

MACDONALD DETTWILER & ASSOCIATES LTD.

DIAGNOSTIC RHYME TEST
EVALUATION OF RELP CODED
SPEECH IN MECHANICAL NOISE
ENVIRONMENT USING A REAL-TIME
HARDWARE SIMULATOR
00-3058-R00
NOVEMBER, 1982



KC L P C654 M34 1982



macdonald dettwiler & associates LTD.

VANCOUVER, CANADA

DIAGNOSTIC RHYME TEST EVALUATION OF RELP CODED SPEECH IN MECHANICAL NOISE ENVIRONMENT USING A REAL-TIME HARDWARE SIMULATOR

00-3058-R00

Industry Canada Library - Queen

AOUT 1 6 2012

Industrie Canada Bibliothèque - Queen

Prepared by:

J.A. MADELEY and P.F. LEE

MacDonald, Dettwiler and Associates Ltd. 3751 Shell Road, Richmond, B.C., Canada, V6X 2Z9 Telephone No.: (604) 278-3411

© MacDonald, Dettwiler and Associates Ltd., 1982

**NOVEMBER, 1982

ORIGINAL DOCUMENTS REVIEW AND PUBLICATION RECORD

SECTOR	BRANCH	DATE
DGSTA	DSS	1982
It is designed to: INSTRUCTIONS * 1 copy of the	record decisions for classes provide for indexing recompleted form must ac	sification and cautionary marking quirements. company the contractor report
	itted to the CRC Library following items as appl	
DOC-CR NO. DOC-CR-SP-82-068	2. DSS CONTRA	CT NO. lable in book
3. TITLE: Diagnostic rhyme test evalumechanical noise environments. CONTRACTOR	ation of RELP coded speech	4. DATE
Macdonald Dettwiler & Association	iates Ltd.	
6. SCIENTIFIC AUTHORITY B. Bryden	7. LOCATION DSS/CRC	8. TEL. NO. 998-2515
9. CONTRACTOR REPORT CLASSIC	FICATION: CONDITIONALLY RELEASABLE	NON-RELEASABLE
* REASONS FOR CLASSIFICA	; KOII	
10. NO. OF COPIES SUBMITTED EXECUTIVE SUMMARY	TO LIBRARY: FINAL REPORT	XX 3 copies .
Scientific Authority's		Date
Scientific Authority's		Date

This form is not official therefore it is not signed.

TABLE OF CONTENTS

		<u>PAG</u>
1.	INTRODUCTION	a 41.1
2.	RELP REAL-TIME SIMULATIONS	2.1
	2.1 Residual Regeneration Through Sample PositionPerturbation2.2 5-Bit Adaptive Residual Quantizer for 9.6 KBP Simulation	,
3.	DIAGNOSTIC RHYME TESTS	3.1
	 3.1 Introduction 3.2 Acoustic Signal to Noise Ratio Measurement 3.3 Preparation of the DRT Tapes 3.4 Test Procedure 3.5 Analysis of Test Results 	3.1 3.1 3.2 3.2 3.4
	3.5.1 Introduction 3.5.2 DRT Scores and Observations 3.5.3 Comparison with the INRS Results	3.4 3.4 3.1
4.	CONCLUSIONS	4.1
٠ 5	REFERENCES	5.1

B. J. B. C.

TABLE OF APPENDICES

		PAGE
APPE	NDIX A - SYNTHESIS SIGNAL PROCESSOR FIRMWARE LISTNGS	
A1.	RESIDUAL REGENERATION THROUGH INTERPOLATION OF LOW PASS COMPONENT BY ZERO AMPLITUDE INSERTION, AND SAMPLE POSITION PERTURBATION OF HIGH FREQUENCY COMPONENT	A.1
A2.	RESIDUAL REGENERATION THROUGH INTERPOLATION OF LOW PASS COMPONENT BY ZERO AMPLITUDE INSERTION, AND SAMPLE POSITION PERTURBATION	A. 2
APPE	NDIX B - MULDEM SIMULATOR FIRMWARE LISTINGS	
B1.	2-BIT ADAPTIVE RESIDUAL QUANTIZER	B.1
B2.	5-BIT ADAPTIVE RESIDUAL QUANTIZER	B.2
ВЗ.	NO RESIDUAL QUANTIZER	B.3
APPE	NDIX C - DESCRIPTION OF THE ORIGINAL DRT TAPE SUPPLIED BY CRC	
ADDE	NDIV D - LIST OF WPONG ANSWERS	90m insk

TABLE OF ILLUSTRATIONS

FIGURE/TABLE	TITLE	FOLLOWING PAGE
FIGURE 2.1-1	Illustration of Spectral Folding	2.1
FIGURE 2.1-2	High Frequency Regeneration With An Unperturbed Baseband	2.1
FIGURE 2.1-3	Conditional Perturbation Probability As A Function of Input Sample Amplitude	2.1
FIGURE 2.1-4	High Frequency Regeneration With A Perturbed Baseband (PERT)	2.1
FIGURE 2.2-1	Mid-Rise Quantizer Characteristic For Step Size Greater Than or Equal To A Preset Threshold	2.2
FIGURE 2.2-2	Mid-Tread Quantizer Characteristic For Step Size Less Than The Preset Threshold	2.2
TABLE 2.2-1	5-Bit Adaptive Residual Quantizer Parameters	*2.3
FIGURE 3.2-1	Schematic of True RMS to DC Converter	3.1
FIGURE 3.2-2	True RMS Waveform of a Typical Spoken Word Observed on an Oscilloscope	3.1
TABLE 3.3-1	Schedule of Tests	*3.3
TABLE 3.5.2-1	DRT Scores by Person and Time of Test	*3.6
TABLE 3.5.2-2	DRT Scores by Attribute	*3.7
FIGURE 3.5.2-1	Diagnostic Rhymes Test Results for RELP Coding Without Noise	3.7
FIGURE 3.5.2-2	Diagnostic Rhyme Test Results For RELP Coding With Armoured Personnel Carrier Noise	3.7
FIGURE 3.5.2-3	Diagnostic Rhyme Test Results for RELP Coding With Helicopter Background Noise	3.7
TABLE 3.5.2-3	Bias in DRT Scores	*3.8
TABLE 3.5.2-4	Average Word Power Versus Attribute	*3.9

^{*} Actual Page Number

TABLE OF ILLUSTRATIONS

FIGURE/TABLE	TITLE	FOLLOWING PAGE
TABLE 3.5.2-5	DRT Scores as a Function of Signal to Noise Ratio	*3.10
TABLE 3.5.2-6	DRT Scores and Standard Errors	*3.11
FIGURE 3.5.2-3	Diagnostic Rhyme Test Results for RELP Coding With Helicopter Background Noise	3.11
FIGURE 3.5.3-1	Comparison with INRS Diagnostic Rhyme Test Scores for 4.8 KBPS RELP	3.12
FIGURE 3.5.3-2	Comparison with INRS Diagnostic Rhyme Test Scores for 4.8 KBPS RELP with 10 dB Signal to Noise Ratio	3.12
FIGURE 3.5.3-3	DRT Scores for Uncoded Speech Without Noise	3.12
FIGURE 3.5.3-4	DRT Scores for Uncoded Speech at 10 dB Signal to Noise Ratio	3.12

^{*} Actual Page Number

1. INTRODUCTION

A Residual Excited Linear Prediction (RELP) codec (coder/decoder) real-time hardware simulator has been developed recently with funding from the Communications Research Centre (CRC). It is believed to be the first of its kind built in Canada. The simulator permits real-time evaluation of RELP coded speech at various data rates, in particular, 4.8 and 9.6 KBPS. It also provides flexibility to fine tune the original INRS (Institut National de la Recherche Scientifique) developed RELP algorithm through simple firmware changes. Preliminary laboratory testing of RELP coded speech at 4.8 KBPS has revealed that speech quality is judged to be very intelligible and of good quality. In addition, the robustness of the RELP algorithm to background noise makes it particularly attractive for military and law enforcement applications.

In the military environment, mechanical noise in the background is frequently encountered. The popular Linear Prediction Codecs (LPCs) performs poorly in the presence of periodic noise because it interfers with the pitch extraction process. This report attempts to measure the intelligibility of RELP coded speech and its robustness to periodic noise quantitatively through the help of Diagnostic Rhyme Test (DRT). Real-time simulation of alternate high frequency regeneration methods on the hardware simulator are also included in this study to realize a more economical implementation of the RELP codec.

2. RELP REAL-TIME SIMULATIONS

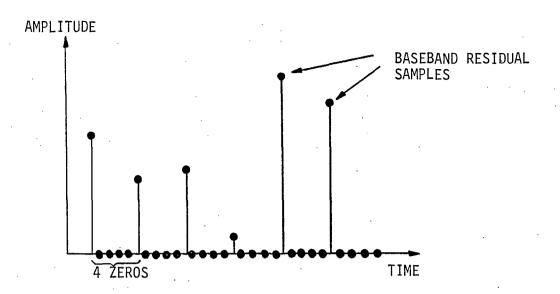
2.1 Residual Regeneration Through Sample Position Perturbation

The primary source of error in the 9.6 KBPS RELP codec is the reconstruction of the fullband residual from the baseband residual. This can be demonstrated by using unquantized residual and predictor, which gives very little improvements in speech quality over 9.6 KBPS RELP.

A paper presented by R. Viswanathan [1], claimed that a spectral folding high frequency regeneration technique produced higher speech quality than the rectification method used in the RELP. In its simplest form this method involves inserting four zeros after each baseband signal. This causes aliasing terms up to 4 kHz to be generated (see Figure 2.1-1). This simple technique generates objectionable tonal noises.

A perturbation method was described in the paper which masked the tonal This involved randomly perturbing the baseband residual sample to the left or right by one sample position. Small samples were perturbed with a greater probability than large samples to reduce the chance of perturbing pitch pulses. The high pass filtered perturbed signal was added to the low pass unperturbed signal to regenerate the fullband residual (see Figure 2.1-2). The random perturbation was accomplished on the hardware simulator by comparing the absolute value of the sample to be perturbed with a random signed number between -1/8 and 1/8. The random number is extracted from bits truncated in an arithmetic operations. If the sample was greater than the random number, no perturbation was performed, otherwise, the sample was perturbed to the left or right based on a random bit. This generates a conditional probability of perturbation as a function of input sample amplitude given in Figure 2.1-3. perceived quality of the processed speech was about the same as the rectification method. A 30% reduction in processing resource may be obtained in the synthesis processor using this method. If the perturbed residual was fed into the predictor directly without the low and high frequency paths suggested in the paper (Figure 2.1-4), a further 40% or a total of 70% of the processing resource may be saved in the synthesizer without affecting the subjective quality of the RELP coded speech.

A small number of diagnostic rhyme tests were performed to see if this algorithm had the same intelligibility as RELP. The perturbation algorithm in Figure 2.1-4 is referred to as "PERT" in the rest of this report.



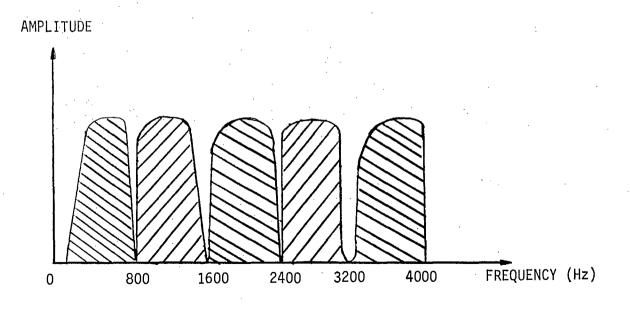


FIGURE 2.1-1 ILLUSTRATION OF SPECTRAL FOLDING

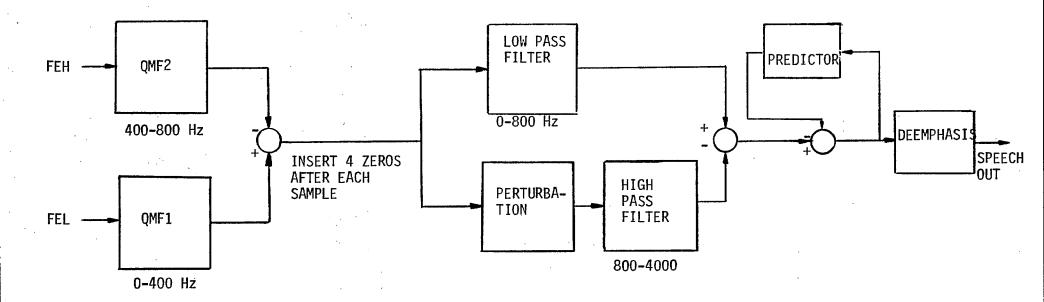
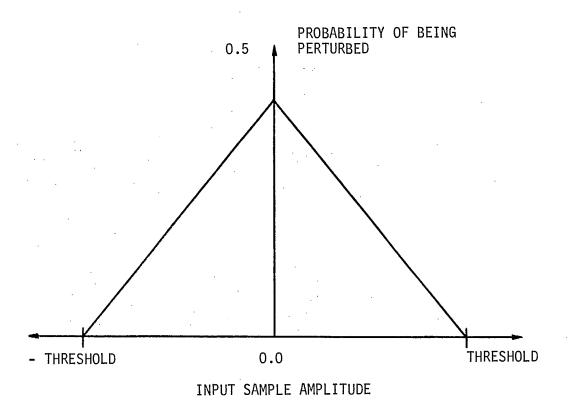


FIGURE 2.1-2 HIGH FREQUENCY REGENERATION WITH AN UNPERTURBED BASEBAND



THRESHOLD = 1/8

FIGURE 2.1-3 CONDITIONAL PERTURBATION PROBABILITY AS A FUNCTION OF INPUT SAMPLE AMPLITUDE

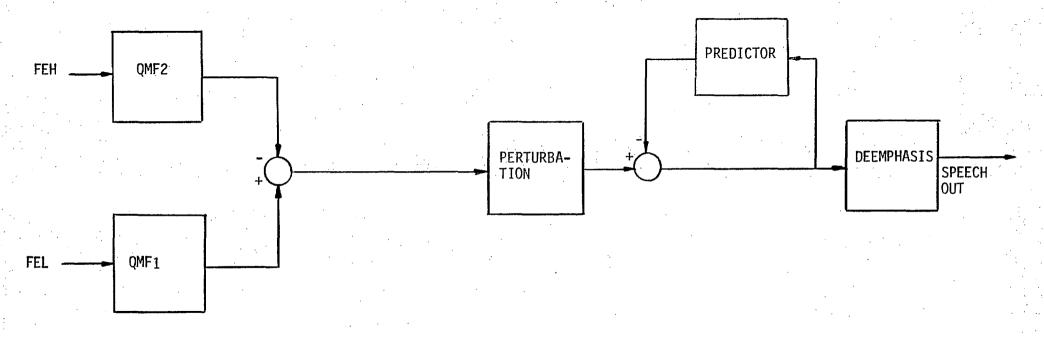


FIGURE 2.1-4 HIGH FREQUENCY REGENERATION WITH A PERTURBED BASEBAND (PERT)

2.2 5-Bit Adaptive Residual Quantizer for 9.6 KBPS Simulation

The existing muldem simulator firmware implements a 2-bit adaptive residual quanitzer which generates 64 bits of residual data for each 20 msec speech frame. These residual data together with the quantized reflection coefficients (22 bits) and housekeeping overhead (10 bits) produce 96 bits of information per speech frame at a rate of 4.8 KBPS. Through the use of a 5-bit adaptive residual quantizer, the information rate will be doubled to 9.6 KBPS. It is expected that the quality of RELP coded speech with 5-bit residual quantization will be perceptibly indistinguishable from that of RELP coded speech without residual quantization.

An adaptive quantizer is one whose step size Δ_n is updated every time an input sample X_n is received. The step size is updated according to the following equation:

$$\Delta_{n} = \begin{cases} \Delta_{n-1} \cdot QMLT(i+1); & i\Delta \leq X_{n} < (i+1)\Delta \text{ and} \\ 0 \leq i \leq N-2 \\ \Delta_{n-1} \cdot QMLT(N); & (N-1)\Delta \leq X_{n} \end{cases}$$
 [2.1]

where N is half the number of quantizer levels, n is the iteration number and QMLT(i) is a set of N postive multiplier constants which are selected to maximize the signal-to-quantization noise ratio based on the statistical property of the quantizer input samples. In addition, the step size is also bounded between Δ_{\min} and Δ_{\max} . A mid-tread/mid-rise adaptive quantizer characteristic [2] is employed according to Figures 2.2-1 and 2.2-2. The mid-tread quantizer performs like a squelch circuit which attempts to eliminate any residual noise in the absence of speech energy. A step size threshold determines which of two quantizer characters (mid-tread or mid-rise) should be employed to quantize the current input sample. The values 1 through N-1 in units of step size are called the quantizer breakpoints, while the values 1/2 through (2N-1)/2 are the quantizer output levels in units of step size.

In the actual implementation of the 5-bit adaptive quantizer, all signed numbers have to be scaled down to between -1 and +1 as a result of integer arithmetic performed by the 8086 subsystem. The breakpoints and output levels are therefore normalized by N or 16 resulting in numbers between -1 and +1. This calls for the establishment of a new step size parameter FSV which is related to the former step size Δ as follows:

$$FSV = 16 \cdot \Delta$$
 [2.2]

The FSV value is updated in pretty much the same way as Δ defined in Equation 2.1. The FSVMN, FSVMX and FSVTH values are chosen to be 10, 1000 and 15 respectively, identical to those used in the 2-bit quantizer. Other parameters of the 5-bit adaptive quantizer is given in Table 2.2-1. The set of multiplier constants QMLT was adopted from Jayant [3] and was

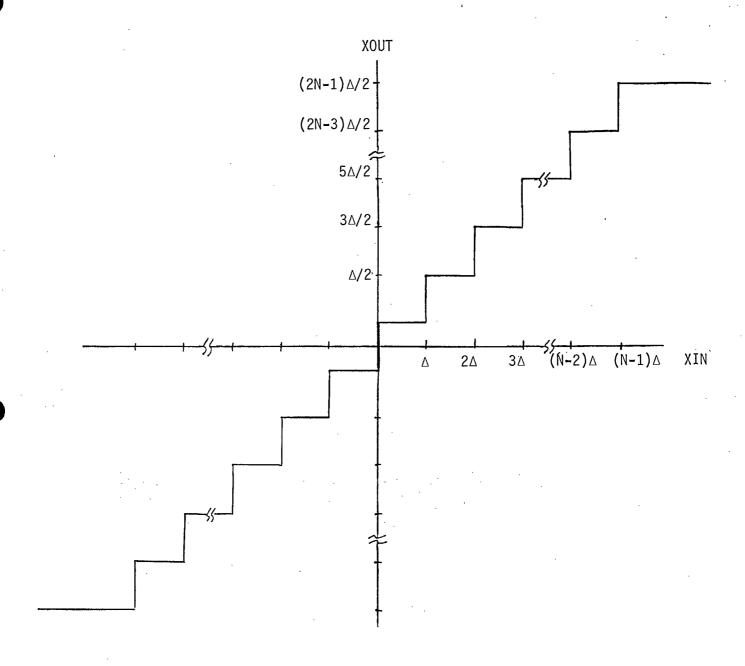


FIGURE 2.2-1 MID-RISE QUANTIZER CHARACTERISTIC FOR STEP SIZE GREATER THAN OR EQUAL TO A PRESET THRESHOLD

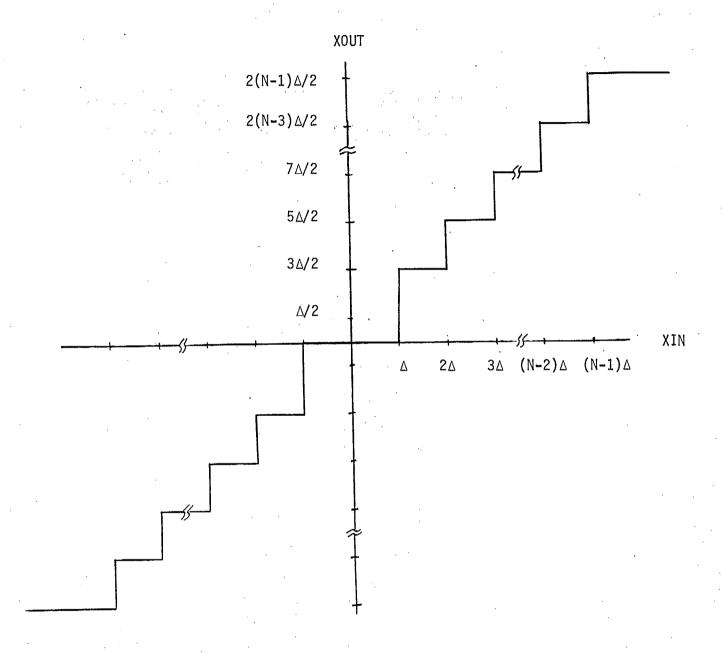


FIGURE 2.2-2 MID-TREAD QUANTIZER CHARACTERISTIC FOR STEP SIZE LESS THAN THE PRESET THRESHOLD

TABLE 2.2-1 5-BIT ADAPTIVE RESIDUAL QUANTIZER PARAMETERS

Mul TTD) TEDG		NORMALIZE		NORMALIZED OUTPUT LEVELS		
MULTIPLIERS		BREAKPOIN	115	UUTPUT LE	VELS	
QMLT(1)	0.85	XQ(1)	1/16	Y/Q(1)	1/32	
-QMLT(2)	0.85	XQ(2)	2/16	Y/Q(2)	3/32	
QMLT(3)	0.85	XQ(3)	3/16	Y/Q(3)	5/32	
QMLT(4)	0.85	XQ(4)	4/16	Y/Q(4)	7/32	
QMLT(5)	0.85	XQ(5)	5/16	Y/Q(5)	9/32	
QMLT(6)	0.85	XQ(6)	6/16	Y/Q(6)	11/32	
QMLT(7)	0.85	XQ(7)	7/16	Y/Q(7)	13/32	
QMLT(8)	0.85	XQ(8)	8/16	Y/Q(8)	15/32	
QMLT(9)	1.2	XQ(9)	9/16	Y/Q(9)	17/32	
QMLT(10)	1.4	XQ(10)	10/16	Y/Q(10)	19/32	
QMLT(11)	1.6	XQ(11)	11/16	Y/Q(11)	21/32	
QMLT(12)	1.8	XQ(12)	12/16	Y/Q(12)	23/32	
QMLT(13)	2.0	XQ(13)	13/16	Y/Q(13)	25/32	
QMLT(14)	2.2	XQ(14)	14/16	Y/Q(14)	27/32	
QMLT(15)	2.4	XQ(15)	15/16	Y/Q(15)	29/32	
QMLT(16)	2.6			Y/Q(16)	31/32	

optimized for PCM speech. No perceptible differences in the quality of RELP coded speech were observed between 5-bit residual quantization and no residual quantization. A listing of the muldem simulator firmware with a 5-bit quantizer is given in Appendix B2.

In the process of modifying the muldem simulator firmware to accommodate the 5-bit adaptive residual quantizer, the software error handling routines were also altered. Up-to-date listings of the muldem simulator firmware with 2-bit residual quantization and no residual quantization are given in Appendix B. The muldem simulator firmware periodically checks software counter and buffer pointer values against their expected values and detects arithmetic overflow conditions to ensure proper firmware operation. In the existing firmware, the residual buffer service counter is checked against a range of values every time the buffer requires service, i.e., read a residual sample and store its quantized value back to the buffer. If the service counter value exceeds a maximum; i.e., too many residual samples in the buffer are not quantized, it is possible for the 8086 CPU to output a residual sample which has not been quantized. Such an error condition will cause the CPU to stop processing indefinitely and flash the double HEX digit LED display (alternating between "00" and "FF"). In the new firmware, such an error condition will increment the most significant digit of the HEX display and reinitialize the CPU such that processing may continue. The existing I/O buffer pointer errors cause the CPU to reset the pointer values and increment the HEX display. As the result of the modification, only the least significant HEX digit is incremented. Overflow error handling remains the same as before and will output "FF" to the HEX display. Under normal operation, the above error conditions will never occur. However, static discharge or edge connector contact problems can occasionally cause a RAM error or noise on the interrupt lines which will in turn produce counter and/or pointer errors.

DIAGNOSTIC RHYME TESTS

3.1 Introduction

The Diagnostic Rhyme Test (DRT) is a method for measuring speech intelligibility. The DRT not only measures total intelligibility but also gives diagnostic scores which measure how well a system preserves six phonetic features. In this test, listeners have to recognize the correct word spoken out of a pair of words given on the answer sheet. These paired words differ only in the presence or absence of one of the six phonetic attributes in the initial consonent.

Diagnostic rhyme tests were performed to measure the effects of RELP coding at 4.8 KBPS and 9.6 KBPS on speech intelligibility. These tests were performed with and without the presence of armoured personnel carrier or helicopter noise.

3.2 Acoustic Signal to Noise Ratio Measurement

Acoustic Signal to Noise Ratio (ASNR) is defined as the ratio of the average peak power of all the words recorded on the DRT tape (four tests) to the average power of the background noise in dB. Speech is first band passed (70-3500 Hz) and the peak power of a spoken word is measured using a fast responding true RMS (Root Mean Square) to DC converter circuit as shown in Figure 3.2-1 and 3.2-2. The measured peak RMS voltages of all words on the DRT tape are listed in Appendix C. The average peak power is obtained through summing the square of all peak RMS voltages and dividing by the total number of words. The type of background noises used are those of M113 armoured personnel carrier and the CH147 Chinook helicopter recorded on CRC supplied tape. The power of these continuous noises are also measured with the true RMS to DC converter. levels were observed to fluctuate slightly with time (about 2 dB peak-Therefore, five short-term power measurements were made at different intervals on each noise tape. The average noise power was computed in a similar fashion as the average word power.

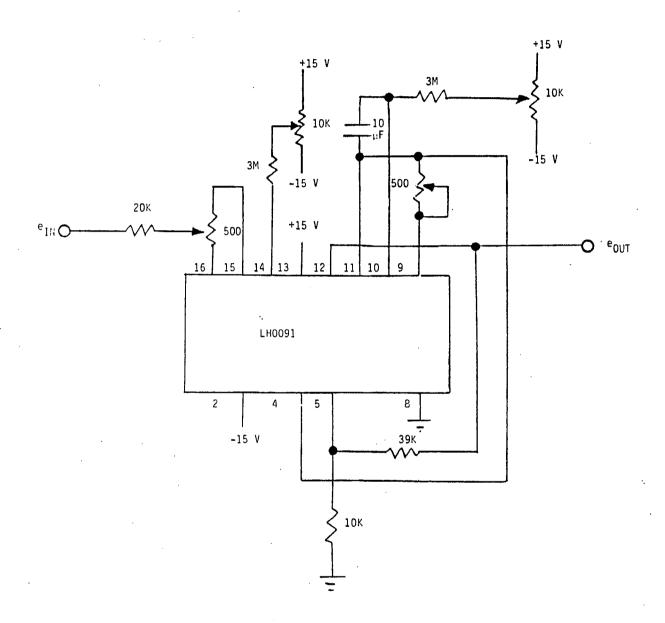


FIGURE 3.2-1 SCHEMATIC OF TRUE RMS TO DC CONVERTER

FIGURE 3.2-2 TRUE RMS WAVEFORM OF A TYPICAL SPOKEN WORD OBSERVED ON AN OSCILLOSCOPE

3.3 Preparation of the DRT Tapes

The noise tape and the DRT tape were played by two cassette recorders (Hitachi DE95). The outputs of the recorders were fed to a summing amplifier (Realistic 32-1200) whose output was recorded on a third cassette recorder (Technics M240X). The average noise power bandlimited to 700-3500 Hz was adjusted to 10 dB below the average peak word power (166 mV RMS). Two recordings being DRT in the presence of helicopter noise and DRT in the presence of armoured personnel carrier noise were produced. These noisy DRT recordings and the original high quality DRT recording were played through the RELP codec real-time simulator whose output was again recorded. A total of 40 DRTs with different coding rates and background noise types were produced and the order of presentation was randomized according to Table 3.3-1. These DRTs were then broken down into five separate recordings, each consisting of eight different DRTs. One recording was to be presented to the test subjects each day to minimize the effects of fatigue, and possibly, adaptation.

3.4 Test Procedure

The five recordings were presented to four test subjects over a period of five days. The tests were performed in a quiet room, using a Hitachi DE95 tape deck and a Vector Research model VR-5000 receiver which drives a KEF speaker. Two male subjects and two female subjects with English as their first language were selected. None of them were familiar with RELP coded speech nor with diagnostic rhyme testing.

TABLE 3.3-1 SCHEDULE OF TESTS

DAY	PART	TEST	CODING	BACKGROUND NOISE
1 1 1 1 1 1 1 1	1 2 3 4 5 6 7 8	1 2 3 4 3 2 1	RELP 9.6 RELP 4.8 RELP 4.8 RELP 4.8 None RELP 4.8 RELP 4.8 RELP 4.8 RELP 4.8	Helicopter Helicopter None Helicopter Armoured Car Armoured Car None
2 2 2 2 2 2 2 2	1 2 3 4 5 6 7 8	3 2 1 4 3 4 1 2	RELP 4.8 RELP 4.8 None None RELP 4.8 RELP 9.6 None PERT 4.8	Helicopter None None Armoured Car Armoured Car Armoured Car Helicopter None
3333333	1 2 3 4 5 6 7 8	2 1 4 3 2 1 4 3	None RELP 4.8 RELP 4.8 RELP 9.6 None RELP 4.8 RELP 9.6 RELP 9.6	Helicopter Armoured Car Helicopter None None None Helicopter Armoured Car
4 4 4 4 4 4	1 2 3 4 5 6 7 8	4 3 2 1 2 3 4 1	None RELP 4.8 RELP 9.6 None RELP 9.6 PERT 9.6 None RELP 9.6	None None Helicopter Armoured Car None None Helicopter Armoured Car
5 5 5 5 5 5 5 5 5 5	1 2 3 4 5 6 7 8	3 4 1 2 3 2 1	None RELP 9.6 PERT 9.6 None RELP 9.6 RELP 9.6 RELP 4.8 RELP 4.8	None None None Armoured Car Helicopter Armoured Car Helicopter None

3.5 Analysis of Test Results

3.5.1 <u>Introduction</u>

The answers given for all the DRTs were entered into a computer. The use of a computer allowed fast, reliable marking of the tests and permitted summaries of the scores based on the person taking the test, coding, background noise and attribute. All scores are given in unadjusted percent correct to allow comparison with INRS results.

The adjusted percent correct score may be obtained from the unadjusted score as follows:

$$Pa = 2 * Pc - 100$$

3.5.2 DRT Scores and Observations

A summary of the results of the DRTs is given in Table 3.5.2-1. The DRT scores over the five days of testing showed no trend which would indicate that the subjects were learning the tests or becoming better at understanding RELP speech.

Table 3.5.2-2 gives a summary of the DRT results for the various coding methods and background noises. RELP coding at 9.6 Kbps results in a 3 percent drop in DRT score over raw speech. Dropping the data rate to 4.8 Kbps decreases the DRT scores by a further 3 The perturbation algorithm does not perform as well as RELP. The RELP scores are about 2 to 3 percent higher than the PERT scores. The perturbation algorithm perturbs the low frequencies as well as the high frequencies in the residual. This may have caused the lower score in the graveness attribute which is mostly responsible for the score differences. In the presence of armoured personnel carrier noise, 9.6 Kbps RELP coding is only marginally affected (-2%) while 4.8 Kbps RELP has a 5 percent reduction in score (Figure 3.5.2-2). Figure 3.5.2-3 shows that RELP coded speech at 4.8 Kbps is just about as intelligible as that at 9.6 Kbps in the presence of helicopter noise. This would suggest that 4.8 Kbps RELP coding may be adequate for noisy environment applications in spite of the fact that 9.6 Kbps RELP coding is judged to be more acceptable.

Of the six speech attributes of speech measured in the DRTs, sustention and graveness have the lowest scores for the various coding and noise conditions. RELP coding has the greatest difficulty with the sustention attribute, often causing no sustention attribute to be heard in the coded speech when it was actually present in the orig-

inal speech. This causes words like "sheet" to be coded like "cheat". The bias added to speech is summarized in Table 3.5.2-3.

Since the words in the DRTs varied by greater than 12 dB in amplitude, an attempt was made to see if this affected the scores. Table 3.5.2-4 shows that there is less than one dB difference among the average powers of words in each attribute. This suggests that it is unlikely that the difference in scores among attributes is caused by different signal to noise ratios. There is no trend of decreasing score with increasing noise levels, Table 3.5.2-5. This is likely due to the fact that the difference in difficulty among the word groups has a greater effect on the score.

Table 3.5.2-6 shows the standard errors of the DRT scores among the four test subjects. In all cases, the standard errors are less than about 3 percent. This may suggest that there is no great variation in test scores among subjects.

TABLE 3.5.2-1 DRT SCORES BY PERSON AND TIME OF TEST

PERCENT CORRECT DIAGNOSTIC RHYME SCORES Score as a function of person and time of test $% \left(1\right) =\left\{ 1\right\}$

						•			
DAY	PART	TEST	CODING	BACKGROUND NOISE	ALVA	LYNNE	PETER	WOLF	AVERAGE
1 1 1 1 1 1	1 2 3 4 5 6 7 8	1 2 3 4 3 2 1	RELP 9.6 RELP 4.8 RELP 4.8 None RELP 4.8 RELP 9.6 PERT 4.8	Helicopter Helicopter None Helicopter Armoured Car Armoured Car None None	85 90 77 96 85 96	94 81 90 85 92 81 96 88	88 83 90 81 98 85 98	88 92 90 92 96 90 98	89 85 90 84 95 85 97
2 2 2 2 2 2 2 2 2 2	1 2 3 4 5 6 7 8	3 2 1 4 3 4 1 2	RELP 4.8 RELP 4.8 None None RELP 4.8 RELP 9.6 None PERT 4.8	Helicopter None None Armoured Car Armoured Car Armoured Car Helicopter None	88 94 100 98 85 96 100 92	79 92 100 98 90 96 98	81 94 100 98 83 90 98 88	90 100 100 96 94 96 100	84 95 100 97 88 94 99
3 3 3 3 3 3 3 3 3	1 2 3 4 5 6 7 8	2 1 4 3 2 1 4 3	None RELP 4.8 RELP 9.6 None RELP 4.8 RELP 9.6 RELP 9.6	Helicopter Armoured Car Helicopter None None Helicopter Armoured Car	94 90 90 96 98 94 92 81	98 90 85 92 100 92 79 90	100 94 90 96 100 94 94	98 94 81 94 100 94 88 96	97 92 86 94 99 93 88 90
4 4 4 4 4 4	1 2 3 4 5 6 7 8	4 3 2 1 2 3 4 1	None RELP 4.8 RELP 9.6 None RELP 9.6 PERT 9.6 None RELP 9.6	None None Helicopter Armoured Car None None Helicopter Armoured Car	100 85 88 100 96 90 100	100 88 88 96 96 90 100 98	98 92 88 100 96 94 96 100	100 94 92 98 98 96 94	99 90 89 98 96 92 97 98
55555555	1 2 3 4 5 6 7 8	3 4 1 2 3 2 1 4	None RELP 9.6 PERT 9.6 None RELP 9.6 RELP 4.8 RELP 4.8	None None None Armoured Car Helicopter Armoured Car Helicopter None	100 98 94 96 88 92 96 94	96 94 90 100 83 96 88 100	100 96 96 98 88 92 94 96	98 98 98 100 90 92 96 94	98 96 94 98 87 93 93

TABLE 3.5.2-2 DRT SCORES BY ATTRIBUTE

PERCENT CORRECT DIAGNOSTIC RHYME TEST SCORES Score as a function of attribute

CODING	BACKGROUND NOISE	VOICING	NASALITY	SUSTEN- TION	SIBILA- TION	GRAVE- NESS	COM- PACT- NESS	TOTAL
None	None	99	100	99	100	98	100	99
None	Armoured Car	98	100	94	100	95	98	97
None	Helicopter	100	100	96	100	92	100	98
RELP 4.8	None	95	100	83	93	86	99	93
RELP 4.8	Armoured Car	91	96	82	83	83	95	88
RELP 4.8	Helicopter	93	95	68	90	81	93	87
RELP 9.6	None	97	100	91	95	94	99	96
RELP 9.6	Armoured Car	98	100	83	93	92	97	94
RELP 9.6	Helicopter	95	97	71	. 86	84	95	88
PERT 4.8	None	97	98	88	91	77	94	91
PERT 9.6	None	98	100	86	91	84	100	93

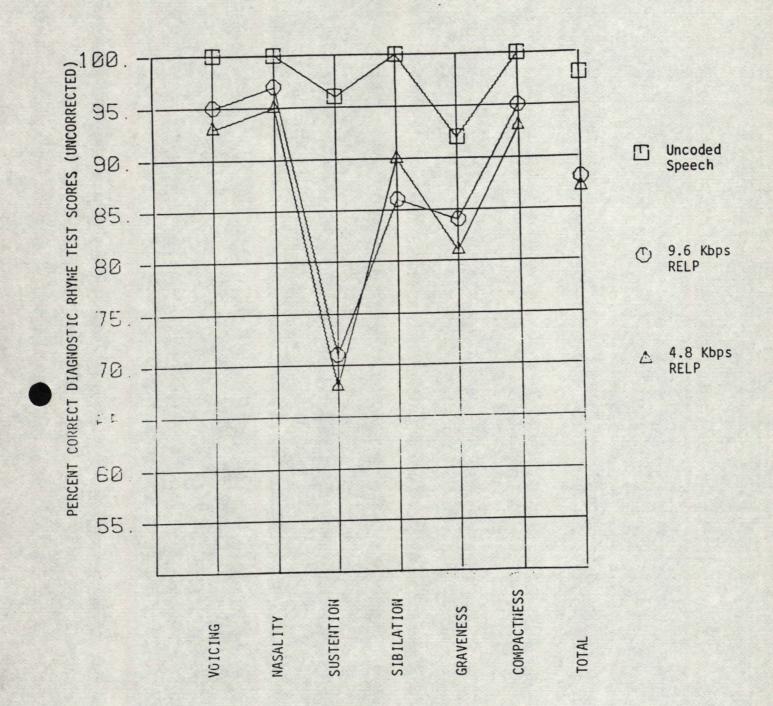


FIGURE 3.5.2-3 DIAGNOSTIC RHYME TEST RESULTS FOR RELP CODING WITH HELICOPTER BACKGROUND NOISE

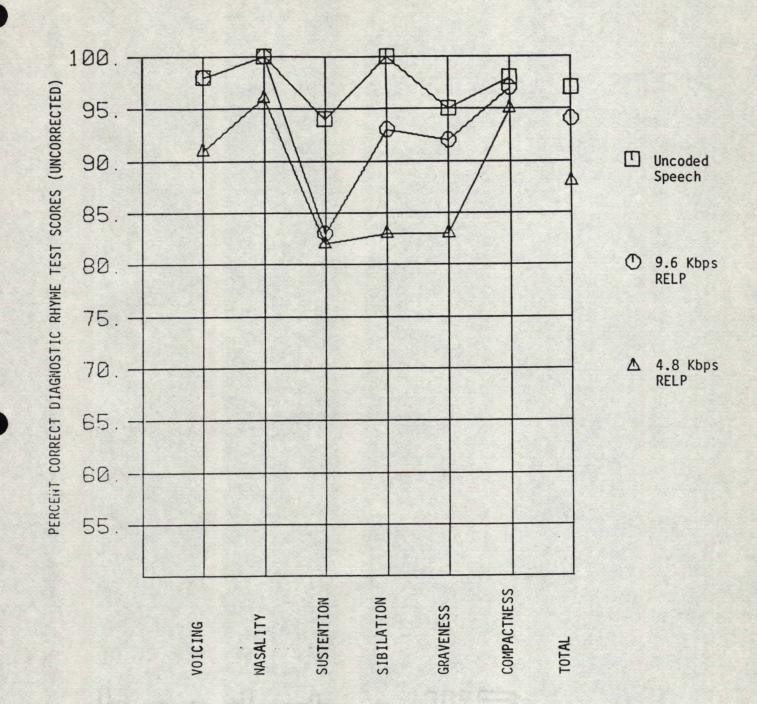


FIGURE 3.5.2-2 DIAGNOSTIC RHYME TEST RESULTS FOR RELP CODING WITH ARMOURED PERSONEL CARRIER NOISE

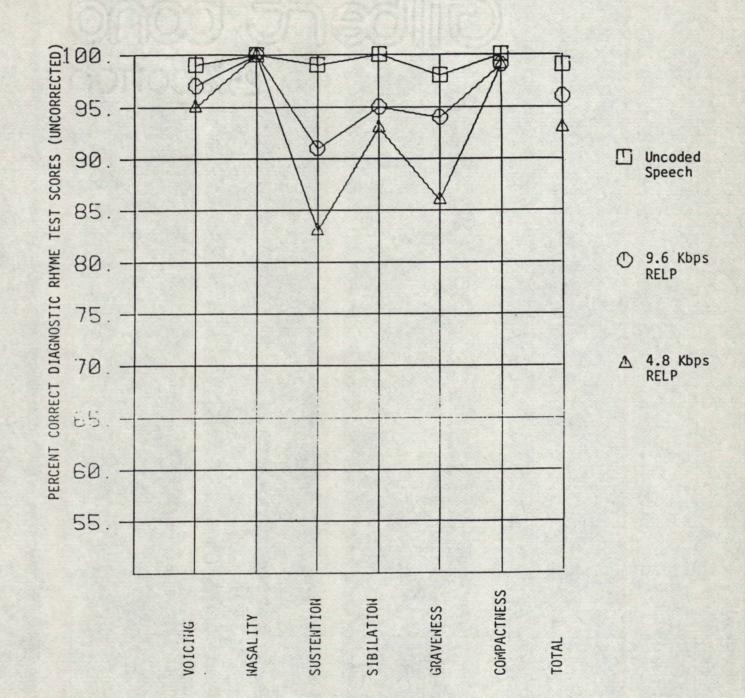


FIGURE 3.5.2-1 DIAGNOSTIC RHYMES TEST RESULTS FOR RELP CODING WITHOUT NOISE

TABLE 3.5.2-3 BIAS IN DRT SCORES

CODING	BACKGROUND NOISE	VOICING	NASALITY	SUSTEN- TION	SIBILA- TION	GRAVE- NESS	COMPACT- NESS
None	None	2	0	-2	0	-5	0
None	Armoured Car	5	0	-3	0	0	5
None	Helicopter	0	0	-1	0	0	0
RELP 4.8	None	10	0	-18	-8	-6	2
RELP 4.8	Armoured Car	18	-5	-17	-13	0	-3
RELP 4.8	Helicopter	6	-7	-18	-8	5	0
RELP 9.6	None	7	0	-16	-6	3	2
RELP 9.6	Armoured Car	4	0	-28	-15	0	0
RELP 9.6	Helicopter	3	-3	-5	-9	4	-3
PERT 4.8	None	.4	6	-19	-10	-13	-4
PERT 9.6	None	-4	0	-21	- 9	-6	0

NOTE:

The above scores are the difference between the percent correct with the attribute present and the percent correct with the attribute absent. A positive number indicates that the attribute is heard when it is not present.

TABLE 3.5.2-4 AVERAGE WORD POWER VERSUS ATTRIBUTE

<u>ATTRIBUTE</u>	AVERAGE RMS VOLTAGE	POWER DEVIATION (dB)
VOICING	163.94	-0.15
NASALITY	161.29	-0.29
SUSTENTION	167.22	0.03
SIBILATION	168,92	0.11
GRAVENESS	160.09	-0.35
COMPACTNESS	178.19	0.58
OVERAL'L AVERAGE	166.72	0.00

TABLE 3.5.2-5 DRT SCORES AS A FUNCTION OF SIGNAL TO NOISE RATIO

CODING	BACKGROUND NOISE	1 <u>dB</u>	4 <u>dB</u>	7 <u>dB</u>	10 dB	13 <u>dB</u>	16 <u>dB</u>	19 <u>dB</u>	NO NOISE
None None None	None Armoured Car Helicopter	. '	100 100	97 94	97 99	97 98	?100 ?100		99
RELP 4.8 RELP 4.8 RELP 4.8	None Armoured Car Helicopter		93 92	84 78	89 90	89 88	? 94		93
RELP 9.6 RELP 9.6 RELP 9.6	None Armoured Car Helicopter		98 89	91 85	96 90	94 87	? 92 ?100		96
PERT 4.8	None								91
PERT 9.6	None			. '					93
PERT 9.6	None	٠							93

NOTE: No entry in table if less than 10 sample points. A '?' precedes scores with less than 48 samples.

TABLE 3.5.2-6 DRT SCORES AND STANDARD ERRORS

CODING	BACKGROUND NOISE	AVERAGE SCORE (%)	STANDARD ERROR (%)
None	None	99.35	0.23
None	Armoured Car	97.40	0.73
None	Helicopter	97.91	0.46
RELP 4.8	None	92.60	1.08
RELP 4.8	Armoured Car	88.38	2.31
RELP 4.8	Helicopter	86.56	2.25
RELP 9.6	None	95.96	1.00
RELP 9.6	Armoured Car	93.50	1.71
RELP 9.6	Helicopter	88.02	1.27
PERT 4.8	None	90.63	1.65
PERT 9.6	None	93.23	2.81

3.5.3 Comparison With The INRS Results

Nakatsui et al [4] gives results of DRTs performed on an earlier version of the RELP Algorithm developed at INRS. The INRS RELP algorithm was simulated on a PDP-11 minicomputer. It has several differences from the current hardware implementation, such as the use of floating point instead of integer arithmetic; interpolation of the predictor coefficients between frames, to avoid abrupt amplitude changes at the frame boundary; use of two residual gain parameters and several other small differences. It is not known if the same speaker was used for the INRS DRTs. Despite these differences, the DRT scores for the two implementations were remarkably similar with three exceptions.

The hardware implementation has lower scores on the sustention attribute (Figure 3.5.3-1). This could be due to the changes in the algorithm such as the removal of predictor coefficient interpolation, or the use of a different speaker on DRT recording.

The overall scores in the presence of noise were higher for the hardware implementation than the INRS software simulation (Figure 3.5.3-2). One explanation for the difference is attributed to the fact that white noise was used by INRS instead of periodic noise used in this study. The high peak-to-average power ratio in white noise could have impaired normal hearing to a greater extent than the periodic noises of equal power levels. Such a phenomenon manifested itself in the score discrepancy between helicopter noise and armoured personnel carrier noise. The DRT scores in the presence of helicopter noise were generally lower as a result of a higher peak-to-average noise power ratio. Subjectively, noise with a higher peak-to-average power ratio sounds more abrupt and annoying.

The scores for the uncoded speech were higher in this report than those described in INRS (Figures 3.5.3-3 and 3.5.3-4). The uncoded speech data base used by INRS was band-limited to 3.2 KHz but not in this study. Another reason for the inconsistency could be due to speaker dependence.

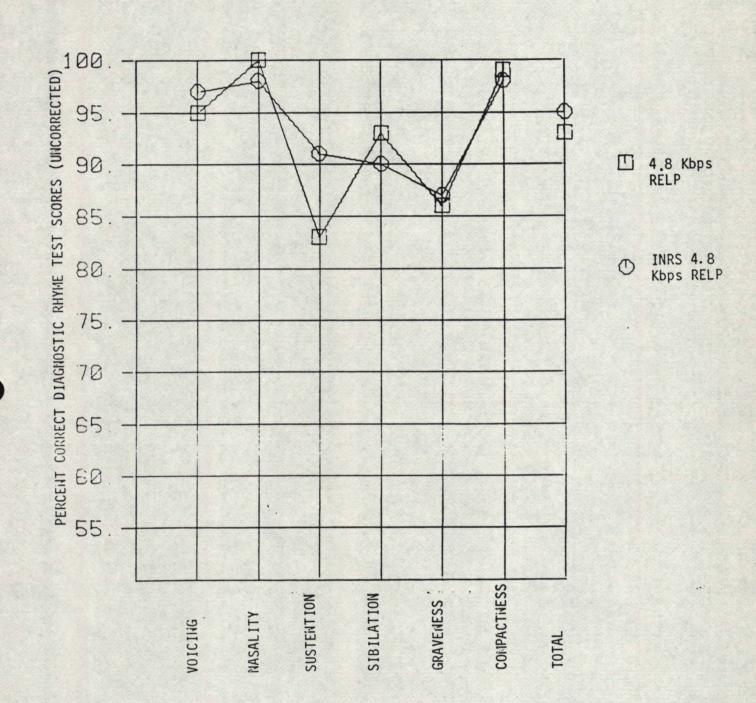


FIGURE 3.5.3-1 COMPARISON WITH INRS DIAGNOSTIC RHYME TEST SCORES FOR 4.8 KBPS RELP

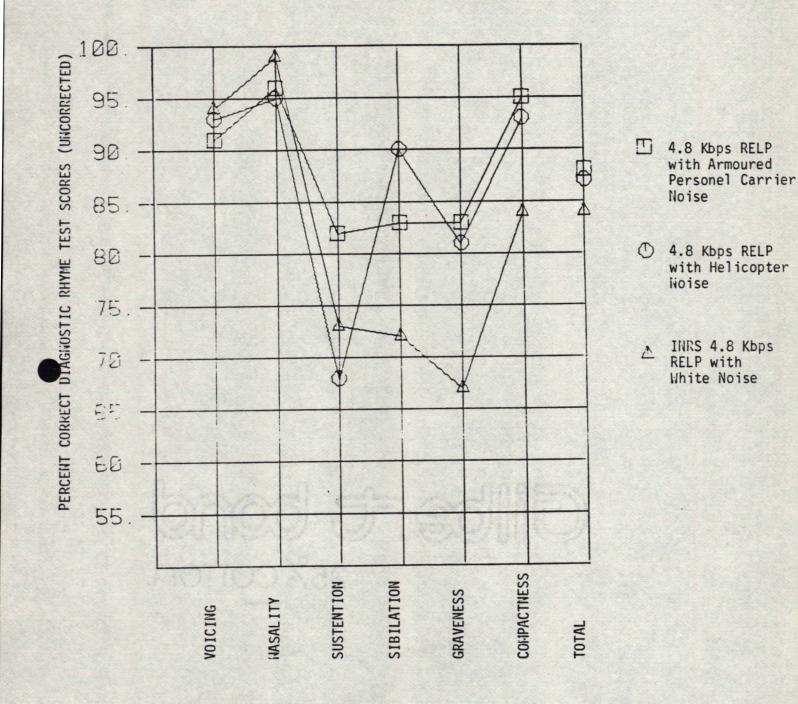
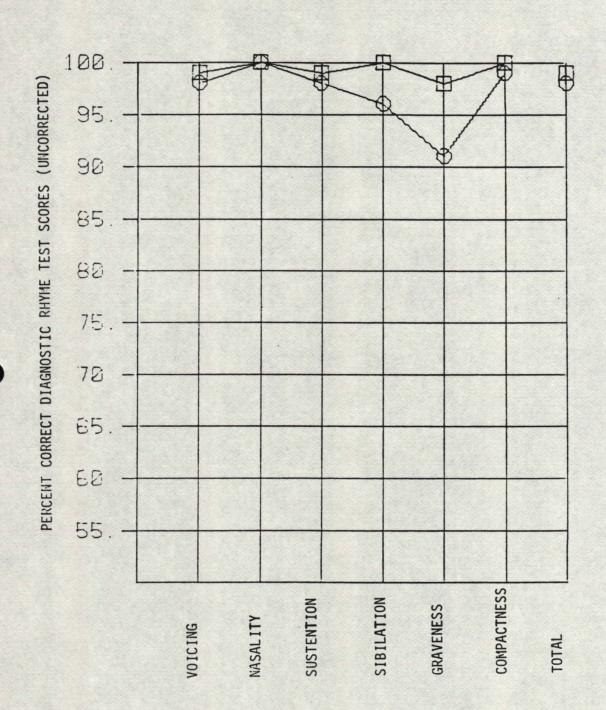


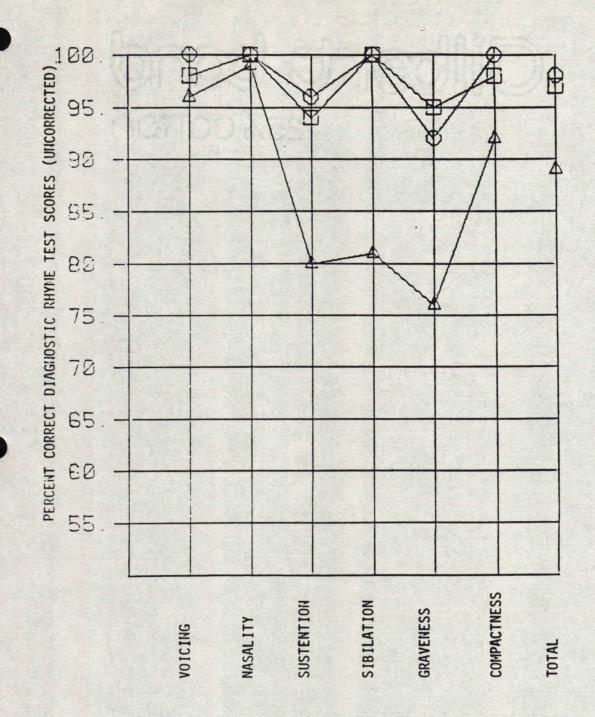
FIGURE 3.5.3-2 COMPARISON WITH INRS DIAGNOSTIC RHYME TEST SCORES FOR 4.8 KBPS RELP WITH 10 dB SIGNAL TO NOISE RATIO



Raw Speech

Bandlimited
Speech
(INRS Result)

FIGURE 3.5.3-3 DRT SCORES FOR UNCODED SPEECH WITHOUT NOISE



- Raw Speech with Armoured Personel Carrier Noise
- ↑ Raw Speech with Helicopte Noise
- A Raw Bandlimited Speech with White Noise (INRS Result)

FIGURE 3.5.3-4 DRT SCORES FOR UNCODED SPEECH AT 10 dB SIGNAL TO NOISE RATIO

4. <u>CONCLUSIONS</u>

Diagnostic Rhyme Tests have been performed on the RELP coded speech using the RELP codec real-time hardware simulator. RELP coding at 4.8 Kbps and 9.6 Kbps together with an alternate full-band residual regeneration method have been simulated. The effects of helicopter noise and armoured personnel carrier noise on RELP coded speech are also measured.

Intelligibility as measured by the Diagnostic Rhyme Tests is 96% for RELP coded speech at 9.6 Kbps and 93% for RELP coded speech at 4.8 Kbps. These represent, respectively, a three and six percent degradation over uncoded speech. Further, 9.6 Kbps RELP coding is only degraded marginally by two percent in the presence of armoured personnel carrier noise while 4.8 Kbps RELP coding is degraded by as much as five percent. RELP coding at 9.6 Kbps and 4.8 Kbps have almost equal intelligibility in the presence of helicopter noise. This may suggest that 4.8 Kbps RELP coding is adequate for applications in noisy environment.

The new full-band residual regeneration method (PERT) scores two to three percent lower than the rectification method proposed by INRS although speech quality is judged to be slightly more acceptable. It is therefore possible to tradeoff minor performance degradation for a seventy percent reduction in the amount of synthesizer processing.

In spite of the various simplifications in the hardware simulation, results are remarkably similar to the INRS software simulation. A 1.5 percent reduction in DRT score has been observed on the hardware RELP simulation at 4.8 Kbps.

5. <u>REFERENCES</u>

- [1] Higgins, A. and R. Viswanathan, "New High-Frequency Regeneration Techniques for Voice-Excited Speech Coders", Proceedings of the 98th Meeting of the Acoustic Society of America, Salt Lake City, Utah, November 26-30, 1979.
- [2] Crochiere, R.E., "A Mid-Rise/Mid-Tread Quantizer Switch for Improved Idle-Channel Performance in Adaptive Coders", B.S.T.J, October, 1978, pp. 2953-2955.
- [3] Jayant, N.S., "Adaptive Quantization with a One-Word Memory", <u>B.S.T.J</u>, September, 1973, pp. 1118-1144.
- [4] Nakatsui, M., D. Stevenson and P. Mermelstein, "Subjective Evaluation of a 4.8 KBPS Residual-Excited Linear Prediction Coder", A Manuscript prepared for IEEE Transactions on Communications, 1981.

APPENDIX A
SYNTHESIS SIGNAL PROCESSOR FIRMWARE LISTINGS

APPENDIX A SYNTHESIS SIGNAL PROCESSOR FIRMWARE LISTINGS

A1. RESIDUAL REGENERATION THROUGH INTERPOLATION OF LOW PASS COMPONENT BY ZERO AMPLITUDE INSERTION, AND SAMPLE POSITION PERTURBATION OF HIGH FREQUENCY COMPONENT

1	00000	TITLE SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 10,1982
2	00000	LIST X
3	00000	**************************************
4	00060	** PROJECT: 472A, RELP CODEC *
5	00000	# BOARD: SYNTHESIS PROCESSOR #
6	00000	# DEVICE: 6 MICROCODE PROMS, 3636 or 3628
7	00000	# LOCATION: 634,P1,P15,P29,P43,P57 #
8	00000	* PROG.DEV#: 60-0690 - 60-0695 *
9	00000	## ASSEMBLER: META, April 82 *
10	00000	;* FILE: SYNTHESIS.ASM *
11	96969	(K REVISIOM: SEFT 10:1982
12	00000	* UPDATE TABLE:
13	00000	\$\$ SETT 9: Initial folding exp. hopeful *
	60000	# Hissins method implemented. *
15	50000	그리아 그들은 이 경우를 느리는 것이 모든 그들은 사람들이 아름다면 보고 있는 것이 없어 없어지 않는 것이 없다면 하는데 없었다.
16	00000	JULY 7: Each filter assigned sain in PROM *
17	90000	* JULY 7: More precision maintained.
18	00000	* JUNE 28: Filter Gains Adjusted. *
19	-06060	## JUNE 14: Mods made to use subtract inst corr #
20	00000	4% AFRIL 23 - MAY 23: Initial prosam entre JM *
21	00000	12
22	00000	

LINE	ADDR	SYN	THESIS PRO	GESSOR ASSEMBLER SOURCE FILE, SEPT 10 PAGE
24	00000	***	*****	*********************
25	00000	**	TA	BLE OF CONTENTS *
26	00000	**		
27	00000	14	PAGE	DESCRIPTION *
28	00000	11		*
29	00000	**	1	File Header and Update Table *
30	00000	**	2	Table of Contents *
31	00000	*	3	Filter size specifications *
32	00000	;*	4	Rins Buffer memors map
33	00000	**	5	Scratch Fad memory map
34	00000	**	5	Coefficient From memory map
35	00000	11	7	Insut/Gutsut definitions *
36	05060	12	8	PROCEDURE start_up; *
37	00000	**	9	PROCEDURE main;
36	00000	1%	14	PROCEDURE operations_done_EVERY_sample *
34	00000	÷ ¥	17	FROCEDURE Quadrature_Mirror_Filter_High *
40	00006	**	20	PROCEBURE Quadrature_Mirror_Filter_Low *
43	60000	: #	27	PROCEDURE Low_Pass_Filter *
42	00000	17	25	PROCEDURE Double_Difference_Filter *
43	00000	13	26	PROCEDURE Hish_Pass_Filter *
44	00000	12	25	PROCEDURE Predict *
45	00000	11	29	PROCEDURE CoPY_PREDictor_Coefficients *
46	06000	18	36	PROCEDURE DE-EMphasize *
47	00000	**	71	Cross Reference Table *
45	00000	1×××	(*************************************	*************************

```
LINE
     ADDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10
                                                   PAGE
    50
     00000 ##
                   FILTER SIZE SPECIFICATIONS
     00000 FF
     00000 J: EQU
                     9 ; Fredictor filter order.
     00000 QMFSZ: EQU 36
                          i Quadrature Mirror Filter Size
     00000 LPFSZ: EQU 31
                          i Low pass filter size
     00000 PREDSZ: ERU 9
 57
                           # Fredictor Filter Size
     00000 DEMSZ: EQU 2
 58
                          i Deemphasis Filter Size
 59
     00000 DLY: EQU 32 | Belay so predictors line up with frame
 60
    00000 ; The following constants define filter sizes in the rectification
     00000 i RELP implementation. They are used here only to specify the location
 61
 62
    00000 : the filters in the coefficient PROM, which is the same for both
    00000 ; the perturbation and rectication implementations.
    00000 PDFSZ! EGU
                    3
 0"
    00000 ARSZ: FRU
 35
65 00000 HPFSZ: ER:
                     33
```

Lint	HITTE	STRIFFENIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10 PAGE 4	
66	00000	*******************	
69	00000	* RING BUFFER MEMORY MAP *	
70	00000	*	
71	00000	**************************************	
72	00000	MFBS: EQU 10(0):% ; Base of quadrature mirror filter	
73	00000	LYBS: EQU 10(QMFB3+5*QMF8Z+10):%; Compensating delay for PERT	
74	00000	PFBS: EQU 10(DLYBS+D#3):%	
75	06/606	UMBS: EQU 10(LPFBS+LFFSZ+D#1):% ; summins node for HPF and LPF path	1
76	00000	REDBS: EQU 10(SUMBS+DLY):% ; Predictor base(101 IS MATCHING DE	E
77	00000	REDOUT:EQU 10(PREDBS+PREDSZ) ; End of predictor	
76	60000	EMBS: EQU 10(PREDBS+PREDSZ):X ; Deemphasis base	
79	00000	ERIPS: EQU 10(DEMBS+D45):% ; Base of compensating delay buffer	
80	00000	WFBS: EQU 10(PERTES:D43):X : End of compensating delay	
61	00000	FROM: EQU 10(HPF38+LPF8Z):% Output of High Free Regeneration	
62	00000	: £00 10(EEM5S+2):%	
83	60000	EAST: EQU 10(D#1023):% Flower location used in double prec	
34	10000	i arithmatic	
85	00006	997: EQU 10(D{1022):X	

LIME	ADDA	SYNTHES	915 PF	RCCESSOR	ASSEMBLER	SOURCE	FILE.	SEFT	10	PAGE	5
87	00000	*******	*4***	******	******	*****	*****	****	***	*****	
88	00000	12 S	CRATC	FAI ME	MORY MAP						
89	00000	188.8888	****	*******	*******	****	*****	****	***	****	
96	00000	P0:	EQU	8(0):%		; ;	ese of	predi	ctor	storage	
91	60000	FIMFO:	EQU	8(F0+J)	:%	; B	ase of	tempo	ersre	predicto	r storage
92	00000	P:	EQU	8(PTMP)+J):2	; L	ast pr	edicto	02 10	ef. input	
93		FE:	EQU	B(F+1):	%	1 F	E(H,L)	stora	se		
74	00000	LSTP:	EQU	8(FE+1	1:%	; L	asi F	used			
95	00000	LSTFEL:	EQU	B(LSTP	1):%	; L	ast FE	Line	it		
96	(0000	LSTFEH!	EQU	8(LSTF	EL+1):%	; L	ast FE	H inst	ıt		
97	00000	1STFE:	ERU	BILSTER	H÷1)1%	; L	ast FE	calcu	late	d	
98		TEHP:	EQL	B(LSTF	E+1)1%	1 1	empora	ry st	92510	location	

LIKE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER S	SOURCE FILE. SEPT 10 PAGE 6
100	00000	*********************	*******
101	00000	** COEFFICIENT PROM MEMORY I	MAP *
102	00000	****************	*********
103	60000	CONO: EQU 8(0):X	f constant zero
104	00000	CON1: EQU 8(1):X	; constant one
165	60000	CONM1: EQU 8(2):%	f constant minus one
106	00000	A: EQU 8(3):%	# Pre-emphasis constant/2
107	00000	CLPF: EQU 8(A+1):%	f Base of low pass filter coef
102	00000	CQMF: EQU 8(CLPF+LPFSZ):%	# Base of quadrature mirror filter
107	06606	CDDF: EQU 8(CQMF+QMF8Z):X	# Base oF double difference filter
110	00000	CHFF: EQU 8(CDDF+DDF82):%	# Base of high pass filter
111	00000	FREIRCLIEGU 8(CHPF4HPF8Z):%	f Scale factor for predictor
112	00000	COMPORTRE EQUIB (PREDSCL):%	f Scale factor is 1/4
113	00000	COW2: EQU 3(PREDSCL+1):%	
114	00000	CGN4: EQU 8(CDN2+1):%	
115	00000	COMBI EGU 8/CON1+31;X	
116	00000	CON128: EQU 8(CON8+1):%	
117	00000	GNIMFUT.F50 8(00M12P41):%	
115	00000	GNOWE: EGG 8(GNINPUT+1):X	; Filter sains follow
115	00000	ONLPF: EQU 8(GNQMF+1):X	
120	00000	SWEAR: EGU 8(GNLPF+1):%	
121	90000	GMMPF: EQU 8(GNDDF+1):%	
122	00000	GAPREL: EGU 3(GAMPF+1)::	
123	00000	GNOUT: EGU 8(GNFRED+1):%	
124	(10000)	THRESH: EPU 8(CLFF41):X	# = 526, USED TO AVOID BURNING
125	00000		FROM BEFORE FINDING OPTIMUM VALUE

```
LINE
    APDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 10 PAGE 7
00000 ## INFUT/DUTPUT DEFINITIONS
 126
 00000 TEST: EQUIDANT F Analos test port address
130
     00000 FPRT: ERU B410
                            f Fredictor input port
 13:
                           ; FE input port
     00000 FEFRI: EQU B#11
 132
103 00000 SPRT: ERU B$10 ; Synthesized speech output port
135 00000 ; Test switch input definitions
i Test point select 1 = FEH
i Test point select 2 = FE
139 00000 TSTFEH: EQU .TEST1
     00000 ISTFE: EQU .TEST2
130
140 60000 TETE: EQU .TEST3
                                 Filest point select 3 = E
141- 00000 TETHEF: EGU . TEST4
                                f Test Point select 4 = HPF

      142
      00000 T8T8UM: EQU .TESTS
      # Test point select 5 = SUM HPF & E

      143
      00000 T8T8F8: EQU .TEST6
      # Test point select 6 = SPE

      144
      00000 T8T8: FQ: .TEST7
      # Test point select 7 = S
```

; Start_of_Frame entry point

177 0000A

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 10 PAGE 9
179	0000A	**************************************
180	0000A	## PROCEDURE main#
181	00006	* { This procedure does the calls for all filtering *
182	00004	4% operations in the synthesis processor, as well as all the#
183	00001	## input. All input and output operations are done after #
18-	60000	# S enable soes low but before S clock soes high. Sample #
185	00006	## output takes place after S enable does high and all #
186	00004	## calculations for the sample interval are complete
		;*************************************
188	00000	f Start of sample interval 0
199	9900A	SOFEMI: JMP .SOF.SOFEMI:X ; Wait for start of frame
		00110111 00000000 10100116 11111111 11000000 00000000
190	9000%	
		60010000 00000101 61100110 11111111 11000000 00000000
192		f Start of sample interval 1
193	00000	
		00010000 00000101 01100110 11111111 11000000
195	60001.	f Start of sample interval 2
156		CALL FEVERY:X
		60010000 00000101 01100110 11111111 11000000
198	0000E	• Start of sample interval 3, set FEL
199	0000E	CALL .EVERY: ; Do stuff done every frame
		00010000 00000101 01100110 11111111 11000000
200	0000F	CALL +OMFL:X ; Do Low QMF filter stuff
		60010000 00001010 61110110 11111111 11000000 00000000
202	00010	+ Start of sample interval 4, get P(0)
203	00010	CALL *EVERY:%
		00010000 60000131 01100116 11111111 11000000 00000000
204	60011	LGAD FOR FINAL FINAL P(0)
		11100000 01000001 00100100 00010010 11000001 10000100
205	60012	STMS PTMPO:PTMPO ; Store in PTMPO
		11100000 01000000 10010100 00001001 11000000
207	00013	; Start of sample interval 5, set P(1)
208	00013	CALL FEVERY:X
		20010000 00005101 01100110 11111111 11000000 00060000
209	00014	LSAD PYP ; Input P(1)
		11100000 01600001 00100100 00010010 11000001 10000100
210	00015	STMS (PTMPO+D#1):2,(PTMPO+D#1):X ; Store in Ptmp1
		11100000 01000000 10100100 00001010 11000000
212	00016	Start of sample interval 6, set P(2)
	00016	CALL *EVERY:%
		00010000 00000101 01100110 11111111 11000000
214	00017	LOAD P.P ; Input P(2)
		11100000 01000001 00100100 00010010 11000001 10000100
215	00018	STMS (PTMP0+D42):X; (PTMP0+D42):X; Store in Ptmp2
		11100000 01000000 10110100 00001011 11000000

LINE	ADD-:	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10 PAGE 10
217 216	00019	Figure 1 Start of sample interval 7 CALL FEVERY:%
219		00010000 00000001 01100110 11111111 11000000
		11100000 01000000 10010100 00001001 11000001 10000100 STMS LETF,LETF
		11160000 01000003 01000100 00010106 11000000 00101010
222 223	00018	f Start of sample interval 8, Get FEH CALL .EVERY:%,
224	00010	00010000 00000101 01100110 11111111 11000000
225	0001E	00010060 00000111 11100110 11111111 11000000 000000
225	0001F	ETMS 157P,LSTF 11106000 01090601 01000100 00010100 11000000 00101010
228		5 Start of sample interval 7, set P(3)
229	00026	CALL *EVERY:% D001609
230	00921	LDAN P.P ; Input P(3) 1:100000 0100000 0010010 11000001 10000100
231		STMS (PThF0+D43):%;(PTMF0+D43):%; Store in Ptmp3 11100000 01000000 11030100 00001100 11000000 00101010
232	00023	LOAD (FTMF0+D+1):X,(FTMF0+D+1):X ; Get F(1) for test point 11100000 01000000 10100100 00001010 11000001 10000100
233	00024	57MS LSTP#LSTP 11708000 01007001 01000100 00010100 11000000 00101010
235 238	00025	Start of sample interval 10; set P(4) CGLL :EVERY:X
		00010000 06000.01 01100110 11111111 11000000 00000000
238	00027	11100000 01000001 00100100 00010010 11000001 10000100 STMF (FTMP0+D#4):X*(PTMP0+D#4):X ; Store in Ptmp4
239	00028	1:100000 01000000 11010100 00001101 11000000
240	00029	1:100000 01000000 10110100 00001011 11000001 10000100 STMS LSTP,LSTP
242	06676	11100000 01000001 01000100 00010100 11000000
243	00024	CALL *EVERY:% D0010000 00000101 01100110 11111111 11000000
244	0002B	LOAD F,F ; Input F(5) 11100000 01000001 00100100 00010010 11000001 10000100
245	00020	STMS (FTMF0+D#5):%,(PTMF0+D#5):%; Store in Ptmp1 11100000 01000000 11100100 00001110 11000000
246	0002D	LOAD (PTMP0+D#3):%,(PTMP0+D#3):%; Get P(3) for test point 11100000 01000000 11000100 00001100 11000001 10000100
247	0002E	STMS LSTF,LSTF 11100000 01000001 01000100 00010100 11000000

LINE	ADER	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 10 PAGE 11
249	0002F	; Start of sample interval 12
250	0002F	CALL •EVERY:X
200		00010000 00000101 01100110 11111111 11000000
251	00030	LOAD (PTMPO+D#4):Z;(PTMFO+D#4):Z; Get P(4) for test point
		11100000 01000000 11010100 00001101 11000001 10000100
252	00031	STMS LSTP,LSTP
		11100000 01000001 01000100 00010100 11000000
254	00032	; Start of sample interval 13, GET FEL
255	60032	CALL ,EVERY:%
		00010000 00000101 01100110 11111111 11000000
256	00033	CALL : QMFL:% ; Put the QMF filter
		00010000 00001010 01110110 11111111 11000000
257	00034	LOAD (PTMPO+D+5):X,(PTMPO+D+5):X; Get P(5) for test point
		11100000 01000000 11100100 00001110 11000001 10000100
258	00035	STMS LSTF, LSTP
		11100000 01000001 01000100 00010100 11000000
260	00036	Start of sample interval 14: set P(6)
261	00036	
		00010000 00000101 01100110 11111111 11000000
262	00037	LDAB F:F ; Input P(6)
		11100600 01000001 00100100 00010010 11000001 10000100
263	00038	STMS (PTMFG+D#6):X,(PTMFO+D#6):X
	Tall	11100000 01000000 11110100 00001111 11000000
264	00039	SIMS LSTP+LSTP
		11100000 61060061 61060100 00010100 11000000 00101010
266		f Start of sample interval 15, Get P(7)
267	0003A	CALL *EVERY\$% 00010000 00000101 01100110 11111111 11000000
268	0003E	LOAB P+P
200	00031	11190000 01000001 00100100 00010010 11000001 10000100
269	00030	STMS (PTMPO+D#7):%,(PTMPO+D#7):%; Store in Ptemp7
207	00000	11100000 01000001 00000100 00010000 11000000
270	0003B	
		11100000 01000001 01000100 00010100 11000000
272	0003E	; Start of sample interval 16
273		
		00010000 00000101 01100110 11111111 11000000
274	0003F	FILTER1 +CCNO+10(0):%
		11100000 01010000 00000000 00000000 11100000 01000100
275	00040	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
276	00041	STMS LSTP,LSTP ; Set test point predictor to zero
		11100000 01000001 01000100 00010100 11000000
277	00042	CALL ,CPYPRED: 7 Copy predictors from temporary buff
		00010000 00010110 00010110 111111111 11000000
278	00043	; to the active area
280		f Start of sample interval 17
281	00043	
1		00010000 00000101 01100110 11111111 11000000

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10 PAGE 12
283 284	00044 00044	<pre>\$ Start of samole interval 18; set FEH CALL ;EVERY:X</pre>
265		
207	00046	00010000 00000111 11100110 11111111 11000000
286	00046	그렇게 하면 사람이 되면 구성 경기에 가장하는 것이 되었다. 그런 사람들은 사람들이 되었다면 그렇게 되었다면 그렇게 되었다면 그렇게 되었다면 그렇게 되었다.
290	00047	; FOR r. := 0 TO 13 DO
291		The same and the s
292	00048	01000000 00000000 11010110 11111111 11000000
	60048	
		f Start of sample interval 20 + 10n
275	00045	DALL FEVERN: 2 00010000 00000101 01100110 11111111 11000000
297	00049	# Start of sample interval 21 ÷ 10n
298	00049	
		00010000 00000101 01100110 11111111 11000000
		* Start of sample interval 22 + 10m
301	00044	DALL FEVERY:% 00010000 00000101 01100110 11111111 11000000
303	0004B	; Start of sample interval 23 + 10m; input FEL
304	0004B	CALL *EVERY:% 00010000 00000101 01100110 11111111 11000000
305	00040	CALL -QMFL:X : Quadrature mirror filter low
		00010000 00001010 01110110 11111111 11000000
		Figure 1 Start of sample interval 24 + 10n
300	00040	CALL *EVERY:X 00010000 00000101 01100110 11111111 11000000
		i Start of sample interval 25 + 10n
311	0004E	CALL ;EVERY:% 00010000 00000101 01100110 11111111 11000000
		VOSTOVO VOVOTOT VIIVOTIV IIIIIII IIVOVOV VOVOVOV
313	0004F	
314	0004F	CALL FEVERY: X 00010600 00000101 01100110 11111111 11000000 000000
		00010000 00000101 01100110 11111111 11000000
	00050	
317	00050	CALL FEVERY:X
		00010000 00000101 01100110 11111111 11000000
319 320	00051 00051	; Start of sample interval 28 + 10m; Input FEH CALL , EVERY: %
020	00031	00010000 00000101 01100110 11111111 11000000
321	00052	CALL , QMFH: % i Input and filter FEH
		00010000 00000111 11100110 11111111 11000000

323	00053	: Start of sample interval 29 + 10n
324	00053	CALL FEVERY:X
tol		60610000 00000101 01100110 11111111 11000000
325	00054	REFLP ; ENDFOR
		10000000 01101111 11110110 11111111 11000000
326	00055	JMP ,SOFENT: % ; Do everything again for next fram
		00110000 00000000 10100110 11111111 11000000

ADDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 10 PAGE 13

LINE

```
LINE
      ADDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10
                                                     PAGE 14
328
    329
     00056 ##
               PROCEDURE operations_done_EVERY_sample_interval
     00056 ## { This procedure performs all the operations which are #
     00056 ## performed every sample interval. All the input ports are #
331
     00056 ** read immediatly after SCLCK soes low. The required input *
 332
     60056 ## data is transferred to working memory in the main procedure*
333
    00056 1# }
335
    00056 #
               Wait for rising edge of 5 enable complement.
    00056 #
               Output Si
336
    00056 i*
               Input FE,F;
337
    0005é ##
               Read test point switch and output appropriate data;
338
               Move data to the delay buffer;
339
    00056 3*
    00056 3*
               Low eass the reconstructed residual?
340
              Ferturb and high pass the reconstructed residual;
341
    00056 #
342
    00055 ix Sum the high and low pass residuals;
    00056 #
343
               Pass data through the predictor
    00056 ##
              Preemphasiz the datat
344
    00056 ##
               Shift the rins buffer:
345
               Store zero in the ring buffer base: (required for inter*
    00056 **
346
    00056 **
347
                                                         75
                                            Polation.
348 00056 11
               EMD:
349
    - OOOS- - 多本京宗朱永本本本等的主义大学李永宗朱永本本本本本文本文本文本文本文本文本文本文本文本文本文本文本文本文本文文文文文
     00056 EVERY: JMF .SEMBL: EVERY: % ; Wait for .5 Enable to be low
350
           00057 WAITLP: UMP .SEMBL/SUTLP: / Wait for .S Enable to be high
          00058
                 JMF .WAITLF:%
 352
           00059 # Now output S
 353
 354
     00009 DUTLP: LAMR 8
                               ; Load S into the Acc
          11160000 01010000 00000001 00111110 11000001 10000100
                 SUTFUI SFFT, SFRT + Output S
          356
     00056 f Now input from both input ports in case we need to read them
 357
                 INPUT GNINPUT, FEPRT, MULTI ; Get FEL on FEH (depending on same
          11100000 01010111 00000110 01100000 11100000 01000100
     00050
                 FILTERL
                               i And double it
          STLS *FE*FE ; Store it awas for Possible later use
          INPUT .PPRT.
                              ; Input a predictor
360
     0005E
          11100000 01010000 00000110 01000000 11000001 10000109
                 STME PIP
     0005F
                           f Store it away for possible later use
361
           362
     00060 ; Test the 'TEST POINT SELECT' switch and output the data to test port
                 JMP TSTFEL, NOTFEL: % : Test if FEL to be output
     00060
          LOAD LSTFEL, LSTFEL ; Yes - Load FEL into Acc
     00061
          11100000 0100000: 01010100 00030101 11000001 10000100
     00062 NOTFEL: JMP TSTFEH, NOTFEH: % ; Test if FER to be output
          00063
                 LOAD LSTFEH, LSTFEH ; Yes - Load FEH into Acc
```

11100000 01000001 01100100 00010110 11000001 10600100

LIN	E ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 10 PAGE 15
36	7 00064	NOTFEH: JMF ISTFE:NOTFE:Z ; Test if FE to be output
		00111010 00000110 01100110 11111111 11000000
36	8 00065	LOAD LSTFE:LSTFE ; Yes - Load FE into Acc
		11100000 01000001 01110100 00010111 11000001 10000100
36	7 00066	NOTFE: JMP TSTErNCTE:% ; Test if AEDD to be output
		00111011 00000110 10000116 11111111 11000006 00000000
37	0 00067	LDMR (LFFBS+LFFSZ+D\$1):% ; Yes - Load AEDD into Acc
		11100000 01010000 00000000 11100001 11000001 10000100
37	1 00058	NOTE: JMF TSTHPF.NOTHFF:X ; Test if HPF to be output
		00111100 00000110 10100110 11111111 11000000
37	2 00069	LDMR (HFFBS+HPFSZ-D41):% f Yes - output HPF
		11100000 01016000 00000001 01106110 11060001 10000106
37	3 0006A	NOTHEF: JMF TSTBUM:NOTBUM:: Test if resenerated residual to be outpu
		00111101 00000110 11000110 11111111 11000 000 00000000
37	4 0006B	
		11100000 01010000 00000000 11100010 11000001 10000100
37	5 00060	NOTBUM: JMP TSTSPE:NOTSPE:% ; Test if SPE to be output
		0-111110 00000110 11100110 11111111 11000000
37	6 0004B	LIMF (FREDBS+D#1):% 1 Yes - Load SFE into Acc
		11100000 01010000 00000001 00110100 11000001 10000100
37	7 0005E	NOTSFE: JMF TSTERWOTS:X
		66111111 60000111 00000110 11111111 11060000 00000000
37	8 0006F	LDAR (DENDSAD#2):% ; Yes - Load S into Acc
		11100000 010 0000 00000000 00111110 11000001 10000100
37	9 00070	NOTS: OUTFUT TEST TEST : Send the value to the analog test port
		11100000 01100010 00000119 00100000 11000000 00100011
38	0 00071	CALL #LPF:X i low pass the unperturbed samples
		00010000 00001111 01000110 11111111 11000000
35	1 00072	CALL *HPF:% High pass the perturbed samples
		00010000 00010001 11010110 11111111 11000000
38	2 00073	FilTER1 .COmi.(LFFPS+LPFS2):% ; ado the HPF and LPF signals
		11100000 01010000 00010000 11100000 11100000 01000100
38	3 00074	Filtenv (CON) (HEFBS+LPFSZ)1%
		11100000 01010200 00010001 01106011 11100010 11000100
38	4 00075	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
38	5 00076	STLR •SUMSA
		11100000 00111111 11110000 11100001 11000000
38	4 00077	i Now pass the data through the predictor
38		CALL *PRED*X
0.	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	00010000 00010100 10110110 11111111 11000000
38	85000 8	* Now deemphasize the data and shift ring buffer data
38		
		00010000 00010111 00010110 11111111 11000000
39	0 00079	# Now load zero into base of rins buffer
39		
0,,		11100000 01101111 11110110 111111111 1100100
39	2 0007A	FILTER1 ,CONC,10(0);%;
37.	VOV/H	11100000 01010000 00000000 00000000 11100000 01000100
39	3 0007B	FILTERL
37	0 000/15	11100000 01101111 11110110 111111111 11100000 11000000
39	4 0007C	STMR (PERTBS-1):%
01	VVV/6	11100000 00111111 11110001 01000000 11000000
39	5 00070	STMR CRIN, OMFBS ; Store zero and return
37	U VVV/D	10100000 00111111 11110000 00000000 11000000
		TOTOVOVO VOTITITI TITTOVOV VVVVVVVV TTOVOVOV VVIVIVIV

```
LINE
      ADDS SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10 PAGE 16
      397
398
      0007E ## PROCEPURE Quadrature Mirror Filter High
      0007E it( This procedure performs a quadrature mirror filter operation*
399
      0007E it assuming that FEH is in the FE input buffer. It place FEH at *
400
401
      0007E is the start of the OHF portion of the rins buffer.
      0007E ##
403
      0007E ## sum != FEHEn3 # CRMF[0]
      0007E ## FOR i := 1 TO 17 DO
404
      0007E ## sum := sum + FEHI: + 2*i] * CRHFE2*i];
405
406
      0007E ## sum := FEL[n] * QMF[0] - sum!
407
      0007E ## FOR i #= 0 TO 17 DO
      0007E ## sum := sum + FELEn + 2*13 * COMF[2*i];
408
409
     0007E ## LPFBSE00 := sum#
410
     0007E #8 END#
411
      412
      DOOTE QHEH!
                     LOAD
                             FEFFE
                                                  i Fetch FEL from the FE input b
            11100000 01000001 00110100 00010011 11000001 10000100
413
      0007F
                     5185
                             . GHFBS
                                                  ; Store at base of QMF rins buf
             11100006 00111111 11116000 00000000 11000000 00161010
414
      00050
                     STORE
                           ,LSTFEH,LSTFEH
                                                  ; Store for the test soint
            415
      00081
                     FILTER1 (COMF+D40):X (OMFF8+D40):X ; Do the FEH terms first
            11100000 01010010 00110000 00000000 11100000 01000100
      00082
416
                     FILTERN (COMF+D#2):%:(QMFBS+D#10):%
            11100000 01010010 01010000 00001010 11100010 11000100
417
      00083
                     FILTERN , (CQMF+D#4):2, (QMFBS+D#20):X
            11100000 01010010 01110000 00010100 11100010 11000100
418
      00084
                     FILTERN . (COMF+D+6):X. (CMFBS+D+30):X
            11100000 01010010 10010000 00011110 11100010 11000100
419
      00085
                     FILTERN (COMF+D+8):X, (QMFBS+D+40):X
            11150600 01010010 10110000 00101000 11100010 11000100
420
      00036
                     FILTERN . (CRMF+D+10):X: (RMFBS+D+50):X
            11100000 01010010 11010000 00110010 11100010 11000100
421
      00087
                     FILTERH , (CGMF+D+12): Z, (QMFBS+D+60): X
            11100000 01010010 11110000 00111100 11100010 11000100
422
     00088
                     FILTERN , (COMF+D$14):X, (QMFBS+D$70):X
            11100000 01010011 00010000 01000110 11100010 11000100
423
     00089
                     FILTERN , (CGMF+D+16):2, (QMFBS+D+80):2
            11100000 01010011 00110060 01010000 11100010 11000100
424
     00034
                     FILTERN .(COMF+D#18):Z.(GMFBS+D#90):Z
            11100000 01010011 01010000 01011010 11100010 11000100
425
     00083
                     FILTERN , (CQMF+D$20):2, (QMFBS+D$100):2
            11100000 01010011 01110000 01100100 11100010 11000100
     00080
426
                     FILTERN (COMF+D#22):X, (QMFBS+D#110):X
            11100000 01010011 10010000 01101110 11100010 11000100
     000BD
427
                     FILTERN . (COMF+D424):2, (QMFBS+D4120):X
            11100000 01010011 10110000 01111000 11100010 11000100
428
     000BE
                     11100000 01010011 11010000 10000010 11100010 11000100
429
     0008F
                     FILTERN , (CQMF+D+28): X, (QMFBS+D+140): X
            11100000 01010011 11110000 10001100 11100010 11000100
430
     00090
                     FILTERN , (CQMF+D#30): %, (QMFBS+D#150): %
            11100000 01010100 00010000 10010110 11100010 11000100
```

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLEF SOURCE FILE, SEPT 10 PAGE 17
431	00671	FILTERN ,(CGMF+D#32):X,(QMFBS+D#160):X
		11199506 01010100 00116000 10100000 11100010 11000100
432	00092	FILTERN .(COMF+D#34):X.(OMFES+D#170):X
		11100000 01010100 01010000 10101010 11100010 11000100
433	00093	FILTENNA (CGMF+D400):%,(QMFBS+D45):% ;sum:=FELEOJ+QMFEOJ-sum
		11100000 01010010 00110000 00000101 11100110 11000100
434	00094	FILTERN (COMF+D402):%,(QMF2S+D415):%
		11100000 01010010 01010000 00001111 11100010 11000100
435	00075	FILTERN (COMF+D464):%,(QMFBS+D425):%
		11100000 010100;0 01110000 00011001 11100010 11000100
436	00076	FILTERN (COMF+D406):X,(QMFBS+D435):X
		11100000 01010010 10010000 00100011 11100010 11000100
437	00077	FILTERN +(COMF+D#0S):X+(QMFBS+D#45):X
		11100000 01010010 10110000 00101101 11100010 11000100
438	00098	FILTERN *(CRMF+D410):2)(GMFBS+D455):2
		11100000 01010010 11010000 00110111 11100010 11000100
439	00099	FILTERN +(COMF+D412);X+(OMFBS+D465);X
		11100000 01010010 11110000 01000001 11100010 11000100
440	0009A	FILTERN (COMF+D414):X:(GMFB8+D475::X
		11100000 01010011 00010000 01001011 11100010 11000100
441	00077	FILTERN (COMF+D#16):%;(OMFBS+D#85):%
		11100000 010.0011 00110000 01010101 11100010 11000100
442	00090	FILTERN (COMF+D418):2,(QMFR8+D495):2
		11100000 01010011 01010000 01011111 11100010 11000100
445	00091	FILTERN (CRMF+D#20):X,(OMFBS+D#105):X
		11100000 01010011 01110000 01101001 11100010 11000100
444	0009E	FILTERN +(CRMF+D*22);%+(RMFBS+D*115);%
		11100000 01010011 10010000 01110011 11100010 11000100
445	0009F	FILTERN (CRMF+D*24):%,(RMFBS+11*)25):%
		11100000 01010011 10110000 01111101 11100010 11000100
446	000A0	FILTERN ,(COMF+D@26):%,(OMFRS+D#135):%
		111000000 01010011 11010000 10000111 11100010 11000106
447	00091	FILTERM (CRMF+D#28):2, (OMFSS+D#145):%
		11100000 01010011 11110000 10010001 11100010 11000100
445	06042	FILTERN ,(COMF+D#30):X,(QMFPS+D#155):X
		11100000 01010100 00010000 10011011 11100010 11000100
447	000A3	FILTERN (COMF+D#32):X,(OMFRS+D#165):X
		11100000 01010100 00110600 10100101 11100010 11000100
450	000R4	FILTERN ,(CRMF+D#34):X,(RMFBS+D#175):X
		11100600 01010100 01610000 10161111 11100010 11000100
451	00045	FILTERL
AFD		11100000 01101111 11110110 111111111 11100000 11000000
452	000A6	JMF
4==		00110906 00001101 00000110 11011111 11000000 000000
453	000A7	f and return

```
455
    00047 ## PROCEDURE Quadrature_Mirror_Filter_Low
450
457
    060A7 ix . This procedure does the quadrature mirror filtering*
458
    000A7 ix operation assuming FEL is in the FE input buffer. 3%
    000A7 ##
    000A7 ## sum := FEHEn+03 # COMFE13
460
     000A7 ## FOR i != 1 TG 17 DG
451
    000A7 ## sum := sum + FEHEn+2*i0 * COMFE2*i+1]#
462
    000A7 ## FDR i := 0 TO 17 DO
463
464
    000A7 ## sum := sum + FEHEn+2*i] * COMFE2*i+i];
465
    000A7 #x LPFB5 := sum!
     000A7 ##
465
     467
465
     000A7 QMFL:
                   LOAD
                            FE , FE
           11100000 01000001 00110100 00010011 11000001 10000100
                            ,OMFB0
469
     000AB
                    STAF
                                                 i Load FEL and store in Rins b
           000097
                    STORE
                            TLATFEL LATFEL
                                                i Store for the test point
           FILTER: (COMF+D41):X, (QMFBS+D45):X + Do the FEH terms first
471
     00044
           11100000 01010010 01000000 00000101 11100000 01000100
     GOOGR
                    FILTERN (COMF+D#3):X,(OMFBS+D#15):X
           11100000 01010010 01100000 00001111 11100010 11000100
                    FILTERN (COMF+D+5):X,(QMFBS+D+25):X
     00040
           11100600 81010010 10600000 80011001 11100010 11000100
                    FILTERN *(COMF+D*7):X*(OMFBS+D*35):X
474
     GACCO
           11100000 01010010 10100000 00100011 11100010 11000100
                    FILTERN (COMF+D49):X, (QMFBS+D445):X
     000AE
           11100000 01010010 11000000 00101101 11100010 11000100
                    FILTERN +(CONF+D#11)(%+(QMFBS+D#55))(%
476
     OGOAF
            11100000 01010010 11100000 00110111 11100010 11000100
                    FILTERN +(COMF+D#03):2+(OMF88+D#65):2
     00070
            11160000 01010011 00000000 016000001 11100010 11000100
478
                    FILTERN (COMF+D&15):X, (QMFBS+1)475):X
     000b1
           11100500 61615011 00100000 01001011 11100010 11006100
                    FILTERN +(COMF+D&17)1%+(QMFBS+D#85)1%
477
     000B2
            11100006 01010011 01000006 01010101 11100010 11000100
480
     00003
                    FILTERN (COMF+D+19):%, (GMFBS+D495):%
           11100000 01010011 01100000 01611111 11100010 11000106
                    FILTERN , (COMF+D#21):2, (QMFBS+D#105):2
481
     000B4
           11100000 01010011 10000000 01101001 11100010 11000100
                    FILTERN .(CRMF4D423):X.(RMFRS4D4115):X
482
     00095
           11100000 01010011 10100000 01110011 11100010 11000100
483
     00086
                    11100000 01010011 11000000 01111101 11100010 11000100
                    FILTERN , (COMF+D#27): %: (OMFBS+D#135): %
     00097
           11100000 01010011 11100000 10000111 11100010 11000100
                    FILTERN ,(CRMF+D#29):X,(QMFBS+D#145):X
485
     DOODE
           11100000 01010100 00000000 10010001 11100010 11000100
486
     000B9
                    FILTERN , (CQMF+D#31):X, (QMFBS+D#155):X
           11100000 01010100 00100000 10011011 11100010 11000100
487
     000BA
                    FILTERN , (CQMF+D433):2, (QMFBS+D4165):2
```

11100000 01010100 01000000 10100101 11100010 11000100

SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10 PAGE 18

ADDA

LINE

LINE	ADDR	SYNTHES	IS PROCES	SOR	ASSEN	LER	SOUR	CE	FILE.	SEPT	10	PA	GE	19
488	00085		FILTERN	,(COMF+I	‡ 35)	17: (OMF	BS+D4:	175)::				
		11100000	01010100	011	00000	1010	1111	11	10001	1100	00100			
487	00030		FILTERN									do i	FFI	terms
		11100000	01010010											
490	OCCEL		FILTERN											
	-	11100000	01010010								00100			
491	000BE		FILTERN								,,,,,,			
		11100000	01010010								00100			
492	GOORF		FILTERN											
7/-		11100000	01010010			-					00100			
493	00000	11100000	FILTERN								10100			
	00000	11100000	01010010					1111			10100			
454	00001	11100000	FILTERN								10100			
47.4	00001	******	01010010											
*05	AACCO	11100000									20100			
475	00002	*****	FILTERN											
407	*****	11100000	01010011								00100			
495	60003	******	FILTERN											
407	81004	11100000	01010011								20100			
477	00004		FILTERN											
	*****	11100000	01010011											
478	00005		FILTERN											
		11100000	01010011								2000			
499	00006		FILTERN											
		11100000	01010011	100	00000	0110	1110	11	100010	1100	0160			
300	00007		FILTERN	11	COMF+D	123)	:2, ((OMF	BS+D#:	20)17				
		11100000	91019013	101	00000	0111	1000	11	100010	1100	0100			
501	00008		FILTERN	*	COMF+D	125)	:2, ((QMF	BS+14:	30) 13				
		11100000	01010011	110	00000	1000	0010	11	100010	1100	0100			
502	00009		FILTERN	11	COMFHI	\$27)	:%: (2MF	BS+D#:	(40):7				
		11100000	01010011	111	00000	1000	1100	11	100010	1100	0100			
503	000CA		FILTERN	11	COMF+D	\$29)	1%, (QHF.	BS+D4	50) 12				
		11100000	01010100	000	00000	1001	0110	11	100010	1100	0100			
504	000CE		FILTERN	, (CONFID	#31)	:2,((GMF	BS+D+	60) 17				
		11100000	01010100	001	00000	1010	0000	11	100010	1100	0100			
505	00000		FILTERN	,(CQHF+D	133)	:2, ((RMF	BS+D41	70)12				
		11100000	01010100	010	00000	1010	1610	11	100010	1100	0100			
506	00000		FILTERN											
		11100000	01010100	011	00000	1011	0100	11	100010	1100	0100			
507	000CE		FILTERL											
		11100000	01101111		10110	1111	1111	11	100000	1100	0000			
508	COCCE		JMF		PERT 1%			200	rturb			atio	חס	
	176	00110000	00001101	San San								196		
509	00000							000	d ret	-	2.00			
510	00000						i re							
							, , ,	- 00						

....

```
LINE ADDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10 PAGE 20
513 000D0 #% PROCEDURE PERTurbate#
514
   00000 18
515 00000 fki This procedure parterbates the location of the FE 3%
   000D0 ## BEGIN
516
517
   00000 ## Multiple accumulator be CON4;
518 000D0 ## Generatate seed uniformly distributed in [-thresh, thresh]#
519 00000 ## IF magnitude(FE) > seed THEN store at PERTBS
522 600D0 ** END#
523 00000 1%
524 60000 fw i The Fe cample is perturbed with the following probability*
525 00000 ** density function.
526 00000 ##
527 00000 F% Probability of being
528 000D0 ## Femturated
529
    COODS #8
                        +0.5
530 000EC 1%
                       + 1 +
531 00000 18
532 000D0 #8
533 66600 11
534 000B0 1%
535 000D0 +> +
536
   600D0 ## ----
537 000D0 #* -THRESH
                                     THRESH
538 000D0 ##
                   Input sample value
539 000D0 ## END#
541 000D0 PERT:
542 00000 + How do a full precision multiple by CON2
         STLF (LEAST | Store least sis prod
    111000.0 00111111 11110011 11111111 11000000 00111001
             STMR ##097 ; store most sis product
        11100000 00011111 11110011 11111116 11000000 00101010
             FILTER: +80N2+MOST / Multiple most sid bs CON2
        11100000 01010110 11000011 11111110 11100000 01000100
              FILTERL
        STLE .MOST
        000D5 LDMR MOST # Shift left 14 places
        11100000 01010000 00000011 11111110 11000001 10000100
             FILTRAU (CONZILEAST | Add CON4 times less) sig product
        11100000 01010110 11000011 11111111 11000010 01000100
             FILTERL
        STLR *LEAST
        000D9 STMS LSTFE.LSTFE | Save for the test point
    553 000DA STMR , MOST ; Store at base of LPF
```

```
LINE
   ADDA SYNTHESIS FROCESSON ASSEMBLER SOURCE FILE, SEFT 10 PAGE 21
    OCCUP
554
              STHR
                    , DLYBS
         11100006 00111111 11110000 10111110 11000000 00101010
    300DC i Take the absolute value
    COARC
                  NEG: NEGATE: %
                              f Test last bus value was megative
         JMF FOSTV:Z
         000DE NEGATE: FILTER1 ,COMM1,MOST:X ; IF nesative them negate it
        FILTERL
        STLR MOST
        561 000E1 POSTV:
    000E1
              000E2
563
                              i to I-THRESH, THRESH)
564 000E2
             FILTERL
                               : To get random seed
        11100000 01101311 11110110 11111111 11100000 11000000
         STME TEMPETEME | leave result in LSB
        11100000 01000001 10000100 0001:000 11000000 00101010
        SFL /.(D40054DDEFB400N1):X.(D4256*SCRATHTEMP):X..MULT1
        11100000 01010000 00010100 00011000 11100000 01000100
557
    000E5
                               : load seed into accumulator
             FILTER +CONMITMOST
568
    DOOES
                              + Subtract absolute value of signal
         11100000 01010000 00100011 11111110 11100010 11000100
          FILTERL
        11100000 01101)1: 111101/0 111111/1 11100000 11000000
              STER MOST
                              I Put on the bus
        11100000 061.1111 11110011 11111110 11000000 00111001
    0:0E8
              JMP NEG-MGFERTIX # If magnitude of signal > 128 don'T
        FILTER: (CLFF+D415) 1% LEAST (scramble bits in LEAST
        11100000 01010001 00110011 11111111 11100000 01000100
              FILTERL
        STLR :LEAST : Save least sis word of scrambled 1
        LDMR LEAST
                              i Fut MSE of LEAST into sign bit
        11100000 01010000 00000011 11111111 11000001 10000100
              JMP NEG PERTPIX
        LOAD LSTFE-LSTFE
        11100000 01000001 01110100 00010111 11000001 10000100
              STMR CRIN; (PERTBS-D#1):%
   000EF
        579
    000F0 PERTP: LOAD LSTFE, LSTFE
        11100000 01000001 01110100 00010111 11000001 10000130
560
   000F1
         STMR CRTN, (PERTBS+D#1):X
        000F2 NOFERT: LOAD LSTFE, LSTFE
        11100006 01000001 01110100 00010111 11000001 10000100
582 000F3
        STMR CRTN, (PERTBS+0 :2
```

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10 PAGE 22
EO.	00000	1466644464474644746446446446464646464646
584 585		**************************************
		# PROCEDURE Low_Pass_Filter: *
536		This Procedure low pass filters the data in the low pass*
587	000F4	
566		1% sum := CLFFE03 x LFFB9E03;
589	000F4	나는 사람들은 사람들은 사람들은 사람들은 사람들은 사람들은 사람들은 사람들은
590	000F4	
591	000F4	# LPFRS[31] := sum; *
592	000F4	<pre>i* DLYBS[0] := DDFBS[0]; *</pre>
573	000F4	** END;
594	00074	**
595	000F4	***************************************
576	000F4	LPF: FILTER1 .(CLPF+D40):X:(LPFBS+D40):X
		11100000 01016000 01000000 11000601 11100000 01000100
597	000F5	FILTERN (CLPF+D41):X,(LPFBS+D41):X
		11100000 01010000 01010000 11000010 11100010 11000100
598	00076	FILTERN (CLPF+D#2):X,(LFFB5+D#2):X
		11100000 013000 01100000 11000011 11100010 11000100
599	000F7	FILTERN + (CLPF+D43)1X+(LPFRS+D43)1X
		11100000 01010000 01110000 11000100 11100010 11000100
600	000F8	FILTERN (CLPF+D#4):X;(LPFRS+D#4):X
		11100000 01010000 10000000 11000101 11100010 11000100
601	000F9	FILTERN +(CLPF+D#5):%+(LPFES+D#5):%
		11109000 01010000 10010000 11000110 11100010 11000100
602	600FA	FILTERN (CLPF+D46):X, (LPFBS+D46):X
		11100000 01010000 10100000 11000111 1110001(11000100
603	GOOFE	FOLTERN > (CLPF+D#7):%,(LPFBS+D#7):%
000		11100000 01010000 10110000 11001000 11100010 11000100
604	000F3	FILTERN (CLFF+D48):X,(LPFBS+D48):X
		11100000 01010000 11000000 11001001 11100010 11000100
605	000FH	FILTERH (CLFF+D49):%,(LPFBS+D49):%
		11160000 01010000 11010000 11061010 11100010 11000100
605	000FE	FILTERN + (CLFF+D\$10):2, (LFFBS+D\$10):2
		11100000 01016000 11106000 11001011 11100016 11060100
607	000FF	FILTERN (CLFF+D411):X,(LPFBS+D411):X
	200.	11100000 61010000 11110000 11001100 11100010 11000100
608	00100	FILTERN ,(CLPF+D#12);X,(LPFBS+D#12);X
0.0	00200	11100000 01010001 00000000 11001101 11100010 11000100
609	06101	FILTERN (CLFF+D413):2,(LFFBS+D413):2
00,	00101	11100000 01010001 00010000 11001110 11100010 11000100
610	00102	FilterN +(CLFF+D&i4):Z,(LPFB8+D&i4):Z
010	00101	11160600 01016001 00100000 11001111 11100010 11000100
611	00103	FILTERN + (CLPF+D#15):2, (LPFBS+D#15):2
	20100	11100000 01010001 00110000 11010003 11100010 11000100
612	00104	FILTERN + (CLFF+D#16):X, (LFFES+D#16):X
011	VV1V4	11100000 01010001 01000000 11010001 11100010 11000100
613	00105	FILTERN ,(CLPF+D&17):%,(LPFBS+D&17):%
010	00103	11100000 01010001 01010000 11010010 11100010 1100010C
614	00106	FILTERN ,(CLPF+D#18):%,(LPFBS+D#18):%
014	00100	11100000 01010001 01100000 11010011 11100010 11000100
115	00107	
615	00107	FILTERN ,(CLPF+D#19):Z,(LPFBS+D#19):Z 11100000 01010001 01110006 11010100 11100010 11000100
411	00+00	
616	00108	FILTERN ,(CLPF+D#20):%,(LPFBS+D#20):%
-		11100000 01010001 10000000 11010101 11100010 11000100

LINE

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10 PAGE 23
617	00109	FILTERN (CLFF+D#21):X,(LPFBS+D#21):X
		11100000 01010001 10010000 11010110 11100010 11000100
618	0010A	FILTERM +(CLPF+D#22):X+(LPFBS+D#22):X
		11100000 01010001 10100000 11010111 11100010 11000100
619	0010B	FILTERN (CLFF+N#23):%,(LFFBS+D#23):%
		11100000 01010001 10110000 11011000 11100010 11000100
620	00100	FILTERN (CLPF+D*24):X, (LPFBS+D*24):X
		11100000 01010001 11000000 11011001 11100010 11000100
621	0010D	FILTERN (CLPF+D#25):2,(LPFBS+D#25):X
		11100000 01010001 11010000 11011010 11100010 11000100
622	0010E	FILTERN (CLPF+D#26):X,(LPFBS+D#26):X
		11100000 01010001 11100000 11011011 11100010 11000100
623	0010F	FILTERN (CLPF+D#27):X,(LPFBS+D#27):X
		11100000 01010001 11110000 11011100 11100010 11000100
624	00110	FILTERN (CLPF+D428):X, (LPFBS+D428):X
		11100000 01010010 00000000 11011101 11100010 11000100
625	00111	FILTERN +(CLPF+D#29):X+(LPFB8+D#29):X
		11190000 01010010 00010000 11011110 11100010 11000100
626	00112	FILTERN (CLFF+D430):X:(LFFBS+D430):X
		11100000 01010010 00100000 11011111 11100016 11000100
627	00113	
		11100000 01101111 11110110 11111111 11100000 11000000
628	00114	STER (LEAST) Store least sis prod
		11100000 00111111 11110011 11111111 11000000
629	00115	57MK •MOST • store most sis product
		11100000 00111111 11110011 11111110 11000000
630	00116	FILTER: JONLPF:MOST ; Multiple most sis be GNLPF
		11100000 01010111 00100011 11111110 11100000 01000100
631	00117	FILTERL
/	*****	11100000 01101111 11110110 111111111 11100000 11000000
632	00118	STLR MOST
	****	11100000 00111111 11110011 11111110 11000000
633	00117	LDMR MOST : Shift left 16 places
	*****	11100000 01010000 00000011 111111110 11000001 10000100
634	0011A	FILTRAU , GRILFF, LEAST ; Add GNLFF times least sis product
175	0011R	11100000 01010111 00100011 11111111 11000010 01000100
635	UUIII	FILTERL 11100000 01101111 11110110 11111111 11100000 11000000
636	00110	STMR CRIN.(LPFRS+LPFSZ):% ; Store it away
030	00116	10100000 0011111; 11110000 11100000 11000000 00101010
		TATAAAAA AATITII TIITAAAA IITAAAAA IITAAAAA AATAIAIA

L	INE	ADDE	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 10 PAGE 24
	538	ASSIT	· ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
			# PROCEDURE Hish_Pass_Filter; *
	640		<pre>i#i This Procedure high pass filters the data in the high *</pre>
		0011D	<pre>ix rass Filter buffer.</pre>
	542	00110	## sum != CLPFE03 # HFFE8E03;
- 6	643	00111	## FOR i != 1 TO 30 PO #
6	544	00119	<pre># sum := sum + CLFF[i] * HFFRS[i]; *</pre>
	645	0011B	# HPFBS[31] := 5#HPFBS[15]-sum; *
4		0011B	# END;
			;*

		0011E	
	7	OULIE	HPF: FILTER: (CLPF+D+0):X,(HFFBS+D+0):X
			11100066 01010000 01000001 01000100 11100000 01000100
(650	0011E	FILTERN (CLPF+D+1):X,(HPFBS+D+1):X
			11160000 01010000 01010001 01000101 11100010 11000100
6	551	0011F	FILTERN +(CLFF4D&2):X+(HFFF8+N+2):X
1			11100000 01010000 01100001 01000110 11100010 11000100
é	552	00120	FILTERN +(CLPF+D\$3):2+(HPFB3+D\$3):2
			11100000 01010000 01110001 01000111 11100010 11000100
é	553	00121	FILTERM *(CLPF+D#4):X*(HPFBS+D#4):X
			11100000 01010000 10000001 01001000 11100010 11000100
4	554	00132	FILTERN (CLPF+D45)1% (HPFB8+D45/1%
			11300000 01010000 10010001 01001001 11100010 11000100
. ,	555	00123	
	223	00123	FILTERN , CLFF+D#6):X, (HPFRS+D#6):X
,		00151	11100000 01010000 10100001 01001010 11100010 11000100
	556	00124	FILTERN +(CLPF+D#7):X+(APFBS+D#7):X
,			11106000 01010000 10110001 01001011 11100010 11000100
_ 0	557	00125	FILTERN *(CLFF+D#S):X*(HPFBS+D#S):X
			11100000 01010000 11000001 01001100 11100010 11000100
	358	00126	FILTERN (CLPF+D&F):Z,(HPFBS+D&F):X
			11100000 01010000 11010001 01001101 11100010 11000100
ć	59	00.27	FILTERN (CLEF+D#10):X. (HPFBS+D#10):X
			11100000 01010006 11100001 01001110 11100610 11000100
6	660	00128	FILTERN *(CLPF+D*11):X*(HPFBS+D*11):X
			11100000 01010000 11110001 01001111 11100010 11000100
	61	00129	FILTERN (CLFF+D#12):Z,(HPFB3+D#12):Z
			11100000 01010001 00000001 01010000 11100010 11000100
6	662	0012A	FILTERN (CLFF+D413):Z,(HPFBS+D413):Z
			11100000 01010001 00010001 01010001 11100010 11000100
6	63	0012B	FILTERN ,(CLPF+D\$14);X,(HPFRS+D\$14);X
			11100000 01010001 00100001 01010010 11100010 11000100
1	64	00120	FILTERN +(CLFF+D#15);Z+(HPFBS+D#15);Z
	727	00121	
	/e	****	11100000 01010001 00110001 01010011 11100016 11000100
0	65	0012B	FILTERN +(CLPF+D#16):X+(HPFRS+D#16):X
,		22455	11100000 01010001 01000001 01010100 11100010 11000100
. 6	666	0012E	FILTERN (CLFF+D#17):X,(HPFBS+D#17):Z
. ,			11100000 01010001 01010001 01010101 11100010 11000100
. 6	67	0012F	FILTERN (CLPF+D#18):X,(HPFRS+D#18):X
			11100000 01010001 01100001 01010110 11100010 11000100
ó	866	00130	FILTERN (CLFF+D#19):Z,(HPFBS+D#19):Z
1			11100000 01010001 01110001 01010111 11100010 11000100
. 6	65	00131	FILTERN (CLPF+D#20):X,(HPFBS+D#20):X
1			11100000 01010001 10000001 01011000 11100010 11000100

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 10 PAGE 25
670	00132	FILTERN +(CLPF+D#21):X+(HFFBS+D#21):X
		11100000 01010001 10010001 01011001 11100010 11000100
671	60133	FILTERN (CLPF+D422):X, (HPFBS+D422):X
		11100000 01010001 10100001 01011010 11100010 11000100
672	00134	FILTERN (CLFF+D423):%,(HFFBS+D423):%
		11100000 01010001 10110001 01011011 11100010 11000100
673	00135	FILTERN ,(CLPF+D#24):X,(HPFBS+D#24):X
		11100000 01010001 11000001 01011100 11100010 11000100
674	00136	FILTERN (CLPF+D425):X,(HPFBS+D425):X
		11100000 01010001 11010001 01011101 11100010 11000100
675	00137	FILTERN (CLPF+D#26):X,(HPFBS+D426):X
		11106000 01010001 11100001 01011110 11100010 11000100
676	00138	FILTERN ,(CLPF+D427);X,(HFFBS+D427);X
		13100000 01010001 11110001 01011111 11100010 11000100
677	- 00137	FILTERN ,(CLPF+D#28);%,(HPFBS+D#28);%
		11100000 01010010 00000001 01100000 11100010 11600100
678	00134	
		11100000 01010010 00010001 01100001 11100010 11000100
679	00138	FilTERN +(CLPF+D#30):X+(HPFRS+D#30):X
		11100000 01010010 00100001 01100010 11100010 11000100
650	00130	FILTERI
		11190630 01101111 11110110 11111111 11100000 11000000
681	0013D	STLR : LEAST : Store least sis prod
		11100000 00111111 1111001: 11111111 11000000 00111001
682	0013E	STMR *MOST ; store most sis product
		11100000 00111111 11110011 11111110 11000000
683	0013F	FILTER1 (GNLPF, MOST ; Multiple most sis be GNLPF
		11166600 01010111 00100011 11111110 11100000 01000100
684	90140	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
685	00141	STLR MOST
		11100000 00111111 11110011 11111110 11000000
686	90142	IDMR MOST ; Shift left 16 places
		11100000 01010000 00000011 111111110 11000001 10000100
697	00143	FILTRAU ,GNLPF, LEAST ; Add GNLPF times least sis product
		11100000 01010111 00100011 11111111 11000010 01000100
688	00144	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
689	00145	STMR (HPFBS+LPFSZ):% : Store it away
		11100000 00111111 11110001 01100011 11000000
690	00146	FILTER: (CON), (HPFBS+LPFSZ):2 ; subtract 5*middle element
		11100000 01010000 00010001 01100011 11100000 01000100
691	00147	i to set complement HPF from Li
672	00147	FILTRNM ,CON4,(HPFBS+D415):%
		11100000 01010110 11010001 01010011 11100110 11000100
693	00148	FILTERN (CON1, (HPFBS+D+15):X
		11100000 01010000 00010001 01010011 11100010 11000100
674	00149	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
695	0014A	STLR CRTN, (HPFBS+LPFSZ):X ; Store that away
		10100000 00111111 11110001 01100011 11000000

	LINE	ADDR	BYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 10 PAGE 26
	697	G014E	\$842224-42224444444444444444444444444444
	698		
	699		() { This procedure predictor filters data in the
1	700	00148	
	701	00145	*
	702		**************************************
	703	00149	PRED: FILTER: SCRAT, (PO+D#O): %, (PREDBS+D#1): %
			11100000 01000000 00000001 00110100 11100000 01000100
*	704	00140	FILTERM SCRAT, (PO+D#1):X, (PREDBS+D#2):X
			11100000 01000000 00010001 00110101 11100010 11000100
	705	0014D	FILTERN SCRAT, (PO+D#2):X, (FREDBS+D#3):X
			11100000 01000000 00100001 00110110 11100010 11000100
	706	0014E	FILTERN SCRAT: (PO+D#3)1%, (FREDBS+D#4)1%
			11100000 01000000 00110001 00110111 11100010 11000100
	707	0014F	FILTERN SCRAT: (PO+D#4):X: (PREDBS+D#5):X
			11100000 01000000 01000001 00111000 11100010 11000100
	708	00150	FILTERN SCRAT, (PO+D#5):X, (PREDBS+D#6):X
			11100000 01000006 01010001 00111001 11100010 11000100
	709	00151	FILTERN SCRAT, (PO+D#6):X, (PREDBS+D#7):X
			1:100000 01000000 01100001 00111010 11100010 11000100
	710	00152	FILTERM SCRAT, (PG+D#7):%, (FREDBS+D#8):%
			11190000 01000000 01110001 00111011 11100010 11000100
	711	09153	FILTRNN COEFB.PREDBCL.PREDBS
			11100000 01010110 10110001 00110011 11100116 11000106
	712	00154	FILTER.
			11100000 01101111 11110110 11111111 11100000 11000000
	713	00155	# Do a double precion builtiple by 8
	714	00155	STLR *LEAST
-			11100000 00111111 11110011 11111111 11000000
	715	00156	STMR #MOST store most sid product
			11100000 00111111 11110011 11111110 11000000
	716	00157	FILTER: +COM3+hO37 ; Multiply most sis by 3
			11100000 01010110 11100011 11111110 11100000 01000100
	717	00158	FILTERL
			11100000 01101111 11110110 11111111 11100000 11000000
	716	00154	STLR MOST
			11100000 00111111 11110011 11111110 11000000
	719	0015A	LDMR MOST i Shift left 16 places
			11100000 01010000 00000011 11111110 11000001 10000100
	720	0015E	FILTRAU (CONS)LEAST Add 8 times least sis product
			11100000 01010110 11100011 11111111 11000010 01000100
	721	00150	FILTERL
			11100000 01:01:11 11110110 111:1111 11100000 11000000
	722	00151	STMR ,PREDBS
Start.			11100000 00111111 11110001 00110011 11000000
17.00	723	0015E	FILTER: +GMPRED+PREDBS + Scale so no overflow in deemphasis
			11100000 01010111 01010001 00110011 11100000 01000100
7 ., 2	724	0015F	FILTERL
			11100000 01101111 11110110 11111111 11100000 11000000
	725	00160	
			10100000 00111111 11110001 00111100 11000000
0.00			

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 10 PAGE
727	00161	;*************************************
728	00161	** PROCEDURE CoPY_PREDictor_coefficients *
729	00161	<pre>##{ This procedure copies the predictor coefficients from *</pre>
730	00161	** the temporary predictor buffer to the active predictor *
731	00161	# buffer. }*
732	00161	**
733	00161	;************************************
734	00161	CPYPRED: LOAD (PTMPO):X,(PTMPO):X
		11100000 01000000 10010100 00001001 11000001 10000100
735	00162	STORE ,(PO):%,(PO):%
		11100000 01000000 00000100 00000000 11000000
736	00163	LOAD (PTMPO+D#1):X+(PTMPO+D#1):X
		11100000 01000000 10100100 00001010 11000001 10000100
737	00164	STORE +8(F0+D+5):X+(F0+D+1):X
		11100000 01000000 00010100 00000001 11000000
738	00165	LOAD B(FTMP0+D#2):%;(PTMP0+D#2):%
		11100000 01000000 10110100 00001011 11000001 10000100
739	00165	STORE +8(P0+D42):%+(P0+D42):%
		11100000 01000000 00100100 00000010 11000000
740	00167	LOAD 8(FTMP0+D*3):X:(P0+D*3):X
		11100000 01000000 11000100 00000011 11000001 10000100
741	(.0168	STORE +8(P0+D\$3):% (P0+D\$3):%
		11100000 01000000 00110100 00000011 11000000
742	00169	LGAE B(PTMPO+D*4):X*(PTMPO+D*4):X
		11100000 01000000 11010100 00001101 11000001 10000100
743	001ch	STORE :8(P0+D\$4):%;(P0+D\$4):%
711	004/0	11100000 01000000 01000100 00000100 11000000
744	00168	LOAD 8(PTMPO+D#5):X,(PTMPO+D#5):X
745	001/0	11100000 01000000 11100100 00001110 11000001 16000100
745	00160	STORE :8(P0+D+5):%,(P0+D+5):% 11100000 01000000 01010100 00000101 11000000
747	00475	
746	0016E	LOAD 8(F7MF0+D#6):%,(PTMF0+D#6):% 11100000 01000000 11110100 00001111 11000001 10000100
747	0016E	STORE *B(P0+D46):X* (P0+D46);X
/4/	00105	
748	00115	11100000 01000000 01100100 00000110 11000000
/46	0016F	11100000 01000001 00000100 00010000 11000001 10000100
740	00170	STORE CRIN,8(P0+D#7);%,(P0+D#7);%
749	00170	5:URE UNINESTPUTE #//: 4/ (TUTE #//: 4/)

TINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 10 PAGE 28
751	00171	;*************************************
752	00171	* PROCEDURE DEEMPhasize; *
753	00171	** { This procedure deemphasizes data in the deemphasis *
754	00171	## filter memors.
755	00171	1*
756	00171	**************************************
757	00171	DEEM: LDMR DEMBS
		11100000 01010000 00000001 00111100 11000001 10000100
758	00172	FILTERA ,A, (DEMBS+D#1):X
		11100000 01010000 00116001 00111101 11100010 01000100
759	00173	FILTERN ,A, (DEMBS+D#1):X
		11100000 01010000 00110001 00111101 11100010 11000100
760	00174	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
761	00175	SIMR DEMSS
		11100000 00111111 11110001 00111100 11000000
762	00176	FILTER: .GNOUT.DEMBS ; Scale output to prevent overflow
		11100000 01010111 01100001 00111100 11100000 01000100
763	00177	FILTERE
		11100000 01101111 11110110 11111111 11100000 11000000
764	00178	STMS CRTN*(DEMSS+D*1):X
		10100000 00111111 11110001 00111101 11000000
765	00179	END END

TOTAL ASSEMBLY ERRORS =

CROSS REFERENCE TABLE

(DEEL	TYPE	VALUE	REF	ERENCE	s ·					
	Ĥ	4	00003	-106	107	758	759				
	ABSZ	4	00001	-65							
	ACC	A	00200	0							
	AUT01	D		0							
	AUTOL	D		0							
	AUTON	D		0							
	CALL	D		190	193	196	199	200	203	208	213
				218	223	224	229	236	243	250	255
				256	261	267	273	277	281	284	285
				238	295	298	301	304	305	308	311
				314	317	320	321	324	380	381	387
				389							
	CDDF	A	00047	-109	110						
	CHFF	A	00044	-110	111						
	CJF	Ė	00003	0							
	CJPP	A	00003	0							
	CJ8	4	00001	0							
	CFKE	A	00000	0							
	CLKXY	A	00040	0							
	CLFF	A	00004	-107	108	124	572	596	597	598	599
				600	601	602	603	604	605	606	607
				608	609	610	611	612	613	614	615
				616	617	618	619	620	621	622	623
				624	625	626	649	650	651	652	653
1				654	655	654	657	658	659	660	661
				662	663	664	665	666	667	668	669
				670	671	672	673	674	675	676	677
				678	679						
	COEFB	A	00005	167	566	711					
	CONO	A	00000	-103	167	274	392				
	COM1	Ř	00001	-104	382	383	566	690	693		
	CON128	A	0006F	-116	117						
	CONTRATA	A	0005B	-112							
	CON2	A	36000	-113	114	545	549				
	CGN4	A	0006B	-114	115	692					
	COMB	A	0006E	-115	116	716	720				
	CONM1	A	00002	-105	558	568					
	CONT	A	0000E	0							
	CPYPRED	A	00161	277	-734						
	COMF	A	00023	-108	109	415	416	417	418	419	420
				421	422	423	424	425	426	427	428
				429	430	431	432	433	434	435	436
				437	438	439	440	441	442	443	444
				445	446	447	448	449	450	471	472
				473	474	475	476	477	478	479	480
	E THE			481	482	483	484	485	486	487	488
- 1				489	490	491	492	493	494	495	496
				497	498	499	500	501	502	503	504
				505	506						
200											

CROSSEN	A	00008	0							
CRTN	A	6000A	395	578	580	582	636	695	725	749
			764							
DDFSZ	A	00003	-64	110						
VEEH	A	00171	389	-757						
DEMES	£	00130	-78	79	82	378	725	757	758	759
			761	762	764					
DEMBI	A	00002	-58							
DIR	A	00010	0							
DLY	A	00052	-59	76						
DLYBS	A	OOOBE	-73	74	554					
EVERY	A	00056	190	193	196	199	203	208	213	218
			223	229	236	243	250	255	261	267
			273	281	284	288	295	298	301	304
			308	311	314	317	320	324	-350	350
FALSE	A	00001	0							
FE	A	00013	-93	94	359	359	412	412	468	468
FEPRT	A	06003	-132	357						
FILTERI	D		167	274	382	392	415	471	545	558
			562	572	596	630	649	683	690	703
			716	723	762					
FILTERA	Ī:		758							
FILTERL	I		168	275	358	384	393	451	507	546
			550	559	564	569	573	627	631	635
			690	684	688	694	712	717	721	724
			760	763						
FILTERN	D		383	416	417	418	419	420	421	422
TILILING	**		423	424	425	426	427	428	429	430
			431	432	434	435	436	437	438	439
			440	441	442	443	444	445	446	447
			448	449	450	472	473	474	475	476
			477	478	479	480	481	482	483	484
			485	436	487	488	489	490	491	492
			493	494	455	496	497	498	499	500
			501	502	503	504	505	506	569	597
			598	599	600	601	602	603	604	605
			606	607	808	609	610	611	612	613
			614	615	616	617	618	619	620	621
			622	623	624	625	626	650	651	652
			653	654	655	656	657	658	659	660
			661	662	663	664	665	666	667	668
			669	670	671	672	673	674	675	676
			677	678	679	693	704	705	706	707
			708	709	710	759				
FILTRAM	p		0	,,,						
FILTRAU	I.		549	634	687	720				
FILTRN	D		433	692	711	, 20				
GNDDF	A	00073	-120	121	,,,,					
GNHPF	A	00073	-121	122						
GNINFUT	A	00074	-117	118	357					
			-117	120	630	634	683	687		
GNLPF	A	00072		762	030	034	000	00/		
GNOUT	A	00076	-123	123	723					
GNPRED	A	0 0075	-122	123	123					

	GNOHE	A	00071	-118	119						
	HEROUT	A	00153	-81							
	HPF	A	0011B	381	-649						
	HPFBS	É	00144	-80	81	372	383	649	650	651	652
				653	654	655	656	657	658	659	660
				661	662	663	664	665	666	667	668
				669	670	671	672	673	674	675	676
	UNEST		00001	677	678	679	689	690	692	693	695
	HPF8Z	A	00021	-66 357	111 360	372					
	INPUT IO	A	30000	0	267						
	10-0	A	00000	0							
		A	00009	-54	91	92					
	JMF	D	VVVV.	176	189	326	350	351	352	363	365
	Din			367	369	371	373	375	377	452	508
				556	557	571	576				
	JRF		00007	0							
	JSRP	Ė	00005	0							
	JZ	fi	0.000	0							
	LECT	\$ 1	00.00	0							
	LDMR	1		354	370	372	374	376	378	548	575
				633	685	719	75?				
	LEAST	k	003FF	-63	543	549	551	562	572	574	575
				626	634	681	687	714	720		
	LOAD	P		204	209	214	219	225	230	232	237
				239	244	246	251	257	262	268	364
				366	368	412	968	577	579	581	734
				736	738	740	742	744	746	748	
	LOGF	P	****	0							
	PF .	A	0668D 066F4	380	-596						
	LPFB8	Ŕ	00001	-74	75	370	382	596	597	598	599
	T: 1 P.P.		V. V.	600	601	602	603	604	605	606	607
				608	609	610	611	612	613	614	615
				616	617	618	619	620	621	622	623
				624		626	636				
	LPF82	P.	0001F	-56	75	81	108	370	382	383	636
				689	690	695					
	LSTFE	A	00017	-97		368	368	552	552	577	577
				579	579	581	581				
	LSTFEH	Ä	00016	-96		366	366	414	414		
	LSTFEL	A	00015		96	364	364	470	470		
	LSTF	A	00014	-94		220	220	226	226	233	233
				240		247		252	252	258	258
				264		270	270	276	276		
	MEMEN	A	00004	0		FAF	EAT	EAD	F57	558	560
	MOST	A	003FE	-85		545	547 630	548	553 633	682	683
				568 685		629 715		632 718		002	903
	NEG	A	00002	556		576	/10	/10	/17		
	NEGATE	A	00002 000DE	556		370					
	NEWAUTO	A	01000	0							
7	NOMEN	A	006FF	0							
	RUHLH		40011	v							

NOF	ERT	A	000F2	571	-581						
NOR	MAL.	À	06000	0							
TEM	=	Ĥ	00068	369	-371						
TON	FE	A	00065	367	-369						
YOTH	FEH	É	00054	365	-367						
NOT	FEL	À	90062	363	-365						
NOT	HFF	A	0006A	371	-373						
NOT	3	A	00070	377	-379						
NOT	SPE	Á	0006E	375	-377						
NOT	SUM	A	38000	373	-375						
OUT	F	A	00059	351	-354						
OUT	TUF	I		169	170	355	379				
P		Ė.	00012	-92	93	204	204	209	209	214	214
				230	230	237	237	244	244	262	262
				269	269	361	361				
P0		4	00000	-90	91	703	704	705	706	707	708
				709	710	735	735	737	737	739	739
				740	741	741	743	743	745	745	747
				747	749	749					
PAL		Ė.	00003	0							
PER		A	00000	452	508	-541					
PER		A	00141	-79	80	304	578	580	582		
PER	F	A	000F0	576	-579						
P05		4	00051	557	-561						
PPR		Ĥ	00002	-131	350						
PRE		f.	00147	383	-703						
PRE	DBS	A	00133	-76	77	78	376	703	704	705	706
				707	708	709	710	711	722	723	
	DOUT	f.	00130	-77							
PRE		A	00085	-111	112	113	711				
PRE		A	00009	-57	77	78					
PREL		A	00100	0							
PEH		Ė	0000	6							
FTMF	0.0	4	00000	-71		205	205	210	210	215	215
				219	219	225	225	231	231	232	232
				238	238	239	239	245	245	246	246
				251	251	257	257	263	263	269	269
				734	734	736	736	738	738	740	742
				742	744	744	746	746	748	748	
PUSH		I.		171	291						
QMF	35	A	00000	-72		395	413	415	416	417	418
				419		421	422	423	424	425	426
				427	428	429		431	432	433	434
				435	43ć	437	436	439	440	441	442
				443	444	445	446	447	448	449	450
				469	471	472	473	474	475	476	477
				478	479	480	481	482	483	434	485
				486	487	488	489	490	491	492	493
				494	495	496	497	498	499	500	501
				502	503	504	505	506			
QMF		A	0007E	224	285	321	-412				
GMFL		A	000A7	200	256	305	-468				
QMF	67	Ŕ	00024	-55	73	109					
2											

.

	REFLE	0		175	325						
	RET	I.		0							
	RFCT	A	00008	0							
	RINGE	6	00000	0							
	APCT	A	00009	0							
	S	A	0013E	-82	354						
	SCRAT	Á	00004	566	703	704	705	706	707	708	709
				710							
	SHETCLK	A	00800	0							
	SHFTLFT1	D		0							
	SHFTLFT2	D		0							
	SOFEKT	í	0000A		-189	189	326				
	SFL	D		172	173	391	566				
	SPRT	Ĥ	00002		169	169	355	355			
	STLR	I		385		547	551	560	570	574	628
				632	681	685	695	714	718		
	STLS	i		359							
	STMR	D		174	394	395	413	469	544	553	554
				578	580	582	629	636	682	689	715
				722	725	761	764				
	STME	Ŀ.		205	210	215	220	226	231	233	238
				240		247	252	258	263	264	269
				270	276	361	552	565			
	STORE	I		414	470	735	737	739	741	743	745
				747	749						
	SUP	A	00400	(7,		705				
	SUMBS	Ĥ	000E1	-75	76	374	385				
	TE	A	02000	0		F/F	-//				
	TEMP	A	00018	-98	565	565		770			
	TEST	A	00001	-136	170	170	379	379			
	THRESH	Ĥ	00005	-124	562						
	TRUE		00000	0							
	TSL	A	00001	0							
	TSK	A	00002	0	369						
	TSTE	A	0000E	-140	367						
	TSTFEH TSTFEH	A	0000A 00007	-139 -138	365						
	TSTFEL		00003	-137							
	TSTHFF	4	20000	-141	371						
	TSTS	A	0000F	-144	377						
	TSTSPE	A	0000E	-143	375						
	TSTSUM	A	0000D	-142	373						
	TWB	A	0000F	0							
	WAITLE	A	00057	-351	352						
	WRITE	A	00020	0							
	.Lb	A	00184	0							
	. MARK	A	01000	0							
	.HULT1	A	02044	357	566						
14	MULTA	A	02244	0							
	·MULTAM	A	02644	0							
	.MULTAU	A	00244	0							
	.MULTL	A	02000	0							
	.HULTN	A	02204	0							

. KUI. TRA	6	02604	0		
.NOF	Ĥ	60000	0		
.SENEL	À	00005	-134	350	351
SHIFT	A	00300	172	391	
.50F	A	00007	189		
.STL	À	00039	0		
.STK	A	0002A	0		
.STR	A	00023	0		
.TESTO	A	00008	137		
.TEST1	A	00009	138		
.TEST2	A	0000A	139		
.TEST3	A	000038	140		
.TEST4	A.	30000	141		
TEST5	A	00000	143		
.TEST6	4	0000E	143		
.TEST7	A	0000F	144		

5 ...

....

A2. RESIDUAL REGENERATION THROUGH INTERPOLATION OF LOW PASS COMPONENT BY ZERO AMPLITUDE INSERTION, AND SAMPLE POSITION PERTURBATION

1	00000	TITLE SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 13,1982
2	00000	LIST X
3	00000	**************************************
4	00000	PROJECT: 472A, RELP CODEC *
5	00000	** BOARD: SYNTHESIS PROCESSOR *
6	00000	# DEVICE: 6 MICROCODE PROMS, 3636 or 3628 *
7	00000	* LOCATION: 634,P1,P15,P29,P43,P57 *
8	00000	** PROG.DEV4: 60-0690 - 60-0695 *
9	00000	* ASSEMBLER: MEYA, April 82 *
10	00000	* FILE: SYNTHESIS.ASM
11	00000	** REVISION: SEPT 13-1982 *
12	00000	** UPDATE TABLE: *
13	00000	# OCT 4: HPF and LPF removed. #
14	00000	SEPT 9: Initial folding exp. hopeful *
15	00000	# Hissins method implemented. *
16	00000	* AUG 3: Aliasing high free regeneration *
17	00000	## JULY 7: Each filter assigned gain in PROM *
18	00000	* JULY 2: More precision maintained. *
19	00000	# JUNE 28: Filter Bains Adjusted. *
20	00000	## JUNE 14: Mods made to use subtract inst corr #
21	00000	# APRIL 23 - MAY 23: Initial prosem entry JM #
22	00000	*
23	00000	\$25xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

	00000			**************************************
	00000	**	11	BLE OF CONTENTS
	00000	**		
	00000	The same	PAGE	DESCRIPTION
29	00000	非		
	00000	**	1	File Header and Update Table
31	00000	; *	2	Table of Contents
32	00000	*	3	Filter size specifications
33	00000	**	4	Rins Buffer memors map
34	00000	;×	5	Scratch Pad memory map
35	00000	**	6	Coefficient Prom memory map
36	00000	*	7	Input/Output definitions
37	00000	**	8	PROCEDURE start_up;
38	00000	*	9	FROCEDURE mainf
39	00000	98	14	PROCEDURE operations_done_EVERY_sample
40 -	00000	7.8	17	PROCEDURE Quadrature_Mirror_Filter_Hish
41	00000	;*	20	PROCEBURE Quadratura_Mirror_Filter_Low
42	00000	**	23	PROCEDURE Low_Pass_Filter
43	00000	**	25	PROCEDURE Double_Difference_Filter
44	00000	**	26	PROCEDURE High_Pass_Filter
45	00000	**	28	PROCEDURE Predict
	00000	11	29	PROCEDURE Coby_PREDictor_Coefficients
47	00000	j.k	30	PROCEDURE DE-EMPhasize
	00000	**	31	Cross Reference Table
	00000			*************************************

```
00000 ##
52
                 FILTER SIZE SPECIFICATIONS
53
    00000 ##
    54
    00000 J: EQU
                   9
                       Fredictor filter order.
    00000 QMFSZ: EQU 36
                         i Quadrature Mirror Filter Size
    00000 LPFSZ: EQU 31
57
                         f Low pass filter size
58
   00000 PREDSZ: EQU 9
                         # Predictor Filter Size
    00000 DEMSZ: ERU 2
                        # Deemphasis Filter Size
   00000 DLY: EQU 100 ; Delay so predictors line up with frame
60
    00000 | The following constants define filter sizes in the rectification
61
   00000 | RELP implementation. They are used here only to specify the location
62
63
   00000 ; the filters in the coefficient PROM, which is the same for both
    00000 ; the perturbation and rectication implementations.
64
65
    00000 DDFSZ: EQU
                   3
   00000 ABSZ: EQU
66
                   1
   00000 HFFSZ: EQU
67
                   33
```

ADDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE

PAGE 4

ADDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13

LINE	ADDA	SYNTHE	SIS F	PROCESSOR	ASSEMBLER	SOUR	CE FILE	. SEFT	13	PAGE	5
88	00000	*******	EXXXX	*****	*****	****	*****	*****	****	*****	
89	00000	11 S	CRAT	OH PAL HEM	IOR'S MAF					1	
90	00000	******	***	******	****	***	*****	*****	***	*****	
91	00000	P01	EQL	8(0):%		;	base o	f pred	ictor	storage	
92	00000	PTMF0:	EQU	8(F0+J)	: *	;	Base of	f temp	erare	predictor	storage
93	00000	P:	EQU	BIFTMPO	+J) 12	;	Last P	redict	0: 00	ef, input	
94	00000	FE:	EQU	B(P+1):	%	;	FE(H.L) stora	920		
95	00000	LSTP:	EQU	8(FE+1)	:7	;	Last F	used			
96	00000	LSTFEL!	EQU	B(LETP+	1)17	;	Last Fi	EL imp	ut		
97	00000	LSTFEH:	EQU	8(LSTFE	L+1):%	;	Last F	EH inp	ut		
98	00000	LSTFE:	EQU	8(LSTFE	H+1 +1%	;	Last F	E calc	ulate	d	
99	00000	TEMF:	EQU	8(LSTFE	+1)1%	+	Tempor	ary st	ezsno	location	

```
ADDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 13
LINE
                                                        PAGE
      101
 102
     00000 1#
                 COEFFICIENT PROM MEMORY MAP
103
     00000 COM0: EQU 8(0):2 | constant zero
104
105
     00000 CON1: EGU 8(1):X
                                       ; constant one
     00000 COMM1: EQU 8(2):% | constant minus one
00000 A: EQU 8(3):% | Fre-emphasis constant/2
106
107
    00000 A:
108 00000 CLPF: EQU B(A+1):X
                                      ; Base of low pass filter coef
    00000 CDMF: EQU 8(CLFF+LPFSZ):% ; Base of quadrature mirror filter 00000 CDDF: EQU 8(CQMF+QMFSZ):% ; Base of double difference filter
109
110
111 00000 CHPF: EQU 8(CDDF+DDFSZ):% ; Base of high pass filter
112 00000 PREDSCL:EQU 8(CHPF+HPFSZ):%
                                        ; Scale factor for predictor
113
     00000 CONTORTR: EQU 8(PREDSCL):%
                                        # Scale factor is 1/4
114
     00000 CCN2: E0U 8:PRE0SCL41):%
115 00000 CDN4: EQU 8/(DN2+1):%
     00000 CONR: EGU F(CON4+1)1%
116
     00000 CON128: EQU 8(CON6+1):%
117
    00000 ENINFUT: E00 8 CON: 2841:1%
118
119
     00000 GNOWER EGU 8(GNINFUT41):2
                                        Filter sains follow
120 00000 SWLFF: ERU 8(GNGMF+1):7
121
     00000 BNDEF: EQU 8(GNLPF+1):3
122 00000 GMHFF: ERU 8(GKD0F+1):18
123 00000 GAFRED: EQU 8(GMHFF+1):2
124 00000 SNOUT: EQU 8(SWPRED+1):X
125 00-00 THREEH: EQU 8(640UT):12
                                   ; = 526, USED TO AVOID BURNING
```

PROM BEFORE FINDING OFTIMUM VALUE

126 00000

.

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 7
128	00000	;**;**********************************
129	00000	* INPUT/OUTPUT DEFINITIONS *
130	00000	; ************************************
131	00000	TEST: EQU B#01
132	00000	FPRT: EQU B\$10
133	00000	FEPRT: EQU B#11
134	90000	SPRT: EQU B#10
135	00000	.SEMBL: EQU 4H\$5
137	00000	Filest switch input definitions
138	00000	TESTFEL: EQU .TESTO ; Test point select 0 = FEL
139	00000	TSTFEH: EQU .TEST: Fest point select 1 = FEH
140	00000	TRIFE: EDU .TERIZ Froint select 2 = FE
141	00000	TSTE: E00 .TEST3 ; Test point select 3 = E
142	00000	TSTHPF: ERU .TEST4
143	00000	TST3DWC EGU .TEST5
144	00000	TRIBPE: ERU .TERI6 : Test point select 6 = SPE
145	00000	TSTS: EGU .TEST7 # Test point select 7 = \$

ENDIFOR

; Jump into main prodedure at

Start_of_Frame entry point

30000

00009

0000A

176

178

REPLE

JMF ,(SOFENT):2

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 9
160	0000A	;*************************************
181	0000A	
182		# (This procedure does the calls for all filtering #
183		** operations in the synthesis processor, as well as all the*
154		## input. All input and output operations are done after #
185		# S enable soes low but before S clock soes high. Sample *
186		# output takes place after E enable soes high and all #
187		** calculations for the sample interval are complete
188	0000A	***************************************
189	0000A	f Start of samele interval 0
190	0000A	SOFENT: JMF .SOF.SOFENT:X Wait for start of frame
		00110111 00000500 10100115 11111111 11000000 00000000
191	00005	
		00010000 00000101 01:00110 11111111 11000000 00000000
		i Start of sample interval 1
194	30000	
		00010000 00000101 01100110 11111111 11000000
104	ACAAT:	Figure of sample interval 2
		CALL FEVERY:X
177		00010000 00000101 01100110 11111111 11000000
		00010000 00000000
199	0000E	f Start of seasle interval 3; set FEL
200		CALL FEVERY:
		00510000 00005101 01100110 11111111 11000000 00000000
201		CALL .RMFL:% Do Low RMF filter stuff
		00010000 00001010 00010110 11111111 11000000
203	00010	
204	00010	CALL FEVERY1%
		00010000 00000101 01100110 11111111 11000000
205	00011	LOAD FOF f Input P(0)
		11100000 01000001 00100100 00010010 11000001 10000100
206	00012	STHS FTMP0,FTMP0 ; Store in PTMP0
		11100600 01600000 10010166 00001001 11000000 00101010
900	00017	* Start of sample interval 5, set P(1)
	00013	
207	00013	00010000 00000101 01100110 11111111 11000000
210	00014	LOAD F.F ; Input P(1)
210	00014	11100000 01000001 00100100 00010010 11000001 10000100
211	60015	STMS (PTMPO+D41):%;(PTMFO+D41):%; Store in Ptmp1
	00010	11100000 01000000 10100100 00001010 11000000
		TANAMA TANAMA TANAMA AND TANAMA AND TANAMA
213	00016	; Start of sample interval 6, set P(2)
214	00016	CALL *EVERY:X
		00010000 00000101 01100110 11111111 11000000
215	00017	LOAD P:P ; Input P(2)
		11100000 01000001 00100100 00010010 11000001 10000100
216	00018	STMS (PTMPO+D#2):%,(PTMPO+D#2):%; Store in Ptmp2
		11100000 01000000 10110100 00001011 11000000

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 10
	00019 00019	Start of sample interval 7 CALL FEVERY: X
220		00010000 00000101 01100110 11111111 11000000
		11100000 01000000 16010100 00601001 11000001 10000100
221	0001B	STMS LSTF,LSTF
223 224	0001C 0001C	f Start of samele interval 8, Get FEH CALL *EVERY:2,
		00010000 00000101 01100110 11111111 11000000
225		CALL : GMFH: X ; Do GMF for FEH input 00010000 00000111 10000110 111111111 11000000
226	0001E	LDAD (PTMF0+D40):X,(PTMF0+D40):X ; Get P(0) for test point
227	00017	11100000 01000000 10010100 00001001 11000001 10000100 STH5 LSTF;LSTF
		11100000 01000001 01000100 00010100 11000000
		; Start of sample interval 7, set F(3)
230	00026	CALL *EVERY:X 00010000 00000101 01100110 11111111 11000000
. 231	00021	LOAD PyP # Input P(3)
232	00022	11100000 01000001 00100105 00010010 11000001 10000100 STMS (FTMF0+D43):%,(FTMF0+D43):% # Store in Ptmr3
233	00023	11100000 01000000 11000100 00001100 11500000 00101010 LOAB (PTMF0+D#1):%,(PTMF0+D#1):%; Get F(1) for test point
200	00023	11100000 01000000 10100100 0000101 11000001 10000100
234	00024	STM3 LSTF+LSTF 11100000 01000001 01000100 00010100 11000000
236 237	00025	<pre>f Start of sample interval 10, set P(4) CALL ,EVERY:12</pre>
		06010000 06000105 01100110 11111111 11000000 60000000
238	00026	LDAD FyP
239	00027	STMS (PTMP0+D#4):%,(FTMP0+D#4):%; Store in Ptmp4 11100000 01000000 11010100 00001101 11000000
240	00028	LDAD (FTMF0+D\$2):%,(PTMF0+D\$2):%; Get P(2) for the test point
241	00029	11100000 01000000 10110100 00001011 11000001 10000100 STME LSTP,LSTP
2712	00027	11100000 01000001 01000100 00010100 11000000
243	0002A	<pre>f Start of sample interval 11, set P(5)</pre>
244	0002A	CALL FEVERY:X 00010000 00000101 01160110 11111111 11000000 00000000
245	0002B	LOAD PyP ; Input P(5)
246	00020	11100000 01000001 00100100 00010010 11000001 10000100 STMS (PTMF0+D#5):%,(PTMF0+D#5):%; Store in Ptmp1
		11100000 01000000 11100100 00001110 11000000
247	0002B	LOAD (PTMF0+D#3):%,(PTMF0+D#3):%; Get P(3) for test Point 11100000 01000000 11000100 00001100 11000001 10000100
248	0002E	STMS LSTP.LSTP
		11100000 01000001 01000100 00010100 11000000

```
LINE
        SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 13 PAGE 11
250
    0002F ; Start of sample interval 12
251
    0002F
              CALL , EVERY: 2
        252
              LOAD (PTMF0+D$4):X.(PTMF0+D$4):X : Get P(4) for test point
    00030
        11100000 01000000 11010100 00001101 11000001 10000100
203
    00031
              STMS LSTP , LSTP
        00032 ; Start of sample interval 13, GET FEL
256
    00032
              CALL , EVERY:2
        CALL . OMFL:%
                              ; Fut the QMF filter
        00034
              LOAD (PTMPO+D#5):X, (PTMPO+D#5):X ; Get P(5) for test point
258
        11100000 01000000 11100100 00001110 11000001 10000100
    00035
              STMS LSTP , LSTP
        11100060 01000001 01000100 00010100 11000000 00101010
261
    00036 : Start of sample interval 14, get F(6)
              CALL , EVERY: 2
        263
    00037
              LOAD PIP
                               : Input P(6)
        11100006 01000001 00100100 00010010 11000001 10000100
    00038
              STMS (PTMF0+D46): Z, (PTMF0+D46): Z
        11100000 01600000 11110100 00001111 11000000 00101016
    00039
              STMS LSTP LSTP
        267
    0003A ; Start of sample interval 15, Get P(7)
268
    0003A
              CALL , EVERY : %
        267
    0003B
              LUAD P.F
                              i Input P(7)
        11100000 51005661 06160100 06010010 11000061 10000100
   00038
              STMS (FTMF0+D47):X (FTMF0+D47):X ; Store in Ptems7
        271
   6003B
              STMS LSTP, LSTF
        273
   0003E : Start of sample interval 16
274
   0003E
              CALL . EVERY: %
        FILTER1 ,CONO,10(0):%
                              # Load zero into Acc
        00040
              FILTERL
        277
   00041
              STMS LSTP, LSTP
                              ; Set test point predictor to zero
        278
   00042
              CALL , CFYPRED: %
                               ; Copy predictors from temporary buff
        279
   00043
                               ; to the active area
281
   00043 ; Start of sample interval 17
   00043
              CALL , EVERY: %
```

284	00044	; Start of sample interval 18; set FEH
285	60044	CALL *EVERY:2
		60010000 00000101 01100110 11111111 11000000
256	00043	CALL .GMFH:X
		00010000 00000111 10000110 11111111 11000000
288	00044	; Start of sample interval 19
	00046	
		00010000 00000101 01100110 11111111 11000000
291	00047	# FOR n := 0 TO 13 DO
		PUSH D413:% Define next instruction
		01000000 00000000 11010110 11111111 11000000
295	84600	
	00048	
		# Start of sample interval 20 + 10m
		CALL FEVERY!X
		06010000 00000101 01100110 11111111 11000600 00000000
592	00045	F Start of sample interval 21 4 10m
		CALL FEVERY:X
		06010600 00000101 01106110 11111111 11000000 00000000
701	58344	+ Start of sample interval 22 + 10m
		CALL *EVERY:%
202		06010060 00006101 01100110 11111111 11000000 00000000
304	00048	# Start of sample interval 23 + 10n, input FEL
		CALL FEVERY:12
		00010000 00000101 01100110 11111111 11000000
30€	00040	CALL *RMFL:X
		CALL *RMFL:X
308	00045	+ Start of sample interval 24 + 10n
309	00041	CALL DEVERY:X
		00010000 0000010: 01100110 11111111 11000000 00000000
311	0004E	; Start of sample interval 25 + 10n
	0004E	
		00010000 00000101 01100110 11111111 11000000
314	0004F	f Start of sample interval 26 f 10n
315	0004F	CALL *EVERY:X
		00010000 00000101 01100110 11111111 11000000
317	00050	; Start of sample interval 27 + 10n
318	00050	CALL FEVERY:X
		00010000 00000101 01100110 11111111 11000000
320	00051	; Start of sample interval 28 + 10m, Input FEH
321	00051	CALL FEVERY:X
		00010000 00000101 01100110 11111111 11000000
322	00052	CALL ;QMFH:% ; Input and filter FEH
		00010000 00000111 10000110 11111111 11000000
100000000000000000000000000000000000000		

	324	00053	# Start of sample interval 29 # 10n
	325	00053	CALL *EVERY:X
)			00010000 00000101 01100110 11111111 11000000
	326	00054	REPLP ; ENDFOR
			10000000 01101111 11110110 11111111 11000000
	327	00055	JMP .SOFENT:: Do everything again for next fram
			00110000 00000000 10100110 11111111 11000000

LINE ADDA SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 13

```
350 00056 ** PROCEDURE operations_done_EVERY_sample_interval *
331
   00056 ## { This procedure performs all the operations which are #
337 00056 ## performed every sample interval. All the input ports are #
333 00056 i* read immediatly after SCLCK sees low. The required input *
334 00005 f* data is transferred to working memory in the main procedure*
335 00056 ## 3
336 00056 1*
            Wait for rising edge of S enable complement.
337 00056 ## Output S#
338 00036 ## Input FE,P#
339 00056 ix Read test point switch and output appropriate data;
340 00056 ## Move data to the delay buffer;
341 00056 it Low pass the reconstructed residual;
            Ferturb and high pass the reconstructed residual;
342 00056 ##
343 . 00056 ik Sum the high and low pass residuals!
344 00056 18
            Fass data through the predictor
345 00056 3%
            Preemphasiz the dataf
346 00056 1x
            Smaft the ring buffert
347 00056 4% Store zero in the ring buffer base! (required for inter*
348 00006 14
                                        Polation. }*
347 00056 ##
351 00056 EVERY: JMF .SEMBL/EVERY: 1 Wait for .8 Enable to be low
         00037 WAITLP: UMF .SENEL: DUTLP: N ; Wait for .S Enable to be high
         JMP .WAITLP:%
         354 00059 ; Now output 8
355 00059 DUTLP: LDMR 8 # Load 8 into the Acc
         11100000 01010000 00000001 00110000 11000001 10000100
          OUTFUT SPRIFBERT : Output S
356
         357 0005B : Now input from both input ports in case we need to read them
           INPUT GNINPUT: FEPRI: NULTI : Get FEL or FER (depending on same
358 00055
         11100000 01010111 00000110 01100000 11100000 01000100
         FILTERL | | And double it
     STLS ,FE,FE ; Store it away for possible later use
         11100000 01000001 00116100 00010011 11000000 00111001
              INPUT , PPRT, ; Input a predictor
         11100000 01010000 00000110 01000000 11000001 10000100
          STMS Fif | Store it away for possible later use
         363
    00060 ; Test the 'TEST PDINT SELECT' switch and output the data to test port
               JMP TSTFEL.NOTFEL:%; Test if FEL to be output
364 00060
         LOAD LOTFEL, LOTFEL ; Yes - Load FEL into Acc
         11100000 01000001 01010100 00010101 11000001 10000100
366 00062 NOTFEL: JMP TSTFEH: NOTFEH: % ; Test if FEH to be output
         357 00063
              LOAD LSTFEH: LSTFEH ; Yes - Load FEH into Acc
```

11100000 01000001 01100100 00010110 11000001 10000100

ADDE SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 14

```
LINE
     ADDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 13 PAGE 15
    00064 NOTFEH: JMF TSTFE, NOTFE: % ; Test if FE to be output
368
          LOAD LOTFE: LOTFE : Yes - Load FE into Acc
309
    00055
          11100000 01000001 01110100 00010111 11000001 10000100
     60666 NUTFE: JMP TSTE, NOTE:2
                              ; Test if AEDI to be output
         LDMR (LPFBS+LPFSZ+D#1):% ; Yes - Load AEDD into Ace
         11100000 01010000 00000000 11100001 11000001 10000100
     00068 NOTE: JMP TSTHPF:NOTHPF:%; Test if HPF to be output
         LDMR (HPFBS+HPFSZ+D#1):2 ; Yes - output HPF
    00069
         11100000 01010000 00000001 01011000 11000001 10000100
    0006A NOTHER: JMP TSTSUM, NOTSUM: % Test if regenerated residual to be outpu
         LDMR (SUMBS+D41):2 ; Yes - output sum
         11100000 01010000 00000000 11100010 11000001 10000100
    00060 NOTSUM: JMF TSTSPE: NOTSPE: X : Test if SPE to be output
          LDMR (FREDBS+D#1):1 ; Yes - Load SPE into Acc
    00061
          11100000 01010000 00000001 00100110 11000001 10000100
    0006E NOTSFE: JMF TSTS:NOTS:% ; Test of S to be outout
         379
                LDMR (DEMES+D#2):% ; Yes - Load S into Acc
         11100000 01010000 00000001 00110000 11000001 10000100
    00070 NOTS: OUTPUT TEST, TEST ; Send the value to the analog test port
380
          11160000 01100010 060000110 00100000 11000000 00100011
381
    00071 ;
             CALL :LPF:X ; Low pass the unperturbed samples
382
    00071 #
                CALL HPF:2
                              i Hish pass the perturbed samples
383
    00071 +
               FILTER: ,CON:, (LPFBS+LPFSZ):%; add the HPF and LPF signals
384
    00071 ;
               FILTERN +COM1+(HPFBS+LPFSZ):2
385
    00071 ;
               FILTERL
326 00071 +
              ST_R ,SUMBS
387
    00071 | Now pass the data through the predictor
           CALL ,PRED:%
         369
    00072 | Now deemshasize the data and shift rins buffer data
390
    00072
                CALL
                    ,DEEM: %
         391
    00073 # Now load zero into base of rins buffer
392
    00073
                SFL ****.SHIFT | Shift rins buffer data
         00074
               FILTER1 .CONO.10(0):% ;
          11100000 01010000 00000000 00000000 11100000 01000100
394
    00075
                FILTERL
         395
    00076
                STHR
                      , (PERTBS-1):%
         396
    00077
                    CRTN. QMFBS i Store zero and return
                STMR
```

```
SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 13 PAGE 16
LINE
398
     - 00078 - 专家市总术主义等家家的工作的工作。 100078 - 专家市总术主义的工作的工作, 100078 - 专家市总术主义的工作的工作的工作的工作的工作的工作的工作的工作的工作的工作。
 395
      00078 ## PROCEDURE Quadrature Mirror Filter High
      00078 1#4 This procedure performs a quadrature mirror filter operation#
 400
      00076 ## assuming that FEH is in the FE input buffer. It place FEH at #
      00078 ## the start of the QMF portion of the ring buffer.
 403
      00076 1*
      00078 ## sum := FEHEn3 # CGNFE03
 404
     00078 ## FOR i := 1 TO 17 DO
 405
     00078 ## sum := sum + FEHEn + 2*i] * CRMFE2*i];
 406
     00078 ## sun := FELEn] # 9MFE03 - sun;
 407
408
     00078 ## FOR i := 0 TO 17 DO
 400
      00078 ## sum := sum + FELEn + 2*i3 * CRMFE2*i];
410
     00078 ## LFFBSE03 := sum#
      00078 ** END:
411
412
     00078 QMFH:
                       LOAD
                               FE .FE
                                                     Fetch FEL from the FE input b
             11100000 01000001 00110100 00010011 11000001 10000100
                       STME
                               , OMFPS
                                                    + Store at base of QMF ring buf
             11100000 30111111 11119000 00000000 11000000 00101010
                       STORE
                               ,LSTFEH,LSTFEH
                                                    i Store for the test point
             416
      00077
                       FILTER: (COMF+D40):X-(OMFBS+D40):X ; Do the FEH terms first
             11100000 01010010 00110000 00000000 11100000 01000100
 417
      00070
                       FILTERN +(COMF+D#2):% (QMFBE+D#10):%
             11100000 01010010 01010000 00001010 11100010 11000100
 418
      00070
                      FILTERN ,(DGMF+D$4):%,(QMFES+D$20)::
             11100000 01010010 01110000 00010100 11100010 11000100
 419
      0007E
                      FILTERN , (CQMF+B+6):X, (QMFBS+D+30):X
             11100000 01010010 10010000 00011110 11100010 11000100
      0007F
                      FILTERN (COMF+D&8):%,(OMFBS+D#40):%
             11160000 01010010 10110000 00101000 11100010 11000100
421
      00060
                      FILTERN > (CRMF+D410):%: (RMFBS+D450):%
             11100000 01010010 11010000 00110010 11160010 11000100
                      FILTERA . (CRMF+D#12);%, (ONFES+D#60);%
             11100000 01010010 11110000 00111100 11100010 11000100
423
      00082
                      FILTERN , (CGMF+D#14): %, (QMFBS+D#70): %
             11100000 01010011 00010000 01000110 11100010 11000100
424
      00003
                      FILTERN (COMF+D&16):2, (GMFBS+D+80):2
             11100000 01010011 00110000 01010000 11100010 11000100
425
      00034
                      FILTERN , (COMF+D#18): %, (QMFPS+D#90): %
             11106000 01010011 01010000 01011010 11100010 11000100
                      FILTERN , (COMF-D#20):X, (GMFBS+D#100):X
426
      00085
             11100000 01010011 01110000 01100100 11100010 11000100
427
      00086
                      FILTER4 ,(CRMF+D#22):X,(QMFBS+D#110):X
             11100000 01010011 10010000 01101110 11100010 11000100
428
      00087
                      FILTERN , (CQMF+D#24): %, (QMFBS+D#120): %
             11100000 01010011 10110000 01111000 11100010 11000100
      00088
                      FILTERN , (CQMF+D#26): Z, (QMFBS+D#130): Z
             11100000 01010011 11010000 10000010 11100010 11000100
430
                      FILTERN , (CQMF+D#28): X, (QMFBS+D#140): X
      00089
             11100000 01010011 11110000 10001100 11100010 11000100
431
      0008A
                      FILTERN , (CRMF+D#30): Z, (RMFBS+D#150): Z
             11100000 01010100 00010000 10010110 11100010 11000100
```

.

LINE	PPPR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 17
432	88000	FILTERN ,(CQMF+D\$32):X,(QMFBS+D\$160):X
433	00080	11100000 01010100 00110000 10100000 11100010 11000100 F1LTERN *(CQMF+D*34):%*(QMFBS+D*170):%
434	00081	11100000 01010100 01010000 10101010 11100010 11000100 FILTRNM ,(CQMF+D#00):%;(QMFBS+D#5):% ;sum:=FELE03+QMFE03-sum
435	000SE	11100000 01010010 00110000 00000101 11100110 11000100 FILTERN (CRMF+N402):X,(RMFBS+D415):X
436	000BF	11100000 01010010 01010000 00001111 11100010 11000100 FILTERN ,(CRMF+P#04):X,(QMFBS+D#25):X
437	00090	11100000 01010010 01110000 00011001 11100010 11000100 FILTERN *(CQMF+D*06):X*(QMF88+D*35):X
438	00001	11100000 01010010 10010000 00100011 11100010 11000100 FILTERN (CQMF+D#08):%,(QMFES+D#45):%
439	00092	11100000 01010010 10110000 00101101 11100010 11600100 FILTERN (CQMF+D410):X:(QMFBS+D455):X
440	00073	11100000 01010010 11010000 00110111 11100010 11000160 FILTERN +(CRMF+D+12); X+(RMFRS+D+65); X
	00073	11100000 01010010 11110000 01000001 11100010 11000100
441		FILTERN (CRMF+D#14);%,(RMFBS+D#75);% 11100000 01010011 00010000 01001011 11100010 11000100
442	00095	FILTERN *(COMF+D416):%;(QMFBS+D485):% 11100000 01010011 00110000 01010101 11100010 11000100
443	00095	FILTERN ,(CRMF+D#18):%,(RMFFS+D#95):% 11100000 01010011 01010000 0101111 11100010 11000100
444	06097	FILTERN +(CGMF+D#20):%+(GMFBS+D#105):% 11100000 01010011 01110000 01101001 11100010 11000100
445	00095	FILTERW *(CRMF+D*22):X*(QMFBS+D*115):X 11:00000 01010011 10010000 01110011 11100010 11000100
446	00077	FILTERN ;(COMF+D\$24):X;(QMFBS+D\$125):X 11100000 01010011 10110000 01111101 11100010 11000100
447	0009A	FILTERN +(CRMF+D\$26):%+(RMFRS+D\$135):% 11100600 01010011 11010000 10000111 11100010 11000100
448	00093	FILTERN +(CQMF+D#28):%;(QMFBS+D#145):% 11100000 01010011 11110000 10010001 11100010 11000100
447	00050	FILTERN ,(COMF+D#30):%,(QMFBS+D#155):% 11160000 0101010C 00010000 10011011 11100010 11000100
450	00091	FILTERN ,(CRMF+D#32):X,(RMFBS+D#165):X 11100000 01010100 00110000 10100101 11100010 11000100
451	00098	FILTERN ,(CRMF+D#34):%,(QMFBS+D#175):%
452	00093	11100000 01010100 01010000 10101111 11100010 11000100 FILTERL
453	050A0	11100000 01101111 11110110 11111111 11100000 11000003 JMF
454	000A1	00:10000 00001100 10:00110 11:111111 11:00000 00000000

```
LINE
             APDE
                         SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 13
                                                                                                                     PAGE
                                                                                                                               18
           456
           000A: 1* PROCEDURE Quadrature_Mirror_Filter_Low
           000A1 ## { This procedure does the quadrature mirror filtering*
 458
           000A1 ## operation assuming FEL is in the FE input buffer.
 460
           000A1 ##
           000A1 ## sum := FEHEn+03 * CQMF[1]
           000A1 ## FOR i #= 1 TO 17 DO
 462
 453
           000A1 | sum := sum + FEHEn+2%i3 * CRMFC2%i+1];
           000A1 ## FOR i := 0 TO 17 DO
 464
           000A1 ## sum := sum + FEHEn+2*i] * CQMFE2*i+1];
          000A1 ## LPFBS != sum #
 466
 467
           00041 ##
 468 000A1 PREFERENCE FERENCE F
 467
           060A1
                                        LOAD
                                                        FE,FE
                       Qhi-L:
                       11100000 01000001 00110100 00010011 11000001 10000100
                                        STHE
                                                        . OMFBS
           VVVH2
                                                                                               ; Load FEL and store in Rins b
                       00043
                                        STURE
                                                       ,LSTFEL,LSTFEL
                                                                                              i Store for the test point
                       00004
                                        FILTER: (CQMF+D41):X;(QMFBS+D45):X ; Do the FEH terms first
                       11100006 01010010 01000000 00000101 11100000 01000100
                                        FILTERN . (COMF+D#3):2. (QMFBS+D#15):2
 473
           00045
                       11100000 01616016 01160000 00001111 11106010 11060100
 474
           000A6
                                        FILTERN + (COMF+P#5):X+(OMFBS+D#25):X
                       11100360 61610010 10000000 30011001 11100010 11000100
           000A7
                                        FILTERN (CQMF+D$7):X, (QMFBS+D$35):X
                       11100006 01010010 10100000 00100011 11100010 11000100
           36005
 476
                                        FILTERN + (COMF+D#9):X+(QMFBS+D#45):X
                       11160000 01010010 11000000 00101101 11100010 11000100
 477
                                        FILTERN , (COMF+D#11):2, (OMF35+D#55):2
           00049
                       11100000 01010010 11100000 00110111 11100010 11000100
 478
           00046
                                        FILTERN +(CGMF+D#13):%+(GMFBS+D#65):%
                       11100000 01010011 00600000 01000001 11100010 11000100
 479
                                        FILTERN + (COMF+D415):X+ (QMFBS+D475):X
           COOAH
                       11100000 01010011 00100000 01001011 11100010 11000106
 480
                                        FILTERN +(CQMF+D#17):X+(QMFBS+D#85):X
           000AC
                       11100000 01010011 01000000 01010101 11100010 11000100
 481
           000AL
                                       FILTERN , (COMF+D+19): X, (QMFBS+D+95): X
                       11100000 01010011 01100000 01011111 11100010 11000100
 482
           OGGAE
                                       FILTERN (COMF+D#21):X, (OMFBS+D#105):X
                       11100000 01010011 10000000 01101001 11100010 11000100
 483
           000AF
                                        FILTERN , (CRMF+D#23): X, (RMFBS+D#115): X
                       11100000 01010011 10100000 01110011 11100010 11000100
 484
           000B0
                                        FILTERN + (COMF+D#25):2, (QMFBS+D#125):2
                       11100000 01010011 11000000 01111101 11100010 11000100
 485
           00001
                                       FILTERN + (COMF+D#27):2, (QMFBS+D#135):%
                       11100000 01010011 11100000 10000111 11100010 11000100
 486
           000B2
                                        FILTERN , (CRMF+D#29):Z, (RMFBS+D#145):Z
                       11100000 01010100 00000000 10010001 11100010 11000100
 487
           000B3
                                        FILTERN , (CQMF+D#31): Z, (QMFBS+D#155): Z
                       11100000 01010100 00100000 10011011 11100010 11000100
          000B4
                                       FILTERN (CRMF+D#33): %, (QMFBS+D#165): %
```

11100000 01010100 01000000 10100101 11100010 11000100

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 19
489	000B5	FILTERN ,(CQMF+D#35):Z,(QMFES+D#175):Z
407	OVOLU	11100000 01010100 01100000 10101111 11100010 11000100
490	000Bá	FILTERN (CRNF+D#1):%,(RMFFS+D#10):%; Now do FEL term
470	VVVDC	11100000 01010010 01000000 00001010 11100010 11000100
471	000B7	FILTERN ,(CQMF+D#3):X,(QMFB5+D#20):X
471	00057	11100000 01010010 01100000 00010100 11100010 11000100
492	000B8	FILTERN ,(CQMF+D#5):2,(QMFBS+D#30):2
474	VVVDO	11100000 01010010 10000000 00011110 11100010 11000100
493	000B9	FILTERN ,(CGMF+D#7):X,(QMFBS+D#40):X
473	DUUDT	11100005 01010010 10100000 00101000 11100010 11000100
404	AAAAA	FILTERN ,(CGMF+D#9):X,(GMFBS+D#50):X
474	AECOO	
495	AAAADD	11100000 01010010 11000000 00110010 11100010 11000100
473	OOOEB	FILTERN +(COMF+D#11):Z,(GMFBS+D#60):Z
401	00000	11100000 01010010 11100000 00111100 11100010 11000100
496	00080	FILTERN ,(COMF+D#13):%,(QMFBS+D#70):%
407	****	11163000 01010011 00000000 01000110 11100010 11000100
497	000BD	FILTERN (COMF+D+15):X, (QMFBS+D+80):X
400	AAAA	11100000 01010011 00100000 01010000 11100010 11000100
498	000BE	FILTERN *(COMF+D*17):X*,(QMFBS+D*90):X
		11100000 6161001: 01000000 01011010 11100010 11000100
499	000BF	FILTERN *(COMF+D*19):%:(OMFES+D*100):X
	*****	11100000 01010011 01100000 01100100 11100010 11000100
500	00000	FILTERN (CQMF+D#21):%,(QMFBS+D#110):%
		11100000 01010011 10000000 01101110 11100010 11000100
501	00001	FILTERA +(CRMF+D*23):X+(RMFBS+D*120):X
		11100000 01010011 10100000 01111000 11100010 11000100
502	00002	FILTERW (COMF+D#25):X,(QMFBS+D#130):X
		11100000 01010011 11000000 10000010 11100010 11000100
503	60003	FILTERN *(COMF+D427):X*(QMFBS+D#140):X
	00001	11160060 01010011 11100000 10001100 11100010 11000100
504	00004	FILTERN (COMF+D#29):%, (OMFBS+D#150):%
		11100000 01010:00 00000000 10010110 11100010 11000100
505	00005	FILTERN (COMF+D#31):Z,(OMFBS+D#160):Z
		11100000 01010100 00100000 10100000 11100010 11000100
506	000Co	FILTERN (CRMF+D#33):%,(RMFBS+D#170):%
		11100000 01010100 01000000 10101010 11100010 11000100
507	00007	FILTERN (CRMF+D#35):Z;(RMFBS+D#180):Z
Alg. C		11100006 01010100 01100000 10110100 11100010 11000100
508	00008	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
509	00009	JMF FERT:X Ferturb sample location
		00110000 00001100 10100110 11111111 11000000
510	000CA	i and return
511	000CA	; return

```
ADDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 20
514 0000A i* PROCEDURE PERTurbate;
 515 00004 ##
    00000 ##6 This procedure perterbates the location of the FE
516
    00000 IX BEGIN
 517
    60004 ## Multiply accumulator by CON4;
 518
519
    0000A ## Generatate seed uniformly distributed in [-thresh-thresh]#
520 0000A ## IF magnitude(FE) > seed THEN store at PERTES
521
   0000A ## ELSE IF MSB of least is one THEN store at PERTRS-1
   0000A ## ELSE store at PERTRS+1
 522
523
   0000A ## END#
524 000CA 1%
525 000CA ** ( The Fe sample is perturbed with the following probability*
526 0000A ## density function.
527 000CA ##
528 0000A An Probablilty of being
529 000CA ** Perturbed
530
   0000k ##
                         +0.5
531 0000A #X
532 00004 14
                        1 +
533 00009 14
                          1
534 000EA 18
533 0000A ##
536 0000e ## 4
    0000A # ----
537
538 000CA ## -THRESH
                          0
                                      THRESH
539 000CA ##
                    Input sample value
540 000CA ** END:
542 00000 PERT:
543 00004 | New do a fell precision multiple by CONS
   00006 Silf (LEAS) | Store least sig prod
        0000B STMR #MOST ; store most sis product
        546 00000 FILTER: CONS. MOST | Multiple most sis by CONS
         11100000 01010110 11100011 11111110 11100000 01000100
547 000CB
             FILTERL
         000CE
          STLR ,MOST
        OGOCF LDMR MOST
                            # Shift left 16 places
        11100000 01010000 00000011 11111110 11000001 10000100
    00000
              FILTRAU (CONS, LEAST | Add CONS times least sis product
         11100000 01010110 11100011 11111111 11000010 01000100
    000B1
            FILTERL
        000D2
          STLR
                    , LEAST
     000D3
              STMS LSTFE, LSTFE ; Save for the test point
        554 00004 STMR ,MOST ; Store at base of LPF
```

```
ADUR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 21
LINE
    000D5 $
             STMR
                      ,DLYBS
555
    (000) ; Take the absolute value
556
     00005
                JMP NEG:NEGATE:% ; Test last bus value was negative
          .POSTV:%
     00006
          000D7 NEGATE: FILTER: .CONKI. MOST:X ; IF nesative them nesate it
          11100000 01010000 00100011 11111110 11100000 01000100
     00008
                FILTERL
          STLR ,MOST
          11100000 0011:111 11110011 11111110 11000000 00111001
     000DA POSTVI
                FILTER: THRESH, LEAST | Scale [-1:1) remainder
     000DA
563
          11100000 01010111 01100011 11111111 11100000 01000100
564
     00000
                                   ; to I-THRESH, THRESH)
                 FILTERL
                                   ; To set random seed
 565
     COODE
          STMS TEMP TEMP
                                  i leave result in LSW
 566
          11100000 0100000: 10000100 00011000 11000000 00101010
                SPL .. (D4256*COEFB4CON1):X. (D4256*SCRATHTEHF):X. MULTI
 567
          11100000 01010000 00010100 00011000 11100000 01000100
                                   i loss seed into accumulator
 568
     COORE
                FILTERN .COMMI: MOST
                                   ; Subtract absolute value of sismal
 569
     DOODE
          11100000 01010000 00100011 11111110 11100010 11000100
                FILTERL
          11100000 0110111: 11110110 11111111 11100000 11000000
                 STLR #MOST
                                    ! Put on the bus
     000E0
          JMP NEG-NOPERTIX # If magnitude of signal > 126 don't
 572
     000E1
          FILTER: (CLPF+D01E):%.LEAST (scramble bits in LEAST
     000E2
 573
          11100000 01010001 00110011 11111111 11100000 01000100
 574
     000E3
                FILTER
          ; Save least sig word of scrambled 1
                 STLR *LEAST
     000E4
          LDMR LEAST
                                    ; Fut MSB of LEAST into sign bit
 576
     000E5
          11100000 01010000 00000011 11111111 11000001 10000100
                     NEG PERTP: %
     000EE
                 JMP.
          LOAD LETFE, LETFE
     000E7
 578
          11100000 01000001 01110100 00010111 11000001 10000100
                 STMR CRTN, (LPFBS-D#1):%
 579
     000E8
          000E9 PERTP: LOAD LSTFE, LSTFE
 580
          11100000 01000001 01110100 00010111 11000001 10000100
               STMR CRTN, (LFFBS+D#1):X
     000EA
          OODER NOPERT: LOAD LSTFE, LSTFE
          11100000 01000001 01110100 00010111 11000001 10000100
                 STMR CRTH, (LPFBS+0):X
 583
     OOGEC
```

LINE	HINA	STREETS FREDESDUN MOSERNLER SOURCE FILE. SEFT 13 FADE
585	000F1	;*************************************
586	000ED	
587		##: This Procedure low pass filters the data in the low pass*
588		## Filter buffer. 3*
589		# sum := CLPFE01 & LPFBSE01: *
590	000ED	** FOR i := 1 TO 30 DO **
591	OGGED	<pre>i* sum := sum + CLPFEi3 * LPFRSEi3; *</pre>
592	OOOEB	# LPFBS[31] := su#;
593	GOOED	<pre># DLYBS[0] := DDFBS[0];</pre> #
594	000EI	** END;
595	000EF	
596		;*************************************
597		LPF: FILTER1 ,(CLPF+D#0):X,(LPFBS+D#0):X
		11100000 01010000 01000000 11000001 11100000 01000100
578	ODGEE	FILTERN (CLPF+D+1):X,(LPFBS+D+1):X
		11100200 01010000 01010000 11000010 11100010 11000100
-599	000EF	FILTERN +(CLPF+D#2):X+(LPFBS+D#2):X
		11100000 01010000 01100000 11000011 11100010 11000100
600	000F0	FILTERN (CLPF+D#3):X,(LPFAS+D#3::X
		11100000 01010000 01110000 11000100 11100010 11000100
601	000F1	FILTERN (CLPF+D\$4):X (LPFBS+D\$4):X
		11100000 01010000 10000000 11000101 11100010 1100010
602	000F2	FILTERN ,(CLFF+D45):X,(LFFBS+D45):X
		11100000 01010000 10010000 11000110 11100010 11000100
603	000F3	FILTERN (CLPF+D+6):%(LPFBS+D+6):%
		11100000 01010006 10100000 11000111 11100010 11000100
604	000F4	FILTERN +(CLPF+D#7):X+(LPFBS+D#7):X
		11100000 01010000 10110000 11001000 11100010 11000100
605	000F5	FILTERN (CLPF+D#8):X,(LPFBS+D#8):X
161	000F6	11100006 01010000 11000000 11001001 11100010 11000100
606	VVVFO	FILTERM *(CLPF+D49):%*(LPFE8+D49):% 11100000 01010000 11010000 11001010 11100010 11000100
6 07	000F7	FILTERN (CLPF+D410):%,(LPFB3+D410):%
ev.	00017	11100000 01010000 11100000 11001011 11100010 11000100
603	060F8	Filter ,(CLFF+D#11):%,(LFFBS+D#11):%
000	000.0	11100000 01010000 11110000 11001100 11100010 11000100
609	000F9	FILTERN +(CLPF+D#12):X+(LPFBS+D#12):X
		11100000 01010001 00000000 11001101 11100010 11000100
610	000FA	FILTERN ,(CLFF+D#13):%,(LFFBS+D#13):%
		11100000 01010001 00010000 11001110 11100010 11000100
611	000FB	FILTERN ,(CLPF+D#14):Z,(LPFBS+D#14):X
		11100000 01010001 00100000 11001111 11100010 11000160
612	000FC	FILTERR +(CLPF+D+15):X+(LFFBS+D+15):X
		11100000 01010001 00110000 11010000 11100010 11000100
613	000FD	FILTERN ,(CLPF+D#16):%,(LPFBS+D#16):%
		11100000 01010001 01000000 11010001 11100010 11000100
614	000FE	FILTERN (CLPF+D+17):X,(LPFBS+D+17):X
		11100000 01010001 01010000 11010010 11100010 11000100
615	000FF	FILTERN (CLPF+D*18):X,(LPFBS+D*18):X
		11100000 01010001 01100000 11010011 11100010 11000100
616	00100	FILTERN (CLPF+D#19):X,(LPFBS+D#19):X
147	00404	11100000 01010001 01110000 11010100 11100010 11000100
617	00101	FILTERN ,(CLPF+D#20):X,(LPFBS+D#20):X

11100000 01010001 10000000 11010101 11100010 11000100

LINE ADDR SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 22

LINE	ADDR	SYMTHESIS PROCESSOR ASSEMBLEF SOURCE FILE, SEPT 13 PAGE 23
618	3 00102	FILTERN ,(CLPF+D\$21):X,(LPFBS+D\$21):X 11100000 01010001 10010000 11010110 11100010 11000100
619	9 00103	FILTERN *(CLPF+D*22):X*(LPFBS+D*22):X 11100000 01010001 10100000 11010111 11100010 11000100
620	00104	FILTERN ,(CLPF+D#23):%,(LPFBS+D#23):% 11100000 01010001 10110000 11011000 11100010 11000100
623	00105	FILTERN ,(CLPF+D*24);%,(LPFBS+D*24);% 11100000 01010001 11000000 11011001 11100010 11000100
. 52	2 00106	FILTERN +(CLPF+D*25):%+(LPFBS+D*25):% 11100000 01010001 11010000 11011010 11100010 11000100
623	3 00107	FILTERN +(CLPF-D#26);%,(LPFBS+D#26);% 19100000 01010001 11100000 11011011 11100010 11000100
624	4 00108	FILTERN ,(CLFF+D#27):%,(LPFBS+D#27):% 11100000 01010001 11110000 11011100 11100010 11000100
62	5 00107	FILTERN (CLPF+1:428);%,(LPFB3+D428);% 11160000 01010010 00000000 11011101 11100010 11000100
62	6 00104	FILTERN ,(CLPF+D#29);%,(LPFRS+D#29);% 11100000 01010010 00010000 11011110 11100010 11000100
62	7 00_CF	FILTERN *(CLFF+D\$30):%;(LPFPS+D\$30):% 11100000 01010010 00100000 1101111 11100010 11000100
62	9 00100	FILTERL 11100000 01101111 11110110 11111111 11100000 11000000
62	9 00101	STLR ,LEAST ; Store least sim prod 11:00:00 00:11111 1:1100:1 1:111111 1:000:00 00:1100:
63	0 00105	STMR :MOST ; store most sis product 11100000 00111111 11110011 11111110 11000000
63	1 0010F	FILTER: .ONLPF.MOST
63	2 00110	FILTERL 11100000 01101111 11110110 11111111 11100000 11000000
63	3 00111	STLR :MOST 11100000 00121111 11110011 111111110 11000000 00111001
63	4 00112	LDMA MOST Shift left 16 places 11100000 01010000 00000011 11111110 11000001 10000100
63	5 00113	11100000 01010111 00100011 11111111 11000010 01000100
63	6 00114	11100000 01101111 11110110 11111111 11100000 11000000
63	7 00115	STNR CRIN, (LPFRS+LPFSZ) 1% ; Store it away 10100000 00111111 11110000 11100000 11000000

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 13 PAGE	
639	00116	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	**
640	00116	## PROCEFURE High_Pass_Filter;	*
641	00118	1%: This Procedure high pass filters the data in the high	*
642	00116		*
643	00116	## sum := CLFF[0] * HFFRS[0];	*
644	00116		*
645	00116	<pre># sum := sum + CLPF[i] * HPFPS[i];</pre>	*
646	00116	## HPFB8[31] := 5#HPFB8[15]-sum;	*
647	00116	[10] [10] [10] [10] [10] [10] [10] [10]	*
648	00116	ik	*
	00116	;*************************************	000
649 650	00116	HFF: FILTER: •(CLPF+D*0):X•(HPFBS+D*0):X	**
000	00110		
,		11100000 01010000 C100C001 00110110 11100000 01000100	
65)	00117	Filtern *(CLPF+D#1):X*(HPFBS+D#1):X	
	25445	11100000 01010000 01010001 00110111 11100010 11000100	
652	06118	FILTERN (CLPF+D42):X,(HFFBS+D42):X	
153		11166000 01016000 01100001 00111000 11100010 11006100	
653	00119	FILTERN (CLPF+D#3):%:(HPFBS+D#3):%	
		11100000 01610000 01110001 00111001 11100010 11000100	
654	0011A	FILTERN +(CLPF+D#4):X+(HPFBS+D#4):X	
,		11100000 01010000 10000001 00111010 11100010 11000100	
655	0611B	FILTERN (CLPF+D45):X, (HPFBS+D45):X	
		11100000 01010000 10010001 00111011 11100010 11000100	
656	00110	FILTERN (CLFF+D#6):X, (HPFBS+D#6):X	
		11100000 01010000 10100001 00111100 11100010 11000100	
657	00111	FILTERN (CLPF+D#7):%,(HPFBS+D#7):%	
		11100000 01010000 10110001 00111101 11100010 11000100	
658	0011E	FILTERN (CLPF+D#8):%,(HPFES+D#8):%	
		11190000 01010000 11000001 00111110 11100010 11000100	
659	6011F	FILTERW (CLFF+D49):X,(HPFBS+D49):X	
		11100000 01010000 11010001 00111111 11100010 11000100	
660	00120	FILTERN (CLPF+D+10):X,(AFFB8+D+10):X	
		11100000 01010000 11100001 01000000 11100010 11000100	
661	00121	FILTERN (CLPF+D#11):X,(HPFBS+D#11):X	
		11190960 01619900 11110001 01000001 11100010 11000100	
662	00122	FILTERN (CLPF+D#12):X,(HPFBS+D#12):X	
		11100000 01010001 00000001 01000010 11100010 11000100	
653	00123	FILTERN (CLFF+D#13):X,(HPFBS+D#13):X	
		11100000 01010001 00010001 01000011 11100010 11000100	
664	00124	FILTERN (CLFF+D#14):X, (HFFBS+D#14):X	
		11100000 01010001 00100001 01000100 11100010 11000100	
665	00125	FILTERN (CLFF+D\$15):Z, (HFFBS+D\$15):Z	
		11100000 01010001 00110001 01000101 11100010 11000100	
666	00126	FILTERN (CLPF+D#16):X, (HPFBS+D#16):X	
		11100000 01010001 01000001 01000110 11100010 11000100	
567	00127	FILTERN ,(CLPF+D#17):Z,(HPFBS+D#17):Z	
		11100000 01010001 01010001 01000111 11100010 11000100	
668	00128	FILTERN ,(CLFF+D#18);%,(HPFBS+D#18);%	
		11100000 01010001 01100001 01001000 11100010 11000100	
669	00129	FILTERN ,(CLPF+D#19):%,(HPFES+D#19):%	
		11100000 01010001 01110001 01001001 11100010 11000100	
670	0012A	FILTERN ,(CLFF+D#20):%,(HFFBS+D#20):%	
		11100000 01010001 10000001 01001010 11100010 11000100	

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 25
671	0012B	FILTERN ,(CLPF+D#21):X,(HPFBS+D#21):X
	00122	11100000 01010001 10010001 01001011 11100010 11000100
672	00120	FILTERN ,(CLPF+D#22):%,(HFFRS+D#22):%
		11100000 01010001 10100001 01001100 11100010 11000100
673	00120	FILTERN +(CLFF+D423):X+(HPFBS+D423):X
		11100000 01010001 10110001 01001101 11100010 11000100
674	0012E	FILTERN (CLPF+D#24):Z,(HPFBS+D#24):Z
		11100000 01010001 11000001 01001110 11100010 11000100
675	0012F	FILTERN (CLPF+D#25):%,(HPFBS+D#25):%
171	00130	11100000 01010001 11010001 01001111 11100010 11000100 FILTERN ,(CLPF+D#26):2,(HPFB5+D#26):2
676	00130	11100000 01010001 11100001 01010000 11100010 11000100
677	00131	FILTERN ,(CLPF+D\$27):X,(HPFBS+D\$27):X
0,,	00101	11100000 01010001 11110001 01010001 11100010 11000100
678	00132	FILTERN ,(CLPF+D428):X,(HPFBS+D428):X
0,0	00101	11100000 01010010 00000001 01010010 11100010 11000100
679	00133	Filtern (CLFF+D429):X,(HPFBS+D429):X
		11100000 01010010 00010001 01010011 11100010 11000100
680	00134	FILTERN ,(CLFF+D\$30):X,(HPFBS+D\$30):X
		11100000 01010010 00100001 01010100 11100010 11000100
681	00135	FILTERL
		11100000 01101111 11110110 111111111 11100000 11000000
682	00136	STLR *LEAST
		11160000 00111111 11110011 11111111 11000000
633	00137	STMR :MOST ; store most sis product
		11100000 90111111 11110011 111111110 11000000 00101010
684	00138	FILTER: GONLPF,MOST ; Multiply most six by GNLPF
(AF	22476	11100000 01010111 00100011 11111110 11100000 01000100
685	00139	FILTERL 11100000 01101111 11110116 11111111 11100006 11000000
686	0013A	STLF *MOST
000	00136	11100000 00111111 11110011 11111110 11000000
687	0013B	LDhF MOST ; Shift left 16 places
		11100000 01010000 00000011 11111110 11000001 10000100
685	00130	FILTRAU ,GNLPF:LEAST ; Add GNLPF times least sid product
		11100000 01010111 00100011 11111111 11000010 01000100
689	00130	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
690	0013E	STMR (HPFB5+LPFSZ):% ; Store it awas
		11100000 00111111 11110001 01010101 11000000
691	0013F	FILTER1 ,CON1,(HPFRS+LPFSZ):Z ; subtract 5*middle element
100	ANTAN	11100000 01010000 00010001 01010101 11100000 01000100 ; to set complement HFF from Li
692	00140	
693	00140	FILTRNM +CBN4+(HPFB5+B415):% 11100000 01010110 11010001 01000101 11100110 11000100
694	00141	FILTERN ,CON1, (HPFBS+D#15):%
5/1	00141	11100000 01010000 00010001 01000101 11100010 11000100
695	00142	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
696	00143	STLR CRTN; (HPFBS+LPFSZ):% ; Store that awas
		10100000 00111111 11110001 01010101 11000000

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE. SEPT 13 PAGE 26
678	00144	;*************************************
699	00144	* PROCEDURE PREDict;
700	00144	# { This procedure predictor filters data in the #
		## filter buffer of the rins buffer. }#
	00144	
		;*************************************
704		PRED: FILTER1 SCRAT, (PO+D40): 2, (PREDBS+D41): 2
		11100000 01000000 00000001 00100110 11100000 01000100
705	00145	FILTERN SERAT, (PO+D41):X, (PREDES+D42):X
		11100000 01000000 00010001 00100111 11100010 11000100
706	00146	FILTERN SCRAT, (PO+D+2):%, (PREDBS+D+3):%
		11160006 01000000 00100001 06101000 11100010 11000100
707	00147	FILTERN SCRAT, (PO+D#3): %, (PREDBS+D#4): X
		11100000 01000000 00110001 00101001 11100010 11000100
708	00148	FILTERN SCRAT, (PO+D\$4):%, (PREDBS+D\$5):X
		11100000 01000000 01000001 00101010 11100010 11000100
709	00149	FILTERN SCRAT, (PO+D#5):X, (PREDRS+D#6):X
		11103000 01000000 01010001 00101011 11100010 11000100
710	0014A	FILTERN SCRAT, (PC+D+6):%, (FREDBS+D+7):%
		11100000 01000000 01100001 00101100 11100010 11000100
711	0014E	FILTERN SCRAT, (PO+D#7):%, (PREDBS+D#8):%
		11100000 01000000 01110001 00101101 11100010 11000100
712	00140	FILTRNM COEFB.PREDSCL.PREDBS
		11100000 01010110 10110001 00100101 11100110 11000100
713	66145	
		11100060 01101111 11110110 11111111 11100000 11000000
714	0014E	i Bu a double precion multiple bs 8
	0014E	STLR FLEAST / Store least sis prod
		11100000 00111111 11110011 11111111 11000000
716	00145	
		11100000 00111111 11110011 11111110 11000000
717	00150	FILTER1 (CONG, MOST ; Multiply most sid by 8
		11100000 01010110 11100011 11111110 11100000 01000100
718	00151	
		11150000 01191111 11110116 11111111 11100000 11000000
719	00152	STLR *MOST
		11100000 00111111 11110011 11111110 11000000
720	00153	LDMR MOST # Shift left 16 places
		11100000 01010000 00000011 11111110 11000001 10000100
721	00154	FILTRAU (CON8, LEAST) Add 8 times least sis product
		11100000 01010110 11100011 11111111 11000016 01000100
722	00155	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
723	00156	STMR •PREDBS
		11100000 00111111 11110001 00100101 11000000
724	00157	FILTER1 •GNPRED:PREDBS ; Scale so no overflow in deemphasis
		11100000 01010111 01010001 00100101 11100000 01000100
725	00158	FILTERL
		11100000 01101111 11110110 111111111 11100000 11000000
726	00159	STMR CRTN+DEMBS
		10100000 00111111 11110001 00101110 11000000

LINE	ADDR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 13 PAGE
728	0015A	\$271174477777777777777777777777777777777
729	0015A	* PROCEDURE Copy_PREDictor_coefficients *
730	00154	
731	0015A	ix the temporary predictor buffer to the active predictor *
732	0015A	<pre>i* bufier.)*</pre>
733	0015A	**
734	0015A	**************************************
735	0015A	CPYFRED: LOAD (FTMF0):X,(PTMF0):X
		11100000 01000000 10010100 00001001 11000001 10000100
736	0015H	STORE +(F0):X+(F0):X
		11100000 01000000 00000100 00000000 11000000
737	00158	LOAD (FTMF0+D+1):X,(FTMF0+D+1):X
		11100000 01000000 10100100 00001010 11000001 10000100
738	0015D	STORE +8(F0+041):X,(Fv+041):X
		11100000 01000000 00010100 00000001 11000000
739	0015E	LGAD B(FTMPO+D#2):X,(FTMPO+D#2):X
		11100000 01000000 10110100 00001011 11000001 10000100
740	0015F	STORE (8(PO+D#2)1%)(PO+D#2)1%
		11100000 01000000 00100100 00000010 11000000
741	00160	LOAD 8(PTNFO+D#3/1%;(FO+D#3)1%
7.5	00//4	1:100000 01000000 11000100 00000011 11000001 10000100
742	00161	STORE ,8(FO+D#3):%,(FO+D#3):%
747	00.10	11100000 01000000 C0110100 00000011 11000000 00100011
743	00162	LGAD 8(PTMPO+D44):X, (PTMPO+D44):X 11100000 01000000 11010100 00001101 11000001 10000100
744	00163	STORE *8(PO+D*4):%*(PO+D*4):%
/77	00100	11100000 01000000 01000100 00000100 11000000
745	00164	LOAD 8(FTMF0+D#5):Z, (FTMF0+D#5):Z
,	00201	11100000 01000000 11100100 00001110 11000001 10000100
746	00165	STORE +8(F0+D+5):X+(F0+D+5):X
		11100000 01000000 01010100 00000101 11000000
747	00166	LCAD S(PTMPO+D#6):X.(PTMPO+D#6):X
		11100000 01006000 11110100 00001111 11000001 10000100
748	00167	STORE +8(F0+D#6):X+(F0+D#6):X
		11190000 01000000 01100105 00000110 11000000 00100011
749	00168	LOAD S(PTMPO+D#7):X,(PTMPO+D#7):X
		11100000 01000001 00000100 00010000 11000001 10000100
750	00169	STORE CRTM,8(F0+D#7):%,(F0+D#7):%
		10100000 01000000 01110100 00000111 11000000

LINE	ADUR	SYNTHESIS PROCESSOR ASSEMBLER SOURCE FILE, SEPT 13 PAGE 28
752	0016A	;*************************************
753	00164	* PROCEDURE DEEMPhasize*
754	0016A	# { This procedure deemphasize: data in the deemphasis *
	0015A	<pre>fx filter memory.</pre>
756	0016A	**
757	0016A	;*************************************
758	0016A	DEEM: LDMR DEMBS
		11100000 01010000 00000001 00101110 11000001 10000100
759	0016B	FILTERA (A.(DEMBS+D41):%
		11100000 01010000 00110001 00101111 11100010 01000100
760	00160	FILTERN (A.(DEMBS+D#1):%
		11100000 01010000 00110001 00101111 11100010 11000100
761	0016B	FILTERL
		11100000 01101111 11110110 11111111 11100000 11000000
762	0016E	STMR *DEMBS
		11100000 00111111 11110001 00101110 11000000
763	0016F	
		11100000 01010111 01100001 00101110 11100000 01000100
764	00170	
		11100000 01101111 11110110 11111111 11100000 11000000
765	00171	
		10100000 00111111 1111000; 00101111 11000000 00101010
766	00172	

TOTAL ASSEMBLY ERRORS = 0

CROSS REFERENCE TABLE

LAPEL	TYPE	VALUE	peri	RENCES						
ė.	A	00003	-107	108	759	760				
AFFZ	A	00003	-66	100	101	700				
ACC	A	00200	0							
		00200								
AUTO:	I		0							
AUTOL	B		0							
AUTON	D		0	454				-		
CALL	D		191	194	197	200	201	204	209	214
			219	224	225	230	237	244	251	256
			257	262	268	274	278	282	285	286
			289	296	299	302	305	306	309	312
			315	318	321	322	325	388	390	
CDDt	÷	00047	-110	111						
CHFF	A	0004A	-111	112						
CJP	Á	00003	0							
CJFP	A	0000B	0							
CJS	Ĥ	00001	C							
CLKP	A	06080	0							
CLKXY	A	00040	0							
CLFF	A	00004	-108	109	573	597	598	599	600	601
			602	603	604	605	606	607	608	609
			510	611	612	613	614	615	616	617
			618	619	620	621	622	623	624	625
			626	627	650	651	652	653	654	655
			656	65.	658	659	660	661	662	663
			664	665	666	667	668	669	670	671
			672	673	674	675	675	677	678	679
			690							
COEFB	A	00005	168	567	712					
CONC	6	00000	-104	168	275	393				
CON1	A	00001	-105	567	491	694				
CGN128	4	COUEF	-117	118						
COMIGRIE	*	00069	-113							
COM2	Ř	0004(-114	115						
COK4	A	00060	-115	116	693					
CONS	4	0006E	-115	117	546	550	717	721		
CONM1	A	00002	-106	559	569	000				
CONT	A	0000E	0							
CPYPRED	A	0015A	278	-735						
CRMF	4	00023	-109	110	416	417	418	419	420	421
Ott: II		00020	422	423	424	425	426	427	428	429
			430	431	432	433	434	435	436	437
			438	439	440	441	442	443	444	445
			446	447	448	449	450	451	472	473
			474	475	476	477	478	479	480	481
			474	483	484	485	486	487	488	489
			490	491	492	493	494	495	496	497
			498	471	500	501	502	503	504	505
			506		200	30:	302	303	304	303
CDOCCEN		00000		507						
CROSSEN	A	00008	0							

CRTN	A	0000A	396 765	579	581	583	637	696	726	750
DIFSZ	A	E0000	-65	111						
DEEM	A	0016A	390	-758						
DEMBS	A	0012E	-79	80	83	379	726	758	759	760
Prince		VVILL	762	763	765		, 20	, 00		
DEMSZ	A	00002	-59	,00	. 00					
DIR	A	00010	0							
DLY	A	00064	-60	77						
DLYBS	A	000BE	-74	75						
EVERY	A	00056	191	194	197	200	204	209	214	219
EVERI	n	00000	224	230	237	244	251	256	262	268
			274	282	285	289	296	299	302	305
			309	312	315	318	321	325	-351	351
FAIRE		A4AA4	0	012	210	210	221	323	221	221
FALSE	Å.	00001		05	360	360	413	413	469	469
FE	Ĥ	00013	-94	95	300	300	412	410	407	407
FEPRI	A	60003	-133	358	707	447	470	EAL	FFO	563
FILTER1	D		168	275	393	416	472	546	559	
			573	597	631	650	684	691	704	717
			724	763						
FILTERA	I		759	007	750	774	+50	F00	F.47	FF.
FILTERL	Ii		169	276	359	374	452	508	547	551
			560	565	570	574	628	632	636	691
			685	689	695	713	718	722	725	761
			764					100	407	
FILTERN	D		417	418	419	420	421	422	423	424
			425	426	427	428	429	430	431	432
			433	435	435	437	438	439	440	441
			442	443	444	445	446	447	448	449
			450	451	473	474	475	476	477	478
			479	480	481	482	483	484	485	486
			487	488	489	490	491	492	493	494
			495	496	497	498	499	500	501	502
			503	564	505	506	507	569	598	599
			600	601	602	603	604	605	606	607
			608	609	610	611	612	613	614	615
			616	617	618	619	620	621	622	623
			624	625	626	627	651	652	653	654
			655	656	657	658	659	660	661	662
			663	664	665	666	667	668	669	670
			671	672	673	674	675	676	677	678
			679	680	694	705	706	707	708	709
			710	711	760					
FILTRAH	D		0							
FILTRAU	D		550	635	588	721				
FILTENM	D		434	693	712					
GNDDF	A	00073	-121	122						
GNHPF	A	00074	-122	123						
GNINPUT	A	00070	-118	119	358					
GNLPF	A	00072	-120	121	631	635	684	688		
GNOUT	A	00076	-124	125	763					
GNPRED	6	00075	-123	124	724					
GNOMF	A	00071	-119	120						

	HEROUT	A	00155	-82							
	HPF	A		-650							
	HPFBS	A	00136	-81	82	373	650	651	652	653	654
				655	656	657	658	659	660	661	662
				663	654	655	666	667	668	669	670
				671	672	673	674	675	676	677	678
				679	086	690	691	693	694	696	
	HPFSZ	A	00021	-67	112	373					
	INFUT	D		358	361						
	10	A	0000C	0							
	IDFD	A	00000	0							
	j	A	00009	-55	92	93					
	JMF	1		177	190	327	351	352	353	364	366
				368	370	372	374	376	378	453	509
				557	558	572	577				
	JRF	A	09007	0							
	JSRF	A	00005	0							
	JZ	6	00000	0							
	LDCT	A	00000	0							
	LDMP	0		355	371	373	375	377	379	549	576
				634	687	720	758				
	LEAST	A	003FF	-84	544	550	552	563	573	575	576
				629	635	682	688	715	721		
	LOAD	D		205	210	215	220	226	231	233	238
				240	245	247	252	258	263	269	365
				367	369	413	469	578	580	582	735
				737	739	741	743	745	747	749	
	LOOP	ī		0							
	LP	Å	0000B	0							
	LPF	A	000ED	-597							
	LPFB3	A	00001	-75	76	77	371	579	581	583	597
1111				598	599	600	601	602	603	604	605
				606	697	608	609	610	611	512	613
				614	615	616	617	618	619	620	621
				622	623	624	625	626	627	637	
	LPFSZ	Ė	0001F	-57	76	82	109	371	637	690	691
				696							
	LSTFE	A	00017	-98	99	369	369	553	553	578	578
				580	580	582	582				
	LETFEH	A	0001á	-97	98	367	367	415	415		
	LSTFEL	A	00015	-96	97	365	365	471	471		
	LSTF	A	00014	-95	96	221	221	227	227	234	234
				241	241	248	248	253	253	259	259
				265	265	271	271	277	277		
	MEMEN	A	00004	0							
	MOST	A	003FE	-86	545	546	548	549	554	559	561
				569	571	630	631	633	634	683	684
				686	687	716	717	719	720		
2000	NEG	A	00002	557	572	577					
1000	NEGATE	A	000B7	557	-559						
A STATE	NEWAUTO	A	01000	0							
	NOMEM	A	006FF	0							
1	NOPERT	A	000EB	572	-582						

NORMAL	A	00000	0							
NOTE	A	00048	370	-372						
NOTFE	A	00056	368	-370						
NOTFEH	A	00064	366	-368						
NOTFEL	A	00062	364	-366						
NOTHER	A	0005A	372	-374						
NOTS	A	00070	378	-380						
NOTSPE	A	0006E	376	-378						
NOTSUM	A	0006C	374	-376						
OUTLE	A	00059	352	-355						
OUTFUT	D		170	171	356	380				
F	A	00012	-93	94	205	205	210	210	215	215
			231	231	238	238	245	245	263	263
			269	269	362	362				
P0	A	00000	-91	92	704	705	706	707	708	709
			710	711	736	736	738	738	740	740
			741	742	742	744	744	746	746	748
		*****	748	750	750					
FAD	4.	60000	0	FAC	FAD					
PERT	A	000CA	453	509	-542					
PERTHS	A	00133	-80	81	395					
PERTF	A	000E9	577	-580						
POSTV	A	000DA	558	-562						
PPRT	A	00002	-132	361						
PRED PREDBS	A	00144 00125	388 -77	-704 78	79	377	704	705	706	707
LVERDO	н	00125	708	709	710	711	712	723	724	707
PREDOUT	A	0012E	-78	/ 77	/10	/11	/12	140	124	
PREDSCL	A	0006F	-112	113	114	712				
PREDSZ	A	00005	-58	78	79	/12				
PREL	A	00100	0	, ,	"					
FSH	A	00004	0							
PTHEO	A	00009	-92	93	206	206	211	211	216	216
			220	220	226	226	232	232	233	233
			239	239	240	240	246	246	247	247
			252	252	258	258	264	264	270	270
			735	735	737	737	739	739	741	743
			743	745	745	747	747	749	749	
PUSH	I.		172	292						
QMFB5	A	00000	-73	74	396	414	416	417	418	419
			420	421	422	423	424	425	426	427
			428	429	430	431	432	433	434	435
			436	437	438	439	440	441	442	443
			444	445	446	447	448	449	450	451
			470	472	473	474	475	476	477	478
			479	480	481	482	483	484	485	486
			487	488	489	490	471	492	493	494
			495	496	497	498	499	500	501	502
			503	504	505	506	507			
QMFH	A	00078	225	286	322	-413				
: QMFL	A	000A1	201	257	306	-469				
QMFSZ	A	00024	-56	74	110					
REPLP	D		176	326						

....

......

RET	D		0							
RECT	A	80000	0							
RINGS	A	00000	0							
RPCT	A	00009	0							
S	A	00130	-83	355						
SCRAT	A	00004	567	704	705	706	707	708	709	710
BUETO: V			711							
SHFTCLK	A	00800	0							
SHFTLFT1	D		0							
SHFTLFT2	I	00004	173	-190	100	327				
SOFENT	A	0000A	177 173	174	190 392	567				
SPRT	A	00002	-134	170	170	356	356			
STLR	I	00002	544	548	552	561	571	575	629	633
DILL.	1		682	686	696	715	719	3/3	021	633
STLS	F		360	000	676	110	117			
STMR	0		175	395	396	414	470	545	554	579
Jiiik	•		581	583	630	637	683	690	716	723
			726	762	735	037	000	070	/10	123
STMS	D		206	211	216	221	227	232	234	239
011.0			241	246	248	253	259	264	265	270
			271	277	362	553	566		200	-/-
STORE	T.		415	471	736	738	740	742	744	746
			745	750						
SUB	A	00400	0							
SUMBS	A	00051	-76	375						
TC	A	02000	0							
TEMP	A	00018	-99	566	566	567				
TEST	A	00001	-131	171	171	380	380			
THRESH	A	00076	-125	563						
TRUE	Ĥ	00000	0							
TSL	A	00001	0							
TSK	A	00002	0							
TSTE	À	10000	-141	370						
TSTFE	A	00004	-140	368						
TSTFEH	A	00009	-139	366						
TSTFEL	A	00005	-138	364						
TSTHPF	A	30000	-142	372						
TSTS	A	0000F	-145	378						
TSTSPE	A	0000E	-144	376						
TSTSUM	A	10000	-143	374						
TWB	A	0000F	0	70-						
WAITLF WRITE	f.	00057	-352	353						
	A	00020	0							
+LD	A	00184	0							
.MARK .MULT1	AA	01000	750	567						
.MULTA	A	02044 02244	358	30/						
.MULTAN	A	02644	0							
·MULTAU	A	00244	0							
.MULTL	A	02000	0							
.MULTH	A	02204	0							
.MULTHM	A	02604	0							
*HOLIKE		02007	0							

-NOP	A	00000	0		
.SENEL	A	00005	-135	351	352
.SHIFT	A	00800	173	392	
.SOF	A	00007	190		
STL	A	00039	0		
.STH	A	0002A	0		
.STR	A	00023	0		
.TESTO	A	00008	138		
.TEST1	A	00009	139		
.TEST2	A	0000A	140		
.TEST3	A	0000B	141		
.TEST4	A	00000	142		
.TE575	A	0000D	143		
TEST6	A	0000E	144		
.TEST7	A	0000F	145		

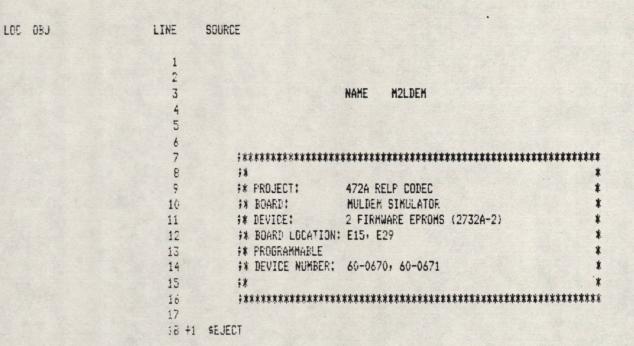
. . .

APPENDIX B
MULDEM SIMULATOR FIRMWARE LISTINGS

APPENDIX B MULDEM SIMULATOR FIRMWARE LISTINGS

B1. <u>2-BIT ADAPTIVE RESIDUAL QUANTIZER</u>

VAX/VHS 8086/8087/8088 MACRO ASSEMBLER V1.0VX ASSEMBLY OF MODULE M2LDEM OBJECT MODULE PLACED IN MOLDEM.OBJ NO INVOCATION LINE CONTROLS



LOC OFF	LINE	SOURCE			
	19				
	20				
	21 22				
	23	********	******	******	*****************
	24	**	****	*****	*******************
	25		ARE SIMU	ATES THE	MULtiplexor/DEMulti- *
	26				STEN. REFER TO MDA *
	27	# DOCUMENT O	0 - 3035	- R01 F0	R DETAIL. *
	29	i*			1
	29	*********** *	*****	******	****************
	30				
	31 32				
	37				
	14	;****************	***		
	35	# SYSTEM CONSTANTS			
	35	*****************			
	37				
	38				
	39				
6020	40	LEN_FE_BUFFER	EQU	32	FACCOMODATE 16 PAIRS OF FEX
0000	41	LEW & BUFFER FRANC	FOU		ACADU CDAVE ADDOVDDATED O DECENOTOR DOCCESORE
8000	42	LEN_P_BUFFER_FRAME NTS	EQU	8	FEACH FRAME ACCOMODATES 8 PREDICTOR COEFFICIE
	43	nio			
0020	44	LAST_P_BUFFER_FRAME	EQU	32	THE BASE OFFSET ADDRESS OF LAST P FRAME
	45				FROM THE BEGINNING OF F BUFFER. THIS VALUE
	46				#MUST BE IN MULTIPLE OF LEN_P_BUFFER_FRAME *
		TYPE P_BUFFER			
	47				THE FRAME DELAY IS EQUAL TO 1 +
	48				(LAST_P_BUFFER_FRAME / 16)
	49				
0001	50 51	FE_HIGH	EQU	1	FTHIS REPRESENTS THE FEH'S TYPE
0000	52	FE_LOW	EQU	The second second	THIS REPRESENTS THE FEL'S TYPE
****	53				THE NEW MEDICAL PRESENTS AND THE PRESENTATION OF THE PRESENTATION
0002	54	NLEV	EQU	2	NUMBER OF QUANTIZER REGIONS
	55				
000A	56	FSVMN	EQU	10	FFSVMN - MINIMUM SCALING FACTOR
	57				
03E9	58 59	FSVMX	EGU	1000	FSVMX - MAXIMUM SCALING FACTOR
000F	60	FSVTH	EQU	15	FFSVTH - THRESHOLD VALUE, IF THE
VVVI	61	134111	200	10	SCALE FACTOR IS LESS THAN FSVTH, A
	62				#MID-TREAD QUANTIZER IS USED INSTEAD
	63				FOF A MID-RISE QUANTIZER, THE MID-TREAD
	64				QUANTIZER USES ZERO AS AN DUTPUT LEVEL
	65				FINSTEAD OF YQ(1).
	66				
0000	67	POSITIVE	EQU	0	POSITIVE SIGN
DOCS	68	NECATTHE	EUI		*NECATTHE CICH
0001	69 70	NEGATIVE	EQU	1	*NEGATIVE SIGN
	71				

LOC OBJ	LINE	SOURCE			
	72 73	HARDWARE DEPENDENT	CONSTA	NTS	
0042	74 75	C_DATA_BUS	EQU	42H	FOR THE 16 BIT CODER DATA BUS
6044	76 77 78 79	C_CONTROL_BUS	EQU	44 H	FOR THE 16 BIT CODER CONTROL BUS
0046	80 81 82 83 84	CODER_BUS_SPVR	EQU	46H	FTHIS IS THE SUPERVISOR PORT(CONTROL PORT) FOR THE CODER BUS(THE CONTROL BUS AND FTHE DATA BUS)
004A	85 86 87 88	D_DATA_BUS	EQU	4AH	FOR THE 16 BIT DECODER DATA BUS
0040	89 90 91 92	D_CONTROL_BUS	EQU	4CH	FTHIS IS THE PORT(C) ADDRESS ON THE 8255'S FFOR THE 14 BIT DECODER CONTROL BUS
064E	93 94 95 96 97 98 99	DECODER_BUS_SPVR	EQU	4EH	FTHIS IS THE SUPERVISOR PORT(CONTROL PORT) FOR THE DECODER BUS(THE CONTROL BUS AND FTHE DATA BUS)
0664	101 102 103	ERRORLLED	EQU	64H	
0066	104 105 106 107	ERROR_BUS_SPVR	EQU	66H	
	108 109	#8259A PIC DEPENDE	ENT CONS	TANTS	
0078 0074	110 111 112	PIC_PORT_0 PIC_PORT_1	EQU EQU	78H 7AH	\$8259A PROGRAMMABLE INTERRUPT CONTROLLER \$CONTROL PORTS.
0013 0008 0003	113 114 115 116 117	ICW1 ICW2 ICW4	EQU EQU	13H 08H 03H	#SINGLE PIC, ICW4 NEEDED #STARTING INTERRUPT VECTOR = 8 #SFNM = 0, AEDI = 1, NON-BUFFERED MODE #SP/EN WILL HAVE INPUT = 1 =>MASTER #UPM = 1 =>8086/88 SYSTEM.
00FE	118 119 120	ENABLE_SOF_MASK	EQU	OFEH	#MASK OUT ALL EXCEPT #SOF INTERRUPT(RO)
00E0	121 122 123 124 125 126	ENABLE_NORMAL_MASK	EQU ENT CONT	0E0H	#MASK OUT UNUSED #INTERRUPT(R7,R6,R5)

8086/8	087/6088	MACRO	ASSEMBLER	MOLDEN
--------	----------	-------	-----------	--------

9898 127 128	
### 130 ### 130 #### 130 #### 130 #### 130 #### 130 #### 130 #### 130 ##### 130 ##### 131 #### 131 #### 131 #### 132 ##### 133 ###### 133 ####### 133 ########	T. All
130	IT NLL
NE PPI	ES TO O
9999 132 C_DATA_BUS_OUT EQU 9999H \$PROGRAM THE CODER DATA BUS TO BE OUTF 133 \$TAKE CONTROL OF THE DATA BUS TO BE OUTF 134 \$PORTS ON THE CODER BUS ARE INPUT. MSI 135 \$AND LSE OF THIS WORD EACH ADDRESSES (136 \$PORTS ON DECODER BUS ARE INPUT. 137 B_DATA_BUS_IN EQU 9898H \$PROGRAM THE DECODER DATA BUS TO BE IN COORDER BUS ARE INPUT. 139 \$PORTS ON DECODER BUS ARE INPUT. 139 \$PORTS ON DECODER BUS ARE INPUT. 140 \$PORTS ON DECODER DATA BUS TO BE OUTF 141 BUS TO BE OUTF 142 \$PORTS ON THE DECODER DATA BUS TO BE OUTF 143 \$PORTS ON THE DECODER BUS ARE INPUT. 144 \$PORTS ON THE DECODER BUS ARE INPUT. 145 \$PORTS ON THE	
### ### ##############################	
PORTS ON THE CODER BUS ARE INPUT. MSS 135 136 9898 137 B_DATA_BUS_IN EQU 9898H PROGRAM THE DECODER DATA BUS TO BE IN 138 139 NE PFI 140 9999H PROGRAM THE DECODER BUS ARE INPUT. 139 NE PFI 140 9999H PROGRAM THE DECODER BUS ARE INPUT. 141 9999H PROGRAM THE DECODER BUS TO BE OUT 142 \$TAKE CONTROL OF THE DATA BUS TO BE OUT 143 \$PORTS ON THE DECODER BUS ARE INPUT. 144 \$PORTS ON THE DECODER BUS ARE INPUT. 145 9092 146 ERROR_LED_ON EQU 0092H PROGRAM THE ERROR BUS TO BE:	UT (IE.
### ##################################	
9B9B 137 BLDATALBUSLIN EQU 9B9BH \$PROGRAM THE DECODER DATA BUS TO BE IN L 138 \$OTHER PORTS ON DECODER BUS ARE INPUT. 139 \$MSB AND LSB OF THIS WORD EACH ADDRESS NE PPI 140 \$9999 141 DLDATALBUSLOUT EQU 9999H \$PROGRAM THE DECODER DATA BUS TO BE OUT 143 \$PORTS ON THE DECODER BUS ARE INPUT. 1144 \$AND LSB OF THIS WORD EACH ADDRESSES (145) \$AND LSB OF THIS WORD EACH ADDRESSES (145) \$PROGRAM THE ERROR BUS TO BE:	
9898 137 B_DATA_BUS_IN EQU 9898H \$PROGRAM THE DECODER DATA BUS TO BE IN L 138 \$OTHER PORTS ON DECODER BUS ARE INPUT. 139 \$MSB AND LSB OF THIS WORD EACH ADDRESS NE PPI 140 \$1999 141 D_DATA_BUS_CUT EQU 9999H \$PROGRAM THE DECODER DATA BUS TO BE OUT \$143 \$PORTS ON THE DECODER BUS ARE INPUT. 1 \$PORTS ON THE DECODER BUS ARE INPUT. 1 \$PORTS ON THE DECODER BUS ARE INPUT. 1 \$PORTS OF THIS WORD EACH ADDRESSES 0 145 \$PROGRAM THE ERROR BUS TO BE:	INE PFI.
SPREAD LSB OF THIS WORD EACH ADDRESS OF THIS WORD EACH ADDRESSES OF THIS	ID. 11
139 NE PPI 140 9999 141 D_DATA_BUS_OUT EQU 9999H PROGRAM THE DECODER DATA BUS TO BE OUT 142 143 PORTS ON THE DECODER BUS ARE INPUT. N 144 PAND LSB OF THIS WORD EACH ADDRESSES OF THIS WORD EACH ADDRE	IPUT, AL
NE PPI 140 9999 141 D_DATA_BUS_OUT EQU 9999H \$PROGRAM THE DECODER DATA BUS TO BE OUT 142 \$TAKE CONTROL OF THE DATA BUS), ALL OT 143 \$PORTS ON THE DECODER BUS ARE INPUT. N 144 \$AND LSB OF THIS WORD EACH ADDRESSES (145 0092 146 ERROR_LED_ON EQU 0092H \$PROGRAM THE ERROR BUS TO BE:	
9999 141 D_DATA_BUS_OUT EQU 9999H \$PROGRAM THE DECODER DATA BUS TO BE OUT 142 \$TAKE CONTROL OF THE DATA BUS), ALL OF 143 \$PORTS ON THE DECODER BUS ARE INPUT. IN 144 \$AND LSB OF THIS WORD EACH ADDRESSES OF 145 ERROR_LED_ON EQU 0092H \$PROGRAM THE ERROR BUS TO BE:	SES TO O
9999 141 D_DATA_BUS_CUT EQU 9999H \$PROGRAM THE DECODER DATA BUS TO BE OUT 142 \$TAKE CONTROL OF THE DATA BUS), ALL DI 143 \$PORTS ON THE DECODER BUS ARE INPUT. II 144 \$AND LSB OF THIS WORD EACH ADDRESSES (145 ERROR_LED_ON EQU 0092H \$PROGRAM THE ERROR BUS TO BE:	
\$\frac{142}{143}\$\$ \$\frac{1}{2}\$\$ \$\	
143 #PORTS ON THE DECODER BUS ARE INPUT. 1 144 #AND LSB OF THIS WORD EACH ADDRESSES (145 #REFORLED ON EQU 0092H #PROGRAM THE ERROR BUS TO BE:	TPUT(IE
144 #AND LSB OF THIS WORD EACH ADDRESSES (145 #AND LSB OF THIS WORD EACH ADDRESSES (145 #AND LSB OF THIS WORD EACH ADDRESSES (145 #AND LSB OF THIS WORD EACH ADDRESSES (145 #AND LSB OF THIS WORD EACH ADDRESSES (145 #AND LSB OF THIS WORD EACH ADDRESSES (145 #AND LSB OF THIS WORD EACH ADDRESSES (145 #AND LSB OF THIS WORD EACH ADDRESSES (145 #AND LSB OF THIS WORD EACH ADDRESSES (145 #AND LSB OF THIS WORD EACH ADDRESSES (145 #AND LSB OF THIS WORD EACH ADDRESSES (146 #AND LSB OF THIS WORD EACH ADDRESSES (147 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD EACH ADDRESSES (148 #AND LSB OF THIS WORD	HER
145 0092 146 ERRORLLEDLON EQU 0092H \$PROGRAM THE ERROR BUS TO BE:	ISB
0092 145 ERROR_LED_ON EQU 0092H \$PROGRAM THE ERROR BUS TO BE:	INE PPI.
147 PORT A - TARRIT 3 NOT HIGH HERE	
그 그 그 그 그 그는 그 그는 그 그는 그 그는 그는 그는 그는 그는	
148 FORT B - INPUT 3 NOT USED HERE	
149 FORT C - OUTPUT) TO ERROR_LED	
00FF 151 ERRORLLED_OFF EQU 009BH #PROGRAM THE ERROR BUS TO BE:	
152 FORT A - INPUT) NOT USED HERE	
153 FORT B - INPUT > NOT USED HERE	
154 FORT C - INPUT > TO ERROR_LED	
155 \$THE LED DISPLAY IN THIS STATE WILL	
156 #BE *FF*	
158	
159 #CONTROL BUS DEPENDENT CONSTANTS	
160	
161 DECODER BUS	
162 ;	
163	
0004 164 SPEN EQU 0004H \$MASK FOR SPEN SIGNAL(LOW TRUE) 0010 165 SFEEN EQU 0010H \$MASK FOR SFEEN SIGNAL(LOW TRUE)	
0010 165 SFEEN EQU 0010H #MASK FOR SFEEN SIGNAL(LOW TRUE)	
167	
16E CODER BUS	
169 ;	
170	
0800 171 CFECLK EQU 0800H \$MASK FOR CFECLK SIGNAL(HIGH TRUE)	
0020 172 CPCLK EQU 0020H #MASK FOR CPCLK SIGNAL (HIGH TRUE)	
173	
174	
175	
176 FMDA IN HOUSE MONITOR ENTRY POINT	
177	

181 +1 \$EJECT

LOC DEJ LINE SOURCE

MONT. 86_IF 0010 178 FE9F MONT_86_CS 179 160

EQU 0010H #OFFSET ADDRESS EQU OFE9FH ISEGMENT ADDRESS

11	47:02	10/07/82	PAGE	
11	+7/ +V-	10/0//02	INDL	

LOC	USO LS0	LINE	SOURCE
		182	
		183	
		184	INTERRUPT_VECTOR SEGMENT WORD AT OH
		185	
		186	
		187	
		188	; *************************************
		189	* INTERRUFT VECTOR TABLE *
		190	; ************************************
		191	AND STREET, AND ST
		192	FINTEL RESERVES INT 5 TO 32 FOR INTERNAL USES, CURRENT IMPLEMENTATION
		193	; VIOLATES THIS RESTRICTION.
		194	
		195	#8086 PREDIFINED INTERRUPTS: (INT 0 TO 4)
		198 197	FOUND FRESHFIRED IN. EMBORIO. (IN) V 10 47
0000	11	198	DIVIDE_INT_IP DW 1 DUP(?)
	1777		
)		
0002		199	DIVIDE_INT_CS DW 1 DUF(?)
	7777		
)		
0004	(1	200	SINGLE_STEP_IP DW 1 DUP(?)
	1111		
)		
0006		201	SINGLE_STEF_CS DW 1 DUP(?)
-	1175		
)	5.05	NMI_IF DW 1 DUF(?)
0008		202	NMI_IF DW 1 DUF(?)
	???? }		
000A		203	NHILCS DW 1 DUP(?)
	7777		
)		
3000	(1	204	BREAKPOINT_IP DW 1 DUP(?)
	????		
)		
000E	(1	205	BREAKPOINTLCS DW 1 DUP(?)
	????		
)		
0010		206	OVERFLOW_IP DW 1 DUP(?)
7.4	????		
)	207	OVERFLOW_CS DW 1 DUP(?)
0012		207	OVERFLUW-US I'W I DOI V: /
	????		
		208	
		209	
12		210	#MULDEM APPLICATION INTERRUPTS: (INT 8 TO 15)
		211	
0020		212	ORG 20H
0020	(1	213	INT_B_IP DW 1 DUP(?)
	7777		
7)		THE A SA BULL A DUBLAN
0022	(1	214	INT_B_CS DW 1 DUP(?)

LOC	OBJ	LINE	SOURCE		
	????)				
0024		215	INT_9_IP	DW	1 DUP(?)
0026	(I	216	INT_9_CS	DW	1 DUP(?)
0028	7777	217	INT_10_IP	DW	1 DUP(?)
002A) (1 ????	218	INT_10_CS	DW	1 DUP(?)
0020	(1 1)11	219	INT_11_IP	D₩	1 DUP(?)
002E	(1 ????	220	INT_11_CS	DW	1 DUP(?)
0030	7171	221	INT_12_IF	DW	1 DUP(?)
0032	1111	222	INT_12_CS	DM	1 DUP(?)
0034	????	223	INT_13_IP	₽₩	1 DUP(?)
0036	7777	224	INT_13_CS	DW	1 DUP(?)
0038	9993	225	INT_14_IP	DW	1 DUF(?)
003A	7777	226	INT_14_CS	DM	1 DUP(?)
0030) (1 7777	227	INT_15_IP	DW	1 DUF(?)
003E	????	228	INT_15_CS	DW	1 DUP(?)
		229 230 231 232 233	INTERRUPT_VECTOR		ENDS
		234 235 +1	\$EJECT		

100	OBJ.	LINE	SOURCE		
		236 237			
		238 239	DATA SEGMENT		
		240	THIS SEGMENT WILL I	RESIDE I	N RAM
0000	(24	241 242	PREDICTOR_BUFFER UP(?)	DW	(((LAST_P_BUFFER_FRAME/16)+1) * LEN_P_BUFFER_FRAME) D
	????)				
0030	(32 ????	243	FE_BUFFER	DW	(LEN_FE_BUFFER) DUP(?)
)	244			
0070	(1 1771)	245	LED_DISPLAY_VALUE	DW	1 DUP(?)
		246			
	(1 17:17	247	ABS_XIN	I.M	1 DUF(?)
0074	????	248	XGUT	DW	1 DUP(?)
)	249			
0076	(1	250	FSV	DW	1 DUP(?)
	????				
0078	(i ????	251	FSVL	DW	1 DUP(?)
007A	7977	253	FSVH	DW	1 DUF(?)
)	253			
	7777	254	I	DW	1 DUF(?)
007E) (1 ????	255	IL	DW	1 DUF(?)
)	256	L2	DW	1 DUF(?)
	???? \				
)	257			
0082	(1 ????	258	FE_SERVICE_COUNTER	DM	1 DUF(?)
		259			
0084	(1 ????)	260	FE_SERVICE_PTR	DM	1 DUP(?)
		261			
0086	(1 ????	262	FE_TYPE	DW	1 DUP(?)

L00	OBJ	LINE	SOURCE			
)					
		263				
0088	(1	264	SI	GN_FLAG	DW	1 DUP(?)
	7771					
)					
		265				
		266				
		267	DATA	ENDS		
		268				
		269				
		270				
		271				
	•	272				
		273				
		274				
		375				
		27 6				
		277	STACK		SEGMENT	
		278			CLONEIN	
		279		*******	*****	********
		280				USED BY THE 8086 TO *
		281				IN INTERRUPT ROUTINES*
		282				*********
0000	(60	283			DW	60 DUF (?)
	????					
)					
0078		284	TOS		LABEL	WORD
		285				
		286	STACK		ENDS	
		287				
		268 +1	\$EJECT			

LOC OBJ

LINE SOURCE #SYSTEM MACROS: j-----THE MACROS ARE NOT LISTED IN THE ASSEMBLER GENERATED LISTING, REFER FTO THE SOURCE FILE FOR MACRO CONTENTS. . 301 35.5 315 +1 \$E.H. 1

LOC DBJ

LINE	SOURCE	
316		
317		
318		
319	;***	********
320	;*	
321	12	REGISTER USAGE IN M2LDEM SIMULATOR SYSTEM: *
322	;*	
323	;***	***************
324		
325		
326	DEDICATED US	AGE FOR ALL PARTS OF THE SYSTEM AT ALL TIME:
327		
328	iSI	- FE_RUFFER INPUT POINTER
329	₹BX	- FE_BUFFER OUTPUT POINTER ETHIS REGISTER IS ALSO USED UNDER
330	;	NON-INTERRUPT DRIVEN QUANTIZATION PROCESS, HOWEVER, THE
331	;	ORIGINAL REGISTER IS PRESERVED
332	#BF	- PREDICTOR_BUFFER FRAME POINTER
333	;DI	- PREDICTOR_BUFFER OFFSET POINTER
334		
335		
336	#CS	- CODE SEGMENT
337	#DS	- DATA SEGMENT
338	;ES	- DATA SEGMENT OR INTERRUPT_VECTOR SEGMENT
339	FSS/8	SF - STACK OPERATION
340		
341		
342	FUNASSIGNED F	EGISTERS:
343		
344	\$THES	E REGISTERS DO NOT CARRY DEDICATED FUNCTIONS IN THE M2LDEM
345	FSIML	LATOR:
346		
347	fAX,	CX. DX
348		
349 +1	\$EJECT	

LOC OBJ	LINE SOURCE		
	350		
	351		
	352	PUBLIC	START_ADDR .
	353		
	354		
	355 CODE		SEGMENT
	356		
	357	ASSUME	CS:CODE, DS:DATA, SS:STACK, ES:INTERRUPT_VECTOR
	358		
	359		
	360		
	361	***********	*****************
	362	* STATIC	VARIABLES *
	363	**********	***********
	364		
	365		
	3óf	;YQ -	
	367	;	(IN INCREASING ORDER)
	365	,	VALUE SCALED BY ** IN ** REPRESENTATION
0000 0000	369	YG	DW 0, 8192, 24576
0002 0020			
0004 0040			
	370		
	371	;XQ -	
	372	!	(IN INCREASING ORDER)
0006 0000	373	XQ ;	VALUE SCALED BY ** IN ** REPRESENTATION
0008 0040	374	XK	DW 0, 16384
0000 0010	373		
	376	*OMLT	- ARRAY OF NLEV, QUANTIZER MULTIPLIERS
	377	;	VALUE SCALED BY ** IN ** REPRESENTATION
000A 0000	378	QMLT	DW 0, 27853, 62259
000E CI4E	3,0	KIIL I	DW 01 2/0331 02231
000E 33F3			
2002 0010	379		
	390		
	381 +1 \$EJECT		
	001 11 *10101		

```
LINE
                              SOURCE
LOC DEJ
                       382
                       383
                                 *******************
                       384
                                 *
                                                  MAIN
                                                            PROGRAM
                       385
                                 386
                       387
                       388
                                     ******************************
                       389
                       390
                                             REGISTER VALUES ARE NOT PRESERVED
                       391
                                             IN ALL PROCEDURES. THEY SHOULD BE
                       392
                                                 SAVED BEFORE ENTERING.
                       393
                       394
                                     ***************************
                       395
                       396
0010 FA
                       397
                              START_ADDR:
                                                                 IDISABLE EXTERNAL INTERRUPT
                                            CLI
0011 B8----
                       398
                                            MOV AX, DATA
                                                                 CANNOT MOVE IMMED. VALUE TO
0014 SED8
                       300
                                            VCM
                                                   DS, AX
                                                                 SEGMENT REGISTER.
                       400
0018 BED0
                       401
                                            MOV
                                                   ES, AX
                                                                 JES AND DS ARE REFERING TO SAME
                       402
                                                                 *SEGMENT
                       403
                       404
                                            FRESET THE LED_DISPLAY_VALUE
                       405
                                                   LED_DISPLAY_VALUE, 0
0018 0706700000000
                                            MOV
                       405
001E B89200
                       407
                                            VOM
                                                   AX, ERROR_LED_DN
0021 E666
                       408
                                            OUT
                                                   ERROR_BUS_SPVR . AL
                       409
                                            #GUTPUT ERROR DISPLAY VALUE
                       410
                       411
0023 B60000
                       412
                                            VOH
                                                   AX, 0
                       413
                                                   ERROR_LED, AL
0026 E664
                                            DUT
                       414
                       415
                                     **********************
                                     * INITIALIZE THE INTERRUPT VECTOR TABLE
                       416
                       417
                       418
                                     * THIS SUBSYSTEM USES FOLLOWING SIGNALS FROM *
                       419
                                     * THE SYSTEM BUS!
                       420
                                          - START OF FRAME (SOF_)
                       421
                       422
                                          - FE ENABLE (SFEEN_)
                       423
                                          - P ENABLE (SPEN_)
                       424
                                          - FE CLOCK (CFECLK)
                       425
                                          - P CLOCK (CPCLK)
                       426
                                     ****************
                       427
                       428
                       429
                       430
0028 B80000
                       431
                              ERROR_ENTRY:
                                            MOV
                                                   AX, INTERRUPT_VECTOR
                                                                        JUSE EXTRA SEGMENT TO ADDRESS
                                                   ES, AX
0025 BECO
                       432
                                            MOV
                                                                        THE INTERRUPT VECTOR TABLE
                       433
                       434
002D 26C70608001000
                       435
                                            MOV
                                                   NMI_IP, MONT_86_IP
                                                   NMI_CS, MONT_86_CS
0034 26C7060A009FFE
                       436
                                            MOV
```

LOC	OBJ		LINE S	OURCE		
			437			
0038	B80901		438		MOV	AX, OFFSET EXCEPTION_INT
003E	BB90	R	439		YOM	BX, SEG EXCEPTION_INT
0042	26A30000		440		MOV	DIVIDE_INT_IP, AX
0046	26891E0200		441		HOV	DIVIDE_INT_CS; BX
004B	26A31000		442		MOV	OVERFLOW_IF, AX
004F	26891E1200		443		VCM	OVERFLOW_CS+ BX
			444			
			445			
0054	26070520000000		446		KOV	INT_8_IP, OFFSET SOF_INTERRUFT
005E	2607062200	R	447		VOM	INT_8_CS; SEG SOF_INTERRUPT
			448			
0062	26070624005D00		449		MOV	INT_9_IP, OFFSET PREDICTOR_OUTPUT_INTERRUPT
0069	2607062600	R	450		YOM	INT_9_CS, SEG PREDICTOR_OUTPUT_INTERRUPT
			431			
0070	26070628008E00		452		VOH	INT_10_IP, OFFSET FE_DUTPUT_INTERRUPT
0077	2607062A00	F	453		MOV	INT_10_CS; SEG FE_OUTPUT_INTERRUPT
			454			
007E	2607062000EA00		455		MOV	INT_11_IP, OFFSET PREDICTOR_INPUT_INTERRUPT
0085	2607062E00	F	458		MOV	INT_11_CS, SEG PREDICTOR_INPUT_INTERRUPT
			457			
0080	250,7063000BF00		458		VCH	INT_12_IP, OFFSET FE_INPUT_INTERRUFT
0093	2607063200	R	459		MOV	INT_12_CS+ SEG FE_INPUT_INTERRUPT
			460 +1 \$	EJECT		

LGC 06J		LINE	SOURCE					
		461						
		462						
		463		ASSUME ES	:DATA			
		464		NOONE EU	· Duin			
		465						
		466						
		457		FINITIAL TZ	7F 8086	PRI	CESSOR ENVI	RONMENT
		468						
009A BB	R	469		HC	OV	AX,	DATA	CANNOT HOVE INHED. VALUE TO
007D 8ED8		470					AX	SEGMENT REGISTER.
		471						
009F 8EC0		472		MC	GV	ES,	AX	JES AND DS ARE REFERING TO SAME
		473						SEGMENT
00A1 B8	R	474		MC	OV	AX,	STACK	
00A4 8ED0		475				55;		
00A6 BC7500		476				SP,	OFFSET TOS	
		477						
		478						
		479		FINITIALIZ	ZE ALL	SYS	TEM HARDWARE	
		480						
		481						
		482						
		463		; *	****	***	*****	
		484		; *	P	I	C \$	
		485		; *	*****	***	******	
		486						
		487						
		453		#CAUTION:	AUTON	ATI	C EDI IS USE	HERE. REFER TO INTEL APPLICATION
		489		i	NOTE	AF-	59 'USING TH	HE 8259A PROGRAMMABLE INTERRUPT
		490		;	CONTE	CLL	ER', UNDER H	HEADING "AUTOMATIC EOI MODE"
		491						
		492						
00A9 B013		493		M	OV			
00AB E678		494		0	UT	PIC	_PORT_O, AL	
		495						
OGAD BA7A00		495			OV		PIC_PORT_1	
00B0 B008		497			VOI		ICW2	
00B2 EE		496		Oi	UT	DX,	AL	
		499						
		500		j.	ICM 3	IS N	OT NEEDED FO	OR CURRENT HARDWARE CONFIGURATION
		501						
00B3 B003		502			OV		ICW4	
OOBS EE		503		0	IUT	DX,	AL	
		504						
OOB6 BOFE		505			OV		ENABLE_SOF.	_MASK
OOBS EE		506		0	TUI	DX,	AL	
		507						
		508			****			
		509					******	
		510				P		
		511		; *	*****	***	******	
		512						
		513		********	V ALL	0007	ADE DOCODAS	WHEN TO BE THEFIT DODTE THE MODE A
		514						MMED TO BE INPUT PORTS IN MODE O ICH WILL BE OUTPUT ALL THE TIME
		515		PEACET I	HE ERK	UK L	ED FUNIT WH	TOU MITT DE OOILO! MET LUE LINE

LOS	OBJ	LINE	SOURCE		
		516			
00B9	B87B9B	517		YON	AX, C_DATA_BUS_IN
OOBC	E746	518		DUT	CODER_BUS_SPVR, AX
		519			
		520			
OOBE	B89B9R	521		MOV	AX, D_DATA_BUS_IN
0001	E74E	522		OUT	DECODER_BUS_SPVR, AX
		523 +1	\$EJECT		

LGC DBJ	LI	NE SOURCE				
	5	24				
		25	******	*****	************	11
	5	26	# IN	ITIALIZE	APPLICATION PROGRAM ENVIRONMENT	
		27			***************	
		28				
		29				
		30				
		31				
		32				
		33		; FILL	BUFFER AREAS WITH ZEROS	
		34			POTTER TRACTOR WITH ELROO	
0003 38000		35		HOV	AX, 0	FILL VALUE
0006 B918J		36			CX: LENGTH PREDICTOR_BUFFER	FITERATION COUNT
0007 RF000		37		HOV	DI. OFFSET PREDICTOR_BUFFER	VIIENATION COUNT
0007 10 000		38		1104	DIT OFFSET FREDICTOR_BOFFER	
		39		tre er	TS TO THE SEGMENT BASE OF DATA S	SEGMENT
		40		123 32	15 TO THE SECTION DASE OF DATE S	SEONERI
6000 53			DED	етпе	PREDICTOR_BUFFER	
0000 F3	,	41	REP	STOS	PREDICTOR_BUFFER	
OUGE AP	-	***				
		42				
3555 BUSIN		43		ND.	OV LEVOTU DE BUECES	ATTERATION COUNT
000E B9200		44		MOV	CX. LENGTH FE_BUFFER	FITERATION COUNT
00D1 BF300		45		MOV	DI: OFFSET FE_BUFFER	
		46				
00D4 F3	5	47	REP	STOS	FE_BUFFER	
00D5 AB						
		48				
		49				
		50			SERVICE COUNTER	
00D6 C706B		51		MOV	FE_SERVICE_COUNTER, 0	
		52				
		53			ALIZE FE TYPE TO LOW	
00DE E7068		54		MOV	FE_TYPE, FE_LOW	
		55				
		56				
		57		FINITI	ALIZE BUFFER POINTERS TO APPROPR	RIATE VALUES
		58				
		59			CTOR BUFFER . BP IS THE FRAME PO	
		60		DI IS	THE OFFSET POINTER WITHIN A FRE	AME.
		61				
00E2 BD200) 5	62		MOV	BP, LAST_P_BUFFER_FRAME	ON FIRST SOF INTERRUPT AFTER
	5	163				FOWER UP, SOF_INTERRUFT ROUT
		INE				
	5	64				#WILL SET BP = 0, DI = 0
	5	65				
00E5 BF100	5	66		HOV	DI, LEN_P_BUFFER_FRAME * (TYPE	PREDICTOR_BUFFER)
	5	67				
	5	48		FE BU	FFER INITIALIZATION	
		69				
00E8 BE100		70		MOV	SI, 16	POINTS TO FEL(4)
00EB BB000		71		MOV	BX, 0	
		72				
		73		FOLIANT	IZATION SERVICE	
		74		- acriii I		
00EE C7068		75		MOV	FE_SERVICE_PTR, 16	
				1101	I LLUCKTIOLLI IN 10	

586 +1 \$EJECT

START OF FRAME INTERRUPT

LOC OEU	LINE	SOURCE		
	587			
	588			
	589	;********	******	******
	590	ix		
	591	# START	OF APPLICATION PROGR	RAM *
	592	;*		1
	593	* AFTER POWE	R UP, THE FOLLOWING APPLIC	CATION *
	594		LL NOT START EXECUTION UN	
	595		INTERRUPT HAS BEEN SERVICE	
	596		*******	
	597			
	598			
0102	597	LAREL_WAIT_FOR_NEW_FE		
	600			
	601	\$L00F	AROUND UNTIL THERE IS A I	NEW FE TO SERVICE
	602			
0102 39368400	603	CMF	FE_SERVICE_PTR, SI	
0106 74FA	604	JE	LABEL_WAIT_FOR_NEW_FE	
	605			
	606	· FTWO	POINTERS ARE NOT EQUAL, PE	DSSIBLE NEW FE'S
	607			
	609			
	609	\$CHEC	K SERVICE COUNTER	
0108 833E826000	610	CMF	FE_SERVICE_COUNTER: 0	
0100 7470	611	· JE	LABEL_PTR_ERROR	THE SERVICE FTR AND
	612			INPUT POINTER(SI) ARE NOT
	613			FEQUAL, SERVICE COUNTER = 0
	614			#SYNC ERROR
	615			
	616			
	617	; SERV	ICE COUNTER >=1 , NORMAL	CONDITION
	618			
010F 833E820008	619	CMF	FE_SERVICE_COUNTER, 8	
0114 7375	620	JAE	LABEL_PTR_ERROR	THERE ARE MORE THAN 4 PAIRS
	621			FOF FEX TO BE SERVICED. CPU
	622			FIS RUNNING TOO SLOW
	623			
	624		SERVICE COUNTER < 8, NOR	MAL CONDITION.
	625	; BEC	SIN SERVICE THE FE'S	
	626			
	627			
	628	AF IFOT		
	629 +1	\$EJECT		

LOC	OEJ	LINE S	OURCE		
		630			
0116			ABEL_SERVICE_FE		
		632		HOVE TH	IE NEW FEX INTO AX AND CORRESPONDING FSV TO CX
		633			
		634			USE THE BX REGISTER TO ACCESS THE FE_BUFFER
0116	8B0E8400	635		MOV	CX, FE_SERVICE_PTR
		636			
0116	FA	637		CLI	
0118	8708	638		XCHG	CX, BX
011F	8B4736	639		VOM	AX; FE_BUFFER[BX]
0120	87CB	640		XCHG	CX, BX
0122	Fh	641		STI	
		642			
0123	833F850000	643		CMP	FE_TYPE, FE_LOW
0128	7520	544		JNE	LAPEL_FE_HIGH
		645			
		646			
		647			
		648			
		649		FIT IS F	FEL TO BE PROCESSED
0124	8E0E7800	650			CX, FSVL
	E86D00	651			APCMQ
		652			
		653		FRESULT	FSV IN DX, XOUT IN CX
		654			
0131	89147800	655		MOV	FSVL, DX SAVE THE NEW FSVL
		656		STORE '	THE NEW XOUT INTO THE FE_BUFFER
0135	A18400	657		MOV	AX, FE_SERVICE_PTR
	FA	658		CLI	
0139		559			AX, BX
		660			
013A	894F30	661		HOV	FE_BUFFER[BX], CX
	FF0E8200	662		DEC	FE_SERVICE_COUNTER
		663			
0141	93	664		XCHG	AX, BX
0142		665		STI	
		666			
		657		*UPDATE	THE FE TYPE TO BE SERVICED NEXT
0143	C70626000100	668		MOV	FE_TYPE, FE_HIGH
		669			
		670			
		671		#INCREM	ENT THE SERVICE POINTER, AX CONTAINS FE_SERVICE_PTR
0149	050200	672		ADD	AX, TYPE FE_BUFFER
	3D4000	673		CMP	AX, LEN_FE_BUFFER * TYPE FE_BUFFER FEND OF BUFFER?
	7331	674		JAE	LABEL_WRAP_AROUND
	A38400	675		HOV	FE_SERVICE_PTR + AX
	EBAC	676		JMP	LABEL_WAIT_FOR_NEW_FE
V1J4	LUNC	677		Jili	ETIPELSWITT STORESTED
		678 +1 \$	FIFCT		
		0/0 11	LULUI		

LOC OBJ	LINE	SOURCE		
	679			
	680			
0156	681	LAREL_FE_HIGH:		H IS TO BE PROCESSED.
0156 EB0E7A00	682			CX, FSVH
015A E84100	683		CALL	APCHQ
	684			
	685		FRESULT	FSV IN DX, XOUT IN CX
	680			
015D 89167A60	687		VOM	FSVH, DX ;SAVE THE NEW FSVH
	688			
	689			THE NEW XOUT INTO THE FE_BUFFER
. 0161 A16400	690		MOV	AX, FE_SERVICE_PTR
	591			
0164 FA	692		CLI	
0165 93	693		XCHG	AX, BX
	694			
0166 894F30	695			FE_BUFFER[BX], CX
0169 FF0E6I00	696		DEC	FE_SERVICE_COUNTER
	697			
016D 93	698		XCHG	AX, BX
016E FB	699		STI	
	700			
	701			THE FE TYPE TO BE SERVICED NEXT
016F C70686000000	702		MOV	FE_TYPE, FE_LOW
	703			
	704			
	705			ENT THE SERVICE POINTER
0175 050200	706		ADI	AX, TYPE FE_BUFFER
0178 354000	707			AX: LEN_FE_BUFFER * TYPE FE_BUFFER FEND OF BUFFER?
017F 7305	708			LABEL_WRAF_AROUND
017D A38400	709		MOV	FE_SERVICE_FTR, AX
A+DA FDO:	710		IMP	LADEL WALL FOR MEN CE
0180 EB80	711 712		JMP	LABEL_WAIT_FOR_NEW_FE
	713			
0400		LADEL HOAD ADD	DIT. +	
0182	714	LABEL_WRAP_AROL		E POINTED HAS POINTING TO THE LAST ELEMENT IN SE
	715 716			E POINTER WAS POINTING TO THE LAST ELEMENT IN FE
	717		THANAL	HOVE TO THE FIRST ELEMENT.
0192 C7049400000	717		MOII	FE_SERVICE_PIR, 0
0182 C70684000000 0188 E977FF	715		MOV JMF	
V100 E7//FF	720		Jnr	LABEL_WAIT_FOR_NEW_FE
		\$EJECT		
	721 11	*EJEUI		

LOC OBJ	LINE	SOURCE
	722	
0181	723	LABEL_PTR_ERROR:
	724	
	725	;**************************************
	726	* THIS PART OF CODE HANDLES THE SYNCHRONIZATION ERROR *
	727	* BETWEEN THE QUANTIZATION PROCESS AND THE VARIOUS
	728	* INTERRUPT DRIVEN I/O PROCESSES. *
	729	;*
	730	# AN ERROR CONDITION MAY BE ONE OR MORE OF THE FOLLOWINGS: #
	731	ii i
	732	<pre>i* (1) BOTH INPUT(REGISTER SI - DEDICATED) AND SERVICE POINTER *</pre>
	733	* ARE POINTING TO DIFFERENT ELEMENT IN THE FE_BUFFER *
	734	* AND THE SERVICE COUNTER HAS A VALUE OF ZERO. *
	735	* (2) THE SERVICE COUNTER HAS A VALUE HIGHER THAN 8. THIS *
	736	# MEANS THAT FOUR PAIRS OF FEX OR MORE ARE NOT QUANTIZED *
	737	# YET.
	738	* THE MULDEM SIMULATOR PROCESS WILL CONTINUE ONCE THE *
	739	** ABOVE ERROR(S) OCCURS. THE FOLLOWING ROUTINE WILL CAUSE THE *
	740	** MOST SIGNIFICANT ERRORLED TO INCREMENT BY 1.
	741	;*************************************
	742	
	743	
0188 FA	744	CLI DISABLE EXTERNAL INTERRUFT
018C A17000	745	MOV AX,LED_DISPLAY_VALUE
018F 80C410	746	ADD AH:10H
0192 7207	747	JC LABEL_PTR_EXT
0194 A37000	748	HOV LED_DISPLAY_VALUE, AX
0197 0404	749	OR AL•AH
0199 E664	750	DUT ERROR_LED,AL
019B E98AFE	751	LABEL_PTR_EXT: JMF ERROR_ENTRY
	752	
	753	;*************************************
	754	# END OF MAIN PROGRAM #
	755	;*************************************
	756	
	737	
	758	
	759 ÷	1 \$EJECT

019E

LINE	SOURCE
760	
761	
762	APCHQ PROC NEAR
763	
764	
765	;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
766	* PURPOSE:
767	* THIS ROUTINE QUANTIZES A SAMPLE USING AN ADAPTVE QUANTIZER.
768	ix
769	*
770	;* DESCRIPTION:
771	# THE INPUT SAMPLE IS QUANTIZED, USING THE GIVEN SCALING FACTOR
772	FOR THE QUANTIZER. THE SCALING FACTOR IS UPDATED ON RETURN.
773	**
774	† *
775	* PARAMETERS:
776	* INPUT :
777	* XIN - INPUT SAMPLE(PASSED IN AX REGISTER)
776	* FSV - SCALING FACTOR FOR THE QUANTIZER. THIS VALUE IS
779	* UPDATED ON OUTPUT. (INPUTED IN CX REGISTER)
780	ix and a second
781	;* OUTPUT:
782	* XOUT - OUTPUT QUANTIZED SAMPLE (PASSED IN CX REGISTER)
783	## FSV - NEW SCALING FACTOR FOR THE QUANTIZER (OUTPUT
784	* IN DX REGISTER)
785	; *
786	** PRE-SPECIFIED:
787	** NLEV - NUMBER OF POSITIVE QUANTIZER LEVELS. THE QUANTIZE
788	is assumed to be symmetric about zero. THE TOTAL
789	** NUMBER OF QUANTIZER LEVELS IS 2*NLEV.
790	IX
771	** ROUTINES REQUIRED:
792	* IQUANTZ - QUANTIZE A POINT
793	;*************************************
774	
795 +1	\$EJECT

SAVE FOR FUTURE REFENRECE

01BB 890E8000

824

825

826 +1 \$EJECT

LOC	OBJ	LINE	SOURCE			
		796				
		797				
019E	870E7600	798		VON	FSV, CX	FSV IS PASSED TO THIS PROCEDURE FROM
		799				FREGISTER CX. SAVE FOR FUTURE REFEREN
			CE			
		800				FTHIS VALUE IS ALSO USED IN PROCEDURE
			IQUANTZ			
		801				
		862		XIN IS	PASSED TO THIS PROCEDU	URE IN REGISTER AX.
		803				
01A2	C70685000000	804		NOV	SIGN_FLAG, POSITIVE	FINITIALIZE THE SIGN FLAG
		805				
01A8	APFFFF	806		TEST	AX, OFFFFH	FGET SIGN
	7908	807		JNS	LABEL_1	FIT IS A POSITIVE NUMBER
		808				
		809		XIN I	S A NEGATIVE NUMBER, GET	T ABSOLUTE VALUE
		810				
OIAD	F7D8	811		NEG	AX	
01AF	C70688000100	812		MOV	SIGN_FLAG, NEGATIVE	#STORE STATE OF SIGN
		813				
01B5	A37200	314	LAREL_1:	VOM	ABS_XIN, AX	SAVE THE ABSOLUTE XIN FOR
		615				JUSE IN IQUANTZ PROCEDURE
		816				
		817				
		818		FREGIS	TER AX, CX AND DX ARE NO	OT PRESERVED IN THIS CALL.
6138	E86300	819		CALL	IQUANTZ	
		820				
		821		FRESUL	T "L2" RETURNED IN CX.	THIS VALUE IS DOUBLED THE
		822		#ACTUA	VALUE TO FACILITATE I	NDEX ADDRESSING.
		823				

VOH

L2, CX

LOC	OBJ	LINE	SOURCE			
		827		*****	******	*******
4		828				JE FOR THE CODED BITS *
		829				******
		830				
OIBF	A17600	831		MOV	AX, FSV	
		832				
0102	3B0F00	833		CMP	AX, FSVTH	
		834				
0105	730E	835		JAE	LABEL_MID_RISE	
		836				
		837		#FSV <	FSVTH	
		838				
		839				
0107	83F902	840		CMP	CX, 2	\$L2 = 1?
		841				
01CA	7509	842		JNE	LABEL_MID_RISE	
		843				
		844		iL2 = 1	, USE MID_TREAD QU	JANTIZER AND SET XOUT = 0
		845				
	C70574000000	846		KOV	XOUT, 0	
0102	EB1B90	847		JMF	INC_STEP	
		848				
		849				
0105		850	LABEL_MID_RISE:			
5		851		*11555 7	O HEE DV AT THE D	DINT. DICADLE INTERDUDT DEEDDE HEE
		852				DINT, DISABLE INTERRUPT BEFORE USE I/O ROUTINES AS A DEDICATED REGISTER
		853		15% 15	USEL BI INTERRUP!	1/O KOOTINES NO H DEDICATED REGISTER
0400		854		CLT		
0105	3 FH 5 87D9	855 856		CLI	BX, CX	CX CONTAINS "L2"
0110	0/07			AUNU	DAT LA	TON CONTINUE LE
Athe	3 2EF727					
VILLE		857		MIII	CS: YOURY1	LAY CONTAINS FOU
	200727	858		KUL	CS:YQEBX3	\$AX CONTAINS FSV
		958 859				\$AX CONTAINS FSV
	6 6719	858 859 860		XCHG	CS:YQEBX3 BX, CX	\$AX CONTAINS FSV
0105	6 6719	858 859 860 861		XCHG		\$AX CONTAINS FSV
	6 6719	858 859 860 861 862				\$AX CONTAINS FSV
01DE 01DI	6 67D9) FB	858 859 860 861 862 863		XCHG STI	BX, CX	
01DE 01DE	6 6709) FB : D1C0	858 859 860 861 862 863 864		XCHG STI ROL	BX, CX	\$AX CONTAINS FSV \$SCALE BY 2**15
01DE 01DE	6 67D9) FB	858 859 860 861 862 863 864 865		XCHG STI	BX, CX	
01DE 01DE	6 6709) FB : D1C0	858 859 860 861 862 863 864 865		XCHG STI ROL RCL	BX, CX	
01DE 01DE 01DE 01EC	6 8709 0 FB 0 D1C0 0 D1D2	858 859 860 861 862 863 864 865 865		XCHG STI ROL RCL	BX, CX AX, 1 DX, 1 PORATE SIGN BIT	
01DB 01DB 01DB 01EC	8 8709 0 FB 1 D1C0 0 D1D2 2 833E880000	858 859 860 861 862 863 864 865 866 867		XCHG STI ROL RCL	BX, CX AX, 1 DX, 1	
01DB 01DB 01DB 01EC	6 8709 0 FB 0 D1C0 0 D1D2	858 859 860 861 862 863 864 865 865		XCHG STI ROL RCL FINCORF	BX, CX AX, 1 DX, 1 PORATE SIGN BIT SIGN_FLAG, 0	
01DB 01DB 01DB 01EC	8 8709 0 FB 1 D1C0 0 D1D2 2 833E880000	858 859 860 861 862 863 864 865 866 867 868		XCHG STI ROL RCL FINCORF CMP JZ	BX, CX AX, 1 DX, 1 PORATE SIGN BIT SIGN_FLAG, 0	\$SCALE BY 2**15
01DB 01DB 01DB 01EC	8 8709 0 FB 1 D1C0 0 D1D2 2 833E880000	858 859 860 861 862 863 864 865 866 867 868 869		XCHG STI ROL RCL FINCORF CMP JZ	BX, CX AX, 1 DX, 1 PORATE SIGN BIT SIGN_FLAG, 0 LABEL_2	\$SCALE BY 2**15
01DB 01DB 01DB 01EC 01EC	8 8709 0 FB 1 D1C0 0 D1D2 2 833E880000	858 859 860 861 862 863 864 865 865 866 867 868 869 870		XCHG STI ROL RCL FINCORF CMP JZ	BX, CX AX, 1 DX, 1 PORATE SIGN BIT SIGN_FLAG, 0 LABEL_2	\$SCALE BY 2**15
01DB 01DB 01DB 01EC 01EC	8 8709 0 FB 1 D1C0 0 D1D2 2 833E880000 7 7402	858 859 860 861 862 863 864 865 866 867 868 869 870 871		XCHG STI ROL RCL FINCORF CMP JZ FXIN W	BX, CX AX, 1 DX, 1 PORATE SIGN BIT SIGN_FLAG, 0 LABEL_2 AS A NEGATIVE NUMBE	\$SCALE BY 2**15
01DB 01DB 01DB 01EC 01EC	8 8709 0 FB 1 D1C0 0 D1D2 2 833E880000 7 7402	858 859 860 861 862 863 864 865 866 867 868 869 870 871 872	LABEL_2:	XCHG STI ROL RCL FINCORF CMP JZ FXIN W	BX, CX AX, 1 DX, 1 PORATE SIGN BIT SIGN_FLAG, 0 LABEL_2 AS A NEGATIVE NUMBE	\$SCALE BY 2**15
01DB 01DB 01DB 01E0 01E7	8 87D9 D FB D D1C0 D D1D2 2 833E880000 7 7402	858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874	LABEL_2:	XCHG STI ROL RCL #INCORF CMP JZ #XIN W#	BX, CX AX, 1 DX, 1 PORATE SIGN BIT SIGN_FLAG, 0 LABEL_2 AS A NEGATIVE NUMBE DX	\$SCALE BY 2**15

LOC	0EJ	LINE	SOURCE			
		878				
		879			*******	
		880			CREMENT STEP SIZE OF P	
		881		*****	********	IXXX
		882				
		883				
OIEF	BB0E8000	884	INC_STEP:	MOV	CX, L2	
		885				
		886		FNEED	TO USE BX AGIAN, REFE	ER TO PREVIOUS USAGE FOR DOCUMENTATION
		887				
01F3		888		CLI		
	87CB	889		XCHG	CX, BX	
01F6	2E884704	890		VOM	AX, CS:QMLTEBX]	
01FA	87CB	891		XCHG	CX, BX	
01FC	FB	892		STI		
01FB	F7267600	873		HUL	FSV	
		894				
0201	D100	895		ROL	AX, 1	SCALE RESULT BY 2**15
0203	D102	896		RCL	DX, 1	
		897				
0205	81FAE803	398		CMP	DX. FSVMX	
	7606	899		JBE	LABEL_3	
	BAE803	706		MOV	DX. FSVMX	
	EB0950	901		JMP	APCMQ_END	
	83FACA	902	LABEL_3:	CMF	DX, FSVMN	
	7303	903		JAE	APCHQ_ENI/	
	BAOAGO	904		KOV	DX, FSVMN	
	2.10.100	905			DATE TOWN	
		906		INTH S	SV IN DX	
		907		/IILW 1	or an an	
0219	8B0E7400	706	APCMG_END:	MOV	CX, XOUT	
021B		909	H CHALLIE	RET	CA7 A001	
0211	00	910		WE!		
		911	AFCMQ	ENDF		
		912	n. one	LITE		
		913				
		914				
			AF 1557			
		915 +1	₹JEU1			

LOC GBJ

LINE	SOURCE		
916			
917	IQUANTZ	PRO	DC NEAR
918			
919	;*** *	*********	*********************
920	* I	PUT : ABS	S_XIN , FSV PASSED FROM MEMORY *
921	**		1
922	i* 01	TPUT: L	IN REGISTER CX. THIS VALUE IS *
923	**		DOUBLE THAT OF ACTUAL VALUE TO *
924	;*		FACILITATE INDEX ADDRESSING *
925	i* RE	GISTER USAGE:	*
926	**	BX	TEMPORARILY, VALUE SAVED AND *
927	*		RESTORED BY THIS PROCEDURE. *
928	i *		INTERRUPT DISABLED WHILE USING THIS*
929	÷*		REGISTER. *
930	**	DX	*
931	**		1
932	1 × R	EGISTER ASSIGNMENT	IT IN THIS PROCEDURE: *
933	**		1
934	į¥.	CX	CONTAINS IU *
935	**	REFER TO FO	ORTRAN LISTING FOR DESCRIPTION OF THESE *
936	;*	VARIABLES.	*
937	****)	********	*******************
938			
937			
940 +1	\$EJECT		

994

LOC	ORJ	LINE	SOURCE			
		941				
021E	C7067E000000	942		MOV	IL, 0	;IL = 0
0224	B90400	943		MOV	CX, NLEV* TYPE XQ	FIU = NLEV [* 2 FACTOR ADDED]
		944				
		945		#I = ()	L + IU) /2	
		946				
0227	8B167E00	947	LABEL_100:	VOM	DX, IL	
022B	03D1	948		ADD	DX, CX	
022D	DIEA	949		SHR	DX, 1	DIVIDE BY 2
022F	B1E2FEFF	950		AND	DX, OFFFEH	MAKE SURE IT IS AN EVEN NUMB
			ER			
0233	89167000	951		MOV	I, DX	SAVE I FOR FUTURE USE
		952			.GT. XQ(I)*FSV) GOTO 220	
		953				
		954		INFED 1	IN USE REGISTER BY AT THIS	POINT, DISABLE INTERRUPT BEFORE
		955		FUSING	o oor heoroten an in into	DINIT PROMPEE INTERNOT I PER ONE
		956		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		
		957				
0237	Få	958		CLI		
	8793	959		XCHG	DX. BX	
	2E8B4706	960		VOM	AX, CS:XQEBX3	#MOV XQ TO AX REGISTER
,023E		961		XCHG	DX, BX	THOU NO TO HA RECIOIEN
0240		962		STI		
V270		963		311		
0241	F7267600	964		MUL	FSV	;XQ IS SCALED BY 2##15
	D126	965			AX: 1	7AG 15 SCHLER B! 24415
	D102	965		REL	DX, 1	
	39167200	967			ABS_XIN, DX	
0240	7707	968		JA	LABEL_200	
		969		;IU = :		
2045	DEACTOR	970				
	8B0E7C00	971			CX, I	
0253	EB0990	972		JMP	LABEL_300	
		973				
0256		974	LABEL_200:	;IL = :		
	8B167C00	975		VOM	DX, I	
025A	89167E00	976		MOV	IL, DX	
		977				
	***************************************	978		WOUL	BV *1	
	8B167E00	979	LABEL_300:	VOM	DX, IL	
	830202	980		ADD	DX; 2) IL+ 1
	3BCA	981		CMP	CX, DX	
0267	77BE	982		JA	LABEL_100	
		983				
		984		FRESUL	T IU IN CX REGISTER	
4		985				
		985				
0269	C3	987		RET		
		988				
		989	IQUANTZ	ENDP		
		990				
		991				
7		992				
		993				

LOC	OBJ	LINE	SOURCE	
		995		
		970		
		997		
		998	CODE	
		999		
		1000		
		1001		
			\$EJECT	

ENDS

1035 +1 \$EJECT

LOC OBJ	LINE	SOURCE			
	1036				
	1037			ERRUPT FLAG (IF) IS DISAB	LED IN MOST PART
	1038		FOF THIS ROUTI	NE.	
0000 F0	1039 1040		PUSH	AX	MAY USE AX REGISTER TO
0000 50	1041		ruan	нл	OUTPUT ERROR_LED IN THIS
	1042				ROUTINE FRONTINE
	1043		************	*******	TROUTTRE
	1043		# FE_BUFFE		
	1045			********	
	1046		1 ********	*********	
	1047				
	1048				
	1049		# CHECK	FE INPUT POINTER	
	1050				
0001 83FE10	1051		CMP	SI, 16	THIS IS THE NORMAL OFFSET
	1052				FVALUE POINTING TO FEL (4)
0004 7413	1053		JE	LABEL_NORMAL_1	
	1054				
	1055 +1				
	1056 +1				
	1057 +1				
	1059 +1				
0006 A17000	1059 +1		KOV	AX, LED_DISPLAY_VALUE	
0009 FEC0	1060 +1		INC	AL	
000H 240F	1061 +1		AND	AL, OFH	
	1062			LED FRANK FUTA	
000D 7407	1063		JZ	LED_ERROR_EXT1	
	1054				
000F A37000	1065 +1 1066 +1		VOM	LED_DISPLAY_VALUE . AX	
0012 0AC4	1067 +1		OR	AL, AH	
0014 E664	1068 +1		OUT	ERROR_LED, AL	
	1069				
0016 BE1000		LED_ERROR_EXT1:	MOV	SI, 16	CORRECT POINTER VALUE
	1071				
0019		LABEL_NORMAL_1:	CHECK	FE OUTPUT POINTER	
	1073				
0019 83FB00	1074		CMP	BX, 0	THIS IS THE NORMAL VALUE
	1075				FPOINTING TO FEL(O)
001C 7413	1076		JE	LABEL_NORMAL_2	
	1077				
	1078 +1				
	1079 +1				
	1080 +1				
	1081 +1				
001E A17000	1082 +1		VON	AX, LED_DISPLAY_VALUE	
0021 FEC0	1083 +1		INC	AL ALI	
0023 240F	1084 +1		AND	AL, OFH	
. 0005 7407	1085		17	LED EDDOD EVTS	
0025 7407	1086 1087		JZ	LED_ERROR_EXT2	
	1087				
0027 A37000	1089 +1		MOV	LED_DISPLAY_VALUE, AX	
0027 A37000 002A 0AC4	1099 +1		OR	AL, AH	
VUZH VHC4	1070 11		UK	HE7RII	

1140 0054 1141 +1 LABEL_SOF_END:

1142 +1

1143 +1 1144 +1

LOC	OBJ	LINE	SOURCE		
0054	FA	1145 +1		CLI	
0055	B8E000	1146 +1		VOM	AX. ENABLE_NORMAL_MASK
0058	E67A	1147 +1		DUT	PIC_PORT_1, AL
005A	FB	1148 +1		STI	
005B	58	1149		POP	AX
0050	CF	1150		IRET	
		1151			
		1152			
		1153	SOF_INTERRUPT	ENDF	
		1154			
		1155			
		1155			
			\$EUECT		

LOC OBJ	LINE	SOURCE		
	1158			
005D	1157	PREDICTOR_OUTPUT_INTERRUPT	PROC	FAR
	1160			
	1161			
	1162	;**********************	*********	*******
	1163	* THIS PROCEDURE IS ACTIVAT	D BY THE SPEN	L SIGNAL ON THE DECODER BUS. #
	1164			THE PREDICTOR COEFFICIENTS FROM #
	1165	## THE PREDICTOR COEFF BUFFE		1
	1166	3;		
	1167	# REGISTER USA	GES:	
	1169	ix AX,		
	1169	11		
	1170		*****	*******
	117:			
	1172			
0050 50	11.3	PUSH	AX	*AX IS USED DURING THIS ROUT!
0000 OC	11.0	NE.	пл	THA 13 BOLD DOKING THIS ROOT.
	1174 +1			
	1175 ±1			
	1175 +1			
ACCE DECE	1177 11	MOU	AV FUADIF	DOE NADY
OCSE PEFEGG	1176 +1	MOV	AX, ENABLE.	[2] [1] [1] [1] [2] [2] [2] [2] [2] [2] [2] [2] [2] [2
00: E679	1179 +1	OUT	PIC_PORT_	l, AL
Oud: FB	1150 +1	SII		
	1181			
	1182			
	1103	FCON	IGURE THE DEC	ODER DATA BUS TO BE OUTPUT
	1184			
0064 B89955	1185	MOV	AX, D_DATA	
00d7 E74E	1198	DUT	DECODER_BU	JS_SPVR, AX
	1137			
	1188	; 001	PUT DATA	
	1189			
0069 3E8103	1170	WO.	AX, DS:PRE	DICTOR_BUFFER[BP][DI]
	1191			
006C E744	1192	OUT	D_DATA_BUS	Si AX
	1193			
	1194	FINC	REMENT BUFFER	POINTER
	1195			
005E 830702	1195	ADD		PREDICTOR_BUFFER
0071 83FF10	1197	CMP		_BUFFER_FRAME * TYPE PREDICTOR_BUFFER
0074 7203	1198	JB	LABEL_P_E	
	1199			
	1200		A STATE OF THE PARTY OF THE PAR	END OF CURRENT FRAME, END OF DUTFUT
	1201	;OF	A FRAME (THE N	- 2ND FRAME) PREDICTOR COEFFICIENTS.
	1202	\$SET	UP THE POINT	ER FOR INPUT NEW PREDICTOR
	1203	₹COE	FFICIENTS AT	END OF CURRENT FRAME(FRAME N).
	1204	THE	INPUT DATA W	ILL BE DEPOSITED INTO THE SAME FRAME
	1205	#BUF	FER, THEREFOR	BP REGISTER NEEDS NOT BE CHANGED.
	1206			
0076 BF0000	1207	MOV	DI, 0	
	1208			
	1209			
0079	1210	LAREL_P_EN: ;CHE	CK FOR END OF	DUTPUT CYCLE

0079 25	50400 1212	AND	AX, SPEN
007E 74	4F9 1213	JZ	LABEL_F_EN
	1214		
	1215	JON \$	SPEN IS FALSE(I.E. SIGNAL IS HIGH), TAKE DATA OFF THE BU
		S	
	1216		
0080 B8		MOV	AX, D_DATA_BUS_IN
0003 E		DUT	DECOPER_BUS_SPVR, AX
	1219		
	1220 41		
	1071 ±1		
	1022 +1		
	1000 11		
0051 F4		CLI	
603A RE		MOV	AX; ENABLE_NORMAL_MASK
0089 Ed			PIC_PORT_1, AL
005F F2		STI	FICEFORILIT HE
vvor -:	3 1227 †1 1228	511	
A-55 E		POP	AX
0030 58			
COBI- CF		IRE	
	1731		
	1232		
	1233	PREDICTOR_OUTPUT_INTERRUPT	ENDP
	1234		
	1235		
	1238		
	1237 +1	\$EJECT	

LCC ORJ	LINE	SOURCE		
	1238			
	1239			
008E	1240	FE_OUTPUT_INTERRUPT	PROC	FAR
VV.	1241	, ELGOTI O'LINIERMOIT	TNOC	
	1242			
		*	-	
	1243			************
	1244			CTIVATED BY FE ENABLE_ SIGNAL ON THE DECODER #
	1245			OUTPUTING FEH AND FEL TO THE DECODER BUS *
	1246	# ON ENTRY TO THE HA	NDLER, AX	IS PUSHED ONTO STACK, DATA BUS PORT IS *
	1247	## CONVERTED FROM INPL	IT STATE	TO OUTPUT STATE, THE APPROPRIATE FEX IS *
	1248	* OUTPUTER TO THE BUS	S, POINTE	R TO BUFFER IS INCREMENTED TO NEXT FEX TO BE *
	1249	* GUTFUT. THIS ROUTIN	WE WILL L	OOP AROUND UNTIL FE ENDABLE_ IS HIGH, THEN *
	1250			THE DATA BUS, CONVERT PORT TO INPUT STATE #
	1251	* AND EXIT		*
	1258	53		
	3253	* REGISTER USAGE!		
	1254	A) + B)		
	1.55	· · · · · · · · · · · · · · · · · · ·	未本本本本本本本本	***********
	1256			
	1257			
006E 30	1258		PUSH	AX
	1259			
	1240 +1			
	1261 +1			
	1222 11			
04.50 5.500	1730 Hi		WOU	AV ENAME DOE MACK
OORF BEFEND	1234 71		VOM	AX; ENABLE_SOF_MASK
6 0(9) E£74	1265 +1		OUT	PIC_PORT_1, AL
0094 FE	1266 +1		STI	
	1267			
	1268			
	1269		#CONFI	GURE DATA BUS PORT TO BE OUTPUT
	1270			
0095 BBP999	1271		MOV	AX, D_DATA_BUS_OUT
0098 E74E	1272		OUT	DECODER_BUS_SPVR, AX
00.0 27 11	1273		001	PLOOPER SOUTH THE THE
009A 8B4730	1274		MOV	AX, FE_BUFFER[BX]
V VV7H 024/3V			HUV	HAY FELDUFFEREDAJ
	1275			
	1276		FRX IS	THE POINTER TO THE NEXT WORD TO BE OUTPUT IN FE BUFFER
	1277			
0091 E74A	1278		OUT	D_DATA_BUS, AX
	1279			
	1280		FINCRE	MENT THE OUTPUT POINTER
	1281			
009F 83C302	1282		ADD	BX, TYPE FE_BUFFER
00A2 83FB40	1283		CMP	
00A5 7203	1284		JB	
	1285		MOV	BX, 0
00A7 BB0000			nuv	DAT V
*	1286			
	1287			
	1288			
00AA E54C	1289	LABEL_FE_ENABLE:	IN	AX, D_CONTROL_BUS
00AC 251000	1290		AND	AX, SFEEN
00AF 74F9	1291		JZ	LABEL_FE_ENABLE
	1292			

LOC 08J	LIME SOURCE	
	1293	NOW SPEEN_ IS FALSE(SIGNAL IS HIGH), TAKE DATA OFF BUS
00F1 R67B9B	1294	MOV AX, D_DATA_BUS_IN
00B4 E74E	1295	OUT DECODER_BUS_SPVR; AX
	1296	
	1297 +1	
	1298 +1	
	1299 +1	
	1300 +1	
0055 FA	1301 +1	CLI
00B7 B8E000	1302 +1	MOV AX, ENABLE_NORMAL_MASK
00BA E67A	1303 +1	OUT PIC_PORT_1, AL
COBC FI	1304 +1	STI
AASA 12	1305	311
AATT CO	1305	DOD AV
00ED 58		POF AX
AARE SE	1307	vorv
OOBE CF	1306	IRET
	1309	
	1310 FE_OUTPUT_INTERRUPT	ENDP
	1311	
	1312	
	1313	
	1314 41 \$EUECT	

11:47:02	10/07/82	PAGE	38
******	20101102		-

L60 03J	LINE	SOURCE			
	1315				
DOBF	1316	FE_INPUT_INTERRUPT	PROC	FAR	
***	1317	, LLIM O'LIM EMMO!	INOD		
	1318	************	*******	*******	*******
	1319			CTIVATED BY THE FE ENABL	
	1320			E FOR TAKING THE FEX(FEH	
	1321			E THEM INTO THE FE BUFFE	
	1322			PUSHES AX REGISTER ONTO	
	1323			IS VALID, THE FEX DATA	
	1324			. THE ROUTINE THEN INCRE	
	1325	## FOINTER AND EXIT.		HIL MODITAL THEA THORE	TERTS THE BOTTER +
	1324	it			
			OTEL HEADE	^.	
	1027		STER USAGE	3.	
	1326	; x	AX,SI		
	1325	1)	ada abada dhala a sada ah abada dhala		
	17.50	*************	********	*******	***************
	1331				
	1532				
	1333			ROUTINE IS ACTIVATED BY	FE_ENABLE SIGNAL ON THE
	1034		CODER	CONTROL BUS	
	1339				
	1335				
	1337				
096F 50	1338		PUSH	AX	
	1339 +1				
	13-0 +1				
	1741 +1				
	1342 41				
0000 EFFEC	1343 41		MOV	AX . ENABLE_SOF_MASK	
0003 E679	1344 +1		OUT	PIC_PORT_1, AL	
0005 Fh	1345 41		STI		
	1346				
6005	1347	LABEL_FE_CLK:	# CHECK	FOR FE CLOCK VALIDITY	
	1348				
0006 E544	1349		IN	AX, C_CONTROL_BUS	
00CE 250008	1350		ANI	AX: CFECLK	
00CB 74F9	1351		JZ	LABEL_FE_CLK	
	1352		FE CL	DCK IS TRUE(I.E. SIGNAL	IS HIGH), READ IN FEX FROM DATA
		BUS			
	1353				
4 00CD E542	1354		IN	AX, C_DATA_BUS	
	1355				
	1356				
00CF 894430	1357		HOV	FE_BUFFERISI], AX	
VVC7 07445V	1358		1104	I ELDOIT ENESTS! HA	
	1359		101 TO	THE OFFSET POINTER FOR	DEDUCTTION
			131 13	THE OFFSET FORMER FOR	DEI 031110R
	1360		********	MENT DOINTEDE	
	1361		FINLKE	MENT POINTERS	
	1362				
	1363		400	AT THE PP BUPPE	
00D2 83C602	1364		ADD	SI, TYPE FE_BUFFER	ATHOREMENT THE OFFICE
00D5 FF06B200	1365		INC	FE_SERVICE_COUNTER	FINCREMENT THE SERVICE
	1366				COUNTER TO ENABLE NON-INTERR
		UPT			
	1367				DRIVEN QUANTIZATION PROCESS

LOC OFJ	LINE SOURCE			
	17/6			
	1368	AVE	01 LEN EE BUEEFE + TVE	E EE BUEEFE AFAIR OF BUEFF
00D9 83FE40	1369	CMP	SI, LEN_FE_BUFFER * TYP	E FE_BUFFER SEND OF BUFFE
	R?			
00DC 7203	1370	JB	LABEL_FE_INPUT_END	
00DE BF0000	1371	VOM	SI, 0	FRESET POINTER TO BEGINNING
	1372			FOF CIRCULAR BUFFER.
	1373			
00E1	1374 +1 LABEL_FE_	INPUT_END:		
	1375 +1			
	1376 +1			
	1377 +1			
00E1_FA	1378 +1	CLI		
00E2 B8E000	1379 +1	MOV	AX, ENABLE_NORMAL_MASK	
00E5 E67A	1380 +1	OUT	PIC_PORT_1, AL	
00E7 FE	1381 +1	STI	TO THE	
00E8 58	1382	POP	AX	
0000 111	1383		nn e	
00E9 CF	1384	IRET		
0067 OF		INE		
	1385	THEFTHER FURN		
		INTERRUPT ENDP		
	1387			
	1388			
	1389 +1 \$EJECT			

11:47:02	10/07/82	PAGE	40
******	10,0,,00	11100	

1442 +1 1443 +1

0100 FA

LOC DEJ	LINE	SOURCE			
	1390				
	1391				
	1392				
00EA	1393	PREDICTOR_INPUT_INTE	RRUPT	PROC FAR	
	1394				
	1395				
	1376				
	1397	;********** *	******	***************	*******
	1398	* THIS PROCEDURE IS	ACTIVATED	BY THE PEN_ SIGNAL ON THE COD	ER BUS. #
	1399	# IT INPUTS THE PRE	DICTOR COE	FFICIENTS FROM THE CODER BUS AN	D STORES THE *
	1400	# DATA IN THE PREDI	CTOR_COEFF	BUFFER.	
	1401	ix			*
	1401	i* REG	ISTER USAGE	S:	1
	1403	ik	AX, DI		
	1404	; ×			1
	1465	**************************************		***********	*****
00EA 50	1406		PUSH	AX	
	1407				
	1403 +1				
	1409 41				
	1410 +1				
	1,11 +1				
OCER BEFECO	1412 +1		MOV	AX, ENABLE_SOF_MASK	
00EE E67A	1413 +1		OUT	PIC_PORT_1, AL	
OCFO FE	1414 +1		STI		
	1415			COD D OLDOW HALTD	
	1416		11251	FOR P CLOCK VALID	
	1417				
A.F	1418	TATEL D. CLASS	TM	AV C CONTROL PUR	
00F1 E344	1419 1420	LABEL_P_CLK:	IN	AX, C_CONTROL_BUS	
ADDT SERVICE			AND	AX, CPCLK	
00F3 252000	1421 1422		JZ	LABEL_P_CLK	
0076 7475	1423		32	LADELLI-CLN	
	1424				
	1425		10 010	OCK IS VALID, READ IN PREDICTOR	COFFETCIENTS FROM
	1426		†DATA		DOET TOTENTO THOM
	1427		7211111	200	
	1428				
00F8 E542	1429		IN	AX, C_DATA_BUS	
	1430				
	1431		FBP SF	PECIFIES THE FRAME STARTING OFFS	SET (0, 16, 32 DR 48)
	1432			PECIFIES THE OFFSET WITHIN EACH	
00FA 3E8903	1433		VOM	DS:PREDICTOR_BUFFERERPJEDIJ,	AX
	1434				
00FD 830702	1435		ADD	DI, TYPE PREDICTOR_BUFFER	INCREMENT OFFSET
	1436				FVALUE WITHIN ONE FRA
		ME			
	1437				
	1438				
	1439 +1				
	1440 +1				
	1441 +1				
The second second	1000 11				

CLI

L90	CEG	LINE	SOURCE	
0101	B6E000	1444 +1	MOV	AX, ENABLE_NORMAL_MASK
	E676	1445 +1	TUO	PIC_PORT_1, AL
0106		1446 +1	STI	
		1447		
0107	::R	1448	POP	AX
0105		1449	IRET	
		1450		
		1451	PREDICTOR_INPUT_INTERRUPT	ENDF
		1452		
		1453		
		1454		
		1405 +1	\$EUECT	

LOC OBJ	LINE	SOURCE			
	1456				
0109	1457	EXCEPTION_INT	PROC	FAR	
	1458				
	1459	*******	********	************	******
	1460	# THIS IS	THE EXCEPTI	ION HANDLER FOR THE 8086	CPU. #
	1451	# UNDER N	ORNAL OPERAT	TON ENVIRONMENT, THIS C	ODE SHOULD *
	1462	## NEVER I	E EXECUTED.		1
	1463	# IN THE	UNLIKELY EVE	ENT THAT THIS HANDLER IS	ACTIVATED,*
	1464	## IT SETS	THE LED_DIS	SPLAY_VALUE TO OFFH AND	OUTPUT *
	1465	# TO THE	ERROR_LED.		1
	1466	11	REGIST	TER USAGES:	
	1467	;*		AX,	1
	1468	ix			1
	1465	*******	********	*********	******
	1470				
	1471				
	1472	PEXTERNAL	INTERRUPTS A	ARE DISABLED DURING EXEC	UTION OF THIS HANDLER
	1473				
	1474				
0107 50	1475		PUSH	AX	
01 VF 88FF FF	1470		MOV	AX, OFFFFH	
0:00 A37939	34.7		VOM	LED_DISPLAY_VALUE: AX	
0110 Eo54	1476		OUT	ERROR_LED, AL	
0112 59	1479		POP	AX	
0113 F1	1480		STI		
0114 CF	1481		IRET		
	1482				
	1493				
	1484	EXCEPTIONLINT	ENDP		
	1485				
	1486				
	1486 1487				
		INTERRUFT_CODE	ENDS		

ASSEMBLY COMPLETE: NO ERRORS FOUND

B2. <u>5-BIT ADAPTIVE RESIDUAL QUANTIZER</u>

VAXZVMS 8056/8088 MACRO ASSEMBLER V1.0VX ASSEMBLY OF MODULE M5LDEM OBJECT MODULE PLACED IN MSLDEM.OBJ NO INVOCATION LINE CONTROLS

LINE	SOURCE		
1			
2			
3		NAME MSLDEM	
4			
5			
6			
7	***********	***********	*****
8	† *		*
9	** PROJECT:	498A RELP CODEC	
10	; » BOARD:	MULDEN SIMULATOR	*
11		2 FIRMWARE EPROMS (2732A-2)	1
12	# BOARD LOCATION:		*
13	* PROGRAMMABLE		
14	** DEVICE NUMBER:	60-0670 - 60-0671	*
15	it	00 00/0/ 00 00/1	*
10		***********	*****
16 17	1*********	**************** ********************	-4-4-4-4-4-4-4-4-4
	\$EJECT		

LOC OBJ	LINE	SOURCE							
	19								
	20								
	21								
	22								
	23	;*************************************							
	24	* * * * * * * * * * * * * * * * * * *							
	25	* THIS FIRMWARE SIMULATES THE MULtiplexor/DEMulti- *							
	26	ix plexor IN	THE RELP	CODEC SY	STEM. REFER TO MDA *				
	27	# DOCUMENT O	0 - 3035	- R01 FD	R DETAIL.				
	28	* *							
	29	;********* *	******	*******	************				
	30								
	31								
	32								
	33								
	4	************	***						
	35	# SYSTEM CONSTANTS	*						
	36	*************	***						
	37								
	35								
	39								
0020	ų(i	LEK_FE_BUFFER	EQU	32	#ACCOMODATE 16 PAIRS OF FEX				
	41								
9008	42	LEN_F_BUFFER_FRAME NTS	EQU	8	#EACH FRAME ACCOMODATES 8 PREDICTOR COEFFICIE				
	43								
0020	44	LAST_P_BUFFER_FRAME	EQU	32	THE BASE OFFSET ADDRESS OF LAST P FRAME				
4414	45	LHOTE LIGHT ENER WHILE	Luc	02	FROM THE BEGINNING OF P BUFFER, THIS VALUE				
	46				MUST BE IN MULTIPLE OF LEN_P_BUFFER_FRAME *				
		TYPE PLBUFFER			THE PERMITTER OF PERMITTER WHILE T				
	47	THE TEDUTEN			THE FRAME DELAY IS EQUAL TO 1 +				
	46				(LAST_P_BUFFER_FRAME / 16)				
	49				THE TENER PROPERTY AND				
	50								
0001	51	FE_HIGH	EQU	1	THIS REPRESENTS THE FEH'S TYPE				
0000	52	FE_LOW	EQU	Ô	THIS REPRESENTS THE FEL'S TYPE				
VIVU	53	I L_LOW	260	•	THIS REPRESENTS THE FEE STITE				
0016	54	NLEV	EQU	16	NUMBER OF QUANTIZER REGIONS				
VVIV	55	MLLY	Lub	10	THORDER OF BOTHLIZER REDICTO				
0050	56	FSVMN	EQU	80	FSVMN - MINIMUM SCALING FACTOR				
0000	57	1 OVIIII	Luo	UV	, TOTAL HARADO BONEIRO I NOTOR				
4540	58	FSVMX	EQU	8000	FSVHX - MAXIMUM SCALING FACTOR				
1F40		LOVIIX	EGO	0000	Frank - Hevinon Scheing Freigh				
0000	59	FOUTU	FOII	100	FSVTH - THRESHOLD VALUE. IF THE				
0078	60	FSVTH	EQU	120					
	61				SCALE FACTOR IS LESS THAN FSVTH, A				
	62				OF A MID-RISE QUANTIZER. THE MID-TREAD				
	63								
	64				QUANTIZER USES ZERO AS AN OUTPUT LEVEL				
	65				FINSTEAD OF YQ(1).				
0000	66	DOCTTINE	EUN	۸	POSITIVE SIGN				
0000	67	POSITIVE	EQU	0	ALDSTITAE STOK				
0004	68	NECATIVE	FOIL		*NECATTHE CICH				
0001	69	NEGATIVE	EQU	1	*NEGATIVE SIGN				
	70								
	71								

LOC OBJ	LINE	SOURCE			
	72 73	# HARDWARE DEPENDE	NT CONSTA	ANTS	
0042	74 75	C_DATA_RUS	EQU	42H	FTHIS IS THE PORT(B) ADDRESS ON THE 8255'S FOR THE 16 BIT CODER DATA BUS
0044	76 77 76	C_CONTROL_BUS	EQU	44 H	FTHIS IS THE PORT(C) ADDRESS ON THE 8255'S
0046	79 80 81 82	CODERLBUS_SPVR	EQU	4 6H	FTHIS IS THE SUPERVISOR PORT(CONTROL PORT) FOR THE CODER BUS(THE CONTROL BUS AND THE DATA BUS)
	83 84 85				7 THE DATA BOS7
004A	86 87 83	D_DATA_BUS	EQU	4 AH	FITHIS IS THE PORT(B) ADDRESS ON THE 8255'S FFOR THE 16 BIT DECODER DATA BUS
004C	37 90 91 92	D_CONTROLLEUS	EQU	4CH	FOR THE 16 BIT DECODER CONTROL BUS
004E	93 94 95	DECODER_BUS_SPVR	EGN	4EH	FTHIS IS THE SUPERVISOR PORT(CONTROL PORT) FOR THE DECODER BUS(THE CONTROL BUS AND
	96 97 93				FTHE DATA BUS)
	100 101				
0064	102 103 104	ERRORLLED	EQU	64H	
0066	105 106 107	ERROR_BUS_SPVR	EQU	66H	;
	108 105	\$8259A PIC DEPEN	DENT CONS	TANTS	
0078 007A	110 111 112	PIC_PORT_0 PIC_PORT_1	EQU	78H 7AH	#8259A PROGRAMMABLE INTERRUPT CONTROLLER #CONTROL PORTS.
0013 0008 0003	113 114 115 116	ICW1 ICW2 ICW4	EQU EQU EQU	13H 08H 03H	#SINGLE PIC, ICW4 NEEDED #STARTING INTERRUPT VECTOR = 8 #SFNM = 0, AEDI = 1, NON-BUFFERED MODE
00FE	117 118 119	ENABLE_SOF_MASK	EQU	OFEH	#SP/EN WILL HAVE INPUT = 1 =>MASTER #UPM = 1 =>8086/88 SYSTEM. #MASK OUT ALL EXCEPT #SOF INTERRUPT(RO)
00E0	120 121 122 123 124	ENABLE_NORMAL_MASK	EQU	0E0H	#MASK OUT UNUSED #INTERRUFT(R7,R6,R5)
6	125 126	\$8255A PPI DEPEN	TENT CONT	STANTS	

8086/8	8057/6	038	MACRO	ASSEMBLE	R MSLIE
--------	--------	-----	-------	----------	---------

LOC OBJ	LINE	SOURCE			
	127				
9391	128	C_DATA_BUS_IN	EQU	9B9BH	PROGRAM THE CODER DATA BUS TO BE INPUT, ALL
	129				FOTHER PORTS ON CODER BUS ARE INPUT.
	130				#MSB AND LSB OF THIS WORD EACH ADDRESSES TO O
		NE PPI			
	131				
9099	132	C_DATA_BUS_OUT	EQU	9999H	FPROGRAM THE CODER DATA BUS TO BE OUTPUT(IE.
	133				FTAKE CONTROL OF THE DATA BUS), ALL OTHER
	134				PORTS ON THE CODER BUS ARE INPUT. MSB
	135				AND LSE OF THIS WORD EACH ADDRESSES ONE PPI
	135				
9B7B	137	D_DATA_BUS_IN	EQU	9B9BH	PROGRAM THE DECODER DATA BUS TO BE INPUT, AL
		L			
	138				FOTHER PORTS ON DECODER BUS ARE INPUT.
	139				MASE AND LSB OF THIS WORD EACH ADDRESSES TO C
		ME FFI			
	140				
9999	141	D_DATA_BUS_OUT	EQU	9999H	PROGRAM THE DECODER DATA BUS TO BE OUTPUT(IE
	142				TAKE CONTROL OF THE DATA BUS), ALL OTHER
	143				PORTS ON THE DECODER BUS ARE INPUT, MSB
	144				JAND LSB OF THIS WORD EACH ADDRESSES ONE PPI.
	145				
0092	146	ERROR_LED_ON	EGU	0092H	#PROGRAM THE ERROR BUS TO BE:
	147				FORT A - INPUT) NOT USED HERE
	148				; PORT B - INPUT > NOT USED HERE
	149				FORT C - OUTPUT > TO ERROR_LED
	150				
009B	151	ERROR_LED_OFF	EQU	009BH	PROGRAM THE ERROR BUS TO BE:
	152				FORT A - INPUT) NOT USED HERE
	153				FORT B - INPUT) NOT USED HERE
	154				# PORT C - INPUT > TO ERROR_LED
	155				THE LED DISPLAY IN THIS STATE WILL
	150				#BE "FF"
	157				
	158	ACCUITODI DIIO DEI	DENDENT OF	WOTANTO	
	159	CONTROL BUS DE	PENDEN! C	INSTANTS	
	160	*DECORED DUC			
	161 162	#DECODER BUS			
8004	163	onra:	FOU	000411	ANACY FOR CREW CIONAL (LOW TRUE)
0004 0010	164 165	SPEN SFEEN	EQU	0004H	
0010		SPEEN	EQU	0010H	#MASK FOR SFEEN SIGNAL(LOW TRUE)
	166				
	167 168	CODER BUS			
	169	i			
	170				
0800	171	CFECLK	EQU	0800Н	#MASK FOR CFECLK SIGNAL(HIGH TRUE)
0020	172	CPCLK	EQU	0020H	#MASK FOR CPCLK SIGNAL(HIGH TRUE)
3020	173	OI OLI	Luo	VV2VII	THICK TON OF OUR DIDINECTION TRUE!
	174				
	117				
	175				
	175 176	#MDA IN HOUSE MONITO	R ENTRY P	TAIC	

SOURCE LINE

0010 FESF

178 179

MONT_86_IP EQU 0010H #OFFSET ADDRESS
MONT_86_CS EQU 0FE9FH #SEGMENT ADDRESS

180

181 +1 \$EJECT

0022 (1

		11:51:55	10/07/82	PAGE	6
D AT	1 ОН				

*** ES. N.	CURRENT	IMPLEMENTA	TION		

F00	OBJ	LINE	SOURCE		
		182			
		183			
		184	INTERRUPT_VECTOR	SEGMENT	WORD AT OH
		185	IN ENNOT I LYECTOR	SEUNERI	WUNTI HI ON
		186			
		187			
		188		******	
		189		* INTERRUPT VECTOR	
		190	;	**********	*****
		191			
		192		NT 5 TO 32 FOR INTE	
		193	•	VIOLATES THIS RES	TRICTION.
		194			
		195			
		196	18086 PREDIFINED	INTERRUPTS: (INT 0	TO 4)
		197			
0000	(1	123	DIVIDE_INT_IP	DW 1 DUP(?)	
	7777				
	1				
0002		199	DIVIDE_INT_CS	DW 1 DUP(?)	
0002	7224	177	DIVIDELIMILES	DW 1 DUP(:)	
	4		ATUR E ATEL T		
0004		200	SINGLE_STEP_I	P DW 1 DUP(?)	
	7001				
)				
0006		201	SINGLE_STEP_C	S DW 1 DUP(?)	
	9911				
)				
0008	1.	202	NMI_IF	DW 1 DUP(?)	
	9771				
)				
0006	(1	203	NMI_CS	DW 1 DUF(?)	
	7777				
)				
3000	(1	204	BREAKPOINT_IP	DW 1 DUF(?)	
	????				
)				
	(1	205	DECAMPOINT OF	DW 1 DUP(?)	
OUVE	7777	203	DKEHNLO1K1_CO	TOW I DUT(!)	
)				
0010		00/	OUEDELOU TO	DIL 4 DUD/SI	
0010		206	OVERFLOW_IP	DW 1 DUP(?)	
	????				
0010)	507	DUEDEL DIL OD	DIL A DUDIO	
0012		207	OVERFLOW_CS	D₩ 1 DUP(?)	
	????				
)				
		208			
		209			
		210	MULDEM APPLICATI	ON INTERRUPTS: (INT	8 TO 15)
		211			
0020		212	ORG 2	OH	
0020	(1	213	INT_8_IP	DW 1 DUP(?)	
4	????				
)				
		The state of the s			

214 INT_8_CS DW 1 DUP(?)

LOC	083	LÎNE	SOURCE	
	71?7			
0024	1777	215	INT_9_IF	DW 1 DUP(?)
0026) (1 ????)	216	INT_9_CS	DW 1 DUF(?)
0028	(1 ????	217	INT_10_IF	DW 1 DUP(?)
002A	1777	218	INT_10_CS	DW 1 DUF(?)
0020	????	219	INT_11_IP	DW 1 DUP(?)
002E	1111	220	INT_11_CS	DW 1 DUP(?)
0030	1117	221	INT_12_IP	DW 1 DUP(?)
0032	7777	222	INT_12_CS	DW 1 DUP(?)
0034	7?7?	223	INT_13_IP	DW 1 DUP(?)
0036) (1 ••••	224	INT_13_CS	DW 1 DUF(?)
0038) (1 ???? 	225	INT_14_IP	DW 1 DUF(?)
003A	(1 7777	226	INT_14_CS	DW 1 DUP(?)
0030	9999	227	INT_15_IP	DW 1 DUF(?)
003E	9999	228	INT_15_CS	DW 1 DUP(?)
		229		
		230 231	INTERRUPT_VECTOR	ENDS
	No.	232	INTERNOT TEVESTOR	LILDO
		233 234		
		235 +1	\$EJECT	

261

262

0086 (1

????

FE_TYPE

DW 1 DUP(?)

LOC	ORJ	LINE	SOURCE			
)					
		263				
0038	(1	264	SI	GN_FLAG	DW	1 DUP(?)
	????					
)					
		265				
		266				
		267	DATA	ENDS		
		268				
		269				
		270				
		271				
		272				
		273				
		774				
		275				
		276				
		277	STACK		SEGMENT	
		278				
		279		;************	*****	********
		280		# THE STACK IS	SOLELY	USED BY THE 8086 TO #
		281				IN INTERRUPT ROUTINES*
		282				*******
0000		283			DW	60 DUF (?)
	????					
)					
0078		284	708	1	LAREL	WORD
		285				
		28€	STACK		ENDS	
		287				
		288 +1	\$EJECT			

LOC GBJ

315 +1 \$EJECT

LINE SOURCE SYSTEM MACROS! THE MACROS ARE NOT LISTED IN THE ASSEMBLER GENERATED LISTING, REFER FTO THE SOURCE FILE FOR MACRO CONTENTS. 3.05

LINE	SOURCE	
316		
317		
318		
319	;************************	********
320	18	1
321	** REGISTER USAGE IN M5LDEM SIMULATOR SYS	TEH: *
322	ix	1
323	;******************************	*******
324		
325		
326	DEDICATED USAGE FOR ALL PARTS OF THE SYSTEM AT ALL	TIME:
327		
328	FSI - FE_BUFFER INPUT POINTER	
329	#BX - FE_BUFFER OUTPUT POINTER ETHIS RE	GISTER IS ALSO USED UNDER
336	; NON-INTERRUPT DRIVEN QUANTIZATION	PROCESS, HOWEVER, THE
331	; ORIGINAL REGISTER IS PRESERVED]	
332	#BP - FREDICTOR_BUFFER FRAME POINTER	
333	#DI - PREDICTOR_BUFFER OFFSET POINTER	
334		
335		
336	CS - CODE SEGMENT	
337	FDS - DATA SEGMENT	
338	JES - DATA SEGMENT OR INTERRUPT_VECTOR	SEGMENT
339	#SE/SP - STACK OPERATION	
340		
341		
342	;UNASSIGNEH REGISTERS:	
343		
344	THESE REGISTERS DO NOT CARRY DEDICATED FUN	CTIONS IN THE MSLDEM
345	isimulator:	
346		
347	AX, CX, DX	
348		
349 41	\$EJECT SEEDING	

L00	DBJ	LINE SOURCE			
040	0076				
		379			
		380	FRMLT	- ARRAY	OF NLEV, QUANTIZER MULTIPLIERS
		381	j		SCALED BY ** IN ** REPRESENTATION
0042	0000	382	QMLT		0, 13926, 13926, 13926, 13926, 13926, 13926
0044	6636				
	6635				
0048	563c				
	6636				
6645	6636				
004E	6636				
0050	6536	383		DW	13926, 13926, 19661, 22938, 26214, 29491
0052	6636				
0054	CD40				
0056	9A59				
8300	6666				
005A	3373				
0050	0080	384		DW	32768, 36045, 39322, 42598
005E	CDSC				
0050	9499				
0062	66A6				
		385			
		386			
		387 +1 \$EJECT			

LOC	OFJ		LINE S	OURCE		
			443			
008F	PS0901		444		MOV	AX, OFFSET EXCEPTION_INT
0092	FB90	R	445		VOM	BX, SEG EXCEPTION_INT
0096	26A30000		446		VOM	DIVIDE_INT_IP, AX
009A	26891E0200		447		MOV	DIVIDE_INT_CS, BX
009F	26A31000		448		NOV	OVERFLOW_IP, AX
00A3	26891E1200		449		MOV	OVERFLOW_CS, BX
			450			
			451			
00A8	26070420000000		452		MOV	INT_8_IP, OFFSET SOF_INTERRUPT
OOAF	2607062200	6	453		MOV	INT_8_CS; SEG SOF_INTERRUPT
			434			
00B6	26070614005D00		455		MOV	INT_9_IP, OFFSET PREDICTOR_OUTPUT_INTERRUPT
OGEL	2607062500	F	456		MOV	INT_9_CS, SEG PREDICTOR_OUTPUT_INTERRUPT
			457			
0004	26070609008200		458		MOV	INT_10_IP, OFFSET FE_OUTPUT_INTERRUPT
	2607062600	+	459		MOV	INT_10_CS, SEG FE_DUTPUT_INTERRUPT
			460			
00B2	2607062000E400		451		MOV	INT_11_IP, OFFSET PREDICTOR_INPUT_INTERRUPT
	2807082500		452		MOU	INT_11_CS, SEG PREDICTOR_INPUT_INTERRUPT
	200,012111		463			
OOF	2607663000RF00		464		KOV	INT_12_IP, OFFSET FE_INPUT_INTERRUPT
	2607063200		465		MOV	INT_12_CS; SEG FE_INPUT_INTERRUPT
VVL	200, 40, 244	10		EJECT	1101	THE TELEST OF TELLING PLANTS OF THE TELEST O
			700 11 \$	-U-LI		

LOE OF O		LINE	SOURCE					
		467						
		468						
		459		ASSUME	ES:DATA			
		470						
		471						
		472						
		473		ITIMIT	ALIZE BO	86 PR	DCESSOR ENVI	RONHENT
		474						
		475						
00EE B8	R	476			MOV	AX,	DATA	CANNOT MOVE IMMED. VALUE TO
OOF1 BENS		477			MOV		AX	SEGMENT REGISTER.
		478						POLONENT REGISTERS
OOFE BECO		479			MOV	ES,	AX	FES AND DS ARE REFERING TO SAME
		480						#SEGMENT
00FS B6	£	481			VOM	AX.	STACK	POLOTICAL TO A CONTROL OF THE CONTRO
00F8 8ED0		482			MOV	SS,		
00FA BC7800		483			MOV		OFFSET TOS	
VV. 11 20,000		484				0	011021 100	
		485						
		486		:TNTTT	4! T7E AL	I CYC	TEM HARDWARE	
		467		7111-11	neite ne	L 010	ILII HANDWAKE	
		488						
		489						
		490		;	*****	****	******	
		491		;		I		
		492		;			*****	
		497		1	******	4 0 0 0 0	****	
		404						
		495		TTHATE	DU! AUT	TTAMO	FOT TS 11SE	D HERE, REFER TO INTEL APPLICATION
		496		j				E 8259A PROGRAMMABLE INTERRUPT
		497		;				EADING "AUTOMATIC EDI MODE"
		498			CON	INOLL	LN Y DRULK II	ENDING HOTOIRTIC LOT HODE
		499						
00FD E013		500			MOV	۸۱ .	ICW1	
00FF E57E		501			OUT		PORT_O, AL	
VVFF E2/C		502			001	FIL	LICKILOS HL	
0101 BA7A00		503			MOV	nv.	PIC_PORT_1	
0104 B008		504			MOV		ICW2	
0104 BCC6		505			OUT	DX,		
VIVO EE		505			001	DAY	HL	
		507			*TCU 7	TC N	OT MEEDED ED	R CURRENT HARDWARE CONFIGURATION
		508			SILW 2	12 W	DI MEEDED FO	IN CURRENT HANDWHALE CONFIDURATION
0107 B003		509			MOV	A! -	ICW4	
0107 E5						DX,		
V107 E1		510			OUT	DAT	HL	
010A BOFE		511 512			MOV	AI .	ENABLE_SOF_	MACY
								ЛСНП.
010C EE		513			OUT	DX.	HL	
		514						
		515			*****	****	*****	
		516		•			*****	
		517		!		P		
		518		;	*****	****	*****	
		519						
		520			ALL V ALL	pont	ADE DODODAN	WED TO BE TABLE DODGE THE MODE A
		521		ILIMIT	HLLT ALL	. PUKI	HRE PRUBRAM	MED TO BE INPUT PORTS IN MODE O

LOC OBJ	LINE	SOURCE		
	522	;EXCEPT	THE EF	RROR LED PORT, WHICH WILL BE DUTPUT ALL THE TIME
	523			
010D B8780B	524		MOV	AX, C_DATA_BUS_IN
0110 E746	525		OUT	CODER_BUS_SPVR, AX
	526			
	527			
0112 B8989B	528		MOV	AX, D_DATA_BUS_IN
0115 E74E	529		OUT	DECODER_BUS_SPVR, AX
VIII 27 III		\$EJECT		

LOC DBJ	LINE	SOURCE			
	531				
	532			**************	
	533			ZE APPLICATION PROGRAM ENVIRONMENT	
	534	;	*******	************	1
	535				
	536				
	537				
	538				
	539				
	540		; FIL	L BUFFER AREAS WITH ZEROS	
	541				
0117 B80000	542		VOM	AX, 0	FILL VALUE
011A B9180(-	543		MOV	CX, LENGTH PREDICTOR_BUFFER	FITERATION COUNT
011B BF0000	544 545		NOV	DI, OFFSET PREDICTOR_BUFFER	
	546			SETS TO THE SEGMENT BASE OF DATA S	פר מיד איד
	547		723 3	DETO TO THE DEDMENT DADE OF DATA S	DEUNENI
0120 F3	548	D.	EP STOS	PREDICTOR_BUFFER	
0120 F3	340	V	EF' \$105	PREDICTUR_BUFFER	
Olfi Ht	549				
	550				
0122 B92000	551		MOV	CX, LENGTH FE_BUFFER	FITERATION COUNT
0125 BF3000	552		MOV	DI, OFFSET FE_BUFFER	FITERHITON COUNT
V120 1: 0000	553		HOV	DIF OFFSET FELBOFFER	
0128 F3	554	P:	EP STOS	FE_BUFFER	
0129 AF				TELPOTTEN.	
	555				
	556				
	557		FRESE	T SERVICE COUNTER	
0124 C7068200000°	558		MOV	FE_SERVICE_COUNTER, 0	
	559				
	560		FINIT	IALIZE FE TYPE TO LOW	
0130 C70686000000	561		MOV	FE_TYPE, FE_LOW	
	562				
	563				
	564		FINIT	TIALIZE BUFFER POINTERS TO APPROPE	RIATE VALUES
	565				
	556			DICTOR BUFFER , BP IS THE FRAME PO	
	567		; DI :	IS THE OFFSET POINTER WITHIN A FRO	AME.
A47/ PERAAA	568		Wall		100 FIRST DOF THEFES
0136 BD2000	569		VOM	BP, LAST_P_BUFFER_FRAME	ON FIRST SOF INTERRUPT AFTER
	570	THE			FOWER UP, SOF_INTERRUPT ROUT
	E71	INE			THILL SET DD - A. DI - A
	571 572				WILL SET BP = 0, DI = 0
0139 BF1000	573		HOV	DI, LEN_P_BUFFER_FRAME * (TYPE	DOEDITOTOD BUFFED)
VIS/ Drive	574		nov	DIT LENGT EDOTT ENGINE 4 (TITE	L PREDICION_BOITER/
	575		ter r	BUFFER INITIALIZATION	
	576		7FE 1	OFFICE THITTELEFITOR	
013C RE1000	577		HOV	SI, 16	POINTS TO FEL(4)
013F BB0000	578		HOV	BX, 0	, 01110 10 / LE(1)
2201 22000	579		1104		
	580		*RUAN	HTIZATION SERVICE	
	581		Juoni		
0142 C70684001000	582		MOV	FE_SERVICE_PTR, 16	

11:51:55 10/07/82 PAGE	11:51:55	10/07/82	PAGE	19
------------------------	----------	----------	------	----

LOC OBJ	LINE SOURCE			
0148 070878005000	583	MOV	FSVL, FSVMN	
014E 07067A005000	584	VOM	FSVH, FSVMN	
	585			
	58á			
	587			
0154 FB	588	STI		FENABLE EXTERNAL INTERRUPT
	589			
0155 F4	590	HLT		WAIT UNTIL THE FIRST
	591			START OF FRAME INTERRUPT
	592			
	593 +1 \$EJECT			

8086/8087/8088 MACRO ASSEMBLER MSLDEM

594 595 596 58358820000 617 CMP FE.SERVICE COUNTER 6155 8535820000 616 585 618 585 618 585 619 585 619 585 619 585 619 585 619 585 619 585 619 585 619 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 617 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 616 585 619 617 585 619 618 585 619 619 619 619 619 619 619 619 619 619	L00 OFJ	LINE	SOURCE
595 596 5***********************************		594	
597			
597		596	;*************************************
599		597	H I I I I I I I I I I I I I I I I I I I
600		598	* START OF APPLICATION PROGRAM *
601		599	ix ix
150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150		600	## AFTER POWER UP, THE FOLLOWING APPLICATION #
603		601	** PROGRAM WILL NOT START EXECUTION UNTIL THE *
0156 605 605 606 607 608 FLOOP AROUND UNTIL THERE IS A NEW FE TO SERVICE 609 609 609 610 611 JE LABELLWAITLFOR_NEW_FE 612 612 613 614 615 614 615 615 615 615 615 615 615 615 615 616 617 CMP FE_SERVICE COUNTER 617 617 618 617 619 617 619 617 619 617 619 617 619 617 619 619 619 619 619 619 619 619 619 619		602	** FIRST SOF_INTERRUPT HAS BEEN SERVICED. *
0156		603	;*************************************
0156		604	
607 608 608 609 0156 39368460 610 0156 7482 611 612 612 613 614 615 615 615 616 617 CMP FELSERVICE_PTR, SI LABEL_WAIT_FOR_NEW_FE 617 618 619 619 610 610 610 611 611 612 614 615 615 616 617 CMP FELSERVICE_COUNTER 618 619 619 619 619 619 619 619 619 619 619		605	
608 \$100P AROUND UNTIL THERE IS A NEW FE TO SERVICE 609 0156 39368400 \$10 CMP FELSERVICE_PTR, SI 0156 7482 611 JE LABEL_WAIT_FOR_NEW_FE 612 613 \$TWO POINTERS ARE NOT EQUAL, POSSIBLE NEW FE'S 614 615 616 \$CHECK SERVICE COUNTER 0150 8538820000 617 CMP FELSERVICE_COUNTER, 0	0156	606	LABELLWAIT_FOR_NEW_FE:
0156 39368400			
0156 39368440			FLOOP AROUND UNTIL THERE IS A NEW FE TO SERVICE
0150 7482 611 JE LABEL_WAIT_FOR_NEW_FE 612 613 \$7WO POINTERS ARE NOT EQUAL, POSSIBLE NEW FE'S 614 615 616 \$CHECK SERVICE COUNTER 0150 8338820000 617 CMP FE_SERVICE_COUNTER, 0			
612 613 \$TWO POINTERS ARE NOT EQUAL, POSSIBLE NEW FE'S 614 615 616 \$CHECK SERVICE COUNTER 617 CMP FE_SERVICE_COUNTER, 0			
613	015F 74Fa		JE LABEL_WAIT_FOR_NEW_FE
614 615 616 \$16 \$16 \$17 CMP FE_SERVICE_COUNTER 0			
615 615 615 616 FCHECK SERVICE COUNTER 617 CMP FE_SERVICE_COUNTER, 0			FIND POINTERS ARE NOT EQUAL, POSSIBLE NEW FE'S
616 FCHECK SERVICE COUNTER 0150 8338820000 617 CMP FE_SERVICE_COUNTER, 0			
0150 8338820000 617 CMP FE_SERVICE_COUNTER, 0			ADUSON OFFILIAS ADUSTED
	A+E: CE75000000		
JE LABELLFINLERRUK FINE SERVICE FIR AND			[1] 사용하다는 경우 12 10 10 10 10 10 10 10 10 10 10 10 10 10
619 SINPUT POINTER(SI) ARE NOT	V101 /4/6		
619 \$INPUT POINTER(SI) ARE NOT 620 \$EQUAL, SERVICE COUNTER = 0			
621 SYNC ERROR			
622			FOIRC ERROR
623			
624 #SERVICE COUNTER >=1 , NORMAL CONDITION			SERVICE COUNTER >=1 . NORMAL CONDITION
625			SOUNTED BOOKIEN S-1 S ROMINE COMPITION
0163 833E820008 626 CMP FE_SERVICE_COUNTER, B	0163 8334820008		CMP FF SERVICE COUNTER. 8
0168 7375 627 JAE LABEL_PTR_ERROR \$THERE ARE MORE THAN 4 PAIRS			
628 \$OF FEX TO BE SERVICED, CPU			
629 FIS RUNNING TOO SLOW			
630			
631 #1 =< SERVICE COUNTER < B, NORMAL CONDITION.			#1 =< SERVICE COUNTER < B, NORMAL CONDITION.
632 ; BEGIN SERVICE THE FE'S			
633			
634		634	
635		635	
636 +1 \$EJECT		636 +1	\$EJECT

LOC OBJ	LINE SOURCE	
	637	
016H		SERVICE_FE:
	639	MOVE THE NEW FEX INTO AX AND CORRESPONDING FSV TO CX
	640	
	641	FNEED TO USE THE BX REGISTER TO ACCESS THE FE_BUFFER
016A BB0E8400	642	MOV CX, FE_SERVICE_PTR
	643	
016E FA	644	CLI
016F 87CB	645	XCHG CX, BX
0171 8B4730	646	MOV AX, FE_BUFFEREBX]
0174 87CB	647	XCHG CX, BX
017e Fk	648	STI
	640	
0177 833E850000	650	CMP FE_TYPE, FE_LOW
0170 7520	651	JNE LABEL_FE_HIGH
	652	
	653	
	654	
	655	
AAR ALAERSA	456	IT IS FEL TO BE PROCESSED
017E 8B0E7806	657	MOV CX, FSVL
0182 E85D(0)	658	CALL APCMQ
	659	ADPOUNT FOR THE BY WORT THE BY
	660 661	FRESULT FSV IN DX, XOUT IN CX
0185 89167800	662	HOV FSVL, DX SAVE THE NEW FSVL
V100 (0/10/00V	663	STORE THE NEW XOUT INTO THE FE_BUFFER
0189 A18400	664	MOV AX, FE_SERVICE_PTR
0180 FA	665	CLI
018F 93	666	XCHG AX, BX
	667	
018E 894F30	668	MOV FE_BUFFERCBX], CX
0191 FF0E8200	659	DEC FELSERVICE_COUNTER
	670	PEO FEIGURA EL CONTENTANTO DE CONTEN
0195 93	671	XCHG AX, BX
0196 FB	672	STI
	673	
	674	SUPDATE THE FE TYPE TO BE SERVICED NEXT
0197 C76685000106	675	MOV FE_TYPE, FE_HIGH
	676	
	677	
	678	FINCREMENT THE SERVICE POINTER, AX CONTAINS FE_SERVICE_PTR
019D 050200	679	ADD AX, TYPE FE_BUFFER
01A0 3B4000	680	CMP AX, LEN_FE_BUFFER * TYPE FE_BUFFER ;END OF BUFFER?
01A3 7331	681	JAE LABEL_WRAP_AROUND
01A5 A38400	682	MOV FELSERVICE_PTR, AX
01A8 EBAC	683	JMP LABEL_WAIT_FOR_NEW_FE
	684	
	685 +1 \$EJECT	

LOC	DEJ	LINE	SOURCE				
		686					
		687		47115 57			
01AA			LABEL_FE_HIGH:		EH IS TO BE PROCESSI	ED.	
	8B0E7A00	689			CX+ FSVH		
UIAE	E84160	690		LALL	APCNQ		
		691		.05011	T FOIL TAL DV VOUT TO	N CV	
		692		, KESUL	T FSV IN DX, XOUT II	N. CX	
****	55415400	693		WOU	FRUIT BY	ADAME THE MEN P	OUL
VIEL	89167A00	694		MUV	FSVH, DX	SAVE THE NEW F	SVH
		695					
		696			THE NEW XOUT INTO		
0165	A18400	697		MOV	AX, FE_SERVICE_PT	R	
		698					
0186		699		CLI	AV WW		
01B9	75	700		XCHG	AX, BX		
	20.534	701		WOU	EL BREEFERSAS ON		
	894F30	702		YOU	FE_BUFFER[BX], CX		
Olou	FF0E8200	703		DEC	FE_SERVICE_COUNTE	K	
		704		VOUG	AV EV		
0101		705		XCHG	AX, BX		
0102	t b	706		STI			
		707		ALIDEAT	- TUE TVD- TO DE	OFFILTOFT MEVT	
0.07	530.0.0000000	708			E THE FE TYPE TO BE	PEKATCEN MEYI	
0163	E70686000000	709		MOV	FELTYPE, FELLOW		
		710					
		711		4711000	WENT THE DEDUTEE DO	THEF	
		712			MENT THE SERVICE PO		
	05020/	713		ADD	AX, TYPE FE_BUFFE		
	384000	714		CMF		* TYPE FE_BUFFER	JEND OF BUFFERY
	7305	715		JAE	LABEL_WRAF_AROUND		
0101	A38400	716		VOM	FE_SERVICE_PTR, A	X	
		717		11/5	1 45F) 11477 FAR HE		
011/4	EB80	718		JMF	LABEL_WAIT_FOR_NE	W_FE	
		719					
****		720	LABEL MOAD ADD	DIT.			
0106		721	LABEL_WRAF_AROL		OF DOTATED HAS DOTA	TING TO THE LAST ELEME	אד דא דד
		722					MI IN FE
		723		THANAT	, MOVE TO THE FIRST	ELEMENI.	
	070/04000000	724		WOU			
	C70684000000	725		MOV	FE_SERVICE_PTR, 0		
ClbC	E977F#	726		JMF	LABEL_WAIT_FOR_NE	W_FE	
		727	AF 1507				
		728 +1	\$EJEU:				

LDC OFJ	LINE	SOURCE
	729	
01DF	736	LABEL_PTR_ERROR:
	731	
	732	;*************************************
	733	* THIS PART OF CODE HANDLES THE SYNCHRONIZATION ERROR *
	734	* BETWEEN THE QUANTIZATION PROCESS AND THE VARIOUS
	735	# INTERRUPT DRIVEN I/O PROCESSES. *
	736	II The state of th
	737	** AN ERROR CONDITION MAY BE ONE OR MORE OF THE FOLLOWINGS: *
	738	18
	739	## (1) BOTH INPUT(REGISTER SI - DEDICATED) AND SERVICE POINTER #
	740	* ARE POINTING TO DIFFERENT ELEMENT IN THE FE_BUFFER *
	741	* AND THE SERVICE COUNTER HAS A VALUE OF ZERO. *
	742	# (2) THE SERVICE COUNTER HAS A VALUE HIGHER THAN B. THIS *
	743	## MEANS THAT FOUR PAIRS OF FEX OR MORE ARE NOT QUANTIZED *
	744	¥ YET.
	745	* THE MULDEM SIMULATOR PROCESS WILL CONTINUE ONCE THE *
	746	** ABOVE ERROR(S) OCCURS. THE FOLLOWING ROUTINE WILL CAUSE THE *
	747	## HOST SIGNIFICANT ERROR_LED TO INCREMENT BY 1. *
	748	;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
	745	
	750	
DIDE FA	751	CLI ;DISABLE EXTERNAL INTERRUP
01E0 A17000	752	MOV AX,LED_DISPLAY_VALUE
01E3 80E410	753	ADD AH:10H
01E6 7297	754	JC LABEL_PTR_EXT
61E8 A37000	755	HOV LED_DISPLAY_VALUE,AX
01EF 0404	756	OR AL, AH
01ED E664	757	OUT ERROR_LED;AL
CIEF E98AFT	758	LABELLETRLEXT: JMP ERRORLENTRY
	759	
	760	÷*************************************
	761	# END OF MAIN PROGRAM *
	762	;*********************
	763	
	754	
	765	
	766 +	+1 \$EJECT

LOS OBJ

01F1

LINE	SOURCE
767	
768	
769	AFCHQ PROC NEAR
770	
771	
772	;*************************************
773	* PURPOSE:
774	** THIS ROUTINE QUANTIZES A SAMPLE USING AN ADAPTVE QUANTIZER.
775	ix
776	1
777	** DESCRIPTION:
778	** THE INPUT SAMPLE IS QUANTIZED, USING THE GIVEN SCALING FACTOR
779	FOR THE QUANTIZER. THE SCALING FACTOR IS UPDATED ON RETURN.
780	ix
781	**
782	* PARAMETERS:
783	# INFUT:
734	** XIN - INPUT SAMPLE(PASSED IN AX REGISTER)
785	** FSV - SCALING FACTOR FOR THE QUANTIZER. THIS VALUE IS
788	# UPDATED ON OUTPUT. (INPUTED IN CX REGISTER)
787	ix
788	i* OUTFUT:
789	* XOUT - OUTPUT QUANTIZED SAMPLE (PASSED IN CX REGISTER)
790	* FSV - NEW SCALING FACTOR FOR THE QUANTIZER (OUTPUT
791	* IN DX REGISTER)
792	ix
793	FX PRE-SPECIFIED:
754	** NLEV - NUMBER OF POSITIVE QUANTIZER LEVELS. THE QUANTIZE
795	is assumed to be symmetric about zero. The TOTAL
796	** NUMBER OF QUANTIZER LEVELS IS 2*NLEV.
797	**
798	** ROUTINES REQUIRED:
799	* IQUANTZ - QUANTIZE A POINT
800	
801	
802 41	\$EJECT

LOC ORJ	LINE SO	URCE			
	803 8 04				
01F2 890E7600	805 806		DV	FSV, CX	FREGISTER CX. SAVE FOR FUTURE REFEREN
	CE				
	807	OHANTZ			FTHIS VALUE IS ALSO USED IN PROCEDURE
		QUANTZ			
	808		VT11 TC	DAGGED TO THEO DECORES	USE TH PERTATER AN
	809 810		Y14 12	PASSED TO THIS PROCEDU	UKE IN REDISTER AX.
01F6 C70688000000	811	u	OV	SIGN_FLAG, POSITIVE	ATMITTALITY THE CICH FLAC
V1F0 L/V0000VVVVVV	812	п	UV	SIGN_FLHO; FUSITIVE	FINITIALIZE THE SIGN FLAG
O1FC A9FFFF	P13	т	EST	AX. OFFFFH	GET SIGN
01FF 7908	814		The second second	LABEL_1	FIT IS A POSITIVE NUMBER
V111 / 700	315		mo.	LMDCL_I	711 15 H FUSITIVE RUNDER
	816		21 MIY	A NEGATIVE NUMBER, GET	T ADECULITE HALLIE
	817		VIII 12	H REDNITTE ROMBERT DE	HESOLUTE VALUE
0201 F778	818	N	EG	ΑX	
0203 C70688000100	819		IOV	SIGN_FLAG, NEGATIVE	STORE STATE OF SIGN
	820				
0209.437200		PEL_1: M	טט	ABS_XIN, AX	\$SAVE THE ABSOLUTE XIN FOR
	822				JUSE IN IQUANTZ PROCEDURE
	823				
	824				
	825	. ;	REGISTE	R AX. CX AND DX ARE NO	OT PRESERVED IN THIS CALL.
020L E86700	626		ALL	IQUANTZ	or reserved in miss one.
	827				
	828	;	RESULT	"L2" RETURNED IN CX. 1	THIS VALUE IS DOUBLED THE
	E29			VALUE TO FACILITATE I	
	830				
020F 890E8000	831	н	GV	L2, CX	SAVE FOR FUTURE REFENRECE
	832				
	833 41 \$E	JECT			

334	
######################################	
B36	
837 838 MOV AX, FSV 839 0216 3D7800 840 CMP AX, FSVTH 841 0219 730E 842 JAE LABEL_MID_RISE 843 844 FSV < FSVTH 845 846 0218 63F902 847 CMF CX, 2 FSVTH 848 021E 7509 849 JNE LABEL_MID_RISE 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850 850	
0213 A17600 838 MOV AX, FSV 839 0216 3D7800 840 CMP AX, FSVTH 641 0219 730E 842 JAE LABEL_MID_RISE 843 844 FFSV < FSVTH 845 846 021E 63F902 847 CMF CX, 2 \$L2 = 1? 848 021E 7509 849 JNE LABEL_MID_RISE 850	
0216 3D7800 B40 CMP AX, FSVTH 0219 730E 842 JAE LABEL_MID_RISE 843 844 *FSV < FSVTH 845 846 021E 63F962 847 CMF CX, 2 *L2 = 1? 848 021E 7509 849 JNE LABEL_MID_RISE 850	
0216 3D7800 B40 CMP AX, FSVTH 641 0219 730E 842 JAE LABEL_MID_RISE 843 844 FSV < FSVTH 845 846 021E 63F962 847 CMF CX, 2 \$L2 = 1? 848 021E 7509 849 JNE LABEL_MID_RISE 850	
0219 730E 842 JAE LABEL_MID_RISE 843 844 \$FSV < FSVTH 845 846 021E 63F962 847 CMF CX; 2 \$L2 = 1? 848 021E 7509 849 JNE LABEL_MID_RISE 850	
0219 730E 842 JAE LABEL_MID_RISE 843 844 \$FSV < FSVTH 845 846 0218 63F902 8A7 CMF CX; 2 \$L2 = 1? 848 021E 7509 849 JNE LABEL_MID_RISE 850	
843 844	
844	
845 846 0218 63F902 847 CMF CX, 2 \$L2 = 1? 846 0218 7509 849 JNE LABEL_MID_RISE 850	
846 021E 83F962 847 CMF CX, 2 \$L2 = 1? 846 021E 7509 849 JNE LABEL_MID_RISE 850	
021E 63F902 847 CMF CX, 2 \$L2 = 1? 846 021E 7509 849 JNE LABEL_MID_RISE 850	
848 -021E 7509 849 JNE LABEL_MID_RISE 850	
021E 7509 849 JNE LABEL_MID_RISE 850	
850	
852	
0220 E7067400:000 853 MOV XOUT, 0	
0226 EB1B90 854 JMP INCLSTEP .	
855	
856	
0229 857 LABEL_MID_RISE:	
858	
859 FNEED TO USE BX AT THIS POINT, DISABLE INTERRUPT BEFORE USE	USE
850 FRX IS USED BY INTERRUPT I/O ROUTINES AS A DEDICATED REGIST	
861	
0225 FH 862 CLI	
022A 87B9 863 XCHG BX, CX ;CX CONTAINS 'L2'	
864	
022C 2EF727 865 MUL CS:YQERX3 FAX CONTAINS FSV	
866	
022F 87D9 867 XCHG BX CX	
868	
0231 FB 869 STI	
870	
0232 D100 871 ROL AX, 1 #SCALE BY 2##15	
0234 D1D2 872 RCL DX 1	
873	
874 FINCORPORATE SIGN BIT	
0236 833E880000 875 CMF SIGN_FLAG, 0	
023B 7402 876 JZ LABEL_2	
877	
878 FXIN WAS A NEGATIVE NUMBER	
879	
023D F7DA 880 NEG DX	
881	
023F 89167400 882 LABEL_2: MOV XOUT, DX	
883	
884 +1 \$EJECT	

LOC DRJ	LINE	SOURCE			
	885				
	886		****	*********	***
	887		# INC	CREMENT STEP SIZE OF F	SV *
	888		****	*******	***
	889				
	890				
0243 8F0E8000	891	INC_STEF:	HOV	CX, L2	
	892				
	893		FNEED	TO USE BX AGIAN, REFE	R TO PREVIOUS USAGE FOR DOCUMENTATION
	894				
0247 FA	895		CLI		
0248 87CP	895		XCHG	CX, BX	
0248 28884743	897		HOV	AX: CS:RMLTEBX]	
024E 87CB	898		XCHG	CX, BX	
0250 FE	899		STI		
0251 F7267600	700		MUL	FSV	
	901				
0255 B100	902		ROL	AX, 1	DIVIDE RESULT BY 2**14
0257 P1P2	903		RCL	DX, 1	
0259 B100	904		ROL	AX, 1	
0255 P102	905		RCL	DX, 1	
	906				
0250 81FA401F	907		CMP	DX, FSVMX	
0261 7505	908		JBE	LABEL_3	
0263 Be401F	909		KOV	DX, FSVMX	
0266 EB0990	910		JMP	APCMQ_END	
0269 83FA50	711	LABEL_3:	CMP	DX. FSVNN	
0260 7303	912		JAE	APCMQ_END	
026E BA5000	913		KOV	DX. FSVMN	
	914				
	915		INFH F	SV IN DX	
	916		711.	0, 1, 1,	
0271 BB0E7400	917	APOMQLEND:	MOV	CX, XDUT	
0275 C3	918	III CHILLEND	RET	ONT NOOT	
	919				
	920	APCHQ	ENDP		
	921	in ond	LILDI		
	922				
	923				
		\$EJECT			
	127 11	+20201			

LOC OFJ

LINE	SOURCE			
925				
926	IQUANTZ		PROC	NEAR
927				
928	***	*******	******	**********************
929	**	INFUT :	ABS_XIN	, FSV PASSED FROM MEMORY *
930	**			*
931	11	OUTPUT:	L	IN REGISTER CX. THIS VALUE IS *
932	;*			DOUBLE THAT OF ACTUAL VALUE TO *
933	;*			FACILITATE INDEX ADDRESSING *
934	**	REGISTER US	AGE:	1
935	**		BX	TEMPORARILY, VALUE SAVED AND *
936	;*			RESTORED BY THIS PROCEDURE. *
937	**			INTERRUPT DISABLED WHILE USING THIS*
938	**			REGISTER. *
939	1%		DX	*
940	;*			*
941	ix.	REGISTER AS	SIGNMENT IN	THIS PROCEDURE: *
942	**			*
943	;*		CX	CONTAINS IU *
944	**	REF	ER TO FORTRA	N LISTING FOR DESCRIPTION OF THESE *
945	**	VAR	IABLES.	
945	;*)	*******	********	****************
947				
548				
949 +	1 \$EJECT			

1003

LOC	08J	LINE	SOURCE			
		950				
0276	E7057E000000	951		MOV	IL, 0	;IL = 0
0270	B92000	952		VOM	CX, NLEV* TYPE XQ	FIU = NLEV [* 2 FACTOR ADDED]
		953				
		954		#I = (I	L + IU) /2	
		955				
027F	8B167E00	956	LABEL_100:	MOV	DX, IL	
0283	0371	957		ADD	DX, CX	
	DIEA	958		SHR	DX, 1	DIVIDE BY 2
	81E2FEFF	959		AND	DX. OFFFEH	MAKE SURE IT IS AN EVEN NUMB
			ER			
628B	89167000	960		MOV	I, DX	SAVE I FOR FUTURE USE
		961			.GT. XQ(I)*FSV) GOTO 220	YOUNG THE TENTONE OBE
		962		741 17	1011 /4(1/4:00) 0018 220	
		953		SAIFED T	O USE REGISTER BX AT THIS POI	NT DICADIF INTERDUPT DEFORE
		964		JUSING	D DOE REDISTER BY HI INTO LOT	MI - DISHBLE INTERRUFT BEFURE
				102140		
		965				
		966				
0285		967		CLI		
	87D3	968		XCHG	DX, BX	
	26884722	969		MOV	AX, CS:XQ[BX]	MOV XQ TO AX REGISTER
	8703	970		XCHG	IX, PX	
0298	Fb	971		STI		
		972				
	F7267600	973		MUL	FSV	FXQ IS SCALED BY 2**15
	B100	974		ROL	AX, 1	
	61102	975		RCL	DX, 1	
02A1	39167200	976		CMP	ABS_XIN, DX	
0245	7767	977		JA	LABEL_200	
		978				
		979		; IU = I		
02A7	8B0E7C00	980		MOV	CX, I	
02AB	EB0990	781		JMP	LABEL_300	
		982				
02AE		983	LABEL_200:	;IL = I		
	8B167C00	984		KOV	DX, I	
	89167E00	985		HOV	IL, DX	
		986			IL, DA	
*		987				
02BA	8B167E00	988	LABEL_300:	MOV	DX. IL	
	830202	989	LIDELLOVVI	ADD	DX: 2	#IL+ 1
· 0285		990		CMP	CX, DX	JILT I
	77BE	991		JA	LAFEL_100	
VZDI	// DE	992		JH	LHEEL_100	
				4DE0111 T	TH TH OV DESIGNED	
		993		FRESULI	IU IN CX REGISTER	
		994				
		995				
0201	C3	996		RET		
		997				
		998	IQUANTZ	ENDP		
		999				
		1000				
		1001				
		1002				
		1007				

100	UND	LINE	SOURCE
		1004	
		1005	
		1006	
		1007	CODE
		1008	
		1009	
		1010	
		1011 +1	\$EJECT

ENDS

.0C 02J	LINE	SOURCE	
	1012		
	1013		
	1014		
	1015		
	1016	INTERRUPT_CODE	SEGNENT
	1017		
	1018	ASSUME	CS:INTERRUPT_CODE, DS:DATA, ES:NOTHING, SS:STACK
	1019		
000	1020	SOF_INTERRUPT	PROC FAR
	1021		
	1022	***********	*********************
	1023		DUTINE IS ACTIVATED BY THE SOF_(START_OF_FRAME) SINGNAL
	1024		THE SOF_ ON THE DECODER BUS IS SYNCHRONIZED AND OCCURS
	1025		AS THE DECODER BUS SOF_).
	1026		RESPONSIBLE FOR CHECKING THE MULDEM OVERALL FIRMWARE
	1027	IX STATE: INCLUDING	
	1026		INPUT AND DUTPUT POINTER VALUES
	1029		REDICTOR COEFFICIENTS BUFFER FRAME AND OFFSET POINTER
	1030		ALUES
	1031	;*	
	1032	# IF ANY OF THESE !	VALUES IS ABNORMAL, THIS ROUTINE WILL ATTEMP TO CORRECT
	1033		TS KNOWLEDGE, AT THE SAME TIME, IT WILL INCREMENT THE
	1634		AND OUTPUT TO THE LEAST SIGNIFICANT LED.
	1035		ALSO RESPONSIBLE FOR SETTING UP THE PRIDICTION COEFF
	1036		OR THE CURRENT FRAME.
	1037	; x	
	1038	FF RFG	ISTER USAGES:
	1039	11	AX, BX, BP, SI, DI
	1040	9%	
	1041	**********	····
	1042		
	1043		
		\$EJECT	

1094

1095

1096 1097 +1

1098 +1

1099 +1

0025 7407

0027 A37000

002A 0AC4

8085/8087/8088 MACRO	ASSEMBLER	MSLDEH			11:51:55 10/07/82 PAGE
LOC GBU	LINE	SOURCE			
	1045				
	1045		SHOTE THAT INT	ERRUPT FLAG (IF) IS DISA	RIED TH MOST PART
	1047		OF THIS ROUTI		DEED IN HOOF FRANCE
	1048		701 1110 10011	····	
0000 50	1049		PUSH	AX	MAY USE AX REGISTER TO
0000 00	1050		10011	nn.	OUTPUT ERROR_LED IN THIS
	1051				FROUTINE
	1052		:*********	******************	
	1053		* FE_BUFFE		
	1054			***********	
	1055		1444444444	***********	
	1056				
	1057				
			+CUECK	FE INPUT POINTER	
	1058 1059		FLHELI	TE INFUI FUINIER	
AAA4 D7554A			CMD	SI, 16	THIS IS THE NORMAL OFFSET
0001 83FE10	1060		CMP	51, 10	FVALUE POINTING TO FEL (4)
AACA 7447	1061		ır	LABEL_NORMAL_1	TVALUE PUINTING TO PEL(4)
0004 7413	1062 1063		JE	LHDEL_NUNNHL_1	
	1003				
	1065 +1				
	1065 +1				
AAA7 447000	1067 +1		MU.	AV. I ED DICOLAY HALLE	
0006 A17000	1068 +1		MOV	AX, LED_DISPLAY_VALUE	
0009 FEC0	1069 +1		INC	AL ALL	
000R 240F	1070 +1		ANI	AL, OFH	
0375 7465	1071			LED EDDAD EVI	
000D 7407	1072		JZ	LED_ERROR_EXT1	
	1073				
860E 477000	1074 +1		VOU	JED DIODI AV HALLE AV	
000F A37000	1075 +1		NOV	LED_DISPLAY_VALUE, AX	
0012 0AC4	1076 +1		OR SUIT	AL,AH	
0014 E664	1077 +1		DUT	ERROR_LED, AL	
804 / BELAND	1078	LED FOROD EVILL	VOII	07 4/	*PODDECT DOTATED HALVE
0016 BE1000	1079	LED_ERROR_EXT1:	MOV	SI, 16	CORRECT POINTER VALUE
0010	1036	LABEL MODERN 44	ACHECH	FE BUTDUT DOTUTED	
0019	1081	LABEL_NORMAL_1:	TUREUR	FE OUTPUT POINTER	
0040 07FD30	1082		CAD	DV A	THIS IS THE NORMAL VALUE
0 019 83FB00	1083		CMP	BX, 0	POINTING TO FEL(0)
0040 744	1084			LADEL MODWAL O	FOIRITHG TO PELCO
0010 7413	1085		JE	LABEL_NORMAL_2	
	1086				
	1087 +1				
	1088 +1				
	1089 +1				
AAT 447000	1090 +1		MEN	AV. I ED DTODI AV DALUE	
001E A17000	1091 +1		MOV	AX, LED_DISPLAY_VALUE	
0021 FEC0	1092 +1		INC	AL AL-AFU	放弃 国主国际政策
0023 240F	1093 +1		AND	AL+OFH	

LED_ERROR_EXT2

AL, AH

LED_DISPLAY_VALUE, AX

JZ

MOV

OR

LOC 083	LINE	SOURCE			
002C E664	1100 +1		OUT	ERROR_LED, AL	
	1101				
002E BB0000	1102 1103	LED_ERROR_EXT2:	MOV	BX, 0	CORRECT POINTER VALUE
	1104		;*********	********	ixx ·
	1105		# PREDICTOR	BUFFER CHECK	
	1106		*********	************	t**
	1107				
0031	1108	LABEL_NORMAL_2:	CHECK	PREDICTOR INPUT POI	NTER
	1109				
0031 83FF10	1110		CMP	DI, LEN_P_BUFFER_F	RAME * (TYPE PREDICTOR_BUFFER)
	1111				THIS IS THE NORMAL VALUE
	1112				POINTING TO ELEMENT AFTER
	1113				THE LAST ELEMENT OF A FRAME
	1114				
0034 7410	1115		JE	LABEL_NORMAL_3	
	1116				
	1_17		THE P	DINTER CONTAINS ILLE	GAL VALUE
	1118				
	1119				
	1126 +1				
	1121 +1				
	1132 +1				
	1123 +1				
0036 A17000	1124 +1		MOV	AX, LED_DISPLAY_VALU	JE
0039 FEE0	1125 +1		INC	AL	
03B 240F	1126 +1		ANI	AL, OFH	
AATE DAGE	1127				
003D 7407	1128		JZ	LABEL_NORMAL_3	
	1129				
003F A37000	1130 +1		VOII	LED BYDDI AV HALDE	
0042 0AC4	1131 +1 1132 +1		MOV	LED_DISPLAY_VALUE,	AX
0044 E664	1133 +1		OR OUT	AL,AH ERROR_LED, AL	
VV14 LUU1	1134		001	ENNUNLLED! AL	
0046	1135	LADEL MODEAL 74	*CFT T	UE ODERTOTOS DOTUTES	FOR MENT FRAME PURPLIE
0040	1136	LABEL_NORMAL_3:	13E1 1	HE PREDICTOR PUTNIERS	FOR NEXT FRAME OUTPUT
0046 BF0000	1137		MOII	DT. A	
0049 83E516	1138		MOV	DI, O	DAME #/TYPE DEEDTOTOR BUFFER
0047 B36310	1139		нпп	Dri LEM_F_BUFFEM_FI	RAME *(TYPE PREDICTOR_BUFFER) \$INCREMENT TO NEXT FRAME
	1140				FINCHEREN TO NEXT FRAME
004C 83FB20	1141		CMF	DD. I ACT D BHEFED I	FRAME #DETERMINE IF BP POINTS TO
VV 10 001 122	1142		Cin	DIT LASTLI LOUITENLI	LAST FRAME IN BUFFER
	1143				TEROT TRAILE IN BOTTER
004F 7603	1144		JBE	LABEL_SOF_END	
	1145				
	1146		FRF WA	S POINTING TO LAST FE	RAME IN BUFFER BEFORE INCREMENT
	1147				
0051 BD0000	1148		MOV	BF, 0	MOVE TO THE FIRST FRAME IN B
		UFFER			
	1149				
0054	1150 +1	LABEL_SOF_END:			
	1151 +1				
	1152 +1				
	1153 +1				

F00	050	LINE	SOURCE		
0054	FA	1154 +1		CLI	
0055	B8E000	1155 +1		MOV	AX, ENABLE_NORMAL_MASK
0058	E67A	1156 +1		OUT	PIC_PORT_1, AL
005A	FB	1157 +1		STI	
005B	58	1158		POP	AX
0050	CF	1159		IRET	
		1160			
		1161			
		1162	SOF_INTERRUFT	ENDP	
		1163			
		1164			
		1165			
		1166 +1	\$EJECT		

LOC ORJ	LINE	SOURCE			
	1167				
005D	1168 F	REDICTOR_OUTPUT_INTE	RRUPT	PROC	FAR
	1169				
	1170				
		**********	******	******	********
					N_ SIGNAL ON THE DECODER BUS. *
					THE PREDICTOR COEFFICIENTS FROM *
		* THE PREDICTOR COEF		OK CONTO	*
		*	1 DOLLEN		
			STER USAGES		
		*	AX, BI		
			HAT DI		
		*	******	*****	************
		***********	****	****	****
	1180				
	1131		PHEH	AV	AAV TO HOLD DIETHO THTO DOUT
0050 50	1182		PUSH	AX	AX IS USED DURING THIS ROUT
		NE.			
	1185 +1				
	1184 +1				
	1185 +1				
	1186 41				
DOSE BEFERV	1187 +1		VOM	AX, ENABLE	_SOF_MASK
0061 E67A	1186 +1		OUT	PIC_PORT_	.1, AL
063 FF	1189 +1		STI		
	1170				
	1171				
	1192		#CONFIG	URE THE DE	CODER DATA BUS TO BE OUTPUT
	1193				
0064 389999	1194		KOV	AX, D_DAT	A_BUS_OUT
0067 E74E	1195		OUT	DECODER_I	BUS_SPVR, AX
	1196				
	1197		FOUTPUT	DATA	
	1198				
0069 3EBB03	1199		MOV	AX, DS:PR	REDICTOR_BUFFER[BP][DI]
	1200				
006C E74A	1201		GUT	D_DATA_BL	JS, AX
	1202				
	1203		FINCREM	ENT BUFFER	POINTER
	1204				
006E E3C702	1205		ADD	DI, TYPE	PREDICTOR_BUFFER
0071 83FF10	1206		CMP		P_BUFFER_FRAME * TYPE PREDICTOR_BUFFER
0074 7203	1207		JB	LABEL_P_E	
0074 7200	1208				
	1209		ant por	NTS REVONT	END OF CURRENT FRAME, END OF OUTPUT
	1210				N - 2ND FRAME) PREDICTOR COEFFICIENTS.
	1211				TER FOR INPUT NEW PREDICTOR
					END OF CURRENT FRAME(FRAME N).
	1212				WILL BE DEPOSITED INTO THE SAME FRAME
	1213				RE BP REGISTER NEEDS NOT BE CHANGED.
	1214		FOURTE	יי וחבותברטו	VE DE KEDISTEN NEEDS NOT DE CHANGED.
AAD/ BEALL	1215		MOII	D7 A	
0076 BF0000	1216		MOV	DI, 0	
	1217				
	1218				
0079		LABEL_P_EN:			F OUTPUT CYCLE
0079 E54C	1220		IN	AY. D CO	NTROL_BUS

LCC OBJ	LINE	SOURCE	
0078 250400	1221	AND	AX, SPEN
007E 74F9	1222	JZ	LABEL_P_EN
	1223		
	1224	\$NOW	SPEN IS FALSE(I.E. SIGNAL IS HIGH), TAKE DATA OFF THE BU
		S	
	1225		
0080 B87R7B	1226	VOM	AX, D_DATA_BUS_IN
0083 E74E	1227	OUT	DECODER_BUS_SPVR, AX
	1228		
	1229 +1		
	1230 +1		
	1231 +1		
	1232 +1		
0085 FA	1233 +1	CLI	
0086 B8E000	1234 +1	VOM	AX, ENABLE_NORMAL_MASK
0089 E67A	1235 +1	DUT	PIC_PORT_1, AL
0083 FR	1236 +1	STI	
	1237		
0090 58	1238	POP	AX
008D CF	1239	IRET	
	1240		
	1241		
	1242	PREDICTOR_OUTPUT_INTERRUPT	ENDP
	1243		
	1244		
	1245		
	1246 +1	\$EJECT	

8086/8087/8088 MA	CRD ASSEMBLER	MSLDEM		11:51:55 10/07/82 PAGE 37
LOC GAU	LINE	SOURCE		
	1247			
	1248			
008E	1249	FE_OUTPUT_INTERRUPT	PROC	FAR
	1250	12201121121112111		
	1251			
	1252	;****************	*****	
	1253	* THIS INTERRUPT HAND	LER IS	ACTIVATED BY FE ENABLE_ SIGNAL ON THE DECODER *
	1254			DUTPUTING FEH AND FEL TO THE DECODER BUS #
	1255			X IS PUSHED ONTO STACK, DATA BUS PORT IS *
	1256	** CONVERTED FROM INPL	T STATE	TO OUTPUT STATE, THE APPROPRIATE FEX IS *
	1257	** OUTPUTED TO THE BUS	. POINT	ER TO BUFFER IS INCREMENTED TO NEXT FEX TO BE *
	1258			_OOF AROUND UNTIL FE ENDABLE_ IS HIGH, THEN *
	1259			THE DATA BUS, CONVERT PORT TO INPUT STATE *
	1260	* AND EXIT		
	1267	11		
	1262	* REGISTER USAGE:		
	1262	ix AX: B	Y	
	1754			*************
	1265	7, , , , , , , , , , , , , , , , , , ,		
	126e			
003E 50	1267		PUSH	AX
	1265			
	1265 +1			
	1270 +1			
	1271 +1			
	1272 +1			
DOSE BREECO	1273 +1		MOV	AX; ENABLE_SOF_MASK
0072 E67A	1274 +1		OUT	PIC_PORT_1, AL
0094 FB	1275 +1		STI	
00/- 15	1276		0.1	
	1277			
	1278		:CONF	IGURE DATA BUS PORT TO BE OUTPUT
	1279		700111	TOOLE THIN DOC TONE TO BE DON'T
0095 B89999	1280		MOV	AX, D_DATA_BUS_OUT
0098 E74E	1281		OUT	DECODER_BUS_SPVR, AX
VV/0 L/12	1282			PEOPERATOR III
009A 8B4730	1283		MOV	AX, FE_BUFFER[BX]
00711 22 17 20	1294			
	1285		IRX I	S THE POINTER TO THE NEXT WORD TO BE DUTPUT IN FE BUFFER
	1286			
009D E74A	1287		OUT	D_DATA_BUS, AX
OV/D E/AR	1288		001	
1	1289		: TNCP	EMENT THE OUTPUT POINTER
	1290		THUN	EILERT THE BOTT OF TOTALER
009F 83C302	1291		ADD	BX, TYPE FE_BUFFER
00A2 83FR40	1292		CMP	BX, LEN_FE_BUFFER * TYPE FE_BUFFER
00A5 7203	1293		JB	LABEL_FE_ENABLE
	1273		MOV	BX, 0
00A7 BB0000	1274		nuv	DAT V

AX. D_CONTROL_BUS

LABEL_FE_ENABLE

AX, SFEEN

IN

AND

JZ

1295 1296 1297

1298

1299

1300

1301

DOAA ESAC

00AF 74F9

00AC 251000

LABEL_FE_ENABLE:

LOC OBJ	LINE SOURCE	
	1302	FNOW SFEEN_ IS FALSE(SIGNAL IS HIGH), TAKE DATA OFF BUS
00F1 B8959B	1303	MOV AX; D_DATA_BUS_IN
00P4 E74E	1304	OUT DECODER_BUS_SPVR, AX
	1305	
	1306 +1	
	1307 +1	
	1308 +1	
	1309 +1	
00B5 FA	1310 +1	CLI
00b7 B8E006	1311 +1	MOV AX, ENABLE_NORMAL_MASK
00BA E67A	1312 +1	OUT PIC_PORT_1, AL
	1313 +1	STI
OCBS FA	1314	JII
0077 50		POP AX
00ED 58	1315	rur HA
0/05 05	1316	IDET
OOBE CF	1317	IRET
	1318	
	1319 FE_DUTFUT_INTERRUPT	ENDP
	1320	
	1321	
	1322	
	1323 +1 \$EJECT	

LOC	OBJ	LINE	SOURCE				
0000	83FE40	1377 1378		CMP	SI, LEN_FE_BUFFER * TYPE	E FE_BUFFER	SEND OF BUFFE
			R?				
0000	7203	1379		JB	LABEL_FE_INPUT_END		
OODE	BE0000	1380		MOV	SI, 0	FRESET POINTER	TO BEGINNING
		1381				FOF CIRCULAR BU	JFFER.
		1382					
00E3		1383 +1	LABEL_FE_INPUT_END:				
		1384 +1					
		1385 +1					
		1386 41					
00E1	FA	1387 41		CLI			
00E2	B8E000	1388 +1		MOV	AX, ENABLE_NORMAL_MASK		
00E5	E57A	1389 +1		OUT	PIC_PORT_1, AL		
00E7	FE	1390 +1		STI			
00E8	58	1391		POP	AX		
		1392					
00E9	CF	1393		IRET			
		1394					
		1395	FE_INPUT_INTERRUPT	ENDP			
		1396					
		1397					
		1398 +1	SEJECT				
		10/0 11	TECEU				

LOC OBJ	LINE	SOURCE			
	1399				
	1400				
	1401				
00EA	1402	PREDICTOR_INPUT_INTERRU	IPT	PROC FAR	
	1403				
	1404				
	1405				
	1406	; ************ ***	*****	********	******
	1407			BY THE PEN_ SIGNAL ON THE COD	
	1408			FFICIENTS FROM THE CODER BUS AN	
	1409	## DATA IN THE PREDICT			1
	1410	18			1
	1411	ix REGIST	ER USAGE	S:	
	1412	18	AX: DI		1
	1413	i*			*
	1414		*****	************	******
00EP 50	1415		PUSH	AX	
V	1416		10011	na .	
	1417 +1				
	1418 +1				
	1415 +1				
	1426 +1				
OOER BEFECO	1421 +1		MOV	AX, ENABLE_SOF_MASK	
00EF E676	1422 +1		OUT	FIC_PORT_1: AL	
OOFO FE	1423 +1		STI	FIG_FORT_IF HL	
001012	1424		011		
	1425		STEET	FOR P CLOCK VALID	
	1426		FIEDI	FOR F CEDER VALID	
	1427				
00F1 E544	1428	LABEL_P_CLK:	IN	AX, C_CONTROL_BUS	
VV. 1 LU77	1429	LWDCTT TOTAL	714	HAT CLCONTROLLEDS	
00F3 252000	1430		AND	AX, CPCLK	
00F6 74F9	1431		JZ	LABEL_P_CLK	
0070 7417	1432		32	LHDELLFLOLK	
	1433				
	1434		*D CI C	CK IS VALID, READ IN PREDICTOR	COEFFICIENTS FROM
	1435		; DATA		COEFFICIENTS FROM
			TUHTH	000	
	1436 1437				
00F8 E542			TN	AV. C BATA DIE	
VVFD E342	1438 1439		IN	AX, C_DATA_BUS	
	1437		: DD CD	ECIFIES THE FRAME STARTING OFFS	ET(0, 14, 72 00 40)
	1441			PECIFIES THE OFFSET WITHIN EACH	
00FA 3E8903	1442		MOV	DS:PREDICTOR_BUFFER[BP][DI];	
VOFH 3E67V3	1443		nuv	D3.FREDICION_BUFFERLBFILDII)	HA
00FD 83C702	1444		ADD	DI, TYPE PREDICTOR_BUFFER	FINCREMENT OFFSET
OUT 1 630/VZ			ADD	DIT THE PREDICTOR_BUFFER	
	1445				FVALUE WITHIN ONE FRA
		ME			
	1446				
	1447				
	1448 +1				
	1449 +1				
	1450 +1				
	1451 +1				
0100 FA	1452 +1		CLI		

LOC	05.1	LINE	SOURCE		
0101	BSE000	1453 +1	MOV	,	AX, ENABLE_NORMAL_MASK
0104	E67A	1454 +1	OUT	T	PIC_PORT_1, AL
0106	FB	1455 +1	STI	I	
		1456			
0107	58	1457	POF	•	AX
0108		1458	IRE		
		1459			
		1460	PREDICTOR_INPUT_INTERRUPT		ENDF
		1461			
		1462			
		1453			
		1464 +1	\$EJECT		

LO: OR.	LINE	SOURCE			
	1465				
0109	1465	EXCEPTION_INT	PROC	FAR	
	1467				
	1468	*******	*******	*****************	*****
	1469	# THIS IS	THE EXCEPTI	ON HANDLER FOR THE 8086 CPI	U. *
	1470	# UNDER NO	DRMAL OPERAT	ION ENVIRONMENT, THIS CODE	SHOULD *
	1471	# NEVER B	E EXECUTED.		1
	1472	# IN THE I	UNLIKELY EVE	INT THAT THIS HANDLER IS AC	TIVATED,*
	1473	# IT SETS	THE LED_DIS	SPLAY_VALUE TO OFFH AND OUT	PUT #
	1474	# TO THE !	ERROR_LED.		1
	1475	18	REGIST	TER USAGES:	1
	1476	18		AX,	1
	1477	18			1
	1472	;********	********	**************	*****
	1470				
	1480				
	1491	#EXTERNAL	INTERRUPTS A	RE DISABLED DURING EXECUTION	ON OF THIS HANDLE
	1482				
	1400				
	1963				
0109 50	1983		PUSH	AX	
			PUSH MOV	AX AX, OFFFFH	
010A BSFFFF	1083 1484			AX AX, OFFFFH LED_DISPLAY_VALUE, AX	
010A B8FFFF 010D A37000	1483 1484 1485		MOV	AX, OFFFFH	
010A B8FFFF 010D A37000 0110 E664	1663 1484 1485 1485 1487		MOV MOV OUT	AX, OFFFFH LED_DISPLAY_VALUE, AX ERROR_LED, AL	
010A B8FFFF 010D A37000 0110 E664 0112 58	1484 1485 1485 1485 1487		MOV MOV OUT POP	AX, OFFFFH LED_DISPLAY_VALUE, AX	
010A B8FFFF 010D A37000 0110 E664 0112 58 0113 FF	1484 1485 1485 1485 1487 1488		MOV MOV OUT POP STI	AX, OFFFFH LED_DISPLAY_VALUE, AX ERROR_LED, AL	
010A B8FFFF 010D A37000 0110 E664 0112 58 0113 FF	1484 1485 1485 1485 1487 1488 1487		MOV MOV OUT POP	AX, OFFFFH LED_DISPLAY_VALUE, AX ERROR_LED, AL	
010A B8FFFF 010D A37000 0110 E664 0112 58 0113 FF	1484 1485 1485 1485 1487 1488		MOV MOV OUT POP STI	AX, OFFFFH LED_DISPLAY_VALUE, AX ERROR_LED, AL	
010A B8FFFF 010D A37000 0110 E664 0112 58 0113 FF	1484 1485 1485 1486 1487 1488 1487 1490 1491 1492	EXCEPTION_INT	MOV MOV OUT POP STI	AX, OFFFFH LED_DISPLAY_VALUE, AX ERROR_LED, AL	
010A B8FFFF 010D A37000 0110 E664 0112 58 0113 FF	1484 1485 1485 1485 1487 1487 1487 1490 1491	EXCEPTION_INT	MOV MOV OUT POP STI IRET	AX, OFFFFH LED_DISPLAY_VALUE, AX ERROR_LED, AL	
010A B8FFFF 010D A37000 0110 E664 0112 58 0113 FF	1484 1485 1485 1487 1488 1487 1490 1491 1492 1493	EXCEPTION_INT	MOV MOV OUT POP STI IRET	AX, OFFFFH LED_DISPLAY_VALUE, AX ERROR_LED, AL	
010A B8FFFF 010D A37000 0110 E664 0112 58 0113 FF	1484 1485 1486 1487 1488 1487 1490 1491 1492 1493 1494	EXCEPTION_INT	MOV MOV OUT POP STI IRET	AX, OFFFFH LED_DISPLAY_VALUE, AX ERROR_LED, AL	
0107 50 010A B8FFFF 010D A37000 0110 E664 0112 58 0113 FF 0114 CF	1484 1485 1486 1487 1488 1487 1490 1491 1492 1493 1494	EXCEPTION_INT	MOV MOV OUT POP STI IRET	AX, OFFFFH LED_DISPLAY_VALUE, AX ERROR_LED, AL	

ASSEMBLY COMPLETE. NO ERRORS FOUND

B3. NO RESIDUAL QUANTIZER

LOC OBJ

VAX/VMS 8086/8087/8088 MACRO ASSEMBLER V1.0VX ASSEMBLY OF MODULE MOLDEM OBJECT MODULE PLACED IN MOLDEM.OBJ NO INVOCATION LINE CONTROLS

> LINE SOURCE 3 NAME MOLDEN 9 * PROJECT: 472A RELP CODEC 10 11 # BOARD: ** DEVICE: MULDEM SIMULATOR 2 FIRHWARE EPROMS (2732A-2) 12 13 ## BOARD LOCATION: E15, E29 * PROGRAMMABLE 14 ## DEVICE NUMBER: 60-0670, 60-0671 16 17 19 19 41 \$EJECT

2 160 03J LINE

MOLDEM

SOURCE

	20				
	21				
	22				
	23	****************** *	******	*****	******
	24	** THIS FIRMWARE	IS IDEN	TICAL TO	THE MOLDEM VERSION EXCEPT *
	25	# IN THE FOLLOW	ING AREAS	:	1
	26	11			
	27		FEX COFF	FICIENT	S ARE NOT QUANTIZED. *
	28				S IMPLEMENTED IN SUCH A WAY *
	29				IS BELOW THE THRESHOLD *
	30				X VALUE WILL BE SUPPRESSED *
	31		ZERO.	, INC IL	A VALUE WILL BE SUFFRESSED *
	32			******	********
	33	**********	****	*****	*********
	34				
	35				
	36	A about 40 to 40 41 40 40 40 40 40 40	that always at the sheets	le de de de de de de de	******
	37		*****	******	*****************
	38	j*			
	36				E MULtiplexor/DEMulti- *
	40				YSTEM. REFER TO MDA *
	41	* DOCUMENT O	0 - 3035	- K01 F	UK DETAIL.
	42	i X			
	**	专来本来来来来来来	*****	****	**********
	44				
	45				
	40				
	47				
	49				
	49	;***************	***		
	50	# SYSTEM CONSTANTS	*		
	51	*************	***		
	52				
	53				
	54				
0020	55	LEN_FE_BUFFER	EQU	32	FACCOMODATE 16 PAIRS OF FEX
	56				
0005	57	LEN_P_BUFFER_FRAME	EQU	8	FEACH FRAME ACCOMODATES 8 PREDICTOR COEFFICIE
		NTS			
	58				
6020	59	LAST_P_BUFFER_FRAME	EQU	32	THE BASE OFFSET ADDRESS OF LAST P FRAME
0020	60	EHSTET EBOTT ENET WHILE	Lao	32	FROM THE BEGINNING OF P BUFFER, THIS VALUE
	61				#MUST BE IN MULTIPLE OF LEN_P_BUFFER_FRAME *
	01	TYPE P_BUFFER			THOSE DE IN HOLFIFEE OF LENGT-DOFFEN-FRANC 4
	/0	TIPE PLBUFFER			THE FRAME DELAY IS EQUAL TO 1 +
	62				(LAST_P_BUFFER_FRAME / 16)
	63				* TLHS!_F_BUFFER_FRHDE / 10/
	64				
000	65	EE HIOH	F0!!		THIS DEDDECTHE THE FELLO TYPE
0001	66	FE_HIGH	EQU	1	THIS REPRESENTS THE FELL'S TYPE
0000	67	FE_LOW	EQU	0	THIS REPRESENTS THE FEL'S TYPE
	68				
0002	69	NLEV	EQU	2	NUMBER OF QUANTIZER REGIONS
	70				
A000	71	FSVMN	EQU	10	FSVMN - MINIMUM SCALING FACTOR
	72				

LCC OFF	LINE	SOURCE			
6 3 E8	73 74	FSVMX	EQU	1000	FSVMX - MAXIMUM SCALING FACTOR
000F	75 76 77 78 79 80 81	FSVTH	EQU	15	FSVTH - THRESHOLD VALUE. IF THE SCALE FACTOR IS LESS THAN FSVTH, A MID-TREAD QUANTIZER IS USED INSTEAD FOR A MID-RISE QUANTIZER. THE MID-TREAD QUANTIZER USES ZERO AS AN OUTPUT LEVEL FINSTEAD OF YQ(1).
0000	82 63	POSITIVE	EQU	0	POSITIVE SIGN
1000	84 85 86	NEGATIVE	EQU	1	*NEGATIVE SIGN
	87 88	# HARDWARE DEPEND	ENT CONST	ANTS	
0042	89 90 91	C_TATA_RUS	EQU	42H	FOR THE 16 BIT CODER DATA BUS
0044	92 93 94	C_CONTROL_BUS	EQU	44 H	FOR THE 16 BIT CODER CONTROL BUS
0(148	95 96 97 98 99 100	CODER_BUS_SPVR	EQU	46 H	FOR THE CODER BUS(THE CONTROL BUS AND FTHE DATA BUS)
004a	101 102 103 104	DLDATALBUS	EQU	4 AH	FTHIS IS THE PORT(B) ADDRESS ON THE 8255'S FFOR THE 16 BIT DECODER DATA BUS
0040	105 106 107 108	D_CONTROL_RUS	EQU	4CH	FOR THE 16 BIT DECODER CONTROL BUS
0(·4E	109 110 1.1 112 113 114 115 116	DECODER_BUS_SPVR	EQU	4EH	FOR THE DECODER BUS(THE CONTROL BUS AND THE DATA BUS)
0064	118 119	ERROR_LED	EQU	64H	
0066	117 120 121 122	ERROR_BUS_SPVR	EQU	66H	•
	123 124	\$8259A PIC DEPE	NDENT CONS	STANTS	
007B 007A	125 126 127	PIC_PORT_0 PIC_PORT_1	EQU	78H 7AH	\$8259A PROGRAMMABLE INTERRUPT CONTROLLER \$CONTROL PORTS.

LOC DBJ	LINE	SOURCE			
0013	128	ICW1	EQU	13H	SSINGLE PIC, ICW4 NEEDED
0008	129	ICW2	EQU	08H	STARTING INTERRUPT VECTOR = 8
0003	130	ICW4	EQU	03H	#SFNM = 0, AEDI = 1, NON-BUFFERED MODE
	131				FSF/EN WILL HAVE INPUT = 1 =>MASTER
	132				#UFH = 1 =>8086/88 SYSTEM.
00FE	133	ENABLE_SOF_MASK	EQU	OFEH	#MASK OUT ALL EXCEPT
	134				\$SOF INTERRUFT(RO)
	135				
00E0	136	ENABLE_NORMAL_MASK	EDU	0E0H	#MASK OUT UNUSED
	137				FINTERRUPT(R7,R6,R5)
	138				
	139				
	140				
	141	#8255A PPI DEPEN	NTENT CON	TSTANTS	
	142				
9B9B	143	C_DATA_BUS_IN	EQU	9B9BH	FROGRAM THE CODER DATA BUS TO BE INPUT, ALL
	144				FOTHER PORTS ON CODER BUS ARE INPUT.
	145	VID PRO			#MSB AND LSB OF THIS WORD EACH ADDRESSES TO 0
		NE PPI			
0000	146	O DATA DUO DUT	FOU	000011	ADDOODAY THE BOTES BATA BUS TO BE BUTCHTATE
9999	147	C_DATA_BUS_OUT	EQU	9999H	*PROGRAM THE CODER DATA BUS TO BE DUTPUT(IE.
	146				TAKE CONTROL OF THE DATA BUS), ALL OTHER
	149				FORTS ON THE CODER BUS ARE INPUT. MSB
	150				FAND LSE OF THIS WORD EACH ADDRESSES ONE PPI.
0500	151	D DATA DUD TH	FOU	ODODU	ADDROCKAN THE RECORDS DATA BUR TO BE INCHIT AL
9B7B	152	D_DATA_BUS_IN L	EQU	9B9BH	PROGRAM THE DECODER DATA BUS TO BE INPUT, AL
	153				FOTHER FORTS ON DECODER BUS ARE INPUT.
	154				MSB AND LSB OF THIS WORD EACH ADDRESSES TO O
		NE FFI			
	155				
9999	156	D_DATA_BUS_OUT	EQU	9999H	*PROGRAM THE DECODER DATA BUS TO BE OUTPUT(IE
					ATALE CONTROL OF THE BATA BUSY ALL STUFF
	157				TAKE CONTROL OF THE DATA BUS), ALL OTHER
	158				FPORTS ON THE DECODER BUS ARE INPUT. MSB
	159 160				FAND LSB OF THIS WORD EACH ADDRESSES ONE PPI.
0000		CODOS LES ON	FOII	VVOOD	*DODCDAN THE EDDOD DUC TO DE+
0072	161 162	ERROR_LED_ON	EGN	0092H	PROGRAM THE ERROR BUS TO BE: PORT A - INPUT > NOT USED HERE
	163				; PORT B - INPUT > NOT USED HERE
	164				; PORT C - OUTPUT > TO ERROR_LED
	165				7 TORY C CON DI 7 TO ERRORLEED
009B	166	ERROR_LED_OFF	EQU	009BH	#PROGRAM THE ERROR BUS TO BE:
VV/D	167	LKNOK_LLD_OFF	Lao	VV/DII	FORT A - INPUT > NOT USED HERE
	168				PORT B - INPUT > NOT USED HERE
	169				PORT C - INPUT > TO ERROR_LED
	170				THE LED DISPLAY IN THIS STATE WILL
	171				ibe 'FF'
	172				
	173				
	174	CONTROL BUS DE	PENDENT C	ONSTANTS	
	175				
	176	DECODER BUS			
	177	j			
	178				

2.0	100		+.	-	20
- 84	u		11	-	m
- 11	ш	_	20	_	

[L40	LINE	SOURCE			
0004	179	SPEN	EQU	0004H	MASK FOR SPEN SIGNAL (LOW TRUE)
0010	180	SFEEN	EQU	0010H	#MASK FOR SFEEN SIGNAL (LOW TRUE)
	181				
	182				
	183	#CODER BUS			
	184	;			
	185				
0960	186	CFECLK	EQU	0800H	#MASK FOR CFECLK SIGNAL (HIGH TRUE)
6026	187	CPCLK	EQU	0020H	#MASK FOR CPCLK SIGNAL(HIGH TRUE)
	188				
	189				
	190				
	191	MDA IN HOUSE MONITOR	ENTRY PO	THIC	
	192				
0010	193	MORT_86_1P	EQU	0010H	OFFSET ADDRESS
FE9F	194	MUN7_86_CS	EQU	OFE9FH	SEGMENT ADDRESS
	195				
		\$EJECT			

11141110	10/07/82	DAGE
11:41:10	10/0//02	PHUE

6

; HULDEM APPLICATION INTERRUPTS: (INT 8 TO 15)

DW 1 DUP(?)

	226			
0020	227	ORG	20H	
0020 0020 (1 7777	228	INT_B_IP	DW 1 D	UP(?)
)				

INT_8_CS

225

229

0022 (1

LOC	ORJ	LINE	SOURCE		
	????				
0024	????	230	INT_9_IP	DW	1 DUP(?)
0026) (1 1777	231	INT_9_CS	DW	1 DUP(?)
0028) (1 ?Tf?	232	INT_10_IP	DW	1 DUP(?)
002A) (1 ?1??	233	INT_10_CS	DW	1 DUP(?)
6020) (1 ????	234	INT_11_IP	DW	1 DUP(?)
002E) (1 1777	235	INT_11_CS	DW	1 BUP(?)
0630) (1 777)	236	INTL12_IP	DW	1 DUP(?)
0032	; (1 ????	237	INTL12_CS	DW	1 DUP(?)
) (1 ????	236	INT_13_IP	DW	1 DUP(?)
0036) (1 1771	235	INTL13LCS	DW	1 DUP(?)
) (1 7777	240	INT_14_IF	DW	1 DUP(?)
003A) (1 ????	241	INT_14_CS	DW	1 DUP(?)
0030) (1 ????	242	INT_15_IP	DW	1 DUF(?)
003E) (1 ????	243	INT_15_CS	DW	1 DUP(?)
	1	244			
		245 245	INTERRUPT_VECTOR		ENDS
		247	INTERNOT PETEOTON		
		248			
			\$EJECT		

1.00	U±0	LINE SO	URCE		
•		251 252			
		253 DAT 254	TA SEGNENT		
		255	THIS SEGMENT WILL	RESIDE I	N RAM
0006	(24	256 257	PREDICTOR RUFFER	חוו	(((LAST_P_BUFFER_FRAME/16)+1) * LEN_P_BUFFER_FRAME) D
			(?)		THE STATE OF THE PROPERTY OF THE PROPERTY OF THE PROPERTY OF
	? ? ??				
0036		258	FE_BUFFER	DW	(LEN_FE_BUFFER) DUP(?)
		259			
0070	(<u>i</u> 1111)	260	LED_DISPLAY_VALUE	DW	1 DUF(?)
		261			
0072	(1 ????)	262	ABS_XIN	DW	1 DUP(?)
0074		263	XOUT	DW	1 DUF(?)
	'	264			
0076	(1 ????)	265	FSV	D₩	1 DUP(?)
0078		266	FSVL	DW	1 DUP(?)
007A		267	FSVH	DW	1 DUF(?)
		268			
0070	(1 ????	269	1	DW	1 DUP(?)
007E)	074		Till	4 BUD (5)
	???? }	270	IL	DM	1 DUP(?)
0080		271	L2	DW	1 DUP(?)
		272			
	(1 9777)	273	FE_SERVICE_COUNTER	DW	1 DUP(?)
		274			
0084	(1 ????	275	FE_SERVICE_PTR	DW	1 DUP(?)
		276			
008é	(1 ????	277	FE_TYPE	DW	1 DUP(?)

LOC ORJ	LINE	SDURCE
)		
	273	
0088 (1	279	SIGN_FLAG DW 1 DUP(?)
7777		
)		
	280	
	281	
	282	DATA ENDS
	283	
	284	
	265	
	286	
	267	
	288	
	289	
	290	
	290	
	292	STACK SEGMENT
	272	SIRUN SEURENI
	273 294	;**************************************
	295	** THE STACK IS SOLELY USED BY THE 8086 TO *
	273	# STORE RETURN ADDRESS IN INTERRUPT ROUTINES*
	297	7* 5 5 5 KE KEIDAR ADDRESS IN INTERNOT ROUTINES*
AAAA 77A		DW 60 DUF (?)
0000 (60	295	עום אום אום (יו)
7777		
1	62.5	TOD LAREL HODE
0078	299	TOS LABEL WORD
	300	2740
	301	STACK ENDS
	302	
	303 +1	\$EJECT

LOE OBJ

331 41 \$EJECT

LINE SOURCE SYSTEM MACROS: j-----311 THE MACROS ARE NOT LISTED IN THE ASSEMBLER GENERATED LISTING, REFER TO THE SOURCE FILE FOR MACRO CONTENTS. 3.5 3.77

LH ONJ

LINE	COURCE	
332		
3 33		
334		
335	**************************************	******
336	i *	
337	* REGISTER USAGE IN MOLDEM SIMULATOR S	
336		
339	\$41121121111111111111111111111111111111	*******
346		
341		
342	DEDICATED USAGE FOR ALL PARTS OF THE SYSTEM AT A	ALL TIME:
343		
344	FSI - FELBUFFER INPUT POINTER	
345	#BX - FE_BUFFER OUTPUT POINTER ETHIS	
346	; NON-INTERRUFT DRIVEN QUANTIZATI	
347	ORIGINAL REGISTER IS PRESERVED:	
348	#BP - FREDICTOR_BUFFER FRAME POINTER	
349	#DI - PREDICTOR_BUFFER OFFSET POINTER	
350		
351		
350	CS - COLE SEGMENT	
353	IDS - DATA SEGNENT	
354	TES - DATA SEGMENT OR INTERRUPT_VECTO	OR SEGMENT
355	#SG/SF - STACK OPERATION	
355		
357		
35.9	FUNASSIGNED REGISTERS:	
36		
37.0	THESE REGISTERS DO NOT CARRY DEDICATED F	FUNCTIONS IN THE MOLDEM
361	(SINULATOR:	
367		
363	PAX+ CX+ DX	
364		
365 +1	SEJECT SECTION OF THE	

392

393

394

395 396

397 +1 \$EJECT

000A 0000

000C CB60 000E 33F3

QMLT

L00 ORJ	LINE	SOURCE	
	366		
	367		
	368		PUBLIC START_ADDR
	369		
	370		
	371	CODE	SEGMENT
	372		
	373		ASSUME CS:CODE, DS:DATA, SS:STACK, ES:INTERRUPT_VECTOR
	374		
	375		
	376		
	377		;*************************************
	378		* STATIC VARIABLES *
	379		;*************************************
	360		
	391		
	302		FYR - ARRAY OF NLEV NORMALIZED QUANTIZER DUTPUT VALUES
	333		(IN INCREASING ORDER)
	384		; VALUE SCALED BY ** IN ** REPRESENTATION
0000 0000	385		YQ DW 0, 8192, 24576
0002 0020			
0004 00aC			
	386		
	387		;x0 - ARRAY OF NLEV-1 NORMALIZED QUANTIZER BREAK POINTS
	388		; (IN INCREASING ORDER)
	389		; VALUE SCALED BY ** IN ** REPRESENTATION
0004 0000 0008 0040	390		XQ DW 0, 16384
	391		

FRMLT - ARRAY OF NLEV, QUANTIZER MULTIPLIERS

IN 0, 27853, 62259

VALUE SCALED BY ** IN ** REPRESENTATION

002D 26C7060B001000

452

```
LOC OBJ
                      LINE
                             SOURCE
                      398
                      399
                             400
                                                MAIN PROGRAM
                      401
                                402
                      403
                      404
                                    *******************************
                       405
                       406
                                           REGISTER VALUES ARE NOT PRESERVED
                       407
                                           IN ALL PROCEDURES. THEY SHOULD BE
                       408
                                               SAVED BEFORE ENTERING.
                       469
                       419
                                    411
                       412
                            START_ADDR:
0010 FA
                      4:3
                                         CLI
                                                              DISABLE EXTERNAL INTERRUPT
                       414
                      117
                       416
                                    ******************************
                       417
                                    * INITIALIZE THE INTERRUPT VECTOR TABLE
                       ATE
                      419
                                    * THIS SUBSYSTEM USES FOLLOWING SIGNALS FROM *
                       420
                                    * THE SYSTEM BUS:
                       421
                       422
                                         - START OF FRAME (SOF_)
                      427
                                         - FE ENABLE (SFEEK_)
                      476
                                         - P ENABLE (SPEN_)
                      405
                                         - FE CLOCK (CFECLK)
                      4/4
                                        - P CLOCK (CPCLK)
                       427
                                    **************************
                      428
                      429
                      430
0011 Bb----
                      431
                                          MOU
                                                 AX: DATA
                                                              CANNOT MOVE INMED. VALUE TO
0014 8EDS
                       432
                                          VOH
                                                 DS. AX
                                                               SEGMENT REGISTER.
                      433
0016 BECO
                      434
                                                 ES, AX
                                                              JES AND DS ARE REFERING TO SAME
                      435
                                                               SEGMENT
                      435
                      437
                                          FRESET THE LED_DISPLAY_VALUE
0018 C70670000000
                      438
                                          VOM
                                                 LED_DISPLAY_VALUE, 0
                      439
001E B89200
                      440
                                          MOV
                                                 AX, ERROR_LED_ON
0021 E666
                      441
                                                 ERROR_BUS_SPVR, AL
                      442
                                          FOUTPUT ERROR DISPLAY VALUE
                      443
                      444
0023 B80000
                                          MOV
                      445
                                                 AX, 0
0026 E664
                                                 ERROR_LED, AL
                      446
                                          DUT
                      447
0028 B80000
                      448
                             ERROR_ENTRY:
                                          MOV
                                                 AX, INTERRUPT_VECTOR
                                                                     FUSE EXTRA SEGMENT TO ADDRESS
002B 8EC0
                      447
                                          VOM
                                                 ES, AX
                      450
                                                                     THE INTERRUPT VECTOR TABLE
                      451
```

VOM

NMI_IP, MONT_86_IF

		- National Control		-
10	10/07/	יפו	PARE	14

LOC	OBJ		LINE	SOURCE		
0034	2607060A009FFE		453		HOV	NMI_CS, MONT_86_CS
***	20704		454		VOII	AV OFFACT EVAPOTTAL THE
	BE0301		455		MOV	AX, OFFSET EXCEPTION_INT
	BB90	K	456		MOV	BX, SEG EXCEPTION_INT
	26430000		457		MOV	DIVIDE_INT_IP, AX
0046	26891E0200		458		MOV	DIVIDE_INT_CS; BX
004B	26A31000		459		MOV	OVERFLOW_IP, AX
004F	25891E1200		460		MOV	OVERFLOW_CS+ BX
			461			
			462			
0054	25070620000000		463		VOM	INT_R_IP, OFFSET_SOF_INTERRUPT
-	2607062200		464		KOV	INT_8_CS; SEG SOF_INTERRUPT
0001	200/001200	,,	465		1101	INTEGERS DEG GOT EINTERNOTT
0042	26070624005760		465		KOV	INT_9_IP, OFFSET PREDICTOR_OUTPUT_INTERRUPT
	2507062506		467		NOV	INT_9_CS, SEG PREDICTOR_OUTPUT_INTERRUPT
0007	202/102000		468		HUV	TRILITESS SEO FREDICIONLOUIFUILIRIERROFI
1500	0/070/000000000				VOU	THE 40 TO OFFICE PE DISTORE INTERDUCT
The same of the same of	26070629008600		469		HOV	INT_10_IP, OFFSET FE_OUTPUT_INTERRUPT
0077	2607062400	'n	470		MOA	INT_10_CS: SEG FE_OUTPUT_INTERRUPT
			471			
007E	26070620005400		472		AOA	INT_11_IP, OFFSET PREDICTOR_INPUT_INTERRUPT
0085	2607062E00	F	473		MOV	INT_11_CS, SEG PREDICTOR_INPUT_INTERRUPT
			474			
3800	2607063000H900		475		MOV	INT_12_IP, OFFSET FE_INPUT_INTERRUPT
0093	2607063200	F	476		VOM	INT_12_CS; SEG FE_INPUT_INTERRUPT
			477 +1	\$EJECT		

LOC OFJ	LINE	SOURCE			
	478				
	479				
	480	ASSUME ES:	DATA		
	481				
	482				
	483				
	484	#INITIALIZ	E 8086 PRI	CESSOR ENVI	RONMENT
	485				
	496				
009A BS	R 497	MO		DATA	CANNOT MOVE IMMED. VALUE TO
SOER GEOO	488	MC	DS,	AX	SEGMENT REGISTER.
	489				
009F 8EC0	490	MO	V ES,	AX	FES AND DS ARE REFERING TO SAME
****	491	V.		07400	SEGMENT
00A1 PS	R 492	MO		STACK	
00A4 8ED0 00A6 BC7800	493 494	·MC		OFFSET TOS	
COMO DEZOU	495	MO	V 3F1	טרר זבו וטס	
	475 496				
	497	# TATTIAL 17	E VII GAG	EN HARDWARE	
	498	71K11INC12	r Hrr 313	TH HUNTAUN	
	453				
	500				
	501	; **	*****	*****	
	502		PI		
	503	; **	*****		
	504				
	505				
	506	#CAUTION:	AUTOMATI	EDI IS USE	ED HERE. REFER TO INTEL APPLICATION
	597	i			HE 8259A PROGRAMMABLE INTERRUPT
	508	;	CONTROLL	R', UNDER I	HEADING 'AUTOMATIC EDI MODE'
	509				
	510				
00A9 B013	511	MO		ICW1	
00AE E67E	512	OL	IT PIC	PORTLO, AL	
**** *****	513	V.			
00AD BA7A00	514	MO		PIC_PORT_1	
00B0 B008 00B2 EE	515 516	MC OU		ICW2	
VVDZ EE	517	OU	I DAY	HL	
	518	:1	רט ז דכ או	T MEEDED E	OR CURRENT HARDWARE CONFIGURATION
	519		P# 2 12 W	A RECEED FO	OR CORRER! HARDWARE COM IDORATION
00B3 B003	520	ОМ	υ Δι.	ICW4	
OOBS EE	521	OU			
VVPO LL	522	00		n.	
OOB6 BOFE	523	MO	V AL,	ENABLE_SOF.	_MASK
00B6 EE	524	OL			
	525				
	526				
	527	; #1	******	*****	
	528	; ;			
	529	; 11	*****	*****	
	530				
	531				
	532	FINITIALLY	ALL PORT	ARE PROGRAM	MMED TO BE INPUT PORTS IN MODE O

11:41:10	10/07/82	PAGE	16

LOC OBJ	LINE SOURCE		
	533	FEXCEPT THE	ERROR LED PORT, WHICH WILL BE OUTPUT ALL THE TIME
	534		
00B9 B8939F	535	MOV	AX, C_DATA_BUS_IN
009C E746	536	OUT	CODER_BUS_SPVR, AX
	537		
	538		
OOBE B8989B	539	MOV	AX, D_DATA_BUS_IN
00C1 E74E	540	DUT	DECODER_BUS_SPVR, AX
	541 +1 \$EJECT		

MOLDEM

8086/8087/8088 MACRO ASSEMBLER

LOC DEL	LINE	SOURCE				
	542					
	543		*****	*****	**********	**
	544				APPLICATION PROGRAM ENVIRONMENT	
	545				*************	
	546					
	547					
	548					
	549					
	550					
	- 551			; FILL	. BUFFER AREAS WITH ZEROS	
	552					
0003 B80000	553			MOV	AX, 0	FILL VALUE
0004 B91800	554			HOV	CX: LENGTH PREDICTOR_BUFFER	FITERATION COUNT
0009 BF3000	505			MOV	DI, OFFSET PREDICTOR_BUFFER	
	550					
	1157			FES SE	TS TO THE SEGMENT BASE OF DATA S	SEGMENT
	558					
0000 F3	559		REF	STOS	PREDICTOR_BUFFER	
COCD AB						
	560					
	561					
00CE 392000	563			MOV	CX, LENGTH FE_BUFFER	FITERATION COUNT
0001 BF3000	563			MOV	DI, OFFSET FE_BUFFER	
	564					
00D4 F3	565		REF	STOS	FE_BUFFER	
6015 AE						
	566					
	567					
	559				SERVICE COUNTER	
0006 07068200000	569			MOV	FE_SERVICE_COUNTER, 0	
	570					
	511				ALIZE FE TYPE TO LOW	
6090 076686000000	572			MOV	FE_TYPE, FE_LOW	
	573					
	574					
	275			FINITI	ALIZE BUFFER POINTERS TO APPROFF	RIATE VALUES
	576					
	577				CTOR BUFFER , BP IS THE FRAME PO	
	578			FDI 18	THE OFFSET POINTER WITHIN A FR	AME.
	579					
00E2 BP2000	590			KOU	BF. LAST_P_BUFFER_FRAME	ON FIRST SOF INTERRUPT AFTER
	581					POWER UP, SOF_INTERRUPT ROUT
	F00	INE				MITH OFT DD A DT A
	582					#WILL SET BF = 0, DI = 0
00FF TF1000	583			WOLL	DI LEU D BUEFFE FRANC & /TVDI	ODERTOTOR DUFFERN
00E5 BF1000	584 585			HOV	DI, LEN_P_BUFFER_FRAME * (TYPE	FREDICION_BUFFER)
	586			IFF DI	FFER INITIALIZATION	
	587			TIL DE	FREN INTITALIZATION	
00E8 BE1000	588			MOV	SI, 16	POINTS TO FEL(4)
00EB BB0000	589			MOV	BX, 0	TOTAL TO TELLY
OVED DEVOVO	590			1104	Var.	
	591			:DIIANT	IZATION SERVICE	
	592			JUOMIN	IZHITON DENVIOL	
00EE C70684001000	593			HOV	FE_SERVICE_PTR, 16	
VVLL 0/0004001000	373			HUV	, LLOUNTOLLI IN TO	

LOC OFJ	LINE SOURCE		
00°4 C70578000A00	594	MOV FSVL, FSVMN	
00FA C7057AC00A00	595	MOV FSVH, FSVMN	
	596		
	597		
	598		
0100 FB	599	STI	FENABLE EXTERNAL INTERRUPT
	600		
0101 F4	601	HLT	#WAIT UNTIL THE FIRST
	602		START OF FRAME INTERRUPT
	603		
	604 +1 \$EJECT		

LGC OFJ	LINE	SOURCE
	605	
	606	
•	607	;*************************************
	608	i *
	609	* START OF APPLICATION PROGRAM *
	610	; *
	611	* AFTER POWER UP, THE FOLLOWING APPLICATION *
	612	* PROGRAM WILL NOT START EXECUTION UNTIL THE *
	613	# FIRST SOF_INTERRUPT HAS BEEN SERVICED. *
	614	;*************************************
	615	
	616	
0102		LABELLWAITLFORLNEWLFE:
	618	
	610	\$LOOF AROUND UNTIL THERE IS A NEW FE TO SERVICE
	620	TEST INSOLE SITE THERE IS IT HER IS TO SERVICE
0102 39358400	621	CMP FELSERVICE_PTR, SI
010a 74F)	622	JE LABEL_WAIT_FOR_NEW_FE
	623	OL ENGLEM IS IN ONLINE L
	624	TWO POINTERS ARE NOT EQUAL, POSSIBLE NEW FE'S
	625	THE POINTERS HIE REP ENGLE POSSIBLE REP TE S
	62.5	
	627	CHECK SERVICE COUNTER
0105 833E820000	629	CMP FE_SERVICE_COUNTER: 0
010B 744C	629	. JE LABEL_PTR_ERROR \$THE SERVICE PTR AND
	630	FINPUT POINTER(SI) ARE NOT
	631	#EQUAL, SERVICE COUNTER = 0
	632	#SYNC ERROR
	633	
	634	
	633	SERVICE COUNTER >=1 , NORMAL CONDITION
	636	The second of th
010F 833E820008	637	CMF FE_SERVICE_COUNTER, 8
0114 7345	638	JAE LABEL_PTR_ERROR FTHERE ARE MORE THAN 4 PAIRS
	639	FOF FEX TO BE SERVICED, CPU
	640	FIS RUNNING TOO SLOW
	641	The Heritage Toe Green
	642	#1 =< SERVICE COUNTER < B, NORMAL CONDITION.
	643	# BEGIN SERVICE THE FE'S
	644	PECTA SERVICE THE PE S
	645	
	646	
	647 +1	\$F.IFCT
	0.7 12	TAVANT

F30 0 FJ	LINE	SOURCE		
	648			
	649			
0116	650	LABEL_SERVICE_FE:		
	651	#MOVE	THE NEW FEX INTO AX AND CO	ORRESPONDING FSV TO CX
	652			
	653		TO USE THE BX REGISTER TO	ACCESS THE FE_BUFFER
0116 BB0E8400	654	VOM	CX, FE_SERVICE_PTR	
	655			
011A FA	656	CLI	AV AV	
011B 87CB	657	XCHG	CX, BX	
011D 8E4730	658		AX, FE_BUFFER[BX]	
0120 8703 0122 Fb	659	XCHG	CX, BX	
V122 7 E	661	STI		
0123 8338840000	562	PMD	FE_TYPE, FE_LOW	
0128 7542	663	JNE	LABEL_FE_HIGH	
V120 /G11	654	JITE	CHDECTLETHION	
	665			
	666			
	657			
	653	117 10	FEL TO BE PROCESSED	
012A SECE7800	659	KOV	CX, FSVL	
012E E87600	676	CALL	APCMQ	
	671	One	THE CITE	
	672	(RESIL	FSV IN DX, XOUT IN CX	
	673		TOT IN DAT ADDI IN DA	
0131 89167800	674	VON	FSVL, DX	SAVE THE NEW FSVL
	675			
0135 A19400	676	VOK	AX, FE_SERVICE_PTR	#FE_SERVICE_PTR WILL BE USED LATER
	677			
	678	FIF XO	IT IS ZERO THEN, STORE THE	NEW XOUT INTO THE FE_BUFFER
	679	# BACKG	ROUND NOISE LEVEL IS SUPPR	RESSED THROUGH THIS MACHANISM
	680	FIF XO	IT IS NON-ZERO , THE ORIGI	INAL VALUE IS PRESERVED. NO
	681	# QUANT	IZATION IS DONE TO THE FEL	L ITSELF.
	682			
0136 83F º 00	683	CMF	CX, 0	FCX CONTAINS THE XOUT VALUE
013F 7507	684	JNE	LABEL_NO_FEL_UPDATE	
	685			
	686		THE FEL, XOUT = 0 IN CX	
0130 FA	687	CLI		
013E 73	333	XCHG	AX, BX	
013F 894F30	689	MOV	FE_BUFFER[BX], CX	
0142 93	690	XCHG	AX, BX	
0143 FB	691	STI		
A	692	110 FF: 1100.4764		
0144	693	LABEL_NO_FEL_UPDATE:	EE OFFILIAGE ONLINES	
0144 FF0E8200	694	DEC	FE_SERVICE_COUNTER	
	695	*HDDATI	THE EE TYPE TO BE CEDITE	SED NEVI
0148 C70685000100	696	HOV	THE FE TYPE TO BE SERVICE	ELI REAL
VITO 6/V055000VIVO	697 698	ΠUV	FE_TYPE, FE_HIGH	
	699			
	700	: INCDE	ENT THE SERVICE POINTED	AX CONTAINS FE_SERVICE_PTR .
014E 050200	701	ADD	AX, TYPE FE_BUFFER	TA DUNINING FELDERVICE_FIR .
0151 3P4000	702	CMP	AX, LEN_FE_BUFFER * TYPE	FE_BUFFER \$END OF BUFFER?
0202 027000	142	Citi	IN LENE LEDOTTER & TIPE	TELEBOTTEN TENE OF BUTTEN:

LOC ORJ SOURCE LINE 0154 7348 703 0156 A38400 704 0159 EBAT 705 706 707 708 HI SEJECT

JAE LABEL_WRAF_AROUND VON FE_SERVICE_PTR, AX JMP LABEL_WAIT_FOR_NEW_FE

750 +1 \$EJECT

L00 0EC	LINE	SOURCE
	709	
	710	
	711	
	712	
	713	THIS LABEL_PTR_ERROR HANDLER IS POSITIONED HERE SO THAT IT IS WITHIN
	714	#127 BYTES FROM THE ORINGIN OF JUMP(OR RELATED) INSTRUCTIONS
	715	
)15B	716	LABEL_PTR_ERROR:
	717	
	718	;*************************************
	719	* THIS PART OF CODE HANDLES THE SYNCHRONIZATION ERROR
	720	* BETWEEN THE QUANTIZATION PROCESS AND THE VARIOUS
	721	* INTERRUPT DRIVEN I/O PROCESSES. *
	722	**
	723	## AN ERROR CONDITION MAY BE ONE OR MORE OF THE FOLLOWINGS: *
	724	* * * * * * * * * * * * * * * * * * *
	725	## (1) BOTH INPUT(REGISTER SI - DEDICATED) AND SERVICE POINTER #
	725	* ARE POINTING TO DIFFERENT ELEMENT IN THE FELBUFFER *
	727	* AND THE SERVICE COUNTER HAS A VALUE OF ZERO.
	729	# (2) THE SERVICE COUNTER HAS A VALUE HIGHER THAN 8. THIS *
	729	** MEANS THAT FOUR PAIRS OF FEX OR MORE ARE NOT QUANTIZED *
	730	* YET.
	731	* THE MULDEM SIMULATOR PROCESS WILL CONTINUE AFTER THE
	732	* ABOVE ERROR(S) OCCURS, THE FOLLOWING ROUTINE WILL CAUSE THE *
	733	** MOST SIGNIFICANT ERROR_LED TO INCREMENT.
	734	;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
	735	
	736	
MISE FE	737	CLI FDISABLE EXTERNAL INTERRUP
	738	
0150 A17666	739	MOV AX, LED_DISPLAY_VALUE
0157 907410	740	ADD AH, 10H
0162 AD7660	741	MOV LED_DISPLAY_VALUE, AX
0165 0AD4	742	OF ALT AH
0167 E664	743	OUT ERROR_LED; AL
0169 EPPCFE	744	JMF ERROR_ENTRY
	745	
	746	
	747	·
	748	;*************************************
	104	

		751					
0160		752	LABEL_FE_HIGH:		H IS TO BE PROCESSED.		
	280E7A00	753		MOV	CX, FSVH		
0170	E83400	754		CALL	APCMQ		
		755					
		755		FRESULT	FSV IN DX, XOUT IN CX		
		757					
0173	89167A00	758		MOV	FSVH, DX	SAVE THE NEW	FSVH
		759					
0177	A18400	760		VOM	AX, FE_SERVICE_PTR	#FE_SERVICE_PT	R WILL BE USED LATER
		761					
		762					
		763		FIF XOU	IT IS ZERO THEN, STORE	THE NEW XOUT INTO	THE FE_BUFFER
		764		# BACKGE	ROUND NOISE LEVEL IS SU	PPRESSED THROUGH	THIS MACHANISH
		765		FIF XOU	IT IS NON-ZERO , THE OR	IGINAL VALUE IS F	RESERVED. NO
		766		#QUANT]	ZATION IS DONE TO THE	FEH ITSELF.	
		767					
0176	835900	763		CMP	CX • 0	CX CONTAINS)	COUT
017	7507	760		JNE	LABEL_NO_FEH_UPDATE		
		770					
		771		#UPDATE	THE FEH , XOUT = 0 IN	CX	
017	F¢.	772		CLI			
	93	. 773		XCHG	AX, BX		
	994930	774			FE_BUFFER[BX], CX		
0184		775		XCHG	AX. BX		
0185	FF	776		STI			
		777					
018		778	LABEL_NO_FEH_UF	DATE:			
0188	FF0E8200	779		DEC	FE_SERVICE_COUNTER		
		780					
		781		SUPDATE	THE FE TYPE TO BE SER	VICED NEXT	
0186	4 C70686000000	782			FE_TYPE, FE_LOW		
	0,000,000	783					
		784					
		785		ITNEREN	ENT THE SERVICE POINTE	R	
019	0 50200	78 <i>è</i>			AX, TYPE FE_BUFFER		
	3D4000	787		CMF	AX, LEN_FE_BUFFER # T	YPE FE_BUFFER	FEND OF BUFFER?
	7305	788		JAE	LABEL_WRAP_AROUND		
	2 A38400	789		MOV	FE_SERVICE_PTR; AX		
	8 E964F:	790		JMF	LABEL_WAIT_FOR_NEW_FE		
		791					
			\$EJECT				

LOC OBJ	LINE	SOURCE
	793	
	794	
	795	
	796	
019E	797	LABEL_WRAP_AROUND:
	798	SERVICE POINTER WAS POINTING TO THE LAST ELEMENT IN FE
	799	FARRAY, MOVE TO THE FIRST ELEMENT.
	800	
019E 070684000000	801	MOV FE_SERVICE_PTR, 0
0184 E955FF	802	JMF LABEL_WAIT_FOR_NEW_FE
	803	STOCKET IN STATE OF S
	804	
	803	
	806	;x************************************
	807	* END OF MAIN PROGRAM *
	808	**
	809	7 64 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
	510	
	811	

612 41 \$EJECT

848 +1 \$EJECT

LUC DBJ

01A7

UGC - 0 80	LINE	SOURCE			
	849				
	850				
01A7 890E7600	851		VOM	FSV, CX	FFSV IS PASSED TO THIS PROCEDURE FROM
	852				REGISTER CX. SAVE FOR FUTURE REFEREN
		CE			
	853				THIS VALUE IS ALSO USED IN PROCEDURE
		IQUANTZ			
	854				
	855		FXIN 15	PASSED TO THIS PROCEDU	URE IN REGISTER AX.
	856				
01AB E70688000000	857		VOM	SIGN_FLAG, POSITIVE	FINITIALIZE THE SIGN FLAG
	858				
O1B1 A9FFFF	859		TEST	AX, OFFFFH	JGET SIGN
0154 7908	860		JNS	LABEL_1	FIT IS A POSITIVE NUMBER
	861				
	862		IXIN IS	A NEGATIVE NUMBER, GET	T ABSOLUTE VALUE
	863				
0136 F708	364		NEG	AX	
0188 670888000100	865		MOV	SIGN_FLAG, NEGATIVE	STORE STATE OF SIGN
	865				
01BE A37200	867	LABELL1:	MOV	ABS_XIN, AX	SAVE THE ABSOLUTE XIN FOR
	868				JUSE IN IQUANTZ PROCEDURE
	869				
	870				
	871				OT PRESERVED IN THIS CALL.
0101 E8630v	872		CALL	IQUANTZ	
	873				
	874				THIS VALUE IS DOUBLED THE
	675		FACTUAL	. VALUE TO FACILITATE I	NDEX ADDRESSING.
	876				
0104 89028030	877		VOK	L2, CX	SAVE FOR FUTURE REFENRECE
	078				
	879 41	SEJECT			

LOC OBJ	LINE SOURCE			
	850			******
	881			E FOR THE CODED BITS *
	882	*****	*********	**********
	883			
0108 A17600	884	MOV	AX, FSV	
	885			
01CB 3D0F00	886	CMP	AX, FSVTH	
	887			
01DE 730E	338	JAE	LABEL_MID_RISE	
	889			
	890	FSV (FSVTH	
	891			
	892			
01D0 83F902	853	CMP	CX, 2	\$L2 = 1?
	894			
01D3 7509	895	JNE	LABEL_MID_RISE	
	£96			
	877	;L2 =	1 , USE MID_TREAD QU	ANTIZER AND SET XOUT = 0
	875			
0105 270674000000	899	MOV	XOUT, 0	
01BB EB1B90	900	JMF	INC_STEP	
	901			
	902			
OIDE	903 LABELLMIDLRI	SE:		
	904			
	905			INT, DISABLE INTERRUPT BEFORE USE
	906	BX IS	S USEL BY INTERRUPT I	/O ROUTINES AS A DEDICATED REGISTER
	507			
GIDE FA	908	CLI		
01UF 8719	909	XCHG	BX, CX	CX CONTAINS "L2"
	910			
01E: 2EF727	911	MUL	CS:YQCBX3	FAX CONTAINS FSV
	912			
01E4 8709	913	XCHG	BX, CX	
	914			
O1Eć Fi	915	STI		
	916			
01E7 D100	917	ROL	AX, 1	SCALE BY 2**15
01E9 D1D2	918	RCL	DX, 1	
	919			
	920		RPORATE SIGN BIT	
01EF 833E880000	921	CMF	SIGN_FLAG, 0	
01F0 7402	922	JZ	LABEL_2	
	923			
	924	FXIN W	NAS A NEGATIVE NUMBER	
	925			
O1F2 F7DA	926	NEG	DX	
	927			
01F4 89167400	928 LABEL_2:	MOV	XOUT, DX	
	929			
	930 +1 \$EJECT			

LOC OBJ	LINE	SOURCE			
	931				
	932		*****	*********	*****
	933		# IN	CREMENT STEP SIZE OF	FSV *
	934		*****	***********	*****
	935				
	936				
01F8 8F0E8000	937	INC_STEP:	MOV	CX, L2	
	938				
	939		FREED	TO USE BX AGIAN, RE	FER TO PREVIOUS USAGE FOR DOCUMENTATION
	940				
OIFU FA	941		CLI		
OIFF STOR	942		XCHG	CX, BX	
01FF ZE8147v2	943		MOV	AX: CS:QMLTEBX3	
0203 BTCB	944		XCHG	CX, BX	
0205 Fb	945		STI		
0206 F7267600	945		MUL	FSV	
	947				
020A D100	948		ROL	AX: 1	SCALE RESULT BY 2**15
0206 Delk	949		RCL	DX, 1	
	950				
020E 8154E863	751		CMP	DX: FSVMX	
0212 7evs	952		JBE	LABEL_3	
0214 BAEB03	953		MOV	DX+ FSVMX	
0217 EP0596	954		JMP	APCKQ_END	
021A 83FA0A	955	LAREL_3:	CMF	DX. FSVMN	
021F 7303	956		JAE	APCMQ_END	
021F BA0600	957		MOV	DX; FSVMN	
	953				
	737		INEW I	SV IN DX	
	950				
0222 6B0E74C0	901	AFCMQ_END:	MOV	CX, XOUT	
022e L3	962		RET	0	
	963				
	964	AFCMQ	ENDP		
	965	rii oria	LILLI		
	966				
	967				
		\$EJECT			
	750 T.	*LULUI			

LGC OFJ

LINE	SOURCE		
969			
970	IQUANTZ	PROC	NEAR
971			
972	***** *	*********	*******************
973	# INFL	T: ABS_XIN	, FSV PASSED FROM MEMORY *
974	;*		1
975	# OUTF	UT: L	IN REGISTER CX. THIS VALUE IS
976	**		DOUBLE THAT OF ACTUAL VALUE TO *
977	*		FACILITATE INDEX ADDRESSING *
978	## REGI	STER USAGE:	
979	;*	BX	TEMPORARILY, VALUE SAVED AND *
980	**		RESTORED BY THIS PROCEDURE. *
981	;×		INTERRUPT DISABLED WHILE USING THIS*
982	* *		REGISTER. *
983	i×	DX	*
984	;*		
985	** REGI	STER ASSIGNMENT IN	THIS PROCEDURE: *
985	**		*
987	18	CX	CONTAINS IU *
988	ix	REFER TO FORTRA	LISTING FOR DESCRIPTION OF THESE *
989	7.*	VARIABLES.	
990			*********************
991	, 4 11, 1111		
902			
997 41	\$EUECT		

1046 1047

L00 080	LINE	SOURCE			
	994				
ni 27 C7067E000000	995		MOV	IL, 0	*TI = A
622I 596406	996		MOV	CX, NLEV* TYPE XQ	FIL = 0
V221 D70400 .	997		UUA	CAT NLEVE TITE AU	FIU = NLEV [* 2 FACTOR ADDED]
	998		±T - /	IL + IU) /2	
	999		11 - 1	IL T 107 72	
0230 8B167E00	1000	LABEL_100:	MOV	DX, IL	
0234 03D1	1001	LHDLL_IVV+	ADD	DX, CX	
0236 D1EA	1002		SHR	DX; 1	DIVIDE BY 2
023E 81E2FEFF	1003		ANI	DX; OFFFEH	MAKE SURE IT IS AN EVEN NUMB
VEGC CILLIFE.	1000	ER	unt	DAT VITTEN	THE SURE II IS HE EVEN RUND
0230 89167000	1004	-"	HOV	I, DX	\$SAVE I FOR FUTURE USE
0200 0710 000	1005			(.GT. XQ(I)*FSV) GOTO 220	TORVE I FOR FOTORE USE
	1006		72: 17	**** XE(174) 507 6510 220	
	1007		:NFED	TO HEE REGISTER BY AT THIS	POINT. DISABLE INTERRUPT BEFORE
	1008		FUSING		TOTALL PISHBEE TRIERROLL BELOKE
	1009		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		
	1010				
0240 Fr	1011		CLI		
0241 8702	1012		XCHG	DX, BX	
0243 2F884706	1013		MOV	AX, CS:XQ[BX]	MOV XQ TO AX REGISTER
0247 6783	1014		XCHG	DX: BX	AUDA YR IN HY VERTRIEV
0249 Ft	1015		STI	DAT DA	
V277 11	101d		211		
024A F7267600	1017		MUL	FSV	FXQ IS SCALED BY 2**15
024E D1C0	1018		ROL	AX, 1	7AU 15 SCHLED B! 24413
0250 P1F2	1019		RCL	DX, 1	
0252 39167200	1020		CMF	ABS_XIN, DX	
0256 7707	1021		JA	LABEL_200	
V200 7177	1022		JH.	LHDELL200	
	1023		;IU =	T	
0258 8F0E7000	1024		MOV	CX, I	
025C EB0990	1025			LABEL_300	
0200 20	1026		5111	LHILLIOVV	
025F	1027	LABEL_200:	FIL =	1	
025F 8B167C00	1028	LHILLILOVA	MOV	DX, I	
0263 89167E00	1029		HOV	IL, DX	
0200 0710.200	1030		nov	ILY DA	
	1030				
0267 BB167E00	1032	LABEL_300:	NOV	DX+ IL	
026F 83C202	1032	THEETTSOO!	ADD	DX, 2	FIL+ 1
026E 3BCA	1033		CMP	CX, DX	FILT I
0270 77EE	1035		JA	LABEL_100	
VLIV IIDE	1036		JH	LHBELLIVO	
	1037		*PFCIII	T IU IN CX REGISTER	
	1038		TRESUL	I TO IN ON NEDICIEN	
	1039				
0272 C3	1040		RET		
V2/2 00	1041		NE i		
	1042	IQUANTZ	ENDP		
	1042	TUDHRIZ	ENTIF		
	1043				
	1045				

1045 1046 1047 1050 ---- 1051 CODE 1052 1053 1054

1055 +1 \$EJECT

ENDS

LGC GEJ	LINE	SOURCE		
	1056			
	1057			
	1058			
	1059			
	1060	INTERRUPT_CODE	SEGMENT	
	1061			
	1062	ASSUME	CS: INTERRL	PT_CODE, DS:DATA, ES:NOTHING, SS:STACK
	1063			
0000	1064	SOF_INTERRUPT	PROC	FAR
	1065			
	106a	**********	******	****************
	1067	# THIS INTERRUPT	ROUTINE IS ACTIV	VATEL BY THE SOF_(START_OF_FRAME) SINGNAL *
	1968	** ON THE CODER BU	SITHE SOF_ ON TH	HE DECODER BUS IS SYNCHRONIZED AND OCCURS *
	1060	IN AT THE SAME TIM	E AS THE DECODE	R BUS SOF_).
	1//70	## THIS ROUTINE IS	RESPONSIBLE FOR	R CHECKING THE MULDEM OVERALL FIRMWARE *
	1071	* STATE INCLUDIN	G FOLLOWING:	
	1072	++ -	FE INPUT AND DUT	TPUT POINTER VALUES *
	1073	ii -	PREDICTOR COEFF	ICIENTS BUFFER FRAME AND OFFSET POINTER *
	1674	1).	VALUES	
	1675	11		
	1076	ST JF ANY OF THESE	VALUES IS ABNOT	RMAL, THIS ROUTINE WILL ATTEMP TO CORRECT #
	1077	# TO THE BEST OF	ITS KNOWLEDGE.	AT THE SAME TIME, IT WILL INCREMENT THE
	1078	FE ERROR LOG VALUE	AND OUTPUT TO 1	THE LED.
	1079	# THIS ROUTINE IS	ALSO RESPONSIB	LE FOR SETTING UP THE PRIDICTION COEFF
	1686	* BUFFER POINTERS	FOR THE CURRENT	FRAME.
	108.	11		
	1082		GISTER USAGES:	
	1083	**	AX, BX, B	P, SI, DI
	1084	; t		
	1065		******	************
	1086			
	1087			
	1088 +1	SEJECT		

L05	680	LINE	SOURCE			
•		1089				
		1090			NTERRUPT FLAG (IF) IS DISA	BLED IN MOST PART
		1091		FOF THIS ROU	TINE.	
		1092				
0000	50	1093		PUSH	AX	MAY USE AX REGISTER TO
		1094				FOUTPUT ERROR_LED IN THIS
		1095				FROUTINE
		1096			********	
		1097 1098		* FE_BUF	FER CHECK * ******************	
		1099		144444444	**********	
		1100				
	•	1101				
		1102		10HE	CK FE INPUT POINTER	
		1103) GNL	DR FE IRFUI FUIRIER	
0001	83FE10	1104		CMF	SI, 16	THIS IS THE NORMAL OFFSET
		1105		C.III	01, 10	IVALUE POINTING TO FEL (4)
00.04	7411	1106		JE	LABEL_NORMAL_1	TAREOL TOTALING TO TELLAT
		1107		02	Energia	
		1108 +1				
		1105 +1				
0005	A17000	1110 +1		MOV	AX, LED_DISPLAY_VALUE	
0009	FE: 0	1111 +1		INC	AL	三、《西里斯》 经产品
4000E	2404	1112 +1		AND	AL, OFH	
		1113				
0000	7405	1114		JZ	LED_ERROR_EXT1	
		1115				
		1116 +1				
	A27000	1117 +1		MOV	LED_DISPLAY_VALUE, AX	
0012	E654	1118 +1		DUT	ERROR_LED, AL	
		2119				
C014	BE1000	1120	LED_ERROR_EXT1:	MOV	SI, 16	CORRECT POINTER VALUE
2017		1121	LARCE MODULAL AA	AGUE	OF EE BUTCUT BOTHTED	
0017		1122	LABEL_NORMAL_1:	TUNEL	CK FE OUTPUT POINTER	
0017	83FB00	1123		CAL	DV A	ATUTO TO THE MODWAL HALLIE
0017	BOFFOO	1124 1125		CMF	BX, 0	FTHIS IS THE NORMAL VALUE FOINTING TO FEL(0)
0014	7411	1126		JE	LABEL_NORMAL_2	FOIRTING TO FEELO
VVIN		1127		J.	LHBELLNONIHL12	
		1128 +1				
		1129 +1				
0010	A17000	1130 +1		HOV	AX. LED_DISPLAY_VALUE	
001F	FECO	1131 +1		INC	AL	
	240F	1132 +1		AND	AL, OFH	
		1133				
0023	7405	1134		JZ	LED_ERROR_EXT2	
1		1135				
		1136 +1				
	A37000	1137 +1		VOM	LED_DISPLAY_VALUE, AX	
0028	E654	1138 +1		OUT	ERROR_LED, AL	
		1139				
002A	BB0000	1140	LED_ERROR_EXT2:	MOV	BX, 0	CORRECT POINTER VALUE
		1141				
		1142			*********	
		1143		** PREDICT	DR BUFFER CHECK *	

LOC OBJ	LINE	SOURCE			
	1144	;**	******	******	
	1145				
0020	1146	LABEL_NORMAL_2:	*CHECK	PREDICTOR INPUT POINTER	
	1147				
002D B3FF10	1148		CMP	DI, LEN_P_BUFFER_FRAME	* (TYPE PREDICTOR_BUFFER)
	1149				THIS IS THE NORMAL VALUE
	1150				POINTING TO ELEMENT AFTER
	1151				THE LAST ELEMENT OF A FRAME
5575 7465	1152		-	LABEL NOBYAL Z	
0030 740E	1153		JE	LABEL_NORMAL_3	
	1154				
	1155		FIRE PL	DINTER CONTAINS ILLEGAL V	ALUE
	1156				
	1157 1158 +1				
	1156 +1				
0032 617000			MOLI	AV LED DICOLAY HALLE	
	1160 +1		MOV	AX, LED_DISPLAY_VALUE	
0035 FEC0 0037 240F	1162 41		AND	AL+ OFH	
VVC/ 24VF	1163		HRD	HL! VFN	
0039 7405	1164		JZ	LABEL_NORMAL_3	
0007 7500	1165		JL	ENDELLINONINELS	
	1155 †1				
003B A37000	1167 -1		MOV	LED_DISPLAY_VALUE, AX	
003E E6:4	1165 +1		OUT	ERROR_LED; AL	
VV02 20.4	1169		001	ENNON-LEDY HE	
0040	1170	LABEL_NORMAL_3:	SET TH	HE PREDICTOR POINTERS FOR	NEXT ERAME DISTRICT
	1171	ENDEETHORNMETO.	FUL: 11	IL THEDISTON TOTALENS TO	REAL TRAIL BOTT OF
0040 BF0000	1172		MOV	DI, 0	
0043 836510	1173		ADI	SALES OF THE PARTY	*(TYPE PREDICTOR_BUFFER)
	1174				FINCREMENT TO NEXT FRAME
	1175				
0046 83FD20	1176		CMP	BP. LAST_P_BUFFER_FRAME	DETERMINE IF BP POINTS TO
	1177				LAST FRAME IN BUFFER
	1178				
0049 7693	1179		JBE	LABEL_SOF_END	
	1180				
	1181		FBP WAS	S POINTING TO LAST FRAME	IN BUFFER BEFORE INCREMENT
	1182				
004B BD0000	1183		NOV	BF, 0	MOVE TO THE FIRST FRAME IN B
		UFFER			
	1184				
004E	1185 +1	LABEL_SOF_END:			
	1186 +1				
	1187 +1				
	1188 +1				
004E FA	1189 +1		CLI		
004F B8E000	1190 +1		MOV	AX, ENABLE_NORMAL_MASK	
0052 E67A	1191 +1		OUT	PIC_PORT_1, AL	
0054 FB	1192 +1		STI		
0055 58	1193		POP	AX	
0056 CF	1194		IRET		
	1195				
	1196				
	1197	SOF_INTERRUPT	ENDP		

LOC 05J

LINE SOURCE

1198 1199

> 1200 1201 +1 \$EJECT

LGC OBJ	LINE	SOURCE	
	1202		
0057	1203	PREDICTOR_OUTPUT_INTERRUPT	PROC FAR
	1204		
	1205		
	1206	:******************** *	******************
	1207		VATED BY THE SPEN_ SIGNAL ON THE DECODER BUS. *
	1208		SIBLE FOR OUTPUT THE PREDICTOR COEFFICIENTS FROM *
	1209	* THE PREDICTOR COEFF BUF	
	1210	in the property court pur	
	1211	# REGISTER U	HICAREC+
	1212		
	1213	j*	K, DI
			*
	1214	**********************	**************
	1215		
	1216		
0057 50	1217		JSH AX FAX IS USED DURING THIS ROUTI
		NE.	
	1218 +1		
	1219 +1		
	1220 +1		
	1221 +1		
0056 BBFE00	1272 +1	HO	DV AX, ENABLE_SOF_MASK
005B E67A	1223 +1	OU	UT PIC_PORT_1, AL
005D FF	1224 +1	ST:	I
	1225		
	1226		
	1227	‡CI	CONFIGURE THE DECODER DATA BUS TO BE OUTPUT
	1228		
005E B87779	1229	ION CONTRACTOR OF THE CONTRACT	DV AX, D_DATA_BUS_OUT
0061 E74E	1230	OU	
	1231		The second section and the second sec
	1232	in.	DUTPUT DATA
	1233	,0	JOHOT PRIN
0063 3E6F03	1234	MON	OV AX. DS:PREDICTOR_BUFFEREBPJEDIJ
0000 023141	1235	110	ANY BOTT REDICTOREDOTT EREDI JEDIT
0066 E748	1236	OU	JT D_DATA_BUS, AX
VVVV L. TA	1237	00	A DEDMINEDOST NA
	1238	.11	INCREMENT BUFFER POINTER
	1239		MCKLIER) POFFER FOIRTER
0068 830702	1246	AD	DD DI, TYPE PREDICTOR_BUFFER
00ab 83FF10	1241	CM	
006E 7203	1241		
OUDE 7203		JB	S LABELLYLEN
	1243		ST BATUTA BEVAUE FUE OF BURBEUT FRAVE. FUE OF BUTBUT
	1244		DI POINTS BEYOND END OF CURRENT FRAME, END OF OUTPUT
	1245		DF A FRAME(THE N - 2NI FRAME) PREDICTOR COEFFICIENTS.
	1246		SET UP THE POINTER FOR INPUT NEW PREDICTOR
	1247		COEFFICIENTS AT END OF CURRENT FRAME(FRAME N).
	1248		THE INPUT DATA WILL BE DEPOSITED INTO THE SAME FRAME
	1249	\$B	BUFFER, THEREFORE BP REGISTER NEEDS NOT BE CHANGED.
	1250		
0070 BF0000	1251	MO	DV DI, O
	1252		
	1253		
0073	1254	LABEL_P_EN: \$C	CHECK FOR END OF OUTPUT CYCLE
0073 E54C	1255	IN	N AX, D_CONTROL_BUS

0078 250400 1256 AND AX, SPEN 1257 JZ LABEL_P_EN 1258 1259 \$NOW SPEN IS FALSE(I.€. SIGNAL IS HIGH	H), TAKE DATA OFF THE BU
S	
1260 007A BB9B9B 1261 MOV AX, D_DATA_BUS_IN 007B E74E 1262 OUT DECODER_BUS_SPVR, AX	
1263 1264 +1 1265 +1 1266 +1	
1267 +1	
007F FA 1268 +1 CLI	
0080 B8E000 1269 +1 MOV AX, ENABLE_NORMAL_MASK	
0083 E674 1270 +1 OUT PIC_PORT_1; AL	
0085 FB 1271 +1 STI 1272	
0086 58 1273 POP AX	
0087 CF 1274 IRET	
1275	
1276	
1277 PREDICTOR_OUTPUT_INTERRUPT ENDP	
1278	
1279	
1266	
126% +1 \$EJECT	

LCC OBU	LINE	SOURCE		
	1282			
	1283			
0088	1284	FE_OUTPUT_INTERRUPT	PROC	FAR
	1285	, LLOUI, C. LLIN LINIO, I	, ,,,,,	
	1286			
		الله الله الله الله الله الله الله الله		
	1287			***************
	1288			ACTIVATED BY FE ENABLE_ SIGNAL ON THE DECODER *
	1289			OUTPUTING FEH AND FEL TO THE DECODER BUS *
	1296			X IS PUSHED ONTO STACK, DATA BUS PORT IS *
	1291			TO DUTPUT STATE, THE APPROPRIATE FEX IS *
	1292			ER TO BUFFER IS INCREMENTED TO NEXT FEX TO BE *
	1293			LOOP AROUND UNTIL FE ENDABLE_ IS HIGH, THEN *
	1294		FF FROM	THE DATA BUS, CONVERT PORT TO INPUT STATE *
	1295	* AND EXIT		
	1297	**		
	1297	** REGISTER USAGE:		
	1298	## AX: BX		
	1,749	*************	*****	************
	1300			
	1301			
0068 50	1302		PUSH	AX
	1303			
	1304 +1			
	1305 +1			
	1306 +1			
	1307 +1			
089 R8FE00	1308 +1		MOV	AX, ENABLE_SOF_MASK
0080 E67A	1309 +1		OUT	PIC_PORT_1, AL
008E FA	1310 +1		STI	FIC_FURI_IF HL
OVOL . 1	1311		311	
	1312			
			*00115	TOURS BATA BUILD BORT TO BE OUTDUT
	1313		1 LUNE 1	IGURE DATA BUS PORT TO BE OUTPUT
AADE BOOGG	1314		VOII	
008F B89999	1315		NOV	AX, D_DATA_BUS_OUT
0092 E74E	1316		OUT	DECODER_BUS_SPVR, AX
	1317			
0094 8B4730	1318		MOV	AX, FE_BUFFER[BX]
	1319			
	1320		FBX IS	B THE POINTER TO THE NEXT WORD TO BE OUTPUT IN FE BUFFER
	1321			
0097 E744	1322		OUT	D_DATA_BUS, AX
	1323			
	1324		FINCRE	MENT THE OUTPUT POINTER
	1325			
0099 830302	1326		ADD	BX, TYPE FE_BUFFER
009C 83FB40	1327		CMP	BX, LEN_FE_BUFFER * TYPE FE_BUFFER
009F 7203	1328		JB	LABEL_FE_ENABLE
00A1 BB0000	1329		MOV	BX, 0
	1330			
	1331			
	1332			
0044 E54C	1333	LABEL_FE_ENABLE:	IN	AX, D_CONTROL_BUS
00A6 251000	1334	LADELLI CLEMADLE I	AND	AX, SFEEN
00A9 74F9	1335		JZ	
VVN7 /417			JL	LABEL_FE_ENABLE
	1336			

11:51:55 10/07/82 PAGE	3	١	١											į				į	į	į			į					į																			į	į																											į	ė	ĺ	l		į	į	ĺ		ĺ	į	۱	å	ĺ	į					ĺ			1	į								2	3		١		Ē	۱	l	ì	١	1	į	G	ì	į	۱	į	,	ľ	,	į			١	l	į	ľ	ļ	ł	į	i	ĺ		i							
------------------------	---	---	---	--	--	--	--	--	--	--	--	--	--	---	--	--	--	---	---	---	--	--	---	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	---	---	---	--	---	---	---	--	---	---	---	---	---	---	--	--	--	--	---	--	--	---	---	--	--	--	--	--	--	--	---	---	--	---	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---	---	---	---	---	---	---	---	--	---	--	--	--	--	--	--	--

LOC OHJ	LINE	SOURCE			
	1324	EE TUDUT THEFDOMET	0000	FAD	
00BF	1325	FE_INPUT_INTERRUFT	PROC	FAR	
	1326				
	1327			********	
	1328			CTIVATED BY THE FE ENABLE	
	1329			FOR TAKING THE FEX(FEH	
	1330			E THEM INTO THE FE BUFFE	
	1331			PUSHES AX REGISTER ONTO	
	1332			IS VALID, THE FEX DATA	
	1303		FE_BUFFER	. THE ROUTINE THEN INCRE	
	1334	** POINTER AND EXIT.			
	1335	9 章			
	1336	** REGIS	TER USAGE	S:	•
	1337	7#	AX,SI		1
	1338	1*			*
	1339	*****************	*******	*********	******
	1340				
	1341				
	1342		FTHIS I	ROUTINE IS ACTIVATED BY	FELENABLE SIGNAL ON THE
	1343		CODER	CONTROL BUS	
	1344				
	1345				
	1345				
00BF 50	1347		PUSH	AX	
VVIII C.	1348 +1			***	
	1349 +1				
	1350 +1				
	1351 +1				
00CO BELEGO	1352 +1		MOV	AX ENABLE_SOF_MASK	
0003 E67A	1353 +1		OUT	PIC_PORT_1: AL	
0003 E871	1354 41		STI	110-10M1-11 HL	
VV53 T P	1355		0.1		
1000		LADEL DE CLET	+ CHECK	FOR FE CLOCK VALIDITY	
9006	1356	LABELLFE_CLK:	TUNEUN	LOW LE CEOCH AWEIDILL	
AAA. PP.11	1357		TH	AV C CONTROL BUC	
00C6 E544	1358		IN	AX, C_CONTROL_BUS	
0008 250608	1359		AND	AX, CFECLK	
00CB 74F9	1360		JZ	LABEL_FE_CLK	TO HIGH DEAD IN FEV FROM DATA
	1361	5.113	ITE LL	OCH 15 INDECTIFE STORME	IS HIGH), READ IN FEX FROM DATA
	47/0	BUS			
	1362		711	AV C DATA DUC	
00CD E542	1363		IN	AX, C_DATA_BUS	
	1364				
	1365				
00CF 894430	1366		MOV	FE_BUFFER[SI], AX	
	1367				
	1368		;SI IS	THE OFFSET POINTER FOR	DEPUSITION
	1369				
	1370		FINCRE	MENT POINTERS	
	1371				
	1372				
00D2 83C602	1373		ADD	SI, TYPE FE_BUFFER	
00D5 FF068200	1374	1 1	INC	FE_SERVICE_COUNTER	FINCREMENT THE SERVICE
	1375				COUNTER TO ENABLE NON-INTERR
		UPT			
	1376				DRIVEN QUANTIZATION PROCESS
The second secon					

LOC ORJ	LINE SOURCE	
	1337	FNOW SPEEN_ IS FALSE(SIGNAL IS HIGH), TAKE DATA OFF BUS
00AB B89BFB	1338	NOV AX, D_DATA_BUS_IN
00AE E74E	1339	OUT DECODER_BUS_SPVR, AX
	1340	
	1341 +1	
	1342 +1	
	1343 +1	
	1344 +1	
00E0 FA	1345 +1	CLI
00B1 P8E000	1346 +1	MOV AX, ENABLE_NORMAL_MASK
00B4 E67A	1347 +1	OUT PIC_PORT_1, AL
00Fo FF	1348 +1	STI
0010 11	1349	
0087 38	1350	POP AX
VVP/ GG	1351	101 11
OOBB CF)352	IRET
VVDO C.	1353	The I
	1354 FE_OUTPUT_INTERRUPT	ENDP
		ERM
	1355	
	1356	
	1357	
	1358 +1 \$EJECT	

LOC OBJ	LINE	SOURCE			
	1359				
0019	1360	FE_INPUT_INTERRUPT	PROC	FAR	
	1361				
	1362			*********	
	1363			CTIVATED BY THE FE ENABL	
	1364			E FOR TAKING THE FEX(FEH	
	1365			E THEM INTO THE FE BUFFE	
	1366			PUSHES AX REGISTER ONTO	
	1367			IS VALID, THE FEX DATA	
	1366		TE_BUTTER	. THE ROUTINE THEN INCRE	MENIS THE BUFFER #
	136° 1370	IN POINTER AND EXIT.			
	137)	i* REGIS	TEF USAGE	· ·	
	1372	** REGIS	AX,SI	•	
	1373	##	HV191		
	1374		******	*******	*****************
	1375	A man to the desirable des	and dededed and deded	************	\$ \$ \$ \$ C \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$
	1376				
	1377		ETHIS I	ROUTINE IS ACTIVATED BY	FE FNARIE STRNAL ON THE
	1378			CONTROL BUS	TEENWEE STORIE ON THE
	1377		700221	COMMOL POD	
	1380				
	1381				
00B9 50	1382		PUSH	AX	
	1383 +1				
	1384 +1				
	1385 +1				
	1385 +1				
OOFA REFEOO	1387 +1		MOV	AX, ENABLE_SOF_MASK	
GOBI: E67A	1388 +1		OUT	PIC_PORT_1, AL	
OOBF FR	1389 +1		STI		
	1390				
0000	1391	LAPEL_FE_CLK:	FCHECK	FOR FE CLOCK VALIDITY	
	1392				
0 000 E544	1393		IN	AX, C_CONTROL_BUS	
0002 250008	1394		AND	AX, CFECLK	
00C5 74F9	1395		JZ	LABEL_FE_CLK	10 H10H
	1396		FFE CL	UCK IS TRUE(I.E. SIGNAL	IS HIGH), READ IN FEX FROM DATA
	1700	BUS			
00E7 E542	1397		TAI	AX, C_DATA_BUS	
UUL/ E342	1398 1399		IN	HAY C_DHIH_DUS	
	1400				
00C9 894430	1401		MOV	FE_BUFFER[SI], AX	
0007 674450	1402		nuv	TELBUTTERESITY MA	
	1403		IST IS	THE OFFSET POINTER FOR	DEPOSITION
	1404		,,,,,	THE OFFICE FOR THE FOR	
	1405		INCRE	MENT POINTERS	
	1406		7 2 110112		
	1407				
00CC 83C602	1408		ADD	SI, TYPE FE_BUFFER	
00CF FF068200	1409		INC	FE_SERVICE_COUNTER	FINCREMENT THE SERVICE
	1410				COUNTER TO ENABLE NON-INTERR
		UPT			
	1411				DRIVEN QUANTIZATION PROCESS

LOC OB.	BJ LINE	SOURCE				
O003 83	1412 2FE40 1413		CMP	CT I FN FF DUFFFD * TV	DE EE DUEEED	AFAID OF DUFFE
- 0013 011	1413	R?	LIT	SI, LEN_FE_BUFFER * TYP	TE PE_BUTTER	SEND OF BUFFE
0006 72	203 1414		JB	LABEL_FE_INPUT_END		
0008 BE	E0000 141 5		MOV	SI, 0	RESET POINTE	R TO BEGINNING
	1416				FOF CIRCULAR	BUFFER.
	1417					
OODE	1418 +					
	1419 +					
	1420 +					
	1421 +					
OODB FA			CLI			
000C BS			MOV	AX, ENABLE_NORMAL_MASK		
00DF E6			OUT	PIC_PORT_1, AL		
00E1 FB			STI			
00E3 58			POP	AX		
	1427					
00E3 OF			IRET			
	1429					
	1430	FE_INFUT_INTERRUFT	ENDP			
	1431					
	1432					
	1433 +	1 \$EJECT				

LOC OBJ	LINE	SOURCE			
	1434				
	1435				
	1436				
00E4	1437	PREDICTOR_INPUT_INTER	RUPT	PROC FAR	
	1438				
	1439				
	1440				
	1441	***************	******	****************	*******
	1442			BY THE PEN_ SIGNAL ON THE COI	
	1443			FFICIENTS FROM THE CODER BUS AN	ND STORES THE *
	1444	# DATE IN THE PREDIC	TOR_COEFF	BUFFER.	1
	1445	†*			
	1445		TER USAGE		
	1447	**	AX, DI		1
	1448	18			*
2051 51	1449	*************		*******************	TAXXXXXXXXXXXX
00E4 50	1450		PUSH	AX	
	1451 1452 +1				
	1453 +1				
	1454 +1				
	1455 +1				
00ES BEFEOO	1456 +1		MOV	AX, ENABLE_SOF_MASK	
00E8 E67A	1457 +1		OUT	PIC_PORT_1, AL	
OOEA FB	1458 +1		STI	TIOLI ON LIFT HE	
	1459				
	1460		;TEST	FOR P CLOCK VALID	
	1461				
	1462				
00ER E544	1463	LAREL_PLCLK:	IN	AX, C_CONTROL_BUS	
	1464				
00ED 252000	1465		AND	AX, CPCLK	
0050 7459	1466		JZ	LABEL_P_CLK	
	1467				
	1468				
	1469			CK IS VALID, READ IN PREDICTOR	COEFFICIENTS FROM
	1470		#DATA	BUS	
	1471				
	1472				
00F2 E542	1473		IN	AX, C_DATA_BUS	
	1474				
	1475			ECIFIES THE FRAME STARTING OFFS	
AACA 7500A7	1476			PECIFIES THE OFFSET WITHIN EACH	
00F4 3E8903	1477		MOV	DS:PREDICTOR_BUFFER[BP][DI],	HA .
00E7 970700	1478 1479		ADD	DI, TYPE PREDICTOR_BUFFER	FINCREMENT OFFSET
00F7 83C702	14/9		HDD	DIT THE PREDICTOR_BUFFER	FVALUE WITHIN ONE FRA
	1400	HE			ANTOE MILITA ONE PRH
	1481	112			
	1482				
	1483 +1				
	1484 +1				
	1485 +1				
	1486 +1				
OOFA FA	1487 +1		CLI		
	1101 11		-		

F00:	OBJ	LINE	SOURCE	
OOFB	R5E000	1488 +1	VON	AX, ENABLE_NORMAL_MASK
OOFE	E67A	1489 +1	DUT	PIC_PORT_1, AL
0100	FB	1490 +1	STI	
		1491		
0101	58	1492	POP	AX
0102	CF	1493	IRET	
		1494		
		1495	PREDICTOR_INPUT_INTERRUPT	ENDP
		1496		
		1497		
		1498		
		1499 +1	\$EJECT	

LOC ORJ	LINE	SOURCE			
	1500				
0103	1501	EXCEPTION_INT	PROC	FAR	
	1502				
	1503	;*********	********	*********	
	-1504	# THIS IS	THE EXCEPTI	ON HANDLER FOR THE 8086 CPU. *	
	1505	# UNDER NO	DRMAL OPERAT	ION ENVIRONMENT, THIS CODE SHOULD *	
	150è	IN NEVER BI	EXECUTED.	1	
	150?	# IN THE L	INLIKELY EVE	INT THAT THIS HANDLER IS ACTIVATED.	
	1508	# IT SETS	THE LED_DIS	SPLAY_VALUE TO OFFH AND OUTPUT *	
	1509	# TO THE E	ERROR_LED.	1	
	1510	;*	REGIS1	TER USAGES: *	
	1511	;*		AX,	
	1512	i*		1	
	1513	*********	********	***********	
	1514				
	1515				
	1516	JEXTERNAL 1	INTERRUPTS A	RE DISABLED DURING EXECUTION OF THIS HANDLE	R
	1517				
	1518				
0103 50	1517		PUSH	AX	
0104 BSFFFF	1520		MOV	AX, OFFFFH	
0107 A37000	1521		MOV	LED_DISPLAY_VALUE, AX	
010A E664	1522		OUT	ERROR_LED, AL	
0100 58	1523		POP	AX	
010B FF	1524		STI		
- 010E CF	1525		IRET		
	1526				
	1527				
	1528	EXCEPTIONLINT	ENDP		
	1529				
	1530				
	1531				
	1532	INTERRUPT_CODE	ENDS		
	1533		END		

ASSEMBLY COMPLETE: NO ERRORS FOUND

 $\frac{\text{APPENDIX C}}{\text{DESCRIPTION OF THE ORIGINAL DRT TAPE SUPPLIED BY CRC}}$

APPENDIX C DESCRIPTION OF THE ORIGINAL DRT TAPE SUPPLIED BY CRC

F٠	_	~	_	1	
г	c	⋍	E	1	

TEST	WORD NUMBER	LEFT WORD	RIGHT WORD	ANSWER	ATTRIBUTE	PEAK STRENGTH (milivolts)	POWER (dB)
1	01	BOR	GOB	3	. 7	231	3
1	02	DAUNT	TAUAT	1	1	187	0
1	03	TOOM	BOOT	2	2	130	- 3
1	0.4	SHEET	CHEAT	2	3	091	- 6
1	05	JAB	GAR	2	4	176	0
1	06	POT	TOT	i	5	231	3
1 1	07 08	GHOST RILL	BOAST NILL	1 3	6 7	185 154	0
1	09 09	ZED	SAID	2	i	158	ŏ
1	10	GNAW	DAW	i	2	205	3
1	11	SHOES	CHOOSE	1	3	111	- 3
i	12	CHEEP	KEEP	2	4	083	- 6
1	13	BANK	DANK	1	5	146	0
1	14	GOT	POT	i	6	209	3
1	15	ROSE	NOSE	3	7	139	- 3
1	16	DINT	TINT	2	i	099	- 6
1	17	NECK	DECK	2	2	190	0
1	18	THONG	TONG	1	3	205	3
1	19	CHEW	000	2	4	130	- 3
1	20	WEED	REED	1	5	068	- 9
1	21	SHAG	SAG	2 3	6 7	222	3
1 1	22 23	KNOB VOLE	ROB FOAL	្ន 1	, 1.	195 127	0 - 3
1	24	NIF	DIP	2	2	150	Ö
1	25	FENCE	FENCE	2	3	086	- 6
ī	26	SAW	THAW	<u>-</u>	4	214	3
1	27	POOL	TOOL	2	5	110	- 3
1	28	YIELD	WIELD	1	6	104	- 3
1	29	GNAT	RAT	3	7	153	0
1	30	COOT	TOOT	3	7	116	- 3
1	31	BOND	FOND	1	1_	189	0
1	32	MOAN	BONE	1	2	159	0
1	33	VILL	BILL	2	3 .	137	- 3
1	34 35	JEST FOUGHT	GUEST THOUGHT	2 1	4 5	165 246	0 3
1 1	36	COOP	POOP	1	6	09B	- 6
i	.37	NEAP	REAP	3	7	075	- 6
î	38	VAST	FAST	2		178	ō
i	39	KNOCK	DOCK	1	1 2 3 4	186	0
1	40	THOSE	DOZE		3	210	3
1	41	SING	THING	2 2 2		148	0
1	42	MET	NET		5	156	0
1	43	CAUGHT	TAUGHT	1	6	283	6
1	44	NUDE	RUDE	3	7	131	- 3
1	45	BEAN	PEEN	i	1	125	- 3
1	46	MAD	BAD	1	2	165 303	0
1	47	VOX	BOX	. 2	2 3 4 5 6	210	6 3
1	48 49	BID JOE	GO DID	2	τ. Τ	160	0
1	47 50	AEM RID	WREN	2 2 2 3	Ä	138	- 3
1	51	ROT	NOT	3	7	287	6
•	O.I.	NO I	1. U I	<u>.</u>	•	que ser e	•

TEST	WORD NUMBER	LEFT WORD	RIGHT WORD	ANSWER	ATTRIBUTE	PEAK STRENGTH (milivolts)	POWER (dB)
i	52	Z00	SUE	1	1	152	0
1	53	NEED	DEED	i	2	091	- 6
1	54	THAN	DAN	2	3	167	0
1	55	CHOP	COF	2	4	264	3
1	56	FORE	THOR	2	5	171	0
1	57	HIT	FIT	2	6	181	0
1	58	NEST	REST	3	7	199	3

TEST	WORD NUMBER	LEFT WORD	RIGHT WORD	ANSWER	ATTRIBUTE	PEAK STRENGTH (milivolts)	POWER (dB)
1 222222222222222222222222222222222222	NUM 1 023 045 07 8 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	PEULS K W SE STANDON SRIEDOO E DE DONNO HOUSE K W SRIEDOO E DONNO HOUSE K W SRIEDOO E DE DONNO HOUSE K	TEAUE NO ED ED ENDUSE TEAUE NO ED ED ED ENDUSE TEAUE NO ED	AN 32111223122122311212132112123322112231211123221113	ATTRIBUTE 712345671234567123456712345677123456712345671234567	194 268 094 123 155 230 175 133 112 233 127 108 262 175 116 117 214 094 175 235 151 131 125 197 080 188 162 142 128 193 122 096 163 196 120 223 166 133 170 196 147 125 139	03630303333333333333336603033303000333333
_	~.	Ψ11·1π	******	-	•		_

TEST	WORD NUMBER	LEFT WORD	RIGHT WORD	ANSWER	ATTRIBUTE	PEAK STRENGTH (milivolts)	POWER
2	52	DUNE	TUNE	2	1	095	- 6
2	53	MEAT	REAT	2	2	076	- 6
2	54	SHAD	CHAD	1	3	151	0
2	55	TOL	GOT .	1	4.	199	. 3
2	56	BOWL	DOLE.	1	5	148	0
2	57	GILL	DILL	1	. 6	121	- 3
2	58	RED	NED	3	7	116	- 3

3	TEST	WORD NUMBER	LEFT WORD	RIGHT WORD	ANSWER	ATTRIBUTE	PEAK STRENGTH (milivolts)	POWER
3 03 MODT BOOT 1 2 113 - 3 3 04 SHEET CHEAT 1 3 062 - 9 3 05 JAB GAB 1 4 125 - 3 3 06 POT TOT 2 5 192 0 3 07 GHOST BOAST 2 6 162 0 3 08 KILL NILL 3 7 111 - 3 3 10 GMAW DAW 2 2 2 217 3 3 11 SHOES CHOOSE 2 3 097 - 6 3 12 CHEEF KEEP 1 4 077 - 6 3 13 BANK DANK 2 5 118 - 3 3 14 GOT DOT 2 6 182 0 3 15 ROSE HOSE 3 7 147 0 3 16 DINT TINT 1 1 103 - 3 3 17 NECK DECK 1 2 133 - 3 3 18 THONG TOMG 2 3 183 0 3 19 CHEW COD 1 4 095 - 6 3 22 KHOB ROB 3 7 217 3 3 24 NIP DIP 1 2 128 - 3 3 24 NIP DIP 1 2 128 - 3 3 26 SAW THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 29 GWAT TAW 2 7 119 - 3 3 29 GWAT TAW 2 7 119 - 3 3 20 GWAT THOUGHT 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	3							
3 04 SHEET CHEAT 1 3 062 - 9 3 05 JAR GAB 1 4 125 - 3 3 06 POT TOT 2 5 1792 0 3 07 GHOST BOAST 2 6 162 0 3 08 RILL NILL 3 7 7 111 - 3 3 09 ZED SAID 1 1 1 121 - 3 3 10 GHAW DAN 2 2 2 217 3 3 11 SHOES CHOOSE 2 3 097 - 6 3 12 CHEEF KEEF 1 4 077 - 6 3 13 BANK DANK 2 5 118 - 3 3 14 GOT DOT 2 6 182 0 3 15 ROSE NOSE 3 7 147 0 3 15 ROSE NOSE 3 7 147 0 3 16 DINT TINT 1 1 103 - 3 17 REEK BEEK 1 2 133 - 3 18 THONG TONG 2 3 183 0 3 19 CHEW COO 1 4 0975 - 6 3 20 WEED REED 2 5 096 - 6 3 21 SHAG SAG 1 6 174 0 3 22 KNOB ROB 3 7 217 3 3 24 NIP DIP 1 2 128 - 3 3 23 VOLE FOAL 2 1 139 - 3 3 24 NIP DIP 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 POUL TOUL 1 5 118 - 3 3 28 YIELD WIELD 2 6 194 2 2 3 3 3 27 POUL TOUL 1 5 118 - 3 3 29 GNAT RAT 3 7 180 0 3 3 3 WOLE FOAL 2 1 3 3 115 - 3 3 29 MORN BONE 2 1 1 203 3 3 3 3 WOLE FOAL 2 1 139 - 3 3 29 MORN BONE 2 1 1 203 3 3 3 3 WILL BILL 1 3 164 0 3 3 3 WOLE FOOL 1 1 5 118 - 3 3 29 GNAT RAT 3 7 180 0 3 3 3 WOLE FOOL 1 1 5 118 - 3 3 29 MORN BONE 2 1 1 203 3 3 3 3 WOLE FOOL 1 1 5 118 - 3 3 3 3 WOLE FOOL 1 1 5 118 - 3 3 3 3 WOLE FOOL 1 1 5 118 - 3 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 118 - 3 3 4 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 5 119 - 3 3 3 WOLE FOOL 1 1 1 5 111 - 3 3 4 WOLE FOOL 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	3							
3 05 JAB 6AB 1 4 125 -3 3 06 POT TOT 2 5 192 0 3 07 6HOST ROAST 2 6 162 0 3 08 RILL NILL 3 7 111 -3 3 09 ZED SAID 1 1 121 -3 3 10 GNAW DAW 2 2 2 217 3 3 11 SHOES CHOOSE 2 3 097 -6 3 12 CHEEP KEEP 1 4 077 -6 3 13 BANK DANK 2 5 118 -3 14 6OT DOT 2 6 182 0 3 15 ROSE NOSE 3 7 147 0 3 16 DINT TINT 1 1 1 103 3 -3 17 NECK BECK 1 2 133 -3 3 18 THONG TONG 2 3 183 0 3 17 NECK BECK 1 2 133 -3 3 18 THONG TONG 2 3 183 0 3 19 CHEW COO 1 4 095 -6 3 20 WEED REED 2 5 096 -6 3 22 KNOB ROB 3 7 227 3 3 22 KNOB ROB 3 7 227 3 3 24 NIP DIP 1 2 128 -3 3 25 FENCE PENCE 1 3 115 -3 3 26 SAW THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 -3 3 28 YIELD WIELD 2 6 124 -3 3 3 28 YIELD WIELD 2 6 124 -3 3 3 3 3 NOLE FOR TOOL 1 5 118 -3 3 28 YIELD WIELD 2 6 124 -3 3 3 3 NOLE FOR TOOL 1 5 118 -3 3 3 3 NOLE FOR TOOL 1 5 118 -3 3 28 YIELD WIELD 2 6 124 -3 3 3 3 NOLE TOOL 1 5 118 -3 3 3 18 BOND POND 2 1 203 3 3 3 3 NOLE TOOL 1 5 118 -3 3 28 YIELD WIELD 2 6 124 -3 3 3 3 NOLE TOOL 1 5 118 -3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 118 -3 3 3 3 3 NOLL BILL 1 5 119 -3 3 3 3 3 NOLL BILL 1 5 119 -3 3 3 3 3 NOLL BILL 1 5 119 -3 3 3 4 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 158 0 3 3 3 NOCK BOOK 2 2 15					1			
3	3				1			
3 07 GHOST BOAST 2 6 142 0 3 08 RILL NILL 3 7 111 - 3 3 09 ZED SAID 1 1 121 - 3 3 10 GHAM DAW 2 2 2 217 3 3 11 SHOES CHOOSE 2 3 097 - 6 3 12 CHEEP KEEP 1 4 077 - 6 3 13 BANK DANK 2 5 118 - 3 14 GOT DOT 2 6 182 0 3 15 ROSE NOSE 3 7 147 0 3 16 DINT TINT 1 1 103 - 3 3 17 NECK DECK 1 2 133 - 3 3 18 THONG TONG 2 3 183 0 3 19 CHEW COO 1 4 095 - 6 3 20 WEED REED 2 5 096 - 6 3 21 SHAG SAG 1 6 174 0 3 22 KNOB ROB 3 7 217 3 3 23 VOLE FOAL 2 1 139 - 3 3 24 NIP DIP 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 3 28 YIELD WIELD 2 6 124 - 3 3 3 3 VILL BILL 1 3 164 0 3 3 3 VOLE THOW THOM DOT 1 19 - 3 3 3 3 VILL BILL 1 3 164 0 3 3 3 VILL BILL 1 3 164 0 3 3 3 VOLE THOW THOM DOT 1 19 0 3 3 3 VILL BILL 1 3 164 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 3 VILL BILL 1 3 164 0 3 3 3 VILL BILL 1 3 164 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 3 VANCE DOCK 2 2 158 0 3 3 4 VAST FAST 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 1 186 0 3 3 4 VAST FAST 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	3				-			
3 08 KILL NILL 3 7 111 - 3 3 09 ZED SAID 1 1 1 121 - 3 3 10 GNAW DAW 2 2 2 217 3 3 11 SHOES CHOOSE 2 3 097 - 6 3 12 CHEEF KEEF 1 4 077 - 6 3 13 BANK DANK 2 5 118 - 3 3 14 GOT DOT 2 6 182 0 3 15 ROSE NOSE 3 7 147 0 3 16 DINT TINT 1 1 1 103 - 3 3 17 NECK BECK 1 2 133 - 3 3 17 NECK BECK 1 2 133 - 3 3 19 CHEW COO 1 4 4 095 - 6 3 21 SHAG SAG 1 6 174 0 3 22 KNOB ROB 3 7 217 3 3 22 KNOB ROB 3 7 217 3 3 22 KNOB ROB 3 7 217 3 3 23 VOLE FOAL 2 1 139 - 3 3 24 NIF DIP 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 29 GNAT RAT 3 7 180 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 128 0 3 37 NEAR AST 1 1 186 0 3 37 NEAR SAG 1 1 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 31 BOND POND 2 1 203 3 3 32 HOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 37 NEAR REAP 3 7 119 - 3 3 36 COOP POPP 2 6 181 0 3 37 NEAR REAP 3 7 118 0 3 37 NEAR REAP 3 7 118 0 3 39 KNOCK DOCK 2 2 198 0 3 40 THOSE DOCK 1 1 13 158 0 3 34 JEST GUEST 1 4 158 0 3 37 NEAR REAP 3 7 118 6 0 3 37 NEAR REAP 3 7 118 6 0 3 37 NEAR REAP 3 7 118 6 0 3 37 NEAR REAP 3 7 118 6 0 3 38 CAUGHT THOUGHT 2 5 225 3 3 44 NUBE RUBE 3 7 118 6 0 3 34 JEST GUEST 1 4 158 0 3 34 JEST GUEST 1 5 110 - 3 3 44 NUBE RUBE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 48 JOE GO 1 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 49 BID DID 1 5 117 - 3 3 50 YEN WEEN 1 6 098 - 6								
3	3							
3 10 GNAW DAW 2 2 2 217 3 3 11 SHOES CHOOSE 2 3 097 -6 3 12 CHEEF KEEF 1 4 077 -6 3 13 BANK DANK 2 5 118 -3 3 14 GOT BOT 2 6 182 0 3 15 ROSE NOSE 3 7 147 0 3 16 DINT TINT 1 1 1 103 -3 3 17 NECK DECK 1 2 133 -3 3 18 THONG TONG 2 3 183 0 3 19 CHEW COD 1 4 095 -6 3 21 SHAG SAG 1 6 174 0 3 22 KNOB ROB 3 7 217 3 3 22 KNOB ROB 3 7 217 3 3 22 KNOB ROB 3 7 217 3 3 23 VOLE FOAL 2 1 139 -3 3 24 NIF DIP 1 2 128 -3 3 25 FENCE PENCE 1 3 115 -3 3 26 SAW THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 -3 3 26 SAW THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 -3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 -3 3 31 BOND POND 2 1 203 3 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POP 2 6 181 0 3 37 NEAP REAP 3 7 118 0 3 38 VAST FAST 1 1 186 0 3 37 NEAP REAP 3 7 118 0 3 38 VAST FAST 1 1 186 0 3 39 KNOCK DOCK 2 2 198 0 3 40 THOSE DOCK 1 3 175 0 3 41 SING THING 1 4 111 -3 3 42 HET NET THOUGHT 2 5 225 3 3 44 NIF OUT THOUGHT 2 5 225 3 3 36 COOP POP 2 6 181 0 3 37 NEAP REAP 3 7 118 0 3 38 VAST FAST 1 1 186 0 3 37 NEAP REAP 3 7 118 0 3 39 KNOCK DOCK 2 2 198 0 3 40 THOSE DOCK 2 2 198 0 3 44 NUDE RUDE 3 7 132 -3 3 45 BEAN PEEN 2 1 072 -6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 1 4 179 0 3 49 BID DID 1 5 117 -3 3 50 YEN WEEN 1 6 098 -6								
3	<u>ئ</u> ح							
12	స 7							
3 13 BANK DANK 2 5 118 - 3 3 14 GOT DOT 2 6 182 0 3 15 ROSE NOSE 3 7 147 0 3 16 DINT TINT 1 1 1 103 - 3 3 17 NECK DECK 1 2 133 - 3 3 17 NECK DECK 1 2 133 - 3 3 18 THONG TONG 2 3 183 0 3 19 CHEW COO 1 4 4 095 - 6 3 20 WEED REED 2 5 096 - 6 3 21 SHAG SAG 1 6 174 0 3 22 KNOB ROB 3 7 217 3 3 23 VOLE FOAL 2 1 1379 - 3 3 24 NIF DIP 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 POOL 1 5 118 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 180 0 3 31 BOND POND 2 1 203 3 3 32 MOON BONE 2 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 37 NEAF REAF 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK DOCK 2 2 198 0 3 40 THOSE DOCK 2 2 198 0 3 40 THOSE DOCK 2 2 198 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 5 225 3 3 44 NUE NUE DOCK 2 2 198 0 3 45 BEAN PEEN 2 1 7 132 - 3 3 46 HAD BAB 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO TEN HEEN 1 5 098 - 6								
3								
3	7							
3 16 DINT TINT 1 1 103 -3 3 17 NECK DECK 1 2 133 -3 3 18 THONG TONG 2 3 183 0 3 19 CHEW CGO 1 4 095 -6 3 20 WEED REED 2 5 096 -6 3 21 SHAG SAG 1 6 174 0 3 22 KNOB ROB 3 7 217 3 3 23 VOLE FOAL 2 1 139 -3 3 23 VOLE FOAL 2 1 139 -3 3 24 NIP DIP 1 2 128 -3 3 25 FENCE PENCE 1 3 115 -3 3 26 SAW THAW 2 4 222 3 3 27 POOL <td>7</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	7							
3 17 NECK DECK 1 2 133 - 3 3 18 THONG TONG 2 3 183 0 3 19 CHEW COO 1 4 4 095 - 6 3 20 WEED REED 2 5 096 - 6 3 21 SHAG SAG 1 6 174 0 3 22 KNOB ROB 3 7 217 3 3 22 KNOB ROB 3 7 217 3 3 23 VOLE FOAL 2 1 139 - 3 3 24 NIF DIP 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAH THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 31 BOND POND 2 1 203 3 3 27 MOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAP REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK DOCK 2 2 198 0 3 40 THOSE DOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 44 NUDE RUDE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 177 - 3 3 49 BID DID 1 5 117 - 3 3 49 BID DID 1 5 117 - 3 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 0998 - 6	7.							
3	3							
3 22 KNOB ROB 3 7 217 3 3 23 VOLE FOAL 2 1 137 - 3 3 24 NIF 10)F 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 FOOL TOOL 1 5 118 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 1 203 3 3 32 WILL BILL 1 3 164 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAP REAP 3 7 118 0 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUBE RUBE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 177 0 3 49 BID BID 1 1 5 117 - 3 3 49 BID BID 1 1 5 117 - 3 3 49 BID BID 1 1 5 117 - 3 3 50 YEN WEN 1 6 098 - 6	3							
3 22 KNOB ROB 3 7 217 3 3 23 VOLE FOAL 2 1 137 - 3 3 24 NIF 10)F 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 FOOL TOOL 1 5 118 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 1 203 3 3 32 WILL BILL 1 3 164 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAP REAP 3 7 118 0 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUBE RUBE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 177 0 3 49 BID BID 1 1 5 117 - 3 3 49 BID BID 1 1 5 117 - 3 3 49 BID BID 1 1 5 117 - 3 3 50 YEN WEN 1 6 098 - 6	3				i			
3 22 KNOB ROB 3 7 217 3 3 23 VOLE FOAL 2 1 137 - 3 3 24 NIF 10)F 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 FOOL TOOL 1 5 118 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 1 203 3 3 32 WILL BILL 1 3 164 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAP REAP 3 7 118 0 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUBE RUBE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 177 0 3 49 BID BID 1 1 5 117 - 3 3 49 BID BID 1 1 5 117 - 3 3 49 BID BID 1 1 5 117 - 3 3 50 YEN WEN 1 6 098 - 6	3				2			
3 22 KNOB ROB 3 7 217 3 3 23 VOLE FOAL 2 1 137 - 3 3 24 NIF 10)F 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 FOOL TOOL 1 5 118 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 1 203 3 3 32 WILL BILL 1 3 164 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAP REAP 3 7 118 0 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUBE RUBE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 177 0 3 49 BID BID 1 1 5 117 - 3 3 49 BID BID 1 1 5 117 - 3 3 49 BID BID 1 1 5 117 - 3 3 50 YEN WEN 1 6 098 - 6	3		SHAG	SAG	1	6	174	Ò
3 24 NIF D)P 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAP REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 37 NEAP REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUBE RUBE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 46 MAD BAB 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	3	22	KNOB	ROB		7	217	3
3 24 NIF D)P 1 2 128 - 3 3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 - 3 3 28 YIELD WIELD 2 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAP REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 37 NEAP REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUBE RUBE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 46 MAD BAB 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	3	23	VOLE	FOAL	2	1	139	
3 25 FENCE PENCE 1 3 115 - 3 3 26 SAW THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 - 3 3 28 Y1ELD WIELD 2 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAP REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 37 NEAP REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUBE RUBE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	3	24	NIF	IO P	i	2	128	- 3
3 26 SAW THAW 2 4 222 3 3 27 POOL TOOL 1 5 118 -3 3 28 Y1ELD WIELD 2 6 124 -3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 -3 3 31 BOND POND 2 1 203 3 3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEA	3	25	FENCE	PENCE	1	3		
3 28 Y1ELD WIELD 2 6 124 - 3 3 29 GNAT RAT 3 7 180 0 3 30 COOT TOOT 3 7 119 - 3 3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAF REAF 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUDE RUDE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 49 BID DID 1 5 117 - 3 3 49 BID DID 1 5 117 - 3 3 49 BID DID 1 5 117 - 3 3	3							
3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAF REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUDE RUDE 3 7 132 - 3 4 5 BEAN PEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 5 50 YEN WREN 1 6 098 - 6						5		
3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAF REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUDE RUDE 3 7 132 - 3 4 5 BEAN PEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 5 50 YEN WREN 1 6 098 - 6	3							
3 31 BOND POND 2 1 203 3 3 32 MOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAF REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUDE RUDE 3 7 132 - 3 4 5 BEAN PEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 5 50 YEN WREN 1 6 098 - 6	3							
3 32 MOAN BONE 2 2 158 0 3 33 VILL BILL 1 3 164 0 3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAF REAP 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUDE RUDE 3 7 132 - 3 3 45 BEAN FEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 5 50 YEN WREN 1 6 0998 - 6	3					•		
3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAF REAF 3 7 118 -3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 -3 3 42 MET NET 1 5 110 -3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUBE RUBE 3 7 132 -3 3 45	3					-		
3 34 JEST GUEST 1 4 158 0 3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP POOP 2 6 181 0 3 37 NEAF REAF 3 7 118 -3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 -3 3 42 MET NET 1 5 110 -3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUBE RUBE 3 7 132 -3 3 45	ა -							
3 35 FOUGHT THOUGHT 2 5 225 3 3 36 COOP FOOP 2 6 181 0 3 37 NEAF REAF 3 7 118 - 3 3 38 VAST FAST 1 1 186 0 3 39 KNOCK DOCK 2 2 198 0 3 40 THOSE DOZE 1 3 175 0 3 41 SING THING 1 4 111 - 3 3 42 MET NET 1 5 110 - 3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUIE RUDE 3 7 132 - 3 3 45 BEAN PEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 50 YEN WREN 1 6 098 - 6								
3 36 COOP FOOP 2 6 181 0 3 37 NEAP REAP 3 7 118 -3 3 38 VAST FAST 1 1 1 186 0 3 39 KNOCK BOCK 2 2 198 0 3 40 THOSE BOZE 1 3 175 0 3 41 SING THING 1 4 111 -3 3 42 MET NET 1 5 110 -3 3 43 CAUGHT TAUGHT 2 6 228 3 3 44 NUDE RUDE 3 7 132 -3 3 45 BEAN FEEN 2 1 072 -6 3 46 MAD BAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 -3 5 50 YEN WREN 1 6 098 -6	ა 7							
3 45 BEAN FEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	ئ. ع				2			
3 45 BEAN FEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	3 7				7			
3 45 BEAN FEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	7							
3 45 BEAN FEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	3							
3 45 BEAN FEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	3							
3 45 BEAN FEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	3							
3 45 BEAN FEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	3							
3 45 BEAN FEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	3							
3 45 BEAN FEEN 2 1 072 - 6 3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 - 3 3 50 YEN WREN 1 6 098 - 6	3							
3 46 MAD BAD 2 2 170 0 3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 -3 3 50 YEN WREN 1 6 098 -6	3				2			
3 47 VOX BOX 1 3 249 3 3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 -3 3 50 YEN WREN 1 6 098 -6 3 51 ROT NOT 3 7 209 3	3				2			0
3 48 JOE GO 1 4 179 0 3 49 BID DID 1 5 117 -3 3 50 YEN WREN 1 6 098 -6 3 51 ROT NOT 3 7 209 3	3	47	Vox	BOX		3		3
3 49 BID DID 1 5 117 -3 3 50 YEN WREN 1 6 09B -6 3 51 ROT NOT 3 7 209 3	3					4		
3 50 YEN WREN 1 6 09B - 6 3 51 ROT NOT 3 7 209 3	3							
3 51 ROT NOT 3 7 209 3	3					6		- 6
	3	51	ROT	TON	3	7	209	3

TEST	WORD NUMBER	LEFT WORD	RIGHT WORD	ANSWER	ATTRIBUTE	PEAK STRENGTH (milivolts)	POWER (dB)
3	52	Z00	SUE	2	1	127	- 3
3	53	NEED	DEED	2	2	084	- 6
3	5.4	THAN	DAN	1	3	138	- 3
3	55	CHOP	COP	i	4	201	3
3	56	FORE	THOR	1	5	165	0
3	57	HIT	FIT	1	6	137	- 3
3	58	NEST	REST	3	7	162	0

TEST	WORD NUMBER	LEFT WORD	RIGHT WORD	ANSWER	ATTRIBUTE	PEAK STRENGTH (milivolts)	POWER (dB)
4	01	PEST	TEST	3	7	216	3
4	02	VAUL T	FAULT	<u>i</u>	1	243	3
4	03	NEWS	DUES	2	2	120	- 3 - 3
4	04 05	VEE SANK	BEE THANK	2 2	3 4	113 146	- 3
4) A	08	WAD	ROD	1	5	131	- 3
4	07	SHOW	\$0 \$0	1	. 6	156	ō
. 4	08	RIF	NIF	- 3	. 7	156	0
. 4	09	DENSE	TENSE	2	1	104	- 3 .
4	10	MOSS	BOSS	1	2	236	3
. 4	11	F00	P00	1	3	126	- 3
4	12	ZEE	THEE	2	4	135	- 3
4	13	FAD .	THAD	1	, 5	208	3
4	14	HOP	FOP	1	<u> 6</u>	256	3
4	15	ROAD	NODE	3	7	126	- 3
4	16	GIN	CHIN	2	1	092 119	- 6 - 3
Ą	17	MEND	BEND	2	2 3	203	3
4	18 19	SHAW JUICE	CHAW GOOSE	i 2	4	133	- 3
4 4	20	PEAK	TEAK	1	5	095	- 6
4	21	GAT	BAT	2		187	0
4	22	ТОИ	ROT	3	7	216	3
4	23	GOAT	COAT	i	1	182	0
4	24	MIT	BIT	2	2	151	0
4	25	THEN	DEN	2	3	118	- 3
4	26	JAWS	GAUZE	1	4	240	3
. 4	27	ноон	иоои	2	5	091	- 6
4	28	KEY	TEA	1	6	095	- 6
4	29	RAP	NAP	ა 7	7	167 192	0
4	30	FAN JOCK	PAN CHOCK	3	,	294	6
4 4	31 32	NOTE	DOTE	1	2	189	Ö
. 4	33	THICK	TICK	2	. 2 3	197	Ŏ
4	34	CHAIR	CARE	2		186	0
4	35	BONG	DONG	1	4 5 6	232	0 3 - 3
4	36	YOU	RUE	1		136	
4	37	WREATH	NEATH	3 .	7	088	6
4	38	GAFF	CALF	2	1	201	3
4	39	MOM	BOMB	1	2	217 201	3 3
. 4	40	THOUGH	DOUGH	2 2	3 4	179	0
4	41	JILT	GILT	$\frac{2}{2}$		141	ŏ
4	42 43	FENT YAWL	TENT WALL	i	5 6	239	. 3
4 4	44	ROOSE	NOOSE	3	7	138	- 3
4	45	VEAL	FEEL	1		127	- 3
4	46	NAB	DAB	1	1 2 3	177	0
4	47	VON	BON	2	3	218	3
4	48	SOLE	THOLE	2	4	179	0
.4	49	FIN	THIN	2	5	126	- 3
4	50	KEG	PEG	2 2 3	6	203	- 3 3 3
4	51	GNAW	RAW	· 3	7	257	. .

TEST	WORD NUMBER	LEFT WORD	RIGHT WORD	ANSWER	ATTRIBUTE	PEAK STRENGTH (milivolts)	POWER (dB)
. 4	52	DUNE	TUNE	1.	1	152	. 0.
4	53	MEAT	BEAT	i	2	087	- 6
4	54	SHAD	CHAD	2	3	153	0
4	55	JOT	GOT	2	4	241	3
4	56	BOWL	DOLE	2	5	205	3
4	57	GILL	DILL	2	6	187	0
4	58	RED	NED	3	7	187	0

APPENDIX D
LIST OF WRONG ANSWERS

APPENDIX D LIST OF WRONG ANSWERS

CODING	BACKGROUND NOISE	PERSON	LEFT WORD	RIGHT ₩ORD	ATTRIBUTE	REPLY
None	None	LYNNE	DAUNT	TAUNT	1	Left
None	None	LYNNE	VOX	BOX	3	Right
None	None 🗥	PETER	FAD	THAD	5	Risht
None	None	ALVA	FIN	THIN	5	Right
None	None	WOLF	FORE	THOR	5	Risht

	BACKGR	תאטם		LEFT	RIGHT		
CODING	HDISI	Ξ	PERSON	WORD	WORI	ATTRIBUTE	REPLY
					i.		
None	Armoured	Car	WOLF	BONI	POND	i	Left
None	Armoured	Car	LYNNE	DAUNT	TAUNT	1	Left
None	Armoured	Car	PETER	DAUNT	TAUNT	i	Left
None	Armoured	Car	ALVA	SHAD	CHAD	3	Risht
Norie	Armoured	Car	ALVA	THOSE	DOZE	3	Risht
Norie	Armoured	Car	WOLF	VEE	BEE	. 3	Left
None	Armoured	Car	LYNNE	VEE	BEE	3	Left
None	Armoured	Car	ALVA	VEE	BEE	3	Risht
None	Armoured	rsO	PETER	VEE	BEE	3	Left
None	Armoured	Car	LYNNE	VILL	BILL	3	Risht
None	Armoured	rsC	ALVA	vox	BOX	3	Risht
None	Armoured	Car	PETER	FIN	THIN	5	Risht
Norie	Armoured	Car	WOLF	FIN	THIN	5	Left
None	Armoured	Car	WOLF	FORE	THOR	5	Risht
None	Armoured	Car	LYNNE	MET	ИЕТ	5	Left
None	Armoured		LYNNE	MET	NET	5	Neither
None	Armoured		WOLF	MET	NET	5	Left
None	berwoured	Car	LYNNE	CAUGHT	TAUGHT	6	Neither
None	Armoured	Car	ALVA	KEG	PEG	6	Neither
None	Armoured	Car	LYNNE	SHAG	SAG	6	Left

	BACKGROUND		LEFT	RIGHT		•
CODING	NOISE	PERSON	WORD	WORD	ATTRIBUTE	REFLY
	,					
None	Helicorter	ALVA	SHAI	CHAD	3	Risht
None	Helicopter	LYNNE	SHOES	CHOOSE	3	Risht
None	Helicopter	WOLF	VEE	BEE	3	Left
None	Helicorter	PETER	VEE	REE	3	Left
None	Helicorter	WOLF	FAI	THAD	5	Risht
None	Helicorter	ALVA	FAD	THAD	5	Left
None	Helicopter	LYNNE	FIN	THIN	5	Right
None	Helicorter	WOLF	FIN	THIN	5	Left
None	Helicorter	WOLF	FIN	THIN	5	Risht
None	Helicorter	PETER	FIN	THIN	5	Left
Norie	Helicopter	ALVA	FIN	THIN	5	Right
None	Helicopter	PETER	MET	NET	5 .	Left

CODING	BACKGROUND NOISE	PERSON	LEFT	RIGHT	A T T D T D T T	pent V
COLING	MOTOE	LEKSUM	WORD	WORD	ATTRIBUTE	REPLY
DELD A D	N	1 671712	TAUNT.	******		1 - 61
RELP 4.8 RELP 4.8	None	LYNNE ALVA	DAUNT	TAUNT	1	Left
RELP 4.8	None :		GIN	CHIN	1	Left
RELF 4.8	None	LYNNE	GOAT	COAT	1	Left
RELF 4.8	None None	LYNNE PETER	VOLE VOLE	FOAL FOAL	1 1	Left Left
RELF 4.8	None	WOLF	VOLE	FOAL	1	Left
RELP 4.8	None	PETER	VOLE	FOAL	i	Left
RELF 4.8	None -	PETER	Z00	SUE	1	Left
RELF 4.8	None	LYNNE	FENCE	PENCE	3	Risht
RELF 4.8	None	ALVA	SHAD	CHAD	3	Right
RELF 4.8	None	LYNNE	SHAD	CHAD	3	Risht
RELF 4.8	None	PETER	SHAD	CHAD	3	Right
RELP 4.8	None	WOLF	SHAW	CHAW	. 3	Right
RELP 4.8	None	PETER	SHAW	CHAW	3	Right
RELP 4.8	None	ALVA	SHAW	CHAW	3 3	Right
RELP 4.8	None	ALVA	SHOES	CHOOSE	3	Risht
RELF. 4.8	None	LYNNE	SHOES	CHOOSE	3	Risht
RELP 4.8	None	WOLF	SHOES	CHOOSE	· 3	Right
RELF 4.8	None	ALVA	THAN	DAN	3	Risht
RELF 4.8	None	ALVA	THAN	DAN	3	Right
RELF 4.8	None	WOLF	THEN	DEN	3	Left
RELP 4.8	None	ALVA	THEN	DEN	3	Left
RELP 4.8	None	LYNNE	THONG	TONG	3	Left
RELP 4.8	None	LYNNE	THONG	TONG	3	Left
RELF 4.8	Мопе	WOLF	VEE	BEE	3	Left
RELP 4.8	Mone	ALVA	VILL	BILL	3	Right
RELP 4.8	None	ALVA	VILL	BILL	3	Right
RELP 4.8	None	LYNNE	VOX	BOX	3 3	Right
RELP 4.8 RELP 4.8	None None	ALVA LYNNE	VOX VOX	BOX	ა 3	Risht Risht
RELP 4.8	None	PETER	VOX	BOX	3	Risht
RELP 4.8	None	WOLF	VOX	BOX	3	Right
RELF 4.8	None	WOLF	VOX	BOX	3	Risht
RELP 4.8	None	PETER	Vox	BOX	3	Right
RELP 4.8	None	ALVA	vox	BOX	3	Risht
RELP 4.8	None	LYNNE	JAB	GAR	4	Left
RELP 4.8	None	WOLF	JAB	GAB	4	Left
RELP 4.8	None	PETER	JOE	60	4	Right
RELP 4.8	None	PETER	JOE	60	4	Risht
RELP 4.8	None	LYNNE	SING	THING	4	Right
RELF 4.8	None	LYNNE	SING	- THING	4	Right
RELP 4.8	None	PETER	SING	THING	4	Right
RELP 4.8	None	WOLF	SING	THING	4	Risht
RELP 4.8	None	WOLF	SING	THING	4	Right
RELP 4.8	None	PETER	SING	THING	4	Right
RELP 4.8	None	ALVA DETER	SING	THING	4 4	Right Right
RELP 4.8	None	PETER	ZEE	THEE Dank		Left
RELP 4.8	None	ALVA ₩OLF	BANK BANK	DANK	5 5	Left
RELP 4.8	None None	LYNNE	BANK	DANK	5	Left
RELP 4.8	yous	ALVA	BID	DID	5	Left
1/441 7+0	RUNG	HLYN	2.2.2.		₩	I V

	BACKGROUND	•	LEFT	RIGHT		
CÓDING	NOISE	PERSON	WORD	WORD	ATTRIBUTE	REPLY
RELP 4.8	None	PETER	BID	DID	5	Left
RELP 4.8	None	LYNNE	BID	DID	5	Left
RELF 4.8	None	ALVA	FAD	THAD	5	Left
RELP 4.8	None	PETER	FAD	THAD	5	Risht
RELF 4.8	None	ALVA	FIN	THIN	5	Right
RELF 4.8	None	PETER	FIN	THIN	5	Risht
RELF 4.8	None	PETER	FORE	THOR	5	Left
RELF 4.8	None	ALVA	FORE	THOR	5	Risht
RELF 4.8	None	WOLF	FORE	THOR	5	Risht
RELF 4.8	None	ALVA	FOUGHT	THOUGHT	5	Right
RELF 4.8	Norie	PETER	FOUGHT	THOUGHT	5	Right
RELF 4.8	None	LYNNE	FOUGHT	THOUGHT	5	Risht
RELF 4.8	None	ALVA	FOUGHT	THOUGHT	5	Left
RELF 4.8	Norie	WOLF	FOUGHT	THOUGHT	5	Risht
RELF 4.8	None	LYNNE	MET	NET	5	Risht
RELF 4.8	None	WOLF	MLT	NET	5	Risht
RELF 4.8	None	ALVA	MET	NET	5 .	Risht
RELF 4.8	None	LYNNE	MOON	иоои	5	Risht
RELF 4.8	None	LYNNE	PENT .	TENT	5	Risht
RELP 4.8	None	AL VA	YIELD	WIELD	. 6	Left

COLING	BACKGROUND NOISE	PERSON	LEFT	RIGHT WORD	ATTRIBUTE	REPLY
CODING	MOTOE	FERSON	WORD	MOUT	HIINTDUIE	KECLI
RELP 4.8	Armoured Car	ALVA	DAUNT	TAUNT	1	Left
	Armoured Car	PETER	DAUNT	TAUNT	i	Left
RELF 4.8	Armoured Car	PETER	VAST	FAST	1	Left
RELP 4.8	Armoured Car	LYNNE	VAST	FAST	1	Left
	rsO berwomrA	ALVA	VAST	FAST	1	Left
	Armoured Car	WOLF	VAULT	FAULT	, 1	Left
	Armoured Car	WOLF	VOLE	FOAL	1	Left
	Armoured Car	LYNNE	VOLE	FOAL	1	Left
RELF 4.8	Armoured Car	PETER	VOLE	FOAL	1	Left
	Armoured Car	ALVA	MAD	BAD	2	Neither
	Armoured Car	PETER	MEND	BEND	2	Risht
	Armoured Car	PETER	MOAN	BONE	2	Right
	Armoured Car	ALVA	NIP	DIP	2	Neither
	Armoured Car	LYNNE	SHOES	CHOOSE	3 3	Risht Risht
	Armoured Car	ALVA	SHOES	CHOOSE		Right
	Armoured Car	LYNNE	THEN	DEN	3 3	Left '
RELP 4.8	Armoured Car	WOLF	THONG	T ONG T ONG	3	Left
	Armoured Car Armoured Car	PETER ALVA	THONG THOSE	DOZE	3	Risht
RELP 4.8 RELP 4.8	Armoured Car	WOLF	THOSE	DOZE	3	Left
RELF 4+8	Armoured Car	LYNNE	THOUGH	DOUGH	3	Risht
RELF 4.8	Armoured Car	PETER	VEE	BEE	3	Risht
RELP 4.8	Armoured Car	WOLF	VEE	BEE	3	Risht
RELP 4.8	Armoured Car	ALVA	VILL	BILL	3	Risht
RELP 4.8	Armoured Car	LYNNE	VILL	BILL	3	Risht
RELF 4.8	Armoured Car	LYNNE	VDN	BON	3	Risht
RELP 4.8	Armoured Car	PETER	VON	BON	3	Risht
RELF 4.8	Armoured Car	ALVA	VOX	BOX	3	Risht
RELP 4.8	Armoured Car	LYNNE	VOX	BOX	3	Risht
RELF 4.8	Armoured Car	FETER	V OX	BOX	3	Risht
RELP 4.8	Armoured Car	LYNNE	CHEEP	KEEP	4	Left
RELP 4.8	Armoured Car	LYNNE	JAB	GAB	4	Left
RELF 4.8	Armoured Car	ALVA	JAB	GAB	4	Right
	Armoured Car	WOLF	JVB	GAB	4	Left
RELP 4.8	Armoured Car	LYNNE	JILT	GILT	4	Risht
	Armoured Car	PETER	JILT	GILT	4	Right
	Armoured Car	LYNNE	SANK	THANK	4	Risht
RELP 4.8	Armoured Car	WOLF	SANK SAW	THANK Thaw	4 4	Right Neither
RELP 4.8	Armoured Car	ALVA ALVA	SING	THING	4	Right
RELF 4.8 RELF 4.8	Armoured Car Armoured Car	LYNNE	SING	THING	4	Risht
RELP 4.8	Armoured Car	PETER	SING	THING	4	Risht
RELP 4.8	Armoured Car	ALVÄ	ZEE /	THEE	4	Right
RELP 4.8	Armoured Car	LYNNE	ZEE	THEE	4	Risht
RELP 4.8	Armoured Car		. ZEE	THEE	4	Right
RELP 4.8	Armoured Car	WOLF	ZEE	THEE	4	Right
RELP 4.8	Armoured Car	LYNNE	BANK	DANK	5	Left
RELF 4.8	Armoured Car	PETER	BANK	DANK	5	Left
RELF 4.8	Armoured Car	ALVA	BOWL	DOLE	5	Risht
RELP 4.8	Armoured Car	ALVA	FAD	THAD	5	Left
RELP 4.8	Armoured Car	LYNNE	FIŅ	THIN	5	Right

CODING	BACKGROUND NOISE	PERSON	LEFT WORD	RIGHT WDRD	ATTRIBUTE	REPLY
•		•				
RELF 4.8		PETER .	FIN	THIN	5	Right
RELF 4.8 RELF 4.8	Armoured Car Armoured Car	LYNNE PETER	FORE FOUGHT	THOR THOUGHT	5 5	Left Right
RELP 4.8	Armoured Car	ALVA	FOUGHT	THOUGHT	5	Left
RELP 4.8	Armoured Car	WOLF	FOUGHT	THOUGHT	5	Right
RELF 4.8	Armoured Car	PETER	FOUGHT	THOUGHT	5	Left
RELF 4.8	Armoured Car	ALVA	моом	иоои	5	Risht
RELP 4.8.	Armoured Car	ALVA	PEAK	TEAK	5	Left
RELF 4.8	Armoured Car	LYNNE	PEAK	TEAK	5	Left
RELP 4.8	Armoured Car	LYNNE	PENT	TENT	5	Risht
RELP 4.8	Armoured Car	PETER	PENT	TENT	5	Risht
RELF 4.8	Armoured Car	ALVA	GIIL	DILL	6	Risht
RELP 4.8	Armoured Car	WOLF	HIT	FIT	6	Risht
RELP 4.8	Armoured Car	PETER	HIT	FIT	6 .	Risht
RELP 4.8	Armoured Car	ALVĄ	KEY	TEA	6	Left
RELF 4.8	Armoured Car -	WOLF	KEY	TEA	6	Left

BACKGROUND	·	LEFT	RIGHT		
NOISE	PERSON	WORD	WORD	ATTRIBUTE	REPLY
**				·	
•			*		

	BACKGROUND		LEFT	RIGHT		* *
CODING	NOISE	PERSON	WORD	WORD	ATTRIBUTE	REPLY
					•	
RELP 4.8	Helicopter	LYNNE	DAUNT	TAUNT	4	Left
					1	
RELP 4.8	Helicorter	ALVA	DUNE	TUNE	1	Right
RELF 4.8	Helicopter	PETER	JOCK	CHOCK	1	Left
RELF 4.8	Helicorter	WOLF	VAST	FAST	1	Right
RELP 4.8	Helicopter	LYNNE	VAULT	FAULT	1	Left
RELF 4.8	Helicorter	WOLF	VAULT	FAULT	1	Left
RELP 4.8	Helicoster.	LYNNE	VEAL	FEEL	1	Left
RELP 4.8	Helicopter	PETER	VEAL	FEEL	1	Left
RELP 4.8	Helicorter	ALVA	VEAL	FEEL	1	Right
RELF 4.8	Helicopter	LYNNE	VOLE	FOAL	1	Left
RELP 4.8	Helicorter	LYNNE	VOLE	FOAL	i	Risht
*	•				1	Left
RELP 4.8	Helicorter	PETER	VOLE	FOAL		
RELF 4.8	Helicorter	ALVA	MOM	BOMB	2	Neither
RELF 4.8	Helicorter	PETER	HOH	BOMB	2	Risht
RELP 4.8	Helicopter	WOLF	TOOM	BOOT	2	Risht
RELP 4.8	Helicorter	PETER	HOOT	BOOT	2 2	Right
RELP 4.8	Helicopter	LYNNE	MOSS	BOSS	2	Right .
RELF 4.8	Helicorter	PETER	MOSS	B055	2	Right
RELP 4.8	Helicopter	ALVA	MOSS	B088		Risht
RELP 4.8	Helicorter	WOLF	NEWS	DUES	2 2 3	Left
RELP 4.8	Helicorter	LYNNE	FENCE	PENCE	- 3	Risht
RELP 4.8	Helicorter	PETER	FENCE	PENCE	3	Right
RELF 4.8	Helicorter	WOLF	FOO	F00	3	Left
RELP 4.8	Helicopter	ALVA	F00	F00	3	Neither
RELP 4.8	Helicopter	PETER	F00	PO0	. 3	Right
		WOLF				
RELP, 4.8	Helicorter		F00	P00	. 3	Risht
RELP 4.8	Helicorter	ALVA	F00	P00	3	Risht
RELP 4.8	Helicopter	ALVA	SHAD	CHAD	3	Right
RELP 4.8	Helicopter	LYNNE	SHAD	CHAD	3	Left
RELP 4.8	Helicorter	PETER	SHAW	CHAW	- 3	Risht
RELP 4.8	Helicorter	LYNNE	SHEET	CHEAT	3	Risht
RELP 4.8	Helicorter	LYNNE	SHOES	CHOOSE	3	Risht
RELP 4.8	Helicorter	WOLF	SHOES	CHOOSE	3	Risht
RELF 4.8	Helicorter	ALVA	SHOES	CHOOSE	3	Right
RELP 4.8	Helicorter	ALVA	THAN	DAN	3	Risht
RELP 4.8	Helicopter	PETER	THAN	DAN	3	Risht
RELP 4.8	Helicopter	ALVA	THEN	DEN	3	Left
RELF 4.8	Helicopter	LYNNE	THEN	DEN	3	Left
RELP 4.8	Helicopter	PETER	THEN	DEN	3	Left
RELF 4.8	Helicorter	WOLF	THEN	DEN		Left
RELF 4.8	Helicopter	LYNNE	THICK	TICK	7	Left
				TONG	3 3 3	Left
RELP 4.8	Helicorter	LYNNE	THONG		3	
RELP 4.8	Helicopter	LYNNE	THOSE	DOZE	ა -	Left
RELP 4.8	Helicopter	PETER	THOSE	DOZE	3	Right
RELF 4.8	Helicorter	ALVA	THOUGH	DOUGH	3	Neither
RELF 4.8	Helicorter	WOLF	THOUGH	DOUGH	3	Left
RELP 4.8	Helicorter	LYNNE	VEE	BEE	3	Risht
RELP 4.8	Helicorter		VEE	REE	3	Left
RELP 4.8	Helicorter	WOLF	VEE .	BEE	. 3	Right
RELP 4.8	Helicorter	ALVA	VEE	BEE	3	Left
RELP 4.8	Helicorter	LYNNE	VEE	BEE	3	Left
-						

CODING	BACKGROUND NOISE	PERSON	LEFT WORD	RIGHT WORD	ATTRIBUTE	REPLY
RELF 4.8	Helicopter	PETER	VEE	BEE	3	Left
RELP 4.8	Helicopter	PETER	VEE	BEE	3	Left
RELP 4.8	Helicorter	WOLF	VEE	BEE	3	Left
RELP 4.8	Helicorter	WOLF	VEE	BEE	3	Left
RELF 4.8	Helicorter	ALVA	VEE	BEE	3	Neither
RELF 4.8	Helicorter	LYNNE	VILL	BILL	3	Risht
RELF 4.8	Helicorter	ALVA	VILL	BILL	3	Risht
RELF 4.8	Helicorter	WOLF	VILL	BILL	3	Risht
RELP 4.8	Helicorter	PETER	VILL	BILL	3	Right
RELP 4.8	Helicorter	ALVA -	NOV	BON	3	Right
RELP 4.8	Helicorter	LYNNE	VON	BON	3 3	Risht
RELP 4.8	Helicorter	PETER	VON	BON	3	Risht
RELF 4.8	Helicorter	WOLF	VON	BON	3	Right
RELP 4.8	Helicopter	WOLF	VON	BON	7	Left
RELF 4.8	Helicorter	WOLF	VON	BON	3 3	Left >
RELF 4.8	Helicorter	LYNNE	vox	BOX	3	Right
RELF 4.8	Helicorter	ALVA	VOX	BOX	3	Right
RELF 4.8	Helicopter	WOLF	vox	BOX	3	Right
RELP 4.8	Helicorter	ALVA	VOX	BOX	3	Left
RELF 4.8	Helicopter	PETER	VOX	BOX	3	Right
RELF 4.8	Helicorter	LYNNE	CHAIR	CARE	4	Left
RELF 4.8	Helicorter	LYNNE	CHEEP	KEEP	4	Left
RELP 4.8	Helicorter	PETER	CHEEP	KEEP	4	Left
RELP 4.8	Helicorter	PETER	JAB	GAB	4	Left
RELP 4.8	Helicorter	PETER	JILT	GILT	4	Right
RELP 4.8	Helicopter	ALVA	JILT	GILT	4	Left
RELP 4.8	Helicorter	LYNNE	JILT	GILT	4	Left
RELF 4.8	Helicorter	PETER	JOT	GOT	4	Risht
RELP 4.8	Helicorter	ALVA	SANK	THANK	4	Right
RELP 4.8	Helicorter	ALVA	SANK	THANK	4	Left
RELP 4.8	Helicorter	LYNNE	SING	THING	4	Right
RELP 4.8	Helicorter	ALVA	SING	THING	4	Right
RELP 4.8	Helicorter	WOLF	SING	THING	4	Risht
RELF 4.B	Helicorter	PETER	SING	THING	4	Risht
RELP 4.8	Helicorter	LYNNE	ZEE	THEE	4	Right
RELF 4.8	Helicorter	PETER	ZEE	THEE	4	Right
RELP 4.8	Helicorter	ALVA	BANK	DANK	5	Left
RELF 4.8	Helicorter	LYNNE	BANK	DANK	5	Left
RELP 4.8	Helicorter	PETER	BANK	DANK	5	Left
RELF 4.8	Helicorter	ALVA	FAD	THAD	5	Left
RELP 4.8	Helicorter	LYNNE	FAD	THAD	5	Left
RELP 4.8	Helicorter	LYNNE	FAD	THAD	5 5	Right
RELF 4.8	Helicorter	ALVA	FAD	THAD		Neither
RELP 4.8	Helicopter	LYNNE	FAD	THAD	5 5	Right Right
RELP 4.8	Helicoster	PETER	FAD FAD	AAHT AAHT	ა 5	Right
RELP 4.8 RELP 4.8	Helicorter Helicorter	PETER WOLF	FAD	THAD	ว 5	Right
RELF 4.8	Helicorter Helicorter	ALVA	FIN	THIN	5 5	Right
RELP 4.8	Helicopter	LYNNE	FIN	THIN	5	Right
RELF 4.8	Helicopter	LYNNE	FIN	THIN	5	Left
RELP 4.8	Helicopter	PETER	FIN	THIN	5	Right
RELP 4.8	Helicopter	LYNNE	FIN	THIN	5	Left
					•	

	BACKGROUND		LEFT	RIGHT		
CODING	NOISE	PERSON	WORD	WORD	ATTRIBUTE	REPLY
RELP 4.8	Helicopter	PETER	FIN	THIN	5	Left
RELP 4.8	Helicopter	PETER	FIN	THIN	5	Left
RELP 4.8	Helicopter	WOLF	FIN	THIN	5	Left
RELP 4.8	Helicorter	WOLF	FIN	THIN	5	Left
RELP 4.8	Helicopter	LYNNE	FOUGHT	THOUGHT	5	Right
RELP 4.8	Helicorter	A! VA	MET	NET	5	Right
RELP 4.8	Helicorter	LYNNE	MET	NET	5	Left
RELP 4.8	Helicorter	WOLF	MET	NET	5	Left
RELF 4.8	Helicorter	PETER	MET	NET	5	Left
RELP 4.8	Helicorter	ALVA	PEAK	TEAK	5	Left
RELP 4.8	Helicorter	PETER	PEAK	TEAK	5	Risht
RELF 4.8	Helicorter	ALVA	PENT	TENT	5	Left
RELP 4.8	Helicorter	WOLF	PENT	TENT	5	Left
RELP 4.8	Helicorter	LYNNE	FOOL	TOOL	5	Risht
RELP 4.8	Helicorter	LYNNE	GILL	DILL	6	Right
RELP 4.8	Helicopter	LYNNE	GILL	DILL	6	Left
RELP 4.8	Helicopter	PETER	GILL	DILL	6	Right
RELP 4.8	Helicorter	PETER	GILL	DILL	6	Left
RELF 4.8	Helicopter	WOLF	GILL	DILL	6	Left
RELF 4.8	Helicoster	ALVA	KEG.	PEG	6	Neither
RELF 4.8	Helicopter	ALVA	KEY	TEA	6	Neither
RELP 4.8	Helicorter	LYNNE	KEY	TEA	6	Left
RELP 4.8	Helicopter	LYNNE	KEY	TEA	6	Risht
RELF 4.8	Helicorter	ALVA	KEY	TEA	6	Right
RELP 4.8	Helicopter	PETER	KEY	TEA	6	Risht
RELF 4.8	Helicorter	ALVA	KEY	TEA	6	Right

CODING	BACKGROUND NOISE	PERSON	LEFT WORD	RIGHT WORD	ATTRIBUTE	REPLY
RELF 9.6	None	LYNNE	воми	F DND	1	Left
RELP 9.6	None	LYNNE	DAUNT	TAUNT	i	Left
RELP 9.6	None	PETER	VOLE	FOAL	î	Left
RELP 9.6	None	WOLF	VOLE	FOAL	i	Left
RELP 9.6	None	PETER	F00	P00	3	Right
RELF 9.6	None	LYNNE	SHAD	CHAD	3	Risht
RELP 9.6	None	WOLF	SHAD	CHAD	3	Right
RELP 9.6	None	ALVA	SHAD	CHAD	3	Right
RELP 9.6	None	WOLF	SHAW	CHAW	3	Risht
RELF 9.6	None	ALVA	SHAW	CHAW	3	Right
RELP 9.6	None	PETER	SHAW	CHAW	3	Risht
RELF 9.6	None	WOLF	SHOES	CHOOSE	3	Risht
RELF 9+6 -	None	ALVA	VILL .	BILL	3	Right
RELF 9.6	None	PETER	VOX	BOX	. 3	Left
RELF 9.6	None	FETER	XOV	BOX	. 3	Risht
RELP 9.6	None	LYNNE	VOX	BOX	3	Risht
RELF 9.6	None	ALVA	JEST	GUEST	4	Left
RELP 9.6	None	PETER	JILT	GILT	4	Risht
RELF 9.6	Mone	LYNNE	SANK	THANK	4	Risht
RELP 9.6	None	ALVA	SING	THING	4	Risht
RELP 9.6	None	LYNNE	SING	THING	4	Risht
RELF 9.6	None	WOLF	SING	THING	4	Risht
RELP 9.6	None	ALVA	BID	DID	5	Left
RELF 9.6	None	LYNNE	BID	DID	5	Left
RELP 9.6	None	LYNNE	FAD	THAD	5	Risht
RELF 9.6	Notic .	ALVA	FAD	THAD	5	Left
RELP 9.6	None	PETER	FIN	THIN	5	Risht
RELP 9.6	None	WOLF	FORE	THOR	5	Risht
RELP 9.+6	None	LYNNE	FORE .	THOR	5	Left
RELP 9.6	None	LYNNE	PENT:	TENT	· 5	Left
RELP 9.6	None :	LYNNE	GILL	DILL	6	Left

RELP 9.6 Armoured Car ALVA VOLE FOAL 1 Left RELP 9.6 Armoured Car PETER FENCE PENCE 3 Risht RELP 9.6 Armoured Car PETER FENCE PENCE 3 Risht RELP 9.6 Armoured Car LYNNE FENCE PENCE 3 Risht RELP 9.6 Armoured Car WOLF FOO POO 3 Risht RELP 9.6 Armoured Car WOLF FOO POO 3 Risht RELP 9.6 Armoured Car WOLF SHAB CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAB CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAB CHAD 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISHT RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISHT RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISHT RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISHT RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISHT RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISHT RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISHT RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISHT RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISHT RELP 9.6 Armoured Car FETER SHE SHE ARISHT RELP 9.6 Armoured Car FETER SHE SHE ARISHT RELP 9.6 Armoured Car PETER FIN SHING A RISHT RELP 9.6 Armoured Car PETER FIN SHING A RISHT RELP 9.6 Armoured Car ALVA SHE SHE SHE ARISHT RELP 9.6 Armoured Car ALVA SHE SHE SHE ARISHT RELP 9	CODING	BACKGROUND NOISE	PERSON	LEFT WORD	RIGHT	ATTRIBUTE	REPLY
RELP 9.6 Armoured Car ALVA VOLE FOAL 1 Left RELP 9.6 Armoured Car PETER FENCE PENCE 3 Risht RELP 9.6 Armoured Car WOLF FOD POO 3 Risht RELP 9.6 Armoured Car WOLF FOD POO 3 Risht RELP 9.6 Armoured Car WOLF FOD POO 3 Risht RELP 9.6 Armoured Car WOLF SHAD CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAM 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAM 3 Risht RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF THAN DAN 3 Risht RELP 9.6 Armoured Car WOLF THAN DAN 3 Risht RELP 9.6 Armoured Car WOLF THAN DAN 3 Risht RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car WOLF THAN DAN 3 Risht RELP 9.6 Armoured Car WOLF THAN DAN 3 Risht RELP 9.6 Armoured Car WOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car WOLF VON BON 3 Risht RELP 9.6 Armoured Car WOLF VON BON 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car PETER FIN THING 4 Risht RELP 9.6 Armoured Car PETER FIN THING 4 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP	RELP 9.6	Armoured Car	LYNNE	VAST	FAST	1	Left
RELP 9.6 Armoured Car PETER FENCE PENCE 3 Risht RELP 9.6 Armoured Car WOLF FOO POO 3 Risht RELP 9.6 Armoured Car WOLF FOO POO 3 Risht RELP 9.6 Armoured Car WOLF SHAR CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAR CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAR CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAR CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAR CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAR CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAR CHAM 3 Risht RELP 9.6 Armoured Car ALVA SHAR CHAM 3 Risht RELP 9.6 Armoured Car ALVA SHAR CHAM 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 RISht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 RISht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 RISht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 RISht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 RISht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 RISht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 RISht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 RISht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 RISht RELP 9.6 Armoured Car WOLF VON BON 3 RISHT RELP 9.6 Armoured Car WOLF VON BON 3 RISHT RELP 9.6 Armoured Car WOLF VON BON 3 RISHT RELP 9.6 Armoured Car WOLF VON BON 3 RISHT RELP 9.6 Armoured Car ALVA VOX BOX 3 RISHT RELP 9.6 Armoured Car ALVA VOX BOX 3 RISHT RELP 9.6 Armoured Car PETER VOX BOX 3 RISHT RELP 9.6 Armoured Car PETER JILT GILT 4 RISHT RELP 9.6 Armoured Car PETER JILT GILT 4 RISHT RELP 9.6 Armoured Car PETER JILT GILT 4 RISHT RELP 9.6 Armoured Car PETER SING THING 4 RISHT RELP 9.6 Armoured Car PETER SING THING 4 RISHT RELP 9.6 Armoured Car PETER SING THING 4 RISHT RELP 9.6 Armoured Car PETER SING THING 4 RISHT RELP 9.6 Armoured Car PETER SING THING 4 RISHT RELP 9.6 Armoured Car PETER FIN THIN 5 RISHT RELP 9.6 Armoured Car PETER FIN THIN 5 RISHT RELP 9.6 Armoured Car WOLF FIN THIN 5 RISHT RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6				VOLE	FOAL	1	Left
RELP 9.6 Armoured Car WOLF FOO POO 3 Right RELP 9.6 Armoured Car WOLF FOO POO 3 Right RELP 9.6 Armoured Car WOLF SHAD CHAD 3 Right RELP 9.6 Armoured Car WOLF SHAD CHAD 3 Right RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Right RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Right RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Right RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Right RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Right RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Right RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Right RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Right RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Right RELP 9.6 Armoured Car MOLF SHEET CHEAT 3 Right RELP 9.6 Armoured Car MOLF SHEET CHEAT 3 Right RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car MOLF VON BON 3 Right RELP 9.6 Armoured Car MOLF VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VOX BOX 3 Right RELP 9.6 Armoured Car ALVA VOX BOX 3 Right RELP 9.6 Armoured Car ALVA VOX BOX 3 Right RELP 9.6 Armoured Car ALVA VOX BOX 3 Right RELP 9.6 Armoured Car PETER JILT GILT 4 Right RELP 9.6 Armoured Car PETER SING THING 4 Right RELP 9.6 Armoured Car PETER SING THING 4 Right RELP 9.6 Armoured Car PETER SING THING 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA FORE THEE 5 RIGHT THOUGHT 5 Right RELP 9.6 Armoured Car ALVA FORE THOUGHT THOUGHT 5 Right RELP 9.6 Armoured Car ALVA FORE THOUGHT THOUGHT 5 Right RE				FENCE	PENCE	.3	Risht
RELP 9.6 Armoured Car PETER FOO POO 3 Risht RELP 9.6 Armoured Car WOLF SHAD CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Risht RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF USE BEE 3 Left RELP 9.6 Armoured Car WOLF USE BEE 3 Left RELP 9.6 Armoured Car WOLF USE BEE 3 Left RELP 9.6 Armoured Car WOLF USE BEL 3 Risht RELP 9.6 Armoured Car ALVA USE BON 3 Risht RELP 9.6 Armoured Car ALVA USE BON 3 Risht RELP 9.6 Armoured Car ALVA USE BON 3 Risht RELP 9.6 Armoured Car PETER USX BOX 3 Risht RELP 9.6 Armoured Car PETER USX BOX 3 Risht RELP 9.6 Armoured Car PETER USX BOX 3 Risht RELP 9.6 Armoured Car PETER USX BOX 3 Left RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 5 Risht RELP 9.6 Armoured Car MOLF FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THEE 4 RISht RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THEE ARMOURED Car	RELP 9.6	Armoured Car	LYNNE	FENCE	PENCE	3	Risht
RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAD CHAD 3 Risht RELP 9.6 Armoured Car ALVA SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car ALVA SHAW CHAW 3 Risht RELP 9.6 Armoured Car ALVA SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA THAN DAN 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Risht RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car PETER JILT GILT 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car ALVA SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 5 Risht RELP 9.6 Armoured Car ALVA FORE THEE 4 Risht RELP 9.6 Armoured Car ALVA FORE THEE 4 Risht RELP 9.6 Armoured Car ALVA FORE THEE 4 Risht RELP 9.6 Armoured Car ALVA FORE THEE 4 Risht RELP 9.6 Armoured Car ALVA FORE THEE 4 Risht RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5	RELP 9.6	Armoured Car	WOLF	F00	P00	3	Risht
RELP 9.6 Armoured Car ALVA SHAD CHAW 3 Risht RELP 9.6 Armoured Car ALVA SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car MOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car ALVA THAN DAN 3 Risht RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BOX 3 Left RELP 9.6 Armoured Car MOLF VOX BOX 3 Left RELP 9.6 Armoured Car MOLF VOX BOX 3 Left RELP 9.6 Armoured Car MOLF VOX BOX 3 Left RELP 9.6 Armoured Car MOLF VOX BOX 3 Left RELP 9.6 Armoured Car MOLF VOX BOX 3 Left RELP 9.6 Armoured Car MOLF VOX BOX 3 Left RELP 9.6 Armoured Car MOLF VOX BOX 3 Left RELP 9.6 Armoured Car MOLF VOX BOX 3 Left RELP 9.6 Armoured Car LYNNE SANK THANK 4 RISHT RELP 9.6 Armoured Car LYNNE SANK THANK 4 RISHT RELP 9.6 Armoured Car LYNNE SANK THANK 4 RISHT RELP 9.6 Armoured Car LYNNE SING THING 4 RISHT RELP 9.6 Armoured Car DETER SING THING 4 RISHT RELP 9.6 Armoured Car MOLF ZEE THEE 4 RISHT RELP 9.6 Armoured Car MOLF ZEE THEE 4 RISHT RELP 9.6 Armoured Car MOLF ZEE THEE 4 RISHT RELP 9.6 Armoured Car MOLF FIN THIN 5 RISHT RELP 9.6 Armoured Car MOLF FIN THIN 5 RISHT RELP 9.6 Armoured Car MOLF FIN THIN 5 RISHT RELP 9.6 Armoured Car MOLF FIN THIN 5 RISHT RELP 9.6 Armoured Car MOLF FIN THIN 5 RISHT RELP 9.6 Armoured Car MOLF FIN THIN 5 RISHT RELP 9.6 Armoured Car MOLF FIN THIN 5 RISHT RELP 9.6 Armoured Car MOLF FIN THIN 5 Left RELP 9.6 Armoured Car MOLF FIN THIN 5 Left RELP 9.6 Armoured Car MOLF FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THEE THORST THOUGHT 5 Le	RELF 9.6	Armoured Car	PETER	F00	P00	. 3	Risht
RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHAW CHAW 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Risht RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car MUDF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car MUDF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car MUDF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car MUDF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car MUDF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car MUDF UDN BON 3 Risht RELP 9.6 Armoured Car MUDF UDN BON 3 Risht RELP 9.6 Armoured Car MUDF UDN BON 3 Risht RELP 9.6 Armoured Car MUDF UDN BON 3 Risht RELP 9.6 Armoured Car MUDF UDN BON 3 Risht RELP 9.6 Armoured Car MUDF UDX BOX 3 Risht RELP 9.6 Armoured Car MUDF UDX BOX 3 Risht RELP 9.6 Armoured Car MUDF UDX BOX 3 Risht RELP 9.6 Armoured Car MUDF UDX BOX 3 Risht RELP 9.6 Armoured Car MUDF UDX BOX 3 Risht RELP 9.6 Armoured Car MUDF UDX BOX 3 Left RELP 9.6 Armoured Car MUDF UDX BOX 3 Left RELP 9.6 Armoured Car MUDF UDX BOX 3 Left RELP 9.6 Armoured Car MUDF UDX BOX 3 Left RELP 9.6 Armoured Car MUDF UDX BOX 4 Risht RELP 9.6 Armoured Car MUDF SING THING 4 Risht RELP 9.6 Armoured Car MUDF SING THING 4 Risht RELP 9.6 Armoured Car MUDF SING THING 4 Risht RELP 9.6 Armoured Car MUDF SING THING 4 Risht RELP 9.6 Armoured Car MUDF SING THING 4 Risht RELP 9.6 Armoured Car MUDF SING THING 5 Risht RELP 9.6 Armoured Car MUDF FIN THIN 5 Risht RELP 9.6 Armoured Car MUDF FIN THIN 5 Risht RELP 9.6 Armoured Car MUDF FIN THIN 5 Risht RELP 9.6 Armoured Car MUDF FIN THIN 5 Risht RELP 9.6 Armoured Car MUDF FIN THIN 5 Risht RELP 9.6 Armoured Car MUDF FIN THIN 5 Risht RELP 9.6 Armoured Car MUDF FIN THIN 5 Risht RELP 9.6 Armoured Car MUDF FIN THIN 5 Risht RELP 9.6 Armoured Car MUDF FIN THIN 5 Left RELP 9.6 Armoured Car MUDF FIN THIN 5 Left RELP 9.6 Armoured Car MUDF FIN THIN 5 Left RELP 9.6 Armoured Car MUDF FIN THINGHT 5 Left	RELF 9.6	Armoured Car	WOLF	SHAD	CHAD		
RELP 9.6 Armoured Car PETER SHAW CHAW 3 Risht RELP 9.6 Armoured Car PETER SHAW CHAW 3 Risht RELP 9.6 Armoured Car ALVA SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car MOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car MOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car MOLF THAN DAN 3 Risht RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car MOLF VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BOX 3 Risht RELP 9.6 Armoured Car DOLF VOX BOX 3 Risht RELP 9.6 Armoured Car DOLF VOX BOX 3 Risht RELP 9.6 Armoured Car DOLF VOX BOX 3 Risht RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Risht RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF VOX BOX 3 Left RELP 9.6 Armoured Car DOLF RELP SING THING 4 RISHT RELP 9.6 Armoured Car DOLF RELP SING THING 4 RISHT RELP 9.6 Armoured Car DOLF RELP SING THING 4 RISHT RELP 9.6 Armoured Car DOLF RE	RELF 9.6	Armoured Car	ALVA	SHAD	CHAD		
RELP 9.6 Armoured Car PETER SHAW CHAW 3 Risht RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Risht RELP 9.6 Armoured Car LYNNE SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Risht RELP 9.6 Armoured Car WOLF THAN DAN 3 Risht RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car WOLF VDN BON 3 Risht RELP 9.6 Armoured Car WOLF VDN BON 3 Risht RELP 9.6 Armoured Car ALVA VDN BON 3 Risht RELP 9.6 Armoured Car ALVA VDN BON 3 Risht RELP 9.6 Armoured Car PETER VDX BOX 3 Risht RELP 9.6 Armoured Car PETER VDX BOX 3 Risht RELP 9.6 Armoured Car PETER JILT GILT 4 Risht RELP 9.6 Armoured Car PETER JILT GILT 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car PETER ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armour	RELF 9.6	Armoured Car	ALVA	SHAW			
RELP 9.6 Armoured Car	RELF 9.6	Armoured Car					
RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Right RELP 9.6 Armoured Car WOLF SHEET CHEAT 3 Right RELP 9.6 Armoured Car ALVA THAN DAN 3 Right RELP 9.6 Armoured Car WOLF THAN DAN 3 Right RELP 9.6 Armoured Car WOLF THAN DAN 3 Right RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VILL BILL 3 Right RELP 9.6 Armoured Car WOLF VON BON 3 Right RELP 9.6 Armoured Car WOLF VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VOX BOX 3 Right RELP 9.6 Armoured Car PETER VOX BOX 3 Right RELP 9.6 Armoured Car PETER VOX BOX 3 Right RELP 9.6 Armoured Car PETER VOX BOX 3 Left RELP 9.6 Armoured Car PETER JILT GILT 4 Right RELP 9.6 Armoured Car LYNNE SANK THANK 4 Right RELP 9.6 Armoured Car LYNNE SANK THING 4 Right RELP 9.6 Armoured Car ALVA SING THING 4 Right RELP 9.6 Armoured Car PETER SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car DETER ZEE THEE 4 Right RELP 9.6 Armoured Car WOLF ZEE THEE 4 Right RELP 9.6 Armoured Car WOLF ZEE THEE 4 Right RELP 9.6 Armoured Car WOLF ZEE THEE 4 Right RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA							
RELP 9.6 Armoured Car ALVA THAN DAN 3 Risht RELP 9.6 Armoured Car ALVA THAN DAN 3 Risht RELP 9.6 Armoured Car WOLF THAN DAN 3 Risht RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELF 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car WOLF VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car PETER JILT GILT 4 Risht RELP 9.6 Armoured Car PETER JILT GILT 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car ALVA SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured		•					
RELP 9.6 Armoured Car ALVA THAN DAN 3 Right RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VILL BILL 3 Right RELP 9.6 Armoured Car LYNNE VILL BILL 3 Right RELP 9.6 Armoured Car WOLF VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car PETER VOX BOX 3 Right RELP 9.6 Armoured Car PETER VOX BOX 3 Right RELP 9.6 Armoured Car PETER VOX BOX 3 Left RELP 9.6 Armoured Car PETER JILT GILT 4 Right RELP 9.6 Armoured Car LYNNE SANK THANK 4 Right RELP 9.6 Armoured Car LYNNE SANK THANK 4 Right RELP 9.6 Armoured Car LYNNE SANK THANK 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 5 Right RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Right RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Right RELP 9.6 Armoured Car MOLF ZEE THEE 4 Right RELP 9.6 Armoured Car MOLF ZEE THEE 4 Right RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Right RELP 9.6 Armoured Car MOLF FIN THIN 5 Right RELP 9.6 Armoured Car MOLF FIN THIN 5 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car MOLF FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car A						3	
RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car WOLF VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VOX BOX 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car WOLF VOX BOX 3 Risht RELP 9.6 Armoured Car PETER JILT GILT 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car ALVA SING THING 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 Risht RELP 9.6 Armoured Car ALVA HET NET 5 RISHT						3	
RELP 9.6 Armoured Car LYNNE VEE BEE 3 Left RELF 9.6 Armoured Car A!VA VILL BILL 3 Risht RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car WOLF VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BOX 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car WOLF VOX BOX 3 Left RELP 9.6 Armoured Car PETER JILT GILT 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car PETER ZEE THEE 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT DILL 6 Left				•		<u>3</u>	
RELP 9.6 Armoured Car AIVA VILL BILL 3 Right RELP 9.6 Armoured Car LYNNE VILL BILL 3 Right RELP 9.6 Armoured Car WOLF VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car PETER VOX BOX 3 Right RELP 9.6 Armoured Car WOLF VOX BOX 3 Right RELP 9.6 Armoured Car PETER JILT GILT 4 Right RELP 9.6 Armoured Car LYNNE SANK THANK 4 Right RELP 9.6 Armoured Car LYNNE SANK THANK 4 Right RELP 9.6 Armoured Car FETER SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THEE 4 Right RELP 9.6 Armoured Car LYNNE SING THEE 4 Right RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Right RELP 9.6 Armoured Car WOLF ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Right RELP 9.6 Armoured Car PETER FIN THIN 5 Right RELP 9.6 Armoured Car PETER FIN THIN 5 Right RELP 9.6 Armoured Car PETER FIN THIN 5 Right RELP 9.6 Armoured Car WOLF FIN THIN 5 Right RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Left RELP 9.6 Armoured Car ALVA HET NET 5 Left RELP 9.6 Armoured Car ALVA HET NET 5 Le						ა 7	
RELP 9.6 Armoured Car LYNNE VILL BILL 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VON BON 3 Risht RELP 9.6 Armoured Car ALVA VOX BOX 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car PETER VOX BOX 3 Left RELF 9.6 Armoured Car PETER JILT GILT 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THEE 4 Risht RELP 9.6 Armoured Car LYNNE SING THEE 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 Left RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISht RELP 9.6 Armoured Car ALVA HET NET 5 RISHT RELP 9.6 Armoured Car ALVA HET NET					·		
RELP 9.6 Armoured Car		•	*		*		
RELP 9.6 Armoured Car ALVA VON BON 3 Right RELP 9.6 Armoured Car ALVA VOX BOX 3 Right RELP 9.6 Armoured Car PETER VOX BOX 3 Right RELP 9.6 Armoured Car WOLF VOX BOX 3 Left RELP 9.6 Armoured Car PETER JILT GILT 4 Right RELP 9.6 Armoured Car LYNNE SANK THANK 4 Right RELP 9.6 Armoured Car ALVA SING THING 4 Right RELP 9.6 Armoured Car PETER SING THING 4 Right RELP 9.6 Armoured Car PETER SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car PETER ZEE THEE 4 Right RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Right RELP 9.6 Armoured Car MOLF ZEE THEE 4 Right RELP 9.6 Armoured Car MULF ZEE THEE 4 Right RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Right RELP 9.6 Armoured Car WOLF FIN THIN 5 Right RELP 9.6 Armoured Car WOLF FIN THIN 5 Right RELP 9.6 Armoured Car WOLF FIN THIN 5 Right RELP 9.6 Armoured Car WOLF FIN THIN 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FOREHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOREHT THOUGHT 5 Right RELP 9.6 Armoured Car ALVA FOREHT THOUGHT 5 Right RELP 9.6 Armoured Car ALVA HET 5 RIght							
RELP 9.6 Armoured Car ALVA VOX BOX 3 Right RELP 9.6 Armoured Car PETER VOX BOX 3 Right RELP 9.6 Armoured Car WOLF VOX BOX 3 Left RELP 9.6 Armoured Car PETER JILT GILT 4 Right RELP 9.6 Armoured Car LYNNE SANK THANK 4 Right RELP 9.6 Armoured Car ALVA SING THING 4 Right RELP 9.6 Armoured Car PETER SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car LYNNE SING THING 4 Right RELP 9.6 Armoured Car PETER ZEE THEE 4 Right RELP 9.6 Armoured Car WOLF ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Right RELP 9.6 Armoured Car WOLF FIN THIN 5 Right RELP 9.6 Armoured Car WOLF FIN THIN 5 Right RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA HET NET 5 Right RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA KEY TEA	and the second s					3 7	
RELP 9.6 Armoured Car PETER VOX BOX 3 Risht RELP 9.6 Armoured Car WOLF VOX BOX 3 Left RELP 9.6 Armoured Car PETER JILT GILT 4 Risht RELP 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELP 9.6 Armoured Car ALVA SING THING 4 Risht RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET 6 Left RELP 9.6 Armoured Car ALVA MET							
RELF 9.6 Armoured Car WOLF VOX BOX 3 Left RELF 9.6 Armoured Car PETER JILT GILT 4 Risht RELF 9.6 Armoured Car LYNNE SANK THANK 4 Risht RELF 9.6 Armoured Car ALVA SING THING 4 Risht RELF 9.6 Armoured Car FETER SING THING 4 Risht RELF 9.6 Armoured Car LYNNE SING THING 4 Risht RELF 9.6 Armoured Car LYNNE SING THING 4 Risht RELF 9.6 Armoured Car LYNNE SING THING 4 Risht RELF 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELF 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELF 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELF 9.6 Armoured Car PETER FAB THAB 5 Left RELF 9.6 Armoured Car PETER FIN THIN 5 Risht RELF 9.6 Armoured Car WOLF FIN THIN 5 Risht RELF 9.6 Armoured Car WOLF FIN THIN 5 Risht RELF 9.6 Armoured Car WOLF FIN THIN 5 Risht RELF 9.6 Armoured Car PETER FIN THIN 5 Left RELF 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left RELP 9.6 Armoured Car ALVA MET NET 6 Left		*				3 7	
RELF 9.6 Armoured Car PETER JILT GILT 4 Risht RELF 9.6 Armoured Car ALVA SING THING 4 Risht RELF 9.6 Armoured Car FETER SING THING 4 Risht RELF 9.6 Armoured Car LYNNE SING THING 4 Risht RELF 9.6 Armoured Car LYNNE SING THING 4 Risht RELF 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELF 9.6 Armoured Car NOLF ZEE THEE 4 Risht RELF 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELF 9.6 Armoured Car PETER FIN THIN 5 RISHT RELF 9.6 Armoured Car PETER FIN THIN 5 RISHT RELF 9.6 Armoured Car PETER FIN THIN 5 RISHT RELF 9.6 Armoured Car PETER FIN THIN 5 RISHT RELF 9.6 Armoured Car WOLF FIN THIN 5 RISHT RELF 9.6 Armoured Car WOLF FIN THIN 5 RISHT RELF 9.6 Armoured Car WOLF FIN THIN 5 Left RELF 9.6 Armoured Car PETER FIN THIN 5 Left RELF 9.6 Armoured Car ALVA FORE THOR 5 RISHT RELP 9.6 Armoured Car ALVA FORE THOR 5 RISHT RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FORE THOUGHT 5 RISHT RELF 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 RISHT RELF 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 RISHT RELF 9.6 Armoured Car ALVA GILL BILL 6 Left RELF 9.6 Armoured Car ALVA GILL BILL 6 Left RELF 9.6 Armoured Car ALVA GILL BILL 6 Left RELF 9.6 Armoured Car ALVA GILL BILL 6 Left RELF 9.6 Armoured Car ALVA GILL BILL 6 Left RELF 9.6 Armoured Car ALVA KEY TEA 6 Left							
RELF 9.6 Armoured Car ALVA SING THING 4 Right RELF 9.6 Armoured Car ALVA SING THING 4 Right RELF 9.6 Armoured Car PETER SING THING 4 Right RELF 9.6 Armoured Car LYNNE SING THING 4 Right RELF 9.6 Armoured Car LYNNE ZEE THEE 4 Right RELF 9.6 Armoured Car MOLF ZEE THEE 4 Right RELF 9.6 Armoured Car ALVA ZEE THEE 4 Right RELF 9.6 Armoured Car PETER FAD THAD 5 Left RELF 9.6 Armoured Car PETER FIN THIN 5 Right RELF 9.6 Armoured Car WOLF FIN THIN 5 Right RELF 9.6 Armoured Car WOLF FIN THIN 5 Right RELF 9.6 Armoured Car WOLF FIN THIN 5 Right RELF 9.6 Armoured Car WOLF FIN THIN 5 Right RELF 9.6 Armoured Car WOLF FIN THIN 5 Right RELF 9.6 Armoured Car ALVA FORE THOR 5 Right RELF 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA KEY TEA		· ·					
RELP 9.6 Armoured Car ALVA SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA KEY TEA 6 Left		· · · · · · · · · · · · · · · · · · ·					
RELP 9.6 Armoured Car PETER SING THING 4 Risht RELP 9.6 Armoured Car LYNNE SING THING 4 Risht RELP 9.6 Armoured Car PETER ZEE THEE 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 RISHT RELP 9.6 Armoured Car ALVA MET NET 5 RISHT RELP 9.6 Armoured Car ALVA MET NET 5 RISHT RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA KEY TEA 6 Left		,					
RELP 9.6 Armoured Car Lynne SING THING 4 Risht RELP 9.6 Armoured Car PETER ZEE THEE 4 Risht RELP 9.6 Armoured Car Lynne ZEE THEE 4 Risht RELP 9.6 Armoured Car NOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car Lynne FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA KEY TEA 6 Left		•					
RELP 9.6 Armoured Car PETER ZEE THEE 4 Risht RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car PETER HOP FOP 6 Risht RELP 9.6 Armoured Car ALVA KEY TEA 6 Left							
RELP 9.6 Armoured Car LYNNE ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA KEY TEA 6 Left			•				
RELP 9.6 Armoured Car WOLF ZEE THEE 4 Risht RELP 9.6 Armoured Car ALVA ZEE THEE 4 Risht RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA KEY TEA 6 Left							Risht
RELP 9.6 Armoured Car ALVA ZEE THEE 4 Right RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Right RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Right RELP 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA MET NET 5 Right RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA KEY TEA 6 Left					THEE	4	Risht
RELP 9.6 Armoured Car PETER FAD THAD 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Risht RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Risht RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Risht RELP 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA MET NET 5 Risht RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car PETER HOP FOP 6 Risht RELP 9.6 Armoured Car ALVA KEY TEA 6 Left	•		ALVA	ZEE	THEE	4	Risht
RELP 9.6 Armoured Car PETER FIN THIN 5 Right RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Right RELP 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA MET NET 5 Right RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car PETER HOP FOP 6 Right RELP 9.6 Armoured Car ALVA KEY TEA 6 Left		Armoured Car	PETER	FAD	THAD	5	
RELP 9.6 Armoured Car WOLF FIN THIN 5 Left RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Right RELP 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA MET NET 5 Right RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car PETER HOP FOP 6 Right RELP 9.6 Armoured Car ALVA KEY TEA 6 Left	,		PETER	FIN	THIN		
RELP 9.6 Armoured Car PETER FIN THIN 5 Left RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Right RELP 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA MET NET 5 Right RELP 9.6 Armoured Car ALVA GILL BILL 6 Left RELP 9.6 Armoured Car PETER HOP FOP 6 Right RELP 9.6 Armoured Car ALVA KEY TEA 6 Left	RELF 9.6	Armoured Car	WOLF	FIN	THIN	5	
RELP 9.6 Armoured Car ALVA FORE THOR 5 Right RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Right RELP 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA MET NET 5 Right RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car PETER HOP FOP 6 Right RELP 9.6 Armoured Car ALVA KEY TEA 6 Left	RELP 9.6	Armoured Car	WOLF	FIN		5	
RELP 9.6 Armoured Car ALVA FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car WOLF FOUGHT THOUGHT 5 Right RELP 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELP 9.6 Armoured Car ALVA MET NET 5 Right RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car PETER HOP FOP 6 Right RELP 9.6 Armoured Car ALVA KEY TEA 6 Left	RELF 9.6	Armoured Car				5	
RELF 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELF 9.6 Armoured Car ALVA MET NET 5 Right RELF 9.6 Armoured Car ALVA GILL DILL 6 Left RELF 9.6 Armoured Car PETER HOP FOP 6 Right RELP 9.6 Armoured Car ALVA KEY TEA 6 Left	RELP 9.6	Armoured Car	ALVA	FORE		5	
RELF 9.6 Armoured Car LYNNE FOUGHT THOUGHT 5 Left RELF 9.6 Armoured Car ALVA MET NET 5 Right RELF 9.6 Armoured Car ALVA GILL DILL 6 Left RELF 9.6 Armoured Car PETER HOP FOP 6 Right RELP 9.6 Armoured Car ALVA KEY TEA 6 Left						5	
RELP 9.6 Armoured Car ALVA MET NET 5 Right RELP 9.6 Armoured Car ALVA GILL BILL 6 Left RELP 9.6 Armoured Car PETER HOP FOP 6 Right RELP 9.6 Armoured Car ALVA KEY TEA 6 Left						5	
RELP 9.6 Armoured Car ALVA GILL DILL 6 Left RELP 9.6 Armoured Car PETER HOP FOP 6 Right RELP 9.6 Armoured Car ALVA KEY TEA 6 Left						. 5	
RELP 9.6 Armoured Car PETER HOP FOP 6 Right RELP 9.6 Armoured Car ALVA KEY TEA 6 Left		•					
RELP 9.6 Armoured Car ALVA KEY TEA 6 Left							

CODING	BACKGROUND NOISE	PÉRSON	LEFT WORD	RIGHT WORD	ATTRIBUTE	REPLY
RELP 9.6	Helicopter	LYNNE	VAST	FAST	1	Left .
RELP 9.6	Helicopter	PETER	VAST	FAST	1	Left
RELF 9.6	Helicopter	PETER	VOLE	FOAL	1	Left
RELF 9.6	Helicorter	ALVA	VOLE	FOAL	1	Left
RELF 9.6	Helicorter	ALVA	VOLE	FOAL	1	Right
RELP 9.6	Helicopter	WOLF	VOLE	FOAL	1	Right
RELP 9.6	Helicopter	PETER	MOAN	BONE	2	Right
RELP 9.6	Helicopter	ALVA	MOSS	Boss	2	Risht
RELP 9.6	Helicopter	PETER	HOSS	BOSS	2	Right
RELP 9.6	Helicorter	LYNNE	NEWS	DUES	2	Left
RELP 9.6	Helicopter	PETER	FENCE	FENCE	3	Right
RELF 9.6	Helicopter	LYNNE	FENCE	FENCE	3	Right
RELF 9.6	Helicopter	WOLF	FENCE	PENCE	3	Right
RELP 9.6	Helicorter	ALVA	FENCE	PENCE	3	Neither
RELP 9.6	Helicorter	LYNNE	F00	P00	3	Right
RELF 9.6	Helicopter	WOLF	FOO	F00	3	Right
RELF 9.6	Helicorter	LYNNE	SHAD	CHAD	3	Right
RELP 9.6	Helicopter	WOLF	SHAD	CHAD	3	Risht
RELF 9.6	Helicorter	PETER	SHAD	CHAD	3	Right
RELP 9.6	Helicopter	ALVA	SHAD	CHAD	3	Right
RELF 9.6	Helicopter	WOLF	SHEET	CHEAT	3	Risht
RELF 9.6	Helicopter	LYNNE	SHEET	CHEAT	3	Right
RELP 9.6	Helicorter	LYNNE	SHEET	CHEAT	3	Left
RELF 9.6	Helicorter	PETER	SHEET	CHEAT	3	Left
RELF 9.6	Helicoster	WOLF	SHEET	CHEAT	3	Left
RELF 9.6	Helicopter	LYNNE	THONG	TONG	3	Left
RELF 9.6	Helicorter	ALVA	THOSE	DOZE	3	Left
RELF 9.6	Helicopter	PETER	THOSE	DOZE	3	Left
RELP 9.6 RELP 9.6	Helicopter	WOLF	THOSE	DOZE	3	Left
	Helicopter	LYNNE	THOUGH	DOUGH	3	Left
RELF 9.6	Helicopter	PETER WOLF	THOUGH	DOUGH	3	Right Left
RELP 9.6 RELP 9.6	Helicorter Helicorter	ALVA	THUUGH VEE	DOUGH BEE	3 3	Left
RELF 9.6	Helicopter	PETER	VEE	REE	3	Left
RELP 9.6	Helicopter	LYNNE	VEE	BEE '	3	Left
RELF 9.6	Helicopter	WOLF	VEE	BEE	3	Left
RELP 9.6	Helicopter	WOLF	VILL	BILL	. 3	Right
RELP 9.6	Helicopter	ALVA	VILL	BILL	3	Right
RELP 9.6	Helicopter	LYNNE	VON	BON	3	Right
RELF 9.6	Helicopter	ALVA	VON .	BON	3	Left
RELP 9.6	Helicopter	PETER	אסע	BON	3	Left
RELP 9.6	Helicorter	PETER	VON	BON	3	Risht
RELP 9.6	Helicopter	ALVA	VON	BON	. 3	Neither
RELP 9.6	Helicorter	WOLF	VON	BON	3	Left
RELP 9.6	Helicopter	PETER	VOX	BOX	3	Right
RELF 9.6	Helicorter	LYNNE	VOX	BOX	3	Right
RELP 9.6	Helicorter	ALVA	VOX	BOX	3	Right
RELP 9.6	Helicopter	ALVA	CHEEP	KEEP	4	Left
RELP 9.6	Helicorter	PETER	CHEEP	KEEP	4	Left
RELP 9.6	Helicorter	WOLF	CHEEP	KEEF	4	Left
RELP 9.6	Helicorter	ALVA	CHEW	COO	4	Neither

	BACKGROUND	,	LEFT	RIGHT		
CODING	NOISE	PERSON	WORD	WORD	ATTRIBUTE	REPLY
*		•	٠			
RELP 9.6	Helicorter	ALVA	JAB	GAB	4	Left
RELF 9.6	Helicorter	WOLF	JAB	GAB	4	Left
RELF 9.6	Helicorter	LYNNE	JILT	GILT	4	Right
RELP 9.6	Helicorter	PETER	JILT	GILT	4	Risht
RELP 9.6	Helicopter	LYNNE	SANK	THANK	4	Risht
RELP 9.6	Helicopter	ALVA	SAW	THAW	4	Right
RELP 9.6	Helicorter	PETER	SING	THING	4	Risht
RELF 9.6	Helicopter	LYNNE	SING	THING	4	Risht
RELP 9.6	Helicopter	WOLF	SING	THING	4	Risht
RELP 9.6	Helicopter	ALVA	SING	THING	4	Risht
RELP 9.6	Helicopter	LYNNE	ZEE	THEE	4	Risht
RELP 9.6	Helicorter	WOLF	Zt'E	THEE	4	Right
RELP 9.6	Helicorter	PETER	ZEE	THEE	4	Risht
RELP 9.6	Helicorter	ALVA	ZEE	THEE	4	Risht
RELF 9.6	Helicopter	LYNNE	BANK	DANK	5	Left
RELP 9.6	Helicorter	PUTER	BANK	DANK	5	Left
RELP 9.6	Helicopter.	LYNNE	BOWL	DOLE	5	Left
RELP 9.6	Helicorter	LYNNE	FAD	THAD	5	Left
RELP 9.6	Helicorter	LYNNE	FAD	THAD	5	Risht
RELP 9+6	Helicorter	WOLF	FAD	THAD	5	Risht
RELP 9.6	Helicopter	LYNNE	FIN	THIN	5	Left
RELP 9.6	Helicopter.	WOLF	FIN	THIN	5	Risht
RELP 9.6	Helicorter	PETER	FIN	THIN	5	Risht
RELF 9.6	Helicorter	WOLF	FIN	THIN	5	Left
RELP 9.6	Helicopter	PETER	FORE	THOR	5	Risht
RELF 9.6	Helicopter	LYNNE	FORE	THOR	, <u>5</u>	Risht
RELP 9.6	Helicorter	ALVA	FORE	THOR	5	Risht
RELP 9.6	Helicopter	LYNNE	FORE	THOR	5	Left
RELF 9.6	Helicorter	WOLF	FORE	THOR	5	Left
RELP 9.6	Helicopter	WOLF	HET	NET	5	Right
RELF 9+6	Helicorter	PETER	MET	NET	5	Left
RELF 9.6	Helicopter	WOLF	MOON	иоои	5 5	Right
RELP 9.6	Helicopter	LYNNE	POT	TOT TOT	- 5	Left Left
RELP 9.6	Helicopter	ALVA	WAD	ROD	5	Left
RELF 9.6 RELF 9.6	Helicorter Helicorter	ALVA LYNNE	GILL	DILL	6	Left
	Helicopter	ALVA	GILL	DILL	6	Right
RELP 9.6 RELF 9.6	Helicopter	LYNNE	HOP	FOP	6	Risht
RELP 9.6	Helicopter Helicopter	ALVA	KEY	TEA	6	Right
	Helicopter Helicopter	LYNNE	KEY	TEA	6	Right
RELP 9.6			KEY	TEA	6	Left
RELF 9.6	Helicorter	ALVA 💎	NE I	ICM	O	L=16

	BAC	CKGROUND	•		LEFT	RIGHT	•	
CODING		101SE	PERS	IN :	WORD		ATTRIBU	TE REPLY
-	•		· · · · · · · · · · · · · · · · · · ·	٠.,				
					• •			
PERT 4+8	None	•	WOLF		VAULT	FAULT	1	Left
PERT 4.8	None		WOLF		VAULT	FAULT	. 1	Risht
PERT 4.8	Копе		WOLF		MEAT	BEAT	2	Left
PERT 4.8	None		PETER		SHAD	CHAD	3	Right
PERT 4.8	Norie		ALVA	-	SHAD	CHAD	3	Risht
PERT 4.8	None		LYNNE		SHAW	CHAW	. 3	Right
PERT 4.8	None		LYNNE		THEN	DEN	. 3	Left
PERT 4.8	None		PETER		THEN	DEN	· , 3	Left
PERT 4.8	None	•	WOLF		THEN	DEN	3	Left
PERT 4.8	None		LYNNE	* , ,	VEE	BEE	3	Left
PERT 4.8	None		PETER		VEE	BEE	3	Left
PERT 4.8	None		PETER	1	JILT	GILT	4	Right
FERT 4.8	None		WOLF		SANK	THANK	4	Right
PERT 4.8	None	100	ALVA		SANK	THANK	4	Risht
PERT 4.8	None		LYNNE		SANK	THANK	4	Risht
PERT 4.8	Копе		PETER		SANK	THANK	· 4	Right
FERT 4.8	None	A	LYNNE		ZEE	THEE	4	Risht
PERT 4.8	None		ALVA		BONG	DONG	5.	Risht
PERT 4.8	None	*	LYNNE	٠.	FAD		5	
PERT 4.8	None	•			*	THAD	5 5	Risht
PERT 4.8	None		ALVA		FAD	THAD	ວ . 5	Left
PERT 4.8			LYNNE		FAD	THAD		Left
PERT 4.8	None		PETER ALVA		FIN	THIN	5 5	Right
PERT 4.8	None		LYNNE		FIN	THIN		Left
PERT 4.8	None		PETER		FIN FIN	THIN	5 5	Left
FERT 4.8	None		WOLF		FIN	THIN THIN	. J	Left Left
PERT 4.8	None		ALVA		FIN	THIN	5	Risht
FERT 4.8	None		WOLF	, ,	FIN	THIN	5 5	Risht
PERT 4.8	None		LYNNE	•	FIN	THIN	ა 5	Risht
PERT 4.8	None		LYNNE		MOON	NOON	5	Right
FERT 4.8	None		PETER		FENT	TENT	5	Risht
PERT 4.8	None		WOLF		PENT	TENT	5	Risht
FERT 4.8	None		PETER	;	GILL	DILL	6	Right
PERT 4.8	None		ALVA		KEY	TEA	. 6	Right
PERT 4.8	None	•	LYNNE		KEY	TEA	. 6	Risht
PERT 4.8	None		PETER		KEY	TEA	6	Right
PERT 9.6	None		LYNNE		DINT	TINT	1	Right
PERT 9.6	None		LYNNE		SHEET	CHEAT	* *	Left
PERT 9.6	None	•	ALVA	•	SHEET	CHEAT	3 3	Risht
PERT 9.6	None		LYNNE		SHEET	CHEAT	7	Risht
PERT 9.6	None		LYNNE		SHOES	CHOOSE	3 3	Risht
PERT 9.6	None	•	ALVA		SHOES	CHOOSE		Risht
PERT 9.6	None	•	ALVA		VILL	BILL	3	Risht
PERT 9.6	None		LYNNE		VOX	BOX	3 3 3	Risht
PERT 9.6	None		PETER		VOX	BOX	· . 3	Risht
PERT 9.6	None	•	ALVA		VOX	BOX	3	Risht
PERT 9.6	None		ALVA		CHEEP	KEEF	4	Risht
PERT 9.6	None		PETER		JEST	GUEST	· 4	Risht
PERT 9.6	None		LYNNE		SING	THING	. 4	Risht
PERT 9.6	None		WOLF		SING	THING	4	Risht
PERT 9.6	None		PETER		SING	THING	4	Right
1 EN 1 -7 10	KUITE	* .	· - i - 1/		W 2 17 W	THERW	. 77	MEDILO

CODING	BACKGROUND NOISE	PERSON	LEFT WORD	RIGHT Word	ATTRIBUTE	REPLY
			v +			
PERT 9.6	None	ALVA	SING	THING	· 4	Risht
PERT 9.6	None	LYNNE	BANK	DANK	5	Left
PERT 9.6	None	LYNNE	RID	DID	5	Left
FERT 9.6	None	PETER	BID	DID	5,	Left
PERT 9.6	None	WOLF	FORE	THOR	5	Risht
PERT 9.6	None	LYNNE	FORE	THOR	. 5	Left
FERT 9.6	None	LYNNE	FOUGHT	THOUGHT	5	Risht
FERT 9.6	None	WOLF	FOUGHT	THOUGHT	5	Right
PERT 9.6	None	ALVA	FOUGHT	THOUGHT	5	Right
PERT 9.6	None	PETER	FOUGHT	THOUGHT	5	Risht
PERT 9.6	Nonè	ALVA	POT	TOT	5	Risht 🦟