Introducing the PIC 16 Series and the 16F84A

Chapter 2
Sections 1-8

Dr. Iyad Jafar
Outline

- Overview of the PIC 16 Series
- An Architecture Overview of the 16F84A
- The 16F84A Memory Organization
- Memory Addressing
- Some Issues of Timing
- Power-up and Reset
- The 16F84A On-chip Reset Circuit
Overview of the PIC 16 Series

- The PIC 16 series is classified as a mid range microcontroller.

- The series has different members all built around the same core and instruction set, but with different memory, I/O features, and package size.
<table>
<thead>
<tr>
<th>Device number</th>
<th>No. of pins*</th>
<th>Clock speed</th>
<th>Memory (K = Kbytes, i.e. 1024 bytes)</th>
<th>Peripherals/special features</th>
</tr>
</thead>
<tbody>
<tr>
<td>16F84A</td>
<td>18</td>
<td>DC to 20 MHz</td>
<td>1K program memory, 68 bytes RAM, 64 bytes EEPROM</td>
<td>1 8-bit timer, 1 5-bit parallel port, 1 8-bit parallel port</td>
</tr>
<tr>
<td>16LF84A</td>
<td>As above</td>
<td>As above</td>
<td>As above</td>
<td>As above, with extended supply voltage range</td>
</tr>
<tr>
<td>16F84A-04</td>
<td>As above</td>
<td>DC to 4 MHz</td>
<td>As above</td>
<td>As above</td>
</tr>
<tr>
<td>16F873A</td>
<td>28</td>
<td>DC to 20 MHz</td>
<td>4K program memory 192 bytes RAM, 128 bytes EEPROM</td>
<td>3 parallel ports, 3 counter/timers, 2 capture/compare/PWM modules, 2 serial communication modules, 5 10-bit ADC channels, 2 analog comparators</td>
</tr>
<tr>
<td>16F874A</td>
<td>40</td>
<td>DC to 20 MHz</td>
<td>4K program memory 192 bytes RAM, 128 bytes EEPROM</td>
<td>5 parallel ports, 3 counter/timers, 2 capture/compare/PWM modules, 2 serial communication modules, 8 10-bit ADC channels, 2 analog comparators</td>
</tr>
<tr>
<td>16F876A</td>
<td>28</td>
<td>DC to 20 MHz</td>
<td>8K program memory 368 bytes RAM, 256 bytes EEPROM</td>
<td>3 parallel ports, 3 counter/timers, 2 capture/compare/PWM modules, 2 serial communication modules, 5 10-bit ADC channels, 2 analog comparators</td>
</tr>
<tr>
<td>16F877A</td>
<td>40</td>
<td>DC to 20 MHz</td>
<td>8K program memory 368 bytes RAM, 256 bytes EEPROM</td>
<td>5 parallel ports, 3 counter/timers, 2 capture/compare/PWM modules, 2 serial communication modules, 8 10-bit ADC channels, 2 analog comparators</td>
</tr>
</tbody>
</table>

*For DIP package only.
An Architecture Overview of the 16F84A

- 18 Pins / DC to 20MHz / 1K program Memory / 68 Bytes of RAM / 64 Bytes of EEPROM / 1 8-bit Timer / 1 5-bit Parallel Port / 1 8-bit Parallel Port
An Architecture Overview of the 16F84A

[Diagram of the 16F84A architecture, showing blocks and connections such as FLASH program memory, 8-level stack (13-DI), RAM file registers, and various components like TMR0, I/O ports, and power-up timer.]
The PIC 16F84A ALU and Working Register

**Arithmetic & Logic Unit**
- 8-bit ALU
- Supports 35 simple instructions
- Input operands are
  - The working register
  - Content of some file register or a literal
- The result is stored in Working register or in a File register

**The Working Register**
- Inside the CPU
- For many instructions, it can be chosen to hold the result of the last instruction executed by the CPU
The PIC 16F84A Memory Organization

- **Program Memory and Related Units**

- 16 Series instructions which invoke the Stack
- The Interrupt Service Routine must start here
- Program Counter
- The program must start here
- Program Counter points to locations in program memory
- Unimplemented memory space, still addressable by the 16F84A program
The PIC 16F84A Memory Organization

- **Program Memory**
  - 1K x 14 Bits
  - Address range 0000H – 03FFH
  - Flash (nonvolatile)
  - 10000 erase/write cycles
  - Location 0000H is reserved for the reset vector
  - Location 0004H is reserved for the Interrupt Vector

- **Program Counter**
  - Holds the address of the instruction to be executed (next instruction)

- **Stack**
  - 8 levels (each is 13 bits)
  - SRAM (volatile)
  - Used to store the return address with instruction like CALL, RETURN, RETFIE, and RETLW (interrupts and subroutines)

- **Instruction Register**
  - Holds the instruction being executed
The PIC 16F84A Memory Organization

- **The Configuration Word**
  - A special part of the program memory
  - Allows the user to configure different features of the microcontroller at the time of program download and is not accessible within the program or while it is running.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP</td>
<td>Code Protection bit</td>
<td>1 = Code protection disabled, 0 = All program memory is code protected</td>
</tr>
<tr>
<td>PWRTE</td>
<td>Power-up Timer Enable bit</td>
<td>1 = Power-up Timer is disabled, 0 = Power-up Timer is enabled</td>
</tr>
<tr>
<td>WDTE</td>
<td>Watchdog Timer Enable bit</td>
<td>1 = WDT enabled, 0 = WDT disabled</td>
</tr>
<tr>
<td>FOSC1: FOSC0</td>
<td>Oscillator Selection bits</td>
<td>11 = RC oscillator, 10 = HS oscillator, 01 = XT oscillator, 00 = LP oscillator</td>
</tr>
</tbody>
</table>
The PIC 16F84A Memory Organization

Data Memory and Special Function Registers (SFRs)
- SRAM (volatile)
- Banked addressing

- Special Function Registers (SFRs)
  - Locations 01H-0BH in bank 0 and 81H-8BH in bank 1
  - Used to communicate with I/O and control the microcontroller operation
  - Some of them hold I/O data

- General Purpose Registers
  - Addresses 0CH – 4FH (68 Bytes)
  - Used for storing general data
## The PIC 16F84A Memory Organization

### Special Function Registers (SFRs)

<table>
<thead>
<tr>
<th>Address</th>
<th>Bank 0</th>
<th>Bank 1</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>INF</td>
<td>80h</td>
<td></td>
</tr>
<tr>
<td>01h</td>
<td>TMRO</td>
<td>OPTION_REG</td>
<td>81h</td>
</tr>
<tr>
<td>02h</td>
<td>PCL</td>
<td>82h</td>
<td></td>
</tr>
<tr>
<td>03h</td>
<td>STATUS</td>
<td>83h</td>
<td></td>
</tr>
<tr>
<td>04h</td>
<td>FSR</td>
<td>84h</td>
<td></td>
</tr>
<tr>
<td>05h</td>
<td>PORTA</td>
<td>TRISA</td>
<td>85h</td>
</tr>
<tr>
<td>06h</td>
<td>PORTB</td>
<td>TRISB</td>
<td>86h</td>
</tr>
<tr>
<td>07h</td>
<td>Unimplemented</td>
<td>87h</td>
<td></td>
</tr>
<tr>
<td>08h</td>
<td>EEDATA</td>
<td>EECON1</td>
<td>88h</td>
</tr>
<tr>
<td>09h</td>
<td>EADDR</td>
<td>EECON2</td>
<td>89h</td>
</tr>
<tr>
<td>0Ah</td>
<td>PCLATH</td>
<td>8Ah</td>
<td></td>
</tr>
<tr>
<td>0Bh</td>
<td>INTCON</td>
<td>8Bh</td>
<td></td>
</tr>
<tr>
<td>0Ch - 4Fh</td>
<td>GPR</td>
<td>8Ch - CFh</td>
<td></td>
</tr>
</tbody>
</table>

- **INDF**: Data memory contents by indirect addressing
- **TMR0**: Timer counter
- **PCL**: Low order 8 bits of program counter
- **STATUS**: Flag of calculation result
- **FSR**: Indirect data memory address pointer
- **PORTA**: PORTA DATA I/O
- **PORTB**: PORTB DATA I/O
- **EEPROM**: Data for EEPROM
- **EEDATA**: Address for EEPROM
- **EADDR**: Write buffer for upper 5 bits of the program counter
- **INTCON**: Interruption control
- **OPTIN_REG**: Mode set
- **TRISA**: Mode set for PORTA
- **TRISB**: Mode set for PORTB
- **EECON1**: Control Register for EEPROM
- **EECON2**: Write protection Register for EEPROM
The PIC 16F84A Memory Organization

- Special Function Registers (SFRs) interacting with peripherals
The PIC 16F84A Memory Organization

- **Data Memory Addressing**
  - For PIC 16F84A, the address of any memory location (File Register is 8 bits)
    - One bit is used to select the bank
    - Seven bits to select a location in the bank
  - Bank selection is done through using bits 5 and 6 of the STATUS registers (RP0 and RP1)
  - For the 16F84A, only RP0 is needed since we have two banks
  - In general, two forms to address the RAM (File Registers)
    - **Direct addressing** – the 7-bit address is part of the instruction
    - **Indirect addressing**
      - the 7-bit address is loaded in lower 7 bits of the *File Select Register* (FSR, 04H)
      - Bank selection is done using the most significant bit of FSR and the *IRP bit* in the STATUS register
The PIC 16F84A Memory Organization

- Data Memory Addressing
The PIC 16F84A Memory Organization

- **The STATUS Register (03H, 83H)**

<table>
<thead>
<tr>
<th>Bit 7-6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimplemented: Maintain as ‘0’</td>
<td>RP0: Register Bank Select bits (used for direct addressing)</td>
<td>TO: Time-out bit</td>
<td>PD: Power-down bit</td>
<td>Z: Zero bit</td>
<td>DC: Digit carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for borrow, the polarity is reversed)</td>
<td>C: Carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for borrow, the polarity is reversed)</td>
</tr>
</tbody>
</table>

**Note:** 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.
The PIC 16F84A Memory Organization

- **Data Related**
  - **EEPROM Data Memory**
    - 64 bytes Non-volatile
    - 10,000,000 erase/write cycles
    - Used to store data that is likely to be needed for long term
    - Operation is controlled through EEDATA (08H), EEADR (09H), EECON1 (88H), and EECON2 (89H) SFRs
  - **To read a location**
    - store the address in EEADR and set the RD bit in EECON1
    - data is copied to EEDATA register
  - **To write to a location**
    - data and address are placed in EEDATA and EEADR, respectively
    - enable writing by setting the WREN bit in EECON1 SFR
    - store 55H then AAH in EECON2
    - commit writing by enabling the WR bit
    - Once the write is done, the EEIF flag is set in EECON1.
### The PIC 16F84A Memory Organization

- **The EECON1 Register (88H)**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Unimplemented: Read as '0'</td>
</tr>
</tbody>
</table>
| 4   | **EEIF**: EEPROM Write Operation Interrupt Flag bit  
   | 1 = The write operation completed (must be cleared in software)  
   | 0 = The write operation is not complete or has not been started  |
| 3   | **WRERR**: EEPROM Error Flag bit  
   | 1 = A write operation is prematurely terminated  
   | (any MCLR Reset or any WDT Reset during normal operation)  
   | 0 = The write operation completed  |
| 2   | **WREN**: EEPROM Write Enable bit  
   | 1 = Allows write cycles  
   | 0 = Inhibits write to the EEPROM  |
| 1   | **WR**: Write Control bit  
   | 1 = Initiates a write cycle. The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.  
   | 0 = Write cycle to the EEPROM is complete  |
| 0   | **RD**: Read Control bit  
   | 1 = Initiates an EEPROM read RD is cleared in hardware. The RD bit can only be set (not cleared) in software.  
   | 0 = Does not initiate an EEPROM read  |
Some Issues of Timing

- **The Clock**
  - The microcontroller is made up of combinational and sequential logic. Thus, it requires a clock!
  - Clock – a continuously running fixed frequency logic square wave
  - Timers, counters, serial communication functions are also dependent on the clock
  - Operating frequency has direct impact on power consumption
  - Every microcontroller has a range for its clock
Some Issues of Timing

- **Instruction Cycle**
  - The main clock is divided by a fixed value (4 in the 16 series) into a lower-frequency signal.
  - The cycle time of this signal is called the *instruction cycle*.
  - The primary unit of time in the action of processor.

<table>
<thead>
<tr>
<th>Clock frequency</th>
<th>Instruction cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Frequency</td>
</tr>
<tr>
<td>20 MHz</td>
<td>5 MHz</td>
</tr>
<tr>
<td>4 MHz</td>
<td>1 MHz</td>
</tr>
<tr>
<td>1 MHz</td>
<td>250 kHz</td>
</tr>
<tr>
<td>32.768 kHz</td>
<td>8.192 kHz</td>
</tr>
</tbody>
</table>
Some Issues of Timing

- **Pipelining**
  - Every instruction in the computer has to be fetched from memory and then executed. These steps are usually performed one after another.
  - The CPU can be designed to fetch the next instruction while executing the current instruction. This improves performance significantly!
  - This is called *Pipelining*
  - All PIC microcontrollers implement pipelining (RISC+Harvard make it easy)
Power-up and Reset

- On power-up, the microcontroller must start to execute the program stored in the program memory from its beginning *(address 0000H)*

- A specialized circuit inside the microcontroller detects this and is responsible for putting the microcontroller in the *reset state*:
  - the program counter is set to zero
  - the SFRs are set such that the peripherals are in safe and disabled

- Another way to put the microcontroller in the reset state is to apply logic zero to the Master Clear input (MCLR)

- Some reset circuit configurations
The 16F84A on-Chip Reset Circuit

Logic ‘1’ on this activates the Reset Signal

Logic ‘1’ on this input deactivates the Reset Signal and causes the microcontroller to exit the reset state
The 16F84A on-Chip Reset Circuit

Example on reset timing when MCLR is connected to VDD
Summary

- The PIC 16F84A series is a diverse and cost effective family of microcontrollers.
- The PIC 16F84A is pipelined RISC processor with Harvard architecture.
- The PIC 16F84A has three different memory types.
- An important memory area is the Special Function Register area which act as link between the CPU and peripherals.
- Reset operation must be understood for proper operation of the microcontroller.