A020
A Cookbook Approach to High Speed Data Acquisition and Microprocessor Interfacing

The purpose of this application bulletin is to show that any competent engineer can design a complex data acquisition system as a series of manageable building blocks. It also explains the wide range of LSI devices that have recently become available, with moderate ad-mixtures of suitable MSI and SSI and discrete devices.

THE FUNDAMENTALS

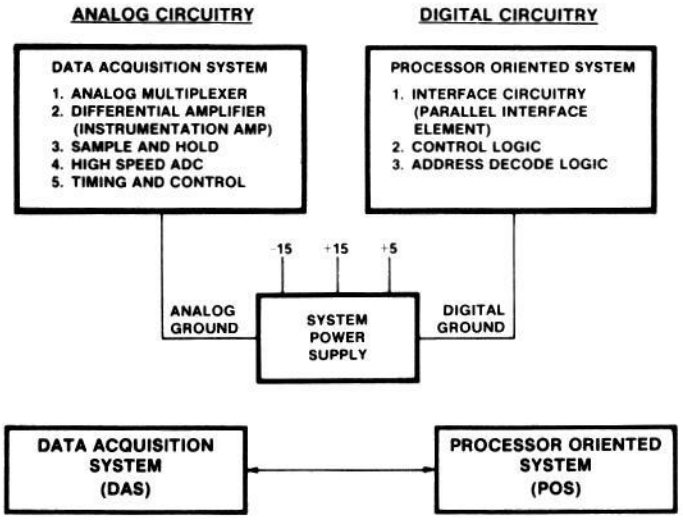


Figure 1: Basic Building blocks of Data Acquisition System

The first thing to consider when designing a Data Acquisition System (DAS) to interface to a Processor Oriented System (POS), is to recognize the physical handicaps which arise when mating the two in the same surroundings. The two biggest influences on an unhappy relationship are the analog and digital ground routing in the DAS, and noise emissions from the POS. Since the DAS depends on the accuracy of voltage levels, it is imperative to maintain that accuracy within an environment of large digital ground currents and noise. Not only must care be taken when laying out the PC board, the analog section must be as isolated as possible from its digital neighbor. The isolation can be obtained by physical separation and lots of ground plane; both analog and digital.

The analog and digital ground routing should be two separate networks originating at the system power supply as one common ground. Figure 2 shows an example of an ideal grounding scheme for an analog and digital system. It is important to ensure that the analog ground path back to the system power supply contains little or no varying currents and at the same time is as physically short and hefty as possible. This is necessary in order to keep a constant analog ground reference throughout the entire system. The separation between analog and digital ground networks should not stop at the system bus but should continue on every card in the system. Clamping diodes may be installed between the two grounds on cards containing both analog and digital circuitry, to help prevent damage to circuitry in the event of accidental separation (excess potential difference) between the two grounds.

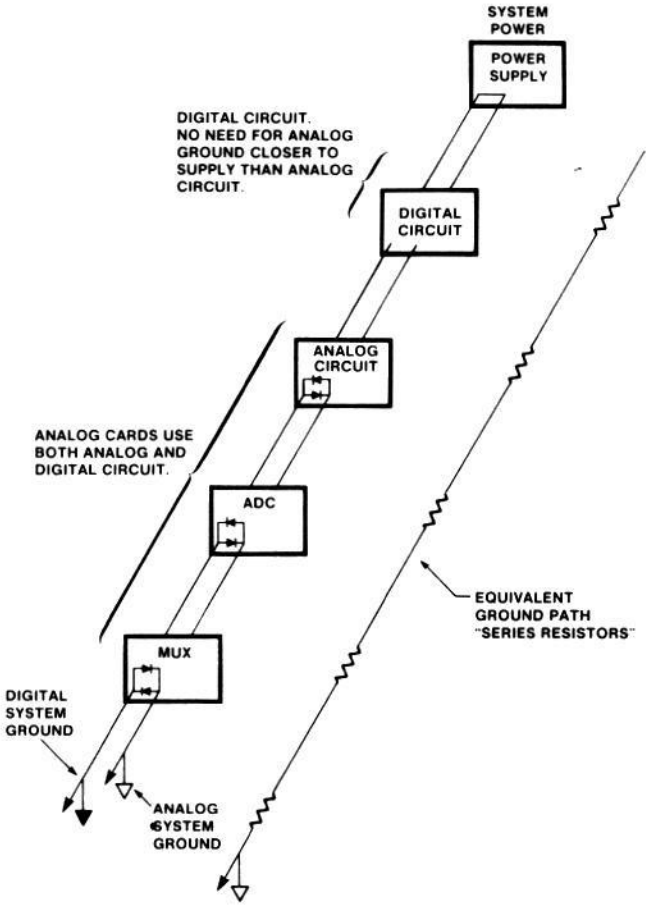


Figure 2: Ideal grounding scheme for an analog and digital system

HIGH SPEED DATA ACQUISITION - A BLOCK APPROACH

Figure 3 illustrates the basic building blocks which make up the Data Acquisition System (DAS). Starting with the input, an analog multiplexer is needed to direct the various

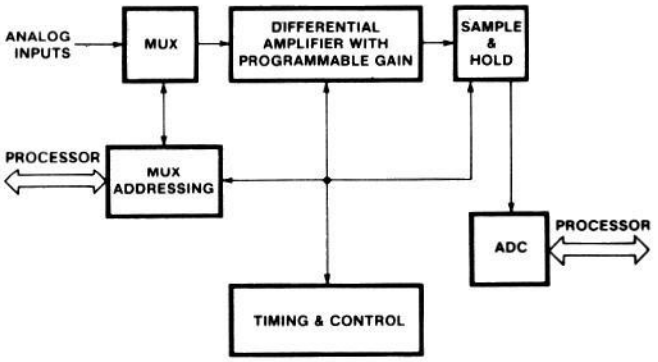


Figure 3: Block Diagram - High Speed Data Acquisition System

INTERSIL, INC., 10710 N. TANTAU AVE., CUPERTINO, CA 95014
(408) 996-5000 TWX: 910-338-0171

channels of analog information to the analog to digital converter. Before reaching the A/D converter the analog inputs will usually require some sort of signal conditioning; the second block of the DAS (the differential amplifier) performs this function. The differential amplifier or instrumentation amplifier may be assigned the task of summing a pair of analog inputs together in a differential fashion with electrically programmable gain and filtering or may function as a non-inverting buffer stage for a single multiplexer channel. The complexity of this block will depend largely on the application and the types of inputs to be digitized. Once the input signal has been conditioned for the A/D converter, it may pass through a sample and hold amplifier before reaching the A/D converter input. The sample and hold amplifier block is a prerequisite for the successive-approximation technique of Analog to Digital conversion. This is due to the nature of the conversion process, and will be discussed in more detail later.

The analog to digital converter is the heart of the DAS and its architecture is solely dependent on the through-put rate required. Although the successive-approximation approach is emphasized in this bulletin, greater resolution and accuracy may be obtained with the slower dual slope technique as this system block. The last block to be considered is the timing and control section. In any DAS system there is a definite order in which events should take place. To illustrate this point, consider the normal sequence of events following a start of conversion command in a DAS. First an input channel must be selected for digitizing at the analog multiplexer, and a programmed gain set up. Next, before a "hold" command is issued, a time delay must be initiated, to allow for the settling time of both the diff-amp and sample/hold amplifier. Finally, the A/D converter is strobed and the conversion complete signal is monitored until the end of conversion takes place. With the end of conversion comes the final digitized analog input value available at the output of the A/D converter in binary form.

PLACEMENT CONSIDERATIONS

When examining the architecture of the Data Acquisition System one may wonder why the system blocks are configured the way they are. It is readily apparent why the analog multiplexer is the input block and the A/D converter is the output block but what about the placement of the differential amplifier and the sample and hold amplifier? The answer to this question becomes apparent when studying the effect of the offset errors introduced by each block.

When considering the overall data acquisition function, the final offset trim can best be made by nulling the system offset error at the analog input of the A/D converter; this single offset adjustment requires that all offset errors introduced into the system prior to the A/D converter be constant with any gain setting of the differential amplifier. Since the multiplexer contributes no offset error and the sample and hold amplifier does, placing the sample and hold amplifier in front of the differential amplifier would cause the total offset error seen by the A/D converter to be a function of the gain setting in the differential amplifier block. Not only would this placement of the sample and hold amplifier require the DAS to have more than one offset adjustment, it would also require it to have two sample and hold amplifiers when the system is configured for differential measurements. Inserting the differential amplifier in front of the sample and hold and behind the multiplexer eliminates these problems.

MULTIPLEXER CONSIDERATIONS

When choosing an LSI device for the front end of a Data Acquisition System there are several inherent properties of the device which must be considered before the successful mating of its inputs with external circuitry can be achieved.

Input Impedance

The high input impedance of many analog multiplexing devices can be deceptive because of the dynamic properties that limit the maximum usable source impedance to a relatively low value. Figure 4 shows an impedance model for a typical IC multiplexer. At the bottom of Figure 4 is the equivalent circuit for an ON channel which consists of the ON-resistance of the channel, in series with a 35pF capacitor forming a low pass filter. This capacitance becomes an important consideration when determining the maximum usable source impedance of the data acquisition system.

Since the through-put rate of the Data Acquisition System is dependent upon the individual settling times of each block, it is essential to keep the multiplexer settling time to a minimum.

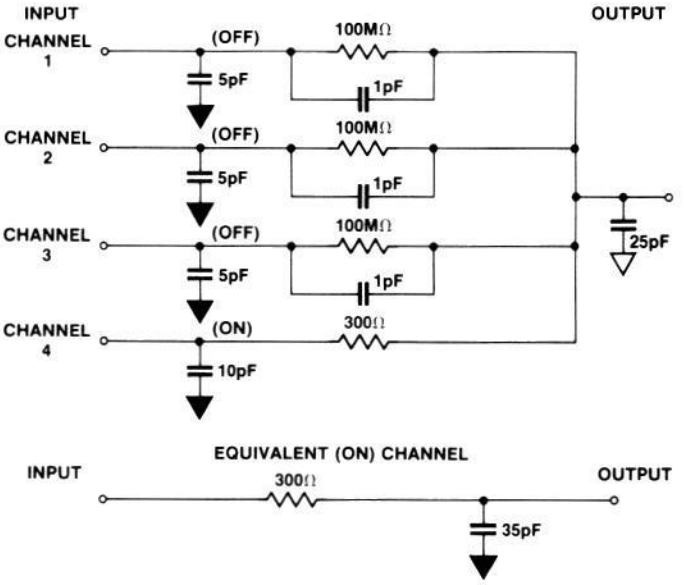


Figure 4: Analog multiplexer equivalent circuit for source impedance calculations.

With a settling time of approximately five to eight micro seconds for the differential amplifier and sample and hold amplifier, it would be beneficial to keep the multiplexer's maximum settling time for any channel under one micro second. With this in mind, and using the model in Figure 4, the maximum usable source impedance can be calculated as follows.

Assuming the resolution and accuracy of the system to be 12 bits, a settling time of 9.2τ ($\tau = (R_{SOURCE} + r_{DS(ON)})\, C_{MUX}$) is required to settle an input signal to 1/2 of the least significant bit (LSB) or .01%. Therefore, with an output capacitance of 35pF and on ON resistance of 300Ω, the settling time is (9.2) (R_{SOURCE} + 300) (35 x 10^{-12}). Substituting 1 microsecond for the settling time and solving for R_{SOURCE}, the maximum usable source impedance is 2.9kΩ. A more exact determination of the maximum usable source impedance can be made by adding the capacitance of external components, particularly those of the sample-hold circuit, and printed circuit traces before making the calculation.

Leakage

A very influential contributor to the DC errors in the front end of a data acquisition system is the multiplexer leakage current over temperature. Even though at room temperature the leakage current may be only a few nano-amps, at 70° C it may increase to several micro-amps. This characteristic of the multiplexer must be considered when a wide temperature operating range is required of the data acquisition system.

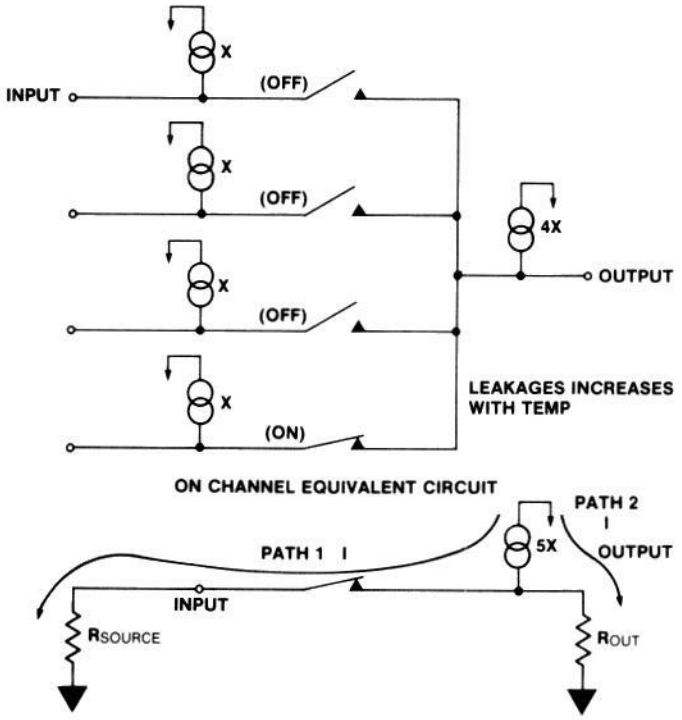


Figure 5: Analog Multiplexer equivalent circuit for leakage current considerations.

At the bottom of Figure 5 is a model of an ON channel of a multiplexer, for leakage current considerations. It can be seen that the leakage current has two paths to follow; either through the input or output of the multiplexer. This current, which is the sum of the individual channel leakages, will almost always flow through the input into the source impedance of the external circuitry. This is because of the high input impedance of the differential amplifier which follows the multiplexer. With the leakage current taking the path of least resistance into the source impedance at the multiplexer input, another error is introduced into the data acquisition function. This is an offset error, which varies with temperature and has a magnitude equal to the product of the leakage current and the source impedance. To illustrate the magnitude of this error, consider a 16 channel multiplexer whose selected channel has an input impedance of 3kΩ. Assuming the device is spec'd at 70°C to have 500nA of leakage per channel, the total leakage current would be 8μA. Now with 3kΩ at the input, the 8μA of leakage will introduce 24mV of offset error; equivalent to ten LSBs for a 12 bit system if the input range is 0-10 volts.

Since this characteristic of the analog multiplexer can contribute a significant offset error to the system over temperature, the multiplexer selection must be carefully made when considering overall system performance.

Latch-up

Another consideration which must be made when selecting a multiplexer is whether or not the device will be subject to latch-up. Latch-up is an SCR type of action which the multiplexer may enter when one of the 15V power supplies powering the device, especially the +15V supply, falls below the selected channel's input level.

To illustrate this point, consider an OFF channel of a CMOS multiplexer (Figure 6). The input to this OFF channel is −10V, and the output is at +10V, probably held there by another ON channel. Now if the +15V supply were to drop below 9.3V for any reason, the SCR action would take place, freezing this channel ON. The input would then be effectively shorted to the multiplexer output and to the selected ON channel, not to mention the power supply. This is an undesirable condition which may occur at power on, during a power supply "glitch" or during momentary shut down of a power supply, etc. Not only is this an unhealthy mode for the multiplexer, but undoing it requires that all power to the device is turned off and then back on in the proper sequence. This means that when a multiplexer with this possible condition is selected for the front end of a DAS, special circuitry must be implemented to prevent the SCR action from taking place at power on or during power failure.

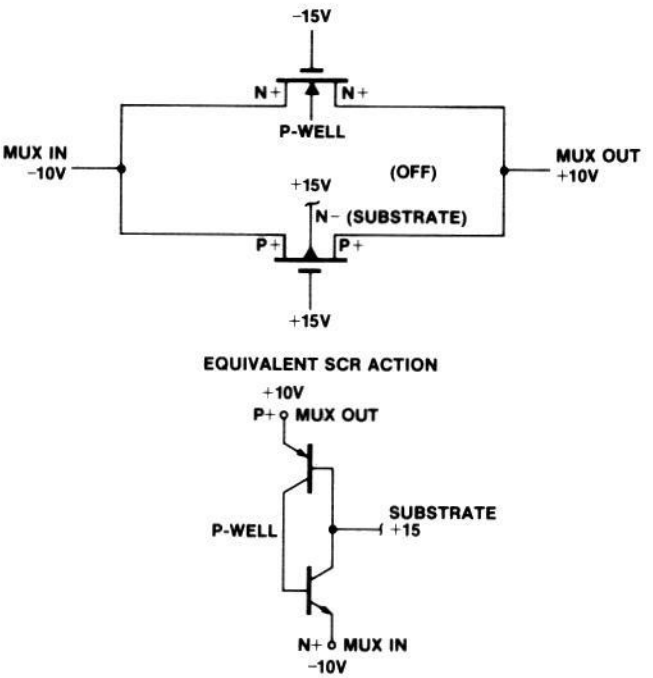


Figure 6: Analog Multiplexer off channel for latchup considerations.

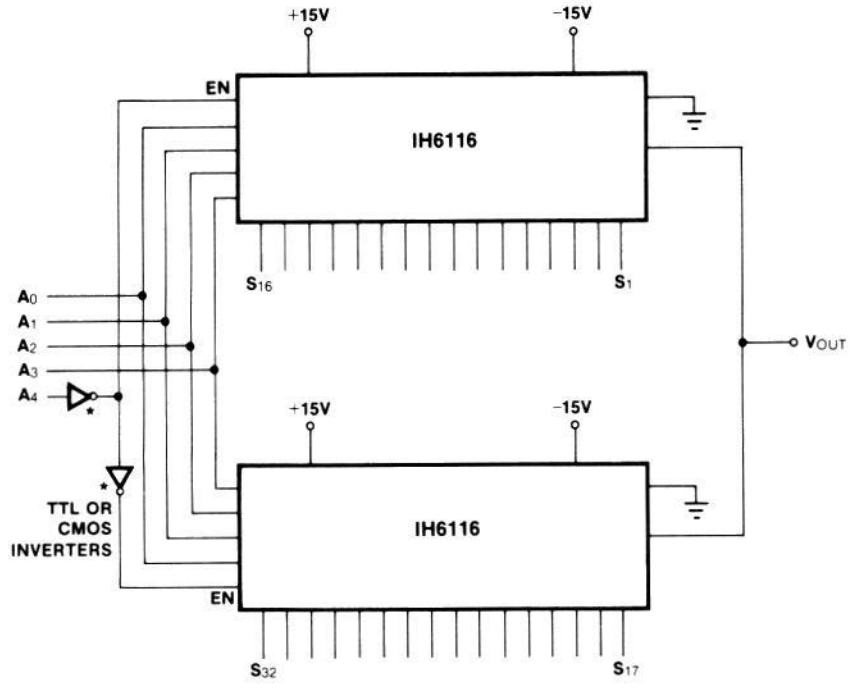
The best way to eliminate this problem is to select a device which has no latch-up problems. See Intersil Application Bulletin A006 for a description of some latch-up proof analog multiplexers and switches.

Cascading

As a general rule, the cascading of multiplexers is easily accomplished. See Figure 7. When the outputs of two multiplexers are connected together to form one larger multiplexer, the output capacitance and the leakage current doubles. If the DAS were originally tuned for maximum through-put rate and input source impedance with one multiplexer in the system, cascading two or more multiplexers in this fashion could degrade system performance drastically. This problem can be somewhat overcome by adding a third tier of submultiplexing, as shown

in Figure 8. Both leakage current and output capacity are reduced significantly, however channel ON resistance and switching times are now increased. This increase is generally insignificant when compared to that of Figure 7, and will not usually hinder system throughput performance. The benefits obtained by submultiplexing in this way will substantially increase with the number of channels to be accessed.

1 OUT OF 32 CHANNEL MULTIPLEXER USING 2 IH6116'S



* TTL inverter must have resistor pullup to drive EN input.

DECODE TRUTH TABLE

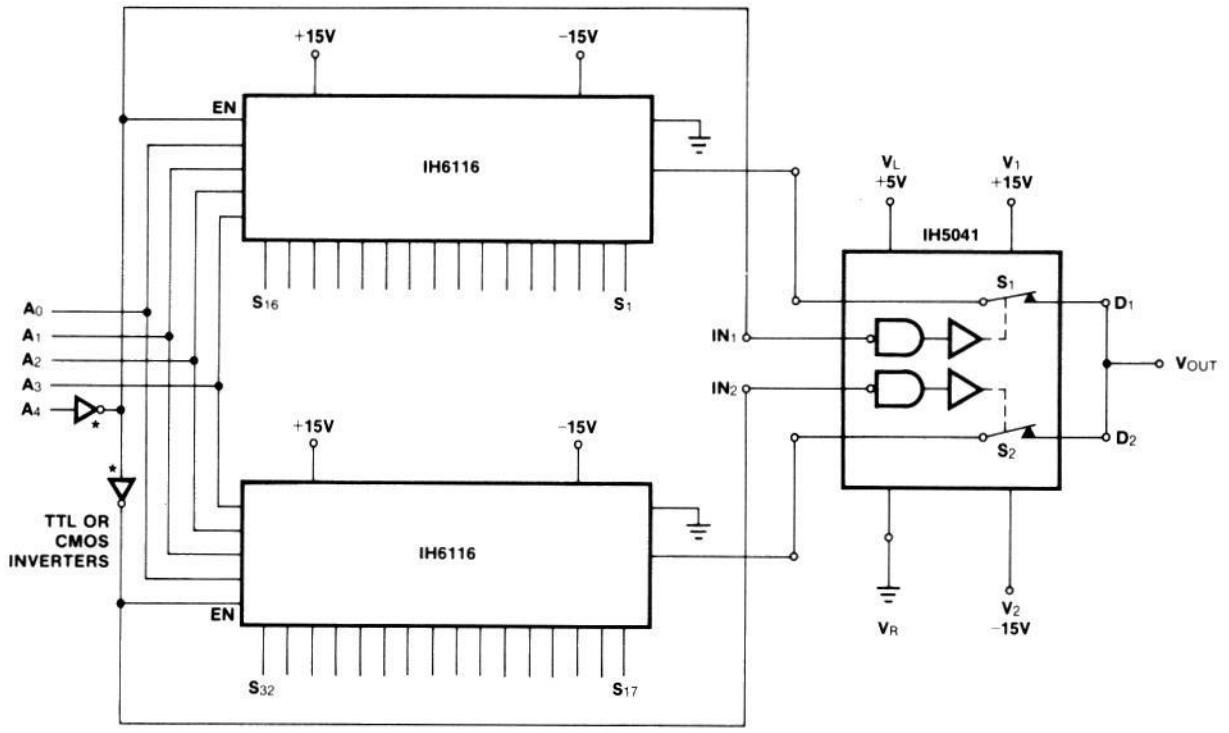
A_4	A_3	A_2	A_1	A_0	ON SWITCH
0	0	0	0	0	S1
0	0	0	0	1	S2
0	0	0	1	0	S3
0	0	0	1	1	S4
0	0	1	0	0	S5
0	0	1	0	1	S6
0	0	1	1	0	S7
0	0	1	1	1	S8
0	1	0	0	0	S9
0	1	0	0	1	S10
0	1	0	1	0	S11
0	1	0	1	1	S12
0	1	1	0	0	S13
0	1	1	0	1	S14
0	1	1	1	0	S15
0	1	1	1	1	S16

DECODE TRUTH TABLE (cont'd)

A_4	A_3	A_2	A_1	A_0	ON SWITCH
1	0	0	0	0	S17
1	0	0	0	1	S18
1	0	0	1	0	S19
1	0	0	1	1	S20
1	0	1	0	0	S21
1	0	1	0	1	S22
1	0	1	1	0	S23
1	0	1	1	1	S24
1	1	0	0	0	S25
1	1	0	0	1	S26
1	1	0	1	0	S27
1	1	0	1	1	S28
1	1	1	0	0	S29
1	1	1	0	1	S30
1	1	1	1	0	S31
1	1	1	1	1	S32

Figure 7: Easy cascading of two multiplexers can be accomplished by connecting in parallel.

1 OUT OF 32 CHANNEL MULTIPLEXER USING 2 IH6116S AND AN IH5041 FOR SUBMULTIPLEXER



*TTL gate must have resistor pullup to +5V to drive "EN" input.

DECODE TRUTH TABLE

A_4	A_3	A_2	A_1	A_0	ON SWITCH
0	0	0	0	0	S1
0	0	0	0	1	S2
0	0	0	1	0	S3
0	0	0	1	1	S4
0	0	1	0	0	S5
0	0	1	0	1	S6
0	0	1	1	0	S7
0	0	1	1	1	S8
0	1	0	0	0	S9
0	1	0	0	1	S10
0	1	0	1	0	S11
0	1	0	1	1	S12
0	1	1	0	0	S13
0	1	1	0	1	S14
0	1	1	1	0	S15
0	1	1	1	1	S16

DECODE TRUTH TABLE (cont'd)

A_4	A_3	A_2	A_1	A_0	ON SWITCH
1	0	0	0	0	S17
1	0	0	0	1	S18
1	0	0	1	0	S19
1	0	0	1	1	S20
1	0	1	0	0	S21
1	0	1	0	1	S22
1	0	1	1	0	S23
1	0	1	1	1	S24
1	1	0	0	0	S25
1	1	0	0	1	S26
1	1	0	1	0	S27
1	1	0	1	1	S28
1	1	1	0	0	S29
1	1	1	0	1	S30
1	1	1	1	0	S31
1	1	1	1	1	S32

Figure 8: Using another tier of submultiplexing to reduce the effects of output capacitance and leakage current.

Single Ended or Differential

The selection of either a single ended or differential multiplexer for the front end of the DAS is dependent upon the application. Since the 16 channel IC multiplexer is the basic building block, the designer must decide whether to go with 16 single ended or 8 dual differential inputs. The types of analog inputs to be digitized will dictate which type is used.

If the transducer circuitry to be monitored is far enough away from the DAS to allow excessive interference pickup on the return cable, the need for a shielded twisted pair and a differential front end becomes likely. With the differential front end, the common mode "noise" signal on the twisted pair will be rejected at the differential amplifier provided that its magnitude does not exceed the input voltage range of the DAS. If the magnitude of noise is greater than the input range, the signal to be digitized will be lost. There are several methods for overcoming this problem. First, the "noise" source could be eliminated; second, the cable length from the transducer to the DAS could be shortened; third, a very expensive high voltage differential amplifier could be used to buffer the multiplexer; finally, local conversion of transducer signals could be established to allow transmission of digital (serial) signals over long distance to the central processor instead of the low level analog signals produced by most transducers.

Even though the first and second solutions would eliminate the noise magnitude problems, implementing either of these two solutions would probably require the moving of some heavy piece of equipment away from the cable (a generator for example) or closer to the DAS (a smelting pot for example).

Therefore, a choice between solutions three and four must be made. After examining the cost for each solution, the local conversion technique is usually most viable.

THE DIFFERENTIAL AMPLIFIER

The differential amplifier becomes an essential part of the DAS when low level transducer signals must be recovered from noise. The complexity of this block may vary from a single op-amp buffer all the way to a software programmable signal conditioner for several different types of transducers (pressure, temperature, flow, etc.).

Before deciding on the configuration needed for a particular application, there are a few parameter prerequisites which must be met, no matter how simple or complex this block of the DAS is. First, the differential amplifier circuitry following the multiplexer must have a high input impedance. This is necessary to avoid the effects of the unpredictable multiplexer channel ON-resistance. This resistance, which varies with everything (voltage, current, temperature, etc.), must not be a part of the overall data acquisition function; in other words, if the resistance of the ON-channel were to double for some reason, there should be no noticeable change at the output of the differential amplifier. Next, the common mode rejection ratio of the differential amplifier should be better than 80db. This insures that for ±10 volts of common mode input noise only 1mV could slip through to the output. (Again, we are assuming a 12 bit system.) Another important factor, especially in DASs of 12 bits or more, is the stability of critical components over temperature. Maintaining 12 bits of absolute accuracy over any reasonable temperature range can be very expensive, and this aspect of the system's design must be thoroughly examined before specifying any critical components.

Programmable Gain Changing Provisions

The provision of programmable gain at the input stage of a DAS should be considered only after it has been determined that a single, optimum gain setting will not give satisfactory performance. Not only is it expensive to incorporate software programmable gain into a DAS, it almost always ends up being the greatest single source of errors within the system.

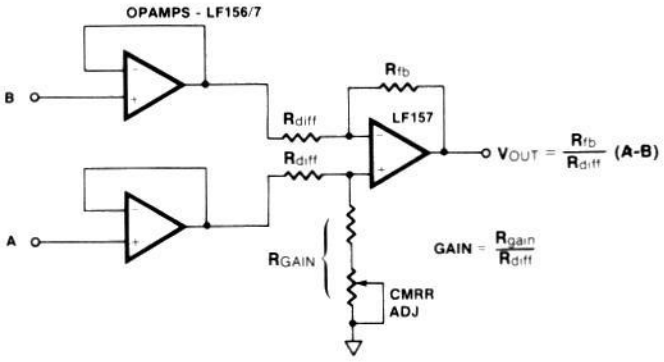


Figure 9: Fixed gain high impedance input. Differential amplifier with CMRR fine adjustment.

Figure 9 shows all that is required for a fixed gain high input impedance differential amplifier; Figure 10 shows the same differential amplifier with the software programmable gain feature. The actual differential amplifier is made up of two LF156s, for input buffers, and one LF157 (better CMRR) for the differential stage. Since better than .005% ratio matching of the gain resistors would be needed to obtain a CMRR of 80db, a potentiometer is used for fine tuning the CMRR at low frequencies. The high frequency CMRR is dependent entirely on the op amp selected for the differential stage, and for the LF157 this is 90dB at 1kHz.

Considering the difficulty in maintaining a high degree of common mode rejection, the addition of a programmable gain feature should not be included within the differential amplifier stage itself, but rather be a separate stage following the differential amplifier, as shown in Figure 10. This

b	a	GAIN
0	0	X1
0	1	X2
1	0	X4
1	1	X8

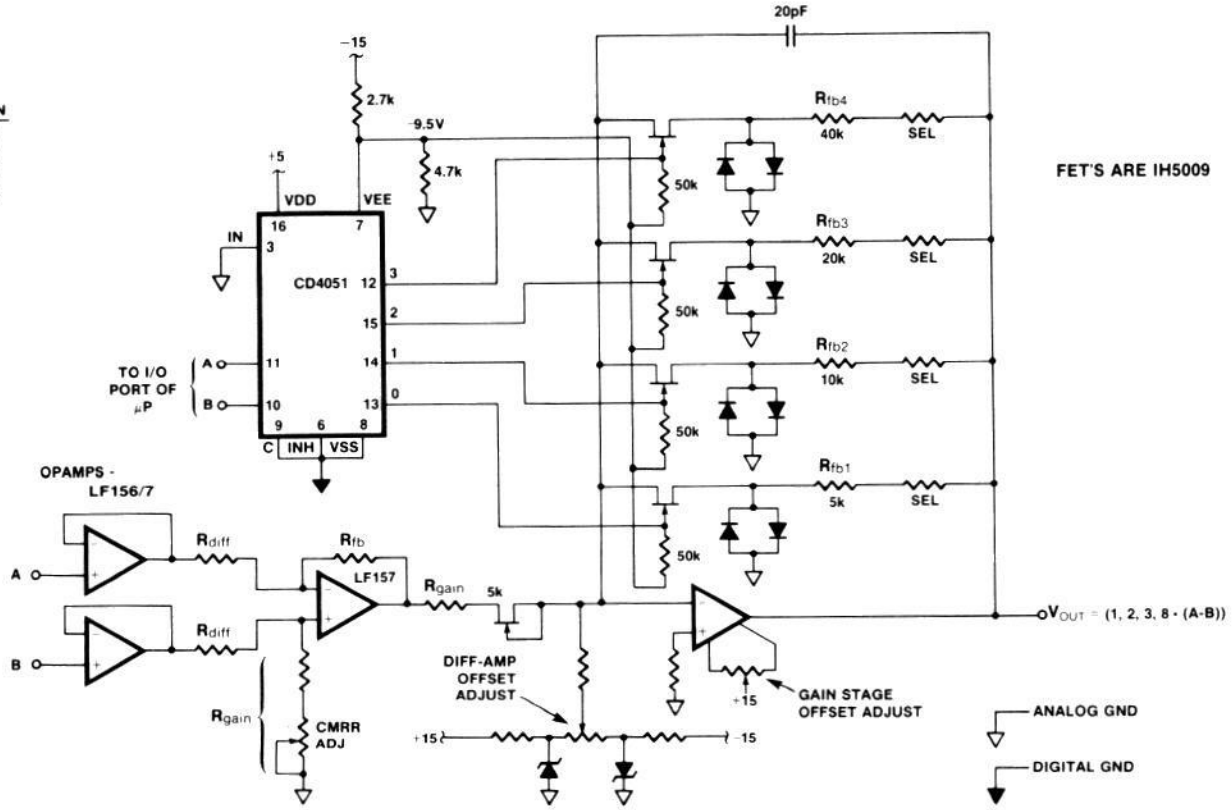


Figure 10: Programmable Gain Differential Input Amplifier

particular programmable gain circuit employs a CD4051 (CMOS Analog Multiplexer) as a two to four line decoder, with appropriate FET drive for switching between feedback resistors to program the gain to any one of four values. The problems associated with a circuit of this type are rather obvious. First, it was previously determined that one offset adjustment for the entire DAS would be advantageous. Not only does the programmable gain stage require offset adjustment, but both the offset of the gain op amp and the offset of the differential stage must be nulled. This is to prevent the offset error of the system up to this point from becoming a function of the programmed gain. Probably the biggest disadvantage in using a programmable gain stage is the initial calibration required; in addition to the two offset adjustments, there are four gain adjustments which must be carefully made by selecting FETs and feedback resistors. The calibration of this gain stage is not only troublesome, but is also very time consuming and, for a high volume production environment, not very appealing.

SAMPLE AND HOLD AMPLIFIER

Successive approximation is the most popular technique for high speed analog to digital conversion. This technique requires that the analog input signal remain constant during the entire conversion process, and for this reason, a sample and hold amplifier is needed to buffer the successive approximation A/D converter. Like the other blocks of the DAS, the proper selection of sample and hold circuitry is necessary to insure maximum system performance.

Aperture Time

Aperture time is defined as the time required by the sample and hold to switch from a tracking mode into a hold mode, once a hold command is given. The aperture time is also the characteristic of the sample and hold circuitry which limits the maximum input frequency or slew rate for which the DAS can accurately sample. For example, in a 12 bit system with a 10 volt full scale input range and an aperture time of 100ns, the maximum input slew rate would be 10mV/μs, or 300Hz (dV/dt = 2πVpk maximum slew rate of a sine function). This is due to the input changing more than 1/2 LSB after the command to sample is given, and before the sample is taken. Even though the aperture time limits the maximum input frequency for real time event recording (exact input value at t = Xμs after starting the event), higher frequency input may be sampled at system accuracies. Applications do exist where aperture time errors are not important, since the sampling of the input data is not referenced to a point in time. An example of this would be the recording of a high speed event (vibration analysis) for graphical analysis. The only requirement is that the sampled data be equally spaced with respect to time. If the aperture uncertainty (the difference in aperture time from sample to sample) is small in comparison to the aperture time, the aperture time is really nothing but a time delay from the time when the command to sample was given to the time when the actual sample was taken. Thus the aperture error can effectively be nulled by taking the sample one aperture time period before the time when it would normally have been taken.

Acquisition Time

This is defined as the time required for the sample and hold amplifier to slew and settle to the input signal when switched from hold to sample mode. Along with the settling time of the Differential Amplifier, the acquisition time of the sample and hold also effects the maximum throughput rate of the DAS. For this reason it is important to select a sample and hold amplifier with an acquisition time as low as possible ($\leq 5\mu$s). If the sample and hold amplifier to be used is an IC type, the input step magnitude for which the acquisition time is specified should conform to the input requirements of the sample and hold block. Figure 11 shows the various relationships between the acquisition time and the other parameters of the sample and hold amplifier.

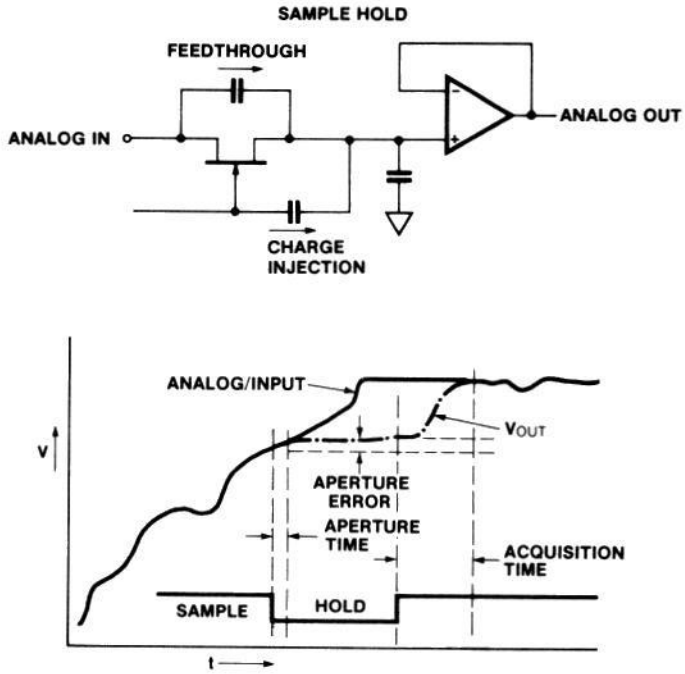


Figure 11: Properties of the Sample and Hold displayed graphically.

Feed Through

When the Sample and Hold Amplifier is in the hold mode, the amount of input voltage change seen at the output is considered the feed through. Feed through is caused by the stray capacitance from the analog input of the sample and hold to the top of the sampling capacitor (across the input switch; see Figure 11). This capacitance forms a capacitive voltage divider with the sampling capacitor, and allows a fraction of the input signal to feed through to the output when the input switch is open and the sample and hold is in the hold mode. The effect of the stray capacitance can be minimized by careful printed circuit layout and by selecting a larger value of sampling capacitor, however tradeoffs between the amount of feed through and the acquisition time may have to be made when selecting the sampling capacitor, because of the change in the AC characteristics of the sample and hold circuitry.

Charge Injection

Charge injection results in an error to the sampled value when switching from the sample mode to the hold mode. A voltage step is created on the sampling capacitor by the charge injected through the capacitance between the gate of the input switch and the sampling capacitor (see Figure 11). Circuit board strays may also contribute to this capacitance. Again this effect may be minimized by careful printed circuit layout and a larger value of sampling capacitor. However, although the charge injected causes a voltage step on the sampling capacitor, and therefore is a source of error, the fact that it is relatively constant in magnitude means that it can be nulled as a system offset if not too severe.

Droop Rate

Droop Rate is the rate at which the sampled voltage decays at the output of the sample and hold amplifier during the hold mode. The rate of decay is proportional to the sum of the sample and hold leakage and bias currents which may either charge or discharge the sampling capacitor during the hold period. To calculate the maximum droop rate simply divide 1/10 LSB of the A/D converter's lowest input range by its conversion time. This will result in a maximum value of droop rate, and at the same time not produce any significant error which could degradate system accuracy.

ANALOG TO DIGITAL CONVERTER

The successive approximation technique of analog to digital conversion is by far the most popular technique for high speed, high accuracy, microprocessor compatible analog to digital converters. Conversion times as low as two microseconds with twelve bits of resolution and accuracy are obtainable.

Figure 12 shows a block diagram of the successive-approximation analog to digital converter (ADC). The basic ADC loop consists of a successive approximation register (SAR), a current output digital to analog converter (DAC) and a comparator. To initiate a conversion, the start conversion input is pulsed and the conversion sequence begins. The SAR initially sets up one "Ø" and the rest "1's" on its outputs, equivalent to half-scale minus a least significant bit (LSB). Assuming unipolar operation, the digitizing of the analog input signal proceeds as follows. The initial setting of the SAR outputs programs the DAC to half-scale minus an LSB, and that value of current flows into the DAC. The magnitude of the input will determine the polarity of the comparator output, greater than half scale being positive. This in turn signals the SAR to make a decision on the most significant bit (MSB) with the arrival of the first pulse from the system clock. On the rising edge of the first clock pulse the SAR programs the MSB to its final value, and at the same time sets the second bit to a logic low. This allows the SAR to make a decision on the second bit, and set up for the third bit on the rising edge of the second clock pulse. This process will continue in descending bit order until the LSB has been programmed. At that time the conversion complete signal will change states, signifying an end of conversion, with the final digitized input remaining latched at the outputs of the SAR.

ADC DESIGN

The two primary factors which control the complexity of the ADC design are accuracy and speed. The accuracy specification directly specifies the resolution (accuracy) required of the ADC, however the throughput rate does not directly specify the speed. To calculate the conversion time required, subtract from the inverse of the throughput rate (in hertz) the sum of the settling times of all blocks up to the ADC. For example, assume that the throughput requirement for a DAS is 30 kilo-samples per second. If the multiplexer, differential amplifier and sample and hold amplifier have a combined settling time of 8 microseconds, the ADC would have to convert in 25 microseconds or less to maintain the 30 kilohertz throughput rate. It turns out that the design of 12 bit or less converters with conversion times greater than 25 microseconds is relatively easy. However, as the conversion time drops below 20 microseconds, the difficulty of design seems to increase, somewhat exponentially.

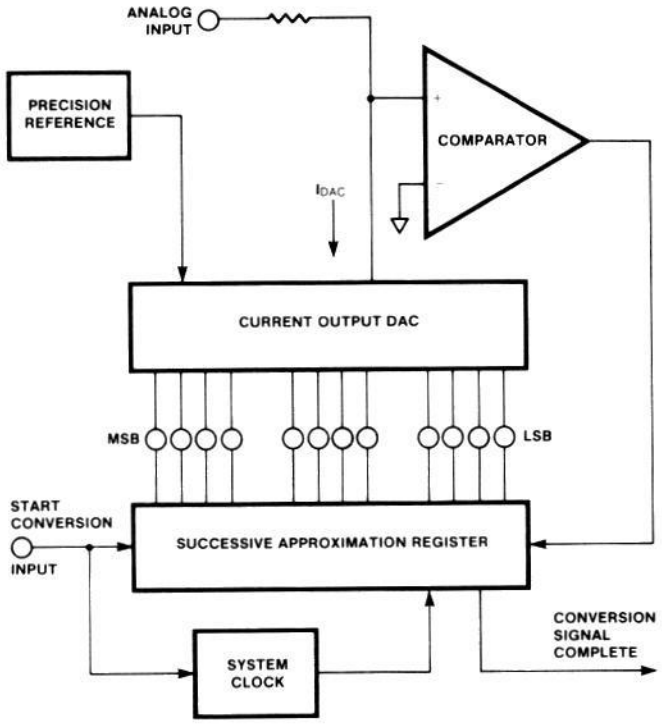


Figure 12: Block Diagram of a 12 bit Successive Approximation Analog to Digital Converter.

Loop Speed

Several things limit minimum conversion times. Looking at Figure 12, the slowest point in the loop is the node where the DAC output, comparator input and analog input join together. To understand why this is so, it is necessary to remember what is happening at this node during the conversion process. During a conversion, the SAR programs the DAC output current in an attempt to equal the current through the input resistor, by the time the conversion ends. While the successive approximation process is taking place, the difference between the two currents (which will vary in magnitude throughout the conversion) will charge the nodal capacitance which drives the comparator's output to the proper polarity to signal the SAR. The comparator's speed problem (due to lack of overdrive) evolves when this current difference is very small (LSB levels), and insufficient time is available before the next clock pulse for the nodal capacitance to charge to a voltage large enough to switch the comparator.

The inadequate overdrive results in a conversion error. The use of a high speed comparator and a high current DAC will help eliminate this problem, which is quite severe in sub 20μs converters.

The Comparator

The loop comparator must be able to switch fast with very little overdrive. A very fast comparator is the National Semiconductor LM361, with a switching time of typically 14ns with 5mV of overdrive. Not only is it fast, it is also very expensive; however, for sub 20μs conversion times such speed is a must. Figure 13 shows a comparator scheme using an LM301 op-amp in a feed forward configuration, combined with an LM311 comparator to bring high speed at low cost. This particular comparator, along with a fast settling current output DAC, will provide 12 bit analog to digital conversion with conversion times as low as 25 μs.

Probably the most difficult aspect of using a high speed comparator in a successive approximation loop is the

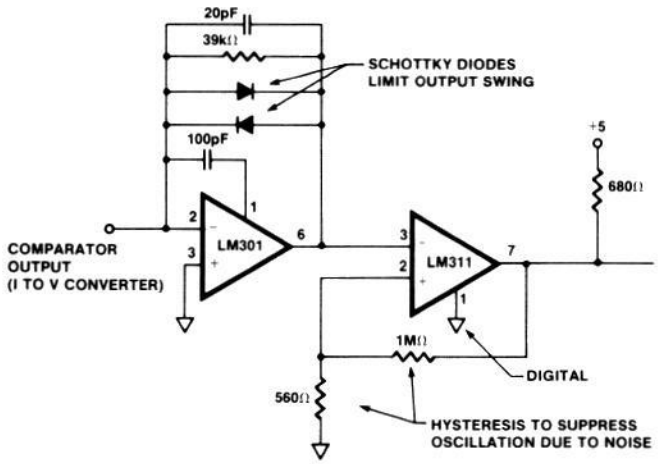


Figure 13: Cheap high speed comparator.

problem of maintaining frequency stability while operating within the linear region of the comparator (when very low overdrive is present at comparator inputs). Since the comparator is operating at a high gain-band-width with minimal feedback compensation, even small amounts of parasitic feedback will cause oscillations. To avoid this, careful PC layout must be observed. Lots of analog and digital ground planes and extensive bypassing of the supplies is recommended. Comparator output to input isolation, along with analog and digital signal isolation, should also be considered during PC layout. It is also a wise idea to buffer the load driven by the comparator output with a discrete transistor, thereby eliminating thermal feedback effects which could cause instability.

The Digital to Analog Converter

The DAC within the system ADC is the one element of the entire DAS which is most responsible for maintaining overall system accuracy. The DAC, therefore, must not only be fast settling, but it must also be linear, monotonic and stable over temperature.

What type of DAC should be used in the system ADC? Again it depends on the DAS throughput requirements. For sub 25μs conversion times, the author recommends a multiple IC high current DAC (see A010 for such a device) along with a high speed comparator like the LM361. However, for conversion times greater than 25μs the system DAC can be implemented very easily with a monolithic current output type device. The Intersil AD7541 is an excellent example of a laser trimmed monolithic multiplying current output DAC with true 12-bit accuracy and resolution. Once the monolithic DAC is selected, all that is usually required to make it play is power and a reference. The reference must be added externally to get the required temperature stability, since on board monolithic references are generally very poor in this regard. The most widely used reference device is a temperature compensated zener diode along with an op-amp for gain buffering.

Successive Approximation Logic

Figure 14 shows how to combine a standard TTL 7400 quad-NAND gate with an Intersil AM2504 successive-approximation register to obtain all the necessary logic to support a 12-bit A/D converter. NAND gates B, C, and D form an oscillator circuit, controlled by the SAR signals, start (S) and conversion complete (CC). The Start Conversion input requires a positive pulse, buffered and inverted by gate A. On the positive transition of this pulse, gate C will provide the CP input of the SAR with a positive transition. This initializes the SAR by setting the CC output (conversion complete) signal to a logic one state, and the bit outputs to a condition of one zero and the rest ones, the MSB being equal to zero. The RC delay between the output of gate A and the input of gate C insures that the S input will have the necessary set up time before the CP input is clocked by the rising edge of the start conversion pulse. As soon as the start conversion input pulse returns to a logic low the conversion will begin. It is important to note the conversion complete signal (CC) represents the conversion time of the ADC plus the pulse width of the start conversion input pulse. Once the conversion process starts, gates B, C and D will provide the SAR with a series of clock pulses at a frequency set by the clock timing capacitor. The hysteresis effect of the 200Ω and 2kΩ resistors around gates C and D help provide a fast, clean clock for the CP input of the SAR. As soon as the twelfth clock pulse strobes the CP input, the SAR responds by making a decision on the twelfth bit and resetting the CC signal to a logic low signifying an end of conversion.

A serial form of the conversion is also available at the serial data output (DO). Used along with a delayed clock signal as a strobe, this output feature provides one means of transmitting the digitized analog input signal over long distances with only three wires.

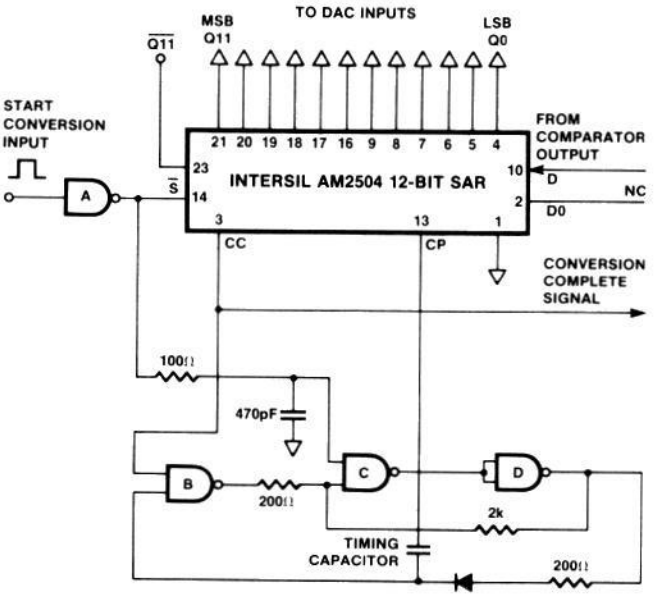


Figure 14: Successive-approximation logic for a 12 bit analog to digital converter.

Other Considerations

The A/D converter can be set up to operate in any one of many input range configurations. The three most popular are Straight Binary, Offset Binary and Twos-Complement Binary. Straight Binary is the unipolar input range, for which inputs of zero and full scale volts result in output codes of all zeros and all ones respectively. Figure 12 shows the Straight Binary configuration where the value of the input resistor is selected to produce the desired input range with respect to the full scale current of the DAC. Offset Binary can be obtained from the Straight Binary configuration of Figure 12 by adding half scale current to the summing input of the comparator in the form of a current source or a resistor to a positive reference. The only difference between Offset Binary and Straight Binary is a negative shift in the input range of half scale volts. This input shift, or offset, allows both positive and negative inputs to be digitized (Bipolar

input range). An input voltage of minus full scale results in an output code of all zeros and plus full scale results in all ones. Twos-complement Binary is almost identical to Offset Binary, the only difference being the Twos-complement code has an inverted MSB. To obtain a twos-complement operation from Figure 12, set up for the offset binary configuration as previously described then read the digital output using an inverted MSB. The AM2504 successive-approximation register makes this easy by providing both MSB polarities. The MSB (pin 21) must be connected to the DAC for the successive-approximation process to work but, for twos-complemented Binary, "$\overline{\text{MSB}}$" is read as the most significant bit of the digital output word.

An interesting feature of the AM2504 SAR is the ease with which it can be short-cycled. Short cycling of the SAR is the process of reducing both the resolution and conversion time of the ADC by using the new LSB+1 bit as conversion complete signal for the SAR logic, instead of the CC output of the SAR. This permits the original 12-bit DAS to operate at a much higher throughput rate, albeit with less resolution. In many instances, not all of the DAS transducer inputs will require the 12 bits of resolution and accuracy available, and when this is the case the throughput rate can be optimized by software control of the ADC resolution and clock frequency. During the conversion process the 12-bit ADC requires the comparator input to settle in ten time constants (1/2 LSB), which is equal in time to one period of the ADC's clock (assuming the clock is set to a maximum frequency while still maintaining 12-bits of accuracy). Short cycling the 12-bit converter to 8-bits would decrease the minimum conversion time by 58%, since only 6.2 time constants are now required for the comparators input to settle to .2% or 1/2 LSB at 8-bits ($8 \times 6.2\tau$ instead of $12 \times 10\tau$).

ANALOG SECTION AS A WHOLE

Figure 15 shows the complete analog section of a 12-bit, binary, 2's complement DAS with the timing and control necessary to interface the analog components to each other and to a microprocessor.

The front end of the DAS is configured differentially using a dual eight input IC multiplexer (IH6216) and three LM156 op-amps. Following the differential amplifier is the programmable gain stage discussed earlier, with a low pass filter on the output feeding the IH5110 sample and hold amplifier. The output of the IH5110 is connected to the comparator input (- input LM301) through the internal 10k feedback resistor of the 7541 multiplying D/A converter. The AD7541, along with a ±10 Volt reference and successive approximation logic, make up the 2's complement A/D converter.

A conversion is initiated by programming the multiplexer and programmable gain stage before strobing the 74123 dual one-shot. This can be accomplished by simply outputting one word to the port of the microprocessor which is responsible for the control of the DAS. The time delay created by the 74123 allows the front end of the DAS to settle before the strobing of the A/D converter; this time delay can also be implemented in software, thus eliminating the need for the 74123. As soon as the A/D converter is strobed, the conversion complete signal (busy signal) of the AM2504 SAR commands the IH5110 sample and hold amplifier to enter the HOLD mode and the actual conversion process begins. The microprocessor, after allowing for the set up time of the conversion complete signal (settling time incorporated by the 74123), should monitor the conversion complete signal for the end of conversion. This can also be monitored using the microprocessor's interrupt facilities. With the end of conversion, the digitized analog input signal is available for the microprocessor to input by tri-stating the outputs of the AM2504 SAR onto the microprocessor's bus.

MICROPROCESSOR INTERFACING

Figure 16 shows a method for interfacing the DAS to a microprocessor. There are three basic building blocks which make up the parallel interface. They are the data bus buffers, address decode logic and handshake and control circuitry. In order for the microprocessor to communicate effectively with the DAS, these three blocks are always needed, and may be implemented by merely using a single LSI parallel interface element (provided by each microprocessor manufacturer) or by using several MSI discrete logic packages. No matter which way is chosen, the interfacing is relatively easy, assuming there is some prior knowledge of the microprocessor to be interfaced. Figure 17 shows a block diagram of the DAS to 8080 CPU interface.

Data Bus Buffering

The characteristic which is responsible for the dynamic flow of data into and out of the microprocessor also requires that its data bus be buffered for both directions of data flow to and from external devices. This means that an external device must latch data from the bus at the appropriate time, and enable data to the bus at the appropriate time. The use of Three-State buffers for enabling data onto the bus, and D-type latches for removing data from the bus, permit this type of data transfer if their respective enable and clock control lines are activated at the right times by the handshake and control circuitry. Depending on the type of logic used in the microprocessor system and the number of loads on the bus, the logic family to be used in the DAS interface can be determined. Figure 18 shows the first part of the design of a discrete logic interface between the DAS and the Intel 8080 microprocessor. In the upper right hand corner of the figure are the two sets of octal three-state buffers and the octal latch used by the CPU to obtain communication with the DAS through the bidirectional data bus, DBØ-DB7. The CPU executes a data transfer by supplying the hand shake and control circuitry of the interface with the proper signals to allow the selected buffer or latch to be enabled.

Examples of three-state buffers include the TTL 80T9X and 74LS36X series and the CMOS 80C9X series; all have six buffers per package in different polarities and three-state configurations. Also available are 8-bit CMOS latch/buffers 74C373 and 74C374 which are very attractive for 8-bit CMOS applications, and the 74LS374 for heavier loads.

Address Decoding

In a control system environment, where many processes are being controlled using a single microprocessor as the controller, the requirements of the system are such that the microprocessor must communicate with several different I/O devices over the same data lines. When interfacing hardware to a data bus with this type of structure, the microprocessor must have the ability to enable, or "address", a specific I/O device for data transfer through the bus. This brings about the need for address decoding hardware at the interface itself. The type and quantity of circuitry used for this task depends primarily on the architecture of the overall system. Since the addressing capabilities of the

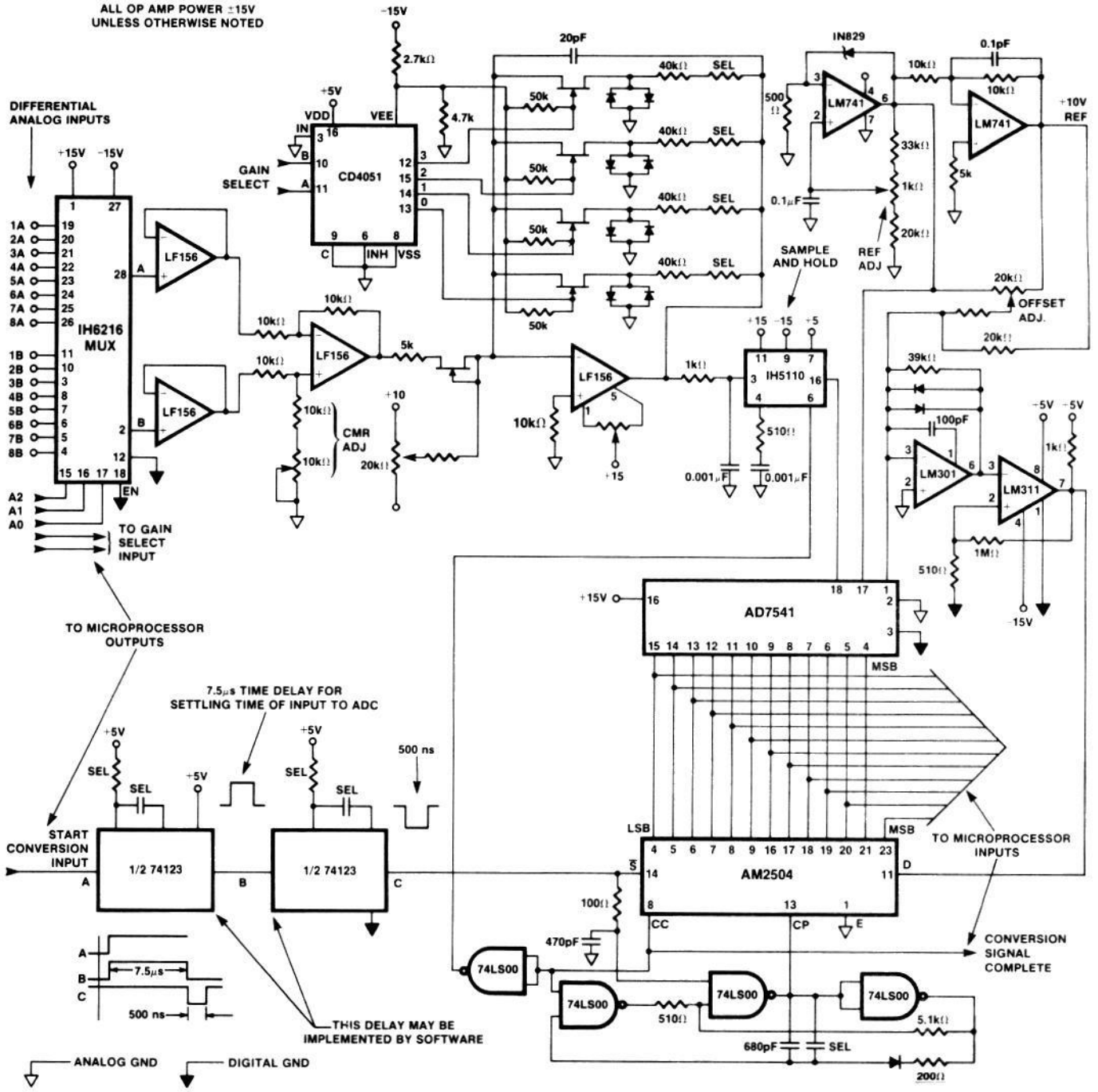


Figure 15: Complete A/D Conversion System.

microprocessor may overwhelmingly exceed the actual number of unique addresses used within a system, the amount of discrete logic necessary for address decoding at each interface may be kept to a minimum by initially allocating only the number of microprocessor address lines which are actually needed to satisfy the system's I/O requirements.

There are basically two techniques for I/O addressing: Programmed I/O and memory mapped I/O. Programmed I/O refers to specific instructions and microprocessor signals set aside by the manufacturer for the special purpose of communicating with I/O devices. The Intel 8080, for example, has an input and output instruction which uses the eight lower address lines (AØ-A7), together with two special read ($\overline{I/OR}$) or write ($\overline{I/OW}$) signals to control all I/O access to the CPU. Figure 19 illustrates this type of address decoding, which uses exclusive-OR logic to buffer the address lines into an eight input NAND gate. The exclusive-OR gates can be programmed for polarity (inverting or noninverting), thus allowing for jumper selectable addressing of the particular device onto the microprocessor data bus. It is important to note that the two signals which control the read ($\overline{I/OR}$) and write ($\overline{I/OW}$) operations from the microprocessor are active only during the execution of either the input or output instructions of the 8080 instruction set. Thus the memory addressing capabilities of the microprocessor are not affected since the memory read/write control is activated by two completely different control signals: Memory Read ($\overline{MEMR}$) and Memory Write ($\overline{MEMW}$). In memory mapped I/O, this is not the case. The same control lines used to control memory access, (again $\overline{MEMR}$) and $\overline{MEMW}$), are also used to control the I/O transfer, see Figure 20. The address decoding shown here is an extension of the previously described decoding of Figure 19, the only difference being the number of address lines decoded. The extra address lines are needed because the I/O address now looks like a memory address to the CPU, and with the normal microprocessor memory mapped in the lower addresses of the usable addressing range the I/O addresses are partitioned above the memory, starting at a point greater than the highest memory address.

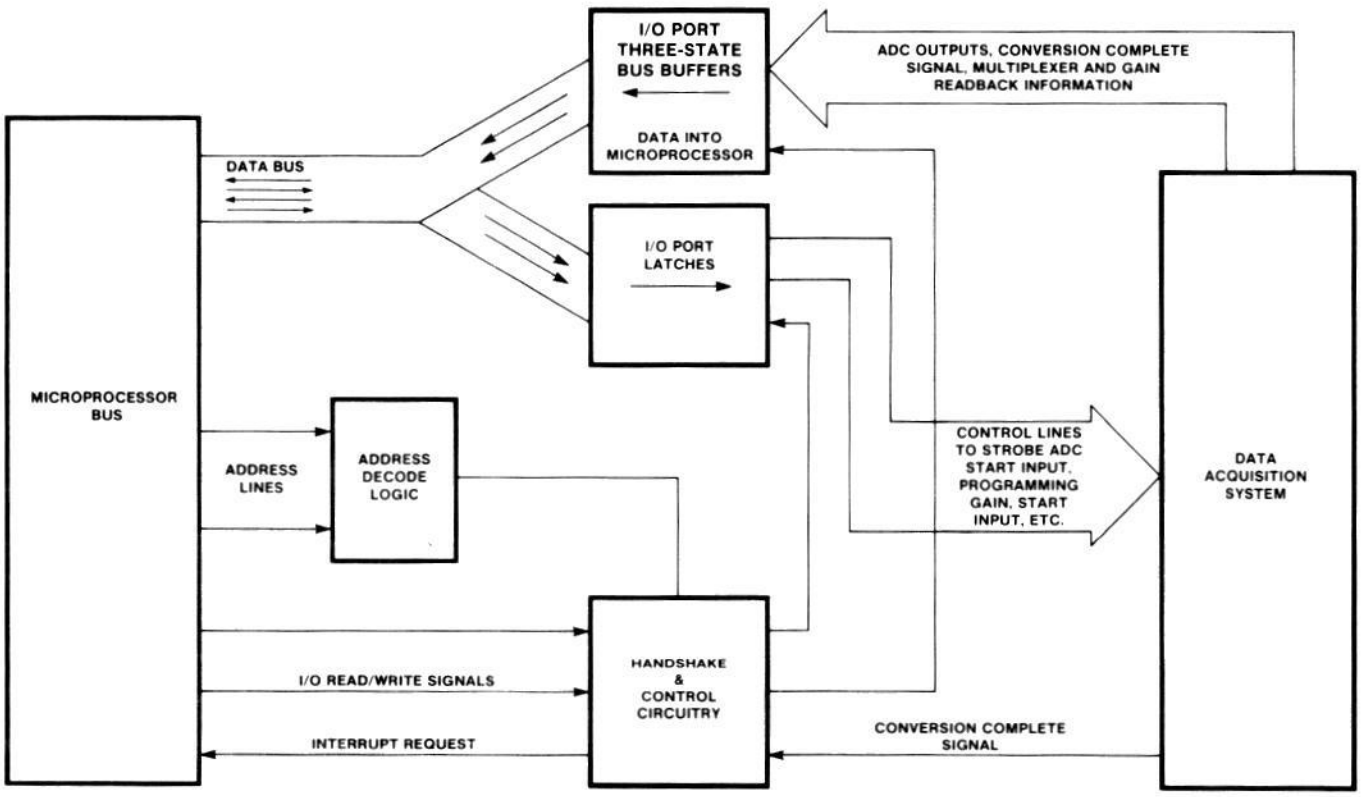


Figure 16: Block diagram multiprocessor to data acquisition system interface.

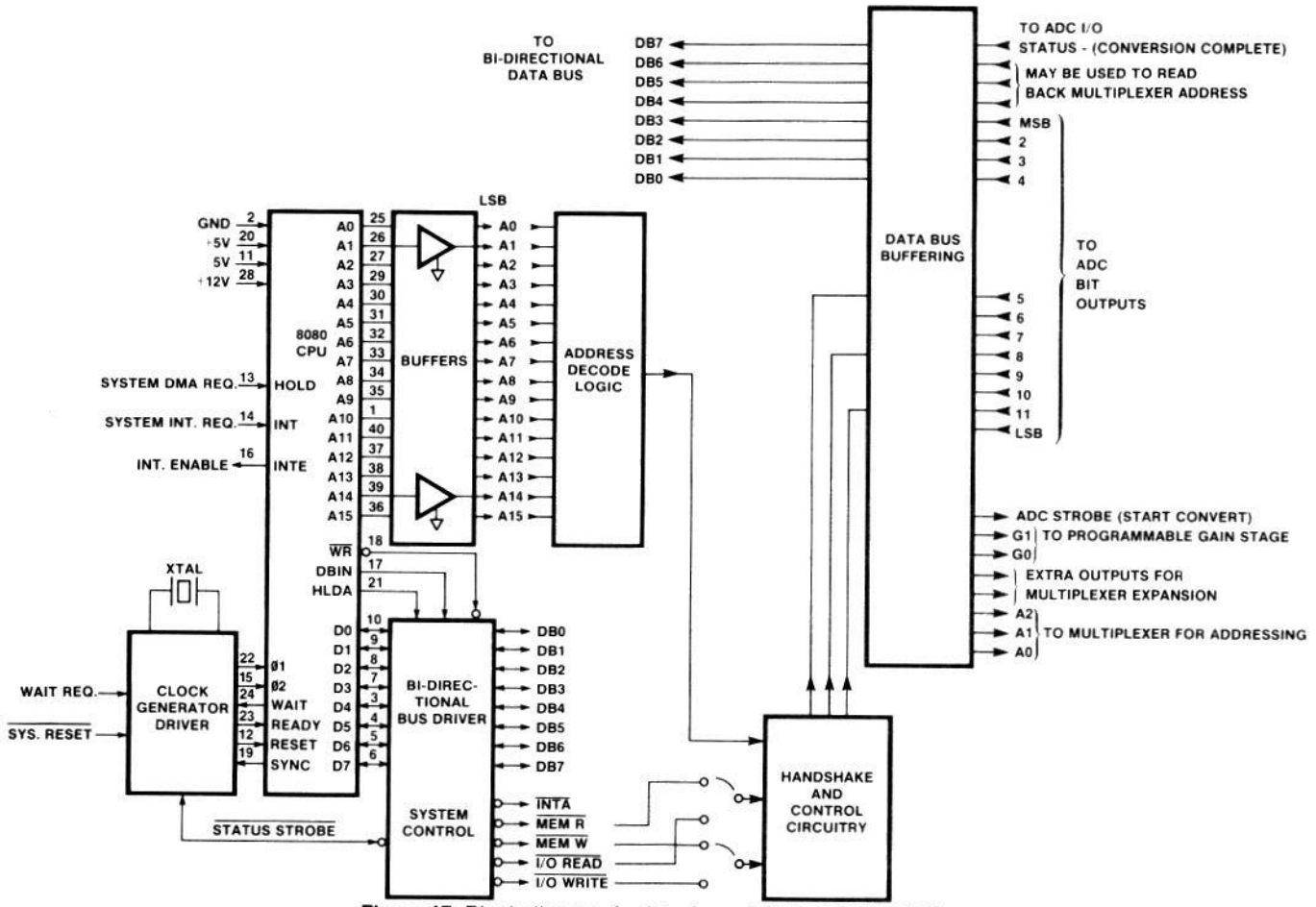


Figure 17: Block diagram for interface of DAS to 8080 CPU.

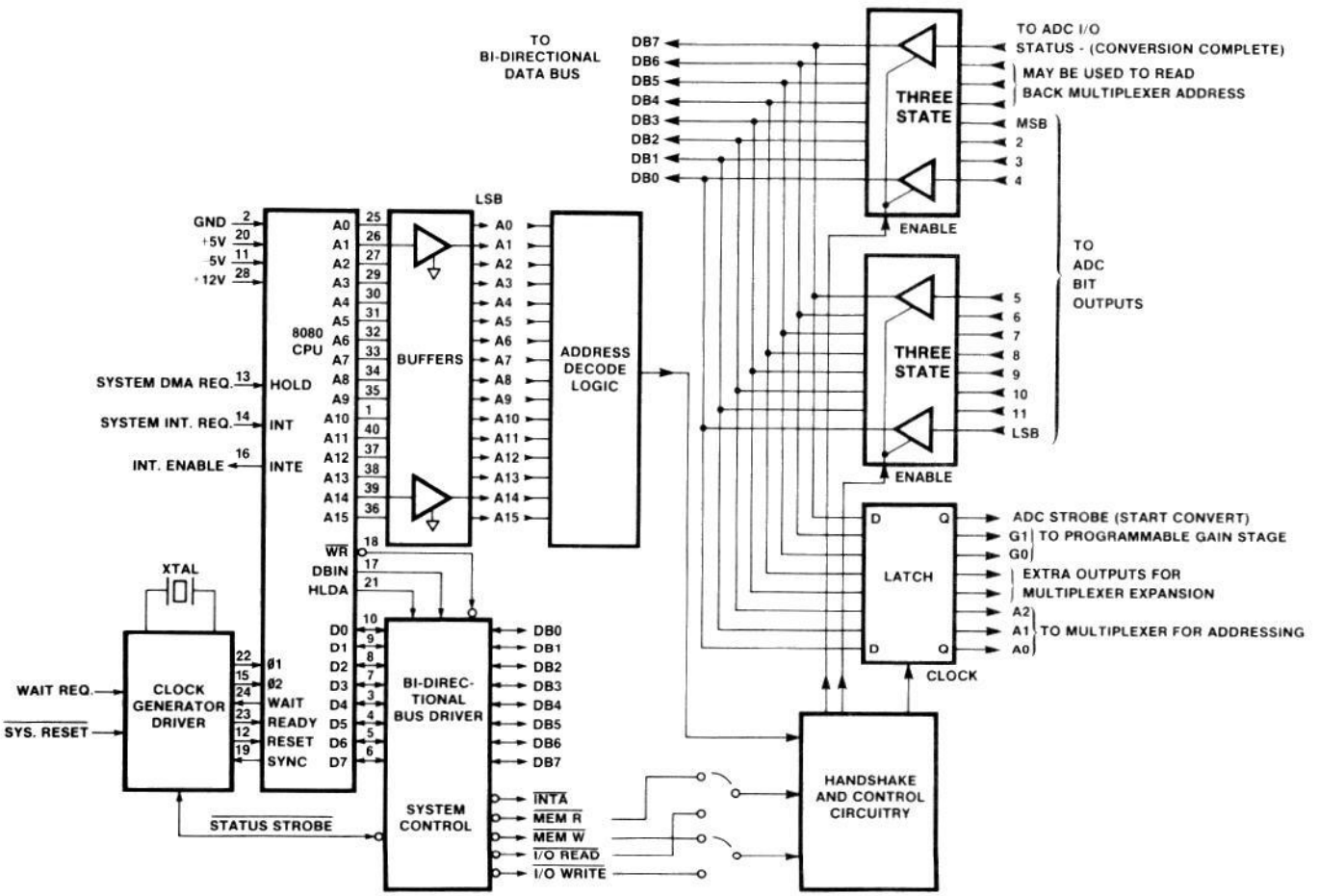


Figure 18: DAS to 8080 interface showing circuitry necessary for data bus buffering to 8080 bus.

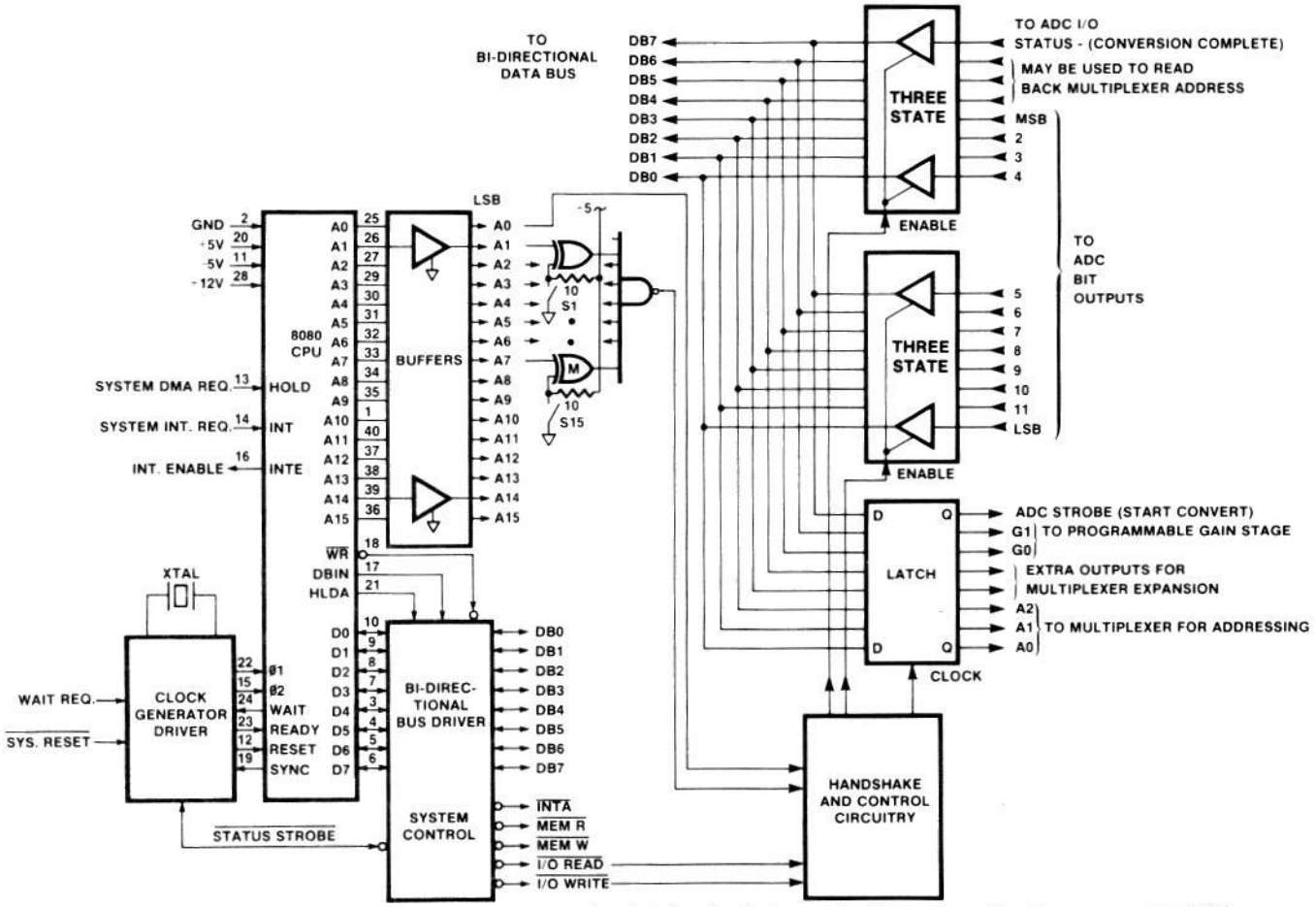


Figure 19: DAS to 8080 showing circuitry necessary for data bus buffering and address decoding for programmed I/O.

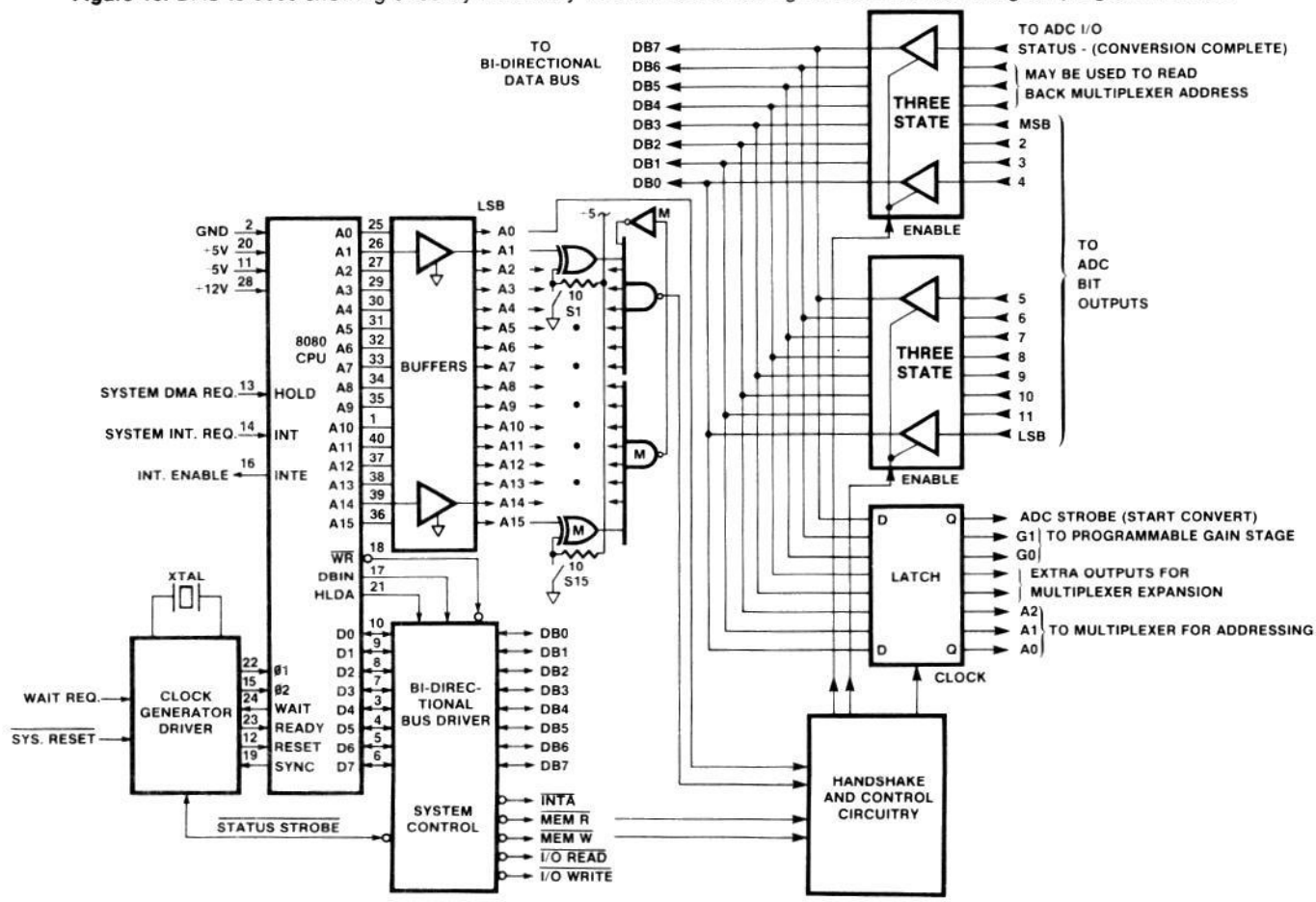


Figure 20: DAS to 8080 Interface showing circuitry necessary for data bus buffering and address decoding for memory mapped I/O.

The benefits of memory mapped I/O over programmed I/O are realized when considering the software required to access the I/O device under each configuration. When using programmed I/O there are only two instructions available to access the device, while memory mapped I/O makes available all of the memory reference instructions. With the 8080 CPU this means that two 8-bit ports, or both 8-bit words forming the ADC and control information may be read directly into the H & L registers using the single LHLD instruction. The significance of this feature is obvious: software I/O transfers can be made faster and more efficiently, and operations such as incrementing a multiplexer address can be accomplished directly with one instruction, rather than the three or four required in programmed I/O. The key to this efficiency is the fact that data to and from the I/O device does not have to pass through the accumulator when communicating with the I/O device, as it does in programmed I/O.

Handshake and Control Circuitry

The handshake and control circuitry is shown in Figures 21 and 22 for both the programmed I/O interface (Figure 21) and the memory mapped I/O interface (Figure 22). The circuitry required for the handshake and control block consists of only a few gates and is identical for either interface. NOR gates 1 and 2 are fed by the two read/write signals $\overline{I/OR}$ and $\overline{I/OW}$ for the programmed I/O interface, and $\overline{MEMR}$ and $\overline{MEMW}$ for the memory mapped I/O interface. They enable either a write into the latch, or a read from one of the three-state buffer groups, but only if the output of the address decode circuitry is enabled, or gate D's output is at a logic low level. During a read instruction, the selected interface, via the AØ address line, will select the three-state group controlled by gate 6 (if AØ = high level) to enable data onto the data bus. The use of the least significant address lines for control (in this case AØ) provides an easy way to select between multiple ports on the same interface.

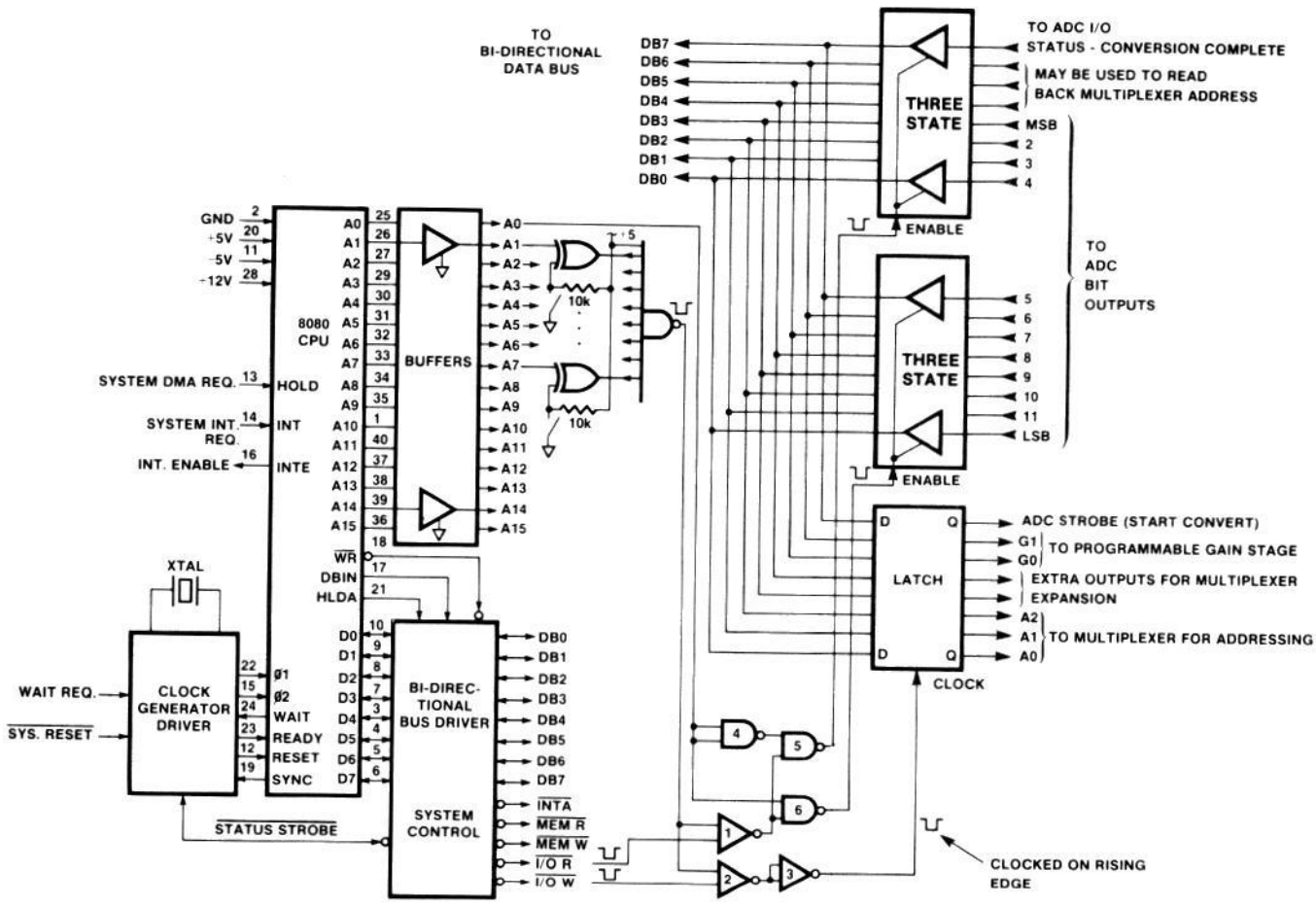


Figure 21: Block diagram for complete interface of DAS to 8080 using programmed I/O.

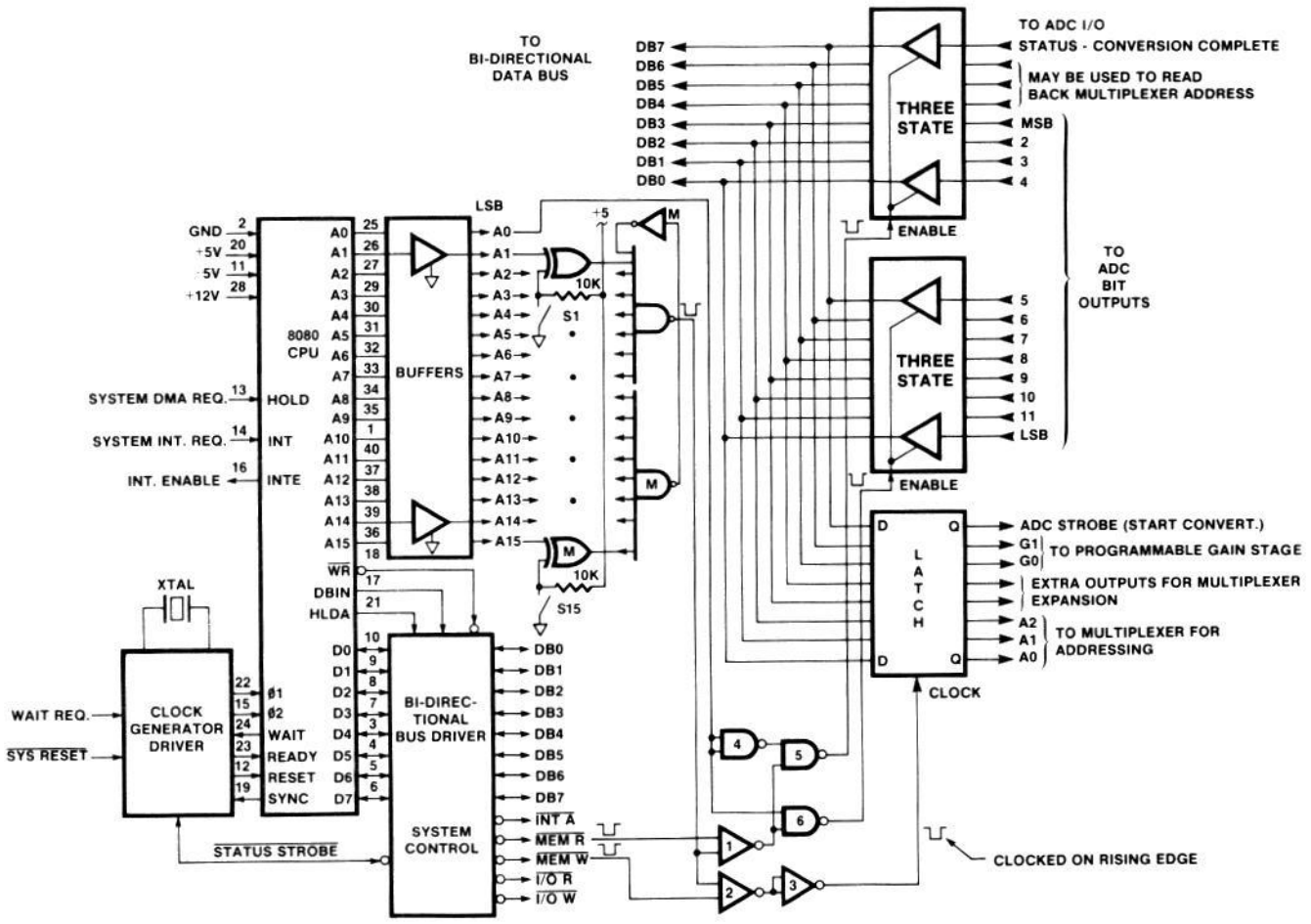


Figure 22: Block diagram for complete interface of DAS to 8080 using memory mapped I/O.

Interrupts

Figure 23 shows the previously described Programmed I/O Interface with the necessary circuitry for an interrupt driven interface. As shown, only one interface to the microprocessor can be interrupt driven in a system that may have several I/O devices processing data. Figure 23 shows the Q output of the interrupt latch being tied directly to the interrupt request input (INT) of the 8080. As soon as the STATUS signal makes the transition to a logic one (positive transition), signifying an end of conversion of the DAS's ADC, the Q output of the interrupt latch is clocked high. This immediately generates an interrupt request at the 8080 INT input, and assuming that the microprocessor strobed the ADC some time ago and is now currently executing an instruction for some other routine in memory, program execution will jump to the interrupt service routine for the interrupting device at the end of the current instruction execution.

Usually the interrupt service routine is set up to service a device when the CPU cannot afford to spend the time monitoring the device's data ready flags. The service routine for the interface in Figure 23 should include reading the status information and most significant ADC MSBs via gate 5 and will thus clear the interrupt request to the 8080 before the interrupt service routine is over. This is important because if the interrupt request is not removed before a return to normal program execution occurs, the interrupt service routine will be immediately re-entered.

When configuring a system with several interrupt driven interfaces, there is a need for additional hardware between the interrupt latch output of the interface and the interrupt request input of the 8080. The multiple interrupt hardware is generally never located on a single device interface, and is beyond the scope of this application note.

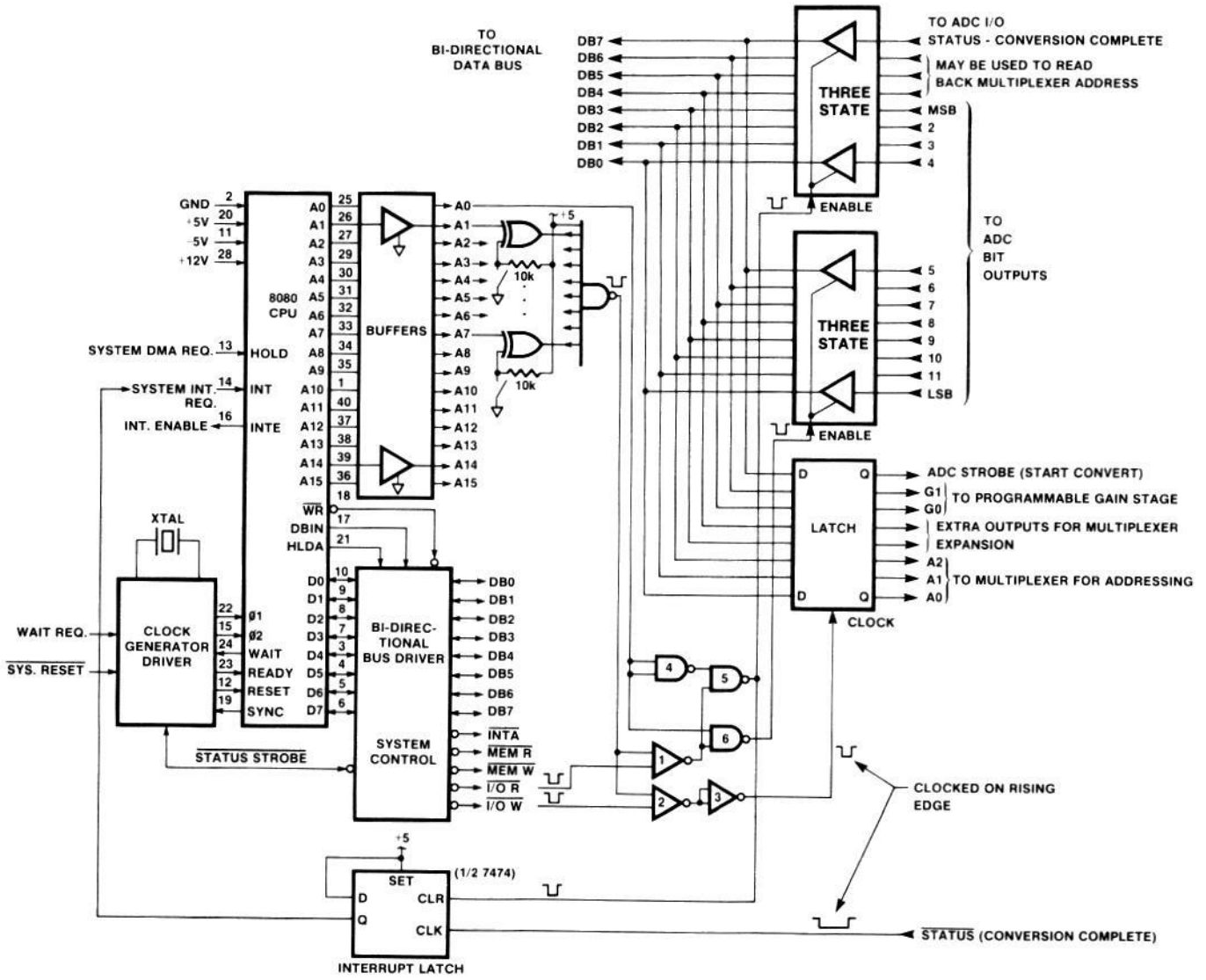


Figure 23: Block diagram for complete interface of DAS to 8080 using programmed I/O and interrupt driven.

INTERFACE TO THE SBC-80/10 BUS

Figure 24 shows a complete interface for a DAS using the 8255 parallel interface element, which will plug directly into the SBC 80/10 bus. The 8255, when used in mode Ø (See Intel "Peripheral Design Handbook" for a detailed description of the 8255's operating modes), will replace the three-state buffers, latch and handshake and control circuitry of Figure 23, and provide a simplified interface. When using the 8255, or any other parallel interface element to directly drive a data bus, care must be taken not to overload the three-state drive capabilities of its outputs, otherwise the interface will not drive the bus and improper transfer of data will occur. For a heavily loaded bus, which can be considered more than three TTL loads when connecting the data lines of the 8255 directly to the bus, the 8255 must itself be buffered by three-state buffers. The enable lines for these buffers would then be driven by the same read/write signals which enter the 8255 at pins 5 and 36 respectively. By initializing the 8255 with a control word of 231_8 in mode Ø, the I/O pins will be programmed as shown in Figure 24, with PBØ-PB7 as outputs and all others as inputs.

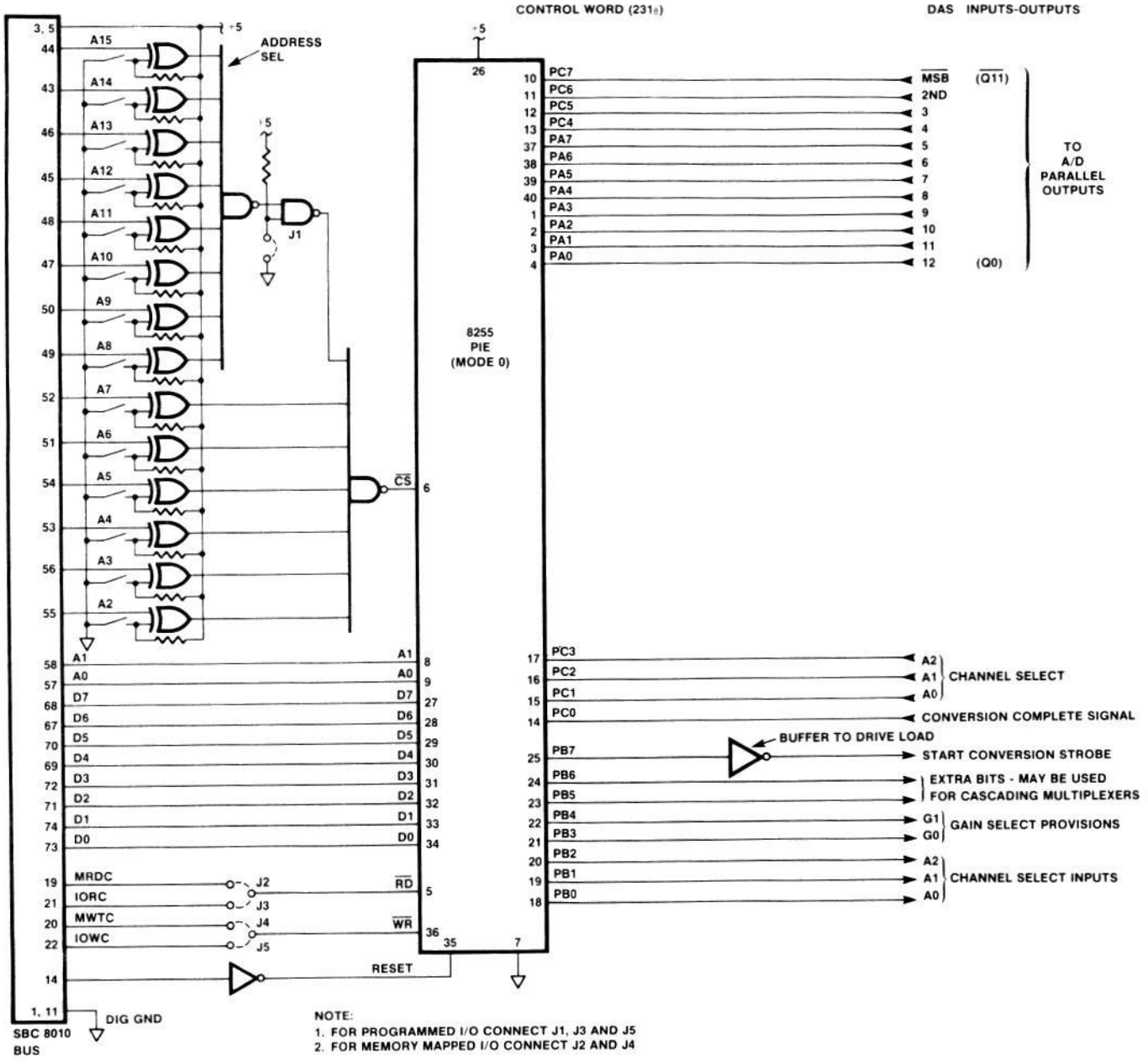


Figure 24: Interface to SBC 80/10 bus.

INTERFACING TO THE INTERSIL INTERCEPT II

The Intercept II is a stand alone minicomputer, which uses the 6100 CMOS microprocessor and emulates the DEC PDP8-E instruction set. Even though the basic LSI elements are CMOS devices, the computer operates over a low power schottky TTL bus. This is an advantage to the computer as far as speed and the amount of bus loading are concerned, although it increases the complexity for the user when interfcing the CMOS peripheral devices (eg. - PARALLEL INTERFACE ELEMENT) to the TTL bus. This is evident in Figure 25 with the DAS to Intercept II interface requiring buffering of all signals between the TTL bus and the CMOS parallel interface element. In a small all CMOS system, much less buffering is required.

To understand the workings of this interface it is necessary to know a little bit about the IM6100 microcomputer and the IM6101 parallel interface element. First of all the system is a bi-directional system, with both address and data transfers over the same 12-bit bus. Addresses are not stable during the entire instruction cycle, and must be latched into the I/O device on the falling edge of "LXMAR". The signals "Device Select" and "XTC" provide the means for transferring data to and from the CPU, while "C1" and "C2" aid in the accumulator manipulations once the data is transmitted or received to or from the I/O device. "PRIN", "PROUT", "INTGNT" and "SKP/INT" along with the IM6101 supply the necessary controls to provide a superior priority based interrupt system with virtually no limit to the number of devices in the chain.

The parallel interface element (IM6101) is itself actually an I/O controller, not an I/O device, even though it has 8 bits of I/O. Not only does the device have on board address decoding, it also has the ability to control several I/O ports by merely adding on the three-state buffers and latches required for each port. Like the 8255, the IM6101 must be software initialized, and part of the initialization includes programming the four sense inputs to detect any combination of level or edge inputs which may be applied. Referring to Figure 25, the falling edge of the conversion complete signal from the ADC of the DAS can be sensed directly by the parallel interface element as opposed to the need of providing an external flip-flop (as in Figure 23) when operating in an interrupt driven mode. Another nice feature is the ability to use the write signal to strobe the ADC. This allows a start of conversion to the ADC at the same time the multiplexer is being programmed, and is a lot faster than a software generated pulse. The time saved in strobing the ADC in this fashion is appreciable when the DAS is desired to operate at high throughput rates.

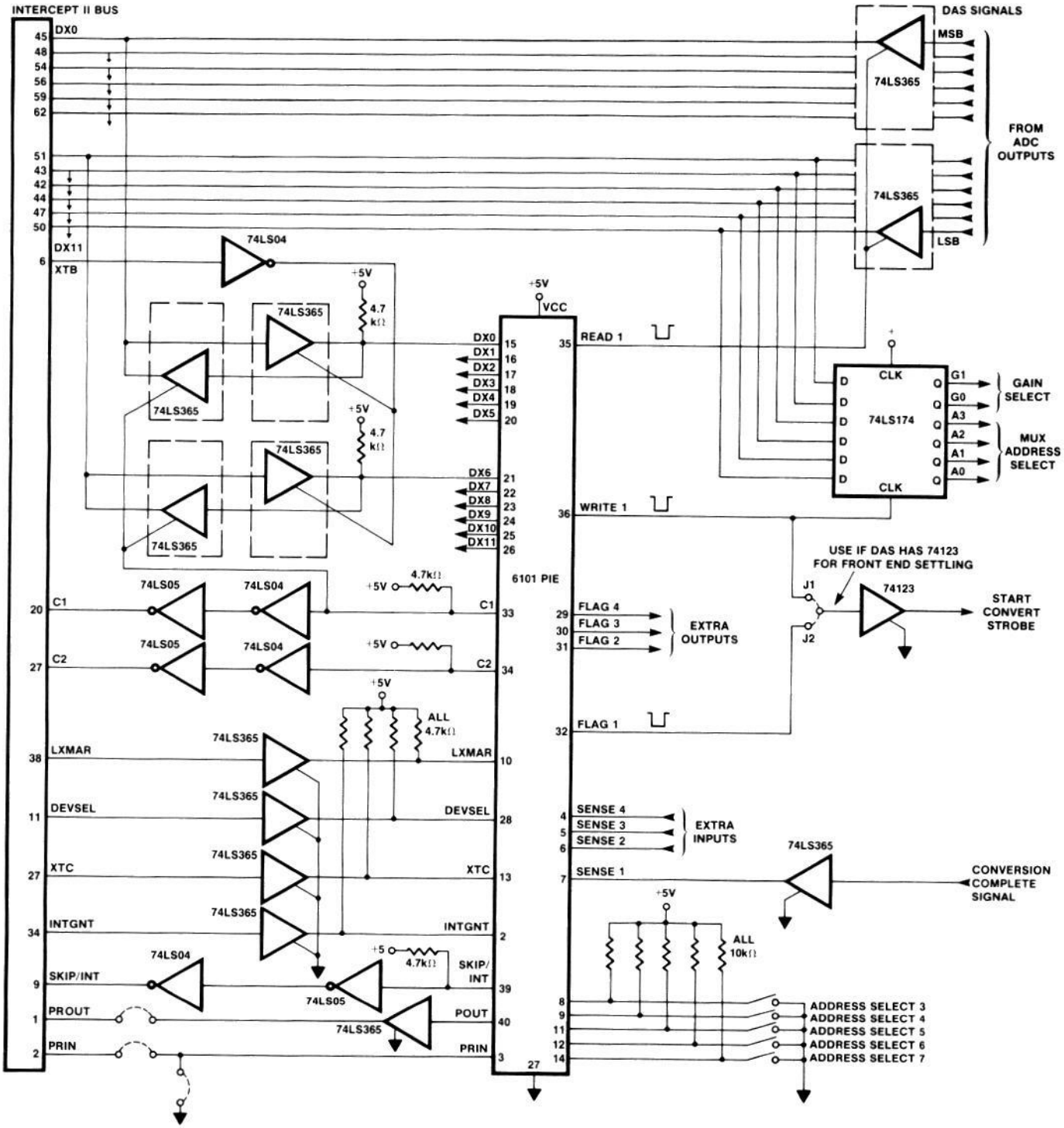


Figure 25: Interface to Intercept II computer.

SUMMARY

Although a wide variety of circuits have been presented, the actual requirements for a specific application will dictate the selections which must be made from them. Again, an accurate 12 bit system has been covered at every stage, and would be adequate for any less accurate system. Some improvements in speed or economy could be achieved by the redesign of the building blocks under the guidelines explored here, but at the cost of lowered accuracy. Increases in accuracy, correspondingly, would necessitate redesign, or at least reformulation, of most of the blocks as required.

Nevertheless, it should be clear that the cookbook approach, in a building block orientation as presented here, is capable of achieving an altogether satisfactory design of a high speed data acquisition and microprocessor interface system. And for those who hate cooking, many of the more complex blocks are available as modules(generally designed as described here, and using the same components),simplifying the process considerably. For the ultimate in "table service", complete systems can be purchased, including all the elements described here. But that is outside the purview of a cookbook.

Some other applications bulletins that may be found useful are listed here:

A003 "Understanding and Applying the Analog Switch", by Dave Fullagar.

A004 "The IH5009 Series of Low Cost Analog Switches".

A006 "A New CMOS Analog Gate Technology", by Dave Fullagar.

A016 "Selecting A/D Converters", by Dave Fullagar.

A017 "The Integrating A/D Converter", by Lee Evans.

A018 "Do's and Dont's of Applying A/D Converters", by Peter Bradshaw and Skip Osgood.

A019 "4-1/2 Digit Panel Meter Demonstrator/Instrumentation Boards", by Michael Dufort.

A023 "Low Cost Digital Panel Meter Designs", by David Fullagar & Michael Dufort.

A025 "Building a Remote Data Logging Station", by Peter Bradshaw.

A028 "Building an Autoranging DMM with the ICL7103A/8052A A/D Converter Pair", by Larry Goff.

A030 "The ICL7104 - A Binary Output A/D Converter for Microprocessors", by Peter Bradshaw.

R005 "Interfacing Data Converters & Microprocessors", by Peter Bradshaw et al, Electronics, Dec. 9, 1976.

R009 "Reduce CMOS Multiplexer Troubles Through Proper Device Selection", by Dick Wilenken.

R011 "Switching Signals with Semiconductors", by Paresh Manair.

Most of these are available in the Intersil Data Acquisition Handbook, together with other material.

INTERSIL

10710 N. Tantau Avenue, Cupertino, CA 95014 U.S.A., Tel: (408) 996-5000, TWX: 910-338-0171

9th Floor, Snamprogetti House, Basing View, Basingstoke, RG21 2YS, Hampshire, England, Tel: 0256-57361, TLX: 847227 INTRSL G

Liaison Office, 217, Bureaux de la Colline, de St. Cloud, Batiment D, 92213 Saint-Cloud, Cedex, France, Tel: (1) 602-57-11, TLX: Datelem 204280F

Bavariaring 8, 8000 Munchen 2, West Germany, Tel: 89/539271, TLX: 5215736 INSL D