SLAC-PUB-759
June 1970
(ACC)

HIGH SPEED ANALOG DATA PORT FOR A PDP-9*

W. C. Struven

Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

Introduction

A high speed analog data port has been designed and built for the SLAC accelerator control computer — a PDP-9. The computer has a DM09A adapter - multiplexer as an interface for the DMA channel. The DM09A is a three-port adapter with the highest priority port hard-wired to a Burroughs million word disk. The second priority channel has been wired to the high speed analog input to be described. A number of pulsed analogs are available from the injector and each of the 30 sectors of the accelerator. Three sector analogs are currently viewed in Central Control (CCR). These are: log Q (which is a logarithm of transmitted beam current), an "x" or horizontal beam position signal and a "y" or vertical position signal.[1] An existing multiplexer in CCR samples each log Q, x, and y analog at a 360 pps repetition rate. This multiplexer samples up to 108 analog signals at a 125 KC rate. Sequential sampling of the first 36 channels commences 150 μsec after each beam pulse and lasts for 320 μsec. The second 36 channels are scanned for 320 μsec, starting 1000 μsec after the beam pulse. The third 36 channels are scanned for 320 μsec, starting 1860 μsec after the beam pulse. The present display system presents these analog signals as three rows of dots on the control room oscilloscopes. The accelerator can deliver up to six

*Work supported by the U. S. Atomic Energy Commission.

(To be submitted for publication)

sequential beams at a 360 pps rate, therefore, software associated with this interface has been written to selectively sample and load the PDP-9 memory with up to 108 analogs from any of these six beams.

Interface Description

An interface device was designed to digitize each analog signal. This eight bit "word" is strobed directly into the PDP-9 memory every 8 μsec, after transfer has been initiated. The block diagram of the interface is shown in Fig. 1. A memory address counter (MAC - a 15 bit register) has been provided which is initialized via the I/O bus. The address of the first data word is loaded into the register prior to the beginning of transfer. This address is incremented by the DM09A after each transfer. A negative number corresponding to the number of words to be transferred is preloaded into the word counter (W.C. - a 15 bit register). When the W.C. register has been initialized, the transfer operation is enabled. The W.C. register is also incremented by the DM09A after each transfer. When the word count reaches zero, a word count overflow Flip Flop (WCOF) is set thus setting the "Done" flag which in turn produces a program interrupt. The IORS command is then issued to determine which device has interrupted. This part of the interface is very similar to the Burroughs Disk Controller (RC09). A "Device" flag has been incorporated to request a data word transfer after each analog signal has been digitized and loaded into the data buffer.

Analog Signal Digitizer (Fig. 2)

This portion of the interface is designed around a fast eight bit analog digital converter (ADC). The ADC used is a Bunker Ramo model 850, which will digitize a 0 to +5 volt signal in 1 μsec. The maximum time available to sample and store this signal is 5 μsec. During this period, the analog signal must be

reduced in gain by one half, offset by 2-1/2 volts, digitized, level shifted, converted from gray code to binary code and finally loaded into the data buffer. The most unusual circuits in this part of the interface are the fast pulse amplifier and the gray to binary converter.

Fast Pulse Amplifier

The high speed linear amplifier utilizes a National Semiconductor LM101A operational amplifier, which settles in less than 500 nsec. The circuit uses a feed forward compensation capacitor (150 pf) connected between the inverting input and the compensation terminal.[2] This type of compensation increases the unity gain bandwidth from 1 mc to 8.5 mc. The slew rate or rise time is increased by an addition of a diode from the inverting input to ground. Offset compensation is achieved by operating the inverting input at a slight negative level sufficient to offset the output to exactly +2.5 volts. The overall effect is to reduce the +5 to -5 volts swing to ±2.5 volts with respect to a reference level of 2.5 volts. The output swing then exactly matches the input range required by the ADC.

Gray to Binary Converter

The gray to binary converter utilizes DEC R131 exclusive or (XOR) circuits in a special configuration to reduce the conversion time. The logic function performed by the R131 is actually a coincidence or $\overline{\text{XOR}}$. The usual circuit for conversion of an eight bit code requires 7 XOR's. Using DEC R131's, the settling time is about 2.5 μsec. The circuit which was used in this interface settles in 1 μsec (see Fig. 3). Figure 4 is included to illustrate this form of converter implemented using XOR gates. The R131 outputs are derived in such a manner so that a maximum of three levels of gating exist in the worst case, that is, a change in the MSD (most significant digit) must pass through a maximum of three

gating levels to effect the least significant bit (LSB). In fact, a change in any input bit will pass through a maximum of three gates to affect any output bit. This time saving resulted in a workable system, i.e., a system that could digitize up to 108 signals in one accelerator interpulse period of 2.7 msec.

The logic for this gray to binary converter was suggested by Dr. K. B. Mallory, Group Leader, Instrumentation and Control.

Conclusion

This interface is able to digitize up to 108 analog signals and load the PDP-9 memory with negligible effect on the operational program. The signal accuracy is better than 1%.

References

1. R. B. Neal, Editor, The Stanford Two-Mile Accelerator (W. A. Benjamin, New York, 1968); pp. 515-516.
2. Linear Brief 2, National Semiconductor Corp., Santa Clara, California 95051.

APPENDIX

GRAY - BINARY CONVERTER

Conversion from gray to binary code can be accomplished by the MSD stage, which toggles for each gray code of "1" and holds for each gray code of "0". This can be expressed as follows:

$$b_i = g_i \oplus b_{(i-1)}$$

Where $\oplus$ is the exclusive OR (XOR) of the two terms, and MSD is b_0. This results in a table of functions as follows for eight bits:

$$b_0 = g_0$$

$$b_1 = g_0 \oplus g_1$$

$$b_2 = g_0 \oplus g_1 \oplus g_2 = b_1 \oplus g_2$$

$$b_3 = g_0 \oplus g_1 \oplus g_2 \oplus g_3 = b_2 \oplus g_3$$

$$b_4 = g_0 \oplus g_1 \oplus g_2 \oplus g_3 \oplus g_4 = b_3 \oplus g_4$$

etc. By factoring and redefinition, we obtain:

$b_0 = g_0$

$b_1 = g_0 \oplus g_1$

$b_2 = b_1 \oplus g_2$

$b_3 = b_1 \oplus (g_2 \oplus g_3)$: let $(g_2 \oplus g_3) = A$, then $b_3 = b_1 \oplus A$

$b_4 = b_1 \oplus A \oplus g_4$

$b_5 = b_1 \oplus A \oplus (g_4 \oplus g_5)$: let $(g_4 \oplus g_5) = B$, then $b_5 = b_1 \oplus A \oplus B$

$b_6 = b_1 \oplus A \oplus (B \oplus g_6)$: let $(B \oplus g_6) = E$, then $b_6 = b_1 \oplus A \oplus E$

$b_7 = b_1 \oplus A \oplus B \oplus (g_6 \oplus g_7)$: let $(g_6 \oplus g_7) = C$

Then

$$b_7 = b_1 \oplus A \oplus (B \oplus C): \quad \text{let}(B \oplus C) = D$$

and finally,

$$b_7 = b_1 \oplus A \oplus D \quad .$$

With this method of factoring and redefinition, all XOR gates will be arranged in three levels (where all gates in a level will operate at the same time). The resulting converter then has four gates at the input level, that is: 1, A, B and C and four gates at the second or intermediate level, that is: 2, 3, D, E and four gates at the last or third level, that is: 4, 5, 6, and 7. The maximum settling time through this converter is reduced to three gate times or about 1 μsec. With this implementation in hardware as shown, the conversion time is reduced to about 1/3 the time required by the conventional converter. This method requires much less time than performing the XOR function using software. Inverters are required in the odd binary outputs because the R131 gate produces on inverted output. The DEC R131 circuit should be referred to as a "coincidence" gate rather than an XOR.

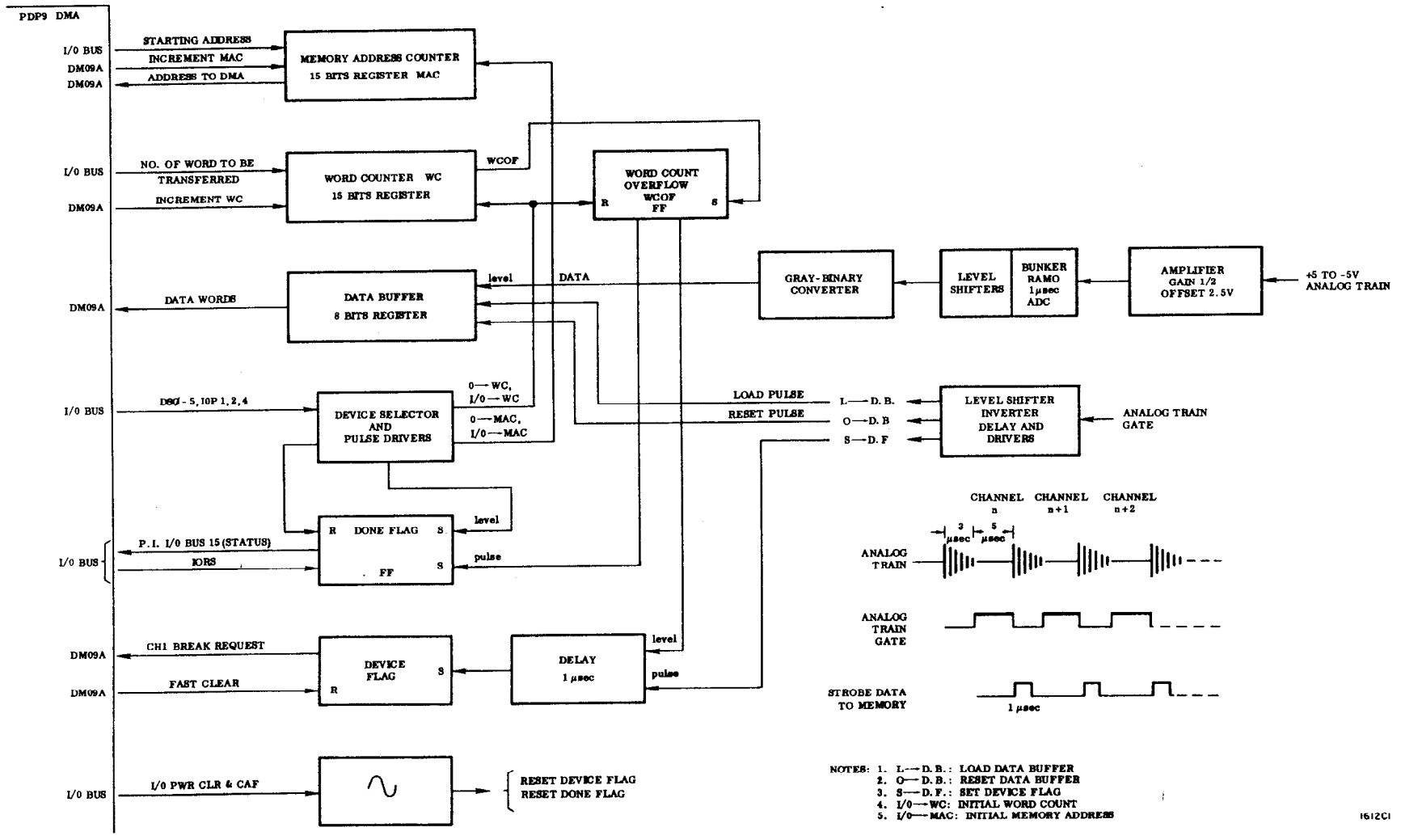

Fig. 1

Log Q — DMA interface block diagram.

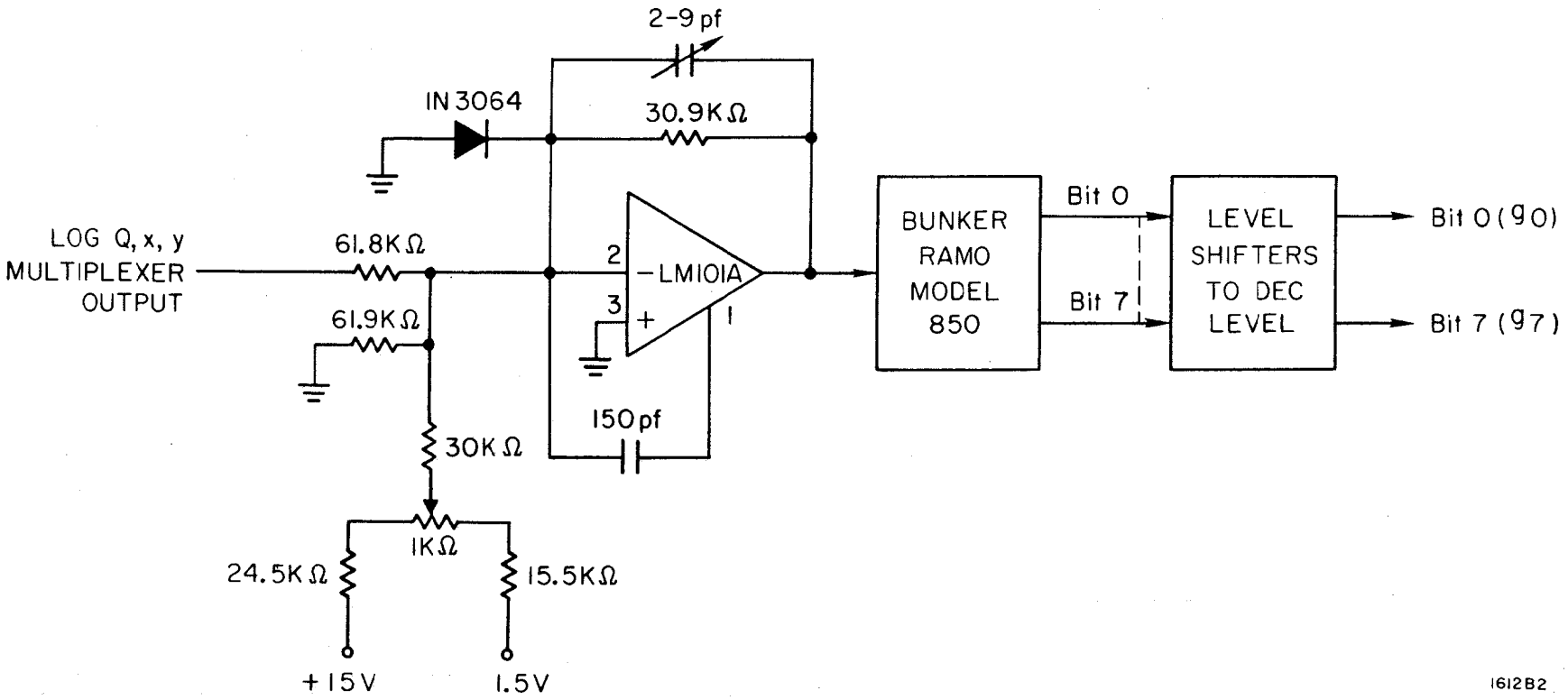

Fig. 2

Analog signal digitizer.

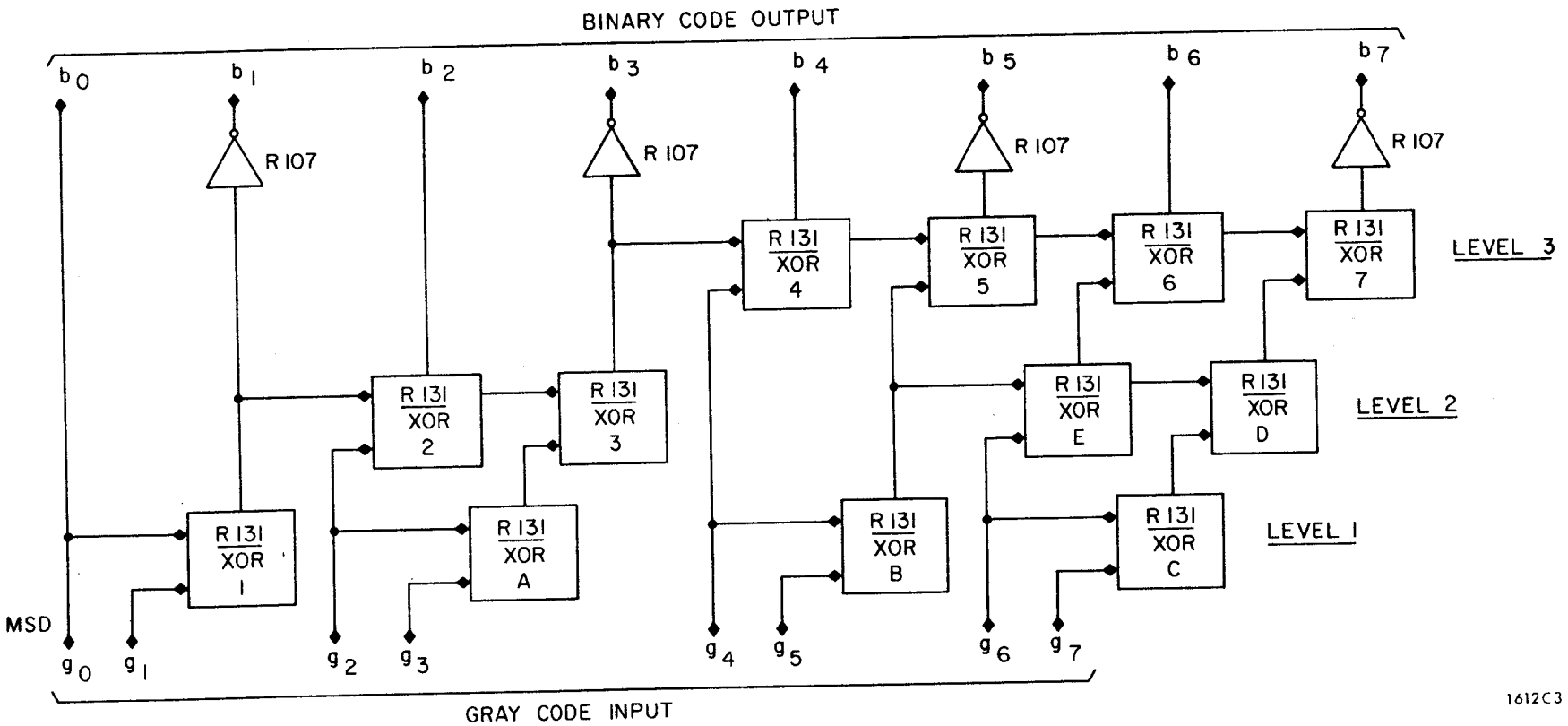

Fig. 3

DEC R131 gate version of gray to binary converter. This implementation uses coincidence gates.

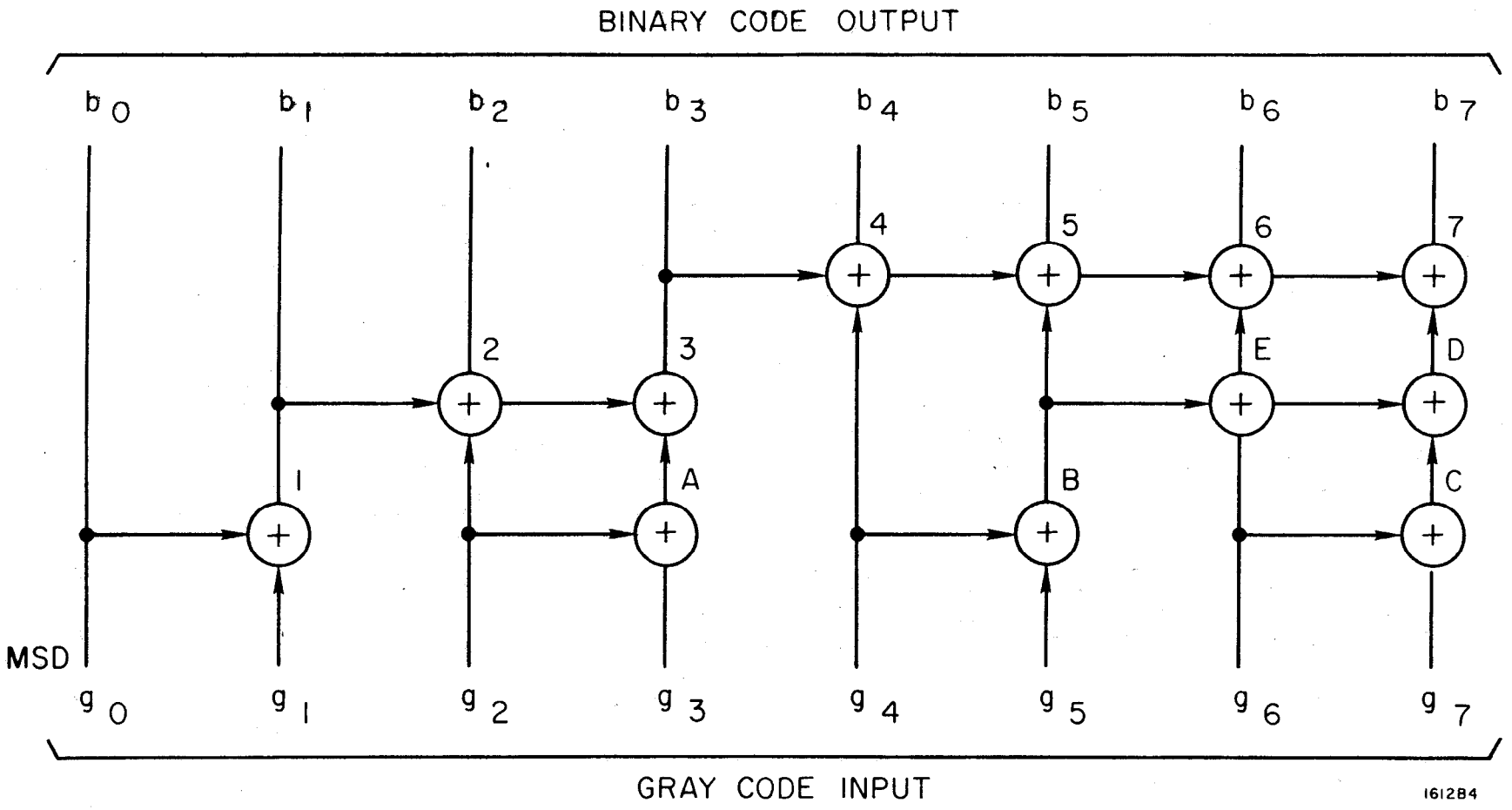

Fig. 4

Gray to binary converter implemented with XOR gates.