Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

PIC16(L)F720/721

20-Pin Flash Microcontrollers

Devices Included In This Data Sheet:

- PIC16F720
- PIC16LF720
- PIC16F721
- PIC16LF721

High-Performance RISC CPU:

- Only 35 Instructions to Learn:
  - All single-cycle instructions except branches
- Operating Speed:
  - DC – 16 MHz oscillator/clock input
  - DC – 250 ns instruction cycle
- Up to 4K x 14 Words of Flash Program Memory
- Up to 256 bytes of Data Memory (RAM)
- Interrupt Capability
- 8-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes
- Processor Self-Write/Read access to Program Memory

Memory

- High-Endurance Flash Data Memory
  - 128B of nonvolatile data storage
  - 100K erase/write cycles

Special Microcontroller Features:

- Precision Internal Oscillator:
  - 16 MHz or 500 kHz operation
  - Factory calibrated to ±1%, typical
  - Software tunable
  - Software selectable ÷1, ÷2, ÷4 or ÷8 divider
- Power-Saving Sleep mode
- Industrial and Extended Temperature Range
- Power-on Reset (POR)
- Power-up Timer (PWRT)
- Brown-out Reset (BOR)
- Multiplexed Master Clear with Pull-up/Input Pin
- Programmable Code Protection
- In-Circuit Serial Programming™ (ICSP™) via Two Pins
- Wide Operating Voltage Range:
  - 1.8V to 5.5V (PIC16F720/721)
  - 1.8V to 3.6V (PIC16LF720/721)

Extreme Low-Power (XLP) Features:

- Sleep Current:
  - 40 nA @ 1.8V, typical
- Low-Power Watchdog Timer Current:
  - 500 nA @ 1.8V, typical

Peripheral Features:

- Up to 17 I/O Pins and One Input-only Pin:
  - High-current source/sink for direct LED drive
  - Interrupt-on-change pins
  - Individually programmable weak pull-ups
- A/D Converter:
  - 8-bit resolution
  - 12 channels
  - Selectable Voltage reference
- Timer0: 8-Bit Timer/Counter with 8-Bit Programmable Prescaler
- Enhanced Timer1
  - 16-bit timer/counter with prescaler
  - External Gate Input mode with toggle and Single Shot modes
  - Interrupt-on-gate completion
- Timer2: 8-Bit Timer/Counter with 8-Bit Period Register, Prescaler and Postscaler
- Capture, Compare, PWM module (CCP)
  - 16-bit Capture, max resolution 12.5 ns
  - 16-bit Compare, max resolution 250 ns
  - 10-bit PWM, max frequency 15 kHz
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART)
- Synchronous Serial Port (SSP)
  - SPI (Master/Slave)
  - I^2C (Slave) with Address Mask

PIC16(L)F72X Family Types

Device	Data Sheet Index	Program Memory Flash (words)	Data SRAM (bytes)	High-Endurance Flash Memory (bytes)	I/O's(2)	8-bit ADC (ch)	CapSense (ch)	Timers (8/16-bit)	AUSART	SSP (I^2C/SPI)	CCP	Debug(1)	XLP
PIC16(L)F707	(1)	8192	363	0	36	14	32	4/2	1	1	2	I	Y
PIC16(L)F720	(2)	2048	128	128	18	12	—	2/1	1	1	1	I	Y
PIC16(L)F721	(2)	4096	256	128	18	12	—	2/1	1	1	1	I	Y
PIC16(L)F722	(4)	2048	128	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F722A	(3)	2048	128	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F723	(4)	4096	192	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F723A	(3)	4096	192	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F724	(4)	4096	192	0	36	14	16	2/1	1	1	2	I	Y
PIC16(L)F726	(4)	8192	368	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F727	(4)	8192	368	0	36	14	16	2/1	1	1	2	I	Y

Note 1: I - Debugging, Integrated on Chip; H - Debugging, Requires Debug Header.

2: One pin is input-only.

Data Sheet Index: (Unshaded devices are described in this document.)

1: DS41418 PIC16(L)F707 Data Sheet, 40/44-Pin Flash, 8-bit Microcontrollers

2: DS41430 PIC16(L)F720/721 Data Sheet, 20-Pin Flash, 8-bit Microcontrollers

3: DS41417 PIC16(L)F722A/723A Data Sheet, 28-Pin Flash, 8-bit Microcontrollers

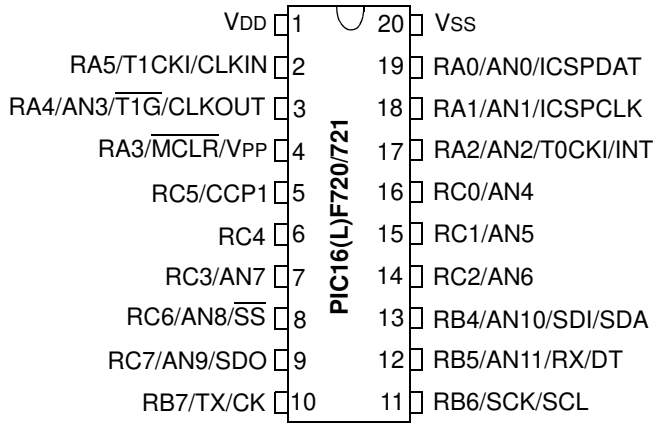
4: DS41341 PIC16(L)F72X Data Sheet, 28/40/44-Pin Flash, 8-bit Microcontrollers

Note: For other small form-factor package availability and marking information, please visit **http://www.microchip.com/packaging** or contact your local sales office.

PIN DIAGRAMS

FIGURE 1: 20-PIN DIAGRAM FOR PIC16(L)F720/721

PDIP, SOIC, SSOP



Pin Diagrams – 20-PIN DIAGRAM FOR PIC16(L)F720/721

QFN (4x4)

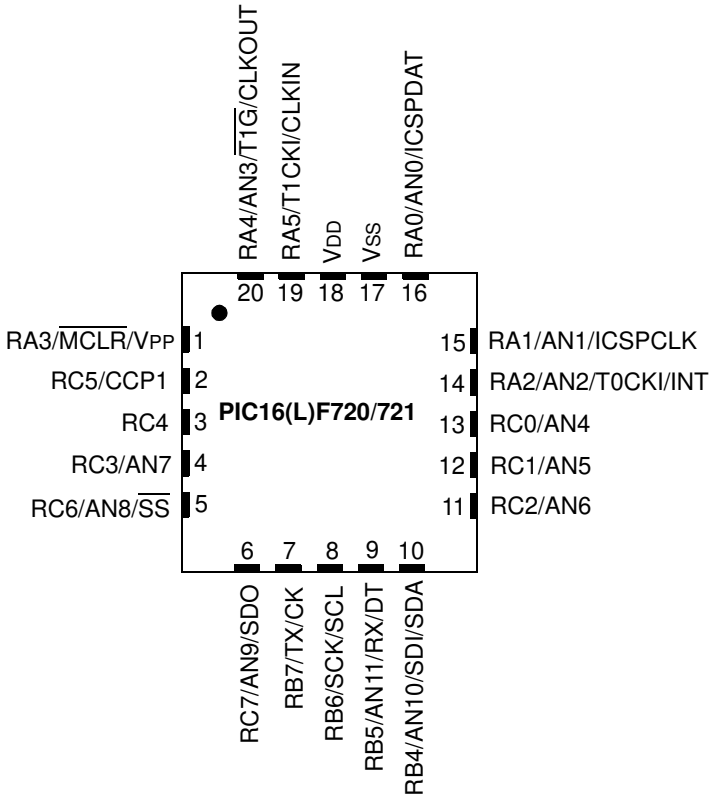


TABLE 1: 20-PIN ALLOCATION TABLE (PIC16(L)F720/721)

I/O	20-Pin PDIP/SOIC/SSOP	20-Pin QFN	A/D	Timers	CCP	AUSART	SSP	Interrupt	Pull-up	Basic
RA0	19	16	AN0	—	—	—	—	IOC	Y	ICSPDAT
RA1	18	15	AN1	—	—	—	—	IOC	Y	ICSPCLK
RA2	17	14	AN2	T0CKI	—	—	—	INT/IOC	—	—
RA3	4	1	—	—	—	—	—	IOC	Y	$\overline{MCLR}$/VPP
RA4	3	20	AN3	$\overline{T1G}$	—	—	—	IOC	Y	CLKOUT
RA5	2	19	—	T1CKI	—	—	—	IOC	Y	CLKIN
RB4	13	10	AN10	—	—	—	SDI/SDA	IOC	Y	—
RB5	12	9	AN11	—	—	RX/DT	—	IOC	Y	—
RB6	11	8	—	—	—	—	SCK/SCL	IOC	Y	—
RB7	10	7	—	—	—	TX/CK	—	IOC	Y	—
RC0	16	13	AN4	—	—	—	—	—	—	—
RC1	15	12	AN5	—	—	—	—	—	—	—
RC2	14	11	AN6	—	—	—	—	—	—	—
RC3	7	4	AN7	—	—	—	—	—	—	—
RC4	6	3	—	—	—	—	—	—	—	—
RC5	5	2	—	—	CCP1	—	—	—	—	—
RC6	8	5	AN8	—	—	—	$\overline{SS}$	—	—	—
RC7	9	6	AN9	—	—	—	SDO	—	—	—
VDD	1	18	—	—	—	—	—	—	—	VDD
Vss	20	17	—	—	—	—	—	—	—	VSS

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Website at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Website; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

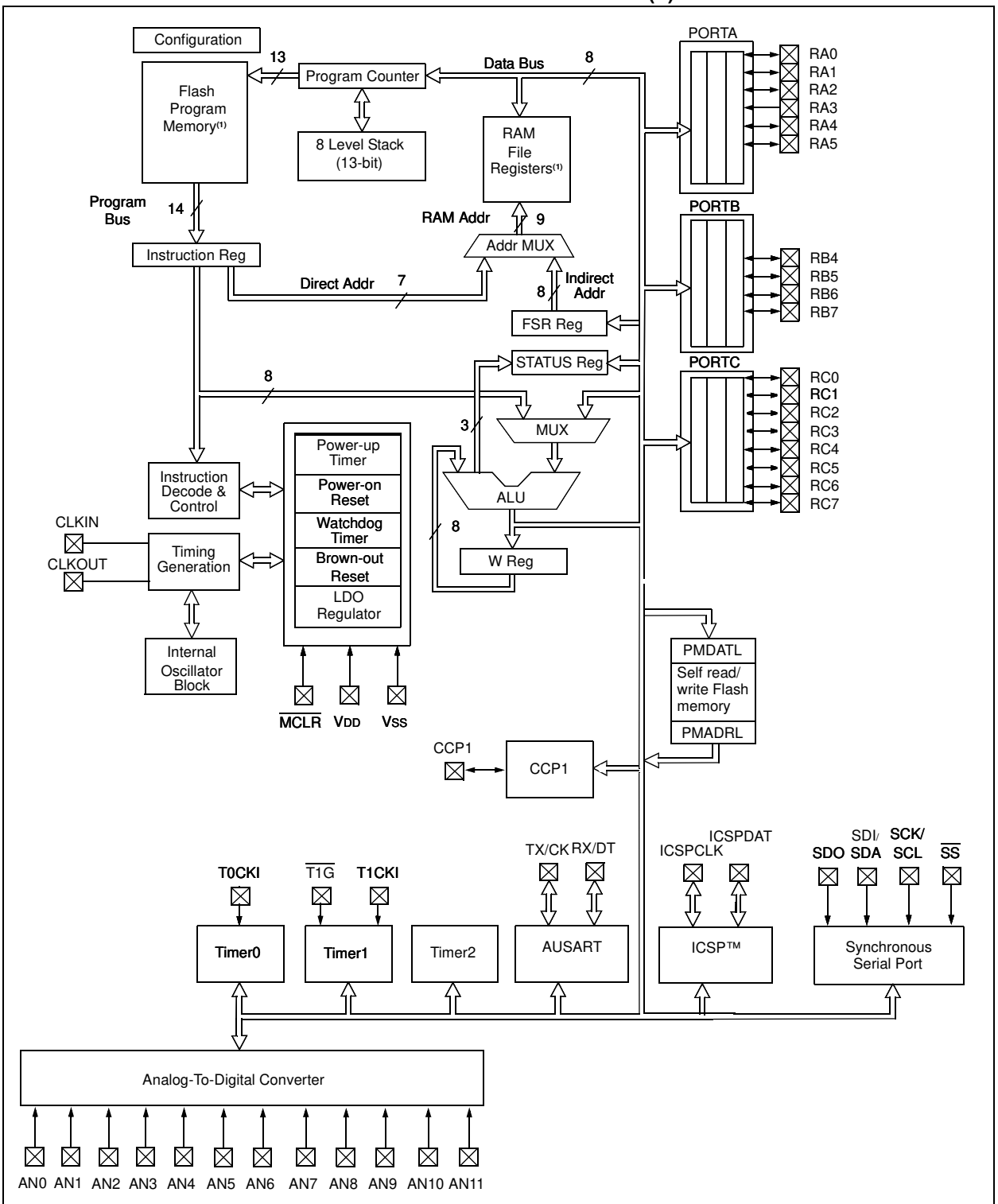
Customer Notification System

Register on our website at **www.microchip.com** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

The PIC16(L)F720/721 devices are covered by this data sheet. They are available in 20-pin packages. Please refer to **Section 25.0 "Packaging Information"** for further package information. Figure 1-1 shows a block diagram of the PIC16(L)F720/721 devices. Table 1-1 shows the pinout descriptions.

FIGURE 1-1: 20-PIN DEVICE BLOCK DIAGRAM FOR PIC16(L)F720/721



Note: PIC16(L)F720 – 2k x 14 Flash, 128 x 8 RAM
PIC16(L)F721 – 4k x 14 Flash, 256 x 8 RAM.

TABLE 1-1: PINOUT DESCRIPTION

Name	Function	IN	OUT	Description
RA0/AN0/ICSPDAT	RA0	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN0	AN	—	A/D Channel 0 Input.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
RA1/AN1/ICSPCLK	RA1	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN1	AN	—	A/D Channel 1 Input.
	ICSPCLK	ST	—	ICSP™ Clock.
RA2/AN2/T0CKI/INT	RA2	TTL	CMOS	General purpose I/O with IOC and WPU.
	AN2	AN	—	A/D Channel 2 Input.
	T0CKI	ST	—	Timer0 Clock Input.
	INT	ST	—	External interrupt.
RA3/$\overline{MCLR}$/VPP	RA3	TTL	—	General purpose input-only with IOC and WPU.
	$\overline{MCLR}$	ST	—	Master Clear with internal pull-up.
	VPP	HV	—	Programming Voltage.
RA4/AN3/$\overline{T1G}$/CLKOUT	RA4	TTL	CMOS	General purpose I/O with IOC and WPU.
	AN3	AN	—	A/D Channel 3 Input.
	$\overline{T1G}$	ST	—	Timer1 Gate Input.
	CLKOUT	—	CMOS	FOSC/4 output.
RA5/T1CKI/CLKIN	RA5	TTL	CMOS	General purpose I/O with IOC and WPU.
	T1CKI	ST	—	Timer1 Clock input.
	CLKIN	ST	—	External Clock Input (EC mode).
RB4/AN10/SDI/SDA	RB4	TTL	CMOS	General purpose I/O with IOC and WPU.
	AN10	AN	—	A/D Channel 10 Input.
	SDI	ST	—	SPI Data Input.
	SDA	I^2C	OD	I^2C Data.
RB5/AN11/RX/DT	RB5	TTL	CMOS	General purpose I/O with IOC and WPU.
	AN11	AN	—	A/D Channel 11 Input.
	RX	ST	—	USART asynchronous input.
	DT	ST	CMOS	USART synchronous data.
RB6/SCK/SCL	RB6	TTL	CMOS	General purpose I/O with IOC and WPU.
	SCK	ST	CMOS	SPI Clock.
	SCL	I^2C	OD	I^2C Clock.
RB7/TX/CK	RB7	TTL	CMOS	General purpose I/O with IOC and WPU.
	TX	—	CMOS	USART asynchronous transmit.
	CK	ST	CMOS	USART synchronous clock.
RC0/AN4	RC0	ST	CMOS	General purpose I/O.
	AN4	AN	—	A/D Channel 4 Input.
RC1/AN5	RC1	ST	CMOS	General purpose I/O.
	AN5	AN	—	A/D Channel 5 Input.
RC2/AN6	RC2	ST	CMOS	General purpose I/O.
	AN6	AN	—	A/D Channel 6 Input.
RC3/AN7	RC3	ST	CMOS	General purpose I/O.
	AN7	AN	—	A/D Channel 7 Input.

Legend: AN = Analog input or output, CMOS = CMOS compatible input or output, OD = Open Drain, TTL = TTL compatible input, ST = Schmitt Trigger input with CMOS levels, I^2C = Schmitt Trigger input with I^2C, HV = High Voltage, XTAL = Crystal levels

TABLE 1-1: PINOUT DESCRIPTION (CONTINUED)

Name	Function	IN	OUT	Description
RC4	RC4	ST	CMOS	General purpose I/O.
RC5/CCP1	RC5	ST	CMOS	General purpose I/O.
	CCP1	ST	CMOS	Capture/Compare/PWM 1.
RC6/AN8/$\overline{SS}$	RC6	ST	CMOS	General purpose I/O.
	AN8	AN	—	A/D Channel 8 Input.
	$\overline{SS}$	ST	—	Slave Select input.
RC7/AN9/SDO	RC7	ST	CMOS	General purpose I/O.
	AN9	AN	—	A/D Channel 9 Input.
	SDO	—	CMOS	SPI Data Output.
VDD	VDD	Power	—	Positive supply.
Vss	Vss	Power	—	Ground supply.

Legend: AN = Analog input or output, CMOS = CMOS compatible input or output, OD = Open Drain, TTL = TTL compatible input, ST = Schmitt Trigger input with CMOS levels, I^2C = Schmitt Trigger input with I^2C, HV = High Voltage, XTAL = Crystal levels

2.0 MEMORY ORGANIZATION

2.1 Program Memory Organization

The PIC16(L)F720/721 has a 13-bit program counter capable of addressing a 8K x 14 program memory space. Table 2-1 shows the memory sizes implemented. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h.

TABLE 2-1: DEVICE SIZE AND ADDRESSES

Device	Program Memory Size (Words)	Last Program Memory Address	High-Endurance Flash Memory Address Range (1)
PIC16F720 PIC16LF720	2048	07FFh	0780h-07FFh
PIC16F721 PIC16LF721	4096	0FFFh	0F80h-0FFFh

Note 1: High-Endurance Flash applies to the low byte of each address in the range.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F720

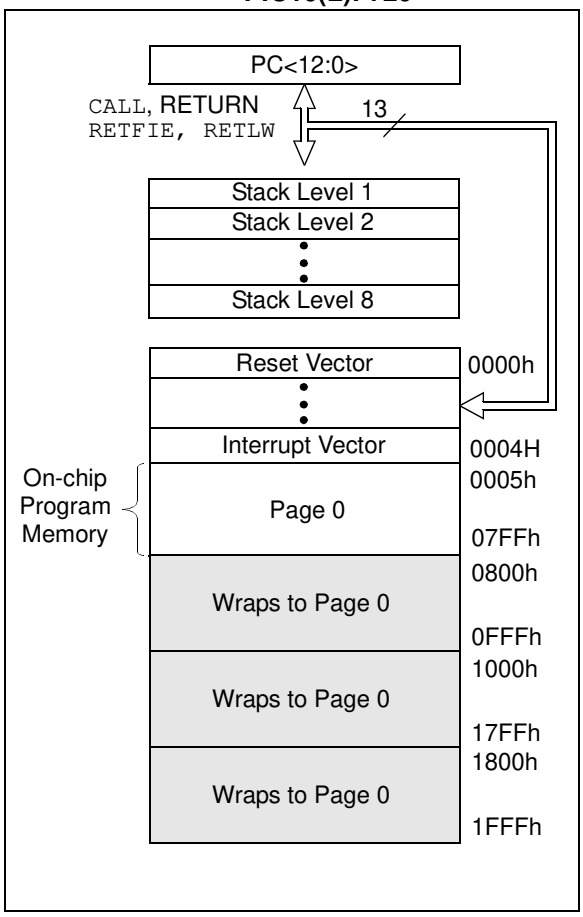
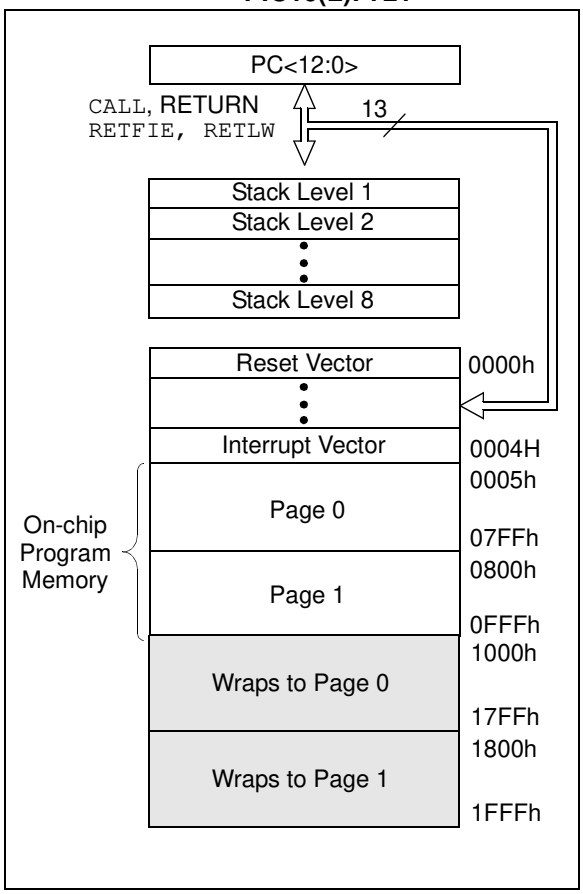


FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F721



2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers (GPRs) and the Special Function Registers (SFRs). Bits RP0 and RP1 are bank select bits.

RP1	RP0		
0	0	→	Bank 0 is selected
0	1	→	Bank 1 is selected
1	0	→	Bank 2 is selected
1	1	→	Bank 3 is selected

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are the General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank are mirrored in another bank for code reduction and quicker access.

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 128 x 8 bits in the PIC16(L)F720, 256 x 8 bits in the PIC16(L)F721. Each register is accessed either directly or indirectly through the File Select Register (FSR), (Refer to **Section 2.5 "Indirect Addressing, INDF and FSR Registers"**).

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (refer to Table 2-2). These registers are static RAM.

The Special Function Registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the "core" are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

FIGURE 2-3: PIC16(L)F720 SPECIAL FUNCTION REGISTERS

	File Address						File Address
INDF(*)	00h	INDF(*)	80h	INDF(*)	100h	INDF(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h	ANSELA	185h
PORTB	06h	TRISB	86h		106h	ANSELB	186h
PORTC	07h	TRISC	87h		107h	ANSELC	187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	PMDATL	10Ch	PMCON1	18Ch
	0Dh		8Dh	PMADRL	10Dh	PMCON2	18Dh
TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh		18Eh
TMR1H	0Fh	T1GCON	8Fh	PMADRH	10Fh		18Fh
T1CON	10h	OSCCON	90h		110h		190h
TMR2	11h	OSCTUNE	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD/SSPMSK	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h	WPUA	95h	WPUB	115h		195h
CCPR1H	16h	IOCA	96h	IOCB	116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
	1Bh		9Bh		11Bh		19Bh
	1Ch		9Ch		11Ch		19Ch
	1Dh	FVRCON	9Dh		11Dh		19Dh
ADRES	1Eh		9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 80 Bytes	20h – 06Fh	General Purpose Register 32 Bytes	A0h – BFh		120h – 16Fh		1A0h – 1EFh
			C0h – EFh				
Access RAM	070h – 7Fh	Accesses 70h – 7Fh	F0h – FFh	Accesses 70h – 7Fh	170h – 17Fh	Accesses 70h – 7Fh	1F0h – 1FFh
BANK 0		BANK 1		BANK 2		BANK 3	

Legend: = Unimplemented data memory locations, read as '0'.
* = Not a physical register.

FIGURE 2-4: PIC16(L)F721 SPECIAL FUNCTION REGISTERS

	File Address		File Address		File Address		File Address
INDF(*)	00h	INDF(*)	80h	INDF(*)	100h	INDF(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h	ANSELA	185h
PORTB	06h	TRISB	86h		106h	ANSELB	186h
PORTC	07h	TRISC	87h		107h	ANSELC	187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	PMDATL	10Ch	PMCON1	18Ch
	0Dh		8Dh	PMADRL	10Dh	PMCON2	18Dh
TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh		18Eh
TMR1H	0Fh	T1GCON	8Fh	PMADRH	10Fh		18Fh
T1CON	10h	OSCCON	90h		110h		190h
TMR2	11h	OSCTUNE	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD/SSPMSK	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h	WPUA	95h	WPUB	115h		195h
CCPR1H	16h	IOCA	96h	IOCB	116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
	1Bh		9Bh		11Bh		19Bh
	1Ch		9Ch		11Ch		19Ch
	1Dh	FVRCON	9Dh		11Dh		19Dh
ADRES	1Eh		9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 80 Bytes	20h – 06Fh	General Purpose Register 80 Bytes	A0h – EFh	General Purpose Register 80 Bytes	120h – 16Fh		1A0h – 1EFh
Access RAM	070h – 7Fh	Accesses 70h – 7Fh	F0h – FFh	Accesses 70h – 7Fh	170h – 17Fh	Accesses 70h – 7Fh	1F0h – 1FFh
BANK 0		BANK 1		BANK 2		BANK 3	

Legend: = Unimplemented data memory locations, read as '0'.
* = Not a physical register.

TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 0											
00h(2)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
01h	TMR0	Timer0 module Register								xxxx xxxx	uuuu uuuu
02h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
03h(2)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h(2)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--xx xxxx
06h	PORTB	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	uuuu ----
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah(1),(2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh(2)	INTCON	GIE	PEIE	TMR0IE	INTE	RABIE	TMR0IF	INTF	RABIF	0000 000x	0000 000x
0Ch	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	—	$\overline{T1SYNC}$	—	TMR1ON	0000 -0-0	uuuu -u-u
11h	TMR2	Timer2 module Register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	Capture/Compare/PWM Register Low Byte								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register High Byte								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	DC1	B1	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	AUSART Transmit Data Register								0000 0000	0000 0000
1Ah	RCREG	AUSART Receive Data Register								0000 0000	0000 0000
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRES	ADC Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

2: These registers can be addressed from any bank.

3: Accessible only when SSPM<3:0> = 1001.

4: This bit is unimplemented and reads as '1'.

5: See Register 6-2.

TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 1											
80h(2)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
81h	OPTION_REG	$\overline{RABPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
83h(2)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h(2)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
85h(5)	TRISA	—	—	TRISA5	TRISA4	—(4)	TRISA2	TRISA1	TRISA0	--11 -111	--11 -111
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah(1),(2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
8Bh(2)	INTCON	GIE	PEIE	TMR0IE	INTE	RABIE	TMR0IF	INTF	RABIF	0000 000x	0000 000x
8Ch	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	---- --qq	---- --uu
8Fh	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{DONE}$	T1GVAL	T1GSS1	T1GSS0	0000 0x00	uuuu uxuu
90h	OSCCON	—	—	IRCF1	IRCF0	ICSL	ICSS	—	—	--10 qq--	--10 qq--
91h	OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	--00 0000	--uu uuuu
92h	PR2	Timer2 module Period Register								1111 1111	1111 1111
93h	SSPADD	ADD<7:0>								0000 0000	0000 0000
93h(3)	SSPMSK	MSK<7:0>								1111 1111	1111 1111
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000
95h	WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	--11 1111	--11 1111
96h	IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	--00 0000
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	FVRCON	FVRRDY	FVREN	TSEN	TSRNG	—	—	ADFVR1	ADFVR0	q000 --00	q000 --00
9Eh	—	Unimplemented								—	—
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	-000 ----

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

2: These registers can be addressed from any bank.

3: Accessible only when SSPM<3:0> = 1001.

4: This bit is unimplemented and reads as '1'.

5: See Register 6-2.

TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 2											
100h(2)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
101h	TMR0	Timer0 module Register								xxxx xxxx	uuuu uuuu
102h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
103h(2)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
104h(2)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
105h	—	Unimplemented								—	—
106h	—	Unimplemented								—	—
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah(1),(2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
10Bh(2)	INTCON	GIE	PEIE	TMR0IE	INTE	RABIE	TMR0IF	INTF	RABIF	0000 000x	0000 000x
10Ch	PMDATL	Program Memory Read Data Register Low Byte								xxxx xxxx	xxxx xxxx
10Dh	PMADRL	Program Memory Read Address Register Low Byte								0000 0000	0000 0000
10Eh	PMDATH	—	—	Program Memory Read Data Register High Byte						--xx xxxx	--xx xxxx
10Fh	PMADRH	—	—	—	Program Memory Read Address Register High Byte					---0 0000	---0 0000
110h	—	Unimplemented								—	—
111h	—	Unimplemented								—	—
112h	—	Unimplemented								—	—
113h	—	Unimplemented								—	—
114h	—	Unimplemented								—	—
115h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	1111 ----
116h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	0000 ----	0000 ----
117h	—	Unimplemented								—	—
118h	—	Unimplemented								—	—
119h	—	Unimplemented								—	—
11Ah	—	Unimplemented								—	—
11Bh	—	Unimplemented								—	—
11Ch	—	Unimplemented								—	—
11Dh	—	Unimplemented								—	—
11Eh	—	Unimplemented								—	—
11Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

2: These registers can be addressed from any bank.

3: Accessible only when SSPM<3:0> = 1001.

4: This bit is unimplemented and reads as '1'.

5: See Register 6-2.

TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 3											
180h(2)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
181h	OPTION_REG	$\overline{\text{RABPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
183h(2)	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu
184h(2)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
185h	ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	---1 -111	---1 -111
186h	ANSELB	—	—	ANSB5	ANSB4	—	—	—	—	--11 ----	--11 ----
187h	ANSELC	ANSC7	ANSC6	—	—	ANSC3	ANSC2	ANSC1	ANSC0	11-- 1111	11-- 1111
188h	—	Unimplemented								—	—
18Ah(1),(2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
18Bh(2)	INTCON	GIE	PEIE	TMR0IE	INTE	RABIE	TMR0IF	INTF	RABIF	0000 000x	0000 000x
18Ch	PMCON1	—(4)	CFGS	LWLO	FREE	—	WREN	WR	RD	1000 -000	1000 -000
18Dh	PMCON2	Program Memory Control Register 2 (not a physical register)								---- ----	---- ----
190h	—	Unimplemented								—	—
191h	—	Unimplemented								—	—
192h	—	Unimplemented								—	—
193h	—	Unimplemented								—	—
194h	—	Unimplemented								—	—
195h	—	Unimplemented								—	—
196h	—	Unimplemented								—	—
197h	—	Unimplemented								—	—
198h	—	Unimplemented								—	—
199h	—	Unimplemented								—	—
19Ah	—	Unimplemented								—	—
19Bh	—	Unimplemented								—	—
19Ch	—	Unimplemented								—	—
19Dh	—	Unimplemented								—	—
19Eh	—	Unimplemented								—	—
19Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

2: These registers can be addressed from any bank.

3: Accessible only when SSPM<3:0> = 1001.

4: This bit is unimplemented and reads as '1'.

5: See Register 6-2.

2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, contains:

- the arithmetic status of the ALU
- the Reset status
- the bank select bits for data memory (SRAM)

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as '000u u1uu' (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits (Refer to **Section 21.0 "Instruction Set Summary"**).

Note 1: The C and DC bits operate as $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ out bits, respectively, in subtraction.

REGISTER 2-1: STATUS: STATUS REGISTER

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC(1)	C(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h-1FFh)
0 = Bank 0, 1 (00h-FFh)

bit 6-5 **RP<1:0>:** Register Bank Select bits (used for direct addressing)
00 = Bank 0 (00h-7Fh)
01 = Bank 1 (80h-FFh)
10 = Bank 2 (100h-17Fh)
11 = Bank 3 (180h-1FFh)

bit 4 **$\overline{TO}$:** Time-out bit
1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction
0 = A WDT time out occurred

bit 3 **$\overline{PD}$:** Power-down bit
1 = After power-up or by the `CLRWDT` instruction
0 = By execution of the `SLEEP` instruction

bit 2 **Z:** Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/$\overline{Digit\ Borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)(1)
1 = A carry out from the 4th low-order bit of the result occurred
0 = No carry out from the 4th low-order bit of the result

bit 0 **C:** Carry/$\overline{Borrow}$ bit(1) (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)(1)
1 = A carry out from the Most Significant bit of the result occurred
0 = No carry out from the Most Significant bit of the result occurred

Note 1: For $\overline{Borrow}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate instructions (`RRF`, `RLF`), this bit is loaded with either the high-order or low-order bit of the source register.

2.2.2.2 OPTION_REG Register

The OPTION_REG register, shown in Register 2-2, is a readable and writable register, which contains various control bits to configure:

- Software programmable prescaler for the Timer0/ WDT
- External RA2/INT interrupt
- Timer0
- Weak pull-ups on PORTA or PORTB

Note: To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT by setting the PSA bit of the OPTION_REG register to '1'. Refer to **Section 12.1.3 "Software Programmable Prescaler"**.

REGISTER 2-2: OPTION_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RABPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **$\overline{\text{RABPU}}$:** PORTA or PORTB Pull-up Enable bit
1 = PORTA or PORTB pull-ups are disabled
0 = PORTA or PORTB pull-ups are enabled by individual bits in the WPUA or WPUB register, respectively

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of INT pin
0 = Interrupt on falling edge of INT pin

bit 5 **T0CS:** Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **T0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	Timer0 Rate	WDT Rate
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

2.2.2.3 PCON Register

The Power Control (PCON) register contains flag bits (refer to Table 3-4) to differentiate between a:

- Power-on Reset ($\overline{POR}$)
- Brown-out Reset ($\overline{BOR}$)
- Watchdog Timer Reset (WDT)
- External $\overline{MCLR}$ Reset

The PCON register also controls the software enable of the BOR.

The PCON register bits are shown in Register 2-3.

REGISTER 2-3: PCON: POWER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-q	R/W-q
—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
q = Value depends on condition			

bit 7-2 **Unimplemented:** Read as '0'

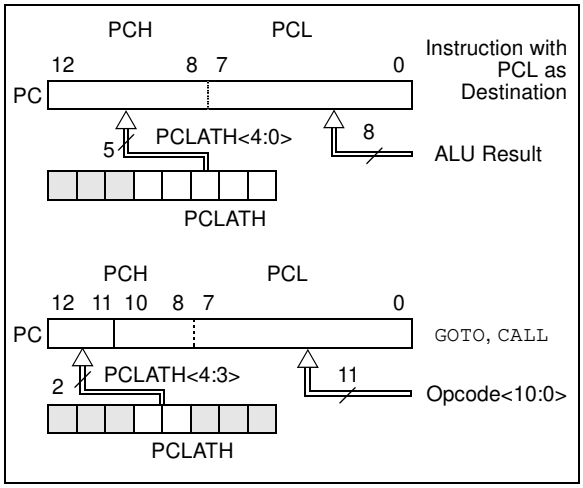
bit 1 **$\overline{POR}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-5 shows the two situations for the loading of the PC. The upper example in Figure 2-5 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-5 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the Application Note AN556, *"Implementing a Table Read"* (DS00556).

2.3.2 STACK

All devices have an 8-level x 13-bit wide hardware stack (refer to Figures 2-1 and 2-2). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth PUSH overwrites the value that was stored from the first PUSH. The tenth PUSH overwrites the second PUSH (and so on).

Note 1: There are no Status bits to indicate stack overflow or stack underflow conditions.

2: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

2.4 Program Memory Paging

All devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper two bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page Select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is POPed off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the RETURN instructions (which POPs the address from the stack).

Note: The contents of the PCLATH register are unchanged after a RETURN or RETFIE instruction is executed. The user must rewrite the contents of the PCLATH register for any subsequent subroutine calls or GOTO instructions.

Example 2-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the Interrupt Service Routine (if interrupts are used).

EXAMPLE 2-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```
        ORG 500h
        PAGESEL SUB_P1  ;Select page 1
                        ;(800h-FFFh)
        CALL    SUB1_P1 ;Call subroutine in
        :               ;page 1 (800h-FFFh)
        :
        ORG     900h    ;page 1 (800h-FFFh)
SUB1_P1
        :               ;called subroutine
                        ;page 1 (800h-FFFh)
        :
        RETURN          ;return to
                        ;Call subroutine
                        ;in page 0
                        ;(000h-7FFh)
```

2.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

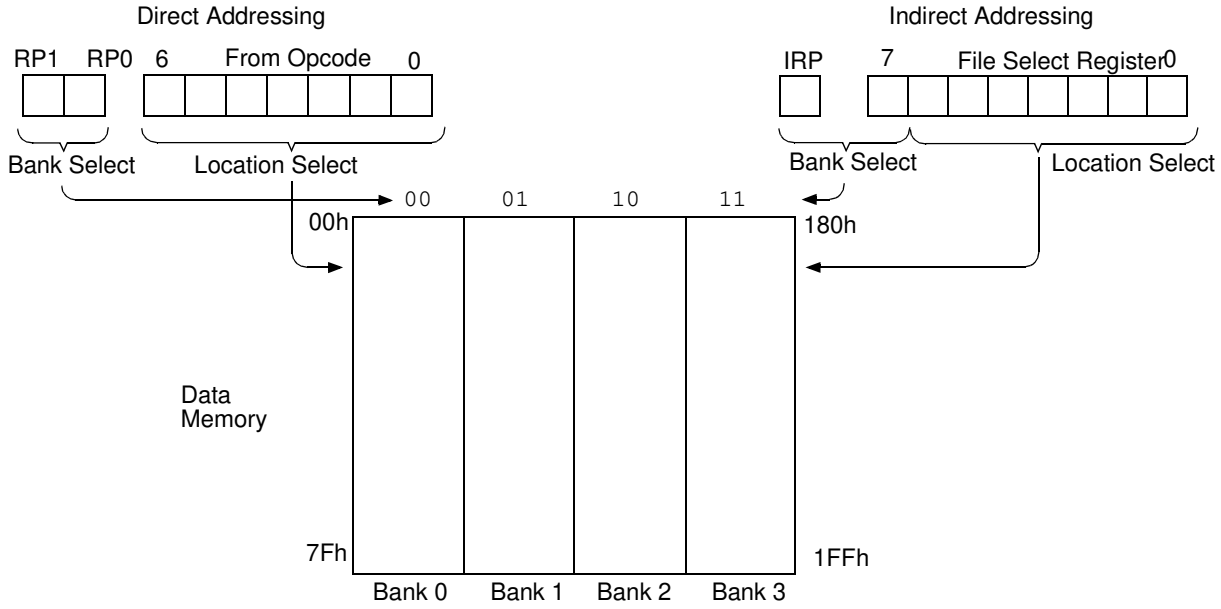
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit of the STATUS register, as shown in Figure 2-6.

A simple program to clear the RAM location 020h-02Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: INDIRECT ADDRESSING

```
    MOVLW  020h  ;initialize pointer
    MOVWF  FSR   ;to RAM
    BANKISEL 020h
NEXT CLRF  INDF  ;clear INDF register
     INCF  FSR   ;inc pointer
     BTFSS FSR,4 ;all done?
     GOTO  NEXT  ;no clear next
CONTINUE         ;yes continue
```

FIGURE 2-6: DIRECT/INDIRECT ADDRESSING



Note: For memory map detail, refer to Figures 2-3 and 2-4.

3.0 RESETS

The PIC16(L)F720/721 differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) WDT Reset during normal operation
c) WDT Reset during Sleep
d) $\overline{\text{MCLR}}$ Reset during normal operation
e) $\overline{\text{MCLR}}$ Reset during Sleep
f) Brown-out Reset (BOR)

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on:

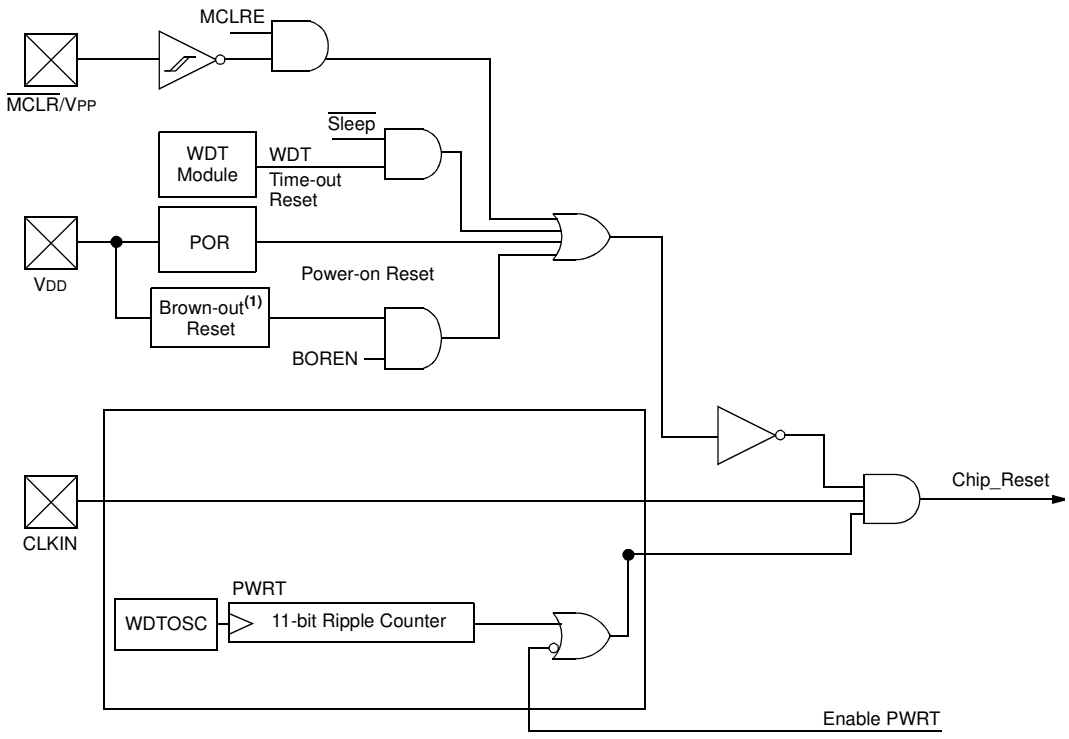
- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

Most registers are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations, as indicated in Table 3-5. These bits are used in software to determine the nature of the Reset.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 3-1.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. See **Section 23.0 "Electrical Specifications"** for pulse-width specifications.

FIGURE 3-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: Refer to the Configuration Word Register 1 (Register 8-1).

TABLE 3-1: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{POR}$	$\overline{BOR}$	$\overline{TO}$	$\overline{PD}$	Condition
0	x	1	1	Power-on Reset or LDO Reset
0	x	0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	x	x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
1	0	1	1	Brown-out Reset
1	1	0	1	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	$\overline{MCLR}$ Reset during normal operation
1	1	1	0	$\overline{MCLR}$ Reset during Sleep or interrupt wake-up from Sleep

TABLE 3-2: RESET CONDITION FOR SPECIAL REGISTERS(2)

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	0001 1xxx	---- --0x
$\overline{MCLR}$ Reset during normal operation	0000h	000u uuuu	---- --uu
$\overline{MCLR}$ Reset during Sleep	0000h	0001 0uuu	---- --uu
WDT Reset	0000h	0000 1uuu	---- --uu
WDT Wake-up	PC + 1	uuu0 0uuu	---- --uu
Brown-out Reset	0000h	0001 1uuu	---- --u0
Interrupt Wake-up from Sleep	PC + 1(1)	uuu1 0uuu	---- --uu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

Note 1: When the wake-up is due to an interrupt and Global Interrupt Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

2: If a Status bit is not implemented, that bit will be read as '0'.