EMBEDDED SYSTEMS

TRAFFIC LIGHT

Presented By

KEROLOS HANY SALAMA

SEC:3 B.N:41

FROM DATA SHEET:

A) Describe all the pins of PIC16f877A:

The PIC16F877A microcontroller comes in a 40-pin DIP package. Here is a description of each pin:

Power Supply Pins

1. **VDD (Pin 11 and Pin 32)**: Positive supply voltage. Typically +5V.
2. **VSS (Pin 12 and Pin 31)**: Ground reference voltage.

Oscillator Pins

3. **OSC1/CLKIN (Pin 13)**: Oscillator crystal or external clock input.
4. **OSC2/CLKOUT (Pin 14)**: Oscillator crystal or clock output. When used with a crystal, this pin connects to the other terminal of the crystal.

Reset Pin

5. **MCLR (Pin 1)**: Master Clear (Reset) input. An active-low pin used to reset the microcontroller.

Port A Pins (RA0-RA7)

6. **RA0/AN0 (Pin 2)**: Analog input channel 0.
7. **RA1/AN1 (Pin 3)**: Analog input channel 1.
8. **RA2/AN2/VREF- (Pin 4)**: Analog input channel 2 or negative voltage reference input.
9. **RA3/AN3/VREF+ (Pin 5)**: Analog input channel 3 or positive voltage reference input.
10. **RA4/T0CKI (Pin 6)**: Digital I/O or Timer0 external clock input.
11. **RA5/AN4/SS (Pin 7)**: Analog input channel 4 or Slave Select for SPI communication.

Port B Pins (RB0-RB7)

12. **RB0/INT (Pin 33)**: Digital I/O or external interrupt input.
13. **RB1 (Pin 34)**: Digital I/O.
14. **RB2 (Pin 35)**: Digital I/O.
15. **RB3/PGM (Pin 36)**: Digital I/O or low voltage programming input.
16. **RB4 (Pin 37)**: Digital I/O.
17. **RB5 (Pin 38)**: Digital I/O.

18. **RB6/PGC (Pin 39)**: Digital I/O or In-Circuit Debugging clock.
19. **RB7/PGD (Pin 40)**: Digital I/O or In-Circuit Debugging data.

Port C Pins (RC0-RC7)

20. **RC0/T1OSO/T1CKI (Pin 15)**: Digital I/O, Timer1 oscillator output, or Timer1 clock input.
21. **RC1/T1OSI/CCP2 (Pin 16)**: Digital I/O, Timer1 oscillator input, or Capture/Compare/PWM module 2.
22. **RC2/CCP1 (Pin 17)**: Digital I/O or Capture/Compare/PWM module 1.
23. **RC3/SCK/SCL (Pin 18)**: Digital I/O, SPI clock, or I2C clock.
24. **RC4/SDI/SDA (Pin 23)**: Digital I/O, SPI data input, or I2C data.
25. **RC5/SDO (Pin 24)**: Digital I/O or SPI data output.
26. **RC6/TX/CK (Pin 25)**: Digital I/O or USART transmit.
27. **RC7/RX/DT (Pin 26)**: Digital I/O or USART receive.

Port D Pins (RD0-RD7)

28. **RD0/PSP0 (Pin 19)**: Digital I/O or Parallel Slave Port data bit 0.
29. **RD1/PSP1 (Pin 20)**: Digital I/O or Parallel Slave Port data bit 1.
30. **RD2/PSP2 (Pin 21)**: Digital I/O or Parallel Slave Port data bit 2.
31. **RD3/PSP3 (Pin 22)**: Digital I/O or Parallel Slave Port data bit 3.
32. **RD4/PSP4 (Pin 27)**: Digital I/O or Parallel Slave Port data bit 4.
33. **RD5/PSP5 (Pin 28)**: Digital I/O or Parallel Slave Port data bit 5.
34. **RD6/PSP6 (Pin 29)**: Digital I/O or Parallel Slave Port data bit 6.
35. **RD7/PSP7 (Pin 30)**: Digital I/O or Parallel Slave Port data bit 7.

Port E Pins (RE0-RE2)

36. **RE0/RD/AN5 (Pin 8)**: Digital I/O, Read control for Parallel Slave Port, or Analog input channel 5.
37. **RE1/WR/AN6 (Pin 9)**: Digital I/O, Write control for Parallel Slave Port, or Analog input channel 6.
38. **RE2/CS/AN7 (Pin 10)**: Digital I/O, Chip Select for Parallel Slave Port, or Analog input channel 7.

B) Explain the functions of the main blocks in PIC16F877A: ALU, Status and Control, Program Counter, Flash Program Memory, Instruction Register, and Instruction Decoder:

1. Arithmetic Logic Unit (ALU)

- **Function:** The ALU performs arithmetic and logical operations. It handles operations such as addition, subtraction, AND, OR, XOR, and bit shifts.
- **Details:** It is the core computational unit of the microcontroller, taking input from the registers and performing the necessary operations to process data or control functions.

2. Status and Control Registers

- **Function:** These registers hold the status of the ALU operations and control the operation of the microcontroller.
- **Details:**
    - **STATUS Register:** Contains flags that indicate the result of the last ALU operation (e.g., Zero, Carry, Digit Carry, and Overflow flags). These flags are used for conditional branching and decision-making in programs.
    - **CONTROL Registers:** Control various aspects of the microcontroller's operation, such as enabling/disabling interrupts, configuring I/O ports, and setting operating modes.

3. Program Counter (PC)

- **Function:** The Program Counter holds the address of the next instruction to be executed.
- **Details:**
    - The PC is incremented automatically after fetching an instruction, ensuring the sequential execution of instructions.
    - It can be modified directly for branching and jump instructions, allowing for loops and conditional execution in programs.

4. Flash Program Memory

- **Function:** This is non-volatile memory that stores the program code.
- **Details:**
    - The PIC16F877A has 14-bit wide Flash memory for storing instructions.

  - It retains the program even when the power is turned off, allowing the microcontroller to retain its functionality upon power-up.
  - The size of Flash memory in PIC16F877A is 8KB (8192 words).

5. Instruction Register

- **Function:** Temporarily holds the current instruction fetched from the program memory.
- **Details:**
  - The instruction register receives the 14-bit instruction code from the program memory.
  - It provides the instruction code to the instruction decoder for decoding and execution.

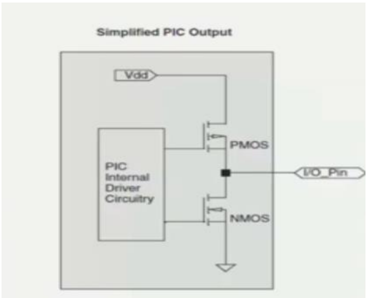
6. Instruction Decoder

- **Function:** Decodes the instruction fetched into the instruction register and generates the necessary control signals to execute the instruction.
- **Details:**
  - The instruction decoder interprets the binary instruction code and translates it into control signals.
  - These control signals orchestrate the operation of various parts of the microcontroller, such as the ALU, data memory, and I/O ports, to perform the desired operation.

Summary of Functions

- **ALU**: Performs arithmetic and logical operations.
- **Status and Control Registers**: Store operation results and control the microcontroller's operation.
- **Program Counter:** Tracks the address of the next instruction.
- **Flash Program Memory:** Stores the program code.
- **Instruction Register:** Temporarily holds the current instruction.
- **Instruction Decoder:** Decodes and executes instructions by generating control signals.

C) Why led connected in RA4 for flashing prepose not working:

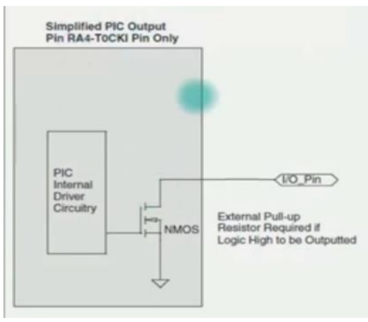


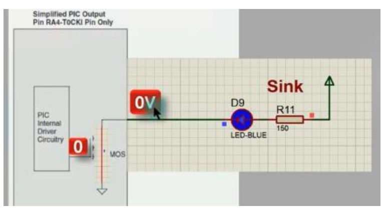
Configration of pins of port A and port E except RA4

PMOS will work when the out of pic is one and the out of pin is one because PMOS connect with Vdd(5v) and NMOS does not work

NMOS will work when the out of pic is zero and the out of pin become zero because NMOS connect with ground and PMOS does not work

When we connet led as a scource or sink will work





Configration of RA4 pin

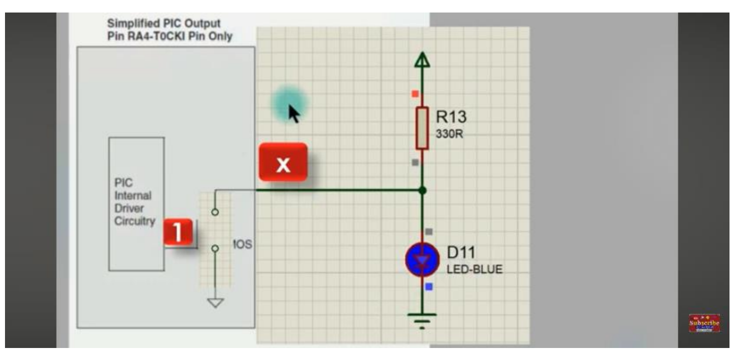
PMOS is not here

NMOS will work when the out of pic is zero and the out of pin become zero because NMOS connect with ground

When we connet led as a scource led dose not work (no flashing) because when the out of pic is one NMOS dose not work and PMOS is not here and when the out of pic is zero the NMOS will work and the voltage of cathode=volatage of anode = zero

When we connet led as a sink led will work (flashing) because when the out of pic is one NMOS dose not work and PMOS is not here and the (open circuit) and when the out of pic is zero the NMOS will work and the voltage of cathode<volatage of anode and the led will work

We can make led work when it connected as a source by connecting the led with ground and connecting the resistor with power as the following:



D) ATMega328P VS PIC16F877A:

1. Memory Size

- **ATMega328P:**
  - **Flash Program Memory:** 32KB
  - **SRAM:** 2KB
  - **EEPROM:** 1KB
- **PIC16F877A:**
  - **Flash Program Memory:** 8KB
  - **SRAM:** 368 Bytes
  - **EEPROM:** 256 Bytes

Winner: ATMega328P (larger memory sizes across all types).

2. Power Consumption

- **ATMega328P:**
  - **Operating Voltage:** 1.8V to 5.5V
  - **Active Mode:** 0.2 mA at 1 MHz, 1.8V
  - **Power-down Mode:** 0.1 μA at 1.8V
- **PIC16F877A:**
  - **Operating Voltage:** 2.0V to 5.5V
  - **Active Mode:** 1.6 mA at 5V, 4 MHz
  - **Power-down Mode:** 1 μA at 2V

Winner: ATMega328P (lower power consumption).

3. Pin Count

- **ATMega328P:** 28 Pins (DIP package)
- **PIC16F877A:** 40 Pins (DIP package)

Winner: Depends on application needs (more pins for more I/O options vs fewer pins for smaller footprint).

4. Clock Speed

- **ATMega328P:** Up to 20 MHz
- **PIC16F877A**: Up to 20 MHz

Winner: Tie (both have the same maximum clock speed).

5. Peripherals

- **ATMega328P:**
  - 1 USART
  - 1 I2C
  - 1 SPI
  - 6 PWM Channels
  - 8 ADC Channels (10-bit)
- **PIC16F877A:**
  - 1 USART
  - 1 I2C
  - 1 SPI
  - 2 PWM Channels
  - 8 ADC Channels (10-bit)

Winner: ATMega328P (more PWM channels).

6. Development Ecosystem

- **ATMega328P:** Widely supported by the Arduino ecosystem, large community**,** numerous libraries, and development tools.
- **PIC16F877A:** Supported by MPLAB X IDE, fewer community libraries, and examples compared to Arduino.

Winner: ATMega328P (more beginner-friendly with Arduino support).

Examples of Embedded Systems where ATMega328P is a Better Choice

Example 1: Low-Power Battery-Powered Sensor Node

- **Reason:** The ATMega328P has lower power consumption, which is crucial for battery-powered applications. It also has more SRAM and EEPROM, which can be beneficial for data logging and complex sensor data processing.

Example 2: Small Footprint Embedded System

- **Reason:** The ATMega328P has fewer pins, making it suitable for compact designs. The extensive support from the Arduino ecosystem simplifies development and prototyping, which is advantageous for quick and iterative design processes in smaller embedded systems.

Summary

- **ATMega328P:** Better suited for applications requiring larger memory, lower power consumption, and extensive community support (e.g., battery-powered sensor nodes, compact designs).
- **PIC16F877A:** Suitable for applications requiring more I/O pins and simpler, more traditional development environments (e.g., larger systems with more peripherals).