Sequential Digital Data Acquisition

James Holekamp
Agricultural Science & Engineering Technology
Sam Houston State University
Huntsville, TX 77341 United States

***Abstract*—This paper covers a digital data acquisition system that transmits information using time division multiplexing. The data acquisition system consists of digital to analog converters, multiplexers, counters, and code. The system is used to examine the accuracy and reliability of low-resolution data transmitted via time division multiplexing and decoded in LabVIEW. Due to the size and detail of the diagrams for this report, larger zoomed-in diagrams will be attached to the end of the paper.**

I. Introduction

In this paper, a sequential digital data acquisition system is developed and analyzed. The content of this paper will include a technical overview of the design of the system and provide an in-depths analysis of the results obtained from its data acquisition. The digital data acquisition system is designed to convert two analog signals to an 8-bit digital value, which undergoes time division multiplexing for sequential data transmission to LabVIEW. After the signal has been transferred, a software demuxer made in LabVIEW is used to decode the information for interpretation and analysis. The system is made of three core components: 8-bit Analog to digital converters, 16-to-1 multiplexing circuit, and software based de-multiplexor.

II. Analog to Digital Converter

The 8-bit analog to digital converter (ADC) circuits are constructed using two ADC0804 integrated circuits. The ADC0804 IC is a CMOS 8-bit successive approximation converter made by Texas Instruments [http://www.ti.com/lit/ds/symlink/adc0802-n.pdf]. The ADCs are used in the system to quantize and encode a differential analog voltage from one or two transducers and output the digitized signal to the inputs of the multiplexing circuit. A block diagram of the ADC0804 is shown in the figure below.

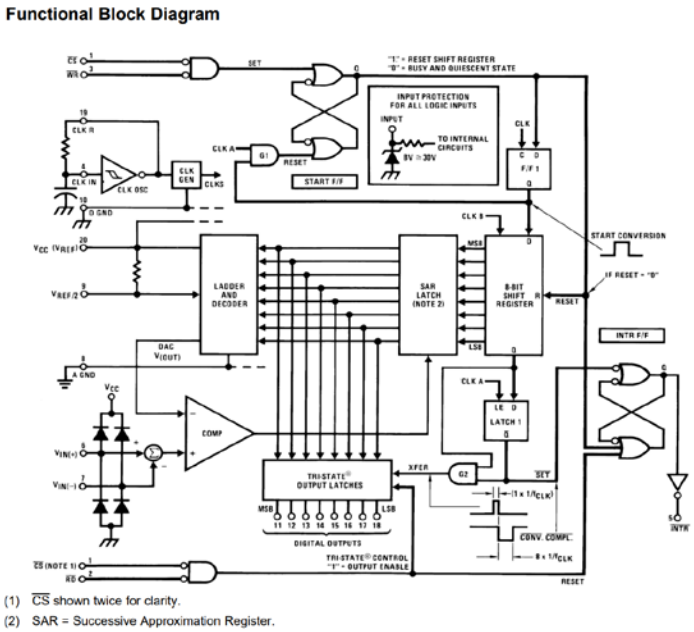


Figure 1. Functional Block Diagram of the ADC0804

Understanding the operation of the successive approximation type ADC was crucial when building the circuit. As shown in figure 1, a comparator, successive approximation register, and digital to analog converter (DAC) are used to construct a closed-loop feedback system to improve the accuracy of the digitized output. The closed loop feedback system works by initializing the most significant bit as one, which is converted to an analog value that matches approximately one half of the voltage supplied to the ADC. This parameter is initialized because the binary value 10000000 is the halfway point of an 8-bit range of values. A comparator is then used to compare this value to the analog input signal. If the voltage of the input signal is greater than the voltage from the DAC, then the bit is set to one and stored in the SAR. If the input voltage is less than the voltage from the DAC, the bit is set to zero and the next bit is tested.

III. Quantization Error and Resolution

For an ADC, resolution is defined as the smallest change in an analog signal that will result in a change in

the digital output [1]. An ADC's resolution can be calculated using the following equation:

$$\Delta V_{resolution} = V_{reference\ range} / 2^{N\ bits}$$

Since the ADC circuit in this project has a voltage reference range of 5V and an 8-bit output, the resolution is equal to 19.53 mV. This means that the least significant bit of the digital output will be equal to 19.53 mV and that each binary increment will represent an additional 19.53 mV. The quantization error of an ADC is directly related to its resolution. The resolution represents the quantization error inherent in the conversion of the signal to digital form [1]. Quantization error can be calculated using the following equation:

$$Quantization\ Error = \pm\frac{1}{2}\Delta V_{resolution}$$

Based on this calculation, the quantization error for this project is equal to ±9.77 mV. Figure 2 shows the quantization error of the ADC0804 IC.

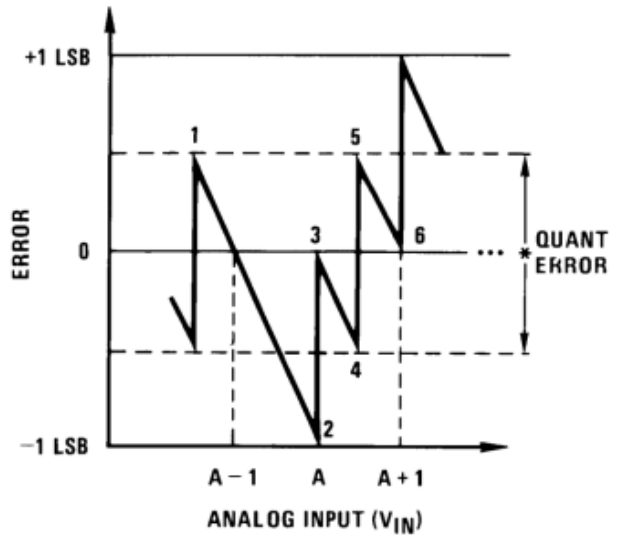


Figure 2 ADC0804 Quantization Error [2]

IV. MULTIPLEXING CIRCUIT

The analog to digital converters output to a 16-to-1 time-division multiplexing circuit constructed from three 8-to-1 multiplexor ICs and a 4-bit counter made in LabVIEW. The multiplexer IC used in this circuit is the SN74151A by Texas Instruments. The T74S151B1 has eight digital inputs, 1 digital output, and three selector inputs. Figure 3 shows the logic symbols for the multiplexer.

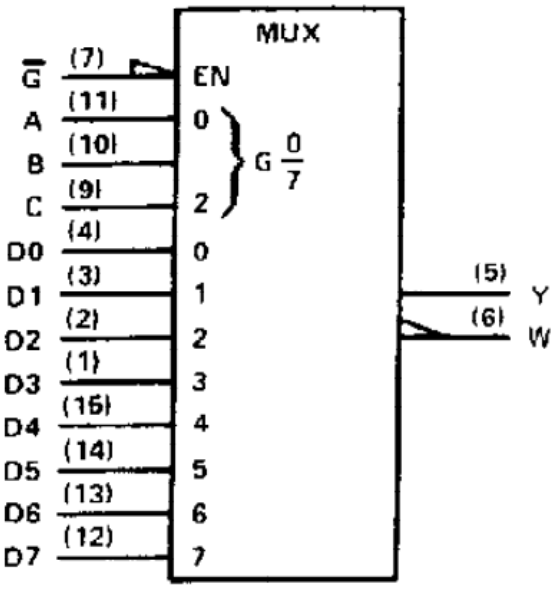


Figure 3 SN74151A Logic Symbols [3]

The binary sequence of selector inputs *A*, *B*, and *C* are used to control which input bit will be displayed at output *Y*. The figure below shows the function table for the multiplexor's selector bits.

INPUTS				OUTPUTS	
SELECT			STROBE		
C	B	A	$\overline{G}$	Y	W
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

Figure 4 SN74151A Function Table [3]

If a 3-bit counter is connected to input bits A, B, and C, then output Y will sequentially output all digital input bits, which can then be decoded to reconstruct the full 8-bit binary sequence. The same principal applies to the multiplexing circuit constructed in this project. However,

the principal is extended to use a 4-bit counter and 16-to-1 multiplexor. To make a 16-to-1 multiplexer from three 8-to-1 multiplexors, the output of mux 1 and mux 2 are fed into input bits D0 and D1 of mux 3, and selector bits A, B, and C of mux 1 and 2 are connected together. The most significant bit in the 4-bit counter is then connected to selector bit A of mux 3. Figure 5 shows a schematic for the 16-to-1 multiplexing circuit

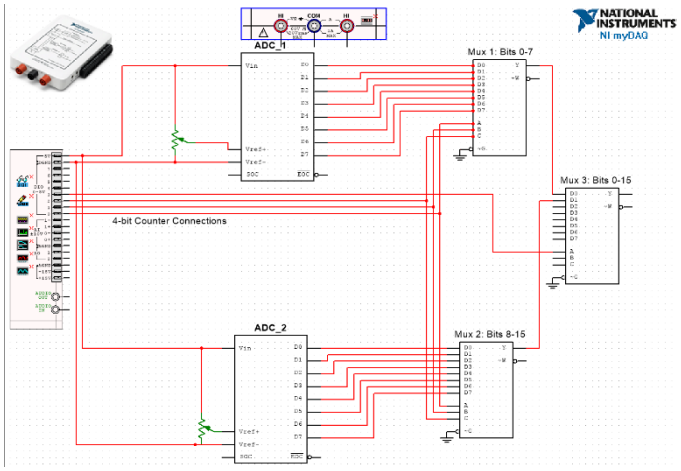


Figure 5 ADC to 16:1 multiplexer

As shown in figure 5, bits 1-3 of the 4-bit counter will simultaneously control mux 1 and mux 2. While the most significant bit of the counter is low, D0 of mux 3 will output bits 0-7 of the multiplexing circuit. When the most significant bit of the counter goes high, D1 of mux 3 will then output bits 8-15 of the circuit.

The counter was configured to increment by one every 40 milliseconds. This allows the transmission rate of the multiplexer to operate at a rate of 25 bits per second. The counter was constructed in LabVIEW and is made entirely from logic gates. The logic gates are configured to make a sequence of flip-flop circuits that are linked together to form a 4-bit counter. Figure 6 shows a block diagram of the 4-bit counter.

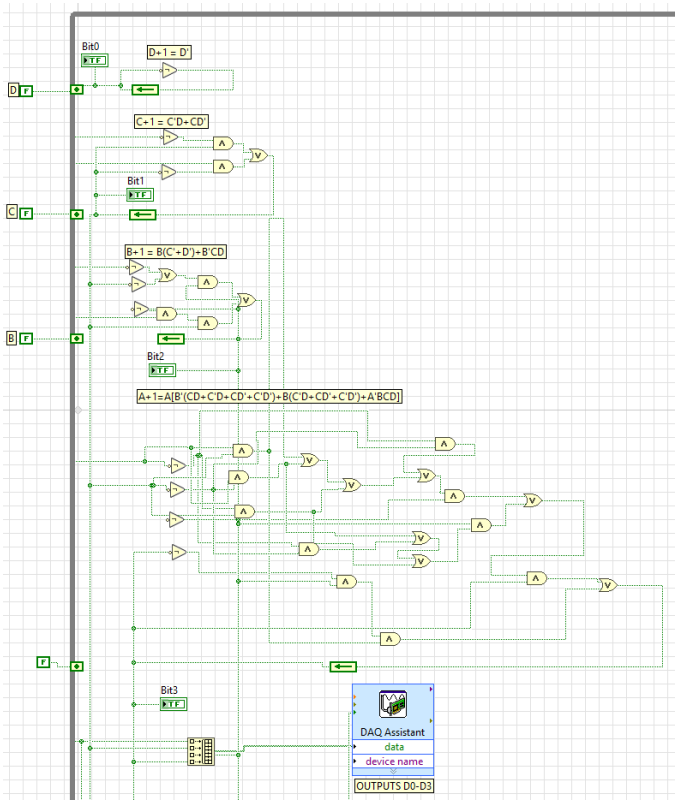


Figure 6 LabVIEW 4-Bit Counter

V. DE-MULTIPLEXING

The de-multiplexing component of the digital data acquisition system was made in the same LabVIEW virtual instrument (VI) as the 4-bit counter so the count can be referenced when demuxing. This was a necessary step to keep the entire system synchronized. A formula node consisting of if statements is used to determine the weight of each bit in relation to the counter's binary sequence. The weight of each enabled bit from count 0-7 and 8-15 is then summed to calculate the 8-bit digital values from the two analog-to-digital converters. Each digital value is then multiplied by the resolution of the 8-bit ADCs to compute the voltage reading of the original analog signals. At this point, the raw signal voltage is acquired in LabVIEW and the necessary conversions can be made to display the appropriate units for the sensors.

VI. TROUBLESHOOTING

Several challenges were encountered when building the circuit and the VI. It was necessary to individually test each component of the system to determine how everything would be implemented. The first component that was tested was the multiplexer. To test the multiplexer, the digital inputs were pulled high or low directly to the power source. The output was then

connected to an oscilloscope to view the voltage as different selector bits were manually pulled high or low. Everything worked as expected except for one observation; if a digital input or selector bit was neither pulled high or low, then the default reading for that pin would be high.

The next component that was tested was the VI for the counter and demuxer. Static values were assigned to the digital inputs of the multiplexer circuit and the selector bits were connected to the counter. The probe feature in LabVIEW was then used to compare the values in software with the associated values directly on the multiplexing circuit. When testing this component, I found that the synchronization between hardware and software was off by exactly one iteration in the VI when using LabVIEW 2017. However, I later discovered that there were no synchronization issues when using LabView 2012. As a result, attempts to correct the iteration offset were discarded.

The last component that was tested was the analog to digital converter. To determine if the ADCs were operating correctly, LEDs were connected to each digital output and a potentiometer was connected to the analog input. In theory, this would allow the ADC's binary output to be observed when applying 0-5 volts. Early attempts failed to make the circuit operational, so further research had to be conducted. During my research, I found that it was necessary to construct a voltage divider circuit to apply exactly ½ the source voltage to pin nine of the ADC. After probing the digital outputs with a multimeter while adjusting the analog input voltage, I discovered that the outputs were the exact opposite of what I had originally anticipated. This was because the digital outputs were based on active low, or inverted logic. According to De Morgan's Theorem, the output of a negated input will result in the complement of its input. To account for this theorem, the output connections of the multiplexer ICs were switched to the inverted output pin. Applying these adjustments resulted in active high logic on the final output of the circuit by negating the active low outputs of the ADCs.

VII. RESULTS

After the functionality of each component was verified, the full system design was implemented and tested. Upon testing, a few minor bugs were discovered in the demuxing code and the appropriate corrections were applied. The system worked exactly as planned and the results reflected the theory with impressive accuracy. The difference when comparing the computed analog signal voltage in LabVIEW to the actual analog signal voltage with a multimeter only varied by 0.3 millivolts. Another notable observation was that the 5V power supply on the myDAQ was not sufficient to power the circuit. A voltage of 4.6V was observed when measuring across the 5V power supply on the myDAQ when the load was applied. The variations were greater when comparing the differences between actual signal voltage and computed signal voltage while the 5V myDAQ source powered the circuit. This is likely because the lower voltage altered the resolution of the ADCs.

VIII. FINAL WORDS

Overall, I found this project to be a very good learning experience. When I first began my project, I knew very little about multiplexing circuits. However, I was able to apply my knowledge in digital logic and circuitry to analyze and interpret the datasheets and diagrams of multiplexer ICs to improve my understanding in a time efficient manner. I was then able to use this knowledge to implement theory into design with much success.

Additionally, I gained experience in overcoming unforeseen challenges, which pushed me to think outside of the box. I am happy with my results and found this to be a good project.

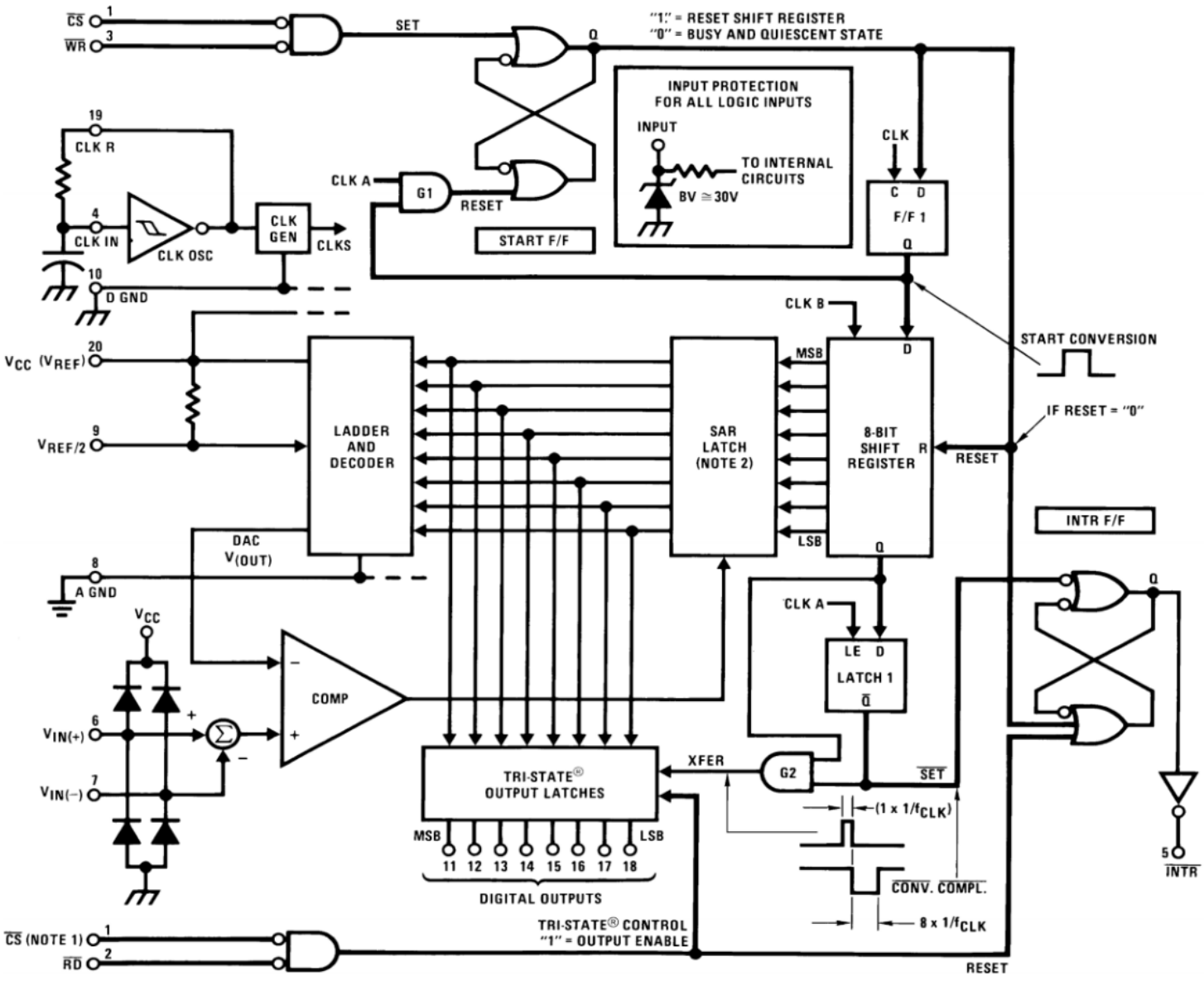
REFERENCES

[1] Ume.gatech.edu. (2017). *Cite a Website - Cite This For Me*. [online] Available at: http://ume.gatech.edu/mechatronics_course/ADC_F08.pdf [Accessed 25 Nov. 2017].

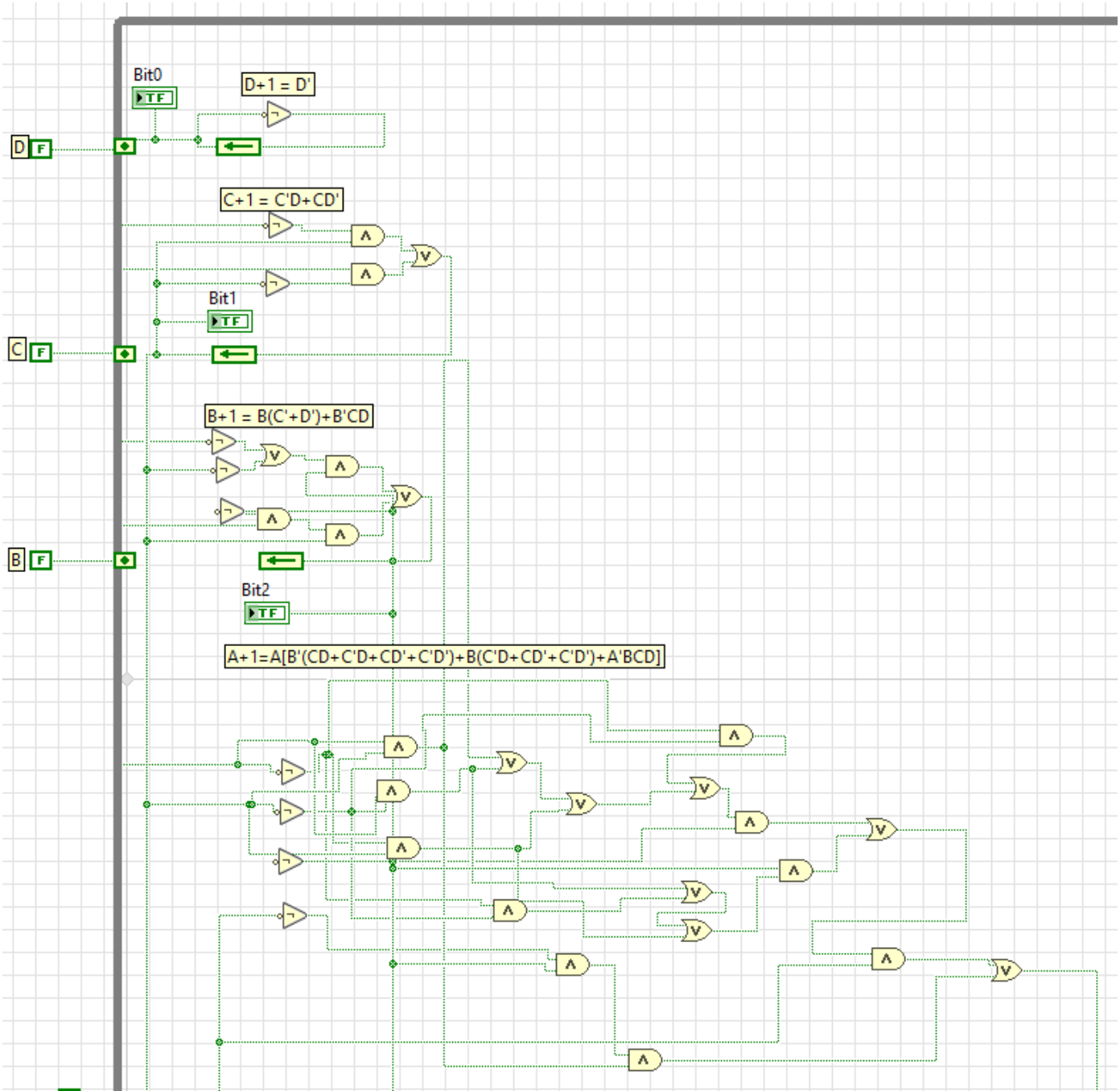
[2] Ti.com. (2017). *Cite a Website - Cite This For Me*. [online] Available at: http://www.ti.com/lit/ds/symlink/adc0802-n.pdf [Accessed 21 Nov. 2017].

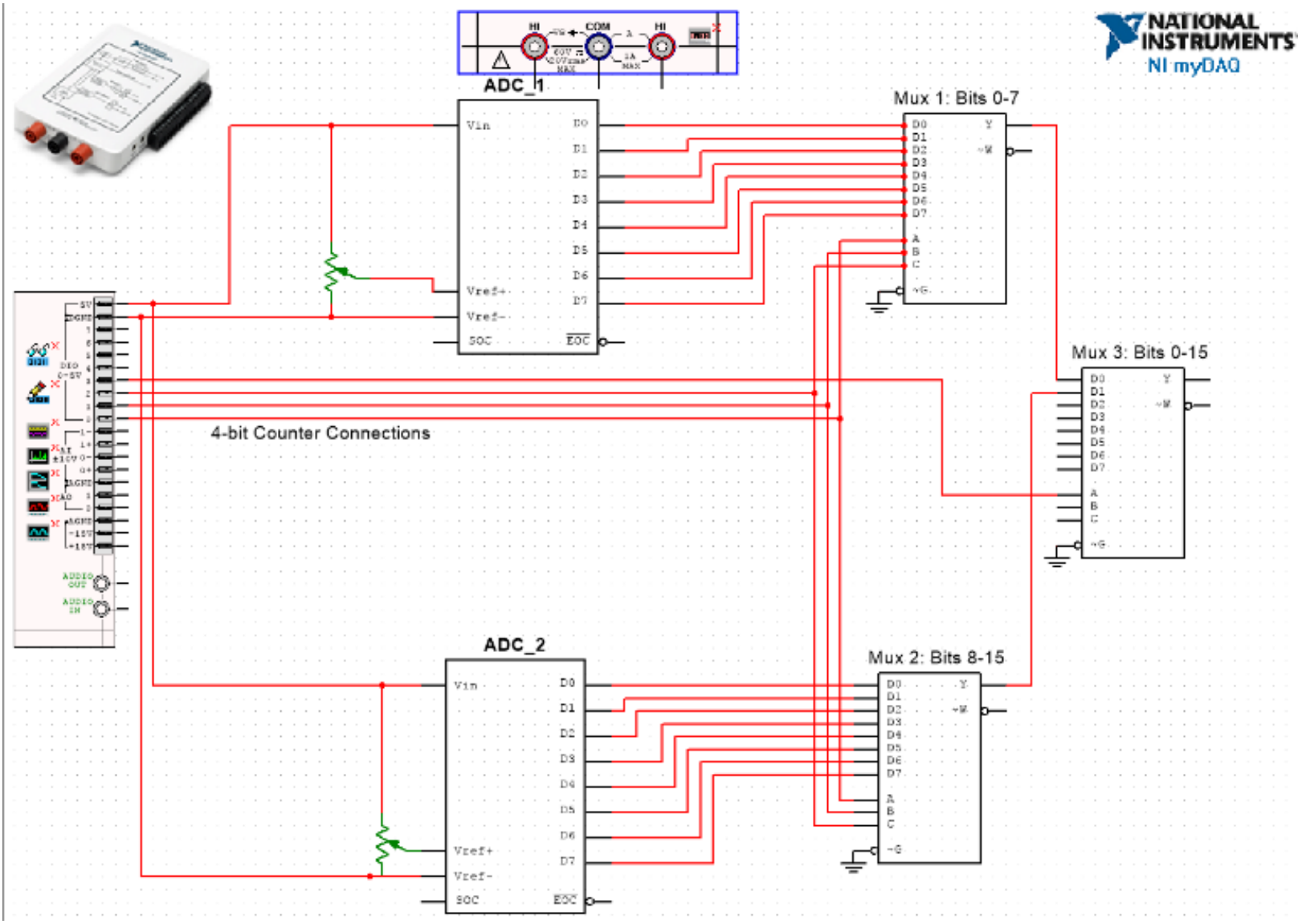
[3] Alldatasheet.com. (2017). *SN74151A Datasheet(PDF) - Texas Instruments*. [online] Available at: http://www.alldatasheet.com/datasheet-pdf/pdf/204648/TI/SN74151A.html [Accessed 24 Nov. 2017].

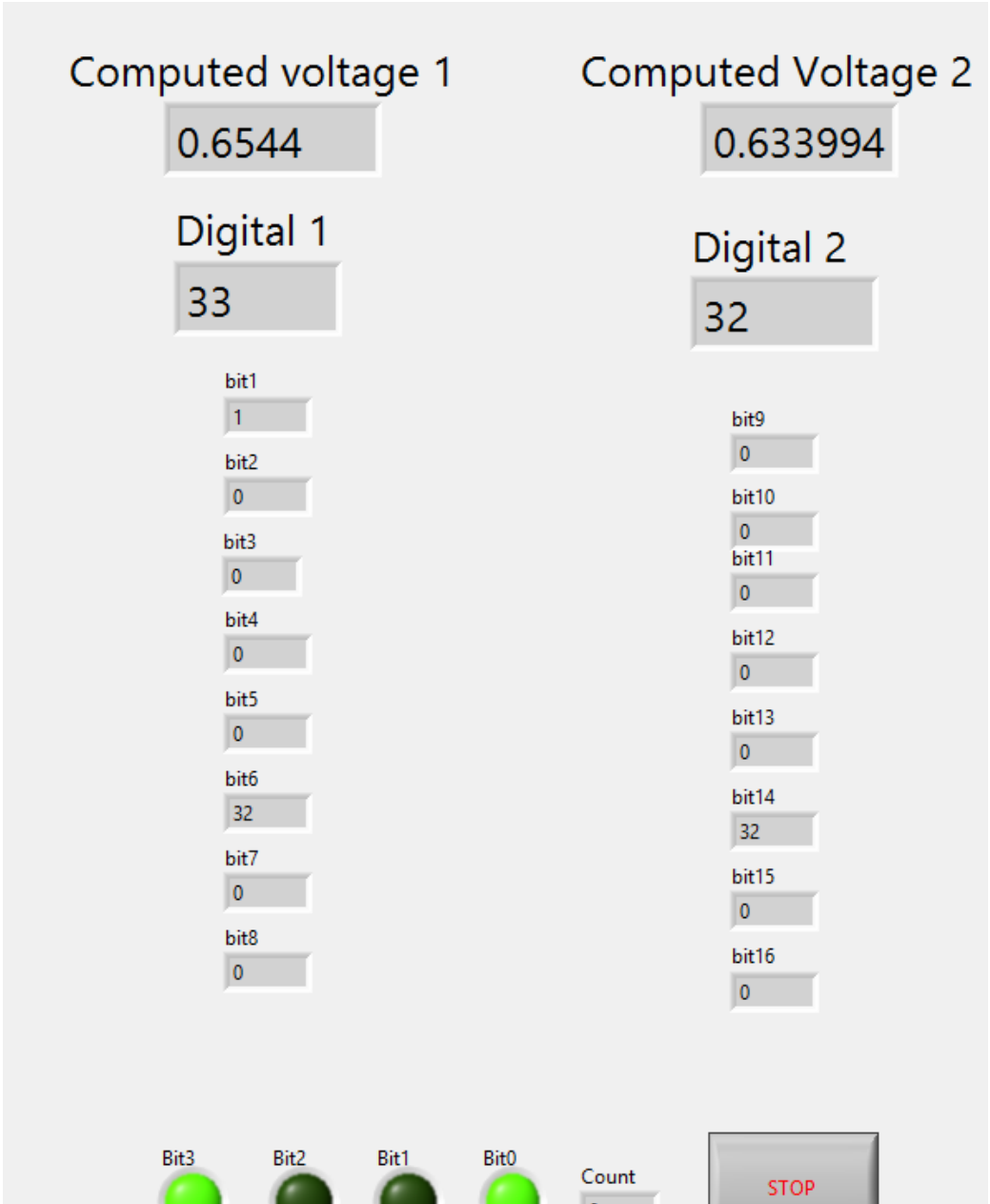
Functional Block Diagram

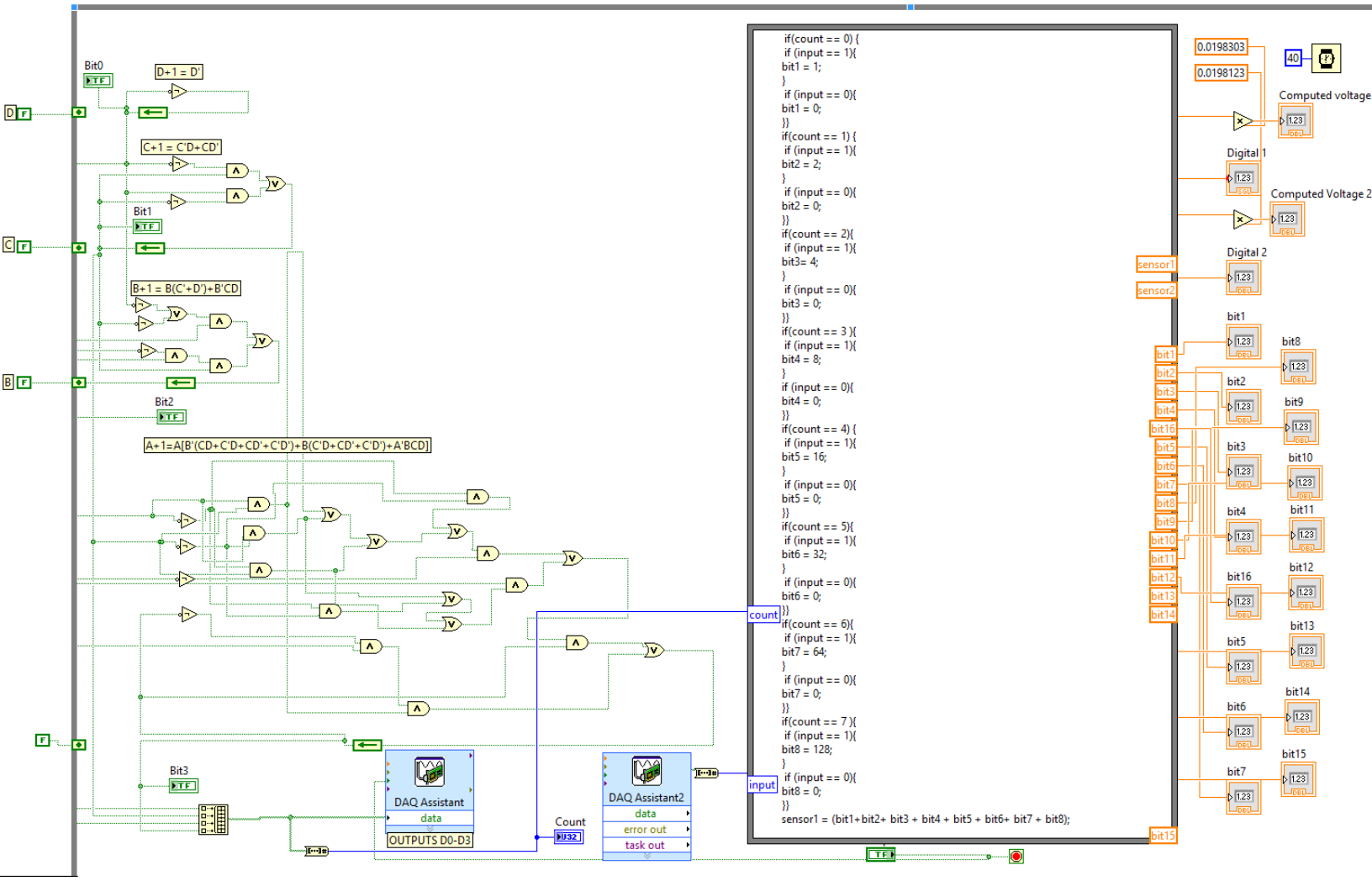


(1) $\overline{CS}$ shown twice for clarity.


Bit0
D+1 = D'
D
C+1 = C'D+CD'
Bit1
C
B+1 = B(C'+D')+B'CD
B
Bit2
A+1=A[B'(CD+C'D+CD'+C'D')+B(C'D+CD'+C'D')+A'BCD]


NATIONAL INSTRUMENTS
NI myDAQ
ADC_1
Vin
Vref+
Vref-
SOC
EOC
D0
D1
D2
D3
D4
D5
D6
D7
Mux 1: Bits 0-7
Mux 3: Bits 0-15
4-bit Counter Connections
ADC_2
Mux 2: Bits 8-15
A
B
C
Y
~W
~G


Computed voltage 1
0.6544
Computed Voltage 2
0.633994
Digital 1
33
Digital 2
32
bit1
1
bit2
0
bit3
0
bit4
0
bit5
0
bit6
32
bit7
0
bit8
0
bit9
0
bit10
0
bit11
0
bit12
0
bit13
0
bit14
32
bit15
0
bit16
0
Bit3
Bit2
Bit1
Bit0
Count
STOP


Bit0
D+1 = D'
D
C+1 = C'D+CD'
Bit1
C
B+1 = B(C'+D')+B'CD
B
Bit2
A+1=A[B'(CD+C'D+CD'+C'D')+B(C'D+CD'+C'D')+A'BCD]
F
Bit3
DAQ Assistant
data
OUTPUTS D0-D3
Count
I32
DAQ Assistant2
data
error out
task out
count
input
if(count == 0) {
if (input == 1){
bit1 = 1;
}
if (input == 0){
bit1 = 0;
}}
if(count == 1) {
if (input == 1){
bit2 = 2;
}
if (input == 0){
bit2 = 0;
}}
if(count == 2){
if (input == 1){
bit3= 4;
}
if (input == 0){
bit3 = 0;
}}
if(count == 3){
if (input == 1){
bit4 = 8;
}
if (input == 0){
bit4 = 0;
}}
if(count == 4) {
if (input == 1){
bit5 = 16;
}
if (input == 0){
bit5 = 0;
}}
if(count == 5){
if (input == 1){
bit6 = 32;
}
if (input == 0){
bit6 = 0;
}}
if(count == 6){
if (input == 1){
bit7 = 64;
}
if (input == 0){
bit7 = 0;
}}
if(count == 7){
if (input == 1){
bit8 = 128;
}
if (input == 0){
bit8 = 0;
}}
sensor1 = (bit1+bit2+ bit3 + bit4 + bit5 + bit6+ bit7 + bit8);
0.0198303
0.0198123
40
Computed voltage
Digital 1
Computed Voltage 2
Digital 2
sensor1
sensor2
bit1
bit2
bit3
bit4
bit16
bit5
bit6
bit7
bit8
bit9
bit10
bit11
bit12
bit13
bit14
bit15
bit8
bit9
bit10
bit11
bit12
bit13
bit14
bit15