, da es hier im Falle eines Übertrags (negative Zahl) gelöscht wird.

Das Digital-Carry-Bit (Bit 1) erfüllt die gleiche Funktion wie das Carry-Flag, wobei es sich lediglich auf die unteren 4 Bit bezieht. D. h., entsteht zum Beispiel bei einer Addition eine Zahl, die 15 überschreitet, wird das Bit besetzt.

Das Zero-Bit (Bit 2) wird gesetzt, wenn sich nach einer Operation der Wert Null ergibt.

Das Power-Down-Bit (Bit 3) ist gelöscht, wenn sich der Controller im Power-Down-Mode befindet.

Das Time-Out-Bit (Bit 4) ist gelöscht, wenn der Watchdog nicht zurückgesetzt wurde.

Die Bits PA 0 und PA 1 dienen zur Auswahl der Seiten des Programmspeichers, während das Bit PA 2 nicht genutzt wird.

OPTION-Register

Das OPTION-Register besitzt 6 Bit und ist in Abbildung 12 dargestellt. Bei der PIC 16C5x-Familie kann auf dieses Register mit dem OPTION-Befehl zugegriffen werden, wobei es nur beschrieben werden kann. Es dient zur Steuerung des Timers und des Watchdogs.

Timer/Watchdog-Timer

Das Timer-Modul des PIC-Controllers kann als Timer oder Zähler über das OPTION-Register konfiguriert werden. Mit dem Bit T0SC wird zwischen dem internen Takt und dem externen Takt am T0CKI-Pin gewählt.

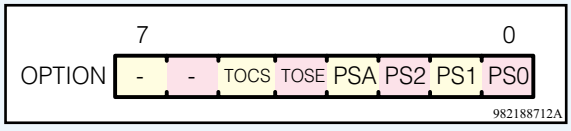


Bild 12: OPTION-Register

Ist das Bit gelöscht, so wird das interne Taktsignal genutzt, das ein Viertel der Oszillatorfrequenz beträgt. Ist das Bit gesetzt, wird der Takt vom T0CKI-Pin genutzt.

Das T0SE-Bit invertiert den Pegel des T0CKI-Pins. Ist das Bit gelöscht, so erhält der Zähler einen Takt bei steigender Flanke am T0CKI-Pin und ist das Bit gesetzt, wird ein Takt bei fallender Flanke generiert.

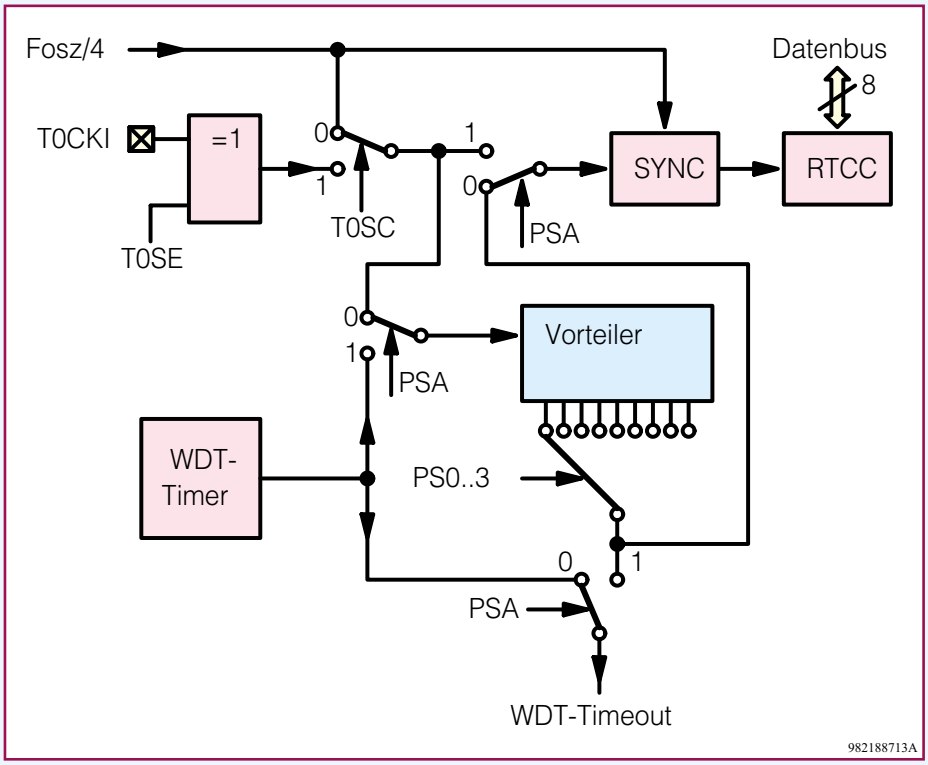


Bild 13: Blockschaltbild des Timers

Der Takt kann direkt oder über einen Vorteiler auf den Zähler gegeben werden, der durch das PSA-Bit bestimmt ist. Ist das Bit gesetzt, gelangt der Takt über eine Synchronisationsstufe, die den Takt mit dem Oszillator synchronisiert auf den Zähler. Ist das PSA-Bit gelöscht, so wird der Takt über den Vorteiler geführt, dessen Teilerfaktor über die Bits PS 0 bis PS 2 bestimmt wird. Der Zählerstand kann über das TMR0-Register an Adresse 01h ausgelesen und gesetzt werden.

Der PIC-Mikrocontroller verfügt über einen internen Watchdog, um zu verhindern, daß sich der Controller „aufhängt". Der Watchdog ist bei der Programmierung des Controllers zu aktivieren und muß durch die Software kontinuierlich zurückgesetzt werden. Erfolgt dieses nicht, da der Controller zum Beispiel abgestürzt ist, so wird nach ca. 18 ms ein Reset durchgeführt, der den Controller definiert in den Anfangszustand versetzt.

Wird der Vorteiler nicht für den Timer genutzt, so kann er für den Watchdog Einsatz finden, indem das PSA-Bit im OPTION-Register gesetzt wird. Wird hier zum Beispiel ein Teilerfaktor von 128 gewählt, ist die Timeoutzeit des Watchdogs auf ca. 2,3 Sekunden verlängerbar.

Der Faktor des Vorteilers ist durch die Bits PS 0 bis PS 3 bestimmt und entspricht dabei den Teilerfaktoren, die in Tabelle 2 dargestellt sind.

STACK

Die PIC16C5x-Familie verfügt über einen 2-Ebenen-Hardwarestack. Dieser nimmt bei einer CALL-Anweisung den Inhalt des Programmzählers auf, der dann bei Beenden der Unterfunktion zurückgeschrieben wird.

Der Stack kann maximal zwei Rücksprungadressen speichern, so daß nicht mehr als zwei ineinander verschachtelte CALL-Anweisungen verwendet werden dürfen.

Im dritten Teil dieser Artikelserie wenden wir uns der Beschreibung der Assemblerbefehle zu. ELV