901075

②

NWC TP 6929

Ka-Band Cubically Symmetric Turnstile Power Combiner

by
D. R. Bowling
Research Department
and
C. L. Kidner
T. W. Williams
Strike Weapons Department

FEBRUARY 1991

NAVAL WEAPONS CENTER
CHINA LAKE, CA 93555-6001

Approved for public release; distribution is unlimited.

Naval Weapons Center

FOREWORD

This report describes the investigation of a novel waveguide six-port turnstile junction used in power combining microwave or millimeter-wave power sources for missile seeker applications. It documents work performed at the Naval Weapons Center during fiscal years 1986 and 1987. The work was supported by the Solid-State Missile Transmitters Task of the Missile Support Technology Block and by Independent Exploratory and Development Funds.

The purpose of this publication is to document initial investigations. Hopefully, it can serve as a starting point for further investigations.

This report was reviewed for technical accuracy by M. Afendykiw.

Approved by
R. L. DERR, *Head*
Research Department
25 February 1991

Under authority of
D. W. COOK
Capt., U.S. Navy
Commander

Released for publication by
W. B. PORTER
Technical Director

NWC Technical Publication 6929

Published by .... Technical Information Department
Collation .... Cover, 107 leaves
First printing .... 65 copies

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE February 1991	3. REPORT TYPE AND DATES COVERED Final, Oct 85 to Sep 87

4. TITLE AND SUBTITLE Ka-Band Cubically Symmetric Turnstile Power Combiner	5. FUNDING NUMBERS PE 62234N PR RS34R40 TA #1 WU 906577
6. AUTHOR(S) D. R. Bowling, C. L. Kidner, and T. W. Williams	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Weapons Center China Lake, CA 93555-6001	8. PERFORMING ORGANIZATION REPORT NUMBER NWC TP 6929
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research Arlington, VA 22217	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION/AVAILABILITY STATEMENT A Statement; public release; distribution unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

(U) This report describes a novel six-port turnstile-type power combiner for use as a four-way transmission- (or reflection-) type power combiner/divider. This unique turnstile is rotationally symmetric about its four major diagonals, thus ensuring equal power division/combination. Each pair of opposing waveguide inputs is orthogonally oriented to provide excellent cross-port isolation. Generalized scattering matrices are used to analyze six-port junction turnstiles of varying degrees of symmetry. Scattering matrix analysis is used to demonstrate further the reflective and transmissive modes of turnstile power combining. Three versions of cubically symmetric turnstile that were evaluated for use in the 26.5- to 40.0-gigahertz frequency band are described. The final turnstile version was integrated with four single-IMPATT oscillator modules to demonstrate transmission-mode power combining. Potential benefits of cascaded and extended level turnstile combining include high-power generation at millimeter- and submillimeter-wave frequencies.

14. SUBJECT TERMS Waveguides — Turnstile Junctions — IMPATT Power Combining Power Combining — Six-Port Junctions	15. NUMBER OF PAGES 212
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

CONTENTS

Introduction 3

Theory of Cubic Turnstile Operation 6
- Premultiplying a Matrix by [M] 13
- Postmultiplying a Matrix by [M] 13
- Computing the Transpose of [M] 14
- Computing the Inverse of [M] 15
- Power-Combining Properties of Matched, Lossless, Cubic Turnstile Junction (Transmission Mode) 19
- Power-Combining Properties of Matched, Lossless, Cubic Turnstile Junction (Reflection Mode) 22
- Effects of Terminator-Pair Amplitude and Phase Imbalance Upon Transmission- and Reflection-Mode Power Combining 22

Experimental Turnstile Hardware 25

IMPATT Oscillator Module Design and Turnstile Integration 30
- Oscillator Design 30
- Transfer Efficiency Optimization 30
- Design of Impedance Transformers 32
- Oscillator and Turnstile Integration 39
- Uniformity Considerations 40
- Measuring the Reflection Amplitude and Phase of Injection-Locked IMPATT Oscillators Under Pulsed RF Conditions 41
- Adjusting the Gain and Phase of Injection-Locked Oscillator Modules . 44
- Oscillator Phase Matching for Initial Demonstration of Turnstile Transmission-Mode Power Combiner 44
- Thermal and Mechanical Considerations 45
- Turnstile Transmission-Mode Power-Combining Demonstration (Initial Test Results) 46

Conclusions and Future Work 48
- Conclusions 48
- Future Work 48

Figures 51

Appendixes:

A. S-Parameter Constraint Matrix of a Six-Port Waveguide Junction . 109
B. List of Thirty Symmetry Operators for Cubic Turnstile Junction . . 157
C. Three-Dimensional Model of the Cubic Turnstile Junction 171
D. Additional Experimental Information Regarding Kurokawa Oscillator Design 173
E. Expression for Power-Added Efficiency of Reflection Amplifier . . 179
F. Effect of Iris Size, Coaxial Line Placement, and Cavity Length Upon Midcavity Impedance 183
G. Package and Mounting Parasitics for Ka-Band IMPATT Used in the Turnstile Oscillators 187
H. Computer Program for Modeling IMPATT Impedance 189
I. Touchstone Circuit and S-Parameter Files 201
J. Oscillator Circuit Uniformity Test Results 207

References 211

ACKNOWLEDGMENT

Special acknowledgments are extended to the following individuals for contributing their unique professional skills resulting in successful completion of this work: Charles F. Smith, for designing and machining the precision hardware components; Arthur R. Craddock, for technical assistance in performing critical experiments to demonstrate hardware performance; and Marko Afendykiw, Darry M. Kinman, and David J. White, for many discussions of power-combining theory and techniques.

Special thanks are given to Jim Cauffman of the Office of Naval Technology for his continued interest and support of this technology.

INTRODUCTION

This report describes a novel waveguide six-port turnstile junction ideally suited for purposes of power combining active microwave or millimeter-wave power modules. To demonstrate the features of this novel power-combining approach, this report focuses on power combining Ka-band impact avalanche and transit time (IMPATT) diode modules. However, the turnstile approach should be suited equally well to power combining other active modules such as Gunn diodes, magnetrons, or klystrons at microwave or even higher millimeter-wave frequencies.

This unique turnstile junction exhibits several properties that are particularly desirable in power-combining applications. As Figure 1 indicates, the structure exhibits a high degree of symmetry with respect to the central combining region. The cubic structure is rotationally symmetric at 120-degree intervals about the four major diagonals of the cube [Figure 2(a)]. Also, the structure is rotationally symmetric at 180-degree intervals about the central axis of any pair of cube faces [Figure 2(b)]. This high degree of symmetry results in equal power division (or combination) properties that are necessary for efficient power combining.

Another feature of this structure, which enhances its power-combining properties, is the inherent cross-port isolation resulting from the orthogonal orientation of each pair of opposing waveguide inputs (see Figure 1). Successful power combining depends upon providing device-to-device isolation to prevent unwanted oscillation modes.

Figures 3(a) and (b) show two possible turnstile configurations for combining the outputs from four power modules. Depending upon the phasing between the modules, the cubic turnstile can be operated in either a reflection [Figure 3(a)] or a transmission [Figure 3(b)] mode. In the reflection mode, all of the combined power is reflected back out the input port with no power transmitted through to the output port. To achieve the reflection mode, all modules must have the same output amplitude and phase. In the transmission mode, all power is transmitted through the junction to the output port with no power reflected from the input port. The transmission mode requires that all modules have the same output amplitude and that opposing pairs of modules have opposite phases.

By combining a single cube operating in the transmission mode with four cubic turnstiles operating in the reflection mode, a combining structure with 16 combination sites could be realized. Such a configuration is shown in Figure 4. The ability to operate in either the reflection or transmission mode and the modular nature of this structure allow great versatility in achieving high-level power combining from a few simple building blocks. For example, the extended-level turnstile configuration of Figure 4 with 16 combination sites, each occupied by a four-diode Kurokawa cavity combiner filled with Ka-band IMPATTS (each generating 3.0 watts of average power), would be capable of a combined output of 150 watts of average power at Ka-band. The size of such a structure (excluding bias circuitry) would be approximately 7 in^3.

Extended-level power combining requires that all ports exhibit good input matching. An essential attribute of the cubic turnstile junction for power-combining applications is its ability to simultaneously match all waveguide inputs over a 10% bandwidth without the addition of any external or internal matching structures. A Ka-band version of the junction demonstrated matched performance (return loss <-30 decibels) over a 2-gigahertz bandwidth and (return loss <-17 decibels) over a 4-gigahertz bandwidth at a center frequency of 34.5 gigahertz. Because of its excellent input matching, the cubic turnstile junction and its complement of four power modules can be treated as a subcircuit module for extended-level power combining, making direct cascade of successive turnstile modules (without interstage circulators) possible. In this fashion, an injection-locked chain of directly cascaded turnstile junctions each with a complement of successively higher powered modules can be envisioned. In Figure 5, the initial stage consists of a turnstile with single IMPATT power modules. The second turnstile stage is surrounded by two diode Kurokawa modules, and the third stage is comprised of a turnstile surrounded by four Kurokawa cavities each containing four IMPATTs. (The extended-level combiner of Figure 4 could be added to complete a 150-watt Ka-band transmitter.)

While the overall turnstile-based transmitter would be extremely compact (<15 in^3), the power-generating devices are distributed throughout the structure and allow more even dissipation of excess heat. The distributed heat sinking of the turnstile alleviates thermal "hot spots" encountered using conventional approaches in which many high-power devices are confined to a single power-combining structure. Thermal management becomes increasingly important as more devices are required to achieve higher power levels and as higher frequencies require smaller circuit dimensions and closer device spacings. If necessary, active cooling channels can be incorporated in the body of the cubic turnstile junction to carry away excess heat from adjacent power modules. In this way, modules only require provisions for passive cooling, thus reducing cost, size, and weight.

At higher millimeter-wave frequencies (i.e., 94 gigahertz and above), minimum spacing requirements because of physical device size exceed available waveguide circuit dimensions; consequently, conventional multidevice power-combining approaches such as Kurokawa cavities are not feasible. The turnstile junction offers a viable approach to multidevice power combining at high millimeter-wave frequencies. Even at G-band (140 to 200 gigahertz), waveguide dimensions of 0.0255 x 0.051 inch are sufficient to allow a single coaxial line through the waveguide cavity, thus permitting construction of a single-device module. The cubic turnstile junction, in turn, can provide the modular approach necessary for achieving cascaded or extended-level combining at high millimeter-wave frequencies.

Another attractive feature of the cubic turnstile structure is its high-power handling capability. The central power-combining region contains no probes or sharp discontinuities that might be susceptible to arcing under high-power conditions. There are no lossy structures internal to the combiner to absorb power and produce excessive heating under high-power conditions.

Since the cubic turnstile structure is essentially lossless, its combining efficiency approaches 100%. High combining efficiency is essential for extended-level power combining. Measured data indicate that high-level combining efficiency can be maintained over a 25% frequency bandwidth at Ka-band.

It is believed that this turnstile configuration is uniquely different from those previously used in power-combining applications. Eisenhart *et al.* describe a combiner configuration in which the device ports are all coplanar, while the input and output ports are mutally orthogonal (Reference 1). Depicted by Ginzton is a symmetrical six-arm waveguide junction for use as a microwave equivalent of a Wheatstone bridge (Reference 2). Ginzton's junction included external screws for matching; its interior was nonspherical. A six-port junction whose opposing input pairs are all mutually orthogonal with three adjacent coplanar input pairs is described by Purcell (Reference 3). (The geometry of these various six-port waveguide junctions is discussed in Appendix A.)

The following section of this report describes a method for characterizing a six-port waveguide junction in terms of generalized scattering parameters based upon the various types and degrees of symmetries exhibited by the junction. Also contained in the section are detailed discussions regarding orientation and sign conventions, shorthand notation of symmetry operators, and instructions for using a computer program developed for analyzing six-port waveguide junctions of arbitrary port configurations. Reflection- and transmission-mode operations of the cubic turnstile combiner are explained in detail using scattering- (S) parameter analysis. Additional performance parameters such as combining efficiency and active input impedance are also explained using S-parameter techniques.

The subsequent section on turnstile hardware discusses the design and integration of the IMPATT oscillator modules used to demonstrate transmission-mode power combining of the cubic turnstile combiner. Various issues of IMPATT oscillator design are discussed briefly. Topics include oscillator transfer efficiency optimization, oscillator cavity impedance characterization, impedance matching transformer design, oscillator injection-locked power performance, and amplitude and phase matching of IMPATT power modules. Initial test results demonstrating successful transmission-mode power combining of IMPATT modules using the cubic turnstile combiner conclude these discussions.

Demonstrated and potential capabilities of the cubic turnstile power combiner are summarized in the last section; also discussed are the potential impacts of turnstile power-combining techniques upon high-power, high-millimeter-wave frequency applications. Much additional investigation is required before turnstile power-combining performance can be fully assessed. Current and future plans addressing critical performance parameters such as bandwidth, interstage isolation, etc. also are summarized in the final sections.

THEORY OF CUBIC TURNSTILE OPERATION

Outlined here is the use of symmetry analysis (Reference 1) in determining the S-matrix of a junction exhibiting various degrees of symmetry. Examples are included to illustrate the analysis method and to demonstrate salient operational characteristics of turnstile power combiners. Appendix A contains a description and listing of a computer program that implements the symmetry analysis method to determine the scattering matrix for a six-port cubic turnstile junction (spherical combining region) with rectangular waveguide port orientations selected arbitrarily from the vertical, horizontal, or ±45-degree planes. Appendix A also contains detailed instructions for using the computer program as well as more information on specifying port orientations and port polarities. The final segments of this section make use of the S-matrix to illustrate reflection- and transmission-mode turnstile power combining.

The procedure for using symmetry analysis to determine the S-matrix of a junction can be broken into four steps:

1. The ports of the junction are labeled and assigned a sense.
2. Geometrical symmetries of the junction are determined.
3. The constraints on the S-matrix caused by each symmetry are calculated.
4. The constraints caused by all of the symmetries are combined to form the overall S-matrix of the junction.

The first step of symmetry analysis is to label the ports of the junction. Each port must be assigned both an index number and a polarity indicator. Assignment of port index numbers is arbitrary. However, for purposes of this report, the input port will be designated as port 1 and the opposite port (sometimes the output port) will be designated as port 2. Ports 3 through 6 are assigned to ports normally connected to active devices when the junction is used as a power combiner. Figure 6 illustrates a typical numbering scheme used in describing cubic turnstile power combiners.

In addition to an index number, each port must be assigned a polarity indicator. The polarity indicator designates which direction will be ascribed the direction of positive voltage at the external port/junction interface. For rectangular waveguides operating in the TE_{10} mode, the E-field (and hence voltage) vector is perpendicular to the long axis of the waveguide cross section. Which direction is "up" (or positive) must be defined by the user before a meaningful S-matrix can be determined. Assignment of port polarities is particularly important when comparing junction scattering parameters measured on a network analyzer to those predicted by symmetry analysis. For measured S-parameter data, the "through-path" calibration of the network analyzer defines the positive polarities at the input and output measurement ports. For example, if a waveguide junction is inserted in the network analyzer transmission path for measurement, it is desirable that the positive orientations defined by the network analyzer calibration match orientations defined by the user at the cube-network analyzer interface. However, as Figures 7 and 8 illustrate, polarity agreement is not always possible. Figure 7 illustrates the set of polarities chosen to define the cubic turnstile S-matrix in this report. Figures 8(a) and (b) demonstrate polarity choice considerations for measuring junction transmission- and reflection-scattering parameters, respectively. Notice that at the interface between the twists and the cube, the defined polarities do not match. To maintain the consistency of measured phase data, each mismatch in polarity must be accounted for by a 180-degree phase reversal in the reflection or transmission measurement being considered. For example, in Figure 8(a) during a transmission measurement, a wave from the network analyzer (twist) experiences a defined switch in polarity as it enters the cube at port 3. Likewise, the transmitted wave also experiences a similar switch when reentering the output port (twist) from the cube. In this case, the measured phase would match the predicted value, since an even number of phase reversals were encountered. (If an odd number of phase reversals were encountered, the measured phasc would have to be inverted to match the predicted value.)

Notice that for reflection measurement of Figure 8(b) the incident and reflected waves traverse identical paths such that the polarities of the incident and reflected waves are always either both in agreement or both in opposition at the interface between the twist and the cube. Therefore, the reflection-scattering parameter always remains unaffected by choice of positive polarity in either the network analyzer or the waveguide junction.

If one is careful and consistent in defining port polarities of both the network analyzer and waveguide junction, the phase properties of the waveguide junctions can be measured and do indeed match those predicted by symmetry analysis. As will be shown later, the phase behavior of waveguide turnstile junctions is critical for successful power combining.

The second step in determining the constraints imposed by symmetry upon the S-matrix of the junction is to determine which symmetries are associated with a particular junction and port configuration. A symmetry operator can be thought of as a linear operation that moves (via rotations, mirrors, etc.) the boundaries of the junction and its ports to new locations in space. The junction is said to possess a particular symmetry if after a symmetry operation the boundaries of the transformed junction are coincident with the boundaries of the original junction. In addition, the port boundaries after transformation must be coincident with the original port locations. However, if the ports are ordered and assigned polarities prior to application of the symmetry operator (as prescribed in the first step), the resulting order and polarity of the ports after transformation can be quite different from the original configuration.

In general, there is an infinite number of possible symmetry operations. However, for the cubic turnstile junction with rectangular waveguide inputs, there are fewer than 30 possible symmetries that can occur. It is a relatively easy task to operate on the junction with each of these symmetry operators and find those under which the junction remains invariant. Figure 9 illustrates three symmetry operators: 180-degree rotation about the z-axis; 180-degree rotation about x = y; and 120-degree rotation about x = y = z. Appendix B contains a complete list of 30 symmetry operators for a cubic turnstile junction.

The third step in the symmetry-analysis approach to S-matrix determination is to find the constraints imposed on the S-matrix by each symmetry operator. The symmetry operators force various elements of the S-matrix to be equal in magnitude and either equal or opposite in sign, depending upon the polarity indicators. One way of representing the effects of a symmetry operator upon the S-matrix of a junction is shown in Figure 10. In this figure, the junction is shown embedded in an external measurement circuit. This representation will prove useful in keeping track of port orderings and polarities. Figure 10(a) represents the junction before application of the symmetry operator. The Aj and Bj refer to the incident and reflected waves, respectively, in the external measurement circuit. The index j corresponds to the external port index. Just inside the junction, the incident and reflected waves are represented by lower case ai and bi, respectively. The index i corresponds to the internal port index. Before applying the symmetry operator, the internal and external port indices (and polarities) all match. The polarity is denoted by (+) or (-) signs located at the interface between the external measurement circuit and the internal junction circuit. All polarities are positive before transformation.

In vector notation, matching incident and exiting waves across the boundary (before transformation) requires the following:

$$Aj = \begin{bmatrix} A1 \\ A2 \\ A3 \\ A4 \\ A5 \\ A6 \end{bmatrix} = ai = \begin{bmatrix} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \end{bmatrix} \quad ; \quad Bj = \begin{bmatrix} B1 \\ B2 \\ B3 \\ B4 \\ B5 \\ B6 \end{bmatrix} = bi = \begin{bmatrix} b1 \\ b2 \\ b3 \\ b4 \\ b5 \\ b6 \end{bmatrix}$$

Figure 10(b) shows the junction and external circuit after performing the symmetry operation. In this case, the symmetry operator represented is a 180-degree rotation about the z-axis [Figure 9(a)]. The junction under consideration is the cubic turnstile junction of Figure 1. After rotation, the internal port indices of the junction no longer correspond to the port indices of the external circuit. The symmetry operation has rearranged the internal port indices (and polarities) with respect to the external circuit ports. The port rearrangement and polarity changes can be represented by a symmetry operator matrix [M]:

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Symmetry Operator Matrix. Rotation about z-axis; cubic turnstile junction symmetric configuration.

This [M] can be interpreted as indicating that, after a rotation of 180 degrees about the z-axis, ports 1 and 2 of the cubic turnstile junction remain at the same location; however, a change in polarity is experienced because of the rotation. Ports 3 and 5 change position but do not change polarity. Ports 4 and 6 also change position with no change in sign. To aid in visualizing the effects of various symmetry operators on the cubic turnstile junction, a fold-up three-dimensional model of the junction is provided in Appendix C. The model folds up into a cube with appropriate indications of indexing, coordinate-axis positioning, port polarity, and port orientations. This model should prove extremely useful in following through various symmetry operators.

The [M] can be used to determine the constraints imposed by the symmetry operator upon the S-matrix of the junction. In Figure 10(b), the external excitations A are unaffected by the symmetry operation. However, these excitations are rerouted to different internal ports because of the symmetry operation. In vector notation,

$$aj' = \begin{bmatrix} a1' \\ a2' \\ a3' \\ a4' \\ a5' \\ a6' \end{bmatrix} = [M]\,ai = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \end{bmatrix} = \begin{bmatrix} -a1 \\ -a2 \\ a5 \\ a6 \\ a3 \\ a4 \end{bmatrix}$$

Note: The superscript ' indicates that the symmetry operation has been performed. Thus, aj' is interpreted as the incident internal wave present at port j (external port index) after application of the symmetry operation.

Similarly, for the exiting waves,

$$bj' = \begin{bmatrix} b1' \\ b2' \\ b3' \\ b4' \\ b5' \\ b6' \end{bmatrix} = [M]\,bi = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \\ b4 \\ b5 \\ b6 \end{bmatrix} = \begin{bmatrix} -b1 \\ -b2 \\ b5 \\ b6 \\ b3 \\ b4 \end{bmatrix}$$

Applying the boundary conditions for incident and exiting waves after the symmetry operation requires that

$$Aj = aj' = [M]\,ai \quad \text{and} \quad Bj = bj' = [M]\,bi$$

Before transformation, the external incident and reflected waves are related by the S-matrix of the junction:

$$Bj = [Sj]\,Aj \text{ and } bi = [Si]\,ai; \text{ since } Bj = bi \text{ and } Aj = ai,\ [Sj] = [Si]$$

This is a restatement of the fact that before transformation the external and internal port indices and polarities all match; hence, the internal and external S-matrices are equivalent.

After transformation,

$$\begin{aligned} Bj' &= [Sj']\,Aj' \\ bj' &= [Sj']\,aj' \\ [M]\,bi &= [Sj']\,[M]\,ai \\ bi &= [M]^{-1}\,[Sj']\,[M]\,ai \end{aligned}$$

If the transformation is a symmetry operator, then Bj' = Bj, Aj' = Aj, and [Sj'] = [Sj]. (The ' denotes posttransformation quantities.) The waves exiting a transformed junction must be identical to those exiting from the original junction. This constraint imposes conditions on the internal S-matrix of the junction, which are a direct result of the symmetry condition.

Before symmetry operation,

$$bi = [Si]\ ai$$

After applying symmetry,

$$bi = [M]^{-1}\ [Si]\ [M]\ ai$$

since [Sj'] = [Sj] = [Si] for a symmetric junction.

Therefore, the constraints on the S-matrix of the internal junction imposed by the symmetry condition can be expressed in terms of the symmetry operator as

$$[Si] = [M]^{-1}\ [Si]\ [M]$$

The following example is included to illustrate how the above expression can be efficietly evaluated (because of special properties of the symmetry operator matrix) and the constraints on the S-matrix of the junction can be determined. The symmetry operator to be examined is the 180-degree rotation about the z-axis shown in Figure 9(a).

For this symmetry operator, the condition on the S-matrix of the junction can be written as

$$\begin{bmatrix} S11 & S12 & S13 & S14 & S15 & S16 \\ S21 & S22 & S23 & S24 & S25 & S26 \\ S31 & S32 & S33 & S34 & S35 & S36 \\ S41 & S42 & S43 & S44 & S45 & S46 \\ S51 & S52 & S53 & S54 & S55 & S56 \\ S61 & S62 & S63 & S64 & S65 & S66 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^{-1}$$

$$\times \begin{bmatrix} S11 & S12 & S13 & S14 & S15 & S16 \\ S21 & S22 & S23 & S24 & S25 & S26 \\ S31 & S32 & S33 & S34 & S35 & S36 \\ S41 & S42 & S43 & S44 & S45 & S46 \\ S51 & S52 & S53 & S54 & S55 & S56 \\ S61 & S62 & S63 & S64 & S65 & S66 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

A shorthand notation has been developed to facilitate dealing with [M]. The notation is based upon two properties of the [M]. First, all nonzero elements of the [M] are either ±1. Second, each row (or column) of the matrix has only one nonzero element. Thus, by specifying the row (or column) of each nonzero element and its accompanying sign, the symmetry matrix can be completely specified. Two shorthand representations of an [M] are possible.

The "row" representation of a symmetry matrix examines the matrix row by row and generates a column vector whose i^{th} entry specifies the column that contains the only nonzero element for that i^{th} row. The sign of the entry corresponds to the sign of the nonzero element (±1).

[M] Row representation

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 \\ -2 \\ 5 \\ 6 \\ 3 \\ 4 \end{bmatrix}$$

The "column" representation of a symmetry matrix examines the matrix column by column and results in a row vector (transpose of a column vector) whose i^{th} element indicates which row contains the nonzero element corresponding to the i^{th} column. The sign of the entry indicates the sign of the nonzero element.

[M] Column representation

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad [-1\ -2\ \ 5\ \ 6\ \ 3\ \ 4]$$

The shorthand notation greatly simplifies the computation of

$$[Si] = [M]^{-1}\ [Si]\ [M]$$

in determining the constraints imposed by symmetry on the S-matrix of the junction. By using the shorthand representation of [M]s, elementary matrix operations such as pre- and postmultiplication, matrix transpose, and matrix inverse can be determined by inspection. These elementary matrix computations using [M]s are illustrated by the following examples.

PREMULTIPLYING A MATRIX BY [M]

[M] ; Matrix (before premultiplying by [M])

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} a11 & a12 & a13 & a14 & a15 & a16 \\ a21 & a22 & a23 & a24 & a25 & a26 \\ a31 & a32 & a33 & a34 & a35 & a36 \\ a41 & a42 & a43 & a44 & a45 & a46 \\ a51 & a52 & a53 & a54 & a55 & a56 \\ a61 & a62 & a63 & a64 & a65 & a66 \end{bmatrix} =$$

Matrix (after premultiplying by [M]) ; Row representation of symmetry operator

$$\begin{bmatrix} -a11 & -a12 & -a13 & -a14 & -a15 & -a16 \\ -a21 & -a22 & -a23 & -a24 & -a25 & -a26 \\ a51 & a52 & a53 & a54 & a55 & a56 \\ a61 & a62 & a63 & a64 & a65 & a66 \\ a31 & a32 & a33 & a34 & a35 & a36 \\ a41 & a42 & a43 & a44 & a45 & a46 \end{bmatrix} \qquad \begin{bmatrix} -1 \\ -2 \\ 5 \\ 6 \\ 3 \\ 4 \end{bmatrix}$$

Premultiplying a matrix by a symmetry operator rearranges the rows of the matrix (and multiplies specific rows by -1). The final order (and signs) of the rows are specified by the row representation of the [M].

POSTMULTIPLYING A MATRIX BY [M]

Matrix (before postmultiplying by [M]) ; [M]

$$\begin{bmatrix} a11 & a12 & a13 & a14 & a15 & a16 \\ a21 & a22 & a23 & a24 & a25 & a26 \\ a31 & a32 & a33 & a34 & a35 & a36 \\ a41 & a42 & a43 & a44 & a45 & a46 \\ a51 & a52 & a53 & a54 & a55 & a56 \\ a61 & a62 & a63 & a64 & a65 & a66 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} =$$

Matrix (after postmultiplying by [M])	Symmetry operator column representation
$$\begin{bmatrix} -a11 & -a12 & a15 & a16 & a13 & a14 \\ -a21 & -a22 & a25 & a26 & a23 & a24 \\ -a31 & -a32 & a35 & a36 & a33 & a34 \\ -a41 & -a42 & a45 & a46 & a43 & a44 \\ -a51 & -a52 & a55 & a56 & a53 & a54 \\ -a61 & -a62 & a65 & a66 & a63 & a64 \end{bmatrix}$$	[-1 -2 5 6 3 4]

Postmultiplying a matrix by a symmetry operator rearranges the columns of the matrix (and multiplies certain columns by -1). The final order (and signs) of the columns are specified by the column representation of the [M].

COMPUTING THE TRANSPOSE OF [M]

[M] 120-degree rotation about x=y=z	Row representation of symmetry operator	Column representation of symmetry operator
$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$	$$\begin{bmatrix} 3 \\ 5 \\ 6 \\ 2 \\ 4 \\ 1 \end{bmatrix}$$	[6 4 1 5 2 3]

Transpose of [M] 120-degree rotation about x=y=z	Row representation of symmetry operator transposed	Column representation of symmetry operator transposed
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$	$$\begin{bmatrix} 6 \\ 4 \\ 1 \\ 5 \\ 2 \\ 3 \end{bmatrix}$$	[3 5 6 2 4 1]

In shorthand notation, the transpose of [M] can be found by simply interchanging the row and column representations.

COMPUTING THE INVERSE OF [M]

By definition, a matrix postmultiplied by its inverse must equal the identity matrix. If the matrix is a symmetry operator, each row (column) of the matrix contains only a single nonzero element. As a given row consecutively multiplies each column of its inverse matrix during matrix multiplication, nonzero products can arise only along the diagonal. Therefore, each column of the inverse matrix must be identical to the corresponding row of the original matrix. The inverse of [M] is simply its transpose. To verify this, the previous 120-degree rotation about x=y=z will be postmultiplied by its transpose.

[M], 120-degree rotation about x=y=z		Transpose of symmetry operator		Identity matrix

$$\begin{bmatrix} 0&0&1&0&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \\ 0&1&0&0&0&0 \\ 0&0&0&1&0&0 \\ 1&0&0&0&0&0 \end{bmatrix} \times \begin{bmatrix} 0&0&0&0&0&1 \\ 0&0&0&1&0&0 \\ 1&0&0&0&0&0 \\ 0&0&0&0&1&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \end{bmatrix} = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{bmatrix}$$

Matrix row representation	Matrix column representation	Inverse row representation	Inverse column representation
$\begin{bmatrix} 3 \\ 5 \\ 6 \\ 2 \\ 4 \\ 1 \end{bmatrix}$	[6 4 1 5 2 3]	$\begin{bmatrix} 6 \\ 4 \\ 1 \\ 5 \\ 2 \\ 3 \end{bmatrix}$	[3 5 6 2 4 1]

Equipped with shorthand notation and knowledge of the special properties of [M]s, the computation of the constraints imposed by symmetry operators upon the S-matrix of the junction given by

$$[Si] = [M]^{-1} [Si] [M]$$

can be determined. To illustrate the method, the constraints upon the turnstile junction imposed by a 120-degree rotation about x=y=z will be examined.

[M], 120-degree rotation about x=y=z	Row representation of symmetry operator	Column representation of symmetry operator
$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 5 \\ 6 \\ 2 \\ 4 \\ 1 \end{bmatrix}$	[6 4 1 5 2 3]

Inverse of symmetry operator matrix 120-degree rotation about x=y=z	Row representation of inverse	Column representation of inverse
$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 6 \\ 4 \\ 1 \\ 5 \\ 2 \\ 3 \end{bmatrix}$	[3 5 6 2 4 1]

In shorthand notation, $[M]^{-1}$ [Si] [M] is written as

Row representation of inverse of [M]		Untransformed S-matrix
$\begin{bmatrix} 6 \\ 4 \\ 1 \\ 5 \\ 2 \\ 3 \end{bmatrix}$	x	$\begin{bmatrix} S11 & S12 & S13 & S14 & S15 & S16 \\ S21 & S22 & S23 & S24 & S25 & S26 \\ S31 & S32 & S33 & S34 & S35 & S36 \\ S41 & S42 & S43 & S44 & S45 & S46 \\ S51 & S52 & S53 & S54 & S55 & S56 \\ S61 & S62 & S63 & S64 & S65 & S66 \end{bmatrix}$

Columm
representation
of [M]

x [6 4 1 5 2 3]

After performing premultiplication,

S-matrix after premultiplying by inverse of [M] | Column representation of [M]

$$\begin{bmatrix} S61 & S62 & S63 & S64 & S65 & S66 \\ S41 & S42 & S43 & S44 & S45 & S46 \\ S11 & S12 & S13 & S14 & S15 & S16 \\ S51 & S52 & S53 & S54 & S55 & S56 \\ S21 & S22 & S23 & S24 & S25 & S26 \\ S31 & S32 & S33 & S34 & S35 & S36 \end{bmatrix} \times [6\ 4\ 1\ 5\ 2\ 3]$$

After performing postmultiplication,

[M] [Si] [M] = [Si]

$$\begin{bmatrix} S66 & S64 & S61 & S65 & S62 & S63 \\ S46 & S44 & S41 & S45 & S42 & S43 \\ S16 & S14 & S11 & S15 & S12 & S13 \\ S56 & S54 & S51 & S55 & S52 & S53 \\ S26 & S24 & S21 & S25 & S22 & S23 \\ S36 & S34 & S31 & S35 & S32 & S33 \end{bmatrix} = \begin{bmatrix} S11 & S12 & S13 & S14 & S15 & S16 \\ S21 & S22 & S23 & S24 & S25 & S26 \\ S31 & S32 & S33 & S34 & S35 & S36 \\ S41 & S42 & S43 & S44 & S45 & S46 \\ S51 & S52 & S53 & S54 & S55 & S56 \\ S61 & S62 & S63 & S64 & S65 & S66 \end{bmatrix}$$

From a single symmetry operator (120-degree rotation about the x=y=z axis), the following constraints are imposed on the turnstile junction S-matrix:

S66 = S11 = S33
S44 = S22 = S55
S64 = S12 = S35
S61 = S13 = S36
S65 = S14 = S32
S62 = S15 = S34

S63 = S16 = S31
S46 = S21 = S53
S41 = S23 = S56
S45 = S24 = S52
S42 = S25 = S54
S43 = S26 = S51

Each of the 30 symmetries listed in Appendix B is tested for applicability to a desired junction configuration. Applicability of a symmetry is established by defining a polarity vector at each port of the junction configuration under consideration. These port vectors are then transformed in a manner prescribed by the symmetry operator. If all of the transformed port vectors are coincident with the original port vectors (even though the two vectors could be pointing in opposite directions), the symmetry operator is ascribed to that junction configuration. The constraints on the S-matrix for that junction imposed by each applicable symmetry operator are determined using shorthand notation and computing $[M]^{-1}$ [Si] [M] for each symmetry operator.

The final step in the symmetry analysis procedure is to combine all of the constraints from all applicable symmetry operators. The constraints can be arranged in equality strings. For the cubic turnstile junction with port polarities defined as in Figure 11, the following equality strings are generated:

S11 = S22 = S33 = S44 = S55 = S66

S13 = S23 = -S25 = -S15 = -S14 = S36 = S24 = S45 = S34 = S16 = S65 = S42

= S43 = S32 = S31 = -S26 = -S41 = -S51 = -S52 = S54 = S56 = S61

= -S62 = S63

S12 = S21 = -S64 = S53 = S35 = S64 = S46 = -S35 = -S53 = -S21 = -S12

The final equality string contains terms of the form Sij = -Sij, which can be satisified only if *Sij* is *identically zero.*

The constraints imposed by symmetry on the cubic turnstile junction can be written in matrix form as:

Symmetry contraints
on cubic turnstile
junction

$$\begin{bmatrix} g & 0 & h & -h & -h & h \\ 0 & g & h & h & -h & -h \\ h & h & g & h & 0 & h \\ -h & h & h & g & h & 0 \\ -h & -h & 0 & h & g & h \\ h & -h & h & 0 & h & g \end{bmatrix}$$

The terms of the first equality string have been replaced by "g", and the terms of the second equality string have been replaced by ±"h".

Appendix A contains a computer program for generating the symmetry-constraint matrix for any junction configuration in which the waveguide ports are aligned vertically, horizontally, or diagonally (or any combination) with respect to the cube faces. Symmetry-contraint matrices for Eisenhart's turnstile, Purcell's junction, and Ginzton's six-port are also presented in Appendix A.

If the cubic turnstile junction is matched (g = 0) and lossless (magnitude of "h" = 1/2), then the S-matrix is given by

S-matrix of matched, lossless, cubic turnstile junction

$$\begin{bmatrix} 0 & 0 & 1/2 & -1/2 & -1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 & -1/2 & -1/2 \\ 1/2 & 1/2 & 0 & 1/2 & 0 & 1/2 \\ -1/2 & 1/2 & 1/2 & 0 & 1/2 & 0 \\ -1/2 & -1/2 & 0 & 1/2 & 0 & 1/2 \\ 1/2 & -1/2 & 1/2 & 0 & 1/2 & 0 \end{bmatrix}$$

The matched, lossless, turnstile junction has several useful properties that can be exploited for power-combining applications. The next section will examine the transmission- and reflection-mode power-combining properties of the matched, lossless, cubic turnstile junction.

POWER-COMBINING PROPERTIES OF MATCHED, LOSSLESS, CUBIC TURNSTILE JUNCTION (TRANSMISSION MODE)

The relationship between incident (ai) and emanating (bi) waves of the matched, lossless, turnstile junction is given by

$$\begin{bmatrix} b1 \\ b2 \\ b3 \\ b4 \\ b5 \\ b6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1/2 & -1/2 & -1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 & -1/2 & -1/2 \\ 1/2 & 1/2 & 0 & 1/2 & 0 & 1/2 \\ -1/2 & 1/2 & 1/2 & 0 & 1/2 & 0 \\ -1/2 & -1/2 & 0 & 1/2 & 0 & 1/2 \\ 1/2 & -1/2 & 1/2 & 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \end{bmatrix}$$

The port-numbering scheme of Figure 6 will be adopted for use in this section. The input port is port 1; the output port is port 2. Ports 3 through 6 will be designated as device ports; in power-combining applications, they will be terminated with active reflection-type devices such as IMPATT diodes, Gunn diodes, or magnetrons. Ports 3 and 5 are isolated from each other as are ports 4 and 6 (since S35 = S53 = S46 = S64 = 0). The input and output ports are also isolated from each other (S12 = S21 = 0). The zeros along the diagonal assume that the turnstile junction is matched at all ports.

To demonstrate the transmission mode of power combining, the following boundary conditions on the incident and emanating waves will be imposed:

$a1 = 1 \;;\; a2 = 0$

An incident wave is present at the input port, and the output port is assumed terminated in a perfect termination (i.e., no reflected wave).

At ports 3 and 5, the reflection coefficient with respect to the port terminator (i.e., the network or device terminating a given port) at each port will be required to be equal and will be represented by the complex constant c.

$a3/b3 = c \;;\; a5/b5 = c$

Similarly, at ports 4 and 6, the terminator reflection coefficients are required to be equal but opposite in sign to the terminator reflection coefficient at ports 3 and 5. These are the conditions that produce transmission-mode power combining. A further multiplicative factor k also will be introduced in the expression for the reflection coefficient at ports 3 and 5. This factor will be used to examine the effects of amplitude and phase imbalance between the active device pairs. [The reflection coefficients within each pair (3 and 5; 4 and 6) will still be assumed equal, but a phase or amplitude imbalance may exist between pairs.]

$a4/b4 = -kc \;;\; a6/b6 = -kc$

Substituting these wave boundary conditions in $b_i = [S]a_i$ results in the system of equations:

$$\begin{bmatrix} 2 & 0 & -c & -kc & c & kc \\ 0 & 2 & -c & kc & c & -kc \\ 0 & 0 & 2 & kc & 0 & kc \\ 0 & 0 & -c & 2 & -c & 0 \\ 0 & 0 & 0 & kc & 2 & kc \\ 0 & 0 & -c & 0 & -c & 2 \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \\ b4 \\ b5 \\ b6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ a1 \\ -a1 \\ -a1 \\ a1 \end{bmatrix}$$

A solution of the system of equations is given by

$$\begin{bmatrix} b1 \\ b2 \\ b3 \\ b4 \\ b5 \\ b6 \end{bmatrix} = \begin{bmatrix} c(1-k)\,a1/2 \\ c(1+k)\,a1/2 \\ a1/2 \\ -a1/2 \\ -a1/2 \\ a1/2 \end{bmatrix}$$

A simple illustration of transmission-mode power combining is observed by setting k = 1 (no amplitude or phase imbalance) and c = -1 (perfect shorts placed on ports 3 and 5; perfect opens or quarter-wavelength offset shorts placed on ports 4 and 6). The solution for k = 1 and c = -1 is given by

Shorts on ports 3 and 5;
quarter-wavelength
offsets on ports 4
and 6; no amplitude
or phase imbalance.
k = 1, c = -1.

$$\begin{bmatrix} b1 \\ b2 \\ b3 \\ b4 \\ b5 \\ b6 \end{bmatrix} = \begin{bmatrix} 0 \\ -a1 \\ a1/2 \\ -a1/2 \\ -a1/2 \\ a1/2 \end{bmatrix}$$

Under these conditions (k = 1 and c = -1), no power emanates from port 1 (b1 = 0) and all incident power appears at the output port (b2 = -a1). The incident power is split evenly between the four device ports, reflects from the port terminators, and is totally recombined at the output.

If the positions of the shorts and offset shorts are reversed (c = 1 and k = 1), the solution becomes

Shorts on ports 4 and 6;
quarter-wavelength
offsets on ports 3
and 5; no amplitude
or phase imbalance.
k = 1, c = 1.

$$\begin{bmatrix} b1 \\ b2 \\ b3 \\ b4 \\ b5 \\ b6 \end{bmatrix} = \begin{bmatrix} 0 \\ a1 \\ a1/2 \\ -a1/2 \\ -a1/2 \\ a1/2 \end{bmatrix}$$

Again the wave is recombined completely at the output port but differs in sign from the previous example. Thus, reversing the positions of the shorted and quarter-wavelength-offset shorted pairs reverses the sign of b2; however, complete transmission still results.

Before considering the effects of terminator-pair phase and amplitude imbalance upon transmission-mode power-combining performance, the above results will be used to examine reflection-mode power combining.

POWER-COMBINING PROPERTIES OF MATCHED, LOSSLESS, CUBIC TURNSTILE JUNCTION (REFLECTION MODE)

Reflection-mode power combining will be examined by setting c = -1 and k = -1. With this choice of k and c, all port terminators are identical and correspond to perfect short circuits:

$$a3/b3 = a5/b5 = a4/b4 = a6/b6 = -1$$

Under these conditions, a solution for the emanating waves is given by

$$\begin{bmatrix} b1 \\ b2 \\ b3 \\ b4 \\ b5 \\ b6 \end{bmatrix} = \begin{bmatrix} a1 \\ 0 \\ a1/2 \\ -a1/2 \\ -a1/2 \\ a1/2 \end{bmatrix}$$

In reflection-mode power combining, the incident wave recombines completely at the input port and no power is transmitted through to the output port. If all port terminators are changed to quarter-wavelength offset shorts (c = 1 and k = -1), then the sign of the wave emanating from port 1 is reversed.

EFFECTS OF TERMINATOR-PAIR AMPLITUDE AND PHASE IMBALANCE UPON TRANSMISSION- AND REFLECTION-MODE POWER COMBINING

To examine the effects of amplitude imbalance between the pair of port terminators on ports 3 and 5 and the pair on ports 4 and 6, the coefficient c will be set to 1 and k will vary from 1 (no pair-amplitude imbalance) to 0.1 (reflection coefficient of one pair is 10% that of the other). With k = 1, all power is transmitted and none is reflected. The transmitted and reflected wave amplitudes are given by

$$b1 = (1 - k)a1/2 \quad ; \quad b2 = (1 + k)a1/2 \; (c = 1)$$

Figure 12 shows a plot of the return loss (RL) at port 1 given by

$$\text{RL @ port 1} = 20 \log (b1/a1) = 20 \log [(1 - k)/2]$$

for varying values of the amplitude imbalance parameter k.

To achieve a RL of better than -20 decibels, the pairs of port terminators must match within 80%.

Figure 12 also plots the transmission loss (TL) encountered when the port terminators on the turnstile junction are mismatched in amplitude. The expression for TL is given by

$$\text{TL from port 1 to port 2} = 20 \log (b2/a1) = 20 \log [(1 + k)/2]$$

for varying values of the amplitude imbalance parameter k.

If k is set to zero, one pair of ports is terminated by perfect shorts and the other pair by perfect terminations (i.e., all incident power is absorbed by the termination and none is reflected back to the turnstile junction). Under these conditions, one-quarter of the incident power is reflected back from port 1, one-quarter of the power is transmitted through to port 2, and two-quarters are absorbed in the perfect terminations.

To examine the effects of phase imbalance between pairs of port terminators, c will again be set to unity and k will be set as follows:

$$k = \exp j(\theta)$$

The phase imbalance θ will be allowed to vary from 0 to 90 degrees. Under these conditions, the transmitted and reflected wave amplitudes are given by

$$b1 = [1 - \exp j(\theta)]a1/2 \quad ; \quad b1 = [1 + \exp j(\theta)]a1/2$$

Figure 13 shows the RL and TL performance of the turnstile junction as θ is varied from 0 to 90 degrees.

The results of Figure 13 indicate that up to 12 degrees of phase imbalance (between pairs of port terminators) can exist and still maintain an input RL of less than -20 decibels. At 90 degrees of imbalance, one-half of the incident power is reflected back from port 1 and one-half is transmitted through to port 2.

In the previous development, the reflection coefficient boundary conditions were specified such that opposite port terminators were identical; however, the two pairs of port terminators were allowed to differ in amplitude and phase. If each device port terminator is specified separately (not paired with its opposite port), the following reflection coefficient boundary conditions result:

$$a_3/b_3 = k_3c \quad ; \quad a_4/b_4 = -k_4c \quad ; \quad a_5/b_5 = k_5c \quad ; \quad a_6/b_6 = -k_6c$$

If $a_2 = 0$, the following system of equations relating the emanating waves from the matched, lossless, turnstile junction is generated.

$$\begin{bmatrix} 2 & 0 & -k_3c & -k_4c & k_5c & k_6c \\ 0 & 2 & -k_3c & k_4c & k_5c & -k_6c \\ 0 & 0 & 2 & k_4c & 0 & k_6c \\ 0 & 0 & -k_3c & 2 & -k_5c & 0 \\ 0 & 0 & 0 & k_4c & 2 & k_6c \\ 0 & 0 & -k_3c & 0 & -k_5c & 2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ a_1 \\ -a_1 \\ -a_1 \\ a_1 \end{bmatrix}$$

The solution for this system of equations is given by

$$b_1 = \frac{c[k_3-k_4+k_5-k_6]+c^2[k_3k_4-k_3k_6-k_4k_5+k_5k_6]}{4+c^2[k_3k_4+k_3k_6+k_4k_5+k_5k_6]}\,a_1$$

$$+\frac{c^3[k_3k_4k_5-k_3k_4k_6+k_3k_5k_6-k_4k_5k_6]}{4+c^2[k_3k_4+k_3k_6+k_4k_5+k_5k_6]}\,a_1$$

$$b_3 = \frac{c\{k_3+k_4+k_5+k_6]+c[k_3k_4k_5+k_3k_4k_6+k_3k_5k_6+k_4k_5k_6]}{4+c^2[k_3k_4+k_3k_6+k_4k_5+k_5k_6}\,a_1$$

$$b_3 = \frac{2+c[k_4-k_6]+c^2[k_4k_5+k_5k_6]}{4+c^2[k_3k_4+k_3k_6+k_4k_5+k_5k_6]}\,a_1$$

$$b_4 = \frac{2-c[k_3-k_5]+c^2[k_3k_6+k_5k_6]}{4+c^2[k_3k_4+k_3k_6+k_4k_5+k_5k_6]}\,a_1$$

$$b_5 = \frac{2-c[k_4-k_6]+c^2[k_3k_4+k_3k_6]}{4+c^2[k_3k_4+k_3k_6+k_4k_5+k_5k_6]}\,a_1$$

$$b_6 = \frac{2+c[k_3-k_5]+c^2[k_3k_4+k_4k_5]}{4+c^2[k_3k_4+k_3k_6+k_4k_5+k_5k_6]}\,a_1$$

If $k_3 = k_4 = k_5 = k_6 = k$ (transmission-mode conditions), the above results reduce to

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} 0 \\ ck\, a_1 \\ a_1/2 \\ -a_1/2 \\ -a_1/2 \\ a_1/2 \end{bmatrix}$$

If $k_3 = k_5 = k$; $k_4 = k_6 = -k$ (reflection-mode conditions), the above results reduce to

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} ck\, a_1 \\ 0 \\ a_1/2 \\ -a_1/2 \\ -a_1/2 \\ a_1/2 \end{bmatrix}$$

Further analysis is needed to assess the effects of device-to-device phase and amplitude imbalance upon turnstile combining performance. Future work will address this task.

EXPERIMENTAL TURNSTILE HARDWARE

This section describes three iterations of experimental turnstile hardware produced during transmission-mode power-combining investigations. The details of mechanical construction and design will be presented for each iteration. Also included in this section are discussions of S-parameter performance and passive two-port transmission-mode performance for each design.

The first turnstile design is shown in Figure 14. A unique feature of this design is that the orientation of each of the six waveguide ports is rotationally adjustable. Each port can be oriented vertically, horizontally, or at increments of ±45 degrees. Using this design, a large number of turnstile port configurations can be evaluated with a single piece of hardware. To maintain symmetry, the axis of each input waveguide is coincident with the axis of rotation of the individual input ports and mutually intersect at the center of a spherical combining region. The inner-end surface of each waveguide is spherically concave to match the radius of curvature of the central combining region.

Four symmetry configurations were selected for evaluation using the hardware of Figure 14. Figure 15(a) shows the configuration of the planar-device-port turnstile described in Reference 2. Figure 15(b) is the configuration described by Purcell in Reference 3. Figure 15(c) is a configuration similar to the cubic-symmetric turnstile except that ports 4 and 6 are rotated 90 degrees. Figure 15(d) is the cubic turnstile junction. Of primary interest in the initial turnstile evaluations was the degree to which the various configurations exhibited input matching at all ports. Figure 16 contains the RL versus frequency data for each of the four configurations. The planar-device-port configuration [Figure 16(a)] exhibited matching at only the orthogonally oriented ports; Purcell's junction exhibited some degree of match at each port. Configuration #3 [Figure 16(c)] exhibited a good match at four ports but remained unmatched at the rotated ports. The symmetric-cubic-junction configuration exhibited matching at all ports and, therefore, was selected for further evaluation.

Figure 17 shows the phase of the reflection coefficient at each port for the symmetric turnstile junction. The phase uniformity is marginal (but will improve in later iterations). Possible factors contributing to nonuniformity include (1) cracks introduced on combining sphere surface because of rotating waveguide assembly, (2) relatively long line length from cube face to combining spheres, and (3) termination nonuniformity used in performing S-parameter measurements.

The adjacent port-coupling coefficients for the symmetric turnstile junction are shown in Figure 18. The theoretical value for adjacent port coupling is one-quarter, or -6 decibels. Typical measured values in Figure 18 are -7 to -8 decibels. The additional measured coupling loss could be attributed to joints introduced in the combining region to allow rotational adjustment. A rapid falloff in adjacent port coupling is observed at frequencies above 35.5 gigahertz. The mechanism causing this upper frequency limit in adjacent port coupling has yet to be identified. (Subsequent design iterations indicate that the position of the upper frequency limit is linked to the diameter of the spherical combining region.) Figure 18(a) shows the amplitude and phase of the scattering parameters S13, S14, S15, and S16 (coupling between the input port and each device port). Figure 18(b) depicts S23, S24, S25, and S26 (coupling between the output port and each device port). Figure 18(c) contains the adjacent device port couplings: S34, S45, S56, and S63.

The isolation or cross-port coupling coefficients of the symmetric turnstile junction are shown in Figure 19. Below 35 gigahertz, cross-port isolation values between -25 and -35 decibels are observed. Inherent cross-port isolation between device ports is an important attribute of the symmetric turnstile junction not present in the planar-turnstile junction. Later design iterations exhibit improved cross-port isolation.

The S-parameters in Figures 16 through 19 were measured on a network analyzer. The reflection coefficients were measured with all unused ports terminated by waveguide loads. Coupling and isolation coefficients were computed from direct transmission measurements of a cascaded combination of appropriate waveguide twists and the cubic turnstile junction. The twists were characterized separately and de-embedded from the final data. (In subsequent designs, the coupling coefficients were measured using an unterminating reflection-measurement technique. The latter technique utilizes a series of reflection measurements performed at a selected input port, while the output port is sequentially terminated by a short and two known offset short reflection standards. Unused ports are terminated by waveguide loads.)

This measurement technique is used also to characterize the turnstile junction in a passive power-combining configuration. In the passive configuration, device ports are terminated by shorts (reflection mode) or by pairs of shorts and quarter-wavelength offset shorts (transmission mode). Figure 20 presents the measured S-parameters of the symmetric turnstile junction of Figure 14 in a passive transmission-mode configuration. For clarity, only S11 and S21 are shown. Figure 21 is a corresponding plot of the transfer efficiency based upon the measured S-parameter data. The data indicate that both an upper (33 gigahertz) and lower (27.5 gigahertz) frequency limit of efficient transmission-mode power combining exists for this structure. These limits are not yet well understood; they probably are the result of higher order modes present in the spherical combining region.

The theoretical value of efficiency for the ideal turnstile combiner is 100%, since the structure contains no lossy elements (conductor losses are neglected in the ideal turnstile). The measured value of transfer efficiency in the range of 27 to 33 gigahertz is between 80 and 90%. Later turnstile designs of overall smaller size and fixed port orientations exhibited efficiencies closer to 100%; consequently, the observed reduction in the efficiency of the hardware of Figure 14 may be attributable to longer line lengths or joints in the combining region.

Another important parameter that affects the utility of the turnstile transmission-mode combiner is the input reflection coefficient in the transmission-mode configuration (i.e., S11 in Figure 20). This parameter is of particular importance when cascading successive turnstile-combiner stages without interstage circulators. Figure 20 indicates that input match occurs at four frequencies (28.2, 29, 31, and 32.7 gigahertz). The most pronounced match condition occurs at 31 gigahertz, which is near the center of the efficiency bandwidth measured for this structure.

The objective of the next design iteration was to move this central match point closer to the desired operating frequency of 35 gigahertz. This was accomplished by reducing the diameter of the spherical combining region. In addition, for improved efficiency, the overall size of the cube was reduced to eliminate any additional line losses. Finally, to reduce further losses and eliminate possible reflections arising from joints in the central combining region, the second design was fabricated using a two-piece construction technique.

Using this technique, the turnstile body was machined from a single brass block. The spherical combining region was then formed at the center of the block by using a ball-end mill entering from one face. Next, on the remaining five faces, waveguides were broached directly through to the central combining sphere. The sixth waveguide port was broached in a separate cylindrical plug. The end of this plug was spherically shaped to match the central combining region and then soft-soldered in place at the proper orientation to complete the second turnstile design. (No provisions for cooling were included in this design, as only passive measurements were performed using this hardware).

Figures 22 through 24 show the measured S-parameter performance of the second turnstile design. Figure 22 shows the magnitude and phase of the reflection coefficient at each port. The data indicate that matching is improved at the higher frequencies. The port-to-port uniformity is very good. Figure 23 contains the adjacent port-coupling data. The adjacent port-coupling data are approaching the -6-decibel theoretical limit at the higher frequencies. The coupling uniformity is good. The bulk of the coupling data falls within a ±0.25-decibel band. The two exceptions (S52 and S43) lie within 0.75 decibel and may have arisen because of measurement error. These values have not been verified. Figure 24 shows the cross-port isolation achieved for design #2. The data indicate a cross-port isolation of greater than 35 decibels across the measured frequency band (33 to 37 gigahertz).

Figures 25 and 26 contain the passive transmission-mode data for design #2. The passive two-port S-parameters are shown in Figure 25. The reduced combining region diameter and reduced cube size of design #2 resulted in a much broader frequency response. In addition, only two S11 match points are observed for design #2 (versus four for design #1). The upper match point occurs at 35.8 gigahertz, which is slightly above our desired operating frequency of 35 gigahertz. Figure 26 is the associated transfer efficiency for design #2. A lower frequency limit in efficiency occurs at 31 gigahertz. An upper limit was not discernible from the data available. A transfer efficiency greater than 90% is measured for design #2.

The final design iteration described in this section is shown in Figure 27. The objective of design #3 was to lower frequency of the S11 match point (in Figure 25) by slightly increasing the diameter of the combining junction.

Figures 28 through 30 contain the S-parameter data for design #3. Figure 28 shows the magnitude and phase of the reflection coefficient at each port. The data indicate that turnstile #3 exhibits a good input match at all ports over a 3-gigahertz bandwidth. In addition, port-to-port uniformity of reflection phase and amplitude remains good. The adjacent port couplings for design #3 are shown in Figure 29. Excellent port-to-port uniformity is achieved, and the magnitude of adjacent port coupling closely approximates the theoretical value of -6 decibels. The measured cross-port isolation of design #3 is shown in Figure 30. A cross-port isolation value of 35 decibels or greater is observed over a 4-gigahertz bandwidth.

Figure 31 shows S11 data for the passive transmission-mode configuration of design #3. The input port remains matched over a 2-gigahertz bandwidth centered at 34 gigahertz. The passive two-port S-parameters are strongly affected by the length of intervening waveguide between the combining sphere and the device reference plane (i.e., position of reference short). Evidence of this strong dependence can be seen in the S11 passive transmission-mode data of Figure 32. These data were taken with an additional 0.150-inch length added to the device ports (ports 3 through 6). With the additional 0.150-inch spacers, the observed S11-match bandwidth increased to 4.5 gigahertz. Transfer-efficiency data of design #3 are not yet available, but excellent performance is expected.

Integral cooling channels were machined in the central block of design #3. Figure 27 shows the connecting inlets for these channels at opposing corners of the input face. Turnstile design #3 was further integrated with IMPATT diode oscillator modules to evaluate the active transmission-mode performance of the symmetric turnstile design. The following section describes the IMPATT oscillator design, turnstile integration, and active transmission-mode evaluation. Table 1 is provided as a data summary for the three turnstile designs presented in this section.

TABLE 1. Data Summary for Three Turnstile Design Iterations.

Design	Junction diameter, in.	Cube length, in.	S11-Match frequencies, GHz	Efficiency bandwidth, GHz	S11-Match bandwidth, GHz
1	0.500	2.25	28.3 29.0 31.0[a] 32.7	5.8	0.2
2	0.400	0.997	32.0 35.8[a]	>8.0	0.3
3	0.461	1.010	30.5 34.3[a]	data not avail.	2.3

[a] Primary match frequencies.

IMPATT OSCILLATOR MODULE DESIGN AND TURNSTILE INTEGRATION

Here we discuss the design and integration of the IMPATT oscillator modules used to demonstrate transmission-mode power combining using the symmetric turnstile combiner.

OSCILLATOR DESIGN

The oscillator modules used in this demonstration are shown in Figure 33. The oscillator is a Kurokawa-type design and consists of a half-wavelength rectangular cavity. The cavity is magnetically coupled to a 50-ohm transmission line that is terminated at one end by a tapered 50-ohm load and matched to the IMPATT at the other end by a single-section coaxial impedance transformer. The length and impedance of the coaxial transformer are selected to match the impedance of the IMPATT diode at the desired frequency and radio-frequency (RF) injection power level. The cavity is coupled to the output waveguide via a coupling iris. The iris diameter and coaxial-line-to-sidewall separation are determined experimentally to optimize the transfer efficiency and bandwidth of the module. Opposite the iris at the other end of the cavity is an adjustable backshort. Several backshort designs were evaluated during the course of oscillator development. Designs evaluated include fixed backshorts, multisectioned barbell type, adjustable wedge backshort (Reference 4), and Johansen tuning screw type. The current design uses the tuning screw type, selected mainly for simplicity and ease of *in situ* phase adjustment. Figure 34 shows the various backshort designs. Figure 35 contains a sketch of the cavity design selected for this demonstration. Appendix D contains additional experimental design information regarding iris size and coaxial line placement.

Two aspects of oscillator design—optimization of transfer efficiency versus bandwidth and the design approach for impedance matching—will be considered in the remainder of this section.

TRANSFER EFFICIENCY OPTIMIZATION

The transfer efficiency of an oscillator is defined as the ratio of the power transferred to the output compared to the added power applied to the input from the active device. Also, it is a measure of the ability of the oscillator circuit to transfer the power generated by the device to the output port of the oscillator. The diagram of Figure 36 is useful in explaining oscillator transfer efficiency in terms of the incident and reflected waves observed at the oscillator input and output ports. (In a similar manner, an "amplifier" efficiency for injection-locked oscillators (i.e., a1 is nonzero) can be defined as the added power at the output divided by the added power at the input. The expression is much more complicated; it is not only a function of the oscillator-scattering parameters but also depends upon the reflection coefficient of the active device at the input port. Appendix E derives an expression for amplifier efficiency and provides additional details.) In the following development, the optimization of the oscillator transfer efficiency will be discussed. Accommodation of injection locking will be addressed separately by compensating the real part of the oscillator circuit impedance to account for the presence of the injection-locking signal. (For further discussions of injection-locking behavior, see the section entitled Impedance Transformer Design.)

The transfer efficiency can be expressed in terms of the ratios of these waves given by the S-parameters of the oscillator two-port circuit. In terms of S-parameters, the transfer efficiency is given by

$$\eta_{osc} = \frac{|S12|^2}{1 - |S22|^2}$$

A typical transfer-efficiency-versus-frequency plot for the oscillator modules used in the turnstile transmission-mode power-combining demonstration is shown in Figure 37. A peak transfer efficiency of 70% was achieved with a -1.0-decibel frequency bandwidth of 1.2 gigahertz. In general, a tradeoff between bandwidth and maximum efficiency is observed as the degree of coupling between the cavity and external circuits is varied. Coupling to the external circuit is accomplished via the iris and may be varied by changing the iris diameter. Coupling between the cavity and active device is accomplished via the coaxial line. This coupling can be varied by changing the size and placement of the coaxial line within the cavity. Appendix D contains additional experimental data that illustrate this tradeoff.

To conclude the discussion of transfer efficiency optimization, additional comments regarding the measurement and application of S-parameters in oscillator design are included.

The S-parameters used to generate the efficiency plot of Figure 37 were measured on a Ka-band network analyzer and were obtained using a short, 2-offset-short measurement technique. The offset shorts were composed of one-sixth- and one-third-wavelength sections of 50-ohm transmission line terminated by a short. (Note: During transfer efficiency optimization, the impedance transformers are not present. The transforming sections are designed separately after the cavity configuration has been optimized for the desired transfer efficiency, bandwidth, and resonant frequency.)

Proper use of S-parameters in oscillator design requires that careful attention be given to the selection of the physical planes of reference for the input and output ports. The reference planes selected for the oscillator S-parameter measurements are shown in Figure 38. The input reference plane is (mathematically) positioned at the plane of the iris, while the output reference plane is (mathematically) placed on the coaxial line at midcavity. As shown in Figure 38, the actual measurements are performed at input and output reference planes sufficiently removed from the iris or cavity/coaxial interface such that dominant mode propagation can be assumed.

Another important aspect of S-parameter design is proper wave reference impedance designation. It is of particular importance when designing waveguide Kurokawa-type oscillators, which transition from waveguide at the input port to coax at the output port. The reference impedance at the waveguide port is chosen to be

$$Z_{WG} = \frac{377}{\sqrt{1 - (f_c/f)^2}}$$

where f_c is the waveguide cutoff frequency. The reference impedance for the coaxial output port was selected to be 50 ohms. The emphasis upon careful distinction of port reference impedances becomes especially important when computing oscillator input impedances or when using commercial computer-aided-design (CAD) programs to design transformers. (For example, some CAD programs require that measured S-parameter input data be referenced to the same impedance at both ports.)

DESIGN OF IMPEDANCE TRANSFORMERS

Once the iris size, coaxial-line placement, and cavity length have been selected for the desired transfer efficiency bandwidth at the desired operating frequency, the transformer section can be designed. Proper transformer design requires knowledge of both the cavity impedance and the diode impedance.

The cavity impedance can be computed from the S-parameter S22 measured during the transfer efficiency optimization previously described. A convenient reference plane for the definition of the cavity impedance is chosen on the coaxial line midway between the upper and lower walls of the cavity. This "midcavity" impedance is actually computed based upon S-parameters measured in the 50-ohm coax at a sufficient distance from the cavity such that transverse electromagnetic perturbations are negligible. These S-parameters are then rotated through 50-ohm coax (assumed lossless) to the midcavity position. The midcavity impedance is then given by

$$Z_{midcavity} = Z_0 \left[\frac{1 + S_{22_{rotated}}}{1 - S_{22_{rotated}}} \right]$$

where Z_0 is the characteristic impedance of the coaxial line (50 ohms).

Figure 39 contains a Smith Chart plot of the midcavity impedance for the cavity configuration of Figure 35. The resonant behavior of the cavity structure shapes the impedance locus. At the desired operating frequency of 34.5 gigahertz, the measured midcavity impedance is 151.0 + j 65.8 ohms. The impedance locus can be changed in various ways by modifying the cavity configuration. Appendix F contains experimental data showing the effect of iris size, coaxial-line placement, and cavity length upon the midcavity impedance.

Before transformer design can begin, the diode impedance also must be determined. Diode impedance varies nonlinearly with incident power and, therefore, is more difficult to characterize. The incident power upon a diode (hence, diode impedance) is influenced by several factors such as circuit impedance, amplitude of injection signal, and mounting and packaging parasitics. These factors will be discussed briefly below.

In a free-running oscillator, the incident power is primarily established by the reflection coefficient (impedance) of the oscillator circuit. A given value of circuit reflection coefficient establishes an incident wave of particular magnitude and phase that evokes a corresponding response from the diode. If conditions are "just right," the diode will deliver maximum power to the circuit at a desired frequency. If the magnitude of the incident signal varies from the optimum value, diode output power will decrease. If the phase of the incident signal varies, the frequency of oscillation will change.

In the impedance plane (especially in close proximity to an ideal short), the magnitude of the reflection coefficient correlates strongly with the real part of the diode (or circuit) impedance, while the phase angle correlates with the reactance or imaginary part of the impedance. So, as might be expected, the real part of the diode (or circuit) impedance governs the power-related aspects of oscillator behavior, while the circuit reactance determines frequency behavior. This separation of function can be exploited in oscillator-transformer design. The real part of the circuit impedance is matched for maximum added power from the diode, while the imaginary part is matched for desired oscillation frequency.

The real part of the device (or circuit) impedance controls the power-related aspects of oscillator performance. Typical power performance versus device resistance for a Ka-band gallium-arsenide double-drift IMPATT is shown in Figure 40. A maximum added power of 2.0 watts (average) is generated at a resistance value of 1.33 ohms. Oscillation ceases for circuit resistances above 3.8 ohms. This limit corresponds to the small signal impedance of the diode.

Figure 41 shows a plot of oscillation frequency versus circuit reactance. The circuit reactance of this figure is referenced to the top surface of the diode package, as shown in Figure 38. Mounting parasitics have a significant effect upon the circuit reactance of the oscillator (frequency behavior). Other parasitic parameters, such as lead inductance, ceramic package capacitance, and mounting capacitance, also affect the circuit impedance. Appendix G discusses the package and mounting parasitics for the Ka-band IMPATT used in the turnstile oscillators. The package parasitics are a major contributor in limiting the frequency-bandwidth performance of an oscillator. Package parasitics also reduce the effective power-combining efficiency of multidiode combiners by increasing device-to-device nonuniformity.

Thus far, discussions of diode impedance have considered only free-running oscillator circuits. The incident wave upon a diode (hence, diode impedance) also can be influenced by an externally applied injection signal. If the frequency of the externally applied signal matches the natural or free-running frequency of the oscillator, no change in resulting output frequency occurs. However, the injection signal does change the incident power reaching the diode. The change in incident power evokes a corresponding change in output power from the diode. Therefore, the circuit resistance optimized for maximum added power under free-running conditions must be modified for injection-locked conditions to compensate for the additional incident power reaching the diode. The amount of compensation depends upon the relative power of the injection signal compared to the maximum added power of the diode.

Figure 42 quantifies required compensation (expressed as a ratio of the resistance required under injection-locked conditions compared to the free-running resistance, which results in maximum added power) versus the injection ratio. This figure is useful in designing transformers for injection-locked oscillator applications. Injection locking is essential to the proper operation of the transmission-mode turnstile power combiner. Each oscillator must remain locked to a single reference in order to maintain the proper 0- and 180-degree phase relationships between the port pairs.

Knowledge of diode behavior such as that shown in Figures 40 and 41 is accumulated from a combination of experimental measurement and theoretical modeling. Theoretical models can predict overall trends in behavior and are useful in initial transformer design before experimental data are available. (A workable oscillator design must be achieved before experimental device-characterization efforts can begin.) Reference 5 describes a model that can be used to generate an initial estimate of the impedance for Ka-band pulsed IMPATTs of the type used in the turnstile oscillators. Appendix H contains necessary input parameters and a listing of a computer program for implementation of the model. Figures 43 through 45 show model predictions of Ka-band IMPATT behavior displayed in various formats. Figure 43 shows predicted diode admittance at the chip reference plane (package parasitics are excluded) for several levels of RF voltage and for frequencies extending from 33 to 50 gigahertz. Figure 44 expresses the same predictions in terms of impedance. By transforming the chip impedance through the package parasitics of Appendix G, the packaged diode impedance (referenced to the top of the diode package) of Figure 45 is generated. The model of Reference 5 does not incorporate saturation effects and, therefore, cannot be used to predict the maximum added power available from the device. This value is determined experimentally by measuring several devices under various operating conditions. In the vicinity of the desired frequency (34.5 gigahertz) at an RF voltage of 12 to 15 V_p [Volts (peak)] (indicated in Reference 5 as the nominal RF voltage range for the IMPATT being considered), the model predicts an impedance of -2.0 ohms real and +j 12.5 ohms imaginary.

To assist in transformer design, Table 2 displays a summary of predicted diode impedances and measured midcavity impedances near the operating frequency of 34.5 gigahertz.

TABLE 2. Summary of Diode and Midcavity Impedance Data.

Frequency, GHz	Diode impedance, ohms ($V_{RF}=12V_p$)	Midcavity impedance, ohms
33.0	-2.29 +j 10.36	36.3 +j 71.5
33.5	-2.25 +j 10.89	45.1 +j 84.2
34.0	-2.21 +j 11.42	75.7 +j 108.7
34.5[a]	-2.18 +j 11.97	151.0 +j 65.8
35.0	-2.15 +j 12.51	119.8 - j 24.5
35.5	-2.12 +j 13.07	73.1 - j 23.9
36.0	-2.09 +j 13.63	57.4 - j 14.8

[a]Operating frequency.

The model also can be used to predict representative levels of RF output power (although saturation effects are not included). Table 3 summarizes the predicted diode impedance and RF output power at the operating frequency. The RF output power is given by

$$P = \frac{1}{2} \frac{|V_{RF}|^2}{|Z_d|^2} R_d$$

As expected, no evidence of power saturation is observed in the predicted power levels. A small signal resistance of -3.39 ohms is predicted.

The information of Tables 2 and 3 and the background regarding the injection-locked behavior of oscillators provide a basis for designing the oscillator-transformer section.

The function of the transformer section is to transform the impedance at midcavity to that value of circuit impedance at the diode, which results in maximum added power at the oscillator input (at the desired frequency). This functional objective was modified slightly to simplify the design of the

TABLE 3. Predicted RF Power Versus RF Voltage.

V_{RF}, V_p	Diode impedance, ohms	RF power, watts
0	-3.39 + j 10.89	0
3	-3.26 + j 11.02	0.11
6	-2.93 + j 11.42	0.39
9	-2.54 + j 11.66	0.72
12	-2.18 + j 11.97	1.06
15	-1.87 + j 12.21	1.38
18	-1.63 + j 12.40	1.69

Note: Frequency, 34.5 gigahertz.

turnstile oscillators. The modified design approach requires only maximum added power generation at the device port (rather than maximum added power at the oscillator output port). For free-running oscillators with optimized transfer efficiency, the distinction is of no consequence, since maximum added power at the device plane and maximum oscillator transfer efficiency assure maximum added power at the output. For injection-locked oscillators, however, maximum added power at the output requires simultaneous optimization of device-added power, as well as power-added efficiency of the oscillator circuit. (The optimization of these quantities is not independent, since the power-added efficiency of the circuit is termination dependent. Appendix E contains further discussion of power-added efficiency considerations.) Transformer design appears to exercise a stronger influence upon the generation of power from the device than it does upon the transferral of power from the input to the output of the oscillator module. Therefore, the focus of transformer design for the turnstile oscillators was limited to generation of maximum device-added power. Consideration of overall power-added efficiency optimization has been deferred for later consideration.

The transformer design of Figure 46 was selected for use with the oscillator cavity of Figure 35 for initial turnstile transmission-mode power-combining evaluations. The design consists of a 50-ohm, 0.0404-inch-length coaxial spacer followed by a 23-ohm, 0.053-inch-length Rexolite transformer. This design presents a transformed impedance at the diode cap of 5 ohms resistive and -16 ohms reactance at the desired operating frequency (34.5 gigahertz). These values are slightly shifted from those predicted by the theoretical diode model but resulted in acceptable oscillator operation, as shown in Figure 47. Figure 47 is a plot of the injection-locked output power versus the injection frequency for the selected transformer design. Table 4 lists a summary of the input (injected), output, and output-added powers at 34.5 gigahertz.

TABLE 4. Injection, Output, and Added Powers at 34.5 Gigahertz.

Injection power, watts (average)	Output power, watts (average)	Added power, watts (average)
0	0.92	0.92
0.045	1.26	1.22
0.12	1.44	1.32
0.23	1.59	1.36
0.45	1.66	1.21

From Table 4, it is seen that maximum added power occurs at an input injection level of 0.23 watt. (In the turnstile transmission-mode power combiner, this would correspond to approximately 1 watt of power at the turnstile input, since the injected power splits evenly four ways.) The injection-locking bandwidth of the oscillator at this level of injection power is approximately 350 megahertz. By comparison with Figure 37, the transfer efficiency bandwidth is 900 megahertz. This comparison indicates that the reduction in bandwidth arises primarily in the generation of the power at the diode plane rather than from losses or attenuation associated with transferral of the power through the cavity to the oscillator output.

An estimate of injection-locking bandwidth can be computed based upon the rate of change of the circuit reactance with frequency. Figure 48 shows the impedance at the diode cap for the selected transformer design over an extended frequency range. An expression for the locking bandwidth in terms of the resonant frequency, injection gain, and loaded Q of the oscillator circuit is given by

$$BW = \frac{2Fr\sqrt{\frac{P_{inj}}{P_{out}}}}{Q}$$

The Q for a system whose impedance is known at a given terminal pair is given as

$$Q = \frac{w}{2R} \frac{dX}{dw}$$

Table 5 compares the injection-locking bandwidth computed from the above equations based upon the data of Figure 48 using the measured values of Figure 47. From Figure 48, the R at resonance is 5 ohms. The dX/df at resonance is 10 ohms/800 megahertz. This corresponds to a Q of 43.13.

TABLE 5. Comparison of Computed and Measured Injection-Locking Bandwidth.

P_{inj}/P_{out}	Calculated bandwidth, MHz	Measured bandwidth, MHz	
		10%	Break-lock
0.036 (-14.4 dB)	304	180	360
0.083 (-10.8 dB)	460	260	550
0.145 (-8.4 dB)	609	370	800
0.271 (-5.7 dB)	833	560	1050

The circuit impedance data (referenced at the diode cap) in Figure 48 was generated using Touchstone, a CAD computer program developed by EEsof Incorporated. Touchstone was used to analyze and to optimize the transformer design of Figure 48 based upon the cavity design of Figure 35. The oscillator cavity was represented by a two-port S-parameter data file containing measured S-parameter data referenced to 50 ohms at both ports. (Touchstone requires a uniform reference impedance at both ports.) Touchstone then retransforms the cavity S-parameter data to reference the output port to external Ka-band waveguide and the input port to 50-ohm coax.

Appendix I contains a listing of the Touchstone circuit file used in circuit design and optimization. Also contained in Appendix I are listings of the cavity-S-parameter data with both ports referenced to 50 ohms and with the output normalized to the waveguide impedance.

The circuit file of Appendix I can be used also to optimize the single-section transformer design to other desired impedances at the diode cap. The desired circuit resistance and reactance are entered as target values in the optimization block of the Touchstone circuit file. The spacer length, transformer length, and transformer characteristic impedance are variables in the optimization process. As the optimization proceeds, updated values of discontinuity capacitance—because of changes in the outer conductor—are automatically computed each time the characteristic impedance of the transformer is changed. The primary objective in the optimization procedure is achievement of the target resistance and reactance values at the desired center frequency. A secondary objective of flat circuit resistance with frequency also can be specified by defining the optimization error function to include target resistance values at other frequencies near the operating frequency. Secondary emphasis is conveyed by assigning reduced weightings at the other frequencies. For the selected design of Figure 48, the procedure produced a real part impedance locus that is fairly symmetric about the operating frequency and remains close to the target value of 5 ohms over 400 megahertz of bandwidth. The design procedure has been extended to include multiple transformer sections for future extended frequency designs.

This concludes discussion of the design and optimization of the turnstile oscillator module. The next section addresses aspects of integrating four such oscillators with a symmetric turnstile combiner in the transmission-mode configuration.

OSCILLATOR AND TURNSTILE INTEGRATION

This section considers several aspects of integrating IMPATT oscillator modules with the turnstile power combiner. As indicated in the section entitled Theory of Cubic-Turnstile Operation, proper operation of the turnstile power combiner requires amplitude and phase matching of the oscillators being combined. In the reflection mode of turnstile operation, all modules should be matched in amplitude and phase. In the transmission mode, all modules should be of equal amplitude with one pair of opposing oscillators exhibiting 180 degrees of reflection phase with respect to the other opposing pair. The major portion of this section deals with various aspects of achieving the required module amplitude and phase matching for successful turnstile power combining.

Topics for discussion include (1) device (IMPATT) and circuit uniformity considerations, (2) measurement (and adjustment) of the reflection amplitude and phase of injection-locked IMPATT oscillators under pulsed RF conditions, (3) compatible passive and active phase and amplitude-matching conditions for turnstile transmission-mode demonstration, (4) combiner cooling, and (5) mechanical interface considerations.

UNIFORMITY CONSIDERATIONS

A key consideration in turnstile power combining (as in other methods of power combining) is unformity. For most efficient power combining, all active devices and oscillator circuits should be uniform and oriented symmetrically with respect to the turnstile junction.

The greatest challenge to preserving uniformity is posed by the active device. Figure 49 illustrates this point by displaying a typical distribution of free-running output powers versus oscillation frequencies for several Ka-band IMPATTs. The IMPATTs were all tested in the same oscillator circuit and operated at nearly the same bias conditions. (No diode preselection was performed prior to testing.) A resonant frequency spread of approximately 900 megahertz is observed. By referring to the circuit-impedance-versus-frequency plot of Figure 48, 900 megahertz corresponds to about ±4.5 ohms of variation in diode reactance. The free-running power variation in the distribution is considerable (about 1.3 watts). This distribution is the result, in part, of a variation in circuit resistance of ±0.3 ohms over the 900-megahertz frequency span. Additionally, a spread in bias current of ±0.1 ampere also contributed to the observed power variation. However, these contributions considered, the remaining variation in free-running power indicates substantial device-to-device nonuniformity.

Device uniformity for power-combining applications can be improved by preselecting diodes with similar characteristics based upon power-versus-frequency-distribution data as in Figure 49. The effects of device nonuniformity can be reduced further by including adjustable elements in the oscillator circuit design to compensate for dissimilar devices. For example, the cavity backshort tuner can be used to make small adjustments to the oscillation frequency (and, hence, the reflection phase). Additionally, the bias current to each device may be adjusted slightly to compensate for differences in device-added powers.

To assess oscillator circuit uniformity, a single diode was installed and tested in four oscillators each of identical construction and design. Each module employed a fixed (nonadjustable) cavity backshort. The bias current to the IMPATT was adjusted to the same value (1.7-ampere peak) for each module test. Appendix J contains the output-power-versus-frequency results under free-running and various levels of injection-power inputs for each module. Table 6 summarizes these results at the nominal operating frequency of 34.5 gigahertz.

A spread of 107 megahertz in oscillation frequency is observed between the four modules. A variation in free-running power of 0.26 watt (average) was measured. Comparing these data to the data of Figure 49, it is clear that circuit-related uniformity is about an order of magnitude better than device-related uniformity.

TABLE 6. Summary of Oscillator Uniformity Tests at 34.5 Gigahertz.

	Module #2	Module #3	Module #4	Module #5
Free-running frequency, GHz	34.482	34.375	34.470	34.450
Free-running power, W (average)	1.15	1.27	1.01	1.18
Device-Added Power, W (Average)				
P_{inj} = 0.11, W (average)	1.46	1.60	1.44	1.46
P_{inj} = 0.11, W (average)	1.59	1.59	1.59	1.66
P_{inj} = 0.23, W (average)	1.51	1.39	1.65	1.62
P_{inj} = 0.45, W (average)	1.39	1.06	1.55	1.36

MEASURING THE REFLECTION AMPLITUDE AND PHASE OF INJECTION-LOCKED IMPATT OSCILLATORS UNDER PULSED RF CONDITIONS

The reflection amplitude and phase performance of an oscillator module at various levels of pulsed RF injection lock can be evaluated by using a pulsed reflectometer system. Figure 50 shows the Ka-band system used to perform the final phase and amplitude adjustments of the turnstile oscillator modules. Figure 51 is a block diagram of the system, the operation of which is briefly described in the following paragraphs. Table 7 lists the components of the reflectometer system.

The test signal is generated by sweep generator (1). For single-frequency measurements, the generator is phase locked through the source-locking frequency counter (2). The amplitude of the generated signal is externally leveled to provide a constant power level at the reflectometer input. RF switch (7) impresses the pulsed waveform (500-nanosecond pulsewidth, 45% duty factor) upon the generated signal.

Timing unit (32) generates the appropriate timing pulses for the RF switch and the pulsed bias modulators that supply pulsed current drive for the IMPATTs. The timing unit has provisions for up to three separate IMPATT stages of an injection-locked oscillator chain. The timing unit allows independent adjustment of the delay and pulse width of each stage with a resolution of 2 nanoseconds. This provision allows a "stairstep" turn-on and turn-off injection-lock sequencing of successive IMPATT stages. The stairstep sequencing ensures that (1) during turn-on an injection-locking signal is present before bias is applied to the IMPATT and (2) during turn-off the IMPATT bias is removed before the injection signal is turned off.

Coupler (8) and power meter (10a) monitor the traveling wave tube (TWT) (11) input power to help prevent overdriving the tube. Coupler (12) and power meter (10b) provide the sensor for the sweep-generator leveling loop.

The reflectometer phase bridge begins with coupler (16). The coupled port is used to provide a constant amplitude, adjustable phase reference to the L port of the double balanced mixer (22). The direct current output from the I port of the mixer is proportional to the phase difference between the reference phase at port L and the reflected phase from the device under test at port R. The reflected signal is sampled by directional coupler (20a), while coupler (20b) samples the incident wave. A portion of the reflected wave sample is supplied to a spectrum analyzer (31) for frequency-domain analysis, and a portion is supplied through a series of waveguide switches (23 and 24) to a crystal detector and power meter for time-domain viewing. The remaining portion of the reflected wave sample supplies the test signal to the R port of the phase detector. Waveguide switch (23) allows selection of the incident signal for power measurement and waveform characterization.

Control processor (18) controls a rotary vane attenuator (19), which is used to adjust the incident (injection) power level. The processor also controls the programmable phase shifter, which is used to maintain phase quadrature between the L and R ports of the phase detector (22). The difference in the absolute phase setting of (17) at quadrature for a reference short and the quadrature setting for the device under test is used to compute the phase angle of the device reflection coefficient. The magnitude of the reflection coefficient is computed by comparing the power measured by power sensor (25) for the device under test to that measured for a reference short.

TABLE 7. Parts List for Reflectometer System.

Item No.	Item	Manufacturer	Model No.
1	Sweep generator	Integra Microwave	MSG-2100B-CO
2	Frequency counter	EIP Microwave Inc.	578
3	Harmonic mixer	EIP Microwave Inc.	7030022
4	Direct. coupler, 20 dB	Microwave Components and Systems Corp.	R-362-D
5	Variable attenuator	PRD Electronics Inc.	192AF2
6	Fixed attenuator, 6 dB	Microwave Research	A60-6
7	RF switch	MaCom	MA7-28-278DS
8	Direct. coupler, 3 dB	Baytron	3-28-401/1:1
9	Thermistor mount	Hewlett Packard	R486 A
10	Power meter	Hewlett Packard	432 A
11	TWT, 3 W	Hughes	8001 H
12	Direct. coupler, 10 dB	Waveline	1074
13	Fixed attenuator, 20 dB	Microwave Research	A60-20
14	Circulator	P & H Labs	B3-R15320
15	Low-power termination	Defense Commun.	028-600
16	Direct. coupler, 20 dB	Microlab/FXR	T716U
17	Programmable rotary phase changer	Flann Microwave	2266
18	Control processor	Flann Microwave	CP 7942
19	Programmable rotary attenuator	Flann Microwave	2261
20	High directivity directional coupler	Flann Microwave	22132-20
21	Medium-power term.	Flann Microwave	2210
22	Mixer	Spacek Labs	MKa-4
23	Remote-controllable waveguide switch	Flann Microwave	2233-2E
24	Manual waveguide switch	Baytron	3A-44
25	Power sensor	Hewlett Packard	R8486 A
26	Power meter	Hewlett Packard	436 A
27	Fixed attenuator, 10 dB	Microwave Research	A60-10
28	Crystal detector	Hewlett Packard	422A
29	Oscilloscope	Hewlett Packard	180 A
30	Harmonic mixer	Hewlett Packard	11971 A
31	Spectrum analyzer	Hewlett Packard	8569 B
32	Timing unit	In-house design	---
33	Pulsed bias modulator	In-house design	---
34	Direct. coupler, 10 dB	Baytron	3-20-400/10

ADJUSTING THE GAIN AND PHASE OF INJECTION-LOCKED OSCILLATOR MODULES

Figure 52 plots the predicted values for the magnitude and phase of the reflection coefficient at the input (cavity iris) of an IMPATT oscillator module operating under injection lock. The data of Figure 52 were generated using "Touchstone" CAD software. The prediction was based upon measured scattering-parameter data for the cavity of Figure 35, the transformer design of Figure 46, and a simulated IMPATT impedance modeled by a negative resistance of -2.0 ohms in series with a lumped inductor of value 0.0757 nanohenry. The inductor value was chosen to resonate with a nominal circuit reactance of -j 16 ohms at 34.6 gigahertz. (This simple IMPATT model is only valid at resonance but can be used qualitatively for first-order frequency behavior.) At resonance (34.5 gigahertz), a nominal reflection phase angle of -35 degrees is predicted.

The measured phase angle for a given IMPATT oscillator module depends upon several factors. These include the characteristics (package parasitics, doping profile, etc.) of the particular IMPATT installed in the module, the level of injection power incident upon the IMPATT, the bias current supplied to the diode, the ambient heat-sink temperature of the oscillator, and the circuit reactance presented to the diode at the diode cap. However, in general, the resonance value of -35 degrees predicted by Figure 52 agrees well with observed measurements.

To accommodate these possible sources of phase variation between modules, the input reflection phase angle can be adjusted to a common value by adjusting the cavity backshort tuner while maintaining a constant injection frequency. Moving the backshort changes the circuit reactance (hence, reflection phase) without significantly affecting the circuit resistance (output power) of the IMPATT oscillator. By adjusting the backshort, the reflection phase can be tuned over an appreciable range (±50 degrees) without dramatically affecting output power. Any small change in output power that might occur during phase adjustment can be offset by readjusting the IMPATT bias current. The bias current change may induce a slight change in resonance frequency, which, in turn, affects reflection phase. By alternately adjusting the cavity length and IMPATT bias current, the amplitude and phase of the reflection coefficient for each turnstile oscillator can be set to a desired value.

OSCILLATOR PHASE MATCHING FOR INITIAL DEMONSTRATION OF TURNSTILE TRANSMISSION-MODE POWER COMBINER

The purpose of the initial turnstile testing was to demonstrate (at a single frequency) that transmission-mode power combining of IMPATT diode oscillators could be achieved. Previous passive mode testing (with shorts and offset shorts in place of IMPATT oscillators) demonstrated that the turnstile could be used as a power combiner in the transmission mode. Furthermore, these measurements indicated (in the passive mode) that the junction remained matched (RL <-20 decibels) over a 2-gigahertz bandwidth (4-gigahertz with additional spacers) and demonstrated combining efficiencies in excess of 95% over this bandwidth. The task remained to evaluate turnstile combining with active devices installed.

The initial approach to demonstrate active transmission-mode power combining was to duplicate (as closely as possible) the passive-mode conditions that already had yielded successful results. Passive measurements indicated that the transmission-mode RL at the input port was less than -30 decibels at 34.5 gigahertz (see Figure 31). Therefore, 34.5 gigahertz was selected as the operating frequency for the active-mode demonstration. To simplify initial module adjustment, phase and amplitude matching are confined to a single frequency for this demonstration. The final configuration was evaluated over a range of frequencies, but broadband operation was not an initial design goal. In a further effort to duplicate passive measurement conditions, the reflection phase angles of the oscillators were adjusted to values of 180 and 0 degrees to match the shorts and offset shorts used in the passive configuration. Two oscillators were adjusted to 0 degrees by changing the cavity length and bias current using the method described previously. The other pair of oscillators were adjusted for 180 degrees of reflection phase by installing quarter-wave waveguide shims immediately preceding the oscillator. Final phase trimming was again accomplished by changing the cavity length. All four oscillators were adjusted for an equal output power of 1 watt (average) at 34.5 gigahertz. The injection power for module setup was 0.45 watt (average).

Figure 53 shows the output power versus frequency for each of the four modules. Figure 54 contains photographs of the pulsed reflectometer phase detector outputs (see Figure 51, mixer 22) for each module after phase adjustment. The photographs detail the reflection phase variation throughout the 500-nanosecond pulsewidth. Each vertical division corresponds to 8 degrees of phase variation. Modules 2 and 4 were adjusted for a midpulse phase value of 0 degrees. Modules 3 and 5 were adjusted for a midpulse phase of 180 degrees. The measurements shown for modules 3 and 5 include the additional quarter-wavelength waveguide shim. Modules 2, 4, and 5 track extremely well in phase throughout the pulse duration. Module 3 shows an additional increase of 8 degrees at the leading edge. (Note: To characterize the reflection phase with frequency, additional time waveform sets at other RF frequencies must be measured. Future work will address this task.)

THERMAL AND MECHANICAL CONSIDERATIONS

Figures 55 and 56 show the turnstile junction and oscillator modules in integrated and exploded configurations. The oscillator modules are passively cooled and, therefore, rely upon adjacent cooling channels in the turnstile body to maintain proper heat-sink temperatures. (During testing of individual oscillators, a waveguide shim section with integral cooling channels was placed next to the oscillator under test.) Distributed heat sinking of oscillator modules is an attractive feature of the turnstile combiner for high-power applications.

The integrated turnstile/oscillator combination is a very compact structure, as shown in Figure 55. This compactness led to an unforeseen mechanical interference between two adjacent oscillator modules. The oscillator body "stems" that contain the stabilizing loads experienced a minor mechanical interference at one vertex of the cubic turnstile body. The interference was alleviated by a slight shaving of the stem exterior. Because of this interference, however, the current configuration does not allow evaluation of the reflection mode of turnstile power combining. (Reflection-mode combining requires that all modules be of equal phase. This would require that the waveguide quarter-wavelength shims be removed, thus creating further mechanical interference.)

Elimination of possible interference problems can be accomplished in two ways. First, an additional waveguide shim could be added to each oscillator port. (Passive measurements indicated that the addition of a 0.150-inch shim increased the bandwidth over which the turnstile junction remained matched from 2 to 4 gigahertz.) A second approach for alleviating mechanical interference is to reduce the total length of the stabilizing load "stem." Since the stabilizing loads in this design are matched to the coaxial line (50 ohms), it is possible to move the loads closer to the cavity without affecting RF performance. Additionally, it is possible to reduce the stem length by reducing the length of the load taper. Future work could address achievement of a more compact oscillator design.

TURNSTILE TRANSMISSION-MODE POWER-COMBINING DEMONSTRATION (INITIAL TEST RESULTS)

In the active-mode demonstration, modules 2 and 4 (0-degree reflection phase) were situated at opposing turnstile ports (previously occupied by quarter-wavelength offset shorts in turnstile passive measurements). Modules 3 and 5 (180-degree reflection phase) were situated at opposing turnstile ports, which were occupied previously by reference shorts.

Figure 57 shows the turnstile transmission-mode test configuation. The pulsed reflectometer system (Figure 51) was used as the source of the external injection signal. The reflectometer system also monitored the power reflected from the turnstile junction. The transmitted (or combined) power was monitored by a directional coupler situated between the turnstile and a medium-power waveguide termination. An injection-locked, circulator-coupled, single IMPATT oscillator module (identical in design to the turnstile modules) was used to increase the injection power level from the 0.5-watt average available from the reflectometer to 1 to 2 watts average at the input to the turnstile. Figure 58 shows the injection power versus frequency measured at the input to the turnstile junction of various levels of reflectometer-injected power (0.45-, 0.225-, 0.1125-, and 0.045-watt averages).

The injection-locking bandwidth achieved in this initial demonstration was 100 megahertz—much narrower than the passive measurements of the turnstile (2.0 gigahertz) or the injection-locked bandwidths of the single modules (600 megahertz). The most likely factor to influence combining bandwidth for this demonstration is the phase or amplitude runout with frequency for the various modules. As described previously, the modules were phase and amplitude matched at a single frequency of 34.5 gigahertz. For the purpose of the initial demonstration, the phase runout with frequency of each module was not characterized. Follow-on efforts will address this task.

The injection-locked power characteristic of Figure 59 is centered approximately 30 megahertz above 34.5 gigahertz. This increase is probably caused by the backshort adjustments performed to change the reflection phase from a nominal value of -30 to 0 degrees. This adjustment was not required for proper turnstile operation but was included to assure compatibility with previous passive turnstile measurements. To accomplish phase compatibility, the cavity length was slightly decreased, thus raising the natural oscillation frequency of the module. The natural (free-running) frequency establishes the center frequency of the injection-locked power characteristic. This explains the slight increase in center frequency of Figure 59. Future testing will verify this explanation by eliminating the 0, 180-degree phase constraint in favor of -30, 150 degrees (or whatever phase adjustment results in a natural oscillation at frequencies closer to 34.5 gigahertz for each module). Figure 60 shows the reflected power measured over the bandwidth. Only 0.2 watt is observed at midband.

The injection-locked power characteristic exhibits very rapid falloff near the extreme edges of the band. Slight anomalies are observed further out in frequency. At the lower edge, a resonance-like perturbation is observed; on the upper edge, a slight bump is indicated. To investigate these anomalies further, the injection power input to the turnstile was increased to 2.0 watts average (the uppermost curve in Figure 58). As a result, the injection-locked power-versus-frequency characteristic of Figure 61 was generated. The injection-locked bandwidth increased to 150 megahertz. The anomalous effects increased significantly. These effects may originate because of phase (or amplitude) imbalances between the modules at frequencies near the edge of the injection-locked frequency band. Further investigations (experimental and theoretical) are needed before these anomalies can be satisfactorily explained. The data of Figure 61 give possible indication that with proper phase balance an injection-locked bandwidth of 300 megahertz is realizable with these oscillator modules.

Figure 62 is included as further indication that the turnstile transmission-mode power combiner does indeed achieve injection lock. Shown in Figure 62 is the frequency spectrum of the turnstile output. By comparing this spectrum with those of a locked and free-running oscillator module (Figures 63 and 64, respectively), definite evidence of injection locking is indicated.

CONCLUSIONS AND FUTURE WORK

CONCLUSIONS

This report has described a novel power-combining approach based upon a waveguide junction that exhibits a high degree of symmetry. This power-combining approach possesses advantages particularly suited to high-power, high-frequency applications. These advantages include modularity, which can be utilized to permit cascading and extended-level combining; distributed heat-sinking, which enhances high-power-handling capability; and distributed device placement, which allows extension of this technique to higher millimeter-wave frequencies.

Theory was presented that outlined a method for finding the scattering- (S) matrix of a six-port waveguide junction that exhibits various types and degrees of symmetries. This method was implemented by a computer program contained in Appendix A. S-parameter analysis was used to quantify effects of device phase and amplitude nonuniformity upon power-combining performance.

Three versions of Ka-band turnstile power combiners were evaluated. The final version exhibited a matched bandwidth of 2.3 gigahertz. (Bandwidth of 4.0 gigahertz was achieved with additional spacers on device ports.) The measured combining efficiency of the turnstile exceeded 95% over a 8-gigahertz bandwidth.

The final turnstile design was integrated with four impact avalanche and transit time oscillator modules and evaluated as an active transmission-mode power combiner. Injection locking of the turnstile combiner was achieved with a locking bandwidth of 250 megahertz at an injection gain of 4 decibels. The measured combining efficiency of the transmission-mode turnstile was 95%.

FUTURE WORK

The work presented here demonstrates the concept of turnstile transmission-mode power combining. Additional work remains, however, before the full potential of the turnstile is realized. The following identifies high-payoff areas that require additional work:

Cascaded and extended-level turnstile combining. Development of this area offers tremendous potential for realizing compact, low-cost, high-power, millimeter-wave active power sources.

High-frequency (i.e., 95 gigahertz) turnstile power combining. Turnstile power-combining techniques can be extended to very high millimeter-wave frequencies.

Electromagnetic analysis of waveguide spherical-junction interface. Design of all turnstile hardware to date has been accomplished using "cut-and-try" methodology, since no tools have been developed for analyzing the waveguide/spherical-junction interface. Even so, it is particularly intriguing that such a simple structure exhibits matched performance over a relatively large bandwidth. With the development of proper design and analysis tools, a full waveguide-band turnstile could be possible.

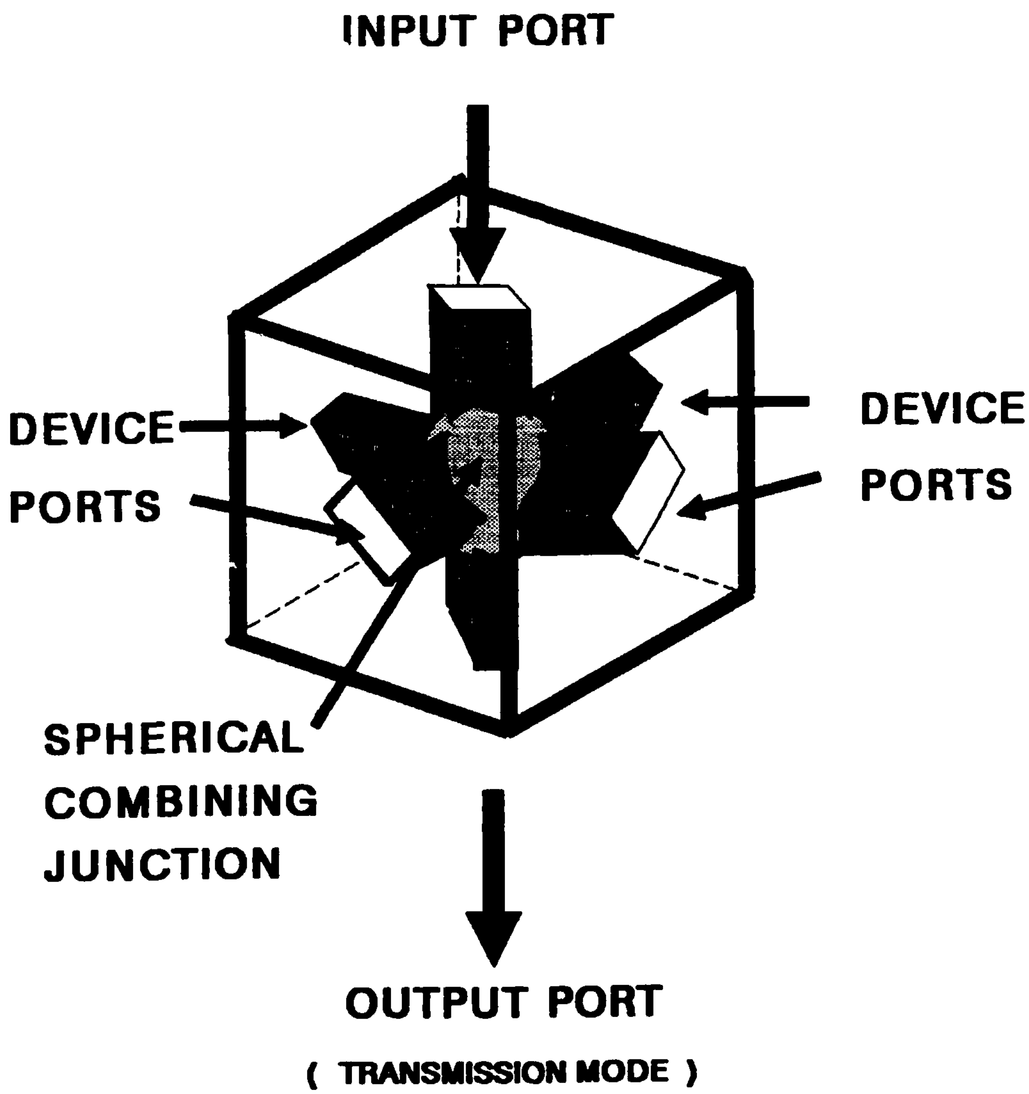


FIGURE 1. Cubic Turnstile Combiner.

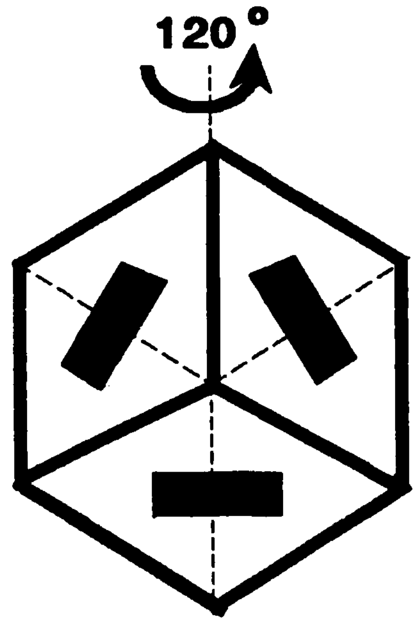


FIGURE 2(a). 120-Degree Rotational Symmetry About Major Cube Diagonal.

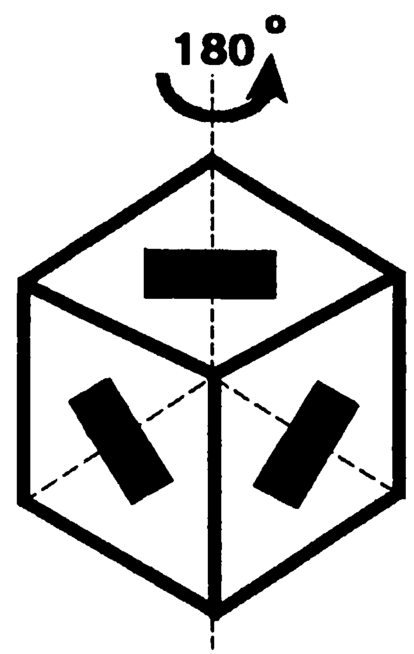


FIGURE 2(b). 180-Degree Rotational Symmetry About Cube Central Axis.

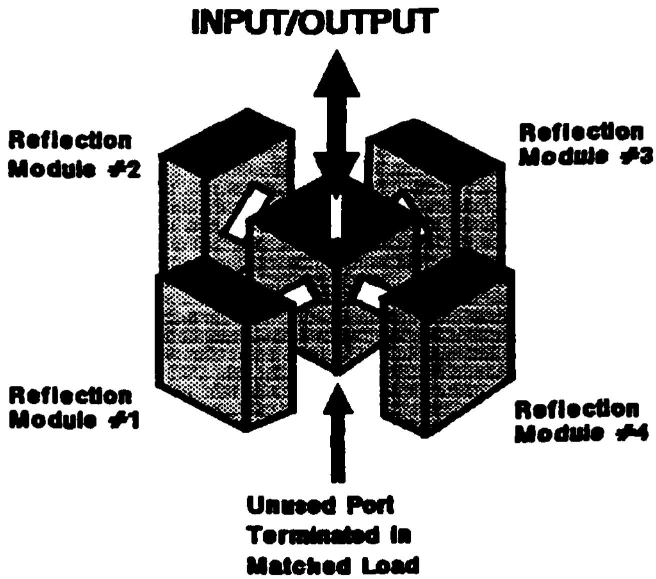


FIGURE 3(a). Cubic Turnstile Combiner Reflection Configuration.

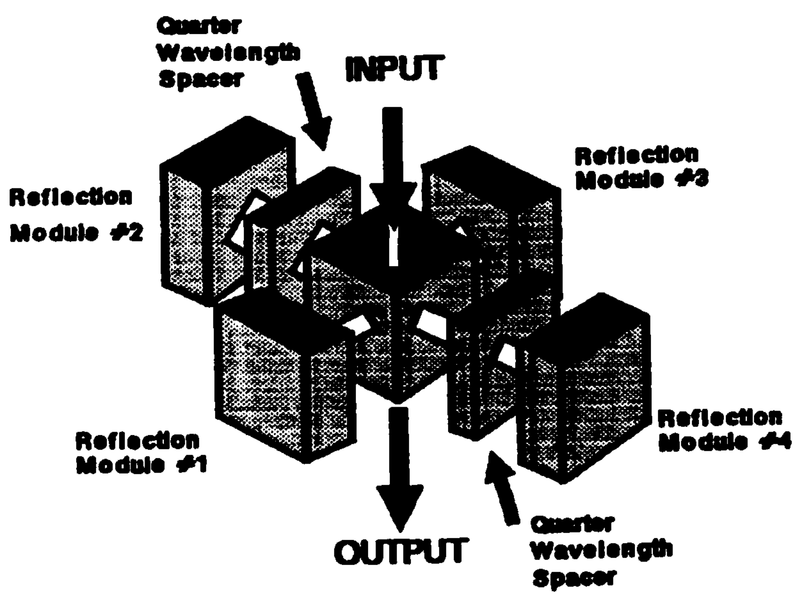


FIGURE 3(b). Cubic Turnstile Combiner Transmission Configuration.

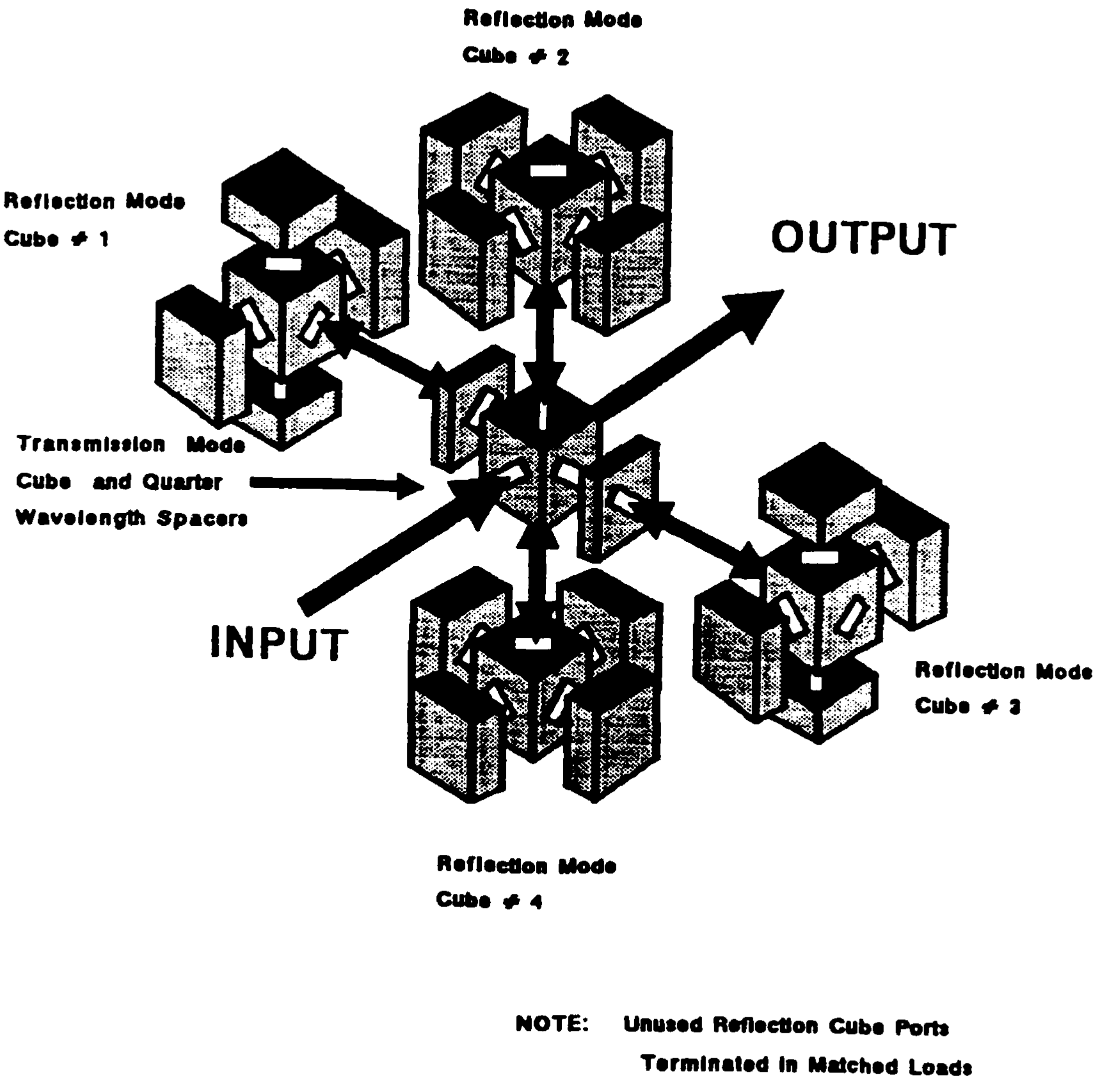


FIGURE 4. Extended-Level Power Combining, Single Transmission-Mode Cube Surrounded by Four Reflection-Mode Cubes for a Total of 16 Active Device Sites.

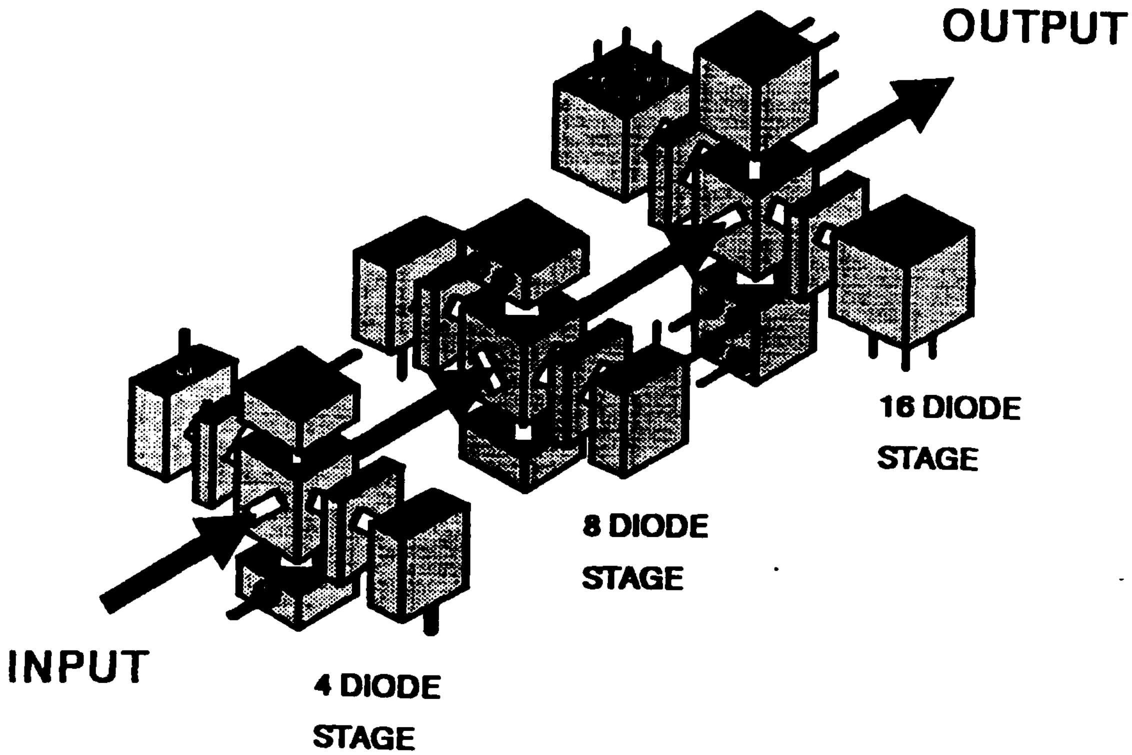


FIGURE 5. Injection-Locked Chain of Cascaded Turnstile Junctions.

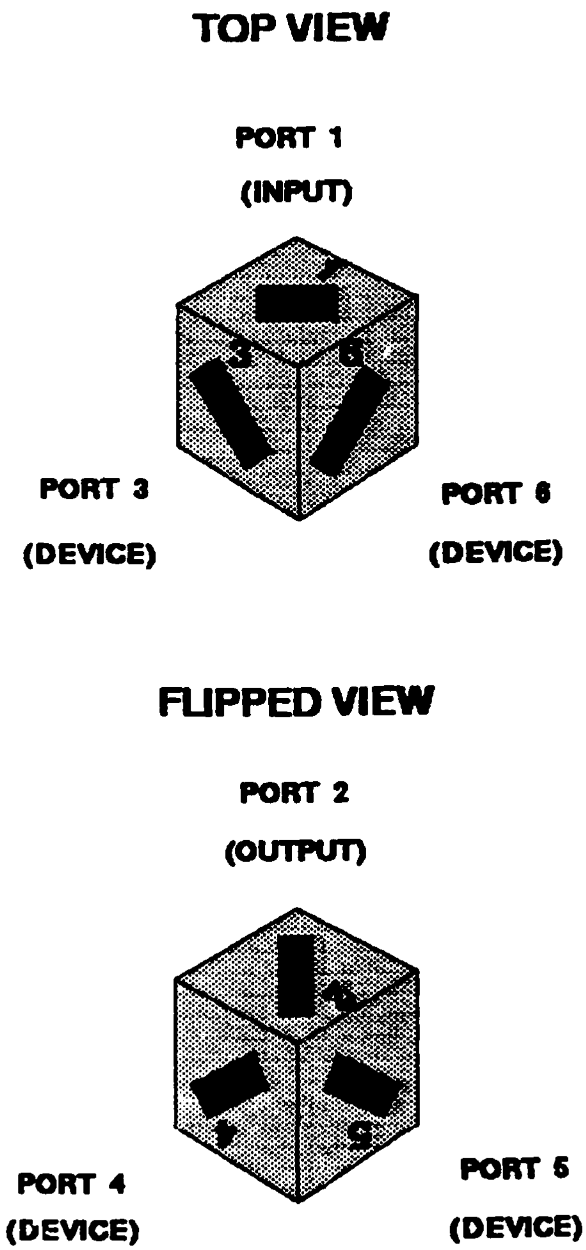


FIGURE 6. Port-Numbering Scheme of Turnstile.

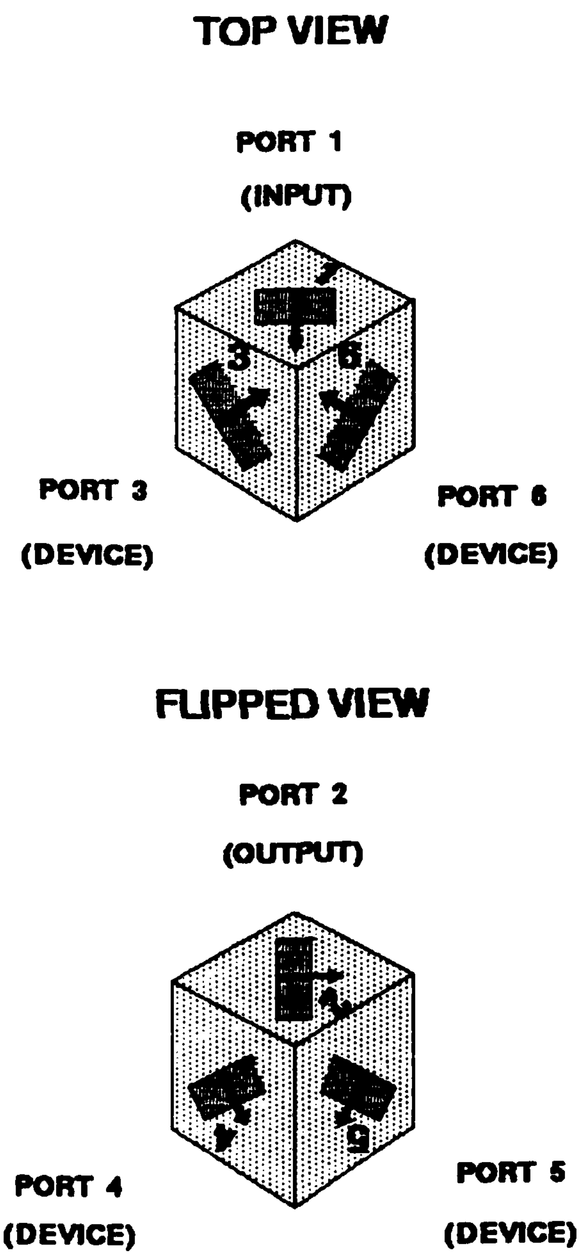
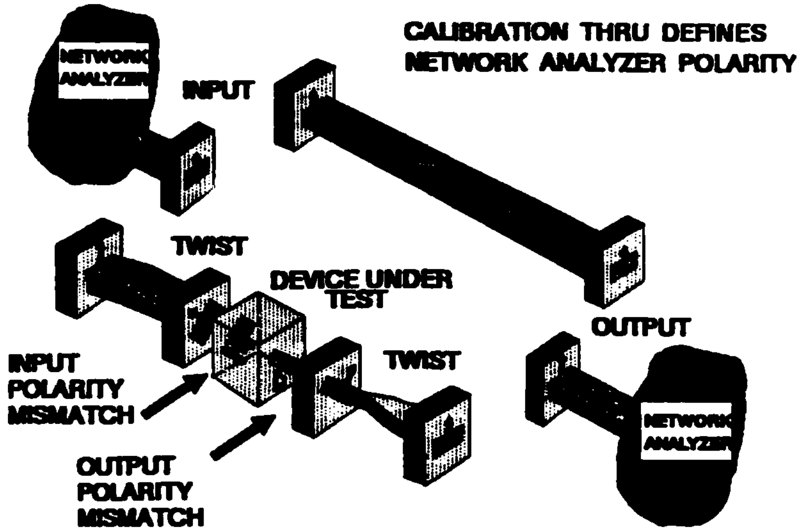
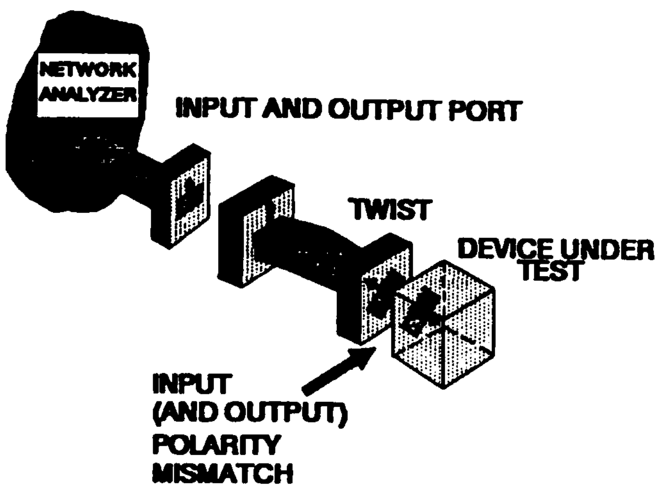


FIGURE 7. Choice of Port Polarities.



EACH POLARITY MISMATCH WILL INTRODUCE A 180-DEGREE PHASE REVERSAL FROM PREDICTED VALUE (IN THIS CASE, MEASURED Sij WOULD EQUAL PREDICTED VALUE SINCE EVEN NUMBER OF REVERSALS ENCOUNTERED)

FIGURE 8(a). Polarity Considerations in Turnstile Transmission Measurements.



REFLECTION MEASUREMENTS ARE UNAFFECTED BY POLARITY CONSIDERATIONS, SINCE IF A POLARITY MISMATCH OCCURS, AN EVEN NUMBER OF POLARITY REVERSALS ALWAYS RESULT AND MEASURED Sii ALWAYS EQUALS THE PREDICTED VALUES

FIGURE 8(b). Polarity Considerations in Turnstile Reflection Measurements.

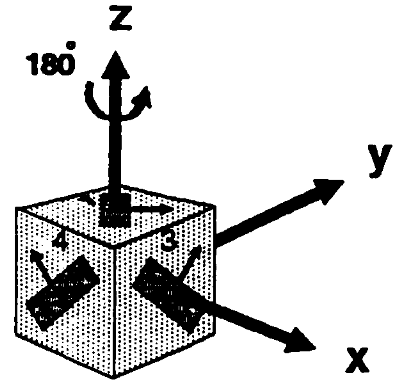


FIGURE 9(a). 180-Degree Rotation About z-Axis.

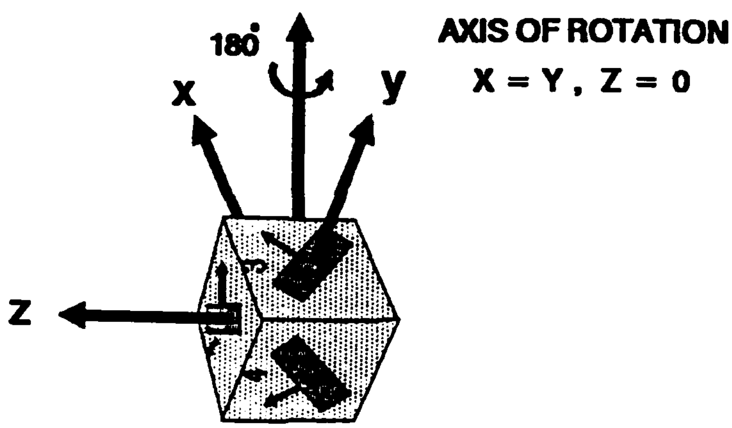


FIGURE 9(b). 180-Degree Rotation About x = y.

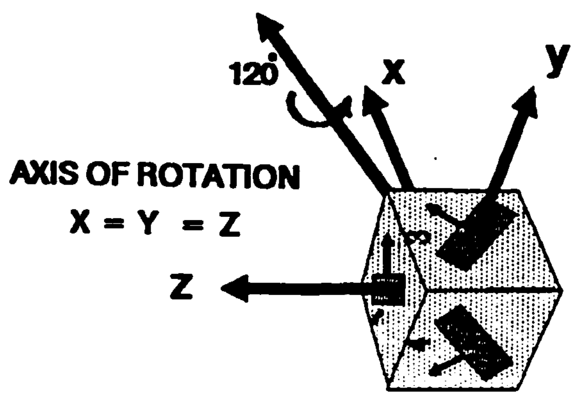


FIGURE 9(c). 120-Degree Rotation About x = y = z.

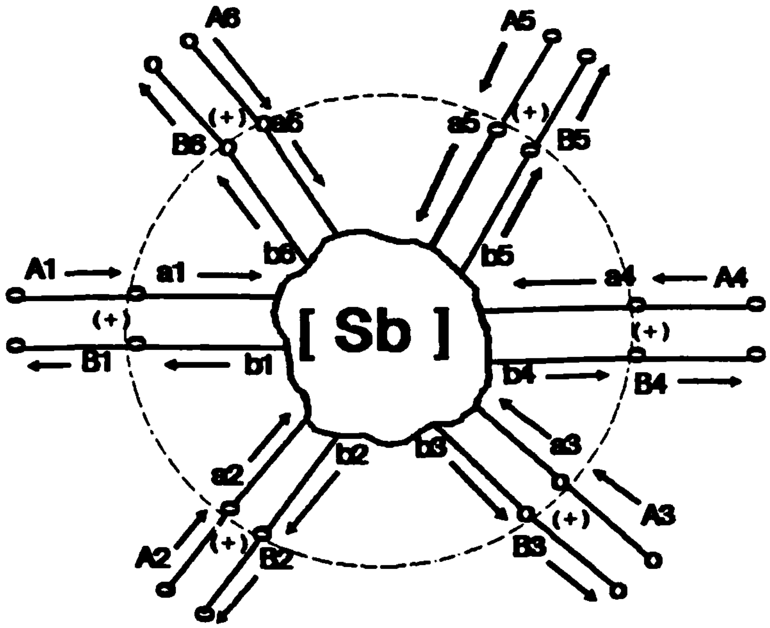


FIGURE 10(a). Junction Before Applying Symmetry Operator.

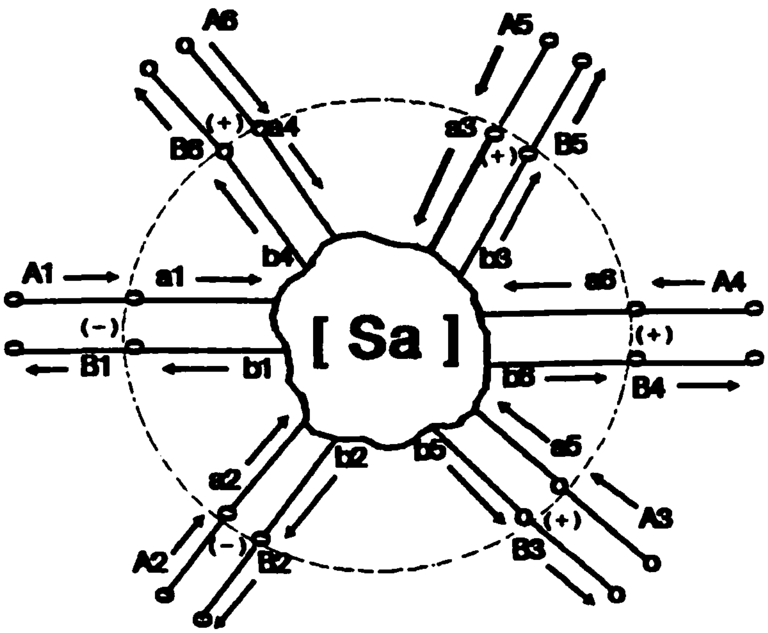


FIGURE 10(b). Junction After Applying Symmetry Operator.

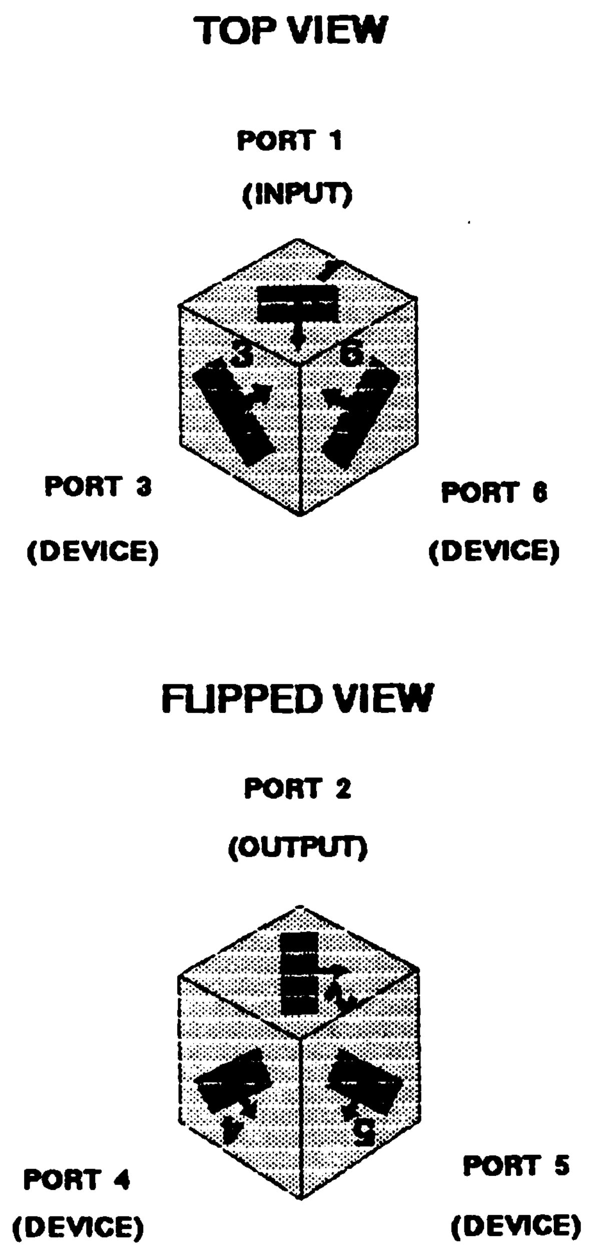


FIGURE 11. Port Polarity Definitions for Cubic Combiner.

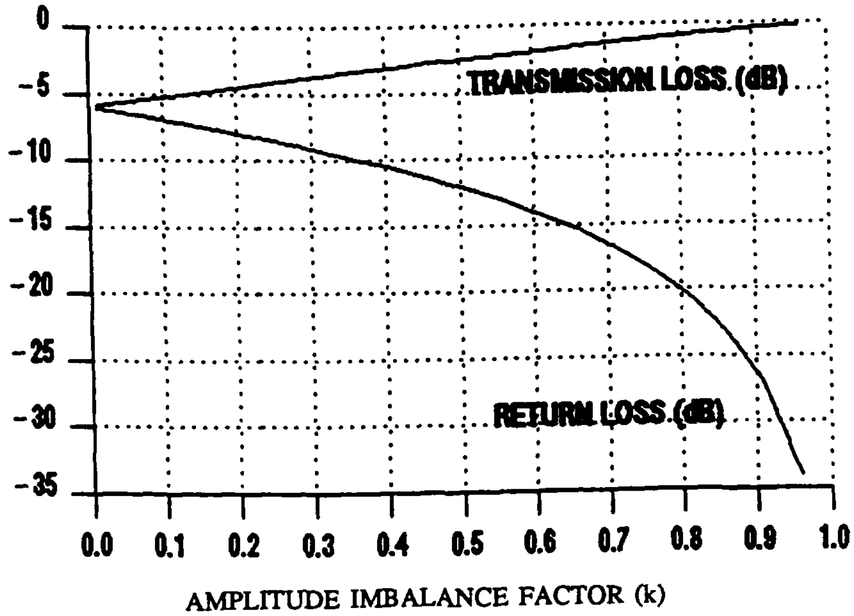


FIGURE 12. Effect of Amplitude Imbalance Between Pairs of Port Terminators Upon Turnstile Performance.

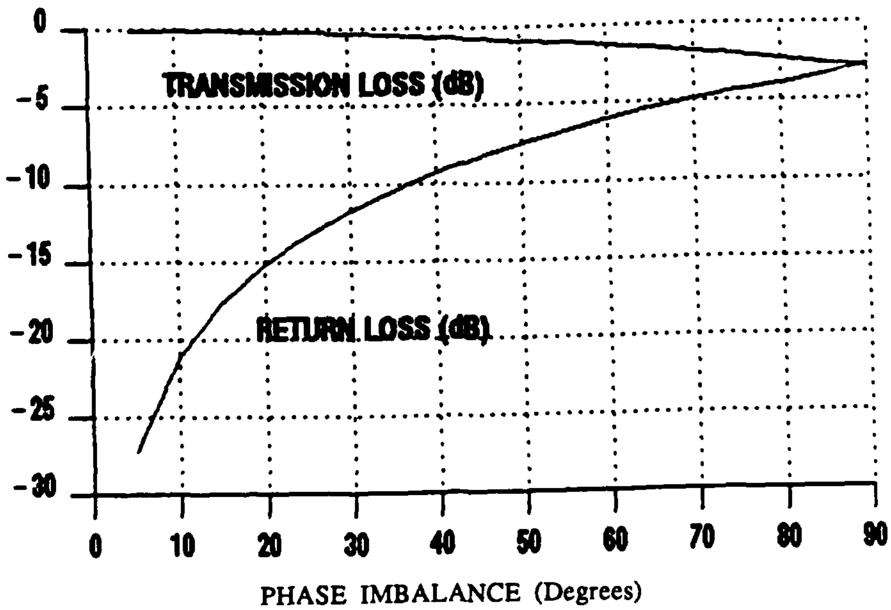


FIGURE 13. Effect of Phase Imbalance Between Pairs of Port Terminators Upon Turnstile Performance.

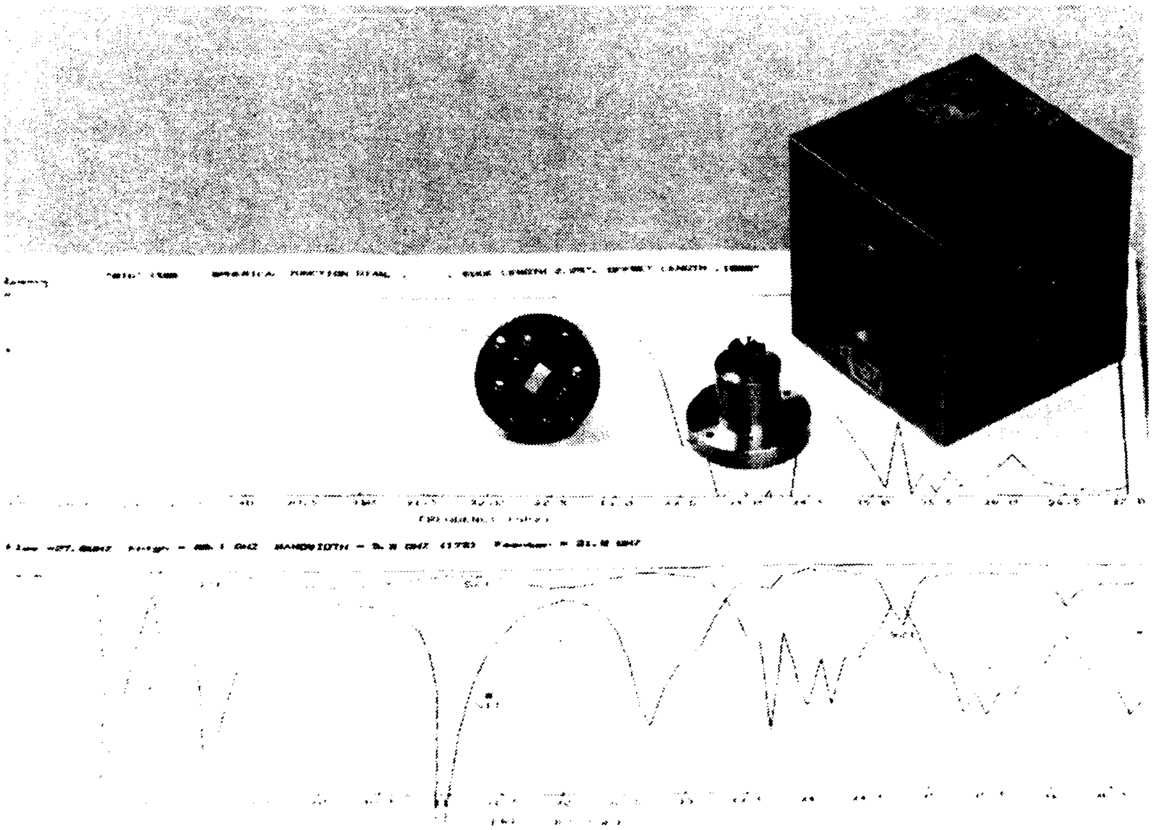


FIGURE 14. Adjustable Turnstile Junction (Design #1).

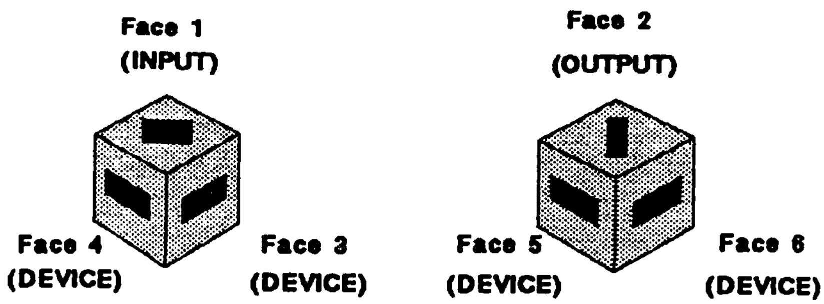


FIGURE 15(a). Planar Device Port Turnstile.

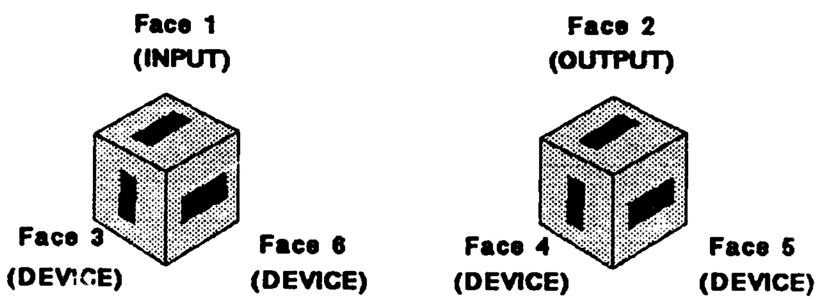


FIGURE 15(b). Purcell's Junction.

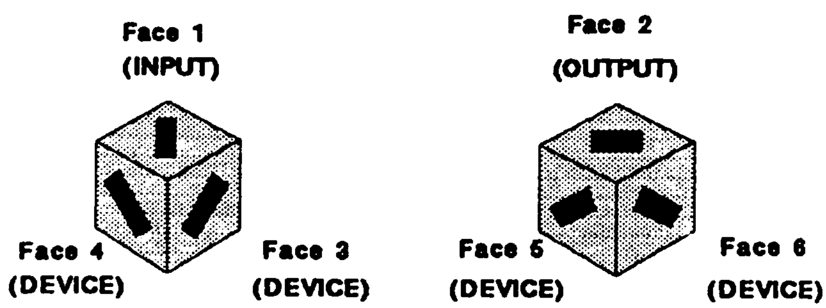


FIGURE 15(c). Configuration #3 Turnstile.

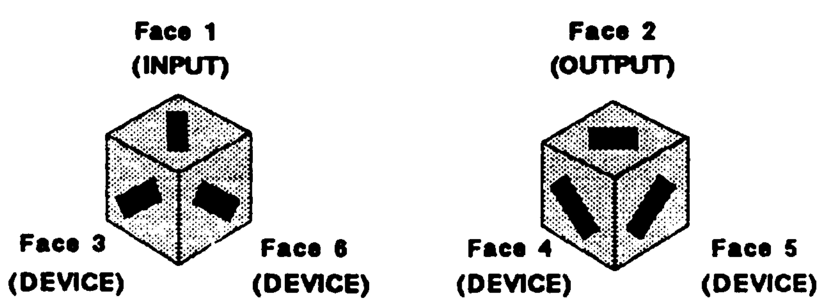


FIGURE 15(d). Cubically Symmetric Junction.

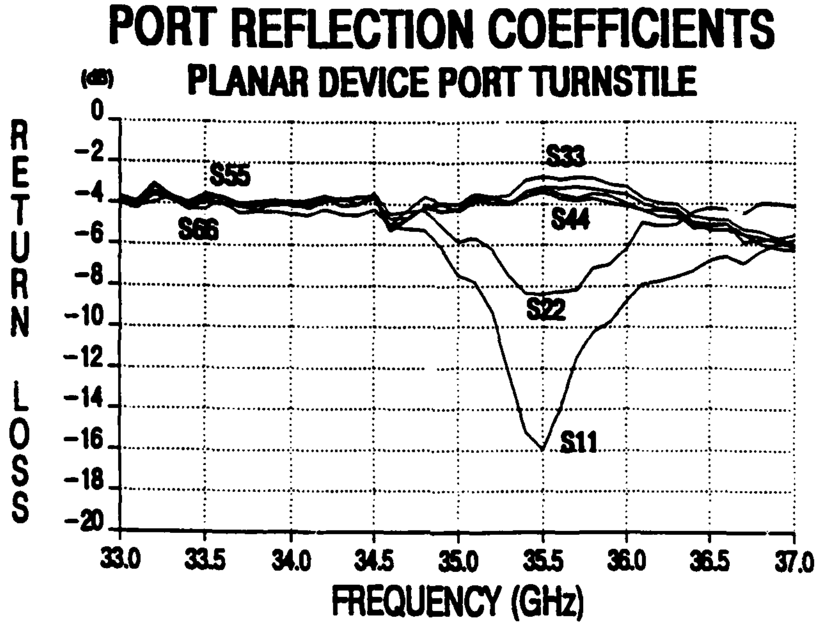


FIGURE 16(a). Return Loss of Planar-Turnstile Junction.

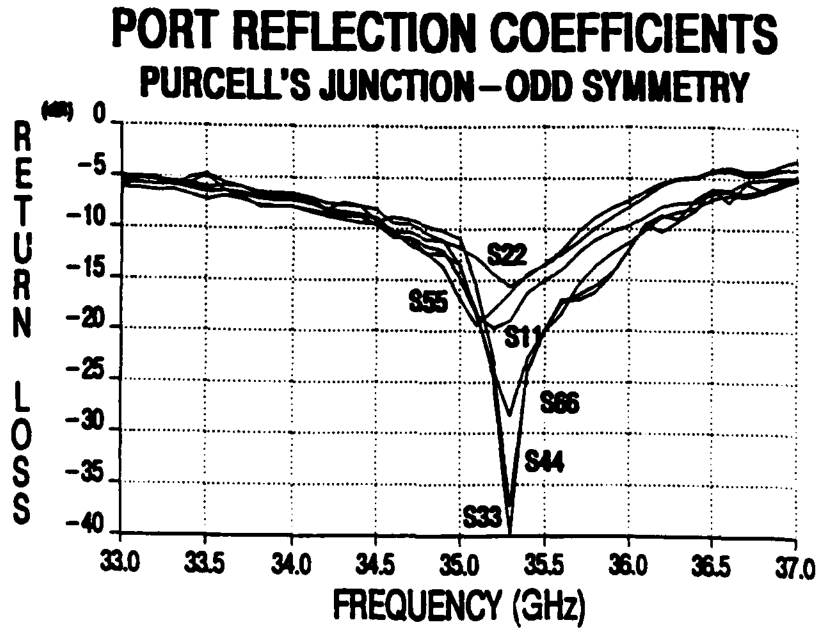


FIGURE 16(b). Return Loss of Purcell's Junction.

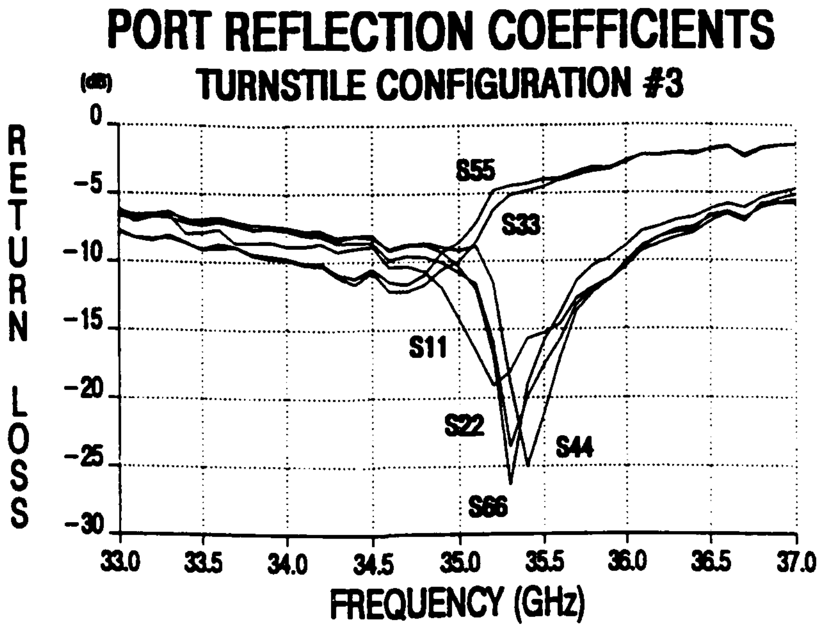


FIGURE 16(c). Return Loss of Configuration #3.

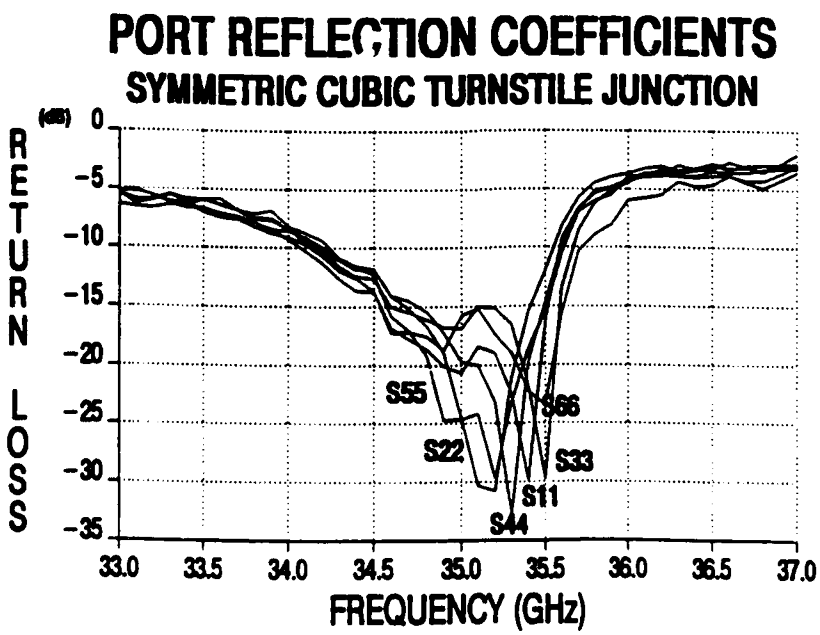


FIGURE 16(d). Return Loss of Cubically Symmetric-Turnstile Junction.

PORT REFLECTION COEFFICIENTS

SYMMETRIC CUBIC TURNSTILE JUNCTION

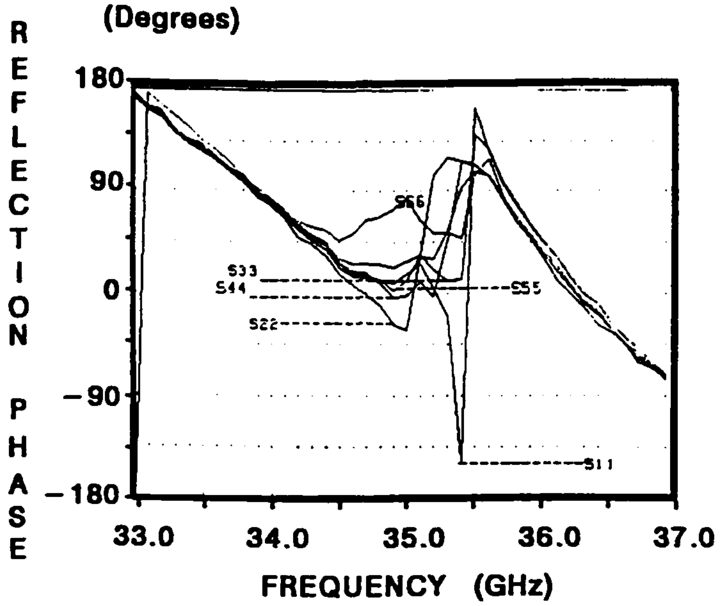


FIGURE 17. Phase of Reflection Coefficient for Symmetric Cubic Turnstile Junction.

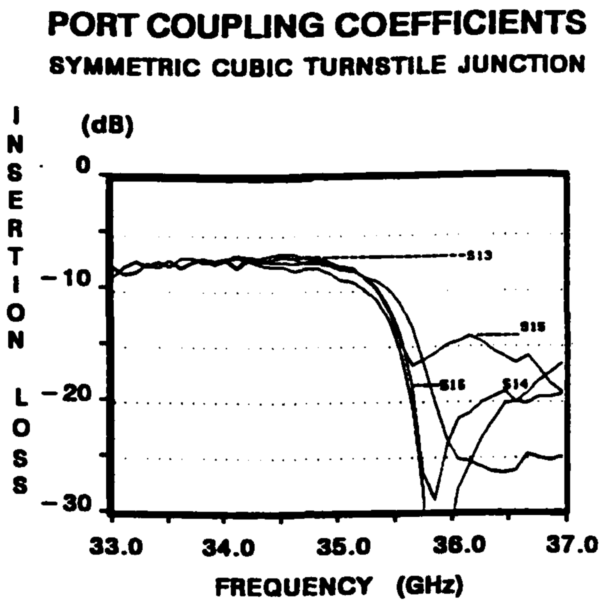


FIGURE 18(a1). Amplitude of Coupling Coefficient Between Input Port and Device Ports for Symmetric Cubic Turnstile.

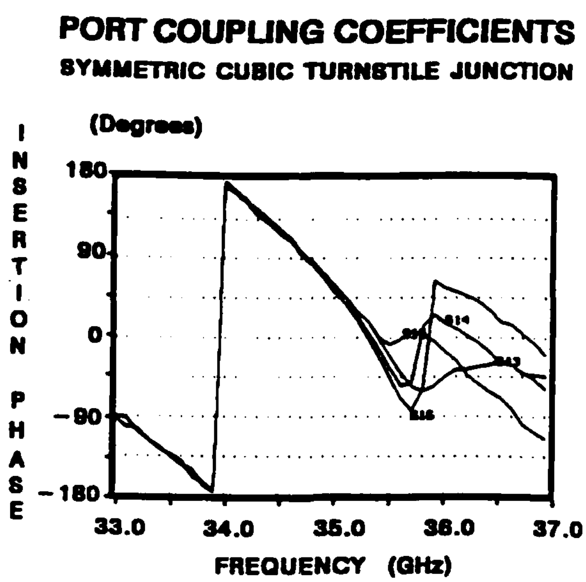


FIGURE 18(a2). Phase of Coupling Coefficients Between Input Port and Device Ports for Symmetric Cubic Turnstile.

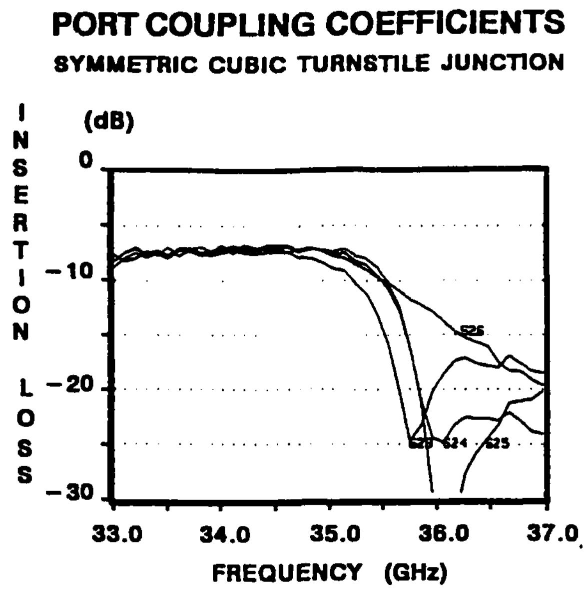


FIGURE 18(b1). Amplitude of Coupling Coefficients Between Output Port and Device Ports for Symmetric Cubic Turnstile.

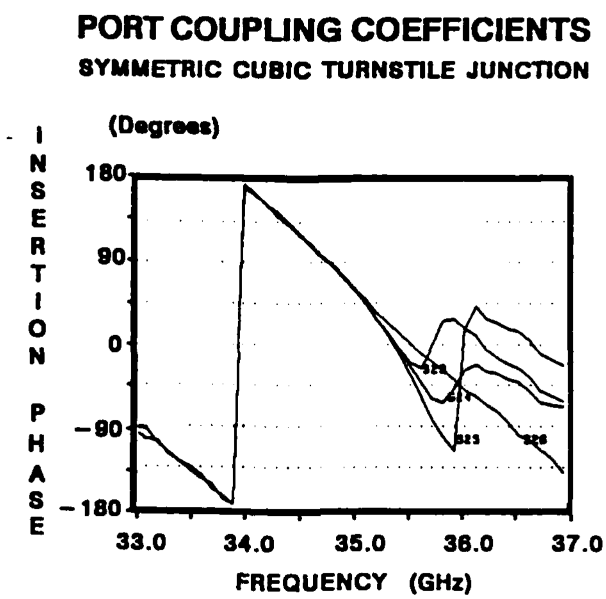


FIGURE 18 (b2). Phase of Coupling Coefficients Between Output Port and Device Ports for Symmetric Cubic Turnstile.

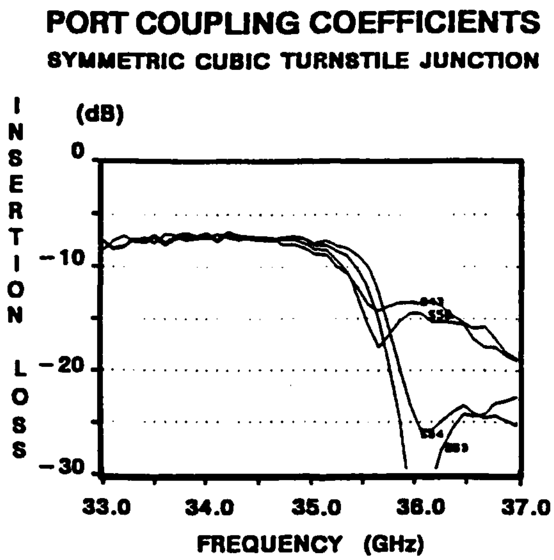


FIGURE 18(c1). Amplitude of Coupling Coefficients Between Device Ports of Symmetric Cubic *Turnstile.*

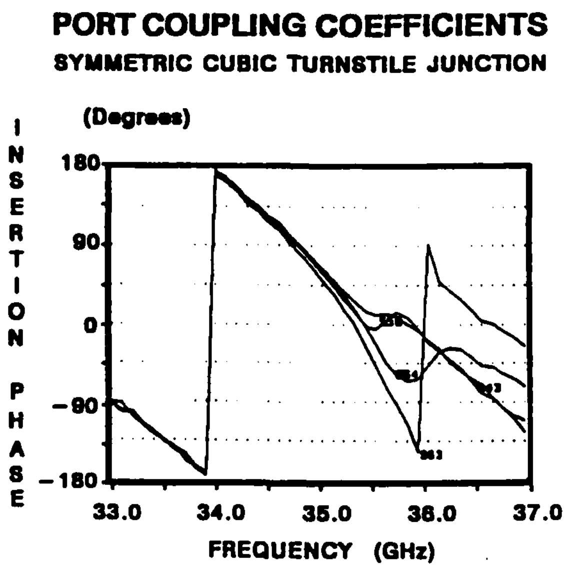


FIGURE 18(c2). Phase of Coupling Coefficients Between Device Ports of Symmetric Cubic Turnstile.

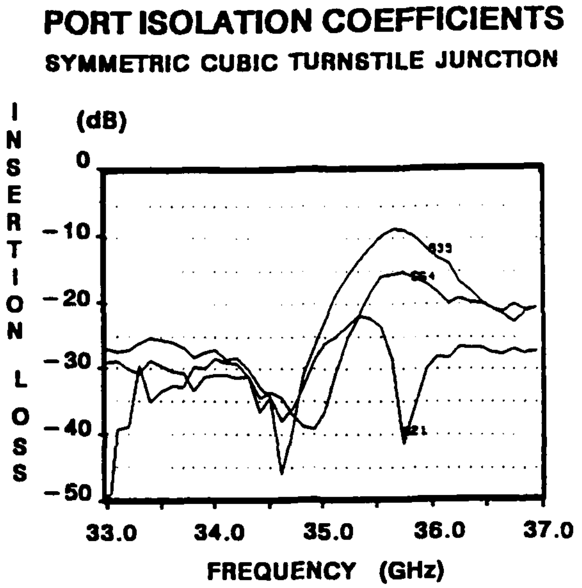


FIGURE 19(a). Amplitude of Cross-Port Isolation Coefficients for Symmetric *Cubic Turnstile Junction.*

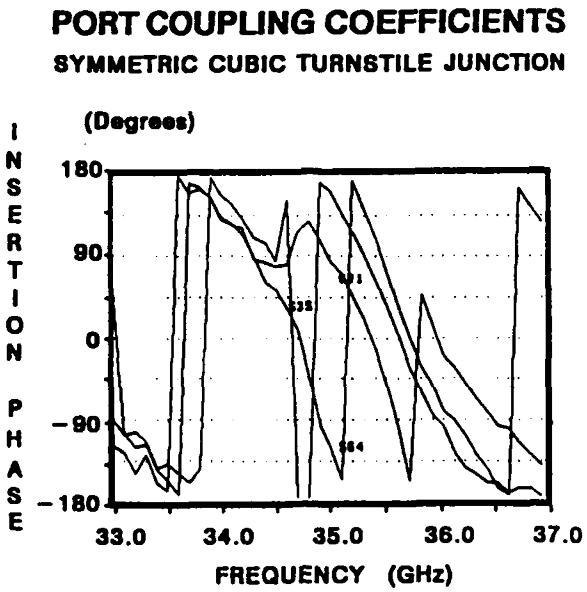


FIGURE 19(b). Phase of Cross-Port Isolation Coefficients of Symmetric Cubic Turnstile Junction.

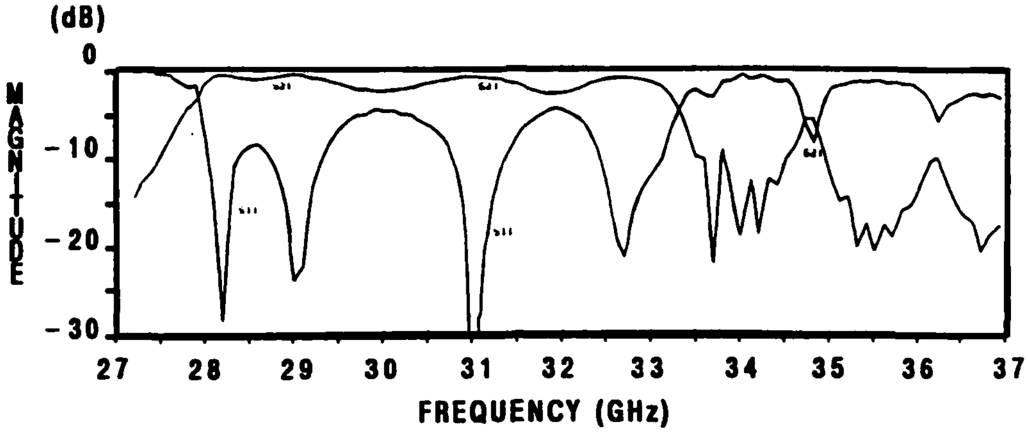


FIGURE 20. Passive Two-Port S-Parameters of Symmetric Cubic Turnstile Junction, Design #1.

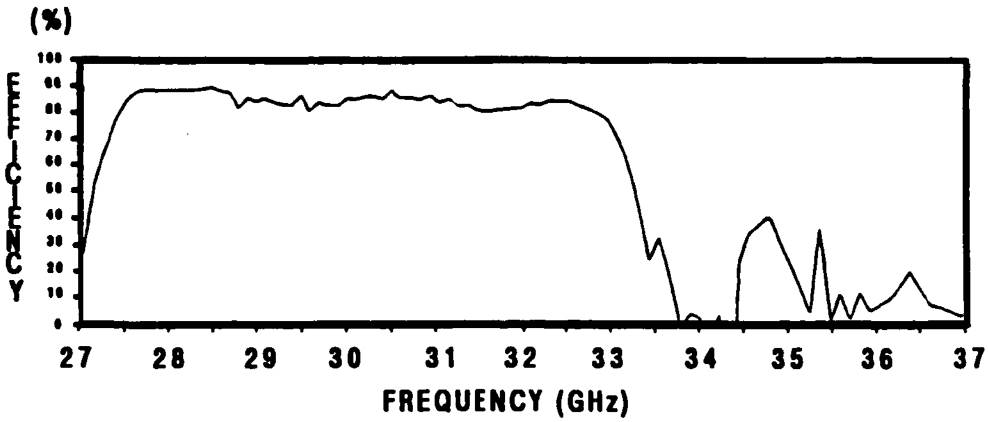


FIGURE 21. Passive Two-Port Transfer Efficiency of Symmetric Cubic Turnstile Junction, Design #1.

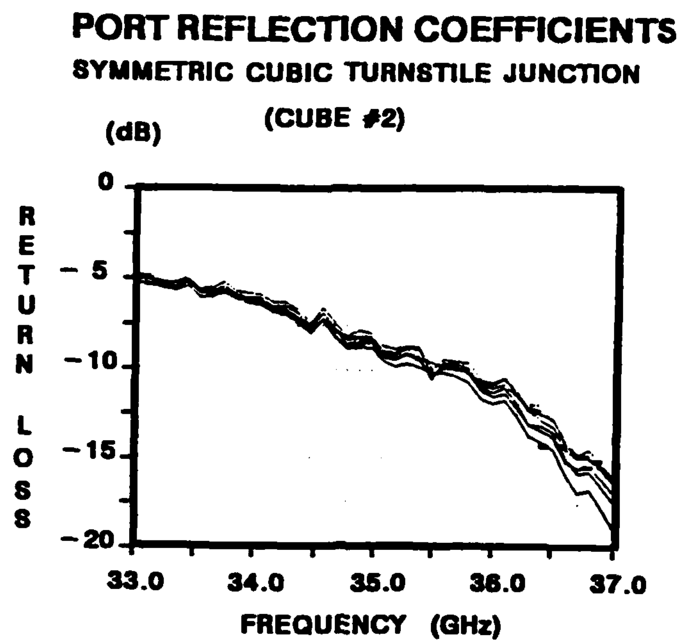


FIGURE 22(a). Magnitude of Reflection Coefficient Data for Symmetric Cubic Turnstile Junction, Design #2.

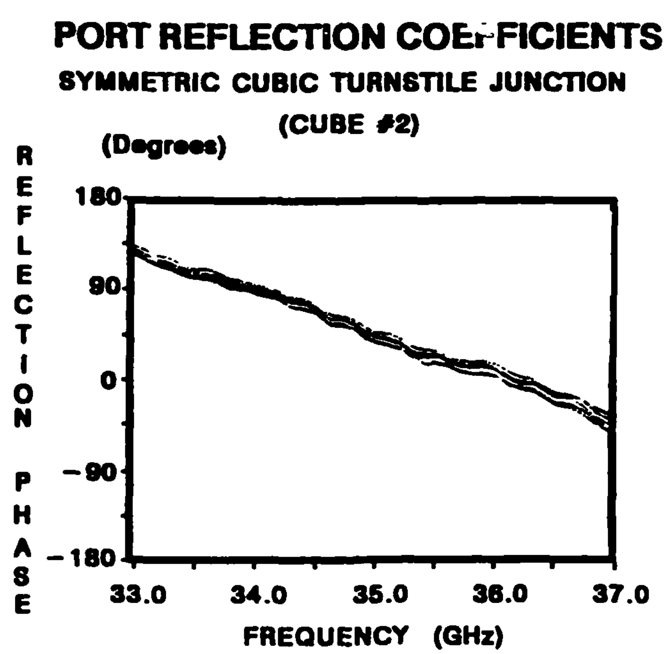


FIGURE 22(b). Phase of Reflection Coefficient Data for Symmetric Cubic Turnstile Junction, Design #2.

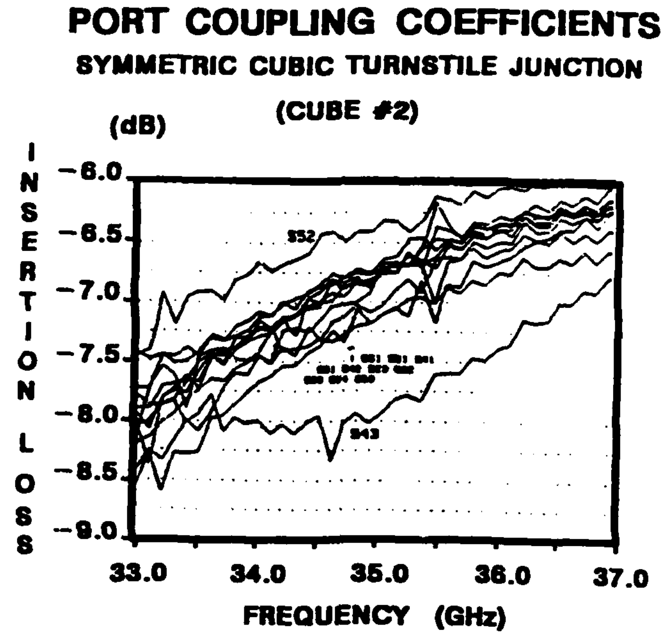


FIGURE 23(a). Magnitude of Adjacent Port Coupling Coefficient Data for Symmetric Cubic Turnstile Junction, Design #2.

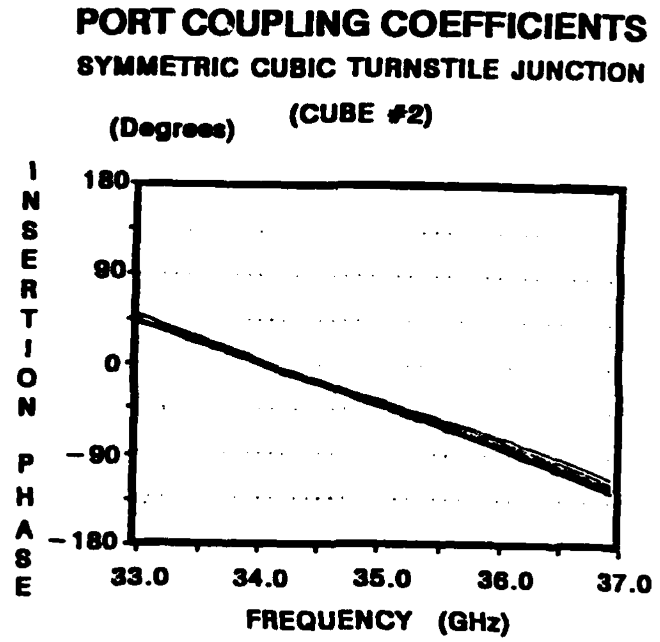


FIGURE 23(b). Phase of Adjacent Port Coupling Coefficient Data for Symmetric Cubic Turnstile Junction, Design #2.

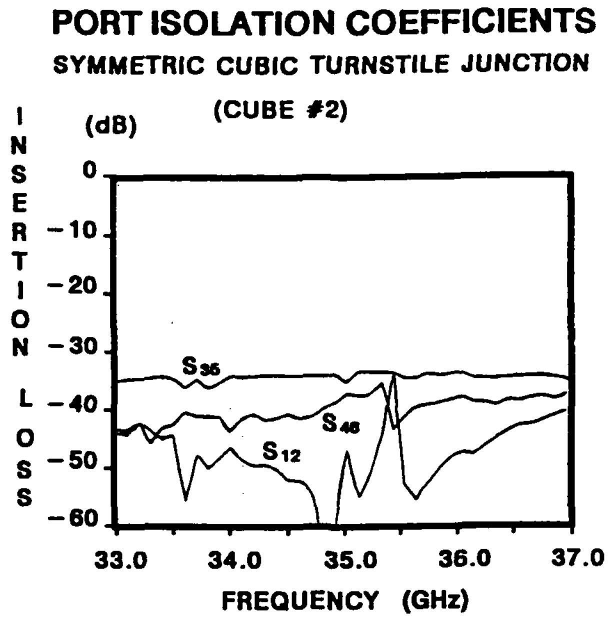


FIGURE 24. Magnitude of Cross-Port Coupling Coefficient Data for Symmetric Cubic Turnstile Junction, Design #2.

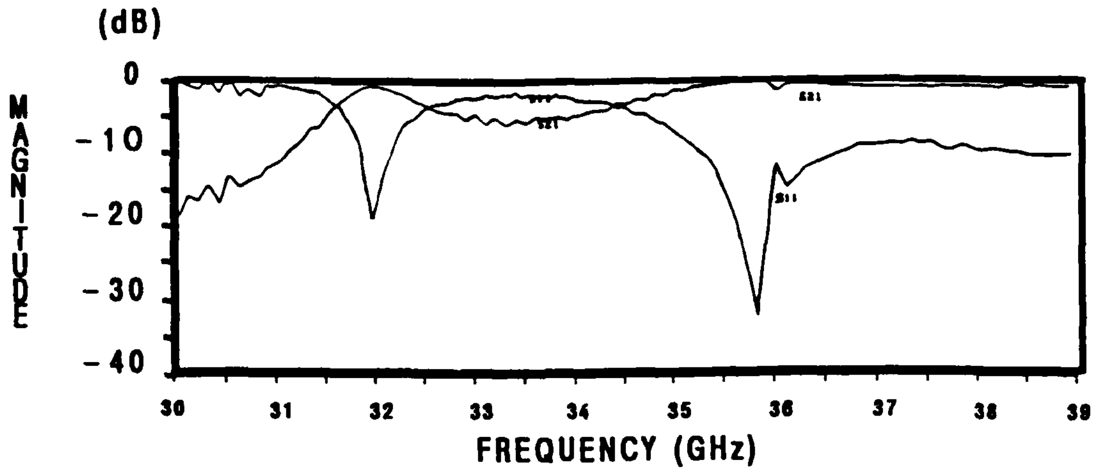


FIGURE 25. Passive Two-Port S-Parameter Data for Symmetric Cubic Turnstile Junction, Design #2.

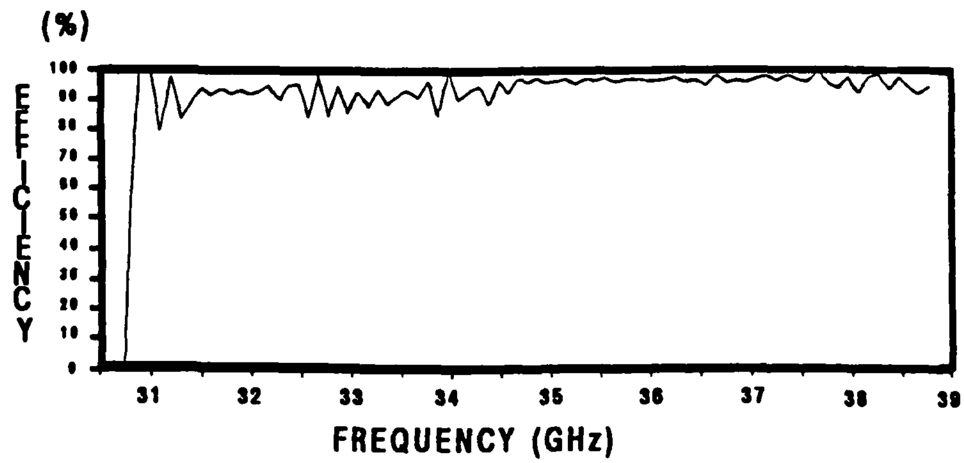


FIGURE 26. Measured Transfer Efficiency Data for Symmetric Cubic Turnstile Junction, Design #2.

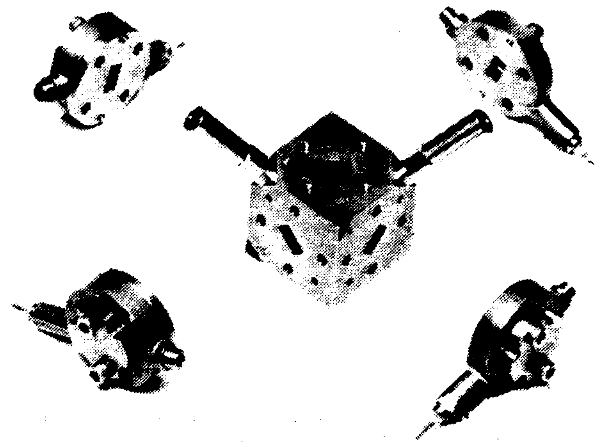


FIGURE 27. Turnstile Junction of Design #3 With Oscillator Modules.

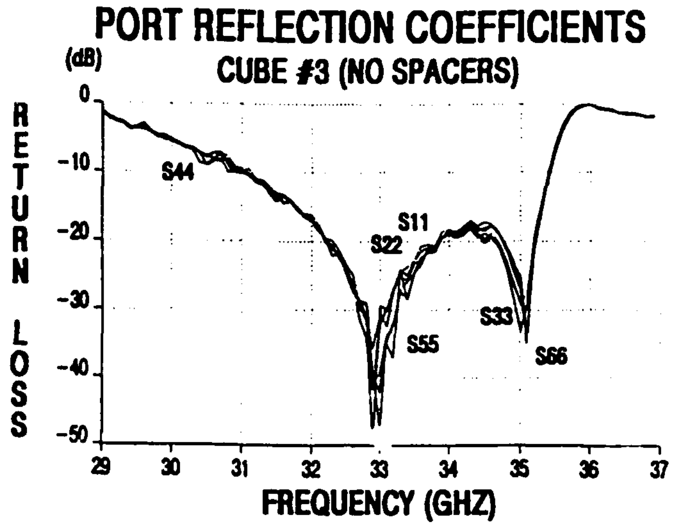


FIGURE 28(a). Magnitude of Reflection Coefficient Data for Symmetric Cubic Turnstile Junction, Design #3.

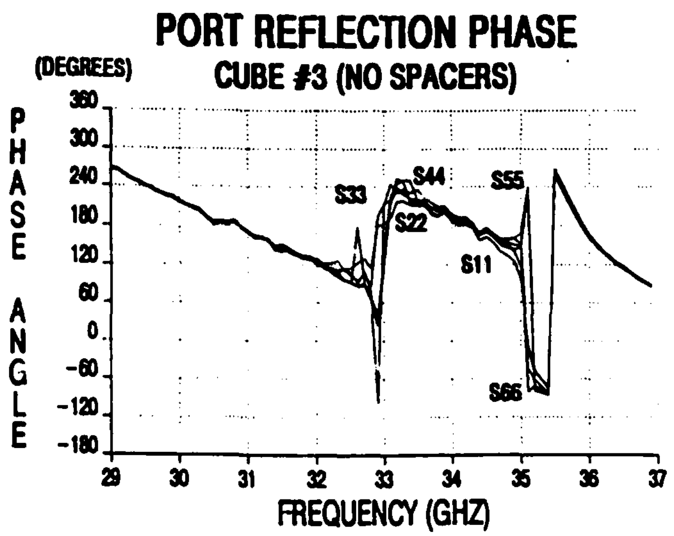


FIGURE 28(b). Phase of Reflection Coefficient Data for Symmetric Cubic Turnstile Junction, Design #3.

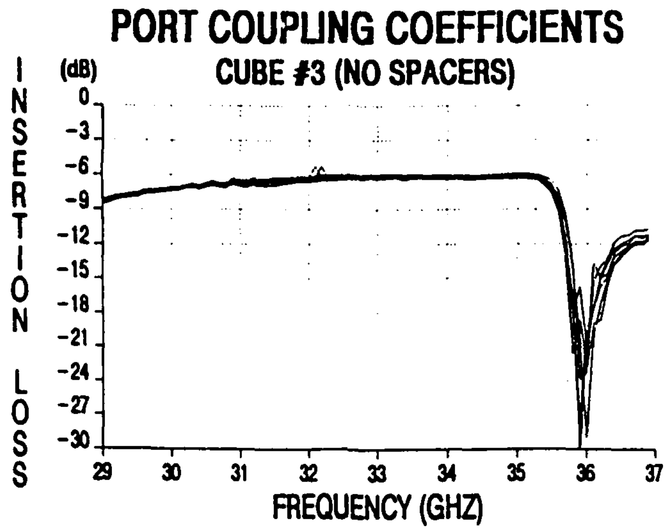


FIGURE 29. Adjacent Port Coupling Data for Symmetric Cubic Turnstile Junction, Design #3.

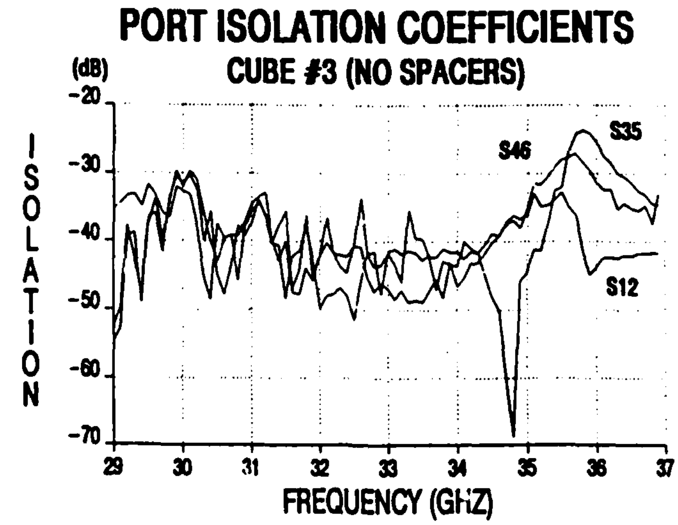


FIGURE 30. Cross-Port Isolation Data for Symmetric Cubic Turnstile Junction, Design #3.

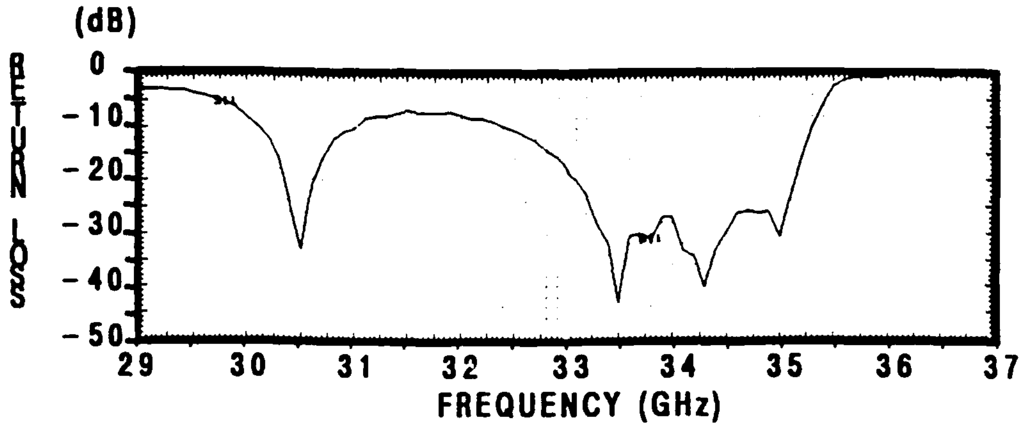


FIGURE 31. Passive Two-Port S11 Data for Symmetric Cubic Turnstile Junction, Design #3 (No Spacers).

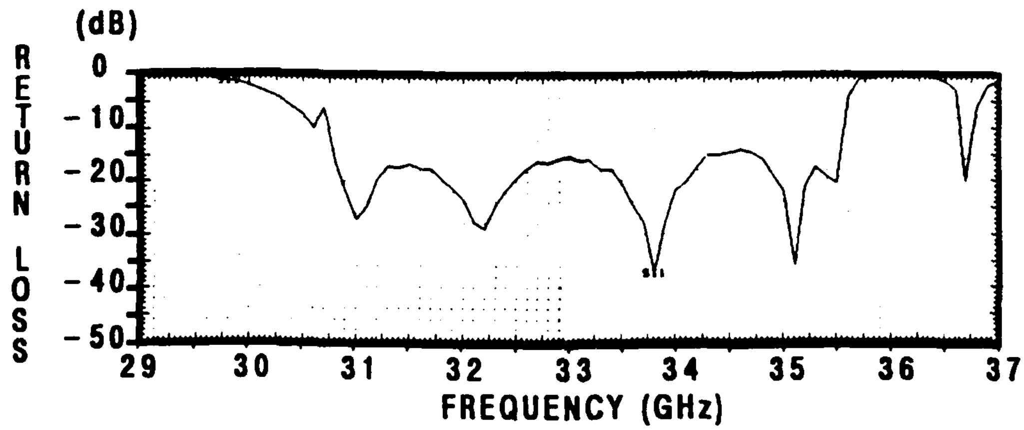


FIGURE 32. Passive Two-Port S11 Data for Symmetric Cubic Turnstile Junction, Design #3 (0.150-Inch Spacers).

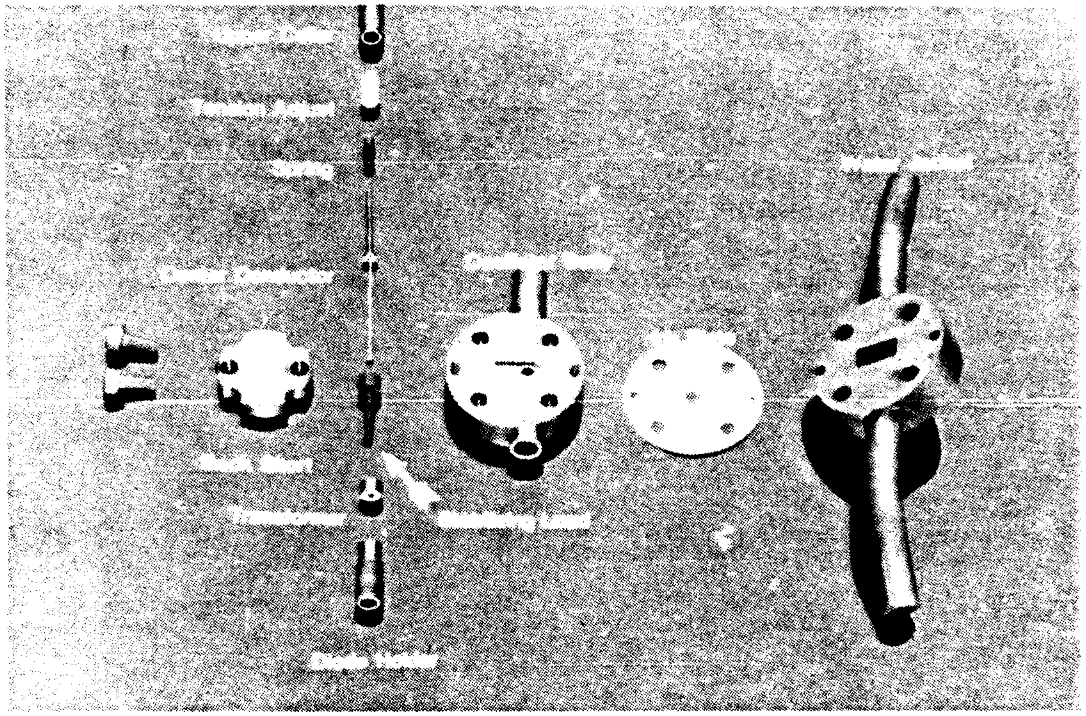


FIGURE 33. Ka-Band Oscillator.

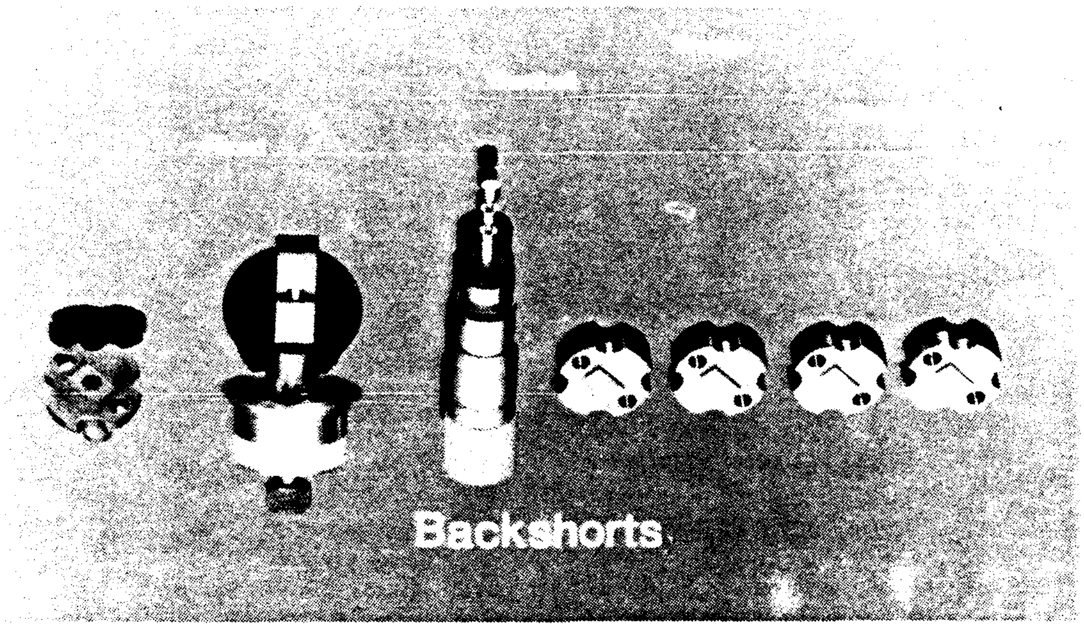


FIGURE 34. Waveguide Backshort Designs.

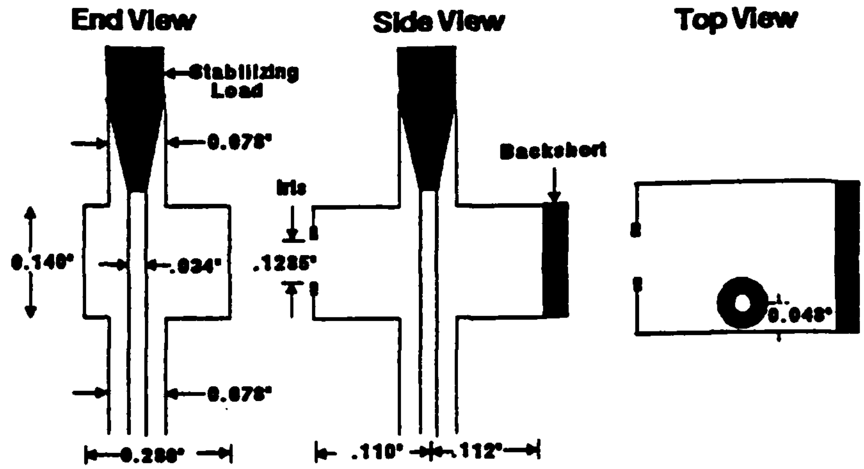


FIGURE 35. Sketch of Cavity Configuration for Ka-Band Oscillators.

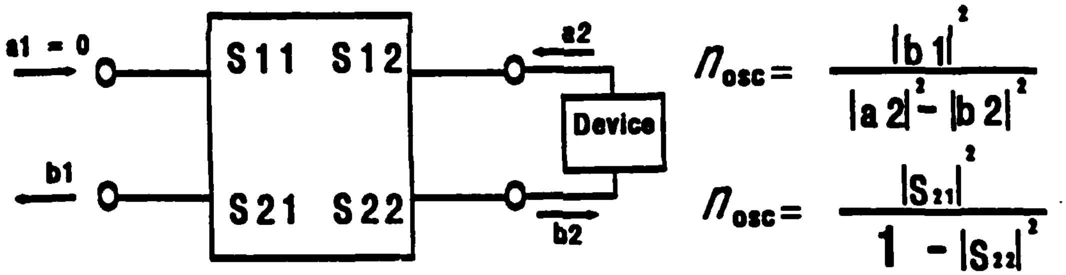


FIGURE 36. Wave Diagram Explaining Oscillator Transfer Efficiency.

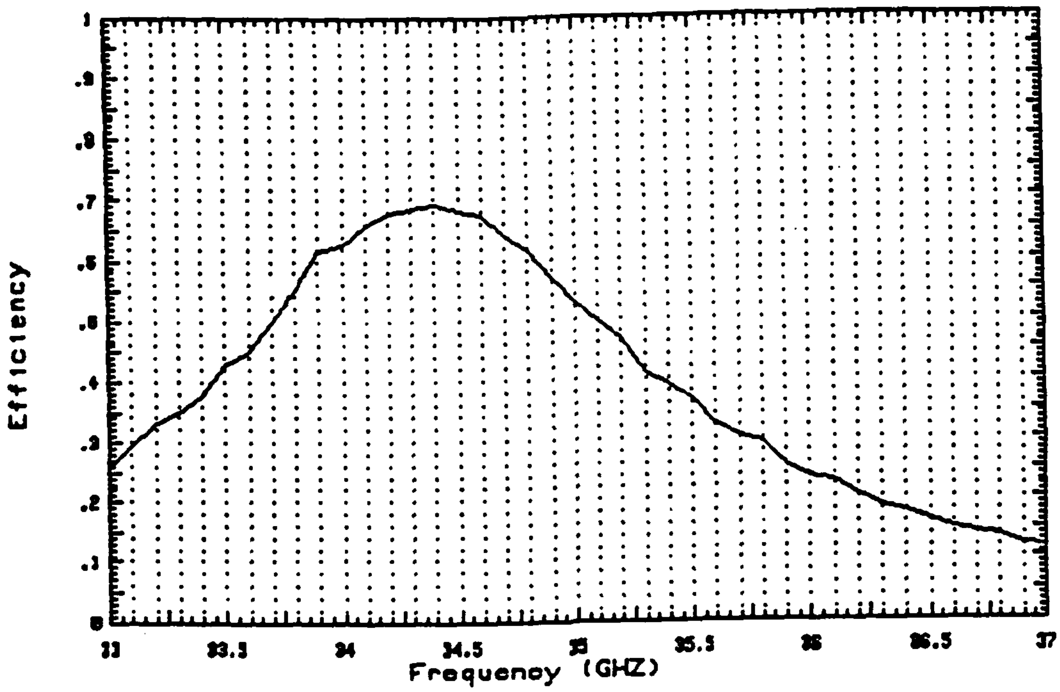


FIGURE 37. Transfer Efficiency Versus Frequency Plot.

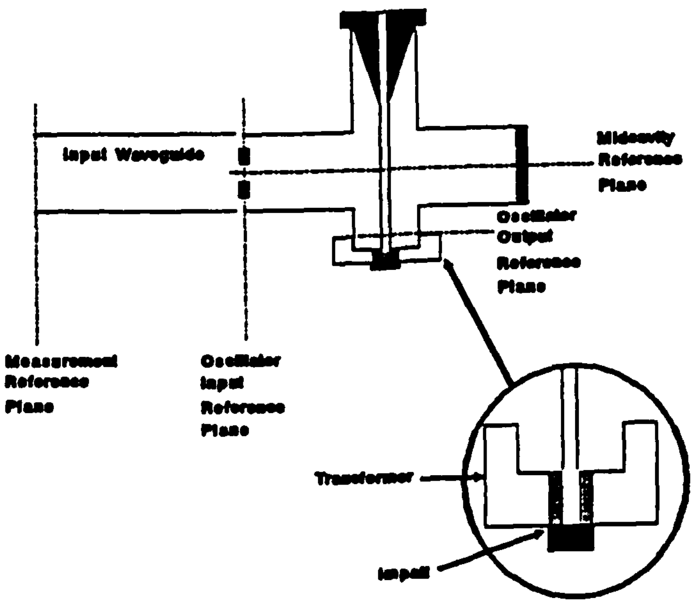


FIGURE 38. S-Parameter Reference Plane Definition.

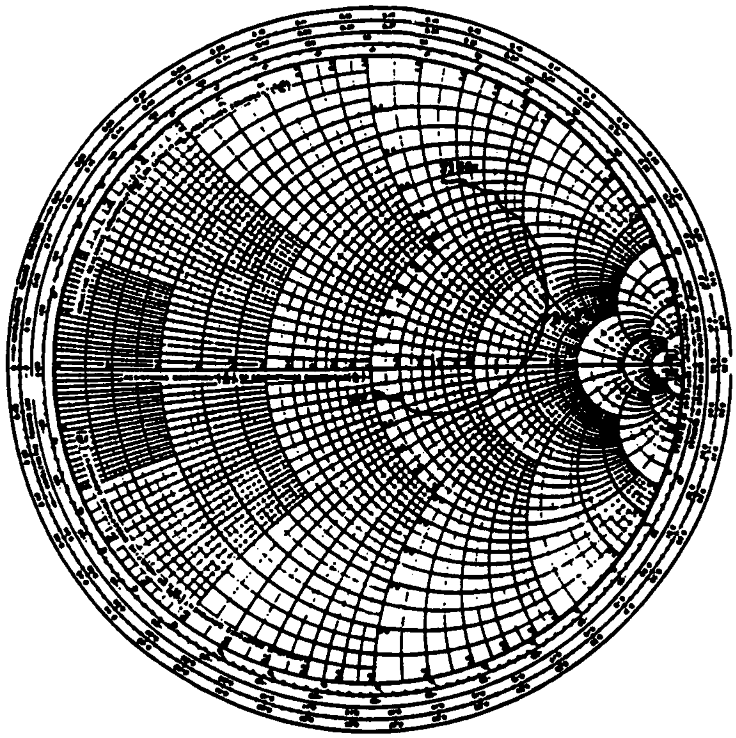


FIGURE 39. Smith Chart Plot of Midcavity Impedance.

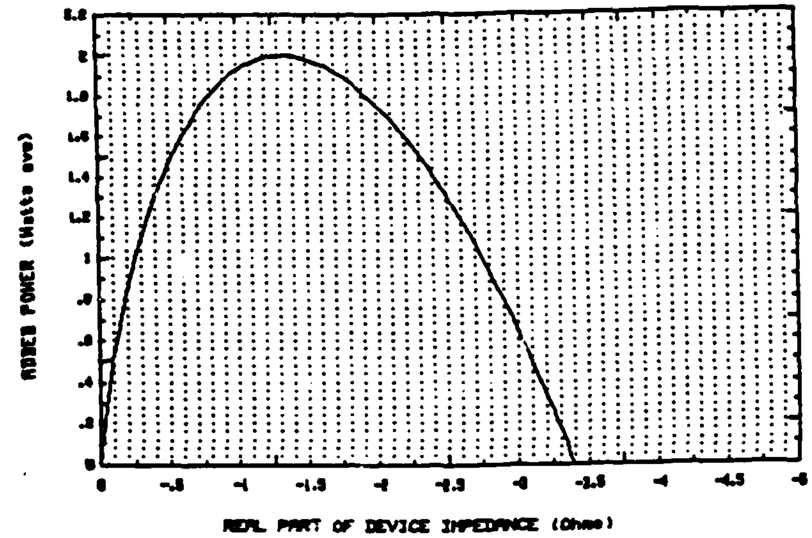


FIGURE 40. Added-Power-Versus-Resistance Plot for Ka-Band Gallium-Arsenide Double-Drift IMPATT.

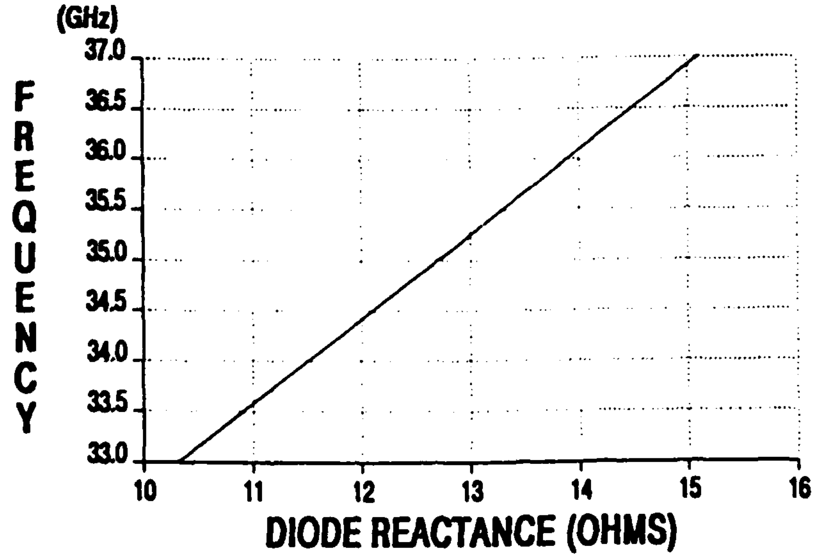


FIGURE 41. Oscillation-Frequency-Versus-Reactance Plot for Ka-Band Gallium-Arsenide Double-Drift IMPATT.

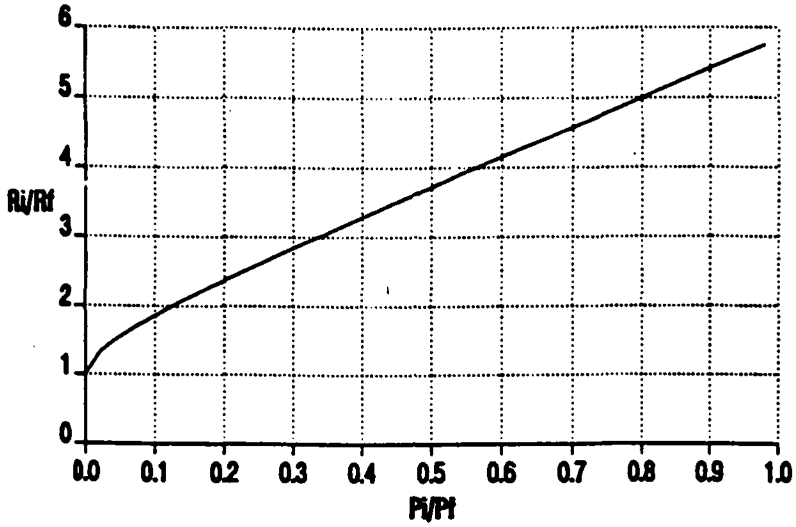


FIGURE 42. Injection-Locking Compensation.

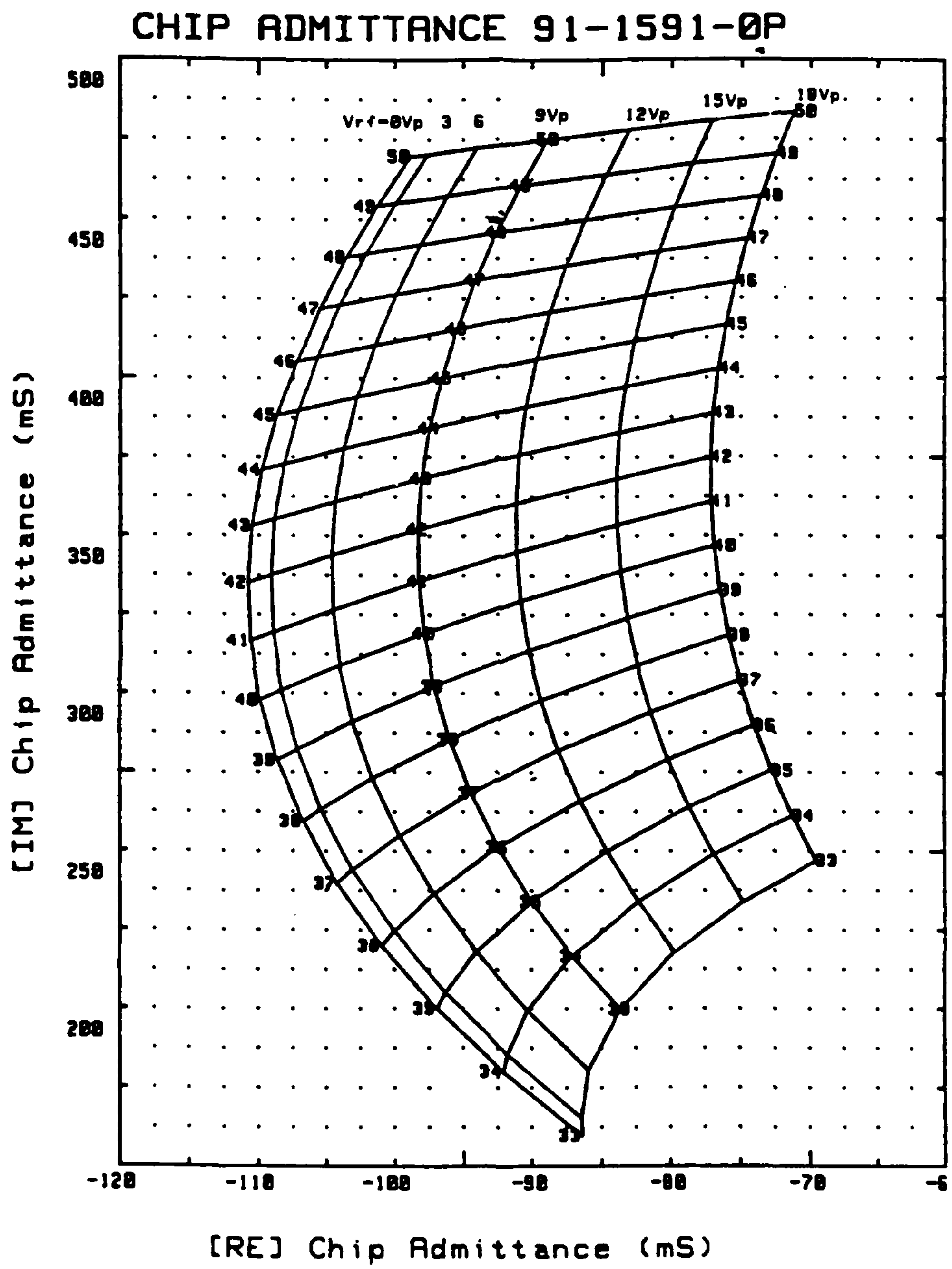


FIGURE 43. Predicted Diode Admittance at the Chip Reference Plane.

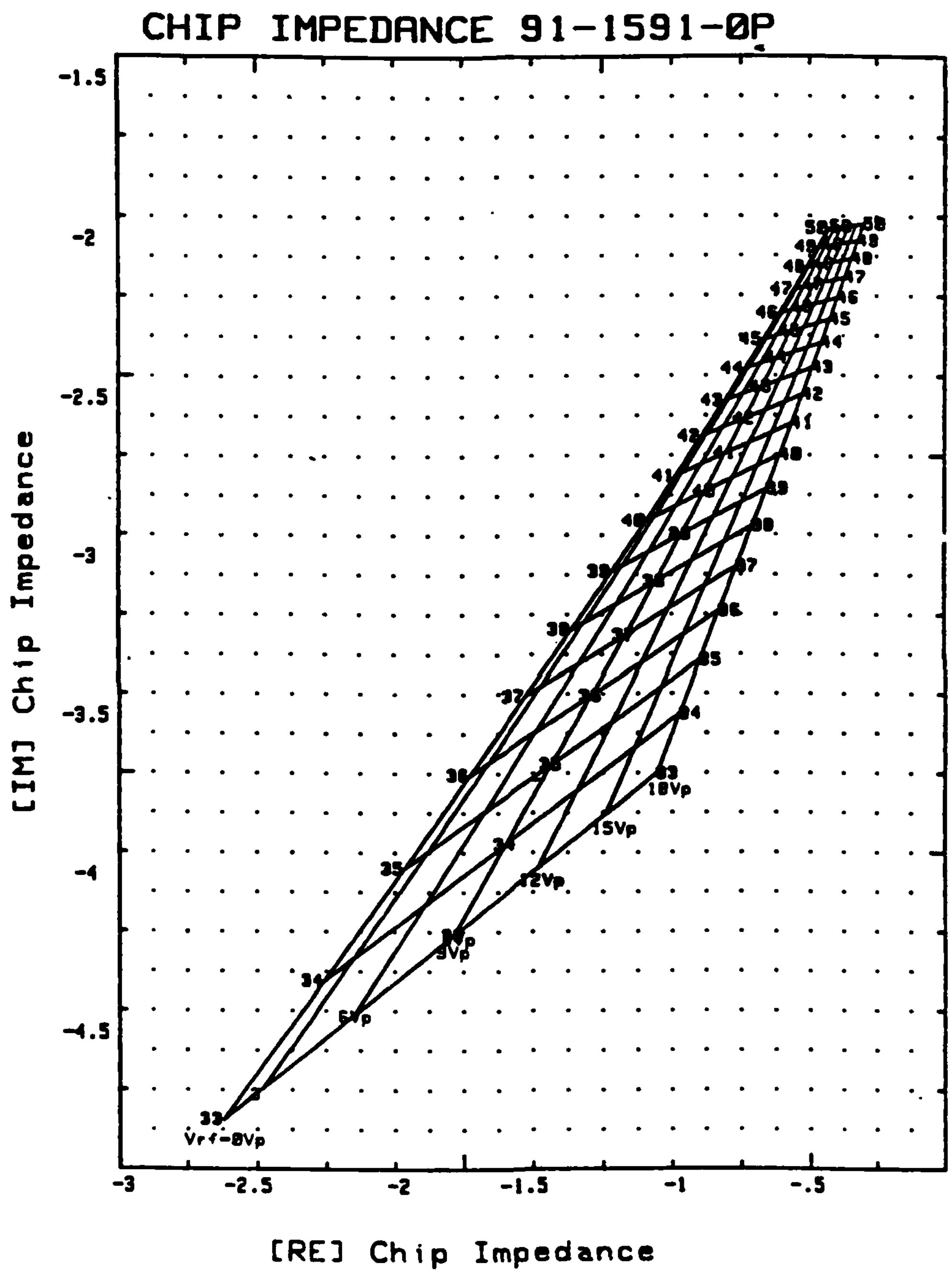


FIGURE 44. Predicted Diode Impedance at the Chip Reference Plane.

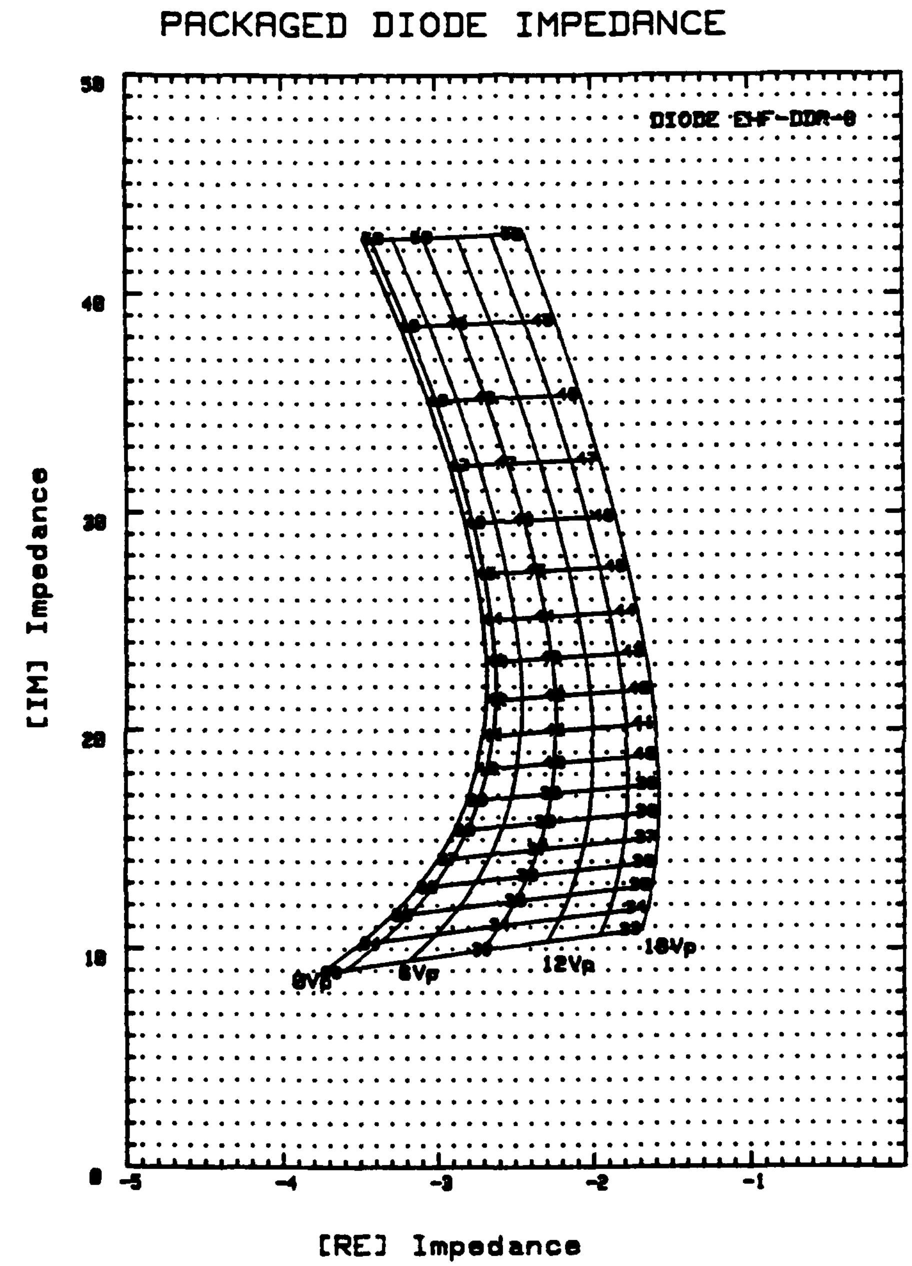


FIGURE 45. Predicted Diode Admittance at the Package Reference Plane.

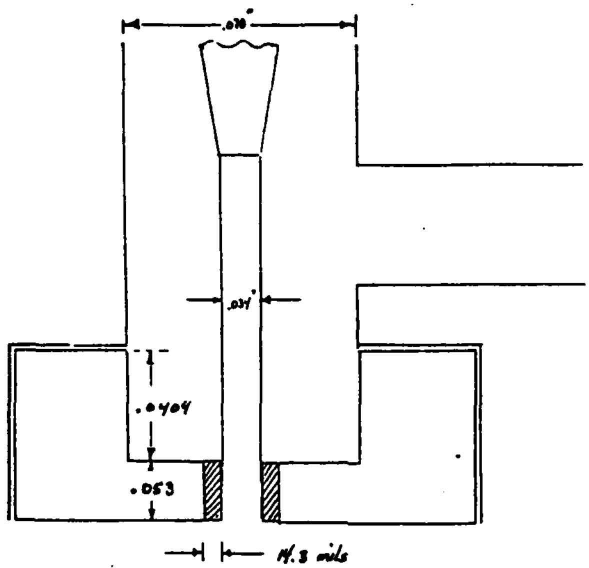


FIGURE 46. Turnstile Oscillator Transformer Design.

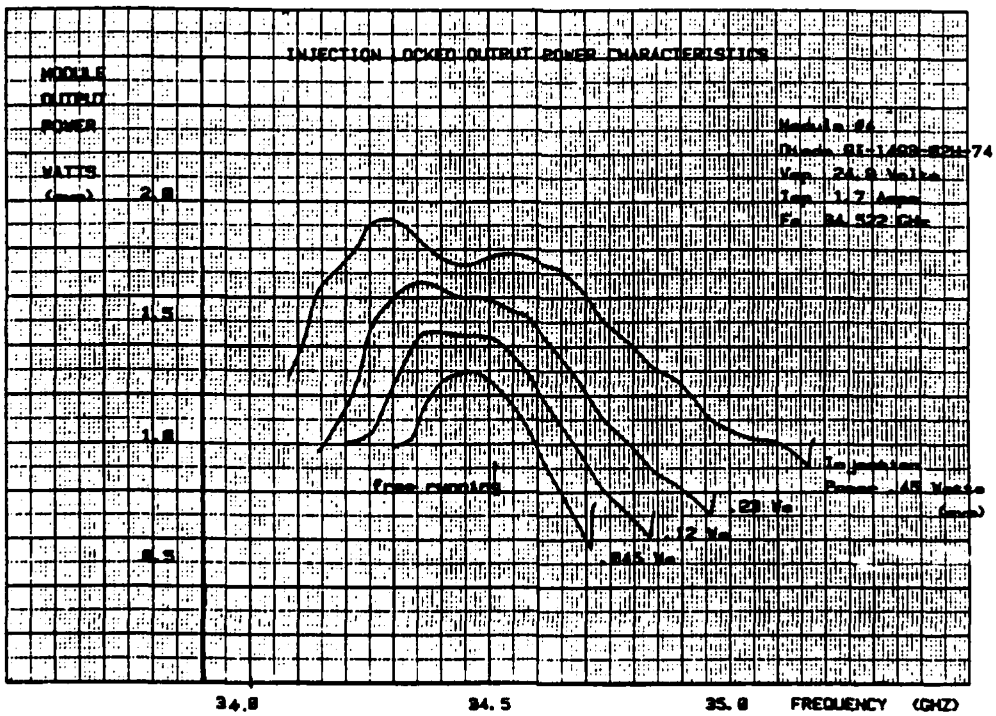


FIGURE 47. Plot of Injection-Locked Output Power Versus Frequency for Various Levels of Injection Input Power.

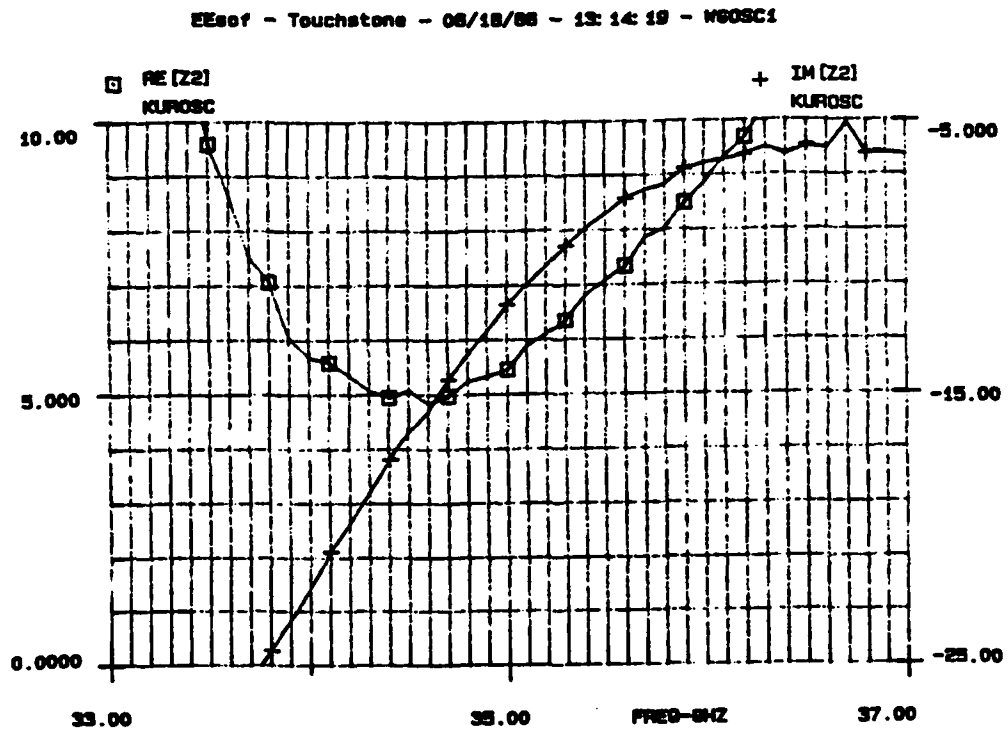


FIGURE 48. Circuit Impedance at the Diode Cap for Selected Transformer Design.

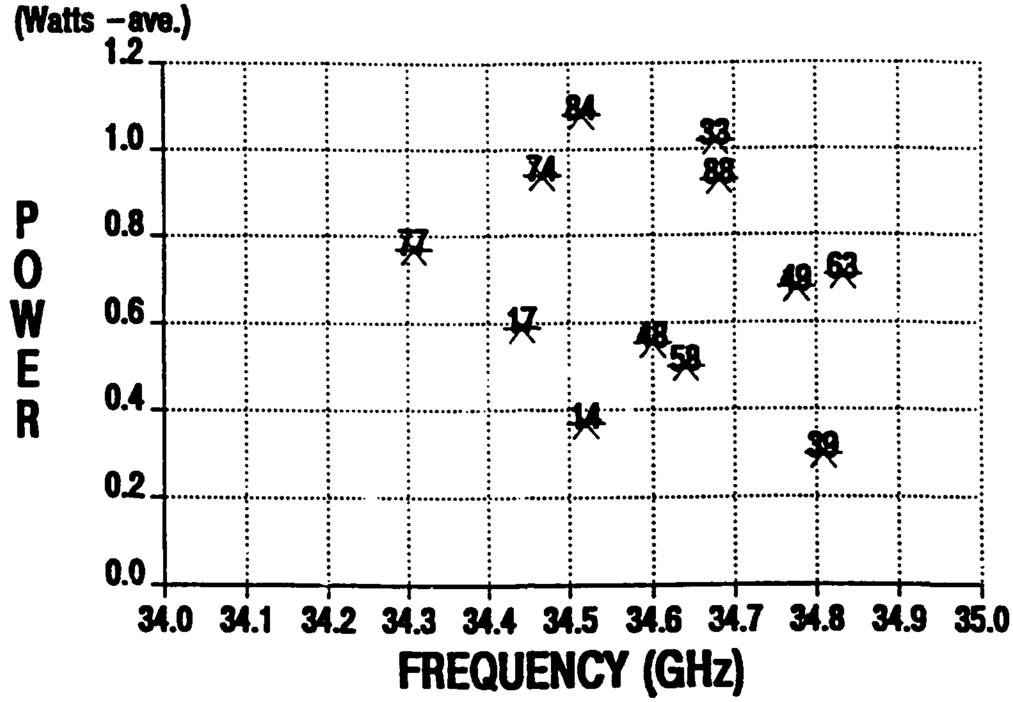


FIGURE 49. Typical Distribution of Ka-Band .MPATT Free-Running Output Power Versus Oscillation Frequency.

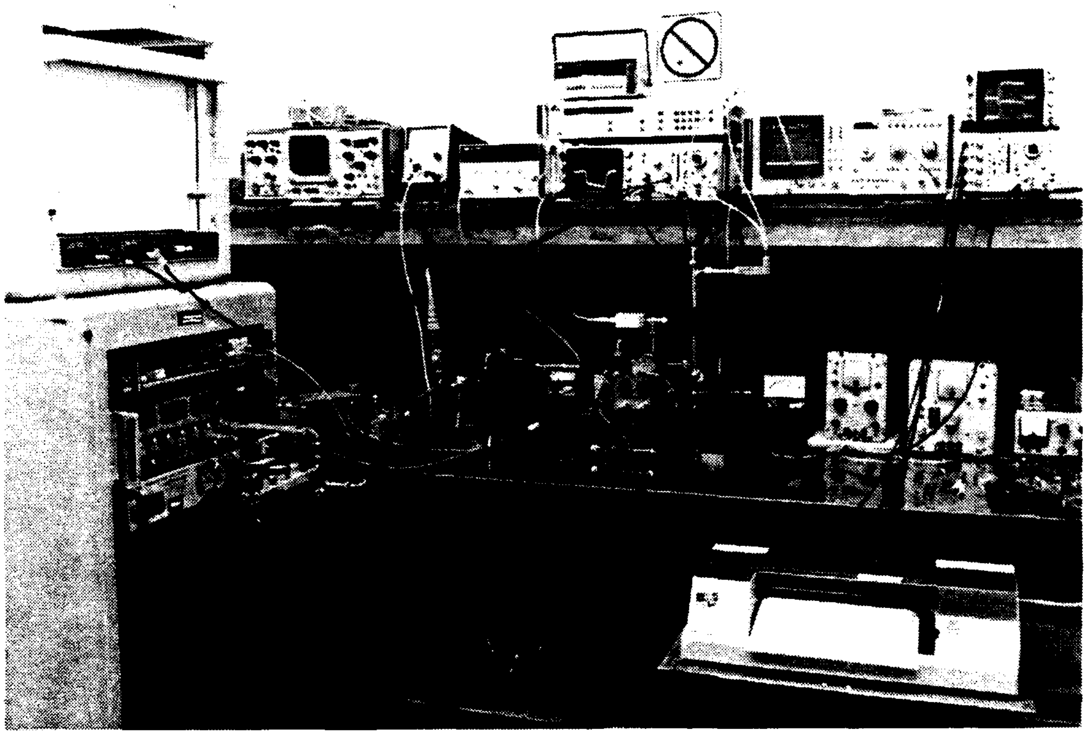


FIGURE 50. Ka-Band Reflectometer System.

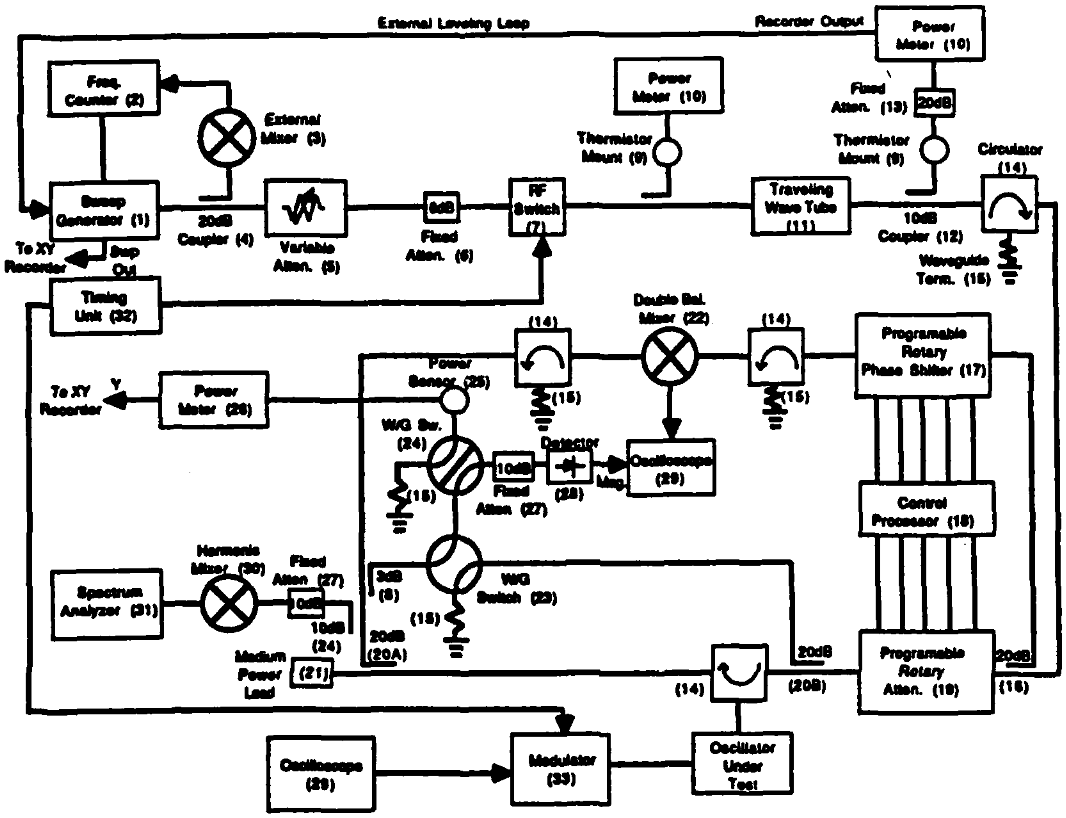


FIGURE 51. Block Diagram of Ka-Band Reflectometer System.

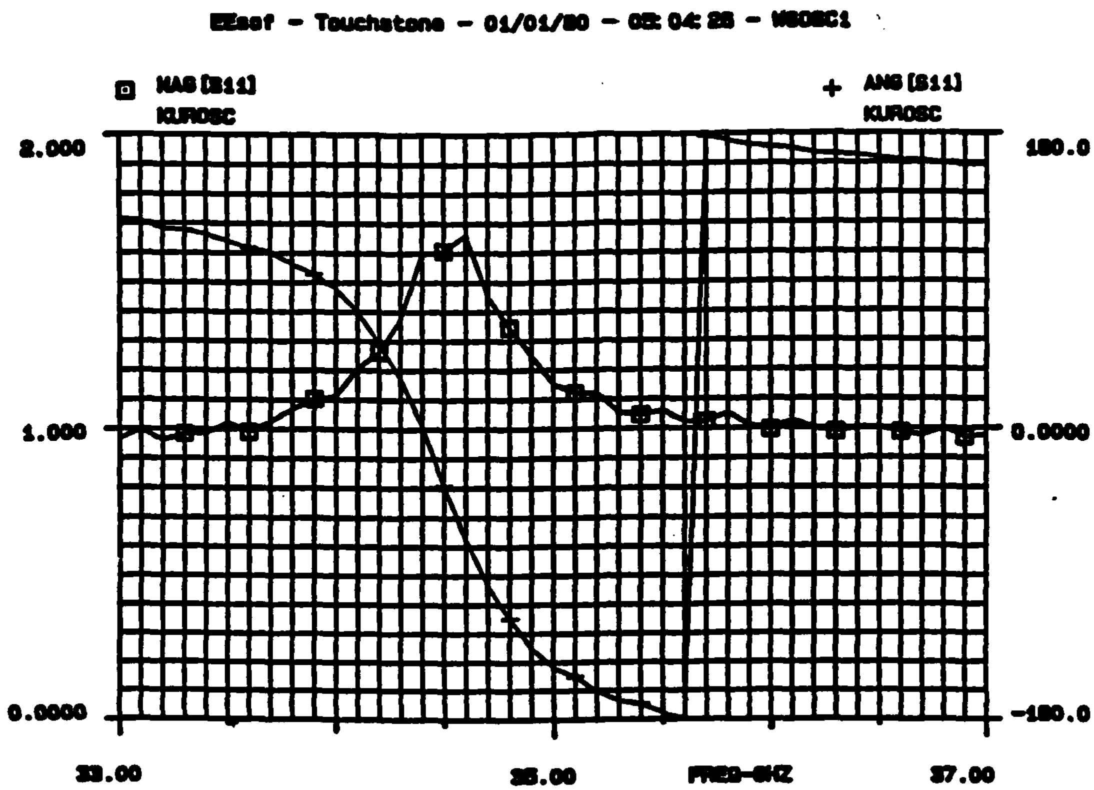


FIGURE 52. Predicted Input Reflection Coefficient (Magnitude and Phase) of an IMPATT Oscillator Module.

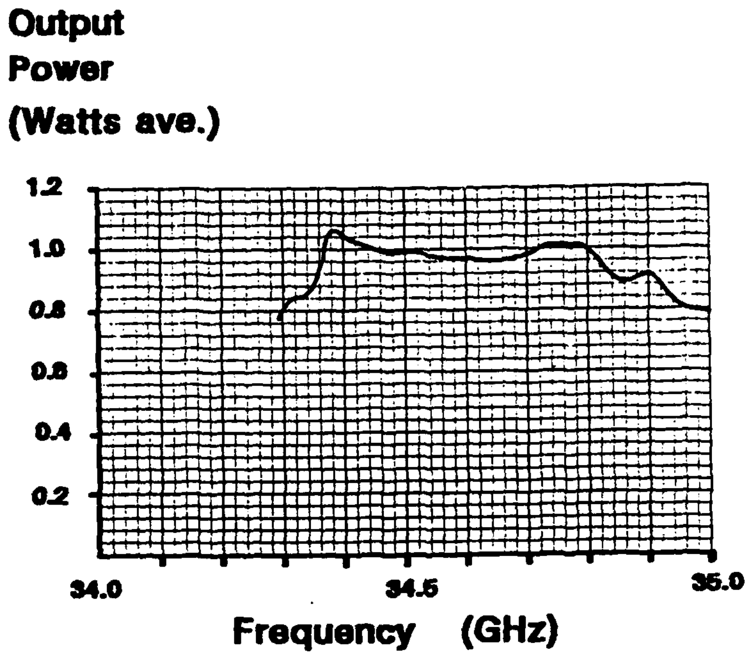


FIGURE 53(a). Output Power Versus Injection Frequency, Module #2.

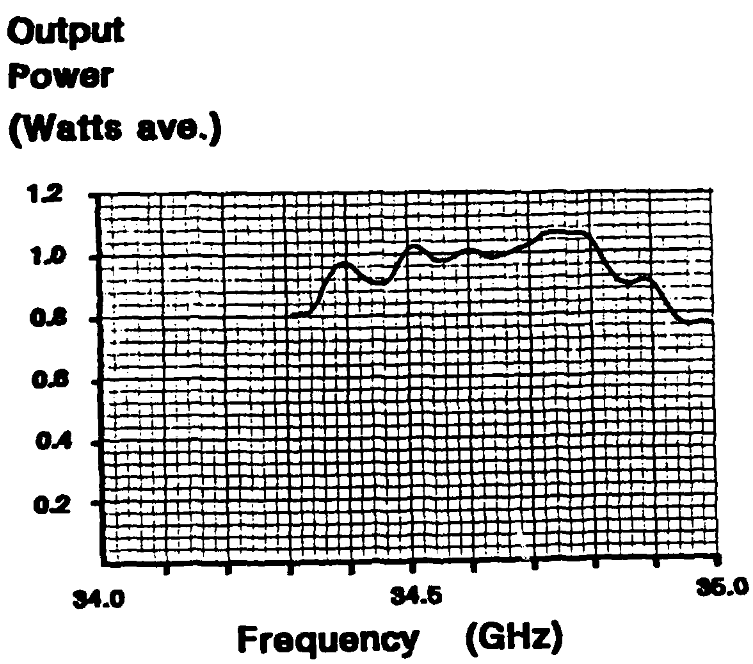


FIGURE 53(b). Output Power Versus Injection Frequency, Module #3.

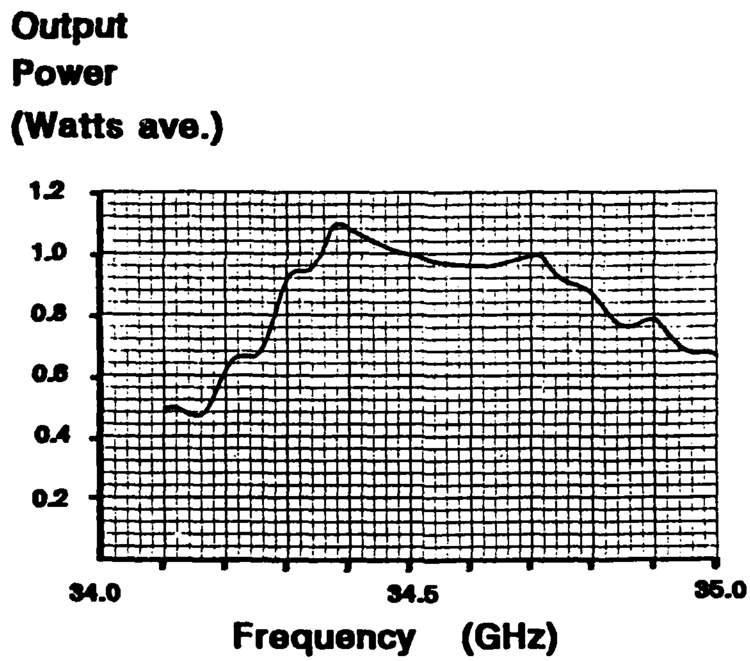


FIGURE 53(c). Output Power Versus Injection Frequency, Module #4.

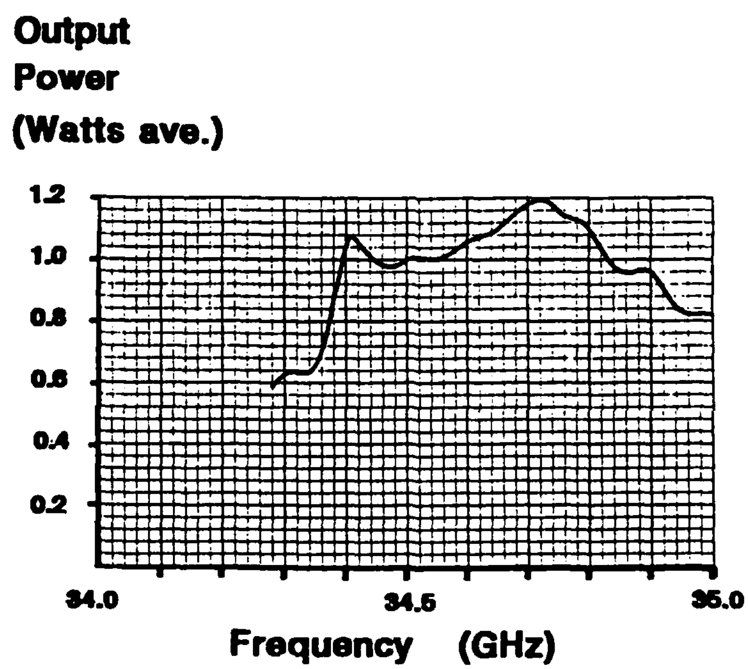


FIGURE 53(d). Output Power Versus Injection Frequency, Module #5.

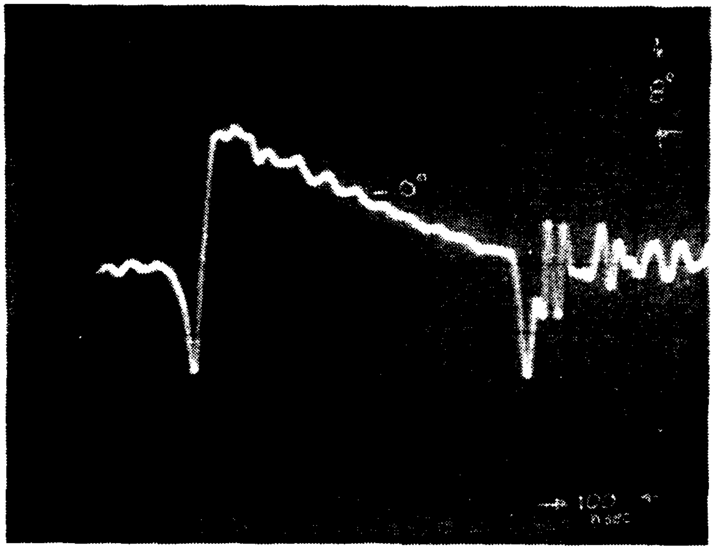


FIGURE 54(a). Reflected Phase (From Phase Detector Output), Module #2.

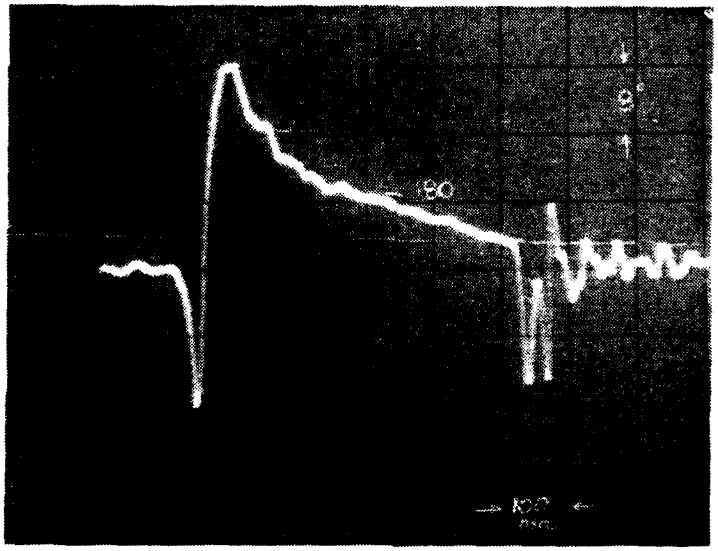


FIGURE 54(b). Reflected Phase (From Phase Detector Output), Module #3.

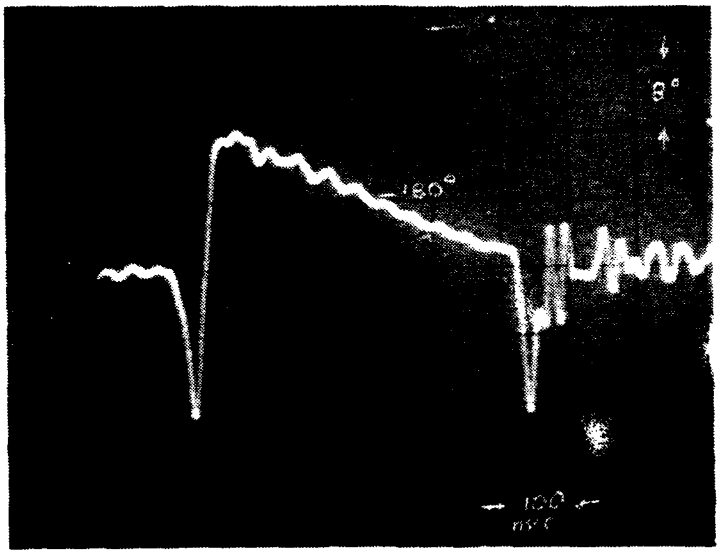


FIGURE 54(c). Reflected Phase (From Phase Detector Output), Module #4.

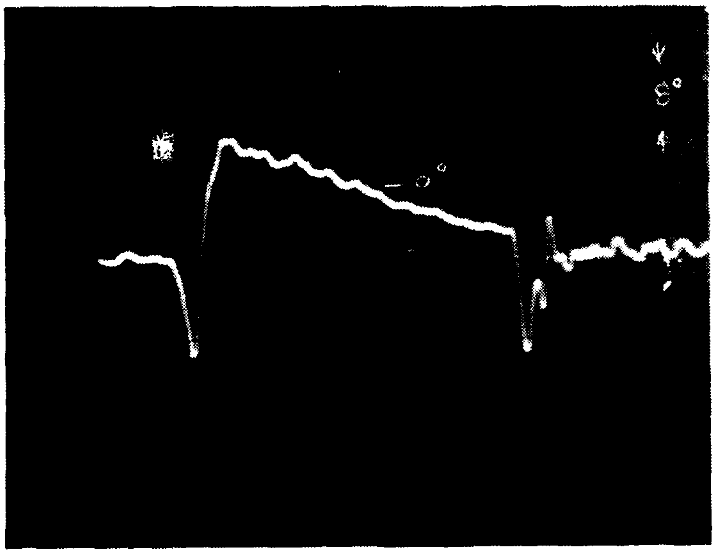


FIGURE 54(c). Reflected Phase (From Phase Detector Output), Module #5.

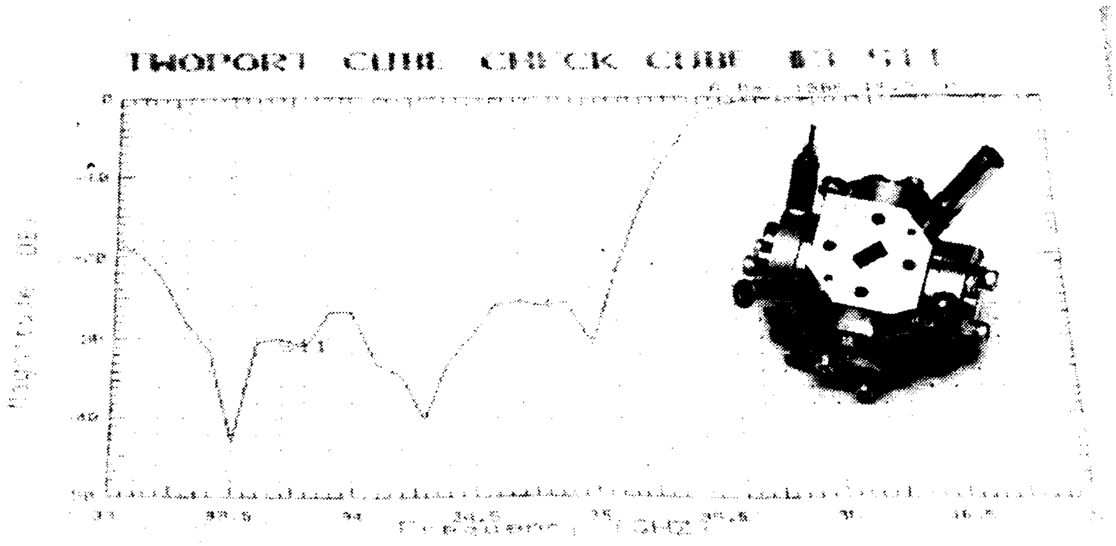


FIGURE 55. Turnstile Junction and Oscillator Module (Integrated Configuration).

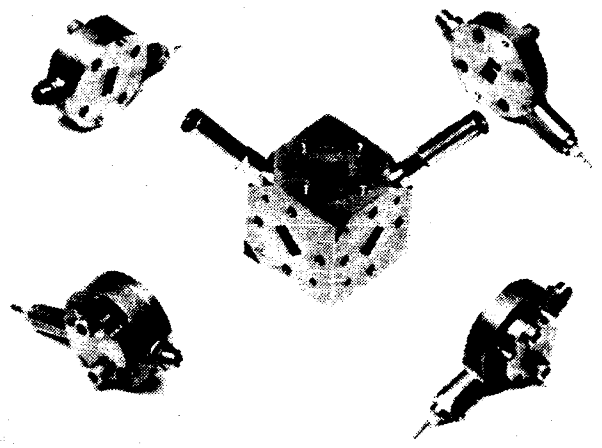


FIGURE 56. Turnstile Junction and Oscillator Modules (Exploded View).

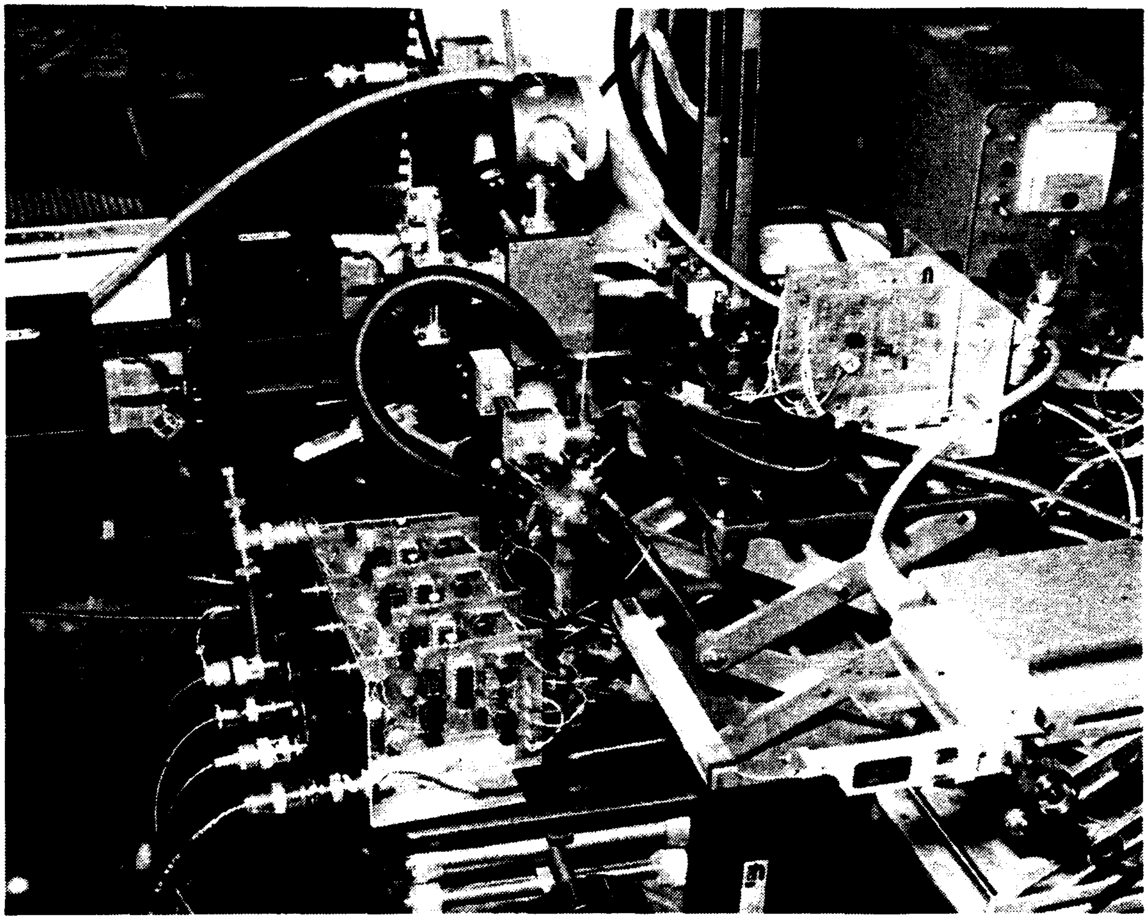


FIGURE 57. Turnstile Transmission-Mode Test Configuration.

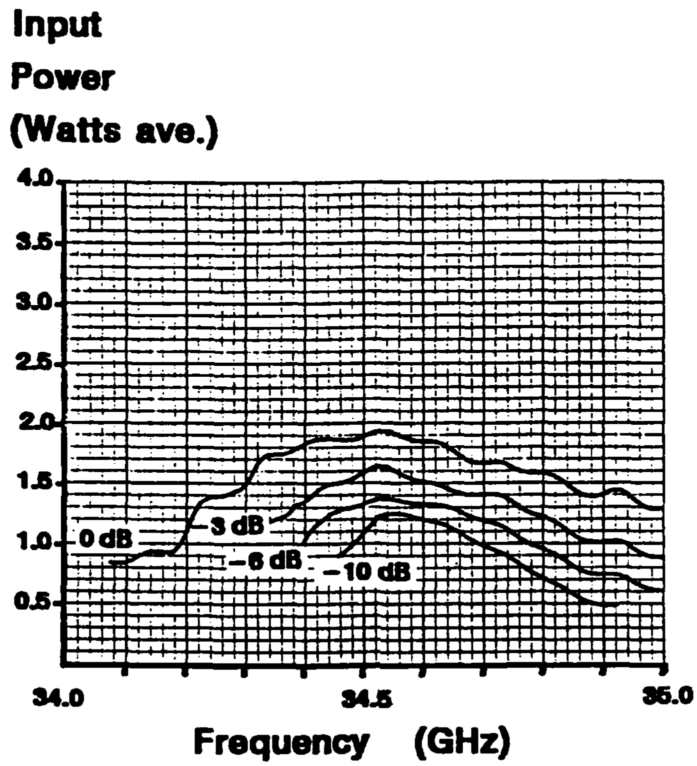


FIGURE 58. Turnstile Input Injection Power Versus Frequency for Various Reflectometer Attenuator Settings.

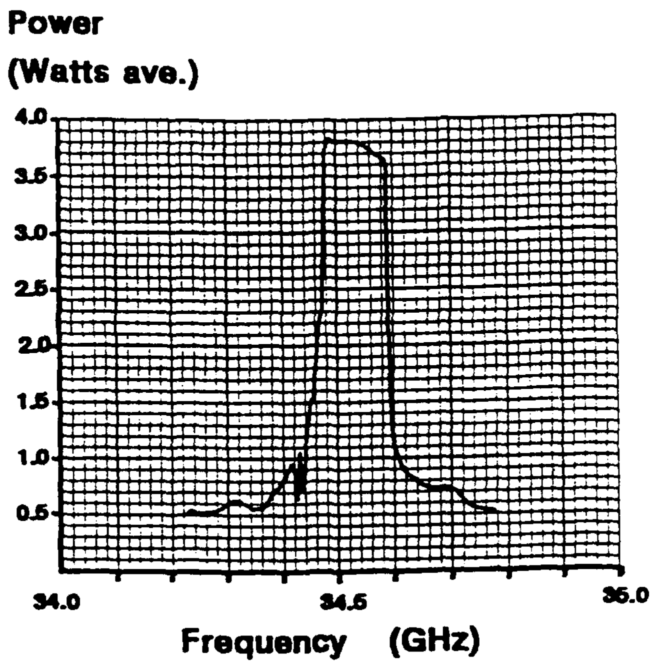


FIGURE 59. Turnstile Combined Output Power Versus Injection Frequency (1.6-Watt Input).

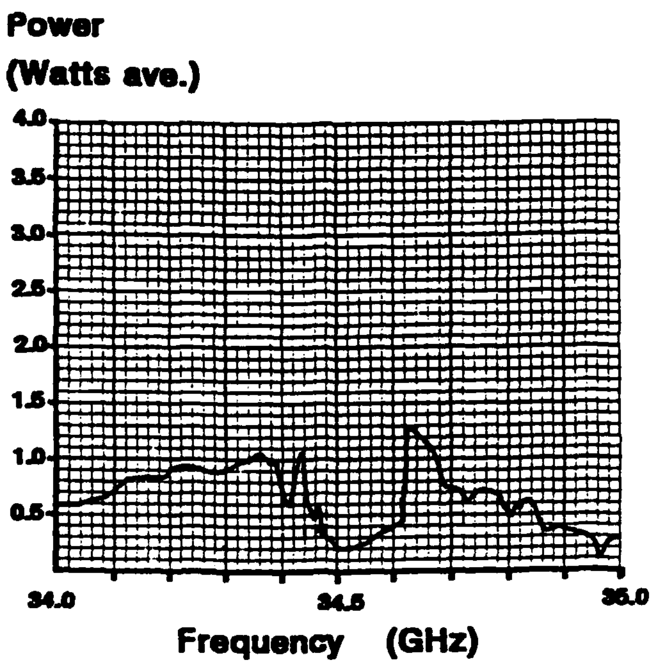


FIGURE 60. Reflected Power Versus Injection Frequency Measured by Reflectometry.

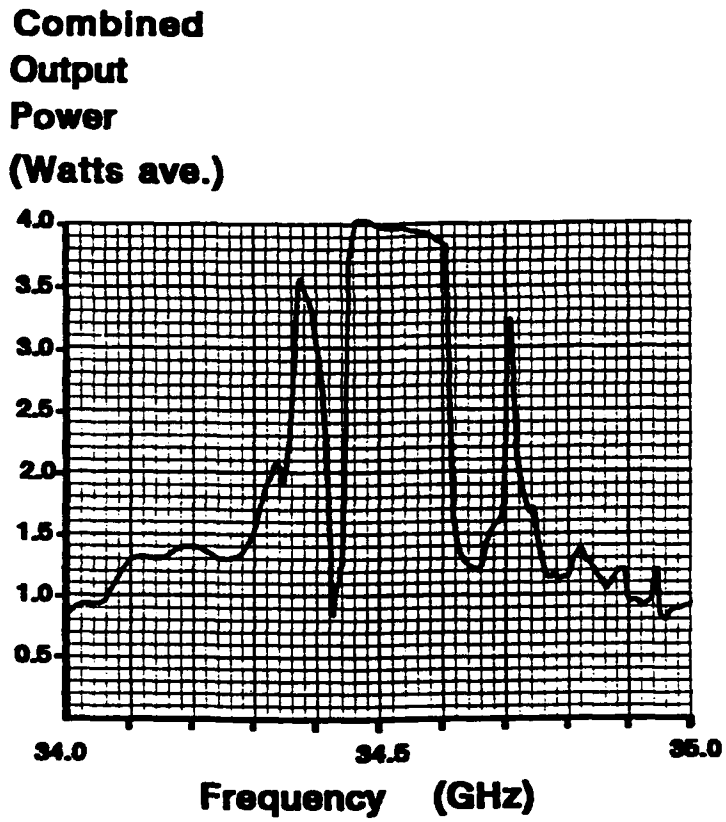


FIGURE 61. Turnstile Combined Output Power Versus Injection Frequency (2.0-Watt Input).

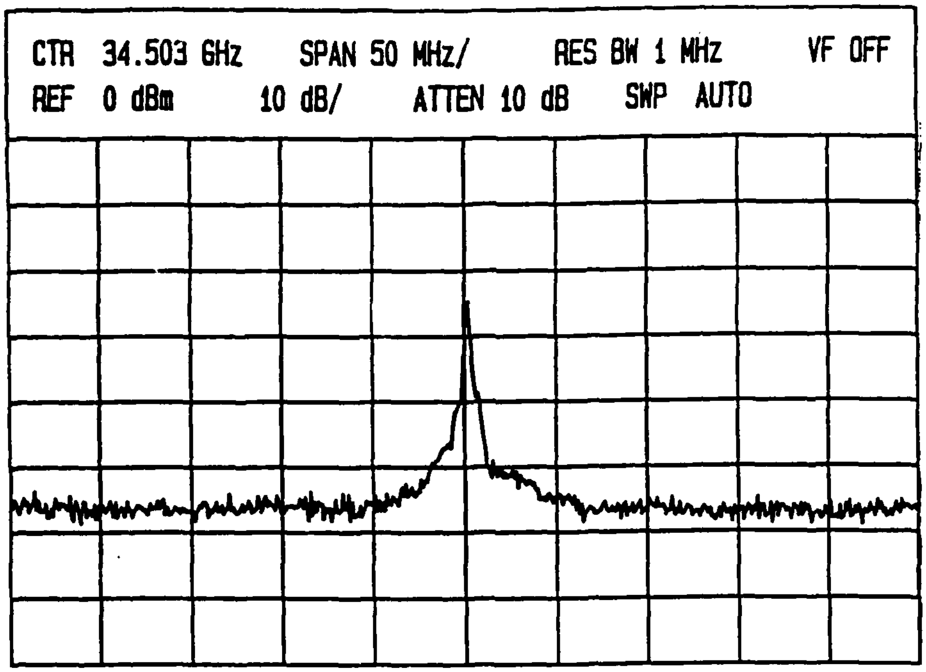


FIGURE 62. Injection-Locked Frequency Spectrum of Turnstile Output (34.5 Gigahertz).

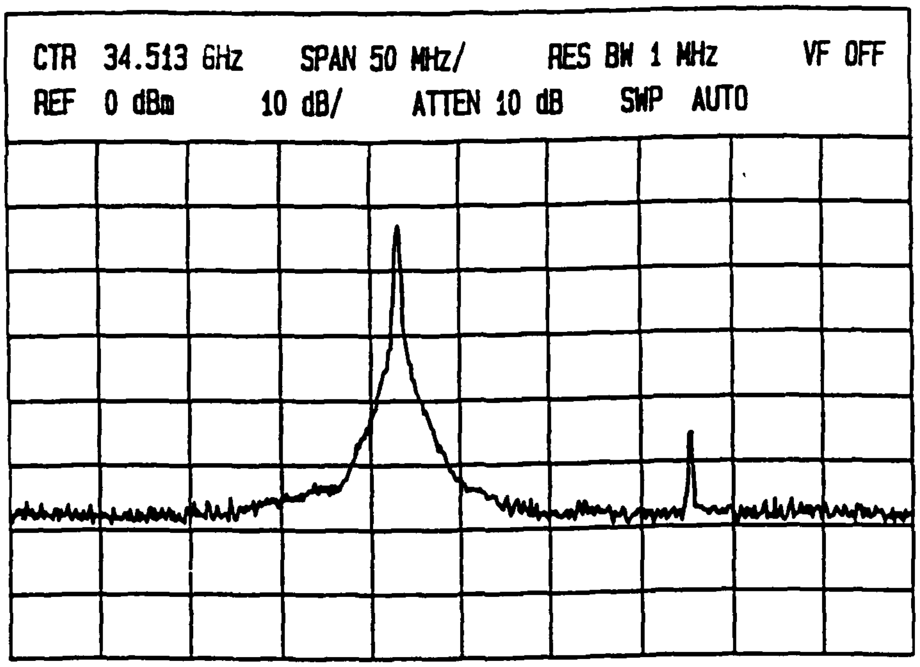


FIGURE 63. Injection-Locked Frequency Spectum of Oscillator Module.

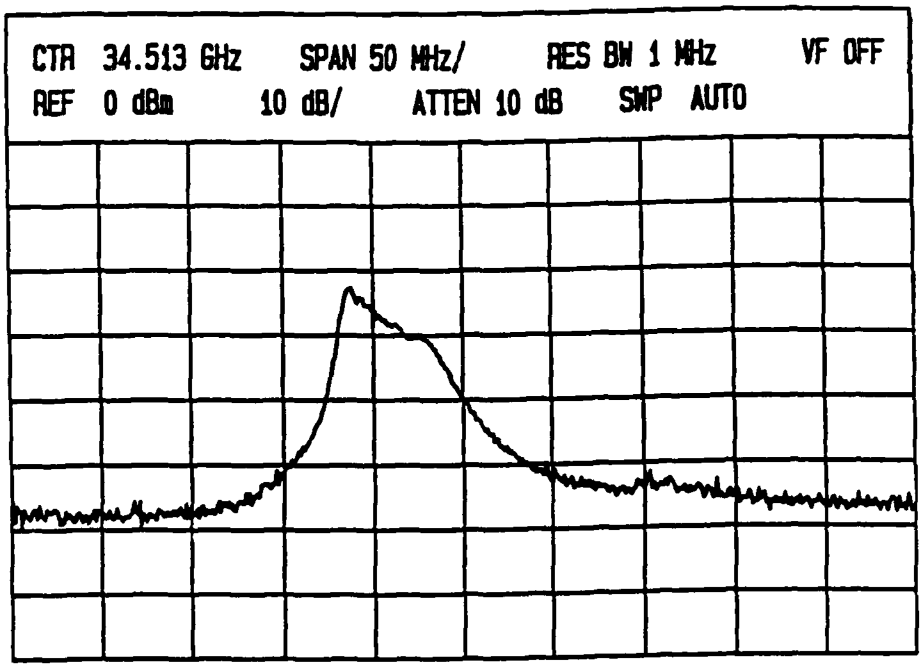


FIGURE 64. Free-Running Frequency Spectrum of Oscillator Module.

Appendix A

S-PARAMETER CONSTRAINT MATRIX OF A SIX-PORT WAVEGUIDE JUNCTION

A description, user instructions, and listing of the computer program for determining the S-parameter constraint matrix of a six-port waveguide junction are presented. The program is written in Pascal and consists of the following segments.

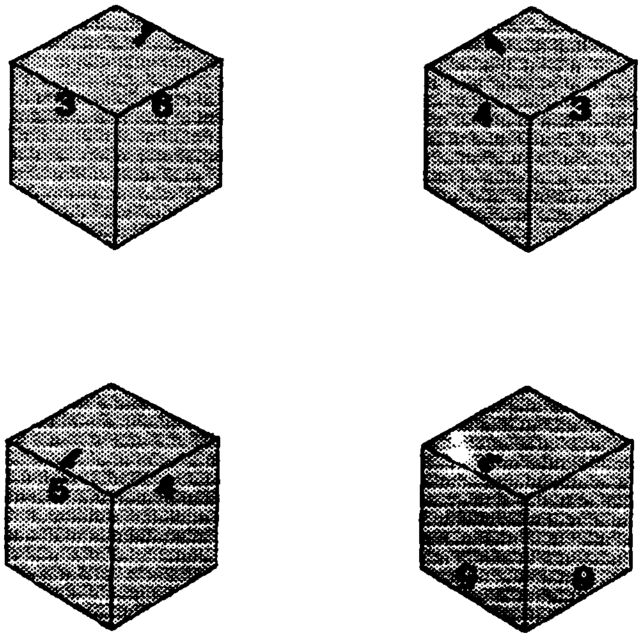
Symmetry	Main program
Inverter	Procedure to find inverse of a matrix
Sgen	Procedure to find inv(M)*S*M
Maintest	Procedure *for finding S-matrix*

The following procedures correspond to symmetry operators
Rotate_about_y_eq_z
Mirror_about_O
Mirror_about_x_eq_y_plane
Mirror_about_x_eq_z_plane
Mirror_about_z_eq_y_plane
Mirror_about_x_y_plane
Mirror_about_x_z_plane
Mirror_about_y_z_plane
Mirror_about_x_eq_minus_y_plane
Mirror_about_x_eq_minus_z_plane
Mirror_about_x_eq_y_plane
Mirror_about_z_eq_neg_y_plane
Rotat_120_about_y_eq_neg_x_eq_z
Rotat_240_about_y_eq_neg_x_eq_z
Rotate_about_x
Rotate_about_y
Rotate_about_z
Rotate_about_y_eq_neg_x
Rotate_about_x_eq_neg_z
Rotate_about_y_eq_x
Rotate_about_x_eq_z
Rotat_120_about_neg_y_eq_x_eq_z
Rotat_240_about_neg_y_eq_x_eq_z
Rotat_120_about_y_eq_x_eq_neg_z

Rotat_240_about_y_eq_x_eq_neg_z
Rotate_120_about_y_eq_x_eq_z
Rotate_240_about_y_eq_x_eq_z
Rotate_about_y_eq_neg_z

Compare	Procedure to check for equivalent matrices
Combiner	Procedure to combine constraints into
resultant S-matrix	
Operator_out	Procedure to print out operator vector
S-matrix_out	Procedure to print out S-matrix

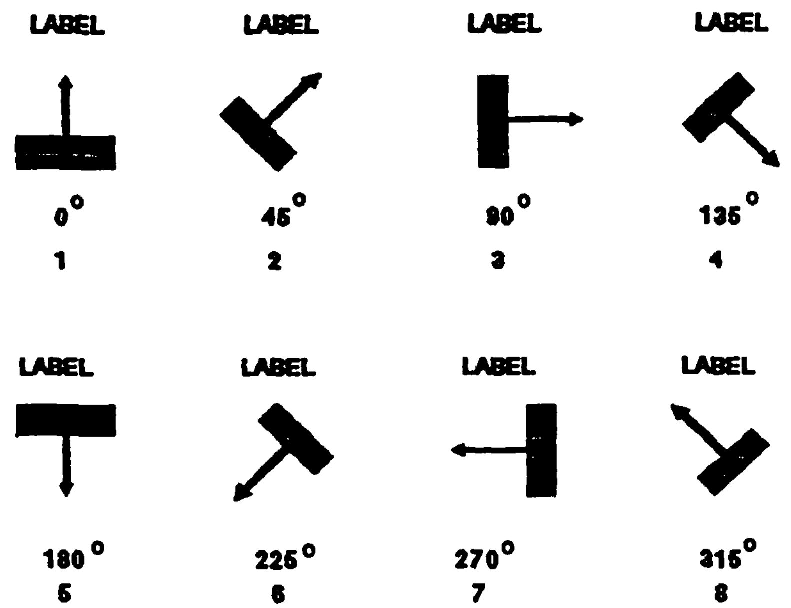
PROGRAM LABELING CONVENTION



The positions of the labels on the faces of the cube are used in defining the orientations of the ports. The positioning of the labels as given above must be followed for correct correspondence with the computer program.

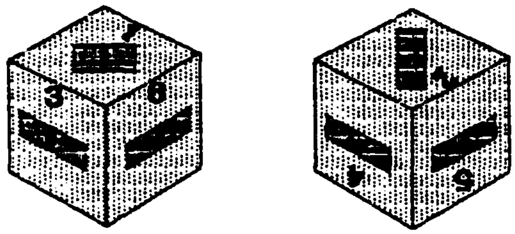
FACE ORIENTATION

To designate the orientation of the waveguide input on each face, an arrow perpendicular to the long axis of the waveguide is used. The zero-degree position corresponds to an orientation direction pointing toward the label. The orientation angle is measured clockwise positive. The eight possible waveguide orientations are shown below. (An integer is used to designate desired waveguide orientation at each face for input to the computer program.)



SAMPLE TEST CASES

Sample test cases are provided for Eisenhart's turnstile, Purcell's junction, and Ginzton's junction (Figures A-1 through A-3, respectively). These figures define each configuration and their computed symmetry constraint matrix. Sample runstreams for each configuration follow each figure.



Face #	Angle
1	135
2	315
3	0
4	0
5	0
6	0

Symmetry Constraint Matrix

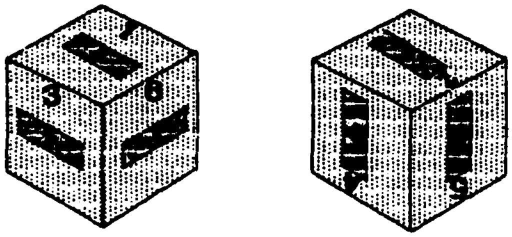
$$\begin{bmatrix} 11 & 0 & 13 & -13 & -13 & 13 \\ 0 & 11 & 13 & 13 & -13 & -13 \\ 31 & 31 & 33 & 43 & 53 & 43 \\ -31 & 31 & 43 & 33 & 43 & 53 \\ -31 & -31 & 53 & 43 & 33 & 43 \\ 31 & -31 & 43 & 53 & 43 & 33 \end{bmatrix}$$

FIGURE A-1. Eisenhart's Turnstile.

EISENHART TURNSTILE RUNSTREAM

When entering the orientations of the faces, use the following:
1 = 0
2 = 45
3 = 90
4 = 135
5 = 180
6 = 225
7 = 270
8 = 315
Enter the orientation of face 1
4
Enter the orientation of face 2
8
Enter the orientation of face 3
1
Enter the orientation of face 4
1
Enter the orientation of face 5

```
1
Enter the orientation of face 6
1
Face 1 is at 135 degrees.  Is that correct?  y
Face 2 is at 315 degrees.  Is that correct?  y
Face 3 is at 0 degrees.  Is that correct?  y
Face 4 is at 0 degrees.  Is that correct?  y
Face 5 is at 0 degrees. Is that correct?  y
Face 6 is at 0 degrees.  Is that correct?  y
rotate_about_x
Matrix s2 = RSR-1
Matrix s3
rotate_about_y
Matrix s2 = RSR-1
Matrix s3
rotate_about_z
Matrix s2 = RSR-1
Matrix s3
mirror_about_x_eq_y_plane
Matrix s2 = RSR-1
Matrix s3
mirror_about_x_eq_minus_y_plane
Matrix s2 = RSR-1
Matrix s3

(  11    0   13  -13  -13   13 )
(                              )
(   0   11   13   13  -13  -13 )
(                              )
(  31   31   33   43   53   43 )
(                              )
( -31   31   43   33   43   53 )
(                              )
( -31  -31   53   43   33   43 )
(                              )
(  31  -31   43   53   43   33 )
(                              )
```



Face #	Angle
1	180
2	270
3	0
4	270
5	90
6	180

Symmetry Constraint Matrix

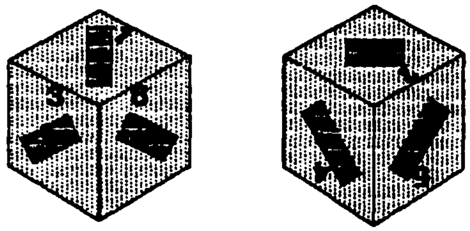
$$\begin{bmatrix} 11 & 21 & 13 & 14 & 15 & 16 \\ 21 & 11 & -13 & 14 & 15 & -16 \\ 31 & -31 & 33 & 0 & 0 & 36 \\ 41 & 41 & 0 & 44 & 45 & 0 \\ 51 & 51 & 0 & 54 & 55 & 0 \\ 61 & -61 & 63 & 0 & 0 & 66 \end{bmatrix}$$

FIGURE A-2. Purcell's Junction.

RUNSTREAM FOR PURCELL'S JUNCTION

When entering the orientations of the faces, use the following:
1 = 0
2 = 45
3 = 90
4 = 135
5 = 180
6 = 225
7 = 270
8 = 315
Enter the orientation of face 1
5
Enter the orientation of face 2
7
Enter the orientation of face 3
1
Enter the orientation of face 4
7
Enter the orientation of face 5

```
3
Enter the orientation of face 6
5
Face 1 is at 180 degrees.  Is that correct?  y
Face 2 is at 270 degrees.  Is that correct?  y
Face 3 is at 0 degrees.  Is that correct?  y
Face 4 is at 270 degrees.  Is that correct?  y
Face 5 is at 90 degrees.  Is that correct?  y
Face 6 is at 180 degrees.  Is that correct?  y
mirror_about_x_y_plane
Matrix s2 = RSR-1
Matrix s3

( 11   21   13   14   15   16 )
(                             )
( 21   11  -13   14   15  -16 )
(                             )
( 31  -31   33    0    0   36 )
(                             )
( 41   41    0   44   45    0 )
(                             )
( 51   51    0   54   55    0 )
(                             )
( 61  -61   63    0    0   66 )
(                             )
```



Face #	Angle
1	45
2	45
3	135
4	225
5	135
6	225

Symmetry Constraint Matrix

$$\begin{bmatrix} 11 & 0 & 31 & 31 & -31 & -31 \\ 0 & 11 & -31 & 31 & 31 & -31 \\ 31 & -31 & 11 & 31 & 0 & 31 \\ 31 & 31 & 31 & 11 & 31 & 0 \\ -31 & 31 & 0 & 31 & 11 & 31 \\ -31 & -31 & 31 & 0 & 31 & 11 \end{bmatrix}$$

FIGURE A-3. Ginzton Junction.

RUNSTREAM FOR GINZTON JUNCTION

When entering the orientations of the faces, use the following:
1 = 0
2 = 45
3 = 90
4 = 135
5 = 180
6 = 225
7 = 270
8 = 315
Enter the orientation of face 1
2
Enter the orientation of face 2
2
Enter the orientation of face 3
4
Enter the orientation of face 4
6
Enter the orientation of face 5

```
4
Enter the orientation of face 6
6
Face 1 is at 45 degrees.  Is that correct?  y
Face 2 is at 45 degrees.  Is that correct?  y
Face 3 is at 135 degrees.  Is that correct?  y
Face 4 is at 225 degrees.  Is that correct?  y
Face 5 is at 135 degrees.  Is that correct?  y
Face 6 is at 225 degrees.  Is that correct?  y
rotate_about_x
Matrix s2 = RSR-1
Matrix s3
rotate_about_y
Matrix s2 = RSR-1
Matrix s3
rotate_about_z
Matrix s2 = RSR-1
Matrix s3
rotate_120-about-neg-y-eq-x-eq-z
Matrix s2 = RSR-1
Matrix s3
rotate_240_about_neg_y_eq_x_eq_z
Matrix s2 = RSR-1
Matrix s3
rotate_120_about_y_eq_x_eq_z
Matrix s2 = RSR-1
Matrix s3
rotate_240_about_y_eq_x_eq_z
Matrix s2 = RSR-1
Matrix s3
rotate_120_about_y_eq_x_eq_neg_z
Matrix s2 = RSR-1
Matrix s3
rotate_240_about_y_eq_x_eq_neg_z
Matrix s2 = RSR-1
Matrix s3
rotate_120_about_y_eq_neg_x_eq_z
Matrix s2 = RSR-1
Matrix s3
rotate_240_about_y_eq_neg_x_eq_z
Matrix s2 = RSR-1
Matrix s3
mirror_about_x_eq_y_plane
Matrix s2 = RSR-1
Matrix s3
mirror_about_x_eq_minus_y_plane
Matrix s2 = RSR-1
Matrix s3
mirror_about_x_eq_minus_z_plane
Matrix s2 = RSR-1
```

```
Matrix s3
mirror_about_x_eq_z_plane
Matrix s2 = RSR-1
Matrix s3
mirror_about_z_eq_y_plane
Matrix s2 = RSR-1
Matrix s3
mirror_about_z_eq_neg_y_plane
Matrix s2 = RSR-1
Matrix s3

(  11    0   31   31  -31  -31 )
(                              )
(   0   11  -31   31   31  -31 )
(                              )
(  31  -31   11   31    0   31 )
(                              )
(  31   31   31   11   31    0 )
(                              )
( -31   31    0   31   11   31 )
(                              )
( -31  -31   31    0   31   11 )
(                              )
```

PROGRAM SYMMETRY(INPUT,OUTPUT)

(* This is the first step in the program for finding the S-matrix associated with different configurations of the BOWLING CUBE. The subroutine allows the user to input the configuration he wants studied. After the user has entered the orientations, the procedure converts the units given by the user into the symbology used by the program as a whole.

After completing this task, SYMMETRY passes control to the procedure MAINTEST. MAINTEST finds the symmetry operators associated with the configuration being studied. *)

```
TYPE SMATRIX = ARRAY[1..6] OF INTEGER;
TYPE SIX_BY_SIX = ARRAY[1..6,1..6] OF INTEGER;

VAR     pointer: ARRAY[1..6,'x'..'z'] OF INTEGER;
(* Vector representation of the orientation of the 6 faces*)

VAR before:  ARRAY [1..6, 'x'..'z'] OF INTEGER;
VAR after:   ARRAY [1..6, 'x'..'z'] OF INTEGER;
VAR move:  ARRAY [1..6] OF INTEGER;
VAR s1,s2,s3: SIX_BY_SIX;
VAR result:  BOOLEAN;
VAR sign:    ARRAY[1..6] OF INTEGER;
```

```
TYPE
      angles=1..8;
      faces=1..6;

VAR
      ans: CHAR;    (* User reply to yes and no questions*)
      i: INTEGER;    (* Counting integer*)
      ii: faces;     (* Counter for a CASE statement*)
      orientation: ARRAY[1..6] OF angles;  (* Orientation of the 6 faces*)

{$P128}   (*Lists program input and output on printer*)
{$I  a:compare.pas}
{$I  a:combiner.pas}
{$I  a:operators.pas}
{$I  a:printout.pas}
{$I  a:maintest.pas}


BEGIN

(* Ask the user for the orientations of the cube faces *)

      WRITELN(' When entering the orientations of the faces use the
following');
            WRITELN(' 1 = 0');
            WRITELN(' 2 = 45');
            WRITELN(' 3 = 90');
            WRITELN(' 4 = 135');
            WRITELN(' 5 = 180');
            WRITELN(' 6 = 225');
            WRITELN(' 7 = 270');
            WRITELN(' 8 = 315');

      FOR i:= 1 TO 6 DO
            BEGIN
                  WRITELN(' Enter the orientation of face ',i:1);
                  READLN(orientation[i])
            END;

(* Allow the user to correct mistakes in entering face orientations *)

      FOR i:= 1 TO 6 DO
            BEGIN
                  WRITE ('Face ',i:1,' is at');
                  CASE orientation[i] OF
                        1:  WRITE(' 0');
                        2:  WRITE(' 45');
                        3:  WRITE(' 90');
                        4:  WRITE(' 135');
                        5:  WRITE(' 180');
```

```
                    6:  WRITE(' 225');
                    7:  WRITE(' 270');
                    8:  WRITE(' 315');
        END;
        WRITE (' degrees.  Is that correct?  ');
        READLN (ans);
        IF ((ans='N')OR(ans='n')) THEN
            BEGIN
                WRITE(' Enter the correct orientation ');
                READLN (orientation[i])
            END
        END;

(* Translate the user input to coordinates on the cube*)

    FOR i:= 1 to 6 DO
        BEGIN
            ii:=i;
            CASE ii OF
                1: BEGIN
                    pointer[1,'z']:= 1;
                    CASE orientation[i] OF
                        1: BEGIN
                            pointer[1,'x']:= -1;
                            pointer[1,'y']:= 0;
                          END;
                        2: BEGIN
                            pointer[1,'x']:= -1;
                            pointer[1,'y']:= 1;
                          END;
                        3: BEGIN
                            pointer[1,'x']:= 0;
                            pointer[1,'y']:= 1;
                          END;
                        4: BEGIN
                            pointer[1,'x']:= 1;
                            pointer[1,'y']:= 1;
                          END;
                        5: BEGIN
                            pointer[1,'x']:= 1;
                            pointer[1,'y']:= 0;
                          END;
                        6: BEGIN
                            pointer[1,'x']:= 1;
                            pointer[1,'y']:= -1;
                          END;
                        7: BEGIN
                            pointer[1,'x']:= 0;
                            pointer[1,'y']:= -1;
                          END;
```

```
                    8: BEGIN
                          pointer[1,'x']:=  -1;
                          pointer[1,'y']:=  -1;
                       END;
END;
      END;
2: BEGIN
      pointer[2,'z']:=  -1;
      CASE orientation[i] OF
            1: BEGIN
                  pointer[2,'x']:=  -1;
                  pointer[2,'y']:=  0;
END;
            2: BEGIN
                  pointer[2,'x']:=  -1;
                  pointer[2,'y']:=  -1;
END;
            3: BEGIN
                  pointer[2,'x']:=  0;
                  pointer[2,'y']:=  -1;
END;
            4: BEGIN
                  pointer[2,'x']:=  1;
                  pointer[2,'y']:=  -1;
END;
            5: BEGIN
                  pointer[2,'x']:=  1;
                  pointer[2,'y']:=  0;
END;
            6: BEGIN
                  pointer[2,'x']:=  1;
                  pointer[2,'y']:=  1;
END;
            7: BEGIN
                  pointer[2,'x']:=  1;
                  pointer[2,'y']:=  0;
END;
            8: BEGIN
                  pointer[2,'x']:=  -1;
                  pointer[2,'y']:=  1;
               END;
END;
      END;
3: BEGIN
      pointer[3,'x']:=  1;
      CASE orientation[i] OF
            1: BEGIN
                  pointer[3,'z']:=  1;
                  pointer[3,'y']:=  0;
               END;
```

```
                    2: BEGIN
                            pointer[3,'z']:=  1;
                            pointer[3,'y']:=  1;
                       END;
                    3: BEGIN
                            pointer[3,'z']:=  0;
                            pointer[3,'y']:=  1;
                       END;
                    4: BEGIN
                            pointer[3,'z']:=  -1;
                            pointer[3,'y']:=  1;
                       END;
                    5: BEGIN
                            pointer[3,'z']:=  -1;
                            pointer[3,'y']:=  0;
                       END;
                    6: BEGIN
                            pointer[3,'z']:=  -1;
                            pointer[3,'y']:=  -1;
                       END;
                    7: BEGIN
                            pointer[3,'z']:=  0;
                            pointer[3,'y']:=  -1;
                       END;
                    8: BEGIN
                            pointer[3,'z']:=  1;
                            pointer[3,'y']:=  -1;
                       END;
END;
        END;
4: BEGIN
        pointer[4,'y']:=  -1;
        CASE orientation[i] OF
                    1: BEGIN
                            pointer[4,'x']:=  0;
                            pointer[4,'z']:=  1;
                       END;
                    2: BEGIN
                            pointer[4,'x']:=  1;
                            pointer[4,'z']:=  1;
                       END;
                    3: BEGIN
                            pointer[4,'x']:=  1;
                            pointer[4,'z']:=  0;
                       END;
                    4: BEGIN
                            pointer[4,'x']:=  1;
                            pointer[4,'z']:=  -1;
                       END;
                    5: BEGIN
```

```
                        pointer[4,'x']:=  0;
                        pointer[4,'z']:=  -1;
                   END;
               6: BEGIN
                        pointer[4,'x']:=  -1;
                        pointer[4,'z']:=  -1;
                   END;
               7: BEGIN
                        pointer[4,'x']:=  -1;
                        pointer[4,'z']:=  0;
                   END;
               8: BEGIN
                        pointer[4,'x']:=  -1;
                        pointer[4,'z']:=  1;
                    END;
END;
        END;
5: BEGIN
        pointer[5,'x']:=  -1;
        CASE orientation[i] OF
               1: BEGIN
                        pointer[5,'z']:=  1;
                        pointer[5,'y']:=  0;
                   END;
               2: BEGIN
                        pointer[5,'z']:=  1;
                        pointer[5,'y']:=  -1;
                   END;
               3: BEGIN
                        pointer[5,'z']:=  0;
                        pointer[5,'y']:=  -1;
                   END;
               4: BEGIN
                        pointer[5,'z']:=  -1;
                        pointer[5,'y']:=  -1;
                   END;
               5: BEGIN
                        pointer[5,'z']:=  -1;
                        pointer[5,'y']:=  0;
                   END;
               6: BEGIN
                        pointer[5,'z']:=  -1;
                        pointer[5,'y']:=  1;
                   END;
               7: BEGIN
                        pointer[5,'z']:=  0;
                        pointer[5,'y']:=  1;
                   END;
               8: BEGIN
                        pointer[5,'z']:=  1;
```

```
                                        pointer[5,'y']:=  1;
                                    END;
                END;
                        END;
                6: BEGIN
                        pointer[6,'y']:=  1;
                        CASE orientation[i] OF
                                1: BEGIN
                                        pointer[6,'x']:=  0;
                                        pointer[6,'z']:=  1;
                                    END;
                                2: BEGIN
                                        pointer[6,'x']:=  -1;
                                        pointer[6,'z']:=  1;
                                    END;
                                3: BEGIN
                                        pointer[6,'x']:=  -1;
                                        pointer[6,'z']:=  0;
                                    END;
                                4: BEGIN
                                        pointer[6,'x']:=  -1;
                                        pointer[6,'z']:=  -1;
                                    END;
                                5: BEGIN
                                        pointer[6,'x']:=  0;
                                        pointer[6,'z']:=  -1;
                                    END;
                                6: BEGIN
                                        pointer[6,'x']:=  1;
                                        pointer[6,'z']:=  -1;
                                    END;
                                7: BEGIN
                                        pointer[6,'x']:=  1;
                                        pointer[6,'z']:=  0;
                                    END;
                                8: BEGIN
                                        pointer[6,'x']:=  1;
                                        pointer[6,'z']:=  1;
                                    END;
                END;
                        END;
            END;
        END;
    maintest;
    END.

(* This program finds the symmetries of a given configuration *)

(*
```

```
****************************************************************

****************************************************************
************ PROCEDURE TO FIND THE INVERSE OF A MATRIX
*************

****************************************************************

****************************************************************
*)


PROCEDURE INVERTER( matrix: SMATRIX;

      VAR matrix_inverse: SMATRIX );
(* This procedure produces the inverse to a matrix.  Both the matrix and
      the inverse are given in row representation. *)

VAR
      column_representation: ARRAY [1..6] OF INTEGER;
      i: INTEGER;

BEGIN
(* Find the column representation inverse corresponding to the row
      representation 'matrix'   *)
FOR i:= 1 TO 6 DO
      column_representation[i]:=  matrix[i];

(* convert the column representation to row representation *)
FOR i:= 1 TO 6 DO
      matrix_inverse[  abs(column_representation[i])]:=
      i*  column_representation[i]  div  abs(column_representation[i]);
END;

(*
****************************************************************
****************************************************************
****************************  FIND  INV(M)*S*M  ****************
****************************************************************
****************************************************************
*)


PROCEDURE sgen (matrix,matrix_inverse: SMATRIX; VAR return: SIX_BY_SIX);


VAR
      column: SMATRIX;
      i: INTEGER;  (* counting integer *)
      j: INTEGER;  (* counting integer *)
```

```
      temp: SIX_BY_SIX;

BEGIN

(* convert 'matrix' to column representation *)
      FOR i:= 1 TO 6 DO
      column[abs(matrix[i])]:= i* (matrix[i] div abs(matrix[i]));


(* Find the return matrix = matrix_inverse * temp * matrix *)
      FOR i:= 1 TO 6 DO
      FOR j:= 1 TO 6 DO
      return[i,j]:= (abs(matrix_inverse[i])*10+abs( column[j]))
            * (abs( matrix_inverse[i]) div matrix_inverse[i])
            * (abs( column[j]) div column[j]);
END;


(*
*****************************************************************
***************** MAIN PROCEDURE FOR FINDING S-MATRIX
***********
*****************************************************************
*)


PROCEDURE maintest;

VAR i: INTEGER; (* counting integer for a FOR loop *)
VAR j: INTEGER; (* counting integer for a FOR loop *)
VAR operator, inverse: SMATRIX;

BEGIN
(* Initialize the matrix 'before'. It is equal to the orientation matrix
      (pointer) furnished by the subroutine SYMTEST *)
      FOR i:= 1 TO 6 DO
      BEGIN
            before[i,'x']:= pointer[i,'x'];
            before[i,'y']:= pointer[i,'y'];
            before[i,'z']:= pointer[i,'z'];
      END;

(* Initialize the S-MATRIX *)
      FOR i:= 1 TO 6 DO
      FOR j:= 1 TO 6 DO
            s1[j,i]:= j*10 + i;

      rotate_about_x ;
      (* Rotates the cube 180 degrees about the x axis*)
```

```
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'rotate_about_x');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
FOR i:= 1 TO 6 DO
FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

rotate_about_y ;
        (* Rotates the cube 180 degrees about the y axis*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'rotate_about_y');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

rotate_about_z;
(* Rotates the cube 180 degrees about the z axis*)
compare ;
IF (result)
THEN
BEGIN
```

```
        FOR i:= 1 TO 6 DO
        operator[i]:=  move[i]*sign[i];
        WRITELN({LST,}'rotate_about_z');
        operator_out(operator);
        inverter  (operator,  inverse);
        sgen(operator,  inverse,  s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix  s3');
        S_Matrix_out(s3);
END;

rotate_about_x_eq_neg_z  ;
(* Rotates the cube 180 degrees about the x=-z axis*)
compare  ;
IF  (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:=  move[i]*sign[i];
        WRITELN({LST,}'rotate_about_x_eq_neg_z');
        operator_out(operator);
        inverter  (operator,  inverse);
        sgen(operator,  inverse,  s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix  s3');
        S_Matrix_out(s3);
END;

rotate_about_y_eq_neg_z  ;
(* Rotates the cube 180 degrees about the y=-z axis*)
compare  ;
IF  (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:=  move[i]*sign[i];
        WRITELN({LST,}'rotate_about_y_eq_neg_z');
        operator_out(operator);
```

```
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

rotate_about_x_eq_z ;
(* Rotates the cube 180 degrees about the x=z axis*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'rotate_about_x_eq_z');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

rotate_about_y_eq_z ;
(* Rotates the cube 180 degrees about the y=z axis*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'rotate_about_y_eq_z');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
```

```
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

rotate_about_y_eq_neg_x ;
(* Rotates the cube 180 degrees about the y=-x axis*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'rotate_about_y_eq_neg_x');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

rotate_about_y_eq_x ;
(* Rotates the cube 180 degrees about the y=x axis*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'rotate_about_y_eq_x');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
```

```
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix  s3');
        S_Matrix_out(s3);
END;

rotat_120_about_neg_y_eq_x_eq_z  ;
(* Rotates the cube 120 degrees about the -y=x=z axis*)
compare  ;
F  (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:=  move[i]*sign[i];
        WRITELN({LST,}'rotate_120_about_neg_y_eq_x_eq_z');
        operator_out(operator);
        inverter  (operator,  inverse);
        sgen(operator,  inverse,  s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix  s3');
        S_Matrix_out(s3);
END;


rotat_240_about_neg_y_eq_x_eq_z  ;
(* Rotates the cube 240 degrees about the -y=x=z axis*)
compare  ;
IF  (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:=  move[i]*sign[i];
        WRITELN({LST,}'rotate_240_about_neg_y_eq_x_eq_z');
        operator_out(operator);
        inverter  (operator,  inverse);
        sgen(operator,  inverse,  s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
         FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix  s3');
        S_Matrix_out(s3);
```

```
END;


rotate_120_about_y_eq_x_eq_z  ;
(* Rotates the cube 120 degrees about the y=x=z axis*)
compare ;
IF (result)
THEN
BEGIN
      FOR i:= 1 TO 6 DO
      operator[i]:=  move[i]*sign[i];
      WRITELN({LST,}'rotate_120_about_y_eq_x_eq_z');
      operator_out(operator);
      inverter (operator, inverse);
      sgen(operator, inverse, s2);
      WRITELN({LST,}'Matrix s2 = RSR-1');
      S_Matrix_out(s2);
      combiner;
      (* Save the result in s1 *)
      FOR i:= 1 TO 6 DO
      FOR j:= 1 TO 6 DO
      s1[i,j]:= s3[i,j];
      WRITELN({LST,}'Matrix s3');
      S_Matrix_out(s3);
END;

rotate_240_about_y_eq_x_eq_z  ;
(* Rotates the cube 240 degrees about the y=x=z axis*)
compare ;
IF (result)
THEN
BEGIN
      FOR i:= 1 TO 6 DO
      operator[i]:=  move[i]*sign[i];
      WRITELN({LST,}'rotate_240_about_y_eq_x_eq_z');
      operator_out(operator);
      inverter (operator, inverse);
      sgen(operator, inverse, s2);
      WRITELN({LST,}'Matrix s2 = RSR-1');
      S_Matrix_out(s2);
      combiner;
      (* Save the result in s1 *)
      FOR i:= 1 TO 6 DO
      FOR j:= 1 TO 6 DO
      s1[i,j]:= s3[i,j];
      WRITELN({LST,}'Matrix s3');
      S_Matrix_out(s3);
END;

rotat_120_about_y_eq_x_eq_neg_z  ;
```

```
(* Rotates the cube 120 degrees about the y=x=-z axis*)
compare ;
IF (result)
THEN
BEGIN
      FOR i:= 1 TO 6 DO
      operator[i]:= move[i]*sign[i];
      WRITELN({LST,}'rotate_120_about_y_eq_x_eq_neg_z');
      operator_out(operator);
      inverter (operator, inverse);
      sgen(operator, inverse, s2);
      WRITELN({LST,}'Matrix s2 = RSR-1');
      S_Matrix_out(s2);
      combiner;
      (* Save the result in s1 *)
      FOR i:= 1 TO 6 DO
      FOR j:= 1 TO 6 DO
      s1[i,j]:= s3[i,j];
      WRITELN({LST,}'Matrix s3');
      S_Matrix_out(s3);
END;

rotat_240_about_y_eq_x_eq_neg_z ;
(* Rotates the cube 240 degrees about the y=x=-z axis*)
compare ;
IF (result)
THEN
BEGIN
      FOR i:= 1 TO 6 DO
      operator[i]:= move[i]*sign[i];
      WRITELN({LST,}'rotate_240_about_y_eq_x_eq_neg_z');
      operator_out(operator);
      inverter (operator, inverse);
      sgen(operator, inverse, s2);
      WRITELN({LST,}'Matrix s2 = RSR-1');
      S_Matrix_out(s2);
      combiner;
      (* Save the result in s1 *)
      FOR i:= 1 TO 6 DO
      FOR j:= 1 TO 6 DO
      s1[i,j]:= s3[i,j];
      WRITELN({LST,}'Matrix s3');
      S_Matrix_out(s3);
END;

rotat_120_about_y_eq_neg_x_eq_z ;
(* Rotates the cube 120 degrees about the y=-x=z axis*)
compare ;
IF (result)
```

```
THEN
BEGIN
      FOR i:= 1 TO 6 DO
      operator[i]:= move[i]*sign[i];
      WRITELN({LST,}'rotate_120_about_y_eq_neg_x_eq_z');
      operator_out(operator);
      inverter (operator, inverse);
      sgen(operator, inverse, s2);
      WRITELN({LST,}'Matrix s2 = RSR-1');
      S_Matrix_out(s2);
      combiner;
      (* Save the result in s1 *)
      FOR i:= 1 TO 6 DO
      FOR j:= 1 TO 6 DO
      s1[i,j]:= s3[i,j];
      WRITELN({LST,}'Matrix s3');
      S_Matrix_out(s3);
END;

rotat_240_about_y_eq_neg_x_eq_z ;
(* Rotates the cube 240 degrees about the y=-x=z axis*)
compare ;
IF (result)
THEN
BEGIN
      FOR i:= 1 TO 6 DO
      operator[i]:= move[i]*sign[i];
      WRITELN({LST,}'rotate_240_about_y_eq_neg_x_eq_z');
      operator_out(operator);
      inverter (operator, inverse);
      sgen(operator, inverse, s2);
      WRITELN({LST,}'Matrix s2 = RSR-1');
      S_Matrix_out(s2);
      combiner;
      (* Save the result in s1 *)
      FOR i:= 1 TO 6 DO
      FOR j:= 1 TO 6 DO
      s1[i,j]:= s3[i,j];
      WRITELN({LST,}'Matrix s3');
      S_Matrix_out(s3);
END;

mirror_about_x_y_plane ;
(* Performs the mirror symmetry about the x-y plane. i.e. replaces
z with -z*)
compare ;
IF (result)
THEN
BEGIN
      FOR i:= 1 TO 6 DO
```

```
        operator[i]:=  move[i]*sign[i];
        WRITELN({LST,}'mirror_about_x_y_plane');
        operator_out(operator);
        inverter  (operator,  inverse);
        sgen(operator,  inverse,  s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix  s3');
        S_Matrix_out(s3);
END;

mirror_about_x_z_plane   ;
(* Performs the mirror symmetry about the x-z plane.  i.e. replaces
        y with -y*)
compare  ;
IF  (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:=  move[i]*sign[i];
        WRITELN({LST,}'mirror_about_x_z_plane');
        operator_out(operator);
        inverter  (operator,  inverse);
        sgen(operator,  inverse,  s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix  s3');
        S_Matrix_out(s3);
END;

mirror_about_y_z_plane   ;
(* Performs the mirror symmetry about the y-z plane.  i.e. replaces
        x with -x*)
compare  ;
IF  (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:=  move[i]*sign[i];
        WRITELN({LST,}'mirror_about_y_z_plane');
```

```
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

mirror_about_x_eq_y_plane ;
(* Performs the mirror symmetry about the x=y plane.
        z is a free variable*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'mirror_about_x_eq_y_plane');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

mirror_about_x_eq_minus_y_plane ;
(* Performs the mirror symmetry about the x=-y plane.
        z is a free variable*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'mirror_about_x_eq_minus_y_plane');
        operator_out(operator);
        inverter (operator, inverse);
```

```
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

mirror_about_x_eq_minus_z_plane ;
(* Performs the mirror symmetry about the x=-z plane.
        y is a free variable*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'mirror_about_x_eq_minus_z_plane');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

mirror_about_x_eq_z_plane ;
(* Performs the mirror symmetry about the x=z plane.
        y is a free variable*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'mirror_about_x_eq_z_plane');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
```

```
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

mirror_about_z_eq_y_plane ;
(* Performs the mirror symmetry about the z=y plane.
        x is a free variable*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'mirror_about_z_eq_y_plane');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
        (* Save the result in s1 *)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s1[i,j]:= s3[i,j];
        WRITELN({LST,}'Matrix s3');
        S_Matrix_out(s3);
END;

mirror_about_z_eq_neg_y_plane ;
(* Performs the mirror symmetry about the z=-y plane.
        x is a free variable*)
compare ;
IF (result)
THEN
BEGIN
        FOR i:= 1 TO 6 DO
        operator[i]:= move[i]*sign[i];
        WRITELN({LST,}'mirror_about_z_eq_neg_y_plane');
        operator_out(operator);
        inverter (operator, inverse);
        sgen(operator, inverse, s2);
        WRITELN({LST,}'Matrix s2 = RSR-1');
        S_Matrix_out(s2);
        combiner;
```

```
                (* Save the result in s1 *)
                FOR i:= 1 TO 6 DO
                FOR j:= 1 TO 6 DO
                s1[i,j]:= s3[i,j];
                WRITELN({LST,}'Matrix s3');
                S_Matrix_out(s3);
        END;
        FOR i:= 1 TO 6 DO
        BEGIN
                WRITE(' (');
                FOR j:= 1 TO 6 DO
                WRITE (s1[i,j]:3,' ');
                WRITELN (' )');
                WRITELN (' (                    )');
        END;

END;

PROCEDURE ROTATE_ABOUT_Y_EQ_Z ;

(* This procedure effects a rotation of 180 degrees about the
        axis y=z and returns the new array in "after". *)


VAR i: INTEGER;

        BEGIN
                move[1]:=6;
                move[2]:=4;
                move[3]:=5;
                move[4]:=2;
                move[5]:=3;
                move[6]:=1;
                FOR i:= 1 TO 6 DO
        BEGIN
                after[i,'x']:=  -before[move[i],'x'];
                after[i,'y']:=  before[move[i],'z'];
                after[i,'z']:=  before[move[i],'y']
        END
END;


PROCEDURE MIRROR_ABOUT_0 ;

(* This procedure effects a mirror about the origin and
        returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)
```

```
        BEGIN
                move[1]:=2;
                move[2]:=1;
                move[3]:=5;
                move[4]:=6;
                move[5]:=3;
                move[6]:=4;
                FOR i:= 1 TO 6 DO
        BEGIN
                after[i,'x']:=  -before[move[i],'x'];
                after[i,'y']:=  -before[move[i],'y'];
                after[i,'z']:=  -before[move[i],'z']
        END
END;

PROCEDURE MIRROR_ABOUT_X_EQ_Y_PLANE ;

(* This procedure effects a mirror about the plane x=y (z is a free variable)
        and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

        BEGIN
                move[1]:=1;
                move[2]:=2;
                move[3]:=6;
                move[4]:=5;
                move[5]:=4;
                move[6]:=3;
                FOR i:= 1 TO 6 DO
        BEGIN
                after[i,'x']:=  before[move[i],'y'];
                after[i,'y']:=  before[move[i],'x'];
                after[i,'z']:=  before[move[i],'z']
        END
END;

PROCEDURE MIRROR_ABOUT_X_EQ_Z_PLANE ;

(* This procedure effects a mirror about the plane x=z
        and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

        BEGIN
                move[1]:=5;
```

```
        move[2]:=3;
        move[3]:=2;
        move[4]:=4;
        move[5]:=1;
        move[6]:=6;
        FOR i:= 1 TO 6 DO
    BEGIN
        after[i,'x']:=  -before[move[i],'z'];
        after[i,'y']:=   before[move[i],'y'];
        after[i,'z']:=  -before[move[i],'x']
    END
END;

PROCEDURE MIRROR_ABOUT_Z_EQ_Y_PLANE ;

(* This procedure effects a mirror about the plane z=y (x is free)
    and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

    BEGIN
        move[1]:=4;
        move[2]:=6;
        move[3]:=3;
        move[4]:=1;
        move[5]:=5;
        move[6]:=2;
        FOR i:= 1 TO 6 DO
    BEGIN
        after[i,'x']:=   before[move[i],'x'];
        after[i,'y']:=  -before[move[i],'z'];
        after[i,'z']:=  -before[move[i],'y']
    END
END;

PROCEDURE MIRROR_ABOUT_X_Y_PLANE ;

(* This procedure effects a mirror about the x-y plane (z=0)
    and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

    BEGIN
        move[1]:=2;
        move[2]:=1;
        move[3]:=3;
```

```
            move[4]:=4;
            move[5]:=5;
            move[6]:=6;
            FOR i:= 1 TO 6 DO
        BEGIN
            after[i,'x']:=before[move[i],'x'];
            after[i,'y']:=before[move[i],'y'];
            after[i,'z']:=-before[move[i],'z']
        END
END;

PROCEDURE MIRROR_ABOUT_X_Z_PLANE ;

(* This procedure effects a mirror about the x-z plane (y=0)
        and returns the new array in "after". *)

VAR i: INTEGER;   (*counting integer*)

        BEGIN
            move[1]:=1;
            move[2]:=2;
            move[3]:=3;
            move[4]:=6;
            move[5]:=5;
            move[6]:=4;
            FOR i:= 1 TO 6 DO
        BEGIN
            after[i,'x']:=before[move[i],'x'];
            after[i,'y']:=-before[move[i],'y'];
            after[i,'z']:=before[move[i],'z']
        END
END;

PROCEDURE MIRROR_ABOUT_Y_Z_PLANE ;

(* This procedure effects a mirror about the y-z plane (x=0)
        and returns the new array in "after". *)

VAR i: INTEGER;   (*counting integer*)

        BEGIN
            move[1]:=1;
            move[2]:=2;
            move[3]:=5;
            move[4]:=4;
            move[5]:=3;
```

```
            move[6]:=6;
            FOR i:= 1 TO 6 DO
        BEGIN
            after[i,'x']:=-before[move[i],'x'];
            after[i,'y']:=before[move[i],'y'];
            after[i,'z']:=before[move[i],'z']
        END
END;

PROCEDURE MIRROR_ABOUT_X_EQ_MINUS_Y_PLANE ;

(* This procedure effects a mirror about the plane x=-y (z is a free variable)
        and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

        BEGIN
            move[1]:=1;
            move[2]:=2;
            move[3]:=4;
            move[4]:=3;
            move[5]:=6;
            move[6]:=5;
            FOR i:= 1 TO 6 DO
        BEGIN
            after[i,'x']:=  -before[move[i],'y'];
            after[i,'y']:=  -before[move[i],'x'];
            after[i,'z']:=  before[move[i],'z']
        END
END;

PROCEDURE MIRROR_ABOUT_X_EQ_MINUS_Z_PLANE ;

(* This procedure effects a mirror about the plane x=-z
        and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

        BEGIN
            move[1]:=3;
            move[2]:=5;
            move[3]:=1;
            move[4]:=4;
            move[5]:=2;
            move[6]:=6;
            FOR i:= 1 TO 6 DO
```

```
        BEGIN
                after[i,'x']:=  before[move[i],'z'];
                after[i,'y']:=  before[move[i],'y'];
                after[i,'z']:=  before[move[i],'x']
        END
END;


PROCEDURE MIRROR_ABOUT_Z_EQ_NEG_Y_PLANE ;

(* This procedure effects a mirror about the plane z=-y   (x free)
        and returns the new array in "after". *)


VAR i: INTEGER;  (*counting integer*)

        BEGIN
                move[1]:=6;
                move[2]:=4;
                move[3]:=3;
                move[4]:=2;
                move[5]:=5;
                move[6]:=1;
                FOR i:= 1 TO 6 DO
        BEGIN
                after[i,'x']:=  before[move[i],'x'];
                after[i,'y']:=  before[move[i],'z'];
                after[i,'z']:=  before[move[i],'y']
        END
END;


PROCEDURE ROTAT_120_ABOUT_Y_EQ_NEG_X_EQ_Z ;

(* This procedure effects a rotation of 120 degrees about the axis
        x=-y=-z and returns the new array in "after". *)


VAR i: INTEGER;  (*counting integer*)

        BEGIN
                move[1]:=5;
                move[2]:=3;
                move[3]:=4;
                move[4]:=2;
                move[5]:=6;
                move[6]:=1;
                FOR i:= 1 TO 6 DO
        BEGIN
                after[i,'x']:=-before[move[i],'y'];
```

```
            after[i,'y']:=before[move[i],'z'];
            after[i,'z']:=-before[move[i],'x']
        END
END;

PROCEDURE ROTAT_240_ABOUT_Y_EQ_NEG_X_EQ_Z ;

(* This procedure effects a rotation of 240 degrees about the axis
        x=-y=-z and returns the new array in "after". *)

VAR i: INTEGER;   (*counting integer*)

        BEGIN
            move[1]:=6;
            move[2]:=4;
            move[3]:=2;
            move[4]:=3;
            move[5]:=1;
            move[6]:=5;
            FOR i:= 1 TO 6 DO
        BEGIN
            after[i,'x']:=-before[move[i],'z'];
            after[i,'y']:=-before[move[i],'x'];
            after[i,'z']:=before[move[i],'y']
        END
END;

PROCEDURE ROTATE_ABOUT_X ;

(* This procedure effects a rotation of 180 degrees about the x axis and
        returns the new array in "after". *)

VAR i: INTEGER;   (*counting integer*)

        BEGIN
            move[1]:=2;
            move[2]:=1;
            move[3]:=3;
            move[4]:=6;
            move[5]:=5;
            move[6]:=4;
            FOR i:= 1 TO 6 DO
        BEGIN
            after[i,'x']:=  before[move[i],'x'];
            after[i,'y']:=  -before[move[i],'y'];
              after[i,'z']:=  -before[move[i],'z']
        END
END;
```

```
PROCEDURE ROTATE_ABOUT_Y ;

(* This procedure effects a rotation of 180 degrees about the y axis and
      returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

      BEGIN
            move[1]:=2;
            move[2]:=1;
            move[3]:=5;
            move[4]:=4;
            move[5]:=3;
            move[6]:=6;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=  -before[move[i],'x'];
            after[i,'y']:=   before[move[i],'y'];
            after[i,'z']:=  -before[move[i],'z']
      END
END;

PROCEDURE ROTATE_ABOUT_Z ;

(* This procedure effects a rotation of 180 degrees about the z axis and
      returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

      BEGIN
            move[1]:=1;
            move[2]:=2;
            move[3]:=5;
            move[4]:=6;
            move[5]:=3;
            move[6]:=4;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=  -before[move[i],'x'];
            after[i,'y']:=  -before[move[i],'y'];
            after[i,'z']:=   before[move[i],'z']
      END;
END;
```

```
PROCEDURE ROTATE_ABOUT_Y_EQ_NEG_X ;

(* This procedure effects a rotation of 180 degrees about the
      y=-x axis and returns the new array in "after". *)


VAR i: INTEGER;   (*counting integer*)

      BEGIN
            move[1]:=2;
            move[2]:=1;
            move[3]:=4;
            move[4]:=3;
            move[5]:=6;
            move[6]:=5;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=   -before[move[i],'y'];
            after[i,'y']:=   -before[move[i],'x'];
            after[i,'z']:=   -before[move[i],'z']
      END
END;


PROCEDURE ROTATE_ABOUT_X_EQ_NEG_Z ;

(* This procedure effects a rotation of 180 degrees about the
      axis x=-z and returns the new array in "after". *)


VAR i: INTEGER;   (*counting integer*)

      BEGIN
            move[1]:=5;
            move[2]:=3;
            move[3]:=2;
            move[4]:=6;
            move[5]:=1;
            move[6]:=4;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=   -before[move[i],'z'];
            after[i,'y']:=   -before[move[i],'y'];
            after[i,'z']:=   -before[move[i],'x']
      END
END;
```

```
PROCEDURE ROTATE_ABOUT_Y_EQ_X ;

(* This procedure effects a rotation of 180 degrees about the
      axis x=y and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

      BEGIN
            move[1]:=2;
            move[2]:=1;
            move[3]:=6;
            move[4]:=5;
            move[5]:=4;
            move[6]:=3;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=  before[move[i],'y'];
            after[i,'y']:=  before[move[i],'x'];
            after[i,'z']:=  -before[move[i],'z']
      END
END;

PROCEDURE ROTATE_ABOUT_X_EQ_Z ;

(* This procedure effects a rotation of 180 degrees about the
      axis x=z and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

      BEGIN
            move[1]:=3;
            move[2]:=5;
            move[3]:=1;
            move[4]:=6;
            move[5]:=2;
            move[6]:=4;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=  before[move[i],'z'];
            after[i,'y']:=  -before[move[i],'y'];
            after[i,'z']:=  before[move[i],'x']
      END
END;
```

```
PROCEDURE ROTAT_120_ABOUT_NEG_Y_EQ_X_EQ_Z ;

(* This procedure effects a rotation of 180 degrees about the z axis and
      returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

      BEGIN
            move[1]:=4;
            move[2]:=6;
            move[3]:=1;
            move[4]:=3;
            move[5]:=2;
            move[6]:=5;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=before[move[i],'z'];
            after[i,'y']:=-before[move[i],'x'];
            after[i,'z']:=-before[move[i],'y']
      END
END;

PROCEDURE ROTAT_240_ABOUT_NEG_Y_EQ_X_EQ_Z ;

(* This procedure effects a rotation of 120 degrees about the
      axis x=-y=z and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

      BEGIN
            move[1]:=4;
            move[2]:=6;
            move[3]:=1;
            move[4]:=3;
            move[5]:=2;
            move[6]:=5;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=  before[move[i],'z'];
            after[i,'y']:=  -before[move[i],'x'];
            after[i,'z']:=  -before[move[i],'y']
      END
END;
```

```
PROCEDURE ROTAT_120_ABOUT_Y_EQ_X_EQ_NEG_Z;

(* This procedure effects a rotation of 120 degrees about the axis
     x=y=-z and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

     BEGIN
          move[1]:=4;
          move[2]:=6;
          move[3]:=2;
          move[4]:=5;
          move[5]:=1;
          move[6]:=3;
          FOR i:= 1 TO 6 DO
     BEGIN
          after[i,'x']:=-before[move[i],'z'];
          after[i,'y']:=before[move[i],'x'];
          after[i,'z']:=-before[move[i],'y']
     END
END;

PROCEDURE ROTAT_240_ABOUT_Y_EQ_X_EQ_NEG_Z;

(* This procedure effects a rotation of 240 degrees about the axis
     x=y=-z and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

     BEGIN
          move[1]:=5;
          move[2]:=3;
          move[3]:=6;
          move[4]:=1;
          move[5]:=4;
          move[6]:=2;
          FOR i:= 1 TO 6 DO
     BEGIN
          after[i,'x']:=before[move[i],'y'];
          after[i,'y']:=-before[move[i],'z'];
          after[i,'z']:=-before[move[i],'x']
     END
END;
```

```
PROCEDURE ROTATE_120_ABOUT_Y_EQ_X_EQ_Z;

(* This procedure effects a rotation of 120 degrees about the axis
      x=y=z and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

      BEGIN
            move[1]:=3;
            move[2]:=5;
            move[3]:=6;
            move[4]:=2;
            move[5]:=4;
            move[6]:=1;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=  before[move[i],'y'];
            after[i,'y']:=  before[move[i],'z'];
            after[i,'z']:=  before[move[i],'x']
      END
END;

PROCEDURE ROTATE_240_ABOUT_Y_EQ_X_EQ_Z;

(* This procedure effects a rotation of 240 degrees about the axis
      x=y=z and returns the new array in "after". *)

VAR i: INTEGER;  (*counting integer*)

      BEGIN
            move[1]:=6;
            move[2]:=4;
            move[3]:=1;
            move[4]:=5;
            move[5]:=2;
            move[6]:=3;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=before[move[i],'z'];
            after[i,'y']:=before[move[i],'x'];
            after[i,'z']:=before[move[i],'y']
      END
END;
```

```
PROCEDURE ROTATE_ABOUT_Y_EQ_NEG_Z;

(* This procedure effects a rotation of 180 degrees about the
      axis y=-z and returns the new array in "after". *)


VAR i: INTEGER;  (*counting integer*)

      BEGIN
            move[1]:=4;
            move[2]:=6;
            move[3]:=5;
            move[4]:=1;
            move[5]:=3;
            move[6]:=2;
            FOR i:= 1 TO 6 DO
      BEGIN
            after[i,'x']:=  -before[move[i],'x'];
            after[i,'y']:=  -before[move[i],'z'];
            after[i,'z']:=  -before[move[i],'y']
      END
END;


PROCEDURE COMPARE;

(* This subroutine compares the two matrices to check if they are equivalent
      note that a 180-degree rotation about the axis is counted as equivalent*)


VAR
      i: INTEGER;  (* counting integer*)

      BEGIN
            result:=true;
            i:=1;
            WHILE ((result = true) AND (i < 7)) DO
      BEGIN
            CASE i OF
            1,2:   If  ((before[i,'x']=after[i,'x'])
                   and(before[i,'y']=after[i,'y']))
            THEN
            BEGIN
                   result:=  true;
                   sign[i]:=  1;
            END
            ELSE
            IF  ((before[i,'x']=-after[i,'x'])
                   and(before[i,'y']=-after[i,'y']))
            THEN
```

```
                BEGIN
                        result:=true;
                        sign[i]:= -1;
                END
                ELSE  result:=false;
                3,5:    IF  ((before[i,'y']=after[i,'y'])
                        and(before[i,'z']=after[i,'z']))
                THEN
                BEGIN
                        result:=  true;
                        sign[i]:=  1;
                END
                ELSE
                IF  ((before[i,'y']=   -after[i,'y'])
                        and(before[i,'z']=-after[i,'z']))
                THEN
                BEGIN
                        result:=  true;
                        sign[i]:=  -1;
                END
                ELSE  result:=false;
                4,6:    IF  ((before[i,'x']=after[i,'x'])
                        and(before[i,'z']=after[i,'z']))
                THEN
                BEGIN
                        result:=  true;
                        sign[i]:=  1;
                END
                ELSE
                IF  ((before[i,'x']=   -after[i,'x'])
                        and(before[i,'z']=-after[i,'z']))
                THEN
                BEGIN
                        result:=true;
                        sign[i]:=  -1;
                END
                ELSE  result:=false;
        END;
                i:=i+1;
        END;
END;

PROCEDURE  combiner;

VAR
        a: INTEGER;    (* the row of the matrix*)
        b: INTEGER;
        i:  INTEGER;    (* counting integer*)
        j:  INTEGER;    (* counting integer*)
```

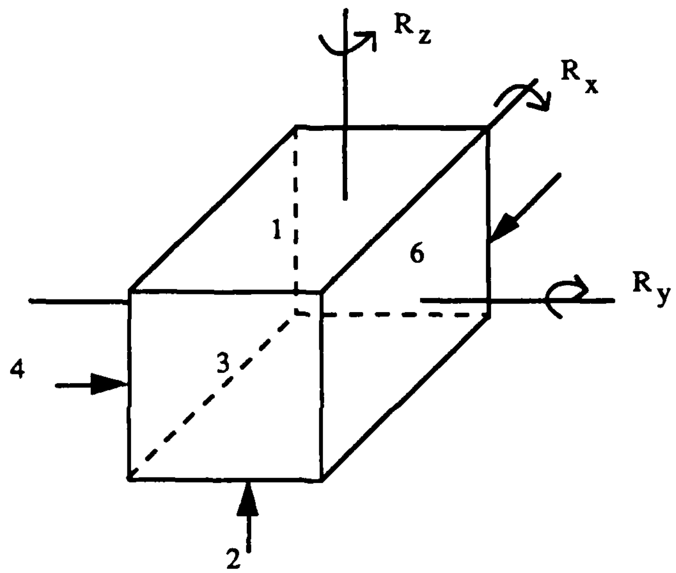
```
        l: INTEGER;     (* counting integer*)
        l2: INTEGER;    (* counting integer*)
        m: INTEGER;     (* counting integer*)
        n: INTEGER;     (* counting integer*)
        s4: ARRAY [1..6,1..6]  OF INTEGER; (* temporary storage for a matrix *)
        s5: INTEGER;
        sign: INTEGER;
        sign1: INTEGER;
        sign2: INTEGER;
        temp:  INTEGER;
        temp2: INTEGER;

BEGIN

(*Initialize the resultant matrix.*)
        FOR i:= 1 TO 6 DO
        FOR j:= 1 TO 6 DO
        s3[i,j]:= s1[i,j];

(*Begin finding the elements of the resultant matrix*)

        FOR j:=1 TO 6 DO (* i is the row of a matrix sent by the user*)
        FOR i:= 1 TO 6 DO (* j is the column of a matrix sent by the user*)
        BEGIN
        l2:=abs(s2[i,j]);
        a:=l2 div 10;
        b:=l2-a*10;
        IF((s3[a,b]<>0)AND(s3[i,j]<>0))
        THEN
        BEGIN
                sign:=s2[i,j] div l2;
                sign2:=s3[a,b] div abs(s3[a,b]);
        END;
        temp:=s3[i,j];
        temp2:=s3[a,b];
        l2:=abs(s3[a,b]);
        IF  (s3[a,b]=0)
        THEN
        BEGIN
                l2:=abs(temp);
                FOR m:=1 TO 6 DO
                FOR n:=1 TO 6 DO
                IF(abs(s3[m,n])=abs(s3[i,j])) THEN  s3[m,n]:=0;
        END
                ELSE
                FOR m:=1 TO 6 DO
                FOR n:=1 TO 6 DO
                IF(abs(s3[m,n])=l2)
                THEN  s3[m,n]:=temp*sign*sign2*(s3[m,n]  div  abs(s3[m,n]));
        END;
```

```
        FOR j:=1 TO 6 DO
        FOR i:=1 TO 6 DO
        IF(s2[j,i]=-(10*j+i))
        THEN
        BEGIN
                temp:=(10*j+i);
                FOR m:=1 TO 6 DO
                FOR n:=1 TO 6 DO
                IF(abs(s3[m,n])=temp) THEN s3[m,n]:=0;
        END;
END;

PROCEDURE operator_out(VAR vector : smatrix);

VAR I: INTEGER; (*counting integer for a FOR loop *)

BEGIN
        { FOR I:=1 to 6 DO
        WRITELN(LST,'           ( ',vector[I]:2,' )');
        WRITELN(LST,' ');}
END;

PROCEDURE S_Matrix_out(VAR Matrix:six_by_six);

VAR I,J: INTEGER; (*counting integers for FOR loops *)

BEGIN
        { FOR I:= 1 TO 6 DO
        BEGIN
                WRITE(LST,' (');
                FOR J:= 1 TO 6 DO
                WRITE (LST,matrix[i,j]:3,' ');
                WRITELN (LST,' )');
                WRITELN (LST,' (                    )');
                WRITELN (LST,' ');
        END }
END;
```

Appendix B

LIST OF THIRTY SYMMETRY OPERATORS FOR CUBIC TURNSTILE JUNCTION

ROTATIONAL SYMMETRY ABOUT MAJOR AXES

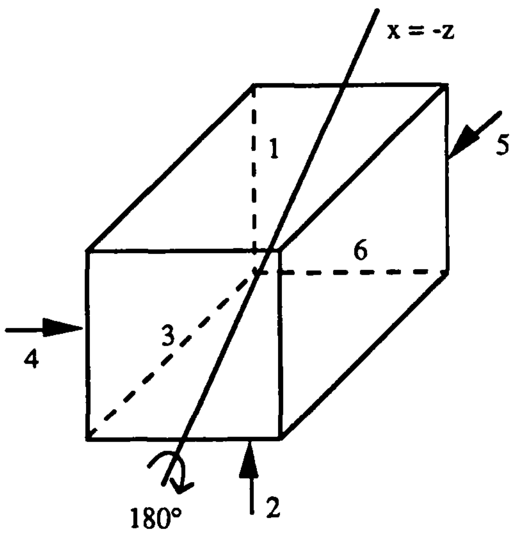


Since the waveguides are rectangular, the symmetrics are 180° rotations

$$R_z = \begin{bmatrix} \pm1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \pm1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad R_x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \pm1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \pm1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad R_y = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \pm1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \pm1 \end{bmatrix}$$

rotate z	rotate x	rotate y
rotate-about-z	rotate-about-x	rotate-about-y
$x \to -x$	$x \to x$	$x \to -x$
$y \to -y$	$y \to -y$	$y \to y$
$z \to z$	$z \to -z$	$z \to -z$

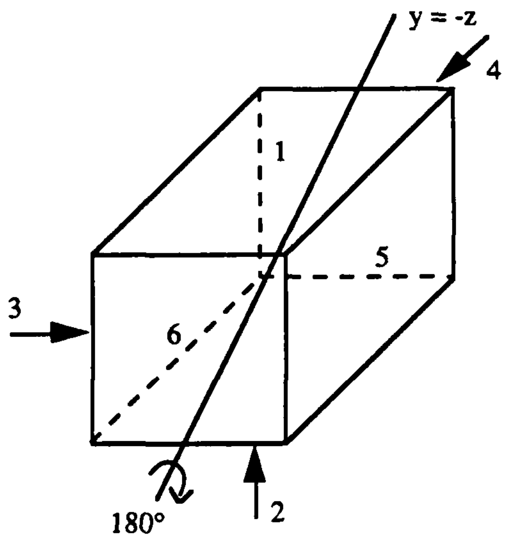
ROTATIONAL SYMMETRY ABOUT DIAGONAL EDGE CENTER TO EDGE CENTER



$$R_{E1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

rotate-about-x-eq-neg-z

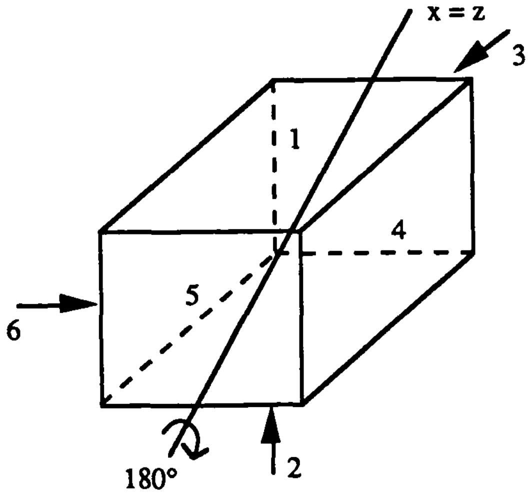
$-x \rightarrow z$
$-y \rightarrow y$ roxequnegz
$-z \rightarrow x$



$$R_{E2} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

rotate-about-y-eq-neg-z

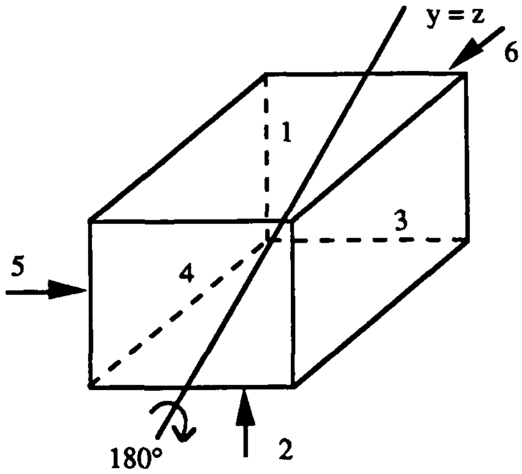
$-x \rightarrow x$
$-y \rightarrow z$ royeqnegz
$-z \rightarrow y$



$$R_{E3} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

rotate-about-x-eq-z

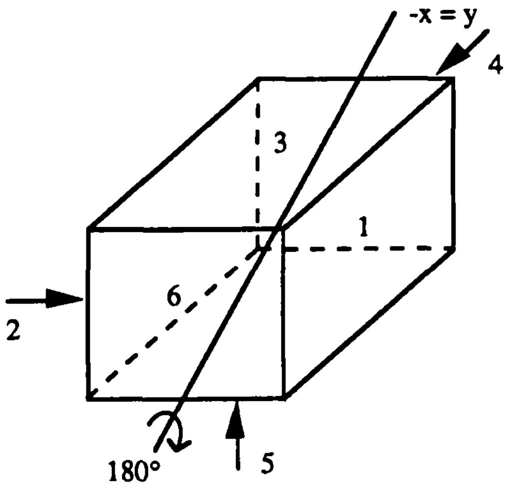
$x \rightarrow z$
$z \rightarrow x$ roxeqz
$-y \rightarrow y$



$$R_{E4} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

rotate-about-y-eq-z

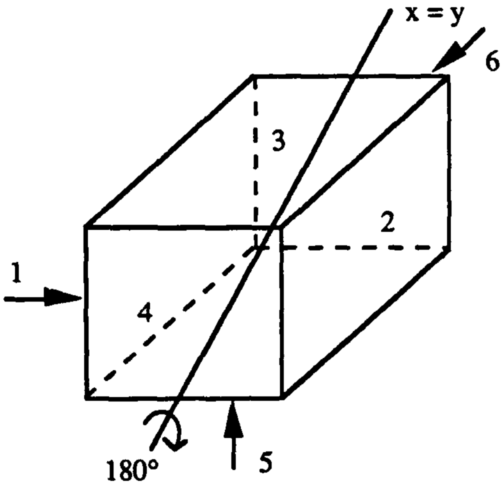
$y \rightarrow z$
$z \rightarrow y$ royeqz
$x \rightarrow -x$



$$R_{E5} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

rotate-about-y-eq-neg-x

$-x \rightarrow y$
$-y \rightarrow x$ royeqnegy
$-z \rightarrow z$



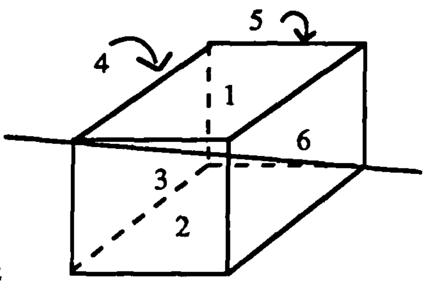
$$R_{E6} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

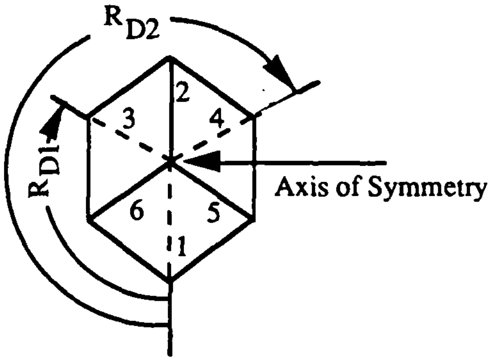
rotate-about-y-eq-x

$-z \rightarrow z$
$y \rightarrow x$ roxeqx
$x \rightarrow y$

ROTATIONAL SYMMETRY ABOUT A DIAGONAL - CORNER TO CORNER

(-1, -1, 1)
(0, 0, -0)
(-1, 1, -1)
x = -y = z





$$R_{D1} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

rotate-120-about-neg-y-eq-x-eq-z

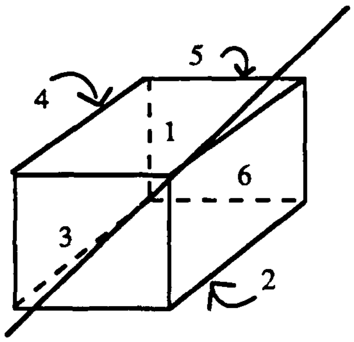
$z \rightarrow x$
$-x \rightarrow y$ roxnegyz 1
$-y \rightarrow z$

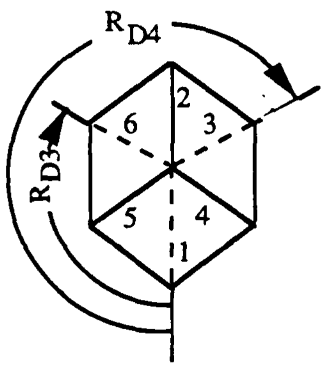
$$R_{D2} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

rotate-240-about-neg-y-eq-x-eq-z

$-z \rightarrow y$
$-y \rightarrow x$ roxnegyz 2
$x \rightarrow z$

(1, 1, 1)
(0, 0, 0)
(-1, -1, -1)
x = y = z





$$R_{D3} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

rotate-120-about-y-eq-x-eq-zx

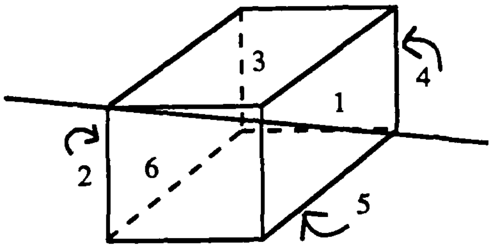
$y \rightarrow x$
$x \rightarrow z$ roxyz 1
$z \rightarrow y$

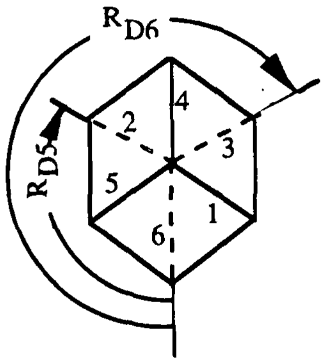
$$R_{D4} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

rotate-240-about-y-eq-x-eq-z

$z \rightarrow x$
$x \rightarrow y$ roxyz 2
$y \rightarrow z$

(1, 1, -1)
(-1, -1, 1)
(0, 0, 0)
x = y = -z





$$R_{D5} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

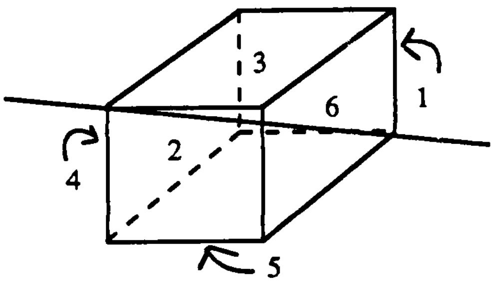
rotate-120-about-y-eq-x-eq-neg-z

$-y \rightarrow z$
$-z \rightarrow x$ roxynegz 1
$x \rightarrow y$

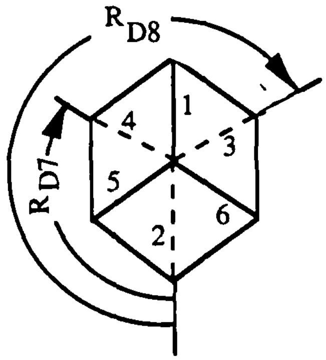
$$R_{D6} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

rotate-240-about-y-eq-x-eq-neg-z

$-x \rightarrow z$
$y \rightarrow x$ roxynegz 2
$-z \rightarrow y$



x = -y = -z



$$R_{D7} = \begin{bmatrix} 0&0&0&0&1&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&1&0&0&0&0 \\ 0&0&0&0&0&1 \\ 1&0&0&0&0&0 \end{bmatrix}$$

rotate-120-about-y-eq-neg-x-eq-z

z → y
y → -x   ronegxyz 1
x → -z

$$R_{D8} = \begin{bmatrix} 0&0&0&0&0&1 \\ 0&0&0&1&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 1&0&0&0&0&0 \\ 0&0&0&0&1&0 \end{bmatrix}$$

rotate-240-about-y-eq-neg-x-eq-z

- x → y
y → z   ronegxyz 2
-z → x

MIRROR SYMMETRY ABOUT A POINT

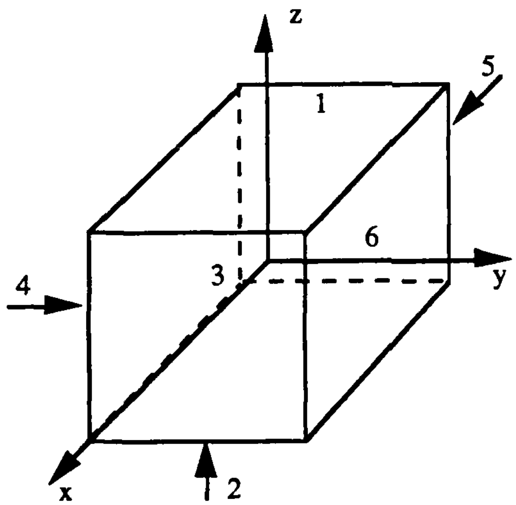
The only point of symmetry is the center of the cube.

$$M_O = \begin{bmatrix} 0&1&0&0&0&0 \\ 1&0&0&0&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \end{bmatrix}$$

mirorigin

x → -x
y → -y
z → -z

MIRROR SYMMETRY ABOUT LINES



Mirror symmetry about the x-axis z → -z y → -y x → -x

$$M_x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Mirror symmetry about the y-axis z → -z y → y x → -x

$$M_y = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

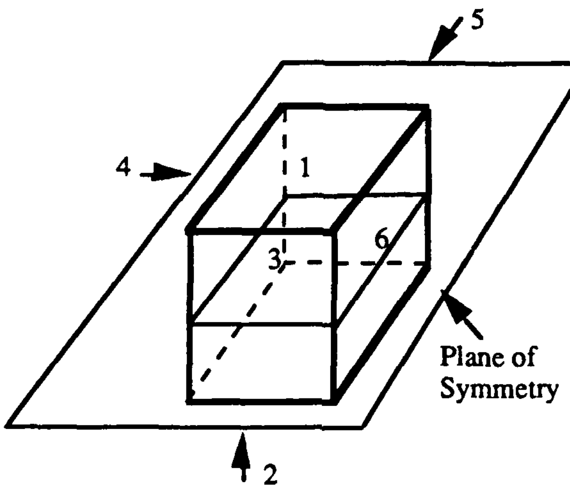
Mirror symmetry about the z-axis z → z y → -y x → -x

$$M_z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Note that $M_z = R_z$ (rotation about the z-axis of 180°)
$M_y = R_y$
$M_x = R_x$

Since any line is an axis line for a coordinate system any mirror about a line is the same as a rotation of 180° about that line.

MIRROR SYMMETRIES

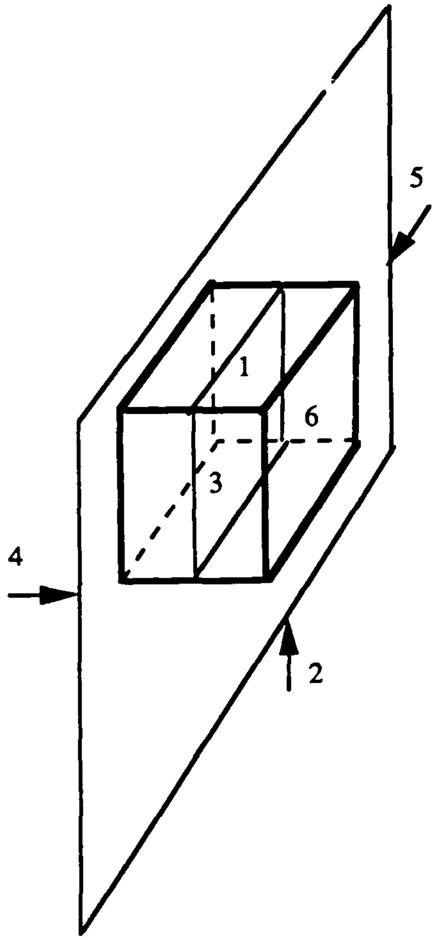


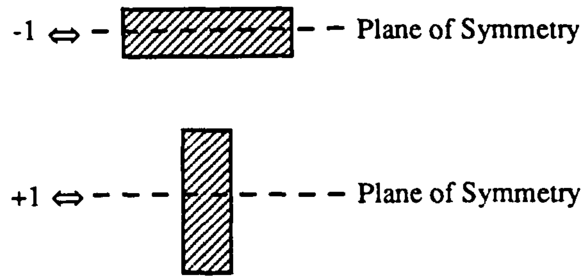
$$M_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \pm1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \pm1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \pm1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \pm1 \end{bmatrix}$$

x-y plane
z → -z
x → x
y → y

mirror xy

mirror-about-x-y-plane

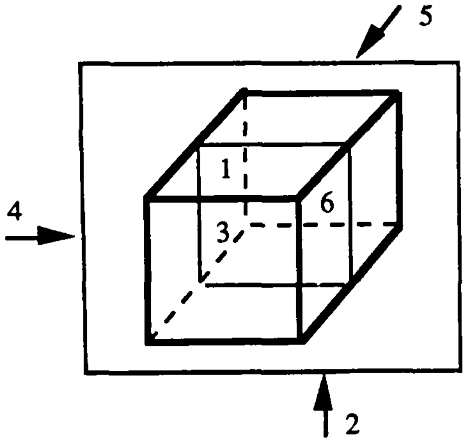




$$M_2 = \begin{bmatrix} \pm1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \pm1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \pm1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \pm1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

x-z plane
$x \rightarrow x$
$y \rightarrow -y$
$z \rightarrow z$

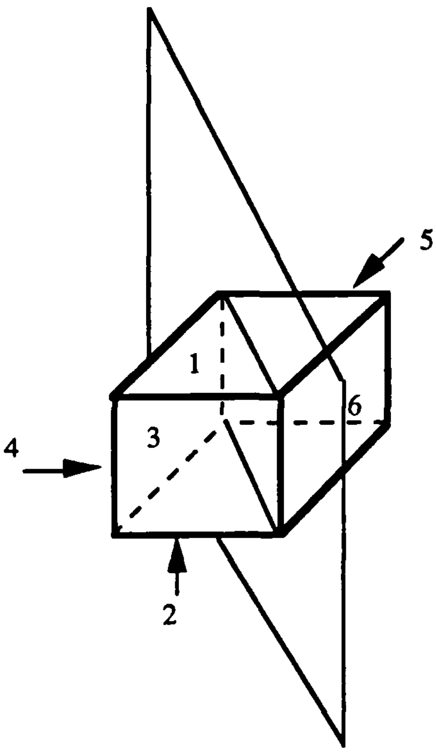
mirror xz
mirror-about-x-z-plane



$$M_3 = \begin{bmatrix} \pm1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \pm1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \pm1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \pm1 \end{bmatrix}$$

y-z plane
$x \rightarrow -x$
$y \rightarrow y$
$z \rightarrow z$

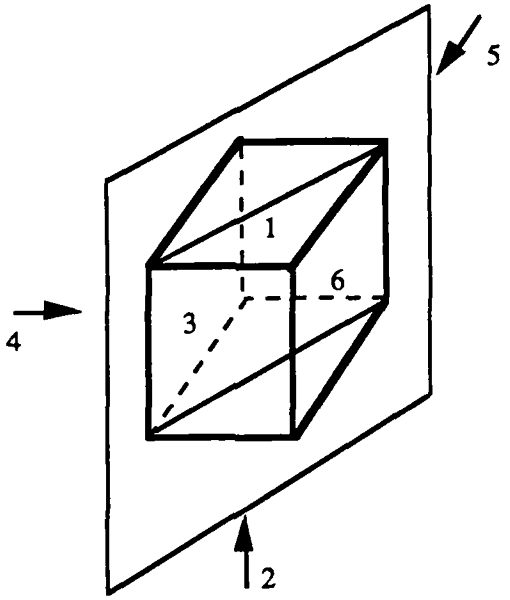
mirror yz
mirror-about-y-z-plane



$$M_4 = \begin{bmatrix} \pm1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \pm1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

y = x plane
$z \rightarrow z$
$x \rightarrow y$
$y \rightarrow x$

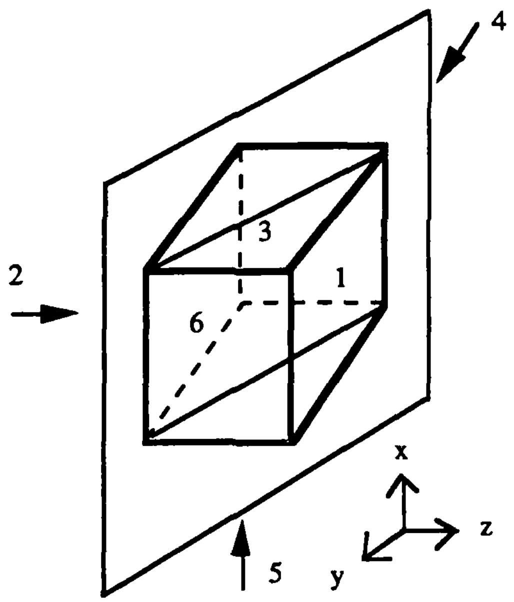
mirorxeqy
mirror-about-x-eq-y-plane



x = -y plane
x → -y
y → -x
z → z

$$M_5 = \begin{bmatrix} \pm 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \pm 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

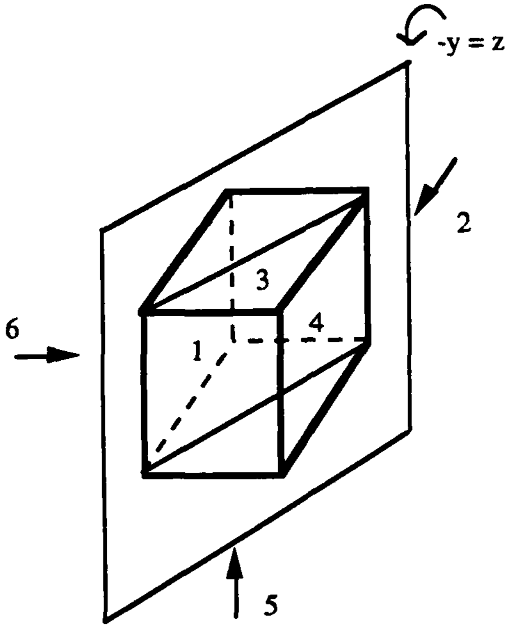
mrexeqnegy
mirror-about-x-eq-minus-y-plane



-y + = 2x plane
z → -y
y → -z
x → x

$$M_6 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \pm 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \pm 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

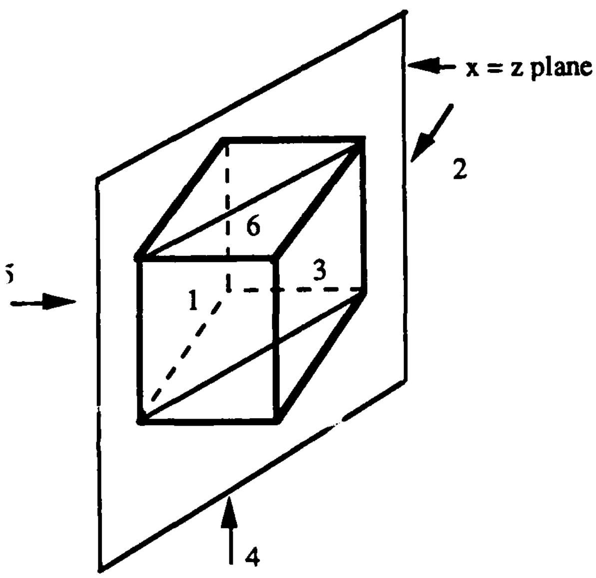
mirorzeqy
mirror-about-z-eq-y-plane



$$M_7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \pm1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \pm1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$z \rightarrow y$
$x \rightarrow x$
$y \rightarrow z$
$z + y = 2x$

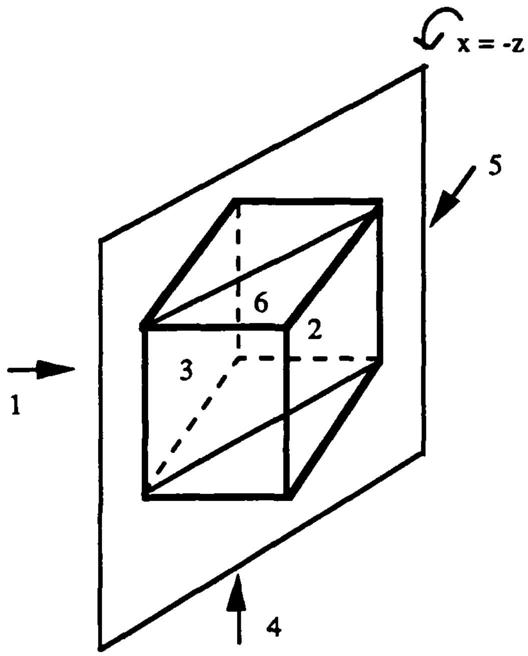
mrzeqnegy
mirror-about-z-eq-neg-y-plane



$$M_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \pm1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \pm1 \end{bmatrix}$$

$z \rightarrow -x$
$x \rightarrow -z$
$y \rightarrow y$

mirorxeqz
mirror-about-x-eq-z-plane



$$M_9 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \pm 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \pm 1 \end{bmatrix}$$

$x \rightarrow z$
$y \rightarrow y$
$z \rightarrow x$

mrxeqnegz
mirror-about-x-eq-minus-z-plane

Appendix C

THREE-DIMENSIONAL MODEL OF CUBIC TURNSTILE JUNCTION

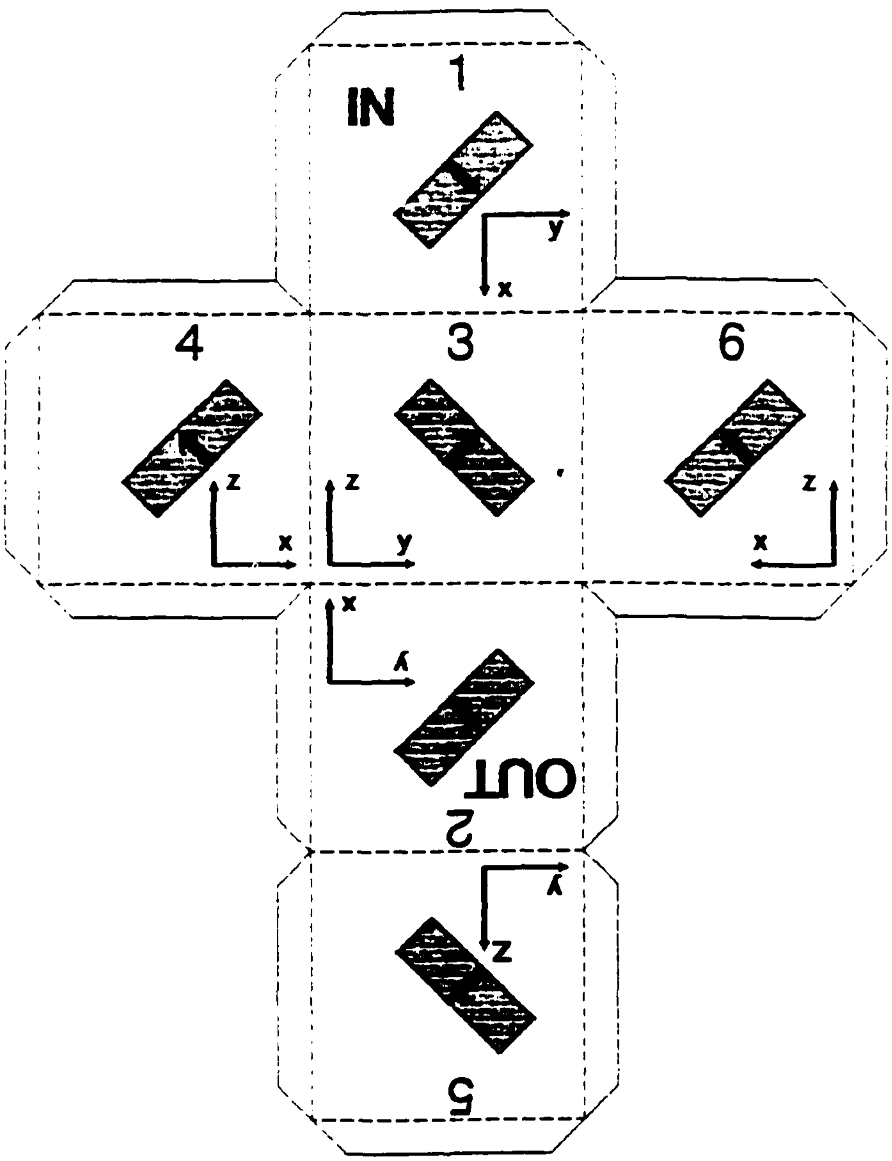


FIGURE C-1. Three-Dimensional Model of Cubic Turnstile Junction.

Appendix D

ADDITIONAL EXPERIMENTAL INFORMATION REGARDING KUROKAWA OSCILLATOR DESIGN

The basic cavity design layout is shown below. The optimization of the design involves proper choice of iris diameter, coaxial line placement, and cavity length. The coaxial line placement is specified by the distance between the sidewall of the cavity and the center line of the coaxial center conductor.

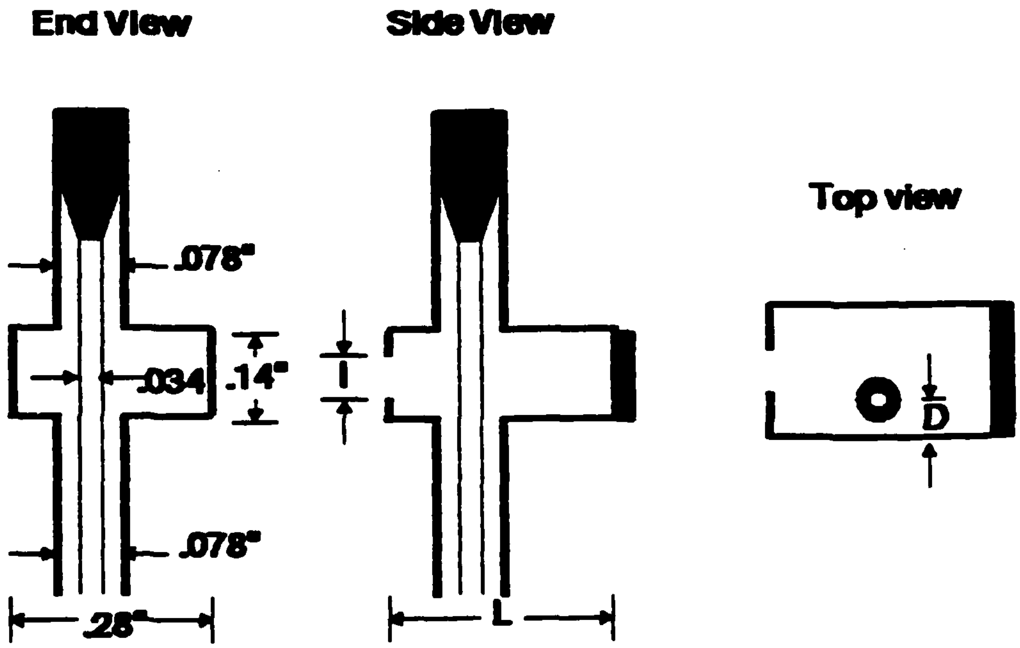


Figure D-1 is a plot of the proper cavity length and sidewall to coax separation (as a function of iris diameter) that results in a resonant frequency of 35 GHz. The data of Figure D-1 were generated by constructing a series of four oscillators with sidewall separations of 0.032, 0.048, 0.058, and

0.079 inches. Each oscillator was tested with various iris diameters by adjusting a movable backshort for maximum transfer efficiency at 35 GHz.

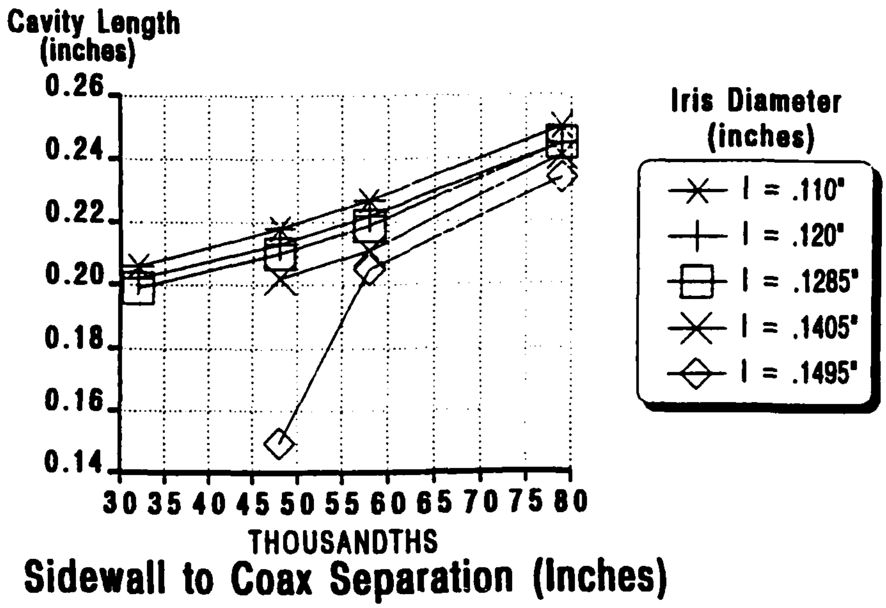


FIGURE D-1. Kurokawa Cavity Dimensions at 35 GHz.

Figure D-2 is a plot of the transfer efficiencies achieved for each of the configurations.

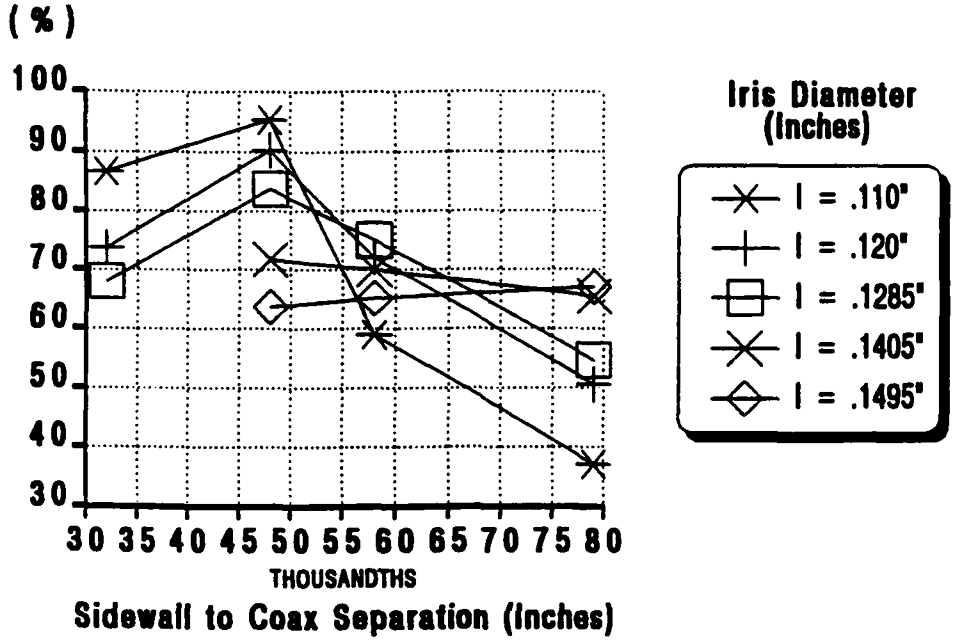


FIGURE D-2. Maximum Transfer Efficiency at 35 GHz.

Figures D-3 through D-6 show the efficiency bandwidth versus frequency characteristics for each configuration.

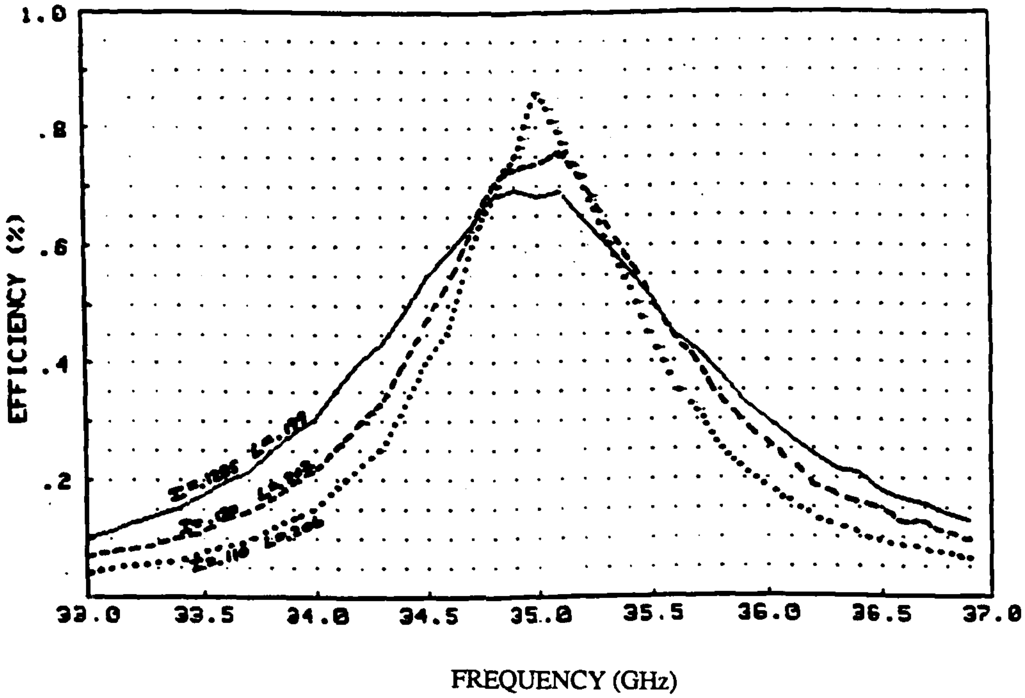


FIGURE D-3. D = 0.032.

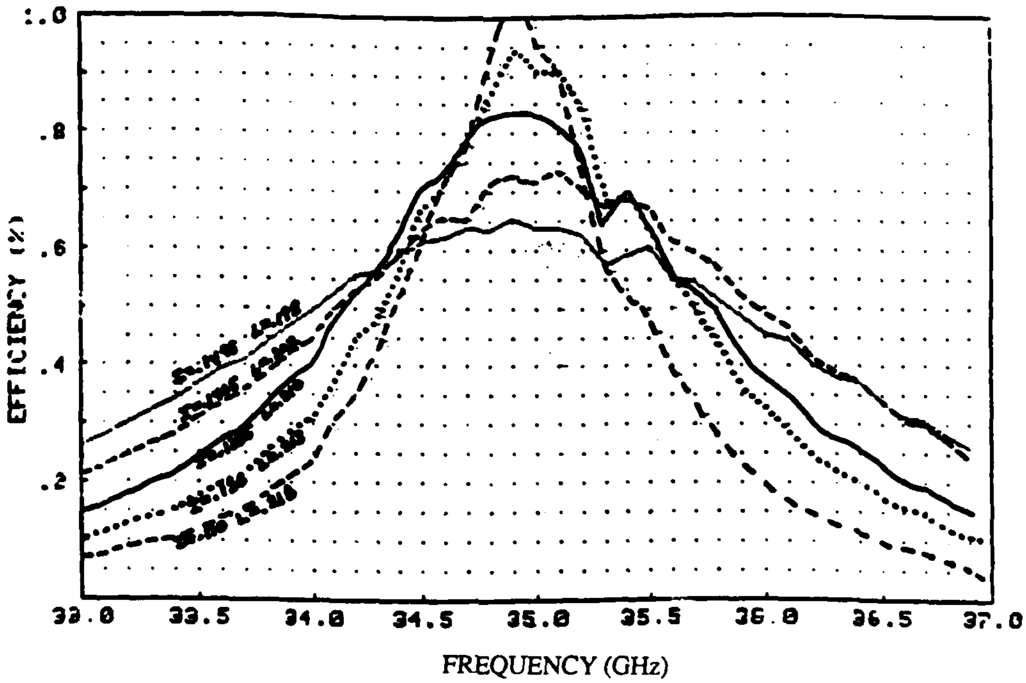


FIGURE D-4. D = 0.048.

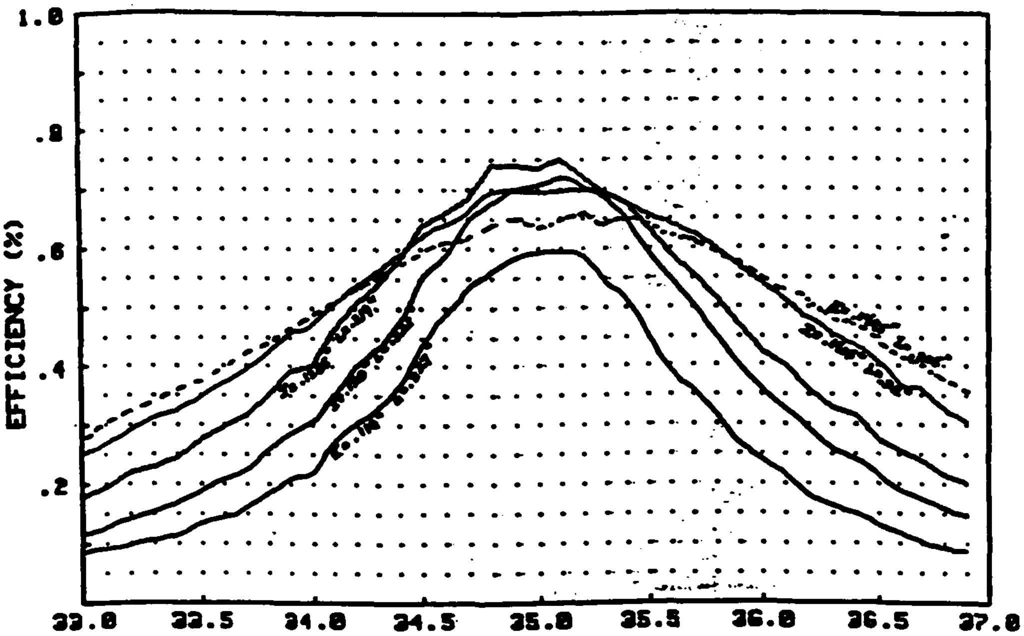


FIGURE D-5. D = 0.058.

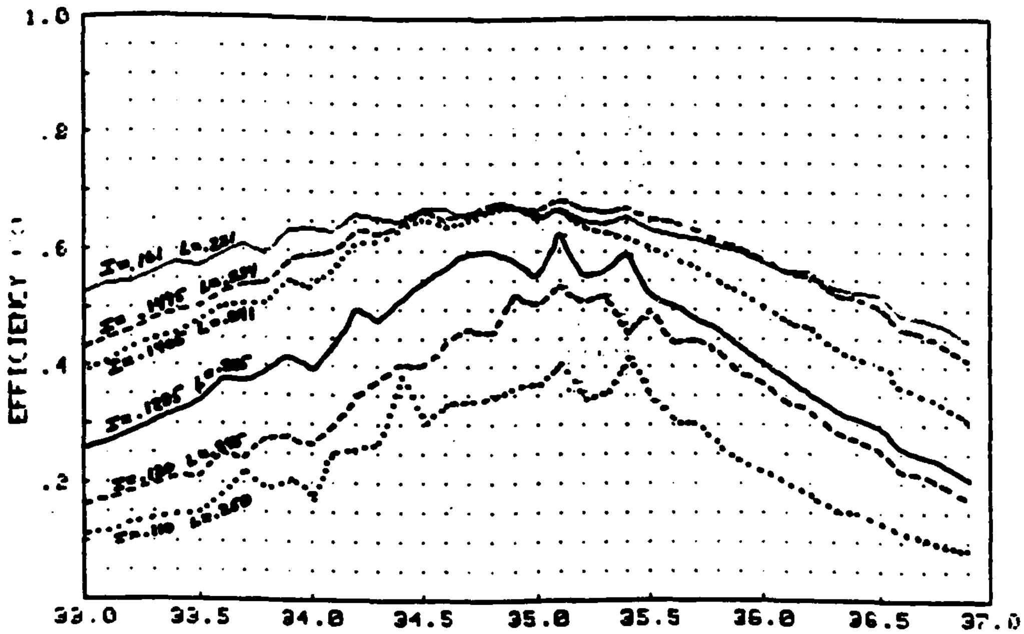


FIGURE D-6. D = 0.079.

Appendix E

EXPRESSION FOR POWER-ADDED EFFICIENCY OF REFLECTION AMPLIFIER

The power-added efficiency of such a circuit can be defined in terms of the incident (a_i's) and reflected (b_i's) waves at both ports:

$$\text{power-added efficiency} = \frac{\text{power added at port 1}}{\text{power added at port 2}} = \frac{|b_1|^2 - |a_1|^2}{|a_2|^2 - |b_2|^2}$$

The intent of this Appendix is to derive an expression for power-added efficiency in terms of Γd and the scattering- (S) parameters of the two-port network. The network equation associated with two-port S-parameters is

$$b_1 = S_{11}a_1 + S_{12}a_2$$

$$b_2 = S_{21}a_1 + S_{22}a_2 \qquad \text{(E-1)}$$

Expressing reflection coefficients as ratios of incident and reflected waves,

$$\Gamma_1 = \frac{b_1}{a_1} \;;\; \Gamma_d = \frac{a_2}{b_2} \qquad \text{(E-2)}$$

Combining Equations E-1 and E-2,

$$\frac{b_1}{a_1} = S_{11} + \frac{S_{12}S_{21}\Gamma_d}{1 - S_{22}\Gamma_d} = \frac{S_{11} - \Delta\Gamma_d}{1 - S_{22}\Gamma_d} \text{ where } \Delta = \det\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \qquad \text{(E-3)}$$

Or in terms of power,

$$|b_1|^2 = \left|\frac{S_{11} - \Delta\Gamma_d}{1 - S_{22}\Gamma_d}\right|^2 |a_1|^2 \qquad \text{(E-4)}$$

$$\text{power-added efficiency} = \eta = \frac{|b_1|^2 - |a_1|^2}{|a_2|^2 - |b_2|^2}$$

$$\eta = \frac{\left(\dfrac{|b_1|^2}{|a_1|^2} - 1\right)|a_1|^2}{\left(\dfrac{|a_2|^2}{|b_2|^2} - 1\right)|b_2|^2} \tag{E-5}$$

Substituting Equation E-4 in Equation E-5,

$$\eta = \frac{\left\{\left|\dfrac{S_{11} - \Delta\Gamma d}{1 - S_{22}\Gamma d}\right|^2 - 1\right\}}{\left(|\Gamma d|^2 - 1\right)} \frac{|a_1|^2}{|b_2|^2} \tag{E-6}$$

From the equation for b_2 in Equation E-1 using $\Gamma_d = \dfrac{a_2}{b_2}$,

$$b_2 = S_{21}a_1 + S_{22}\Gamma d\, b_2$$

$$S_{21}a_1 = (1 - S_{22}d)\, b_2$$

$$\frac{|a_1|^2}{|b_2|^2} = \frac{|1 - S_{22}\Gamma_d|^2}{|S_{21}|^2} \tag{E-7}$$

Combining Equations E-6 and E-7 yields the final result

$$\eta = \frac{|S_{11} - \Delta\Gamma d|^2 - |1 - S_{22}\Gamma d|^2}{|S_{12}|^2\left(|\Gamma d|^2 - 1\right)} \tag{E-8}$$

but note that $\Gamma_d = \Gamma_d(b_2)$, i.e., the diode reflection coefficient is a function of incident wave. For a free-running oscillator (a_i = 0; no injection-locking signal),

$$\eta_{osc} = \frac{|b_1|^2}{|a_2|^2 - |b_2|^2}$$

and

$$b_1 = S_{12}a_2$$

$$b_2 = S_{22}a_2$$

$$= \frac{|S_{12}|^2\,|a_2|^2}{(1 - |S_{22}|^2)\,|a_2|^2}$$

$$\eta_{osc} = \frac{|S_{12}|^2}{\left(1 - |S_{22}|^2\right)} \qquad \text{(E-9)}$$

Oscillation efficiency is only a function of the S-parameters of the two-port network; whereas, amplifier efficiency is also a function of the device reflection coefficient.

Appendix F

EFFECT OF IRIS SIZE, COAXIAL LINE PLACEMENT, AND CAVITY LENGTH UPON MEASURED MIDCAVITY IMPEDANCE

The specification of iris size (I), coaxial line placement (D), and cavity length (L) is explained in Appendix D. Figures F-1 through F-3 present midcavity impedance data for various values of iris size, cavity length, and separation between sidewall and coaxial line.

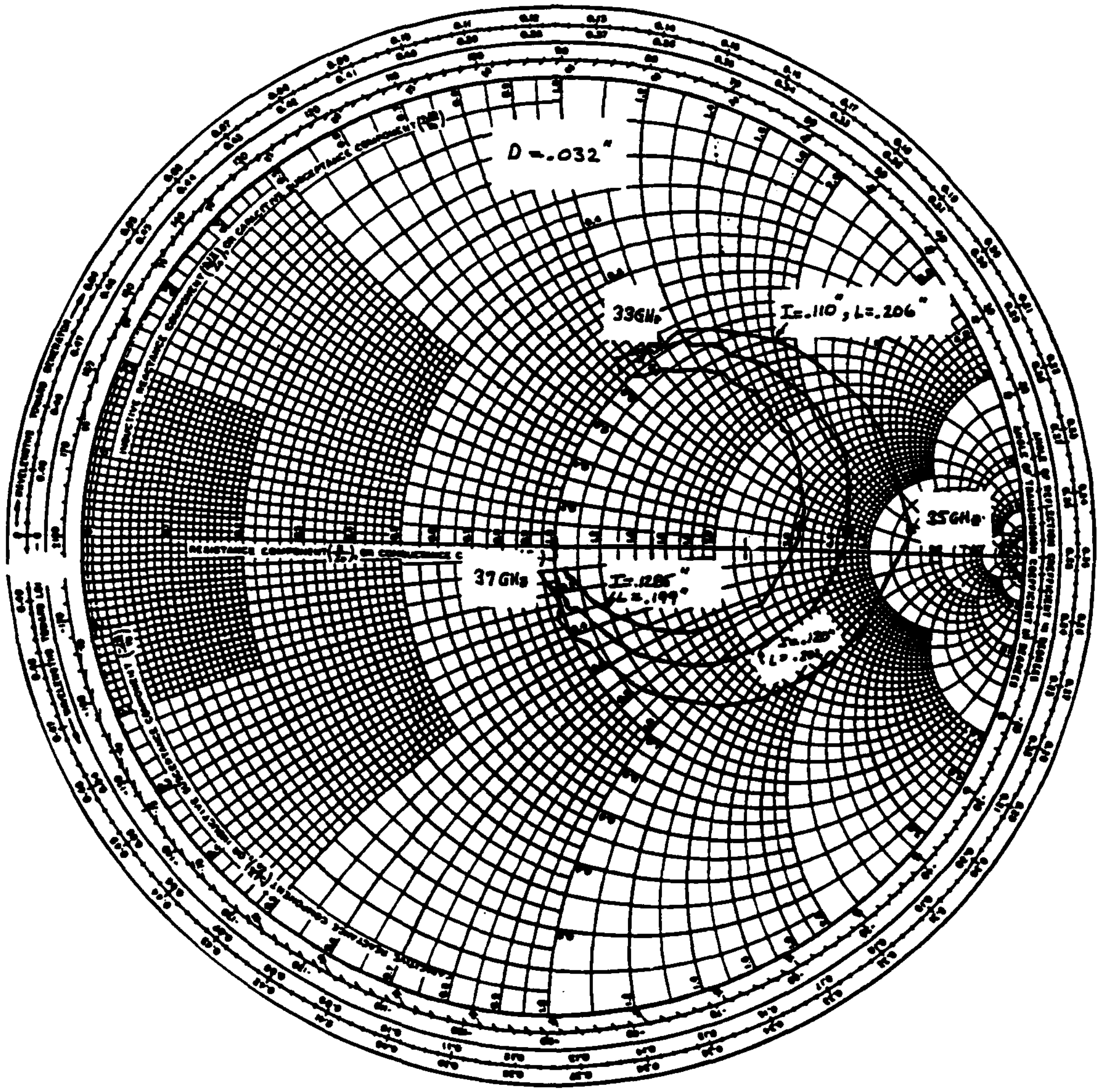


FIGURE F-1. D = 0.032 Inch.

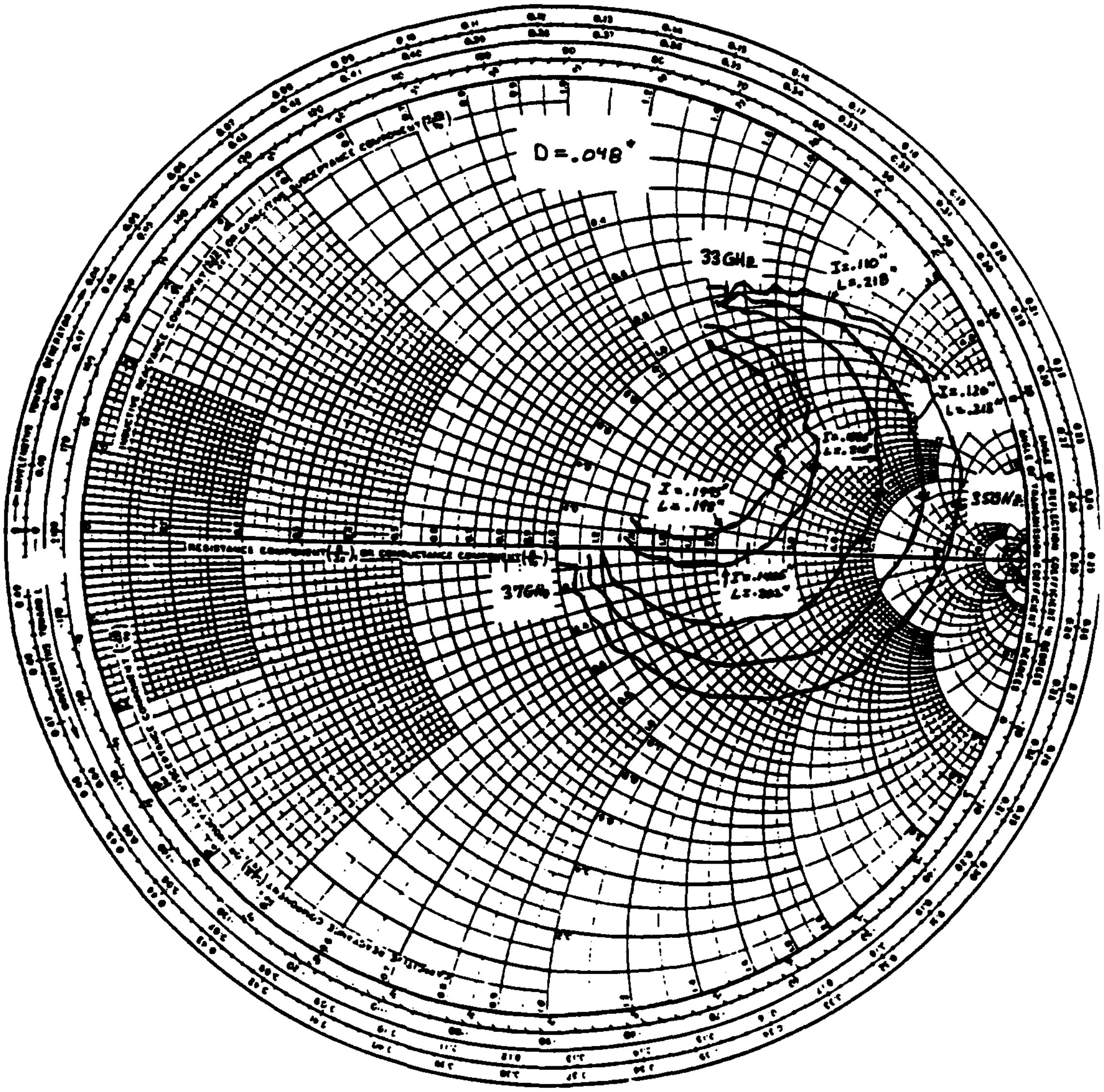


FIGURE F-2. D = 0.048 Inch.

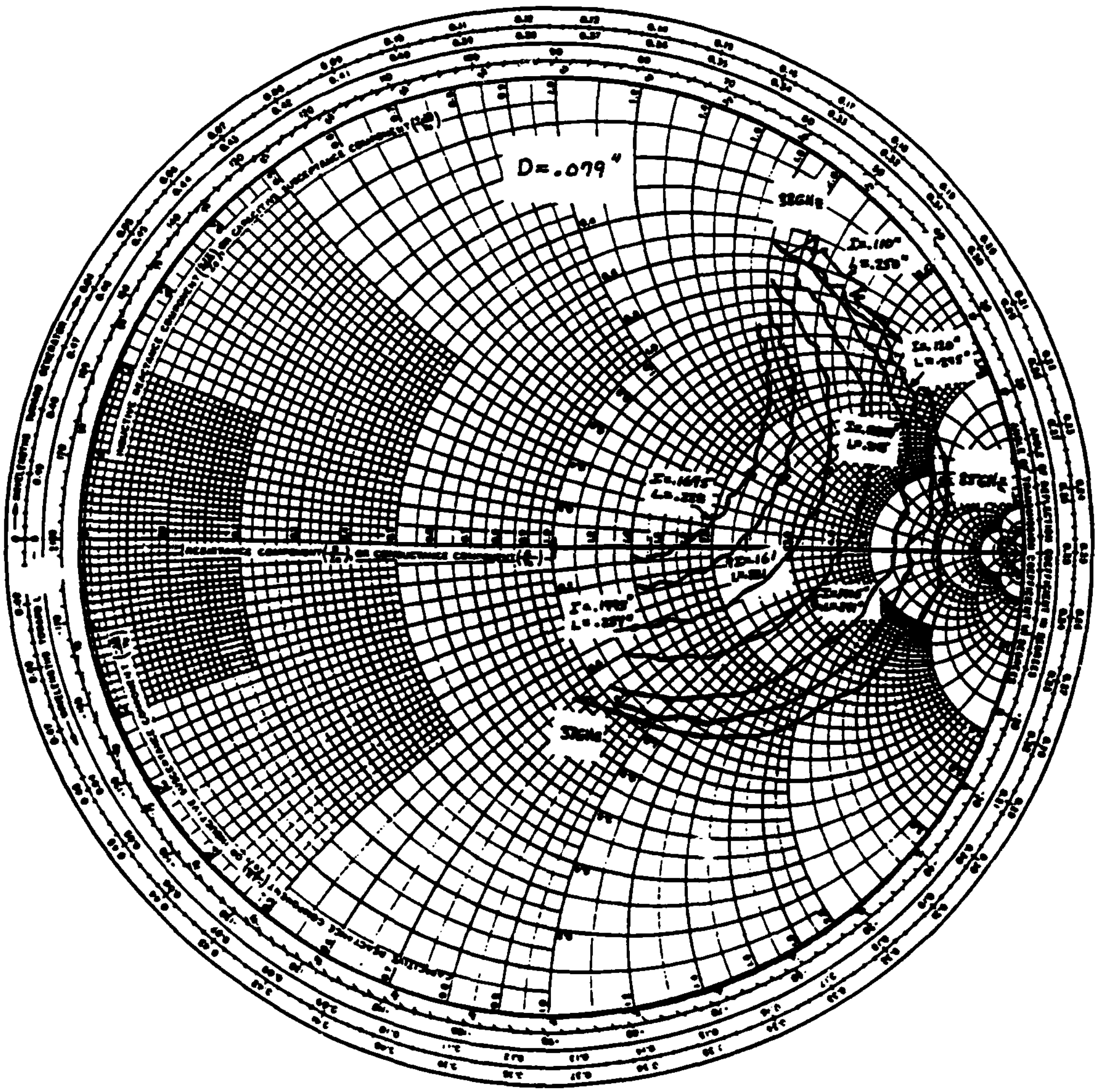


FIGURE F-3. D = 0.079 Inch.

Appendix G

PACKAGE AND MOUNTING PARASITICS FOR Ka-BAND IMPATT USED IN TURNSTILE OSCILLATORS

Details of the package and mounting parasitics have been given elsewhere (Reference 6). Table G-1 summaries these results. Figure G-1 presents a diagram of the circuit model used to represent package parasitics.

TABLE G-1. Calculated Lumped Parasitic Elements. Values are calculated for alumina package with outer diameter of 0.032 inch, inner diameter of 0.016 inch, and height of 0.008 inch.

Element	Symbol	Value
Strap inductance	Ls	0.024 nH
Package inductance	Lp	0.028 nH
Mount inductance	Lm	0.010 nH
Package capacitance	Cp	0.154 pF
Mount capacitance	Cm	0.028 pF

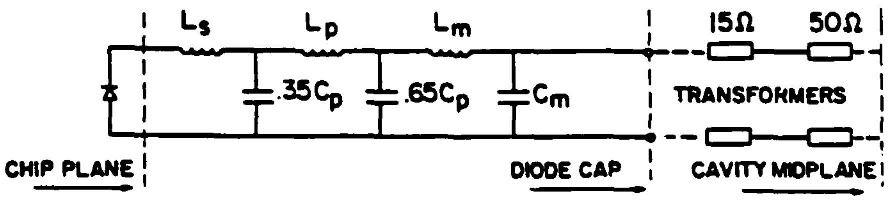


FIGURE G-1. Circuit Model Representing Package Parasitics.

Appendix H

COMPUTER PROGRAM FOR MODELING IMPATT IMPEDANCE

```
0   OPTION BASE 1
11      COM  Yra(171),Yia(171),Zra(171),Zia(171),Zpr(171),Zpi(171)
12      DIM  Crtr(6,3,3),Crti(6,3,3),Ar(3),Br(7,18),Bi(7,18)
14      COM  /Plut/  Xmin,Xmax,Xstep,Ymin,Ymax,Ystep,Pltt,Xindo,Yindo
15   Np=18
16    Indo=0
17   RAD
18   Op=CRT
19   !PACKAGE PARASITICS
21    Lsp=2.4E-11
22    Lpp=2.8E-11
23    Lmp=1.E-11
24    Cpp=1.54E-13
25    Cmp=2.8E-14
26   !DIODE MODEL PARAMETERS
40    Vs=4.53E+4
50    Alp=.157
60    Er=12.5
70    E0=8.884E-12
90    Ln=4.2E-7
100   Lp=4.2E-7
120   Gma=.7
121   A=1.21E-8 !DIODE AREA
122   !OPERATING CONDITIONS
123   Idc=1.5 !DC BIAS CURRENT
155   FOR Vf=0 TO 18 STEP 3   !LOOP TO INCREMENT RF VOLTAGE
158    IF Vf=0 THEN Vrf=1.E-6
159    IF Vf<>0 THEN Vrf=Vf
161   FOR J=1 TO Np
162     F=(33+(J-1)*1)*1.E+9
164 Op=CRT
165   OUTPUT Op;"FREQ=";F
167    W2=FNW2(F)
180     Wa2=FNWa2(Vs,Alp,Idc,Er,E0,A)
200      Thetap=FNFntpn(F,Lp,Vs)
220      Thetan=FNFntpn(F,Ln,Vs)
240     Cn=FNCncp(Er,E0,A,Ln)
250     Cp=FNCncp(Er,E0,A,Lp)
```

```
251   C=Cn*Cp/(Cn+Cp)
272  Gma=1
273   Gama2=1
275  IF Gma<>Gama2 THEN Gma=Gama2
280      B=FNB(Alp,Vrf,Thetan,Thetap,Gma)
300    Hob=FNHob(B)
320    Wr2=Wa2*Hob
322       Gama2=FNGnm(W2,Wr2,Ln,Lp,Thetap,Thetan)
323   IF NOT (ABS(Gama2-Gma)<.005) THEN 275
340      CALL  Fnfp(Thetap,Fpr,Fpi)
350      CALL  Fnfp(Thetan,Fnr,Fni)
400      Numy_i=W2^.5*C*(1-Wr2/W2)
410        Rldem=1-(Wr2/W2)*C*((1+Fnr)/Cn+(1+Fpr)/Cp)
420        Imdem=-(Wr2/W2)*C*(Fni/Cn+Fpi/Cp)
430      CALL  Cdiv(0,Numy_i,Rldem,Imdem,Yr,Yi)
431     CALL  Cdiv(1,0,Yr,Yi,Zra(J),Zia(J))
437     CALL  Zpack(Zra(J),Zia(J),F,Lsp,Lpp,Lmp,Cpp,Cmp,Zpr(J),Zpi(J))
451     Yra(J)=Yr*1000
452    Yia(J)=Yi*1000
464    IF J MOD 1=0 THEN
465      Cvbt=Cvbt+1
467        Br(Vf/3+1,Cvbt)=Zra(J)
468        Bi(Vf/3+1,Cvbt)=Zia(J)
469    END IF
471  NEXT J
474     CALL  Plot(Zpr(*),Zpi(*),Np,Indo)
475  Cvbt=0
477 NEXT Vf
478 FOR I=1 TO 18
479 FOR J=1 TO 7
480   Zra(J)=Br(J,I)
481   Zia(J)=Bi(J,I)
482 NEXT J
483  PENUP
484    CALL  Plot(Zra(*),Zia(*),7,Indo)
485  PENUP
486 NEXT I
487  INPUT "WANT ANOTHER PLOT? ",Ans$
488  IF UPC$(Ans$[1,1])="Y" THEN 472
489  END
490      !************************** BESSEL FUNCTION IO ***************
492   DEF FNI0(B)
493   T=B/3.75
494  IF B>=3.75 THEN
500   K1=B^.5
510   K2=EXP(-B)
520      P=.39894228+.01328592*T^(-1)+.00225319*T^(-2)-.00157565*T^(-3)
         +.00916281*T^(-4)-.02057706*T^(-5)+.02635537*T^(-6)-.01647633*T^(-7)
   +.00392377*T^(-8)
530    P2=P/(K1*K2)
```

```
540  ELSE
550  IF ABS(B)<3.75 THEN
560     P2=1+3.5156229*T^2+3.0899424*T^4+1.2067492*T^6+.2659732*T^8+.0360768*
       T^10+.0045813*T^12
570  ELSE
580  BEEP
590  PRINT "B= "&VAL$(B)&" B OUT OF RANGE IN I0"
600  PAUSE
610  END IF
620  END IF
630  RETURN P2
640  FNEND
641      !************************** BESSEL  FUNCTION  I1 ***************
650   DEF FNI1(B)
660   T=B/3.75
670  IF B>=3.75 THEN
680   K1=B^(.5)
690   K2=EXP(-B)
700      P=.39894228-.03988024*T^(-1)-.00362018*T^(-2)+.00163801*T^(-3)
       -.01031555*T^(-4)+.02282967*T^(-5)-.02895312*T^(-6)
       +.01787654*T^(-7)-.00420059*T^(-8)
710    P2=P/(K1*K2)
720  ELSE
730  IF ABS(B)<3.75 THEN
740   K1=B^(-1)
750
P=.5+.87890594*T^2+.51498869*T^4+.15084934*T^6+.02658733*T^8+.00301532*
    T^10+.00032411*T^12
760    P2=P/K1
770  ELSE
780  PRINT "B= "&VAL$(B)&" B OUT OF RANGE IN I1"
790  BEEP 89,1
800  PAUSE
810  END IF
820  END IF
830  RETURN P2
840  FNEND
841          !**********************************************************
850    DEF FNFa(Vs,Alp,Idc,Er,E0,A)
860       P=(.5/PI)*((3*Vs*Alp*Idc)/(Er*E0*A))^.5
870  RETURN P
880  FNEND
881          !**********************************************************
890    DEF  FNFntpn(Freq,L,Vs)
900     P=2*PI*Freq*L/Vs
910  RETURN P
920  FNEND
921          !**********************************************************
930   DEF FNCncp(Er,E0,A,L)
940    P=Er*E0*A/L
```

```
950  RETURN P
960  FNEND
961       !*****************************************************
970   SUB  Fnfp(Theta,Fr,Fi)
980  RAD
990    Fr=-SIN(Theta)/Theta
1000   Fi=-(COS(Theta)-1)/Theta
1010 SUBEND
1011      !*****************************************************
1020   DEF  FNB(Alp,Vrf,Thetan,Thetap,Gma)
1030    B=3*Alp*Vrf/((Thetan+Thetap)*Gma)
1040 RETURN B
1050 FNEND
1051      !*****************************************************
1060  DEF FNHob(B)
1070   P=(2/B)*FNI1(B)/FNI0(B)
1080 RETURN P
1090 FNEND
1091      !*****************************************************
1100  DEF  FNWa2(Vs,Alp,Idc,Er,E0,A)
1110   P=3*Vs*Alp*Idc/(Er*E0*A)
1120 RETURN P
1130
1131
!*****************************************************
1140 DEF FNW2(F)
1150  P=(2*PI*F)^2
1160 RETURN P
1170 FNEND
1171
!*****************************************************
1180  SUB Cmult(X1,Y1,X2,Y2,X3,Y3)
1190   X4=X1*X2-Y1*Y2
1200   Y4=X1*Y2+Y1*X2
1210  X3=X4
1220   Y3=Y4
1230 SUBEND
1231
!*****************************************************
1240 SUB Cdiv(X1,Y1,X2,Y2,X3,Y3)
1250   Denom=X2*X2+Y2*Y2
1260    X4=(X1*X2+Y1*Y2)/Denom
1270     Y4=(Y1*X2-X1*Y2)/Denom
1280  X3=X4
1290   Y3=Y4
1300 SUBEND
1301
!*****************************************************
2000 Plot:   SUB  Plot(Yra(*),Yia(*),Np,Indo)
2010 OPTION BASE 1
```

```
2011  COM /Plut/ Xmin,Xmax,Xstep,Ymin,Ymax,Ystep,Pltt,Xindo,Yindo
2020  ALLOCATE P(Np,3),P_title$[72],Mag(Np),Phase(Np)
2021  DIM Lb$(18)[2],V$[25],H$[25]
2022  DATA "33","34","35","36","37","38","39","40","41","42","43","44","45","46",
      "47","48","49","50"
2023  READ Lb$(*)
2030  DEG
2031  Indo=Indo+1
2032  IF Indo>1 THEN
2033  GRAPHICS ON
2034  GOTO Qp
2035  END IF
2040  INPUT "WANT PLOT ON SCREEN OR 7475A PLOTTER  (S OR P)",Ans$
2050  IF NOT (Ans$="S") AND NOT (Ans$="P") THEN 2040
2060  Pltt=0
2070  IF Ans$="P" THEN Pltt=1
2080  OUTPUT CRT;"Zimag MAX = ";MAX(Yia(*));"  Zimag MIN = ";MIN(Yia(*))
2180  INPUT "ENTER Zimag SCALE MIN (ohms)",Xmin
2181  INPUT "ENTER Zimag SCALE MAX (ohms)",Xmax
2200  INPUT "ENTER zIMAG SCALE STEP (ohms)",Xstep
2210  OUTPUT CRT;"Zreal MAX = ";MAX(Yra(*));"  Zreal MIN = ";MIN(Yra(*))
2230  INPUT "ENTER Zreal SCALE MIN (ohms)",Ymin
2240  INPUT "ENTER Zreal SCALE MAX  (ohms)",Ymax
2250  INPUT "ENTER Zreal SCALE STEP (ohms)",Ystep
2260  INPUT "ENTER PLOT TITLE ",P_title$
2280  C$=CHR$(255)&"K"
2290  OUTPUT 2 USING "#,K";C$        ! Clear screen
2370  OUTPUT 2 USING "#,K";C$        ! Clear screen for graph
2380  GINIT                          ! Initialize various graphics parameters.
2382  IF Pltt=0 THEN
2390  PLOTTER IS 3,"INTERNAL"        ! Use the internal screen
2391  ELSE
2392  PLOTTER IS 705,"HPGL"
2393  END IF
2400  GRAPHICS ON                    ! Turn on the graphics screen
2410  LORG 6                         ! Reference point: center of top of label
2411  Ans$="N"
2413  IF Pltt=0 THEN INPUT "WANT PRINTER PLOT TO MATCH PLOTTER SIZE
(Y/N)", Ans$
2414  K1=1
2415  K2=1
2416  IF Ans$="Y" THEN K1=.8121212121212
2417  IF Ans$="Y" THEN K2=.859200
2420  X_gdu_max=K2*100*MAX(1,RATIO)    ! Determine how many GDUs wide the
        screen is
2430  Y_gdu_max=K1*100*MAX(1,1/RATIO)  ! Determine how many GDUs high
        the screen is
2431  CSIZE 4.3
2432  LDIR -90
2433  PEN 1
```

```
2440  FOR I=-.1 TO .1 STEP .1         ! Offset of X from starting point
2450      MOVE  X_gdu_max-.03*(X_gdu_max),Y_gdu_max/2+I
2460    LABEL P_title$
2470  NEXT I
2490  LDIR -90
2500  CSIZE 3.5
2510  MOVE X_gdu_max*.03,Y_gdu_max/2              ! Move to center of left edge
        of screen
2511  H$="[RE] Impedance"
2520  LABEL H$
2530  LORG 4                         ! Reference point: center of bottom of label
2540  LDIR 0                      ! Horizontal labels again
2550  MOVE X_gdu_max/2,.93*Y_gdu_max! X: center of screen; Y: above key
        labels
2551  V$="[IM] Impedance"
2560  LABEL V$
2570  VIEWPORT .081*X_gdu_max,.94*X_gdu_max,.0031*Y_gdu_max,.85*
        Y_gdu_max
2580  WINDOW Xmin,Xmax,Ymax,Ymin
2590  AXES Xstep/2,Ystep/2,Xmin,Ymin,5,5,3            ! Draw axes intersecting at
        lower left
2591  AXES Xstep/2,Ystep/2,Xmax,Ymax,5,5,3            ! Draw axes intersecting at
        lower left
2610 !GRID Xstep,Ystep,Xmin,Ymin              ! Draw grid with no minor ticks
2611   GRID  Xstep/4,Ystep/4,Xmin,Ymin,1000,1000,0
2620  CLIP OFF                    ! So labels can be outside VIEWPORT limits
2630  CSIZE 2.5,.5                 ! Smaller chars for axis labelling
2640  LORG 6                    ! Ref. pt: Top center
2641  LDIR -90
2650  FOR I=Xmin+Xstep TO Xmax STEP Xstep         ! Every 10 units
2660      MOVE  I-.07*Xstep,Ymin-.04*(Ymax-Ymin)
2670    LABEL USING "#,K";I
2680  NEXT I
2690  LORG 8
2691  LDIR -90
2700  FOR I=Ymin TO Ymax STEP Ystep
2710      MOVE  Xmin-.015*(Xmax-Xmin),I+(Ymax-Ymin)*.02
2720    LABEL USING "#,K";I
2730  NEXT I
2740  PENUP                        ! LABEL statement leaves the pen down
2741  CSIZE 2
2743  Cvt=0
2750 Qp:  FOR I=1 TO Np
2760      PLOT Yia(I),Yra(I)
2763    IF Indo=1 OR Indo=4 OR Indo=7 THEN
2764    IF Indo=1 THEN LORG 8
2765    IF Indo=4 THEN LORG 5
2766    IF Indo=7 THEN LORG 2
2767    IF I MOD 1=0 THEN
2768    Cvt=Cvt+1
```

```
2769    LDIR -90
2770    LABEL Lb$(Cvt)
2771      PLOT Yia(I),Yra(I)
2772    END IF
2773    END IF
2774    IF I=Np AND Indo<=7 THEN
2778      PLOT Yia(I),Yra(I)
2779  CALL  Plotdoc(Xmin,Xmax,Xstep,Ymin,Ymax,Ystep,4,Xindo,Yindo,Pltt,Smt)
2780 PEN 1
2781  MOVE  Yia(I),Yra(I)
2782 END IF
2783  NEXT I                        ! et cetera
2784  PENUP
2790 Exit:  GRAPHICS OFF
2820  SUBEND                        ! finis
2821
!**********************************************************
2900    SUB  Ptor(Mag,Phase,X,Y)
2910    DEG
2920      X=Mag*COS(Phase)
2930       Y=Mag*SIN(Phase)
2940  SUBEND
2941
!**********************************************************
2950    SUB  Rtop(X,Y,Mag,Phase)
2960    DEG
2970     Mag=SQR(X*X+Y*Y)
2980     !DEFAULT ON       makes math errors non-fatal
2990      Phase=2*ATN(Y/(X+Mag))
3000   !DEFAULT OFF
3010  SUBEND
3011
!**********************************************************
3020     DEF  FNGnm(W2,Wr2,Ln,Lp,Thetap,Thetan)
3030  RAD
3040   Lt=Ln+Lp
3050        S1=(1-(Wr2/W2)*((Ln/Lt)*(1-SIN(Thetan)/Thetan)+(Lp/Lt)*(1-
            SIN(Thetap)/Thetap)))^2
3052        S2=((Wr2/W2)*((Ln/Lt)*((1-COS(Thetan))/Thetan)+(Lp/Lt)*((1-
            COS(Thetap))/Thetap)))^2
3062   S3=(S1+S2)^.5
3072  RETURN S3
3082  FNEND
3083
!**********************************************************
3092     SUB  Zpack(Zdr,Zri,Freq,Ls,Lp,Lm,Cp,Cm,Zpr,Zpi)
3102 OPTION BASE 1
3112     DIM  Bigr(4,6),Bigi(4,6),Ar(2,2),Ai(2,2)
3122     DIM  Tr1(2,2),Ti1(2,2),Tr2(2,2),Ti2(2,2),Sr(2,2),Si(2,2)
3142 FOR I=1 TO 6
```

```
3143   Bigr(1,I)=1
3152   Bigr(4,I)=1
3162 NEXT I
3172   W=2*PI*Freq
3182   Bigi(3,1)=(W*Cm)
3183   Bigi(2,2)=(W*Lm)
3184   Bigi(3,3)=(W*Cp*.65)
3185   Bigi(2,4)=(W*Lp)
3186   Bigi(3,5)=(W*Cp*.35)
3187   Bigi(2,6)=(W*Ls)
3197 FOR I=1 TO 6
3210     CALL  Canit(Bigr(*),Bigi(*),I,Tr1(*),Ti1(*))
3215 Ket:  IMAGE DD.D,3X,MD.DDD,3X,MD.DDD,3X,MD.DDD,3X,MD.DDD,3X,MD.DDD,
      3X,MD.DDD
3216 Ket2:  IMAGE DD.D,3X,MDDDD.DDD,3X,MDDDD.DDD,3X,MDDDD.DDD,3X,
      MDDDD.DDD
3217 IF I<>1 THEN
3227     CALL  Cmatmult(Ar(*),Ai(*),2,2,Tr1(*),Ti1(*),2,2,Tr2(*),Ti2(*),J,K)
3237  MAT Ar= Tr2
3247  MAT Ai= Ti2
3257  ELSE
3267  MAT Ar= Tr1
3277  MAT Ai= Ti1
3287  END IF
3297  NEXT I
3299 !           CALL  Tabcsts(Ar(*),Ai(*),Sr(*),Si(*))
3300   ! OUTPUT  1  USING  Ket;Freq/1.E+9;Sr(1,1);Si(1,1);Sr(2,1);Si(2,1);
        Sr(2,2);Si(2,2)
3307    CALL  Cmult(Ar(1,1),Ai(1,1),Zdr,Zdi,R1,I1)
3308    CALL  Cmult(Ar(2,1),Ai(2,1),Zdr,Zdi,R2,I2)
3318     CALL  Cdiv((R1+Ar(1,2)),(I1+Ai(1,2)),(R2+Ar(2,2)),(I2+Ai(2,2)),Zpr,Zpi)
3328 SUBEND
3329
!**********************************************************
3331  Canit:      SUB  Canit(Br(*),Bi(*),Whch,Lr(*),Li(*))
3332  OPTION BASE 1
3333     Lr(1,1)=Br(1,Whch)
3334     Li(1,1)=Bi(1,Whch)
3335     Lr(1,2)=Br(2,Whch)
3336     Li(1,2)=Bi(2,Whch)
3337     Lr(2,1)=Br(3,Whch)
3338     Li(2,1)=Bi(3,Whch)
3339     Lr(2,2)=Br(4,Whch)
3340     Li(2,2)=Bi(4,Whch)
3341  SUBEND
3342
!**********************************************************
3350  Cmatmult:         SUB  Cmatmult(Rm(*),Im(*),N,M,Rm2(*),Im2(*),O,P,Pr(*),
        Pi(*),Y,Z)
3351    OPTION BASE 1
```

```
3352   IF M=O THEN Ok
3353   BEEP
3354   PRINT "INCOMPATIBLE MATI FOR MULTIPLICATION"
3355   PAUSE
3356   SUBEXIT
3357 Ok:  Y=N
3358     Z=P
3359     FOR I=1 TO N
3360     FOR J=1 TO P
3361       Tempr=0
3362       Tempi=0
3363     FOR K=1 TO M
3364           CALL  Cmult(Rm(I,K),Im(I,K),Rm2(K,J),Im2(K,J),Tr,Ti)
3365        Tempr=Tempr+Tr
3366        Tempi=Tempi+Ti
3367     NEXT K
3368        Pr(I,J)=Tempr
3369        Pi(I,J)=Tempi
3370     NEXT J
3371     NEXT I
3372     SUBEND
3380  Tabcsts:  SUB  Tabcsts(Ar(*),Ai(*),Str(*),Sti(*))
3381  OPTION BASE 1
3382  Zi=50
3383  Zo=50
3384    Dr=Zo*Ar(1,1)+Ar(1,2)+Zo*Zi*Ar(2,1)+Zi*Ar(2,2)
3387    Di=Zo*Ai(1,1)+Ai(1,2)+Zo*Zi*Ai(2,1)+Zi*Ai(2,2)
3390    Rt=Zo*Ar(1,1)+Ar(1,2)-Zo*Zi*Ar(2,1)-Zi*Ar(2,2)
3393    It=Zo*Ai(1,1)+Ai(1,2)-Zo*Zi*Ai(2,1)-Zi*Ai(2,2)
3396   CALL  Cdiv(Rt,It,Dr,Di,Rt1,It1)
3401   Str(1,1)=Rt1
3402   Sti(1,1)=It1
3403   CALL  Cdiv(2*Zi,0,Dr,Di,Rt2,It2)
3404   Str(1,2)=Rt2
3405   Sti(1,2)=It2
3406   CALL  Cdiv(2*Zo,0,Dr,Di,Rt3,It3)
3407   Str(2,1)=Rt3
3408   Sti(2,1)=It3
3409    Rt4=Zo*Ar(1,1)-Ar(1,2)+Zo*Zi*Ar(2,1)-Zi*Ar(2,2)
3410    It4=Zo*Ai(1,1)-Ai(1,2)+Zo*Zi*Ai(2,1)-Zi*Ai(2,2)
3411   CALL  Cdiv(Rt4,It4,Dr,Di,Rt5,It5)
3412   Str(2,2)=-1*Rt5
3413   Sti(2,2)=-1*It5
3414  SUBEND
3415
!*****************************************************************
3417   SUB  Csparms(C,Freq)
3422   W=2*PI*Freq
3432  Zo=50
3433   Den=4+(Zo*W*C)^2
```

```
3442  S11r=-Zo^2*W^2*C^2/Den
3452  S11i=-2*Zo*W*C/Den
3462  S21r=4/Den
3472  S21i=-2*Zo*W*C/Den
3482  OUTPUT 701;"S11R,S11I=",S11r,S11i
3492  OUTPUT 701;"S21R,S21I=",S21r,S21i
3502 SUBEND
3503
!*****************************************************************
3510   SUB Plotdoc(Xmin,Xmax,Xstep,Ymin,Ymax,Ystep,Pnum,Xindo,Yindo,
        Pltter,Smt)
3520  DIM Lkl$[72]
3521  DIM Color$(15)[11]
3522   DATA "Black","White","Red","Yellow","Green","Cyan","Blue","Magenta",
        "Black","Olive Green","Aqua","Royal
Blue","Marroon","Brick  Red","Orange","Brown"
3523  READ Color$(*)
3530    Ypoints=(INT((Ymax-Ymin)/Ystep+1))*20+11
3531   Xpoints=(INT((Xmax-Xmin)/Xstep+1))*20+11
3541  ALLOCATE X(Xpoints),Y(Ypoints)
3551   Deltax=(Xmax-Xmin)/Xpoints
3552    Deltay=(Ymax-Ymin)/Ypoints
3562 FOR I=0 TO Xpoints STEP 1
3563   X(I)=Xmin+(I-6)*Deltax
3564 NEXT I
3572 FOR J=0 TO Ypoints STEP 1
3582  Y(J)=Ymin+(J-6)*Deltay
3592 NEXT J
3595 LORG 1
3597 GRAPHICS INPUT IS KBD,"KBD"
3598 Czz=2
3599 GOSUB Lblit
3601 CSIZE Czz
3602 SYSTEM PRIORITY 12
3603 Noth:!
3612 ON KEY 7 LABEL "",13 GOTO Noth
3622 IF NOT (Pltter) THEN ON KEY 8 LABEL "ERASE",13 GOTO Erase
3623 IF (Pltter) THEN ON KEY 8 LABEL "POS-CHECK ",13 GOTO Poscheck
3632 ON KEY 0 LABEL "MOVE Left ",13 GOTO Left
3633 ON KEY 1 LABEL "MOVE Right",13 GOTO Right
3634 ON KEY 2 LABEL "MOVE Up ",13 GOTO Up
3635 ON KEY 3 LABEL "MOVE Down",13 GOTO Down
3636 ON KEY 4 LABEL "DONE",13 GOTO Don
3646 ON KEY 9 LABEL "LABEL",13 GOTO Lbl
3656 ON KEY 5 LABEL "CSIZE",13 GOTO Cz
3657 ON KEY 6 LABEL "PEN#&COLOR",13 GOTO Pn
3658 ON KEY 10,13 GOTO Fleft
3659 ON KEY 11,13 GOTO Fright
3660 ON KEY 12,13 GOTO Fup
3661 ON KEY 13,13 GOTO Fdown
```

```
3667 Wt: GOTO Wt
3668  Fright:Fast=15
3670 Right:     ON KEY 1 LABEL "RIGHT",12 GOTO Right
3677 IF NOT (Pltter) THEN GOSUB Eraselbl
3687         Xindo=ABS((Xindo+1*Fast) MOD Xpoints+1)
3697  GOSUB Lblit
3698       Fast=1
3707      GOTO Noth
3708 Poscheck:        OFF KEY
3709                PEN Down
3710                WAIT .75
3711                PEN Up
3712                GOTO Noth
3714  Fleft:  Fast=15
3717 Left:  ON KEY 0 LABEL "LEFT",12 GOTO Left
3727  IF NOT (Pltter) THEN GOSUB Eraselbl
3737          Xindo=ABS((Xindo-1*Fast) MOD (Xpoints+1))
3738 GOSUB Lblit
3747       GOSUB Lblit
3748       Fast=1
3757      GOTO Noth
3758  Fup:  Fast=15
3767 Up:  ON KEY 2 LABEL "UP",12 GOTO Up
3777 IF NOT (Pltter) THEN GOSUB Eraselbl
3778          Yindo=ABS((Yindo+1*Fast) MOD Ypoints+1)
3788 GOSUB Lblit
3789       Fast=1
3798      GOTO Noth
3799  Fdown:Fast=15
3808 Down:      ON KEY 3 LABEL "DOWN",12 GOTO Down
3818    IF NOT (Pltter) THEN GOSUB Eraselbl
3821          Yindo=ABS((Yindo-2*Fast) MOD Ypoints+1)
3826  GOSUB Lblit
3827       Fast=1
3830      GOTO Noth
3840  Eraselbl:   Pstore=Pnum
3841          IF NOT (Smt) THEN
3850          PEN -1
3851         ELSE
3852           PEN -Pnum
3853         END IF
3860           MOVE X(Xindo),Y(Yindo)
3870  IF NOT (Pltter) THEN LABEL USING "A,#";"+"
3871           PEN Pstore
3880         RETURN
3890  Lblit:  MOVE X(Xindo),Y(Yindo)
3900  IF NOT (Pltter) THEN LABEL USING "#,K,#";"+"
3910       RETURN
3920 Lbl: OFF KEY
3930     INPUT "ENTER LABEL FOR CURRENT POS & >(72 MAX)",Lkl$
```

```
3940      LABEL USING "#,K,#";Lkl$
3950      Lkl$=""
3960     GOTO Noth
3970 Erase:  OFF KEY
3971          MOVE X(Xindo),Y(Yindo)
3973           Pstore=Pnum
3974        PEN -1
3980        INPUT "ENTER LABELTO BE ERASED",Lkl$
3990          LABEL USING "#,K,#";Lkl$
4000         Lkl$=""
4010          PEN Pstore
4020        GOTO Noth
4030 Pn:      OFF KEY
4040          OUTPUT CRT;" THE CURRENT PEN NUMBER IS "&VAL$(Pnum)
4050           Preal=Pnum MOD 15
4060          OUTPUT CRT;" THE CURRENT PEN COLOR IS "&Color$(Preal)
4061           Ans$="N"
4070          INPUT "WANT TO CHANGE PEN COLOR (Y/N)",Ans$
4080         IF Ans$<>"Y" THEN GOTO Noth
4090           INPUT "ENTER PEN NUMBER",Pnum
4100        OUTPUT CRT;"THE NEW PEN NUMBER AND COLOR ARE "
4101            OUTPUT CRT;Pnum;"     ";Color$(Pnum)
4102           PEN Pnum
4111         GOTO Noth
4121 Cz:         OFF KEY
4131            OUTPUT CRT;"THE CURRENT CSIZE IS "&VAL$(Czz)
4141           INPUT "ENTER NEW CSIZE OR CONT FOR OLD",Czz
4151            CSIZE Czz
4161            GOTO Noth
4162 Don:  GOSUB Eraselbl
4171     SUBEND
```

Appendix I

TOUCHSTONE CIRCUIT AND S-PARAMETER FILES

This Appendix contains the circuit and data files required by Touchstone, a commercially available microwave computer-aided design package. This program is used to optimize design dimensions of the impedance transformers used in the Kurokawa cavity oscillators. A circuit diagram of the oscillator circuit appears in Figure I-1. The Touchstone scattering- (S) parameter data file for a 0.222-inch-length cavity, 0.1285-inch-diameter iris, and 0.048-inch sidewall separation is presented in Table I-1. The port reference impedances for these data are Zwg at port 1 and 50 ohms at port 2. Table I-2 provides the same information, except the data have been transformed to a reference impedance of 50 ohms at both ports.

```
! THIS CIRCUIT FILE DESCRIBES TRANSFORMER #17 DESIGN   6/18/86

DIM
      FREQ    GHZ
      RES     OH
      IND     NH
      CAP     PF
      LNG     IN
      TIME    PS
      COND    /OH
      ANG     DEG

VAR
      Ep0 = 8.854E-12 !DIELECTRIC CONSTANT OF FREE SPACE
      A1 = .280   !WIDTH OF KA-BAND WAVEGUIDE
      B1 = .140   !HEIGHT OF KA-BAND WAVEGUIDE
      LEN1 = .115  !DISTANCE FROM CENTER OF WAVEGUIDE TO REF SHORT
            POSITION
      Z01 = 50.   !CHARACTERISTIC IMPEDANCE OF COAXIAL LINE
      LEN2 =.0404  !LENGTH OF 50 OHM SPACER
      Z02 =50     !CHARACTERISTIC IMPEDANCE OF SPACER
      LEN3 = .0380 !LENGTH OF TRANSFORMER
      Z03 = 23.0295!CHARACTERISTIC IMPEDANCE OF TRANSFORMER
      Krex = 2.5275 !DIELECTRIC CONSTANT OF REXOLITE
      Er1 = 1     !DIELECTRIC CONSTANT OF SECTION 1
      SQRTEr1 =1    !SQUARE ROOT OF Er1
```

```
     Er2 = 1    !RELATIVE DIELECTRIC CONSTANT OF SPACER
     SQRTEr2 =1    !SQUARE ROOT OF Er2
     Er3 = 2.53  !RELATIVE DIELECTRIC CONSTANT OF TRANSFORMER
     SQRTEr3 =1.5898!SQUARE ROOT OF Er3
     Rinner3 = .017!RADIUS OF INNER CONDUCTOR

EQN
!**************************************************************
! COMPUTATION OF DISCONTINUITY CAPACITANCE DUE TO CHANGE IN OUTER
CONDUCTOR
! REFERENCE: MTT JAN 1967, P. I. SOMLO
!**************************************************************

     !SUBSCRIPT o DENOTES OUTER
     !SUBSCRIPT i DENOTES INNER
     !SUBSCRIPT b DENOTES BIGGER RADIUS
     !SUBSCRIPT s DENOTES SMALLER RADIUS
     !NUMBER SUBSCRIPT DENOTES THE INDEX OF XMISSN LINE SECTION

!#####DISCONTINUITY OF OUTER CONDUCTOR BETWEEN SECTIONS 1 AND 2 ####
     Tob2 = EXP(2*PI*Z02*SQRTEr2/376.7)
     Tos1 = EXP(2*PI*Z01*SQRTEr1/376.7)
     Ao12 = (Tos1-1)/(Tob2-1)
     QA12a = (Ao12**2+1)/Ao12*LN((1+Ao12)/(1-Ao12))
     QA12b = -2*LN(4*Ao12/(1-Ao12**2))
     QB12 = 4.12E-15*(0.8-Ao12)*(Tob2-1.4)
     C12    = Er2*2*PI*Rinner3*2.54E12*(Ep0/(100*PI)*(QA12a+QA12b)+QB12)

!#####DISCONTINUITY OF OUTER CONDUCTOR BETWEEN SECTIONS 2 AND 3 ####
     !Tob2 = EXP(2*PI*Z02*SQRTEr2/376.7)        duplicate variable
     Tos3 = EXP(2*PI*Z03*SQRTEr3/376.7)
     Ao23 = (Tos3-1)/(Tob2-1)
     QA23a = (Ao23**2+1)/Ao23*LN((1+Ao23)/(1-Ao23))
     QA23b = -2*LN(4*Ao23/(1-Ao23**2))
     QB23 = 4.12E-15*(0.8-Ao23)*(Tob2-1.4)
     C23    = Er2*2*PI*Rinner3*2.54E12*(Ep0/(100*PI)*(QA23a+QA23b)+QB23)


!**************************************************************
CKT
     RWGT      9 A^A1 B^B1 ER=1 RHO=0
     DEF1P     9 WGT1 !DEFINE KA-BAND WAVEGUIDE TERMINATION
     S2PA  1 2 0 A:MOD2A1T.S2P !DATAFILE CONTAINING S-PARAMETERS OF
          CAVITY
     DEF2P   1 2 CAVITY
     CAVITY   1 2
     TLINP   2 3 Z^Z01 L^LEN1 K=1 A=0 F=0
     CAP     3 0 C^C12
     TLINP   3 4 Z^Z02 L^LEN2 K=1 A=0 F=0
     CAP     4 0 C^C23
```

```
    TLINP   4 5 Z ^ Z03 L ^ LEN3 K ^ Krex A=0 F=0
    DEF2P   1 5 KUROSC


TERM
    KUROSC WGT1  0 0 !TERMINATES PORT 1 IN KA-BAND WAVEGUIDE
         TERMINATION

!PROC

OUT
    KUROSC RE[Z2] GR1
    KUROSC IM[Z2] GR1A

FREQ
    SWEEP 33 37 .1


GRID
    RANGE 33 37 .1
    GR1 0 10 1
    GR1A -5 -25 5

OPT                    !OPTIMIZER SETUP
    RANGE 34.5 34.51           !NO USED FOR ANALYSIS
    KUROSC RE[Z2] = 2.0 40
    RANGE 34.5  34.51
    KUROSC IM[Z2] = -16 40
    RANGE 33.5  35.5
    KUROSC RE[Z2] = 2.0 1
!TOL
```

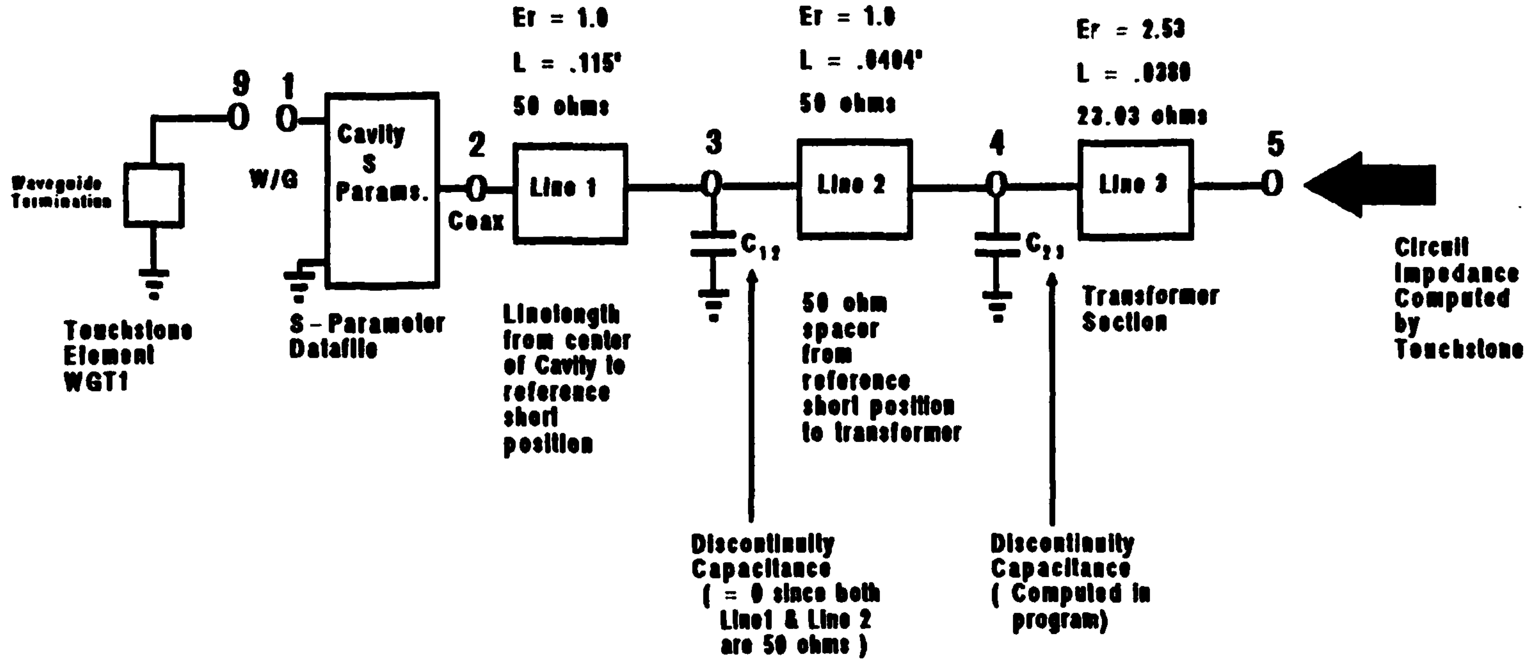


FIGURE I-1. Touchstone Circuit Diagram for Waveguide Kurokawa Oscillator Design.

TABLE I-1. S-Parameter Data File of Kurokawa Waveguide Oscillator. (Port 1 Ref Zwg, Port 2 Ref 50 Ohms.)

S-parameter output units real/imaginary (natural). Single-diode module #2; 13:29:24 19 May 1986; rotated data.

Freq(MHz)	S11_Rl	S11_Im	S21_Rl	S21_Im	S12_Rl	S12_Im	S22_Rl	S22_Im
33000	-.532	.614	.337	.184	.337	.184	.312	.570
33100	-.523	.626	.366	.187	.366	.187	.329	.561
33200	-.473	.613	-.406	-.150	-.406	-.150	.297	.586
33300	-.457	.610	-.412	-.163	-.412	-.163	.368	.549
33400	-.439	.597	-.440	-.152	-.440	-.152	.371	.537
33500	-.411	.610	-.468	-.142	-.468	-.142	.411	.522
33600	-.370	.575	-.486	-.115	-.486	-.115	.434	.509
33700	-.357	.555	-.510	-.111	-.510	-.111	.471	.482
33800	-.331	.546	-.549	-.081	-.549	-.081	.486	.453
33900	-.296	.512	-.572	-.055	-.572	-.055	.528	.429
34000	-.277	.469	-.586	-.018	-.586	-.018	.545	.393
34100	-.264	.444	-.617	.024	-.617	.024	.549	.342
34200	-.233	.385	-.631	.084	-.631	.084	.559	.301
34300	-.243	.332	-.632	.130	-.632	.130	.571	.253
34400	-.227	.303	-.641	.170	-.641	.170	.570	.194
34500	-.229	.232	-.637	.232	-.637	.232	.551	.147
34600	-.254	.192	-.614	.284	-.614	.284	.556	.106
34700	-.269	.143	-.597	.323	-.597	.323	.527	.047
34800	-.304	.089	-.570	.379	-.570	.379	.488	.005
34900	-.345	.057	-.519	.423	-.519	.423	.460	-.037
35000	-.385	.040	-.481	.445	-.481	.445	.423	-.083
35100	-.420	.002	-.447	.471	-.447	.471	.372	-.101
35200	-.475	-.009	-.395	.502	-.395	.502	.335	-.122
35300	-.510	-.005	-.346	.497	-.346	.497	.296	-.142
35400	-.544	-.025	-.319	.505	-.319	.505	.249	-.147
35500	-.601	-.017	-.269	.516	-.269	.516	.217	-.152
35600	-.625	0.000	-.230	.503	-.230	.503	.186	-.159
35700	-.655	.007	-.204	.499	-.204	.499	.153	-.144
35800	-.703	.012	-.172	.504	-.172	.504	.143	-.139
35900	-.714	.037	-.137	.481	-.137	.481	.106	-.138
36000	-.728	.051	-.118	.466	-.118	.466	.086	-.126
36100	-.770	.058	-.103	.463	-.103	.463	.068	-.109
36200	-.774	.089	-.071	.443	-.071	.443	.053	-.098
36300	-.784	.104	-.060	.426	-.060	.426	.024	-.084
36400	-.808	.109	-.050	.418	-.050	.418	.023	-.056
36500	-.822	.141	-.024	.405	-.024	.405	-.030	-.029
36600	-.820	.157	-.015	.388	-.015	.388	-.025	-.022
36700	-.836	.151	-.007	.377	-.007	.377	-.023	-.077
36800	-.855	.189	.008	.370	.008	.370	-.034	-.002
36900	-.835	.206	.015	.349	.015	.349	-.044	.007
37000	-.852	.204	.016	.344	.016	.344	-.033	.005

TABLE I-2. S-Parameter Data File of Kurokawa Waveguide Oscillator (Port 1 Ref Zwg, Port 2 Ref 50 Ohms).

S-parameter output units real/imaginary (natural). Single-diode module #2; 13:31:13 19 May 1986; rotated data

Freq(MHz)	S11_R1	S11_Im	S21_R1	S21_Im	S12_R1	S12_Im	S22_R1	S22_Im
33000	.819	.360	.286	-.065	.286	-.065	.160	.527
33100	.826	.356	.300	-.078	.300	-.078	.155	.523
33200	.823	.329	-.300	.102	-.300	.102	.105	.581
33300	.823	.320	-.308	.092	-.308	.092	.169	.532
33400	.818	.310	-.319	.104	-.319	.104	.152	.533
33500	.828	.299	-.322	.116	-.322	.116	.177	.534
33600	.818	.275	-.324	.121	-.324	.121	.195	.539
33700	.811	.266	-.339	.125	-.339	.125	.211	.518
33800	.813	.253	-.348	.147	-.348	.147	.203	.526
33900	.807	.232	-.356	.153	-.356	.153	.233	.524
34000	.796	.214	-.361	.164	-.361	.164	.247	.518
34100	.790	.203	-.372	.188	-.372	.188	.236	.516
34200	.780	.174	-.375	.205	-.375	.205	.254	.525
34300	.761	.158	-.386	.224	-.386	.224	.273	.519
34400	.758	.143	-.388	.239	-.388	.239	.282	.498
34500	.741	.114	-.398	.259	-.398	.259	.283	.503
34600	.720	.101	-.397	.287	-.397	.287	.332	.506
34700	.703	.079	-.403	.302	-.403	.302	.329	.475
34800	.676	.054	-.412	.335	-.412	.335	.340	.483
34900	.647	.037	-.399	.370	-.399	.370	.390	.460
35000	.617	.029	-.392	.398	-.392	.398	.407	.420
35100	.589	.002	-.397	.421	-.397	.421	.401	.415
35200	.542	-.008	-.383	.475	-.383	.475	.454	.400
35300	.507	-.005	-.350	.496	-.350	.496	.469	.332
35400	.472	-.027	-.354	.518	-.354	.518	.453	.325
35500	.402	-.022	-.323	.583	-.323	.583	.511	.291
35600	.368	0.000	-.274	.599	-.274	.599	.512	.217
35700	.323	.012	-.248	.628	-.248	.628	.512	.200
35800	.240	.023	-.218	.693	-.218	.693	.568	.173
35900	.224	.071	-.143	.680	-.143	.680	.525	.083
36000	.200	.104	-.102	.677	-.102	.677	.500	.048
36100	.112	.140	-.070	.731	-.070	.731	.520	.037
36200	.124	.213	.020	.694	.020	.694	.473	-.042
36300	.116	.256	.057	.673	.057	.673	.419	-.062
36400	.067	.294	.088	.687	.088	.687	.418	-.059
36500	.080	.390	.176	.652	.176	.652	.334	-.103
36600	.111	.421	.198	.605	.198	.605	.298	-.114
36700	.065	.437	.212	.606	.212	.606	.289	-.179
36800	.108	.553	.289	.563	.289	.563	.245	-.152
36900	.176	.538	.276	.493	.276	.493	.183	-.133
37000	.145	.573	.291	.498	.291	.498	.192	-.141

Appendix J

OSCILLATOR CIRCUIT UNIFORMITY TEST RESULTS

This Appendix contains output power versus frequency plots for each of the four oscillators used in the power-combining experiment (Figures J-1 through J-4). Each oscillator is tested under free-running (no injection-locking) input signals of 0.045 watts, 0.12 watts, 0.23 watts, and 0.45 watts average power.

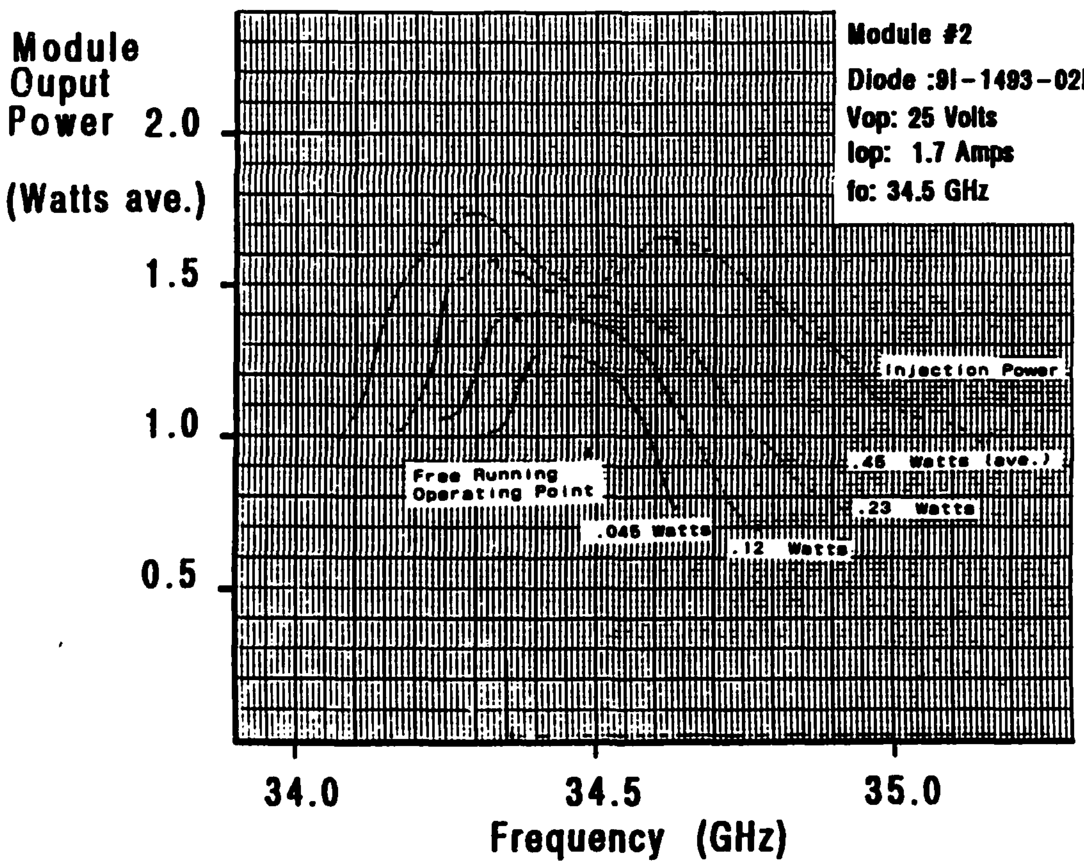


FIGURE J-1. Injection-Locked Power Characteristics, Module #2.

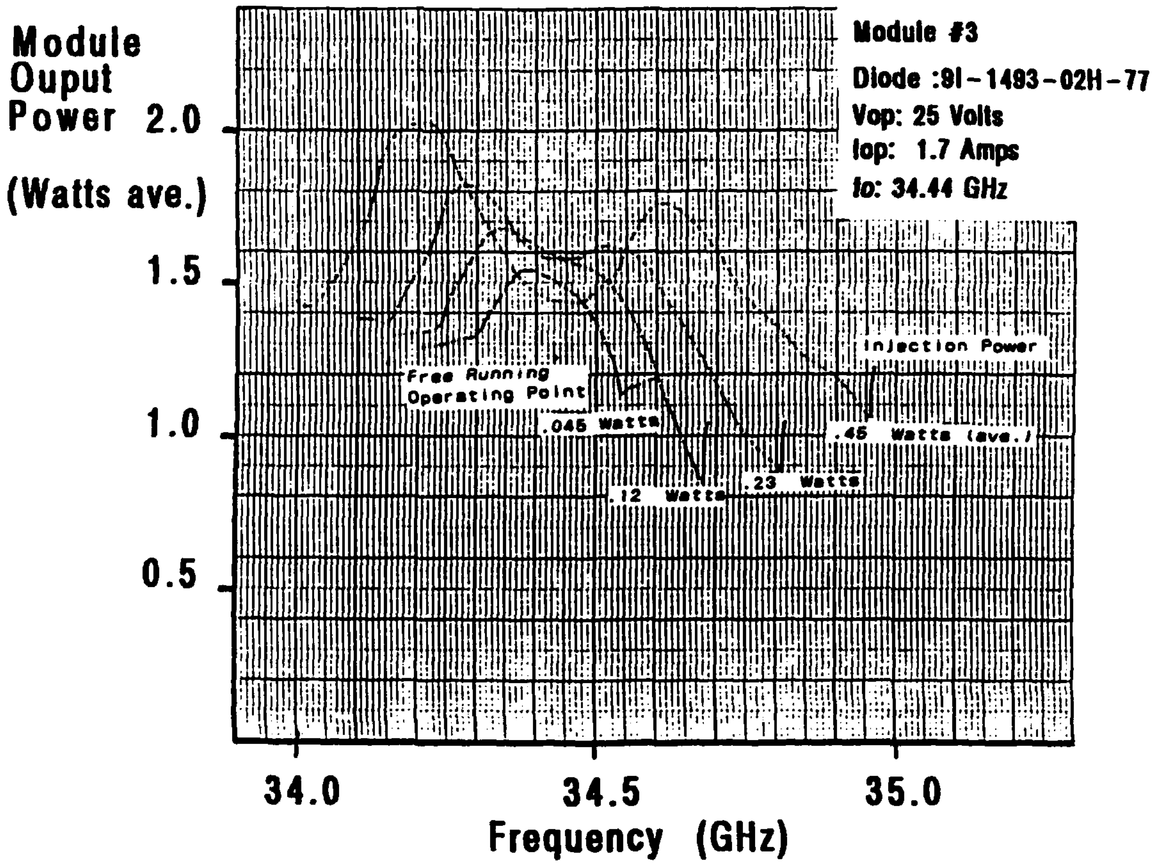


FIGURE J-2. Injection-Locked Power Characteristics, Module #3.

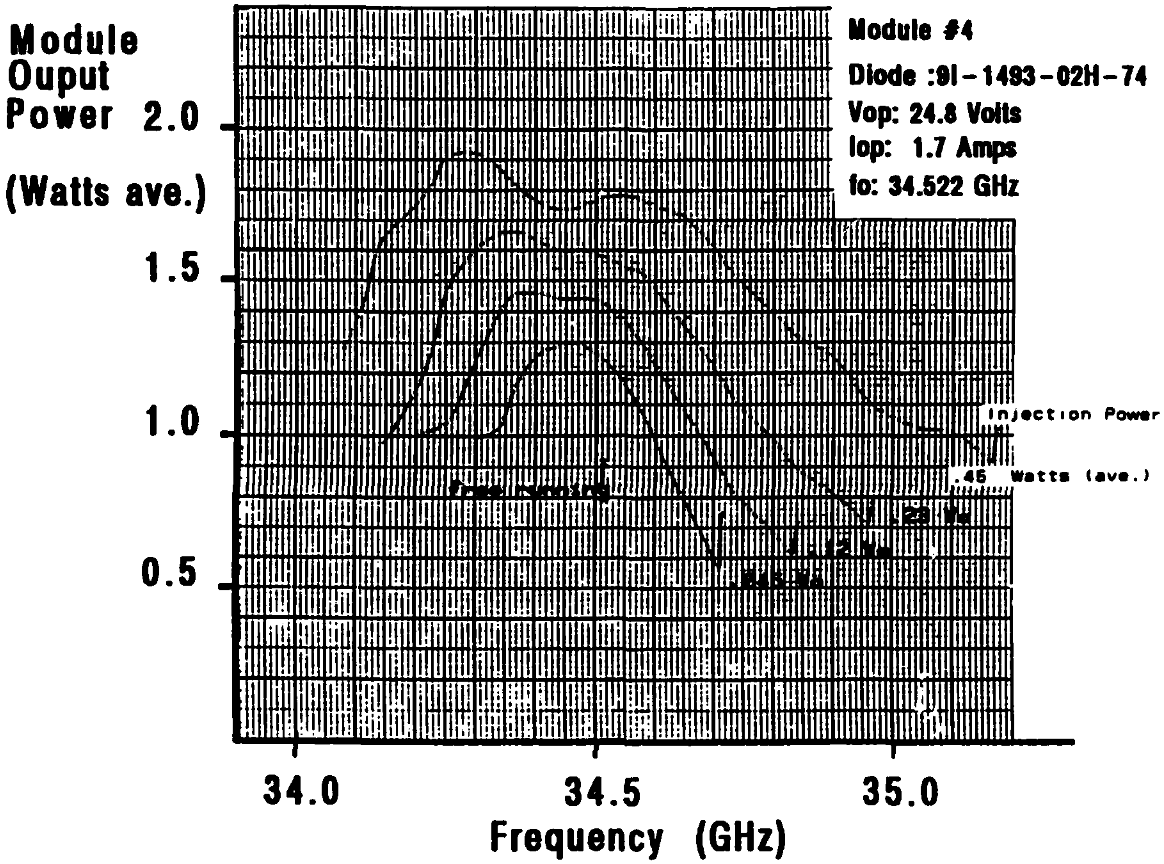


FIGURE J-3. Injection-Locked Power Characteristics, Module #4.

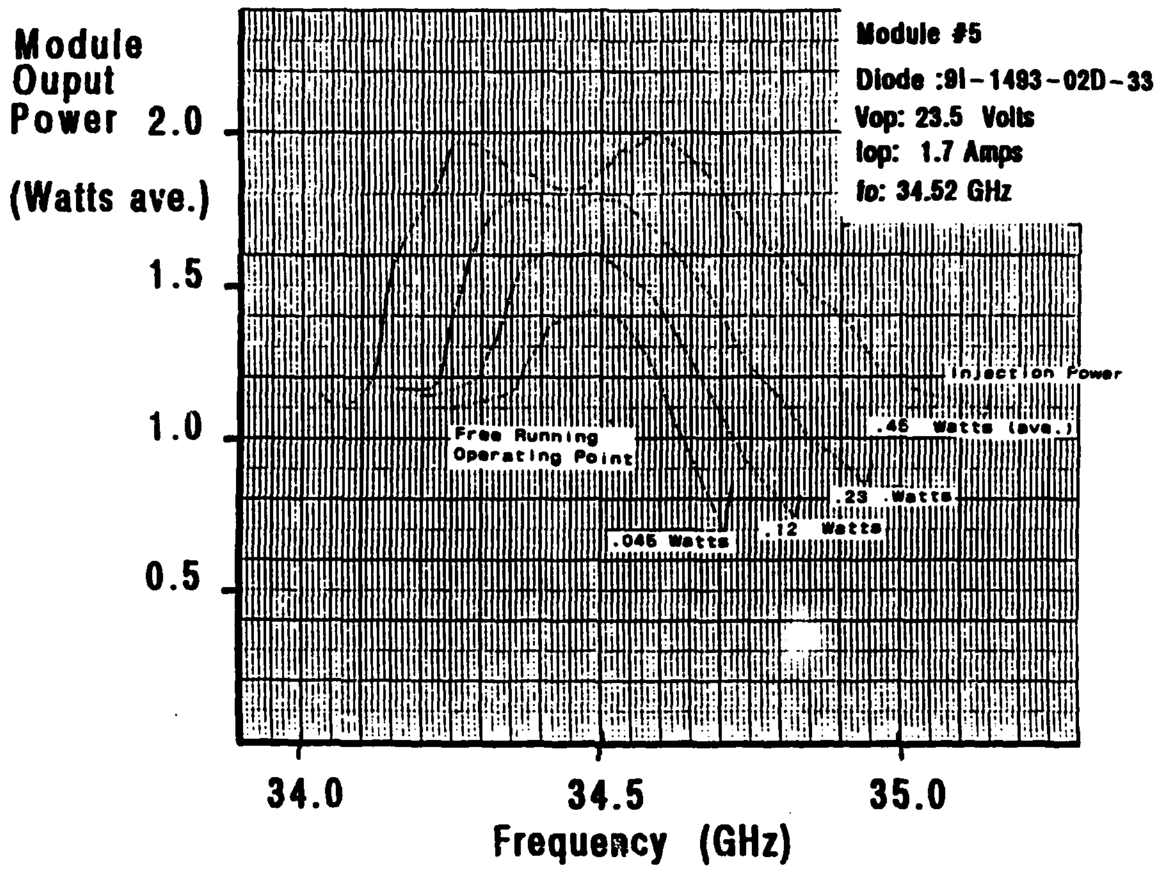


FIGURE J-4. Injection-Locked Power Characteristics, Module #5.

REFERENCES

1. R. L. Eisenhart, N. W. Nevils, J. J. Gulick, and R. C. Monzello. "A Matched Turnstile Type 4-Way Divider/Combiner," *1983 IEEE MTT-S Digest*, pp. 166-68.

2. Edward L. Ginzton. *Microwave Measurements*. New York, McGraw-Hill, 1957. Pp. 296-97.

3. C. G. Montgomery, R. H. Dicke, and E. M. Purcell. *Principles of Microwave Circuits*, Radiation Lab Series, Vol 8. New York, McGraw-Hill, 1948. P. 467.

4. R. L. Eisenhart and R. C. Monzello. "A Better Waveguide Short Circuit," *1982 IEEE MTT-S Digest*, pp. 360-62.

5. M. Adlerstein, L. Holway, and S. Chu. "Measurement of Series Resistance in IMPATT Diodes," *IEEE Trans. Electron Devices*, Vol. ED-3, No. 2 (February 1983), pp. 179-82.

6. Raytheon Company. *Development of Ka-Band Pulsed IMPATT Diodes*. Raytheon, Research Division, Lexington, Mass., November 1, 1983 - October 31, 1984. (Final Report, Navy Contract No. N60530-83-C-0213.)

INITIAL DISTRIBUTION

2 Naval Air Systems Command (AIR-5004)
2 Naval Sea Systems Command (Technical Library)
1 Commander in Chief, U. S. Pacific Fleet, Pearl Harbor (Code 325)
1 Commander, Third Fleet, San Francisco
1 Commander, Seventh Fleet, San Francisco
2 Naval Academy, Annapolis (Director of Research)
2 Naval Research Laboratory (Code 6852)
  N. Papanicolaou (1)
  K. J. Sleger (1)
1 Naval War College, Newport
4 Office of Naval Research, Arlington, VA
  OCNR-213, D. Siegel (1)
  OCNR-214, J. Hall (1)
  OCNR-22, J. Kauffman (1)
  OCNR-224, I. Mack (1)
1 Air Force Intelligence Agency, Bolling Air Force Base (AFIA/INTAW, Maj. R. Esaw)
2 Defense Technical Information Center, Alexandria
1 Hudson Institute, Incorporated, Center for Naval Analyses, Alexandria, VA (Technical Library)
1 Microwave Associates, Incorporated, Burlington, MA (M. Hines)
3 Raytheon Company, Lexington, MA
  M. Adlerstein (1)
  J. Fines (1)
  J. McClymonds (1)
3 University of Michigan, Department of Electrical Engineering and Computer Science, Ann Arbor, MI
  G. Haddad (1)
  C. Kidner (1)
  R. Mains (1)
1 Varian Associates, Beverly Microwave Division, Beverly, MA (W. S. Best)