

Dedan Kimathi University of technology

Department of Electrical and Electronic  
Engineering

EEE2406: Instrumentation

Lab 2

Title: Analogue to Digital Conversion

October 2, 2015

# 1 Analogue to Digital Conversion

## 1.1 Objectives

The aim of this laboratory practice is to

- Introduce the student to the concept of analogue to digital conversion in instrumentation systems
- Introduce the student to microprocessor application in analogue to digital conversion in instrumentation systems.
- Develop an analogue to digital converter based on the Raspberry Pi

## 1.2 Requirements

1. 10  $k\Omega$  variable resistor
2. MCP3008 chip (analogue to digital converter)
3. Raspberry Pi with with necessary accessories
4. Breadboard and connectors

# 2 Theory

## 2.1 Introduction

Modern design trends use the power and precision of the digital world of components to process analogue signals. However, the link between the digital/processing world and the analogue/real world is based on the analogue-to-digital and digital-to analogue converter ICs, which generally are grouped together as the data converters [1]

An analogue to digital converter takes an analogue input voltage and produces a digital output code which represents the analogue input. There are many methods of analogue to digital conversion, the most common methods being [1].

### 1. Digital Ramp ADC

A digital ramp analog-to-digital converter is shown in Figure 1 (For the comparator shown in the gure, if  $V_A > V_B$ ,  $EOC = \text{HIGH}$ , and  $V_A < V_B$ ,  $EOC = \text{LOW}$ )

A START pulse is applied and this resets the counter to zero. When the START pulse is HIGH, it inhibits the AND gate so that no clock pulses get through to the counter. With the counter set to zero,  $V_B = 0$  so the comparator output is HIGH. When the START pulse returns LOW, the AND gate is enabled and pulses allowed into the counter. As the counter advances, the DAC's output  $V_B$  increases in steps of voltage equal to its resolution. When  $V_A$  reaches  $V_B$ , the comparator output goes LOW, inhibiting pulses to the counter so that the counter stops and holds the desired digital code representing  $V_A$ . The HIGH to LOW transition at the comparator output signals the End Of Conversion (EOC). This is illustrated by Figure 2. The conversion time is the interval between  $t_0$  and  $t_1$ .

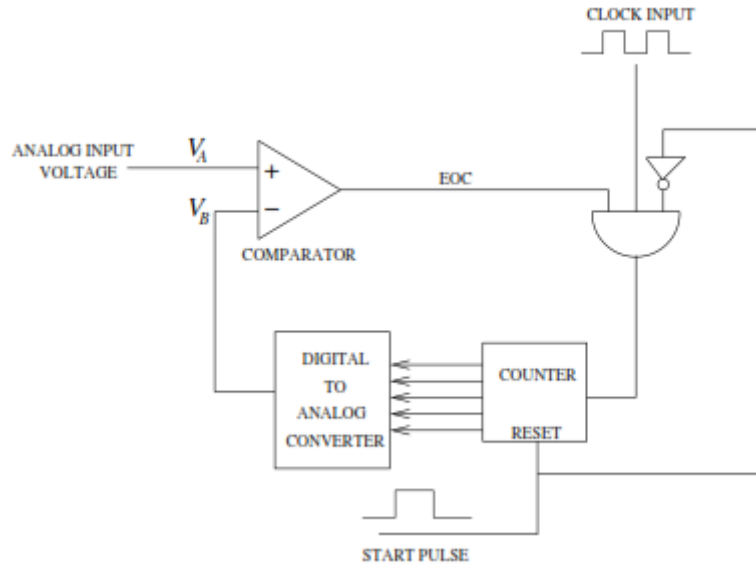


Figure 1: Digital Ramp ADC

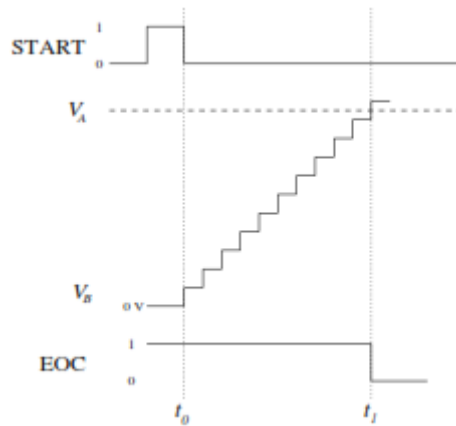


Figure 2: Digital Ramp ADC timing

Note that the conversion time for this converter is not constant: the higher the analogue input, the higher the conversion time.

## 2. Successive Approximation Converter

This method is essentially a binary search. It works by trying various binary codes and feeding them into a DAC and comparing the result with the analogue input via a comparator. This converter is illustrated in Figure 3. (For the comparator shown in the Figure, if  $V_A > V_B$ ,  $\text{COMP} = \text{HIGH}$ , and if  $V_A < V_B$ ,  $\text{COMP} = \text{LOW}$ .)

The control logic sets the MSB of the register HIGH and all the other bits LOW. This produces a value of  $V_B$  at the DAC output that is equal to the weight of the MSB. If  $V_B$  is now greater than  $V_A$ , the comparator output goes LOW and causes the control logic to clear the MSB back to LOW. Otherwise the MSB is kept HIGH.

The control logic sets the next bit of the register to 1. This produces a new value

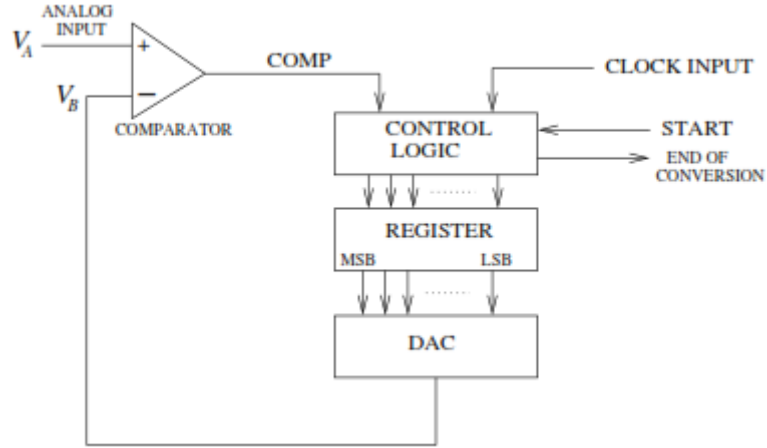


Figure 3: Successive Approximation Converter

of  $V_B$ . If the value of  $V_B$  is greater than  $V_A$ , the comparator output goes LOW to tell the control logic to clear the bit back to 0. Otherwise the bit is kept at 1.

The process is repeated for each of the bits in the register. After all the bits have been tried, the register holds the digital equivalent of  $V_A$ .

### 3. Dual Slope Integration Converter

The block diagram of the dual-slope integration converter is shown in Figure 4.

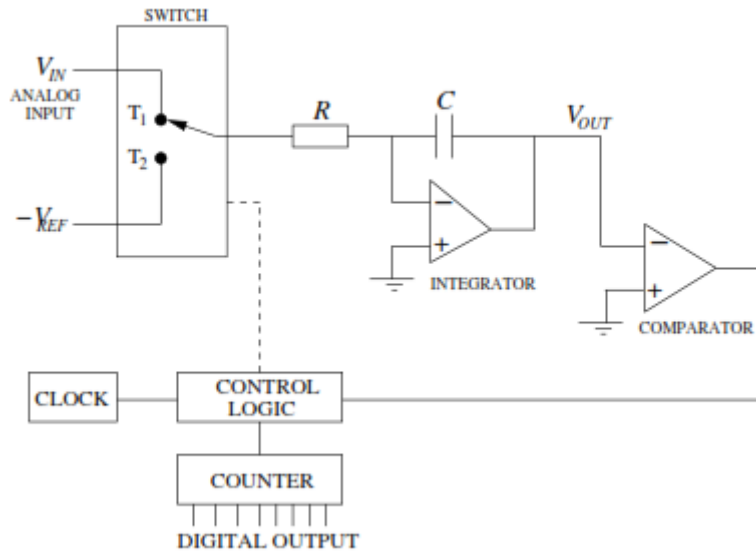


Figure 4: Dual Slope Integration Converter

The conversion begins with the electronic switch in the  $T_1$  position, with the counter reset to zero and the capacitor fully discharged. The input voltage causes the capacitor connected to the op-amp to charge at a rate dependent on the magnitude

of the input voltage.

After  $T_1$  seconds have elapsed, the switch is moved to  $T_2$  by the control logic. The higher the input voltage  $V_{IN}$ , the greater the charge stored in the capacitor during  $T_1$ . Because the reference voltage is of the opposite polarity to the analogue input, the capacitor now discharges. The control logic also starts the counter the moment the switch is moved to  $T_2$ . When the capacitor has fully discharged, the comparator output level switches and the counter is halted. The value in the counter is directly proportional to the input voltage  $V_{IN}$

## 2.2 MCP3008 Analogue to Digital Converter

The Microchip Technology Inc. MCP3004/3008 devices are successive approximation 10-bit Analogue to-Digital(A/D) converters with on-board sample and hold circuitry. single-ended inputs. The MCP3008 is programmable to provide four pseudo-differential input pairs or eight single-ended inputs. It comes as a 16 pin IC, see Figure 5.

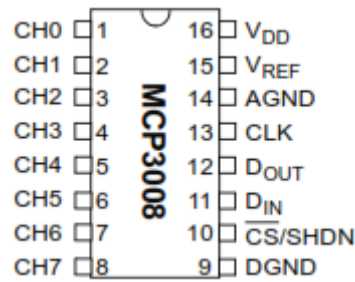


Figure 5: MCP3008 IC schematic

The analogue input is fed to any of the input channels (pins 1-8, labelled CH0-CH7). MCP3008 being a 10 bit analogue to digital converter gives a decimal output of values between 0 and 1023 ( $2^{10} = 1024$ ) depending on the voltage supplied to the analogue reference pin ( $V_{REF}$  pin 12). For complete operation of the chip, see attached data sheet.

## 3 Procedure

The potentiometer is connected onto the breadboard as follows; The first pin is connected to 5 V from the Pi (pin 2 on Pi) while the last pin is grounded (connected to pin 39 on Pi).

The output of the potentiometer (middle pin 6) is connected to channel 0 (pin 1) of the MCP3008 chip.

The other pins of the MCP3008 are connected as follows

- $V_{DD}$  pin (pin 16) connected to 5 V (Raspberry Pi pin 1)
- $V_{REF}$  pin (pin 15) connected to 5 V (Raspberry Pi pin 1)

- *AGND* pin (pin 14) connected to ground (Raspberry Pi pin 39)
- *CLK* pin (pin 13) connected to General Purpose Input Output GPIO pin 18 (Raspberry Pi pin 12)
- *D<sub>OUT</sub>* pin (pin 12) connected to General Purpose Input Output GPIO pin 23 (Raspberry Pi pin 16)
- *D<sub>IN</sub>* pin (pin 11) connected to General Purpose Input Output GPIO pin 24 (Raspberry Pi pin 18)
- *CS* pin (pin 10) connected to General Purpose Input Output GPIO pin 25 (Raspberry Pi pin 22)
- *DGND* pin (pin 9) connected to ground (Raspberry Pi pin 39)

A program to read the digital equivalent of the output of the potentiometer is written in python. See attached Appendix I.

The program is run and by adjusting the potentiometer, different readings are obtained for the analogue input from the potentiometer. The output voltage is measured using a multimeter and corresponding decimal value is displayed on the screen. This information is tabulated as shown below

SR. No	Output Voltage	Decimal Equivalent
1		
2		
3		
4		
5		

## 4 Results

Provide results for at least 10 different analogue inputs

## 5 Discussion

Compare the actual outputs obtained with the expected outputs. Carry out error analysis

## 6 Conclusion

Make conclusion based on the outcome of practical

## References

- [1] N. Kularatna, *Digital and Analogue Instrumentation*. Institution of Engineering and Technology, 2003.