Title

Molecular Ion Photofragment Spectroscopy

Author

Bustamente, S.W.

Publication Date

1983-11-01

Thesis/dissertation

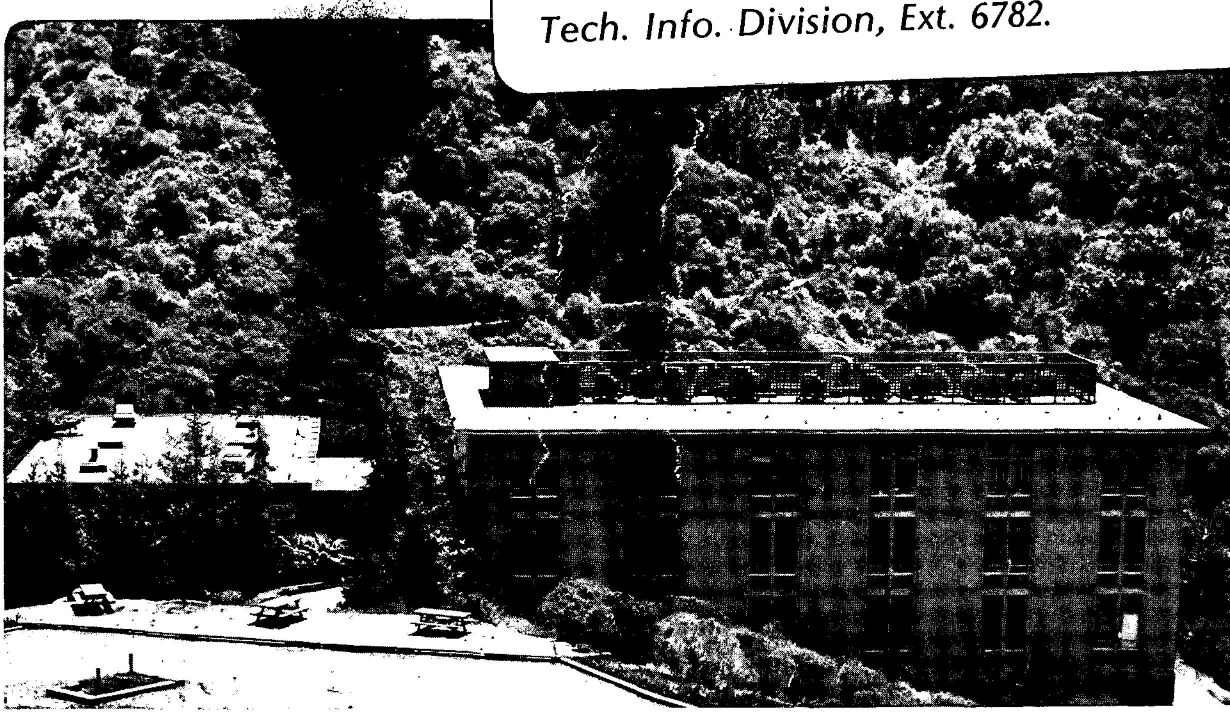
Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Materials & Molecular Research Division

MOLECULAR ION PHOTOFRAGMENT SPECTROSCOPY

S.W. Bustamente
(Ph.D. Thesis)

November 1983



Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

DISCLAIMER

MOLECULAR ION PHOTOFRAGMENT SPECTROSCOPY

Sanford Will Bustamente

Materials and Molecular Research Division
Department of Chemistry
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

This work was supported by the Director, Office of Energy Research, Office of Basic Energy Science, Chemical Sciences Division of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

MOLECULAR ION PHOTOFRAGMENT SPECTROSCOPY

Materials and Molecular Research Division
Department of Chemistry
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

Sanford Will Bustamente

ABSTRACT

A new molecular ion photofragment spectrometer is described which features a supersonic molecular beam ion source and a radio frequency octapole ion trap interaction region. This unique combination allows several techniques to be applied to the problem of detecting a photon absorption event of a molecular ion. In particular, it may be possible to obtain low resolution survey spectra of exotic molecular ions by using a direct vibrational predissociation process, or by using other more indirect detection methods. The use of the spectrometer is demonstrated by measuring the lifetime of the $O_2^+(^4\pi_u)$ metastable state which is found to consist of two main components: the $^4\pi_{5/2}$ and $^4\pi_{-1/2}$ spin components having a long lifetime (~129 msec.) and the $^4\pi_{3/2}$ and $^4\pi_{1/2}$ spin components having a short lifetime (~6 msec.).

ACKNOWLEDGEMENTS

I would like to thank Professor Yuan T. Lee for his tolerance over the years. Also, I would like to thank Lee group members, present and past, for any assistance they may have rendered. Of special mention are Lee Carlson, who worked with me during the early days of this project, Dieter Gerlich, who designed the radiofrequency ion traps used in the spectrometer, and Frances Houle, who took most of the supersonic ion source data. In addition, I would like to thank Cydney Ramos for punching computer cards for me and for helping me assemble the final version of the octapole ion trap. Lisa Yeh and Mitchio Okumura, two swell guys who have taken over the ion spectroscopy project, are wished the best of luck.

Many thanks are extended to the Technical Information Department of LBL. Especially Shirley Ashley and Diana Morris for providing prompt, professional service.

Much appreciation is expressed to Mitchio, Lisa, and Alec Wodtke for their kindness, friendship, and assistance during my last weekends here at Berkeley. Bird, Lizard, Beetleman, and the Dog will be well remembered. But far above all, the Rambus family will always stand as the most meaningful and the most remembered part of my graduate school years.

This work was supported by the Director, Office of Energy Research, Office of Basic Energy Science, Chemical Sciences Division of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

TABLE OF CONTENTS

I. INTRODUCTION: CONSEQUENCE SPECTROSCOPY

Spectroscopy in the traditional sense is the measurement of those wavelengths of the electromagnetic spectrum at which molecular or atomic species will absorb or emit radiation. From detailed measurements one can infer energy levels and from these one can obtain a distinct picture of the potential energy surfaces influencing the motions of the atoms within a molecular species and how these atoms are arranged geometrically in space. For nonreactive neutral species, spectroscopy can be just a matter of getting a sample of the species in question, shining light through the sample and making one or both of two basic measurements: one can measure the amount of light getting through the sample as a function of wavelength (the absorption spectrum) or one can measure the fluorescence from the sample. These measurements can be conducted with ease. By the astute application of lasers, with their well known properties, one may produce extremely high resolution spectra (the discussion here being limited to the gas phase spectroscopy of molecular and atomic species). But when speaking of the spectroscopy of molecular ions in this context, one finds that even at the most basic level, spectral information is quite limited. As for neutrals, knowing the spectroscopy of molecular ions would lead to the characterization of the potential energy surfaces of the ion and to its geometry. Knowing these facts about the ion can also lead to a better understanding of the neutral species the ion is derived from, as was the case in atomic spectroscopy.

The presence of gas phase ions in flames, plasmas, electrical discharges, planetary atmospheres and interstellar clouds as well as their more common existence in liquids and solids has made their spectroscopy a valued resource to many scientists. With the spectra of an ion known, its presence in outer space can be evaluated and its influence in the chemistry of interstellar space and the atmospheres of planets (like Earth) can be modelled more accurately. In the laboratory, knowledge of the potential energy surfaces of ions in experiments measuring ion-molecule reactions in a state-to-state determination is, of course, vital.

The molecular ion CH^+ was detected spectroscopically in space some forty years ago[(1)] and for years stood as the only known absorption spectrum of a molecular ion. In the last ten years, however, there has been a virtual revolution in the field of ion spectroscopy. New sensitive techniques[(2)] along with laser light sources have been applied with gratifying success. About sixty ions have been studied with techniques which yielded at least rotational resolution.[(2f)] But this fact can be placed somewhat in perspective when one learns that to this date the structure of the common ion CH_5^+ has not been spectroscopically determined and that the high resolution spectra of H_3O^+ was reported only very recently by Saykally et al.[(3)] With these facts in mind, we set out to build a spectrometer which would help fill in the void found in gas phase molecular ion spectroscopy by using a technique which would be more generally applicable than the methods used by existing instruments.

The difficulty.

Why is the gas phase spectroscopy of molecular ions so difficult? Well, to put the relative difficulties somewhat in perspective, let us consider taking the infrared spectrum of HCl gas at room temperature. Suppose one fills some sample cell with 1 torr of HCl gas. With very little trouble, then, one has a sample density of ~3×10^{-16} particles/cm^3 of essentially one species, with a well defined thermal population distribution to accomplish one's measurements with. One may generate an absorption spectrum with ease.

Now that one has the infrared spectrum of HCl neutral, it would be nice to have the spectrum of HCl^+ say, to be able to make all the comparisons one would like. Putting aside all the details of how the following condition is actually arrived at, we now have a container with electrons whizzing about creating HCl^+ from the neutral HCl gas originally in the container. But in addition to the two species above, one also has other neutral and ionic species—a veritable zoo of charged and uncharged species. So here at the start our spectroscopy is complicated by the fact that we have a mixture in our container, rather than just a single species as in the neutral case. This condition is exacerbated by the fact that ion-molecule reactions are among the fastest of chemical reactions. Thus, the HCl^+ created tend to be short lived because they recombine with electrons found in their environment, react with other species present, or find their way to a container surface and thereby annihilating themselves. This all leads to ion densities in typical experimental arrangements of 10^4 to 10^8

ions/cm^3. Note that the best case here is down by a factor of 10^8 from our 1 torr sample of HCl neutral. Of the HCl^+ that is created in this hodgepodge one has to put up with an additional obstacle: because their environment is generally hot, and because of the mechanism by which they are created, their population is partitioned among a large number of translational, rotational, and vibrational states, and perhaps even long lived electronic states. In short one does not have a 1 Torr sample of HCl^+ gas to work with. What one does have is a sample so dilute and dirty that spectroscopy becomes a very difficult, if not impossible, task.

The problems in trying to do spectroscopy on molecular ions in the gas phase have been mentioned above. What course should be taken if one wants to succeed in overcoming these problems? Many approaches have been taken, several very similar to ours. The following discussion will develop our approach to the problem of gas phase molecular ion spectroscopy, with particular interest in the infrared region, and concurrently will outline some detection techniques which have been tried by us and others or that could possibly be used with some success.

A pure sample.

The first problem to be taken care of is the "hodgepodge" which exists in most ion sources. That is, we want only one molecular ion species in our "sample bottle," i.e. our interaction region. Thus, no matter what our source of ions is, the source must be followed by some sort of mass filter which allows only the ion of interest through into the interaction region. We then have a nice pure sample of a known species to work with. How this situation is maintained will be addressed later.

Once the ions are in the interaction region they encounter a flux of photons from a light source and if the energy of these photons is equal to the spacing of some allowed transition in the ion, photon absorption by the ions will take place. Here comes the part where "technique" is written in large letters: How does one tell whether or not the molecular ions have absorbed any of the incident photons? Well, one searches for possible consequences of the adsorption process and then chooses the one best suited as a detection technique under the prevailing circumstances. At this point I wish to emphasize that the following is not intended to be, by any means, a comprehensive review of the techniques used in gas phase molecular ion spectroscopy. Rather, it is just a bare bones sampling of the techniques, to somewhat acquaint the reader with the field. Details may be gathered from the references found at the end of this chapter.

Absorption and emission.

At the beginning of this chapter, two standard spectroscopic techniques were mentioned: absorption and emission. Direct absorption measurements are generally not practical in molecular ion spectroscopy. For example, suppose one has 10,000 ions of species A^+ in an interaction region through which is passed an infrared light pulse containing 10^{16} photons (~1 mj/pulse at 3.5μ). And suppose one ion in 100 absorbs a photon, i.e. 100 photons were absorbed by the sample. One can easily see that there is no way that this absorption loss will be measured in a photon pulse containing 10^{16} photons originally. However, although uncommon, there are experimental arrangements which make the above technique possible. For example, in 1974, a direct microwave measurement of CO^+ through the plasma of a DC glow discharge was accomplished by Dixon and Woods[(4)] yielding very high resolution spectra. While the source provided a greater-than-average ion density (~10^{10} ion/cm^3), the spectra of the ions were quite weak. Therefore use of sensitive modulation methods and computer signal averaging are required to make the measurements possible. More recently the technique of velocity-modulation absorption spectroscopy developed by Saykally et al.[(5)] has been successful in observing the spectra of several ions with high resolution in the infrared.

Emission and fluorescence studies of molecular ions in the visible and near ultraviolet parts of the spectrum is feasible, but difficult because of low ion number density. Unfortunately, in the infrared region of the spectrum, lack of detectors with extremely high sensitivity, coupled with the long lifetimes of vibrationally excited states, make this method impractical.

Laser magnetic resonance[(2f)] has also been successfully applied to the study of molecular ions. However, since this technique, applies only to paramagnetic species, it is not as general as one would like.

Also the spectroscopy is complicated by the required presence of a magnetic field and by the fact the plasma source used in such experiments usually produces a mixture of ions which is vibrationally very hot. The technique does, however, yield nice high resolution data.

Ions are charged.

Most of the methods described above do not make any use of the fact that ions, by their defining characteristic, are charged species. The charged quality is of course the root of the difficulty in attempting their spectroscopy, but one may in turn use this apparently baneful characteristic to great advantage and thus overcome some of the shortcomings involved. One attribute of ions is that they can be detected with efficiencies close to unity. Thus if the "consequence" to be measured pertained to ion counting, rather than photon counting as above, one could possibly achieve the very high sensitivity necessary to combat the problem of low ion density.

Consider again the 10,000 ions in an interaction region in their ground state and call them species A^+. Suppose that in the ground state species A^+ will not react with the neutral background species B, which is also in the interaction region, but if A^+ absorbs an IR photon, it will then react with B to give products C^+ and D, according to the following reaction:

$A^+ + B$ -/-/→ ground state does not react

$A^+ + h\nu \longrightarrow A^{+*}$ absorption

$A^{+*} + B \longrightarrow C^+ + D$ exited state reacts.

Furthermore, suppose that in some way all the ions can be guided to a mass filter which transmits only species C^+ to an ion detector having near unit counting efficiency. Now before the light pulse passes through the interaction region, all the A^+ are in their ground state. This implies there will be no reaction, and thus no C^+ production. If the ions were to be sent on towards the detector, no counts would result. After the light pulse passes through, 100 A^+ ions are no longer in their ground state (using the same figures as before) and may now react with species B to form C^+ and D. Now sending the ions forward through the mass filter would result in approximately 100 counts being registered (100 C^+ product). These 100 counts are attributed to the absorption events. Here we have a case where by using ion counting, one can detect virtually every absorption event which took place in this very small sample of ions.

The types of reactions which may be used in schemes such as the one above are:

(1) charge exchange

$A^+ + B \longrightarrow A + B^+$

(2) collision induced dissociation

$A^+ + B \longrightarrow C^+ + D + B$

(3) proton transfer

$AH^+ + B \longrightarrow A + BH^+$

and (4) other reactions where mass A $\neq$ mass C

$$A^+ + B \longrightarrow C^+ + D \,.$$

The situation described above is of course the ideal case with no C^+ being produced when the light is off and all C^+ produced when the light is on being caused by the absorption events. There are numerous examples which approach this ideal. But more generally one may expect that the lower state will have a reactive cross section comparable to the upper state and of course the species A^+ in the interaction region may well be spread over several states initially. Thus, when the A^+ absorbs photons, only a change in the amount of C^+ produced will be seen; but this percentage change is still very much larger than the percentage of photons lost to the sample. Thus, in most cases the detection process is more difficult than the ideal flip-in experiment, but is still very doable.

Charge exchange was the detection scheme used by Wing et al.(6) who were the first to observe the infrared spectrum of a molecular ion, specifically HD^+. After the HD^+ had interacted with a laser beam, it was passed through a target gas (Ar, N_2, or H_2) and the surviving HD^+ exiting the target region was measured by a Faraday cup. Since the charge exchange process occurring in the target gas region is a function of the vibrational state of the HD^+, small changes in the beam current (~3 ppm) were observed when on resonance. Carrington et al.(7) also used the charge exchange technique to detect electronic absorptions in the molecular ion CO^+ and used the reactive cross section method in observing a single rotational line of H_2O^+.(2f)

Photodetachment.

Photodetachment is a technique applicable to negative ions.[2e] The technique hinges on the fact that absorption of a photon with energy above some limit will cause the ejection of an electron from the ion, thus creating a neutral species:

$$(5) \quad A^- + h\nu \longrightarrow A + e^-$$

Here one monitors the decrease of the A^- parent ion as a function of wavelength. This results in data with more noise than a flop-in experiment since one is often times observing small changes in the parent ion count rate caused by the photodetachment process. This would be like detecting the decrease in the parent ion A^+ rather than detecting the appearance of C^+ product in the ideal chemical reaction technique considered earlier.

Photofragmentation.

Dunbar says in his review[2b] "photodissociation spectroscopy relies on the truism that before an ion can dissociate, it must absorb a photon." Photofragmentation is another one of the possible consequences of an ion absorbing a photon, but it is a very important one in the field of molecular ion spectroscopy. Photofragmentation offers extremely high detection sensitivity because one detects the daughter ion fragment formed in the process

(6) $A^+ + h\nu \dashrightarrow B^+ + C$

just as one would detect the C^+ product ion in the chemical reaction scheme discussed before. But in photofragmentation all one needs in the interaction region is the species being studied. There is no need for a neutral reaction partner. So one always has an "ideal" case as long as the vacuum in the interaction region is good enough to prevent ion-molecule reactions, now unwanted, from occurring. The success or failure of the photofragmentation detection method rest mainly in the properties of the ion under study. Are the potential energy surfaces of the ion placed so that upon absorption of a photon of a particular energy, fragmentation will occur?

Electronic transitions leading to direct dissociation or predissociation have now been used extensively to obtain the photofragment spectra of gas phase molecular ions both large and small. When conditions are right, as for the O_2^+ ($^4\Sigma_g^- \longleftarrow {}^4\pi_u$) bands, photofragment spectroscopy with fast ion beam techniques yield very high resolution spectra which are unsurpassed by other techniques. The photofragment spectrometer at SRI International is one of the most sophisticated fast ion beam instruments,[8] and it can be operated in several modes (e.g., optical-optical double reasonance has been performed on O_2^+)[9] yielding a remarkable quantity of detailed information on the O_2^+ quartet potentials.

Vibrational Predissociation.

Vibrational predissociation has not been used as a detection scheme in molecular ion studies. This is simply because, unlike optical transitions which pump a considerable amount of energy into a molecule, vibrational transitions in infrared (400-4000 cm^{-1}) typically do not have enough energy to initiate fragmentation. But there are exceptions. Some ions produced by electron impact ionization are highly vibrationally excited. To the extent that some levels populated are near a dissociative channel of the ion, absorption of an IR photon can induce fragmentation of the ion.

Clusters, or van der Waals molecules, are chemical species which are held together by weak bonds; usually about one tenth the energy of an IR photon. In the case of neutral van der Waals complexes, vibrational predissociation has been observed and has been used as a spectroscopic tool to generate the spectra of molecular clusters:(10)

(7) $(C_6H_6)_2 + h\nu \longrightarrow 2C_6H_6$

(8) $(H_2O)_n + h\nu \longrightarrow (H_2O)_{n-1} + H_2O$

Lasers.

Just as one may take advantage of the great sensitivity afforded by ion detection schemes to help overcome the low number densities typically found in gas phase ion spectroscopy, so has the advent of the laser as a common spectroscopic light source been invaluable in the study of molecular ions. The laser, with its well known beam qualities, makes it possible to achieve high photon fluxes through the ion interaction region, thus helping to compensate for the low ion number densities one has to work with.

One should not take a cavalier attitude when it comes to the design or selection of the laser light source. There are numerous factors which must be kept in mind in this regard. One simple feature which should not be overlooked in evaluating the effectiveness of the light source is its linewidth relative to that of the molecular transition one is expecting to excite. Serious deviations in the expected signal-to-noise can result if this is not considered. For example, say the laser produces lots of IR photons ($\sim 10^{16}$ per pulse), but that the spectral width of the pulse is $\sim 1\ cm^{-1}$ while the transition of the molecular ion has a total width of $\sim 0.01\ cm^{-1}$. In this case, because of the spectral width mismatch, only ~ 1 percent of the laser energy will be resonant with the transition. It becomes obvious that one should tailor the laser's characteristics so as to optimize the desired resolution and signal of the particular measurement being made.

Design goals.

One desired condition in the spectroscopy of molecular ions is to have a pure sample. Hence, it is desirable to use a mass filter following the ion source. In addition several detection schemes have been outlined. Before making any further statements about how the spectroscopy is to be accomplished, one should state what the desired goals are for the spectrometer. These design goals should be within the limits set by experimental practicability.

What is it that we want to accomplish in this field where practically any piece of information is a new piece of information? Our primary objective was to have a generally applicable technique able to generate survey infrared spectra of gas phase molecular ions. High resolution spectroscopy was relegated to subsequent investigations. That is, once one has a low resolution spectra of an ion, then one can go back and use techniques that yield higher resolution, but that would have been impractical to use as a survey method.

Thus, our primary aim is to do survey (low resolution) infrared spectroscopy of ions in a general fashion. Clearly, it would be nice to have a sample of cold ions—ions in their ground electronic and vibrational state. This would make the spectrum generated easier to interpret and would give information concerning the lower levels of the ground electronic state, information generally difficult to obtain for ions. If one had enough resolution to resolve rotational structure, then moments of inertia could be calculated, yielding structural information. Of the techniques mentioned previously, which would give the ability to observe the infrared absorption process directly with the required high sensitivity so necessary for success Well, vibrational predissociation is a likely candidate, but what about having cold ions and the desired generality of the scheme

The molecular beam mentality.

Our research group is heavily into molecular beams. As a consequence, solutions to many problems are viewed with a certain molecular beam mentality, which many times is quite fruitful. In this case, the knowledge that with molecular beams one can easily make very cold clusters of various compositions, made the idea of vibrational predissociation of neutral and ionic van der Waals complexes a natural.

The idea goes like this. Suppose one wants to get the infrared spectrum of the molecular ion M^+. What one does is form the van der Waals species $M^+\cdots X$, where X = He or Ar. The binding energy of this complex is bound to be quite low, undoubtedly much lower than the energy of an infrared photon (400–4,000 cm^{-1}).[12] Once we have formed this species, things are a snap. The selected complex is deposited into the interacton region, which is a region of ultra high vacuum thus minimizing collisional dissociation of the weakly bond complex by neutrals. While in the interaction region, the ions are bathed with a healthy pulse of IR photons. If the wavelength is correct, the $M^+\cdots X$ complex will absorb a photon. The energy absorbed will rattle around the complex until enough is transferred to the weak bond to cause its rupture,

$$(9) \qquad M^+ \cdots X + h\nu \longrightarrow M^+ + X \; (h\nu = IR).$$

So here we have a method which promises to be general (so long as the complex can be made) and which uses vibrational predissociation to directy view the absorption process. Very high sensitivity would be achieved by mass selective ion counting. A spectrum would be generated by counting the M^+ product as a function of the infrared wavelength, which is scanned.

Drawbacks.

The scheme looks good. Much could be learned about the $M^+\cdots X$ complex itself and about the M^+ ion, if all worked out well. But, as hinted at above, barriers exist which might prevent our goal from being achieved. For one thing no proven source for generating lots of the $M^+\cdots X$ complex exist. The end result may be that we will have very few complexes to work with––much less than typically found in ion spectroscopy. But it should be easier to generate ion clusters of the type $(M)_n^+$ or the protonated $(M)_nH^+$ species with which one could perform vibrational predissociation experiments if the cluster size is large enough:

$$(10) \quad (M)_n^+ + h\nu \longrightarrow (M)_{n-1}^+ .$$

The number of ion clusters may be so low that the number of ions illuminated by a light beam with 1 cm^2 cross section, crossing at right angles to the ion beam, would be quite a bit less than one. To help mitigate the problem of having extremely few ions, a coaxial light source, slow-ion-beam arrangement was decided upon. Thus, in an ultra high vacuum environment, slowly moving ions travel through a long interaction region through which a light pulse propogates. This configuration provides good spatial overlap of the ions with the light source and increases the number of ions in the interaction region per light pulse.

In addition to the need for a good ion beam-light beam overlap, one also needs all ions (targets and fragments) to be guided to the detector. A radio frequency (RF) ion guide can accomplish this task quite efficiently. And as an added bonus, by placing ion mirrors at either end of the ion guide, one can trap the ions within the guide.

Having an RF ion trap for an interaction region proves beneficial in other respects. With the trap one need not worry about the time it takes for the vibrational predissociation event to take place. One just keeps the excited complex trapped until the bond rupture takes place, then feeds the products to the detector. Experience gained from the vibrational predissociation of neutrals indicates that these times are short, and thus are of no practical concern.

If one is working with ions which are vibrationally or electronically hot, one may use the trap as a cooling tank. Not only could one possibly use this method to get the spectrum of the relaxed ion, but one could also measure the radiative lifetimes of various excited states. One excellent use of the ion trap is to guide reactant ions and their products through a background gas and finally to the detector in schemes which fall under the "chemical reaction" category discussed earlier. A machine in our group uses this "guided ion technique" to measure total cross sections in ion-molecule reactions.[11] Having an RF ion trap for the interaction region is very convenient.

Let us get back to the draw backs of using $M^{+}\cdots X$ complexes in our spectroscopy scheme. The spectrum we are after is that of the M^{+} ion, but what we are generating is the spectrum of the complex. Now this in itself is of interest, since it gives us structural information of the complex and possibly the time scale for the predissociation. However, the intent is to study the of M^{+} ion. The situation here is not too bad. Although the rotational structure of the transitions would be expected to be characteristic of the complex as a whole, the vibrational structure should be essentially that of M^{+} alone. Furthermore, as stated before, once one has found the wavelength regions where the M^{+} ion absorbs by using the survey method, perhaps one can return to the problem with a high resolution technique, and acquire the rotational structure of the uncomplexed ion.

Roundup.

Here the essential qualities of a spectrometer capable of acquiring the low resolution infrared spectrum of a molecular ion M^{+} by directly observing the vibrational predissociation products of the related van der Waals ion $M^{+}\cdots X$ (X = He, Ar), are stated. The basic requirements are:

1) An ion source region with high pumping speed and high throughput capability able to support the gas loads created by the

supersonic molecular beam source used to form the cold $M^{+}\cdots X$ type van der Waals molecular ions.

2) A mass filter following the ion source so that one may select the species placed into the interaction region. The filter should provide good ion-beam quality and luminosity.

3) An ultra high vacuum interaction region should provide a clear through shot for the light source and should support slowly moving ions over a long straight path so that good spatial overlap between the ion beam and light beam can be achieved. The ion optics here must be able to guide all ions, parents and fragments, to the final sections of the spectrometer.

4) After the interaction region comes a second mass filter which allows one to select which mass the ion detector views. This gives the high sensitivity, low background detection necessary for success.

5) One needs an ion detector with low inherent dark count and reasonably fast response time.

6) And finally, a necessary component which is physically separate from the above, but is an integral part of the spectrometer system, is the light source. Without the advantages rendered by a good infrared laser light source, the project would be doomed. Lots of photons are needed to help overcome the lack of ion density.

A functional block diagram of the spectrometer is shown in Fig. 1. The success of the instrument as an infrared molecular ion spectrometer of general utility depends on the successful operation of the following three components:

1) the ion source producing $M^{+}\cdots X$ complexes.

2) the ion trap used to achieve adequate ion-laser spatial overlap and to guide all ions to the detection stage.

3) the infrared laser light source.

If any of the three fails, then the spectrometer becomes just a run-of-the-mill tandem mass spectrometer capable of low resolution photofragment spectroscopy.

REFERENCES

1. W. S. Adams, Astrophys. J. 93, 11 (1941).

2. (a) G. Herzberg, Rev. Chem. Soc. 25, 201 (1971).

(b) R. Dunbar, "Ion Photodissociation," in Kinetics of Ion-Molecule Reactions, edited by P. Ausloos, Plenum Press, New York (1979).

(c) J. Moseley and J. Durup, "Fast Ion Beam Photofragment Spectroscopy," Ann. Rev. Phys. Chem. 32, 53 (1981).

(d) L. Andrews, "Spectroscopy of Molecular Ions in Noble Gas Matrices," Ann. Rev. Phys. Chem. 30, 79 (1979).

(e) R. R. Corderman and W. C. Lineberger, "Negative Ion Spectroscopy," Ann. Rev. Phys. Chem. 30, 347 (1979).

(f) R. J. Saykally and R. C. Woods, "High Resolution Spectroscopy of Molecular Ions," Ann. Rev. Phys. Chem. 32, 403 (1981).

(g) Molecular Ions: Geometric and Electronic Structures, edited by G. J. Berkowitz and K. O. Groeneveld, Proc. NATO Advanced Study Institute, Sep. 1980, Plenum, New York (1983).

3. M. H. Begemann, C. S. Gudeman, J. Pfaff, and R. J. Saykally, Phys. Rev. Lett. 51, 554 (1983).

4. T. A. Dixon and R. C. Woods, Phys. Rev. Lett., 34 , 61 (1975).

5. C. S. Gudeman, M. H. Begemann, J. Pfaff, and R. J. Saykally, J. Chem. Phys. 78, 5837 (1983).

6. W. H. Wing, G. A. Ruff, W. E. Lamb, and J. J. Spezeski, Phys. Rev. Lett. 36, 1488 (1976).

7. A. Carrington and P. J. Sarre, Molec. Phys. 33, 1495 (1977).

8. B. A. Huber, T. M. Miller, P. C. Cosby, H. D. Zeman, R. L. Leon, J. T. Moseley and J R. Peterson, Rev. Sci. Instrum. 48, 1306 (1977).

9. P. C. Cosby and H. Helm, J. Chem. Phys., 76, 4720 (1982).

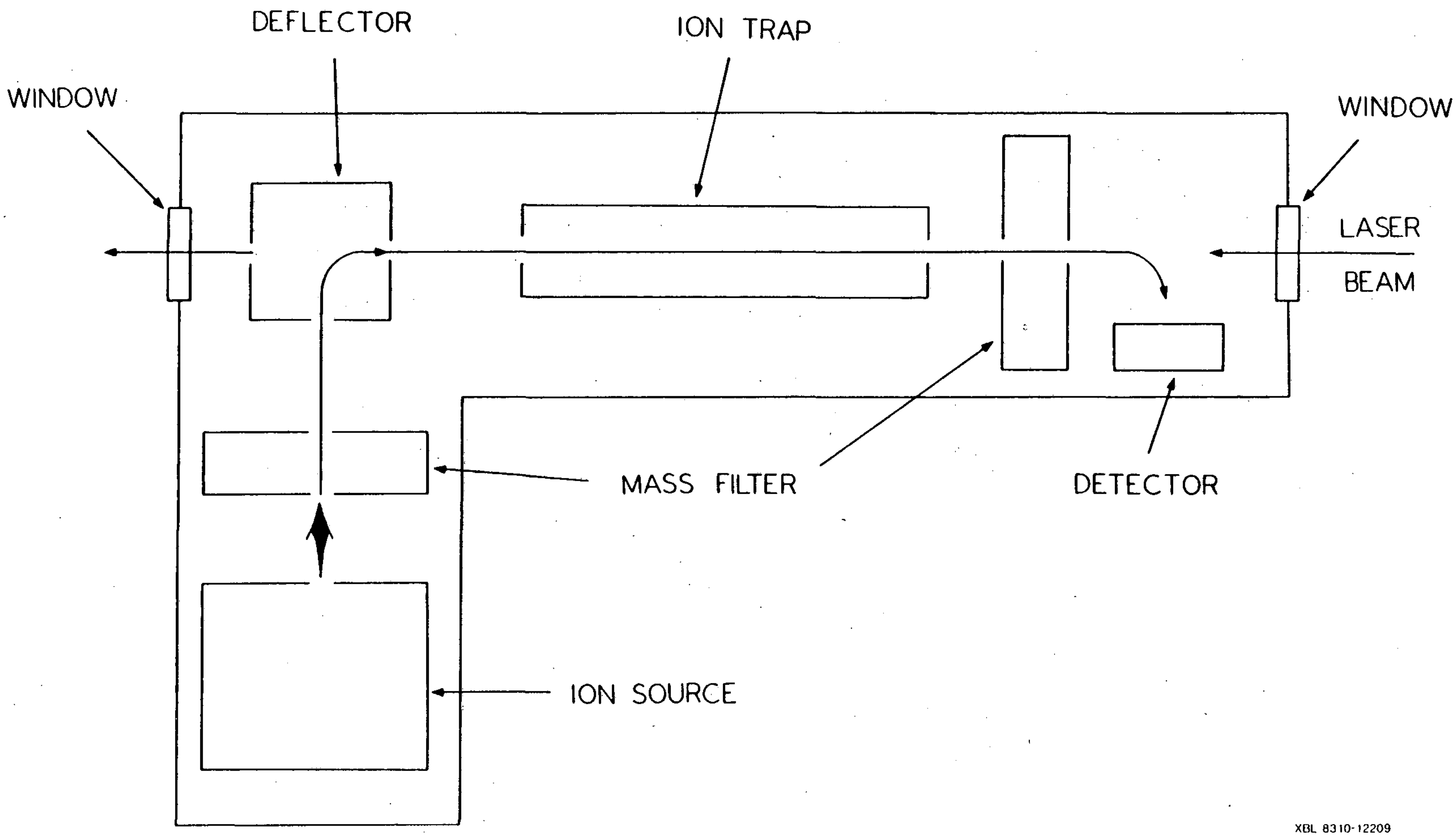
10. M. F. Vernon, "Molecular Beam Scattering," Chap. 2, Ph.D. Thesis, University of California, Berkeley, CA (1983).

11. S. L. Anderson, "The Effect of Vibrational Excitation on the Dynamics of Ion-Molecule Reactions," Ph.D. Thesis, University of California, Berkeley, CA (1981).

12. D.L. Cooper and S. Wilson, Molec. Phys. 44, 161 (1981).

FIGURE CAPTION

Fig. 1. A functional block diagram of the molecular ion photofragment spectrometer using a tandem mass filter configuration and an ion guide interaction region.



XBL 8310-12209

Figure 1

II. THE SPECTROMETER

In Chapter I the general description of the spectrometer we wished to construct was laid out. It was to be a low resolution infrared photofragment spectrometer of general applicability. This chapter will take a look at what was actually built in the attempt to reach our design goals.

A. The Ion Source

On first seeing our spectrometer, one is apt to ask "why in the heck is the darn thing so big?" The answer is two-fold. First, some parts are large because they are just remakes of existing beam designs found in our group or are surplus items, such as the large sector magnet used as the first mass filter. Second, and more importantly, the source region is large because it has to accommodate not only conventional ion sources, such as electron impact ionizers, duoplasmatrons, etc., but also the high pressure expansion beam source. The molecular beam source operates at about one atmosphere through a 100 μm diameter orifice creating a large demand on the system's pumping speed and throughput capabilities. The spectrometer must have sufficient differential pumping to maintain the interaction region at about 10^{-12} atmospheres when the nozzle is operating at 1 atmosphere.

The ion source is mounted in Region I of the spectrometer (see Fig. 1). This region is pumped by a high speed 10 inch diffusion pump (Varian VHS-10) charged with Convalex-10 pump oil, as are the other two diffusion pumps used on the spectrometer. This pump is trapped by a liquid nitrogen cooled chevron baffle in order to minimize the back-streaming of oil into the ion source region. Metal surfaces must be kept free of oil, so that surface charge build up will not be a problem. If the throughput from this region is large, the diffusion pump is backed by a 150 $l sec^{-1}$ Roots blower. This combination can maintain the high pumping speeds and throughputs which are necessary when the supersonic beam source is employed.

Region II of the source (see Fig. 1) contains the lens system (Fig. 2) used to accelerate and focus the ions issuing from the ion source. Deflection plates steer the ion beam either to a quadrupole mass spectrometer for source optimization or to the sector magnet which selects the ion mass sent on into the interaction region. Region II is pumped by a liquid nitrogen trapped, six inch diffusion pump (Varian M-6). This region should have pressures below 10^{-5} torr when the supersonic source is being used.

The supersonic ion beam source to be described is basically an atmospheric pressure ionizer(1) used in mass spectrometry, except that here the intent is to form clusters, or van der Waals complexes of some particular species M^+. The working idea is to form a weak plasma at pressures of about one atmosphere by using the β-rays of some radioactive element, usually ^{63}Ni or ^{3}H. Because high temperature bakeouts are often needed to clean out the nozzle system, ^{63}Ni is perhaps the better choice, but we bought ^{3}H foils opting for the higher activity available from tritiated foils. As Fig. 3

shows, the radioactive foil, curled to form a cylinder, is placed in a small tube so that it is near the nozzle orifice. The gas mixture to be ionized is fed in through the inlet tube until it fills the nozzle cavities. If the vent line of the nozzle is closed, the only flow of gas is through the orifice placed near the foil's active region. If the vent line is opened, one may have a controlled flow of gas above that which flows out the orifice, so that one may dictate the amount of time the volume of gas remains in the region of the radioactive foil. This extra flow can mitigate the effects of unwanted side reactions.

The light plasma, formed within the nozzle by the β-particles acting on the gaseous species, gives rise to a complicated array of ion chemistry. One wants the β-particles to interact with some carrier gas directly, to form ions and excited states, and then subsequenty to have these ions and/or metastable species interact more softly with the species one really wants ionized, but not torn to bits. For example, suppose one wants to ionize species M by using the above scheme. One would whip up a mixture of say 99 percent argon and 1 percent species M, and then send it down the tube to meet its foil. Within the foil's volume, the β-particles would, by chance, interact with the argon atoms directly, forming Ar^+ and Ar^* (metastable state). These two species would then interact with M via collisions within the light plasma:

(1) $\beta + Ar \longrightarrow Ar^+ + 2e$

(2) $\beta + Ar \longrightarrow Ar^* + e$

(3) $Ar^{+} + M \dashrightarrow M^{+} + Ar$

(4) $Ar^{*} + M \dashrightarrow M^{+} + Ar + e$

Process (3) is simply a charge exchange reaction and process (4) is termed Penning ionization. Ideally these would be the only processes occurring within the plasma. The light plasma consisting of charged species M^{+}, e, and Ar^{+} would expand into the vacuum along with the neutrals Ar and M to form the supersonic molecular beam. One would then adjust parameters such as temperature and stagnation pressure to optmize the formation of the desired van der Waals ion, which might be $M^{+}\cdots Ar$ in this case. If enough of the complex is made, then the fun part begins. What color light makes it fall apart again

But things are somewhat more compicated than outlined above. The complication arises because ion-molecule reactions, as stated earlier, are very fast, and so the charge can be passed from one species to another. The positively charged species found in the plasma, referred to as the terminal ions, are those which are the most stable. Water seems to be present in any vacuum system, so given sufficient time, one usually ends up with the terminal ions being $H_3O\cdot(H_2O)_n^+$ ions, where n range from zero to surprisingly large numbers. This distribution of water cluster ions depends on the temperature, the water concentration, other species present, and the time given to react.[1a] Thus, it becomes more clear why it is desirable to be able to bake out the system, control the temperature, and by using the extra flow via the vent tube, to control the residence time of the plasma in the foil region. The ion spectrum one gets in a particular case is difficult to predict because of the wide variety of ionization routes available. These can be characterized as follows:

(5)	$A^* + B \longrightarrow (AB)^+ + e$	associative ionization
(6)	$A^* + B \longrightarrow B^+ + A + e$	Penning ionization
(7)	$A^\pm + B \longrightarrow A + B^\pm$	charge transfer
(8)	$A + B + e \longrightarrow A^- + B$	electron attachment
(8a)	$AB + e \longrightarrow A^- + B$	
(9)	$AH^+ + B \longrightarrow A + BH^+$	proton transfer
(10)	$A^- + BH \longrightarrow AH + B^-$	deprotonation
(11)	$A^+ + BC \longrightarrow A + B^+ + C$	dissociative charge transfer
(12)	$A + B^\pm + C \longrightarrow AB^\pm + C$	clustering
(13)	$MCl + O_2^- \longrightarrow MO^- + OCl$	substitution

Let's consider again the action of this type of source on a gas mixture composed of argon and the species M we wish to ionize. In particular say the mixture is 95 percent Ar and 5 percent O_2 (used because it was on hand). Then a realistic scheme representing the primary positive ion chemistry in the foil region of the nozzle would be the following:

(14)	$\beta + Ar \longrightarrow Ar^+ + 2e$
(15)	$\beta + Ar \longrightarrow Ar^* + e$
(16)	$Ar^+ + 2Ar \longrightarrow Ar_2^+ + Ar$

(17) $Ar^{+} + O_2 \longrightarrow O_2^{+} + Ar$

(18) $Ar^{+} + H_2O \longrightarrow H_2O^{+} + Ar$

(19) $Ar^{+} + H_2 \longrightarrow ArH^{+} + OH$

(20) $Ar^{*} + H_2O \longrightarrow H_2O^{+} + Ar + e$

(21) $H_2O^{+} + O_2 \longrightarrow O_2^{+} + H_2O$

(22) $H_2O^{+} + H_2O \longrightarrow H_3O^{+} + OH$

(23) $H_3O^{+} + H_2O + Ar \longrightarrow H_3O\ H_2O^{+} + Ar$

Reactions similar to (23) would lead to clusters of the type $H_3O\cdot(H_2O)_n^{+}$ with the distribution of n being highly dependent—as noted before—on the temperature, the water concentration, other impurity species present, and the time given to react. Of course clustering of the various species with each other will be taking place also, not to mention the added complication of other impurity species found in the nozzle.

The prototype nozzle shown in Fig. 3 was designed to discover the influence of gas flow, nozzle aperture size, nozzle-skimmer acceleration gradients, nozzle-skimmer distance, and gas mixture on the ion spectrum produced by the nozzle. In addition we hoped to learn whether the pumping capabilities met the demands of the source. The tests which were carried out using this nozzle were very preliminary in nature—just to get an inkling of whether the idea would work and how one might set up to study the nozzle more systematically. Unfortunately, in the following years, the time was never found to return to characterize the nozzle well. In short, the ion mass spectra presented here, which display the effects of varying some of the various parameters mentioned above, are not of good quality. Nevertheless the spectra are presented to show that one does have some control over the cluster types and the cluster distribution produced by this ion source, giving hope that one may be able to optimize the production of a particular ion cluster.

The first ion mass spectra to be presented show how an extra flow of gas (that is in addition to the flow of gas through the nozzle orifice) can change the product ion distributions. Figure 4 shows how an additional flow can eliminate the production of ions derived from an impurity. In this case the impurity is ethanol which was used as a cleaning solvent. The flow is enough to remove the ethanol clusters from the spectrum, but it changes the water cluster distribution minimally. In Fig. 5 a more drastic flow is used and in addition to suppressing the ethanol clusters, the water cluster distribution is radically altered. Sometimes extra flow through the foil region does not bring a benefit. For example, if one were trying to produce the van der Waals ion $Ar{\cdot}H_3O^+$, then having less residence time in the foil region would seem to be a definite disadvantage. Figure 6 shows that extra flow results in the disappearance of the $Ar{\cdot}H_3O(H_2O)_n^+$ type clusters from the mass spectrum.

One would like the ions which leave the nozzle to go through the skimmer orifice and into Region II of the source, but of course only a fraction of the total ion flux makes it through. To increase the number of ions which pass through this orifice, an electric field is established between the nozzle and the skimmer to accelerate the ions towards the skimmer. But there are consequences to this procedure. Figure 7 shows that there is a decrease in average cluster size with increasing accelerating electric field. This shift in cluster size may be understood by recognizing that the acceleration takes place in a region of high pressure. Thus, the clusters may be fragmented by colliding with neutrals which are present in the beam. Another possibility is that ions are pulled away from the expansion zone where, via three body collisions, the clusters are becoming larger. But one must also add focusing effects to the above. It may be that there is a shift in ion mass transmission through the skimmer orifice as the electric field gradient is changed.

Although the ion mass spectra seen so far were taken as a mixture of Ar/O_2 was fed into the source, one does not see any O_2^+ in Figs. 4, 5, 6 or 7. Figure 8 is included to show that with the addition of heat (nozzle tip 100°C) and extra flow one can produce the O_2^+ ion. Thus, one may conclude that with a source of this type, one does have a handle on what is produced by it. Hopefully, there is enough control to generate the prerequisite quantity of the species needed for spectroscopy.

A good case.

Suppose that the species that one is trying to create is $Ar{\cdot}H_3O^+$. Figure 9 shows the ion spectrum generated from the Ar/O_2 gas mixture discussed before, but with various parameters adjusted to optimize the generation of the $Ar \cdot H_3O^+$ ion. If the spectrum shows all the masses being produced under the conditions, then the $Ar \cdot H_3O^+$ peak comprises about 25 percent of the total ion flux. That is about as good as one might expect to achieve for the production of a van der Waals ion. If the total ion flux from the nozzle is 5×10^8 ion sec^{-1} (~50 mCi tritiated foil) then one would have $\sim 1.2 \times 10^8$ ion sec^{-1} of $Ar \cdot H_3O^+$. Assuming an efficiency of 10^{-5} (that is one ion in 10^5 survives the skimmer, Region II collisions with neutrals, and all the ion optics), one gets $\sim 1.2 \times 10^3$ ion sec^{-1} arriving at the interaction region. It takes about 1 msec for the ions to traverse the interaction region, so at any given time one has about one $Ar \cdot H_3O^+$ to work with. This is less than originally expected and is unacceptable.

Ten to one hundred times this many ions are needed in the interaction region to have a workable experiment. What went wrong The original error was in over estimating the number of β-particles that could be delivered per second into the ionization volume by the tritiated foil. Tritiated foils were chosen because they are available with higher activities than are the ^{63}Ni foils. The error made was that the quoted activities of the various foils were converted directly to β-particles sec^{-1} without considering the correction necessary for saturation due to self-absorption. This cost about a factor of 20 in the ion flux estimates and may spell doom for the experiment as presently configured. Some improvement may be realized by using ^{63}Ni foils which are bakeable and yield β particles which are more energetic than those produced by ^{3}H.

B. The Sector Magnet: momentum analysis

The ions which are extracted from the source are accelerated to a given potential, focused, and sent on to Region III by the action of the ion optics in Region II. Region III, shown schematically in Fig. 10, is maintained at $2x10^{-8}$ torr by a liquid nitrogen baffled, 4 inch diffusion pump (Varian VHS 4). Ions of differing masses travel together until encountering the magnetic field of the sector magnet. The magnetic field is arranged to be perpendicular to the ion trajectories. Each ion type, all having the same average kinetic energy, is forced to travel in a circular path by the influence of the magnetic field. The radius of the circular path followed by a particular ion is proportional to the momentum of the ion and thus to the square root of the ion's mass. Therefore, if only a limited range of curvatures are allowed to be transmitted (by using physical beam stops), one has a mass separator. In this case, since the energy and transmitted radius is fixed, one selects which ion mass gets passage by setting the value of the magnetic field.

The magnet used is a 60° sector. It is situated so that the ions enter the sector region with trajectories perpendicular to the pole piece and exit with trajectories perpendicular to the exit pole boundary if their radius of curvature is 8 in. while in the magnetic field. Ions traveling through the magnetic sector in this fashion are focused in the radial direction but experience no focusing in the direction parallel to the magnetic field. This poses a serious problem, because if the ion beam approaches the mass filter with an expanding circular cross section, lack of z direction focusing in the sector would result in a substantial loss of ions—something which cannot be endured in the van der Waals ion experiment.

The sector magnet was chosen as a mass separator in part for its excellent beam quality, but to work well it must also exhibit adequate resolution and high transmission. Usually, the resolution is obtained by proper use of slits and lenses (other factors being constant). However, this would only add to the losses mentioned above unless the lens corrects for the lack of z focusing. One wants a method which will satisfy the resolution requirements and at the same time, increase, rather than decrease, ion transmission efficiencies.

C. The Quadrupole Lens Pair

The focusing properties of an electrostatic quadrupole lens pair (QLP) has these desired qualities. The QLP consists of a double set of four rods each creating an electrostatic quadrupole field. The orientation of one quadrupole field is rotated 90° from that of the other. The QLP will transform a rectangular ion beam entering it into one which is parallel in one plane and is converging in the other, thus focusing the ions onto a line (see Fig. 11). This result is excellent because focusing to a line will enhance the mass separator's resolution, and having a wedge-shaped beam means that no z direction focusing is required; thus, losses in the sector magnet can be reduced. The QLP's characteristics have been detailed by several authors.[(2)] It is enough to say here that the characteristics of the lens are ideally suited for use with magnetic mass separators.[3] Our mass spectrometer uses no mechanical slits within Region III. (The only apertures are those used to establish the differential pumping regions.) Instead, one QLP is used at the entrance to transform the beam into a wedge and another used at the exit is to transform it back to a collimated beam. One added feature of the QLP is that it can be operated with pole voltages offset symmetrically to act as beam deflectors without loss of focusing. This is quite convenient.

Figure 12 shows the improvement afforded by the QLP's action. The resolution goes from $M/\Delta M = 85$ with the QLP's off to $M/\Delta M = 150$ with them on. The ion count rate is improved by 16 fold with the use of the QLPs. These mass spectra were taken using an ion energy of only 350 eV. The resolution at this energy is rather poor, but is typically good enough for our purposes for $M/e < 100$. The resolution can be greatly improved by increasing the ion energy should the need arise. The present condition is tolerated because (1) it is quite adequate for experiments being done and (2) raising the ion energy would require a slight modification of the quadrupole mass filter used down stream of the interaction region.

When leaving Region III the ions pass through a small tube which serves to limit conductance between Region III, which is held at $\sim 2\times10^{-8}$ torr, and Region IV, which is usually $\sim 2\times10^{-9}$ torr or less under operating conditions. Region IV (see Fig. 13) is pumped by a 220 $l\,sec^{-1}$ ion pump. It contains the ion-laser interaction region and the ion photofragment detection system. Both Regions III and IV are designed to be bakeable.

The discussion in Chapter I pointed out that one of the design objectives of the spectrometer was to have a "clear through" shot for the light beam. This would make alignment less of a problem, especially when using infrared light. Because of our configuration, reaching this objective necessitated bending the ion beam. Electrostatic deflection was selected to accomplish this job since one voltage setting would track all masses transmitted by the sector magnet. There are several types of electrostatic deflectors in common use, but most of them will not allow the ion beam to be merged with a light beam without having holes in one or more of the electrodes. The presence of these holes disturbs the electric field within the deflector and thus its operating characteristics. If a wire mesh is used over the hole to preserve the electric field, then the light beam may produce unwanted side affects when it strikes the mesh.

D. The Quadrupole Field Deflector

There is, however, a marvelous electrostatic deflector which is simple to build, has excellent deflection properties and provides an unobstructed merging path for the light and ion beams. The device is a two dimensional electrostatic quadrupole field deflector (QFD) which is detailed by Zeman.[4] If designed properly the QFD gives excellent beam quality. In the present case its full abilities in this regard were not needed and so a simplified design was used (see

Figs. 13 and 14). Figure 15 shows the equipotentials calculated for the simple QFD used here. The grounded elements are used to terminate the fringing field extending from the electrodes. The hyperbolic electrodes are approximated by circular ones.

The figures which are presented (Figs. 16-21) show ion trajectories through a QFD and serve to display some of its characteristics. The trajectories are those for an ideal quadrupole field with a sharp cut off fringe field. Figure 16 shows the lowest energy solution for which ions execute a 90° deflection. This solution is also the one with the least aberrations. The energy of the ions in this case is assigned the relative energy 1.00. Figure 17 shows the action of the QFD on a narrow beam of ions which is not monoenergetic and Fig. 18 shows a monoenergetic wide beam entering the deflector. Figure 19 has a converging monoenergetic beam entering the QFD. Figures 20 and 21 show the trajectories of two higher order solutions (with relative energy 1.925 and 1.858, respectively), which have aberrations making them unsuitable for this application; besides, the ions would strike the pole piece unless it were modified to provide the necessary clearance.

Slow them down.

Following the QFD the ions enter a constant gradient decelerator(5) which has an einzel lens as a last stage. The ions are slowed down to about 5 eV and then focused into the octapole RF trap with the einzel lens.

E. The RF Octapole Trap

The octapole region is one of central importance to the spectrometer. It must contain all ions, parents and offspring, and deliver them to the final mass filter. In selecting an ion guide for this purpose, one must be aware of its functional limitations. For example, the most common ion guide used is the RF quadrupole. But this ion guide has strict initial conditions which must be met by the ions for them to be contained by the guide.(6) In our spectrometer an RF ion guide developed by Teloy and Gerlich(10) is used because it is felt that their octapole imposes initial condition requirements which are less severe than those of the quadrupole guide. Thus, more ions are confined in the octapole guide. The octapole used consists of eight parallel, 20 inch long, 1/8 inch diameter, molybdenum rods, equally spaced about a circle of 1/2 inch diameter (see Fig. 22). Alternate poles are connected to opposite ends of a coil so as to form a resonant circuit; the parallel rods forming the "capacitor" of the circuit. This parallel LC circuit is excited inductively by a single loop of wire powered by an RF generator tuned to the resonant frequency of the tank. Molybdenum is the material used for the rods and the entrance and exit lenses of the octapole because of its ability to maintain uniform surface potentials.(7) This is a must in applications using low energy charged particles. But if one must have an extensive lens system, the cost of using molybdenum can be sobering. Pollard(8) describes the fabrication of a less expensive, but completely satisfactory, lens system made of aluminum coated with molybdenum. For large systems this may be the preferred route to take. A detailed drawing showing how the eight rods are held in place at the ends of the guide is shown in Fig. 23. Only the topmost and bottommost rods are shown for clarity. The ends of the rods are rounded to minimize the tendency for arcing under the influence of the RF field. The machinable ceramic annulus, used for electrical isolation, is kept well away from the ion beam by use of stainless steel spacers. The entrance (exit) lens element controls the flow of ions into (out of) the octapole ion guide.

Trapped trajectories.

Based on Kapitza's analysis of classical mechanics in a fast oscillating field,[9] the average motion of a slow moving charged particle in an RF n-pole field is described by:

$$m \frac{d^2\vec{r}}{dt^2} = -\vec{\nabla}V_{eff}(r) \tag{24}$$

where

$$V_{eff}(r) = \frac{q^2E^2(r)}{4m\omega^2} + q\phi_s(r) \tag{25}$$

and q is the charge of the particle, E is the amplitude of the RF, ω is the angular frequency of the RF, r is the smoothed distance from the axis of the trap to the ion, and ϕ_s is the electrostatic potential (see Fig. 22). The potential for an ideal octapole oscillatory field is given by

$$\phi(t) = \frac{V_o}{r_o^4}(x^4 + y^4 - 6x^2y^2)\cos(\omega t + \theta) \tag{26}$$

where any static component is neglected. This potential yields amplitudes for the E field components and $E^2(r)$ as follows:

$$E_x = -\frac{\partial\phi}{\partial x} = -\frac{4V_o}{r_o^4}(x^3 - 3xy^2) \tag{27}$$

$$E_y = -\frac{\partial\phi}{\partial y} = -\frac{4V_o}{r_o^4}(y^3 - 3yx^2) \tag{28}$$

$$E^2 = E_x^2 + E_y^2 = \frac{16V_o^2}{r_o^8}(x^2 + y^2)^3 = \frac{16V_o^2}{r_o^2}(\frac{r}{r_o})^6 \tag{29}$$

Thus, for an octapole field one has

$$V_{eff}(r) = 4K(\frac{r}{r_o})^6 \quad ,$$

where

$$K = \frac{q^2V_o^2}{m\omega^2r_o^2} \quad .$$

In a similar fashion the effective potential for a quadrupole field is given by

$$V_{eff}(r) = K(\frac{r}{r_o})^2 \tag{31}$$

The effective potentials for an octapole and a quadrupole with the same K value are shown in Fig. 24. It is immediately evident that the octapole field provides an effective potential which has higher walls than does the quadrupole field. As stated before, we believe that the octapole is better suited for "trapping" ions. Figure 25 shows the trajectory (x-y coordinates) for a singly charged ion in an octapole ion guide with $r_0 = 0.64$ cm, $\omega/2\pi = 7.5$ MHz, and $m = 32$ amu. The trajectory is generated by integrating out the equations of motion and not by using any effective potential approximations. The trajectory shown follows the course of the ion over a 112 μsec. period given an initial radial energy (ε_0) of ~1 eV. If this ion had a z-axis (the axis of the trap) energy of 0.65 eV, then it would take ~250 μsec. to traverse the length of the octapole (m/e = 32), and would "bounce" off the effective potential wall about 40 times during this time. Figure 26 shows the trajectories for singly charged ions with masses of 32 amu and 16 amu traveling with initial radial energies of 0.6 eV and 0.3 eV, respectively. The octapole field parameters are $V_0 =$ 200V, $\omega/2\pi = 15$ MHz, and $r_0 = 0.48$ cm. The trajectory is carried out for 20 μsec. for both ions. Figure 27 displays the probability of finding the ion of mass 32 at a reduced distance r/r_0 from the center of the ion guide, while Fig. 28 gives the radial energy proability distribution of the ion (with $\varepsilon_0 = 100$). Note that while the most probable energy is the initial energy, the ion has radial energies substantially below and above its initial value for very significant amounts of time.

When using the octapole to confine ions one must always bear in mind that the equation describing its effective potential (Eq. 30) is valid only if the approximating conditions used in its derivation are met.[10] Also note that the effective potential is a function not only of the angular frequency, but also of the mass of the ion: ions of different mass coexisting in the ion guide will be governed by different effective potentials. This means that an octapole guide operating so as to guide a reactant ion, say H_2^+, may fail to effectively confine the product mass created within the guide if the product is more massive than the reactant. For example, in the following reaction

$$H_2^+ + D_2 \longrightarrow D_2H^+ + H \quad , \tag{32}$$

the product has greater mass than the reactant. One must ensure that the ion guide is operating with its parameters set so it will confine both species, or measurements made on the system will be misleading.

The octapole ion guide can also operate as an ion trap if ion "mirrors" cap the ends of the guide. Such an arrangement is used in our system. A neat way to observe the trapping effect is to lower the potential of the entrance mirror for a short length of time and then raise it again. This allows a bunch of ions into the trap. This bunch of ions can now be made to "bounce" back and forth in the ion guide by having the mirrors totally reflecting. But if no ions escape to the detector, one observes nothing. Now if the exit mirror is made partially transmitting, then every time the bunch of ions strikes it, some ions will pass through to be detected and the rest will be reflected back down the axis of the guide. The results of such a measurement (ion counts vs time) are shown in Fig. 29. Note that at the end of each round trip the transmitted peak becomes broader until a peak no longer exists and all one gets is a constant leakage of ions from the trap. At this point there is no longer any bunching in the trap: it is homogeneously filled with ions which have a broad kinetic energy distribution. This loss of bunching occurs because the "mirrors" are scattering the ions back into the guide in a diffuse manner. This transfers kinetic energy originally in the z coordinate to the x and y coordinates until the energy is equally shared. This means that ions which have been stored in the trap for more than one millisecond become translationally very hot. This is a bad consequence in terms of the photofragment spectroscopy because of the large Doppler widths which result from it.

Another problem associated with our octapole mirrors is that they leak ions out of the trap when they are not supposed to. The leaking ions are not escaping down the axis of the spectrometer since none reach the detector. They must be popping out to the sides or slamming into metal surfaces. It seems though, that only ions above some critical kinetic energy leak out; for once these escape, the remainig ions are trapped efficiently. Figure 30 shows O_2^+ trapped out to 100 msec. Note that almost all the ions which will escape have done so by 50 msec. of trapping time. Figure 31 shows the trapping efficiency out to 5 minutes. This figure clearly demonstrates that after the initial loss of ions, the remaining ions are trapped very well indeed. Overall, the RF octapole ion guide works quite satisfactorily. The use of the guide as an ion trap may be made more efficient by carefully designing the mirrors at the ends of the octapole. These mirrors must be able to reflect the ions well while not disturbing the effective potential created by the RF field.

Aside from the obvious assets of large ion-laser overlap, containment of all product ions, and the use of the trap to allow the relaxation of excited states, one may also use the trapping feature for temporal discrimination. That is, if the firing of the light source creates electronic noise or photon induced background in the detector, one may confine the meaningful light correlated products in the ion trap until the noise settles down--then allow the products to pass on to the detector. This capability is very advantageous to have since its application can increase the signal-to-noise ratio significantly.

F. The QMF and Detector

After the octapole ion guide has done its thing with the ions, they are extracted from the guide and focused into a commercially produced quadrupole mass filter (Extranuclear Laboratories, Inc.). The mass filter is set to transmit the laser correlated products to the final stage of the spectrometer in a typical flop-in type experiment. The ions which exit the quadrupole are directed towards a Daly[11] detector or a Ceratron[12] electron multiplier. The Ceratron, although having less gain then the Daly scheme, is convenient because both positive and negative ions can be detected by just changing the polarity of the power supplies. Both detectors were configured for pulse counting.

G. The Light Source

The light source used with the spectrometer is based on a pulsed Nd:YAG (Quanta-Ray DCR) laser pump which produces up to 700 mj (10 nsec. pulse duration) of 1.06 μ radiation. This fundamental can be doubled, tripled, or quadrupled with harmonic generating crystals. These harmonics are used either directly in the spectrometer or are used to pump a dye laser (Quanta-Ray PDL) to yield tunable visible radiation. In addition, the 1.06 μ light can be sent into an optical parametric oscillator[13] (OPO) thus producing tunable infrared (~1.5 to 4.3μ) light.

H. Data Acquisition and Control

Many of the spectrometer functions are automated making long data taking runs requiring many changes in instrumental parameters possible without much personal attention. A microcomputer (DEC LSI-11) performs the task of controlling these parameters and of collecting the data being generated. Communication between the computer and the spectrometer is accomplished via A2D and D2A converters in the computer backplane, in addition to having a CAMAC crate interfaced to the computer. A programmable clock controls the trapping time of the octapole ion trap (see Fig. 32). Other accessories include a video terminal, a plotter, a DMA driven display scope, a printer and a floppy disk. The program used to operate the system is written in assembly language (see Appendix). It allows one to operate the sector magnet, the octapole ion trap, the quadrupole mass filter, and the laser via computer command.

I. Performance: Photofragmentation in the Visible

The performance of the spectrometer was evaluated by observing the photofragmentation of some ions using visible light. This choice was made because with visible light one does not need a very weak bond to be present in the ion for photofragmentation to occur and so ions created by an electron impact ionizer could be used. Using the Nd:YAG pumped dye laser as the light source, CH_3I^+, $C_6H_6^+$ and O_2^+ were photofragmented.

First consider the CH_3I^+ ion. This ion can dissociate yielding the CH_3^+ ion[14] if it absorbs a photon with sufficient energy. Figure 33 shows the laser correlated M/e = 15 product as a function of wavelength from 5000 to 5400Å sampled in 0.96Å steps. The structure seen is due to transitions to a bound upper state which subsequently leads to fragmentation.[14a] As another trial, the photofragmentation of $C_6H_6^+$ was observed. In this case the dye laser was not frequency scanned; instead, the laser correlated mass spectrum was generated at a single frequency. Figure 34 shows the result. It is seen that at the laser intensity used, two daughter fragments are produced, $C_3H_3^+$ and $C_4H_4^+$, as a result of a multiphoton absorption process.[15] Finally, Fig. 35 shows the laser correlated product mass spectrum of O_2^+ being irradiated by 5100Å light. In each of these cases, the photofragmentation events were trivial to observe. But of these trials, only the photofragmentation of O_2^+ was studied in more detail; the results of the work on O_2^+ is the subject of Chapter III.

REFERENCES

1. (a) E. C. Horning, D. I. Carroll, I. Dzidic, K. D. Haegele, M. G. Horning, and R. N. Stillwell, J of Chromatographic Sci. 12, 725 (1974).

(b) M. McKeown and M. W. Siegel, American Laboratory, November (1975).

(c) M. W. Siegel and W. L. Fite, J Phys. Chem. 80, 2871 (1976).

(d) M. Armbruster, H. Haberland, and H. G. Schindler, Phys. Rev. Lett. 47, 323 (1981).

2. (a) H. A. Enge, Rev. Sci. Instrum. 30, 248 (1959).

(b) C. F. Giese, Rev. Sci. Instrum. 30, 260 (1959).

(c) C-S. Lu and H. E. Carr, Rev. Sci. Instrum. 33, 823 (1962).

(d) M. L. Vestal, "Crossed-Beam Studies of Ion-Molecule Reactions" Ph.D. Thesis, University of Utah, 1975.

3. (a) E. T. Kinzer and H. Carr, Rev. Sci. Instrum. 30, 1132 (1959).

(b) S. Taya, I. Kanomatata, H. Hirose, T. Noda, and H. Matsuda, Int. J. Mass. Spect. and Ion Phys. 26, 237 (1978).

4. H. D. Zeman, Rev. Sci. Instrum. 48, 1079 (1977).

5. Ion Beams: with Application to Ion Implantation, R. G. Wilson and G. R. Brewer, John Wiley and Sons (1973).

6. P. H. Dawson and J. E. Fulford, Int. J. Mass. Spectra. and Ion Phys. 42, 195 (1982).

7. J. H. D. Eland, J. Phys. E.: Sci. Instrum. 11, 969 (1978).

8. J. E. Pollard, "Photoelectron Spectroscopy of Supersonic Molecular Beams," Ph.D. Thesis, University of California, Berkeley, CA (1982).

9. Mechanics, L. D. Landau and E. Lifschitz, p. 93, Pergamon Press, Oxford, 1960.

10. E. Teloy and D. Gerlich, Chem. Phys. 4, 417 (1974).

11. R. N. Daly, Rev. Sci. Instrum. 31, 264 (1960).

12. S. Ehara, K. Wakino, H. Yamamoto, JEE, June, 1974.

13. (a) S. E Harris, Proc. IEEE 57, 2096 (1969).

(b) R. L. Byer and R. L. Herbst, Top. Appl. Phys. 16, 81 (1977).

14. (a) S. P. Goss, D. C. McGilvery, J. D. Morrison and D. L. Smith, J. Chem. Phys. 75, 1820 (1981).

(b) S. P. Goss, J. D. Morrison, and D. L. Smith, J. Chem. Phys. 75, 757 (1981).

15. (a) P.N.T. Van Velzen and W. J. Van Der Hart, Chem. Phys. 61, 325 (1981).

(b) U. Boesl, H. J. Neusser, and E. W. Schlag, J. Chem. Phys. 72, 4327 (1980).

FIGURE CAPTIONS

Fig. 1. General schematic of source regions.

Fig. 2. The ion lens system following the ion source: (1) nozzle assembly, (2) skimmer, (3) accelerating and focusing elements, (4) and (5) ion beam deflectors.

Fig. 3. Detail of the supersonic nozzle ion source.

Fig. 4. Mass scans of ions produced by the nozzle source. The top scan is with a light extra flow of gas through the nozzle: the ethanol impurity peaks disappear with this extra flow.

Fig. 5. Two mass scans of ions produced by the nozzle source. The bottom scan is with no extra flow; the top is with a heavy flow. Impurities peaks disappear and water cluster distribution shifts downward with extra flow.

Fig. 6. Extra flow through the nozzle can produce unwanted rsults. The top mass scan taken with extra flow has no $Ar \cdot (H_2O)_nH^+$ clusters.

Fig. 7. The cluster size distribution from the nozzle source shifts to smaller average size as the electric field between the nozzle and the skimmer is increased.

Fig. 8. Illustration of the use of nozzle temperature and flow to enhance the quantity of O_2^+ produced.

Fig. 9. Mass scans of nozzle source with parameters set to optmize the creation of $Ar \cdot H_3O^+$.

Fig. 10. Schematic of Region III: the magnetic sector.

Fig. 11. The quadrupole lens pairs (QLP) transforms a rectangular beam into a wedge-shaped beam.

Fig. 12. A comparison of mass spectra taken with the QLP on (top) and QLP off (bottom).

Fig. 13. Schematic of Region IV, the interaction region: (1) to Region III, (2) to window, (3) to ion detector, (4) quadrupole field deflector, (5) decelerator, (6) einzel lens (7) octapole trap, (8) quadrupole mass filter.

Fig. 14. A closer view of the quadrupole field deflector.

Fig. 15. Calculated equipotentials for the quadrupole field deflector.

Fig. 16. Solution through the quadrupole field deflector with the least aberration: relative ion energy of 1.00.

Fig. 17. A narrow beam of ions which is not monoenergetic deflected by the QFD.

Fig. 18. A wide monoenergetic ion beam deflected by the QFD.

Fig. 19. A converging monoenergetic beam deflected by the QFD.

Fig. 20. A solution with a higher aberrations with a relative ion energy of 1.925.

Fig. 21. A solution with higher aberrations with a relative ion energy of 1.858.

Fig. 22. A cross section of the octapole trap: r_o is the radius of the circle inscribed within the rods and r is the distance of the ion from the origin. Also shown schematically is the tank coil and the exciter loop.

Fig. 23. A close up view of one end of the octapole assembly.

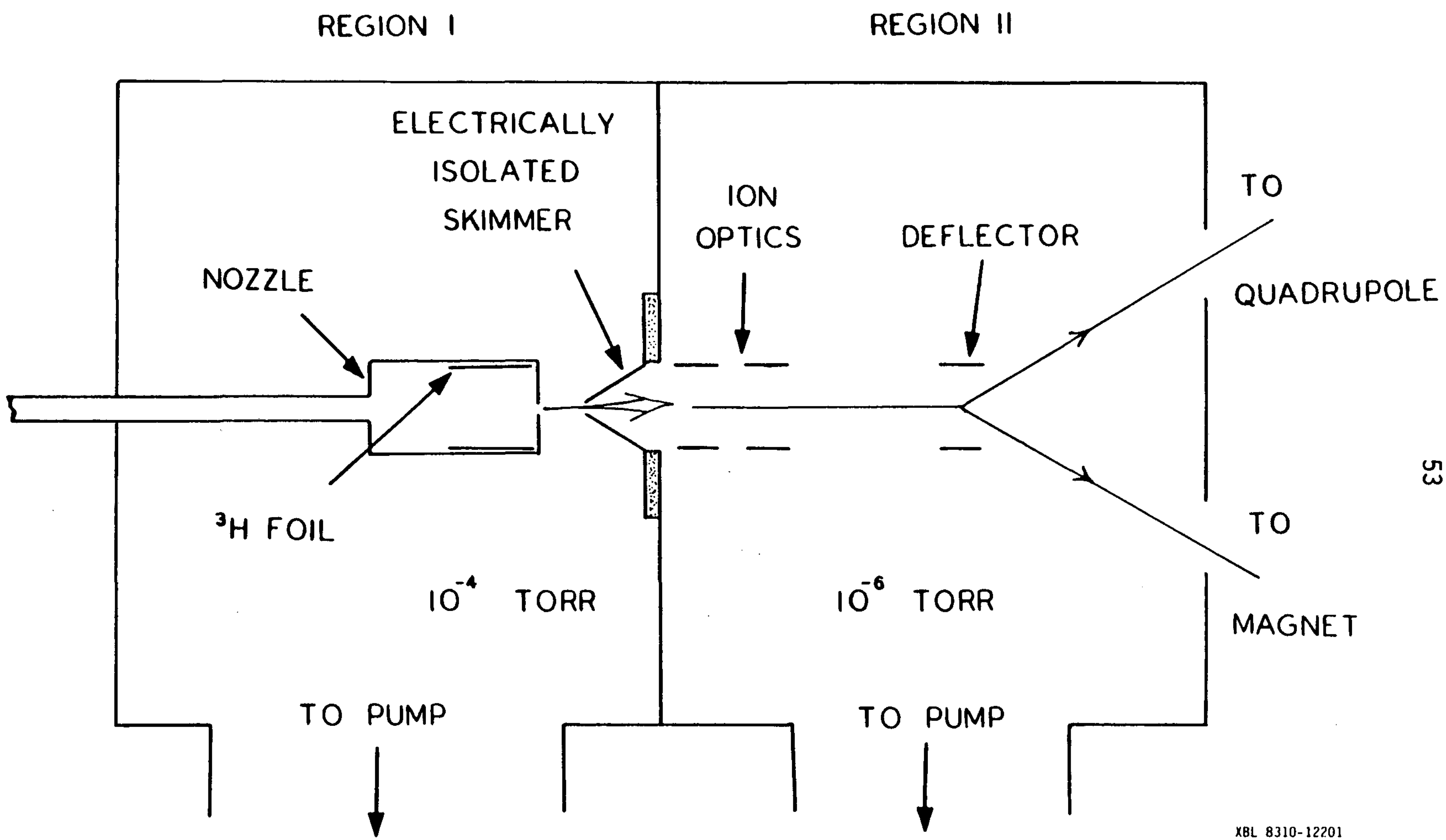
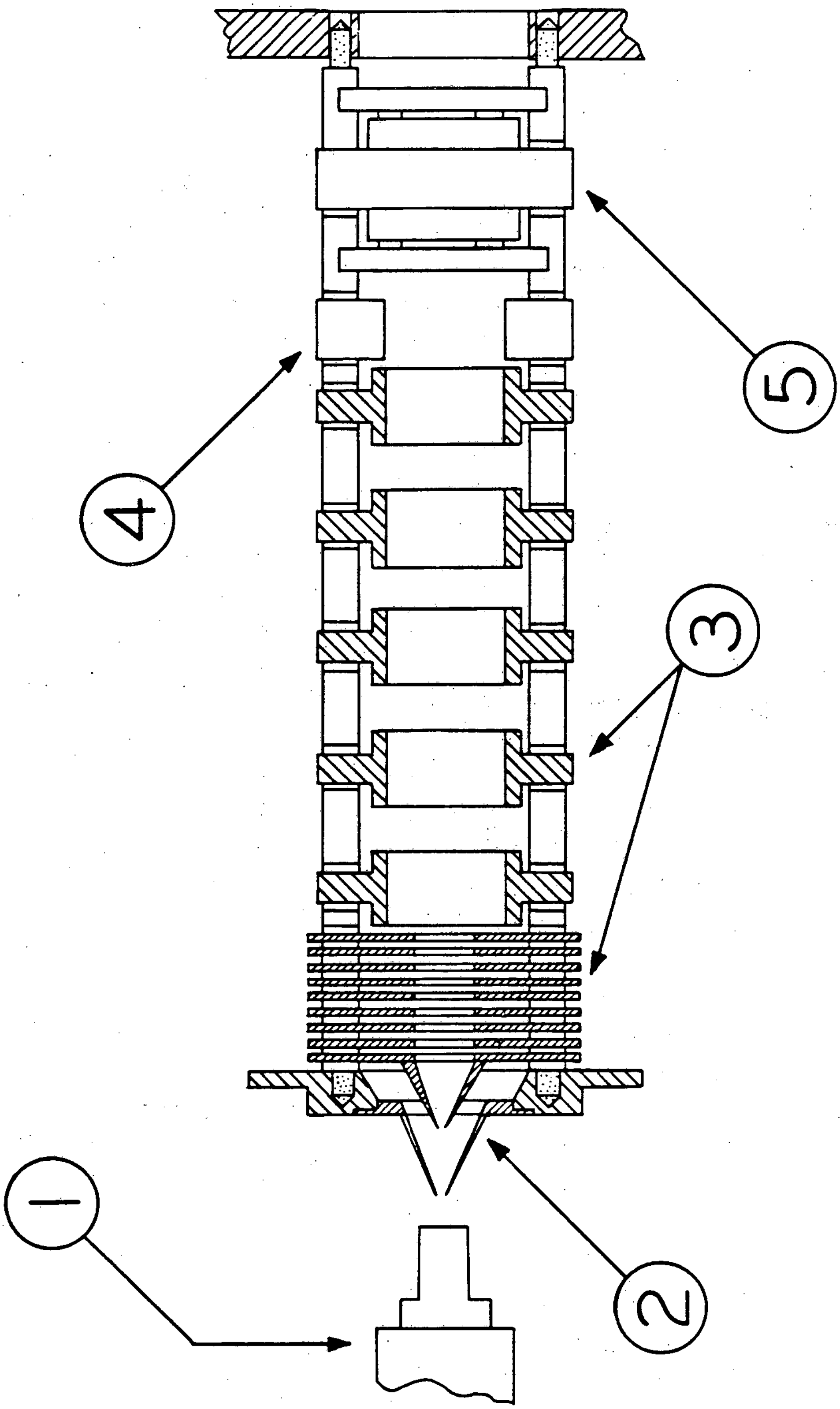
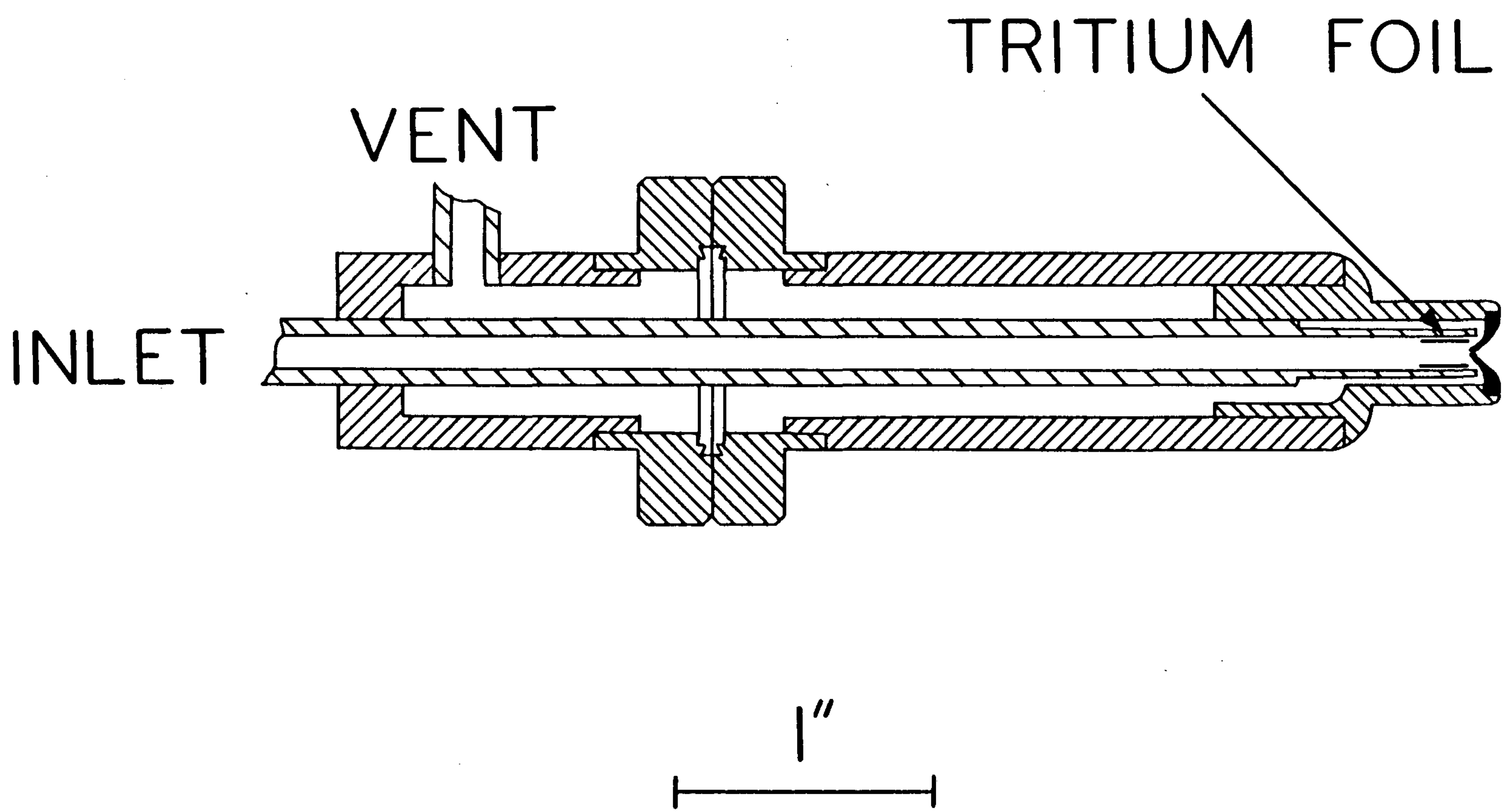

REGION I
REGION II
ELECTRICALLY
ISOLATED
SKIMMER
ION
OPTICS
DEFLECTOR
TO
QUADRUPOLE
NOZZLE
³H FOIL
TO
MAGNET
10⁻⁴ TORR
10⁻⁶ TORR
TO PUMP
TO PUMP
XBL 8310-12201

Figure 1



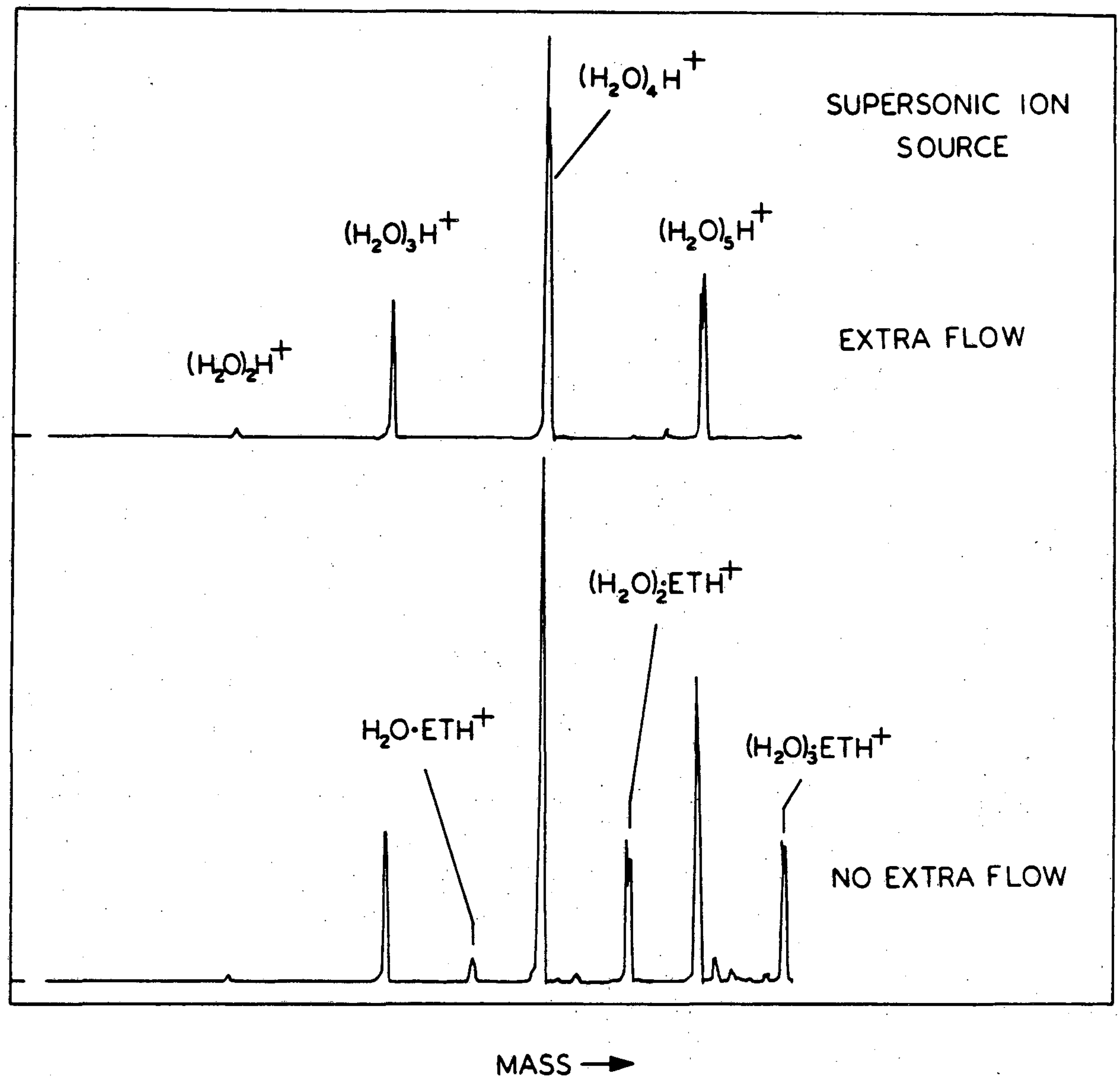
XBL 795-9759

Figure 2.



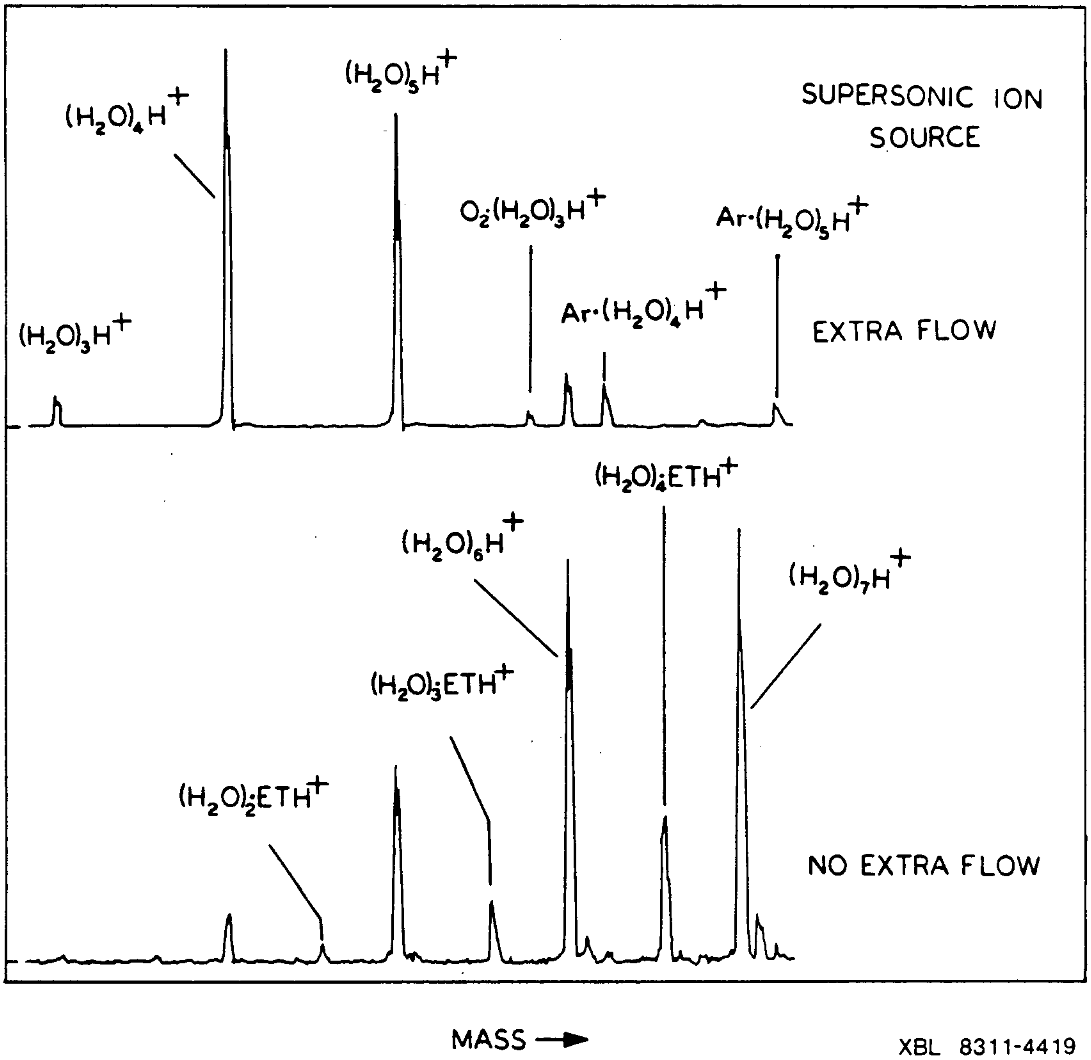
XBL 795-9758

Figure 3



XBL 8310-12202

Figure 4



$(H_2O)_4H^+$
$(H_2O)_5H^+$
SUPERSONIC ION SOURCE
$O_2\cdot(H_2O)_3H^+$
$Ar\cdot(H_2O)_5H^+$
$Ar\cdot(H_2O)_4H^+$
$(H_2O)_3H^+$
EXTRA FLOW
$(H_2O)_4\cdot ETH^+$
$(H_2O)_6H^+$
$(H_2O)_7H^+$
$(H_2O)_3\cdot ETH^+$
$(H_2O)_2\cdot ETH^+$
NO EXTRA FLOW
MASS →
XBL 8311-4419

Figure 5

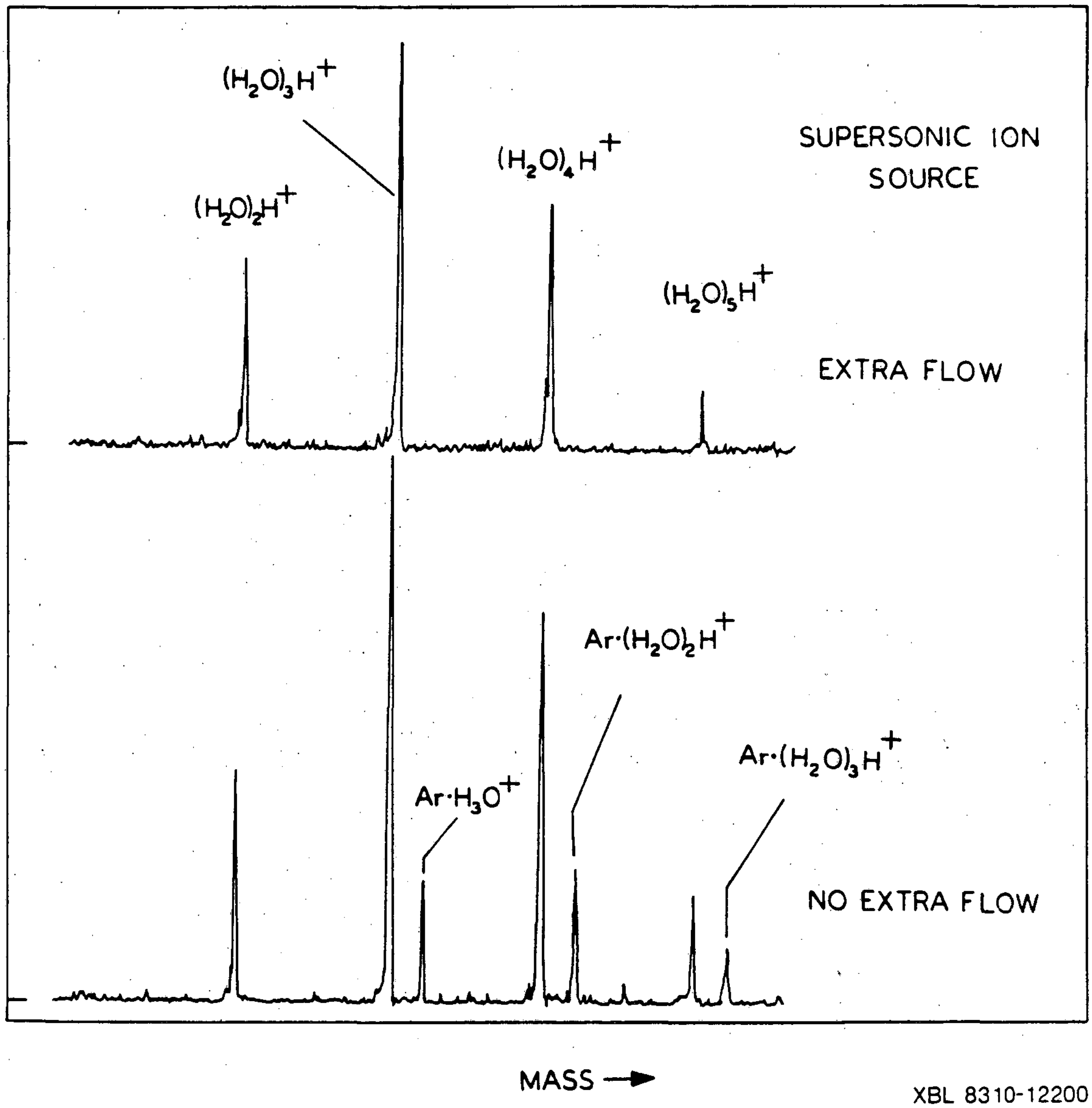
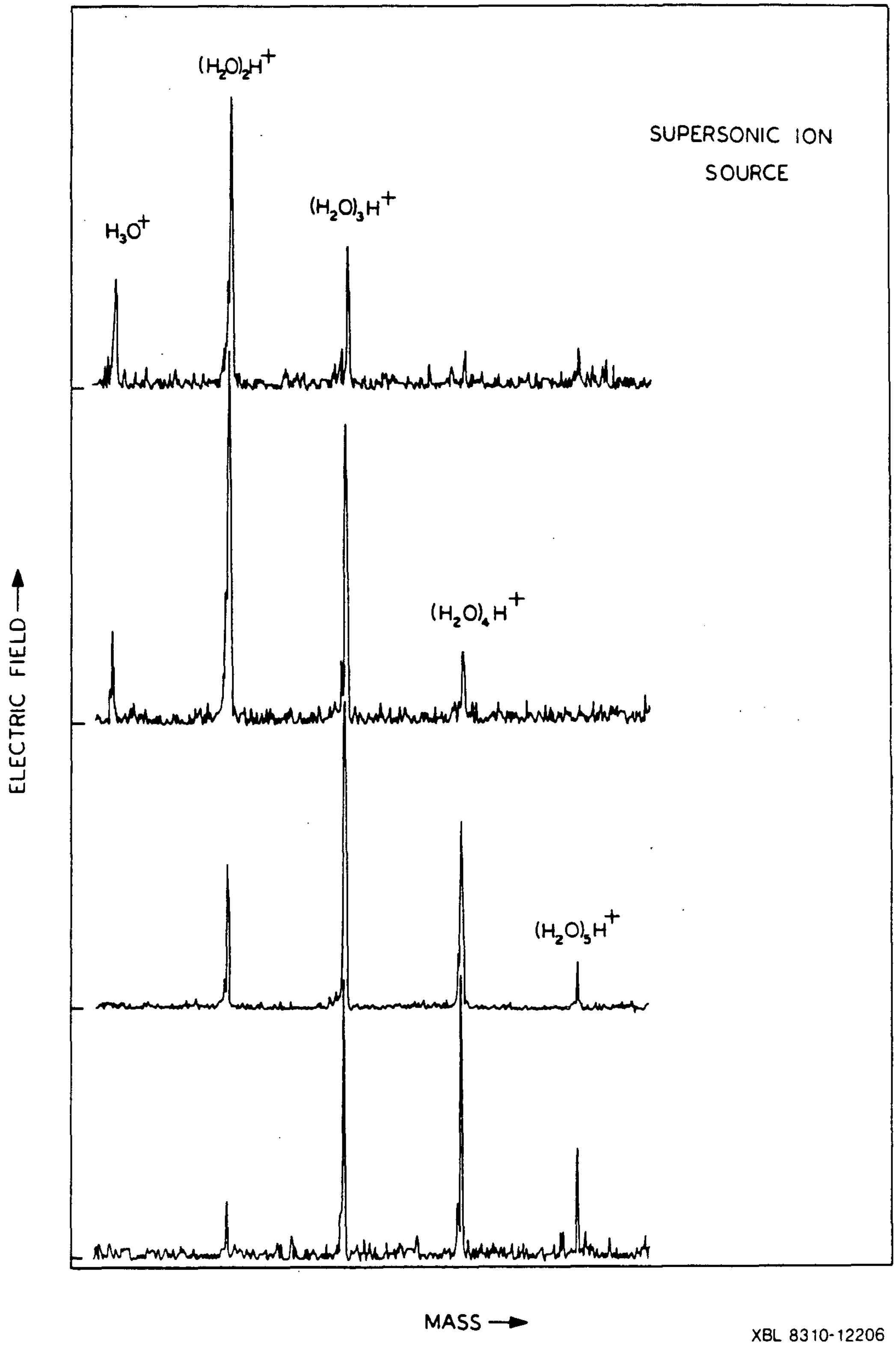

$(H_2O)_3H^+$
$(H_2O)_2H^+$
$(H_2O)_4H^+$
$(H_2O)_5H^+$
SUPERSONIC ION SOURCE
EXTRA FLOW
$Ar \cdot (H_2O)_2H^+$
$Ar \cdot (H_2O)_3H^+$
$Ar \cdot H_3O^+$
NO EXTRA FLOW
MASS →
XBL 8310-12200

Figure 6



$(H_2O)_2H^+$
SUPERSONIC ION SOURCE
H_3O^+
$(H_2O)_3H^+$
$(H_2O)_4H^+$
$(H_2O)_5H^+$
ELECTRIC FIELD →
MASS →
XBL 8310-12206

Figure 7

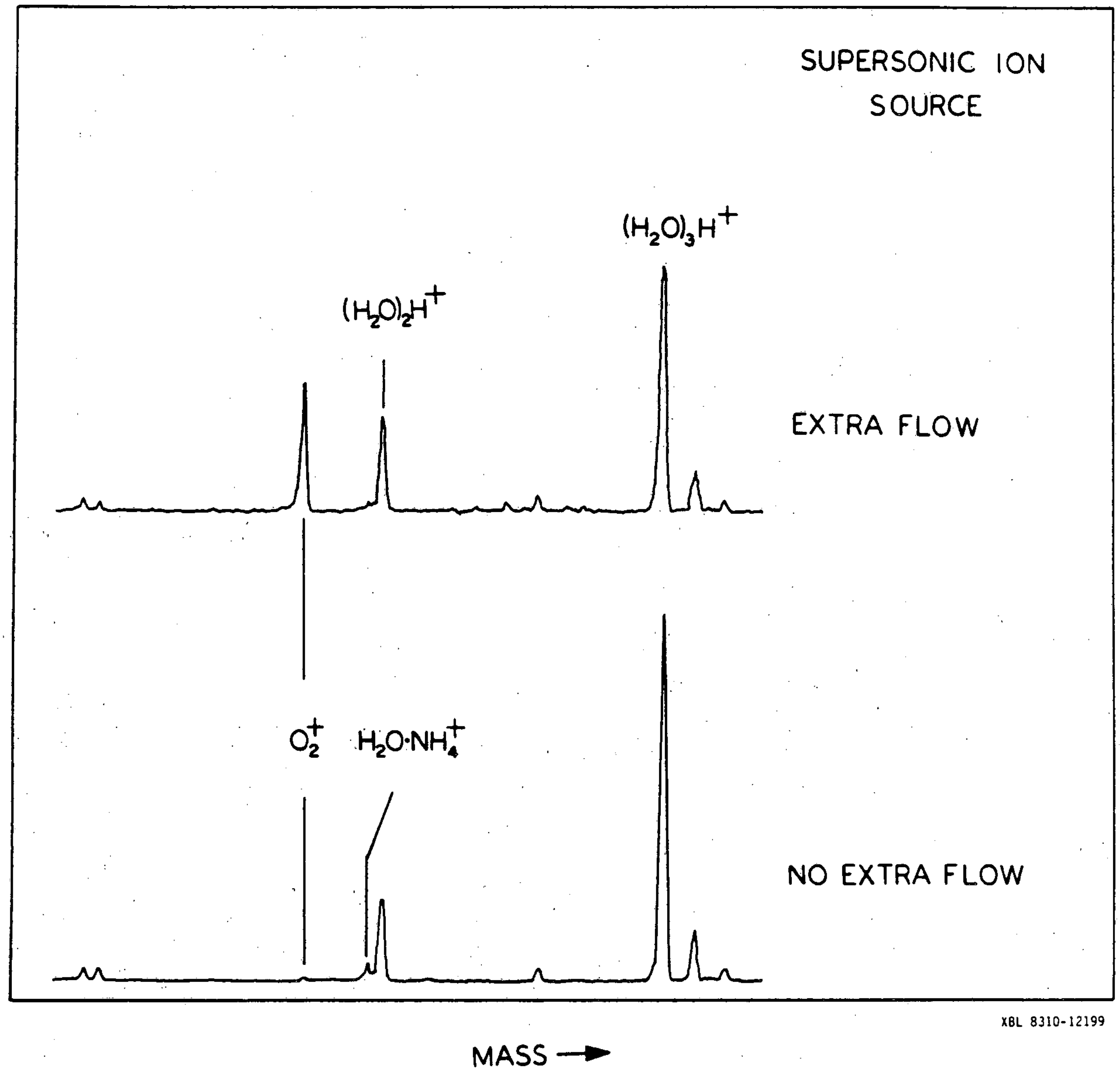

SUPERSONIC ION SOURCE
$(H_2O)_3H^+$
$(H_2O)_2H^+$
EXTRA FLOW
O_2^+
$H_2O \cdot NH_4^+$
NO EXTRA FLOW
XBL 8310-12199
MASS →

Figure 8

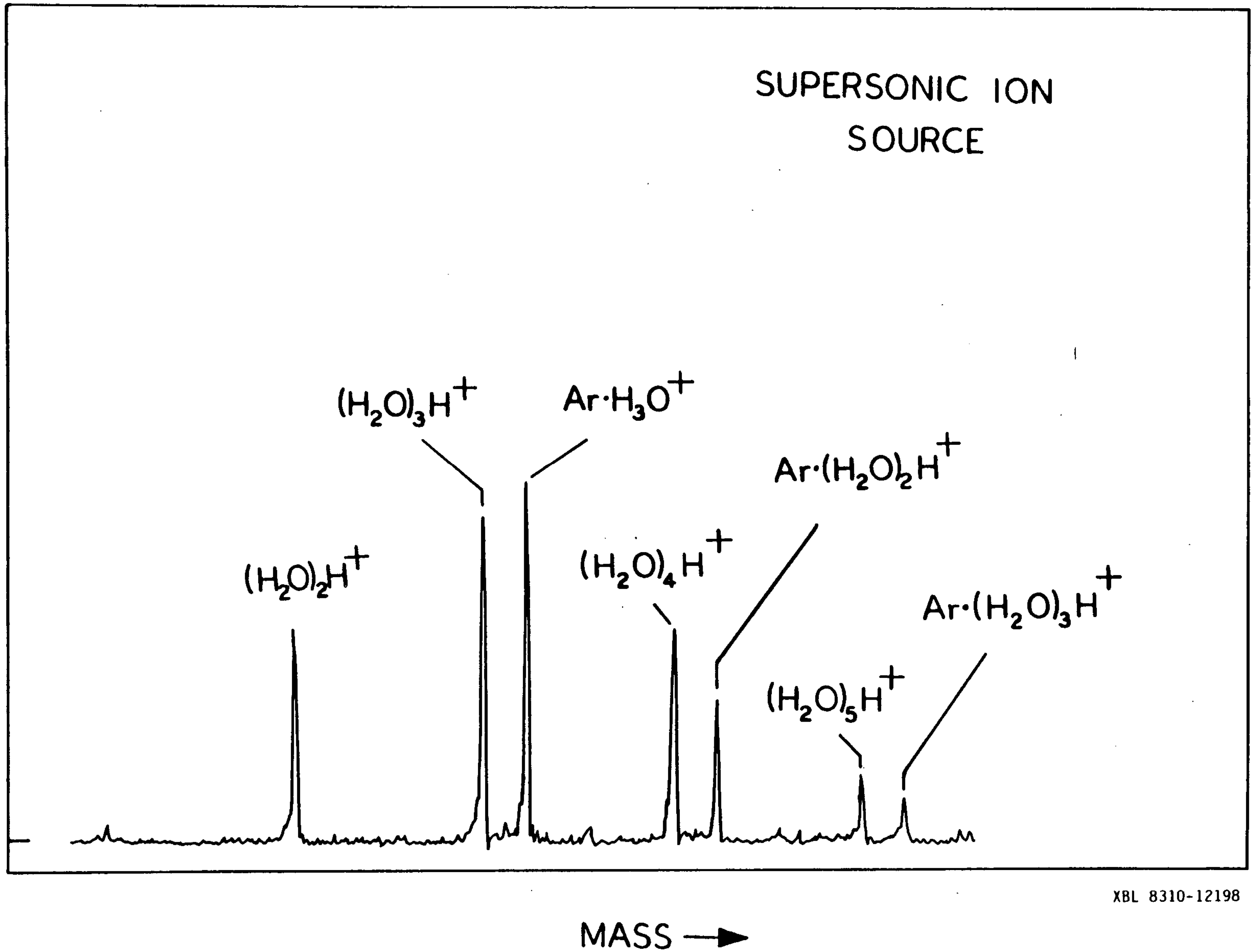
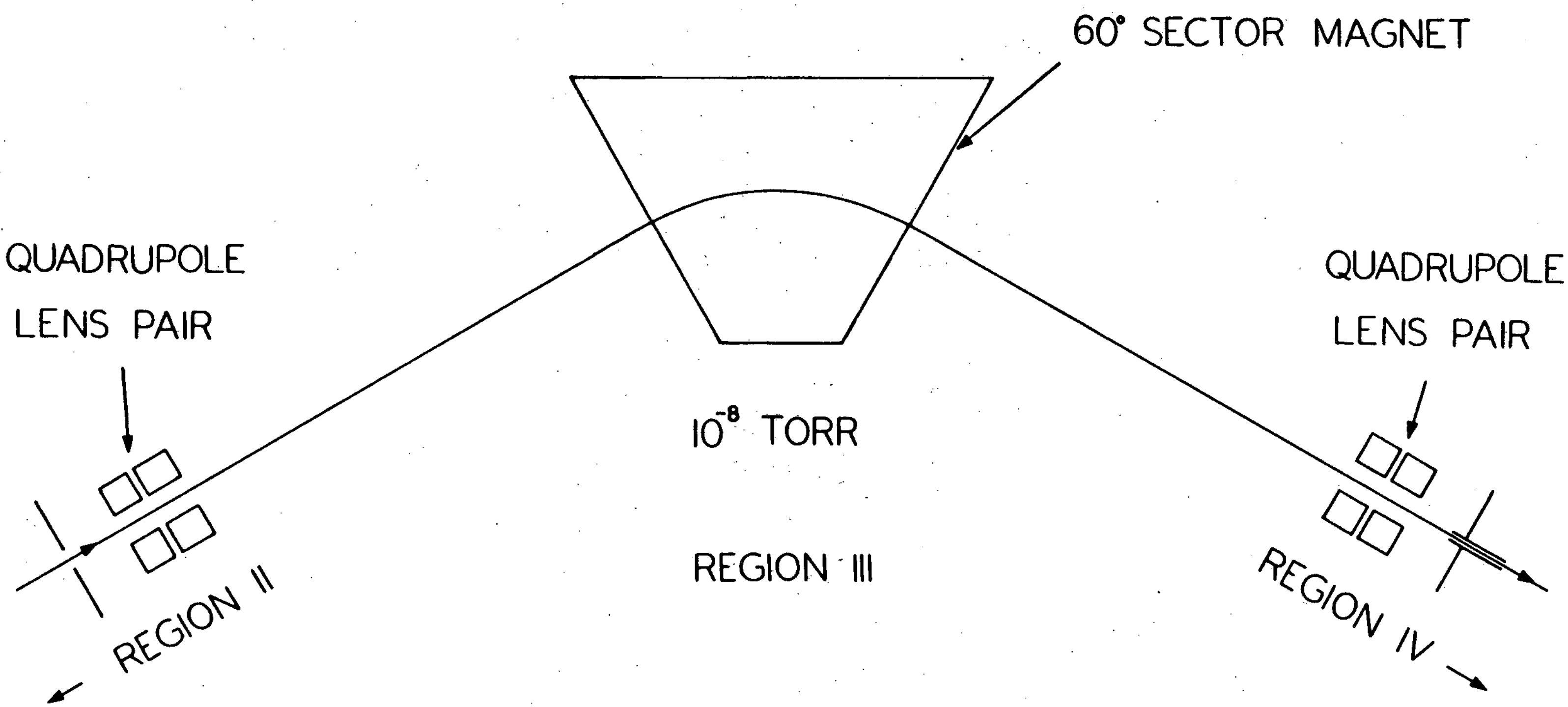
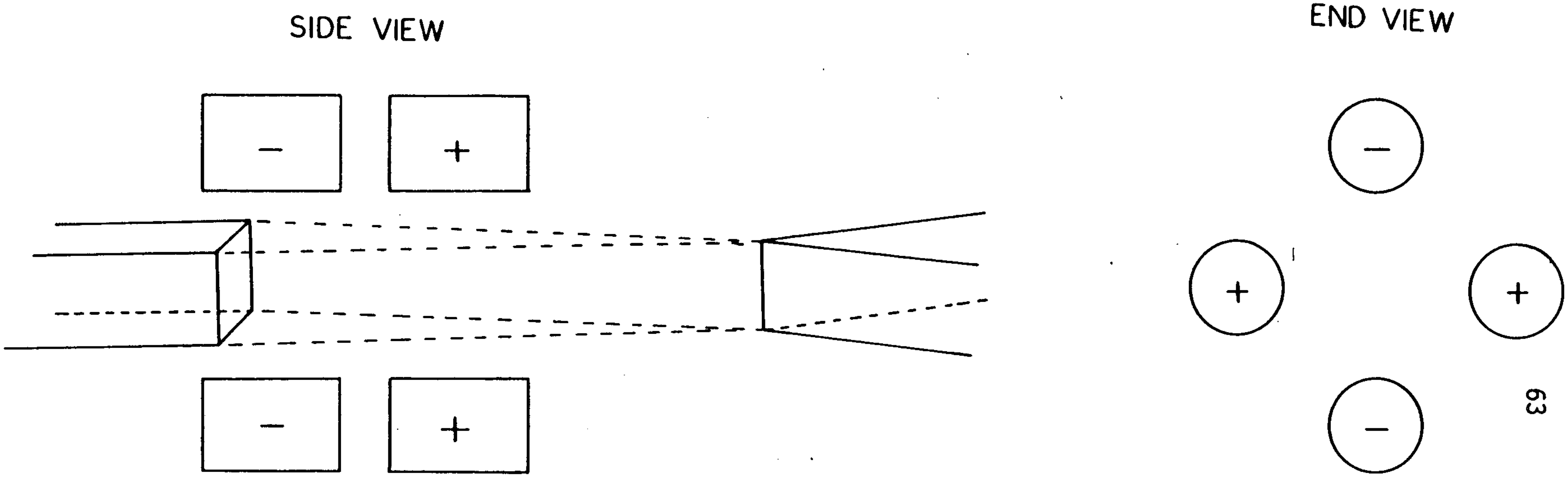

SUPERSONIC ION SOURCE
(H2O)2H+
(H2O)3H+
Ar·H3O+
(H2O)4H+
Ar·(H2O)2H+
(H2O)5H+
Ar·(H2O)3H+
XBL 8310-12198
MASS

Figure 9



XBL 8310-12208

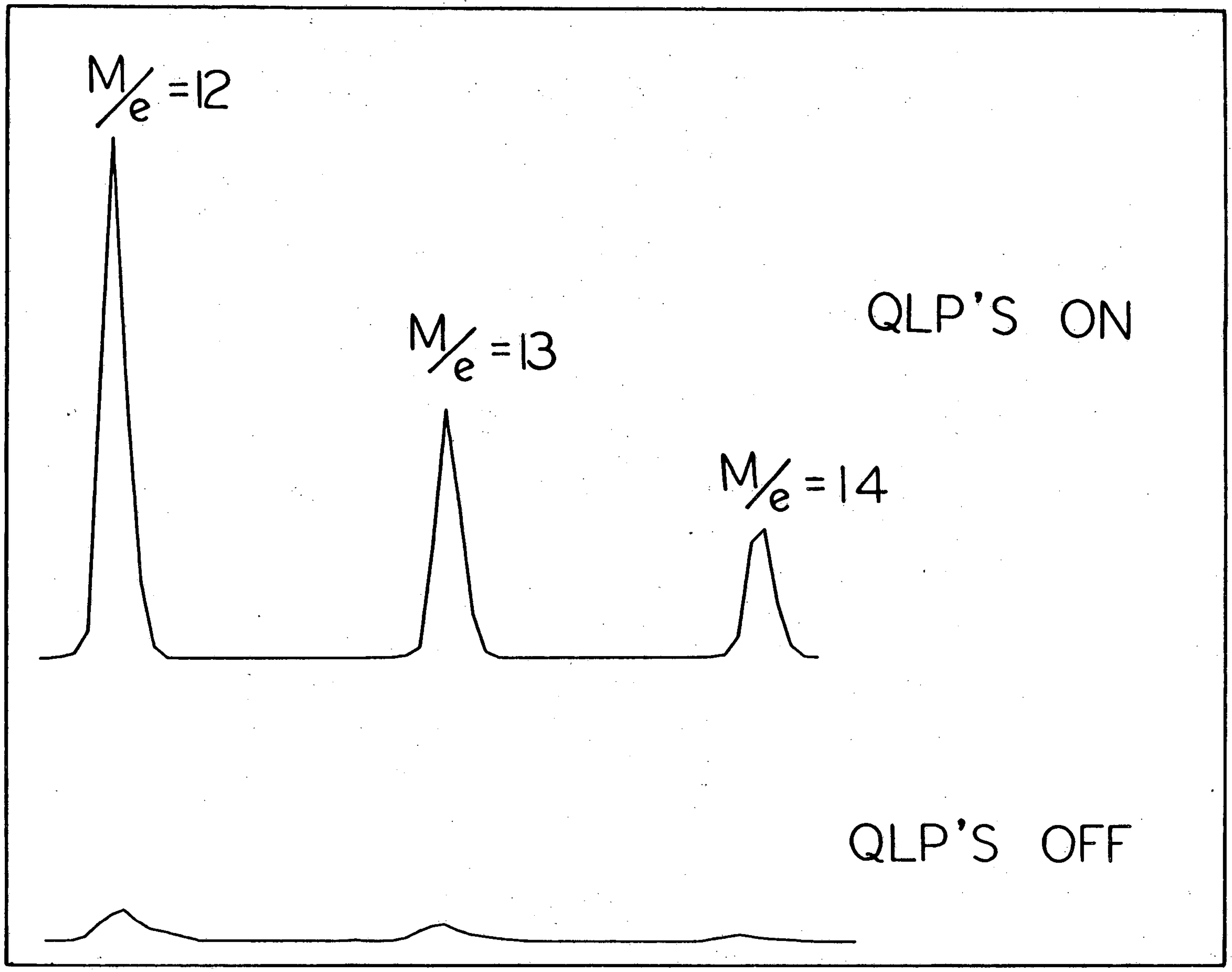
Figure 10



QUADRUPOLE
LENS PAIR

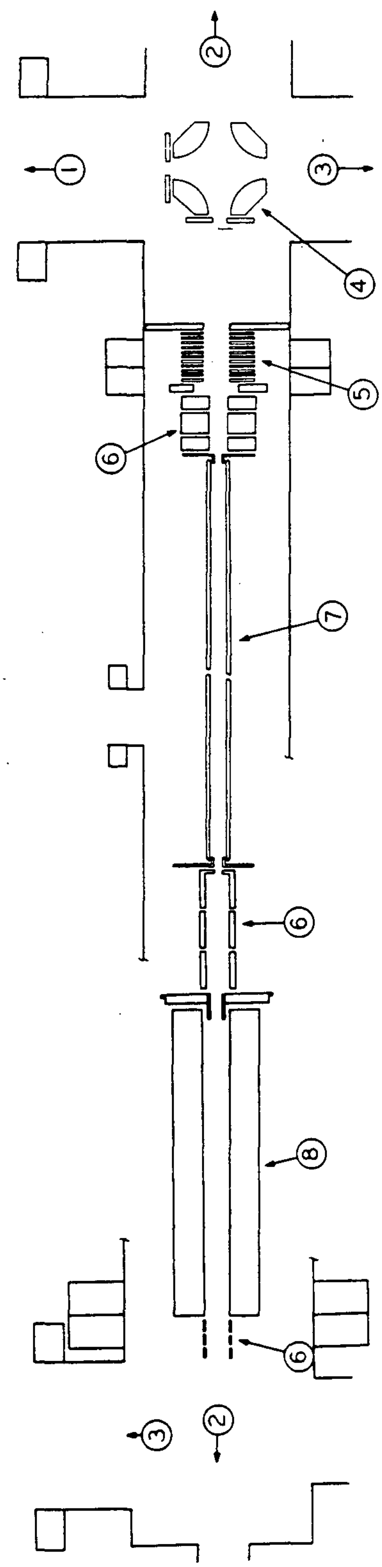
Figure 11

XBL 8310-12203



XBL 8310-12205

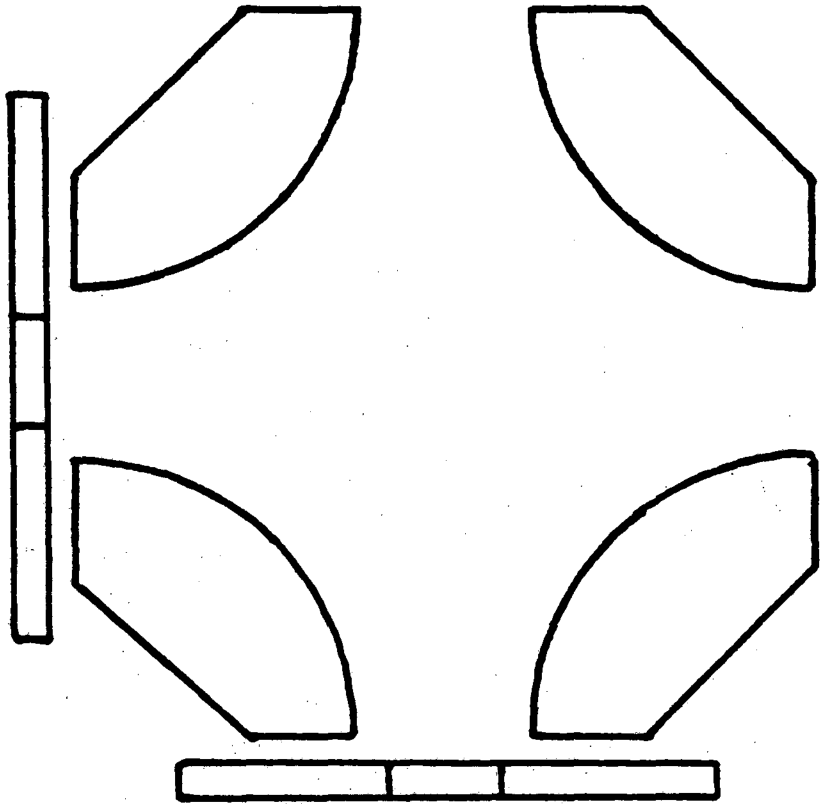
Figure 12



XBL 8310-12210

REGION IV

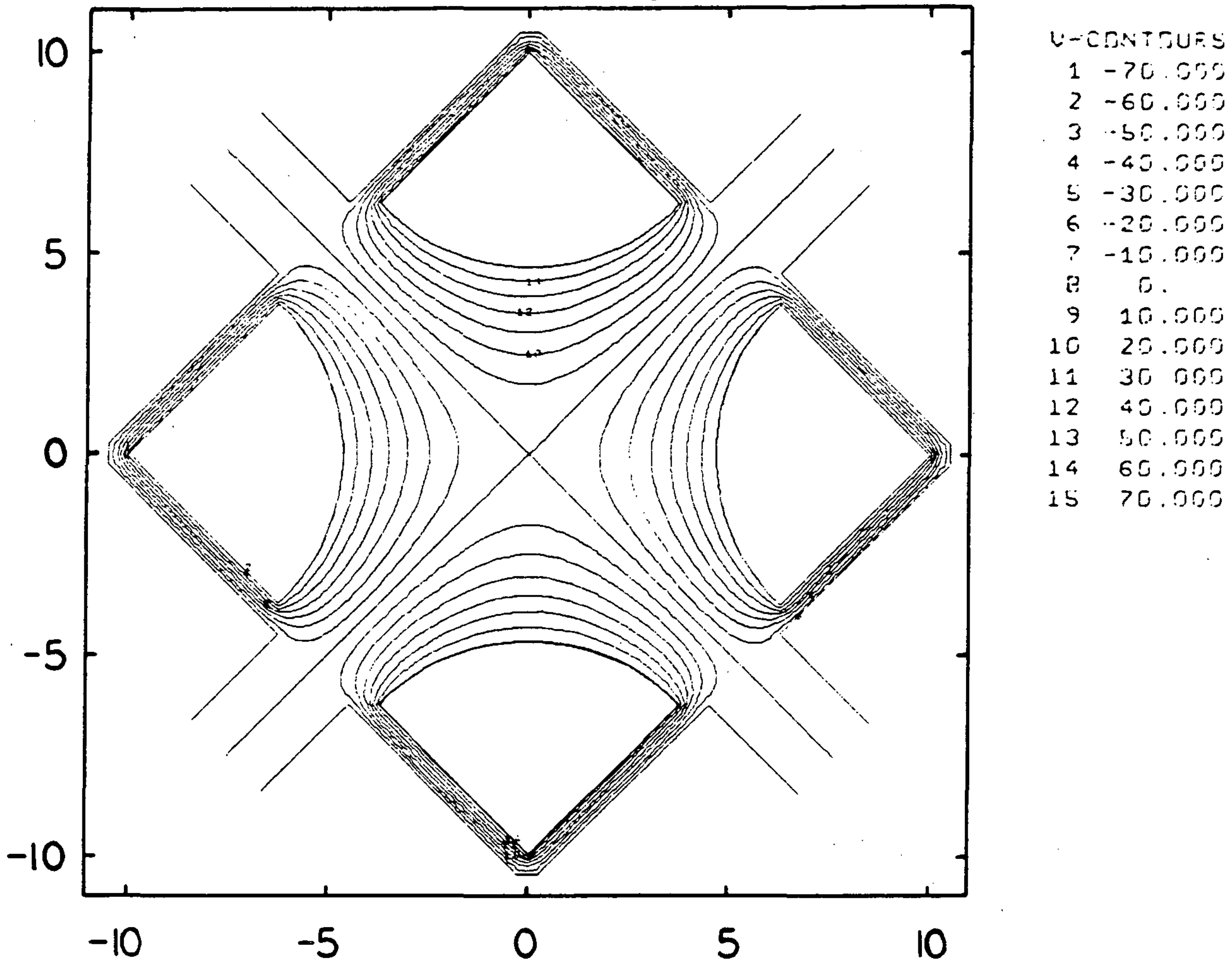
Figure 13



QUADRUPOLE FIELD
DEFLECTOR

XBL 8310-12196

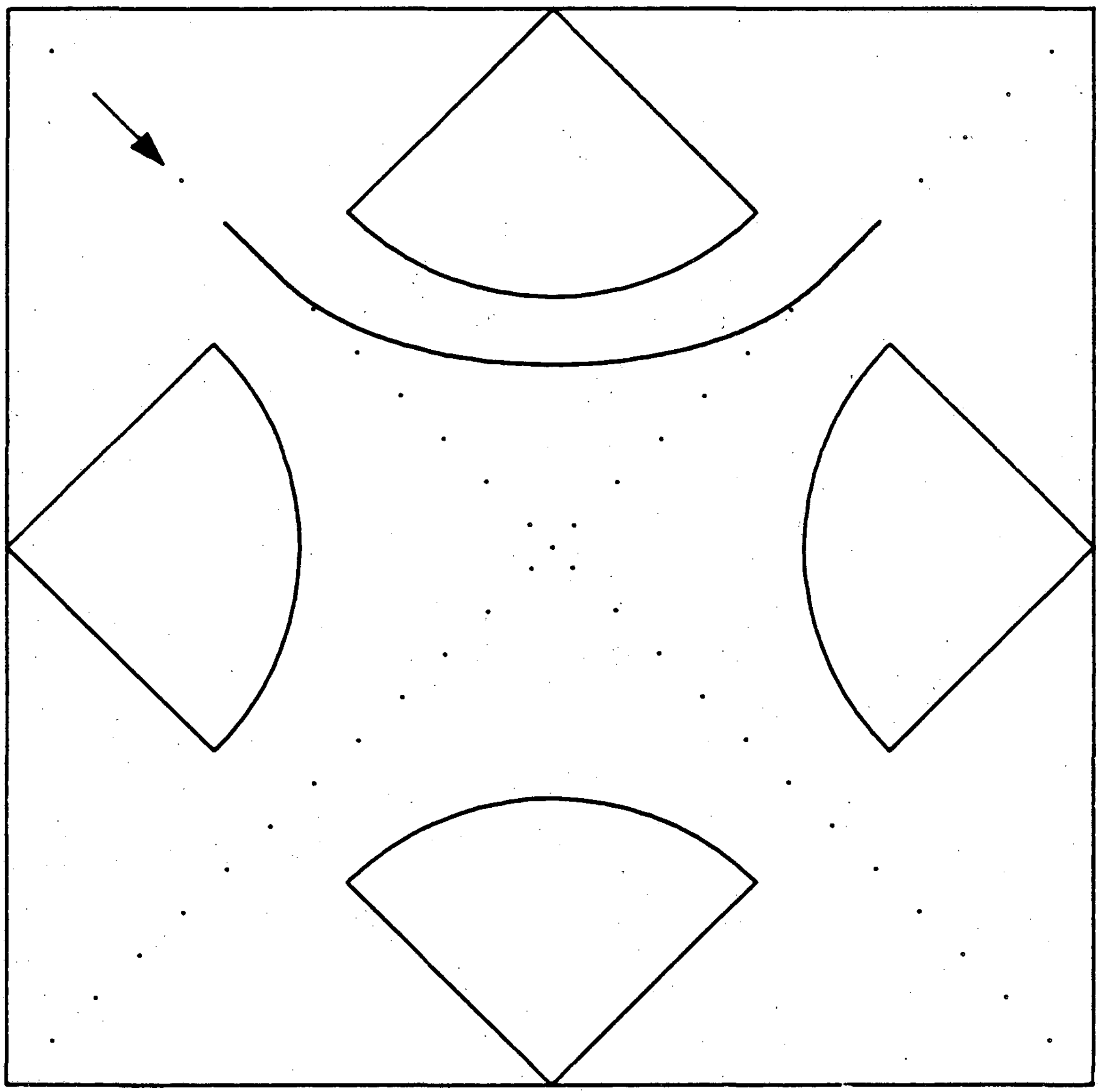
Figure 14



QFD EQUIPOTENTIAL PLOT

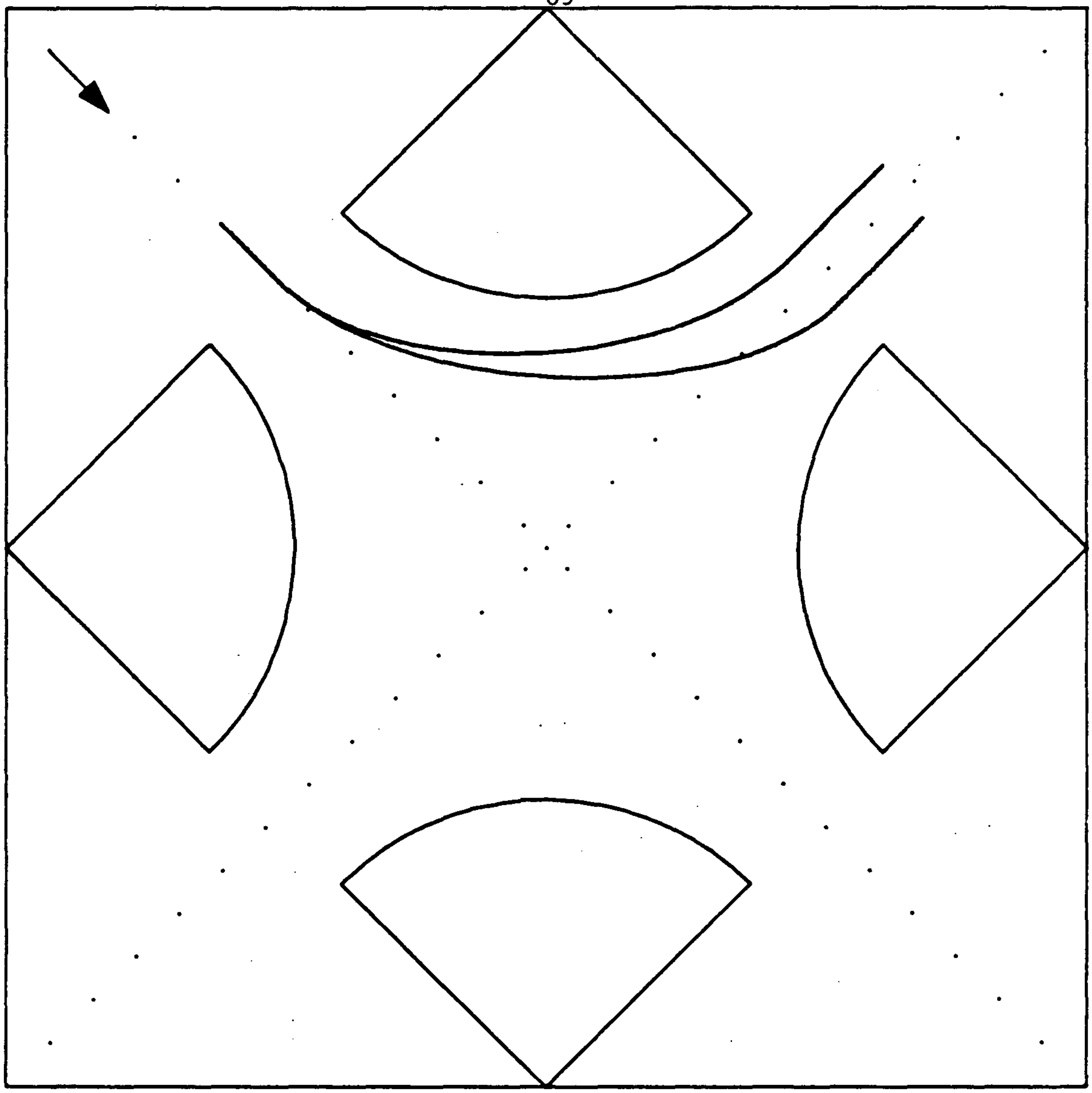
-- XBL 8310-12197 --

Figure 15



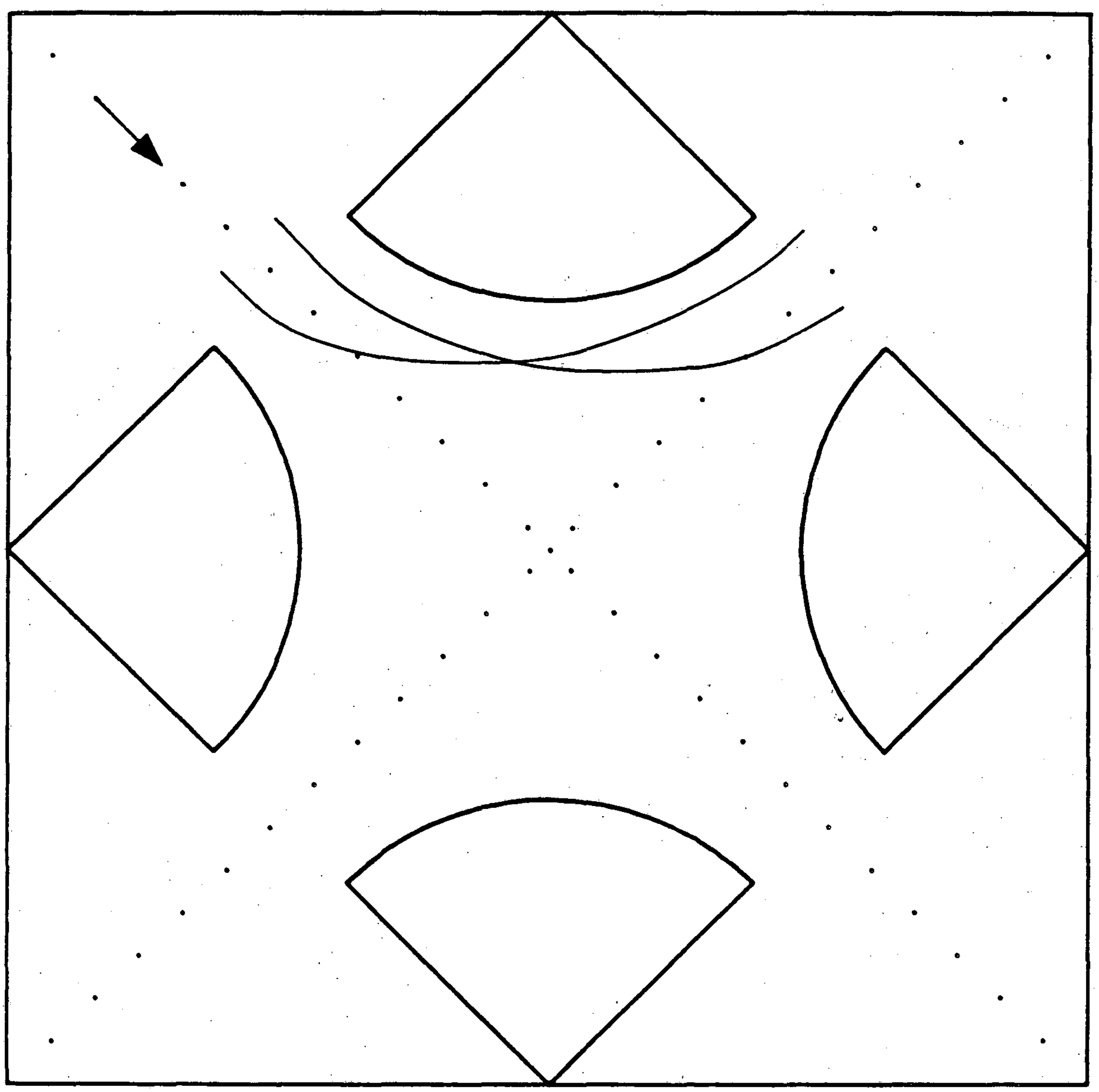
XBL 8310-12184

Figure 16



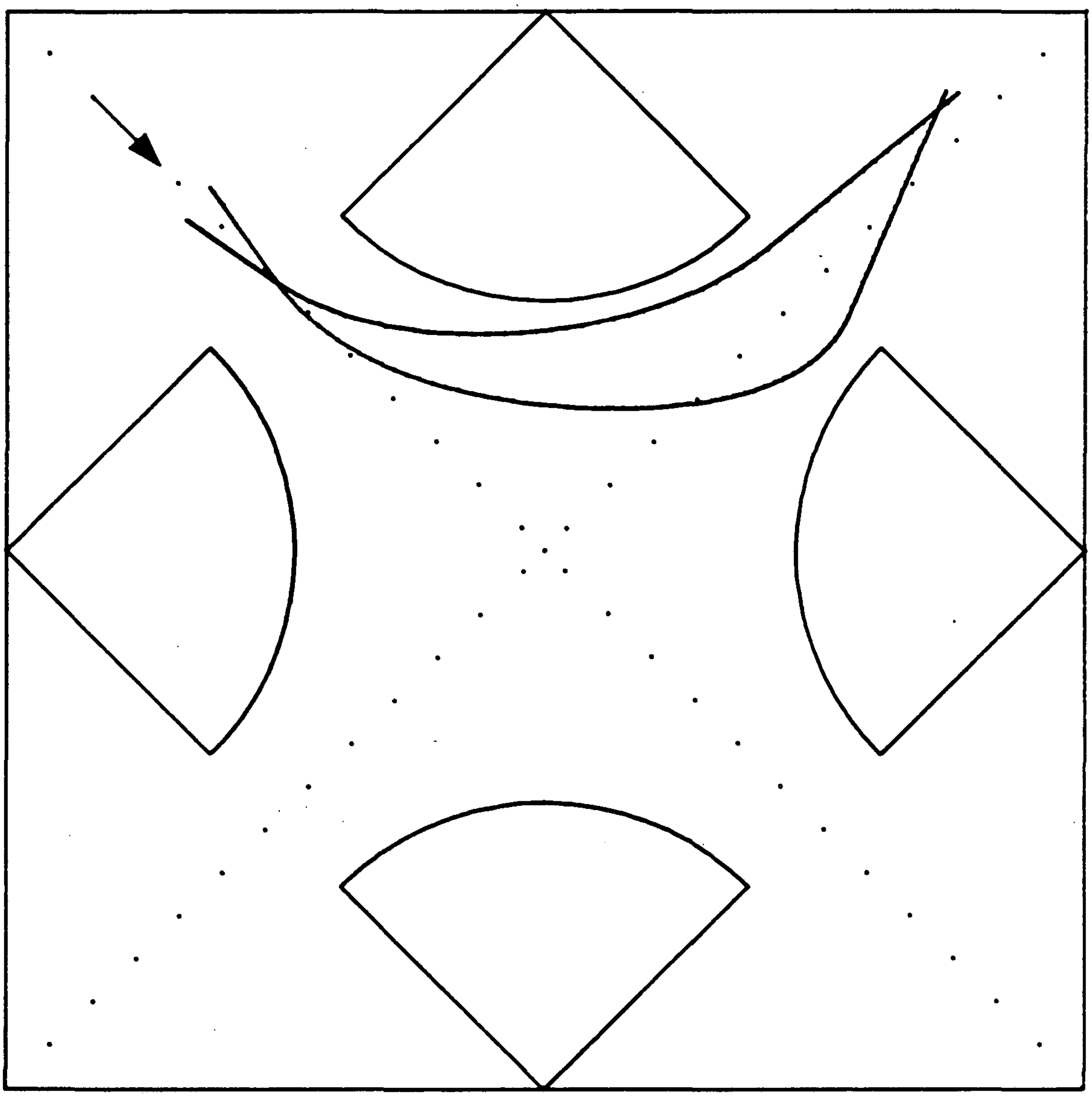
XBL 8310-12191

Figure 17



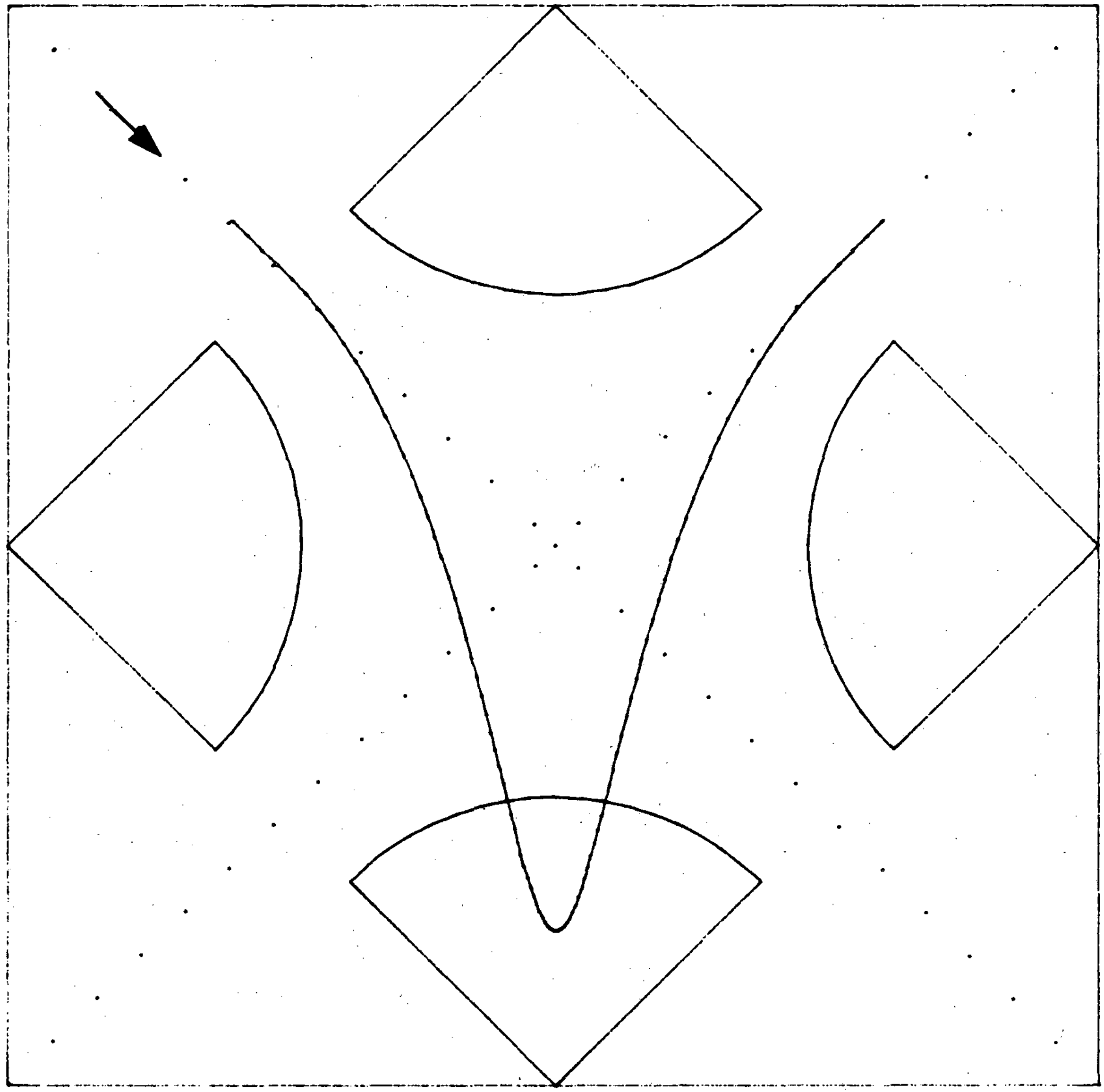
XBL 8310-12192

Figure 18



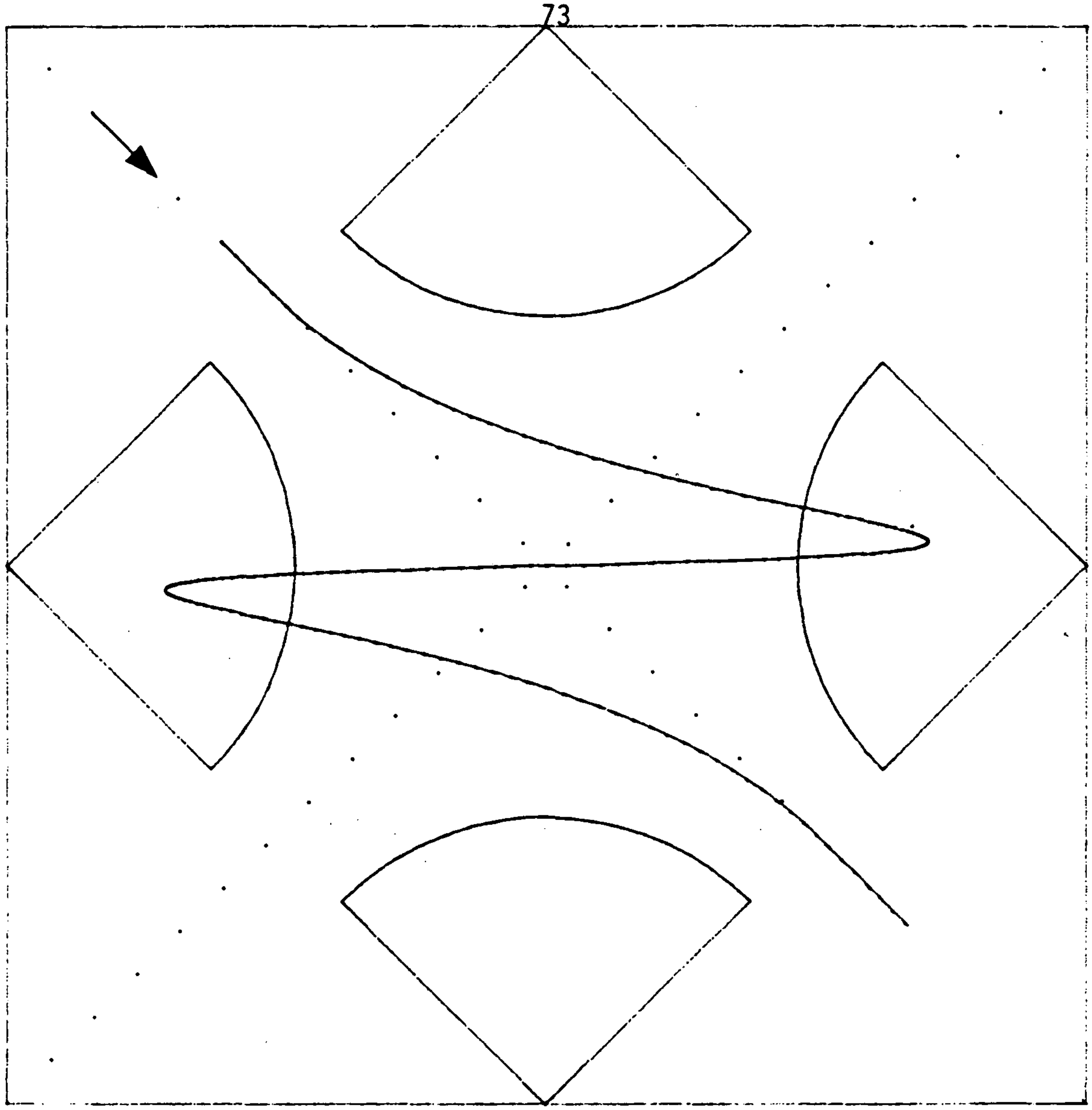
XBL 8310-12193

Figure 19



XBL 8310-12194

Figure 20



XBL 8310-12195

Figure 21

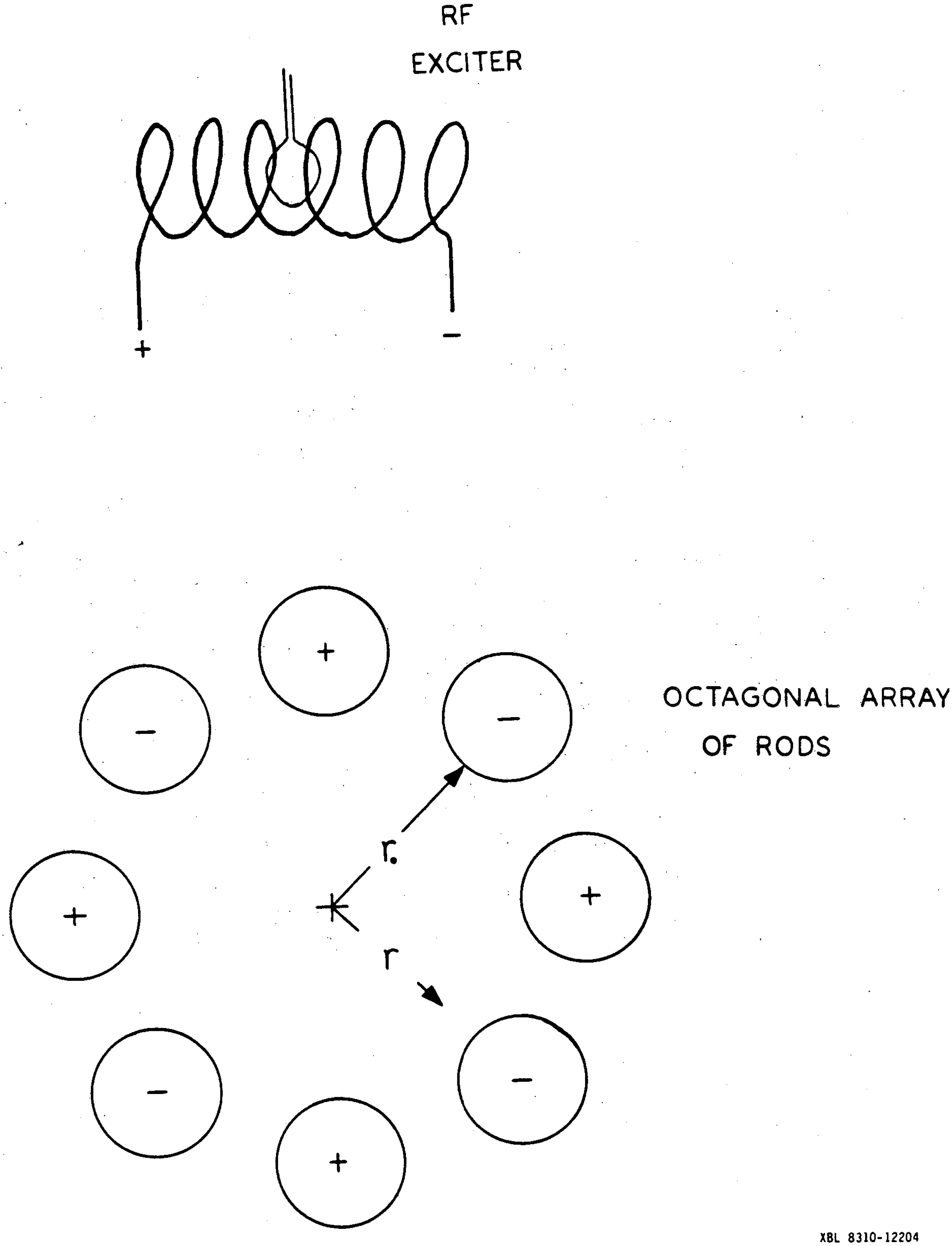
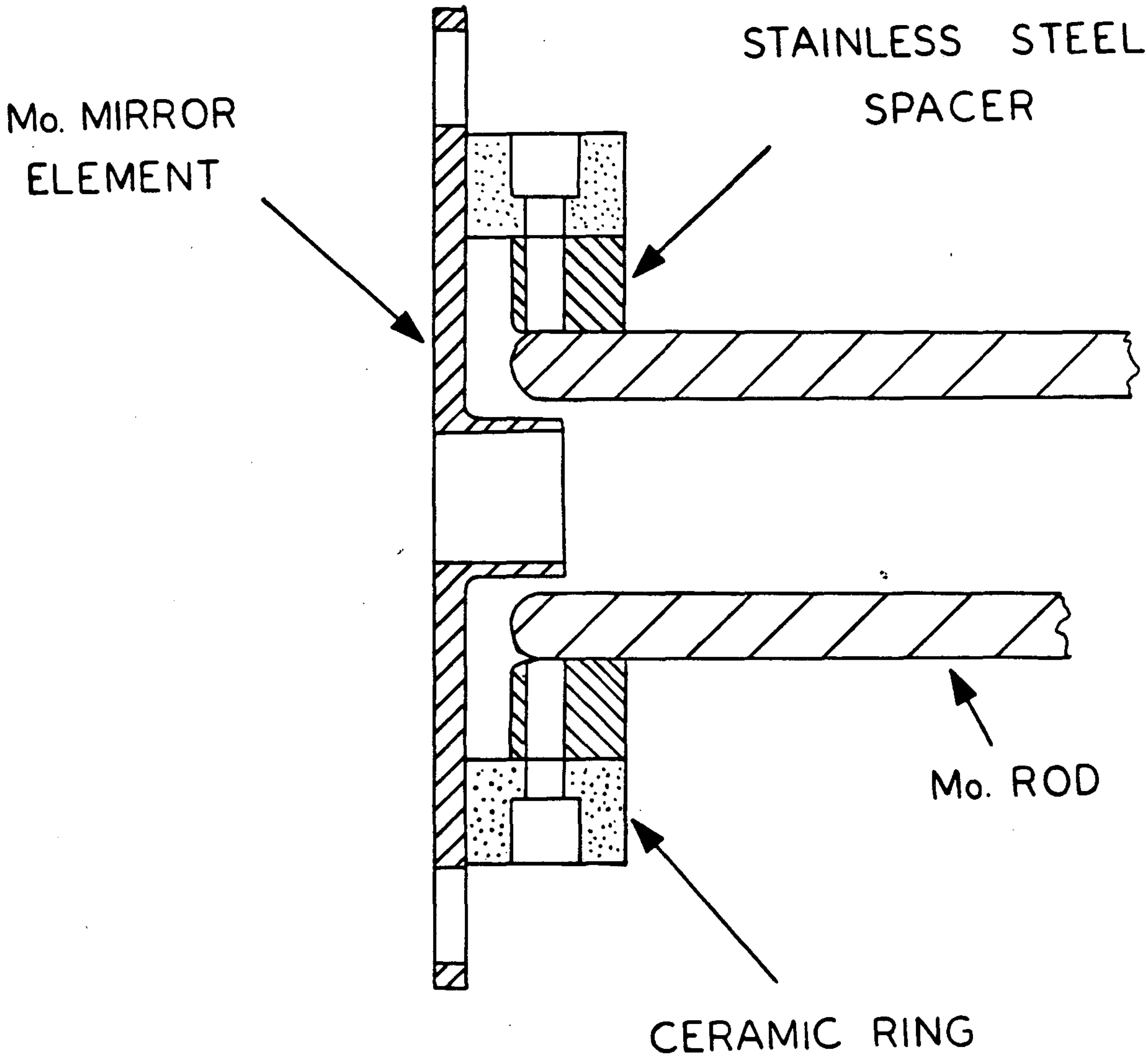
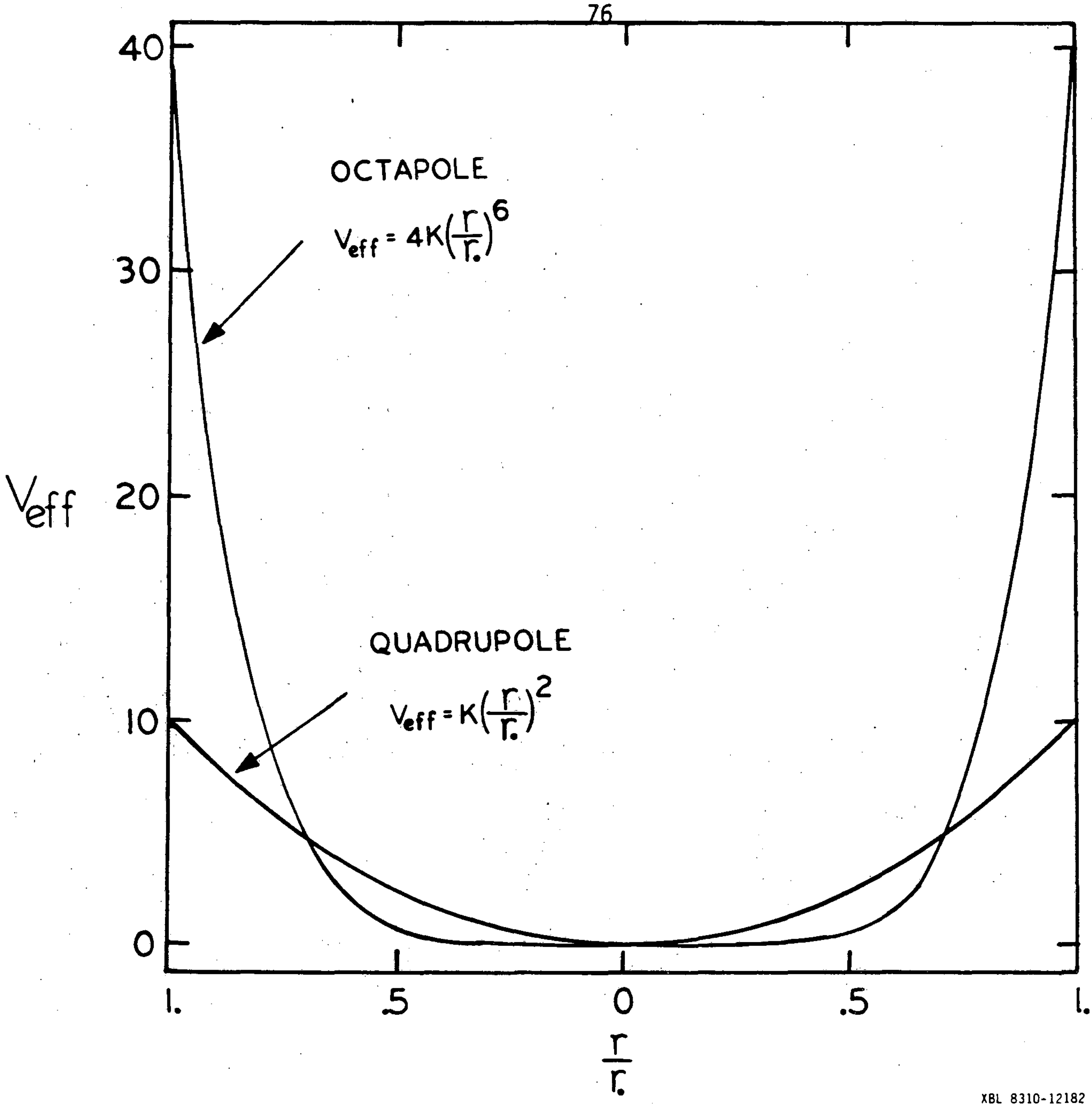


Figure 22



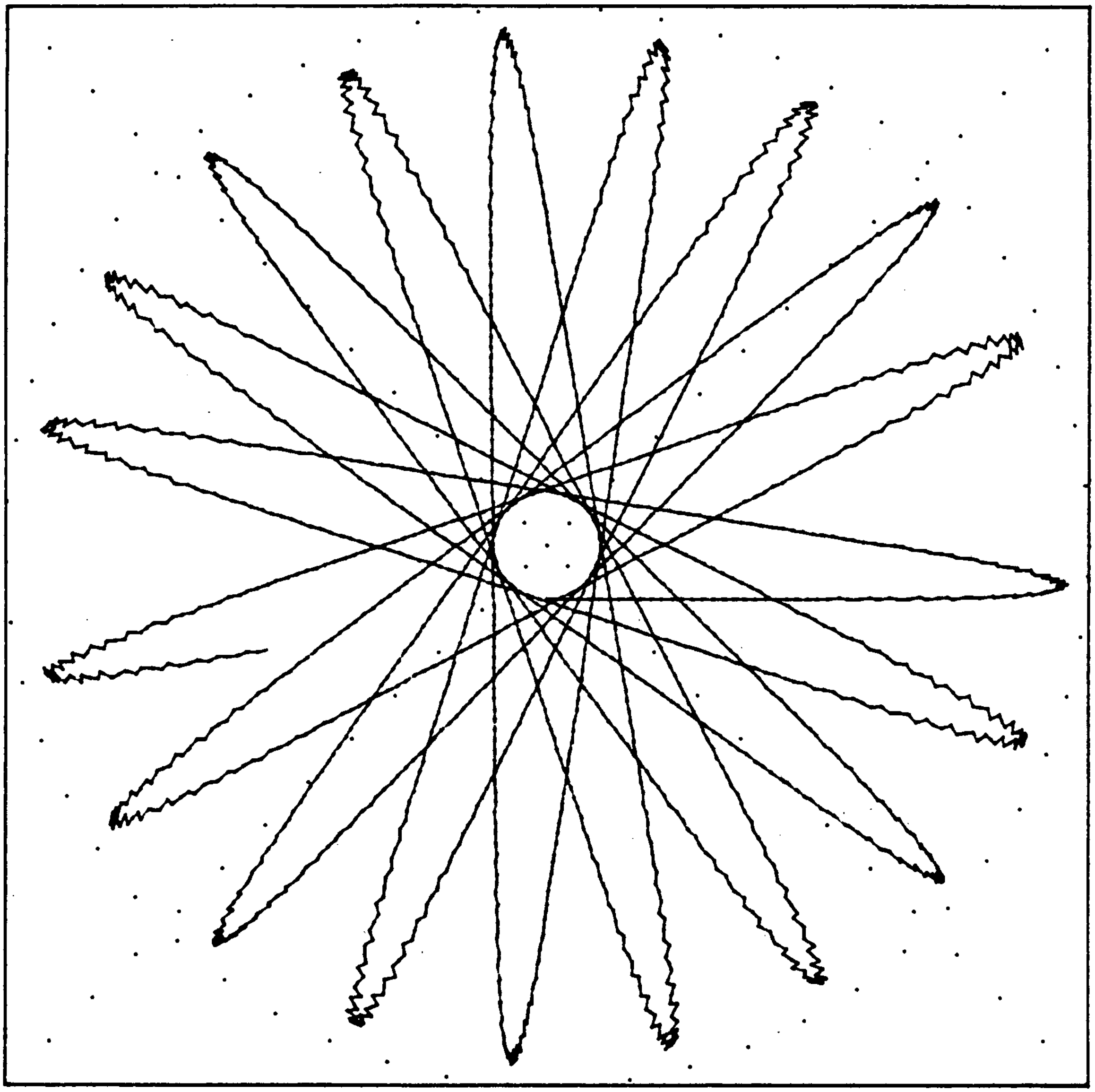
XBL 8310-12183

Figure 23



40
30
20
10
0
V_{eff}
OCTAPOLE
$V_{eff} = 4K\left(\frac{r}{r_\circ}\right)^6$
QUADRUPOLE
$V_{eff} = K\left(\frac{r}{r_\circ}\right)^2$
1.
.5
0
.5
1.
$\frac{r}{r_\circ}$
XBL 8310-12182

Figure 24



XBL 8310-12181

Figure 25

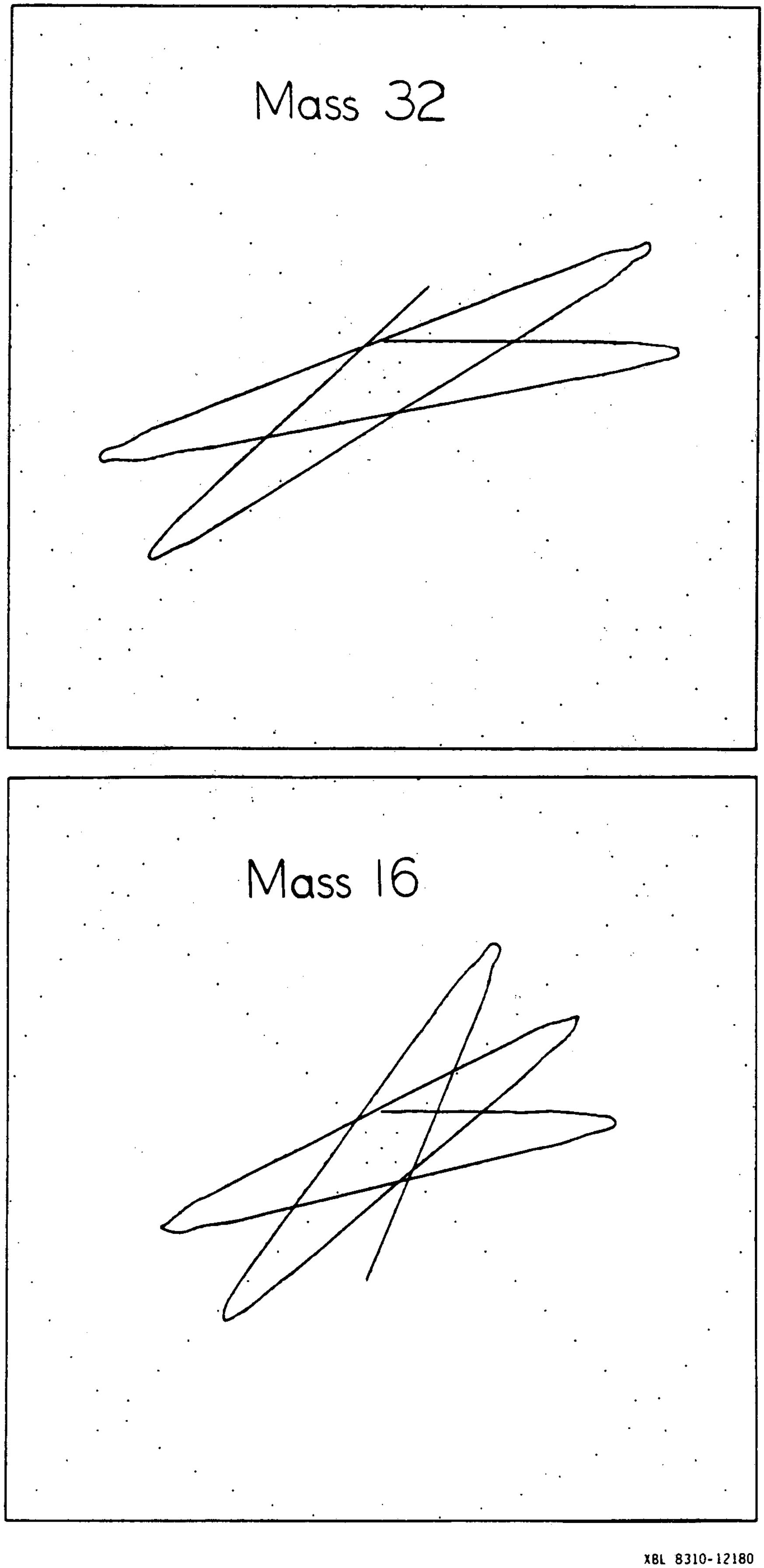


Figure 26

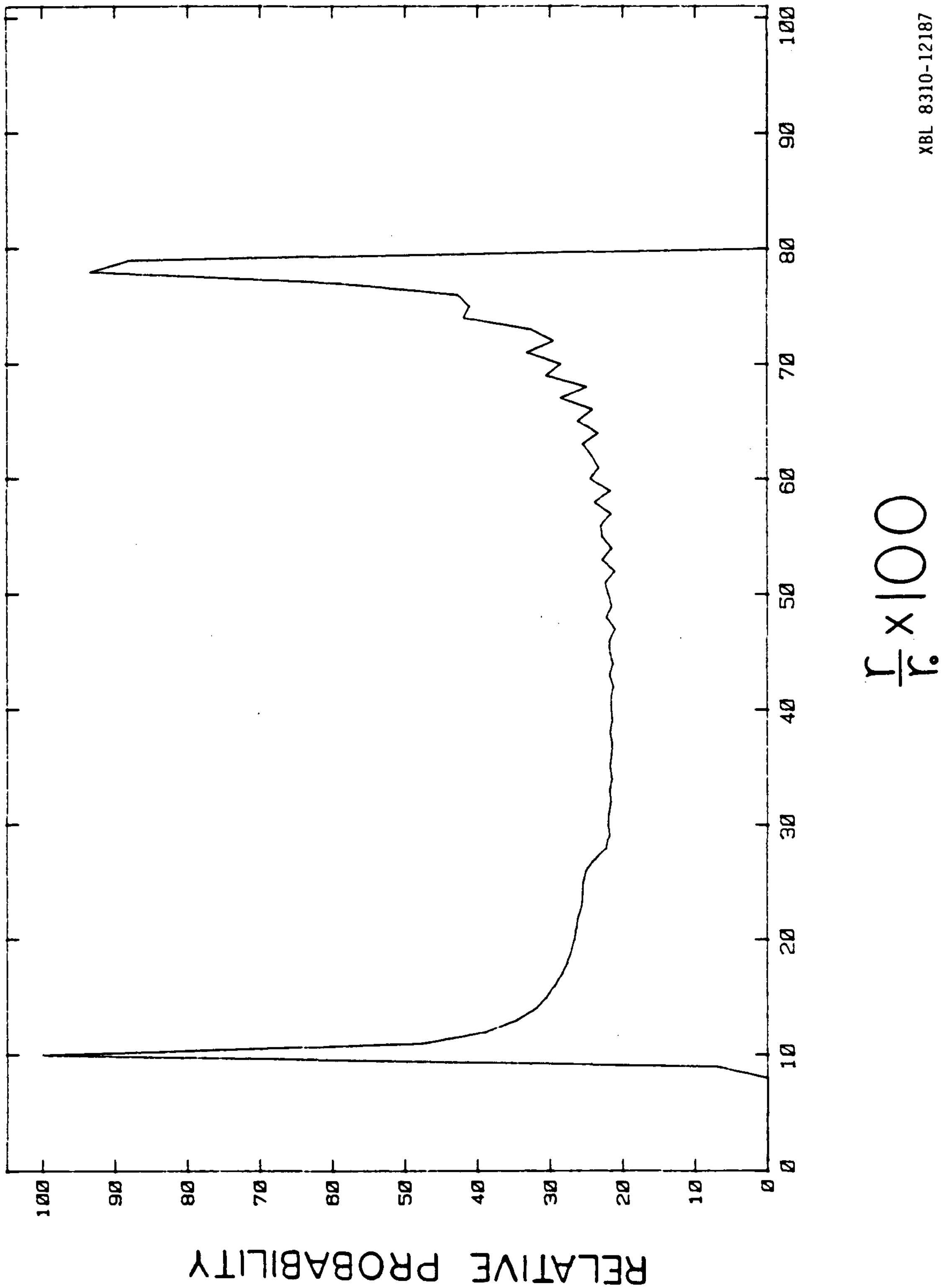
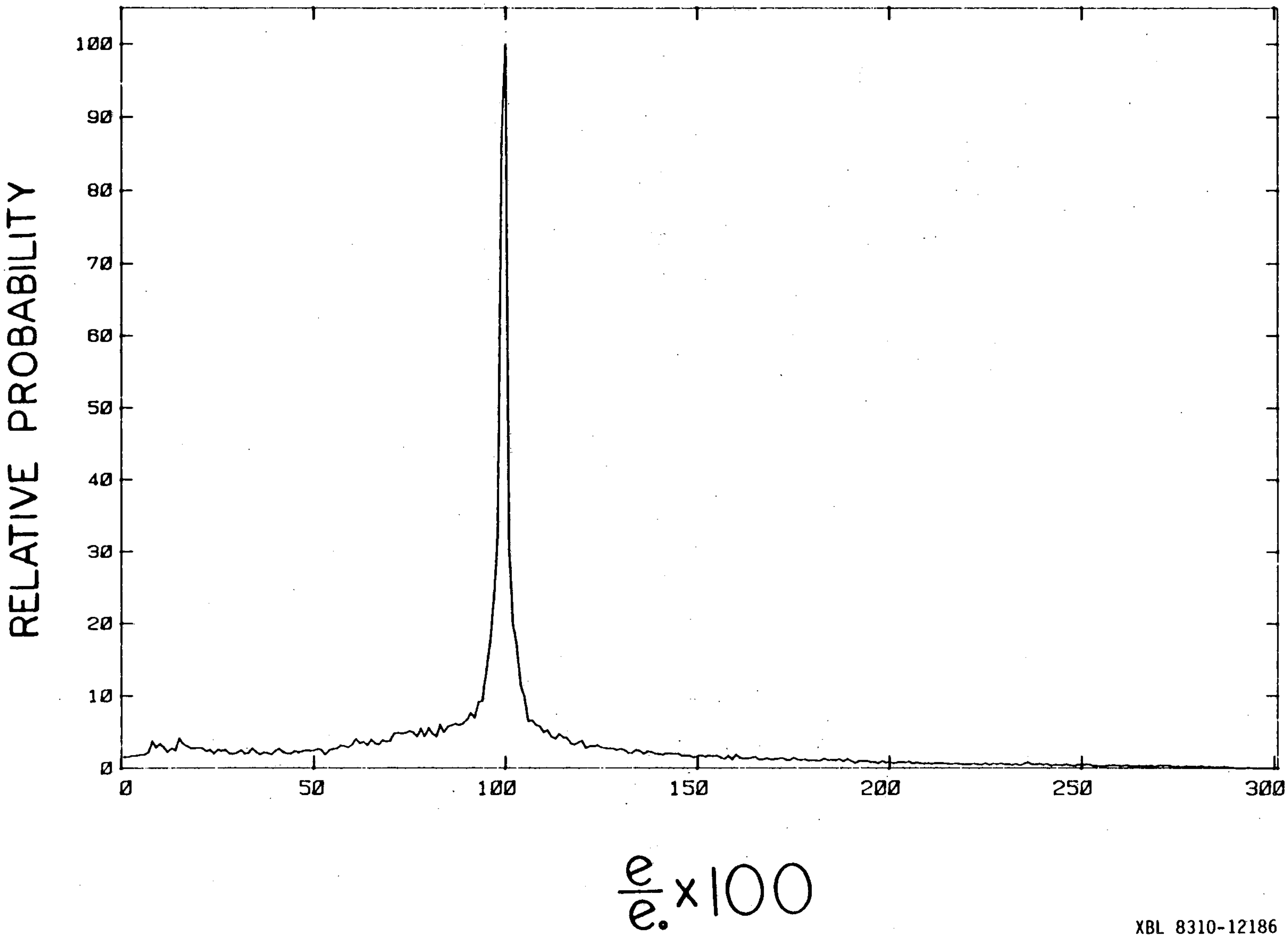
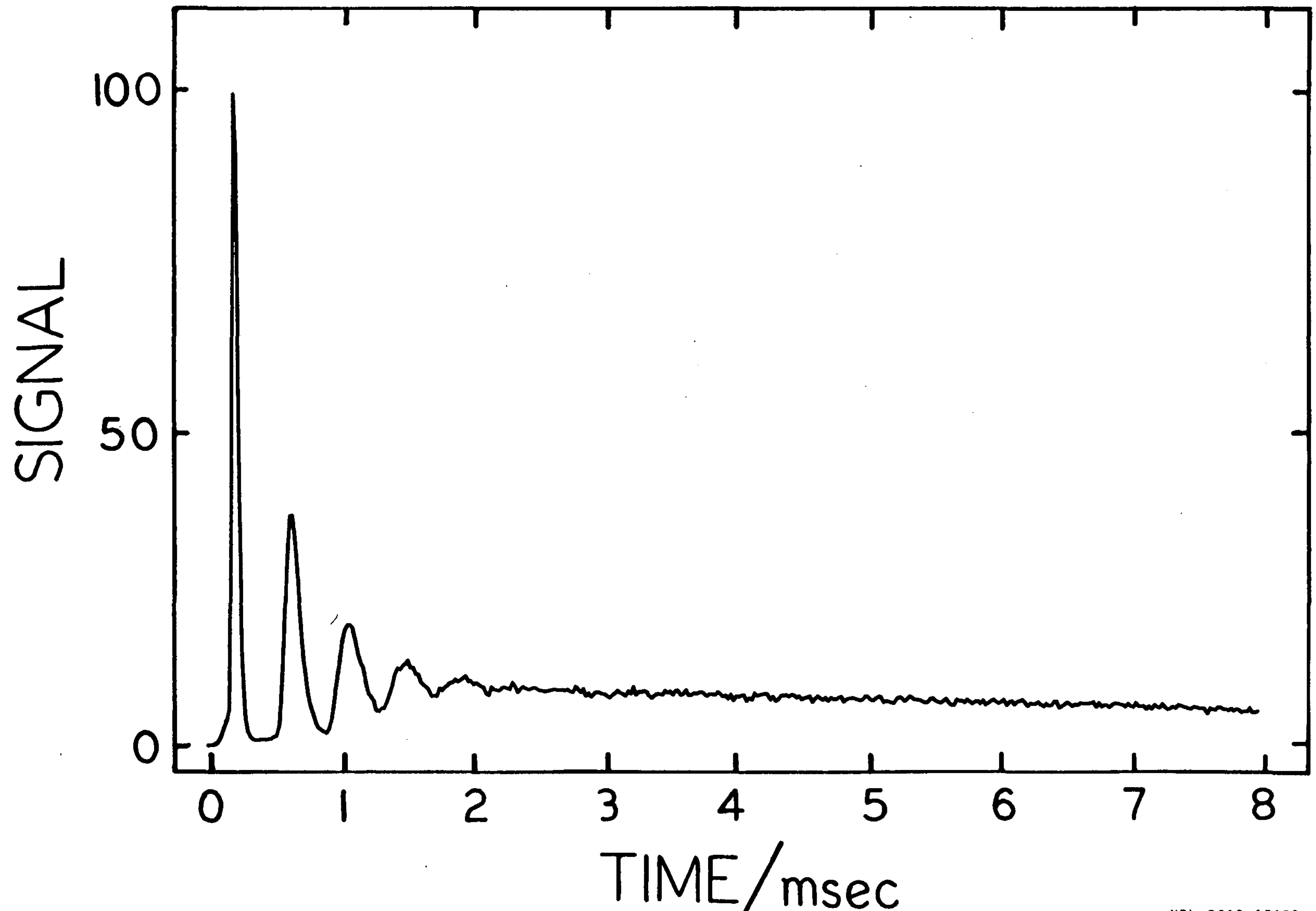

RELATIVE PROBABILITY
0
10
20
30
40
50
60
70
80
90
100
$\frac{r}{r_0} \times 100$
XBL 8310-12187

Figure 27



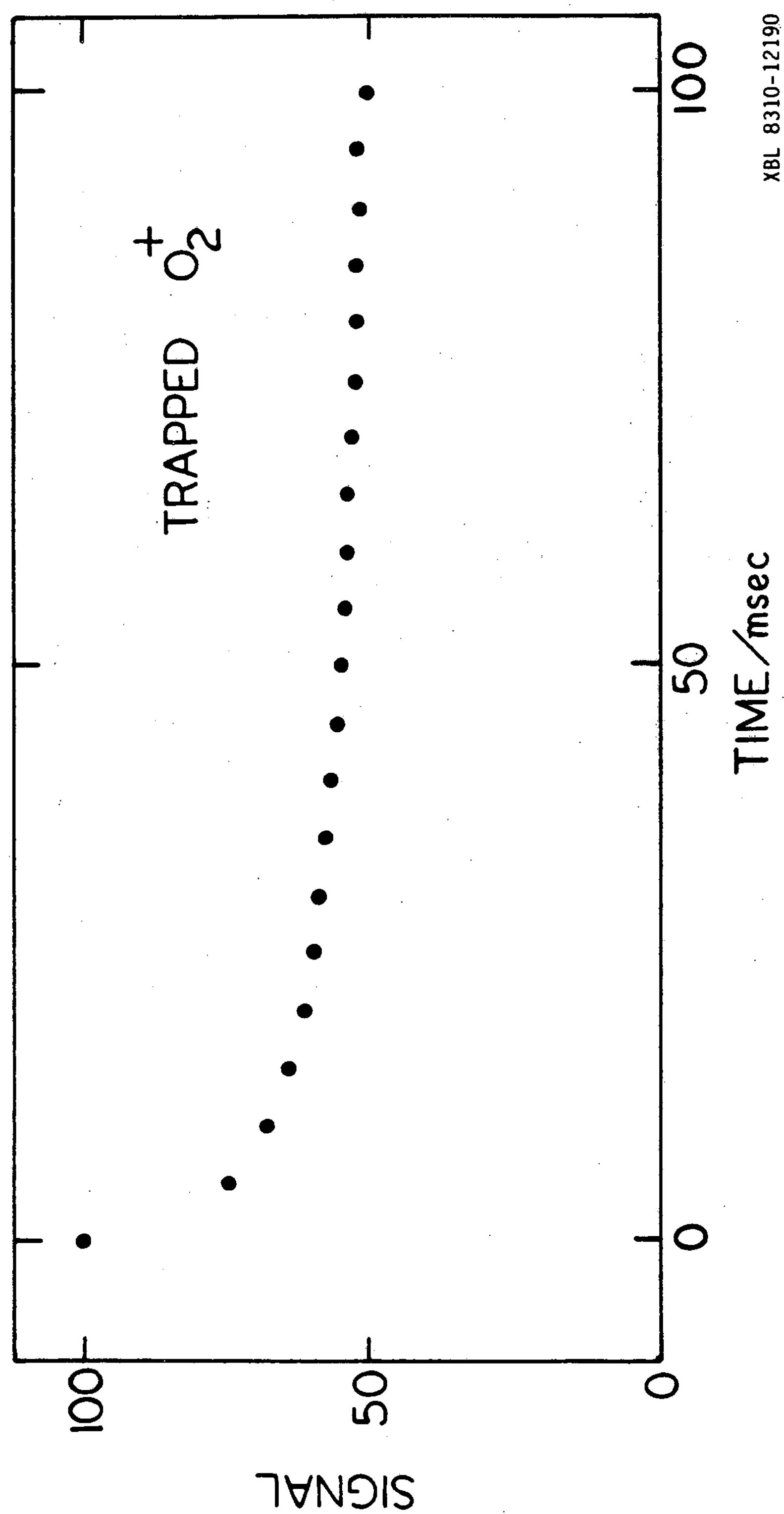
XBL 8310-12186

Figure 28



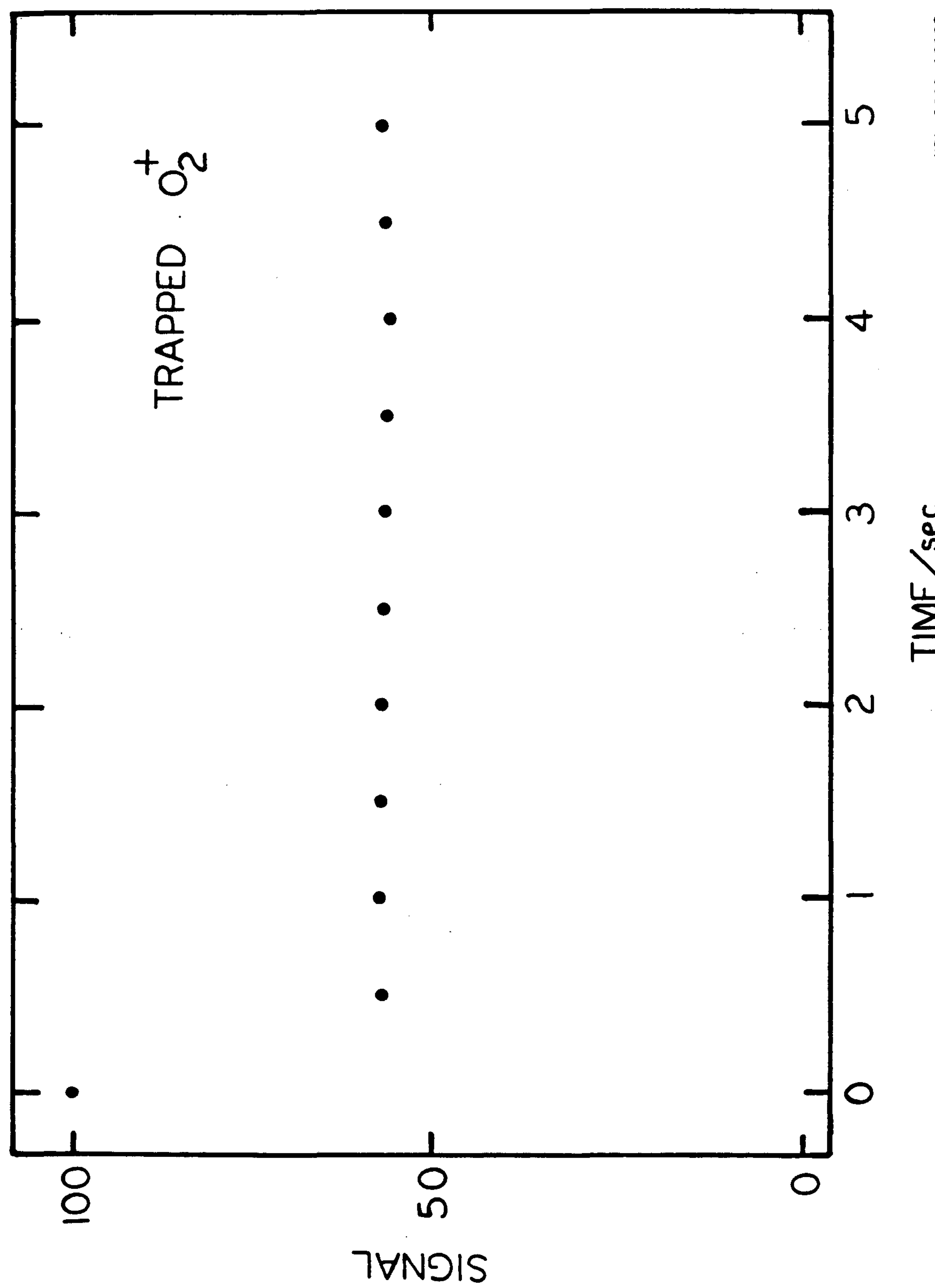
XBL 8310-12188

Figure 29



XBL 8310-12190

Figure 30



XBL 8310-12189

Figure 31

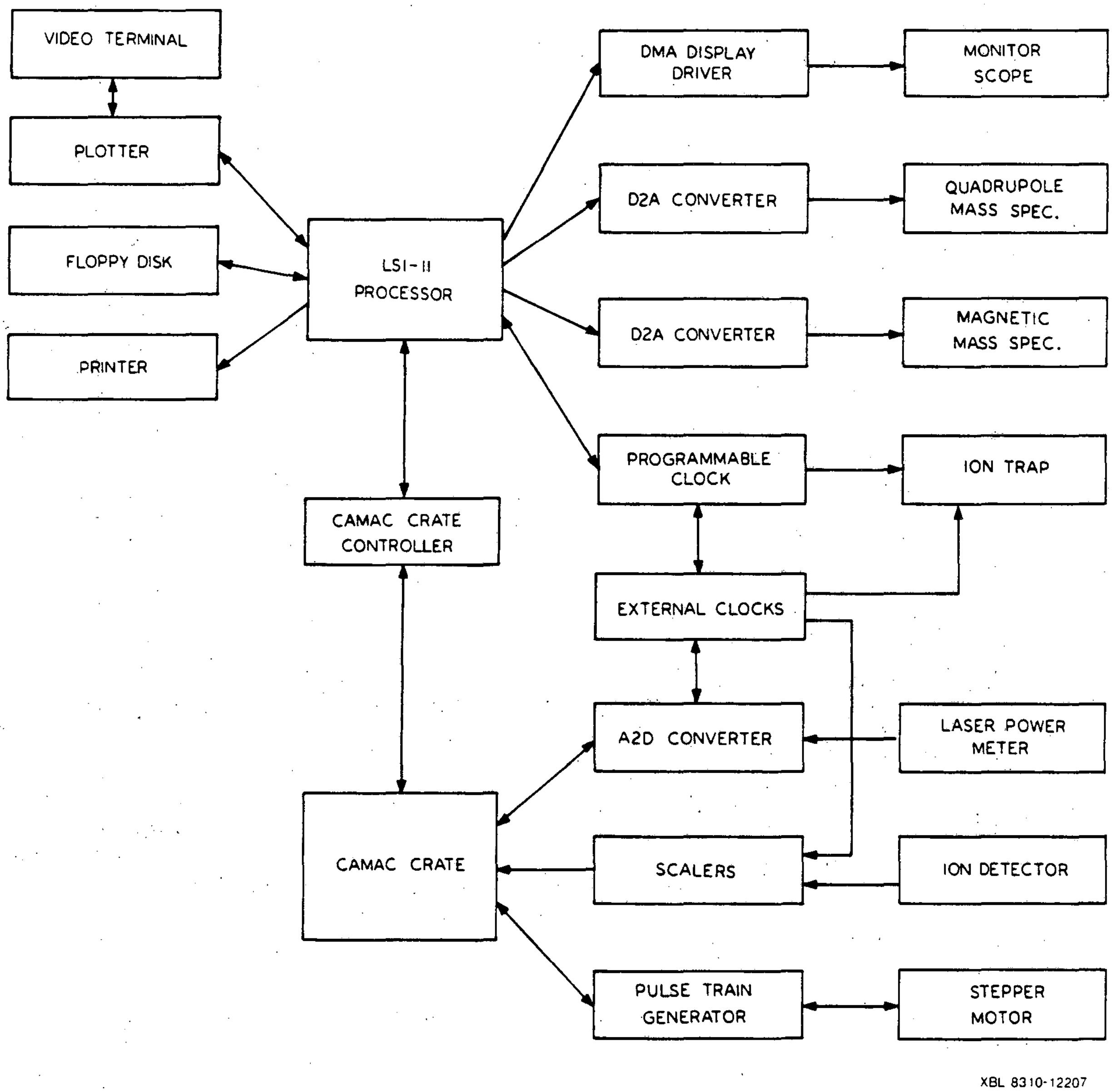
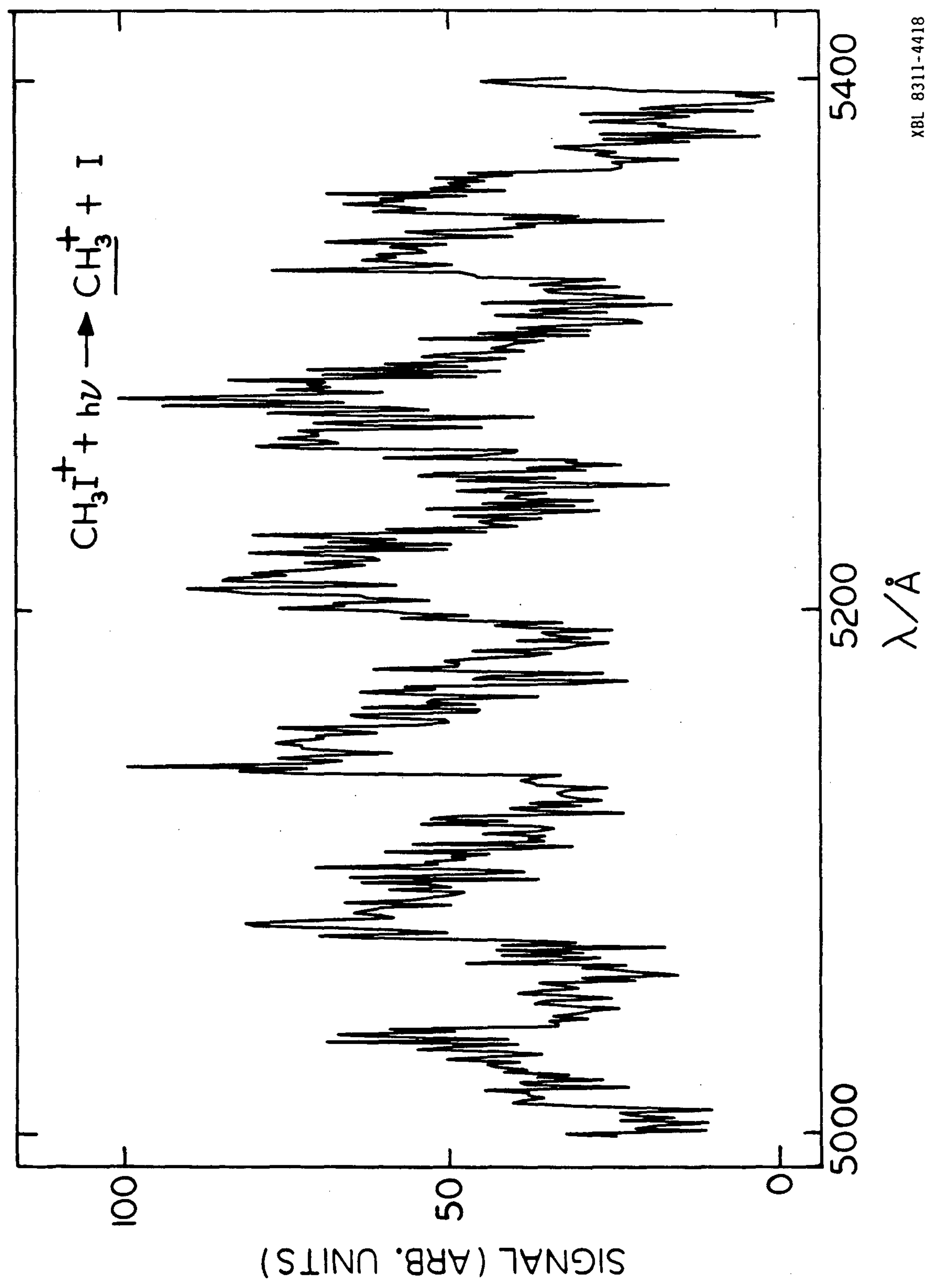
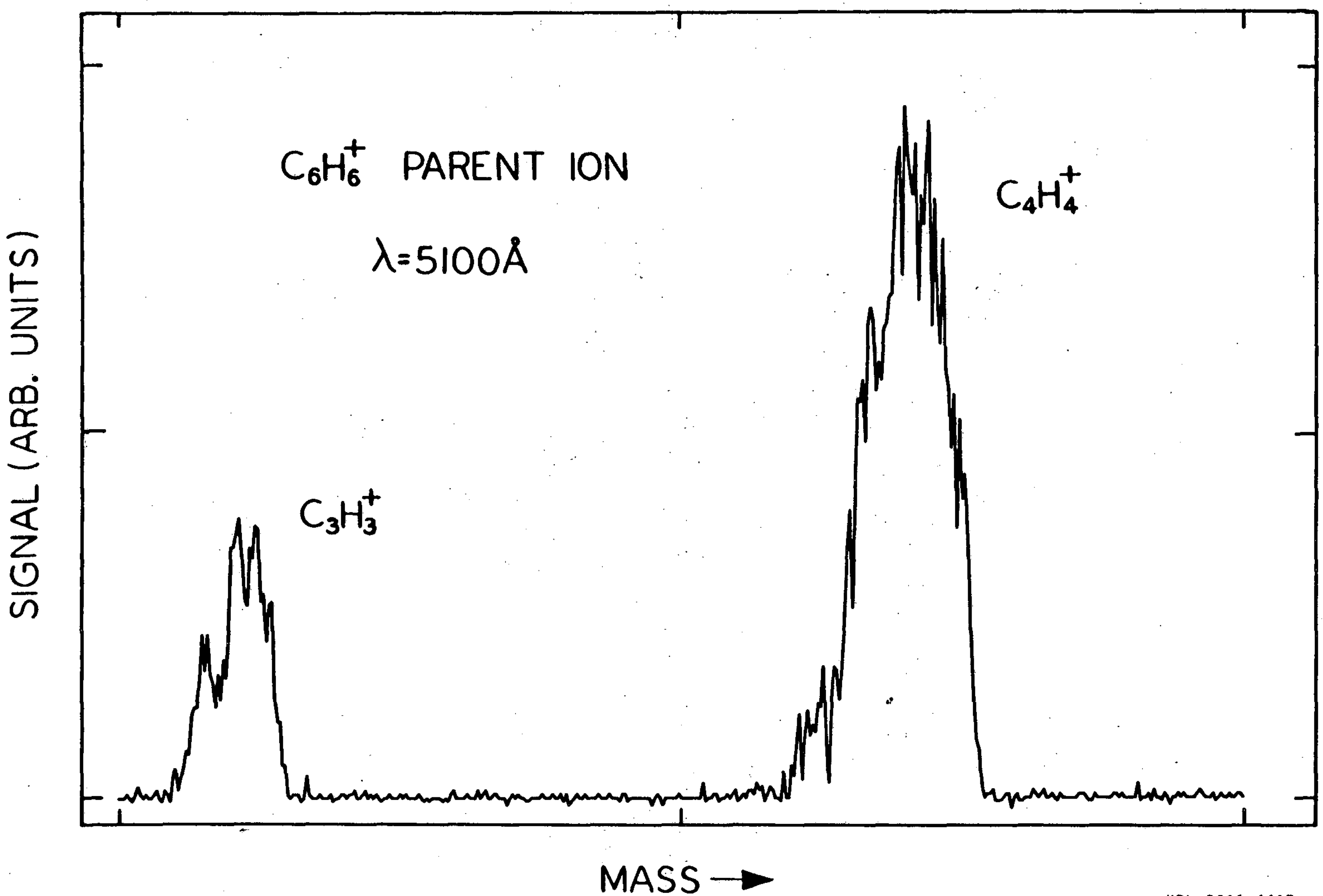

VIDEO TERMINAL
PLOTTER
FLOPPY DISK
PRINTER
LSI-11 PROCESSOR
CAMAC CRATE CONTROLLER
CAMAC CRATE
DMA DISPLAY DRIVER
MONITOR SCOPE
D2A CONVERTER
QUADRUPOLE MASS SPEC.
D2A CONVERTER
MAGNETIC MASS SPEC.
PROGRAMMABLE CLOCK
ION TRAP
EXTERNAL CLOCKS
A2D CONVERTER
LASER POWER METER
SCALERS
ION DETECTOR
PULSE TRAIN GENERATOR
STEPPER MOTOR
XBL 8310-12207

Figure 32



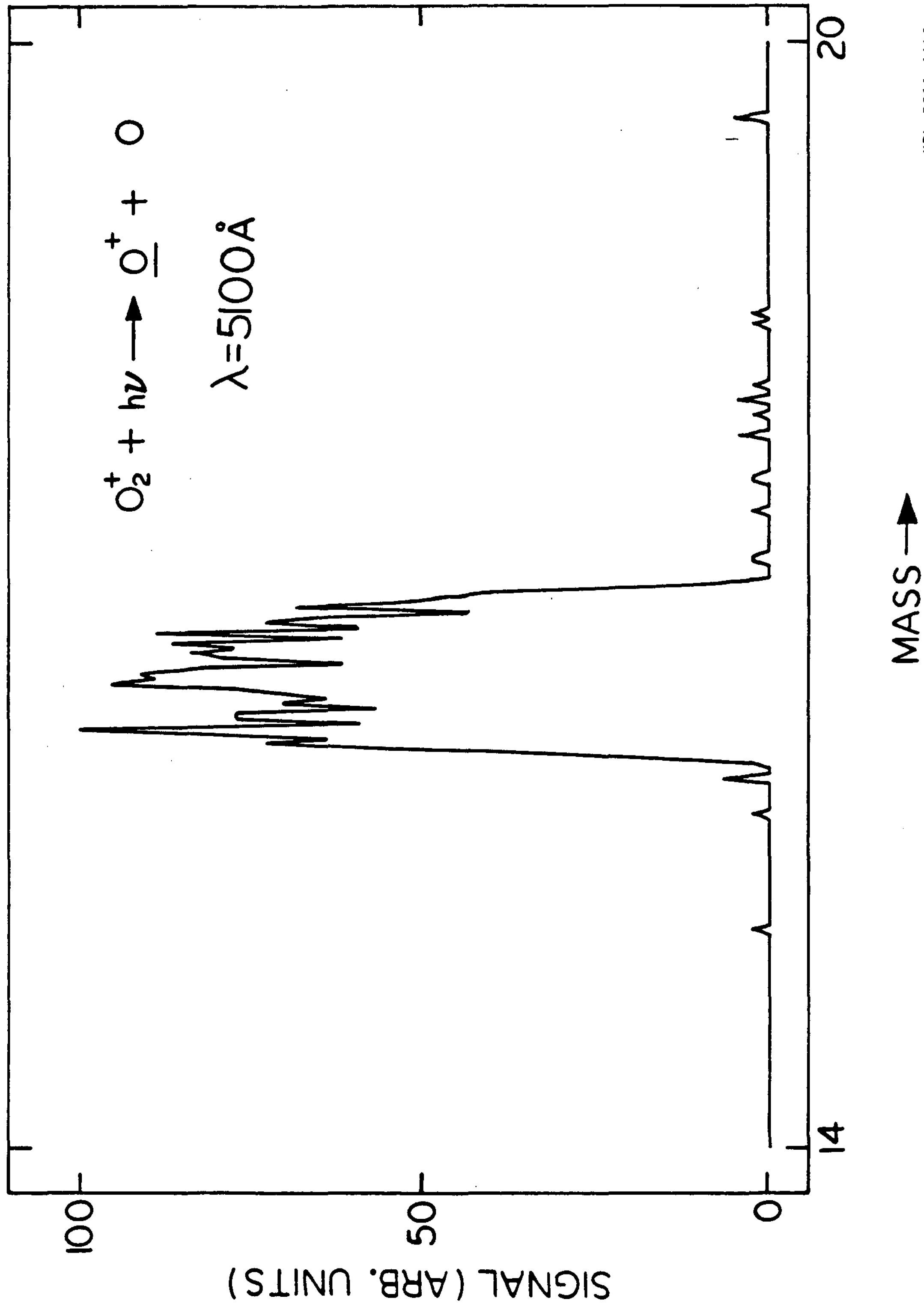
XBL 8311-4418

Figure 33



XBL 8311-4417

Figure 34



XBL 8311-4416

Figure 35

III. PHOTOFRAGMENTATION OF O_2^+: THE LIFETIME OF THE a $^4\pi_u$ METASTABLE STATE

A. Introduction

Molecular oxygen, an abundant and reactive species, is involved in many chemically important environments. Of interest is the fact that many of these environments are harsh enough to create the O_2^+ ion. In the gas phase one finds O_2^+ in flames (combustion), plasmas, electrical discharges, and by solar ionization, in the upper atmosphere. The ambition to better understand and predict the chemical evolution of these systems leads to the desire to know as much as possible about this important species.

Because O_2^+ has been the subject of much experimental and theoretical work, it is one of the most understood molecules. Its spectroscopy, in particular the $b^4\Sigma_g^- \longrightarrow a^4\pi_u$ First Negative band, has been studied for many years by emission techniques. The high resolution emission work of this band by Nevin[1] in 1938 provided high quality information on the $b^4\Sigma_g^-$ and $a^4\pi_u$ states. Figure 1 shows the pertinent potential curves for this discussion.

More recently Vestal et al.[2] using 6000Å radiation observed photodissociation of O_2^+ $a^4\pi_u$ created by electron impact ionization of O_2. This led to much work on the ion using laser photofragment spectroscopy by observing the predissociation of O_2^+ $b^4\Sigma_g^-$ state resulting from transitions shown in Fig. 1 as type I.[3] In this manner the $b^4\Sigma_g^- \longleftarrow {}^4\pi_u$ band was studied at very high resolution. In addition, the $f^4\pi_g \longleftarrow a^4\pi_u$ system was studied using photofragment spectroscopy--this resulted in the only reported case of a $^4\pi \leftarrow ^4\pi$ system.[2g,2j]

Numerous chemical studies have been done on O_2^+. Here great advantage may be taken of the wealth of information available on the potential energy surfaces for this ion. The reaction of O_2^+ to yield O_3^+ first gave evidence that O_2^+ was formed in a metastable state by electron impact ionization.[4] The metastable state responsible for the reaction and the photodissociation by visible light was identified as the $a^4\pi_u$ state: Much of the reactivity of the O_2^+ ion is due to the $a^4\pi_u$ metastable state.[5] Thus it is clear that the lifetime of this state is an important parameter in modelling studies. For example, A. Dalgarno et al.[6] state that the composition of the ionosphere in the E- and F-regions may depend significantly on the O_2^+ $a^4\pi_u$ state. In order to evaluate how strongly the $a^4\pi_u$ state will influence its chemical environment, its lifetime must be known. However, in many studies, both chemical and spectroscopic, assumptions have had to be made concerning the lifetime of the O_2^+ $a^4\pi_u$ metastable state, since a direct measurement was lacking. The lifetime has been generally assumed to be $\tau \geq 1$ msec[7] since its radiative relaxation to the $X^2\pi_g$ state is spin forbidden.

Our spectrometer is a low resolution instrument, being Doppler limited. The study of the $b^4\Sigma_g^- \leftarrow a^4\pi_u$ predissociation spectrum was not of interest to us for the sake of the spectroscopy alone since it is already well known. However, because our spectrometer could generate the photofragment spectrum as a function of time, where the time could be varied from ~1 millisecond to 100's of milliseconds, we could readily measure the lifetime of the $a^4\pi_u$ state—a measurement worth making. We recorded the photofragment spectrum of O_2^+, generated by electron impact ionization of O_2, at low resolution (~1 cm^{-1}) as a function of time. This led us to discover that the decay of the $a^4\pi_u$ state is strongly dependent on its spin component identity.

B. Experimental

The O_2^+ was produced by electron impact ionization of O_2 at 5×10^{-5} torr. The electron energy was ~80 eV. This ionization produces, after a few microseconds, O_2^+ in essentially two electronic states: ~30% in the $a^4\pi_u$ metastable state and ~70% in the $X^2\pi_g$ ground state.[4c,8] The ions thus produced were passed through a sector magnet mass filter which allowed only O_2^+ to be placed into the RF octapole interaction region. This region was maintained at about 7×10^{-10} torr during the photofragmentation measurements.

The experiment was run in a pulsed mode. Ions were pulsed into the RF octapole trap. After a selected length of time the laser was fired, and then the product ions were sent through the final quadrupole mass filter to be counted by the ion detection system. The counts from this laser sequence were placed into a signal-plus-background (S + B) buffer. This laser sequence was followed by an identical pulse sequence, except that the laser was not fired. The counts obtained from this no-laser sequence were placed into a "background" (B) buffer. The laser correlated signal was then taken to be (S + B) − B. Figure 2 shows the details of the pulse sequence used in this data collection cycle. The RF dumping pulse was used to pulse the RF supplied to the octapole rods off for a short period of time, thus ensuring that the ion trap was free of ions at the start of a new pulse sequence. The data were collected in two modes. In the first mode delay-x (D_x), the amount of time the ions remained in the trap before the laser was fired, was held constant while the laser frequency was varied: thus signal versus laser frequency was measured for a given ion storage time. In the second mode, the laser frequency was held constant while delay-x was varied: photofragment yield as a function of time for a given laser frequency was thus measured. In each mode the O^+ photofragment was detected. Figure 3 shows typical results for a wavelength dependence scan and Figure 4 shows the O^+ yield as a function of time for the indicated wavelengths.

The linewidth of the pulsed dye laser was about 0.4 cm^{-1} and its pulse width was ~8 nsec. The lifetime measurements were carried out using ~0.003 watts of time averaged laser power. The repetition rate was 4.2 Hz and the beam diameter 0.4 cm. The laser beam was carefully aligned so that it passed through the spectrometer without striking any metal surfaces: no laser correlated signal was observed when no O_2^+ molecules were in the interaction region. Figure 5 shows the O^+ photofragment signal as a function of laser pulse energy. The lifetime measurements were carried out using pulse energies of ~0.7 mj.

The best resolution observed in the wavelength scans was ~1 cm^{-1}. The average energy of the trapped ions was ~0.65 eV with a spread of about 0.3 eV. As discussed in the previous chapter, the initial axial energy of the ions is distributed to the other coordinates producing wide kinetic energy distributions in each coordinate. For O_2^+ at ~0.65 eV median energy, one might expect a Doppler width of ~0.3 cm^{-1} at 5000Å. Our laser had a line width of ~0.4 cm^{-1}. Thus, the observed resolution of ~1 cm^{-1} is in keeping with expectation.

Before continuing, some elaboration is necessary concerning experimental difficulties encountered during the course of the measurements. The first problem, evident in Fig. 4, was the escape of parent ion O_2^+ from the ion trap (see Chap. II). This problem was not very serious since one could easily account for the loss of parent ions when treating the data. Also, as mentioned before, the losses were minimal after about 50 msec of trapping time.

There was however, a more bothersome consequence of the escaping O_2^+ parent ions. Some of these ions were being accelerated into metal surfaces during their "escape", and the resulting ion-surface collision fragmented some of the O_2^+ creating O^+. A portion of these collisionally produced O^+ ions recoiled back into the ion trap region and were contained by the trap. This interaction was taking place at the ends of the ion guide with the ion mirrors. This was demonstrated by measuring the O_2^+ loss and O^+ appearance as a function of time while a pulse of ions was bouncing back and forth in the ion guide. Figure 6 shows the abundance of O_2^+ and O^+ in the ion guide as a function of time. The times at which the traveling pulse of ions encounters a mirror are indicated by the arrows. (The vertical scale for each ion is different.) Note that the encounter causes a loss of O_2^+ and at the same time results in the increase of O^+ ions within the ion guide. This increase of O^+ was unexpected and produced signal-to-noise ratios which were much worse than anticipated, especially for long trapping times. Figure 7 shows how this O^+ background increased with time.

C. Results and Discussion

Figure 8 shows the results of scanning the laser wavelength from 4902 to 5025Å in 0.48Å steps with ion trapping times held at 0.1 msec and 100 msec. This wavelength region covers the left hand peak in Fig. 3. Figure 9 shows spectral simulations of the same wavelength region with the time equal to 0.1 msec and 100 msec. The spectra were calculated assuming that the F_2'' and the F_3'' spin components of the $a^4\pi_u$ state decay with a time constant of 6 msec, while the F_1'' and F_4'' components have a 129 msec time constant.

At this point I will briefly describe the two states involved in these photofragment spectra. The upper $b^4\Sigma_g^-$ state, being a state, is a Hund's case (b) state. Each rotational level is split into four components. These four components are labeled as follows:

F_1' , $J' = N' + 3/2$

F_2' , $J' = N' + 1/2$

F_3' , $J' = N' - 1/2$

F_4' , $J' = N' - 3/2$

where J' is the total angular momentum quantum number and N' is the quantum number for the nuclear angular momentum apart from spin.

The lower $a^4\pi_u$ state is split into four sublevels by the spin-orbit interaction: the four levels are designated as above (e.g., F_1'', $J'' = N'' + 3/2$). The spin-orbit interaction constant for this state is negative,[9] thus one has an inverted energy ordering of the F" components. Using the $^4\pi_\Omega$ designation[10] for pure Hund's case (a) coupling, where $\Omega = \Lambda + \Sigma$ and Λ and Σ are the projections of the orbital and spin angular momenta respectively, one associates F_1'' with $^4\pi_{5/2}$, F_2'' with $^4\pi_{3/2}$, F_3'' with $^4\pi_{1/2}$, and F_4'' with the $^4\pi_{-1/2}$ component. The order written is increasing in energy.

The O_2^+ $a^4\pi_u$ state is characterized as Hund's case(a) for low J" values. But as the nuclear rotational velocity increases with increasing J", the electron spin uncouples from the molecular axis until the state approaches a Hund's case (b) description at high J" values. For most J" values an intermediate description must be used to correctly characterize the levels. Figure 10 shows the quartet splitting in the $a^4\pi_u$ state for sets of J" values. The first set has J" = 3-1/2, 4-1/2, 5-1/2, 6-1/2, and 7-1/2. The figure is drawn with the lowest level of the set equal to zero and the largest value equal to 100. The second set of values plotted has each J" value of the first set increased by 10, and so forth. Four such sets of J" values are plotted. Note that for the low J" value set, the grouping is according to Ω value (Hund's case (a) grouping), while the set with the highest J" values is tending toward K value grouping (Hund's case (b)), where K is the resultant angular momentum quantum number formed from Λ" and N". Note that this figure indicates that an intermediate Hund's case description is certainly necessary from J" = 16-1/2 to J" = 26-1/2.

The spin component dependence of the lifetime for the $a^4\pi_u$ state of O_2^+ is similar to that found in the Cameron system of CO,(11) where the dependence arises from different spin-orbit mixing of the $A'\pi$ and $a^3\pi$ states. The $A^2\pi_u$ state of O_2^+ is just above the $a^4\pi_u$ state. If this $A^2\pi_u$ state mixes with the $a^4\pi_u$ metastable state via the spin-orbit interaction, then the $a^4\pi_u \longrightarrow X^2\pi_g$ transition will have some allowed character.

Kovacs(12) has shown how the spin-orbit mixing of a $^4\pi_u$ state and a $^2\pi_u$ state depends on the Hund's coupling case used to describe the states. The $A^2\pi_u$ state of O_2^+ is found to be a Hund's case (b) state.(13) As discussed above, the $a^4\pi_u$ state is characterized as Hund's case (a) for low J", but transforms into a Hund's case (b) for high J" values.

According to Kovacs, if the $a^4\pi_u$ state is Hund's case (a) and the $A^2\pi_u$ state is case (b), then only the $\Omega" = 3/2$ and 1/2 spin components of the $a^4\pi_u$ state, that is F_2'' and F_3'', will mix well with the $A^2\pi_u$ state. Thus for low J" values, the F_2'' and F_3'' components have $^2\pi_u$ character and are able to relax faster to the $X^2\pi_g$ state. For rotational levels with J" great enough, the $a^4\pi_u$ state is Hund's case (b). In this case all the Σ'' components of the $a^4\pi_u$ state couple with the $A^2\pi_u$ state, although the Σ'' = 1/2 and -1/2 couple more strongly. Therefore in the high J" limit all the spin components gain intensity through spin-orbit mixing with the $A^2\pi_u$ state. Kovacs' treatment suffices in giving a qualitative understanding of the origins of the spin component dependence of the lifetimes.

The $\Delta v = 3$ band of the $b^4\Sigma_g^- \longleftarrow a^4\pi_u$ system is fortuitously placed so that the F_1'' and F_4'' components are fairly easily distinguished from the F_2'' and F_3'' components. This is seen in Fig. 3 where a hole is seen in the middle of the $\Delta v = 3$ peak at t = 100 msec because of the absence of the F_2'' and F_3'' spin states. By using the spectra of the $\Delta v = 3$ band taken at delayed times, two wavelengths were chosen at which to perform the lifetime measurements: 4947.0Å, which contains mostly F_1'' intensity and 4967.2Å which has mostly F_2'' and F_3'' intensity. The O^+ photofragment signal, which is proportional to the amount of $a^4\pi_u$ accessed at the particular wavelength, versus time (corrected for parent ion loss) is shown in Fig. 11. The decay at each wavelength is characteristic of several decay times. The data is complicated by the fact that at each wavelength one produces O^+ signal via direct photodissociation to the repulsive wall of the $f^4\pi_g$ state (type II transitions in Fig. 1). This signal also has a time dependence.

To get approximate values for the lifetimes of the F" = 2 and 3 and F" = 1 and 4 spin components, I used a very simple model. I assumed that there were only two decay rates: one fast (F" = 2 and 3) and one slow (F" = 1 and 4). Added to this is the wavelength independent (but time dependent) signal from the direct dissociation. To get rid of this unwanted direct dissociation signal, the difference of the two sets of data is taken. The direct dissociation signal, being wavelength independent, vanishes from the picture. A plot of this difference will rise quickly due to the fast decay term and then fall gradually because of the slow decay. Figure 12 is a plot of the difference of the data shown on Fig. 11. The solid line is the curve generated using

$$p(t) = A_1 e^{-t/\tau_1} - A_2 e^{-t/\tau_2}$$

with τ_1 = 130 msec and τ_2 = 7 msec. A_1 and A_2 are population factors.

With the data we have, and using this simple approach, the following lifetimes were obtained: τ_1 = 130 ± 50 msec at 4967.2 Å (the F_2" and F_3" spin components) and τ_2 = 7 ± 2 msec at 4947.0 Å (the F_1" and F_4" spin components). Keep in mind that these are average values. As illustrated by James for the Cameron band of CO, the lifetime within a given multiplet can vary by an order of magnitude from low J to high J.

D. Conclusion

The laser photofragment spectrum of the O_2^+ $b^4\Sigma_g^- \longleftarrow a^4\pi_u$ system from 4902 to 5025Å measured as a function of time indicates that the $^4\pi_{5/2}$ and $^4\pi_{-1/2}$ spin components of the $a^4\pi_u$, metastable state radiatively decay to the $X^2\pi_u$ ground state with a time constant of 130 ± 50 msec at 4967.2Å; and similarly the $^4\pi_{3/2}$ and $^4\pi_{1/2}$ spin components decay with a time constant of 7 ± 2 msec at 4947.0Å. This observation is qualitatively in keeping with the interpretation that these lifetimes depend on the amount of spin-orbit mixing of the $a^4\pi_u$ state with the $A^2\pi_u$ state.

REFERENCES

1. (a) T. E. Nevin, Philos. Trans. R. Soc. London Ser. A 237, 471 (1938).

(b) T. E. Nevin, Proc. R. Soc. London Ser. A 174, 371 (1940).

(c) T. E. Nevin and T. Murphy, Proc. R. Ir. Acad. Sect. A 46, 169 (1941).

2. M. Vestal, G. Mauclaire and J. H. Futrell, 23. Annu. Conf. Mass Spectrom., Houston, Texas (1975).

3. (a) A. Tabche-Fouhaille, J. Durup, J. T. Moseley, J.-B. Ozenne, C. Pernot, and M. Tadjeddine, Chem. Phys. 17, 81 (1976).

(b) J. T. Moseley, M. Tadjeddine, T. Durup, J.-B. Ozenne, C. Pernot, and A. Tabche-Fouhaille, Phys. Rev. Lett. 37, 891 (1976).

(c) M. Tadjeddine, R. Abouaf, P. C. Cosby, B. A. Huber, and J. T. Moseley, J. Chem. Phys. 69, 710 (1978).

(d) J. T. Moseley, P. C. Cosby, J.-B. Ozenne, and J. Durup, J. Chem. Phys. 70, 1474 (1979).

(e) C. Pernot, J. Durup, and J.-B. Ozenne, J. A. Beswick, P. C. Cosby, and J. T. Moseley, J. Chem. Phys. 71, 2387 (1979).

(f) F. J. Grieman, J. T. Moseley, R. P. Saxon, and P. C. Cosby, Chem. Phys. 51, 169 (1980).

(g) H. Helm, P. C. Cosby, and D. L. Huetis, J. Chem. Phys. 73, 2629 (1980).

(h) P. C. Cosby, J.-B. Ozenne, and J. T. Moseley, J. Mol. Spect. 79, 203 (1980).

(i) J. C. Hanson, M. M. Graff, and J. T. Moseley, J. Chem. Phys. 74, 2195 (1981).

(j) P. C. Cosby and H. Helm, J. Chem. Phys. 76, 4720 (1982).

(k) J. C. Hansen and J. T. Moseley, J. Chem. Phys. 77, 1206 (1982).

(l) J. C. Hansen, J. T. Moseley, and P. C. Cosby, J. Mol. Spect. 98, 48 (1983).

(m) M. Carre, M. Druetta, M. L. Gaillard, H. H. Bukow, M. Horani, A. L. Roche, and M. Velghe, Mol. Phys. 40, 1453 (1980).

(n) A. Carrington, P. G. Roberts, and P. J. Sarre, Mol. Phys. 34, 291 (1977).

(o) A. Carrington, P. G. Roberts, and P. J. Sarre, Mol. Phys. 35, 1523 (1978).

(p) D. C. McGilvery, J. D. Morrison and D. L. Smith, J. Chem. Phys. 68, 4759 (1978).

(q) D. C. McGilvery, J. D. Morrison, and D. L. Smith, J. Chem. Phys. 70, 4761 (1979).

(r) R. Frey, R. Katoschke, K. Muller-Dethlefs, and E. W. Schlag, Z. Phys. A-Atoms and Nuclei 307, 25 (1982).

4. (a) J. T. Herron and H. I. Schiff, Can. J. Chem. 36, 1159 (1959).

(b) R. K. Curran, J. Chem. Phys. 38, 2974 (1963).

5. (a) K. M. Refaey and W. A. Chupka, J. Chem. Phys. 43, 2544 (1965).

(b) J. J. Leventhal and L. Friedman, J. Chem. Phys. 46, 997 (1966).

(c) K. R. Ryan, J. Chem. Phys. 51, 4136 (1969).

(d) J. M. Ajello, W. T Huntress, A. L. Lane, P. R. LeBreton, and A. D. Williamson, J. Chem. Phys. 60, 1211 (1974).

(e) J. M. Ajello, K. D. Pang, and K. M. Monahan, J. Chem. Phys. 61, 3152 (1974).

(f) J. M. Ajello and P. Rayermann, J. Chem. Phys. 62, 2917 (1975).

(g) P. M. Dehmer and W. A. Chupka, J. Chem. Phys. 62, 2228 (1975).

(h) J. M. Ajello, J. Chem. Phys. 63, 1863 (1975).

6. A. Dalgarno and M. B. McElroy, Planet. Space Sci. 13, 947 (1965).

7. D. W. Vance, Phys. Rev. 169, 263 (1968).

8. B. R. Turner, J. A. Rutherford, and D. M. J. Compton, J. Chem. Phys. 48, 1602 (1968).

9. D. L. Albritton, A. L. Schmeltekopf, W. J. Harrop, R. N. Zare, and J. Czarny, J. Mol. Spec. 67, 157 (1977).

10. Molecular Spectra and Molecular Structure: I. Spectra of Diatomic Molecules, G. Herzberg, p. 214, Van Nostrand Reinhold, New York (1950).

11. T. C. James, J. Chem. Phys. 55, 4118 (1971).

12. (a) I. Kovacs, Can. J. Phys. 36, 309 (1958).

(b) I. Kovacs, Can. J. Phys. 36, 329 (1958).

13. D. L. Albritton, W. J. Harrop, A. L. Schmeltekopf, and R. N Zare, J. Mol. Spec. 46, 89 (1973).

FIGURE CAPTIONS

Fig. 1. Relevant potential energy curves of O_2^+.

Fig. 2. Data collection timing sequence.

Fig. 3. The photodissociation spectrum of O_2^+ taken at trapping times of 0.1 msec and 100 msec. The data shown has had a constant subtracted from it to remove signal from the direct dissociation process.

Fig. 4. The O^+ photofragment yield as a function of time.

Fig. 5. Power dependence of O_2^+ photofragmentation at 5320Å.

Fig. 6. The number of O_2^+ ions decrease while the number of O^+ ions increase when an ion bunch collides with the octapole trap mirror.

Fig. 7. The O^+ background in the ion trap as a function of time.

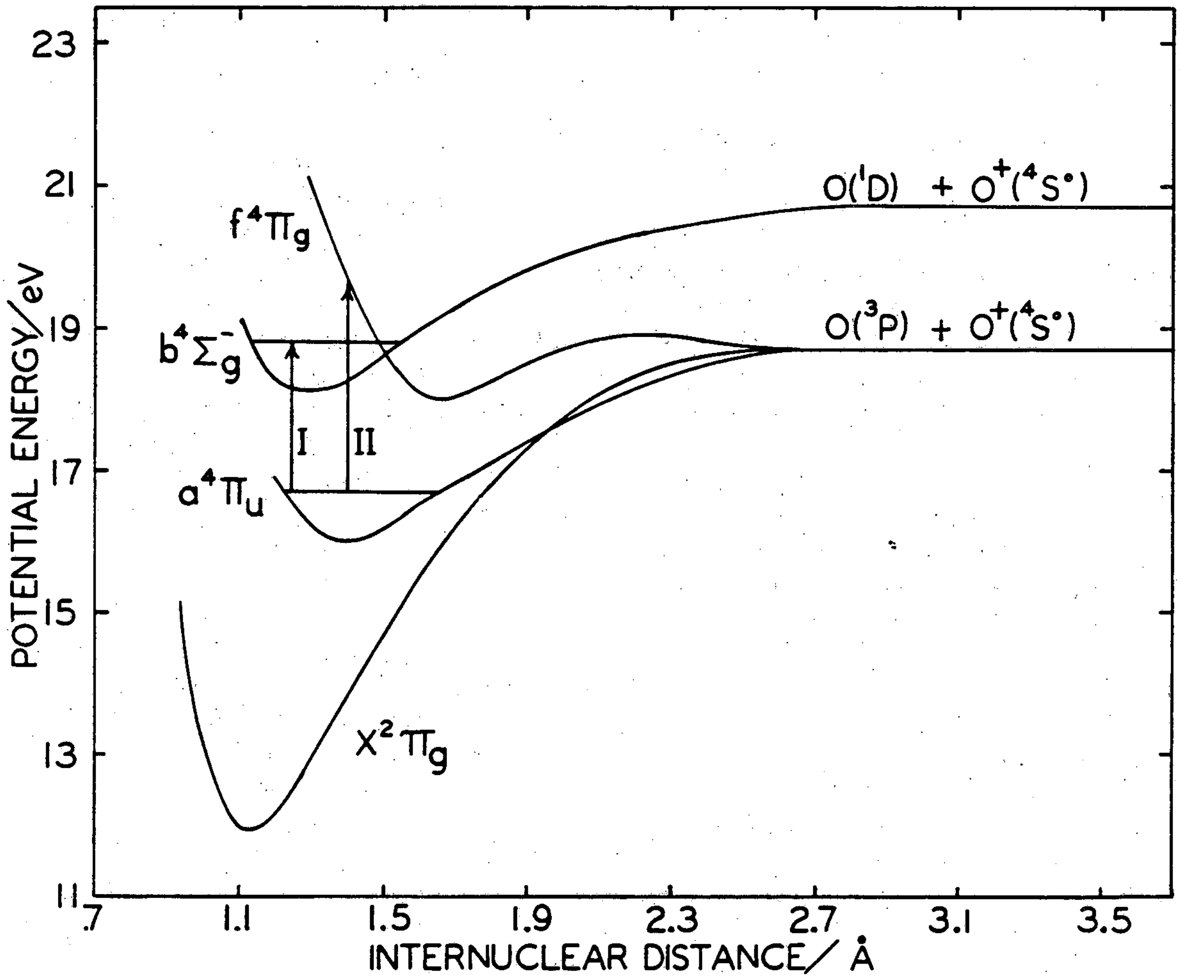
Fig. 8. The photodissocation spectrum of O_2^+ (the $\Delta v = 3$ peak) taken at trapping times of 0.1 msec and 100 msec. Again the data has a constant subtracted from it.

Fig. 9. Computer simulation of the data in Fig. 8 calculated using a lifetime of 120 msec for the $^4\pi_{5/2}$ and $^4\pi_{-1/2}$ spin components and 6 msec for the $^4\pi_{3/2}$ and $^4\pi_{1/2}$ components.

Fig. 10. Correlation diagram for the $^4\pi_u$ state of O_2^+ showing the transition from a Hund's case (a) to nearly a Hund's case (b) description.

Fig. 11. The O^+ photofragment yield as a function of time at 4947.0Å and 4967.2Å corrected for parent ion leakage from the trap.

Fig. 12. This figure shows the difference of the data shown in Fig. 11. The solid curve represents a fit to the data with a long decay constant (τ_1) of 130 msec for the F_1'' and F_2'' spin components and a short decay constant (τ_2) of 7 msec for the F_2'' and F_3'' spin components. The upper and lower dotted curves use τ_1 = 180 msec, τ_2 = 5 msec and τ_1 = 80 msec, τ_2 = 9 msec, respectively.



XBL 7812-13896

Figure 1

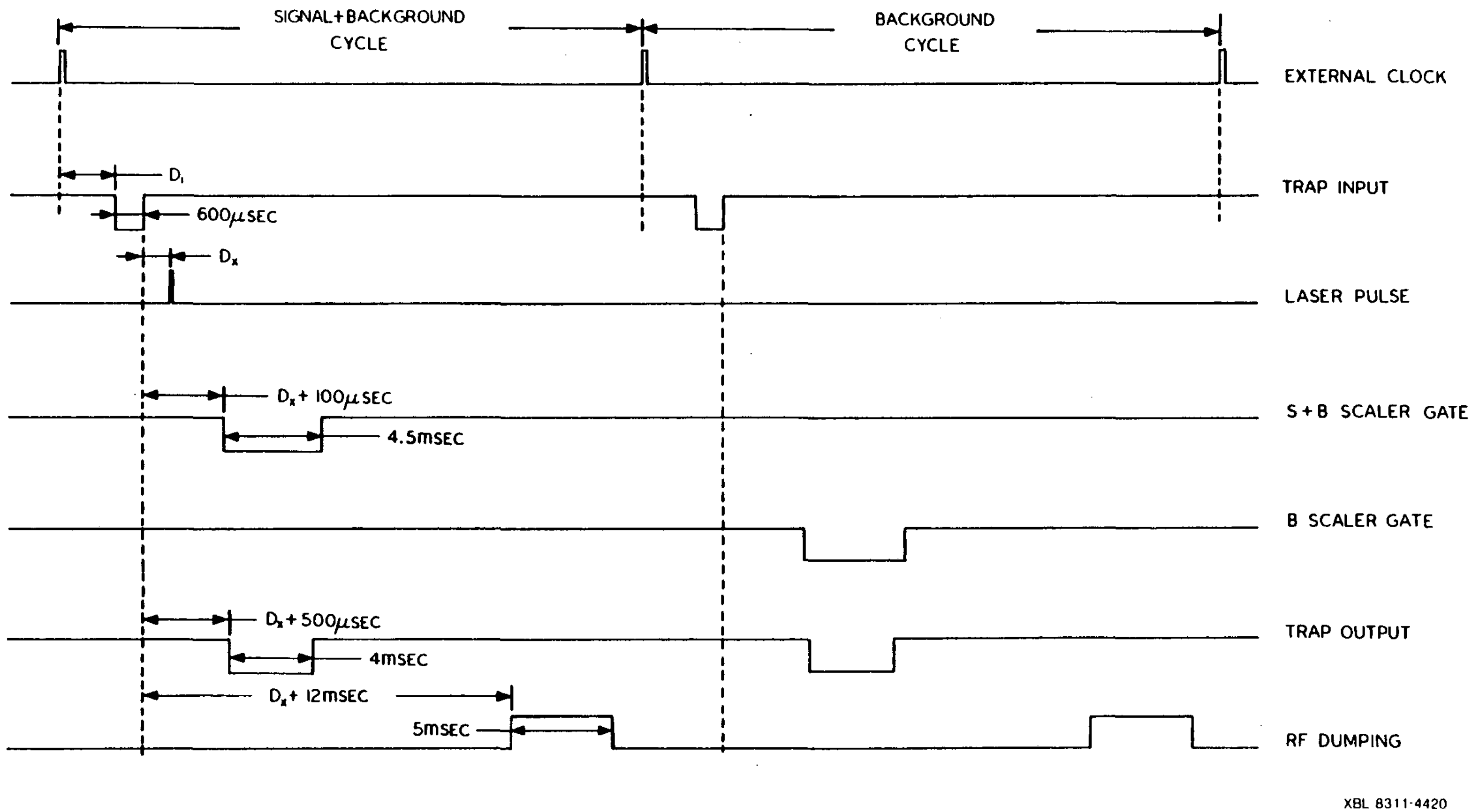
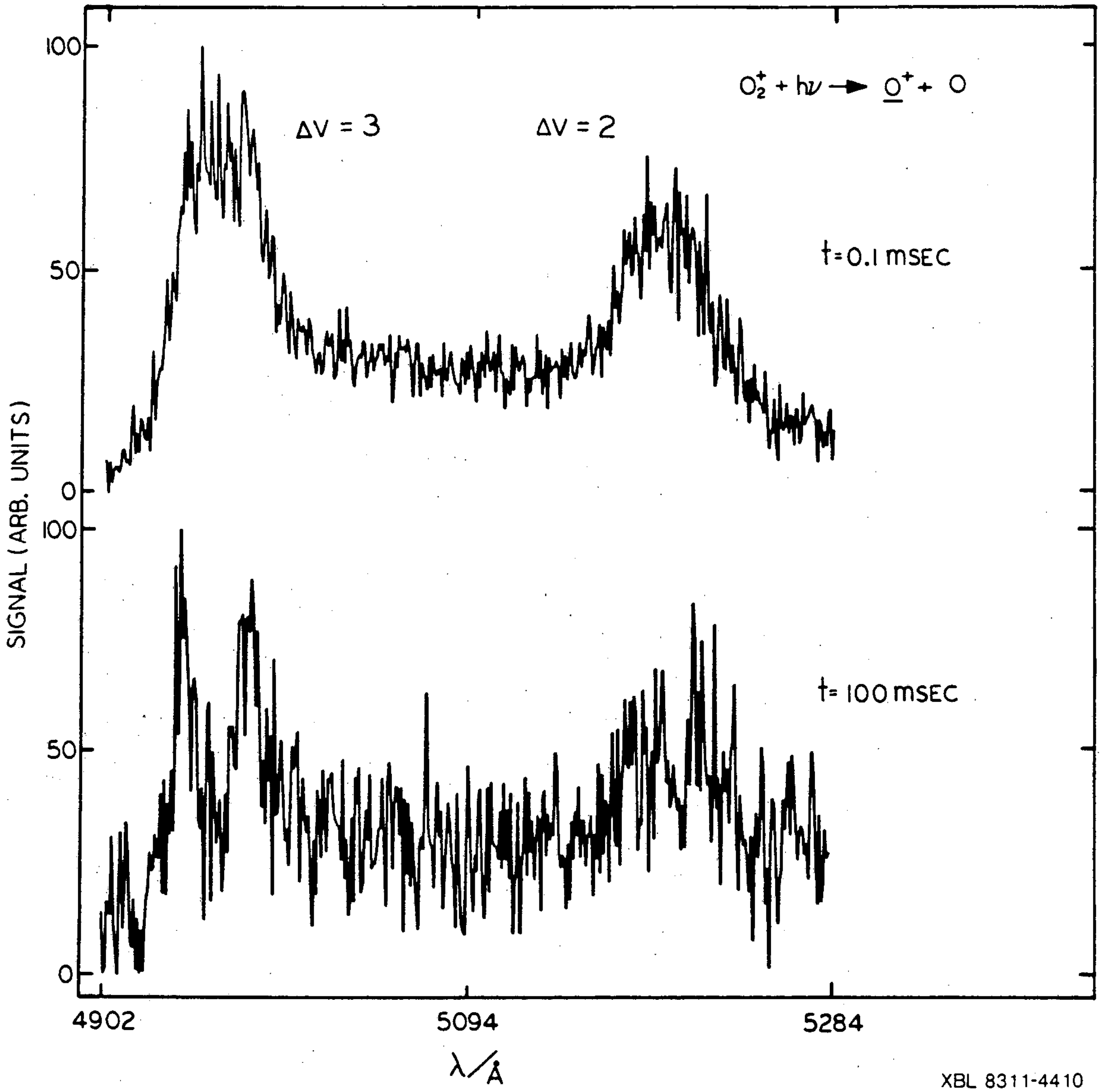

SIGNAL+BACKGROUND
CYCLE
BACKGROUND
CYCLE
EXTERNAL CLOCK
D_1
TRAP INPUT
600μSEC
D_x
LASER PULSE
D_x+ 100μSEC
S+B SCALER GATE
4.5mSEC
B SCALER GATE
D_x+ 500μSEC
TRAP OUTPUT
4mSEC
D_x+ 12mSEC
5mSEC
RF DUMPING
XBL 8311-4420

Figure 2



$O_2^+ + h\nu \rightarrow \underline{O}^+ + O$
ΔV = 3
ΔV = 2
t=0.1 msec
t= 100 msec
SIGNAL (ARB. UNITS)
100
50
0
100
50
0
4902
5094
5284
λ/Å
XBL 8311-4410

Figure 3

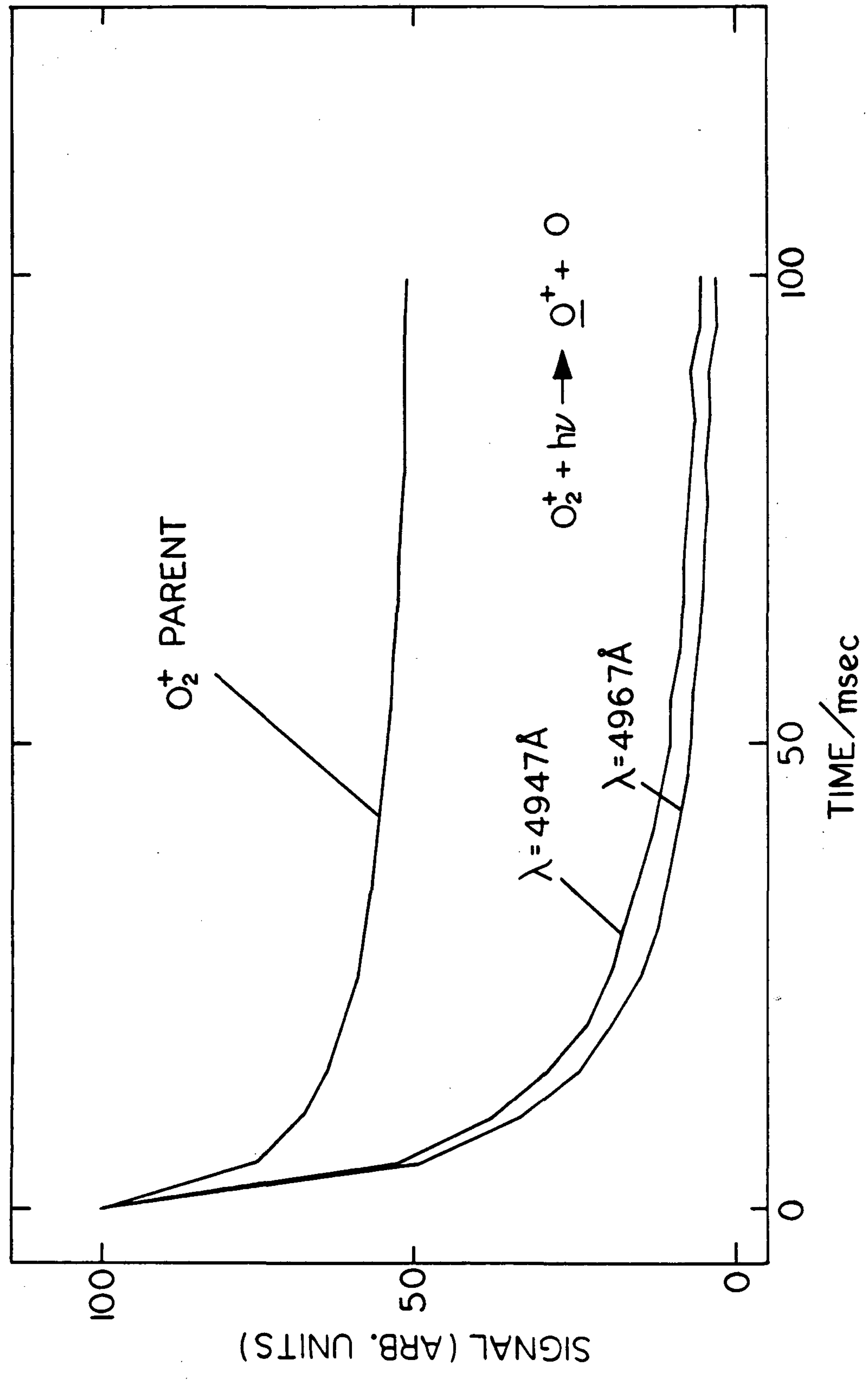
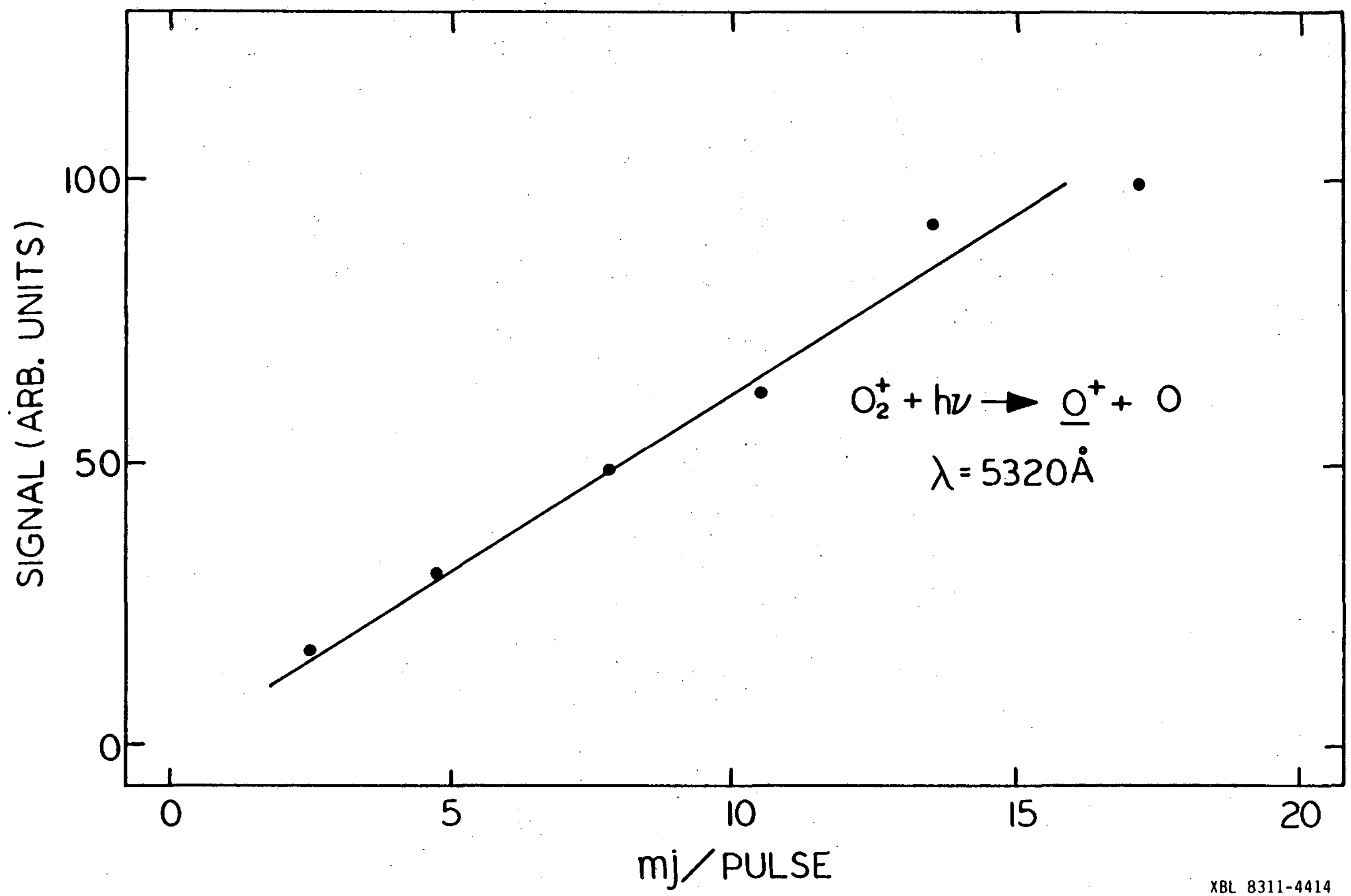


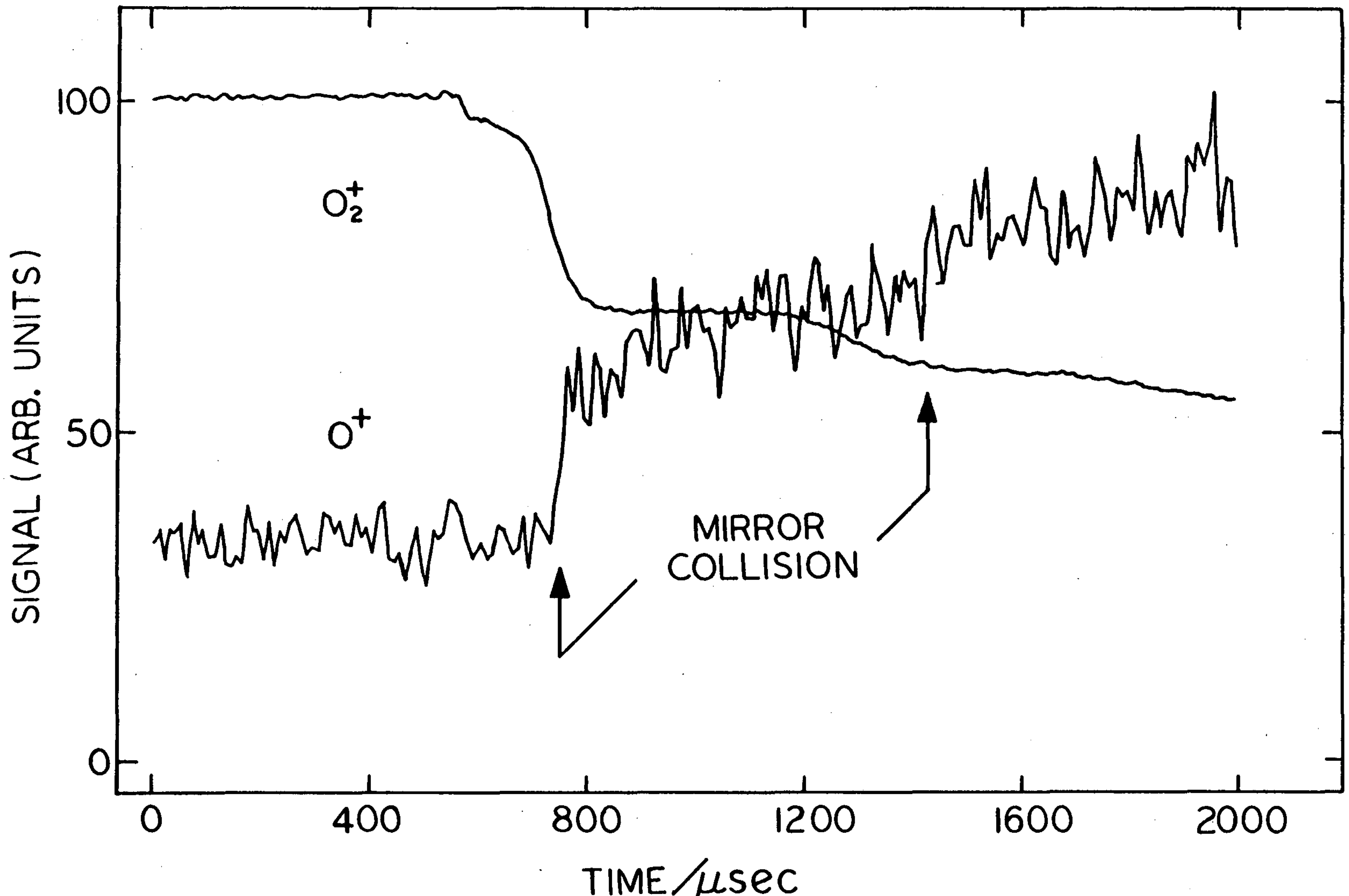
Figure 4

XBL 8311-4415



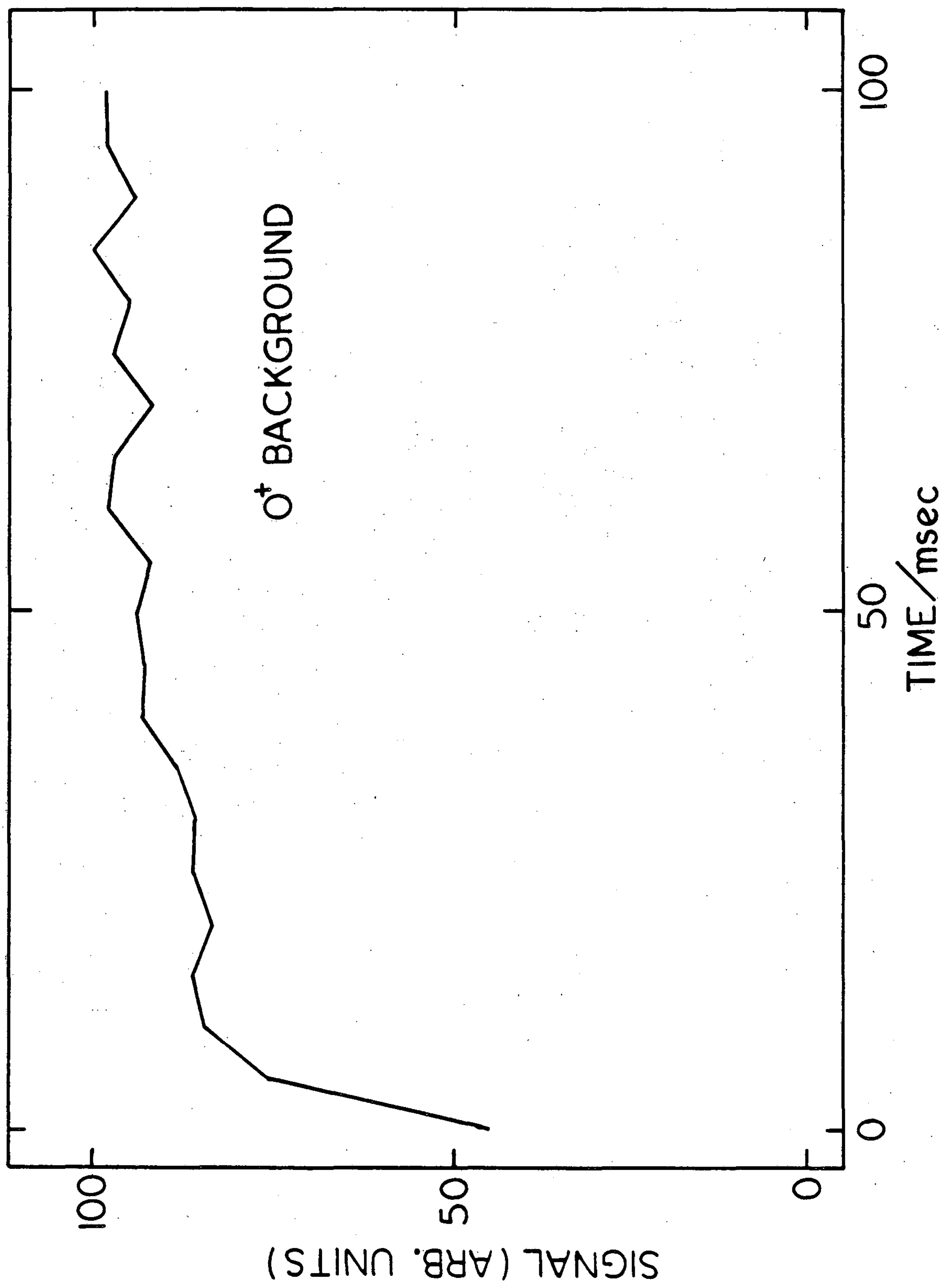
XBL 8311-4414

Figure 5



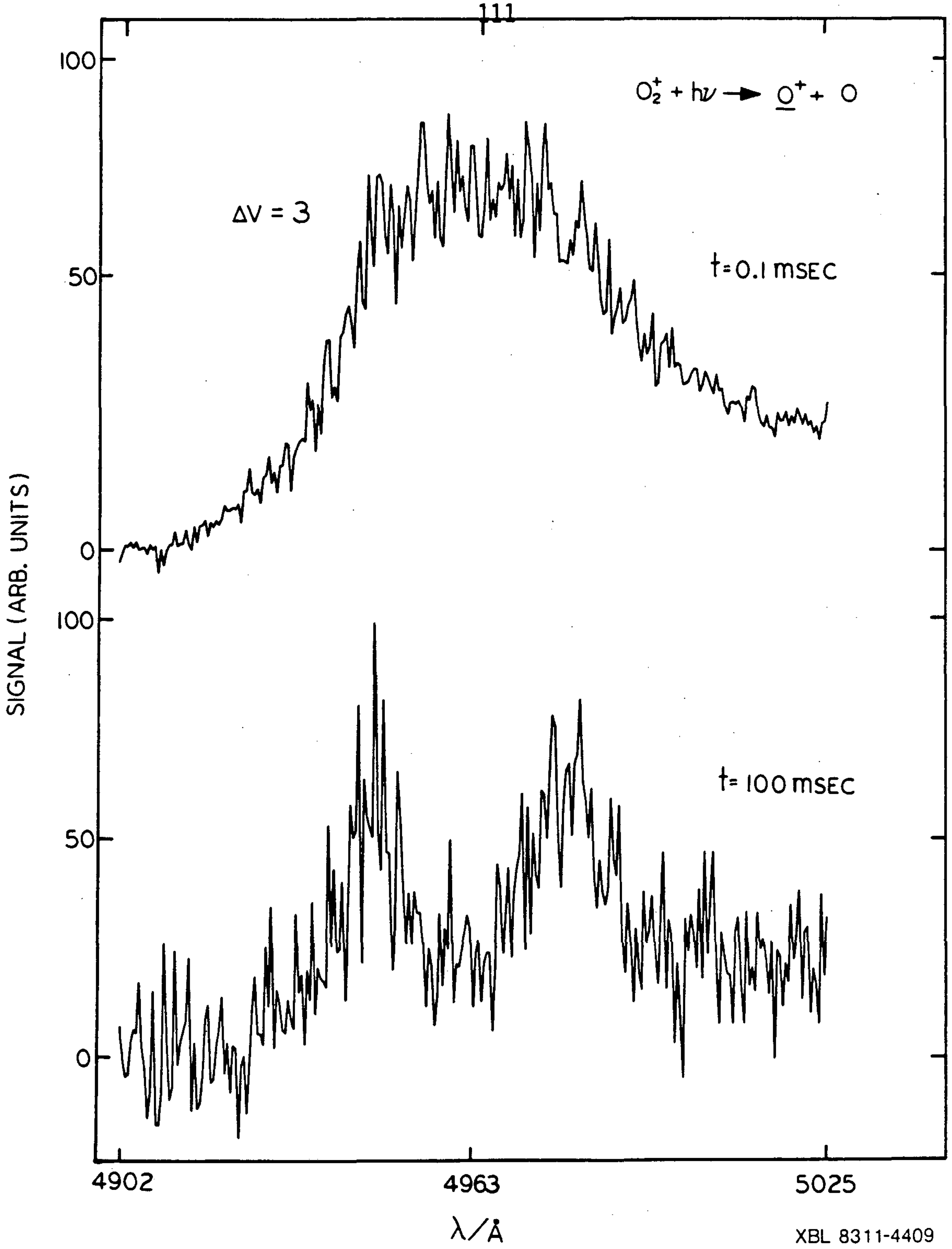
XBL 8311-4413

Figure 6



XBL 8311-4412

Figure 7



$O_2^+ + h\nu \rightarrow \underline{O}^+ + O$
$\Delta V = 3$
t=0.1 MSEC
t= 100 MSEC
100
50
0
100
50
0
SIGNAL (ARB. UNITS)
4902
4963
5025
λ/Å
XBL 8311-4409

Figure 8

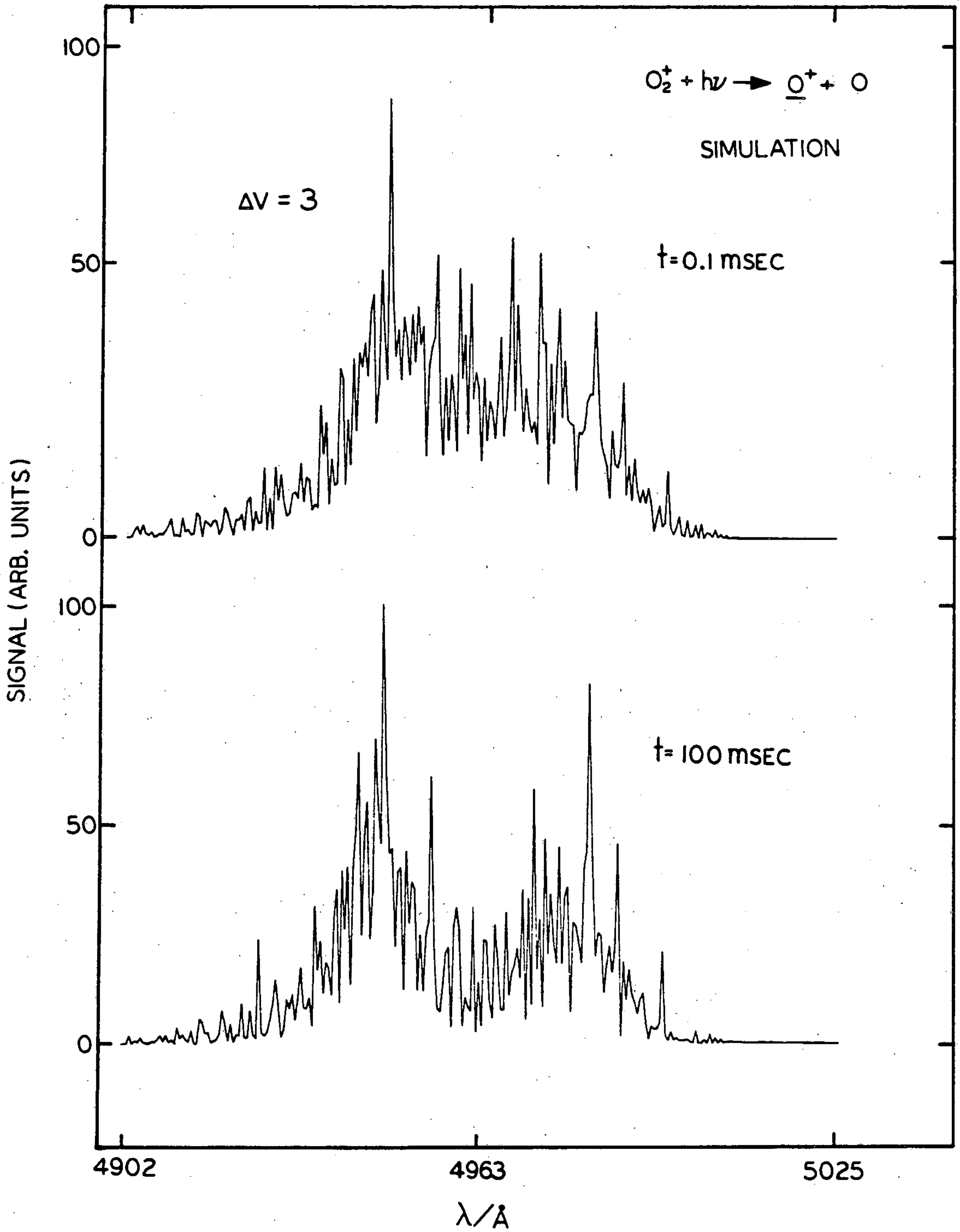
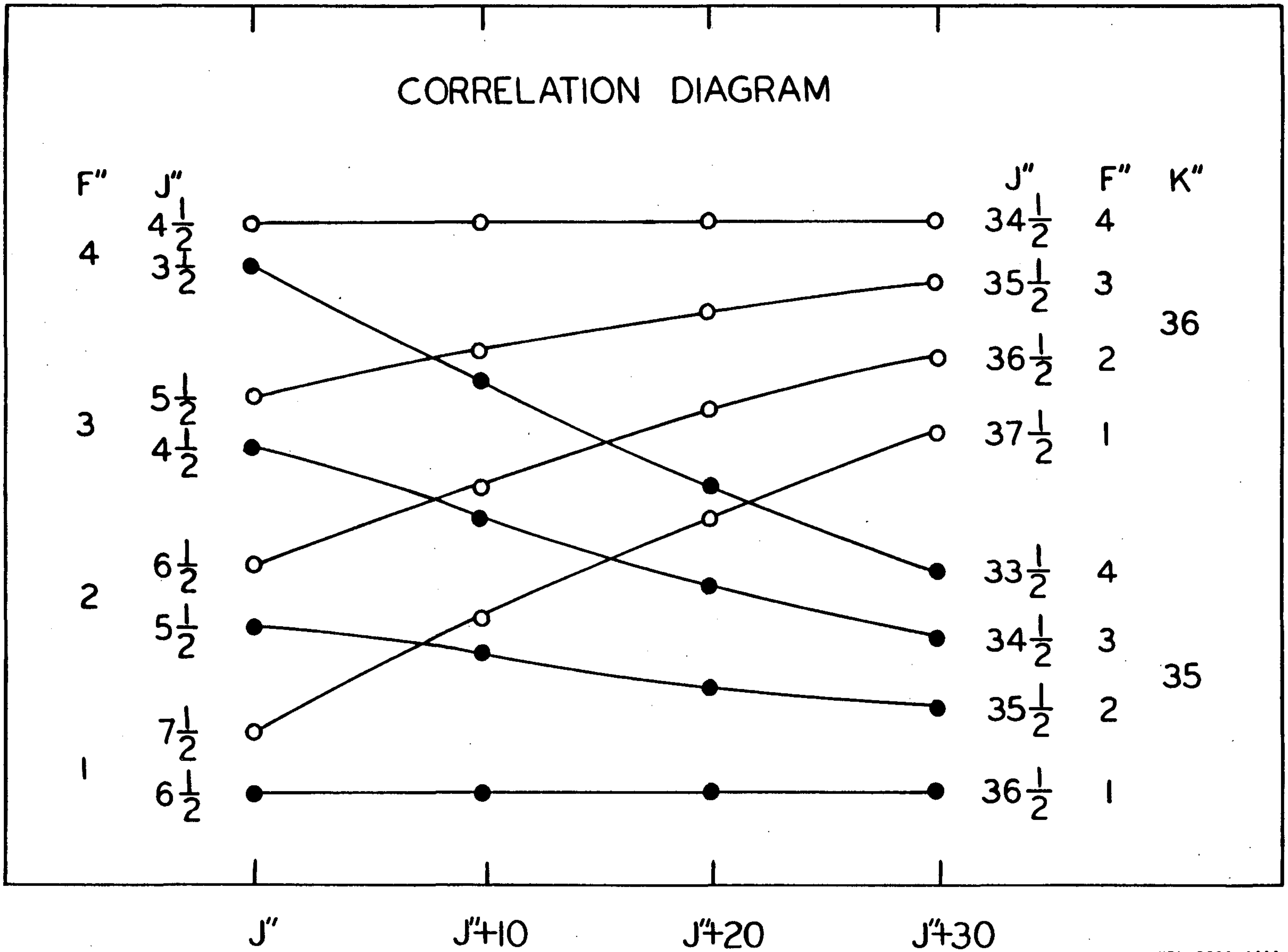


Figure 9

XBL 8311-4470



XBL 8311-4411

Figure 10

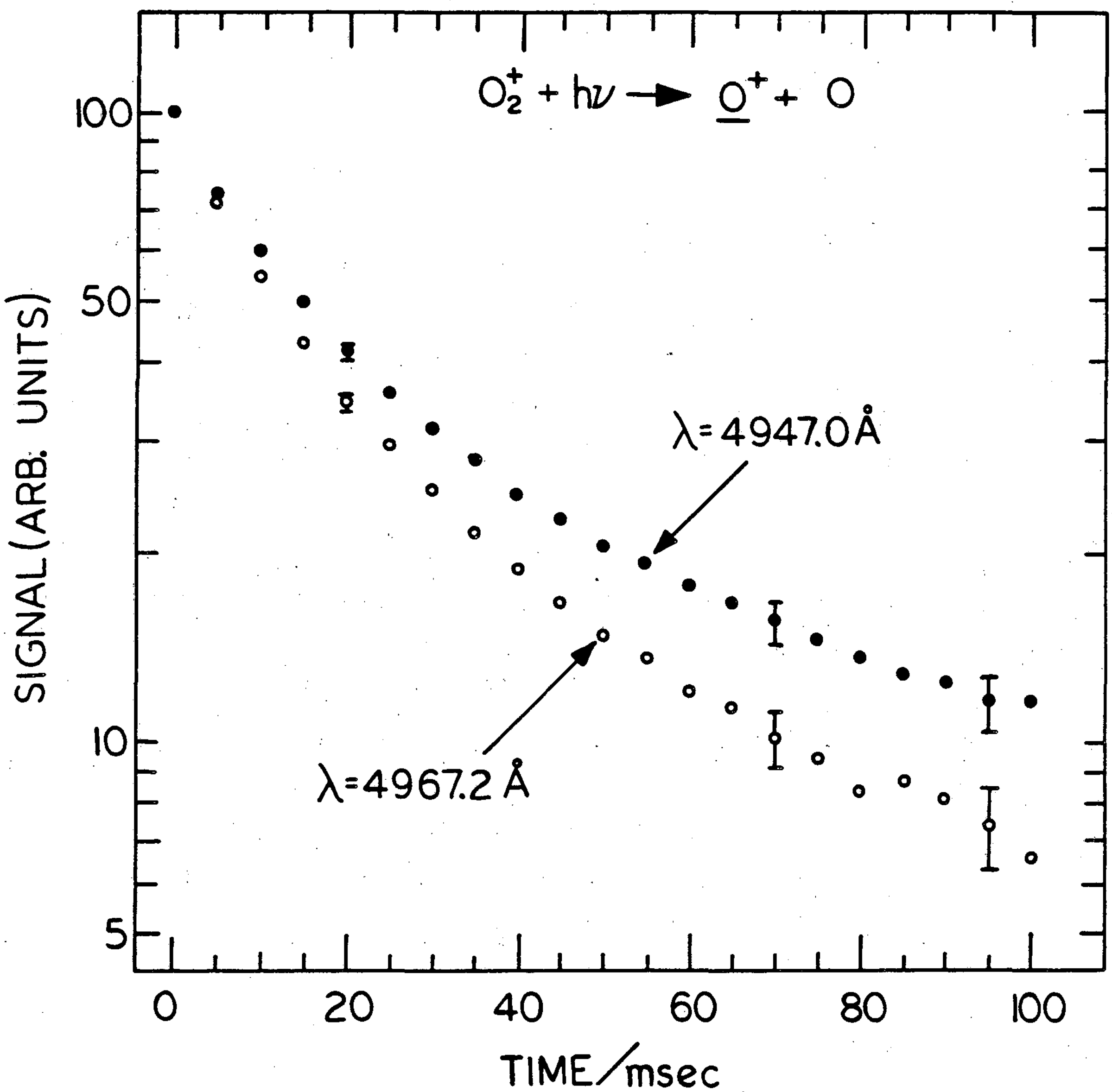

$O_2^+ + h\nu \rightarrow \underline{O}^+ + O$
λ=4947.0 Å
λ=4967.2 Å
SIGNAL (ARB. UNITS)
100
50
10
5
0
20
40
60
80
100
TIME/msec
XBL 8311-4408

Figure 11

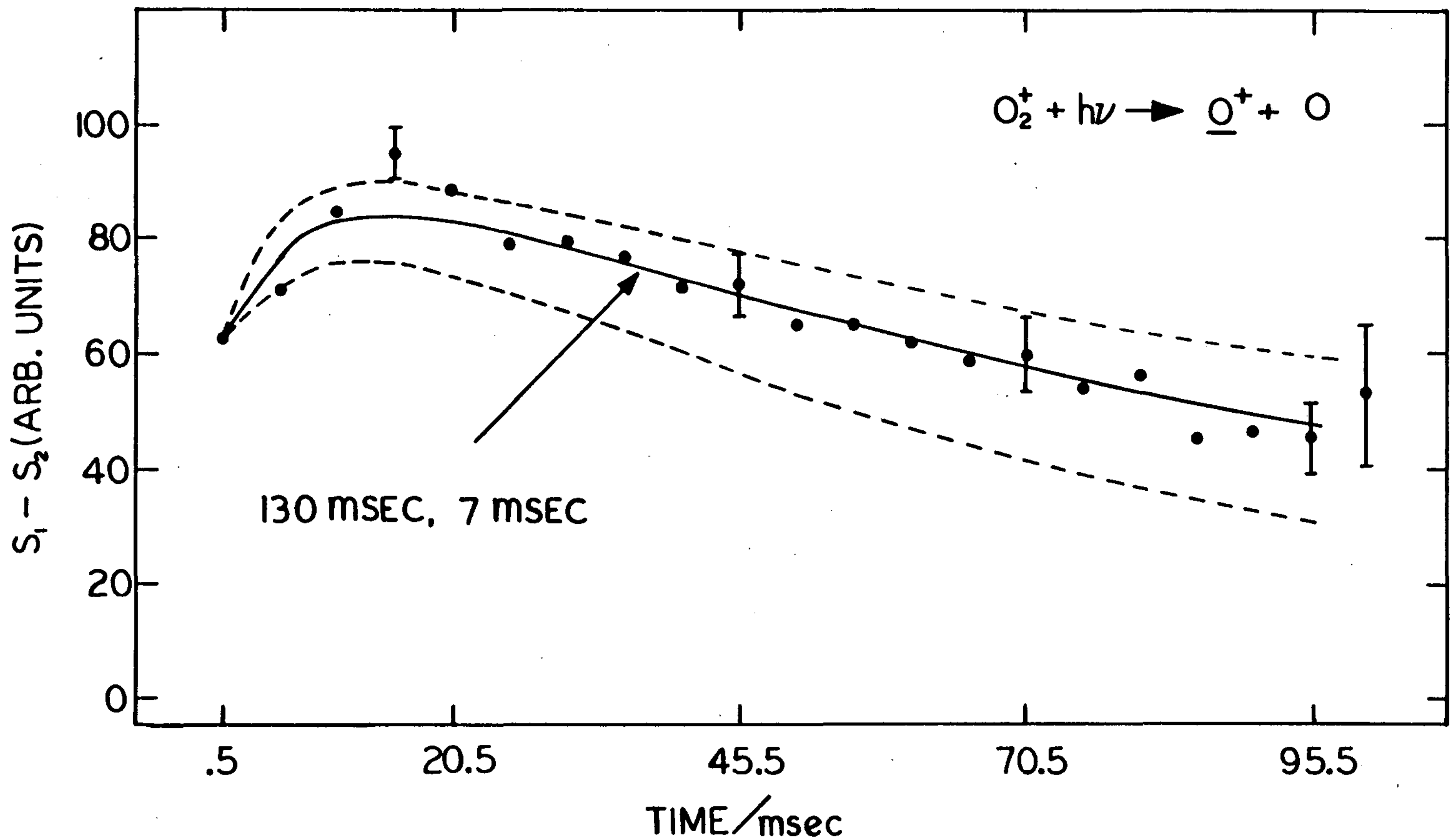


Figure 12

XBL 8311-4469

IV. APPENDIX: DATA ACQUISITION PROGRAM

The program to be listed here is a first version data acquisition/data management program written for the molecular ion spectrometer. It is to be considered in the developmental stage. The program contains oddities, inconsistencies, and some bugs (harmless bugs in the display routine and some timing constraints in the data collection cycles). But I think the program is basically sound and is presented here for reference.

ABBREVIATED MIPDS COMMAND LIST

AB		ACTIVE CHANNEL BRIGHTNESS
AC		ACCUMULATE DATA
AS		ABSOLUTE SCALING
CL		CLEAR DATA BUFFER
CU		CLOCK UNITS
CS		CHANNEL SCALE
DC		DYE LASER CALIBRATION
DI		DISPLAY
	BA	BACKGROUND
	DA	DATA
	EX	EXPAND
	PA	PARAMETERS
	PN	POWER NORMALIZE

	PO	POWER BUFFER
	RE	REVERSE
	SB	SIGNAL + BACKGROUND BUFFER
	SI	SIGNAL (SB-BA)
ER		ENTER RUN NUMBER
EX		EXIT PROGRAM
GA		GAIN OF A2D CONVERTER
	LO	LOW
	ME	MEDIUM
	HI	HIGH
HA		HALT DATA ACCUMULATION
HE		HEADER
IW		INTEGRATE WINDOW
KM		KEY MODE INTERRUPT
LA		LASER
LOCK		LOCK PROGRAM CONDITIONS
MA		MAGNET
MC		MCS UNIT
	BA	PLACE DATA IN BACKGROUND BUFFER
	HA	HALT DATA COLLECTION
	IN	INITIALIZE
	OF	OFF
	SB	PLACE DATA IN SIGNAL+BACK BUFFER
	ST	START DATA COLLECTION
	ZZ	RELOAD COUNTERS

NC		NUMBER OF DATA CYCLES
ND		NEW DATE
NO		NUMBER OF CHANNELS
NS		NUMBER OF SCANS (SWEEPS)
PR		PRINTER COMMANDS
	DA	DATA
	HE	HEADER
	FU	FULL INFO.
	ST	STATUS
PB		PLOT BOX
PC		PLOT CHANNEL NUMBERS
PH		PLOT HORIZONTAL LINE
PL		PLOT FULL FIGURE
PM		PLOT MOVE
PP		PLOT POINTS
PV		PLOT VERTICAL LINE
QU		QUADRUPOLE
QP		QUADRUPOLE PEAKS OPTION
RE		RESET DEVICE
RD		READ DISK
RR		RECALL RUN NUMBER
RT		RE-TAKE MODE/ACCUME MODE
SA		SCALE ADVANCE
SC		SCAN DEVICE
	LA	LASER

	MA	MAGNET
	MC	MCS UNIT
	NO	"NOTHING"
	QU	QUADRUPOLE
	TR	TRAP
SD		SCALE DOWN
SE		SET DEVICE
	LA	LASER
	MA	MAGNET
	QU	QUADRUPOLE
	TR	TRAP
SF		SCROLL FASTER
SL		SCROLL LEFT
SM		SMOOTH DISPLAYED DATA
SP		SCANNING PROCEDURE
	BI	BIDIRECTIONAL
	UN	UNIDIRECTIONAL
SR		SCROLL RIGHT
SS		SCROLL SLOWER
ST		START (PERFORMS HA, RE, CL, AC SEQUENCE)
SZ		SET ZERO FOR DISPLAY
TR		TRAP
TY		TYPE TO CONSOLE DEVICE
	DA	DATA
	HE	HEADER

	ST	STATUS
UP		UPDATE DISPLAY
US		UPDATE SET (CHANNELS BETWEEN UPDATES)
WA		WINDOW APERTURE
WB		WINDOW BRIGHTNESS
WD		WRITE TO DISK
WI		WINDOW INITIAL CHANNEL
WL		WINDOW LEFT
WR		WINDOW RIGHT
WB		WINDOW SCALE

```
        .TITLE  MACHINE PROGRAM
;
;THE TANDEM MASS SPECTROMETER
;
;
;       SYSTEM MACROS
        .MCALL  .TTYOUT,.TTYIN,.TTINR,.PRINT,.DATE,.INTEN
;
;       GLOLIB LIBRARY GLOBALS
;
        .GLOBL  READ,ASCBIN,BINASC,DPBASC,DECOUT,DCOUT2
        .GLOBL  LPOUT,LPRINT,LPCRLF,LPDCO2,LPDECO
        .GLOBL  REGSAV,REGRES,DYECAL,DYELAS,DYESET
        .GLOBL  MOTOR1,MOTOR2,CRLF,QUAD,WRDISK,REDISK
        .GLOBL  LOWORD,HIWORD,A2DAMP,A2DGNS,MAGNET,PLOTTR
;
;       SYSTEM  GLOBALS
;
        .GLOBL  JMUL,JDIV,JCMP,R50ASC
;
;
;       GLOBAL LABELS DEFINED IN THIS MODULE
;
        .GLOBL  LIST,ARG1,ARG2,RESULT,IOBUF,DATBUF,DATALO
        .GLOBL  SCCODE,LONWAV,SHOWAV,SETWAV,WAVINC,CALWAV
        .GLOBL  NSTEP,NCHAN,MDAC,MDAC1,MRES,MDAC2,MASBUF
        .GLOBL  SCFUNC,QP,CLEAR,BUFINIT
        .GLOBL  MADAC,MADAC1,MADAC2,MARES,DICHNS,DISTRT
        .GLOBL  STCHAN,PLCTRL,PLMOV,PLLINE,PLCHAR,MCHSUB
        .GLOBL  RUN,YMAX,UPDATE
;
KWCSR=170420                            ;CONTROL STATUS REGISTER FOR KWV11
KWBPR=170422                            ;CLOCK BUFFER/PRESET REGISTER(KWV11)
KWVEC=350                               ;CLOCK INTERRUPT VECTOR
MDAC=170440                             ;AAV11 FOUR CHANNEL A/D CONVERTER
MADAC=170442
XDAC=170444
YDAC=170446
ADCSR=170450                            ;A/D CONTRIL-STATUS REGISTER(DT2771)
XYBUF=170452                            ;X/Y BUFFER
DMAPCR=170454                           ;DMA POINT COUNT REGISTER
DMACAR=170456                           ;DMA CURRENT ADDRESS REGISTER
DMAVEC=310                              ;DMA INTERRUPT VECTOR
DATALO=166000                           ;CAMAC CONTROLLER ADDRESSES
DATAHI=166002
LAMLO=166004
LAMHI=166006
STATU1=166010
STATU2=166012
SCAL0=167040                            ;SCALER 0 (BIN 17)
SCAL1=167042                            ;SCALER 1
SCAL2=167044                            ;SCALER 2
SCAL3=167046                            ;SCALER 3
A2DCON=167100                           ;A/D CONVERTER (BIN 18)
A2DVEC=510                              ;A/D CONVERTER INTERRUPT VECTOR
```

```
DMAPRI=0
MCSA0=167200                                   ;MCS CRATE LOCATIONS
MCSA1=167202
MCSA2=167204
MCSA3=167206
MCSA4=167210
MCSA5=167212
MCSA6=167214
MCSNVC=520
JSW=44                                         ;JOB STATUS WORD
;
;
;
;
;
MCHSUB: MOV     #DATAIN,A2DVEC                 ;SET VECTORS
        MOV     #200,A2DVEC+2
        MOV     #CLKINT,KWVEC                  ;CLOCK VECTOR
        MOV     #200,KWVEC+2
        MOV     #DMAINT,DMAVEC                 ;POINT PLOTTER VECTOR
        MOV     #200,DMAVEC+2
        MOV     #MCSINT,MCSNVC                 ;MCS VECTOR
        MOV     #200,MCSNVC+2
        MOV     #1.,NSCAN+2                    ;DEFAULT VALUE
        MOV     #50.,UPDATE+2                  ;UPDATE PERIOD
        MOV     #1,MARES                       ;DEFAULT STEP SIZE FOR MAGNET
        MOV     #1,MARES+2                     ;COPY
        MOV     #340.,MADAC1                   ;DEFAULT LOW CURRENT FOR MAGNET
        MOV     #10.,MADAC1+2                  ;COPY
        MOV     #340.,MADAC                    ;SET MAGNET LOW CURRENT
        MOV     #20.,MADAC2                    ;DEFAULT HIGH CURRET VALUE
        MOV     #1,MRES                        ;DEFAULT STEP SIZE FOR QUAD
        MOV     #1,MRES+2                      ;COPY
        MOV     #40.,MDAC1                     ;DEFAULT LOW MASS FOR QUAD
        MOV     #2,MDAC1+2                     ;COPY
        MOV     #40.,MDAC                      ;SET QUAD LOW MASS
        MOV     #4,MDAC2                       ;DEFAULT HIGH MASS FOR QUAD
        MOV     #1,NSTEP                       ;DEFAULT STEP SIZE FOR LASER
        MOV     #1,DACYCL                      ;ONE DATA CYCLE PER CHANNEL
        MOV     #1,DACYCL+2                    ;SAVE A COPY
        MOV     #1,DELINC                      ;1MSEC DELAY INCREMENT
        MOV     #1,DELOFF                      ;1MSEC DELAY OFFSET
        MOV     #-1,KWBPR                      ;LOAD CLOCK BUFFER/RESET REG.
        MOV     #200.,NCHAN                    ;200 CHANNELS BY DEFAULT
        MOV     #'UN,SCTYPE                    ;UNIDIRECTIONAL SCANNING
        MOV     #NO,SCFUNC                     ;'NOTHING'FUNCTION BEING SCANNED
        MOV     #'NO,SCCODE                    ;'NOTHING' CODE WORD
        MOV     #SI,DIFUNC                     ;SIGNAL BEING DISPLAYED
        MOV     #SI,DIFUNC+2                   ;SAVE CODE
        MOV     #'SI,DICODE                    ;LOAD DISPLAY CODE
        MOV     #'SI,DICODE+2                  ;SAVE DICODE
        MOV     #'LO,A2DGNS                    ;A2D CONVERTER GAIN LOW(X1)
        MOV     #0,DISFLG                      ;AUTO-SCALE
        MOV     #0,FLGDIS                      ;NORMAL--NOT POWER NORM.
        MOV     #0,ZERFLG                      ;ZERO SHIFT
```

```
        CLR     SCMODE                  ;EXT. INTERRUPT MODE
        CLR     RETAKE                  ;ACCUMULATE MODE
        MOV     #4,ACTIVE               ;ACTIVE CHANNEL BRIGHTNESS
        MOV     #4,ACTIVE+2             ;ACTIVE CHANNEL BRIGHTNESS
        CLR     SLRFLG                  ;NO SCROLLING
        CLR     YMIN
        CLR     YMIN+2
        MOV     #4096.,SCROLL           ;SCROLL SPEED
        MOV     #2048.,YSCALE           ;SCALE FOR Y-AXIS
        MOV     #1,RUNFLG               ;RUN COMPLETED FLAG
        CLR     RUN                     ;NO RUNS COMPLETED
        CLR     RUN+2
        CLR     WINBRT                  ;DISPLAY WINDOW NOT BRIGHT
        CLR     WINFLG                  ;CLEAR WINDOW FLAG
        MOV     #10.,WINSIZ             ;DEFAULT WINDOW SIZE
        CLR     ADCSR                   ;CLEAR AD CONTROL STATUS REG.
        MOV     #DATBUF,R1              ;DATA BUFFER STARTING LOCATION TO R1
        ADD     #100.,R1                ;POINT TO FIRST HEADER LOCATION
        MOV     #200,(R1)               ;FILL WITH TERMINATION SYMBOL
        CALL    BUFINIT                 ;GENERATE BUFFER POINTERS
        CALL    CLEAR                   ;CLEAR DATA BUFFER
        MOV     #'>,R0                  ;PROMPT CHARACTER
        .TTYOUT
;
;       ENABLE CAMAC INTERRUPTS
;
        BIC     #70,STATU1              ;CLEAR NOE,NXE,I
        BIS     #301,STATU1             ;SET INE,REIN,OII
        MOVB    #26.,A2DCON             ;ENABLE A2D CONVERTER
;
;
        MOV     #20140,KWCSR            ;SET CLOCK CONTROL STATUS REGISTER
;
;
        .DATE                           ;GET SYSTEM DATE
        MOV     R0,DATE                 ;SAVE
        CALL    DATES                   ;DECODE
;
;
DISPLA: TST     UPDATE                  ;UPDATE?
        BNE     5$                      ;BRANCH IF NOT EQ TO ZERO
        MOV     DICODE+2,DICODE         ;GET RID OF 'SM' IF THERE
        MOV     DIFUNC+2,DIFUNC         ;RESTORE FUNCTION CODE
        CALL    DISCAL                  ;IF ZERO UPDATE DISPLAY BUFFER
        MOV     UPDATE+2,UPDATE         ;REFRESH UPDATE
5$:     BIT     #1,RUNFLG               ;RUN COMPLETED?
        BNE     10$
        CALL    CRLF                    ;TYPE CARRIAGE RETURN-LINE FEED
        .PRINT  #M23A
        MOV     #M23,R0                 ;BEEP
        CALL    LPRINT
        MOV     #'>,R0                  ;TYPE '>'
        .TTYOUT
        BIS     #1,RUNFLG
        BR      TTIN
```

```
10$:     CMP     #'MC,SCCODE             ;IS MCS ACTIVE?
         BNE     TTIN
         BIT     #1,OVF
         BGT     CNTOVF
         BIT     #2,OVF
         BGT     RAMOVF
         BIT     #4,OVF
         BGT     ENDRUN
         BR      TTIN
CNTOVF: .PRINT   #M82                    ;COUNTER OVER FLOW
         BIC     #1,OVF
         BR      DISPLA
RAMOVF: .PRINT   #M83                    ;RAM OVER FLOW
         BIC     #2,OVF
         BR      DISPLA
ENDRUN: BIC      #4,OVF                  ;RUN COMPLETED
         BIC     #1,RUNFLG               ;GENERATE MESSAGE
         INC     RUN                     ;RUNS FOR THE DAY
         MOV     RUN,RUN+2               ;SAVE COPY
         BR      DISPLA
TTIN:    MOV     #IOBUF,R1               ;STARTING ADDRESS OF I/O BUFFER IN R
1
20$:     .TTINR                          ;IS THERE A CHARACTER?
         BCS     DISPLA                  ;BRANCH TO DISPLA IF NONE
         MOVB    R0,(R1)+                ;PLACE CHARACTER IN I/O BUFFER
         CMPB    #12,R0                  ;WAS IT A LINE FEED?
         BNE     20$                     ;IF NO,GET MORE
         CMP     DICHNS,#11.             ;HALT DMA IF TOO FEW CHANNELS
         BPL     OPTION
         CALL    DISHLT
;
;
;THIS OPTION SEARCHES FOR THE DESIRED OPTION BY COMPARING THE
;COMMAND ENTERED WITH THE OPTION LIST
;
OPTION: CMP      #'AC,IOBUF              ;ACCUMULATE DATA
         BNE     1$
         MOV     #1,ACCFLG
         BIS     #2,RUNFLG               ;RUN IN PROGRESS
         JMP     OPTOUT                  ;LEAVE
1$:      CMP     #'DC,IOBUF              ;DYE CALIBRATION
         BNE     2$
         CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
         BEQ     1001$                   ;BRANCH IF UNLOCKED
         JMP     OPTLOC                  ;LOCKED! LEAVE WITH MESSAGE
1001$:   CALL    DYECAL
         JMP     OPTOUT                  ;LEAVE
2$:      CMP     #'DI,IOBUF              ;DISPLAY OPTION
         BNE     3$
         CALL    DISOPT
         JMP     OPTOUT                  ;LEAVE
3$:      CMP     #'LA,IOBUF              ;DYE LASER
         BNE     4$
         CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
         BEQ     1003$                   ;BRANCH IF UNLOCKED
```

```
        JMP     OPTLOC              ;LOCKED! LEAVE WITH MESSAGE
1003$:  CALL    DISHLT              ;HALT   DISPLA
        CALL    DYELAS
        CMP     #'LA,SCCODE         ;IS IT BEING SCANNED?
        BNE     1004$
        CALL    BUFINIT
        CALL    CLEAR               ;CLEAR DATA BUFFER
        CLR     UPDATE
1004$:  JMP     OPTOUT              ;LEAVE
4$:     CMP     #'EX,IOBUF          ;EXIT MACRO PROGRAM
        BNE     5$
        .PRINT  #M71                ;EXIT PROGRAM!!
        .PRINT  #M48                ;ARE YOU SURE?
        CALL    READ                ;YES OR NO
        CMPB    #'Y,IOBUF
        BEQ     1006$
        JMP     OPTOUT
1006$:  JMP     EXIT                ;RETURN TO FORTRAN CALLER
5$:     MOV     #SA,YFUNC           ;SCALE UP
        CMP     #'SA,IOBUF
        BEQ     1005$
        CMP     #'SU,IOBUF
        BEQ     1005$
        MOV     #SD,YFUNC           ;SCALE DOWN
        CMP     #'SD,IOBUF
        BEQ     1005$
        MOV     #AS,YFUNC           ;AUTO-SCALE
        CMP     #'AS,IOBUF
        BEQ     1005$
        MOV     #CS,YFUNC           ;CHANNEL SCALE
        CMP     #'CS,IOBUF
        BEQ     1005$
        MOV     #WS,YFUNC           ;WINDOW SCALE
        CMP     #'WS,IOBUF
        BEQ     1005$
        BR      6$
1005$:  CALL    YAXIS               ;Y-AXIS SCALING
        JMP     OPTOUT              ;LEAVE
6$:     CMP     #'HA,IOBUF          ;HALT DATA TAKING
        BNE     7$
        CLR     ACCFLG
        JMP     OPTOUT              ;LEAVE
7$:     CMP     #'MA,IOBUF          ;SCAN THE SECTOR MAGNET
        BNE     8$
        CALL    DISHLT              ;HALT   DISPLA
        CALL    MAGNET
        JMP     OPTOUT              ;LEAVE
8$:     CMP     #'NS,IOBUF          ;NUMBER OF SCANS DESIRED
        BNE     9$
        CALL    NOSCNS              ;SET NUMBER OF SCANS TO BE MADE
        JMP     OPTOUT              ;LEAVE
9$:     CMP     #'IW,IOBUF          ;INTEGRATE WINDOW?
        BNE     10$
        CALL    IW
        JMP     OPTOUT
```

```
10$:    CMP     #'QU,IOBUF              ;QUADRUPOLE
        BNE     11$
        CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
        BEQ     1010$                   ;BRANCH IF UNLOCKED
        JMP     OPTLOC                  ;LOCKED! LEAVE WITH MESSAGE
1010$:  CALL    DISHLT                  ;HALT   DISPLA
        CALL    QUAD
        JMP     OPTOUT                  ;LEAVE
11$:    CMP     #'RD,IOBUF              ;READ THE DISK
        BNE     12$
        CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
        BEQ     1011$                   ;BRANCH IF UNLOCKED
        JMP     OPTLOC                  ;LOCKED! LEAVE WITH MESSAGE
1011$:  CALL    RDISK                   ;READ A DATA FILE
        JMP     OPTOUT                  ;LEAVE
12$:    CMP     #'RE,IOBUF              ;RESET THE DEVICE BEING SCANNED
        BNE     13$
        CALL    RESQUE
        JMP     OPTOUT                  ;LEAVE
13$:    CMP     #'SC,IOBUF              ;SCAN DEVICE
        BNE     14$
        CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
        BEQ     1013$                   ;BRANCH IF UNLOCKED
        JMP     OPTLOC                  ;LOCKED! LEAVE WITH MESSAGE
1013$:  CALL    SC
        JMP     OPTOUT                  ;LEAVE
14$:    CMP     #'GA,IOBUF              ;SET GAIN OF A2D CONVERTER?
        BNE     15$
        CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
        BEQ     1014$                   ;BRANCH IF UNLOCKED
        JMP     OPTLOC                  ;LOCKED! LEAVE WITH MESSAGE
1014$:  CALL    A2DAMP
        JMP     OPTOUT                  ;LEAVE
15$:    CMP     #'TR,IOBUF              ;SC TRAP
        BNE     16$
        CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
        BEQ     1015$                   ;BRANCH IF UNLOCKED
        JMP     OPTLOC                  ;LOCKED! LEAVE WITH MESSAGE
1015$:  CALL    TRAP
        JMP     OPTOUT                  ;LEAVE
16$:    CMP     #'WD,IOBUF              ;WRITE TO DISK
        BNE     17$
        CALL    WDISK
        JMP     OPTOUT                  ;LEAVE
17$:    CMP     #'CL,IOBUF              ;CLEAR...INIT
        BNE     18$
        CALL    CLRQUE
        CLR     UPDATE
        JMP     OPTOUT                  ;LEAVE
18$:    CMP     #'SP,IOBUF              ;CHANGE METHOD OF SCANNING?
        BNE     19$
        CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
        BEQ     1018$                   ;BRANCH IF UNLOCKED
        JMP     OPTLOC                  ;LOCKED! LEAVE WITH MESSAGE
1018$:  CALL    SCNPRO
```

```
        JMP     OPTOUT                  ;LEAVE
19$:    CMP     #'UP,IOBUF              ;UPDATE?
        BNE     20$
        CLR     UPDATE
        JMP     OPTOUT                  ;LEAVE
20$:    CMP     #'SM,IOBUF              ;SMOOTH DISPLAYED DATA?
        BNE     21$
        CALL    SMOOTH                  ;GET INPUT
        JMP     OPTOUT                  ;LEAVE
21$:    CMP     #'SR,IOBUF              ;SCROLL WINDOW RIGHT?
        BNE     22$
        MOV     #1,SLRFLG
        CALL    SLR
        JMP     OPTOUT
22$:    CMP     #'SL,IOBUF              ;SCROLL WINDOW LEFT?
        BNE     23$
        MOV     #-1,SLRFLG
        CALL    SLR
        JMP     OPTOUT
23$:    CMP     #'NO,IOBUF              ;CHANGE SCANNING CONDITION?
        BNE     24$
        CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
        BEQ     1022$                   ;BRANCH IF UNLOCKED
        JMP     OPTLOC                  ;LOCKED! LEAVE WITH MESSAGE
1022$:  CALL    DISHLT                  ;HALT   DISPLA
        CALL    NOTHIN
1023$:  JMP     OPTOUT                  ;LEAVE
24$:    CMP     #'IN,IOBUF              ;START PULSE SEQUENCE?
        BNE     25$
        .PRINT  #M55                    ;PULSE SEQUENCE INIT.
        .PRINT  #M48                    ;ARE YOU SURE?
        CALL    READ                    ;YES OR NO
        CMPB    #'Y,IOBUF
        BEQ     1024$
        JMP     OPTOUT
1024$:  MOV     #1,R1
        CALL    MOTOR2                  ;SEND ONE PULSE
        JMP     OPTOUT                  ;LEAVE
25$:    CMP     #'PR,IOBUF              ;PRINT OPTIONS
        BNE     26$
        CALL    PR
        JMP     OPTOUT                  ;LEAVE
26$:    CMP     #'SF,IOBUF              ;SCROLL FASTER
        BNE     27$
        CALL    SF
        JMP     OPTOUT
27$:    CMP     #'SS,IOBUF              ;SCROLL SLOWER
        BNE     28$
        CALL    SS
        JMP     OPTOUT
28$:    CMP     #'TY,IOBUF              ;TYPE TO CONSOLE
        BNE     30$
        CALL    TY
        JMP     OPTOUT                  ;LEAVE
30$:    CMP     #'NC,IOBUF              ;SET NUMBER OF DATA CYCLES PER CHANN
```

```
EL?
        BNE     31$
        CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
        BEQ     1030$                   ;BRANCH IF UNLOCKED
        JMP     OPTLOC                  ;LOCKED! LEAVE WITH MESSAGE
1030$:  CALL    NOCYCL
        JMP     OPTOUT                  ;LEAVE
31$:    CMP     #'FU,IOBUF              ;FOUL LANGUAGE USAGE?
        BNE     32$
        CMP     #'CK,IOBUF+2
        BNE     32$
        .PRINT  #M44                    ;WATCH YOUR LANGUAGE!!
        JMP     OPTOUT                  ;LEAVE
32$:    CMP     #'LO,IOBUF              ;LOCK MACHINE CONDITION?
        BNE     33$
        CMP     #'CK,IOBUF+2
        BNE     33$
        MOV     #0,LOCKEY               ;LOCK IT!
        .PRINT  #M52                    ;'LOCKED' MESSAGE
        JMP     OPTOUT                  ;LEAVE
33$:    CMP     LOCKEY+2,IOBUF          ;UNLOCK PROGRAM VARIABLES?
        BNE     34$
        MOV     IOBUF,LOCKEY            ;ENTER KEY
        .PRINT  #M53                    ;'UNLOCKED' MESSAGE
        JMP     OPTOUT                  ;LEAVE
34$:    CMP     #'HE,IOBUF              ;ENTER HEADER?
        BNE     35$
        CALL    HEADER
        JMP     OPTOUT                  ;LEAVE
35$:    CMP     #'QP,IOBUF              ;SCAN FROM PEAK TO PEAK?
        BNE     36$
        CMPB    #'Q,SCCODE              ;IS THE FIRST LETTER A 'Q'?
        BNE     36$                     ;BRANCH IF NO
        MOV     #'QP,SCCODE             ;CODE FOR PEAK TO PEAK SCANNING
        CALL    DISHLT                  ;HALT DISPLAY
        CALL    QUAD
        JMP     OPTOUT
36$:    CMP     #'MC,IOBUF              ;MCS UNIT
        BNE     37$
        CMP     LOCKEY,LOCKEY+2         ;ARE PROGRAM VARIABLES LOCKED?
        BEQ     360$                    ;BRANCH IF UNLOCKED
        JMP     OPTLOC                  ;LOCKED! LEAVE WITH MESSAGE
360$:   CALL    MCS
        JMP     OPTOUT                  ;LEAVE
37$:    MOV     #WL,DISCOM              ;ENTER COMMAND CODE
        CMP     #'WL,IOBUF              ;SHIFT WINDOW LEFT?
        BEQ     38$
        MOV     #WR,DISCOM
        CMP     #'WR,IOBUF              ;WINDOW SHIFT RIGHT
        BEQ     38$
        MOV     #WI,DISCOM
        CMP     #'WI,IOBUF              ;DEFINE INITIAL WINDOW CHAN.
        BEQ     38$
        MOV     #WA,DISCOM
        CMP     #'WA,IOBUF              ;DEFINE WINDOW APERTURE
```

```
        BEQ     38$
        MOV     #WB,DISCOM
        CMP     #'WB,IOBUF                      ;WINDOW BRIGHTNESS
        BEQ     38$
        MOV     #AB,DISCOM                      ;ACTIVE CHAN. BRT.
        CMP     #'AB,IOBUF
        BNE     39$
38$:    CALL    DISDAT
        JMP     OPTOUT
39$:    CMP     #'DH,IOBUF                      ;DISPLAY(DMA) HALT?
        BNE     40$
        CALL    DISHLT
        JMP     OPTOUT
40$:    CMP     #'PL,IOBUF                      ;FULL PLOT?
        BEQ     41$
        CMP     #'PP,IOBUF                      ;PLOT POINTS ONLY?
        BNE     42$
41$:    CALL    PLOT
        JMP     OPTOUT
42$:    CMP     #'PC,IOBUF                      ;PLOT CHANNELS ONLY?
        BNE     43$
        CALL    PLOTPC
        JMP     OPTOUT
43$:    CMP     #'PM,IOBUF                      ;MOVE PLOT
        BNE     44$
        CALL    PLOTPM
        JMP     OPTOUT
44$:    CMP     #'PH,IOBUF                      ;PLOT HORIZONTAL LINE?
        BNE     45$
        CALL    PLOTPH
        JMP     OPTOUT
45$:    CMP     #'PV,IOBUF                      ;PLOT VERTICAL LINE?
        BNE     46$
        CALL    PLOTPV
        JMP     OPTOUT
46$:    CMP     #'PB,IOBUF                      ;DRAW BOX ONLY
        BNE     47$
        CALL    PLOTIT
        JMP     OPTOUT
47$:    CMP     #'ND,IOBUF                      ;NEW DATE?
        BNE     48$
        .DATE
        MOV     R0,DATE                         ;SAVE
        CALL    DATES                           ;DECODE
        JMP     OPTOUT
48$:    CMP     #'ER,IOBUF                      ;SET RUN NUMBER?
        BNE     49$
        CALL    ER
        JMP     OPTOUT
49$:    CMP     #'RR,IOBUF                      ;RECALL PRESENT RUN #
        BNE     50$
        MOV     RUN+2,RUN
        JMP     OPTOUT
50$:    CMP     #'US,IOBUF                      ;UPDATE SCALE
        BNE     51$
```

```
        CALL    US
        JMP     OPTOUT
51$:    CMP     #'SZ,IOBUF          ;SHIFT ZERO?
        BNE     52$
        CALL    SZ                  ;SHIFT ZERO OF DISPLAY
        JMP     OPTOUT
52$:    CMP     #'ST,IOBUF          ;AUTO RESTART
        BNE     54$
        CMP     LOCKEY,LOCKEY+2     ;LOCKED?
        BEQ     53$
        JMP     OPTLOC              ;LOCKED!
53$:    CLR     ACCFLG              ;HALT
        CALL    RESET
        CALL    CLEAR
        CLR     UPDATE
        MOV     #1,ACCFLG           ;START ACCUM.
        JMP     OPTOUT
54$:    CMP     #'CU,IOBUF          ;CLOCK UNITS
        BNE     55$
        CALL    CU
        JMP     OPTOUT
55$:    CMP     #'RT,IOBUF          ;RE-TAKE MODE?
        BNE     56$
        CMP     LOCKEY,LOCKEY+2     ;LOCKED?
        BEQ     550$
        JMP     OPTLOC              ;OPTION LOCKED!
550$:   TST     RETAKE
        BEQ     555$
        CLR     RETAKE
        .PRINT  #M77                ;ACCUM. MODE
        JMP     OPTOUT
555$:   INC     RETAKE
        .PRINT  #M76                ;RE-TAKE MODE MSG.
        JMP     OPTOUT
56$:    CMP     #'KM,IOBUF          ;KEY MODE
        BNE     59$
        CALL    KM
        JMP     OPTOUT
59$:    CMP     #'SE,IOBUF          ;SET DEVICE
        BNE     60$
        CMP     LOCKEY,LOCKEY+2     ;ARE PROGRAM VARIABLES LOCKED?
        BEQ     590$                ;BRANCH IF UNLOCKED
        JMP     OPTLOC              ;LOCKED! LEAVE WITH MESSAGE
590$:   CALL    SE
        JMP     OPTOUT              ;LEAVE
60$:    .PRINT  #M51                ;NONEXISTENT OPTION
        JMP     OPTOUT              ;LEAVE
OPTLOC: .PRINT  #M50                ;OPTION LOCKED MESSAGE
OPTOUT: CALL    CRLF
        MOV     #'>,R0
        .TTYOUT
        JMP     DISPLA
;
;
;
```

```
;       THIS IS THE INTERRUPT SERVICE ROUTINE SECTION
;
;
DATAIN: MOV     R1,R1DATA               ;SAVE R1
        MOV     R4,R4DATA               ;% R4
        CMP     #1,ACCFLG               ;ACCUMULATE??
        BEQ     1$                      ;BRANCH IF YES
        JMP     DATOUT                  ;DON'T ACCUMULATE!!!
1$:     MOVB    #1,@#166040             ;READ MOTOR1 STATUS WORD
        BIT     #1,DATALO               ;IS MOTOR1 LAM SET?
        BNE     5$                      ;BRANCH IF NOT BUSY
        JMP     DATOUT                  ;BUSY--BRANCH OUT
5$:     CMP     DACYCL,#1               ;CYCLE COUNTDOWN OVER?
        BLE     8$                      ;BRANCH IF YES
        DEC     DACYCL                  ;DECREMENT CYCLE COUNTER
        MOVB    #0,A2DCON               ;SET LAM DOWN
        JMP     CYCOUT                  ;NO DATA READ
8$:     MOV     DACYCL+2,DACYCL         ;RE-LOAD CYCLE COUNTER
        TST     UPDATE                  ;IF NOT ZERO,DEC. UPDATE
        BEQ     9$
        DEC     UPDATE
9$:     MOV     SBBUF,R4                ;S+B BUFFER POINTER TO R4
        ADD     BUFOFF,R4               ;ADD OFFSET TO THE POINTER
        MOVB    #0,SCAL0                ;READ SCALER ZERO
        MOVB    #2,SCAL0
        ADD     DATALO,(R4)+            ;ADD DATA VALUE
        ADC     (R4)
        ADD     DATAHI,(R4)
        MOV     BABUF,R4                ;PLACE BAC-GRD POINTER INTO R4
        ADD     BUFOFF,R4               ;ADD IN BUFFER OFFSET
        MOVB    #0,SCAL2                ;READ SCALER TWO
        MOVB    #2,SCAL2
        ADD     DATALO,(R4)+            ;ADD VALUE TO DATA
        ADC     (R4)
        ADD     DATAHI,(R4)
        MOV     POBUF,R4                ;PLACE POWER BUF. POINTER IN R4
        ADD     BUFOFF,R4               ;ADD BUFFER OFFSET
        MOVB    #0,A2DCON               ;READ A-2-D CONVERTER,CLEAR LAM
        ADD     DATALO,(R4)+            ;ADD VALUE TO BUFFER
        ADC     (R4)
        CMP     DIRFLG,#1               ;SCANNING WHICH WAY?
        BNE     30$
        CMP     LCHAN,BUFOFF            ;AT LAST CHANNEL YET?
        BNE     20$                     ;BRANCH IF NO
        CMP     #'BI,SCTYPE             ;BIDIRECTIONAL SCANNING??
        BNE     10$                     ;BRANCH IF NO
        MOV     #-1,DIRFLG
        JMP     CYCOUT                  ;LEAVE
10$:    CLR     DIRFLG
        CLR     BUFOFF                  ;SET BUFFER OFFSET TO ZERO
        CLR     UPDATE                  ;UPDATE DISPLAY
        INC     NSCAN                   ;KEEP TRACK OF NUMBER OF SCANS COMPL
ETEDD
        CMP     NSCAN,NSCAN+2           ;FINISHED YET?
        BMI     15$                     ;BRANCH IF NO
```

```
        CLR     ACCFLG                  ;STOP ACCUMULATING DATA
        CLR     RUNFLG                  ;MESSAGE FLAG
        INC     RUN                     ;NUMBER OF RUNS FOR THE DAY
        MOV     RUN,RUN+2               ;COPY
15$:    CALL    @SCFUNC                 ;CALL SCAN FUNCTION
        JMP     CYCOUT                  ;LEAVE
20$:    ADD     #4,BUFOFF               ;BUF OFFSET TO NEXT CHANNEL
        CALL    @SCFUNC
        JMP     CYCOUT
30$:    CMP     BUFOFF,#0               ;FIRST DATA CHANNEL?
        BNE     40$                     ;BRANCH IF NO
        CLR     UPDATE                  ;UPDATE DISPLAY
        INC     NSCAN                   ;INCREMENT SCAN COUNTER
        CMP     NSCAN,NSCAN+2           ;TAKE MORE SCANS?
        BMI     35$                     ;BRANCH IF NO
        CLR     ACCFLG                  ;STOP ACCUMULATING DATA!
        CLR     RUNFLG                  ;RUN COMPLETED MESSAGE FLAG
        INC     RUN                     ;NUMBER OF RUNS COMPLETED
        MOV     RUN,RUN+2
35$:    MOV     #1,DIRFLG               ;GO FORWARD NOW
        JMP     CYCOUT                  ;LEAVE
40$:    SUB     #4,BUFOFF
50$:    CALL    @SCFUNC                 ;GO TO FUNCTION
        BR      CYCOUT
;
DATOUT: MOVB    #9.,SCAL0               ;CLEAR SCALER ZERO
        MOVB    #9.,SCAL2               ;CLEAR SCALER TWO
        MOVB    #0,A2DCON               ;LAM DOWN
CYCOUT: BIS     #300,STATU1             ;SET INE AND REIN
        MOV     R1DATA,R1               ;UNSAVE R1
        MOV     R4DATA,R4               ;& R4
        BIT     #1,SCMODE               ;WHICH MODE?
        BNE     10$
        RTI                             ;RETURN FROM INTERRUPT
10$:    RETURN                          ;SUBROUTINE RETURN
;
;DISPLAY DMA INTERRUPT SERVICE ROUTINE
;
;
DMAINT: MOV     R0,R0DMA                ;SAVE REGS
        MOV     R1,R1DMA
        MOV     R2,R2DMA
        MOV     R3,R3DMA
        .INTEN  DMAPRI
        MOV     R0DMA,R0                ;UNSAVE
        MOV     R1DMA,R1
        MOV     R2DMA,R2
        MOV     R3DMA,R3
        TST     EXPFLG                  ;EXPAND DISPLAY?
        BEQ     10$                     ;BRANCH IF YES
        DEC     WINFLG                  ;WINDOW BRIGHTNESS
        BGE     10$                     ;BRANCH IF YES
        DEC     ACTIVE                  ;ACTIVE CHAN. BR'TNESS
        BGE     20$
        MOV     DIBUF,DMACAR            ;LOAD DMA REGISTERS
```

```
        MOV     NCHAN,DMAPCR
        NEG     DMAPCR                          ;2'S COMPLEMENT
        MOV     ACTIVE+2,ACTIVE                 ;ACTIVE CHAN. BRT.
        MOV     WINBRT,WINFLG
        BR      DMAOUT
10$:    MOV     DISWIN,DMACAR                   ;DISPLAY WINDOW VALUES
        MOV     WINSIZ,DMAPCR
        BR      30$
20$:    MOV     BUFOFF,DMACAR                   ;ACTIVE CHAN. POSITION
        ADD     DIBUF,DMACAR
        MOV     #1,DMAPCR
30$:    NEG     DMAPCR                          ;2'S COMPLEMENT
DMAOUT: TST     DMAHLT                          ;HALT DMA?
        BNE     10$                             ;BR IF NO
        CLR     ADCSR
        RTS     PC                              ;RETURN FROM INTERRUPT
10$:    BIS     #102,ADCSR                      ;START DMA
        BIS     #1,ADCSR
        RTS     PC                              ;RETURN FROM INTERRUPT
;
;
;
CLKINT: BIS     #20000,KWCSR
        RTI                                     ;RETURN FROM INTERRUPT
;
;
;MCS INTERRUPT SERVICE ROUTINE
;
MCSINT: MOV     R1,R1DATA                       ;SAVE REGS
        MOV     R4,R4DATA
        BIC     #1,STATU1                       ;CLEAR DII BIT
        MOV     MCSA2,OVF                       ;READ OVFL AND STOPRUN
        BIT     #3,OVF                          ;CHECK FOR OVFL
        BEQ     1$
        JMP     MCSOUT
1$:     MOV     MCSBUF,R1
        MOV     #377,R4                         ;255 CANNELS
2$:     ADD     @#MCSA0,(R1)+                   ;TRANSFER DATA TO LSI-11 DATA BUFFER
        ADC     (R1)+
        SOB     R4,2$
        CLR     UPDATE                          ;UPDATE DISPLAY
        BIS     #1,STATU1                       ;SET DII BIT
        MOVB    #9.,MCSA0                       ;F9A0 CLEAR RAM
        INC     NSCAN                           ;NUMBER OF 'SWEEPS'
        MOV     #100.,R4                        ;WAIT LOOP FOR RAM CLEAR
3$:     SOB     R4,3$
        BIT     #4,OVF                          ;? IS RUN FINISHED
        BNE     MCSOUT                          ;YES, THEN RETURN
        MOVB    #26.,MCSA6                      ;NO, SET ACCUMULATE
MCSOUT: MOV     R1DATA,R1
        MOV     R4DATA,R4
        BIS     #300,STATU1                     ;SET INE,REIN
        BIT     #1,SCMODE                       ;WHICH MODE?
        BNE     10$
        RTI                                     ;RETURN FROM INTERRUPT
```

```
10$:    RETURN                          ;SUBROUTINE RETURN
;
;HERE ARE ALL THE FUNCTIONS FOR A2DCON INTERRUPT
;
LA:     MOV     NSTEP,R1                ;STEP THE DYE LASER MOTOR
        CMP     DIRFLG,#1               ;DETERMINE WHICH WAY
        BNE     10$
        CALL    MOTOR1                  ;CLOCKWISE
        ADD     WAVINC,SETWAV           ;NEW WAVELENGTH SETTING
        ADC     SETWAV+2
        ADD     WAVINC+2,SETWAV+2
        RETURN
10$:    CMP     DIRFLG,#0               ;RESET THE DYE LASER?
        BEQ     20$
        NEG     R1
        CALL    MOTOR1                  ;COUNTER CLOCKWISE
        SUB     WAVINC,SETWAV           ;NEW WAVELENGTH SETTING
        SBC     SETWAV+2
        SUB     WAVINC+2,SETWAV+2
        RETURN
20$:    MOV     SHOWAV,RESULT
        MOV     SHOWAV+2,RESULT+2
        CALL    DYESET
        MOV     #1,DIRFLG
        RETURN
;
;
MA:     CMP     DIRFLG,#1               ;SECTOR MAGNET--SCANNING WHICH WAY?
        BNE     10$
        ADD     MARES,MADAC             ;SCAN FORWARD
        RETURN
10$:    CMP     DIRFLG,#0               ;RESET THE MAGNET?
        BEQ     20$
        SUB     MARES,MADAC             ;SCAN BACKWARDS
        RETURN
20$:    MOV     MADAC1,MADAC            ;RESET MAGNET
        MOV     #1,DIRFLG
        RETURN
;
;
QU:     CMP     DIRFLG,#1               ;QUADRUPOLE--SCANNING WHICH WAY?
        BNE     10$
        ADD     MRES,MDAC               ;SCAN FORWARD
        RETURN
10$:    CMP     DIRFLG,#0               ;RESET THE QUADRUPOLE?
        BEQ     20$
        SUB     MRES,MDAC               ;SCAN BACKWARDS
        RETURN
20$:    MOV     MDAC1,MDAC              ;RESET QUAD
        MOV     #1,DIRFLG
        RETURN
;
;
TR:     CMP     DIRFLG,#1               ;SCAN ION TRAP
        BNE     10$
```

```
        ADD     DELINC,DELTOT
        MOV     DELTOT,KWBPR
        NEG     KWBPR
        RETURN
10$:    CMP     DIRFLG,#0               ;RESET THE ION TRAP?
        BEQ     20$
        SUB     DELINC,DELTOT           ;BACKWARDS
        MOV     DELTOT,KWBPR
        NEG     KWBPR
        RETURN
20$:    MOV     DELOFF,DELTOT           ;RESET
        MOV     DELTOT,KWBPR
        NEG     KWBPR
        MOV     #1,DIRFLG               ;SET FOR FORWARD SCANNING
        RETURN
;
;
NO:     CMP     DIRFLG,#1               ;NO OPERATION!!
        BNE     10$                     ;SCAN FORWARD
        RETURN
10$:    CMP     DIRFLG,#0               ;RESET THE SCAN POINTERS?
        BEQ     20$
        RETURN
20$:    MOV     #1,DIRFLG               ;SET SCAN DIRECTION FLAG:FORWARD
        RETURN
;
;
QP:     MOV     BUFOFF,R4               ;BUFFER OFFSET TO R4
        ASR     R4                      ;DIVIDE BY TWO
        ADD     #MASBUF,R4              ;R4 HAS MASS POINTER
        CMP     DIRFLG,#1               ;QUADRUPOLE--SCANNING WHICH WAY?
        BNE     10$
        MOV     (R4),MDAC               ;SET QUADRUPOLE TO MASS PEAK
        RETURN
10$:    CMP     DIRFLG,#0               ;RESET THE QUADRUPOLE?
        BEQ     20$
        MOV     (R4),MDAC               ;SET QUADRUPOLE TO MASS PEAK
        RETURN
20$:    MOV     (R4),MDAC               ;RESET QUAD
        MOV     #1,DIRFLG
        RETURN
;
;
;END PAGE ONE
```

```
;PAGE TWO
;
;
;SUBROUTINE TO HALT DMA DRIVEN DISPLAY
;
DISHLT: CLR     DMAHLT                  ;HALT DMA
10$:    BIT     #100,ADCSR              ;HALTED?
        BNE     10$                     ;LOOP TILL YES
        RETURN
;
;
;
;SUBROUTINE 'BUFINIT' GENERATES DATA BUFFER ADDRESSES GIVEN THE
;NUMBER OF DATA CHANNELS IN USE.
;
BUFINIT:MOV     R1,R1BUF                ;SAVE REGISTERS
        MOV     R2,R2BUF
        MOV     NCHAN,R1                ;R1 GETS # OF CHANNELS
        MOV     NCHAN,DICHNS            ;NUMBER OF DISPLAYED CHANNELS
        CMP     NCHAN,WINSIZ            ;WINDOW TOO LARGE?
        BPL     10$
        MOV     NCHAN,WINSIZ            ;MAX WINDOW SIZE
10$:    ASH     #2,R1                   ;MULTIPLY BY FOUR
        SUB     #4,R1
        MOV     R1,LCHAN                ;# OF ADDRESS LOCATIONS TO LAST CHAN
NEL
        MOV     #DATBUF,R2              ;STARTING ADDRESS OF DATA BUFFER IN
R2
        ADD     #1000.,R2               ;500 HEADER WORDS
        MOV     R2,SBBUF                ;STARTING LOC. FOR SIGNAL+BACKGROUND
 BUF.
        ADD     #4,R1                   ;# OF ADDRESSES IN R1
        ADD     R1,R2                   ;STARTING LOC. FOR BACKGROUND BUFFER
        MOV     R2,BABUF                ;LOAD BACKGROUND BUFFER POINTER
        ADD     R1,R2                   ;STARTING LOCATION FOR POWER BUFFER
        MOV     R2,POBUF                ;LOAD POWER BUFFER POINTER
        ADD     R1,R2                   ;STARTING LOCATION FOR DISPLAY BUFFE
R
        MOV     R2,DIBUF                ;LOAD DISPLAY BUFFER POINTER
        MOV     R2,DISTRT               ;STARTING LOCATION FOR DISPLAY
        MOV     R2,DISWIN               ;WINDOW POINTER
        ADD     R1,R2
        MOV     R2,DISMAX               ;DISPLAY BUFFER LAST ADDRESS +2
        MOV     #1,EXPFLG               ;DON'T EXPAND DISPLAY
        MOV     #1,DIRFLG               ;SCAN FORWARD FIRST
        CLR     BUFOFF                  ;CLEAR BUFFER OFFSET
        CALL    XJUMP                   ;CALCULATE XJUMP
        CLR     UPDATE                  ;UPDATE DISPLAY
        BIC     #2,RUNFLG               ;RUN NOT IN PROGRESS
        MOV     R1BUF,R1                ;UNSAVE REGISTERS
        MOV     R2BUF,R2
        RETURN                          ;RETURN TO CALLER
;
;
;
```

```
;THIS SUBROUTINE LOADS 'DIFUNC' AND 'DICODE' WHICH ARE WORDS THAT DIRECT
;THE MAIN PROGRAM MODULE DISPLAY ROUTINE.
;
DISOPT: MOV     IOBUF+2,IOBUF           ;MOVE IN SECOND COMMAND IF ANY
        CMPB    IOBUF,#15               ;IS IT A CARRIAGE RETURN?
        BEQ     1$                      ;IF YES,ASK FOR SECOND PART OF COMMA
ND
        BR      5$                      ;GO CHECK THE COMMAND
1$:     .PRINT  #M10                    ;WHICH OPTION?
        CALL    READ
5$:     MOV     IOBUF,DICODE            ;STORE RESPONSE IN DISPLAY CODE
        CMP     #'SB,IOBUF              ;DISPLAY SIGNAL+BACKGRD.?
        BEQ     15$                     ;BRANCH IF YES
        CMP     #'SI,IOBUF              ;DISPLAY SIGNAL BUFFER?
        BEQ     20$                     ;BRANCH IF YES
        CMP     #'BA,IOBUF              ;DISPLAY THE BACKGRD. BUFFER?
        BEQ     30$                     ;BRANCH IF YES
        CMP     #'PO,IOBUF              ;DISPLAY THE POWER BUFFER?
        BEQ     40$                     ;BRANCH IF YES
        CMPB    #'?,IOBUF               ;PRINT OPTIONS?
        BEQ     50$                     ;BRANCH IF YES
        MOV     DICODE+2,DICODE         ;INSERT OLD CODE
        CMP     #'DA,IOBUF              ;DISPLAY DATE?
        BNE     6$
        CALL    TYDATE
        BR      DISOUT
6$:     CMP     #'PA,IOBUF              ;DISPLAY PARAMETERS?
        BNE     8$
        CALL    DP
        BR      DISOUT
8$:     CMP     #'EX,IOBUF              ;EXPAND DISPLAY?
        BNE     9$
        CALL    DE
        BR      DISOUT
9$:     CMP     #'PN,IOBUF              ;POWER NORMALIZE?
        BNE     10$
        CALL    PN
        BR      DISOUT
10$:    CMP     #'RE,IOBUF              ;DISPLAY REVERSE?
        BNE     DISOUT
        CALL    RE
        BR      DISOUT
15$:    MOV     #SB,DIFUNC              ;LOAD FUNCTION CODE WORD
        BR      60$                     ;GET OUT
20$:    MOV     #SI,DIFUNC
        BR      60$                     ;GET OUT
30$:    MOV     #BA,DIFUNC
        BR      60$                     ;GET OUT
40$:    MOV     #PO,DIFUNC
        BR      60$                     ;GET OUT
50$:    .PRINT  #M7                     ;PRINT OPTIONS
        .PRINT  #M11
        CALL    CRLF
        JMP     1$                      ;TRY AGAIN
60$:    MOV     DIFUNC,DIFUNC+2         ;SAVE FUNCTION CODE
```

```
        MOV     DICODE,DICODE+2         ;SAVE NEW DISPLAY CODE
        CLR     UPDATE                  ;UPDATE DISPLAY
DISOUT: RETURN                          ;RETURN TO CALLER
;
;
;SUBROUTINE 'DISCAL' SCALES THE NUMBERS IN THE DISPLAY BUFFER SO
;THAT THE LARGEST VALUE RESULTS IN FULL SCALE DEFLECTION OF THE
;DISPLAY SCOPE
;
DISCAL: CALL    DISHLT                  ;HALT DISPLAY
        MOV     R1,R1DIS                ;SAVE THE REGISTERS
        MOV     R2,R2DIS
        MOV     R3,R3DIS
        MOV     R4,R4DIS
        MOV     R5,R5DIS
        MOV     @#DIBUF,R1              ;STARTING ADDRESS OF BUFFER IN R1
        CMP     FLGDIS,#1               ;FORWARD OR REVERSE?
        BLE     10$
        ADD     LCHAN,R1                ;DISPLAY BUFFER MAXIMUM
10$:    MOV     NCHAN,R4                ;NUMBER OF CHANNELS IN R4
        JMP     @DIFUNC                 ;GO TO DISPLAY FUNCTION
;
;DISPLAY SIGNAL---S+B BUFFER MINUS B BUFFER=SIGNAL BUFFER
SI:     MOV     @#BABUF,R3              ;ADDRESS OF BACKGRD. BUFFER
        MOV     @#SBBUF,R2              ;ADDRESS OF S+B BUFFER
        MOV     @#POBUF,R0              ;ADDRESS OF POWER BUF.
        BIT     #1,FLGDIS               ;POWER NORM.?
        BNE     30$
        BIT     #2,FLGDIS               ;FORWARD OR REVERSE
        BNE     20$
10$:    MOV     (R2)+,(R1)              ;MOVE DOUBLE WORD
        MOV     (R2)+,2(R1)
        SUB     (R3)+,(R1)+             ;DO SUTRACTION
        SBC     (R1)
        SUB     (R3)+,(R1)+
        SOB     R4,10$                  ;DO THEM ALL
        JMP     @#SCALE
20$:    MOV     (R2)+,(R1)
        MOV     (R2)+,2(R1)
        SUB     (R3)+,(R1)
        SBC     2(R1)
        SUB     (R3)+,2(R1)
        SUB     #4,R1
        SOB     R4,20$
        JMP     @#SCALE
30$:    MOV     R1,R5SMO
        MOV     NCHAN,SMOPTR
        MOV     @#POBUF,R1              ;POWER BUFFER ADDRESS
        CALL    MAXFND
        MOV     @#ARG2,@#POWMAX
        MOV     @#ARG2+2,@#POWMAX+2
        TST     @#POWMAX                ;IS IT ZERO?
        BNE     35$
        MOV     #1,@#POWMAX
35$:    MOV     R5SMO,R1                ;RESTORE DIBUF ADDRESS IN R1
```

```
        MOV     @#BABUF,R3              ;ADDRESS OF BACKGRD. BUFFER
        MOV     @#SBBUF,R2              ;ADDRESS OF S+B BUFFER
        MOV     @#POBUF,R0              ;ADDRESS OF POWER BUF.
        MOV     NCHAN,R4
40$:    MOV     (R2)+,@#ARG1
        MOV     (R2)+,@#ARG1+2
        SUB     (R3)+,@#ARG1
        SBC     @#ARG1+2
        SUB     (R3)+,@#ARG1+2
        MOV     @#POWMAX,@#ARG2
        MOV     @#POWMAX+2,@#ARG2+2
        MOV     (R0)+,@#POVALU
        MOV     (R0)+,@#POVALU+2
        MOV     #LIST,R5                ;ARG. LIST
        MOV     R0,R0SMO                ;SAVE
        MOV     R1,R1SMO                ;SAVE
        MOV     R2,R2SMO                ;SAVE
        MOV     R3,R3SMO                ;SAVE
        MOV     R4,R4SMO                ;SAVE
        CALL    JMUL
        CMP     R0,#-2                  ;OVERFLOW?
        BNE     50$
        .PRINT  #M68                    ;OV FLO!!
50$:    MOV     @#RESULT,@#ARG1
        MOV     @#RESULT+2,@#ARG1+2
        MOV     @#POVALU,@#ARG2
        MOV     @#POVALU+2,@#ARG2+2
        TST     @#ARG2                  ;ZERO?
        BNE     53$
        MOV     #1,@#ARG2
53$:    MOV     #LIST,R5
        CALL    JDIV
        MOV     R4SMO,R4                ;UNSAVE
        MOV     R3SMO,R3
        MOV     R2SMO,R2
        MOV     R1SMO,R1
        MOV     R0SMO,R0
        MOV     @#RESULT,(R1)+          ;LOAD DIBUF
        MOV     @#RESULT+2,(R1)+
        BIT     #2,FLGDIS               ;FOR. OR REV.?
        BEQ     55$
        SUB     #8.,R1
55$:    DEC     R4                      ;LOOP COUNTER
        BEQ     60$
        JMP     40$
60$:    JMP     @#SCALE
;
;DISPLAY SIGNAL+BACKGROUND BUFFER
SB:     MOV     @#SBBUF,R2              ;ADDRESS OF S+B BUFFER
        BR      MW
;
;DISPLAY THE POWER BUFFER
PO:     MOV     @#POBUF,R2              ;ADDRESS OF POWER BUFFER
        BR      MW
;
```

```
;DISPLAY THE BACKGROUND BUFFER
BA:     MOV     @#BABUF,R2              ;ADDRESS OF BACKGROUND BUFFER
        BR      MW
;
;DISPLAY THE SMOOTHED BUFFER(I.E. DON'T SCALE AND DON'T UPDATE)
SM:     JMP     @#SCAOUT                ;NO SCALING --GET OUT
;
;
MW:     BIT     #2,FLGDIS               ;WHICH WAY?
        BEQ     MOVDAT                  ;FORWARD
        BR      DATMOV                  ;REVERSE
;
;
MOVDAT: MOV     (R2)+,(R1)+             ;FORWARD
        MOV     (R2)+,(R1)+
        SOB     R4,MOVDAT               ;TRANSFER UNTIL DONE
        JMP     @#SCALE
;
;
DATMOV: MOV     (R2)+,(R1)              ;REVERSE
        MOV     (R2)+,2(R1)
        SUB     #4,R1
        SOB     R4,DATMOV               ;TRANSFER UNTIL DONE
        JMP     @#SCALE
;
;
MINFND: MOV     #177777,@#ARG2          ;SEARCH FOR THE SMALLEST
        MOV     #77777,@#ARG2+2         ;VALUE IN THE ARRAY
10$:    MOV     (R1)+,@#ARG1
        MOV     (R1)+,@#ARG1+2
        MOV     #LIST,R5                ;SYSF4  ARG. LIST
        MOV     R1,@#R1SMO              ;SAVE R1
        CALL    JCMP
        MOV     @#R1SMO,R1              ;UNSAVE R1
        CMP     #-1,R0
        BNE     15$
        MOV     @#ARG1,@#ARG2
        MOV     @#ARG1+2,@#ARG2+2
15$:    DEC     @#SMOPTR                ;DEC. LOOP COUNTER
        BNE     10$
        RETURN
;
;
MAXFND: MOV     #0,@#ARG2               ;SEARCH FOR THE LARGEST
        MOV     #0,@#ARG2+2             ;ABSOLUTE VALUE IN THE ARRAY
10$:    MOV     (R1)+,@#ARG1
        MOV     (R1)+,@#ARG1+2
        BGE     20$
        COM     @#ARG1+2
        COM     @#ARG1
        ADD     #1,@#ARG1
        ADC     @#ARG1+2
20$:    MOV     #LIST,R5                ;SYSF4  ARG. LIST
        MOV     R1,@#R1SMO              ;SAVE R1
        CALL    JCMP
```

```
        MOV     @#R1SMO,R1              ;UNSAVE R1
        CMP     #1,R0
        BNE     25$
        MOV     @#ARG1,@#ARG2
        MOV     @#ARG1+2,@#ARG2+2
25$:    DEC     @#SMOPTR                ;DEC. LOOP COUNTER
        BNE     10$
        RETURN
;
;
;DISPLAY BUFFER HAS BEEN LOADED--NOW SCALE THEM FOR OUTPUT.
SCALE:  TST     RETAKE                  ;ACCUMULATING?
        BEQ     1$
        CALL    CLEAR                   ;CLEAR DATA
1$:     CMP     #1,ZERFLG               ;WHICH ZERO?
        BLT     20$                     ;YMIN = 'ENTERED VALUE'
        BGT     MAXIE                   ;YMIN = 0
        MOV     @#DISTRT,R1             ;DISPLAY START
        MOV     @#DICHNS,@#SMOPTR
        CALL    MINFND                  ;FIND MINIMUM
        MOV     @#ARG2,@#YMIN           ;MOVE BUFMIN
        MOV     @#ARG2+2,@#YMIN+2
20$:    MOV     @#YMIN,R3
        MOV     @#YMIN+2,R2
        MOV     @#DIBUF,R1
        MOV     NCHAN,R4
25$:    SUB     R3,(R1)+
        SBC     (R1)
        SUB     R2,(R1)+
        SOB     R4,25$
MAXIE:  MOV     @#DIBUF,R1              ;ADDRESS OF DISPLAY BUFFER
        CMP     DISFLG,#1               ;AUTO-SCALE OR WINDOW SCALE?
        BEQ     6$
        BLT     4$
        CMP     DISFLG,#2               ;CHANNEL OR ABSOLUTE SCALING?
        BNE     30$
        ADD     CSCHAN,R1               ;CHANNEL SCALE
        MOV     (R1)+,YMAX
        MOV     (R1),YMAX+2
        BR      30$
4$:     MOV     @#NCHAN,@#SMOPTR        ;AUTO SCALE
        BR      8$
6$:     MOV     DISWIN,R1               ;WINDOW SCALE
        MOV     WINSIZ,SMOPTR
8$:     CALL    MAXFND                  ;GET MAX. OF ARRAY
        MOV     @#ARG2,@#YMAX           ;MOVE BUFMAX
        MOV     @#ARG2+2,@#YMAX+2
30$:    MOV     @#YMAX,R3
        MOV     @#YMAX+2,R2
        CLR     @#SHIFT
        CMP     #0,R2
        BNE     35$
        MOV     #15.,@#SHIFT
35$:    CMP     #0,R3                   ;LOW WORD ZERO?
        BNE     37$
```

```
        MOV     #1,R3
        MOV     #30.,@#SHIFT
        BR      40$
37$:    SWAB    R3
        BNE     38$
        MOV     #23.,@#SHIFT
38$:    SWAB    R3
40$:    ASHC    @#SHIFT,R2
        BIT     #40000,R2
        BNE     43$
        MOV     @#YMAX,R3
        MOV     @#YMAX+2,R2
        INC     @#SHIFT
        BR      40$
43$:    MOV     R3,@#ARG1
        MOV     R2,@#ARG1+2
        MOV     @#YSCALE,ARG2
        CLR     @#ARG2+2
        MOV     #LIST,R5
        CALL    JDIV
        MOV     @#RESULT,@#ARG2
        MOV     @#RESULT+2,@#ARG2+2
        MOV     @#DISTRT,R1
        MOV     @#DICHNS,@#SMOPTR
45$:    MOV     (R1)+,R3
        MOV     (R1)+,R2
        ASHC    @#SHIFT,R2
        BVS     47$                     ;SIGN CHANGE?
        MOV     R3,@#ARG1
        MOV     R2,@#ARG1+2
        MOV     #LIST,R5
        MOV     R1,@#R1SMO              ;SAVE R1
        CALL    JDIV
        MOV     @#R1SMO,R1              ;UNSAVE R1
        ADD     #3777,@#RESULT
        CMP     #7777,@#RESULT
        BPL     50$
47$:    MOV     #7777,@#RESULT
50$:    MOV     @#RESULT,-2(R1)
        DEC     SMOPTR                  ;DEC. LOOP COUNTER
        BNE     45$
SCAOUT: CALL    XLOAD                   ;LOAD X-VALUES
        MOV     @#R1DIS,R1              ;UNSAVE THE REGISTERS
        MOV     @#R2DIS,R2
        MOV     @#R3DIS,R3
        MOV     @#R4DIS,R4
        MOV     @#R5DIS,R5
        MOV     #1,DMAHLT               ;DMA GO
        MOV     DIBUF,DMACAR            ;LOAD DMA REGISTERS
        MOV     NCHAN,DMAPCR
        NEG     DMAPCR                  ;2'S COMPLEMENT
        BIS     #102,ADCSR              ;ENABLE DMA INT
        BIS     #1,ADCSR                ;START DMA
        RETURN                          ;RETURN TO CALLER
;
```

```
;
;
;THIS SUBROUTINE DEFINES AND LOADS CODE VALUES INTO THE WORDS
;'SCTYPE' AND 'SCFUNC'. SCTYPE DETERMINES THE 'SCAN TYPE' TO
;BE PERFORMED DURING DATA TAKING--BIDIRECTIONAL OR UNIDIRECTIONAL.
;SCFUNC DETERMINES THE 'SCAN FUNCTION';THAT IS ,WHICH DEVICE IS
;TO BE SCANNED--THE DYE LASER,THE OPO,ETC..
;
SC:     CALL    DISHLT                  ;HALT   DISPLAY
        MOV     #1,DIRFLG               ;SCAN DIRECTION FLAG
        MOV     IOBUF+2,IOBUF           ;MOV IN SECOND COMMAND,IF ANY
        CMPB    IOBUF,#15               ;IS IT A CARRIAGE RETURN?
        BEQ     1$                      ;IF YES,ASK FOR SECOND PART OF COMMA
ND
        BR      5$                      ;WHAT COMMAND IS IT?
1$:     .PRINT  #M5
        CALL    READ
5$:     MOV     IOBUF,SCCODE
        CMP     #'LA,IOBUF              ;SCAN THE LASER?
        BEQ     10$
        CMP     #'QU,IOBUF              ;SCAN THE QUADRUPOLE?
        BEQ     20$
        CMP     #'TR,IOBUF              ;SCAN THE ION TRAP?
        BEQ     30$
        CMP     #'MA,IOBUF              ;SCAN THE MAGNET?
        BEQ     40$
        CMP     #'MC,IOBUF              ;THE MCS UNIT?
        BEQ     50$
        CMP     #'NO,IOBUF              ;SCAN NOTHING?
        BEQ     60$
        CMPB    #'?,IOBUF               ;PRINT OPTIONS AVAILABLE?
        BEQ     SC05

        .PRINT  #M4
        BR      1$
10$:    MOV     #LA,SCFUNC
        CALL    DYELAS                  ;DYE LASER SUBROUTINE
        CALL    BUFINIT                 ;GENERATE NEW BUFFER POINTERS
        CALL    CLEAR                   ;CLEAR DATA BUFFER
        JMP     SCOUT
20$:    MOV     #QU,SCFUNC
        CALL    QUAD                    ;CALL QUADRUPOLE SUBROUTINE
        CALL    BUFINIT                 ;GENERATE NEW BUFFER POINTERS
        CALL    CLEAR                   ;CLEAR DATA BUFFER
        JMP     SCOUT
30$:    MOV     #TR,SCFUNC
        CALL    TRAP
        JMP     SCOUT                   ;LEAVE
40$:    MOV     #MA,SCFUNC              ;SCAN SECTOR MAGNET
        CALL    MAGNET                  ;CALL MAGNET SUBROUTINE
        CALL    BUFINIT                 ;GENERATE NEW BUFFER POINTERS
        CALL    CLEAR                   ;CLEAR DATA BUFFER
        JMP     SCOUT
50$:    .PRINT  #M98                    ;ACTIVATE MCS UNIT?
        .PRINT  #M48                    ;ARE YOU SURE?
```

```
        CALL    READ                    ;GET RESPONSE
        CMPB    #'Y,IOBUF               ;IS IT YES?
        BNE     SCOUT                   ;IF NO,GET OUT
        MOV     #377,NCHAN              ;255. CHANNELS
        CALL    BUFINIT
        CALL    CLEAR
        .PRINT  #M85                    ;MCS ACTIVE MESSAGE
        MOVB    #24.,A2DCON             ;DISABLE A2DCON INTERRUPT
        CLR     ACTIVE+2                ;NO ACTIVE CHANNEL BRT.
        MOV     SBBUF,MCSBUF            ;DATA INTO S+B BUFFER (DEFAULT)
        CALL    MCSIN
        JMP     SCOUT
60$:    MOV     #NO,SCFUNC
        MOVB    #15,IOBUF+2             ;'NOTHIN' LOOKS FOR THIS CARRIAGE RE
TURN
NOTHIN: MOVB    #'0,IOBUF               ;ZERO TO THE FIRST TWO BYTES
        MOVB    #'0,IOBUF+1
        CMPB    #15,IOBUF+2             ;A CARRIAGE RETURN?
        BEQ     10$                     ;BRANCH IF YES
        BR      20$                     ;CONVERT TO BINARY IF NO
10$:    .PRINT  #M6
        CALL    READ
20$:    CALL    ASCBIN
        BCC     47$                     ;BRANCH IF NO ERROR
        .PRINT  #M4                     ;PRINT ERROR MESSAGE
        BR      10$                     ;TRY AGAIN
47$:    TST     R1                      ;GT ZERO LE 1000. ?
        BEQ     55$
        BGT     50$
        NEG     R1                      ;MAKE POSITIVE IF IT WAS NEGATIVE
50$:    CMP     #1000.,R1
        BMI     55$
        MOV     R1,NCHAN
        CALL    BUFINIT
        CALL    CLEAR
        JMP     SCOUT
55$:    .PRINT  #M4
        BR      10$
SC05:   .PRINT  #M7
        .PRINT  #M8
        JMP     SC
SCOUT:  CLR     UPDATE                  ;UPDATE DISPLAY
        RETURN                          ;RETURN TO CALLER
;
;SUBROUTINE TO 'SET' A DEVICE
;LOADS 'ROUTINE' WITH THE ADDRESS OF SUBROUTINE TO BE EXECUTED
;
SE:     MOV     IOBUF+2,IOBUF           ;MOV IN SECOND COMMAND,IF ANY
        CMPB    IOBUF,#15               ;IS IT A CARRIAGE RETURN?
        BEQ     2$                      ;IF YES,ASK FOR SECOND PART OF COMMA
ND
        BR      3$                      ;WHAT COMMAND IS IT?
2$:     .PRINT  #M5
        CALL    READ
3$:     CMP     SCCODE,IOBUF            ;IS THE DEVICE BEING 'SCANNED'?
```

```
        BNE     5$
        .PRINT  #M91
        RETURN
5$:     CMP     #'LA,IOBUF              ;SET THE LASER?
        BNE     10$
        MOV     #LA,ROUTINE
        JMP     SEOUT                   ;GO CALL THE ROUTINE
10$:    CMP     #'QU,IOBUF              ;SET THE QUADRUPOLE?
        BNE     20$
        MOV     #QU,ROUTINE
        JMP     SEOUT                   ;GO CALL THE ROUTINE
20$:    CMP     #'TR,IOBUF              ;SET THE ION TRAP?
        BNE     30$
        MOV     #TR,ROUTINE
        JMP     SEOUT                   ;GO CALL THE ROUTINE
30$:    CMP     #'MA,IOBUF              ;SET THE MAGNET?
        BNE     40$
        MOV     #MA,ROUTINE             ;SECTOR MAGNET
        JMP     SEOUT                   ;GO CALL THE ROUTINE
40$:    .PRINT  #M51                    ;NONEXISTENT OPTION
        RETURN
SEOUT:  .PRINT  #M96                    ;KEY MODE READY
        CALL    KEYHOL
        RETURN
;
;
;SUBROUTINE TO IMPLEMENT 'KEY-MODE' OPERATION.
;
KEYHOL: BIS     #10000,JSW              ;SPECIAL TTY MODE(NO CR/LF)
10$:    .TTINR                          ;CHARACTER INPUT?
        BCS     30$                     ;BRANCH IF NO
        CMPB    #',,R0                  ;DECREASE?
        BNE     15$
        MOV     #-1,DIRFLG              ;SET DIRECTION FLAG
        BR      20$
15$:    CMPB    #'.,R0                  ;INCREASE?
        BNE     25$
        MOV     #1,DIRFLG
20$:    CALL    @ROUTINE                ;CALL THE ROUTINE
        BR      30$
25$:    CMPB    #'S,R0                  ;STOP COMMAND?
        BEQ     35$
30$:    TST     UPDATE                  ;UPDATE WILL OCCUR
        BNE     10$
        MOV     DICODE+2,DICODE         ;GET RID OF POSSIBLE 'SM'
        MOV     DIFUNC+2,DIFUNC         ;RESTORE FUNCTION CODE
        CALL    DISCAL                  ;UPDATE
        MOV     UPDATE+2,UPDATE
        BR      10$
35$:    .TTINR                          ;EAT ALL OTHER CHARACTERS
        BCC     35$
        BIC     #10000,JSW              ;RETURN TTY TO NORMAL MODE
        CALL    CRLF                    ;TYPE CR-LF
        RETURN
;
```

```
;
;SUBROUTINE TO SET THE SCANNING PROCEDURE
;
SCNPRO: MOV     IOBUF+2,IOBUF           ;MOV IN SECOND COMMAND,IF ANY
        CMPB    IOBUF,#15               ;IS IT A CARRIAGE RETURN?
        BEQ     10$                     ;IF YES,ASK FOR SECOND PART OF COMMA
ND
        BR      20$                     ;WHAT COMMAND IS IT?
10$:    .PRINT  #M9                     ;TYPE OF SCANNING?
        CALL    READ
20$:    CMP     #'BI,IOBUF              ;BIDIRECTIONAL?
        BEQ     1010$
        CMP     #'UN,IOBUF              ;UNIDIRECTIONAL?
        BEQ     1010$
        .PRINT  #M4
        CALL    CRLF
        BR      10$
1010$:  MOV     IOBUF,SCTYPE
SCNOUT: RETURN                          ;RETURN TO CALLER
;
;
;SUBROUTINE TO CLEAR DATA BUFFER AND RE-INIT
;
CLRQUE: .PRINT  #M49                    ;CLEAR MESSAGE
        .PRINT  #M48                    ;ARE YOU SURE MESSAGE
        CALL    READ                    ;GET RESPONSE
        CMPB    #'Y,IOBUF               ;IS IT YES?
        BNE     CLROUT                  ;NO,GET OUT
CLEAR:  CLR     NSCAN                   ;CLEAR SCAN COUNTER
        MOV     NCHAN,R1                ;NOW CLEAR BUFFER
        MUL     #6.,R1
        MOV     SBBUF,R2                ;DATA BUFFER POINTER
10$:    MOV     #0,(R2)+
        SOB     R1,10$
        MOV     DACYCL+2,DACYCL         ;UPDATE CYCLE COUNT
CLROUT: RETURN                          ;RETURN TO CALLER
;
;
;THIS SUBROUTINE SETS THE DEVICE BEING SCANNED TO ITS STARTING
;POINT,RESETS THE BUFFER OFFSET FLAG AND THE SCAN DIRECTION FLAG.
;
RESQUE: .PRINT  #M46                    ;RESET THE DEVICE?
        .PRINT  #M48                    ;ARE YOU SURE?
        CALL    READ                    ;GET REPLY
        CMPB    #'Y,IOBUF               ;YES OR NO
        BNE     RESOUT                  ;NO,LEAVE
RESET:  CMP     #'LA,SCCODE             ;DYE LASER
        BEQ     10$
        CMP     #'QU,SCCODE             ;QUADRUPOLE
        BEQ     20$
        CMP     #'TR,SCCODE             ;ION TRAP
        BEQ     30$
        CMP     #'QP,SCCODE             ;QUAD SCANNING PEAK POSITION
        BEQ     40$
        CMP     #'MA,SCCODE             ;SECTOR MAGNET?
```

```
        BEQ     50$
        CMP     #'NO,SCCODE               ;'NOTHING'
        BEQ     60$
        BR      RESOUT
10$:    MOV     SHOWAV,RESULT
        MOV     SHOWAV+2,RESULT+2
        CALL    DYESET
        BR      60$
20$:    MOV     MDAC1,MDAC
        BR      60$
30$:    MOV     DELOFF,DELTOT
        MOV     DELTOT,KWBPR
        NEG     KWBPR
        BR      60$
40$:    MOV     MASBUF,MDAC
        BR      60$
50$:    MOV     MADAC1,MADAC
60$:    MOV     #1,DIRFLG
        CLR     BUFOFF
        BIC     #2,RUNFLG                 ;RUN NOT IN PROGRESS
RESOUT: RETURN                            ;RETURN TO CALLER
;
;
;THIS SUBROUTINE CREATES AND WRITES A DATA FILE
;
WDISK:  MOV     R2,R2DSK                  ;SAVE R2
        MOV     #DATBUF,R2                ;STARTING LOCATION OF DATA BUFFER
        MOV     NCHAN,(R2)+               ;STORE PROGRAM VARIABLES
        MOV     NCHAN+2,(R2)+
        MOV     SHOWAV,(R2)+
        MOV     SHOWAV+2,(R2)+
        MOV     LONWAV,(R2)+
        MOV     LONWAV+2,(R2)+
        MOV     WAVINC,(R2)+
        MOV     WAVINC+2,(R2)+
        MOV     SETWAV,(R2)+
        MOV     SETWAV+2,(R2)+
        MOV     CALWAV,(R2)+
        MOV     CALWAV+2,(R2)+
        MOV     MDAC1+2,(R2)+
        MOV     MDAC2,(R2)+
        MOV     MRES+2,(R2)+
        MOV     NSTEP,(R2)+
        MOV     DELINC,(R2)+
        MOV     DELOFF,(R2)+
        MOV     LASHOTS,(R2)+
        MOV     LASHOTS+2,(R2)+
        MOV     SCCODE,(R2)+
        MOV     SCFUNC,(R2)+
        MOV     NSCAN,(R2)+
        MOV     NSCAN+2,(R2)+
        MOV     A2DGNS,(R2)+
        MOV     DACYCL+2,(R2)+
        MOV     BUFOFF,(R2)+
        MOV     MADAC1+2,(R2)+            ;SECROR MAGNRT
```

```
        MOV     MADAC2,(R2)+
        MOV     MARES+2,(R2)+
        MOV     DATE,(R2)+
        MOV     RUN,(R2)+               ;RUN NUMBER OF DATE
        MOV     SCTYPE,(R2)+            ;SCANNING PROCEDURE
        MOV     TRGBIN,(R2)+
        CALL    WRDISK                  ;WRITE THE DATA FILE
        MOV     R2DSK,R2                ;RESTORE R2
        RETURN                          ;RETURN TO CALLER
;
;SUBROUTINE TO READ A DATA FILE OFF THE DISK.
;
RDISK:  .PRINT  #M47                    ;READ THE DISK?
        .PRINT  #M48                    ;ARE YOU SURE?
        CALL    READ                    ;GET REPLY
        CMPB    #'Y,IOBUF               ;YES?
        BNE     RDIOUT                  ;NO,GET OUT
        CALL    DISHLT                  ;HALT DISPLAY
        CALL    CRLF
        CLR     ACCFLG                  ;STOP TAKING DATA
        MOV     R2,R2DSK                ;SAVE R2
        CALL    REDISK                  ;READ THE FILE
        MOV     #DATBUF,R2              ;STARTING LOCATION OF DATA BUFFER
        MOV     (R2)+,NCHAN             ;RESTORE PROGRAM VARIABLES
        MOV     (R2)+,NCHAN+2
        MOV     (R2)+,SHOWAV
        MOV     (R2)+,SHOWAV+2
        MOV     (R2)+,LONWAV
        MOV     (R2)+,LONWAV+2
        MOV     (R2)+,WAVINC
        MOV     (R2)+,WAVINC+2
        MOV     (R2)+,SETWAV
        MOV     (R2)+,SETWAV+2
        MOV     (R2)+,CALWAV
        MOV     (R2)+,CALWAV+2
        MOV     (R2)+,MDAC1+2
        MOV     (R2)+,MDAC2
        MOV     (R2)+,MRES+2
        MOV     (R2)+,NSTEP
        MOV     (R2)+,DELINC
        MOV     (R2)+,DELOFF
        MOV     (R2)+,LASHOTS
        MOV     (R2)+,LASHOTS+2
        MOV     (R2)+,SCCODE
        MOV     (R2)+,SCFUNC
        MOV     (R2)+,NSCAN
        MOV     (R2)+,NSCAN+2
        MOV     (R2)+,A2DGNS
        MOV     (R2)+,DACYCL+2
        MOV     (R2)+,BUFOFF
        MOV     (R2)+,MADAC1+2
        MOV     (R2)+,MADAC2
        MOV     (R2)+,MARES+2
        MOV     (R2)+,DATE              ;FILE DATE
        MOV     (R2)+,RUN               ;FILE RUN NUMBER
```

```
        MOV     (R2)+,SCTYPE
        MOV     (R2)+,TRGBIN
        MOV     BUFOFF,R2               ;SAVE
        CALL    BUFINIT
        MOV     R2,BUFOFF
        CALL    DATES
        MOV     A2DGNS,<IOBUF+2>
        CALL    A2DAMP                  ;SET GAIN SETTING
        CMP     #'UN,SCTYPE             ;MISSING CODE?
        BEQ     10$
        CMP     #'BI,SCTYPE
        BEQ     10$
        CALL    TYSTAT                  ;TYPE STATUS
        .PRINT  #M74                    ;WARNING TO USER
10$:    MOV     R2DSK,R2                ;RESTORE R2
RDIOUT: RETURN
;
;
;SETS NUMBER OF SCANS TO BE MADE
;
NOSCNS: MOV     R1,R1NOS                ;SAVE R1
        MOV     #'00,IOBUF              ;PLACE ZEROES IN THE FIRST WORD
        CMPB    #15,IOBUF+2             ;IS IT A CARRIAGE RETURN?
        BEQ     10$                     ;BRANCH IF YES
        BR      15$                     ;CONVERT TO BINARY IF NO
10$:    .PRINT  #M14                    ;ASK FOR NUMBER OF SCANS TO BE MADE
        CALL READ                       ;READ ENTRY
15$:    CALL    ASCBIN                  ;CONVERT TO BINARY NUMBER
        BCC     20$                     ;ERROR?
        .PRINT  #M4                     ;BAD ENTRY MESSAGE
        BR      10$                     ;TRY AGAIN
20$:    MOV     R1,NSCAN+2              ;STORE ENTRY
        MOV     R1NOS,R1                ;RESTORE R1
NOSOUT: RETURN                          ;RETURN TO CALLER
;
;
;SUBROUTINE TO SET THE NUMBER OF DATA CYCLES PER CHANNEL.
;
NOCYCL: MOV     R1,R1NOC                ;SAVE R1
        MOV     #'00,IOBUF              ;PLACE ZEROES IN THE FIRST WORD
        CMPB    #15,IOBUF+2             ;IS IT A CARRIAGE RETURN?
        BEQ     10$                     ;BRANCH IF YES
        BR      15$                     ;CONVERT TO BINARY IF NO
10$:    .PRINT  #M41                    ;ASK FOR CYCLES
        CALL    READ                    ;GET THE REPLY
15$:    CALL    ASCBIN                  ;CONVERT TO BINARY
        BCC     20$                     ;BRANCH IF OK
        .PRINT  #M4                     ;BAD ENTRY MESSAGE
        BR      10$                     ;TRY AGAIN
20$:    MOV     R1,DACYCL+2             ;SAVE IT
        MOV     R1,DACYCL               ;LOAD CYCLE COUNTER
        MOV     R1NOC,R1                ;RESTORE R1
NOCOUT: RETURN                          ;RETURN TO CALLER
;
;
```

```
;
;SUBROUTINE TO SMOOTH DATA IN THE DISPLAY BUFFER
;
SMOOTH: MOV     R1,R1SMO                ;SAVE REGISTERS
        MOV     R2,R2SMO
        MOV     R3,R3SMO
        MOV     R4,R4SMO
        MOV     R5,R5SMO
        MOV     #SM,DIFUNC              ;SMOOTH FUNCTION CODE FOR DISPLAY
        MOV     #"SM,DICODE             ;"SMOOTH" BEING DISPLAYED
        MOV     #IOBUF,R0               ;BUFFER POINTER
        MOV     #"00,(R0)+              ;ZEROES IN FIRST WORD
10$:    CMPB    #15,(R0)                ;CR?
        BNE     20$
        CALL    ASCBIN
        BCC     15$
        BR      25$
15$:    MOV     R1,NPTAVG
        BR      45$
20$:    CMPB    #'X,(R0)+               ;X?
        BNE     10$
        MOVB    #15,-1(R0)              ;REPLACE X WITH CR
        CALL    ASCBIN                  ;CONVERT
        BCC     30$                     ;ERROR?
25$:    .PRINT  #M4                     ;INVALID ENTRY
        BR      SMOOT
30$:    MOV     R1,NPTAVG               ;NUM. OF POINTS TO BE AVERAGED
        MOV     #IOBUF,R0
35$:    CMPB    #15,(R0)                ;CR?
        BEQ     40$
        MOVB    #'0,(R0)+               ;FILL WITH ZERO
        BR      35$
40$:    MOVB    #'0,(R0)                ;REPLACE CR WITH ZERO
        CALL    ASCBIN                  ;CONVERT
        BCC     50$
45$:    MOV     #1,TIMES
        BR      60$
50$:    MOV     R1,TIMES                ;NUM. OF TIMES TO SMOOTH
60$:    CALL    SMUZ                    ;SMOOTH DATA
        DEC     TIMES
        BNE     60$
SMOOT:  MOV     R1SMO,R1                ;UNSAVE REGISTERS
        MOV     R2SMO,R2
        MOV     R3SMO,R3
        MOV     R4SMO,R4
        MOV     R5SMO,R5
        RETURN
SMUZ:   CMP     #1,NPTAVG               ;DO NOT AVERAGE IF ONE
        BGE     SMOUT                   ;LEAVE
        CMP     NPTAVG,NCHAN            ;MUST BE LESS THAN NUM. OF CHANNELS
        BGT     SMOUT
        MOV     NPTAVG,R1               ;PLACE # OF POINT SMOOTH IN R1
        MOV     NCHAN,R4                ;DIVIDE DATA BY NPTAVG
        MOV     DIBUF,R5                ;STARTING LOCATION OF DATA
        TST     (R5)+                   ;DATA AS X,Y PAIRS
```

```
25$:    MOV     (R5),R3                 ;GET A NUMBER
        SUB     #3777,R3                ;REMOVE D/A BIAS
        CLR     R2                      ;CLEAR HIGH WORD
        DIV     NPTAVG,R2               ;DIVIDE BY NPTAVG
        MOV     R2,(R5)+                ;PLACE RESULT IN BUFFER
        TST     (R5)+
        SOB     R4,25$                  ;DIVIDE THEM ALL
        BIT     #1,R1                   ;IS THE NUMBER ODD?
        BEQ     30$                     ;BRANCH IF NO
        SUB     #1,R1                   ;MAKE EVEN
30$:    MOV     R1,OFFSET               ;BUFFER OFFSET VALUE
        MOV     NCHAN,R2                ;NUMBER OF CHANNELS TO R2
        SUB     R1,R2                   ;CALCULATE LAST CHANNEL
        ASH     #2,R2
        SUB     #4,R2
        MOV     R2,LCHN                 ;LAST CHANNEL
        CLR     SMOPTR                  ;CLEAR SMOOTH-POINTER
40$:    MOV     DIBUF,R2                ;DISPLAY BUFFER STARTING ADDRESS
        TST     (R2)+
        MOV     NPTAVG,R4               ;NO. OF POINTS TO R4
        CLR     R1                      ;SUM REGISTER
        ADD     SMOPTR,R2               ;POINT TO CHANNEL
50$:    ADD     (R2)+,R1                ;ADD CHANNELS
        TST     (R2)+
        SOB     R4,50$                  ;GET "NPTAVG" OF THEM
        ADD     #3777,R1                ;ADD D/A BIAS
        MOV     DIBUF,R2                ;RE-INIT
        TST     (R2)+
        ADD     SMOPTR,R2               ;ADD POINTER
        MOV     R1,(R2)                 ;PLACE SUM IN THE DATA BUFFER
        CMP     LCHN,SMOPTR             ;DONE YET?
        BEQ     55$                     ;BRANCH IF YES
        ADD     #4,SMOPTR               ;INCREASE POINTER--CONTINUE
        BR      40$                     ;DO MORE
55$:    MOV     OFFSET,R1               ;RE-INIT
        MOV     DIBUF,R2
        TST     (R2)+
        ADD     LCHN,R2
        ADD     #4,R2
60$:    MOV     #3777,(R2)+             ;THIS ZEROES UNUSED DATA CHANNELS
        TST     (R2)+
        SOB     R1,60$
        MOV     OFFSET,R1               ;SHIFT THE DATA IN THE BUFFER
        MOV     DIBUF,R2                ;DISPLAY BUF LOCATION
        TST     (R2)+
        ADD     LCHN,R2                 ;POINT TO LAST SMOTHED DATA CHANNEL
        MOV     R2,R3
        ADD     R1,R3                   ;SHIFT BY OFFSET
        ADD     R1,R3
        TST     (R2)+
        TST     (R3)+
        ASR     R1                      ;DIVIDE OFFSET BY TWO
130$:   MOV     -(R2),-(R3)             ;SHIFT!!
        TST     -(R2)
        TST     -(R3)
```

```
        SUB     #4,LCHN
        BNE     130$                    ;SHIFT THEM ALL
        TST     -(R2)
140$:   MOV     #3777,(R2)+             ;ZERO THE FIRST OFFSET/2 CHANNELS
        TST     (R2)+
        SOB     R1,140$
SMOUT:  RETURN                          ;RETURN TO CALLER
;
;
;SURROUTINE TO SET 'ION TRAP' PARAMETERS
;
TRAP:   MOV     R1,R1TRA                ;SAVE REGISTER ONE
10$:    .PRINT  #M17                    ;ASK FOR DELAY OFFSET
        MOV     DELOFF,R1               ;PRINT EXISTING VALUE
        CALL    DECOUT
        MOV     #' ,R0                  ;TYPE A SPACE
        .TTYOUT
        CALL    READ                    ;READ IT IN
        CMPB    #15,IOBUF               ;IS IT A CR?
        BEQ     35$
        CALL    ASCBIN                  ;CONVERT TO BINARY
        BCC     30$                     ;ALL OK?
20$:    .PRINT  #M4                     ;PRINT MESSAGE IF NOT OK
        BR      10$                     ;TRY IT AGAIN
30$:    TST     R1                      ;NO NEGATIVES
        BLT     20$                     ;BAD ENTRY--GO BACK
        MOV     R1,DELOFF               ;STORE VALUE
        NEG     R1                      ;TWO'S COMPLEMENT
        MOV     R1,KWBPR                ;MOVE TO BUFFER/PRESET REGISTER
35$:    CMP     SCCODE,#'TR             ;IS THE TRAP BEING SCANNED?
        BNE     TRAOUT                  ;BRANCH OUT IF NOT
40$:    .PRINT  #M18                    ;ASK FOR DELAY INCREMENT
        MOV     DELINC,R1               ;TYPE VALUE
        CALL    DECOUT
        MOV     #' ,R0                  ;THE SPACE
        .TTYOUT
        CALL    READ                    ;GET IT
        CMPB    #15,IOBUF               ;CR?
        BEQ     65$
        CALL    ASCBIN                  ;CONVERT IT
        BCC     60$                     ;ERROR?
50$:    .PRINT  #M4                     ;PRINT ERROR MESSAGE
        BR      40$                     ;TRY AGAIN
60$:    TST     R1                      ;NO NEGATIVES
        BLT     50$                     ;GO BACK IF NEGATIVE
        MOV     R1,DELINC               ;STORE ENTRY IN DELINC
65$:    MOV     DELOFF,DELTOT           ;RESET TOTAL DELAY WORD
TRAOUT: MOV     R1TRA,R1                ;RESTORE R1
        RETURN                          ;RETURN TO CALLER
;
;END OF PAGE TWO
```

```
;PAGE THREE
;
CU:     MOV     R1,R1SAV            ;SAVE R1
        CLR     KWCSR               ;DISABLE CLOCK INTERRUPT
        MOV     #'00,IOBUF          ;PLACE A ZERO IN THE FIRST WORD
        CMPB    #15,IOBUF+2         ;IS IT A CARRIAGE RETURN?
        BEQ     10$                 ;BRANCH IF YES
        BR      15$                 ;CONVERT TO BINARY IF NO
10$:    .PRINT  #M75                ;ASK FOR CLOCK UNITS
        CALL READ                   ;READ ENTRY
15$:    CALL    ASCBIN              ;CONVERT TO BINARY NUMBER
        BCC     20$                 ;ERROR?
        .PRINT  #M4                 ;BAD ENTRY MESSAGE
        BR      10$                 ;TRY AGAIN
20$:    CMP     R1,#4               ;LESS THAN 4?
        BGE     CUOUT               ;LEAVE IF NOT
        ASH     #3,R1               ;MUL BY EIGHT
        ADD     #8264.,R1
        MOV     R1,KWCSR            ;CLOCK CONTROL STATUS REGISTER
CUOUT:  MOV     R1SAV,R1            ;USAVE R1
        RETURN
;
;SUBROUTINE TO SET 'RUN' NUMBER
;
ER:     MOV     R1,R1SAV            ;SAVE R1
        MOV     #'00,IOBUF          ;PLACE A ZERO IN THE FIRST WORD
        CMPB    #15,IOBUF+2         ;IS IT A CARRIAGE RETURN?
        BEQ     10$                 ;BRANCH IF YES
        BR      15$                 ;CONVERT TO BINARY IF NO
10$:    .PRINT  #M67                ;ASK FOR RUN NUMBER
        CALL READ                   ;READ ENTRY
15$:    CALL    ASCBIN              ;CONVERT TO BINARY NUMBER
        BCC     20$                 ;ERROR?
        .PRINT  #M4                 ;BAD ENTRY MESSAGE
        BR      10$                 ;TRY AGAIN
20$:    MOV     R1,RUN              ;SAVE ENTRY
        MOV     R1,RUN+2            ;COPY
        RETURN
;
;
;SUBROUTINE TO SET 'UPDATE' NUMBER
;
US:     MOV     R1,R1SAV            ;SAVE R1
        MOV     #'00,IOBUF          ;PLACE A ZERO IN THE FIRST WORD
        CMPB    #15,IOBUF+2         ;IS IT A CARRIAGE RETURN?
        BEQ     10$                 ;BRANCH IF YES
        BR      15$                 ;CONVERT TO BINARY IF NO
10$:    .PRINT  #M72                ;ASK FOR UPDATE NUMBER
        CALL READ                   ;READ ENTRY
15$:    CALL    ASCBIN              ;CONVERT TO BINARY NUMBER
        BCC     30$                 ;ERROR?
20$:    .PRINT  #M4                 ;BAD ENTRY MESSAGE
        BR      10$                 ;TRY AGAIN
30$:    TST     R1                  ;NEG OR ZERO?
        BLE     20$
```

```
        MOV     R1,UPDATE                   ;SAVE ENTRY
        MOV     R1,UPDATE+2                 ;COPY
        RETURN
;
;
;PRINT OUT (LP) ROUTINE
;
PR:     MOV     IOBUF+2,IOBUF               ;OPTION TO FIRST WORD
        CMPB    IOBUF,#15                   ;CR?
        BNE     10$
        .PRINT  #M10                        ;OPTION?
        CALL    READ
10$:    CMP     #'HE,IOBUF                  ;PRINT HEADER?
        BNE     20$
        CALL    PRHEDR
        BR      50$
20$:    CMP     #'FU,IOBUF                  ;PRINT FULL INFO?
        BNE     30$
        CALL    PRFULL
        BR      50$
30$:    CMP     #'DA,IOBUF                  ;PRINT DATA
        BNE     40$
        CALL    PRDATA
        BR      50$
40$:    CMP     #'ST,IOBUF                  ;PRINT STATUS?
        BNE     50$
        CALL    PRSTAT
50$:    RETURN
;
TY:     MOV     IOBUF+2,IOBUF               ;OPTION TO FIRST WORD
        CMPB    IOBUF,#15                   ;CR?
        BNE     10$
        .PRINT  #M10                        ;OPTION?
        CALL    READ
10$:    CMP     #'HE,IOBUF                  ;PRINT HEADER?
        BNE     20$
        CALL    TYHEDR
        BR      40$
20$:    CMP     #'ST,IOBUF                  ;PRINT FULL INFO?
        BNE     30$
        CALL    TYSTAT
        BR      40$
30$:    CMP     #'DA,IOBUF                  ;PRINT DATA
        BNE     40$
        CALL    TYDATA
40$:    RETURN
;
;
;SUBROUTINE TO ENTER THE HEADER TO DATA BUFFER.
;
HEADER: MOV     R1,R1HEA                    ;SAVE REGISTERS R1,R2
        MOV     R2,R2HEA
5$:     .PRINT  #M54                        ;NEW HEADER?
        CALL    READ                        ;YES OR NO
        CMPB    #'Y,IOBUF                   ;LOOK FOR Y
```

```
        BEQ     10$                             ;BRANCH IF YES
        BR      HEAOUT                          ;LEAVE OTHERWISE
10$:    MOV     #DATBUF,R2                      ;DATA BUFFER STARTING LOCATION
        ADD     #100.,R2                        ;START OF HEADER SECTION
        MOV     #450.,R1                        ;MAX. LENGTH OF MESSAGE
15$:    CLR     (R2)+                           ;CLEAR HEADER SECTION
        SOB     R1,15$
        MOV     #DATBUF,R2                      ;STARTING LOCATION
        ADD     #100.,R2                        ;START OF HEADER
        .PRINT  #M56                            ;ENTER HEADER MESSAGE
        MOV     #'>,R0                          ;PRINT PROMPT CHARACTER
        .TTYOUT
        MOV     #448.,R1                        ;NUMBER OF CHARACTERS PERMITTED
20$:    .TTYIN                                  ;READ IN THE HEADER
        CMPB    #'^,R0                          ;CHECK FOR HEADER TERMINATER
        BEQ     40$                             ;BRANCH IF AT THE END
        DEC     R1                              ;COUNT DOWN SPACE LEFT
        BEQ     30$                             ;BRANCH IF NO SPACE LEFT
        MOVB    R0,(R2)+                        ;PLACE CHARACTER INTO BUFFER
        BR      20$                             ;GO GET MORE
30$:    .TTYIN                                  ;EAT THE LEFT OVERS
        CMPB    #12,R0                          ;THRU EATING?
        BNE     30$                             ;IF NO GET MORE
        .PRINT  #M57                            ;TRUNCATION MESSAGE
        BR      50$                             ;CONTINUE
40$:    .TTYIN                                  ;EAT CR-LF
        CMP     #12,R0                          ;HAVE LINE FEED YET?
        BNE     40$                             ;NO? WELL GO GET IT
50$:    MOV     #200,(R2)                       ;HEADER TERMINATER SYMBOL
HEAOUT: MOV     R1HEA,R1                        ;RESTORE R1,R2
        MOV     R2HEA,R2
        RETURN                                  ;RETURN TO CALLER
;
;
;SUBROUTINE PRINT OUTPUTS DATA TO THE LINE PRINTER
;
PRFULL: MOV     R1,R1PRI                        ;SAVE THE REGISTERS
        MOV     R2,R2PRI
        MOV     R3,R3PRI
        MOV     R4,R4PRI
        MOV     R5,R5PRI
        MOV     #14,R0                          ;FORM FEED TO LP
        CALL    LPOUT
        CALL    PRHEDR                          ;PRINT THE HEADER
        CALL    LPCRLF                          ;SKIP A LINE
        CALL    PRSTAT                          ;PRINT THE STATUS
        CALL    LPCRLF                          ;SKIP A LINE
        BR      PRINT                           ;GO PRINT DATA
PRDATA: MOV     R1,R1PRI                        ;SAVE THE REGISTERS
        MOV     R2,R2PRI
        MOV     R3,R3PRI
        MOV     R4,R4PRI
        MOV     R5,R5PRI
        MOV     #14,R0                          ;FORM FEED TO LP
        CALL    LPOUT
```

```
PRINT:  BIS     #10000,JSW          ;SPECIAL TTY MODE
        MOV     #M21,R0             ;ADDRESS OF ASCII STRING
        CALL    LPRINT              ;OUTPUT TO LINE PRINTER
        CALL    LPCRLF              ;CARRIAGE RETURN-LINE FEED
        MOV     DISTRT,R1           ;CHANNEL COUNTER
        SUB     DIBUF,R1
        MOV     SBBUF,R2            ;STARTING ADDRESS OF S+B BUFFER
        ADD     R1,R2
        MOV     BABUF,R3            ;STARTING ADDRESS OF BACKGRD.BUFFER
        ADD     R1,R3
        MOV     POBUF,R4            ;STARTING ADDRESS OF LASER POWER BUF
FER
        ADD     R1,R4
        ADD     #4,R1               ;STARTING CHANNEL NUMBER
        ASH     #-2,R1
        MOV     DICHNS,R5           ;NUMBER OF DATA CHANNELS
10$:    CALL    LPDECO              ;PRINT THE CHANNEL NUMBER(IN R1)
        MOV     #M22,R0             ;ASCII STRING
        CALL    LPRINT              ;PRINT THE SPACES
        MOV     (R2),LOWORD         ;LOAD SIG.+BACK. FOR OUTPUT
        MOV     2(R2),HIWORD
        CALL    LPDCO2              ;PRINT THE NUMBER
        MOV     #M22,R0             ;PRINT THE SPACES
        CALL    LPRINT
        MOV     (R3),LOWORD         ;LOAD BACKGROUND
        MOV     2(R3),HIWORD
        CALL    LPDCO2              ;PRINT IT
        MOV     #M22,R0             ;PRINT THE SPACES
        CALL    LPRINT
        MOV     (R2)+,LOWORD        ;CALCULATE THE SIGNAL
        MOV     (R2)+,HIWORD        ;SUBTRACT BACKGROUND FROM SIG.+BACK.
        SUB     (R3)+,LOWORD
        SBC     HIWORD
        SUB     (R3)+,HIWORD
        CALL    LPDCO2              ;PRINT THE SIGNAL
        MOV     #M22,R0             ;PRINT THE SPACES
        CALL    LPRINT
        MOV     (R4)+,LOWORD        ;LOAD LASER-POWER FOR OUTPUT
        MOV     (R4)+,HIWORD
        CALL    LPDCO2              ;PRINT IT!
        CALL    LPCRLF              ;CARRIAGE RETURN-LINE FEED
        DEC     R5                  ;COUNT THE LOOPS
        BEQ     30$                 ;BRANCH OUT IF DONE
        INC     R1                  ;INCREMENT CHANNEL NUMBER
        .TTINR                      ;CHECK FOR 'STOP' COMMAND
        BCS     10$                 ;NONE--GO AHEAD
        CMPB    #'S,R0              ;'S' IS THE STOP COMMAND
        BNE     10$                 ;CONTINUE IF NOT 'S'
30$:    .TTINR                      ;EAT OTHER CHARACTERS
        BCC     30$
        BIC     #10000,JSW          ;NORMAL TTY MODE
        MOV     R5PRI,R5            ;RESTORE THE REGISTERS
        MOV     R4PRI,R4
        MOV     R3PRI,R3
        MOV     R2PRI,R2
```

```
        MOV     R1PRI,R1
PRIOUT: RETURN                          ;RETURN TO CALLER
;
;
;THIS SUBROUTINE OUTPUTS THE 'STATUS' OF THE SPECTROMETER TO THE LINE PRINTE
R.
;
PRSTAT: MOV     R1,R1PRS                ;SAVE R1
        CALL    PRDATE                  ;PRINT THE DATE
        MOV     #M67,R0                 ;RUN NUMBER
        CALL    LPRINT
        MOV     RUN,R1
        BIT     #2,RUNFLG               ;RUN IN PROGRESS?
        BEQ     1$
        INC     R1
1$:     CALL    LPDECO
        CALL    LPCRLF
        CMP     #'LA,SCCODE             ;IS THE LASER BEING SCANNED?
        BEQ     10$                     ;BRANCH IF YES
        CMP     #'QU,SCCODE             ;IS THE QUADRUPOLE BEING SCANNED?
        BEQ     20$                     ;BRANCH IF YES
        CMP     #'TR,SCCODE             ;IS THE ION TRAP BEING SCANNED?
        BEQ     30$                     ;BRANCH IF YES
        CMP     #'MA,SCCODE             ;SECTOR MAGNET?
        BEQ     40$
        CMP     #'QP,SCCODE             ;QUAD:PEAKS OPTION?
        BNE     3$
        JMP     50$
3$:     CMP     #'MC,SCCODE             ;MCS UNIT?
        BNE     5$
        JMP     55$
5$:     JMP     60$                     ;NO DEVICE IS BEING SCANNED.
;
;
10$:    MOV     #M24,R0                 ;MESSAGE STRING TO R0
        CALL    LPRINT                  ;OUTPUT TO LP
        MOV     #M25,R0                 ;MESSAGE STRING
        CALL    LPRINT                  ;OUTPUT MESSAGE
        MOV     SHOWAV,LOWORD           ;LOAD LOW WORD
        MOV     SHOWAV+2,HIWORD         ;LOAD HIGH WORD
        CALL    LPDC02                  ;SEND NUMBER TO LP
        CALL    LPCRLF                  ;CARRIAGE RETURN
        MOV     #M26,R0                 ;MESSAGE STRING
        CALL    LPRINT                  ;SEND IT
        MOV     LONWAV,LOWORD           ;PRINT OUT LONG WAVELENGTH
        MOV     LONWAV+2,HIWORD
        CALL    LPDC02                  ;SEND IT TO LP
        CALL    LPCRLF                  ;CARRIAGE RETURN
        MOV     #M27,R0                 ;MESSAGE STRING
        CALL    LPRINT                  ;SEND TO LP
        MOV     WAVINC,LOWORD           ;OUTPUT WAVELENGTH INCREMENT
        MOV     WAVINC+2,HIWORD
        CALL    LPDC02                  ;PRINT IT
        CALL    LPCRLF                  ;CARRIAGE RETURN
        MOV     #M28,R0                 ;MESSAGE STRING
```

```
        CALL    LPRINT                  ;SEND IT!!
        MOV     NSTEP,R1                ;SEND NUMBER OF MOTOR STEPS
        CALL    LPDECO                  ;TO THE LP
        CALL    LPCRLF                  ;CARRIAGE RETURN
        JMP     70$                     ;KEEP GOING
;
;
20$:    MOV     #M30,R0                 ;MESSAGE 'QUAD. SCANNING'
        CALL    LPRINT                  ;SEND TO LP
        MOV     #M31,R0                 ;MESSAGE STRING
        CALL    LPRINT                  ;TO LP
        MOV     MDAC1+2,R1              ;LOW MASS
        CALL    LPDECO                  ;OUTPUT TO LP
        CALL    LPCRLF                  ;CARRIAGE RETURN
        MOV     #M32,R0                 ;MESSAGE STRING
        CALL    LPRINT                  ;SEND TO LP
        MOV     MDAC2,R1                ;HIGH MASS
        CALL    LPDECO                  ;TO THE LP!
        CALL    LPCRLF                  ;CARRIAGE RETURN
        MOV     #M19,R0                 ;MESSAGE STRING
        CALL    LPRINT                  ;SEND TO LP
        MOV     MRES+2,R1               ;# OF CHANNELS PER UNIT MASS
        CALL    LPDECO                  ;SEND IT
        CALL    LPCRLF                  ;CARRIAGE RETURN
        JMP     70$                     ;GO ON
;
;
30$:    MOV     #M29,R0                 ;ION TRAP MESSAGE
        CALL    LPRINT                  ;SEND MESSAGE TO LP
        MOV     #M42,R0                 ;DELINC MESSAGE
        CALL    LPRINT                  ;SEND IT
        MOV     DELINC,R1               ;LOAD THE VALUE
        CALL    LPDECO                  ;PRINT NUMBER
        CALL    LPCRLF                  ;CARRIAGE RETURN
        JMP     70$                     ;CONTINUE
;
;
40$:    MOV     #M12,R0                 ;SECTOR MAGNET
        CALL    LPRINT
        MOV     #M13,R0                 ;LOW CURRENT SETTING
        CALL    LPRINT
        MOV     MADAC1+2,R1
        CALL    LPDECO
        MOV     #M15,R0                 ;HIGH CURRENT
        CALL    LPRINT
        MOV     MADAC2,R1
        CALL    LPDECO
        MOV     #M19,R0                 ;CHANNELS/UNIT
        CALL    LPRINT
        MOV     MARES+2,R1
        CALL    LPDECO
        CALL    LPCRLF
        BR      70$
;
;
```

```
50$:    MOV     #M20,R0                 ;QUAD PEAKS OPTION
        CALL    LPRINT
        CALL    LPCRLF
        BR      70$
;
;
55$:    MOV     #M85,R0                 ;MCS UNIT
        CALL    LPRINT
        MOV     #M97,R0                 ;TRIGGER COUNT=
        CALL    LPRINT
        MOV     TRGBIN,R1               ;TRIGGER COUNT
        CALL    LPDECO
        CALL    LPCRLF                  ;CR-LF OUTPUT
        BR      70$
;
;
60$:    MOV     #M34,R0                 ;NO DEVICE SCANNING!!
        CALL    LPRINT                  ;SEND MESSAGE
70$:    MOV     #M43,R0                 ;DELOFF MESSAGE
        CALL    LPRINT                  ;SEND IT
        MOV     DELOFF,R1               ;LOAD VALUE
        CALL    LPDECO                  ;PRINT IT
        CALL    LPCRLF                  ;CARRIAGE RETURN
        MOV     #M35,R0                 ;LASER SETTING MESSAGE
        CALL    LPRINT                  ;SEND IT
        MOV     SETWAV,LOWORD           ;LOAD THE NUMBER
        MOV     SETWAV+2,HIWORD
        CALL    LPDC02                  ;PRINT IT
        CALL    LPCRLF                  ;CARRIAGE RETURN
        MOV     #M36,R0                 ;NUMBER OF CHANNELS MESSAGE
        CALL    LPRINT                  ;SEN
        MOV     NCHAN,R1                ;LOAD ALUE
        CALL    LPDECO                  ;SEND IT OUT!
        MOV     #M41,R0                 ;CYCLES MESSAGE
        CALL    LPRINT                  ;SEND IT
        MOV     DACYCL+2,R1             ;LOAD VALUE
        CALL    LPDECO                  ;SEND THE NUMBER
        CALL    LPCRLF                  ;CARRIAGE RETURN
        MOV     #M37,R0                 ;NUMBER OF SCANS MESSAGE
        CALL    LPRINT                  ;SEND IT
        MOV     NSCAN,R1                ;LOAD THE NUMBER
        CALL    LPDECO                  ;PRINT IT
        CALL    LPCRLF                  ;CARRIAGE RETURN
        MOV     #M38,R0                 ;CHANNEL NUMBER MESSAGE
        CALL    LPRINT                  ;SEND THE MESSAGE
        MOV     BUFOFF,R1               ;LOAD R1
        ASH     #-2,R1                  ;DIVIDE BY FOUR
        ADD     #1,R1                   ;ADD 1(ZERO=CHANNEL ONE)
        CALL    LPDECO                  ;PRINT THE NUMBER
        MOV     #M45,R0                 ;A2D GAIN SETTING MESSAGE
        CALL    LPRINT                  ;SEND IT
        MOVB    A2DGNS,R0               ;GET THE CHARACTER
        CALL    LPOUT                   ;SEND TO LP
        MOVB    <A2DGNS+1>,R0           ;GET THE OTHER ONE
        CALL    LPOUT                   ;SEND IT
```

```
        MOV     #M39,R0                 ;SCANNING PROCEDURE MESSAGE
        CALL    LPRINT                  ;SEND TO LP
        MOVB    SCTYPE,R0               ;GET THE FIRST CHARACTER
        CALL    LPOUT                   ;SEND TO LP
        MOVB    <SCTYPE+1>,R0           ;GET THE NEXT CHARACTER
        CALL    LPOUT                   ;SEND IT
        CALL    LPCRLF                  ;LP CARRIAGE RETURN-LINEFEED
        MOV     R1PRS,R1                ;RESTORE R1
PRSOUT: RETURN                          ;RETURN TO CALLER
;
;
;SUBROUTINE TO SEND MACHINE STATUS TO THE CONSOLE
;
TYSTAT: MOV     R1,R1TYS                ;SAVE R1
        CALL    TYDATE                  ;SEND THE DATE
        .PRINT  #M67                    ;RUN NUMBER
        MOV     RUN,R1
        BIT     #2,RUNFLG               ;RUN IN PROGRESS?
        BEQ     1$
        INC     R1
1$:     CALL    DECOUT
        CALL    CRLF
        CMP     #'LA,SCCODE             ;IS THE LASER BEING SCANNED?
        BEQ     15$                     ;BRANCH IF YES
        CMP     #'QU,SCCODE             ;IS THE QUADRUPOLE BEING SCANNED?
        BEQ     20$                     ;BRANCH IF YES
        CMP     #'TR,SCCODE             ;IS THE ION TRAP BEING SCANNED?
        BEQ     30$                     ;BRANCH IF YES
        CMP     #'MA,SCCODE             ;SECTOR MAGNET
        BEQ     40$
        CMP     #'QP,SCCODE             ;QUAD:PEAKS OPTION
        BNE     3$
        JMP     50$
3$:     CMP     #'MC,SCCODE             ;MCS UNIT?
        BNE     5$
        JMP     55$
5$:     JMP     60$                     ;NO DEVICE IS BEING SCANNED.
;
;
15$:    .PRINT  #M24                    ;LASER SCANNING MESSAGE
        .PRINT  #M25                    ;SHORT WAVELENGTH=
        MOV     SHOWAV,LOWORD           ;LOAD LOW WORD
        MOV     SHOWAV+2,HIWORD         ;LOAD HIGH WORD
        CALL    DCOUT2                  ;SEND NUMBER TO TT
        CALL    CRLF                    ;CARRIAGE RETURN
        .PRINT  #M26                    ;LONG WAVELENGTH =
        MOV     LONWAV,LOWORD           ;PRINT OUT LONG WAVELENGTH
        MOV     LONWAV+2,HIWORD
        CALL    DCOUT2                  ;SEND IT TO TT
        CALL    CRLF                    ;CARRIAGE RETURN
        .PRINT  #M27                    ;WAVELENGTH INCREMENT =
        MOV     WAVINC,LOWORD           ;OUTPUT WAVELENGTH INCREMENT
        MOV     WAVINC+2,HIWORD
        CALL    DCOUT2                  ;PRINT IT
        CALL    CRLF                    ;CARRIGE RETURN
```

```
        .PRINT  #M28                    ;NUMBER OF MOTOR STEPS =
        MOV     NSTEP,R1                ;SEND NUMBER OF MOTOR STEPS
        CALL    DECOUT                  ;TO THE TT
        CALL    CRLF                    ;CARRIAGE RETURN
        JMP     70$                     ;KEEP GOING
;
;
20$:    .PRINT  #M30                    ;MESSAGE 'QUAD. SCANNING'
        .PRINT  #M31                    ;LOW MASS =
        MOV     MDAC1+2,R1              ;LOW MASS
        CALL    DECOUT                  ;OUTPUT TO TT
        CALL    CRLF                    ;CARRIAGE RETURN
        .PRINT  #M32                    ;HIGH MASS =
        MOV     MDAC2,R1                ;HIGH MASS
        CALL    DECOUT                  ;TO THE TT!
        CALL    CRLF                    ;CARRIAGE RETURN
        .PRINT  #M19                    ;TYPE MESSAGE
        MOV     MRES+2,R1               ;# OF STEPS PER CHANNEL
        CALL    DECOUT                  ;SEND IT
        CALL    CRLF                    ;CARRIAGE RETURN
        JMP     70$                     ;GO ON
;
;
30$:    .PRINT  #M29                    ;ION TRAP MESSAGE
        .PRINT  #M42                    ;DELINC MESSAGE
        MOV     DELINC,R1               ;LOAD THE VALUE
        CALL    DECOUT                  ;PRINT NUMBER
        CALL    CRLF                    ;CARRIAGE RETURN
        JMP     70$                     ;CONTINUE
;
;
40$:    .PRINT  #M12                    ;SECTOR MAGNET
        .PRINT  #M13                    ;LOW CURRENT
        MOV     MADAC1+2,R1
        CALL    DECOUT
        .PRINT  #M15                    ;HIGH CURRENT
        MOV     MADAC2,R1
        CALL    DECOUT
        .PRINT  #M19                    ;CHANNELS/UNIT
        MOV     MARES+2,R1
        CALL    DECOUT
        CALL    CRLF
        BR      70$
;
;
50$:    .PRINT  #M20                    ;QUAD:PEAKS OPTION
        CALL    CRLF
        BR      70$
;
;
55$:    .PRINT  #M85                    ;MCS UNIT
        .PRINT  #M97                    ;TRIGGER COUNT=
        MOV     TRGBIN,R1               ;TRIGGER COUNT
        CALL    DECOUT                  ;DECIMAL OUTPUT
        CALL    CRLF
```

```
        BR      70$
;
;
60$:    .PRINT  #M34                    ;NO DEVICE SCANNING
70$:    .PRINT  #M43                    ;DELOFF MESSAGE
        MOV     DELOFF,R1               ;LOAD VALUE
        CALL    DECOUT                  ;PRINT IT
        CALL    CRLF                    ;CARRIAGE RETURN
        .PRINT  #M35                    ;LASER SETTING MESSAGE
        MOV     SETWAV,LOWORD           ;LOAD THE NUMBER
        MOV     SETWAV+2,HIWORD
        CALL    DCOUT2                  ;PRINT IT
        CALL    CRLF                    ;CARRIAGE RETURN
        .PRINT  #M36                    ;NUMBER OF CHANNELS MESSAGE
        MOV     NCHAN,R1                ;LOAD VALUE
        CALL    DECOUT                  ;SEND IT OUT!
        .PRINT  #M41                    ;DATA CYCLES MESSAGE
        MOV     DACYCL+2,R1             ;LOAD THE NUMBER
        CALL    DECOUT                  ;TYPE IT TO CONSOLE
        CALL    CRLF                    ;CARRIAGE RETURN
        .PRINT  #M37                    ;NUMBER OF SCANS MESSAGE
        MOV     NSCAN,R1                ;LOAD THE NUMBER
        CALL    DECOUT                  ;PRINT IT
        CALL    CRLF                    ;CARRIAGE RETURN
        .PRINT  #M38                    ;CHANNEL NUMBER MESSAGE
        MOV     BUFOFF,R1               ;LOAD R1
        ASH     #-2,R1                  ;DIVIDE BY FOUR
        ADD     #1,R1                   ;BUFOFF=0 IS CHANNEL ONE
        CALL    DECOUT                  ;PRINT THE NUMBER
        .PRINT  #M45                    ;A2D GAIN SETTING MESSAGE
        MOVB    A2DGNS,R0               ;GET CHARACTER
        .TTYOUT                         ;TYPE IT
        MOVB    <A2DGNS+1>,R0           ;GET THE OTHER ONE
        .TTYOUT                         ;TYPE IT
        .PRINT  #M39                    ;SCANNING PROCEDURE MESSAGE
        MOVB    SCTYPE,R0               ;GET THE FIRST CHARACTER
        .TTYOUT                         ;SEND TO CONSOLE
        MOVB    <SCTYPE+1>,R0           ;GET THE NEXT ONE
        .TTYOUT                         ;SEND IT
        .PRINT  #M40                    ;WHICH DATA BUFFER DISPLAYED
        MOVB    DICODE,R0               ;GET CHARACTER
        .TTYOUT                         ;SEND TO CONSOLE
        MOVB    <DICODE+1>,R0           ;GET THE OTHER ONE
        .TTYOUT                         ;SEND IT
        CALL    CRLF                    ;CARRIAGE RETURN-LINEFEED
        MOV     R1TYS,R1                ;RESTORE R1
TYSOUT: RETURN                          ;RETURN TO CALLER
;
;
;
;SUBROUTINE TYDATA SENDS DISPLAYED DATA CHANNELS TO THE CONSOLE
;
TYDATA: MOV     R1,R1PRI                ;SAVE THE REGISTERS
        MOV     R2,R2PRI
        MOV     R3,R3PRI
```

```
        MOV     R4,R4PRI
        MOV     R5,R5PRI
        BIS     #10000,JSW                 ;SPECIAL TTY MODE
        MOV     DISTRT,R1                  ;START OF DISPLAY BUFFER
        SUB     DIBUF,R1                   ;R1=BUFFER OFFSET
        MOV     SBBUF,R2                   ;STARTING ADDRESS OF S+B BUFFER
        ADD     R1,R2                      ;ADD OFFSET
        MOV     BABUF,R3                   ;STARTING ADDRESS OF BACKGRD.BUFFER
        ADD     R1,R3
        MOV     POBUF,R4                   ;STARTING ADDRESS OF LASER POWER BUF
FER
        ADD     R1,R4
        MOV     DICHNS,R5                  ;NUMBER OF WINDOW CHANNELS
        .PRINT  #M21                       ;TYPE HEADER
        CALL    CRLF                       ;CARRIAGE RETURN-LINE FEED
        ADD     #4,R1                      ;STARTING CHANNEL NUMBER
        ASH     #-2,R1
10$:    CALL    DECOUT                     ;TYPE THE CHANNEL NUMBER(IN R1)
        .PRINT  #M22                       ;SPACES
        MOV     (R2),LOWORD                ;LOAD SIG.+BACK. FOR OUTPUT
        MOV     2(R2),HIWORD
        CALL    DCOUT2                     ;TYPE THE NUMBER
        .PRINT  #M22                       ;SPACES
        MOV     (R3),LOWORD                ;LOAD BACKGROUND
        MOV     2(R3),HIWORD
        CALL    DCOUT2                     ;TYPE IT
        .PRINT  #M22                       ;SPACES
        MOV     (R2)+,LOWORD               ;CALCULATE THE SIGNAL
        MOV     (R2)+,HIWORD               ;SUBTRACT BACKGROUND FROM SIG.+BACK.
        SUB     (R3)+,LOWORD
        SBC     HIWORD
        SUB     (R3)+,HIWORD
        CALL    DCOUT2                     ;TYPE THE SIGNAL
        .PRINT  #M22                       ;SPACES
        MOV     (R4)+,LOWORD               ;LOAD LASER-POWER FOR OUTPUT
        MOV     (R4)+,HIWORD
        CALL    DCOUT2                     ;TYPE IT!
        CALL    CRLF                       ;CARRIAGE RETURN-LINE FEED
        DEC     R5                         ;COUNT THE LOOPS
        BEQ     30$                        ;BRANCH OUT IF DONE
        INC     R1                         ;INCREMENT CHANNEL NUMBER
        .TTINR                             ;CHECK FOR 'STOP' COMMAND
        BCS     10$                        ;NONE--GO AHEAD
        CMPB    #'S,R0                     ;'S' IS THE STOP COMMAND
        BNE     10$                        ;CONTINUE IF NOT 'S'
30$:    .TTINR                             ;EAT OTHER CHARACTERS
        BCC     30$
        BIC     #10000,JSW                 ;NORMAL TTY MODE
        MOV     R5PRI,R5                   ;RESTORE THE REGISTERS
        MOV     R4PRI,R4
        MOV     R3PRI,R3
        MOV     R2PRI,R2
        MOV     R1PRI,R1
TYDOUT: RETURN                             ;RETURN TO CALLER
;
```

```
;
;
;SUBROUTINE TO OUTPUT THE 'HEADER' TO THE LINE PRINTER
;
PRHEDR: MOV     R1,R1PRH                ;SAVE R1
        MOV     #DATBUF,R1              ;POINT TO DATA BUFFER
        ADD     #100.,R1                ;POINT TO HEADER
10$:    MOVB    (R1)+,R0                ;GET A CHARACTER
        CMPB    #200,R0                 ;IS IT THE TERMINATOR?
        BEQ     20$                     ;BRANCH IF YES
        CALL    LPOUT                   ;SEND CHARACTER TO LINE PRINTER
        BR      10$                     ;GET MORE
20$:    CALL    LPCRLF                  ;RETURN CARRIAGE
        MOV     R1PRH,R1                ;RESTORE R1
PRHOUT: RETURN                          ;RETURN TO CALLER
;
;
;SUBROUTINE TO OUTPUT THE HEADER TO THE CONSOLE
;
TYHEDR: MOV     R1,R1TYH                ;SAVE R1
        MOV     #DATBUF,R1              ;POINT TO DATA BUFFER
        ADD     #100.,R1                ;POINT TO HEADER
10$:    MOVB    (R1)+,R0                ;GET CHARACTER
        CMPB    #200,R0                 ;TERMINATOR?
        BEQ     20$                     ;BRANCH IF YES
        .TTYOUT                         ;CHARACTER OUT
        BR      10$                     ;GET MORE
20$:    CALL    CRLF                    ;SKIP A LINE
        MOV     R1TYH,R1                ;RESTORE R1
TYHOUT: RETURN                          ;RETURN TO CALLER
;
;
;SUBROUTINE TO SEND DATE TO CONSOLE
;
TYDATE: CALL    CRLF                    ;SKIP LINE
        CMPB    #'N,DAY                 ;IS THERE A DATE?
        BEQ     10$
        MOV     DAY,R1
        CALL    DECOUT
        MOV     #' ,R0
        .TTYOUT
        .TTYOUT
        .PRINT  #M66                    ;SEND MONTH
        MOV     YEAR,R1
        ADD     #1900.,R1
        CALL    DECOUT
        BR      20$
10$:    .PRINT  #M64                    ;NO DATE MESSAGE
20$:    CALL    CRLF
        RETURN
;
;
;SUBROUTINE TO SEND DATE TO PRINTER
;
PRDATE: CALL    LPCRLF                  ;SKIP LINE
```

```
        CMPB    #'N,DAY                 ;NO DATE?
        BEQ     10$
        MOV     #M65,R0
        CALL    LPRINT                  ;BACK SPACES
        MOV     DAY,R1
        CALL    LPDECO
        MOV     #'-,R0
        CALL    LPOUT
        MOV     #M66,R0                 ;MONTH
        CALL    LPRINT
        MOV     #'-,R0
        CALL    LPOUT
        MOV     #M65,R0                 ;BACK SPACES
        CALL    LPRINT
        MOV     YEAR,R1
        CALL    LPDECO
        BR      20$
10$:    MOV     #M64,R0                 ;NO DATE MESSAGE
        CALL    LPRINT
20$:    CALL    LPCRLF
        RETURN
;
;END OF PAGE THREE.
```

```
;PAGE FOUR.
;
;SUBROUTINE TO DECODE SYSTEM DATE WORD (RETURNED BY .DATE) AND TO LOAD
;VALUE INTO 'DAY,MONTH,AND YEAR'.(MODIFIED 'DEC' PROGRAM--VOL3 PG 2-48)
;
DATES:  MOV     DATE,R0                         ;COPY DATE WORD
        MOV     R0,R2
        BEQ     10$                             ;BR IF NO DATE
        BIC     #^C37,R2                        ;ISOLATE YEAR
        ADD     #72.,R2                         ;ADD BIAS
        MOV     R2,YEAR
        ASH     #-2,R0
        MOV     R0,R1                           ;COPY THE DATE
        SWAB    R1                              ;MONTH IN LOW BYTE
        BIC     #^C37,R1                        ;ISOLATE MONTH
        ASH     #-3,R0
        BIC     #^C37,R0                        ;ISOLATE DAY
        MOV     R0,DAY
        SUB     #1,R1                           ;MONTH STRING OFFSET
        ASL     R1
        ADD     #R50STR,R1
        MOV     (R1),MONTH                      ;MONTH POINTER
        MOV     #DATLST,R5                      ;ARGUMENT LIST FOR CALL
        CALL    R50ASC                          ;LOAD 'MONTH' WITH ASCII
        BR      20$
10$:    MOVB    #'N,DAY                         ;NO DATE FLAG
20$:    RETURN
;
;
;
;DISPLAY SUBROUTINES
;
XLOAD:  MOV     DISTRT,R1
        MOV     DICHNS,R2
        MOV     XJMP,R3
        NEG     R3
XLOOP:  ADD     XJMP,R3
        MOV     R3,(R1)+
        TST     (R1)+
        SOB     R2,XLOOP
        RETURN
;
;
XJUMP:  MOV     #4096.,R1
        ASH     #2,R1
        CLR     R0
        DIV     DICHNS,R0
        ASH     #-2,R0
        MOV     R0,XJMP
        RETURN
;
;
;DISPLAY CONTROL SUBROUTINE
;
DISDAT: CALL    DISHLT                          ;HALT DISPLAY
```

```
        MOV     #'00,IOBUF              ;PLACE ZEROES IN FIRST WORD
        CMPB    #15,IOBUF+2             ;CARIAGE RETURN?
        BEQ     10$
        BR      15$
10$:    .PRINT  #M58                    ;ASK FOR VALUE
        CALL    READ                    ;READ REPLY
15$:    CALL    ASCBIN                  ;CONVERT TO BINARY
        BCC     20$
        .PRINT  #M4                     ;ERROR MESSAGE
        BR      10$
20$:    ASH     #2,R1                   ;TIMES FOUR
        MOV     R1,DISNUM               ;SAVE DISPLAY NUMBER
        SUB     #4,DISNUM               ;OFFSET
        TST     DISNUM                  ;POSITIVE?
        BPL     25$
        MOV     #0,DISNUM
25$:    JMP     @DISCOM                 ;GO TO COMMAND
;
;
WL:     SUB     R1,DISWIN               ;SHIFT WINDOW LEFT
        CMP     DISWIN,DIBUF            ;WITHIN BOUNDS?
        BPL     10$
        MOV     DIBUF,DISWIN
        CMP     #0,SLRFLG               ;SCROLLING?
        BEQ     10$
        NEG     SLRFLG                  ;SCROLL RIGHT
10$:    BR      DWCHK
;
;
WR:     ADD     R1,DISWIN
        MOV     WINSIZ,R2
        ASH     #2,R2
        ADD     DISWIN,R2
        CMP     DISMAX,R2
        BPL     10$
        SUB     R1,DISWIN
        CMP     #0,SLRFLG               ;SCROLLING?
        BEQ     10$
        NEG     SLRFLG                  ;SCROLL LEFT
10$:    BR      DWCHK                   ;CHECK WINDOW SIZE
;
;
WI:     ADD     DIBUF,DISNUM            ;MEMORY POINTER
        CMP     DISMAX,DISNUM           ;TOO LARGE?
        BPL     10$
        MOV     DISMAX,DISNUM
10$:    MOV     DISNUM,DISWIN           ;STARTING ADDRESS
        BR      DWCHK                   ;CHECK WINDOW SIZE
;
;
WA:     ADD     #4,DISNUM               ;SELECT WINDOW SIZE
        MOV     DISNUM,R1
        ASH     #-2,R1
        MOV     R1,WINSIZ               ;NEW WINDOW SIZE
DWCHK:  MOV     DISMAX,R1               ;CHECK WINDOW SIZE COMPATABILITY
```

```
        SUB     DISWIN,R1
        ASH     #-2,R1
        CMP     R1,WINSIZ
        BPL     30$
        MOV     R1,WINSIZ
30$:    CMP     #0,EXPFLG                ;DISPLAY CONTRACTED?
        BNE     DICOUT
        MOV     #1,EXPFLG
;
;
DE:     CALL    DISHLT                   ;HALT DISPLAY
        TST     EXPFLG                   ;EXPAND OR CONTRACT?
        BEQ     20$
        CLR     EXPFLG                   ;EXPAND
        MOV     WINSIZ,DICHNS
        MOV     DISWIN,DISTRT
        CALL    XJUMP
        CALL    XLOAD
        BR      DICOUT
20$:    MOV     #1,EXPFLG                ;CONTRACT
        MOV     NCHAN,DICHNS
        MOV     DIBUF,DISTRT
        CALL    XJUMP
        CALL    XLOAD
        BR      DICOUT
;
;
;
WB:     ASH     #-2,R1                   ;DISPLAY BRIGHTNESS
        TST     R1
        BGE     10$
        CLR     R1
10$:    MOV     R1,WINBRT
        BR      DICOUT
;
;
;
;
AB:     ASH     #-2,R1                   ;ACTIVE DISPLAY BRIGHTNESS
        TST     R1
        BGE     10$
        CLR     R1
10$:    MOV     R1,ACTIVE+2
        BR      DICOUT
;
DICOUT: MOV     #1,DMAHLT                ;DMA GO
        MOV     DISTRT,DMACAR            ;LOAD DMA REGISTERS
        MOV     DICHNS,DMAPCR
        NEG     DMAPCR                   ;2'S COMPLEMENT
        BIS     #102,ADCSR               ;ENABLE DMA INT
        BIS     #1,ADCSR                 ;START DMA
        RETURN
;
;
;SUBROUTINE TO SCROLL WINDOW
```

```
;
SLR:    MOV     #4,R1                   ;SCROLL LEFT
        CALL    DISHLT                  ;STOP DMA
        CMP     #1,SLRFLG               ;SCROLL RIGHT?
        BEQ     1$
        CALL    WL                      ;SHIFT DISPLAY LEFT
        BR      5$
1$:     CALL    WR                      ;SHIFT DISPLAY RIGHT
5$:     MOV     SCROLL,R1               ;WAIT LOOP
10$:    SOB     R1,10$
        BIS     #10000,JSW              ;SPECIAL TTY MODE
        .TTINR                          ;CHECK FOR 'STOP' COMMAND
        BCS     SLR                     ;NONE--GO AHEAD
        CMPB    #'S,R0                  ;'S' IS THE STOP COMMAND
        BNE     SLR                     ;CONTINUE IF NOT 'S'
20$:    .TTINR                          ;EAT THE OTHER CHARACTERS
        BCC     20$
        BIC     #10000,JSW              ;NORMAL TTY MODE
        RETURN
;
;
;YAXIS PACKAGE SETS Y-AXIS SCALING OF DISPLAYED DATA.
;
YAXIS:  JMP     @YFUNC                  ;GO TO SCALING FUNCTION
;
;
;SUBROUTINE TO SET AUTO-SCALING OF DISPLAY OR ABSOLUTE SCALING.
;
AS:     MOV     #'00,IOBUF              ;ZEROES FIRST WORD
        CMPB    #15,IOBUF+2             ;CR?
        BEQ     10$
        CMP     #'AS,IOBUF+2            ;AUTO-SCALE?
        BEQ     30$
        BR      12$
10$:    .PRINT  #M59                    ;Y-MAX
        CALL    READ
        CMP     #'AS,IOBUF              ;AUTO SCALE?
        BEQ     30$
12$:    CALL    ASCBIN
        BCC     20$
15$:    .PRINT  #M4                     ;BAD ENTRY
        BR      10$
20$:    TST     R2
        BLT     15$
        MOV     R1,YMAX                 ;LOW WORD
        MOV     R2,YMAX+2               ;HIGH WORD
        MOV     #3,DISFLG               ;SCALING FLAG
25$:    MOV     #2048.,YSCALE
        CLR     UPDATE
        RETURN
30$:    MOV     #0,DISFLG
        BR      25$
;
WS:     MOV     #1,DISFLG               ;WINDOW SCALING
        MOV     #2048.,YSCALE
```

```
        CLR     UPDATE
        RETURN
;
;
;SCALE DOWN DISPLAY BY 1/2
;
SD:     ASR     YSCALE
        BNE     SEX
        MOV     #1,YSCALE
SEX:    .PRINT  #M60                    ;YSCALE:
        MOV     YSCALE,R1
        CALL    DECOUT
        CALL    CRLF
        CLR     UPDATE
        RETURN
;
;SCALE UP DISPLAY BY 2
;
SA:     ASL     YSCALE
        BPL     SEX
        MOV     #32767.,YSCALE
        BR      SEX
;
;SCALE DISPLAY SO SPECIFIED CHANNEL=100.
;
CS:     MOV     #'00,IOBUF              ;ZEROES IN FIRST WORD
        CMPB    #15,IOBUF+2             ;CR?
        BEQ     10$
        BR      12$
10$:    .PRINT  #M61                    ;CHANNEL:
        CALL    READ
12$:    CALL    ASCBIN
        BCC     20$                     ;BR IF OK
15$:    .PRINT  #M4                     ;BAD ENTRY
        BR      10$
20$:    CMP     R1,#1
        BLT     15$
        CMP     NCHAN,R1
        BLT     15$
        DEC     R1                      ;ADDRESS OFFSET
        ASH     #2,R1
        MOV     R1,CSCHAN
        CLR     UPDATE
        MOV     #2048.,YSCALE
        MOV     #2,DISFLG
        RETURN
;TYPE DISPLAY PARAMETERS
;
DP:     .PRINT  #M59                    ;YMAX:
        MOV     YMAX,LOWORD
        MOV     YMAX+2,HIWORD
        CALL    DCOUT2
        CALL    CRLF
        .PRINT  #M73                    ;YMIN:
        MOV     YMIN,LOWORD
```

```
        MOV     YMIN+2,HIWORD
        CALL    DCOUT2
        CALL    CRLF
        .PRINT  #M62                    ;WINDOW SILL
        MOV     DISWIN,R1
        SUB     DIBUF,R1                ;FIND CHANNEL NUMBER
        ADD     #4,R1
        ASH     #-2,R1
        CALL    DECOUT
        CALL    CRLF
        .PRINT  #M63                    ;WINDOW SIZE
        MOV     WINSIZ,R1
        CALL    DECOUT
        CALL    CRLF
        .PRINT  #M60                    ;YSCALE
        MOV     YSCALE,R1
        CALL    DECOUT
        CALL    CRLF
        BIT     #2,FLGDIS               ;REVERSED DISPLAY?
        BEQ     10$
        .PRINT  #M78                    ;DISPLAY REVERSED
        BR      20$
10$:    .PRINT  #M79                    ;DISPLAY NORMAL
20$:    BIT     #1,FLGDIS               ;POWER NORMALIZED?
        BEQ     30$
        .PRINT  #M80                    ;YES
        BR      40$
30$:    .PRINT  #M81                    ;NO
40$:    RETURN
;
;
RE:     BIT     #2,FLGDIS               ;DIPLAY FORWARD OR REVERSE?
        BEQ     10$                     ;CHANGE IT
        BIC     #2,FLGDIS               ;GO FORWARD
        .PRINT  #M79
        CLR     UPDATE                  ;CAUSE DISPLAY UPDATE
        RETURN
10$:    BIS     #2,FLGDIS               ;GO 'BACKWARDS'
        .PRINT  #M78
        CLR     UPDATE
        RETURN
;
;
PN:     BIT     #1,FLGDIS               ;POWER NORMALIZE
        BEQ     10$                     ;PRESENTLY?
        BIC     #1,FLGDIS               ;DO NOT POWER NORM.
        .PRINT  #M81
        CLR     UPDATE                  ;CAUSE DISPLAY UPDATE
        RETURN
10$:    BIS     #1,FLGDIS               ;POWER NORM.
        .PRINT  #M80
        CLR     UPDATE
        RETURN
;
;
```

```
;SUBROUTINE TO SET 'KEY' INTERRUPT MODE
;
KM:     BIS     #1,SCMODE               ;KEY-INT. MODE
        MOV     ACCFLG,ROSMO            ;SAVE ACCUM. CODE
        MOV     #1,ACCFLG               ;SET TO ACUME MODE
        MOVB    #24.,A2DCON             ;DISABLE A2D CONVERTER
        .PRINT  #M92                    ;KEY INT. MODE
        .PRINT  #M96                    ;KEY MODE READY
        MOV     #DATAIN,ROUTINE         ;SUBROUTINE LOCATION
        CALL    KEYHOL                  ;ACTIVATE 'KEYS'
        BIC     #1,SCMODE               ;EXT. INT. MODE
        MOVB    #26.,A2DCON             ;ENABLE A2D CONVERTER
        MOV     ROSMO,ACCFLG            ;RESTORE ACCUME FLAG
        .PRINT  #M93
        RETURN
;
;
;INITIALIZE TO CALL FORTRAN PLOTTR SUBROUTINE
;
PLOTPH: MOV     #0,PLMOV                ;DEFAULT VALUES
        MOV     #';,PLLINE              ;SOLID LINE
        BR      PLOTIT
PLOTPV: MOV     #0,PLMOV                ;X-COORD.(CHANNEL)
        MOV     #';,PLLINE              ;SOLID LINE
        BR      PLOTIT
PLOTPC: MOV     #12.,PLMOV              ;Y-COORD.
        BR      PLOTIT
PLOTPM: MOV     #10.,PLMOV              ;Y-COORD.
        BR      PLOTIT
;
PLOTIT: MOV     IOBUF,PLCTRL            ;CONTROL WORD
        CMPB    #15,<IOBUF+2>           ;CR?
        BEQ     PLOTEX
        MOV     #IOBUF,R3               ;POINTERS
        MOV     #<IOBUF+2>,R0
10$:    MOVB    (R0),(R3)               ;SHIFT
        CMPB    #15,(R3)                ;CR?
        BNE     30$
15$:    CALL    ASCBIN                  ;CONVERT TO BINARY
        BCC     20$                     ;BR IF OK
        BR      PLOTEX                  ;DEFAULT
20$:    MOV     R1,PLMOV                ;ENTERED VALUE
        BR      PLOTEX
30$:    CMPB    #54,(R0)+               ;COMMA?
        BEQ     40$
        TSTB    (R3)+
        BR      10$                     ;CONTINUE
40$:    MOVB    (R0),PLLINE             ;ENTERED VALUE
        MOVB    #15,(R3)
        BR      15$
;
PLOT:   MOV     IOBUF,PLCTRL            ;PLOTTR CONTROL
        CLR     PLMOV
        CLR     PLCHAR
        CLR     PLLINE
```

```
        CMPB    IOBUF+2,#15             ;CR?
        BEQ     10$                     ;IF YES,DEFAULT
        MOVB    IOBUF+2,PLCHAR          ;PLOT CHARACTER
        CMPB    IOBUF+3,#15             ;CR?
        BEQ     20$                     ;YES-THEN DEFAULT
        MOVB    IOBUF+3,PLLINE          ;LINE TYPE
        BR      PLOTEX
10$:    MOV     #';,PLCHAR              ;DEFAULTS
20$:    MOV     #'0,PLLINE
PLOTEX: CALL    DISHLT                  ;HALT DMA
        MOV     DISTRT,R1               ;STARTING-CHANNEL NUMBER
        SUB     DIBUF,R1
        ADD     #4,R1
        ASH     #-2,R1
        MOV     R1,STCHAN
        MOV     #PLLIST,R5              ;ARGUMENT LIST FOR CALL
        CALL    PLOTTR
        MOV     #1,DMAHLT               ;ALLOW DMA TO START
        BIS     #103,ADCSR              ;START DMA DISPLAY DRIVER
        RETURN
;
;
;SUBROUTINE TO FIND THE INTEGRAL OF WINDOWED CHANNELS.
;
IW:     MOV     DISWIN,R1               ;POINT TO FIRST CHANNEL
        SUB     DIBUF,R1
        CLR     LOWORD
        CLR     HIWORD
        MOV     WINSIZ,R4               ;WINDOW SIZE
        CMP     #"SI,DICODE             ;SIGNAL DISPLAYED?
        BEQ     10$
        CMP     #"SB,DICODE             ;SIG.+BAK.?
        BEQ     20$
        CMP     #"BA,DICODE             ;BAK.?
        BEQ     30$
        CMP     #"PO,DICODE             ;POWER?
        BEQ     40$
        JMP     IWOUT
10$:    MOV     BABUF,R2
        ADD     R1,R2
        ADD     SBBUF,R1
15$:    ADD     (R1)+,LOWORD
        ADC     HIWORD
        BVS     60$                     ;OVERFLOW?
        ADD     (R1)+,HIWORD
        BVS     60$                     ;OVERFLOW?
        SUB     (R2)+,LOWORD
        SBC     HIWORD
        BVS     60$                     ;OVERFLOW?
        SUB     (R2)+,HIWORD
        BVS     60$                     ;OVERFLOW?
        SOB     R4,15$
        BR      50$
20$:    ADD     SBBUF,R1
25$:    ADD     (R1)+,LOWORD
```

```
        ADC     HIWORD
        BVS     60$                     ;OVERFLOW?
        ADD     (R1)+,HIWORD
        BVS     60$                     ;OVERFLOW?
        SOB     R4,25$
        BR      50$
30$:    ADD     BABUF,R1
35$:    ADD     (R1)+,LOWORD
        ADC     HIWORD
        BVS     60$                     ;OVERFLOW?
        ADD     (R1)+,HIWORD
        BVS     60$                     ;OVERFLOW?
        SOB     R4,35$
        BR      50$
40$:    ADD     POBUF,R1
45$:    ADD     (R1)+,LOWORD
        ADC     HIWORD
        BVS     60$                     ;OVERFLOW?
        ADD     (R1)+,HIWORD
        BVS     60$                     ;OVERFLOW?
        SOB     R4,45$
        BR      50$
50$:    .PRINT  #M69                    ;INTEGRAL=
        CALL    DCOUT2
        CALL    CRLF
        BR      IWOUT
60$:    .PRINT  #M68                    ;OVERFLOW
IWOUT:  RETURN
;
;SCROLL SPEED UP BY 2
;
SF:     ASR     SCROLL
        BNE     SEXY
        MOV     #1,SCROLL
SEXY:   .PRINT  #M70                    ;SCROLL:
        MOV     SCROLL,R1
        CALL    DECOUT
        CALL    CRLF
        RETURN
;
;SCROLL SPEED DOWN BY 1/2
;
SS:     ASL     SCROLL
        BPL     SEXY
        MOV     #32767.,SCROLL
        BR      SEXY
;
;
;SUBROUTINE TO SUBTRACT A CONSTANT FROM DATA
;
SZ:     CMPB    #15,IOBUF+2             ;CR?
        BEQ     20$
        MOV     #'00,IOBUF              ;ZEROES IN FIRST WORD
        MOV     #<IOBUF+2>,R0           ;POINTER
1$:     CMPB    #'-,(R0)                ;MINUS?
```

```
        BNE     5$
        MOVB    (R0),IOBUF              ;MOVE TO FRONT
        MOVB    #'0,(R0)                ;REPLACE WITH ZERO
        BR      8$
5$:     CMPB    #15,(R0)+               ;CR?
        BNE     1$
8$:     CALL    ASCBIN                  ;CONVERT TO BINARY
        BCC     10$                     ;OK?
        .PRINT  #M4                     ;BAD ENTRY
        BR      SZOUT
10$:    MOV     R1,YMIN                 ;LOW WORD
        MOV     R2,YMIN+2               ;HIGH WORD
        MOV     #2,ZERFLG               ;SHIFT FLAG = 2
        BR      SZOUT
20$:    TST     ZERFLG
        BEQ     30$
        CLR     ZERFLG                  ;SHIFT FLAG = 0
        CLR     YMIN
        CLR     YMIN+2
        BR      SZOUT
30$:    MOV     #1,ZERFLG               ;SHIFT FLAG = 1
SZOUT:  CLR     UPDATE                  ;UPDATE DISPLAY
        RETURN
;
;
;MULTICHANNEL SCALER SUBROUTINE
;
MCS:    CMP     #'MC,SCCODE             ;IS MCS ACTIVE?
        BEQ     1$
        .PRINT  #M84                    ;DEVICE IS NOT ACTIVE
        RETURN
1$:     MOV     IOBUF+2,IOBUF           ;MOVE IN SECOND COMMAND
        CMPB    IOBUF,#15               ;A CR?
        BEQ     3$
        BR      5$
3$:     .PRINT  #M10                    ;REQUEST OPTION
        CALL    READ                    ;GET RESPONSE
5$:     CMP     #'IN,IOBUF              ;INIT?
        BNE     10$
        JMP     MCSIN
10$:    CMP     #'ST,IOBUF              ;START?
        BNE     15$
        JMP     MCSST
15$:    CMP     #'ZZ,IOBUF              ;LOAD AGAIN?
        BNE     20$
        JMP     MCSZZ
20$:    CMP     #'HA,IOBUF              ;HALT MCS?
        BNE     25$
        JMP     MCSHA
25$:    CMP     #'MS,IOBUF              ;MORE SWEEPS?
        BNE     30$
        JMP     MCSMS
30$:    CMP     #'OF,IOBUF              ;MCS UNIT 'OFF'?
        BNE     35$
        JMP     MCSOF
```

```
35$:    CMP     #'SB,IOBUF              ;PLACE DATA IN S+B BUFFER?
        BNE     40$
        MOV     SBBUF,MCSBUF            ;LOAD BUFFER LOCATION
        RETURN
40$:    CMP     #'BA,IOBUF              ;PLACE DATA IN BACK. GR'D. BEFFER?
        BNE     45$
        MOV     BABUF,MCSBUF            ;LOAD MCSBUF POINTER
        RETURN
45$:    CMP     #'CL,IOBUF              ;CLEAR MCS RAM?
        BNE     50$
        .PRINT  #M99                    ;CLEAR RAM?
        .PRINT  #M48                    ;ARE YOU SURE?
        CALL    READ                    ;GET REPLY
        CMPB    #'Y,IOBUF               ;YES?
        BEQ     450$
        RETURN
450$:   JMP     MCSCL                   ;GO DO IT
50$:    JMP     3$                      ;TRY ANOTHER COMMAND
;
;
MCSIN:  MOVB    #9.,MCSA0               ;ALL CLEAR
        MOV     #100.,R0                ;WAIT FOR MEMORY CLEAR
1$:     SOB     R0,1$
        MOVB    #26.,MCSA0
        .PRINT  #M87                    ;DWELL TIME
        CALL    MCSINP
        MOV     BCD,DWLBCD
        MOV     DWLBCD,DATALO
        MOVB    #18.,MCSA0              ;LOAD DWELL TIME
        MOV     BIN,DWLBIN
        .PRINT  #M88                    ;TRIGGER COUNT
        CALL    MCSINP
        MOV     BCD,TRGBCD
        MOV     TRGBCD,DATALO
        MOVB    #18.,MCSA2              ;LOAD TRIGGER COUNT
        MOV     BIN,TRGBIN
        .PRINT  #M89                    ;SWEEP COUNT
        CALL    MCSINP
        MOV     BCD,SWPBCD
        MOV     BIN,SWPBIN
        CLR     DATALO
        MOVB    #18.,MCSA1              ;LOAD INITIAL ADDRESS
3$:     MOV     SWPBCD,DATALO           ;LOAD SWEEP COUNTER
        BIS     #1,STATU1
        MOVB    #18.,MCSA3              ;LOAD SWEEP COUNT
        BIC     #1,STATU1               ;CLR DII FOR COMPARE
        CMP     MCSA3,SWPBCD
        BNE     3$
        CLR     NSCAN                   ;CLEAR SCAN COUNTER
        BIS     #1,STATU1
        MOVB    #18.,MCSA4              ;SET INTERNAL DWELL CLOCK
        RETURN
;
;
MCSZZ:  MOV     DWLBCD,DATALO           ;TRY LOADING AGAIN
```

```
        MOVB    #18.,MCSA0              ;LOAD DWELL
        MOV     TRGBCD,DATALO
        MOVB    #18.,MCSA2              ;LOAD TRIGGER
5$:     MOV     SWPBCD,DATALO
        BIS     #1,STATU1               ;SET DI1 BIT AFOR LOOP BACK AFTER CL
EARING
        MOVB    #18.,MCSA3              ;F18A3 LOAD SWEEP COUNTER
        BIC     #1,STATU1               ;CLEAR DII BIT FOR CMP OPERATION
        CMP     MCSA3,SWPBCD            ;F0A3 COMPARE SWEEP COUNTER
        BNE     5$                      ;RELOAD UNTIL EQUAL
        BIS     #1,STATU1               ;RESET DII
        RETURN
;
;
MCSST:  MOVB    #26.,MCSA1              ;F26A1, ENABLE LAM
        MOVB    #26.,MCSA6              ;F26A6, SET ACCUMULATE
        BIS     #301,STATU1             ;SET REIN,INE,DII
        RETURN
;
;
MCSMS:  .PRINT  #M90                    ;ADDITIONAL SWEEPS
        MOVB    #26.,MCSA0
        CALL    MCSINP
        MOV     BCD,SWPBCD
        MOV     BIN,SWPBIN
3$:     MOV     SWPBCD,DATALO
        BIS     #1,STATU1               ;SET DII BI1 FOR KLOOP BACK AFTRER C
LEARING
        MOVB    #18.,MCSA3              ;F18A3 LOAD SWEEP COUNTER
        BIC     #1,STATU1               ;CLEAR DII FOR CMP OPERATION
        CMP     MCSA3,SWPBCD            ;F0A3 COMPARE SWEEP COUNTER
        BNE     3$                      ;CHECK FOR EQUALITY, RELOAD IF NOT
        BIS     #1,STATU1               ;RESET DII BIT
        MOVB    #26.,MCSA6              ;SET ACCUM
        RETURN
;
;
MCSHA:  MOVB    #26.,MCSA0              ;HALT MCS
        RETURN
;
;
MCSCL:  MOVB    #26.,MCSA0              ;HALT MCS--BLOCK TRIGGERS
        MOVB    #9.,MCSA0               ;CLEAR RAM MEMMORY
        RETURN
;
;
MCSOF:  MOVB    #26.,MCSA0              ;HALT MCS UNIT
        MOVB    #26.,A2DCON             ;ENABLE A2D INTERRUPT
        MOV     #4,ACTIVE+2             ;ACTIVE CHANNEL BR'TNESS
        MOV     #'NO,SCCODE             ;RETURNS IN 'NOTHING' MODE
        MOV     #NO,SCFUNC              ;'NOTHING' FUNCTION
        .PRINT  #M95
        RETURN
;
;
```

```
;MCS INPUT SUBROUTINE
;
MCSINP: CLR     NEGA
        CLR     FLN
        CLR     BCD
        CLR     R1
        CLR     CNT
1$:     .TTYIN
        CMP     R0,#56
        BNE     2$
        INC     FLN
        BR      1$
2$:     CMP     R0,#55
        BNE     3$
        INC     NEGA
        BR      1$
3$:     CMP     R0,#15
        BEQ     4$
        CMP     R0,#60
        BLT     1$
        CMP     R0,#71
        BGT     1$
        INC     CNT
        CMP     #5,CNT
        BLE     1$
        MOV     BCD,R2
        ASH     #4,R2
        SUB     #60,R0
        ADD     R0,R2
        MOV     R2,BCD
        MUL     #12,R1
        ADD     R0,R1
        BR      1$
4$:     .TTYIN
        MOV     R1,BIN
        RETURN
;
;
EXIT:   CLR     DMAHLT                      ;HALT DMA
10$:    BIT     #100,ADCSR                  ;HALTED?
        BNE     10$                         ;WAIT TILL YES
        CLR     KWCSR                       ;DIABLE KWV11-A
        BIS     #4,STATU1                   ;CAMAC Z-CYCLE
        RETURN
;
;
;
;
;       THIS SECTION CONTAINS THIS MODULE'S MESSAGES TO THE USER
;
;
M1:     .ASCII<15><12>/LOW MASS = /<200>
M2:     .ASCII<15><12>/HIGH MASS = /<200> '
M3:     .ASCII<15><12>/CHANNELS PER MASS UNIT = /<200>
M4:     .ASCIZ<15><12>/INVALID ENTRY!!  TRY AGAIN./
```

```
M5:     .ASCII<15><12>/WHICH DEVICE? /<200>
M6:     .ASCII<15><12>/NUMBER OF DATA CHANNELS = /<200>
M7:     .ASCIZ<15><12>/YOUR CURRENT OPTIONS ARE THE FOLLOWING:/
M8:     .ASCIZ<15><12>/LASER, TRAP, QUADRUPOLE, MAGNET, MCS UNIT, AND NOTHIN
G/
M9:     .ASCII<15><12>/BIDIRECTIONAL OR UNIDIRECTIONAL SCANNING? /<200>
M10:    .ASCII<15><12>/    OPTION: /<200>
M11:    .ASCIZ<15><12>/SIGNAL, BACKGROUND, SB (SIG.+ BAK.), POWER/
M12:    .ASCIZ<15><12>/    SECTOR MAGNET MASS /<15><12>
M13:    .ASCII<15><12>/LOW CURRENT FACTOR:  /<200>
M14:    .ASCII<15><12>/NUMBER OF SCANS = /<200>
M15:    .ASCII<15><12>/HIGH CURRENT FACTOR: /<200>
M16:    .ASCII<15><12>/NUMBER OF POINTS TO BE AVERAGED = /<200>
M17:    .ASCII<15><12>/TRAP DELAY OFFSET =/<200>
M18:    .ASCII<15><12>/TRAP DELAY INCREMENT =/<200>
M19:    .ASCII<15><12>/STEPS PER CHANNEL:   /<200>
M20:    .ASCIZ<15><12>/    QUAD: PEAKS OPTION/<15><12>
M21:    .ASCIZ/CHANNEL    SIG.+BACK.     BACKGROUND        SIGNAL    LASER
POWER/
M22:    .ASCII  /    /<200>
M23:    .ASCII<7><200>
M23A:   .ASCIZ<15><12>/**** RUN COMPLETED ****/<15><12>
M24:    .ASCIZ<15><12>/    DYE LASER FREQUENCY/<15><12>
M25:    .ASCII  /THE SCAN STARTS AT WAVELENGTH:   /<200>
M26:    .ASCII  /THE SCAN ENDS AT WAVELENGTH:     /<200>
M27:    .ASCII  /WAVELENGTH INCREMENT PER CHANNEL:/<200>
M28:    .ASCII  /NUMBER OF MOTOR STEPS PER INCREMENT:  /<200>
M29:    .ASCIZ<15><12>/    ION TRAP STORAGE TIME/
M30:    .ASCIZ<15><12>/    QUAD. MASS SPEC./<15><12>
M31:    .ASCII  /SCAN STARTS AT ABOUT MASS:/<200>
M32:    .ASCII  /SCAN ENDS AT ABOUT MASS:  /<200>
M34:    .ASCIZ<15><12>/    NO DEVICE/<15><12>
M35:    .ASCII  /LASER WAVELENGTH SETTING:/<200>
M36:    .ASCII  /NUMBER OF DATA CHANNELS:  /<200>
M37:    .ASCII  /NUMBER OF SCANS COMPLETED:/<200>
M38:    .ASCII  /PRESENT CHANNEL NUMBER:   /<200>
M39:    .ASCII<15><12>/SCANNING PROCEDURE:           /<200>
M40:    .ASCII<15><12>/DISPLAY BUFFER:               /<200>
M41:    .ASCII<15><12>/DATA CYCLES PER CHANNEL:  /<200>
M42:    .ASCII<15><12>/ION TRAP DELAY INCREMENT: /<200>
M43:    .ASCII<15><12>/ION TRAP DELAY OFFSET:    /<200>
M44:    .ASCIZ<15><12>/************  WATCH YOUR LANGUAGE!!!  ************/
M45:    .ASCII<15><12>/A2D CONVERTER GAIN SETTING:    /<200>
M46:    .ASCII<15><12>/RESET DEVICE:  /<200>
M47:    .ASCII<15><12>/READ THE DISK:  /<200>
M48:    .ASCII  /ARE YOU SURE? /<200>
M49:    .ASCII<15><12>/CLEAR DATA BUFFER:  /<200>
M50:    .ASCIZ<15><12>/THIS OPTION IS 'LOCKED'./
M51:    .ASCIZ<15><12>/NONEXISTENT OPTION./
M52:    .ASCIZ<15><12>/   *** LOCKED ***/
M53:    .ASCIZ<15><12>/   *** UNLOCKED ***/
M54:    .ASCII<15><12>/DO YOU WISH TO ENTER A HEADER MESSAGE? /<200>
M55:    .ASCII<15><12>/PULSE SEQUENCE INIT.:  /<200>
M56:    .ASCIZ<15><12>/ENTER HEADER MESSAGE./
M57:    .ASCIZ<15><12>/MESSAGE TOO LONG--MESSAGE WAS TRUNCATED./
```

```
M58:    .ASCII<15><12>/ENTER REQUIRED NUMBER: /<200>
M59:    .ASCII<15><12>/YMAX: /<200>
M60:    .ASCII<15><12>/YSCALE: /<200>
M61:    .ASCII<15><12>/CHANNEL: /<200>
M62:    .ASCII<15><12>/WINDOW START: /<200>
M63:    .ASCII<15><12>/WINDOW SIZE: /<200>
M64:    .ASCII/    NO DATE/<200>
M65:    .ASCII  <10><10><10><10><200>
M66:    .BYTE   0,0,0,200
M67:    .ASCII  /    RUN NUMBER: /<200>
M68:    .ASCIZ<15><12>/OVERFLOW!!/<15><12>
M69:    .ASCII<15><12>/    WINDOW INTEGRAL=/<200>
M70:    .ASCII<15><12>/SCROLL: /<200>
M71:    .ASCII<15><12>/EXIT PROGRAM!!? /<200>
M72:    .ASCII  / ENTER NUMBER OF CHANNELS BETWEEN UPDATE: /<200>
M73:    .ASCII<15><12>/YMIN: /<200>
M74:    .ASCIZ<15><12>/*** WARNING!! *** --- SCANNING PROCEDURE CODE MISSING
!!/
M75:    .ASCII / ENTER CLOCK UNITS (0,1,2,OR 3): /<200>
M76:    .ASCIZ<15><12>/   *** RE-TAKE MODE ***/
M77:    .ASCIZ<15><12>/   *** ACCUME MODE ***/
M78:    .ASCIZ<15><12>/   DISPLAY REVERSED/
M79:    .ASCIZ<15><12>/   DISPLAY NORMAL/
M80:    .ASCIZ<15><12>/   'SIGNAL' IS POWER NORMALIZED./
M81:    .ASCIZ<15><12>/   'SIGNAL' IS NOT POWER NORMALIZED./
M82:    .ASCIZ<15><12>/COUNT OVERFLOW /
M83:    .ASCIZ<15><12>/RAM OVERFLOW /
M84:    .ASCIZ<15><12>/DEVICE IS NOT ACTIVE/
M85:    .ASCIZ<15><12>/    MCS UNIT ACTIVE/
M86:    .ASCIZ<15><12>/SYNTAX ERROR/
M87:    .ASCII<15><12>/  DWELL TIME=/<200>
M88:    .ASCII<15><12>/  TRIGGER COUNT=/<200>
M89:    .ASCII<15><12>/  SWEEP COUNT=/<200>
M90:    .ASCII<15><12>/  NUMBER OF ADDITIONAL SWEEPS = /<200>
M91:    .ASCIZ<15><12>/*** DEVICE ALREADY ACTIVE ***/
M92:    .ASCIZ<15><12>/*** KEY-INT. MODE ***/
M93:    .ASCIZ<15><12>/*** EXT. INT. MODE ***/
M94:    .ASCIZ<15><12>/*** KEY-INT. MODE IS ACTIVE ***/
M95:    .ASCIZ<15><12>/*** A2D CONVERTER ENABLED ***/
M96:    .ASCII<15><12>/KEY MODE READY: /<200>
M97:    .ASCII<15><12>/TRIGGERS PER SCAN:        /<200>
M98:    .ASCII<15><12>/ACTIVATE MCS UNIT:  /<200>
M99:    .ASCII<15><12>/CLEAR MCS RAM:  /<200>
        .EVEN
;
;
;
;
;       LABEL DEFINITIONS
;
ACCFLG: .WORD                           ;ACCUMULATE FLAG
ACTIVE: .BLKW   2                       ;ACTIVE CHAN. BR'TNESS
ARG1:   .BLKW   2                       ;SYSF4 ARGUMENT ONE
ARG2:   .BLKW   2                       ;SYSF4 ARGUMENT TWO
BABUF:  .WORD                           ;BACKGROUND BUFFER POINTER
```

```
BCD:    .WORD                                   ;MCS UNIT
BIN:    .WORD                                   ;MCS UNIT
BUFOFF: .WORD                                   ;DATA BUFFER OFFSET
CALWAV: .BLKW   2                               ;CALIBRATION WAVELENGTH
CNT:    .WORD                                   ;MCS UNIT
CSCHAN: .WORD                                   ;SCALING CHANNEL
DACYCL: .BLKW   2                               ;WORDS FOR NUMBER OF DATA CYCLES/CHA
NNEL
DATE:   .WORD                                   ;PROGRAM DATE WORD
DATLST: .WORD   3                               ;LIST FOR 'R50ASC' SYSF4 CALL
        .WORD   <DATLST+8.>
        .WORD   MONTH
        .WORD   M66
        .WORD   3
DAY:    .WORD                                   ;DAY OF THE MONTH(DATE)
DELINC: .WORD
DELOFF: .WORD                                   ;ION STORAGE TIME OFFSET
DELTOT: .WORD                                   ;TOTAL ION STORAGE TIME
DIBUF:  .WORD                                   ;DISPLAY BUFFER POINTER
DICHNS: .WORD
DIFUNC: .BLKW   2                               ;DISPLAY FUNCTION WORD
DIRFLG: .WORD                                   ;SCAN DIRECTION FLAG
DICODE: .BLKW   2                               ;DISPLAY CODE WORD
DISCOM: .WORD                                   ;DISPLAY COMMAND WORD
DISFLG: .WORD                                   ;DISPLAY YAXIS SCALE FLAG
DISMAX: .WORD                                   ;DISPLAY BUFFER MAX
DISNUM: .WORD                                   ;DISPLAY COMMAND NUMBER
DISTRT: .WORD                                   ;DISPLAY START-FOR XLOAD
DISWIN: .WORD                                   ;DISPLAY WINDOW STARTING ADDRESS
DMAHLT: .WORD                                   ;DMA 'HALT' WORD
DWLBCD: .WORD
DWLBIN: .WORD                                   ;MCS UNIT
EXPFLG: .WORD                                   ;DISPLAY EXPAND FLAG
FLGDIS: .WORD                                   ;SECOND DISPLAY FLAG.
FLN:    .WORD                                   ;MCS UNIT
LASHOTS:.BLKW   2                               ;NUMBER OF LASER SHOTS
LCHAN:  .WORD                                   ;# OF ADDRESSES TO LAST CHANNEL
LCHN:   .WORD                                   ;LAST AVERAGED DATA CHANNEL
LIST:   .WORD   3                               ;ARGUMENT LIST FOR SYSF4 CALLS
        .WORD   ARG1
        .WORD   ARG2
        .WORD   RESULT
LONWAV: .BLKW   2                               ;LONG WAVELENGTH
MASBUF: .BLKW   100.                            ;BUFFER FOR MASS PEAKS
MADAC1: .BLKW   2                               ;LOW VALUE FOR SECTOR MAGNET
MADAC2: .WORD                                   ;HIGH VALUE FOR SECTOR MAGNET
MARES:  .BLKW 2                                 ;SECTOR MAGNET RESOLUTION WORD
MCSBUF: .WORD                                   ;MCS UNIT DATA BUFFER POINTER
MDAC1:  .BLKW   2                               ;LOW MASS FOR QUADRUPOLE
MDAC2:  .WORD                                   ;HIGH MASS FOR QUADRUPOLE
MONTH:  .WORD                                   ;MONTH(DATE)
MRES:   .BLKW 2                                 ;QUAD. RESOLUTION WORD
NCHAN:  .BLKW   2                               ;NUMBER OF DATA CHANNELS
NEGA:   .WORD                                   ;MCS UNIT
NPTAVG: .WORD                                   ;NUMBER OF POINTS AVERAGED
```

```
NSTEP:    .WORD                            ;NUMBER OF MOTOR STEPS
XJMP:     .BLKW   2                        ;X-AXIS INCREMENT VALUE
NSCAN:    .BLKW   2                        ;NUMBER OF DATA SCANS
OFFSET:   .WORD                            ;OFFSET IN SUBROUTINE SMOOTH
OVF:      .WORD                            ;MCS UNIT
PLCHAR:   .WORD                            ;PLOT CHARACTER
PLCTRL:   .WORD                            ;PLOTTER CONTROL WORD
PLMOV:    .WORD                            ;PLOT MOVEMENT
PLLINE:   .WORD                            ;PLOT LINE TYPE
PLLIST:   .WORD   8.                       ;ARG. LIST FOR PLOTTER CALL
          .WORD   PLCTRL
          .WORD   PLCHAR
          .WORD   PLLINE
          .WORD   DISTRT
          .WORD   DICHNS
          .WORD   STCHAN
          .WORD   PLMOV
          .WORD   YMAX
POBUF:    .WORD                            ;POWER BUFFER POINTER
POVALU:   .BLKW   2                        ;POWER CHANNEL VALUE
POWMAX:   .BLKW   2                        ;POWER BUFFER MAX.
RESULT:   .BLKW   2                        ;SYSF4 RESULT WORDS
RETAKE:   .WORD                            ;RETAKE MODE FLAG
ROUTINE:.WORD                              ;CONTAINS SUB-RTN LOCATION FOR 'KEYH
OL'
RUN:      .BLKW   2                        ;RUN NUMBER WORD
RUNFLG:   .WORD                            ;RUN COMPLETED FLAG
R1DATA:   .WORD                            ;SAVE WORDS FOR 'DATAIN'
R4DATA:   .WORD
RODMA:    .WORD                            ;SAVE WORDS FOR 'DMAINT'
R1DMA:    .WORD
R2DMA:    .WORD
R3DMA:    .WORD
R1BUF:    .WORD                            ;REGISTER SAVE SUBROUTINE BUFINIT
R2BUF:    .WORD
R2DSK:    .WORD                            ;REG. SAVE FOR DISK INPUT/OUTPUT
R1HEA:    .WORD                            ;REG. SAVE SUROUTINE 'HEADER'
R2HEA:    .WORD
R1EXP:    .WORD                            ;REGISTER SAVE SUBROUTINE 'EXPAND'
R1MD:     .WORD                            ;REGISTER SAVE 'MD' SUBROUTINE
R1NOC:    .WORD                            ;REG. SAVE SUBROUTINE 'NOCYCL'
R1NOS:    .WORD                            ;REG. SAVE SUBROUTINE 'NOSCN'
R1PRI:    .WORD                            ;REG. SAVE SUBROUTINE 'PRINT'
R2PRI:    .WORD                            ;AND SUBROUTINE 'TYDATA'
R3PRI:    .WORD
R4PRI:    .WORD
R5PRI:    .WORD
R1PRH:    .WORD                            ;REG. SAVE FOR SUBROUTINE PRTITL
R1PRS:    .WORD                            ;REG. SAVE FOR SUBROUTINE PRSTAT
R1SAV:    .WORD                            ;REG. SAVE GENERAL USE
ROSMO:    .WORD                            ;REG. SAVE SUBROUTINE 'SMOOTH'
R1SMO:    .WORD
R2SMO:    .WORD
R3SMO:    .WORD
R4SMO:    .WORD
```

```
R5SMO:  .WORD
R1UPD:  .WORD                                   ;REG. SAVE SUBROUTINE 'UPDAT'
R1DIS:  .WORD                                   ;REGISTER SAVE SUBROUTINE DISCAL
R2DIS:  .WORD
R3DIS:  .WORD
R4DIS:  .WORD
R5DIS:  .WORD
R1TRA:  .WORD                                   ;REGISTER SAVE 'TRAP' SUBROUTINE
R1TYH:  .WORD                                   ;REG. SAVE FOR SUBROUTINE TYHEDR
R1TYS:  .WORD                                   ;REG. SAVE FOR SUBROUTINE TYSTAT
SBBUF:  .WORD                                   ;S+B BUFFER POINTER
SCCODE: .WORD                                   ;DEVICE SCAN CODE
SCFUNC: .WORD                                   ;SCAN FUNCTION WORD
SCMODE: .WORD                                   ;SCAN INTERRUPT MODE
SCROLL: .WORD                                   ;WINDOW SCROLL SUBROUTINE
SCTYPE: .WORD                                   ;SCAN METHOD
SETWAV: .BLKW   2                               ;THE WAVELENGTH SET PRESENTLY
SHIFT:  .WORD                                   ;ASHC FAC. IN SCALING ROUTINE
SHOWAV: .BLKW   2                               ;THE SHORT WACVELENGTH OF SCAN
SLRFLG: .WORD                                   ;L-R SCROLL FLAG
SMOPTR: .WORD                                   ;SMOOTH-POINTER
STCHAN: .WORD                                   ;STARTING-CHANNEL OF DISPLAY
SWPBCD: .WORD                                   ;MCS UNIT
SWPBIN: .WORD                                   ;MCS UNIT
TIMES:  .WORD                                   ;# OF TOMES TO 'SMOOTH'
TRGBCD: .WORD                                   ;MCS UNIT
TRGBIN: .WORD                                   ;MCS UNIT
UPDATE: .BLKW   2                               ;DISPLAY UPDATE COUNTER
WINBRT: .WORD                                   ;DISPLAY WINDOW BRIGHTNESS
WINFLG: .WORD                                   ;FLAG TO DISPLAY WINDOW
WINSIZ: .WORD                                   ;DISPLAY WINDOW SIZE
WAVINC: .BLKW   2                               ;WAVELENGTH STEP SIZE
YEAR:   .WORD                                   ;YEAR(DATE)
YFUNC:  .WORD                                   ;FUNCTION WORD FOR YAXIS SCALING
YMAX:   .BLKW   2                               ;Y-AXIS VALUE FOR 'FULL SCALE'
YMIN:   .BLKW   2                               ;Y-AXIS SUBTRACTION
YSCALE: .WORD                                   ;Y-AXIS EXTRA SCALE FACTOR
ZERFLG: .WORD                                   ;ZERO SHIFT
;
LOCKEY: .ASCII  /^X^X/                          ;KEY TO UNLOCK PROGRAM VARIABLES
;
;DATA STRING FOR 'DATES' SUBROUTINE
;
R50STR: .RAD50  /JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC/
;
;
;
;       INPUT/OUTPUT  AND  DATA  BUFFER  DEFINITIONS
IOBUF:  .BLKB   100.                            ;INPUT/OUTPUT BUFFER
DATBUF: .BLKW   10000.                          ;DATA/DISPLAY BUFFER
;
;
;
        .END MCHSUB
```

```
        .TITLE  A2DAMP
;SUBROUTINE TO SET THE GAIN OF THE 3553-Z1A ANALOG-TO-DIGITAL COVERTER'S
;AMPLIFIER. WHEN QUERIED ONE REPLIES 'LOW' FOR A GAIN OF '1','MEDIUM' FOR
;A GAIN OF '10',AND 'HIGH' FOR A GAIN OF '100'. GLOBAL IOBUF
;MUST BE DEFINED BY THE CALLING PROGRAM. THIS ROUTINE IS USED BY A
;'CALL A2DAMP' INSTRUCTION.
;
;
;       SYSTEM MACRO
        .MCALL  .PRINT
;
;       GLOLIB  GLOBAL LIBRARY ROUTINE
        .GLOBL  READ
;
;       GLOBAL DEFINED IN MAIN PROGRAM
        .GLOBL  IOBUF
;
;       GLOBAL DEFINED BY THIS MODULE
        .GLOBL  A2DAMP,A2DGNS
;
;
;
;
A2DCON=167100                           ;A2D CONVERTER (BIN 18)
A2DAMP: MOVB    #24.,A2DCON             ;DISABLE THE LAM REQUEST
        MOV     IOBUF+2,IOBUF           ;GET SECOND COMMAND IF ANY
        CMPB    IOBUF,#15               ;IS IT A CARRIAGE RETURN?
        BEQ     10$                     ;ASK FOR SECOND PART OF COMMAND:IF Y
ES
        BR      15$                     ;GO DECODE COMMAND
10$:    .PRINT  #M1                     ;WHAT SETTING IS WANTED?
        CALL    READ                    ;READ THE REPLY
15$:    MOV     IOBUF,A2DGNS            ;SAVE THE DESIRED GAIN SETTING
        CMP     #'LO,IOBUF              ;A GAIN OF '1'?
        BEQ     20$                     ;YES,SET GAIN TO 1.
        CMP     #'ME,IOBUF              ;A GAIN OF '10'?
        BEQ     30$                     ;IF YES,SET GAIN TO 10.
        CMP     #'HI,IOBUF              ;A GAIN OF '100'?
        BEQ     40$                     ;IF YES,SET GAIN TO 100.
        .PRINT  #M2                     ;INVALID RESPONSE--TELL USER
        .PRINT  #M3
        BR      10$                     ;GO BACK--TRY AGAIN
20$:    MOVB    #25.,<A2DCON+2>         ;SELECT GAIN OF 1.
        BR      50$                     ;GO WAIT
30$:    MOVB    #25.,<A2DCON+4>         ;SELECT GAIN OF 10.
        BR      50$                     ;GO WAIT
40$:    MOVB    #25.,<A2DCON+6>         ;SELECT GAIN OF 100.
50$:    MOV     #20000.,DELAY           ;LOAD DELAY WORD
60$:    DEC     DELAY                   ;DELAY LOOP
        BNE     60$                     ;WAIT FOR CONVERTION TO FINISH
        MOVB    #0,A2DCON               ;CEAR THE LAM
        MOVB    #26.,A2DCON             ;ENABLE THE LAM REQUEST
        RETURN                          ;RETURN TO CALLER
;
;
```

```
A2DGNS: .WORD                                   ;WORD FOR GAIN SETTING
DELAY:  .WORD                                   ;WORD FOR DELAY LOOP
;
;
M1:     .ASCII<15><12>/WHAT GAIN SETTING DO YOU WANT? /
        .BYTE   200
M2:     .ASCIZ<15><12>/ILLEGAL COMMAND--PLEASE USE 'LO(W),ME(D),OR HI(GH)'/
M3:     .ASCIZ<15><12>/      LO = X1      ME = X10      HI = X100/
;
;
        .EVEN
        .END
```

```
        .TITLE  ASCBIN
;SUBROUTINE TO CONVERT ASCII CHARACTERS IN I/O BUFFER(IOBUF) TO
;A BINARY NUMBER IN R1 (LOW WORD) AND R2 (HIGH WORD).
;IOBUF IS NOT CHANGED.
        .GLOBL  IOBUF,ASCBIN,JMUL
        .GLOBL  REGSAV,REGRES
;
ASCBIN: MOV     R5,R5ASC                ;SAVE REGISTER
        CLR     R2                      ;CLR CHARACTER REGISTER
        MOV     #LIST,R5                ;TO USE SYSF4 JMUL
        MOV     #IOBUF,R1               ;STARTING ADDRESS IN R1
        MOV     #1,NEGFLG               ;NEGATIVE-VALUE FLAG UP
        CMPB    #'-,(R1)                ;IS FIRST CARACTER A MINUS?
        BNE     10$                     ;BRANCH IF NO
        CLR     NEGFLG                  ;NEGATIVE-VALUE FLAG DOWN
        MOVB    #'0,(R1)                ;YES,REPLACE IT WITH ZERO
        BR      20$
10$:    CMPB    #'+,(R1)                ;IS FIRST CHARACTER A PLUS?
        BNE     20$                     ;BRANCH IF NO
        MOVB    #'0,(R1)                ;YES,REPLACE IT WITH A ZERO
20$:    MOV     #0,RESULT               ;CLEAR RESULT
        MOV     #0,RESULT+2
        MOV     #10.,ARG2               ;PLACE TEN IN ARGUMENT TWO
        MOV     #0,ARG2+2
30$:    MOV     RESULT,ARG1             ;ARG1 TO BE MULTIPLIED BY TEN
        MOV     RESULT+2,ARG1+2
        MOVB    (R1)+,R2                ;MOVE CHARACTER TO R2
        CMPB    #15,R2                  ;IS IT A CRRIAGE RETURN?
        BEQ     60$                     ;BRANCH IF YES
        CMPB    #60,R2                  ;CHECK FOR LETTERS,ETC
        BGT     40$                     ;IF ANY FOUND,SET CARRY BIT
        CMPB    #71,R2
        BGE     50$
40$:    SEC                             ;ERROR: SET CARRY BIT
        BR      ASCOUT                  ;GO TO CALLER
50$:    SUB     #60,R2                  ;SUBTRACT ASCII BIAS
        CALL    REGSAV                  ;SAVE REGISTERS R1-R5
        CALL    JMUL                    ;SYSF4 MULTIPLICATION
        CALL    REGRES                  ;RESTORE REGISTERS R1-R5
        CMP     #-2,R0                  ;CHECK ERROR CODE
        BEQ     40$                     ;BRANCH IF YES  (OVERFLOW!)
        ADD     R2,RESULT               ;ADD TO RESULT
        ADC     RESULT+2
        BR      30$                     ;GO BACK FOR MORE
60$:    CMP     #0,NEGFLG               ;WAS IT NEGATIVE?
        BNE     130$                    ;NO SKIP NEGATION
        COM     RESULT
        COM     RESULT+2
        ADD     #1,RESULT
        ADC     RESULT+2
130$:   MOV     RESULT,R1               ;ANSWER IN R1
        MOV     RESULT+2,R2             ;AND R2
        CLC                             ;CLEAR CARRY:   NO ERRORS
ASCOUT: MOV     R5ASC,R5                ;UNSAVE REGISTER
        RETURN                          ;RETURN TO CALLER
```

```
;
;
LIST:    .WORD   3                       ;JMUL SYS. GLOBAL ARGUMENT LIST
         .WORD   ARG1
         .WORD   ARG2
         .WORD   RESULT
ARG1:    .BLKW   2
ARG2:    .BLKW   2
RESULT:  .BLKW   2
NEGFLG:  .WORD
R5ASC:   .WORD
;
;
         .END
```

```
        .TITLE  BINASC SUBROUTINE
;SUBROUTINE 'BINASC' IS A ROUTINE TO CONVERT A SIGNED BINARY NUMBER
;IN R1 TO AN ASCII BYTE STRING WITH STARTING LOCATION 'IOBUF'. THE
;ASCII CODED STRING CONSIST OF SIX CHARACTERS. A MINUS SIGN FOR
;NEGATIVES WITH LEADING ZEROES SUPPRESSED;THE MINUS SIGN IS PLACED
;JUST BEFORE THE FIRST NONZERO DIGIT. USE WITH 'CALL BINASC'. THE
;BUFFER 'IOBUF' MUST BE DEFINED AS A GLOBAL IN THE CALLING PROGRAM.
        .GLOBL  IOBUF,BINASC
BINASC: MOV     R1,R1BIN        ;SAVE REGISTERS
        MOV     R2,R2BIN
        MOV     R3,R3BIN
        MOV     R4,R4BIN
        MOV     #IOBUF,R4       ;BUFFER STARTING LOCATION
        TST     R1              ;IS IT NEGATIVE?
        BPL     10$             ;NO
        NEG     R1              ;YES,TAKE ABS. VALUE
        MOVB    #'-,(R4)+       ;PLACE MINUS SIGN IN BUFFER
        BR      15$             ;GO TO CONVERTION SECTION
10$:    MOVB    #' ,(R4)+       ;PLACE A SPACE IN BUFFER
15$:    MOV     #BIN1,R2        ;POINTER TO DIVISOR TABLE
        MOV     #5,R3           ;FIVE DIVISORS
20$:    CLR     R0              ;CLEAR HIGH WORD
        DIV     (R2)+,R0        ;FIS DIVIDE
        ADD     #60,R0          ;ADD ASCII BIAS
        MOVB    R0,(R4)+        ;PLACE RESULT IN BUFFER
        SOB     R3,20$          ;DO IT FIVE TIMES
;NOW REPLACE LEADING ZEROES WITH SPACES AND MOVE THE 'SIGN' ACCORDINGLY.
        MOV     #IOBUF,R4       ;BUFFER STARTING ADDRESS
        MOVB    IOBUF,R1        ;PLACE SIGN IN R1
        CLR     R3              ;CLEAR LOOP COUNTER
30$:    INCB    R4              ;GET NEXT BYTE LOCATION
        INCB    R3              ;INCREMENT LOOP COUNTER
        CMPB    #5,R3           ;IS IT THE FIFTH TIME?
        BEQ     40$             ;IF YES,EXIT
        CMPB    #'0,(R4)        ;IS THE CHARACTER A ZERO?
        BNE     40$             ;NO,EXIT ROUTINE
        MOVB    R1,(R4)         ;YES:REPLACE IT WITH SIGN
        MOVB    #' ,-1(R4)      ;REPLACE FIRST SIGN WITH SPACE
        BR      30$             ;LOOK FOR MORE ZEROES
40$:    MOV     R4BIN,R4        ;UNSAVE REGISTERS
        MOV     R3BIN,R3
        MOV     R2BIN,R2
        MOV     R1BIN,R1
        RETURN                  ;RETURN TO CALLING PROGRAM
BIN1:   .WORD   10000.,1000.,100.,10.,1
R1BIN:  .WORD
R2BIN:  .WORD
R3BIN:  .WORD
R4BIN:  .WORD
        .END
```

```
        .TITLE   BCDBIN
;BCDBIN SUBROUTINE CONVERTS BCD NUMBER
;IN R1 TO A BINARY NUMBER IN R1
        .GLOBL   BCDBIN
BCDBIN: MOV      R2,R2BCD
        MOV      R3,R3BCD
        MOV      R4,R4BCD
        MOV      R5,R5BCD
        MOV R1,R5         ;COPY INTO R5
        MOV #ASHIFT,R2    ;INIT.
        MOV #AMUL,R3
        CLR BIN
        MOV #4,R4         ;FOUR OF THEM
BCD1:   MOV R5,R1         ;GET THE NUMBER
        ASH (R2)+,R1      ;MASK THE BITS
        BIC #177760,R1
        MUL (R3)+,R1      ;MULTIPLY BY POWER OF TEN
        ADD R1,BIN
        SOB R4,BCD1       ;FOUR TIMES
        MOV BIN,R1        ;PLACE BINARY IN R1
        MOV      R2BCD,R2
        MOV      R3BCD,R3
        MOV      R4BCD,R4
        MOV      R5BCD,R5
        RTS      PC               ;RETURN
ASHIFT: .WORD -12.,-8.,-4,0
AMUL:   .WORD 1000.,100.,10.,1
BIN:    .WORD
R2BCD:  .WORD
R3BCD:  .WORD
R4BCD:  .WORD
R5BCD:  .WORD
        .END
```

```
        .TITLE  DECOUT SUBROUTINE
;DECOUT SUBROUTINE OUTPUTS THE BINARY NUMBER CONTAINED IN R1
;IN DECIMAL FORMAT TO THE CONSOLE DEVICE. SIX CHARACTERS ARE
;OUTPUT; A MINUS SIGN IF THE NUMBER IS NEGATIVE AND WITH ANY
;LEADING ZEROES REPLACED BY SPACES. THE MINUS SIGN APPEARS
;NEXT TO THE FIRST NONZERO DIGIT.
        .MCALL  .TTYOUT
        .GLOBL  DECOUT,BINASC,IOBUF
DECOUT: MOV     R1,R1DEC        ;SAVE REGISTERS
        MOV     R2,R2DEC
        CALL    BINASC          ;CONVERT R1 TO ASCII STRING
        MOV     #6,R1           ;SIX CHARACTERS OUTPUT
        MOV     #IOBUF,R2       ;BUFFER STARTING ADDRESS
10$:    MOVB    (R2)+,R0        ;OUTPUT CHARACTERS ONE BY ONE
        .TTYOUT
        SOB     R1,10$
        MOV     R2DEC,R2        ;UNSAVE REGISTERS
        MOV     R1DEC,R1
        RETURN                  ;RETURN TO CALLING PROGRAM
R1DEC:  .WORD
R2DEC:  .WORD
        .END
```

```
        .TITLE  DCOUT2 SUBROUTINE
;DCOUT2 SUBROUTINE OUTPUTS THE BINARY NUMBER CONTAINED IN WORDS
;LOWORD AND HIWORD IN DECIMAL FORMAT TO THE CONSOLE DEVICE. TEN CHARACTERS
;ARE OUTPUT; A MINUS SIGN IF THE NUMBER IS NEGATIVE AND WITH ANY
;LEADING ZEROES REPLACED BY SPACES. THE MINUS SIGN APPEARS
;NEXT TO THE FIRST NONZERO DIGIT.
        .MCALL  .TTYOUT
        .GLOBL  DCOUT2,DPBASC,IOBUF
DCOUT2: MOV     R1,R1DEC        ;SAVE REGISTERS
        MOV     R2,R2DEC
        CALL    DPBASC          ;CONVERT 'WORDS' TO ASCII STRING
        MOV     #11.,R1         ;ELEVEN CHARACTERS OUTPUT
        MOV     #IOBUF,R2       ;BUFFER STARTING ADDRESS
10$:    MOVB    (R2)+,R0        ;OUTPUT CHARACTERS ONE BY ONE
        .TTYOUT
        SOB     R1,10$
        MOV     R2DEC,R2        ;UNSAVE REGISTERS
        MOV     R1DEC,R1
        RETURN                  ;RETURN TO CALLING PROGRAM
R1DEC:  .WORD
R2DEC:  .WORD
        .END
```

```
        .TITLE  DISKIO INPUT-OUTPUT TO THE DISK
;
;
;THIS MODULE CONTAINS SUBROUTINES TO WRITE AND READ WITH THE DISK.
;'WRDISK' WRITES N WORDS TO THE DISK. N/6 IS PLACED INTO THE FIRST
;LOCATION OF 'DATBUF' THE BUFFER WHICH HAS THE N WORDS TO BE SENT.
;THE FIRST 50 WORDS ARE RESERVED FOR 'VARIABLES' AND THE NEXT
;450 WORDS ARE FOR A HEADER IF DESIRED.
;'REDISK' READS N WORDS FROM THE SPECIFIED FILE (N IS THE FIRST
;WORD OF THE DATA FILE) AND PLACES IT INTO 'DATBUF'.
;
;
;       SYSTEM  MACROES
;
        .MCALL  .TTYIN,.TTYOUT,.PRINT,.CLOSE,.CSIGEN
        .MCALL  .WRITW,.READW
;
;       MAIN PROGRAM DEFINED GLOBALS
;
        .GLOBL  NCHAN,DATBUF,IOBUF
;
;
;       GLOLIB GLOBAL SUBROUTINE
;
        .GLOBL  READ
;
;
;       GLOBAL ENTRY POINTS OF THIS MODULE
;
        .GLOBL  WRDISK,REDISK
;
;
;'WRDISK' WRITES DATBUF TO THE DISK
WRDISK: MOV     R1,R1SAV                ;SAVE REGISTER 1
        MOV     R2,R2SAV                ;SAVE REGISTER TWO
5$:     .PRINT  #M8                     ;NEW HEADER?
        CALL    READ                    ;YES OR NO?
        CMPB    #'Y,IOBUF               ;YES?
        BEQ     10$                     ;BRANCH IF YES
        CMPB    #'N,IOBUF               ;NO?
        BEQ     50$                     ;BRANCH IF NO
        BR      5$                      ;MUST ANSWER YES OR NO
10$:    MOV     #DATBUF,R2              ;DATA BUFFER STARTING LOCATION
        MOV     NCHAN,(R2)              ;STORE NUMBER OF CHANNELS
        ADD     #100.,R2                ;GO TO HEADER SECTION
        MOV     #450.,R1                ;MAX. HEADER LENGTH
15$:    CLR     (R2)+                   ;CLEAR HEADER SECTION
        SOB     R1,15$
        MOV     #DATBUF,R2              ;STARTING LOCATION IN R2
        ADD     #100.,R2                ;START OF HEADER SECTION
        .PRINT  #M6                     ;ENTER HEADER MESSAGE
        MOV     #'>,R0                  ;PRINT PROMPT CHARACTER
        .TTYOUT
        MOV     #448.,R1                ;NUMBER OF CHARATERS PERMITTED
20$:    .TTYIN                          ;READ IN THE HEADER
```

```
        CMPB    #'^,R0                      ;CHECK FOR END OF HEADER
        BEQ     40$                         ;BRANCH IF AT END
        DEC     R1                          ;COUNT DOWN SPACE
        BEQ     30$                         ;BRANCH IF NO SPACE LEFT
        MOVB    R0,(R2)+                    ;PLACE CHARACTER INTO BUFFER
        BR      20$                         ;GO GET MORE
30$:    .TTYIN                              ;EAT THE LEFT OVERS!
        CMPB    #12,R0                      ;THRU EATING?
        BNE     30$                         ;GET MORE TO EAT IF NOT
        .PRINT  #M3                         ;TELL USER TRUNCATION HAS OCCURRED
        BR      50$                         ;CONTINUE
40$:    .TTYIN                              ;EAT CR-LF
        CMP     #12,R0                      ;HAVE LF IN MOUTH?
        BNE     40$                         ;GO GET IT!!
50$:    MOV     #200,(R2)                   ;HEADER TERMINATOR SYMBOL
        .PRINT  #M7                         ;ASK FOR FILE SPECIFICATIONS
        MOV     SP,SPSAV                    ;SAVE THE STACK POINTER
        .CSIGEN #DEVSPC,#DEFEXT,#0          ;COMMAND STRING GENERATOR
        MOV     SPSAV,SP                    ;RESTORE SP
        MOV     NCHAN,R1                    ;CALCULATE NUMBER OF WORDS
        MUL     #6,R1                       ;TO BE TRANSFERED
        ADD     #502.,R1
        .WRITW  #WAREA,#0,#DATBUF,R1,#0     ;WITE THRU CHANNEL ZERO
        BCS     60$                         ;BRANCH IF HARDWARE ERROR
        CMP     R1,R0                       ;CORRECT NUMBER OF WORDS WRITTEN?
        BEQ     150$                        ;BRANCH IF YES
        .PRINT  #M5                         ;PRINT ERROR MESSAGE
60$:    CMPB    #0,@#52                     ;CHECK  FOR THE ERROR
        BNE     130$
        .PRINT  #M0                         ;READ MESSAGES
        BR      150$
130$:   CMPB    #1,@#52
        BNE     140$
        .PRINT  #M1
        BR      150$
140$:   CMPB    #2,@#52
        BNE     150$
        .PRINT  #M2
150$:   .CLOSE  #0                          ;CLOSE CHANNEL
        .PRINT  #M4                         ;SUCCESS
        MOV     R1SAV,R1
        MOV     R2SAV,R2                    ;RESTORE R2
        RETURN                              ;RETURN TO CALLER
;
;
;THIS SUBROUTINE READS THE DATA FROM THE DISK
;
REDISK: MOV     R1,R1SAV                    ;SAVE R1
        .PRINT  #M7                         ;WHICH FILE?
        MOV     SP,SPSAV                    ;SAVE THE STACK POINTER
        .CSIGEN #DEVSPC,#DEFEXT,#0          ;GET COMMAND STRING GEN.
        MOV     SPSAV,SP                    ;RESTORE SP
        .READW  #WAREA,#3,#DATBUF,#5,#0
        BCS     20$                         ;HARDWARE ERROR?
        CMP     R0,#5                       ;CHECK NUMBER OF WORDS READ
```

```
        BEQ     10$
        .PRINT  #M5
        BR      20$                     ;WHICH ERROR?
10$:    MOV     DATBUF,NCHAN            ;FIRST WORD= # OF DATA CHANNELS
        MOV     NCHAN,R1
        MUL     #6,R1                   ;HOW MANY WORDS IN ALL?
        ADD     #502.,R1
        .READW  #WAREA,#3,#DATBUF,R1,#0
        BCS     20$
        CMP     R0,R1
        BEQ     50$
        .PRINT  #M5
20$:    CMPB    #0,@#52                 ;CHECK THE ERROR CODES
        BNE     30$
        .PRINT  #M0
        BR      50$
30$:    CMPB    #1,@#52
        BNE     40$
        .PRINT  #M1
        BR      50$
40$:    CMPB    #2,@#52
        BNE     50$
        .PRINT  #M2
50$:    .CLOSE  #3                      ;CLOSE CHANNE 3
        MOV     R1SAV,R1
        RETURN                          ;RETURN TO CALLER
;
R1SAV:  .WORD   0
R2SAV:  .WORD
SPSAV:  .WORD   0
;
;
DEFEXT: .RAD50  /DAT/
        .RAD50  /DAT/
        .WORD   0
        .WORD   0
WAREA:  .BLKW   5
DEVSPC: .WORD   0
M0:     .ASCIZ  /ATTEMPTED TO WRITE(READ) PAST END-OF-FILE/
M1:     .ASCIZ  /HARDWARE ERROR/
M2:     .ASCIZ  /CHANNEL WAS NOT OPEN/
M3:     .ASCIZ  /MESSAGE TOO LONG--TRUNCATED:NOT FATAL CONTINUE/
M4:     .ASCIZ  /DATA TRANSFER SUCCESSFUL/
M5:     .ASCIZ  /IMPROPER NUMBER OF WORDS WRITTEN(READ)/
M6:     .ASCIZ  /TYPE IN HEADER MESSAGE/
M7:     .ASCIZ  /GIVE FILE SPECIFICATIONS/
M8:     .ASCII<15><12>/DO YOU WISH TO ENTER A NEW HEADER MESSAGE? /
        .BYTE   200
        .EVEN
;
;
        .END
```

```
        .TITLE  DPBASC SUBROUTINE
;DOUBLE PRECISION INTEGER ASCII DECIMAL OUTPUT PROGRAM-WILL PLACE INTO A
;SPECIFIED BYTE STRING 10 ASCII NUMERALS AND A SIGN IF NEGATIVE. LEADING
;ZEROES ARE SUPRESSED AND THE SIGN IS PLACED IMMEDIATELY BEFORE THE
;FIRST NON-ZERO DIGIT OF THE NUMBER.
;TO ENTER:
;       1. MOVE HIGH ORDER PART OF WORD TO BE OUTPUT TO HIWORD
;       2. MOVE LOW ORDER PART OF WORD TO BE OUTPUT TO LOWORD
;       3.CALLING PROGRAM MUST DEFINE 'IOBUF' AS GLOBAL
;       4.CALLING PROGRAM MUST DEFINE LOWORD AND HIWORD AS GLOBALS
;       5.USE ROUTINE WITH 'CALL DPBASC'
;ROUTINE WRITTEN BY SPARKS, MODIFIED DY BUSTAMENTE.
;
        .GLOBL  IOBUF,DPBASC,LOWORD,HIWORD
DPBASC: MOV     R3,R3DPB                ;SAVE REGISTERS
        MOV     R5,R5DPB
        MOV     #IOBUF,R3               ;BUFFER ADDRESS IN R3
        CLR     NEGFLG                  ;CLEAR NEGATIVE # FLAG
        CLR     LZSFLG                  ;CLEAR LEADING ZERO SUPPRESS FLAG
        MOV     #-1,CNTR                ;INITIALIZE SUBTRACT LOOP COUNTER
        TST     HIWORD                  ;IS INTEGER NEGATIVE
        BMI     NEGSET                  ;YES, THEN CONVERT IT
POSOUT: MOV     #DIG9,R5                ;LOAD BEGINNING ADDRESS OF 10^N LIST
MORYET: INC     CNTR                    ;COUNT # SUBTRACTION LOOPS
        SUB     2(R5),LOWORD            ;THREE STEP DOUBLE PREC. SUBTRACT
        SBC     HIWORD
        SUB     (R5),HIWORD
        BPL     MORYET                  ;SUBTRACT AGAIN IF STILL POSITIVE
        ADD     2(R5),LOWORD            ;KILL LAST SUBTRACTION IF NEGATIVE
        ADC     HIWORD
        ADD     (R5),HIWORD
        ADD     #4,R5                   ;READY FOR 10^N-1
        JSR     PC,ASCOUT               ;OUTPUT THE CHARACTER
        MOV     #-1,CNTR                ;RESET LOOP COUNTER
        CMP     R5,#<DIG1+4>            ;? LAST DIGIT YET
        BLT     MORYET                  ;NO, DO ANOTHER
        MOV     LOWORD,CNTR             ;YES, SEND IT OUT
        INC     R5                      ;TRACER FOR LAST DIGIT
        JSR     PC,ASCOUT
        MOV     R3DPB,R3                ;UNSAVE REGISTERS
        MOV     R5DPB,R5
        RTS     PC                      ;RETURN TO CALLING PROGRAM
NEGSET: BIS     #1,NEGFLG               ;SET FLAG FOR MINUS SIGN OUTPUT
        SUB     #1,LOWORD               ;4 STEP CONVERSION TO POSITIVE INTEG
ER
        SBC     HIWORD
        COM     HIWORD
        COM     LOWORD
        JMP     POSOUT                  ;DECODE TO ASCII
ASCOUT: CMP     CNTR,#0                 ;? GOT A ZERO
        BEQ     LZS                     ;YES, SEE IF NEEDS SUPPRESSED
        BIS     #1,LZSFLG               ;NO, PREVENT FURTHER SUPPRESSION
LZSDN:  ADD     #60,CNTR                ;ADD ASCII BIAS
        BIT     #2,NEGFLG               ;IS SIGN OR SPACE ALREADY PRINTED?
        BGT     CHROUT                  ;YES, JUST SEND CHARACTER
```

```
        BIT     #1,NEGFLG       ;NO, DO WE NEED A MINUS SIGN?
        BGT     MINOUT          ;YES, DO IT
        MOVB    #40,(R3)+       ;NO, SEND A SPACE
        MOVB    CNTR,(R3)+      ;THEN SEND CHARACTER
        BIS     #2,NEGFLG       ;NO MORE SPACES
        RTS     PC              ;BACK TO CALLER
MINOUT: MOVB    #55,(R3)+       ;SEND MINUS SIGN
        MOVB    CNTR,(R3)+      ;SEND CHARACTER
        BIC     #1,NEGFLG       ;MINUS SIGN DONE
        BIS     #2,NEGFLG       ;NO SPACES WANTED
        RTS     PC
LZS:    TST     LZSFLG          ;LEADING ZEROES OVER WITH?
        BGT     LZSDN           ;YES, DON'T SUPPRESS
        CMP     R5,#<DIG1+4>    ;LAST DIGIT YET?
        BGT     LZSDN           ;YES, THEN PRINT EVEN IF ZERO
        MOVB    #40,(R3)+       ;NO, SEND A SPACE
        RTS     PC
CHROUT: MOVB    CNTR,(R3)+      ;SEND CHARACTER
        RTS     PC
CNTR:   .WORD
NEGFLG: .WORD
HIWORD: .WORD
LOWORD: .WORD
LZSFLG: .WORD
DIG9:   .WORD   35632,145000
DIG8:   .WORD   2765,160400
DIG7:   .WORD   230,113200
DIG6:   .WORD   17,41100
DIG5:   .WORD   1,103240
DIG4:   .WORD   0,23420
DIG3:   .WORD   0,1750
DIG2:   .WORD   0,144
DIG1:   .WORD   0,12
R3DPB:  .WORD
R5DPB:  .WORD
        .END
```

```
        .TITLE  CRLF
;CRLF SUBROUTINE INITIALIZES CONSOLE DEVICE CARRIAGE
        .MCALL  .TTYOUT
CRLF::  MOV #15,R0      ;PRINT CR
        .TTYOUT
        MOV #12,R0      ;PRINT LF
        .TTYOUT
        RTS     PC              ;RETURN
        .END
```

```
        .TITLE  PRINTER SUBROUTINES
;
LPRCSR=176500
LPRBUF=176502
LPXCSR=176504
LPXBUF=176506
;
;
;
;SUBROUTINE LPOUT
;
;LPOUT SUBROUTINE PRINTS CHARACTERS OUT ON THE LINE PRINTER
;CHARACTER TO BE PRINTED IS IN RO. USE WITH 'CALL LPOUT'.
        .GLOBL  LPOUT
;
LPOUT:  BIT     #200,LPXCSR         ;READY TO PRINT?
        BEQ     LPOUT               ;NO
        BIT     #200,LPRCSR         ;CHARACTER RECEIVED?
        BEQ     20$                 ;BR IF NO
5$:     MOVB    LPRBUF,LPIN         ;MOVE IN CHARACTER
        BIC     #200,LPIN           ;CLEAR PARITY BIT
        CMPB    #21,LPIN            ;TRANSMIT?
        BEQ     20$                 ;BR IF YES
        CMPB    #23,LPIN            ;STOP TRANSMISSION?
        BNE     20$
10$:    BIT     #200,LPRCSR         ;WAIT FOR 'RESUME'
        BEQ     10$
        BR      5$
20$:    NOP                         ;CONTINUE
        MOVB    RO,LPXBUF           ;YES,PRINT IT
        RETURN                      ;RETURN TO CALLING PROGRAM
;
;
;
;SUBROUTINE LPRINT
;
;SUBROUTINE LPRINT EMULATES THE MONITOR .PRINT REQUEST BUT SENDS
;OUTPUT TO THE PRINTER. IT IS ACCESSED BY PLACING THE BIGINNING
;ADDRESS OF THE BYTE BY BYTE ASCII STRING TO BE OUTPUT IN RO
;FOLLOWED BY THE INSTRUCTION 'CALL LPRINT'. CARRIAGE RETURN AND
;LINE FEED HANDLING ARE THE SAME AS THE MONITOR .PRINT REQUEST
;SUBROUTINE BY R.K.SPARKS MODIFIED BY S.W.BUSTAMENTE
        .GLOBL  LPRINT
;
LPRINT: CMPB    (RO),#200           ;?TERMINATOR FOR NO CRLF
        BEQ     2$                  ;YES,THEN ARE FINISHED
        CMPB    (RO),#0             ;? TERMINATOR FOR CRLF
        BEQ     OUT                 ;YES,THEN OUTPUT LF
1$:     BIT     #200,LPXCSR         ;READY FOR NEW CHARACTER?
        BEQ     1$                  ;NO,THEN LOOP
        BIT     #200,LPRCSR         ;CHARACTER RECEIVED?
        BEQ     20$                 ;BR IF NO
5$:     MOVB    LPRBUF,LPIN         ;MOVE IN CHARACTER
        BIC     #200,LPIN           ;CLEAR PARITY BIT
        CMPB    #21,LPIN            ;TRANSMIT?
```

```
        BEQ     20$                     ;BR IF YES
        CMPB    #23,LPIN                ;STOP TRANSMISSION?
        BNE     20$
10$:    BIT     #200,LPRCSR             ;WAIT FOR 'RESUME'
        BEQ     10$
        BR      5$
20$:    NOP                             ;CONTINUE
        MOVB    (R0)+,LPXBUF            ;SEND CHARACTER
        BR      LPRINT                  ;GO FOR ANOTHER
2$:     RETURN                          ;RETURN WITH NO CRLF
OUT:    BIT     #200,LPXCSR             ;READY?
        BEQ     OUT                     ;NO,LOOP
        BIT     #200,LPRCSR             ;CHARACTER RECEIVED?
        BEQ     20$                     ;BR IF NO
5$:     MOVB    LPRBUF,LPIN             ;MOVE IN CHARACTER
        BIC     #200,LPIN               ;CLEAR PARITY BIT
        CMPB    #21,LPIN                ;TRANSMIT?
        BEQ     20$                     ;BR IF YES
        CMPB    #23,LPIN                ;STOP TRANSMISSION?
        BNE     20$
10$:    BIT     #200,LPRCSR             ;WAIT FOR 'RESUME'
        BEQ     10$
        BR      5$
20$:    NOP                             ;CONTINUE
        MOVB    #12,LPXBUF              ;SEND LF,GET AUTO CR
        RETURN                          ;RETURN TO CALLER
;
;
;
;SUBROUTINE LPCRLF
;
;
;THIS SUBROUTINE CAUSES THE LINE PRINTER TO EXECUTE A
;LINEFEED-CARRIAGE RETURN SEQUENCE WHEN CALLED. USE
;WITH 'CALL LPCRLF'.
        .GLOBL  LPCRLF
;
LPCRLF: BIT     #200,LPXCSR             ;READY?
        BEQ     LPCRLF                  ;NO,LOOP
        BIT     #200,LPRCSR             ;CHARACTER RECEIVED?
        BEQ     20$                     ;BR IF NO
5$:     MOVB    LPRBUF,LPIN             ;MOVE IN CHARACTER
        BIC     #200,LPIN               ;CLEAR PARITY BIT
        CMPB    #21,LPIN                ;TRANSMIT?
        BEQ     20$                     ;BR IF YES
        CMPB    #23,LPIN                ;STOP TRANSMISSION?
        BNE     20$
10$:    BIT     #200,LPRCSR             ;WAIT FOR 'RESUME'
        BEQ     10$
        BR      5$
20$:    NOP                             ;CONTINUE
        MOVB    #12,LPXBUF              ;SEND LF,GET AUTO CR
        RETURN                          ;RETURN TO CALLER
;
;
```

```
;
;SUBROUTINE LPDECO
;
;LPDECO SUBROUTINE OUTPUTS THE BINARY NUMBER CONTAINED IN R1
;IN DECIMAL FORMAT TO THE LINE PRINTER. SIX CHARACTERS ARE
;OUTPUT; A MINUS SIGN IF THE NUMBER IS NEGATIVE AND WITH ANY
;LEADING ZEROES REPLACED BY SPACES. THE MINUS SIGN APPEARS
;NEXT TO THE EIRST NONZERO DIGIT.
        .GLOBL  LPDECO,BINASC,IOBUF,LPOUT
;
LPDECO: MOV     R1,R1DEC          ;SAVE REGISTERS
        MOV     R2,R2DEC
        CALL    BINASC            ;CONVERT R1 TO ASCII STRING
        MOV     #6,R1             ;SIX CHARACTERS OUTPUT
        MOV     #IOBUF,R2         ;BUFFER STARTING ADDRESS
10$:    MOVB    (R2)+,R0          ;OUTPUT CHARACTERS ONE BY ONE
        CALL    LPOUT
        SOB     R1,10$
        MOV     R2DEC,R2          ;UNSAVE REGISTERS
        MOV     R1DEC,R1
        RETURN                    ;RETURN TO CALLING PROGRAM
;
;
;
;SUBROUTINE LPDCO2
;
;LPDCO2 SUBROUTINE OUTPUTS THE BINARY NUMBER CONTAINED IN WORDS
;LOWORD AND HIWORD IN DECIMAL FORMAT TO THE CONSOLE DEVICE. ELEVEN CHARACTER
S
;ARE OUTPUT; A MINUS SIGN IF THE NUMBER IS NEGATIVE AND WITH ANY
;LEADING ZEROES REPLACED BY SPACES. THE MINUS SIGN APPEARS
;NEXT TO THE FIRST NONZERO DIGIT.
;
        .GLOBL  LPDCO2,DPBASC,IOBUF,LPOUT
;
LPDCO2: MOV     R1,R1DEC          ;SAVE REGISTERS
        MOV     R2,R2DEC
        CALL    DPBASC            ;CONVERT 'WORDS' TO ASCII STRING
        MOV     #11.,R1           ;ELEVEN CHARACTERS OUTPUT
        MOV     #IOBUF,R2         ;BUFFER STARTING ADDRESS
10$:    MOVB    (R2)+,R0          ;OUTPUT CHARACTERS ONE BY ONE
        CALL    LPOUT
        SOB     R1,10$
        MOV     R2DEC,R2          ;UNSAVE REGISTERS
        MOV     R1DEC,R1
        RETURN                    ;RETURN TO CALLING PROGRAM
R1DEC:  .WORD
R2DEC:  .WORD
LPIN:   .WORD
        .END
```

```
        .TITLE  LASER CONTROL GLOBAL ROUTINES
;
;
;THIS 'LASER' MODULE CONTAINS GLOBAL SUBROUTINES WHICH ARE USED TO
;OPERATE THE QUANTA RAY DYE LASER AND THE OPO.
;
;       SYSTEM  MACROS
;
        .MCALL  .PRINT,.TTYIN
;
;
;
;       SYSF4   LIBRARY   GLOBALS
;
        .GLOBL  JDIV,JCMP,JMUL
;
;
;
;       GLOLIB  LIBRARY   GLOBALS
;
        .GLOBL  READ,ASCBIN,MOTOR1,REGSAV,REGRES
;
;
;
;       MAIN PROGRAM DEFINED GLOBAL LABELS
;
        .GLOBL  SCCODE,NSTEP,NCHAN,IOBUF
        .GLOBL  LIST,ARG1,ARG2,RESULT
        .GLOBL  SHOWAV,LONWAV,SETWAV,CALWAV,WAVINC
        .GLOBL  DATALO
;
;
;
;       GLOBAL ENTRY POINTS OF THIS SUBROUTINE
;
        .GLOBL  DYESET,DYECAL,REWAVE,DYELAS
;
;
;
;
;
;'DYELAS' SUBROUTINE ALLOWS ONE TO SET THE DYE LASER TO A GIVEN WAVELENGTH
;OR TO SET PARAMETERS OF THE CALLING PROGRAM WHICH RESULT IN THE LASER BEING
;SCANNED FROM WAVELENGTH ONE TO WAVELENGTH TWO WITH THE DESIRED GRID SIZE.
;
DYELAS: MOV     R5,R5DL                 ;SAVE R5
        MOV     R1,R1DL                 ;SAVE R1
        CMP     SCCODE,#'LA             ;IS THE LASER BEING SCANNED?
        BEQ     5$
        .PRINT  #M1
        CALL    REWAVE                  ;READ IN WAVELENGTH
        CALL    DYESET                  ;SET THE GRATING
        MOV     R5DL,R5                 ;REGISTER UNSAVE
        RETURN                          ;RETURN TO CALLER
5$:     .PRINT  #M15                    ;AUTO SET??
```

```
        CALL    READ                    ;CHECK FOR "SKIP" WORD
        CMPB    #'Y,IOBUF               ;SKIP THIS PART?
        BEQ     10$                     ;DO NOT SKIP IF NOT EQUAL
        JMP     210$                    ;SKIP IF EQUAL
10$:    .PRINT  #M2                     ;GET SHORT WAVELENGTH
        CALL    REWAVE
        MOV     RESULT,SHOWAV
        MOV     RESULT+2,SHOWAV+2
30$:    .PRINT  #M3                     ;ASK FOR LONG WAVELENGTH
        CALL    REWAVE
        MOV     RESULT,LONWAV
        MOV     RESULT+2,LONWAV+2
        MOV     RESULT,ARG2             ;COMPARE SHORT TO LONG WAVELENGTH
        MOV     RESULT+2,ARG2+2
        MOV     SHOWAV,ARG1
        MOV     SHOWAV+2,ARG1+2
        MOV     #LIST,R5
        CALL    JCMP
        CMP     #-1,R0
        BEQ     40$
        .PRINT  #M6
        BR      30$
40$:    .PRINT  #M4                     ;GET STEP SIZE
        CALL    REWAVE
        MOV     RESULT,ARG1
        MOV     RESULT+2,ARG1+2
        MOV     #48.,ARG2               ;ONE STEP=48 UNITS
        MOV     #0,ARG2+2
        MOV     #LIST,R5
        CALL    JDIV
        CMP     #1,R0                   ;ANY NEGATIVES OR ZEROES?
        BEQ     50$
        .PRINT  #M5                     ;MESSAGE
        .PRINT  #M16
        BR      40$                     ;TRY AGAIN
50$:    MOV     RESULT,ARG1
        MOV     RESULT+2,ARG1+2
        MOV     #32767.,ARG2
        MOV     #LIST,R5
        CALL    JCMP
        TST     R0
        BLE     160$
        .PRINT  #M7                     ;STEP SIZE TOO LARGE
        JMP     40$
160$:   MOV     ARG1,NSTEP              ;NUMBER OF STEPS PER CHANNEL
        MOV     #48.,ARG2               ;MAKE WAVINC=48*NSTEP
        MOV     #0,ARG2+2
        MOV     #LIST,R5
        CALL    JMUL
        MOV     RESULT,WAVINC
        MOV     RESULT+2,WAVINC+2
        MOV     LONWAV,ARG1
        MOV     LONWAV+2,ARG1+2
        SUB     SHOWAV,ARG1
        SBC     ARG1+2
```

```
        SUB     SHOWAV+2,ARG1+2
        MOV     WAVINC,ARG2
        MOV     WAVINC+2,ARG2+2
        MOV     #LIST,R5
        CALL    JDIV
        TST     RESULT+2
        BEQ     180$
170$:   .PRINT  #M8                     ;RESULTS IN IMPROPER # OF CHANNELS
        BR      40$                     ;GET NEW STEP SIZE
180$:   CMP     #1000.,RESULT           ;OVER 1000. CHANNELS?
        BGE     190$
        BR      170$
190$:   CMP     RESULT,#0               ;ZERO CHANNELS?
        BGT     200$
        BR      170$
200$:   MOV     RESULT,NCHAN
        ADD     #1,NCHAN                ;NUMBER OF CHANNELS
        MOV     SHOWAV,RESULT           ;NOW MOVE THE LASER GRATING!!
        MOV     SHOWAV+2,RESULT+2
        CALL    DYESET
        BR      230$
210$:   .PRINT  #M9                     ;ENTER NUMBER OF MOTOR STEPS
        CALL    READ
        CALL    ASCBIN                  ;CONVERT TO BINARY NUMBER
        BCC     212$                    ;BRANCH IF NO ERROR
        .PRINT  #M12                    ;PRINT ERROR MESSAGE
        BR      210$                    ;GET NEW NSTEP
212$:   TST     R1
        BGE     215$
        .PRINT  #M5
        BR      210$
215$:   CMP     #20800.,R1              ;LESS THAN 20800.?
        BPL     220$                    ;YES,BRANCH AWAY
        .PRINT  #M7                     ;NO! PRINT MESSAGE
        BR      210$                    ;GO BACK
220$:   MOV     R1,NSTEP
        MOV     R1,ARG1                 ;WAVINC=NSTEP*48
        MOV     #0,ARG1+2
        MOV     #48.,ARG2
        MOV     #0,ARG2+2
        MOV     #LIST,R5
        CALL    JMUL
        MOV     RESULT,WAVINC
        MOV     RESULT+2,WAVINC+2
        MOV     SETWAV,SHOWAV           ;ASSUMES SETWAV IS STARTING POINT
        MOV     SETWAV+2,SHOWAV+2
        MOV     RESULT,ARG1             ;LONWAV=NCHAN*WAVINC
        MOV     RESULT+2,ARG1+2
        MOV     NCHAN,ARG2
        MOV     #0,ARG2+2
        MOV     #LIST,R5
        CALL    JMUL
        MOV     RESULT,LONWAV
        MOV     RESULT+2,LONWAV+2
230$:   MOV     R5DL,R5                 ;UNSAVE REGISTER
```

```
        MOV     R1DL,R1
DLOUT:  RETURN                          ;RETURN TO CALLER
;
;
;
;
;
;DYESET SUBROUTINE SETS THE DYE LASER GRATING TO THE GIVEN VALUE. THE
;VALUE IS PASSED TO DYESET THROUGH RESULT(LOW WORD) AND RESULT+2 (THE
;HIGH WORD).
;
DYESET: MOV     R1,R1DS                 ;SAVE REGISTERS
        MOV     R2,R2DS
        MOV     R3,R3DS
        MOV     R4,R4DS
        MOV     R5,R5DS
        MOV     RESULT,R1
        MOV     RESULT+2,R2
        SUB     SETWAV,RESULT
        SBC     RESULT+2
        SUB     SETWAV+2,RESULT+2
        MOV     R1,SETWAV               ;NEW SET VALUE
        MOV     R2,SETWAV+2
        MOV     RESULT,ARG1
        MOV     RESULT+2,ARG1+2
        MOV     #48.,ARG2
        MOV     #0,ARG2+2
        MOV     #LIST,R5
        CALL    JDIV
        MOV     R0,NEGFLG
        CMP     #-1,R0
        BNE     10$
        COM     RESULT+2
        COM     RESULT
        ADD     #1,RESULT
        ADC     RESULT+2
10$:    MOV     #32767.,ARG1
        MOV     #0,ARG1+2
20$:    MOV     RESULT,ARG2
        MOV     RESULT+2,ARG2+2
        MOV     #LIST,R5
        CALL    JCMP
        TST     R0
        BGT     30$
        MOV     #32767.,R1
        MUL     NEGFLG,R1
        CALL    MOTOR1
        SUB     #32767.,RESULT
        SBC     RESULT+2
        BR      20$
30$:    MOV     RESULT,R1
        MUL     NEGFLG,R1
        CALL    MOTOR1
        MOV     R5DS,R5
        MOV     R4DS,R4
```

```
        MOV     R3DS,R3
        MOV     R2DS,R2
        MOV     R1DS,R1
DSOUT:  RETURN                          ;RETURN TO CALLER
;
;
;
;
;SUBROUTINE REWAVE READS ASCII CODED NUMBER INTO SYSTEM. A DECIMAL
;MAY BE USED. FOR EXAMPLE 4567.234 WILL BE READ CORRECTLY; BUT NO
;MORE THAN THREE PLACES PAST THE DECIMAL ARE ALLOWED. THE ENTRIES
;4567,4567.,4567.0,4567.00,4567.000 ARE ALL READ CORRECTLY. NUMBERS
;FROM 0.000 TO 7200 ARE READ. LARGER NUMBERS ARE VOIDED.
;BINARY NUMBER IS RETURNED IN RESULT(LOW WORD) AND RESULT+2(HIGH WORD).
;
REWAVE: MOV     R1,R1REW                ;SAVE REGISTERS
        MOV     R2,R2REW
        MOV     R3,R3REW
        MOV     R4,R4REW
        MOV     R5,R5REW
10$:    CLR     R1                      ;CHARACTER COUNTER
        MOV     #IOBUF,R2               ;I/O BUFFER POINTER IN R2
        MOV     #MULFAC,R4              ;MULTIPLICATION FACTORS
        CLR     R3                      ;DECIMAL POINT FLAG
20$:    .TTYIN                          ;GET CHARACTER
        CMPB    #12,R0                  ;IS IT A LINE FEED?
        BEQ     60$                     ;IF YES,LEAVE
        CMP     R1,#8.                  ;ONLY EIGHT CHARACTERS PASS
        BNE     30$
        .PRINT  #M10                    ;TOO MANY CHARACTERS!!!
25$:    .TTYIN                          ;SWALLOW OTHER CARACTERS
        CMPB    #12,R0
        BNE     25$
        BR      10$                     ;BRANCH TO START
30$:    CMPB    #'.,R0                  ;LOOK FOR DECIMAL
        BNE     40$
        MOV     #1,R3                   ;DECIMAL FLAG UP
        BR      20$
40$:    CMP     #1,R3                   ;HAS THERE BEEN A DECIMAL?
        BNE     50$
        CMP     #15,R0                  ;SKIP THE CARRIAGE RETURN
        BEQ     50$
        TST     (R4)+                   ;GET DIFFERENT MULTIPLIER
50$:    MOVB    R0,(R2)+                ;PLACE CHARACTER INTO BUFFER
        INC     R1                      ;INCREMENT CHARACTER COUNTER
        BR      20$                     ;GET MORE
60$:    CALL    ASCBIN                  ;CONVERT TO 32 BIT BINARY NO.
        BCC     130$                    ;BRANCH IF NO ERRORS
        .PRINT  #M12
        BR      10$                     ;TRY AGAIN
130$:   MOV     R1,ARG1
        MOV     R2,ARG1+2
        MOV     (R4),ARG2
        MOV     #0,ARG2+2
        MOV     #LIST,R5
```

```
        CALL    REGSAV                  ;SAVE REGISTERS R1-R5
        CALL    JMUL
        CALL    REGRES                  ;RESTORE REGISTERS R1-R5
        CMP     #-2,R0                  ;WAS THERE AN OVERFLOW?
        BNE     135$                    ;IF NO,BRANCH
        .PRINT  #M17
        .PRINT  #M11                    ;PRINT OVERFLOW MESSAGE
        BR      10$                     ;START OVER
135$:   MOV     RESULT,ARG1             ;IS THE RESULT TOO LARGE?
        MOV     RESULT+2,ARG1+2
        MOV     #0,ARG2
        MOV     #160,ARG2+2             ;7300.000 MAXIMUM
        CALL    REGSAV                  ;SAVE   R1-R5
        CALL    JCMP
        CALL    REGRES                  ;RESTORE  R1-R5
        CMP     #-1,R0
        BEQ     140$
        .PRINT  #M11
        BR      10$                     ;START OVER. TOO LARGE!
140$:   MOV     R5REW,R5                ;UNSAVE REGISTERS
        MOV     R4REW,R4
        MOV     R3REW,R3
        MOV     R2REW,R2
        MOV     R1REW,R1
RWOUT:  RETURN                          ;RETURN TO CALLER
;
;
;
;
;       THIS SUBROUTINE ALLOWS ONE TO 'CALIBRATE' THE DYE LASER
;BY SETTING THE GRATING TO A GIVEN POSITION AND ASSOCIATING A
;WAVELENGTH WITH THE POSITION.
;
DYECAL: MOV     R1,R1DC                 ;SAVE R1
        .PRINT  #M13                    ;NEW CAL WAVELENGTH??
20$:    CALL    READ
        CMPB    #'Y,IOBUF               ;YES OR NO
        BEQ     30$
        CMPB    #'N,IOBUF
        BEQ     DCOUT
        .PRINT  #M13
        BR      20$
30$:    .PRINT  #M14                    ;GET NEW WAVELENGTH
        CALL    REWAVE
        MOV     RESULT,SETWAV
        MOV     RESULT,CALWAV
        MOV     RESULT+2,SETWAV+2
        MOV     RESULT+2,CALWAV+2
        .PRINT  #M18
        .PRINT  #M19
DCOUT:  MOV     R1DC,R1
        RETURN                          ;RETURN TO CALLER
;
;
;
```

```
;
;        MESSAGE SECTION
;
M1:      .ASCII<15><12>/DESIRED WAVELENGTH = /
         .BYTE   200
M2:      .ASCII<15><12>/SHORT WAVELENGTH = /
         .BYTE   200
M3:      .ASCII<15><12>/LONG WAVELENGTH = /
         .BYTE   200
M4:      .ASCII<15><12>/STEP SIZE = /
         .BYTE   200
M5:      .ASCIZ<15><12>/PLEASE!  NO NEGATIVES!/
M6:      .ASCIZ<15><12>/LONG WAVELENGTH MUST BE GREATER THAN SHORT WAVELENGTH
/
M7:      .ASCIZ<15><12>/STEP SIZE TOO LARGE!! ENTER SMALLER VALUE/
M8:      .ASCIZ<15><12>/VALUES ENTERED RESULT IN IMPROPER NO. OF DATA CHANNEL
S/
M9:      .ASCII<15><12>/NUMBER OF MOTOR STEPS = /
         .BYTE   200
M10:     .ASCII<15><12>/TOO MANY CHARACTERS.  TRY AGAIN: /
         .BYTE   200
M11:     .ASCII<15><12>/WAVELENGTH TOO LONG.  TRY AGAIN: /
         .BYTE   200
M12:     .ASCII<15><12>/PLEASE ENTER A NUMBER! TRY AGAIN: /
         .BYTE   200
M13:     .ASCII<15><12>/DO YOU WISH TO CHANGE THE CALIBRATION WAVELENGTH? /
         .BYTE   200
M14:     .ASCII<15><12>/ENTER CALIBRATION WAVELENGTH: /
         .BYTE   200
M15:     .ASCII<15><12>/AUTO SET? /
         .BYTE   200
M16:     .ASCIZ<15><12>/PLEASE!! NO ZEROES!!/
M17:     .ASCIZ<15><12>/MULTIPLICATION OVERFLOW !!!/
M18:     .ASCIZ<15><12>/SET THE DYE LASER DIAL TO READ/
M19:     .ASCIZ<15><12>/FIVE TIMES THE CALIBRATION WAVELENGTH./
         .EVEN
;
;
;
;        LABEL DEFINITIONS
;
R1DC:    .WORD
R1DL:    .WORD
R5DL:    .WORD
R1DS:    .WORD
R2DS:    .WORD
R3DS:    .WORD
R4DS:    .WORD
R5DS:    .WORD
R1REW:   .WORD
R2REW:   .WORD
R3REW:   .WORD
R4REW:   .WORD
R5REW:   .WORD
NEGFLG:  .WORD
```

```
MULFAC: .WORD   1000.,100.,10.,1
;
;
;
        .END
```

```
        .TITLE  MAGNET
;
;SUBROUTINE TO CONTROL THE MAGNETIC SECTOR MASS SPECTROMETER UNDER SCAN MODE
.
;
;
;       SYSTEM MACROS
;
        .MCALL  .PRINT,.TTYOUT
;
;       GLOLIB  LIBRARY GLOBALS
;
        .GLOBL  READ,ASCBIN,DECOUT,BUFINIT,CLEAR
;
;
;       GLOBALS DEFINED IN CALLING PROGRAM
;
        .GLOBL  SCCODE,NSTEP,NCHAN,MADAC,MADAC1,MADAC2,MARES
        .GLOBL  SCFUNC,IOBUF,UPDATE
;
;
;       THIS IS GLOBAL ROUTINE 'MAGNET'
        .GLOBL  MAGNET
;
;
MAGNET: MOV     R1,R1MAG                ;SAVE REGISTERS
        MOV     R2,R2MAG
        MOV     R3,R3MAG
        MOV     R4,R4MAG
        MOV     R5,R5MAG
        CMP     #'MA,SCCODE             ;IS THE MAGNET BEING SCANNED?
        BEQ     4$
        .PRINT  #M6                     ;DEVICE NOT ACTIVE
        CLR     UPDATE                  ;CAUSE DISPLAY UPDATE
        JMP     MAGOUT
4$:     .PRINT  #M1
        MOV     MADAC1+2,R1             ;TYPE PRESENT VALUE
        CALL    DECOUT
        MOV     #' ,R0                  ;TYPE A SPACE
        .TTYOUT
        CALL    READ
        CMPB    #15,IOBUF               ;A CR?
        BEQ     15$
        CALL    ASCBIN                  ;REPLY IN R1
        BCC     5$
        .PRINT  #M4
        BR      4$
5$:     TST R1
        BMI     10$
        CMP     #99.,R1
        BPL 15$
10$:    .PRINT  #M4
        BR      4$                      ;TRY AGAIN
15$:    MOV     R1,MADAC1+2             ;SAVE LOW VALUE
18$:    .PRINT  #M2
```

```
        MOV     MADAC2,R1               ;TYPE PRESENT VALUE
        CALL    DECOUT
        MOV     #' ,R0                  ;TYPE A SPACE
        .TTYOUT
        CALL    READ
        CMPB    #15,IOBUF               ;A CR
        BEQ     20$
        CALL    ASCBIN                  ;HIGH VALUE IN R1
        BCC     20$
        .PRINT  #M4
        BR      18$
20$:    CMP     R1,MADAC1+2             ;HIGH VALUE>LOW VALUE?
        BLE     25$
        CMP     #100.,R1                ;HIGH VALUE<100?
        BPL     30$
25$:    .PRINT  #M4
        BR      18$                     ;TRY AGAIN
30$:    MOV     R1,MADAC2               ;SAVE HIGH VALUE
        MOV     MADAC1+2,R1             ;MULT. BY 34.
        MUL     #34.,R1                 ;34. #'S PER UNIT
        MOV     R1,MADAC                ;SET A-2-D TO LOW CURRENT
        MOV     R1,MADAC1               ;SAVE COPY
        MOV     MADAC2,R3               ;HIGH CURRENT VALUE TO R3
        SUB     MADAC1+2,R3             ;CURRENT SPAN
35$:    .PRINT  #M3
        MOV     MARES+2,R1              ;TYPE PRESENT VALUE
        CALL    DECOUT
        MOV     #' ,R0                  ;TYPE A SPACE
        .TTYOUT
        CALL    READ
        CMPB    #15,IOBUF               ;A CR?
        BEQ     55$
        CALL    ASCBIN                  ;REPLY IN R1(BINARY)
        BCC     40$
        .PRINT  #M4
        BR      35$
40$:    TST     R1
        BLE     45$                     ;MUST BE GREATER THAN ZERO
        CMP     #35.,R1                 ;AND LESS THAN 35.
        BPL     50$
45$:    .PRINT  #M4
        BR      35$                     ;TRY AGAIN
50$:    MOV     R1,MARES+2              ;SAVE ENTERED VALUE
55$:    MOV     R1,MARES                ;INCREMENT VALUE
        MUL     #34.,R3                 ;R3 CONTAINS MASS SPAN
        CLR     R2
        DIV     MARES,R2
        CMP     #1000.,R2               ;IS IT LESS THAN 1000.?
        BGE     60$                     ;BRANCH IF YES
        .PRINT  #M5
        BR      35$
60$:    MOV     R2,NCHAN
MAINIT: CALL    BUFINIT                 ;GENERATE PROPER BUFFER POINTERS
        CALL    CLEAR                   ;CLEAR DATA BUFFERS
MAGOUT: MOV     R5MAG,R5                ;RESTORE REGISTERS
```

```
        MOV     R4MAG,R4
        MOV     R3MAG,R3
        MOV     R2MAG,R2
        MOV     R1MAG,R1
        RETURN
;
;
;       LABELS
R1MAG:  .WORD
R2MAG:  .WORD
R3MAG:  .WORD
R4MAG:  .WORD
R5MAG:  .WORD
;
;
M1:     .ASCII<15><12>/LOW VALUE(0-100) =/<200>
M2:     .ASCII<15><12>/HIGH VALUE       =/<200>
M3:     .ASCII<15><12>/STEPS PER CHANNEL=/<200>
M4:     .ASCIZ<15><12>/INVALID REPLY!! TRY AGAIN./
M5:     .ASCIZ<15><12>/RESULTS IN TOO MANY CHANNELS: ONLY 1000 ARE ALLOWED./
M6:     .ASCIZ<15><12>/DEVICE IS NOT ACTIVE/
        .EVEN
;
;
        .END
```

```
        .TITLE  MOTOR1
;MOTOR SUBROUTINE SENDS 'R1' PULSES TO STEPPING MOTOR.
;POSITIVE NUMBER FOR CLOCKWISE ROTATION AND A NEGATIVE
;NUMBER FOR COUNTERCLOCKWISE ROTATION. USE WITH A
;'CALL MOTOR1'STATEMENT.
        .MCALL  .PRINT
        .GLOBL  MOTOR1
DATALO=166000
MOTOR1: MOVB    #1,@#166040     ;READ MOTOR STATUS WORD
        BIT     #1,DATALO       ;IS LAM SET?
        BEQ     MOTOR1          ;NO,WAIT
        BIT     #2,DATALO       ;IS CCWL SET?
        BEQ     10$             ;NO
        .PRINT #M1              ;YES-PRINT WARNING
10$:    BIT     #4,DATALO       ;IS CWL SET?
        BEQ     20$             ;NO
        .PRINT  #M2             ;YES-PRINT WARNING
20$:    TST     R1              ;IS R1 NEGATIVE?
        BGE     30$             ;NO
        NEG     R1              ;YES-MAKE POSITIVE
        BIS     #100000,R1      ;SIGN BIT = 1
30$:    BEQ     40$             ;DO NOT LOAD A ZERO
        MOV     R1,@#166040     ;LOAD COUNTER
40$:    RETURN
M1:     .ASCIZ<15><12>/WARNING!! CCW LIMIT OF MOTOR 1 REACHED!/
        .EVEN
M2:     .ASCIZ<15><12>/WARNING!! CW LIMIT OF MOTOR 1 REACHED!/
        .EVEN
        .END
```

```
        .TITLE  MOTOR2
;MOTOR SUBROUTINE SENDS 'R1' PULSES TO STEPPING MOTOR.
;POSITIVE NUMBER FOR CLOCKWISE ROTATION AND A NEGATIVE
;NUMBER FOR COUNTERCLOCKWISE ROTATION. USE WITH A
;'CALL MOTOR2'STATEMENT.
        .MCALL  .PRINT
        .GLOBL  MOTOR2
DATALO=166000
MOTOR2: MOVB    #1,@#166040       ;READ MOTOR STATUS WORD
        BIT     #20,DATALO        ;IS LAM SET?
        BEQ     MOTOR2            ;NO,WAIT
        BIT     #40,DATALO        ;IS CCWL SET?
        BEQ     10$               ;NO
        .PRINT #M1                ;YES-PRINT WARNING
10$:    BIT     #100,DATALO       ;IS CWL SET?
        BEQ     20$               ;NO
        .PRINT  #M2               ;YES-PRINT WARNING
20$:    TST     R1                ;IS R1 NEGATIVE?
        BGE     30$               ;NO
        NEG     R1                ;YES-MAKE POSITIVE
        BIS     #100000,R1        ;SIGN BIT = 1
30$:    BEQ     40$               ;DO NOT LOAD A ZERO
        MOV     R1,@#166042       ;LOAD COUNTER
40$:    RETURN
M1:     .ASCIZ<15><12>/WARNING!! CCW LIMIT OF MOTOR 2 REACHED!/
        .EVEN
M2:     .ASCIZ<15><12>/WARNING!! CW LIMIT OF MOTOR 2 REACHED!/
        .EVEN
        .END
```

```
        .TITLE   QUAD
;
;SUBROUTINE TO CONTROL THE QUADRUPOLE MASS SPECTROMETER UNDER SCAN MODE.
;
;
;       SYSTEM MACROS
;
        .MCALL   .PRINT,.TTYOUT
;
;       GLOLIB   LIBRARY GLOBALS
;
        .GLOBL   READ,ASCBIN,DECOUT,BUFINIT,CLEAR
;
;
;       GLOBALS DEFINED IN CALLING PROGRAM
;
        .GLOBL   SCCODE,NSTEP,NCHAN,MDAC,MDAC1,MDAC2,MRES
        .GLOBL   MASBUF,SCFUNC,QP,IOBUF,UPDATE
;
;
;       THIS GLOBAL ROUTINE 'QUAD'
        .GLOBL   QUAD
;
;
QUAD:   MOV      R1,R1QUA                ;SAVE REGISTERS
        MOV      R2,R2QUA
        MOV      R3,R3QUA
        MOV      R4,R4QUA
        MOV      R5,R5QUA
        CMP      #'QU,SCCODE             ;IS THE QUAD. BEING SCANNED?
        BEQ      4$
        CMP      #'QP,SCCODE             ;SCAN PEAK TO PEAK?
        BNE      1$
        MOV      #QP,SCFUNC
        JMP      QUPKS
1$:     .PRINT   #M9                     ;DEVICE NOT ACTIVE
        CLR      UPDATE                  ;CAUSE A DISPLAY UPDATE
        JMP      QUAOUT
4$:     .PRINT   #M1
        MOV      MDAC1+2,R1              ;PRINT EXISTING VALUE
        CALL     DECOUT
        MOV      #'  ,R0                 ;TYPE A SPACE
        .TTYOUT
        CALL     READ
        CMPB     #15,IOBUF               ;IS IT A CR?
        BEQ      18$                     ;IF YES GO TO NEXT QUESTION
        CALL     ASCBIN                  ;REPLY IN R1
        BCC      5$
        .PRINT   #M4
        BR       4$
5$:     TST R1
        BMI      10$
        CMP      #200.,R1
        BPL 15$
10$:    .PRINT   #M4
```

```
        BR      4$                      ;TRY AGAIN
15$:    MOV     R1,MDAC1+2              ;SAVE LOW MASS VALUE
18$:    .PRINT  #M2
        MOV     MDAC2,R1                ;PRINT EXISTING VALUE
        CALL    DECOUT
        MOV     #' ,R0                  ;TYPE A SPACE
        .TTYOUT
        CALL    READ
        CMPB    #15,IOBUF               ;A CR?
        BEQ     20$
        CALL    ASCBIN                  ;HIGH MASS IN R1
        BCC     20$
        .PRINT  #M4
        BR      18$
20$:    CMP     R1,MDAC1+2              ;HIGH MASS>LOW MASS?
        BLE     25$
        CMP     #200.,R1                ;HIGH MASS<200?
        BPL     30$
25$:    .PRINT  #M4
        BR      18$                     ;TRY AGAIN
30$:    MOV     R1,MDAC2                ;SAVE HIGH MASS VALUE
        MOV     MDAC1+2,R1              ;MUL BY 20.
        MUL     #20.,R1                 ;20. #'S PER UNIT MASS
        MOV     R1,MDAC                 ;SET A-2-D TO LOW MASS
        MOV     R1,MDAC1                ;SAVE COPY
        MOV     MDAC2,R3                ;HIGH VALUE TO R3
        SUB     MDAC1+2,R3              ;MASS SPAN
35$:    .PRINT  #M3
        MOV     MRES+2,R1               ;EXISTING VALUE
        CALL    DECOUT
        MOV     #' ,R0                  ;TYPE A SPACE
        .TTYOUT
        CALL    READ
        CMPB    #15,IOBUF               ;A CR?
        BEQ     55$
        CALL    ASCBIN                  ;REPLY IN R1(BINARY)
        BCC     40$
        .PRINT  #M4
        BR      35$
40$:    TST     R1
        BLE     45$                     ;MUST BE GREATER THAN ZERO
        CMP     #21.,R1                 ;AND LESS THAN 21.
        BPL     50$
45$:    .PRINT  #M4
        BR      35$                     ;TRY AGAIN
50$:    MOV     R1,MRES+2               ;SAVE ENTERED MRES VALUE
55$:    MOV     R1,MRES                 ;INCREMENT VALUE
        MUL     #20.,R3                 ;R3=MASS SPAN*20.
        CLR     R2
        DIV     MRES,R2
        CMP     #1000.,R2               ;IS IT LESS THAN 1000.?
        BGE     60$                     ;BRANCH IF YES
        .PRINT  #M5
        BR      35$
60$:    MOV     R2,NCHAN
```

```
QUINIT: CALL    BUFINIT         ;GENERATE PROPER BUFFER POINTERS
        CALL    CLEAR           ;CLEAR DATA BUFFERS
QUAOUT: MOV     R5QUA,R5        ;RESTORE REGISTERS
        MOV     R4QUA,R4
        MOV     R3QUA,R3
        MOV     R2QUA,R2
        MOV     R1QUA,R1
        RETURN
;
;
QUPKS:  .PRINT  #M6             ;ASK FOR NUMBER OF SEGMENTS
        CALL    READ            ;GET RESPONSE
        CALL    ASCBIN          ;CONVERT TO BINARY
        BCC     5$              ;BRANCH IF NO ERROR
1$:     .PRINT  #M4
        BR      QUPKS
5$:     TST     R1              ;ZERO?
        BEQ     1$
        MOV     R1,R5           ;LOOP COUNTER
        CLR     NCHAN           ;CLEAR CHANNEL COUNTER
        MOV     #1,R4           ;SEGMENT COUNTER
        MOV     #MASBUF,R3      ;MASS BUFFER STARTING LOCATION
10$:    .PRINT  #M7             ;ASK FOR STARTING CHANNEL NUMBERS
        MOV     R4,R1
        CALL    DECOUT
        MOV     #' ,R0          ;TYPE A SPACE
        .TTYOUT
        MOV     #'=,R0          ;TYPE =
        .TTYOUT
        MOV     #' ,R0          ;TYPE ANOTHER SPACE
        .TTYOUT
        CALL    READ            ;GET REPLY
        CALL    ASCBIN          ;CONVERT TO BINARY
        BCC     15$             ;BRANCH IF NO ERROR
12$:    .PRINT  #M4
        BR      10$             ;TRY AGAIN
15$:    TST     R1              ;ZERO?
        BEQ     12$
        SUB     #1,R1           ;CHANNEL 1 = ZERO OFFSET
        MUL     MRES,R1         ;'MRES' UNITS PER CHANNEL
        ADD     MDAC1,R1        ;STARTING A-2-D VALUE
        MOV     R1,(R3)+        ;PLACE IN MASS BUFFER
20$:    .PRINT  #M8             ;NUMBER OF CHANNELS IN SEGMENT?
        CALL    READ            ;GET REPLY
        CALL    ASCBIN          ;CONVERT TO BINARY(REPLY IN R1)
        BCC     23$             ;BRANCH IF NO ERROR
22$:    .PRINT  #M4
        BR      20$             ;TRY AGAIN
23$:    TST     R1              ;ZERO?
        BEQ     22$
25$:    MOV     -2(R3),(R3)
        ADD     MRES,(R3)+      ;CREATE SCAN ARRAY
        INC     NCHAN           ;COUNT NUMBER OF CHANNELS
        CMP     #100.,NCHAN     ;ALLOW ONLY 100 CHANNELS
        BPL     30$
```

```
        BR      35$
30$:    SOB     R1,25$                  ;DO ALL CHANNELS
        INC     R4                      ;SEGMENT COUNTER
        DEC     R5                      ;DO ALL SEGMENTS
        BEQ     35$
        JMP     10$                     ;GO DO 'EM
35$:    MOV     MASBUF,MDAC             ;SET QUADRUPOLE TO FIRST MASS
        JMP     QUINIT                  ;LEAVE
;
;
;
;       LABELS
R1QUA:  .WORD
R2QUA:  .WORD
R3QUA:  .WORD
R4QUA:  .WORD
R5QUA:  .WORD
;
;
M1:     .ASCII<15><12>/LOW MASS  =/<200>
M2:     .ASCII<15><12>/HIGH MASS =/<200>
M3:     .ASCII<15><12>/STEPS PER CHANNEL=/<200>
M4:     .ASCIZ<15><12>/INVALID REPLY!! TRY AGAIN./
M5:     .ASCIZ<15><12>/RESULTS IN TOO MANY CHANNELS: ONLY 1000 ARE ALLOWED./
M6:     .ASCII<15><12>/NUMBER OF MASS SEGMENTS = /<200>
M7:     .ASCII<15><12>/STARTING CHANNEL NUMBER OF SEGMENT/<200>
M8:     .ASCII<15><12>/NUMBER OF CHANNELS IN THE SEGMENT = /<200>
M9:     .ASCIZ<15><12>/DEVICE IS NOT ACTIVE/
        .EVEN
;
;
        .END
```

```
        .TITLE  READ
;READ SUBROUTINE READS  CHARACTERS ON CONSOLE DEVICE
;LOADS BUFFER STARTING AT LOCATION 'IOBUF'
        .MCALL  .TTYIN
        .GLOBL READ,IOBUF
READ:   MOV     R2,R2REA
        MOV     R1,R1REA
        MOV     #IOBUF,R2
        CLR     R1                  ;LOOP COUNTER
10$:    .TTYIN
        CMP     #100.,R1            ;ONLY 100. CHARACTERS ARE ALLOWED
        BEQ     20$                 ;IF YES,EXIT
        CMPB    #12,R0
        BEQ     RE2
        MOVB    R0,(R2)+
        BR      10$
20$:    .TTYIN                      ;EAT THE REST
        CMPB    #12,R0
        BEQ     RE2
        BR      20$
RE2:    MOV     R2REA,R2            ;UNSAVE REGISTERS
        MOV     R1REA,R1
        RTS     PC
R1REA:  .WORD
R2REA:  .WORD
        .END
```

```
      SUBROUTINE PLOTTR(PLCTRL,PLCHAR,PLLINE,DISTRT,DICHNS,STCHAN,PLMOV,
     1YMAX)
      INTEGER PLCTRL,DISTRT,DICHNS,STCHAN
      INTEGER PLLINE,PLCHAR,PLMOV,CTRLC,PLMOVE
      EXTERNAL RUN
      INTEGER*4 YMAX
      REAL*8 DJFLT
      DIMENSION ITOCK(9)
C
C
C     DEFINE SPECIAL CHARACTERS
      CTRLC=3
C
C
C
C     TURN PLOTTER ON.
      CALL    ON
      CALL    ONMODE
      CALL    XONOFF
C
C
C
C     INITIALIZE.
      TYPE    100
100   FORMAT  (' IN;PU;IP704,1850,10306,8103;')
      IF(PLCTRL .EQ. 'PP' .OR. PLCTRL .EQ. 'PM') GO TO 850
      IF(PLCTRL .EQ. 'PC') GO TO 1510
      IF(PLCTRL .EQ. 'PH') GO TO 2110
      IF(PLCTRL .EQ. 'PV') GO TO 2200
      IF(PLCTRL .EQ. 'PB') GO TO 2300
C
C
C
C     DRAW THE BOX.
      TYPE    200
200   FORMAT  (' VS10;PA10306,8103;PD;')
      TYPE    250
250   FORMAT  (' PA10306,1135,704,1135,704,8103,10306,8103;PU;VS;')
C
C
C
C     CHANGE SCALE--DRAW THE Y-AXIS TICKS.
      TYPE    300
300   FORMAT  (' SC0,101,0,105;PA0,-10;TL1,0;')
      TYPE    350,CTRLC
350   FORMAT  (' YT;CP-3.5,-.25;LB-10',A1)
360   DO      500,I=0,100,10
      TYPE    450,I,I,CTRLC
450   FORMAT  (' PA0,',I3,';YT;CP-3.5,-.25;LB',I3,A1)
500   CONTINUE
      TYPE    550
550   FORMAT  (' TL0,1;')
      DO      600I=-10,100,10
      TYPE    575,I
575   FORMAT  (' PA101,',I3,';YT;')
```

```
600     CONTINUE
C
C
C
C       LABEL THE AXES.
C
        TYPE    700,CTRLC
700     FORMAT  (' SC;SI.25,.5;PA4275,583;LBCHANNEL NUMBER',A1)
        TYPE    800,CTRLC
800     FORMAT  (' DIO,10;PA232,3733;LBCOUNTS/CHANNEL',A1,'DI;')
C
C
C
C       PLOT THE DATA
C
850     PLMOVE=PLMOV*20+PLMOV/2
        TYPE     900,PLLINE,PLCHAR
900     FORMAT  (' LT',A1,';SM',A1';')
        TYPE    1000,DICHNS+1
1000    FORMAT  (' SC0,',I3,',0,2150;S1;')
        JADDR=DISTRT+2
        DO       1500 I=1,DICHNS
        TYPE     1100,I,(IPEEK(JADDR)-2047)+PLMOVE
1100    FORMAT  (' PA',I3,',',I4,';')
        JADDR=JADDR+4
        IF(I .NE. 1) GO TO 1500
        IF(PLLINE .EQ. 32) GO TO 1400
        TYPE     1200
1200    FORMAT  (' PD;')
        GO TO 1500
1400    TYPE     1450
1450    FORMAT  (' PU;')
1500    CONTINUE
        IF(PLCTRL .EQ. 'PP' .OR. PLCTRL .EQ. 'PM') GO TO 9999
C
C
C
C       MAKE THE X-TICKS
C
1510    ITICK=DICHNS
        ITICK=ITICK/10
        DO       1520 I=1,5
        ITOCK(I+2)=I*5
1520    CONTINUE
        ITOCK(1)=1
        ITOCK(2)=2
        ITOCK(8)=50
        ITOCK(9)=100
        DO       1590 I=0,50
        DO       1550 J=1,9
        IF(ITICK .EQ. ITOCK(J)) GO TO 1600
1550    CONTINUE
        ITICK=ITICK+1
1590    CONTINUE
C
```

```
C
C       PLOT THE X-AXIS TICKS AND CHANNEL NUMBERS
C
1600    TYPE    1700
1700    FORMAT  (' PU;SM;LT;TL1.6,0;')
        TYPE    1750,DICHNS+1
1750    FORMAT  (' SC0,',I3,',0,105;SI;')
        IR=0
        IS=0
        ID=5
        IL=DICHNS-1
        IF(ITICK.LT.5)ID=ITICK
        ISAV=MOD(STCHAN,ID)
        IF(ITICK-ISAV.GE.ITICK/2)GO TO 1760
        IS=ITICK
        IR=ISAV
        IL=IL+ITICK
1760    DO      1900 IX=IS,IL,ITICK
        TYPE    1800,IX-IR+1,PLMOV-12,STCHAN+IX-IR,CTRLC
1800    FORMAT  (' PA',I3,',',I3,';XT;CP-2,-1;LB',I3,A1)
        IR=ISAV
1900    CONTINUE
        IX=IX-ITICK
        IF(DICHNS-IX+IR-1.LT.ITICK/2)GO TO 1950
        IX=DICHNS+IR-1
        TYPE    1800,IX-IR+1,PLMOV-12,STCHAN+IX-IR,CTRLC
1950    IF(PLCTRL .EQ. 'PC') GO TO 9999
        TYPE    2000
2000    FORMAT  (' TL0,1.6;')
        DO      2100,IX=0,DICHNS-1,ITICK
        TYPE    2050,IX+1
2050    FORMAT  (' PA',I3,',105;XT;')
2100    CONTINUE
        TYPE    3000,DJFLT(YMAX),CTRLC
3000    FORMAT  (' PA1,-17;CP0,-2.0;LB',F11.0,' FULL SCALE',A1)
        TYPE    3050,IPEEK(IADDR(RUN)),CTRLC
3050    FORMAT  (' CP;CP5,0;LB',I5,'. RUN NUMBER',A1)
        GO TO 9999
C
C       MAKE HORIZONTAL LINES
C
2110    TYPE    2150,PLMOV,PLLINE
2150    FORMAT  (' SC0,101,0,105;PA0,'I3,';LT',A1,';PD;')
        TYPE    2175,PLMOV
2175    FORMAT  (' PA101,',I3,';')
        GO TO 9999
C
C       MAKE VERTICAL LINES.
C
2200    TYPE    2250,DICHNS+1,PLLINE,PLMOV
2250    FORMAT  (' SC0,',I3,',0,105;LT',A1,';PA',I3,',0;PD;')
        TYPE    2260,PLMOV
2260    FORMAT  (' PA',I3,',101;')
        GO TO 9999
C
```

```
C
      SUBROUTINE ON
      INTEGER ESCAPE
      ESCAPE=27
C     EMPTY RING
10    ICHAR=ITTINR(1)
      IF(ICHAR .GT. 0) GO TO 10
      TYPE    20,ESCAPE
20    FORMAT  (1H ,A1,'.Y')
      RETURN
      END
C
C
C
      SUBROUTINE OFF
      INTEGER ESCAPE
      ESCAPE=27
C     SEND 'ESC.L' BUFFER SIZE?
C     PLOTTER RESPONDS WHEN ITS BUFFER IS EMPTY.
      TYPE    10,ESCAPE
10    FORMAT  (1H ,A1,'.L')
20    ICHAR=ITTINR(1)
      IF(ICHAR .LT. 0) GO TO 20
25    ICHAR=ITTINR(1)
      IF(ICHAR .GT. 0) GO TO 25
C     SEND 'ESC.Z' PLOTTER OFF COMMAND
      TYPE    30,ESCAPE
30    FORMAT  (1H ,';',A1,'.Z')
      RETURN
      END
C
C
C
      SUBROUTINE XONOFF
      INTEGER ESCAPE
      ESCAPE=27
      TYPE 20,ESCAPE
20    FORMAT  (1H ,A1,'.I200;;17:')
      TYPE    40,ESCAPE
40    FORMAT  (1H ,A1,'.N0;19:')
      RETURN
      END
C
C
C
      SUBROUTINE ONMODE
      INTEGER ESCAPE
      ESCAPE=27
      TYPE    20,ESCAPE
20    FORMAT  (1H ,A1,'.@;2:')
      RETURN
      END
```

```
C        DRAW BOX ONLY
C
2300     TYPE     2310
2310     FORMAT  (' VS10;PA10306,8103;PD;')
         TYPE     2315
2315     FORMAT  (' PA10306,1850,704,1850,704,8103,10306,8103;PU;')
         GO TO 9999
C
C
C        EXIT THE PROGRAM
C
9999     TYPE     10010
10010    FORMAT  (' PU;SM;SC0,101,0,101;S1;PA101,101;')
         CALL     OFF
         RETURN
         END
```

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720