

Funktionsweise des PIC 16F84

Die folgende Präsentation zeigt Ihnen, wie

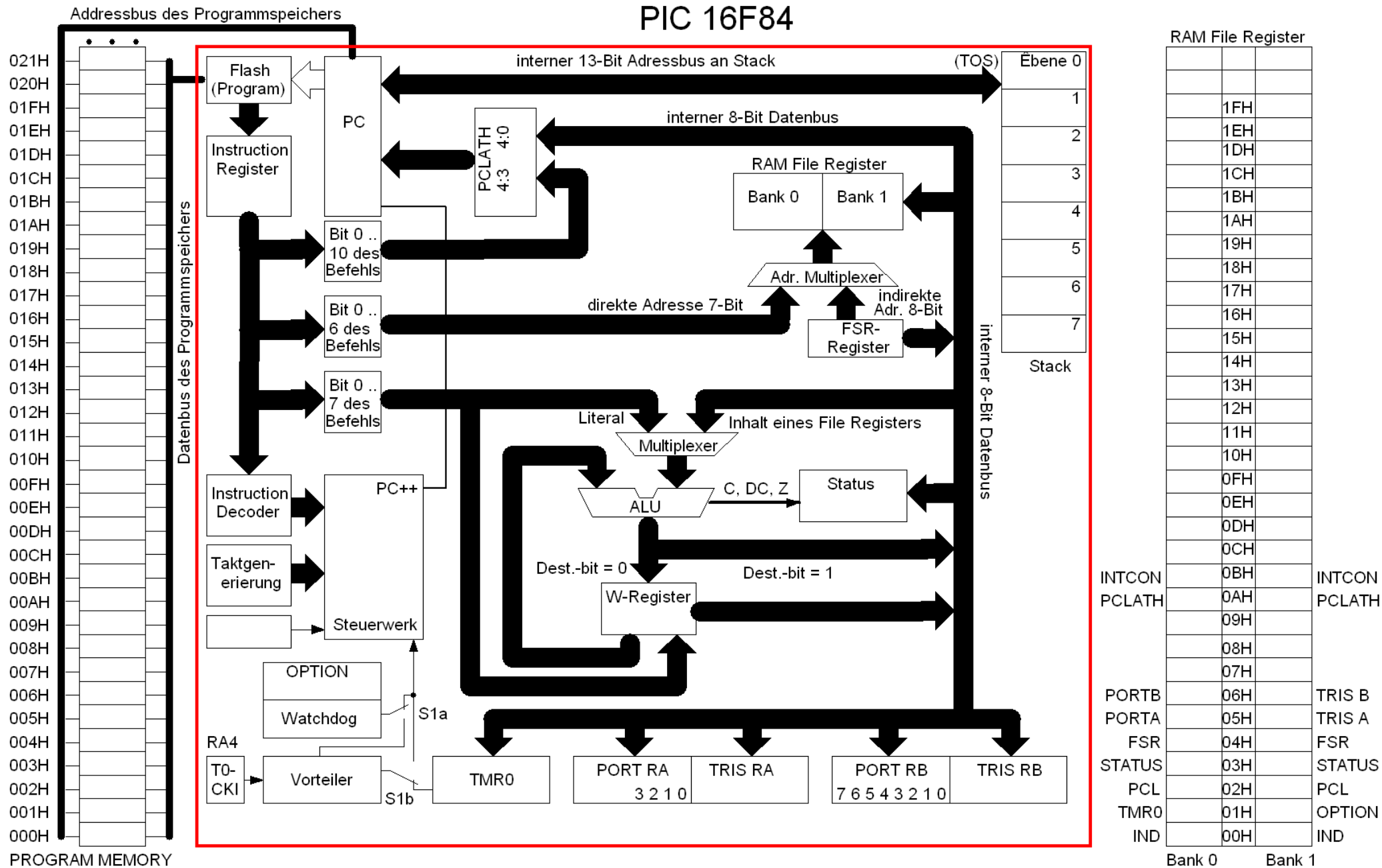
- Die Funktionsgruppen dieses μC interagieren
- Ein RESET wirkt
- Ein Befehl Schritt-für-Schritt abgearbeitet wird
- Ein einfaches Pipelining funktioniert
- Daten intern zwischen Registern verschoben werden
- Die Ports funktionieren

Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Der rot umrandete Teil ist der Mikrocontroller. Der Programm- und Datenspeicher sind seitlich vergrößert dargestellt.



PROGRAM MEMORY

Interne Zustand:



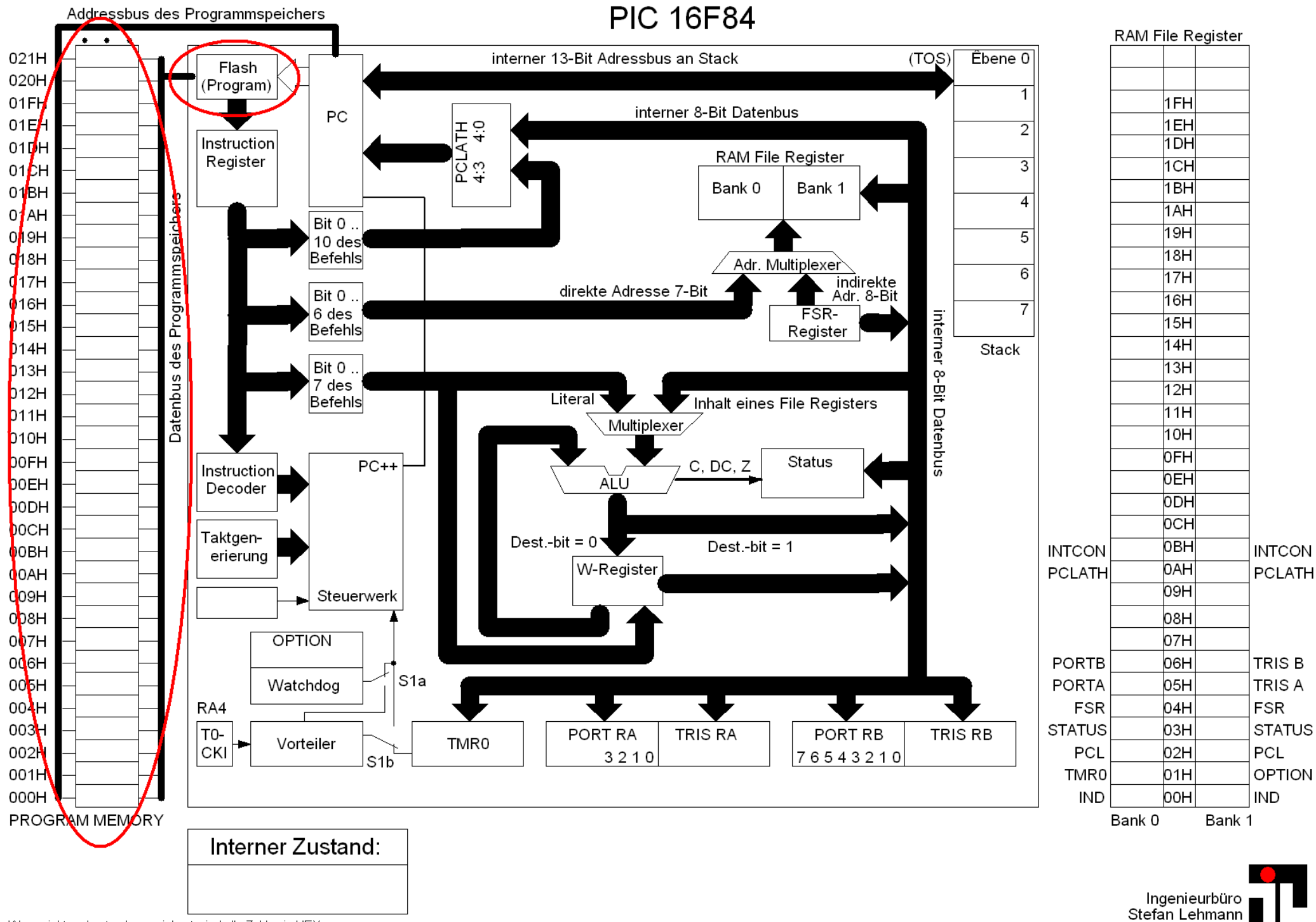
Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Der Programmspeicher

Er ist 1024 x 14 Bit großer EEPROM-Speicher. In ihm stehen die einzelnen Befehle die nach und nach abgearbeitet werden sollen.



Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Der Programmspeicher

Angesteuert wird der Programmspeicher vom Programmzähler (PC). Dieser zeigt auf den Befehl, der als nächstes eingelesen wird.

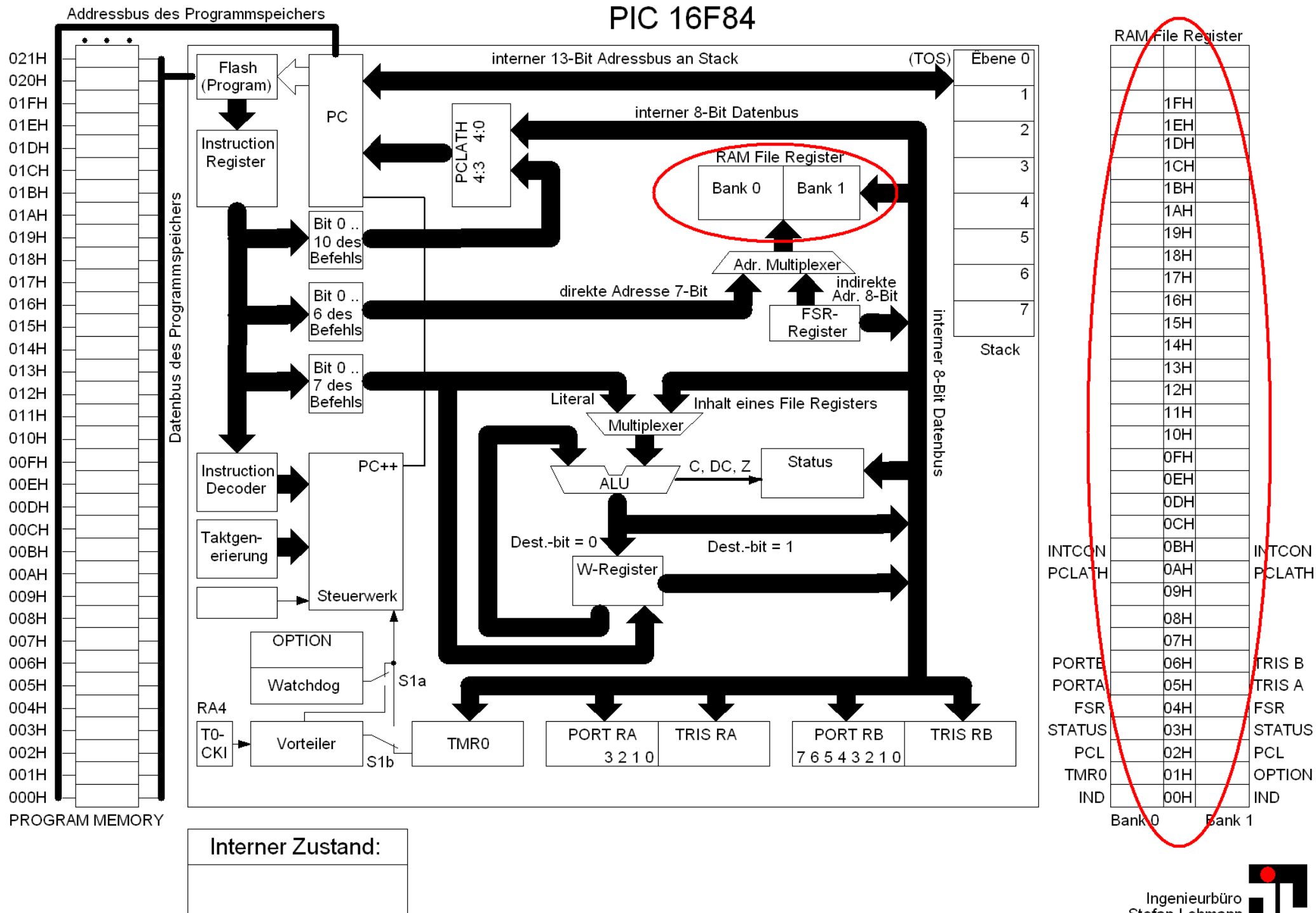
Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Der Datenspeicher

Der Datenspeicher ist von Programmspeicher völlig getrennt. Er hat einen eigenen Daten-, Adress- und Steuerbus. Es gibt relativ wenige 8-Bit RAM-Speicher.



Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Der Datenspeicher

Der Datenspeicher ist in zwei Bänke aufgeteilt. Die SFR sind Register die die Funktionsweise des μ Cs beeinflussen. In den GPR legt der Programmierer seine Variablen ab.

Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Das W-Register

Das Working-Register liegt nicht im bekannten RAM-Adressbereich. Es ist lediglich ein Hilfsregister, in dem ein Argument steht, das in der ALU verknüpft werden soll.

RAM File Register		
	1FH	
	1EH	
	1DH	
	1CH	
	1BH	
	1AH	
	19H	
	18H	
	17H	
	16H	
	15H	
	14H	
	13H	
	12H	
	11H	
	10H	
	0FH	
	0EH	
	0DH	
	0CH	
N	0BH	INTCON
H	0AH	PCLATH
	09H	
	08H	
	07H	
B	06H	TRIS B
A	05H	TRIS A
R	04H	FSR
S	03H	STATUS
L	02H	PCL
O	01H	OPTION
D	00H	IND
Bank 0	Bank 1	

--	--

Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Die ALU (Arithmetic Logic Unit)

Die ALU verknüpft zwei Argumente. Die Verknüpfungsarten sind hier: ADD, SUB, AND, OR, XOR und NOT (1-er Komplement).

In Abhängigkeit vom Ergebnis werden Flags gesetzt.

RAM File Register		
	1FH	
	1EH	
	1DH	
	1CH	
	1BH	
	1AH	
	19H	
	18H	
	17H	
	16H	
	15H	
	14H	
	13H	
	12H	
	11H	
	10H	
	0FH	
	0EH	
	0DH	
	0CH	
N	0BH	INTCON
H	0AH	PCLATH
	09H	
	08H	
	07H	
B	06H	TRIS B
A	05H	TRIS A
R	04H	FSR
S	03H	STATUS
L	02H	PCL
O	01H	OPTION
D	00H	IND
Bank 0	Bank 1	

Interner Zustand:

Funktionsweise des PIC 16F84

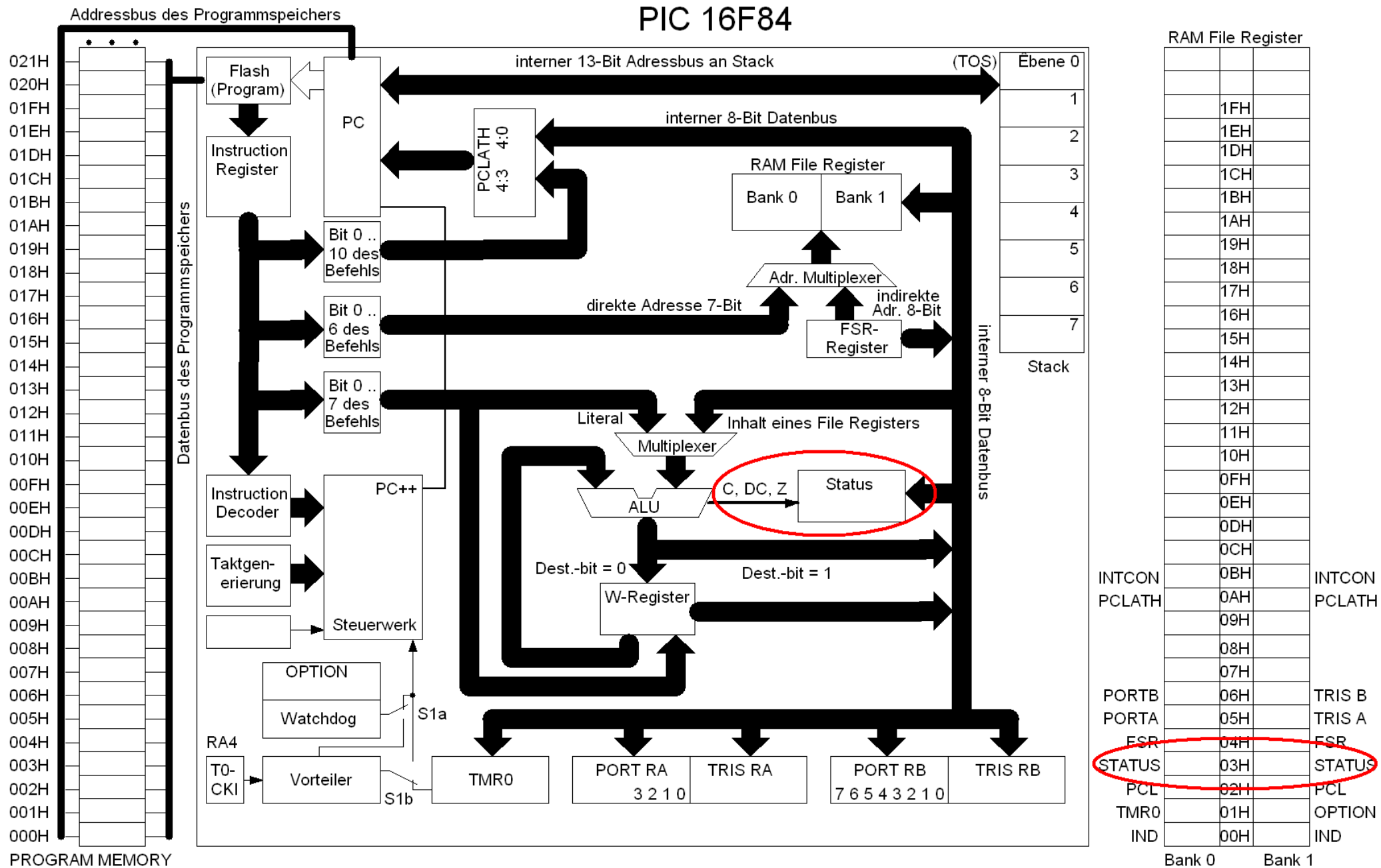
Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Das Statusregister

Ist das Ergebnis einer Verknüpfung Null, wird das Zero-Flag gesetzt. Das Carry- und Digitcarry-Flag zeigen einen Bereichsüberlauf an. Der Zahlenbereich bei 8-Bits umfasst die Werte 0 bis 255.

Vorsicht: Carry bei ADD, $\overline{\text{Carry}}$ (=Borrow) bei SUB



Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Das Statusregister

Von der ALU unabhängige Bits:

IRP, RP1, RP0, \overline{TO} und \overline{PD}

Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Das File-Select-Register (FSR)

Dieses Register enthält den Zeiger (Fileadresse) wenn eine indirekte Adressierung gemacht werden soll. Beim Zugriff auf Fileadresse 0 (IND) wird diese 0 durch den Inhalt des FSR ersetzt.

RAM File Register		
	1FH	
	1EH	
	1DH	
	1CH	
	1BH	
	1AH	
	19H	
	18H	
	17H	
	16H	
	15H	
	14H	
	13H	
	12H	
	11H	
	10H	
	0FH	
	0EH	
	0DH	
	0CH	
N	0BH	INTCON
H	0AH	PCLATH
	09H	
	08H	
	07H	
B	06H	TRIS B
A	05H	TRIS A
R	04H	FSP
S	03H	STATUS
L	02H	PCL
O	01H	OPTION
D	00H	IND

Bank 0
Bank 1

Interner Zustand:

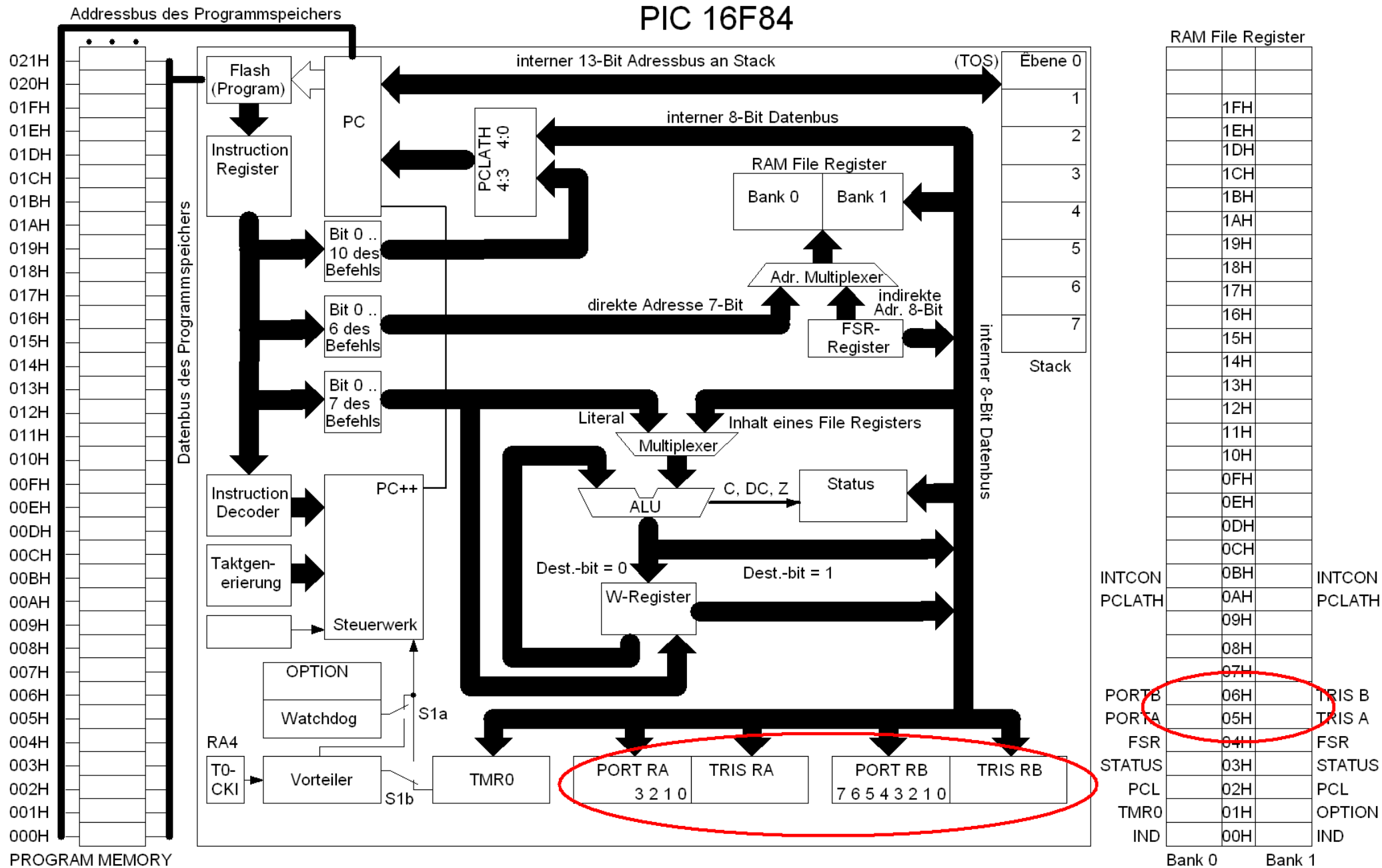
Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Die IO-Leitungen

Ein 4-Bit Port RA und ein 8-Bit Port RB stellen die Verbindung zur Außenwelt her. Jede Leitung kann individuell im TRIS-Register als Ein- oder Ausgang definiert werden.



Funktionsweise des PIC 16F84

Welche Funktionsgruppen hat ein PIC 16F84?

Eine Übersicht

Die IO-Leitungen

Die eigentliche Ausgabe erfolgt durch Beschreiben des PORT-Registers. Um die Eingangspegel einzulesen, reicht es aus, dieses PORT-Register zu lesen.

Funktionsweise des PIC 16F84

AUF DIE PLÄTZE

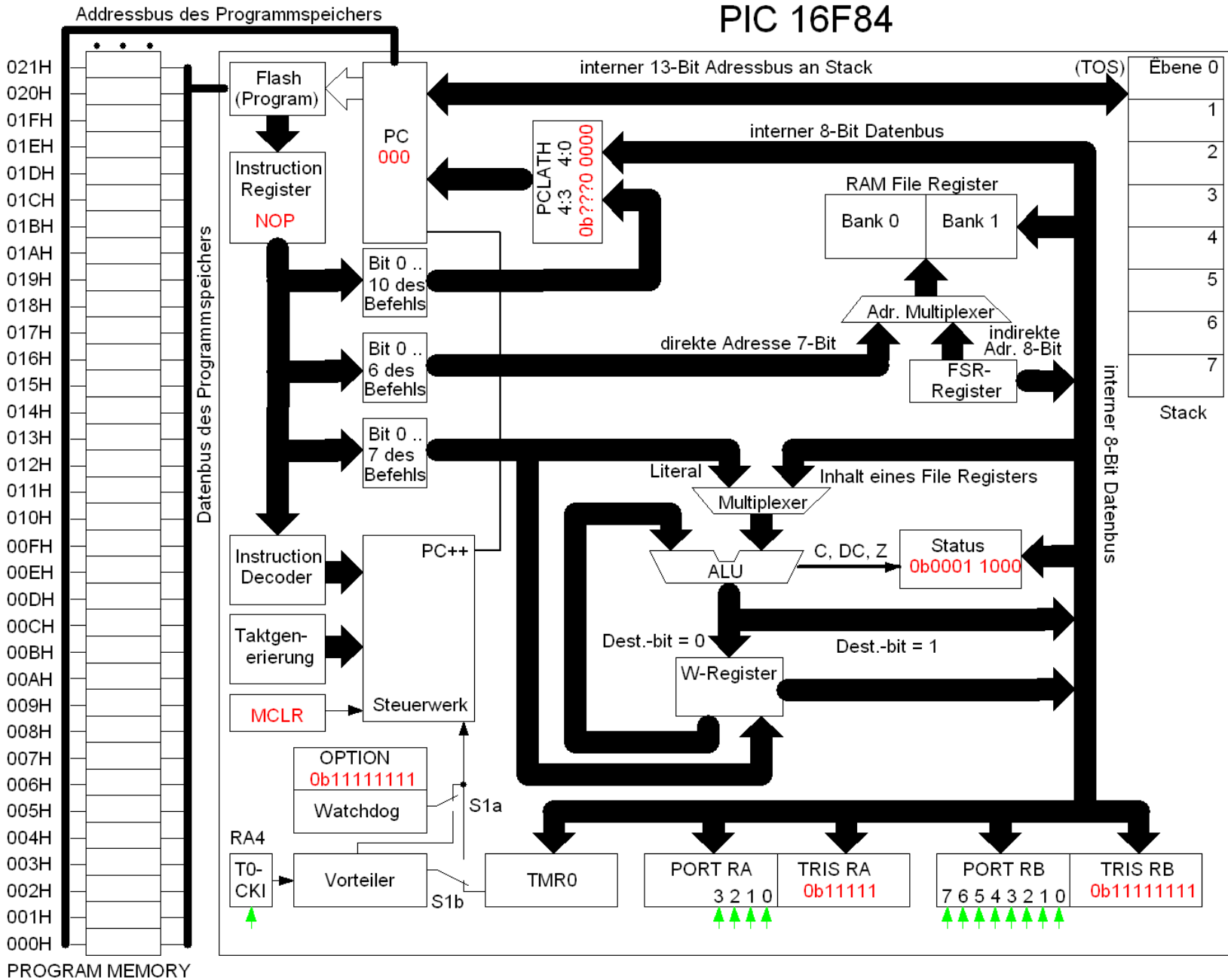
FERTIG

LOS

Funktionsweise des PIC 16F84

Anfangs ist alles undefiniert. Um einen geordneten Start zu erzwingen, muss ein RESET ausgeführt werden.

Dazu wird am $\overline{\text{MCLR}}$ -Pin ein Low-Pegel angelegt.



RAM File Register

		1FH		
		1EH		
		1DH		
		1CH		
		1BH		
		1AH		
		19H		
		18H		
		17H		
		16H		
		15H		
		14H		
		13H		
		12H		
		11H		
		10H		
		0FH		
		0EH		
		0DH		
		0CH		
INTCON	00	0BH	00	INTCON
PCLATH	00	0AH	00	PCLATH
		09H		
		08H	00	
		07H		
PORTB		06H	FF	TRIS B
PORTA		05H	FF	TRIS A
FSR		04H		FSR
STATUS	18	03H	18	STATUS
PCL	00	02H	00	PCL
TMR0		01H	FF	OPTION
IND		00H		IND
Bank 0		Bank 1		

Funktionsweise des PIC 16F84

Der RESET bewirkt, dass einige bestimmte Register mit Werten initialisiert werden.

Programmzähler = 0x0000 (PCL, PCLATH)

TRIS-Register = 0xFF

OPTION-Register = 0xFF

INTCON-Register = 0x00

STATUS-Register = 0x18 ($\overline{\text{PD}}$, $\overline{\text{TO}}$)

Pipeline = 0x00 (=NOP)

Funktionsweise des PIC 16F84

Sobald der RESET-Zustand verlassen wird, beginnt der μC mit der Befehlsabarbeitung.

Pro Befehlstakt sind vier Quarztakte notwendig. Dabei werden sowohl die steigende, als auch die fallende Flanke genutzt.

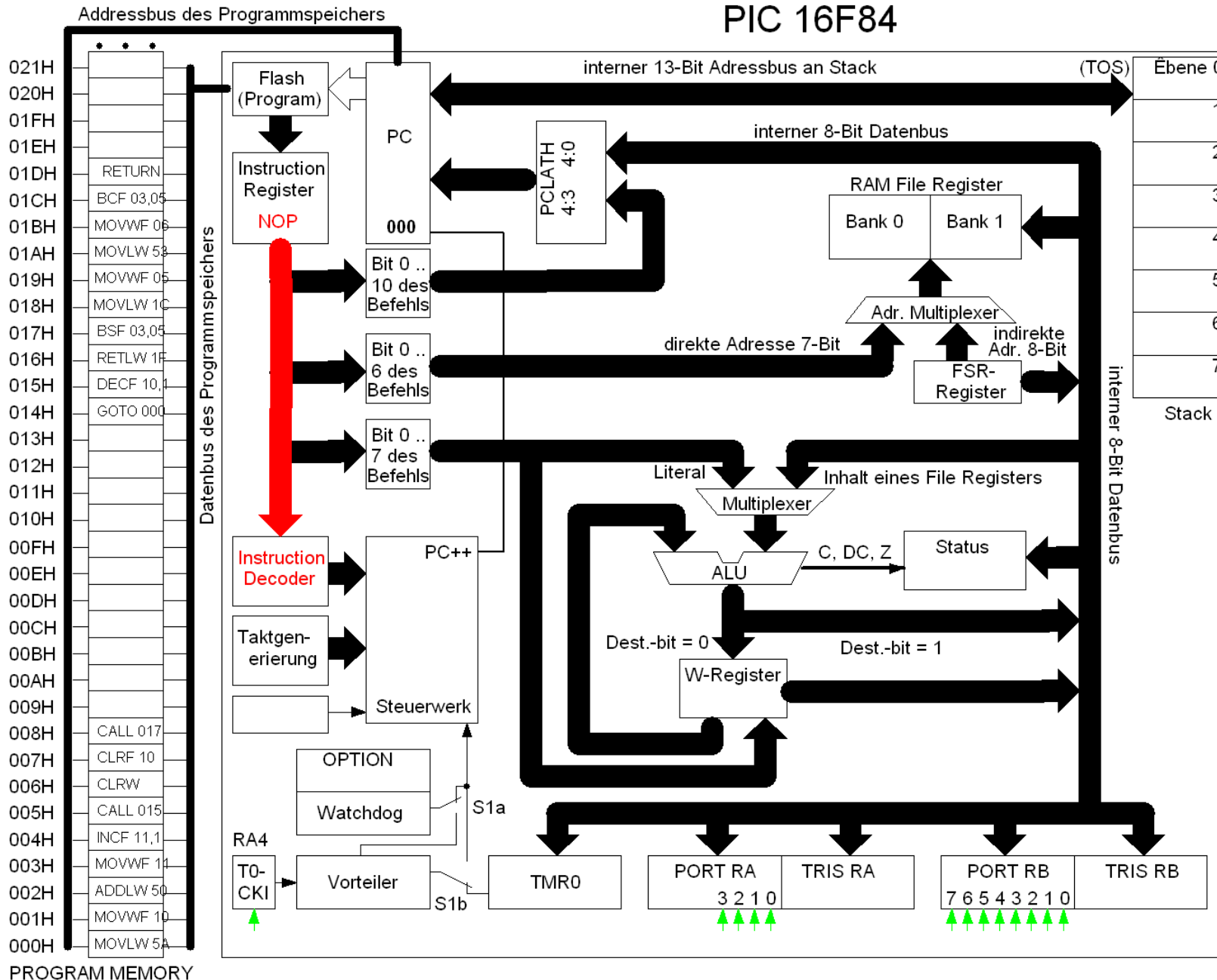
Funktionsweise des PIC 16F84

Im FETCH-Zyklus wird der Befehl aus dem Befehlsregister in den Befehlsdekoder übernommen.

Hier wird der Befehl analysiert und die weiteren Schritte vorbereitet. Was und wie das passiert hängt vom einzelnen Befehl ab. Dies ist die Dekodierphase (Decode-Zyklus).

Aktuell wird der Befehl NOP abgearbeitet.

PIC 16F84



RAM File Register

		1FH			
		1EH			
		1DH			
		1CH			
		1BH			
		1AH			
		19H			
		18H			
		17H			
		16H			
		15H			
		14H			
		13H			
		12H			
		11H			
		10H			
		0FH			
		0EH			
		0DH			
		0CH			
INTCON	00	0BH	00	INTCON	
PCLATH	00	0AH	00	PCLATH	
		09H			
		08H	00		
		07H			
PORTB		06H	FF	TRIS B	
PORTA		05H	FF	TRIS A	
FSR		04H		FSR	
STATUS	18	03H	18	STATUS	
PCL	00	02H	00	PCL	
TMR0		01H	FF	OPTION	
IND		00H		IND	

Bank 0 Bank 1

Interner Zustand:

Takt 1

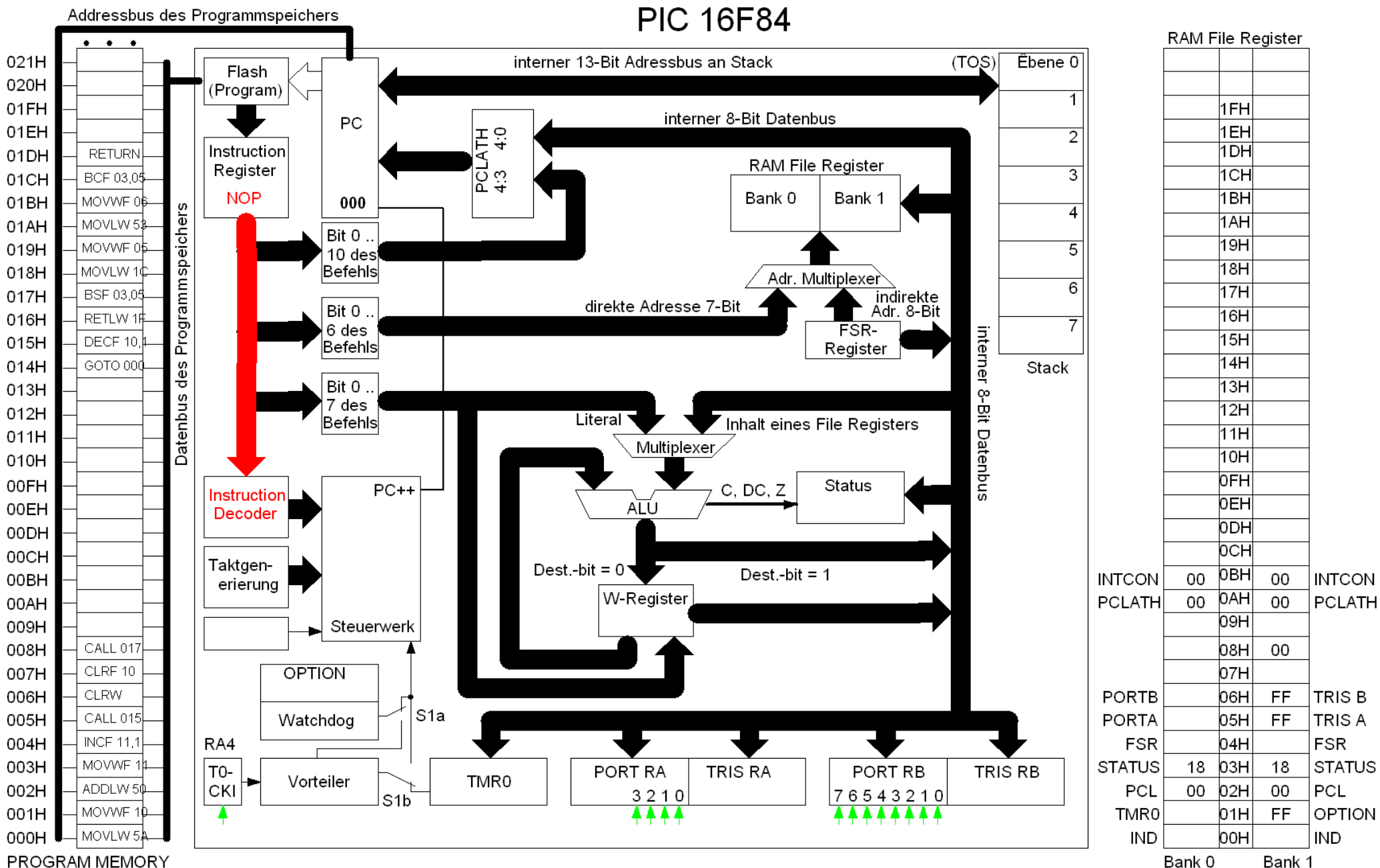
Befehl in Pipeline wird abgearbeitet

Befehl aus Pipeline in Befehlsdekode



Funktionsweise des PIC 16F84

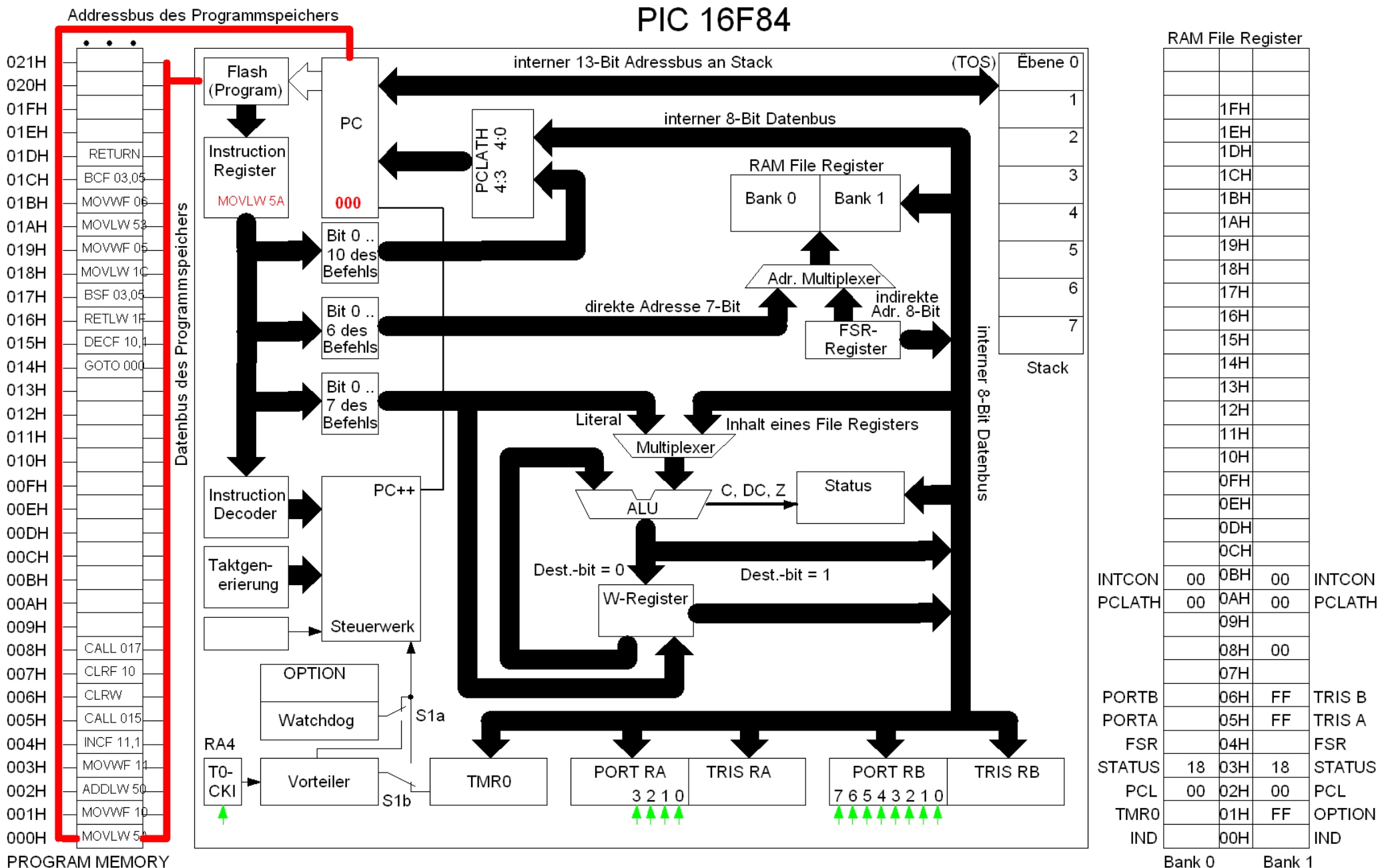
Der NOP-Befehl macht im Grund nichts. Er benötigt einen Befehlstakt an Zeit. Warum dieser offensichtlich „unnütze“ Befehl trotzdem sinnvoll ist, werden wir später sehen.



Funktionsweise des PIC 16F84

Im EXECUTE-Zyklus, passiert beim NOP-Befehl nichts, aber es kann zu diesem Zeitpunkt bereits der nächste Befehl in das Befehlsregister geladen werden.

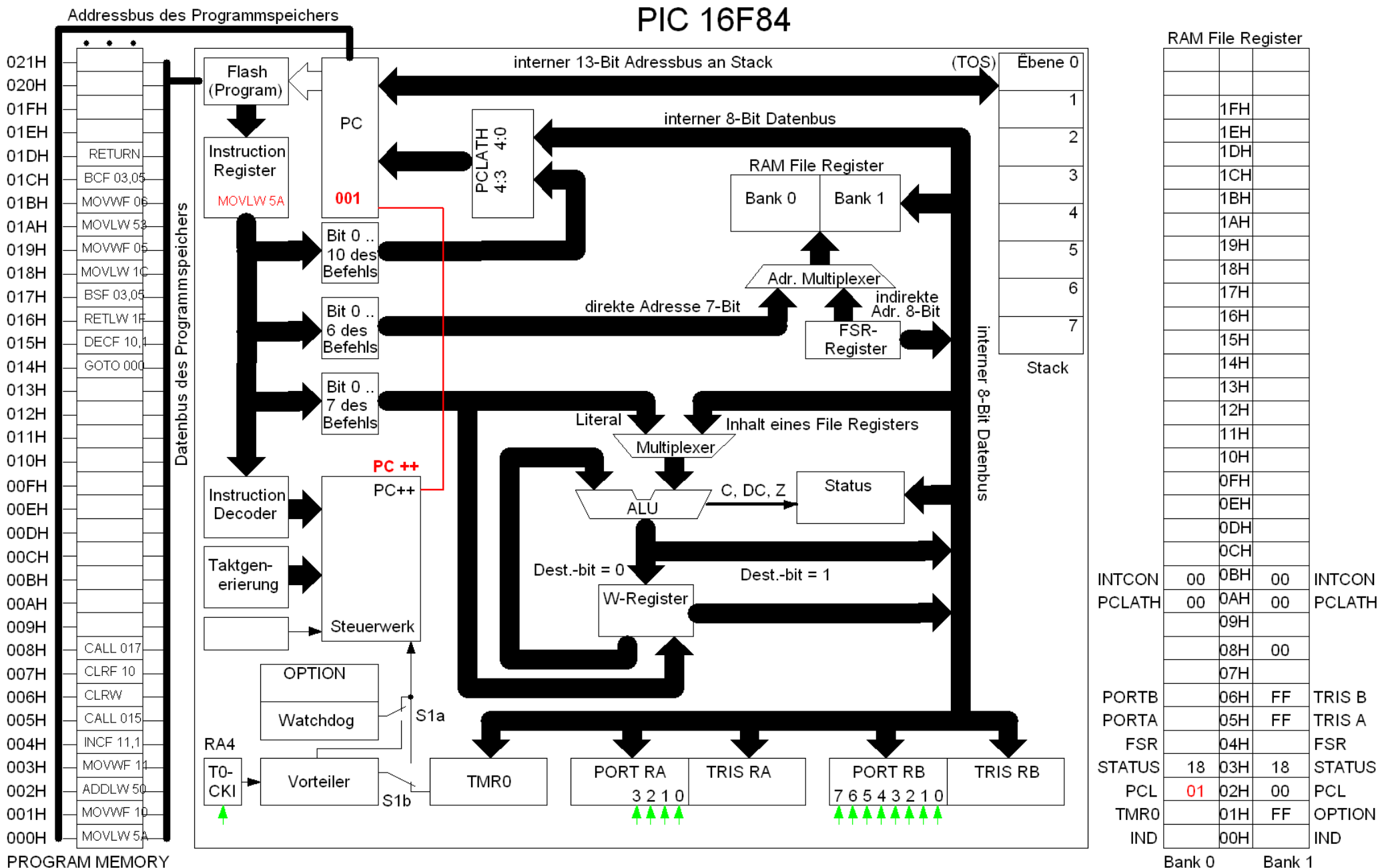
Da der Programm- und Datenspeicher physikalisch getrennt sind, ist diese Überlappung von EXECUTE und dem FETCH des nächsten Befehls möglich.



Funktionsweise des PIC 16F84

Beim FETCH wird der Inhalt des Programmzählers auf den Adressbus des Programmspeichers gelegt. Dann wird die entsprechende Speicheradresse gelesen und dessen Inhalt über den Datenbus des Programmspeichers in das Befehlsregister kopiert.

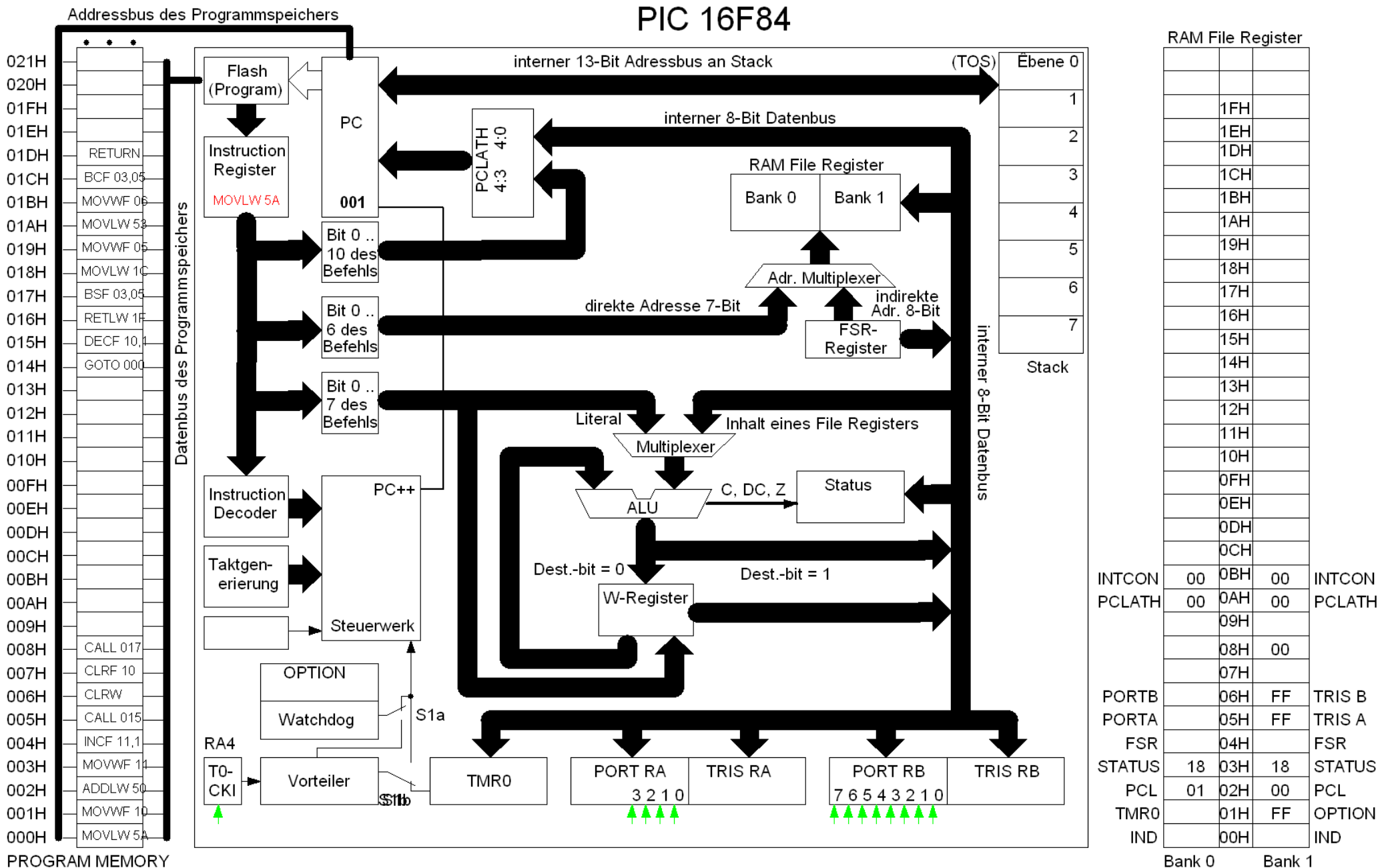
Sobald dieser Lesevorgang beendet ist, wird der Programmzähler um eins erhöht.



Funktionsweise des PIC 16F84

Damit ist der erste Befehl, er stammt noch vom RESET, abgearbeitet.

Der erste Befehl aus dem Programmspeicher steht nun im Befehlsregister zur Abarbeitung bereit.



Interner Zustand:

-

Befehl ist komplett abgearbeitet
In der Pipeline steht der Befehl aus Adresse 000
Der Programmzähler zeigt bereits auf Adresse 001



Funktionsweise des PIC 16F84

Im FETCH-Zyklus wird der Befehl aus dem Befehlsregister (Pipeline) in den Befehlsdekoder übernommen.

		1FH	
		1EH	
		1DH	
		1CH	
		1BH	
		1AH	
		19H	
		18H	
		17H	
		16H	
		15H	
		14H	
		13H	
		12H	
		11H	
		10H	
		0FH	
		0EH	
		0DH	
		0CH	
N	00	0BH	00
H	00	0AH	00
		09H	
		08H	00
		07H	
B		06H	FF
A		05H	FF
R		04H	
S	18	03H	18
-	01	02H	00
0		01H	FF
0		00H	

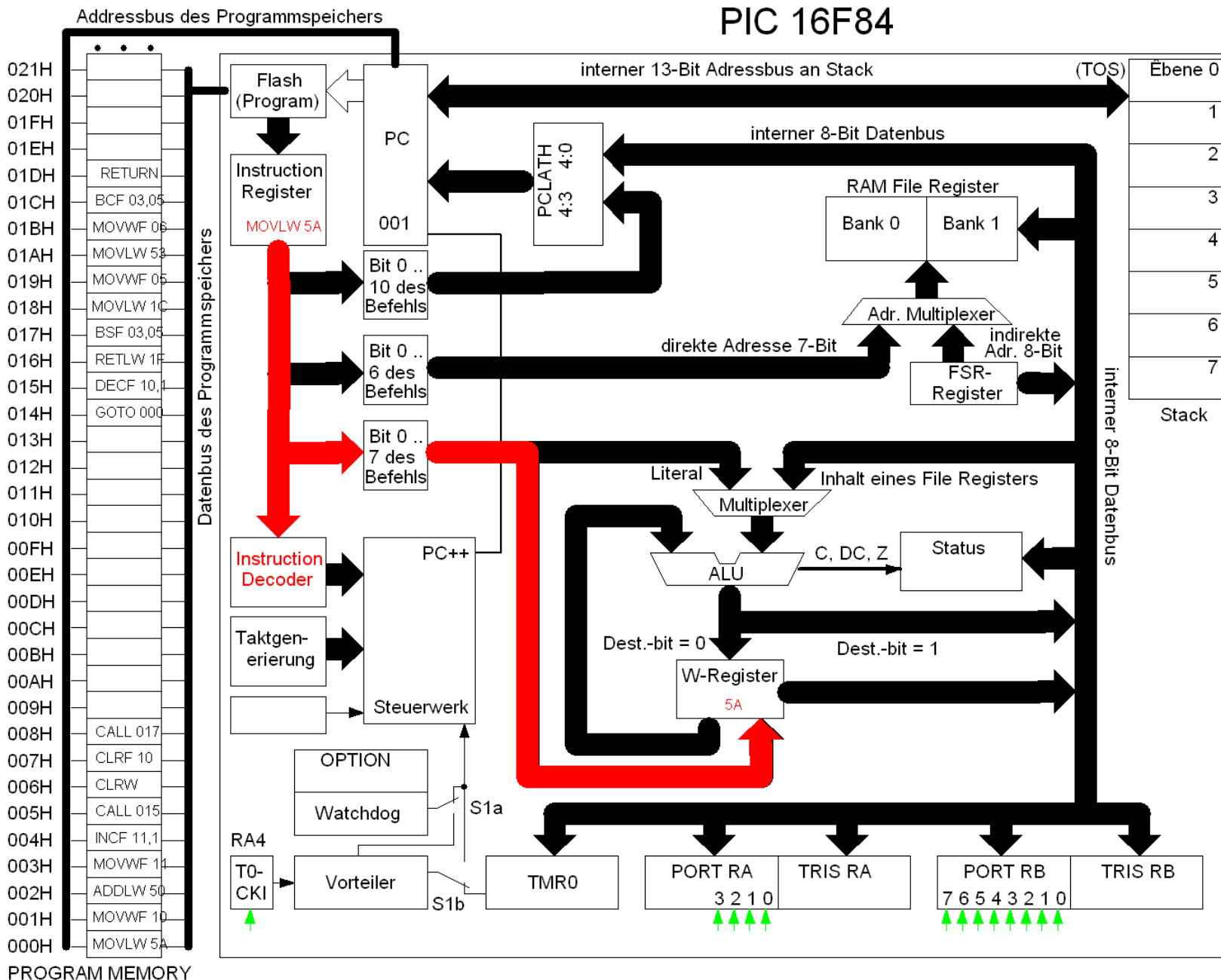
Ingenieurbüro
Stefan Lehmann

Funktionsweise des PIC 16F84

Beim Dekodieren erkennt der μC , dass die unteren 8-Bit des Binärcodes als Literal verwendet werden sollen.

Das Literal wird abgetrennt und dem W-Register zugeführt.

Mehr passiert bei diesem Befehl nicht mehr.



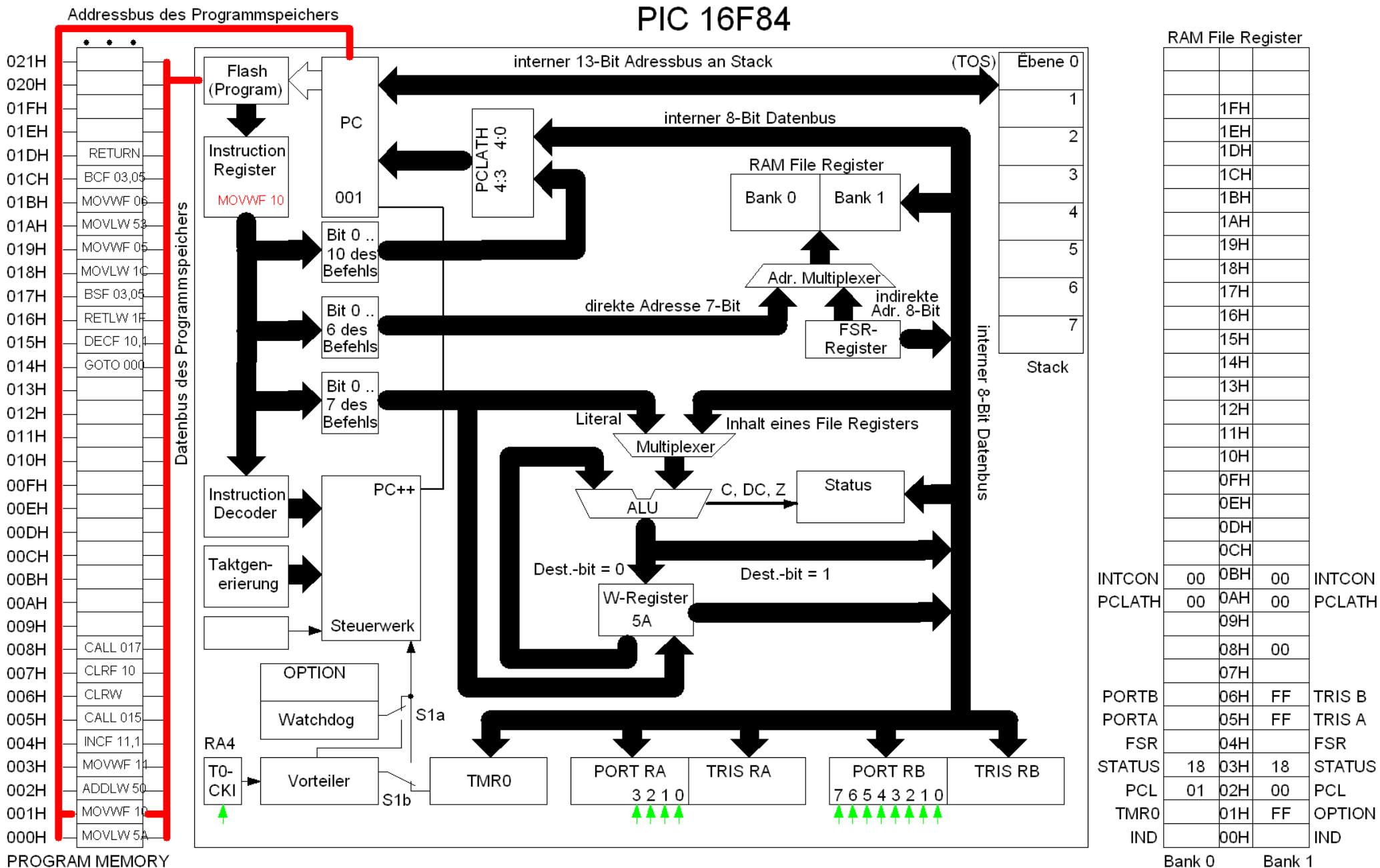
RAM File Register			
		1FH	
		1EH	
		1DH	
		1CH	
		1BH	
		1AH	
		19H	
		18H	
		17H	
		16H	
		15H	
		14H	
		13H	
		12H	
		11H	
		10H	
		0FH	
		0EH	
		0DH	
		0CH	
INTCON	00	0BH	00
PCLATH	00	0AH	00
		09H	
		08H	00
		07H	
PORTB		06H	FF
PORTA		05H	FF
FSR		04H	
STATUS	18	03H	18
PCL	01	02H	00
TMR0		01H	FF
IND		00H	
Bank 0		Bank 1	



Funktionsweise des PIC 16F84

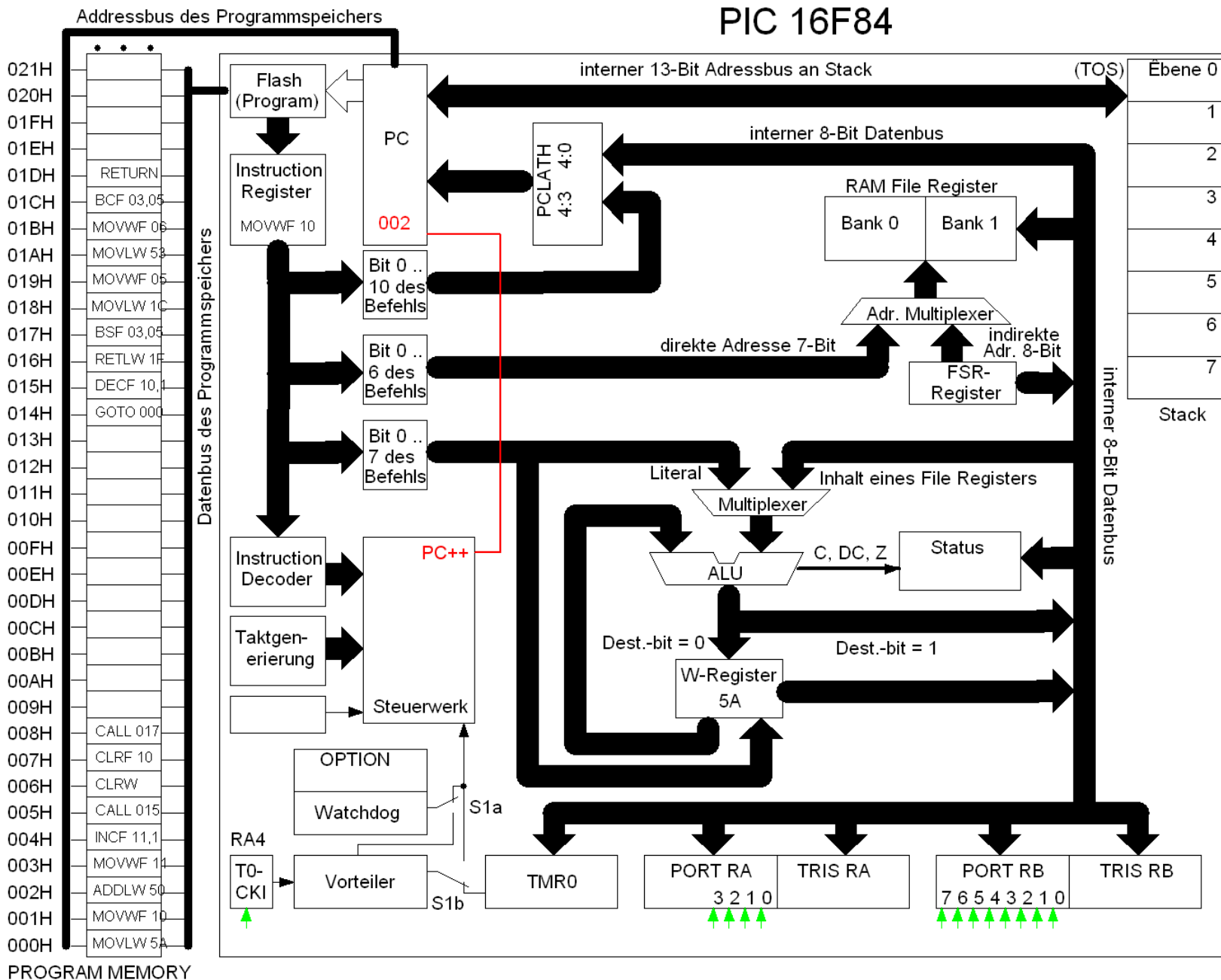
Der nächste Befehl kann nun aus dem Programmspeicher gelesen werden.

Dazu wird der Inhalt des Programmzählers auf den Adressbus des Programmspeichers gelegt und der Inhalt der Adresse 001 über den Datenbus in das Befehlsregister (Pipeline) eingelesen.



Funktionsweise des PIC 16F84

Und zum Abschluss des FETCH-Zykluses wird der Programmzähler wieder inkrementiert.



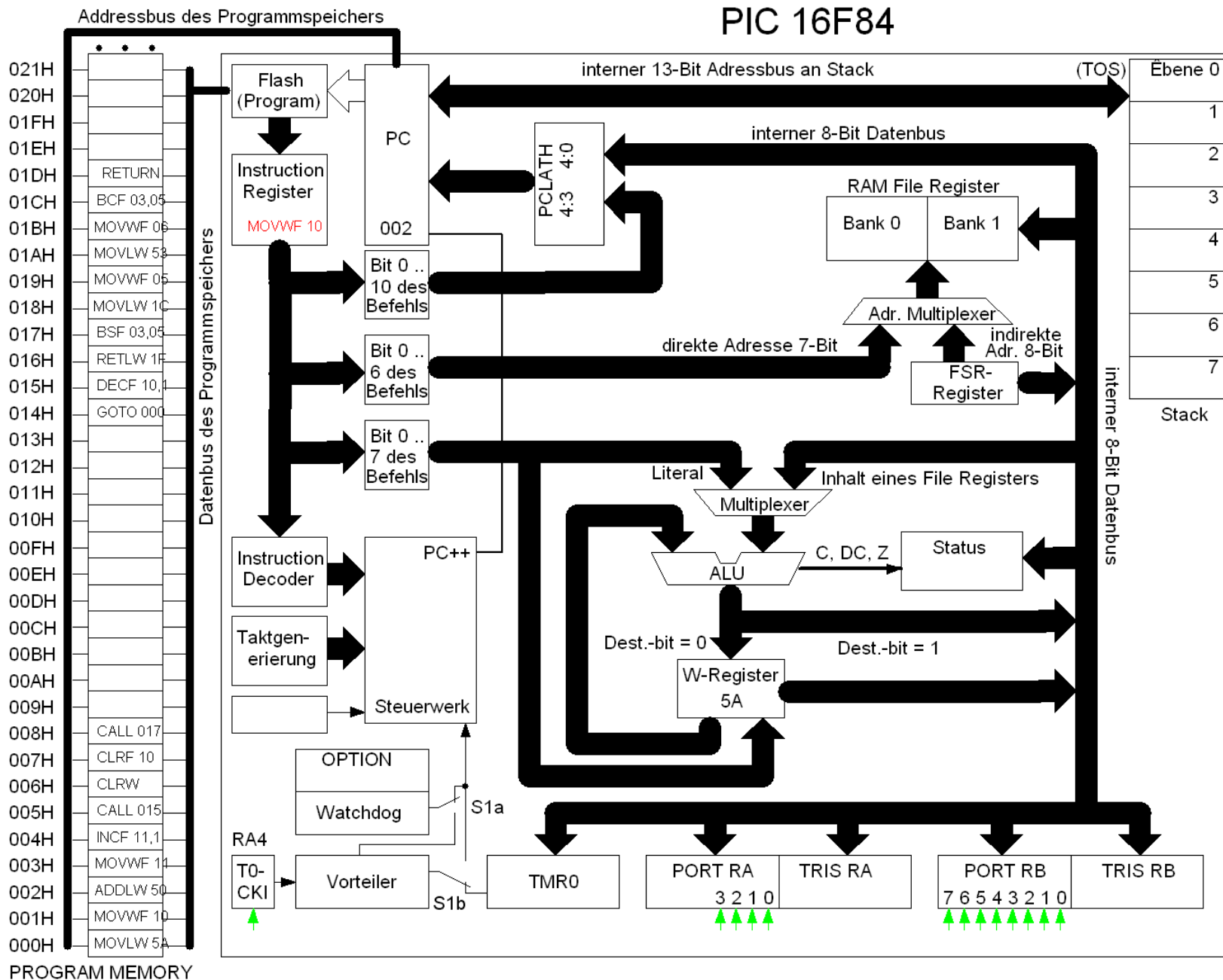
Interner Zustand:

Takt 2/4**Programmzähler wird inkrementiert**

RAM File Register

		1FH			
		1EH			
		1DH			
		1CH			
		1BH			
		1AH			
		19H			
		18H			
		17H			
		16H			
		15H			
		14H			
		13H			
		12H			
		11H			
		10H			
		0FH			
		0EH			
		0DH			
		0CH			
INTCON	00	0BH	00	INTCON	
PCLATH	00	0AH	00	PCLATH	
		09H			
		08H	00		
		07H			
PORTB		06H	FF	TRIS B	
PORTA		05H	FF	TRIS A	
FSR		04H		FSR	
STATUS	18	03H	18	STATUS	
PCL	02	02H	00	PCL	
TMR0		01H	FF	OPTION	
IND		00H		IND	

Bank 0 Bank 1



RAM File Register				
		1FH		
		1EH		
		1DH		
		1CH		
		1BH		
		1AH		
		19H		
		18H		
		17H		
		16H		
		15H		
		14H		
		13H		
		12H		
		11H		
		10H		
		0FH		
		0EH		
		0DH		
		0CH		
INTCON	00	0BH	00	INTCON
PCLATH	00	0AH	00	PCLATH
		09H		
		08H	00	
		07H		
PORTB		06H	FF	TRIS B
PORTA		05H	FF	TRIS A
FSR		04H		FSR
STATUS	18	03H	18	STATUS
PCL	02	02H	00	PCL
TMR0		01H	FF	OPTION
IND		00H		IND
Bank 0		Bank 1		

Interner Zustand:

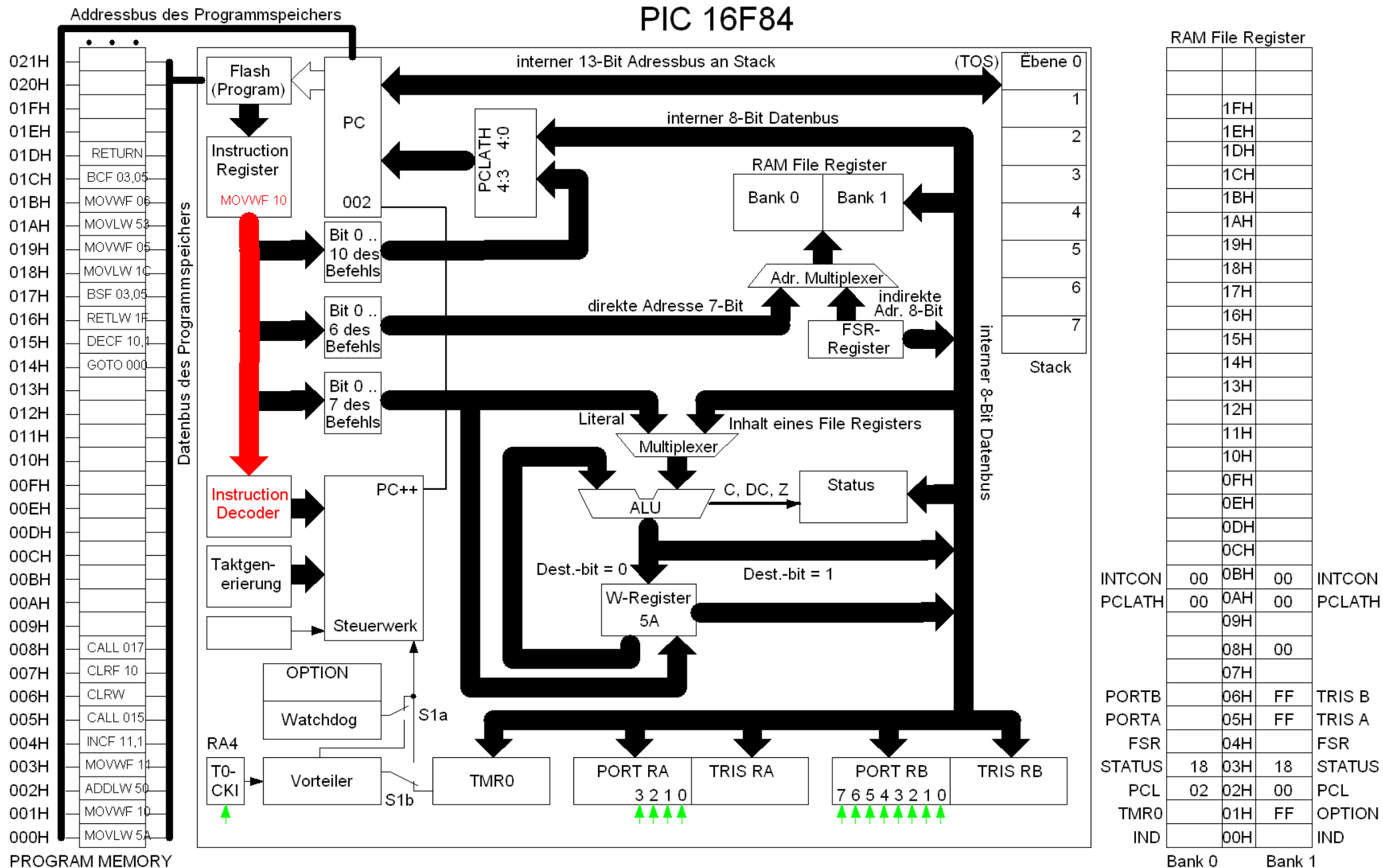
1

Der Befehl aus Adresse 000 ist abgearbeitet
In der Pipeline steht Befehl aus Adresse 001
Der Programmzähler zeigt auf Adresse 002

Funktionsweise des PIC 16F84

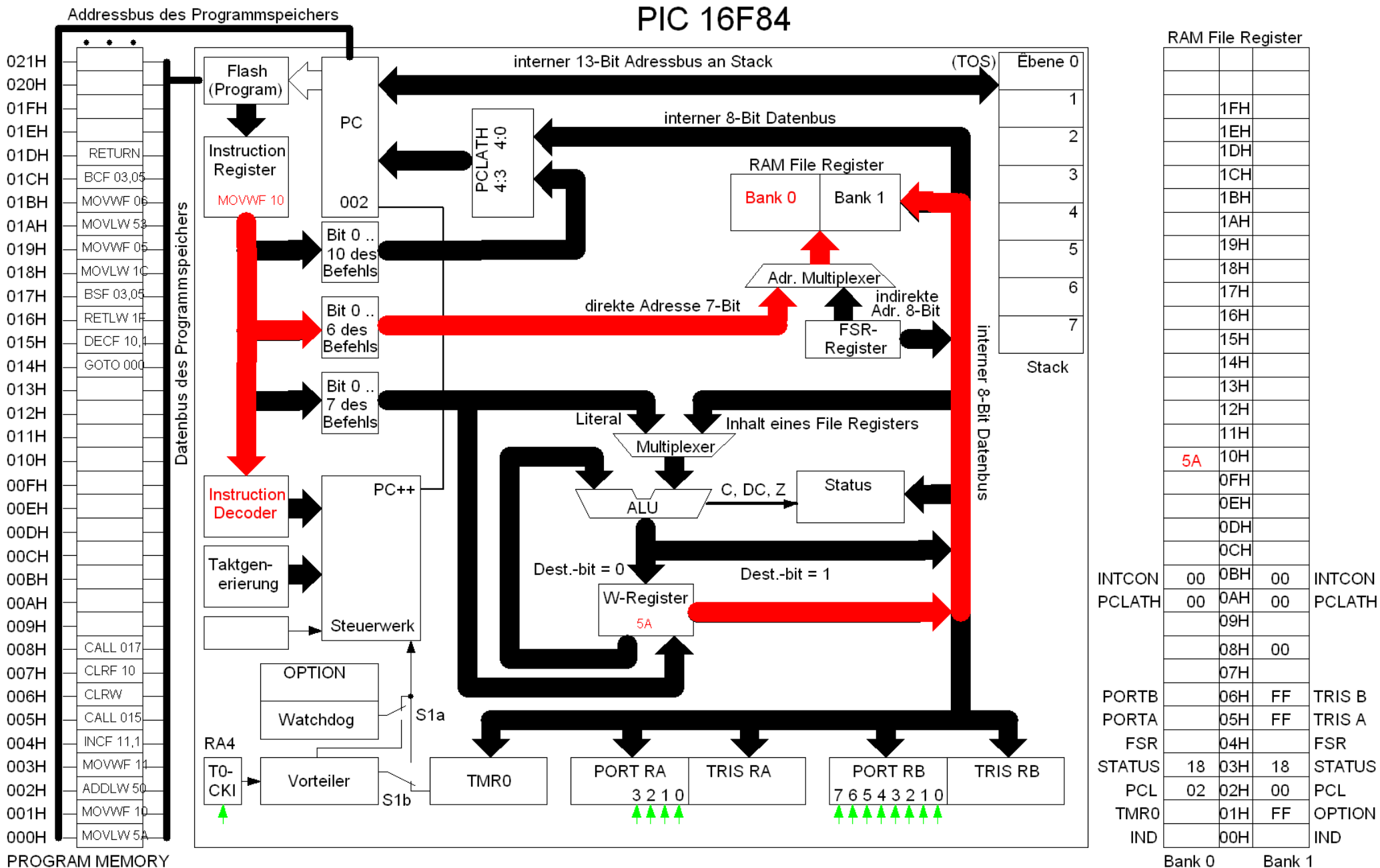
Als nächstes soll der Wert aus dem W-Register an die Adresse 0x10 im Datenspeicher (File Register) kopiert werden.

MOVWF 10H

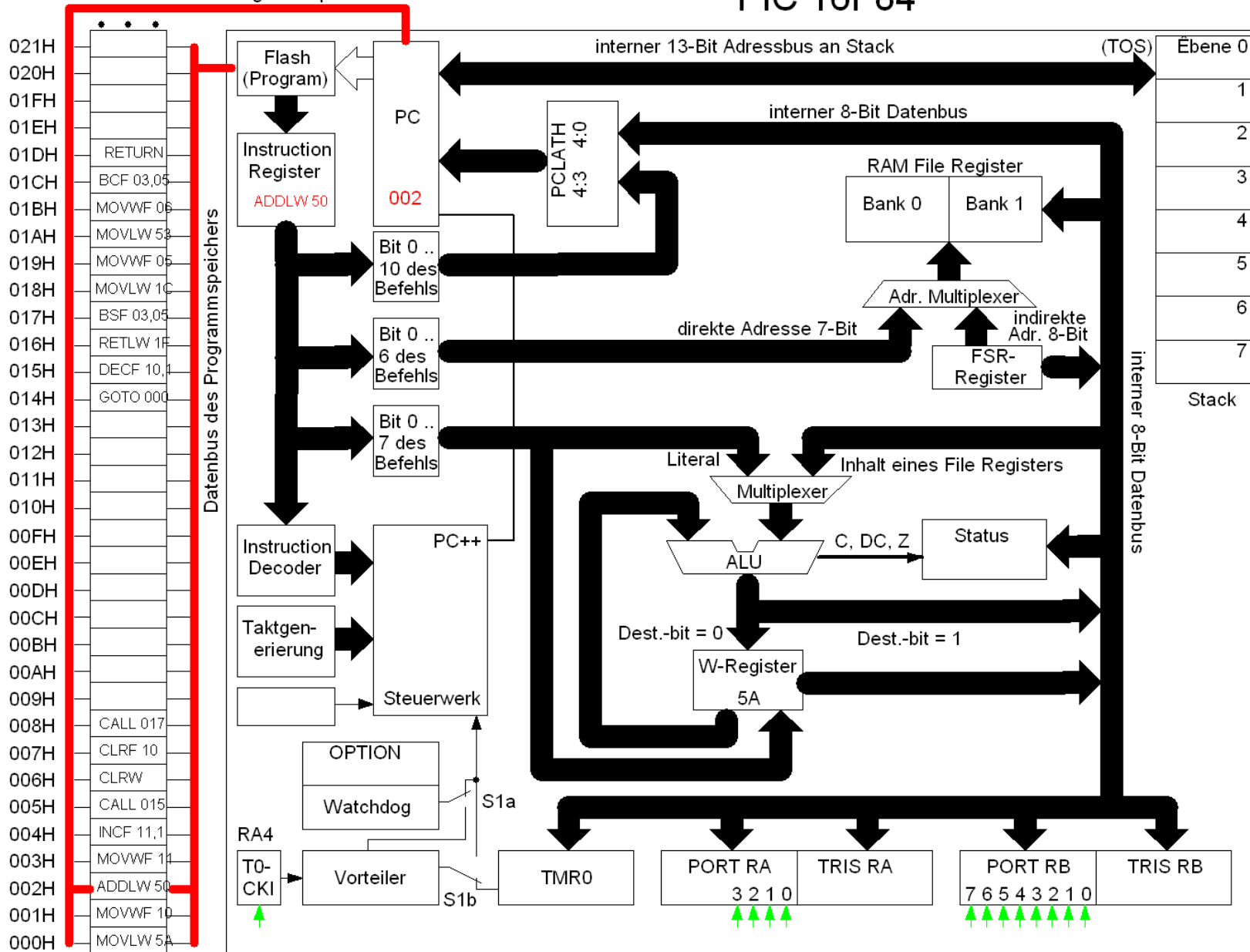


Interner Zustand:

Takt 3/1**Befehl kommt aus Pipeline in den Befehlsdeko**



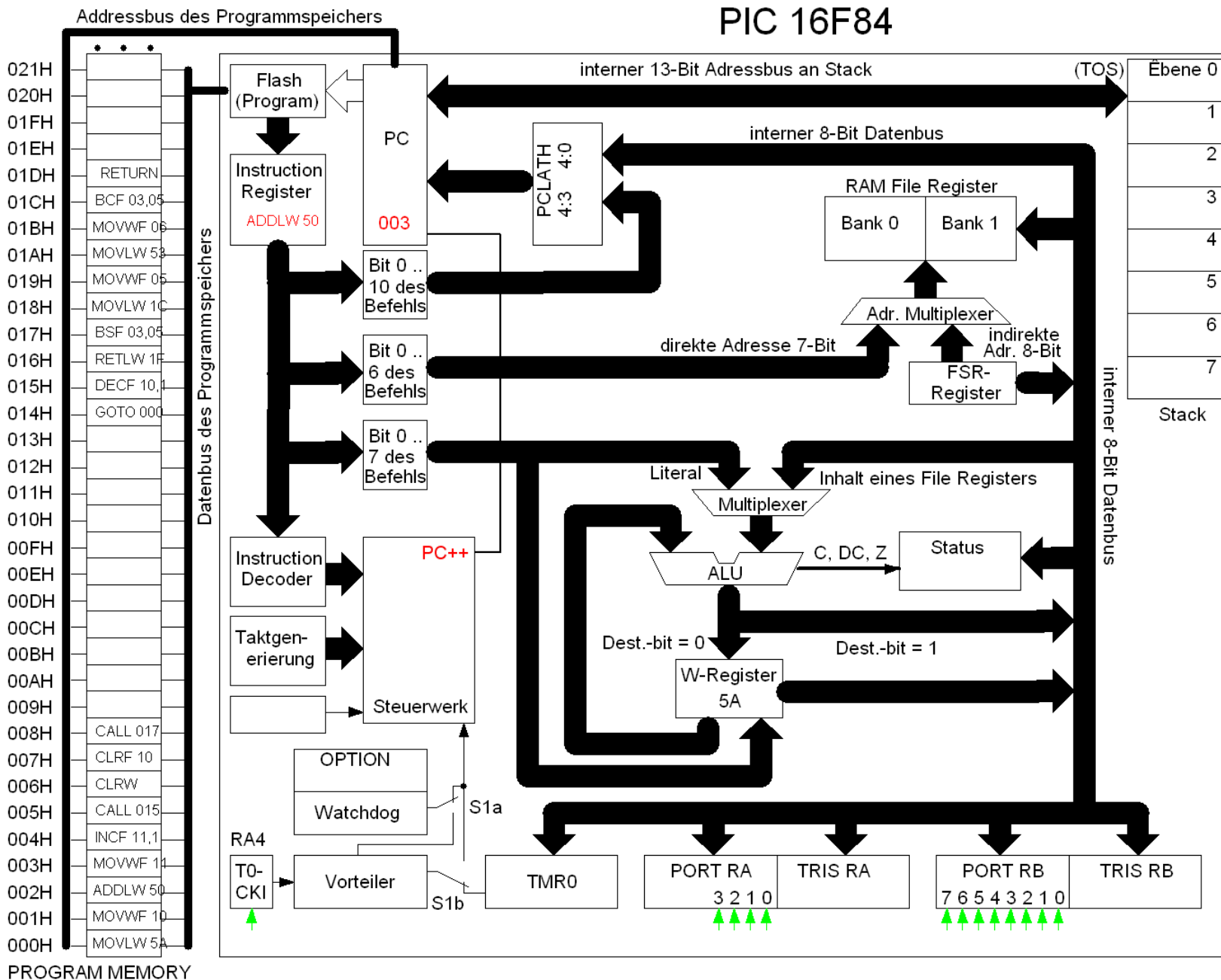
PROGRAM MEMORY



Takt 3/3

Befehl aus Adresse 002 kommt in die Pipeline

RAM File Register				
		1FH		
		1EH		
		1DH		
		1CH		
		1BH		
		1AH		
		19H		
		18H		
		17H		
		16H		
		15H		
		14H		
		13H		
		12H		
		11H		
	5A	10H		
		0FH		
		0EH		
		0DH		
		0CH		
INTCON	00	0BH	00	INTCON
PCLATH	00	0AH	00	PCLATH
		09H		
		08H	00	
		07H		
PORTB		06H	FF	TRIS B
PORTA		05H	FF	TRIS A
FSR		04H		FSR
STATUS	18	03H	18	STATUS
PCL	02	02H	00	PCL
TMR0		01H	FF	OPTION
IND		00H		IND



Interner Zustand:

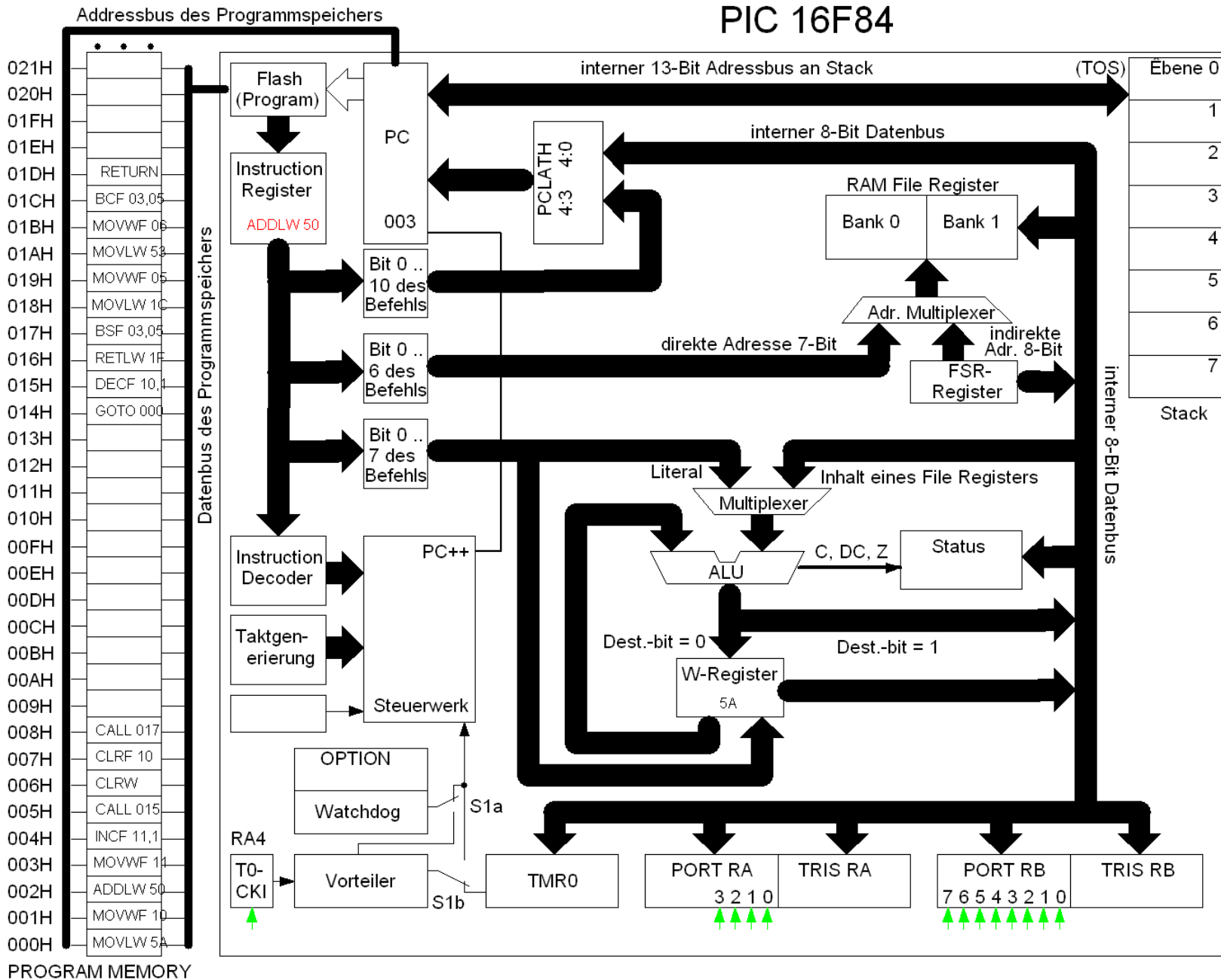
Takt 3/4

Programmzähler wird auf 003 erhöht

RAM File Register

	1FH		
	1EH		
	1DH		
	1CH		
	1BH		
	1AH		
	19H		
	18H		
	17H		
	16H		
	15H		
	14H		
	13H		
	12H		
	11H		
	10H		
	5A		
	0FH		
	0EH		
	0DH		
	0CH		
INTCON	00	0BH	00
PCLATH	00	0AH	00
		09H	
		08H	00
		07H	
PORTB		06H	FF
PORTA		05H	FF
FSR		04H	
STATUS	18	03H	18
PCL	03	02H	00
TMR0		01H	FF
IND		00H	

Bank 0 Bank 1



Interner Zustand:

-

Der Befehl aus Adresse 001 ist abgearbeitet
In der Pipeline steht Befehl aus Adresse 002
Der Programmzähler zeigt auf Adresse 003

RAM File Register

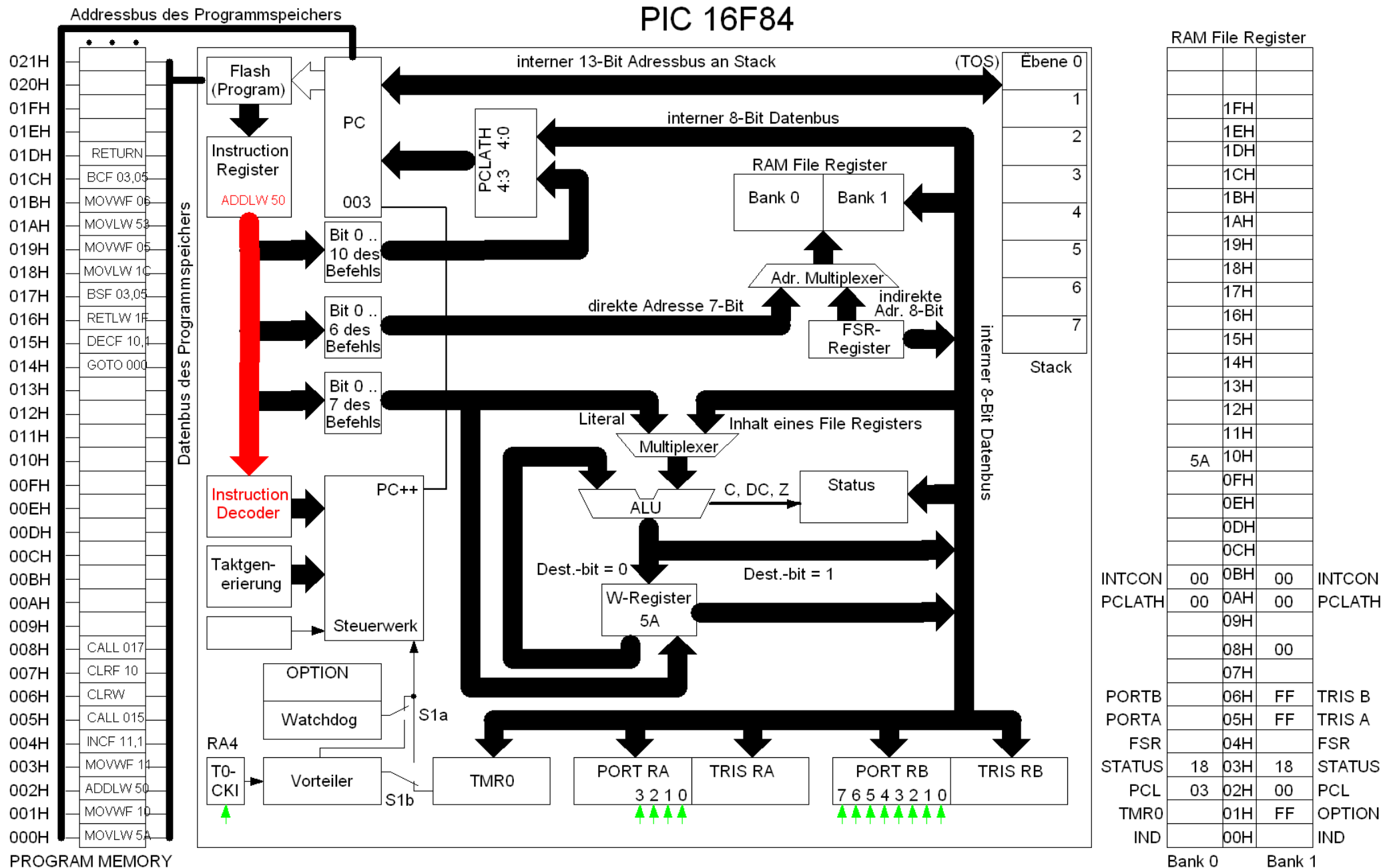
		1FH		
		1EH		
		1DH		
		1CH		
		1BH		
		1AH		
		19H		
		18H		
		17H		
		16H		
		15H		
		14H		
		13H		
		12H		
		11H		
	5A	10H		
		0FH		
		0EH		
		0DH		
		0CH		
INTCON	00	0BH	00	INTCON
PCLATH	00	0AH	00	PCLATH
		09H		
		08H	00	
		07H		
PORTB		06H	FF	TRIS B
PORTA		05H	FF	TRIS A
FSR		04H		FSR
STATUS	18	03H	18	STATUS
PCL	03	02H	00	PCL
TMR0		01H	FF	OPTION
IND		00H		IND
Bank 0		Bank 1		



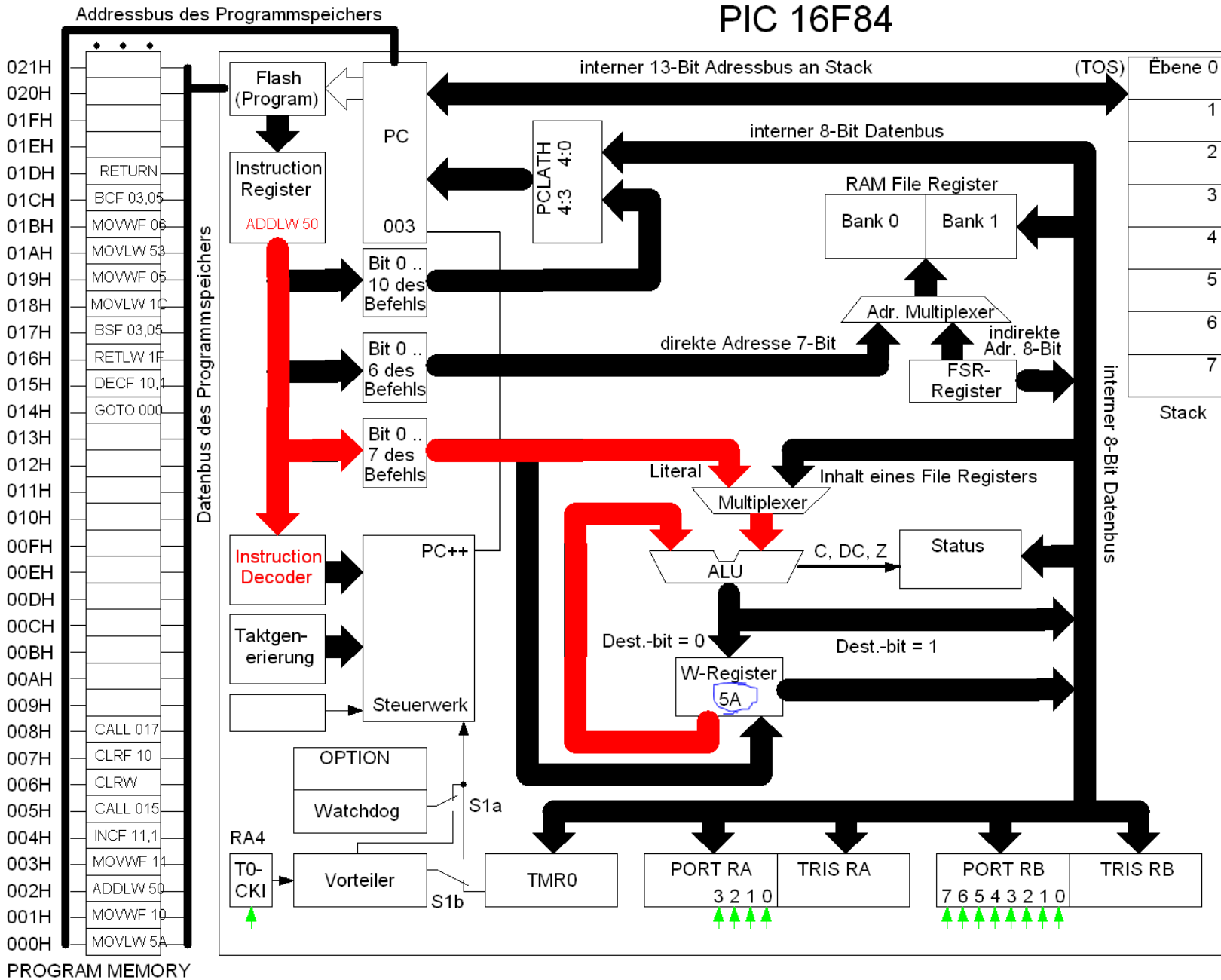
Funktionsweise des PIC 16F84

Nun wird zum Inhalt des W-Registers (0x5A) das Literal 0x50 addiert werden.

Hier wird deutlich, wie sich der Execute-Zyklus (das Ablegen des Ergebnisses ins W-Register) und der Fetch-Zyklus des nächsten Befehls überlappen.



PIC 16F84



		1FH	
		1EH	
		1DH	
		1CH	
		1BH	
		1AH	
		19H	
		18H	
		17H	
		16H	
		15H	
		14H	
		13H	
		12H	
		11H	
	5A	10H	
		0FH	
		0EH	
		0DH	
		0CH	
INTCON	00	0BH	00
PCLATH	00	0AH	00
		09H	
		08H	00
		07H	
PORTB		06H	FF
PORTA		05H	FF
FSR		04H	
STATUS	18	03H	18
PCL	03	02H	00
TMR0		01H	FF
IND		00H	

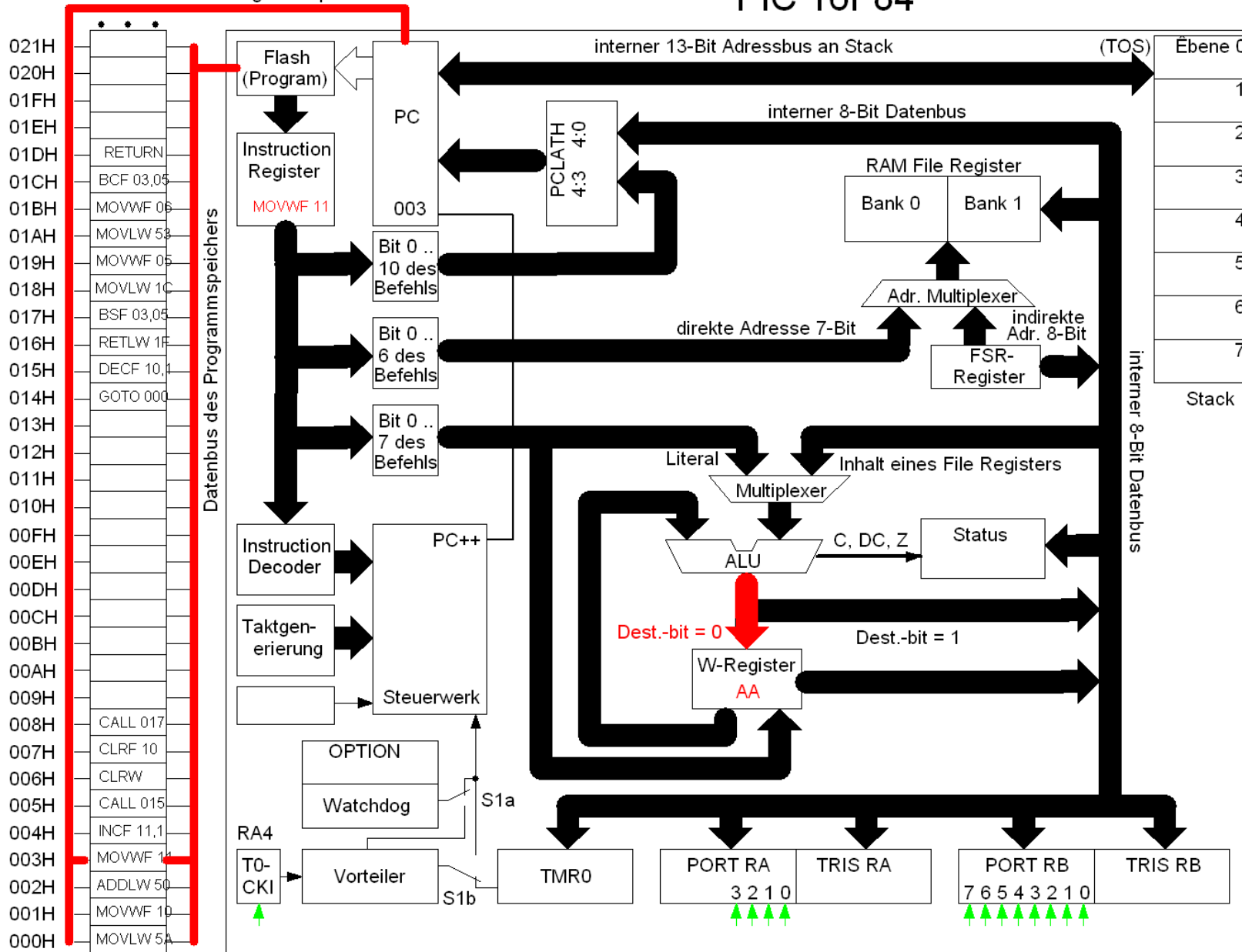
Bank 0 Bank 1

Interner Zustand:

Takt 4/2

Befehl kommt aus Pipeline in den Befehlsdeko

PROGRAM MEMORY



Takt 4/3

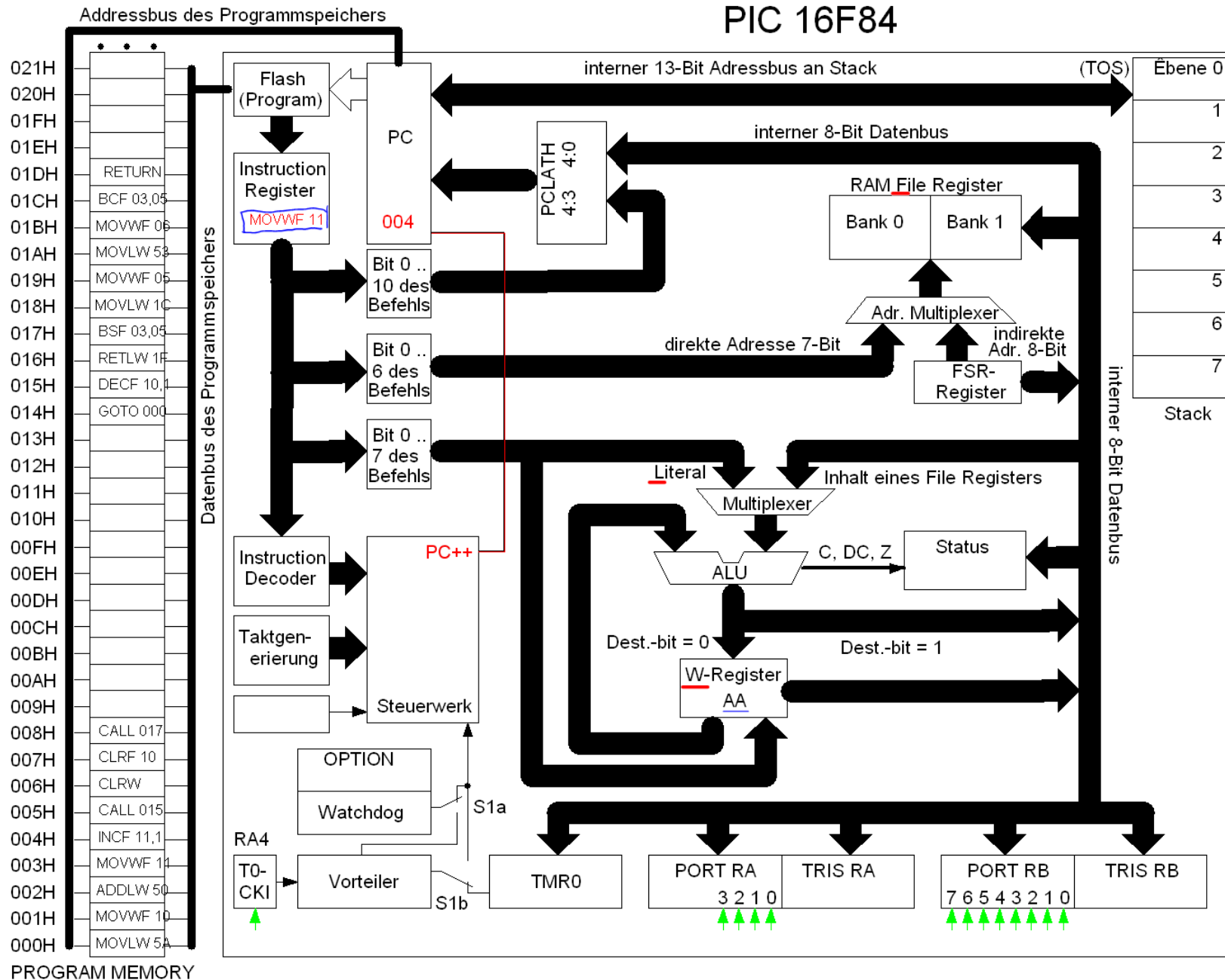
Das Ergebnis der Addition kommt ins W-Register
Gleichzeitig kann der nächste Befehl aus dem
Programmspeicher geladen werden

INTCON	00	0BH	00	INTCON
PCLATH	00	0AH	00	PCLATH
		09H		
		08H	00	
		07H		
PORTB		06H	FF	TRIS B
PORTA		05H	FF	TRIS A
FSR		04H		FSR
STATUS	18	03H	18	STATUS
PCL	03	02H	00	PCL
TMR0		01H	FF	OPTION
IND		00H		IND

Bank 0 Bank 1

Bank 0	Bank 1
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99

PIC 16F84



Interner Zustand:

Takt 4/4

Programmzähler wird auf 004 erhöht

RAM File Register				
		1FH		
		1EH		
		1DH		
		1CH		
		1BH		
		1AH		
		19H		
		18H		
		17H		
		16H		
		15H		
		14H		
		13H		
		12H		
	AA	11H		
	5A	10H		
		0FH		
		0EH		
		0DH		
		0CH		
INTCON	00	0BH	00	INTCON
PCLATH	00	0AH	00	PCLATH
		09H		
		08H	00	
		07H		
PORTB		06H	FF	TRIS B
PORTA		05H	FF	TRIS A
FSR		04H		FSR
STATUS	18	03H	18	STATUS
PCL	04	02H	00	PCL
TMR0		01H	FF	OPTION
IND		00H		IND

Funktionsweise des PIC 16F84

Vorläufiges ENDE