PHILCO
A SUBSIDIARY OF Ford Motor Company
COMPUTER DIVISION

ELECTRONIC DATA PROCESSING SYSTEMS

PHILCO 2000 PERT SYSTEM

TM 19

PHILCO 2000
PERT SYSTEM

February 1963

PHILCO CORPORATION

A SUBSIDIARY OF Ford Motor Company

Computer Division • 3900 Welsh Road

Willow Grove, Pennsylvania

TM—19A

PREFACE

The PERT Manual is a reference manual for the Philco 2000 PERT System; it is intended for persons having a general knowledge of PERT. Chapter I is a general review of PERT terms as they are used in this manual. The remainder of the manual describes specifically how to submit and run PERT problems on the Philco 2000.

This manual (TM-19A) incorporates all changes announced for the Philco 2000 PERT Manual TM-19, dated April 1962.

CONTENTS

CONTENTS (Cont'd)

INTRODUCTION

The Philco 2000 PERT System allows PERT computations to be performed on the Philco 2000. The following are special features of the Philco 2000 PERT System:

- Handles projects consisting of up to 7400 activities and 3700 events.
- Operates at extremely high speeds. PERT projects of 1000 activities have been run on the Philco 212 in 7 seconds, including input/output time.
- Permits the use of symbolic event names.
- Allows activities to be submitted in random order; the Philco 2000 PERT System resequences them and creates the project network.
- Provides the option of giving either one or three time estimates for each event.
- PERT input may be on a special PERT tape or on the general operating system input tape.
- Allows for buffered input if the special PERT input tape is used.
- Program data for any project is contained in memory, thus eliminating the need for intermediate tapes.
- Includes a complete set of integrated prognostic, diagnostic, and service routines.
- Incorporates automatic machine methods of testing PERT networks for consistency and legitimacy of input data.
- Computes variance to determine the probability of meeting scheduled dates.
- The system maintains a complete history tape which permits subsequent runs, and easy project modification and updating.
- Input for successive runs may be on punched cards or on the history tape.

- Allows changes to be made to initial program data through the use of a new ID Card, without destroying the original data.

- Does not require a knowledge of any computer or of programming logic.

I. GENERAL DESCRIPTION

PERT is a process by which, after designating the tasks comprising a project, establishing the interrelations between the various tasks, and specifying the amount of time required to complete each task, the crucial tasks of the project may be determined. Any delay in accomplishing these crucial tasks will delay the final completion of the entire project.

PERT TERMS

In describing PERT as prepared manually or as solved on the Philco 2000, the following common PERT terms are used:

- A *project* is defined as a network of activities and events.
- *Activities* are time consuming elements; tasks to be achieved.
- *Events* are the termini of activities. They designate either specific accomplishments or points at which the programs start. Activities are separated from each other by events which are used as activity identifiers.
- The *duration* of an activity is the time period required to complete an activity successfully.
- The *critical path* is the specific sequence of events which comprises the most rigorous time constraint in the accomplishment of the end event. If any event on the critical path is delayed beyond the expected date of accomplishment, the final completion date of the entire project can be expected to be delayed by the same amount of time.
- The *slack* of an event is defined as the time interval by which the completion of a certain event can be delayed without delaying the final completion date of the end event.

GRAPHICAL DESCRIPTION OF A PROJECT

A project plan can be represented by a flow diagram. Each activity of the project is depicted by an arrow. The inter-connection among the arrows shows the sequence and interrelationship among the events. The nodes of the graph correspond to the events of the project.

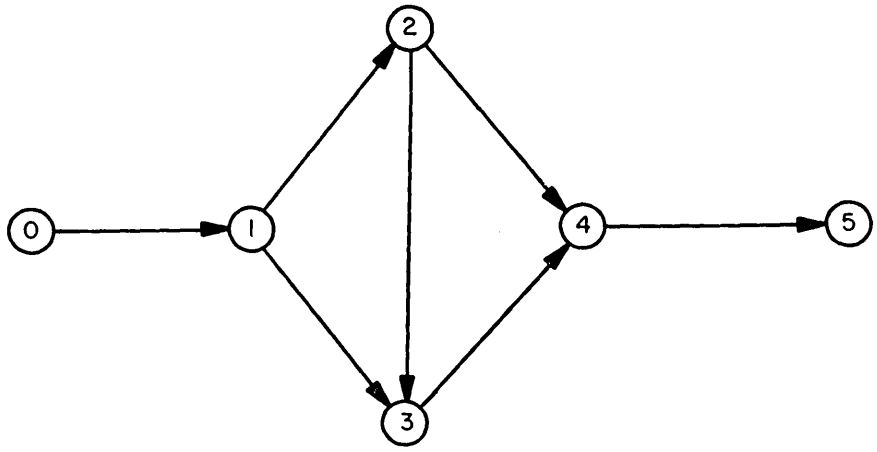

Figure 1. Sample PERT Project Flow Diagram

ESTABLISHING A NETWORK

The first step in constructing a project network is to list the activities that must take place in order to accomplish the end objective. The activities may be listed in random order and do not have to be sequenced.

The next step is to define the interrelation among the activities. Two events are then associated with each activity, a preceding and a succeeding event. These two events identify the activity and are called the predecessor and successor events of the activity. An activity cannot start until the predecessor event has occurred. Similarly, an event cannot occur until all the activities that lead to it have been completed.

The last step is to estimate each activity's duration. Three time estimates should be made for each activity - the *optimistic*, *most likely* and *pessimistic* time estimates. PERT then calculates the duration, utilizing the formula:

$$d_{ij} = \frac{a+4m+b}{6}$$

where:

d_{ij} = calculated duration of activity i,j

a = optimistic time estimate

m = most likely time estimate

b = pessimistic time estimate

Philco PERT can operate with the three time estimates described above, or with only one estimate, the most likely time.

DUMMY EVENTS AND ACTIVITIES

In order to construct the network correctly, it is sometimes necessary to add *dummy events* and *activities*. For instance, if two activities A and B begin and terminate with the same events i,j (see Figure 2), a dummy activity and a dummy event (k) should be added to prevent ambiguity. This activity has a duration zero.

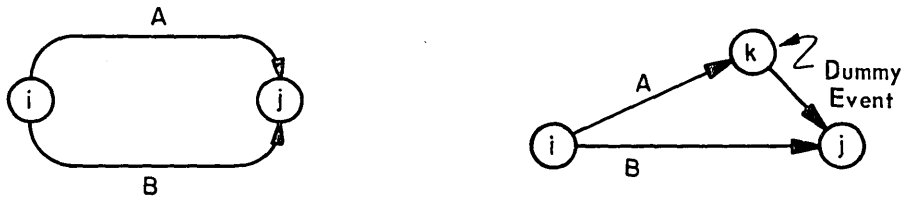

Figure 2. The Use of Dummy Events

It can also happen that activities P and Q are predecessors of R, while P is at the same time a predecessor of S; S, however, is independent of Q (see Figure 3). It is therefore necessary to posit a dummy activity D between P and R. The duration of D is zero.

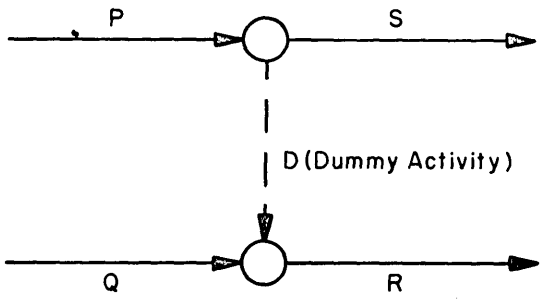

Figure 3. The Use of Dummy Activities

BEGINNING AND END EVENTS

It is necessary to introduce a unique event which initiates the project. This event is the only one which has no predecessor.

Similarly, a unique event should terminate the project. This is the only event that has no successor. If a project starts with several events (or ends with more than one event), one particular event should be selected as the initiating (or terminating) event, and the rest should succeed (or precede) it, utilizing dummy activities as connectors.

II. INITIAL INPUT FORMATS

The input data for the initial run consists of a list of the activities of the project, each activity being identified by its predecessor and successor events. For each activity there should be one or three time estimates. If there is a scheduled date for the completion of an activity, this date can be submitted with that activity.

In addition to the information given with each activity, the project starting date must be given; specifying a deadline date is optional.

ACTIVITY CARDS

Activity cards may be submitted in any order and should be prepared in the following format:

Column	Content
1	Card code. Should be 1 to indicate an initial run.
2-10	A nine-character predecessor event identifier. The characters can be any legitimate Philco 2000 characters,* provided they are not all spaces.
11-19	A nine-character successor identifier, restricted as above.
20-23	Optimistic time estimate.**
24-27	Most likely time estimate.**
28-31	Pessimistic time estimate.**

* Refer to Philco 2000 Code Combinations, TF 17.

** The time estimates are four digits each and are given in tenths of a week. The decimal point is not punched. If only one time estimate is given, it should be punched in columns 24-27. For example, a time estimate of 2 weeks is represented as 0020.

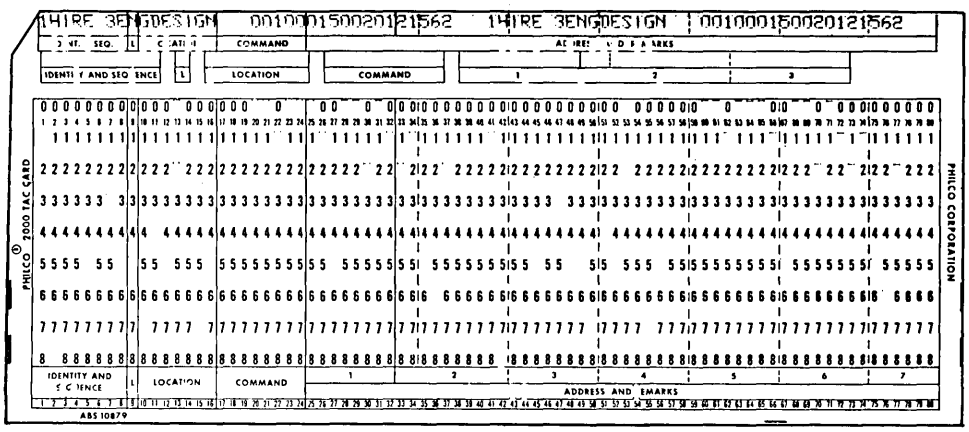

Figure 4. Sample Activity Card

Predecessor and Successor Events: HIRE 3ENG and DESIGN
Time Estimates: One Week, One and One-Half Weeks, and Two Weeks
Scheduled Date: December 15, 1962

CONTROL CARDS

Project Card

A project card must head each project deck and is punched as follows:

Column	Content
1-7	PROJECT
8	I (Indicates initial run)
9-24	Project identification
25-30	Project's beginning date
33-38	Project's end date (optional)
41	1, if one time estimate is given; blank, if three time estimates are given.
49-53	CARDS, if the history is on cards; blank, if the history is on the history tape.
55-80	Remarks

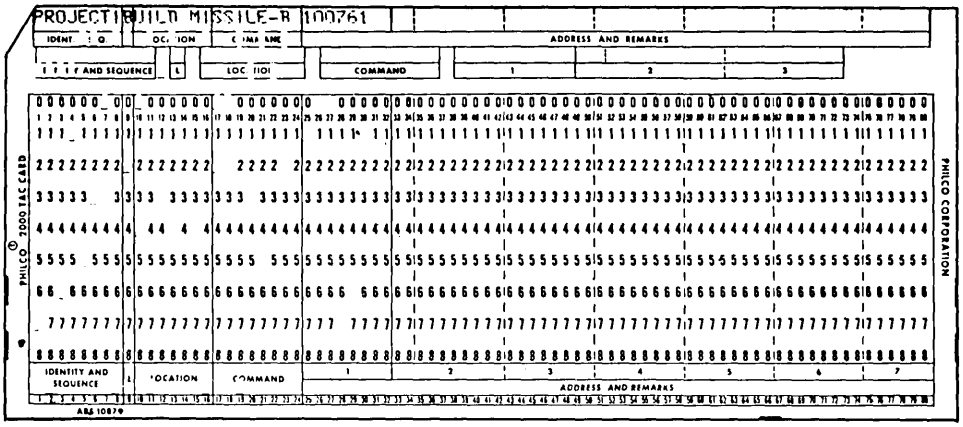

Figure 5. Sample Project Card, Showing:

Project Identification: BUILDMISSILE–B
Project's Beginning Date: October 7, 1961
Project's End Date: Not Given
Number of Time Estimates: Three
Location of Project's History: History Tape

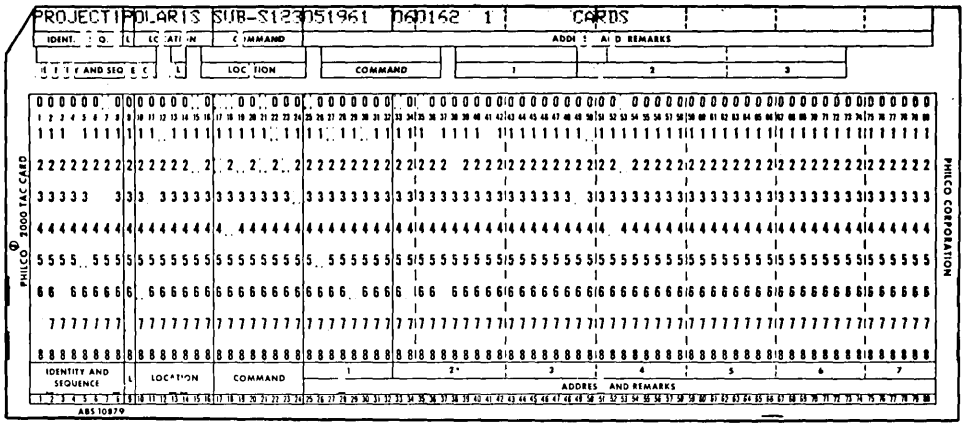

Figure 6. Sample Project Card, Showing:

Project Identification: POLARIS SUB–S123
Project's Beginning Date: May 19, 1961
Project's End Date: June 1, 1962
Number of Time Estimates: One
Location of Project's History: Cards

Ending Card

The last card of each project should have the character 9 punched in columns 1 and 41. The rest of the card is ignored. Note that the last project of an entire deck should also be followed by an END card (see page 31).

SAMPLE INPUT

```
PROJECT1SPACETRACKCAD    110161  010262
1START     PROGRAM  000000000000          1START     PROGRAM  000000000000
1START     POWERSUPP000000000000          1START     POWERSUPP000000000000
1START     WIRING   000000000000          1START     WIRING   000000000000
1START     CARDSA   000000000000          1START     CARDSA   000000000000
1START     CABLES   000000000000          1START     CABLES   000000000000
1POWERSUPPUNITTEST 000600100014          1POWERSUPPUNITTEST 000600100014
1LAYOUT    CABLES   000600100014          1LAYOUT    CABLES   000600100014
1START     LAYOUT   000000000000          1START     LAYOUT   000000000000
1CARDSA    CARDSB   004000600070          1CARDSA    CARDSB   004000600070
1CABLES    SPATESTEQ000800120024          1CABLES    SPATESTEQ000800120024
1LAYOUT    CABLESATT000600100012          1LAYOUT    CABLESATT000600100012
1WIRING    UNITTEST 001000140020          1WIRING    UNITTEST 001000140020
1CABLES    UNITTEST 002800420050          1CABLES    UNITTEST 002800420050
1CARDSB    UNITTEST 001000140020          1CARDSB    UNITTEST 001000140020
1SPATESTEQUNITTEST 002800400044          1SPATESTEQUNITTEST 002800400044
1PROGRAM   SYSTTEST 001000140018          1PROGRAM   SYSTTEST 001000140018
1CABLESATTSYSTTEST 004000600080          1CABLESATTSYSTTEST 004000600080
1UNITTEST  SYSTTEST 002000280042          1UNITTEST  SYSTTEST 002000280042
1SYSTTEST  INSTALL  002000300040          1SYSTTEST  INSTALL  002000300040
9                                         9
```

Figure 7. A Complete Input Card Deck for PROJECT SPACETRACKCAD

(Output from this project is shown in Figure 8, page 10.)

III. OUTPUT FORMATS

The output from a PERT program consists of a complete list of all the activities of the project including the dummy activities. Each activity is identified by the original predecessor and successor events of the activity card. The printed output for each activity is described below.

- The *expected date* is the earliest possible completion date of the activity. Calculation of this date is based on the beginning date of the project, the durations of the activities and the interrelations among the preceding activities.

- The *late date* is the latest possible date for completion of an activity without delaying the completion of the end event.

- The *slack* of an activity is the difference between the late date and the earliest expected date for completion of the activity. The activities with slack zero are the critical path activities. The slack is printed in weeks. For example, a slack of 2.5 is a slack of two and a half weeks.

- The *schedule date* is the date specified on the activity card in columns 32-37.

- The *actual completion date* is the date on which an activity was actually completed. (See Updating, page 21 and Input Data for Successive Runs, page 22.)

- The *duration* of an activity is the calculated value a+4m+b/6 if three time estimates were made. (See Establishing a Network, page 2.) If only one time estimate is made, the duration is equal to that time estimate.

- The program also calculates the *variance* of the duration. Utilizing the computed variance it is possible to estimate the probability of actually meeting scheduled dates.

PERT calculates the variance using the formula

$$V_{ij} = \left(\frac{b-a}{6}\right)^2$$

where:

V_{ij} = The variance of activity i,j

a = Optimistic time estimate

b = Pessimistic time estimate

SAMPLE OUTPUT

The standard output does not include the variance. The following is an illustration of standard PERT output.

PERT SYSTEM
SPACETRACKCAD

PAGE 1

ACTIVITY		COMP. DATE	EXP. DATE	LATE DATE	SCHED. DATE	SLACK	DURATION
START	PROGRAM		11/01/61	12/02/61		4.4	.0
START	POWERSUPP		11/01/61	11/15/61		1.9	.0
START	WIRING		11/01/61	11/12/61		1.5	.0
START	CARDSA		11/01/61	10/02/61		- 4.3	.0
START	LAYOUT		11/01/61	10/09/61		- 3.3	.0
START	CABLES		11/01/61	10/16/61		- 2.3	.0
PROGRAM	SYSTTEST		11/11/61	12/12/61		4.4	1.4
POWERSUPP	UNITTEST		11/08/61	11/22/61		1.9	1.0
WIRING	UNITTEST		11/11/61	11/22/61		1.5	1.4
CARDSA	CARDSB		12/12/61	11/12/61		- 4.3	5.8
LAYOUT	CABLES		11/08/61	10/16/61		- 3.3	1.0
LAYOUT	CABLESATT		11/08/61	10/31/61		- 1.2	1.0
CABLES	SPATESTEQ		11/17/61	10/25/61		- 3.3	1.3
CABLES	UNITTEST		12/07/61	11/22/61		- 2.2	4.1
CARDSB	UNITTEST		12/22/61	11/22/61		- 4.3	1.4
CABLESATT	SYSTTEST		12/20/61	12/12/61		- 1.2	6.0
SPATESTEQ	UNITTEST		12/15/61	11/22/61		- 3.3	3.9
UNITTEST	SYSTTEST		1/11/62	12/12/61		- 4.3	2.9
SYSTTEST	INSTALL		2/01/62	1/02/62		- 4.3	3.0

Figure 8. Output from PROJECT SPACETRACKCAD

(Input for this project is shown in Figure 7, page 8.)

IV. DIAGNOSTIC AND ERROR CORRECTING ROUTINES

INTRODUCTION

The PERT diagnostic program is a monitoring and correcting program which tests the input cards for legitimacy of the characters and consistency of the data. Processed data is analyzed for system errors such as internal loops within a project, concurrent activities, open ends, and other trouble spots.

It also prints out a complete description of the errors or possible errors and attempts to correct them. The programmed corrections are not always sufficient, but they enable continued processing and calculations that result in an initial output which provides at least a general indication of the status of the project.

These corrections also facilitate later updating and corrections. (See Updating, page 21 and Error Correction, page 24.) As a result, errors can be corrected simply by submitting correction cards, rather than by searching for bad cards in a large unsequenced deck, replacing them, and then reprocessing all the data.

The error output medium is the High-Speed Printer unless otherwise specified.

Descriptions of PERT error print-outs and diagnostic procedures follow.

BAD CARD

Description Columns 1-40 of the input card are not identical to columns 41-80, indicating a bad card-to-tape transmission and/or a keypunch error.

Print-out BAD CARD followed by the contents of the card in error. All 80 columns are printed.

Format BAD CARD XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Procedure The program processes the data in columns 1-40.

ILLEGAL TIME ESTIMATE

Description A time estimate is illegal if any of its characters are not numeric (space is an illegitimate character), or if the three time estimates do not satisfy the inequality $a \leq m \leq b$.

Print-out ILLEGAL TIME ESTIMATE followed by the contents of the activity card in error.

Format

```
ILLEGAL TIME ESTIMATE    XXXXXXXXXXXXXXXXXXXXX
```

Procedure PERT assigns the value zero to the duration and the value .01 to the variance.

CARD CODE ERROR

Description An illegitimate character is detected in column 1 of the activity card.

Print-out CODE followed by the contents of the card in error.

Format

```
CODE          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Procedure The program interprets the code as code 1 and continues the processing.

EVENT TABLE PACKED

Description The number of events of a project is larger than the maximum allowable number of events (see Appendix B).

Print-out EVENT TABLE PACKED followed by the contents of the card that caused the error or by the activity identifiers only.

Format

```
EVENT TABLE PACKED      XXXXXXXXXXXXXXXXXXX
                  or
CONCURRENT ACTIVITIES   EVENT TABLE PACKED I J
```

Procedure The event is ignored and the program completes the processing of the input data but does not perform the analysis and scheduling. The processed data is put on the history tape or on cards; adjustments should be made during successive runs. (See Chapter V, Successive Runs, page 21.)

Remarks The event table could be packed as a result of too many events in the input data, in which case the contents of the card containing the event that could not be entered into the event table is printed. If the event table is packed due to the need to add a dummy event (see Concurrent Activities, page 14), the identifying events of the concurrent activities are printed.

ACTIVITY TABLE PACKED

Description

The number of activities is larger than the maximum allowable number of activities (refer to Appendix B).

Print-out

ACTIVITY TABLE PACKED followed by the contents of the card or the identifying events of the activity that could not be entered into the activity table.

Formats

ACTIVITY TABLE PACKED	XXXXXXXXXXXXXXXXXXXXX
or	
CONCURRENT ACTIVITIES	ACTIVITY TABLE PACKED I J
or	
OPEN END	ACTIVITY TABLE PACKED HAS NO PREDECESSOR
or	
OPEN END	ACTIVITY TABLE PACKED I HAS NO SUCCESSOR

Procedure

The activity is ignored and the program completes the processing of the input data, but does not perform the analysis and scheduling. The processed data is stored on the history tape or on cards. Adjustments should be made during successive runs.

Remarks

The activity table could be packed as a result of too many activities in the input data. In this case, the card which contains the activity that could not be entered into the activity table is printed. If the activity which cannot be entered into the activity table is a programmed dummy activity, this information will be printed. (Refer to Formats above; Blank Identifier, page 14; and Concurrent Activities, page 14.)

ILLEGAL DATE

Description

An illegal date is detected on an activity card. The following are illegal dates:

- A date with a non-numeric character (e.g., spaces)
- A date that does not exist, such as 15/1/60
- A date in a year previous to the year or to the beginning of the project.

Print-out

ILLEGAL DATE followed by the contents of the card which contains an illegal date.

Format

ILLEGAL DATE XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Procedure The date is ignored. If the date error concerns the initial date of the project during an initial run, PERT processes the input and puts the data on the history tape or on cards, but does not proceed with the analysis and scheduling. If the date error concerns the initial date during a successive run, the program uses the initial date used in the previous run, and ignores the new starting date.

ACTIVITY NOT FOUND

Description An activity with a code different from 1 (new activity card) is not in the activity table (see Successive Runs, page 21).

Print-out ACTIVITY NOT FOUND followed by the contents of the activity card.

Format ACTIVITY NOT FOUND XXXXXXXXXXXXXXXXXXXXXXXXXX

Procedure The activity card is ignored.

BLANK IDENTIFIER

Description An identifier event which is all spaces (9 spaces) is detected.

Print-out BLANK IDENTIFIER followed by the contents of the card.

Format BLANK IDENTIFIER XXXXXXXXXXXXXXXXXXXXXXXXXX

Procedure The identifier is changed to ????????? and the program continues processing.

CONCURRENT ACTIVITIES

Description Concurrent activities are activities with identical predecessor and successor identifiers.

Print-out CONCURRENT ACTIVITIES followed by the predecessor and successor identifiers of the two activities which are formed by dividing one of the concurrent activities (see Procedure). The first activity printed has the same duration and variance as the original undivided activity. The second activity is an added dummy activity. If a dummy activity cannot be added, a further explanation is printed.

Formats

CONCURRENT ACTIVITIES I DUMMYn DUMMYn J

or

CONCURRENT ACTIVITIES EVENT TABLE PACKED I J

or

CONCURRENT ACTIVITIES ACTIVITY TABLE PACKED I J

Procedure

One of the concurrent activities I, J is broken into two activities I, DUMMYn and DUMMYn, J. I, DUMMYn has the same duration and variance as the original activity I, J; DUMMYn, J is a dummy activity with the duration zero and variance .01.

The dummy event DUMMYn is added to the event table and the dummy activity is added to the activity table. The addition of events and activities may result in packing the activity table or event table. Should that happen, the dummy activity and dummy event are not added and an appropriate error indication is printed.

Remarks

PERT sequences the dummy events added to the project. The first added event is denoted by DUMMY1, the second by DUMMY2 and the nth by DUMMYn. (Refer to Card Code Error, page 12; Event Table Packed, page 12; and Error Correction, page 24.)

OPEN END

Description

An event without a successor event which is not the end event, or an event without a predecessor event which is not the initial event, indicates an open end.

Print-out

OPEN END followed by the identifiers of the added dummy activity.

Formats

```
OPEN END    B  I
     or
OPEN END    I  E
     or
OPEN END    ACTIVITY TABLE PACKED  I HAS NO
            PREDECESSOR
     or
OPEN END    ACTIVITY TABLE PACKED  I HAS NO
            SUCCESSOR
```

Procedure

The program adds a dummy activity B, I in the case of an event I without a predecessor (B being the initial event). It adds a dummy activity I, E in the case of an event without a successor (E is the end event). The dummy activities have durations zero and variances .01.

MISSING COMPLETION DATE

Description

Upon the submission of the completion date of an activity i, j, PERT detects that the completion date of some activity k, i terminating with i is missing (see Updating, page 21).

Print-out MISSING COMPLETION DATE followed by the contents of the completed activity card.

Format MISSING COMPLETION DATE XXXXXXXXXXXXXXXXXXXXX

Procedure The date is stored in a date list, but the processing associated with the completion date is not done.

Remarks Adjustment has to be made during successive runs. The printed message does not indicate which completion date is missing.

COMPLETION DATE ENTERED TWICE

Description Upon the submission of the completion date of an activity i, j, PERT detects that the completion date of some other activity terminating with j was entered twice. (See Updating, page 21 and Error Correction, page 24.)

Print-out COMPLETION DATE ENTERED TWICE followed by the contents of the completed activity card.

Format COMPLETION DATE ENTERED TWICE XXXXXXXXXXXXXXX

Procedure The completion date is stored in a date list, but the processing associated with the completion date of an acitivity is not done. Adjustment has to be made during successive runs.

Remarks The printed message does not indicate which completion date was entered twice.

LOOP ERROR

Description A loop is a set of n consecutive events such as a_i, $a_i + 1 \ldots a_n$ in which a_i precedes $a_i + 1$ and a_n precedes a_i. Figure 9 illustrates a loop formed by events C, D and E. Any loop of events is considered by PERT to be an error.

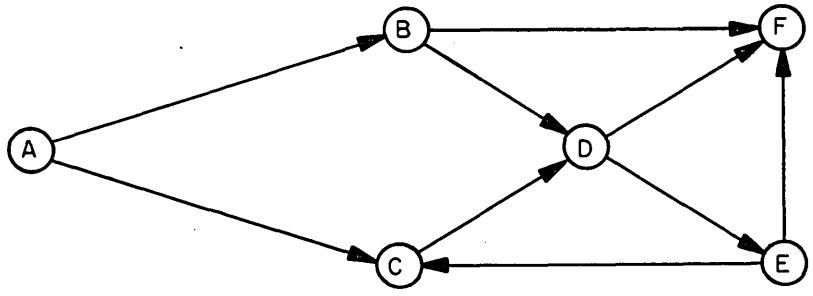

Figure 9. Sample Loop

Print-out — A list of all elements of the loop and the elements attached to the loop.

Type-out — LOOP ERROR will be typed on the Console Typewriter.

Printer Format — $I_1\ I_2\ I_3$ .....

$I_n\ I_{n+1}$ .....

Procedure — A loop error is a serious error which the program cannot correct. When a loop error is detected, the data is put on the history tape or on cards without any further processing.

CONTROL CARD ERROR

1. Description — An illegal character is detected in column 8 of the PROJECT card.

Procedure — The project is ignored and a search is made for the next control card.

2. Description — BININPUT card is not preceded by a PROJECT card.

Procedure — The contents of the BININPUT card is typed and a search is made for the next control card.

3. Description — A character other than a blank or 1 is detected in column 41 of a PROJECT card.

Procedure — One time estimate is assumed and the project is processed.

4. Description — An illegal tape unit is specified in columns 57-58 of the PERTSERV card, i.e., a unit other than 0-15.

Procedure — The PERTSERV card is ignored and a search is made for the next control card.

5. Description — An illegal PERTSERV routine is specified in columns 17-26.

Procedure — The PERTSERV card is ignored and the next control card is searched for.

Remarks — All CONTROL CARD ERROR type-outs occur immediately following the card in error.

CONTROL CARD INTERCEPTED

Description — A PROJECT, PERTSERV or END card is detected among the cards of another project while processing, i.e., before a card with code 9 is found. (See Input Preparation and Operating Instructions, Chapter VII, page 31; also see PERTSERV, page 25.)

Type-out CONTROL CARD INTERCEPTED followed by the contents of the detected control card is typed by the Console Typewriter.

Formats

PROJECT CARD INTERCEPTED XXXXXXXXXXXXXXXXX

PERSERV CARD INTERCEPTED XXXXXXXXXXXXXXXXX

END CARD INTERCEPTED XXXXXXXXXXXXXXXXX

Procedure Action is initiated according to which control card (PROJECT, PERTSERV or END) is intercepted.

PROJECT NOT ON HISTORY TAPE

Description A project searched for by PERT or PERTSERV is not found on the history tape.

Type-out PROJECT NOT FOUND followed by the contents of the control card.

Format PROJECT NOT FOUND XXXXXXXXXXXXXXXXXXXXXX

Procedure PERT will search for the next control card.

EXAMPLE OF ERROR PROCEDURES

Description Figures 10, 11 and 12 show an input deck containing errors, the PERT print-out indicating these errors, and the PERT output with these errors corrected, respectively.

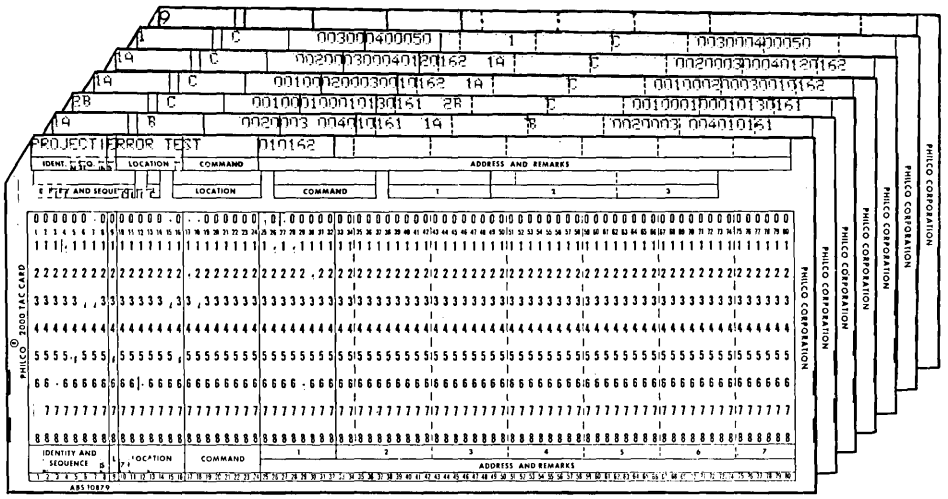

Figure 10. Sample PERT Input Deck Containing Errors

This input deck contains the following PERT input errors:

- BAD CARDS. Punching for the right-hand side of the cards was begun in column 40 rather than in column 41.
- ILLEGAL TIME ESTIMATE. The optimistic time estimate for activity AB is illegal - it contains a non-numeric character (a space).
- ILLEGAL DATE. The date of activity AB is illegal since it is earlier than the beginning date of the project. The date of activity BC specifies the thirteenth month, and is therefore illegal.
- CODE. Activity BC contains Code 2 which is illegal for an initial run.
- BLANK IDENTIFIER. The last activity has an identifier which is all spaces (the predecessor event).
- CONCURRENT ACTIVITIES. Two activities are labeled AC.

PERT SYSTEM
ERROR TEST

PAGE 1

BAD CARD	2B	C	001000100010130161	2B	C	001000100010130161
BAD CARD	1A	C	001000200030010162	1A	C	001000200030010162
BAD CARD	1A	C	002000300040120162	1A	C	002000300040120162
BAD CARD	1	C	003000400050	1	C	003000400050
BAD CARD	9					
ILLEGAL TIME ESTIMATE	1A	B	0020003 004010161	1A	B	0020003 004010161
ILLEGAL DATE	1A	B	0020003 004010161	1A	B	0020003 004010161
CODE	2B	C	001000100010130161	2B	C	001000100010130161
ILLEGAL DATE	2B	C	001000100010130161	2B	C	001000100010130161
BLANK IDENTIFIER	1	C	003000400050	1	C	003000400050
CONCURRENT ACTIVITY A		DUMMY 1	DUMMY 1C			
OPEN END A	?????????					

Figure 11. PERT Error Print–Out of the Input Errors Shown in Figure 10, page 19

PAGE 2

PERT SYSTEM
ERROR TEST

ACTIVITY		COMP. DATE	EXP. DATE	LATE DATE	SCHED. DATE	SLACK	DURATION
A	?????????		1/01/62	1/01/62		0	.0
A	B		1/01/62	1/22/62		3.0	.0
A	DUMMY 1		1/15/62	1/29/62	1/01/62	2.0	2.0
A	C		1/22/62	1/29/62	12/01/62	1.0	3.0
?????????	C		1/29/62	1/29/62		0	4.0
B	C		1/08/62	1/29/62		3.0	1.0
DUMMY 1	C		1/15/62	1/29/62		2.0	.0

Figure 12. Final PERT Output with the above Input Errors Corrected

V. SUCCESSIVE RUNS

THE HISTORY TAPE

The PERT System maintains a record of projects either on magnetic tape (referred to as the history tape) or on binary cards (referred to as cards). Data for each project on the history tape or cards consists of an identification block followed by four tables in the order listed below.

When preparing a new history tape, a block of Z's must be written at the beginning of the tape. The last project on the history tape is followed by a block of Z's.

Table	Content
Activity Table	All project activities including the predecessor and successor activity identifiers and the duration and variance of each activity. In this table, the predecessor and successor events are identified by their sequence numbers rather than by their alphanumeric symbols.
Event Table	All events in the project including the alphanumeric name and the sequence number of each event and the number of incomplete activities terminating with the event.
Completed Events Table	The completion dates of all completed events. This table is put on tape only during successive runs; it is always put on cards.
Scheduled Date Table	All scheduled and completion dates of the project activities.

UPDATING

After an initial project run, adjustments in the project network may necessitate the addition or deletion of activities, changes in time estimates, and adjustments in the initial date or in the project's deadline. Any errors encountered by the diagnostic program will also require corrections.

Once the project starts, periodic runs may be necessary since every time an activity is completed, the date of its completion must be submitted. The completion date of an activity should not be entered more than once.

When the history tape is maintained, the obsolete history is deleted from the history tape and the new history is added, unless a NEW ID card follows the project card of the deck. (Refer to A NEW ID Card, page 23.)

INPUT DATA FOR SUCCESSIVE RUNS

When the history is maintained on cards, the order of cards for a successive run is as follows:

The Project Card
A PERTSERV BININPUT Card
PERT Binary History Cards
Activity Cards
An Ending Card (Code 9)

When the history is on the history tape, the PERTSERV BININPUT card and the binary history cards are omitted.

Activity Cards

The activity cards for successive runs have exactly the same fields as those for an initiial run, i.e., Columns 2-10 and 11-19, are the activity predecessor and successor events, respectively; Columns 20-23, 24-27, and 28-31 are the three time estimates; and Columns 32-37 contain the date (if any is submitted) of either the actual completion date or the scheduled date. On cards for successive runs, columns 41-80 must be a duplicate of columns 1-40.

The major difference between initial and successive activity cards is in the card code (Column 1). The card code indicates the type of changes to be made and may be any number between 1-5.

Code 1: Indicates a new activity to be added to the program.

Code 2: Indicates a change in the time estimates. If, in addition to a change in the time estimates, a new scheduled completion date is required, it may be indicated in the date field of a Code 2 card. However, for change of date only, Code 3 must be used.

Code 3: Indicates a change in the scheduled date. The date field is the only field processed on Code 3 cards. A legal date in the date field will be interpreted as a new scheduled date to be inserted, while a zero in the date field will be interpreted as a cancellation of the scheduled date of the activity.

Code 4: Indicates a completed activity. The date field is the only field processed on Code 4 cards. The date will be stored by the program as the completion date of the activity.

Code 5: Indicates a deleted activity. The activity will be deleted from the activity table.

Control Cards

The deck of cards for a successive run, as for an intial run, should be headed by a PROJECT card. The last card of the project should be a code 9 card as in the initial run. A PROJECT card could be followed by a new ID card.

The deck of cards for a successive run, as for an initial run, should be headed by a PROJECT card. The format of the PROJECT card is as follows.

Column	Content
1-7	PROJECT
8	S, indicating the history is on the history tape C, indicating the history is on cards.
9-24	The project ID - This should be the same as that of the initial run of the project.
25-30	The initial date of the project.
33-38	Ending date. If this field is blank, there is no change in the deadline. A legal date replaces the formerly used date; an illegal date is ignored.
39-80	Ignored.

A New ID Card

When making changes, it might be desirable to keep both the changed and unchanged history on tape. This could, for example, be desirable for comparison purposes. Should this be the case, a NEW ID card must follow the PROJECT card. The updated history will then be on the history tape under the new identification. The NEW ID card feature may be used only when the project's history is maintained on the history tape. An illustration of a NEW ID card for a project is shown in Figure 13.

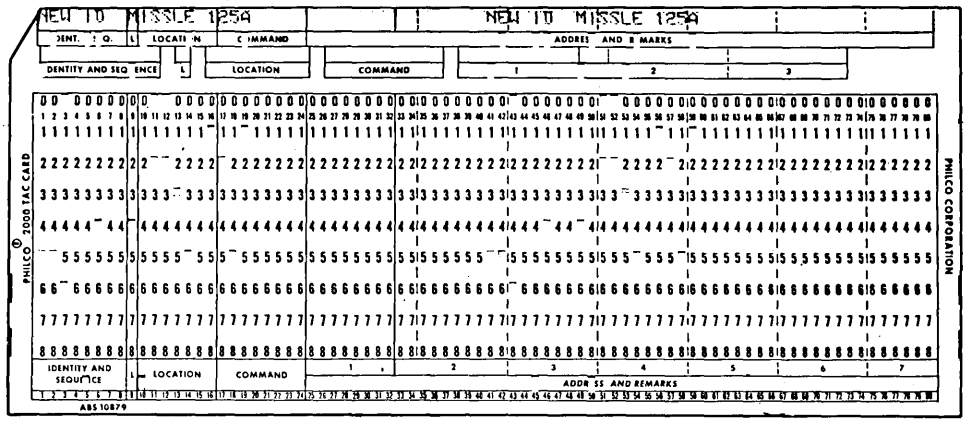

Figure 13. New ID Card Sample

The NEW ID card format is:

Column	Content
1-7	NEW ID
9-24	New project identification
41-47	NEW ID (same as 1-7)
49-64	New project identification (same as 9-24)

The rest of the card is ignored.

ERROR CORRECTION

During successive runs the errors of previous runs should be corrected. If the errors occurred in the time estimates or date fields only, a change in this field is sufficient. For example, if the error was an illegal time estimate, a code 2 card should be used to correct the error. If the error involves one of the identifiers of an activity, two correction cards must be used: (1) a code 5 card to delete the incorrect activity and (2) a code 1 card to insert the corrected activity.

In case of concurrent activities, the programmed DUMMY activities must be replaced even if the program correction was adequate. The new activity could be a dummy activity, but could not have an identification DUMMYn, since in successive runs, concurrent activities may recur and the program starts adding the events DUMMY1, DUMMY2, ..., and ambiguity may occur.

VI. SERVICE ROUTINES

INTRODUCTION

PERTSERV is a service program incorporated into the Philco 2000 PERT System. It provides the following service functions:

- Prints out the complete data of any project on the history tape.
- Copies projects from one history tape onto another.
- Adds or deletes projects.
- Transmits the history of a project from the history tape to binary or Hollerith punched cards, and from binary punched cards to tape.
- Analyzes projects for loops.
- Lists all the projects on a history tape.
- Initializes and rewinds history tapes.

Each reference to a service routine is made by a PERTSERV control card. A PROJECT card may not immediately precede a PERTSERV card except when it is a PERTSERV BININPUT card followed by the binary history cards. PERSERV cards can follow each other or any project's deck of cards, or may precede any project deck.

PERTSERV CARD FORMATS

Column	Content
1-8	The control word PERTSERV.
17-32	The name of the service routine to be used.
33-48	The ID of the project, if an ID is necessary.
57-58	The tape unit in decimal, if a tape unit must be specified.

Any columns not specified must be left blank.

The service routines available with PERT are described below.

PRINTHIST

Action

PRINTHIST prints out the history of the project from the specified history tape. This history is a list of all activities with their durations and scheduled and actual completion dates. The history tape is positioned at the end of the printed project.

Card Format

Columns 1-8: PERTSERV

Columns 17-25: PRINTHIST

Columns 33-48: ID

Columns 57-58: TAPE UNIT

If columns 57-58 are blank, tape 10 is assumed.

Example

PERTSERV PRINTHIST BUILD AEROA1 10

PERTSERV will search tape unit 10 for the project BUILD AEROA1. When the project is found, the history will be printed out.

PUNCHIST

Action

PUNCHIST punches the history of a project from the specified history tape 10 in the input format (see page 5) with one time estimate only.

Card Format

Columns 1-8: PERTSERV

Columns 17-25: PUNCHIST

Columns 33-48: ID

Columns 57-58: TAPE UNIT

If columns 57-58 are blank, tape 10 is assumed.

Example

PERTSERV PUNCHIST BUILD AEROA1 9

PERTSERV will search tape unit 9 for the project BUILD AEROA1. When the project is found, a deck of cards of the project will be prepared on output tapes. The cards should be punched off line in code mode.

DELETE

Action

DELETE removes the designated project from the specified history tape. The history tape is then positioned at the end of the deleted project.

Card Format

Columns 1-8: PERTSERV

Columns 17-22: DELETE

Columns 33-48: ID

Columns 57-58: TAPE UNIT

If columns 57-58 are blank, tape unit 10 is assumed.

Example

```
PERTSERV    DELETE    BUILD AIRO1A    7
```

PERTSERV will search tape unit 7 for the project BUILD AIRO1A; when the project is found, it will be deleted from history tape 7.

Remarks

A project is deleted from a history tape by altering certain data in the ID block. The remainder of the data is unchanged. If several projects have been deleted from the history tape, it is desirable to use COMPRESS.

COMPRESS

Action

COMPRESS copies all the non-deleted projects from tape 10 to the assigned tape. The data of the deleted activities is ignored. Both input and output tapes are rewound when the copying is complete.

Card Format

Columns 1-8: PERTSERV

Columns 17-24: COMPRESS

Columns 57-58: TAPE UNIT

Example

```
PERTSERV    COMPRESS                6
```

PERTSERV will copy all non-deleted projects from tape unit 10 to tape unit 6.

COPY ADD

Action

PERTSERV copies the specified project from the history tape on unit 10 to the history tape on the specified tape unit. A sentinel block of Z's is written following the copied project. Both input and output tapes are positioned at the end of the copied project.

Card Format

Columns 1-8: PERTSERV

Columns 17-20: COPY

Columns 34-48: ID

Columns 57-58: TAPE UNIT

Example

```
PERTSERV    COPY    BUILD AIRO1A    9
```

Project BUILD AIRO1A will be copied from the history tape on unit 10 onto the history tape on unit 9.

Remarks

When copying a project onto a history tape, PERTSERV looks for a sentinel block and writes the specified project on top of the sentinel block just beyond the last project on the tape. Therefore, if there are no projects on the assigned tape, the COPY control instruction should be preceded by a WRTSENT instruction (see WRTSENT below).

LIST

Action

Types out a list of all projects on the history tape on the specified tape unit. At the end of the list, END OF LIST is typed out. The history tape is rewound at the end of the operation.

Card Format

Columns 1-8: PERTSERV

Columns 17-20: LIST

Columns 57-58: TAPE UNIT

If columns 57-58 are left blank, tape unit 10 is assumed.

Example

PERTSERV LIST 3

A list of all projects of the history tape on unit 3 will be typed out.

Remarks

It is advisable to follow COMPRESS, DELETE and COPY cards by a LIST card.

WRTSENT

Action

PERTSERV writes a sentinel block on the specified tape unit. The output tape is positioned at the beginning of the sentinel block.

Card Format

Columns 1-8: PERTSERV

Columns 17-23: WRTSENT

Columns 57-58: TAPE UNIT

Example

PERTSERV WRTSENT 11

A sentinel block will be written on tape 11.

REWIND

Action

PERTSERV rewinds the specified tape unit.

Card Format

Columns 1-8: PERTSERV

Columns 17-22: REWIND

Columns 57-58: TAPE UNIT

Example

PERTSERV REWIND 10

PERTSERV will rewind tape 10.

TAPE-TO-CARD

Action

PERTSERV will put the history of the specified project from the history tape mounted on a specified unit onto the normal output tape. Binary cards could be punched off line from this tape. The binary punched card format is as follows:

Columns 1-4: The first four characters of the project's ID.

Columns 5-8: Sequence numbers.

Columns 9-80: The history in image mode.

Card Format

Columns 1-8: PERTSERV

Columns 17-26: TAPETOCARD

Columns 33-48: ID

Columns 57-58: TAPE UNIT

If no tape is specified, tape 10 is assumed.

Example

```
PERTSERV      TAPETOCARD      BUILD AERO1A    9
```

Project BUILD AERO1A will be copied from tape 9 onto the output tape for punching cards. Cards punched in the format described above could be obtained from the tape using off-line tape to card equipment.

BININPUT

Action

PERSTSERV reads the binary history cards from the input tape into memory and then transfers control to PERT for processing and updating.

Card Format

Columns 1-8: PERTSERV

Columns 17-24: BININPUT

Example

```
PERTSERV      BININPUT
```

The project on the history tape will be read into memory.

MESSAGE

Action

PERTSERV types out the message written in columns 33-80 of the PERTSERV card.

Card Format

Columns 1-8: PERTSERV

Columns 17-23: MESSAGE

Columns 33-80: A message to be typed

Example

```
PERTSERV    MESSAGE    MOUNT TAPE 242 ON 9

PERTSERV    MESSAGE    PUNCH CARDS FROM TAPE  6
```

HALT

Action

PERTSERV types out the message written in columns 33-80 of the PERTSERV card, and then halts. PERT proceeds to the next project or PERTSERV card as soon as the ADVANCE bar is pressed.

Card Format

Columns 1-8: PERTSERV

Columns 17-23: HALT

Columns 33-80: A message to be typed

Example

```
PERTSERV    HALT    MOUNT SCRATCH TAPE ON 6
```

VII. INPUT PREPARATION AND OPERATING INSTRUCTIONS

INPUT PREPARATION

A number of projects may be processed at the same time. The project decks should follow each other; PERTSERV cards can be placed anywhere ahead of or between projects. (Refer to PERTSERV, page 25.)

AN END card should terminate the entire deck. An END card has the word END in columns 1-3; the rest of the card is blank.

Input cards should be transmitted off line in code mode, ten words per card, twelve cards per block.

The input tape should be placed on tape unit 0.

The history tape should be placed on tape unit 10.

PROGRAM HALTS

PERTSERV Halt Instruction

PERTSERV control instruction HALT will type out a message and then halt (see HALT, page 30). After obeying the instruction, the operator should press the ADVANCE bar and the program will continue.

Irrecoverable Tape Error

In the event of irrecoverable tape error, the program types out:

IO TROUBLE

and halts.

At this point the operator should press the ADVANCE bar. PERT will then attempt to process the next project.

Final Halt

When all projects have been completed, PERT types out:

END PERT

and halts.

OUTPUT PROCESSING

All output is written on tape unit 5. In order to obtain both the printed data and the punched cards, the procedure below should be followed:

- For printed output, use Data Select 0.
- For binary punched cards, use Data Select 1.
- For Hollerith punched cards, use Data Select 2.

The punched card decks are separated as follows:

- The first card of each binary card deck has punches in all rows of the first eight columns.
- Each Hollerith deck is followed by several blank cards.

The last block of each project deck contains a conditional stop.

The Hollerith punched cards should be interpreted.

APPENDIX A
CONSOLE TYPEWRITER TYPE-OUTS

The system starts with the type-out:

PERT SYSTEM INITIALIZED

For each project, the new ID card (if such a card follows the project card) and the number of blocks on the history tape are typed-out. All PERTSERV cards are also typed out. When all projects have been processed, PERT types:

END PERT

PERT SYSTEM INITIALIZED

PROJECT CONTRUCTION S123
HISTORY 20 BLOCKS

PROJECT MFG SN2
NEW ID MFG SN3
HISTORY 6 BLOCKS

PERTSERV PRINTHIST AEROA1 6

. .

. .

. .

. .

PROJECT ID
HISTORY N BLOCKS

PERTSERV MESSAGE TAPE 242 ON 10

END PERT

In the event of an irrecoverable tape error the program types out:

IO TROUBLE

and halts. If the ADVANCE bar is pressed, the program proceeds to the next project.

Notes:

The error type-outs are described in Chapter IV, Diagnostic and Error Correcting Routines, pages 11-19. PERTSERV type-outs are described in Chapter VI, Service Routines, pages 25-31.

APPENDIX B
PROJECT SIZE

The maximum size of a PERT project is a function of the size of memory of the Philco 2000 system used and of the specific operating system. The number of activities, A, is:

$$A = \frac{M - S - 4330}{3.75} - 46$$

where M is the size of memory, and S is the size of the operating system. The maximum number of events is always half the maximum number of activities.

For a 32,768 word Philco 2000 computer operating under SYS,

$$A = \frac{32,768 - 512 - 4330}{3.75} - 46$$

yielding a maximum PERT project of 7400 activities.

APPENDIX C
ADAPTING PERT TO AN OPERATING SYSTEM

A tape table is incorporated in the PERT System which allows easy adaptation of PERT to operating systems other than SYS. The table consists of 16 consecutive locations and is assigned the symbol TPUNIT by the card:

```
ASGN    TPUNIT,TAPETBL
```

The input, output and history tapes are assigned the symbols UNIT0X, UNIT5X, and UNIT10X, respectively, by the cards:

```
ASGN    UNIT0X,TPUNIT+s

ASGN    UNIT5X,TPUNIT+t

ASGN    UNIT10X,TPUNIT+u
```

where s, t, and u are relative positions within the table.

Logical tape unit numbers are at T23 of table locations. References to tape units are made indirectly. For example, if columns 57-58 of a PERTSERV card contain 12, the actual tape unit referenced is in location TPUNIT+12 at T23.

Data select reassignments can be made at compile time by changing the following cards:

```
ASGN    PRINTSL,x

ASGN    DATAPUB,y

ASGN    DATAPUH,z
```

where x, y, and z are integers from 0-15 and are data selects for printing, punching binary cards, and punching Hollerith cards, respectively.

PHILCO®

A SUBSIDIARY OF *Ford Motor Company*

COMPUTER DIVISION • WILLOW GROVE, PENNSYLVANIA

Printed in U.S.A.