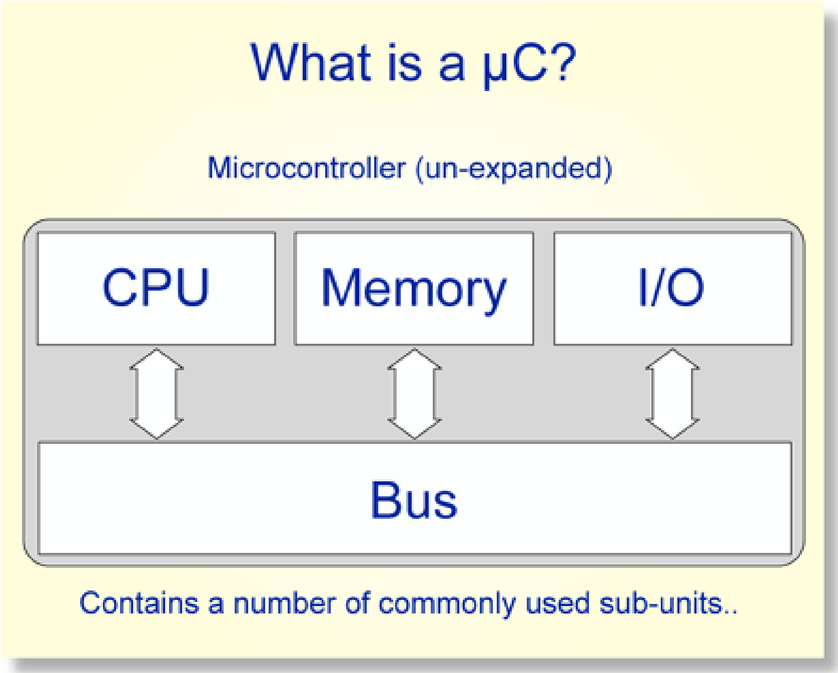
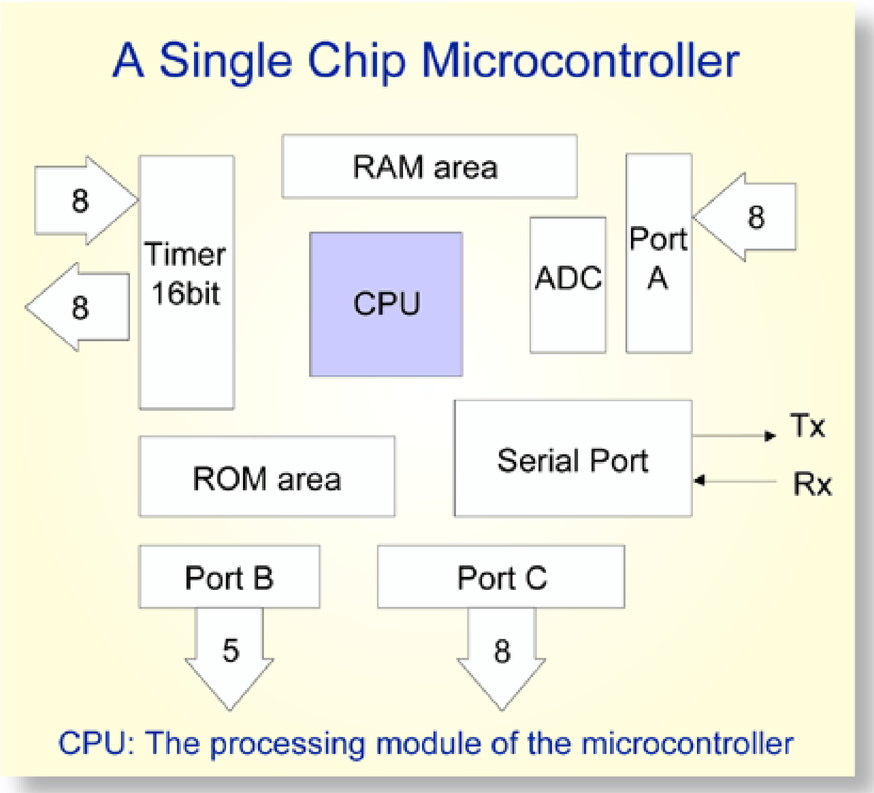
Microcontroller Peripherals

Overview of Microcontrollers


What is a μC?
Microcontroller (un-expanded)
CPU
Memory
I/O
Bus
Contains a number of commonly used sub-units..


A Single Chip Microcontroller
8
8
Timer 16bit
RAM area
CPU
ADC
Port A
8
Tx
Rx
Serial Port
ROM area
Port B
Port C
5
8
CPU: The processing module of the microcontroller

- Basically, a microcontroller is a device which integrates a number of the components of a microprocessor system onto a single microchip.

Reference: http://mic.unn.ac.uk/miclearning/modules/micros/ch1/micro01notes.html#1.4

Microcontroller features

- Processor
  - Usually general-purpose but can be app-specific
- On-chip memory
  - Often RAM for data, EEPROM/Flash for code
- Integrated peripherals
  - Common peripherals
    - Parallel I/O port(s)
    - Clock generator(s)
    - Timers/event counters
  - Special-purpose devices such as:
    - Analog-to-digital converter (sensor inputs)
    - Mixed signal components
    - Serial port + other serial interfaces (SPI, USB)
    - Ethernet

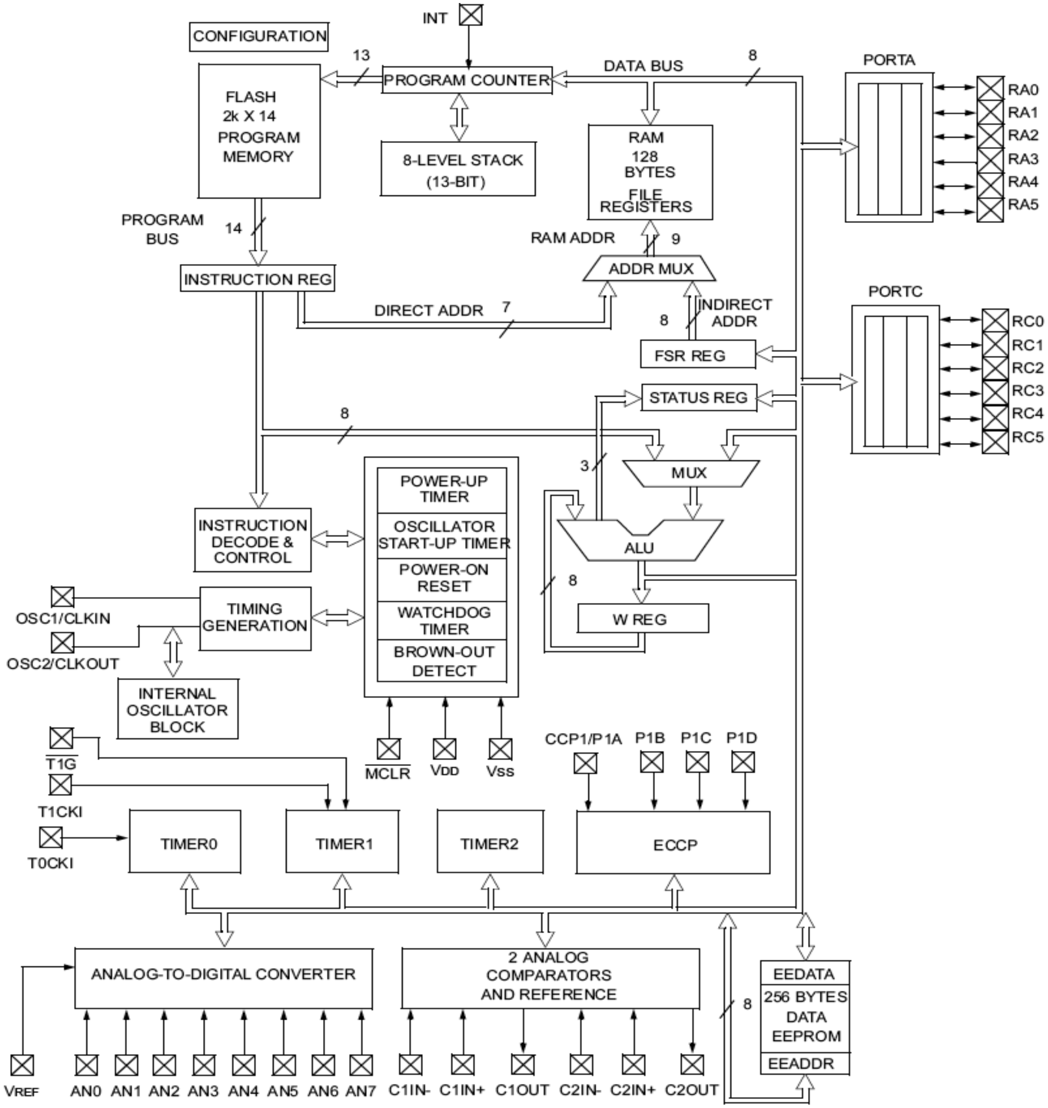
Microcontroller features

- Benefits
  - Typically low-power/low-cost
    - Target for embedded applications
  - Easily programmable
    - Simple ISAs (RISC processors)
    - Use of development kits simplifies process
- Limitations
  - Small storage space (registers, memory)
  - Restricted instruction set
  - May be required to multiplex pins
  - Not typically used for high performance

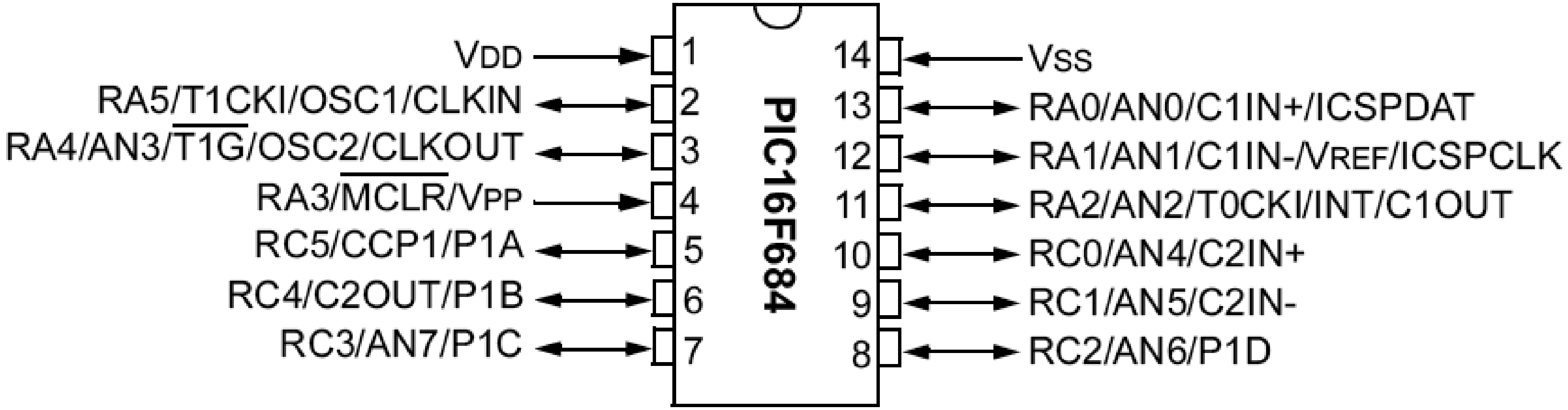
PIC Microcontroller (PIC16F684)

- High performance, low cost, for embedded applications
  - Only 35 different instructions
  - Interrupt capability
  - Direct, indirect, relative addressing mode
- Low Power
  - 8.5uA @ 32KHz, 2.0V
- Peripheral Features
  - 12 I/O pins with individual direction control
  - 10-bit A/D converter
  - 8/16-bit timer/counter
- Special Microcontroller Features
  - Internal/external oscillator
  - Power saving sleep mode
  - High Endurance Flash/EEPROM cell

PIC16F684 Block Diagram



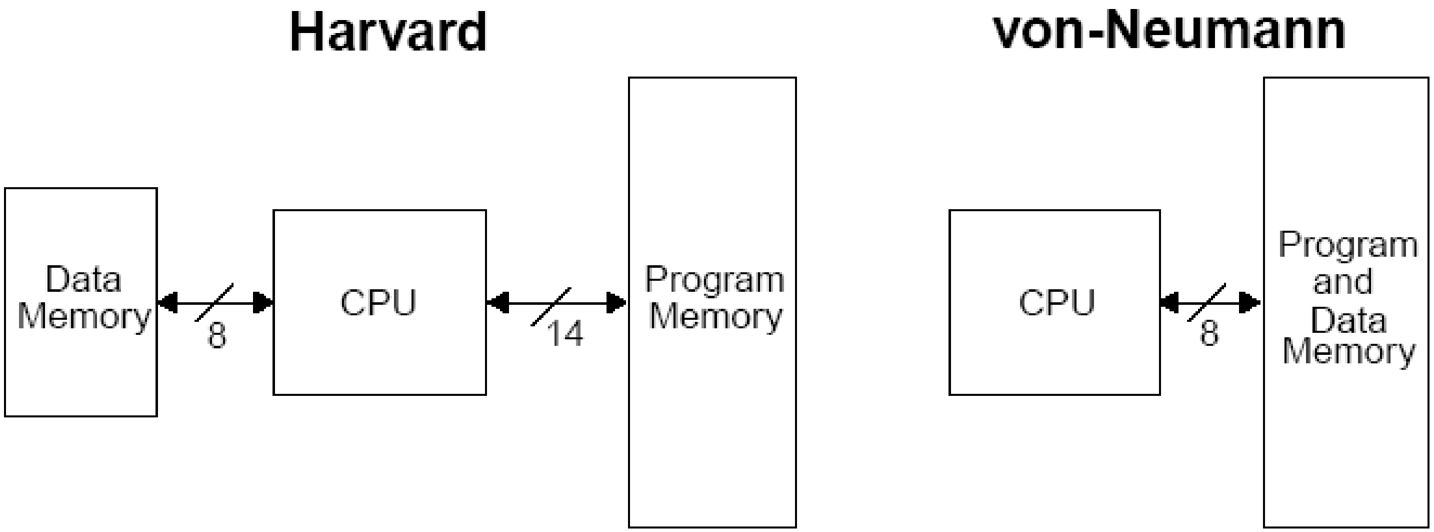
PIC16F684



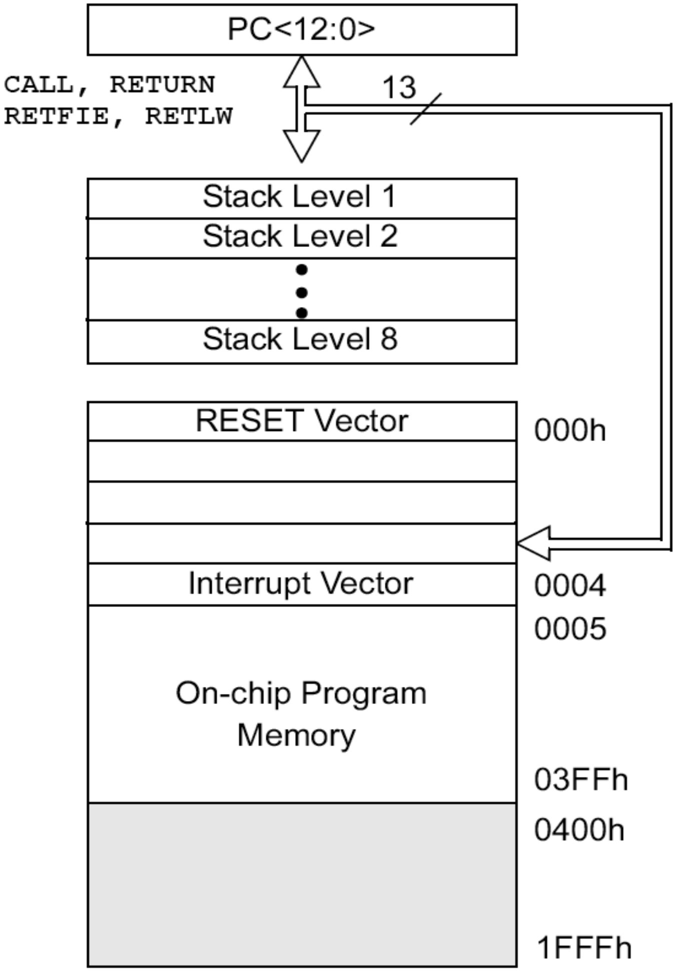
- 12 pins, 2048 instructions, 128 byte variable memory, ADC, comparator, Timers, ICD

Harvard vs Von Neumann

- Organization of program and data memory

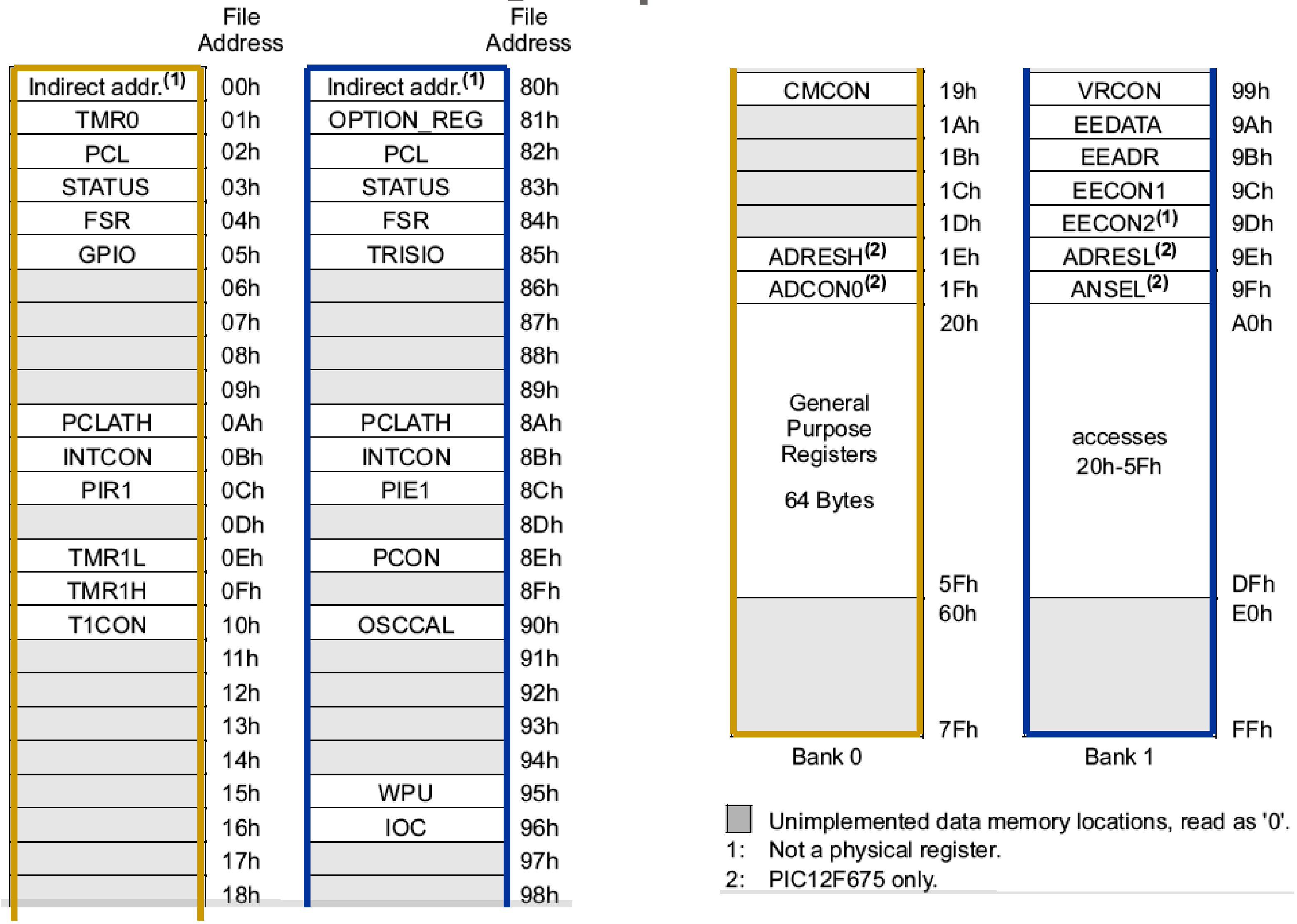


Program Memory Space



- 13-bit program counter to address 8K locations
- Each location is 14-bit wide (instructions are 14 bits long)
- RESET vector is 0000h
  - When the CPU is reset, its PC is automatically cleared to zero.
- Interrupt Vector is 0004h
  - 0004h is automatically loaded into the program counter when an interrupt occurs
- Vector → address of code to be executed for given interrupt

Data Memory Space



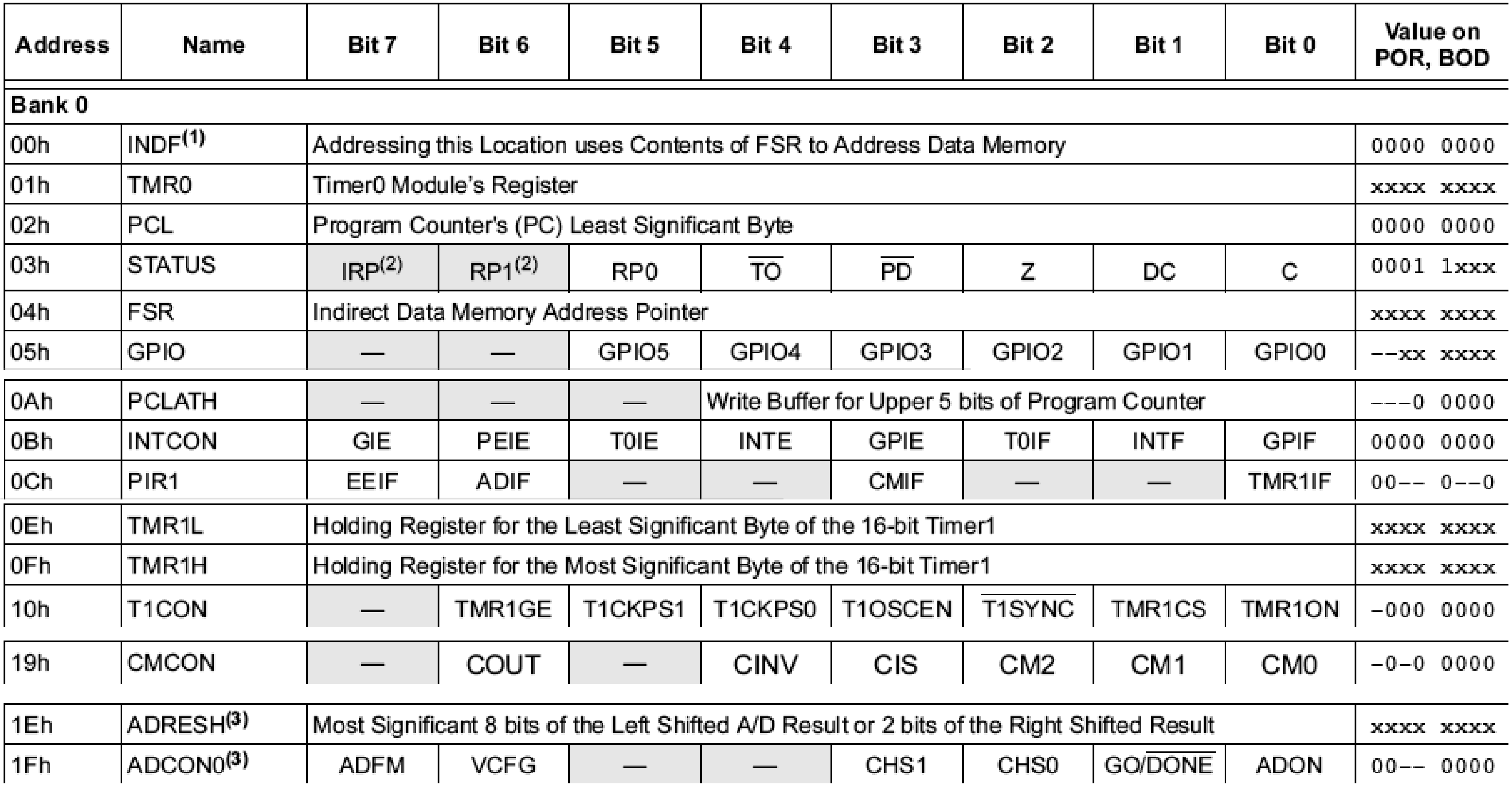
	File Address		File Address
Indirect addr.[1]	00h	Indirect addr.[1]	80h
TMR0	01h	OPTION_REG	81h
PCL	02h	PCL	82h
STATUS	03h	STATUS	83h
FSR	04h	FSR	84h
GPIO	05h	TRISIO	85h
▨	06h	▨	86h
▨	07h	▨	87h
▨	08h	▨	88h
▨	09h	▨	89h
PCLATH	0Ah	PCLATH	8Ah
INTCON	0Bh	INTCON	8Bh
PIR1	0Ch	PIE1	8Ch
▨	0Dh	▨	8Dh
TMR1L	0Eh	PCON	8Eh
TMR1H	0Fh	▨	8Fh
T1CON	10h	OSCCAL	90h
▨	11h	▨	91h
▨	12h	▨	92h
▨	13h	▨	93h
▨	14h	▨	94h
▨	15h	WPU	95h
▨	16h	IOC	96h
▨	17h	▨	97h
▨	18h	▨	98h
CMCON	19h	VRCON	99h
▨	1Ah	EEDATA	9Ah
▨	1Bh	EEADR	9Bh
▨	1Ch	EECON1	9Ch
▨	1Dh	EECON2[1]	9Dh
ADRESH[2]	1Eh	ADRESL[2]	9Eh
ADCON0[2]	1Fh	ANSEL[2]	9Fh
General Purpose Registers 64 Bytes	20h 5Fh	accesses 20h-5Fh	A0h DFh
▨	60h 7Fh	▨	E0h FFh
Bank 0		Bank 1	

▨ Unimplemented data memory locations, read as '0'.

1: Not a physical register.

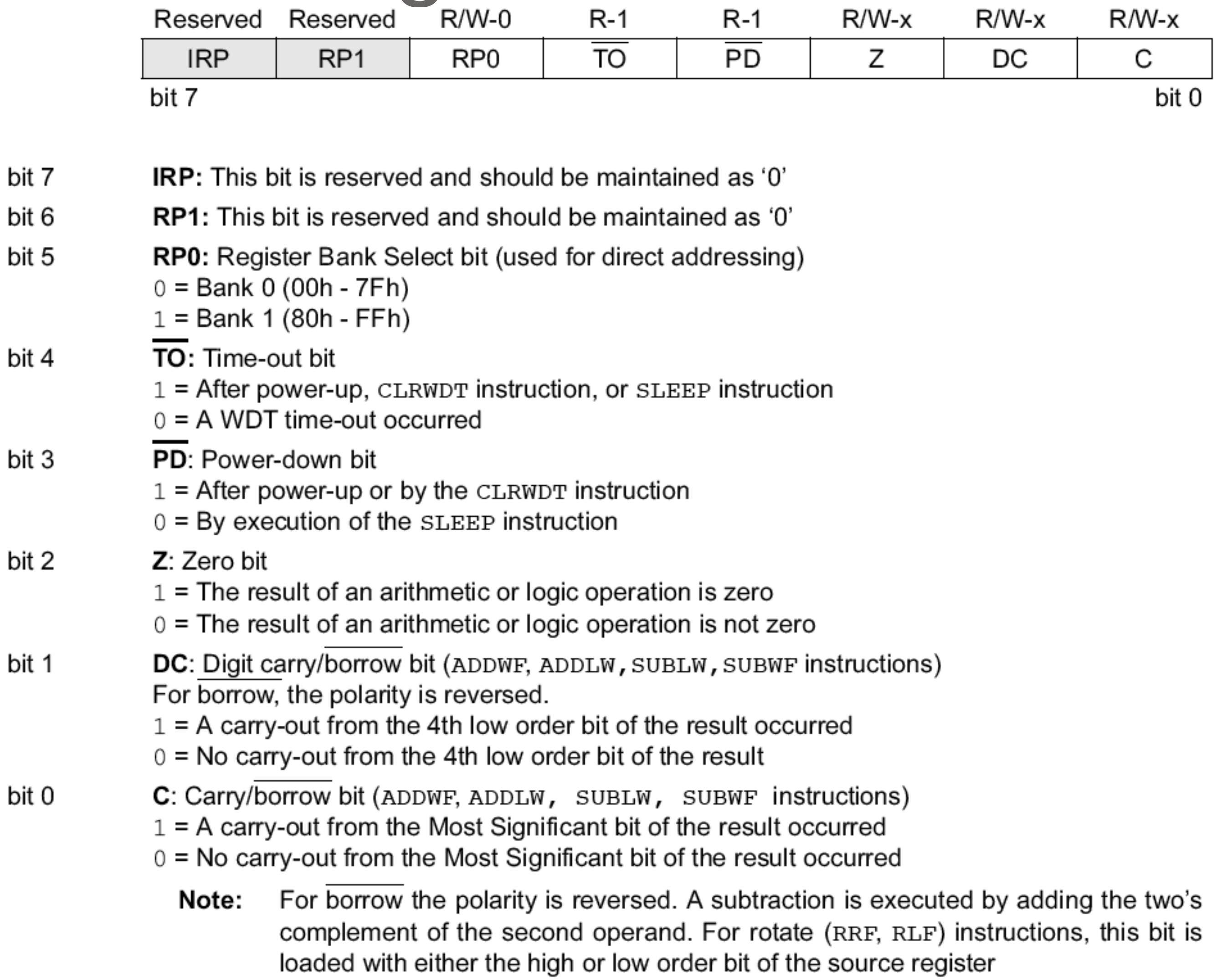
2: PIC12F675 only.

Special Function Registers



Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD
Bank 0										
00h	INDF(1)	Addressing this Location uses Contents of FSR to Address Data Memory								0000 0000
01h	TMR0	Timer0 Module's Register								xxxx xxxx
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000
03h	STATUS	IRP(2)	RP1(2)	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx
04h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx
05h	GPIO	—	—	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0	––xx xxxx
0Ah	PCLATH	—	—	—	Write Buffer for Upper 5 bits of Program Counter					–––0 0000
0Bh	INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000
0Ch	PIR1	EEIF	ADIF	—	—	CMIF	—	—	TMR1IF	00–– 0––0
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit Timer1								xxxx xxxx
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit Timer1								xxxx xxxx
10h	T1CON	—	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	–000 0000
19h	CMCON	—	COUT	—	CINV	CIS	CM2	CM1	CM0	–0–0 0000
1Eh	ADRESH(3)	Most Significant 8 bits of the Left Shifted A/D Result or 2 bits of the Right Shifted Result								xxxx xxxx
1Fh	ADCON0(3)	ADFM	VCFG	—	—	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	00–– 0000

Status Register



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

bit 7 **IRP:** This bit is reserved and should be maintained as '0'

bit 6 **RP1:** This bit is reserved and should be maintained as '0'

bit 5 **RP0:** Register Bank Select bit (used for direct addressing)
0 = Bank 0 (00h - 7Fh)
1 = Bank 1 (80h - FFh)

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

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

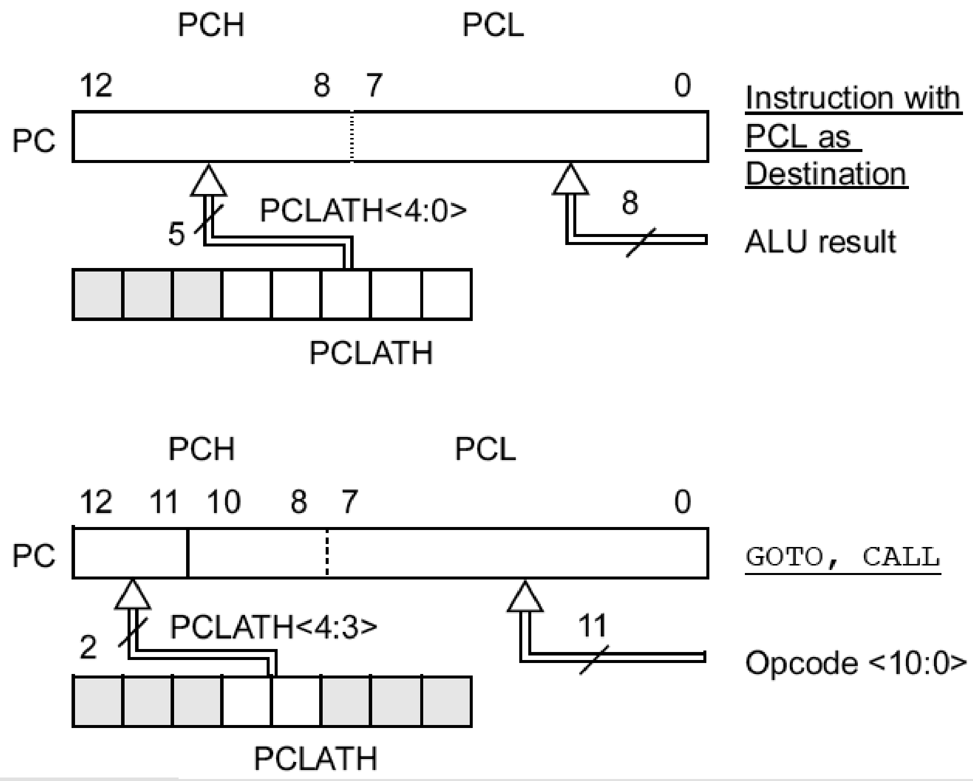
bit 2 **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/$\overline{\text{borrow}}$ bit (ADDWF, ADDLW,SUBLW,SUBWF instructions)
For borrow, the polarity is reversed.
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/$\overline{\text{borrow}}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)
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

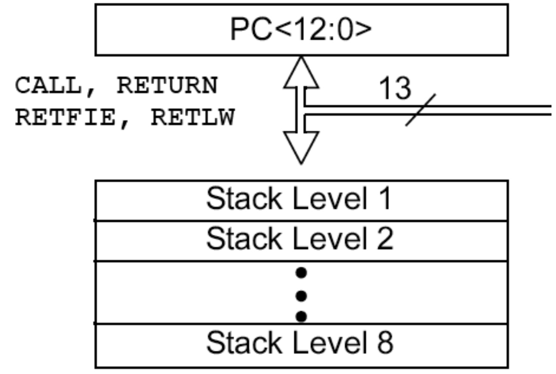
Note: For $\overline{\text{borrow}}$ the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low order bit of the source register

PCL and PCLATH



- PC: Program Counter, 13 bits
- PCL (02h): 8 bits, the lower 8 bits of PC
- PCLATH (0Ah): PC Latch, provides the upper 5 (or 2) bits of PC when PCL is written to
- 1st example: PC is loaded by writing to PCL
- 2nd example: PC is loaded during a CALL or GOTO instruciton

Stack



- 8-level deep x 13-bit wide hardware stack
- The stack space is not part of either program or data space and the stackpointer 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.
- However, NO PUSH or POP instructions !
- PCLATH is not affected by a “PUSH” or “POP” operation.
- The stack operates as a circular buffer:
  - after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push.