NIST Technical Note 1281

The NIST Digitally Synthesized Power Calibration Source

N. M. Oldham
O. B. Laug
B. C. Waltrip
R. H. Palm

Electronic Instrumentation and Metrology Group
Electricity Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

August 1990

U.S. Department of Commerce
Robert A. Mosbacher, Secretary

National Institute of Standards and Technology
John W. Lyons, Director

List of Figures

List of Tables

THE NIST DIGITALLY SYNTHESIZED POWER CALIBRATION SOURCE

N. M. Oldham, O. B. Laug, B. C. Waltrip
and R. H. Palm
National Institute of Standards and Technology
Electricity Division
Electronic Instrumentation and Metrology Group
Gaithersburg, MD 20899

A digitally synthesized source of "phantom" power for calibrating electrical power and energy meters is described. Independent sources of voltage, current, and phase angle are programmable between 0-240 volts, 0-5 amperes, and 0-360 degrees, respectively. The uncertainty of the active and reactive power is estimated to be within ±100 ppm of the full scale apparent power (volt-amperes).

Key Words: calibration, digital waveform synthesis, power measurement, TDM wattmeter, transconductance amplifier, voltage amplifier, wattmeter, watt-hour meter.

1. INTRODUCTION

Calibration of wattmeters and watthour meters has traditionally been made by comparing the meter under test (MUT) to a standard wattmeter or watthour meter. The advantage of that approach is that a precise knowledge of the source parameters is not required. Voltage and current amplitudes and the phase angle between them need only be known approximately and the stability of each of these parameters is not critical as long as the power or energy output of the MUT is averaged or integrated over the identical period as that of the standard instrument. With the advent of multifunction instruments capable of measuring voltage, current, power factor, and active and reactive power, a knowledge of each of the source parameters has become advantageous. The measurement of reactive power and energy, in particular, is greatly simplified if the source of voltage, current, and phase angle is known and stable.

This approach has led to the development of a dual-channel sinewave source of voltage and current, which is shown in figure 1. Previous experience with digital waveform synthesis methods [1-6] provided the basis for designing a precision digital waveform generator to synthesize two low-level sinusoidal signals, which are programmable in both amplitude and phase angle. An illustration of a 20-step digitally synthesized signal, and the sine wave from which the sampled points were derived, is given in figure 2. A special voltage amplifier A1 [7] was designed to scale the low-level voltage V_1 to typical test levels ranging from 60-240 V, while test currents ranging from 1-5 A are obtained with a specially designed transconductance amplifier A2 [8]. The source is controlled by a desk-top computer that is linked to auxiliary instrumentation for measuring the analog and digital outputs of the MUT.[1]

---

[1] *In order to describe adequately the systems and tests discussed in this report, commercial equipment and instruments are identified by manufacturer's name and/or model number. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the material or equipment identified is necessarily the best available for the purpose.*

The controller for the power calibration source is a Hewlett Packard 9836 desk-top computer equipped with one megabyte of memory, a general purpose input/output (GPIO) interface card, and a DMA card. Data is transmitted and received at the generator rear panel through the 50-line GPIO cable (see HP 98622A GPIO Interface Installation manual for pin connections).

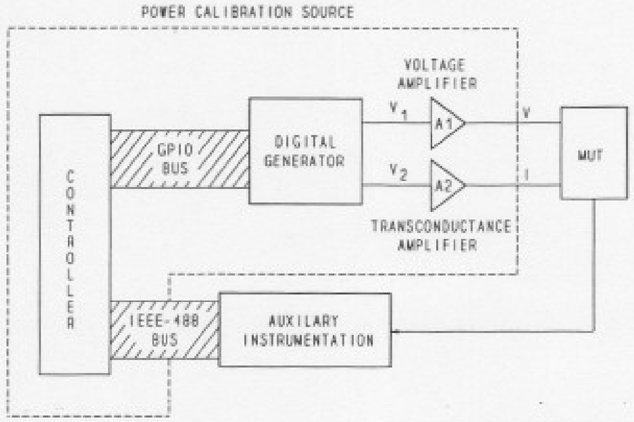

Figure 1. Block diagram of the power calibration source.

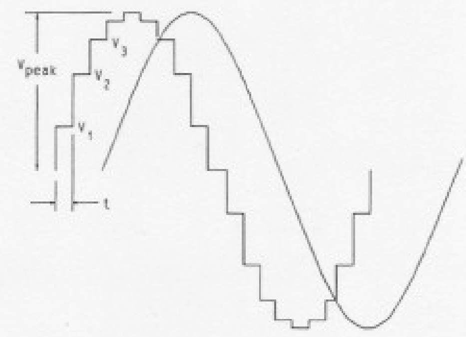

Figure 2. A digitally synthesized waveform.

2. HARDWARE

2.1 Digital Generator

The heart of the power calibration source is a digital generator (see fig. 3), which synthesizes voltages V_1 and V_2 in a staircase or "zero-order-hold" approximation to a sinusoidal waveform. Two signals whose waveforms are independently programmable are constructed by sequentially applying digital values stored in random access memories, RAM1 and RAM2 to 18-bit multiplying digital-to-analog converters, MDAC1 and MDAC2. The values stored in memory are equally spaced samples of this waveform with up to 2048 discrete steps per period. For normal operation the stored functions are sinewaves; however, any arbitrary waveforms with up to 1024 harmonic components may be stored. The phase angle between the two waveforms is determined by changing the set of function values stored in RAM2.

The theoretical resolution of the phase angle separating a pair of digitally synthesized sinewaves is a function of the resolution of the processor used to calculate the sample points, the resolution of the generating DACs (this determines to what extent each step is quantized), and the number of steps (sample points) per period. The algorithm used to calculate the sample points is performed with adequate precision in the computer to introduce negligible errors. The generating MDACs, capable of 18-bit precision, are normally used as 16-bit converters to speed up data transmission. These MDACs may be updated at 1 μs intervals, and at 60 Hz, 2048 steps are used to synthesize one period. The angular resolution under these conditions (based on computer simulations) is approximately one microradian [9].

The amplitudes of V_1 and V_2 may be set independently between 0-10 volts-peak by controlling the dc reference voltages supplied to MDAC1 and MDAC2 with a second pair of 18-bit DACs. This technique provides an amplitude resolution of approximately 38 μV ($10/2^{18}$ volts). The offsets of all four DACs may be adjusted remotely over a range of $\pm$500 ppm by employing four additional 8-bit DACs. This technique provides a means for trimming the dc offset and gain of each of the generated waveforms that is adjustable under software control.

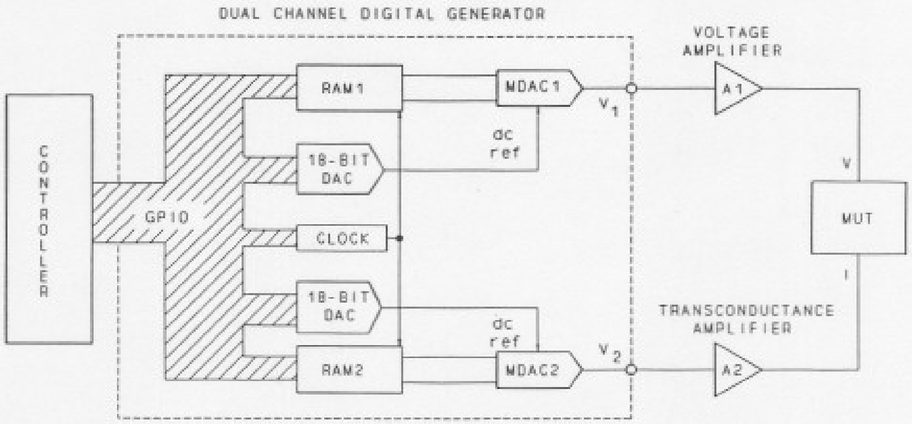

Figure 3. Block diagram of the dual-channel digital generator.

The frequency of the generated waveforms is a function of the number of steps per period and the sample rate (for a 60 Hz sine wave with 2048 steps the sample rate is 122.88 kHz). A programmable frequency synthesizer is used to generate the clock signal, which provides the strobe pulse for each sample point. It has a short-term stability of approximately 1 part in 10^8.

A detailed description of the digital generator hardware is given below. Note that figure 3 is a simplified diagram that does not include some of the hardware discussed below.

2.1.1 Optical Isolator

Data from the HP GPIO passes through the Optical Isolator Board (fig. 4) to separate the generator ground from the computer ground. Since the generator is a "listener," only the GPIO output lines and the handshake signal lines PCTL, PFLG, and CTL0 are optically isolated. The HP 2630 (U1-U8, U12) optical isolators invert the data, which is restored to the correct polarity with inverting buffers (U9-U11). The PFLG, which originates at the generator (and is the only outgoing line), is isolated separately in U13. The 5-volt power supply used to terminate the GPIO lines is a Datel dc-dc converter mounted on the board.

2.1.2 Interface

Once isolated, the data is decoded in the Interface Board (fig. 5). The GPIO data is separated into a "data" bus and a "control" bus which steers the data to the appropriate board. Whenever the GPIO data is changed, the PCTL line is pulled high (about 1 μs after the data is valid). Data is latched into U1 and U2 by this PCTL line when $\overline{CTL0}$ is high. When $\overline{CTL0}$ goes low, the output buffers U6-U9 are enabled, placing the control word on edge connector pins 3-18.

The PFLG, which is normally low, is pulsed high for 1 μs (time set by R1 and one-shot U4), about 1μs (set by R2 and C1) after the positive edge of PCTL. This logic provides a timed handshake with the GPIO interface, holding the data on the data bus (edge connector pins 38-53) for 1 μs.

2.1.3 Memories

The sample point values of the generated waveforms are stored in random access memory (RAM) located on the Memory Board (fig. 6). GPIO data is transferred from edge connector pins 21-28 (most

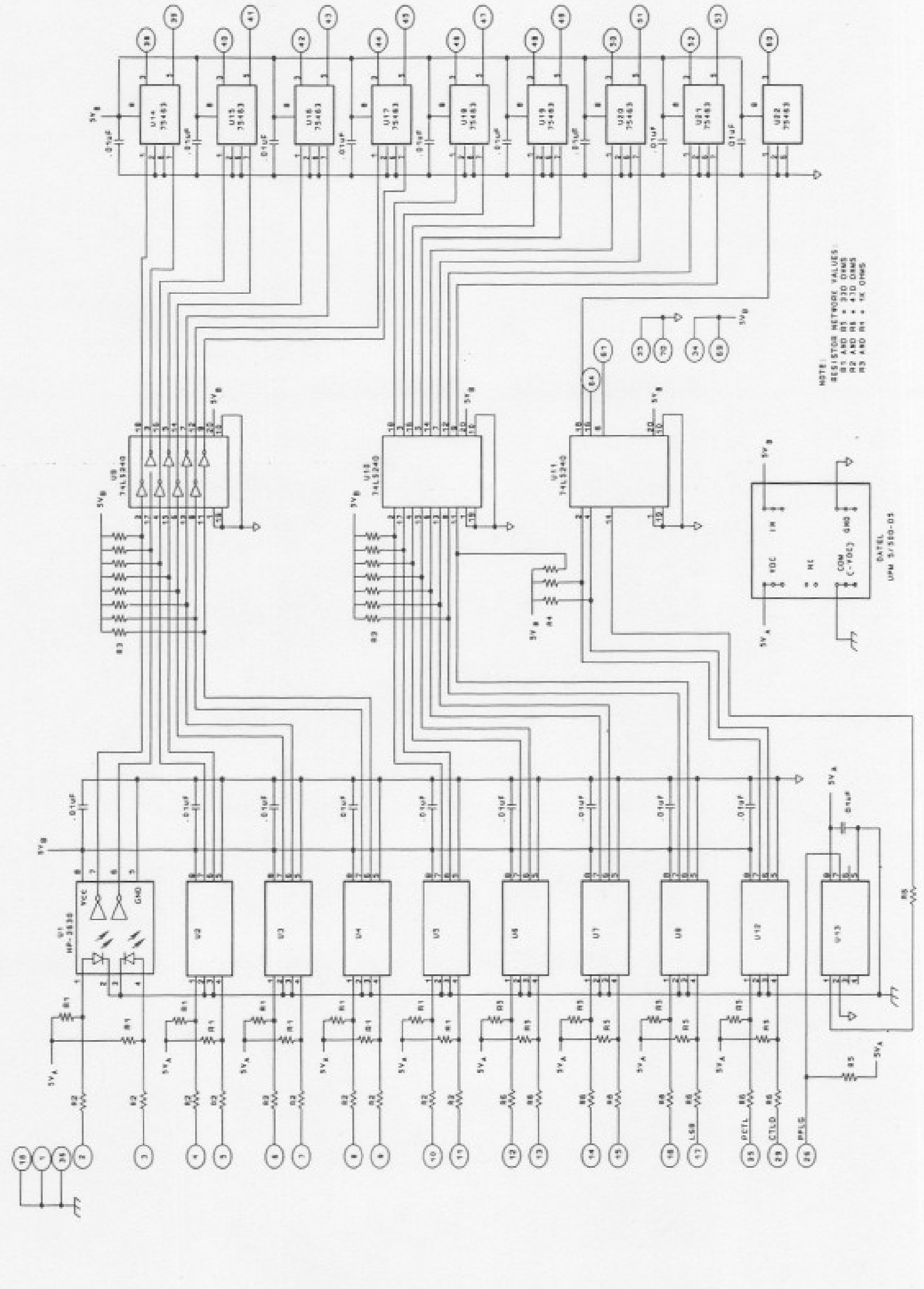

Figure 4. Optical isolator board.

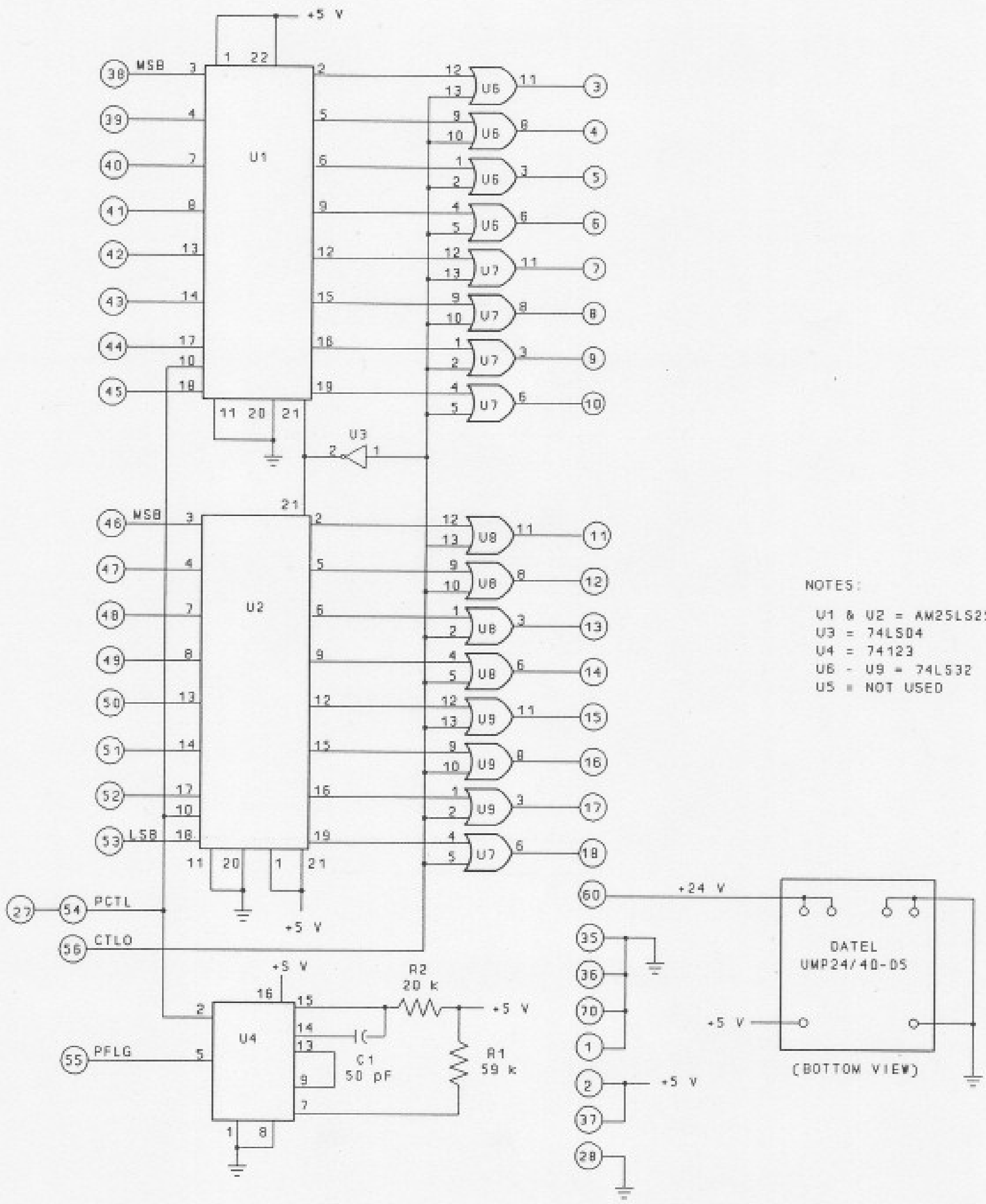

Figure 5. Interface board.

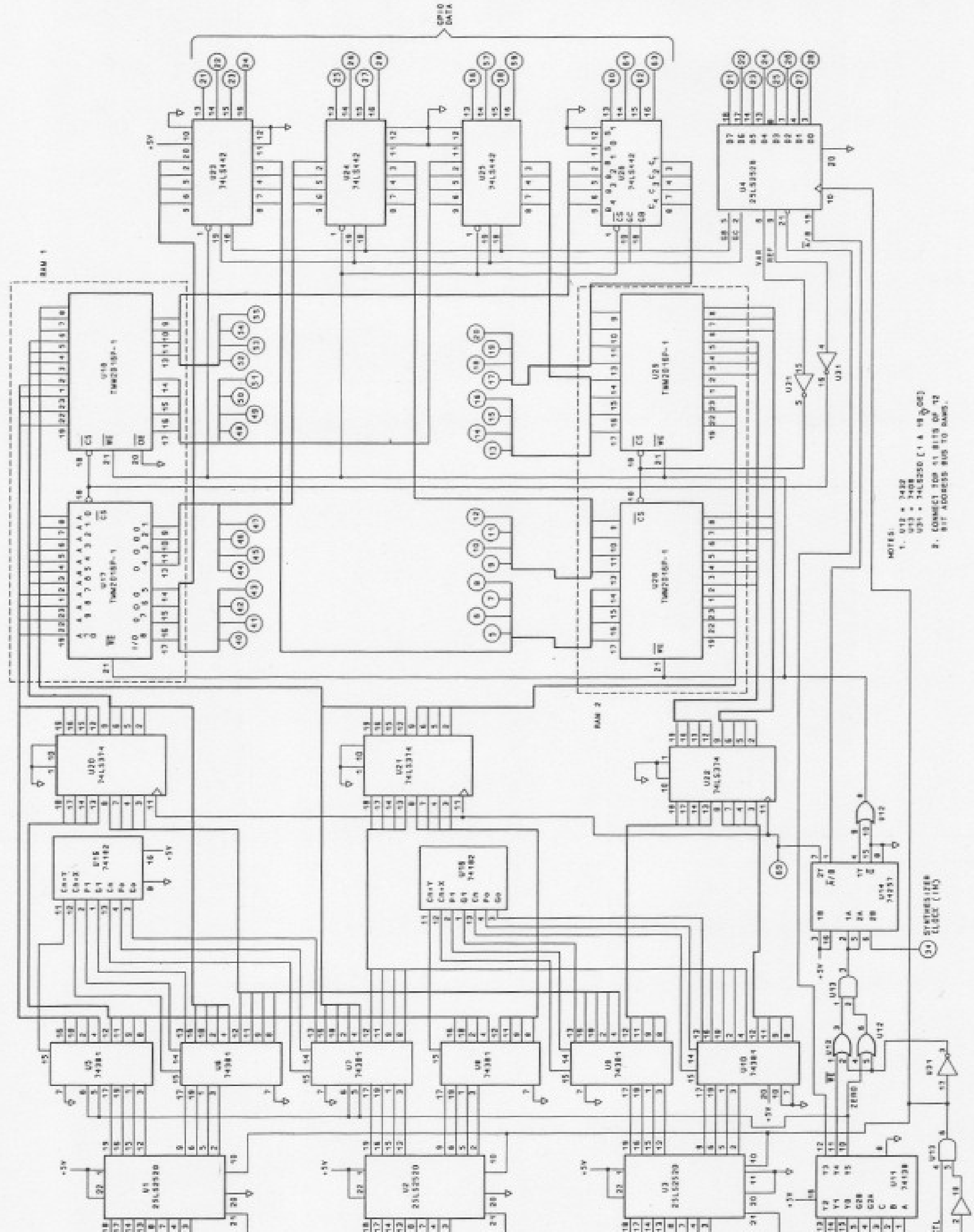

Figure 6. Memory board.

significant 8 bits) and 56-63 (least significant 8 bits) through transceivers U23-U26 to the data lines of the reference channel RAM1 (U17,U18) and the variable channel RAM2 (U28,U29). The transfer path and the RAM chip selects are latched into U4 before the data is transmitted, and the RAM write enable signals are derived from the PCTL signal through U12-U14. Addresses for RAM1 are generated in the address-counter circuit (U1,U2,U5,U6,U7,U15,U20 and U21) which is driven either from PCTL (during loading) or the synthesizer clock (during operation). Similarly, U2,U3,U8,U9,U10,U16,U21 and U22 comprise the address counter for RAM2.

Each memory holds 2048 16-bit words, and the waveforms can be constructed from all or a binary fraction of the 2048 data values. The selection of steps is made by programming the appropriate bits into U1,U2 and U3 to increment the address counters by 2^n where n can vary from 0-11.

2.1.4 Digital-to-Analog Converters

The waveforms stored in RAM1 and RAM2 are converted to low-level voltage waveforms in the digital-to-analog converter (DAC) Boards - one for each output channel (fig. 7). Data to generate the waveform appears at edge connector pins 40-57 and are latched into U1,U2 (for 16 bits) and U3 (for 18 bits) by the SYNTH CLK line. Multiplying digital-to-analog converter (MDAC) U8 converts the data word to a corresponding voltage level within a maximum swing of ± 10 volts. U10, an optional buffer amplifier, boosts the output current from ± 10 mA to ± 100 mA.

The output waveform amplitude is adjustable from 0 - 10 volts-peak by adjusting the dc reference to MDAC U8 (pins 24,25) between 0 - 10 volts-dc. The amplitude resolution of MDAC U9 used for this adjustment is 18 bits, and this data is loaded in 8-bit bytes into latches U4,U5,U6 from data on edge connector pins 5-12. The data in these latches is converted to a dc voltage in MDAC U9 and amplifier U2. Two edge connections, REF IN and INTERNAL REF, are provided on the board so that an external reference may be used to generate MDAC U9's reference voltage. If no external reference is available, the INTERNAL REF pin must be connected to the REF IN pin.

The dc-offsets for both U8 and U9 are programmable through U11 (a dual 8-bit DAC) and U13 (dual operational amplifiers), which provide dc-offset adjustment of about ± 250 ppm of full scale amplitude. The resistors connected to pins 2 and 6 of U13 convert the unipolar 10-volt outputs to bipolar ± 5-volt outputs.

2.1.5 Frequency Synthesizer

The system clock pulse, used to control waveform generation by latching new data into each DAC Board, originates in the Frequency Synthesizer Board (fig. 8). This is a composite, two-piece board that uses a commercial frequency synthesizer (Syntest model SM-102 with 5½ digit resolution) and a decoder/latch board, which holds the 5-BCD digits and the 3-bit frequency range data. The data is held in latches U1,U2 and U3, which are enabled for loading by U4. The Syntest board has been modified (voltage regulators removed) so that it will operate with +5 V and +15 V power supplies instead of the original +5 V and +24 V supplies.

2.1.6 Power Supply and Rear Panel Wiring

The generator requires one 5-volt, 5-ampere power supply for the digital circuitry and one ± 15-volt, 1-ampere power supply for the analog circuitry. These supplies are connected to the analog and digital hardware through an Amphenol connector as shown in figure 9. Pin connections for the rear panel GPIO connectors are also shown in this figure. The 50-pin connector attaches to the controller, while the 24-pin IEEE-488 type connector provides optically isolated GPIO-data to the SWITCH BOX (described in 2.4) and other peripheral devices as needed.

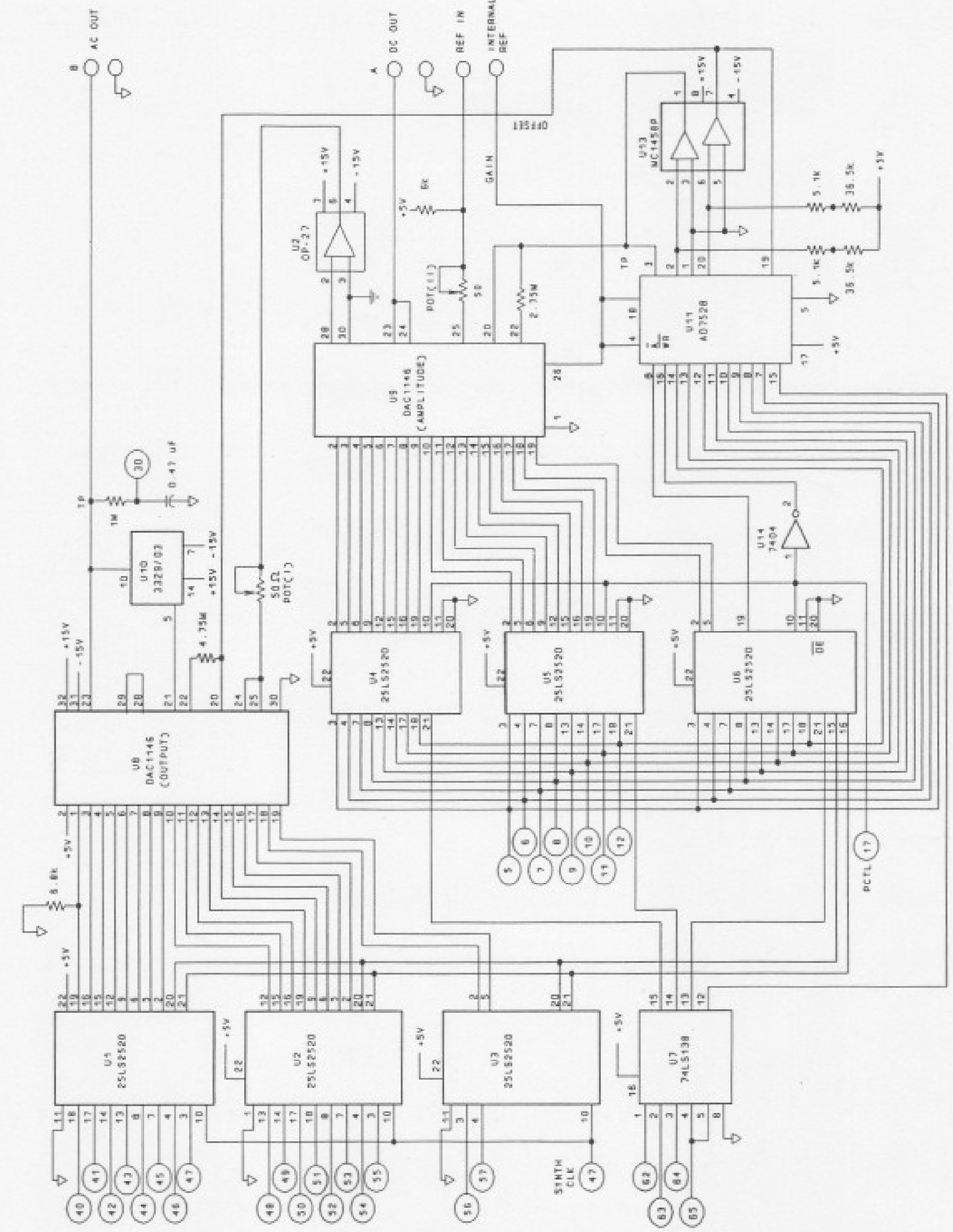

Figure 7. DAC board.

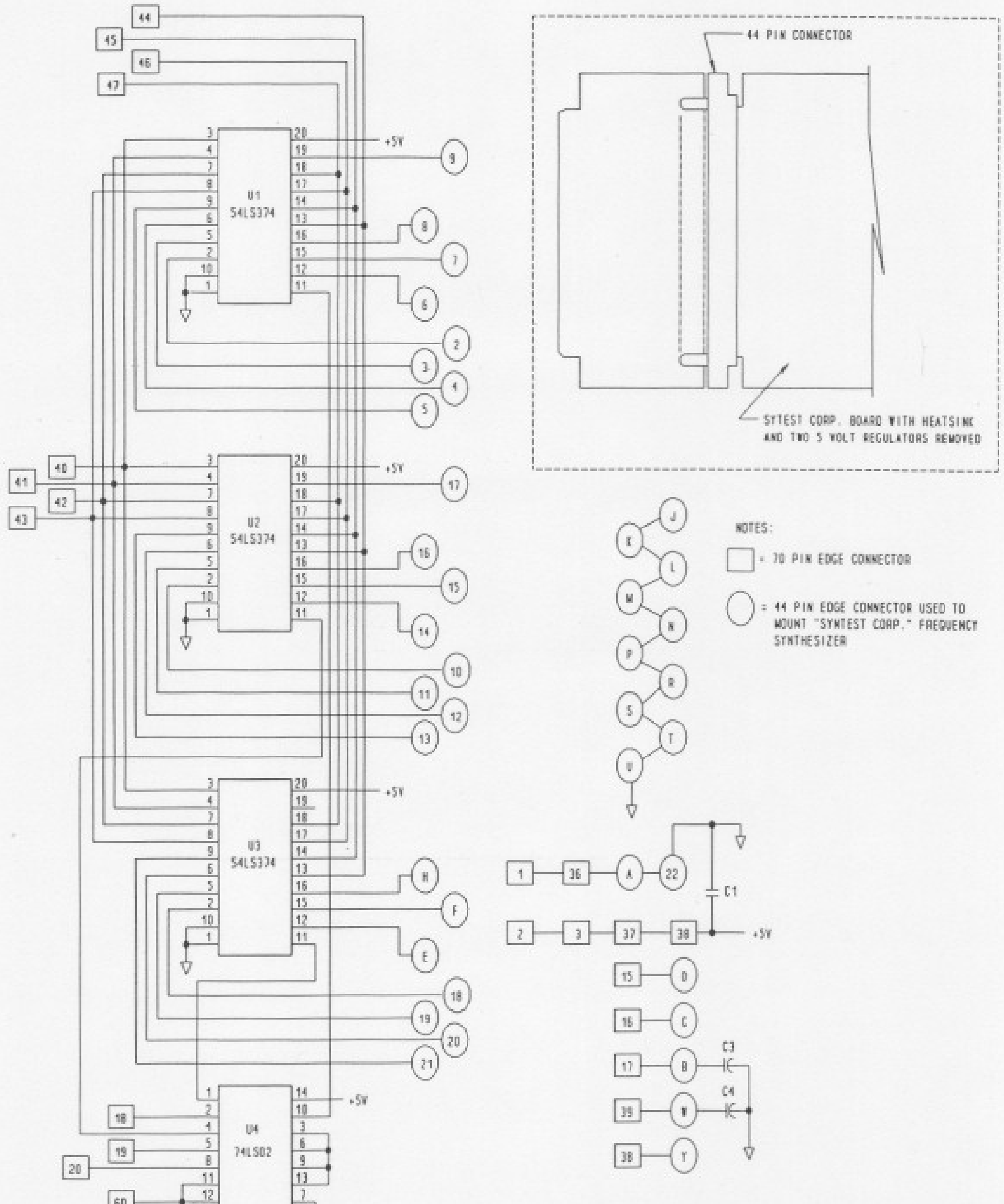

Figure 8. Frequency synthesizer board.

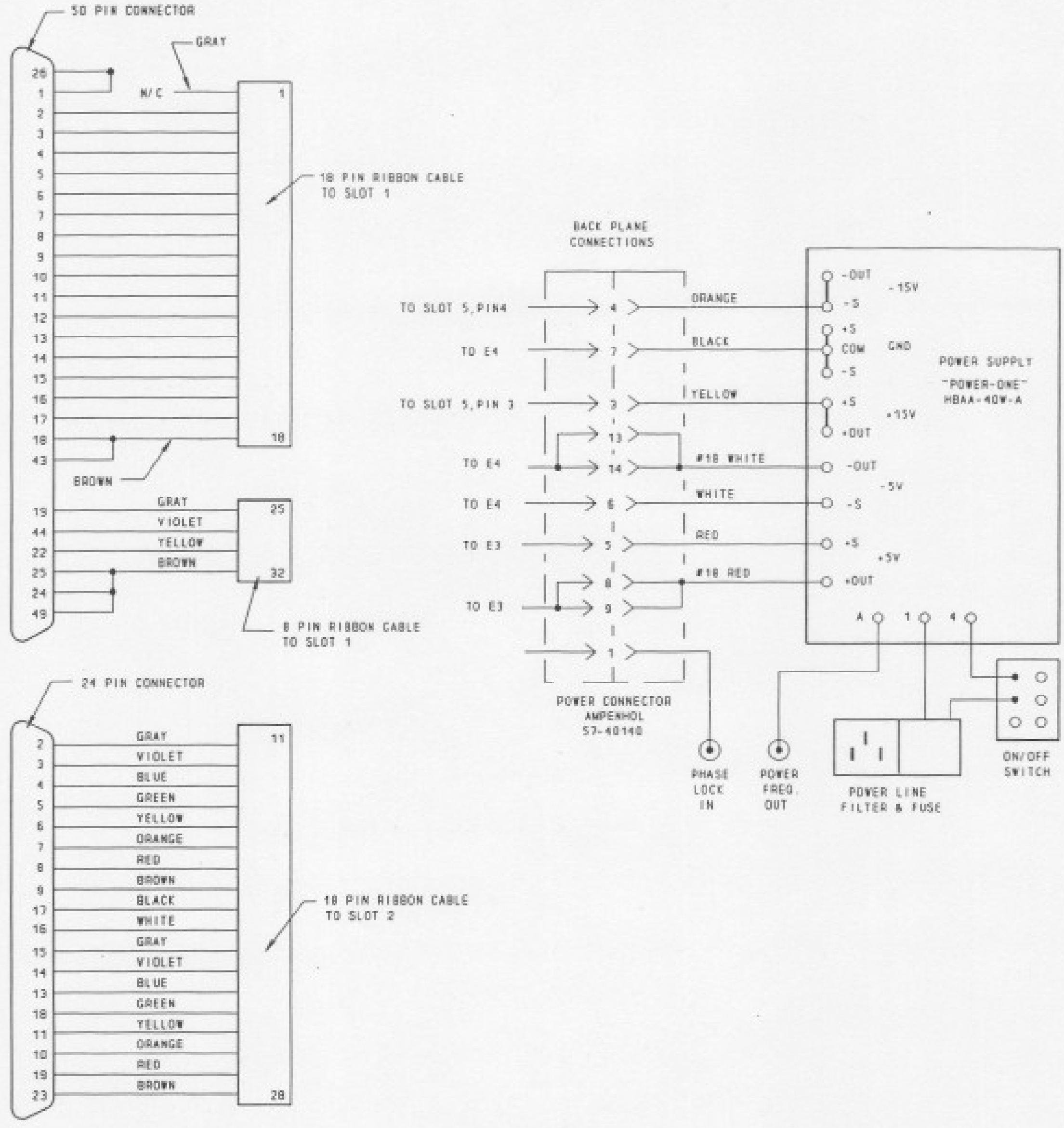

Figure 9. Rear panel GPIO connections and power supply connector.

2.1.7 Board Location and Back Plane

The generator consists of six boards, occupying slots 1,2,3,5,7,9 (see fig. 10) of a commercial card cage made by Cambion. The back plane wiring diagram for this card cage is shown in figure 11.

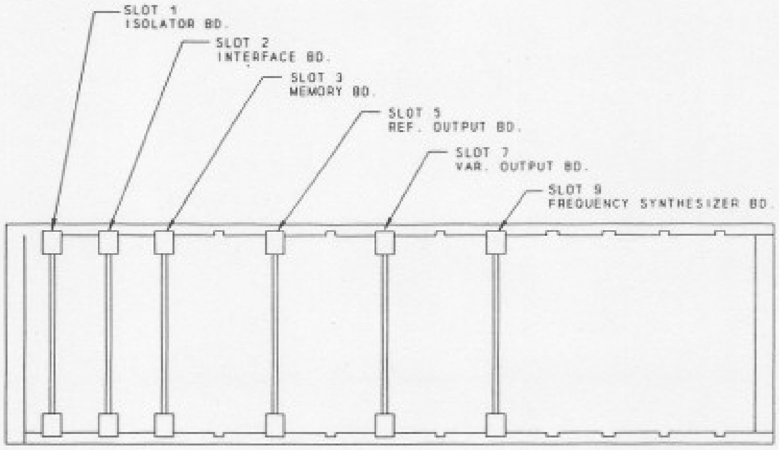

Figure 10. Cambion cardcage slots showing generator board locations.

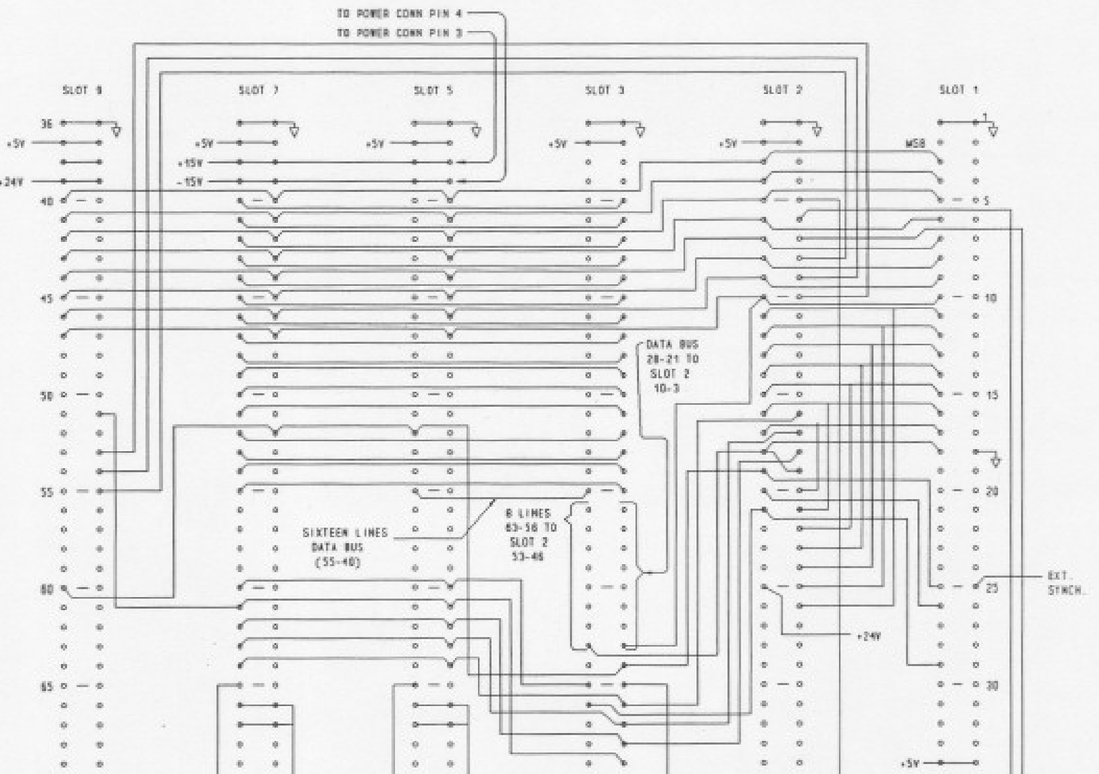

Figure 11. Generator back plane wiring.

2.2 Voltage Amplifier

2.2.1 Circuit Description

The voltage amplifer (described in detail in [7]) was primarily designed to boost the output amplitude of the digital generator in order to provide the nominal 120 or 240 rms voltage output of the power calibration source. This amplifier has a fixed gain of 40 and can provide a maximum output voltage swing of 970 volts peak-to-peak or 340 V rms at 100 mA rms. The bandwidth is from dc to 150 kHz and at 60 Hz the observed no-load, short-term amplitude and phase instabilities are ±5 ppm and ±5 micro-radians, respectively. The amplifier design uses high voltage N-channel MOSFETs in the output driver stage together with a unique circuit topology of opto-isolators between the low-level input stage and the high-level output stage. In addition, the amplifier was designed to supply up to 100 mA rms to accommodate the burden requirements of electrodynamic type meters without causing significant error. A prime goal was to maintain the excellent short-term amplitude and phase stability inherent in the digital generator.

Figure 12a shows the circuit diagram of the voltage amplifier. Each polarity output driver uses a pair of 1000 V, N-channel MOSFETs to provide a 2000 V capability to each polarity driver. The stacked pair of MOSFET drivers provides an operating voltage safety factor of two for each device when operating at maximum peak-to-peak output signal swing. In addition, the total power dissipation for each driver pair is equally divided between two devices. A small trimmer capacitor C23, helps to balance the differential capacitance across the opto-isolator pair U3 so that the high frequency response for the positive and negative output signal can be better matched. Diodes CR3 & CR4 separate the signal at the output of U2 and steer the respective polarities to each driver. U2 is a high gain wideband operational type amplifier that provides the major portion of the open-loop gain for the voltage amplifier. U2's input offset errors are reduced by the gain of U1, which in effect servos out any offset errors at the summing junction. In order for this scheme to be effective, U1 must be a precision low-offset type of amplifier. A local compensating network (C1,R7) around U2 is necessary in order to shape the gain-bandwidth response to avoid loop oscillation. The diodes at the summing junction protect the amplifiers U1 & U2 against high voltages during turn-on and output voltage slew-rate limiting.

Figure 12b shows the voltage amplifier's monitoring, control, and overload protection circuitry. U4 converts the amplifier's ac output to a dc value corresponding to the output rms amplitude, while the low pass network (R20, R21, C11, C12) and diodes CR1, CR2 enable relatively small dc output offsets to be monitored in the presence of large ac output signals. Switch SW2 selects which of these signals is to be displayed by the front panel meter. Power to the output stage is controlled by relay K3, K2, and K1, switch SW1, and the state of flip-flop U1. U3 serves as a bipolar peak detector that toggles U1 when a preset output current is reached, which in turn removes the power to the output drivers. The output stage will remain shut down until U1 is reset by Switch SW1.

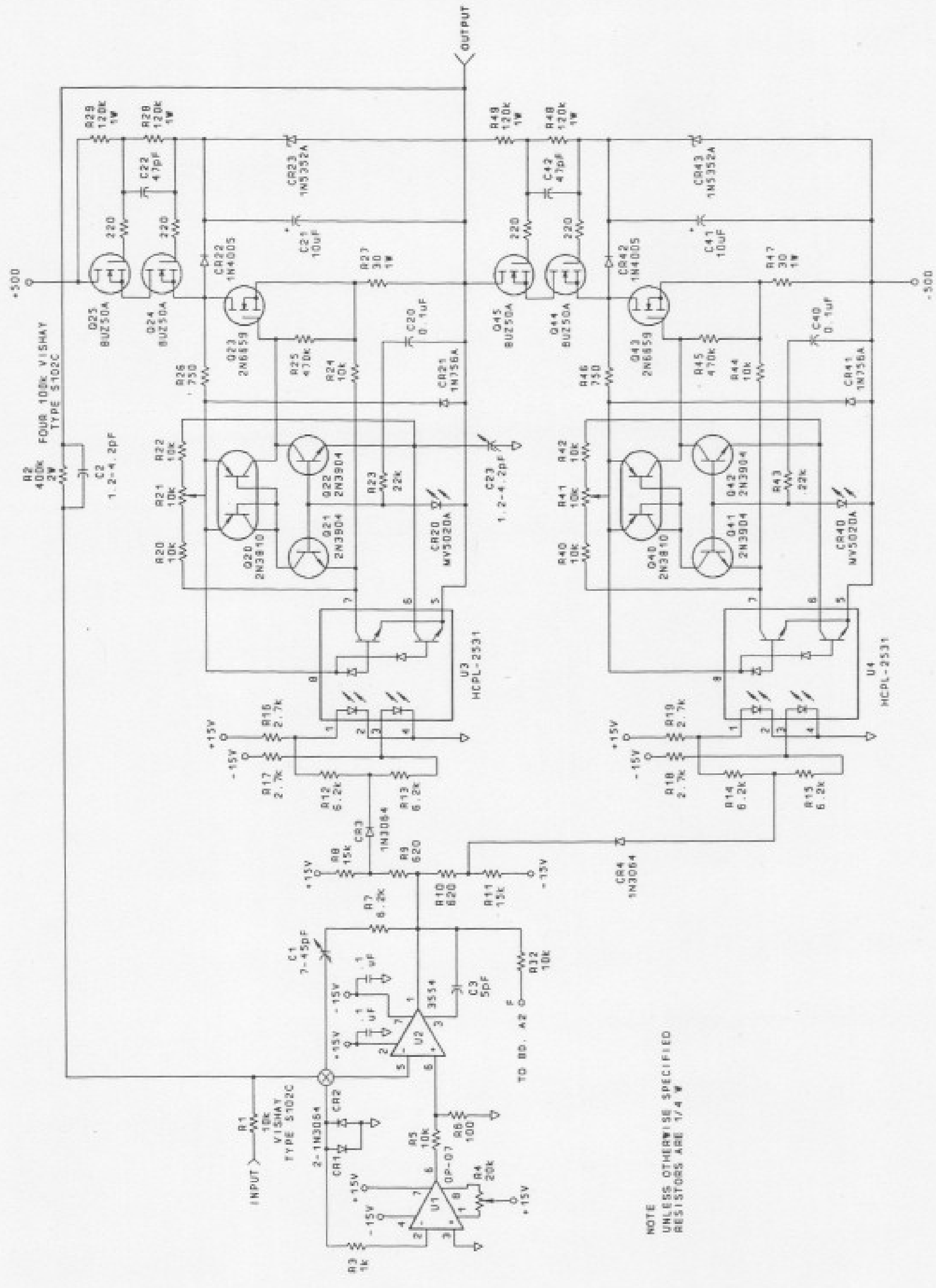

Figure 12a. Complete circuit diagram of precision voltage amplifier: Amplifier circuit.

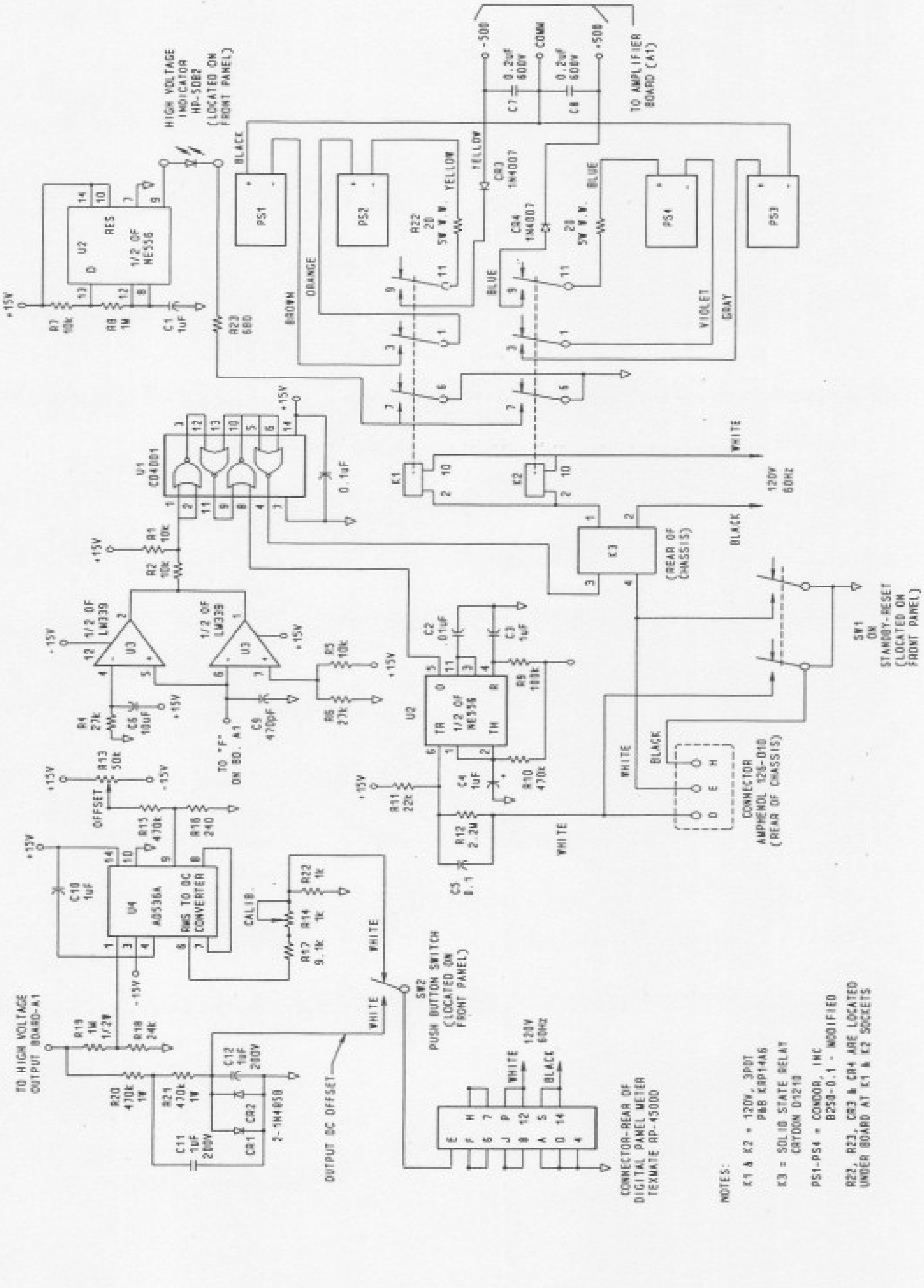

Figure 12b. Complete circuit diagram of the precision voltage amplifier: Control and overload protection circuitry.

2.3 Transconductance Amplifier

2.3.1 Circuit Description

A complete circuit diagram of the transconductance amplifier (described in detail in [8]) is shown in figure 13. The operation of the circuit can be described as follows: A voltage applied to the main input at F, produces a current into a load connected across the output terminals. The load current causes a voltage drop across R20 which is amplified by a factor of 10 by a differential amplifier circuit composed of U5 and resistors R16 through R19. The output of voltage the differential amplifier is fed back via R21, where it is compared with the input voltage at the summing junction of U2. Thus, the output current is made proportional to the input voltage. Within the output compliance voltage range, the transconductance amplifier will maintain the same output current for a fixed input voltage regardless of any load change. Since the differential amplifier circuit that senses the voltage across R20 is part of the feedback loop, its phase lag creates a potential source of instability for which compensation must be added. Compensation is provided by modifying the loop-gain response to have a single dominant pole by means of a capacitor C7 across R21.

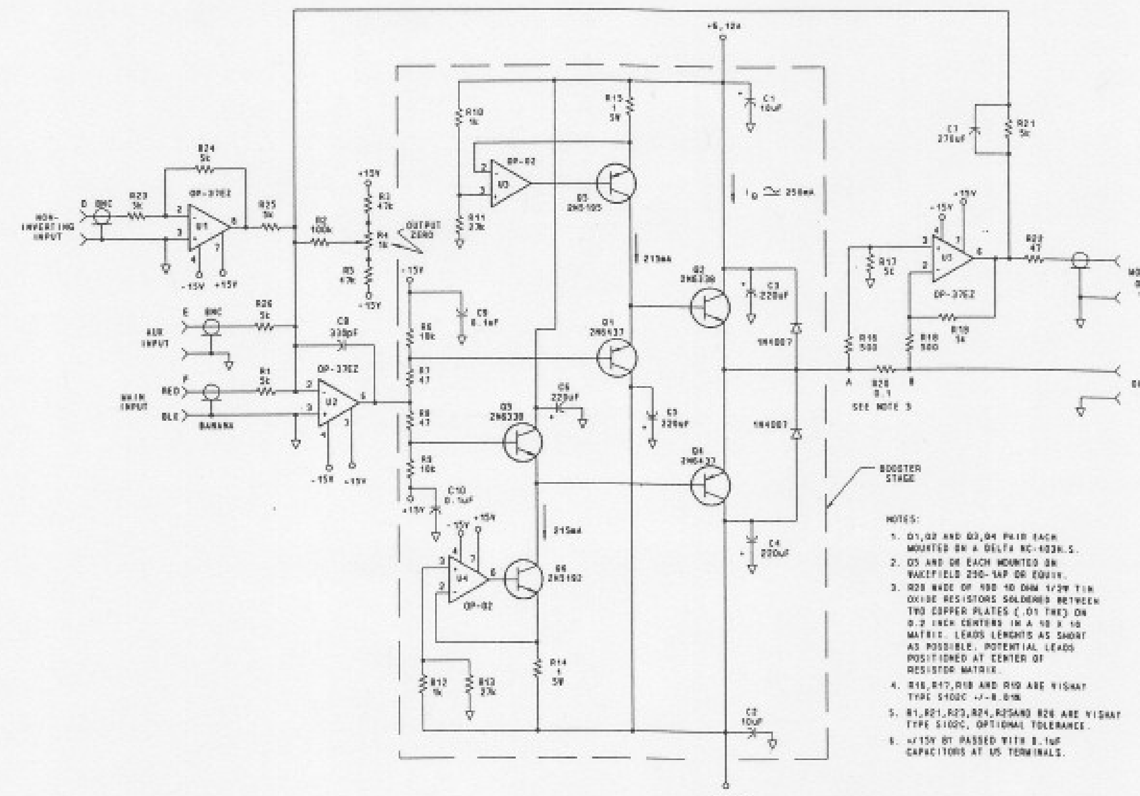

Figure 13. Complete circuit diagram of the transconductance amplifier.

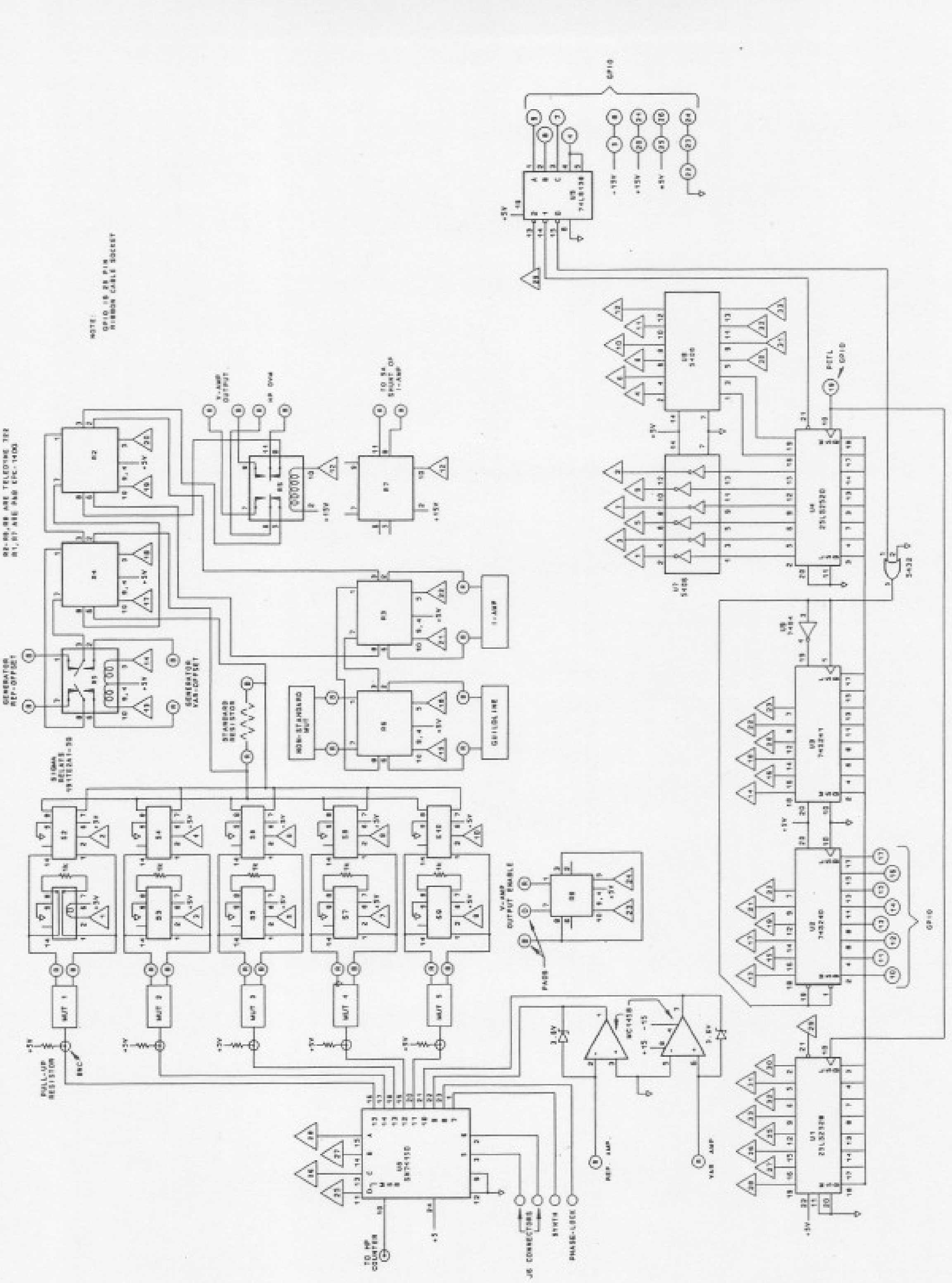

Figure 14. Complete circuit diagram of the switch box.

The output power booster stage, consisting of Q1 through Q4, is a class AB complementary-symmetry emitter follower designed to provide an output current of approximately 15 amperes. Amplifiers U3, U4, and transistors Q5, Q6 provide complementary fixed current sources for the output booster stage. The output stage is designed to have a quiescent operating current of about 250 mA, which keeps crossover distortion to an acceptable level. Practical considerations for achieving stable operation require proper high-frequency power supply bypassing with both electrolytic and ceramic capacitors at each collector of the booster stage. Also, careful attention must be given to circuit grounding and load ground returns.

2.4 Switch Box

The main purpose of the switch box is to facilitate the automation of watt/watthour meter calibration and system test procedures. The switch box is also used for source monitoring. A complete schematic of the switch box circuitry is given in figure 14. The switch box contains switching and control circuitry which enables up to 6 meters under test, MUT1-MUT5 and a non-standard MUT, to be calibrated conveniently. For each of the first 5 meters, MUT1-MUT5, there are two switch box inputs provided, one for the MUT's analog current output and one for the MUT's pulse output, which is assumed to be configured as a standard TTL, open-collector. The non-standard MUT is provided with only one analog voltage input. The relays in the box are remotely programmed over a portion of the HP GPIO bus that has been channeled through the digital generator. These bus output pins are located on card cage slot 2 pins 11-28 and exit the digital generator via a rear panel IEEE-488 type connector (see fig. 9).

2.4.1 Circuit Description/Programming

The triangular, numbered tabs in figure 14 represent wiring traces present on the actual circuit board of the switch box. The traces were represented this way for the sake of schematic clarity. Connections to and from the switch box, measuring devices, and MUTs are made at the circles labeled "R" and "B", for "red" and "black" banana plugs, as well as at other connectors labeled in the figure. The analog outputs of each standard wattmeter are connected to the switch box inputs as shown in figure 14. For energy measurements, each MUT's open-collector pulse output is first "pulled up" to +5V before being channeled through multiplexer U6 to the switch box's counter output. The frequency of the pulse output may be measured by any frequency counter capable of being connected to the controller via an IEEE-488 bus.

For power measurements, the circuit includes an array of both latching and non-latching relays. The relays are switched using either line drivers, (U1,U4), or latches, (U2,U3). The non-latching relays (S1-S10) are switched using the buffered outputs of latches U1 and U4. These relays are used to switch the current output of a selected MUT from a low-precision 1-kΩ resistor to the standard 1-kΩ resistor used in an actual power measurement. The latching relays (R1-R6) require only a brief pulse from line drivers U2 and U3. This pulse is provided by programming U5, a 3-8 line decoder, to toggle output pin #15 low by toggling pins 4,5 with the generator's CTL0 signal (control bus enable). These latching relays (R1-R6) channel the voltage outputs of various devices to the system DVM input. These voltages include the generator's dc-offsets, the output of the non-standard MUT, the Guildline 1300A Transfer Standard's output (used in the Power Bridge [10]), the transconductance amplifier's monitor output, and the voltage amplifier's output. With the ability to automatically sample these signals using the system DVM and counter, power and energy measurements can be made simultaneously in intervals of ten seconds per MUT.

The generator's control/data bus is also used for programming the switch box (for programming codes, see sec. 3.5.4, table 2). The programming steps are similar to those required to program the generator's internal boards. These steps are performed in the subprogram, Switch, as described in section 3.4.4.

3 SOFTWARE

3.1 The MET_6 Program

There are three basic functions performed by the MET_6 software. First, to control the digital generator, second, to characterize and correct generator errors so that they fall within certain bounds, and third, to semi-automate wattmeter testing procedures. The latter tasks are performed by a group of routines that are very system dependent and may be omitted with only minor changes to the basic software. The MET_6 program was written initially in HP BASIC version 2.1 for use on an HP 9836C desktop computer. With the required binary drivers, the program will also run on HP BASIC 3.0 and 5.0 on HP series 200 and series 300 machines (equipped with the series 200 keyboard).

3.2 Software Considerations Based on System Configuration

3.2.1 Switch Box vs. no Switch Box

There are several MET_6 variables that must be properly defined, depending on whether the power/energy calibration system includes a switch box (as described in sec. 2.4) or not. Certain subprograms use these variables to determine the proper measurement algorithm to follow (see sec. 3.4). Table 1 below lists these variables and their values based on system configuration. For a complete explanation of all variables used in the MET_6 program, see APPENDIX A.

Table 1. Settings for System Configuration Variables

Configuration	Variable	Line #	Value
No switch box, 4 counters, 1 HP 3457 A DVM with Multiplexed input channels.	No_switch_box	595	1
	Col_max	425	4 or6
	@Dvm	330	Any unique HPIB interface select code (ISC)
	@Cntr1	335	unique HPIB ISC
	@Cntr2	340	unique HPIB ISC
	@Cntr3	345	unique HPIB ISC
	@Cntr4	350	unique HPIB ISC
Switch box, 1 counter, 1 HP 3456A DVM.	No_switch_box	595	0
	Col_max	425	4 or 6
	@Dvm	330	unique HPIB ISC
	@Cntr1	335	unique HPIB ISC

To set the variables listed above in table 1, the procedure is as follows:

1. "PAUSE" the program and "EDIT" the line # given;
2. Change the value in the indicated line and "ENTER" it;
3. "RE-STORE" the program (or, "PURGE" and "STORE").

3.3 Generator Control Routines

Figure 15 outlines the relationships between the various routines and is intended to show only their basic interactions, not the detailed program execution flow. These routines may be separated into two groups consistent with their relative utility. The first to be discussed, the interactive I/O group, share a common symmetry of operation that allows them to be easily understood and maintained. These are the Adj_phase, Synth_load, Amplitude, and Cor_dac subprograms. The rest of the subprograms may be loosely grouped together, based on the fact that they perform specialized tasks that require little or no interaction with the user.

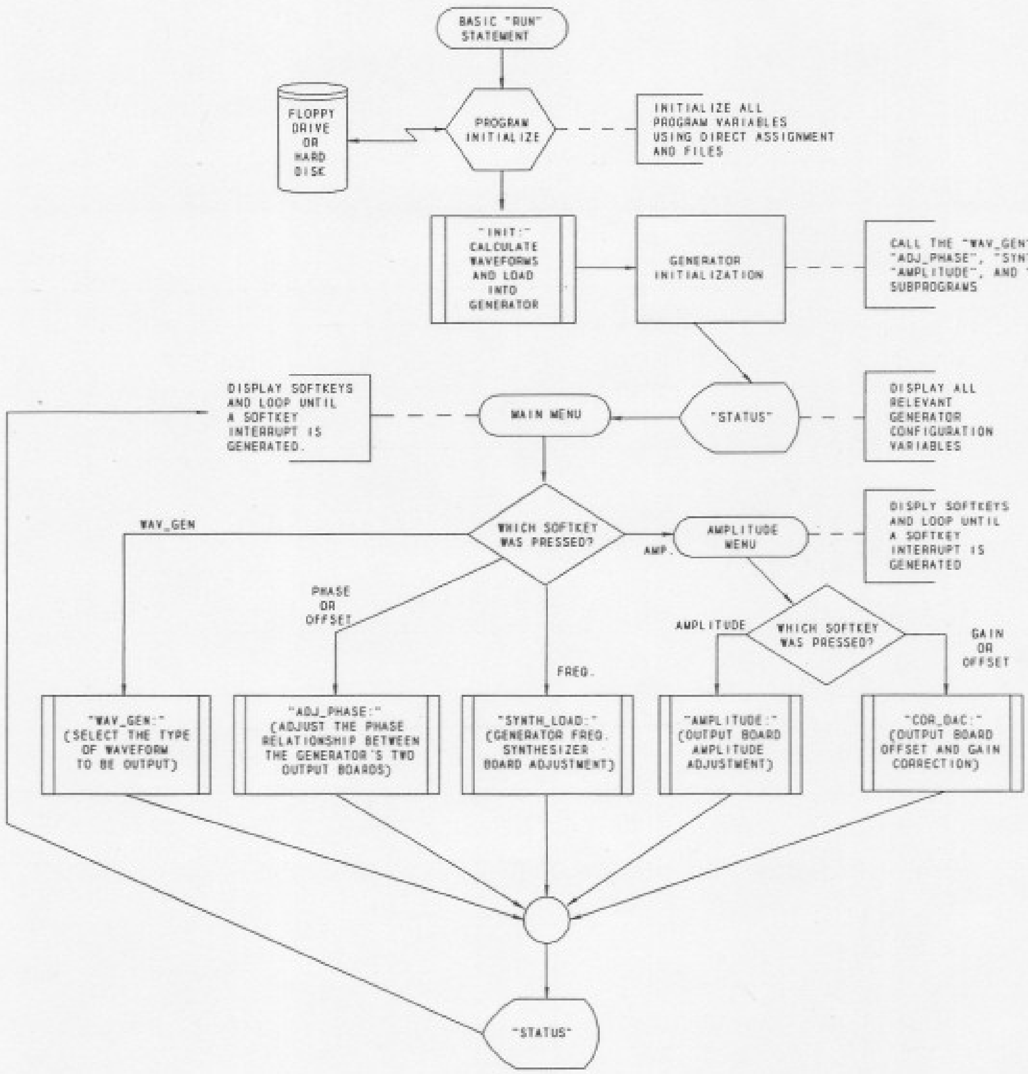

Figure 15. Basic relationships of various routines in the MET_6 program.

In the discussion below, the term 'function key' refers to the ten rectangular keys labelled k0 through k9 found on all HP series 200 desktop computers. Two lines at the bottom of the computer's screen indicate the alphanumeric labels assigned to each k0-k9 key. For clarity, the labels that appear on the computer screen during the MET_6 program execution are enclosed in quotes in the discussion below. Also, numbers, shown in brackets refer to the length of a string, not to the references at the end of this report.

3.3.1 Interactive I/O Routines

3.3.1.1 Amplitude

To enter this routine, press the "AMP." function key when in the main menu. Three additional choices will appear - "REF_WAVE", "VAR_WAVE", or "EXIT". The first two choices are for reference waveform and variable waveform amplitude control, respectively, while the third will cause an exit to the main menu. Pushing the "REF_WAVE" function key will cause four additional keys to appear - "AMP", "GAIN", "OFFSET", AND "LINEAR". In order to enter the Amplitude subroutine, press the "AMP" function key. Once inside the subroutine, there are three ways in which the user may alter the amplitude value: 1) pressing the ± 100 to ± 1 least significant bit (LSB) function keys, 2) turning the keyboard knob, or 3) directly entering the desired value. The three columns located on the lower portion of the screen indicate the amplitude, change from the initial value in LSB's, and the apparent and actual amplitude DAC settings, respectively. On the screen, the binary string under the "APPARENT" heading is the DAC amplitude setting corresponding to the amplitude value on the lower left of the screen, while the binary string under the "ACTUAL" heading includes any gain corrections that may be present.

Gain, dc offset, and amplitude linearity corrections are handled by the routine Cor_dac (see sec. 3.3.1.3 below). Since the amplitude values Ref_amp and Var_amp are used within the software in many calculations, including gain corrections, the Amplitude routine should only be used to set up the ideal (desired) amplitude. The Amplitude routine automatically calls Cor_dac after every amplitude change, resulting in a constantly updated dc offset voltage correction corresponding to a linear correction curve. In addition, Amplitude routine computes a new gain correction according to a similar correction curve set up in Cor_dac. After a call to Amplitude routine, the internal variables Ppm and Voltage (see app. A) are corrected to values allowable by the 18-bit amplitude DAC (U9 on the output board). A discussion of how to characterize the generator and construct these curves is given in section 5.

Due to software gain DAC corrections, which are transparent to the user, the Amplitude routine behaves rather unpredictably when incrementing the amplitude DAC near full scale, which is the amplitude where the voltage amplifier is automatically enabled. The DAC settings on the lower right of the screen were included mainly as a way of keeping track of exactly what is happening during this discontinuity of operation.

If, when the keyboard knob is used to increment the gain or amplitude, a value is selected that exceeds the maximum output DAC binary setting, the user will be alerted to this fact and the output DAC set to its maximum value.

Finally, provisions are made in the Amplitude routine to remotely enable or disable the voltage amplifier with the (k0) function key. Enabling or disabling is accomplished by entering the Amplitude routine via the "REF_AMP" key and then using the "ENABLE" (k0) function key. The amplifier is automatically enabled when the operator enters a reference voltage above 7.071 V (full scale of amplitude DAC for sinewave functions).

3.3.1.2 Adj_phase

To enter this routine from the main menu, use either the "PHASE" or "OFFSET" function key. To set up a direct reading phase difference between the reference and variable waveform channels, first use the "OFFSET" key to adjust the inherent channel phase error to zero and then use the "PHASE" key to set the desired phase difference. The phase difference between the two channels is computed as the difference between the global variables Phaz and Offst. For example, the effective phase difference between the two waveforms resulting from a Phaz of 30 degrees and an Offst of 1 degree is 29 degrees.

The behavior of the I/O subroutines in Adj_phase is similar to that of Amplitude, Cor_dac, and Synth_load. The main difference between Adj_phase and the others is that the actual output of the new waveform values and address difference is done in another subprogram, Change_phase, which is described in section 3.3.2.

Adj_phase has two modes of operation, fine and coarse. In the coarse mode, only the RAM address difference is changed, resulting in a phase resolution of 1/2048. In the fine mode, although the algorithm used to compute the values in RAM allows for an extremely fine resolution, a reasonable value for the highest resolution was chosen to be 1 μrad. When changing the phase at intervals lower than 1/2048, Adj_phase calls Change_phase with Sel_ect=2 (1=coarse, 2=fine) and also passes the real arrays Real_var and Var_diff to Change_phase, where it recalculates the integer buffer values and outputs them to RAM.

3.3.1.3 Cor_dac

This subprogram may be accessed from the main menu by pressing the "AMP." function key, choosing either "REF_AMP" or "VAR_AMP," and then deciding whether Cor_dac is to process "GAIN," "OFFSET," or "LINEARITY." In each of the three cases, the operation of Cor_dac is as follows:

(a) <u>"GAIN" key (k3 or k8)</u>: The functionality of the routine is similar to that of the Amplitude subprogram (see discussion above in sec. 3.3.1.1). The values across the lower portion of the screen represent the gain correction in volts presently being applied to the Ref_amp or Var_amp variables, the change (in ppm of reading) between the desired voltage (Ref_amp or Var_amp) and the corrected voltage, and the "APPARENT/ACTUAL" DAC settings, respectively. The routine uses the input parameters Slope, Intercept, and Voltage to calculate Ppm and then calls Amplitude (see fig. 15) to load the new corrected amplitude DAC word into the appropriate generator output board. An additional function performed by the call to the Amplitude routine is to return to Cor_dac a corrected value of Ppm, since only discrete values are permitted by the 18-bit DAC. In the gain mode, Cor_dac does not directly manipulate Ppm. Instead, Slope is altered and Ppm recalculated.

(b) <u>"OFFSET" key (k4 or k9)</u>: Operation is similar to "GAIN" above. Cor_dac controls the amplitude DAC's offset voltage by altering the setting of the 8-bit MDAC (U11 on the output board). To do this, Cor_dac must be called with the input parameter, Dac_choice, set to 1. In the offset mode, Cor_dac uses the input parameters Slope, Voltage, and Intercept to calculate an 8-bit value, Vo_lts, that is then loaded into U11 of the appropriate output board. In this mode, Cor_dac does not manipulate Vo_lts directly. Instead, it increments or decrements Intercept. This seems to be the best method because the characteristic slope of the dc offset correction curve seems to stay relatively constant over time. The values displayed across the lower portion of the screen are from left to right the applied DAC setting in volts and the change (in LSB's) between the present 8-bit DAC setting and the initial setting (shown at the top of the screen).

(c) <u>"LINEARITY" key (k2 or k7)</u>: In this mode, Cor_dac controls the A-channel of the 8-bit dual MDAC (U11) on the appropriate output board. The behavior of Cor_dac in this mode is similar to that discussed in the above "OFFSET" mode with the exception that Cor_dac manipulates Voltage directly with no reference to either of the input parameters Slope or Intercept. The purpose of this routine is to ensure that the amplitude errors are constant (the same ppm of reading) throughout the voltage range.

A discussion of how this is accomplished is given in the section on generator characterization. Once this value is determined, it should not be changed until the next output board adjustment, or else serious amplitude non-linearity may result.

3.3.1.4 Synth_load

This is the controller subprogram for the generator's frequency synthesizer board. To enter this routine from the main menu, simply press the "FREQ." function key (k7). The methods of data input are identical to the Amplitude, Adj_phase, and Cor_dac routines, which are direct entry, knob turning, or choosing a defined function key.

Because of the synthesizer's design, the actual value of an LSB in Hz is highly dependent on which frequency range the synthesizer board is presently set on. In the event of a frequency change, the Synth_load routine determines both the proper range and frequency resolution of the synthesizer board's output, the latter of which may range from as fine as 10 μHz to as coarse as 100 Hz.

The internal structure of the subprogram is very straightforward. Since there are no frequency corrections, the Synth_load routine basically performs only two tasks: 1) to handle user I/O, and 2) to update F, the variable containing the current frequency synthesizer setting. The Synth_load routine contains only one input parameter, Freq, that is usually passed by value. The seemingly redundant presence of both F and Freq stems from the fact that Freq does not always indicate the true synthesizer setting. For instance, when Freq = 10,000 (10.000 kHz), F = 10.000 and F_unit$ = "kHz." Calling Synth_load with Freq = 0, an illegal value for the synthesizer board (see data sheet), causes the subprogram to initialize its I/O handling subroutines, Format and Softkey. Thus, Synth_load may be used to simply set up a synth board setting or to also interact with the user.

3.3.2 Subprograms to Perform Specific Generator Control Tasks

3.3.2.1 Change_phase

This is a subprogram invoked by Adj_phase that performs the actual generator channel phase changes. Depending on the value of Sel_ect, Change_phase will either calculate the address offset between the two channels and call Cnt_vals to output this value to the memory board (Sel_ect=1) or Change_phase will use matrix operations to calculate new variable channel RAM values and then call Out_put to load the new values and set the address offset (Sel_ect=2). This process can be more clearly understood by examining fig. 15.

3.3.2.2 Cnt_vals

This subprogram is transparent to the user and is responsible for loading the memory board's RAM address counter with the desired increment in order to create a phase difference between the two channels. The desired increment is passed (usually by value) to Cnt_vals via the input parameter, Address_add. The variables L2_mss and L2_lss, which correspond to latch U2 on the memory board, are updated during each call to Cnt_vals.

3.3.2.3 Init

This subprogram initializes the appropriate channel's Real and Diff arrays as well as the integer buffer used to output the values to RAM. Depending on the parameter Function$, Init will calculate 2048 sin, ramp, triangle, square, zeroes, ones, minus ones, or arbitrary function values, load the values into the Real, Diff, and BUFFER data structures, output the BUFFER to the appropriate memory(s), and finally, reset the correct amplitude values on the output boards.

3.3.2.4 Out_put

This is a subprogram responsible for loading the appropriate memory channel(s) with the contents of the INTEGER buffer passed to it. Out_put also takes care of restoring the proper counter increment (RAM address offset) by calling Cnt_vals.

3.3.2.5 Process_key

This is a subprogram designed to eliminate the use of the Basic "INPUT" command, which halts program execution and disables the function key interrupt routines when waiting for an "ENTER" command. Process_key acts upon a keyboard interrupt and checks the keyboard buffer. The subprogram then interprets the data as either legal or not, depending on the Message input parameter. Upon the occurrence of an "ENTER" key, Process_key then either returns the accumulated data in Value (numeric input only) or shifts it into Value$ (alpha-numeric data). Process_key may also update Message according to the data passed to it (see the code documentation).

3.3.2.6 Set_inc

This subprogram sets up the memory address counter according to the variable Inc, the number of steps per period. The desired increment is stored in another variable, Inc_rement, because Inc must be set to zero during certain memory board operations.

3.3.2.7 Status

This subprogram is usually called after the completion of an interactive I/O subprogram. Status displays the following parameters: Fr_eq, F*Inc, Inc, Rivd, Civd, Offst, Phaz, Ref_amp, Var_amp, R_func$, and V_func$.

3.3.2.8 Step_num

This is a subroutine that allows for the manipulation of the number of steps per period of the output waveform. The default value of 2048 may be decreased in binary steps down to a minimum of 2 steps per period by using either the defined function keys or by direct entry. The basic steps taken to accomplish this are: 1) call Set_inc to set the proper memory board counter increment 2) call Synth_load to set the new sampling rate 3) recalculate the rms value of the waveform, and 4) restore the proper phase value by calling Change_phase.

3.3.2.9 Wav_gen

This subroutine handles the I/O tasks for the subprogram Init (see 3.3.2.3 above). The type of waveform to be loaded in memory as well as the desired channel, may be chosen by using the defined function keys.

In order to load a specific arbitrary waveform, the program must first be "PAUSE"d and line 4100 of Init modified. To preserve this function as the arbitrary waveform the program containing this updated line 4100 must be "RE-STORE"d.

3.3.2.10 Machine_state

This subprogram may be accessed from the main menu by using the "MORE" (k0) key and then the "GEN VALS" (k4) key.

The purpose of this routine is to allow the user to either store or retrieve all major generator variables from disk storage. These groups of variables (generator states) are contained in the files "Gen_state," the file reserved for temporary state storage, and "Met6vals," the file reserved for permanent (default) state storage, resident on the MET_6 program disk.

Once inside the Machine_state subprogram via the "GEN_VALS" key, the three resulting choices are "EXIT" (k0), "DEFAULT" (k4), and "TEMP" (k9). "DEFAULT" is used to retrieve/store values from "Met6vals" while "TEMP" operates on "Gen_state." To store the present state as the default state, use the "DEFAULT" and then "STORE" function keys. To retrieve the generator's default state from disk, use the "DEFAULT" and "RETRIEVE" keys. The same procedure applies when using the "TEMP" function key.

The "GEN VALS" key may also be used to save calibration system configurations for several different test points by setting up the system at the desired point, inserting a disk dedicated for power/energy calibrations at this test point and "STORE"-ing the system state in the "TEMP"-orary state file. In this way, system setup data as well as error data for specific test points may be collected on separate disks (see sec. 3.4.1 below for information concerning calibration error data storage).

3.4 Generator Characterization Subprograms

The subprograms in this section are ones which aid in characterizing the digital generator. Once the correction factors are determined, these routines may also be used to fine tune the corrections or to adjust the correction for a different operating point. The discussion presented here deals mainly with how the routines function and with gaining an understanding of how to use them. Section 5. discusses in detail how an actual characterization of the generator is carried out using these routines.

3.4.1 Calibrate

This routine employs a rotating menu system that gathers relevant input parameters for use in the subprogram, Measure. To enter this subprogram from the main menu, press the "MORE" (k0) and then the "CALIBRATE" (k5) function keys. From this point, there should be three function key options, "SELECT>OPTION," "SECOND PAGE" or "EXIT." In addition, the user may move around the menu by using the keyboard knob.

The menu used in Calibrate contains two levels. The first page, or primary menu, contains the choices between which generator characteristics are to be measured. Pressing the "SECOND PAGE" function key (k9) will result in the relevant Measure input parameters to be displayed in a similar (secondary) menu. These parameters relate to the row at which the arrow was pointing prior to entering the secondary menu by pressing the "SECOND PAGE" function key. In order to verify this, the top line of the screen in the secondary menu should be the same as the selected row in the primary menu. To return to the primary menu, press the "FIRST PAGE" key (k9) that has replaced the "SECOND PAGE" key.

The various input parameters for both the Calibrate and Measure subprograms are stored in string, integer, and real array structures, which are named Calib_val$(12,4)[30], Cal_int_vals(10,5), and Cal_r_vals(10,6), respectively. These values are stored in the ASCII file "CAL_VALS" and are loaded during program initialization. The "CAL_VALS" file may be created by running the program Calval_config in the PROG file "C_VAL_CON."

It is sometimes necessary to alter the values stored in the above data structures. To do this, the user must be in the secondary menu. Once there, the keyboard knob is used to point to the desired parameter, which will be highlighted in blue, if the monitor is color, or white, if the monitor is monochromatic. The new value may then be "ENTER"ed through the keyboard. If the value is a valid one,it should appear in the red (color monitor) or white (monochrome monitor) field after hitting the "ENTER" key.

The measurement cycle must be started in the primary menu, so to get there, use the "FIRST PAGE" key and then the "SELECT>OPTION" (k0) key. At this point Calibrate will turn control over to the Measure subprogram. If a data disk with the proper filename is not present in drive ":INTERNAL," the subprogram Measure will pause and wait for either the file to be created or the disk to be put in the drive. To find out the filename, press "filename$" and "EXECUTE." If at any time a filename other than the default name is desired, a different one may be set by performing the following steps:

1. Leave the ":INTERNAL" drive empty before using the "SELECT>OPTION" key.
2. Wait for Measure to pause.
3. "EXECUTE" a command like "Filename$=xxx."
4. Reinsert the disk with the filename on it.
5. "CONTINUE" the program.

3.4.2 Compliance

This subprogram is used to add an extra gain term to the variable channel waveform in order to correct for transconductance amplifier gain changes when operating at different output compliance voltages. Compliance first calls Measure to measure the ac compliance voltage across the output of the transconductance amplifier. This value is them used in a linear correction equation involving C_slope and C_intercept to evaluate C_ppm, which is then added (in ppm of reading) to all subsequent variable channel amplitude DAC settings. The procedure for characterizing the generator for compliance related amplitude errors is given in section 5.3.

To enter Compliance from the main menu, press the "AMP.", "VAR_WAVE" and "COMPLIANCE" softkeys in that order. When entering Compliance, there are only two choices, to "EXIT" or to "READ COMPLIANC." Before enabling the "±10 PPM" or "±1 PPM" function keys, the current amplifier's compliance voltage must first be read. After that, these keys may be used to increase or decrease the intercept of the compliance correction curve, C_intercept.

To ensure the proper variable channel gain correction is present, Compliance should be used whenever the burden on the transconductance amplifier changes.

3.4.3 Measure

Subprogram Measure is used to take automated measurements using either a Hamburger K2004, HP 3456A, HP 3457, or FLUKE 8506A. Measure may be used to take either single data points returned in the input parameter, Value, or it can take groups of data in which some generator parameter (usually amplitude) has been incremented throughout a range specified in the subprogram, Calibrate.

If the variable Row > 0 upon entry to Measure, the routine uses the contents of Calib_val$(12,4)[30], Cal_int_vals(10,5), and Cal_r_vals(10,6) assigned in Calibrate (see discussion above) to control its remaining execution path. The various fields of these data structures are assigned in the following way:

Calib_val$(12,4)[30]:

(x,1) = Type of Calibration	(1,4) = "Reset_delay"	(7,4) = "Interval"
(x,2) = Device$	(2,4) = "Num_of_reps"	(8,4) = "Limit"
(x,3) = Filename$	(3,4) = "Ac_or_dc"	(9,4) = "Percent_of_fs"
	(4,4) = "Gain"	(10,4) = "P_of_fs_limit"
	(5,4) = "Settle_time"	(11,4) = "Enable"
	(6,4) = "Full_scale"	(12,4) = "Device$"

Cal_int_vals(10,5):
- (x,1) = Reset_delay
- (x,2) = Num_of_reps
- (x,3) = Ac_or_dc
- (x,4) = Gain
- (x,5) = Settle_time

Cal_r_vals(10,6):
- (x,1) = Full_scale
- (x,2) = Interval
- (x,3) = Limit
- (x,4) = Percent_of_fs
- (x,5) = P_of_fs_limit
- (x,6) = Enable

In the above list, the expression "(x,x)," where "x" stands for any allowable array index or the index given, represents the two-dimensional index for the listed data structures. For an explanation of these variables, see the table of MET_6 variables, APPENDIX A.

When subprogram Measure is called with variable Row <= 0, the routine takes only a single measurement, returning the value read in Value. The conventions are:

On Entry:

Row = -1 :	Take one reading without initializing meter (HP 3456A).
Row = 0 :	Initialize the meter (HP 3456A) to read dc voltage and take one reading.
Row = >0 :	Use the data structures Calib_val$, Cal_int_vals, and Cal_r_vals to control subprogram execution.

3.4.4 Offset_zero

This is a simple, fast subprogram that employs recursion to zero the dc offset of either the reference or variable waveform channel. The code for this routine is very simple, but because it calls Measure, the output of the channel to be zeroed must be connected to the HP 3456A input. IMPORTANT: For the variable channel waveform, the inverting output monitor of the transconductance amplifier must be used or Offset_zero won't behave properly. This routine is called by Measure when Row = 3 or Row = 6 ("V(I)-AMP OFFSET LINEARITY") in the Calibrate subprogram.

Offset_zero may also be reached from the main menu by pressing the "AMP.", "REF_AMP" or "VAR_AMP", "OFFSET" and "ZERO DC OFFSET" function keys in that order.

3.5 Subprograms useful for Power/Energy Calibrations

The subprograms discussed here were developed primarily for the calibration of power related instrumentation using the generator alone or in a power bridge implementation. The purpose of these routines is to automate the testing procedures with the aid of the Switch Box (see sec. 2.4). As mentioned initially, these routines are not necessary for the basic operation of the generator and may be deleted if so desired.

3.5.1 Digit

Because of its relative complexity, this subprogram has been separated into two parts, Digit (the I/O portion) and Body (the measurement and error calculation portion).

To enter Digit, press the "DVM" (k1) function key. The screen is arranged into separate fields containing values that control both the execution flow and error calculations present in Body. Each column contains the complete set of variables required by Body to test the Power/Energy instrument at a single test point. Each row has the following meaning:

1. Serial #	When present, the name or number in this field serves two purposes: (1) to enable the channel for testing, and (2) to identify both the column on the screen and the test results printed on the printer.
2. Active Channels	After entering the meter's name or serial number in row 1, the field on this row should fill with "*****", which indicates that the corresponding switch box channel is to be included in the test.
3.& 4. Meas status	These rows indicate which type of tests are to be performed. The choices are "VARS," "WATTS," "VAR HRS" and "WATT HRS."
5. Nominal Voltage	The nominal voltage across a 1 kΩ standard resistor corresponding to the nominal 120 V rms, 5 A rms, unity power factor output current/voltage of the meter under test.
6. Nominal Frequency	The nominal pulse output of the meter under test at 120 V rms, 5 A rms, unity power factor.
7. Voltage Range	The meter's voltage (V) range setting.
8. Current Range	The meter's current (I) range setting.

The contents of these fields are stored in the arrays Meas_str$ and Meas_vals in the following way:

Meas_str$(1:8,1:6,0:2)[8]:

(1,I,0) = Serial number	(4,I,1) = "WATT HRS"
(2,I,0) = ""	(5,I,0) = "1.200000"
(2,I,1) = " **** "	(6,I,0) = "277.7777"
(3,I,0) = " VARS "	(7,I,0) = " 120.0 "
(3,I,1) = " WATTS "	(8,I,0) = " 5.0 "
(4,I,0) = " VAR HRS"	

Meas_vals(1:9,1:6):

(1,x) = Volt_nom	(6,x) = Dvm_avg
(2,x) = Count_nom	(7,x) = Count_avg
(3,x) = Diff_volt	(8,x) = Ppm_volt
(4,x) = Diff_count	(9,x) = Ppm_count
(5,x) = Scale	

For an explanation of the Meas_vals values, see the table of MET_6 variables, APPENDIX A.

The INTEGER array Scr_mat(1:8,1:6) serves as storage for indexes to the third dimension of the Meas_val$ array. These indexes are then utilized by Digit to keep track of what to print in the chosen field and by Body for controlling its execution cycle.

To perform a test using Digit, the various parameters must be set up on the screen. First, enter the serial number of the meter under test (MUT) in row 1 of the column on the screen corresponding to the desired Switch Box channel. Move to this field by using the knob or arrow keys. When using the knob, the highlighted field will move in the direction of the last direction key hit. Once in the appropriate field the channel will not become active until the blinking serial number field is "ENTER"ed. At this point the field should stop blinking and a string of asterisks should appear below the serial number. Continue down the column until all parameters are set appropriately for the specific MUT. When in rows 3 and 4, the values in these fields may be changed by using the "[]" (k0) function key. Repeat this procedure for the remaining meter columns.

Column 6 on the screen is intended for the use of non-standard meters that have voltage rather than current outputs. When performing a power/energy measurement with the MUT's information entered in this column on the screen, Digit reads the dc output of the meter, not the value across the standard resistor.

The "DATA STORAGE=" field indicates whether or not the error information printed on the printer is also to be stored on disk (see the "STORE DATA" (k9) softkey explanation below).

Once that all the screen parameters have been set up, push the "MORE" (k9) function key. The behavior of the remaining keys is as follows:

"PRINT MESSAGE" (k1): This routine will print the desired message on the measurement data printout. The message is saved in the array Message$(1:5)[85] while inside Digit. To clear the displayed line, either "BACK SPACE" it off or use the "CLR LN" key. "ENTER"ing a blank line causes the message to be terminated and printed within the measurement data.

"PRINT HEADING" (k2): Pressing this key causes the header information to be printed.

"RESISTOR CORR" (k3): Digit usually uses a default value of 1 kΩ for the standard resistor value. This routine allows the user to enter the exact value of an available resistor. This value will be used in subsequent error calculations for all the active channels except the non-standard meter under test (NSMUT).

"DVM CORR" (k4): Pressing this key results in two choices, "10 V RANGE COR" (k1) or "1 V RANGE COR" (k6). These represent the error (in Ppm) of the HP 3456A when reading either 10 V dc or 1 V dc. Like "RESISTOR COR," these values will be used in subsequent error calculations for all the active channels including the non-standard meter under test (NSMUT).

"DUMP DATA" (k8): Pressing this key results in a catalog of the :INTERNAL,4,1 drive to be displayed along with a prompt asking for the desired filename. The only allowable files are BDAT files; all others will generate an error. The program interprets the contents of the file as ASCII characters and dumps them on the screen, so a file containing data of type

REAL, for instance, will generate garbage on the screen. This key is useful for viewing files containing calibration error data; it will only work, however, when the "DATA STORAGE=" field is enabled.

"STORE DATA" (k9): This key enables or disables the subprogram's data storage option. When enabled, the "DATA STORAGE=" field indicates "YES" and whatever is printed on the printer by the calibration subprogram is also stored as ASCII data in a file named according to the current date. For instance, the filename generated for the first file of Oct 2, 1955 would be "Oct255_1." Successive files for the same date are generated as the data buffer becomes full, i.e., "Oct255_2." This key must be enabled in order to enable an HPIB interrupt service routine within Digit called Send_to_pc (see discussion below).

Digit may also be configured to send error files to other computers over the IEEE-488 bus. Whenever the "DATA STORAGE=" is enabled, Digit will service an SRQ interrupt using the routine "Sen_to_pc" and transfer specified files according to various parallel poll and serial poll responses. This routine is not enabled, however, whenever Digit is busy conducting a test.

3.5.2 Guildline

This subprogram is used to measure the 120 V rms signal of the voltage amplifier with the Guildline 7100A Thermal Transfer Standard. This subprogram may be reached from the main menu by using the "MORE" (k0) and "7100A" (k1) function keys. The user may select the number of readings to be performed at each of the four settings. The routine will sound a long, low-pitched beep when the setting of the Guildline requires changing. After being switched, the routine allows a settling time of 12 seconds. Once the Guildline has settled, the routine gives a noticeably higher-pitched, shorter beep that indicates it is beginning its measurements. The results of the measurement appear on the screen as ppm of reading based on a nominal value of 120 V rms.

3.5.3 Ivd

This routine is called from Status where its results are displayed on the screen and stored in the variables Rivd and Civd. These values correspond to the resistive and capacitive inductive divider settings that are required to balance the Power/Energy Bridge at the present phase angle, amplitude values, and frequency.

3.5.4 Switch

This is the switch box controller subprogram. Switch uses two input parameters: 1) Latch, the 8-bit relay data word, and 2) Lat_code, the integer value (0-2) that determines which latch is loaded on the switch box. Table 2 is a list of the most frequently used parameter settings (X = Don't Care).

Table 2. Switch Box Program Codes

INPUT	OUTPUT	Latch	Lat_code
MUT1	HP DVM	62(00111110)	1(U4)
		0(XXXXXX00)	2(U1)
		32(XX10XXXX)	0(U2 & U3)
MUT1	HP Cntr	240(1111XXXX)	2(U1)
MUT2	HP DVM	93(01011101)	1
		0(XXXXXX00)	2
		32(XX10XXXX)	0
MUT2	HP Cntr	224(0111XXXX)	2
MUT3	HP DVM	155(10011011)	1
		0(XXXXXX00)	2
		32(XX10XXXX)	0
MUT3	HP Cntr	208(1011XXXX)	2
MUT4	HP DVM	23(00010111)	1
		1(XXXXXX01)	2
		32(XX10XXXX)	0
MUT4	HP Cntr	192(0011XXXX)	2
MUT5	HP DVM	15(00001111)	1
		2(XXXXXX10)	2
		32(XX10XXXX)	0
MUT5	HP Cntr	176(1101XXXX)	2
NSMUT	HP DVM	16(X0X10XXX)	0
GENERATOR REF-OFFSET	HP DVM	0(0X00XXXX)	0
GENERATOR VAR-OFFSET	HP DVM	128(1X00XXXX)	0
GUILDLINE	HP DVM	80(X1X10XXX)	0
I-AMP	HP DVM	24(XXX11XXX)	0
V-AMP	HP DVM	0(XXXXX0XX)	0

3.6 Miscellaneous MET_6 subprograms

3.6.1 Dec_to_bin

This is a recursive subprogram that converts the input parameter Quotient into a string representation, String$, of the binary equivalent. Dec_to_bin is mainly used in the Amplitude subprogram, where it is used to display the 18-bit DAC word for amplitude. The main purpose of the routine is for the debugging of control routines. A command that performs this same task is included in some versions of HP BASIC.

3.6.2 Enable

This routine calls Switch to enable or disable the voltage amplifier, depending on the global variable, V_amp.

3.6.3 Print_time

This is a routine used to print the time and date. After using the "MORE" (k0) key in the main function key menu, Print_time is called every second to update the time and date located in the upper-right of the screen.

To set the time, use the "MORE" and "SET TIME" (k2) keys. Likewise, to set the date, use the "MORE" then "SET DATE" keys.

4. GENERATOR OUTPUT BOARD ADJUSTMENT

Each of the digital generator's two output boards contains two trim pots that must be adjusted (Pot(i) and Pot(ii) on fig. 7). The following steps are required to properly adjust each of the generator's output boards:

(i) Set the amplitude (Ref_amp or Var_amp) to zero. To do this from the main function key menu, use the "AMP." (k8), "REF_WAVE" (k1) or "VAR_WAVE" (k6), and finally "AMP" function keys. Next, exit this routine and enter Cor_dac to alter the "LINEARITY" value (see sec. 3.3.1.3). This routine alters the dc offset voltage of the amplitude DAC (U9 on fig. 7). Adjust the 8-bit MDAC until point A is as close to zero as possible. Make sure to note or store this value of R(V)_u9_offst using the "STORE VALUES" (k4) key.

(ii) Set the amplitude to full scale. To do this, enter the Amplitude subprogram and increase the voltage (current) until the "ACTUAL" binary string on the lower left of the screen reads all ones. Now, adjust Pot(i) until point A equals approximately 10 V dc.

(iii) With the amplitude (Ref_amp or Var_amp) still set to all ones, enter Wav_gen (see sec. 3.3.3.9) and set the proper memory channel to all zeroes. Now set the dc offset voltage (point B on fig. 7) of the output dac (U8 on Output Board) to zero using Cor_dac in the "OFFSET" mode (see sec. 3.3.1.3).

(iv) Finally, with the amplitude still at full scale, use Wav_gen to load all ones into the appropriate memory channel and adjust Pot(ii) until point B equals approximately 10 V dc.

5. GENERATOR CHARACTERIZATION

MET_6 provides for the characterization and correction of the generator's dc offset voltage, waveform amplitude, and the transconductance amplifier compliance voltage errors. Although the subprogram Calibrate is intended mainly for formatting plot data for a plot program, it can also be used for characterizing the generator.

The first step is to adjust the generator's output boards (see sec. 4 above). Second, before performing any runs with Calibrate, all correction terms should be zeroed. This may be accomplished by pausing the program and typing "CALL MACHINE_STATE(-1)" and then "EXECUTE."

The remaining procedures for characterizing each of the mentioned parameters are fairly similar and straightforward.

5.1 Dc Offset Voltage

Enter the Calibrate subprogram and choose either the "V-AMP OFFSET LINEARITY" or "I-AMP OFFSET LINEARITY" rows. In the secondary menu, set the parameters to appropriate values if the default ones are incorrect. Connect the output of the desired channel to the DVM and use the "SELECT>OPTION" function key. The Measure subprogram will then call Offset_zero at each test point and store the values R(V)o_intercept and the applied voltage/current in the chosen ASCII file. NOTE: Offset_zero expects the dc offset of the variable channel to be inverted, so either the output monitor of the transconductance amplifier or the inverted sense of the generator's variable channel output must be used. Finally, retrieve these values and perform a linear fit to find R(V)o_slope and R(V)o_intercept.

5.2 Amplitude

With R(V)_slope and R(V)_intercept still equal to zero, enter the Calibrate subprogram and use either the "V-AMP INTEGRAL LINEARITY" or "I-AMP INTEGRAL LINEARITY" rows. Both the error (in ppm of full scale) and the nominal voltage are stored in the chosen ASCII file. A linear fit will probably return values for R(V)_slope and R(V)_intercept that are slightly in error due to the waveform's amplitude errors result in gain correction errors. To avoid the above problem, another method of characterizing the gain is to zero the gain errors at several points, note R(V)_intercept at each point and use a linear fit to calculate the proper R(V)_slope and R(V)_intercept.

5.3 Transconductance Amplifier Compliance Voltage

Since the Calibrate subprogram does not provide a compliance correction mode, the values for C_slope and C_intercept may be found using the following method. First, zero C_slope, C_intercept and C_ppm as described above. Now, use the "COMPLIANCE" subprogram (see sec. 3.3.1) at 5 A rms to find the proper C_intercept for several different transconductance amplifier burden conditions. The resulting linear fit should provide adequate values for C_intercept and C_slope.

6. SYSTEM PERFORMANCE AND CONCLUSIONS

6.1 System Performance

The power source was originally intended to operate at 60 Hz with 120 V and 5 A sinusoidal waveforms. However, the present source is programmable between 0-240 V and 0-5 A with sinusoidal as well as arbitrary waveforms at frequencies from 0.001 Hz to 100 kHz. These figures represent the limits of amplitude and frequency. Measurements, described in this paper, were performed at 60 Hz and at amplitudes between 20-100% of full scale (FS). The amplitude and phase angle errors given below were obtained by measuring the source using a thermal wattmeter [10] and a current-comparator power-bridge [11].

The amplitudes of the reference and variable channels are changed by adjusting the dc voltages supplied to MDAC1 and MDAC2. However, the output voltages V_1 and V_2 are not ideal linear functions of these dc voltages and thus a gain adjustment is required at different amplitudes. Software gain corrections for any amplitude (based on a linear fit to a few data points) reduce this voltage dependent gain error by a factor of 5-10. Figure 16 shows the residual amplitude nonlinearity, after correction, over a 5 to 1 amplitude range where 100% of FS represents 240 V and 5 A respectively. Differential nonlinearity around 120 volts and 5 amperes is shown in figure 17. The sample points represent a one least-significant-bit change (4 ppm of FS) of the respective scaling DACs.

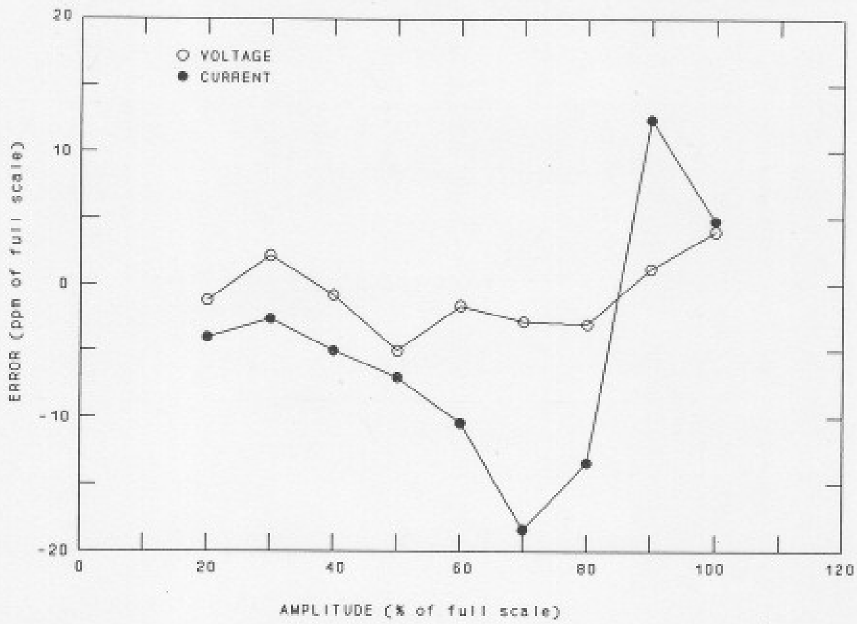

Figure 16. Residual voltage and current integral nonlinearity after gain corrections.

Phase angle accuracy depends upon the initial offset (differential phase shift between channels) and the phase linearity, which is a function of quantization errors, DAC nonlinearity, and the number of steps per period. The initial phase offset between the voltage and current waveforms was measured by the current comparator power bridge and adjusted to zero at unity power factor (0°) with an uncertainty of less than 10 microradians.

Subsequent power measurements, with constant voltage and current, between +90° and -90°, indicate that the integral phase nonlinearity in this range is less than 20 microradians.

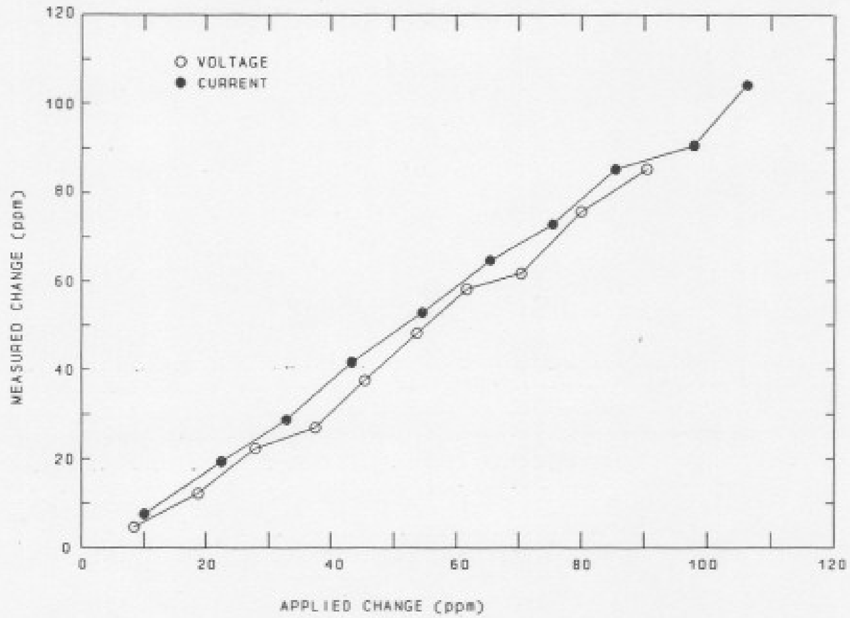

Figure 17. Voltage and current differential nonlinearity around 120 V and 5 A.

Figure 18 shows the differential phase linearity at zero power factor (90°) as measured by a time-division-multiplier (TDM) wattmeter [12]. This plot not only confirms the computer simulation predictions of 1 microradian phase resolution of the digital generator, but demonstrates the potential of TDM wattmeters for performing extremely precise measurements around zero power factor as well.

Once the three parameters (voltage, current, and phase angle) have been adjusted and corrected, the major concern becomes stability. Measurements at 120 V, 5 A over a three week period are given in figure 19. The precision of these measurements was approximately 10 ppm in amplitude and 10 microradians in phase. The current drift of 80 ppm has been attributed to aging of the 0.1 ohm shunt in the transconductance amplifier. A simple dc calibration of this amplifier is useful in detecting the gain drift due to the shunt and the results may be applied as an additional gain correction to improve long-term current stability.

Finally, the source was evaluated over a three week period as a power calibrator. Measurements were performed at 120 V and 5 A at a number of phase angles between ±90°. The source was adjusted at the beginning of the testing period and used to calibrate a TDM wattmeter over the next 20 days without further adjustments. Measurements were also performed on the TDM wattmeter using the power bridge, and an envelope which encloses all of the differences between the source and the bridge, using the TDM wattmeter as a transfer standard, is plotted in figure 20. There is a direct correlation between these differences and the current drift from figure 19 as the data for both plots were collected during the same period. If corrected for this drift, the maximum power differences fall within a ±30 ppm band, as shown in figure 21. These figures include the short-term drift of the TDM wattmeter between the source and bridge calibrations.

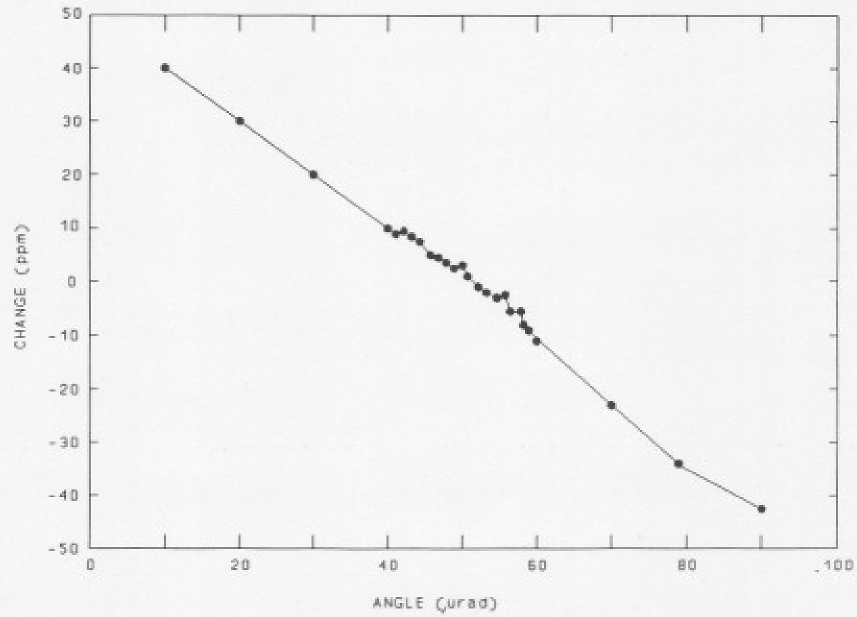

Figure 18. Phase differential nonlinearity - generator angle vs change in power indication of a TDM wattmeter at zero power factor.

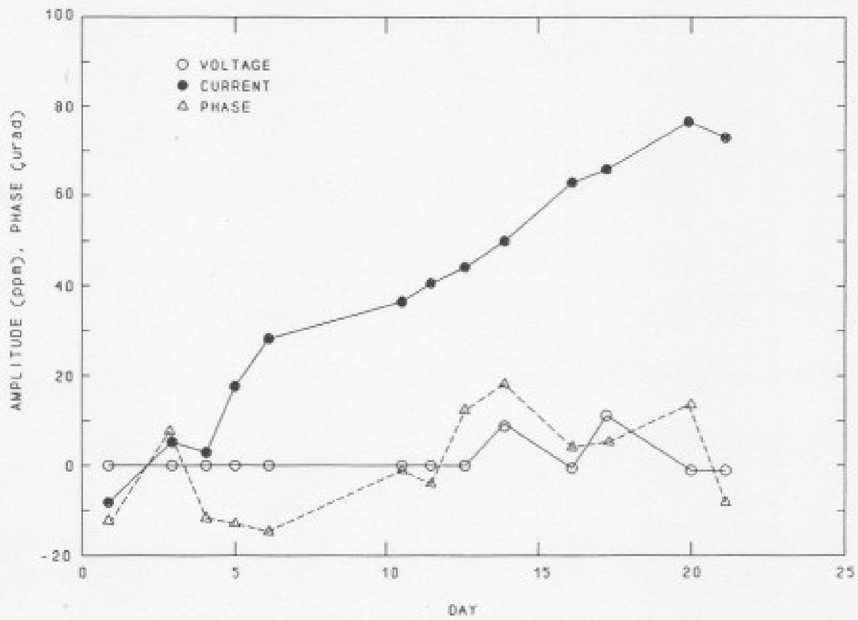

Figure 19. Long term stability of the source voltage, current, and phase angle as measured by a thermal wattmeter.

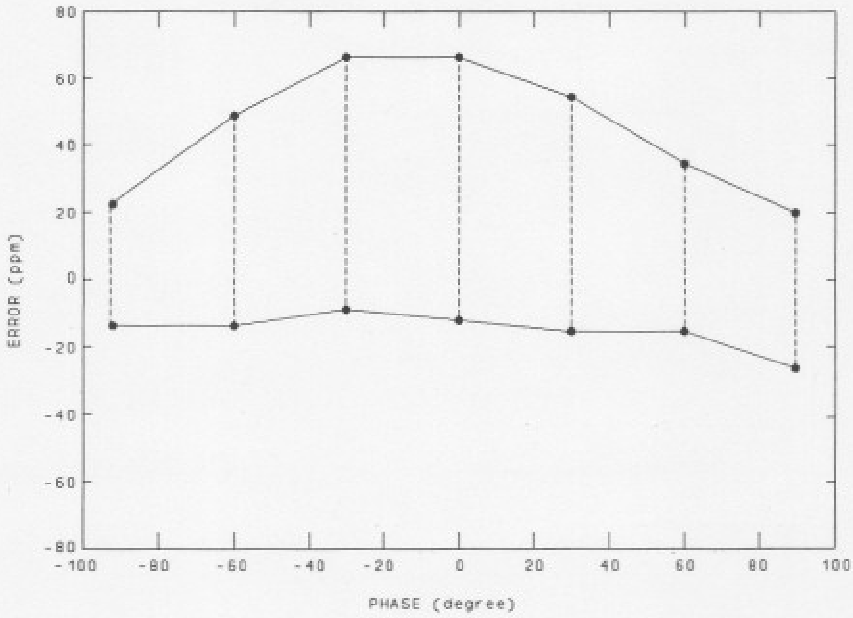

Figure 20. Maximun differences between the power calibration source and the power bridge using a TDM wattmeter as a transfer standard.

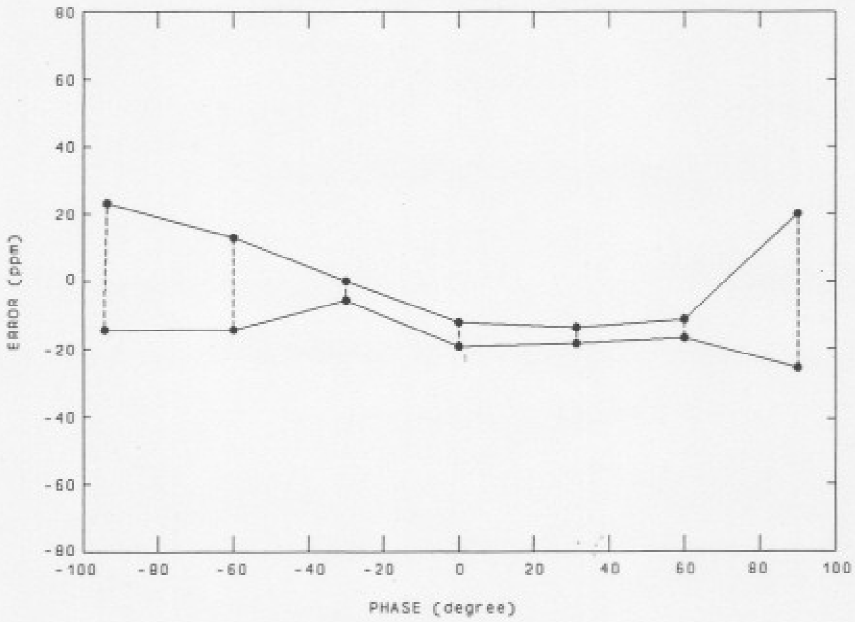

Figure 21. Maximum differences between the power calibration source and the power bridge after correcting for the current drift.

6.2 Conclusions

An accurate and precise source of synthetic power for calibrating watt/watthour and var/varhour meters at the 100 ppm level has been described. The source consists of a dual-channel digital waveform generator followed by direct-coupled high-voltage and transconductance amplifiers to provide signal levels of 60-240 V and 1-5 A at any phase angle. Control is provided by a desk-top computer and auxiliary instrumentation supports the calibration of up to six test instruments. The uncertainties of source parameters at power frequencies are:

1. voltage <30 ppm of FS.
2. current <50 ppm of FS (requires a periodic monitor of the current amplifier gain).
3. phase <20 microradians at FS voltage and current (degrades slightly at lower amplitudes).
4. Power (active and reactive) <100 ppm of FS volt-amperes.

While the source is normally operated under sinusoidal conditions, future applications will utilize its ability to synthesize arbitrary waveforms with dc components. A direct coupled system permits calibration of gain factors at dc where, in general, the resolution and accuracy of measuring instrumentation is greater. Extrapolating results at dc to power frequencies is reasonable and offers the possibility of a calculable ac source based on dc measurements. Furthermore, experience with a direct-coupled system has focused attention on the ever present ground loops that always seem to evolve in a system. In fact, monitoring and eliminating dc offsets created by ground loops has turned out to be a good technique for assuring that ground loops do not create measurement errors which might otherwise go undetected.

7. REFERENCES

[1] J. R. Kinard and L. A. Harris, "Wattmeter Calibration at Zero Power Factor Using Digitally Generated Sinewaves," IEEE Trans. on Instrum. Meas., IM-25, No. 4, Dec. 1976.

[2] R. S. Turgel and N. M. Oldham, "High-Precision Audio-Frequency Phase Calibration Standard," IEEE Trans. on Instrum. Meas., IM-27, No. 4, Dec. 1978.

[3] N. M. Oldham and R. S. Turgel, "A Power Factor Standard Using Digital Waveform Generation," IEEE Trans. on PAS, Vol. PAS-100, No. 11, Nov. 1981.

[4] C. Vastrade, R. DeVre, M. Couvreur, and R. Huybrechts, "Calibration of High Accuracy Wattmeters with a Standard Generator," IEEE Trans. on Instrum. Meas., IM-34, No. 2, June 1985.

[5] H. Ahlers, L. Rahf, H. Hartwig, and J. D. Sievert, "Programmable Digital Two-Channel Function Generator for Testing Power Meters," IEEE Trans. on Instrum. Meas., IM-34, No. 2, June 1985.

[6] N. M. Oldham, "A Power Calibration Standard Based on Digitally Synthesized Sinewaves," IEEE Trans. on PAS, Vol. PAS-104, No. 11, Nov. 1985.

[7] O. B. Laug, "A Precision Power Amplifier for Power/Energy Calibration Applications," IEEE Trans. on Instrum. and Meas. IM-36, No. 4, Dec. 1987.

[8] O. B. Laug, "A Wide-Band Transconductance Amplifier for Current Calibrations," IEEE Trans. on Instrum. Meas., IM-34, No. 4, Dec. 1985.

[9] R.S. Turgel, "A Precision Phase Angle Calibration Standard for Frequencies up to 50 kHz," IEEE Trans. Instrum. Meas., vol. IM-34, no. 4, pp. 509-516, December 1985.

[10] G. Schuster, "Thermal Instrument for Measurement of Voltage, Current, Power, and Energy at Power Frequencies," IEEE Trans. Instrum. Meas., vol. IM-29, no. 3, Sept. 1980.

[11] N. M. Oldham and O. Petersons, "Calibration of Standard Wattmeters using a Capacitance Bridge and a Digital Generator," IEEE Trans. Instrum. Meas., vol. IM-34, no. 4, Dec. 1985.

[12] P. Miljanic, B. Stojanovic, and P. Bosnjakovic, "The Development of a High Precision Time-Division Power Meter," in CPEM 84 Dig., Proc. Conf. Precision Electromagnetic Meas. (Delft, the Netherlands), 1984, pp. 67-68.

APPENDIX A: Table of Met_6 Variables

APPENDIX A: Table of Met_6 Variables

Name	Type*	Description	Found In:
@Buffer	P	Input parameter of various subprograms that holds the I/O path designator for either the Coarse_ref(*) or Coarse_var(*) INTEGER Buffers.	Out_put Init
@C_ref	P	I/O path designator for the INTEGER Buffer, Coarse_ref(*).	Main
@C_var	P	I/O path designator for the INTEGER Buffer, Coarse_var(*).	Main
@Counter	P	I/O path for the HP 5384A Freq. Counter.	Main
@Dvm	P	I/O path for the HP 3456A Dvm.	Main
@Gpio	P	I/O path for the GPIO interface.	Main
@Kbd	P	I/O path for the keyboard. Used for outputing special keys to the keyboard such as Clear_crt$.	Main
@Meter	P	I/O path for either the HP 3456A, FLUKE 8506A, or Hamburger K2004.	Measure
@Storage	P	I/O path for the ASCII file, Met6vals.	Main
A	R	The number of (1/2048) steps corresponding to the computed Di_ff.	Change_phase
A	R	Most significant 8-bits of the computed Ivd setting.	Ivd
A(2048)	R	Matrix used to compute the Rms value of the chosen waveform.	Init
A1	R	The generator's interface board address for enabling the most significant 8-bit latch of the desired Ivd.	Ivd
A1	R	Top 8-bits of the uncorrected amplitude.	Amplitude
A1	R	The truncated number of (1/2048) steps corresponding to the computed Di_ff.	Change_phase
A2	R	The fractional portion of the number of (1/2048) steps corresponding to Di_ff.	Change_phase
A2	R	Top 8-bits of the gain-corrected amplitude word. Does not include compliance correction.	Amplitude
A3	R	Top 8-bits of the fully corrected amplitude word. Includes all corrections.	Amplitude
Aborted	I	Flag indicating that too many bus errors have occurred. If Aborted=1, then Digit terminates as soon as possible.	Digit
Ac_or_dc	I	Flag indicating that the chosen meter is to measure ac (1) or dc (0) values.	Measure
Act_string$[20]	$	Holds the string representation of the present 18-bit amplitude dac setting.	Amplitude
Actual	R	The computed amplitude of the relevant channel including all corrections.	Amplitude

* B=Buffer, I=Integer, P=I/O Path, R=Real, $=String

Name	Type	Description	Found In:
Actual_resistor	R	The actual value of the resistance used in the conversion of wattmeter output current to voltage.	Main
Add_bits	R	Input parameter of the address increment corresponding to the present phase difference. Passed on to Cnt_vals.	Out_put
Address_add	R	Input parameter of the address increment corresponding to the present phase difference.	Cnt_vals
All_blanks	I	Flag set to 1 if the field being manipulated is presently all blanks.	Digit
Alpha$[2]	$	Non ASCII key sequence output to the keyboard that sets the crt into alpha mode.	Main
Angle	R	Input parameter representing the desired phase angle.	Change_phase Adj_phase
Applied	R	Local variable representing the nominal value of interest.	Measure
Av_num	I	Measurement averaging counter.	Measure
Average	R	Average of the Guildline 7100A measurements.	Guildline
B	R	2nd 8-bits of the 24-bit programmable Ivd.	Ivd
B1	R	The generator's interface board address for enabling the 2nd slice 8-bit latch of the proper Ivd.	Ivd
B1	R	2nd 8-bits of the uncorrected amplitude.	Amplitude
B2	R	2nd 8-bits of the gain-corrected amplitude dac word.	Amplitude
B3	R	2nd 8-bits of the fully corrected amplitude dac word. Includes all corrections.	Amplitude
Bage$[80]	$	Entry string for the date. (see Date_Main set:).	
Blink	I	Flag set to 1 if the field of interest is to be blinking.	Digit
Bot_row	R	Defines the last permissible menu row.	Calibrate
Bus_error	I	Flag indicating the cause of a timeout. =0: Hpib RESET timeout. =1: Dvm configuration timeout. =2: Counter config. timeout. =3: Dvm Status Register config. timeout. =4: Counter Status Register config. timeout. =5: Timeout on returning Dvm and counter to local.	Digit
C	R	3rd 8-bit slice of the 24-bit Ivd setting.	Ivd

Name	Type	Description	Found In:
C_intercept	R	Intercept of the Transconductance Amplifier's linear compliance correction curve.	Main
C_ppm	R	Ppm (of reading) correction calculated from C_slope and C_ppm that is added to variable channel amplitude dac settings.	Main
C_slope	R	Slope of the Transconductance Amplifier's linear compliance correction curve.	Main
C1	R	The generator's interface board address for enabling the 3rd 8-bit latch of the desired Ivd.	Ivd
C1	R	Last 2-bits of the uncorrected amplitude.	Amplitude
C2	R	Last 2-bits of the gain-corrected amplitude dac word. Does not include the compliance correction.	Amplitude
C3	R	Last 2-bits of the fully corrected amplitude dac word. All corrections.	Amplitude
Cal_int_vals(10,5)	I	Array set up in Calibrate and used by Measure to control its execution flow. For an explanation of its contents, see section 3.4.3.	Main
Cal_r_vals(10,6)	R	Array set up in Calibrate and used by Measure to control its execution flow. For an explanation of its contents, see section 3.4.3.	Main
Calib_val$(12,4)[30]	$	Array set up in and used by Calibrate to operate its menu. For an explanation of its contents, see section 3.4.3.	Main
Cap	R	Scale factor used to calculate Civd, the Capacitive Ivd setting.	Ivd
Cha_nge	R	Value used in the Interactive I/O routimes that represents the change, in terms of smallest divisible units, between the value at time of entry and the present value. Cha_nge is scaled so that it is easily displayed.	Adj_phase Synth_load
Change	R	The unscaled equivalent to Cha_nge, discussed above.	Adj_phase Cor_dac Synth_load
Changed	I	Flag that is set to 1 if the amplitude value of the reference channel crosses the 7.07 Vrms boundary. In which case the V-amp is either enabled or disabled.	Amplitude
Changed	I	Flag that is set to 1 if the phase resolution changes (Sel_ect changes).	Adj_phase

Name	Type	Description	Found In:
Channel$[7]	$	String containing either "VOLTAGE" or "CURRENT" depending on which output board is being addressed.	Amplitude
Check	I	If Check=1 then Init does not print the documentation title and revision message.	Init
Choice$	$	String containing either "GAIN", "OFFSET" or "LINEAR" depending on Dac_choice.	Cor_dac
Civd	R	The setting of the capacitive Ivd.	Ivd, Main
Clear_crt$[2]	$	Non ASCII key sequence that is output to the keyboard and clears the CRT.	Main
Cntr_flag	I	Flag set to 1 in the subroutine, Intr_service, after the counter requests service.	Digit
Coarse_ref(2048)	B	Buffer used to store and transfer the 2048 waveform data points to the generator's reference channel memory.	Main
Coarse_var(2048)	B	Buffer used to store and transfer the 2048 waveform data points to the generator's variable channel memory.	Main
Col	I	Used in various routines to keep track of the display column of that routine.	Calibrate Digit
Col_index	I	Index to the second dimension of Calib_val$(12,4)[30]. See section 3.4.3. Set according to the page of the menu.	Calibrate
Col_max	I	Maximum number of columns allowable in the Digit subprogram.	Main
Column	I	The present column, regardless of the menu page, that is used in the TABXY PRINT statements.	Calibrate
Column1	I	Variable to keep track of the 1st menu's Column variable.	Calibrate
Comp_volts	R	Value read by the HP 3456A (the ac compliance voltage of the Transconductance Amplifier).	Compliance
Constant	R	Constant (Usually 5.0) that is added to the desired offset or linearity voltage that is then added to the 8-bit dac word so that the value is direct reading.	Cor_dac
Cosine	R	Constant (the COS(Phaz)) that is used in Ivd to calculate Rivd, the resistive Ivd setting.	Ivd
Count	I	The number of measurements performed on the present Guildline setting.	Guildline
Counter	R	The measured pulse output of the MUT.	Digit

Name	Type	Description	Found In:
Dac_choice	I	The input parameter of Cor_dac which determines which output board function is manipulated. =0: LINEARITY =1: DC OFFSET =2: GAIN	Cor_dac
Date$	$	String used to print the current date on the upper right of the screen.	Main
Dec_pt_mask	I	Used to calculate the decimal point scale factor of the Hamburger K2004 output data.	Measure
Decimal$[2]	$	Used to calculate Dec_pt_mask.	Measure
Device$	$	Value set by Calibrate that determines which meter is to be used in the Measure run.	Measure
Di_ff	R	The difference (in degrees) between the variables Phaz and Offst. This value is then used as an input parameter to Cnt_vals (Sel_ect=1) or as a scale factor to calculate the new variable channel waveform values (Sel_ect=2).	Change_phase
Diff(*)	R	Real array containing the differences between consecutive Real(*) array values. Real(*) and Diff(*) are then used to calculate the new INTEGER buffer values.	Init
Diff_applied	R	Value used in differential nonlinearity calibrations for calculating errors.	Measure
Diff_meas	R	Used in conjunction with Diff_applied to calculate the differential errors.	Measure
Direction	I	Used by several routines to store the last arrow key that was hit. The knob will then move in this direction until a different direction key is hit.	Process_key Digit
Done	I	Used in various interrupt service routines. If Done=1 then the interrupt has been serviced.	Measure Digit
Done_once	I	Flag indicating that a certain subroutine has been branched to once already. If Done_once=1, the subprogram is usually aborted.	Digit
Dont_store	I	Flag used in error storage subroutines which indicates that no data is to be stored.	Measure
Dvm_corr_high	R	Dvm-measured value correction to compensate for Dvm errors. Used in Digit for 10 Vdc range error calculations.	Main
Dvm_corr_low	R	Dvm_measured value correction to compensate for Dvm errors. Used in Digit for 1 Vdc range error calculations.	Main

Name	Type	Description	Found In:
Dvm_flag	I	Flag set to 1 in the subroutine, Intr_ service, after the counter requests service.	Digit
En_able	R	Interface board latch setting corresponding to the desired output board. (61432 for reference, 57336 for variable)	Amplitude Cor_dac
Enable	R	Value set up in Calibrate and used in Measure in calls to Amplitude (see En_able above).	Measure
Enable	R	See En_able above.Offset_zero	
Entered	I	Flag indicating that the "ENTER" key was hit. This results in the accumulated data in Num$ to be transferred to Value$ or Value. (see Process_key code)	Process_key
Error	R	The measured error of a particular trial	Measure
Error$(1:2,1:6)[7]	$	String array used to print the formatted wattmeter errors at the end of a test point run.	Digit
Ex	I	Used in TABXY statements for displaying field data. Ex is computed in the subroutine, Ex_why_vals, within Digit.	Digit
F	R	Generator waveform output frequency. Manipulated by Synth_load.	Main
F_scale	R	Full scale of either voltage or current allowable by the 18-bit mdac (U9).	Amplitude Cor_dac
F_unit$[3]	$	Generator output frequency units. Computed by Synth_load and used in Status.	Main
F5	R	Intermediate variable used in the calculation of the synthesizer configuration codes.	Synth_load
F6	R	See F5 above.	Synth_load
F6_range	R	Intermediate variable used in the calculation of the synthesizer configuration codes.	Synth_load
F7	R	See F6_range above.	Synth_load
F8	R	See F6_range above.	Synth_load
F9	R	See F6_range above.	Synth_load
Filename$[30]	$	Defines the file to which the error data is written.	Measure
Finished	I	Exit flag used in the subroutine Print_ message.	Digit
First_time	I	If First_time=1 on entry then Offset_ zero will call Measure with Row=0, which results in the meter being initialized before being used.	Offset_zero
First_time	I	If First_time=1 then the subroutine Next_value ignores the variable, Settle_ time.	Measure

Name	Type	Description	Found In:
Found	R	Loop exit condition in the subroutine Read_meter within Measure.	Measure
Fr_eq	R	The generator's waveform output freq.	Ivd Main
Freq_range	R	Used to compute the synthesizer board's proper range setting.	Synth_load
Full_scale	R	The full scale value of the parameter being measured.	Measure
Func_buf(*)	B	Input parameter that references either the Coarse_ref(*) or Coarse_var(*) INTEGER buffers.	Out_put Init
Function$	$	Input parameter that determines which function will be loaded into the appropriate memory channel.	Main Init
Function$[9]	$	Local variable that contains a slightly altered form of the parameter, Choice$.	Cor_dac
G3	R	Intermediate variable used in the calculation of the synthesizer configuration codes.	Synth_load
G4	R	See G3 above.	Synth_load
G5	R	See G3 above.	Synth_load
G6	R	Intermediate variable used in the calculation of the memory board counter increment.	Set_inc
G7	R	See G6 above.	Set_inc
G9	R	See G3 above.	Synth_load
Gain	R	Used in the calculation of X1, X2 and X3. Gain=40 if En_able=61432 and Ref_amp > 7.07 Vrms, =0 otherwise.	Amplitude
Gain	R	Used in the calculation of errors. =40 if the measurements are taken at the output of the Voltage amplifier, =0 otherwise.	Gain Measure
Gain$	$	Used in the Interactive I/O group of subprograms for display purposes. Gain= "GAIN" always.	Main
Gar$[80]	$	Used for display purposes in the routine Date_set.	Main
Graphics$[2]	$	Non ASCII key sequence output to the keyboard that sets the crt into graphics mode.	Main
Ham_flag	I	Signals the subroutine, Int_service, that the meter of interest has sent the reading.	Measure
Highlight$[1]	$	Non ASCII key sequence output to the keyboard that sets the crt into reverse video mode.	Digit

Name	Type	Description	Found In:
Home$[2]	$	Non ASCII key sequence output to the keyboard that positions the cursor at the home position.	Main
How_much$[3]	$	String used as the Cor_dac input parameter, Mode$. See Mode$ for details.	Offset_zero
I$[12]	$	Storage for the image string used to print the applied current and power factor on the test results.	Digit
Icap	R	Intermediate variable used to calculate the Ivd settings, Civd and Rivd.	Ivd
Illegal	I	Flag indicating that rows 3 and 4 have been left in an illegal state within the subprogram Digit.	Main
Image$[25]	$	Storage for the image string used to display the phase and phase units.	Adj_phase
Inc	R	The desired counter increment. In certain instances, Inc may be set to zero.	Main
Inc_rement	R	Storage for the counter increment. Inc may be restored with Inc_rement.	Main
Index	I	Keeps track of the index to Message$ in the subroutine, Print_message.	Digit
Ins	R	Nominal scale for the current waveform where Ins=Applied/Range. Used in the subroutine Range_correct for wattmeter error calculations.	Digit
Intercept	R	Input parameter for determining various correction factors from linear correction curves.	Cor_dac Amplitude
Interval	R	The incremental change (in terms of percent of full scale) that is added to the parameter of interest at each successive test point.	Measure
Inverse	I	Flag indicating whether the field should be displayed in inverse video.	Digit
Ires	R	Intermediate variable used to calculate the Ivd settings, Civd and Rivd.	Ivd
It_rep_num	I	Counter for the number of measurement groups to be taken at each test point.	Digit
Key$	$	Used to hold the contents of the KBD buffer for processing.	Process_key
Knob_activate	I	Flag that enables the knob for changing the contents of the fields.	Digit
L2_lss	R	Least significant slice of the memory board's variable counter increment latch.	Main
L2_mss	R	Most significant slice of the memory board's variable counter increment latch.	Main
Lat_code	R	Enable code for the Switch Box's various relay latches.	Switch

Name	Type	Description	Found In:
Lat_u6	R	Input parameter which contains the last 6-bits of the latch U6 on the output board. Intended to keep track of gain/ linearity mdac settings.	Cor_dac Amplitude
Lat2	R	Used for combining L2_mss and L2_lss into one word and restoring latch 2 on memory board.	Cnt_vals
Lat3	R	Used for computing and loading the new memory board counter increment.	Cnt_vals
Latch	I	8-bit relay data word. 0<=Latch<=255	Switch
Lead_or_lag$[2]	$	Used for error data printing purposes and specifies the phase relation of the two waveforms.	Digit
Limit	R	The limit (in Ppm) of the allowable errors that will be accepted and processed.	Measure
Linear$	$	="LINEAR" always. Used for display purposes in the Cor_dac subprogram.	Main
Load_code	R	Input parameter of various routines which is used ultimately by Out_put to decide which memory channel(s) to load. = 9 : Load channel 1 (ref) only. = 6 : Load channel 2 (var) only. =12 : Load both channels.	Adj_phase Change_phase Init, Out_put
Meas_str$(1:8,1:6,0:2)	$	String values initialized in the mainMain, program and used by Digit to display the various wattmeter testing parameters. See section 3.5.1 for contents.	Digit
Meas_vals(1:9,1:6)	R	Storage for specific error computation values used for a single test point. See section 3.5.4 for contents.	Digit
Message	R	Flag to avoid incorrect keystroke acceptance. For a list of meanings, see the code.	Process_key
Message$(1:5)[85]	$	Used in the Print_message subroutine to store the printout message data.	Digit
Mode$	$	Input parameter which determines the initial behavior of the Interactive I/O Routines. See the code for further explanation.	Adj_phase Amplitude Cor_dac
New_line	R	Flag used in the Print_message routine to indicate the starting of a new line.	Digit
New_range	I	Used in the subroutine Check_range to insure that the applied value is not out of the Hamburger's input range.	Measure
Next_col	I	Used in the subroutine Next_field to indicate the column of the next field.	Digit

Name	Type	Description	Found In:
Next_quotient	R	Exit condition for the subprogram. When Next_quotient=0, the subprogram terminates.	Dec_to_bin
Next_row	I	Used in the subroutine Next_field to indicate the row of the next field.	Digit
No_cntr	I	Flag to indicate whether or not a counter reading is to be taken for the specific meter.	Digit
No_dvm	I	Flag to indicate whether or not a dvm reading is to be taken for the specific meter.	Digit
Num	I	Count specifier of the number of readings to be taken of each MUT in a measurement group.	Digit
Num$[160]	$	Used by any subprogram which calls Process_key. Num$ is the storage for the keyboard input processed by Process_key.	Main
Num_ber	R	Input parameter for Process_key in the subroutine Key, which processes keyboard input for the number of measurement iterations and groups.	Digit
Num_of_readings	I	The number of readings to be taken at each Guildline setting.	Guildline
Num_of_reps	I	The number of readings to be averaged for each test point.	Measure
Number	R	Storage for each dvm reading.	Digit
Number	R	The value in the subroutine D_to_b which is converted into a binary string.	Main
Offset	R	Input parameter for the call to Measure. It represents the dc offset of the channel of interest.	Offset_zero
Offset$	$	="OFFSET" always. Used for display purposes in the Cor_dac subprogram.	Main
Offst	R	The phase (in degrees) of the reference channel waveform.	Main
Old_range	I	Used in the subroutine Check_range to insure that the applied value is not out of the Hamburger's input range.	Measure
Old_value	R	Used in the differential non linearity calibrations for calculating the differential reading values.	Measure
Omega	R	Intermediate variable used to calculate the Ivd settings, Civd and Rivd.	Ivd
Only_1st_time	I	Used in the subroutine Key for processing the number of measurement iterations and groups.	Digit
Output_down	R	Flag used to inhibit the V-amp from being enabled at unnecessary times.	Main, Enable

Name	Type	Description	Found In:
P_factor	R	Storage for the present applied power power factor. Used for printing purposes.	Digit
P_of_fs_limit	R	The limit (in terms of full scale) of the incremented calibration value.	Measure
Page	I	Menu page indicator.	Calibrate
Page1_row	I	Storage for the row that the first page menu was left on.	Calibrate
Percent_of_fs	R	Multiplier (in terms of full scale) used to determine the applied calibration value.	Measure
Phase$	$	="PHASE" always. Used for display purposes in the Adj_phase subprogram.	Main
Phaz	R	The phase (in degrees) of the variable channel waveform.	Ivd Main
Ppm	R	The error (in Ppm of 120Vrms) of the reference channel waveform.	Guildline
Ppm	R	The correction (in Ppm of reading) to be added to the computed value.	Cor_dac Amplitude
Ppm	R	The error (in Ppm of full scale or reading, depending on the row) of the measured wattmeter output.	Digit
Quotient	R	Input parameter of the recursive routine which is divided by 2 until a quotient of zero is found.	Dec_to_bin
R	R	Intermediate variable used to calculate the Ivd settings, Civd and Rivd.	Ivd
R_func$[8]	$	String used for display purposes which identifies the present reference channel waveform type.	Main
R_intercept	R	Intercept of the reference channel's linear gain correction curve.	Main
R_rms	R	The computed Rms value of a 1Vp reference channel waveform (0<=R_rms<=1).	Main
R_slope	R	Slope of the reference channel's linear gain correction curve.	Main
R_u9_offst	R	The dc offset setting for the reference channel's amplitude dac (U9 on artwork). This value is set in Cor_dac's "LINEARITY" mode during output board adjustment and should not be changed between adjustments.	Main
R_unit$[3]	$	String used for display purposes which identifies the present reference channel amplitude units (V, mV, etc).	Main
Rad	R	Intermediate variable used to calculate the Ivd settings, Civd and Rivd.	Ivd
Range	R	Value used for display purposes to determine the proper unit string for the parameter being manipulated.	Adj_phase Amplitude Cor_dac

Name	Type	Description	Found In:
Range_changed	I	Flag indicating that at least one range field in the Digit display has been changed, which signals the subprogram, Next_col (within Body), to restore the proper ranges into the Meas_str$(7,X,0) locations.	Digit Main
Rate	R	Used by the interactive I/O subprograms to determine the rate at which the parameter of interest is changed by the knob, function keys or keyboard input.	Amplitude Compliance Cor_dac
Reading$[10]	$	Hamburger K2004 input variable. Since the Hamburger outputs BCD string data, a string input variable must be used. This string is then parsed to extract the numeric and decimal point information and transferred to Value.	Meas
Real(*)	R	Init input parameter that corresponds to Real_ref(*) or Real_var(*) depending on which channel is to be loaded.	Init
Real_ref(2048)	R	Contains the 2048 REAL function values which are then combined with the values in Ref_diff(*) and loaded into the INTEGER buffer Coarse_ref(2048).	Main
Real_var(2048)	R	Contains the 2048 REAL function values which are then combined with the values in Var_diff(*) and loaded into the INTEGER buffer Coarse_var(2048).	Main
Ref$	$	="REFERENCE" always. Used for display purposes.	Amplitude Cor_dac Main
Ref_amp	R	Contains the rms value of the reference channel waveform amplitude.	Main
Ref_c1	R	Global storage for the last 2-bits of the reference channel amplitude dac setting (U9 on Output Board).	Amplitude Cor_dac Main
Ref_diff(2048)	R	See Real_ref(2048) above.	Main
Ref_gain	R	Ppm of reading gain correction calculated by both Amplitude and Cor_dac using Ref_amp, R_slope and R_intercept.	Main
Remainder	R	Intermediate variable used to determine the equivalent binary string representation of the input parameter, Quotient.	Dec_to_bin
Reset_delay	I	ON DELAY parameter used to reset the Hamburger K2004's interface box.	Meas
Resistor_corr	R	Used in the subroutine Dvm_read to correct the dvm reading of the wattmeter's analog output across the standard resistor.	Digit
Rivd	R	The setting of the resistive Ivd.	Ivd, Main

Name	Type	Description	Found In:
Rms_array(2048)	R	Used in the subroutine Step_num to calculate the rms amplitude value of the chosen waveform from discrete steps.	Main
Rms_volts	R	Assigned to either R_rms or V_rms depending on En_able. See R_rms or V_rms for an explanation of its use.	Amplitude Cor_dac
Ro_intercept	R	Intercept of the reference channel's linear dc offset correction curve.	Main
Ro_slope	R	Slope of the reference channel's linear dc offset correction curve.	Main
Row	I	Used in various routines to keep track of the display row of that routine. Also, Row is used to reference various parameter arrays.	Calibrate Digit Measure
Row_index	I	Index to the first dimension of Calib_val$(12,4)[30]. See section 3.4.3. Set according to the page of the menu.	Calibrate
Row_max	I	Maximum number of columns allowable in the Digit subprogram.	Main
Scale	R	The variable Change is scaled by this value according to Sel_ect. See Change and Sel_ect for details.	Adj_phase
Scale$[12]	$	String identifying the value of scale, above. Scale$="(PARTS/2048)" or Scale$=" (RAD) " depending on the value of Sel_ect.	Adj_phase
Scale_factor	R	Used in the subroutine Error_process to scale the nominal value of what is being measured according to the location of the measurement point, which is expected to be either the input or output of the amplifier.	Measure
Scr_mat(1:8,1:6)	I	Array used to keep track of the indexes to the third dimension of Meas_str$.	Main
Sec	I	Used by the subroutine Time_left to display the time remaining until completion of the Init subprogram.	Init
Sel_ect	I	Flag indicating the phase resolution mode setting. For fine resolution, Sel_ect=2, and for coarse, Sel_ect=1.	Main
Sel_ect$[6]	$	This display string = "FINE " or "COARSE" depending on the value of Sel_ect.	Adj_phase
Ser_pol	I	Used in various measurement routines to store an interface serial poll in the event of a device service request interrupt.	Digit Measure

Name	Type	Description	Found In:
Setting_number	I	Counter (1-4) used for display and calculation purposes that corresponds to the number of Guildline settings.	Guildline
Settle_time	I	Guildline TE settling time after a setting change.	Guildline
Settle_time	I	Settling time of a new applied value.	Measure
Settled	I	Flag indicating that the Guildline TE has settled.	Guildline
Sine	R	Intermediate variable used to calculate the Ivd settings, Civd and Rivd.	Ivd
Slope	R	Input parameter containing the slope of the linear correction curve relevant to the value being changed.	Cor_dac Amplitude
Srq	I	Serial request bit of the meter being used that is set according to the Calib_val$(12,4)[30] field, Device$.	Measure
String$	$	Accumulation for the resultant binary equivalent string.	Dec_to_bin
String$[160]	$	Input variable for the Hamburger K2004's configuration string.	Measure
Temp	R	Storage for the initial phase value upon entry to the subprogram.	Adj_phase
Temp$	$	Used to clear the keyboard buffer.	Change_phase
Temp_v	R	Storage for the initial value upon entry to the subprogram.	Amplitude Cor_dac
Time$	$	Display variable used in the subroutine Time_set.	Main
Tmp_col	I	Temporary storage for the display column	Digit
Tmp_row	I	Temporary storage for the display row.	Digit
Top_row	I	Top row of the menu.	Calibrate
True_rms	R	Calculated using matrix operations. See R_rms or V_rms for details.	Init
U1	I	Used by Switch to decide which Switch Box latch to enable.	Main
U2_u3I		See U1 above.	Main
U4I		See U1 above.	Main
Unit$	$	Unit display string that is set according to the scaled output value, Vol_tage or Angle.	Adj_phase Amplitude Cor_dac
V$[7]	$	Used to print the present voltage range setting on the test results printout.	Digit
V_amp	I	Flag indicating whether or not the V-amp is enabled or not.	Main
V_func(2048)	R	Intermediate REAL array used in the calculation of the new Coarse_var array.	Change_phase
V_func$[8]	$	String used for display purposes which identifies the present variable channel waveform type.	Main

Name	Type	Description	Found In:
V_intercept	R	Intercept of the variable channel's linear gain correction curve.	Main
V_rms	R	See R_rms above.	Main
V_slope	R	Slope of the variable channel's linear gain correction curve.	Main
V_u9_offst	R	The dc offset setting for the variable channel's amplitude dac (U9 on artwork). This value is set in Cor_dac's "LINEAR-ITY" mode during output board adjustment and should not be changed between adjust-ments.	Main
V_unit$[3]	$	String used for display purposes which identifies the present variable channel amplitude units (V, mV, etc).	Main
Value	R	Storage for numeric only data when an ENTER key is hit.	Cor_dac Process_key Synth_load
Value	R	Input variable for dvm error readings.	Guildline
Value	R	Value returned from a call to Measure.	Measure
Value$	$	Num$ is accumulated here until an ENTER key is hit.	Digit Process_key
Var$	$	="VARIABLE" always. Used for display purposes.	Main
Var_amp	R	Contains the rms value of the variable channel waveform amplitude.	Main
Var_c1	R	Global storage for the last 2-bits of the variable channel amplitude dac setting (U9 on Output Board).	Main
Var_diff(2048)	R	See Real_var(2048) above.	Main
Var_gain	R	Ppm of reading gain correction calcu-lated by both Amplitude and Cor_dac using Var_amp, V_slope and V_intercept.	Main
Vns	R	Nominal scale for the voltage waveform where Vns=Applied/Range. Used in the subroutine Range_correct for wattmeter error calculations.	Digit
Vo_intercept	R	Intercept of the variable channel's linear dc offset correction curve.	Main
Vo_lts	R	Scaled version of the parameter being manipulated used for display purposes. See Voltage below.	Amplitude Cor_dac
Vo_slope	R	Slope of the variable channel's linear dc offset correction curve.	Main
Vol_string$[20]	$	Holds the string representation of the present 18-bit amplitude dac setting.	Amplitude
Voltage	R	Un-scaled version of the parameter being manipulated used for display purposes. See Vo_lts above.	Amplitude Cor_dac

Name	Type	Description	Found In:
What_was_meas$[3]	$	String printed along with the errors to identify the test that was performed.	Digit
Why	I	Used in TABXY statements for displaying field data. Why is computed in the subroutine, Ex_why_vals, within Digit.	Digit
Within_cor_dac$	$	If Offset_zero is called with this string="PRINT IT", then the Print_volt subroutine within Cor_dac will be called for every iteration of Offset_zero.	Offset_zero
X	R	Used to clear the keyboard buffer.	Amplitude
X	R	Used in the subroutine Key_in to protect against superfluous BACKSPACE entries.	Digit
X1	R	Intermediate variable used in the calculation of the 18-bit amplitude dac word.	Amplitude
X2	R	See X1 above.	
X2	R	The 8-bit correction dac setting.	Cor_dac
X3	R	See X1 above.	Amplitude
Y	I	Loop counter for wattmeter measurements.	Digit
Z	R	Value used for debugging purposes to determine the actual 24-bit Ivd setting.	Ivd