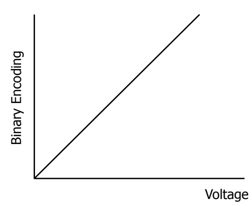
ECE 372 – Microcontroller Design

Analog to Digital Conversion

ECE 372 – Microcontroller Design

Analog to Digital Conversion (ADC)

- Characteristic of an Ideal A/D Converter
  - Needs infinite number of bits to encode the ADC conversion result

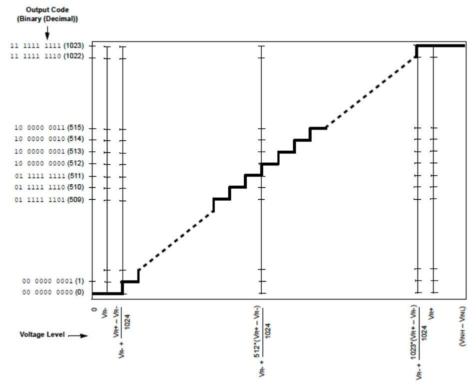


Ideal ADC Output

ECE 372 – Microcontroller Design

Analog to Digital Conversion (ADC)

- Analog to Digital Converter (ADC)
  - Use ***n*** bits to represent voltage input
  - Stepwise conversion/quantization of infinite voltage levels to discrete binary encodings
    - Quantization error is difference between encoded value and real voltage input



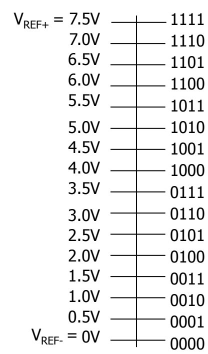
  - ADC Range = $V_{REF+} - V_{REF-}$
  - ADC Resolution = $(V_{REF+} - V_{REF-})/(2^n - 1)$

  - ***Note***: *resolution is <u>NOT</u> the same as accuracy*

ECE 372 – Microcontroller Design

Analog to Digital Conversion (ADC)

- ADC typically uses a low reference voltage (V_{REF-}) and a high reference voltage (V_{REF+}) to perform conversion
  - Most ADCs are *ratiometric*
    - An analog input of V_{REF-} is converted to digital code 0
    - An analog input of V_{REF+} is converted to digital code $2^n – 1$
  - The A/D conversion result k corresponds to the following analog input
    - $V_K = V_{REF-} + k*(\text{ADC Resolution})$
    - $V_k = V_{REF-} + k*((V_{REF+} - V_{REF-}) * k)/(2^n – 1)$



Proportionality/ ratiometric

ECE 372 – Microcontroller Design

Analog to Digital Conversion (ADC)

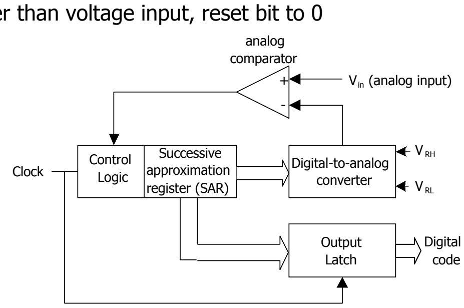
- Given a 10-bit ADC with V_{REF-} = 1 V and V_{REF+} = 4V, what voltage corresponds to a digital value of 720?

  - V_{720} = 1V + 720* (5 - 1) / (2^{10} – 1) = 3.11 V

ECE 372 – Microcontroller Design

Analog to Digital Conversion (ADC) – Successive Approximation

- Successive Approximation Conversion Method
  - Most commonly used A/D conversion method for 8- and 16-bit microcontrollers
  - Procedure:
    - Initialize all bits in register to 0
    - Starting with most significant bit (MSB) to least significant bit
      - Set bit to 1
      - If current voltage is greater than voltage input, reset bit to 0



ECE 372 – Microcontroller Design

Analog to Digital Conversion (ADC) – Successive Approximation Example

- Successive Approximation Conversion Method
  - Given an 8-bit ADC with an analog input whose voltage can range from 0 *(V_{REF-})* to 15 *(V_{REF+})* volts, using the successive approximation conversion approach, find the encoding for an input of 5 V.

Estimate	b7	b6	b5	b4	b3	b2	b1	b0
$V_{estimated}$ = 7.5 volts > 5	0	0	0	0	0	0	0	0
$V_{estimated}$ = 3.75 volts < 5	0	1	0	0	0	0	0	0
$V_{estimated}$ = 5.63 volts > 5	0	1	0	0	0	0	0	0
$V_{estimated}$ = 4.69 volts < 5	0	1	0	1	0	0	0	0
$V_{estimated}$ = 5.16 volts > 5	0	1	0	1	0	0	0	0
$V_{estimated}$ = 4.93 volts < 5	0	1	0	1	0	1	0	0
$V_{estimated}$ = 5.05 volts > 5	0	1	0	1	0	1	0	0
$V_{estimated}$ = 4.99 volts < 5	0	1	0	1	0	1	0	1

ECE 372 – Microcontroller Design

ADC Configuration Registers

- Basic ADC Configuration
  - AD1CON1:
    - ADON: enable ADC
    - FORM: controls format of digital encoding
    - SSRC: controls sampling and conversion periods
    - ASAM: configures auto-sampling mode that internally controls sampling/conversion process *(recommended)*
    - SAMP: controls start of ADC conversion
  - AD1CON2:
    - VCFG: configures voltage reference source
    - SMPI: configures how many samples/conversions will lead to setting ADxIF
    - BUFM: configures ADC conversion buffer
    - ALTS: configures using MUXA or alternating between MUXA and MUXB
    - ADRC: ADC clock source
    - SAMC: Number of sampling cycles (T_{AD}) for ADC sampling period
    - ADCS: Number of conversion cycles (T_{CY}) for ADC conversion period

ECE 372 – Microcontroller Design

ADC Configuration Registers

- Basic ADC Configuration
  - AD1CON3:
    - CH0SA: configures which analog input (*ANx*) is connected to MUXA
    - CH0SB: configures which analog input (*ANx*) is connected to MUXB
  - AD1PCFG
    - Configures ports on PIC24F as analog (0) or digital (1)
    - Each ANx input is associated with 1 bit in AD1PCFG register
  - AD1IF:
    - Interrupt flag for ADC
  - AD1BUF0 – AD1BUF15:
    - Registers for reading result of ADC conversion
    - 16 entries within ADC conversion buffer
    - ADC configurations control how buffer is used
  - *This sure doesn't look basic ☹*