Microprocessors and Microcontrollers

Analog-to-Digital Conversion

EE3954

by

Maarten Uijt de Haag, Tim Bambeck
Figure 4-2 in the Reference Manual
Analog-to-Digital Conversion (ADC)

Example

Sound → Voltage (Transducer) → Electric Circuit → Voltage (Signal Conditioner) → RA1/AN1 (ADC) → PIC16F877 (Microcontroller)

Analog Signal

$t$
Analog-to-Digital Conversion

Analog Signal
\[ v(t) \]

Discrete-Amplitude Signal
\[ v'(t) \]

Discrete-Time Signal
\[ x[k] \]

Digital Signal
\[ x'[k] \]
Digital-to-Analog Conversion (DAC)

Example

Microcontroller

PIC16F877

Transducer

Sound

Voltage

DAC

Binary Number

 DAC

Analog Signal

\( x(t) \)

Digital Signal

\( x'[k] \)

PIC Microcontroller does NOT have a built-in DAC!!!
ADC
Comparator

Basic Component:

Input signal, Vin

Reference Voltage, Vref

Output, Vout

If Vin > Vref, Vout = 5VDC (Logic ‘1’)
If Vin ≤ Vref, Vout = 0VDC (Logic ‘0’)
ADC
1-bit Example

Vin > 2.5V: Signal is ‘1’

Vin ≤ 2.5V: Signal is ‘0’

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

2-bit Example

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

5 v / 4 levels = 1.25 each increment (thresholds)

- $3.75 < V_{in} < 5.00 \text{V}$: Signal is ’11’
- $2.50 < V_{in} \leq 3.75 \text{V}$: Signal is ’10’
- $1.25 < V_{in} \leq 2.50 \text{V}$: Signal is ’01’
- $0.00 \text{V} < V_{in} \leq 1.25 \text{V}$: Signal is ’00’
ADC
3-bit Example

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

<table>
<thead>
<tr>
<th>Input Voltage</th>
<th>A/D Converter Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.00 V</td>
<td>‘111’</td>
</tr>
<tr>
<td>4.375 V</td>
<td>‘110’</td>
</tr>
<tr>
<td>3.750 V</td>
<td>‘101’</td>
</tr>
<tr>
<td>3.125 V</td>
<td>‘100’</td>
</tr>
<tr>
<td>2.500 V</td>
<td>‘011’</td>
</tr>
<tr>
<td>1.875 V</td>
<td>‘010’</td>
</tr>
<tr>
<td>1.250 V</td>
<td>‘001’</td>
</tr>
<tr>
<td>0.625 V</td>
<td>‘000’</td>
</tr>
<tr>
<td>0.000 V</td>
<td></td>
</tr>
</tbody>
</table>
### ADC

#### 3-bit Example

<table>
<thead>
<tr>
<th>Analog Input Voltage</th>
<th>Digital Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.8756V</td>
<td>100</td>
</tr>
<tr>
<td>-0.234V</td>
<td>000</td>
</tr>
<tr>
<td>4.9876V</td>
<td>111</td>
</tr>
<tr>
<td>1.1V</td>
<td>001</td>
</tr>
<tr>
<td>3.2V</td>
<td>101</td>
</tr>
</tbody>
</table>
ADC
The PIC16F877 Microcontroller

- **10-bits:**
  - $2^{10} = 1024$ levels $\Rightarrow$ 1023 thresholds
  - Resolution $= \frac{5V}{1024} = 0.0048828125$ V
  - Thresholds:
    - 0.0048828125 V
    - 0.009765625 V
    - 0.0146484375 V
    - Etc.
ADC

Implementation: Parallel-Encoded or Flash ADC

Example: 3-bit ADC

Vref = 5V

Vin

O0
O1
O2

Digital Value out

I7
I6
I5
I4
I3
I2
I1
I0

Vref = 5V

4.375V
3.750V
3.125V
2.500V
1.875V
1.250V
0.625V
**Digital Logic - Revisited (?)**

*Priority Encoder*

<table>
<thead>
<tr>
<th>$\bar{I}_7$</th>
<th>$\bar{I}_6$</th>
<th>$\bar{I}_5$</th>
<th>$\bar{I}_4$</th>
<th>$\bar{I}_3$</th>
<th>$\bar{I}_2$</th>
<th>$\bar{I}_1$</th>
<th>$\bar{I}_0$</th>
<th>$\bar{O}_2$</th>
<th>$\bar{O}_1$</th>
<th>$\bar{O}_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>X</td>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>X</td>
<td>X</td>
<td>H</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>L</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>L</td>
<td>L</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: if one uses active low, LLH = 110
ADC

Implementation: Parallel-Encoded or Flash ADC

Vref = 5V

Vin

I7
I6
I5
I4
I3
I2
I1
I0

O0
O1
O2

Digital Value Out

Major drawback:
An n-bit ADC Requires 2ⁿ⁻¹ comparators, thus expensive

Advantage:
Highest speed ADC available

Example:
3-bit ADC

4.375V
3.750V
3.125V
2.500V
1.875V
1.250V
0.625V

ADC.14
ADC in 16F877
- Successive Approximation ADC

Note: for an $n$-bit ADC it will take $n$ clock-cycles to find an output.
Successive Approximation – HW Flowchart

1. Set all bits D7...D0 to '0'
2. Set D7 to '1' and compare V to Vin
   - If V < Vin, go to Clear D7
   - If V ≥ Vin, set D7 to '0'
3. Set D6 to '1' and compare V to Vin
   - If V < Vin, go to Clear D6
   - If V ≥ Vin, set D6 to '0'
4. Set D5 to '1' and compare V to Vin
   - If V < Vin, go to Clear D5
   - If V ≥ Vin, set D5 to '0'
5. Set D0 to '1' and compare V to Vin
   - If V < Vin, go to Clear D0
   - If V ≥ Vin, set D0 to '0'
Successive Approximation

Example - 4.3V - Step 1 - Bit 7 (D7)

D7 = Most Significant Bit

Threshold = 2.5V

D7 = 0  (0XXX XXXX)

D7 = 1  (1XXX XXXX)

Result: 1XXX XXXX
Successive Approximation

Example - 4.3V - Step 2 - Bit 6 (D6)

D7 = Most Significant Bit

\[ \text{Result: } 11XX \text{ } XXXX \]
Example

Voltage input range: 0V – 5V

Number of bits: $8 \Rightarrow 2^8 = 256$ **levels** ( = 255 thresholds)

Resolution = $\frac{5V}{256} = 0.01953125 \text{ V}$

Thresholds at:
- $0.01953125 \text{ V}$
- $2 \times 0.01953125 \text{ V} = 0.0390625 \text{ V}$
- $3 \times 0.01953125 \text{ V} = 0.05859375 \text{ V}$
- ...

$4.3 \text{ V} / \text{resolution} = 220.16 \Rightarrow 220 = \text{binary value} = 1101\ 1100$
ADC
As part of the microcontroller ...

• **Reference Manual:**
  - Section 21 & 22 (8-bit ADC),
  - **Section 23 (10-bit ADC)**

• **Datasheet:**
  - **Section 11 (10-bit ADC)**
ADC
As part of the microcontroller ...

*or A/D
# ADC

The configuration ...

## TABLE 11-2: REGISTERS/BITS ASSOCIATED WITH A/D

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 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>
<th>Value on POR, BOR</th>
<th>Value on MCLR, WDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0Bh, 8Bh, 10Bh, 18Bh</td>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>R1F</td>
<td>0000 000x</td>
<td>0000 000u</td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td>PSPIF</td>
<td>ADIF</td>
<td>RCIF</td>
<td>TXIF</td>
<td>SSPIF</td>
<td>CCP1IF</td>
<td>TMR2IF</td>
<td>TMR1F</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>8Ch</td>
<td>PIE1</td>
<td>PSPIE</td>
<td>ADIE</td>
<td>RCIE</td>
<td>TXIE</td>
<td>SSPIE</td>
<td>CCP1IE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>1Eh</td>
<td>ADRESH</td>
<td>V/D Result Register High Byte</td>
<td>xxxx xxxx</td>
<td>uuuu uuuu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9Eh</td>
<td>ADRESI</td>
<td>V/D Result Register Low Byte</td>
<td>xxxx xxxx</td>
<td>uuuu uuuu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1Fh</td>
<td>ADCON0</td>
<td>ADCS1</td>
<td>ADCS0</td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td>—</td>
<td>ADON</td>
<td>0000 00-0</td>
<td>0000 00-0</td>
</tr>
<tr>
<td>9Fh</td>
<td>ADCON1</td>
<td>ADFM</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PCFG3</td>
<td>PCFG2</td>
<td>PCFG1</td>
<td>PCFG0</td>
<td>--0- 0000</td>
<td>--0- 0000</td>
</tr>
<tr>
<td>85h</td>
<td>TRISA</td>
<td>—</td>
<td>—</td>
<td>PORTA Data Direction Register</td>
<td>--11 1111</td>
<td>--11 1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05h</td>
<td>PORTA</td>
<td>—</td>
<td>—</td>
<td>PORTA Data Latch when written: PORTA pins when read</td>
<td>--0x 0000</td>
<td>--0u 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89h(1)</td>
<td>TRISE</td>
<td>IBF</td>
<td>OBF</td>
<td>IBOV</td>
<td>PSPMODE</td>
<td>—</td>
<td>PORTE Data Direction bits</td>
<td>0000 -111</td>
<td>0000 -111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09h(1)</td>
<td>PORTE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>RE2</td>
<td>RE1</td>
<td>RE0</td>
<td>----- -xxx</td>
<td>----- -uuu</td>
</tr>
</tbody>
</table>

Legend:  
- x = unknown, a = unchanged, - = unimplemented, read as ‘0’. Shaded cells are not used for A/D conversion.  

**Note 1:** These registers/bits are not available on the 28-pin devices.
# ADC Register (SFR) settings

## REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

<table>
<thead>
<tr>
<th>Bit</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>ADCS1</td>
<td>ADCS0</td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td>—</td>
<td>ADON</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-3</td>
<td>CHS2</td>
<td>CHS0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>GO/DONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Unimplemented</td>
<td>Read as '0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>ADON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **bit 7-6** `ADCS1:ADCS0`: A/D Conversion Clock Select bits
  - 00 = Fosc/2
  - 01 = Fosc/8
  - 10 = Fosc/32
  - 11 = FRC (clock derived from the internal A/D module RC oscillator)

- **bit 5-3** `CHS2:CHS0`: Analog Channel Select bits
  - 000 = channel 0, (RA0/AN0)
  - 001 = channel 1, (RA1/AN1)
  - 010 = channel 2, (RA2/AN2)
  - 011 = channel 3, (RA3/AN3)
  - 100 = channel 4, (RA5/AN4)
  - 101 = channel 5, (RE0/AN5)(1)
  - 110 = channel 6, (RE1/AN6)(1)
  - 111 = channel 7, (RE2/AN7)(1)

- **bit 2** `GO/DONE`: A/D Conversion Status bit
  - If ADON = 1:
    - 1 = A/D conversion in progress (setting this bit starts the A/D conversion)
    - 0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

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

- **bit 0** `ADON`: A/D On bit
  - 1 = A/D converter module is operating
  - 0 = A/D converter module is shut-off and consumes no operating current
## ADC Register (SFR) settings

**REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W-0</td>
<td>ADCS1</td>
<td>A/D Conversion Clock Select bits</td>
</tr>
<tr>
<td>R/W-0</td>
<td>ADCS0</td>
<td></td>
</tr>
<tr>
<td>R/W-0</td>
<td>CHS2</td>
<td></td>
</tr>
<tr>
<td>R/W-0</td>
<td>CHS1</td>
<td></td>
</tr>
<tr>
<td>R/W-0</td>
<td>CHS0</td>
<td></td>
</tr>
<tr>
<td>R/W-0</td>
<td>GO/DONE</td>
<td>A/D Conversion Status bit</td>
</tr>
<tr>
<td>U-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R/W-0</td>
<td>ADON</td>
<td>A/D On bit</td>
</tr>
</tbody>
</table>

### Bit 7
- **ADCS1:ADCS0**: A/D Conversion Clock Select bits
  - 00 = Fosc/2
  - 01 = Fosc/8
  - 10 = Fosc/32
  - 11 = FRC (clock derived from the internal A/D module RC oscillator)

### Bit 5-3
- **CHS2:CHS0**: Analog Channel Select bits
  - 000 = channel 0, (RA0/AN0)
  - 001 = channel 1, (RA1/AN1)
  - 010 = channel 2, (RA2/AN2)
  - 011 = channel 3, (RA3/AN3)
  - 100 = channel 4, (RA4/AN4)
  - 101 = channel 5, (RE0/AN5)\(^{(1)}\)
  - 110 = channel 6, (RE1/AN6)\(^{(1)}\)
  - 111 = channel 7, (RE2/AN7)\(^{(1)}\)

### Bit 2
- **GO/DONE**: A/D Conversion Status bit
  - If ADON = 1:
    - 1 = A/D conversion in progress (setting this bit starts the A/D conversion)
    - 0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

### Bit 1
- **Unimplemented**: Read as '0'

### Bit 0
- **ADON**: A/D On bit
  - 1 = A/D converter module is operating
  - 0 = A/D converter module is shut-off and consumes no operating current
Digital input: the standard input as we used it before!!

ADC.26
## ADC

The result ...

### TABLE 11-2: REGISTERS/BITS ASSOCIATED WITH A/D

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit  7</th>
<th>Bit  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>
<th>Value on POR, BOR</th>
<th>Value on MCLR, WDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0Bh, 8Bh, 10Bh, 18Bh</td>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RBIF</td>
<td>0000 000x</td>
<td>0000 000u</td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td>PSPIF(1)</td>
<td>ADIF</td>
<td>RCIF</td>
<td>TXIF</td>
<td>SSPIF</td>
<td>CCP1IF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>8Ch</td>
<td>PIE1</td>
<td>PSPIE(1)</td>
<td>ADIE</td>
<td>RCIE</td>
<td>TXIE</td>
<td>SSPIE</td>
<td>CCP1IE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>1Eh</td>
<td>ADRESH</td>
<td>A/D Result Register High Byte</td>
<td>xxxx xxxx</td>
<td>uuuu uuuu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9Eh</td>
<td>ADRESL</td>
<td>A/D Result Register Low Byte</td>
<td>xxxx xxxx</td>
<td>uuuu uuuu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1Fh</td>
<td>ADCON0</td>
<td>ADCS1</td>
<td>ADCS0</td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td>—</td>
<td>ADON</td>
<td>0000 00-0</td>
<td>0000 00-0</td>
</tr>
<tr>
<td>9Fh</td>
<td>ADCON1</td>
<td>ADFM</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PCFG3</td>
<td>PCFG2</td>
<td>PCFG1</td>
<td>PCFG0</td>
<td>-0- 0000</td>
<td>-0- 0000</td>
</tr>
<tr>
<td>85h</td>
<td>TRISA</td>
<td>—</td>
<td>—</td>
<td>PORTA Data Direction Register</td>
<td>-11 1111</td>
<td>-11 1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05h</td>
<td>PORTA</td>
<td>—</td>
<td>—</td>
<td>PORTA Data Latch when written: PORTA pins when read</td>
<td>-0x 0000</td>
<td>-0u 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89h(1)</td>
<td>TRISE</td>
<td>IBF</td>
<td>OBV</td>
<td>IBOV</td>
<td>PSPMODE</td>
<td>—</td>
<td>PORTE Data Direction bits</td>
<td>0000 -111</td>
<td>0000 -111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09h(1)</td>
<td>PORTE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>RE2</td>
<td>RE1</td>
<td>RE0</td>
<td>—----- -xxx ------ uuu</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  

- **x** = unknown, **u** = unchanged, **-** = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

**Note 1:** These registers/bits are not available on the 28-pin devices.
ADC
The result ...

Figure 23-6: A/D Result Justification

10-Bit Result

ADF = 1

0000 00
RESULT

ADRESH
ADRESL
10-bits
Right Justified

ADF = 0

0 7 6 5 0
RESULT

ADRESH
ADRESL
10-bits
Left Justified

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

<table>
<thead>
<tr>
<th></th>
<th>U-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>ADFM</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>PCFG3</td>
<td>PCFG2</td>
<td>PCFG1</td>
<td>PCFG0</td>
</tr>
<tr>
<td>bit 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FIGURE 11-2: ANALOG INPUT MODEL

Legend
CPIN = input capacitance
VT = threshold voltage
I LEAKAGE = leakage current at the pin due to various junctions
RIC = interconnect resistance
SS = sampling switch
CHOLD = sample/hold capacitance (from DAC)
ADC
Timing ...

Figure 23-2: A/D Conversion Sequence

A/D Sample Time

Acquisition Time  A/D Conversion Time

A/D conversion complete, result is loaded in ADRES register. Holding capacitor begins acquiring voltage level on selected channel ADIF bit is set

When A/D conversion is started (setting the GO bit)

When A/D holding capacitor starts to charge. After A/D conversion, or when new A/D channel is selected
Sample/Hold Acquisition Time

\[ Tacq = Tamp + Tc + Tcoff \]

- **Tamp** = 2 \( \mu \)s for PIC 16F877

Next few slides find **Tc & Tcoff**:
Calculations for $T_c$:

Capacitor Charges through Series Resistances:

R$_s$ = External signals source resistance (use 10,000 in lab)
R$_{ic}$ = Interconnect Resistance (R$_{ic}$ ≤ 1,000 ohm)
R$_{ss}$ = Sample Switch Resistance (function of V$_{dd}$)
(for V$_{dd}$ = 5v, R$_{ss}$ = 7,000 ohm)

CHOLD = Capacitor = $120 \times 10^{-12}$ farads

So Time for capacitor to charge:

$T_c = -\text{CHOLD} \times (R_s + R_{ic} + R_{ss}) \times \ln(1/2047)$
Calculations for $T_c$ (continued):

$T_c = - \text{CHOLD}( R_s + R_i + R_s) \times \ln \left( \frac{1}{2047} \right)$

$T_c = - (120 \times 10^{-12})(10,000 + 1,000 + 7,000) \times \ln \left( \frac{1}{2047} \right)$

$T_c = - (120 \times 10^{-12}) \times (18,000) \times (-7.624)$

Therefore: \hspace{1cm} $T_c = 16.5 \mu$s
Finding Time for Temperature Coefficient Term (Tcoff):

\[ T_{coff} = (\text{Operating Temp in } ^\circ\text{C} - 25^\circ\text{C}) \times (0.05 \mu\text{s/}^\circ\text{C}) \]

Let's assume worst case operating temperature is 50°C (= 122°F)

\[ T_{coff} = (50^\circ\text{C} - 25^\circ\text{C}) \times (0.05 \mu\text{s/}^\circ\text{C}) \]

Therefore: \( T_{coff} = 1.25 \mu\text{s} \)
So total Acquisition time $T_{acq} =$

$T_{acq} = T_{amp} + T_{c} + T_{coff}$

$T_{acq} = 2 \mu s + 16.5 \mu s + 1.25 \mu s$

$T_{acq} = 19.75 \mu s$

With our 4 MHz PIC’s in lab – (1 \mu s instruction cycle)
We would delay 20 \mu s before we start conversion (GO is set)
Figure 23-2: A/D Conversion Sequence

A/D Sample Time

- Acquisition Time
- A/D Conversion Time

19.75 µs (example)

When A/D conversion is started (setting the GO bit)

- A/D conversion complete, result is loaded in ADRES register.
- Holding capacitor begins acquiring voltage level on selected channel
- ADIF bit is set

When A/D holding capacitor starts to charge.

After A/D conversion, or when new A/D channel is selected.
ADC Conversion Time for Successive Approx. Converter

T_{AD} must be ≥ 1.6µs

TABLE 11-1: T_{AD} vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))

<table>
<thead>
<tr>
<th>AD Clock Source (TAD)</th>
<th>ADCS1:ADCS0</th>
<th>Maximum Device Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td></td>
<td>Max.</td>
</tr>
<tr>
<td>2Tosc</td>
<td>00</td>
<td>1.25 MHz</td>
</tr>
<tr>
<td>8Tosc</td>
<td>01</td>
<td>5 MHz</td>
</tr>
<tr>
<td>32Tosc</td>
<td>10</td>
<td>20 MHz</td>
</tr>
<tr>
<td>RC(1, 2, 3)</td>
<td>11</td>
<td>(Note 1)</td>
</tr>
</tbody>
</table>
Selecting $T_{AD}$ Clock Source Value Using PIC’s in Lab:

Minimum $T_{AD}$ must be $\geq 1.6 \, \mu s$

PIC’s in Lab use 4 MHz oscillator, so $T_{osc} = 0.250 \, \mu s$

We need to set ADCS1, ADCS0 bits in ADCON0 to select $T_{AD}$:

<table>
<thead>
<tr>
<th>ADCS1,ADCS0</th>
<th>Operation</th>
<th>$T_{AD}$ Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>$2 \times T_{osc}$</td>
<td>0.50 \mu s</td>
</tr>
<tr>
<td>0 1</td>
<td>$8 \times T_{osc}$</td>
<td>2.00 \mu s</td>
</tr>
<tr>
<td>1 0</td>
<td>$32 \times T_{osc}$</td>
<td>8.00 \mu s</td>
</tr>
<tr>
<td>1 1</td>
<td>use internal ‘RC’ (≈ 2 – 6 \mu s)</td>
<td></td>
</tr>
</tbody>
</table>

Select ADCS1, ADCS0 to 0,1 since this gives 2.0 \mu s time for $T_{AD}$.

So A/D conversion Time = $11.5 \times 2.0 \, \mu s = 23 \, \mu s$
Time for A/D Conversion - Summary

Figure 23-2: A/D Conversion Sequence

- A/D Sample Time
  - Acquisition Time
    - 19.75 \( \mu \text{s} \) (example)
  - A/D Conversion Time
    - 11.5 * TAD = 23 \( \mu \text{s} \)

A/D conversion complete, result is loaded in ADRES register. Holding capacitor begins acquiring voltage level on selected channel ADIF bit is set

When A/D conversion is started (setting the GO bit)
- When A/D holding capacitor starts to charge.
- After A/D conversion, or when new A/D channel is selected

Note - You must wait at least 2 * TAD before next acquisition is started.
# ADC

## The configuration ...

### TABLE 11-2: REGISTERS/BITS ASSOCIATED WITH A/D

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 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>
<th>Value on POR, BOR</th>
<th>Value on MCLR, WDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0Bh,8Bh,10Bh,18Bh</td>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RBIF</td>
<td>0000 000x</td>
<td>0000 000u</td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td>PSPIF(1)</td>
<td>ADIF</td>
<td>RCIF</td>
<td>TXIF</td>
<td>SSPIF</td>
<td>CCP1IF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>8Ch</td>
<td>PIE1</td>
<td>PSPIE(1)</td>
<td>ADIE</td>
<td>RCIE</td>
<td>TXIE</td>
<td>SSPIE</td>
<td>CCP1IE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>1Eh</td>
<td>ADRESH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxx xxxxx</td>
<td>uuuu uuuu</td>
</tr>
<tr>
<td>9Eh</td>
<td>ADRESI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxx xxxxx</td>
<td>uuuu uuuu</td>
</tr>
<tr>
<td>1Fh</td>
<td>ADCON0</td>
<td>ADCS1</td>
<td>ADCS0</td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td>—</td>
<td>ADON</td>
<td>0000 00-0</td>
<td>0000 00-0</td>
</tr>
<tr>
<td>9Fh</td>
<td>ADCON1</td>
<td>ADFM</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PCFG3</td>
<td>PCFG2</td>
<td>PCFG1</td>
<td>PCFG0</td>
<td>--0- 0000</td>
</tr>
<tr>
<td>85h</td>
<td>TRISA</td>
<td>—</td>
<td>—</td>
<td>PORTA Data Direction Register</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>--11 1111</td>
<td>--11 1111</td>
</tr>
<tr>
<td>05h</td>
<td>PORTA</td>
<td>—</td>
<td>—</td>
<td>PORTA Data Latch when written: PORTA pins when read</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>--0x 0000</td>
<td>--0u 0000</td>
</tr>
<tr>
<td>89h(1)</td>
<td>TRISE</td>
<td>IBF</td>
<td>OBF</td>
<td>IBOV</td>
<td>PSPMODE</td>
<td>—</td>
<td>PORTA Data Direction bits</td>
<td>0000 -111</td>
<td>0000 -111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09h(1)</td>
<td>PORTE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>RE1</td>
<td>RE0</td>
<td>xxxx xxxx</td>
<td>--- -uuu</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

**Note 1:** These registers/bits are not available on the 28-pin devices.
ADC
Use ... Flow Chart

- Configure analog pins/voltage reference and digital I/O (ADCON1)
- Select ADC input channel (ADCON0)
- Select ADC conversion clock (ADCON0)
- Turn on ADC module (ADCON0)

Configure A/D Module

Configure A/D Interrupt*

Wait for $T_{ACQ}$

Start Conversion (set GO)

Wait for conversion to complete

Read A/D Result

Wait for at least $2T_{AD}$ before next acquisition

*if desired
ADC

Use ... Flow Chart

Configure A/D Module

Configure A/D Interrupt* -
  - Clear ADIF bit
  - Set ADIE bit
  - Set PEIE bit
  - Set GIE bit

Wait for $T_{ACQ}$

Start Conversion (set GO)

Wait for conversion to complete

Read A/D Result

Wait for at least $2T_{AD}$ before next acquisition

*if desired
ADC
The interrupt ...

**TABLE 11-2: REGISTERS/BITS ASSOCIATED WITH A/D**

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 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>
<th>Value on POR, BOR</th>
<th>Value on MCLR, WDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0Bh, 8Bh, 10Bh, 18Bh</td>
<td>INTCN</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RBIF</td>
<td>0000 000x</td>
<td>0000 000u</td>
</tr>
<tr>
<td>0Ch</td>
<td>PIR1</td>
<td>PSPIF</td>
<td>ADIF</td>
<td>RCIF</td>
<td>TXIF</td>
<td>SSPIF</td>
<td>CCP1IF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>8Ch</td>
<td>PIE1</td>
<td>PSPIE</td>
<td>ADIE</td>
<td>RCIE</td>
<td>TXIE</td>
<td>SSPIE</td>
<td>CCP1IE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
<td>0000 0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>1Eh</td>
<td>ADRESH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxxx xxxxx</td>
<td>uuuuu uuuuu</td>
</tr>
<tr>
<td>9Eh</td>
<td>ADRESL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>xxxxxx xxxxx</td>
<td>uuuuu uuuuu</td>
</tr>
<tr>
<td>1Fh</td>
<td>ADCON0</td>
<td>ADCS1</td>
<td>ADCS0</td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td>—</td>
<td>ADON</td>
<td>0000 00-0</td>
<td>0000 00-0</td>
</tr>
<tr>
<td>9Fh</td>
<td>ADCON1</td>
<td>ADFM</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PCFG3</td>
<td>PCFG2</td>
<td>PCFG1</td>
<td>PCFG0</td>
<td>--0- 0000</td>
<td>--0- 0000</td>
</tr>
<tr>
<td>85h</td>
<td>TRISA</td>
<td>—</td>
<td>—</td>
<td>PORTA Data Direction Register</td>
<td>--11 1111</td>
<td>--11 1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05h</td>
<td>PORTA</td>
<td>—</td>
<td>—</td>
<td>PORTA Data Latch when written: PORTA pins when read</td>
<td>--0x 0000</td>
<td>--0u 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89h</td>
<td>TRISE</td>
<td>IBF</td>
<td>OBF</td>
<td>IBOV</td>
<td>PSPMODE</td>
<td>—</td>
<td>PORTE Data Direction bits</td>
<td>0000 -111</td>
<td>0000 -111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09h</td>
<td>PORTE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>RE2</td>
<td>RE1</td>
<td>RE0</td>
<td>——- —xxx</td>
<td>——- —uuu</td>
<td></td>
</tr>
</tbody>
</table>

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

Note 1: These registers/bits are not available on the 28-pin devices.
ADC

Use ... Flow Chart

Configure A/D Module

Configure A/D Interrupt*

Wait for $T_{ACQ}$

Start Conversion (set GO)

Wait for conversion to complete

Read A/D Result

Wait for at least $2T_{AD}$ before next acquisition

Use **POLLING**: Keep checking the DO/DONE bit, if it is cleared the conversion is done.

Use **INTERRUPTS**: Wait for the ADC interrupt

*if desired
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.
Example

Configure A/D Module

Configure A/D Interrupt*

Wait for $T_{ACQ}$

Start Conversion (set GO)

Wait for conversion to complete

Read A/D Result

Wait for at least $2T_{AD}$ before next acquisition
Example
Configure ADC Module

ADCON0 REGISTER (ADDRESS: 1Fh)

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADCS1</td>
<td>ADCS0</td>
<td>CHS2</td>
<td>CHS1</td>
<td>CHS0</td>
<td>GO/DONE</td>
<td>—</td>
<td>ADON</td>
</tr>
</tbody>
</table>

(bit 7) 0 1 0 0 1 0 0 1

Conversion clock 4MHz < 5MHz thus
ADCS1:ADCS0 = ’01’

RA1/AN1 thus
CHS2:CHS0 = ’001’

Turn on Module:
ADON = ’1’

ADCON1 REGISTER (ADDRESS 9Fh)

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADFM</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PCFG3</td>
<td>PCFG2</td>
<td>PCFG1</td>
<td>PCFG0</td>
</tr>
</tbody>
</table>

(bit 7) 1 0 0 0 0 0 0 0

Result is Right Justified
thus ADFM = ’1’

RA1/AN1 thus
PCFG3:PCFG0 = ’0000’
Configure ADC Module

ADCON0  equ  0x1F
ADCON1  equ  0x1F
org  0x000

INIT_ADC:
  movlw  B’01001001’
  movwf  ADCON0 ; Configure ADC via ADCON0
  bsf  STATUS, RP0 ; Access bank 1
  movlw  B’10000000’
  movwf  ADCON1 ; Configure ADC via ADCON1
  bcf  STATUS, RP0 ; Switch back to bank 0
...
...
...
...
Example

1. Configure A/D Module
2. Configure A/D Interrupt*
3. Wait for $T_{ACQ}$
4. Start Conversion (set GO)
5. Wait for conversion to complete
6. Read A/D Result
7. Wait for at least $2T_{AD}$ before next acquisition
Example

Wait for $T_{ACQ}$

Need a delay for about/at least 20μseconds:

```assembly
DELAY: movlw 0x05 ; 1
       movwf COUNT ; 1
NXT:   decfsz COUNT ; 1/2
       goto NXT ; 2
       return ; 2
```

DELAY duration including ‘call’ and return: $5 + 3 \times \text{COUNT}$

$5 + 3 \times \text{COUNT} = 20 \Rightarrow 3 \times \text{COUNT} = 15 \Rightarrow \text{COUNT} = 5$
Code
Main Loop

MAIN:
  call     DELAY          ; Wait for $T_{ACQ}$
  bsf      ADCON0,2      ; Start conversion (set GO)

CHECK:
  btfsc    ADCON0,2      ; Wait for conversion to complete*
  goto     CHECK          ; use POLLING

  movf     ADRESH,W      ; Read A/D result (high byte)
  movwf    TEMPH         ; Write to TEMPH

  bsf      STATUS,RPO    
  movf     ADRESL,W      ; Read A/D result (low byte)
  bcf      STATUS,RPO    
  movwf    TEMPL         ; Write to TEMPL

  nop       ; 1 Wait for 4 instruction cycles
  nop       ; 1
  goto MAIN ; 2

---

$2T_{AD} = 2 \times 8T_{OSC} = 2 \times 8 \times 0.25 \mu s = 4 \mu s = 4$ instruction cycles

*GO/DONE = 0
Code

Main Loop - Sample Period

... MAIN:
   call  DELAY  ; 20
   bsf    ADCON0,2  ; 1
   goto  CHECK  ;

   CHECK:
   btfsc  ADCON0,2  ; ~12  $T_{ad}$ = 24
   goto  CHECK  ;
   movf   ADRESH,W  ; 1
   movwf   TEMPH  ; 1
   bsf    STATUS,RPO  ; 1
   movf    ADRESL,W  ; 1
   movwf   TEMPL  ; 1
   bcf    STATUS,RPO  ; 1
   nop  ; 1
   nop  ; 1
   goto  MAIN  ; 2

Sample Period = $T_s \approx 20 + 1 + 24 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 2 = 55T_{CY} = 55\mu\text{sec}$

Sample Frequency = $1/T_s \approx 1/55\mu\text{sec} = 18,182\text{Hz} \approx 18\text{kHz}$
Nyquist Criterion

Why is this Sample Frequency Important?

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

ADC.53
Nyquist Criterion

Example

Range of human hearing:
20 to 20,000 Hz

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

For example:
- CD - 44.1kHz sampling rate
- DVD - up to 96 kHz

And CD - 16 bits per sample
ADC

Example ... Using Interrupts

ADCON0 equ 0x1F
ADCON1 equ 0x1F

org 0x000
goto INIT_ADC
org 0x004
goto AD_ISR

INIT_ADC:
  movlw B'01001001'
  movwf ADCON0 ; Configure ADC via ADCON0
  bsf STATUS, RP0 ; Access bank 1
  movlw B'10000000'
  movwf ADCON1 ; Configure ADC via ADCON1
  bcf STATUS, RP1

INIT_IRQ:
  bcf PIR1, ADIF ; Clear the ADIF flag (bit 6)
  bsf STATUS, RP0 ; Access bank 1
  bsf PIE1, ADIE ; Enable AADC interrupt
  bcf STATUS, RP0 ; Access bank 0
  bsf INTCON, PEIE ; Enable peripheral interrupts
  bsf INTCON, GIE ; Enable global interrupts

...
Code

Main Loop & Interrupt Service Routine

... call DELAY ; Wait for TACQ
bsf ADCON0,2 ; Start conversion (set GO)
goto MAIN ...

AD_ISR:

movf ADRESH,W ; Read A/D result (high byte)
movwf TEMPH ; Write to TEMPH
movf ADRESL,W ; Read A/D result (low byte)
movwf TEMPL ; Write to TEMPL
call DELAY ; Wait for TACQ
bsf ADCON0,2 ; Start conversion (set GO)
bcf PIR1, ADIF ; Clear the interrupt
retfie

*GO/DONE = 0
NOTE that in this case RA5 is a **DIGITAL** input and RA1 is an **ANALOG** input
INIT_ADC:

```
movlw  B'01001001'
movwf  ADCON0; Configure ADC via ADCON0
bsf   STATUS, RP0; Access bank 1
movlw  B'10000100'
movwf  ADCON1; Configure ADC via ADCON1
movlw  B'11111111'; Configure RA5 as an input pin
movwf  TRISA
bcf   STATUS, RP1
```