BNWL-1378
UC-32

PROCESS CONTROL COMPUTER SYSTEM
SPECIFICATION GUIDELINE

June 1970

AEC RESEARCH &
DEVELOPMENT REPORT

BNWL-1378

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

PACIFIC NORTHWEST LABORATORY

RICHLAND, WASHINGTON

operated by

BATTELLE MEMORIAL INSTITUTE

for the

UNITED STATES ATOMIC ENERGY COMMISSION UNDER CONTRACT AT(45-1)-1830

3 3679 00061 6690

BNWL-1378

UC-32, Mathematics
and Computers

PROCESS CONTROL COMPUTER SYSTEM
SPECIFICATION GUIDELINE

M. D. Erickson
R. E. Mahan
T. A. Seim
R. A. Walker

Control and Instrumentation Department
Systems and Electronics Division

Work performed under LMFBP Program Subtask 4-4.2.8
"Applications for Computers - Component Development"

June 1970

FIRST UNRESTRICTED
DISTRIBUTION MADE

JUL 21 '70

BATTELLE MEMORIAL INSTITUTE
PACIFIC NORTHWEST LABORATORIES
RICHLAND, WASHINGTON 99352

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U.S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy \$3.00; Microfiche \$0.65

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	1
SPECIFICATION GUIDELINE	3
1.0 SUMMARY DESCRIPTION	3
2.0 CENTRAL PROCESSING UNIT	3
2.1 CPU Organization	3
2.2 Instruction Set	3
2.3 Arithmetic Unit	5
2.4 Core Memory	6
2.5 Addressing	8
2.6 Interrupts	10
2.7 Direct Memory Access	12
2.8 Clocks	13
2.9 Stall Alarm	13
2.10 Power Failure/Restart	14
2.11 Operators Console	15
2.12 Input/Output	15
3.0 PERIPHERALS	18
3.1 Digital-to-Analog Converters	18
3.2 Analog Multiplexer	20
3.3 Analog-to-Digital Converter	23
3.4 Magnetic Rotating Bulk Storage	25
3.5 Magnetic Tape Bulk Storage	26
3.6 High Speed Printers	30
3.7 Low Speed Printers	33
3.8 Paper Tape Reader/Punch	34
3.9 Card Readers and Punches	35
3.10 Cathode-Ray-Tube Displays	36
3.11 Plotters	40
3.12 Miscellaneous Output Equipment	41
4.0 SOFTWARE	42
4.1 Loaders	42
4.2 Editor Program	43

4.0	SOFTWARE (continued)	
4.3	Debugging Aid Program	43
4.4	Assembler	43
4.5	Subroutine Library	45
4.6	Utility Programs	46
4.7	Hardware Diagnostic and Maintenance Aids	47
4.8	FORTRAN Compiler	47
4.9	Other Higher Level Languages	47
4.10	Executive System	48
4.11	Benchmark Problems	51
4.12	Software Documentation	51
5.0	MISCELLANEOUS HARDWARE SPECIFICATIONS	53
5.1	Environmental	53
5.2	Power	53
5.3	Enclosures and Cabinet Configuration	53
5.4	Spare Parts	53
5.5	Special Tools	54
5.6	Documentation	54
5.7	Reliability	54
5.8	Quality Assurance	56
5.9	Source Inspection Prior to Shipment	56
5.10	Installation and Acceptance Test Procedure	56
5.11	Warranty	57
6.0	TRAINING COURSES AND MAINTENANCE	58
6.1	Software Course	58
6.2	Hardware Course	58
6.3	Maintenance	58
	ACKNOWLEDGEMENTS	59
	REFERENCES	59

**PROCESS CONTROL COMPUTER SYSTEM
SPECIFICATION GUIDELINE**

M. D. Erickson
R. E. Mahan
T. A. Seim
R. A. Walker

INTRODUCTION

The purpose of this **document** is to provide detailed specification guidelines for preparing **procurement** documents for **computer-based** data acquisition and process control systems. Its preparation is based in part upon past experience at Battelle-Northwest in the development of **computer** systems as applied to nuclear research and development.

The guideline outlines the general requirements for on-line real-time process control systems. Each application would of course require a configuration which **was** designed to uniquely meet specific needs. It is not the intention of this **document** to function as a specification for all systems. Rather, it may be used as an example, knowing that the bulk of the **requirements** are included, such that only minimal effort will be required to adapt each specification to the particular application.

There are two basic approaches to specifying computer systems. The first is to use functional specifications wherein the basic purpose and operation of the system is described along with the inputs **from** the process, outputs to the process (e .g. control points) and outputs to the operator and/or experimenter. It is then the responsibility of the bidder to propose how the functional requirements can best be met. He does this by configuring the **system** hardware and software, describing how it would function and what it would cost. Occasionally this approach is advantageous to the buyer in that the vendor must do all the systems proposal work, and the buyer merely selects the lowest cost **system** meeting functional requirements.

The second method is to use detailed technical specifications. This approach is usually more applicable to **competitive** bidding on a least cost

basis. It requires the specification of each central processing unit component such as memory size, speed, etc., and detailed peripheral characteristics. The specification **would** serve as a firm requirement for bidder response. Any exceptions taken to the specification would result in the bid being judged nonresponsive and therefore not to be **considered**.

It is to the preparation of detailed technical specifications that this **document** addresses itself. Its goal is to aid the systems designer in the specification and development of data acquisition and process control systems.

In addition to the specifications, the terms and conditions under which the purchase will be made should be included. This so called "Boiler Plate" includes such topics as **delivery** period, delivery location (e.g. f.o.b. Richland, Washington), qualifications of the bidder (e.g. dependability, experience, manufacturing facilities, and adequate financial resources), payment **terms**, and details of the required bidder response. The bidder should be required to respond to the specification describing any exception taken to any paragraph.

SPECIFICATION GUIDELINE

1.0 SUMMARY DESCRIPTION

A brief description of the application for which the computer system is to be purchased should be presented to orient the prospective bidder. It would include a description of the process, the data to be acquired, the variables to be controlled, and the outputs desired. The location of the computer and any special environmental problems should be mentioned. A brief statement of the desired functions such as scan, log, alarm, display, direct digital control or supervisory control would be appropriate. The description could include a list and/or block diagram detailing each major piece of equipment. A sample block diagram is shown in Figure 1.

2.0 CENTRAL PROCESSING UNIT

2.1 CPU Organization

Most contemporary CPU's are organized for fully parallel operation although all combinations from serial memory - serial arithmetic to byte memory parallel arithmetic to parallel memory - parallel arithmetic may be found. Organization is primarily based on speed/economic trade-offs with serial organization permitting lower overall acquisition costs. Where speed cannot be sacrificed, parallel organization should be specified.

2.2 Instruction Set

This section is optional. It can be used to list certain critical instructions to be supplied with the system. Examples of special instructions that are valuable in process control applications include the following:

- Three way branch (skip on >, <, or =)
- Increment/decrement
- Memory referenced conditional branch
- Rotate (or normalize) and conditional branch
- Increment index and conditional branch.

Some computers contain the ability to be microprogrammed using non-memory referenced instructions. This is done by using the

FIGURE 1
SAMPLE BLOCK DIAGRAM

basic instructions built into the hardware to specify several shift, skip or input/output transfer commands to be performed within one instruction. This capability may be useful in applications where the utilization of time and memory is critical.

2.3 Arithmetic Unit

The arithmetic unit performs arithmetic operations between accumulators or general purpose registers; a word contained in the accumulator, or accumulators, and a word, or words, from memory. A single address machine fetches a single word from memory. Data can be represented by one of several codes (e.g., one's or two's complement binary). Normally, either is acceptable.

2.3.1 Hardware Arithmetic

Arithmetic calculations may be done in a computer by hardware or software, depending on the sophistication of the command structure of the machine. Hardware implementation of arithmetic operations always allows them to run much faster (typically ten times), but makes the machine more complex and expensive.

The following means of performing arithmetic should be considered.

Software - Single Precision

Double Precision

Floating Point

Hardware - Single Precision

Double Precision

Floating Point

At the present time most small computers offer only software arithmetic or single precision, fixed point hardware arithmetic. If greater precision is required, double precision arithmetic can be specified. Double precision hardware provides accumulators and arithmetic registers having twice the basic length of the standard

computer word. These double length words are stored in two adjacent memory locations. Double precision hardware is available from only a limited number of small computer manufacturers. Where a greater range of numbers needs to be used (e.g. $\pm 10^{60}$), floating point hardware can be specified. Floating point hardware, like double precision hardware, is only available from a limited number of manufacturers.

2.3.2 Execution Times

The maximum allowable execution times required to add, subtract, multiply, and divide are normally specified. The specification should define execution times including memory fetch times in addition to arithmetic operation times and should indicate whether these times relate to hardware or software operations and whether the times defined are for fixed point single precision, fixed point double precision, or floating point executions.

2.4 Core Memory

2.4.1 Word Length

Specify the number of bits, both for the instruction word and the data word. Always define whether or not the specified length includes parity or protect bits. The word length necessary to **fulfill** a particular requirement is **determined** by a number of factors. Consideration of the complexity of the operations being performed, the accuracy of the data to be acquired and stored in a single word, and the addressing capability required are a few of the many parameters to be weighed.

2.4.2 Number of Words

Specify the minimum number of complete words required. some manufacturers define memory size in terms of bytes (parts of a complete word, usually 8 bits). Working memory is normally magnetic core type and can usually be expanded for future application. If expansion will be

necessary in the future so specify. Memory size is normally specified in integer multiples of 1024 word blocks (e.g. 1024, 2048, 4096, 8192, etc.). Memory sizing must reflect the amount of storage needed to contain the largest resident program or programs, including all necessary data. Care should be exercised in defining the anticipated storage requirements to allow sufficient space to contain manufacturer's software as well as user programs. For example, two different FORTRAN packages may be offered by one manufacturer, a basic one which will operate a 4K memory, and an extended version which requires an 8K memory. The 8K version cannot be used on a 4K machine.

2.4.3 Memory Speed

Memory speed is normally specified in memory read/restore cycle time. If desired, access time (read only time) may also be separately specified. The use of a "bench mark" program (typical or worst case of the application) may be useful in determining memory speed requirements.

2.4.4 Memory Parity

If the specified application requires high reliability, memory parity is justified. Memory parity is used to detect certain types of errors during the transfer into or out of the main memory. Parity adds one bit to each memory word. This addition is usually optional although it is standard in some machines and not available in others. This option is not readily added in the field without major memory changes and should be specified during the original purchase activity.

2.4.5 Memory Protect

For some applications, especially real-time process control, it is desirable to protect certain areas of the working memory from being disturbed (e.g., erased or otherwise modified). A memory protect option is

normally available which permits read/restore operations, but inhibits write operations in specified areas. Memory protect options vary depending upon the manufacturer from protection by word to protection by block (usually 512 words). Protection by word provides the greatest flexibility since one programmable bit is added to each word. Block protection is more economical since only one bit per block is required, but the user cannot selectively write within the protected block.

2.4.6 Read-Only Memory

Some computers allow a portion of working memory to be replaced by a read-only memory (ROM). The use of an ROM provides more rapid access to the information it contains since the normal restore operation is no longer required. The information stored in this section of memory is non-destructable in the sense that it cannot be altered dynamically by the system. Information is initially entered by the manufacturer by hard-wire connection or other means so that later changes or modifications may be somewhat difficult to achieve.

Read-only memories may also be utilized as a system bootstrap so that system regeneration or a cold start is simplified. Since this feature is not offered on all computers, it should be specified only when required.

2.5 Addressing

Addressing is a relatively complex problem for the computer since the address field of the instruction word is limited. Paging schemes are sometimes used to divide working memory into units that are convenient to work with (e.g., 256-2048 words/page). Techniques to manipulate data within core memory include direct and indirect addressing, relative addressing, indexing, and the use of variable base registers (VBR's). Each technique has its unique advantages and penalties, usually trading execution time for cost.

2.5.1 Direct Addressing

Specify the number of words of working memory that must be directly addressable. This figure will usually be smaller than the working memory size specified and depend upon the architecture of the CPU.

2.5.2 Indirect Addressing

Indirect addressing is supplied by most manufacturers and is highly desirable. A single-level indirect address is the address of a **memory** location that contains the address of an operand (e. g., an instruction word) rather than the operand itself. Indirect addressing extends the address capability of direct addressing, but requires additional time for execution. **Multilevel** indirect addressing provides the capability of specifying **additional** addresses which are also indirect. If multiple level indirect addressing is required, the specification should so state.

2.5.3 Relative Addressing

Relative addressing permits relocation by relative position rather than absolute. Two **common** techniques are page relative and program counter relative. In the former, the page boundaries are fixed and addressing is direct within the page. Indirect addressing is used to cross page boundaries. **The** latter method sets up a dynamic page boundary that extends a half page before and after the address in the program counter. The boundaries move with the program counter. Both methods require indirect addressing to leave the current page.

2.5.4 Indexing

Indexing provides a powerful tool for address modification and is normally offered by **computer** manufacturers. A distinction should be drawn between the various types of indexing such as memory indexing, shared indexing, dedicated indexing, and extension registers. In memory

indexing, (sometimes called auto-indexing which adds one to the contents of the indexer and executes that location), specific memory locations whose contents are used for indexing are set aside. This method requires additional time and memory since a memory reference is required to load or change the index. Another common method, shared indexing, is provided by the B-Accumulator (or accumulator extension) where the B-Register is used for index and multiply, divide, and double length operations. The shared operation prohibits easy indexing during any of the shared operations and requires a good deal more house-keeping on the part of the programmer. Dedicated indexing is the most desirable method whereby one, or more, registers are set aside solely for indexing tasks. When the programmer specifies an indexed address, the contents of the index register are added to, or subtracted from, the contents of the instruction register to generate the operand address. Extension registers, or variable base registers are not true indexers, but are used for address modification to address locations outside the current page.

2.5.5 General Purpose Registers

Some machines utilize general purpose registers for arithmetic operations, data manipulation, indexing, increment/decrement, and temporary storage. These registers typically increase the efficiency of the computer regarding storage requirements and the time required to execute a given functional operation.

2.6 Interrupts

In a process control system where multiple tasks are being handled, and are determined by events external to the computer, several needs arise. The computer must be aware of the occurrence of the events as they happen, so that it can tell what demands will be made on its time. A method should exist to make sure that the central processor in the system is always attending to the most

important task. A priority interrupt system should be specified to fulfill these needs.

Hardware priority interrupts normally cause program control to transfer to a specified memory location associated with each interrupt. The specified instruction provides a pointer to the service routine for the interrupting device and saves a return address to re-enter the program.

There are several basic types of priority interrupts. The single level indicator method is the simplest and the least expensive. All interrupt lines develop an OR output, which serves as the interrupt request signal to the central processor. At the completion of the current instruction, the interrupt signal causes the program counter to be stored and an interrupt processing subroutine to be entered. This subroutine tests each interrupt line in sequence to determine which request caused the interrupt. Either by program or automatically, the interrupt line recognized is reset and program control is transferred to the correct routine. The response time and overhead for this method are both high due to the number of program steps that must be executed before the central processor services the request. Although this method tests the highest priority level first, the overhead may make the lower levels ineffective. This method is single-level since no interrupt can be recognized while one is being processed.

Another method of priority interrupt utilizes more hardware, thereby costing more initially. However, because the operational overhead is much lower the efficiency may more than offset this cost. A flip-flop matrix is used to provide the memory necessary to determine the current status of an interrupt level. Each level is assigned a priority either by hardware or with software. When an interrupt occurs, the current instruction will be completed, then the central processor will transfer control to the memory location associated with the interrupt level.

Some interrupt systems of this type may be individually enabled and disabled by channel, and priority assigned and reassigned by software dynamically.

2.6.1 Number of Levels

The specification should require the number of levels deemed necessary plus one, or more, spares to allow for additions or modifications. Priority levels are usually expandable in groups of 4, 8, 16, etc. Future expansion requirements should also be specified.

2.6.2 Response Time

The response time starts with the occurrence of an external interrupt signal and terminates with the start of execution of the first useful instruction requested by the interrupt. This includes the time required to finish the instruction being executed, storage of active registers, and addressing the first instruction of the service routine. Typical response times are 10 to 200 microseconds depending upon the CPU hardware configurations and programming methods used. The maximum response time acceptable under qualified conditions should be stated.

2.6.3 Priority Structure

The interrupt structure should be specified. This includes the hardware hierarchy and architecture and the software control for **enable/disable** and level assignment.

2.7 Direct Memory Access

Hardware which permits access to main memory without going through the central processor arithmetic unit can be purchased for most computers. Access is gained through an independent memory port. Normally this option is required where very high speed transfer to and from memory is required. Typical data transfer rates occur at the memory cycle speed. This option is often a required prerequisite for addition of a disk memory.

2.8 Clocks

Two basic clocks may be specified, either the real-time clock, or the interval timer, or both. Resolution and frequency should be considered. It may also be desirable to include a specified drift tolerance. The interval timer is useful for timing process control operations whereas the real time clock is better suited to chronological event keeping and housekeeping functions.

2.8.1 Real-Time Clock

A real-time clock can be a simple pulse source at a given frequency, usually a 60 Hz synchronized power source that operates through the priority interrupts. It may also include a digital register that keeps track of the time of day.

2.8.2 Interval Timers

Interval timers are usually more sophisticated than time-of-day clocks although they often include the capability for input from a real-time clock. Interval timers usually consist of a crystal controlled clock driving one or more programmed registers. The registers can be loaded with a preset value under program control and subsequently incremented or decremented by the clock. When the register reaches zero an external interrupt is generated. Several frequencies (divisions of the basic clock) are usually available and should be specified where required for varying degrees of resolution. The degree of program control varies widely and should also be specified by the inclusion of desired instructions. Loading, on/off control, and frequency are examples of features normally available.

2.9 Stall Alarm

Real-time applications involving process control often require detection of computer stalls or looping endlessly so that corrective action can be taken. The stall alarm feature implements

this safeguard. Some machines detect the failure to change the program counter in a fixed number of machine cycles. Others provide an instruction that updates an integrating one-shot (sometimes called a "watch dog" timer) that will time-out if it is not periodically updated at a specified interval through program control. The specific application will dictate the type of stall alarm that is required. The corrective action which should occur as a result of a stall alarm should be specified along with the stall alarm, e.g. audio alarm, mass-memory to core memory data transfer or halt. This is usually a software interrupt service routine.

2.10 Power Failure/Restart

This option allows detection of primary power loss, and automatic re-initialization of the program after power is re-established. When a power loss is detected, the operating program is interrupted and either with software and/or hardware the pertinent registers are stored in memory and the processor is halted to await the re-establishment of power. With automatic restart, the processor is restarted without operator intervention, and software and/or hardware restores from memory the registers which were saved when the power failure was detected. Without automatic restart, the registers are saved in response to the low power condition, but operator intervention is required to restart the processor. It is important in real-time applications, where an executive is employed which disables the interrupts, that it be ascertained whether or not the computer will be able to respond to the low power condition in time to store registers before the complete loss of power. To this end, the minimum time from detection of the low power condition (by external hardware) to the time when the processor will no longer function should be specified (e.g., 2 ms). This time should be closely compared with the maximum time that interrupts are disabled by the executive.

Since the computer will have no way to **determine** how long it was in a halted condition prior to restart, it **may** be necessary in a real-time system to provide a battery powered clock that is activated by a power failure. **When** power is restored, this clock is read, reset and turned off. **Internal** computer time bases **may** thus be reset and **normal** operation resumed.

2.11 Operators Console

It is usually desirable to specify certain features on the operators console depending on the application. For real-time systems key lockout of console controls is often desirable. Key lockout prevents an accidental shutdown of the machine or accidental entry of data from the console switches. Where **programmer** access to certain registers is required, it should be specified. **Sense** switch requirements and special display features (**e.g.**, run/halt and register lamps) should also be specified.

2.12 Input/Output

The **input/output** (I/O) capability of a computer **system** is especially important in real-time applications. Two principal modes of I/O transfer are available, direct memory access (**DMA**) and I/O bus transfers. **DMA**, as discussed earlier, provides high speed transfers at the **memory** cycle speed. Practically all computers use the "party-line" I/O bus structure as a **standard** item. Here, all peripheral devices share a **common input/output data bus**. There is an intermediate speed transfer **system**, sometimes called a data channel, available for some machines. It is usually significantly faster than standard **I/O** bus transfers requiring some 3-5 **memory** cycles. While not as fast as **DMA** (1 **memory**, cycle), it is very inexpensive. The **specification** should reflect the **minimum** acceptable I/O transfer rate for the **DMA** or data channel, whichever is required. Other items which can be specified include (1) block transfer which provides the transfer of large groups of data without program interference, (2) automatic power turn on/off for selected peripherals,

especially tape and card punches and readers, (3) program sensed flags for timing and simplified I/O which eliminate the need to build software timers. Other desirable features such as the ability to status test a peripheral are not as easy to specify, but are determined by the design and operation of the basic instruction set.

2.12.1 I/O Parity

Input/output parity hardware is required in some critical applications where undetected I/O errors can cause unacceptable process control. Each time an I/O transfer is executed a hardware parity test is performed such that if an error is detected a program interrupt is generated. This feature normally requires memory parity since an extra bit is added to the input/output word.

2.12.2 I/O Addresses

There are a limited number of external devices (peripherals) that can be connected to and be under program control of a computer. This number is usually specified by the number of device addresses (or bits of identification) available. Each manufacturer assigns certain addresses or device numbers (both logical and hardware) to his standard peripherals. It is important that the specification require the manufacturer to tabulate these assignments in the bid response so that the user does not inadvertently assign a device number to a process peripheral and then later purchase a manufacturer supplied peripheral with the same assignment. This is especially important with regard to the usability of standard software packages. The expansion requirements of the system should be anticipated to ensure that the unused device addresses are sufficient for future requirements.

2.12.3 Signal Types

The types of signals nonnally accommodated in digital I/O facilities are (1) voltage and (2) contact closure. While only binary conditions are sensed, two logic possibilities arise. These possibilities are as follows:

1. Voltage
 - a. Positive True
 - b. Negative True
2. Contact
 - a. Closure True
 - b. Open True

It may not always be necessary to specify the conditions required, but a definition of each **manufacturer's** usage should be a required part of the bid response for interfacing purposes. The conditions required would be necessary to interface the computer to an already designed device or process.

2.12.4 Capacity

The capacity of the digital facility is usually specified by the number of input or output groups required where each group is composed of n bits equal to one word. Current drive, or sink, capacity should also be specified for voltage output terminals.

2.12.5 Signal Filtering

The binary signals applied to a digital input device are often noisy (contact bounce or erratic voltage). If necessary this noise can be filtered. Specify the type of filter and the time constant.

2.12.6 Speed

Transfer rates may be specified, but they will be limited by I/O capability as defined earlier. It is usually more appropriate to specify contact opening and closing times and voltage rise and fall times under full load conditions.

3.0 PERIPHERALS

Specifications for peripherals are heavily dependent on the specific application and the type of equipment available from the respective computer manufacturers. An outline of points to consider for each peripheral device is given below as a checklist of the major points to remember when specifying each type of peripheral. A few comments are included here with general applicability to all peripherals.

High speed peripheral devices (bulk memory, etc.) requiring high data transfer rates should be purchased with block transfer hardware as an integral part of the interface. This avoids the difficult (and **sometimes** nearly impossible) task of servicing this type of peripheral device by highly sophisticated software routines. This requirement does not apply to slower devices like paper tape equipment and tele-types.

Some peripherals will not operate properly when they are situated too far from the central processor. If the equipment configuration demands the flexibility of locating peripherals more than 30 feet from the central processor, this dimension should be specified to insure proper **equipment** operation responsibility.

Most modern peripheral interface designs provide for automatic mechanical equipment turn on and turn off to avoid the necessity of leaving equipment running in anticipation of use. Where desirable, the automatic turn **on/turn** off feature can be specified.

The number of priority interrupts required for each peripheral device should be stated by the manufacturer in the bid response.

3.1 Digital-to-Analog Converters

Analog output is provided by a class of instruments called Digital-to-Analog Converters (DAC's). DAC's are supplied by most computer manufacturers as standard peripherals.

3.1.1 Number of Channels

Specify **the** number of DAC channels required. A distinction should be made as to whether multiplexed outputs

are suitable. Some applications such as hybrid computers cannot readily utilize multiplexed DAC's because of output sag.

3.1.2 Output Range

There are two general types of DAC, voltage output and current output. For the voltage output DAC specify the full scale output voltage range, the current driving capacity, and the maximum acceptable source impedance. Current output type DAC's usually have a "live zero" and an output range of 1-5 ma, 4-20 ma, or 10-50 ma, corresponding to 0 to full scale. The maximum load impedance should also be specified for a current output DAC.

3.1.3 Data Input

Specify the number of bits to be used as the digital input signal. Specify whether full digital buffering is required. Most manufacturers offer a buffer register for each input channel. Where buffering is not specified, the output sag may be excessive.

3.1.4 Accuracy, Linearity, and Monotonicity

Accuracy is specified in **terms** of percent of full scale and is **determined** by the number of bits selected and the accuracies of the components of the DAC. Overall accuracy is **usually** specified. Linearity is specified in terms of deviation from a straight line through zero and full scale (+ or -). Monotonicity relates to an increasing output value for every increasing input code. The DAC **must** be capable of providing every output value corresponding to every input value without dropping codes, or **losing** codes.

3.1.5 Output Sag

Specify output sag in **terms** of volts/second (e.g., 1 millivolt/microsecond).

3.1.6 Slew Rate

Slew rate is a measure of the full scale frequency response for step change inputs. It may be specified in terms of volts/second, or more conveniently in terms of the output rise time for a full scale step input.

3.1.7 Settling Time

While slew rate specifies the large signal gross response, settling time specifies the response time required to reach a given accuracy and is specified by a time to reach a given percentage error (e.g., 10 microseconds to reach .1% of the final value).

3.1.8 Short Circuit Capability

All outputs should be specified short circuit proof.

3.2 Analog Multiplexer

Analog input systems include multiplexers (MUX's), sample and hold amplifiers, and analog-to-digital converters (ADC's)⁷. In complex systems, several MUX's and ADC's may be required to fully satisfy system requirements due to the mix of different input signals, different sample rates, etc. In cases where a large mix of input signals and sample rates are required it is useful to provide a table of requirements defining the signal types by (1) sample rate, (2) signal range, and (3) number of signals. The following specifications can then be related to specific sensor groups.

3.2.1 Signal Conditioners

Most systems will require signal conditioners to be used on the process analog signal lines. The signal conditioners may include noise filters, thermocouple reference junctions, strain gauge bridge networks, transducer excitation sources and preamplification. Each application must be appraised to determine the type and number of conditioners required. Process characteristics such as environmental electrical noise (its frequency and

magnitude), common mode voltages, type of **primary** power source and any significant magnetic fields should be considered.

3.2.2 Input Switches

It may be advisable in special cases to specify acceptable switch types (e.g., reed relay, bipolar transistor, field effect transistor, etc.). If Metal-Oxide Semiconductor Field Effect Transistors (MOSFET's) are allowed, specify protection **requirements** to limit failures induced by overvoltage. MOSFET devices are susceptible to damage from induced charge and can be protected with diodes and/or other devices.

3.2.3 Number of Channels

Specify the number of channels required, by signal type. Expansion requirements should be specified.

3.2.4 Input Configuration

Specify the type of input configuration required. Single ended, differential, and differential-guarded should be considered. For differential-guarded systems specify whether or not the guard must be switched. In addition, the minimum number of levels of **sub-commutation** should be specified. Limitations in input isolation and access speed will influence this choice. Usually at least ~~two~~ levels are specified except in simple systems.

3.2.5 Sampling Rate

Specify the maximum rate required. Where more than one group of signal types are used, specify maximum rate by signal type. For example, 100 high level signals (0-5V) should be scanned, digitized, and input to the computer at a rate of 1000 **points/sec.**

3.2.6 Input Impedance

Specify the input impedance of "off" channels. The decision here will relate to errors caused by a number of "off" channels in parallel with an "on" channel.

3.2.7 Full Scale Input Signal

Specified in terms of the full scale signal parameter.

3.2.8 Crosstalk

Crosstalk is usually specified as a percent of full scale for any channel with a 100% overload on an adjacent (physically) channel.

3.2.9 Scatter

Scatter relates to the uniformity of all **channels** and is specified by channel-to-channel deviation in terms of percent full scale.

3.2.10 Common Mode Rejection

Specify for balanced and unbalanced sources. The range from d.c. to 60 Hertz is usually specified.

3.2.11 Maximum Common Mode Voltage

Specify the maximum common mode voltage to be tolerated without damage. For FET switches the maximum voltage condition should not turn on de-selected channels.

3.2.12 Address Modes

Specify the address modes required. Usually, three or four are available including (1) Random, (2) Sequential, (3) Dwell, and (4) Manual. Random address allows the computer to access any channel at random. The minimum time required between the address execution and channel selected should be specified since not all MUX's are designed for random access at the highest sampling rate. Sequential mode permits the ordered sampling of a fixed sequence of channels without **external** interference once the initial start signal is given. Dwell is used to

sample continuously a single channel. Manual permits local operation for maintenance operations. Local (front panel) address display can be specified to aid maintenance and is a recommended option.

3.2.13 Sample and Hold Amplifier

The sample and hold may not be a required part of a system depending primarily on the input data frequency.

3.2.14 Acquisition Time - Sample and Hold

Acquisition time is the time required to acquire and track a step input response.

3.2.15 Aperture - Sample and Hold

Aperture represents the disconnect time when changing from track to hold and is influenced by the input data frequency response.

3.2.16 Sag - Sample and Hold

Sag, or output decay should be specified and will limit the accuracy of the a/d conversion. Sag should typically be less than 1/10 the value of the least significant bit for the duration of one a/d conversion cycle.

3.3 Analog-to-Digital Converter

Process signal noise, conversion accuracy, and speed of analog conversion are considerations which often determine the type of analog-to-digital (ADC) used. The conventional type ADC uses successive approximation techniques. They typically have 10 to 14 bit resolution (one part in 1024 to one part in 16,384) and convert in 5-20 microseconds. Oftentimes ADC's of the integrating type (sometimes called integrating DVM's) will give superior accuracy and noise rejection compared to the successive approximation type. Integrating ADC's have resolutions up to one part in 10^6 , but require up to one second to convert. This long conversion time is too slow for many process control applications.

3.3.1 Word Length

Specify by number of bits (i.e., 10, 11, 12, etc.) including the designation of the most significant bit (msb) as a sign bit (where applicable).

3.3.2 Output Code

Output code does not usually represent a problem when the converter and computer interface are supplied by the computer seller. In either case the code (e.g. 2's complement, 1's complement, etc.) should be specified to be compatible with the basic computer format. Two codes, binary and binary coded decimal (BCD) are in general use. When BCD is specified (usually for the integrating type ADC) also specify the bit weight, 1-2-4-8 or 1-2-2-4.

3.3.3 Conversion Speed

Specify the maximum time required for a complete conversion, or alternatively the maximum number of conversions per second (frequency).

3.3.4 Accuracy

Accuracy will be limited by the word length and other factors. It is specified as a percent of full scale output plus or minus one-half the least significant bit (e.g., $\pm 0.05 \text{ f.s.} \pm 1/2 \text{ lsb}$).

3.3.5 Aperture Time

The aperture time is that period during which the analog signal is sampled prior to holding for conversion to a digital output. Typical times are ten to 100 nanoseconds. The time specified should be determined based upon the signal frequencies being sampled, i.e. higher frequencies require a shorter aperture to minimize skew.

3.3.6 Linearity

Linearity may be specified as the maximum deviation from:
(1) the best straight line through the full range (called

best fit linearity), (2) a straight line passing through zero (called zero-based linearity), and (3) a straight line through both zero and full scale (called terminal linearity). Specify the ~~maximum~~ permissable deviation and the measurement reference.

3.3.7 Overscale Indication

Usually an option, overscale indication can be specified. ~~The~~ overscale flag, or bit, operates ~~when~~ the input exceeds the full scale range by the value of the least significant bit (1sb).

3.3.8 Display

Front panel display of the output code provides a maintenance aid for system trouble-shooting and calibration.

3.3.9 Self-Test and Automatic Calibration

~~Some~~ manufacturers offer optional aids for self-testing and ~~automatic~~ calibration. The characteristics tested include linearity, speed and accuracy. The techniques available should be surveyed before specifying, as many different approaches are used.

3.4 Magnetic Rotating Bulk Storage

Where storage requirements exceed 10^5 bits and medium speed **access** (millisecond response) is desired, some form of magnetic rotating storage is usually specified. Both drums and disks are available as standard peripherals for this task. Disks are subdivided into moving head and head per track configurations. Most real-time processes will require the head per track configuration with its higher reliability and faster access.

3.4.1 Capacity

Capacity is usually specified by the total number of bits stored. ~~Some~~ systems provide a breakdown of a specified number of fixed length words (e.g., 250K, 24 bit words).

3.4.2 Access Time

The access time is a combination of the head positioning time (for moving head disks) and the time required for the rotating surface to move into position under the head. Worst case is one revolution. Average access time (1/2 revolution) is often specified.

3.4.3 Transfer Rate

Specified in bits per second, transfer rate is the rate at which data is removed after it is accessed.

3.4.4 Error Control - Reliability

Error control is performed to a varying degree by different manufacturers by parity generation and error detection. The specification should clearly state the degree of error control required. Most current systems do not use sophisticated error detection schemes. They depend on the inherent ~~system~~ reliability and minimal detection schemes. Nominal error rates are about 1 in 10^{10} bit transfers.

3.4.5 Write Lock

Write lock can be specified for selected portions of the drum or disk to prevent writing on the selected portion. Selection is usually by manual switch on the disk or drum which prohibits addressing the specified memory location.

3.5 Magnetic Tape Bulk Storage

Magnetic tape bulk storage is widely used for storage in the range of 10^6 to 10^9 bits. It features low cost, high density storage. Two principal approaches to recording, continuous and incremental, are in widespread use. The major differences are price (incremental recorders are usually lower in cost) and performance (continuous recorders are usually better performers). The specification of the type of tape unit should be based upon the

required usage. Incremental tape units ~~may~~ be **used** in applications where input data is received at a variable rate (**e.g.** keyboards, alarm logging, telemetry) but must be recorded at regular density and format for computer input or synchronous transmission. It ~~may~~ thus function as an effective data rate-changer.

Small, relatively inexpensive, cassette type recorders are available for use with small **computers**. Data are recorded in **bit-serial format** by these devices on one or more selectable tracks per cassette. There is also the blocked-address type **format** wherein timing and identifying marks **are** pre-recorded on the magnetic tape which are used in writing and reading the tape. This feature allows **replacement** of blocks of data on tape in a random manner without altering other previously recorded information.

3.5.1 Standards

Present standards are ASA X3.22-1967 for 800 CPI, NRZI tape and the proposed standard for 200 CPI NRZI. Both standards refer to 9-track tape. A de facto standard exists for 7-track tape which is referred to as **IBM** compatible tape. Compatibility is required where tape is to be pre- or post-processed on another computer.

3.5.2 Operating Capability

Specify the operating capability (read only, write only, **read/write**, synchronous read/incremental write, etc.). ~~Many~~ combinations are available, one of which will fit each systems needs. The buyer should specify the **combination** based upon worst case data **read/write** rates and the total number of transports to be used. Continuous **type** transports generally have a controller capable of handling eight units.

3.5.3 Speed

Specify in inches per second for continuous operation and steps per second for incremental. The speed requirement is determined by calculating the maximum quantity of data to be transferred per unit time taking into consideration the packing density.

3.5.4 Packing Density

Specify in characters per inch (CPI). Standard densities are 200, 556, and 800 CPI. In the past, packing density was often specified in bits per inch (bpi) where bits implied characters. The American Standard Association (ASA) has recognized this defect and X3.22 uses CPI. The selection of 200, 556, or 800 CPI will depend on individual system requirements, but keep in mind that some batch systems will not accept 200 CPI tape without an additional tape packing pass. If a remote system is going to be used with the specified system, the choice of compatible density can save substantial processing costs.

3.5.5 Number of Tracks

Specify 7 or 9 track (channel) tape depending on system requirements. Mere again, compatibility should be considered. Virtually all large computers can handle 7 track tape, not all can process the newer 9 track tape.

3.5.6 Reel Size

Specify by reel diameter in inches. Seven inch (about 600 feet), 8 1/2 inch (about 1200 lineal feet) or 10 1/2 inch (about 2400 feet) reel sizes are available for most IBM compatible drives. Some drives, especially those built for 7 or 8 1/2 inch reels will not accept the larger reel.

3.5.7 File Protect

Specify this feature for tape decks where write lockout is desired to prevent accidental writing on the wrong tape.

3.5.8 Lateral Parity

Specify internal parity generation if required for process reliability and whether even or odd parity is to be used. Odd parity is most widely used for data acquisition and numeric data while even parity is usually used in text oriented systems. The special case of even parity and the character "0" cannot be adequately handled by some tape decks since "0," even parity is all tracks zero. For this case some manufacturers provide a zero detection circuit which records a BCD 10 for the "0" character. This requirement should be specified for systems utilizing even parity.

3.5.9 IBM - Compatible Gaps

Two types of gaps are used for IBM compatible tape. These are the inter-record gap and the file gap. Specific formats are defined for 7-track and 9-track tape. These formats are indicated in ASA Standards. Each gap and its associated check character(s) is usually inserted automatically by a single external command.

3.5.10 Load Point

Two load points should be specified, one for writing and one for reading. The write load point, when selected, automatically advances the tape to the beginning-of-tape (BOT) marker, erases 3.4 inches of tape and stops. The read load point, when selected, advances to the BOT marker and stops. Specify the type of control required for both functions. Both manual and automatic (program control) modes are available.

3.5.11 Tape Status Sensing

Specify the capability for sensing broken tape and the end-of-tape.

3.5.12 Rewind and Fast Forward

Specify in **terms** of rewind speed (**or** time) and fast forward speed.

3.5.13 Write Testing

Options are available to test the write circuits without re-reading the tape. The two most widely used are read-after-write and echo check. The method specified will depend on overall **system** reliability requirements.

Read-after-write is the most satisfactory test from a technical standpoint since the recorded data is read and compared with the input data. Echo check does not actually test the recorded data as it monitors the magnetic head current during the recording process to assure compliance with input data.

3.5.14 Start-Stop-Backspace

Most of the more expensive recorders will start and stop in the record gaps and **automatically** backspace one, or more, records. This feature, where required, should be specified since some of the inexpensive systems do not provide this capability.

3.5.15 Buffer and Word Assembly

The manner in which data is to be buffered between the CPU and the magnetic tape should be clearly defined.

3.6 High Speed Printers

Several different methods are available for high speed printing. These include impact, electrostatic, optical, and ink squirting devices. The specification should require a description of the printing technique used. Impact or mechanical type printers are subject to wear-out and require periodic and corrective

maintenance. The electrostatic or optical types have fewer moving parts, but are generally more expensive and have special paper requirements.

3.6.1 Print Speed

Specify in lines per minute or characters per second.

3.6.2 Number of Columns

Specify the **number** of print positions (horizontal).

3.6.3 Character Spacing

Specify the **number** of characters per inch for each line (horizontal) and the maximum cumulative character spacing error for the total line width.

3.6.4 Character Size

Specify the **nominal** character size (horizontal x vertical).

3.6.5 Character Registration

Specify the horizontal and vertical tolerance for registration. This will limit the waviness of the printed output in either direction.

3.6.6 Character Set

Specify the required character set. A table is usually convenient.

3.6.7 Character Replacement

Specify individual or group replacement requirements.

3.6.8 Line Spacing

Specify line-to-line registration requirements.

3.6.9 Paper Handling

Specify paper width requirements. Where two different **forms** must be handled simultaneously, specify the requirement. Specify the **maximum** number of sheets for multiple copy printing. Automatic reversal ribbon feeds, where applicable, can be specified. For **non-permanent** copy,

specify the fading time. The specification should also require the seller to indicate the types of special paper, if any, required by the printer and its characteristics such as storability, smudging. Both feed and takeup paper storage requirements should be specified.

3.6.10 Cabinet Soundproofing

Specify acceptable noise levels or the nominal thickness of sound deadening material required.

3.6.11 Buffering and Word Assembly

The manner in which data is to be buffered between the CPU and the printer should be clearly defined.

3.6.12 Code

Specify the code desired where compatibility may be a problem. Available codes include United States of America Standard Code for Information Interchange (USA-SCII), Extended Binary Coded Decimal Interchange Code (IBM-EBCDIC), etc. A sample code chart is shown in Figure 2.

3.6.13 Feed

Both sprocket feed and pressure feed are available. Sprocket feed is required where standard forms are to be used or where accurate paper indexing is desired. Automatic form feed is optionally available for fast indexing.

Row \ Column	0000	0001	0010	0011	0100	0101	0110	0111
0	0	1	2	3	4	5	6	7
0 0 0 0	NUL	DLE	SP	0	@	P		p
0 0 0 1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	STX	DC2	"	2	B	R	b	r
0 0 1 1	ETX	DC3	#	3	C	S	c	s
0 1 0 0	EOT	DC4	\$	4	D	T	d	t
0 1 0 1	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	ACK	SYN	&	6	F	V	f	v
0 1 1 1	BEL	ETB	'	7	G	W	g	w
1 0 0 0	BS	CAN	(	8	H	X	h	x
1 0 0 1	HT	EM	)	9	I	Y	i	y
1 0 1 0	LF	SUB	*	:	J	Z	j	z
1 0 1 1	VT	ESC	+	;	K	[	k	{
1 1 0 0	FF	FS	,	<	L	\	l	!
1 1 0 1	CR	GS	-	=	M	]	m	}
1 1 1 0	SO	RS	.	>	N	^	n	~
1 1 1 1	SI	US	/	?	O	_	o	DEL

FIGURE 2

USASCII 1967

3.7 Low Speed Printers

All of the specifications of high speed printers also apply to low speed. **Typewriters** and teletypes are representative low speed printing devices plus having keyboard input capability. The Teletype **Corp. Models 33 and 35** are the most widely used devices being offered by the majority of **computer** manufacturers.

3.7.1 Communication Channel

There are three basic types of **communication** channels: simplex, half duplex, and full duplex. A simplex line can carry data in only one direction, i.e. to the printer or from the keyboard. A half duplex line can carry data in either direction, printer or keyboard, but only one direction at a time. A full duplex line can carry data in **both** directions simultaneously such

that printer and keyboard function independently. Most manufacturers systems utilize half duplex. Specify the operating mode desired.

3.7.2 Heavy Duty/Light Duty

Specification of light or heavy duty performance is especially important concerning teletypes. The ASR/KSR-33 is a light duty device and is recommended for environments where usage is less than one or two hours per day. The ASR/KSR-35 is a heavy-duty device designed for continuous operation and is the most satisfactory for medium to heavy duty use.

3.8 Paper Tape Reader/Punch

The attractiveness of paper tape derives from its low initial cost simplicity and dependability. Because of these advantages, it has become very popular. Its drawbacks include low storage density, low speed data input and output, tape wearout characteristics, and substantial punch and reader operator and maintenance requirements.

3.8.1 Standards

Applicable paper tape standards are ASA X3.6-1965, X3.18-1967, X3.19-1967, and X3.20-1967 Interchange.

3.8.2 Speed

Specify read/punch speeds in characters per second.

3.8.3 Channel Capacity

Most computers use 1 inch tape with a capacity of eight channels,

3.8.4 Directional Capability (Readers)

Most readers are unidirectional, therefore one must specify if bi-directional reading is required.

3.8.5 Malfunction Sensing

Specify requirements for broken tape and/or end of tape sensing.

3.8.6 Error Checking (Punch)

Some punches are capable of checking the actual punched character by sensing whether or not the individual punches are extended. This should be specified if desired.

3.8.7 Tape Requirements

Restrictions on physical tape characteristics are imposed by all manufacturers. Some of these restrictions lead to incompatibility. In order to avoid some of these problems, require each bidder to supply a list of suitable tape characteristics. The list should include (1) tape **composition** restrictions (e.g., mylar, dry paper, lubricated paper, etc.), (2) color restrictions (black, grey, etc.), (3) transmissivity restrictions, (4) dimensional **limits** (width, thickness), and (5) tape form restrictions (fan-fold, spool).

3.8.8 Tape Handlers

The method of tape handling required should be specified. Fan-fold and spooling are available. Specify for both reader and punch. Where spooling is desired, specify (1) feed and rewind speed, (2) leader **requirements**, (3) malfunction sensing, (4) tape capacity, and (5) reel size. Specify fan-fold tape supply and take-up bins if required.

3.9 Card Readers and Punches

Cards are also a widely used **form** of computer compatible source and object data. They offer advantages over paper tape, but are subject to many of the same disadvantages such as poor packing density, sensitivity to environment, and they are subject to misuse (folding, mutilation).

3.9.1 Standards

Applicable standards include ~~ASA~~ X3.11-1966 and ASA X3.21-1967.

3.9.2 Speed

Specify read/punch speeds in cards per minute.

3.9.3 Feed Method

Specify the feed method (pinch roller, air pressure, etc.) where this is a requirement.

3.9.4 Reading Method

Row reading or column reading may be specified. If the card reader is purchased separately to be interfaced to an existing computer the column (direct character), a method would be easier to interface. However, if the reader were purchased with a system the interface would be provided thus making the method unimportant.

3.9.5 Hopper Capacity

Specify hopper capacity in cards.

3.9.6 Non-Standard Cards

Many non-standard card input techniques are available including mark-sense, porta-punching, and others. Special consideration should be given any non-standard requirements.

3.10 Cathode-Ray-Tube Displays

Cathode-ray-tube (CRT) display systems are becoming popular in many types of computer systems. Although most operating displays are black and white, color displays provide many advantages and should be considered. The trend in process control is toward color graphics. The most widely accepted systems utilize either beam steering, raster scanning, or storage tube display methods. The following items do not apply in every respect to all three display types, but do cover the important parameters for all three.

3.10.1 Display Area

Specify the usable viewing area for the display. Alternatively, the gross CRT dimensions can be specified. All specifications should be met over the usable viewing area.

A bonded safety shield should be specified. Clear or darkened shields are available depending on viewing requirements.

3.10.2 Phosphor

Where important, specify the phosphor type(e.g. P-31) to attain desired color, decay time, etc.

3.10.3 Color

If a color display is specified the fundamental colors (normally red, blue, and green) should be specified along with mixing capability.

3.10.4 Spot Size

Spot size is important in beam steered and storage tube displays and is specified in circular mils at the half-light points.

3.10.5 Random Positioning Time

Specify the need here. It is not generally available for raster systems. Specify by the maximum time required to position the beam anywhere on the screen and settle to within one spot width of the final position.

3.10.6 Positioning Repeatability

Specify repeatability in terms of percent of full scale, independent of the previous position.

3.10.7 Jitter

Specify in percent of full scale for a fixed period of time. Both short term and long term can be specified.

3.10.8 Resolution

Specify by the number of points or bits for both x and y. For raster systems specify the number of lines. Differentiate between television and optical resolution.

3.10.9 Contrast Ratio

Specify the black and white contrast ratio required under specified ambient lighting conditions.

3.10.10 Brightness

Specify the brightness (usually in foot-lamberts) under specified ambient lighting conditions.

3.10.11 Vector Data Format

Specify the required format for vectors. This should include long and short vectors, and relative and absolute location requirements.

3.10.12 Vector End Point Registration & Linearity

Specify requirements for end point registration and full screen linearity.

3.10.13 Vector Writing Rate

Specify the maximum time allowed for writing vectors qualified by length. Intensity should not be affected by length or orientation.

3.10.14 Character Set

Specify the character generator character set for alpha-numerics and special characters. Specify the maximum time for character generation independent of character type.

3.10.15 Character Sizes

Specify the number of selectable character sizes required and define each size. Include requirements for program control of character sizes. Specify the aspect ratio required for characters.

3.10.16 Number of Characters Displayed

Specify the number of flicker-free characters to be displayed at one time. Define the refresh rate for this requirement (e.g., 50 Hertz).

3.10.17 Intensity

Specify the number of programmable intensity levels to be supplied.

3.10.18 Phosphor Protection

Specify requirements for protection against phosphor burn-

3.10.19 Light Pens

Specify the light pen requirements in terms of speed of response, method of detecting a light pen hit, use of tracking crosses, spectral response of the sensor, resistance to ambient light triggering, and method of light pen initialization.

3.10.20 Alphanumeric Keyboard

Specify the keyboard requirements and key placement.

3.10.21 Functional Keyboards

Specify the functional keys required. Include programmed overlays where required. Define the requirements for engraving overlays with functional key names.

3.10.22 Joystick/Tracking Ball

Specify functional requirements.

3.10.23 Cursor

Specify the required cursor characteristics.

3.10.24 Memory

Specify the requirements for display memory that are not included in the main computer memory. Local refresh may be desired in the form of delay line, core, or disc memory. For storage tube displays include requirements for storage time and erasure time.

3.10.25 Interrupts

Clearly state the requirements for display interrupts. Include light pen, keyboard, and availability interrupts.

3.10.26 Enclosure

Specify operator requirements for control placement, work space at the console, tilt angle for CRT (if desired), etc.

3.11 Plotters

3.11.1 Bed Size

Specify the paper size to be handled. For continuous roll paper specify the width.

3.11.2 Accuracy and Linearity

Specify in terms of percent of full scale.

3.11.3 Resolution

For digital plotters specify the number of points per axis or the smallest resolvable increment. For analog plotters specify the smallest resolvable deflection in percent of full scale.

3.11.4 Paper Handling

Specify ~~the~~ paper handling method, paper hold down requirements and the storability of plots.

3.11.5 Alphanumeric

Specify whether the plotter must provide alphanumeric capability.

3.11.6 Point/Line Plots

Specify either continuous line plotting or point plotting, or both.

3.11.7 Pen Characteristics

Specify the number of pens to be **used** on a single plot (multiple color plotting).

3.11.8 Speed

Specify the plotting speed in points per second or inches per second.

3.11.9 Data Entry

Specify the data input format to be used.

3.11.10 Interface Requirements

The input **and** control signal requirements should be specified. Examples are I.C. logic compatible digital pulse response, **serial/parallel** data, **etc.**

3.12 Miscellaneous Output Equipment

There are other types of process data output devices available in addition to those described above. These include audio alarms, nixie type numerical indicators, electroluminescent panels, back-lighted or front-lighted slide projection with or without overlay, solid state matrix displays and laser displays. Large screen displays are **suitable** for outputting process data to a group of people covering a large area. Whereas, console displays are appropriate for a single operator or several people in the immediate area. Some of the more exotic displays are not yet practical from either a cost or functional standpoint for process control use. It is not within the scope of this guideline to go into detail discussion on these devices. Reference 2 covers their functions in sufficient detail to determine applicability.

4.0 SOFTWARE

Software systems vary in size from a minimum system which would be suitable for a basic minicomputer to very complex operating systems for a large computer configuration. This guideline will consider the minimum system (loader, editor, assembler, etc.), and then progress to include software required for larger scale systems. It is important that the hardware system specified be supported by an adequate software package, otherwise the development and application of the system will be inefficient.

Specific software to be furnished with the hardware specified should be defined. In addition, each bidder should be required to include a description of all software available (including extra price where applicable) for the specific configuration (core memory, peripherals, etc.) described in the hardware specification. All software provided should take advantage of hardware options such as multiply, divide and double precision.

4.1 Loaders

A loader is a program utilized by the computer to read into memory another program, routine, or various data. A bootstrap loader is a short routine sometimes keyed in through the console for the purpose of loading other relocatable or absolute programs. Some systems provide hardware bootstrap loaders which enable a rapid cold start without manual loading. Absolute programs are written such that absolute addresses are used internal in the program requiring loading in specific locations. Relocatable programs are written in a special manner enabling them to be located and executed in many areas of the memory.

Loaders should be available which load both absolute binary code and linking relocatable code produced by the symbolic assembler. Loaders which enable data entry from the ASR 33/35 and bulk storage devices, with provisions for other I/O devices on a device independent assignment basis should be specified.

4.2 Editor Program

An editor (correction) program enables corrections and additions to be made through the keyboard on source programs stored in memory by adding, changing or deleting lines of text. All modification, reading, punching, etc. is controlled by symbols typed at the keyboard. The result is a new source program which includes the corrected statements.

The editor may also be used for the creation of source text directly from the console teletypewriter keyboard. This is typically the method used on basic systems without a card reader.

4.3 Debugging Aid Program

Debug programs are designed to aid the programmer in checking out a particular program by furnishing him with the ability to examine and/or change the contents of selected areas of memory. On basic systems all operations are performed at the teletype with core memory used to store programs or program segments to be debugged.

Debugging aids for use with an on-line system having bulk storage and a control executive typically allow the user to:

- Modify, update, and relocate programs or portions thereof, and transfer programs to and from all bulk storage devices
- Execute programs under break-point control and examine any memory location by command from the keyboard
- Search for lines of text
- Dump selected portions of core memory on any bulk storage device in binary or symbolic format
- Insert or delete statements.

The manufacturer should be required to fully describe the capabilities of debugging aids.

4.4 Assembler

The assembler is a computer program that operates on symbolic input data to produce from such data, machine instructions by

carrying out such functions as: translation of symbolic-operation codes into computer-operating instructions, assigning locations in storage for successive instructions, or computation of absolute addresses from symbolic addresses. A basic assembler generally translates input symbolic codes into machine instructions, item for item, and produces as an output essentially the same number of instructions or constants that were defined in the input symbolic codes.

The assembler should provide pseudo instructions for the purpose of defining symbols, reserving memory, linking subroutines, and controlling I/O options. A pseudo instruction is a group of characters having the general form of a computer instruction, but not executed by the computer as an actual instruction. They are translated into machine language by the assembler.

The assembler may be one or two pass; meaning that the source program must be read and processed by the assembler program once or twice in order to create an object program. Literal (a word, number or symbol which defines itself) capability should be specified to allow for single and double precision in both fixed and floating point format, for hexadecimal values, and for alphanumeric character strings. The assembler, in conjunction with the loader, should provide automatic paging and program relocation, linking of programs and routines, subroutine calls, and input-output control instructions for peripheral devices.

The symbolic assembler typically has the following characteristics:

1. The basic assembler is capable of running in a minimum size core memory, e. g. 4,096 words, without a bulk storage device, and using the ASR 33/35 for all input and output.
2. An extended assembler is provided which may operate in a larger core size and is capable of using high speed bulk storage devices,

3. The extended assembler is usually capable of functioning with the executive routine which is used for calling programs in and out of core from bulk storage devices or any other I/O devices on a software device assignment basis.
4. The extended assembler usually has a provision for calling independent I/O handling routines for any device which may be interfaced to the machine at a future date.
5. The assembler should be capable of generating either absolute or relocatable programs which can be linked to the computer.

An advanced version of the assembler which is typically provided with a medium configuration should provide macro-instruction capability. A macro-instruction is a source language statement that is capable of generating a predetermined sequence of machine instructions. This one-for-many instruction results in compiler power with assembler efficiency. It permits segmentation of a large program permitting extensive program analysis to aid in debugging.

4.5 Subroutine Library

A subroutine is a set of instructions in machine code which direct the computer to carry out a well-defined mathematical or logical operation. A subroutine library is normally provided with the system for use by the assembler and/or compiler. Included would be subroutines such as sine, cosine, log, exponent, tangent, arctangent, and square root.

The following subroutines are typical of those which should be provided as re-entrant and capable of being called from all program levels:

- Fixed point and floating point add/subtract and multiply/divide in multiple precision
- Floating point square root, logarithms, exponentials, sine/cosine and arctangent
- Fixed to floating point conversion, and floating to fixed point conversion

- BCD to binary, and binary to BCD
- I/O subroutines for **communications** with teletypewriters.

If hardware multiply and divide is provided, the mathematical and utility subroutines should utilize those instructions to reduce execution times. Execution times for all available subroutines should be required with the manufacturers bid.

4.6 Utility Programs

A utility program is used to assist in the operation of the computer. They assist in housekeeping and program management. Utility programs include the following:

- a Dump Core Memory
- List Core Memory
 - Create Files
 - Transfer Files Between Peripherals
 - Program Segmentation (chaining)
 - System Generation (Instructions to the Monitor regarding hardware configuration).

Those utility routines desired should be specified.

4.7 Hardware Diagnostic and Maintenance Aids

Diagnostic aids are used to locate a hardware malfunction in the central processing unit and peripheral devices. A given computer system will typically have multiple tests each of which pertains to a given functional section (**e.g.** memory) or peripheral. The diagnostic will typically locate the failure down to a functional component thereby facilitating repair.

Representative diagnostic aids are listed below:

- Verification of complete **CPU** operation and all I/O interfaces supplied with the system and peripheral devices where possible
- Fault location of any malfunction in the system using diagnostic programs furnished
- Complete **written** instructions for the use of the diagnostics including step-by-step instructions for using the fault location programs, and source **program** listings with **comments**.

4.8 FORTRAN Compiler

A FORTRAN Compiler is a program which can accept a problem-solving procedure, written in English and Mathematical like statements, and convert it into the proper elementary machine instructions. FORTRAN (FORmula TRANslation) is an algebraic compiler designed primarily for the solution of engineering and scientific problems.

In process control applications it is desirable for the compiler to be capable of operating in a real-time environment and, as a minimum, meet the following requirements:

- Compliance with an ASA Standard such as X3.9-1966 FORTRAN
- Intermixing of FORTRAN statements and assembly language statements by macros or other suitable means
- Provision for re-entrant subroutines
- Provision for queuing and utilization of priority interrupts in real-time.

The number of passes required to obtain the compiler object program, and the core memory and peripheral support required should be requested if not specified. At compilation time, the following optional output should be available:

- Source listing
- Machine language object output
- List of subroutines called
- Storage requirements
- Mnemonic listing of object code generated
- Ability to execute without reloading object program.

4.9 Other Higher Level Languages

Other higher-level languages such as COBOL (COmmon Business Oriented Language), BASIC, ALGOL (ALGOrithmic Language) are sometimes available with the system. The manufacturer should be requested to supply the pertinent information on those available. As the names imply COBOL is a business language, BASIC is a conversational language for scientific computation, and ALGOL is an arithmetic language.

Since the manufacturers of general purpose computers suitable for process control do not generally provide all of these languages since they are business or scientific oriented, they should be specified only when a real need exists for their usage.

4.10 Executive System

The real-time executive system controls the execution of all programs that are used in the control or data acquisition system. This includes timing, deciding which programs shall share the hardware system simultaneously, handling **input/output** operations, servicing interrupts, handling error conditions and **communications** with the operator(s) .

Care should be exercised in the specification of the executive system to include those features which meet the particular requirements of the application. In **some** cases, the software requirements detail the machine configuration and processor options required.

An executive system oriented to on-line real-time processing should be specified. It may be either core or disc memory resident. It generally resides on disc memory, along with the processor and library routines, and utilizes the disc for scratch storage if necessary. The capability of **foreground-background** processing may also be desirable. Foreground-background processing permits real-time operation and control of **the** process on a time and event basis in the foreground, and low priority interruptable processing such as calculation, etc. in the background when real-time foreground operations are not occurring. The foreground capability should include provision for both resident (in-core) and non-resident (**e.g.** disc) real-time programs.

The executive system may be sophisticated enough to require the following equipment and processor configuration:.

1. Disc memory
2. Core memory protection
3. Automatic priority interrupt
4. Programmable interrupt priority structure with nesting capabilities
5. Magnetic tape bulk storage.

Both the monitor and real-time programs should be memory protected against inadvertent destruction by a background program. The monitor should include automatic interrupt, context switching and storing, programmable interrupt priority structure, nested interrupt queries, program priority queries, and linkages and facilities for handling interval timers and non-standard peripheral equipment that may be supplied by the buyer. Device-independent I/O programming should be employed in the monitor.

The bidder should be required to state the maximum time that the monitor disables interrupts. This time should be evaluated on the basis of system saturation. To be effective, the executive must permit the interrupt system to react quickly to capture time-critical data or control an important process variable, and operate efficiently so that noncritical tasks can be handled as well.

A system I/O handler program within the executive orders and schedules the use of all system I/O in such a way that any I/O device is made available to any system program through I/O calls. System programs should not have to contain separate subroutines to share the use of common I/O.

4.10.1 Device Independence

The executive should have the ability to handle any I/O device on the system on a device (hardware) independent basis, i.e., it should be possible to re-configure the system hardware without affecting the existing operating programs.

4.10.2 Speed

The speed in which all I/O devices on the interrupt can be handled by the real-time executive should be specified.

4.10.3 On-Line Control

The real-time executive should provide operator/computer interaction through the console teletype. The software should allow the operator or programmer to perform a variety of functions such as dump tables, modify tables, load tape to core, etc. The process is normally controlled through the means of an operator's panel having function switches and buttons. The software must provide on-line interaction between the operator and the computer to effectively use the control panel.

4.10.4 High Speed Bulk Storage

It is desirable for the real-time executive to provide the following high speed bulk storage handling functions:

- Indexed storage of system and user programs, data sets created by the system and the user programs, and core images
- Loading user programs, system programs and core images into core for execution.

4.10.5 Low Speed Bulk Storage

It is desirable for the real-time executive to include the following low speed bulk storage handling functions:

- The ability to store a complete image of the high speed bulk storage device on the low speed bulk storage device (magnetic tape).
- The above image should be callable from a short bootstrap program (20 words or less) to restore the high speed bulk storage device to on-line service from a cold start.

4.11 Benchmark Problems

Computers manufactured by different firms possess different design characteristics. For example, they will have different memory speeds, instruction sets, instruction execution times, and input/output characteristics. These differences often make it difficult to evaluate or compare one computer relative to another for a given application. One method that is sometimes valuable in reducing these differences to a common denominator is the use of benchmark testing.³ The buyer may choose to include the requirement in the specification that a benchmark problem (data supplied by the buyer) be run on the bidders proposed system. The results of this benchmark (speed, storage requirements, etc.) would then be required with the bid.

The benchmark problem should be a typical operation that the computer would perform in the actual application. The frequencies with which different types of instructions are used vary widely in different applications. Another quality that is measured by benchmark testing is the efficiency of the software system. The assemblers, compilers and executive routines may be evaluated.

In addition to indicating overall computer performance the benchmark also yields information on the ease or difficulty of programming and operating each system.

4.12 Software Documentation

All software provided with the computer system should be fully documented. The documentation should be complete, accurate (i.e. for the latest version of the software provided) and suitable for both the beginning programmer and the experienced. Software summaries and library indices are useful to the user for rapid location of the material and operating instructions desired.

The most basic systems will provide programs on paper tape, whereas advanced systems use cards, magnetic tape or disc. The instruction manual for each program package discussed in section 4 of this

guideline should include operating instructions as well as descriptive information about the program. Flow diagrams are useful as an aid in understanding program logic and design. Source listings are desirable if the user expects to modify or rewrite system software. For example, a user may want to change the real-time executive to provide additional capability or unique features.

5.0 MISCELLANEOUS HARDWARE SPECIFICATIONS

5.1 Environmental

Specify the temperature and humidity requirements. Air conditioning requirements will be derived from these requirements.

5.2 Power

Specify input power requirements. It will be very useful to require a descriptive literature response to this section. The response should include:

1. Power required - by voltage, current, phase
2. Power dissipation in BTU/hr and KVA
3. Connector wiring
4. Protection from line voltage fluctuations such as a solo transformer.

Each items to be supplied for the central processor and each peripheral device specified and for the total system.

5.3 Enclosures and Cabinet Configuration

Specify cabinetry requirements including radio frequency interference (r .f.i.) requirements. Descriptive literature which describes the following items should be required:

1. Cabinet configuration and mechanical position assignment for each peripheral device and for each portion of the central processor with unused rack space identified and dimensioned.
2. Cabinet rack width (e.g. standard 19 inch).
3. Service clearances required for all racks and cabinets.
4. Installation dimensions for assessing doorway, elevator, and loading dock clearances.
5. Total system weight and weight by cabinet. Shipping weight.
6. Peripheral cabling limitations.

5.4 Spare Parts

Specify the inclusion of extender circuit boards for each type of circuit card used (where applicable). Require the offeror to quote a complete itemized list of all **recommended** spare parts with prices according to USA manufacturer's EIA part number.

5.5 Special Tools

All special tools (not **normally** found in an electronic technician's tool box) required to wire, connect, disconnect, assemble, and disassemble the unit of its components for maintenance should be provided.

5.6 Documentation

The Vendor should be required to supply the following instruction manuals and **information**:

1. Complete instruction manuals for software (with source listings)
2. Complete operating and maintenance manuals which include the following:
 - a. Complete schematics of all circuits supplied:
 - Logic diagrams
 - Schematics and specifications of all logical blocks
 - Schematics of all front **panel** switches, lamps, **etc.**
 - Schematics of all electrical **power** supplies and wiring in the computer cabinet
 - All schematics shall have correct manufacturer's part numbers for components listed.
 - b. Theory of circuit operation
 - c. Complete list of replacement components and pieces including ordering **information** (i .e. , component manufacturer's own part number).

5.7 Reliability

Obtaining **information from** a **computer** manufacturer relative to the reliability of his computer **equipment** is **at** best a difficult task, and obtaining meaningful reliability figures for a total system is **almost** an impossible task. A few simple concepts are suggested in this area. A computer system is no more reliable than its weakest link if that link is involved in the primary functions of the system, These weak links **should** be identified before the system is specified and a requirement for their reliability should be included in the specification. Since computer technology

changes so rapidly it is difficult to predict in advance the portions of a **system** which might be unreliable. It is sometimes helpful to check with users of the same **equipment** for this information.

In general, unreliability is more of a problem in peripherals, new equipment and **one-of-a-kind designs**.

Six ways in which system reliability might be specified are listed below:

1. Exclusions of certain types of equipments or certain manufacturers known to be unreliable.
2. A requirement for mean-time-to-failure on individual peripherals and system components. This is a mathematical formulation of the likelihood of failure of the total unit based on the likelihood of failure of the individual components.
3. A requirement for percent availability on the total system. This is an indication of the portion of time one could expect to find the system available for operation, **e.g.**, not being repaired or **maintained**.
4. Redundancy for some weak peripherals may be specified. It is desirable to have the computer sense the peripheral failure and select an alternate unit on line without need for manual intervention,
5. New equipment should be avoided if possible. Debugging a manufacturer's new design can be a very **frustrating, time** consuming and costly experience. If one must accept new equipment, require proof that the unit has been adequately tested and a workable system exists whereby unit modifications to correct deficiencies will be supplied promptly by the vendor without cost or inconvenience.
6. To obtain reliability in one-of-a-type designs specify the following when possible:

- The use of standard, generally available modules and components
- Conservative design procedures
- Operation over wide margins
- Quality control
- Design review by the buyer
- Detailed acceptance procedures which include stressed operation over a long period of time
- A guarantee.

5.8 Quality Assurance

Specify quality assurance requirements. Military specification MIL-Q-9858 "**Quality Control System Requirements**" dated April 9, 1959 may be used as a standard of measure.

5.9 Source Inspection Prior to Shipment

All components of the specification should be subject to verification and/or inspection at the Vendor's facility by a Buyer's representative at the point of final inspection. The Vendor should be required to supply data and **perform** tests as required by applicable drawings and/or specifications to verify that **specifications** are met.

5.10 Installation and Acceptance Test Procedure

The Vendor should be required to furnish ~~the~~ trained personnel to perform the installation and initial startup and checkout. The Vendor's installation personnel should perform all diagnostic tests after installation in **the** presence of Buyer personnel. Tests and inspections conducted by Buyer personnel will be the final basis of acceptance. Test procedures are to be signed off by Buyer personnel. An acceptance specification sufficiently detailed to reduce the possibility of a contract dispute should be made part of initial contractual specifications. Generally the specification itself will form the basis for the acceptance tests.

5.11 Warranty

The warranty period on computer system hardware is typically 90 days beginning on the first day of the successful performance period. The performance period follows the acceptance test period. The warranty should cover all maintenance and parts. The manufacturer should bear all costs for packing, equipment and parts transportation, and field service engineer travel, living and salary costs.

6.0 TRAINING COURSES AND MAINTENANCE

Provisions for training ~~some~~ buyer personnel in programming and maintenance of the computer system should be required as part of system purchase. Arrangements may be made at either the vendor's plant or at the buyer's site. The caliber of training should enable sufficiently experienced programmers and maintenance personnel to adequately work with the system. If possible, training should be scheduled in such a way that buyer personnel attending training courses at the vendor's plant follow the system through final checkout and acceptance at the vendor's plant. If the buyer is to write ~~some~~ of the software in conjunction with the vendor, then software training will have to begin well in advance of system delivery.

6.1 Software Course

Prior to the delivery of the computer system, the vendor should provide a comprehensive software training course covering all aspects of symbolic programming, FORTRAN and assembly language programming, and central processor operation.

6.2 Hardware Course

After the purchase of the computer system, the vendor should provide a comprehensive equipment maintenance course sufficient to train personnel, experienced in the field of electronic maintenance, to maintain the central processor. Additional training on peripheral maintenance should be available.

6.3 Maintenance

There are a number of possible types of maintenance contracts. The desired type should be specified in the invitation to bid or quotations on alternatives should be requested. Possible types of contracts include on-site or on-call coverage, total system maintenance responsibility, contracts for portions of the system, etc.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the review of the draft document and constructive suggestions of C. S. Lissner (ORNL), J. B. Bullock (ORNL), J. T. Meyer (LMEC), K. L. Gimmy (SR), E. A. Womack (AEC/RDT), J. R. Patterson (AEC/RDT), M. A. Rosen (AEC/RDT), G. W. blain (BNW) and G. E. Driver (BNW).

REFERENCES

1. R. E. Mahan. A State-of-the-Art Survey of Low-Level Multiplexers, BNWL-838, September 1968.
2. R. E. Mahan. A State-of-the-Art Survey of the Data Display Field, BNWL-725, May 1968.
3. J. R. Hillegass. Systematic Techniques For Computer Evaluation and Selection, Management Services, July-August 1969.

DISTRIBUTIONNo. of
Copies1 AEC Chicago Patent Group

G. H. Lee

215 AEC Division of Technical Information Extension35 AEC Division of Reactor Development and Technology

Director, RDT

Asst Dir for Nuclear Safety

Analysis & Evaluation Br, RDT:NS

Environmental & Sanitary Engrg Br, RDT:NS

Research & Development Br, RDT:NS

Asst Dir for Plant Engrg, RDT

Facilities Br, RDT:PE

Components Br, RDT:PE

Instrumentation & Control Br, RDT:PE (7)

Systems Engineering Br, RDT:PE

Asst Dir for Program Analysis, RDT

Asst Dir for Project Mgmt, RDT

Liquid Metals Projects Br, RDT:PM

FFTF Project Manager, RDT:PM (3)

Asst Dir for Reactor Engrg

Control Mechanisms Br, RDT:RE

Core Design Br, RDT:RE (2)

Fuel Engineering Br, RDT:RE

Fuel Handling Br, RDT:RE

Reactor Vessels Br, RDT:RE

Asst Dir for Reactor Tech

Coolant Chemistry Br, RDT:RT (2)

Fuels & Materials Br, RDT:RT

Reactor Physics Br, RDT:RT

Special Technology Br, RDT:RT

1 AEC Idaho Operations OfficeNuclear Technology Division

C. W. Bills, Director

1 AEC San Francisco Operations OfficeDirector, Reactor Division4 AEC Site Representatives

Argonne National Laboratory

Atoms International

Atomic Power Development Associates

General Electric Company

No. of
Copies

- 3 Argonne National Laboratory
 R. A. Jaross
 LMFBR Program Office, P. F. Gast
 R. W. Hyndman (EBR-II)
- 5 Atomics International
 L. E. Glasgow
 Liquid Metal Engrg Center
 R. W. Dickinson
 J. T. Meyer
- 1 Atomic Power Development Associates
 B. V. D. Farris
- 2 Babcock & Wilcox Co.
 Atomic Energy Division
 S. H. Esleeck
 Boiler Division
 T. P. Farrell
- 1 BNW Representative
 R. M. Fleishman (ZPR III)
- 1 Combustion Engineering
 1000 MWe Follow-On Study
 W. P. Staker, Project **Manager**
- 5 General Electric Co.
 Advanced Products Operation
 Karl Cohen (3)
 M. Temme
 W. Wagn^{er}
- 2 Idaho Nuclear Corporation
 D. R. deBoisblanc
 R. L. Heath
- 2 Oak Ridge National Laboratory
 G. S. Liss^{er}
 J. B. Bullock
- 2 Purdue University
 T. J. Williams
 G. Saridis

No. of
Copies

- 1 Savannah River
K. L. Gimmy
- 1 Stanford University
Nuclear Division
Division of Mechanical Engrg
Stanford, California 94305
R. Sher
- 6 Westinghouse Electric Corp.
Atomic Power Division
Advanced Reactor Systems
D. C. Spencer (5)
M. Heck

ONSITE - HANFORD

- 1 AEC Chicago Patent Group
R. K. Sharp (Richland)
- 4 AEC RDT Site Representative
P. G. Holsted (2)
L. R. Lucas
A. D. Toth
- 6 AEC Richland Operations Office
FFTF Project Office (2)
Engineering & Construction Div (1)
J. H. Crema
C. L. Robinson
Technical Information Library
F. J. Hughes
- 3 Battelle Memorial Institute
- 3 Douglas United Nuclear, Inc.
R. G. Lauer
C. F. Poor
DUN File
- 2 RDT Assistant Director for Pacific Northwest Programs

No. of
Copies69 Battelle-Northwest

L. E. Addison	T. A. Seim
F. W. Albaugh	N. B. Smith
R. J. Ausere	W. G. Spear
J. M. Batch	C. D. Swanson
R. D. Benham	G. R. Taylor
C. A. Bennett	G. G. Thieme
J. R. Boldt	G. L. Waldkoetter
R. A. Burnett	G. A. Worth
K. M. Busness	FFTF File (2)
P. D. Cohn	Technical Information Files (5)
C. R. Cole	Technical Publications (2)
M. R. Compton	
D. L. Condotta	
R. R. Cone	
V. L. Crow	
M. H. Deardorff	
P. J. Dionne	
B. E. Dozer	
G. E. Driver (2)	
M. D. Erickson (10)	
E. A. Eschbach	
C. D. Flowers	
H. P. Foote	
L. H. Gerhardstein	
R. S. Hammond	
D. W. Hartmann	
R. A. Harvey	
R. E. Heineman	
P. J. Hof	
J. A. Hubbard	
W. F. Lenzke	
C. W. Lindenmeier	
J. D. Lodge	
R. E. Mahan (2)	
G. W. Main	
W. R. Markillie	
W. R. McSpadden	
J. W. Mitchell	
W. H. Pazant	
W. D. Richmond	
J. T. Russell	