LESSON 4 – COMPUTER INPUT/OUTPUT SUBSYSTEMS

- This lecture introduces some details on ***computer subsystems devices*** and their operation and leads to the general ***concept of control program logic***. This information is intended to supplement the instrumentation & controls lectures with additional information on the input & output subsystems for digital control with implementation details.

Lecture Topics

1. Computer Subsystem Overview

2. The Digital Input (DI)

3. The Digital Output (DO)

4. Digital On/Off Control Example

5. Pseudo- Code for the On/Off Level Control Application

6. Analog to Digital Conversion (ADC)

7. Digital to Analog Conversion (DAC)

8. Stylized Digital Control System

9. Basics of a typical Control Program

1. COMPUTER SUBSYSTEM OVERVIEW

- In order for a computer system *to succesfully apply control logic* to a process system, the plant ***status must be recognized*** by the computer and some means must be provided to ***apply the computer control decision to the plant manipulated variables***.
- The means of ***obtaining plant data*** and converting that information into a ***digital format*** that is recognizeable and useable by the computer is the role of ***a computer input subsytem***.
- The means of outputing the digital control decision in ***an analog format*** that is useable by the field devices is the role of a computer output subsystem.

2. The Digital Input (DI)

- A simple input for a computer would be an ***on/off digital*** plant field value which can apply a ***high voltage level*** (say above 4.8 V dc) if the circuit is ***energized*** or a ***low voltage level*** (say below 0.6 V dc) if the plant field circuit is ***de-energized***.
- This voltage can be sensed by special input subsystem circuitry to signal the computer with a numerical ***one state*** when the plant circuit is ***energized*** or a numerical ***zero state*** when the plant circuit is ***not energized***.
- This input circuitry is referred to as a ***digital input*** or ***DI*** for the computer system.
- In this way, a plant field device such as an electrical pressure switch or flow switch circuit could be connected to a computer.
- The ***computer would be able to recognize*** when the process parameter was too high (for example, when the pressure is too high, the pressure switch is tripped, completing the 5 volt circuit allowing the computer to sense ***the energized*** or ***1 state***) or conversely when the process parameter was too low (the pressure switch is not tripped, the 5 volt circuit is opened and the computer senses the ***de-energize*** or ***0 state***).
- You could visualize 16 such plant field driven circuits being connected to a 16-Bit input device so that a ***unique bit position*** corresponds to an ***exact pressure switch*** circuit.
- Then by reading the status of the 16 bit word, the computer program could manipulate the data to recognize which pressure switches were energized and which were not.

```
 0001   0110  1111  0011 - 16 Bit Word Status
FEDC  BA98  7654   3210 - Hexadecimal Bit Position
```

Figure 1. A 16 bit status word reflecting Pressure Switch Input status

- In Figure 1., nine (9) Pressure Switch Circuits numbered 0, 1, 4, 5, 6 ,7, 9, A and C have the ***status of 1*** indicating that the pressure switch circuit is ***energized***.
- Once this data state has been determined, the computer logic can initiate ***annunciation*** messages (with time, process parameter, system identification, operating manual references, etc), change ***display*** information, initiate corrective ***contro***l actions (like starting a pump or opening a valve) and update the central plant operating ***database*** for future reference for maintenance or operations purposes.

3. The Digital Output (DO)

- A ***simple output*** from a computer would be an ***on/off digital*** plant field signal value which can apply a ***high voltage output level*** (say above 4.8 V dc) if the circuit is ***to be energized*** and a ***low voltage output level*** (say below 0.6 V dc) if the plant field circuit is ***to be de-energized***.
- This voltage can be output by special circuitry to apply the binary computer logic to the field with the 5 V dc value corresponding to the ***one logic state*** and the 0 V dc value (or at least less than 1 V dc) corresponding to the ***zero logic state.***
- This is referred to as a ***digital output*** or ***DO*** for the computer system.
- In this way, an ***electric solenoid valve*** (for example) could be connected via interfacing circuits to the computer DO. The electrically actuated solenoid valve could admit or vent a pneumatuc signal to a larger pneumatically actuated control valve.
- The computer would be able ***energize*** or ***de-energize*** the field mounted solenoid valve as a function of the logic state developed by a control algorithm and then output the resultant logic state to that particular bit of an output register.
- You could visualize 16 such computer driven plant field circuits being connected to a 16-Bit register output circuitry device so that a ***unique bit position*** corresponds to an ***exact solenoid valve*** circuit.
- Then by writing to that bit of the register to change the status of the 16 bit word, the computer subsystem could manipulate the status of the field solenoid valves.

1010 0011 1100 1111 - 16 Bit Word Status
FEDC BA98 7654 3210 - Hexadecimal Bit Position

Figure 2. A 16 bit status word reflecting Solenoid Valve Output Signals status

- In Figure 2., ten (10) of sixteen Solenoid Valve Circuits numbered 0, 1, 2, 3, 6 ,7, 8, 9, D and F have the status of 1 indicating that the solenoid valve should be energized.
- The remaining six (6) solenoid valve circuits numbered 4, 5, A, B, C and E have the status of 0 and the solenoid valves would be de-energized.
- Once this logically determined data state has been applied to the field to ***change the solenoid valve position***, the computer will have initiated a practical on/off control action.
- The solenoid valves could be used to apply or remove pneumatic or hydraulic signals to implement control actions by opening or closing valves, positioner dampers or guides and so forth.

4. Digital On/Off Control Example

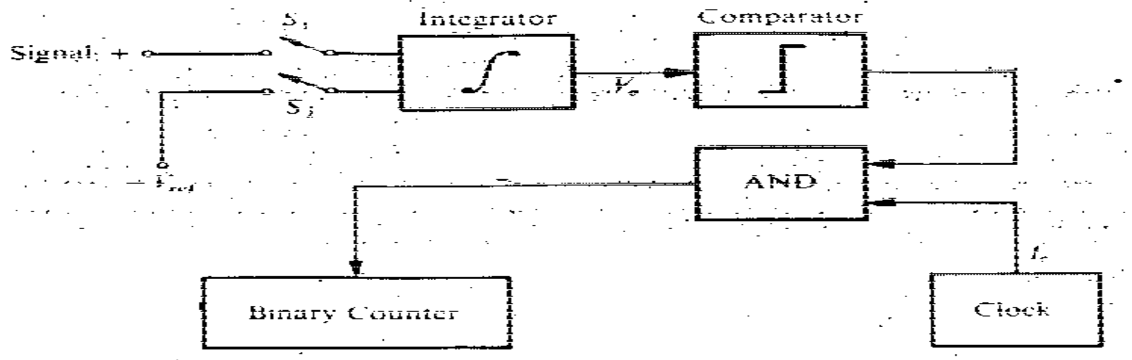
- If we have a plant/computer configuration where the on/off status of a plant field condition can be determined by ***reading a digital input*** and the corresponding status of a final device can be set by ***writing to a digital output***; then a functional ***on/off control*** strategy can be implemented by this computer system.
- For example, assume that the ***level*** of an open tank is to be regulated by on/off control of the ***inflow*** valve.
- Assume the inflow valve is an ***air-to-open*** style and that a ***three way solenoid valve*** (supply, vent, actuator) is connected into the pneumatic actuator supply circuit
- The pneumatic actuator signal pressure can be applied if the ***solenoid valve is energized*** - allowing the ***inflow valve to open***
- Or the actuator for the inflow valve can be vented if the ***solenoid valve is de-energized*** - allowing the ***inflow control valve to close.***
- Similarly, a level switch (S1 - near the top of the tank) for the tank level can be read by a digital input for this computer.
- As long as the tank level is ***below the level switch threshold***, the input to the DI will be zero and so the inflow valve can be held open (i.e. the ***DO is energized***).
- Once the level rises above the level switch threshold value, the DI will sense the 1 state and then the computer can de-energize the DO to close the inflow valve and let the level start dropping back toward the level switch threshold.
- A second level switch (S2 - lower position in the tank) could be used to provide a wider initiation tolerance so that the inflow valve is not repeatedly snapped open and closed on a high frequency basis.
- In this manner, the ***on/off control logic*** rules would require that the inflow valve (actually inflow solenoid valve - DO) be energized if the level drops below the S2 (say DI-2nd bit) position and should remain energized until the level rises above the S1 (say DI-1st bit) positon (near the top of the tank).
- Once S1 position has been ***exceeded***, the DO should be de-energized and remain de-energized until the level drops below the S2 position.
- In this manner, the computer can monitor the tank point level indications by ***reading*** S1 and S2 via the ***DI's*** and then make a control logic decision to ***drive the inflow valve position*** via the ***DO*** state.

<u>5. Pseudo- Code for the On/Off Level Control Application</u>

```
*** DOF = digital output flag,  DIF1 = Digital Input Bit 1 Flag
*** LDOF = last iteration DOF value
*** DIF2 = Digital Input Bit 2 Flag

*** clear the DOF - set to de-energize
DOF=0
*** check the field status - Read the DI's
Read (DIF1, DIF2)
** Switch S1=1 if the level is above the high tank level mark (DIF1 =1)
** Switch S2=1  if the level is above the low tank level mark (DIF2=1 )

** set level flags
*** check if the tank was filling ***
** if so, keep the in-flow valve open until the level rises to the top switch
IF(LDOF.EQ.1.AND.DIF1.EQ.0) DOF=1

*** check if the tank was emptying **
*** if so keep the in-flow valve closed until the level drops to the lower switch
IF(LDOF.EQ.0.AND.DIF2.EQ.1) DOF=0

*** if the tank level is high - de-energize the Solenoid valve
IF (DIF1.EQ.1) DOF=0
*** if the tank level is too low, energize the solenoid valve
IF (DIF2 .EQ.0) DOF=1

*** output the control flag - drive the DO
Write (DOF)
*** update the old flag
LDOF = DOF
```

6. Analog to Digital Conversion (ADC)

- The plant field analog current signal (4-20 mA) representing a measured process variable must be converted to a recognizeable value for the digital computer. The incoming analog current signal must be ***digitized*** or converted to a digital value that can be accepted and used by the computer.
- An ***analog to digital convertor*** or ADC performs this function of converting the analog signal to a ***digital count***.
- One style of ADC, the ***dual ramp convertor*** is a good example to consider to understand the basic operation and purpose of an ADC.
- For discussion purposes, assume that this ***ADC*** will develop a ***digital output*** ranging from 20-100 counts as the analog signal changes from 1-5 volts as a result of applying the 4-20 mA signal current across a dropping resistor.
- This simple model is presented to demonstrate the concept of an ADC function, but not to detail a specific ADC operation. Once the digital count value is developed, the computer can recogize the ***actual value*** of the plant field signal and use this digitized value for control decision purposes.



igure 3. A components of a dual ramp ADC

6. Analog to Digital Conversion (ADC).....continued

- The ***dual ramp ADC*** consists of an ***integrator***, a ***comparator***, and a ***counter register***.
- Inititially the counter is ***cleared to zero*** by applying a reset signal so the register content is zero.
- A sampling switch, S1, ***connects the process value signal voltage*** to the integrator is closed to start the conversion process.
- The signal voltage can be considered ***constant*** for the short sample time by the ADC - therefore the integrator output will rise as a ***time function*** of the applied voltage signal input.
- The ***slope*** of the integration curve is proportional to the ***magnitude of the signal***.
- The ***integrator output*** is applied to a comparator and as long as the integrator output is ***positive***, the comparator output will be high and so the AND gate will pass the clock pulses to the register.
- This means that the ***register will count the clock pulses*** as long as the ***integrator signal is a positive value***.
- When the counter reaches a ***preset maximum value***, the ***counter is reset*** and the switch S1 is opened while switch S2 is closed to apply the ***negative reference signal*** to the integrator (at this point the integrator output is (+Vsig * Integration Time).

6. Analog to Digital Conversion (ADC).....continued

- Closing switch S2 applies the ***constant negative*** reference voltage (-Vref) to the integrator so that the integrator output begins to ***ramp down at a constant rate***.

- As long as the integrator output is greater than zero, the comparator output will be positive so that the AND gate will allow the clock pulses to be totalized in the counter.

- When the integrator reaches zero value, the ***comparator output changes state*** (now not 1) and so the ***AND gate does not pass anymore clock pulses***.

- The time required to ramp the integrator down to zero is proportional to the ***original signal value*** which is now ***represented by the number in the counter register***.

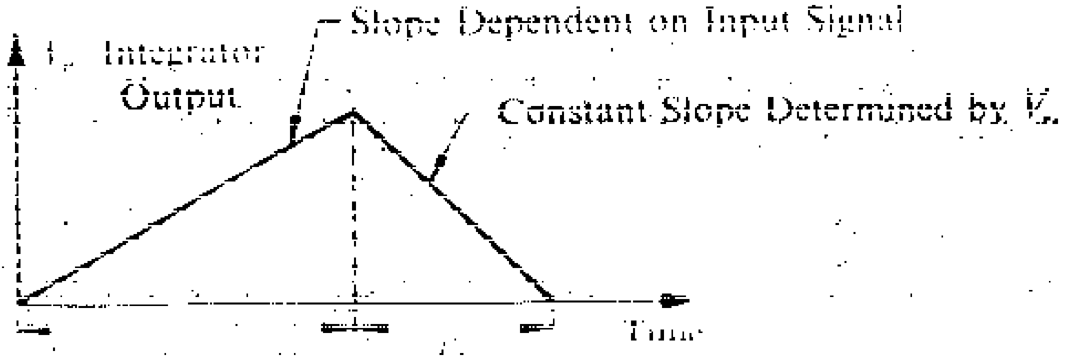


Figure 4. The Integrating Slopes for the Dual Slope ADC

6. Analog to Digital Conversion (ADC).....continued

- For example - assume that the original input signal is at 32.5% (***2.3 V dc***) and the counter will overflow in ***1000 microseconds*** (assume the clock pulses every microsecond).

- The integrator output would rise to 2.3 x 1000 = ***2300 units*** (some scaled value).

- Now if the negative reference voltage applied is -50 units then the integrator will ramp down to zero in 46 microseconds (i.e. 2300/50 = 46) - so the number in the counter will be 46 at the end of the conversion (***i.e twenty times times the analog voltage value of 2.3 - a scaled value***).

- For a second example, assume that the process signal is 10.0% (***1.4 V dc***).

- The integrator will ramp up with a lower slope value so that after 1000 microseconds, the integrator output will be (1.4 x 1000) ***1400 units***.

- Now the counter is reset and the reference negative voltage is applied at -50 units so that the integrator counts down to zero in 28 microseconds.

At the end of the conversion, the counter register contains the number 28 (***i.e. twenty times the analog signal value of 1.4 V dc - a scaled value***). In this way, the ADC would provide a digital count of 20 corresponding to the 1 volt signal and a digital count of 100 corresponding to the 5 volt signal. Now obviously, this sort of resolution would not be adequate for control purposes - but it is suitable for explanation purposes. We would usually require 10 (1 in 1024) or 11 (1 in 2048) bit resolution for a control ADC application.

Multiple Channel Input ADC - Mutliplexer

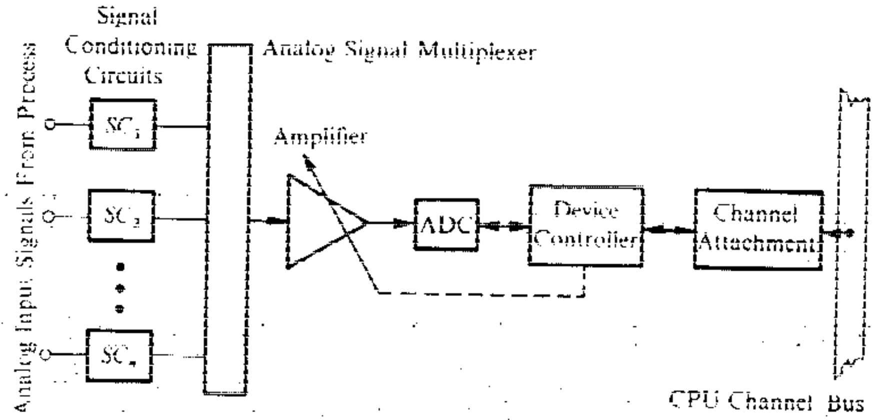

Signal Conditioning Circuits
Analog Signal Multiplexer
Analog Input Signals From Process
SC1
SC2
SCn
Amplifier
ADC
Device Controller
Channel Attachment
CPU Channel Bus

Figure 5. General Analog Input Multiplexing

- Multiple signals can be routed into ***one common ADC*** so that the conversion equipment needs can be reduced.
- It is common practice to have 16 signal channels connected to one ADC via a multiplexer to be converted sequentially one after the other.
- This configuration can be thought of as a ***dynamic switching circuit*** which will select a signal circuit and then connect that signal to a conditioning amplifier.
- Once the signal has been prepared, the amplifier output will be connected to the ADC for ***conversion*** and when the conversion is complete, that ***value can be stored*** in an indexed data table before ***selecting the next signal*** for conversion.
- The process signal channel would be selected by a ***sample and convert*** addressing instruction to connect the field signal to a ***sample and hold*** (***S&H***) amplifier.The field analog signal value will charge a capacitance input circuit in the sample & hold amplifier.
- Then the field signal is ***disconnected*** from the amplifier before the ***equivalent*** charged sample & hold amplifier value is connected to the ***analog to digital converter*** (in this way the ADC and computer are always ***protected*** from the possibility of field generated faults).
- A ***convert instruction*** is now initiated so that the ***digital value*** of the equivalent signal from the sample and hold amplifier is prepared by the ADC and stored in dynamic memory.

Multiple Channel Input ADC - Mutliplexer....continued

- You can see how this process lends itself well to multiple instructions since only an ***address pointer*** needs to be updated to select a new analog signal to the sample & hold amplifier and the ***same pointer offset*** can be used to address the storage of the converted value in an indexed array.
- The digital control system can now ***monitor several connected field analog loops***, one after the other, as the signals are ***sequentially fed*** to the computer via the multiplexor and the ADC.
- Depending upon the time response of the process being measured, the input subsystem may ***sample on a time interval*** ranging from 100 milliseconds (quite a fast process) to 2000 milliseconds (quite a slow process).

7. Digital to Analog Conversion (DAC)

- Once the control logic decision has been made by the computer, the control signal must be ***output to the field*** in an analog format.
- The ***digital to analog converter*** or ***DAC*** will accept the ***digital control word*** value and convert this to a viable 4-20 mA current suitable for operating standard plant current driven final actuators or transducers.

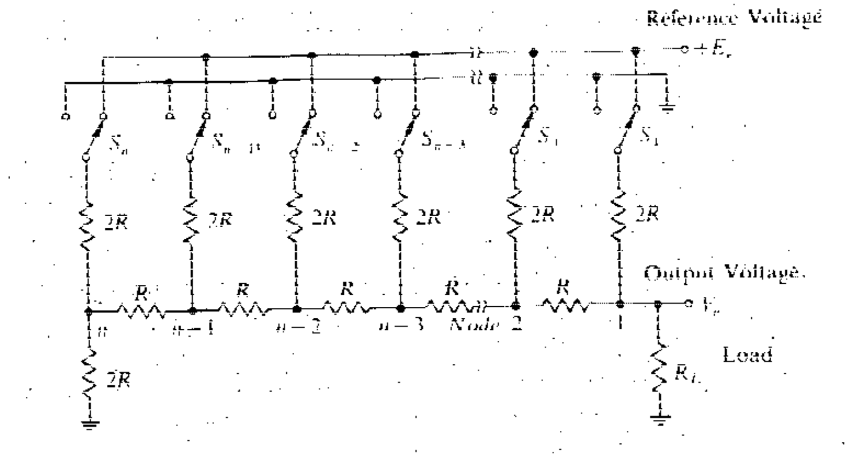


Figure 6. The R-2R Voltage Divider Ladder Network

- The ***digital to analog converter*** is based upon switching a resistance network in what is called an ***R-2R configuration*** because of the relative values of adjacent resistance components (i.e. resistances of 2R in parallel are separated by a 1R resistance in series) .
- The output voltage is developed by ***switching resistances*** into a precision reference voltage supply.

The R-2R DAC Function

- The switches for this resistance network are ***controlled by the bits*** in the digital control word that is to be converted.

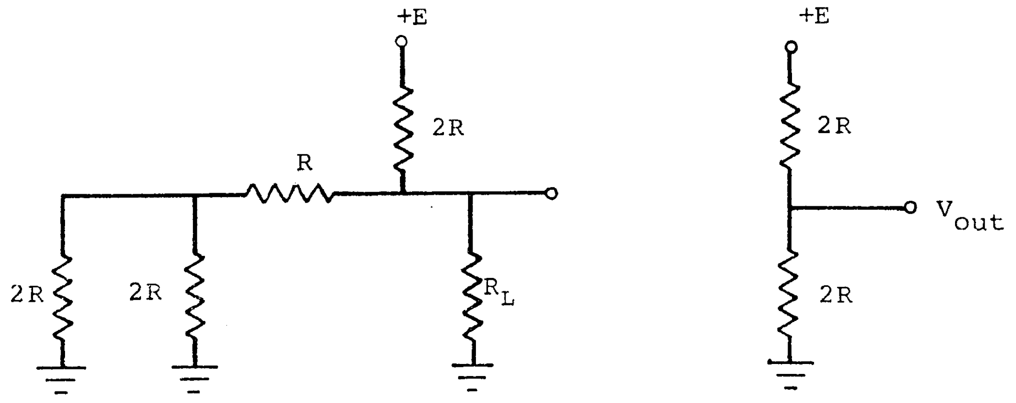
- For example, if ***all of the bits were set to zero***, the switches would be set ***to select the R-2R circuit to the reference ground line*** and so the output voltage would be ***zero volts***.

- If only the ***most significant bit*** (MSB) (bit 0) was set to 1, then switch number one (S_1) will be closed to connect the reference voltage to the resistance network.
- The configuration of the resistance network is such that the output voltage will be ***one half*** (i.e. a voltage divider) of the applied reference voltage when connected via Switch S_1.

- Notice that if Sn is set to ground that the resistance values from the left will equal 2R up to point X^1 .

- Since all of the the switch lines have the same resistance values (R and 2R), the resistance to X^1 would also be 2R if all the switches are set to ground level.

- Load resistor R_L is selected ***much larger*** than the value of 2R so that the resistance resulting from R_L and 2R being in parallel is approximately 2R.



R-2R circuit with Bit 1 = 1

The Equivalent Circuit

Figure 7. A simplified R-2R Equivalent Circuit

- The simplified equivalent circuit (Figure 7) shows that the ouput voltage developed with only the **MSB** set to one will be ***one half of the applied reference*** voltage (+E/2) so that half of the applied voltage is developed if the digital word is 1000 0000.

- Since this is a base two or binary system, as the bit position moves toward the least significant bit (LSB), the digital requested value and the corresponding voltage value will be ***reduced by one half*** of the previous value.

- The values for an 8-bit word would be **50** (7th), **25** (6th), **12.5** (5th), **6.25** (4th) , **3.125** (3rd), **1.562** (2nd), **0.781** (1st), **0.39** (0th).

7. Digital to Analog Conversion (DAC)....continued

As an example, determine the portion of a 5 volt analog signal that will be developed when the 8 bit number **1011 1001** is output to the DAC:

Digital Word	Bit Position	% Equvalent	Voltage value (5 V dc)
1	7th	50.0% x 1	2.5 V dc
0	6th	25.0% x 0	0.0
1	5th	12.5 % x 1	0.625 V dc
1	4th	6.25 % x 1	0.3125 V dc
1	3rd	3.125 % x 1	0.1562 V dc
0	2nd	1.562 % x 0	0.0
0	1st	0.781 % x 0	0.0
1	0th	0.39 % x 1	0.0195 V dc
	Totals	72.265%	3.6132 V dc

- The analog voltage signal developed by the DAC is usually converted to an ***equivalent current signal*** by a voltage to current (E/I) transducer or ***current driver circuit***.

- The resulting 4-20 mA current signals can be run throughout the plant to field mounted ***current to pneumatic*** (I/P) transducers to allow the operation of the pneumatically actuated control valves.

8. Stylized Digital Control System

- With these components then, we have the means to provide a digital control system in that we can ***sense binary input*** values via ***digital inputs*** (DI's) and drive corresponding ***digital output values*** (DO's).

- We can also sense and convert analog values to a dedicated digital value via the ***analog to digital converter*** (ADC) to obtain a digitized working parameter representative of the process condition under study or control.

- Complete control related logic and develop a control decision value that must be applied to change the plant condition.

- Similarly, the digital control decision can be applied to the field via a ***digital to analog converter (DAC)*** so that the logic from the control algorithm solved by the computer can be ***output to the analog field device*** as an analog voltage or current signal.

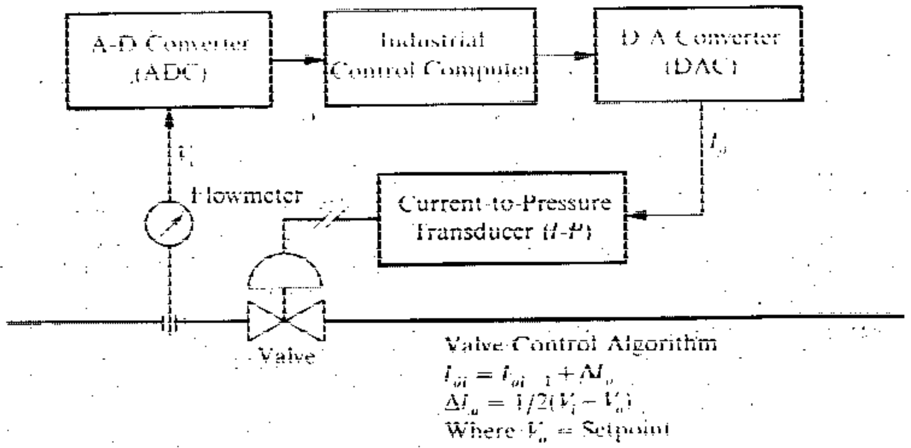

A-D Converter (ADC)
Industrial Control Computer
D A Converter (DAC)
V_i
I_o
Flowmeter
Current-to-Pressure Transducer (I-P)
Valve
Valve Control Algorithm
$I_{oi} = I_{oi-1} + \Delta I_o$
$\Delta I_o = 1/2(V_i - V_o)$
Where V_o = Setpoint

Figure #8 Typical Digitized Control Loop Application

9. Representative Digital Control Program

Control Program Coding Considerations

1. Read, prepare and scale the measurement (M) and setpoint (SP) parameters.

Read the necessary ADC values to obtain the control algorithm inputs. Apply any rationality and validity checks on these signal values, apply any necessary scaling and identify those parameters that will be used for control purposes. You should also decide if this is the first run for this program-in-control (is any initialization required? is the program starting from restart, transfer, manual, etc)

2. Configure comparator logic to determine the control error (E) & action.

Increasing, Increasing Action (***direct***) will be **:** **$E = M - SP$**
Increasing, Decreasing Action (***reverse***) will be: **$E = SP - M$**
Where:
E = control error
M = measured variable parameter
SP = Setpoint parameter

3. Prepare the straight proportional term (PT) :

$PT = K * E$
Where
PT = the Proportional Control Term
E = the control error
K = the controller gain

<u>9. Representative Digital Control Program....continued</u>

4. Sum (integrate) the error:

SUM = SUM + E * TS
where:
SUM = integral error summation term
E = control error
TS = is the control program iteration or sample time in seconds

Note that SUM must be initialized on ***first program operation*** to provide the starting integral value. This initializing value is usually obtained by tracking the actual valve position and then back calculating the needed SUM term to provide that valve position. (i.e. if the valve = 75%, then SUM = f (75%))

5. Prepare the Integral control term (IT) :

IT = (K* SUM) / RT
where:
IT = intergral or reset mode term
K = control **<u>Gain</u>**
SUM = Integral error summation
RT = the **<u>Reset Time</u>** value

You should also monitor to see if this control function has been switched to manual (say by the operator action of a Computer/Auto/Manual station) , and if so - ***track*** the manual signal with the integral term so as to be ready to return to automatic control in a ***bumpless*** fashion.

9. Representative Digital Control Program....continued

6. Prepare the first estimate control signal (CS) :

CS = PT + IT

where:
CS = Control Signal value (usually 0.0-100.0)
PT = Proportional Term
IT = Integral Term

7. Check the control signal for a windup condition:

```
    * if wound up, set the integral term (IT) so that the signal just equals 100.0 or 0.0 with the present proportional term (PT).
    * check if the control signal is acceptable
    IF (CS. GE. 0.0  AND CS. LE. 100.0) THEN 500
    ** ELSE WINDUP EXISTS **
    * Recalculate integral SUM term for next iteration
    * this balances the integral term so the signal is just 100.0
    IF (CS.GT.100.0) THEN SUM = ((100.0 - PT) * RT) / K
          * this balances the integral term so the signal is just 0.0
    IF (CS.LT.0.0) THEN SUM = ((0.0-PT) * RT) / K

    * Recalculate IT for this iteration with the new SUM value
    IT = (K * SUM) / RT
    * set the final control signal for output
    CS = PT + IT
500 CONTINUE
```

8. Output the final control signal to develop a 4-20 mA signal
Output the CS parameter value to the appropriate DAC channel to drive the controlled variable.

9. Service the loop again as per executive scheduler (i.e. 500 millisec)

Computer Subsystem Assignment

1. Briefly explain how pressure switches can be used to provide a discrete level position status information input to a computer digital input subsytem.

Answer: *The pressure switch can be calibrated to provide an inferred level position based on the hydrostatic pressure measured. The pressure switch can be connected into a Digital Input (DI) circuit of a computer subsystem. When the level changes sufficient to trip the pressure switch, the DI will sense this change in status – the DI status will be processed by an input register and then control or monitoring logic would be able to assess the DI condition to give a point level assessment for the process level value by say turning on an associated LED etc on a stylized level display.*

2. Briefly explain how an electrical solenoid valve can be driven by a computer digital output subsystem to apply on/off inflow control for an open tank level control application.

Answer: The control logic decision would be output to a digital output (DO) via a transfer register. The DO signal (perhaps interfaced through a higher powered relay) would be connected to the coil of a solenoid valve. If the solenoid coil is energized – the solenoid three way valve body would be driven to apply a pneumatic signal to the actuator of the flow control valve – admitting inflow to the tank. If the DO is de-energized, then the 3 way solenoid valve would be driven to vent the pneumatic signal from the control valve actuator – closing the control valve and stopping inflow to the tank.

3. Sketch a typical solenoid valve installation in which the solenoid determines the on/off pressure in a spring opposed diaphragm control valve actuator chamber by admitting or blocking a relatively constant pressure pneumatic supply. Make sure you consider the entire control valve operation cycle so that the valve is able to fully open and to fully close (Hint: you must be able to vent the trapped actuator signal).

4. Make a logic flow chart diagram to show the logic you would implement to monitor the tank level via pressure switch signals andto control the inflow valve by energize/de-energize solenoid valve operation. Explain your logic to describe one complete tank level cycle of operation.

Computer Subsystem Assignment…continued

5. Briefly explain the principle of operation for a dual ramp analog to digital convertor.
Answer: The measured process signal is applied to the input of an integrating amplifier. The output of the integrating amplifier will effectively ramp upwards at a rate proportional to the magnitude of the original signal. This upward ramp is always maintained for a defined period of time and then the counter is reset to zero and a negative reference voltage is applied to the integrating amplifier. Now the previous integrated signal is ramped down at a constant rate - each clock pulse will increment the counter until the integrating amplifier output reaches zero at which time the counter is stopped and the count is proportional to the previous integrating amplifier output signal which in turn was proportional to the measured field signal. So a digital count has been prepared for use by the input register.

6. Why is it superior to have a non-zero digital count representative of the lowest signal range value?
Answer: This approach will allow you to discriminate against a valid zero value signal as opposed to a zero signal which could result say from a failed calculation, conversion, or output subsystem.

7. What is the general purpose of a sample and hold amplifier circuit in a computer input subsystem?
Answer: The S&H amplifier circuit provides the means (by a capacitance circuit) to apply the magnitude of the field signal to the ADC for conversion while actually physically disconnecting the field circuit from the ADC input.

8. Make a sketch to show how an incrementing pointer value can be used to address an ADC selection circuit and to address a data table array entry for contiguous parameters.

9. Briefly explain how an R-2R voltage divider circuit can be used in a digital to analog convertor to develop an output voltage as a function of a digital word value.
Answer: The R-2R circuit is configured in such a way that each time a switch is closed (from MSB toward LSB) that one half of the remaining voltage is included for the output signal. This is possible since the resistances are all arranged either as 2R in parallel separated by 1R in series. The logic need only decide what circuit configuration is needed to develop the desired output voltage and those bits are set accordingly.

10. Make a logic flow chart diagram and use it to explain the basic operation of a programmed proportional plus integral control algorithm.

DIGITAL CONTROL CONCEPTS

This information is intended to supplement the instrumentation & controls lectures with additional information on the digital computer application for controls with implementation details.

Lecture Topics

1. Digital Logic Overview
2. Decimal Numbering System
3. Binary Numbering System
4. Octal Numbering System
5. Hexadecimal Numbering System
6. Computer Use of Numbering Systems
7. Input Subsystem
8. Computer Memory
9. Central Processing Unit
10. Output Subsystem

1. DIGITAL LOGIC OVERVIEW

- A logic decision can easily be displayed by changing the particular ***state of a device*** from say energized to de-energized.
- Imagine a list of questions that can be answered unequivocally by a ***yes*** or ***no*** answer. We could allocate the ***yes answer*** (or state) to the ***energized state*** and the ***no answer*** (or state) to the ***de-energized state***.
- As we only have two choices here (yes or no), this system can be classed as a ***binary*** system.
- A simple electric circuit with an ***on/off*** switch can be used to signal the binary ***yes*** or ***no*** condition by lighting a lamp in the circuit.
- The yes state or ***energized state*** could have the ***bulb illuminated.***
- The no state or ***de-energized*** state could have the ***lamp extinguished***.
- And so an observer could determine the ***answer decision*** by monitoring the bulb status.

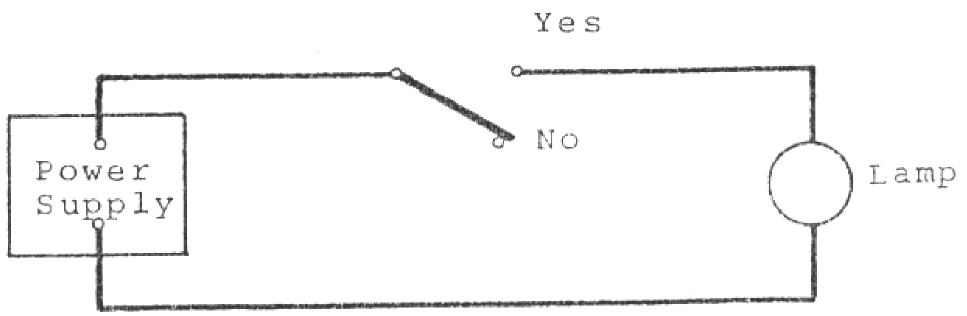


Figure 1. Binary Lamp State used to indicate yes or no condiiton

1. DIGITAL LOGIC OVERVIEW....continued

- Since this simple system has only two states, we can arbitrarily assign the numerical value of ***one*** (1) to the ***energized state*** and the value of ***zero*** (0) to the ***de-energized state***.

- A base two or ***binary number system*** is required to allow a computer to operate as a decision making (yes or no?) device.

- The smallest storage unit in the digital computer is one **bi**nary digi**t** which is contracted to **bit**.

- The bit will be displayed as either a ***0*** or a ***1*** in the binary machine.

- Such a simple logic system consisting of these two states can be efficiently and conveniently utilized in computers to make ***continual binary logic decisions*** which can then be used for ***control purposes***.

NUMBERING SYSTEMS

<u>2. The Decimal Numbering System</u>

- The most common numbering system used is the ***base 10*** or decimal system.

- For example, the number 124 can be written as follows:

$(\underline{\mathbf{1}} \text{ x } 10^2) + (\underline{\mathbf{2}} \text{ x } 10^1) + (\underline{\mathbf{4}} \text{ x } 10^0) = \underline{\mathbf{124}}$ Base 10

- By knowing the ***sum*** of the assigned values for ***hundreds***, ***tens*** and ***ones***, we are able to determine the total value of 124 (which we already know because we are familiar with this system).

- It is worthwile to consider a specific value (i.e. 999) so that we can consider what happens when we add one to the number.

$(\underline{\mathbf{9}} \text{ x } 10^2) + (\underline{\mathbf{9}} \text{ x } 10^1) + (\underline{\mathbf{9}} \text{ x } 10^0) = \underline{\mathbf{999}}$ Base 10

- Now let's ***add one*** to this number:

$999 + 1 = (\underline{\mathbf{9}} \text{ x } 10^2) + (\underline{\mathbf{9}} \text{ x } 10^1) + (\underline{\mathbf{9+1}} \text{ x } 10^0)$

$= (\underline{\mathbf{9}} \text{ x } 10^2) + (\underline{\mathbf{9}} \text{ x } 10^1) + (\underline{\mathbf{10}} \text{ x } 10^0)$

- But note that $(10 \text{ x } 10^0)$ is equal to $(1 \text{ x } 10^1)$ and so the ***ones*** value should be ***shifted upward*** to the ***tens*** position.

2. Decimal Numbering System......Continued

- ....and so the ***ones*** value should be ***shifted upward*** to the ***tens*** position. , resulting in the following:

$(\underline{\mathbf{9}} \times 10^2) + (\underline{\mathbf{9}} \times 10^1) + (\underline{\mathbf{10}} \times 10^0) = (\underline{\mathbf{9}} \times 10^2) + (\underline{\mathbf{10}} \times 10^1) + (\underline{\mathbf{0}} \times 10^0)$

- Again, the ***tens*** value of $(\underline{\mathbf{10}} \times 10^1)$ is equal to $(\underline{\mathbf{1}} \times 10^2)$ and so the ***tens*** value should be ***shifted*** to the ***hundreds*** position, resulting in the following:

$(\underline{\mathbf{9}} \times 10^2) + (\underline{\mathbf{10}} \times 10^1) + (\underline{\mathbf{0}} \times 10^0) = (\underline{\mathbf{10}} \times 10^2) + (\underline{\mathbf{0}} \times 10^1) + (\underline{\mathbf{0}} \times 10^0)$

- Finally, the hundreds value of $(\underline{\mathbf{10}} \times 10^2)$ is equal to $(\underline{\mathbf{1}} \times 10^3)$ so that the ***hundreds*** value should be shifted to the ***thousands*** position. So after one final ***shift*** the sum of 999 + 1 becomes:

$(\underline{\mathbf{1}} \times 10^3) + (\underline{\mathbf{0}} \times 10^2) + (\underline{\mathbf{0}} \times 10^1) + (\underline{\mathbf{0}} \times 10^0) =$ **1000** Base 10

- It can be seen from this exercise that the ***largest number*** that can appear in a particular value position must be the ***base value*** (10 in this case) ***less one*** (so 9 in this case). If this largest number is exceeded, the numerical indication must ***shift*** upward to the next higher power position value.

3. The Binary Numbering System

- The binary or **base two** system used in computers will indicate the values for various powers of 2.

- The maximum digit allowed at any one position will be the ***base value*** (2 in this case) ***less one*** (or 1 in this case) so that only ***0*** or ***1*** will be utilized.

- Now to consider several ***binary examples*** to demonstrate this numbering system:

1. Represent **decimal 4** in binary code.
Decimal 4 = $2^2 = (\underline{\mathbf{1}} \text{ x } 2^2) + (\underline{\mathbf{0}} \text{ x } 2^1) + (\underline{\mathbf{0}} \text{ x } 2^0) = \underline{\mathbf{100}}$ Base 2

2. Represent **decimal 9** in binary code. Decimal 9 = 8 + 1:
Decimal $8 + 1 = 2^3 + 2^0 = (\underline{\mathbf{1}} \text{ x } 2^3) + (\underline{\mathbf{0}} \text{ x } 2^2) + (\underline{\mathbf{0}} \text{ x } 2^1) + (\underline{\mathbf{1}} \text{ x } 2^0)$
$= \underline{\mathbf{1001}}$ Base 2

3. Represent **decimal 49** in binary code. Decimal 49 = 32 + 16 + 1:
Decimal $32 +16 + 1 = 2^5 + 2^4 + 2^0$
$= (\underline{\mathbf{1}} \text{ x } 2^5) + (\underline{\mathbf{1}} \text{ x } 2^4) (\underline{\mathbf{0}} \text{ x } 2^3) + (\underline{\mathbf{0}} \text{ x } 2^2) + (\underline{\mathbf{0}} \text{ x } 2^1) + (\underline{\mathbf{1}} \text{ x } 2^0)$
$= \underline{\mathbf{110001}}$ Base 2

3. The Binary Numbering System......continued

- You can see from these examples that although the binary numbering system is simple, it may be easy for a person to make a mistake reading a large group of 1's and 0's in a row.
- As a result, the numers are usually grouped in an ***octal system*** (based on 8) or a ***hexadecimal system*** (based on 16).
- As before, the largest single value in one position must be one less than the base number, so for ***octal the largest value is** 7* while for ***hexadecimal the largest value is 15***.

4. Octal Numbering System

- The largest numerical value (i.e. 7) in any one position in the octal system is represented as:

Decimal 7 = $\underline{\mathbf{4}} + \underline{\mathbf{2}} + \underline{\mathbf{1}} = (\underline{\mathbf{1}} \times 2^2) + (\underline{\mathbf{1}} \times 2^1) + (\underline{\mathbf{1}} \times 2^0) = \underline{\mathbf{111}}$ Base 2

This becomes $(0 \times 8^1) + (\underline{7} \times 8^0) = \underline{7}$ Octal

- As an example, convert ***Decimal 75*** to the ***Octal*** equivalent.

Decimal 75 = 64 + 8 + 3:

Decimal $64 + 8 + 3 = (\underline{\mathbf{1}} \times 8^2) + (\underline{\mathbf{1}} \times 8^1) + (\underline{\mathbf{3}} \times 8^0) = \underline{\mathbf{113}}$ Octal

- Note that the octal system is just a ***short hand way*** of writing the base two numbers. For the example;

Decimal 75 = 64 + 8 + 3 could be written in base two as follows:

64 + 8 + 3 = (shown on next line to allow enough room)

$= (\underline{\mathbf{1}} \times 2^6) + (\underline{\mathbf{0}} \times 2^5) + (\underline{\mathbf{0}} \times 2^4) + (\underline{\mathbf{1}} \times 2^3) + (\underline{\mathbf{0}} \times 2^2) + (\underline{\mathbf{1}} \times 2^1) + (\underline{\mathbf{1}} \times 2^0)$

- Grouping this value in sets of ***3 bits*** for maximum number seven at each position results in:

$\underline{\mathbf{1}}\ \ 00\underline{\mathbf{1}}\ \ 0\underline{\mathbf{11}} = 00\underline{\mathbf{1}}\ \ 00\underline{\mathbf{1}}\ \ 0\underline{\mathbf{11}} = \underline{\mathbf{113}}$ Octal (which we had found before)

5. Hexadecimal Numbering System

- The largest numerical value (i.e. 15) in one position in the hexadecimal system is represented as:

Decimal 15 = **8** + **4** + **2** + **1** = (**1** x 2^3) + (**1** x 2^2) + (**1** x 2^1) + (**1** x 2^0)
= **1111** Base 2
This becomes (0 x 16^1) + (**15**x 16^0)

- The hexadecimal number system corrsponds to the decimal system in the following manner:

Decimal:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Hexadec:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

so that **15** Decimal = **1111** Base 2 or **F** hexadecimal

Hexadecimal Coversion Example

- As an example, convert ***Decimal 279*** to the ***Hexadecimal*** equivalent.

Decimal 279 = 256 + 16 + 7:
Decimal 256 + 16 + 7 = $(\underline{\mathbf{1}} \times 16^2) + (\underline{\mathbf{1}} \times 16^1) + (\underline{\mathbf{7}} \times 16^0)$
= $\underline{\mathbf{117}}$ Hexadecimal

- Note that the hexadecimal system is also just a short hand way of writing the base two numbers. For the example Decimal 279 = 256 + 16 + 7 could be written as base two as follows:

256 + 16 + 7 =
$= (\underline{\mathbf{1}} \times 2^8) + (\underline{\mathbf{0}} \times 2^7) + (\underline{\mathbf{0}} \times 2^6) + (\underline{\mathbf{0}} \times 2^5) + (\underline{\mathbf{1}} \times 2^4) + (\underline{\mathbf{0}} \times 2^3) + (\underline{\mathbf{1}} \times 2^2)$
$+ (\underline{\mathbf{1}} \times 2^1) + (\underline{\mathbf{1}} \times 2^0)$

- Grouping in sets of ***four bits*** for maximum number fifteen at each position results in:

$\underline{\mathbf{1}}$ 000$\underline{\mathbf{1}}$ 0$\underline{\mathbf{111}}$ = 000$\underline{\mathbf{1}}$ 000$\underline{\mathbf{1}}$ 0$\underline{\mathbf{111}}$ = $\underline{\mathbf{117}}$ Hexadecimal
(as we had shown before)

6. Computer Use of Numbering Systems

- Each ***task*** for the computer operation, such as ***add***, ***subtract***, ***multiply***, ***divide***, etc, can be assigned a unique number.

- The unique combination of 1's and 0's will always be recognized by the computer as the ***assigned task***.

- In this fashion the computer can execute an assigned program task by executing the ***coded numerical sequence instructions***.

- ***Descriptive labels*** can be applied to the resulting instruction set to faciliate use and checking by programmers or maintenance staff.

- For example, an ***add immediate*** instruction could be given the mnemonic ***ADDI*** and assigned the value 06120 Octal or 0C50 Hexadecimal.

- This means that anytime the computer sees the instruction
000 110 001 010 000, then an ***add immediate*** action will be completed.

- It is the responsibility of the programmer to ensure that ***data*** and ***instructions*** are always presented in the expected order and format.

General Computer Operation

- The computer can be considered as being made up of several ***interfacing sections***, each of which has a defined task.

These sections are:

- ***Input Subsystem***, (contact the field to read devices)
- ***Computer Memory***, (store designated data for future use)
- ***Central Processing Unit*** (CPU), (perform logic assessments)

  and

- ***Output Subsystem.*** (contact the field to write to devices)

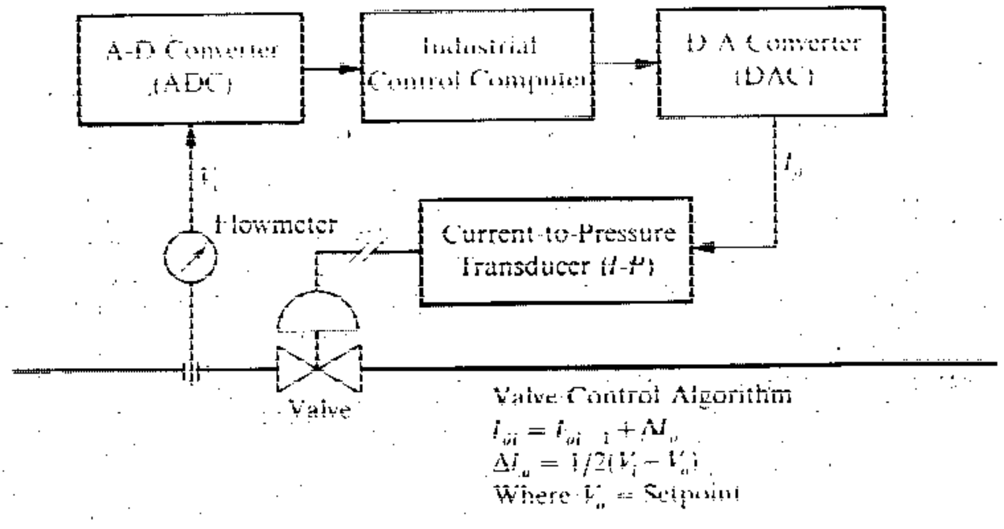

A-D Converter (ADC)
Industrial Control Computer
D-A Converter (DAC)
V_i
I_o
Flowmeter
Current-to-Pressure Transducer (I-P)
Valve
Valve Control Algorithm
$I_{oi} = I_{oi-1} + \Delta I_o$
$\Delta I_o = 1/2(V_i - V_o)$
Where V_o = Setpoint

Figure 2. Digitized Analog Control Loop Diagram

7. Input Subsystem

- The Input Subsystem allows the computer to ***read devices*** so that actual ***plant data*** can be dynamically entered into the computer on a continual basis.

- Typical ***input devices*** may be ***analog to digital converters*** (ADC's), ***Digital Inputs*** (DI's), ***Keyboards***, ***Priority Interrupt Modules*** (PIM's), ***data links*** from other computers, ***reader devices*** (tape, disc, card,etc).

- The data input subsystem must be ***fast enough*** to track significant parametric changes to allow adequate control response times.

- The input subsystem must not ***burden*** (i.e. load down) the computer operation by the ***input read*** and ***conversion activities***.

- Sufficient ***barriers*** must also be provided to protect the computer from ***field generated faults*** and ***electrical disturbances***.

- Adequate ***redundancy*** of signals should be provided to allow high confidence assessment of signal ***rationality*** and ***validity*** as well as preventing a ***single failure*** from disabling the automatic computer control mode.

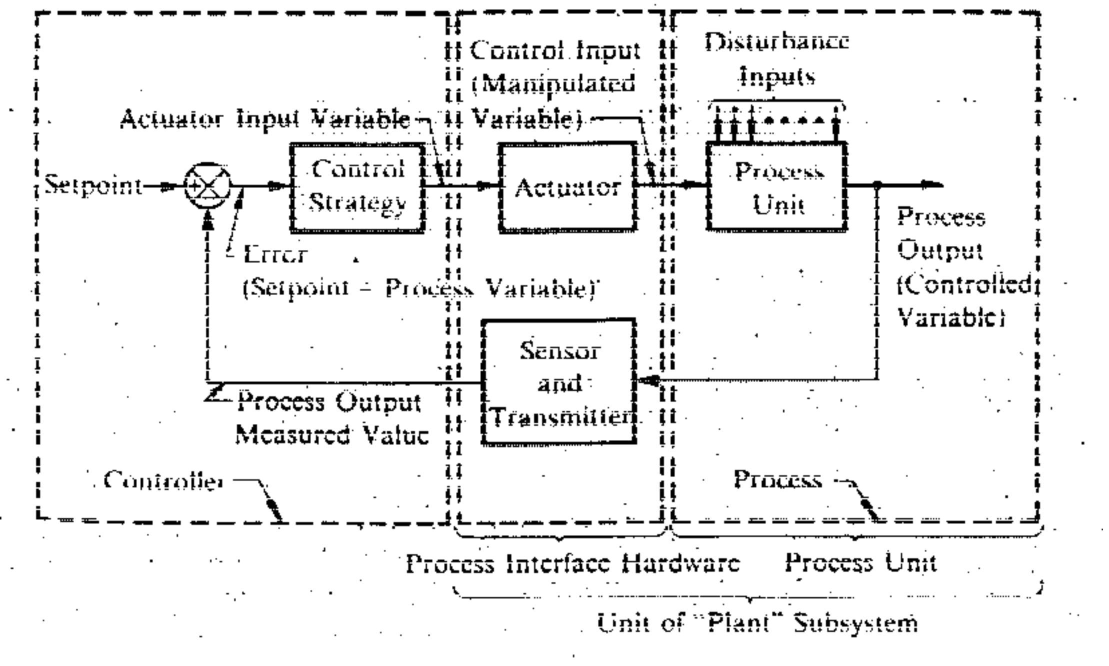

Control Input (Manipulated Variable)
Disturbance Inputs
Actuator Input Variable
Setpoint
Control Strategy
Actuator
Process Unit
Process Output (Controlled Variable)
Error (Setpoint − Process Variable)
Sensor and Transmitter
Process Output Measured Value
Controller
Process
Process Interface Hardware
Process Unit
Unit of "Plant" Subsystem

Figure #3 Generalized Plant Control Installation

8. Computer Memory

- This is the memory that the computer requires for ***hardware operation*** and ***program execution*** purposes.
- This memory is divided into Read Only Memory (**ROM**) and Random Access Memory (**RAM**).
- The ROM would contain the power up, ***bootstrap*** information necessary to start the computer and to conduct ***essential low level support*** functions in a continual, reliable manner.
- The RAM would contain the ***executive*** and application programs which monitors the plant status and makes corrective control decisions.
- RAM application programs are characteristically ***loaded from disc*** at power-up time once the computer system has been powered up and execution mode started (i.e booted up).
- RAM is an immediate or ***short term memory*** (for that execution cycle and perhaps some short term storage capability).
- For longer term data storage, the designated data could be ***stored to disc*** within allocated datafiles so that specific plant data over a requested calendar date and time could be called up for future review - this is usually referred to as a ***historical data storage*** and ***retrieval*** system (HDSR).

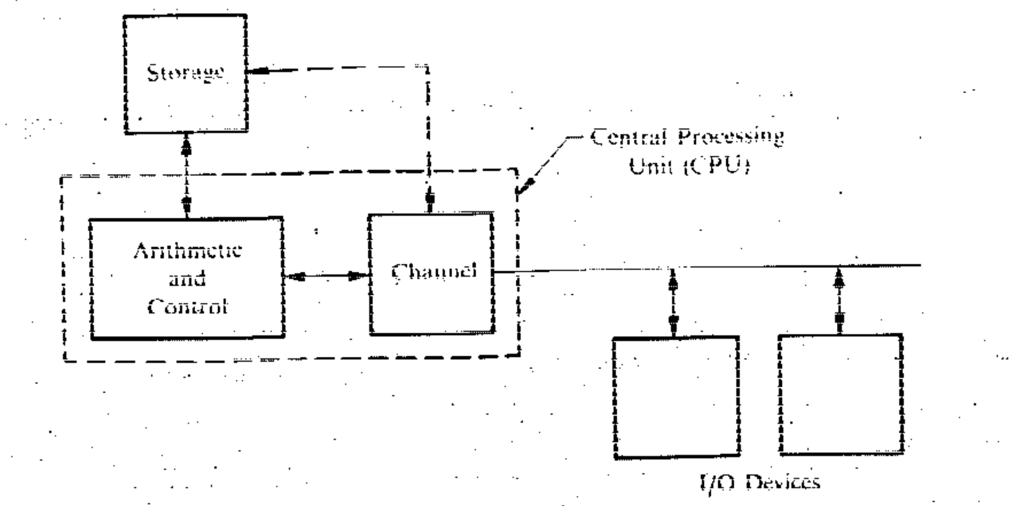


Figure #4 Key Computer System Components

9. Central Processing Unit (CPU)

- The central processing unit or CPU performs three essential jobs for the computer system. These are the ***control*** section, the ***arithmetic*** section and the ***computer control panel***.

The Control Section - controls the computer operations and interfaces.

- The control section makes all computer operation decisions by ***executing*** the programmed ***instructions*** with ***data*** read from memory or received from the input subsystem.
- The control section is a group of circuits which allows the interpretation of the defined instruction set and the development of ***resulting commands*** which are then sent to logic circuits, memory systems and input/output devices.

The Arithmetic Section - performs all arithmetic operations for the computer.

- The arithmetic section does the actual arithmetic ***problem solving*** (e.g. add or subtract) and manipulates the data as directed by the control section.
- The arithmetic section contains ***registers*** (i.e. think of the register as a dynamic data word) that hold the results of the calculations and logic elements (electronic circuitry) so that the data in the registers can be combined.
- The registers of the computer allow for the ***temporary storage*** of data, the shifting or ***arithmetic manipulation*** of that data, loading of ***instructions*** as well as ***addressing*** functions and ***transfer*** of the final data to a target destination.

Control Panel - indicates the computer ***status*** and ***diagnostics***

- The computer control panel provides the operator with ***direct access*** to the CPU ***registers*** and ***memory***. Panel switches and indicators allows register contents to be examined and/or set to check or modify ***computer status*** or the ***program progress***. Usually, the first power-up or ***restart*** for the computer would be initiated from the computer control panel.

10. Output Subsystem

- The Output Subsystem allows the computer to ***drive field devices*** or ***write to devices*** so that data generated inside the computer can be dynamically applied to the plant equipment to effect plant control functions.
- Typical output devices may be ***digital to analog*** converters (DAC's), ***Digital Outputs*** (DO's), ***Pulsed Output Points*** (POP's) graphics ***displays***, ***printers***, data links to other computers, ***storage devices*** (tape, disc, etc).
- The data output subsystem must be ***reliable*** and ***fast*** enough to initiate control parametric changes to provide adequate control response times without burdening the computer operation (i.e. loading down).
- The output system must also provide sufficient barriers to protect the computer from ***field generated faults*** and ***electrical disturbances.***
- At the same time, checks should be made to ensure that computer generated signal changes are ***not a source of electrical noise*** or fault signals for the other plant equipment.
- Precautions that can be taken to minimize such an introduction of noise would include ***signal routing***, ***device specification***, ***suppression techniques***, ***modes of operation***, etc
- Adequate ***redundancy*** of control signals should be provided to prevent a ***single failure*** from disabling the automatic computer control mode.

Digital Control Introduction Assignment

1. A digital input system is organized to read 16 digital inputs by way of a 16 bit register in the subsytem. What decimal number equivalent can be expected if all of the odd bits are set (value = 1) and all of the odd bits are off (value = 0). First sketch the register to show the binary bit pattern, then determine the hexadecimal value and finally convert that value to the decimal equivalent. Treat the odd bits as bit position 1,3,5.....15 and the even bits as bit position 0,2,4....14 with bit 0 the LSB and bit 15 the MSB.

2. A computer instruction is known to be 6643 octal. What will this instruction be in hexadecimal - sketch the 16 bit pattern.

3. What sort of performance problems could possibly be encountered if a programmer used a dynamic data area to store instructions words for future execution purposes? What would be a good general programming rule to follow?

4. Why is it important for an Input subsystem to be fast enough to representatively convert field parameter changes? How would the lack of this responsiveness effect the controlability of the loop?

5. Why is it important to take the precaution to ensure that one field input electrical problem (such as grounding) can not disrupt the entire input subsystem. Use a multiple input, computer control application as the example to explain your answer.

6. List some illustrative industrial computer input and output devices that may be encountered in an instrumentation or control application.

7. In general, how could a computer driven device introduce electrical noise into the plant? What are some precautions that could be taken to minimize this effect?

8. A control decision requires that the equivalent of 4728 decimal to be output as a hexadecimal number. Convert 4728 decimal to the equivalent hexadecimal value and then show the expected bit pattern for a 16 bit register that will output this value. (i.e. show the binary equivalent for this hexadecimal number).

<u>Digital Control Introduction Assignment – suggested answers</u>

1. A digital input system is organized to read 16 digital inputs by way of a 16 bit register in the subsytem. What decimal number equivalent can be expected if all of the odd bits are set (value = 1) and all of the even bits are off (value = 0). First sketch the register to show the binary bit pattern, then determine the hexadecimal value and finally convert that value to the decimal equivalent. Treat the odd bits as bit position 1,3,5.....15 and the even bits as bit position 0,2,4....14 with bit 0 the LSB and bit 15 the MSB.

   All Odd bits on: 1010 1010 1010 1010
   A A A A
   10* 4096 + 10 * 256 + 10 * 16 + 10 = 43,690 decimal

2. A computer instruction is known to be 6643 octal. What will this instruction be in hexadecimal - sketch the 16 bit pattern.

   6 6 4 3 = 110 110 100 011 in octal
   or 1101 1010 0011 in hexadecimal
   which is D A 3 = DA3 hexadecimal

3. What sort of performance problems could possibly be encountered if a programmer used a dynamic data area to store instructions words for future execution purposes? What would be a good general programming rule to follow? *This could result in corrupted data in that data rewrites could change the instructions so that the execution is unexpected and not according to the original program rules – the results from such a program would be unpredictable – and this is not good practice.*

4. Why is it important for an Input subsystem to be fast enough to representatively convert field parameter changes? How would the lack of this responsiveness effect the controlability of the loop? *If the conversion was too slow, then the reading would be skewed by the apparent value and so additioanl error and lag would be included in the control loop, degrading the quality of the control obtained.*

5. Why is it important to take the precaution to ensure that one field input electrical problem (such as grounding) can not disrupt the entire input subsystem. Use a multiple input, computer control application as the example to explain your answer. *Otherwise a single failure could disrupt the entire control system and initiate a unit outage condition.*

6. List some illustrative industrial computer input and output devices that may be encountered in an instrumentation or control application.

7. In general, how could a computer driven device introduce electrical noise into the plant? What are some precautions that could be taken to minimize this effect?

8. A control decision requires that the equivalent of 4728 decimal to be output as a hexadecimal number. Convert 4728 decimal to the equivalent hexadecimal value and then show the expected bit pattern for a 16 bit register that will output this value. (i.e. show the binary equivalent for this hexadecimal number).