Analog-to-Digital Coverter (ADC)

Spring 2019
EE3954: Microprocessors and Microcontrollers

Avinash Karanth
Professor, Electrical Engineering and Computer Science
Ohio University
E-mail: karanth@ohio.edu
Acknowledgment: Harsha Chenji, Jim Goble, Maarten Uijt de Haag

1

1

Course Administration

- Laboratory
  - Lab 5 next week 3/25.
  - Lab 6 the week after 4/1
- Homework and Quiz
  - Hw 4 will be posted 3/22, due on 4/1
  - Quiz 2 will be made available on Friday 3/29
- Lectures
  - Mon (3/25) and Wed (3/27) covered by Jim Goble on ADC and System Engineering
- No class
  - On Friday 3/29 (travel)

2

2

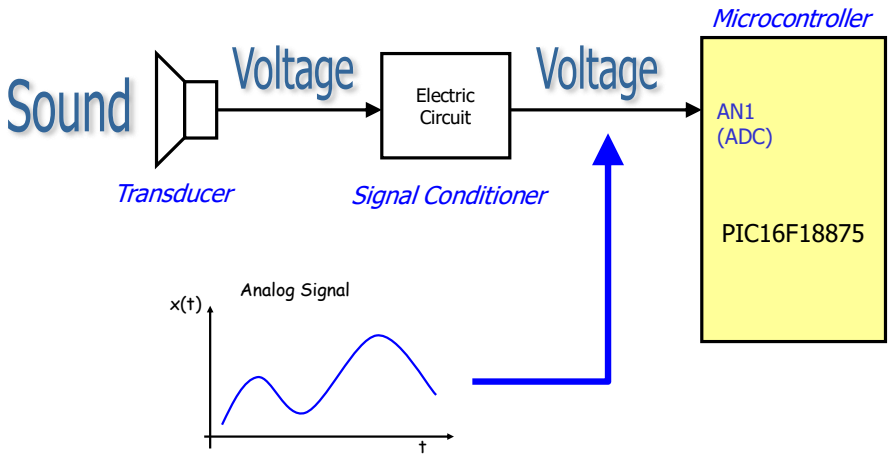
Reference Sections

- Microcontroller 16F18875 Datasheet (DS)
  - Chapter 23 – Analog-to-Digital Converter (ADC)
    - Section 23.1 ADC Configuration
    - Section 23.2 ADC Operation
    - Section 23.3 Acquisition Requirements
    - Section 23.6 Register Definitions

3

3

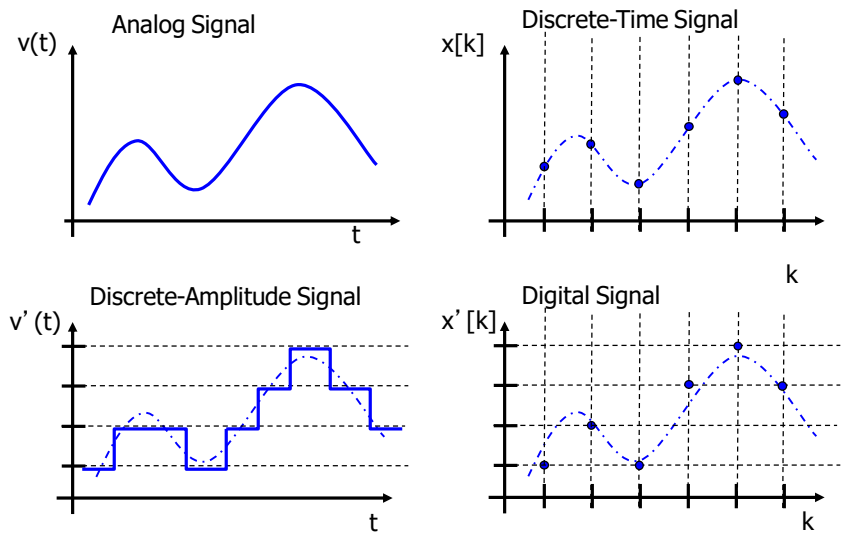
ADC Example



4

4

Analog and Digital Domains



5

5

Nyquist Sampling

There is a limit to the highest frequency that can be represented in a signal when the sampling rate is set at some value. It turns out that for a given sampling rate, call it "SR", the largest frequency component of a signal to be sampled must not exceed the frequency "SR / 2". If that signal *does* have frequencies exceeding "SR / 2", those frequencies will be masked as lower frequencies and will interfere with the valid frequencies that are less than "SR / 2".

These false lower frequencies are called the "alias" frequencies of a sampled signal. Unless these frequency components which are greater than "SR / 2" are removed by an analog filter prior to being sampled, they will be "aliased"; they will interfere with the frequency components which are less than "SR / 2", and there will be no way to remove them once they are sampled!

ADC.6

6

Nyquist Sampling

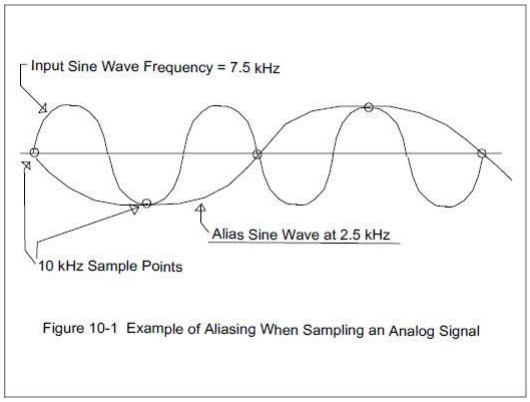


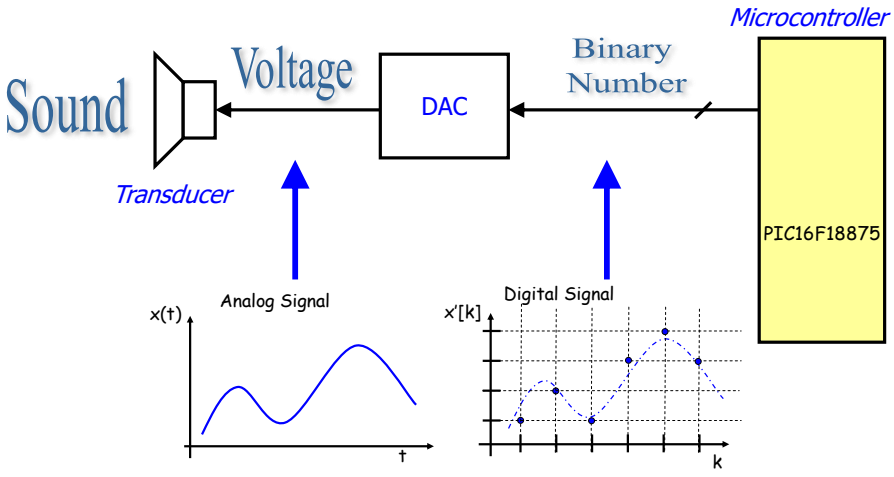
Figure 10-1 Example of Aliasing When Sampling an Analog Signal

- We are sampling a 7.5 KHz signal at 10 KHz in this example. Note that the samples we take cannot be distinguished from 2.5 KHz signal. This is called aliasing. The problem comes when we try to reconstruct the original signal. The DAC is as likely to replace the 7.5 KHz component with a 2.5 KHz one.

- Signals are rarely just a single frequency. So the reconstructed signal will contain the lower frequencies undistorted, but then we will have an overlay of aliased signals.

7

Digital-to-Analog Conversion (DAC)

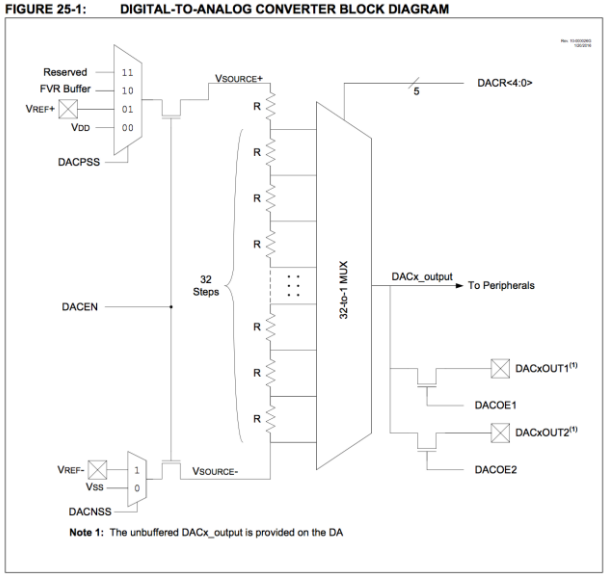


8

8

DAC – PIC16F18875 Has One Built-in

FIGURE 25-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM



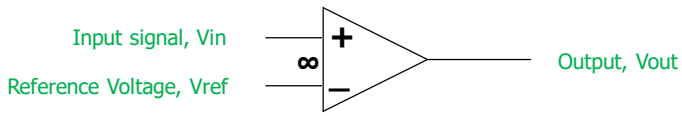
Section 25

9

9

ADC - Comparator

Basic Component:



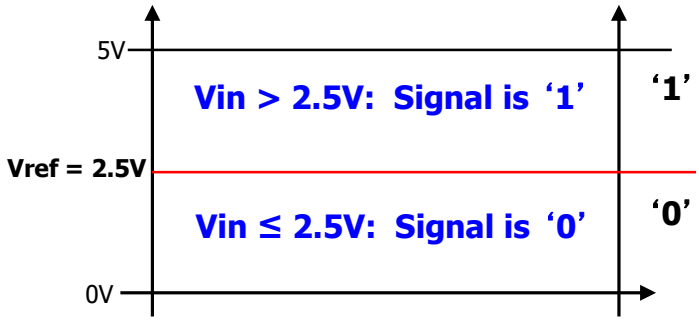
If Vin > Vref , Vout = 5VDC (Logic '1')

If Vin ≤ Vref , Vout = 0VDC (Logic '0')

10

10

Example - 1-bit ADC



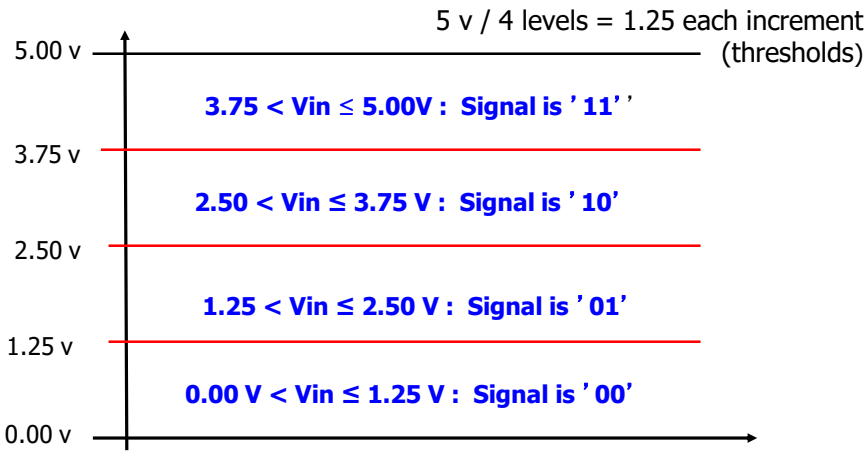
1-bit => 2^1 = 2 levels => 1 threshold

11

11

Example – 2-bit ADC

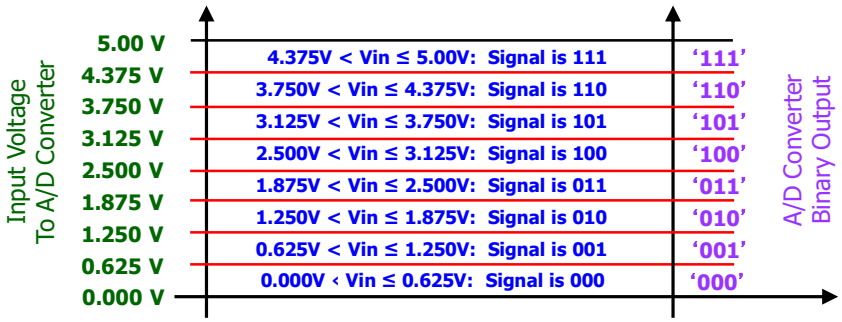
2-bit => 2^2 = 4 levels => 3 thresholds



12

12

Example – 3-bit ADC



3-bits => 2^3 = 8 levels => 7 thresholds (5v/8 = 0.625)

13

13

Example – 3-bit ADC

Examples:	Examples:
Analog Input Voltage:	Digital Output:
2.8756V	100
-0.234V	000
4.9876V	111
1.1V	001
3.2V	101

14

14

Quantization Error

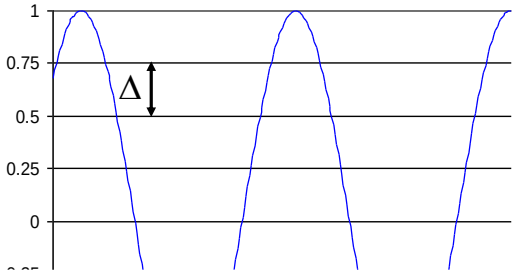
- Difference between actual and sampled value
  - amplitude between [-A, A] (i.e -5V to 5V)
  - quantization levels = N

$$\Delta = \frac{2A}{N}$$

- e.g., if A = 1, N = 8, Δ = 1/4
- In our case we sample from 0 to A

$$\Delta = \frac{A}{N}$$

QE = Δ/2



15

ADC Parameters

- Dynamic range (DR): $V_{input,max} - V_{input,min}$
- Number of levels: 2^n
- Number of thresholds: 2^n-1
- Resolution: $DR/2^n$

In PIC16F18875: $V_{input,max} = V_{ref+}$
$V_{input,min} = V_{ref-}$

16

16

ADC – In PIC16F18875

- 10-bits:
  - 2^{10} = 1024 levels => 1023 thresholds
  - Given default DR = 5V
  - Resolution = 5V/1024 = 0.0048828125 V
  - Thresholds:
    - 0.0048828125 V
    - 0.009765625 V
    - 0.0146484375 V
    - Etc.

17

17

ADC Implementations

18

18

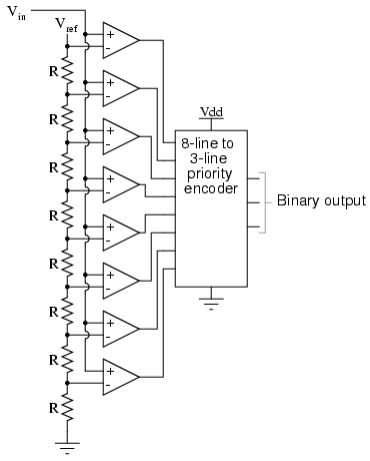
Flash ADC

1. Consists of a series of comparators, each one comparing the input signal to a unique reference voltage.

2. The comparator outputs connect to the inputs of a priority encoder circuit, which produces a binary output

19

Flash ADC Circuit



20

How Flash Works

1. As the analog input voltage exceeds the reference voltage at each comparator, the comparator outputs will sequentially saturate to a high state.
2. The priority encoder generates a binary number based on the highest-order active input, ignoring all other active inputs.

21

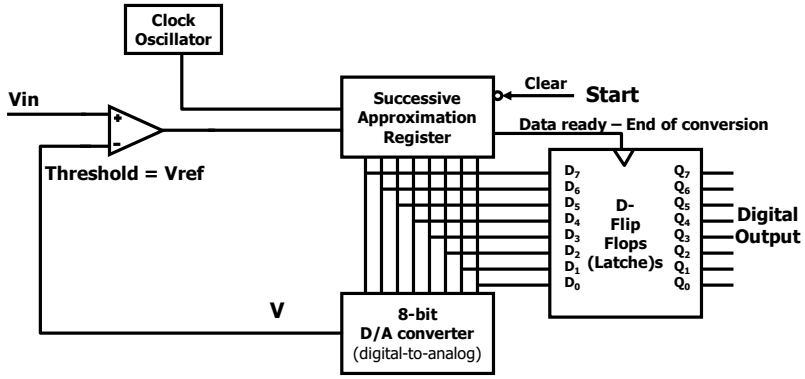
Flash

Advantages	Disadvantages

• Simplest in terms of operational theory
• Most efficient in terms of speed, very fast
  • limited only in terms of comparator and gate propagation delays
	• Lower resolution
	• Expensive
	• For each additional output bit, the number of comparators is doubled
	  • i.e. for 8 bits, 256 comparators needed

22

ADC– Successive Approximation

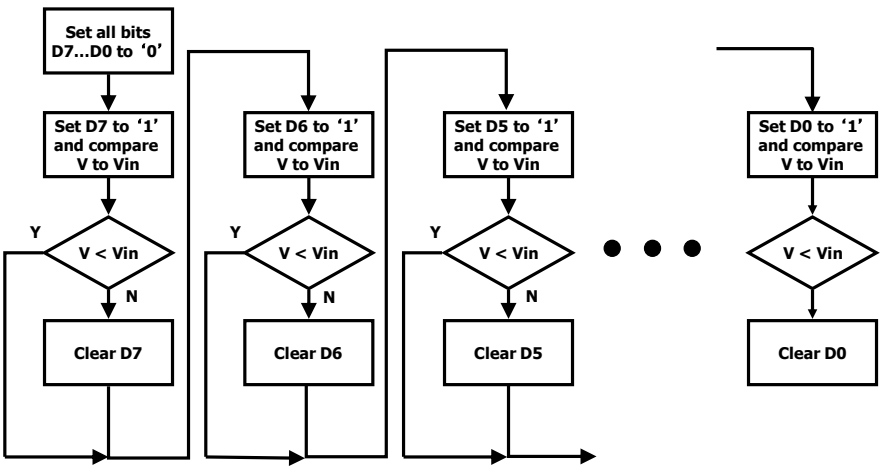


Note: for an *n*-bit ADC it will take *n* clock-cycles to find an output.

23

23

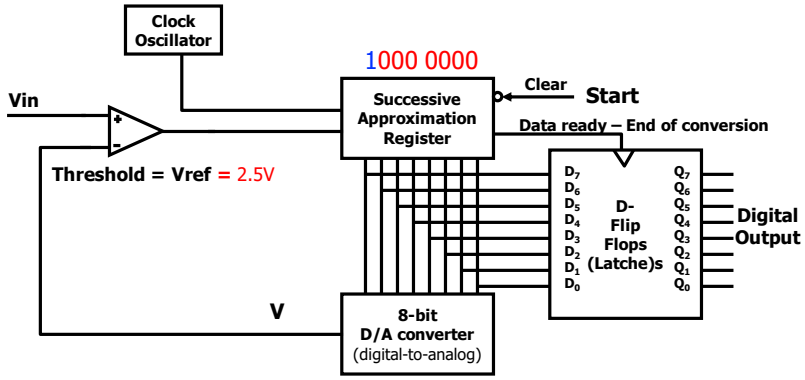
ADC– Successive Approximation



24

24

Example – 4.3V, step 1, bit 7 (D7)

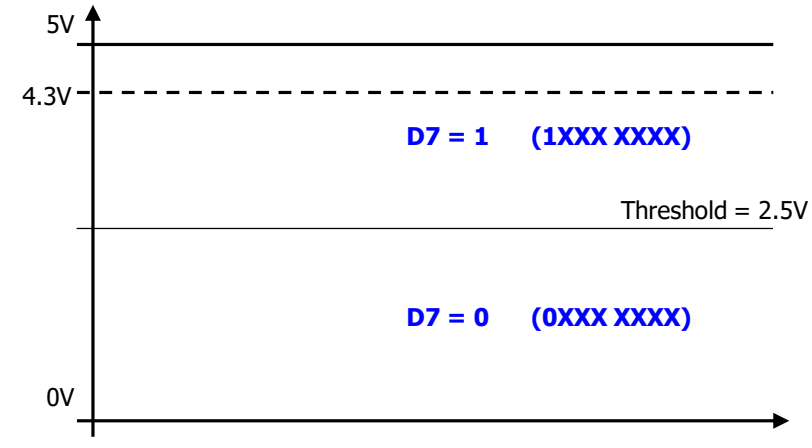


1000 0000 = 128 => (128/256)*5V = 2.5V

25

25

Example – 4.3V, step 1, bit 7 (D7)



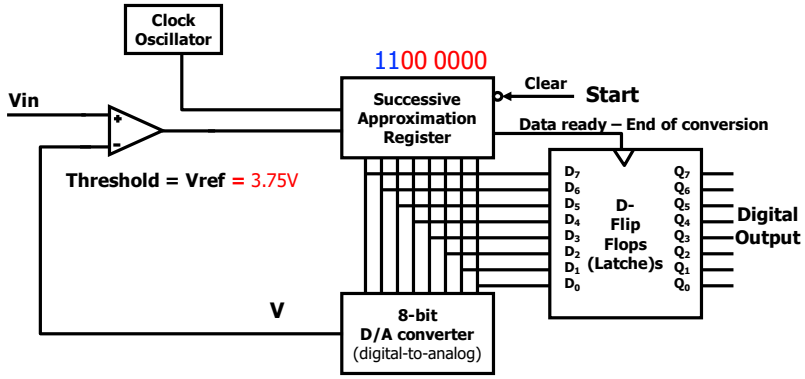
D7 = Most Significant Bit

Result: 1XXX XXXX

26

26

Example – 4.3V, step 2, bit 6 (D6)

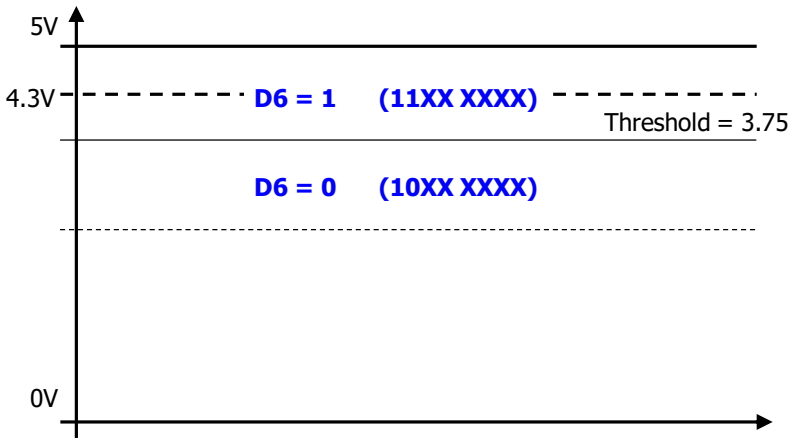


1100 0000 = 192 => (192/256)*5V = 3.75V

27

27

Example – 4.3V, step 2, bit 6 (D6)



D7 = Most Significant Bit

Result: 11XX XXXX

Etc. etc.

28

28

Example

- Voltage input range: 0V – 5V
- Number of bits: 8 => 2^8 = 256 **levels** (= 255 thresholds)
- **Resolution** = 5V / 256 = 0.01953125 V
- **Thresholds** at : 0.01953125 V
  2x0.01953125 V = 0.0390625 V
  3x0.01953125 V = 0.05859375 V
  …
  …
- 4.3V / resolution = 220.16 => 220 = binary value = 1101 1100

29

29

PIC16F18875 ADC

- The PIC16F18875 Analog to Digital Converter has 10-bits of resolution
- We can connect any Port Pin to the input of the ADC. This means we can switch between various sensors.
- It can be configured to automatically perform math functions on the input signal.
  1. Averaging
  2. Low Pass Filter
  3. Oversampling
  4. Threshold Comparison

30

30

How it Works

- The ADC is a successive approximation type converter
- It can theoretically make up to 100KSamp/Sec
- It uses a capacitor as it's sample and hold circuit
- When the GO bit in the ADCON register is set, the sample and hold circuit is disconnected from the input and the conversion begins.
- It takes around 13 AD clock cycles to achieve a conversion.
- The result is stored in the two 8-bit register wide ADRES register (ADRESL and ADRESH).

31

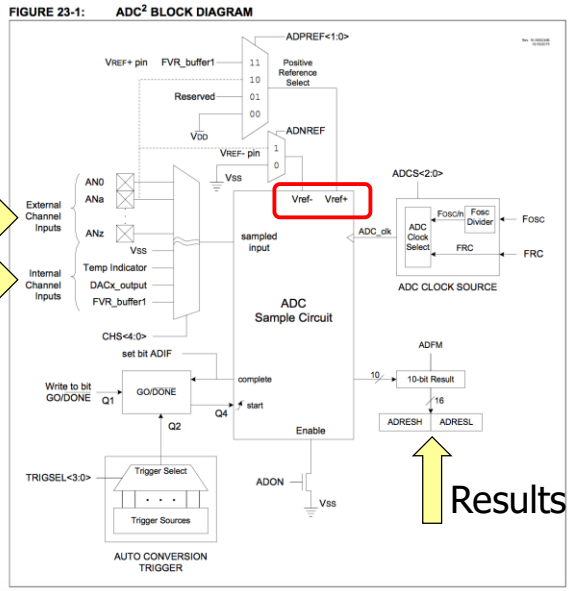
31

ADC – PIC16F18875

Inputs

Eight PORTA pins (RA<7:0>)
Eight PORTB pins (RB<7:0>)
Eight PORTC pins(RC<7:0>)
Eight PORTD pins(RD<7:0>)
Three PORTE pins(RE<2:0>)

Temperature Indicator
DAC output
Fixed Voltage Reference (FVR)
AVSS (ground)



32

32

ADC Configuration

When configuring and using the ADC the following functions must be considered:

1. Port configuration
2. Channel selection
3. ADC voltage reference selection
4. ADC conversion clock source
5. Interrupt control
6. Result formatting
7. Conversion Trigger Selection
8. ADC Acquisition Time
9. ADC Precharge Time
10. Additional Sample and Hold Capacitor

33

33

ADC – Input Selection

REGISTER 23-8: ADPCH: ADC POSITIVE CHANNEL SELECTION REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	ADPCH<5:0>					
bit 7							bit 0

bit 5-0 **ADPCH<5:0>**: ADC Positive Input Channel Selection bits

111111 = Fixed Voltage Reference (FVR)[2]
111110 = DAC1 output[1]
111101 = Temperature Indicator[3]
111100 = AVss (Analog Ground)
111011 = Reserved. No channel connected.
•
•
•
100010 = ANE2[4]
100001 = ANE1[4]
100000 = ANE0[4]
011111 = AND7[4]
011110 = AND6[4]
011101 = AND5[4]
011100 = AND4[4]
011011 = AND3[4]
011010 = AND2[4]
011001 = AND1[4]
011000 = AND0[4]
010111 = ANC7
010110 = ANC6
010101 = ANC5
010100 = ANC4
010011 = ANC3
010010 = ANC2
010001 = ANC1
010000 = ANC0
001111 = ANB7
001110 = ANB6
001101 = ANB5
001100 = ANB4
001011 = ANB3
001010 = ANB2
001001 = ANB1
001000 = ANB0
000111 = ANA7
000110 = ANA6
000101 = ANA5
000100 = ANA4
000011 = ANA3
000010 = ANA2
000001 = ANA1
000000 = ANA0

34

34

ADC – Reference Selection

REGISTER 23-7: ADREF: ADC REFERENCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	ADNREF	—	—	ADPREF<1:0>	
bit 7							bit 0

bit 7-5 **Unimplemented**: Read as '0'

bit 4 **ADNREF**: ADC Negative Voltage Reference Selection bit
1 = VREF- is connected to VREF- pin
0 = VREF- is connected to AVSS

bit 3-2 **Unimplemented**: Read as '0'

bit 1-0 **ADPREF**: ADC Positive Voltage Reference Selection bits
11 = VREF+ is connected to FVR_buffer 1
10 = VREF+ is connected to VREF+ pin
01 = Reserved
00 = VREF+ is connected to VDD

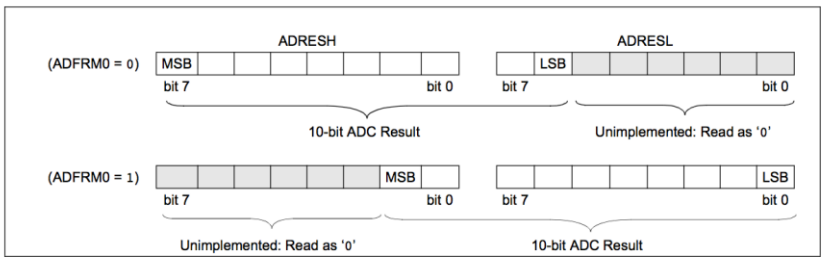
RA2/ANA2/C1IN0+/C2IN0+/VREF-/DAC1OUT1/IOCA2	RA2	TTL/ST	CMOS/OD	General purpose I/O.
	ANA2	AN	—	ADC Channel A2 input.
	C1IN0+	AN	—	Comparator positive input.
	C2IN0+	AN	—	Comparator positive input.
	VREF-	AN	—	External ADC and/or DAC negative reference input.
	DAC1OUT1	—	AN	Digital-to-Analog Converter output.
	IOCA2	TTL/ST	—	Interrupt-on-change input.
RA3/ANA3/C1IN1+/VREF+/MDCARL(1)/IOCA3	RA3	TTL/ST	CMOS/OD	General purpose I/O.
	ANA3	AN	—	ADC Channel A3 input.
	C1IN1+	AN	—	Comparator positive input.
	VREF+	AN	—	External ADC and/or DAC positive reference input.
	MDCARL(1)	TTL/ST	—	Modular Carrier input 1.
	IOCA3	TTL/ST	—	Interrupt-on-change input.

35

35

ADC – Result

FIGURE 23-3: 10-BIT ADC CONVERSION RESULT FORMAT



REGISTER 23-1: ADCON0: ADC CONTROL REGISTER 0

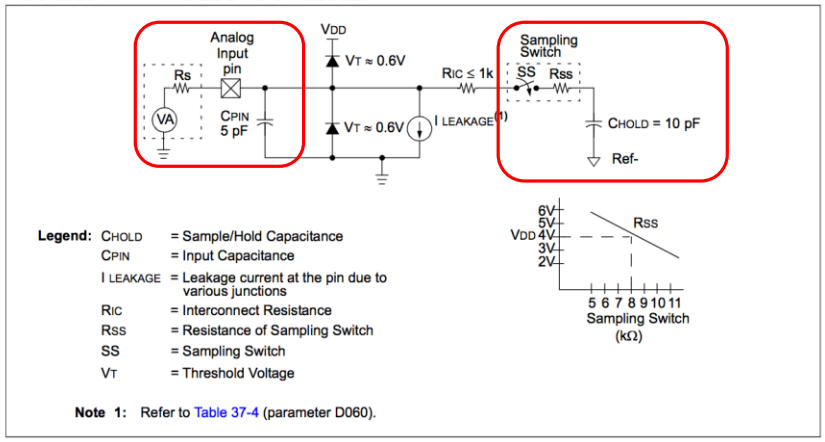
R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	U-0	R/W/HC-0
ADON	ADCONT	—	ADCS	—	ADFRM0	—	ADGO
bit 7							bit 0

36

36

ADC – Inside

FIGURE 23-4: ANALOG INPUT MODEL



Legend:

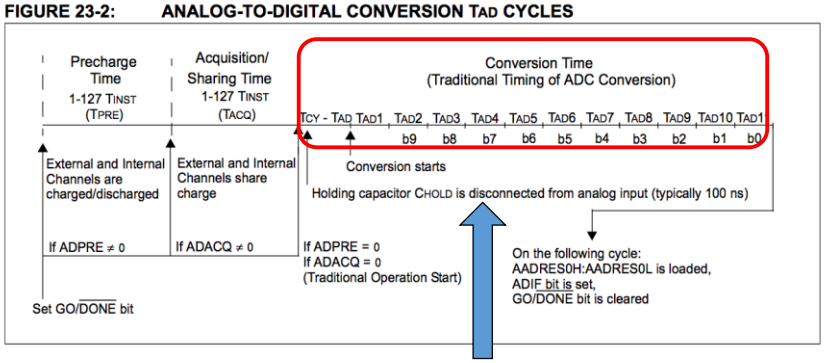
CHOLD	= Sample/Hold Capacitance
CPIN	= Input Capacitance
I LEAKAGE	= Leakage current at the pin due to various junctions
RIC	= Interconnect Resistance
RSS	= Resistance of Sampling Switch
SS	= Sampling Switch
VT	= Threshold Voltage

Note 1: Refer to Table 37-4 (parameter D060).

37

Conversion

FIGURE 23-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES



Successive Approximation

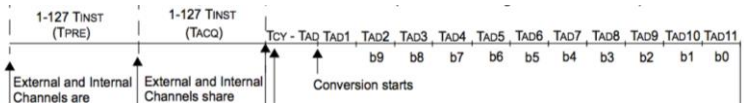
Speed at which this is done is determined by the conversion clock

38

ADC Clock Period

TABLE 23-1: ADC CLOCK PERIOD (TAD) VS. DEVICE OPERATING FREQUENCIES

ADC Clock Period (TAD)		Device Frequency (FOSC)					
ADC Clock Source	ADCCS<5:0>	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
FOSC/2	000001	62.5ns(2)	100 ns(2)	125 ns(2)	250 ns(2)	500 ns(2)	2.0 µs
FOSC/4	000010	125 ns(2)	200 ns(2)	250 ns(2)	500 ns(2)	1.0 µs	4.0 µs
FOSC/6	000011	187.5 ns(2)	300 ns(2)	375 ns(2)	750 ns(2)	1.5 µs	6.0 µs
FOSC/8	000100	250 µs(2)	400 ns(2)	500 µs(2)	1.0 µs	2.0 µs	8.0 µs(3)
...	...	...	...	...	...	...	...
FOSC/16	000101	500 ns(2)	800 ns(2)	1.0 µs	2.0 µs	4.0 µs	16.0 µs(2)
...	...	...	...	...	...	...	...
FOSC/128	111111	4.0 µs	6.4 µs	8.0 µs	16.0 µs(3)	32.0 µs(2)	128.0 µs(2)
FRC	ADCS(ADCON0 <4>)=1	1.0-6.0 µs(1)	1.0-6.0 µs(1)	1.0-6.0 µs(1)	1.0-6.0 µs(1)	1.0-6.0 µs(1)	1.0-6.0 µs(1)



REGISTER 23-6: ADCLK: ADC CLOCK SELECTION REGISTER

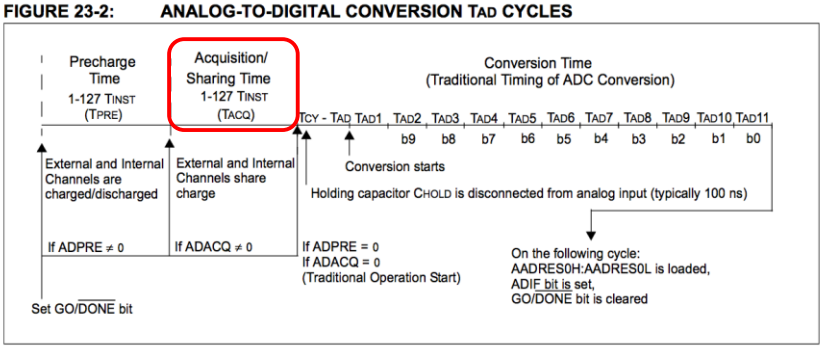
U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	ADCCS<5:0>					
bit 7							bit 0

39

39

Acquisition Time

FIGURE 23-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES



Tacq = Amplifier Settling Time (Tamp)
+ Holding Capacitor Charging Time (Tc)
+ Temperature Coefficient Time (Tcoff)

Tacq = Tamp + Tc + Tcoff

Tamp = 2 µs for PIC 16F18875

Tcoff = (temperature – 25^0)(0.05µs/^{0}C)

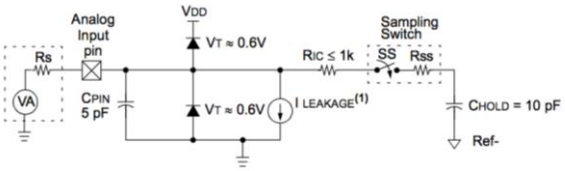
40

40

Acquisition Time - Tc

Capacitor Charges through Series Resistances:

Rs = External signals source resistance (Rs ~ 10kΩ use 10,000 in lab)
Ric = Interconnect Resistance (Ric ~1kΩ)
Rss = Sample Switch Resistance (Rss~ 7kΩ given Vdd=5V)



C_{HOLD} = Capacitor = $10 * 10^{-12}$ farads

So Time for capacitor to charge: Tc = - C_{HOLD}*(Rs + Ric + Rss) * ln (1/2047) = $1.37\mu s$

Thus: Tacq = Tamp + Tc + Tcoff = $2\mu s + 1.37\mu s + (50^0 - 25^0)\left(\frac{0.05\mu s}{^oC}\right) = 4.62\mu s$

41

41

ADC Conversion - Configuration

1. Configure port (TRISx and ANSELx)
2. Configure ADC module
   1. Select ADC conversion clock (ADCLK)
   2. Configure voltage reference (ADREF)
   3. Select ADC input channel (ADPCH)
   4. Turn on ADC module (ADCON0)
3. Configure ADC Interrupt (optional)
   1. Clear ADC flag (PIR1<ADIF>),
   2. Enable ADC interrupt (PIE1<ADIE>),
   3. Enable peripheral interrupt (INTCON<PEIE>),
   4. Enable global interrupt (INTCON<GIE>)

Remember for I/O notes: to set up a pin as either a digital I/O pin or an analog pin

See next slides

42

42

ANSELx Register

REGISTER 12-5: ANSELA: PORTA ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSA7	ANSA6	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **ANSA<7:0>**: Analog Select between Analog or Digital Function on pins RA<7:0>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

43

43

Configuration Register

REGISTER 23-1: ADCON0: ADC CONTROL REGISTER 0

R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	U-0	R/W/HC-0
ADON	ADCONT	—	ADCS	—	ADFRM0	—	ADGO
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled

bit 6 **ADCONT:** ADC Continuous Operation Enable bit
1 = ADGO is retriggered upon completion of each conversion trigger until ADTIF is set (if ADSOI is set) or until ADGO is cleared (regardless of the value of ADSOI)
0 = ADGO is cleared upon completion of each conversion trigger

bit 5 **Unimplemented:** Read as '0'

bit 4 **ADCS:** ADC Clock Selection bit
1 = Clock supplied from FRC dedicated oscillator
0 = Clock supplied by Fosc, divided according to ADCLK register

bit 3 **Unimplemented:** Read as '0'

bit 2 **ADFRM0:** ADC results Format/alignment Selection
1 = ADRES and ADPREV data are right-justified
0 = ADRES and ADPREV data are left-justified, zero-filled

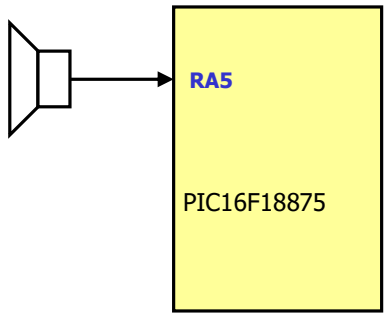
bit 1 **Unimplemented:** Read as '0'

bit 0 **ADGO:** ADC Conversion Status bit
1 = ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle. The bit is cleared by hardware as determined by the ADCONT bit
0 = ADC conversion completed/not in progress

44

44

Example 1 – ADC Sound Conversion



Write a program that makes continuous measurements of the output voltage of the microphone and writes the results to TEMPH (0x20) and TEMPL (0x21)

Assume that the Oscillator frequency clock is 4MHz

45

Example 1 – TRISA and ANSELA

REGISTER 12-3: TRISA: PORTA TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
bit 7							bit 0

1

REGISTER 12-5: ANSELA: PORTA ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSA7	ANSA6	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0
bit 7							bit 0

1

46

Example 1 – ADCLK

TABLE 23-1: ADC CLOCK PERIOD (T_{AD}) Vs. DEVICE OPERATING FREQUENCIES

ADC Clock Period (T_{AD})		Device Frequency (F_{OSC})					
ADC Clock Source	ADCCS<5:0>	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
F_{OSC}/2	000001	62.5ns(2)	100 ns(2)	125 ns(2)	250 ns(2)	500 ns(2)	2.0 µs
F_{OSC}/4	000010	125 ns(2)	200 ns(2)	250 ns(2)	500 ns(2)	1.0 µs	4.0 µs
F_{OSC}/6	000011	187.5 ns(2)	300 ns(2)	375 ns(2)	750 ns(2)	1.5 µs	6.0 µs
F_{OSC}/8	000100	250 µs(2)	400 ns(2)	500 µs(2)	1.0 µs	2.0 µs	8.0 µs(3)
...	...	...	...	...	...	...	...
F_{OSC}/16	000101	500 ns(2)	800 ns(2)	1.0 µs	2.0 µs	4.0 µs	16.0 µs(2)
...	...	...	...	...	...	...	...
F_{OSC}/128	111111	4.0 µs	6.4 µs	8.0 µs	16.0 µs(3)	32.0 µs(2)	128.0 µs(2)
FRC	ADCS(ADCON0 <4>)=1	1.0-6.0 µs(1)	1.0-6.0 µs(1)	1.0-6.0 µs(1)	1.0-6.0 µs(1)	1.0-6.0 µs(1)	1.0-6.0 µs(1)

REGISTER 23-6: ADCLK: ADC CLOCK SELECTION REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	ADCCS<5:0>					
bit 7							bit 0

0 0 0 0 0 0 1 0

Will take 1 microsecond to convert 1 bit

Example 1 - ADREF

bit 7-5 **Unimplemented**: Read as '0'

bit 4 **ADNREF**: ADC Negative Voltage Reference Selection bit
1 = V_{REF-} is connected to V_{REF-} pin
0 = V_{REF-} is connected to AV_{SS} ⟵

bit 3-2 **Unimplemented**: Read as '0'

bit 1-0 **ADPREF**: ADC Positive Voltage Reference Selection bits
11 = V_{REF+} is connected to FVR_buffer 1
10 = V_{REF+} is connected to V_{REF+} pin
01 = Reserved
00 = V_{REF+} is connected to V_{DD} ⟵

REGISTER 23-7: ADREF: ADC REFERENCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	ADNREF	—	—	ADPREF<1:0>	
bit 7							bit 0

0 0 0 0 0 0 0 0

Example 1 - ADPCH

REGISTER 23-8: ADPCH: ADC POSITIVE CHANNEL SELECTION REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	ADPCH<5:0>					
bit 7							bit 0

0 0 0 0 0 1 0 1

bit 5-0 **ADPCH<5:0>**: ADC Positive Input Channel Selection bits

111111 = Fixed Voltage Reference (FVR)(2)
111110 = DAC1 output(1)
111101 = Temperature Indicator(3)
111100 = AVSS (Analog Ground)
111011 = Reserved. No channel connected.
•
•
•
100010 = ANE2(4)
100001 = ANE1(4)
100000 = ANE0(4)
011111 = AND7(4)
011110 = AND6(4)
011101 = AND5(4)
011100 = AND4(4)
011011 = AND3(4)
011010 = AND2(4)
011001 = AND1(4)
011000 = AND0(4)
010111 = ANC7
010110 = ANC6
010101 = ANC5
010100 = ANC4
010011 = ANC3
010010 = ANC2
010001 = ANC1
010000 = ANC0
001111 = ANB7
001110 = ANB6
001101 = ANB5
001100 = ANB4
001011 = ANB3
001010 = ANB2
001001 = ANB1
001000 = ANB0
000111 = ANA7
000110 = ANA6
000101 = ANA5 ⟸
000100 = ANA4
000011 = ANA3
000010 = ANA2
000001 = ANA1
000000 = ANA0

49

49

Example 1 – ADCON0

REGISTER 23-1: ADCON0: ADC CONTROL REGISTER 0

R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	U-0	R/W/HC-0
ADON	ADCONT	—	ADCS	—	ADFRM0	—	ADGO
bit 7							bit 0

1 0 0 0 0 1 0 0

bit 7 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled

bit 6 **ADCONT:** ADC Continuous Operation Enable bit
1 = ADGO is retriggered upon completion of each conversion trigger until ADTIF is set (if ADSOI is set) or until ADGO is cleared (regardless of the value of ADSOI)
0 = ADGO is cleared upon completion of each conversion trigger

bit 5 **Unimplemented:** Read as '0'

bit 4 **ADCS:** ADC Clock Selection bit
1 = Clock supplied from FRC dedicated oscillator
0 = Clock supplied by FOSC, divided according to ADCLK register

bit 3 **Unimplemented:** Read as '0'

bit 2 **ADFRM0:** ADC results Format/alignment Selection
1 = ADRES and ADPREV data are right-justified
0 = ADRES and ADPREV data are left-justified, zero-filled

bit 1 **Unimplemented:** Read as '0'

bit 0 **ADGO:** ADC Conversion Status bit
1 = ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle. The bit is cleared by hardware as determined by the ADCONT bit
0 = ADC conversion completed/not in progress

50

50

Example 1 – Configuration Code

Assembly:

```
; PORT A configuration
banksel     PORTA
clrf        PORTA
clrf        LATA
movlw       b'11111111'
movwf       TRISA
banksel     ANSELA
movlw       b'11111111'
movwf       ANSELA

; Set conversion clock speed
banksel     ADCLK
movlw       b'00000010'
movwf       ADCLK

; Set voltage references (Vref+ and Vref-)
banksel     ADREF
clrf        ADREF

; Set input channel
banksel     ADPCH
movlw       b'00000101'
movwf       ADPCH

; Set global configuration
banksel     ADCON0
movlw b'10000100'
movwfADCON0
```

C:

```
// PORTA configuration
PORTA = 0;
LATA = 0;
TRISA = 0xFF;
ANSELA = 0xFF;

// Set Conversion clock speed
ADCLK = 0x02;

// Set voltage references
ADREF = 0;

// Set input channel
ADPCH = 0x05;

// Set global configuration
ADCON0 = 0x84;
```

51

51

ADACQ Register

REGISTER 23-10: ADACQ: ADC ACQUISITION TIME CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ADACQ<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **ADACQ<7:0>**: Acquisition (charge share time) Select bits(1)

11111111= Acquisition time is 255 clocks of the selected ADC clock

11111110= Acquisition time is 254 clocks of the selected ADC clock

•

•

•

00000001= Acquisition time is 1 clock of the selected ADC clock

00000000= Acquisition time is not included in the data conversion cycle(2)

52

52

ADC Conversion – Basic Mode

- If ADACQ = 0 software must wait the required acquisition time
  - Use a Timing Loop or Delay for this
- Start conversion by setting the ADGO bit
- Wait for the conversion to be complete by:
  - Polling the ADGO bit
  - Waiting for the ADC interrupt (ADIF)
- Read ADC results
- Clear ADC interrupt flag (if interrupts are used)

53

53

Example 1 – Performing Conversions

Assembly:

```
MAIN: call    DELAY          ; Wait for TACQ – for example 5 microseconds (see slide 41)
      banksel       ADCON0
      bsf     ADCON0, ADGO      ; Start conversion
CHECK:      btfsc   ADCON0, ADGO      ; Wait for conversion to complete*
      goto    CHECK             ; use POLLING
      banksel       ADRESH
      movf    ADRESH, W         ; Read A/D result (high byte)
      movwfTEMPH                ; Write to TEMPH
      banksel       ADRESL
      movf    ADRESL,W          ; Read A/D result (low byte)
      movwfTEMPL        ; Write to TEMPL
      goto    MAIN
```

C:

```
int8_t        TEMPH, TEMPL;

while(1) {
              __delay_us(5);
              ADGO = 1;
              while(ADGO == 1) continue;
              TEMPH = ADRESH;
              TEMPL = ADRESL;
}
```

54

54

Example 1 – Timing Analysis

Assembly:

```
MAIN: call    DELAY        ; 5
      banksel      ADCON0          ; 1
      bsf     ADCON0, ADGO     ; 1
CHECK:      btfsc   ADCON0, ADGO     ; ~12Tad = 12
      goto    CHECK            ;
      banksel      ADRESH          ; 1
      movf    ADRESH, W        ; 1
      movwfTEMPH            ; 1
      banksel      ADRESL          ; 1
      movf    ADRESL,W         ; 1
      movwfTEMPL       ; 1
      goto    MAIN        ; 2
```

Sample Period = $T_s \approx 5 + 1 + 1 + 12 + 1 + 1 + 1 + 1 + 1 + 1 + 2 = 27T_{CY} = 27\mu sec$

Sample Frequency = $1/T_s \approx 1/27\mu sec = 37{,}037Hz \approx 37kHz$

55

55

Nyquist Criterion

The **Nyquist criterion** states that, in order to prevent undesired aliasing, one must sample a signal at a rate equal to **at least twice its bandwidth**.

$$f_s > 2B$$

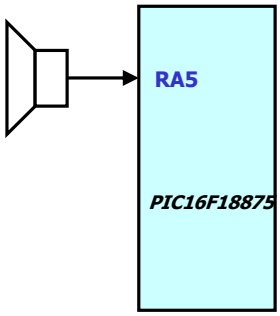
Sample frequency

Signal bandwidth

56

56

Example 1 - Nyquist



Range of human hearing:
20 to 20,000 Hz

Must sample at
$f_s > 2*(20{,}000 - 20) = 39{,}960\text{Hz} \approx 40\text{kHz}$

For example: CD – 44.1kHz sampling rate
DVD – up to 96 kHz/192kHz
Blu-ray – 192kHz
And
CD – 16 bits per sample

57

57

Other Functionality in ADC

- Continuous conversions
- Built-in acquisition time and pre-charge time
- Computational features:
  - Averaging and low-pass filter functions
  - Reference comparison
  - 2-level threshold comparison

58

58