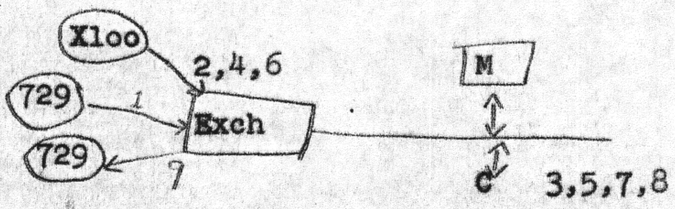
~~SECRET~~

Memo for the Record

Robert E. Lyons

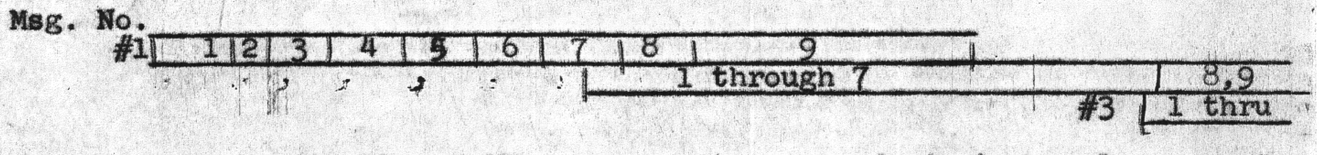
Problem Solution

Basic Procedure



1. Read input message into Harvest.
2. Read segment no. 1 of code book into Harvest (AAA-HZZ). Redistribute in memory for table look up.
3. Decode portion of message pertinent to segment no. 1.
4. Same as 2 for segment no. 2. (1AA-PZZ)
5. Same as 3 for segment no. 2.
6. Same as 2 for segment no. 3. (QAA-ZZZ)
7. Same as 3 for segment no. 3.
8. Editing run to assemble output block.
9. Output.
10. Reset X100 tape to starting point (NO TIME).

Timing Chart: (approx.-times to be figured out later).



Exchange unit allows all processes to proceed simutaneously, except that processes 2,4,6 will pre-emp control of the exchange.

Memory allocation

Input block size: max. msg. length-500 groups (1500 characters)= 50 blocks.
Total blocks per message (including heading) = 51 max.
Total words 1 message=510 (80 bytes=10 words per block).
Output block size: 5000 groups, one block per group, 500 blocks.
@ 10 words 1 block, 500 words of output allowed per message.
Total storage words per message=5510.

0.5 micro sec mem: 2048-4095: 2048-2557 - Input block

lock

3000-4095 - Program

~~SECRET~~

~~SECRET~~

TO : Memo for the Record DATE:
FROM : Robert E. Lyons
SUBJECT: Problem Solution

2 micro sec mem: 32768-98303: 32768-73727 - Code book segment
96304-98303 - Decoded meanings
91304-96303 - Output block

TIMING

1. Read input message into Harvest:
   Av. msg. length: 200 groups
   Av. blocks per message = 21
   Av. characters per message = 21X80 = 1680
   Input rate using 80 char. clocks = 5.5 KC approx.
   $T_1 = \frac{1680}{5500} = 0.306$ sec.

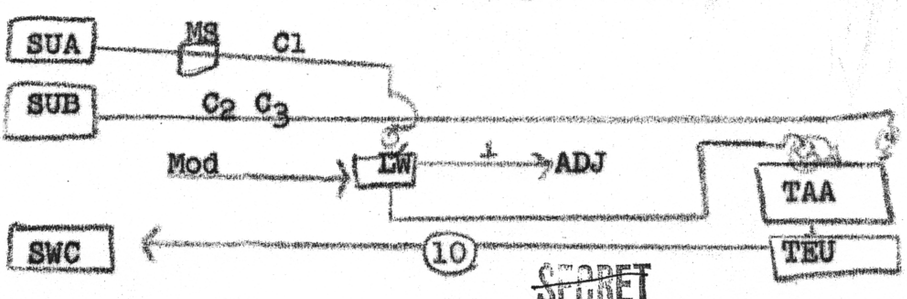
2. Load segment #1 of code book. Redistribute into memory for table look up.
   Bytes loaded: 183,892
   Loading rate: 1.0 M bytes 1 sec. (approx.) loading time,
   183,892 / 1 X 10^1 = 0.184 sec (This ignores inter block spacing, which is negligible. Also ignores .005 sec. start time.)
   Redistribution: From densely packed, 34 bytes per meaning, to packed for look up*, 32 bytes per meaning:
   Allow 0.2 micro sec/byte, 183,892 bytes.
   Time for distribution = (0.2) (183872) X 10^{-6} = 0.037 sec.
   $T_2 = 0.184 + 0.037 = 0.221$ sec.

* Packed for look-up:

rel. ad	table entry	
000	AAA	
004	AAB	
:	:	Leave blank spaces in
4X25	AAZ	table so indexing in TAA
:	blanks	(a mod 32 process) can be
4X32	ABA	used to look up in the
	etc.	table.

3. Decode portion of message pertinent to segment no. 1:
   Basic procedure:

look for end of message indicator to stop streamming process.



~~SECRET~~

~~SECR~~

TO : Memo for the Record DATE:
FROM : Robert E. Lyons
SUBJECT: Problem Solution

Code groups: C_1 C_2 C_3
Indexing in A selects C_1 four times, then next C_1 four times, etc.
Indexing in B selects C_2 C_3 four times, then next C_2 C_3 four times, etc.
Indexing in C merely selects consecutive words in the block of memory reserved for decoded meanings. However, there is a second level which selects every fourth word and is used for adjust.
Indexing in TEU same as in C
Indexing in TAA:
Base address = 32768 ($=2^{15}$) C_1 C_2 C_3 MD is used and cycles 00,01,10,11.
to be added with assembled address.

LU reduces by mod 8. 1 output if there is a mod reduction stops look up by adjusting indexing. This advances A to pick up next C_1 four times, B to pick up next C_2C_3 four times, jumps second level in C and TEU (skips 4 words), prevents an actual look-up in TAA which prevents MD from advancing.

If no mod reduction occurs, code group is pertinent to segment of book now in storage, four look-ups occur (same $C_1C_2C_3$ each time, different MD setting) to extract the four consecutive words of meaning and place them in the output.

(Processes 5,7 same as 3 except: In case 5, mod = 8, but 1 output of LU is inverted so that 1AA-ZZZ groups are looked up (QAA-ZZZ are in correct but will be changed in process 7). In case 7, mod = 16, otherwise same as 5.)

Timing - Allow 0.2 micro sec/byte for streaming operations C_2C_3 streamed once in $\frac{18}{26}$ of cases, four times in $\frac{8}{26}$ of cases (C_1 simultaneous, notime).

For 200 gp. message, $\frac{8}{26}$ (200) = 62 gps. looked up
(38 gps not looked up
Time of streaming = (0.4)(138)+(0.4)(4) (62) = 154 micro sec.

All 0.5 micro sec per address assembled in TAA in table look-up operations which sequence through four 2.0 micro sec. memorites:

(0.5) (4) (62) = 124
T_3 = 154 + 124 = 278 micro sec.

4. Process 4 same as 2
$T_4 = T_2 = 0.221$ sec.

5. Process 5 similar to process 3 except probability of look up is $\frac{16}{26}$ instead of $\frac{26}{26}$.

T_5 = (0.4) (62) + (0.4) (4) (138) + (0.5) (4) (138) = 25 +
(0.9) (552) = 25 + 497 = 522 micro sec.

~~SECRET~~

~~SECRET~~

TO : Memo for the Record
FROM : Robert E. Lyons
SUBJECT: Problem Solution

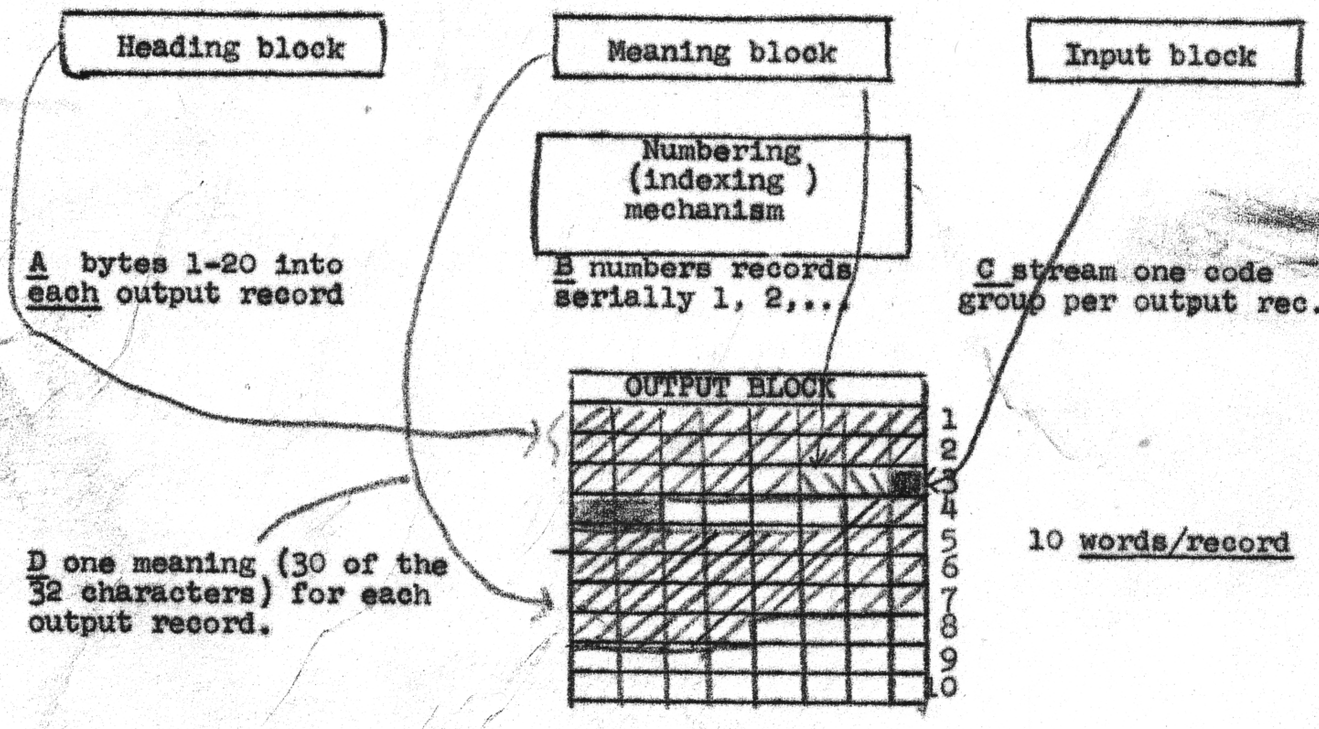
DATE:

6. Process 6 same as 2 except $\frac{10}{8}$ as much information is involved:
$T_6 = \frac{10}{8}$ $T_2 = \frac{10}{8}$ (0.221) = 0.278 sec.

7. Process 7 same as 3 except probability of look up is $\frac{10}{26}$ instead of $\frac{8}{26}$. $\frac{10}{26}$ (200) = 77, 200-77 = 123

$T_7 = (0.4)(123) + (0.4)(4)(77) + (0.5)(4)(77) = 326$ m sec.

8. Process 8 - editing:



TIMING

A. 20 bytes from .5 micro sec. mem. to 20 bytes from 2 micro sec. memory X200 gps. av.: 0.2 micro sec/byte X20 X200 = 800 micro sec.
B. 3 bytes. Use NCTR. 3X0.2X200 = 120 micro sec.
C. 3 bytes. 120 micro sec.
D. 30 bytes. 1200 micro sec.
T_8 = 2240 micro sec.
T_9 = (200 X 80 characters) (5500 char./ sec. = 2.95 sec

~~SECRET~~

~~SECRET~~

TO : Memo for the Record
FROM : Robert E. Lyohs
SUBJECT: Problem Solution

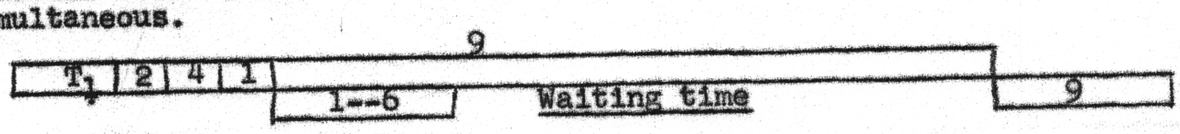
DATE:

Summary:

T_1 = .306 seconds
T_2 = .221 seconds
T_3 = .0003 seconds
T_4 = .221 seconds
T_5 = .0005 seconds
T_6 = .278 seconds
T_7 = .0003 seconds
T_8 = .0022 seconds
T_9 = 2.95 seconds

.723 sec exclusive of 729 time.

Actual total time. Use of exchange permits other processing to be simultaneous.



Process severely output limited.

CONCLUSION:
For this particular process to be feasible on Harvest, there should be a X100 (or at least X10) converter to use peripherally.

Other Comments:
I have assumed an ADDER in TAA for adding base address rather than Oring. In my opinion, this is a MUST.
I have not treated the problem of how to handle much larger code books. Disc memory may be a solution. Another feasible approach would be to use a large number of segments of the code book and sort the messages (a number of messages should be processed at once) before look-up. Then restore the order for output.
I have not had time to treat the problem of file maintenance of the code book, deciphering before decoding, etc. I think decodes would be much more feasible if such extensions were handled at the same time. Other extensions that should be considered on Harvest are such as:

1. editing of original messages
2. indexing
3. deciphering
4. code statistics (frequency, etc.)
5. language studies - translation.

and others.

A medium speed high volume storage would be most desirable. Some thing of the order of 10 to 100X capacity and 10X access time (ie. 10 times slower) as 2.0 micro seconds memory.

ROBERT E. LYONS

~~SECRET~~