Eight-bit Binary Adders and Analog to Digital Conversion

◊

Prepared for:

Professor Pattengale

◊

Engineering 17L - Circuit Theory Lab Section

Tuesday 2:00 - 5:05

◊

Vince Anicich, Fabrizio Farina, Zack Garza

◊

Effective Date of Report:

May 20, 2014

Table of Contents

Introduction / Abstract

Until the 1960's, measurement and control of physical quantities was mostly based on analog systems. In analog circuits the magnitude of the physical quantity is expressed proportionally to a voltage or current applied; since voltage and current can take any values, the measurement's values are also continuous. For this reason analog systems are also known as *continuous-state* systems. The advent of transistors, logic gates, and integrated circuits changed radically the way of measuring and controlling physical quantities. Since most of the silicon based devices operates within a very limited voltage range -which is translated into binary numbers, the measurement value is expressed in discrete steps. Therefore, digital systems are also known as *discrete-state* systems.

In this lab the students will explore the operation of logic gates, the theory behind binary addition and subtraction, and the basic working principle of Analog to Digital Converters. With the use of a software simulator (Circuit Maker) the student will draw and simulate the circuitry of an 8-bit adder/subtractor. The students will then build and test the final working model. Moreover, through the use of an analog devise (a potentiometer) the student will simulate the measurement of a physical quantity (i.e. temperature) as a way of testing the accuracy of the Digital to Analog Converter.

Theory

Adders

Adders find a variety of uses, forming the fundamental building blocks of arithmetic logic units (ALUs) in computers of all sizes. In this laboratory, our goal is to find the sum of two digits using a combination of discrete circuit elements that perform logical operations. To address this task, we turn to circuit combinations that are referred to as adders – in particular, half-adders and full-adders – to develop the theory that motivated the design of this circuit.

Half Adders

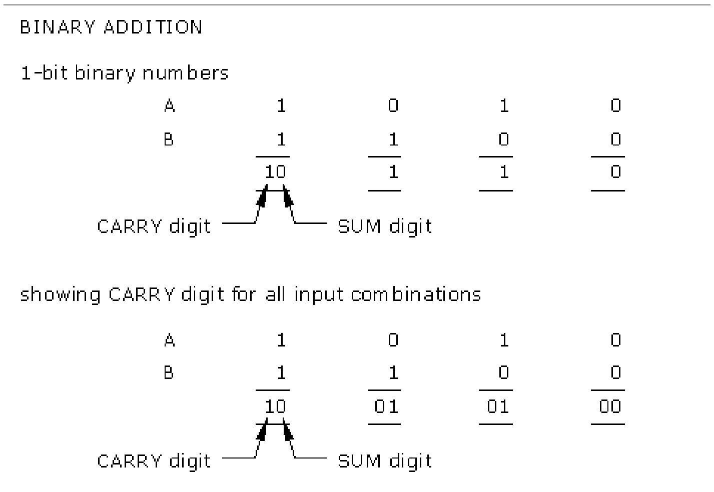


Figure 1: Binary Addition

It is important to distinguish between a full adder and a half adder, particularly when it comes to summing two digits with a "carry". In traditional decimal arithmetic, the concept of a carry is addressed by indexing digits – for example, starting at zero and incrementing by 1 digit at a time, the largest number that can be represented is 9. To address this, we use a second digit to represent the "ten's place", and reset the digit in the "one's place" to its initial value, giving us 10. Similarly, reaching 19, the ten's place is incremented and the one's place is reset, yielding 20. In base 10, there are only finitely many numbers that can summed that do not overflow the one's place position – while it is possible to represent 4+5 with only one digit, sums such as 8+3 require another digit to communicate what the sum really is. We refer to extra digit as the "carry", which can be generalized to number systems with arbitrary bases. For the purposes of this laboratory, we will only examine the binary case.

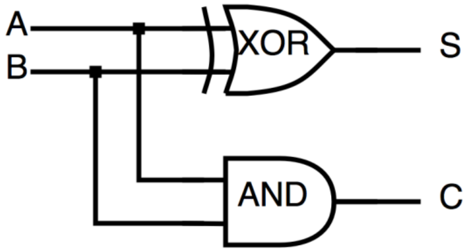


Figure 2: Logical Schematic for a Half Adder

Figure 2 shows a simple implementation of a half adder that takes two inputs, **A** and **B**, and produces a sum **S** and a carry **C**. This is ideal for adding two single bytes together, but presents a problem when adding multi-digit numbers together. For example, as in traditional decimal arithmetic, overflowing the one's position contributes an additional factor of ten to the overall sum. The case is similar in binary arithmetic. However, a half adder circuit has no way to differentiate between which bits contribute more to the overall sum – or, in other words, which bits are more significant than others. While it does allow the addition of single digits, what is needed is a way for the carry to propagate through the addition of higher order (i.e., more significant) digits as they are being added.

Half Adder	**Inputs**	**Outputs**	
A	**B**	**S**	**C_0**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Figure 3: Truth Table for a Half Adder

Full Adders

In order to account for a carry that can effectively "ripple" through a calculation, a logical circuit can be constructed that functions much like the half adder, but takes into account the possibility of a number being carried in from a previous calculation. This is denoted in Figure 6 as C_{in}. The truth tables for a half adder and a full adder are compared in Figures 3 and 4, and it can be seen that for all combinations of **A** and **B**, the inputs are identical. However, the functionality is extended by including the carry in as a third input, allowing multiple adders to be chained together to compute the sum of arbitrarily large digits.

Full Adder	**Inputs**		**Outputs**	
A	**B**	**C_i**	**S**	**C_0**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 4: Truth Table for a Full Adder

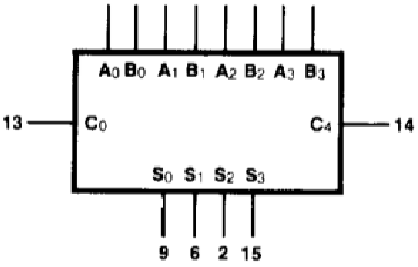


Figure 5: Pin-out Diagram of a 4 bit Full Adder

In this laboratory, we utilize dual inline-pin chips, equipped with four full adders each, to accomplish the task of summing together binary words consisting of 4 bits. Each adder is capable of summing together 4 bit inputs to produce a 4 bit

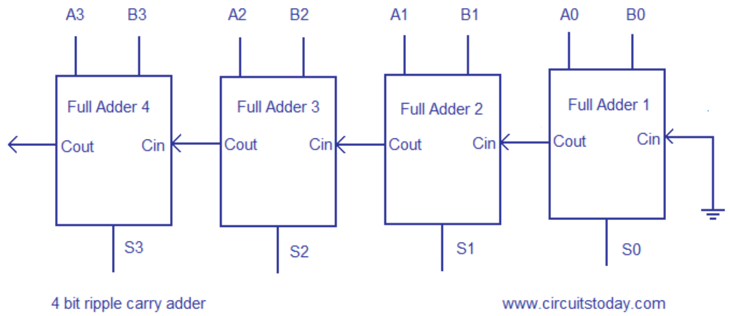


Figure 6: Ripple Carry Chaining of Full Adders

output and a single carry bit. By sending this carry bit into a second sequence of adders, we can extend the circuit to sum together two 8 bit inputs to produce an 8 bit output and an additional carry bit. For a sequence of such adders wired in series, such that the carry out of each adder is sent to the subsequent adder, the sum of the two inputs is generated in order from its least to most significant bit.

Thus, given a binary input A for which the bits are represented as $A_{0,}A_{1,}A_2 ... A_i$, an input B with the bits $B_{0,}B_{1,}B_{2.}...B_i$ a carry in C_0 **,** the digits of the sum S are given by:

That is, adding two corresponding bits in the *i*th position contribute a term of 2^i in the sum.

$$S = C_0 + (A_0 + B_0) 2^0 + (A_1 + B_1) 2^1 + (A_2 + B_2) 2^2 + ... + (A_i + B_i) 2^i + (C_i) 2^{i+1}$$

$$\rightarrow S = C_0 + (C_{i+1}) 2^{i+1} + \sum_0^i (B_i + C_i) 2^i$$

When the sum is overflowed, the carry can be interpreted to represents an error flag, indicating, that the sum is out of range – however, the sum is still valid as a 9 bit number, where the carry denotes most significant bit. Thus, the carry makes an additional contribution of 2^{i+1} . Converting to decimal, an 8 bit input can contain any number between 0 and 255 ($2^8 - 1$), and the sum can represent any number between 0 and 510 ($2^8 - 1 + 2^8 - 1 = 2^9 - 2$), as it is represented by 8 bits plus a 9th carry bit. In addition mode, this represents 2^{16} possible input combinations, and 2^9 possible outputs.

Subtraction

The functionality of the adder can be inverted to produce the difference **A-B**. This is done by routing each inputs of B through one branch *XOR* gate before it reaches the adder, while the other branch of each gate is wired to a mode-select switch. This schematic is shown in Figure 7 where the switch is denoted SUB. Examining the truth table for a *XOR* gate, given these inputs, shows that when the SUB switch is deactivated, the input of B is unchanged. However, when SUB is activated, the gate functions as an inverter acting on B's input.

XOR	**Inputs**	
SUB	**B_i**	**Output**
0	0	0
0	1	1
1	0	1
1	1	0

Figure 7: Truth table of a XOR gate

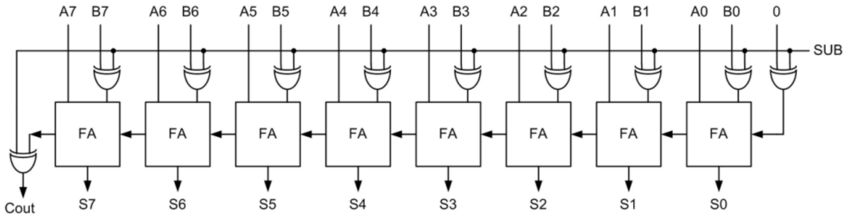


Figure 8: Ripple Carry of Adder-Subtractor

In order to perform subtraction, we take advantage of a property of number's represented in two's complement. In this case, the values **A-B** can be obtained by evaluating **A +~B+1**; that is, by inverting B, adding it to A, and adding 1. The *XOR* gate accomplishes the inversion when SUB is activated, and the adder sums the results. So, in order to obtain the final value, the SUB switch is also sent to the first adder's carry in, effectively adding 1 to the result and producing the correct value. In this case, the carry now represents the “borrowed” digit in subtraction. Given an 8 bit input, the sum can then range from -255 (when A = 0 and B = 255) to +255 (when A = 255 and B = 0), where the carry bit is now interpreted as a negative sign. While this yields the same number of possible inputs and outputs as unsigned addition, the use of one bit as a negative sign limits the absolute value of the range to 255 instead of 510.

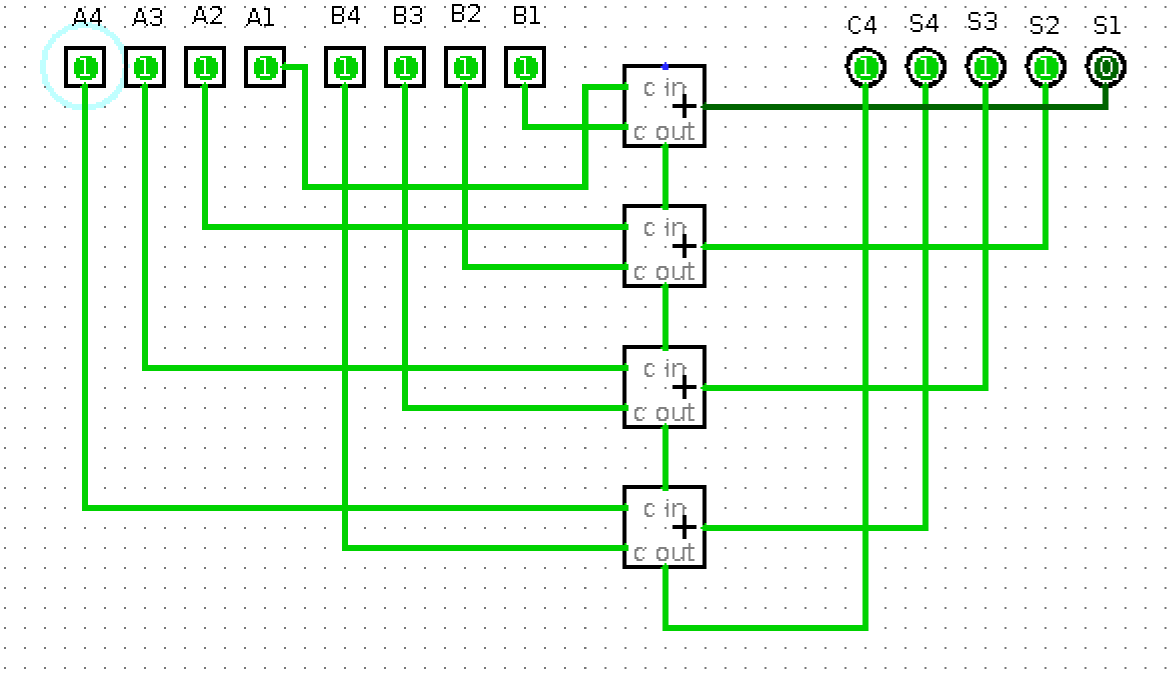


Figure 8b: Logical schematic of the Ripple-Carry Adder's functionality. The carry from the least significant bit is "rippled" through each adder, in this case, resulting in a carry out.

Analog to Digital Conversion

In applications, it is often necessary to collect some form physical data, such as temperature, pressure, light, or sound, and process it as digital data in order to process or perform calculations on it. For this purpose, we turn to ADCs, or Analog to Digital Converters, which take continuous signals and produce discrete output values that are proportional to the input. This provides a crucial link between components that function as transducers and those that are purely digital, such as microprocessors.

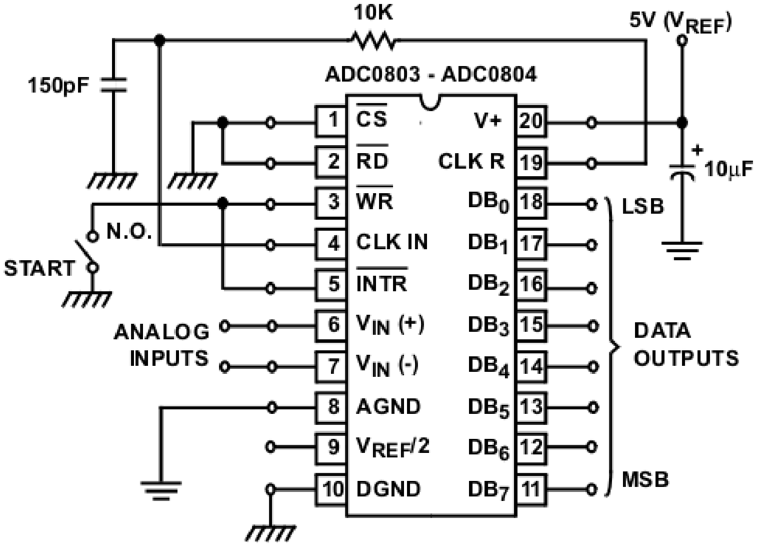
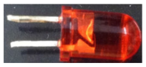


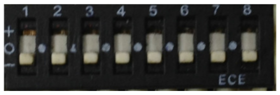
Figure 9: Pin-out of Analog to Digital Converter

If a given ADC has *n* distinct digital outputs, the range of input signals are then partitioned into 2^n distinct values that vary over the entire range. For example, in this circuit, the schematic in Figure 9 is used. The ADC samples a range of 5 volts and produces 8 bits of output. This gives an input resolution of $5/2^8$ volts, or about 19.5 mV. For most ADCs, the range of voltages can be customized or tared to arbitrary "zero" levels, thus providing whatever resolution is necessary for the specific application. However, there are also minimum thresholds to this resolution, as the accuracy of the ADC is dependent on the resistance and capacitance of the input, and several internal components are sensitive to noise. For this reason, many specifications distinguish the analog and digital components and advise that these components are grounded with capacitors, and often suggest using an entirely different bus for the digital ground.

The final key part of the ADC's operation is the clock, which determines the rate at which an input signal is sampled. For most applications, a clock operating at a frequency that is more than twice the input signal's frequency will produce a digital output that can always be reconstructed into the original signal. The clock used determines the frequency at which the input is sampled, or the sampling rate. For this circuit, an RC filter is used, yielding a clock frequency given by $f_{clk} = 1/1.1\mathrm{RC}$.

Equipment List

Nomenclature	Manufacturer	Model/Serial #	Function	Picture
4 Bit Full Adder (x2)	Texas Instruments	SN7483 AN	Produces the sum of 4 binary inputs	
8 Bit A/D Converter	Intersil	ADC0804 LCN	Map analog input voltages to discrete digital outputs	
Assorted Resistors (x4)	N/A	3x 300 Ω, 1x 10 kΩ	Necessary for LED functionality; reduces circuit noises from back-currents.	
Breadboard	Global Specialties	Proto board PB-6	Foundation of circuit	
Capacitor	N/A	1x 114 pF	Used in auxiliary RC filter that serves as a clock for the 8 Bit A/D Converter	
Circuit Maker	MicroCode Engineering, Inc.	Student Ver 6.2	Calculating theoretical values of circuit variables.	
DC Power Supply	Elenco-Precision	XP-581	Source of power to circuits.	
Digital Multimeter	GW Instek Multi Meter	GDM 393A	Measuring voltage and current of circuit elements.	
Light-Emitting Diodes (x25)	N/A	N/A	State indicators for inputs inputs and outputs	

Potentiometer	N/A	5022F	Produce a continuously varying voltage input	
Tri-state Switch Array (x2)	N/A	N/A	Used to furnish discrete high and low states as input	
Wire Jumper Kit	Jameco Electronics	JE10	Circuit components	
Wire Stripper/Cutter	K.Miller Tool Co.	102	Trimming wire and stripping insulation.	
XOR Gate (x2)	Texas Instruments	7486	Modifies Adder inputs to support subtraction mode.	

Procedure

This project was built iteratively over 3 phases. First, a 4-bit adder was constructed. This took two 4 bit inputs, which were denoted as inputs **A** and **B** respectively, and produced a 4 bit output that indicated the sum of the inputs, as well as an additional bit to represent the carried digit.

The circuit was then extended to include a second adder, allowing the addition of 8 bits from each input and producing an 8 bit output and an additional carry bit. Input B was then routed through an array of *XOR* gates, which with the addition of a mode-select switch, allowed the circuit to perform addition (**A+B**) or subtraction (**A-B**).

In the final phase of the circuit's development, the tri-state switch used to control input B was replaced with a potentiometer to produce analog voltage levels. This was then routed through an analog to digital converter in order to send discrete input values to the adders, ultimately producing a sum that could be continuously varied over the input voltage range.

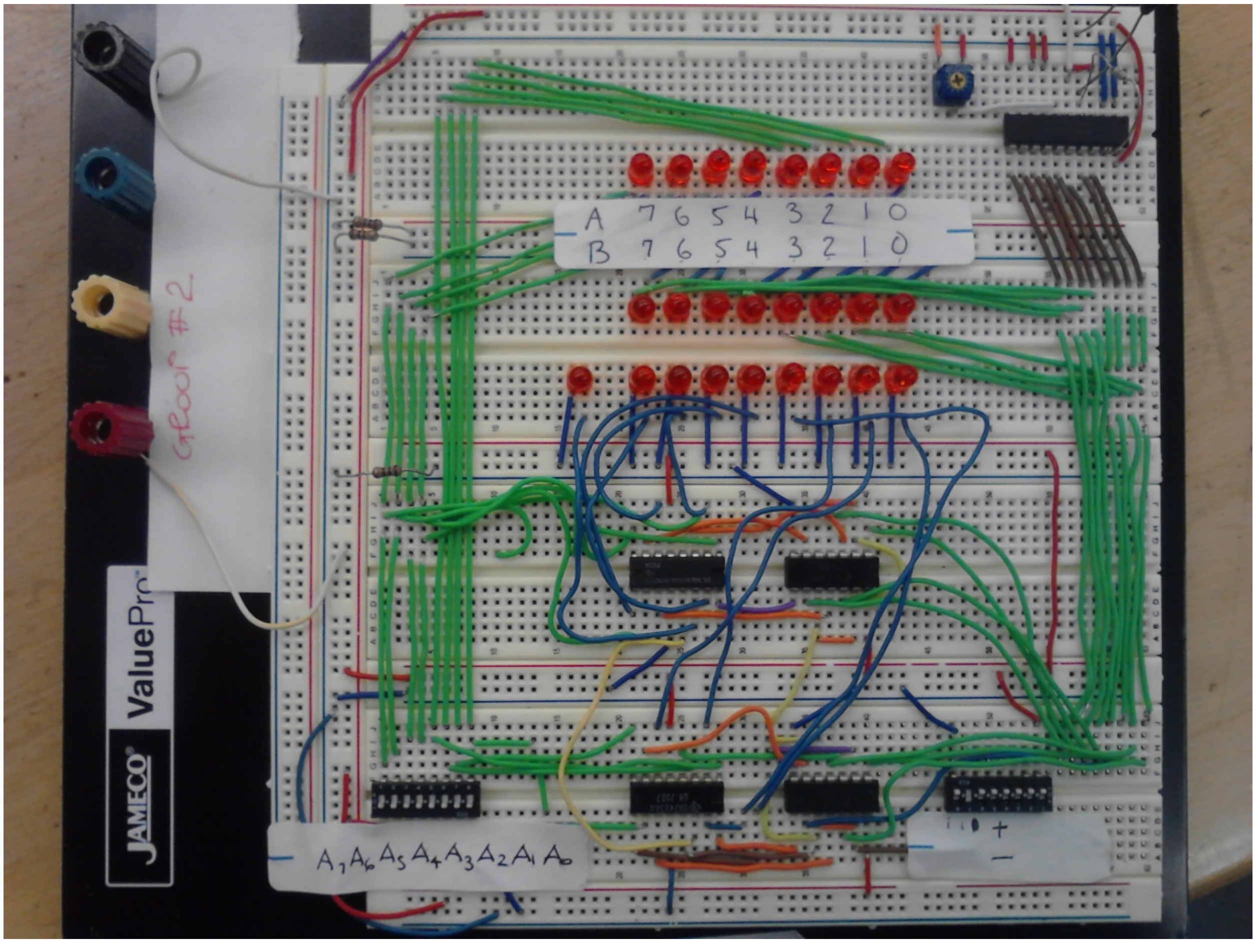


Figure 10: Completed Digital to Analog 8 Bit Adder

Results

4-bit Adder

The 4-bit adder was first tested by selecting all the A's and B's inputs at logic level 1; in this way we verified the correct functioning and layout of all LED's. We then proceeded with several additions by selecting the appropriate addend A and B and by verifying the sum displayed by the LEDs.

8-bit Adder-Subtractor

For the 8-bit adder we followed the same testing procedure developed for testing the 4-bit adder. By selecting the SUB switch we then tested the subtraction operation and obtained positive results.

Analog to Digital Converter

After replacing the tri-state switch with the potentiometer, we first tested the 8-bit adder to verify the functionality of its arithmetic logic. By rotating the potentiometer's knob we selected the B's inputs sequentially ($B_0, B_1, \dots\ B_7$); we then verified the results of the addition operation by looking at the row of LED's ($C_{out}, S_0, S_1, \dots\ S_7$).Several arithmetic operations were made with positive results.

We then simulated a measurement of the outdoor ambient temperature to verify the working principle of the ADC and to test its accuracy. The resolution of the ADC represents how many different voltage steps are discernible as discrete outputs, and is calculated as follows:

$$ADC\ resolution = \frac{5.0\,V}{2^8} = \frac{5.0\,V}{256} = 19.5\,mV/bit$$

For convenience, the temperature range was defined from 0°C (min) to 100°C (max); in this way the binary equivalent of one degree Celsius corresponded to 2.56 bits which, in turn, corresponded to 49.9 mV voltage across the analog input (V+) at pin 6 of the ADC.

Therefore,

$$34°\text{C} \rightarrow 87_{(base\ 10)} \rightarrow 01010111_{(base\ 2)} \rightarrow 1.65\,V$$

We set the potentiometer to obtain the binary value of 87 displayed by the LED's; the measured voltage across the potentiometer was 1.57 V, giving a percent error of **-4.8%.** Figure 11 illustrates number lines corresponding to the range of inputs, along with where the test values lie within these ranges.

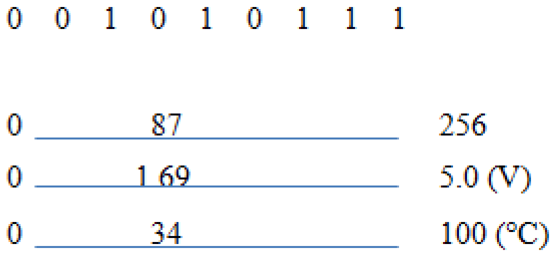


Figure 11: Testing Analog to Digital Conversion

Data was then taken in order to determine the continuous voltage levels that corresponded to each bit of digital output. This was done by measuring the potentiometer's voltage across digital output levels corresponding to each index digit -- that is, a measurement was taken for an output of 2^0, 2^1, 2^2...2^8-1, or equivalently, over the range of 0-255 in decimal. Plotting these data points and interpolating via linear
regression yielded an expression for the
average voltage step required in order to index the output by an additional bit.

The slope of this fit corresponded to approximately 19.40 mV per bit, yielding an error of **0.51%** from the theoretical ADC resolution of 19.5 mV calculated above. While these results are based on the assumption that both the potentiometer and the ADC perform linearly over their respective voltage ranges, outputs produced by arbitrary voltages were in good agreement with this model.

Output (Base 10)	Voltage (mV)
0	3.6
1	29.6
2	48.6
4	79.9
8	156.8
16	316.4
32	611
64	1242
128	2470
255	4960

Table 1: Voltage Levels Over Output Range

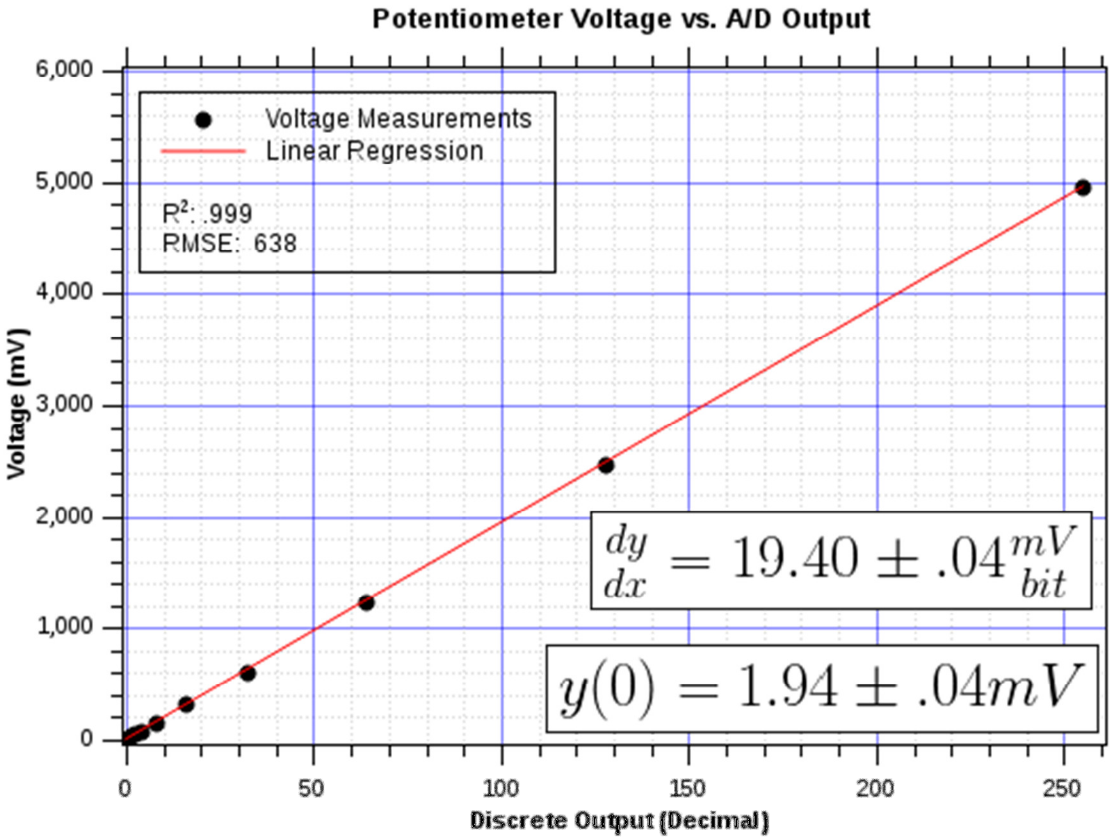


Figure 12: Plot of ordered pairs consisting of input analog voltages over the potentiometer and the corresponding binary output.

Conclusion

The construction of this circuit brought to light many of the unforeseen difficulties that arise in large, multi-phase projects. The first phase was completed with minimal theoretical underpinnings, and functioned as desired - however, as additional specifications were created, more complexity was added to the circuit and the initial design was not ideal for further expansion. This resulted in almost a complete redesign in phase 2, which required a significant amount of planning and theoretical work in Circuit Maker.

We quickly ran into limitations in terms of both space and technology. The small form factor of the bread board required non-ideal wiring in several cases, which became problematic when the circuit malfunctioned as the number of potential failure points increased. Technology was also a limitation in this lab, as combining 16 inputs, 8 outputs, 8 *XOR* gates, 2 adders, and 24 LED state indicators (shown in the figure below) meant quickly running into the 50 device limit imposed by the student version of Circuit Maker.

Additionally, this circuit demonstrated the vital importance of eliminating floating grounds in digital circuits. Many of the errors or apparent malfunctions of the circuit were simply the result of indeterminate floating voltages across the circuit elements. The tri-state switch, for example, was required in order to definitively ground an input as opposed to simply creating a short to deactivate it. Similarly, the adders themselves required their first carry pin to be grounded in order to produce the correct sum. This was also the case with several elements on the ADC, which required specific pins to be jumped and grounded in order to provide the proper signals to its internal circuitry.

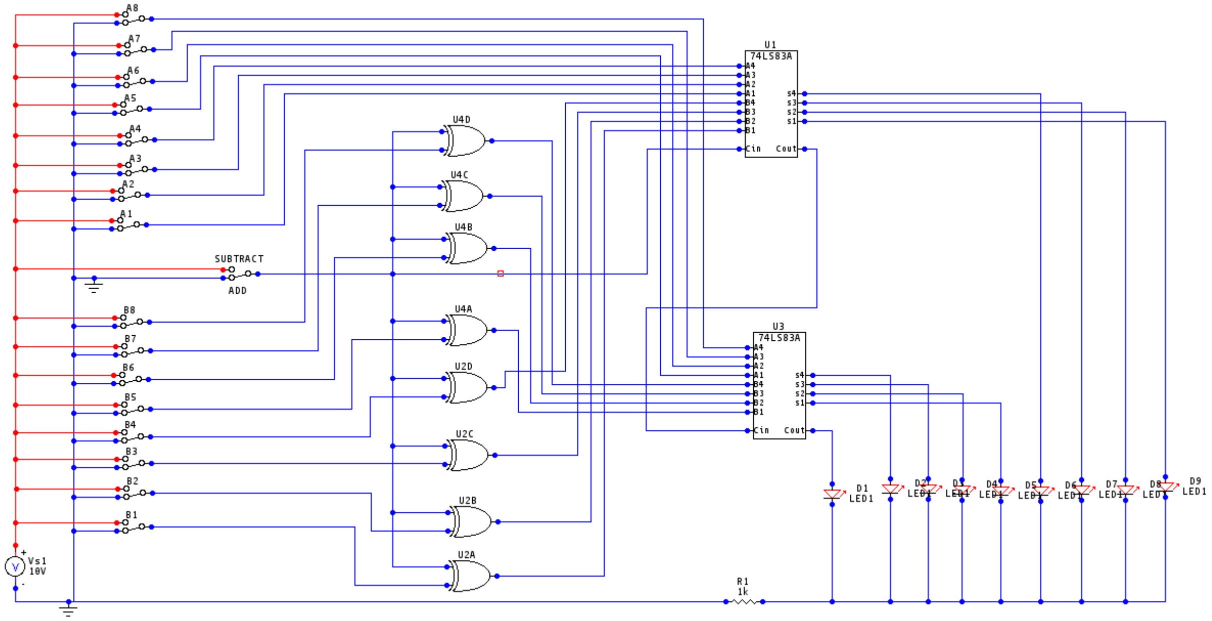


Figure 13: Theoretical schematic made in Circuit Maker

Appendix

Section A: References

Nilsson, James W. and Susan A. Riedel. 2011. Electric Circuits. Ninth Edition. Pearson Education, Inc., Upper Saddle River, NJ.

Serway, Raymond and John Jewett. 2010. Physics for Scientists and Engineers. Eighth Edition. Mary Finch Publishing.

Sarma, Mulukutla S. Introduction to Electrical Engineering. New York: Oxford UP, 2001. Print.

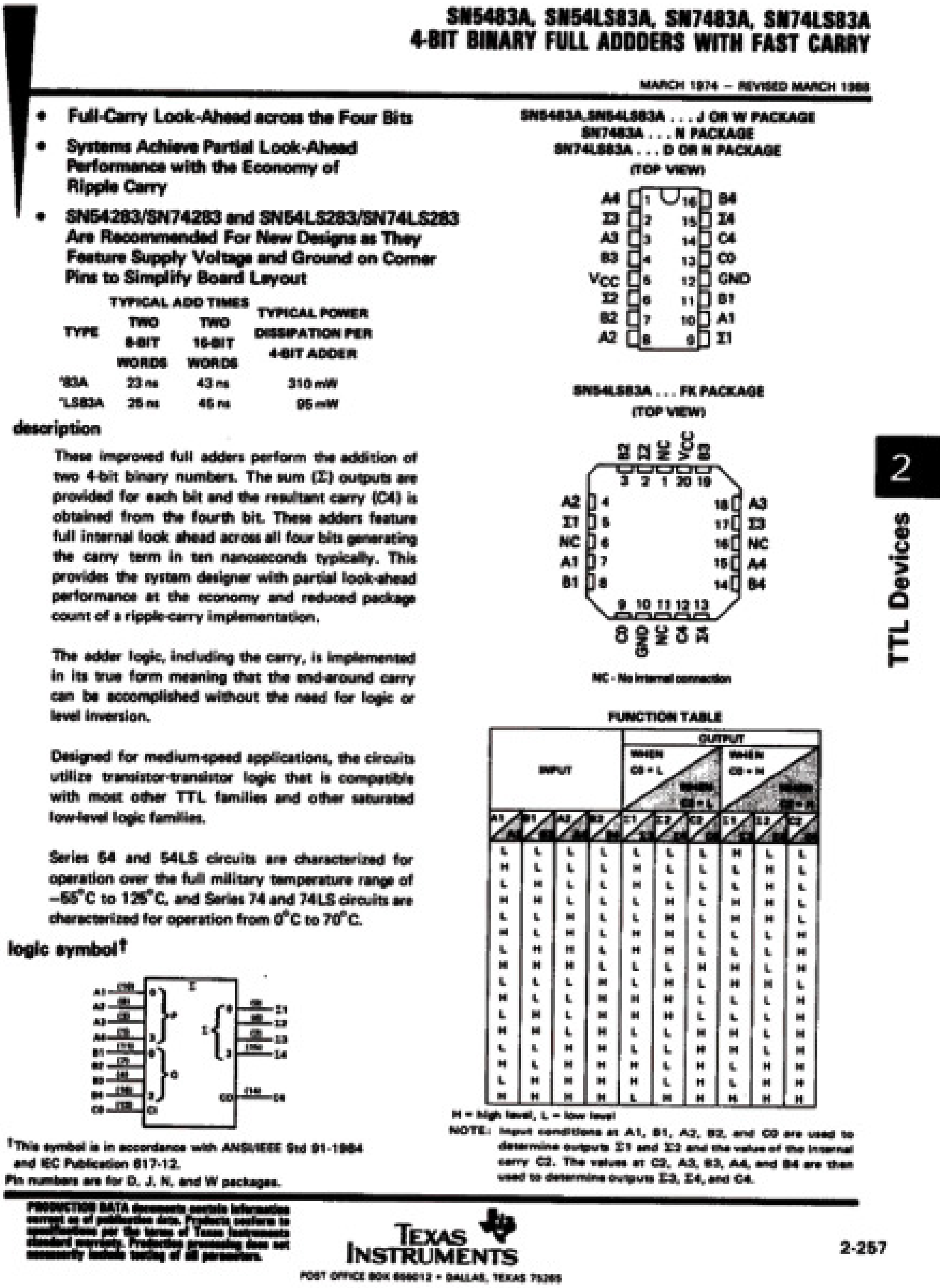
http://www-old.me.gatech.edu

Data Sheet: 7483 Series Adder. http://faculty.spokanefalls.edu/plecoq/7483DS.pdf

Photos: http://www.doctronics.co.uk/

Section B: Data Sheets

4-Bit Full Adder



SN5483A, SN54LS83A, SN7483A, SN74LS83A
4-BIT BINARY FULL ADDDERS WITH FAST CARRY

MARCH 1974 – REVISED MARCH 1988

- Full-Carry Look-Ahead across the Four Bits
- Systems Achieve Partial Look-Ahead Performance with the Economy of Ripple Carry
- SN54283/SN74283 and SN54LS283/SN74LS283 Are Recommended For New Designs as They Feature Supply Voltage and Ground on Corner Pins to Simplify Board Layout

TYPE	TYPICAL ADD TIMES TWO 8-BIT WORDS	TYPICAL ADD TIMES TWO 16-BIT WORDS	TYPICAL POWER DISSIPATION PER 4-BIT ADDER
'83A	23 ns	43 ns	310 mW
'LS83A	25 ns	45 ns	95 mW

SN5483A, SN54LS83A . . . J OR W PACKAGE
SN7483A . . . N PACKAGE
SN74LS83A . . . D OR N PACKAGE
(TOP VIEW)

SN54LS83A . . . FK PACKAGE
(TOP VIEW)

NC - No internal connection

description

These improved full adders perform the addition of two 4-bit binary numbers. The sum (Σ) outputs are provided for each bit and the resultant carry (C4) is obtained from the fourth bit. These adders feature full internal look ahead across all four bits generating the carry term in ten nanoseconds typically. This provides the system designer with partial look-ahead performance at the economy and reduced package count of a ripple-carry implementation.

The adder logic, including the carry, is implemented in its true form meaning that the end-around carry can be accomplished without the need for logic or level inversion.

Designed for medium-speed applications, the circuits utilize transistor-transistor logic that is compatible with most other TTL families and other saturated low-level logic families.

Series 54 and 54LS circuits are characterized for operation over the full military temperature range of −55°C to 125°C, and Series 74 and 74LS circuits are characterized for operation from 0°C to 70°C.

logic symbol†

†This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers are for D, J, N, and W packages.

FUNCTION TABLE

INPUT				OUTPUT WHEN C0 = L / WHEN C2 = L			OUTPUT WHEN C0 = H / WHEN C2 = H		
A1 / A3	B1 / B3	A2 / A4	B2 / B4	Σ1 / Σ3	Σ2 / Σ4	C2 / C4	Σ1 / Σ3	Σ2 / Σ4	C2 / C4
L	L	L	L	L	L	L	H	L	L
H	L	L	L	H	L	L	L	H	L
L	H	L	L	H	L	L	L	H	L
H	H	L	L	L	H	L	H	H	L
L	L	H	L	L	H	L	H	H	L
H	L	H	L	H	H	L	L	L	H
L	H	H	L	H	H	L	L	L	H
H	H	H	L	L	L	H	H	L	H
L	L	L	H	L	H	L	H	H	L
H	L	L	H	H	H	L	L	L	H
L	H	L	H	H	H	L	L	L	H
H	H	L	H	L	L	H	H	L	H
L	L	H	H	L	L	H	H	L	H
H	L	H	H	H	L	H	L	H	H
L	H	H	H	H	L	H	L	H	H
H	H	H	H	L	H	H	H	H	H

H = high level, L = low level

NOTE: Input conditions at A1, B1, A2, B2, and C0 are used to determine outputs Σ1 and Σ2 and the value of the internal carry C2. The values at C2, A3, B3, A4, and B4 are than used to determine outputs Σ3, Σ4, and C4.

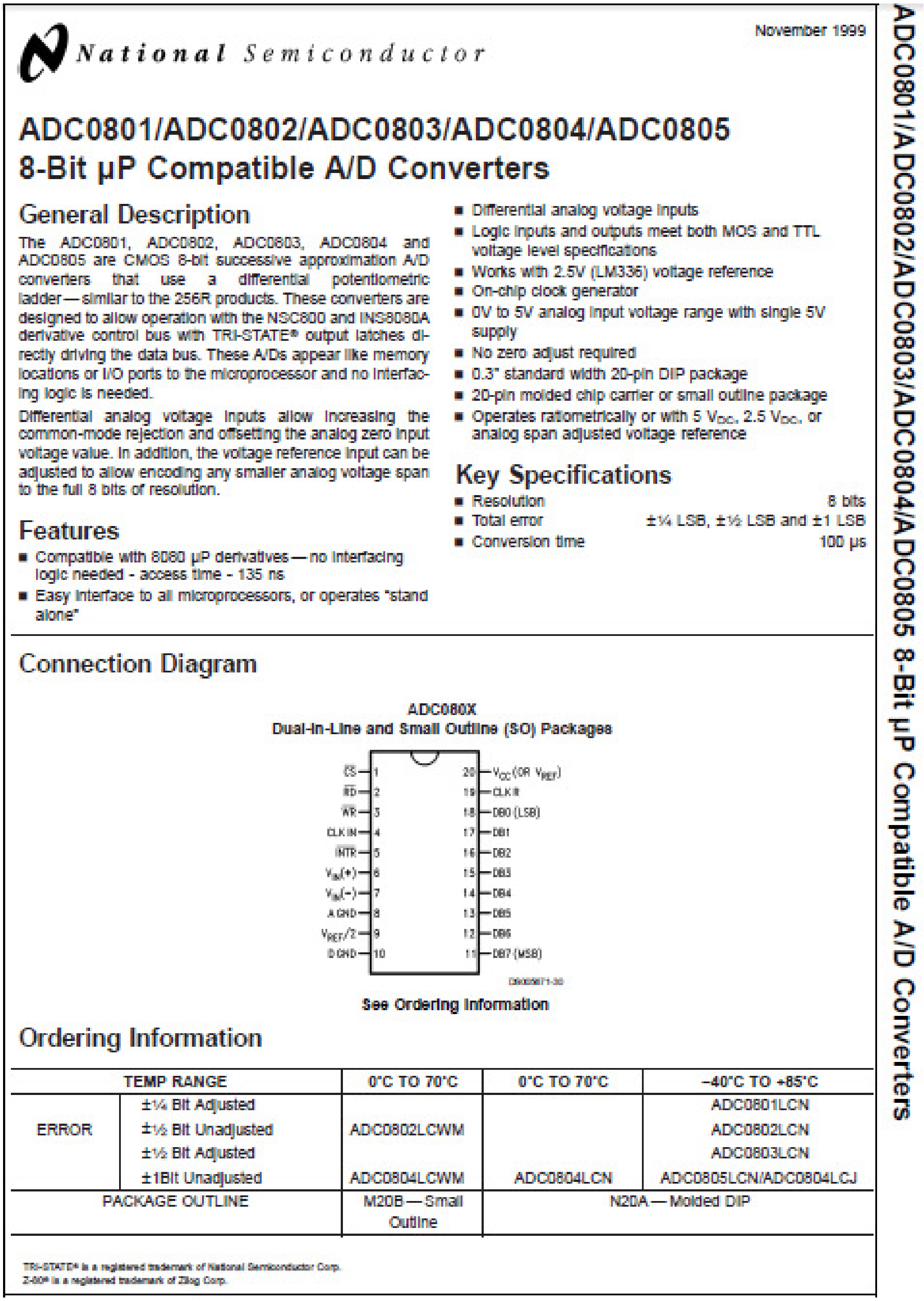
PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS INSTRUMENTS
POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

2-257

2 TTL Devices

ADC Converter



National Semiconductor

November 1999

ADC0801/ADC0802/ADC0803/ADC0804/ADC0805 8-Bit µP Compatible A/D Converters

General Description

The ADC0801, ADC0802, ADC0803, ADC0804 and ADC0805 are CMOS 8-bit successive approximation A/D converters that use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the NSC800 and INS8080A derivative control bus with TRI-STATE® output latches directly driving the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

Features

- Compatible with 8080 µP derivatives—no interfacing logic needed - access time - 135 ns
- Easy interface to all microprocessors, or operates "stand alone"
- Differential analog voltage inputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package
- 20-pin molded chip carrier or small outline package
- Operates ratiometrically or with 5 V_{DC}, 2.5 V_{DC}, or analog span adjusted voltage reference

Key Specifications

■ Resolution	8 bits
■ Total error	±¼ LSB, ±½ LSB and ±1 LSB
■ Conversion time	100 µs

Connection Diagram

ADC080X
Dual-In-Line and Small Outline (SO) Packages

See Ordering Information

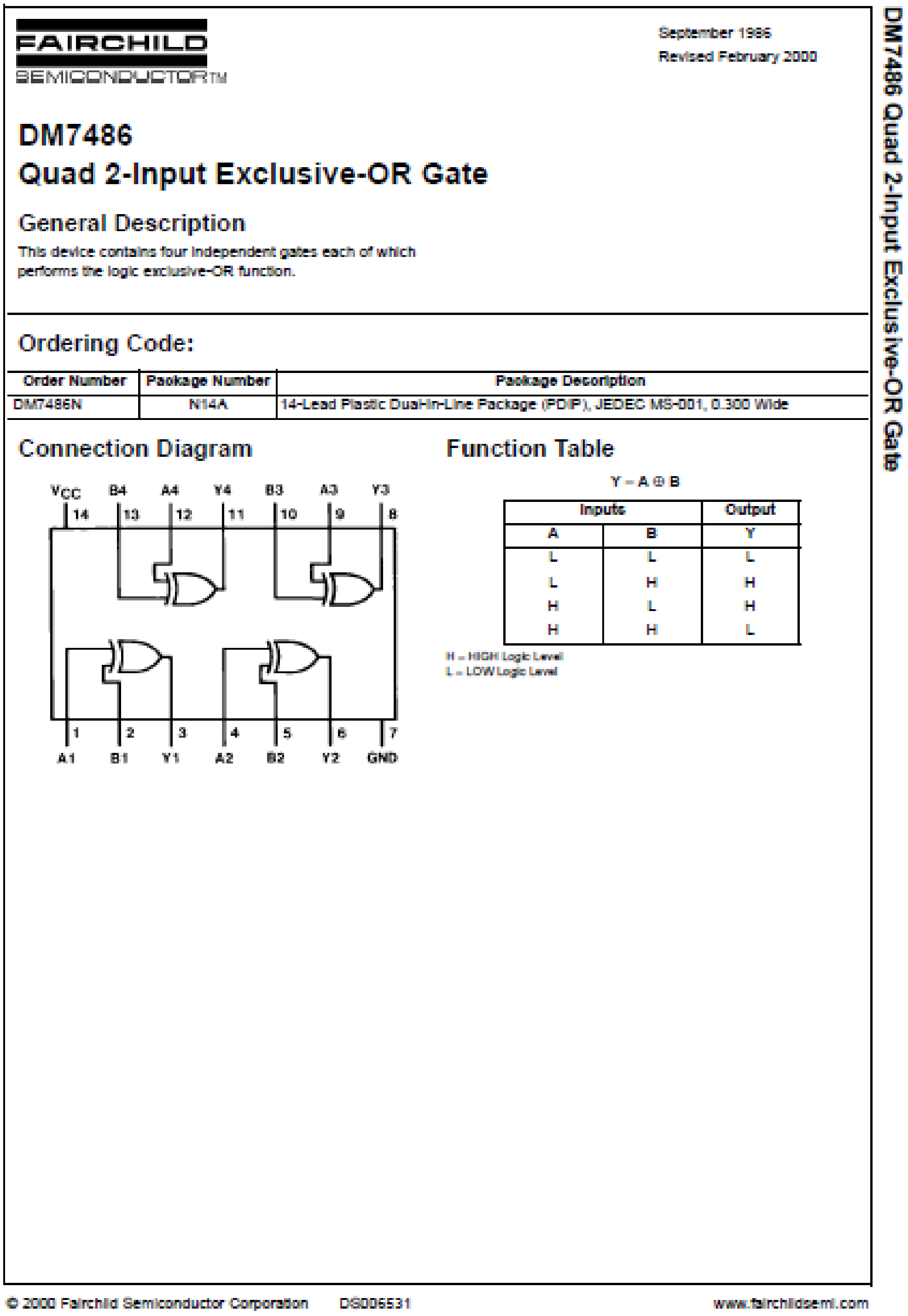
Ordering Information

TEMP RANGE		0°C TO 70°C	0°C TO 70°C	−40°C TO +85°C
ERROR	±¼ Bit Adjusted			ADC0801LCN
	±½ Bit Unadjusted	ADC0802LCWM		ADC0802LCN
	±½ Bit Adjusted			ADC0803LCN
	±1Bit Unadjusted	ADC0804LCWM	ADC0804LCN	ADC0805LCN/ADC0804LCJ
PACKAGE OUTLINE		M20B—Small Outline	N20A—Molded DIP	

TRI-STATE® is a registered trademark of National Semiconductor Corp.
Z-80® is a registered trademark of Zilog Corp.

ADC0801/ADC0802/ADC0803/ADC0804/ADC0805 8-Bit µP Compatible A/D Converters

XOR Gate



FAIRCHILD
SEMICONDUCTOR™

September 1986
Revised February 2000

DM7486 Quad 2-Input Exclusive-OR Gate

DM7486
Quad 2-Input Exclusive-OR Gate

General Description

This device contains four independent gates each of which performs the logic exclusive-OR function.

Ordering Code:

Order Number	Package Number	Package Description
DM7486N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Connection Diagram

Function Table

$Y = A \oplus B$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

DS006531 www.fairchildsemi.com