University of San Carlos | Department of
COMPUTER ENGINEERING

CpE 3201
Embedded Systems

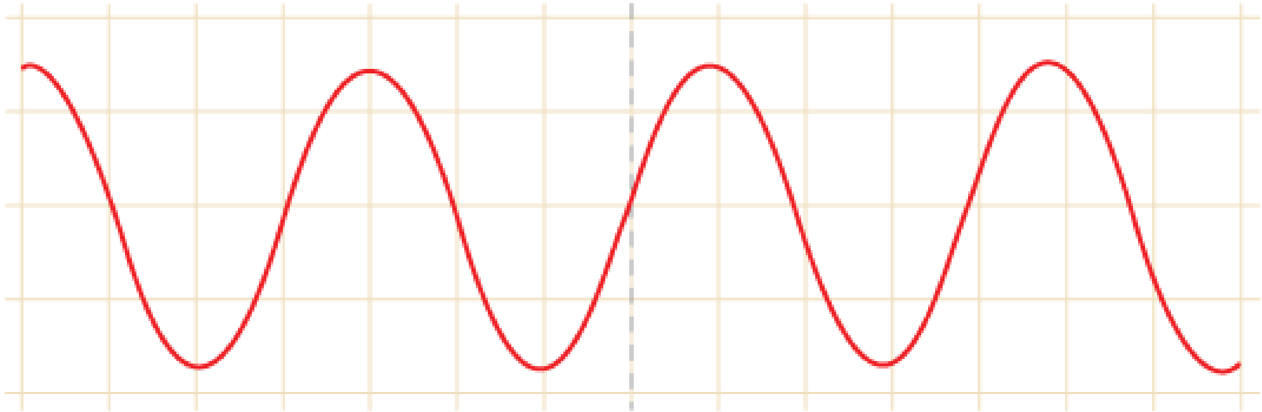
Analog to Digital Converter

Peripherals Features Covered:

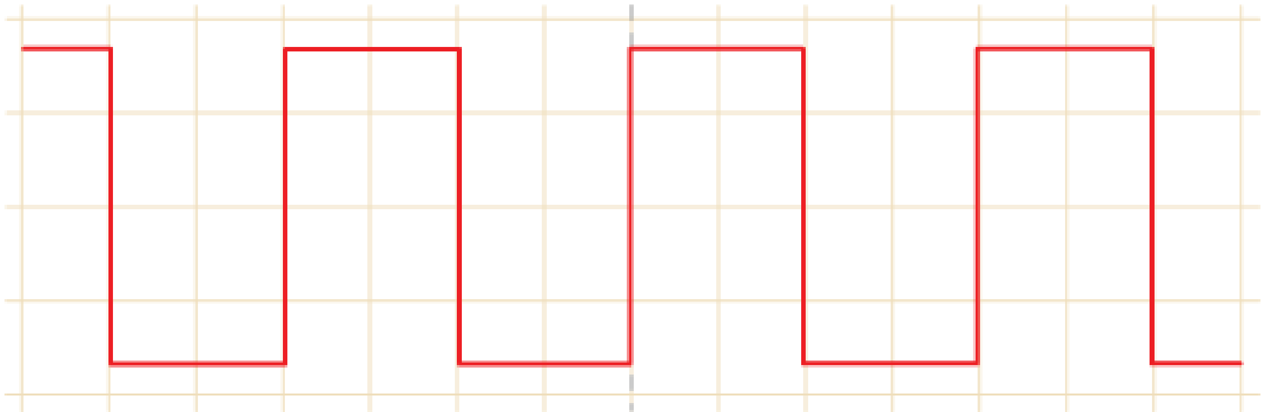
- Analog to Digital Converter (ADC)
- Universal Synchronous Asynchronous Receiver Transmitter (USART)
- Inter-Integrated Circuit Communication (I^2C)

Analog to Digital Conversion

- **Analog** signals are continuous with infinite values in a given range
- **Digital** signals have discrete values such as on and off, 0 and 1 or +5V and 0.7V.



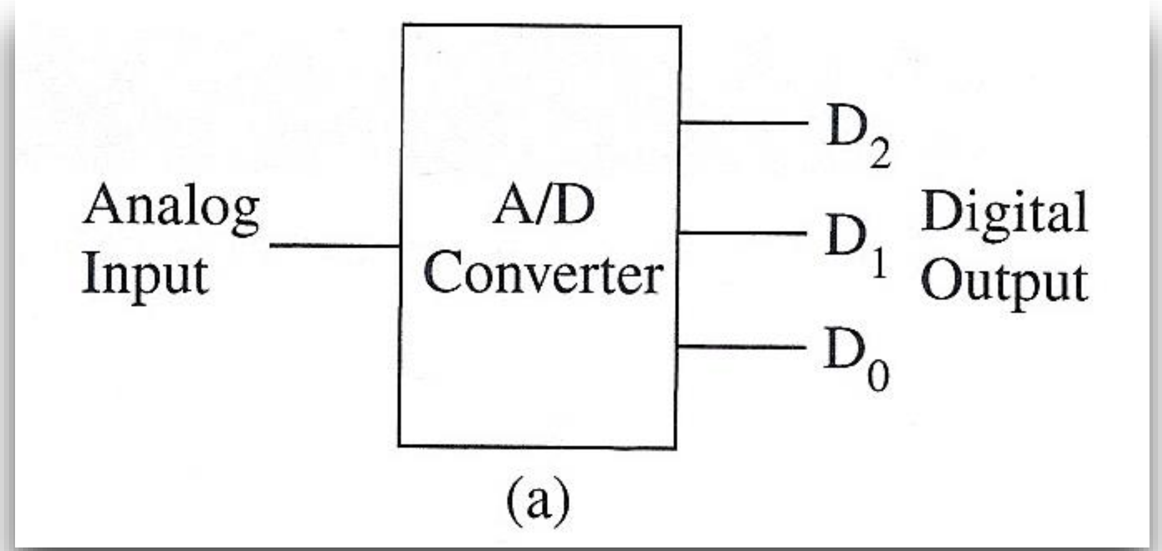
Analog Signal



Digital Signal

Conversion Concepts

- A/D conversion is the process of converting continuously varying signal such as voltage or current into discrete digital quantities that represent the magnitude of the signal compared to a standard or reference voltage.

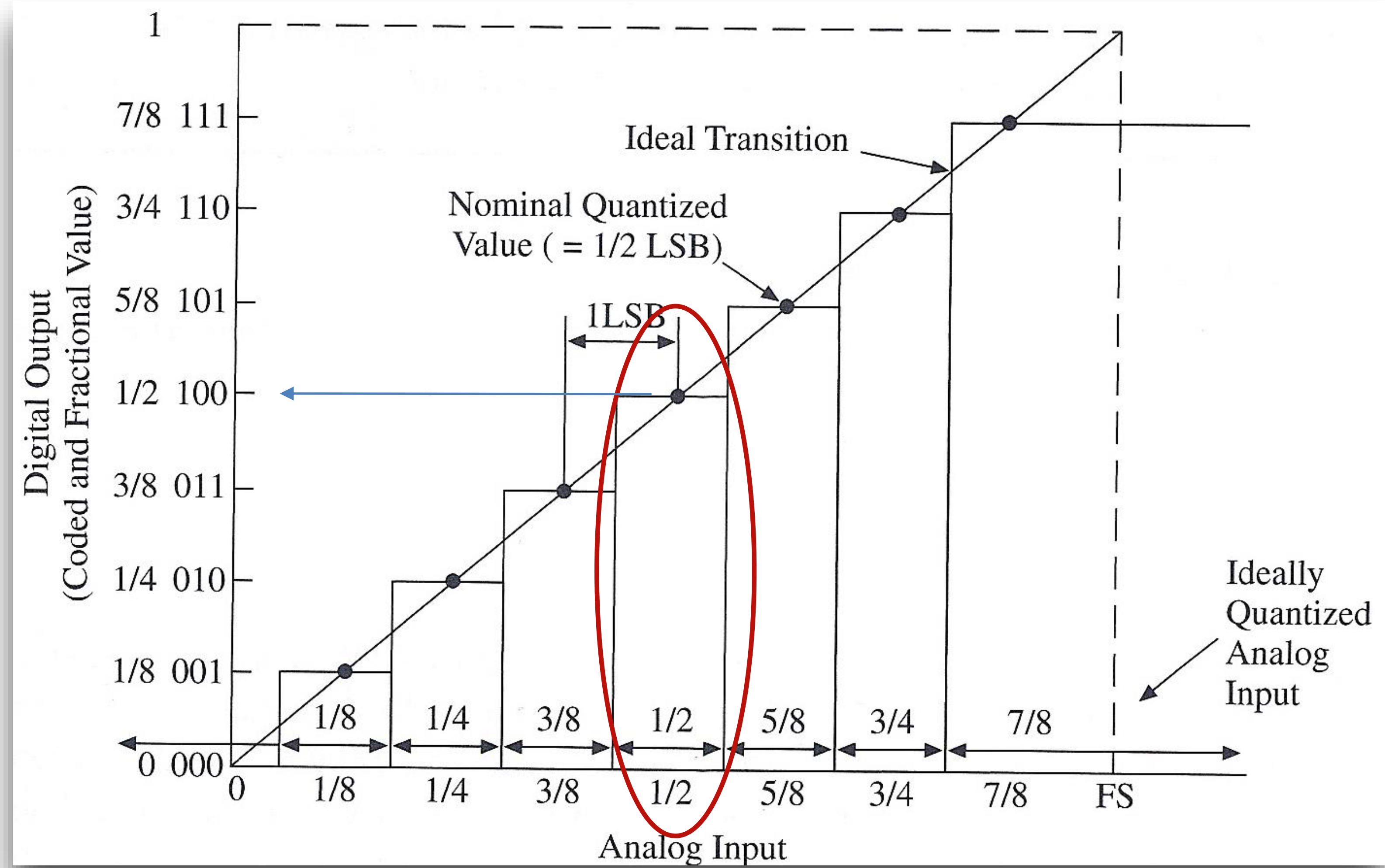


Hypothetical 3-bit A/D Converter

Reference Voltage: **0V - 1V**

Conversion Concepts

- A/D resolution: 3 bits
- No. of Voltage Steps: $2^3 = 8$
- Step Voltage: 1V / 8 steps = 0.125 V/step
- If analog input voltage is 0.5V, therefore:
  - 0.5V / 0.125V/step = 4 steps
  - Digital Value = “100”


1
7/8 111
3/4 110
5/8 101
1/2 100
3/8 011
1/4 010
1/8 001
0 000
Digital Output
(Coded and Fractional Value)
Ideal Transition
Nominal Quantized
Value (= 1/2 LSB)
1LSB
Ideally
Quantized
Analog
Input
1/8
1/4
3/8
1/2
5/8
3/4
7/8
0
1/8
1/4
3/8
1/2
5/8
3/4
7/8
FS
Analog Input

Example #1

- Given a 0 to 5V analog input signal and an 8-bit A/D converter, calculate the values LSB and MSB, resolution and the full-scale output.
- **Solution**
  - 2^8 = 256 steps from 00H to FFH
  - LSB (00000001_2) = 5V/256 steps = 19.53mV
  - MSB = (10000000_2) = 1/2 of full scale value = 2.5V
  - Resolution = 256 steps @ 19.53mV per step
  - Full-scale output = Full-scale Analog Signal - LSB = 5V - 0.01953V = 4.98 V

Example #2

- Calculate the voltage resolution for a 10-bit converter when the full-scale input voltage ranges from -5V to 5V.
- **Solution**
  - 2^{10} = 1024 steps from 000H to 3FFH
  - Voltage Range: (5-(-5)) = 10V
  - Voltage Resolution: 10V / 1024 steps = 9.76mV

Resolution

- Resolution defines the step voltage of an A/D converter. The higher the resolution, the finer the step voltage value and is ideal for minute changes in the analog signal. For example an 8-bit vs 10-bit A/D converters:
  - Steps: 256 vs 1024
  - Step voltage: 19.53mV vs 4.88mV

Conversion Methods

- Flash
  - uses multiple comparators in parallel
  - If the analog signal is higher than the known signal, the output of the comparator goes to 1
- Integrating
  - this A/D converter type charges a capacitor for a given amount of time using the analog signal
  - It discharges back to zero with a known voltage and the counter provides the value of the unknown signal

Conversion Methods

- Successive Approximation
  - this converter includes a D/A converter and a comparator
  - an internal analog signal is generated by turning on successive bits in the D/A converter
- Counter
  - similar to successive approximation circuitry
  - The counter starting from zero feeds the signal to the D/A converter instead of turning on successive bits

A/D Conversion Module (PIC16F877A)

- The 8-/10-bit A/D converter module of the PIC16F877A has 8-analog input channels.
- It has a 10-bit digital output.
- It has high and low-voltage reference input that is software selectable to some combination of VDD, VSS, RA2 or RA3.
- It is able to operate while the device is in Sleep mode.

A/D Module Registers

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits (ADCON0 bits in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	**00**	FOSC/2
0	**01**	FOSC/8
0	**10**	FOSC/32
0	**11**	FRC (clock derived from the internal A/D RC oscillator)
1	**00**	FOSC/4
1	**01**	FOSC/16
1	**10**	FOSC/64
1	**11**	FRC (clock derived from the internal A/D RC oscillator)

bit 5-3 **CHS2:CHS0:** Analog Channel Select bits

000 = Channel 0 (AN0)
001 = Channel 1 (AN1)
010 = Channel 2 (AN2)
011 = Channel 3 (AN3)
100 = Channel 4 (AN4)
101 = Channel 5 (AN5)
110 = Channel 6 (AN6)
111 = Channel 7 (AN7)

Note: The PIC16F873A/876A devices only implement A/D channels 0 through 4; the unimplemented selections are reserved. Do not select any unimplemented channels with these devices.

bit 2 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

When ADON = 1:
1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)
0 = A/D conversion not in progress

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

bit 0 **ADON:** A/D On bit

1 = A/D converter module is powered up
0 = A/D converter module is shut-off and consumes no operating current

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

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit

`1` = Right justified. Six (6) Most Significant bits of ADRESH are read as ‘`0`’.
`0` = Left justified. Six (6) Least Significant bits of ADRESL are read as ‘`0`’.

bit 6 **ADCS2:** A/D Conversion Clock Select bit (ADCON1 bits in shaded area and in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	FOSC/2
0	01	FOSC/8
0	10	FOSC/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	FOSC/4
1	01	FOSC/16
1	10	FOSC/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

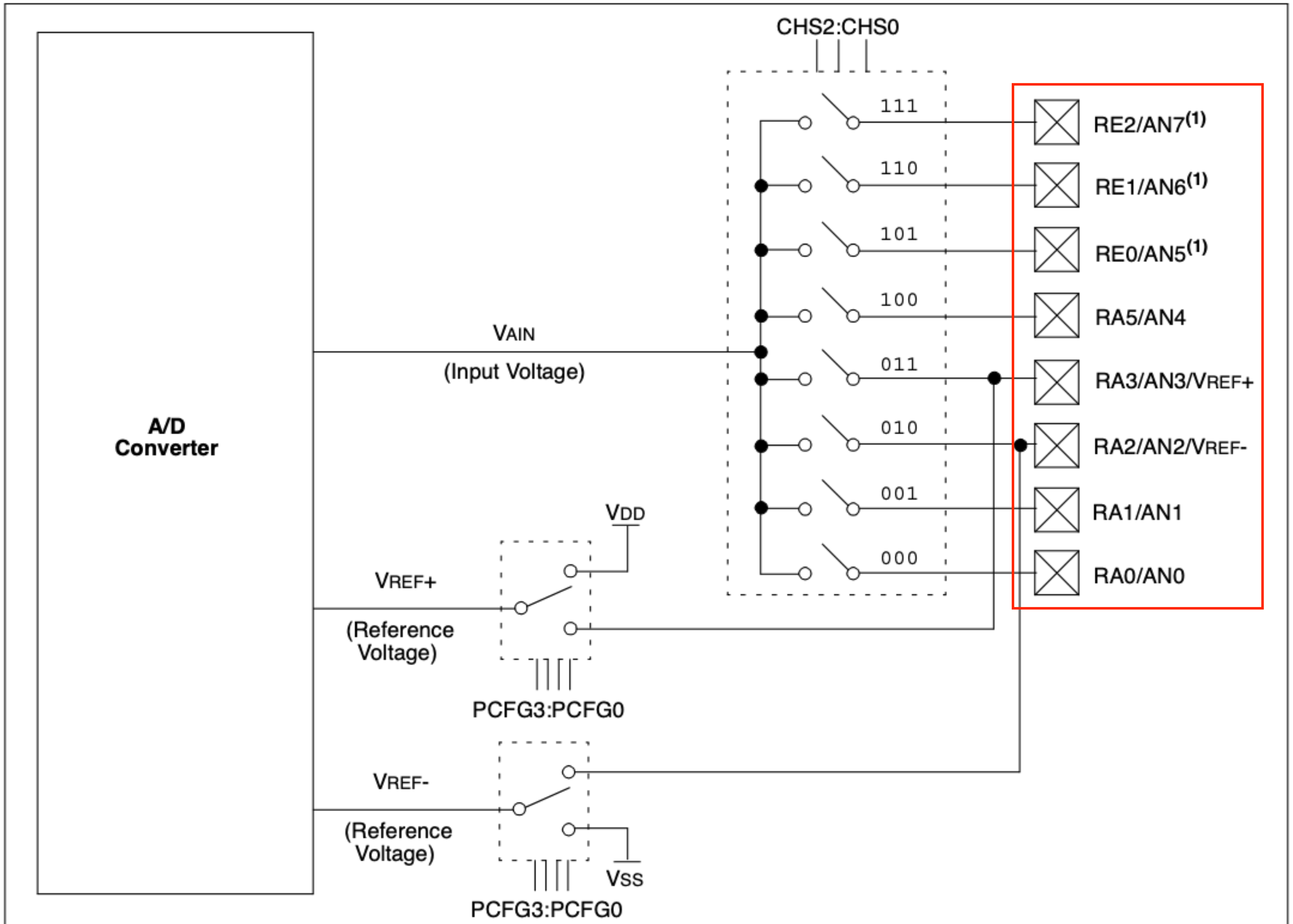
bit 5-4 **Unimplemented:** Read as ‘`0`’

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O
C/R = # of analog input channels/# of A/D voltage references

FIGURE 11-1: A/D BLOCK DIAGRAM



Note 1: Not available on 28-pin devices.

A/D Module Registers

- The **ADRESH:ADRESL** registers contain the 10-bit result of the A/D conversion.
- When the A/D conversion is complete, the result is loaded into this A/D Result register pair, the GO/DONE bit (ADCON0<2>) is cleared and the A/D interrupt flag bit ADIF is set.
- Interrupt enable ADIE is at PIE1 register and interrupt flag ADIF is at PIR1 register.

Steps in Implementing A/D Conversion

1. Configure the A/D module:
   - Configure analog pins/voltage reference and digital I/O (ADCON1)
   - Select A/D input channel (ADCON0)
   - SelectA/D conversion clock (ADCON0)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set PEIE bit
   - Set GIE bit

* Analog input pins are automatically set to input once configured as “analog”.

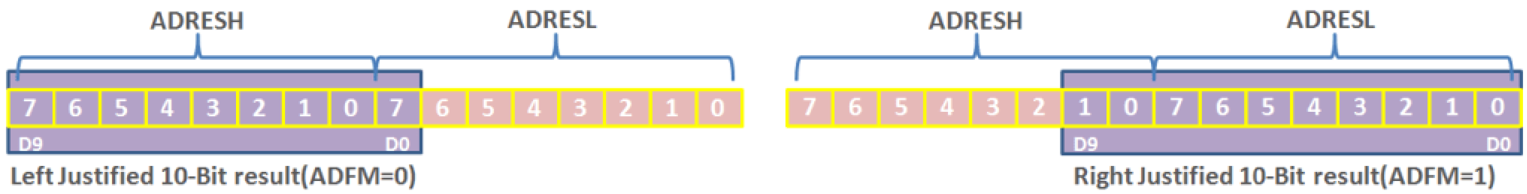
Steps in Implementing A/D Conversion

3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/DONE bit (ADCON0)
5. Wait for A/D conversion to complete by either:
   - Polling for the GO/DONE bit to be cleared (interrupts disabled); OR
   - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For the next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as T_{AD}.

* Analog input pins are automatically set to input once configured as “analog”.

A/D Converter Resolution

- The digital result is 10-bits at ADRESH:ADRESL.
- Depending on the A/D Result Format Select (ADFM) bit in the ADCON1 register, the 10-bit data can either be right or left justified.



A/D Converter Resolution

- Let's say that the V_{REF} is 5V. Converting the analog input using the 10-bit resolution will have a step voltage of:

$$StepVoltage = \frac{V_{REF}}{2^{resolution}}$$

$$StepVoltage = \frac{5V}{2^{10}} = 4.882mV$$

- The digital value range of 0-169 will have an analog value range of:

$$169x4.882mV = 0.825V$$

Therefore the equivalent analog range is:

$$0V - 0.825V$$

Example (A/D Module)

- A variable resistor is connected to AN0 (RA0) with a reference voltage of +5V and five LEDs connected to PORTB <RB4:RB0>. The LEDs shall light up depending on the applied analog signal.

Applied voltage	Obtained A/D value	LED lighting
0 to 0.83 V	0 to 169	No LED is on.
0.83 to 1.67 V	170 to 340	LED1 is on.
1.67 to 2.50 V	341 to 511	LEDs 1 to 2 are on.
2.50 to 3.33 V	512 to 682	LEDs 1 to 3 are on.
3.33 to 4.17 V	683 to 853	LEDs 1 to 4 are on.
4.17 to 5.00 V	854 to 1024	All LEDs are on.

```
void main(void)
{
  int d_value = 0;
  TRISB = 0x00;     // set all PORTB as output
  PORTB = 0X00;     // all LEDs OFF
  ADCON1 = 0x80;    // result register: right justified, clock: Fosc/2
                    // all ports in PORTA are analog
                    // VREF+=VDD, VREF-=VSS
  ADCON0 = 0x01;    // clock: Fosc/2, analog channel: AN0
                    // A/D conversion: STOP, A/D module: ON

  for(;;)          // foreground routine
  {
      d_value = readADC();
      /* setting the LEDs */
      if(d_value>=0 && d_value<=169)
              PORTB = 0x00;          // all LEDs OFF
      else if(d_value>=170 && d_value<=340)
              PORTB = 0x01;          // RB0 LED ON
     …
     …


  }
}
```

For this example the step voltage is 4.88mV. Therefore from 0 to 169, the voltage range is from 0V to 0.83V (see table).

```
int readADC(void)
{
    int temp = 0;
    delay(1000);                 // delay before reading value
    GO = 1;                      // start A/D conversion (ADCON0 reg)
      while(GO_DONE==1);         // wait until conversion is done (ADCON0 reg)
    /* read result register */
    temp = ADRESH;               // read ADRESH
    temp = temp << 8;            // move to correct position
    temp = temp | ADRESL;        // read ADRESL
    return temp;
}
void delay(int cnt)
{
    while(cnt--);
}
```

GO bit is automatically reset when conversion is done (GO_DONE='0').

Using Interrupts

- When implementing A/D converter using interrupts, a slower conversion clock must be used.
- In the ISR, ADIF should be checked. The value should be ‘1’ when the conversion is complete.
- In the next example, the conversion clock used if Fosc/32.

```
void main(void)
{
  TRISB = 0x00;     // set all PORTB as output
  PORTB = 0x00;     // all LEDs OFF
  ADCON1 = 0x80;    // result register: right Justified, clock: Fosc/8
                    // all ports in PORTA are analog
                    // VREF+=VDD, VREF-=VSS
  ADCON0 = 0x41;    // clock: Fosc/8 analog channel: AN0
                    // A/D conversion: STOP, A/D module: ON
  ADIE = 1;         // A/D conversion complete interrupt enable (PIE1 reg)
  ADIF = 0;         // reset interrupt flag (PIR1 reg)
  PEIE = 1;         // enable all peripheral interrupt (INTCON reg)

  GIE = 1;          // enable all unmasked interrupts (INTCON reg)
  GO = 1;           // start A/D conversion (ADCON0 reg)

  for(;;)          // foreground routine
  {


  }
}
```

```
void interrupt ISR(void)
{
  int d_value = 0;
  GIE = 0;                        // disable all unmasked interrupts (INTCON reg)
   if(ADIF==1)                    // checks CCP1 interrupt flag
   {
      ADIF = 0;                   // clears interrupt flag (INTCON reg)
      /* read result register */
      d_value = ADRESH;                 // read ADRESH
      d_value = d_value << 8;           // move to correct position
      d_value = d_value | ADRESL;       // read ADRESL
      /* setting the LEDs */
      if(d_value>=0 && d_value<=169)
         PORTB = 0x00;            // all LEDs OFF
      else if(d_value>=170 && d_value<=340)
         PORTB = 0x01;            // RB0 LED ON
      …
      …
   }
   GO = 1;                        // restart A/D conversion (ADCON0 reg)
   GIE = 1;                       // enable all unmasked interrupts (INTCON reg)
}
```

University of San Carlos | Department of
COMPUTER ENGINEERING

CpE 3201
Embedded Systems

End of Lecture

References:

- De Leon, Hilary L., Microcontroller Programming and Interfacing, 2006.
- Goankar, Ramesh, Fundamentals of Microcontrollers and Applications in Embedded Systems (with the PIC18 MCU Family),2007.
- PIC16F87X Data Sheet, Microchip Technology Inc. 2003.