Lecture 13

Pulse Width Modulation,
Analog to Digital Converter

Topics

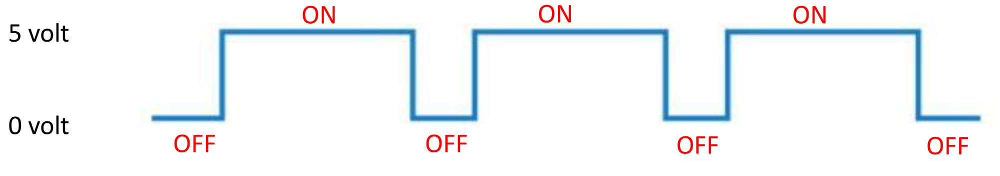
- Pulse Width Modulation
- Analog to Digital Converter

Pulse Width Modulation

- A digital signal can be only 0 volt (logical 0) or 5 volt (logical 1).
- But an analog signal can be any value between 0 and 5 volts.
- Pulse Width Modulation (PWM) is a simulation method for getting analog output results by using digital output signals.
- Examples of PWM usage:
  - Controlling brightness of an LED.
  - Controlling rotation speed of a Direct Current (DC) Motor.
  - Controlling temperature of a Heating Resistor.
- PWM can be implemented by hardware (PWM circuit) or by software.

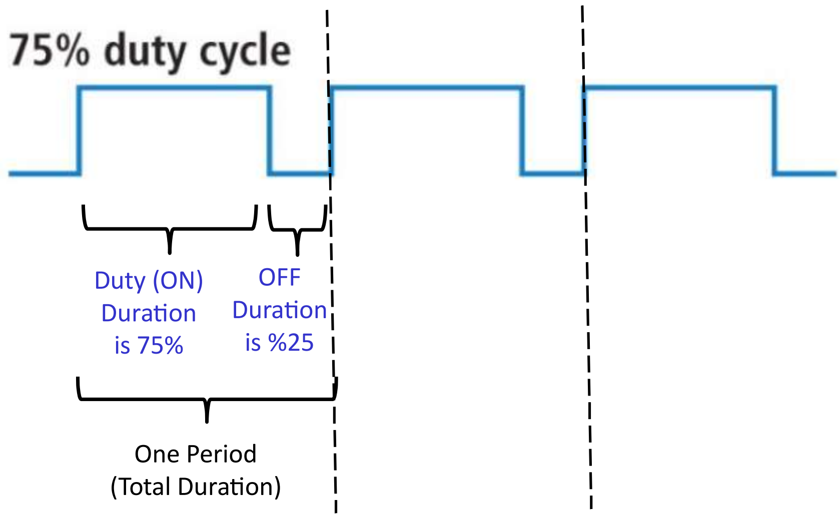
Digital signals in PWM

- Digital signals are square waves that are switched between ON and OFF.
- ON/OFF operations can simulate intermediate voltages between 0 volts and 5 volts.
- The PWM program changes (modulates) the portion of the "ON time" duration and "OFF time" duration in a loop.
- The duration of "ON time" is called the Duty Cycle (Pulse Width).
- To get varying analog output results, the pulse width is changed repeatedly in a loop.



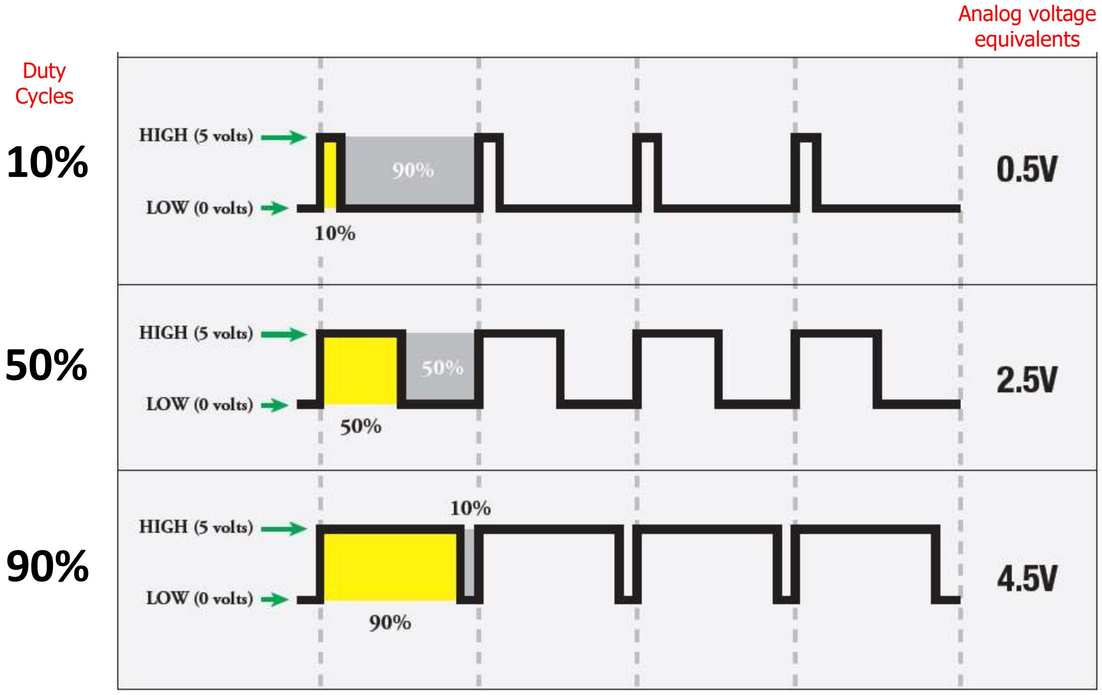
Example : Duty Cycle Duration

Suppose Duty Cycle is = 75%		
Total Duration (One period) = 100 milli seconds		
Duty (ON) Duration	= 75 milli seconds	(Pulse Width)
OFF Duration	= 25 milli seconds	



Examples of PWM Duty Cycles

The following graph shows a comparison of different Duty Cycles and their corresponding analog voltage levels.



Formula of Duty Cycle Duration

- The percentage of the time signal that is HIGH (ON) as the fraction of Total Time is called the Duty Cycle Duration (Pulse Width).

$$\boxed{Duty\ Cycle\ Percentage = \frac{Digital\ Value}{255}\ x\ 100}$$

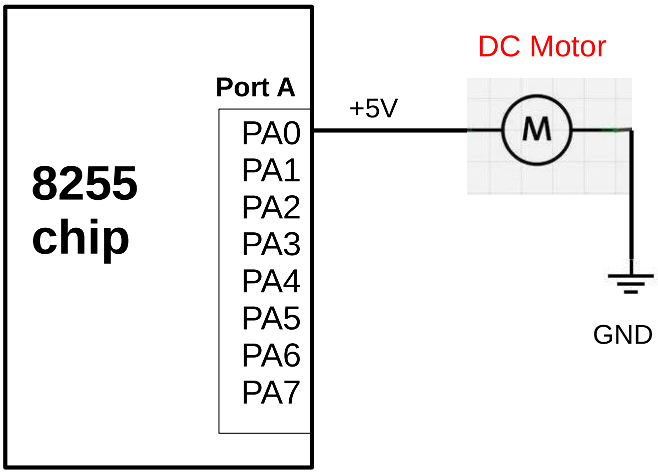
- The 8-bit digital value (0-255) directly controls how long a 5V value (HIGH, ON) is applied to the output period.
- The ON duration is called as the Duty Cycle.

Example: Calculation of analog voltage equivalents

Duty Cycle Percentage	Digital Value	Duty Cycle Formula	Analog Voltage Equivalent
100%	255	255 / 255	5 Volt * 1.00 = 5 Volt
75%	191	191 / 255	5 Volt * 0.75 = 3.75 Volt
50%	127	127 / 255	5 Volt * 0.50 = 2.5 Volt
25%	63	63 / 255	5 Volt * 0.25 = 1.25 Volt
0%	0	0 / 255	5 Volt * 0.00 = 0 Volt

Application Example: Controlling Speed of DC Motor with PWM

- A 5V DC Motor (Direct Current motor) is connected directly to Port A (the PA0 pin) of a 8255 peripheral interface chip.
- Suppose the following port addresses will be used for the 8255 chip.
  Port A address= 00h
  Control Register address= 06h



PWM Application Program

- Write an Assembly program that uses PWM method to control the speed of a DC motor, through the 8255 interface chip.

- In program, define the variables described below.

- TOTAL DURATION : 2 bytes
  - Arbitrary value of Total time (milli seconds) for one period.
  - Initialize with 30000.

- DUTY CYCLE : 1 byte
  - The value (between 0 and 100) will be used to control the speed of motor.
  - Initialize with 75.

- ON DURATION : 2 bytes
- OFF DURATION : 2 bytes
  - The ON and OFF values will be calculated by the percentage formula.

Program (PWM)

- Program uses the PWM method to control the speed of a DC motor.
- The 8255 chip is used as peripheral interface for the DC motor.

Part1

```
;Define the port addresses of 8255 chip.
PortA         EQU   00h
KontrolPort  EQU   06h

.model small
.stack
.data

;Define the variables (PWM parameters).

TOTAL_DURATION   DW   30000
;Arbitrary value of Total time (milli seconds) for one period

ON_DURATION     DW  ?
OFF_DURATION   DW  ?

DUTY_CYCLE    DB   75
;Arbitrary value of Time for Duty (ON)
```


Part2

```
.code
BASLA PROC
.STARTUP

  ;Perform the ON and OFF duration calculations.
  MOV  AX, TOTAL_DURATION   ; Copy variable to AX
  MUL  DUTY_CYCLE           ; Multiply AX by the Duty Cycle
  MOV  BL, 100              ; For percentage calculation
  DIV  BL                   ; Divide AX by 100
  MOV  ON_DURATION, AX      ; Copy register to variable

  MOV  AX, TOTAL_DURATION   ; Copy variable to register
  SUB  AX, ON_DURATION      ; Subtract variable from register
  MOV  OFF_DURATION, AX     ; Copy register to variable

  mov  AL, 10000000b        ; The control word of 8255 chip
  out  KontrolPort, AL      ; Initialize the 8255 chip
```

Calculation Formulas

$$\boxed{ON\ Duration\ =\ \frac{Total\ Duration\ *\ Duty\ Cycle}{100}}$$

$$\boxed{OFF\ Duration\ =\ Total\ Duration\ -\ ON\ Duration}$$

Part3

```
MainDongu:                    ; Endless loop

;Turn on the motor during duty (on) time
  MOV AL, 00100000b      ; PA5 pin bit is 1
  OUT PortA, AL          ; Send to port
  call WAIT_FOR_ON

;Turn off the motor during idle (off) time
  MOV AL, 00000000b      ; PA5 pin bit is 0
  OUT PortA, AL          ; Send to port
  call WAIT_FOR_OFF

  JMP MainDongu           ; Goto endless loop

.EXIT
BASLA ENDP
```


Part4

```
; SUBROUTINES
;------------------------------------------------------------------------------------
WAIT_FOR_ON  PROC
  MOV  CX, ON_DURATION    ; CX is counter for local loop
LokalDongu1:
  NOP
  LOOP LokalDongu1         ; Decrements CX implicitly
RET                        ; Return from procedure
WAIT_FOR_ON ENDP

;------------------------------------------------------------------------------------
WAIT_FOR_OFF  PROC
  MOV  CX, OFF_DURATION   ; CX is counter for local loop
LokalDongu2:
  NOP
  LOOP LokalDongu2         ; Decrements CX implicitly
RET                        ; Return from procedure
WAIT_FOR_OFF ENDP
;------------------------------------------------------------------------------------

END BASLA                  ;End of file
```

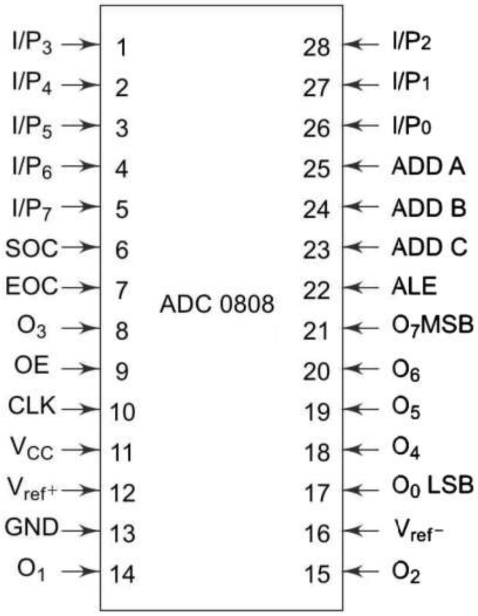
Topics

- Pulse Width Modulation
- Analog to Digital Converter

ADC 0808
Analog to Digital Converter chip

- The ADC 0808 chip is an input device to 8086 microprocessor.
- It is connected to CPU through the 8255 peripheral interface chip.
- CPU sends an initializing signal to the ADC to start the analog signal to digital conversion process.
- The Start of Conversion (SOC) signal is a pulse of a specific duration.
- Analog to digital conversion is a slow process and the processor has to wait for the digital data until the conversion is over.
- After the conversion, the ADC sends the End of Conversion (EOC) signal to inform the processor, and the result is ready at the output buffer of ADC.
- The tasks of issuing SOC pulse to ADC, reading EOC signal from the ADC and reading the digital output of the ADC are carried out by the CPU using 8255 peripheral interface chip ports.

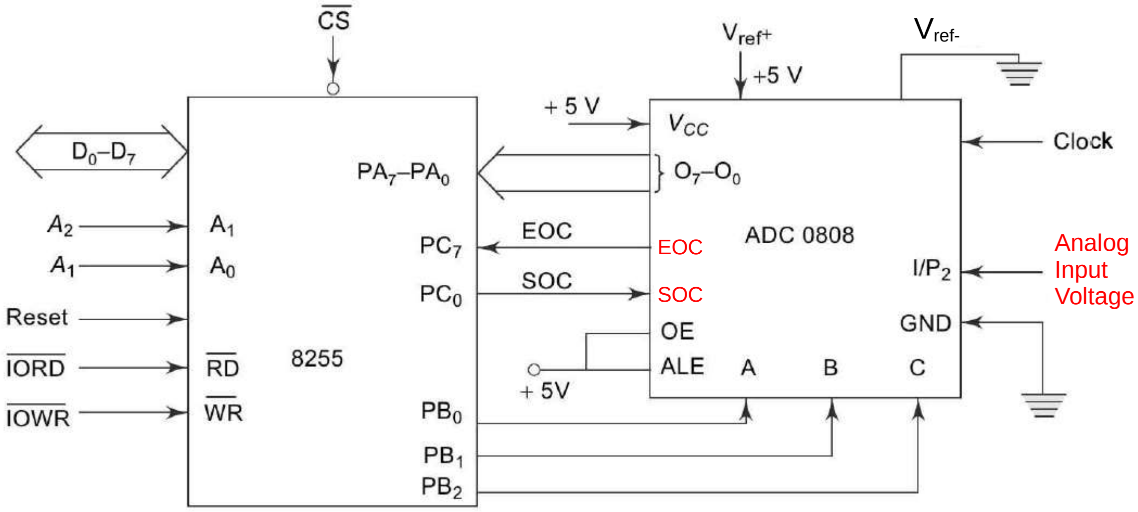
ADC 0808
Analog to Digital Converter chip



Pin Descriptions
of ADC 0808

I/P_0–I/P_7	Analog inputs (8 input channels)
ADD A, B, C	Address lines for selecting analog inputs
O_7–O_0	Digital 8-bit output with O_7 MSB and O_0 LSB
SOC	Start of conversion signal pin
EOC	End of conversion signal pin
OE	Output latch enable pin, if high enable output
CLK	Clock input for ADC
V_{CC}, GND	Supply pins +5V and GND
V_{ref+} and V_{ref-}	Reference voltage positive (+5 Volts maximum) and Reference voltage negative (0V minimum)

Connection diagram of 8255 Chip and ADC 0808 Chip



Control signals:
SOC: Start of Conversion
EOC : End of Conversion

Address Lines: A, B, C

Pin Connections

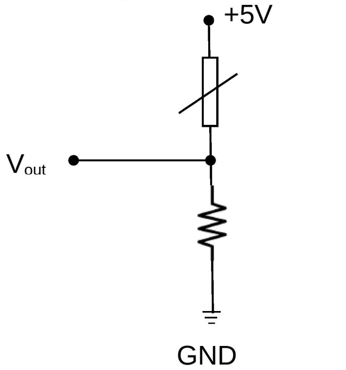
- Pin connections of ADC 0808 :
  - The analog input pin I/P2 is used for analog voltage input.
  - Address pins A, B, C should be 0,1,0 respectively to select the I/P2 pin.
  - The OE pin and ALE pin are at +5V to select the ADC chip and enable the outputs.

- Pin connections of 8255 :
  - PC7 pin acts as the input port to receive the EOC signal.
  - PC0 pin acts as the output port to send SOC to ADC.
  - PA0-PA7 pins act as a 8-bit input data port to receive the digital data output from the ADC.

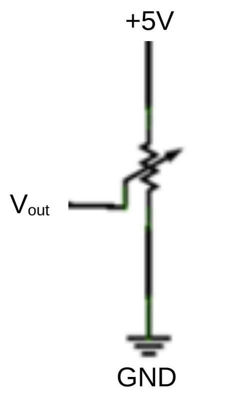
Examples of Analog Devices
(Connected to analog input pins of ADC 0808)

- Temperature Sensor: A temperature sensor (thermistor) will measure the environment temperature, and will send analog voltage to the ADC.
- Potentiometer : It is a variable resistor which can be used to change the voltage in a circuit manually. Its analog voltage reading pin is connected to the input of the ADC. When user moves the handle of potentiometer, its analog voltage reading will increase or decrease (depending on the moving direction of the handle) between 0 and 5 V.

Temperature Sensor



Potentiometer



Analog to Digital Conversion Table

- Input of ADC : Analog input voltage between 0 Volt and +5 Volts.
- Output of ADC : 8-bit digital number between 0 and 255.

ADC Input (Analog input voltage)	ADC Output (8-bit digital number)	Binary Value (Send to 8255 interface port)
0.00	0	0000 0000
0.02	1	0000 0001
0.04	2	0000 0010
0.06	3	0000 0011
0.08	4	0000 0100
. . . .		. . . .
4.98	254	1111 1110
5.00	255	1111 1111

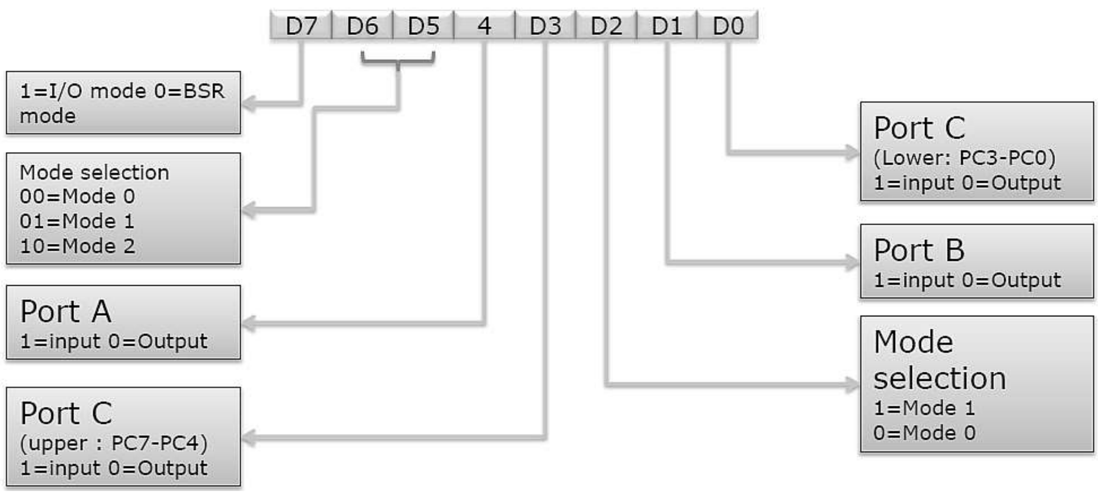
- Precision (P) of converted values are calculated by the formula below.
- Precision is also called as resolution.

$$\boxed{P = \frac{(5-0)}{255} = 0.02}$$

Application Program using 8255 and ADC 0808

- Write an 8086 Assembly program that reads 8-bit digital values from 8255 interface chip.
- Suppose the 8255 chip ports will be used for interfacing with a ADC 0808 chip.
- Use the Port A of 8255 chip for transferring digital data output of ADC to the 8086 CPU.
- Use the Port C of 8255 for control signals.
- Suppose that an analog input is present at the I/P2 pin of ADC.

Control Word for 8255 Interface Chip



Defining the Control Word for 8255 Interface Chip

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	0

Control word = 98h

D7=1; I/O Mode
D6=0 and D5=0; Port A Mode
D4=1; Port A is input port
D3=1; Port C (Upper) is input
D2=0; Port B Mode 0
D1=0; Port B is output
D0=0; Port C (Lower) is output

Program (ADC)

Program reads 8-bit input data thru the 8255 interface chip, from the ADC 0808 chip.

```
; Define the port addresses of 8255 chip.
PortA   EQU   00h
PortB   EQU   02h
PortC   EQU   04h
KontrolPort   EQU   06h
.model small
.code
.STARTUP
mov AL, 98h            ; The control word of 8255 chip
out  KontrolPort, AL   ; Initialize the 8255 chip

mov AL, 02h            ; Select the I/P2 pin of ADC chip as analog input
out  PortB, AL         ; PortB of 8255 is connected to I/P2 pin of ADC chip

mov AL, 00h            ; Send SOC pulse to the ADC (SOC=Start Of Conversion)
out  PortC, AL         ; PortC of 8255 is connected to SOC pin of ADC chip
BEKLE:
IN   AL, PortC         ; Check for EOC signal (EOC=End Of Conversion)
                       ; by reading PortC upper of 8255 chip
RCL  AL,1              ; Rotate AL trough carry
JNC  BEKLE             ; Goto the loop if Not Carry

IN   AL, PortA         ; If EOC signal exists, then read 8-bit equivalent data in AL.
.EXIT
END                    ; End of file
```