INTEROFFICE MEMORANDUM

PM 35-2

DATE: 1[illegible] January 1969

SUBJECT: How Good is the PDP-11?

TO: Ted Johnson
Nick Mazzarese

FROM: John Cohen

As you know, we have devoted much time and effort programming comparison examples of the PDP-11 and other machines. The results have been quite significant - the PDP-11 turns out to be significantly ahead of all other small computers in both speed and memory economy.

Some of the most startling statistics are:

--the PDP-11A runs programs 5 times faster than <u>any</u> other small computer

--programs can be coded on the PDP-11 using 33% fewer bits than an equivalent program on any other computer

We are naturally quite excited about these statistics. In addition, the feed-back we get from the groups producing the software indicates that the machine is very easy to program. It almost seems too good to be true, but I have checked through the technical material myself and find that it is accurate.

The detailed programming comparison examples appear in Project Memoranda PM 18-3 and PM 35-1. I shall summarize the results here.

We used five test problems. Two are principally character oriented and two arithmetic. Two of the routines will require to be "subroutinized", in that they picked up arguments from calling routine or restored values. The problems were:

1. Move characters in edit
2. Multiply subroutine
3. Tolerance check
4. Histogram compellation
5. Decimal to binary conversion

Machines in the comparison were:

PDP-11A (fast PDP-11)
PDP-11B (slow PDP-11)
PDP-8
Data General NOVA
Varian 520I
Varian 620I

Hewlett-Packard 2114A
Interdata I3 (internal instruction set)
Interdata I3 (external instruction set)
PDP-X (a machine we never built)
SPC-12
PDC-808

The following table shows the number of bits need to encode each of the five problems on the 12 instruction sets:

	1	2	3	4	5
PDP-11A	280	280	240	320	200
PDP-11B	280	280	240	320	200
PDP-8	468	432	420	376	256
NOVA	640	272	320	400	368
520I	752	488	632	496	320
620I	1130	496	780	896	440
2114A	1040	384	576	656	340
I3 (INT)	864	1100	1200	784	512
I3 (EXT)	784	640	688	640	384
PDP-X	576	368	432	448	304
SPC-12	824	1030	1420	752	344
PDC-808	664	1080	1320	728	632

Normalizing this to 100 for the PDP-11, we get:

	1	2	3	4	5	ave
PDP-11A	100	100	100	100	100	100
PDP-11B	100	100	100	100	100	100
PDP-8	167	160	175	118	128	150
NOVA	229	97	133	125	184	154
520I	269	171	264	155	160	203
620I	404	177	326	280	220	281
2114A	372	137	240	206	170	225
I3 (INT)	309	393	500	245	256	340
I3 (EXT)	280	229	287	200	192	237
PDP-X	206	131	180	140	152	161
SPC-12	295	368	592	235	172	332
PDC-808	237	386	550	228	316	343

The average over the five problems for each machine appears in the right hand column. It is very significant to notice the differences between machines. Some machines (Interdata Internal, SPC-12, PDC-808) actually use more than three times as many bits to encode a problem than the best machine in this category.

The number of machine cycles to execute each problem was also tallied. These results were:

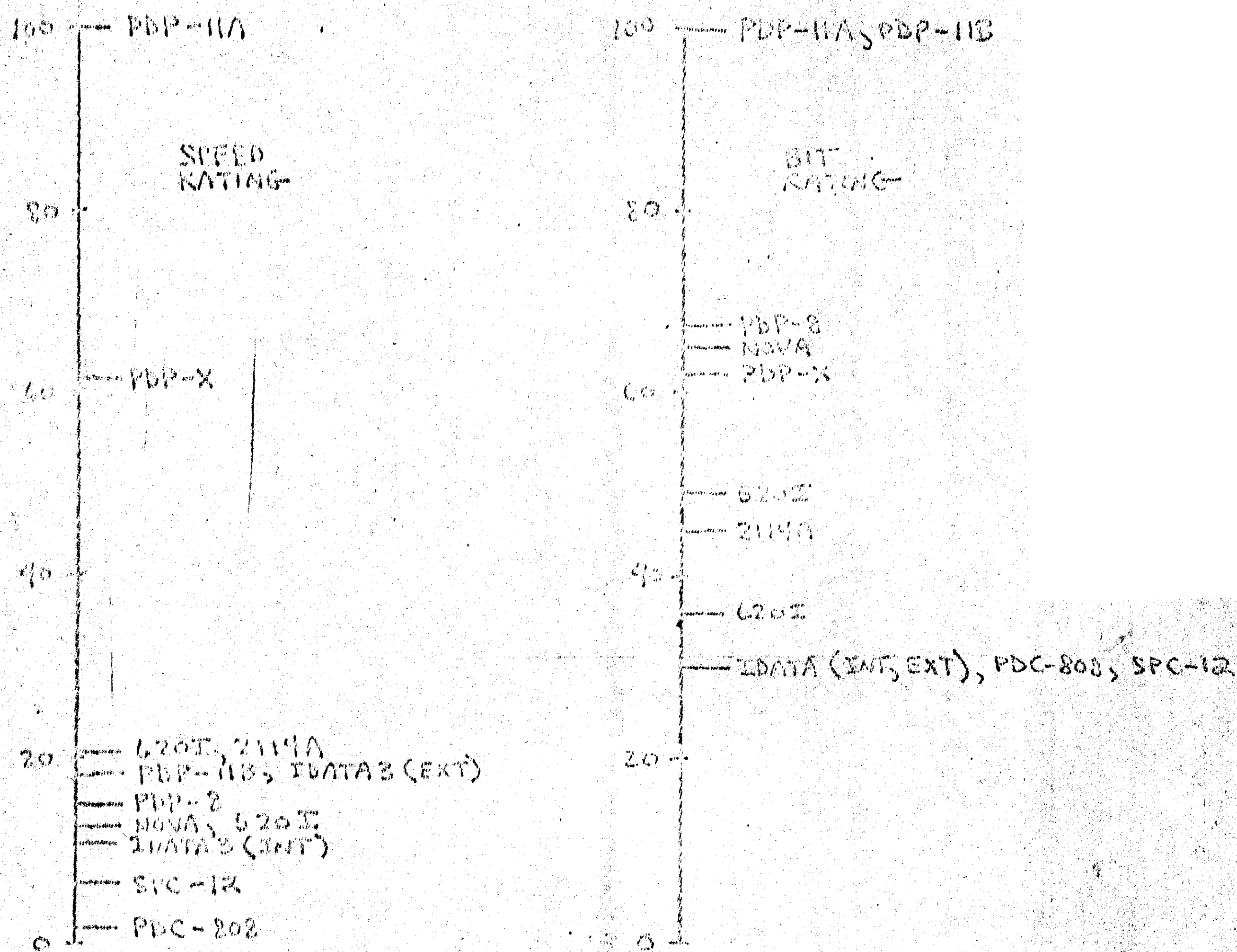
	1	2	3	4	5
PDP-11A	4260	149	1210	12400	49
PDP-11B	8210	297	2220	20600	98
PDP-8	50500	313	2640	21600	198
NOVA	31200	135	1530	16600	190
520I	35200	260	4880	40000	218
620I	43100	451	2260	51600	190
2114A	17400	149	1750	33800	133
I3 (INT)	18200	294	4080	32700	125
I3 (EXT)	21000	500	2560	40500	102
PDP-X	11400	188	1320	16500	85
SPC-12	32200	843	10500	57400	196
PDC-808	27000	979	12000	63700	529

Again normalizing to a 100 for the PDP-11A, we get:

	1	2	3	4	5	ave	cyc	spd
PDP-11A	100	100	100	100	100	100	1.0	100
PDP-11B	193	199	182	166	200	188	3.0	564
PDP-8	1190	210	218	174	404	439	1.5	658
NOVA	734	91	127	134	388	294	2.6	764
520I	1210	175	404	322	446	511	1.5	767
620I	1010	303	187	418	388	461	1.0	461
2114A	409	100	145	272	272	239	2.0	478
I3 (INT)	428	195	337	264	256	296	3.0	887
I3 (EXT)	494	336	212	326	208	315	1.8	567
PDP-X	268	126	109	133	173	161	1.0	161
SPC-12	760	567	869	462	400	611	2.4	1460
PDC-808	635	656	992	514	1080	775	8.0	6190

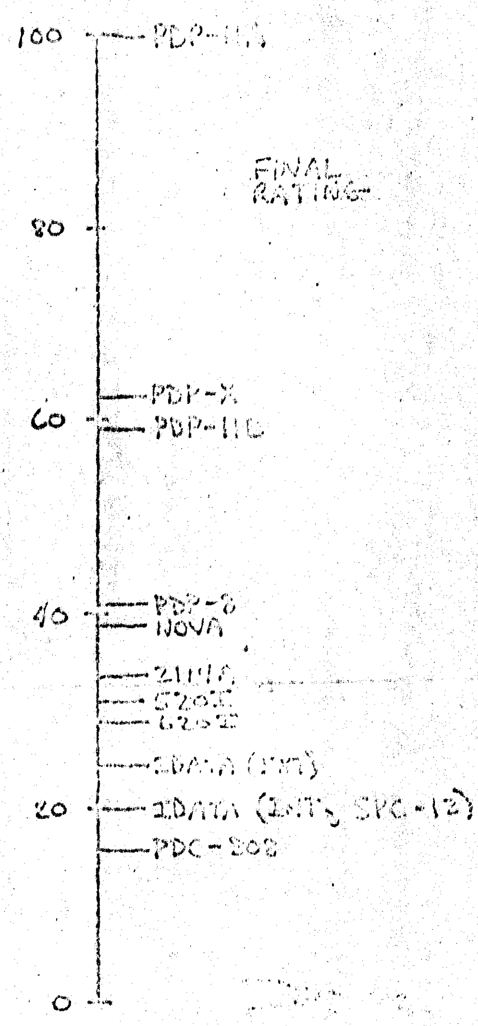
The number of cycles for each machine was averaged in the column headed "ave". The next column (cyc) contains the cycle time for the machine. Finally, the right hand column gives the normalized averaged speed for the five problems on each machine. Again there is a tremendous spread. The fact that the PDP-11 wins so strongly in both speed and memory economy, couple with its low price, clearly indicates that the product has unlimited potential.

We have compiled ratings by taking the inverse of the bit and speed normalized averages for each machine. The plots for both speed and memory economy are:



This graphically indicates how the PDP-11A simply runs away from all the other machines in processing speed and how the entire PDP-11 family handily defeats all other computers in bit efficiency.

As a final rating, we averaged the speed with the bit efficiency, counting each equally. The results speak for themselves:



CONFIDENTIAL

digital INTEROFFICE MEMORANDUM

PM 35-1

DATE: 11 January 1969

SUBJECT: Comparison of PDP-11, PDP-8 and the NOVA

TO: Distribution List

FROM: John Cohen

I. Introduction

When the PDP-11 instruction set design was completed, a number of people asked how it compared with other computers. To get the answer, it was decided to compare typical small computer problems on the PDP-11, PDP-8 and the NOVA. Five examples were chosen - some character oriented and some arithmetic. Each problem was coded for the three computers. Instruction bits and execution cycles were counted. The results of these counts were used to evaluate the performance of the various instruction sets.

The PDP-11 turned out to be best both in speed and memory economy. Specifically:

--the PDP-8 used 50% more instruction bits than the PDP-11
--the NOVA used 54% more instruction bits that the PDP-11
--the PDP-11A runs the example problems 6 times faster than the PDP-8
--the PDP-11A runs the problems 7 times faster than the NOVA
--the PDP-11B is 20% faster than the PDP-8
--the PDP-8 is 15% faster than the NOVA

The study procedure which we followed is defined in Section II. The results of the study are given in Section III. Appendix A contains the actual code which we generated for the test problems.

II. Data Gathering

This section discusses the test applications in the study and describes the evaluation procedure.

A. Applications

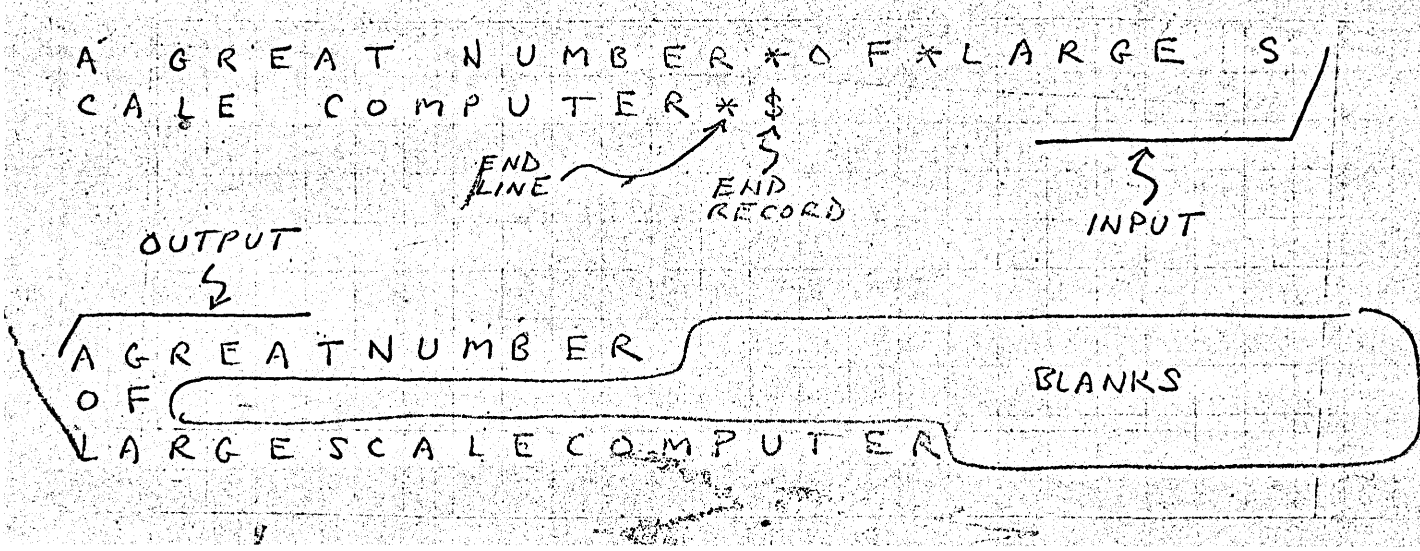
We used five test problems. Two were principally character oriented and two arithmetic. Another operated on both 8-bit and 16-bit data. Two of the routines were required to be "subroutinized", in that they picked up arguments from a calling routine and restored values.

1. Move Characters and Edit

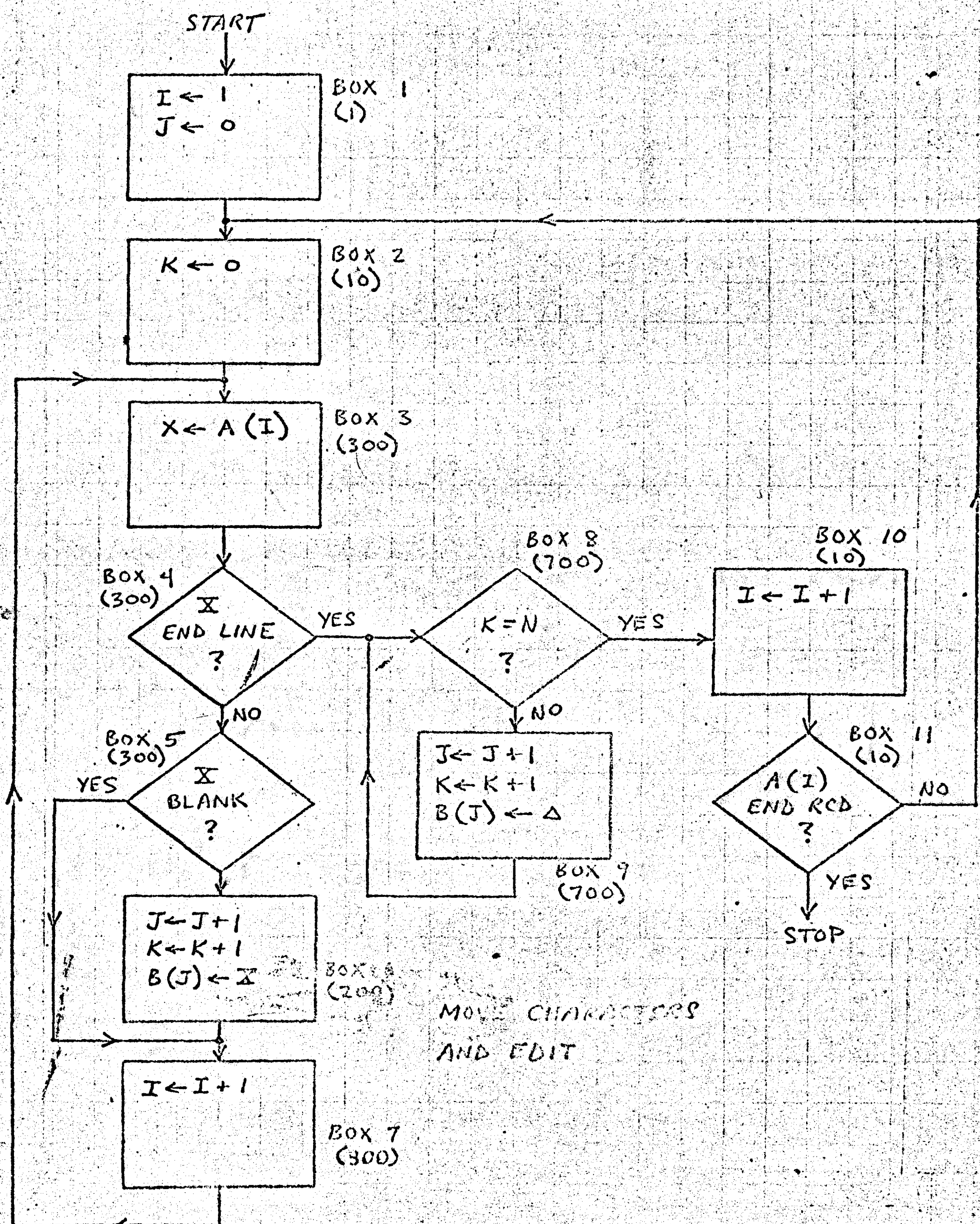
This problem tested the ability of the machine to move characters (8-bits) from one memory block to another. It also made use of array indices - which had ranges of less than 256 and also which had ranges greater than 256. The ability of the machine to branch on character matches was also important.

The input characters were broken down into a variable number of lines, and terminated with a special end of record character. Each line was of variable length, terminated by a special end of line character. The output character array was the same as the input, except that the lines were edited to a fixed length. The end of line, end of record characters were removed, as well as all embedded blanks. The individual lines were blank filled on the right to make them all the same length (the output line length was greater than the number of nonblank characters in each input line).

An example of the operation of this routine is:



The flow chart is:



where A is the input array
B is the output character array
I is an index to the input
J is an index to the output
K counts the number of characters in an output line
X holds the current input character
N is the (variable) number of characters in an output line

Notice that the boxes in the flow chart are numbered. The number of times each box is executed is given by the assoicated number in parenthesis.

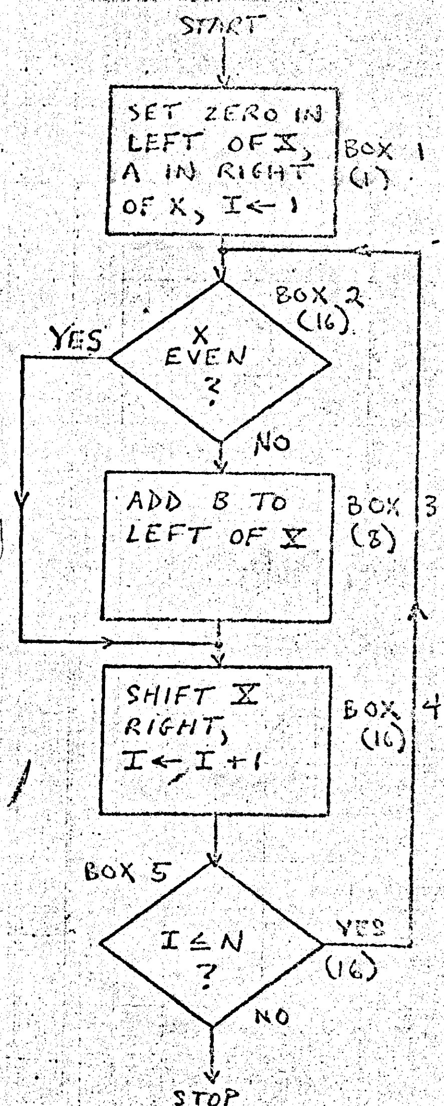
We assumed that there were 10 input lines of 30 characters each, 20 of which were nonblank. The output line length was 100. Box 1 is executed once for initialization. Boxes 2, 10 and 11 are entered 10 times - once for each line. Boxes 3, 4, 5 and 7 are executed for each input character - in our example 300 or 30 per line. Box 6 is executed for only the nonblank characters - thus 200 times or 20 times per line. Boxes 8 and 9 are used to fill the input lines with blanks. This occurs 70 times per line for a total of 700 in the problem.

2. Multiply Subroutine

This problem had a number of objectives - testing the ability of the machine to set up subroutine linkage, to sense bit configurations and branch conditionally and to easily shift double words. The test is also important because the multiply operation is commonly used.

The program first picked up two 16-bit operands from the calling program. These were multiplied together by the usual shifting and adding method. The 32-bit result was returned to the calling program. The routine operated only on unsigned integers.

The flow chart is:



where X is a double word to hold one of the operands and the result
A is one of the operands
B is the other operand
I is an index to the number of bits in each operand
N is the number of bits in each operand

We assumed 16-bit operands in all cases. On the PDP-8, the program was written for 12-bit words, but the cycle count was later adjusted as if the main loop was executed 16 times. We assumed that the first input argument contains exactly 8 one's.

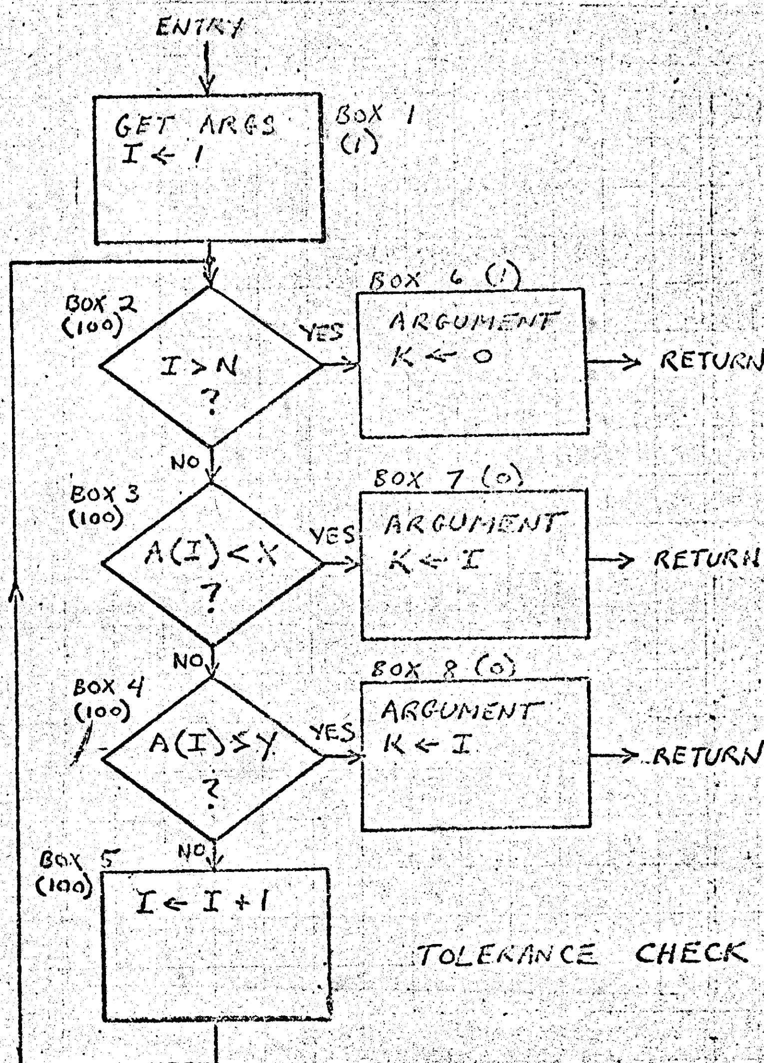
Box 1 is initialization and hence executed once. Boxes 2, 3 and 5 are entered for each bit - 16 times. Box 4 is executed only for 1 bits in the first operand - 8 times according to our assumptions.

3. Tolerance Check

The objective of this problem is to test arithmetic comparison capabilities and the ease in which the machine can index through an array of 16-bit quantities. Subroutine linkage was also considered in that the calling sequence to this problem was more complicated than that to the multiply subroutine.

The program picked up an array address, the count of the number of elements in the array and two tolerance limits from the calling program. It indexed through the array, checking each element against the low and high limits. If all elements were within tolerance, the program returned an output value of zero to the caller. If any were out of tolerance, the index in the array of the offender was returned.

The flow chart is:



where I is an index to the number of elements in the array
N is the number of elements in the array
A is the array of numbers
X is the low limit
Y is the high limit
K is the return argument

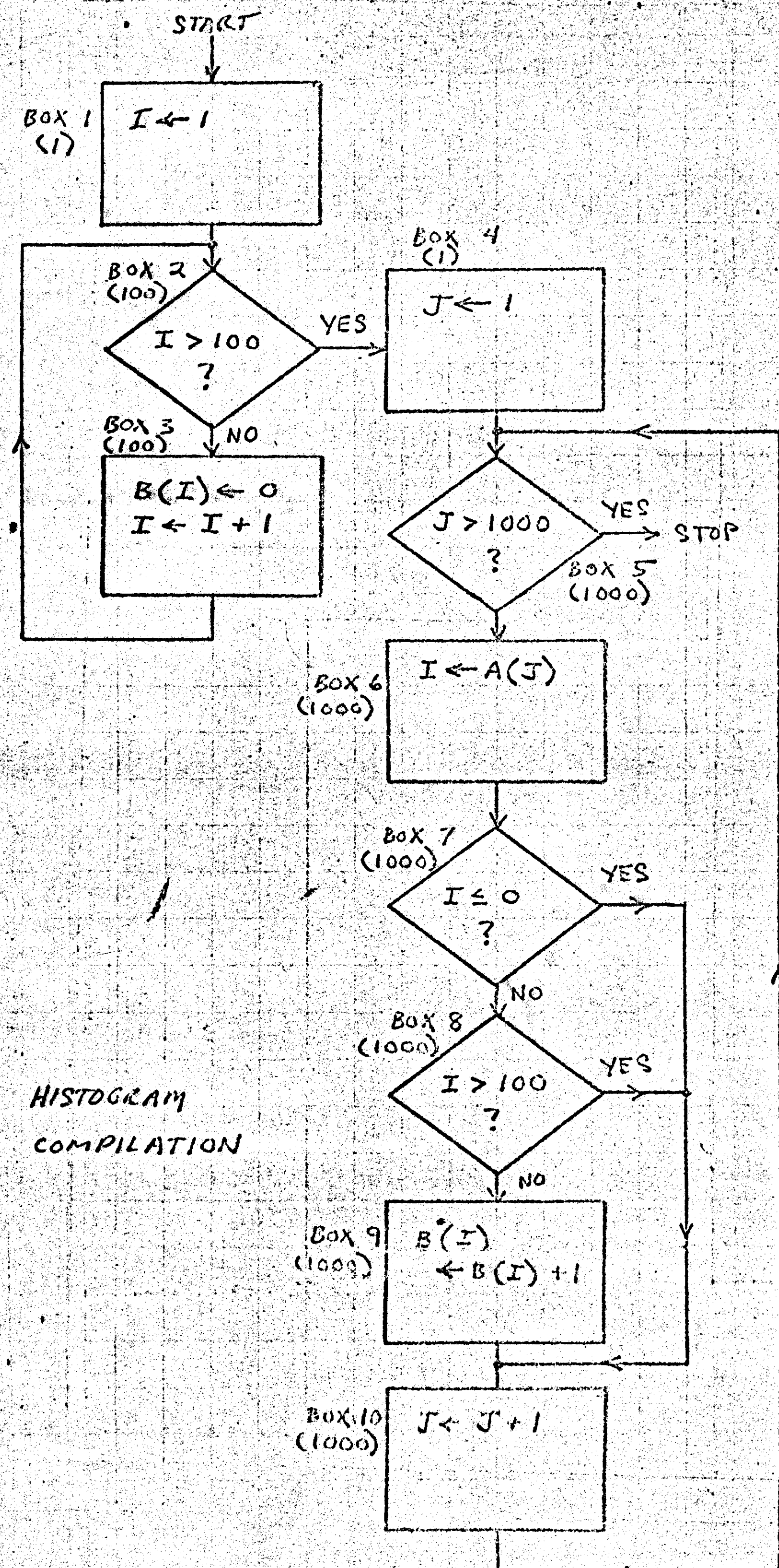
For timing considerations, we assumed that there were 100 entries in the array and that they were all in tolerance. Boxes 1 and 6 are executed once, as they involve initialization and termination, respectively. Boxes 2,3,4 and 5 are executed 100 times - once per array entry. Boxes 7 and 8 are not executed because of our assumption that all entries are in tolerance.

4. Histogram Compilation

It tests the ability of the machine to randomly index to memory arrays and to increment 16-bit memory integers.

The input is an array of 1,000 16-bit numbers, with values normally in the range of 1 through 100. The program must contain code to ignore other values - 0 or 101 through 256. The output is a memory array of 100 16-bit numbers. These contain the counts of occurences of the 100 possible input values. For example, if the 16-bit number 20 occurs exactly 15 times in the input array, the contents of the 20th element of the output array must be 15.

The flow chart for this problem is:



where A is the input data array
B is the output histogram array
I is an index through the output array
J is an index through the input array

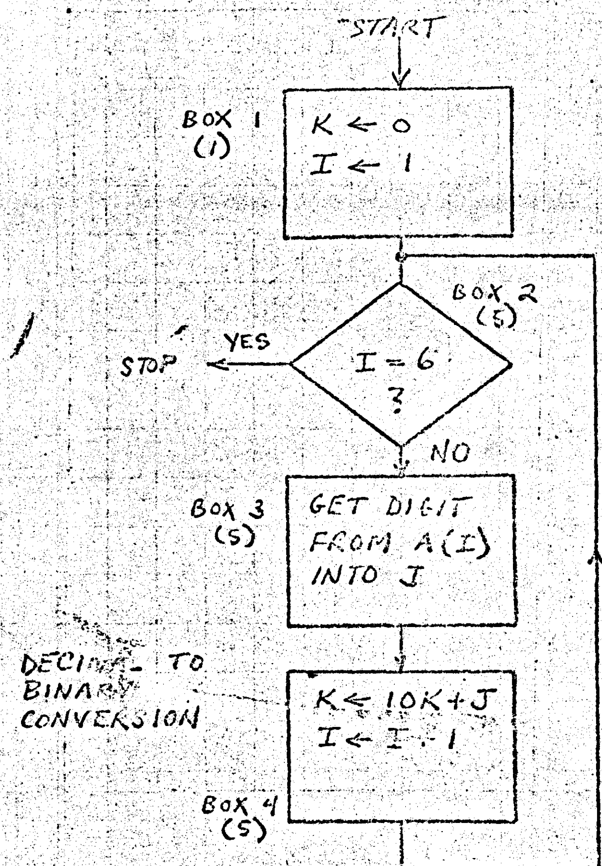
Boxes 1 and 4 are for initialization and hence, executed once. Boxes 2 and 3 are used to zero the output array and are executed once per output entry of 100 times. Boxes 5 through 10 are entered once per input entry, 1,000 times total.

5. Decimal to Binary Conversion

The objective of this test was to determine how the machines performed in this rather common application. The problem also tests character manipulation and the ability to do specialized multiplication using shifts and adds.

The input is a five character array and the output is an unsigned integer less than 32768. On the PDP-8, the routine was written for a 12-bit operation, but the cycle counts were multiplied by 16.

The flow chart is:



where A is the input character array
I is an index to the input characters
J is a temporary storage location for each binary digit
K is the binary result

B. Procedure

After the problems had been coded, we counted the number of bits and cycles used. The bit count involved a tally of the instruction words and storage for constant data. Temporary storage was not tallied, as it could be shared among routines or was contained in general registers. The word count was simply multiplied by the word length to get the program bit count.

Each program was partitioned as specified by the boxes in the flow charts. The number of machine cycles corresponding to each box was counted. This number was then multiplied by the corresponding box frequency, which appears with the box number on the flow charts. These products were then added together for each program, giving the cycle total for the problem.

Time figures for each computer were computed by multiplying the average cycle counts by the cycle time. This is assumed to be 1.5 microseconds for the PDP-8, 3.0 for the PDP-11B, 1.0 for the PDP-11A and 2.6 for the NOVA. The times for the latter were computed by assuming that every instruction took 2 cycles except for JMP and JSR which took 1.

III. Results

The following table summarizes the number of bits used for each problem:

prob	PDP-11A	PDP-11B	PDP-8	NOVA
1	280	280	468	640
2	280	280	432	272
3	240	240	420	320
4	320	320	376	400
5	200	200	256	368

Normalizing this to 100 for the PDP-11, this becomes:

prob	PDP-11A	PDP-11B	PDP-8	NOVA
1	100	100	167	229
2	100	100	160	97
3	100	100	175	133
4	100	100	118	125
5	100	100	128	184
average	100	100	150	154

The PDP-8 and the NOVA both have the liability of no byte handling instructions. Subroutines to load and store bytes were coded and called when relevant, but the bit count for these is not included (assuming that the routines are shared over a large number of programs).

The number of bits used by the PDP-8 varied from 18% more than the PDP-11 for the histogram example to 75% more in the tolerance check. The NOVA performed fairly well in this area for 16 bit arithmetic, but very poorly for character manipulations - even though a subroutine call was used.

B. Speed

Appendix B contains a detailed listing of the cycle counts for each problem on the three computers. The table for each problem has one row for each box on the flow chart. The number of times that box is executed is given in the frequency column. Then for each computer, the number of cycles to execute the box and the cycle total is given. The cycle totals are added together to form a grand total of memory cycles for each computer. A summary of these is:

prob	PDP-11A	PDP-11B	PDP-8	NOVA
1	4260	8210	50500	31200
2	149	297	313	135
3	1210	2220	2640	1530
4	12400	20600	21600	16600
5	49	98	198	190

Normalized to the PDP-11A, this becomes:

prob	PDP-11A	PDP-11B	PDP-8	NOVA
1	100	193	1190	734
2	100	199	210	91
3	100	182	218	127
4	100	166	174	134
5	100	200	248	258
average	100	188	408	269
time	100	564	610	700

The time row above was computed by multiplying the average cycles by the memory cycle time. As was mentioned before, the NOVA times were simplified by assuming all instructions took 2 memory cycles except JMP and JSR which took 1. The NOVA cycle is assumed to be 2.6 microseconds. The superiority of the PDP-11A is quite startling and it is significant that the PDP-11B edges out both the PDP-8 and the NOVA.

Appendix A. Coding For Example Problems

```
*       EXAMPLE 1A,B
*       MOVE CHARACTERS AND EDIT
*       PDP-11
START   LDW,I   A           1
        STW,M   MØ          1
        LDW,I   B-1         1
        STW,M   M1          1
Q2      LDW,I   -N          2
        STW,M   M2          2
Q3      LDB,LD  MØ          3
        CPB,I   1           4
        JEQ     Q8          4
        JLT     Q7          5
        INC,M   M2          6
        STB,MD  M1          6
Q7      INC,M,  MØ          7
        JMP     Q3          7
Q8      CLA                 8
Q9      STB,MD  M1          9
        INC,M   M2          9
        JNE     Q9          9
        LDB,MD  MØ          1Ø
        CPB,I   2           11
        JNE     Q2          11
```

```
*       EXAMPLE 1C
*       MOVE CHARACTERS AND EDIT
*       PDP-8
START   CLA                 1
        DCA     I           1
        DCA     J           1
Q2      TAD     MN          2
        DCA     K           2
Q3      TAD     I           3
        JMS     LDC         3
                A           3
        DCA     X           3
        TAD     X           4
        TAD     MENDL       4
        SNA CLA             4
        JMP     Q8          4
        TAD     X           5
        SNA                 5
        JMP     Q7          5
        ISZ     J           6
        ISZ     K           6
        JMS     STC         6
                J           6
                B           6
Q7      ISZ     I           7
        JMP     Q3          7
Q8      ISZ     J           9
        JMS     STC         9
                J           9
                B           9
        ISZ     K           9
        JMP     Q8          9
        ISZ     I           1Ø
        TAD     I           11
        JMS     LDC         11
                A           11
        TAD     MENDR       11
        SZA CLA             11
        JMP     Q2          11


MN              -N
MENDL           -1
MENDR           -2
LDC     Ø
        CLL RAR
        TADI    LDC
        SNL
        JMP     P1
        RTR
        RTR
        RTR
```

```
P1      AND     MR
        ISZ     LDC
        JMPI    LDC
STC     Ø
        DCA     T1
        TADI    STC
        DCA     T2
        TADI    T2
        CLL RAR
        ISZ     STC
        TADI    STC
        DCA     T2
        SNL
        JMP     P2
        CLL
        TAD     T1
        RTL
        RTL
        RTL
        DCA     T1
        TADI    T2
        AND     MR
P3      TAD     T1
        DCAI    T2
        ISZ     STC
        JMPI    STC
P2      TADI    T2
        AND     ML
        JMP     P3

ML              77ØØ
MR              ØØ77
```

```
*       EXAMPLE 1D
*       MOVE CHARACTERS AND EDIT
*       NOVA
START   SUB     Ø,Ø             1
        STA     J               1
        INC     Ø,Ø             1
        STA     I               1
Q2      LDA     KP              2
        STA     K               2
Q3      LDA     1,AADR          3
        LDA     2,I             3
        JSR     LDC             3
        LDA     3,ENDL          4
        SUBN    Ø,3,SNR         4
        JMP     Q8              4
        MOVN    0,0,SNR         5
        JMP     Q7              5
        ISZ     J               6
        DSZ     K               6
        LDA     1,BADR          6
        LDA     2,J             6
        JSR     STC             6
Q7      ISZ     I               7
        JMP     Q3              7
Q8      SUB     Ø,Ø             8
Q9      ISZ     J               9
        LDA     1,BADR          9
        LDA     2,J             9
        JSR     STC             9
        DSZ     K               9
        JMP     Q9              9
        ISZ     I               1Ø
        LDA     1,AADR          11
        LDA     2,I             11
        JSR     LDC             11
        LDA     3,ENDR          11
        SUBN    Ø,3,SZR         11
        JMP     Q2              11
AADR            A
BADR            B
ENDL            1
ENDR            2

KP              N
LDC     MOVZR   2,2
        ADD     1,2
        LDA     Ø,(2)
        MOV     Ø,Ø,SZC
        MOVS    Ø,Ø
        LDA     1,MR
        AND     1,Ø
        JMP     Ø,3
```

```
STC     STA      3,RTN
        MOVZR    2,2
        ADD      1,2
        LDA      1,(2)
        MOV      Ø,Ø,SZC
        JMP      P1
        LDA      3,ML
P2      AND      3,1
        ADD      Ø,1
        STA      1,(2)
        JMPI     RTN
P1      MOVS     Ø,Ø
        LDA      3,MR
        JMP      P2
ML               1774ØØ
MR               377
```

```
*       EXAMPLE 2A,B
*       MULTIPLY
*       PDP-11
MPY     CLA                  1
        STW,M    MØ          1
        LDW,XD   (1)         1
        STW,M    M1          1
        LDB,I    2Ø          1
        STW,M    M2          1
Q2      LDW,M    M1          2
        RAR                  2
        JLR      Q4+1        2
        LDW,M    MØ          3
        ADW,X    2(1)        3
        STW,M    MØ          3
Q4      CLL                  4
        LDW,M    MØ          4
        RAR                  4
        STW,M    MØ          4
        LDW,M    M1          4
        RAR                  4
        STW,M    M1          4
        DEC,M    M2          4
        JNE      Q2          4
        LDW,X    4(1)        Ø
        STW,M    M2          Ø
        LDW,M    MØ          Ø
        STW,MD   M2          Ø
        LDW,M    M1          Ø
        STW,MD   M2          Ø
        JMP      6(1)        Ø
```

```
*       EXAMPLE2C
*       MULTIPLY
*       PDP-8
MPY     Ø                    1
        DCA      XL          1
        TADI     MPY         1
        DCA      T1          1
        TADI     T1          1
        DCA      XR          1
        ISZ      MPY         1
        TADI     MPY         1

        DCA      BADR        1
        TAD      M16         1
        DCA      T1          1
Q2      TAD      XR          2
        RAR                  2
        SNL CLA              2
        JMP      Q4+1        2
        TADI     BADR        3
        TAD      XL          3
        DCA      XL          3

Q4      TAD      XL          4
        CLL RAR              4
        DCA      XL          4
        TAD      XR          4
        RAR                  4
        DCA      XR          4
        ISZ      T1          4
        JMP      Q2          5
        ISZ      MPY         Ø
        TADI     MPY         Ø
        DCA      AUTØ/       Ø
        TAD      XL          Ø
        DCAI     AUTØ        Ø
        TAD      XR          Ø
        DCAI     AUTØ        Ø
        ISZ      MPY         Ø
        JMPI     MPY         Ø
M16              -2Ø
```

```
*       EXAMPLE 2D
*       MULTIPLY
*       NOVA
MPY     LDA     Ø,M16       1
        STA     Ø,I         1
        LDAI    1,(3)       1
        LDAI    2,1(3)      1
        SUB     Ø,Ø         1
Q2      MOVRM   1,Ø,SZC     2
        ADD     B,Ø         3
        MOVZR   Ø,Ø         4
        MOVR    1,1         4
        ISZ     I           5
        IMP     Q2          5
        LDA     2,2(3)      Ø
        STA     2,AUTØ      Ø
        STAI    Ø,AUTØ      Ø
        STAI    1,AUTØ      Ø
        JMP     3,(3)       Ø
M16             -2Ø
```

```
*       EXAMPLE 3A,B
*       TOLERANCE CHECK
*       PDP-11
TOL     LDW,XD  (1)         1
        NEG                 1
        STW,M   MØ          1
        LDW,X   2(1)        1
        STW,M   M1          1
Q2      LDW,MD  M1          3
        CPW,XD  4(1)        3
        JLT     Q7          3
        CPW,XD  6(1)        4
        JGT     Q7          4
        INC,M   MØ          5
        JNE     Q2          5
        CLA                 6
Z1      STW,XD  1Ø(1)       6
        JMP     12(1)       6
Q7      LDW,M   MØ          7
        ADW,X   (1)         7
        INA                 7
        JMP     Z1          7
```

```
*       EXAMPLE 3C
*       TOLERANCE CHECK
*       PDP-8
TOL     Ø
        CMA                 1
        TAD     TOL         1
        DCA     AUTØ        1
        TADI    AUTØ        1
        DCA     IP          1
        TADI    IP          1
        CIA                 1
        DCA     IP          1
        TADI    AUTØ        1
        DCA     AADR        1
        TADI    AUTØ        1
        DCA     XADR        1
        TADI    AUTØ        1
        DCA     YADR        1
        TADI    AUTØ        1
        DCA     KADR        1
        INA                 1
        DCAI    KADR        1
Q2      TADI    XADR        3
        CIA                 3
        TADI    AADR        3
        SPA CLA             3
        JMP     Q7          3
        TADI    AADR        4
        CIA                 4
        TADI    YADR        4
        SPA CLA             4
        JMP     Q7          4
        ISZ     AADR        5
        ISZI    KADR        5
        ISZ     IP          5
        JMP     Q2          5
        DCAI    KADR        6
Q7      JMPI    AUTØ        6
```

```
*       EXAMPLE 3D
*       TOLERANCE CHECK
*       NOVA
TOL     LDAI    Ø,(3)           1
        STA     I               1
        LDA     Ø,1(3)          1
        STA     Ø,AUTØ          1
        LDAI    1,2(3)          1
        LDAI    2,3(3)          1
        LDA     Ø,K1            1
        STA     Ø,4(3)          1
Q2      LDAI    AUTØ            3
        SUBNZ   1,Ø,SZC         3
        JMP     Q7              3
        SUBNZ   Ø,2,SZC         4
        JMP     Q7              4
        ISZI    4(3)            5
        DSZ     I               5
        JMP     Q2              5
        SUB     Ø,Ø             6
        STAI    Ø,4(3)          6
Q7      JMP     5(3)            6
K1              1
```

```
*       EXAMPLE4A,B
*       HISTOGRAM
*       PDP-11
START   LDW,I   B               1
        STW,M   MØ              1
        LDB,I   1ØØ             1
        STW,M   M1              1
        CLA                     1
Q2      STW,MD  MØ              3
        DEC,M   M1              3
        JNE     Q2              3
        LDW,I   A               4
        STW,M   MØ              4
        LDW,I   1ØØØ            4
        STW,M   M1              4
        LDW,I   B-1             4
        STW,L   XØ              4
Q5      LDW,MD  MØ              6
        JLE     Q1Ø             7
        CPW,I   1ØØ             8
        JGT     Q1Ø             8
        STB,R   .+2             9
        INC,X   Ø(Ø)            9
        DEC,M   M1              1Ø
        JNE     Q5              1Ø
```

```
*       EXAMPLE4C
*       HISTOGRAM
*       PDP-8
START   CLA                     1
        TAD     BADR            1
        DCA     AUTØ            1
        TAD     M1ØØ            1
        DCA     I               1
Q2      DCAI    AUTØ            3
        ISZ     I               3
        JMP     Q2              3
        TAD     M1ØØØ           4
        DCA     J               4
        TAD     AADR            4
        DCA     AUTØ            4
Q5      TADI    AUTØ            6
        CIA                     7
        SNA                     7
        JMP     Q1Ø             7
        CIA                     8
        TAD     M1Ø1            8
        SMA                     8
        JMP     Q1Ø             8
        TAD     BP              9
        DCA     T1              9
        ISZI    T1              9
Q1Ø     ISZ     J               1Ø
        JMP     Q5              1Ø
AADR            A
BADR            B
BP              B+144
M1ØØ            -144
M1Ø1            -145
M1ØØØ           -175Ø
```

```
*       EXAMPLE 4D
*       HISTOGRAM
*       NOVA
START   LDA     Ø,AADR      1
        STW     Ø,AUTØ      1
        LDA     Ø,M1ØØ      1
        SUB     1,1         1
Q2      STAI    1,AUTØ      3
        INC     Ø,Ø,SZR     3
        JMP     Q2          3
        LDA     Ø,AADR      4
        STA     Ø,AUTØ      4
        LDA     1,P1ØØ      4
        LDA     Ø,BADR      4
        LDA     3,M1ØØØ     4
Q5      LDAI    2,AUTØ      6
        NEGZN   2,2,SNC     7
        JMP     Q1Ø         7
        NEGZN   2,1,SZC     8
        JMP     Q1Ø         8
        ADD     Ø,2         9
        ISZ     (2)         9
Q1Ø     INC     3,3,SZR     1Ø
        JMP     Q5          1Ø
AADR            A
BADR            B-1
P1ØØ            144
M1ØØØ           -175Ø
```

```
*       EXAMPLE 5A,B
*       DECIMAL TO BINARY
*       PDP-11
START   LDW,I   A           1
        STW,M   MØ          1
        LDB,I   6           1
        STW,M   M1          1
        CLA                 1
        STW,M   M2          1
Q2      LDW,M   M2          4
        CLL                 4
        RAL                 4
        RAL                 4
        RAL                 4
        ADW,M   M2          4
        ADW,M   M2          4
        STW,M   M2          4
        LDW,MD  MØ          3
        ANB,I   17          3
        ADW,M   M2          4
        STW,M   M2          4
        DEC,M   M1          4
        JNE     Q2          4
```

```
*       EXAMPLE 5C
*       DECIMAL TO BINARY
*       PDP-8
START   TAD     M6          1
        DCA     IP          1
        DCA     I           1
        DCA     K           1
Q2      TAD     K           4
        CLL RTL             4
        RAL                 4
        TAD     K           4
        TAD     K           4
        DCA     K           4
        TAD     I           3
        JMS     LDC         3
                A           3
        AND     MSK         3
        TAD     K           4
        DCA     K           4
        ISZ     I           4
        ISZ     IP          4
        JMP     Q2          4
M6              -6
MSK             17
```

```
*       EXAMPLE 5D
*       DECIMAL TO BINARY
*       NOVA
START   SUB     Ø,Ø         1
        STA     Ø,K         1
        STA     Ø,I         1
        LDA     Ø,P6        1
        STA     Ø,IP        1
Q2      LDA     1,AADR      3
        LDA     2,I         3
        JSR     LDC         3
        LDA     1,MSK       3
        AND     1,Ø         3
        LDA     1,K         4
        MOVZL   1,1         4
        MOVZL   1,2         4
        MOVZL   2,2         4
        ADD     1,2         4
        ADD     Ø,2         4
        STA     2,K         4
        ISZ     I           4
        DSZ     IP          4
        JMP     Q2          4
AADR            A
P6              6
MSK             17
```

Example 1

		PDP-11A		PDP-11B		PDP-8		NOVA	
box	freq	ct	tot	ct	tot	ct	tot	ct	tot
0	1	0	0	0	0	0	0	0	0
1	1	4	4	8	8	5	5	8	8
2	10	2	20	4	40	4	40	4	40
3	300	1	300	2	600	23	6900	13	3900
4	300	2	600	4	1200	6	1800	5	1500
5	300	1	300	2	600	4	1200	3	900
6	200	2	400	3	600	43	8600	24	4800
7	300	2	600	3	900	3	900	3	900
8	700	0	0	1	700	0	0	2	1400
9	700	3	2100	5	3500	44	30800	25	17500
10	10	1	10	2	20	2	20	2	20
11	10	3	30	4	40	21	210	19	190
			4264		8208		50475		31158

Example 2

		PDP-11A		PDP-11B		PDP-8		NOVA	
box	freq	ct	tot	ct	tot	ct	tot	ct	tot
0	1	8	8	15	15	21	21	11	11
1	1	5	5	10	10	23	23	12	12
2	16	2	32	4	64	5	21	2	32
3	8	3	24	6	48	7	56	2	16
4	16	5	80	10	160	12	192	4	64
			149		297		313		135

Example 3

		PDP-11A		PDP-11B		PDP-8		NOVA	
box	freq	ct	tot	ct	tot	ct	tot	ct	tot
0	1	0	0	0	0	0	0	0	0
1	1	7	7	13	13	39	39	19	19
2	100	0	0	0	0	0	0	0	0
3	100	6	600	11	1100	9	900	6	600
4	100	4	400	8	800	9	900	3	300
5	100	2	200	3	300	8	800	6	600
6	1	5	5	9	9	5	5	6	6
7	0	5	0	8	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
			1212		2222		2644		1525

Example 4

		PDP-11A		PDP-11B		PDP-8		NOVA	
box	freq	ct	tot	ct	tot	ct	tot	ct	tot
0	1	0	0	0	0	0	0	0	0
1	1	5	5	9	9	9	9	8	8
2	100	0	0	0	0	0	0	0	0
3	100	4	400	6	600	6	600	6	600
4	1	7	7	13	13	8	8	10	10
5	1000	0	0	0	0	0	0	0	0
6	1000	2	2000	3	3000	3	3000	3	3000
7	1000	2	2000	2	2000	3	3000	3	3000
8	1000	2	2000	5	5000	5	5000	3	3000
9	1000	4	4000	7	7000	7	7000	4	4000
10	1000	2	2000	3	3000	3	3000	3	3000
			12412		20622		21617		16618

Example 5

		PDP-11A		PDP-11B		PDP-8		NOVA	
box	freq	ct	tot	ct	tot	ct	tot	ct	tot
0	0	0	0	0	0	0	0	0	0
1	1	4	4	8	8	8	8	10	10
2	5	0	0	0	0	0	0	0	0
3	5	3	15	5	25	19	95	17	85
4	5	6	30	13	65	19	95	19	95
			49		98		198		190