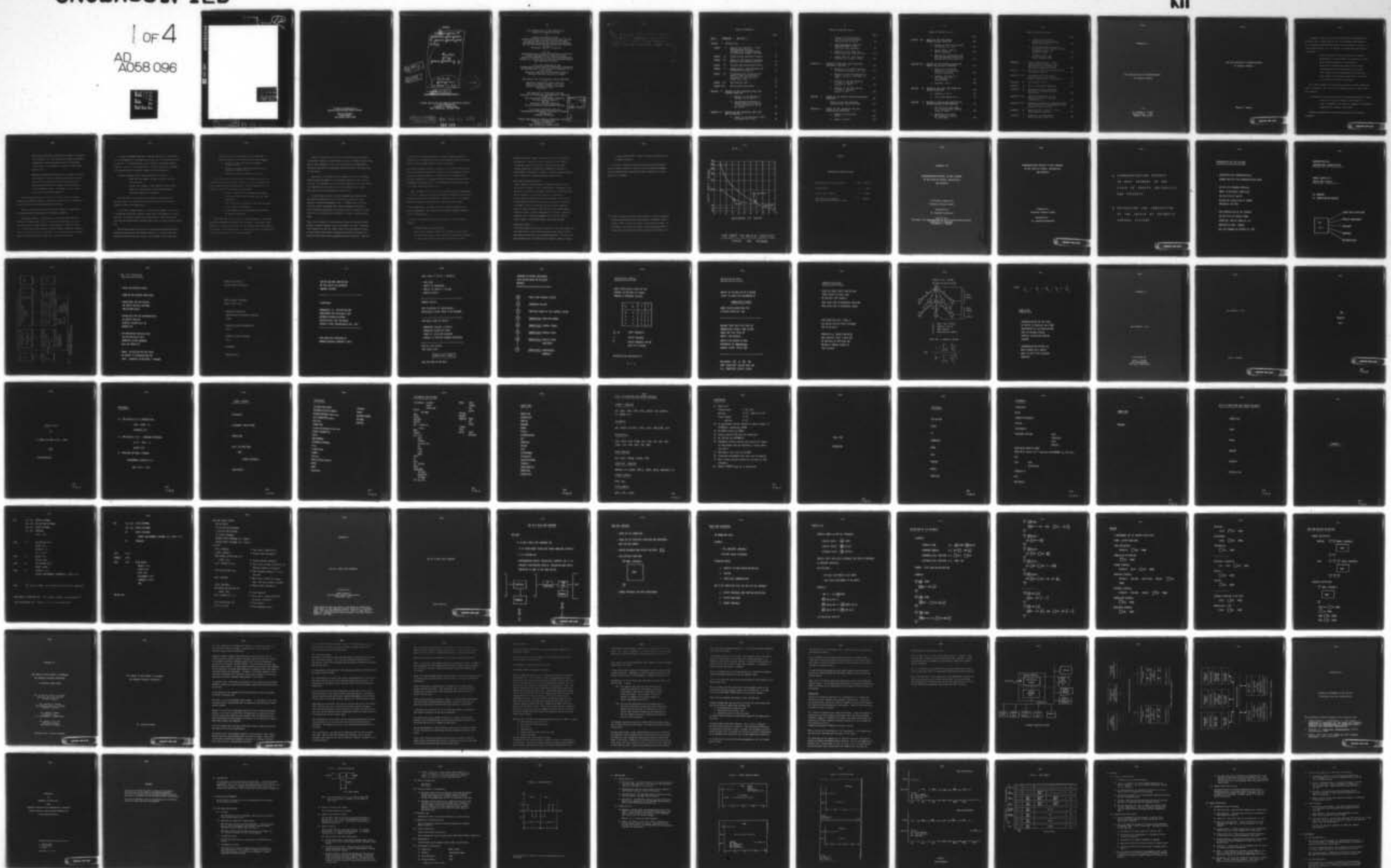
AD-A058 096 PURDUE UNIV LAFAYETTE IND PURDUE LAB FOR APPLIED IND--ETC F/G 9/2
ANNUAL MEETING OF INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL C--ETC(U)
1977 N00014-78-C-0127

UNCLASSIFIED NL

1 OF 4
AD
A058 096



78 08 01 073

This Report is Published as Part of
Engineering Experiment Station Bulletin 143 Series

Schools of Engineering
Purdue University
West Lafayette, Indiana 47907

MINUTES

FIFTH ANNUAL MEETING of

INTERNATIONAL PURDUE WORKSHOP

ON INDUSTRIAL COMPUTER SYSTEMS

(5th)

Minutes.

PART II

TECHNICAL APPENDICES II - X.

342 p.

1977

Held at

Purdue University

on October 3-6, 1977.

Contract N00014-78-C-0127

Purdue Laboratory for Applied Industrial Control
Purdue University
Schools of Engineering
West Lafayette, Indiana 47907

The International Purdue Workshop on
Industrial Computer Systems

is

Jointly sponsored by the
Purdue Laboratory for Applied Industrial Control;
by the Automatic Control Systems Division,
the Chemical and Petroleum Industries Division,
and the Data Handling and Computations Division
of the
Instrument Society of America;

by the
Japan Electronic Industry Development Association
(JEIDA) through the IPW Japan Committee;
and by the Commission of the European Communities
through its General Directorate for Internal Markets
and Industrial Affairs.

It is also sponsored by the
International Federation for Information Processing
as Working Group 5.4 of Technical Committee TC-5,
and by the
Associate Committee for Automatic Control,
National Research Council of Canada.

The work of the International Purdue Workshop
is
partially funded by a grant from the
Naval Air Systems Command, U.S. Navy,
through the Office of Naval Research,
Washington, D.C.

The Workshop is affiliated with the
Institute of Electrical and Electronic Engineers
through the
Data Acquisition and Control Committee
of the Computer Society
and the
Industrial Control Committee
of the Industrial Applications Society.

It is also affiliated with the
International Federation of Automatic Control
through its
Computers Committee.

Purdue Laboratory for Applied Industrial Control
Schools of Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

This Document contains Minutes (Part II) and Technical Appendices 2-10 of the Fifth Annual Meeting of the International Purdue Workshop of Industrial Computer Systems.

ABSTRACT

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

APPENDIX II

THE APPLICATIONS OF MICROPROCESSORS TO PROCESS CONTROL

by

Mr. Thomas G. Gaspar
Merck and Co., Inc.
Rahway, New Jersey

THE APPLICATIONS OF MICROPROCESSORS TO PROCESS CONTROL

Thomas G. Gaspar

I am going to talk to you about the application of microprocessors to process control. My perception is that of the end-user whose objective is to increase the operating efficiency of flow processes in a refinery or a batch chemical plant, etc. Our methods to increase operating efficiency are limited:

- We may use whips, but those are going out of fashion.
- Mechanization - in other words, the replacement of muscle power with machine power, is an alternative.
- We may want to improve the economy of scale through increasing the size of production units or increase their integration by letting material flow from stage to stage without being touched by human hands, and finally
- We can increase the speed of operations.

All of these methods, with the exception of whips of course, imply some level of automation. That is, the use of machines to operate and/or control other machines.

I'm sure many of you are familiar with the following definition of design:

> Given a set of rules of behavior ("The Process") and a set of components, find the "best compromise" in organizing components into a system. ("The Plant")

Computers as components of flow process plants fall into two basic categories:

PRECEDING PAGE BLANK

- Application independent (commercially available, no customization required) -- in other words, using computers as hidden components in purchased equipment which are transparent to us as users. (For example, in cathode ray tubes, wrist watches, etc.)

- Application dependent (commercially not available, customization required) -- that is, computers used for new functions such as direct digital control of batch processes <u>or</u> as replacement alternatives for familiar functions which we can now implement cheaper. (For example, one may replace relay circuits with small computers).

From this particular point of view, I will focus on the use of small computers as application dependent components in the design of a flow process plant with the eye on increasing operating efficiency.

Before I go on with the discussion of small computers, I would like to distinguish between microprocessors and miniature computers.

A miniature computer is something like a very small IBM S370/168 or a CDC Cyber 175 which would fit in your briefcase but otherwise can do all the functions of its bigger brother. By microprocessor I mean the "chip", i.e. a collection of slow, primitive, computer elements (registers, arithmetic unit, and control logic) on one monolithic silicone substrate with or without memory, optionally purchasable mounted on a printed circuit board.

Of course if somebody hands you a small box and you can't tell whether it is a minicomputer or a microprocessor you may try to run corporate payroll in that box. If the payroll runs, we are talking of a miniature business computer. If not, it is a microprocessor. In other words, try a benchmark to distinguish between a miniature computer and a microprocessor.

End-users typically justify microprocessors on the basis of

- Small size (for example, one must include a controller in an actuator).
- Low cost (for example, a more expensive classical minicomputer or large amounts of slow, special purpose, logic would otherwise be required).

Note that while size-related special packaging considerations are generally valid, cost considerations may very well favor classical minicomputers, instead of microprocessors.

Let me show an example (Figure 1). Here we have compared the unit cost of automating a laboratory instrument using either a minicomputer or a microprocessor. As the curve shows we have found that you must build at least seven identical systems before microprocessors start to have an edge on capital cost.

When we also compute the total cost of ownership including spare parts, maintenance documentation, maintenance training, etc., we have found that, taking the technical risks into account, the following rule of thumb holds:

From the total cost of ownership point of view micro-processors should not be used for process control unless:

1. You need at least 100 identical (hardware/software) units.
2. Sources of re-supply (hardware/software) can be provided for about 18 years.
3. Long-term maintenance capability is assured.

This clearly very cautious posture is supported not only by cost calculations, but also by bitter experience. In fact, it seems to me that microprocessors are manufactured to serve the fabricator's convenience not that of the end-user. Specifically:

- Design is simplified to the extreme; for example, instructions which require complex logic are seldom implemented.
- Quality assurance is minimized to hold the costs down in the competitive marketplace.
- The programming tools provided with microprocessors are generally primitive.

Please note mine is not a lone voice in the wilderness. In Toronto, during the International Federation for Information Processing's (IFIP) Congress 77, I. Barron pointed out that the customary 8-bit word length is "very silly". At the same meeting, E. Dijkstra noted that "microprocessors are not great" and in fact are "very unreliable".

However, with all their architectural and quality control problems, micropressors represent a step forward in the art of hardware manufacturing. What is truly remarkable is that this step forward is accompanied by a twenty-year regression in programming capability and very few people seem to take notice.

What most of us overlook is that in addition to the art of hardware manufacturing, programming has also advanced dramatically over the last 20 years. The programming of a process which would have cost of the order of $1,000,000 in the late 1950's, cost about $200,000 in the middle 60's, and as little as $10,000 now in the late 70's.

In other words, the cost of control programming declined by about two orders-of-magnitude in the last 20 years, primarily due to the evolution of advanced programming tools. I remember when, in the day of million dollar programming jobs, end-user management insisted on machine language programming, since computers were too valuable to waste their time on foolishness such as FORTRAN. Of course, time proved them wrong.

Today, we hear promoters of microprocessors argue that end-users should program in machine language since microprocessors are cheap. Presumably, cheap computers are what the "market" wants and micropressors are cheap not only because their design is simplified and quality assurance minimized, but also because their programming tools are primitive. Hence the regression to the computing methods of a quarter century ago and the introduction of our break-even rule that micropressors should not be used unless at least a hundred identical units are required.

The apparent marketing strategy of "low price at any price" may be caused by the microprocessor manufacturers' generally valid perception of their customers as computer hobbyists. A hobbyist wants to eliminate the expensive middleman, the programmer; considers keyboarding the key skill of programming; and equates the art of programming with telling the machine in 40 simple instructions what to do.

What the hobbyist fails to realize is that microprocessor programming is the art of completely specifying the behavior (including error states) of a complex automaton with a vocabulary of 40 words or less -- an expensive hobby which frequently frustrates even experienced professionals.

Do not misunderstand: My thesis is not that end-users should not do computer programming. They should, given the right equipment instead of today's microprocessors.

In closing, let me share with you my conjectures as to what is really required:

- Firmware/hardware operating systems

  Most systems computers require basic software, usually called the "operating system", to make them work. While this arrangement is satisfactory for large EDP systems, integration of user programs with small computer operating systems is quite difficult and expensive. Therefore, software operating systems should go. A transitory step in this direction could replace them with firmware equivalents, followed by a fundamental rethinking of small computer architecture, leading to an end-user oriented generation of hardware not requiring operating systems at all.

- Multi-engine Arithmetic Machines

  Current computers are Single-engine Arithmetic Machines: they can do only one thing at a time. Historically, considerable effort was spent on operating systems which "timeshare" or "multi-program" process computers, i.e. make them appear as if working on several subtasks simultaneously and independently. More recently, the "distributed system" idea came into vogue where the subtasks of running a process are performed by interconnected computers. Either of these alternatives imply the need for the Multi-engine Arithmetic Machine, possibly implemented as a network of miniature computers, each of which was designed to be a low-cost network element capable of performing simple, specialized subtasks.

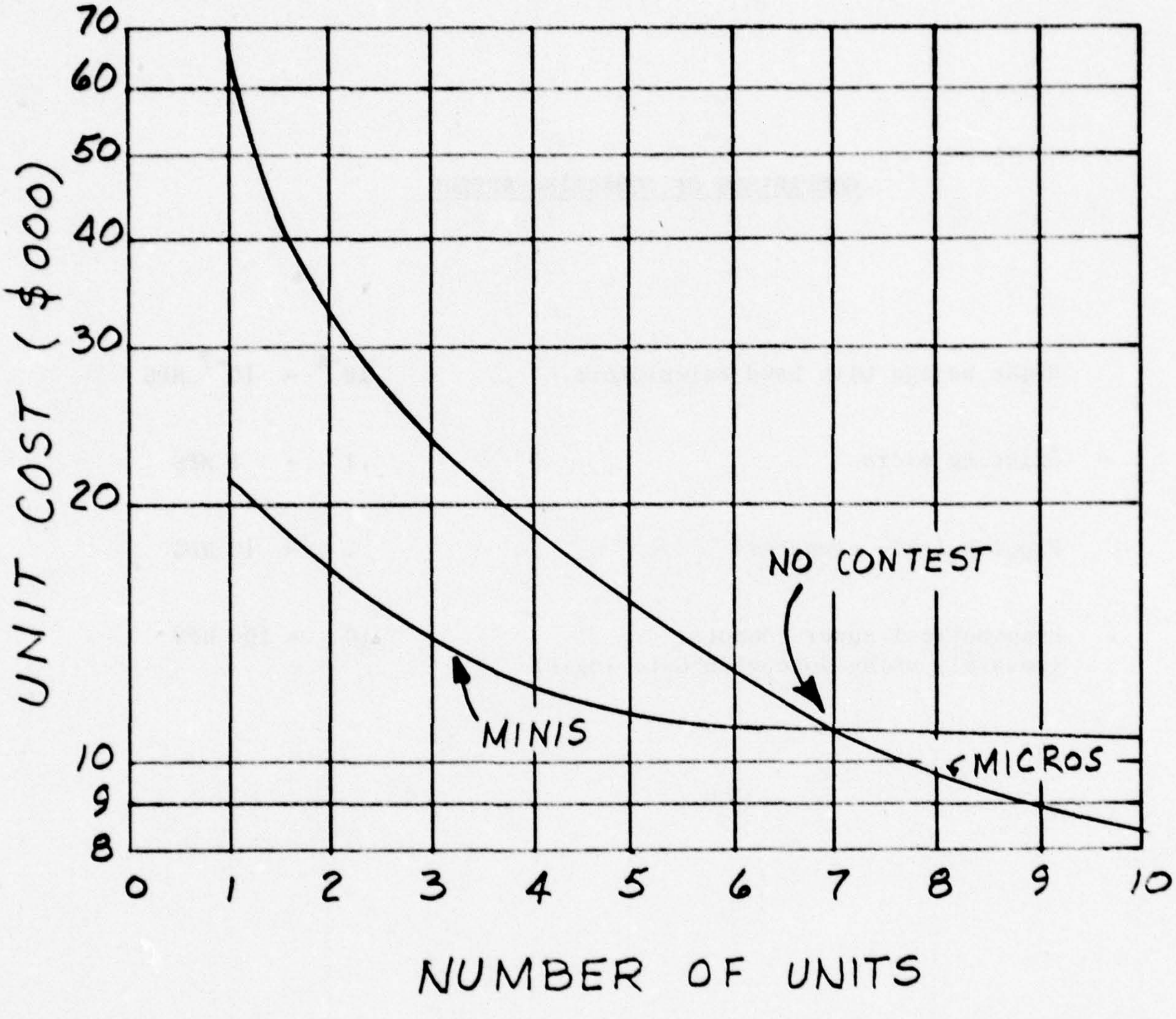
- Superspeed computers

  I know some end-users who would get ecstatic if they could simply compute trajectories of large chemical processes and optimize them in real-time using advanced linear programming methods. To do this, one would probably need a hypothetical super computer capable of speeds of around 10-100 HPS[(1)] - people or micros just would not do as shown in Figure 2.

Of course, to work with operating system-free superspeed Multi-engine Arithmetic Machines we end-users will need superb programming tools, including perhaps new problem-oriented languages and a fresh approach to de-bugging.

(1) A Horner is about one million calculations, i.e. a mix of additions subtractions, multiplications, and divisions. Therefore, Horner/second, or HPS, is about one millions calculations per second. On this subject, start with the chapter on "Superspeed Computers" of "Future Facts" by Stephen Rosen (Simon & Shuster, New York), p. 47.

FIG. 1



THE COST TO BUILD CONTEST

MINIS VS. MICROS

FIGURE 2

COMPARISON OF COMPUTING SPEEDS

. Human beings with hand calculators	10^{-6} - 10^{-7} HPS
. Existing micros	.1 - 1 HPS
. Bipolar logic computers	1 - 10 HPS
. Hypothetical super computer (possibly using Josephson Gate logic)	10 - 100 HPS

APPENDIX III

STANDARDIZATION EFFORTS IN WEST GERMANY IN THE FIELD OF SAFETY, RELIABILITY, AND SECURITY

A Tutorial prepared by

Professor Richard Lauber

presented by

Dr. Rudolph Konakovsky

both of the
Institute fur Regelungstechnik und Prozessautomatisierung
University of Stuttgart
Stuttgart 1, Germany

STANDARDIZATION EFFORTS IN WEST GERMANY IN THE FIELD OF SAFETY, RELIABILITY, AND SECURITY

Prepared by
Professor Richard Lauber

presented by
Dr. Rudolph Konakovsky

1. STANDARDIZATION EFFORTS IN WEST GERMANY IN THE FIELD OF SAFETY, RELIABILITY AND SECURITY

2. DEFINITION AND COMPUTATION OF THE SAFETY OF AUTOMATIC CONTROL SYYTEMS

PRECEDING PAGE BLANK

STANDARDIZATION INSTITUTIONS

- AUTHORIZED FOR STANDARDISATION:
  GERMAN INSTITUT FOR STANDARDIZATION (DIN)

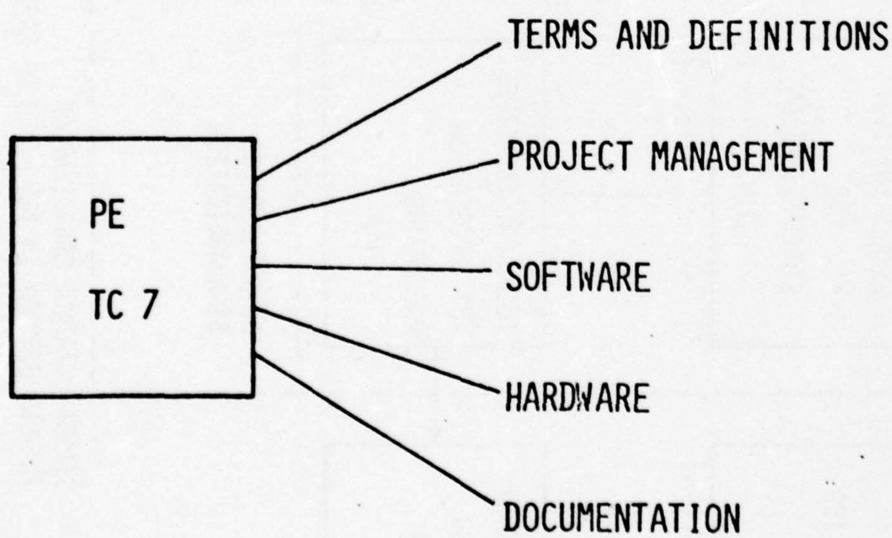
- ACTIVE FOR STANDARD-PROPOSALS
  ABOUT 10 DIFFERENT COMMITTEES
  IN THE FIELD OF SAFETY
  WITHIN THE ASSOCIATION OF GERMAN
  ENGINEERS (VDI/VDE)

- FOR COORDINATION OF ALL EFFORTS
  IN THE FIELD OF SAFETY TERMS
  COMMITTEE: SAFETY TERMS (A 1.7)
  PROPOSED BY PROF. LAUBER
  WILL BE FOUNDED ON OCTOBER 21, 1977

COOPERATION WITH
INTERNATIONAL ORGANIZATIONS

PURDUE EUROPE TC 7
SAFETY AND SECURITY

32 MEMBERS
17 CORRESPONDING MEMBERS



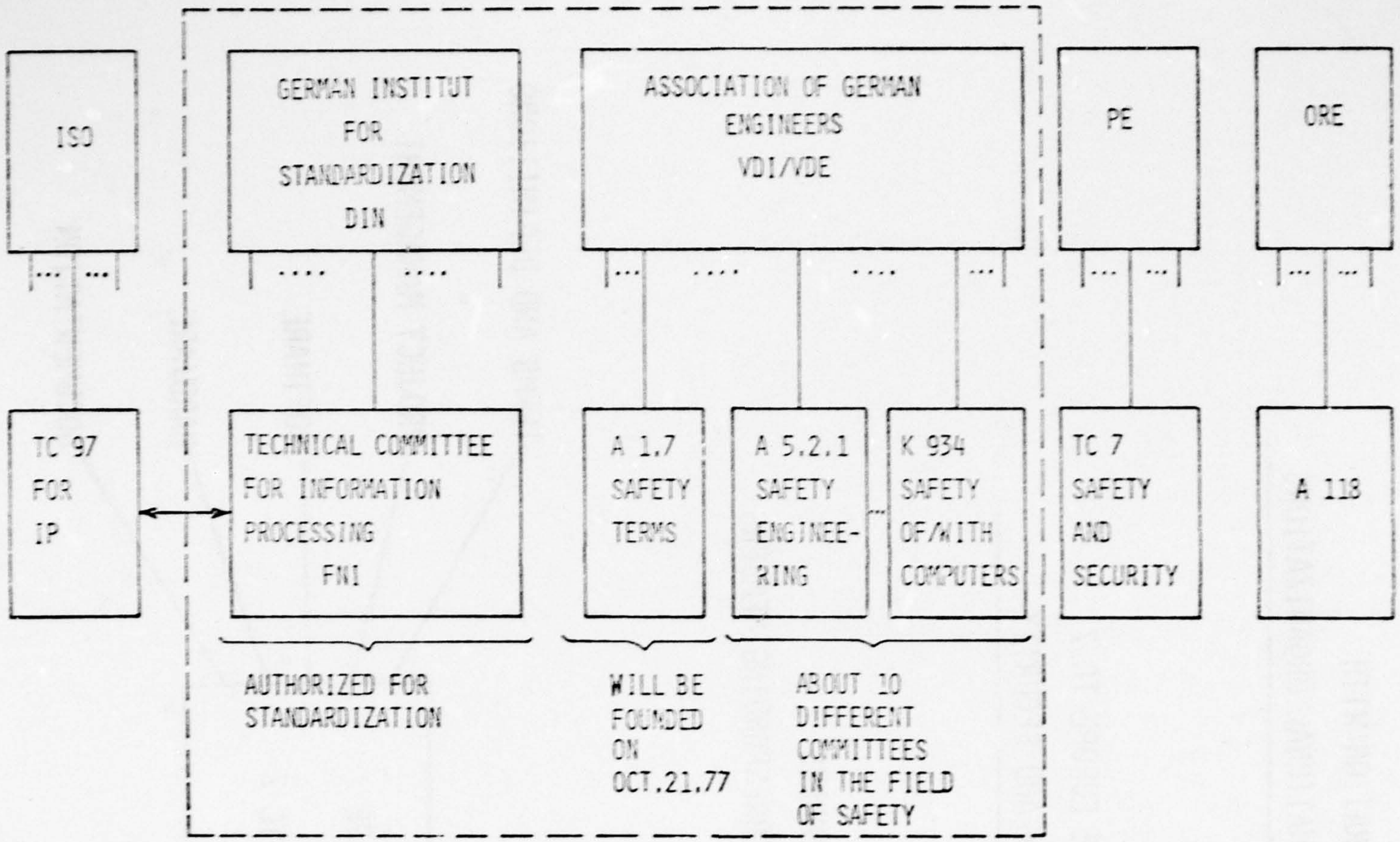


FIG. 1 STANDARDIZATION ORGANIZATION
IN WEST GERMANY IN THE FIELD OF SAFETY

PE / TC 7 ACTIVITIES

- OVER 100 WORKING PAPERS

- SOME WP ARE ALREADY PUBLISHED

- GUIDELINES FOR THE DESIGN OF SAFETY RELATED SOFTWARE ARE ACCOMPLISHED

- GUIDELINES FOR THE DOCUMENTATION OF SAFETY RELATED COMPLEX SYSTEMS WILL BE WORKED OUT

- WP CONCERNING POSSIBILITIES IN THE DESIGN OF SAFE COMPUTER SYSTEM HARDWARE WILL BE COMPLETED

- PANEL DISCUSSION IN THE FIELD OF SAFETY IS SCHEDULED ON THE IFAC - CONGRESS IN HELSINKI / FINLAND

FURTHER ACTIVITIES
IN THE FIELD OF SAFETY

INSTITUTIONS SPONSORED
THESE ACTIVITIES:

- RESEARCH PROMOTION
  INSTITUTION FOR PROCESS CONTROL
  (PDV)

- GERMAN RESEARCH FOUNDATION
  (DFG)

- GERMAN FEDERAL RAILWAYS
  (DB)

- COMPANIES

- UNIVERSITIES

DEFINITION AND COMPUTATION
OF THE SAFETY OF AUTOMATIC
CONTROL SYSTEMS

LITERATURE:

KONAKOVSKY, R.: DEFINITION UND BERECHNUNG DER SICHERHEIT VON AUTOMATISIERUNGSSYSTEMEN. DISSERTATION, UNI STUTTGART, VIEWEG & SOHN, BRAUNSCHWEIG OKT. 1977

THIS WORK WAS SPONSORED BY
GERMAN RESEARCH COMMUNITY (DFG)

SOME KINDS OF SAFETY / SAFENESS:

- FAIL-SAFE
- SAFETY OF OPERATION
- SAFETY IN TRAFFIC / FLYING
- REACTOR SAFETY

COMMON FEATURE:

THE OCCURRENCE OF PARTICULARLY UNDESIRABLE EVENTS HAVE TO BE EXCLUDED

FOR ABOVE KINDS OF SAFETY:

- DANGEROUS FAILURE (-EFFECT)
- DANGEROUS OPERATION STATE
- TRAFFIC / AVIATION ACCIDENT
- DAMAGES TO REACTOR THROUGH EXPLOSION

FOR ALL SUCH EVENTS
THE COMMON TERM

INADMISSIBLE EVENTS

WILL BE USED IN THE NEXT

SEQUENCE OF EVENTS OBSERVABLE
IN A SYSTEM WHERE AN ACCIDENT
OCCURES

(0) FAULT-FREE CONTROL SYSTEM

(1) COMPONENT FAILURE

(2) FUNCTION CHANGE OF THE CONTROL SYSTEM

(3) <u>INADMISSIBLE</u> FUNCTION CHANGE

(4) <u>INADMISSIBLE</u> CONTROL SIGNAL

(5) <u>INADMISSIBLE</u> PROCESS STATE

(6) <u>INADMISSIBLE</u> PROCESS EVENT
(ACCIDENT)

(7) <u>INADMISSIBLE</u> CONSEQUENCES
(DAMAGES)

ILLUSTRATIVE EXAMPLE

LOGIC FUNCTION OF A UNIT OF THE CONTROL SYSTEM AND ITS CHANGE THROUGH A COMPONENT FAILURE :

U_1	U_2	Y	$\hat{Y}$
0	0	0	0
0	L	0	L
L	0	0	0
L	L	L	L

U_1, U_2 INPUT VARIABLES

Y OUTPUT VARIABLE

$\hat{Y}$ OUTPUT VARIABLE IN THE CASE OF A FAILURE

CRITERIUM FOR ADMISSIBILITY :

$$\hat{Y} \leq Y$$

DEFINITION OF SAFETY

SAFETY IS THE ABILITY OF A SYSTEM (ITEM) TO AVOID THE OCCURRENCE OF

INADMISSIBLE EVENTS

UNDER STATED CONDITIONS FOR A STATED PERIOD OF TIME

BECAUSE THERE ARE FIVE TYPES OF INADMISSIBLE EVENTS (③ TO ⑦) THERE ARE FIVE TYPES OF SAFETY, FOR EXAMPLE:

SAFETY WITH REGARD TO NON-OCCURRENCE OF INADMISSIBLE CONTROL SIGNAL (EVENT ④)

FOR EVENTS ③ TO ⑤ THE TERM "DANGEROUS" CAN BE USED TOO, E.G.: DANGEROUS CONTROL SIGNAL

COMPUTATION METHOD

FROM THE ABOVE SAFETY DEFINITION
SOME FIGURES OF MERIT CAN
BE DERIVED, FOR EXAMPLE:

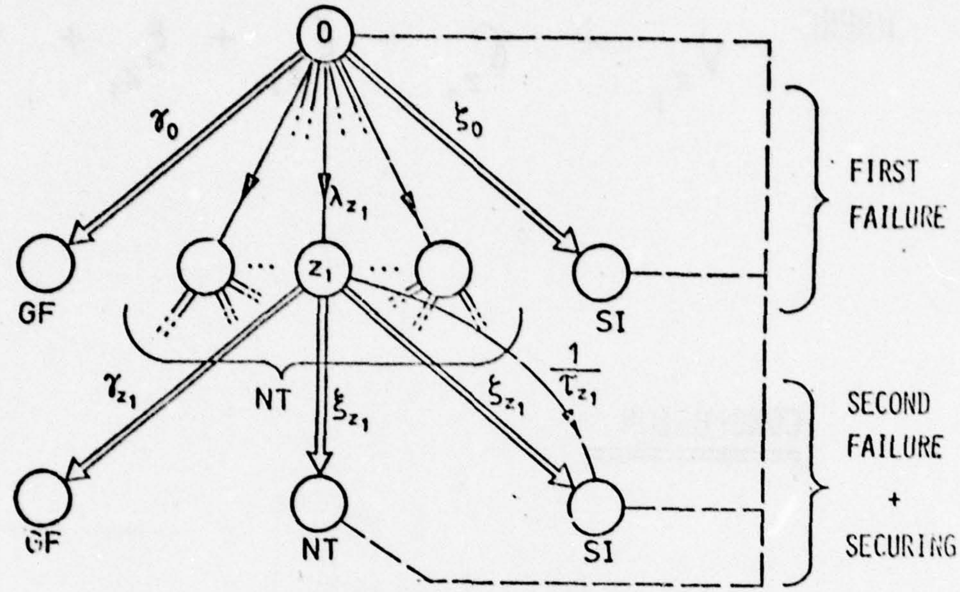
MTDF MEAN TIME TO DANGEROUS FUNCTION
MTDS MEAN TIME TO DANGEROUS SIGNAL

FOR COMPUTING MTDF / MTDS A
SO CALLED FAILURE-EFFECT-DIAGRAM
HAS TO BE BUILT

PROVIDED ALL TRANSITION RATES
ARE CONSTANT, MTDF / MTDS CAN
BE OBTAINED BY APPLYING THE
METHOD OF MARKOV CHAINS TO
THIS DIAGRAM

FAILURE - EFFECT - DIAGRAM

FOR SINGLE AND DOUBLE FAILURES



0 FAULT - FREE FUNCTION

GF DANGEROUS FUNCTION

SI SAFE FUNCTION

NT FUNCTION NOT YET CHANGED

MEAN TIME TO DANGEROUS FUNCTION

$$\text{MTDF} = \frac{1 + \sum\limits_{z_1 \in NT} \frac{\lambda_{z_1}}{\nu_{z_1}}}{\gamma_0 + \sum\limits_{z_1 \in NT} \frac{\lambda_{z_1}\gamma_{z_1}}{\nu_{z_1}}}$$

WHERE $$\nu_{z_1} = \gamma_{z_1} + \xi_{z_1} + \zeta_{z_1} + \frac{1}{\tau_{z_1}}$$

CONCLUSION

- STANDARDIZATION IN THE FIELD OF SAFETY IS POSSIBLE IN A FORM INDEPENDENT OF THE APPLICATIONS SUCH AS RAILWAY SYSTEM, CHEMICAL OR NUCLEAR REACTOR SYSTEMS

- STANDARDIZATION EFFORTS IN WEST GERMANY WILL SURELY HELP TO SOLVE THE DISCUSSED PROBLEMS

APPENDIX IV

IBM SERIES/1 PL/1

A Tutorial by

Alex J. Arthur
IBM Corporation
San Jose, California

IBM SERIES/1 PL/1

Alex J. Arthur

IBM

SERIES/1

PL/I

PRECEDING PAGE BLANK

AJA
77.06.16

SERIES/1 PL/I

IS

- A SUBSET OF ANSI X3.53 - 1976

PLUS

- IBM EXTENSIONS

AJA
77.06.16

REFERENCES

1) IBM SERIES/1 PL/I INTRODUCTION,

GC34 - 0084 - 0,

FEBRUARY 1977

2) IBM SERIES/1 PL/I: LANGUAGE REFERENCE

GC 34 - 0085 - 0

AUGUST 1977

3) AMERICAN NATIONAL STANDARD

PROGRAMMING LANGUAGE PL/I,

ANSI X3.53 - 1976

AJA
77.09.30

SUBSET CONTENT

- ATTRIBUTES
- STATEMENTS AND OPTIONS
- CONDITIONS
- BUILT-IN FUNCTIONS
  AND
  PSEUDO-VARIABLES
- CONSTRAINTS

AJA

77.06.16

ATTRIBUTES

ALIGNED/UNALIGNED
AUTOMATIC/STATIC/BASED
BINARY/DECIMAL/PRECISION
BIT/CHARACTER/LENGTH
BUILTIN
CONDITION
DIMENSION/MEMBER/STRUCTURE
DIRECT/SEQUENTIAL
ENTRY
ENVIRONMENT
EXTERNAL/INTERNAL
FILE
FIXED/FLOAT
FORMAT
INITIAL
INPUT/OUTPUT/UPDATE
KEYED
LABEL
PARAMETER
POINTER
PRINT
RECORD/STREAM
RETURNS
VARYING

AJA

77.06.16

STATEMENTS AND OPTIONS

ASSIGNMENT, ELEMENT
ARRAY
STRUCTURE
BEGIN
OPTIONS
CALL
CLOSE
DECLARE
DELETE
DO - ITERATIVE
WHILE
END
FORMAT
GET LIST
EDIT
STRING
EXPRESSIONS
DO
SKIP
LINE
PAGE
GOTO
IF
NULL
ON SYSTEM
OPEN
PROCEDURE
OPTIONS
RECURSIVE
RETURNS
PUT SEE GET

READ INTO
IGNORE
KEY
KEYTO
RETURN
REVERT
REWRITE FROM
KEY
EVENT
SIGNAL
STOP
WRITE FROM
KEYFROM

AJA
77.06.16

CONDITIONS

CONDITION
CONVERSION
ENDFILE
ENDPAGE
ERROR
FINISH
FIXEDOVERFLOW
KEY
OVERFLOW
RECORD
SIZE
STRINGRANGE
STRINGSIZE
SUBSCRIPTRANGE
TRANSMIT
UNDEFINEDFILE
UNDERFLOW
ZERODIVIDE

AJA
77.06.16

BUILT IN FUNCTIONS AND PSEUDO VARIABLES

STRING - HANDLING

BIT, BOOL, CHAR, COPY, HIGH, LENGTH, LOW, SUBSTR & PV, UNSPEC & PV

ARITHMETIC

ABS, BINARY, DECIMAL, FIXED, FLOAT, PRECISION, SIGN

MATHEMATICAL

ACOS, ASIN, ATAN, ATAND, COS, COSD, EXP, LOG, LOG2, LOG10, SIN, SIND, SQRT, TAN, TAND

ARRAY HANDLING

DIM, EVERY, HBOUND, LBOUND, SOME

CONDITION - HANDLING

ONCHAR & PV, ONCODE, ONFILE, ONKEY, ONLOC, ONSOURCE & PV

STORAGE CONTROL

ADDR, NULL

MISCELLANEOUS

DATE, TIME, LINENO

AJA

77.06.16

CONSTRAINTS

1) PRECISION

FIXED BINARY	$p \le 31$, $q=0$
DECIMAL	$p \le 15$, $-128 \le q \le +127$
FLOAT BINARY	$p \le 53$
DECIMAL	$p \le 16$

2) NO ADJUSTABLE STRING LENGTHS OR ARRAY EXTENTS IN AUTOMATIC, DIMENSION, BASED
3) NO REFER OPTION IN BASED
4) STRING LENGTHS 255 BITS OR CHARACTERS
5) NO INITIAL FOR AUTOMATIC
6) PARAMETER STRING LENGTHS AND EXTENTS OF ARRAYS OF STRUCTURES MUST BE CONSTANTS, SIMPLE ARRAY CAN HAVE *
7) ARITHMETIC DATA MUST BE ALIGNED
8) STRUCTURE ASSIGNMENT ONLY FOR LIKE ATTRIBUTES
9) ONLY SIMPLE SCALAR EXPRESSIONS ALLOWED AS CALL ARGUMENTS
10) REMOTE FORMATS MUST BE IN SAME BLOCK

AJA
77.06.16

REAL TIME

EXTENSIONS

AJA
77.06.16

ATTRIBUTES

ACTIVATION

AFTER

AT

CONNECTED

EVENT

LOCK

PROGRAM

SOURCE

TRANSIENT

AJA
77.06.16

STATEMENTS

CLEAR/POST

DELAY

CONNECT/DISCONNECT

DISPLAY

LOCK/UNLOCK

PROCEDURE OPTIONS MAIN
STACKSIZE
TASK
REPEATS

READ/WRITE/REWRITE EVENT

READ/WRITE SENSOR I/O (REQUIRES ENVIRONMENT ON FILE DCL)

RUN

STOP TASK
ACTIVATION

TRANSFER TO

WAIT

UNSCHEDULE

AJA

77.06.16

CONDITIONS

PENDING

AJA

77.06.16

BUILT-IN FUNCTIONS AND PSEUDO VARIABLES

COMPLETION

COUNT

DAYNO

ONCOUNT

PRIORITY

STATUS & PSV

AJA
77.06.16

EXAMPLES

```
DCL         (T1, T2)  ENTRY EXTERNAL,
            (A1, A2)  ACTIVATION EXTERNAL,
            (E1, E2)  EVENT EXTERNAL,
            (P1, P2)  PROGRAM,
                      I(2), J(2);
               .
               .
RUN         T1        ACTIVATION (A1),
                      EVENT (E1),
                      PRIORITY (I),
                      AT ('1215');
RUN         T2        EVENT (E2),
                      AFTER (200);
RUN         P1        ACTIVATION (A2),
                      EVERY (2000);
RUN         P2        PRIORITY (J),
                      SOURCE ENVIRONMENT (DEVNBR(4), PIBIT (6));
               .
               .
               .
WAIT        (E1, E2)(1); /*WAIT FOR EITHER TASK T1 OR T2 TO COMPLETE*/
               .
               .
               .
UNSCHEDULE ACTIVATION (A2); /*ELIMINATE SCHEDULE FOR PROGRAM P1*/

STOP ACTIVATION (A2); /*STOP IT IF IT IS IN EXECUTION*/
```

```
DCL       (L1, L2)  LOCK EXTERNAL,
          (E1, E2)  EVENT EXTERNAL,
          E3        EVENT EXTERNAL
                       SOURCE ENVIRONMENT (DEVNBR (4), PIBIT (7)),
          P3        PROGRAM;
.
.
.
LOCK      (L1);
CONNECT   (E3);
LOCK      (L2)      ELSE BEGIN:
                       UNLOCK (L1);
                       CLEAR (E1);
                       POST (E2);
                       DISCONNECT (E3);
                       TRANSFER TO P3;
                       END;
.
.
.
UNLOCK ALL;
.
.
.
```

```
DECLARE PCNTRL ENTRY,
   BATCH ENTRY,
   PC ACTIVATION EXTERNAL,
   B ACTIVATION EXTERNAL,
   E1 EVENT EXTERNAL,
   DAYBEG EVENT EXTERNAL AT ('0800'),
   DAYEND EVENT EXTERNAL AT ('1730');
CTLLOOP:
   WAIT (DAYBEG);                 /* WAIT UNTIL 8:00 A.M.*/
   CLEAR (DAYBEG);                /* RESET EVENT VARIABLE*/
   RUN PCNTRL ACTIVATION (PC)
       EVENT (E1);                /* START PROCESS CONTROL*/
   WAIT (DAYEND,E1)(1);           /* WAIT UNTIL EITHER 5:30 P.M. OR
                                     PROCESS CONTROL FINISHES*/
   STOP ACTIVATION (PC);          /* STOP PROCESS CONTROL IF STILL
                                     ACTIVE*/
   WAIT (DAYEND);                 /* WAIT UNTIL 5:30 P.M. UNLESS
                                     THAT TIME HAS ALREADY PASSED*/
   CLEAR (DAYEND);                /* RESET EVENT VARIABLE*/
   RUN BATCH ACTIVATION (B)
       EVENT (E2);                /* START BATCH*/
   WAIT (DAYBEG,E2) (1);          /* WAIT UNTIL EITHER 8:00 A.M.
                                     OR BATCH FINISHES*/
   STOP ACTIVATION (B);           /* STOP BATCH*/
   GO TO CTLLOOP;                 /* START MORNING TASK*/
       .
       .
       .
```

APPENDIX V

APL AS A REAL-TIME LANGUAGE

A Tutorial by

Janos Gertler
Case-Western Reserve University
Cleveland, Ohio

"The reported work was done at the Chemical Engineering Department of Case Western Reserve University, Cleveland, Ohio, with the participation of Dr. Adin J. Mann and Dr. Robert V. Edwards under research grants from NIH (General Medical) and ERDA."

APL AS A REAL-TIME LANGUAGE

Janos Gertler

APL AS A REAL-TIME LANGUAGE

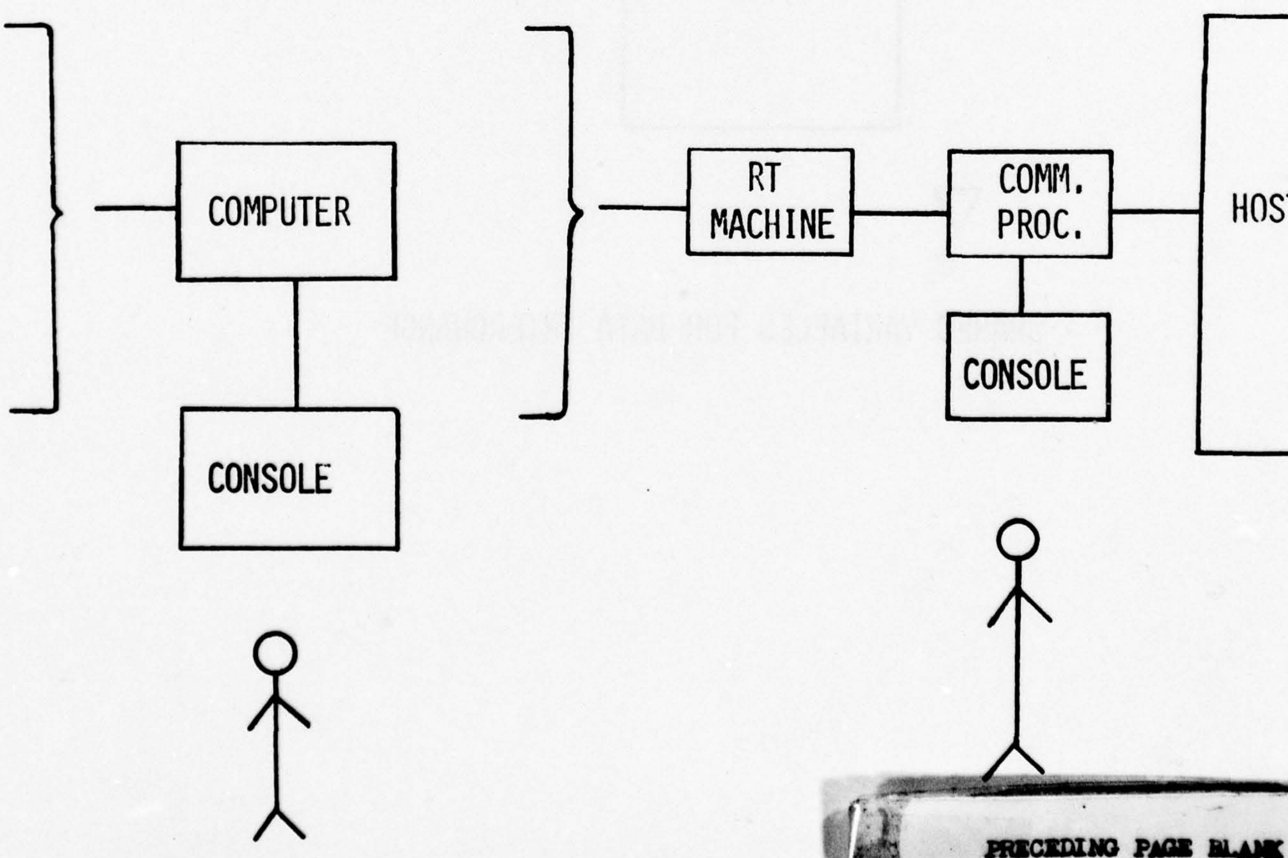
WHY APL?

IT IS NOT A REAL-TIME LANGUAGE BUT

- IT IS INTELLIGENT (EXCELLENT ARRAY HANDLING FEATURES)

- IT IS INTERACTIVE

- EXPERIMENTING PEOPLE (PHYSICISTS, CHEMIST) USE IT TO EVALUATE EXPERIMENTAL RESULTS: EVALUATION AND EXPERIMENTATION IS DONE IN THE SAME SYSTEM.



SOME APL FEATURES

- LARGE SET OF CHARACTERS
- LARGE SET OF PRIMITIVE FUNCTIONS AND OPERATORS, MOST DEFINED ARRAYS
- SYSTEM VARIABLES AND SYSTEM FUNCTIONS (⎕)
- USER DEFINED FUNCTIONS

∇ NAME; VARIABLES

∇

- SHARED VARIABLES FOR DATA INTERCHANGE

REAL-TIME EXTENSIONS

NO BRAND NEW IDEA.

SOURCES:

- LTPL CANDIDATE LANGUAGES
- AN EARLY ALGOL EXTENSION

EXTENSION AREAS:

1. PROCESS I/O AND SYSTEM DEFINITION
2. TASKING
3. INTER-TASK COMMUNICATION

HAS TO BE CONSISTENT WITH THE REST OF THE LANGUAGE:

1. SYSTEM VARIABLES AND FUNCTION DEFINITION
2. SYSTEM FUNCTIONS
3. SHARED VARIABLES

PROCESS I/O

SYMBOLIC NAMES AS SPECIAL VARIABLES

PROCESS INPUT: $\boxed{\rightarrow}$ TEMP6

PROCESS OUTPUT: $\boxed{\leftarrow}$ VALVE3

EXTERNAL EVENT: $\boxed{\downarrow}$ SWITCH1

PROCESS INPUT AND OUTPUT VARIABLES ARE USED IN PROGRAMS AS ORDINARY VARIABLES.

RESTRICTIONS:

- NO VALUE ASSIGNMENT TO AN INPUT
- ONLY VALUE ASSIGNMENT TO AN OUTPUT

EXAMPLES:

$Y \longleftarrow 3 + K \times \boxed{\rightarrow}\ \mathrm{TEMP6}$

$\boxed{\leftarrow}\ \mathrm{VALVE3} \longleftarrow 1$

$\boxed{\leftarrow}\ \mathrm{VALVE3} \longleftarrow \boxed{\rightarrow}\ \mathrm{TEMP6} > \mathrm{TL6}$

$\boxed{\leftarrow}\ \mathrm{VALVE} \longleftarrow \boxed{\rightarrow}\ \mathrm{TEMP} > \mathrm{TL}$

I/O OPERATION IMPLICIT

DEFINITION OF I/O VARIABLES

ELEMENTS:

- SYMBOLIC NAME E.G. ⍈ TEMP6, ⍇ VALVE3
- HARDWARE ADDRESS E.G. AM3 [9] , OM4 [6]
- HARDWARE-LEVEL FUNCTION E.G. ⎕ HI3, ⎕ HO4
- SOFTWARE-LEVEL FUNCTION E.G. CONV, REV

FORMAT: LIKE FUNCTION DEFINITION

EXAMPLES

```
∇ ⍈ TEMP6
  ⍈ ← AM3 [9]
∇

∇ ⍈ TEMP6
  ⍈ ← ⎕ HI3 AM3 [9]
∇

∇ ⍈ TEMP6
  ⍈ ← K + Cx ⎕ HI3 AM3 [9]
∇
```

```
∇ ⍈TEMP6
  ⍈ ← PAR CONV ⎕HI3 AM3 [9]
∇

∇ ⍇VALVE3
  OM4 [6] ← ⍇
∇

∇ ⍇VALVE3
  OM4 [6] ← ⎕HO4 ⍇
∇

∇ ⍇VALVE3
  OM4 [6] ← ⎕HO4 K + Cx⍇
∇

∇ ⍇VALVE3
  OM4 [6] ← ⎕HO4 PAR REV ⍇
∇
```

```
∇ ⍈ TEMP [i6]
  ⍈ ← AM3 [8 + i6]
∇

∇ ⍈ TEMP [i6]
  ⍈ ← PAR [i6] CONV ⎕HI3 AM3 [8 + i6]
∇
```

TASKING

A REASONABLE SET OF TASKING FACILITIES

FORM: SYSTEM FUNCTIONS

TASK ACTIVATION

PRIORITY ☐TAN TNAME

CANCELLING ACTIVATION

☐TCA TNAME

SINGLE SCHEDULE

PRIORITY DELAY ☐TSS TNAME

REPEATED SCHEDULE

PRIORITY INTERVAL REPETITION (DELAY) ☐TSR TNAME

PERIODIC SCHEDULE

PRIORITY INTERVAL (DELAY) ☐TSP TNAME

CANCELLING SCHEDULE

☐TCS TNAME

OBTAINING SCHEDULE

☐TDS TNAME

DELAYING

DELAY ☐ TDT TNAME

SUSPENDING

EVENT ☐ TDE TNAME

TERMINATING

☐ TTE TNAME

DEFINING A SEMAPHORE

TOP START ☐ SEM SNAME

REQUEST

☐ SRQ SNAME

RELEASE

☐ SRL SNAME

LINKING A ROUTINE TO AN EVENT

EVENT ☐ REL RNAME

CANCELLING A LINK

EVENT ☐ RCL RNAME

TASK AND ROUTINE DEFINITION

DIRECT DEFINITION:

ROUTINE ∇∇ RNAME; VARIABLES

BODY

∇∇

TASK ∇∇∇ TNAME; VARIABLES

BODY

∇∇∇

INDIRECT DEFINITION:

∇ NAME; VARIABLES

BODY

∇

CODE ← ⎕CR 'NAME'

⎕ EX 'NAME'

CODE ⎕RD RNAME

CODE ⎕TD TNAME

APPENDIX VI

THE IMPACT OF NEW TRENDS IN HARDWARE ON COMPUTER SYSTEMS INTERFACES

A TUTORIAL ROUND TABLE

Dr. R. Warren Gellie, Chairman
National Research Council
Ottawa, Canada

Mr. Anthony D. Deramo
Westinghouse Electric Company
Orlando, Florida

Mr. Charles Farmer
Honeywell, Inc.
Ft. Washington, Pennsylvania

Dr. Daniel T. W. Sze
IBM Corporation
Boca Raton, Florida

Writeup by Mr. Anthony Deramo

60

THE IMPACT OF NEW TRENDS IN HARDWARE ON COMPUTER SYSTEMS INTERFACES

Mr. Anthony Deramo

Dr. Gellie asked that I highlight the efforts of TC5 and review with you the functional areas where there is potential for support from other technical groups within the Workshop.

Industrial Process Computer Inter-Subsystem Communication has been the primary concern of TC5 for the past two years and the commitee has derived a functional requirements document against which existing and proposed communication subsystems can be evaluated. The discussions have been long and not without emotion. To some, many of the discussions have been redundant. To the impatient, TC5 should have settled on a standard by this time based on available methods and technology. To others, it has been important to lay some basic groundwork which may have a converging influence on the various approaches to process computer communications in the future.

I personally have difficulty in extracting from the functional requirements document, a practical subset for each process communication requirement that we now have.

I have selected a few examples from existing programs within Westinghouse to further illustrate this.

The first is a new microcomputer based product. In the memory of this unit is process data in digital form that is required or could be used by other control systems.

Second, is a new line of functional modules for use in distributed control systems. This type of system offers the opportunity to extend and distribute the digital base making it more flexible than was formerly possible. Also distributing the data presents a different set of coordination requirements than formerly was necessary.

The third example deals with the conventional process computer system and the impact of LSI on this architecture.

The energy crunch, environmental pressures, and the need to make a profit require that more sophisticated levels of control be added. These levels in the form of conventional computers connected in a computer hierarchy will require access to the available data base.

It is at this level that the great majority of TC5 decisions do involve future trends in software methods and there is a need to factor this thinking into our decision making process.

Now! the first example.
We recently concluded the design and development of an electronics package for our insitu 02 sensor. This was done to support polution monitor applications, where the EPA requires that the sensor calibration be checked and recorded daily.

We incorporated a microcomputer in the package to provide a convenient means to record calibration data.

The implementation was simple and straight forward because we knew exactly what had to be done to gain wide spread acceptance for the product. We structured the log format, listed the data in ASCII form and provided an RS232C port.

With the flexibility of the microcomputer, we were able to add a self-calibrating option, and several process controller options. The control options support several applications ranging from damper position control to fuel or air trim control based on excess 02 in a boiler control system.

When doing air trim control for a boiler, the unit has most of the process data in digital form that is required to calculate boiler efficiency. Plant load balance based on boiler efficiency is beyond the capability of this unit but an additional serial communication port was added to the unit to provide access to this process data.

The implementation for this port was not simple and anything but straight forward because we did not know what had to be done to gain product acceptance. No standard exists for serial process data communication of this type.

This is obviously a case where those impatient with TC5 results to date are right. Base on available methods and technology, one of several implementations would do, so long as there was a general acceptance to the selected approach.

New microprocessor-based control system modules called Q-Line have been developed by Westinghouse for use by their system engineers in distributed control systems applied to a broad range of process control applications.

Designed into the Q-line architecture are several communication options.

There is a remotable share memory module which provides the means to establish 100% redundancy for the microprocessor-base control modules and to provide high speed data transfers required for control coordination between unit controllers.

There is a simplex communication link module which provides a read only access for process data when the control system is applied in a safety and protect application.

There a half-duplex supervisory-type link module for the more general process control application. (By supervisory type I mean 32 bit fixed format which includes start, and stop framing bits and a 5 bit Boise Churdi Erroring code.)

There are more options, but point is made, i.e. Communication requirements for distributed microprocessor-based control systems of this type are sensitive to application and the individual suppliers architecture.

It would be very difficult for TC5 to select one supplier's method and gain general acceptance to the selected approach.

The majority of process control computers are single computers which provide a multiprogramming and multitasking environment to control the concurrent execution of a thousand or more tasks.

We can take advantage of the inexpensive large scale integrated circuits in this architecture in several ways without disturbing the basic application.

First, let's replace the central processing unit electronics with LSI being careful to preserve the original instruction set which is required if we are not going to disturb the basic application.

What is gained?

In the case of the W2500 process computer, a dramatic reduction in electronic hardware.

The LSI version occupies less than 20% of the spare required by the original central processing unit.

Less energy is required to operate the unit.

Performance levels are generally improved.

Now let's add programmable LSI to selected units that are connected to the computer bus. This, obviously, leads to distributed processing. Process I/O scans which previously required central computer core memory to store the scan application programs and central processor time to perform the actual scans can now be done in the process I/O unit. The data is passed on to the central data base on a CPU cycle steal basis through the direct memory access bus. Distributed processing has been especially highlighted in the serial communication interface area with announcements by several semiconductor firms of LSI multiprotocol communication chips. (Signetics and SMC-Standard Microsystems Corp). These chips have computer-addressable protocols that enable them, by computer command to handle IBM SDLC; BISYNC; DEC's DDCMP, ADCCP and in the case of the SMC chip, HDLC. These chips do not, however, completely eliminate the software needed for communication and they do not perform any of the hardware functions required to establish the point to point communication channel.

What does this all mean from an application point of view? Well, it means;

1. More available processor time,
2. More available core memory space,
3. Less cost,
4. Improved MTTR,
5. Greater communication flexibility, and
6. Higher performance.

for the conventional single computer system.

In contrast to this type of system, a variety of CPU's are available on a single chip. Add a little memory, process I/O, and a serial communication interface and essentially we have a similar architecture to the conventional single computer system but with a much lower performance level. This type of structure is also effective at the sensor product and process controller level to provide added control flexibility.

Lets look at some of the areas where future trends in software methods could effect TC5 efforts.

I have shown several computers tied together through a process inter-computer subsystem. Depending on your own personal perspective, the process computers may be microcomputers or conventional computers.

Paragraph 4.0 of the TC5 Functional Requirements document deals with ARCHITECTURE. I quote;

"4.1 The communication subsystem shall be capable of supporting centralized intelligence, distributed intelligence, hierarchical intelligence and combinations thereof. In particular, it shall be capable of supporting distributed systems for process measurement and control."

"4.2 The available architectures of the communications subsystem should not preclude direct data interchange between any two subsystems: it should be possible to transmit data directly between any two subsystems on the link without necessarily involving store and forward at a third subsystem."

The hardware required to establish reliable distributed mastership is not a trivial matter but lets forget it in favor of possible software ramafications.

The Data Base manager in this representation as part of its function has responsibility to provide access to the communication system and to establish access through the communication system to other data bases in the system. Normally it is designed to provide access or to limit access only after specific security criteria has been satisfied. In the hierarchy shown, a process data manager would have no need to access the energy management data base. If it did, the energy management unit would not respond.

On the other hand, any of the higher level computer systems may need to access the unit process data base and the security overhead is not required. The establish access function is required in the process unit only for access to the other process unit data base. The process to process access requirement is debatable.

If it is required, then distributed mastership is required to eliminate store and forward operations at the unit process level.

If it is not required, then distributed mastership is not required at the unit process level.

Past experience with the conventional single computer system is that coordination <u>for basic control</u> between two unit processes is minimum. If required the signals needed would be wired to both computers.

There are both software and hardware issues involved here.

Software methods that deal with rules of access for a distributed data base would establish appropriate overheads for;

1. providing access to the data base
2. providing security access, and
3. establishing access to the data base.

or should rules of access be incorporated as part of the communication subsystem?

In dealing with conventional computers, this is still a trade-off decision between hardware and software. With the reduced performance microcomputer architecture, there is probably not horsepower available within the basic unit to perform these tasks so it might be advantageous to incorporate these functions as part of the communication subsystem.

Is it practical to consider distributed mastership at the unit process control level?

Structure of the distributed data base is another area which could impact communication standards.

As an alternate to distributed mastership at the unit process level, a "look at me" indication, would provide notification to the higher levels of a change in configuration and initialization data permitting it to be treated on an exception basis.

In some distributed computer systems, global commands and global peripherals are featured making the individual operating systems act as one unified system.

Again, this is certainly within the performance scope of the conventional computer system. Within the reduced performance microcomputer architecture, there is probably not the requirement or the capability for this level of performance.

CONCLUSIONS

TC5 has as an immediate objective the recommendation of a standard for industrial process computer inter-subsystem communication. To do this, we must make sound, practical, technical judgements concerning the process environment and the communication requirements. There are many more bases to cover if we are to succeed.

As a practical alternate to an immediate standard, I suggest we explore in cooperation with other technical committees within the Purdue Workshop meaningful levels of interface for computers, measurement and control subsystems which contain computers connected together by a serial communication subsystem so that a
standard interconnection method can become a reality.

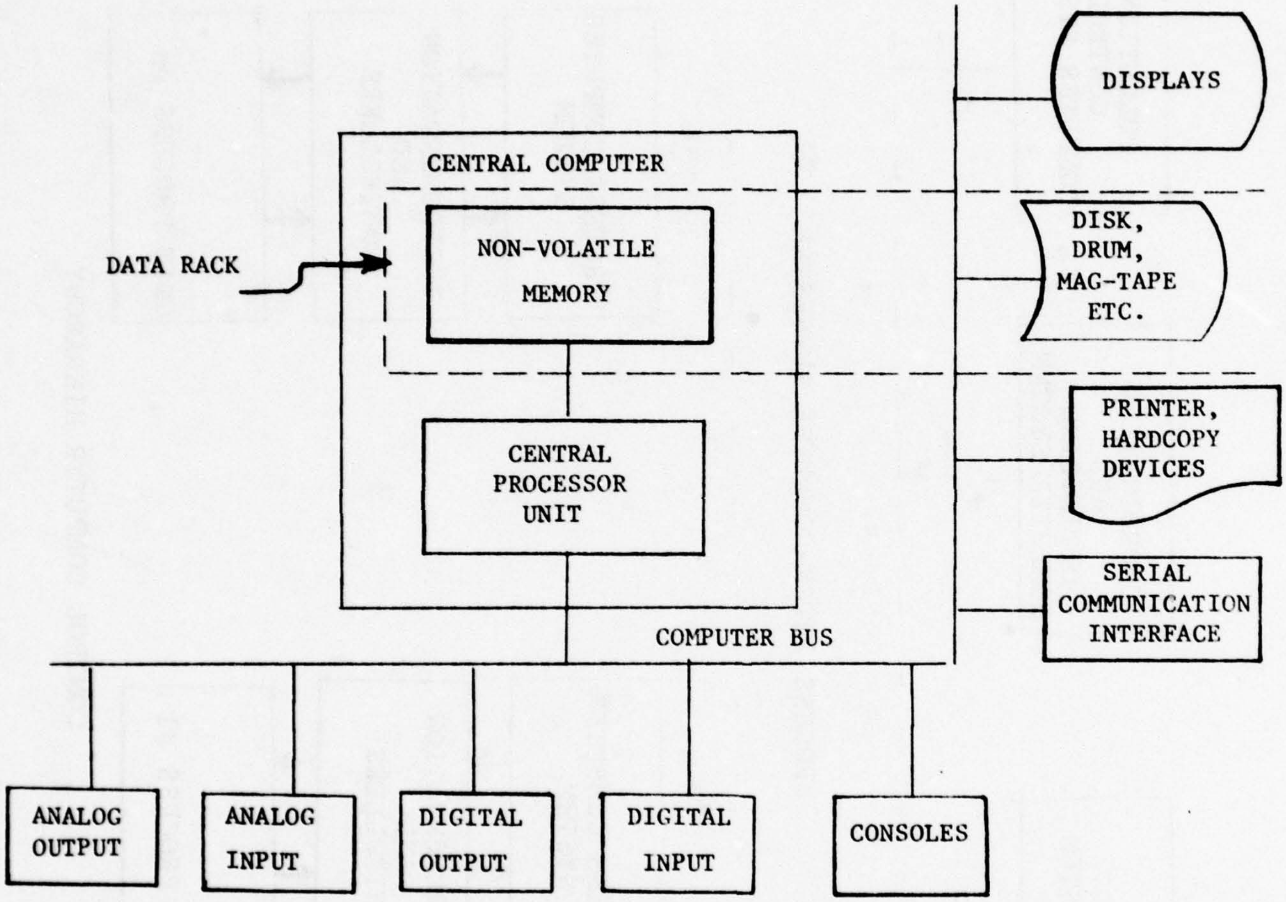
What we are now able to do because of LSI technology is truly amazing and very definitely influences what we expect in the future.

LSI technology and the reduced cost of electronic hardware would tend to support a hardware distributed network approach. Even if the communication hardware cost is appreciable, the cost is fixed, it can be controlled and maintained better than a system where the communication functions are bundled back into the software system.

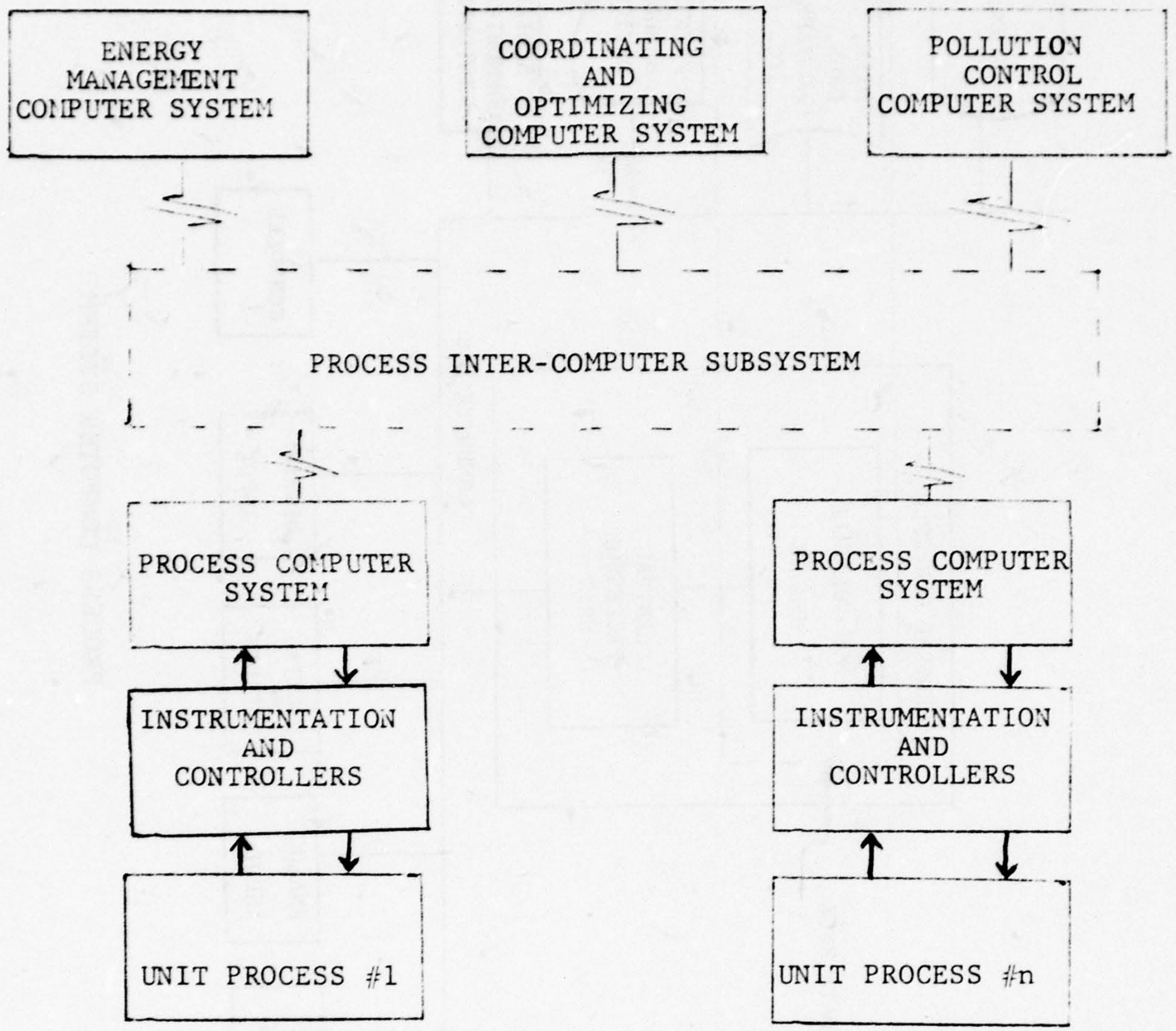
From a control point of view, a distributed network is inherently more reliable than a centralized network and may be more important; it keeps more options open to implement a large variety of individual customer control strategies.

But, is it realistic to consider the many process communication requirements as practical subsets in one general functions/requirements document?

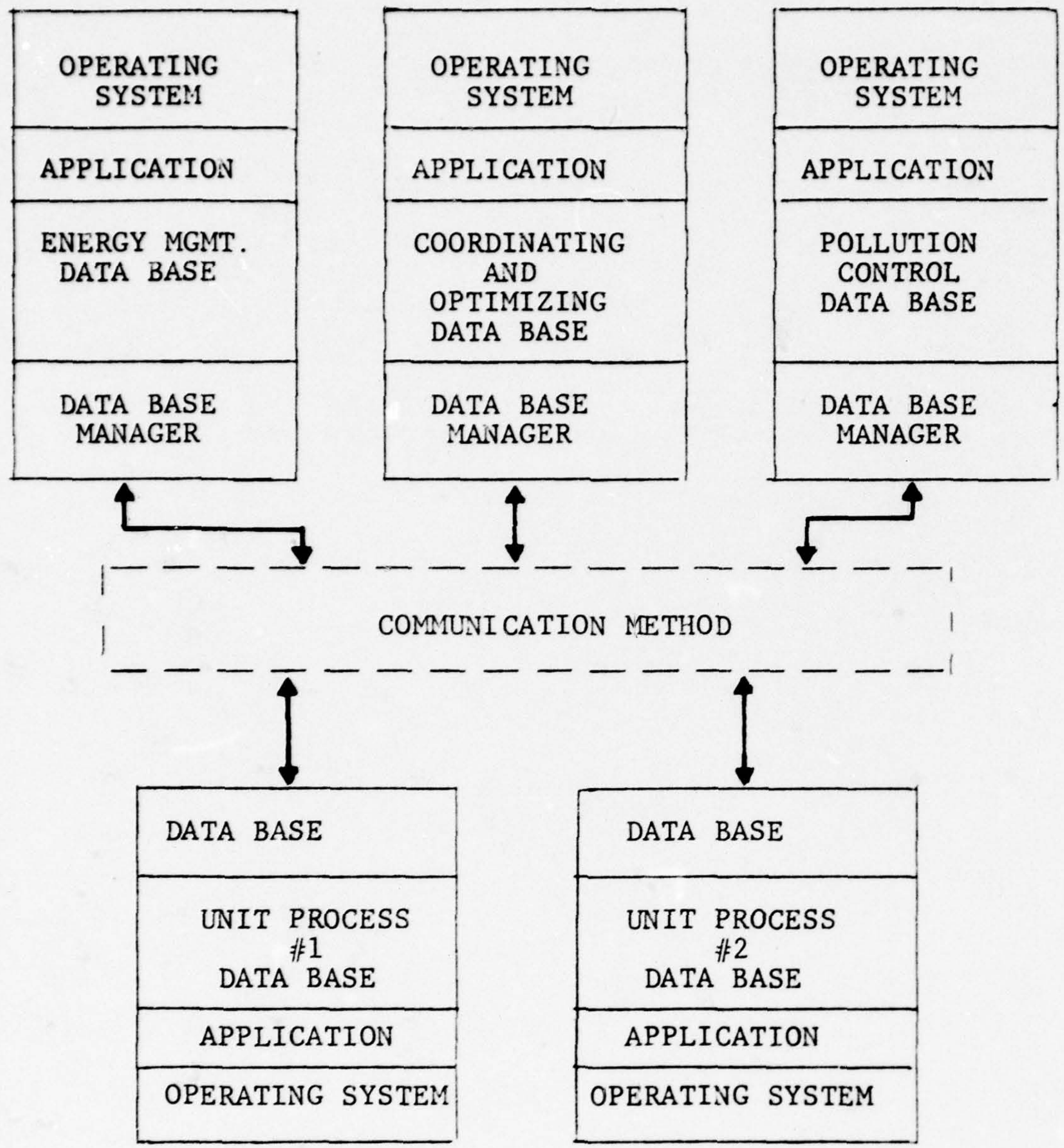
Many of our decisions in TC5, especially at the comprehensive control and optimizing network level do involve future trends in software methods. This thinking must be factored into our decision making process to firm-up the TC5 functional requirements and to make them more meaningful.



PROCESS COMPUTER SYSTEM



CONTROL COMPUTER HIERARCHY



CONTROL COMPUTER HIERARCHY (REDRAWN)

APPENDIX VII

TUTORIALS PRESENTED ON THE SUBJECT INTERFACES AND DATA TRANSMISSION

The following tutorial documents are presented here:

1. Bockett-Pugh, C., Diefenderfer, C., and Farmer, C., Comparison of Honeywell TDC Data Hiway with Proposed Guideline for Implementation of Industrial Process Computer Inter-Subsystem Communication.

2. Podlesny, C., Fiber Optic Communications, Galileo Electro-Optics Corp.

3. Wipfli, John, Yara, Ron, INTEL 8773 SDLC Protocol Controller, Intel Corporation.

PRECEDING PAGE BLANK

COMPARISON

OF

HONEYWELL TDC DATA HIWAY

WITH

PROPOSED GUIDELINE FOR IMPLEMENTATION OF INDUSTRIAL PROCESS COMPUTER INTER-SUBSYSTEM COMMUNICATION

(IEC/SC65A/WG6(JAPAN-2))

Honeywell/Process Control Division

C. Bockett-Pugh
C. Diefenderfer
C. Farmer

September 27, 1977

FOREWORD

This paper describes the characteristics of the Honeywell TDC Data Hiway in the format of the proposed guideline. The guideline is based on both the Purdue Workshop Functional Requirements (ISA SP72) and the IEC WG6 Functional Requirements.

The paper is intended to aid in the comparison of the TDC Data Hiway with these functional requirements.

1.0 INTRODUCTION

The TDC Hiway is based on the proposed guidelines. The TDC Data Hiway is optimized as a secure and efficient data link for process control application. Hence, the protocal is oriented toward 31 bit words including 5 bit BCH check codes. A single 16 Bit word of data can be read or written with respectively 62 or 93 total bits of information transferred.

2.0 APPLICATION ENVIRONMENT

The TDC Hiway is intended for use in the applications as outlined in the proposed guideline.

3.0 FUNCTIONAL SPECIFICATION

3.1 Topology

The TDC Hiway is a dual (redundant) cable system of the multi-drop, multiple branch type.

3.2 Mastership in Industrial Dataway System

The TDC Hiway supports multiple mastership. "Preferred Access" devices (such as computers, and operator consoles) obtain mastership from the Hiway Traffic Director.

The Hiway Traffic Director does not function as a master but as a priority scheduler for the various masters.

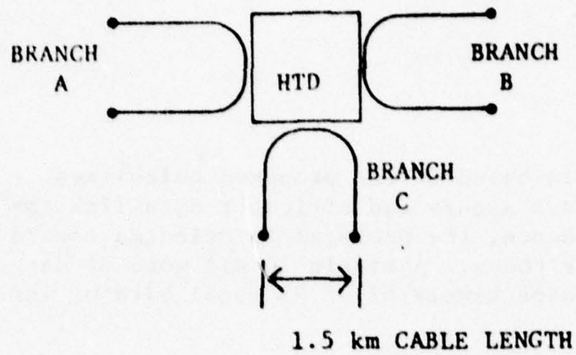
3.3 Transmission Speed

The TDC Hiway operates at a raw bit rate of 250,000 bits per second.

3.4 Transmission Distance

The Hiway Traffic Director supports three radial branches. Each branch may be 1.5 km (5,000 ft.) long, so that a total end to end distance of 3.0 km may be configured. Figure 1 illustrates the Hiway System.

FIGURE 1: HIWAY CONFIGURATION



NOTE: Each branch may have up to 1.5 km of cable, HTD may be at the end, or anywhere in the middle of the branch.

3.5 Number of Stations Per Dataway

The TDC Hiway supports 63 stations.

3.6 Number of Devices Per Station

The TDC Hiway supports stations having varying numbers of devices (Sec. 3.7). Up to 1024 locations are directly addressable in each device. Additional locations may be addressed indirectly.

3.7 Types of Devices

The TDC Hiway supports these type devices: 1) Process Controllers; 2) Process Interface Units; 3) Operator Stations; and 4) Computer Interfaces.

The characteristics of these devices are:

1) Process Controllers: This device provides eight loops of process control, with sixteen analog inputs and eight analog outputs.

2) Process Interface Units: These devices provide analog inputs (high level and low level), analog outputs, digital inputs and digital outputs.

3) Operator Station: This device performs the functions required for operator control of the process. It includes a CRT display and functional keyboard. Specialized displays are provided for overview of control loops, group of control loops, and detail of a single loop as well as other configurations.

4) Computer Interfaces: These devices allow computers to access information from the other devices located on the Hiway. The computer may be performing direct digital control or supervisory control type functions.

3.8 Mode of Transmission

Bit Serial
Half Duplex

3.9 Priority Control of Transmission

1) Bus Master devices such as Computer Interfaces and Operator Stations obtain use of the Hiway by means of Preferred Access lines connected to the Hiway Traffic Director. Access is reassigned when traffic is quiescent for 80 µsec.

2) Asynchronus Event Detection is communicated by the Process Interface Unit utilizing polling and callup commands over the Data Hiway from the Hiway Traffic Director. These polling commands occur when communications is quiescent 600 µs or is scheduled after 10 ms if traffic does not have 600 µs gap.

3.10 Response Time

Asynchronus events are serviced nominally at 10 ms intervals.

3.11 Modulation and Synchronization

Data is transmitted using the base band modulation technique shown in Figure 2.

3.12 Type of Connection

Coaxial Cable with Tee Connector.

Three twisted pair for preferred access, with screw terminal connections.

3.13 Expandability

The TDC Hiway can be expanded with no effect on operations.

3.14 Environmental Conditions*

1) Temperature : -20°C to 80°C

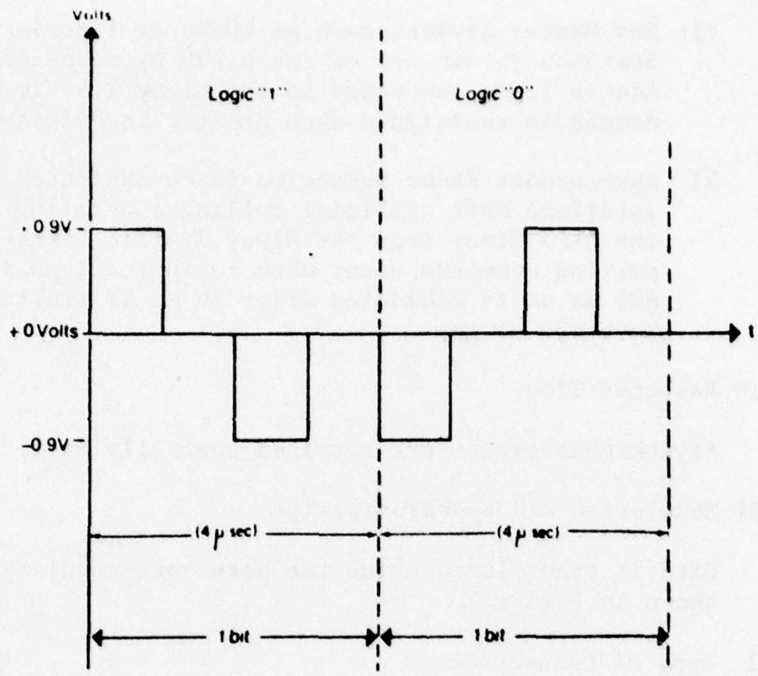
2) Humidity : Full Weather Cable

3) Earth Potential : 250V

4) Voltage Breakdown : 250V

* Applies to cable system.

FIGURE 2: DATA MODULATION



Synchronization is developed from the received data on a per bit basis.

4.0 ARCHITECTURE

4.1 System Structure

1) Physical Link: The physical link is 75Ω coaxial cable with associated BNC connectors. The coupler to the individual stations is through a tee connector.

2) Communication link and logical connection are combined in the various interfaces to the TDC Data Hiway.

3) Network Control: The TDC Hiway network control functions are performed by the Hiway Traffic Director.

4) Application: The TDC Hiway supports various application protocols as exemplified by the TDC-2000 Basic Controller, and the TDC-7100 Process Interface unit.

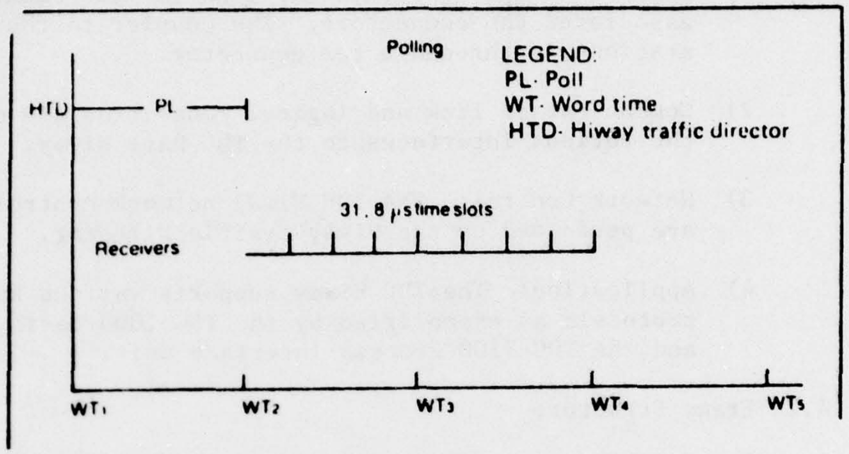
4.2 Frame Structure

1) Messages: The TDC Hiway uses messages made up of 31 bit words. The words may be utilized to perform single word read, or write messages; block read, or write messages; and control (poll or callup) messages.

   Figures 3, 4, 5 illustrate these messages.

2) Words: The format of the 31 bit words transmitted over the TDC Hiway is shown in Figure 6. Note that 5 bits are used as check bits (31, 26 BCH CODE) with the remainder available for information.

FIGURE 3: CONTROL MESSAGE FORMATS



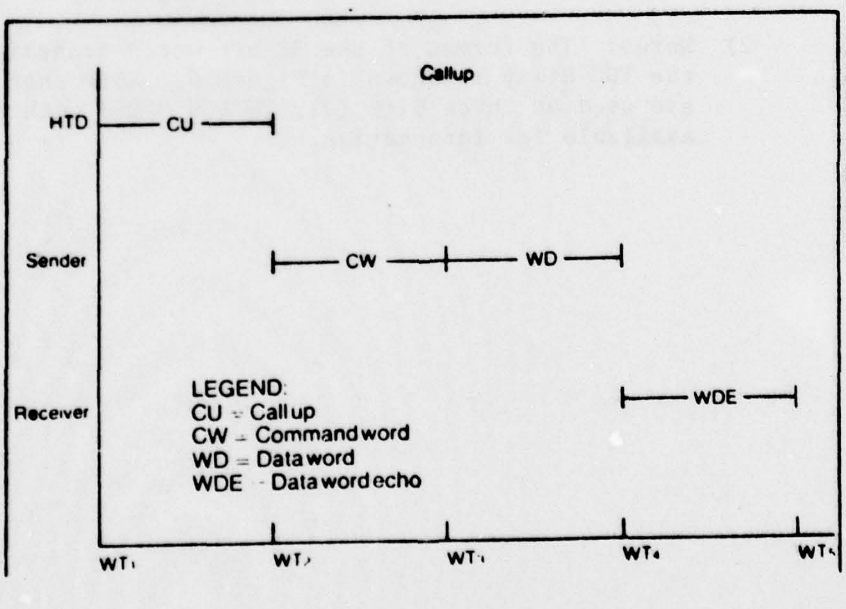
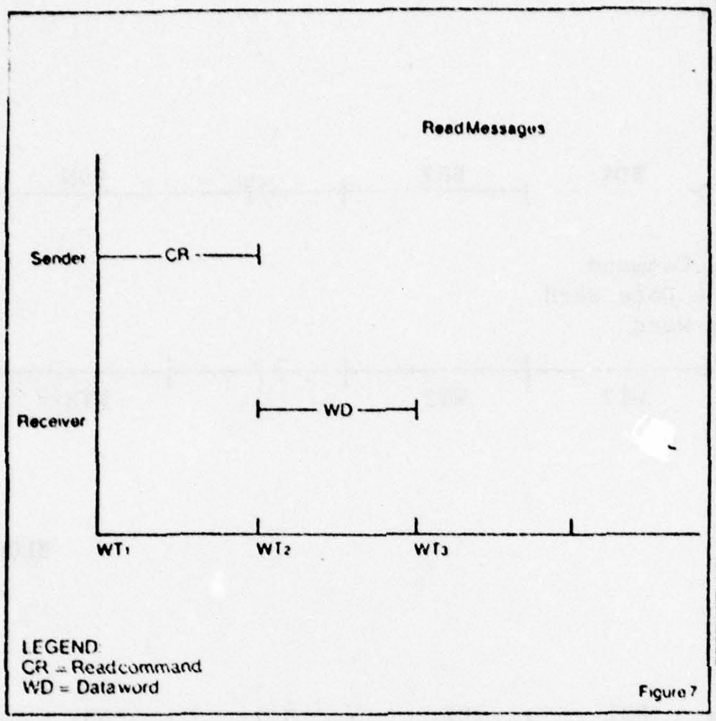
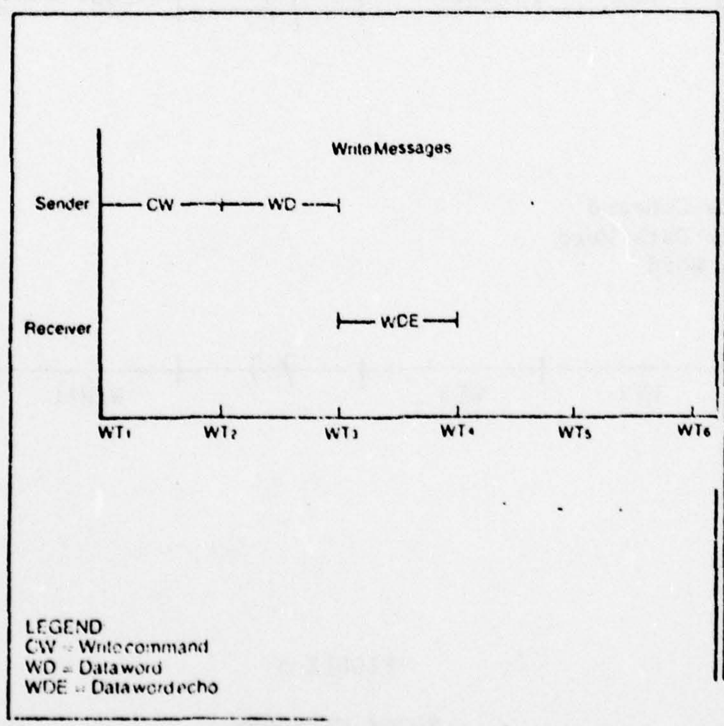
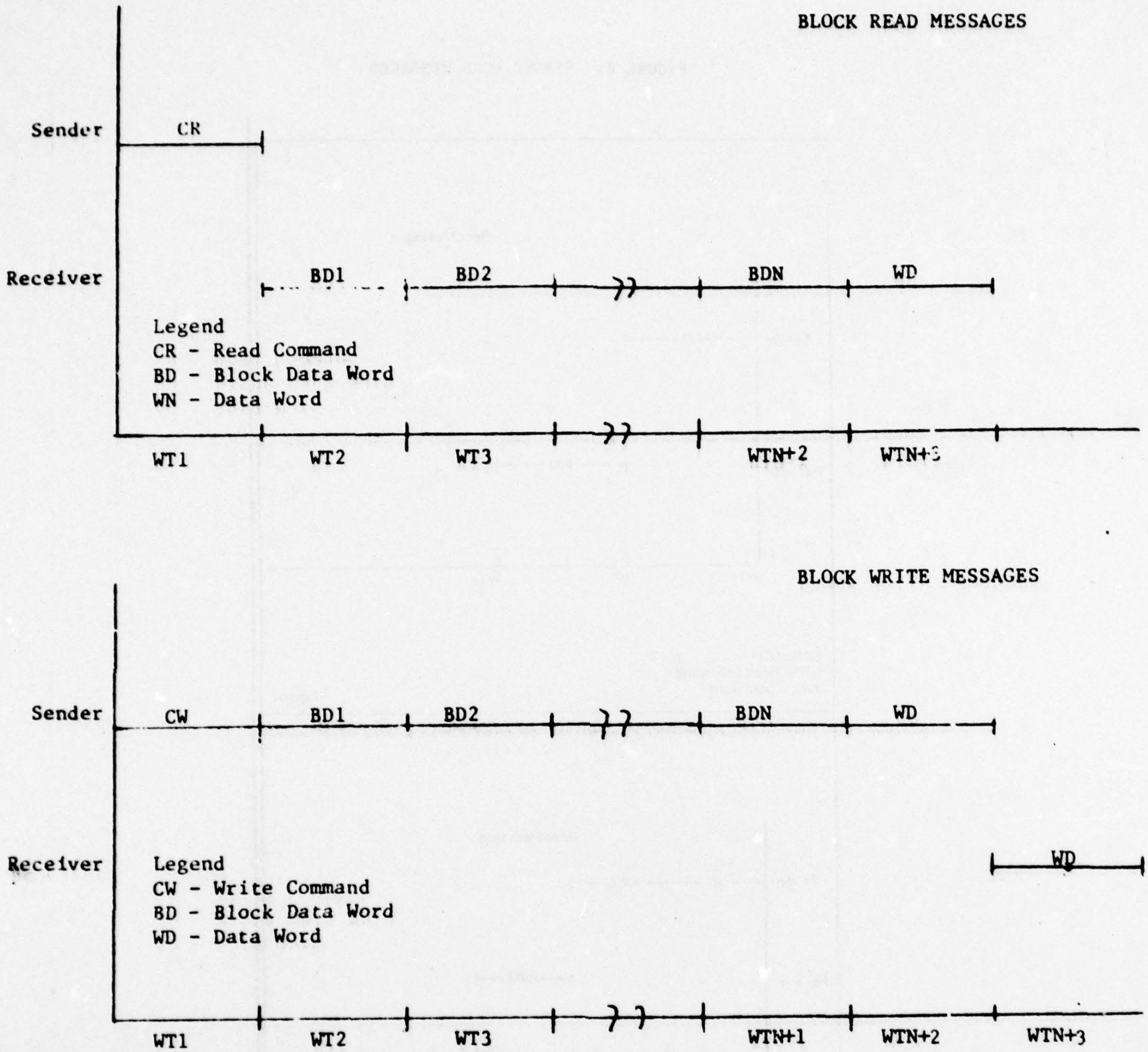


FIGURE 4: SINGLE WORD MESSAGES





BLOCK READ MESSAGES

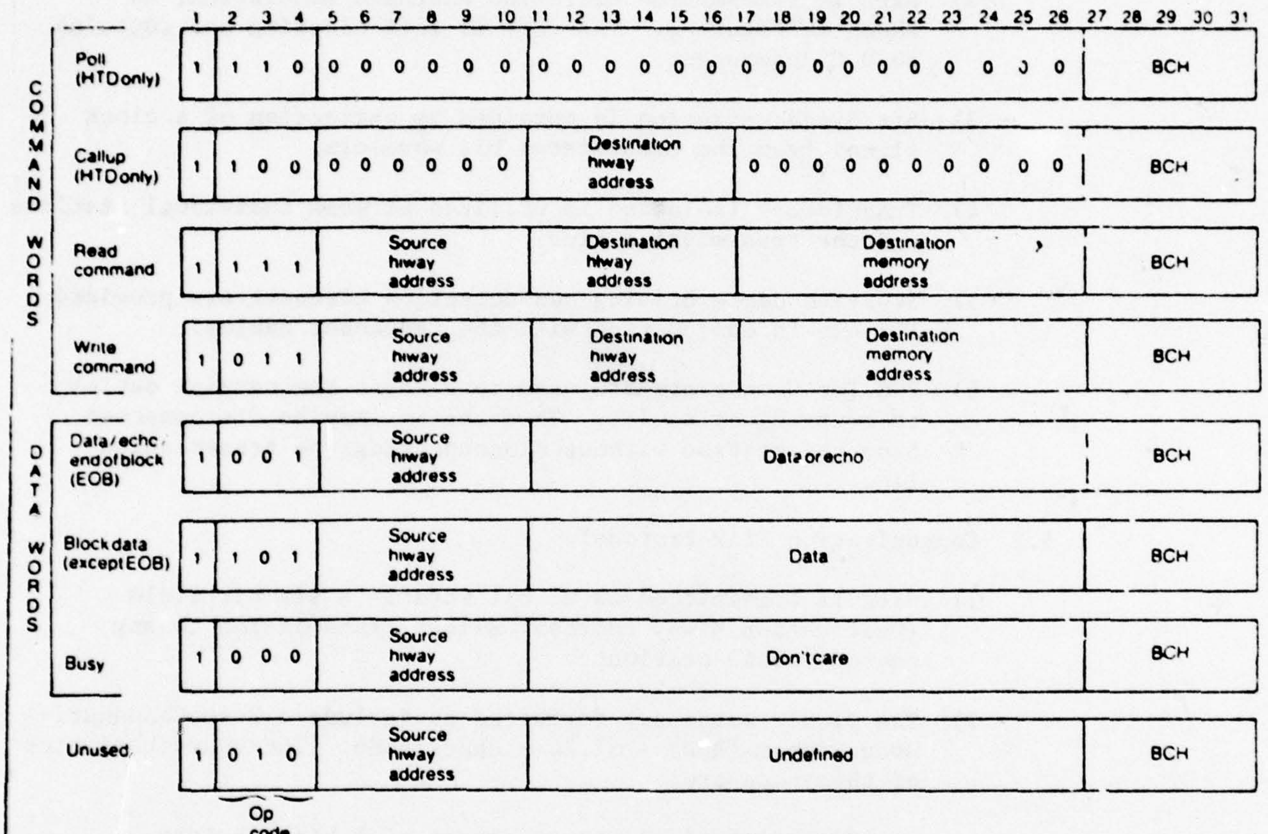


BLOCK WRITE MESSAGES

FIGURE 5

BLOCK MESSAGES

FIGURE 6: WORD FORMATS



Hiway word formats

5.0 PROTOCOL

5.1 Physical Link Protocol

1) Transmission rate 250K bits/second

2) Data is transmitted utilizing baseband modulation, as shown in Figure 2. The data is self-clocking and contains no D.C. component.

3) Bit Synchronization is obtained by extraction of a clock signal from the transmitted bit waveform.

4) Transformer isolation is utilized between individual stations and the transmission line.

5) Separate cable driving and detection circuits are provided for use in conjunction with the redundant cables.

6) BNC Tee Connectors are used to connect the coaxial cables to every hiway device. Thus the tee may be disconnected from the station without disconnecting the transmission line.

5.2 Communication Link Protocol

1) Data is transmitted as 31 bit words. A six bit field (Destination Hiway Address) allows transmission to any one of the 63 stations.

2) The 31 bit words are formatted to include a Bose-Chaudhuri-Hocquenghem (BCH) - 31,26 - check code. The characteristics of this code are:

   a) Detection of all burst errors of 5 bits or less.

   b) Detection of all combinations of 2 random bit errors or less are detected.

   c) Detection of all single inversions of message.

   d) Detection of 98.5% of all burst errors of length 6 bits.

   e) Detection of 97% of all burst errors of length greater than 6.

3) Error recovery is performed by retransmission under control of program in device which originated the message. Missing response and incomplete response checks are included in the hardware.

4) All preferred devices (masters) are responsible for error analysis and switching to backup transmission cable. Slave stations perform error checks and take no action on messages containing an error, they respond on the cable which is currently active.

5.3 Logical Connection Protocol

The preferred device interfaces perform the functions of requesting access to the Hiway automatically upon receipt of an instruction to transmit a message. A watchdog timer is included in the interfaces to detect failure to receive preferred access grant.

6.0 SAFETY AND SECURITY

6.1 Transmission Error Processing

1) BCH Check Code - generated and checked on all transactions.

2) Bit Correlator - receivers check all bits for presence of both pulses in every bit.

3) Start Bit - check that first bit of every word is a one.

4) Operation Code Checking - devices check that each word has proper OP CODE e.g. a data word response to a read command.

5) Automatic Retry - software functions to retry transactions which result in an error response up to three times.

6) Timeout Functions - Time checks are performed on receipt of Data Word or Echo within designated time period.

7) Dropout Detector - Missing bits are detected in the middle of words to eliminate forming 31 bit word out of two partial words.

8) Jabber Halt - A fail safe circuit designed into each station to prevent continuous transmission.

9) ECHO - a retransmission by receiver of DATA WORD (or last word of a block of data words) is performed as part of every write message to positively acknowledge the receipt.

10) Redundancy - All functions of HTD are performed in redundant logic. Coaxial cables are redundant. Cable driver and detector circuits are redundant in every station.

6.2 System Failure Detection, Protection, and Recovery.

1) Transformer Isolation - All stations are isolated from the transmission line so that a component failure does not affect system operation.

2) Switch to Redundant Hiway - The master stations can decide, based on analysis of system performance, to switch to the redundant hiway. Manual control is also provided at the HTD.

3) HTD Alarm - The HTD monitors its own polling activity which is indicative of Hiway loading to determine if the Hiway is overloaded.

4) Manual override of HTD grant signals is possible to disconnect a failed master device.

6.3 Other Features

1) Preferred Access Expander: This device allows expansion of a Data Hiway System beyond the four supported by the basic HTD.

2) Hiway Repeater: This device allows expansion of a Data Hiway System beyond the 1.5 meter limit.

3) Power Distribution: All TDC controllers and PIU's operate off a 24VDC power bus. This bus may be driven by redundant power supplies, and/or a battery backup system.

The HTD uses redundant regulators to power the separate logic functions.

7.0 MAINTENANCE

7.1 On Line Maintenance

The operator station performs an on line maintenance function for the Data Hiway. A program runs periodically to determine the status of all stations on the Hiway.

A display is generated, which can be initiated from the keyboard, to indicate which stations are not operable and the reason.

The station retries all communication errors up to three times and logs those failures as a basis for the switch decision.

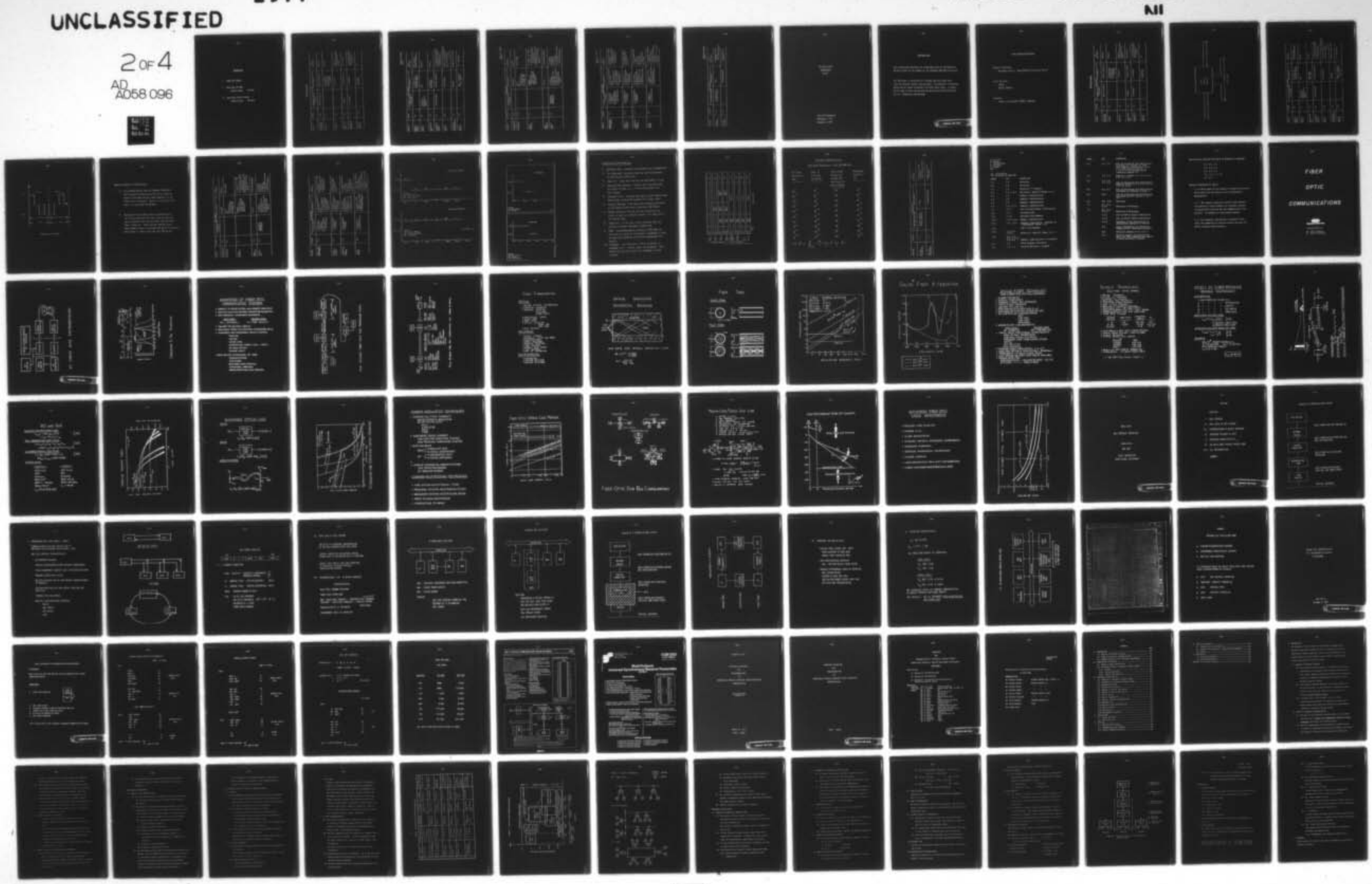
7.2 Off Line Maintenance

A Hiway Exerciser is available to manually perform operations over the Data Hiway for the purpose of validation of the Hiway operation, or maintenance of individual stations disconnected from the Hiway.

AD-A058 096 PURDUE UNIV LAFAYETTE IND PURDUE LAB FOR APPLIED IND--ETC F/G 9/2
ANNUAL MEETING OF INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL C--ETC(U)
1977 N00014-78-C-0127

UNCLASSIFIED NL

2 OF 4
AD A058 096



APPENDICIES

1. Comparison Summary

2. Data Hiway TDC 2000

   Technical Data   HO-10-01

3. Data Hiway Traffic Director

   Technical Data   HO-10-02

COMPARISON SUMMARY

FUNCTION	SECTION	REQUIREMENTS		TDC DATA HIWAY
		ISA SP72, IEC WG6 FUNCTIONAL	WG6 (JAPAN 2)	
Application Environment	2.0	Process Industries	Process Industries	Complies
Topology	3.1	--	1) Distributed 2) Single Line 3) Redundancy 4) Not Message Switching	Complies
Mastership	3.2	--	1) Multiple	Complies (Hiway Traffic Director is scheduler not master)
Transmission Speed	3.3	1 Megabit	1 Megabit minimum	250 Kilobits
Transmission Distance	3.4	2 Kilometers	4 Kilometers	1.5 Kilometers (three branches from HTD) 3.0 End to End Maximum
Number of Stations	3.5		31 Minimum 255 Desired	63

Comparison Summary
Page 2

FUNCTION	SECTION	REQUIREMENTS		TDC DATA HIWAY
		ISA SP72, IEC WG6 FUNCTIONAL	WG6 (JAPAN 2)	
Number of Devices/ Station	3.6		512 Points I/O 128 Points Control 16 Points Terminal	1024 Locations Directly Addressable
Types of Devices	3.7	Analog I/O Digital I/O Man/Machine I/F Computer	Same	Complies
Transmission Mode	3.8		Bit Serial, Half Duplex	Complies
Transmission Priority Control	3.9	Unsolicited Data	Interrupt Capability	1) Preferred Access; Communication Link Masters 2) Polling & Callup; Unsolicited Data
Response Time	3.10		20 MS Maximum	- .5 MS Nominal Dela Solicited - 20 MS Typical Maximum Unsolicited - 20 MS Typical
Modulation & Synchronization	3.11		Baseband	Baseband, Bipolar Data
Type of Coupling	3.12		Coax - T-connection Twisted Pair - Terminal Strip	Tee - Data Hiway

90

FUNCTION	SECTION	REQUIREMENTS		TDC DATA HIWAY
		ISA SP72, IEC WG6 FUNCTIONAL	WG6 (JAPAN 2)	
Expandability	3.13		Expandable	Complies
Environment	3.14		0-50°C 95% Less than 10 earth potential AC voltage, 500V DC voltage, 500V	-20°C to 80°C
System Structure	4.1		<u>5 Levels</u> Physical Link Communication Link Logical Connection Network Control Application	1) Physical Link 75Ω Coax 2) Communication & Logical connection combined in Interface Cards 3) Network Control: in HTD 4) Application Protocal: as required by various devices.
Frame Structure	4.2		H.D.L.C.	31 Bit Word Oriented
Physical Link	5.1	Transparent, Dependent on Line Physics	1) 1 MBPS 2) Bit Synch 3) Clock Extraction 4) Base Band 5) Electrical Isolation 6) Redundant Paths 7) Bypassing & Dis-connect	1) 250 KBPS 2) Complies 3) Complies 4) Complies 5) Transformer 6) Cables & Driver Detector Circuitry 7) BNC TEE Connectors

FUNCTION	SECTION	REQUIREMENTS		TDC DATA HIWAY
		ISA SP72, IEC WG6 FUNCTIONAL	WG6 (JAPAN 2)	
Communication Link Protocol	5.2	1) Binary Data 2) Shared Error Checking 3) Data Blocks 4) Global Messages 5) Solicited & Unsolicited Data	1) n:m (n:m) Capability 2) HDLC Based 3) CRC 4) Error Recovery by Retransmission 5) Multiple Watchdogs 6) Global Messages 7) Initialization Capability	1) 31 Bit Word Oriented 2) 31/26 BCH 3) Retransmission under Program Control 4) All masters check and may switch
Logical Connection Protocol	5.3	1) Alternate Links 2) Transfer of Line Master	1) Arbitration 2) Establish Channels 3) Watchdog Timer 4) Distributed Intelligence	1) Preferred Access 2) Preferred Access 3) In Devices 4) Yes
Transmission Error Checking	6.1	1) Error Detection 2) Frame Sync Security	1) CRC & Retry 2) Status Sensing 3) Response Interval 4) Frame Sync 5) Correction of loss of Sync 6) Data Sequence Check	1) BCH Correlator and Others 2) Complies 3) Complies 4) Complies 5) By Rejection & Retransmission 6) Complies
System Failure	6.2	1) Single Unit Isolation 2) High Availability	1) Automatic Station Bypass 2) Manual Station Bypass 3) Reorganization 4) Self-check Software	1) Transformer Isolation, Jabber Halt 2) Switch on HTD or disconnect cables 3) Switchover by Preferred Devices 4) In Operator Station and Computer

Comparison Summary
Page 5

FUNCTION	SECTION	REQUIREMENTS		TDC DATA HIWAY
		ISA SP72, IEC WG6	WG6 (JAPAN 2)	
On Line Maintenance	7.1	On Line Testing & Fault Diagnosis	Same	1) Operator Station Software Hiway Status Checking
Off Line Maintenance	7.2		Maintenance Panel	2) Hiway Exerciser Available

TDC DATA HIWAY COMPARISON STUDY

CHUCK DIEFENDERFER
HONEYWELL / PCD
OCTOBER 3, 1977

INTRODUCTION

THIS COMPARISON DESCRIBES THE CHARACTERISTICS OF THE HONEYWELL TDC DATA HIWAY IN THE FORMAT OF THE JAPANESE PROPOSED GUIDELINE.

THE TDC HIWAY IS OPTIMIZED AS A SECURE AND EFFICIENT DATA LINK FOR PROCESS CONTROL APPLICATIONS. THE PROTOCOL IS ORIENTED AROUND 31 BIT WORDS INCLUDING 5 BIT BCH CHECK CODES. A SINGLE 16 BIT WORD OF DATA CAN BE READ OR WRITTEN WITH 63 OR 93 TOTAL BITS OF INFORMATION TRANSFERRED.

HIWAY TRAFFIC DIRECTOR

PRIORITY SCHEDULER

PREFERRED ACCESS: REQUEST/GRANT POLLING & CALLUP

CABLE SELECTOR

MANUAL

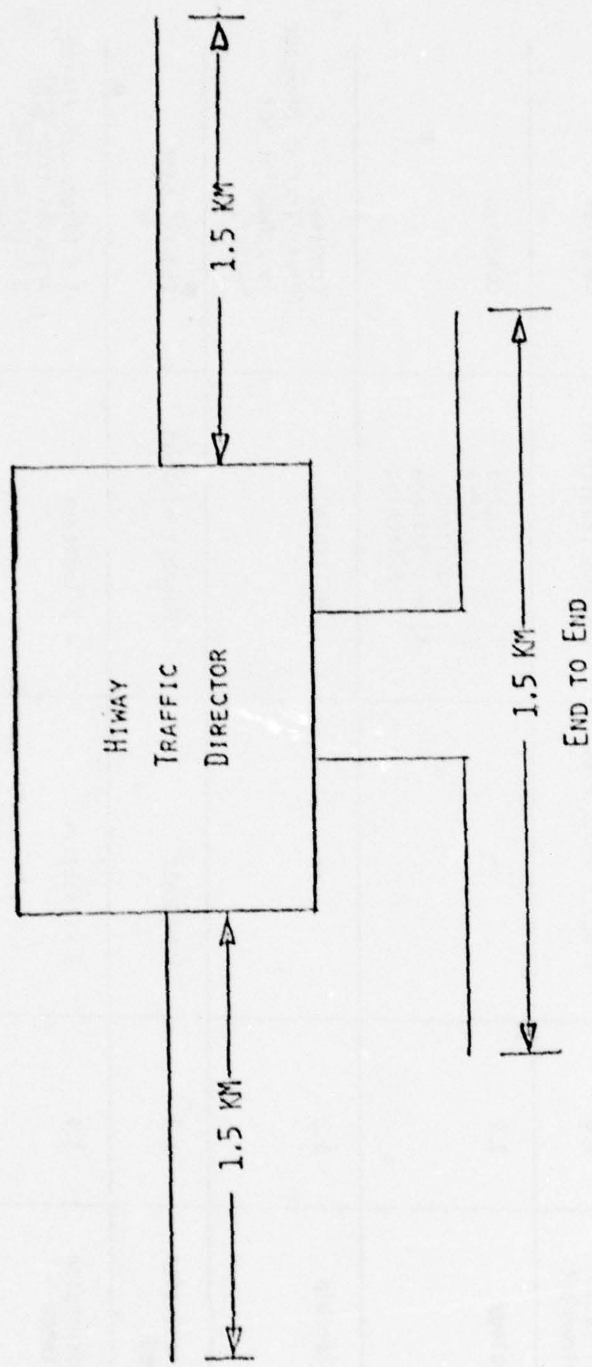
REMOTE CONTROL

REPEATER

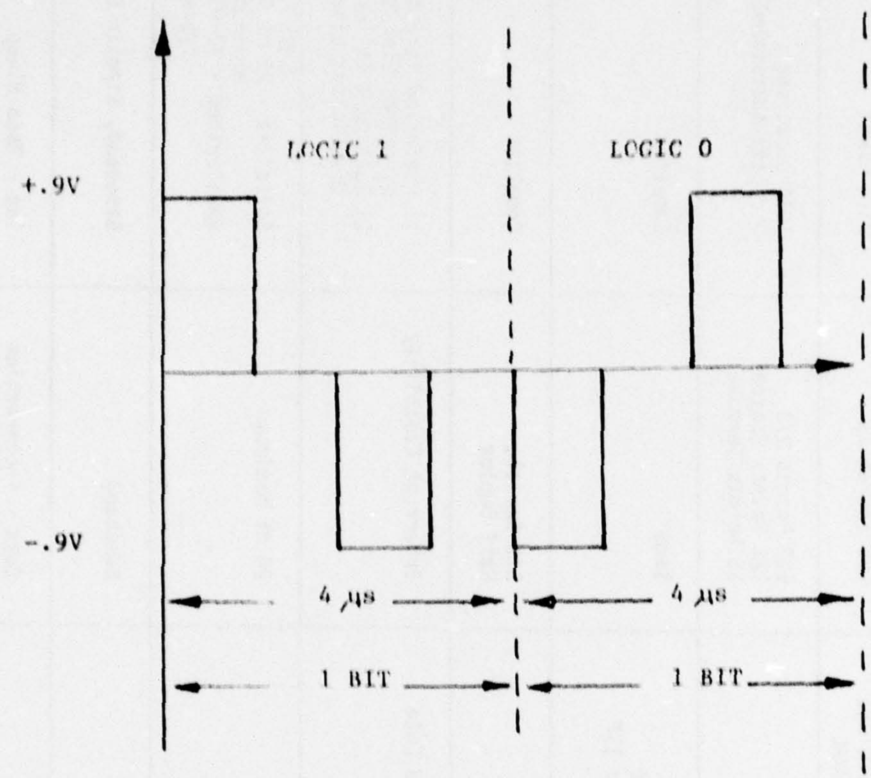
THREE 1.5 KILOMETER (5000') BRANCHES

COMPARISON SUMMARY

FUNCTION	SECTION	REQUIREMENTS: ISA SP72, IEC WG6 FUNCTIONAL	REQUIREMENTS: WG6 (JAPAN 2)	TDC DATA HIWAY
Application Environment	2.0	Process Industries	Process Industries	Complies
Topology	3.1	--	1) Distributed 2) Single Line 3) Redundancy 4) Not Message Switching	Complies
Mastership	3.2	--	1) Multiple	Complies (Hiway Traffic Director is scheduler not master)
Transmission Speed	3.3	1 Megabit	1 Megabit minimum	250 Kilobits
Transmission Distance	3.4	2 Kilometers	4 Kilometers	1.5 Kilometers (three branches from HTD) 3.0 End to End Maximum
Number of Stations	3.5		31 Minimum 255 Desired	63


1.5 KM
HIWAY
TRAFFIC
DIRECTOR
1.5 KM
END TO END
1.5 KM

FUNCTION	SECTION	REQUIREMENTS		TDC
		ISA SP72, IEC WG6 FUNCTIONAL	WG6 (JAPAN 2)	DATA HIWAY
Number of Devices/ Station	3.6		512 Points I/O 128 Points Control 16 Points Terminal	1024 Locations Directly Addressable
Types of Devices	3.7	Analog I/O Digital I/O Man/Machine I/F Computer	Same	Complies
Transmission Mode	3.8		Bit Serial, Half Duplex	Complies
Transmission Priority Control	3.9	Unsolicited Data	Interrupt Capability	1) Preferred Access; Comm-unication Link Masters 2) Polling & Callup; Unsolicited Data
Response Time	3.10		20 MS Maximum	- .5 MS Nominal Dela Solicited - 20 MS Typical Maximum Unsolicited - 20 MS Typical
Modulation & Synchronization	3.11		Baseband	Baseband, Bipolar Data
Type of Coupling	3.12		Coax - T-connection Twisted - Terminal Pair Strip	Tee - Data Hiway



TRANSMITTED DATA FORMAT

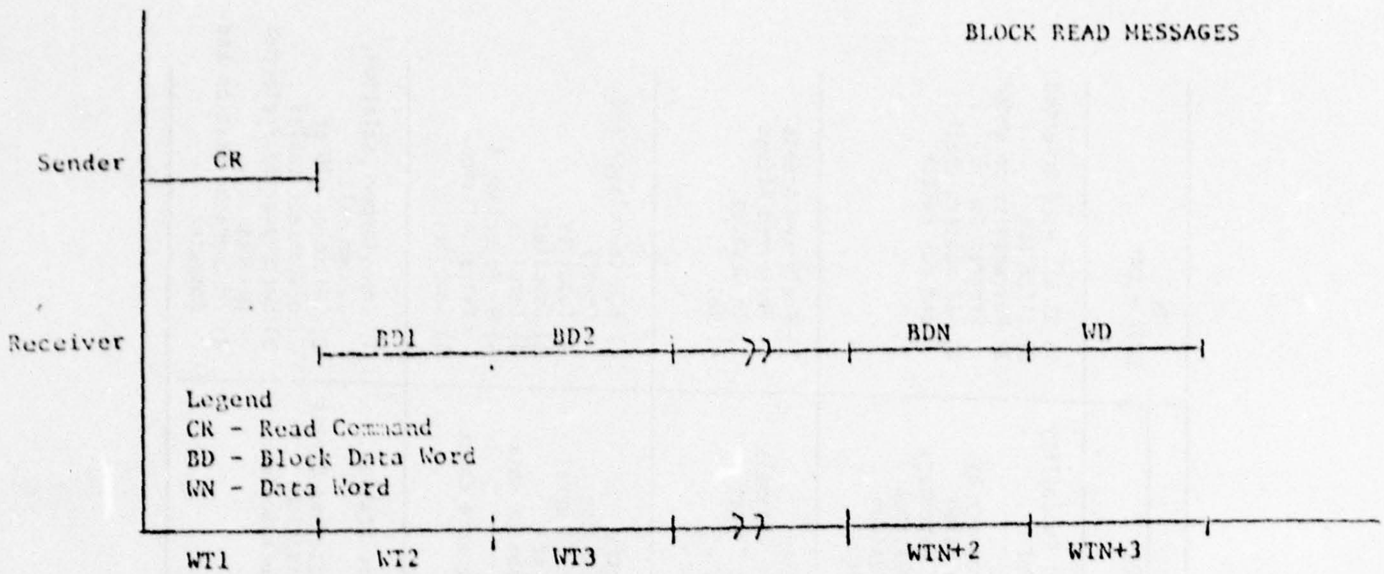
PRIORITY CONTROL OF TRANSMISSION

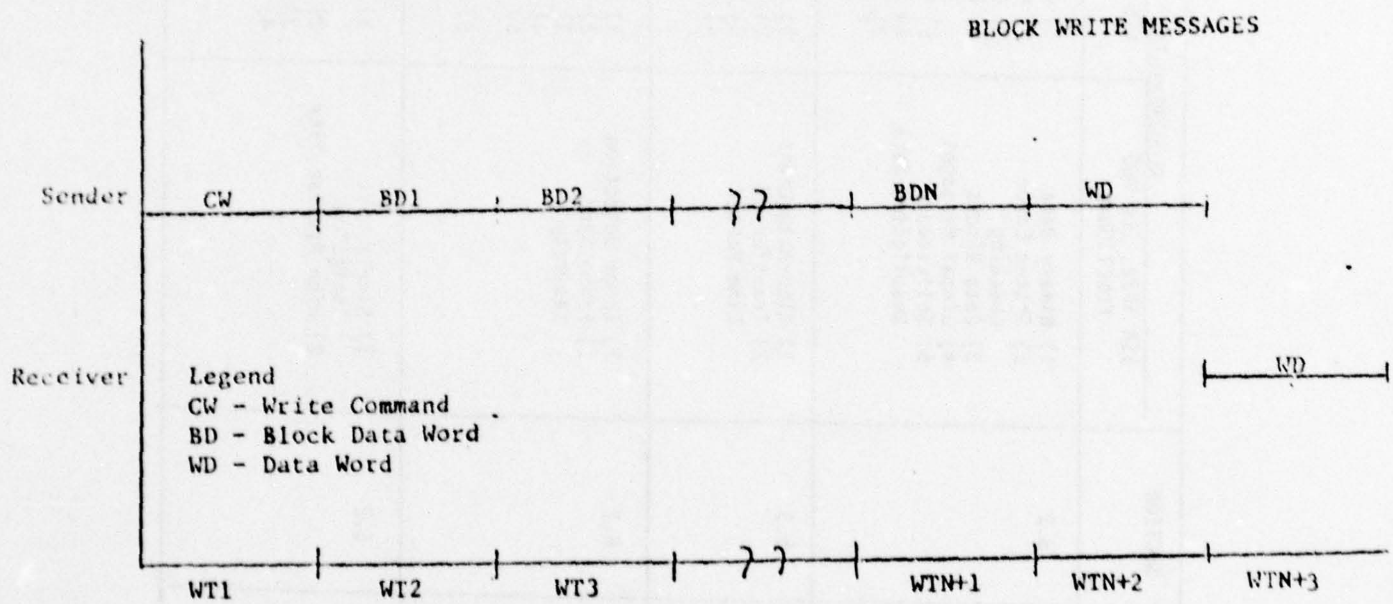
1) LINK MASTER DEVICES SUCH AS COMPUTER INTERFACES AND OPERATOR STATIONS OBTAIN USE OF THE HIWAY BY MEANS OF PREFERRED ACCESS LINES CONNECTED TO THE HIWAY TRAFFIC DIRECTOR. ACCESS IS REASSIGNED WHEN TRAFFIC IS QUIESCENT FOR 80 μSEC.

2) ASYNCHRONOUS EVENT DETECTION IS COMMUNICATED BY THE PROCESS INTERFACE UNIT UTILIZING POLLING AND CALLUP COMMANDS OVER THE DATA HIWAY FROM THE HIWAY TRAFFIC DIRECTOR. THESE POLLING COMMANDS OCCUR WHEN COMMUNICATIONS IS QUIESCENT 600 μS OR IS SCHEDULED AFTER 10 MS IF TRAFFIC DOES NOT HAVE 600 μS GAP.

FUNCTION	SECTION	REQUIREMENTS: ISA SP72, IEC WG6 FUNCTIONAL	REQUIREMENTS: WG6 (JAPAN 2)	TDC DATA HIWAY
Expandability	3.13		Expandable	Complies
Environment	3.14		0-50°C 95% Less than 10 earth potential AC voltage, 500V DC voltage, 500V	-20°C to 80°C
System Structure	4.1		5 Levels Physical Link Communication Link Logical Connection Network Control Application	1) Physical Link 75Ω Coax 2) Communication & Logical connection combined in Interface Cards 3) Network Control: in HTD 4) Application Protocol: as required by various devices.
Frame Structure	4.2		H.D.L.C.	31 Bit Word Oriented
Physical Link	5.1	Transparent, Dependent on Line Physics	1) 1 MBPS 2) Bit Synch 3) Clock Extraction 4) Base Band 5) Electrical Isolation 6) Redundant Paths 7) Bypassing & Dis-connect	1) 250 KBPS 2) Complies 3) Complies 4) Complies 5) Transformer 6) Cables & Driver Detector Circuitry 7) BNC TEE Connectors

FUNCTION	SECTION	REQUIREMENTS		TDC DATA HIWAY
		ISA SP72, IEC WG6 FUNCTIONAL	WG6 (JAPAN 2)	
Communication Link Protocol	5.2	1) Binary Data 2) Shared Error Checking 3) Data Blocks 4) Global Messages 5) Solicited & Unsolicited Data	1) n:m (n:m) Capability 2) HDLC Based 3) CRC 4) Error Recovery by Retransmission 5) Multiple Watchdogs 6) Global Messages 7) Initialization Capability	1) 31 Bit Word Oriented 2) 31/26 BCH 3) Retransmission under Program Control 4) All masters check and may switch
Logical Connection Protocol	5.3	1) Alternate Links 2) Transfer of Line Master	1) Arbitration 2) Establish Channels 3) Watchdog Timer 4) Distributed Intelligence	1) Preferred Access 2) Preferred Access 3) In Devices 4) Yes
Transmission Error Checking	6.1	1) Error Detection 2) Frame Sync Security	1) CRC & Retry 2) Status Sensing 3) Response Interval 4) Frame Sync 5) Correction of loss of Sync 6) Data Sequence Check	1) BCH Correlator and Others 2) Complies 3) Complies 4) Complies 5) By Rejection & Retransmission 6) Complies
System Failure	6.2	1) Single Unit Isolation 2) High Availability	1) Automatic Station Bypass 2) Manual Station Bypass 3) Reorganization 4) Self-check Software	1) Transformer Isolation, Jabber Halt 2) Switch on HTD or disconnect cables 3) Switchover by Preferred Devices 4) In Operator Station and Computer


BLOCK READ MESSAGES
Sender
CR
Receiver
BD1
BD2
BDN
WD
Legend
CR - Read Command
BD - Block Data Word
WN - Data Word
WT1
WT2
WT3
WTN+2
WTN+3


BLOCK WRITE MESSAGES
Sender
CW
BD1
BD2
BDN
WD
Receiver
WD
Legend
CW - Write Command
BD - Block Data Word
WD - Data Word
WT1
WT2
WT3
WTN+1
WTN+2
WTN+3

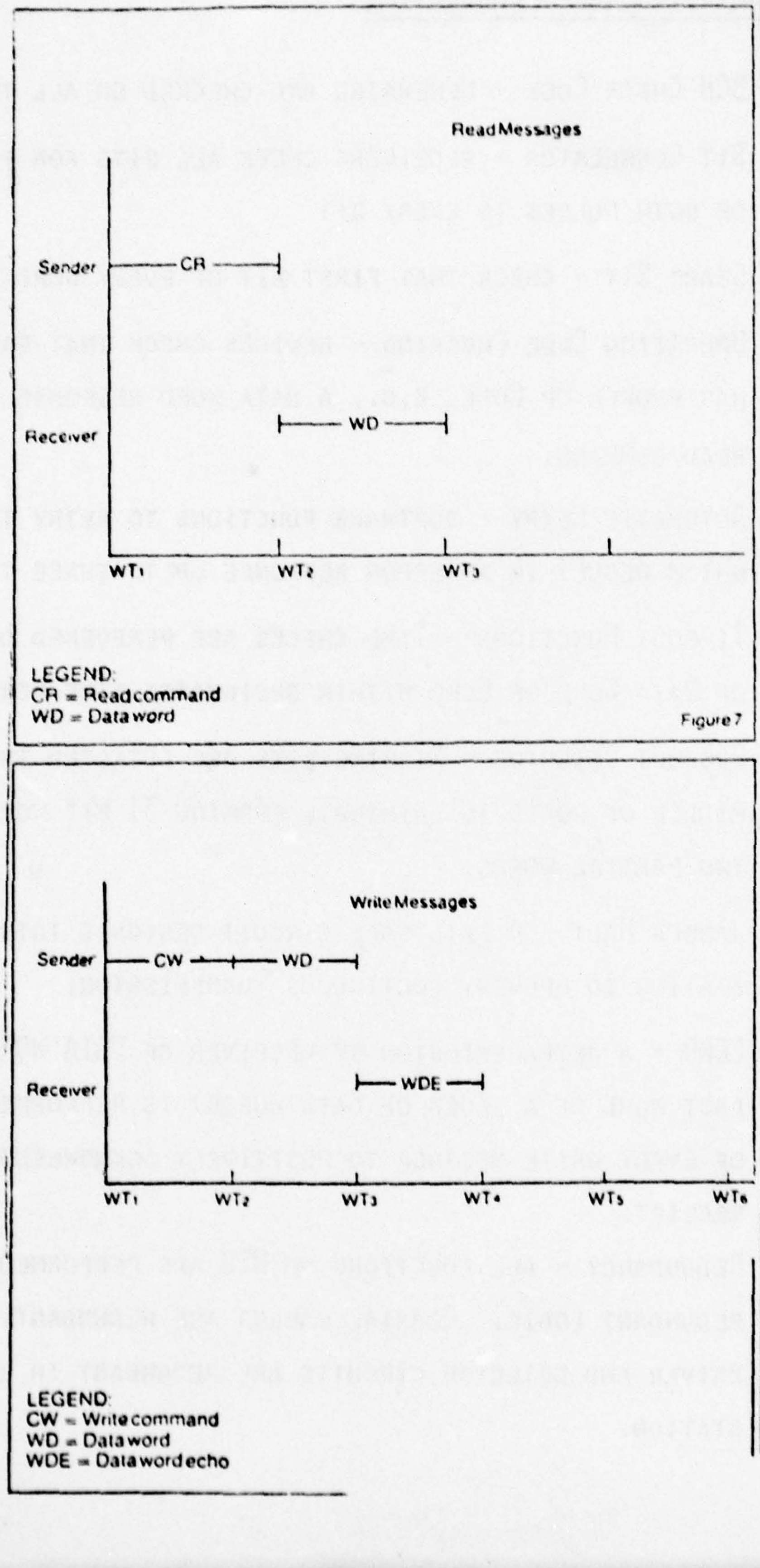

Read Messages
Sender
CR
Receiver
WD
WT1
WT2
WT3
LEGEND:
CR = Read command
WD = Data word
Write Messages
Sender
CW
WD
Receiver
WDE
WT1
WT2
WT3
WT4
WT5
WT6
LEGEND:
CW = Write command
WD = Data word
WDE = Data word echo

Figure 7

Transmission Error Processing

1) BCH Check Code - generated and checked on all transactions.

2) Bit Correlator - receivers check all bits for presence of both pulses in every bit.

3) Start Bit - check that first bit of every word is a one.

4) Operation Code Checking - devices check that each word has proper Op Code, e.g., a data word response to a read command.

5) Automatic Retry - software functions to retry transactions which result in an error response up to three times.

6) Timeout Functions - Time checks are performed on receipt of Data Word or Echo within designated time period.

7) Dropout Detector - Missing bits are detected in the middle of words to eliminate forming 31 bit word out of two partial words.

8) Jabber Halt - A fail safe circuit designed into each station to prevent continuous transmission.

9) ECHO - a retransmission by receiver of DATA WORD (or last word of a block of data words) is performed as part of every write message to positively acknowledge the receipt.

10) Redundancy - all functions of HTD are performed in redundant logic. Coaxial cables are redundant. Cable driver and detector circuits are redundant in every station.

	Word	1	2	3	4	5–10	11–16	17–26	27–31
COMMAND WORDS	Poll (HTD only)	1	1	1	0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	BCH
	Callup (HTD only)	1	1	0	0	0 0 0 0 0 0	Destination hiway address	0 0 0 0 0 0 0 0 0 0	BCH
	Read command	1	1	1	1	Source hiway address	Destination hiway address	Destination memory address	BCH
	Write command	1	0	1	1	Source hiway address	Destination hiway address	Destination memory address	BCH
DATA WORDS	Data/echo/end of block (EOB)	1	0	0	1	Source hiway address	Data or echo		BCH
	Block data (except EOB)	1	1	0	1	Source hiway address	Data		BCH
	Busy	1	0	0	0	Source hiway address	Don't care		BCH
	Unused	1	0	1	0	Source hiway address	Undefined		BCH

Bits 2–4: Cp code

Hiway word formats

DATA HIWAY ERROR DETECTION

WORD ERROR PROBABILITY - (31, 26) BCH CODE

BIT ERROR PROBABILITY (PE)	PROB. OF 3 OR MORE BIT ERRORS	"WORST CASE" PROBABILITY (31,26) BCH CODE FAILS TO DETECT ERRORS	WORD ERROR PROBABILITY (P)
10^{-1}	10^{-1}	.03	10^{-2}
10^{-2}	10^{-3}	.03	10^{-4}
10^{-3}	10^{-6}	.03	10^{-7}
10^{-4}	10^{-9}	.03	10^{-10}
10^{-5}	10^{-12}	.03	10^{-13}
10^{-6}	10^{-15}	.03	10^{-16}
10^{-7}	10^{-18}	.03	10^{-19}
10^{-8}	10^{-21}	.03	10^{-22}
10^{-9}	10^{-24}	.03	10^{-25}
10^{-10}	10^{-27}	.03	10^{-28}

$$P \cong 2^{-5} \times \sum_{J=3}^{31} \binom{31}{J} P_E^{J} (1 - P_E)^{31-J}$$

FUNCTION	SECTION	REQUIREMENTS		TDC DATA HIWAY
		ISA SP72, IEC WG6	WG6 (JAPAN 2)	
On Line Maintenance	7.1	On Line Testing & Fault Diagnosis	Same	1) Operator Station Software Hiway Status Checking
Off Line Maintenance	7.2		Maintenance Panel	2) Hiway Exerciser Available

M-H 9/23/77
R. Sander
C. Diefenderfer
S. Korowitz
C. Farmer
RSC

(1) Connections
(2) as given in IEEE 488

JAPAN	IEC	COMPARISON
1.0	1.0	Narrative
2.0	2.0	Narrative
3.1	4.0	Narrative + Schematic
3.2	4.2, 7.6.2	Narrative + Specific Answer to 4.2
3.3	3.4	Numbers + Qualification
3.4	3.4	Numbers + Qualification
3.5	4.3	Numbers + Qualification
3.6	4.3, 3.1.1	Numbers + Qualification
3.7	3.1.1	List of Device Types + Descriptions
3.8	7.4	Attribute List
3.9	7.5.9	Narrative with Numbers
3.10	7.5.9, 11.2	Number + Qualification
3.11	7.4, 7.4.3	Diagram + Description - Redefine as Transmission Media + (1)
3.12		List + Descriptions
3.13	4.3, 4.4, 7.5.6	Narrative + Specific Answer to 4.4
3.14	2.4, 2.5, 3.4.1, 3.4.2, 8.2, 8.3	Numbers + Qualification + Statements
4.1	7.1	Block Diagram + Narrative
4.2	7.4, 7.5	Detailed Narrative + Diagram

JAPAN	IEC	COMPARISON
5.1	7.3, 7.4	List of Attributes with Description-including all elements of Japan plus physical configuration 7.3 to 7.3.4 with explicit answers to 7.4.1 to 7.4.6 including state transition diagram (2)
5.2	7.3, 7.5	Same as 5.1 except 7.3.1 to 7.3.4 & 7.5.1 to 7.5.9
5.3	7.3, 7.6, 7.7, 7.8	List of Attributes with Description + Specific Answers to 7.3.1 to 7.3.4 & 7.6.1 & 7.6.4
6.1	9.1, 9.2	List of Attributes with Descriptions including all elements of Japan 2
6.2	10.0	List of Attributes with Descriptions including specific answers to 10.1 - 10.4
6.3	8.0, 10.0	See below
7.1	6.1, 6.2, 3.1.4	Narrative & Philosophy
7.2	6.1, 6.2, 3.1.4	Narrative & Philosophy
	3.1.2	List of Device Types & Description
	3.1.3	List of Device Types & Description
	3.1.5	Statement of any restrictions on simultaneous operation of devices
	5.0	State avoidability of translators Common carrier interfaces available
	8.0	Narrative answers to 8.1 to 8.3
	11.1	Specific example calculations for short and long transfers from Master to Slave and Slave to Master

IEC/SC65A/NC6 SECTIONS FOR WHICH NO RESPONSE IS REQUIRED

1.1, 1.2, 1.3

2.1, 2.2, 2.3

3.2, 3.3, 3.5

7.1, 7.2, 7.7, 7.8

4.1, 4.3

SPECIFIC OBJECTIONS TO IEC/SC

7.3.2 cannot apply to all aspects of physical protocols. State explicitly the nature of all time/distance dependencies.

7.6.5 The logical connection protocol must address the question of data integrity on transfers between the application protocol and the communication link protocol. An example is check before operate.

11.2 Give examples calculations of response times under the assumption of various traffic patterns and device response characteristics.

FIBER

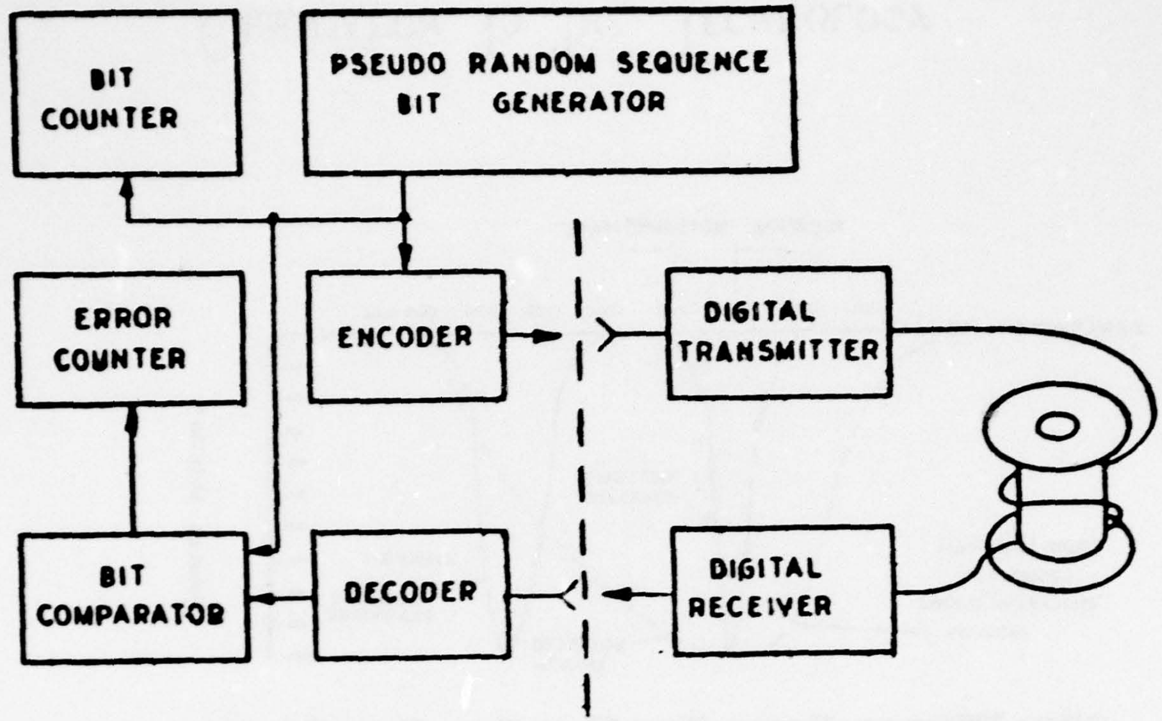
OPTIC

COMMUNICATIONS

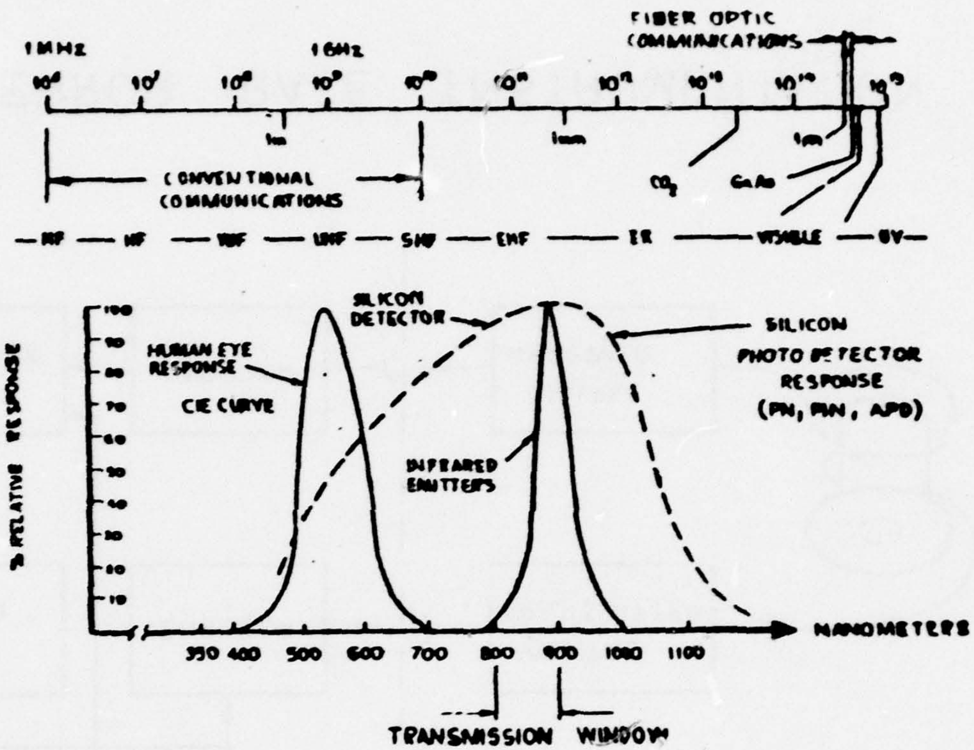
Galileo Electro-Optics Corp.

Presentation by

Mr. Carl Podlesny
Mr. Rodney Andersen



BIT ERROR RATE INSTRUMENTATION



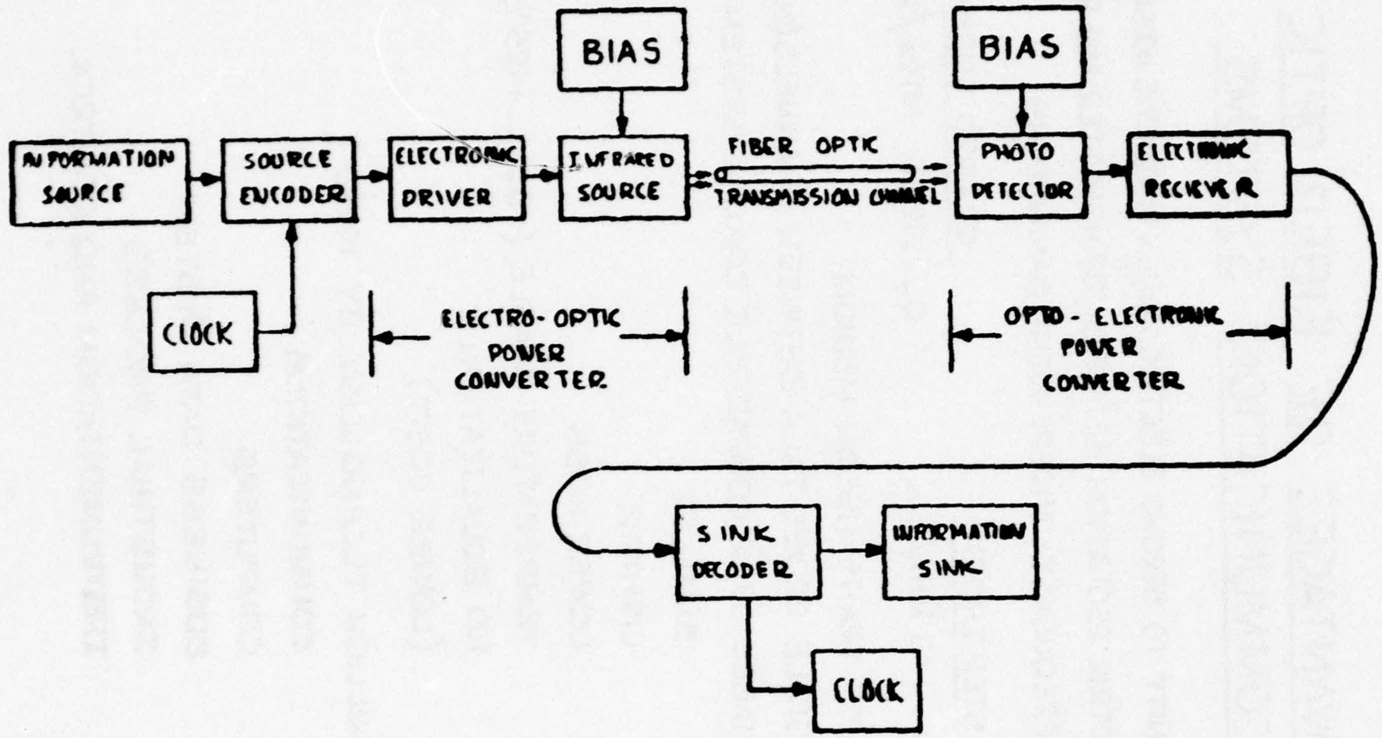
ORIENTATION TO THE TECHNOLOGY

ADVANTAGES OF FIBER OPTIC COMMUNICATION SYSTEMS

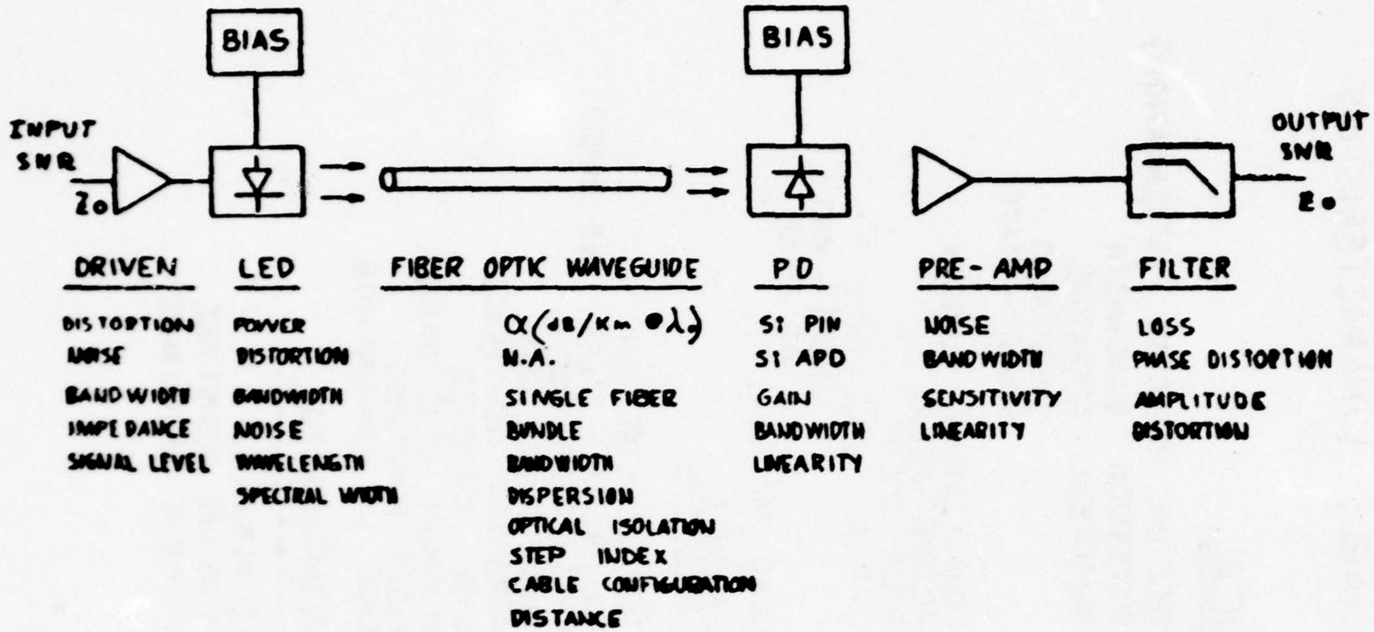
- IMMUNITY TO STRONG ELECTRIC AND/OR MAGNETIC NOISE FIELDS
- DIELECTRIC ISOLATION BETWEEN TRANSMITTER AND RECEIVER
- WIDE FREQUENCY-INDEPENDENT BANDWIDTHS

STEP INDEX	GRADED INDEX
20...40 MHz/km	0...1200 MHz/km

- "SECURE" TRANSMISSON MEDIUM
- NEGLIGIBLE CROSS-TALK BETWEEN CHANNELS (60...80 dB)
- ADVANTAGES OVER COMPARABLE COAXIAL SYSTEMS
  - SMALLER
  - LIGHTER
  - LOWER LOSS
  - TEMPERATURE STABLE (-55...+125°C)
  - NO EQUALIZATION
  - (LOWER COST)
- $100 MILLION TECHNOLOGY BY 1980
  - COMMUNICATIONS
  - COMPUTERS
  - BUSINESS DATA SYSTEM
  - INDUSTRIAL PROCESS
  - INSTRUMENTATION AND CONTROL



NEAR INFRARED FIBER OPTIC DATA TRANSMISSION SYSTEM



TYPICAL BASEBAND FIBER OPTIC COMMUNICATIONS LINK (ANALOG OR DIGITAL)

CABLE CHARACTERISTICS

OPTICAL

- SPECTRAL RESPONSE (ATTENUATION)
- DISPERSION BANDWIDTH
- NUMERICAL APERATURE
  - CALCULATED
  - STEADY - STATE
- EQUALIZATION LENGTH
- ACTIVE AREA
- FIBER COUNT
- OPTICAL ISOLATION
  - NEAR - END
  - FAR - END
- CLAD MATERIAL

MECHANICAL

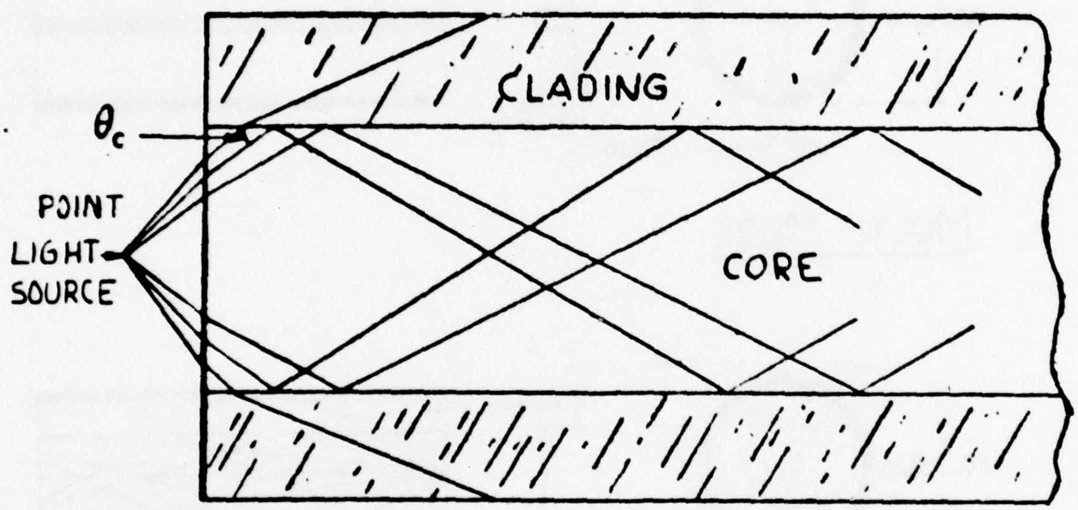
- TENSILE STRENGTH
- FLEXIBILITY (TWIST AND BEND)
- CRUSH RESISTANCE
- IMPACT RESISTANCE
- ABRASION RESISTANCE
- CABLE DIAMETER
- MINIMUM BEND RADIUS
- FIBER BUFFERING
- EASE OF TERMINATION

ENVIRONMENTAL

- TEMPERATURE
- MOISTURE RESISTANCE
- CHEMICAL RESISTANCE

OPTICAL WAVEGUIDE

NUMERICAL APERTURE



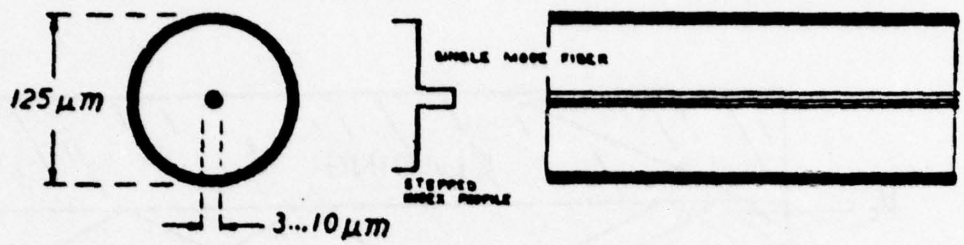
WHEN $\theta \leq \theta_c$ TOTAL INTERNAL REFLECTION OCCURS

$$\theta_c = \cos^{-1} \frac{n \text{ CLAD}}{n \text{ CORE}}$$

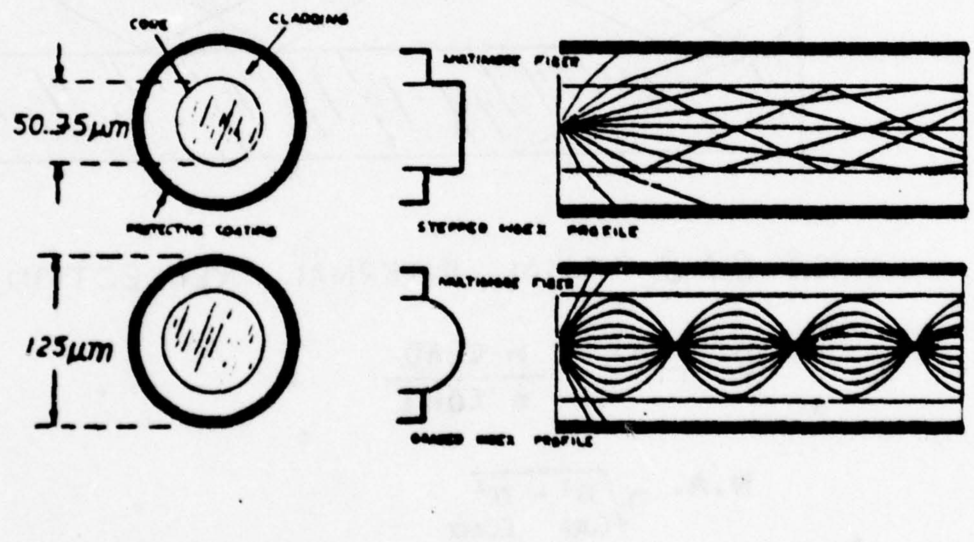
$$\text{N.A.} \sqrt{n^2_{\text{CORE}} - n^2_{\text{CLAD}}}$$

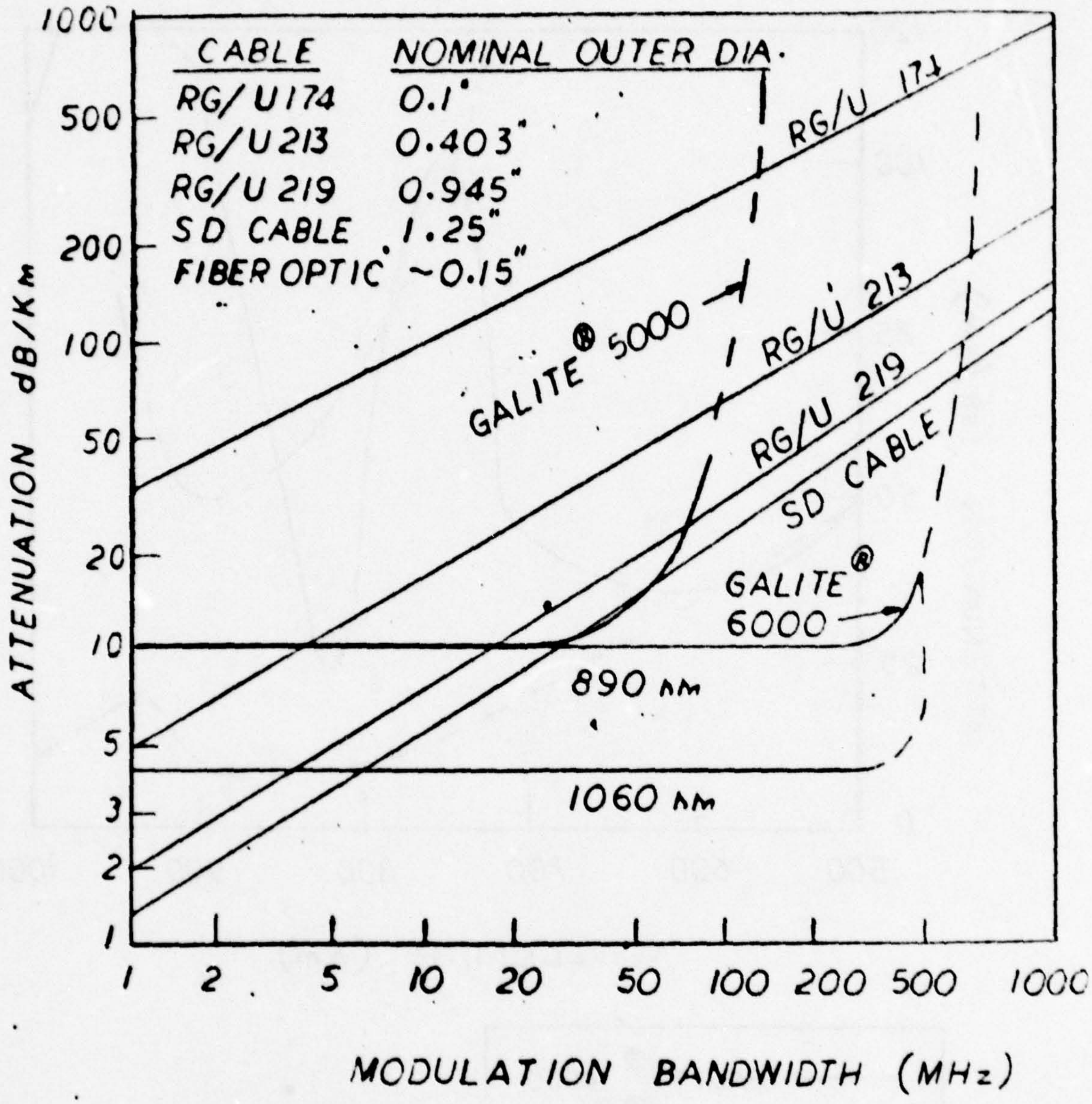
FIBER TYPES

SINGLE-MODE



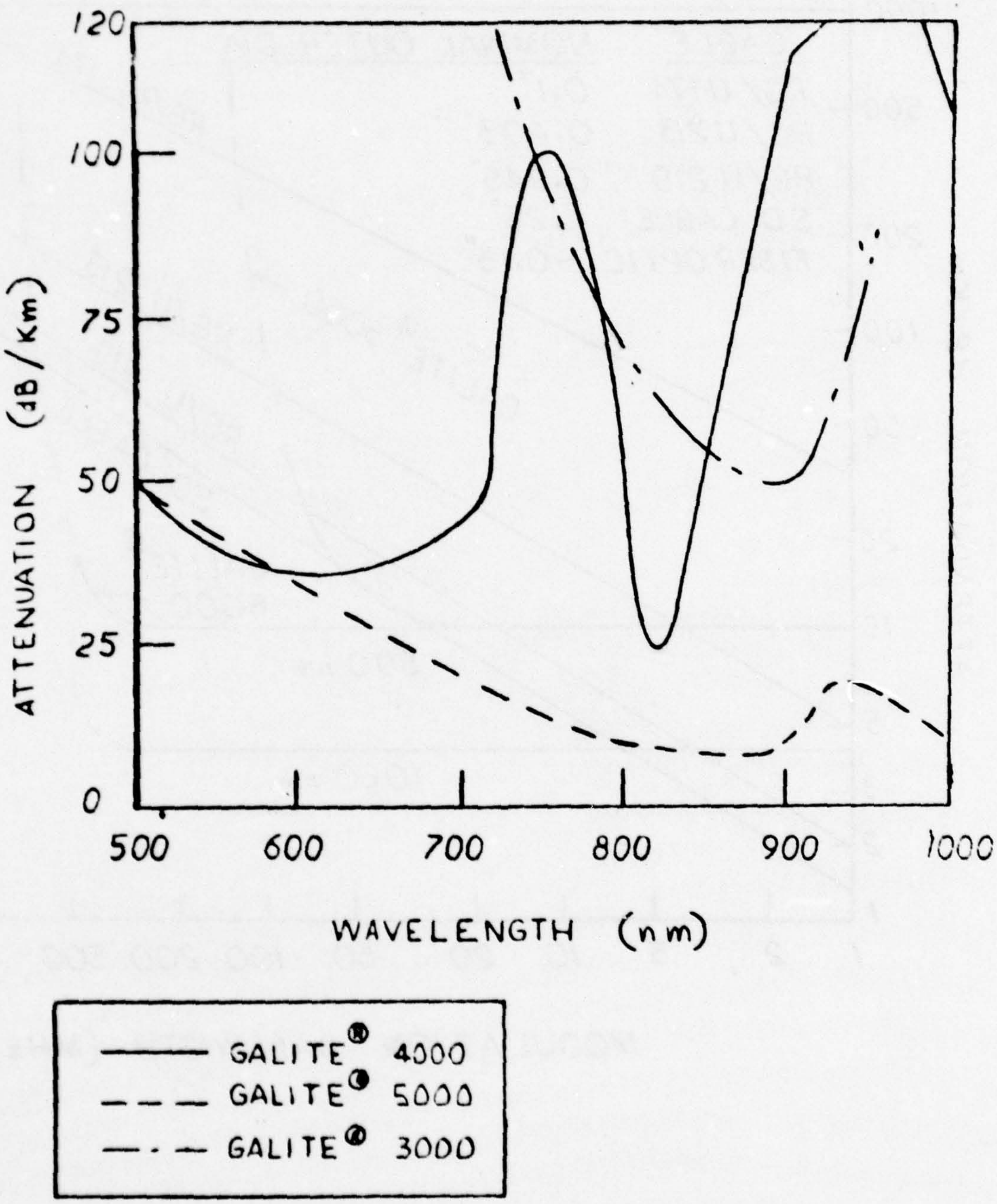
MULTI-MODE





CABLE
NOMINAL OUTER DIA.
RG/U174 0.1"
RG/U213 0.403"
RG/U219 0.945"
SD CABLE 1.25"
FIBER OPTIC ~0.15"
RG/U 174
RG/U 213
RG/U 219
SD CABLE
GALITE® 5000
GALITE® 6000
890 nm
1060 nm
ATTENUATION dB/Km
MODULATION BANDWIDTH (MHz)
1000
500
200
100
50
20
10
5
3
2
1
1
2
5
10
20
50
100
200
500
1000

GALITE® FIBER ATTENUATION



SINGLE-FIBER TECHNOLOGY
(ONE FIBER-ONE OPTICAL CHANNEL)

- GLAMOR TECHNOLOGY
- DIFFICULT TO IMPLEMENT
- STRONGEST FIBERS - BUT EXPENSIVE
- "EASY" TO END-TERMINATE
- LOW LOSS - BUT LOW N.A.
- POOR COUPLING EFFICIENCY (HIGH I.L.s)
- FEW CONNECTOR SUPPLIERS (EXPENSIVE LOSS
- IN-LINE SPLICING TECHNIQUE POPULAR
  - EASY
  - LOW-COST
  - LOW-LOSS
  - REPEATABLE
- MEDIUM TO VERY-HIGH BANDWIDTHS

STEP INDEX	GRADED INDEX
20---40 MHz/km	250---1400 MHz/km

- TYPICAL SINGLE FIBER CHANNEL CHARACTERISTICS
  - POINT-TO-POINT TRANSMISSION
  - LONG HAUL (0.5 km $\leq \ell \leq$ 8 km)
  - PERMANENT LONG-TERM INSTALLATIONS
  - DIGITAL
  - PIG-TAIL ILDs
  - (PiG-TAIL) APDs
- "STANDARD" CABLE FIBER COUNTS: 1, 6, (7)
- EACH FIBER OPTICALLY ISOLATED (60---80dB/km)
- APPROXIMATELY 1.5 nSec/FT SIGNAL DELAY
- LIGHT, MEDIUM, AND HEAVY-DUTY CABLES AVAILABLE
- SEVERAL SUPPLIERS:

  CORNING/SIECOR — GALILEO/REVERE – VALTEC
  ITT/EOPD ——— TIMES FIBER

BUNDLE TECHNOLOGY

(MULTI-FIBER OPTICAL CHANNEL)

- MATURE TECHNOLOGY
- EASIEST TO IMPLEMENT
- CHEAPEST TO IMPLEMENT
- REDUNDANT PATH RELIABILITY
- POOR FIBER-TO-FIBER ISOLATION
- EASY TO END TERMINATE
- MULTITUDE OF CONNECTOR SUPPLIERS
- COMPATABLE WITH LOW COST L.E.D.s
- ADEQUATE BANDWIDTH FOR T.T.L. LOGIC SPEEDS
- CHARACTERISTICALLY HIGH N.A. (0.40...0.66)
- THREE BASIC CLASSES:

DISTANCE (METERS)	FIBER COUNT	ATTENUATION (dB/km)	N.A.
1...30	200	400...500	0.66
30...100	7, 19 (37)	50...100	0.40...0.50
100...1000	7	5...25	0.2 0.4

- LIGHT, MEDIUM, & HEAVY DUTY CABLES AVAILABLE
- HYBRID CABLES AVAILABLE (COAX, STP, FIBER)
- VARIOUS SUPPLIERS
- TYPICAL PRICING (PVC JACKET)

GALITE®	$/FT
1000/200	0.50...0.18
2000/200	0.90...0.33
3000/7	1.20...0.53

- WORST-CASE BW F·O·M OF 20MHz-Rm
- APPROXIMATELY 1.5 nSEC/FT SIGNAL DELAY

— THE EMI-FREE COAXIAL CABLE —

EFFECT OF FIBER BREAKAGE (BUNDLE TECHNOLOGY)

ASSUMPTIONS

1. EACH FIBER HAS SAME N.A.
2. " " " " ATTENUATION
3. " " " " LENGTH
4. " " " " C/C-RATIO
5. " " " " CORE SIZE

BREAKAGE EXCESS LOSS

$$L_{FB} = 10 \log_{10}\left(\frac{N}{N-m}\right) \quad [dB]$$

N = ORIGINAL FIBER COUNT
m = BROKEN FIBER COUNT

ATTENUATION INCLUDING BREAKAGE LOSS

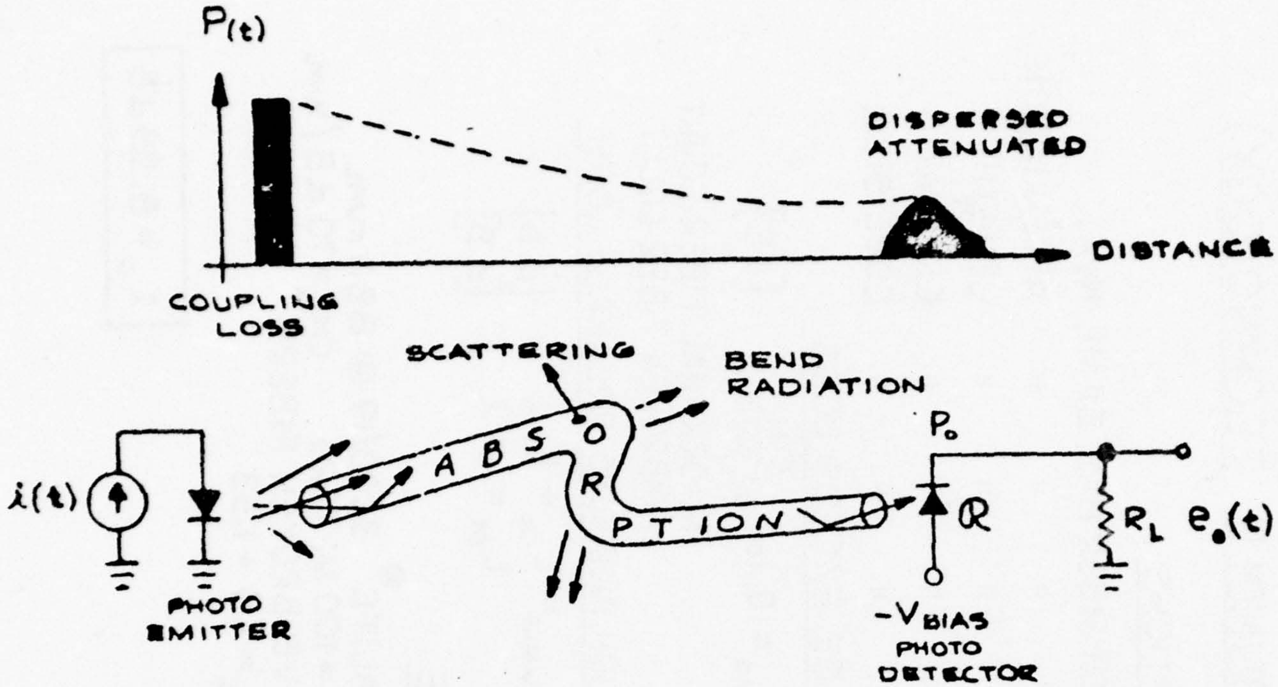
$$L_{CABLE} = L_{\alpha} + L_{FB} \quad [dB]$$

$$L_{\alpha} = \alpha \ell \quad [dB]$$

EXAMPLE

GALITE® 3000/19 @ 880 nm
ℓ = 100 METERS $\quad \alpha_{\lambda}$ = 70 dB/km
FIVE BROKEN FIBERS
$L_c = 7.0 + 1.33$

$L_c = 8.33$ dB



TRANSMISSION LOSS

$$P_{OO} = P_{SA} - L_{CX} - L_{\alpha} - L_{E} - L_{CO} \quad [dB]$$

DETECTOR RESPONSE

$$P_{OE} = [\mathcal{R} P_{OO}]^2 R_L$$

BASICS OF FIBER OPTIC POWER TRANSMISSION

RTI AND OLM

RADIATION TRANSFER INDEX LOSS

$$L_{RTI} = -100 \, Log_{10} (RTI) \qquad [dB]$$
$$= L_{PF} + 2L_F + L_\alpha + L_{NA}$$

TOTAL TRANSMISSION LOSS (TOLM)

$$L_T = L_{CI} + L_\alpha + L_S + L_F + L_{CO} + L_n \qquad [dB]$$

$$\text{WHERE } L_{CI} = L_{AM} + L_{PF} + L_F + L_X + L_{NA} - G_L$$

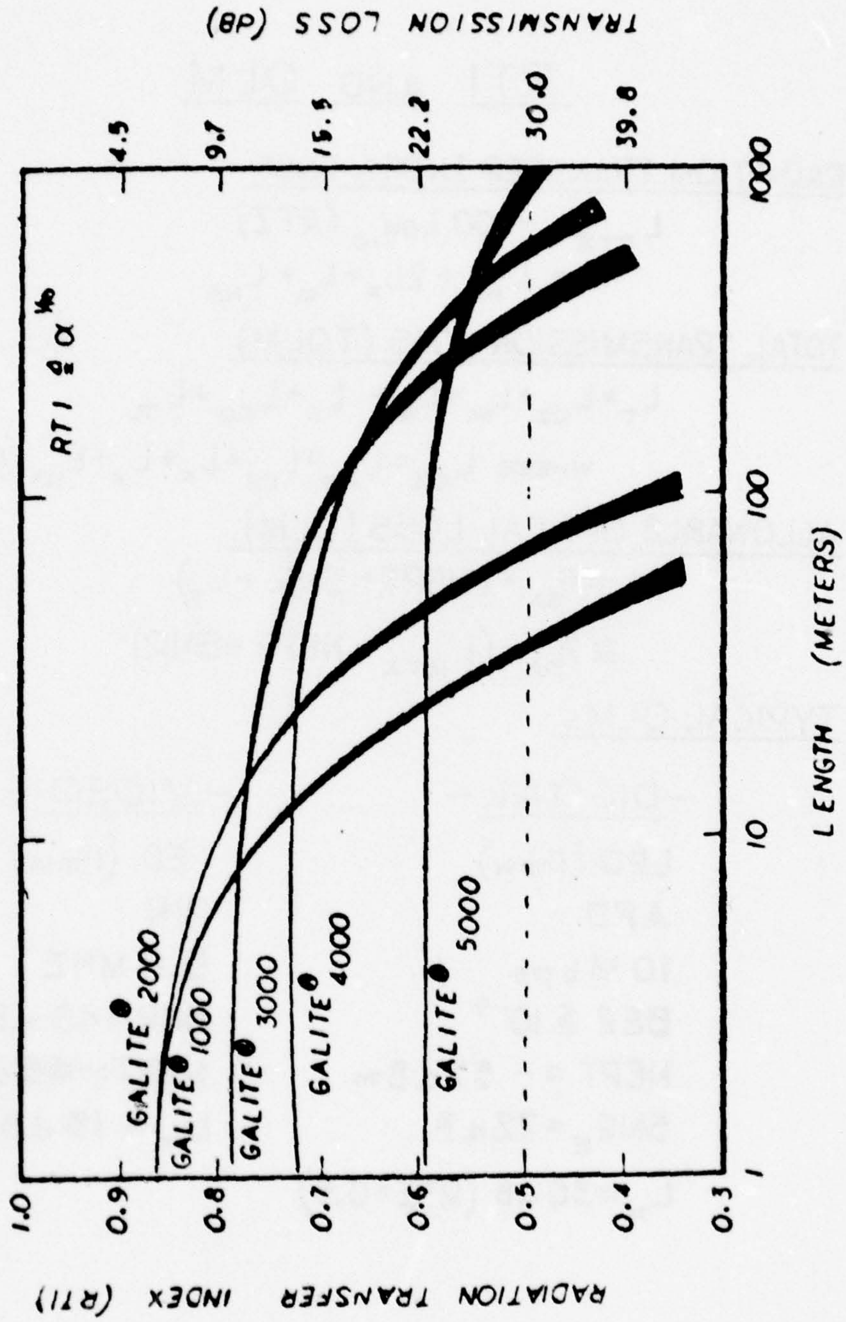
ALLOWABLE OPTICAL LOSS (OLM)

$$L_M = P_{SA} - (NEPT + SNR + L_T) \qquad [dB]$$
$$\cong P_{SA} - (L_{RTI} + NEPT + SNR)$$

TYPICAL OLMs

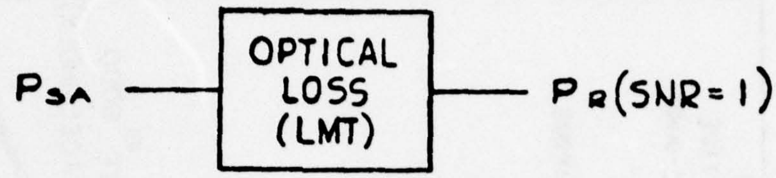
-DIGITAL-	-VIDEO-
LED (1 mw)	LED (1 mw)
APD	PIN
10 Mbps	5.0 MHz
BER $\leq 10^{-9}$	SNR = 45 dB
NEPT = - 55 dBm	NEPT = -45 dBm
SNR_E = 22 dB	L_M = 15 dB
L_M = 30 dB (RTI = 0.5)	

RADIATION TRANSFER INDEX



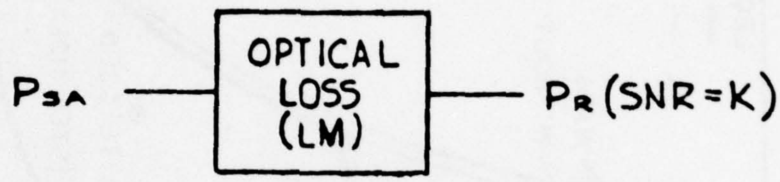
ALLOWABLE OPTICAL LOSS

TOLM



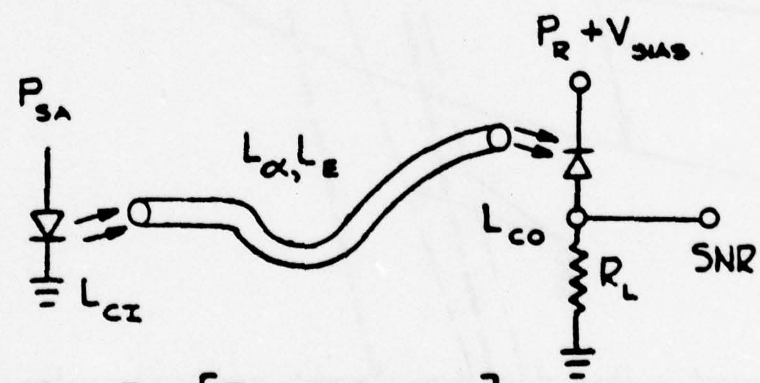
$$L_M = P_{SA} - NEPT \; [dB]$$

OLM

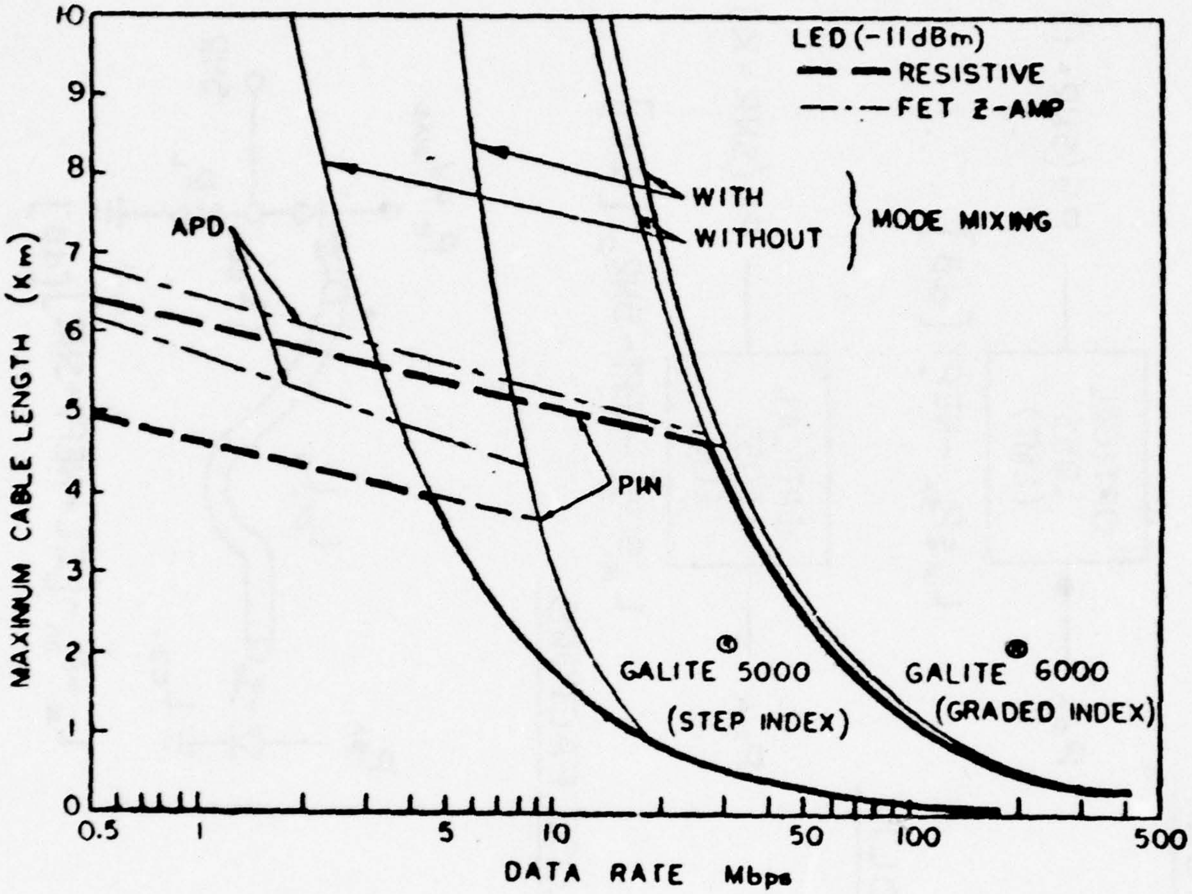


$$L_M \cong P_{SA} - NEPT - SNR_D \; [dB]$$

LOSS FACTORS



$$L_M = P_{SA} - [\Sigma L + NEPT + SNR]_{[dB]}$$



ATTENUATION AND DISPERSION LIMITED OPERATION

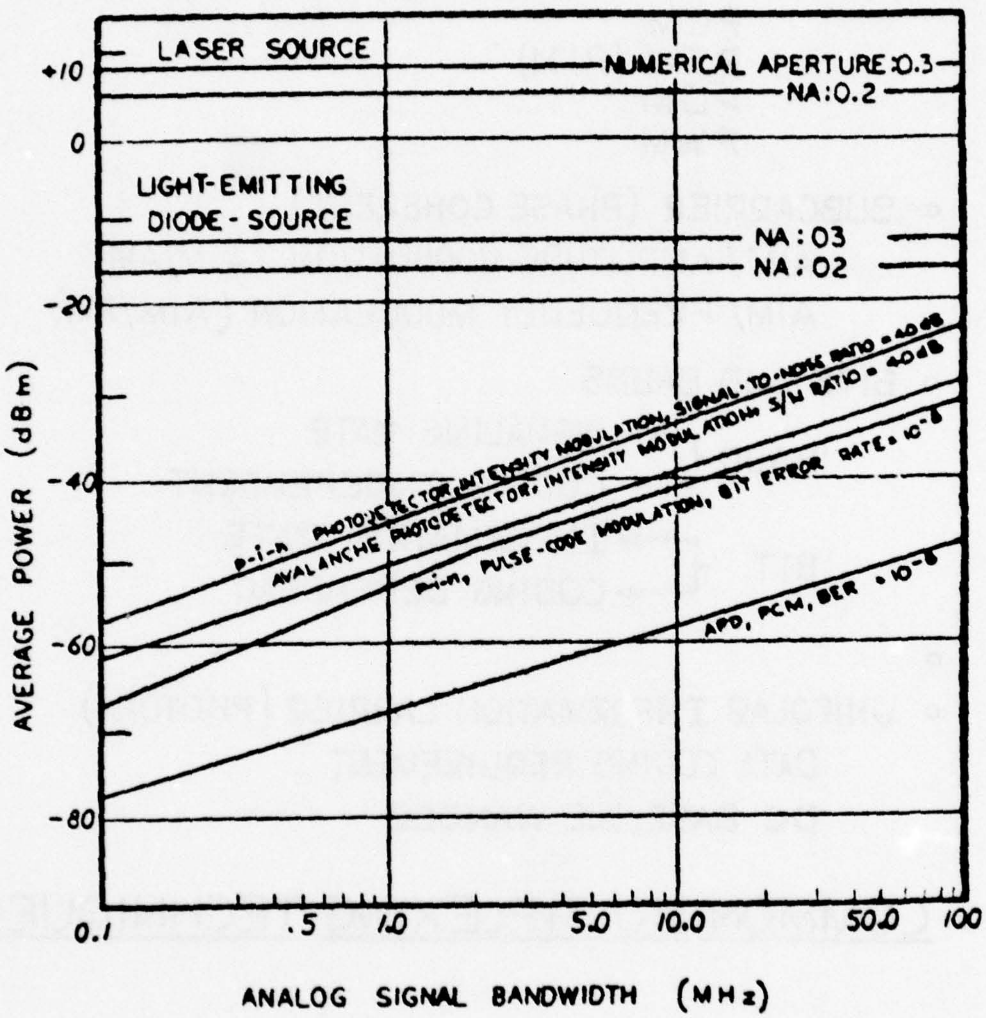
COMMON MODULATION TECHNIQUES

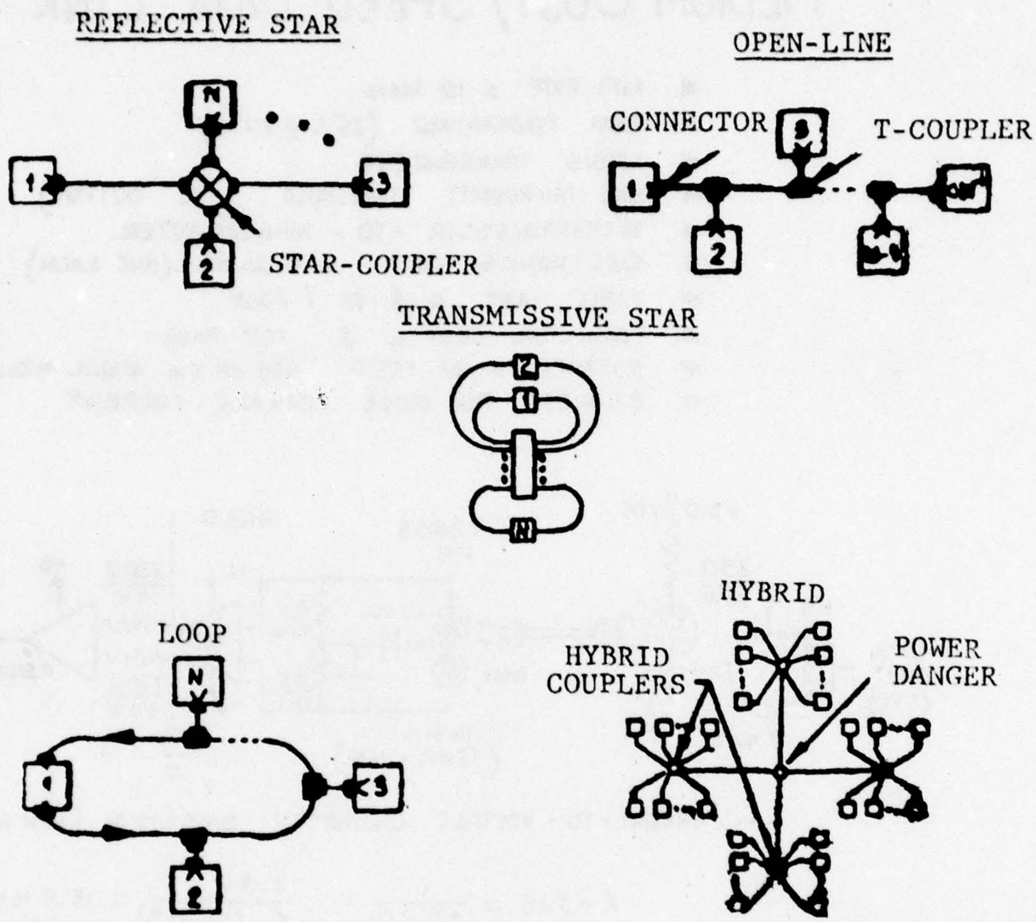
- CARRIER (NON-PHASE COHERENT)
  - ANALOG INTENSITY MODULATION
  - ON-OFF KEYING (OOK)
    - PCM
    - PFM (PIM)
    - PDM
    - PWM
- SUBCARRIER (PHASE COHERENT)
  - AIM/ AMPLITUDE MODULATION (AIM/AM)
  - AIM/ FREQUENCY MODULATION (AIM/FM)
- BITS AND BAUDS
  - BAUD → SIGNALING RATE
    - → CODING INDEPENDENT
  - BIT → INFORMATION RATE
    - → CODING DEPENDENT
- UNIPOLAR INFORMATION CARRIER (PHOTONS)
  - DATA CODING REQUIREMENT
  - DC BASELINE WANDER

COMMON MULTIPLEXING TECHNIQUES

- TIME DIVISION MULTIPLEXING (TDM)
- FREQUENCY DIVISION MULTIPLEXING (FDM)
- WAVELENGTH DIVISION MULTIPLEXING (WDM)
- SPACE DIVISION MULTIPLEXING
- COMBINATIONS OF ABOVE

FIBER OPTIC OPTICAL LOSS MARGINS

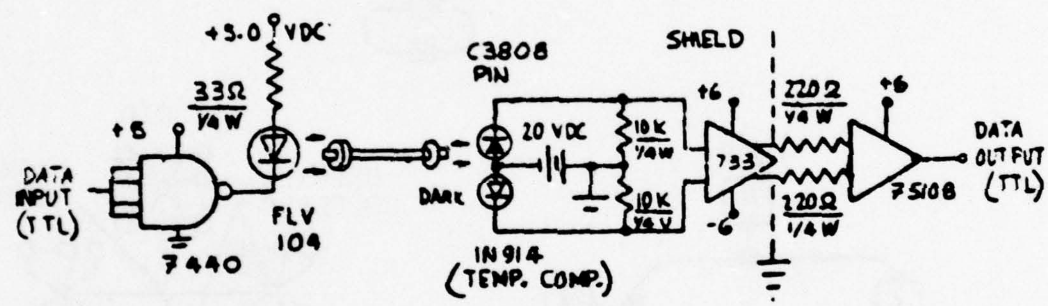




FIBER OPTIC DATA BUS CONFIGURATIONS

MEDIUM COST/SPEED DATA LINK

- DATA RATE $\leq$ 10 MBPS
- ROOM TEMPERATURE (25°C ± 20°C)
- CODING TRANSPARENT
- ONE MICROWATT THRESHOLD (TTL OUTPUT)
- MICROPROCESSOR - TO - MINICOMPUTER
- ELECTRONICS COST $\leq$ \$ 30.00 (ONE EACH)
- CABLE COST $\leq$ \$ 1.00 / FOOT
- CONNECTOR COST $\leq$ \$ PER PAIR
- DISTANCE $\leq$ 100 FEET 600 dB Km @ N.A. = 0.66
- BALANCED PIN DIODE LEAKAGE CURRENT



— CURRENT - TO - VOLTAGE CONVERTER BANDWIDTH (10K Ω)

$$f_{-3dB} = \frac{1}{2\pi RC} = \frac{0.35}{t_{r1}\,(10\text{-}90\%)} = 15.9\ \text{MHZ} \quad @\ C \approx 10\,\text{nF}$$

— PREAMP TOTAL PULSE RISETIME

$$t_r = \sqrt{t_{r1}^2 + t_{r2}^2} \approx 25\,\text{n SEC}\ (15\ \text{MHZ BW})$$

(10-90)

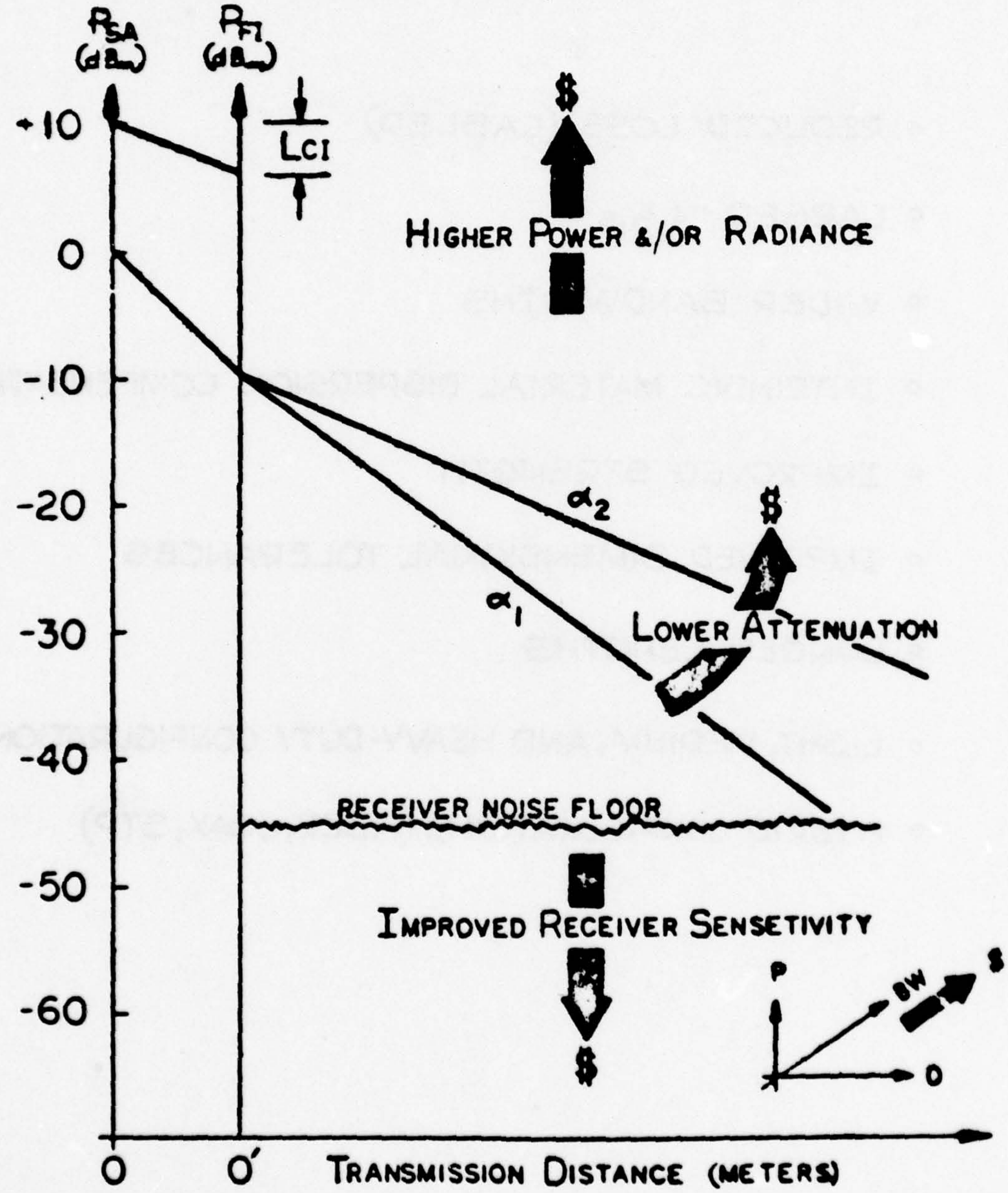
$$\text{WHERE } t_{r2} \approx \frac{0.35}{30\ \text{MHZ}} \approx 12\,\text{nSEC}$$

— 75108 COMPARATOR SUPPRESSES COMMON-MODE DRIFT

— 60 μVrms 733 EINV WITH 10K Ω SOURCE

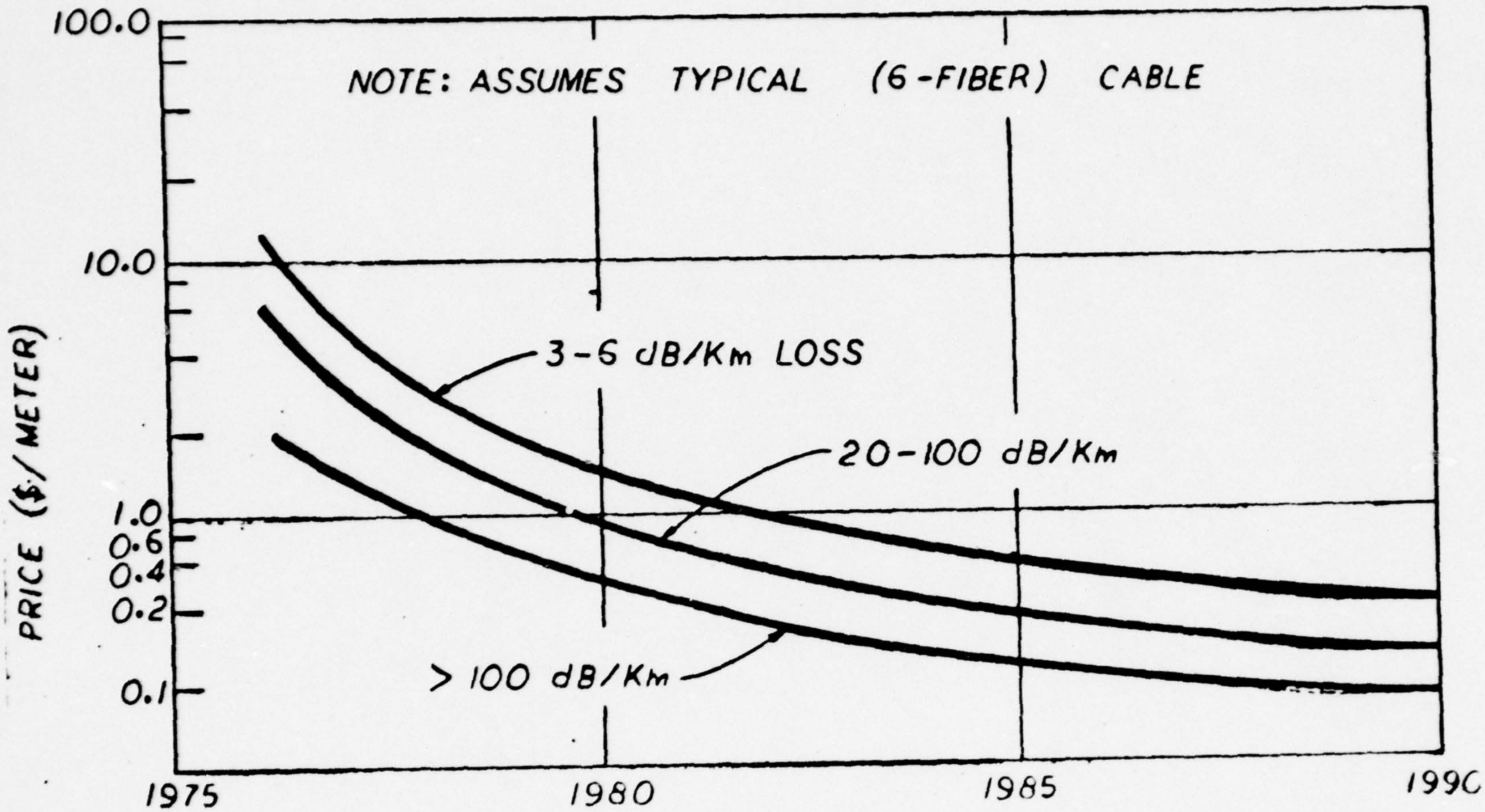
— 100 μVrms AT COMPARATOR (NOISE VOLTAGE)

COST-PERFORMANCE TRADE-OFF SUMMARY



ANTICIPATED FIBER OPTIC CABLE IMPROVEMENTS

- REDUCED LOSS (CABLED)
- LARGER N.A.s
- WIDER BANDWIDTHS
- INTRINSIC MATERIAL DISPERSION COMPENSATION
- IMPROVED STRENGTH
- IMPROVED DIMENSIONAL TOLERANCES
- LONGER LENGTHS
- LIGHT, MEDIUM, AND HEAVY-DUTY CONFIGURATIONS
- HYBRID CONFIGURATIONS (FIBER, COAX, STP)


NOTE: ASSUMES TYPICAL (6-FIBER) CABLE
PRICE ($/METER)
100.0
10.0
1.0
0.6
0.4
0.2
0.1
3-6 dB/Km LOSS
20-100 dB/Km
> 100 dB/Km
1975
1980
1985
1990

INTEL 8273

SDLC PROTOCOL CONTROLLER

JOHN WIPFLI

RON YARA

INTEL CORPORATION
SANTA CLARA, CALIFORNIA

OUTLINE

OVERVIEW

I. SDLC PROTOCOL

II. 8273 LEVEL OF SDLC SUPPORT

III. MICROPROCESSOR TO DEVICE INTERFACE

IV. IMPORTANT FEATURES OF 8273

V. OPERATING CHARACTERISTICS

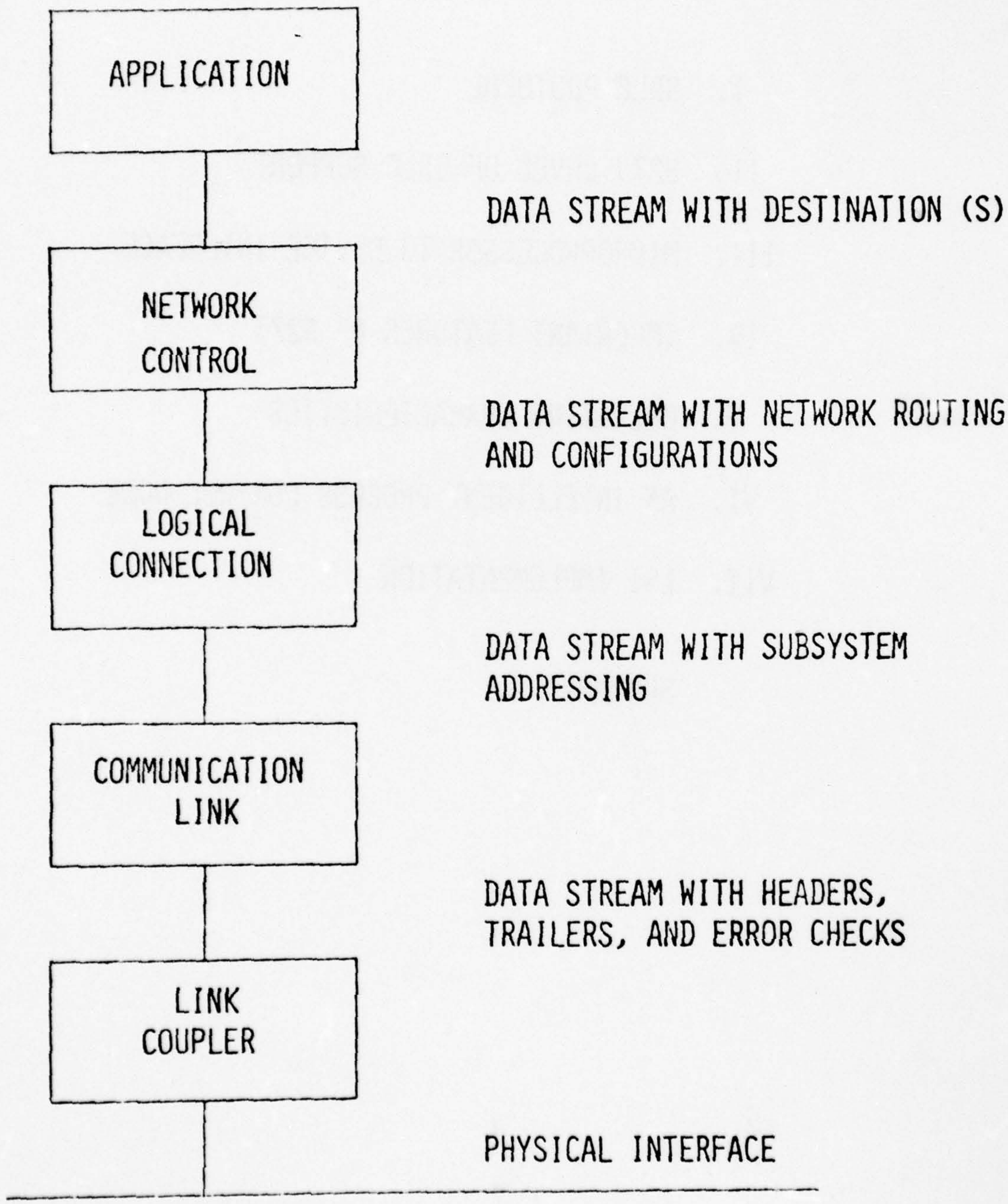
VI. AN INTELLIGENT PROCESS CONTROL NODE

VII. LSI IMPLEMENTATION

SUMMARY

140

LAYERS OF A COMMUNICATIONS SYSTEM



I. SYNCHRONOUS DATA LINK CONTROL (SDLC)

COMMUNICATION DISCIPLINE USED BY IBM TO IMPLEMENT SYSTEM NETWORK ARCHITECTURE (SNA)

IBM, SDLC PROTOCOL CHARACTERISTICS

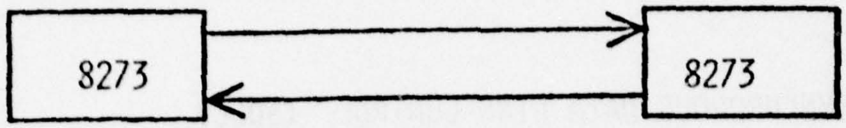
- BIT ORIENTED PROTOCOL
- ZERO BIT INSERTION/DELETION FOR DATA TRANSPARENCY
- CYCLIC REDUNDANCY CHECKING (CRC) FOR ERROR DETECTION
- VARIABLE LENGTH DATA FIELDS
- MULTIPLE MESSAGES MAY BE SENT BEFORE ACKNOWLEDGEMENT IS REQUIRED
- STATION ADDRESSABILITY AND CONTROL FUNCTIONS ARE SPECIFIED
- TIMEOUTS FOR LINK IMPASSE
- MULTIPLE CONFIGURATIONS SUPPORTED:

    DUPLEX

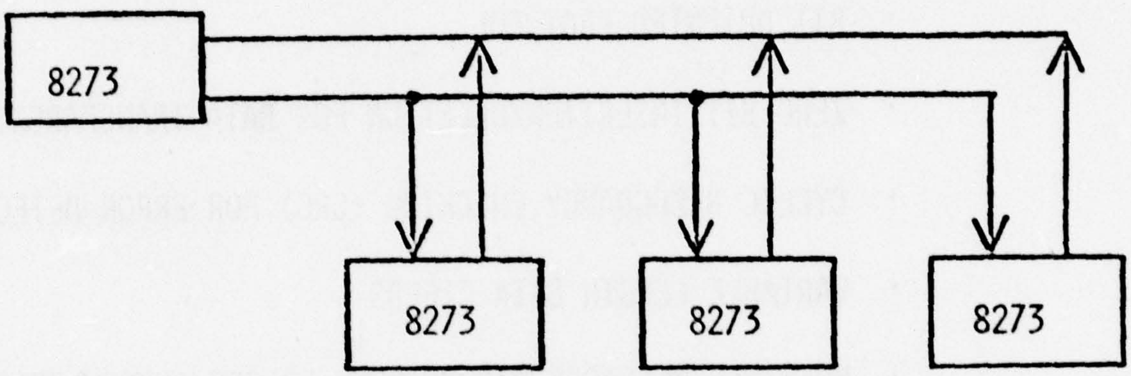
    HALF DUPLEX

    MULTIPOINT

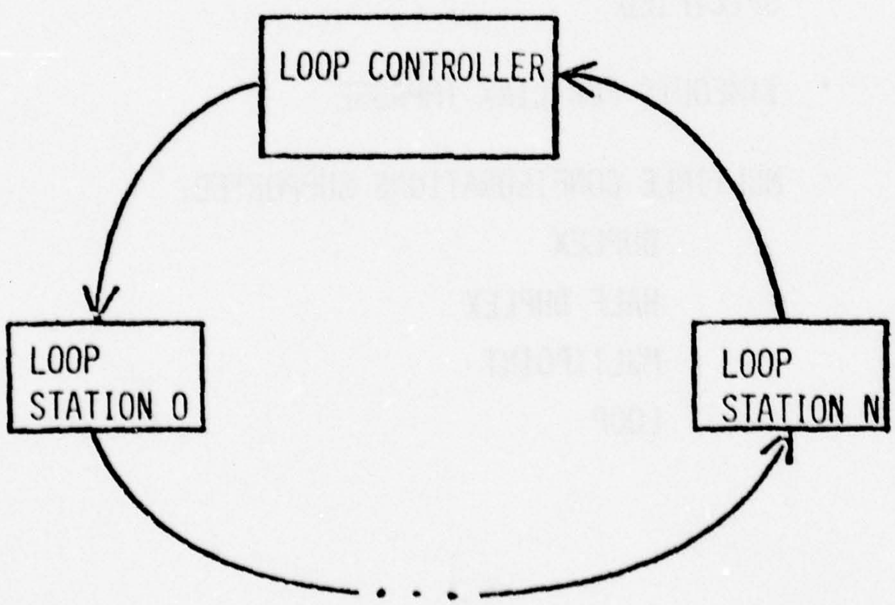
    LOOP



HALF OR FULL DUPLEX



MULTIDROP



SDLC FRAME STRUCTURE

<------- TRANSMIT DIRECTION

FLAG: 01111110 ATTAIN BIT SYNCHRONISM (1)
TERMINATE MESSAGE (2)

A: ADDRESS FIELD (STATION ROUTING) 8BITS

C: CONTROL FIELD (CONTROL/SEQUENCING) 8BITS

DATA: VARIABLE NUMBER OF BITS

FCS: 16 BIT CRC REMAINDER
CRC-CCITT POLYNOMIAL $(X^{16} + X^{12} + X^5 + 1)$
VALIDATES A, C, DATA
FRAME CHECK SEQUENCE

II. 8273 LEVEL OF SDLC SUPPORT

THE 8273 IS A HARDWARE IMPLEMENTATION OF THE SDLC COMMUNICATION LINK LAYER.

LOGICAL CONNECTION AND NETWORK CONTROL LAYERS MUST BE IMPLEMENTED WITH μP SOFTWARE.

DUPLEX, HALF DUPLEX, AND LOOP OPERATION IS SUPPORTED, ALLOWING COMPLETE CONFIGURATION FREEDOM.

III. MICROPROCESSOR (μP) TO DEVICE INTERFACE

CHARACTERISTICS

HIGH LEVEL COMMAND STRUCTURE

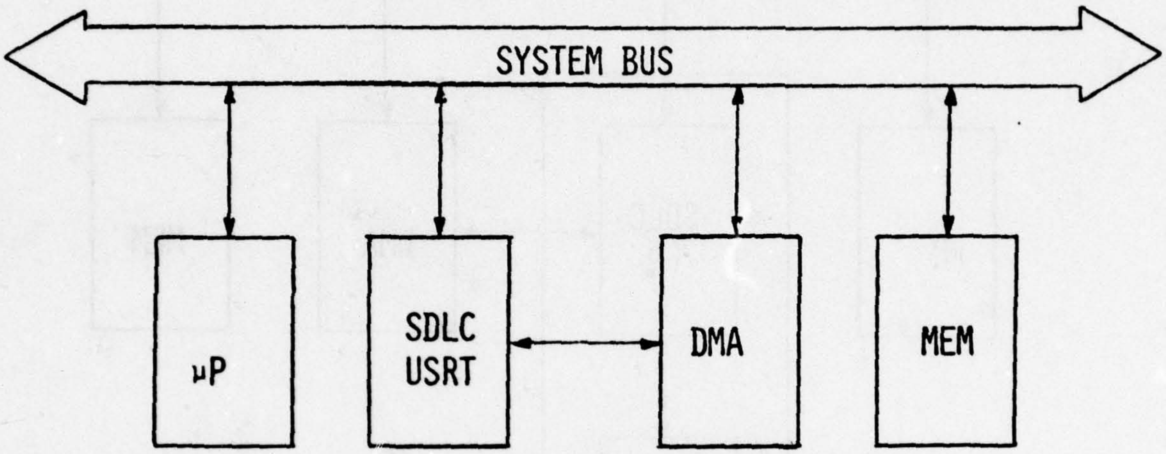
FRAME LEVEL OPERATIONS

DMA - BASED DATA TRANSFER ---→MAXIMUM SPEED (56KBAUD) (raw bit rate)
INTERRUPT - BASED DATA TRANSFER ---→MINIMUM SYSTEMS (9600 BAUD)

MINIMIZATION OF μP INTERRUPTS

PROGRAMMABLE MODES OF OPERATION

μP BASED SDLC WITH USRT



USRT - UNIVERSAL SYNCHRONOUS RECEIVER/TRANSMITTER.

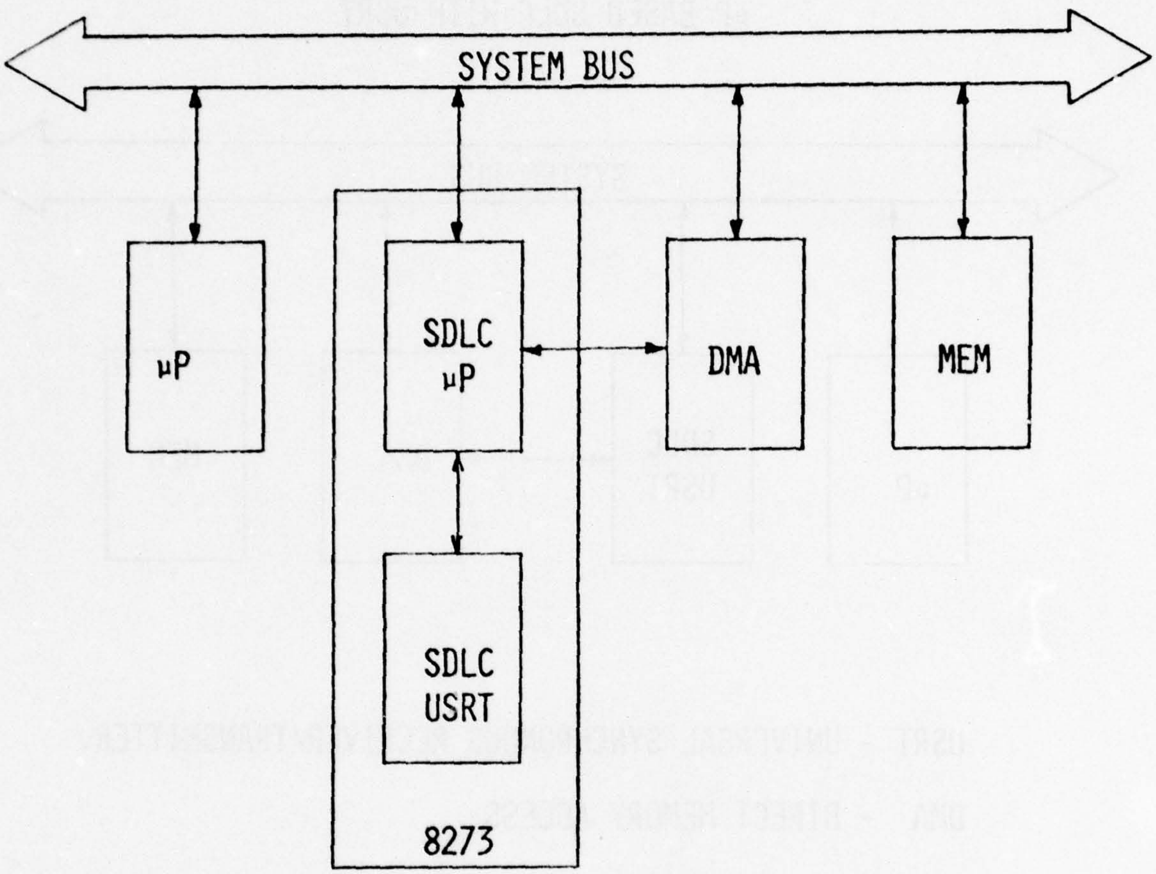
DMA - DIRECT MEMORY ACCESS

MEM - SYSTEM MEMORY

PROBLEM:

SDLC USRT REQUIRES CHARACTER TIME RESPONSE BY μP IN MANAGING SDLC FRAMES.

µP BASED SDLC WITH 8273



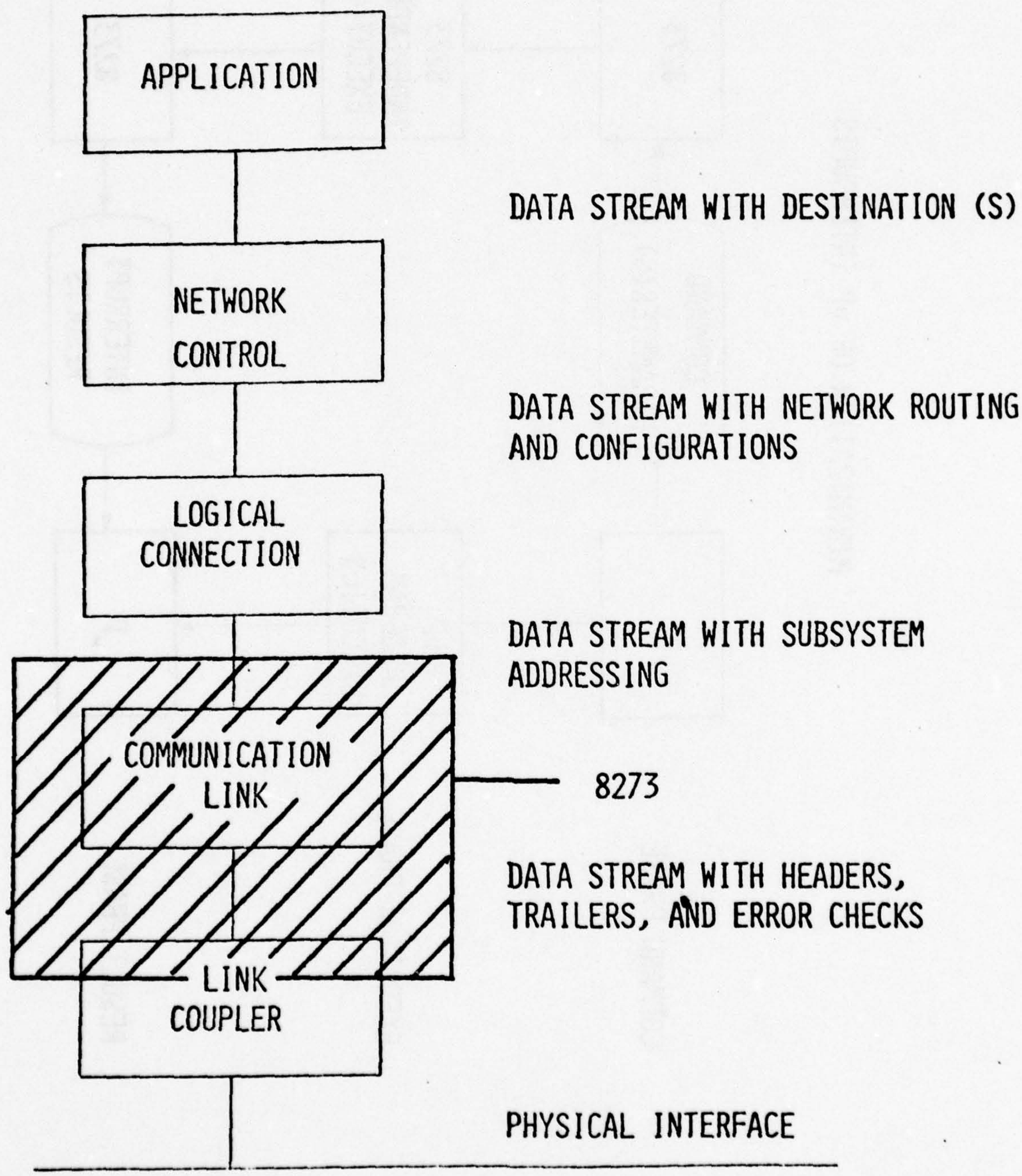
SOLUTION:

INTEGRATING A SPECIAL PURPOSE µP WITH THE USRT, REAL-TIME EVENTS ARE BUFFERED FROM SYSTEM µP.

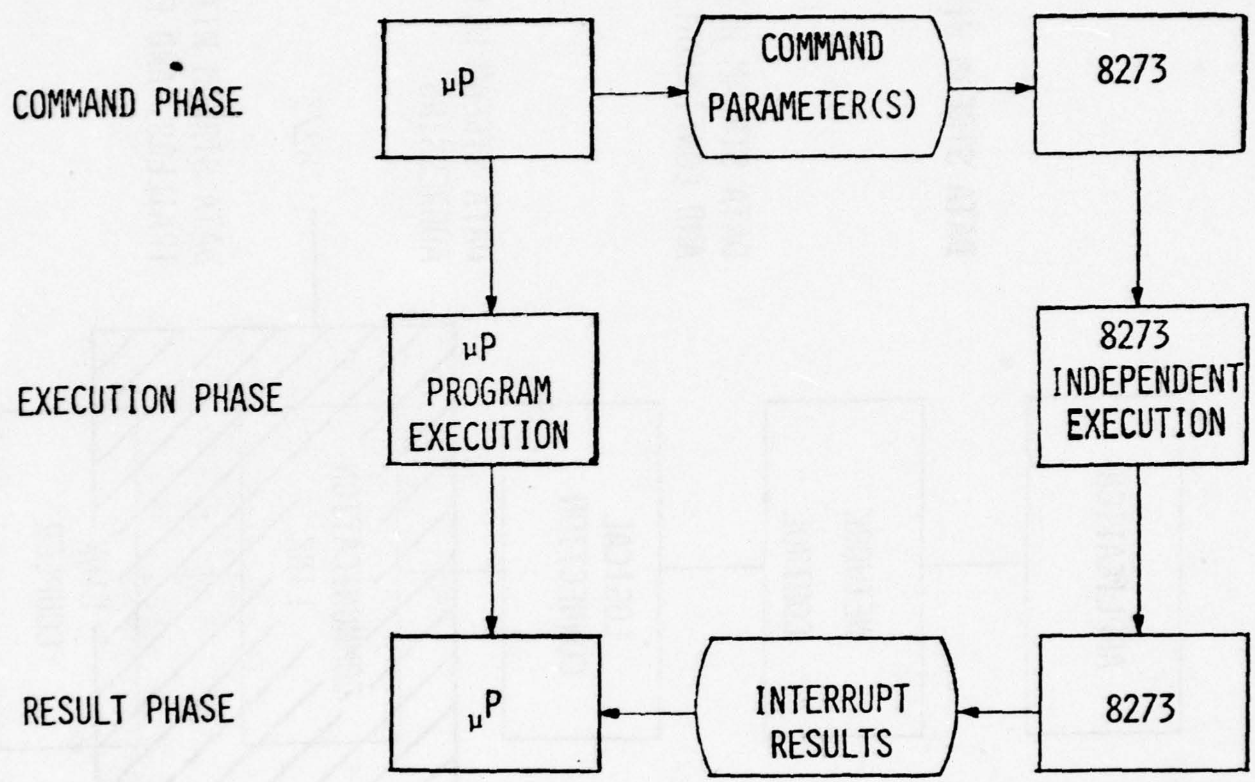
8273 CAN INDEPENDENTLY MANAGE SDLC MESSAGE FORMAT

CPU INVOLVEMENT MINIMIZED

LAYERS OF A COMMUNICATIONS SYSTEM



MINIMIZATION OF μP INTERRUPTS



IV. IMPORTANT FEATURES OF 8273

- DIGITAL PHASE LOCKED LOOP (DPLL)
  CLOCK RECOVERY TO 9600 BAUD
  SINGLE LIGHT PIPE/WIRE PAIR

- LOOP CONFIGURATION SUPPORTED
  ALA - IBM 3650 RETAIL STORE SYSTEM

- NUMEROUS PROGRAMMABLE MODES OF OPERATION
  NRZI ENCODE/DECODE
  DIAGNOSTIC DATA LOOP BACK
  USER DEFINED MODEM CONTROL PORT PINS
  EFFICIENT DMA IMPLEMENTATION

V. OPERATING CHARACTERISTICS

T_A = 0^oC TO 70^oC

V_{CC} = 5.0 V ± 10%

ALL INPUTS AND OUTPUTS TTL COMPATIBLE.

INPUT LEVELS

V_{IL} MAX = 0.8V

V_{IH} MIN = 2.0V

OUTPUT LEVELS

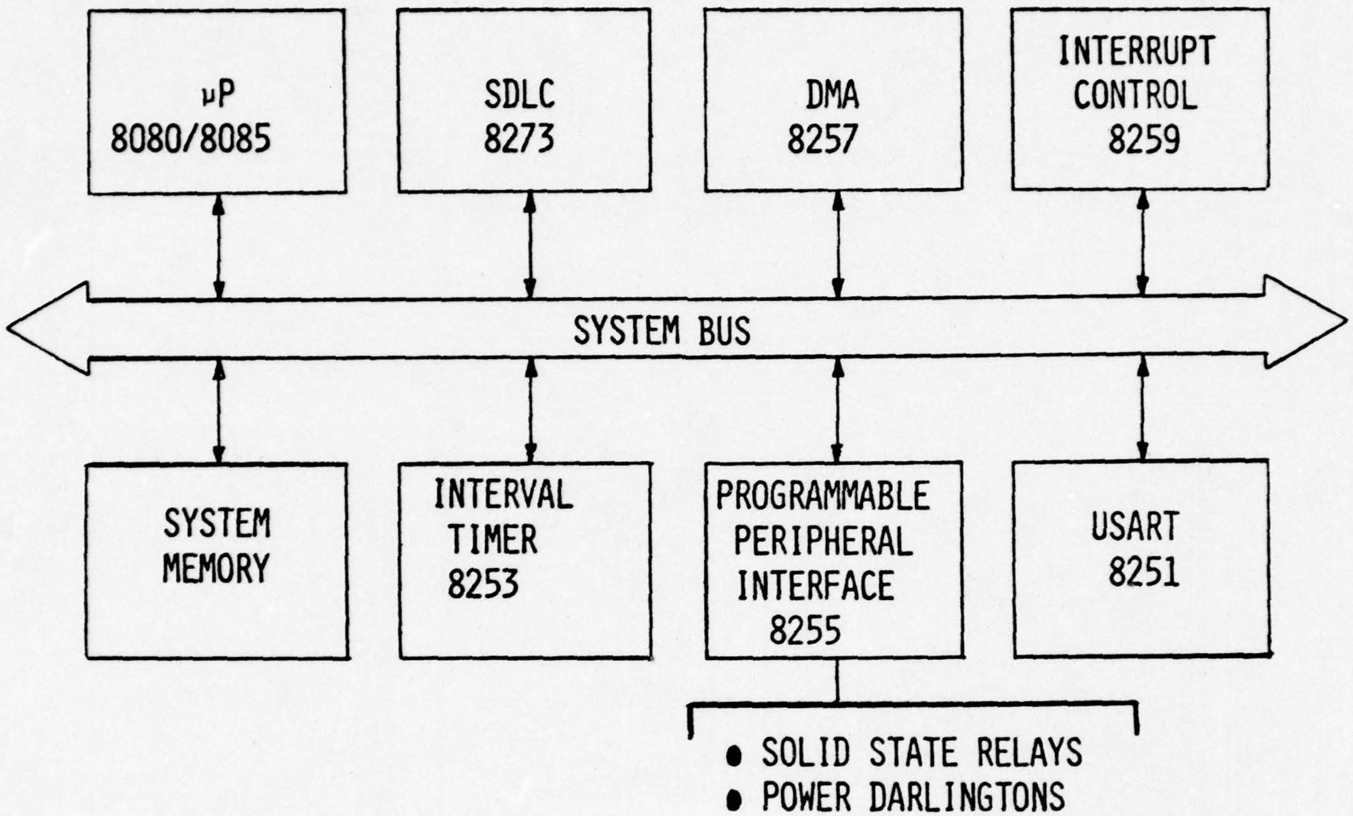
V_{OL} MAX = 0.4V @ 2.0 mA

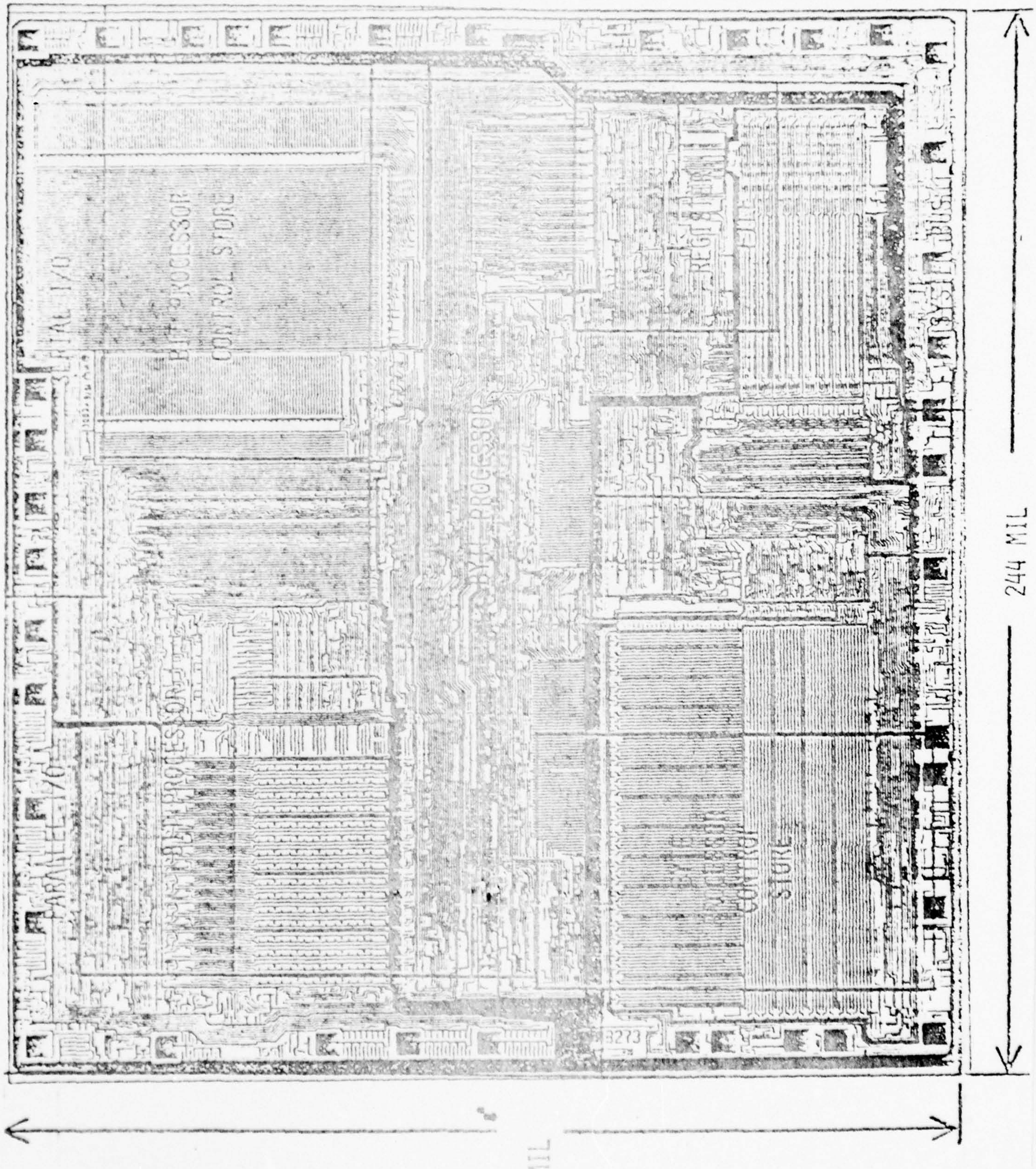
V_{OH} MIN = 2.4V @ -200μA

MOS TECHNOLOGY OFFERS TTL TERMINAL CHARACTERISTICS AT GREATLY IMPROVED FUNCTIONAL DENSITY.

8273 REPLACES ~ 200 TTL COMPONENTS WITH 22,000 DEVICES ON A SINGLE CHIP

VI. AN INTELLIGENT PROCESS CONTROL NODE





CONTROL STORE
BYTE PROCESSOR
CONTROL STORE
8273
244 MIL
218 MIL

SUMMARY

MINIMUM COST INTELLIGENT NODE

- MINIMUM MICROPROCESSOR OVERHEAD
- PROGRAMMABLE NODES/SPECIAL FEATURES
- MULTIPLE CONFIGURATIONS

LSI TECHNOLOGY MAKES THE ENTIRE INTELLIGENT NODE POSSIBLE USING A MINIMUM NUMBER OF PACKAGES:

- 8273 SDLC PROTOCOL CONTROLLER
- 8080/8085 INDUSTRY STANDARD μP
- 8253 INTERVAL TIMER
- 8259 INTERRUPT CONTROLLER
- RAMS & ROMS

SERIAL DATA COMMUNICATIONS IN A MICROCOMPUTER SYSTEM ENVIRONMENT

JOHN WIPFLI

OCTOBER 3, 1977

PRECEDING PAGE BLANK

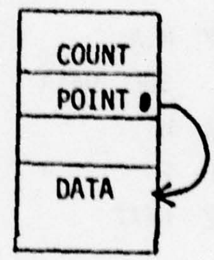
SERIAL DATACOMM IN A MICROCOMPUTER SYSTEM ENVIRONMENT

THE PROBLEM:

MOVE A BUFFER OF DATA FROM ONE MPU SYSTEM TO ANOTHER USING A SERIAL TRANSMISSION DEVICE.

ASSUMPTIONS:

o SIMPLE DATA STRUCTURE



- 2MHZ 8080 SYSTEM
- INTERRUPT CONTROLLER COSTS NO EXECUTION TIME LOSS
- TRANSMITTER IS ONLY OPERATING DEVICE
- NO CHECKS ON TRANSMITTER STATUS
- HALF DUPLEX OPERATION

LET'S FIND THE MPU'S LIMIT ASSUMING A HARDWIRED COMMUNICATIONS CHANNEL

PRECEDING PAGE BLANK

INTERRUPT DEVICE HANDLER FOR TRANSMITTER

```
                                             8080  "T" STATES

DINT:

           PUSH D                              11
           PUSH H                              11
           PUSH PSW                            11      CHANGE CONTEXT
           LHLD COUNT                          16          (69)
           XCHG                                 4
           LHLB POINT                          16

           NOV  M,A.                            7
           IN   SDLC-DATA                      10
           INX  H                               5      TRANSFER BYTE
           DCR  E                               5          (37)
           JNZ  NEXT                           10


                 <SET COMPLETION FLAG>


NEXT:      SHLD  POINTER                       16
           XCHG                                 4
           SHLD  COUNT                         16      RESTORE CONTEXT
           POP   PSW                           10          (66)
           POP   H                             10
           POP   D                             10


           EI                                   4
                                                       RETURN
           RET                                 10       (14)
8080  "T" STATES REQUIRED:  186
                            93  µSEC AT 2 MHZ
```

IMPROVED INTERRUPT HANDLER

```
                                         8080 "T" STATES

DINT:

          PUSH  H                           11
          PUSH  PSW                         11          CHANGE CONTEXT
          LHLD  POINT                       16              (33)


          MOV   A,M                         10
          OUT   SDLC                         7
          INX   H                            5          TRANSFER DATA
          SHLD  POINT                       16              (69)
          LHLD  COUNT                       16
          DCR   L                            5
          JNZ   NEXT                        10


         <SET FLAG>


NEXT:     SHLD  COUNT                       16
          POP   PSW                         10          RESTORE CONTEXT
          POP   H                           10              (36)


          EI                                 4          RETURN
          RET                               10           (14)


8080 "T" STATES REQUIRED:  157
                            78 µSEC AT 2MHZ
```

8257 DMA CONTROLLER

MINIMUM CYCLE = S_I $S_\emptyset$ S_1 S_2 S_3 S_4

= 6 8080 "T" STATES (3 μSEC)

MAXIMUM CYCLE = 6 "T" + INSTRUCTION LATENCY

= 6 "T" + 5 "T"

= 11 "T" (5.5 μSEC)

DEDICATED DEVICE HANDLER

```
                                         "T" STATES
DLOOP:

         IN    SDLC-STAT                   10
         ANI   RXBF                         7          (27)
         JZ    DLOOP                       10

         MOV   A,M                         10
         OUT   SDLC                         7
         INX   H                            5          (37)
         DCR   C                            5
         JNZ   DLOOP                       10
         RET
```

8080 "T" STATES REQUIRED: 64

32 μSEC AT 2MHZ

EVENT TIME TABLE

HALF DUPLEX

BAUD RATE	BIT TIME	BYTE TIME
2.0M	500MS	4 µSEC
1.5M	666MS	5.33 µSEC
1.0M	1 µSEC	8 µSEC
250K	4 µSEC	32 µSEC
100K	10 µSEC	80 µSEC
64K	15.6 µSEC	125 µSEC
56K	17.8 µSEC	142 µSEC
19.2K	52.1 µSEC	416.6 µSEC

FULL DUPLEX EVENTS OCCUR TWICE AS OFTEN, OF COURSE

160

MULTI-PROTOCOL COMMUNICATIONS CONTROLLER (MPCC) 2652

PRELIMINARY SPECIFICATION 2652-1

DESCRIPTION

The 2652 Multi-Protocol Communications Controller (MPCC) is a monolithic n-channel MOS LSI circuit that formats, transmits and receives synchronous serial data while supporting bit-oriented or byte control protocols. The chip is TTL compatible, operates from a single +5V supply, and can interface to a processor with an 8 or 16-bit bidirectional data bus.

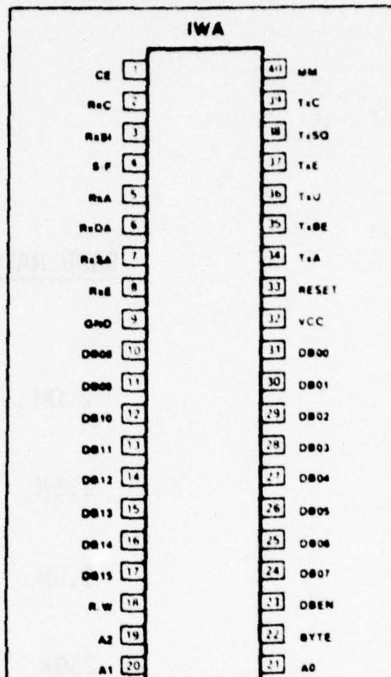
FEATURES

- DC to 500k bps data rate
- Protocol management
  Bit-oriented protocols (BOP): SDLC, ADCCP, HDLC
  Byte-control protocols (BCP): BI-SYNC, DDCMP
- Programmable operation
  8 or 16-bit tri-state data bus
  Protocol selection—BOP or BCP
  Error control—CRC or VRC or no error check
  Character length—1 to 8 bits for BOP or 5 to 8 bits for BCP
  SYNC or secondary station address comparison for BCP-BOP
  Idle transmission of SYNC/FLAG or MARK for BCP-BOP
- Automatic detection and generation of special BOP control sequences, i.e., FLAG, ABORT, GA
- Zero insertion and deletion for BOP
- Short character detection for last BOP data character
- SYNC generation, detection, and stripping for BCP
- Maintenance Mode for self-testing
- Common parameter control registers
- Independent status and data registers for receive and transmit
- Status indicator signals can be used as CPU interrupts
- TTL compatible
- 40-pin package
- Single +5V supply

APPLICATIONS

- Intelligent terminals
- Line controllers
- Network processors
- Front end communications
- Remote data concentrators
- Communication test equipment
- Computer to computer links

PIN CONFIGURATION



NOTE
00 is least significant bit, highest number (that is DB15, A2) is most significant bit

MPCC BLOCK DIAGRAM

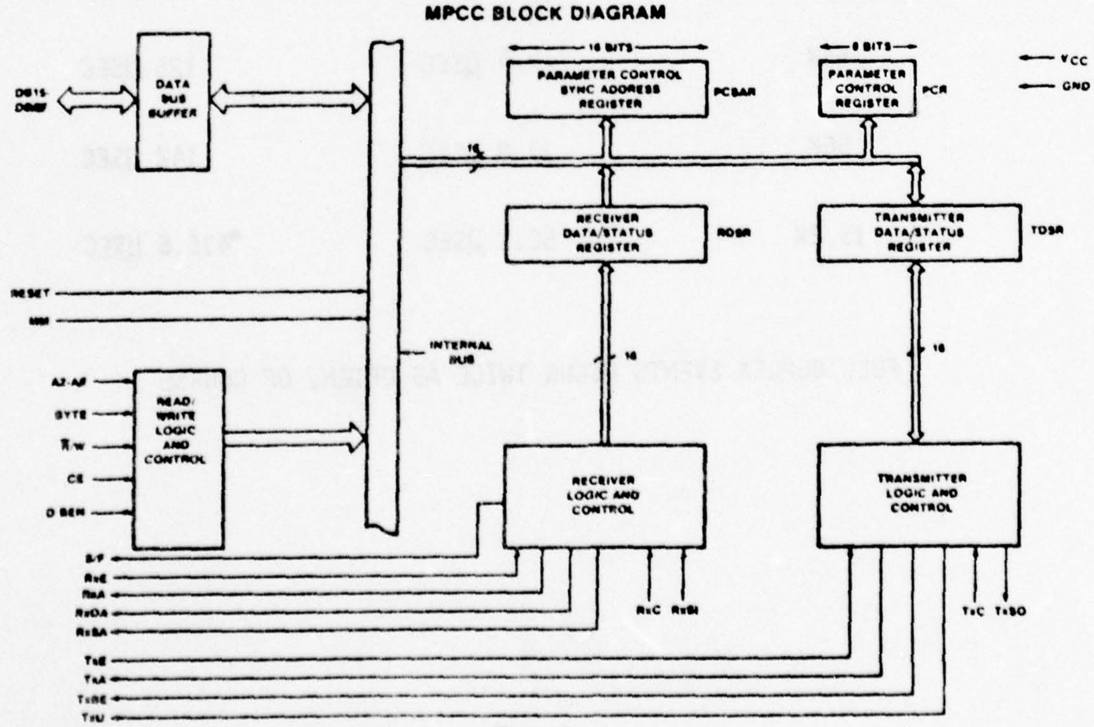


Figure 1

35 Marcus Boulevard
Hauppauge, New York 11787
(516) 273-3100
TWX 510-227-8898

A Standard Microsystems Company

COM 5025

μPC FAMILY

Preliminary Specifications

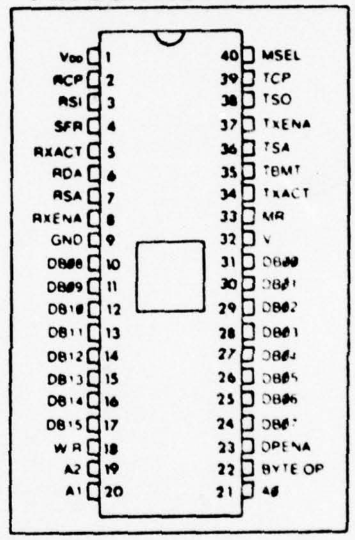
Multi-Protocol Universal Synchronous Receiver/Transmitter

USYNR/T

FEATURES

- ☐ Selectable Protocol—Bit or Byte oriented
- ☐ Direct TTL Compatibility
- ☐ Tri-state Input/Output Bus
- ☐ Processor Compatible—8 or 16 bit
- ☐ High Speed Operation—2.0M Baud—typical
- ☐ Fully Double Buffered—Data, Status, and Control Registers
- ☐ Full or Half Duplex Operation—independent Transmitter and Receiver Clocks
  —individually selectable data length for Receiver and Transmitter
- ☐ Master Reset—resets all Data, Status, and Control Registers
- ☐ Maintenance Select—built-in self checking

PIN CONFIGURATION



BIT ORIENTED PROTOCOLS—SDLC, HDLC, ADCCP

- ☐ Automatic bit stuffing and stripping
- ☐ Automatic frame character detection and generation
- ☐ Valid message protection—a valid received message is protected from overrun
- ☐ Residue Handling—for messages which terminate with a partial data byte, the number of valid data bits is available

SELECTABLE OPTIONS:

- ■ Variable Length Data—1 to 8 bit bytes
- ■ Error Checking—CRC (CRC16, CCITT-0, or CCITT-1)
  —None
- ■ Primary or Secondary Station Address Mode
- ☐ All Parties Address—APA
- ■ Extendable Address Field—to any number of bytes
- ■ Extendable Control Field—to 2 bytes
- ☐ Idle Mode—idle FLAG characters or MARK the line
- ■ Point to Point, Multi-drop, or Loop Configuration

BYTE ORIENTED PROTOCOLS—BiSync, DDCMP

- ☐ Automatic detection and generation of SYNC characters

SELECTABLE OPTIONS:

- ☐ Variable Length Data—1 to 8 bit bytes
- ☐ Variable SYNC character—5, 6, 7, or 8 bits
- ☐ Error Checking—CRC (CRC16, CCITT-0, or CCITT-1)
  —VRC (odd/even parity)
  —None
- ☐ Strip Sync—deletion of leading SYNC characters after synchronization
- ☐ Idle Mode—idle SYNC characters or MARK the line

APPLICATIONS

- ☐ Computer to Modem Interface
- ☐ Modem to Computer Interface
- ☐ Terminal to Modem Interface
- ☐ Modem to Terminal Interface
- ☐ Peripheral to Modem Interface
- ☐ Modem to Peripheral Interface
- ☐ Serial Data Bus

APPENDIX VIII

PROPOSED GUIDELINE FOR IMPLEMENTATION OF INDUSTRIAL PROCESS COMPUTER INTER-SUBSYSTEM COMMUNICATION

IEC/SC65A/WG6
(JAPAN-2)

APRIL 23, 1977

SICS - JEIDA

PRECEDING PAGE BLANK

PROPOSED GUIDELINE
FOR
IMPLEMENTATION
OF
INDUSTRIAL PROCESS COMPUTER INTER-SUBSYSTEM
COMMUNICATION

SICS - JEIDA

COMMITTEEE

FOR

STANDARDIZATION OF INDUSTRIAL COMPUTER SYSTEMS

(JEIDA-JAPAN ELECTRONIC INDUSTRY DEVELOPMENT ASSOCIATION)

SICS-JEIDA

Scope of work

1) Working for ISO/TC97/SC13/WG1

2) Working for IEC/SC65A/WG6

3) Working for International Standardization of Industrial computer system

Membership

Chairman	Prof. M. Terao	Tokyo University
Secretary	Mr. T. Tohyama	Chiyoda Chem. Eng. & Const. Co.
Member	Dr. K. Sato	ETL
	Mr. S. Itaya	Oki Electric Co.
	Mr. T. Ide	Hitachi Ltd.
	Mr. K. Kaneko	DEC Japan
	Mr. A. Ito	Nippon Steel Corp.
	Mr. T. Arima	NEC
	Mr. K. Mori	Hokushin Electric Works
	Mr. H. Nakao	Nippon Mining Co.
	Mr. S. Arai	Fujitsu Ltd.
	Mr. K. Watanabe	Yokogawa Electric Works
	Mr. T. Koda	Tokyo Electric Power
	Mr. S. Watanabe	Mitsubishi Electric Corp
	Mr. T. Matsuda	Yamatake-Honeywell
	Mr. I. Aoki	IBM Japan
	Mr. H. Tanaka	Toshiba Electric Co.
	Mr. T. Murata	MITI
	Mr. S. Nakatani	MITI
	Mr. T. Nakagawa	JEMA
	Mr. S. Sato	JEIDA

IEC/SC65A/WG6
(JAPAN)

Working group for drafting of proposed guideline

in SICS-JEIDA

MEMBERSHIP(WG2)

MR. TAKASHI TOHYAMA	CHIYODA CHEMICAL ENG. & CONST. Co.
MR. AKIHIRO UYETANI	TOSHIBA ELECTRIC Co.
MR. HIROSHI TANAKA	" " "
MR. YOSHIAKI BANDAI	" " "
MR. HISASHI TAMURA	YOKOGAWA ELECTRIC WORKS
MR. KAZUTAKA WATANABE	" " "
MR. IWAO MATSUMOTO	YAMATAKE HONEYWELL Co.
MR. AKIRA FURUSAWA	JEIDA
MR. TSUNEO SAITO	"

CONTENTS

1. INTRODUCTION

The intention of this document is to describe a standard inter-subsystem communication method in order to increase the degree of compatibility between subsystems from various manufacturers, thus enabling them to be used together in more complete computer based process measurement and control systems.

1.1 OBJECTS OF PROPOSED GUIDELINE

(1) This proposed guideline applies to information interchange systems used to interconnect both process control and process computer apparatus with other apparatus and accessories necessary to assemble instrumentation and control system including computer systems.

(2) This proposed guideline describes the design concepts and the specifications for implementation of industrial process computer inter-subsystem communication (hereafter it is called Industrial dataway system).

(3) A primary focus of this document is to set forth a conceptual design guideline of industrial dataway system. (This document deals only with reference for standardizing promotion of Industrial dataway system).

1.2 SCOPE OF GUIDELINE (STANDARDIZATION)

(1) The goal of this guideline (standardization is to recommend main features of digital data communication using bit serial techniques over single line sharing system.

(2) This data communication system (Industrial dataway system) is capable of supporting centralized intelligence, distributed intelligence, hierarchical intelligence and combinations

thereof. In particular, it shall be capable of supporting distributed systems for process measurement and control.

(3) This guideline deals only with the communication interface characteristics of Industrial dataway system to the exclusion of network control and application functions, and design specification of electrical signal in communication line.

(4) This guideline does not take aim at the optimal interface for high speed standard computer with high speed peripherals, such as mass memories, line printers and so on. Also, it is not intended for efficient sharing of mass storage and peripherals between processors.

(5) This guideline is intended;

- to define a general-purpose system for use in long distance communication application (max 50 km straight line).
- to specify the device independent functional, electrical and mechanical interface requirements which devices and/or stations shall meet in order to be interconnected and communicate unambiguously via the system.
- to enable the interconnection of independently manufactured devices into the system.

1.3 MATTERS RELATED TO THE FUNCTIONAL REQUIREMENTS

(1) This guideline is based on the "Draft of Functional Requirements for Industrial Process Computer Intersubsystem Communication" IEC SC 65A/WG6 (Secretary) 9 which is prepared at 1977-1-18 and ISA SP72 (Secretary) 5 which is prepared at 1977-1-14.

(2) The deviations between this guideline and the Functional Requirements (IEC SC 65A/WG 6 and ISA SP72) are shown in Appendix 5.

2. APPLICATION ENVIRONMENTS

2.1 REVIEW OF DATA COMMUNICATION

(1) The line sharing systems for industrial plants are developed and utilized in various types of systems shown in Figure-1 and Table-1.

(2) The existing line sharing system was designed as the above requirements, so there were many kinds of line sharing system that provided for the different grade of each requirements.

(a) Volume for data transmission, and required throughput.

(b) Quick response in data transmission, and required priority interrupt handling.

(c) Distance for data transmission.

(d) Connected devices in data transmission network.

(e) Intelligence for data communication support at connected device, device interface, or communication interface.

(f) Reliability and Maintenability.

(g) Economics, reducing system costs.

(3) The industrial process computer inter-subsystem communications systems of IEC SC 65A/WG6 will be applied for inter-computer or intelligent terminal communication for distributed computer control systems including process data gathering, control or supervisory logic processing, man-machine communication supporting and other related intelligence.

170

For this purpose, the Industrial dataway system shown in Figure-1 seems to be suitable for data communications in process data acquisition and control.

2.2 APPLICABLE FIELD OF INDUSTRIAL DATAWAY SYSTEM

2.2.1 OUTLINE

(1) The Industrial dataway system for on-line real time computer control system or distributed digital control system to be used primarily, but not exclusively, in the process industries (for their continuous and discrete processes).

(2) The following conditions should be included.

- to provide reliable and economic data communication
- to be capable of being incorporated with in centralized and distributed network control system
- to be capable of keeping ultimate correct data communication under industrial process environments.

(3) It is required to simplify the system assemble, maintenance and training of engineer and staff.
Specifically it is wanted to avoid getting involved with a proliferation of different types of equipments.

2.2.2 TYPICAL INDUSTRIES

(1) This guideline applied generally to process data acquisition and control.
This Industrial dataway system is intended to apply a closed loop control system of industries.

(2) Typical application areas are:

- Distributed digital control system
- Process computer control system

For example:

"Industries which manufacture agricultural and industrial chemicals; biologicals and pharmaceuticals; elastomers; foods, paints, varnishes and pigments; petrochemicals and petroleum products; pulp and paper; soaps, glycerin, and detergents; glass; cement; synthetics, such as films and fibers; and other associated and related materials; and refine and produce metals and generate electric power. Such processing may involve a change in composition, shape, or state of matter which may be affected by pressure, temperature, chemical reaction, mixing, separation."

2.2.3 TYPICAL CONFIGURATION

(1) The overall configuration of Industrial dataway system is to provide an effective communication link over which messages and commands are carried in an unambiguous way among a group of interconnected devices.

(2) The data transmission has the capability to transport data from one station to another. In data communication link, the master station due to control of a link during a given transmission.

The slave station dues to receive only during a given transmission.

(3) Master/Slave status may be changeable. But at any given time, master station has responsibility for link control and error recovery during transmission.

(4) The data communications are carried among subsystems of the following types:

Table-1 Classification of Data Transmission System

NO.	CATEGORY	NETWORK STRUCTURE	BASIC DATA TRANSMISSION	BASIC DATA FORMAT	PROTOCOL	MAIN APPLICATION
1	Computer Bus	1:1 Correspond or 1:n Correspond		Fixed Length Data	Not fixed	Computer itself
2	Data Transmission for Instrumentation	1:n Correspond or n:m Correspond	Shake Hand Transmission	Fixed Length Data	Standardized	CAMAC, IEC-Bus
3	PI/O Communication	1:1 Correspond	Polling	Variable Length Data	Simplified	Remote data gathering
4	Industrial Dataway System	n:n Correspond	Shake Hand Transmission	Variable Length Data	Not fixed	Inter-computer or inter-subsystem communication
5	Wire Sharing System	1:1 Correspond or 1:n Correspond	Cyclic Transmission	Fixed Bits Pattern	Cyclic Serial Transfer	Remote signal gathering
6	Computer Network	1:n Correspond or n:m Correspond	Polling	Variable Length Data	Not fixed or HDLC	Wide spread computer and terminal information gathering
7	Telemetry and Telecontrol or Tele-communication	1:n Correspond or 2:n Correspond	Cyclic Transmission or Polling	Fixed Length Data or Byte Oriented Data	Standardized	Wide spread remote signal data transmission

Figure-1 Status of Data Transmission System

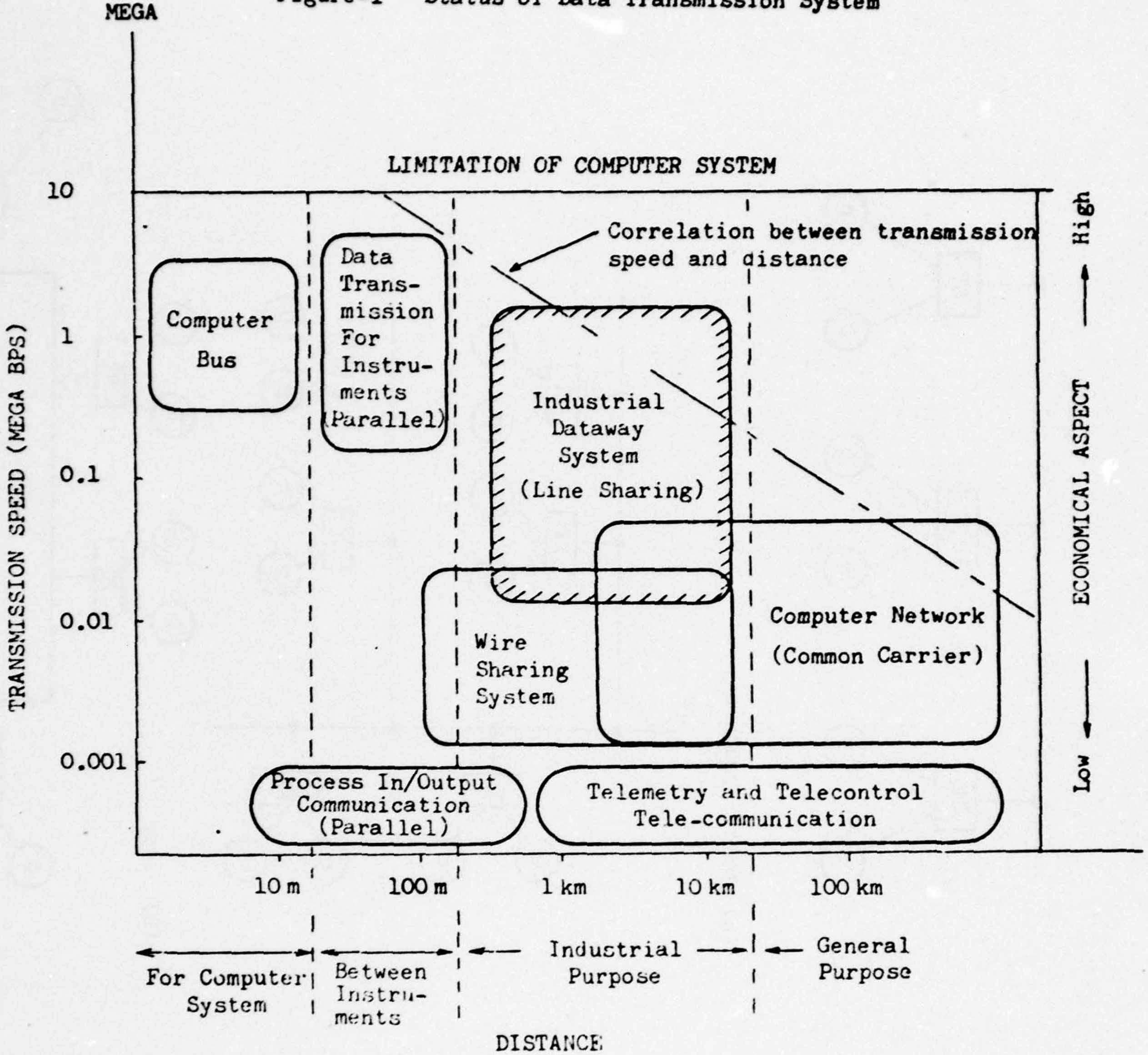
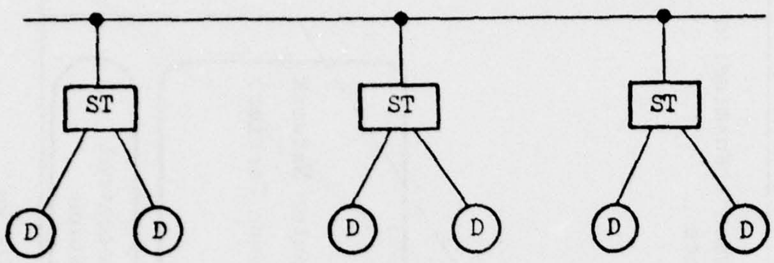


Figure-2 Typical Configuration

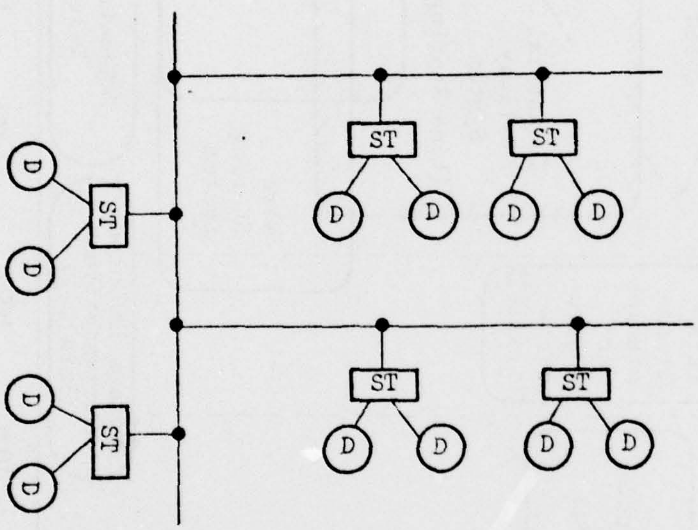
ST : STATION

D : DEVICE

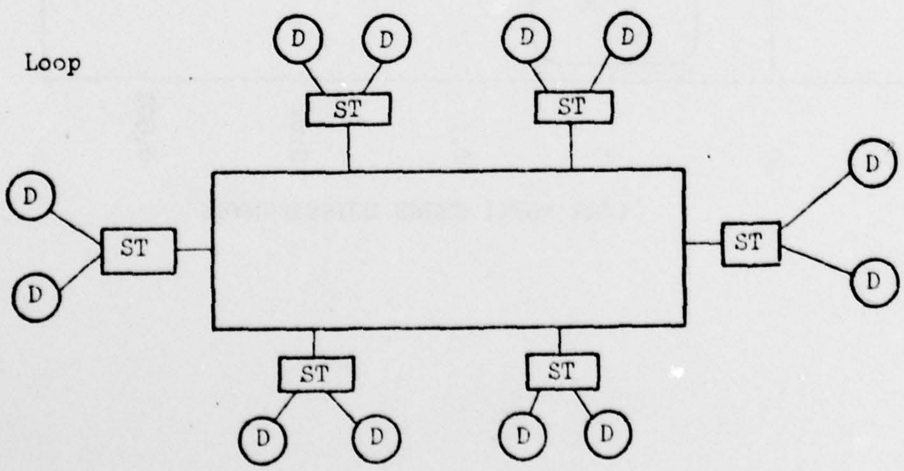
(a) Single branch



(b) Multi branches



(c) Loop



(a) Process signal input, output, and control subsystems

(b) Man/Machine Interfaces and Product Identification Device Subsystems

(c) Supervisory computer systems

(d) Service, support and maintenance

(e) Combinations of any or all of above.

(5) The translator equipment will be available which make it possible to implement portions of the communication subsystem with common carrier channels.

(6) Typical configurations are shown in Figure-2.

3. FUNCTIONAL SPECIFICATION

3.1 TOPOLOGY OF INDUSTRIAL DATAWAY SYSTEM

(1) The Industrial dataway system is required to support distributed system for process data acquisition and control, and field expandability.

The topology of Industrial dataway system is recommended as branch type.

(2) In this industrial dataway system, single line shared communication is applied. In single line, data, control frame and synchronization signal should be included.

(3) In more reliable system, loop type or redundancy like dual or duplex system shall be applied.

(4) This system is not intended to handle message switching.

Note: Typical examples are shown in Appendix-1 and 2 respectively.

3.2 MASTERSHIP IN INDUSTRIAL DATAWAY SYSTEM

(1) The master station which currently controls the link is given to any station as multi mastership.

(2) Address and its related modifier shall be specified to be applicable for n:n corresponded communication

(3) The available architectures of the communications should not preclude direct data interchange between any two stations: It should be possible to transmit data direct between any two subsystems on the link without necessarily involving store and forward at a third subsystem.

3.3 TRANSMISSION SPEED

(1) The industrial dataway system will be optimized for a raw bit rate of 1 MBPS or more.

3.4 TRANSMISSION DISTANCE

The industrial dataway system should be capable of accommodating two class of distance, to achieve broad applicability.

(1) Remote sites accessible over leased lines with repeaters: Maximum 50 km (straight line)

(2) Between any two stations: Maximum 4 km (without repeaters)

3.5 NUMBERS OF STATIONS PER DATAWAY

(1) The Industrial dataway system shall be capable of having stations per one dataway:

(a) Minimum 31 stations

(b) Up to 255 stations

3.6 NUMBERS OF DEVICES PER STATION

(1) The Industrial dataway system shall be capable of having following devices per one station:

(a) Process input/output subsystem : max. 512 pts.

(in case of analog input)

(b) Process controller : max. 128 pts.

(in case of 3 mode controller)

(c) Terminal device : max. 16 pts.

(in case of device subsystem)

3.7 TYPE OF DEVICE

Devices to be connected with station are shown in Appendix-3. [See 2.2.3. (4)]

3.8 MODE OF TRANSMISSION

Bit serial technique is applied to this system. And also half-duplex transmission is good for this application, because of high transmission speed.

3.9 PRIORITY CONTROL OF TRANSMISSION

(1) Asynchronous signals or events shall be served within a definite interval that is short time interval to ignore an effect on delay of asynchronous events.

(2) The communication system shall be served without delay when a local devices or terminals have errors or troubles.

(3) Priority control down by hardware may not be necessary in case of satisfaction of the above mentioned requirements.

3.10 RESPONSE TIME

Detection of asynchronous signals or events shall be served less than 20 msec.

3.11 MODULATION AND SYNCHRONIZATION

Modulation technique for the industrial dataway system shall be enough to apply base-band.

Synchronization should be based on HDLC's definition.

3.12 TYPE OF COUPLING

(1) The Industrial dataway system shall be easy for installation and maintenance, so that coupling of signal line of dataway should be standardized as follows:

1) Coaxial cable : T-connector

2) Twisted-pair cable : Terminal strip

3.13 EXPANDABILITY

(1) The Industrial dataway system shall provide flexibility for the user to economically change or expand the system after installation within practicable address areas and distance limitations.

(2) The Industrial dataway system can be extended or stations or devices added. Such changes may disturb the exchange of messages as a transient effect, provided that the system is able to detect such disturbances and recover full operation within a time appropriate to the application.

3.14 ENVIRONMENTAL CONDITION

The Industrial dataway system is used for applications of process industries.

Environmental condition of this system is recommendable as follows:

1) Operating temperature range : 0 to 50^{o}C

(Do not include terminal device)

2) Operating humidity : below 95% (without dew)

3) Earth potential : less than 10 ohms*

4) Withstand boltage : AC 1,500V 1 min.

(for power line)

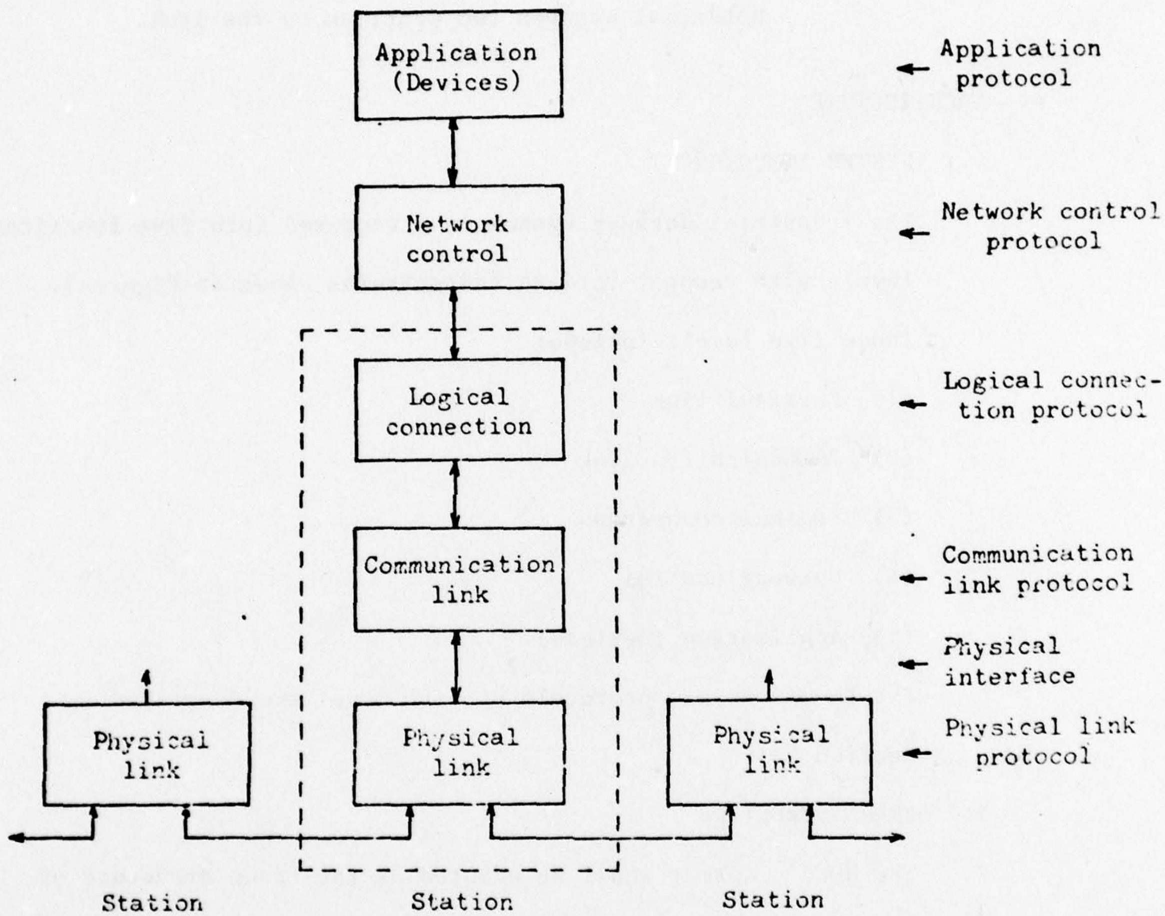


Figure-3 System structure of the Industrial dataway system

DC 500V 1 min.

(for signal line)

Note: *) The system must be able to function normally with extreme static and dynamic differences of ground potential between two stations on the link.

4. ARCHITECTURE

4.1 SYSTEM STRUCTURE

The Industrial dataway system is structured into five functional levels with respect to data transmission shown in Figure-3.

These five levels include:

(1) Physical link

(2) Communication link

(3) Logical connection

(4) Network control

(5) Application (Devices)

The functions and protocols of each level are described in Section 5.

4.2 FRAME STRUCTURE

The HDLC's format shall be adopted as the frame structure of message for data transmission is the Industrial dataway system.

The HDLC's format is

F	DA	C	SA	I	FCS	F

(1) F: Flag field (8 bits)

F field designates the start and end of the frame and has the bit structure of

0 1 1 1 1 1 1 0

(2) DA: Destination address (8 bits)

DA field contains the destination address of the station to be received message.

(3) C: Control field (8 bits)

C field defines type of control (ex. command/reply) depending on the bit structure of this field.

(4) SA: Source address (8 bits)

Basically, SA field contains the source station address of the message. This field is necessary for data transmission on the n:n corresponded communication.

(5) I: Information

I field consists of Header, Control status, Device address, Test (data) and so on. Length of this field is varied upon amount of data to be transferred.

(6) FCS: Frame check sequence (16 bits)

Each message contains the FCS field for the purpose of detecting transmission error.

It is recommendable that checking is based on CRC.

5. PROTOCOL

The allocation of specific functional requirements for protocols is defined as follows:

AD-A058 096 PURDUE UNIV LAFAYETTE IND PURDUE LAB FOR APPLIED IND--ETC F/G 9/2
ANNUAL MEETING OF INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL C--ETC(U)
1977 N00014-78-C-0127

UNCLASSIFIED NL

3 OF 4
AD A058 096

5.1 PHYSICAL LINK PROTOCOL

The detailed requirements of a physical link protocol are dependent on link or line physics and must be transparent to and independent of the message bit stream including communication link addresses, checking codes, etc.

If message frame delimiters are dependent on line physics, they must be added and subtracted by the line coupling unit. (For example, the Modem in FSK carrier situations.)

Typical of existing standards of this protocol are:

(1) C C I T T V.24 (EIA RS-232)

(2) C C I T T X.21

(3) 20mA current loop

The physical interface is characterized in terms as follows:

(1) The transmission data rate of 1 M bps or more.

(2) Bit synchronization.

(3) Clock extraction (Bit element timing).

(4) Base-band transmission.

(5) Electrical isolation from transmission signal line.

(6) Redundant (Dual or Duplex) data transmission paths should be optional.

(7) Bypassing and disconnect capability.

5.2 COMMUNICATION LINK PROTOCOL

The communication protocol is specific to the communication subsystem but is independent of the signalling techniques employed in the physical communication link. This protocol is also independent of the characteristics of the stations attached to the communication subsystem.

The Communication Link Protocol of Industrial Dataway System should be organized on the basis of ISO HDLC and characterized in terms as follows:

(1) n:n (or n:m) communication capability.

(2) Frame structuring based on the HDLC to transmit unformatted binary data (data transparency).

(3) Transmission error detection by CRC.

(4) Error recovery by automatic retransmissions. No-response timer and retransmission counter are required to prevent from unduly delay other messages.

(5) More than one communication subsystems should be responsible to recover from the disappearance of current master.

(6) Broadcasting command is required to transmit global messages to all subsystems.

(7) Initialization capability of communication subsystems.

5.3 LOGICAL CONNECTION PROTOCOL

This layer has to do with the synchronization of the transmit and the receiving hardware on both sides. Since this hardware attaches to the processor channel I/O interface, a function must exist in the hardware to reflect to the processor a logical channel read/ write interface.

(1) This level shall provide arbitration between the message length capabilities of the communication subsystem and the message length requirements at the application level. Examples are rejection or blocking and deblocking of messages which exceed the message length capabilities of the communication subsystem.

(2) Establish and release logical data channels.

(3) Watch Dog Timer to recover from hang-up of Logical Connection.

(4) IPL capability for distributed intelligence.

6. SAFETY AND SECURITY

6.1 TRANSMISSION ERROR PROCESSING

The Industrial dataway system shall be capable of supporting fault diagnosing capability.

This capability is classified into two categories. One shall be included in Industrial dataway system and the other shall be supported by outside scope of Industrial dataway system itself.

The former might include as follows:

(1) Error control by using CRC checking and automatic retrial functions

(2) Sense status of all station

(3) Monitoring response time interval

(4) Shift the frame synchronizing station or clock unit in case of failure

(5) Correction for log of synchronization

(6) Data sequence checking

The latter as follows:

(1) Code and address error checking

(2) Centralized supervisory for all station and communication line operation

(3) Test or diagnostics for all stations under on-line operation

The Industrial dataway system shall be capable of maintaining correct sequencing and integrity of transmitted data through an electrically noisy environment. The error detection scheme must cover noise pickup in both directions and all data transmitted ("data", address, status, and control signals).

6.2 SYSTEM FAILURE DETECTION, PROTECTION AND RECOVERY

(1) Automatic remote station bypassing by using automatic troubles and errors detecting, self loop checking and isolating from active communication link.

(2) Manual station bypassing from active communication link without any system disturbance or error.

(3) Re-organization of communication link (self link configuration)

(4) Extensive use of selfchecking software should provide reliable supervision with possibility to locate and diagnose potential system errors.

6.3 OTHERS

(1) System expansion and modification under on-line operation of data communications.

(2) The communication subsystem shall include optional versions which will survive lightning strikes or power faults on the transmission medium and will prevent the equipment connected to the subsystem from becoming hazardous when these conditions occur.

(3) The communication subsystem shall meet the relevant mandatory standards of licensing agencies.

(4) The communication subsystem shall include optional versions for which intrinsic safety certification can be obtained.

(5) Easy to make dual staitons or dual power supply units.

(6) Reliable transmission cable way system including dual cabling, protection for disconnection and clamp, and noise protection for transmission signal.

(7) The physical implementation of the interface within the subsystem should be designed so that the subsystem is isolated and may perform electronic state transitions such as on-line/off-line, power-on/power-off, ready/not ready, busy/not busy, local/remote, etc. without generating transmission errors between other subsystems.

7. MAINTENANCE

7.1 ON-LINE MAINTENANCE

(1) The communication subsystem shall be capable of supporting on-line testing and fault diagnosing capability.

This might include traffic monitoring and on-line and remote loopback test facilities. There features are not integral functions of the communications subsystem.

7.2 OFF-LINE MAINTENANCE

(1) Maintenance panel shall be connected to the station for test of diagnostics under off-line mode.

GLOSSARY OF PROPOSED GUIDELINE

	Terms	Section
(A)	active communication link	6.2 (2)
	address	3.2 (2), 6.1 (1), 6.1 (3)
	application	4.1
(B)	base band	3.11
	bit serial	1.2 (1), 3.8
	branch	3.1 (1), Figure-2
	broadcasting command	5.2 (6)
(C)	clock extraction	5.1
	clock unit	6.1 (4)
	command	2.2.3 (1)
	common carrier channel	2.2.3 (5)
	communication line	1.2 (3), 6.1 (2)
	communication link	2.2.3 (1), 4.1, 7.2 (3)
	communication subsystem	2.2.3 (5), 6.3, 7.1
	communication system	1.2 (2), 3.9 (2)
	control frame	3.1 (2)
(D)	data communication	1.2 (2), 2.1, 2.2.1 (2), 2.2.3 (4), 6.3 (1)
	data sequence check	6.1 (2)
	data transmission	2.1 (2), 2.2.3 (2), 4.1, 4.2
	data transmission path	5.1
	data transparency	5.2 (2)
	destination address	4.2 (2)
	device	1.2 (5), 2.1 (2), 2.2.3 (1) 3.6, 3.7, 3.13 (2)
	device interface	2.1 (2)
(F)	flag	4.2 (1)
	frame	3.1 (2), 4.2
	frame check sequence	4.2 (6)
	field	4.2

	Terms	Section
(G)	global message	5.2 (6)
(H)	header control status	4.2 (5)
(I)	Industrial Dataway System information	4.2 (5)
(L)	line coupling unit	5.1
	line physics	5.1
	line sharing system	2.1 (1), 2.1 (2)
	link	2.2.3 (1), 3.2 (1), 3.2 (3), 4.1, 5.1
	link control	2.2.3 (3)
	logical channel	5.3
	logical connection	4.1, 5.3 (3)
	loop	3.1 (3), Figure-2
(M)	master	2.2.3 (3)
	mastership	3.2
	master station	2.2.3 (2), 2.2.3 (3), 3.2 (1)
	message	2.2.3 (1), 4.2
	message blocking	5.3 (1)
	message deblocking	5.3 (1)
	message switching	3.1 (4)
	multi-mastership	3.2 (1)
(N)	network control	1.2 (3), 2.2.1 (2), 4.1
	n:n corresponded communication	3.2 (2), 4.2 (4)
(O)	off-line	6.1 (3), 6.3 (1), 6.3 (7), 7.1
	on-line	6.3 (7), 7.2 (1)
(P)	physical interface	5.1
	physical link	4.1
	protocol	Figure-3, 5

	Terms	Section
(R)	receiving	5.3
	remote loopback	7.1 (1)
	repeater	3.4 (1), 3.4 (2)
(S)	self loop checking	6.2 (1)
	signal line	3.12 (1)
	single line sharing	1.2 (1), 3.1 (2)
	slave	2.2.3 (3)
	slave station	2.2.3 (2)
	source address	4.2 (4)
	station	2.2.3 (2), 3.2 (1), 3.2 (3), 3.4 (2), 3.5, 3.6, 3.7, 3.13 (2)
	station bypassing	6.2 (1), 6.2 (2)
	subsystem	1, 2.2.3 (2), 3.6, 3.2 (3) 6.3 (2), 6.3 (7)
	synchronization	3.11, 6.1 (5)
	synchronization signal	3.1 (5), 3.1 (2)
(T)	text	4.2 (5)
	transmission	6.3 (6), 2.2.3 (2), 2.2.3 (3)
	transmit	5.3

190

APPENDIX-1

TYPICAL IMPLEMENTATION IN CASE OF CENTUM F-BUS

APPENDIX-2

TYPICAL IMPLEMENTATION IN CASE OF TOSWAY-1500

APPENDIX-3

TYPE OF DEVICES TO BE CONNECTED INDUSTRIAL DATAWAY SYSTEM

APPENDIX-4

LAYER OF PROTOCOL

APPENDIX-5

DEVIATION BETWEEN FUNCTIONAL REQUIREMENTS AND PROPOSED GUIDELINE (JAPAN)

APPENDIX-1

TYPICAL IMPLEMENTATION

IN CASE OF

CENTUM F-BUS

April 23, 1977

1. INTRODUCTION

Based on the proposed guideline.

2. APPLICATION ENVIRONMENTS

Based on the proposed guideline.

3. FUNCTIONAL SPECIFICATION

3.1 TOPOLOGY OF CENTUM F-BUS

The F-BUS, which is a component of all CENTUM systems, allows data to be transferred around the control system at a 250K bps rate.

It incorporates coaxial cables, coupler units, branch units, repeaters, power supplies and communication controllers.

It can be up to 10 kilometers long, and can be connected through computer adapters to virtually any computer having data communication capabilities. (Fig.-1)

CENTUM F-BUS has a branch type configuration, and each station is connected to coaxial cable with passive (transformer) coupler.

3.2 MASTERSHIP IN CENTUM F-BUS

(1) Based on the proposed guideline.

Mastership is sent and received one after another by "Baton Pass" command.

(2) Based on the proposed guideline.

The communication command frame and response frame have not only destination address but also source address.

(3) Based on the proposed guideline.

3.3 TRANSMISSION SPEED

250K bps

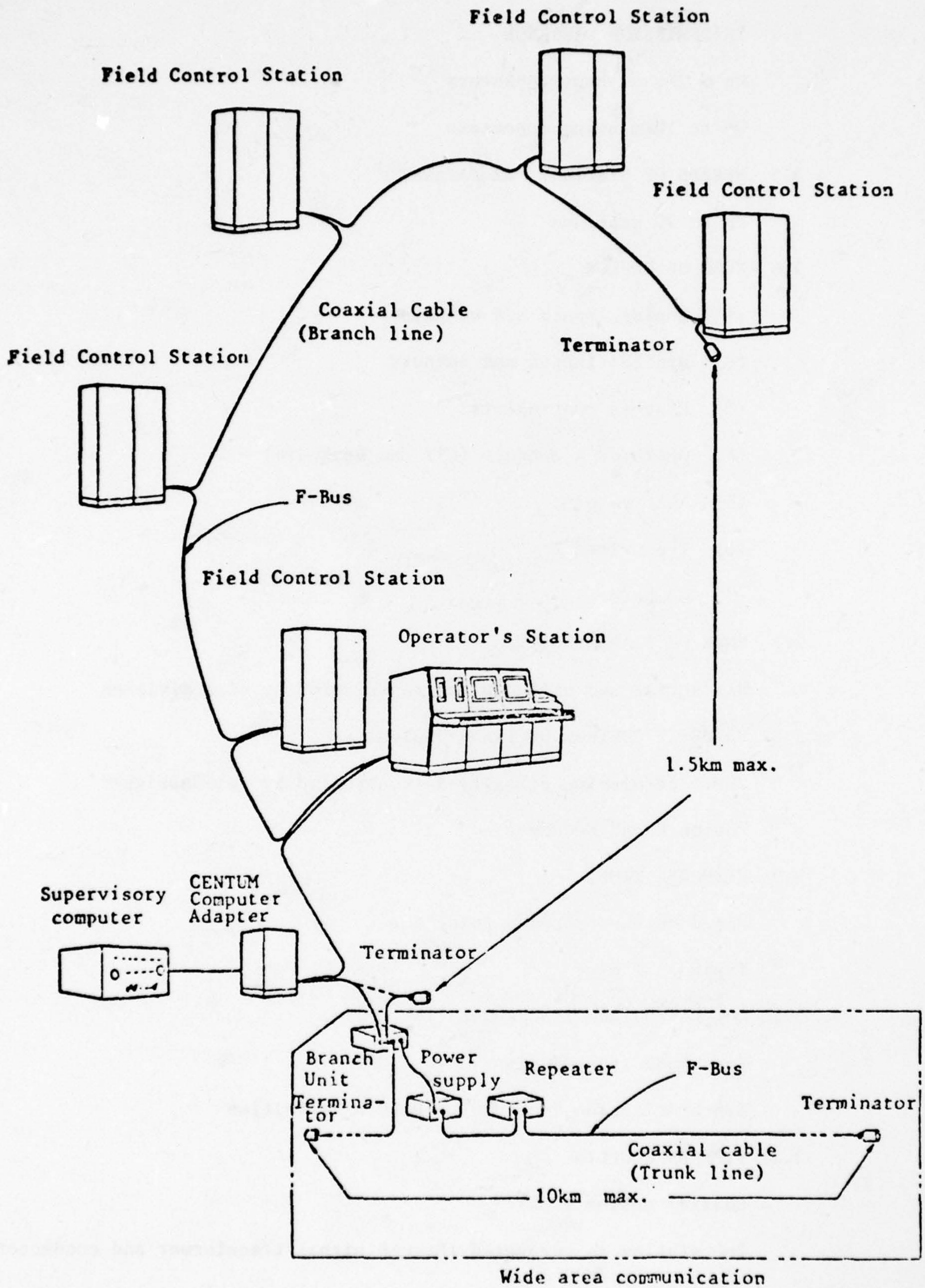


Fig.-1 Example of Inter-Station Connection using F-Bus

3.4 TRANSMISSION DISTANCE

Up to 2Km without repeaters

Up to 10Km using repeaters

3.5 NUMBER OF STATIONS PER DATAWAY

Up to 30 stations

3.6 TYPE OF DEVICE

(1) Analog inputs and outputs

(2) Digital inputs and outputs

(3) Process controllers

(4) Operator's console (CRT and keyboard)

(5) Aux. memory

(6) Typewriter

(7) Computer

3.7 MODE OF TRANSMISSION

Bit serial and half-duplex transmission by time division

3.8 PRIORITY CONTROL OF TRANSMISSION

The transmission priority is controlled by well assigned "Baton Pass" sequence

3.9 RESPONSE TIME

Based on the proposed guideline

Typical 2 msec

3.10 MODULATION AND SYNCHRONIZATION

Base-band transmission

Synchronization is based on HDLC's definition

3.11 TYPE OF COUPLING

Coaxial cables (50Ω)

The station is connected through signal transformer and connector.

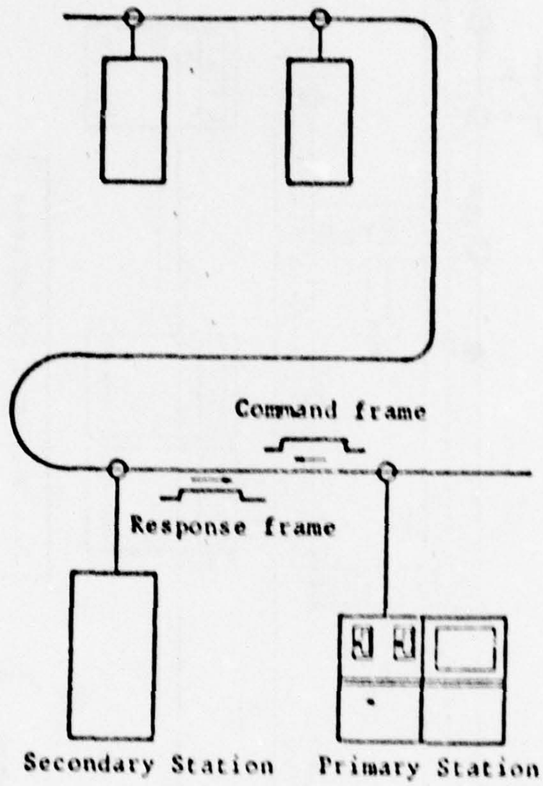


Fig.-2 Outline of F-Bus Communication

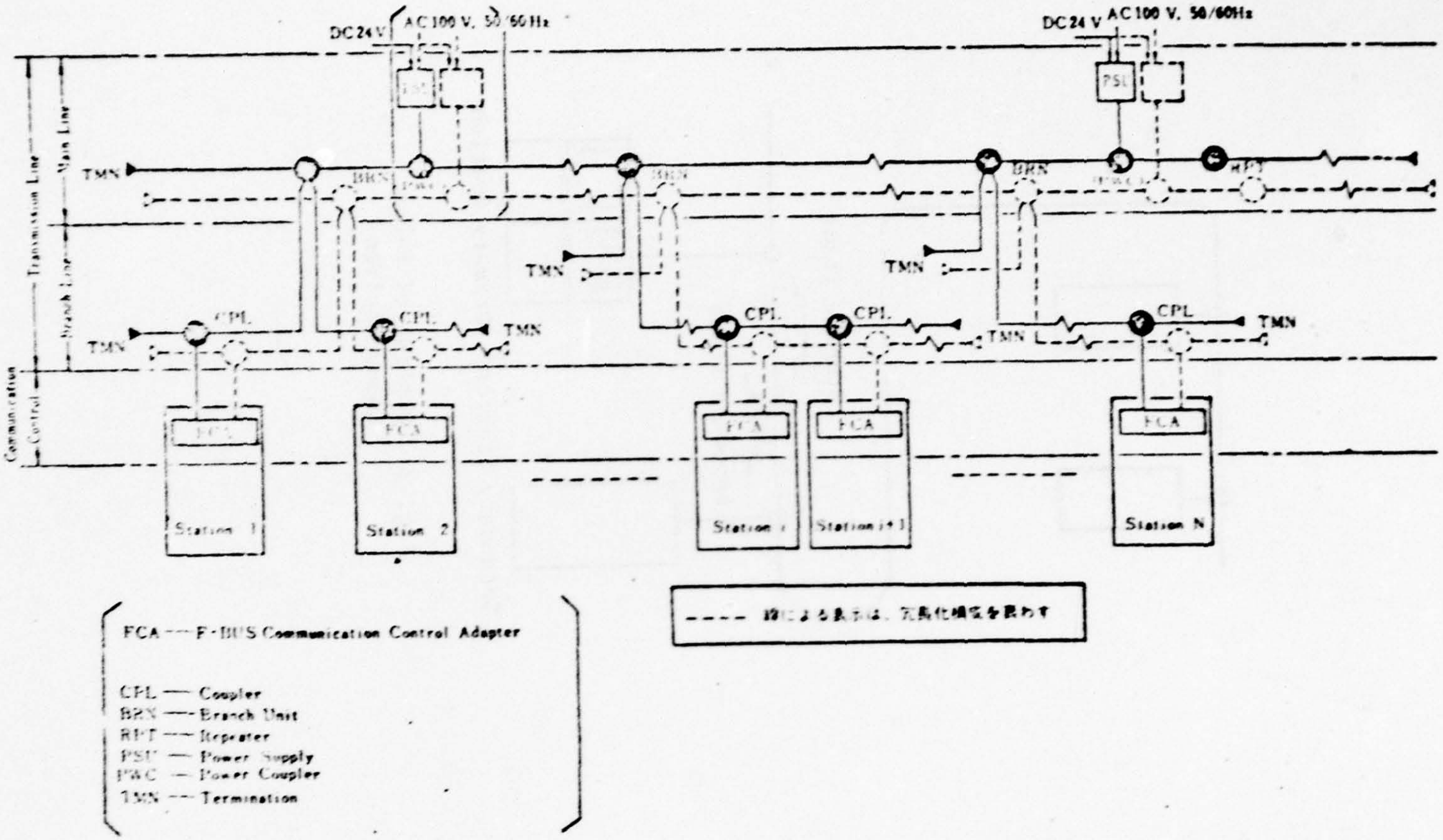


Fig.-3 System Structure of CENTUM F-BUS

3.12 EXPANDABILITY

Based on the proposed guideline

3.13 ENVIRONMENTAL CONDITION

(1)	Operating temperature range	0 to 50°C
(2)	Operating humidity	40 to 90% RH (No condensation)
(3)	Grounding (Earth)	less than 10 ohms
(4)	Withstand voltage	
	for power line	AC 1 KV, 1 min.
	for signal line	DC 500V, 1 min.

4. ARCHITECTURE

4.1 SYSTEM STRUCTURE

From the viewpoint of communication, stations are classified into the primary stations and the secondary stations. The primary station is a station having the initiative of communication. Communication is conducted by a command frame from the primary station and a response frame from the secondary station. Communication on the F-Bus is conducted on a single coaxial cable which conveys a command frame and a response frame one after another. Fig.-2 illustrates the way of communication.

4.2 FRAME STRUCTURE

SYNC	LCW 1		LCW 2		DATA	FCS
	DEST	C	SRCE	MDFY		
8 BITS	8 BITS	8 BITS	8 BITS	8 BITS	UP TO 1008 BITS	16 BITS

(1) SYNC : SYNCHRONOUS FLAG, 0 1 1 1 1 1 1 0

(2) LCW 1,2 : LINK CONTROL WORD

(3) DEST : DESTINATION ADDRESS

(4) C : CONTROL (COMMAND/RESPONSE)

(5) SRCE : SOURCE ADDRESS

(6) MDFY : MODIFIER

(7) DATA : MESSAGE DATA

Message Data has variable length of up to 1008 bits, and 16 bit (1 word) increment.

(8) FCS : FRAME CHECKING SEQUENCE

FCS is the 16 bit CRC of $X^{16} + X^{12} + X^5 + 1$

5. PROTOCOL

There are 4 levels of protocol for the CENTUM F-BUS.

(1) Physical link protocol

(2) Communication link protocol

(3) Logical connection protocol

(4) Application protocol

These protocols are implemented in F-BUS as Fig.-4.

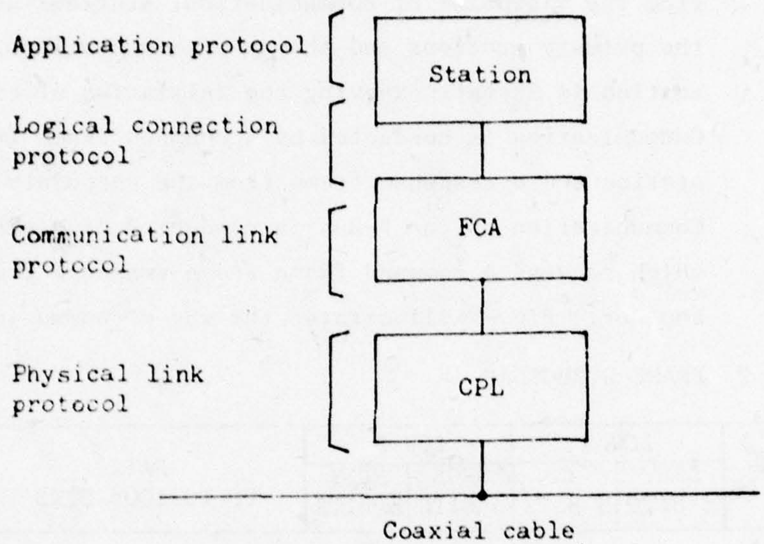
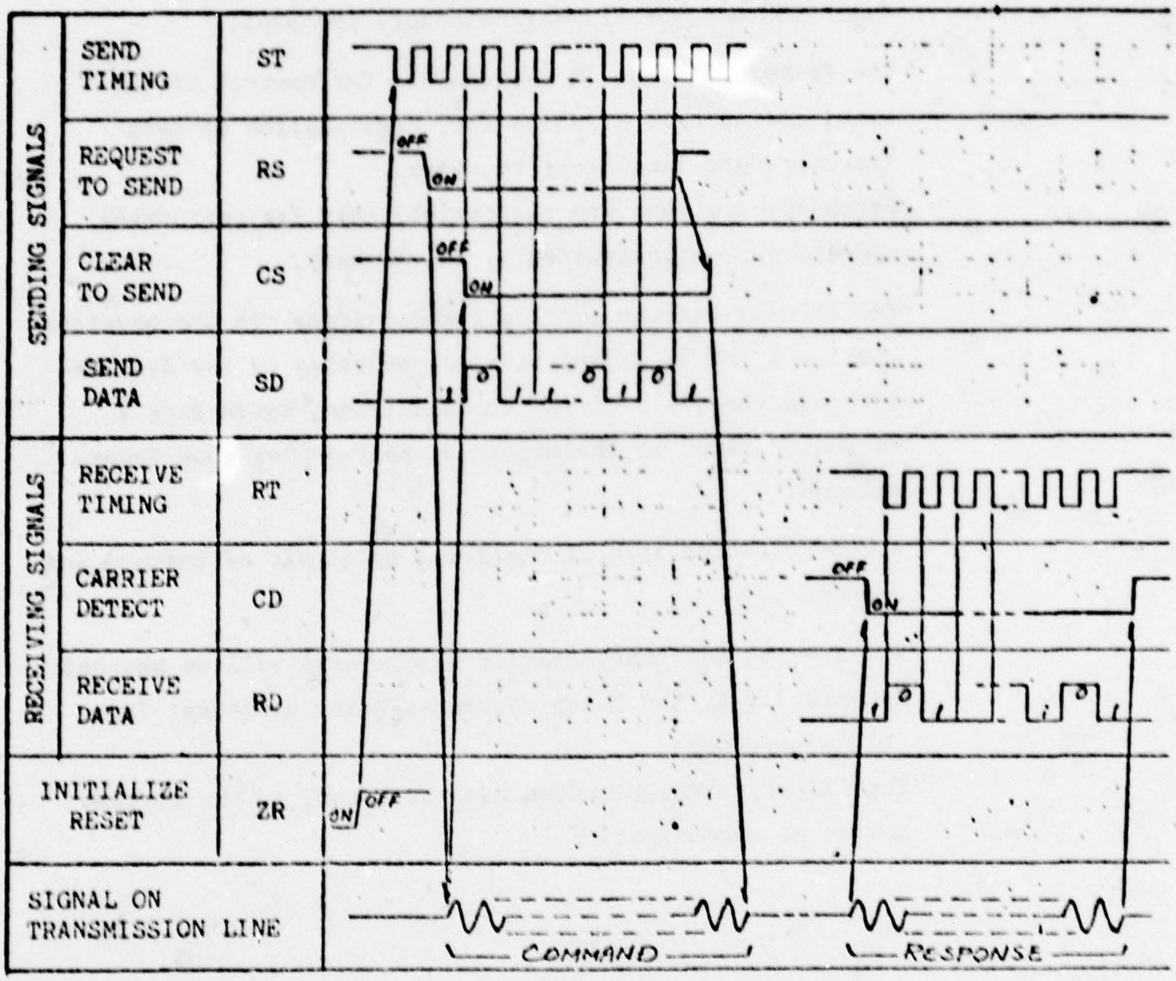
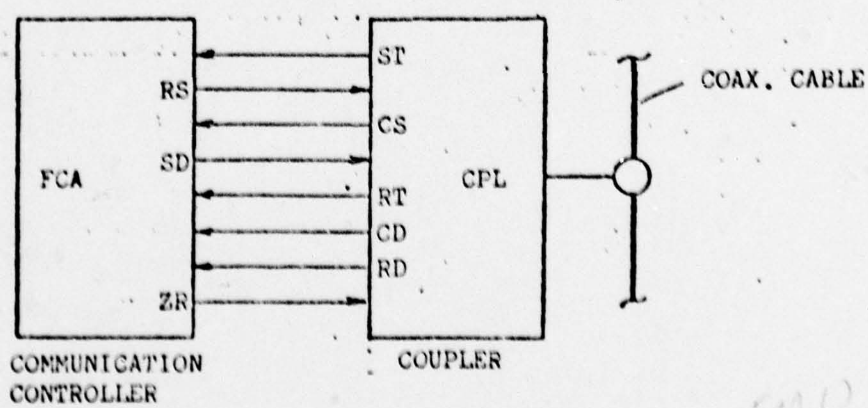


Fig.-4 Layers of protocol in F-BUS

5.1 PHYSICAL LINK PROTOCOL





5.2 COMMUNICATION LINK PROTOCOL

5.2.1 CONTROL PROCEDURES

The F-BUS has the Primary/Secondary protocol.

The Primary station is responsible for control of the link, including initialization, organization of data transfer, and link error recovery.
Secondary stations are responsible only for performing operations, as instructed by the Primary.

The Primary station sends a command frame via the physical link, and the Secondary station, selected by the destination address field of the command frame, sends back a response frame, in the manner of half-duplex mode transmission.

A communication link is completed by a pair of command and response.

Primary and Secondary station assignments will be changed dynamically by the Baton sequence, being different from HDLC's proceudre.
The Primary station, which has the Baton, is the current master of communication.

5.2.2 COMMAND AND RESPONSE

The set of Commands and Responses are defined in Table-1.

COMMAND

CONTROL BITS								MNEMONIC	COMMAND
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7		
0	0	0	0	1	1	0	0	BRD	Buffer Read
		0	0	1	1	1	0	RD	Read
		0	0	1	1	1	1	WT	Write
		0	1	1	1	0	0	EQ	Enquire
		0	1	1	1	1	0	POL/STS	Poll/Sense Station Status
		0	1	1	1	1	1	INZ	Initialize Reset
		1	1	1	1	0	0	BP	Baton Pass
		1	1	1	1	0	1	RBCR	Reset Broadcast Reject
		1	1	1	1	1	0	IPL	Initial Program Load
		1	1	1	1	1	1	BCR	Broadcast Reject
0	1	WORD COUNT						SND	Send
1	0	WORD COUNT						EXC	Exchange
1	1	WORD COUNT						EXT	(Command Extension)

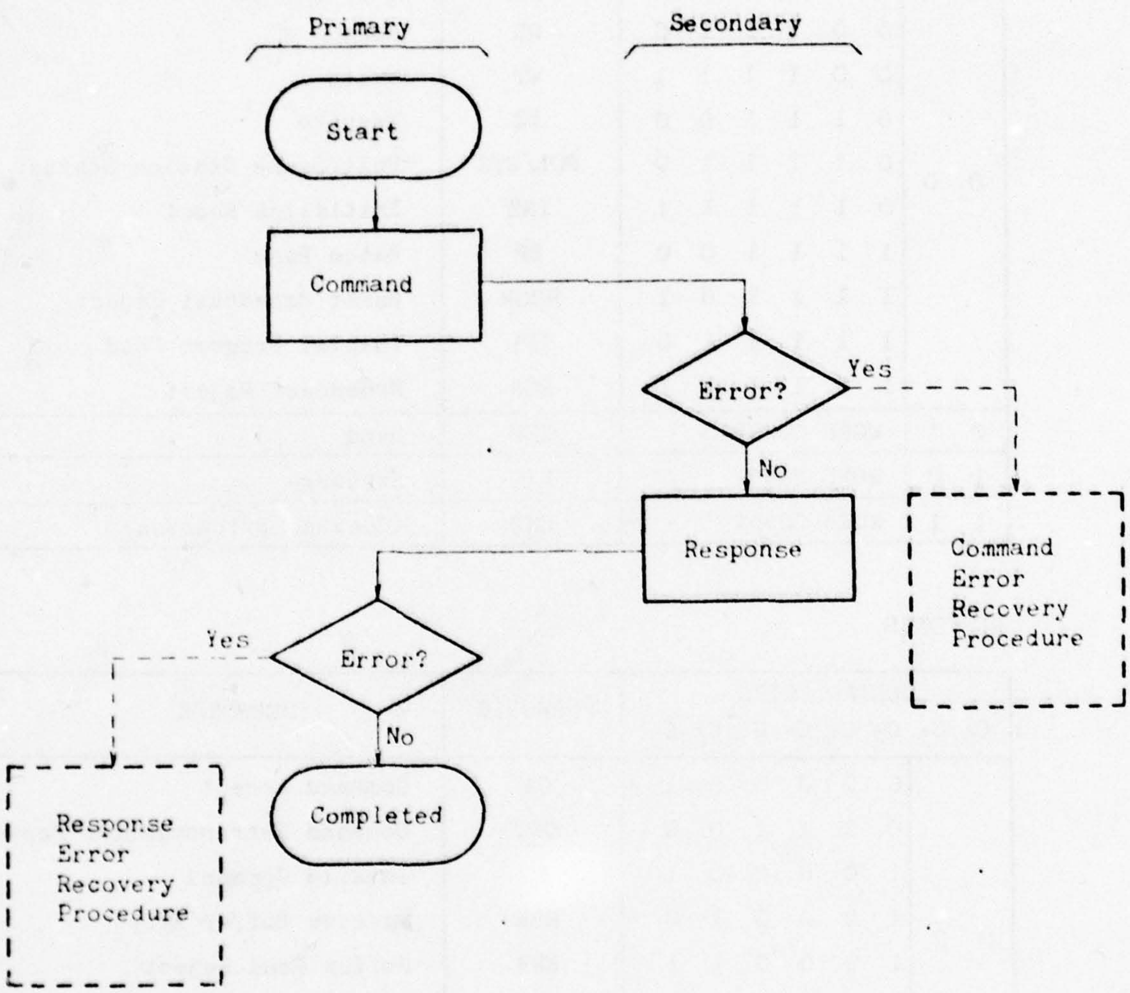
RESPONSE

CONTROL BITS								MNEMONIC	RESPONSE
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7		
0	0	0	0	0	0	0	1	CA	Command Accept
		0	1	1	1	0	0	CRT	Command Retransmission Request
		1	0	0	0	0	1	IC	Invalid Command
		1	0	0	0	1	0	RBB	Receive Buffer Busy
		1	0	0	0	1	1	BRR	Buffer Read Reject
		1	0	0	1	0	0	MNR	Memory No Response
		1	0	0	1	0	1	MPE	Memory Parity Error
		1	0	0	1	1	0	CNR	CPU No Response
0	1	WORD COUNT						ACK	Acknowledge
1	0	WORD COUNT						NAK	Negative Acknowledge
1	1	WORD COUNT						EXR	(Response Extension)

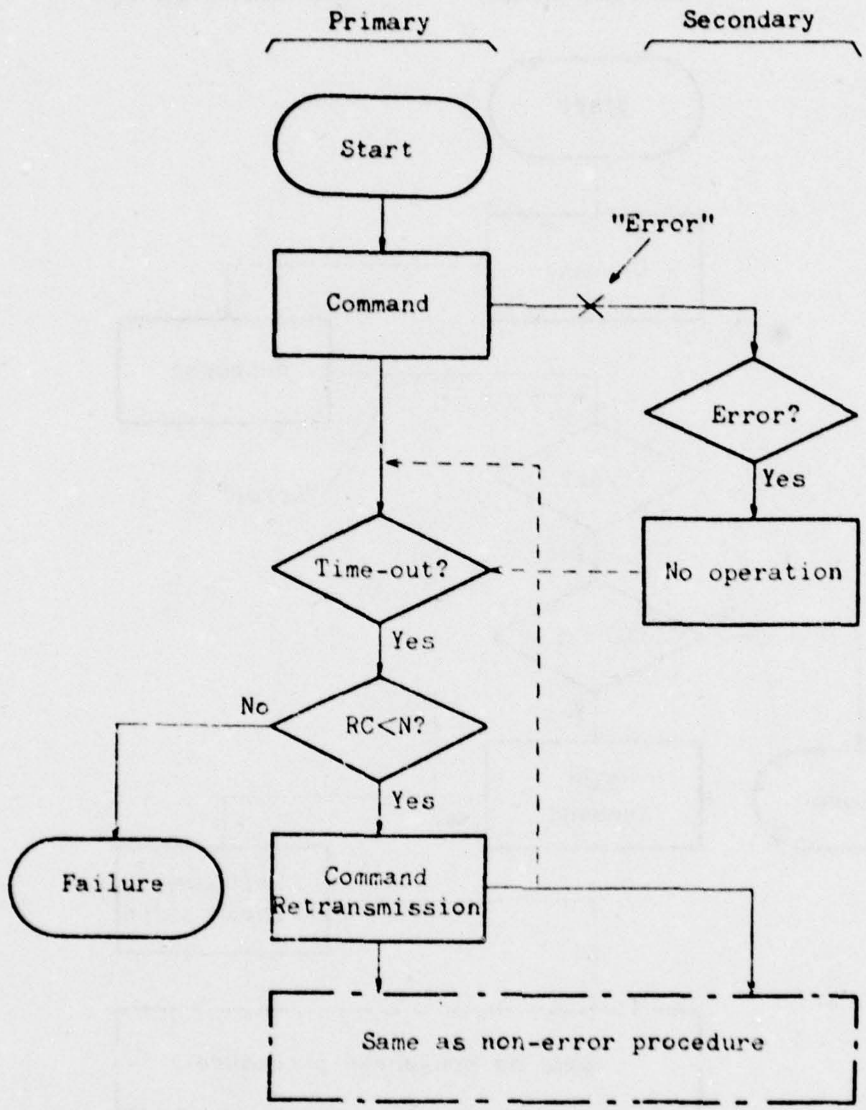
Table-1 Commands and Responses

5.2.3 ERROR RECOVERY PROCEDURES

(1) Non-error Procedures

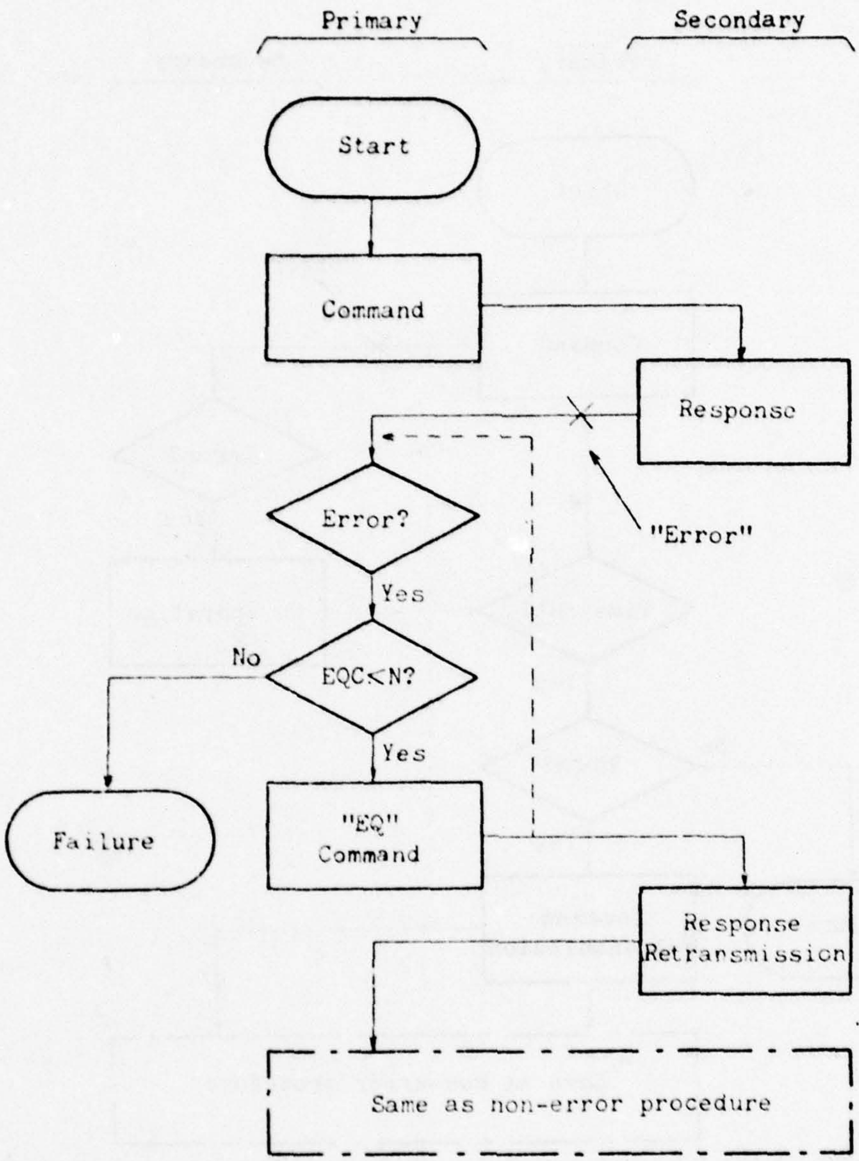


(2) Command Error Recovery Procedure



RC: Retransmission Count

(3) Response Error Recovery Procedure



EQ : Enquire

EQC: Enquire Count

5.3 LOGICAL CONNECTION PROTOCOL

To control and exchange various data between stations, F-Bus has logical procedures as follows.

(1) Send

(2) Send Sequence

(3) Send and Receive

(4) Send and Receive Sequence

(5) Send Wait and Receive

(6) Send Wait and Receive Sequence

(7) Read

(8) Read Sequence

(9) Write

(10) Write Sequence

(11) IPL

(12) Polling and Buffer Read

(13) Status Sense

(14) Initialize Reset

(15) Baton Pass

(16) Broadcast Reject

(17) Broadcast Send

(18) Broadcast Send Sequence

(19) Broadcast Initialize Reset

6. SAFETY AND SECURITY

(1) Error detection by CRC and parity check, and error recovery by automatic retransmission procedures.

(2) Duplicated transmission paths.

(3) Localization of the failed station by automatic and manual disconnection.

(4) No active elements (i.e. IC, transistor) on the transmission line, except for repeaters and branch units.

(5) Non-centralized multi-master stations.

7. MAINTENANCE

(1) System maintenance panel for system supervision.

(2) Remote IPL (Initial Program Load) function.

(3) Remote restart function.

(4) Remote saving the program of failed stations, making easy to analyze the causes of failure.

(5) Transmission signal level monitoring.

APPENDIX-2

TYPICAL IMPLEMENTATION
IN CASE OF
TOSWAY - 1500

April 23, 1977

1. INTRODUCTION

Based on proposed guideline.

2. APPLICATION ENVIRONMENTS

Based on proposed guideline.

3. FUNCTIONAL SPECIFICATION

3.1 TOPOLOGY OF TOSWAY-1500

(1) TOSWAY-1500 has a loop configuration to provide the reliable dataway system. (Fig. 1-a)

(2) TOSWAY-1500 consists of two transmission lines, one for data transmission and another for test signal in normal data transmission. In case of abnormal condition such as degradations of the station, these two lines make the return path at the both sides of the degraded station and disconnect it from the system to prevent the entire system from out-of-served condition. (Fig. 1-b,c)

(3) TOSWAY-1500 is not intended to handle message switching.

3.2 MASTERSHIP in TOSWAY-1500

(1) Satisfied.

(2) Satisfied.

(3) Satisfied.

3.3 TRANSMISSION SPEED

1,544 M bps

3.4 TRANSMISSION DISTANCE

(1) 100 Km with repeater

(2) 4 Km without repeater

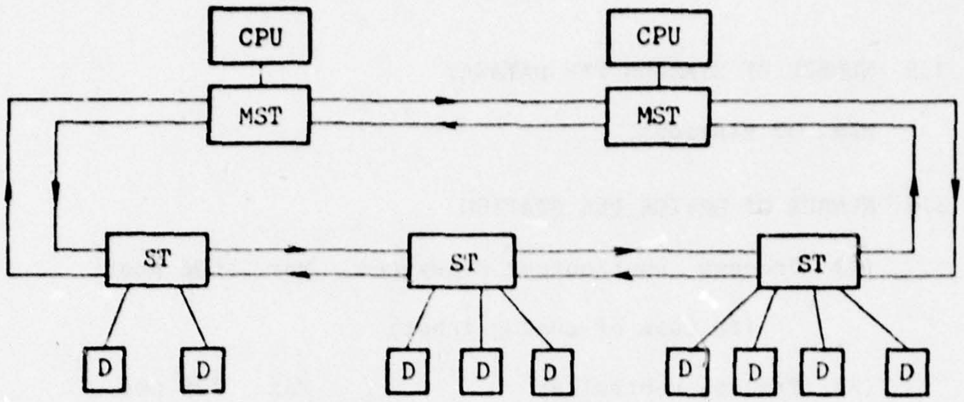


Fig. 1-a System Structure of TOSWAY-1500

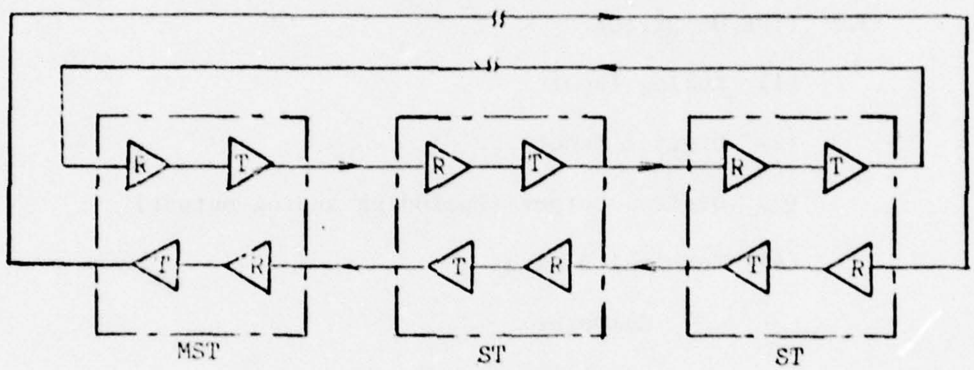


Fig. 1-b Data Transmission in Normal Condition

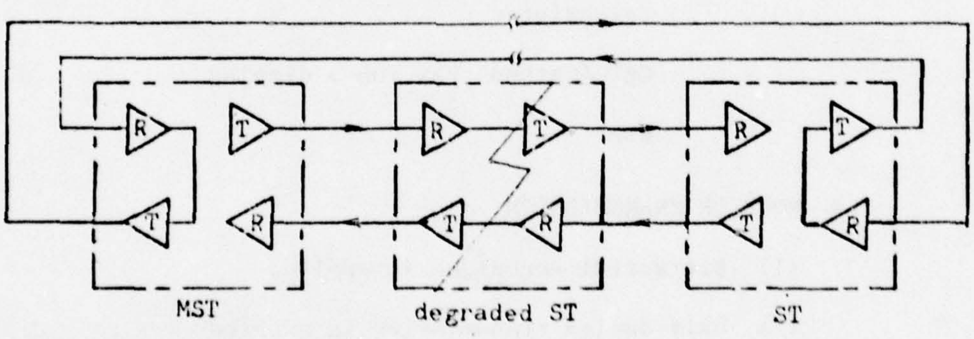


Fig. 1-c Data Transmission in Abnormal Condition

MST	:	Master Station	CPU	:	Central Processor Unit
ST	:	Station	R	:	Receiver
D	:	Terminal Device	T	:	Transmitter

3.5 NUMBER OF STATION PER DATAWAY

Max. 32 stations

3.6 NUMBER OF DEVICE PER STATION

(1) Process input/output subsystem : Max. 4096 pts.

(in case of analog input)

(2) Process controller : Max. 128 pts

(in case of 3 mode controller)

(3) Terminal device : Max. 16 sets

(in case of device subsystem)

3.7 TYPE OF DEVICE

(1) Analog Input

(2) Digital Input

(3) Digital Output (including analog output)

(4) Terminal Device

Computer

Card reader

Paper tape reader

Teleprinter

CRT (Cathod Ray Tube) display

etc.

3.8 MODE OF TRANSMISSION

(1) Bit serial technique is applied

(2) Half-duplex transmission is adopted

3.9 PRIORITY CONTROL OF TRANSMISSION

3.10 RESPONSE TIME

Satisfied.

3.11 MODULATION AND DEMODULATION

(1) Base-band is applied as modulation technique

(2) Synchronization is based on HDLC's definition

3.12 TYPE OF COUPLING

Twisted-pair cable is used for signal line, so that terminatl strip is adopted as coupling of signal line

3.13 EXPANDABILITY

(1) Satisfied

(2) Satisfied

3.14 ENVIRONMENTAL CONDITION

(1) Operating temperature range	0 to 50°C
(2) Operating humidity	below 95% (without dew)
(3) Shock	less than 0.15G at 10 to 50 Hz
(4) Earth potential	less than 10 ohm
(5) Withstand voltage	
for power line	AC 1500 V, 1 min.
for signal line	DC 500 V, 1 min.

4. ARCHITECTURE

4.1 SYSTEM STRUCTURE

TOSWAY-1500 is structured into five functional levels with respect to data transmission. (Fig.-3)

These five levels include:

(1) Physical link ◄► REP (Repeater) <u>in TOSWAY-1500</u>

(2) Communication link ◄► DLC (Data link controller)

(3) Logical connection ◄► STC (Station terminal controller)

(4) Network control ◄► Network control

(5) Application (Devices) ◄► Application (Devices)

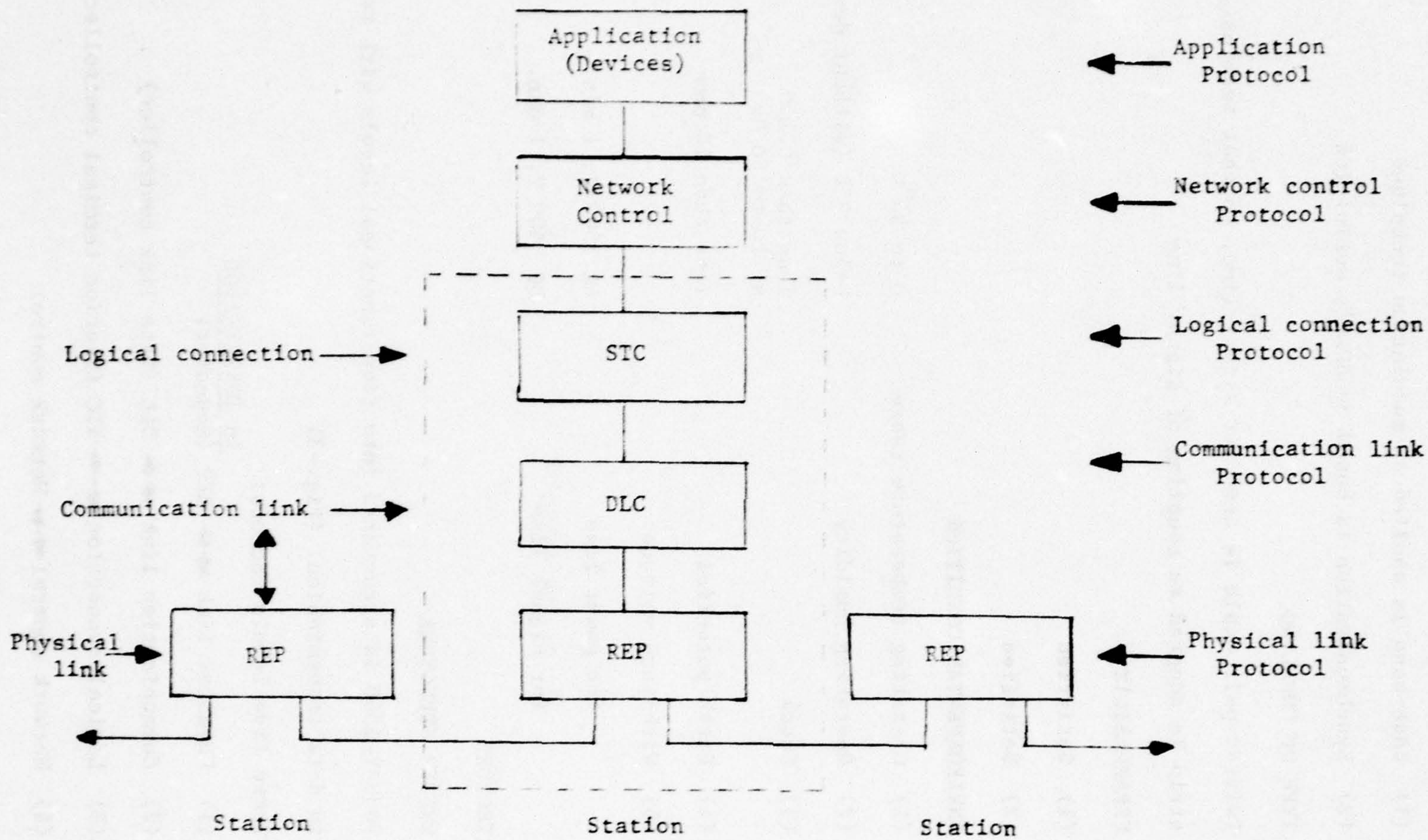


Fig.-3 System Structure of TOSWAY-1500

4.2 FRAME STRUCTURE

Basically, the HALC's format is adopted as the frame structure of message for data transmission in TOSWAY-1500. The format is:

F	DA	C	LV	SA	I	FCS	F

(1) F: Flag field (8 bits)

F field designates the start and end of the frame and has the bit structure of

0 1 1 1 1 1 1 0

(2) DA: Destination address (8 bits)

DA field contains the address of the station to be received the message.

(3) C: Control field (8 bits)

C field defines type of control (ex. command/reply) depending on the bit structure of this field.

(4) LV: Priority level field (8 bits)

LV field decides transmission order of messages in accordance with processing priorities of them.

(5) SA: Source address (8 bits)

SA field contains the source station address of the message.

(6) I: Information field

I field consists of Header, Control status, Device address, Texts (data) and etc. Length of this field is varied upon amount of data to be transferred.

(7) FCS: Frame check sequence field (16 bits)

Each message contains the FCS field for the purpose of detecting transmission error.

Checking adopted in TOSWAY-1500 is based on CRC.

5. PROTOCOL

5.1 PHYSICAL LINK PROTOCOL

Interface (in STC) can be coupled to:

(1) C C I T T V.24 standard devices

(2) C C I T T X.21 standard devices

(3) 20mA current loop

(4) TOSBAC-40 series computer

(5) Devices which are coupled to TOSBAC-40

The physical interface of TOSWAY-1500 is based on that of proposed guideline.

(1) The transmission data rate: 1,544 MBPS

(2) Bit synchronization

(3) Clock extraction (Bit element timing)

(4) Base-band transmission

(5) Electrical isolation from transmission signal line

(6) Redundant (Dual or Duplex) data transmission paths should be optional

(7) Bypassing and disconnect capability

5.2 COMMUNICATION LINK PROTOCOL

Communication link protocol of TOSWAY-1500 is based on proposed guideline.

(1) n:n communication capability

(2) Frame structuring based on the HDLC to transmit unformatted binary data (data transparency)

(3) Transmission error detection by CRC

(4) Error recovery by automatic retransmissions
No-response timer and retransmission counter are required to prevent from unduly delay other messages

(5) More than one communication subsystems should be responsible to recover from the disappearance of current master

(6) Transmitting global messages to all subsystems

(7) Initialization capability of communication subsystems

5.3 LOGICAL CONNECTION PROTOCOL

Logical connection protocol of TOSWAY-1500 is based on proposed guideline.

(1) Establish and release logical data channels

(2) Watch Dog Timer to recover from hand-up of Logical Connection

(3) IPL Capability for distributed intelligence

6. SAFETY AND SECURITY

6.1 TRANSMISSION ERROR PROCESSING

TOSWAY-1500 is capable of supporting fault diagnosing capability. This capability is classified into two categories. One is included in TOSWAY-1500, and the other is supported by outside scope of TOSWAY-1500.
(Ex. processor coupled to TOSWAY-1500)

1) The former capability is as follows:

(1) Error control by using CRC checking and automatic retrial functions

(2) Sense status of all station

(3) Monitoring response time interval

(4) Shift the frame synchronizing station or clock unit in case of failure

(5) Correction for log of synchronization

(6) Data sequence checking

2) The latter as follows:

(1) Code and address error checking

(2) Centralized supervisory for all station and communication line operation

(3) Test or diagnostics for all stations under on-line operation

6.2 SYSTEM FAILURE DETECTION, PROTECTION AND RECOVERY

(1) Automatic remote station bypassing by using automatic troubles and errors detecting, self loop checking and isolating from active communication loop

(2) Manual station bypassing from active communication link without any system disturbance or error

(3) Re-organization of communication link (self link configuration)

6.3 OTHERS

(1) Satisfied

(2) Arrestor equipped

(3) JIS

(4) None

(5) Duplex station is optional version

(6) Dual cabling is standard version

(7) Satisfied

7. MAINTENANCE

7.1 ON-LINE MAINTENANCE

(1) Satisfied

7.2 OFF-LINE MAINTENANCE

(1) Satisfied

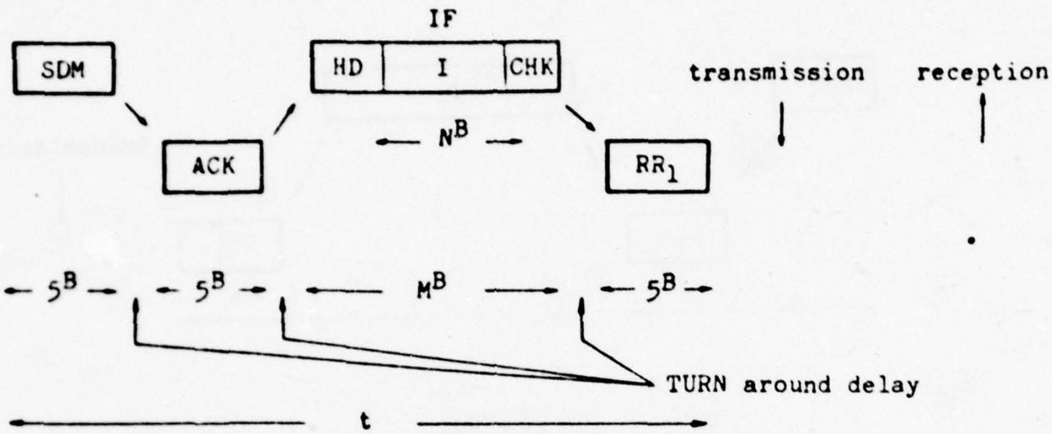
DATA SHEET:

TRAFFIC CHARACTERISTICS

1) TOSWAY-1500 Efficiency Calculation
   Formula of Information Transmission

2) For Example

1) TOSWAY-1500 EFFICIENCY CALCULATION FORMULA OF INFORMATION TRANSMISSION



where, SDM : BUSY/READY ENQUIRE (5 BYTES)

ACK : SDM ACKNOWLEDGE (5 BYTES)

IF : INFORMATION FRAME (N + 8 BYTES)

RR_1 : IF RECEIVED ACKNOWLEDGE (5 BYTES)

B : BYTE

TURN around delay

Translate at Bytes No. ---- 10 BYTES

The efficiency of information transmission is

$$V_{eff} = \frac{N}{t}(\delta^n + P_B)$$

$t = SDM + ACK + IF + RR_1 + 3 \times T$ (TURN around delay)

$= 45 + N$ BYTES

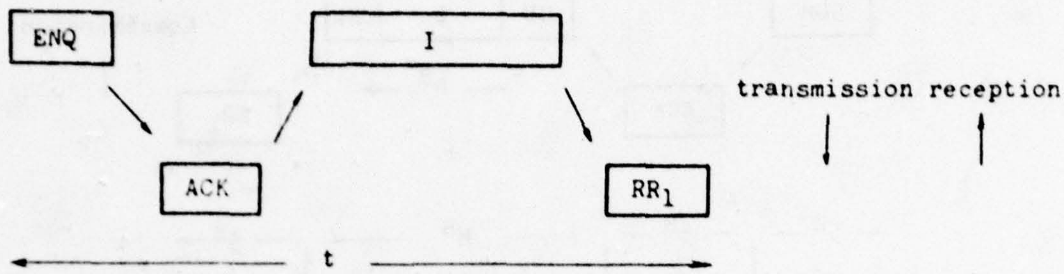
P : Bit error probability

$\delta = 1 - P$

$P_B \doteqdot 0$

n = 100 BYTES

2) FOR EXAMPLE



$$V_{eff} = \frac{I}{t}(o' + P_B)$$

$$= 1 - P \qquad P : \text{BIT Error Probability}$$

I = 100 BYTES

t = 45 + I = 145 BYTE

if $P_B = 0$,

$$V_{eff} = \frac{100}{145}(1 - P)^{100\times 8}$$

$P = 10^{-5}$

$(1 - P)^{800} = 68.4\%$

$P = 10^{-4}$

$(1 - P)^{800} = 63.6\%$

$P = 10^{-3}$

$= 30.9\%$

$P = 10^{-2}$

$= 0.022\%$

Figure X-1 Error Probability

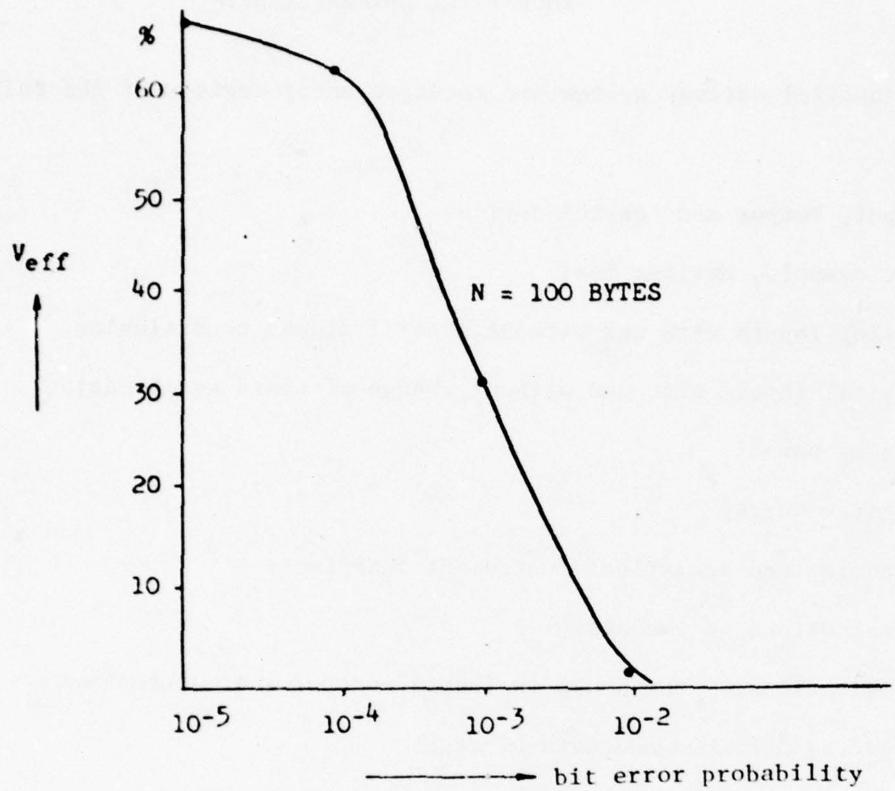
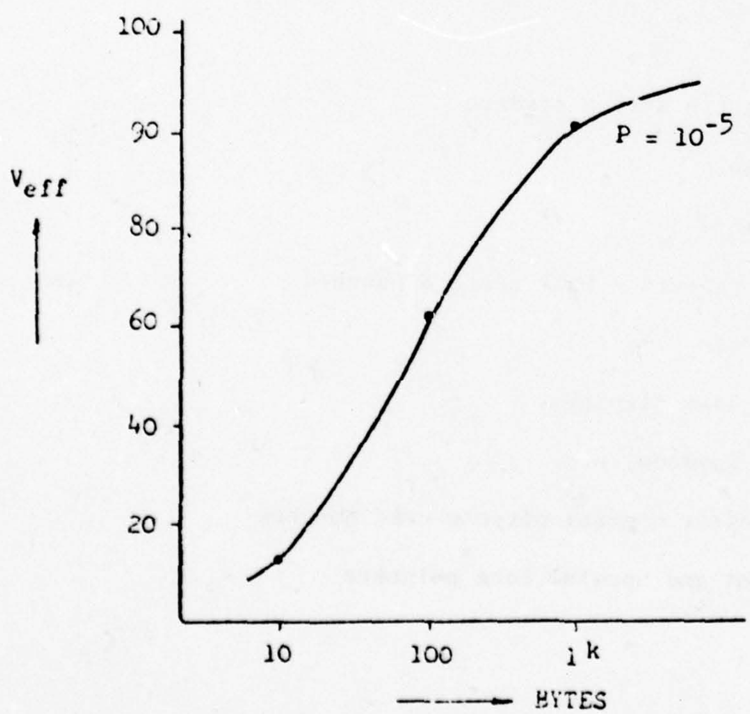


Figure X-2 Efficiency



APPENDIX-3 TYPE OF DEVICES TO BE CONNECTED INDUSTRIAL DATAWAY SYSTEM

The Industrial dataway system are required among devices of the following types:

(a) Input, output and control devices

For example, devices for:

Analog inputs with and without special signal conditioning.

Digital inputs with and without change of state notification.

Analog output

Digital output

Scanning and analytical instrument interfaces

Combinations of the above

Combinations of the above including control and maintenance logic, algorithms and data storage

Distributed digital control system.

(b) Man/Machine Interfaces and Product Devices

For example:

Badge & magnetic stripe readers

Label readers

Teleprinters

Single card readers - Mark sense & punched

Video terminals

Character & line displays

Keyboards & keypads, etc.

Dedicated and/or special purpose card punches

Label, ticket and special form printers

Alarm annunciators and other special displays

Combinations of the above

Combinations of the above including control and maintenance logic, algorithms and data storage

Note: These devices are recommendable to have standard interface such as EIA RS232C interface, 20mA current loop and others. These standards are suitable for standardization of Network control protocol.

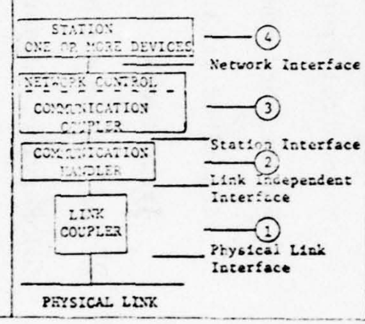
(c) Supervisory computers which control and interrogate devices described in 3.6.1, 3.6.2 and 3.6.4. These computers may exchange large blocks of data and/or programs with the other subsystems on an infrequent basis.

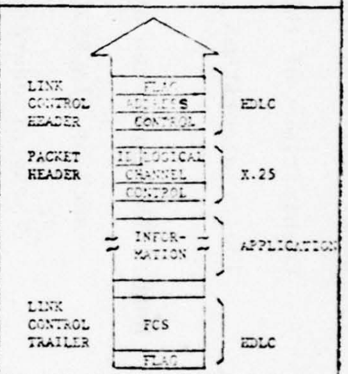
(d) Service, support and maintenance equipment

(e) Combinations of any or all of the above

APPENDIX-4 LAYER OF PROTOCOL

ISA SP72 (Secretary) 5 (1977-1-14)	Structural Layer of An Interface	IEC TC65A/WG6 Different Levels of Interface and Protocol Activity (1977-1-18)	DATAMATION, Mar., 1976 Layers of Network Access Protocols
5. Application Protocol Device-dependent Information Sub-address within the device Device Control or Status Signals Data Formats Character Codes	5. Application Protocol Device Data Format Control	4. System Protocols and Application Functions	4. Application Level Protocol How terminals and application programs talk to each other.
4. Network Control Protocol Choice from Inter-subsystem Alternate Paths, Switching Decisions for Routing	4. Network Description/Protocol (Systems Software) Routing Information Node Service in a Store and Forward Transmission Data Format Control	3. Network Protocol, Routing and Configuration Details	3. Packet Level Protocol How messages are identified to the network for routing and control. CCITT X.25
3. Logical Connection Protocol Establish and Release Logical Data Channels	3. State Transition Skeletal Description (Processor Channel I/O Interface) Read/Write Control Fetching Protocol Seq. of "Are you Ready, I am Ready" and Checking Data Sync. respect to Processor Storage		
2. Communication Link Protocol Data Transparency, Detection of Errors	2. Line Discipline/Protocol Structuring of a Message Frame, Data Blocking and Deblocking, Demand/Response Line Contention, Synchronization Data Checking, Clocking	2. Communication Link Protocol	2. Link Control Protocol How a terminal talks to the network over its physical circuit. BSC SDLC, HDLC
1. Physical Link Protocol	1. Physical Interface Characteristics Data Rate, Distance Modulation, Coding Techniques Signal Strength, Copper Size	1. Physical Link Protocol	1. Physical Circuit Protocol How devices are physically connected to the network. EIA RS.232 CCITT V.24, X.21





APPENDIX-5 DEVIATION BETWEEN FUNCTIONAL REQUIREMENTS AND PROPOSED GUIDELINE (JAPAN)

Contents	IEC SC65/WG6 (Secretary) 9	ISA SP72 (Secretary) 5	Japanese Proposed Guideline
1. Introduction			
o General	----	Refer to 1.0	
1.1 Objects of proposed guideline			
(1)	----	----	
(2)	----	----	
(3)	----	----	
1.2 Scope of guideline			
(1)	Refer to 1.1	Refer to 1.1	
(2)	Refer to 4.1	Refer to 4.1	
(3)	Part of 7.2	Part of 7.2	Add exclusion items.
(4)	Refer to 3.2	Refer to 3.2	
(5)	Refer to 3.4	Refer to 3.4	
1.3 Matters related to the functional requirements	yes	yes	----
2. Application Environments			
2.1 Review of Data Transmission System			
(1)	----	----	Add figure and table.
(2)	----	----	Purpose of line-sharing system is included.
(3)	Refer to 1.2	Refer to 1.2	

Contents	IEC SC65/WG6 (Secretary) 9	ISA SP72 (Secretary) 5	Japanese Proposed Guideline
2.2 Applicable Field of Industrial Dataway System			
2.2.1 Outline			
(1)	Refer to 2.2	Refer to 2.2	
(2)	----	----	Add.
(3)	----	----	Add.
2.2.2 Typical Industries			
(1)	----	----	Add.
(2)	Refer to 2.1	Pefer to 2.1	
2.2.3 Typical Configuration			
(1)	----	----	Add.
(2)	----	----	Add.
(3)	----	----	Add.
(4)	Refer to 3.1	Refer to 3.1	
(5)	Refer to 5.1	Refer to 3.1	
(6)	----	----	Add.
3. Functional Specification			
3.1 Topology of Industrial Dataway System			
(1)	----	----	Add.
(2)	----	----	Add.
(3)	----	----	Add.
(4)	----	----	Add.

Contents	IEC SC65/WG6 (Secretary) 9	ISA SP72 (Secretary) 5	Japanese Proposed Guideline
3.2 Mastership in Industrial Dataway System			
(1)	----	----	Add.
(2)	Refer to 4.3	Refer to 4.2	Add.
3.3 Transmission Speed	Refer to 3.5	Refer to 3.4	
3.4 Transmission Distance	Refer to 3.4 75 m 2000 m 30 km	Refer to 3.4 2 km	4 km (without repeater) 50 km
3.5 Numbers of station per dataway	----	----	Add.
3.6 Numbers of device per station	----	----	Add.
3.7 Type of device	Refer to 3.1	Refer to 3.1	
3.8 Mode of transmission	Refer to 1.1	Refer to 1.1	
3.9 Priority Control of Transmission	----	----	Add.
3.10 Response Time	----	----	Add.
3.11 Modulation and Synchronization	----	----	Add.
3.12 Type of coupling			
3.13 Expandability			
(1)	Refer to 4.3	Refer to 4.3	
(2)	Refer to 4.4	Refer to 4.4	
3.14 Environmental Condition	Refer to 6.3	Refer to 2.4	
4. Architecture			
4.1 System structure	Refer to 7.1 and 7.2 (Four levels)	Refer to 7.1 and 7.2 (Five levels)	

Contents	IEC SC65/WG6 (Secretary) 9	ISA SP72 (Secretary) 5	Japanese Proposed Guideline
4.2 Frame structure	----	----	Add.
5. PROTOCOL			
5.1 Physical Link Protocol	Refer to 7.4	Refer to 7.4	Add standards.
5.2 Communication Link Protocol	Refer to 7.5	Refer to 7.5	
5.3 Logical Connection Protocol	----	Refer to 7.6	
6. Safety and Security			
6.1 Transmission error processing	Refer to 6.1 and 10.1	Refer to 2.5, 6.1 and 10.1	Add details.
6.2 System failure detection, protection and recovery	Refer to 7.3.1, 11.1 and 11.2	Refer to 7.3.1, 11.1 and 11.2	Add details.
6.3 Others	Refer to 6.2,8.1, 8.2,8.3, 11.3 and 11.4	Refer to 6.2,8.1, 8.2,8.3,11.3 and 11.4	Add details
7. Maintenance			
7.1 On-line maintenance	Refer to 6.1	Refer to 6.1	Add details.
7.2 Off-line maintenance	Refer to 6.1	Refer to 6.1	Add details.
	Deleted 1.3 2.3 3.3 7.6 7.7 9.1 12.1	Deleted 1.3 2.3 3.3 7.7 7.8 9.1 12.1	

APPENDIX IX

PAPERS OF THE INDUSTRIAL REAL-TIME FORTRAN COMMITTEE

Documents included here are as follows:

1. Caro, R. H. Draft of S61.3

2. Maliszewski, Kazimierz, "New Subroutines Proposed for RT-FORTRAN."

Draft S61.3

INDUSTRIAL COMPUTER SYSTEM FORTRAN PROCEDURES FOR THE MANAGEMENT OF INDEPENDENT INTERRELATED TASKS

Jan 18, 1978

PRECEDING PAGE BLANK

1.0 SCOPE

This standard presents external procedure references for use in industrial computer control systems. These external procedure references provide means for task intercommunication and cooperation through the orderly management of independent task execution. The tasks are expected to be running in a multi-tasking or a multi-processing environment. These procedures usually will pass the information necessary for this orderly management to an executive routine which will ensure the correct operation of industrial computer control system. The method for ensuring orderly management is left to the processor.

The procedure references in this standard are intended to provide methods by which the task can inform the processor of the relationships between the task and other independent tasks under the control of the same processor. The standard provides the means to integrate independent tasks into a system of cooperating tasks when used in conjunction with sound program design but the implementation of this standard is no assurance that problems of non-cooperation will not arise.

1.1 DEFINITIONS

VIRTUAL PROCESSOR

A virtual processor is an environment in which an executable program can run and in which all the resources needed for the execution of the executable program are always available. A particular implementation serves to map the set of virtual processors on to the actual processor. This mapping may be controlled by an executive routine and is processor dependent. The virtual processor exists only when the set of pending executives is not empty, or when an executable program is in either the state Voluntary Suspend on the state Running.

CYCLIC EXECUTION

Cyclic execution is a set of executions of an executable program by its virtual processor.

FORTRAN

The computing language defined as full FORTRAN ISO R1539-1972 which is the same as American National Standard FORTRAN ANSI X3.9-1966.

MULTI-PROGRAMMING*

A mode of operation that provides for the interleaved execution of two as more computer programs by a single processor.

MULTI-PROCESSING*

A mode of operation that provides for parallel processing by two or more processors of a multi-processor.

MULTI-TASKING*

A multiple operation that provides for the concurrent performance or interleaved execution of two or more tasks.

PROCESSOR*

In a computer a functional unit that interprets and executes instructions.

TASK

The execution of an executable program. The ISO/DIS 2382/X definition of a task is more general, "A basic Unit of work to be accomplished by a computer". In this standard the basic unit of work is restricted to an executable program.

EXECUTABLE PROGRAM

A program in a form suitable for execution.

INDEPENDENT TASK

A task whose existence at any time is independent of the existence of any other task.

DEPENDENT TASK

A task whose existence is dependent on the existence of another task.

CYCLIC EXECUTION OVERRUN

Cyclic execution overrun occurs when a task is a set of cyclic executions of a program and the start of the execution of a program is to occur before the completion of the previous execution of the program. A cyclic execution overrun is an occurrence within a real processor. Note that a cyclic execution overrun can occur in a real processor when it will not occur in the tasks virtual processor.

----

* Definitions taken from ISO/DIS 2382/X

1.2 BACKGROUND INFORMATION

In a computing system individual tasks may have various relationships.

--Tasks can be independent

--Tasks can be independent but interrelated

--Tasks can be dependent

Industrial systems usually consist of independent but interrelated tasks and in these systems the tasks can have various features such as:

--Tasks can be initiated either at a specified time or by an event and such initiation can be carried out repetitively.

--Tasks can be initiated to be complete by a specified time. This is frequently called deadline scheduling.

--Tasks can be synchronized to clock time or with process events.

--Initiation can be deferred depending on time or event.

--Tasks can be suspended voluntarily or involuntarily.

--Tasks can be terminated through normal completion or through unforseen exception conditions.

--Tasks may share information.

--Tasks may send messages to other tasks.

--Tasks can terminate other tasks.

--Tasks can suspend other tasks.

--Tasks can be resumed after suspension

--Tasks may set, clear or test event indicators.

--Tasks may obtain unique control of a shared resource.

1.3 TASK FEATURES INCLUDED IN THIS STANDARD

In industrial computer systems, tasks are independent but interrelated. This standard does not address all the areas of independent tasks interrelationships but is concerned with those features that most commonly arise in industrial computer systems.

Table I shows those features covered by the standard and those excluded; however, the excluded features may affect the result of a request for cooperation from one independent task to another. Such restrictions are processor-dependent and outside the scope of this standard.

TABLE I

INCLUDED

Indpendent but interrelated tasks
Initiation at a given time, upon event, after a period or repetitively
Suspension of a task by itself
Resumption of a task by another task
Resumption of a task by an event, at a given time or after a period
Termination by a task of another task or itself
Setting/clearing/testing of event conditions

EXCLUDED

Dependent tasks
Process-dependent exception conditions such as overflow/underflow
Task suspension by another task
Suspensions caused by the processor
Suspensions caused by the executive routine
All methods of sharing variables or arrays
All methods of sharing files
Sending messages between tasks
Termination of tasks based on events in the executive routine
All other functions of the executive
Deadline scheduling of tasks
Boolean operations on events

2. STATES OF A TASK

A task can exist in four states which define the operation of the task. All procedure references concerned with task interrelationships involve the change of a task from one state to another state.

The states of a task are defined in terms of a task's virtual processor. A virtual processor is defined in Section 1.1. A virtual processor has all of it's resources always available for the execution of the task. The executive performs the necessary mapping of a set of virtual processors onto the actual processors. The mechanism used for this mapping is processor dependent. Included in the resources necessary for task execution are: memory space, central processor(s) time, and general computer peripheral availability. The mapping of the virtual processors onto the real processor frequently implies that the time for a task to execute to completion in its virtual processor is less than the time for the same task to execute to completion in the real processor. Thus the mapping of the virtual processors onto the real processor is a measure of system loading which is clearly both system design dependent and processor dependent.

Each separate initiation of a task creates a new virtual processor. Each such virtual processor is identifiable to the executive program and the mechanism for identification is processor dependent. A cylic or repetitive initiation of a task by a single procedure reference has only one virtual processor for all the repetitive executions of the task and is identifiable as a single entity to the executive program.

A diagram of the states is shown in Figure 1.

The definition of task interrelationships in terms of virtual processors does not imply that a particular implementation must include the concept of a virtual processor or require a virtual storage executive routine, but the particular implementation must ensure that the result of an execution of a reference to the subroutine procedures given in Section 4, follow the definitions in that section.

A task can change the state of any other task except that a task can only be changed from Running to Voluntary Suspend by itself.

2.1 Task State Definitions

The four states of a task are Non-existent, Pending, Running, and Voluntary Suspension defined as follows:

2.1.1 Non-existent. The task's virtual processor does not exist and the real processor is unaware of the existence of the task. However, the executable program, whose execution would become the task, can and usually does exist and the executable programs existence is usually known to the processor.

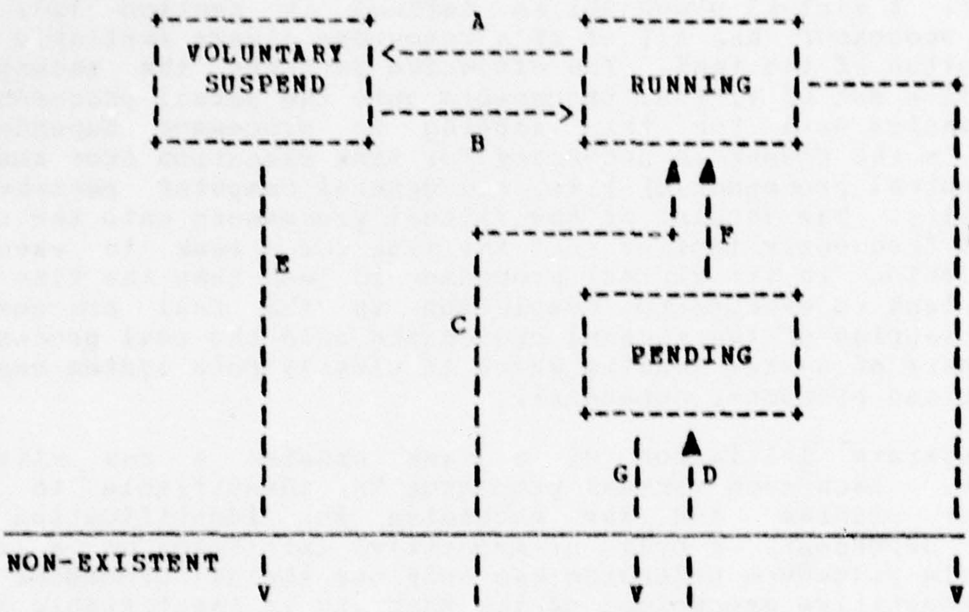


Figure 1

TASK STATE DIAGRAM

A) DELAY, WAIT, WAITE
B) occurrence of the event or completion of the time period
C) SKED, START, TRNON for immediate execution or CON where the event has already occurred
D) SKED, START, TRNON, CON, CYCLE
E) ABORT, STOP
F) occurrence of the event or time
G) DSKED, DCON, DCYCLE

2.1.2 Pending. The task's virtual processor exists and the task is awaiting the occurrence of an event to begin execution at which occurrence the task will move to the state Running. The real processor may or may not have loaded the task into its memory but the task is known and identifiable to the real processor. The mechanisms for this identification are processor dependent.

2.1.3 Running. The task's virtual processor exists and the task is running in its virtual processor. At any time the task may or may not be running in the real processor depending on the mapping procedure of the virtual processors onto the real processor. The mechanisms for this mapping are processor dependent.

2.1.4 Voluntary Suspension. The task's virtual processor exists and task is suspended in its virtual processor awaiting the occurrence of an event to resume execution at which occurrence the task will move to the state Running.

3. EVENTMARKS

In the management of independent interrelated tasks, there are numerous requirements to key actions to the occurrence of events. The mechanism chosen to enable such action is the eventmark.

Eventmarks are integer counters supported by the executive program which must be non-negative. An event is noted by incrementing a specific eventmark by one (1) which action is called "setting" the eventmark. The eventmark is decremented by one (1) by direct program control or by the execution of a task which was keyed to the occurrence of the event noted by the eventmark. This action is called "clearing" the eventmark. Eventmarks can only be changed by the procedure references defined in this standard and by the executive routine.

If an eventmark is defined as zero (0) in a procedure reference, no action will be taken and the execution routine will ignore any actions dependent on that eventmark.

3.1 Eventmark Handling

Eventmarks are properties of the processor which exist as only zero or positive integer values where zero may be considered as OFF and any positive value is considered ON. Eventmarks may be cumulative, in that they may count the number of "sets" thus requiring an equal number of "clears" before the condition will change. Once the condition has been achieved no further accumulation will occur; a single "set" will always establish an ON condition.

3.1.1 Setting an eventmark to the ON condition, SETEM

Execution of a reference to the subroutine SETEM shall cause the specified eventmark to be set to the ON condition. If the eventmark was already ON, the accumulation of such sets shall be incremented. Should the action of SETEM result in a change of state from OFF to ON, any programs pending or delayed for the occurrence of the event will enter their running state and the eventmark will be decremented. The form of the CALL is as follows:

CALL SETEM (e,m)

where:

e specifies the desire eventmark; an integer expression

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - request rejected

This argument shall be an integer variable or integer array element.

3.1.2 Clearing an eventmark, CLREM

Execution of a reference to the subroutine CLREM shall decrement by one the count of sets to the specified eventmark. If the eventmark was already at OFF, there will be no action. The form of the CALL is as follows:

CALL CLREM (e,m)

where:

e specifies the desire eventmark; an integer expression

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - request rejected

This argument shall be an integer variable or integer array element.

3.1.3 Testing the eventmark condition, TESTEM

Execution of a reference to the function TESTEM shall return a logical TRUE value if the specified eventmark was ON (set) and a logical FALSE value if the eventmark was OFF (clear). The condition of the eventmark shall not be affected. The form of this function reference shall be as follows:

TESTEM (e, m)

where:

e specifies the eventmark, an integer expression

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - request rejected

This argument shall be an integer variable or integer array element.

3.1.4 Presetting an eventmark count, PRESEM

Execution of a reference to the subroutine PRESEM will assign a declared value to the specified eventmark. Should the action of PRESEM result in a change of state from zero to some positive value, any programs pending or delayed for the occurrence of the event will enter their running state and the eventmark will be decremented. The form of the call is as follows:

CALL PRESEM (e, v, m)

where:

e specifies the eventmark; an integer expression

v specifies the desired preset value of the eventmark; an integer expression which must be zero or positive in value.

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - request rejected

This argument shall be an integer variable or an integer array element.

4. EXECUTIVE INTERFACE

4.1 Control of Task Execution

Executive interfaces provide the facility to control operation of the tasks within the system. Through these external procedures, one may start, stop, delay or synchronize the execution of tasks.

4.1.1 Exception Handling

The argument m, shown below in each of these executive interface procedures, shall be set equal to or greater than two (2) in value for all instances in which the request was not accepted by the executive routine. Individual implementations may specify unique values of m within the allowable range to designate the specific reason for which the request was rejected.

4.2 Starting a Task Immediately or After a Specified Time Delay or Upon an event occurrence, SKED

Execution of a reference to the subroutine SKED shall, after the expiration of the specified time delay or at the desired time of day or upon the occurrence of a specified event, cause the execution of the designated program. The program thus scheduled becomes known as a task. The actual time resolution obtainable in a specific industrial computer system is subject to the resolution of that system's real time clock. Execution of the designated task will commence at the program's first executable statement. The task is placed in the state PENDING, the task's virtual processor comes into existence and the task is known and identifiable to the real processor. The form of this procedure reference is:

CALL SKED (i, p, s, e^1, t^1, n, t^2, c, t^3, e^2, e^3, m) where:

i specifies the program to be executed.

The argument is either:

a) an integer expression, or

b) an integer array name, or

c) a procedure name

The processor shall define which of the above three forms is acceptable.

p specifies the task priority; an integer expression

s specifies the selection between event initiated execution and time dependent initiation; an integer expression evaluated as follows:

= 1 start immediately

= 2 start at time of day indicated by t^1

= 3 start when eventmark specified by e^1 becomes ON

= 4 start at time t^1 or when the eventmark specified by e^1, becomes ON, whichever is first

= 5 start after time period t^1 has expired

= 6 start after time period t^1 or when the eventmark specified by e^1 becomes ON, whichever is first

e^1 specifies the eventmark; an integer expression

t^1 designates an integer array name whose first eight (8) elements contain the time at which the specified program is to be executed.

These elements are as follows:

First element - Hours (0 to 23)

Second element - Minutes (0 to 59)

Third element - Seconds (0 to 59)

Fourth element - Milliseconds (0 to 1000)

Fifth element - basic units of the system real time clock.

Sixth element - Day (1-31)

Seventh element - Month (1-12)

Eighth element - Year from AD 0

n specifies a time phasing by which the starting time of the designated task would be deferred from the time stated in t^1 or the event specified by e^1. This variable must be an integer expression evaluated as follows:

= 1 no deferred start

= 2 start at the next occurrence of the time of day specified by t^2 after the time or event specified by t^1 or e^1 depending on s, occurs.

= 3 start after a period of time specified by t^2 from the occurrence of time t^1 or event e^1 depending on the value of s.

t^2 specifies the time of day or incremented time for the deferral of the start of a task as specified by n; this must be an integer array name as described in t^1

c specifies the cyclic execution of the task; an integer expression evaluated as follows:

= 1 no cyclic execution (run once only)

= 2 cyclic execution each time period as specified in t^3

= 3 cyclic execution each time the eventmark specified by e^2 becomes ON

t^3 specifies the time period for cyclic execution; this must be an integer array name as described in t^1

e^2 specifies the eventmark for cyclic execution; an integer expression.

e^3 specifies the eventmark to be turned ON upon a cyclic overrun condition; an integer expression. Note: Cyclic execution overrun can only be set by a cyclic SKED (or other) request with c = 2 or 3.

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - request rejected

This argument shall be an integer variable or integer array element.

4.2.1 Cause a program to cycle at a stated frequency, CYCLE (a shortened form of SKED)

Execution of a reference to the subroutine CYCLE shall the execution of the designated program immediately and cause successive executions to occur after specified periods of time. The cyclic period is defined as the period between successive initiations of the specified program. Resolution of time is as always subject to the limitations of the system's real time clock and other processing which may be occurring. If a cyclic overrun condition occurs, the action is processor dependent and no eventmark is set. The form of this procedure reference is:

CALL CYCLE (i, p, t^3, m)

where:

i specifies the program to be executed.

The argument is either:

a) an integer expression, or

b) an integer array name, or

c) a procedure name

The processor shall define which of the above three forms is acceptable.

p specifies the task priority; an integer expression. The definition and usage of this parameter is processor dependent.

t^3 specifies the cyclic time period; this must be an integer array name as described in t^1

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - request rejected

This argument shall be an integer variable or integer array element.

An execution of a reference to CYCLE would have the same effect as a reference to SKED with the following values of the remaining arguments:

s = 1

e^1 = 0

t^1 = (don't care)

n = 1

t^2 = (don't care)

c = 2

e^2 = 0

e^3 = 0

4.2.2 Cause a program to execute each time a stated eventmark becomes ON, CON

Execution of a reference to the subroutine CON shall cause the designated program to become associated with the specified eventmark such that the task shall perform one execution each time the stated eventmark changes from OFF to ON. The task shall begin from the first executable statement. If a cyclic overrun condition occurs, the action is processor dependent and no eventmark is set. The form of this procedure reference is:

CALL CON (i, p, e^1, m)

where:

i specifies the program to be executed.

The argument is either:

a) an integer expression, or

b) an integer array name, or

c) a procedure name

p specifies the task priority; an integer expression

e^1 specifies the eventmark; an integer expression

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - request rejected

This argument shall be an integer variable or integer array element.

An execution of a reference to CON should have the same effect as an execution of a reference to SKED with the following values of the remaining arguments:

s = 3

e^1 = 0

t^1 = (don't care)

n = 1

t^2 = (don't care)

c = 2

t^3 = (don't care)

e^3 = 0

4.3 Terminating a *Scheduled Task*, DSKED

Execution of a reference to the subroutine DSKED shall immediately disassociate the execution of the specified program with either a future time or an event or both. The task associated with the program will be removed from the state PENDING ,the task's virtual processor will become non-existent and the task will become unknown to the real processor. The form of the call is as follows:

```
CALL DSKED (i, j, n, t, m)
```

where:

i specifies the program name to be descheduled.

The argument is either:

a) an integer expression, or

b) an integer array name, or

c) a procedure name

The processor shall define which of the above three forms is acceptable.

j specifies which of several possible pending programs identified by the i parameter shall be descheduled; an integer expression;

= 1 deschedule the next pending execution only; allow others to remain scheduled (applies only to time scheduled programs)

= 2 deschedule all future executions of the program (applies to both time and event scheduled programs)

= 3 deschedule program i then currently pending after waiting for a time period, t, to pass.

n specifies the choice of the content of t as follows:

= 1 deschedule program i after the time of day specified by t occurs.

= 2 deschedule program i after the time period specified by t has elapsed.

t designates an integer array name whose first eight (8) elements contain the time at which the specified program is to be descheduled.

These elements are as follows:

First element - Hours (0 to 23)

Second element - Minutes (0 to 59)

Third element - Seconds (0 to 59)

Fourth element - Milliseconds (0 to 1000)

Fifth element - basic units of the system real time clock

Sixth element - Day (1-31)

Seventh element - Month (1-12)

Eighth element - Year from AD 0

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - request not accepted

This argument shall be an integer variable or integer array element.

4.3.1 Cancelling a task scheduled for future execution, CANCEL

Execution of a reference to the subroutine CANCEL shall immediately remove the specified task from the time schedule of pending executions. The form of the call is as follows:

```
CALL CANCEL (i,m)
```

where:

i specifies the task to be removed from the time schedule. The argument is either:

a) an integer expression, or

b) an integer array name, or

c) a procedure name

The processor shall define which of the above three forms is acceptable.

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - delay as specified has not occurred

This argument shall be an integer variable or integer array element.

Execution of a reference to CANCEL shall have the same effect as an execution of a reference to DSKED with the following values of the other arguments:

j = 1

n = 2

t = 0 for all elements

4.3.2 Removing a task from cyclic scheduling, DCYCLE

Execution of a reference to the subroutine DCYCLE shall immediately remove the specified task from the list of cyclic time based executions but allows the execution currently in the RUNNING State, if any, to be completed. The form of the call is as follows:

CALL DCYCLE (i,m)

where:

i specifies the task to be removed from cyclic scheduling. The argument is either:

a) an integer expression, or

b) an integer array, or

c) a procedure name

The processor shall specify which of the above three forms is acceptable

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - request rejected

This argument shall be an integer variable or integer array element.

Execution of a reference to DCYCLE shall have the same effect as an execution of a reference to DSKED with the following values of the other arguments:

j = 3

n = 2

t = 0 for all elements

4.3.3 Removal of a task from time or event based schedule, DCON

Execution of a reference to the subroutine DCON shall immediately remove the specified task from all future scheduled executions and/or association with any event occurrence. The form of the call is as follows:

CALL DCON (i, m)

where:

i specifies the task name to be descheduled.

The argument is either:

a) an integer expression, or

b) an integer array name, or

c) a procedure name

The processor shall define which of the above three forms is acceptable.

The value must be 1 or greater

1 - request accepted

2 or greater - request not accepted

This argument shall be an integer variable or integer array element.

Execution of a reference to DCON shall have the same effect as an execution of a reference to DSKED with the following values of the other arguments:

j = 2

n = 2

t = 0 for all elements

4.4 Delay Execution of the Program, DELAY

Execution of a reference to the subroutine DELAY will cause suspension of the program execution until the specified event occurs or the specified time of day or time period has elapsed. Whichever occurrence happens first will cause the program to resume execution at the statement following the call. If the program was DELAYed for an event to occur, the value of the eventmark will be automatically cleared prior to the resumption of execution. The task will be moved from the state RUNNING to the State VOLUNTARY SUSPEND. A task may only delay itself and cannot delay another task or be delayed by another task. The format of the call is:

CALL DELAY (s, t', e, m)

where:

s specifies the selection between event initiated end-of-delay and time dependent initiation; an integer expression evaluated as follows:

1 - time increment specified by t' ends delay

2 - time of day specified by t'

3 - event specified by e ends delay

4 - whichever time increment or event occurs first ends delay

5 - whichever time of day or event occurs first ends delay

m is set on return to the calling program to indicate the disposition of the request as follows:

t¹ designates an integer array name whose first eight (8) elements contain the time at which the specified program is to be executed. These elements are as follows:

First element - Hours (0 to 23)

Second element - Minutes (0 to 59)

Third element - Seconds (0 to 59)

Fourth element - Milliseconds (0 to 1000)

Fifth element - Basic units of the system real time clock Sixth element - Day (1-31)

Seventh element - Month (1-12)

Eighth element - Year from AD 0

e specifies the eventmark; an integer expression

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - requested rejected

This argument shall be an integer variable or integer array element.

4.4.1 Suspend execution of a program for event occurrence, WAITE

Execution of a reference to the subroutine WAITE will cause suspension of the program execution until the specified event occurs. Following the occurrence of the event, the program will resume execution at the statement following the call with the value of the eventmark automatically cleared. The format of the call is:

```
CALL WAITE (e,m)
```

where:

e specifies the eventmark; an integer expression

m is set on return to the calling program to indicate the disposition of the request as follows:

1 - request accepted

2 or greater, request rejected

This argument shall be an integer variable or integer array element name.

Execution of a reference to WAITE shall have the same effect as an execution of a reference to DELAY with the value of S=3 and no particular value of t[1].

4.5 Abort a Program, ABORT

Execution of a reference to the ABORT subroutine will cause the named program to immediately terminate execution and to disassociate itself from all forms of scheduled activity whether based upon event occurrence or time. The task is removed from the state RUNNING or VOLUNTARY SUSPEND. The tasks virtual processor becomes non-existent and the task becomes unknown to the real processor. The form of the call is:

CALL ABORT (i,m)

where:

i specifies the program to be aborted.

The argument is either:

a) an integer expression, or

b) an integer array name, or

c) a procedure name

The processor shall define which of the above three forms is acceptable

m is set on return to the calling program to indicate the disposition of the request as follows:

The value must be 1 or greater

1 - request accepted

2 or greater - request rejected

This argument shall be an integer variable or integer array element.

APPENDIX A

This appendix is not part of the ISA Standard S61.2 but is included to facilitate its use.

Considerations leading to Inudstrial Computer System FORTRAN Procedures for File Access.

A.1 Historical Development

This standard is a direct outgrowth of the International Purdue Workshop on Industrial Computer Systems whose goals are:

To make the definition, justification, hardware and software design, procurement, programming, installation, commissioning, operation, and maintenance of industrial computer system more efficient and economical through the development of standards and/or guidelines on an international basis.

The Workshop formed several committees to achieve its objectives. The FORTRAN Language Committee was charged with the task of preparing a set of Industrial Process standards compatible with standard FORTRAN.

This standard is the result of that committee's work.

A.2 Criteria Used in Developing FORTRAN Standards

The committee assessed the status of FORTRAN as used in the industrial environment and followed the guidelines below in the devlopment of the standards:

(1) The standards whould cover features commonly used by existing industrial computer systems.

(2) The standards should be easily implemented by most vendors.

(3) The standards should follow the syntax and intent of FORTRAN as defined by ISO R1539-1972.

(4) The standards should not restrict the future evolution of FORTRAN language.

The development of FORTRAN language standards is presently the responsibility of ANSI/X3J3. In order that ISA standards comply with the ANSI standards as far as possible, external-procedure references were used rather than direct changes or additions to the syntax of FORTRAN. This does not imply that this is the only way to provide these features, nor does it exclude the possibility or desirability that ANSI will develop the language syntax to perform these and other related forms.

APPENDIX B

This Appendix is not part of the ISA Standard S61.3 but is included to facililatate its use.

NOTES BY SECTION

B.1 Section 3 Notes

The eventmark defined in this section provides capability for handling events. The use of events is the responsibility of the system designer but two (2) common usuages can be envisaged.

An event can be considered to be an interrupt, either hardware or software, which exist in most industrial computer systems. The occurrence of an interrupt can cause the executive routine to set an eventmark and the procedure references defined in Sections 3 and 4 of this standard provide all the necessary features to control computer usage based on interrupts.

An event can also considered to be a Dijkstra P-V semaphore and can be used for the control of critical resources and the resolution of contention for such resources. The well-known one-writer-many-readers problem can be solved by using the eventmarks as semaphores.

In the use of eventmarks it is the system designers responsibility to ensure that no lock-out conditions occur. There is no provision in the standard to ensure that inadvisable use of the eventmarks will not cause undesirable consequences.

There is no provision in the standard for BOOLEAN operations on eventmarks. Such features may be provided by the executive program or be available at system generation time and would be processor dependent. The lack of such features is not intended to discourage their inclusion by system suppliers but it is considered premature to standardize on such features at this time.

B.2 Section 4 Notes

This standard is a permissive standard in that it does not prescribe how the executive will respond to external procedure references nor does it describe how the information is passed to the executive routine. In particular, the argument "i" in CALL SKED (4.2), CALL DSKED (4.3) and CALL ABORT (4.4) has three forms, an integer expression, an integer array name or a program name, only one of which is permissible in any particular program. This restriction is a necessary consequence of the requirements of the FORTRAN language that any argument which is an external procedure reference be of a defined type; integer expressions, integer array names and program names are different types (Section 8.4.2 of ISO-R1539-1977). It is also a consequence of the FORTRAN language that if the argument is a program name, this name must appear in an EXTERNAL statement (Section 7.2.15 of ISO-R1539-1972).

Examples of the use of the argument i as an integer expression are:

```
--CALL ABORT (7,M)

--DATA J/7/
  CALL ABORT (J,M)

--DATA J/2HAB/
  CALL ABORT (J,M)
```

An example of the use of the argument i as an integer array name is:

```
INTEGER XYZ
DIMENSION XYZ(3)
DATA XYZ(1),XYZ(2),XYZ(3)/2HAB,2HCD,2HEF/
CALL ABORT (XYZ,M)
```

An example of the use of the argument i as a procedure name is:

```
EXTERNAL ABCD
CALL ABORT (ABCD,M)
```

Interchangeability of programs between different processors is reduced by permitting three possible types for this argument but the committee feels it is premature to standardize to achieve full interchangeability in this area.

E.3 Complexity of Section 4 calls

The calls for SKED and DSKED are very complex but are provided to enable the user to specify in one atomic function everything required for either a scheduling or descheduling action. Obviously, not every call needs to be this complex. Therefore, the simplier forms CYCLE, CON, DCYCLE, DCON and CANCEL are included as subsets of SKED and DSKED. These are proper subsets with the standard providing information upon the "default" properties of the unspecified arguments.

It should also be noted that several of the procedures of ISA S61.1-1976 and the FORTRAN statement STOP are also subsets of SKED, DSKED and DELAY. The correspondence is shown in the following table:

S61.1	S61.3
CALL START	CALL SKED
i	i
j	t^1
k	s=5
m	m
-	p
-	e^1
-	n
-	t^2
-	c
-	t^3
-	e^2
-	e^3
CALL TRNON	CALL SKED
i	i
j	t^1
m	m
-	s=2
-	p
-	e^1
-	n
-	t^2
-	c
-	t^3
-	e^2
-	e^3
CALL WAIT	CALL DELAY
j	t^1
k	s=1
m	m
-	e
STOP	CALL ABORT
	i= name of current program
	m

NEW SUBROUTINES PROPOSED FOR RT-FORTRAN

Kazimierz Maliszewski

Kazimierz Maliszewski
MERA-PIAP, WARSZAWA

September 22, 1977

NEW SUBROUTINES PROPOSED FOR RT-FORTRAN

The present working paper written for the RT-FORTRAN Committee of Purdue Europe contains description of several new RT-FORTRAN references.

1. New definition of ATIME.

In current form the subroutine ATIME gives time of day as integer value expressed in seconds. As 24 hours = 86400 sec., the maximal value would exceed capacity of 16 bit word, frequently used in industrial installations. Besides, 1 second unit seems too large for precise measurements.

I propose two alternative versions of ATIME.

Version 1.

Obtain time of day expressed in desired units.

The form of the call is:

CALL ATIME(a, k)

where:

a designates a variable or array element to receive the time of day. The result shall be given in floating-point form, including fractional part. The type of this argument is either:

a/ real

or b/ double precision.

The processor shall define which form is acceptable as well as the accuracy of result.

k specifies the units in which the time should be placed in "a" as follows:

0 - basic counts of the system's real time clock

1 - milliseconds

2 - seconds

3 - minutes

This argument shall be an integer expression.

Version 2.

Obtain the time of day in seconds.

The form of the call is:

```
CALL ATIME(a)
```

where a is defined as in Version 1 with the following addition: The time since the beginning of day shall be expressed in seconds with maximum accuracy available in the system.

Comments:

1/ A real value is more suitable for computations than an integer one.

2/ If real value is stored on 48 bits, the result has 11-digit accuracy. This is enough to express on day in milliseconds /86,400,000/. In systems with smaller accuracy the double precision can be used.

3/ Real Time FORTRAN.

4/ I prefer the Version 2.

2. Connection of an event to a semaphore.

I submit for discussion a subroutine called SECON. It connects a specified event to a specified semaphore. This connection lasts till the event occurs or the calling activity ends its execution. The executive system will not accept a demand of connection if the specified event is already connected to a semaphore. The occurrence of the event causes an action equivalent to that of the SIGNAL. An activity can await the event by calling the WAITS. Thus the subroutine AWAIT would not be necessary.

The event can be accompanied by a transmission of data, so the implementor may restrict using the SECON to core-resident programs.

The form of the call is:

```
CALL SECON (r, j, k, m)
```

where:

r is an integer expression specifying the semaphore.

j specifies the event to be connected. This argument shall be an integer array name or an integer array element.

k specifies a space in the calling program for input/output data if the event signifies end of transmission. The use /or not/ of this argument depends on the nature of the event. The argument shall be either:

a/ an integer variable or an integer array element

or b/ an integer array name

or c/ an integer expression /conditionally that data are not input/

m is an integer variable or an integer array element which is set by the executive system to indicate the status of the call:

$m \leq 0$ - not defined.

$m = 1$ - the specified event has occurred, the transmission to/from "k" /if any/ has been finished and the semaphore has been increased by 1. If new value of the semaphore equals 1, an opportunity is given to an activity waiting on the semaphore to finish its suspension.

$m = 2$ - request accepted, the specified event has not yet occurred.

$m \geq 3$ - error conditions. The implementor may specify a number of error indicating values and state for every of them if it is accompanied by an incrementation of the semaphore, or not.

Comment

A separate call for event description makes it possible to wait for a logical sum of several events. If so, the waiting activity should search what has happened after every return to state RUNNING. That is achieved by testing status variables "m" used in calls for SECON.

The note concerning two kinds of error /$m \geq 3$/ means that errors in data transmission should increase the semaphore, so that the activity may not wait endlessly. However if the system decides to reject the call immediately after detection, the semaphore will not be touched.

Actually SECON can be used for initiation of a transmission. A simplified flow diagram for a section of an activity awaiting one of three inputs is presented below.

3. Error handling in RT-FORTRAN.

In the current form of RT-FORTRAN a programmer must check error parameter /m/ after each call to be sure that all is right. I propose a subroutine to make this work not necessary. The subroutine is called COSER /Common Service of Errors/. The form of the call is:

```
{ CALL COSER (m, i, j)          }
{ Statement inspecting m.       }
```

Arguments:

m is an integer variable or integer array element. RT-FORTRAN references which will subsequently use m as status parameter shall be handled in case of error in a way described further.

i is an integer expression having a value ≥ 2. Error handling shall take place when m assumes a value greater than or equal to "i" on return from a subroutine.

j is an integer array or integer array element name. The error handling system shall place there all information necessary to identify the error.

Principle of work:

1/ Before the subroutine COSER is invoked in an activity, the present error handling is not undertaken for the activity.

2/ When the COSER is called, it remembers addresses of "m" and "j" as well as the value of "i". The address of the next statement after CALL COSER is also stored.

An error in COSER's parameters shall be signalled in a system-dependent way.

3/ After acceptance of actual arguments a return from COSER is made with m=1.

4/ The value of m should be checked in calling activity. Logically this check should be done immediately after CALL COSER. Nevertheless no constraint exists whether, where and how to do a check.

The check should detect values m ⩾ i signalling error conditions. This can be done with a statement:

```
IF (m.GE.i) GO TO e
```

where "e" is a label in the calling program.

Of course, initially m=1, i ⩾ 2, so the jump is skipped and the activity proceeds farther to do its normal job.

5/ Only RT-FORTRAN subroutines using as status parameter a variable previously specified as "m" in CALL COSER are concerned. When return is made from such a subroutine with m ⩾ i, <u>the next executed statement is one which follows CALL COSER with this "m"</u> /and not the statement following subroutine reference/. m has then an appropriate error indicating value, greater than or equal to 2. Corresponding array "j" contains more information about the error. This information shall have a system-dependent form. At least an identifier of the subroutine which failed is necessary, I suppose.

6/ After a check described in item 4 the activity proceeds to its own error handling /to label "e" in the example above/.

7/ Particular processors may declare how much error variables "m" can be declared by CALL COSER in an activity. Nevertheless it should be possible to specify a number of pairs: m, i. This is essential as not always i=2. Each CALL COSER should be followed by a check of its "m".

On the other hand repeated calling COSER with the same "m" shall introduce new values resulting from i, j and program location.

Example

Here is a fragment of a program using the COSER to detect an error in references to START, WAIT /m ≥ 2/ and CANCL, KILL /m ≥ 3/.

```
.....
C  DECLARE ARRAYS FOR COSER AND KILL
      DIMENSION J(5), K(5)
C  DECLARE VARIABLE M FOR ERROR VALUES .GE. 2
      CALL COSER (M,2,J)
      IF (M.GE.2) GO TO 1
C  DECLARE VARIABLE IN FOR ERROR VALUES .GE. 3
      CALL COSER (N,3,J)
      IF (N.GE.3) GO TO 2
..... /program body/
```

```
C  REFERENCES WITH M
      CALL START (ACT1,0,0,M)
      CALL WAIT (1,2,M)
.....
C  REFERENCES WITH N
      CALL KILL (ACT1,K,N)
      CALL CANCL (ACT2, N)
.....
C  REFERENCE NOT USING COSER
      CALL START (ACT3,0,0,M1)
      IF (M1.NE.1) STOP3
.....
1     error handling for START of ACT1 or WAIT
.....
2     error handling for CANCL or KILL
```

4. <u>Continuation of execution broken by an error</u>.

An activity may resume as execution broken by an error service resulting from use of the COSER. This can be done with a GOTO or IF statement. Sometimes however this may be impossible or inconvenient. A subroutine for this purpose is proposed. The form of the call is:

CALL GOBACK

with no arguments.

The activity shall continue its execution from a statement following the last RT-FORTRAN reference which have caused a break of executing sequence due to COSER's action.

If the GOBACK is called without previous error service originated by the COSER, the result is not defined /unless defined by the processor/.

Acknowledgment

I wish to thank Mr. Andrzej Aderek from MERA-PIAP, Warsaw. I owe him many fruitful discussions and in particular the concept of ATIME.

APPENDIX X

PAPERS OF THE MAN/MACHINE COMMUNICATIONS COMMITTEE

Documents included here are as follows:

1. Messare, R.R., Larocke, J. H., and Shannon, D., "Digital Control Technology and Person Research Interface R&D. Tentative Functional Requirements and Design Criteria for Operator Interfaces". Exxon Research and Eng. Co., Exxon Chemical, U.S.A., and Imperial Oil Enterprises, Ltd.
2. Automation and Man-Machine Communications - Aims and Activities of TC-6, June 1977.

DIGITAL CONTROL TECHNOLOGY

PERSON-MACHINE INTERFACE R&D

TENTATIVE FUNCTIONAL REQUIREMENTS AND DESIGN CRITERIA FOR OPERATOR INTERFACES

BY: R. R. MESSARE - Exxon Research & Eng. Co.
J. H. LAROCHE - Exxon Chemicals, USA
D. SHANNON - Imperial Oil Enterprises, Ltd.

TABLE OF CONTENTS

PRECEDING PAGE BLANK

TABLE OF CONTENTS
(CONT'D)

Exxon Engineering — Technology Department

EXXON RESEARCH AND ENGINEERING COMPANY

DIGITAL CONTROL TECHNOLOGY R&D

TENTATIVE FUNCTIONAL REQUIREMENTS AND DESIGN GUIDELINES FOR OPERATOR INTERFACES

INTRODUCTION

The document describes the basic characteristics of current operator interfaces used in Exxon plants and defines the functional requirements and design criteria for future, centralized, multiscreen consoles, intended to be used as the sole operator interface with distributed and/or hierarchical control systems for process control of Exxon's refineries and chemical plants.

The purpose of this document is multifold and covers the following areas:

- Explain and document the shortcomings of current systems
- Make instrumentation and computer vendors aware of our requirements for future operator interfaces.
- Establish a constructive dialogue with interested vendors actively involved in developmental effort in the area of PMIF.
- Establish the feasibility and cost impact for implementing designs to satisfy these requirements and develop the basis for a stepwise approach.

PART 1

CURRENT OPERATOR INTERFACES FOR COMPUTER BASED CONTROL SYSTEMS

GENERAL

- Current operator interfaces consist of computer consoles and conventional control panels ("back-up" panel).

- Computer consoles are driven by the process control computers and can access and display all process inputs connected to the computer as well as computer generated variables.

- Conventional panels can only display process input signals connected directly to the "back-up" instrumentation and data acquisition systems.

  Usually not all the process variables displayed at or controlled from the panel are accessible from the computer console. This is normally dictated by lack of incentives to associated certain functions with computer control and/or to duplicate all control functions of the panel at the computer console.

  Control or display functions which are usually not available at the computer console include the following:

  - Dedicated sequence controllers
  - On-off controls (pumps, MOV's, mixers, etc.) for onsites
  - Emergency manual shutdown of major process equipment
  - Certain process alarms (normally associated with the status of utilities)

- The panel mounted instrumentation (displays and control stations) is not only used during computer outages but also when computer control is available. In the latter case, the panel instrumentation is mainly used for:

  - Handling process upsets
  - Plant or unit(s) start-up (e.g., after turnarounds)
  - Emergency shutdown

A summary of the main reasons for the use of the "back-up" panel, when the computer (and computer console) are available, is given below:

- Handling Process Upsets

  - Procedures to call display(s), select loop(s) and take corrective action is slow and inadequate to handle most emergencies.
  - No computer console has sufficient overview capability to allow surveillance during corrective action.

- Plant Or Unit(s) Startup
  - Console operator alone cannot handle workload from console during start-up (senior field operators are also assisting using the panel).
  - Console overview capability very limited.
- Emergency Shutdown
  - Combination of the above.

BASIC CHARACTERISTICS OF CURRENT MMIF

Console Operator

- Position In Organization

  Normally (but not always) is a first line supervisor, directing the field operators via two way radio.

- Selection Criteria

- Very good knowledge of process units assigned to his console.
- Good knowledge of process units assigned to other consoles.
- Good supervisory qualities (more stressed in certain plants than others).

- Available Pool

- Senior field operators
- Senior field operators can replace console operators during scheduled and/or unscheduled absence.

Number of Computer Consoles/Control Room

- Usually 2 or 3 consoles - Number, however, can be as high as 5.
- In highly centralized plants, several control rooms may exist in the same control house.
- There may be an engineer's console in the same control room. This console includes all the features of the operator console and can be used by the operators if required.

Characteristics of Consoles

- Normally of the CRT type - (15" diag. black & white or 19" color).
- Most consoles have two CRT screens but number varies from 1 to 3.

- Being supported by the same computer, all consoles can access the entire data base.
- There is one operator assigned per console. However, consoles with multiple screens and keyboards can be used by more than one operator if required.
- In some cases, all pen recorders are mounted on a credenza, installed close to the computer console.

OPERATOR RESPONSIBILITIES AND WORKLOADS

Number of Control Loops/Console

- Varies considerably (from 70 to 250)
- Number depends on:
  - Nature of process
  - Complexity
  - Degree of process integration

Note that since the definition of "control loops" has become rather loose, the above figures are only indicative and should be rather interpreted as number of controlled outputs.

Number of high level (4-20 ma) inputs/console

- Varies from 1.6 to 2 times the number of "control loops"

Number of low level inputs (usually TC inputs)

- Can be as high as twice the number of high level inputs.
- DTI readout is usually installed both at the console and panel.

Number of alarm windows (panel annunciators)

- Varies considerably but on the average is approximately 1.7 times the number of control loops. A ratio, however, of $2 \div 1$ is not unusual. More than 90% of these alarms can be displayed at the computer console. (CRT and/or printer).

Software Generated Alarms

- Additional alarms (computer generated) can only be displayed on console CRT or typed via printer. Again their number varies considerably but in general they represent the major bulk of alarms

Operator responsibilities

- It should be noted that although there is one console operator assigned per console, other console operators or senior field operators may be asked to assist if the situation demands.

As mentioned before a console operator is required to know how to handle process units not normally "assigned" to his console. This is translated into the following requirements, generally met by existing consoles.

- Ability to take under control from his own console, process units assigned to other consoles, installed in the same control room.

- Ability to monitor status of all alarms from his own console.

- Despite extensive use of control computers for logging different process data and selected conditions, operators still maintain hand written logs. This is mainly done in order to retain access to a certain set of data when the computer memory banks are wiped out by a system malfunction.

MAIN DISPLAY CHARACTERISTICS

Pen Recorders

- The number of variables simultaneously recorded on pen recorders (both shared and dedicated) varies from 12% to 42% of all high level inputs. The average is approximately 30%.

- Operator feedback indicates that above % can be reduced to 20% (rule of thumb).

- Almost all console operators use pen recorders only for monitoring general trend of variable.

- Changing pen assignment on shared recorders becomes very unusual after few months of operation. Shared recorders have normalized scales, since each pen can be assigned to several process variables.

Indicators

Next to the recorders is the most easily scanned display. They are used during unit surveillence and during verification of alarm conditions as part of the panel "overview" capability.

Control Stations (Back-up)

- Indicating lights are displaying the status of the back-up instrumentation (absence of light indicates that loop is under computer control)

- The type of back-up station (C/M/A-1, C/M/A-2, C/M/A-T, C/M/A-S) is usually indicated (via "dynotaped" labels) when several back-up types are used.
- Color coding is widely used to identify:
  - Loop function (flow level, etc.)
  - Process unit or process area
- Control stations are mounted in high density arrays. Installation density of 10 rows by 10 columns per panel section is widely used in all recent installations, although actual occupancy rarely exceeds 60%. The density, expressed in terms of loops/linear ft of panel varies from 7 to 11 loops/linear feet (includes spare space, recorders etc.). The higher densities are achieved with Foxboro minaturized displays.

Alarms Annunciators

- Dimensions of windows vary from full size down to 1 1/2" x 3/4"
- Identification bv console operators sitting at console is via pattern recognition.
- Shrill audible alarm signal irritate operators the most. In some instances entire sections of annunciators are permanently silenced
- Different priorities of alarms are indicated by means of color coding (colored celophane placed behind alarm window) and/or different audible signals.
- Extremely useful is the use of different audible signals for different console areas (within the same control room).

Graphic presentation of process

- Semi-graphic panels are not used anymore
- Only OM&S CRT consoles have extensive library of graphics (interactive)
- With few exceptions (mainly experienced operators), onsites operators miss the graphic presentation. (In all plants visited-except one- P&ID's were attached with cellotape on spare panel spaces).

CRT consoles

- The most widely used display format is the multiloop format.

- The following information is normally displayed with multiloop formats
  - Tag I.D. (or loop I.D.)
  - Tag description
  - Current value of variable (measured or calculated)
  - Set point (or target)
  - Tag status (control mode)
  - Valve position (control output)
- The number of loops (or tags) displayed on multiloop formats varies from 10 to a max. of 32. Usually is less than 16.
- Interactive graphic displays of process units or areas are only used in OM&S applications. For onsites applications, graphic capabilities of computer consoles is not used (effort to built displays appears disproportionate compared to usefulness).
- CRT trending capability is hardly used because
  - Procedure to call-up (or built) display is time consuming
  - Trending starts without previous history
  - It is much easier to scan the pen recorders
- Bar graph presentation of process variables is very useful when variables are logically associated and form distinguished patterns (e.g. temperature profiles, flow balancing in multipass furnaces, etc.)

  The majority of operators would prefer this type of presentation also for multiloop formats.

<u>Console alarming features</u>

- Worst presentation of alarm messages is via printer
- CRT alarming is prefered by operators over printed messages
- Only highest priority alarms are automatically displayed on CRT screen (no operator search). Usually, separate screens or dedicated portion of screen is used for alarm messages.
- Usually pattern recognition is not maintained with CRT alarm displays (i.e., alarm messages do not appear on predetermined screen areas but as they occur in a push-up or push-down stack configuration.) Pattern recognition, however, is sometimes maintained on alarm keyboards (normally separate from main operating keyboard).

AD-A058 096 PURDUE UNIV LAFAYETTE IND PURDUE LAB FOR APPLIED IND--ETC F/G 9/2
ANNUAL MEETING OF INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL C--ETC(U)
1977 N00014-78-C-0127

UNCLASSIFIED NL

4 OF 4

AD
A058 096

END
DATE
FILMED
10-78
DDC

Other CRT console display formats

A variety of other CRT display formats are being used such as

- Loop interconnection diagrams (block diagrams of multicascades)
- Logging of significant changes initiated by operators (inhibited alarms, changes of alarm settings etc.).
- Logs of process variables (measured or calculated).
- Plots
- Daily schedule of tasks (usually for OM&S operation)

Reports

A variety of reports can be obtained via printer (Historical data, hourly averages, daily averages, daily reports, monthly averages, etc.).

HUMAN OPERATING MODES

In general, the operator functions in the following basic modes:

a) Process Surveillance

During this mode, operators monitor the overall status of the process units assigned to them using operation by exception techniques. They normally function in this mode when the units are lined-up and on target.

Operators normally monitor:

- trend of key variables (glancing at pen recorders)
- status of variables associated with acknowledged alarms (there are always several alarms in acknowledged condition)
- status of controlled variables that control is placed on manual.

The exceptions could be:

- New alarm conditions (but low priority alarms are normally ignored)
- Wide deviations of process variables from set points
- Computer generated messages
- Sudden change of trends

During this mode of operation, they glance at the panel because:

- of its overview capability (no need for search, such as paging through CRT displays)

- it is there (habit)

Most operators prefer to anticipate disturbances that can result in tripping critical alarms. They don't want, however, to be continuously reminded of the same abnormal condition (as it happens when the same alarm message keeps repeating itself even when it has been acknowledged).

b) Unit Monitoring

During this mode, they take a "closer look" at individual process units. They normally enter this mode as a result of a noticed exception, at the recovery stage of an upset or when they do expect some problems as a result of changing conditions in other process units.

During this mode they usually employ "pattern recognition" techniques such as:

- Bar graph presentation of related variables (when available on CRT)

- Trend of key variables of that unit and relative position of trends

In addition they pay closer attention to digital values of key variables (usually one or two multiloop displays).

Computer consoles appear adequate for this type of operation (except for trend monitoring) but improvements can be made in the following areas:

- Data presentation for multiloop displays (searching for off normal patterns in terms of deviation from targets or set points is difficult when data is presented only digitally).

- Passing from one display to another. Preferred method is by dedicated keys (single action). Other methods that definitely assist operators are:

  + Ability to easily address displays associated with the one under current observation and return to previous display. (These displays may not be always in numerical sequence and "paging" is not sufficient.)

  + Ability to page through multiloop displays (forwards and backwards) is useful when the tags-logically associated with a major piece of equipment or parallel trains-exceed the display capacity of a single multiloop format

During this mode, operators do glance at the panel (normally the section grouping the panel instruments of the unit being monitored). However, if this section is not easily visible from the console-or not visible at all-very few operators walk to the panel.

c) Interactive Mode

In this mode the operator interacts with the process which is changing status in a rather prederternined way. Operator enter in this mode usually under:

- Planned startups or shutdowns
- Beginning or end of a process cycle
- Planned changes in process conditions (e.g., feed changes, target changes, etc.).
- Minor disturbances associated with a major equipment group in the process unit (such as a tower, reactor, furnace, etc.)

During this mode of operation, set point and valve position adjustments as well as loop mode changes are commonly made.

Except for startups or shutdowns, operators remain usually at the console (provided that console allows valve position adjustments in manual mode). Computer control strategies greatly assist operators in most of the above listed cases, reducing the amount of human intervention.

As far as major startups are concerned, no one polled has ever attempted one from a computer console. Start-up is usually carried-out from the back-up instrumentation (mostly on manual). Control is switched to computer when the process is basically within operating limits.

In order to handle startups from a CRT type console, the desireable enhancements briefly mentioned for other operating modes, become now a must. In summary they are:

- Increased overview capability (at least same as that provided by the conventional panel)
- Multiple keyboards and screens to be used by more than one operator.
- Improved display accessibility in multilevel hierarchical display configurations
- Easy association of displays at same hierarchical level (multi-loop level).

d) Emergency

This can be considered as a special case of the "interactive" mode in which one or several process units have been placed in an unexpected upset situation, usually brought about by failure of a major piece of process equipment, by activation of a safety shutdown system or by partial failure of the control instrumentation.

Recovery from a major upset condition is the most critical mode of operation. The steps to recover from such an upset cannot be anticipated in advance except in those cases were the cause of the upset is a predefined possible event (such as the activation of a safety shutdown system tripping a major piece of equipment) and the effects can be predicted to a large extent. Other upsets, caused for example by a tube rupture in a furnace or loss of control signals to several valves must be handled as the situation demands.

During this operating mode, several loops are usually placed in a manual status and the operator's principal attention is focused on the current and near future course of plant operation with the aim of:

- protect the equipment from being damaged
- recover from the upset situation
- prevent the effects of the current upset to propagate to other process units or group of equipment.

As mentioned before, computer consoles are generally inadequate to handle most process upsets. No operator has ever attempted to handle a major upset from the console.

An analysis of "predicted" upsets in 9 different plants indicated that:

- the number of loops to be manipulated in fast sequence (worst cases) varies from 2 to 15. In most cases is around 4-6.
- the time available for corrective action before the upset propagates to other units, varies from 10-15 sec. to 10-20 min.
- the minimum number of trends to be displayed simultaneously ("selective" overview) to assess the impact of corrective action and/or status of key variables of other process units appears to be 20.

It is generally felt that handling upsets from consoles not only can be made feasible but can also offer certain improvements over the current use of existing "combination" interfaces (computer console and back-up panel).

The basic requirements appear to be:

- The procedures to call required displays should be as fast and simple as possible (single action).
- Accessibility to the loop level (loop selection and modification of its status or manipulation of the accessible parameters) should be at least equivalent to the degree of accessibility that operators have from conventional panels.
- "Associated" displays (trends, multiloop, etc.) should be called-up quickly for selective overview at the same or different hierarchical display level from which corrective action is being taken. Selective overview should not substitute higher levels of overview which are intended to provide the operator with a "bird's eye" view of process operation.

The consideration of "special" features, such as specially built multiloop displays with common or preset action may also assist the operator in handling "predictable" upsets.

The use of special software programs to diagnose abnormal process conditions may also assist in handling certain categories of upsets. The definition of such packages, however, is outside the scope of this document.

It should be noted that in general, the course of upsets is by large unpredictable and reliance on automation or predetermined action is by no means considered sufficient. The basic design requirements therefore, should remain focused on the human operator so that he can easily assess the situation, and intervene at any time and with a degree of freedom at least equivalent to that offered by both levels of current interfaces (panel and console).

PART I

APPENDIX

DEFINITIONS

Plant : A group of process units, integrated together with various degrees of interaction. A plant includes the manufacturing facilities for intermediate or finished products, utilities for its operation and facilities for receiving and delivering feed stock or finished products.

Onsites : The part of the plant that is manufacturing basic unblended components or finished products from natural feedstock (crude oil, coal, brine, etc.) or preprocessed chemical feeds (naphtha etc.) Blending of produced components may or may not be included.

Utilities: The part of the plant that generates and distributes basic utilities required for the process equipment. It includes steam, compressed air (instrument and plant), water (cooling, fresh) electricity (high, medium and low voltages, including instrument power supply systems) as well as other utilities such as inert gas.

OM&S : Oil movements and storage is the part of the plant that usually contains the equipment for storing finished products or basic components for blending. Blending may or may not be part of OM&S.

It also includes facilities for receiving or shipping products (such as truck loading, car loading, ship loading etc.) Most shipments are under "custody transfer."

Control House : A building, normally of blastproof construction, where centralized control of several units is carried-out. The control house includes appropriate space for the installation of process control computers and their peripherals, utilities (equipment power supply, air conditioning and filtering), termination for field wiring, instrument racks, etc. It also includes the control room from where process operators monitor the process and intervene as required. The number of control houses in a plant may vary from one (full consolidation) to several (partial consolidation).

PART II

FUNCTIONAL REQUIREMENTS FOR FUTURE OPERATOR INTERFACES

INTRODUCTION

The following requirements are addressing desirable features and improvements over existing designs in order to:

- Overcome identified shortcomings
- Achieve more effective data presentation
- Allow total operation from a single working station under all operating modes (surveillance, monitoring, interactive, emergency)

The requirements listed in this specification have been defined without considering the limitations of current technology or the economic impact for their implementation. As such, they represent a set of basic features for future console designs and should be considered as a starting point for further elaboration aiming at:

- Establish the feasibility, cost impact and time frame for implementing such designs or alternates
- Establish the basis for a stepwise approach
- Initiate a constructive dialogue with computer/instrumentation system vendors

Environmental Considerations

The operator interface covered in this specification, is intended to be installed in a control room type environment. This implies that the room is expected to be air conditioned, will be generally free from harmful gas concentrations and will have a "non-hazardous" electrical classification. Nevertheless, the hardware should be designed to meet the following environmental requirements:

	Normal Band	Operative Limits
Ambient Temperature	15-35°C	0-43°C
Relative Humidity (%)	10-50	10-90
H_2S Concentration	0.5 ppm	10 ppm

Control Configurations

Control configurations applicable in this specification could be one of the following:

a) Computer based, centralized

- Computer(s) and instrumentation installed in the control house
- Supervisory control done entirely at the computer level

- DDC may be:

  + Done entirely at the computer level (role of instrumentation is strictly for computer back-up)

  + Done also at the instrumentation level (degree of integration may vary)

- The computer system may be of a distributed, multiprocessor type. Processor to processor and processor to bulk memory communication is envisaged via a high speed digital communication system (data rates are expected to be several Mbyte/sec.)

- The computer system may be interfaced with the instrumentation and other data acquisition systems (DTI's, DVI's,tank gauging, etc), via a medium speed, serial communication link. The characteristics of this serial link are expected to meet the ISA SP72 requirements. Direct interface to the high speed bus via I/O devices is not excluded.

- The instrumentation and data acquisition devices may be:

  + Digital, μP based (from one or several vendors)

  + Hybrid

b) <u>Computer based, dispersed</u>

Same as above, except that control and data acquisition instrumentation may be totally or partially installed in locations remote from the control house.

c) <u>Stand Alone, Centralized</u>

Same as (a), except for computer system. Control and data acquisition devices may still be interfaced with a medium speed, digital communication link.

d) <u>Stand Alone, Dispersed</u>

Same as (b) except for computer control.

It should be noted that:

- Prevailing types will be either (a) or (b)

- In all cases, operation will be from a central location (control house)

- The data communication subsystem should be able to support remotely located consoles. These consoles may not have all the interactive features of the control room console but they will generally have access to the same data base, addressable over the communication subsystem. Other limitations may be applicable (e.g., changing control loop parameters) but these should be implemented via software interlocks.

BASIC DESCRIPTION

General

The purpose of the operator's console covered in this specification is to provide the process operator with the means of communication with the process units under his control as well as with all the components of the control system (computers and/or instrumentation).

Since this console is intended to be the sole operator interface, the reliability aspects of the design assume primary importance. The design, therefore, should include the required hardware redundancy such that the operator will be able to maintain both surveillance and manipulative contact with the process at all times. Depending on the configuration, degree of sophistication and reliability of the distributed computer control system, the operator console could have one of the following configurations:

a) Stand alone system, independent from the process control system and supported by its own processors, dedicated exclusively to the task of retrieving the required information and support the display formats and other features covered in this specification.

b) Directly supported by the distributed computer system and acting as "shared" peripheral. This configuration will only be acceptable if the overall reliability of the distributed computer system ensures a console availability at least equal to that of the stand alone configuration.

In either case, the operator console:

- should be capable of communicating with any device (passive or active) attached to the communication subsystem that may be used with any of the possible control configurations described previously.
- should include the required degree of redundancy and modularity to eliminate any possibility (at least statistically) of total failure.
- should be free of interactive failure modes that can affect either the communication subsystem or any control or data acquisition device (computers, analog or digital instrumentation, peripheral equipment etc.) connected to the communication subsystem.
- should be capable of maintaining all the manipulative and display features even in case of partial failure.

Since a console configuration of a stand-alone type poses certain specific design requirements that may be either non-existent or dealt with by different approaches for configurations described under "b", this specification will mainly address the stand-alone type. Although different configurations are not excluded, they should provide-as a minimum-the same degree of operability, flexibility and overall reliability of the stand-alone type covered in this specification.

Figure 1 shows schematically an overall system configuration. It is envisaged that the control room operator consoles should be connected to the high speed bus used by the distributed multiprocessor control system, while remotely installed consoles (when provided) may be either connected to the medium speed serial bus or connected to the high speed bus via modems.

In any case, the remote consoles should be able to access data available at the high speed communication subsystem. The reliability requirements for remotely installed consoles may vary depending on how these consoles will be used. As a minimum, and when these consoles are only used for read-out purpose, no console failure mode will ever affect the communication subsystem or any control or data acquisition device connected to this subsystem.

Figure 1 also shows additional terminals intended for the use of other groups such as engineering, control applications and maintenance. This specification will not cover the requirements for such devices except from a very general point of view.

It should be pointed out that the operator console should be designed to satisfy primarily the needs of process operators and not those of other working groups.

Overall Design Requirements
For Stand-Alone Consoles (Control Room)

The operator console should be made up of at least two completely independent modules (or stations) such that failure of one will not affect the operation of the other. Each module should have its own CPU, memory, drivers, etc., required to handle on an independent basis the total load. A console data base should be sized on the basis of worst load conditions summarized in Table 1. Table 2 summarizes the load conditions for a typical OM&S application.

As pointed out, the total number of tags given in Table 1, are approximately 11 times the corresponding number of control "loops". For example, the displayed tags associated with an extra-large plant are estimated on the basis of 1000 loops (defined as control points with "valve" output to the process).

These load conditions, stated for sizing the data base of a console, should not be confused with the "normal" load assigned to a console operator. Based on current data, it is not expected that console operators should ever be assigned more than 300 "loops". On the other hand, the console should be able to access all the tags meant to be displayed or manipulated so that any process unit or group of units can be assigned to, or taken under control from any of the operator consoles installed in the same control room.

TABLE 1

LOAD ANALYSIS FOR ONSITES OPERATOR CONSOLES

Type of Tags	PLANT SIZE SMALL	MEDIUM	LARGE	X-LARGE
Control and Non-Control Inputs	Approx. 11% of X-Large	Approx. 27% of X-Large	Approx. 50% of X-Large	2800-3000
Additional Miscellaneous Inputs (TC Inputs, Tank Levels, etc.)				3000-3200
On-Off Alarm Inputs				1600-2000
Other On-Off Variables (MOV's, Pumps, Mixers, etc.)				400-600
Calculated Variables (Control & Non-Control)				1500-1800
Lab Inputs				350-400
TOTAL (See Note 3)	≃ 1150	≃ 2800	≃ 5300	≃ 10400

Notes:

1. The above numbers represent estimated averages and should be treated as order of magnitudes for sizing purposes.
2. The majority of cases fall between "medium" and "large"
3. Totals are approximately 11 times the corresponding number of control "loops" (i.e., control points with valve output).

TABLE 2

LOAD ANALYSIS FOR OFFSITES CONSOLES

High Level Inputs	50 - 150
Additional Miscellaneous Inputs	500 - 800
On-Off Alarm Inputs	400 - 600
Other On-Off Variables (MOV's, Pumps, etc.)	800 - 1600
Calculated Variables (Task Related)	25 - 100
Lab Inputs	200 - 400

The variety of plant sizes as well as the requirement of unrestricted tag assignment will be probably one of the major factors in defining the degree of modularity and expandability of each module. This specification, however, does not intend to address the design characteristics of a module but, rather the conditions defining the design boundaries for a variety of alternative implementations that will be considered acceptable.

Minimum Data Base/Module

In the case of minimum redundant configuration (two modules), sizing of the data base for each module should be based on the following considerations:

- Each module should be able to access and process all tags for display and/or manipulative purpose as well as support the interactive features described later on under "Display Characteristics".

- It is acceptable that only a portion of the data base (representing the console "normal" load) may be active all the time. The inactive portion (representing the remaining load) may be activated on demand but will neither displace or diactivate the "active" data base, nor slow down the console's response time by more than 25%.

- Activation of the "inactive" data base should not exceed 5 sec., and should be initiated from the console keyboard by single action.

If the console is made-up of several modules, distribution of the data base over several modules will also be acceptable, provided that:

- Availability of data for display/manipulative purpose does not depend exclusively on the availability of a particular module.

- On-line reallocation of tasks is possible via program down-loading, or other techniques, affecting no more than one module and not restricted to specific modules only.

- Downloading should be as simple as possible, should be initiated from the keyboard and should be completed within 10 sec.

- Any "overloaded" modules will still be able to perform its previous tasks in addition to the new ones. Under no circumstances should the response time increase by more than 50% of its normal value.

As mentioned before, this specification does not exclude alternate methods of dealing with the basic requirements which can be summarized as follows:

- The design should statistically exclude total failure of the operator interface.

- The display, manipulative and interractive features of each console should remain intact under partial failures with a minimum degree of degradation.
- Each console operator should be able to take under his control any process unit assigned to other consoles.

Data Base Spare Capacity

A spare capacity of at least 10% should be provided.

Tag Definition

For the purpose of this specification, a "tag" is defined as the first level of logical division of the overall system data base.

A tag is not necessarily associated with a process unit control loop and in general may contain other data such as miscellaneous data shared by a number of control loops, additional inputs or information to calculate certain process variables not directly measurable (e.g., net tank volumes) or to define the status of a control device and its phase in a sequence control application (e.g., motor operated valves).

Although a tag is logically a single entity, it will be normally composed of functional sub-entities that may reside in different places throughout the distributed multicomputer system. These sub-entities are associated with a variety of functions that may be performed on a tag, such as for example filtering or other input processing, control, history, association with other software programs, output processing etc.

In general, process tags that have no control function associated with them are of the "information-only" type. If a control function is associated, then the tag will normally correspond to a feedback loop with an input and output associated with it. It should be pointed out, however, that this is a rather simplistic picture, since the details of tag processing are quite complex.

For the purpose of this specification, it should be assumed that tag processing is done by other devices of the distributed control system. The processors supporting a stand alone console will normally retrieve the already processed data associated with the tags and display them in the desired formats.

Control and Data Presentation Transparency

Undoubtedly, tag processing by other devices that may not be available 100% of the time will have an impact on the operator interface. This may involve (as with todays computer based control systems) a fall back to a different and usually degraded type of control strategy. From an operator console's point of view, it is highly desirable that both data presentation and control strategy remain unchanged. It should be kept in mind that one of the basic shortcomings of current operator interfaces is the need to learn a variety of "back-up" control strategies which are much less effective and in most cases totally different than the normal computer control strategies.

The solution to the problem would involve the overall architecture of the control system , the degree of its redundancy and the level of integration between the instrumentation and the distributed computer system and is therefore, outside the scope of this specification. It is important, however to emphasize that if the control system includes one or more levels of "back-up" control strategies the operator console should meet, as a minimum, the following requirements:

- It must automatically recognize any condition that will force the control system to a "back-up" level of control strategy.
- It must revert automatically to display formats compatible with the degree of tag processing and level of control strategy performed now by the back-up control.
- It should clearly indicate on any affected level of display (by symbolic coding or other techniques) the status of tags that may eventually require operator intervention in order to be commissioned for the type of control they have been assigned to.

Figure 2 illustrates this concept with a very simple example of possible back-up control strategy of the type used currently.

Examples of Tags

This specification does not intend to cover in detail neither the tag structure nor the processing of the tags. The following, are representative examples of tags that should be displayed on the console operator. Accessibility to tag parameters is discussed under "display characteristics".

- controlled variable (control tag) usually including
  - input (eventually conditioned)
  - output
  - set point
  - control algorithm

  associated with each control tag, the following parameters may be included.

- alarm limits (deviation, input absolute, input rate of change, failed input,set point, output)

- operating limits (input, output, set point, integral mode, etc.)

- operating parameters (ratio coeffic., bias, input time constant, tuning constants, etc.)

- non controlled variables (information only)

  - input (eventually conditioned)

  - target (if applicable)

  - input alarm limits (same as control tags)

- calculated variables (control or information only).

- Alarm inputs, including

  - input (two state)

  - operating limits (e.g., adjustable dead band)

- Two state variables including:

  - on-off type inputs (usually two or more)

  - on-off output(s)

  - operating limits (e.g., adjustable time limits for expected completion of command)

  - operating parameter (normal/abnormal/intermediate status)

These tags are usually associated with the status of motorized equipment such as MOV's, pumps, mixers, etc.

For the purpose of this specification, auxiliary calculations (such as multiplication, division, input dynamic compensation, signal selection, etc.) are not considered tags. These calculations are to be considered as associated with the processing of tags.

Usually, elements performing auxiliary calculation (whether defined as "blocks", "slots", "tags" or subroutines in current software packages), are not required to be displayed on the operator console, except in special "block diagram" displays of certain control configurations, or when the information to be displayed is only available from such elements.

Overall Display/Manipulative Capability

- Each module should be capable of supporting up to 4 screens, each with its own independent keyboard.

- Each screen should be capable of displaying all available formats with the possible exception of graphic presentation of process units (i.e. in general, they shall not be dedicated screens to specific type of display formats).

- Special keyboards such as those eventually used for data base configuration, building graphic displays etc,that include dedicated functions used very seldom by process operators, should be detachable via plug-in connectors so that they are used only when required and avoid cluttering of the working area.

- To the maximum extent possible, the operator console should be kept free from any feature required to satisfy the needs of other working groups such as engineering, control applications and maintenance. The operator , however, should be given warning concerning abnormal conditions of the control equipment so that he can initiate certain actions under his responsibility or request the intervention of other groups.

Actions that the operator should eventually initiate without assistance from other groups could be the following:

- Switching to a redundant communication bus (even when automatic switchover facilities are provided)

- Program downloading (e.g., via cassette or diskette) associated with the data base of his console

- Realocation of tasks of a failed module

- Restart/reload of other processors belonging to the distributed control system.

Warnings to the operator about the status of the control equipment should be normally general in nature but clear enough to let him evaluate the effect of a failure on the process and the corrective action he needs to take to recover from such failure (not to fix the failed equipment). Examples of such warnings could be:

- bad input (normally controlled output freezes at last value)

- failure of a device locally or remotely installed, used either for control or data acquisition and list of tags affected by such failure.

- etc.

Extensive on-line diagnostics are expected to be provided throughout the system but should be displayed elsewhere (e.g., "maintenance" console, local diagnostic panels, etc.)

- In general, the operator should be able to communicate with
  - DDC control functions
  - Supervisory control functions
  - Back-up instrumentation (if and when used)
  - Historical data base

  The console must include <u>all</u> functions of currently used analog back-up panels (even in the eventuality that such panel is still provided).
- The console procedures must be simple and logical so that operators should be freed-to the maximum extent possible-from thinking about the mechanics of performing console functions. This should allow them to give maximum attention to process operations.
- Some functions, such as manual shutdown pushbuttons and critical alarms will most likely continue to be implemented via hardwired techniques (totally separated from digital, shared line techniques). These functions should, however, be arranged conveniently and integrated into the console area.

<u>Screens</u>

- It is envisaged that, because of considerable lead over competing technologies, CRT's may continue to be used. This specification, however, does not reject the use of different technologies provided that they offer display capabilities (in terms of resolution, color, number of characters, etc.) at least similar with high quality CRT's, available today.
- The basis for this specification are high resolution, 15" diagonal, color CRT. Other sizes are not excluded but it should be kept in mind that space may be at a premium and screen dimensions should be compatible with the requirements of:
  - Optimum viewing distance
  - Operator's reach
  - Total number of screens
- It is expected that the minimum viewing distance will be 28" (approx. 71 cm). Under this viewing condition the operator's "span of attention" (5 deg. visual angle) corresponds to 2.44 in (6.21 cm) measured on the screen face.

This should be kept in mind so that multicharacter labels fit preferentially into this field. Ideally, the information associated with a tag, especially for multiloop formats, should fit into a screen sector intercepted by the 5 degree solid cone of vision.

The numeric word size associated with this screen sector should not exceed a max. of 12 characters and for textually coded information should not exceed a max of 8 characters.

- The character size should be compatible with a viewing distance at least 1.5 times the minimum one specified above. Compatibility of display resolution and addressibility of the driving electronics is essential. Whenever CRT's are used, the addressibility of the electronics should be in general better than that of the display so that dots can be written to overlap one another in a manner that makes them look like lines and not a string of dots. Character size should not exceed 0.5 cm (0.2 in.)

- Display loading (% of active screen area) should be optimized around 25%. In addition, change of status (e.g., alarm condition), selection of a group of data by the operator (such as a loop) or manipulation of certain parameters (e.g., set point, output) should be always highlighted on the screen (e.g. intensification of characters, "reverse video" etc.). Blinking should be limited to indicate alarm conditions and should be associated with a portion of the displayed information that is either added when the alarm condition appears or has secondary importance in relation to the text being highlighted (e.g., in a horizontal text orientation of an alarm message, the tag and description should only be intensified; blinking may be associated only with say the time of occurance-appearing on the right of the text-or with a symbol or character, such as an asterisk, added to either side of the text). It should be remembered that blinking the entire text, makes it difficult to read. Blinking, however, of a symbol on a graphic presentation (line, pump, valve, etc.) is perfectly acceptable and may certainly be used.

  In general, color coding should also be used to further highlight information of interest and assist the operator in focussing his attention

<u>Response Times</u>

There are several response times associated with the operation from "shared" type display devices. These are discussed below:

- Call-up of new displays

  Basic operating displays such as overview at unit level, multitag control displays and alarm displays should have response times of less than 1/2 second for new display request. This means that the new display will appear on the screen in complete format within 1/2 sec.

Other displays, such as overviews at multiunit level, multitrend (plot) displays or graphic presentation of process diagrams may have somewhat slower response time. This time should normally be between 1 and 2 sec. During periods of high system loading, a response time in the range of 2 to 3 sec. will still be acceptable but this figure should not be considered as design objective. Whenever, the response time exceeds 1 sec., either an immediate feedback should be given to the operator (so that he knows his request is being processed) or the display should be built gradually and completed within the stated response time.

- Updating of displayed information

Ideally, updating of the displayed information should be consistent with the frequency of tag processing. For example, from an operator's point of view there is no point in retrieving and displaying a piece of information say every 2 sec., if this information is processed every 16 sec. On the other hand, syncornization of display updating with the frequency of tag processing may add considerable complexity and should be considered in connection with the other components of the control system and only if there are advantages to be gained in terms of utilization of available resources.

Display updating in the range of 4 to 5 sec. has been found generally acceptable even for tags being processed several times during this time interval. If a fixed update frequency is used for all displayed information, the above figures should be used.

- Response time under direct manipulation

During the "interactive" mode, operators may change certain parameters associated with a tag such as for example change the control mode, adjust manually a set point or the output signal to a control valve.

Whenever the operator manually changes an adjustable parameter, the new value of this parameter should be updated on the screen within 1/2 to 1 sec. of the change. These new values should represent confirmed "echoes" transmitted back from the device that will process the requested change.

The need for executing the requested change within the above stated interval is a more complex matter and may not be generalized for all cases.

In general operator initiated actions for tags in automatic control such as opening a cascade link or changing the hierarchical level of control implementation (e.g., passing from automatic control performed by the "back-up" instrumentation to automatic control performed by the computer or vice-versa) should be executed immediately by the device that process the tag. The same holds true when transferring a control tag to manual mode. On the other hand, a set point change may either be registered by the device processing the tag, to be used later on when the tag is scheduled to be processed, or cause the tag to be processed immediately. The latter case is not required if the tag processing frequency matches the lag in system response and should have no effect on human operators (in terms of "growth of uncertainty") since their behavior has been found to quickly adjust to the particular system lag.

For control tags transferred to manual mode, operator initiated changes concerning the output should always be executed immediately, independently of any preassigned frequency for tag processing.

In addition, under manual control of a process variable operators' uncertainty increases and, consequently, also their need for more frequent process feedback especially when the variable being manipulated is near the limits of its tolerance range. The same holds true when in an emergency the operator fully shuts a valve and expects an immediate feedback from the process, to verify if his control action had the desired effect. In manual mode, therefore, and under direct manipulation of the outputs, the process inputs associated with control tags directly outputting to the process should be updated on the screen at least every second.

Ideally, the frequency of this updating needs only to match the system time lag and the rate of growth of uncertainty of the operator for each condition. However, quite extensive knowledge of the factors contributing to uncertainty is needed to form even approximately realistic estimates. The above figure, therefore, although not optimized for "minimum sampling" conditions, will offer the freedom of unrestricted visual sampling.

Safety Interlocks and Pacing

It is envisioned that several interlocks will continue to be used, either to prevent accessibility to certain parameters or to force the operator to use certain console procedures.

This aspect should be carefully considered since it has considerable effects on the human operator. The operator is said to be "paced" by the machine if he has little choice of what to do, how to do it and when to do it. On the other hand, the operator is said to be "pacing" the machine if he decides on his own what to do, how to do it and when to do it. This leads to the following considerations:

- To the maximum extent possible, rigid procedures should be avoided by providing multiple choices for the execution of a task.
- The operator should be given the maximum possible freedom of building, altering or adding displays as well as changing the way they are cross-referenced.
- Safety interlocks should never be designed as unmoveable blocks and their commissioning should be left to the user.

- Whenever the operator uses the wrong procedure causing the activation of an interlock, a message should be generated telling him not just that he made an error but also what to do or why it is not allowed to proceed. For example, if changing of alarm limits is protected against unauthorized changes, the message on the screen should not simply say, "illegal entry" but should also explain why it is illegal, as for example, "password is needed" or "access interlocked by key". The above consideration should also be applicable to the design of other consoles such as the engineer's console, etc.

Hard Copy

It should be possible to obtain at any time a hard copy of any display information. This hard copy should be preferentially an exact replica of the information being displayed on the screens.

Time Indication

In general the time and date should be displayed with each display format.

Reports, Logs, History and Messages

In general, logs, reports, history and various messages (such as historical data, hourly averages, daily averages, daily reports, etc.) should be stored and generated by other devices of the distributed system. The user should have the ability to generate his own formats to suit his specific needs using separate alphanumeric peripherals.

Unless specifically stated in this specification, the above data are not intended to be displayed on the operator console's screens. The operator, however, should be able to request their presentation from his keyboard so that they can be displayed via other peripherals of the system, such as high speed printers.

The description and/or definition of the various types of logs, reports, etc., is outside the scope of this specification and in general is part of the functional description of the distributed, multi processor system. As far as this specification is concerned, it will only address console related actions that should be logged, the general nature of messages to be presented to the operator and the display format of relatively short term historical data.

- Console related logs

The purpose of these logs is to enable the reconstruction of events that had involved the human operator. In summary, are the following:

300

- General Alarm Log

  Any alarm event, including changes of alarm limit, inhibition of alarms or setting of alarm conditions to "invalid" status should be logged in sequence and kept in bulk memory. This log should retain the last 5000 events. A subset of this log should be available for display on the console's screens (See "display characteristics" for details).

- Console Log

  All keyboard initiated changes or entries should also be logged in sequence. This log should retain as a minimum the last 3000 keyboard initiated actions for each console.

DISPLAY CHARACTERISTICS

Generation, Processing and Presentation of Alarms

Alarms are in general a form of condensed information that is presented to the operator in a manner requiring minimal search and identification. Lenghty messages are not alarms and should never be used to alert operators of critical conditions. Alarms should be interpreted at a glance and shall not require study of the text or extensive search by the operator.

- Alarm processing for display purpose

  - It should be possible to assign up to 2 different levels of priority to each alarm (whether indirectly or directly generated) for display purpose as follows:

    Priority 1: Critical alarms, associated with immediate identification of conditions that may result in injury to personal, damage to process equipment or large operating debits.

    Priority 2: Important abnormal conditions that must be acknowledged by the operator and assist him in initiating actions that would prevent the development of serious upsets.

    Generally, the condition being alarmed will not result in an immediate disaster on the unit and required reaction may occasionally be slower than reaction to Priority 1 alarms.

  For display purpose, all other abnormal conditions to be displayed should be considered as "messages", discussed later on in this specification.

- It should be noted that other alarm priorities may be associated with tag processing but these should not be confused with the display priorities mentioned above. The purpose of alarm priorities associated with tag processing is outside the scope of this specification, but, in general, they may be used to determine which condition, if any, should be displayed to the operator. It should be also noted that alarm conditions having different processing priorities may very well have the same "display" priority, as defined in this specification.

- During configuration of the data base, alarm conditions associated with any tag, should be individually coded for display priority level.

- It should be possible to change alarm priorities from the console's keyboard on a single tag basis. Provisions to prevent free access via interlocks should be included, but the decision to activate these interlocks should be left to the user.

- Alarm priority should affect alarm presentation in accordance with the guidelines listed below (see alarm presentation).

- Alarm limits, whenever adjustable, will be accessible from the console's keyboard. The same holds true for adjustable dead-bands (or time delay) associated with the change of state of non-adjustable contacts (such as field switches).

- Alarm inhibition should be possible for any alarm, independent of its priority, either individually or in groups.

- Any change of alarm limits or alarm inhibition should be automatically logged. This log should be available for display on a screen at any time and should retain a history of the last 30 changes or at least sufficient enough to cover a period of 48 hours. Provisions should be made to clear this log from the keyboard.

- Modification of alarm limits, inhibition of alarms and clearing of the alarm status log mentioned above, should be in general protected by interlocks to prevent free access. Activation of these interlocks, however, should be at the option of the user.

- Alarm messages

Any abnormal condition that, if not acknowledged by the operator, may affect optimum control of the units or may lead in time to priority 2 or priority 1 conditions should be considered as "alarm message".

These messages may have variable formats to suit the specific type of infomation to be presented to the operator and in general they should not be limited to text forms.

Alarms messages, should be used preferentially to indicate to the operator what sort of action is recommended or what is the predicted effect and not just describe the condition that is in abnormal status.

- Alarm generation

  Figure 3 shows schematically how alarms are generated in current, computer based control configurations with analog "back-up" instrumentation. The use of microprocessor-based, digital instrumentation may introduce new options which may or may not be used, depending on the degree of software integration between computers and instrumentation and the use of the instrumentation data base.

  It should be assumed that in future control configurations a certain number of process alarms, generated by on-off type sensors, may still be handwired to independent alarm logic cards driving dedicated windows. These alarms, however, should also be displayed on the console's screens. In general, contact input associated with process alarms could be brought into the console either via a digital I/O box connected to the medium speed bus or the high speed bus or via an I/O peripheral, supported directly by the console processor(s).

  It is expected that all other alarms should be accessible via the digital communication subsystem, using the communication potocol established for digital transactions with other devices connected to the communication subsystem.

  For software generated alarms by the distributed multiprocessor system, it should be noted that the information to be displayed may be uniquely determined by the levels of priority associated with tag processing or other software programs.

  To the maximum extent possible, alarm generation, limits and status (i.e., active or inhibited) should be kept in one location of the system's distributed data base and automatically updated when changes are made from anywhere in the system.

- Alarm presentation

  As it has been pointed out before, several alarm limits may be associated with a single tag. It was also made clear that not all these conditions should necessarily be presented to the operator as alarms. Several may not be displayed at all and only used by the control system to initiate control actions or eventually to generate alarm messages. However, whenever more than one of these conditions is to be displayed to the operator (either as an alarm or as a message), the presentation should be consistent and logical such as to:

  - avoid redundant presentation if one condition is the logical consequence of another.

  - attract the attention of the operator to the more severe condition.

  - inform the operator about the validity of certain displayed conditions if this becomes questionable as a result of another event.

For example, if a bad input is detected during tag processing (e.g., transmitter failure causing the input signal to drop to zero), several other alarm limits that may be associated with this tag, will be eventually violated, such as low input, rate of change or deviation. In this case, however, the detection of these abnormal conditions is not only the logical consequence of the first one (bad input) but their validity is at least questionable. The only information that the operator really needs is that there is a suspected equipment failure and should not be given process oriented alarms (which in fact may not be true). Indeed, if an alarm is associated, say with the low input condition, it should be set to "invalid" status and this should be clearly indicated on all display formats on which this alarm condition may be shown.

It is envisioned that a number of screens (at least two) will be normally used for displaying alarm conditions. The procedure to select a screen for "dedicated" displays will be discussed later.

(a) Priority 1 alarms

- Priority 1 alarms should automatically appear on a selected screen or portion of this screen (no operator action).

- If an alarm is also hardwired to a separate alarm system then:

  + this system should be either mounted on the console or installed close to the console (readily visible).

  + it should be acknowledged by the same keys or pushbuttons used with the console's screens.

  + it will not be affected by any console failure mode.

  + it will not affect the operation of the console in case of total or partial failure.

  Such system could be ,for example, a miniature type annunciator panel (e.g., 3/4" x 3/4" windows), completely divorced by the console hardware and software except as mentioned above.

- Alarm message on the screen will not occupy more than one horizontal line and will include tag I.D., description of alarm, type of alarm, and time of occurrence.

- New alarms will appear at the top, pushing down previous alarms and will be highlighted as described previously (blinking, color coding, etc.).

- Alarms will remain on the screen when acknowledged and will disappear when alarm condition returns to normal. Empty spaces in the stack will be occupied by existing alarms that may be displayed at that time (push-up).

- The screen (or its used portion for priority 1 alarms) should have a display capability optimized for 20 alarms. If overflow occurs, the alarm listing will continue on a next page and this should be indicated on the screen (e.g., page 1 of 2).

- If the same alarm repeats itself, only the latest occurrence will be retained on the screen. The entire sequence, however, should be kept in memory and presented as alarm log on demand (hard copy).

(b) Both priorities

- Any alarm will activate a matrix of individually addressable elements such as a matrix of backlighted pushbuttons Although this specification does not exclude the use of any specific methods to implement this matrix, the design should meet the basic functional requirements described later on and must be consistent with the design criteria concerning the interaction between the modules of each console listed previously. For simplicity, this matrix is referred to in this specification as "pushbutton matrix". The "pushbuttons" of the matrix may have other functions when the associated screens are not selected for alarm displays.

- The above matrix should contain a number of active buttons consistent with the alarm data base. Each button will correspond to an alarm page and used to call-up that page only. A minimum display capacity of 20 alarms per page should be provided but higher densities are definitively desirable when dealing with a large data base. The buttons will have an ISA-1 flashing sequence (same as panel annunicators).

- The alarm pages will have the following characteristics:

  + will continuously display all tags assigned to the page.

  + will highlight the tag in alarm condition (blinking, color coding, etc.)

  + will indicate the priority of the tag in alarm condition, preferably via color coding.

  + it will be possible to inhibit alarms, addressing the tag on the page. Inhibited alarms should be clearly identified, preferably both via textual and color coding (suggested color: blue).

  + it should indicate "invalid" status of alarms, again via textual and color coding.

+ it will indicate the code or page number of multiloop displays associated with each alarm as applicable.

+ the format of each horizontal line should be the one described for the priority 1 alarms.

- The matrix of lighted pushbuttons will act as "group" alarm display and will have realarm features (new alarms will always initiate a flashing sequence). Acknolwedged alarms will be indicated by steady lights.

  The priority of the alarm initiating the alarm sequence will also be indicated on the alarm matrix (same color coding as the one used on the screen). Pages containing inhibited alarms should also be identifiable at the "group" alarm matrix.

- The last row of the matrix should be normally associated with special function keys. These keys will have the same functions for all keyboards and should normally be used to select a screen for special purpose displays (e.g., priority 1 alarms). One of these keys should be used for immediate call-up of the multiloop display, which is logically associated with the alarm condition. This key will have the same function on any given keyboard. and will always call-up on the corresponding screen the multiloop display associated with the first unacknowledged alarm condition that may exist at that time. If more than one alarm condition exist, the multiloop display that will be called-up will correspond to the alarm condition having higher priority.

- Alarms will be acknowledged by pressing the corresponding matrix button or the above mentioned special key. The audible signal will be silenced immediately but the visual coding associated with unacknowledged status (such as blinking) will remain on the addressed display for at least 4 sec. However, it should be also possible to acknowledge priority 1 alarms appearing automatically on another screen from the keyboard of that screen. This acknowledgement will not call-up the alarm page containing the same alarm but will switch to acknowledge status the corresponding matrix button(s).

- The alarm condition, whenever is directly associated with the processing of a tag (e.g., PV of a control tag exceeding its alarm limits) will also be highlighted at (a) the "overview" display level (if contains that tag), (b) the multiloop display containing that tag and (c) the single tag display level.

- If an alarm condition associated with a multiloop display occurs while this display is being shown on any screen, the alarm status at the multiloop display level should also be highlighted (blinking, color coding etc.) Selecting any loop on this display will acknowledge the alarm, and no further acknowledgement will be required, provided that other alarm conditions do not exist.

- Similarly, if the alarm condition occurs while an overview display containing the associated tag is being displayed (on any screen) the above condition should be indicated by blinking the group containing the tag in question. The alarm will again be acknowledged when passing from the "overview" level to the multiloop level with the same procedure mentioned before (audible signal silenced, visual coding changing to acknowleged status after 4 sec.)

- The priority assigned to each alarm condition will be indicated not only visually but also audibly (i.e., different audible signals for different alarm priorities).

  Ideally, it should be possible to adjust the frequency, tone and loudness of the audible signals.

  Each console should have clearly distinguishable audible signals from other consoles in the same control room. These signals should be associated with each console's "normal" data base.

- Presentation of alarm messages

  - It should be possible to display these messages on any screen by using the keyboard associated with that screen. The existance of a message waiting to be displayed should be visually indicated on each keyboard. In general, audible signals should be avoided for messages. Call-up should be prefferentially via single action.

  - If a split screen is used for priority 1 alarms, the lower portion of this screen may be used to display alarm messages.

  - In general, messages should not be displayed automatically (operator call-up). It should be possible, however, to differentiate between general alarm messages and messages associated with equipment failures (lists of affected tags, etc.).

- Invalid Alarms

  - Invalid alarm conditions should be indicated on all display formats on which such alarm conditions are normally displayed. In addition, a special log, listing all invalid alarm conditions (not to be confused with inhibited alarms) existing at any time should be maintained and displayed on request.

  - The existance of invalid alarms should be preferentially indicated on the alarm matrix (same color coding as the one used on the screen). If no other warning is available, the alarm condition being invalidated should initiate an alarm sequence (to be acknowledged as with valid alarm conditions).

Summary of Alarm Handling

In conclusion the main characteristics of alarm processing and presentation can be summarized as follows:

- The alarm pushbutton matrix (group alarm) and the associated alarm pages will be equivalent to scanning panel mounted annunciators. Pattern recognition is, therefore, maintained at both the matrix and alarm page level.

- The priority of the alarm condition will always be indicated both visually and audibly.

- Color coding should be extensively used to highlight alarm conditions together with other visual coding such as intensification and/or blinking.

- Alarm conditions may appear in several formats and should be acknowledged in a logical fashion, avoiding unnecessary steps.

- Maximum flexibility should be retained in assigning to any given screen the role of displaying priority 1 or prearranged alarm pages. Time of alarm occurrence will be indicated in both type of displays.

- Immediate access to the appropriate type of display for verification or action will ensure minimum time spent for search and identification.

- Redundant presentation of alarm conditions should be avoided if one condition is the logical consequence of another.

- Invalid alarm conditions should be clearly indicated as such.

Overview Capability

- Deviation type

  - Surveillience of a large number of variables should be via "overview" type displays. The basic overview display should consist of prearranged groups of variables - corresponding to multitag (multiloop) formats - that will be displayed in dedicated sectors of the screen. The parameter used should be the deviation of a variable from its set-point or target.

The deviations should be shown as vertical bars extending from a horizontal (zero-deviation) line. The maximum number of variables/group should correspond to the max number of tags that should be assigned to multitag formats. This number should be preferably 8 but could be as high as 16 provided that viewing clarity is not compromised and that the multitag formats contain all the information (see multitag displays) without overloading the screen.

- It should be possible to assign any multitag group to a deviation type display independently of its sequential number in the console's data base. The procedure for building or modifying this overview display should be simple and available from the keyboard for online reconfiguration. The % in deviation required to produce a given excursion should be individually programmable for each variable from 0 to 100%.

- A normalized band of acceptable deviation should be clearly identified. Deviations exceeding this band will cause the corresponding group to change display color (e.g., from green to amber). However, none of the characteristics of alarm status (such as blinking etc.) will accompany this warning unless the exceeded limit happens also to be an alarm limit.

- The "zero deviation" (reference) lines should be clearly separated one from another to indicate each group. Within a group, this line should be further subdivided in smaller portions such that each portion should contain no more than 4 variables.

- It should be possible to assign to any screen the role of displaying deviation type "overview" displays either by using the standard keyboard or by using one of the special function keys (last row of pushbutton matrix). In this case the screen will "lock" to the "overview" display unless the key is again depressed. The key should be back illuminated with the light "on" in the locked position.

- Each group on the overview display will be addressable via the pushbutton matrix described before. In this case, the active keys should be used to call-up each group in the format described under multitag displays.

  If the screen is "locked", the multitag format should normally appear on another screen assigned specifically to multitag displays (again via a special function key) or to any "unassigned" screen by default option. Otherwise, the multitag display will appear on the same screen, replacing the overview display.

- Both the groups on the screen and the pushbuttons of the matrix should be numbered for easier identification - Group description will only appear on the screen (approx. 8 character descriptions/group), not on the matrix.

- As mentioned before, certain alarm conditions will also appear on the overview display.

  Any group containing a tag in alarm condition will be highlighted (blinking, change of color etc.) The corresponding pushbutton on the matrix will initiate an ISA-1 flashing sequence.

- The deviation type "overview" is intended for the "unit surveillance" operating mode. Its overview capability may contain 250-350 key variables, but should be optimized for 320 variables. The upper limit should not exceed 400 variables.

- At least two pages should be available at this level of display, selectable on demand. However, more pages may be required, depending on the size of the data base.

● Bar graph type

- This overview display is mainly intended for "unit monitoring" and the level of detail should be between the deviation type "overview" and the multitag format.

- This display should be optimized for 64 tags displayed as groups of vertical bars representing the input (measured or calculated). The set point (or target) and the operating mode for control tags should also be indicated on this format. The groups should appear in preassigned sectors of the screen addressable via a set of dedicated keys that will also be used to address the individual tags on multitag displays. The addressed group should be displayed in the format described under multitag displays.

- Alarm conditions will also be shown at this level of display (highlighted by color coding, blinking, etc.)

- The alarm condition will also be acknowledged when a group is addressed via this set of selection keys.

- At least 20 pages should be available at this level of display, but eventually additional capacity may be needed, depending on the size of the data base. Biderectional "paging" should be possible via paging keys.

- It is expected that a certain number of pages should have a logical association with the grouping of tags used in the deviation type overview. (i.e., the same groups of tags arranged in the same numerical sequence). In general, however, this association should not be rigid and complete flexibility should be left to the user to arrange the grouping of tags in any desired way. The console software, however, should allow the option of group association between the two hierarchial overview levels, such that group allocation for one is at the same time implemented for the other.

  In general, the display capacity at this hierarchial overview level should exceed the display capacity at the deviation overview level.

310

- The library of pages at the "bar graph" overview level should also be addressable via the pushbutton matrix. The matrix will automatically become a page directory whenever the associated screen is displaying this overview format. The active buttons should be clearly identified and should maintain a logical relationship with the arrangement used for the "deviation" type overview whenever the two hierarchical overview levels are associated as mentioned before.

- It should be possible to use any "unassigned" screen for displaying this overview format using the associated keyboard. No special function key is required to lock the screen to this overview level.

- Call-up of trend displays should also be possible via "display association" with other display formats (see multitag and trend formats).

Multitag Formats

- This display format is intended to be the main working format for "interactive" and "emergency" modes.

- The tags should be presented in digital/bar graph combination format,

- The tags shall be addressable by a set of dedicated keys, one for each tag and located in such a way as to make the correspondance between the sector of the screen occupied by a tag and the key itself, unique and unmistakable.

- As a minimum the following information should be shown in the graphic portion of the display:

  - Input (vertical bar graph)
  - Output (vertical bar graph)
  - Setpoint or target (vertical bar graph or horizontal marker)

  The shape of these bars should be clearly differentiated so that visual interpretation of the displayed parameter is facilitated..

  In general, alarm limits (if applicable) should not be displayed on this format except upon request. If these data are superimposed on the main graphic portion, they should be presented in such a way as neither to mask nor confuse the displayed information. The superimposed information should be visible as background by using appropriate display techniques such as low intensification and/or dark color. For superimposed data, the presentation should be as follows:

  - Input alarm limits (horizontal markers or vertical bar next to the input).

- Output limits (horizontal markers).

- Set point limits (horizontal markers or vertical bar, complementing the set point presentation, i.e., if the set point is shown as horizontal marker, the limits should be shown as vertical bar and vice-versa).

As a minimum the following information should be shown in the digital portion of the display:

- Tag I.D. (6 alphanumeric characters, max.).

- Description (8 characters - see also footnote).

- Engineering units (5 characters)

- Input, expressed in eng. units (7 characters max. including decimal point and sign).

- Set point or target expressed in eng.-units (same as input).

- Ratio coefficient or bias coefficients (6 characters when applicable).

- Output in eng. units (7 characters, only when cascading a secondary loop).

- Output in % (6 characters, including decimal point and sign).

- Operating mode (Auto, Manual, Cascade, Computer).

- Back-up function (if and when applicable).

The following information should also be indicated on, or easily deducted from the display by using symbolic coding (shape) or color coding or other appropriate techniques:

- The provenience of the set point for cascaded control tags, (such as from a supervisory program).

- The degree of freedom for parameter manipulation under current operating mode.

• The graphic portion of the multitag format, should be replaceable on demand by a deviation type graphic display. The deviation should be shown as a vertical bar, extending in two opposite directions from a reference (i.e., zero deviation) line.

Note: A full description (24 characters max.) should be usually provided. This full description, however, may only appear at a preassigned screen area when a tag is selected, in order to keep the rest of the display "clean".

- The lower portion of the screen should be used to indicate the associated displays with the one being currently displayed.

  At least up to 4 associated displays may be used in connection with any given multitag display. The associated displays can have any of the following formats:

  - Multitrend
  - Bar graph overview
  - Multitag

  A set of 4 dedicated keys should be used to call-up any one of the "associated" displays. The associated displays should be normally displayed on those screens assigned to trend displays and by default option to any "unassigned" screen. However, they could also replace the multitag display on the screen being used for their call-up if the operator chooses so. For this purpose a 5th key should be used to lock or unlock the screen from the display being currently displayed. This locking function should only be applicable in connection with the presentation or "associated displays". A 6th key should be provided to allow immediate return to the "previous" display. The "return to previous display" key should be always active and should apply to all types of display formats.

- Any alarm condition associated with a tag should be highlighted directly on this display format. The preferred method should be to associate the alarm condition with the graphic portion by highlighting the parameter causing the alarm; the digital portion, however, can also be used.

- Shared keys can be used for mode changes, set point, output etc., but they should be kept well separated from any other cluster of keys.

- Changing of set point and output should be possible by any one of the following methods:

  - Slow ramping (increment/decrement key)
  - Fast ramping (separate increment/decrement keys)
  - Full value (via alphanumeric portion of keyboard)

  Full value changes should be shown on the screen next to the previous value and after visual confirmation by the operator should be entered ("enter" key).

- **The multitag displays should also be :**
  - **Configurable from the console's keyboard (add, delete or change a tag) via simple procedures so that operators can build or change them as they wish.**
  - **Listed in special formats (library).**
  - **Identified by general description and numbered.**
- **The number of multitag pages to be provided should be consistent with the size of the data base and as a minimum should cover the total number of control and information-only tags, including calculated tags. In addition, a certain spare capacity should be provided to be used by the operators to build displays of their own choice as conditions may dictate.**
- **Paging (both directions) should be provided (same keys as for bar graph overview).**
- **All multitag displays, whether corresponding to overview groups (call-up via push button matrix) or not, should be addressable for call-up at any time via the console's keyboard. The procedure should involve one function key and the alphanumeric portion of the keyboard.**

Multitag Formats For On-Off Control

Control of pumps, MOV's and other two state control devices will be normally from a different display format having the following basic characteristics:

- **Both states should be indicated by textual, symbolic and color coding (such as, for example, two squares, labeled and color coded) appearing on the graphic portion. The digital portion may include: equipment tag number, rpm, amps, output pressure, suction pressure,etc. Except for the equipment tag number, other information may or may not be displayed but provisions should be made to allocate appropriate screen space for their inclusion if applicable.**
- **Intermediate state should be indicated with both states activated (e.g., both symbols colored).**
- **Color coding should be in general as follows:**
  - **Green : equipment running (pump, mixer etc.); valve open**
  - **Red : equipment not running (or tripped); valve closed**
  - **Both green and red: intermediate state (MOV's).**

- Blue : equipment not responding to remote command } may or may not be used
- Amber : equipment under local control

This format should be again optimized for 8 tags. Each keyboard should include separate keys (shared by all tags) for the execution of commands such as start, stop, close, open. Operation of MOV's would normally require 3 keys (open, close, stop). Operation of a two state, motor driven equipment (such as a pump), will only require two keys (stop, start).

Single tag displays

These displays are very seldom used by the operator and are mainly intended for application engineers.

From this type of display, operators may have access to data such as:

- Alarm limits
- Applicable indexes for deviation type overview
- Tuning constants
- Loop parameters (ratio, bias, filter time constant, output limits, integral limits etc.)

Interlocks preventing free access should be at least applicable to

- Alarm limits
- Tuning constants
- Integral limits

Special single tag displays should also be available as for example for tank gauging applications. These displays should be normally limited to 30-40 for onsites applications. In general they will include more than one input (e.g., level and temperature of a tank) as well as calculated variables (total gross volume, total net volume, specific gravity etc.)

Trend Formats

- Multitrend Displays

These display formats a[illegible] intended to replace pen recorders. The display characteristics of these formats can be summarized as follows:

- The format should be capable of supporting simultaneous trend displays of a number of variables at least equal to the max. number of tags used in multitag formats.

- It should be possible to configure these formats in advance in the same manner as with the multitag displays and with the same ease.

- Any screen could be selected for displaying trends, either by using the standard keyboard or one of the special function keys of the matrix. A second function key should be normally used to predispose the screen to automatically accept requests for presentation of "associated" displays generated from other screens.

- Call-up of multitrend displays should be:

  + Via the pushbutton matrix of the screen selected for trend displays (single action).

  + Via the keyboard (e.g., "trend" function key & display code).

  + Via association with other display formats.

- The same considerations, listed for the bar graph overview and concerning the grouping of tags, should also be applicable to these display formats. In general, however, a lower degree of association is expected between the grouping of tags for overview or multitag format and multitrend formats.

- The number of pages to be provided at this level of display should be consistent with the size of the data base. As a minimum it should cover 20% of the total number of control and information-only tags.

- The library of pages should be addressable from the pushbutton matrix whenever the associated screen is displaying a multitrend format. The active buttons should be clearly identified.

- The "display association" feature (see multitag displays) should also be applicable to multitrend displays.

- All variables displayed on a multitrend format will always appear with a minimum of 2 hours history and will revert automatically to real time trending.

- In general, a 5 to 6 sec. updating is considered adequate for any variable. Slower updating of the order of 10 or 20 sec may certainly satisfy a large number a process variables, but cannot be generalized. On the other hand the need to display the entire history at a high sampling rate is very seldom required.

Based on the above considerations, the following requirements represent a generalized attempt to satisfy a variety of considerations associated with a consistent presentation of multiple trends of variables having different process lags.

+ The displayed history can be shown in two portions, having different resolution in terms of sampling rates.

+ The "high" resolution portion should consist of the last 10-15 minutes, where each displayed point will correspond to a sample taken every 5-6 sec.

+ The "low" resolution portion should consist of the remaining time interval (105-110 min.); each displayed point will now correspond to samples at 20 sec. intervals.

+ The two portions should be clearly distinguished (e.g., by color coding of the corresponding time scales on the horizontal axis).

- More than one trend can be displayed on the same screen sector using different color for each trend.

- It should be possible to change the display content on-line by deleting, adding, or substituting variables to be trended.

Any variable available within the distributed data base could be assigned to multitrend formats. It should be kept in mind, however, that the purpose of this display format is to present the operator with the type of information he gets today from multipen, shared recorders. For the same reason, the display should be kept as simple as possible and should not emulate the characteristics of recorder charts.

The information displayed with each trend should include:

- Tag I.D.

- Short description (optional and not exceeding 8 character max.)

- Time marks (1/2 hr interval) on horizontal axis.

All scales should be linear 0 to 100% (4 divisions only) and should correspond to the measuring span of the field transmitter.

- General Purpose Trends

It should be possible to display the history of a variable, whether independent (i.e., having tag status) or associated with a tag (input, calculated variable, calculated target etc.) in trend format.

The display requirements for such trends may be quite different from those applicable for multitrend displays. In general the display characteristics should include:

- Possibility to use expanded scale (in engineering units).

- Different time scale options.

The digital information displayed with such trends may in general include the following:

- Digital value of variable (last sample)

- Tag I.D. and description

- Scale in engineering units

- Time marks

More than one trend may be displayed on the same screen. However, time scale syncronization is not a requirement.

Graphic Display Capability

As mentioned in Part I, graphic presentations of process flow diagrams on video screens is extensively used in Oil Movement & Storage (OM&S) applications. It has proven a very powerful tool and significantly assists the console operator in preforming the assigned tasks.

This specification addresses graphic presentation of process units and flow diagrams for onsites applications. It is generally believed that such presentations, if properly engineered, can certainly assist the console operator to better visualize the operating conditions of a process unit during unit monitoring and to reach a better understanding of the impact of his manipulations under "interactive" mode.

Because of certain specific requirements associated with these displays, a separate screen may have to be used eventually. This is not, however, a design objective and to the maximum extent possible, the requirement of maintaining total flexibility in using any available screen should remain the basic design criterion.

In order for this type of displays to be effective, the process flow diagrams should be stripped down to a minimum level of detail and be organized in a logical sequence. In addition, the display should contain enough information as to become self sufficient-at least to the maximum extent possible. Figure 4 illustrates an example of simplified P&ID used for reference by console operators and gives an idea of the degree of detail that may be required.

- Display Requirements

  The basic requirements for this type of display format are summarized below:

  - The display should be fully interactive, i.e., the operator should be able to initiate an interactive mode from this display alone.

  - The process input for all tags should be updated on the screen, next to the instrument symbol.

  - Selection of control tags for interactive mode should be preferably via light-pen or similar techniques.

  - A selected control tag should be displayed on a dedicated sector of the screen (work area) for parameter manipulation (set point, mode, output etc.). These changes should be carried-out from the associated keyboard.

  - Textual, symbolic and color coding should be used to the maximum extent possible to minimize screen overloading. In general:

    + The loop function (Temperature, Pressure, Flow, etc.) should be identified by the shape representing the instrument.

    + The tag I.D. should be simplified (unit code can be shown with the title of the display; loop function can be identified by the shape).

    + Color coding should be used to:

      - reinforce symbolic coding
      - indicate status of equipment (e.g., in service; out of service)
      - indicate abnormal conditions (including alarm status)

    + Blinking should be associated with alarm conditions or equipment malfunctions.

  - The library of available pages should be preferentially addressable via two or more successive hierarchical overview levels of increased detail, displayed in schematic block diagram format. Passage from an overview level to the desired schematic diagram should be again via light-pen selection or similar technique.

  - Alternative methods of selection are not excluded, such as via the pushbutton matrix in combination with successive levels of area/unit directories displayed on the screen, or via display association.

    In any case, each individual page of the library of schematic process diagrams should also be addressable from the console keyboard, entering the appropriate code.

- **Building and maintenance**

  It is recognized that building such display formats as well as maintaining them up-to-date to reflect various alterations could represent a significant effort to the user.

  In general, building the schematics from a keyboard via a user defined library of shapes, is rather difficult compared to techniques where the schematic is drafted and automatically digitized and stored in bulk memory. On the other hand, the ability to alter the schematics from a keyboard appears to offer advantages in terms of maintaining them up-to-date. Consideration, therefore, should be given to a combination of both methods.

- **Other graphic displays**

  In general, the user should have the capability of building different types of graphic displays such as block diagrams of control schemes, special bar graph profiles or other formats he might wish to experiment with.

Pushbutton Matrix

The multipurpose role of this addressing tool requires that its design is given special consideration. The general requirement is for a flexible and self-adaptable addressing system, having the purpose to simplify the console procedures from an operator's point of view. As pointed out before, it should not be viewed as a simple matrix of pushbuttons but as a matrix of elements capable of adapting automatically to the display format being presented on the screen. This self adaptation should be in terms of active/inactive status, illumination, textual coding (labels) and color.

Summary of Basic Requirements

- The operator console, as defined in this specification, is an integrated work station based on a modular and expandable system supporting a number of identical display/manipulative sets. The design objective is that each set, consisting of a screen and entry keyboard, should be able to support any type of display format. On the other hand, the ability to assign any screen to specific type of display formats should be provided.

- To the maximum extent possible, call-up procedures and passage from one hierarchical level of display to the next should be by single action. Passage from one display to another at the same level of display hierarchy should be simple and fast and should also be based on display association.

- For a console intended to be used as the sole operator interface, the reliability aspects of the design assume primary importance and should address the following:

- Interactive failure modes, graceful degradation and fast recovery
- Communication with the outside world
- Communication between components of the console

A total failure of the operator interface is unacceptable.

- The display and address capability of each console should be consistent with the total data base of the control system so that any process area can be assigned to any console.

Additional Considerations

The design of a console should also take into account the following considerations.

- Ease of maintenance
- Auxiliary space requirements for temporary storage of manuals, logs, etc.
- Protection of work area against accidential spills (soft drinks, coffee)

The design of the keyboard should be kept uncluttered and organized such that seldom used keys are kept separate from often used ones. The keyboard must be of rugged construction and keys should withstand rough treatment.

The screens should be, in general, protected against glare but this protection should be designed such that:

- Does not interfere with the readability of displayed information or the screen interaction with special addressing tools, such as light-pens.
- Matches the conditions of the control room interior illumination. The control room illumination should be preferably adjustable from the console (ability to dim the lights, turn on/off spotlights, etc.).

Engineer's Console

- The engineer's console is considered a special purpose tool that should access the entire data base and should be used for:
  - File building
  - Setting-up or modifying control strategies
  - Testing the behavior of new control schemes
- This console should be, in general, a portable device able to communicate with the high speed bus over a distance of 300 ft (without modems). As a minimum, it should consist of a keyboard and video screen, but in general does not have to be a self-supported, "smart" terminal. It should rather be considered as a "shared" peripheral of the distributed computer system.

- The reliability requirements should be similar to those listed for remotely installed consoles.

- It is envisioned that several of these terminals (perhaps as many as six or seven) will be used throughout the system.

- The engineer's console should be totally independent from the operator's console. It may eventually be used as the special purpose plug-in terminal to configure the operator's console data base but this should not be considered a must. In general, however, it should be able to access the operator's console data base, directly or indirectly over the high speed communication bus.

- As a minimum the engineers console should be able to display the basic formats of the operator's console. It is not, however, required that this console supports the interactive alarm features, special addressing tools and cross-reference features of display association.

RRM:smr
June, 1977

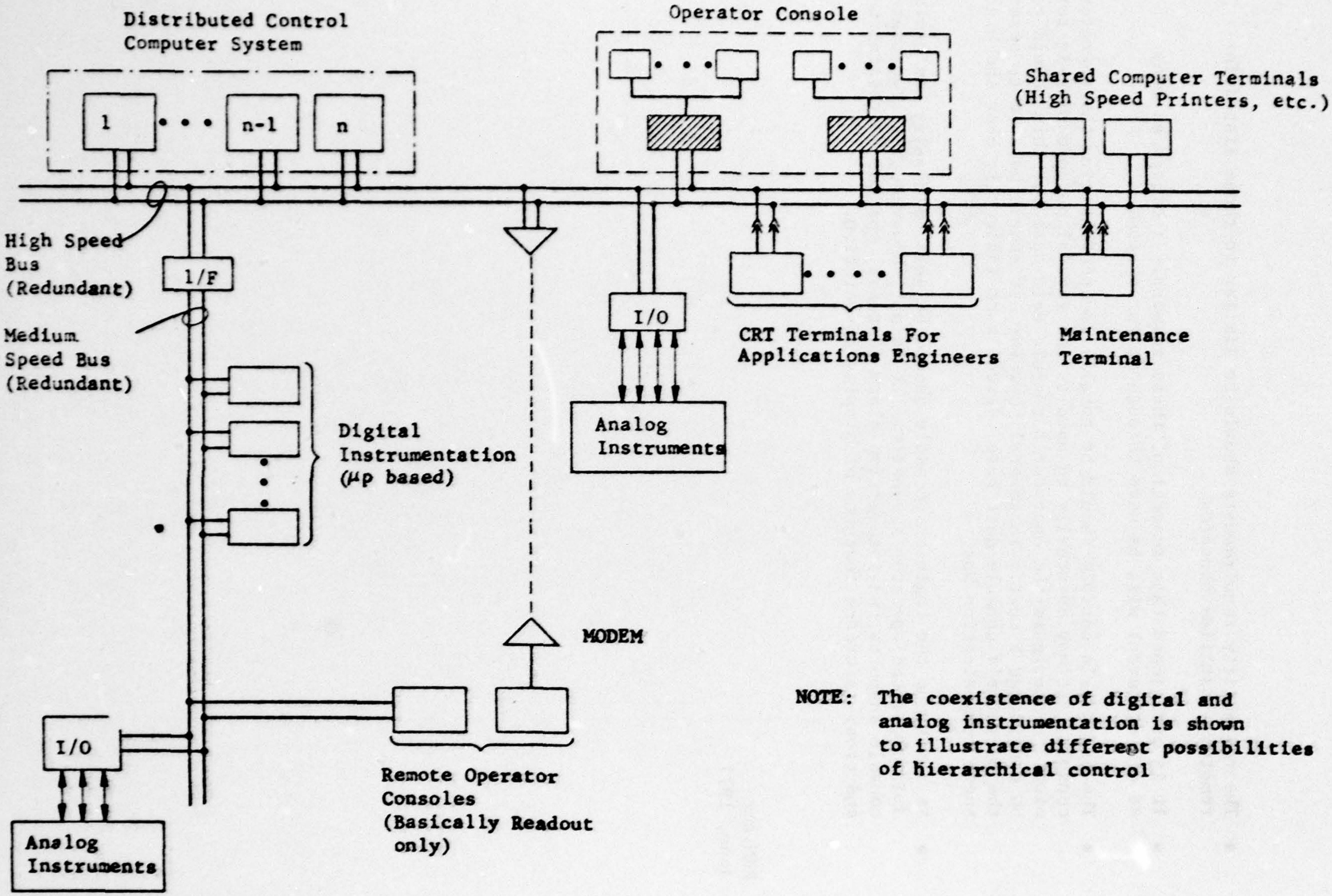
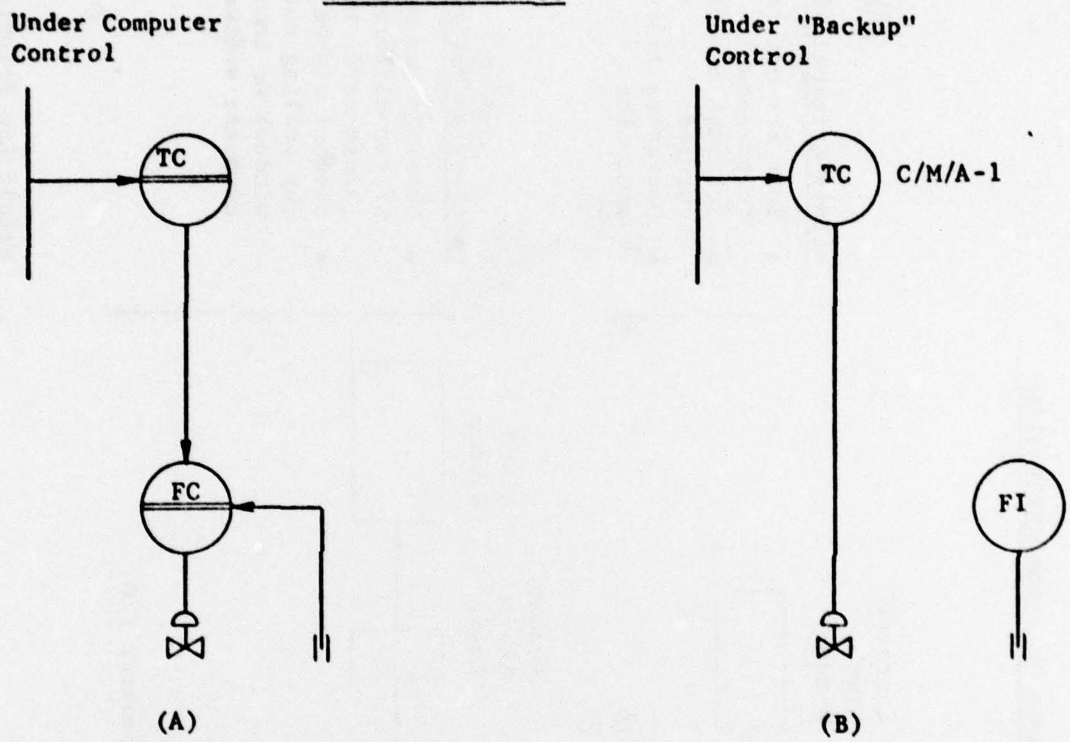


FIGURE 1

SCHEMATIC CONFIGURATION OF DISTRIBUTED/HIERARCHICAL CONTROL SYSTEM

CONTROL STRATEGY



CONTROL (A):
- Temperature cascading flow
- Flow input may be linearized and corrected for temperature and pressure

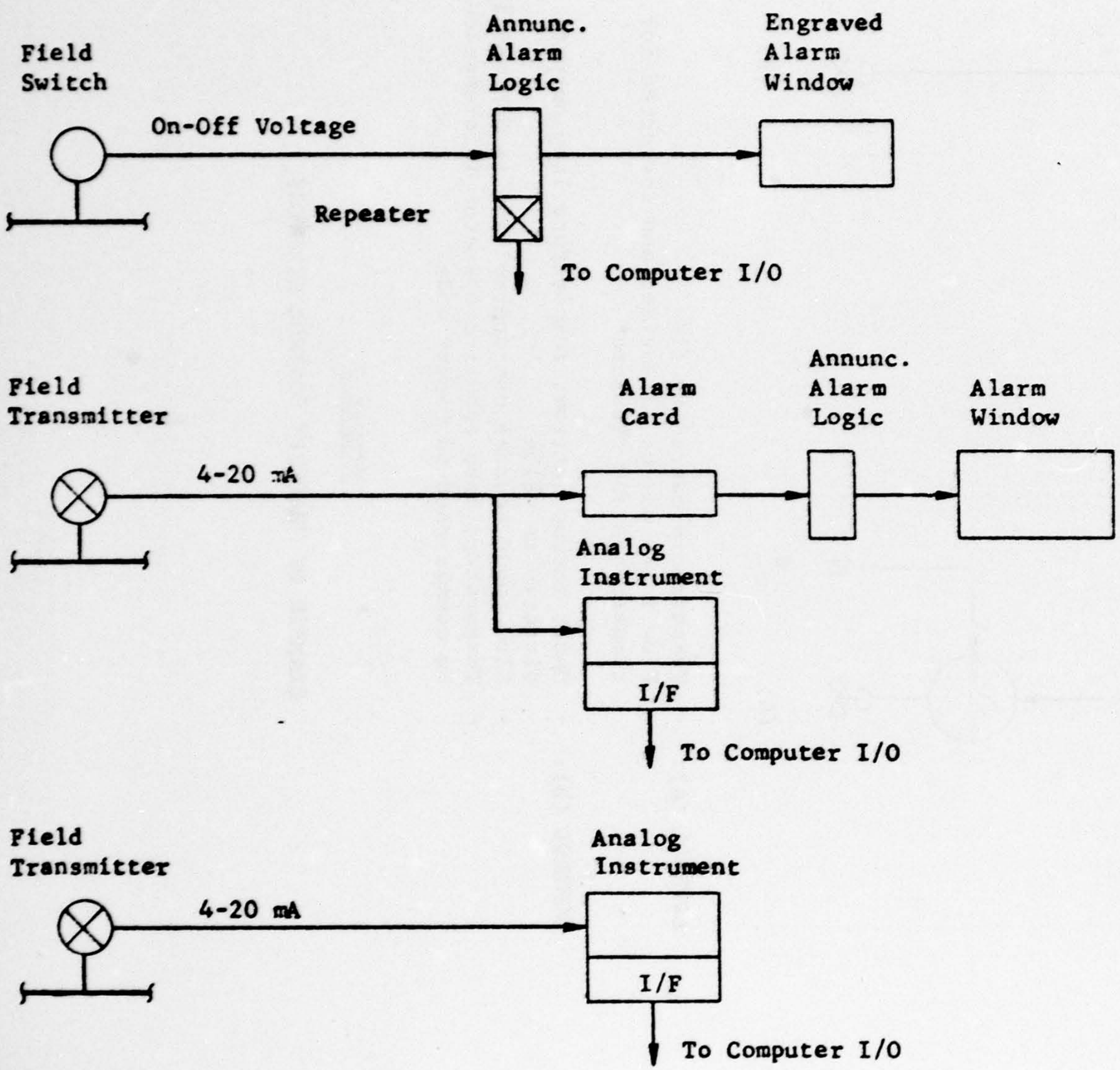
CONTROL (B):
- Under backup control, temperature loop outputs directly to valve.
- Flow input becomes non-control and is not conditioned
- Temperature loop requires operator intervention to be commissoned in "auto" mode.

FIGURE 2

EXAMPLE OF "BACKUP" CONTROL STRATEGY

FIGURE 3

CURRENT METHODS OF ALARM GENERATION



REMARKS

Mainly Critical Alarms

- Some are not repeated to computer
- Settings cannot be changed
- Operators always pay attention

Mainly High Priority Alarms

- Settings can be changed at two different locations (alarm card, computer files)
- Several alarms disabled (by pulling out alarm window) or inhibited at computer software

Mainly Low Priority Alarms

- Majority of operators tend to ignore.

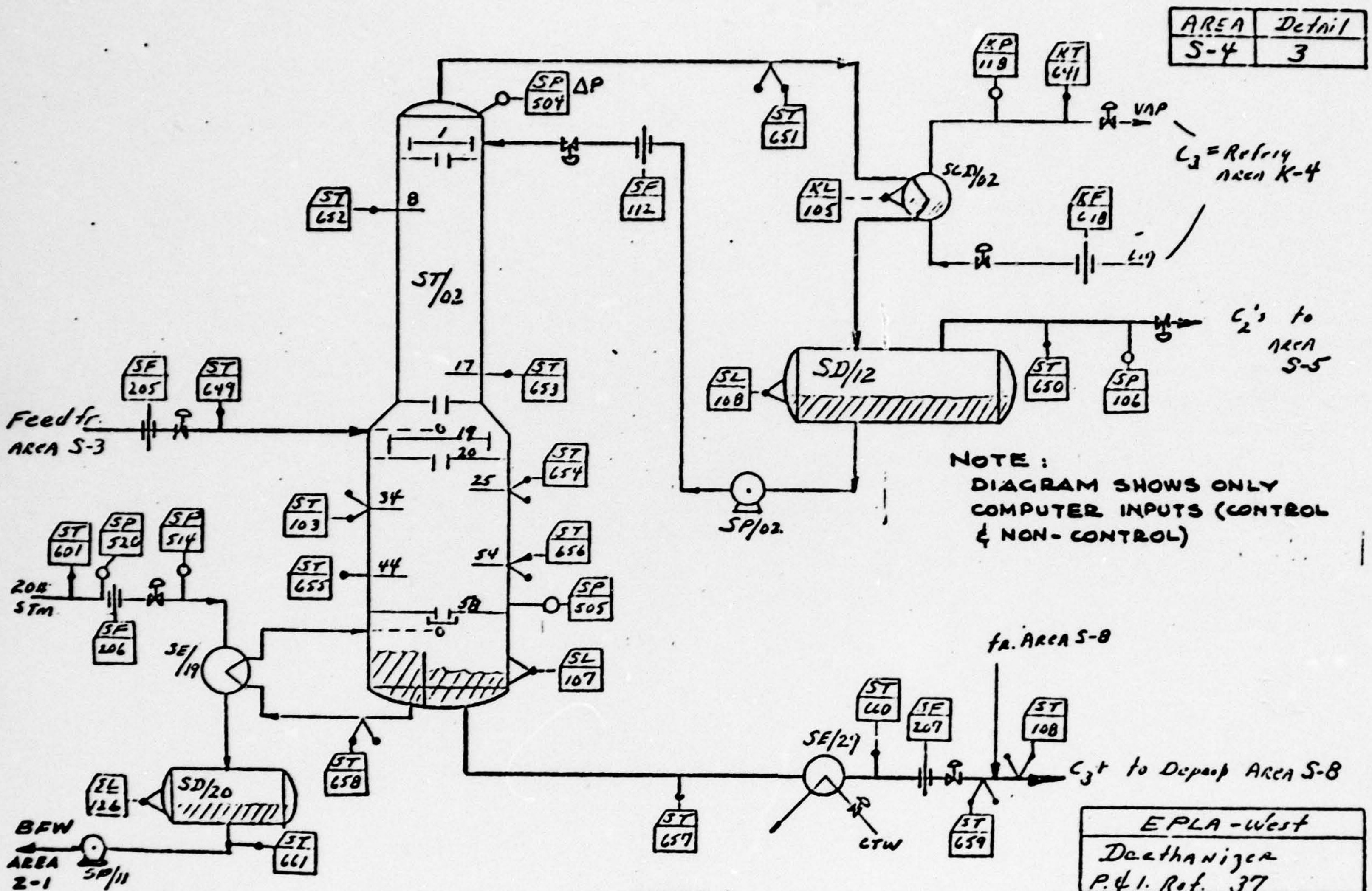


FIGURE 4

SIMPLIFIED SCHEMATIC PROCESS DIAGRAM

PURDUE EUROPE

European Regional Organization
of the International Purdue Workshop on
Industrial Computer Systems

TC 6
Technical Committee on
Man-Machine Communications

AUTOMATION AND MAN-MACHINE COMMUNICATIONS

AIMS AND ACTIVITIES OF TC6

JUNE 1977

AUTOMATION AND MAN/MACHINE COMMUNICATIONS
AIMS AND ACTIVITIES OF TC6

Trends in the design of industrial computer systems

There has been a gradual development in high-level automation of industrial production processes, starting in electrical power generation, petroleum and chemical sectors and spreading to, amongst others, the steel industry, assembly operations and public transport.

During the last few years, however, developments within the scope of industrial computing exercise a considerable influence. A variety of devices, such as microprocessors and visual display units, are now available for industrial and non-industrial use. Their costs, compared to the cost of labour, have become so low that widespread application for enhancing productivity is unavoidable.

This dramatic development of technology has created a situation in which automation systems are radically different for new plants. New designs are being introduces without a thorough knowledge based on prior experience. This is clearly apparent in the application of multiple cathode-ray tube displays for process supervision and the use of distributed computing networks. A fundamental gap is emerging between the revolutionary rate of change in machine technology, and the incorporation of human factors. These encompass the existing knowledge about human skills, proven methods for analysing human performance and the understanding of human attitudes towards automation.

The problem is aggravated by the upgrading of plants and the increasing complexity of processes owing to energy and material recycling. Narrowing profitability margins put a heavy emphasis on production within tolerance limits and on coping effectively with abnormal conditions. The safety aspect of plant operation is also gaining importance, due to legislation in an age of growing public awareness of and concern with the environment in the widest sense.

The operator, whose main task is now to supervise process operation, has a key role in this development. His operational practices and procedures are strongly influenced by the design of the man-machine interface. Careful design of the man-machine interface is therefore crucial to the success of the operator-process system. But conflict can arise between the natural tendency to assign as much responsibility and control to machines as technologically possible against the need to enable the human operator to maintain meaning in an understanding of his job - to provide for his interpretive skills in decision making and for his ultimate responsibility for plant control and for plant safety.

These various factors mentioned above demonstrate the need for:

- design procedures which contribute to acceptable tasks for human operators

- techniques in designing interfaces to match human abilities

- utilisation of information describing human performance.

The following means will be used:

- Survey of successes and failures in the design of MMIF;

- Determination of the parameters and features critical for good MMC design and the incorporation of these in a series of guideline documents, for example on the methodology and techniques for design, implementation, and for training;

- Cooperation with other TCs in matters concerning MMC at all levels of system design, implementation and use;

- Dissemination of the results, encouraging discussions with and acceptance by users, national and international groups and standardisation bodies.

Past and current activities are in three areas:

- Preparation of the results from a questionnaire which has been circulated to various European industries to assess experience and attitudes to problems in interface design, from basic philosophy to realisation. An analysis of the common problem areas will be used to formulate a working remit based on users' needs;

- Preparation, in cooperation with the parallel American TC, of a set of guidelines which will offer procedural and substantive recommendations;

- Establishment of contacts with industry, national laboratories and universities engaged in the field of MMC throughout Europe.

In some respects, human performance is poorer than that of a machine, e.g. in making consistent decisions or calculations. But design for complementary functions of human operators and equipment can exploit the best features of both. In no industry is this more apparent than in that of nuclear energy where risk analysis, safety reporting and incident analysis are becoming increasingly important. Many studies have been undertaken investigating human fallibility and means of allowing for it through good design practice. These considerations are equally appropriate to the many other process industries for systems design to be safe, acceptable in operational requirements for humans, and to be cost effective.

The Role of the TC on Man-Machine Communications

The Technical Committee consists of European members from a range of disciplines: system engineering, software design, and human factors. There are common interests with the other Purdue Europe TCs and links with the parallel TCs in America and Japan under the Purdue International Workshop. Through its composition and formal relationships, the Technical Committee is thus in a unique position in Europe to investigate and influence man-machine communications (MMC) in the context of developments across the field of industrial computing.

A joint activity with the parallel American TC is in progress, in order to develop an approach based on world-wide experience. Prior to presenting guidelines to professional groups and standardisation bodies, there is a need to establish a common frame of reference and common terminology for both vendors and users to improve specifications.

The aim of the Technical Committee is to improve the incorporation of user characteristics in design, through the collection, evaluation and dissemination of information and experience.

Plans and Future Activities

Plans

Membership of the Technical Committee is on a voluntary basis and consequently the acitivies are limited by the members' normal working commitments. This results in slower progress than desirable, with the risk that results and recommendations may be superseded by events. The alternative is to seek support in order to:

- improve communication on and to stimulate activities in research and development in MMC activity in Europe;

- facilitate surveys, visits and seminars;

- formalise co-operation with the CEC to increase interest and acceptance of these issues in European industry.

A clerical support and administration centre is essential to forming a focus for European MMC activity relevant to the industrial rather than the academic sector. Much of the university study in this field is industrially sponsored but resources are necessary for it to be presented in a wider context of applicability across industries. Similarly resources are needed if data and information held by various industries are to be made available by their staff in a suitable form.

The design of an information system should no longer be regarded as a naturally emergent occurrence. It must be planned and monitored throughout the design process. In time, with financial support, it will be desirable to set up an information service which can advise industry and which can act as an agency between industry, universities and other consultants.

Activities

Through contacts with research and development functions and with users, current practice in industry will be reviewed and will be related to the body of information available within the following areas:

- The practical and theoretical constraints of equipment used in interface design, and the software and procedures which may be used in determining its applications and flexibility;

- The information requirements of processes and their control and the impact of new techniques, e.g. predictive aids for control;

- The influence of human factors upon operational practice procedures.

In order to further a growing awareness of these issues and to promote active interest in their investigation, seminars will be organised for industrial users and for representatives of national bodies. The seminars will be used to ensure appropriate direction in the Technical Committee's work for the development of guidelines. The first seminar, proposed to be held in December 1977, is under discussion and will be dealing with the subject of how the human factors in MMC design may be incorporated into a design process at a practical level.

Development of a set of guideline documents will be the principal continuing activity. It is planned that their structure will be suitable for providing information to senior management and project personnel as appropriate. The guidelines will be presented as a hierarchy of procedures and information in consultation with the other Purdue Europe TCs to establish needs and priorities with feedback to be sought from national engineering societies.

It is the Committee's view that only in this participative style can a genuine understanding and commitment be fostered in the complex field of man-machine communications.