Microchip PIC16F684 Microcontroller

14-Pin Flash-Based,
8-Bit CMOS Microcontroller with nanoWatt Technology

PIC16F684

Summary

Features

High-Performance RISC CPU:

- Only 35 instructions to learn:
  - All single-cycle instructions except branches
- Operating speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

Special Microcontroller Features:

- Precision Internal Oscillator:
  - Factory calibrated to ±1%
  - Software selectable frequency range of 8 MHz to 31 kHz
  - Software tunable
  - Two-speed Start-up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-saving Sleep mode
- Wide operating voltage range (2.0V-5.5V)
- Industrial and Extended Temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Detect (BOD) with software control option
- Enhanced low-current Watchdog Timer (WDT) with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with pull-up/input pin
- Programmable code protection
- High Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM retention: > 40 years

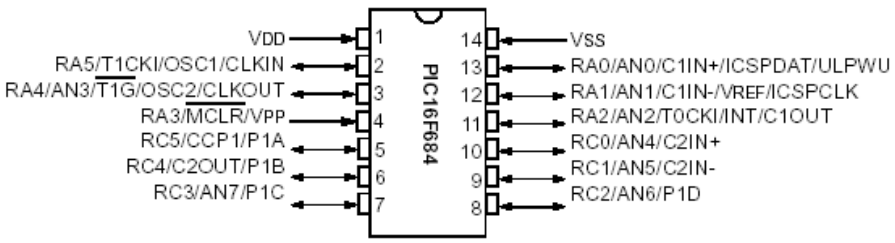
Low-Power Features:

- Standby Current:
  - 1 nA @ 2.0V, typical
- Operating Current:
  - 8.5μA @ 32 kHz, 2.0V, typical
  - 100μA @ 1 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1μA @ 2.0V, typical

Peripheral Features:

- 12 I/O pins with individual direction control:
  - High current source/sink for direct LED drive
  - Interrupt-on-pin change
  - Individually programmable weak pull-ups
  - Ultra Low-power Wake-up (ULPWU)
  - Analog comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (CV_{REF}) module (% of V_{DD})
  - Comparator inputs and outputs externally accessible
- A/D Converter:
  - 10-bit resolution and 8 channels
  - Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Gate Input mode
  - Option to use OSC1 and OSC2 in LP mode as Timer1 oscillator if INTOSC mode selected
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Enhanced Capture, Compare, PWM module:
  - 16-bit Capture, max resolution 12.5 ns
  - Compare, max resolution 200 ns
  - 10-bit PWM with 1, 2 or 4 output channels, programmable “dead time”, max frequency 20 kHz
- In-Circuit Serial Programming™ (ICSP™) via two pins

14-pin PDIP, SOIC, TSSOP

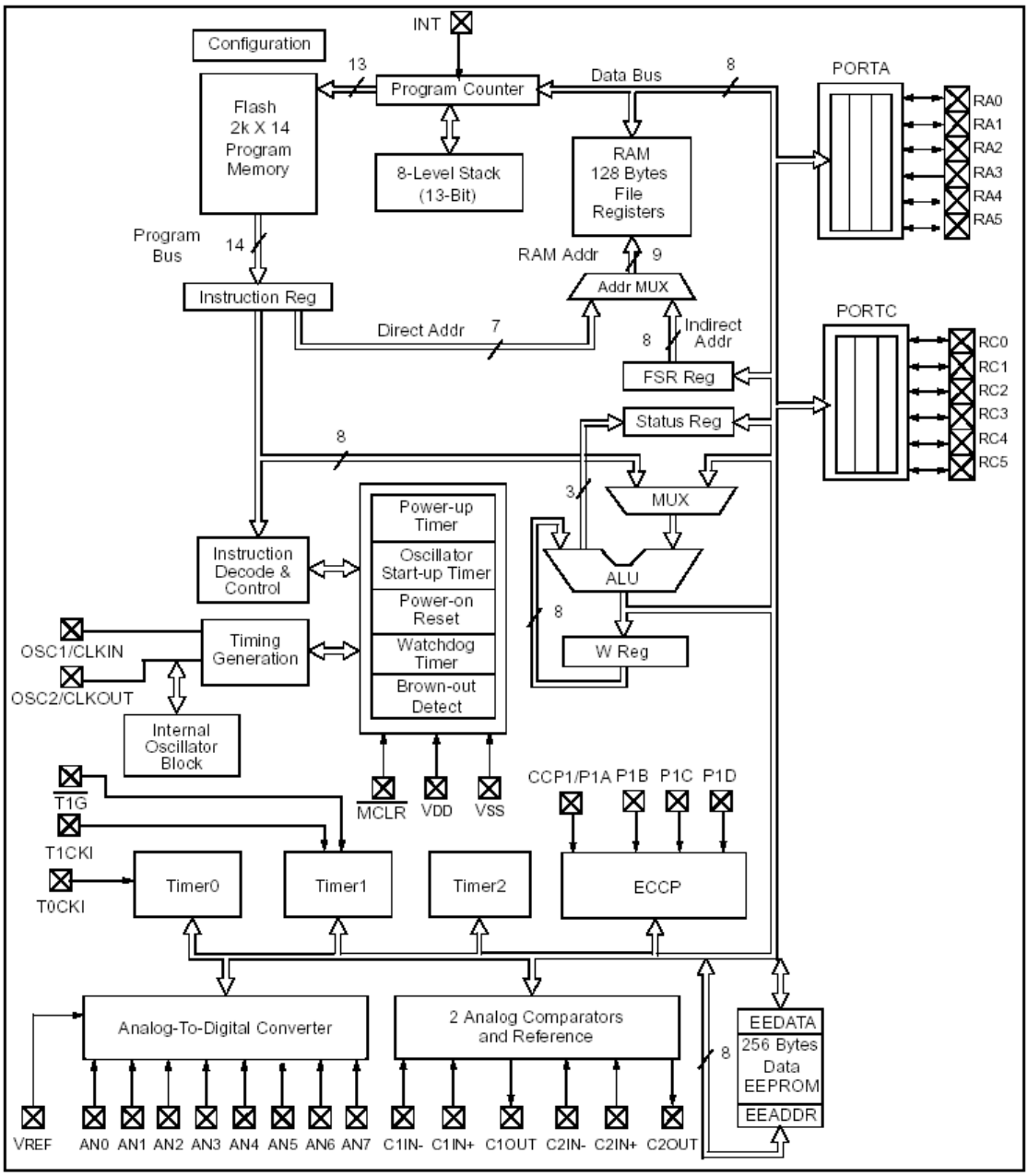


DEVICE OVERVIEW

This document contains device specific information for the PIC16F684. Additional information may be found in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site.

The reference manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.
The PIC16F684 is covered by this data sheet.
It is available in 14-pin PDIP, SOIC and TSSOP packages. Figure 1-1 shows a block diagram of the PIC16F684 device. Table 1-1 shows the pinout description.

FIGURE 1-1: PIC16F684 BLOCK DIAGRAM



PIC16F684 INSTRUCTION SET

Mnemonic,	Operands	
BYTE-ORIENTED FILE REGISTER OPERATIONS		
ADDWF	f, d	Add W and f
ANDWF	f, d	AND W with f
CLRF	f –	Clear f
CLRW	f, d	Clear W
COMF	f, d	Complement f
DECF	f, d	Decrement f
DECFSZ	f, d	Decrement f, Skip if 0
INCF	f, d	Increment f
INCFSZ	f, d	Increment f, Skip if 0
IORWF	f, d	Inclusive OR W with f
MOVF	f –	Move f
MOVWF	f, d	Move W to f
NOP	f, d	No Operation
RLF	f, d	Rotate Left f through Carry
RRF	f, d	Rotate Right f through Carry
SUBWF	f, d	Subtract W from f
SWAPF	f, d	Swap nibbles in f
XORWF	f, d	Exclusive OR W with f
BIT-ORIENTED FILE REGISTER OPERATIONS		
BCF	f, b	Bit Clear f
BSF	f, b	Bit Set f
BTFSC	f, b	Bit Test f, Skip if Clear
BTFSS	f, b	Bit Test f, Skip if Set
LITERAL AND CONTROL OPERATIONS		
ADDLW	k	Add literal and W
ANDLW	k	AND literal with W
CALL	k	Call Subroutine
CLRWDT	–	Clear Watchdog Timer
GOTO	k	Go to address
IORLW	k	Inclusive OR literal with W
MOVLW	K	Move literal to W
RETFIE	–	Return from interrupt
RETLW	k	Return with literal in W
RETURN	–	Return from Subroutine
SLEEP	–	Go into Standby mode
SUBLW	k	Subtract W from literal
XORLW	k	Exclusive OR literal with W

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F684

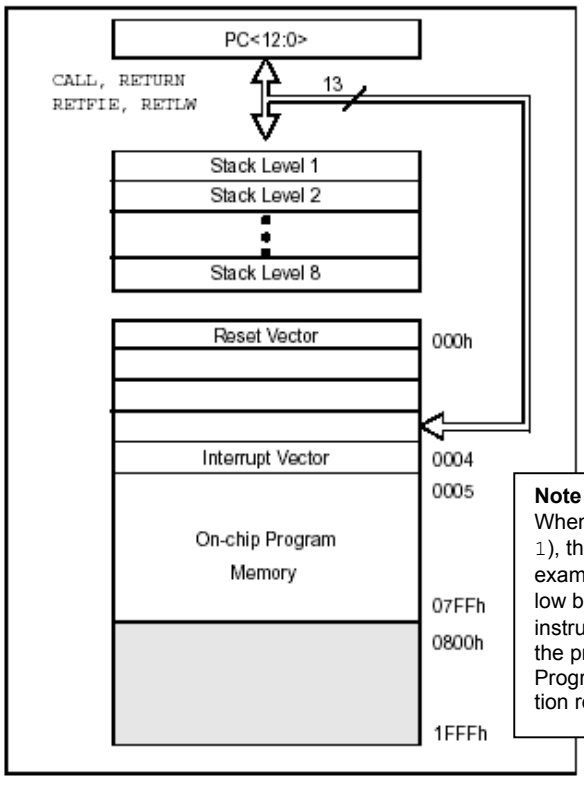


FIGURE 2-2: DATA MEMORY MAP OF THE PIC16F684

Note (Instruction set):
When an I/O register is modified as a function of itself (e.g., `MOVF GPIO, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'. **2:** If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module. **3:** If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

TABLE 1-1: PIC16F684 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/ICSPDAT/ULPWU	RA0	TTL	CMOS	PORTA I/O w/programmable pull-up and interrupt-on-change
	AN0	AN	—	A/D Channel 0 input
	C1IN+	AN	—	Comparator 1 input
	ICSPDAT	TTL	CMOS	Serial Programming Data I/O
	ULPWU	AN	—	Ultra Low-power Wake-up input
RA1/AN1/C1IN-/VREF/ICSPCLK	RA1	TTL	CMOS	PORTA I/O w/programmable pull-up and interrupt-on-change
	AN1	AN	—	A/D Channel 1 input
	C1IN-	AN	—	Comparator 1 input
	VREF	AN	—	External Voltage Reference for A/D
	ICSPCLK	ST	—	Serial Programming Clock
RA2/AN2/T0CKI/INT/C1OUT	RA2	ST	CMOS	PORTA I/O w/programmable pull-up and interrupt-on-change
	AN2	AN	—	A/D Channel 2 input
	T0CKI	ST	—	Timer0 clock input
	INT	ST	—	External interrupt
	C1OUT	—	CMOS	Comparator 1 output
RA3/$\overline{MCLR}$/VPP	RA3	TTL	—	PORTA input with interrupt-on-change
	$\overline{MCLR}$	ST	—	Master Clear w/internal pull-up
	VPP	HV	—	Programming voltage
RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT	RA4	TTL	CMOS	PORTA I/O w/programmable pull-up and interrupt-on-change
	AN3	AN	—	A/D Channel 3 input
	$\overline{T1G}$	ST	—	Timer1 gate
	OSC2	—	XTAL	Crystal/Resonator
	CLKOUT	—	CMOS	FOSC/4 output
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	PORTA I/O w/programmable pull-up and interrupt-on-change
	T1CKI	ST	—	Timer1 clock
	OSC1	XTAL	—	Crystal/Resonator
	CLKIN	ST	—	External clock input/RC oscillator connection
RC0/AN4/C2IN+	RC0	TTL	CMOS	PORTC I/O
	AN4	AN	—	A/D Channel 4 input
	C2IN+	AN	—	Comparator 2 input
RC1/AN5/C2IN-	RC1	TTL	CMOS	PORTC I/O
	AN5	AN	—	A/D Channel 5 input
	C2IN-	AN	—	Comparator 2 input
RC2/AN6/P1D	RC2	TTL	CMOS	PORTC I/O
	AN6	AN	—	A/D Channel 6 input
	P1D	—	CMOS	PWM output
RC3/AN7/P1C	RC3	TTL	CMOS	PORTC I/O
	AN7	AN	—	A/D Channel 7 input
	P1C	—	CMOS	PWM output
RC4/C2OUT/P1B	RC4	TTL	CMOS	PORTC I/O
	C2OUT	—	CMOS	Comparator 2 output
	P1B	—	CMOS	PWM output
RC5/CCP1/P1A	RC5	TTL	CMOS	PORTC I/O
	CCP1	ST	CMOS	Capture input/Compare output
	P1A	—	CMOS	PWM output
VSS	VSS	Power	—	Ground reference
VDD	VDD	Power	—	Positive supply

Legend: TTL = TTL input buffer, ST = Schmitt Trigger input buffer, AN = Analog input

2.0 MEMORY ORGANIZATION

2.1 Program Memory Organization

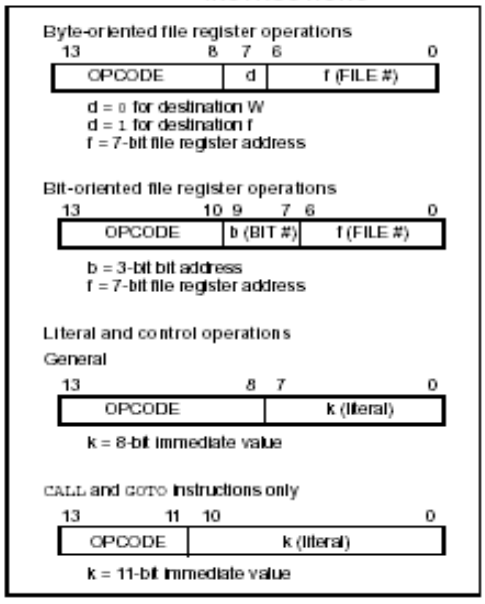
The PIC16F684 has a 13-bit program counter capable of addressing an 8k x 14 program memory space. Only the first 2k x 14 (0000h-07FFh) for the PIC16F684 is physically implemented. Accessing a location above these boundaries will cause a wrap around within the first 2k x 14 space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 2-1).

2.2 Data Memory Organization

The data memory (see Figure 2-2) is partitioned into two banks, which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). The Special Function Registers are located in the first 32 locations of each bank. Register locations 20h-7Fh in Bank 0 and A0h-BFh in Bank 1 are General Purpose Registers, implemented as static RAM. Register locations F0h-FFh in Bank 1 point to addresses 70h-7Fh in Bank0. All other RAM is unimplemented and returns '0' when read. RP0 (Status<5>) is the bank select bit.

- RP0 = 0: -> Bank 0 is selected
- RP0 = 1: -> Bank 1 is selected

FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS



13.0 INSTRUCTION SET SUMMARY

The PIC16F684 instruction set is highly orthogonal and is comprised of three basic categories:

• **Byte-oriented** operations
• **Bit-oriented** operations
• **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 13-1, while the various opcode fields are summarized in Table 13-1. Table 13-2 lists the instructions recognized by the MPASM™ assembler. A complete description of each instruction is also available in the "*PICmicro® Mid- Range MCU Family Reference Manual*" (DS33023). For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction. The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value. One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 ∝s. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

13.1 READ-MODIFY-WRITE OPERATIONS

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register. For example, a CLRF GPIO instruction will read GPIO, clear all the data bits, then write the result back to GPIO. This example would have the unintended result of clearing the condition that set the GPIF flag.

TABLE 2-1: PIC16F684 SPECIAL REGISTERS SUMMARY BANK 0

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Page
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	17, 99
01h	TMR0	Timer0 Module's register								xxxx xxxx	45, 99
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	17, 99
03h	STATUS	IRP(1)	RP1(1)	RP0	$\overline{TO}$	PD	Z	DC	C	0001 1xxx	11, 99
04h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	17, 99
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	31, 99
06h	—	Unimplemented								—	—
07h	PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx xxxx	40, 99
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	17, 99
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	13, 99
0Ch	PIR1	EEIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	0000 0000	15, 99
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1								xxxx xxxx	49, 99
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1								xxxx xxxx	49, 99
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000	51, 99
11h	TMR2	Timer2 Module register								0000 0000	53, 99
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	53, 99
13h	CCPR1L	Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx	75, 99
14h	CCPR1H	Capture/Compare/PWM Register 1 High Byte								xxxx xxxx	75, 99
15h	CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	75, 99
16h	PWM1CON	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	85, 99
17h	ECCPAS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	86, 99
18h	WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	106, 99
19h	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	55, 99
1Ah	CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	59, 99
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRESH	Most Significant 8 bits of the left shifted A/D result or 2 bits of right shifted result								xxxx xxxx	65, 99
1Fh	ADCON0	ADFM	VCFG	—	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	00-0 0000	66, 99

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: IRP and RP1 bits are reserved, always maintain these bits clear.

TABLE 2-2: PIC16F684 SPECIAL FUNCTION REGISTERS SUMMARY BANK 1

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Page
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	17, 99
81h	OPTION_REG	$\overline{RAPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	12, 99
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	17, 99
83h	STATUS	IRP(1)	RP1(1)	RP0	$\overline{TO}$	PD	Z	DC	C	0001 1xxx	11, 99
84h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	17, 99
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	32, 99
86h	—	Unimplemented								—	—
87h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	43, 99
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	17, 99
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	13, 99
8Ch	PIE1	EEIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000	14, 99
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	ULPWUE	SBODEN	—	—	$\overline{POR}$	$\overline{BOD}$	--01 --qq	16, 99
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS(2)	HTS	LTS	SCS	-110 x000	29, 99
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	23, 99
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	65, 99
92h	PR2	Timer2 Module Period Register								1111 1111	53, 99
93h	—	Unimplemented								—	—
94h	—	Unimplemented								—	—
95h	WPUA(3)	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	32, 100
96h	IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	33, 100
97h	—	Unimplemented								—	—
98h	—	Unimplemented								—	—
99h	VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	62, 100
9Ah	EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	71, 100
9Bh	EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	71, 100
9Ch	EECON1	—	—	—	—	WRERR	WREN	WR	RD	---- x000	72, 100
9Dh	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	72, 100
9Eh	ADRESL	Least Significant 2 bits of the left shifted result or 8 bits of the right shifted result								xxxx xxxx	65, 100
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	66, 100

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: IRP and RP1 bits are reserved, always maintain these bits clear.

2: OSTS bit OSCCON <3> reset to '0' with Dual Speed Start-up and LP, HS or XT selected as the oscillator.

3: RA3 pull-up is enabled when MCLRE is '1' in the Configuration Word register.

Näidisülesanne

```
;************************************************************
;**  SEE ON ESIMENE PROGRAMM                                **
;**     Protsessor: PIC16F684                               **
;**           Autor: Jüri Puhang                            **
;**         Kuupäev: 19.11.2005                             **
;**     Programmeerib pordid ja paneb dioodid "põlema"      **
;**         loendusrežiimis                                 **
;************************************************************
      list      p=PIC16F684
      __config 3FF4h    ; Sisemine generaator ja gen. sagedust
      org   0x000       ;   ei väljastata.
      bcf   03h,6       ; STATUS<6:5>:=0
      bcf   03h,5       ; Bank 0 (STATUS<5>=0 -> Bank 0)
      clrf  05h         ; PortA:=0
      clrf  07h         ; PortC:=0
      clrf  0Bh         ;
      movlw 07h         ; Komparaatorit ei  kasutata. CM2:CM0:=111
      movwf 19h         ; CMCON0:=Compar Off
      bsf   03h,5       ; Bank 1
      clrf  11h         ; Analoog sisendit   ei kasutata ANSEL=91H:=00h
                        ; RA5,RA4;RA3;RA2;RA1;RA0
      movlw 09h         ;  0   0   1   0   0   1
                        ; RA<5:4> ja RA<2:1> on väljundid
      movwf 05h         ; TRISA=85h:=09h
      clrf  07h         ; RC<5:1> väljundid
                        ; TRISC=87h:=00h
      clrf  19h         ; VRcon off
      bcf   03h,5       ; Bank 0
      movlw 00h         ; Kui movlw XXh ja LOOP sees, “põleb”
                        ;   üks või kaks dioode
            ;-------------------------------------
            ;    D0  D1  D2  D3  D4  D5  D6  D7
            ;RA5  0   1           1   0
            ;RA4  1   0   1   0
            ;RA2          0   1   0   1   1   0
            ;RA1                          0   1
            ;-------------------------------------
      movwf 05h         ; PORTA=05h:=LED-id
;LOOP:goto  LOOP
;************************************************************
; KÕIK DIOODID TSÜKLILISELT PÕLEMA
;************************************************************
A:    andlw 30h   ; Eraldada RA<4,5>
      movwf 05h   ; PORTA=05h:=LED-id(0)
      movlw 02h   ; +2 konstant
      call  VIIVIS
      addwf 05h,1 ; Liita +2 ja salvestada
                  ;    PORTA=05h:=LED-id(1)
      call  VIIVIS
      addwf 05h,1 ; Liita +2 (LED-id = 2)
      call  VIIVIS
      addwf 05h,1 ; Liita +2 (LED-id = 3)
      call  VIIVIS

      movf  05h,0 ; W:= PORTA
      addlw 10h   ; PORT<4,5>:=1,2,3,0
      goto  A
VIIVIS:decfsz     20h,1 ; viivis 256 x
      goto  VIIVIS
      decfsz      21h,1 ; tsükkel tsüklis
      goto  VIIVIS
      return
      END
```