HP 4062F Semiconductor Parametric Test System

Manual Supplement for System Operation

HP Part No. 04062-90380
Printed in Japan February 1995

Edition 2
E0295

Product Warranty

This Hewlett-Packard system product is warranted against defects in material and workmanship for a period of one year from date of installation. During the warranty period, HP will, at its option, either repair or replace products that prove to be defective.

Warranty service of this product will be performed at Buyer's facility at no charge within HP service travel areas. Outside HP service travel areas, warranty service will be performed at Buyer's facility only upon HP's prior agreement and Buyer shall pay HP's round trip travel expenses. In all other cases, products must be returned to a service facility designated by HP.

For products returned to HP for warranty service, Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country.

HP warrants that its software and firmware designated by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modifications or misuse, operation outside of the environment specifications for the products, or improper site preparation or maintenance.

No other warranty is expressed or implied. Hewlett-Packard specifically disclaims the implied warranties or merchantability and fitness for a particular purpose.

Exclusive Remedies

The remedies provided herein are the Buyer's sole and exclusive remedies. Hewlett-Packard shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. HP further certifies that its calibration measurements are traceable to the National Institute of Standards and Technology (NIST), to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

Printing History

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

The software revision code printed before the date indicates the version level of the software product at the time the manual or update was issued. Many product updates and fixes do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one to one correspondence between product updates and manual updates.

Edition 1	For Rev. F.06.3x, July 1994
Edition 2	February 1995

Safety Summary

The following general safety precautions must be observed during all phases of operation, service, and repair of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual may impair the protections provided by the equipment. In addition, it violates safety standards of design, manufacture, and intended use of the instrument. Hewlett-Packard Company assumes no liability for customer's failure to comply with these requirements.

Note

HP 4062F complies with INSTALLATION CATEGORY II and POLLUTION DEGREE 2 defined in IEC 1010-1.
HP 4062F is INDOOR USE product.

GROUND THE INSTRUMENT

To minimize shock hazard, the instrument chassis and cabinet must be connected to an electrical ground. The power terminal and the power cable must meet International Electrotechnical Commission (IEC) safety standards.

DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE

Do not operate the instrument in the presence of flammable gases or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.

KEEP AWAY FROM LIVE CIRCUITS

Operation personnel must not remove instrument covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. Do not replace components with power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed. To avoid injuries, always disconnect power and discharge circuits before touching them.

DO NOT SERVICE OR ADJUST ALONE

Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

DO NOT SUBSTITUTE PARTS OR MODIFY INSTRUMENT

Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the instrument. Return the instrument to a Hewlett-Packard Sales and Service Office for services and repair to ensure that safety features are maintained.

DANGEROUS PROCEDURE WARNINGS

Warnings, such as example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed.

Warning

Dangerous voltages, capable of causing death, are present in this instrument. Use extreme caution when handling, testing, and adjusting.

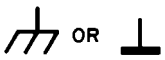
Safety Symbols

The general definitions of safety symbols used on equipment or in manuals are listed below.

Instruction manual symbol: the product will be marked with this symbol when it is necessary for the user to refer to the instruction manual in order to protect against damage to the instrument.

Indicates dangerous voltage (terminals fed from the interior by voltage exceeding 1000 volts must be so marked).

Protective conductor terminal. For protection against electrical shock in case of a fault. Used with field wiring terminals to indicate the terminal which must be connected to ground before operating equipment.



Frame or chassis terminal. A connection to the frame (chassis) of the equipment which normally includes all exposed metal structures.

Alternating current.

Direct current.

ON (Supply).

OFF (Supply).

STANDBY (Supply).

Warning 

The warning sign denotes a hazard. It calls attention to a procedure, practice, condition or the like, which, if not correctly performed or adhered to, could result in injury or death to personnel.

Caution 

The caution sign denotes a hazard. It calls attention to an operating procedure, practice, condition or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product.

Note 

The note sign denotes important information. It calls attention to a procedure, practice, condition or the like, which is essential to highlight.

Preface

HP 4062F Semiconductor Parametric Test System can perform pulse output in addition to performing the dc and capacitance measurements of typical parametric testers. So the HP 4062F system can evaluate flash memory cells in addition to measuring the dc characteristics of typical semiconductor devices.

Hewlett-Packard also supplies HP 4062C and HP 4062UX systems, which are typical parametric testers. The HP 4062F system can also perform the same functions as the HP 4062UX system, so the HP 4062F can considered to be an HP 4062UX that has the additional function of pulse output.

This manual describes the operation of the HP 4062F system, except for operations that are already described in the HP 4062UX system manuals. For information about dc and capacitance measurements, refer to the HP 4062UX system manuals. The HP 4062UX system manuals are furnished with the HP 4062F system.

This manual consists of two parts. Part 1 describes the operation of the HP 4062F system, and Part 2 describes the application programs for evaluating flash memory cells, which are included in the HP 4062F system software. The following briefly shows the structure of this manual:

- Part 1: System Operation

  Chapter 1 "Introduction of HP 4062F System"

  Provides general information about the hardware and software of the HP 4062F system. Also, this chapter covers the differences from the HP 4062UX system: switching matrix and pulse generators.

  Chapter 2 "Getting Started"

  Covers START.F program, which starts the HP 4062F software environment, and VFP.AC program, which enables you to control the pulse generators manually. You can manually force pulse outputs, if you understand this chapter.

  Chapter 3 "Measurement Programming"

  Describes programming techniques for using measurement programs to force pulses.

  Chapter 4 "Programming Reference"

  Provides an alphabetical listing and complete description of the subprograms used to force pulses.

  Chapter 5 "Diagnostics Program"

  Covers diagnostics program "DIAG.F", which verifies the system operation of the HP 4062F.

- Part 2: Application Programs

  Chapter 6 "Introduction of Application Programs"

  Introduces the application programs included in the HP 4062F system software.

  Chapter 7 "Accumulated Pulse Width Dependency Measurement"

  Describes measurement program for the accumulated pulse width dependency measurement, and how to customize the program.

  Chapter 8 "Pulse Width Dependency Measurement"

  Describes measurement program for the pulse width dependency measurement, and how to customize the program.

  Chapter 9 "Times Dependency Measurement"

  Describes measurement program for the times dependency measurement, and how to customized the program.

- Appendixes

  Appendix A "Graphic Subprograms"

  Provides an alphabetical listing and complete description of subprograms that are useful for drawing graphs of the results.

  Appendix B "Program List of Application Programs"

  Appendix C "HP 4062F System Specifications"

Other Manuals.
You should refer to the following manuals when you use the HP 4062F system:

HP 4062C/UX System Information (HP part number 04062-90311)

HP 4062C/UX System Operation (HP part number 04062-90321)

HP 4062C/UX Programming Reference (HP part number 04062-90331)

HP 4062C/UX Quick Reference (HP part number 04062-90341)

Contents

Messages

Index

Figures

Tables

1

Introduction of HP 4062F System

This chapter provides general information on the HP 4062F Semiconductor Parametric Test System.

The HP 4062F system has the same functions as the HP 4062UX system, but in addition can evaluate **flash memory cells**. In other words, the HP 4062F system has same I/V measurement and capacitance measurement functions as the HP 4062UX system, and in addition can write or erase the flash memory cell. So you can evaluate flash memory cells using the HP 4062F system. See the HP 4062C/UX manuals for the I/V measurement and capacitance measurement functions.

To write or erase the flash memory cells on a wafer or in a packaged device, the HP 4062UX system is newly equipped with high-frequency pulse sources, some hardware to pass the high-frequency pulses through, and software to control them. In this chapter, the following are described:

- HP 4062F hardware
- HP 4085F switching matrix (SWM)
- HP 8110A pulse generator
- HP 4062F system software

HP 4062F Hardware

The HP 4062F system has the function to evaluate flash memory cells in addition to the functions of the HP 4062UX system. So pulse generators are added as new system instruments to the system instruments used for the HP 4062UX system. Table 1-1 shows the system instruments for the HP 4062UX and HP 4062F systems.

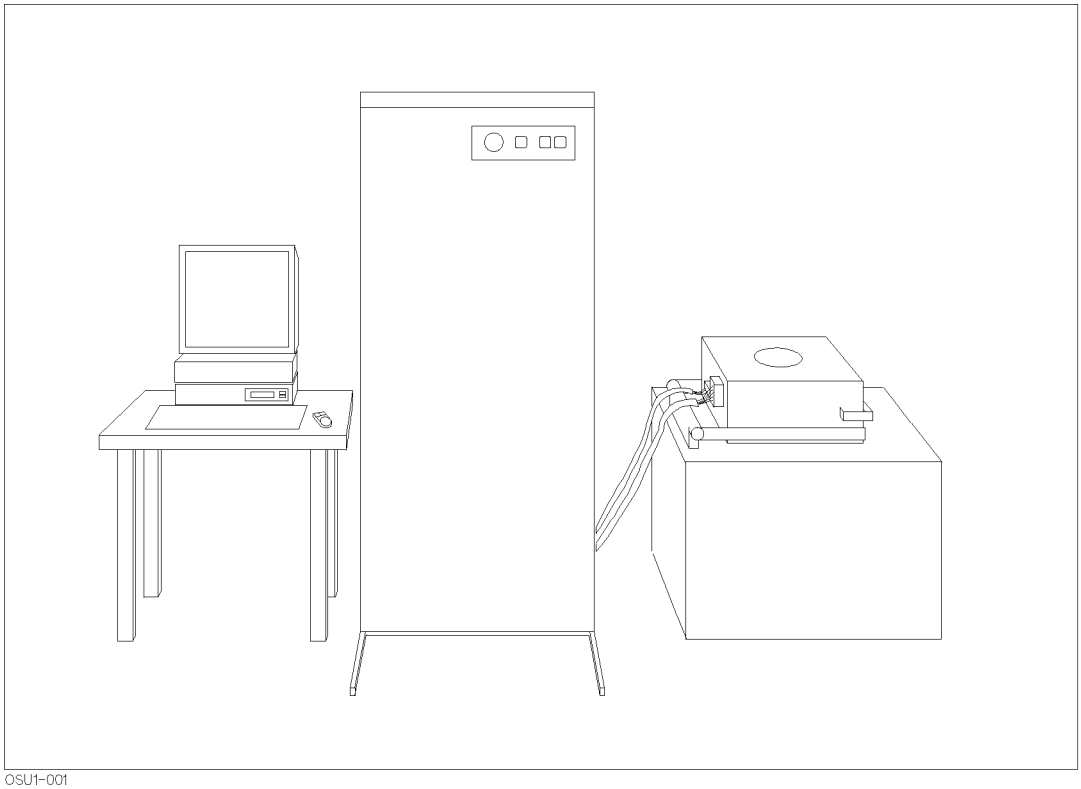
Table 1-1. System Instruments for HP 4062UX and HP 4062F Systems

Subsystem Name	HP 4062UX	HP 4062F
Switching Matrix (SWM)	HP 4085B or HP 4089A/B	HP 4085F[1]
Switching Matrix Controller (SWC)	HP 4084B	
Pulse Source Subsystem (PSS)	———	HP 8110A
DC Measurement Subsystem (DCS)	HP 4142B	
Capacitance Measurement Subsystem (CMS)	HP 4280A or HP 4284A	

1 HP 4062F system supports 48-pin switching matrix only.

The HP 4062F system uses the HP 8110A 150 MHz pulse generator (PG) as a pulse source. The HP 8110A can have one or two pulse output terminals (PGUs), and up to five HP 8110As can be installed in the HP 4062F system. So you can use 1 to 10 ports as pulse signal sources.

Figure 1-1 shows an overview of standard system configuration of the HP 4062F system. System instruments are contained in the system cabinet, and switching matrix (SWM) is used to contact with wafer prober. HP 9000 Series 300 computer is used as a system controller and connected to the system instruments and the wafer prober using HP-IB cables.



OSU1-001

Figure 1-1. Overview of Standard System Configuration of HP 4062F

System Configuration of HP 4062F

The HP 4062F system is designed for installing switching matrix (SWM) on a automatic or semiautomatic wafer prober. You can also install the SWM on a workbench to measure packaged devices.

System instruments except the SWM are installed in the system cabinet. Figure 1-2 shows the inside view of the system cabinet.

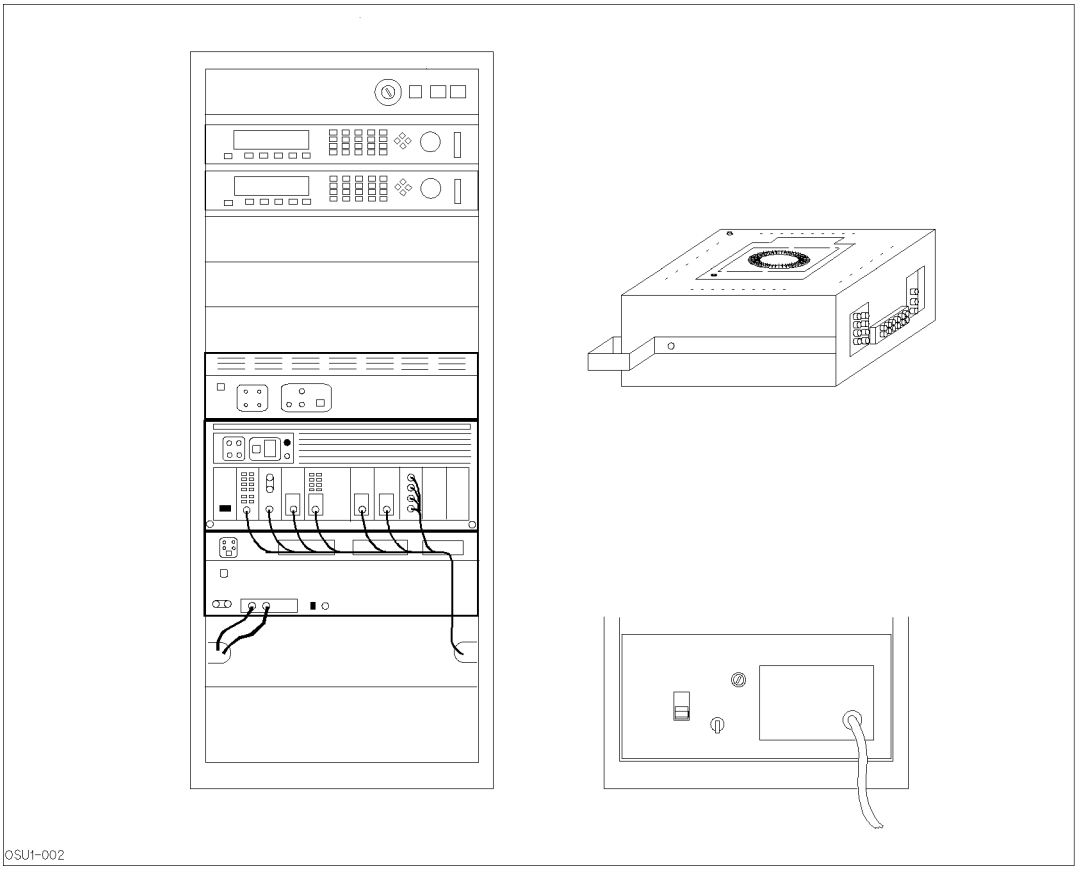


Figure 1-2. HP 4062F System Configuration

Switching Matrix	HP 4085F 48-pin Switching Matrix (SWM)
LINE Switches	Main power switch of the system cabinet. Pressing the LINE 1 push-switch applies power to the system instruments (green indicator on LINE 1 switch turns on). Pressing the LINE 0 push-switch stops the supply of power to the system instruments (green indicator turns off).
EMERGENCY OFF Switch	Pressing the emergency off switch stops the supply of power to the system instruments. This switch is a red, circular push-switch, and you must use a mechanical key to reset the locked switch after pressing the emergency off switch.
Pulse Generator	HP 8110A 150 MHz Pulse Generator. Up to five HP 8110As can be installed in the HP 4062F system.
Switching Matrix Controller	HP 4084B Switching Matrix Controller (SWC).
DC Source/Monitor	HP 4142B Modular DC Source/Monitor. This instrument is used for DC measurement subsystem.
Capacitance Meter	HP 4280A 1 MHz C Meter/C-V Plotter or HP 4284A Precision LCR Meter. This instrument is used for capacitance measurement subsystem.

Circuit Breaker (120, 220, 240 Vac)	Circuit breaker for line voltage of 120 Vac, 220 Vac, and 240 Vac. If an unexpected value of line current is used in the system cabinet, the supply of power to the system instruments stops.
Circuit Breaker with GFCI (100 Vac)	Circuit breaker for line voltage of 100 Vac, which also has a ground-fault circuit interrupter (GFCI) function. If an unexpected value of line current is used in the system cabinet, the supply of power to the system instruments stops.

HP 4085F Switching Matrix

Figure 1-3 shows the view of the HP 4085F Switching Matrix.

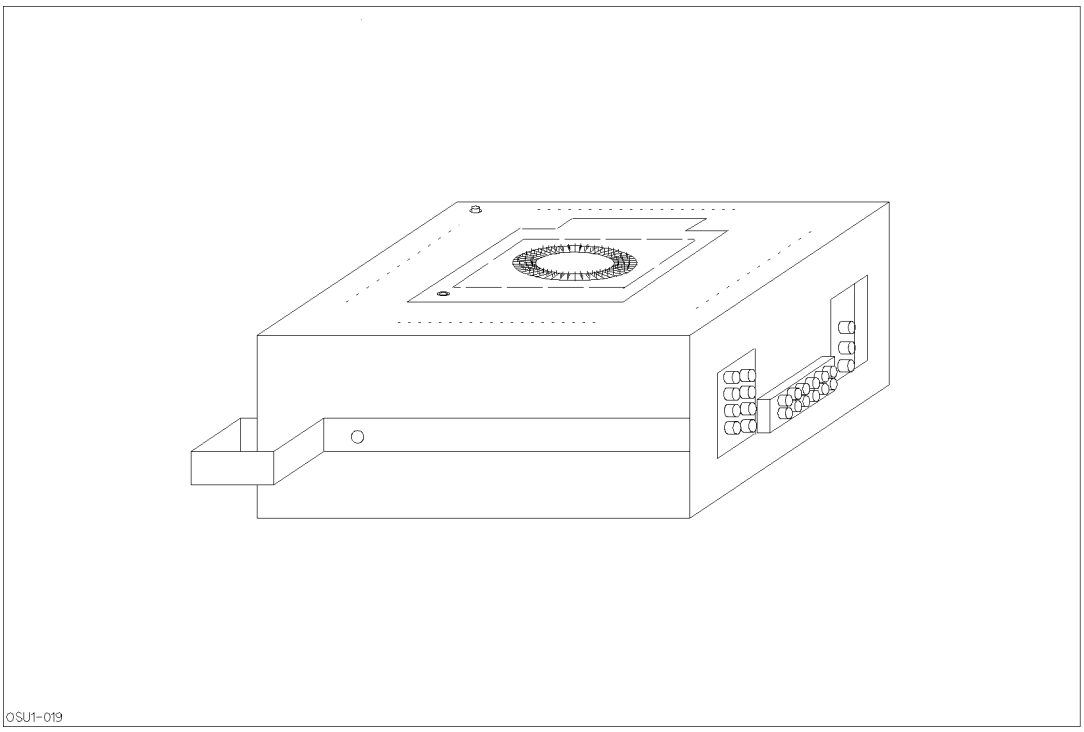


Figure 1-3. View of HP 4085F Switching Matrix

Handle — For raising and lowering the SWM when the SWM is attached to a wafer prober.

⚠

Warning

1. Use the handle to raise and lower the switching matrix. Keep hands clear of the prober when raising or lowering the switching matrix.

2. Do not allow the switching matrix to slam down against the prober.

OUTPUT ENABLE Indicator — Lights when the FIXTURE CLOSED DETECT switch or WAFER PROBER SENSE switch is closed. When lit, programmed pin-port connections are made.

FIXTURE CLOSED DETECT Switch — Detects whether the test fixture adapter's cover is open or closed.

WAFER PROBER SENSE Switch — Detects if the SWM is positioned on a wafer prober.

⚠

Warning

If either the FIXTURE CLOSED DETECT switch or WAFER PROBER SENSE switch is closed, dangerous dc voltage exceeding ±42 V may be present at the measurement pins. Severe electric shock will result if the measurement pins are touched.

Note

The FIXTURE CLOSED DETECT and WAFER PROBER SENSE switches are connected in parallel, so if either switch is closed, SWM output is enabled. If both switches are open, SWM output is disabled, thus disconnecting all SWM measurement ports from the measurement pins. Though disabled, you can still control the SWM using VFP or TIS commands. Any changes made while the SWM is disabled are automatically set when one of these switches is closed.

Measurement Pins — These spring-loaded measurement pins connect (spring pressure contact) to the contacts of a text fixture or wafer prober.

Caution

Do not touch the SWM's measurement pins with bare hands. Oil, perspiration, and dirt prevent good electrical contact and degrade measurement accuracy.

Pin Board Fail Indicators — Light in a sequential pattern during the SWM's Relay Test. If the test detects a defective pin board, the corresponding indicator flashes on and off, as does the FAIL lamp on the SWC.

AUX2, AUX3 (CMH), AUX4 (CML) Ports — Auxiliary ports (BNC connectors) for connecting the DCS's VSs and VMs, the CMS's CMH and CML, and the CMU84's CMH84 and CML84 terminals. All AUX ports can connect to the VSs and VMs, but only the AUX3 and AUX4 ports can connect to the CMH and CML, or CMH84 and CML84 terminals, respectively. AUX port connections must be established before you run the START.F program. Refer to the "START.F Program" in Chapter 2.

HF Ports — High frequency (HF) ports are designed for the pulse generators. To connect the output terminals of the pulse generator to the HF ports, the cable adapter (HP part number 04085-61066) is installed on the SWM, and HF ports on the cable adapter are connected to the output terminals of the pulse generator using the furnished BNC cables.

Note

When connecting the SWM's ports to the system instruments, use only the cables furnished with the HP 4062F system. For more information on how to connect the SWM to the system instruments, see HP 4062F *Installation Manual*.

Switching Matrix (SWM) Connection Concepts

The SWM's main function is to interconnect the HP 4062F's subsystem components and SWM's measurement pins via the SWM's measurement ports. Measurement ports are ports for connecting to the system instruments such as SMU, VS, and VM in the DCS. Measurement pins are spring-loaded pins located on the bottom panel of the SWM in a circle.

Measurement ports and pins provide the interconnection circuits in a 19×48 matrix, and make connections and disconnections between the measurement ports and measurement pins using relays. Figure 1-4 shows a simplified diagram of an SWM interconnected with its SWC, DCS, CMS, and PGs. Device under test (DUT) is usually connected to the measurement pins via test fixture or personality board and probe card, which is mounted on the bottom panel of the SWM. So, the SWM enables you to perform many types of measurements without changing actual connections at DUT.

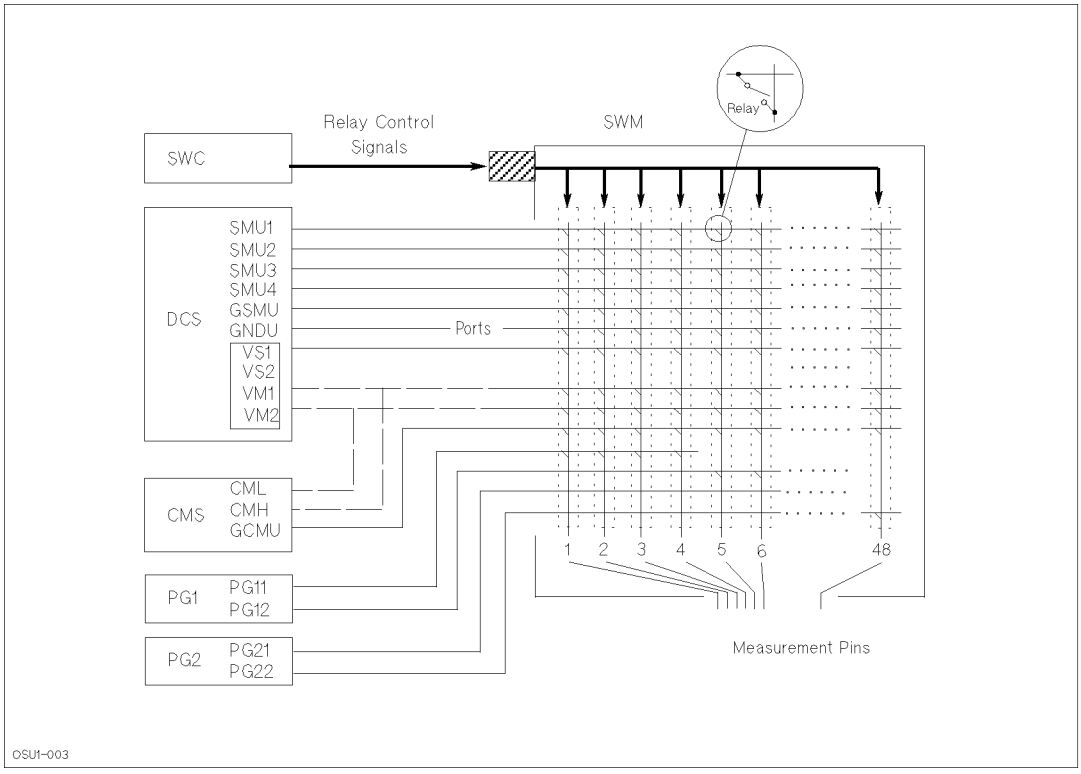


Figure 1-4. Simplified Diagram of Switching Matrix

The HP 4085F SWM has total of nineteen input/output ports: SMU1, SMU2, SMU3, SMU4, HF1, HF2, · · · , HF12, AUX2, AUX3 (CMH), AUX4 (CML). The SMU1–4, AUX2–4, and GNDU ports can be connected to measurement pins 1–48. One HF port can be connected to four measurement pins. That is, HF port connections are restricted: HF 1 port can connect to pins 1–4, HF 2 port can connect to pins 5–8, and so on.

The cable adapter also has multiplexer switches and open/close switches in addition to the twelve HF ports.

Measurement Pins

Measurement pins are circularly located on the bottom panel of the SWM. Each measurement pin, mounted on a single pin board, is actually a set of three spring-loaded pins—SENSE, FORCE, and GUARD—described as follows:

SENSE — SENSE pins function as the input terminal for Kelvin connections implemented with the SMU2, SMU3, or SMU4 ports. SENSE pins connected to the SMU1, HF1–12, AUX2–4 ports are open-terminated, and thus, have no function.

FORCE — FORCE pins function as the output terminals for Kelvin connections implemented with the SMU2, SMU3, or SMU4 ports. FORCE pins function as the input/output terminals for subsystem components connected to the SMU1, HF1–12, AUX2–4 ports.

GUARD — GUARD pins function as guard terminals for the SMU, CMH, and CML ports. They also function as common terminals for the AUX2–4 and HF1–12 ports.

Ports

You can consider the ports to be the measurement terminals of each instrument. The SWM has two types of ports: measurement ports (ports 1 to 9 and ports 41 to 52) and guard ports (ports 10 and 11).

Measurement port is for connecting to the input/output terminals of each instrument, and the measurement signal passes through it. Guard port is for connecting to the guard terminal of specific instruments (guard terminals of SMU1 and CMS). Table 1-2 lists SWM port specifics. Ports are connected to DCS, CMS, and PGU, and the meanings of signal lines depend on the ports. Figure 1-5 shows SWM pin-port connections.

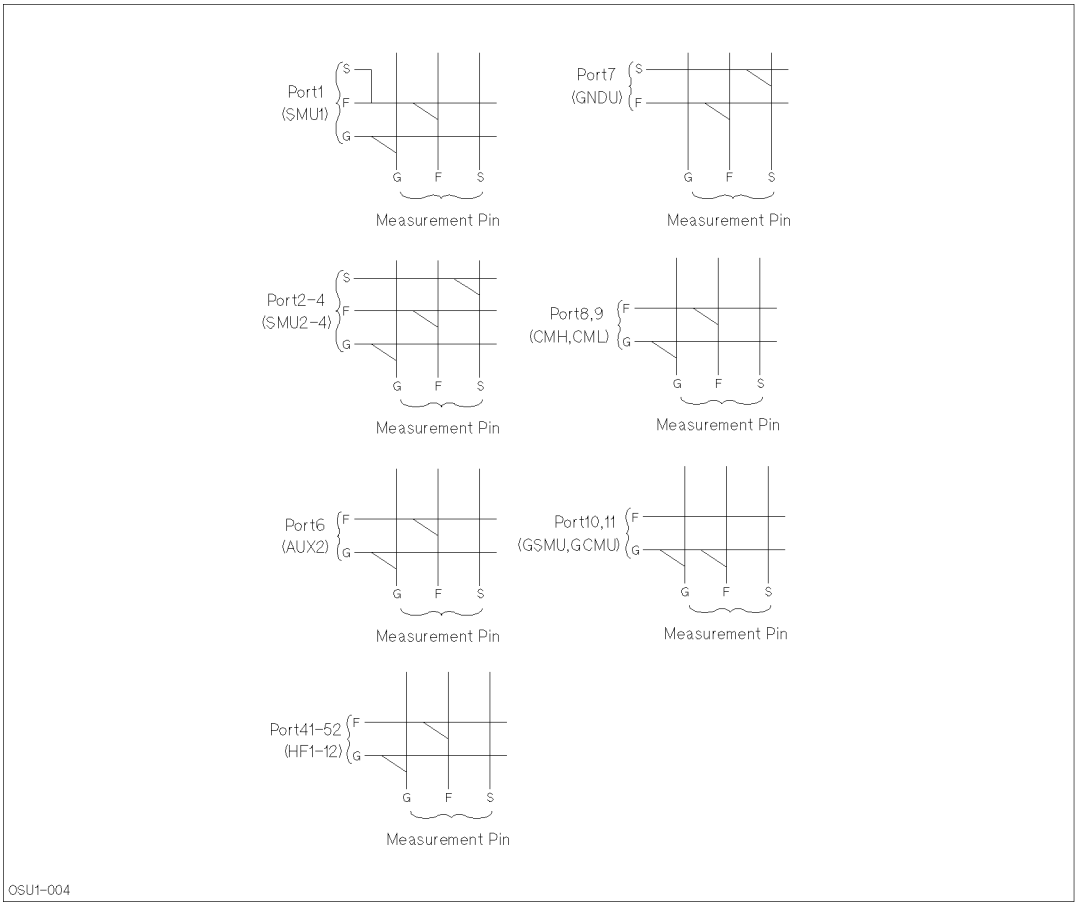


Figure 1-5. SWM Pin-Port Connections

Table 1-2. SWM Port Specifics

Port No.	Port Name	Measurement Unit/Terminal	Function
1	SMU1	SMU1 (F, G)	Low current dc measurements
2	SMU2	SMU2 (S, F, G)	High power dc measurements
3	SMU3	SMU3 (S, F, G)	General dc measurements
4	SMU4	SMU4 (S, F, G)	
5[1]	HF1 to 12	PGU	Pulse force, or auxiliary dc voltage source/voltage measurements
6	AUX2	VS or VM	Auxiliary dc voltage source/voltage measurements
7	GNDU	GNDU (S, F)	Circuit common
8	AUX3 (CMH)	VS, VM, or Hi terminal of CMS	Auxiliary dc voltage source/voltage measurements, or capacitance measurements
9	AUX4 (CML)	VS, VM, or Lo terminal of CMS	
10	GSMU	SMU1 guard terminal	Guarding for low current dc measurements
11	GCMU	CMS guard terminal	Guarding for capacitance measurements
41	HF1	PGU	Pulse forcing
42	HF2	PGU	
43	HF3	PGU	
44	HF4	PGU	
45	HF5	PGU	
46	HF6	PGU	
47	HF7	PGU	
48	HF8	PGU	
49	HF9	PGU	
50	HF10	PGU	
51	HF11	PGU	
52	HF12	PGU	

1 This port exists to keep consistency with the HP 4062C/UX system. Actually this port is assigned to port 41 to 52.

Pin Boards

The HP 4085F SWM uses HP 16320B Pin Boards. The HP 16320B is also used for HP 4085A/B SWM.

Each pin board houses a measurement pin (sense, force, and guard pins) and the switching relays for pin-port connections. One pin board corresponds to one measurement pin (force, sense, and guard). One port can be connected to multiple measurement pins. But one measurement pin can be connected to only one port.

Up to 48 pin boards can be installed in the SWM. And at least 12 pin boards must be installed in proper slots of the SWM to operate the SWM correctly. For details, see Chapter 5.

The pin board has switching relays to connect between ports and measurement pins, and has electronics circuit to control the relays. See chapter 2 of the HP 4062C/UX *System Information* manual about the functions of these relays.

Connections between HF Ports and Measurement Pins

This section describes the status of the pin board relays when an HF port is connected to a measurement pin (sense, force, and guard pins). Figure 1-6 shows the status of each relay when an HF port connects to the measurement pin.

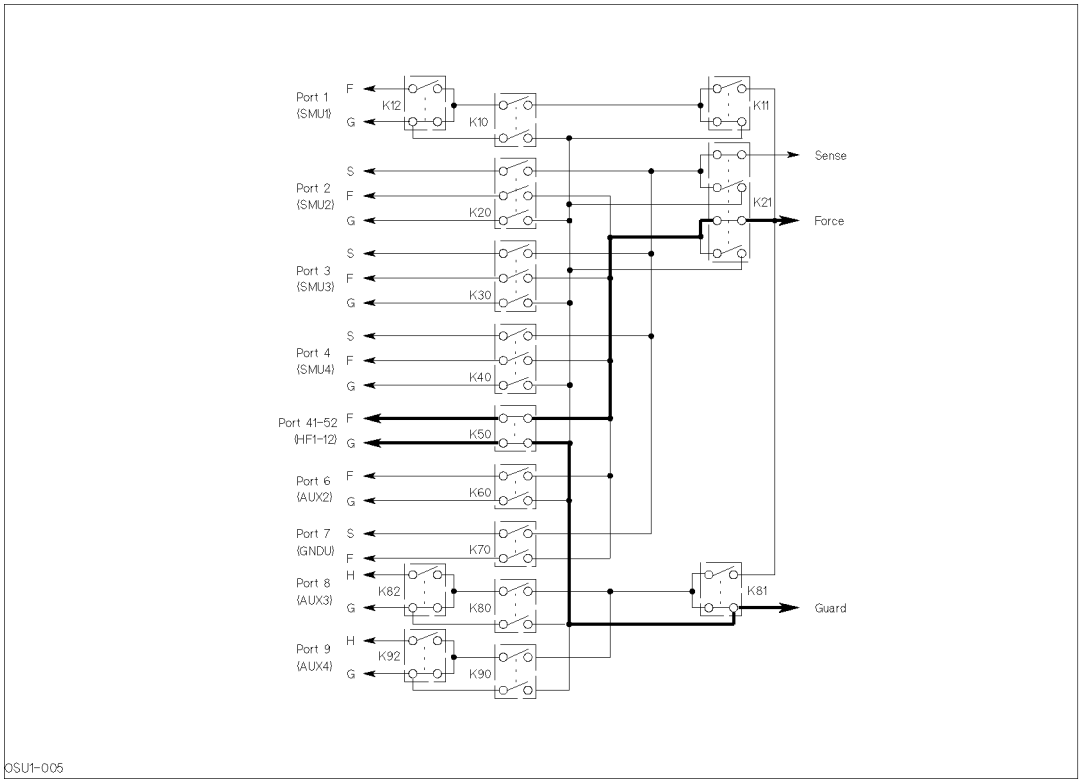


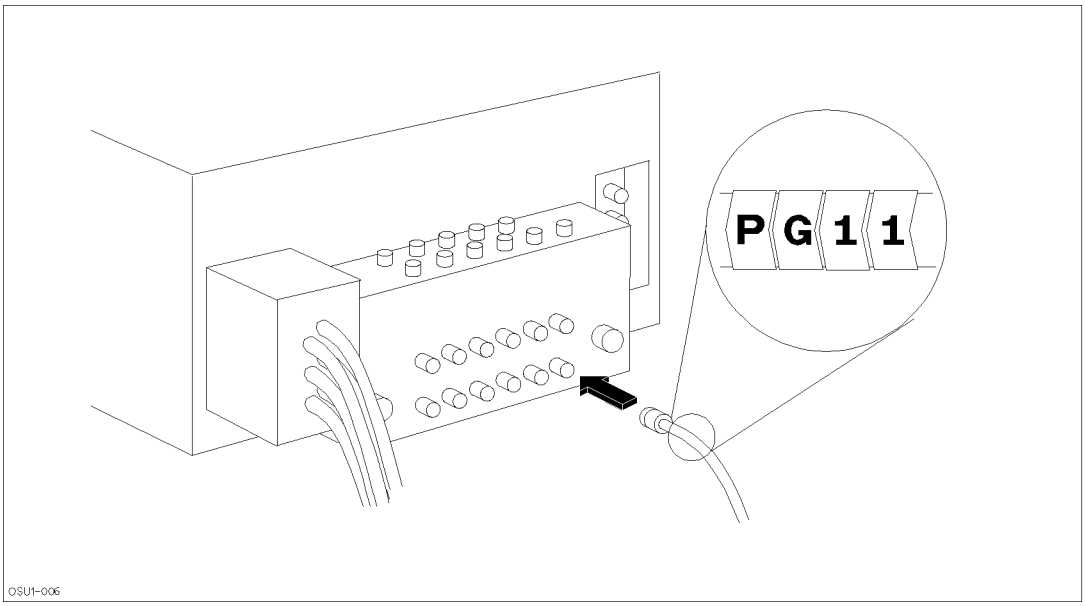
Figure 1-6. Connections between HF Port and Measurement Pins

When an HF port connects to a measurement pin, K50 and K21 are set to make (on), and signal line and common line of PG are connected to force and guard pins, respectively.

As described previously, one HF port can be connected to up to four measurement pins (HF1 to pin1–4, HF2 to pin5–8, and so on).

Connections between HF Ports and Pulse Generator (PG)

Output terminals (PGUs) of the pulse generator (PG) are connected to the input terminals of the HF ports using the BNC cables. These BNC cables are furnished with the HP 4062F system, and have tags like "PG*xx*". The first digit is the PG number, and second digit is the output terminal number. For example, "PG21" means output terminal 1 (OUTPUT 1) of the second PG (PG2).



/usr/pcs/etc/FLCONFIG file includes information on the connections between the output terminals of the PGs and the HF ports. The system software controls the PGs and SWM according to the information in this file. For details on the FLCONFIG file, see "Configuration File /usr/pcs/etc/FLCONFIG".

Cable Adapter Switches and Connections

Cable adapter (HP part number 04085-61066) on SWM has two multiplexer switches and two open/close switches. You need to use a multiplexer switch when you force three level pulse that is more than ±10 V (that is, sum of absolute values of two amplitudes is more than 20 V). You need to use an open/close switch when you want to set DUT's terminals to open state.

One PGU (output unit of PG) is required just for controlling these switches. So, if you want to use the switches, you need two or more PGUs.

Figure 1-7 shows the simplified circuit diagram of these switches. SW1 and SW4 are multiplexer switches, and SW2 and SW3 are open/close switches. SW1, SW2, and SW3 are semiconductor relays, and SW4 is a reed relay. Semiconductor relays have a long life. Reed relays are for higher frequency pulses.

⚠

Note

Do not switch SW4 while forcing voltage or current to SW4. If you do, the relay life will be much shorter.

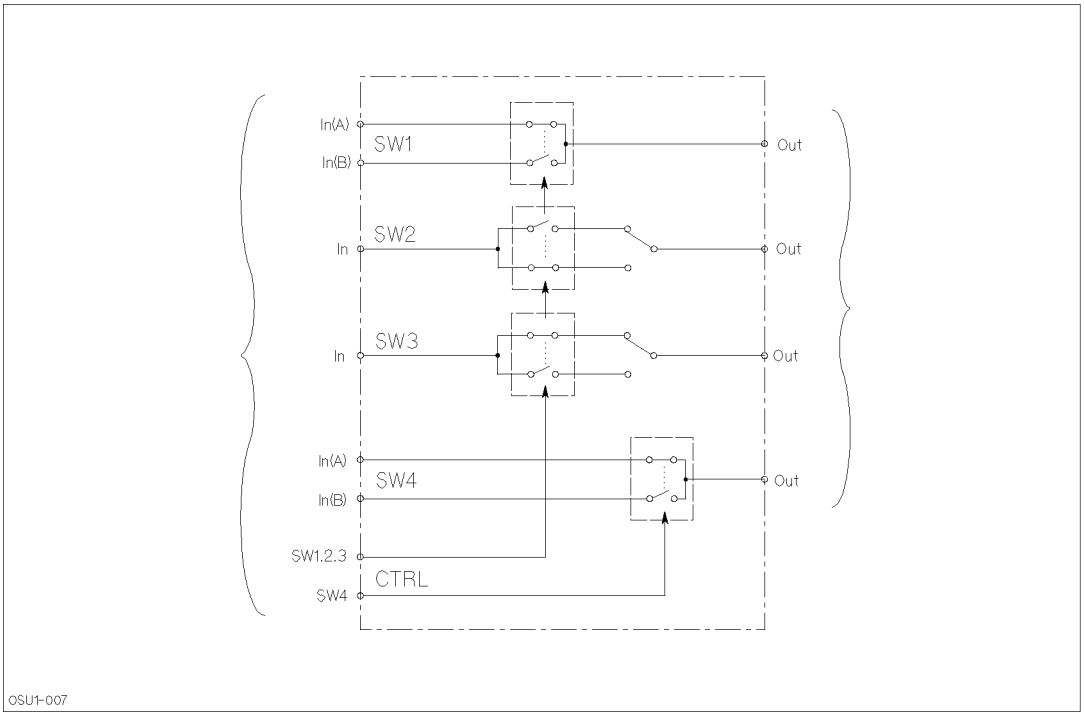


Figure 1-7. Simplified Circuit Diagram of Switches

Cable adapter has two slide switches (labeled "MODE SELECTOR") for setting "default" settings of SW2 and SW3. When slide switch is NC (normally closed), SW2 or SW3 is closed as default. When slide switch is NO (normally open), SW2 or SW3 is open as default.

"Default" means the relay status when the Set_sr_pg subprogram specifies the PGU that will control the open/close switches. It does *not* mean the status when the system software

initializes the system. Initial status for SW2 and SW3 is open and does not depend on mode selector. Initial status for SW1 is not connected. Initial status for (SW4) is A side.

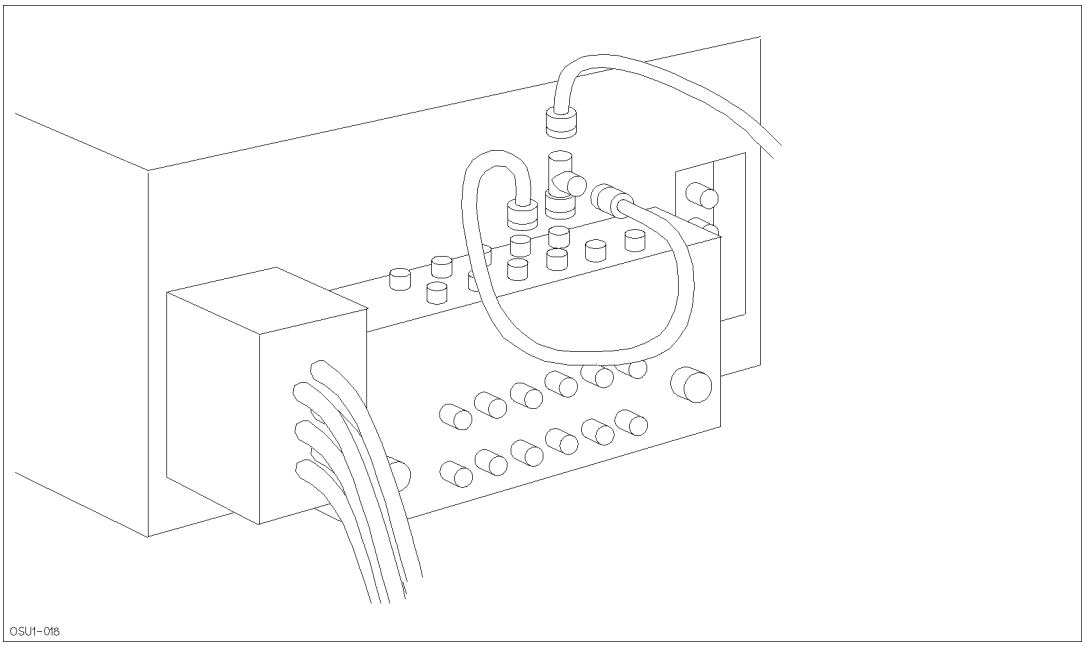
One PGU is used to control the switches. The PGU used for controlling the switches is specified in the measurement program, so you do not need to specify information about this PGU. In the HF port assignment in the FLCONFIG file, do *not* specify the PGU that is used to control the switches.

There are two control (CTRL) terminals on the cable adapter. You connect the desired PGU to the CTRL terminal. SW1, SW2, and SW3 are connected to same CTRL terminal. So, the PGU connected to the SW1·2·3 CTRL terminal, controls SW1, SW2 and SW3 simultaneously. The PGU connected to SW4 CTRL terminal, controls SW4 only.

Note

Do not connect an output terminal other than a PGU to the CTRL terminals. Applying high voltage to the CTRL terminals may destroy the internal circuits in the cable adapter.

To control SW1, SW2, SW3, and SW4 simultaneously, use an m-f-f BNC T connector (HP part number 1250-2486) and a BNC cable as shown in the following figure.



Accessories for Switching Matrix

Figure 1-8 shows available accessories for the HP 4085F SWM. The following briefly describes these accessories.

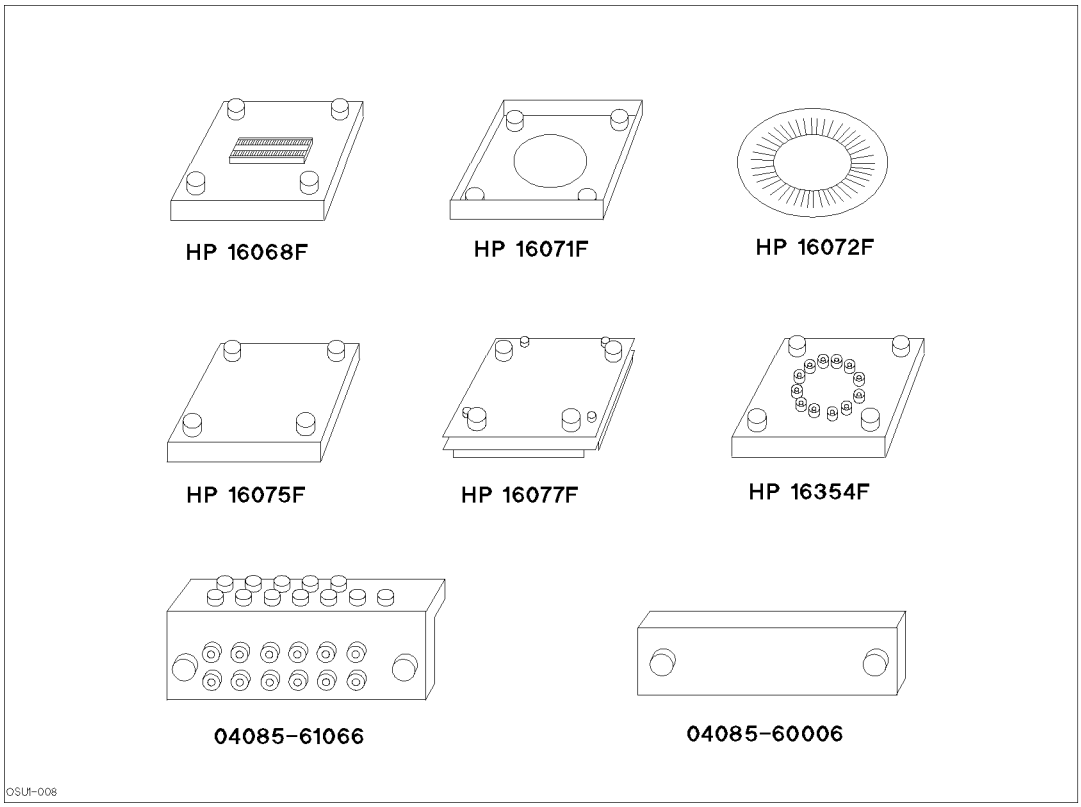


Figure 1-8. Accessories for HP 4085F SWM

HP 16068F 48-pin DIP Fixture
This fixture is for DUT that is packaged in a 48-pin DIP (dual in-line package). This fixture is designed for low leakage current measurements.

HP 16071F Universal Fixture
The center of this fixture is a blank circuit board with standard 2.54 mm (0.1 inch) hole spacing. You can install your desired socket on this fixture.

HP 16072F Personality Board
This board is for connecting the SWM to your automatic or semiautomatic wafer prober. See the HP 16072F *Operation manual* about the connections.

HP 16075F Relay Test Adapter
This adapter is for performing the Relay Test in the diagnostics (DIAG.F) program, and can be used for HP 4062C and HP 4062UX systems.

HP 16077F Extension Cable Fixture

This fixture is for connecting the SWM to a manual wafer prober. You mount this fixture on the SWM, then connect the pins on this fixture to probe needles of your wafer prober. See the HP 16077F *Operation Manual* about the connections.

Warning

Potentially hazardous voltages may be present at the measurement pins of the extension cable fixture when the output enabled lamp on the SWM is lit. Make sure the protective cover plate of the extension cable fixture is installed before you operate the system.

HP 16354F Pulse Test Module

This module is for the AC Test of the diagnostics (DIAG.F) program.

HP p/n 04085-61066 Cable Adapter

This adapter is usually installed on the SWM, and receives output signals from the pulse generators, and passes signal to HF ports in the SWM.

HP p/n 04085-60006 Relay Test Shorting Adapter

When performing the Relay Test, you install this adapter instead of the cable adapter.

HP 8110A Pulse Generator

HP 4062F system uses HP 8110A 150 MHz pulse generators (PGs) as pulse sources. To use the HP 8110As with the HP 4062F system, the HP 8110A must be equipped with one or two HP 81103A 10 V/2 ns output channels (PGUs) and with an HP 81106A PLL/external clock. And you must use HP 8110A option UN2, which means the input/output terminals of HP 8110A are on real panel, not front panel.

Pulse signal forced from the HP 8110A goes through the HF ports and pin boards on the SWM, then to your device under test (DUT). The HF ports are designed for passing high-frequency pulse signals, so you can force more accurate pulse signals to your DUT through the HF ports than through other ports.

Pulse Waveform from PG

Figure 1-9 shows the pulse waveform forced from the PG. The PG can force two types of pulse waveforms:

- *2-level pulse*: has a base level and a peak level
- *3-level pulse*: has a base level and two peak levels.

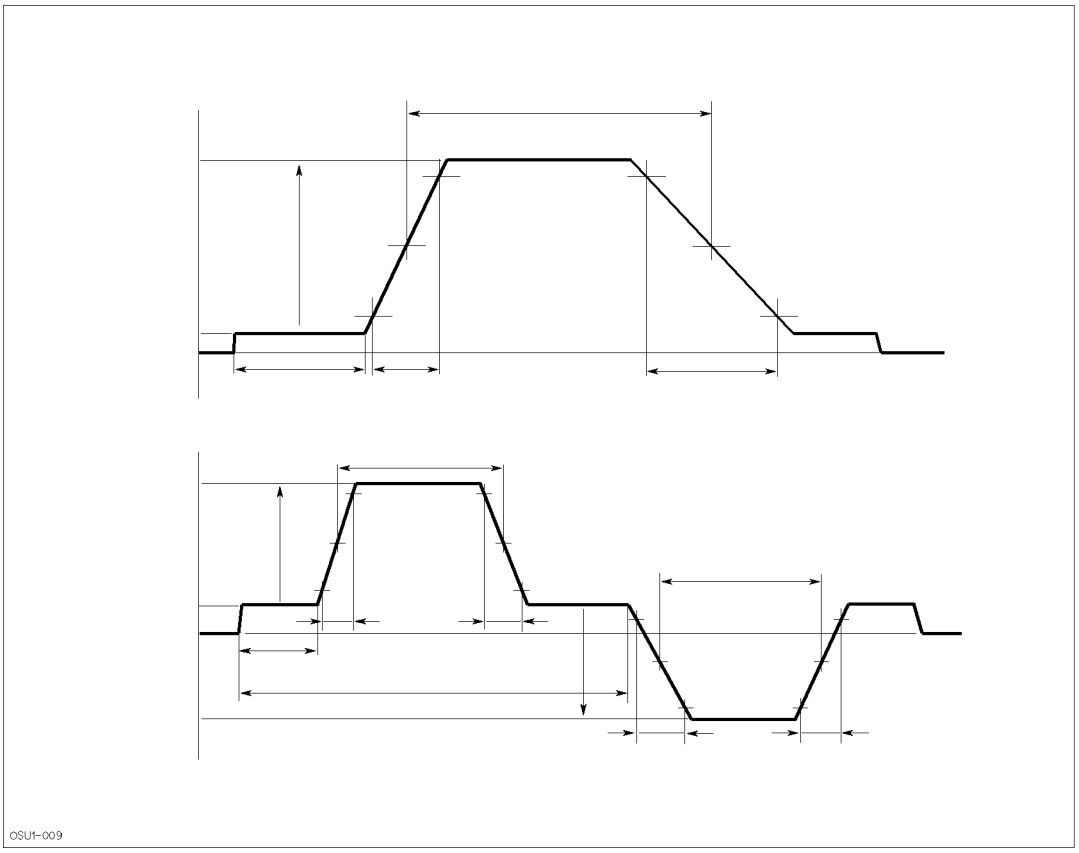


Figure 1-9. Pulse Waveform Forced from PG

2-Level Pulse Output

The following shows the parameters of a 2-level pulse, which has one base level and one peak level. See upper figure of Figure 1-9.

Pulse Base	Base voltage level forced before and after the pulse wave in a pulse period.
Pulse Amplitude	Absolute pulse amplitude from the pulse base.
Pulse Width	Absolute pulse width measured from 50% of base-to-peak pulse (leading edge) amplitude to 50% of peak-to-base pulse (trailing edge) amplitude.
Delay Time	Absolute delay from the start of a pulse period to the start of the leading-edge of the pulse.
Leading-edge Transition Time	Absolute transition time measured from 10% to 90% of pulse amplitude.
Trailing-edge Transition Time	Absolute transition time measured from 90% to 10% of pulse amplitude.

3-Level Pulse Output

The following shows the parameters of the 3-level pulse, which has one base level and two peak levels. See lower figure of the Figure 1-9.

Pulse Base	Base voltage level forced before and after the pulse wave in a pulse period.
Pulse Amplitude 1	Absolute pulse amplitude of the first pulse peak from the pulse base.
Pulse Width 1	Absolute pulse width measured from 50% of base-to-peak pulse (leading edge) amplitude of the first pulse peak to 50% of peak-to-base pulse (trailing edge) amplitude of the first pulse peak.
Delay Time 1	Absolute delay from the start of a pulse period to the start of the leading-edge of the first pulse.
Leading-edge Transition Time 1	Absolute transition time measured from 10% to 90% of pulse amplitude of the first pulse peak.
Trailing-edge Transition Time 1	Absolute transition time measured from 90% to 10% of pulse amplitude of the first pulse peak.
Pulse Amplitude 2	Absolute pulse amplitude of the second pulse peak from the pulse base.
Pulse Width 2	Absolute pulse width measured from 50% of base-to-peak pulse (leading edge) amplitude of the second pulse peak to 50% of peak-to-base pulse (trailing edge) amplitude of the second pulse peak.
Delay Time 2	Absolute delay from the start of a pulse period to the start of the leading-edge of the second pulse. Note that the delay time 2 must be greater than the total of the pulse width 1 and delay time 1.
Leading-edge Transition Time 2	Absolute transition time measured from 10% to 90% of pulse amplitude of the second pulse peak.
Trailing-edge Transition Time 2	Absolute transition time measured from 90% to 10% of pulse amplitude of the second pulse peak.

Synchronization between Pulse Generators

HP 4062F system software has two modes to synchronously force pulse signals from several pulse generators: *normal mode* and *pattern mode*.

Normal mode uses the HP 8110A's pulse mode to force pulse signals. Trigger signal output from the TRIGGER OUT terminal of the master HP 8110A (PG1) is transferred to the EXT INPUT terminal of the slave HP 8110As (PG2 to PG5), and all the output units (PGUs) synchronously force pulse signals.

If pulse period is comparatively long, the time accuracy of the normal mode is inferior to that of the pattern mode.

Pattern mode uses the HP 8110A's pattern mode to force pulse signals. Trigger signal output from the master HP 8110A (PG1) is transferred to the slave 1 HP 8110A (PG2), and the pattern signal forced from the PG2 is transferred to the other HP 8110As (PG3 to PG5). So only the pulse signals forced from PG2 to PG5 can be synchronous.

Trigger Connections between Pulse Generators

To synchronously force pulse signals, all the PGs in the HP 4062F system must be connected using the trigger signal cables. Figure 1-10 shows the trigger cable connections between the PGs for each case.

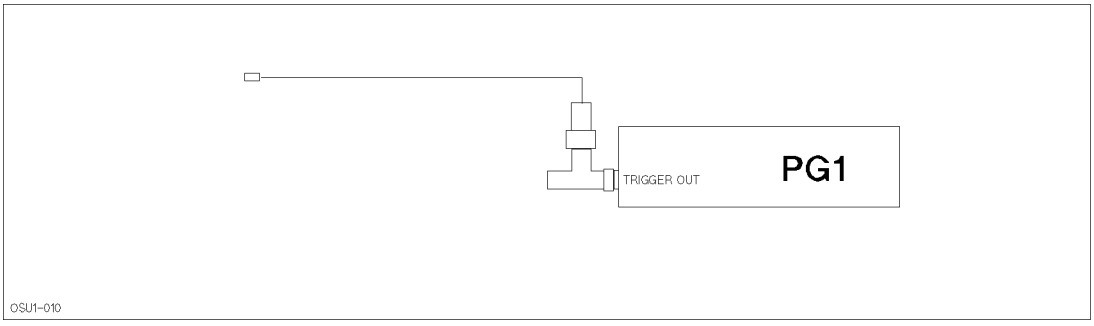
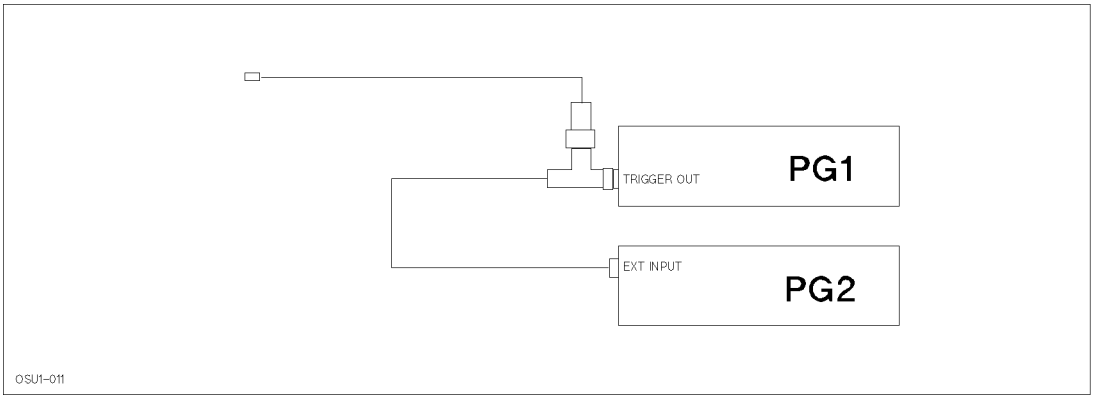
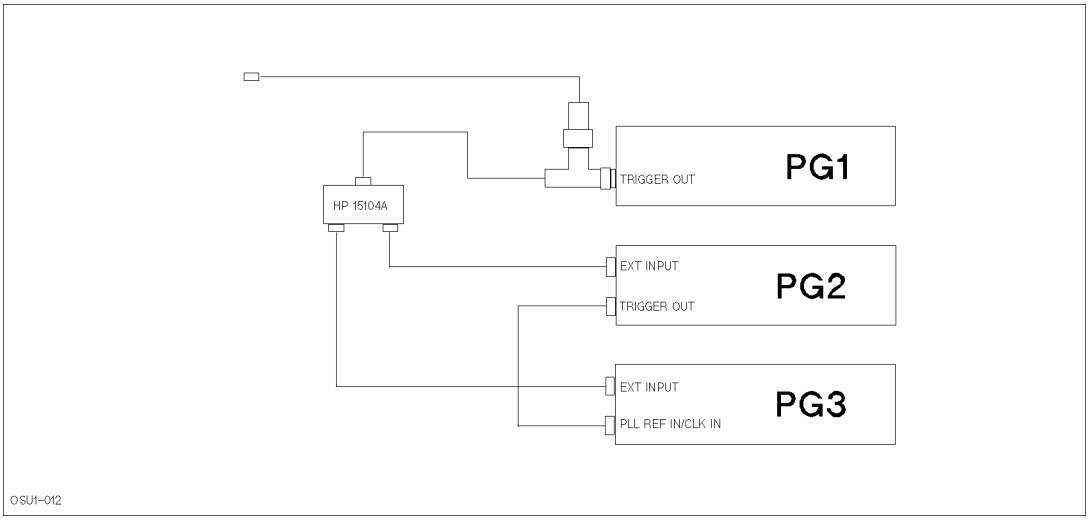


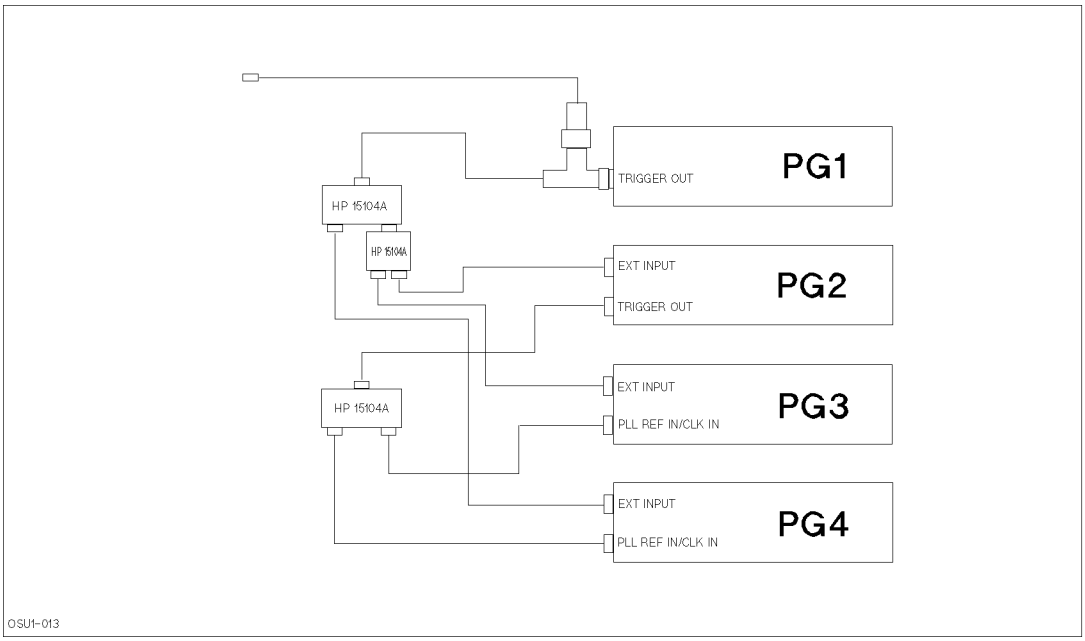
Figure 1-10. Trigger Connections for One Pulse Generator



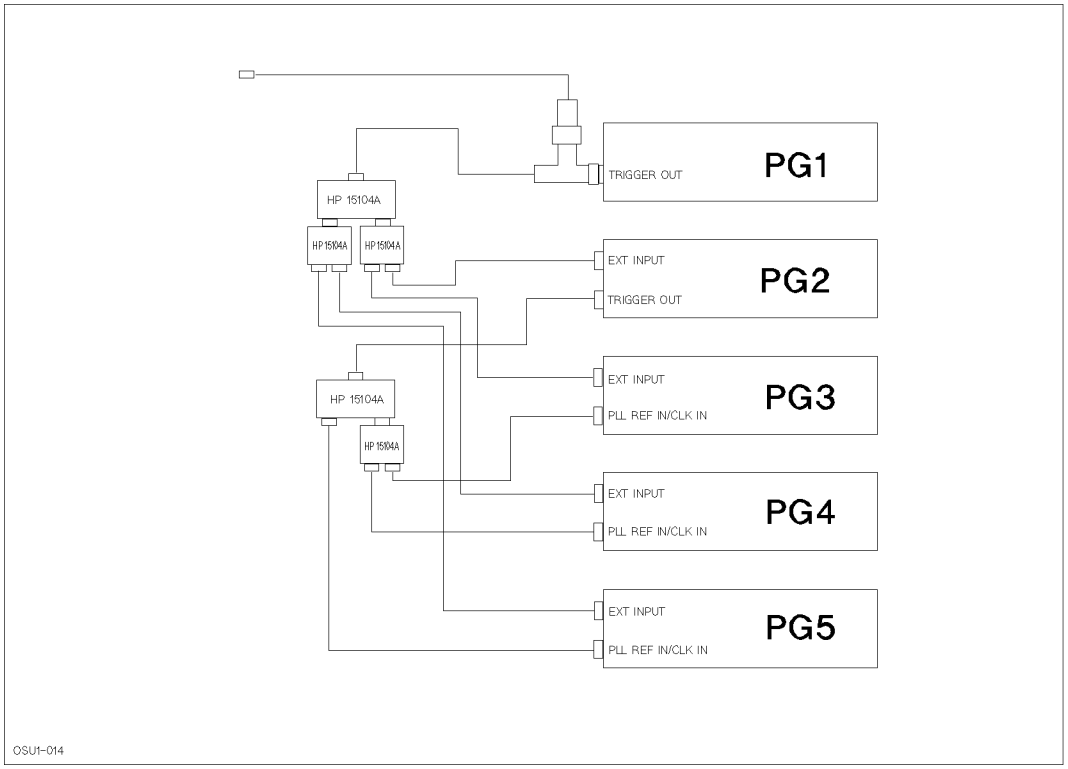
Trigger Connections for Two Pulse Generators (continued)



Trigger Connections for Three Pulse Generators (continued)



Trigger Connections for Four Pulse Generators (continued)



Trigger Connections for Five Pulse Generators (continued)

The trigger signal output from the TRIGGER OUT terminal of the PG1 is used as trigger signal for the other PGs. When you perform performance verification of the HP 4062F system, this trigger signal is also used as trigger signal for an oscilloscope. When trigger signal for the oscilloscope is not used, terminate end of trigger cable with a 50 Ω terminator. PG1 is also called the "master PG".

The trigger signal output from the master PG passes through splitters (HP 15104A), then to EXT INPUT terminals of PG2 to PG5. PG2 to PG5 are also called "slave PGs". For normal mode, the trigger signal from the master PG is transferred to the slave PGs. For the pattern mode, the trigger signal from the TRIGGER OUT terminal of the PG2 is additionally transferred to the PLL REF IN/CLK In terminal of PG3 to PG5.

HP 4062F System Software

HP 4062F system software is supplied on a cartridge tape or DDS formatted DAT cassette. The system software can be installed using the HP-UX UPDATE (1M) command. For details, see the HP 4062F *Installation Manual*.

Figure 1-11 shows the directory structure of the HP 4062F system software. Since the HP 4062F system has the HP 4062UX functions plus flash memory evaluating functions, some system files are added to that of the HP 4062UX system.

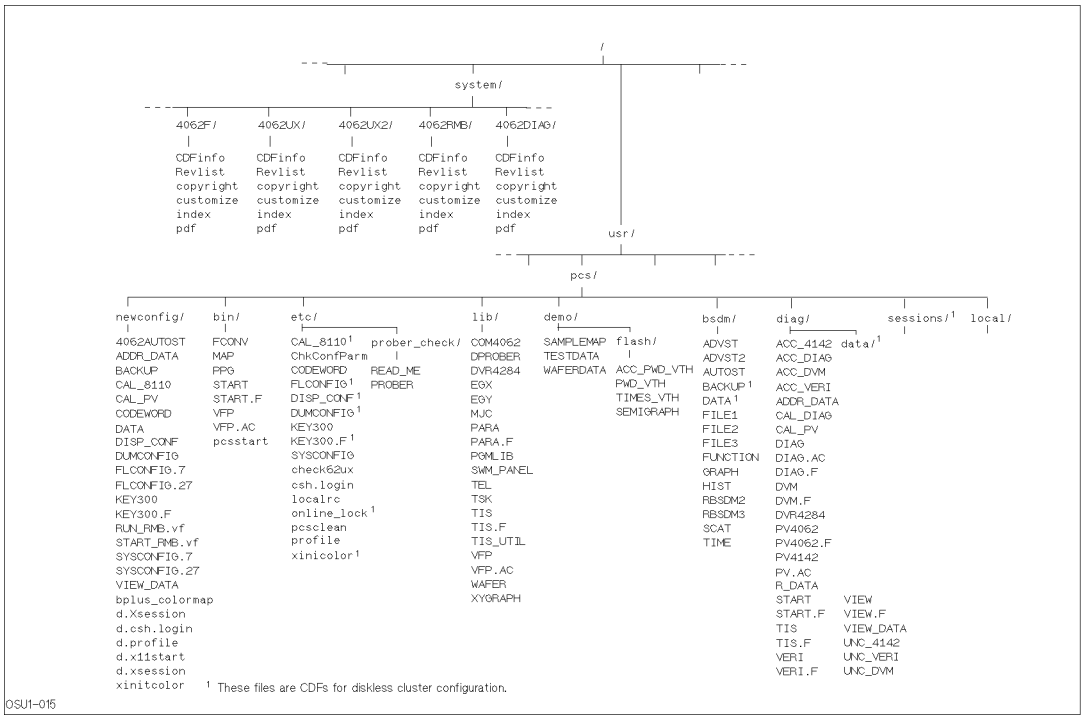


Figure 1-11. Directory Structure of HP 4062F System Software

The following explains important system files of the HP 4062F:

/usr/pcs/etc/SYSCONFIG — This file includes the information on the HP-IB addresses of the system instruments (except the PGs) and wafer probers. This file is also used for the HP 4062UX system.

/usr/pcs/etc/FLCONFIG — This file includes the information on the HP-IB addresses of the PGs in the HP 4062F system, and on assignments between the output terminals of the PGs and HF ports.

/usr/pcs/bin/START.F — This is START program file for HP 4062F system. This program must be executed when you start HP 4062F system.

/usr/pcs/bin/VFP.AC — This program is used for monitoring the pulse wave from the PGs. You can optimize the pulse parameters by changing the pulse parameter settings interactively using this program. You usually execute this program while monitoring the pulse wave using a oscilloscope.

/usr/pcs/lib/TIS.F	This file includes the subprograms and function programs called "Test Instruction Set (TIS)" for the HP 4062F system. This file has the subprograms used for the HP 4062UX system plus subprograms for controlling the PGs.
/usr/pcs/lib/PARA.F	This file includes the subprograms and function programs called "PARA" for the HP 4062F system. This file has the subprograms used for the HP 4062UX system plus subprograms for writing/erasing flash memory cells. To use the PARA.F file, you must also link the TIS.F file
/usr/pcs/diag/DIAG.F	This is a diagnostics (DIAG) program file for HP 4062F system.

Configuration File /usr/pcs/etc/FLCONFIG

In the /usr/pcs/etc/FLCONFIG file, you specify the HP-IB addresses of the PGs, and assign HF ports to output terminals (PGUs). This file is an HP BASIC ASCII file, so you can change the information by using the HP BASIC `EDITor` or the HP-UX editor (for example `vi` editor).

This file contains the following information after the HP 4062F system software is installed.

```
1000 !!  RE-SAVE "/usr/pcs/etc/FLCONFIG"
1010 !!
1020 ! PG1  HP8110A  2725
1030 ! PG2  HP8110A  2726
1040 ! PG3  HP8110A  2727
1050 ! PG4  HP8110A  2728
1060 ! PG5  HP8110A  2729
1070 !
1080 ! HF1  PG11
1090 ! HF2  PG12
1100 ! HF3  PG21
1110 ! HF4  PG22
1120 ! HF5  PG31
1130 ! HF6  PG32
1140 ! HF7  PG41
1150 ! HF8  PG42
1160 ! HF9  PG51
1170 ! HF10 PG52
```

PG number, product number of PG, its HP-IB address, and HF port assignments of the output terminals (PGUs) are defined in the /usr/pcs/etc/FLCONFIG file.

HP-IB Address of PG.
The following shows how to specify the HP-IB addresses of the PGs:

```
1010  !!
1020  ! PG1   HP8110A    2725
1030  ! PG2   HP8110A    2726
1040  ! PG3   HP8110A    2727
1050  ! PG4   HP8110A    2728
1060  ! PG5   HP8110A    2729
```

The first field is PG number, second field is product number of PG, and third field is the device selector (select code $\times$ 100 + HP-IB address) of the PG.

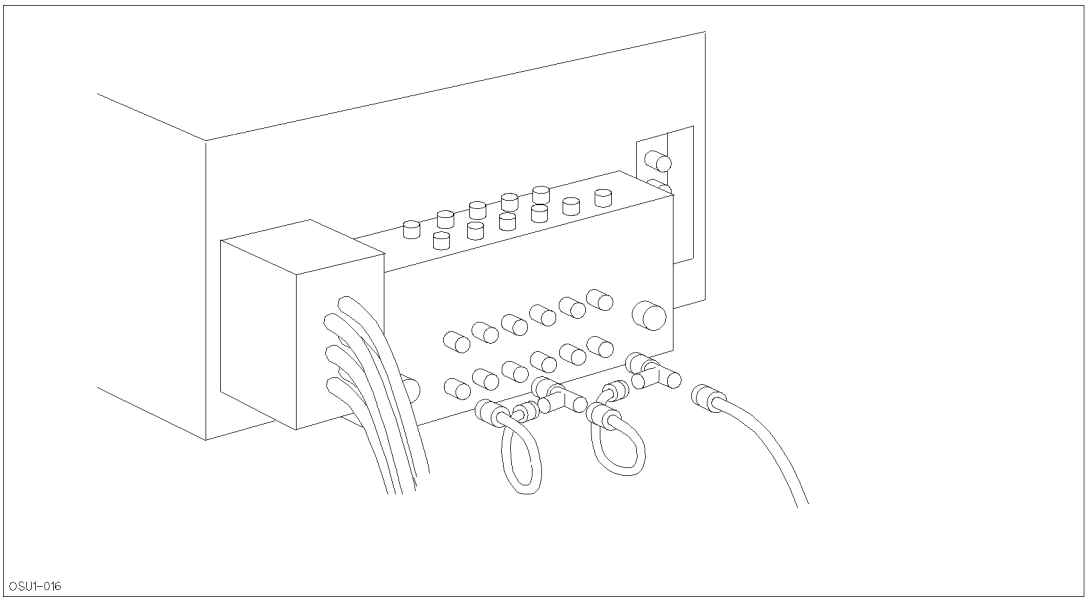
HF Ports and Output Terminals of PGs.
The following example shows how to specify the assignment between the HF ports and output terminals of PGs:

```
1010  !!
1020  ! HF2   PG11
1030  ! HF4   PG12
1040  ! HF6   PG21
1050  ! HF8   PG22
```

First field is the HF port number, and second field is output terminal number of PG. See "Connections between HF Ports and Pulse Generator (PG)" for the output terminal number.

When connecting an output terminal of a PG to more than one HF port using BNC T connectors, specify as shown in the following example:

```
1010  !!
1020  ! HF1   PG11
1030  ! HF7   PG11
1040  ! HF11  PG11
1050  ! HF12  PG12
```



In example above, OUTPUT 1 terminal of PG1 is connected to three HF ports: HF1, HF7, and HF11.

If you use multiplexer switches or open/close switches, be careful about the following:

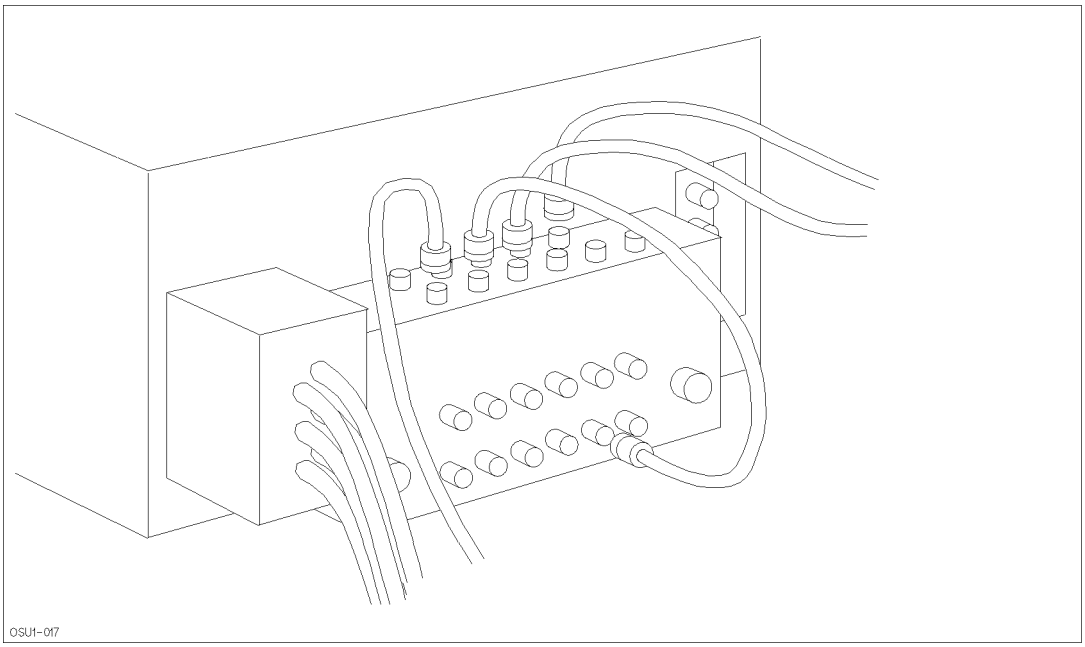
- Do *not* specify an HF port assignment in the FLCONFIG file for output terminal of PG that is connected to a CTRL terminal.
- For the open/close switches, specify HF port assignments in FLCONFIG file as follows. For HF port connected to the OUT terminal, specify the PG output terminal that is connected to the IN terminal.
- For the multiplexer switches, specify HF port assignments in FLCONFIG file as follows:
  - For the HF port connected to the OUT terminal, specify PG output terminal that is connected to IN (A) terminal.
  - Do *not* specify an HF port for the PG output terminal that is connected to IN (B) terminal.

For example, for the following connections:

- IN (A) terminal of SW1 (multiplexer switch) is connected to PG21.
- IN (B) terminal of SW1 is connected to PG22.
- OUT terminal of SW1 is connected to HF3 port.
- CTRL SW1·2·3 terminal is connected to PG12.

Include the following HF port assignment information in the FLCONFIG file:

```
1010 !!
1020 ! HF3 PG21
```



Note that you must *not* specify PG22 (which is connected to IN (B) of SW1) and PG12 (which is connected to CTRL SW1·2·3) in the /usr/pcs/etc/FLCONFIG file.

2

Getting Started

To start the HP 4062F environment, you must do some items in a certain order. Table 2-1 is a guideline for starting the HP 4062F environment and related reference documentation.

Table 2-1. Starting the HP 4062F Environment

	Operation	Reference Point
1.	Starting the HP-UX operating system (usually HP-UX automatically starts when you turn on the computer.)[1]	*HP-UX System Administration Tasks*
2.	Turning on the system instruments	Chapter 1 of HP 4062C/UX *System Operation*
3.	Warning up the system instruments (at least 40 minutes)	——
4.	Logging in	*Using HP-UX*
5.	Starting the HP BASIC/UX environment	*Using the HP BASIC/UX*
6.	Starting the HP 4062F environment	This chapter

1 The HP-UX operating system continuously runs except when you perform some system administration tasks, because the HP-UX is a multiuser/multitask system.

This chapter provides information on the START.F and VFP.AC programs. The START.F program is used to start the HP 4062F environment. The VFP.AC program enables you to force pulse signal interactively without programming.

START.F Program

This section covers the START.F program, which initializes the software environment of the HP 4062F system.

The START.F program functions as follows:

- Checks the security to compare the codeword in the CODEWORD file with the codeword calculated from the serial number of the ID module.
- Generates the parametric test session.
- Checks the responses from the system instruments according to the system configuration information.
- Initializes the individual system instruments.
- Checks the measurement cable connections for SMU1–4 and AUX2–4 ports.
- When the CMU (HP 4280A) is connected to the AUX3 and AUX4 ports, measures the residual impedance of the measurement cables and SWM, and saves it into the system memory for error compensation.

You must execute the START.F program before you can use any other system software. However, you do not need to execute the START.F program if you are only entering the measurement program (program code) without debugging. You must also execute the START.F program after any of the following operations:

- When you turn your HP 4062F on.
- If you turn a system instrument off and on after you have executed START.F.
- If you rearrange system cabling after you have executed START.F.
- If you change the HP 4142B's plug-in module configuration after you have executed START.F.
- If you execute a SCRATCH A command after you have executed START.F with an wafer prober.
- If you change the configuration of the SWM's pin boards after you have executed START.F.
- If you execute a program except MAP before you have linked TIS.F.
- If you execute any of the Statistical Graphics programs.

Note

If the COM blocks declared during START.F are lost, the following symptoms may occur:

- "`ERROR 145 IN 10 May not build COM at this time.`" may be displayed.
- "`TIS ERROR 20 Device not present.`" may be displayed.

If any of these symptoms occur, execute START.F before proceeding.

For more detailed information on COM blocks, refer to the *BASIC/UX Programming Techniques*.

Executing the START.F Program

This section describes how to execute the START.F program. Assume that the HP BASIC/UX environment has already been started.

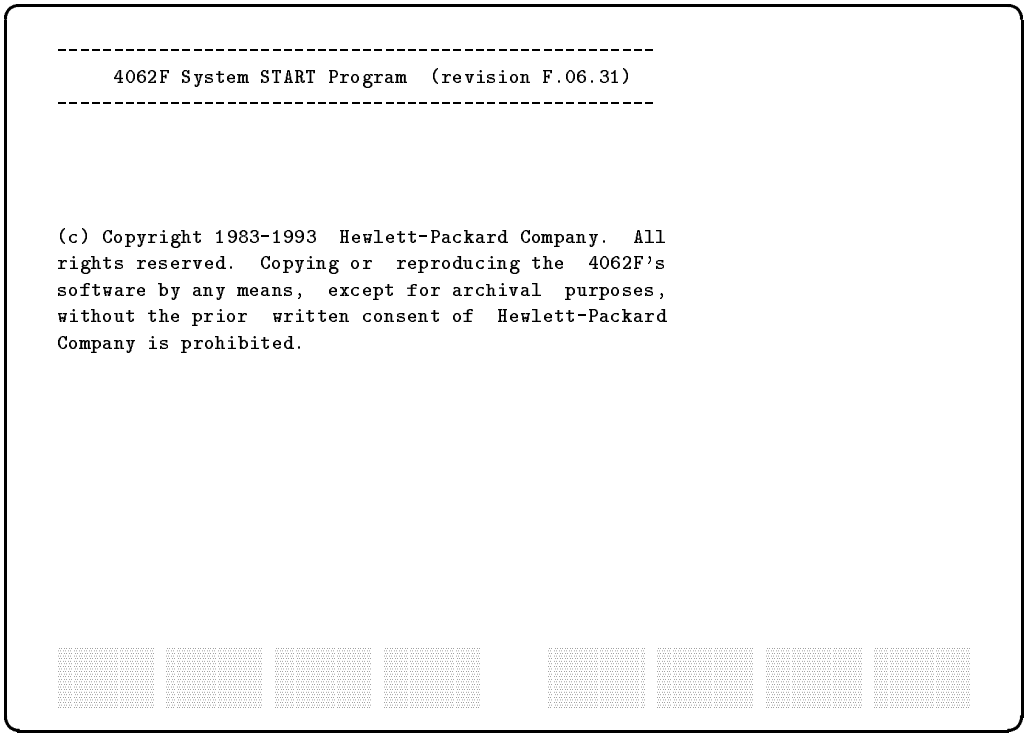
1. Load the START.F program file.

   `LOAD "/usr/pcs/bin/START.F"` Return

2. Start the START.F program.

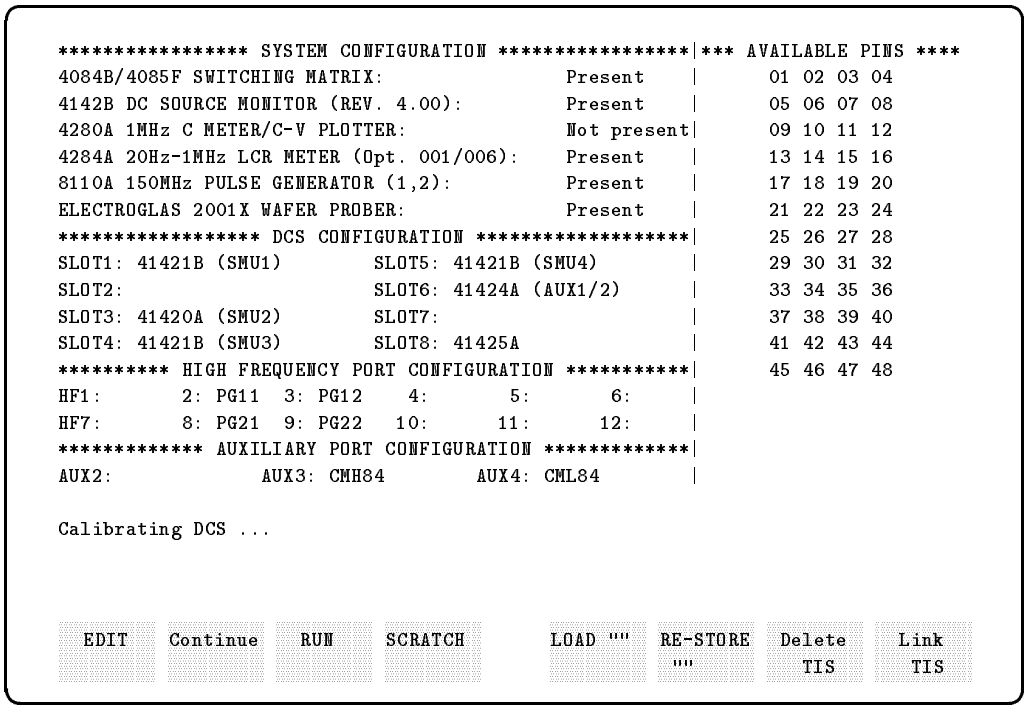
   `RUN` Return *or select* `RUN` *softkey*

   When you execute the START.F program after starting the HP BASIC/UX environment or after executing a SCRATCH A command, the copyright message and revision number of the HP 4062F system software are displayed as follows:



Copyright Message Page

Then your system's configuration is displayed, and the system instruments are initialized.



System Configuration Page while System Instruments are being Initialized

System Configuration Page

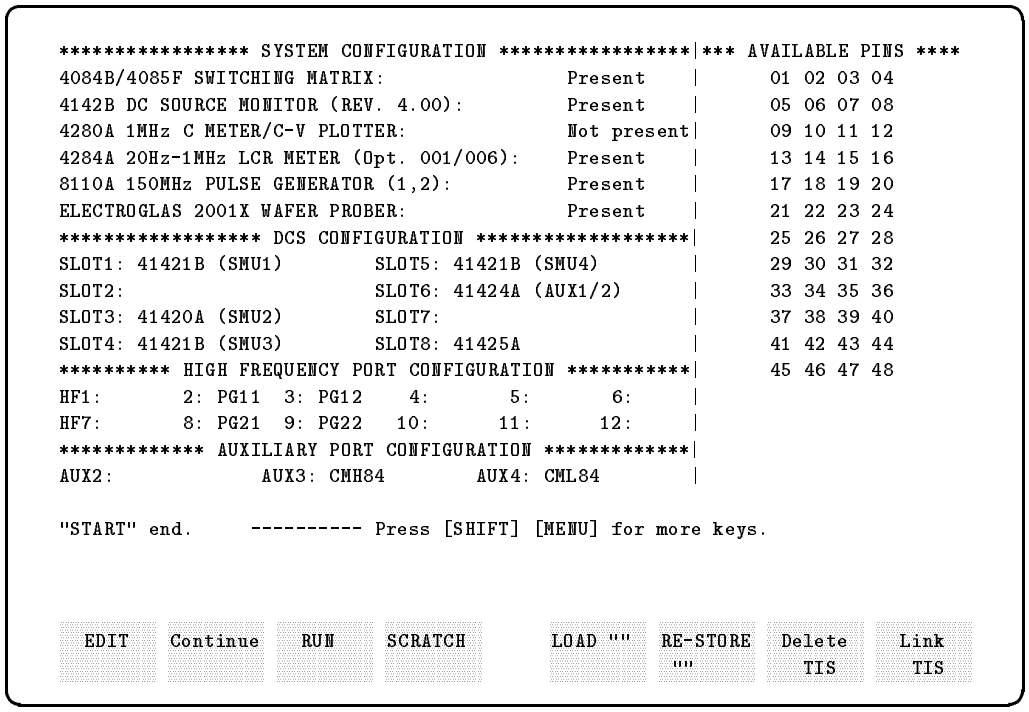


Figure 2-1. System Configuration Page

When you execute the START.F program, your system configuration page similar to Figure 2-1 is displayed.

- SYSTEM CONFIGURATION

  This section shows which system instruments are correctly set ("`Present`") for system operation. If an instrument is displayed as "`Not Present`", make sure the instrument is turned on, check the instrument's HP-IB address setting and HP-IB cable connections, and execute the START.F program again. The system controller conducts a serial poll of the individual system instruments on the HP-IB during START.F.

  "`4142B DC SOURCE MONITOR`" description shows the firmware ROM revision number. If the number is under 3.00, you can force only up to 20 V to search port when you perform analog search measurements.

  "`4284A 20Hz-1MHz LCR METER`" description shows if option 001 Power Amplifier/DC bias and option 006 2 m/4 m Cable Length Operation exist. If option 001 does not exist, you cannot perform C-V measurement. If option 006 does not exist, the measurement accuracy is not guaranteed.

  "`8110A 150MHz PULSE GENERATOR`" description shows available pulse generators. The pulse generator number is specified in the FLCONFIG file.

- DCS CONFIGURATION

  This section shows each HP 4142B DCS slot number, the model number of the plug-in module installed in each slot, and the SWM port (in parentheses) to which each module is connected. If an SWM port is not displayed, that module is not connected to the SWM (in this example, slot 8). Note that the HP 41420A SMU occupies 2 slots, and START.F program lists only higher of the 2 slots where the SMU is installed (in this example, slot 3). Vacant slots are left blank (in this example, slot 7). If you reconfigure any DCS plug-in modules after you have executed START.F, you *must* execute START.F again.

- HIGH FREQUENCY PORT CONFIGURATION

  This section shows HF ports and connected output unit numbers of PGs. The number before the colon (:) shows the HF port number, and the number after the colon shows the PG unit (PG*xy*: *x* is PG number, *y* is output terminal of PG). The connections shown in this section are not actually checked. The assignment information of the FLCONFIG file appears here.

- AUXILIARY PORT CONFIGURATION

  This section shows the configuration of the SWM's AUX ports. If the AUX ports are connected incorrectly, "`Nothing Connected`" is displayed. If you reconfigure any AUX port cables after you have executed START.F, you *must* execute START.F again.

- AVAILABLE PINS

  This section lists the available measurement pins (pin boards) installed in the SWM. Pin boards not present at system START.F are not listed (are not displayed).

Softkeys

The START.F program automatically loads the /usr/pcs/etc/KEY300.F softkey file. Figure 2-2 shows the softkeys available after the START.F program executes.

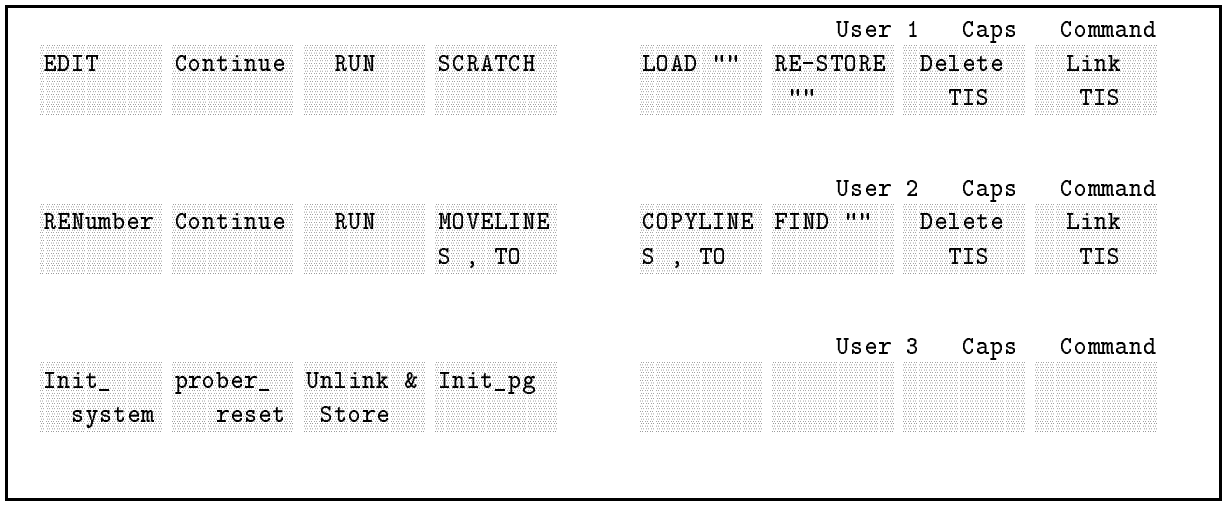
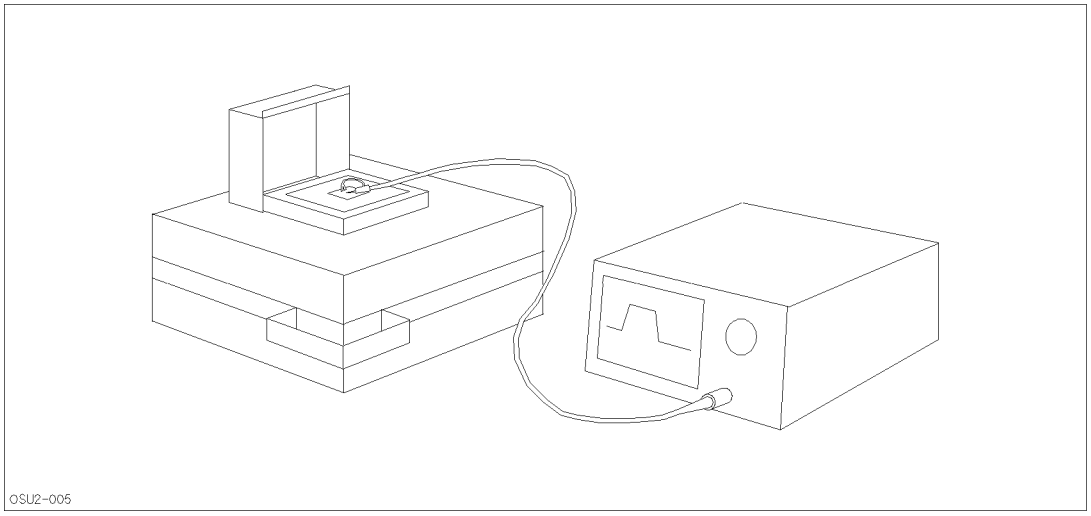


Figure 2-2. System Softkeys

Forcing Pulses Manually

You can use the VFP.AC program to force pulse signal manually. You can use an oscilloscope to observe actual pulse waves forced from the pulse generators, and fine tune the pulse wave by using the VFP.AC program.

Hewlett-Packard recommends you to prepare an oscilloscope to observe actual pulse wave.



Note

You *cannot* make a pulse wave appear at *more than four* measurement pins by using the VFP.AC program even if PGU is connected to multiple HF ports. In this case, only first assignment of HF port and PGU in the FLCONFIG file is available for the VFP.AC program. For example, when the following assignments are described in the FLCONFIG file, only HF2 port is available for PG11, so only measurement pins 5 to 8 can be connected to PG11.

```
 . . . . . .
230  ! HF2  PG11
240  ! HF3  PG11
250  ! HF12 PG11
 . . . . . .
```

Starting and Quitting the VFP.AC Program

This section describes how to start and quit the VFP.AC program. Assume that the HP BASIC/UX environment has already been started.

- Use the following procedure to start the VFP.AC program.

Note

Note that the START.F program must be executed before you run the VFP.AC program. See "Executing the START.F Program" about how to run the START.F program.

---

1. Load the VFP.AC program file.

   `LOAD "/usr/pcs/bin/VFP.AC"` Return

2. Start the VFP.AC program.

   `RUN` Return *or select* `RUN` *softkey*

- Use the following procedure to quit the VFP.AC program.

1. If the pulse signal is forced continuously, select `Num. of Periods` softkey, and enter a number other than 0 (zero). If this softkey does not appear, press Shift+Menu key to show next softkey menu.
2. Select `Quit` softkey.

Setting the Pulse Waveform

When the VFP.AC program starts, a page where you can set the parameters for the output pulse waveform appears as shown in Figure 2-3.

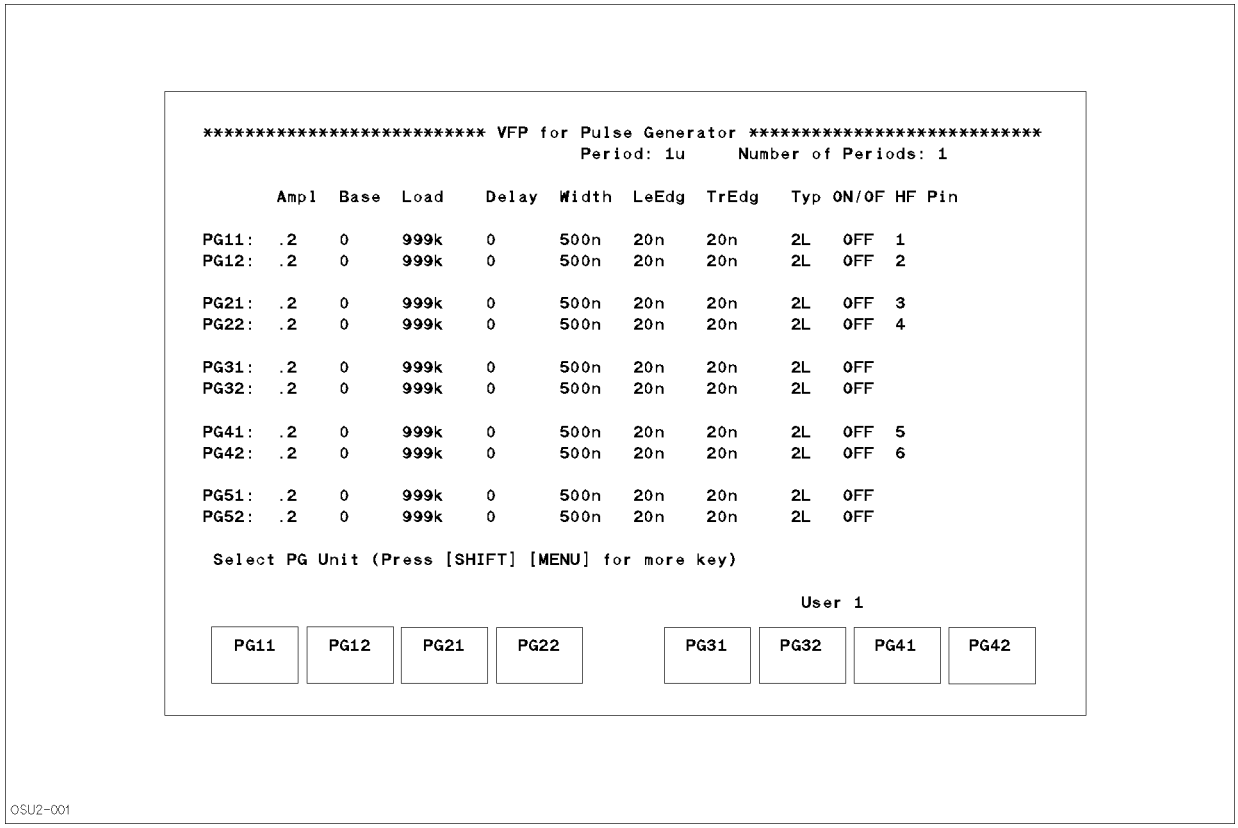


Figure 2-3. Pulse Parameters Setting Page of VFP.AC

For each PGU, you can set the pulse parameters. At the top of the page, you can set the pulse period and number of periods.

Period — Pulse period. This field sets the pulse period for the PGUs. The value must be greater than or equal to the maximum period (T) setting of the PGUs, where T means as follows:

- For the "2-level pulse" output mode:

  T = (pulse width + delay time + 30 ns)×1.1

- For the "3-level pulse with one PGU" output mode:

  T = (pulse width 2 + delay time 2 + 30 ns)×1.1;
  Where, (pulse width 1 + delay time 1)≤(delay time 2)

- For the PGU used to control the switches:

  T = (pulse width + delay time + 30 ns + 2 ms)

For more information about T, see "Force_pg" in chapter 4.

Number of Periods	Number of periods. This field sets the number of pulse periods. If you specify 0 (zero) in this field, the pulse is forced continuously.

You can use the "`Typ`" fields to set the sort of pulse wave forced. The `2L` in the `Typ` fields means 2-level pulse, `3L` means "3-level pulse with one PGU", and the `M`*xx* means "3-level pulse with a multiplexer switch", where *xx* is the partner PGU.

The following covers how to set the parameters for 2-level, 3-level with one PGU, and 3-level pulse with a multiplexer switch.

How to Set the "2-level Pulse" Output

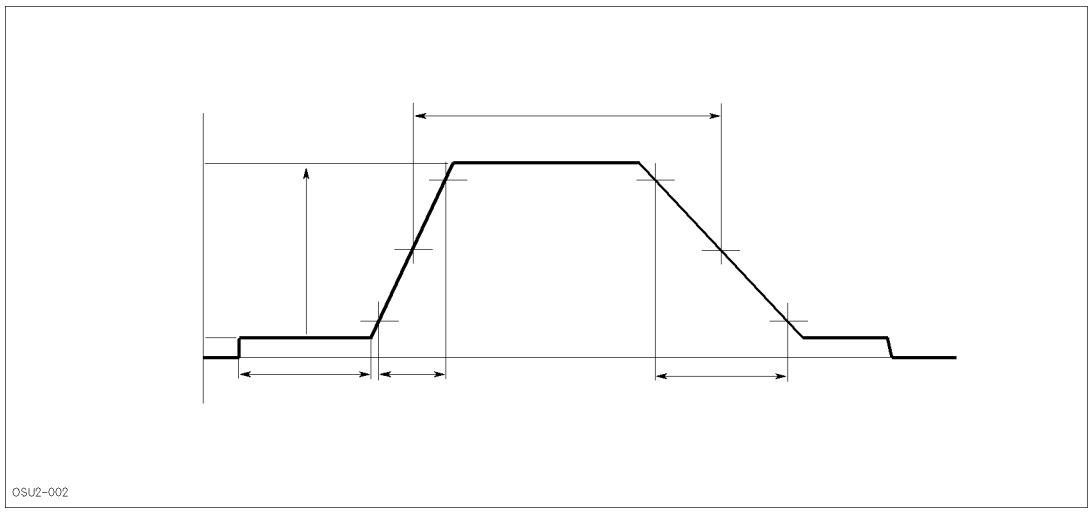


Figure 2-4. Setting the 2-level Pulse Output

One PGU is used for 2-level pulse output. You set the pulse parameters for the 2-level pulse so that the PGU forces one pulse during one period as follows:

1. Select softkey of desired PGU. If the system has more than eight available PGUs, other PGU softkeys are in the next softkey menu. To show next menu, press Shift+Menu.
2. If the selected PGU is not "2L", press Shift+Menu keys, select `Type` softkey, then select `2 Level` softkey.
3. Select softkey of parameter that you want to change. Pressing Shift+Menu shows more softkey menus.
   - `Amplitude` sets the pulse amplitude. After selecting this softkey, you enter the value of pulse amplitude using numeric keys.
   - `Base level` sets the pulse base. After selecting this softkey, you enter the value of pulse base using numeric keys.
   - `Load` sets the load impedance. After selecting this softkey, you enter the value of load impedance using numeric keys.
   - `Delay` sets the delay time from the start of the pulse period to the start of the leading-edge of the pulse. After selecting this softkey, you enter the value of the delay time using numeric keys.

- `Width` sets the pulse width. After selecting this softkey, you enter the value of the pulse width using numeric keys.
- `Leading edge` sets the leading-edge transition time. After selecting this softkey, you enter the transition time using numeric keys.
- `Trailing edge` sets the trailing-edge transition time. After selecting this softkey, you enter the transition time using numeric keys.
- `Output ON/OFF` toggles the open/close state for the output switch of the PGU. This softkey appears only when a PGU that is *not* connected to an HF port is selected.
- `Connect Pin` sets the measurement pin numbers that you want to connect. After selecting this softkey, enter the measurement pin numbers using numeric keys.
- `Release Pin` sets the measurement pin numbers that you want to release. After selecting this softkey, enter the measurement pin numbers using numeric keys.

After setting parameters for the PGU, select `Return To Main` softkey.

For example, when the PG11 and PG12 force 2-level pulses, you set the parameters as follows:

```
************************** VFP for Pulse Generator ****************************
                                  Period: 20u     Number of Periods: 1

       Ampl  Base  Load    Delay  Width  LeEdg  TrEdg   Typ ON/OF HF Pin

PG11:  2.0   0     999k    0      10u    1u     1u      2l   on   5  18
PG12:  -2.0  0     999k    0      10u    1u     1u      2l   on   2  8
```

Hint

How To Enter Numeric Values

To enter a numeric value (for example, pulse width), you can use base-10 exponential expressions ($x.xx\mathrm{E}x$). Also, you can use the following prefixes: `k`, `m`, `u`, and `n`.

Entering 1.23×10^{-5} s	`1.23E-5` return
Entering 1.2 kΩ	`1.2E3` return
Entering 100 kΩ	`100k` return
Entering 15 ms	`15m` return
Entering 20 μs	`20u` return
Entering 500 ns	`500n` return

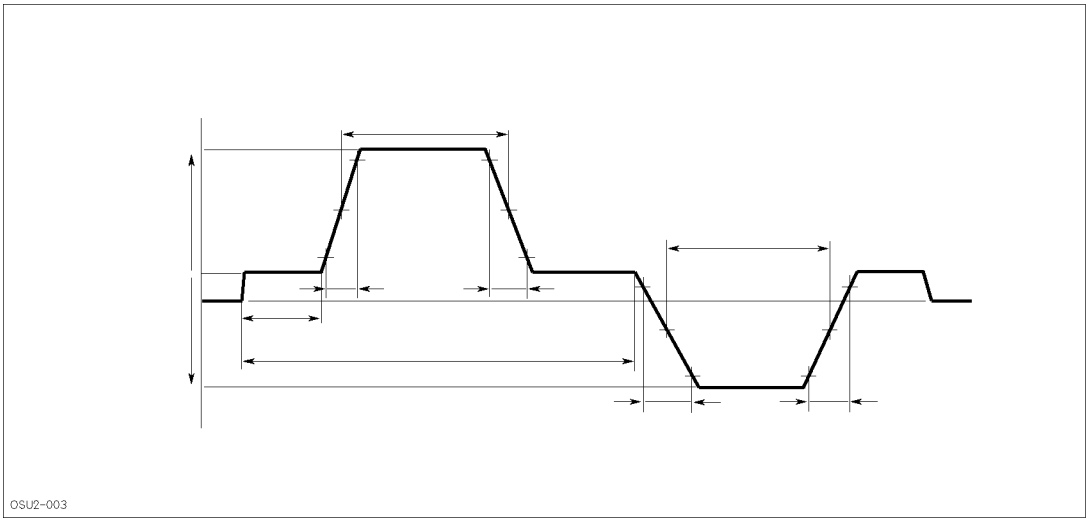


Figure 2-5. Setting "3-level Pulse with One PGU" Output

For 3-level pulse with one PGU, two pulses that have different pulse widths are output during a pulse period by using a PG equipped with two output units. The pulses are forced from the OUTPUT 1 terminal. The OUTPUT 2 terminal is disconnected from the internal circuit.

You set the pulse parameters for the 3-level pulse shown in Figure 2-5 by using the following procedure:

1. Select a softkey of an output unit 1 (PG*x*1). If the system has more than 8 available PGUs, other PGU softkeys are in the next softkey menu. To show next menu, press Shift+Menu key.
2. If the selected PGU is not "3L", press Shift+Menu keys, select `Type` softkey, then select `3 Level (no MUX)` softkey.

   PG*x*2, and its Typ, ON/OFF, HF, Pin fields become blank.
3. Select softkey of parameter that you want to change. The parameters that you can set are as follows:
   - `Amplitude 1` and `Amplitude 2` set the pulse amplitude. After selecting one of these softkeys, you enter the value of pulse amplitude using numeric keys.
   - `Base level` sets the pulse base. After selecting this softkey, you enter the value of pulse base using numeric keys.
   - `Load` sets the load impedance. After selecting this softkey, you enter the value of load impedance using numeric keys.
   - `Delay 1` and `Delay 2` set the delay time from the start of the pulse period to the start of the leading-edge of each pulse. After selecting one of these softkeys, you enter the value of the delay time using numeric keys.
   - `Width 1` and `Width 2` set the pulse width. After selecting one of these softkeys, you enter the value of the pulse width using numeric keys.

- `Leading Edge 1` and `Leading Edge 2` set the leading-edge transition time. After selecting one of these softkeys, you enter the transition time using numeric keys.
- `Trailing Edge 1` and `Trailing Edge 2` set the trailing-edge transition time. After selecting one of these softkeys, you enter the transition time using numeric keys.
- `Output ON/OFF` toggles the open/close state for the output switch of the PGU. This softkey appears only when a PGU that is *not* connected to a HF port is selected.
- `Connect Pin` sets the measurement pin numbers that you want to connect. After selecting this softkey, enter the measurement pin numbers using numeric keys.
- `Release Pin` sets the measurement pin numbers that you want to release. After selecting this softkey, enter the measurement pin numbers using numeric keys.

After setting parameters for the PGU, select `Return To Main` softkey.

For example, when PG11 forces 3-level pulses, you set the parameters as follows:

```
************************** VFP for Pulse Generator ****************************
                                  Period: 30u     Number of Periods: 1

       Ampl  Base  Load    Delay  Width  LeEdg  TrEdg   Typ ON/OF HF Pin

PG11:  2.0   0     999k    0      10u    1u     1u      3L   ON   5  18
       -1.5  0     999k    12u    10u    1u     1u
```

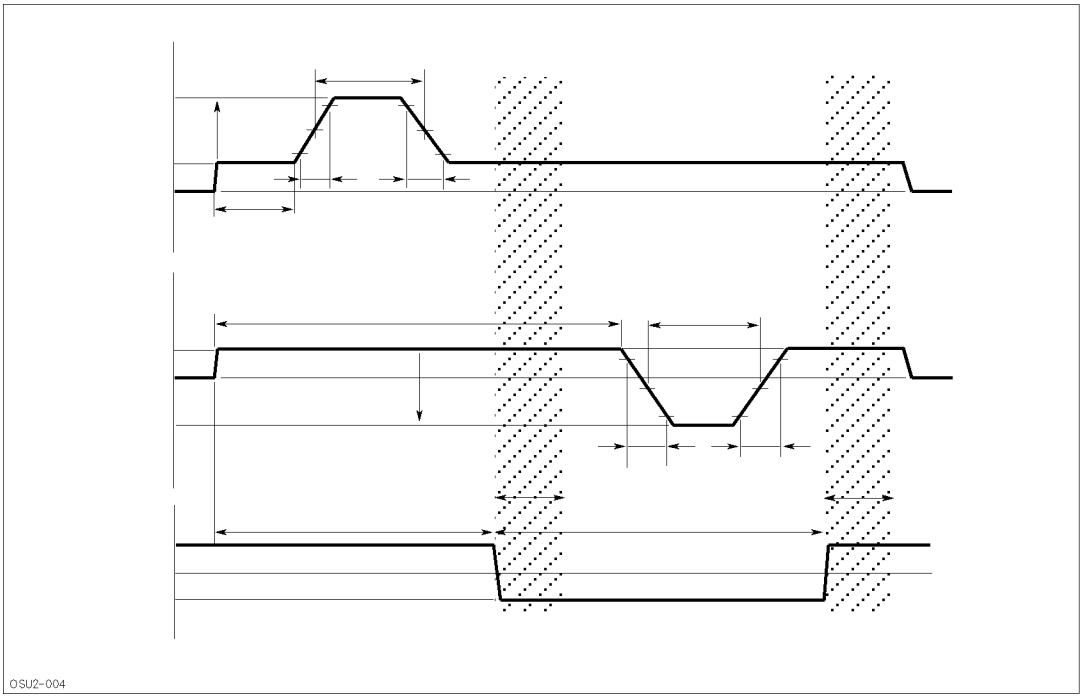


Figure 2-6. Setting the "3-level Pulse with a Multiplexer" Output

This method uses a multiplexer switch (which is controlled by a PGU) and two PGUs (which each forces a pulse with different timing). You set up the control PGU so that multiplexer switch alternates selecting the two PGUs. The 3-level pulses are forced to the DUT as shown in Figure 2-6.

The multiplexer control PGU automatically changes between +9.65 V and -9.65 V to alternately select the other two PGUs. Allow at least 2 ms between the first and second pulses, which is the time it takes for the multiplexer to stabilize after switching. Never force any pulse during this non-stable state.

Use the following procedure to set the pulse parameters for the "3-level pulse with a multiplexer switch":

1. Setting the multiplexer control PGU

   Note that the multiplexer control PGU must *not* be assigned to any HF ports in the FLCONFIG file. See "Configuration File /usr/pcs/etc/FLCONFIG" in Chapter 1.

   a. Select softkey of a PGU connected to the CTRL SW1·2·3 or SW4 connector on the cable adapter for controlling the multiplexer. If the system has more than 8 available PGUs, other PGU softkeys are in the next softkey menu. To show next menu, press Shift+Menu key.

   b. If the selected PGU is not "R" type, press Shift+Menu key, select `Type` softkey, then select `Relay Control` softkey.

   c. Select the softkey of the parameter you want to change, then enter the value. The parameters that you can set are as follows:

- `Delay` sets the delay time from the start of the pulse period to the start for changing the multiplexer switch. After selecting this softkey, you enter the value of the delay time using numeric keys.
- `Width` sets the pulse width, which is time from the start for changing the multiplexer switch after the first pulse output to the start for changing the multiplexer switch after the second pulse output. After selecting this softkey, you enter the value of the pulse width using numeric keys.

2. Setting the PGU for the first pulse

   Note that the PGU that forces the first pulse must be assigned to an HF port in the FLCONFIG file. See "Configuration File /usr/pcs/etc/FLCONFIG" in Chapter 1.

   a. Select softkey of PGU connected to the In(A) terminal of the multiplexer switch (SW1 or SW4). If the system has more than 8 available PGUs, other PGU softkeys are in the next softkey menu. To show next menu, press Shift+Menu key.

   b. If the selected PGU is not type "M*xx*", press Shift+Menu key, select `Type` softkey, then select `3 Level (w/ MUX)` softkey.

   c. Select softkey of PGU connected to the In(B) terminal of the multiplexer switch (SW1 or SW4). If the system has more than 8 available PGUs, other PGU softkeys are in the next softkey menu. To show next menu, press Shift+Menu key.

   d. Select softkey of parameter you want to change, then enter the value. The parameters that you can set are as follows:

      - `Amplitude` sets the pulse amplitude 1. After selecting this softkey, you enter the value of pulse amplitude using numeric keys.
      - `Base level` sets the pulse base. After selecting this softkey, you enter the value of pulse base using numeric keys.
      - `Load` sets the load impedance. After selecting this softkey, you enter the value of load impedance using numeric keys.
      - `Delay` sets the delay time 1 from the start of the a pulse period to the start of the leading-edge of the first pulse. After selecting this softkey, you enter the value of the delay time using numeric keys.
      - `Width` sets the pulse width 1. After selecting this softkey, you enter the value of the pulse width using numeric keys.
      - `Leading Edge` sets the leading-edge transition time 1. After selecting this softkey, you enter the transition time using numeric keys.
      - `Trailing Edge` sets the trailing-edge transition time 1. After selecting this softkey, you enter the transition time using numeric keys.
      - `Connect Pin` sets the measurement pin numbers that you want to connect to an HF port. After selecting this softkey, enter the measurement pin numbers using numeric keys.
      - `Release Pin` sets the measurement pin numbers that you want to release. After selecting this softkey, enter the measurement pin numbers using numeric keys.

3. Setting the PGU for the second pulse

   Note that the PGU that forces the second pulse must *not* be assigned to an HF port in the FLCONFIG file. See "Configuration File /usr/pcs/etc/FLCONFIG" in Chapter 1.

   a. Select softkey of PGU connected to the In(B) terminal of the multiplexer switch (SW1 or SW4), and that is M*xx* type. If the system has more than 8 available PGUs, other PGU softkeys are in the next softkey menu. To show next menu, press Shift+Menu key.

   b. Select softkey of parameter that you want to change, then enter the value. The parameters that you can set are same for the first pulse.

For example, in the following example, PG11 controls the multiplexer switch, PG21 forces the first pulse, and PG22 forces the second pulse.

```
************************** VFP for Pulse Generator ****************************
                                  Period: 30m     Number of Periods: 1

       Ampl  Base  Load    Delay  Width  LeEdg  TrEdg   Typ ON/OF HF Pin

PG11:                      10m    13m                    R    ON
PG12:  .2    0     999k    0      500n   20n    20n     2L   OFF  1

PG21:  2.0   0     999k    0      10m    1u     1u      M22  ON   4  16
PG22:  -2.0  0     999k    13m    10m    1u     1u      M21  ON
```

Forcing the Pulse

There are two methods to force the pulse: repeating the pulse the specified times, and forcing the pulse continuously.

- Repeating the pulse the specified times

  Select `GO` softkey. Then, the pulse is forced the number of times specified in the `Number of Periods` field, and pulses are forced with the specified `Period`.

- Forcing the pulse continuously

  Select `No. of Periods` softkey, then enter 0 (zero). The pulses will be forced continuously with the specified `Period`.

  To stop the pulse output, select `No. of Periods` softkey, then enter a positive number other than 0 (zero).

Initializing HP 4062F System

Select `Initialize` softkey to initialize the HP 4062F system and to be the same state as when the VFP.AC program starts.

Storing the Pulse Setting Data

You can store the pulse setting data in a file, and load the data later.

When you select `File` softkey, the following softkey menu appears:

- To store the setting data in a file

  Select `Store Setup` softkey, then enter file name. If the specified file name exists, select `Re store Setup` softkey.

- To load the setting data from a file

  Select `Load Setup` softkey, then enter the file name.

- To list names of files that are in the present directory

  Select `CAT` softkey.

- To load the pulse setting file in a measurement program and force pulse

  Use Monitor_pg subprogram. For example, the following program forces pulse according to data stored in the specified file:

```
 . . . . . .
1230    Monitor_pg("setup.1011")
 . . . . . .
```

3

Measurement Programming

This chapter provides the guidelines about developing measurement programs and explains the contents of the measurement programs.

Developing and Executing Measurement Programs

Flow for Developing the Measurement Programs

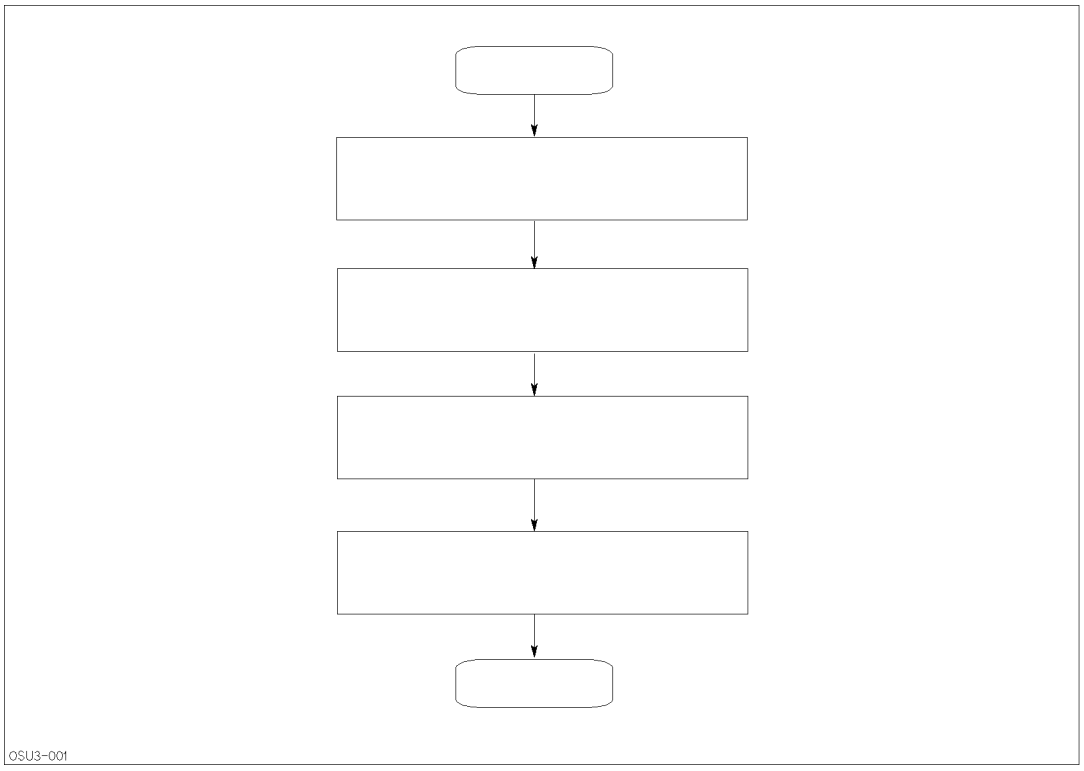


Figure 3-1. Flow for Developing the Measurement Programs

You develop measurement programs in a way similar to the development flow shown in Figure 3-1.

Planning the Test

Decide what device parameters you want to measure, and under what conditions you execute the measurements.

For example, when examining the relation between the number of write/erase cycles and the threshold voltage for the flash

	memory cells, you decide the way to measure the threshold voltage versus the pulse voltage, pulse width, rise time, and fall time for both writing and erasing.
Making a Chart of Test Sequence	You make the test sequence into a chart. For example, you create a flow chart for measurement program.
Programming	Create programs according to the program flow.
Debugging the Programs	Check that the programs work correctly. If not, debug the programs.

Measurement program development flow shown here is only an example. There are many development methods. You are responsible for developing the measurement programs.

Evaluating the Flash Memory Cell

Flash memory cell is a MOSFET structure that has a floating gate between the substrate and gate of MOSFET. See Figure 3-2. This memory cell is written by injecting electrical carriers (mainly electrons) into the floating gate. And is erased by emission of electrical carriers from the floating gate.

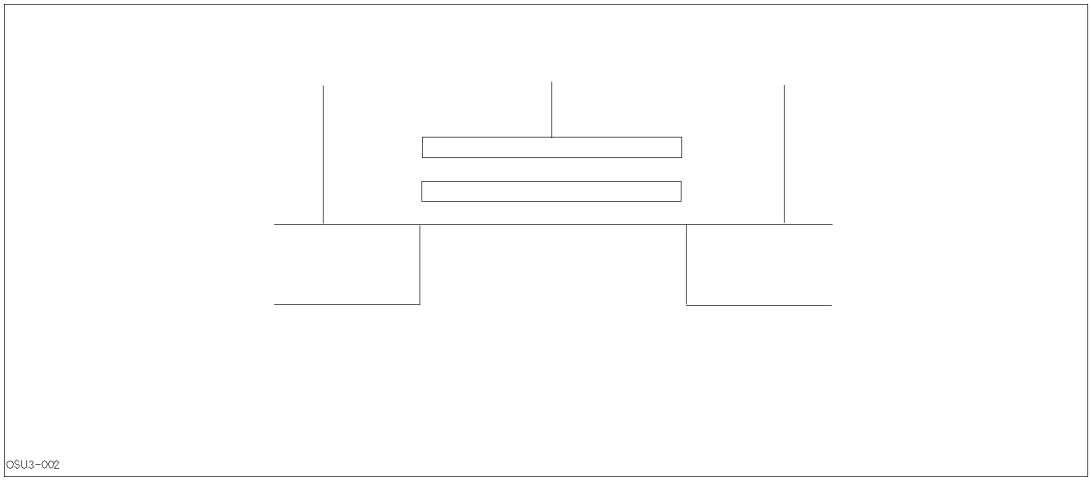


Figure 3-2. Structure of Flash Memory cell

For example, you evaluate the memory cells by measuring the threshold voltage (Vth) for the following purposes:

- To decide the proper pulse width for writing and erasing.
- To investigate the endurance of memory cell by repetitive writing and erasing.

The following example explains the relation between the write-erase cycles and the threshold voltage (Vth) of an N-channel memory cell.

For writing, negative carriers (electrons) are injected into the floating gate, then positive electric carriers gather at surface of the gate oxide layer. To make a channel near the surface of this layer for passing the electrons, higher voltage is forced to the gate. The voltage value to make this channel is called the threshold voltage (Vth), and threshold voltage becomes higher after writing.

For erasing, electrons are emitted from the floating gate, so the electric carriers are not gathered near the surface of the gate oxide layer. To make a channel near the surface of this layer for passing the electrons, a lower voltage is forced to the gate. The threshold voltage becomes lower after erasing.

Figure 3-3 shows measurement examples for pulse width dependency and endurance of the flash memory cells. The example shows that pulse width of more than 100 ms (1E−1 s) writes and erases properly. So you can decide the proper pulse width. The example also shows that the cell starts to deteriorate after 100,000 (1E5) write-erase cycles.

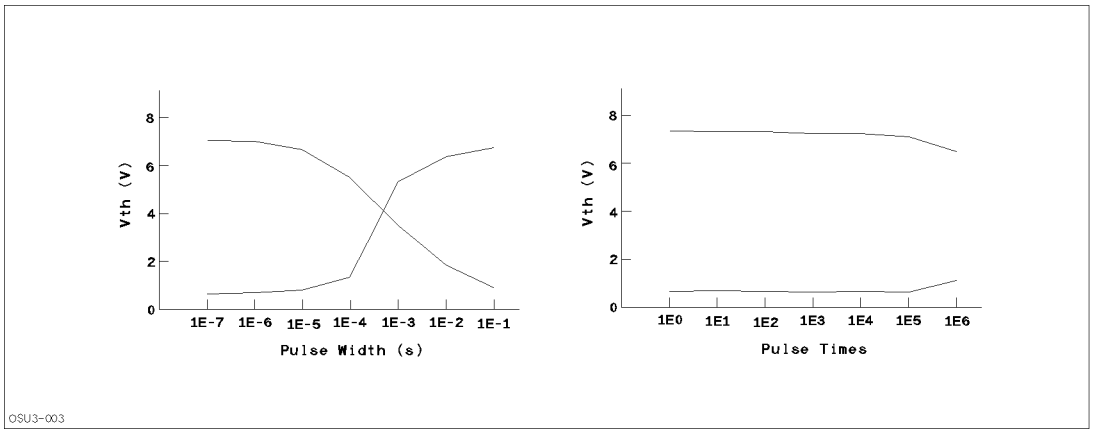


Figure 3-3. Evaluation Examples of Flash Memory Cells

Developing Measurement Programs

You develop the measurement programs in the HP BASIC/UX programming language environment. See the HP BASIC/UX manuals about the programming techniques for HP BASIC/UX.

See the chapter 3, "Measurement Programming", of the HP 4062C/UX *System Operation* manual about basic programming for the measurement programs.

You must set up the necessary environment for the HP 4062F by executing the START.F program before executing your measurement programs. See "START.F Program" in Chapter 2 about how to execute the START.F program.

Linking Libraries

Before executing your measurement programs, you must link necessary library files such as TIS.F and PARA.F.

1. Select `Link TIS` softkey, or execute the following command to link the TIS.F file:

   `LOADSUB ALL FROM "/usr/pcs/lib/TIS.F"` Return

2. Execute the following command to link necessary library file:

   `LOADSUB ALL FROM "/usr/pcs/lib/`*filename*`"` Return

   Where, *filename* means a library file containing necessary subprograms such as PARA.F, PCL, and FCL.

If you execute your measurement program without linking the TIS.F file, the COM blocks created by the START.F program are destroyed and the following message is displayed:

```
ERROR 7 IN 110 Undefined function or sub
```

Even if you link the TIS.F file after the error was reported, your measurement program cannot work correctly. You must re-execute the START.F program to reconstruct the COM blocks, then execute your measurement programs.

Storing Measurement Programs

When storing your measurement programs in files, use store command after unlinking the TIS.F and other subprograms. If you store your measurement programs with TIS.F and other subprograms, they waste disk space.

`Unlink & Store` **softkey.** This softkey is used to unlink TIS.F and other subprograms from your measurement program, then store program in a file. To show this softkey, press Shift+Menu key a few times because this softkey is assigned to function key 19.

Selecting the `Unlink & Store` softkey deletes (from HP BASIC/UX memory) program lines from beginning of TIS.F to end of your measurement program, and displays the following prompt:

```
RE-STORE ""
```

Type file name for the measurement program, then press Return key. This stores measurement program lines (but not the linked libraries) in the file.

Unlinking and storing manually. If libraries are linked *before* TIS.F, use the following command to store your measurement program.

1. Delete from desired subprogram to end of program using the DELSUB command.

   `DELSUB` *subprogram name* `TO END` Return

2. Use the RE-STORE command to store the measurement program.

   `RE-STORE "`*filename*`"` Return

Programming with PARA.F

PARA.F file includes some useful subprograms for writing and erasing to flash memories.

These can be categorized into four types of subprograms:

A. Subprograms to assign terminals of memory cell to measurement pins.

B. Subprograms to execute write once.

C. Subprograms to execute erase once.

D. Subprograms to repeat write and erase.

Category A has two subprograms, Set_fr and Set_multifr, to assign terminals of memory cell to measurement pins. Set_fr subprogram assigns one set of cell terminals, and Set_multifr subprogram assigns several sets of cell terminals. Set_relay_mode subprogram defines use of open/close and multiplexer switches.

Categories B to C have write and erase subprograms depending on the cell type: NAND, NOR, and low-power NOR. See Table 3-1.

Table 3-1. Subprograms for Write/Erase/Repeat in PARA.F

	Write	Erase	Repeat
NAND	Write_nand1	Erase_nand1	Repeat_nand1
NOR	Write_nor1	Erase_nor1	Repeat_nor1
Low-power NOR	Write_nor2, Write_nor3[1]	Erase_nor2, Erase_nor3[1]	Repeat_nor2, Repeat_nor3[1]

1 These subprograms use multiplexer switch to write and erase.

For details of each subprogram, see Chapter 4.

Measurement Programs with PARA.F

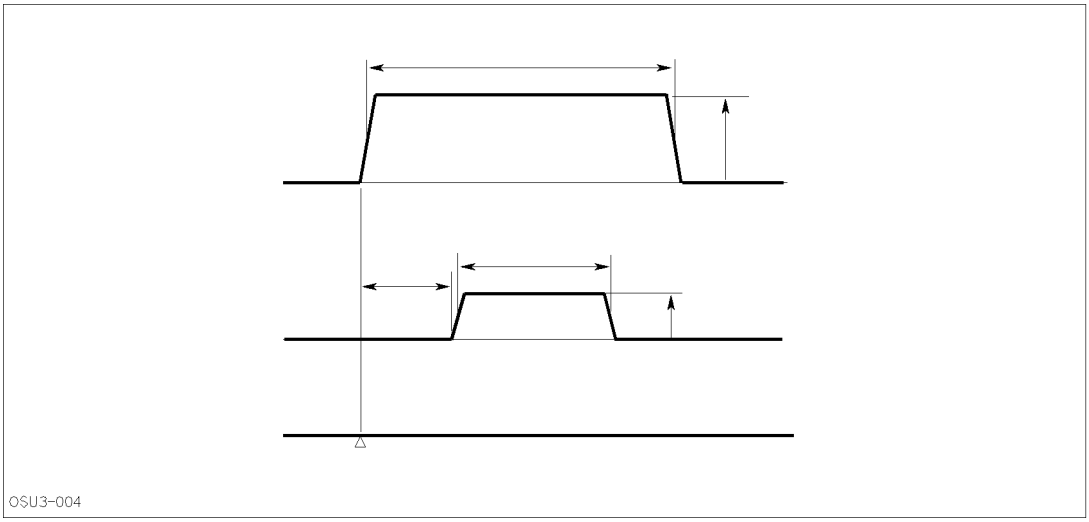
This section shows an example to measure threshold voltage after writing in the memory cell. The program writes in the memory cell with Write_nor1 subprogram, then measures the threshold voltage (Vth) with FNVth3 subprogram. The memory cell is assumed to be normal NOR type.

Figure 3-4 shows a program example. To execute this program, you must execute the following two commands to link necessary library files:

```
LOADSUB ALL FROM "/usr/pcs/lib/TIS.F"
LOADSUB ALL FROM "/usr/pcs/lib/PARA.F"
```

Make sure that you do not use the wrong file names because similar names exist (/usr/pcs/lib/TIS and /usr/pcs/lib/PARA) for the HP 4062UX system.

The example program in Figure 3-4 writes by forcing the following pulse to the memory cell.




```
10   ! Write and measure Vth
20   !
30   ! LOADSUB ALL FROM "/usr/pcs/lib/TIS.F"
40   ! LOADSUB ALL FROM "/usr/pcs/lib/PARA.F"
50   !
60   INTEGER Source,Gate,Drain,Subs
70   REAL Ampl(1:2),Width(1:2),Delay(1:2),Edge1(1:2),Edge2(1:2)
80   !
90   Source=4
100  Gate=8
110  Drain=12
120  Subs=16
130  Ampl(1)=12.
140  Ampl(2)=6.
150  Width(1)=1.E-5
160  Width(2)=5.E-6
170  Delay(1)=0
180  Delay(2)=3.E-6
190  Edge1(1)=1.E-7
200  Edge1(2)=1.E-7
210  Edge2(1)=1.E-7
220  Edge2(2)=1.E-7
230  !
240  Init_system
250  Init_pg
260  !
270  Set_fr(Source,Gate,Drain,Subs)
280  Write_nor1(Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
290  !
```

Figure 3-4. Program Example: Programming with PARA.F

```
300  Set_tr(Source,Gate,Drain,Subs)
310  Vth=FNVth3(Id_set,Vds,10.,0)
320  PRINT Vth;" V"
330  !
340  END
```

Program Description.

lines 60 and 70: Defines the type of variables used in the program.

lines 90 to 120: Assigns the measurement pins corresponding to the source, gate, drain, and substrate terminals of memory cell.

lines 130 and 140: Sets the pulse amplitude for gate and drain.

lines 150 and 160: Sets the pulse width for gate and drain.

lines 170 and 180: Sets the pulse delay time for gate and drain.

lines 190 and 200: Sets the leading-edge transition time of pulse for gate and drain.

lines 210 and 220: Sets the trailing-edge transition time of pulse for gate and drain.

lines 240 and 250: Initializes the system.

line 270: Sets the assignment between terminals of memory cell and measurement pins.

line 280: Connects pins, then forces a write pulse to the memory cell.

line 300: Sets the assignment between terminals of memory cell and measurement pins.

lines 310 and 320: Measures the threshold voltage (Vth) and prints the result.

Program Example: Programming with PARA.F (continued)

Programming with TIS.F

Subprograms in TIS.F

TIS.F file includes subprograms in the TIS file of HP 4062C/UX system, plus subprograms for evaluating flash memories. You can measure the device parameters of the flash memory cells using the measurement functions of the HP 4062C/UX system.

Also, pulse generator is necessary to write and erase the flash memory cells. And a switching matrix (SWM) for high-frequency signals is also necessary because the write and erase require pulses that have short rise and fall time. The TIS.F file also includes subprograms to control the pulse generator and high-frequency SWM.

The following briefly explains the subprograms in the TIS.F file to control the pulse generator and high-frequency SWM. See the HP 4062C/UX *System Operation* manual about other subprograms.

- System initialization

Init_pg	Initializes the system variable area for the pulse generators. Init_system initializes the system variable area for dc and capacitance measurements.

- Pin board control in SWM

Connect_hf	Connects and disconnects between a port and at least one measurement pin.
Connect_hf_th	Connects and disconnects between a port and a series of measurement pins.
FNHf	Returns port address that corresponds to specified HF port number.
FNPort	Returns port address that corresponds to specified SWM port number.

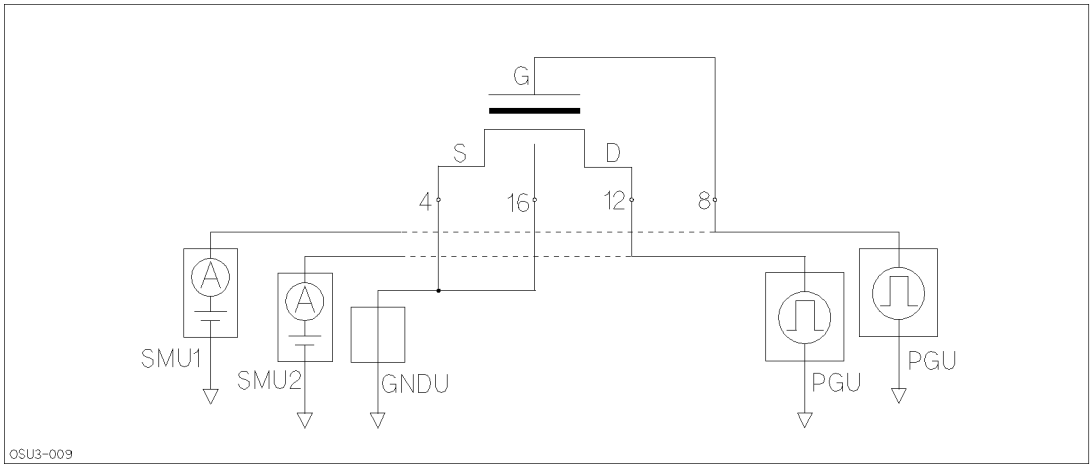
- Pulse generator control

Set_mode_pg	Defines normal or pattern mode.
Set_type_pg	Defines 2-level or 3-level pulse output of output wave form.
Set_level_pg	Sets the pulse amplitude and pulse base of output pulse.
Set_time_pg	Sets the time-dependent parameters of output pulse.
Force_pg	Triggers and forces the specified pulses.
Set_sr_pg	Sets the control parameters for multiplexer and open/close switches.
Ch_sw_on_pg	Closes the output switch of the specified PGU.
Ch_sw_off_pg	Opens the output switch of the specified PGU.
Error_info_pg	Returns error messages that occurred in the HP 8110A.
Connect_sr	Changes state of the multiplexer and open/close switches.
FNPg	Returns unit address corresponding to the unit number of the PGU.

Connections between Pin and Port

You use Connect_hf or Connect_hf_th subprogram to connect or disconnect between the measurement pins and measurement ports. The Connect_hf subprogram connects or disconnects one port to several pins (maximum 12 pins), and the Connect_hf_th subprogram connects or disconnects one port and a series of pins. Note that you cannot connect one measurement pin to multiple ports.

Figure 3-5 shows a example program for the Connect_hf subprogram. This program connects the HF ports to the device under test (DUT) to force the write pulse, then connects the SMU ports to the DUT to measure the threshold voltage (Vth).




```
10   INTEGER Source,Gate,Drain,Subs
20   !
30   Source=4
40   Gate=8
50   Drain=12
60   Subs=16
70   !
80   Init_system
90   Init_pg
100  !
110  Connect_hf(FNGnd,Source,Subs)
120  Connect_hf(FNHf(2),Gate)
130  Connect_hf(FNHf(1),Drain)
140  Write_pulse
150  Connect_hf
160  !
170  Connect_hf(FNGnd,Source,Subs)
180  Connect_hf(FNSmu(1),Gate)
```

Figure 3-5. Example Program with Connect_hf

```
190  Connect_hf(FNSmu(2),Drain)
200  Measure_vth
210  Connect_hf
220  !
230  END
240  !
250  SUB Write_pulse
260  SUBEND
270  !
280  SUB Measure_vth
290  SUBEND
```

Program Description.

line 10:	Defines the type of variables used in the program.
lines 30 to 60:	Sets measurement pin numbers that are connected to DUT.
lines 80 and 90:	Initializes the system.
line 110:	Connects source and substrate to the GNDU.
line 120:	Connects gate to a PGU.
line 130:	Connects drain to a PGU.
line 140:	Calls subprogram for write.
line 150:	Disconnects.
line 170:	Connects source and substrate to the GNDU.
line 180:	Connects gate to the SMU1.
line 190:	Connects drain to the SMU2.
line 200:	Calls subprogram for measuring the Vth.
line 210:	Disconnects.
lines 250 and 260:	Subprogram for write.
lines 280 and 290	Subprogram for measuring the Vth.

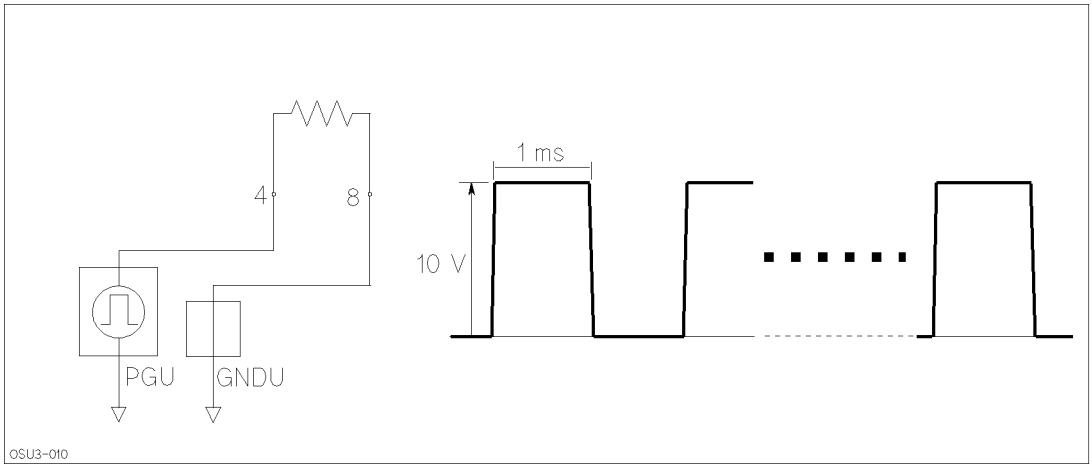
Example Program with Connect_hf (continued)

2-level Pulse Output

2-level pulse is forced from a PGU. If a PG is equipped with two output units (PGUs), each output unit can force pulse.

Set_type_pg, Set_level_pg, Set_time_pg, and Force_pg subprograms are used to force a 2-level pulse.

Figure 3-6 shows an example program to force 2-level pulse. This program forces a pulse to the DUT (resistor) ten times. The pulse has an amplitude of 10 V and a pulse width of 1 ms.




```
10   INTEGER High,Low                    Declares variable type.
20   !
30   High=4
40   Low=8
50   !
60   Init_system
70   Init_pg
80   !
90   Connect_hf(FNHf(1),High)
100  Connect_hf(FNGnd,Low)
110  Set_type_pg(High,1)                 Sets 2-level pulse mode.
120  Set_level_pg(High,10.)              Sets 10 V for pulse amplitude.
130  Set_time_pg(High,1.E-3)             Sets 1 ms for pulse width.
140  Force_pg(10.,2.E-3)                 Forces 10 pulses with period of 2 ms.
150  Connect_hf
160  !
170  END
```

Figure 3-6. Example Program of 2-Level Pulse Output

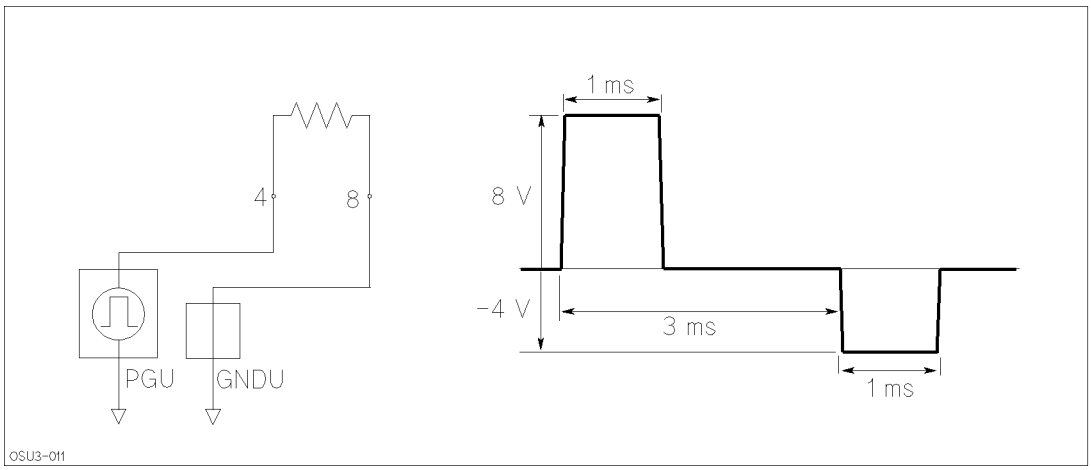
3-Level Pulse Output with One PG

You can force 3-level pulse output with one PG. Note that the PG that forces a 3-level pulse must equipped with two output units.

When a PG forces a 3-level pulse, two output pulses are combined in the PG, then output from OUTPUT 1 terminal. The OUTPUT 2 terminal is disabled. So, when specifying the PGU that forces a 3-level pulse, such as in the Set_type_pg subprogram, you use the port or pin number connected to the OUTPUT 1 terminal of PG.

Set_type_pg, Set_level_pg, Set_time_pg, and Force_pg subprograms are used to force a 3-level pulse.

Figure 3-7 shows an example program to force a 3-level pulse.




```
10   INTEGER High,Low
20   REAL Ampl1,Ampl2,Vbase
30   REAL Width1,Delay1,Ld1,Tl1,Width2,Delay2,Ld2,Tl2
40   !
50   High=4
60   Low=8
70   !
80   Init_system
90   Init_pg
100  !
110  Connect_hf(FNHf(1),High)
120  Connect_hf(FNGnd,Low)
130  Set_type_pg(High,2)
140  Ampl1=8
150  Ampl2=-4
160  Vbase=0
170  Set_level_pg(High,Ampl1,Ampl2,Vbase)
```

Figure 3-7. Example Program of 3-Level Pulse Output with One PG

```
180  Width1=1.E-3
190  Delay1=0
200  Ld1=5.E-5
210  Tl1=5.E-5
220  Width2=1.E-3
230  Delay2=3.E-3
240  Ld2=5.E-5
250  Tl2=5.E-5
260  Set_time_pg(High,Width1,Delay1,Ld1,Tl1,Width2,Delay2,Ld2,Tl2)
270  Force_pg(1.)
280  Connect_hf
290  !
300  END
```

Program Descriptions.

lines 10 to 30:	Defines the type of variables used in the program.
lines 50 and 60:	Assigns measurement pins to variables. These are pins that are connected to DUT (resistor).
lines 80 and 90:	Initializes the system.
lines 110 and 120:	Connects HF1 port and GNDU port to measurement pins of DUT.
line 130:	Sets 3-level pulse mode.
lines 140 to 160:	Assigns the values of pulse amplitude and pulse base to the variables.
line 170:	Sets the pulse amplitude and pulse base.
line 180 to 250:	Assign the values of pulse width, delay time, and transition times to the variables.
line 260:	Sets the pulse width, delay time, and transition times.
line 270:	Forces a pulse.
line 280:	Disconnects the connections between measurement ports and measurement pins.

Example Program of 3-Level Pulse Output with One PG (continued)

3-Level Pulse Output with a Multiplexer

You can force 3-level pulse output by using a multiplexer switch (SW1 or SW4) of the cable adapter. The following describes how to force 3-level pulse by using the multiplexer switch.

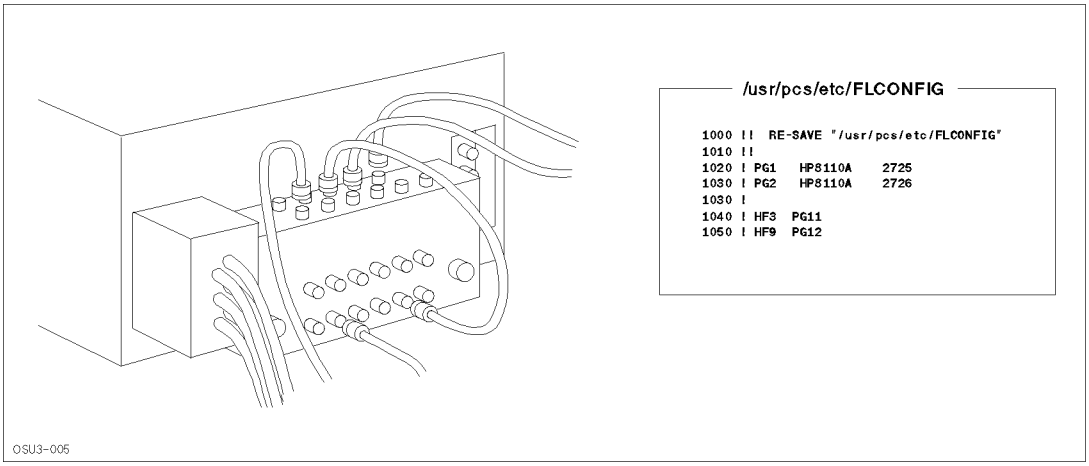
SW1 is a semiconductor relay, so it has a long life. SW4 is a reed relay, which you should use if you need to pass higher frequency pulses. Note that it requires at least 2 ms of wait time to switch the multiplexer switch.

Use the following procedure to connect the multiplexer switch:

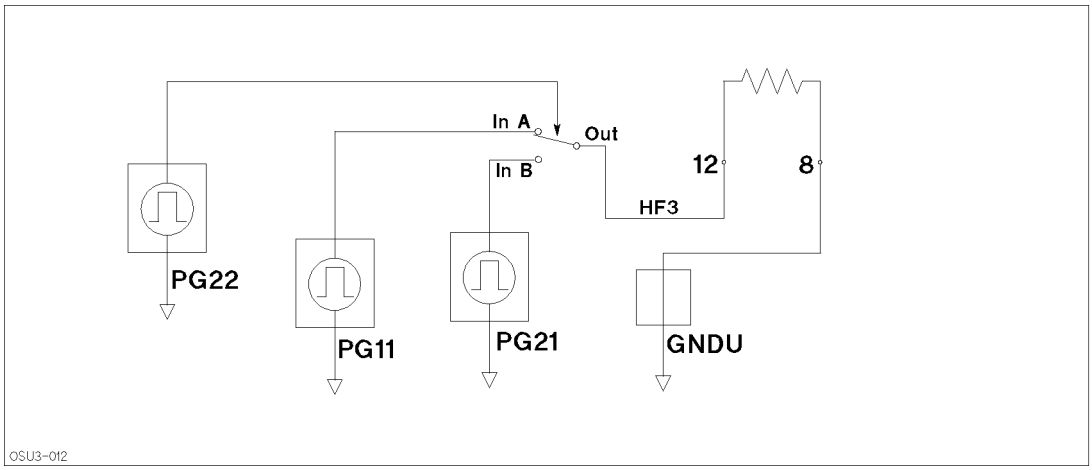
- Connect the output terminal of one PGU to In A (NC) terminal of SW1 or SW4.
- Connect the output terminal of other PGU to In B (NC) terminal of SW1 or SW4.
- Connect the Out terminal of SW1 or SW4 to an HF port.
- Connect the output terminal of the control PGU to CTRL SW1·2·3 or SW4.

Note that the /usr/pcs/etc/FLCONFIG file should include HF port assignment of the PGU connected to the input A terminal. It must *not* include HF port assignment of PGUs connected to the input B terminal and control terminal.

In the following example, PG11 and PG21 are connected to the input A and B terminals of the SW1, respectively. The output terminal of SW1 is connected to the HF3 port, and PG22 is connected to the CTRL SW1·2·3 terminal for switch control. In the /usr/pcs/etc/FLCONFIG file, PG11 is assigned to HF3, and PG21 and PG22 should *not* be assigned to any HF port.



This measurement circuit is shown as follows:




```
10   INTEGER High1,High2,Low,Ctrl
20   REAL Ampl1,Ampl2,Vbase
30   REAL Width1,Delay1,Ld1,Tl1,Width2,Delay2,Ld2,Tl2
40   !
50   High1=12
60   Low=8
70   High2=FNPg(21)
80   Ctrl=FNPg(22)
90   !
100  Init_system
110  Init_pg
120  !
130  Connect_hf(FNHf(3),High1)
140  Connect_hf(FNGnd,Low)
150  Set_type_pg(High1,3,High2)
160  Set_sr_pg(Ctrl,2,5.E-3,2.5E-3)
170  Ampl1=8
180  Ampl2=-4
190  Vbase=0
200  Set_level_pg(High1,Ampl1,Ampl2,Vbase)
210  Width1=2.E-3
220  Delay1=0
230  Ld1=1.E-6
240  Tl1=1.E-6
250  Width2=2.E-3
260  Delay2=5.E-3
270  Ld2=1.E-6
280  Tl2=1.E-6
290  Set_time_pg(High1,Width1,Delay1,Ld1,Tl1,Width2,Delay2,Ld2,Tl2)
```

Figure 3-8. Example Program of 3-Level Pulse Output with Multiplexer

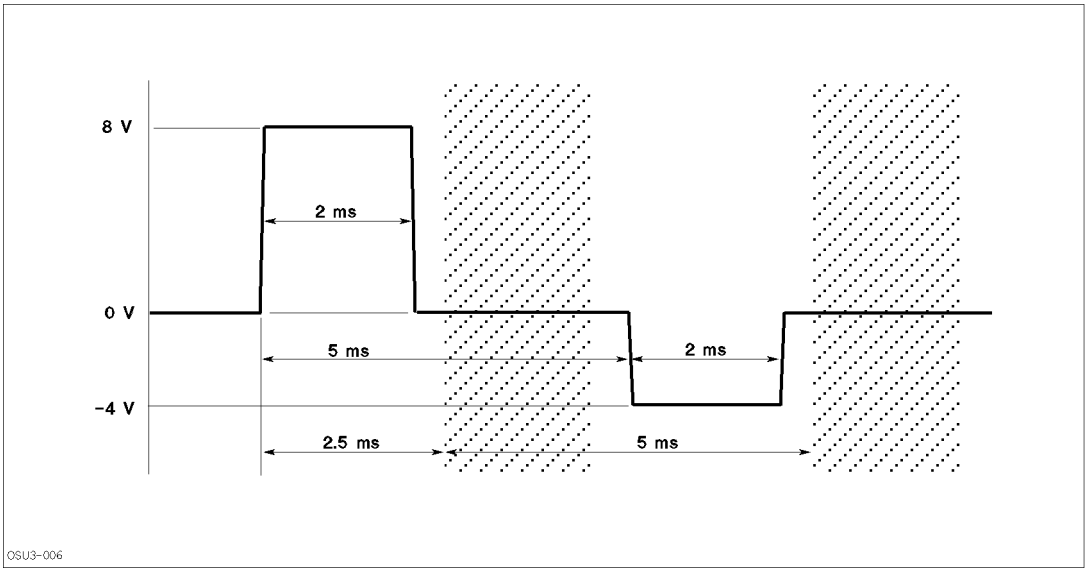
```
300  !
310  Force_pg(100.,1.E-2)
320  !
330  Connect_hf
340  !
250  END
```

Program Description.

line 10 to 30:	Defines the type of variables used in the program.
lines 50 and 60:	Assigns measurement pins to variables. These are pins that are connected to DUT (resistor).
lines 70 and 80:	Assigns PGU unit addresses to variables. These are unit addresses of PGUs connected to input B terminal and to the control terminal, respectively.
lines 100 and 110:	Initializes the system.
lines 130 and 140:	Connects the HF3 port and GNDU port to the measurement pins of DUT.
line 150:	Sets the output mode to 3-level pulse with multiplexer.
line 160:	Specifies the PGU for controlling the switch, and sets the timing for changing the multiplexer switch.
lines 170 to 190:	Assigns the values of pulse amplitude and pulse base to the variables.
line 200:	Sets the pulse amplitude and pulse base.
lines 210 to 280:	Assigns the values of pulse width, delay time, and transition times to the variables.
line 290:	Sets the pulse width, delay time, and transition times.
line 310:	Forces 100 pulses with pulse period of 10 ms.
line 280:	Disconnects the connections between measurement ports and measurement pins.

Example Program of 3-Level Pulse Output with Multiplexer (continued)

Executing this program forces the pulse wave shown in the following figure to the DUT 100 times:



The Set_sr_pg subprogram specifies the PGU used for switch control, and sets the timing to switch the multiplexer switch. Immediately after this subprogram is executed, the control PGU forces a voltage to set the switch to input A of the multiplexer. The Force_pg subprogram starts the pulse output, and the multiplexer changes connections between input A and input B according to the timing set up by Set_sr_pg subprogram.

You can also switch the multiplexer without specifying the switching timing. To do so, you use Connect_sr subprogram. For details, see "Connect_sr" in Chapter 4.

Disconnecting Output Port of PG

A DUT terminal can be set to the open state by using the open/close switch of the cable adapter. SW2 and SW3 are open/close switches on the cable adapter. The following are programming techniques on how to open the switch and how to disconnect the DUT terminal from the PG output port.

These switches are semiconductor relays, which have a long life. Note that it requires at least 2 ms of wait time to switch the open/close switch.

Use the following procedure to connect the multiplexer switch:

- Connect the output terminal of the PGU to In terminal of SW2 or SW3.
- Connect the Out terminal of SW2 or SW3 to an HF port.
- Connect the output terminal of the control PGU to the CTRL SW1·2·3.

In other words, an open/close switch is inserted between a PGU and an HF port. Note that the /usr/pcs/etc/FLCONFIG file must include the HF port assignment of the PGU that is connected to the input terminal. Do not include HF port assignment for control PGU.

Also, you can set the default status of the open/close switch by using the slide switches labeled "MODE SELECTOR". If you set the slide switch to NC (Normally Closed), closed status is default, and if to NO (Normally Open), open status is default. "Default status" means the initial status of the open/close switch immediately after the Set_sr_pg subprogram is executed. It does *not* mean the initial status after Init_pg subprogram is executed.

In the following example, PG12 is connected to the input terminal of SW3, output terminal of SW3 is connected to HF3 port, and PG11 is connected to the CTRL SW1·2·3 terminal for switch control. In the /usr/pcs/etc/FLCONFIG file, PG12 must be assigned to HF3, and PG11 must *not* be assigned to any HF port.

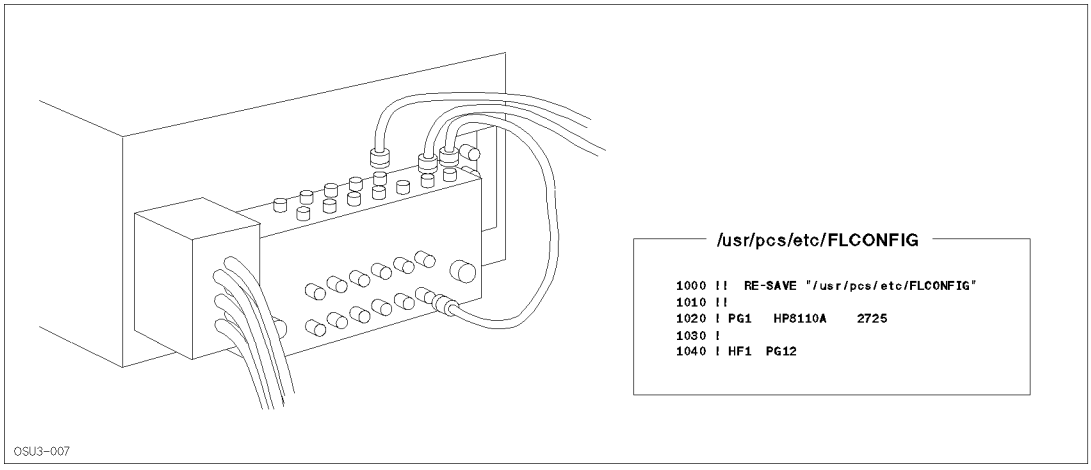
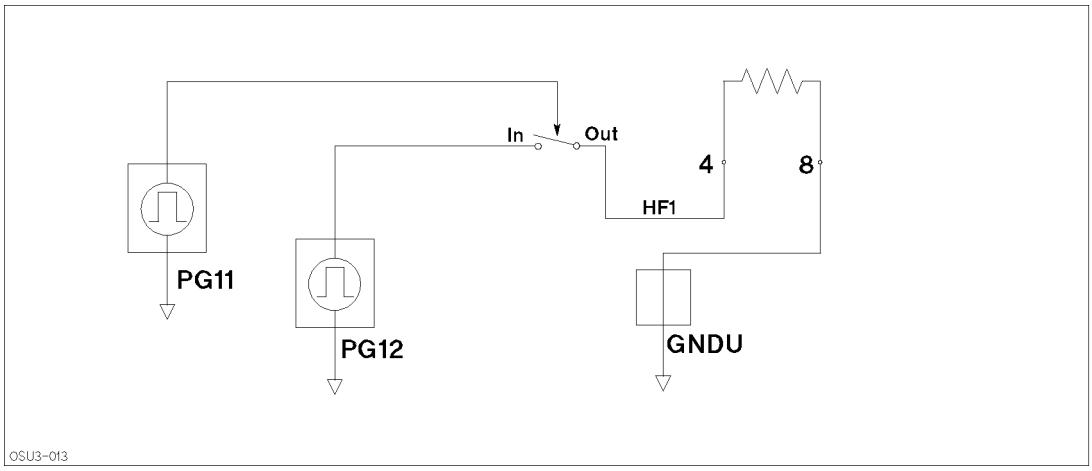


Figure 3-9 shows an example program to control the open/close switches.




```
10   !
20   ! SW3's MODE SELECTOR must be at NC position.
30   !
40   INTEGER High,Low,Ctrl
50   REAL Open_time,Close_time,Ampl
60   !
70   High=4
80   Low=8
90   Ctrl=FNPg(11)
100  !
110  Init_system
120  Init_pg
130  !
140  Connect_hf(FNHf(1),High)
150  Connect_hf(FNGnd,Low)
160  Close_time=.004
170  Open_time=.005
180  Set_sr_pg(Ctrl,2,Open_time,Close_time)
190  Ampl=5
200  Set_level_pg(High,Ampl)
210  Width=.003
220  Delay=0
230  Ld_edge=.0001
240  Tl_edge=.001
250  Set_time_pg(High,Width,Delay,Ld_edge,Tl_edge)
```

Figure 3-9. Example Program of PG Output Port Disconnection

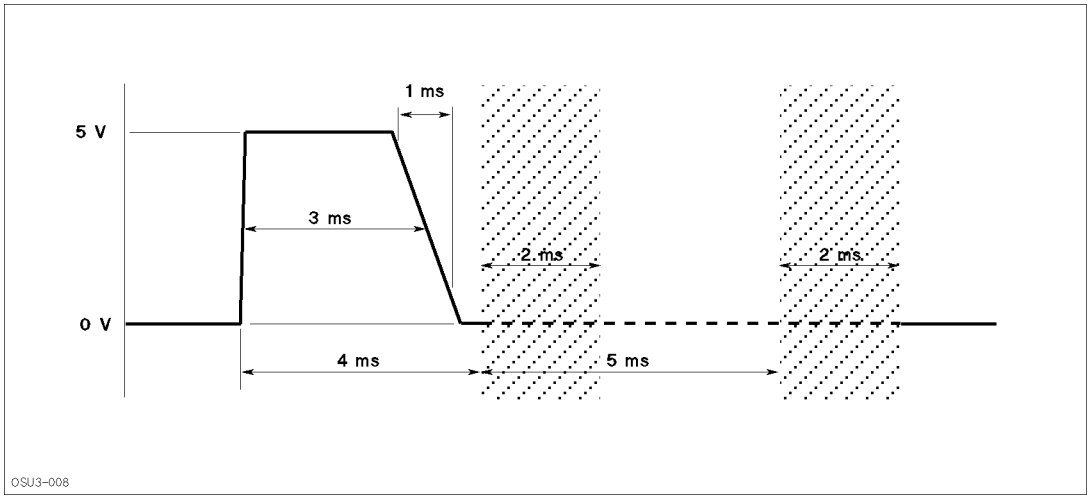
```
260  !
270  Force_pg(10,.02)
280  !
290  Connect_hf
300  !
310  END
```

Program Description.

lines 40 and 50:	Defines the type of variables used in the program.
lines 70 and 80:	Assigns measurement pins to variables. These are pins that are connected to DUT (resistor).
line 90:	Assigns PGU unit address to variable. This is unit address of PGU that controls the switch.
lines 110 and 120:	Initializes the system.
lines 140 and 150:	Connects the HF1 port and GNDU port to the measurement pins of DUT.
lines 160 and 170:	Assigns the values of timing for switching the open/close switch.
line 180:	Specifies the PGU for controlling the switches, and sets the timing for changing the open/close switch.
line 190:	Assigns the value of pulse amplitude to the variable.
line 200:	Sets the pulse amplitude.
lines 210 to 240:	Assigns the values of pulse width, delay time, and transition times to the variables.
line 250:	Sets the pulse width, delay time, and transition times.
line 270:	Forces 10 pulses with pulse period of 20 ms.
line 290:	Disconnects between measurement ports and measurement pins.

Example Program of PG Output Port Disconnection

Executing this program forces the pulse wave shown in the following figure to the DUT 10 times:



The Set_sr_pg subprogram specifies the PGU used for switch control, and sets the timing to switch the open/close switch. Immediately after this subprogram is executed, the control PGU forces a voltage that sets the switch to the default status (closed status in this example). The Force_pg subprogram starts the pulse output, then open/close switch operates according to the specified timing.

You can also switch the open/close switch without specifying the switching timing. To do so, you use Connect_sr subprogram. For details, see "Connect_sr" of the Chapter 4.

4

Programming Reference

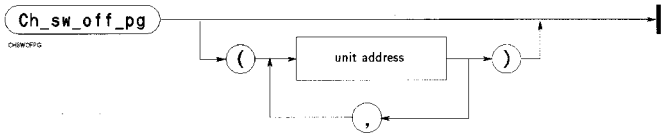
This chapter provides an alphabetical listing of subprograms contained in the system software of the HP 4062F that can be called in user-written programs. Subprograms that can be used for both the HP 4062F and HP 4062C/UX systems are described in the HP 4062C/UX *Programming Reference* manual, and are not described here.

Each entry defines the subprogram name, shows the syntax of the calling context, gives several examples, and includes relevant explanations, as shown below.

Ch_sw_off_pg

Supported on 4062F
File TIS.F
Revision F.06.30
Control PGU

This subprogram sets the output switch of the specified PGU(s) to off (disable output).



Item	Description/Defaults	Range Restrictions
unit address	integer expression default = available all units[1]	32581 to 32590

1 All available PGU that is not connected to HF port.

Example Statements

```
Ch_sw_off_pg
Ch_sw_off_pg(FNPg(11))
Ch_sw_off_pg(FNPg(11),FNPg(12),FNPg(31))
```

Description

This subprogram opens the output switch of the specified PGU(s), when the pulse is forced without the SWM. Note that you cannot specify the PGUs that is assigned to HF ports in the /usr/pcs/etc/FLCONFIG file.

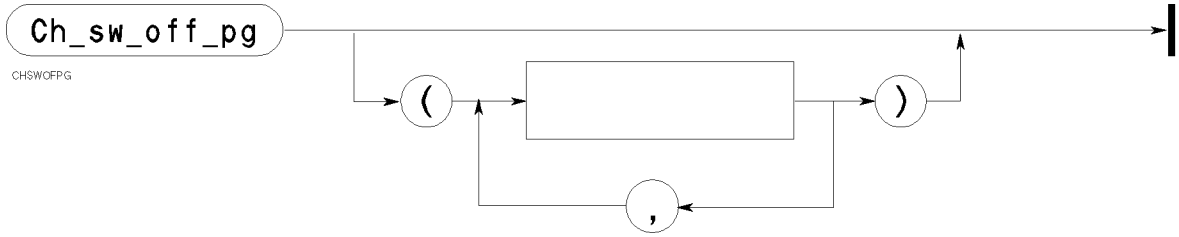
This subprogram without *unit address* parameter opens all the output switch of the PGU that is not connected to HF port.

(1) Subprogram name.
(2) Supported product name.
(3) Name of file that contains this subprogram.
(4) System software revision that this subprogram is supported for.
(5) Device controlled by this subprogram. Note that the device indicated here is the primary device. Other devices, however, may be affected also.
(6) Syntax diagram of the calling context for the subprogram.
(7) Explanation of parameters.

Ch_sw_off_pg

Supported on	4062F
File	TIS.F
Revision	F.06.30
Control	PGU

This subprogram sets the output switch of the specified PGUs to off (disables output).



Item	Description/Defaults	Range Restrictions
unit address	integer expression default = all available units[1]	32581 to 32590

1 All available PGUs that are not connected to HF port.

Example Statements

```
Ch_sw_off_pg
Ch_sw_off_pg(FNPg(11))
Ch_sw_off_pg(FNPg(11),FNPg(12),FNPg(31))
```

Description

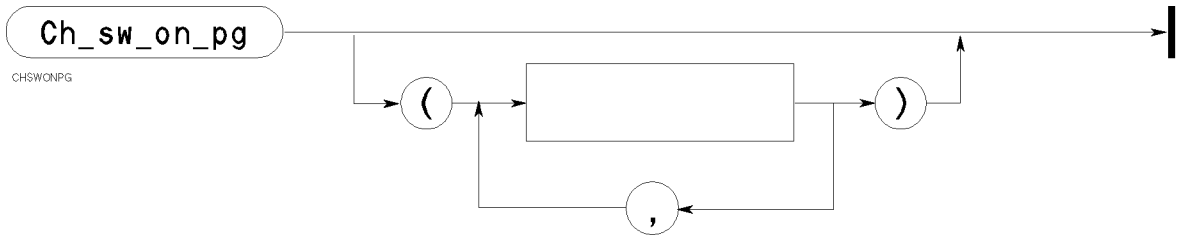
This subprogram opens the output switch of the specified PGUs when the pulse is forced without the SWM. Note that you cannot specify PGUs that are assigned to HF ports in the /usr/pcs/etc/FLCONFIG file.

If *unit address* parameter is omitted, opens the output switch of all PGUs that are *not* connected to an HF port (for example, switch CTRL PGU and directly connected PGU).

Ch_sw_on_pg

Supported on 4062F
File TIS.F
Revision F.06.30
Control PGU

This subprogram sets the output switch of the specified PGUs to on (enables output).



Item	Description/Defaults	Range Restrictions
unit address	integer expression default = all available units[1]	32581 to 32590

1 All available PGUs that are not connected to HF port.

Example Statements

```
Ch_sw_on_pg
Ch_sw_on_pg(FNPg(11))
Ch_sw_on_pg(FNPg(11),FNPg(12),FNPg(31))
```

Description

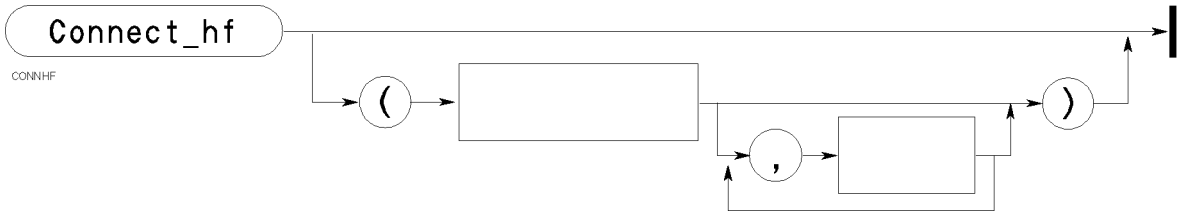
This subprogram closes the output switch of the specified PGUs when the pulse is forced without the SWM. Note that you cannot specify PGUs that are assigned to HF ports in the /usr/pcs/etc/FLCONFIG file.

If *unit address* parameter is omitted, closes the output switch of all PGUs that are *not* connected to an HF port (for example, switch CTRL PGU and directly connected PGU).

Connect_hf

Supported on 4062F
File TIS.F
Revision F.06.30
Control SWM

This subprogram connects a specified port to specified measurement pins.



Item	Description/Defaults	Range Restrictions
port address	integer expression default = 0	0, 32701 to 32711, 32741 to 32752
pin number	integer expression default = 0	0, 1 to 48

Example Statements

```
Connect_hf(FNHf(1),4)
Connect_hf(FNHf(3),9,10,11,12)
Connect_hf(FNHf(5),0)
Connect_hf(32747)
Connect_hf
```

Description

This subprogram connects a specified port (SMU, AUX, HF, guard, GNDU ports) to specified measurement pins, or disconnects the pin-port connections.

Available measurement pins for connecting to the non-HF ports are 1 to 48. HF ports can be connected to maximum 4 pins as shown in the following table:

HF Port	Port Address	Available Pin Number	HF Port	Port Address	Available Pin Number
HF1	32741	1, 2, 3, 4	HF7	32747	25, 26, 27, 28
HF2	32742	5, 6, 7, 8	HF8	32748	29, 30, 31, 32
HF3	32743	9, 10, 11, 12	HF9	32749	33, 34, 35,36
HF4	32744	13, 14, 15, 16	HF10	32750	37, 38, 39, 40
HF5	32745	17, 18, 19, 20	HF11	32751	41, 42, 43, 44
HF6	32746	21, 22, 23, 24	HF12	32752	45, 46, 47, 48

For a complete list of port addresses, see "FNPort".

Though the actual AUX1 port does not exist, an HF port can be connected to the measurement pins using the AUX1 port address. For example, "connect AUX1 port to pin 3" is the same as "connect HF1 port to pin 3".

`Connect_hf(FNAux(1),3)` is same as `Connect_hf(FNHf(1),3)`

0 (zero) specified for *port address* or *pin number* parameter means "disconnection". So the following statements disconnect as described:

`Connect_hf(FNHf(3),0)`	*disconnects all the pins connected to HF3.*
`Connect_hf(32741)`	*disconnects all the pins connected to HF1.*
`Connect_hf(0)`	*disconnects all the pin-port connections.*
`Connect_hf`	*disconnects all the pin-port connections.*

Note

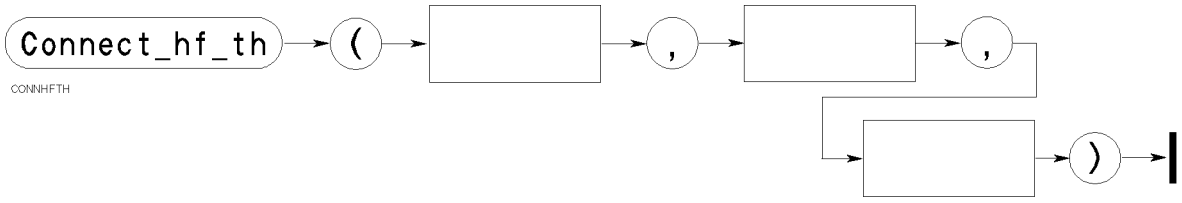
When using Connect subprogram to connect the port to measurement pins, do not use AUX1 port address as the *port address* parameter. If you execute the Connect subprogram with AUX1 port address as the *port address*, correct operation is not guaranteed.

Also, if you execute the Connect subprogram with an HF port specified for *port address*, an error occurs.

Connect_hf_th

Supported on	4062F
File	TIS.F
Revision	F.06.30
Control	SWM

This subprogram connects a specified port to a specified series of measurement pins.



Item	Description/Defaults	Range Restrictions
port address	integer expression	0, 32701 to 32711, 32741 to 32752
first pin number	integer expression	1 to 48
last pin number	integer expression	(*first pin number* + 1) to 48

Example Statements

```
Connect_hf_th(FNHf(1),1,4)
Connect_hf_th(FNPort(2),3,6)
Connect_hf_th(FNGnd,33,48)
```

Description

This subprogram connects a specified port to a specified series of measurement pins, or disconnects the pin-port connections. For HF port restrictions, see Connect_hf subprogram. Also, see Connect subprogram.

Note

To measure safely and exactly, Hewlett-Packard recommends to connect one port to one measurement pin. If one port is connected to multiple measurement pins, the increased stray capacitance may result in increased measurement error for capacitance measurement, and non-exact pulse signal is forced.

Note

When using Connect_th subprogram to connect the port to measurement pins, do not use AUX1 port address as the *port address* parameter. If you execute the Connect_th subprogram with AUX1 port address as the *port address*, correct operation is not guaranteed.

Also, if you execute the Connect_th subprogram with HF port specified for *port address*, an error occurs.

Connect_sr

Supported on 4062F
File TIS.F
Revision F.06.30
Control PGU, SWM

This subprogram switches the open/close switch or multiplexer switch.

Item	Description/Defaults	Range Restrictions
unit address	integer expression	32581 to 32590
connection status	integer expression	For open/close switch (SW2 and SW3): When mode selector[1] is set to "NC", 0: Opens switch 1: Closes switch When mode selector[1] is set to "NO", 0: Closes switch 1: Opens switch For multiplexer switch (SW1 and SW4): 1: Connects "In A" to "Out". 2: Connects "In B" to "Out".

1 Mode selector is on the cable adapter.

Example Statements

```
Connect_sr(FNPg(21),1)
```

Description

This subprogram controls the pulse output of PGU used for switch control, which sets or changes the connection status of the switches.

Before this subprogram is executed, the Set_sr_pg subprogram must set *usage mode* to "independent switching".

unit address is unit address PGU used for switch control. See "FNPg" for details.

connection status depends on type of switch (open/close or multiplexer switch), and for open/close switch also depends on the mode selector settings as follows:

- For open/close switch (SW2/SW3) when the mode selector is set to "NC":

  If *connection status* = `0` (zero), Connect_sr subprogram opens the open/close switch. If `1`, closes the switch.

- For open/close switch (SW2/SW3) when mode selector is set to "NO":

  If *connection status* = `0` (zero), Connect_sr subprogram closes the open/close switch. If `1`, opens the switch.

- For the multiplexer switch (SW1/SW4):

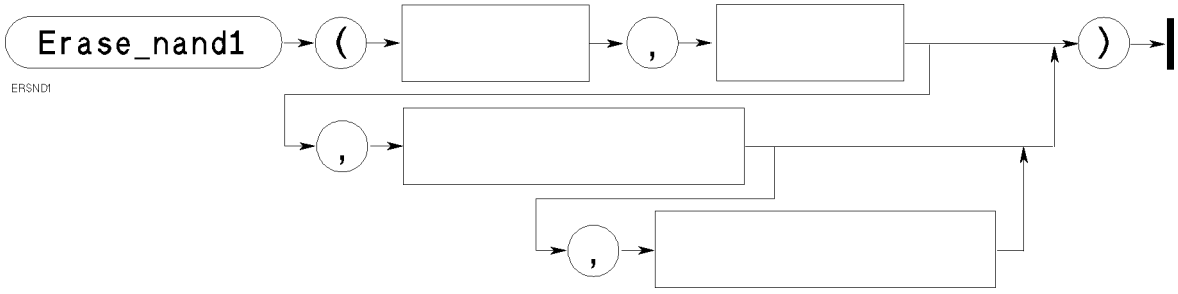
  If *connection status* = "1", Connect_sr subprogram connects to A side of multiplexer.

  If *connection status* = "2", Connect_sr subprogram connects to B side of multiplexer.

Erase_nand1

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram executes the erase operation of the flash memory cell (NAND type).



Item	Description/Defaults	Range Restrictions
pulse amplitude	numeric expression: [V]	−19 to −2.0E−1, 2.0E−1 to 19, resolution: 2.0E−2
pulse width	numeric expression: [s]	3.3E−9 to 9.99E−1, resolution: 3 digits
leading-edge transition time	numeric expression: [s] default = 2.0E−8	2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time	numeric expression: [s] default = leading-edge transition time	2.0E−9 to 2.0E−1, resolution: 3 digits

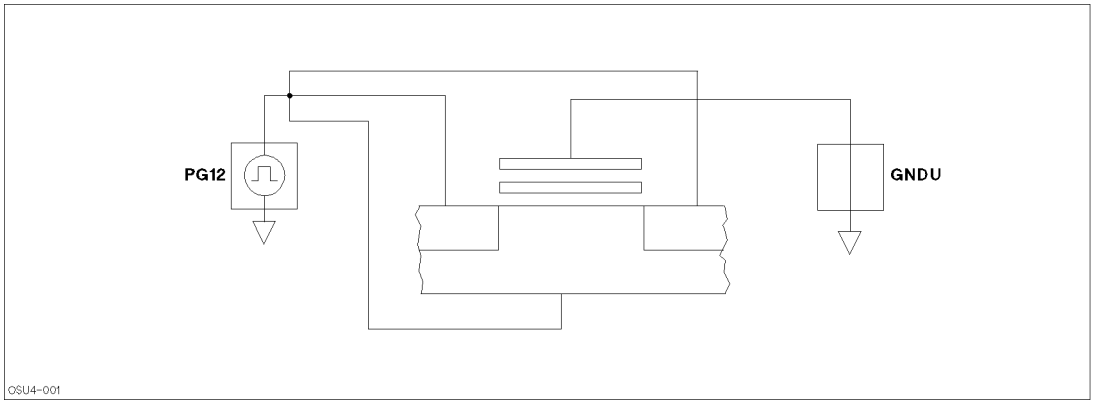
Example Statements

```
Erase_nand1(Ampl,Width,Leading,Trailing)
Erase_nand1(Amplitude,Width)
```

Description

This subprogram executes the erase operation of the flash memory cell (NAND type).

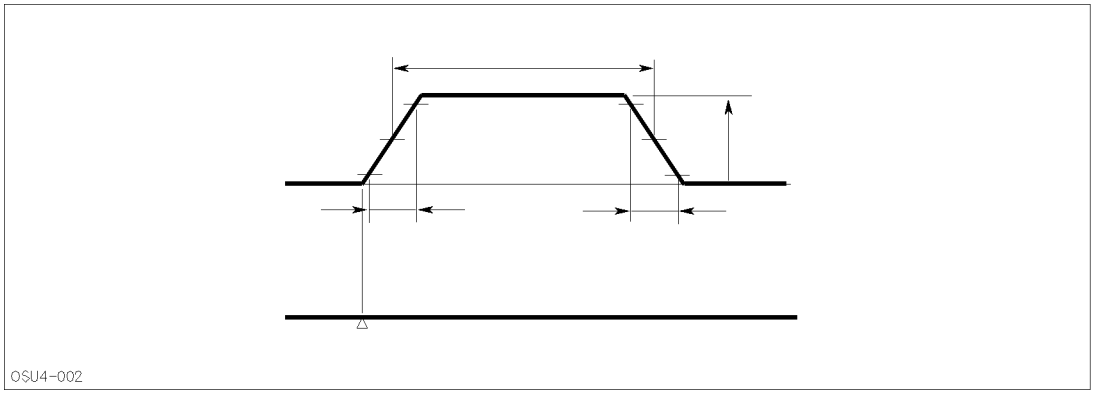
Erase_nand1 connects the measurement pins that were specified in Set_fr subprogram as in following figure.



Attach PG12 cable to HF port that corresponds to measurement pins of source, drain, and substrate terminals. For example, if source, drain, and substrate terminals are connected to measurement pins 1, 2, and 3 respectively, PG12 cable must be attached to HF1 port, which corresponds to measurement pins 1 to 4.

You can use this subprogram with the Write_nand1 and Repeat_nand1 subprograms in a measurement program. For the connections, refer to Repeat_nand1.

This subprogram forces the following pulses to the source, drain, and substrate, which erases the cell. When a certain voltage value is forced to the source, drain, and substrate, the charge carriers trapped in the floating gate move to the source, drain, and substrate through the silicon dioxide and channel, so the charge carrier in the floating gate decreases (that is, the cell is erased).



See Also

Set_fr
Set_multifr
Write_nand1
Repeat_nand1

Programming Example

This example program erases the NAND memory cell by using the following parameters:

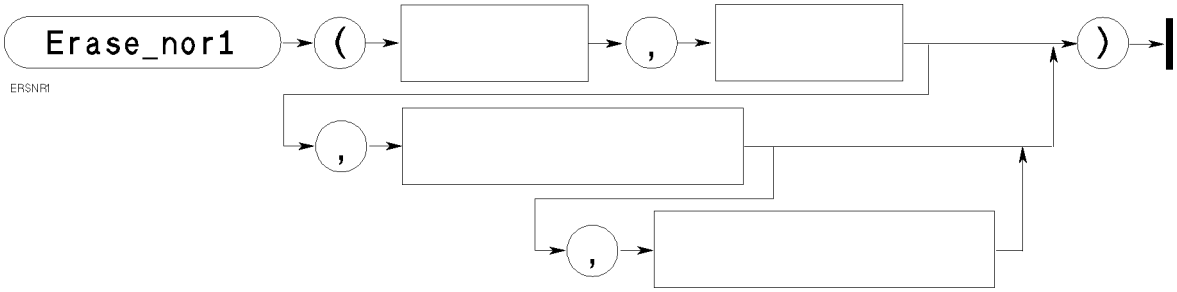
	Pulse Voltage	Pulse Width	Rise Time	Fall Time
Source, Drain, Substrate	15 V	10 μs	100 ns	20 ns
Gate	0 V	—	—	—

```
100   REAL Ampl,Width,Edge1,Edge2
110   INTEGER Gate,Source,Drain,Subs
120   !
130   Gate=1
140   Source=6
150   Drain=7
160   Subs=8
170   Ampl=15.
180   Width=1.E-5
190   Edge1=1.E-7
200   Edge2=2.E-8
210   !
220   Init_system
230   Init_pg
240   Set_fr(Source,Gate,Drain,Subs)
250   Erase_nand1(Ampl,Width,Edge1,Edge2)
260   !
270   END
```

Erase_nor1

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram executes the erase operation of the flash memory cell (normal NOR type).



Item	Description/Defaults	Range Restrictions
pulse amplitude	numeric expression: [V]	−19 to −2.0E−1, 2.0E−1 to 19, resolution: 2.0E−2
pulse width	numeric expression: [s]	3.3E−9 to 9.99E−1, resolution: 3 digits
leading-edge transition time	numeric expression: [s] default = 2.0E−8	2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time	numeric expression: [s] default = leading-edge transition time	2.0E−9 to 2.0E−1, resolution: 3 digits

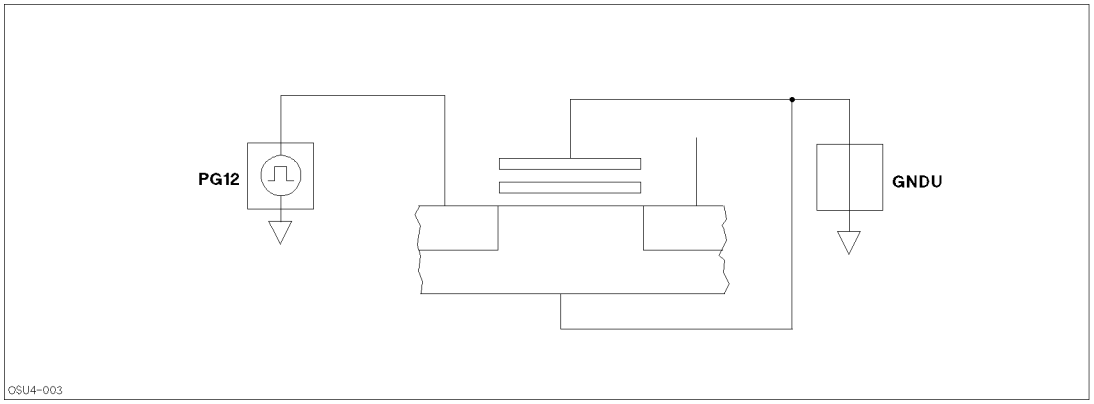
Example Statements

```
Erase_nor1(Ampl,Width,Leading,Trailing)
Erase_nor1(Amplitude,Width)
```

Description

This subprogram executes the erase operation of the flash memory cell (NOR type).

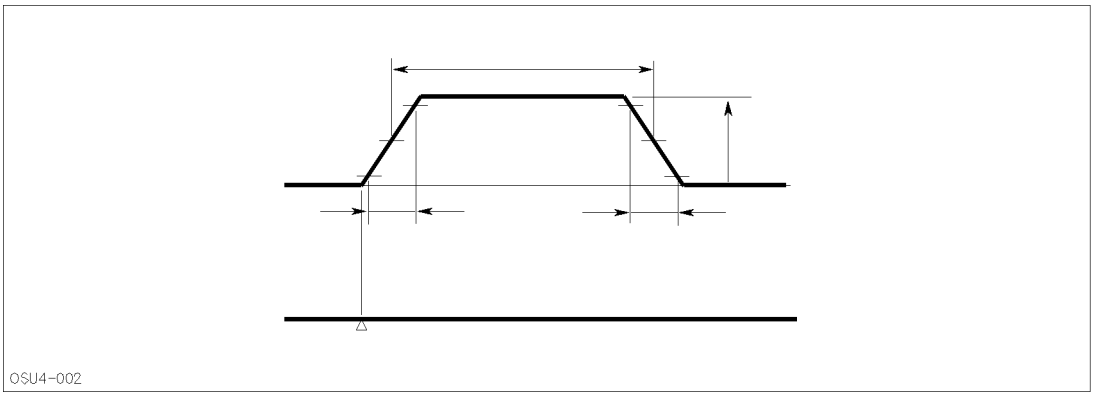
Erase_nor1 connects the measurement pins that were specified in Set_fr subprogram as in following figure. Erase_nor1 does not connect drain to any measurement pin, so drain is open.



Attach PG12 cable to the HF port that corresponds to the source measurement pin.

You can use this subprogram with the Write_nor1 and Repeat_nor1 subprograms in a measurement program. For the connections, refer to the Repeat_nor1 subprogram.

This subprogram forces the following pulses to the source, which erases the cell. When a certain voltage value is forced to the source, the charge carriers trapped in the floating gate move to the source through the gate oxide layer and channel, so the charge carrier in the floating gate decreases (that is, the cell is erased).



See Also

Set_fr
Set_multifr
Write_nor1
Repeat_nor1

Programming Example

This example program erases the NOR memory cell by using the following parameters:

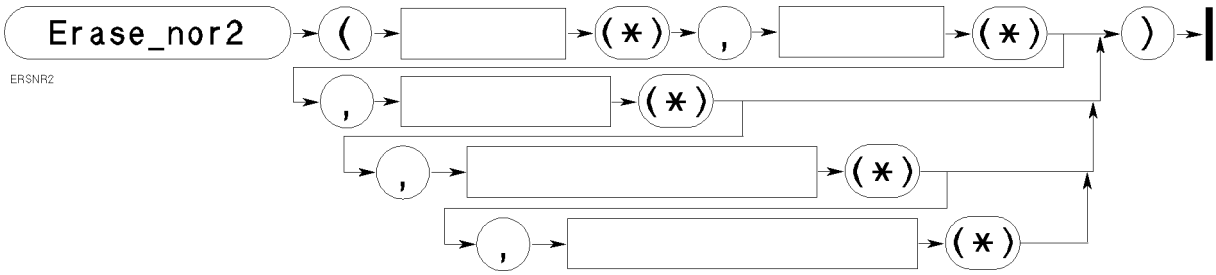
	Pulse Voltage	Pulse Width	Rise Time	Fall Time
Source	15 V	10 μs	100 ns	20 ns
Drain	open	—	—	—
Gate, Substrate	0 V	—	—	—

```
100   REAL Ampl,Width,Edge1,Edge2
110   INTEGER Gate,Source,Drain,Subs
120   !
130   Gate=1
140   Source=5
150   Drain=9
160   Subs=13
170   Ampl=15.
180   Width=1.E-5
190   Edge1=1.E-7
200   Edge2=2.E-8
210   !
220   Init_system
230   Init_pg
240   Set_fr(Source,Gate,Drain,Subs)
250   Erase_nor1(Ampl,Width,Edge1,Edge2)
260   !
270   END
```

Erase_nor2

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram executes the erase operation of a flash memory cell (low-power NOR type).



Item	Description/Defaults	Range Restrictions
pulse amplitude name	numeric array name: [V]	any valid name −19 to −2.0E−1, 2.0E−1 to 19, resolution: 2.0E−2
pulse width name	numeric array name: [s]	any valid name 3.3E−9 to 9.99E−1, resolution: 3 digits
delay time name	numeric array name: [s] default= 0	any valid name 0 to 9.99E−1, resolution: 3 digits
leading-edge transition time name	numeric array name: [s] default= 2.0E−8	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time name	numeric array name: [s] default= leading-edge transition time	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits

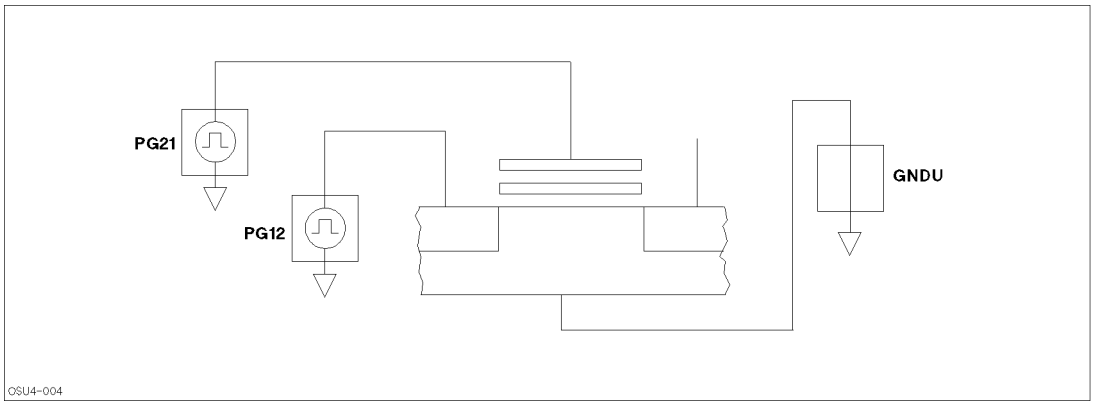
Example Statements

```
Erase_nor2(Ampl(*),Width(*),Delay(*),Leading(*),Trailing(*))
Erase_nor2(Amplitude(*),Width(*),Delay(*))
```

Description

This subprogram executes the erase operation of a flash memory cell (low-power NOR type).

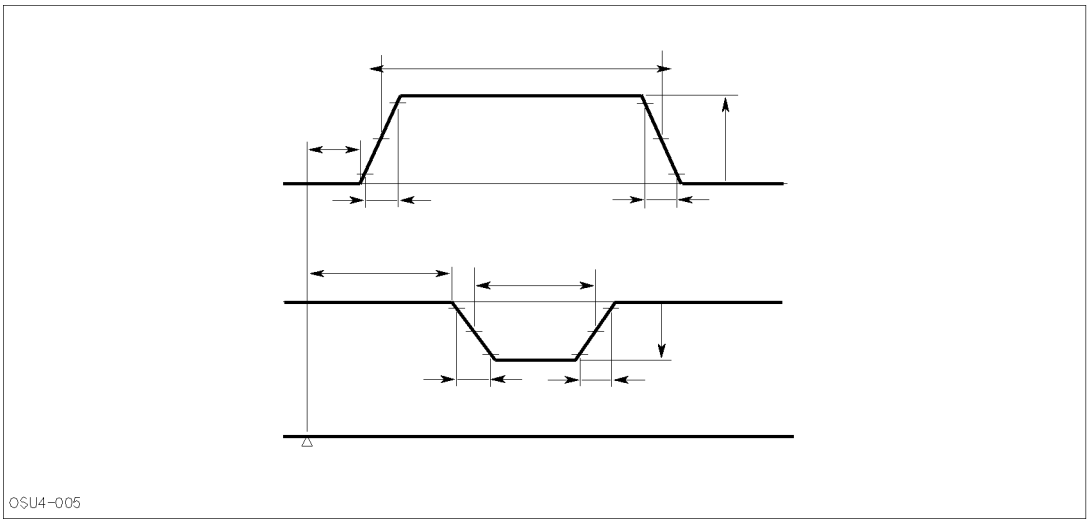
Erase_nor2 connects measurement pins specified in Set_fr subprogram as in following figure. Erase_nor2 does not connect drain to any measurement pin, so drain is open.



Attach PG12 cable to HF port that corresponds to source measurement pin. Attach PG21 cable to HF port that corresponds to gate measurement pin. Make sure that measurement pins connected to gate and source are connected to different HF ports. For example, connect gate to one of pins 1 to 4 (which is HF1), and connect source to one of measurement pins 9 to 12 (which is HF3).

You can use this subprogram with Write_nor2 and Repeat_nor2 subprograms in a measurement program. For the connections, refer to Repeat_nor2 subprogram.

Erase_nor2 forces following pulses to gate and source, which erases the cell. When a certain voltage value is forced to source, and a certain voltage value with opposite polarity is forced to gate, charge carriers trapped in floating gate move to the source through the silicon dioxide and channel, so charge carrier in floating gate decreases (that is, cell is erased).



Element 1 of each array is for source, element 2 is for gate, and other elements are ignored. So, size of each array must be 2 at least.

See Also

Set_fr
Set_multifr
Write_nor2
Repeat_nor2

Programming Example

This example program erases the low-power NOR memory cell by using the following parameters:

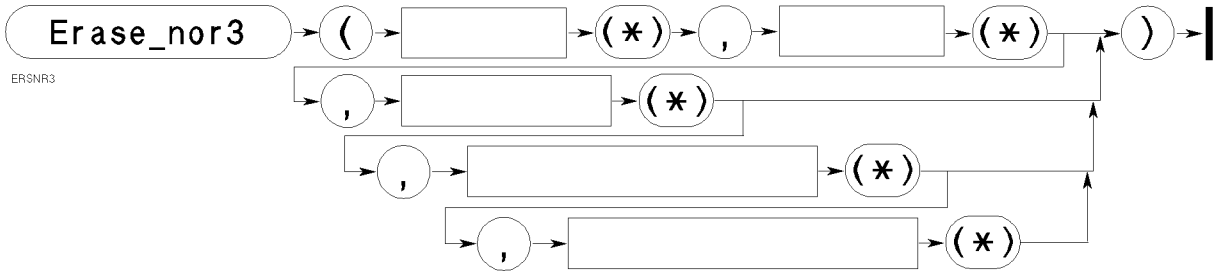
	Pulse Voltage	Delay Time	Pulse Width	Rise Time	Fall Time
Source	15 V	1 μs	10 μs	100 ns	100 ns
Gate	−5 V	1 μs	10 μs	100 ns	100 ns
Drain	open	—	—	—	—
Substrate	0 V	—	—	—	—

```
100   REAL Ampl(1:2),Width(1:2),Delay(1:2),Edge1(1:2),Edge2(1:2)
110   INTEGER Gate,Source,Drain,Subs
120   !
130   Gate=1
140   Source=5
150   Drain=9
160   Subs=13
170   Ampl(1)=15.
180   Ampl(2)=-5.
190   Width(1)=1.E-5
200   Width(2)=Width(1)
210   Delay(1)=1.E-6
220   Delay(2)=Delay(1)
230   Edge1(1)=1.E-7
240   Edge1(2)=Edge1(1)
250   Edge2(1)=Edge1(1)
260   Edge2(2)=Edge1(1)
270   !
280   Init_system
290   Init_pg
300   Set_fr(Source,Gate,Drain,Subs)
310   Erase_nor2(Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
320   !
330   END
```

Erase_nor3

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram executes the erase operation of a flash memory cell (low-power NOR type) by using a multiplexer switch for gate input.



Item	Description/Defaults	Range Restrictions
pulse amplitude name	numeric array name: [V]	any valid name −19 to −2.0E−1, 2.0E−1 to 19, resolution: 2.0E−2
pulse width name	numeric array name: [s]	any valid name 3.3E−9 to 9.99E−1, resolution: 3 digits
delay time name	numeric array name: [s] default= 0	any valid name 0 to 9.99E−1, resolution: 3 digits
leading-edge transition time name	numeric array name: [s] default= 2.0E−8	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time name	numeric array name: [s] default= leading-edge transition time	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits

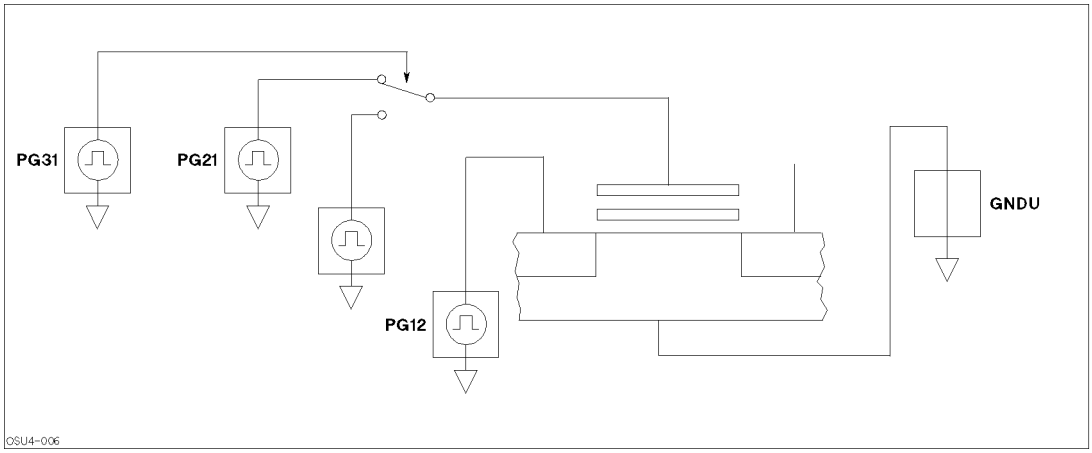
Example Statements

```
Erase_nor3(Ampl(*),Width(*),Delay(*),Leading(*),Trailing(*))
Erase_nor3(Amplitude(*),Width(*),Delay(*))
```

Description

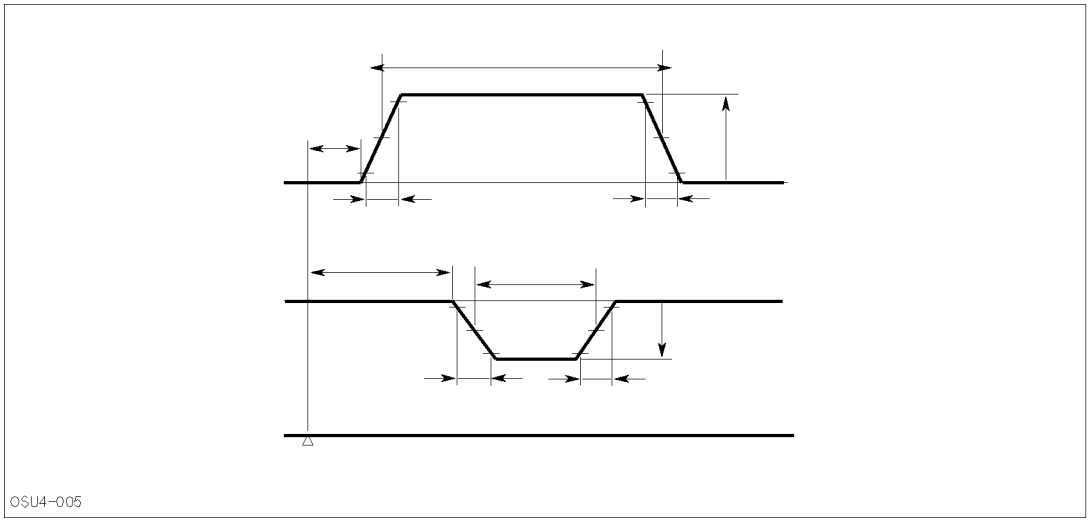
This subprogram executes the erase operation of a flash memory cell (low-power NOR type) by using the multiplexer switch for gate input.

Erase_nor3 connects measurement pins specified in Set_fr subprogram as in following figure, but does not connect drain to any measurement pin, so drain is open. Make sure that measurement pins connected to gate and source are connected to different HF ports. For example, connect gate to one of pins 1 to 4 (which is HF1), and connect drain to one of pins 9 to 12 (which is HF3).



You can use this subprogram with Write_nor3 and Repeat_nor3 subprograms in a measurement program. For the connections, refer to Repeat_nor3 subprogram.

Erase_nor3 forces following pulses to gate and source, which erases the cell. When a certain voltage value is forced to source, and a certain voltage value with opposite polarity is forced to gate, charge carriers trapped in floating gate move to source through the silicon dioxide and channel, and charge carrier in floating gate decreases (that is, cell is erased).



Element 1 of each array is for source, element 2 is for gate, and other elements are ignored. So, size of each array must be 2 at least.

See Also

Set_fr
Set_multifr
Set_relay_mode
Write_nor3
Repeat_nor3

Programming Example

This example program erases the low-power NOR memory cell by using the following parameters:

	Pulse Voltage	Delay Time	Pulse Width	Rise Time	Fall Time
Source	15 V	1 μs	10 μs	100 ns	100 ns
Gate	−5 V	1 μs	10 μs	100 ns	100 ns
Drain	open	—	—	—	—
Substrate	0 V	—	—	—	—

```
100   REAL Ampl(1:2),Width(1:2),Delay(1:2),Edge1(1:2),Edge2(1:2)
110   INTEGER Gate,Source,Drain,Subs
120   !
130   Gate=1
140   Source=5
150   Drain=9
160   Subs=13
170   Ampl(1)=15.
180   Ampl(2)=-5.
190   Width(1)=1.E-5
200   Width(2)=Width(1)
210   Delay(1)=1.E-6
220   Delay(2)=Delay(1)
230   Edge1(1)=1.E-7
240   Edge1(2)=Edge1(1)
250   Edge2(1)=Edge1(1)
260   Edge2(2)=Edge1(1)
270   !
280   Init_system
290   Init_pg
300   Set_fr(Source,Gate,Drain,Subs)
310   Set_relay_mode(1)
320   Erase_nor3(Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
330   !
340   END
```

Error_info_pg

Supported on 4062F
File TIS.F
Revision F.06.30
Control PGU

This subprogram returns the error messages for errors that occurred in the HP 8110A.

Item	Description/Defaults	Range Restrictions
error message name	string array name	any valid string name (size of string= 100, size of array= max. 5)

Example Statements

```
Error_info_pg(Message$(*))
```

Description

This subprogram gets error numbers and error messages from the HP 8110As used in the HP 4062F system, and returns them to the specified string array variables. The format of this string variable is as follows:

PG number:*error number*,*error message*

PG number is number specified in the /usr/pcs/etc/FLCONFIG file. For details about the error numbers and error messages, refer to the *HP 8110A User's Guide*.

Note that the array size must be greater than or equal to the number of HP 8110As used in the HP 4062F system.

FNHf

Supported on	4062F
File	TIS.F
Revision	F.06.30
Control	——

FNHf function returns the port address of the specified HF port of the SWM.

Item	Description/Defaults	Range Restrictions
HF port number	integer expression	1 to 12

Example Statements

```
Gate=FNHf(1)
```

Description

This function returns the port address of the HF port, which is specified by the *HF port number*. The *HF port number* is the number of the HF port name: HF1 to HF12.

HF port names, HF port numbers, and port addresses are as follows:

HF Port Name	HF Port Number	Port Address
HF1	1	32741
HF2	2	32742
HF3	3	32743
HF4	4	32744
HF5	5	32745
HF6	6	32746
HF7	7	32747
HF8	8	32748
HF9	9	32749
HF10	10	32750
HF11	11	32751
HF12	12	32752

FNPg

Supported on 4062F
File TIS.F
Revision F.06.30
Control ——

FNPg function returns the unit address of the specified PGU.

Item	Description/Defaults	Range Restrictions
unit identifier	integer expression	11, 12, 21, 22, 31, 32, 41, 42, 51, 52

Example Statements

```
Drain=FNPg(21)
```

Description

This function returns the unit address of the PGU specified by the *unit identifier*. The *unit identifier* is the two digits of the unit name PG11 to PG52.

Unit names, unit identifiers, and unit addresses are as follows:

Unit Name	*unit identifier*	Unit Address
PG11	11	32581
PG12	12	32582
PG21	21	32583
PG22	22	32584
PG31	31	32585
PG32	32	32586
PG41	41	32587
PG42	42	32588
PG51	51	32589
PG52	52	32590

FNPort

Supported on	4062F
File	TIS.F
Revision	F.06.30
Control	——

FNPort function returns the port address of the specified SWM port

Item	Description/Defaults	Range Restrictions
port number	integer expression	−1 to 11, −8, −9, 41 to 52

Example Statements

```
Source=FNPort(51)
Port=FNPort(N)
```

Description

This function returns the port address of the specified SWM port. Port names, port numbers, and port addresses are as follows:

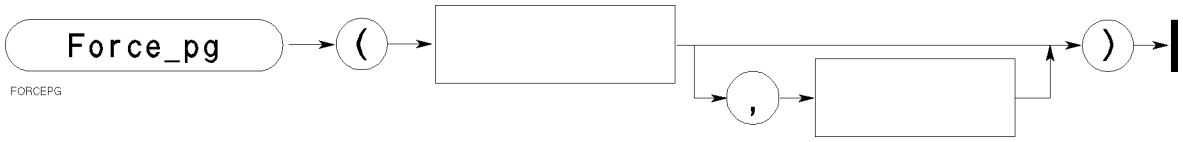
Port Name	Description	Port Number	Port Address
———	Imaginary port to which nothing is connected	0	0
SMU1	SMU1 (for low current measurements)	1	32701
SMU2	SMU2 (for high current measurements)	2	32702
SMU3	SMU3	3	32703
SMU4	SMU4	4	32704
AUX1	HF port	5	32705
AUX2	VS or VM[1]	6	32706
GNDU	Circuit common for dc measurements	7	32707
AUX3 (CMH)	VS, VM, or CMS High port[1]	8	32708
AUX4 (CML)	VS, VM, or CMS Low port[1]	9	32709
GSMU	Guard of SMU1	10, −1	32710
GCMU	Guard of CMS	11, −8, −9	32711
HF1	HF1 (connected to pins 1 – 4)	41	32741
HF2	HF2 (connected to pins 5 – 8)	42	32742
HF3	HF3 (connected to pins 9 – 12)	43	32743
HF4	HF4 (connected to pins 13 – 16)	44	32744
HF5	HF5 (connected to pins 17 – 20)	45	32745
HF6	HF6 (connected to pins 21 – 24)	46	32746
HF7	HF7 (connected to pins 25 – 28)	47	32747
HF8	HF8 (connected to pins 29 – 32)	48	32748
HF9	HF9 (connected to pins 33 – 36)	49	32749
HF10	HF10 (connected to pins 37 – 40)	50	32750
HF11	HF11 (connected to pins 41 – 44)	51	32751
HF12	HF12 (connected to pins 45 – 48)	52	32752

1 The units or terminals physically connected to the AUX2 to AUX4 ports are displayed during the START.F program under AUXILIARY PORT CONFIGURATION.

Force_pg

Supported on	4062F
File	TIS.F
Revision	F.06.30
Control	PGU

This subprogram actually forces the pulse from all PGUs that are setup.



Item	Description/Defaults	Range Restrictions
number of periods	numeric expression	1 to 10000000
pulse period	numeric expression: [s] default= see Description	6.65E−9 to 999[1]

1 For pattern mode, 1.2E−4 to 999.

Example Statements

```
Force_pg(100)
Force_pg(1000000,1.E-3)
```

Description

This subprogram actually forces the pulses from all PGUs that have been set up by the Set_level_pg and Set_time_pg subprograms.

If the specified *number of periods* is a non-integer, the fractional part is truncated. For example, if you specify 13.7, the pulse will be forced 13 times.

The *pulse period* must be T or more. T is determined as follows (30 ns is the skew compensation time, and 2 ms is the switching time):

- Minimum pulse period $T_{PGxx\text{-}2L}$ for 2-level pulse output PGU

  $T_{PGxx\text{-}2L}$ = (pulse width + delay time + 30 ns)×1.1

- Minimum pulse period $T_{PGxx\text{-}3L}$ for 3-level pulse output PGU

  $T_{PGxx\text{-}3L}$ = (pulse width 2 + delay time 2 + 30 ns)×1.1

- Minimum pulse period $T_{PG\text{-}CTRL}$ for switch control PGU

  $T_{PG\text{-}CTRL}$ = (pulse width + delay time + 30 ns + 2 ms)

Pulse period must be T or more:

$$T = MAX(T_{PGxx\text{-}2L}, T_{PGxx\text{-}3L}, \cdots\cdots, T_{PG\text{-}CTRL})$$

Also the pulse period must be more than 120 μs (1.2E−4) for pattern mode operation.

If the *pulse period* is not specified, T is used. If T is less than 120 μs for pattern mode operation, 120 μs is used.

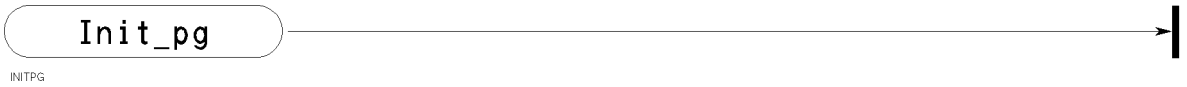
Note

After the Force_pg subprogram starts the pulse output, the pulses are continuously forced the *number of periods*. You cannot abort the pulse output during the pulse forcing.

Init_pg

Supported on	4062F
File	TIS.F
Revision	F.06.30
Control	—

This subprogram initializes the pulse generators, HF ports, and system variables corresponding to the pulse generator.



Description

Executing this subprogram initializes all the pulse generators and settings that correspond to the pulse generators and HF ports. This subprogram must be executed before you execute the measurements that use pulse generators.

Initial settings that correspond to the pulse generators are as follows:

Output value:	0 V
Time accuracy mode:	Normal
Output mode:	2-level pulse output
Pulse base:	0 V
Pulse amplitude 1:	0.2 V
Pulse width 1:	500 ns
Delay time 1:	0
Leading-edge transition time 1:	20 ns
Trailing-edge transition time 1:	20 ns
Pulse amplitude 2:	—
Pulse width 2:	—
Delay time 2:	—
Leading-edge transition time 2:	—
Trailing-edge transition time 2:	—
Load impedance:	999 kΩ

For SWM initial settings that correspond to the HF ports, *no* HF ports of the HP 4085F are connected to any measurement pins.

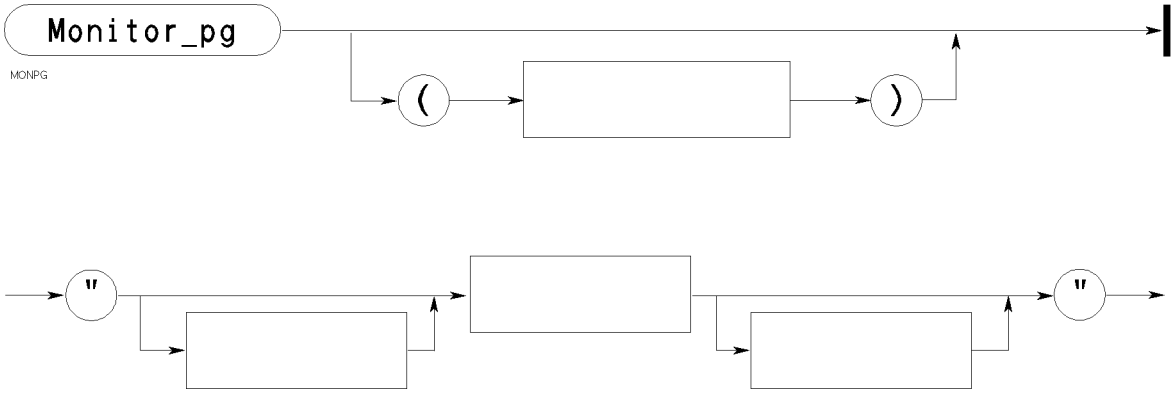
Programming Example

```
10   Init_system
20   Init_pg
30   !
40   Connect_hf(FNHf(1),2)
50   Connect_hf(FNGnd,8)
60   !
70   Connect_hf
80   END
```

Monitor_pg

Supported on 4062F
File VFP.AC
Revision F.06.30
Control PGU, SWM

This subprogram calls the VFP.AC program in the test program and enables you to set the pulse output parameters. If you specify the name of a file that has pulse setting data, pulse is forced according to the data.



Item	Description/Defaults	Range Restrictions
file specifier	string expression default= see Description	(see above drawing)
directory path	literal	see MASS STORAGE IS in the *BASIC Language Reference*
filename	literal	Depends on the format of volume (see Glossary in the *BASIC Language Reference*.)
volume specifier	literal	see MASS STORAGE IS in the *BASIC Language Reference*

Example Statements

```
Monitor_pg
Monitor_pg("setup.1011")
```

Description

This subprogram is used to call the pulse monitor tool VFP.AC from your test program.

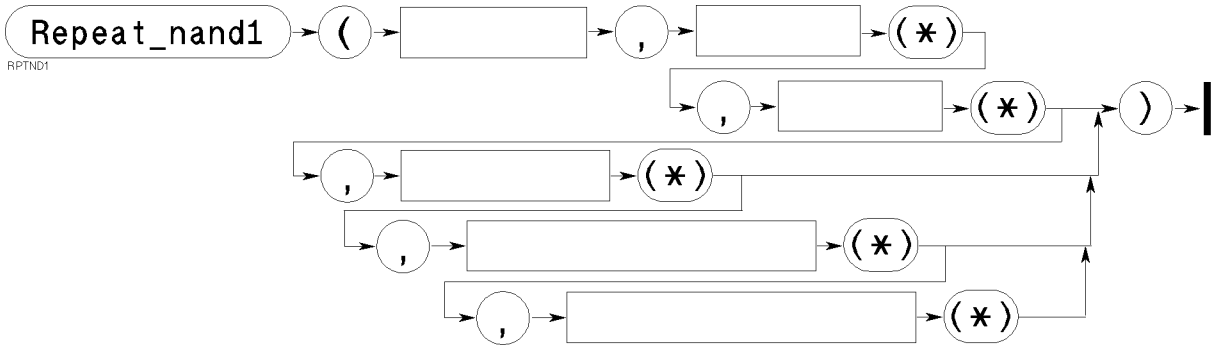
If you call this subprogram without specifying the *file specifier*, the pulse parameter setting page of the VFP.AC is displayed, which enables you to interactively create and force your desired pulse wave. The initial pulse parameters are the same as when this subprogram is called. See "Forcing Pulses Manually" in Chapter 2 about how to set the pulse parameters interactively.

If you specify *file specifier* for this subprogram, pulses are forced according to the pulse setting data contained in the file.

Repeat_nand1

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram continuously executes the write and erase operation of the flash memory cell (NAND type).



Item	Description/Defaults	Range Restrictions
number of periods	numeric expression default= 1	1 to 10000000[1]
pulse amplitude name	numeric array name: [V]	any valid name −19 to −2E−1, 2E−1 to 19, resolution: 2.0E−2
pulse width name	numeric array name: [s]	any valid name 3.3E−9 to 9.99E−1, resolution: 3 digits
delay time name	numeric array name: [s] default= 0	any valid name 0 to 9.99E−1, resolution: 3 digits
leading-edge transition time name	numeric array name: [s] default= 2.0E−8	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time name	numeric array name: [s] default= leading-edge transition time	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits

1 If non-integer is specified, fractional part is truncated.

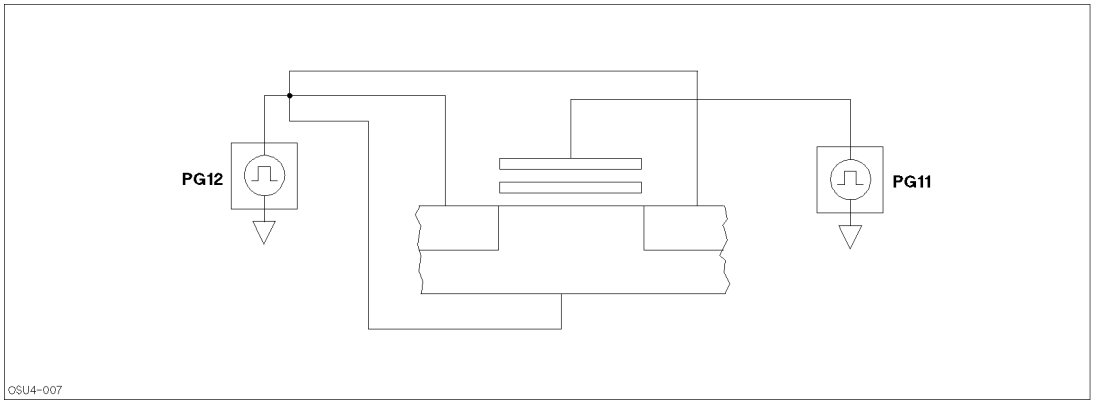
Example Statements

```
Repeat_nand1(Iterate,Ampl(*),Width(*),Delay(*),Leading(*),Trailing(*))
Repeat_nand1(N,Amplitude(*),Width(*))
```

Description

This subprogram continuously executes the write and erase operation of the flash memory cell (NAND type) the specified number of times (*number of periods*).

Repeat_nand1 connects the measurement pins specified in Set_fr subprogram as in the following figure.

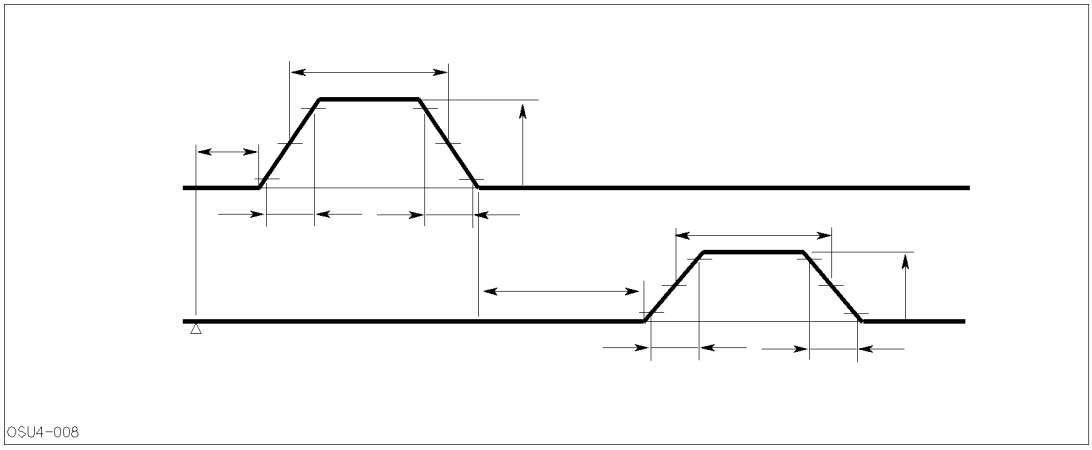


Attach PG12 cable to HF port that corresponds to measurement pins of source, drain, and substrate terminals. For example, if source, drain, and substrate terminals are connected to measurement pins 1, 2, and 3 respectively, PG12 cable must be attached to HF1 port, which corresponds to measurement pins 1 to 4.

Attach PG11 cable to HF port that corresponds to gate measurement pin.

Make sure that HF port that corresponds to source, drain, and substrate pins is not the same HF port that corresponds to gate pin. For example, connect source, drain, and substrate to pins 1, 2, 3 respectively (which is HF1), and connect source to one of measurement pins 9 to 12 (which is HF3).

This subprogram continuously forces the following pulses to the flash memory cell, which writes and erases the cell. First, forcing a pulse to the gate injects the charge carrier into the floating gate (write operation). Then forcing a pulse to source, drain, and substrate removes the charge carrier trapped in the floating gate (erase operation).



The time when the gate pulse forcing completes becomes the time origin for the pulse forcing to the source, drain, and substrate.

Element 1 of each array is for gate, element 2 is for source, drain, and substrate, and other elements are ignored. So, the size of each array must be 2 at least.

See Also

Set_fr
Set_multifr
Erase_nand1
Write_nand1

Programming Example

This example program writes and erases the NAND memory cell by using the following parameters:

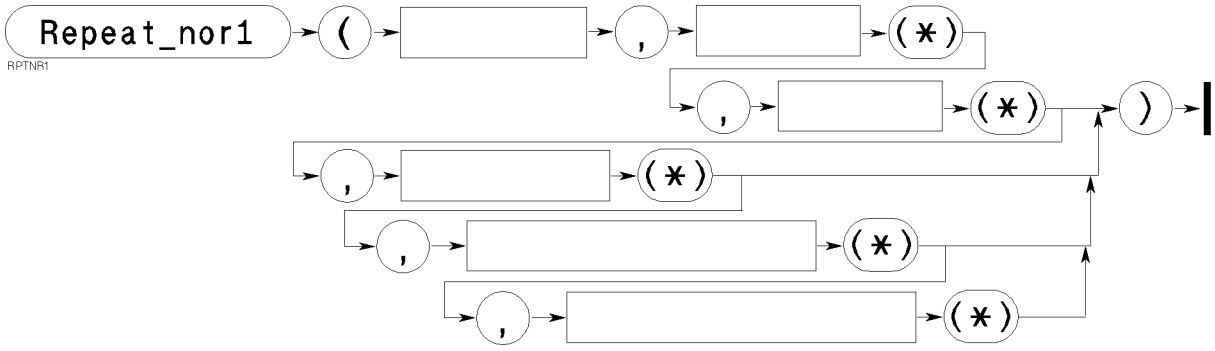
	Pulse Voltage	Delay Time	Pulse Width	Rise Time	Fall Time
Gate	15 V	0 s	10 μs	1 μs	1 μs
Source, Drain, Substrate	15 V	20 μs	10 μs	20 ns	20 ns

```
100   REAL Ampl(1:2),Width(1:2),Delay(1:2),Edge1(1:2),Edge2(1:2)
110   REAL Iterate
120   INTEGER Gate,Source,Drain,Subs
130   !
140   Iterate=100000
150   Gate=1
160   Source=5
170   Drain=9
180   Subs=13
190   Ampl(1)=15.
200   Ampl(2)=15.
210   Width(1)=1.E-5
220   Width(2)=1.E-5
230   Delay(1)=0.
240   Delay(2)=2.E-5
250   Edge1(1)=1.E-6
260   Edge1(2)=2.E-8
270   Edge2(1)=1.E-6
280   Edge2(2)=2.E-8
290   !
300   Init_system
310   Init_pg
320   Set_fr(Source,Gate,Drain,Subs)
330   Repeat_nand1(Iterate,Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
340   !
350   END
```

Repeat_nor1

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram continuously executes the write and erase operation of the flash memory cell (normal NOR type).



Item	Description/Defaults	Range Restrictions
number of periods	numeric expression default= 1	1 to 10000000[1]
pulse amplitude name	numeric array name: [V]	any valid name −19 to −2E−1, 2E−1 to 19, resolution: 2.0E−2
pulse width name	numeric array name: [s]	any valid name 3.3E−9 to 9.99E−1, resolution: 3 digits
delay time name	numeric array name: [s] default= 0	any valid name 0 to 9.99E−1, resolution: 3 digits
leading-edge transition time name	numeric array name: [s] default= 2.0E−8	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time name	numeric array name: [s] default= leading-edge transition time	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits

1 If non-integer is specified, fractional part is truncated.

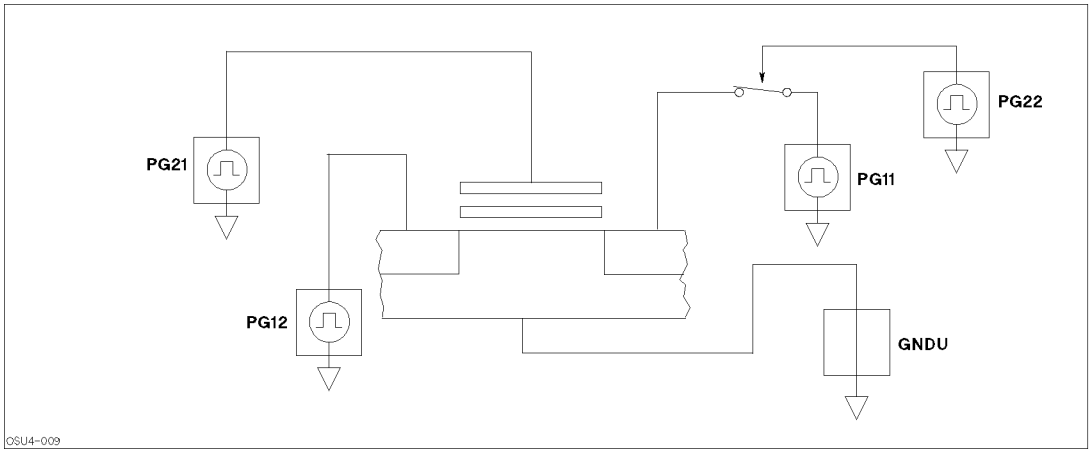
Example Statements

```
Repeat_nor1(Iterate,Ampl(*),Width(*),Delay(*),Leading(*),Trailing(*))
Repeat_nor1(N,Amplitude(*),Width(*))
```

Description

This subprogram continuously executes the write and erase operation of the flash memory cell (NOR type) the specified number of times (*number of periods*).

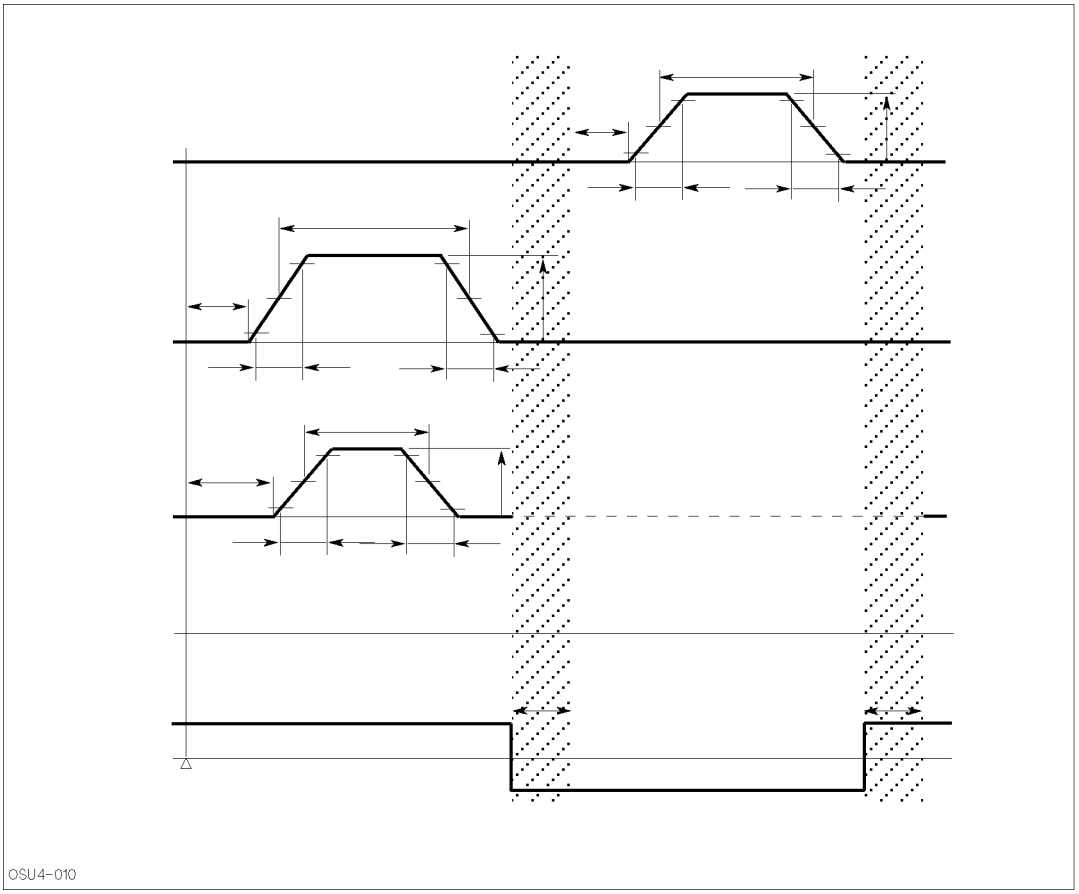
Repeat_nor1 connects the measurement pins specified in Set_fr subprogram as in following figure.



Attach PG11 cable to input of open/close switch. Attach output of open/close switch to HF port that corresponds to drain pin.
Attach PG12 cable to HF port that corresponds to source pin.
Attach PG21 cable to HF port that corresponds to gate pin.
Attach PG22 to switch CTRL of open/close switch.

Make sure the measurement pins connected to the gate, source, and drain are connected to different HF ports. For example, connect gate to one of pins 1 to 4 (which is HF1), connect source to one of pins 5 to 8 (which is HF2), and connect drain to one of pins 9 to 12 (which is HF3).

This subprogram continuously forces the following pulses to the flash memory cell, which writes and erases the cell. First, pulses are forced to the gate and drain, which injects the charge carrier into the floating gate (write operation). Then, a pulse is forced to source, which removes the charge carrier trapped in the floating gate (erase operation).



After the pulse outputs to the gate and drain are completed, the open/close switch starts to open, which takes 2 ms to stabilize. The point at which the switch becomes stable is defined as the time origin of the source output, then the pulse is forced to the source. After the source pulse output is completed, the open/close switch starts to close, which takes 2 ms.

Element 1 of each array is for gate, element 2 is for drain, element 3 is for source, and other elements are ignored. So, the size of each array must be 3 at least.

Set_relay_mode subprogram must execute before Repeat_nor1 to specify the use of open/close switch.

Repeat_nor1

You can use this subprogram with the Write_nor1 and Erase_nor1 subprograms in a measurement program. The use of the open/close switch must be specified by the Set_relay_mode subprogram. When the Write_nor1 or Erase_nor1 subprogram executes, the open/close switch is automatically controlled. For example, you can make a program as follows:

```
 .   .   .   .   .
Init_system
Init_pg
Set_fr(Source,Gate,Drain,Subs)
Set_tr(Source,Gate,Drain,Subs)
Set_relay_mode(1)
Repeat_nor1(Iterate,Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
Write_nor1(Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
Vth=FNVth3(Id,Vds,Vgs_limit,Vsub)
Erase_nor1(Ampl(3),Width(3),Edge1(3),Edge2(3))
Vth=FNVth3(Id,Vds,Vgs_limit,Vsub)
END
 .   .   .   .   .
```

See Also

Set_fr
Set_multifr
Set_relay_mode
Erase_nor1
Write_nor1

Programming Example

This example program writes and erases the NOR memory cell by using the following parameters:

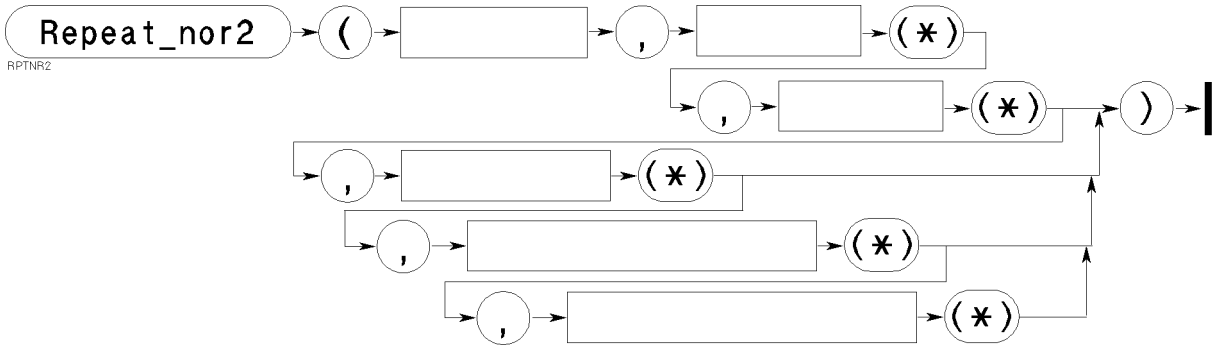
	Pulse Voltage	Delay Time	Pulse Width	Rise Time	Fall Time
Gate	15 V	0 s	10 μs	20 ns	20 ns
Drain	7 V	1 μs	5 μs	20 ns	20 ns
Source	12 V	0 s	1 ms	1 μs	1 μs
Substrate	0 V	—	—	—	—

```
100   REAL Ampl(1:3),Width(1:3),Delay(1:3),Edge1(1:3),Edge2(1:3)
110   REAL Iterate
120   INTEGER Gate,Source,Drain,Subs
130   !
140   Iterate=100000
150   Gate=1
160   Source=5
170   Drain=9
180   Subs=13
190   Ampl(1)=15.
200   Ampl(2)=7.
210   Ampl(3)=12.
220   Width(1)=1.E-5
230   Width(2)=5.E-6
240   Width(3)=1.E-3
250   Delay(1)=0.
260   Delay(2)=1.E-6
270   Delay(3)=0.
280   Edge1(1)=2.E-8
290   Edge1(2)=2.E-8
300   Edge1(3)=1.E-6
310   Edge2(1)=2.E-8
320   Edge2(2)=2.E-8
330   Edge2(3)=1.E-6
340   !
350   Init_system
360   Init_pg
370   Set_fr(Source,Gate,Drain,Subs)
380   Set_relay_mode(1)
390   Repeat_nor1(Iterate,Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
400   !
410   END
```

Repeat_nor2

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram continuously executes the write and erase operation of the the flash memory cell (low-power NOR type).



Item	Description/Defaults	Range Restrictions
number of periods	numeric expression default= 1	1 to 10000000[1]
pulse amplitude name	numeric array name: [V]	any valid name −19 to −2E−1, 2E−1 to 19, resolution: 2.0E−2
pulse width name	numeric array name: [s]	any valid name 3.3E−9 to 9.99E−1, resolution: 3 digits
delay time name	numeric array name: [s] default= 0	any valid name 0 to 9.99E−1, resolution: 3 digits
leading-edge transition time name	numeric array name: [s] default= 2.0E−8	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time name	numeric array name: [s] default= leading-edge transition time	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits

1 If non-integer is specified, fractional part is truncated.

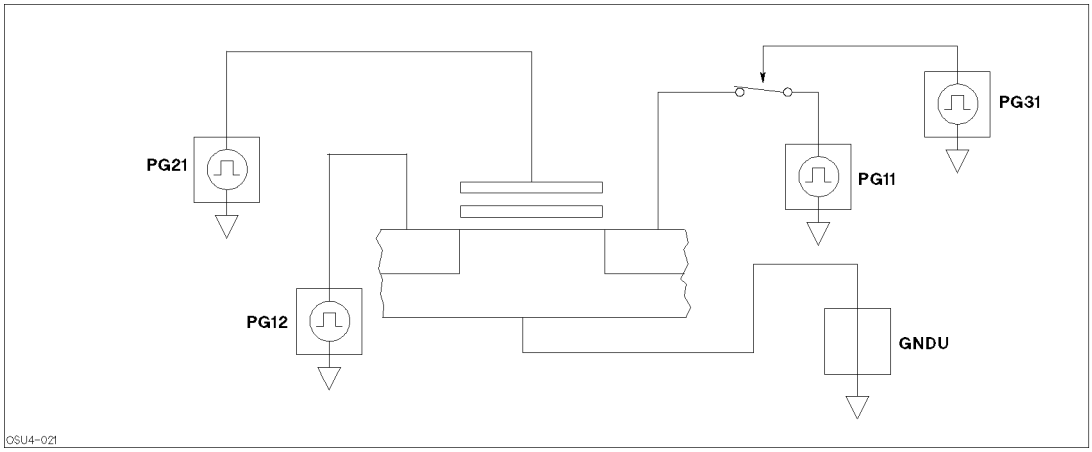
Example Statements

```
Repeat_nor2(Iterate,Ampl(*),Width(*),Delay(*),Leading(*),Trailing(*))
Repeat_nor2(N,Amplitude(*),Width(*))
```

Description

This subprogram continuously executes the write and erase operation of the flash memory cell (low-power NOR type) the specified number of times (*number of periods*).

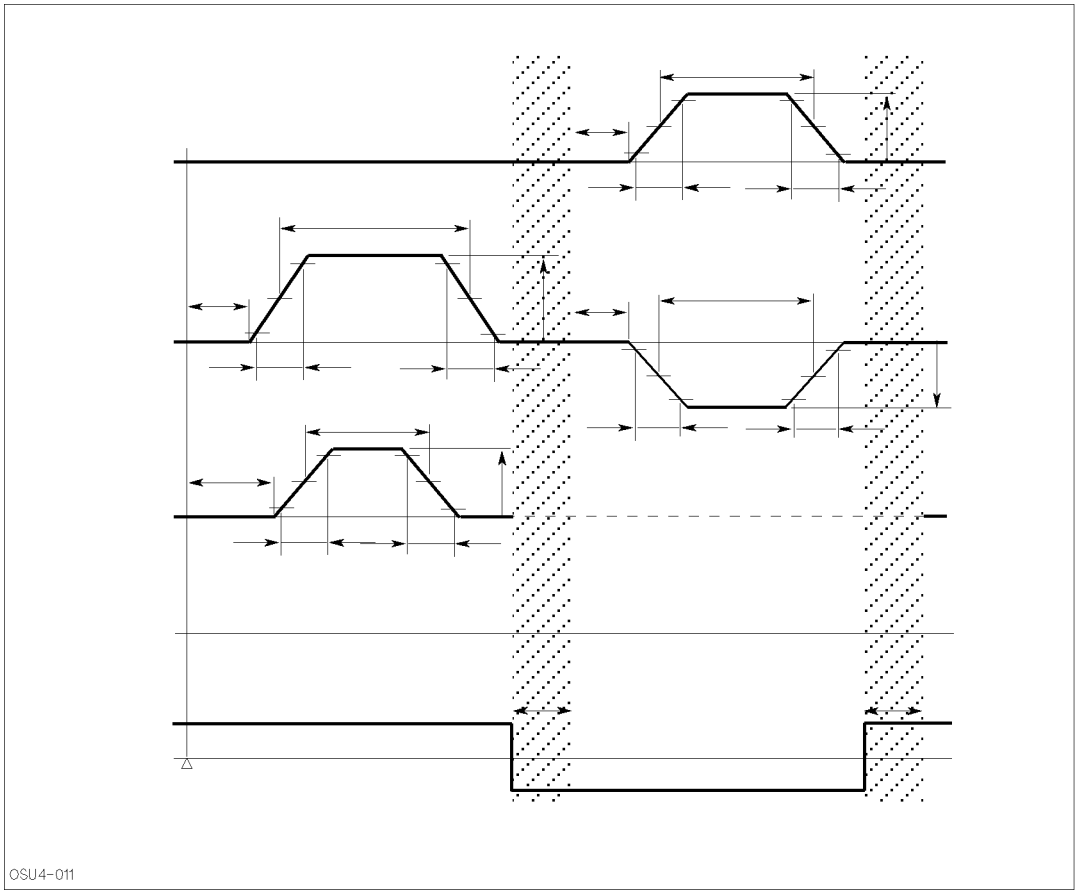
Repeat_nor2 connects the measurement pins specified in Set_fr subprogram as in following figure.



Attach PG11 cable to input of open/close switch. Attach output of open/close switch to HF port that corresponds to drain pin.
Attach PG12 cable to HF port that corresponds to source pin.
Attach PG21 cable to HF port that corresponds to gate pin.
Attach PG31 to switch CTRL of open/close switch.

Make sure the measurement pins connected to the gate, source, and drain are connected to different HF ports. For example, connect gate to one of pins 1 to 4 (which is HF1), connect source to one of pins 5 to 8 (which is HF2), and connect drain to one of pins 9 to 12 (which is HF3).

This subprogram continuously forces the following pulses to the flash memory cell, which writes and erases the cell. First, pulses are forced to the gate and drain, which injects the charge carrier into the floating gate (write operation). Then, pulses (with different level and polarity) are forced to the source and gate with the drain open, which removes the charge carriers from the floating gate (erase operation).



After the pulse outputs to the gate and drain are completed, the open/close switch starts to open, which takes 2 ms to stabilize. The point at which the switch becomes stable is defined as the time origin of the pulse output to source and gate. After pulse output to source and gate is completed, the open/close switch starts to close, which takes 2 ms. Set_relay_mode subprogram must execute before Repeat_nor2 to specify the use of open/close switch.

Note that the sum of absolute values of first and second pulse amplitudes for the gate must not exceed 20 V. If you need to exceed this voltage, use Repeat_nor3.

Element 1 of each array is for first gate, element 2 is for drain, element 3 is for source, element 4 is for second gate, and other elements are ignored. So, the size of each array must be 4 at least.

See Also

Set_fr
Set_multifr
Set_relay_mode
Erase_nor2
Write_nor2

Programming Example

This example program writes and erases the low-power NOR memory cell by using the following parameters:

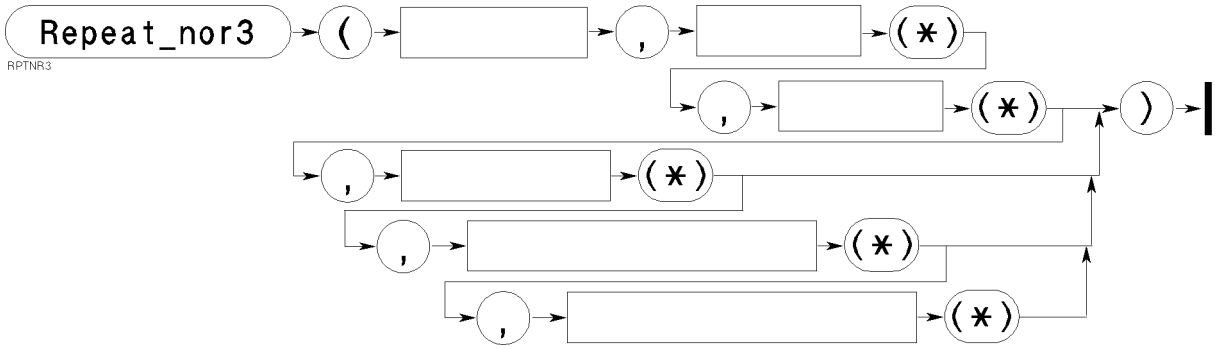
	Pulse Voltage	Delay Time	Pulse Width	Rise Time	Fall Time
Gate 1	12 V	0 s	10 μs	20 ns	20 ns
Drain	15 V	1 μs	5 μs	20 ns	20 ns
Source	7 V	0 s	1 ms	20 ns	20 ns
Gate 2	−3 V	0 V	1 ms	20 ns	20 ns

```
100   REAL Ampl(1:4),Width(1:4),Delay(1:4),Edge1(1:4),Edge2(1:4)
110   REAL Iterate
120   INTEGER Gate,Source,Drain,Subs
130   !
140   Iterate=100000
150   Gate=1
160   Source=5
170   Drain=9
180   Subs=13
190   Ampl(1)=12.
200   Ampl(2)=15.
210   Ampl(3)=7.
220   Ampl(4)=-3
230   Width(1)=1.E-5
240   Width(2)=5.E-6
250   Width(3)=1.E-3
260   Width(4)=1.E-3
270   Delay(1)=0.
280   Delay(2)=1.E-6
290   Delay(3)=0.
300   Delay(4)=0.
310   Edge1(1)=2.E-8
320   Edge1(2)=2.E-8
330   Edge1(3)=2.E-8
340   Edge1(4)=2.E-8
350   Edge2(1)=2.E-8
360   Edge2(2)=2.E-8
370   Edge2(3)=2.E-8
380   Edge2(3)=2.E-8
390   !
400   Init_system
410   Init_pg
420   Set_fr(Source,Gate,Drain,Subs)
430   Set_relay_mode(1)
440   Repeat_nor2(Iterate,Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
450   !
460   END
```

Repeat_nor3

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram continuously executes the write and erase operation of the flash memory cell (low-power NOR type) using a multiplexer switch for gate input.



Item	Description/Defaults	Range Restrictions
number of periods	numeric expression default= 1	1 to 10000000[1]
pulse amplitude name	numeric array name: [V]	any valid name −19 to −2E−1, 2E−1 to 19, resolution: 2.0E−2
pulse width name	numeric array name: [s]	any valid name 3.3E−9 to 9.99E−1, resolution: 3 digits
delay time name	numeric array name: [s] default= 0	any valid name 0 to 9.99E−1, resolution: 3 digits
leading-edge transition time name	numeric array name: [s] default= 2.0E−8	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time name	numeric array name: [s] default= leading-edge transition time	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits

1 If non-integer is specified, fractional part is truncated.

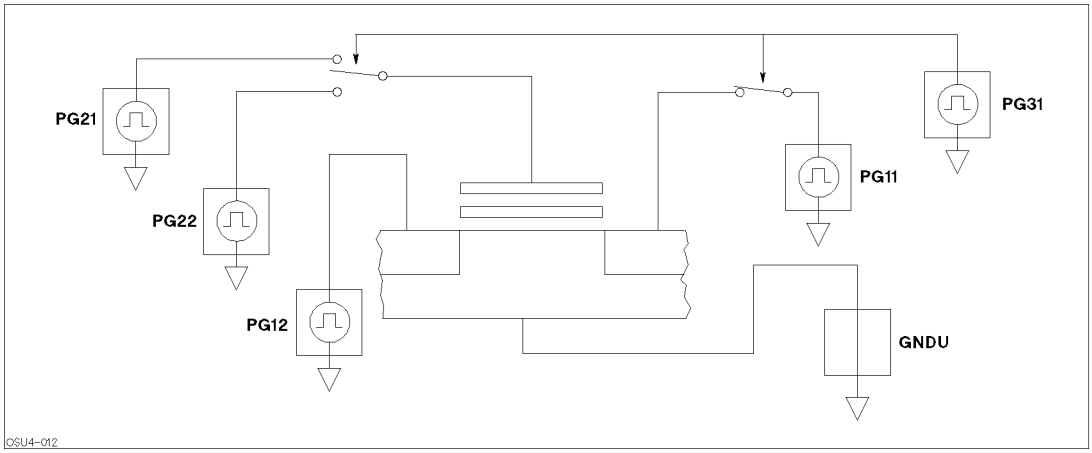
Example Statements

```
Repeat_nor3(Iterate,Ampl(*),Width(*),Delay(*),Leading(*),Trailing(*))
Repeat_nor3(N,Amplitude(*),Width(*))
```

Description

This subprogram continuously executes the write and erase operation of the flash memory cell (low-power NOR type) the specified number of times (*number of periods*).

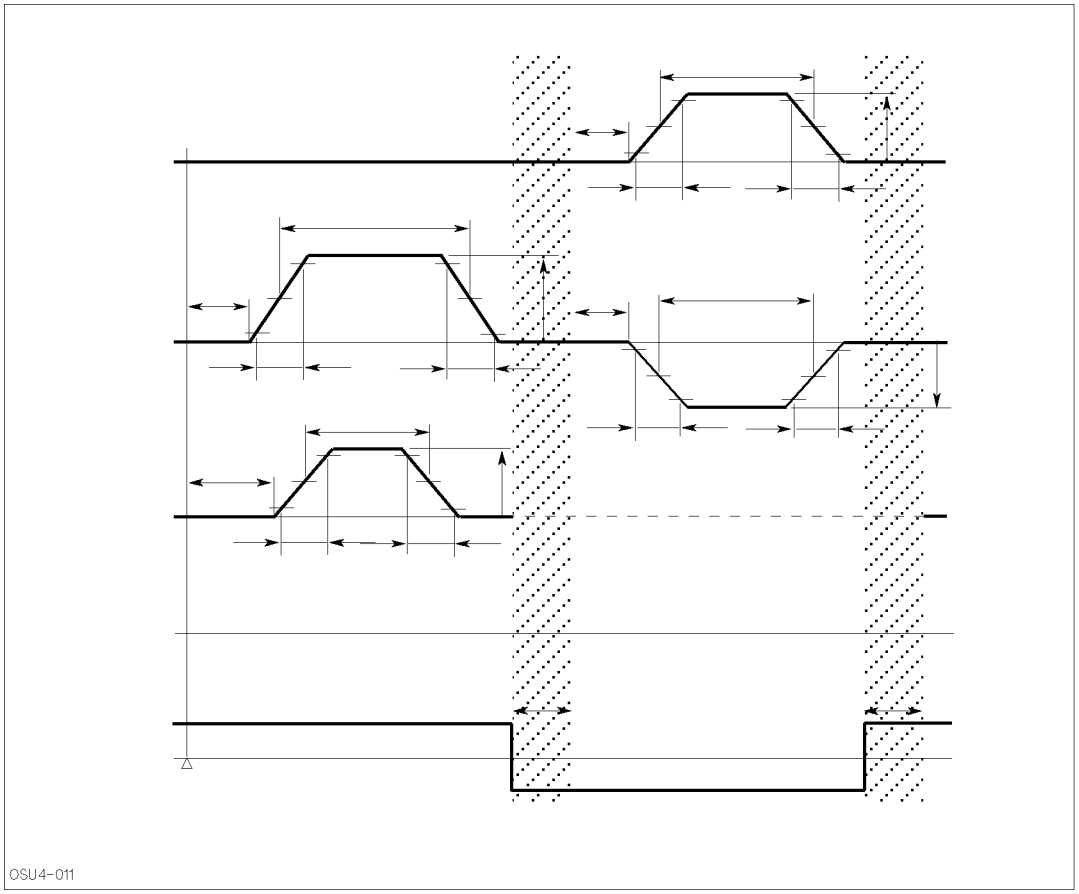
Repeat_nor3 connects the measurement pins specified in Set_fr subprogram as in following figure.



Attach PG11 cable to input of open/close switch. And attach output of open/close switch to HF port that corresponds to drain pin.
Attach PG12 cable to HF port that corresponds to source pin.
Attach PG21 and PG22 cables to input A and B of multiplexer, respectively. And attach output of multiplexer to HF port that corresponds to gate pin.
Attach PG31 cable to switch control for multiplexer and open/close switches.

Make sure the measurement pins connected to the gate, source, and drain are connected to different HF ports. For example, connect gate to one of pins 1 to 4 (which is HF1), connect source to one of pins 5 to 8 (which is HF2), and connect drain to one of pins 9 to 12 (which is HF3).

This subprogram continuously forces the following pulses to the flash memory cell, which writes and erases the cell. First, pulses are forced to the gate and drain, which injects the charge carrier into the floating gate (write operation). Then, pulses (with different level and polarity) are forced to the source and gate with drain open, which removes the charge carriers from the floating gate (erase operation).



After the pulse outputs to the gate and drain are completed, the multiplexer switch starts to switch and the open/close switch starts to open, which takes 2 ms to stabilize. The point at which the switches become stable is defined as the time origin for the output to source and gate. After pulse output to source and gate is completed, the multiplexer and open/close switches start to return to the previous state, which takes 2 ms. Set_relay_mode subprogram must execute before Repeat_nor3 to specify the use of open/close switch.

Element 1 of each array is for first gate, element 2 is for drain, element 3 is for source, element 4 is for second gate, and other elements are ignored. So, the size of each array must be 4 at least.

See Also

Set_fr
Set_multifr
Set_relay_mode
Erase_nor3
Write_nor3

Programming Example

This example program writes and erases the low-power NOR memory cell by using the following parameters:

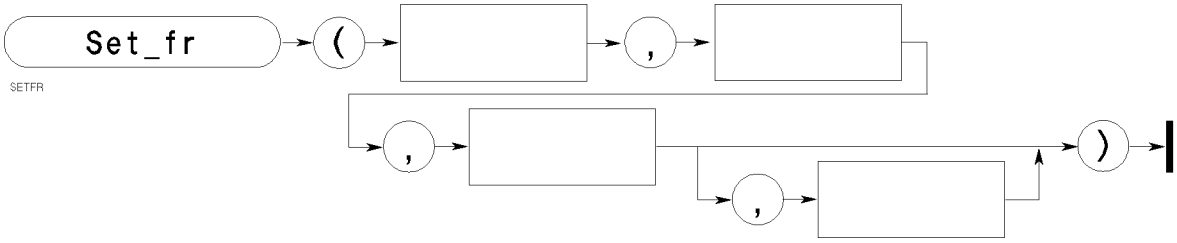
	Pulse Voltage	Delay Time	Pulse Width	Rise Time	Fall Time
Gate 1	12 V	0 s	10 μs	20 ns	20 ns
Drain	15 V	1 μs	5 μs	20 ns	20 ns
Source	7 V	0 s	1 ms	20 ns	20 ns
Gate 2	−3 V	0 s	1 ms	20 ns	20 ns

```
100   REAL Ampl(1:4),Width(1:4),Delay(1:4),Edge1(1:4),Edge2(1:4)
110   REAL Iterate
120   INTEGER Gate,Source,Drain,Subs
130   !
140   Iterate=100000
150   Gate=1
160   Source=5
170   Drain=9
180   Subs=13
190   Ampl(1)=12.
200   Ampl(2)=15.
210   Ampl(3)=7.
220   Ampl(4)=-3
230   Width(1)=1.E-5
240   Width(2)=5.E-6
250   Width(3)=1.E-3
260   Width(4)=1.E-3
270   Delay(1)=0.
280   Delay(2)=1.E-6
290   Delay(3)=0.
300   Delay(4)=0.
310   Edge1(1)=2.E-8
320   Edge1(2)=2.E-8
330   Edge1(3)=2.E-8
340   Edge1(4)=2.E-8
350   Edge2(1)=2.E-8
360   Edge2(2)=2.E-8
370   Edge2(3)=2.E-8
380   Edge2(3)=2.E-8
390   !
400   Init_system
410   Init_pg
420   Set_fr(Source,Gate,Drain,Subs)
430   Set_relay_mode(1)
440   Repeat_nor3(Iterate,Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
450   !
460   END
```

Set_fr

Supported on	4062F
File	PARA.F
Revision	F.06.30
Control	——

This subprogram specifies the measurement pins that will be connected to your device under test (DUT: flash memory cell), and passes the pin numbers to the subprograms for flash memory measurements.



Item	Description/Defaults	Range Restrictions
source pin number	integer expression	1 to 48
gate pin number	integer expression	1 to 48
drain pin number	integer expression	1 to 48
substrate pin number	integer expression default= 0 (no connection)	1 to 48

Example Statements

```
Set_fr(Source,Gate,Drain)
Set_fr(Source,Gate,Drain,Substrate)
```

Description

This subprogram specifies the measurement pins that will be connected to your device under test (DUT: flash memory cell), and passes the pin numbers to the subprograms for flash memory measurements.

Source pin number is the pin number for source terminal, *gate pin number* is pin number for gate terminal, *drain pin number* is pin number for drain terminal, and *substrate pin number* is pin number for substrate terminal.

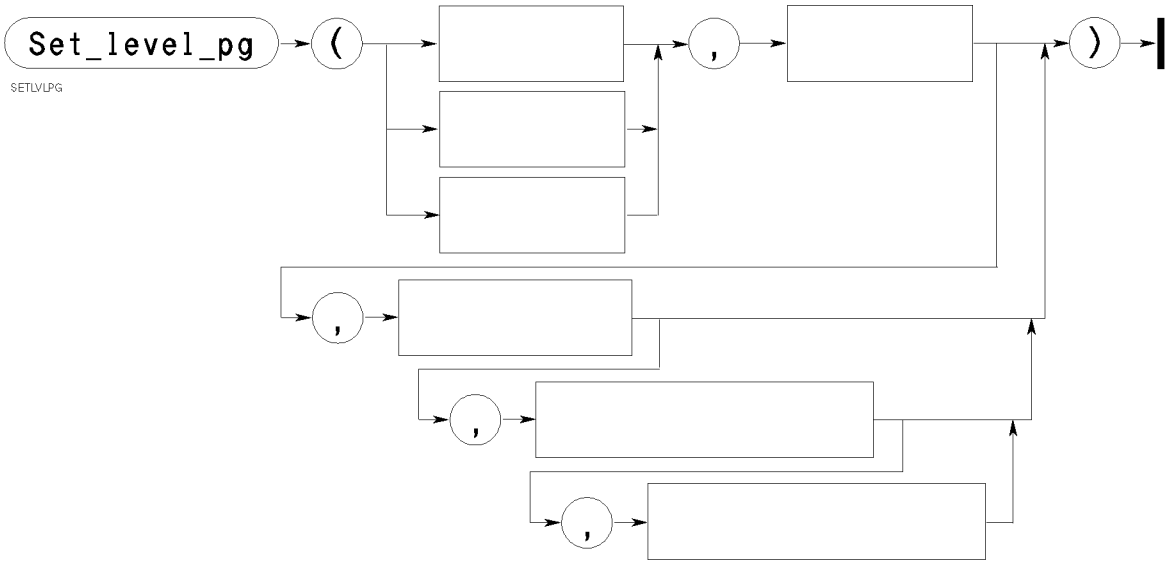
If the *substrate pin number* is omitted, the PGU or GNDU corresponding to the substrate is not connected.

If the Set_multifr subprogram is executed, the pin assignments specified by the Set_fr subprogram are canceled.

Set_level_pg

Supported on 4062F
File TIS.F
Revision F.06.30
Control PGU

This subprogram sets the voltage and impedance parameters of pulse that will be forced from the specified PGU.



Item	Description/Defaults	Range Restrictions
port address	integer expression	32741 to 32752
unit address	integer expression	32581 to 32590
pin number	integer expression	1 to 48[1]
pulse amplitude 1	numeric expression: [V]	−19.0 to 19.0
pulse amplitude 2	numeric expression: [V] default= pulse amplitude 1	−19.0 to 19.0[2]
pulse base	numeric expression: [V] default= 0	−19.0 to 19.0
load impedance	numeric expression: [Ω] default= last setting	2.5 to 999000

1 Must be the number of a measurement pin connected to the desired PGU.

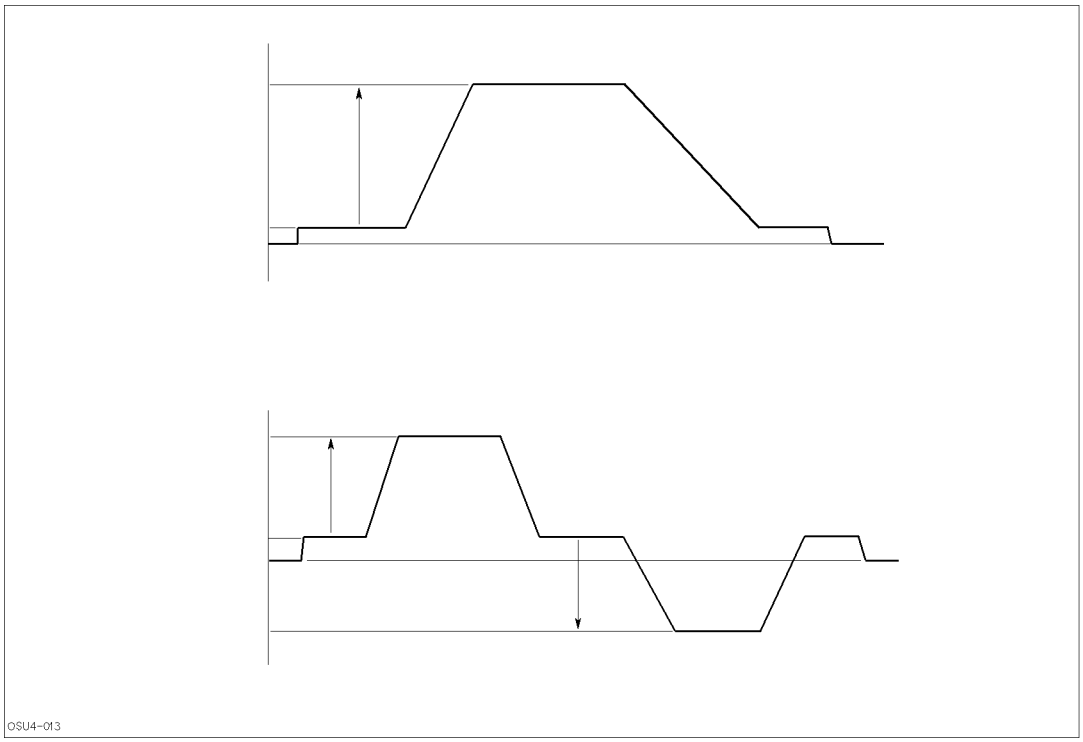
2 For 2-level pulse output mode, this parameter is ignored.

Example Statements

```
Set_level_pg(FNHf(3),7.)
Set_level_pg(FNPort(41),7.,-3.,0.)
```

Description

This subprogram sets the pulse base and pulse amplitudes of pulse that will be forced from the PGU specified by the *port address*, *unit address*, or *pin number*, and specifies the load impedance connected to the specified PGU.



Pulse amplitude is the amplitude from the base voltage (pulse base) to the peak voltage. The meaning of *pulse amplitude 1* and *pulse amplitude 2* depends on the output mode of the specified PGU and whether it is specified by the port address, unit address, or pin number as follows:

- If the specified PGU is set to 2-level pulse output mode:

  Pulse amplitude 1 and *pulse base* decide the pulse voltage. If the *pulse base* is omitted, the default is zero. If *pulse amplitude 2* is specified, it is ignored.

- If specified PGU is set to "3-level pulse with one PGU" output mode:
  or
  If port address or pin number specifies PGU that is set to "3-level pulse using the multiplexer switch" output mode:

  Pulse amplitude 1, *pulse amplitude 2* and *pulse base* decide the pulse voltage. If *pulse base* is omitted, the default is zero. If *pulse amplitude 2* is omitted, the default is same as *pulse amplitude 1*.

- If unit address specifies PGU that is set to "3-level pulse using the multiplexer switch" output mode:

  This specifies the PGU directly, so you need to execute Set_level_pg independently for each PGU. For each PGU, *pulse amplitude 1* and *pulse base* decides the pulse voltage forced from the PGU. If the *pulse base* is omitted, the default is zero. If the *pulse amplitude 2* is specified, it is ignored. Note that the pulse base values for two PGUs must be the same.

If you execute this subprogram for a PGU used for switch control or for OUTPUT 2 unit of the "3-level pulse with one PGU" output mode, an error occurs.

For "2-level pulse" output mode or "3-level pulse using the multiplexer switch" output mode, the following conditions must be satisfied:

	50 Ω Load	**999 kΩ Load**
Pulse Amplitude 1, 2	100 mV to 10.0 V, −10.0 V to −100 mV	200 mV to 19.0 V, −19.0 V to −200 mV
Maximum Pulse Level	−9.90 V to 10.0 V	−18.8 V to 19.0 V
Minimum Pulse Level	−10.0 V to 9.90 V	−19.0 V to 18.8 V

For "3-level pulse with one PGU" output mode, the following conditions must be satisfied:

	50 Ω Load	**999 kΩ Load**
Total of |Pulse Amplitude 1, 2|	200 mV to 19.5 V	400 mV to 20.0 V
Pulse Amplitude 1, 2	100 mV to 9.92 V, −9.92 V to −100 mV	200 mV to 19.0 V, −19.0 V to −200 mV
Maximum Pulse Level	−9.82 V to 9.92 V	−18.8 V to 19.0 V
Minimum Pulse Level	−9.92 V to 9.82 V	−19.0 V to 18.8 V
Pulse Base	−9.92 V to 9.92 V	−19.0 V to 19.0 V

load impedance is the impedance value connected to the device under test (DUT). If the actual load impedance is not the same as the *load impedance* setting, the actual pulse voltage forced to the DUT is different from the pulse voltage setting. If the *load impedance* is omitted, the previous setting is used.

Set_mode_pg

Supported on 4062F
File TIS.F
Revision F.06.30
Control PGU

This subprogram sets the time accuracy mode, which affects the output pulse and switch control signal.

Item	Description/Defaults	Range Restrictions
time accuracy mode	integer expression	0: normal 1: pattern

Example Statements

```
Set_mode_pg(1)
```

Description

This subprogram sets the time accuracy mode, which affects the output pulse and switch control signal. The default setting is normal mode.

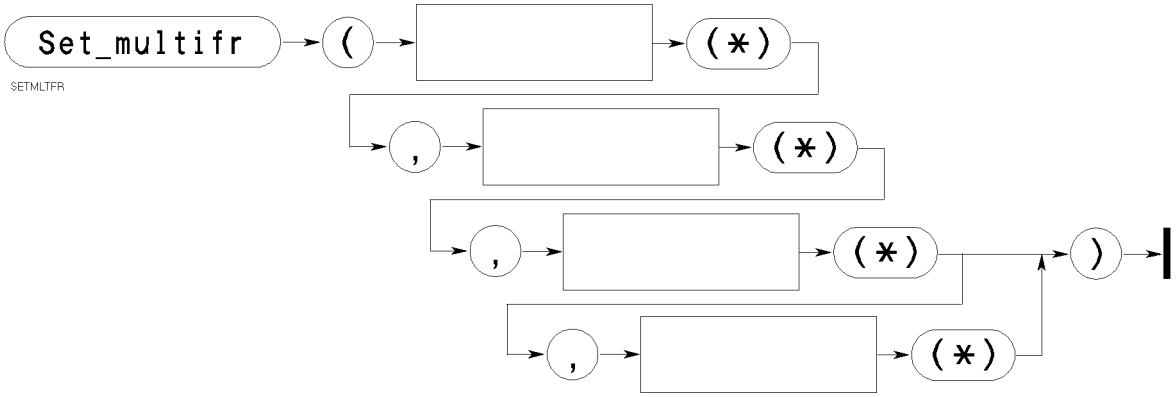
If time accuracy mode is set to "pattern", more accurate pulse wave is forced. For relatively long pulse periods, you can set more accurate pulse width and delay time in the pattern mode. In pattern mode, PG2 to PG5 can be synchronized, but PG1 cannot.

See Appendix C about the actual accuracy.

Set_multifr

Supported on 4062F
File PARA.F
Revision F.06.30
Control ——

This subprogram specifies several sets of measurement pins that will be connected to your devices under test (DUTs: flash memory cells), and passes the pin numbers to the subprograms for flash memory measurements.



Item	Description/Defaults	Range Restrictions
source pin number name	integer array name	any valid name, 1 to 48
gate pin number name	integer array name	any valid name, 1 to 48
drain pin number name	integer array name	any valid name, 1 to 48
substrate pin number name	integer array name default= 0 (no connection)	any valid name, 1 to 48

Example Statements

```
Set_multifr(Source(*),Gate(*),Drain(*))
Set_multifr(Source(*),Gate(*),Drain(*),Substrate(*))
```

Description

This subprogram specifies several sets of measurement pins that will be connected to your devices under test (DUTs: flash memory cells), and passes the pin numbers to the subprograms for flash memory measurements. This subprogram is useful if you will force the same pulse to several DUTs.

source pin number array has the pin numbers for source terminals, *gate pin number* array has the pin numbers for gate terminals, *drain pin number* array has the pin numbers for drain terminals, and *substrate pin number* array has the pin numbers for substrate terminals. Zero (`0`) to ignore (open, not connected to DUT).

If the *substrate pin number* is omitted, the PGU or GNDU corresponding to the substrate is not connected.

If the Set_fr subprogram is executed, the pin assignments specified by the Set_multifr subprogram are canceled.

Programming Example

The following program writes and erases three NOR-type memory cells simultaneously, using the Repeat_nor1 subprogram.

```
100   REAL Ampl(1:3),Width(1:3),Delay(1:3),Edge1(1:3),Edge2(1:3)
110   REAL Iterate
120   INTEGER Gate(1:3),Source(1:3),Drain(1:3),Subs(1:3)
130   !
140   Iterate=100000
150   Gate(1)=1
160   Gate(2)=2
170   Gate(3)=3
180   Source(1)=5
190   Source(2)=6
200   Source(3)=7
210   Drain(1)=9
220   Drain(2)=10
230   Drain(3)=11
240   Subs(1)=13
250   Subs(2)=14
260   Subs(3)=15

.  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .

540   !
550   Init_system
560   Init_pg
570   Set_multifr(Source(*),Gate(*),Drain(*),Subs(*))
580   Set_relay_mode(1)
590   Repeat_nor1(Iterate,Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
600   !
610   END
```

Set_relay_mode

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram sets whether or not the measurement program uses the open/close or multiplexer switches in the subsequent PARA subprograms.

Item	Description/Defaults	Range Restrictions
mode	integer expression	0: Do not use switches. 1: Use switches.

Example Statements

```
Set_relay_mode(1)
Set_relay_mode(Mode)
```

Description

This subprogram sets whether or not the measurement program uses the open/close or multiplexer switches in the subsequent PARA subprograms.

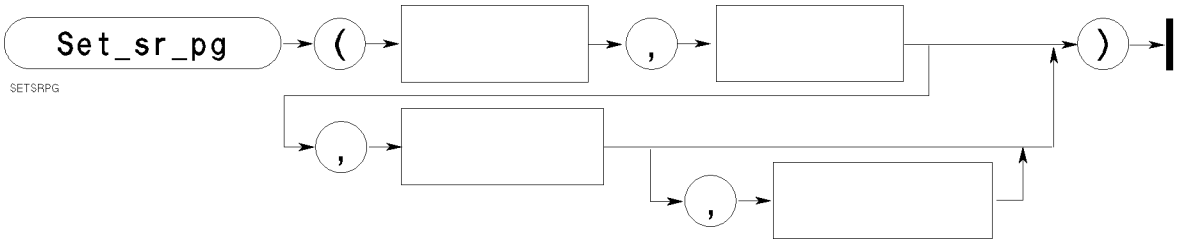
For example, if a measurement program continuously writes and erases NOR-type flash memory cells, the open/close switch is required. So, it is necessary to enable the switches by using this subprogram.

Default setting of this subprogram is 0 (do not use switches).

Set_sr_pg

Supported on 4062F
File TIS.F
Revision F.06.30
Control PGU

This subprogram specifies settings of PGU that controls SW1 to SW4 on cable adapter.



Item	Description/Defaults	Range Restrictions
unit address	integer expression	32581 to 32590
usage mode	integer expression	0: Reset the specified unit. 1: Independent switching. 2: Switching synchronized with output pulse.
pulse width	numeric expression: [s]	2.0E−3 to 9.99E−1 resolution: see Description
delay time	numeric expression: [s] default= 0	0 to 9.99E−1 resolution: see Description

Example Statements

```
Set_sr_pg(FNPg(12),2,1.E-2,1.E-2)
Set_sr_pg(FNPg(31),0)
```

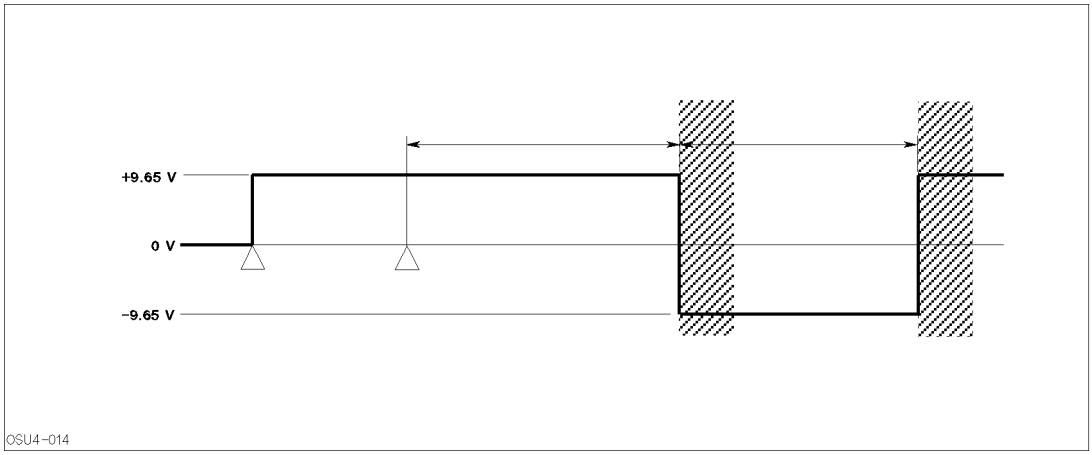
Description

This subprogram specifies PGU that controls switches (SW1 to SW4) on cable adapter, and specifies how switching will be controlled. So specified PGU must be connected to SW1·2·3 or SW4 terminal of CTRL area. Note that if a PGU is used for switch control, then it cannot be assigned to an HF port in /usr/pcs/etc/FLCONFIG file.

After Set_sr_pg subprogram executes, open/close switches are set to state indicated by MODE SELECTORs, and multiplexer switches are set so that input A is connected to output.

Before Set_sr_pg is executed, or after it is executed with *usage mode*=0, open/close switches (SW2 and SW3) are set to open state, multiplexer switch SW1 is set to high-impedance state, and multiplexer switch SW4 is set so that input A is connected to output.

If *usage mode* = 1 (independent switching), *pulse width* and *delay time* are ignored. To switch the switches, you can use Connect_sr subprogram. For details, see Connect_sr subprogram.
If *usage mode* = 2 (switching synchronized with output pulse), you must set the *pulse width*. The relations between *pulse width*, *delay time*, and switching time are as follows:



After the trigger (Force_pg or subprogram that calls Force_pg), switching occurs after the delay time. During the delay time, the open/close switches are in the state indicated by the MODE SELECTORs, and the multiplexer switches have input A connected to output. After the delay time, the switches start to switch, and become stable after 2 ms. After pulse width time, the switches start to return to the previous states, and become stable after 2 ms.

See Appendix C about the setting resolutions of the *pulse width* and *delay time*.

Note

If voltage is forced during the switching, the voltage forced to the DUT may be unstable. Hewlett-Packard recommends to not force voltage during the switching.

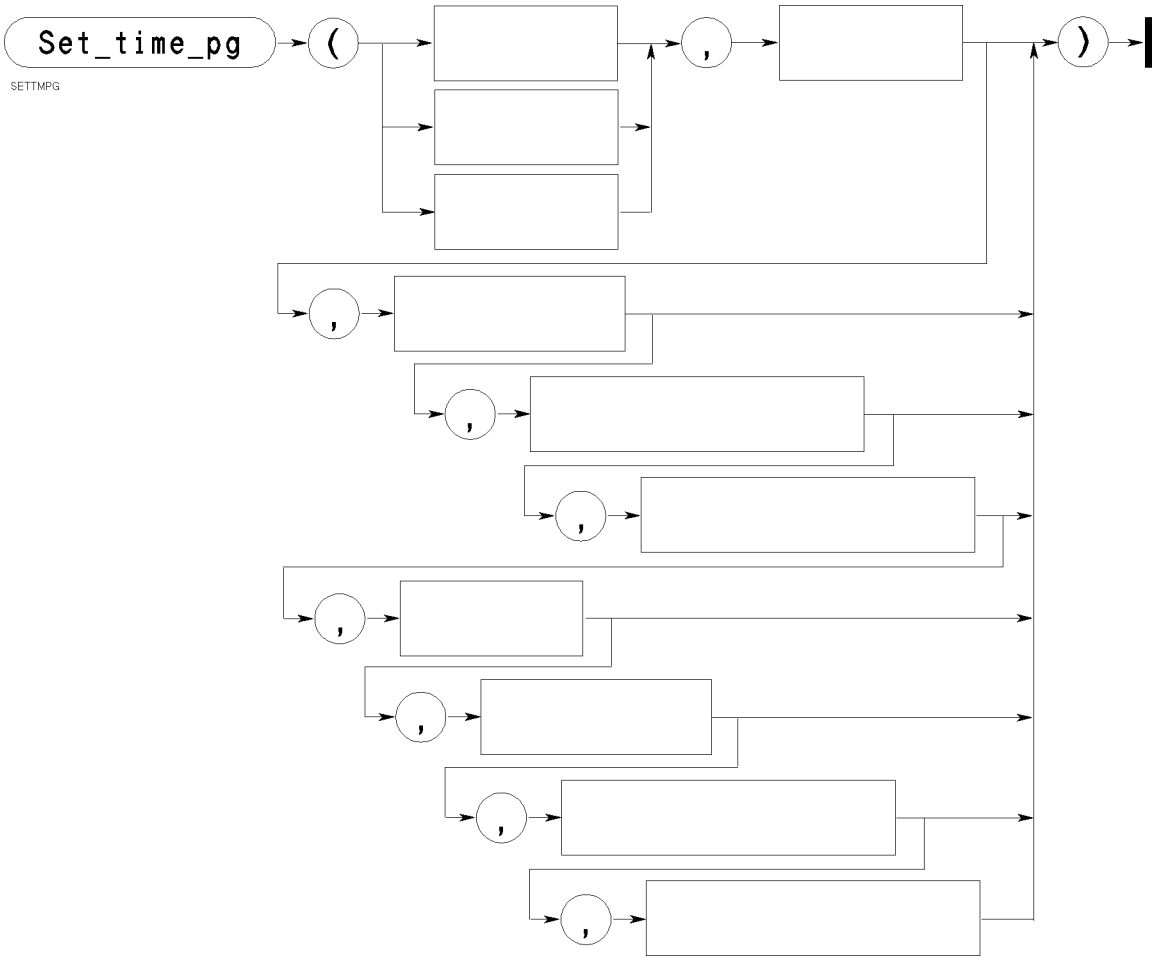
If *usage mode* = 0 (reset the specified unit), the specified PGU is set to the following:

Output value:	0 V
Time accuracy mode:	Previous setting
Output mode:	2-level pulse output
Pulse base:	0 V
Pulse amplitude 1:	0.2 V
Pulse width 1:	500 ns
Delay time:	0
Leading-edge transition time 1:	20 ns
Trailing-edge transition time 1:	20 ns
Pulse amplitude 2:	—
Pulse width 2:	—
Delay time 2:	—
Leading-edge transition time 2:	—
Trailing-edge transition time 2:	—
Load impedance:	999 kΩ

Set_time_pg

Supported on	4062F
File	TIS.F
Revision	F.06.30
Control	PGU

This subprogram sets the pulse timing parameters (such as pulse width, delay time, and transition time) of pulse that will be forced from the specified PGU.



Item	Description/Defaults	Range Restrictions
port address	integer expression	32741 to 32752
unit address	integer expression	32581 to 32590
pin number	integer expression	1 to 48[1]
pulse width 1	numeric expression: [s]	3.3E−9 to 9.99E−1 resolution: see Description
delay time 1	numeric expression: [s] default= 0	0 to 9.99E−1 resolution: see Description
leading-edge transition time 1	numeric expression: [s] default= 2E−8	2.0E−9 to 2.0E−1 resolution: 3 digits
trailing-edge transition time 1	numeric expression: [s] default= leading-edge transition time 1	2.0E−9 to 2.0E−1 resolution: 3 digits
pulse width 2	numeric expression: [s][2]	0 to 9.99E−1 resolution: see Description
delay time 2	numeric expression: [s] default= pulse width 1 + delay time 1	0 to 9.99E−1 resolution: see Description
leading-edge transition time 2	numeric expression: [s] default= leading-edge transition time 1	2.0E−9 to 2.0E−1 resolution: 3 digits
trailing-edge transition time 2	numeric expression: [s] default= leading-edge transition time 2	2.0E−9 to 2.0E−1 resolution: 3 digits

1 Must be the number of a measurement pin connected to the desired PGU.

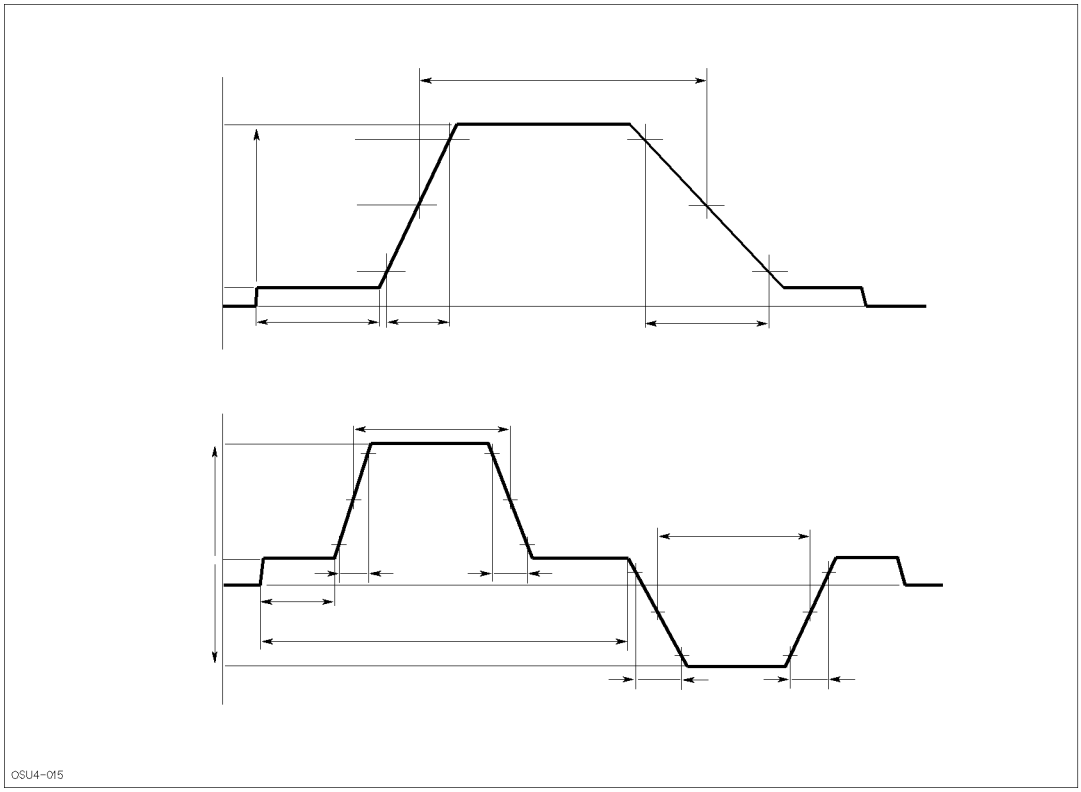
2 Necessary for 3-level pulse output.

Example Statements

```
Set_time_pg(FNHf(1),1.E-6)
Set_time_pg(FNPort(41),1.E-6,0,5.E-8,5.E-8,1.E-6,1.E-2,5.E-8,5.E-8)
```

Description

This subprogram sets the pulse timing parameters for PGU specified by *port address*, *unit address*, or *pin number*. The following figure shows the definition of timing parameters:



Pulse width	Time from 50% of pulse amplitude of leading-edge to 50% of pulse amplitude of trailing-edge.
Delay time	Time from the start of a pulse period to the start of the leading-edge of the pulse.
Leading-edge transition time	The transition time measured from 10% to 90% of pulse amplitude.
Trailing-edge transition time	The transition time measured from 90% to 10% of pulse amplitude.

The parameters you specify depend on the output mode of the specified PGU and whether the PGU is specified by the *port address*, *unit address*, or *pin number* as follows:

- If specified PGU is set to "2-level pulse" output mode:

  pulse width 1, *delay time 1*, *leading-edge transition time 1*, and *trailing-edge transition time 1* decide the timing of the output pulse. If *pulse width 2* is specified, an error occurs.

- If specified PGU is set to "3-level pulse with one PGU" output mode:
  or
  If *port address* or *pin number* specifies PGU that is set to "3-level pulse using the multiplexer switch" output mode:

  All the timing parameters decide the timing of the output pulse. Note that *delay time 2* must be greater or equal to the sum of *pulse width 1* and *delay time 1*. The *pulse width 2* cannot be omitted.

- If *unit address* specifies PGU that is set to the "3-level pulse using the multiplexer switch" output mode:

  This specifies the PGU directly, so you need to execute Set_time_pg to independently set the timing parameters for each PGU. You set the *pulse width 1*, *delay time 1*, *leading-edge transition time 1*, and *trailing-edge transition time 1* for each PGU. Note that if you specify *pulse width 2*, an error occurs.

The resolution of *pulse width* and *delay time* depends on the time accuracy mode as follows:

Time accuracy mode	Resolution
Normal mode	3 digits, min. 10 ps
Pattern mode	PG1: 3 digits, min. 10 ps. PG2 to PG5[1]: For $1000\times10^{-n} < T \leq 4000\times10^{-n}$, resolution is 1×10^{-n} For $4000\times10^{-n} < T \leq 8000\times10^{-n}$, resolution is 2×10^{-n} For $8000\times10^{-n} < T \leq 10000\times10^{-n}$, resolution is 2.5×10^{-n} Where, T means pulse period specified in the Force_pg subprogram. If not specified, T is the maximum value of 1.1×(pulse width + delay time + 30 ns) among all specified PGUs.

1 For example, if T is 250 μs, then according to above table, the resolution is 0.1 μs (1×10^{-7}) because $T = 250\times10^{-6} = 2500\times10^{-7}$.

The leading-edge and trailing-edge transition times for a pulse must both be in one of the following ranges:

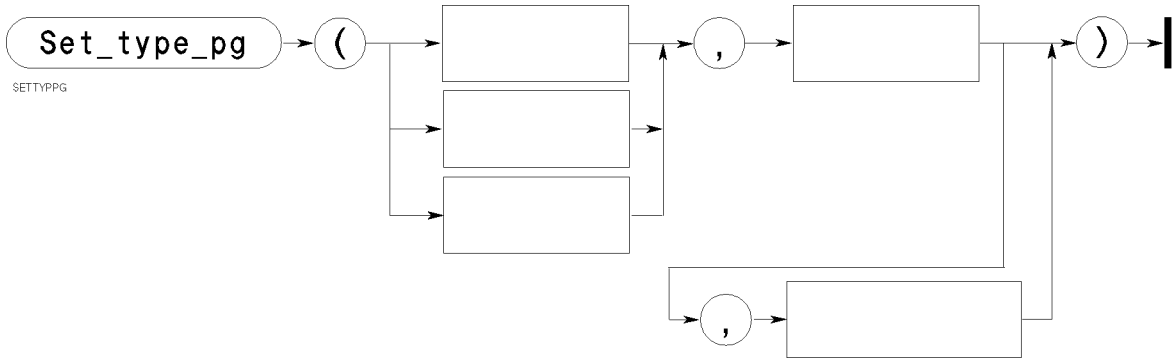
2 ns – 20 ns	10 μs – 200 μs
10 ns – 200 ns	100 μs – 2 ms
100 ns – 2 μs	1 ms – 20ms
1 μs – 20 μs	10 ms – 200ms

For example, if leading-edge transition time is set to 5 μs, the trailing-edge transition time must be in the range of 1 μs to 20 μs.

Set_type_pg

Supported on 4062F
File TIS.F
Revision F.06.30
Control PGU

This subprogram sets the output mode for a specified unit, port, or pin.



Item	Description/Defaults	Range Restrictions
port address	integer expression	32741 to 32752
unit address	integer expression	32581 to 32590
pin number	integer expression	1 to 48[1]
output mode	integer expression	1: "2-level pulse" output mode 2: "3-level pulse with one PGU" output mode 3: "3-level pulse using the multiplexer" output mode
unit address 2	integer expression[2]	32581 to 32590

1 Must be the number of a measurement pin connected to the desired PGU.

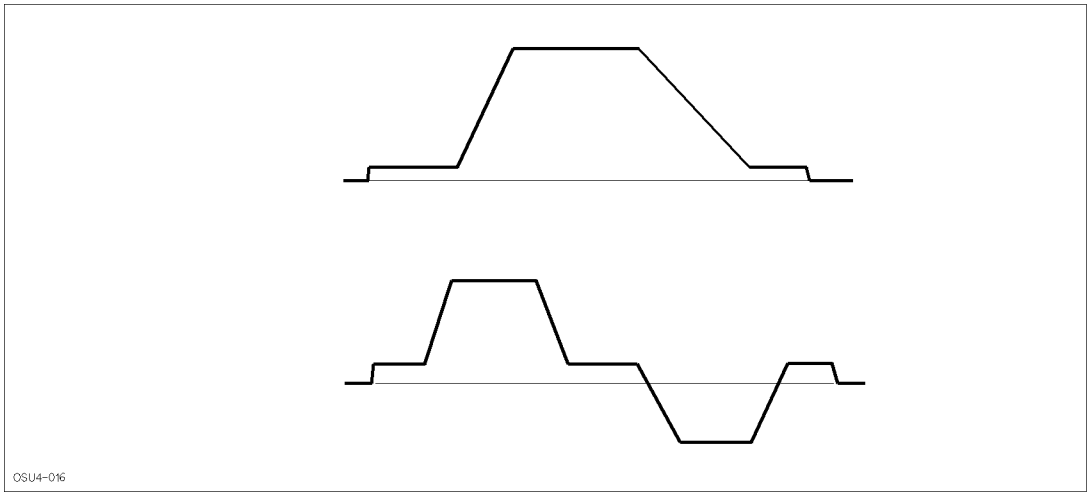
2 If the *output mode* is 1 or 2, this unit address setting is ignored.

Example Statements

```
Set_type_pg(FNHf(2),1)              "2-level pulse" mode
Set_type_pg(Gate,2)                 "3-level pulse with one PGU" mode
Set_type_pg(FNPg(21),3,FNPg(22))    "3-level pulse using the multiplexer" mode
```

Description

The pulse generator can force two types of pulse wave: 2-level pulse output and 3-level pulse output. The following figure shows these two types:



One PGU can force 2-level pulse.

If you use the "3-level pulse with one PGU" output mode, the output terminal (PGU) specified by the *port address*, *unit address*, or *pin number* must be OUTPUT 1. Note that the OUTPUT 2 terminal is disabled.

For example, if you want to force a 3-level pulse from PG2, you can use the following statement:

```
Set_type_pg(FNPg(21),2)
```

If you use the "3-level pulse using the multiplexer switch" output mode, the output terminal (PGU) specified by *port address*, *unit address*, or *pin number* must be connected to input A terminal of SW1 or SW4, and output terminal (PGU) specified by *unit address 2* must be connected to input B terminal of SW1 or SW4. Note that *only* the PGU that is connected to *input A* should be assigned to an HF port in the FLCONFIG file. Do *not* enter an HF assignment for the PGU that is connected to input B.

When Set_type_pg is used to change the *output mode*, the settings are changed as follows:

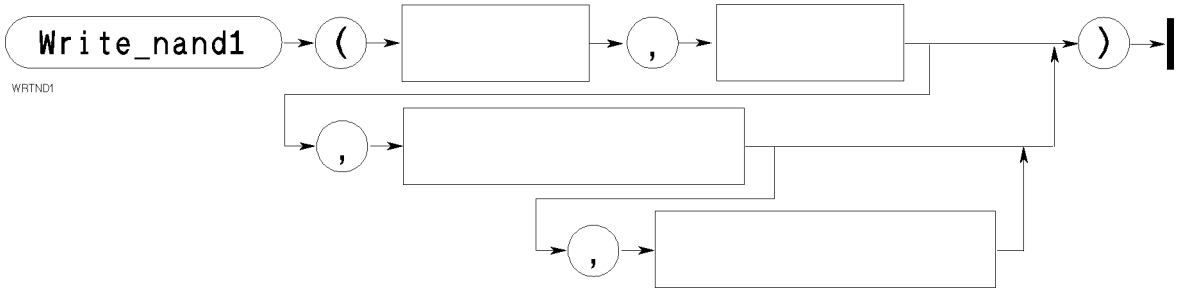
Items	Changed to 3-level Output	Changed to 2-level Output
Output value:	0 V	0 V
Time accuracy mode:	previous setting	previous setting
Output mode:	3-level	2-level
Pulse base:	0 V	0 V
Pulse amplitude 1:	0.2 V	0.2 V
Pulse width 1:	500 ns	500 ns
Delay time 1:	0	0
Leading-edge transition time 1:	20 ns	20 ns
Trailing-edge transition time 2:	20 ns	20 ns
Pulse amplitude 2:	0.2 V	——
Pulse width 2:	500 ns	——
Delay time 2:	500 ns	——
Leading-edge transition time 2:	20 ns	——
Trailing-edge transition time 2:	20 ns	——
Load impedance:	999 kΩ	999 kΩ

When Set_type_pg changes the output mode of the PGU used from "3-level pulse using the multiplexer" to the "2-level pulse" or "3-level pulse with one PGU" output mode, the other PGU is automatically set to 2-level pulse output.

Write_nand1

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram executes the write operation of the flash memory cell (NAND type).



Item	Description/Defaults	Range Restrictions
pulse amplitude	numeric expression [V]	−19 to −2E−1, 2E−1 to 19, resolution: 2.0E−2
pulse width	numeric expression [s]	3.3E−9 to 9.99E−1, resolution: 3 digits
leading-edge transition time	numeric expression [s] default= 2.0E−8	2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time	numeric expression [s] default= leading-edge transition time	2.0E−9 to 2.0E−1, resolution: 3 digits

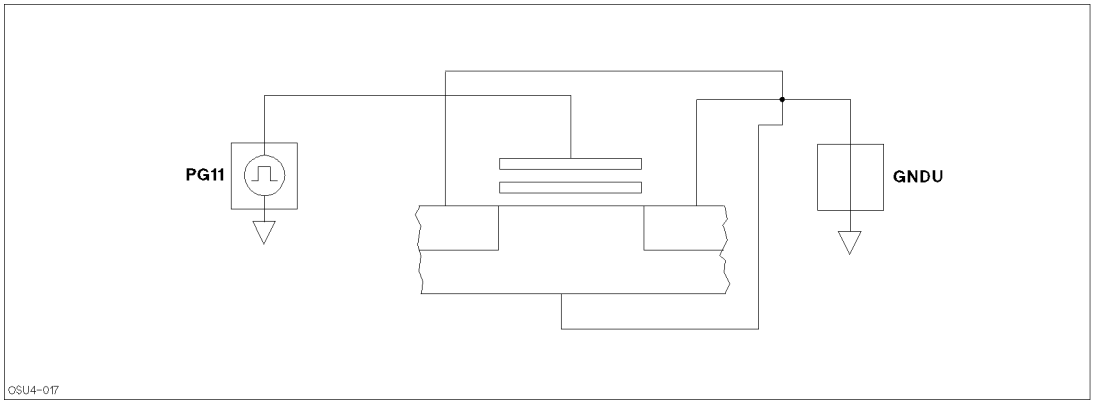
Example Statements

```
Write_nand1(Ampl,Width,Leading,Trailing)
Write_nand1(Amplitude,Width)
```

Description

This subprogram executes the write operation of the flash memory cell (NAND type).

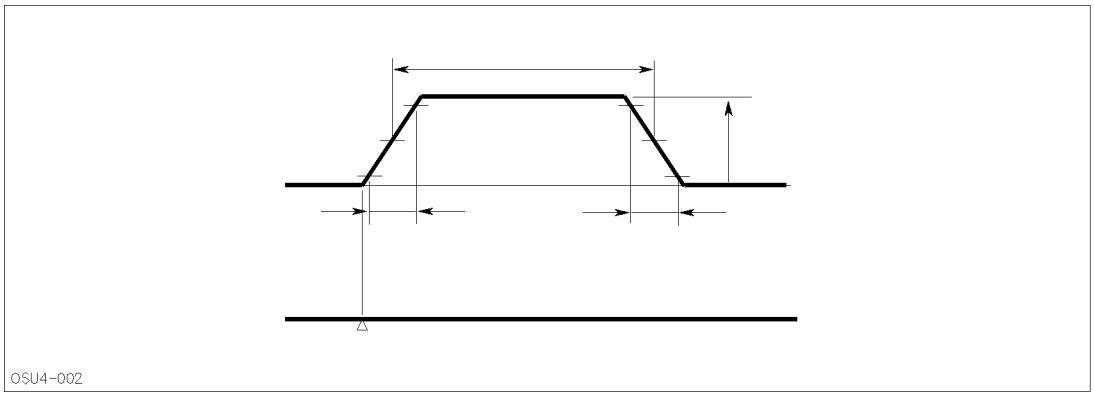
Write_nand1 connects the measurement pins specified in Set_fr subprogram as in following figure.



Attach PG11 cable to the HF port that corresponds to the gate measurement pin.

You can use this subprogram with the Erase_nand1 and Repeat_nand1 subprograms in a measurement program. For the connections, refer to the Repeat_nand1 pages.

This subprogram forces the following pulse to the gate, which writes to the cell. The drain, source, and substrate are connected to circuit common. When a certain voltage value is forced to gate, the charge carriers go through the silicon dioxide and are injected into the floating gate, and are trapped there (write operation).



See Also

Set_fr
Set_multifr
Erase_nand1
Repeat_nand1

Programming Example

This example program executes the write operation of the NAND memory cell by using the following parameters:

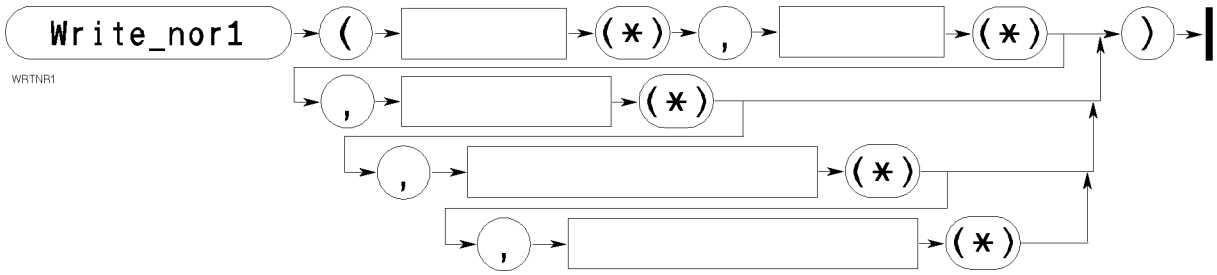
	Pulse Voltage	Pulse Width	Rise Time	Fall Time
Gate	15 V	10 μs	100 ns	20 ns
Source, Drain, Substrate	0 V	—	—	—

```
100   REAL Ampl,Width,Edge1,Edge2
110   INTEGER Gate,Source,Drain,Subs
120   !
130   Gate=1
140   Source=6
150   Drain=7
160   Subs=8
170   Ampl=15.
180   Width=1.E-5
190   Edge1=1.E-7
200   Edge2=2.E-8
210   !
220   Init_system
230   Init_pg
240   Set_fr(Source,Gate,Drain,Subs)
250   Write_nand1(Ampl,Width,Edge1,Edge2)
260   !
270   END
```

Write_nor1

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram executes the write operation of the flash memory cell (normal NOR type).



Item	Description/Defaults	Range Restrictions
pulse amplitude name	numeric array name [V]	any valid name −19 to −2E−1, 2E−1 to 19, resolution: 2.0E−2
pulse width name	numeric array name [s]	any valid name 3.3E−9 to 9.99E−1, resolution: 3 digits
delay time name	numeric array name [s] default= 0	any valid name 0 to 9.99E−1, resolution: 3 digits
leading-edge transition time name	numeric array name [s] default= 2.0E−8	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time name	numeric array name [s] default= leading-edge transition time	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits

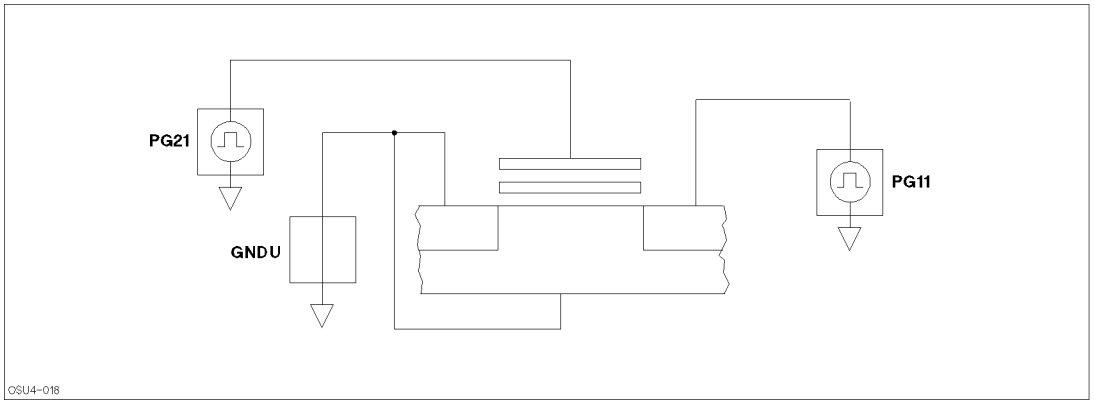
Example Statements

```
Write_nor1(Ampl(*),Width(*),Delay(*),Leading(*),Trailing(*))
Write_nor1(Amplitude(*),Width(*),Delay(*))
```

Description

This subprogram executes the write operation of the flash memory cell (NOR type).

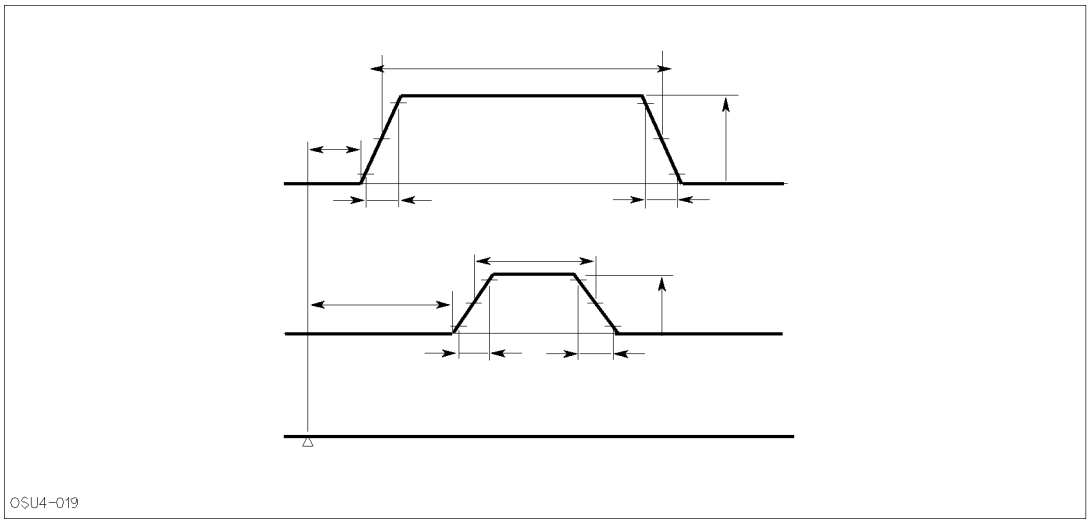
Write_nor1 connects measurement pins specified in Set_fr subprogram as in following figure. Attach PG11 cable to HF port that corresponds to drain measurement pin. Attach PG21 cable to HF port that corresponds to gate measurement pin.



Make sure that the measurement pins connected to the gate and drain are connected to different HF ports. For example, connect gate to one of pins 1 to 4 (which is HF1), and connect drain to one of measurement pins 9 to 12 (which is HF3).

You can use this subprogram with the Erase_nor1 and Repeat_nor1 subprograms in a measurement program. For the connections, refer to the Repeat_nor1 subprogram.

This subprogram forces the following pulses to the gate and drain, which writes to the cell. The source and substrate are connected to the circuit common. While a certain voltage is forced to the gate to make a channel, a voltage is forced to the drain to generate hot carriers near the drain, and hot carriers are injected into the floating gate and trapped there (write operation). Then, the voltages of the drain and gate are set to 0 V.



Element 1 of each array is for gate, element 2 is for drain, and other elements are ignored. So, the size of each array must be 2 or more.

See Also

Set_fr
Set_multifr
Erase_nor1
Repeat_nor1

Programming Example

This example program writes to the NOR memory cell by using the following parameters:

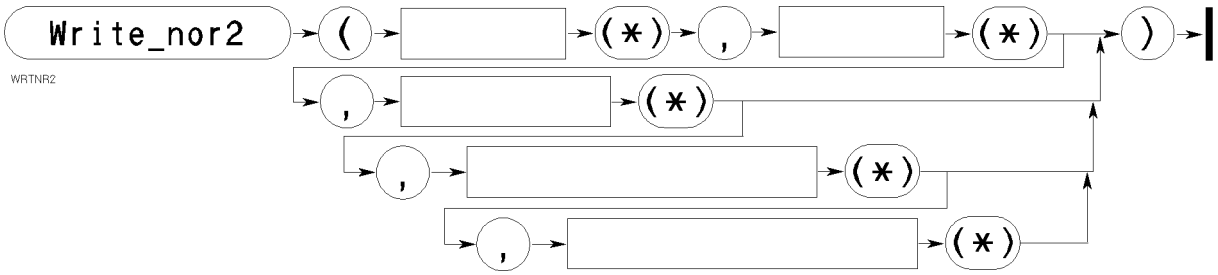
	Pulse Voltage	Delay Time	Pulse Width	Rise Time	Fall Time
Gate	10 V	1 μs	10 μs	100 ns	20 ns
Drain	7 V	4 μs	5 μs	200 ns	40 ns
Source, Substrate	0 V	—	—	—	—

```
100   REAL Ampl(1:2),Width(1:2),Delay(1:2),Edge1(1:2),Edge2(1:2)
110   INTEGER Gate,Source,Drain,Subs
120   !
130   Gate=1
140   Source=5
150   Drain=9
160   Subs=13
170   Ampl(1)=10.
180   Ampl(2)=7.
190   Width(1)=1.E-5
200   Width(2)=5.E-6
210   Delay(1)=1.E-6
220   Delay(2)=4.E-6
230   Edge1(1)=1.E-7
240   Edge1(2)=2.E-7
250   Edge2(1)=2.E-8
260   Edge2(2)=4.E-8
270   !
280   Init_system
290   Init_pg
300   Set_fr(Source,Gate,Drain,Subs)
310   Write_nor1(Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
320   !
330   END
```

Write_nor2

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram executes the write operation of a flash memory cell (low-power NOR type).



Item	Description/Defaults	Range Restrictions
pulse amplitude name	numeric array name: [V]	any valid name −19 to −2E−1, 2E−1 to 19, resolution: 2.0E−2
pulse width name	numeric array name: [s]	any valid name 3.3E−9 to 9.99E−1, resolution: 3 digits
delay time name	numeric array name: [s] default= 0	any valid name 0 to 9.99E−1, resolution: 3 digits
leading-edge transition time name	numeric array name: [s] default= 2.0E−8	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time name	numeric array name: [s] default= leading-edge transition time	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits

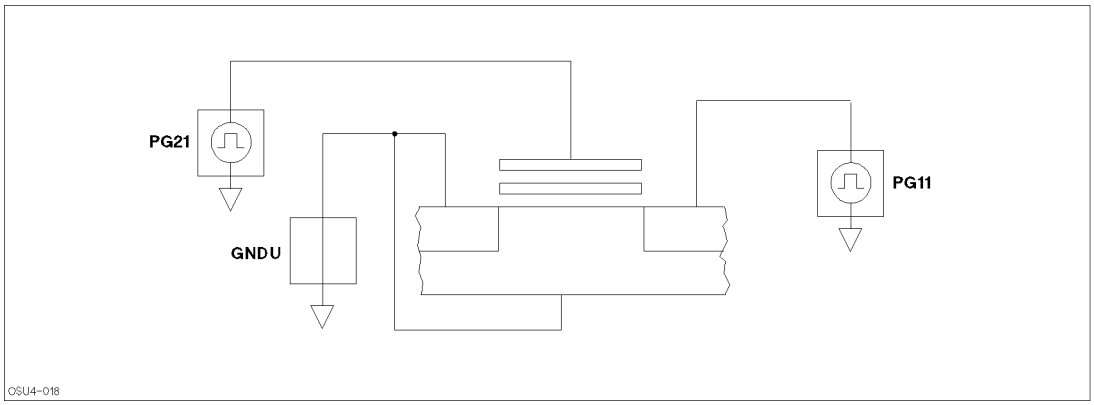
Example Statements

```
Write_nor2(Ampl(*),Width(*),Delay(*),Leading(*),Trailing(*))
Write_nor2(Amplitude(*),Width(*),Delay(*))
```

Description

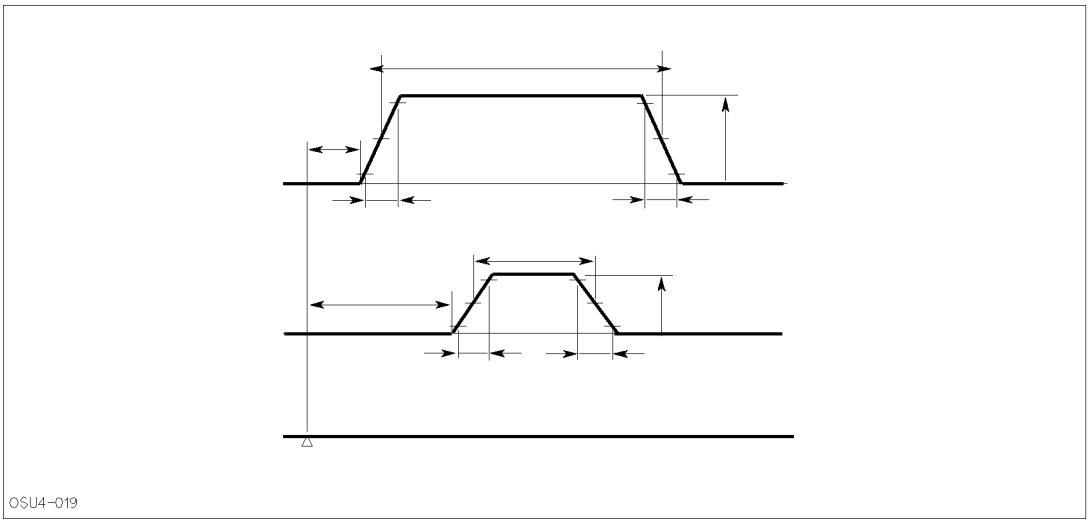
This subprogram executes the write operation of a flash memory cell (low-power NOR type).

Write_nor2 connects measurement pins specified in Set_fr subprogram as in following figure. Attach PGUs as shown in the figure. Make sure that the gate and drain measurement pins are connected to different HF ports. For example, connect gate to one of pins 1 to 4 (which is HF1), and connect drain to one of measurement pins 9 to 12 (which is HF3).



You can use this subprogram with the Erase_nor2 and Repeat_nor2 subprograms in a measurement program. For the connections, refer to the Repeat_nor2 subprogram.

This subprogram forces the following pulses to the gate and drain, which writes to the cell. The source and substrate are connected to the circuit common. While a certain voltage is forced to the gate to make a channel, a voltage is forced to the drain to generate hot carriers near the drain, and hot carriers are injected into the floating gate and are trapped there (write operation). Then, the voltages of the drain and gate are set to 0 V.



Element 1 of each array is for gate, element 2 is for drain, and other elements are ignored. So, the size of each array must be 2 or more.

See Also

Set_fr
Set_multifr
Erase_nor2
Repeat_nor2

Programming Example

This program writes to the low-power NOR memory cell by using the following parameters:

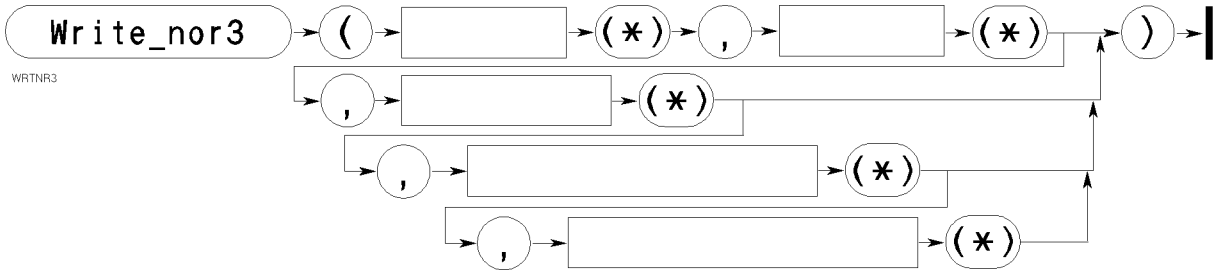
	Pulse Voltage	Delay Time	Pulse Width	Rise Time	Fall Time
Gate	10 V	1 μs	10 μs	100 ns	20 ns
Drain	7 V	4 μs	5 μs	200 ns	40 ns
Source, Substrate	0 V	—	—	—	—

```
100   REAL Ampl(1:2),Width(1:2),Delay(1:2),Edge1(1:2),Edge2(1:2)
110   INTEGER Gate,Source,Drain,Subs
120   !
130   Gate=1
140   Source=5
150   Drain=9
160   Subs=13
170   Ampl(1)=10.
180   Ampl(2)=7.
190   Width(1)=1.E-5
200   Width(2)=5.E-6
210   Delay(1)=1.E-6
220   Delay(2)=4.E-6
230   Edge1(1)=1.E-7
240   Edge1(2)=2.E-7
250   Edge2(1)=2.E-8
260   Edge2(2)=4.E-8
270   !
280   Init_system
290   Init_pg
300   Set_fr(Source,Gate,Drain,Subs)
310   Write_nor2(Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
320   !
330   END
```

Write_nor3

Supported on 4062F
File PARA.F
Revision F.06.30
Control PGU, SWM

This subprogram executes the write operation of a flash memory cell (low-power NOR type) by using a multiplexer switch for gate input.



Item	Description/Defaults	Range Restrictions
pulse amplitude name	numeric array name: [V]	any valid name −19 to −2E−1, 2E−1 to 19, resolution: 2.0E−2
pulse width name	numeric array name: [s]	any valid name 3.3E−9 to 9.99E−1, resolution: 3 digits
delay time name	numeric array name: [s] default= 0	any valid name 0 to 9.99E−1, resolution: 3 digits
leading-edge transition time name	numeric array name: [s] default= 2.0E−8	any valid name 2.0E−9 to 2.0E−1, resolution: 3 digits
trailing-edge transition time name	numeric array name: [s] default= leading-edge transition time	2.0E−9 to 2.0E−1, resolution: 3 digits

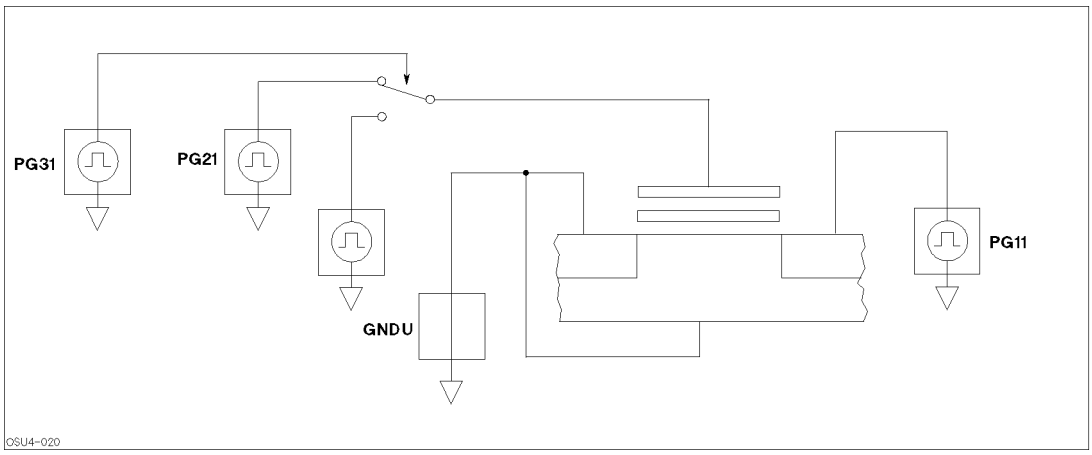
Example Statements

```
Write_nor3(Ampl(*),Width(*),Delay(*),Leading(*),Trailing(*))
Write_nor3(Amplitude(*),Width(*),Delay(*))
```

Description

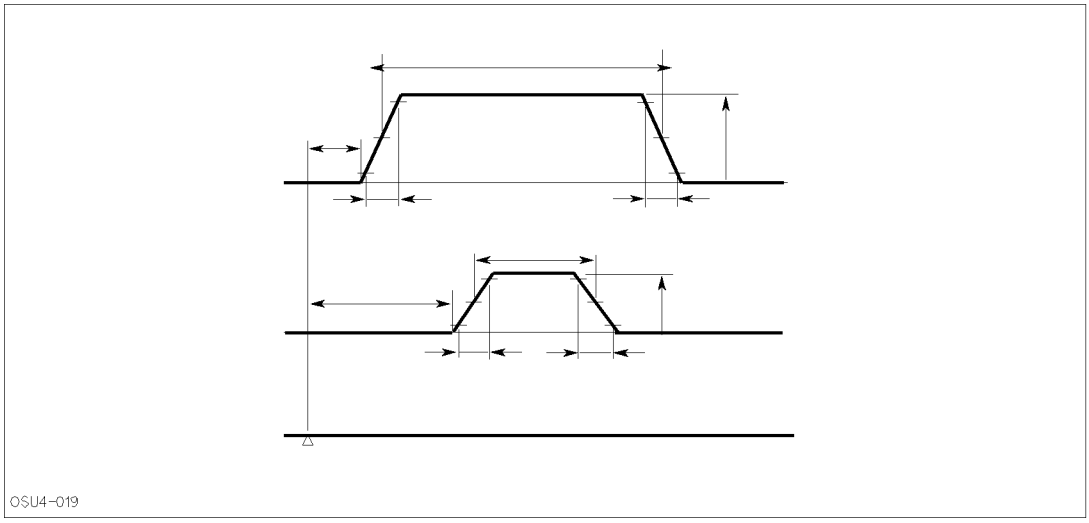
This subprogram executes the write operation of a flash memory cell (low-power NOR type) by using the multiplexer switch for gate input.

Write_nor3 connects measurement pins specified in Set_fr subprogram as in following figure. Attach PGUs as shown in the figure. Make sure that the gate and drain measurement pins are connected to different HF ports. For example, connect gate to one of pins 1 to 4 (which is HF1), and connect drain to one of measurement pins 9 to 12 (which is HF3).



You can use this subprogram with the Erase_nor3 and Repeat_nor3 subprograms in a measurement program. For the connections, refer to the Repeat_nor3 subprogram.

This subprogram forces the following pulses to the gate and drain, which writes to the cell. The source and substrate are connected to the circuit common. While a certain voltage is forced to the gate to make a channel, a voltage is forced to the drain to generate hot carriers near the drain, and hot carriers are injected into the floating gate and are trapped there (write operation). Then, the voltages of the drain and gate are set to 0 V.



Element 1 of each array is for gate, element 2 is for drain, and other elements are ignored. So, the size of each array must be 2 or more.

See Also

Set_fr
Set_multifr
Set_relay_mode
Erase_nor3
Repeat_nor3

Programming Example

The example program executes the write to the low-power NOR memory cell with the following parameters:

	Pulse Voltage	Delay Time	Pulse Width	Rise Time	Fall Time
Gate	10 V	1 μs	10 μs	100 ns	20 ns
Drain	7 V	4 μs	5 μs	200 ns	40 ns
Source, Substrate	0 V	—	—	—	—

```
100   REAL Ampl(1:2),Width(1:2),Delay(1:2),Edge1(1:2),Edge2(1:2)
110   INTEGER Gate,Source,Drain,Subs
120   !
130   Gate=1
140   Source=5
150   Drain=9
160   Subs=13
170   Ampl(1)=10.
180   Ampl(2)=7.
190   Width(1)=1.E-5
200   Width(2)=5.E-6
210   Delay(1)=1.E-6
220   Delay(2)=4.E-6
230   Edge1(1)=1.E-7
240   Edge1(2)=2.E-7
250   Edge2(1)=2.E-8
260   Edge2(2)=4.E-8
270   !
280   Init_system
290   Init_pg
300   Set_fr(Source,Gate,Drain,Subs)
310   Set_relay_mode(1)
320   Write_nor3(Ampl(*),Width(*),Delay(*),Edge1(*),Edge2(*))
330   !
340   END
```

5

Diagnostics Program

You can check that the HP 4062F system is operating properly by executing the diagnostics (DIAG.F) program. This DIAG.F program consists of two parts: dc test and ac test. The dc test checks the dc measurement and capacitance measurement functions, and the ac test checks the pulse output functions. The DIAG.F program also has the VIEW RESULT function to display or print the test results.

Hewlett-Packard recommends that you execute the DIAG.F program once a day before using the system, so you can observe the system operating status. If the system operates correctly, DIAG.F program takes about 15 minutes.

DC Test.
The dc test checks the basic operation of the dc measurement and capacitance measurement functions of the HP 4062F system. This test consists of eight subtests as shown in Table 5-1.

When testing for correct system operation, you execute *all* the tests in the following order. When troubleshooting, you can execute the subtests individually.

Table 5-1. DC Tests of the DIAG.F Program

Subtest	Description
Relay Test	Checks SWM pin board relays for normal *make* (on) or *break* (off) operation and proper conductance levels. Requires the relay test adapter and relay test shorting adapter. Takes about 1 minute to execute.
SMU Test[1]	Performs the HP 4142B DCS's self-test and checks each SMU for correct source/monitor operation. Requires the system test module, and takes about 1 minute to execute.
VS Test[2]	Checks the DCS's voltage sources (VSs) for correct voltage output. Requires the system test module, and takes about 5 seconds to execute.
VM Test[2]	Checks the DCS's voltage monitors (VMs) for correct voltage measurement accuracy. Requires the system test module, and takes about 5 seconds to execute.
AFU Test[2]	Checks the DCS's analog feedback unit (AFU) for correct measurement operation. Requires the system test module, and takes about 15 seconds to execute.
CMS C-G Test[1,3]	Performs the HP 4280A CMU's self-test, except test number 1 and 7, and checks the C-G measurement function for correct operation. Requires the system test module, and takes about 20 seconds to execute.
CMS BIAS Test[2,3]	Checks the CMU's internal dc bias source for correct voltage output. Requires the system test module, and takes about 6 seconds to execute.

1 Requires the Relay Test prior to execution.

2 Requires the Relay and SMU Tests prior to execution.

3 Performed when the HP 4280A CMU exists and is connected to AUX3 (CMH) and AUX4 (CML) ports of the SWM.

Table 5-1. DC Tests of the DIAG.F Program (continued)

Subtest	Description
CMU84 C-G Test[1,2]	Checks the HP 4284A CMU84's C-G measurement function for correct operation. Requires the system test module, and takes about 25 seconds to execute.
CMU84 BIAS Test[3,2]	Checks the internal dc bias source accuracy of the CMU84. Requires the system test module, and takes about 5 seconds to execute.
DC Leak Test[3]	Checks SWM leakage current. Requires the test fixture adapter, and takes about 30 seconds.

1 Requires the Relay Test prior to execution.

2 Performed when the HP 4284A CMU84 exists and is connected to AUX3 (CMH) and AUX4 (CML) ports of the SWM.

3 Requires the Relay and SMU Tests prior to execution.

This dc testing is as same as the DIAG test of the HP 4062C/UX system. For details, see chapter 4 of the HP 4062C/UX *System Information* manual.

AC Test.
The ac test checks the basic operation of the pulse function of the HP 4062F system. This test consists of three subtests as shown in Table 5-2.

When testing for correct system operation, you execute all the tests in the following order. When troubleshooting, you can execute the subtests individually.

Table 5-2. AC Test Contents of the DIAG.F Program

Subtest	Description
8110A Self-test	Performs the HP 8110A PG's self-test, and takes about 5 seconds per PG.
Connection Test	Checks the connections between the pulse generators and SWM. Requires the pulse test module, and takes 1 minute per HF port.
Switch Test	Checks the ON/OFF operation of the cable adapter switches that you use. Requires the pulse test module, and takes about 15 seconds.

VIEW RESULT Function.
VIEW RESULT can display or print the test results that were saved to a file by the dc or ac test. For details on the VIEW RESULT function, see chapter 4 of the HP 4062C/UX *System Information* manual.

Diagnostics Program Execution

This section covers how to execute the HP 4062F diagnostics (DIAG.F) program.

Preparation for DIAG.F Program

The following accessories are required for executing the DIAG.F program:

HP 16066A Test Fixture Adapter[1]
HP 16075F Relay Test Adapter[1]
HP 16076A System Test Module[1]
HP 16354F Pulse Test Module[2]
Relay Test Shorting Adapter (HP part number 04085-60006)[2]

1 These accessories are also used for the HP 4062C/UX system.

2 HP 16354F and 04085-60006 are used only for DIAG.F program of HP 4062F system. Do not use them for other purposes.

You install these accessories on the bottom panel of the SWM. For how to install the accessories, see chapter 4 of the HP 4062C/UX *System Information* manual.

To properly perform the DIAG.F program, pin boards must be installed in the following slots at least:

4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48

If you execute the Relay Test only, at least five pin boards must be installed in any slots of the SWM.

Before executing the DIAG program, make sure about the following items:

- The ID Module must be installed in the system controller.
- The system instrument interconnection cables must be connected correctly.
- The HP 4280A CMU's front panel INT BIAS switch must be set to the ±100 V MAX position.
- The CMU's front panel REMOTE ON/OFF terminal must be terminated with a BNC shorting cap.
- The CMU's rear panel EXT BIAS SLOW terminal must be terminated with a BNC shorting cap.
- HP 4142B DCS's rear panel FILTER switch must be set to the correct line frequency.
- The DCS, CMU, CMU84, or PGs forces no voltage and current.
- Nothing is installed on the bottom panel of the SWM, or only the HP 16066A Test Fixture Adapter is installed.

Then, you turn on the system power. See chapter 3 of the HP 4062C/UX *System Information* manual about how to turn on the system. After turning on your system, let your system warm up with nothing installed on the SWM. Recommended warm-up time is at least 40 minutes.

Note

When you turn your system on, the DCS may exhibit one of the following failure symptoms:

- A character other than "0" is displayed on the DCS's front panel ERROR/FAILURE display after calibration, which displays "0" on the display.
- Except for normal fan operation, the DCS is dead (all LEDs are off).

If either symptom occurs, ignore it for the present and let the system fully warm up. After your HP 4062F is warmed up, turn the DCS off, then on again. In most cases, this clears the failure symptoms.

Note

When you turn your system on, all the front panel LEDs may turn on and may not turn off. If this occurs, turn the SWC off for at least 5 seconds, then turn it back on.

Warning

When you start the DIAG.F program, be sure that the HP 16354F is not installed on the HP 4085F SWM. Then, mount the HP 16354F on the SWM according to messages displayed by the system software.

If you mount the HP 16354F on the SWM when a voltage or current source is forcing voltage or current, dangerous voltages may be present on the connectors of the HP 16354F, and may shock you.

Starting the DIAG.F Program

To start the DIAG.F program, do as follows:

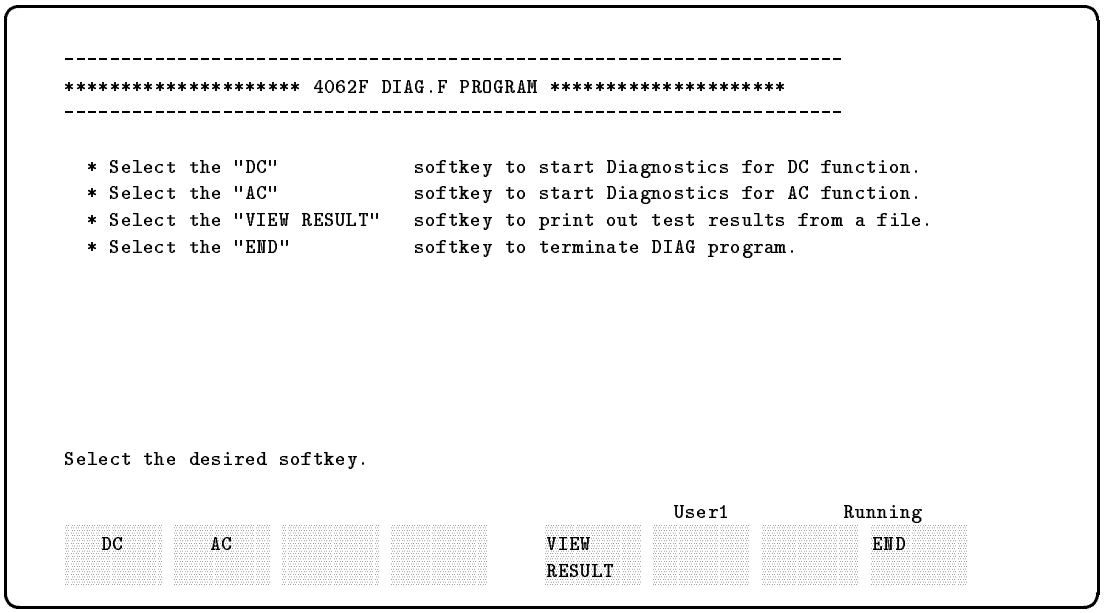
1. Start HP BASIC/UX.
2. Change current mass storage device to /usr/pcs/diag directory.

   `MSI "/usr/pcs/diag"` Return

3. Load the "DIAG.F" program file.

   `LOAD "DIAG.F"` Return

4. Select `RUN` softkey to start the DIAG.F program. The following page is displayed:



5. Select `DC` softkey. This starts the diagnostics execution page for the dc measurement and capacitance measurement of the HP 4062F system. This part is same as the DIAG program of the HP 4062C/UX system, so see chapter 4, "Diagnostics", of the HP 4062C/UX *System Information* manual about the operation procedures.

   Note that the HP 16075F relay test adapter and relay test shorting adapter are required to execute the Relay Test of the dc test. No message is displayed to prompt you to install the relay test shorting adapter, so you must remember to do it yourself.

   Use the following procedure to install:

   a. Remove the cable adapter from the SWM.

   b. Install the relay test shorting adapter.

   c. Remove the HP 16066A test fixture adapter.

   d. Install the HP 16075F relay test adapter.

   After the Relay Test finishes, re-install the cable adapter.

After the diagnostics of the dc measurement and capacitance measurement functions are finished, the previous page is displayed again.

6. Select `AC` softkey. The DIAG.F execution page for the pulse function is displayed.
7. Install the HP 16354F pulse test module on the SWM according to the displayed messages.
8. Select `START` softkey after installing it.

   The SWITCH CONFIGURATION page is displayed, which shows the statuses of the switch connections.
9. If the switch configuration settings *exactly* match the actual configuration, select the `OK` softkey. The ac test menu page is displayed. See "Selecting AC Subtests" about each test item.

   If the switch configuration settings do *not* exactly match the actual connections, select `RECONFIG` softkey to correct them. See "Reconfiguring SWITCH Configuration".

You can save the dc (dc and capacitance measurement) and ac (pulse function) diagnostics results to files. Then, you can display the diagnostics results on the CRT or print them by using the VIEW RESULT function.

See chapter 4, "Diagnostics", of the HP 4062C/UX *System Information* manual about these functions.

Reconfiguring SWITCH Configuration

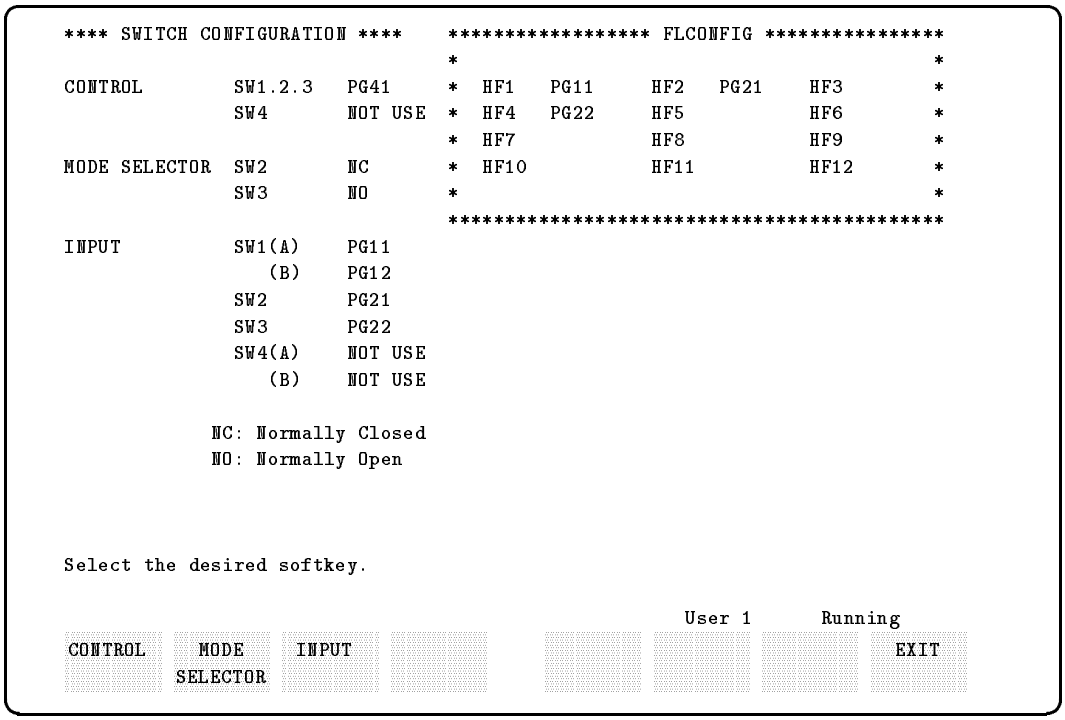


Figure 5-1. SWITCH Configuration Page

`CONTROL` Shows which PGUs are controlling the switches. These PGUs must not be assigned to an HF port in the FLCONFIG file. "`NOT USE`" means that no PGU is connected to the connector of CTRL SW 1·2·3 or SW4.

`MODE SELECTOR` Shows the settings of the MODE SELECTOR. "NC" means normally closed, and "NO" means normally open. SW2 and SW3 are set to the specified status (open or closed) when switch control PGU is specified by Set_sr_pg subprogram.

`INPUT` Shows which PGUs are connected to the input terminals the switches. The PGUs that are connected to input A of SW1 or SW4, or connected to SW2 or SW3 must be assigned to HF ports in the FLCONFIG file. For the PGUs that are connected to input B of SW1 or SW4, do *not* assign the PGUs to HF ports in the FLCONFIG file. "`NOT USE`" means that no PGU is connected to the input terminal.

Use the following procedure to modify the settings:

- Modifying the CONTROL settings:

  1. Select `CONTROL` softkey.
  2. Select `SW 1,2,3` or `SW 4` softkey to specify the target terminal.
  3. Select softkey of desired PGU. Select `NOT USE` softkey for no connection.

- Modifying the MODE SELECTOR settings:
  1. Select `MODE SELECTOR` softkey.
  2. Select softkey of desired switch. This toggles between NC and NO.
- Modifying the INPUT settings:
  1. Select `INPUT` softkey.
  2. Select the softkey of desired input terminal.
  3. Select the softkey of desired PGU. Select `NOT USE` softkey for no connection.

After you complete the modifications, select `EXIT` softkey.

Selecting AC Subtests

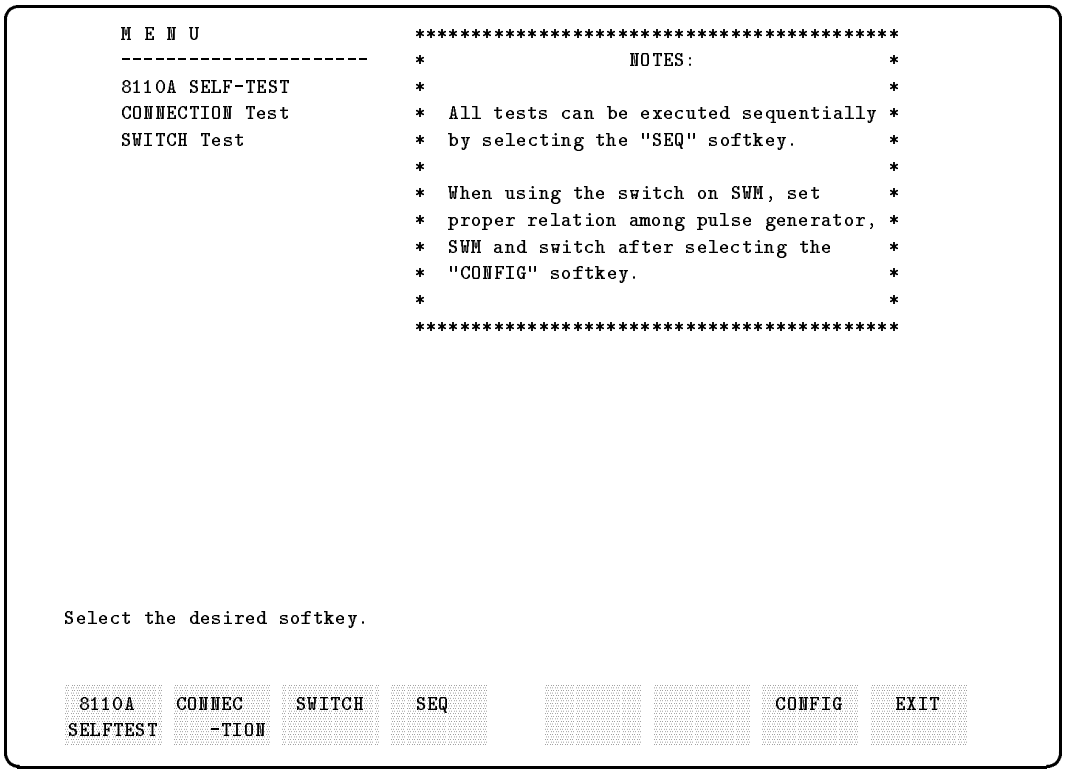


Figure 5-2. Subtest Selection Page

8110A Self-test — Executes the self-test of the HP 8110A pulse generators installed in the HP 4062F system, and shows the test results.

Connection test — Checks the connections between the HF ports and PGUs, and shows the test results. This test is available for the connections described in the FLCONFIG file.

Switch test — Executes the ON/OFF test of the cable adapter switches, and shows the test results.

To execute the subtests individually, select the softkey of desired subtest.

To automatically execute all tests sequentially, select the `SEQ` softkey. You can also save the results of the ac diagnostics tests in a file. For details, see “Saving Diagnostics Results”.

To modify the switch configuration settings, select `CONFIG` softkey. The SWITCH CONFIGURATION page appears. For details, see “Reconfiguring SWITCH Configuration”.

Saving Diagnostics Results

The DIAG.F program has a function for saving the test results to a file. Test results saved to a file are useful for troubleshooting.

Use the following procedure to save the test results to a file:

1. Select `SEQ` softkey on the subtest selection page.

   The prompt "`Do you want to keep result in a file?`" is displayed.

2. Select `Yes` softkey, then go to the next step.

   If you do not want to save results, select `No` softkey. This starts the "Relay Test" for dc test or "HP 8110A Self-test" for ac test.

3. Enter a file name. For example, enter the following filename for the ac test:

   `/usr/pcs/diag/data/DG.AC.940318` Return

Note

Do not use the same filename for the test results of dc test and ac test. If you use the same filename for dc test results, then for ac test results, the dc test results will be lost.

4. Do the following:

   - If you specified an existing file name in the previous step, the "Relay Test" or "HP 8110A Self-test" starts.
   - If you specified a new file name, a test results file data entry page is displayed. You enter the information about test environments, serial numbers of instruments, and so on. After entering the data, the message "`Are you sure?`" is displayed.

     If the data is correct, select `Yes` softkey. The "Relay Test" or "HP 8110A Self-test" starts.

     If not correct, select `No` softkey, then re-enter data as required.

When An Error Occurs in the DIAG Program

When an error occurs during the DIAG.F program execution, its error message is displayed. When an error message is displayed, or when you are not able to execute the DIAG.F program, check for the following possible causes, then execute the DIAG.F program again:

- □ Did you install the required test adapter or module securely on the SWM?
- □ Are the contacts of the test adapter or module clean?
- □ Did you do incorrect operations during the DIAG.F program execution?
- □ Are all system instruments turned on?
- □ Are any devices other than system instruments connected to the select code 27 bus?
- □ Are all system instruments interconnected properly?
- □ Is your HP 4062F system fully warmed up? Minimum warm-up time is 40 minutes.

If you check the above items and the same error occurs after you execute DIAG.F program again, contact the nearest HP service office and describe the following:

- Serial number of the HP 4062F system for which error occurs.
- Test name for which error occurs.
- Displayed error codes and messages.

The following briefly describes how to deal with errors during the ac test. For how to deal with the dc test errors, see "Error Information" in chapter 4 of the HP 4062C/UX *System Information* manual.

When the HP 8110A Self-test fails.
If an HP 8110A fails its self-test, remove the failed HP 8110A from your system, then execute the self-test of the failed HP 8110A by itself.

Use the following procedure to execute the HP 8110A self-test by itself:

1. Turn the HP 8110A on.
2. Select the `CONFIG` softkey on the front panel.
3. Press (▼) key to move the pointer to "`Perform Selftest`".
4. Press (ENTER) key to start the self-test.

If the self-test fails, "`E`" blinks on the screen of the HP 8110A. To display its error message, press (HELP) key.

When the Connection Test fails.
Check the following:

- □ Make sure the PGUs are connected to HF ports specified in the FLCONFIG file.
- □ Make sure cables from PGUs to HF ports are not broken.

When the Switch Test fails.
Check the following, when the Switch test fails:

- □ Make sure interconnection cables are installed as described on the switch configuration page.
- □ Make sure cables between the switches and HF ports are not broken.

6

Introduction of Applecation Programs

This chapter outlines the application programs that are included with the HP 4062F, the cell type of flash EEPROM assumed by these programs, and other cell types that you can test by modifying these programs.

Application Programs

The following three application programs, which use the most typical evaluation methods, are included with the HP 4062F system software.

- Accumulated write/erase pulse width dependency of cell threshold voltage.
- Pulse width or pulse level dependency of cell threshold voltage.
- Write/erase times dependency of cell threshold voltage.

There are several types of flash EEPROM cell. The way to apply pulses to the cell for writing or erasing varies depending on the cell type.

Application programs included with the HP 4062F are written as simple as possible, which allows easy modification to fit any write or erase conditions.

Chapter 2, 3, and 4 describe how to modify these application programs for various cell types. So, you can make your own measurement program based on these application programs.

Cell Type Assumed by Application Programs

These application programs assume that you are testing conventional NOR type flash EEPROM cells. This section describes the writing and erasing conditions for the NOR type cell.

The application programs are for testing conventional NOR type cells that have four terminals as shown in Figure 6-1.

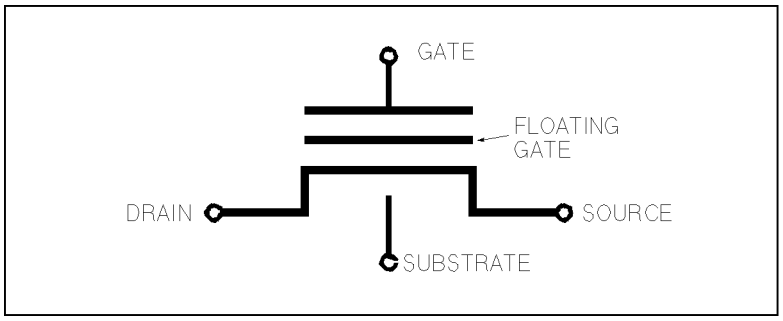


Figure 6-1. Conventional NOR Type Flash EEPROM Cell

Writing Method

While a pulse is forced to the gate to make a channel, a pulse is forced to the drain to generate hot electrons, which are injected into the floating gate and trapped there.

The pulse levels for writing are as follows:

Terminal name	Pulse level to be applied
Gate	14 V
Drain	7 V
Source	0 V
Substrate	0 V

Erasing Method

A pulse is forced to the source while the drain is open. The electrons tunnel from the floating gate to the source by Fowler-Nordheim tunneling.

The pulse levels for erasing are as follows:

Terminal name	Pulse level to be applied
Gate	0 V
Drain	Open
Source	11 V
Substrate	0 V

Other Cell Types

The application programs can be modified for testing types other than conventional NOR type EEPROM cells. Chapter 2, 3, and 4 describe the application programs and how to modify the application programs for testing other cell types.

This section introduces the NAND type and 3.3V NOR type and lists their writing and erasing conditions.

NAND Type Cell

The structure of the NAND cell is similar to conventional NOR type cell with four terminals as shown in Figure 6-2.

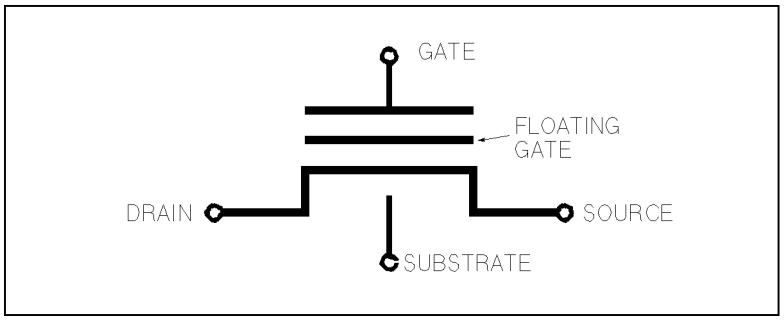


Figure 6-2. NAND Type Flash EEPROM Cell

Writing Method

When a pulse is forced to the gate, electrons are injected into the floating gate by Fowler-Nordheim tunneling.

The pulse levels for writing are as follows:

Terminal name	Pulse level to be applied
Gate	20 V
Drain	0 V
Source	0 V
Substrate	0 V

Erasing Method

When a pulse is applied to the source, drain, and substrate, the electrons tunnel out of the floating gate by Fowler-Nordheim tunneling.

The pulse levels for erasing are as follows:

Terminal name	Pulse level to be applied
Gate	0 V
Drain	20 V
Source	20 V
Substrate	20 V

3.3V NOR Type Cell

The structure of the 3.3V NOR cell is similar to conventional NOR type cell with four terminals as shown in Figure 6-3.

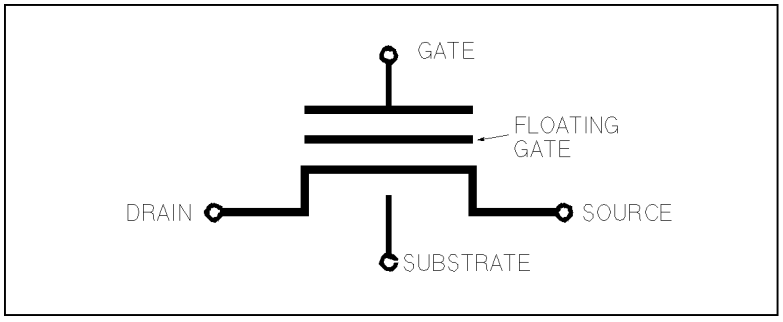


Figure 6-3. 3.3V NOR Type Flash EEPROM Cell

Writing Method

While a pulse is forced to the gate to make a channel, a pulse is forced to the drain to generate hot electrons, which are injected into the floating gate and trapped there. This is the same as the writing method for conventional NOR type cells.

The pulse levels for writing are as follows:

Terminal name	**Pulse level to be applied**
Gate	14 V
Drain	7 V
Source	0 V
Substrate	0 V

Erasing Method

While a pulse is forced to the source, and the drain is open, a pulse with opposite polarity is forced to the gate. The electrons tunnel from the floating gate to the source by Fowler-Nordheim tunneling.

The pulse levels for erasing are as follows:

Terminal name	**Pulse level to be applied**
Gate	−6 V
Drain	Open
Source	6 V
Substrate	0 V

7

Accumulated Pulse Width Dependency Measurement

This chapter describes the ACC_PWD_VTH application program, which is included with the HP 4062F. The meaning of ACC_PWD_VTH is "accumulated write/erase pulse width dependency of the cell threshold voltage".

Outline of Accumulated Pulse Width Dependency Measurement

This measurement is used to determine the best pulse width to apply to the cell. Application program "ACC_PWD_VTH" makes measurements as shown in the following flowchart:

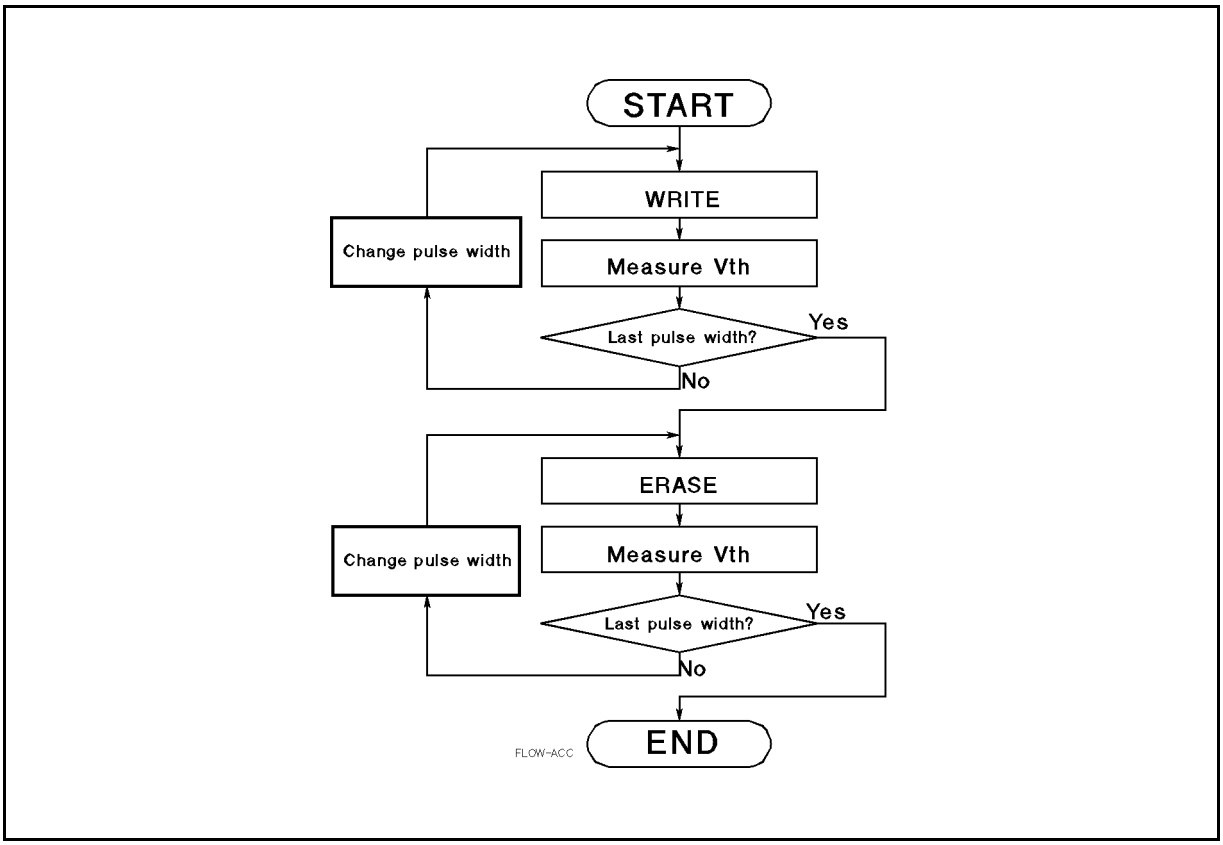


Figure 7-1. Flowchart of ACC_PWD_VTH Program

First, the accumulated *write* pulse width dependency of Vth is measured. After the cell is fully written, the accumulated *erase* pulse width dependency of Vth is measured:

1. Write pulses are applied to the cell with increasing pulse widths. After each write pulse, the cell threshold voltage is measured. The threshold voltage is plotted against the accumulated write pulse width.
2. Erase pulses are applied to the cell with increasing pulse widths. After each erase pulse, the cell threshold voltage is measured. The threshold voltage is plotted against the accumulated erase pulse width.

Figure 7-2 shows example characteristics of accumulated pulse width dependency.

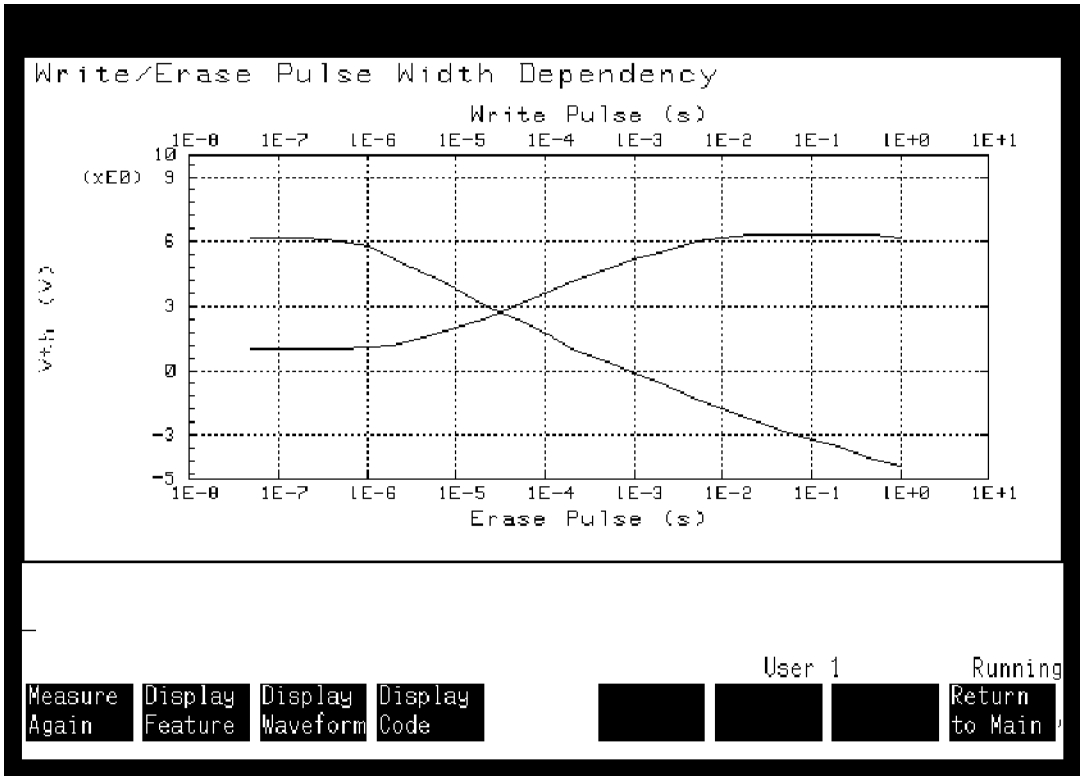


Figure 7-2. Example of Accumulated Pulse Width Dependency Measurement

Necessary Configuration for ACC_PWD_VTH

This section describes the necessary hardware configuration and setup for using ACC_PWD_VTH.

Hardware

- Measurement controller: HP 9000 Series 300
- HP 4062F (at least three PGUs are necessary)

Setup

Connect between PGUs and HF ports as listed in the following table:

Port or PGU name	Port or terminal to be connected
PG11	HF port that can be connected to the drain
PG12	HF port that can be connected to the source
PG21	HF port that can be connected to the gate

Drain, source, and gate terminals must be connected to *different* HF ports. The program assumes the following pin locations for the cell. If the pin locations are different from the following, you can customize the program or can make additional connections with some wires so that the equivalent connections are made. See "Customization of ACC_PWD_VTH" if you need to customize your program.

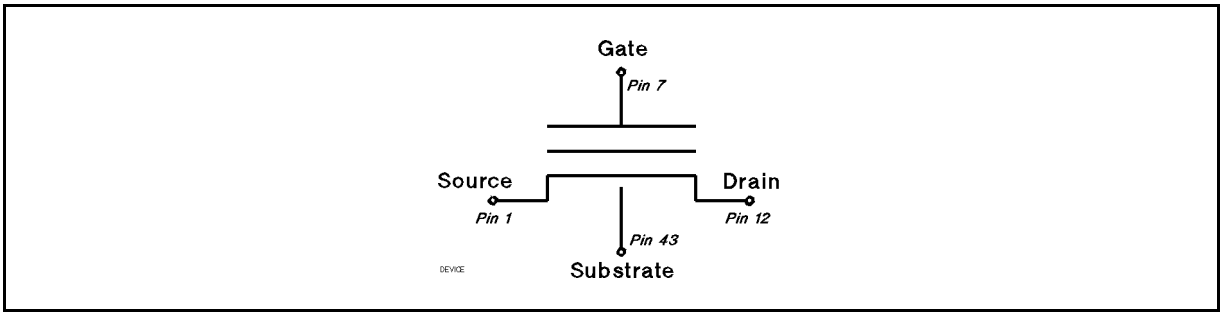


Figure 7-3. Pin Locations of Assumed Cell

Modify the FLCONFIG file according to the connections between the HF ports and PGUs. For example, the following is for the above pin locations.

```
1000 !!  RE-SAVE "/usr/pcs/etc/FLCONFIG"
1010 !!
1020 ! PG1  HP8110A  2725
1030 ! PG2  HP8110A  2726
1040 ! PG3  HP8110A  2727
1050 ! PG4  HP8110A  2728
1060 ! PG5  HP8110A  2729
1070 !
1080 ! HF1  PG12
1090 ! HF2  PG21
1100 ! HF3  PG11
```

Execution of ACC_PWD_VTH

1. Start the HP BASIC/UX environment.

   ```
   $ rmb  Return
   ```

2. Run START.F program of the HP 4062F.

   ```
   LOAD "/usr/pcs/bin/START.F",1  Return
   ```

3. Load the ACC_PWD_VTH program file.

   ```
   LOAD "/usr/pcs/demo/flash/ACC_PWD_VTH"  Return
   ```

4. Make necessary customization. See the following pages.

5. Link TIS.F file by selecting `Link TIS` softkey.

6. Link PARA.F library file.

   ```
   LOADSUB ALL FROM "/usr/pcs/lib/PARA.F"  Return
   ```

7. Link SEMIGRAPH graphic library.

   ```
   LOADSUB ALL FROM "/usr/pcs/demo/flash/SEMIGRAPH"  Return
   ```

8. Execute the program by selecting `RUN` softkey.

Customization of ACC_PWD_VTH

This section describes how to modify ACC_PWD_VTH for cell types that are almost the same as the conventional NOR type described in Chapter 1. After loading ACC_PWD_VTH, go into editing mode by selecting EDIT softkey, then make modifications.

In the main program of ACC_PWD_VTH, pins are assigned to the DUT terminals, then parameters are specified for the threshold voltage (Vth) measurements. Then, system initialization is performed, and subroutine for accumulated pulse width dependency (Acc_pwd_dep) is called.

In the Acc_pwd_dep subroutine, the graphics axes are drawn, and the pulse levels and leading and trailing edges are set. Then, a series of write and Vth measurements with increasing pulse widths is performed. After that, a series of erase and Vth measurements with increasing pulse widths is performed.

Pin Assignments of the DUT

The pin assignments of the DUT are made in the following lines:

```
1040  !-- TEG pin assignment ---------------------------------
1045  Drain=12         ! Drain
1050  Gate=7           ! Gate
1055  Source=1         ! Source
1060  Sub=43           ! Sub
1065  !
```

Modify the pin numbers according to the actual connections.

Threshold Voltage Measurement

The write pulse (or erase pulse) parameters are set, then Write_meas_plot (or Erase_meas_plot) subroutine is called. In the subroutine, write (or erase) pulse is applied to the cell, Vth measurement is made, and the data is plotted on the graph. This loop is repeated with increasing pulse widths.

Two Vth measurement subroutines are available: Meas_vth (which uses AFU and extracts Vth by analog feedback measurement) and Meas_vth2 (which extracts Vth by linear search measurement). In the ACC_PWD_VTH program, Meas_vth2 is used, but you can use Meas_vth by changing lines 1630 and 1675.

```
1625    Write_nor1(Level_w(*),Width_w(*),Delay_w(*),Ledge_w(*),Tedge_w(*))
1630    Meas_vth2(Vth_w(Num_of_data_w))
           :
1670    Erase_nor1(Level_e,Width_e,Ledge_e,Tedge_e)
1675    Meas_vth2(Vth_e(Num_of_data_e))
```

The Vth measurement parameters are set in the following lines. You can modify these values according to the DUT characteristics.

```
1070  !-- Parameters for Vth measurement ----------------------
1075  Vgstart=0              ! Start voltage
1080  Vgstop=8               ! Stop voltage
1085  Vd=.1                  ! Vd
1090  Id_targ=1.E-6          ! Target current
1095  Rmp=50                 ! Ramp rate                  (Meas_vth)
1100  Integ=5.E-3            ! Feedback integration time  (Meas_vth)
1105  Vgstep=.01             ! Vgstep                     (Meas_vth2)
1110  Igcomp=.1              ! Ig compliance              (Meas_vth2)
```

```
1115  Idcomp=1.1E-6          ! Id compliance              (Meas_vth2)
1120  Hold=0                 ! Hold                       (Meas_vth2)
1125  !
```

Parameter Name	Description	Notice
Vgstart	Start gate voltage	Necessary both for Meas_vth and Meas_vth2
Vgstop	Stop gate voltage	Necessary both for Meas_vth and Meas_vth2
Vd	Drain voltage	Necessary both for Meas_vth and Meas_vth2
Id_targ	Target drain current	Necessary both for Meas_vth and Meas_vth2
Rmp	Ramp rate of AFU	Necessary only for Meas_vth
Integ	Feedback integration time	Necessary only for Meas_vth
Vgstep	Gate step voltage	Necessary only for Meas_vth2
Igcomp	Gate current compliance	Necessary only for Meas_vth2
Idcomp	Drain current compliance	Necessary only for Meas_vth2
Hold	Hold time	Necessary only for Meas_vth2

Above parameters are passed to `Meas_vth` or `Meas_vth2` subroutine as COM variables. If `Meas_vth` or `Meas_vth2` is modified and some parameters are added to subroutine, add these parameters to `COM /M_vth/` declaration in lines 1030, 1745, and 1810.

These subroutines connect the gate, drain, source, and substrate pins to *separate* SMUs, and execute the Vth measurements. Modify the source program as required.

Pulse Level, Leading Edge, and Trailing Edge for Write/Erase

The pulse level, leading edge, and trailing edge parameters for write/erase are made in the following lines. Modify these parameters according to the DUT characteristics.

```
1345    Level_w(1)=14      !(V)      ! Gate pulse level (Write)
1350    Level_w(2)=7       !(V)      ! Drain pulse level (Write)
1355    Level_e=11         !(V)      ! Source pulse level (Erase)
1360    Ledge_w(1)=5.E-8   !(s)      ! Gate leading edge (Write)
1365    Ledge_w(2)=5.E-8   !(s)      ! Drain leading edge (Write)
1370    Ledge_e=5.E-8      !(s)      ! Source leading edge (Erase)
1375    Tedge_w(1)=5.E-8   !(s)      ! Gate trailing edge (Write)
1380    Tedge_w(2)=5.E-8   !(s)      ! Drain trailing edge (Write)
1385    Tedge_e=5.E-8      !(s)      ! Source trailing edge (Erase)
1390    !
```

Parameter Name	Description
Level_w(1)	Gate pulse level for write
Level_w(2)	Drain pulse level for write
Level_e	Source pulse level for erase
Ledge_w(1)	Leading edge of gate pulse for write
Ledge_w(2)	Leading edge of drain pulse for write
Ledge_e	Leading edge of source pulse for erase
Tedge_w(1)	Trailing edge of gate pulse for write
Tedge_w(2)	Trailing edge of drain pulse for write
Tedge_e	Trailing edge of source pulse for erase

Pulse Width and Delay Time for Write/Erase

Pulse width settings for *write* pulse are set in the following lines. Modify as required.

```
1395    !--- Accumulated write pulse width vs. Vth --------------------
1400    FOR Loop1=-7 TO -1
1405      Width_w(2)=10^Loop1-Total_width_w   ! Drain pulse width
1410      Width_w(1)=Width_w(2)*1.1           ! Gate pulse width
1415      Delay_w(2)=Width_w(2)*.05           ! Drain pulse delay time
1420      Delay_w(1)=0                        ! Gate pulse delay time
1425      Total_width_w=10^Loop1              ! Accumulated pulse width
1430      GOSUB Write_meas_plot               ! Apply write pulse and meas Vth
1435      !
1440      Width_w(2)=2*10^Loop1-Total_width_w ! Drain pulse width
1445      Width_w(1)=Width_w(2)*1.1           ! Gate pulse width
1450      Delay_w(2)=Width_w(2)*.05           ! Drain pulse delay time
1455      Delay_w(1)=0                        ! Gate pulse delay time
1460      Total_width_w=2*10^Loop1            ! Accumulated pulse width
1465      GOSUB Write_meas_plot               ! Apply write pulse and meas Vth
1470      !
1475      Width_w(2)=5*10^Loop1-Total_width_w ! Drain pulse width
1480      Width_w(1)=Width_w(2)*1.1           ! Gate pulse width
1485      Delay_w(2)=Width_w(2)*.05           ! Drain pulse delay time
1490      Delay_w(1)=0                        ! Gate pulse delay time
1495      Total_width_w=5*10^Loop1            ! Accumulated pulse width
1500      GOSUB Write_meas_plot               ! Apply write pulse and meas Vth
1505    NEXT Loop1
```

Above sets pulse width so that total accumulated pulse width (`Total_width_w`) becomes 100ns (1E−7s), 200ns (2E−7s), 500ns (5E−7s), 1μs (1E-6s), ... , 500ms (5E−1s).

Total accumulated pulse width (`Total_width_w`) is determined by the equation $x \times 10^{\mathrm{Loop1}}$ (x and `Loop1` are integers). `Loop1` is loop counter for above FOR—NEXT loop and is incremented from −7 to −1. Within each loop, x changes as follows: 1, 2, 5.

The width of *write* pulse applied to the drain is `Width_w(2)`, which is determined by subtracting `Total_width_w` (total accumulated pulse width) from $x \times 10^{\mathrm{Loop1}}$, which will be the new total accumulated pulse width after applying pulse to the drain.

The width of pulse applied to gate (`Width_w(1)`)is set to *1.1* times drain pulse (`Width_w(2)`). Delay time is set so that pulses are synchronized.

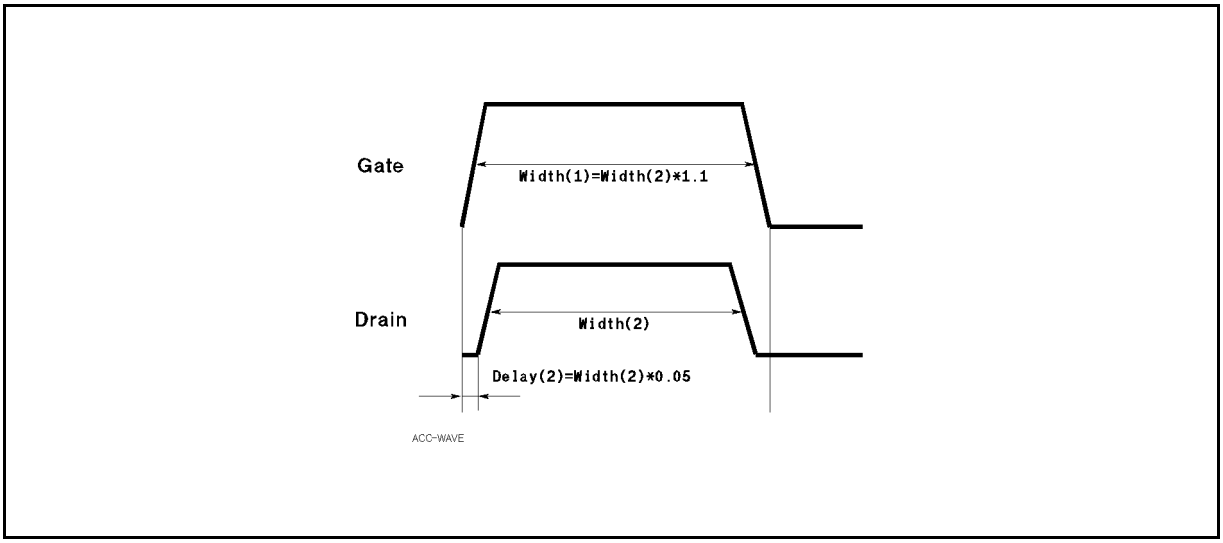


Figure 7-4. Synchronization of Drain and Gate Pulses

Parameter Name	Description
`Width_w(1)`	Gate pulse width for write
`Width_w(2)`	Drain pulse width for write
`Delay_w(1)`	Gate pulse delay time for write
`Delay_w(2)`	Drain pulse delay time for write

Pulse width settings for *erase* pulse are set in the following lines. Modify as required.

```
1515    !--- Accumulated erase pulse width vs. Vth --------------------
1520    FOR Loop2=-7 TO -1
1525      Width_e=10^Loop2-Total_width_e    ! Source pulse width
1530      Delay_e=0                         ! Source pulse delay time
1535      Total_width_e=10^Loop2            ! Accumulated pulse width
1540      GOSUB Erase_meas_plot             ! Apply erase pulse and meas Vth
1545      !
1550      Width_e=2*10^Loop2-Total_width_e  ! Source pulse width
1555      Delay_e=0                         ! Source pulse delay time
1560      Total_width_e=2*10^Loop2          ! Accumulated pulse width
1565      GOSUB Erase_meas_plot             ! Apply erase pulse and meas Vth
1570      !
1575      Width_e=5*10^Loop2-Total_width_e  ! Source pulse width
1580      Delay_e=0                         ! Source pulse delay time
1585      Total_width_e=5*10^Loop2          ! Accumulated pulse width
1590      GOSUB Erase_meas_plot             ! Apply erase pulse and meas Vth
1595    NEXT Loop2
```

The above sets the pulse width so that total accumulated pulse width (`Total_width_e`) becomes 100 ns (1E−7 s), 200 ns (2E−7 s), 500 ns (5E−7 s), 1 μs (1E−6 s), ... 500 ms (5E−1 s), which is same as for the write pulse setting.

Parameter Name	Description
`Width_e`	Source pulse width for erase
`Delay_e`	Source pulse delay time for erase

Write, Vth Measurements, and Plot

Write, Vth measurements, and plot are performed by the `Write_meas_plot` subroutine.

```
1615 Write_meas_plot:!
1620    Num_of_data_w=Num_of_data_w+1
1625    Write_nor1(Level_w(*),Width_w(*),Delay_w(*),Ledge_w(*),Tedge_w(*))
1630    Meas_vth2(Vth_w(Num_of_data_w))
1635    IF Num_of_data_w>1 THEN
1640      Plot_x1y(Total_width_w,Total_width_w-Width_w(2),Num_of_data_w,Clr_wrt,Vth_w(*))
1645    END IF
1650    RETURN
1655    !
```

Write is made by `Write_nor1`, which is a subprogram of `PARA.F` library. Plot is made by `Plot_x1y`, which is a subprogram of `SEMIGRAPH` library. See Appendix A.

Erase, Vth Measurements, and Plot

Erase, Vth measurements, and plot are performed by the `Erase_meas_plot` subroutine.

```
1660 Erase_meas_plot:!
1665    Num_of_data_e=Num_of_data_e+1
1670    Erase_nor1(Level_e,Width_e,Ledge_e,Tedge_e)
1675    Meas_vth2(Vth_e(Num_of_data_e))
1680    IF Num_of_data_e>1 THEN
1685      Plot_x1y(Total_width_e,Total_width_e-Width_e,Num_of_data_e,Clr_ers,Vth_e(*))
1690    END IF
1695    RETURN
1700    !
```

Erase is made by `Erase_nor1`, which is a subprogram of `PARA.F` library. Plot is made by `Plot_x1y`, which is a subprogram of `SEMIGRAPH` library. See Appendix A.

Graphics Axes

The graphics axes are set by the following lines:

```
1265    X1ax_min=1.00E-7                              !
1270    X1ax_max=1.E+0                                !
1275    X2ax_min=1.00E-7                              ! Parameters
1280                                                  !  for Graph axis
1285    Vth_ax_min=0                                  !
1290    Vth_ax_max=8                                  !
1295    Yax_n$="Vth (V)"                              !
1300    G_name$="Accumulated Pulse Width Dep."        !
1305    X1ax_n$="Erase pulse width (s)"               !
1310    X2ax_n$="Write pulse width (s)"               !
1315    Clr_wrt=2    ! Pen Color for Write            !
1320    Clr_ers=4    ! Pen Color for Erase            !
1325    !
1330    !
1335    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0,"","",
X2ax_n$,X2ax_min)
```

You can modify these parameters as required. See Appendix A.

Customization of ACC_PWD_VTH for NAND Type Cells

This section describes how to modify ACC_PWD_VTH for cell types that are similar to the NAND type described in Chapter 1. For NAND type, only two PGUs are necessary. So, you need to modify the connections and the program.

Port or Terminal Name	Port or terminal to be connected
PG11	HF port that can be connected to the gate
PG12	HF port that can be connected to drain, source, and substrate

The drain, source, and substrate terminals must be connected to PG12, so the DUT must be designed so that these terminals can be connected to the same HF port. If this is not possible, you can use BNC T connectors to connect PG12 to multiple HF ports.

After loading ACC_PWD_VTH, go into editing mode by selecting EDIT softkey, then make modifications as described in this section.

If DUT is same type as NAND cell described in Chapter 1, you use the Write_nand1 and Erase_nand1 subprograms for writing and erasing.

Parameters required by Write_nand1 are pulse level, width, leading edge, and trailing edge. But for a conventional NOR cell, Write_nor1 is used, which requires delay time in addition to the other parameters. Delay time is not necessary for NAND type cell, so number of parameters is less than for NOR type cell.

Also, Write_nand1 requires parameters for one pulse waveform, whereas Write_nor1 requires parameters for two pulse waveforms.

Pin Assignments of the DUT

The pin assignments of the DUT are made in the following lines:

```
1040  !-- TEG pin assignment ----------------------------------
1045  Drain=12          ! Drain
1050  Gate=7            ! Gate
1055  Source=1          ! Source
1060  Sub=43            ! Sub
1065  !
```

Modify the pin numbers according to the actual connections. Note that the HF port of the gate pin must *not* be the same as the HF port for the drain, source, and substrate pins. You can connect the drain, source, and substrate pins to the same HF port or to different HF ports, but make sure these pins can connect to PG12. And gate must be able to connect to PG11.

If the DUT cannot be connected properly, redesign bond wiring for package devices, or add some bypass wires on your probe card for wafer devices.

Threshold Voltage Measurement

The write pulse (or erase pulse) parameters are set, then `Write_meas_plot` (or `Erase_meas_plot`) subroutine is called. In the subroutine, write (or erase) pulse is applied to the cell, Vth measurement is made, and the data is plotted to the graph. This loop is repeated with increasing pulse widths.

Two Vth measurement subroutines are available: `Meas_vth` (which uses AFU and extracts Vth by analog feedback measurement) and `Meas_vth2` (which extracts Vth by linear search measurement). `Meas_vth2` is used in the program, but you can use `Meas_vth` by changing lines 1630 and 1675.

```
1625    Write_nor1(Level_w(*),Width_w(*),Ledge_w(*),Tedge_w(*),Delay_w(*))
1630    Meas_vth2(Vth_w(Num_of_data_w))
           :
1670    Erase_nor1(Level_e,Width_e,Ledge_e,Tedge_e,Delay_e)
1675    Meas_vth2(Vth_e(Num_of_data_e))
```

The Vth measurement parameters are set in the following lines. You can modify these values according to the DUT characteristics.

```
1070  !-- Parameters for Vth measurement ----------------------
1075  Vgstart=0              ! Start voltage
1080  Vgstop=8               ! Stop voltage
1085  Vd=.1                  ! Vd
1090  Id_targ=1.E-6          ! Target current
1095  Rmp=50                 ! Ramp rate                  (Meas_vth)
1100  Integ=5.E-3            ! Feedback integration time  (Meas_vth)
1105  Vgstep=.01             ! Vgstep                     (Meas_vth2)
1110  Igcomp=.1              ! Ig compliance              (Meas_vth2)
1115  Idcomp=1.1E-6          ! Id compliance              (Meas_vth2)
1120  Hold=0                 ! Hold                       (Meas_vth2)
1125  !
```

Parameter Name	Description	Notice
`Vgstart`	Gate start voltage	Necessary both for Meas_vth and Meas_vth2
`Vgstop`	Gate stop voltage	Necessary both for Meas_vth and Meas_vth2
`Vd`	Drain voltage	Necessary both for Meas_vth and Meas_vth2
`Id_targ`	Target drain current	Necessary both for Meas_vth and Meas_vth2
`Rmp`	Ramp rate of AFU	Necessary only for Meas_vth
`Integ`	Feedback integration time	Necessary only for Meas_vth
`Vgstep`	Gate step voltage	Necessary only for Meas_vth2
`Igcomp`	Gate current compliance	Necessary only for Meas_vth2
`Idcomp`	Drain current compliance	Necessary only for Meas_vth2
`Hold`	Hold time	Necessary only for Meas_vth2

Above parameters are passed to `Meas_vth` or `Meas_vth2` subroutine as COM variables. If `Meas_vth` or `Meas_vth2` is modified and some parameters are added to subroutine, add these parameters to `COM /M_vth/` declaration in lines 1030, 1745, and 1810.

These subroutines connect the gate, drain, source, and substrate pins to *separate* SMUs, and execute the Vth measurements. Modify the source program as required.

Pulse Level, Leading Edge, and Trailing Edge for Write/Erase

The pulse level, leading edge, and trailing edge parameters for write/erase are set from line 1345. For conventional NOR type, two pulse waveforms are required for write. But NAND type only requires one pulse waveform, so fewer pulse parameters are required as in the following example:

```
1345    Level_w=20          !(V)      ! Gate pulse level (Write)
1350    Level_e=20          !(V)      ! Source, drain, sub pulse level (Erase)
1355    Ledge_w=5.E-8       !(s)      ! Gate leading edge (Write)
1360    Ledge_e=5.E-8       !(s)      ! Source, drain, sub leading edge (Erase)
1365    Tedge_w=5.E-8       !(s)      ! Gate trailing edge (Write)
1370    Tedge_e=5.E-8       !(s)      ! Source, drain, sub trailing edge (Erase)
1390    !
```

Parameter Name	Description
`Level_w`	Gate pulse level for write
`Level_e`	Source, drain, substrate pulse level for erase
`Ledge_w`	Gate pulse leading edge for write
`Ledge_e`	Source, drain, substrate pulse leading edge for erase
`Tedge_w`	Gate pulse trailing edge for write
`Tedge_e`	Source, drain, substrate pulse trailing edge for erase

For NAND type, you need to modify line 1215 as follows:

```
1215  REAL Level_w, Width_w, Delay_w, Ledge_w, Tedge_w
```

Pulse Width and Delay Time for Write/Erase

Pulse width settings for *write* pulse are set after line 1400.
This program sets pulse width so that total accumulated pulse width (`Total_width_w`) becomes 100ns (1E−7s), 200ns (2E−7s), 500ns (5E−7s), 1μs (1E−6s), ... , 500ms (5E−1s).

The total accumulated pulse width (`Total_width_w`) is determined by the equation $x \times 10^{\mathrm{Loop1}}$ (x and `Loop1` are integers). `Loop1` is loop counter for FOR—NEXT loop and is incremented from −7 to −1. Within each loop, x changes as follows: 1, 2, 5.

For NOR type, you specify write parameter settings for both gate and drain, but for NAND type, you specify parameter settings only for the gate as in following example:

```
1400    !--- Accumulated write pulse width vs. Vth --------------------
1405    FOR Loop1=-6 TO -1
1410      Width_w=10^Loop1-Total_width_w   ! Write pulse width
1415      Delay_w=0                        ! Write pulse delay time
1420      Total_width_w=10^Loop1           ! Accumulated pulse width
1425      GOSUB Write_meas_plot            ! Apply write pulse and meas Vth
1430      !
1435      Width_w=2*10^Loop1-Total_width_w ! Write pulse width
1440      Delay_w=0                        ! Write pulse delay time
1445      Total_width_w=2*10^Loop1         ! Accumulated pulse width
1450      GOSUB Write_meas_plot            ! Apply write pulse and meas Vth
1455      !
1460      Width_w=5*10^Loop1-Total_width_w ! Write pulse width
1465      Delay_w=0                        ! Write pulse delay time
1470      Total_width_w=5*10^Loop1         ! Accumulated pulse width
1475      GOSUB Write_meas_plot            ! Apply write pulse and meas Vth
1505    NEXT Loop1
```

Parameter Name	Description
Width_w	Gate pulse width for write
Delay_w	Gate pulse delay time for write

For NAND type, you need to modify line 1215 as follows:

```
1215  REAL Level_w, Width_w, Delay_w, Ledge_w, Tedge_w
```

Pulse width settings for *erase* pulse are set after line 1515.

This program sets pulse width so that total accumulated pulse width (Total_width_e) becomes 100ns (1E−7s), 200ns (2E−7s), 500ns (5E−7s), 1μs (1E−6s), ... , 500ms (5E−1s), which is same as for write pulse setting.

You can modify this part as in example below (loop counter is changed).

```
1515    !--- Accumulated erase pulse width vs. Vth --------------------
1520    FOR Loop2=-6 TO -1
1525      Width_e=10^Loop2-Total_width_e    ! Erase pulse width
1530      Delay_e=0                         ! Erase pulse delay time
1535      Total_width_e=10^Loop2            ! Accumulated pulse width
1540      GOSUB Erase_meas_plot             ! Apply erase pulse and meas Vth
1545      !
1550      Width_e=2*10^Loop2-Total_width_e  ! Erase pulse width
1555      Delay_e=0                         ! Erase pulse delay time
1560      Total_width_e=2*10^Loop2          ! Accumulated pulse width
1565      GOSUB Erase_meas_plot             ! Apply erase pulse and meas Vth
1570      !
1575      Width_e=5*10^Loop2-Total_width_e  ! Erase pulse width
1580      Delay_e=0                         ! Erase pulse delay time
1585      Total_width_e=5*10^Loop2          ! Accumulated pulse width
1590      GOSUB Erase_meas_plot             ! Apply erase pulse and meas Vth
1595    NEXT Loop2
```

Parameter Name	Description
Width_e	Source, drain, substrate pulse width for erase
Delay_e	Source, drain, substrate pulse delay time for erase

Write, Vth Measurements, and Plot

Write, Vth measurements, and plot are performed by the Write_meas_plot subroutine.

Replace Write_nor1 with Write_nand1, which are both programs of the PARA.F library.

Plot is made by Plot_x1y, which is a subprogram of SEMIGRAPH library. See Appendix A.

```
1615 Write_meas_plot:!
1620    Num_of_data_w=Num_of_data_w+1
1625    Write_nand1(Level_w,Width_w,Ledge_w,Tedge_w)
1630    Meas_vth2(Vth_w(Num_of_data_w))
1635    IF Num_of_data_w>1 THEN
1640      Plot_x1y(Total_width_w,Total_width_w-Width_w(2),Num_of_data_w,Clr_wrt,Vth_w(*))
1645    END IF
1650    RETURN
1655    !
```

Erase, Vth Measurements, and Plot

Erase, Vth measurements, and plot are performed by the `Erase_meas_plot` subroutine.

Replace `Erase_nor1` with `Erase_nand1`, which are both subprograms of `PARA.F` library.

Plot is made by `Plot_x1y`, which is a subprogram of `SEMIGRAPH` library. See Appendix A.

```
1660 Erase_meas_plot:!
1665    Num_of_data_e=Num_of_data_e+1
1670    Erase_nand1(Level_e,Width_e,Ledge_e,Tedge_e)
1675    Meas_vth2(Vth_e(Num_of_data_e))
1680    IF Num_of_data_e>1 THEN
1685      Plot_x1y(Total_width_e,Total_width_e-Width_e,Num_of_data_e,Clr_ers,Vth_e(*))
1690    END IF
1695    RETURN
```

Graphics Axes

Setting of graphic axes is made by following lines:

```
1265    X1ax_min=1.00E-7                              !
1270    X1ax_max=1.E+0                                !
1275    X2ax_min=1.00E-7                              ! Parameters
1280                                                  !  for Graph axis
1285    Vth_ax_min=0                                  !
1290    Vth_ax_max=8                                  !
1295    Yax_n$="Vth (V)"                              !
1300    G_name$="Accumulated Pulse Width Dep."        !
1305    X1ax_n$="Erase pulse width (s)"               !
1310    X2ax_n$="Write pulse width (s)"               !
1315    Clr_wrt=2    ! Pen Color for Write            !
1320    Clr_ers=4    ! Pen Color for Erase            !
1325    !
1330    !
1335    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0,"","",X2ax_n$,
 X2ax_min)
```

Modify these parameters as required. See Appendix A.

Customization of ACC_PWD_VTH for Other Cell Types

This section describes how to modify ACC_PWD_VTH for cell types that are not conventional NOR type or NAND type.

Creation of Write and Erase Subprograms

The PARA.F library has the following single write and single erase subprograms:

Cell Type	Write	Erase
Conventional NOR type	Write_nor1	Erase_nor1
3.3V NOR type	Write_nor2, Write_nor3[1]	Erase_nor2, Erase_nor3[1]
NAND type	Write_nand1	Erase_nand1

1 The Write_nor3 and Erase_nor3 subprograms use the multiplexer switch on the cable adapter.

For the 3.3V NOR type cell described in chapter 1, Write_nor2 and Erase_nor2, or Write_nor3 and Erase_nor3 can be used.

For other cell types, you must create new subprograms for writing and erasing.

Before creating new subprograms, you must consider the pin locations of the cell. The TEG must be designed so that the cell terminals that require different pulse waveforms can connect to separate HF ports, and cell terminals that require the same pulse waveform can connect to the same HF port.

If the TEG has already been designed, and the correct connections cannot be made, you need to change the bond wiring for package devices, or use bypass wiring on the probe card for wafer devices.

The above eight subprograms consist of TIS subprograms, so you can easily refer to the source program code and modify the eight subprograms as desired. You can create single write, single erase, and repeating write/erase subprograms for your desired device, which has the pin settings assigned by Set_fr subprogram.

Setup

Connect PGUs to HF ports according to the subprograms that you create.

Pin Assignment of the DUT

The pin assignments of the DUT are made after line 1040. Modify these assignments according to the DUT pin locations.

Threshold Voltage Measurement

The write pulse (or erase pulse) parameters are set, then `Write_meas_plot` (or `Erase_meas_plot`) subroutine is called. In the subroutine, write (or erase) pulse is applied to the cell, Vth measurement is made, and the data is plotted to the graph. This loop is repeated with increasing pulse widths.

Two Vth measurement subroutines are available: `Meas_vth` (which uses AFU and extracts Vth by analog feedback measurement) and `Meas_vth2` (which extracts Vth by linear search measurement). `Meas_vth2` is used in the program, but you can use `Meas_vth` by changing lines 1630 and 1675.

The Vth measurement parameters are set in lines 1070 to 1120. You can modify these values according to the DUT characteristics:

Parameter Name	Description	Notice
`Vgstart`	Start gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vgstop`	Stop gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vd`	Drain voltage	Necessary both for Meas_vth and Meas_vth2
`Id_targ`	Target drain current	Necessary both for Meas_vth and Meas_vth2
`Rmp`	Ramp rate of AFU	Necessary only for Meas_vth
`Integ`	Feedback integration time	Necessary only for Meas_vth
`Vgstep`	Gate step voltage	Necessary only for Meas_vth2
`Igcomp`	Gate current compliance	Necessary only for Meas_vth2
`Idcomp`	Drain current compliance	Necessary only for Meas_vth2
`Hold`	Hold time	Necessary only for Meas_vth2

Above parameters are passed to `Meas_vth` or `Meas_vth2` subroutine as COM variables. If `Meas_vth` or `Meas_vth2` is modified and some parameters are added to subroutine, add these parameters to `COM /M_vth/` declaration in lines 1030, 1745, and 1810.

These subroutines connect the gate, drain, source, and substrate pins to *separate* SMUs, and execute the Vth measurements. Modify the source program as required.

Pulse Level, Leading Edge, and Trailing Edge for Write/Erase

The pulse level, leading edge, and trailing edge parameters for write/erase are set from line 1345. Modify these parameters according to the DUT characteristics.

Pulse Width and Delay Time for Write/Erase

Pulse width settings for *write* are made between lines 1400 and 1505.
Pulse width settings for *erase* are made between lines 1515 and 1595.

This program sets the pulse width so that total accumulated pulse width (`Total_width_w` or `Total_width_e`) becomes 100 ns (1E−7 s), 200 ns (2E−7 s), 500 ns (5E−7 s), 1 μs (1E−6 s), ... , 500 ms (5E−1 s).

The total accumulated pulse width (`Total_width_w` or `Total_width_e`) is determined by the equation $x \times 10^{\mathrm{Loop}y}$ (x is 1, 2, or 5, and `Loop`y means `Loop1` for write pulse or `Loop2` for erase pulse). `Loop`y is the loop counter for the FOR—NEXT loop and is incremented from −7 to −1. Within each loop, x changes as follows: 1, 2, 5.

The width of *write* pulse applied to the *drain* is `Width_w(2)`, which is determined by subtracting `Total_width_w` (total accumulated write pulse width) from $x \times 10^{\mathrm{Loop1}}$, which will be the new total accumulated pulse width after applying the write pulse to the drain.

The width of *erase* pulse applied to the *source* is `Width_e`, which is determined by subtracting `Total_width_e` (total accumulated erase pulse width) from $x \times 10^{\mathrm{Loop2}}$, which will be the new total accumulated pulse width after applying the erase pulse to the source.

Modify these values as required.

Write, Vth Measurements, and Plot

Write, Vth measurements, and plot are performed by the `Write_meas_plot` subroutine.

```
1615 Write_meas_plot:!
1620    Num_of_data_w=Num_of_data_w+1
1625    Write_nor1(Level_w(*),Width_w(*),Delay_w(*),Ledge_w(*),Tedge_w(*))
1630    Meas_vth2(Vth_w(Num_of_data_w))
1635    IF Num_of_data_w>1 THEN
1640      Plot_x1y(Total_width_w,Total_width_w-Width_w(2),Num_of_data_w,Clr_wrt,Vth_w(*))
1645    END IF
1650    RETURN
1655    !
```

`Write_nor1` is a single write subprogram. You can replace `Write_nor1` with your own created subprogram or with another `PARA.F` library subprogram.

Plot is made by `Plot_x1y`, which is a subprogram of `SEMIGRAPH` library. See Appendix A.

Erase, Vth Measurements, and Plot

Erase, Vth measurements, and plot are performed by the `Erase_meas_plot` subroutine.

```
1660 Erase_meas_plot:!
1665    Num_of_data_e=Num_of_data_e+1
1670    Erase_nor1(Level_e,Width_e,Ledge_e,Tedge_e)
1675    Meas_vth2(Vth_e(Num_of_data_e))
1680    IF Num_of_data_e>1 THEN
1685      Plot_x1y(Total_width_e,Total_width_e-Width_e,Num_of_data_e,Clr_ers,Vth_e(*))
1690    END IF
1695    RETURN
```

`Erase_nor1` is a single erase subprogram. You can replace `Erase_nor1` with your own created subprogram or with another `PARA.F` library subprogram.

Plot is made by `Plot_x1y`, which is a subprogram of `SEMIGRAPH` library. See Appendix A.

Graphics Axes

Setting of graphic axes is made by following lines:

```
1265    X1ax_min=1.00E-7                            !
1270    X1ax_max=1.E+0                              !
1275    X2ax_min=1.00E-7                            ! Parameters
1280                                                !  for Graph axis
1285    Vth_ax_min=0                                !
1290    Vth_ax_max=8                                !
1295    Yax_n$="Vth (V)"                            !
1300    G_name$="Accumulated Pulse Width Dep."      !
1305    X1ax_n$="Erase pulse width (s)"             !
1310    X2ax_n$="Write pulse width (s)"             !
1315    Clr_wrt=2    ! Pen Color for Write          !
1320    Clr_ers=4    ! Pen Color for Erase          !
1325    !
1330    !
1335    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0,"","",
X2ax_n$,X2ax_min)
```

Modify these parameters as required. See Appendix A.

8

Pulse Width Dependency Measurement

This chapter describes the PWD_VTH application program, which is included with the HP 4062F. The meaning of PWD_VTH is "Pulse width or pulse level dependency of the cell threshold voltage".

Outline of Pulse Width Dependency Measurement

Pulse width dependency measurement is used to determine the best pulse conditions to apply to the cell. This measurement method is more similar to the actual write/erase operation than the accumulated pulse width dependency measurement, so you can more accurately know the best pulse conditions, such as pulse width. Outline of general pulse width dependency measurement is shown below. Application program "PWD_VTH" makes measurement as shown in the following flowchart:

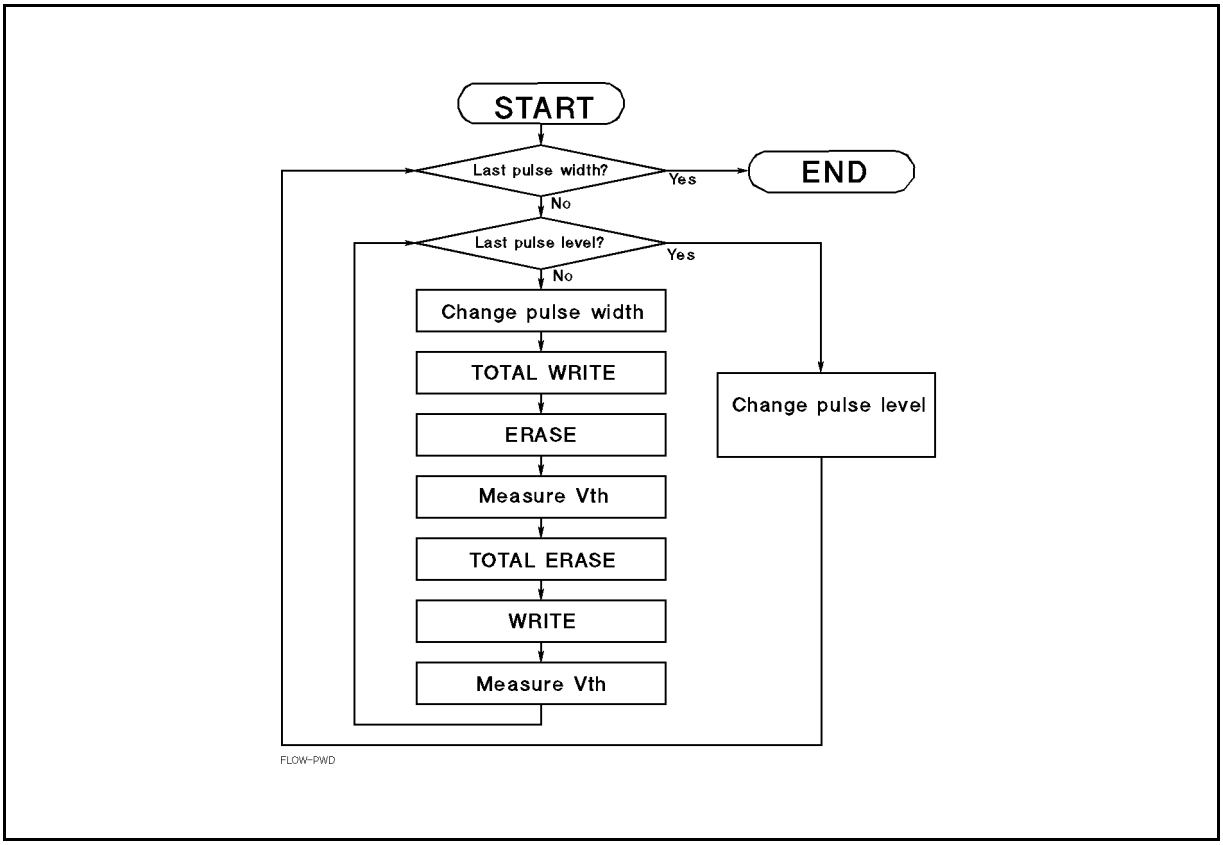


Figure 8-1. Flowchart of PWD_VTH Program

Figure 8-2 shows example characteristics of pulse width dependency.

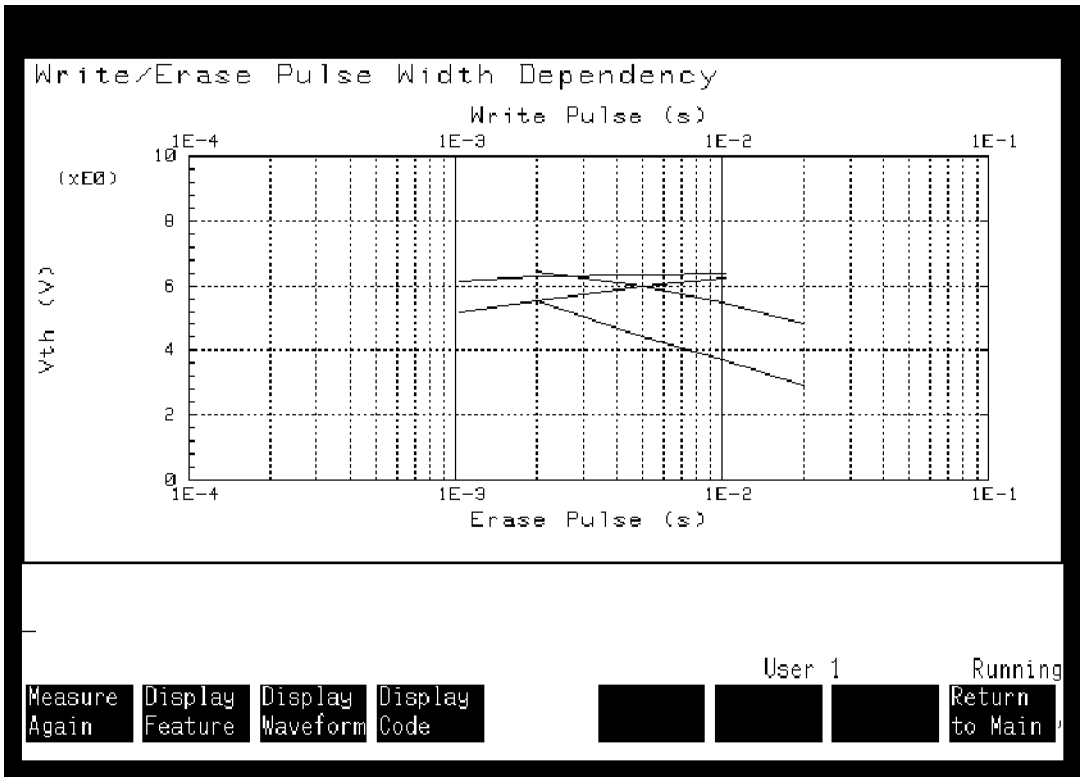


Figure 8-2. Example of Pulse Width Dependency Measurement

Total Write.
The purpose of this write is to perform a complete write, which means Vth reaches its saturated write value. So, for "total write", the pulse width must be long enough that Vth reaches its saturated write value. This pulse width does not change during the program.

Total Erase.
The purpose of this erase is to perform a complete erase, which means Vth reaches its saturated erase value. So, for "total erase", the pulse width must be long enough that Vth reaches its saturated erase value. This pulse width does not change during the program.

Write.
Write is performed by forcing write pulse to the conventional NOR cell as described in Chapter 1. The write pulse width is changed after each loop as shown in Figure 8-1.

Erase.
Erase is performed by forcing erase pulse to the conventional NOR cell as described in Chapter 1. Erase pulse width is changed after each loop as shown in Figure 8-1.

Threshold Voltage (Vth) Measurements.
Threshold voltage (Vth) is measured after each write or erase.

Necessary Configuration for PWD_VTH

This section describes the necessary hardware configuration and setup for using PWD_VTH.

Hardware

- Measurement controller: HP 9000 Series 300
- HP 4062F (at least three PGUs are necessary)

Setup

Connect between PGUs and HF ports as listed in the following table:

Port or PGU name	Port or terminal to be connected
PG11	HF port that can be connected to the drain
PG12	HF port that can be connected to the source
PG21	HF port that can be connected to the gate

Drain, source, and gate terminals must be connected to *different* HF ports. The program assumes the following pin locations for the cell. If the pin locations are different from the following, you can customize the program or can make additional connections with some wires so that the equivalent connections are made. See "Customization of PWD_VTH" if you need to customize your program.

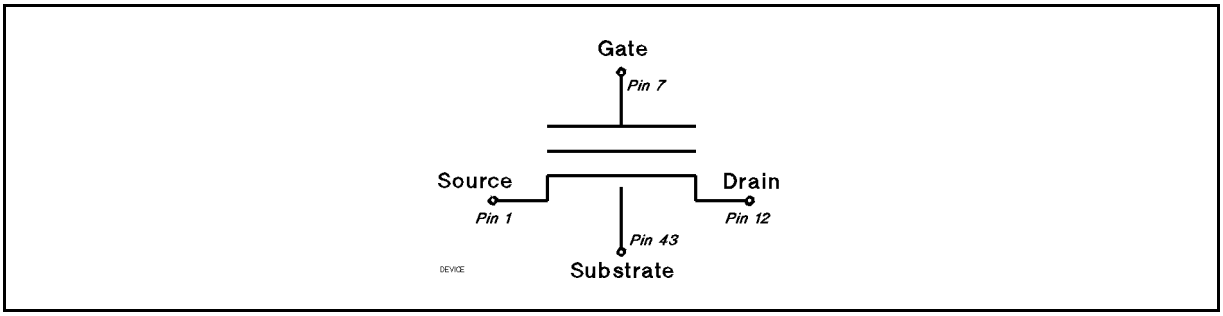


Figure 8-3. Pin Locations of Assumed Cell

Modify the FLCONFIG file according to the connections between the HF ports and PGUs. For example, the following is for the above pin locations.

```
1000 !!  RE-SAVE "/usr/pcs/etc/FLCONFIG"
1010 !!
1020 ! PG1  HP8110A  2725
1030 ! PG2  HP8110A  2726
1040 ! PG3  HP8110A  2727
1050 ! PG4  HP8110A  2728
1060 ! PG5  HP8110A  2729
1070 !
1080 ! HF1  PG12
1090 ! HF2  PG21
1100 ! HF3  PG11
```

Execution of PWD_VTH

1. Start the HP BASIC/UX environment.

   ```
   $ rmb  Return
   ```

2. Run START.F program of the HP 4062F.

   ```
   LOAD "/usr/pcs/bin/START.F",1  Return
   ```

3. Load the PWD_VTH program file.

   ```
   LOAD "/usr/pcs/demo/flash/PWD_VTH"  Return
   ```

4. Make necessary customization. See following pages.

5. Link TIS.F file by selecting `Link TIS` softkey.

6. Link PARA.F library file.

   ```
   LOADSUB ALL FROM "/usr/pcs/lib/PARA.F"  Return
   ```

7. Link SEMIGRAPH graphic library.

   ```
   LOADSUB ALL FROM "/usr/pcs/demo/flash/SEMIGRAPH"  Return
   ```

8. Execute the program by selecting `RUN` softkey.

Customization of PWD_VTH

This section describes how to modify PWD_VTH for cell types that are almost the same as the conventional NOR type described in Chapter 1. After loading PWD_VTH, go into editing mode by selecting EDIT softkey, then make modifications.

In the main program of PWD_VTH, pins are assigned to the DUT terminals, then parameters are specified for the threshold voltage (Vth) measurements. Then, system initialization is performed, and subroutine for pulse width dependency (Pwd_dep) is called.

In the Pwd_dep subroutine, the graphics axes are drawn, and the pulse levels are set. Then, total write, erase, Vth measurement, total erase, write, and Vth measurement are repeated with varying pulse widths, delays, and leading and trailing edges. Then, new pulse levels are set and this cycle is repeated.

Pin Assignments of the DUT

The pin assignments of the DUT are made in the following lines:

```
1050  !-- TEG pin assignment ---------------------------------
1055  Drain=12          ! Drain
1060  Gate=7            ! Gate
1065  Source=1          ! Source
1070  Sub=43            ! Sub
1075  !
```

Modify the pin numbers according to the actual connections.

Note that the drain, gate, and source pins must be connected to *different* HF ports. So, if these three pins are located close to each other, you may need to change the bonding wires of package device, or add some bypass wires on your probe card for wafer devices so that the pins can be connected to *different* HF ports. Connect PGUs to HF ports so that following connections are realized: drain pin—PG11, source pin—PG12, and gate pin—PG21.

Parameters for Total Write

Parameters for total write are set in the following lines:

```
1080  !-- Parameters for Perfect Write ------------------------
1085  Level_w_p(1)=14!(V)      ! Gate pulse level
1090  Level_w_p(2)=7!(V)       ! Drain pulse level
1095  Width_w_p(1)=1.02E-1!(s) ! Gate pulse width
1100  Width_w_p(2)=1.E-1!(s)   ! Drain pulse width
1105  Delay_w_p(1)=1.E-3!(s)   ! Gate pulse delay time
1110  Delay_w_p(2)=2.E-3!(s)   ! Drain pulse delay time
1115  Ledge_w_p(1)=1.E-4!(s)   ! Gate pulse leading edge
1120  Ledge_w_p(2)=1.E-4!(s)   ! Drain pulse leading edge
1125  Tedge_w_p(1)=1.E-4!(s)   ! Gate pulse trailing edge
1130  Tedge_w_p(2)=1.E-4!(s)   ! Drain pulse trailing edge
1135  !
```

Parameter Name	Description
Level_w_p(1)	Gate pulse level for total write
Level_w_p(2)	Drain pulse level for total write
Width_w_p(1)	Gate pulse width for total write
Width_w_p(2)	Drain pulse width for total write
Delay_w_p(1)	Delay time of gate pulse for total write
Delay_w_p(2)	Delay time of drain pulse for total write
Ledge_w_p(1)	Leading edge of gate pulse for total write
Ledge_w_p(2)	Leading edge of drain pulse for total write
Tedge_w_p(1)	Trailing edge of gate pulse for total write
Tedge_w_p(2)	Trailing edge of drain pulse for total write

Modify each parameter setting as required.

Parameters for Total Erase

Parameters for the total erase are set in the following lines:

```
1140  !-- Parameters for Perfect Erase ------------------------
1145  Level_e_p=11!(V)       ! Source pulse level
1150  Width_e_p=.9  !(s)      ! Source pulse width
1155  Ledge_e_p=1.E-4!(s)    ! Source pulse leading edge
1160  Tedge_e_p=1.E-4!(s)    ! Source pulse trailing edge
1165  !
```

Parameter Name	Description
Level_e_p	Source pulse level for total erase
Width_e_p	Source pulse width for total erase
Ledge_e_p	Leading edge of source pulse for total erase
Tedge_e_p	trailing edge of source pulse for total erase

Modify each parameter setting as required.

Threshold Voltage Measurement

Two Vth measurement subroutines are available: `Meas_vth` (which uses AFU and extracts Vth by analog feedback measurement) and `Meas_vth2` (which extracts Vth by linear search measurement). In the `PWD_VTH` program, `Meas_vth2` is used, but you can use `Meas_vth` by changing lines 1815 and 1865.

```
1805        !-- Erase the cell ----------------------------
1810        Erase_nor1(Level_e,Width_e,Ledge_e,Tedge_e)
1815        Meas_vth2(Vth_e(Loop2))
  :
1855        !-- Write the cell ----------------------------
1860        Write_nor1(Level_w(*),Width_w(*),Delay_w(*),Ledge_w(*),Tedge_w(*))
1865        Meas_vth2(Vth_w(Loop2))
```

The Vth measurement parameters are set in the following lines. You can modify these values according to the DUT characteristics.

```
1170  !-- Parameters for Vth measurement ----------------------
1175  Vgstart=0              ! Start voltage
1180  Vgstop=8               ! Stop voltage
1185  Vd=.1                  ! Vd
1190  Id_targ=1.E-6          ! Target current
1195  Rmp=50                 ! Ramp rate                  (Meas_vth)
1200  Integ=5.E-3            ! Feedback integration time  (Meas_vth)
1205  Vgstep=.01             ! Vgstep                     (Meas_vth2)
1210  Igcomp=.1              ! Ig compliance              (Meas_vth2)
1215  Idcomp=1.1E-6          ! Id compliance              (Meas_vth2)
1220  Hold=0                 ! Hold                       (Meas_vth2)
1225  !
```

Parameter Name	Description	Notice
`Vgstart`	Start gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vgstop`	Stop gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vd`	Drain voltage	Necessary both for Meas_vth and Meas_vth2
`Id_targ`	Target drain current	Necessary both for Meas_vth and Meas_vth2
`Rmp`	Ramp rate of AFU	Necessary only for Meas_vth
`Integ`	Feedback integration time	Necessary only for Meas_vth
`Vgstep`	Gate step voltage	Necessary only for Meas_vth2
`Igcomp`	Gate current compliance	Necessary only for Meas_vth2
`Idcomp`	Drain current compliance	Necessary only for Meas_vth2
`Hold`	Hold time	Necessary only for Meas_vth2

Above parameters are passed to `Meas_vth` or `Meas_vth2` subroutine as COM variables. If `Meas_vth` or `Meas_vth2` is modified and some parameters are added to subroutine, add these parameters to `COM /M_vth/` declaration in lines 1040, 1950, and 2015.

These subroutines connect the gate, drain, source, and substrate pins to *separate* SMUs, and execute the Vth measurements. Modify the source program as required.

Pulse Level Parameters for Write/Erase

The pulse level parameters for write/erase are set in lines 1435 to 1485.

The variable `Loop1` controls the outer loop, which determines which pulse level is used while the pulse widths are varied during the measurements. The `PWD_VTH` program uses "`FOR Loop1=1 TO 2`" in line 1435. So two sets of pulse level are used.

So, if you want to use four sets of pulse level, change line 1435 to "`FOR Loop1=1 TO 4`", and add two CASEs of pulse level parameters. If you only want to use one pulse level, change line 1435 to "`FOR Loop1=1 TO 1`", and delete one CASE of pulse level parameters.

```
1435    FOR Loop1=1 TO 2
1440      SELECT Loop1
1445      CASE 1
1450        Level_w(1)=14  !(V)      ! Gate pulse level (Write)
1455        Level_w(2)=7   !(V)      ! Drain pulse level (Write)
1460        Level_e=11     !(V)      ! Source pulse level (Erase)
1465      CASE 2
1470        Level_w(1)=13  !(V)      ! Gate pulse level (Write)
1475        Level_w(2)=6.5 !(V)      ! Drain pulse level (Write)
1480        Level_e=10.5   !(V)      ! Source pulse level (Erase)
1485      END SELECT
1490      !
```

Parameter Name	Description
`Level_w(1)`	Gate pulse level for write
`Level_w(2)`	Drain pulse level for write
`Level_e`	Source pulse level for write

Pulse Timing Parameters for Write/Erase

The pulse width, delay, leading edge, and trailing edge parameters for write/erase are set in lines 1495 to 1785.

The variable `Loop2` controls the inner loop, which determines which pulse timing parameters are used during the measurement. The `PWD_VTH` program uses "`FOR Loop2=1 TO 4`" in line 1495. So four sets of pulse timing parameters will be used for each pulse level.

So, if you want to use five sets, change line 1495 to "`FOR Loop2=1 TO 5`", and add one CASE of pulse timing parameters. If you want to use three sets, change line 1495 to "`FOR Loop2=1 TO 3`", and delete one CASE of pulse timing parameters.

```
1495      FOR Loop2=1 TO 4
1500        SELECT Loop2
1505        CASE 1
1510          Width_w(1)=1.2E-6 !(s)   ! Gate pulse width        !
1515          Width_w(2)=1.0E-6 !(s)   ! Drain pulse width       ! Write
1520          Delay_w(1)=1.E-7 !(s)    ! Delay of Gate pulse     !  Parameters
1525          Delay_w(2)=2.E-7 !(s)    ! Delay of Drain pulse    !
1530          Ledge_w(1)=1.E-7 !(s)    ! Gate leading edge       !
1535          Ledge_w(2)=1.E-7 !(s)    ! Drain leading edge      !
1540          Tedge_w(1)=1.E-7 !(s)    ! Gate trailing edge      !
1545          Tedge_w(2)=1.E-7 !(s)    ! Drain trailing edge     !
1550          !
1555          Width_e=1.0E-4   !(s)    ! Source erase pulse width!
1560          Ledge_e=1.E-6    !(s)    ! Source leading edge     ! Erase
1565          Tedge_e=1.E-6    !(s)    ! Source trailing edge    !  Parameters
1570          !
1575        CASE 2
              :
              :
```

Parameter Name	Description
`Width_w(1)`	Gate pulse width for write
`Width_w(2)`	Drain pulse width for write
`Delay_w(1)`	Delay time of gate pulse for write
`Delay_w(2)`	Delay time of drain pulse for write
`Ledge_w(1)`	Leading edge of gate pulse for write
`Ledge_w(2)`	Leading edge of drain pulse for write
`Tedge_w(1)`	Trailing edge of gate pulse for write
`Tedge_w(2)`	Trailing edge of drain pulse for write
`Width_e`	Source pulse width for erase
`Ledge_e`	Leading edge of source pulse for erase
`Tedge_e`	Trailing edge of source pulse for erase

Sequence for Pulse Width Dependency Measurement

For each set of pulse timing parameters, PWD_VTH executes the following sequence: total write, erase, Vth measurement, plot, total erase, write, Vth measurement, and plot. You can modify this sequence if desired.

```
1795        !-- Write the cell completely -----------------
1800        Write_nor1(Level_w_p(*),Width_w_p(*),Delay_w_p(*),Ledge_w_p(*),Tedge_w_p(*))
1805        !-- Erase the cell ----------------------------
1810        Erase_nor1(Level_e,Width_e,Ledge_e,Tedge_e)
1815        Meas_vth2(Vth_e(Loop2))
1820        IF Loop2>1 THEN
1825          Plot_x1y(Width_e,Width_e_old,Loop2,Clr_ers,Vth_e(*))
1830        END IF
1835        !
1840        !-- Erase the cell completely -----------------
1845        Erase_nor1(Level_e_p,Width_e_p,Ledge_e_p,Tedge_e_p)
1850        !
1855        !-- Write the cell ----------------------------
1860        Write_nor1(Level_w(*),Width_w(*),Delay_w(*),Ledge_w(*),Tedge_w(*))
1865        Meas_vth2(Vth_w(Loop2))
1870        IF Loop2>1 THEN
1875           Plot_x2y(Width_w(2),Width_w_old,Loop2,Clr_wrt,Vth_w(*))
1880        END IF
1885        !
```

For details about Plot_x1y and Plot_x2y, see Appendix A.

Graphics Axes

The graphics axes are set by the following lines:

```
1345    X1ax_min=1.00E-4                              !
1350    X1ax_max=1.E-1                                !
1355    X2ax_min=1.00E-6                              ! Parameters
1360                                                  !  for Graph axis
1365    Vth_ax_min=0                                  !
1370    Vth_ax_max=8                                  !
1375    Yax_n$="Vth (V)"                              !
1380    G_name$="  Write/Erase Pulse Width Dep."      !
1385    X1ax_n$="Erase pulse width (s)"               !
1390    X2ax_n$="Write pulse width (s)"               !
1395    Clr_wrt=2    ! Pen Color for Write            !
1400    Clr_ers=4    ! Pen Color for Erase            !
1405    !
1410    Width_w_old=0
1415    Width_e_old=0
1420    !
1425    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0,"","",
X2ax_n$,X2ax_min)
```

You can modify these parameters if desired.

For details about Semiloggraph1, see Appendix A.

Customization of PWD_VTH for NAND Type Cell

This section describes how to modify PWD_VTH for cell types that are similar to the NAND type described in Chapter 1. For NAND type, only two PGUs are necessary. So, you need to modify the connections and the program.

Port or Terminal Name	Port or terminal to be connected
PG11	HF port that can be connected to the gate
PG12	HF port that can be connected to drain, source, and substrate

The drain, source, and substrate terminals must be connected to PG12, so the DUT must be designed so that these terminals can be connected to the same HF port. If this is not possible, you can use BNC T connectors to connect PG12 to multiple HF ports.

After loading PWD_VTH, go into editing mode by selecting EDIT softkey, then make modifications as described in this section.

If DUT is same type as NAND cell described in Chapter 1, you can use Write_nand1 and Erase_nand1 subprograms for writing and erasing.

Parameters passed to Write_nand1 are pulse level, width, leading edge, and trailing edge. But for a conventional NOR cell, Write_nor1 is used, which requires delay time in addition to the other parameters. Delay time is not necessary for NAND type cell, so number of parameters is less than for NOR type cell.

Also, Write_nand1 requires parameters for one pulse waveform, whereas Write_nor1 requires parameters for two pulse waveforms.

Pin Assignments of the DUT

The pin assignments of the DUT are made in the following lines:

```
1050  !-- TEG pin assignment ---------------------------------
1055  Drain=12          ! Drain
1060  Gate=7            ! Gate
1065  Source=1          ! Source
1070  Sub=43            ! Sub
1075  !
```

Modify the pin numbers according to the actual connections. Note that the HF port of the gate pin must *not* be the same as the HF port for the drain, source, and substrate pins. You can connect the drain, source, and substrate pins to the same HF port or to different HF ports, but make sure these pins can connect to PG12. And gate must be able to connect to PG11.

If the DUT cannot be connected properly, redesign bond wiring for package devices, or add some bypass wires on your probe card for wafer devices.

Parameters for Total Write

The pulse level and pulse timing parameters for total write are set from line 1080. For conventional NOR type, two pulse waveforms are required for write. But NAND type only requires one pulse waveform, so fewer parameters are required as in the following example:

```
1080  !-- Parameters for Perfect Write ------------------------
1085  Level_w_p=18!(V)      ! Gate pulse level
1090  Width_w_p=1.0E-5!(s)  ! Gate pulse width
1095  Ledge_w_p=1.E-6!(s)   ! Gate pulse leading edge
1100  Tedge_w_p=1.E-6!(s)   ! Gate pulse trailing edge
1135  !
```

Parameter Name	Description
Level_w_p	Gate pulse level for total write
Width_w_p	Gate pulse width for total write
Delay_w_p	Delay time of gate pulse for total write
Ledge_w_p	Leading edge of gate pulse for total write
Tedge_w_p	Trailing edge of gate pulse for total write

For conventional NOR type, the total write parameters are set in an array variable. But for NAND type, you use a single variable, so you need to change lines 1025 and 1295 as follows:

```
COM /Pct_write/ Level_w_p, Width_w_p, Delay_w_p, Ledge_w_p, Tedge_w_p
```

Parameters for Total Erase

Parameters for the total erase are set in the following lines. For example, you can change the parameters as follows:

```
1140  !-- Parameters for Perfect Erase ------------------------
1145  Level_e_p=18!(V)       ! Erase pulse level
1150  Width_e_p=1.0E-5 !(s)  ! Erase pulse width
1155  Ledge_e_p=1.E-6!(s)    ! Erase pulse leading edge
1160  Tedge_e_p=1.E-6!(s)    ! Erase pulse trailing edge
1165  !
```

Parameter Name	Description
Level_e_p	Pulse level of source, drain, and substrate for total erase
Width_e_p	Pulse width of source, drain, and substrate for total erase
Ledge_e_p	Leading edge of source, drain, and substrate pulses for total erase
Tedge_e_p	Trailing edge of source, drain, and substrate pulses for total erase

Threshold Voltage Measurement

Two Vth measurement subroutines are available: `Meas_vth` (which uses AFU and extracts Vth by analog feedback measurement) and `Meas_vth2` (which extracts Vth by linear search measurement). In the `PWD_VTH` program, `Meas_vth2` is used, but you can use `Meas_vth` by changing lines 1815 and 1865.

```
1805        !-- Erase the cell ----------------------------
1810        Erase_nor1(Level_e,Width_e,Ledge_e,Tedge_e)
1815        Meas_vth2(Vth_e(Loop2))
  :
1855        !-- Write the cell ----------------------------
1860        Write_nor1(Level_w(*),Width_w(*),Delay_w(*),Ledge_w(*),Tedge_w(*))
1865        Meas_vth2(Vth_w(Loop2))
```

The Vth measurement parameters are set in the following lines. You can modify these values according to the DUT characteristics.

```
1170  !-- Parameters for Vth measurement ----------------------
1175  Vgstart=0              ! Start voltage
1180  Vgstop=8               ! Stop voltage
1185  Vd=.1                  ! Vd
1190  Id_targ=1.E-6          ! Target current
1195  Rmp=50                 ! Ramp rate                  (Meas_vth)
1200  Integ=5.E-3            ! Feedback integration time  (Meas_vth)
1205  Vgstep=.01             ! Vgstep                     (Meas_vth2)
1210  Igcomp=.1              ! Ig compliance              (Meas_vth2)
1215  Idcomp=1.1E-6          ! Id compliance              (Meas_vth2)
1220  Hold=0                 ! Hold                       (Meas_vth2)
1225  !
```

Parameter Name	Description	Notice
`Vgstart`	Start gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vgstop`	Stop gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vd`	Drain voltage	Necessary both for Meas_vth and Meas_vth2
`Id_targ`	Target drain current	Necessary both for Meas_vth and Meas_vth2
`Rmp`	Ramp rate of AFU	Necessary only for Meas_vth
`Integ`	Feedback integration time	Necessary only for Meas_vth
`Vgstep`	Gate step voltage	Necessary only for Meas_vth2
`Igcomp`	Gate current compliance	Necessary only for Meas_vth2
`Idcomp`	Drain current compliance	Necessary only for Meas_vth2
`Hold`	Hold time	Necessary only for Meas_vth2

Above parameters are passed to `Meas_vth` or `Meas_vth2` subroutine as COM variables. If `Meas_vth` or `Meas_vth2` is modified and some parameters are added to subroutine, add these parameters to `COM /M_vth/` declaration in lines 1040, 1950, and 2015.

These subroutines connect the gate, drain, source, and substrate pins to *separate* SMUs, and execute the Vth measurements. Modify the source program as required.

Pulse Level Parameters for Write/Erase

The pulse level parameters for write/erase are set after line 1435.

The variable `Loop1` controls the outer loop, which determines which pulse level is used while the pulse widths are varied during the measurements. The `PWD_VTH` program uses "`FOR Loop1=1 TO 2`" in line 1435. So two sets of pulse level are used.

So, if you want to use four sets of pulse level, change line 1435 to "`FOR Loop1=1 TO 4`", and add two CASEs of pulse level parameters. If you only want to use one pulse level, change line 1435 to "`FOR Loop1=1 TO 1`", and delete one CASE of pulse level parameters.

For NAND type, you set up only one write waveform (for NOR, you set up two write waveforms), so you change the lines as in the following example:

```
1435    FOR Loop1=1 TO 2
1440      SELECT Loop1
1445      CASE 1
1450        Level_w=18     !(V)      ! Write pulse level
1455        Level_e=18     !(V)      ! Erase pulse level
1460      CASE 2
1465        Level_w=20     !(V)      ! Write pulse level
1470        Level_e=20     !(V)      ! Erase pulse level
1475      END SELECT
1485      !
```

Parameter Name	Description
`Level_w`	Gate pulse level for write
`Level_e`	Pulse level of source, drain, and substrate pulse for erase

For conventional NOR type, the write parameters for two waveforms are set in an array variable. But for NAND type, you use a single variable, so you need to change line 1320 as follows:

```
1320    REAL Level_w, Width_w, Delay_w, Ledge_w, Tedge_w
```

Pulse Timing Parameters for Write/Erase

The pulse width, leading edge, and trailing edge parameters for write/erase are set after line 1495.

The variable `Loop2` controls the inner loop, which determines which pulse timing parameters are used during the measurement. The `PWD_VTH` program uses "`FOR Loop2=1 TO 4`" in line 1495. So fours sets of pulse timing parameters will be used for each pulse level.

So, if you want to use five sets, change line 1495 to "`FOR Loop2=1 TO 5`", and add one CASE of pulse timing parameters. If you want to use three sets, change line 1495 to "`FOR Loop2=1 TO 3`", and delete one CASE of pulse timing parameters.

For NAND type, you set up only one write waveform (for NOR, you set up two write waveforms), so you change the lines as in the following example:

```
1495      FOR Loop2=1 TO 6
1500        SELECT Loop2
1505        CASE 1
1510          Width_w=1.0E-6   !(s)    ! Write pulse width
1515          Ledge_w=1.E-7    !(s)    ! Write leading edge
1520          Tedge_w=1.E-7    !(s)    ! Write trailing edge
1525          !
1530          Width_e=1.0E-6   !(s)    ! Write erase pulse width
1535          Ledge_e=1.E-7    !(s)    ! Write leading edge
1540          Tedge_e=1.E-7    !(s)    ! Write trailing edge
1545          !
1550        CASE 2
  :
```

Parameter Name	Description
`Width_w`	Pulse width for write
`Ledge_w`	Leading edge of pulse for write
`Tedge_w`	Trailing edge of pulse for write
`Width_e`	Pulse width for erase
`Ledge_e`	Leading edge of pulse for erase
`Tedge_e`	Trailing edge of pulse for erase

For conventional NOR type, the write parameters for two waveforms are set in an array variable. But for NAND type, you use a single variable, so you need to change line 1320 as follows:

```
1320    REAL Level_w, Width_w, Delay_w, Ledge_w, Tedge_w
```

Sequence for Pulse Width Dependency Measurement

For each set of pulse timing parameters, `PWD_VTH` executes the following sequence: total write, erase, Vth measurement, plot, total erase, write, Vth measurement, and plot. You can modify this sequence if desired.

For NAND type, you use Write_nand1 and Erase_nand1 (not Write_nor1 and Erase_nor1). So change the lines as shown in the following example:

```
1795        !-- Write the cell completely -----------------
1800        Write_nand1(Level_w_p,Width_w_p,Ledge_w_p,Tedge_w_p)
1805        !
1810        !-- Erase the cell ----------------------------
1815        Erase_nand1(Level_e,Width_e,Ledge_e,Tedge_e)
1820        Meas_vth2(Vthm)
1825        !
1830        Vth_e(Loop2)=Vthm
1835        Plot_x1y(Width_e,Width_e_old,Loop2,Clr_ers,Vth_e(*))
1840        !
1845        !-- Erase the cell completely -----------------
1850        Erase_nand1(Level_e_p,Width_e_p,Ledge_e_p,Tedge_e_p)
1855        !
1860        !-- Write the cell ----------------------------
1865        Write_nand1(Level_w,Width_w,Ledge_w,Tedge_w)
1870        Meas_vth2(Vthm)
1875        !
1880        Vth_w(Loop2)=Vthm
1885        Plot_x2y(Width_w,Width_w_old,Loop2,Clr_wrt,Vth_w(*))
1890        !
```

For details about `Plot_x1y` and `Plot_x2y`, see Appendix A.

Graphics Axes

The graphics axes are set by the following lines:

```
1345    X1ax_min=1.00E-4                              !
1350    X1ax_max=1.E-1                                !
1355    X2ax_min=1.00E-6                              ! Parameters
1360                                                  !  for Graph axis
1365    Vth_ax_min=0                                  !
1370    Vth_ax_max=8                                  !
1375    Yax_n$="Vth (V)"                              !
1380    G_name$="  Write/Erase Pulse Width Dep."      !
1385    X1ax_n$="Erase pulse width (s)"               !
1390    X2ax_n$="Write pulse width (s)"               !
1395    Clr_wrt=2    ! Pen Color for Write            !
1400    Clr_ers=4    ! Pen Color for Erase            !
1405    !
1410    Width_w_old=0
1415    Width_e_old=0
1420    !
1425    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0,"","",
X2ax_n$,X2ax_min)
```

You can modify these parameters if desired.

For details about `Semiloggraph1`, see Appendix A.

Customization of PWD_VTH for Other Cell Types

This section describes how to modify PWD_VTH for cell types that are not conventional NOR type or NAND type.

Creation of Write and Erase Subprograms

The PARA.F library has the following single write and single erase subprograms:

Cell Type	Write	Erase
Conventional NOR type	Write_nor1	Erase_nor1
3.3V NOR type	Write_nor2, Write_nor3[1]	Erase_nor2, Erase_nor3[1]
NAND type	Write_nand1	Erase_nand1

1 The Write_nor3 and Erase_nor3 subprograms use the multiplexer switch on the cable adapter.

For the 3.3V NOR type cell described in chapter 1, you use Write_nor2 and Erase_nor2, or Write_nor3 and Erase_nor3.

For other cell types, you must create new subprograms for writing and erasing.

Before creating new subprograms, you must consider the pin locations of the cell. The TEG must be designed so that the cell terminals that require different pulse waveforms can connect to separate HF ports, and cell terminals that require the same pulse waveform can connect to the same HF port.

If the TEG has already been designed, and the correct connections cannot be made, you need to change the bond wiring for package devices, or use bypass wiring on the probe care for wafer devices.

The above eight subprograms contain TIS subprograms, so you can easily refer to the source program code and modify the eight subprograms as desired. You can create single write, single erase, and repeating write/erase subprograms for your desired device, which has the pin settings assigned by Set_fr subprogram.

Setup

Connect PGUs to HF ports according to the subprograms that you create.

Pin Assignments of the DUT

The pin assignments of the DUT are made after line 1050. Modify these assignments according to the DUT pin locations.

Parameters for Total Write

The pulse level and pulse timing parameters for total write are set from lines 1080 to 1130. Modify these parameters according to your created subprograms.

Parameters for Total Erase

The pulse level and pulse timing parameters for total erase are set from lines 1140 to 1160. Modify these parameters according to your created subprograms.

Threshold Voltage Measurement

Two Vth measurement subroutines are available: `Meas_vth` (which uses AFU and extracts Vth by analog feedback measurement) and `Meas_vth2` (which extracts Vth by linear search measurement). In the `PWD_VTH` program, `Meas_vth2` is used, but you can use `Meas_vth` by changing lines 1815 and 1865.

The Vth measurement parameters are set in lines 1170 to 1220. You can modify these values according to the DUT characteristics.

Parameter Name	Description	Notice
`Vgstart`	Start gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vgstop`	Stop gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vd`	Drain voltage	Necessary both for Meas_vth and Meas_vth2
`Id_targ`	Target drain current	Necessary both for Meas_vth and Meas_vth2
`Rmp`	Ramp rate of AFU	Necessary only for Meas_vth
`Integ`	Feedback integration time	Necessary only for Meas_vth
`Vgstep`	Gate step voltage	Necessary only for Meas_vth2
`Igcomp`	Gate current compliance	Necessary only for Meas_vth2
`Idcomp`	Drain current compliance	Necessary only for Meas_vth2
`Hold`	Hold time	Necessary only for Meas_vth2

Above parameters are passed to `Meas_vth` or `Meas_vth2` subroutine as COM variables. If `Meas_vth` or `Meas_vth2` is modified and some parameters are added to subroutine, add these parameters to `COM /M_vth/` declaration in lines 1040, 1950, and 2015.

These subroutines connect the gate, drain, source, and substrate pins to *separate* SMUs, and execute the Vth measurements. Modify the source program as required.

Pulse Level Parameters for Write/Erase

The pulse level parameters for write/erase are set after line 1435.

The variable `Loop1` controls the outer loop, which determines which pulse level is used while the pulse widths are varied during the measurements. The `PWD_VTH` program uses "`FOR Loop1=1 TO 2`" in line 1435. So two sets of pulse level are used.

So, if you want to use four sets of pulse level, change line 1435 to "`FOR Loop1=1 TO 4`", and add two CASEs of pulse level parameters. If you only want to use one pulse level, change line 1435 to "`FOR Loop1=1 TO 1`", and delete one CASE of pulse level parameters.

Pulse Timing Parameters for Write/Erase

The pulse width, delay, leading edge, and trailing edge parameters for write/erase are set after line 1495.

The variable `Loop2` controls the inner loop, which determines which pulse timing parameters are used during the measurement. The `PWD_VTH` program uses "`FOR Loop2=1 TO 4`" in line 1495. So four sets of pulse timing parameters will be used for each pulse level.

So, if you want to use five sets, change line 1495 to "`FOR Loop2=1 TO 5`", and add one CASE of pulse timing parameters. If you want to use three sets, change line 1495 to "`FOR Loop2=1 TO 3`", and delete one CASE of pulse timing parameters.

Sequence for Pulse Width Dependency Measurement

For each set of pulse timing parameters, `PWD_VTH` executes the following sequence: total write, erase, Vth measurement, plot, total erase, write, Vth measurement, and plot. You can modify this sequence (lines 1795 to 1895) if desired.

Graphics Axes

The graphics axes are set by the following lines:

```
1345    X1ax_min=1.00E-4                              !
1350    X1ax_max=1.E-1                                !
1355    X2ax_min=1.00E-6                              ! Parameters
1360                                                  !  for Graph axis
1365    Vth_ax_min=0                                  !
1370    Vth_ax_max=8                                  !
1375    Yax_n$="Vth (V)"                              !
1380    G_name$="  Write/Erase Pulse Width Dep."      !
1385    X1ax_n$="Erase pulse width (s)"               !
1390    X2ax_n$="Write pulse width (s)"               !
1395    Clr_wrt=2    ! Pen Color for Write            !
1400    Clr_ers=4    ! Pen Color for Erase            !
1405    !
1410    Width_w_old=0
1415    Width_e_old=0
1420    !
1425    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0,"","",
X2ax_n$,X2ax_min)
```

You can modify these parameters if desired.

For details about `Semiloggraph1`, see Appendix A.

9

Times Dependency Measurement

This chapter describes the `TIMES_VTH` application program, which is included with the HP 4062F. This program is for measuring the endurance of flash memory cells. The meaning of `TIMES_VTH` is "times dependency of the cell threshold voltage".

Outline of Times Dependency Measurement

Times dependency measurement is necessary to predict the lifetime (endurance) of a cell. This measures the cell characteristics against repeated write and erase.

The flowchart in Figure 9-1 shows how to execute the times dependency measurement. The write/erase cycle is forced repeatedly to the cell. After total number of write/erase cycles reaches 1, 10, 10^2, 10^3, 10^4, 10^5, and 10^6, Vth is measured for write, then measured for erase.

The HP 4062F allows you make these measurements in the minimum amount of time because the open and close timing of the drain terminal can automatically be synchronized with the pulses forced to other terminals by using the open/close switch on the cable adapter.

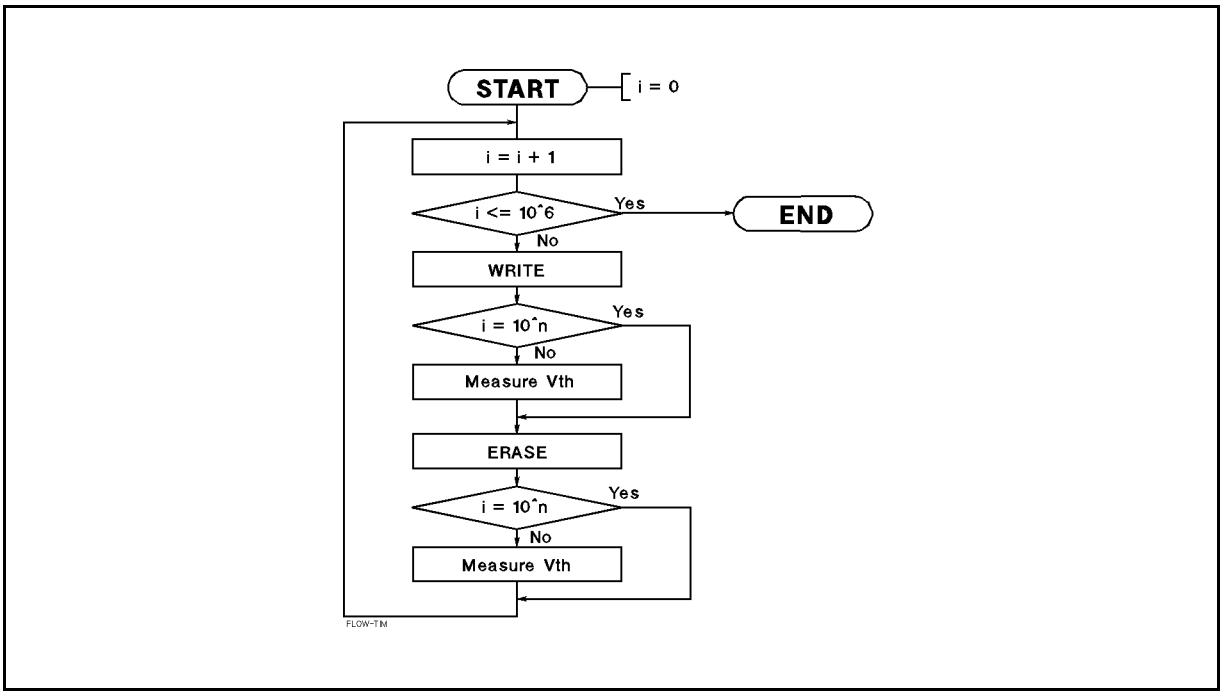


Figure 9-1. Flowchart of Times Dependency Measurement

Figure 9-2 shows example characteristics of the times dependency measurement.

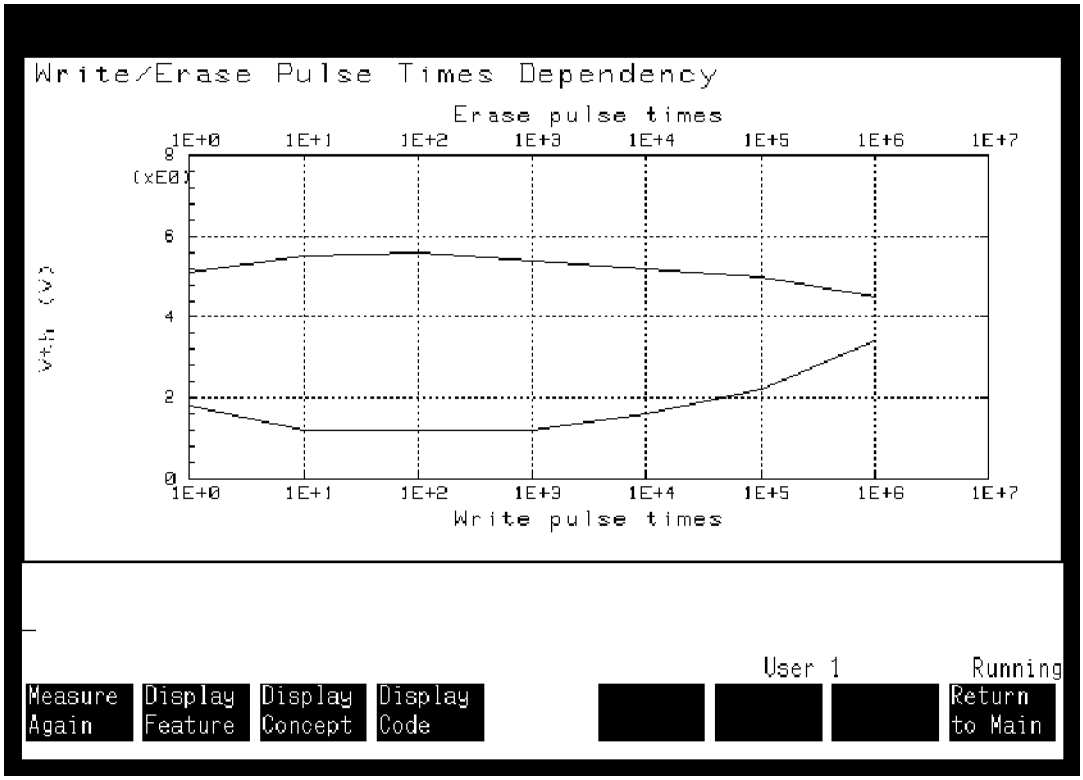


Figure 9-2. Example of Times Dependency Measurement

Necessary Configuration for TIMES_VTH

This section describes the necessary hardware configuration and setup for using TIMES_VTH.

Hardware

- Measurement controller: HP 9000 Series 300
- HP 4062F (at least four PGUs are necessary.)

Setup

Connect between PGUs and HF ports as listed in the following table:

Port or PGU name	Port or terminal to be connected
PG11	In terminal of SW2 or SW3
PG12	HF port that can be connected to the source
PG21	HF port that can be connected to the gate
PG22	CTRL SW1·2·3 terminal
Out of SW2 or SW3	HF port that can be connected to the drain

Drain, source, and gate terminals must be connected to *different* HF ports. TIMES_VTH assumes following pin locations for cell. If pin locations are different from following, customize TIMES_VTH or make additional connections with some wires so that the equivalent connections are made. See "Customization of TIMES_VTH" if you need to customize your program.

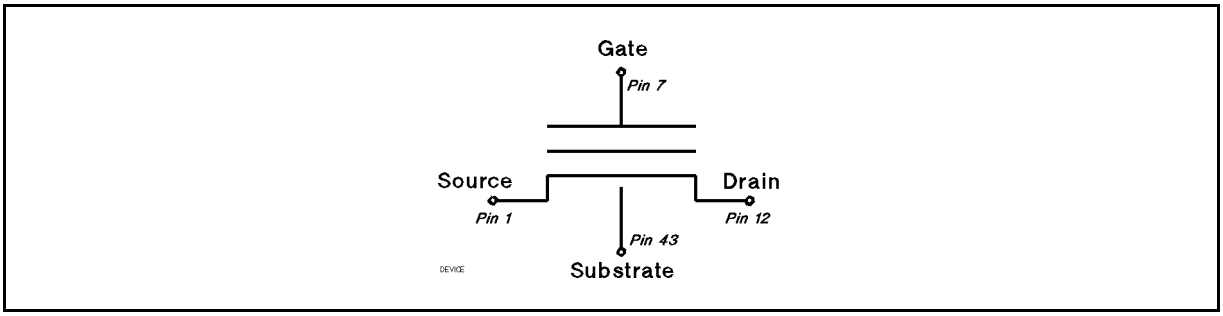


Figure 9-3. Pin Locations of Assumed Cell

Modify FLCONFIG file according to connections between HF ports and PGUs. For example, following is for above pin locations. Note that PG22, which controls the open/close switch, must *not* be specified in FLCONFIG file.

```
1000 !!  RE-SAVE "/usr/pcs/etc/FLCONFIG"
1010 !!
1020 ! PG1  HP8110A  2725
1030 ! PG2  HP8110A  2726
1040 ! PG3  HP8110A  2727
1050 ! PG4  HP8110A  2728
1060 ! PG5  HP8110A  2729
1070 !
1080 ! HF1  PG12
1090 ! HF2  PG21
1100 ! HF3  PG11
```

Execution of TIMES_VTH

1. Start the HP BASIC/UX environment.

   ```
   $ rmb  Return
   ```

2. Run START.F program of the HP 4062F.

   ```
   LOAD "/usr/pcs/bin/START.F",1  Return
   ```

3. Load the TIMES_VTH program file.

   ```
   LOAD "/usr/pcs/demo/flash/TIMES_VTH"  Return
   ```

4. Make necessary customization. See the following pages.

5. Link TIS.F file by selecting `Link TIS` softkey.

6. Link PARA.F library file.

   ```
   LOADSUB ALL FROM "/usr/pcs/lib/PARA.F"  Return
   ```

7. Link SEMIGRAPH graphic library.

   ```
   LOADSUB ALL FROM "/usr/pcs/demo/flash/SEMIGRAPH"  Return
   ```

8. Execute the program by selecting `RUN` softkey.

Customization of TIMES_VTH

This section describes how to modify TIMES_VTH for cell types that are almost the same as the conventional NOR type described in Chapter 1. After loading "TIMES_VTH", go into editing mode by selecting EDIT softkey, then make modifications.

In the main program of TIMES_VTH, pins are assigned to the DUT terminals, then parameters are specified for the threshold voltage (Vth) measurements. Then, system initialization is performed, and subroutine for times dependency (Times_dep) is called.

In the Times_dep subroutine, the graphics axes are drawn, then the pulse levels and pulse timing are set. Then, Times_dep repeatedly executes the write/erase cycle at the cell. After total number of write/erase cycles reaches 1, 10, 10^2, 10^3, 10^4, 10^5, and 10^6, Vth is measured for write, then measured for erase. Vth is plotted versus the total number of write times and erase times.

Pin Assignments of the DUT

The pin assignments of the DUT are made in the following lines:

```
1040  !-- TEG pin assignment ---------------------------------
1045  Drain=12          ! Drain
1050  Gate=7            ! Gate
1055  Source=1          ! Source
1060  Sub=43            ! Substrate
1065  !
```

Modify the pin numbers according to the actual connections.

Note that the drain, gate, and source pins must be connected to *separate* HF ports. So, if these three pins are located close to each other, you may need to modify the bond wiring for package devices, or add some bypass wires on your probe card for wafer devices so that the pins can be connected to separate ports. Connect PGUs to HF ports so that following connections are realized: drain pin—PG11, source pin—PG12, and gate pin—PG21.

Threshold Voltage Measurement

Two Vth measurement subroutines are available: `Meas_vth` (which uses AFU and extracts Vth by analog feedback measurement) and `Meas_vth2` (which extracts Vth by linear search measurement). In `TIMES_VTH` program, Vth measurement is executed after the single write and single erase pulses are forced (see line 1455 or 1495). In `TIMES_VTH` program, `Meas_vth2` is used, but you can use `Meas_vth` by changing lines 1455 and 1495.

```
1445      Write_nor1(Level(*),Width(*),Delay(*),Ledge(*),Tedge(*)) ! Write
1450      !
1455      Meas_vth2(Vth_w(Index))    ! Measure Vth
            :
1485      Erase_nor1(Level(3),Width(3),Ledge(3),Tedge(3))  ! Erase
1490      !
1495      Meas_vth2(Vth_e(Index))     ! Measure Vth
```

The Vth measurement parameters are set in the following lines. You can modify these values according to the DUT characteristics.

```
1070  !-- Parameters for Vth measurement ----------------------
1075  Vgstart=0              ! Start voltage
1080  Vgstop=8               ! Stop voltage
1085  Vd=1                   ! Vd
1090  Id_targ=1.E-6          ! Target current
1095  Rmp=50                 ! Ramp rate                  (Meas_vth)
1100  Integ=5.E-3            ! Feedback integration time  (Meas_vth)
1105  Vgstep=.01             ! Vgstep                     (Meas_vth2)
1110  Igcomp=.1              ! Ig compliance              (Meas_vth2)
1115  Idcomp=1.1E-6          ! Id compliance              (Meas_vth2)
1120  Hold=0                 ! Hold                       (Meas_vth2)
1125  !
```

Parameter Name	Description	Notice
`Vgstart`	Start gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vgstop`	Stop gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vd`	Drain voltage	Necessary both for Meas_vth and Meas_vth2
`Id_targ`	Target drain current	Necessary both for Meas_vth and Meas_vth2
`Rmp`	Ramp rate of AFU	Necessary only for Meas_vth
`Integ`	Feedback integration time	Necessary only for Meas_vth
`Vgstep`	Gate step voltage	Necessary only for Meas_vth2
`Igcomp`	Gate current compliance	Necessary only for Meas_vth2
`Idcomp`	Drain current compliance	Necessary only for Meas_vth2
`Hold`	Hold time	Necessary only for Meas_vth2

Above parameters are passed to `Meas_vth` or `Meas_vth2` subroutine as COM variables. If `Meas_vth` or `Meas_vth2` is modified and some parameters are added to subroutine, add these parameters to `COM /M_vth/` declaration in lines 1030, 1575, and 1640.

These subroutines connect the gate, drain, source, and substrate pins to *separate* SMUs, and execute the Vth measurements. Modify the source program as required.

Pulse Parameters for Write/Erase

The pulse parameters for write/erase are set in the following lines. Modify the parameter settings as desired.

```
1305    Level(1)=14       !(V)      ! Gate pulse level (Write)
1310    Level(2)=7        !(V)      ! Drain pulse level (Write)
1315    Level(3)=11       !(V)      ! Source pulse level (Erase)
1320    !
1325    Width(1)=1.02E-2  !(s)      ! Gate pulse width (Write)
1330    Width(2)=1.E-2    !(s)      ! Drain pulse width (Write)
1335    Width(3)=2.E-2    !(s)      ! Source pulse width (Erase)
1340    !
1345    Ledge(1)=1.E-5    !(s)      ! Gate pulse leading edge (Write)
1350    Ledge(2)=1.E-5    !(s)      ! Drain pulse leading edge (Write)
1355    Ledge(3)=1.E-5    !(s)      ! Source pulse leading edge (Erase)
1360    !
1365    Tedge(1)=1.E-5    !(s)      ! Gate pulse trailing edge (Write)
1370    Tedge(2)=1.E-5    !(s)      ! Drain pulse trailing edge (Write)
1375    Tedge(3)=1.E-5    !(s)      ! Source pulse trailing edge (Erase)
1380    !
1385    Delay(1)=1.E-4    !(s)      ! Gate pulse delay time (Write)
1390    Delay(2)=2.E-4    !(s)      ! Drain pulse delay time (Write)
1395    Delay(3)=1.2E-2   !(s)      ! Source pulse delay time (Erase)
1400    !
```

Parameter Name	Description
`Level(1)`	Gate pulse level for write
`Level(2)`	Drain pulse level for write
`Level(3)`	Source pulse level for erase
`Width(1)`	Gate pulse width for write
`Width(2)`	Drain pulse width for write
`Width(3)`	Source pulse width for erase
`Ledge(1)`	Leading edge of gate pulse for write
`Ledge(2)`	Leading edge of drain pulse for write
`Ledge(3)`	Leading edge of source pulse for erase
`Tedge(1)`	Trailing edge of gate pulse for write
`Tedge(2)`	Trailing edge of drain pulse for write
`Tedge(3)`	Trailing edge of source pulse for erase
`Delay(1)`	Delay time of gate pulse for write
`Delay(2)`	Delay time of drain pulse for write
`Delay(3)`	Delay time of source pulse for erase

Sequence for Times Dependency Measurement

The following part of TIMES_VTH program actually performs the times dependency measurement. The program repeatedly executes the write/erase cycle at the cell. After total number of write/erase cycles reaches 1, 10, 10^2, 10^3, 10^4, 10^5, and 10^6, Vth is measured for write, then measured for erase. Vth is plotted versus the total number of write times and erase times.

The Count variable is the *total* number of write/erase cycles that will have been performed by all loops at the end of present loop. Index variable is the loop counter and is used to calculate Count value. No_pls variable is number of write/erase cycles to repeat to reach the Count value.

The repeated write/erase cycle is performed by Repeat_nor1. The single write is performed by Write_nor1, and the single erase is performed by Erase_nor1.

This sequence can be modified if desired.

```
1405    FOR Index=1 TO 6
1410      Count=10^(Index-1)        ! Calculate sum # of pulses
1415      No_pls=Count-Count_old-1  ! Calculate repetitive # of pulses
1420      !
1425      IF No_pls>0 THEN          ! Skip if # of pulse = 0
1430        Repeat_nor1(No_pls,Level(*),Width(*),Delay(*),Ledge(*),Tedge(*))
1435      END IF
1440      !
1445      Write_nor1(Level(*),Width(*),Delay(*),Ledge(*),Tedge(*)) ! Write
1450      !
1455      Meas_vth2(Vth_w(Index))    ! Measure Vth
1460      !
1465      IF Index>2 THEN
1470        Plot_x1y(Count,Count_old,Index,Clr_wrt,Vth_w(*))  ! Plot data
1475      END IF
1480      !
1485      Erase_nor1(Level(3),Width(3),Ledge(3),Tedge(3))  ! Erase
1490      !
1495      Meas_vth2(Vth_e(Index))     ! Measure Vth
1500      !
1505      IF Index>2 THEN
1510        Plot_x1y(Count,Count_old,Index,Clr_wrt,Vth_e(*))  ! Plot data
1515      END IF
1520      !
1525      Count_old=Count
1530    NEXT Index
```

Graphics Axes

The graphics axes are set by the following lines:

```
1245    X1ax_min=1.0                                !
1250    X1ax_max=100000                             ! Parameters
1255    Vth_ax_min=0                                !  for Graphics
1260    Vth_ax_max=8                                !
1265    Yax_n$="Vth (V)"                            !
1270    G_name$="  Write/Erase Pulse times Dep."     !
1275    X1ax_n$="Write/Erase pulse times"           !
1280    Clr_wrt=2    ! Pen Color for Program        !
1285    Clr_ers=4    ! Pen Color for Erase          !
1290    !
1295    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0)
```

You can modify these parameters if desired. See Appendix A.

Customization of TIMES_VTH for NAND type cell

This section describes how to modify TIMES_VTH for cell types that are similar to the NAND type described in Chapter 1. For NAND type, only two PGUs are necessary. You need to modify the connections and the program.

Port or Terminal Name	Port or terminal to be connected
PG11	HF port that can be connected to the gate
PG12	HF port that can be connected to drain, source, and substrate

The drain, source, and substrate terminals must be connected to PG12, so the DUT must be designed so that these terminals can be connected to the same HF port. If this is not possible, you can use BNC T connectors to connect PG12 to multiple HF ports.

After loading TIMES_VTH, go into editing mode by selecting EDIT softkey, then make modifications as described in this section.

If the DUT is same type as NAND cell described in Chapter 1, Write_nand1 and Erase_nand1 subprograms can be used for single writing and single erasing, and Repeat_nand1 subprogram can be used to repeat the write/erase cycle.

For NOR type cells, Write_nor1, Erase_nor1, and Repeat_nor1 subprograms are used. For NAND type cells, Write_nand1, Erase_nand1, and Repeat_nand1 subprograms are used. Be aware that the NAND subprograms (Repeat_nand1 and Write_nand1) require *fewer* parameters than the corresponding NOR subprograms.

Pin Assignments of the DUT

The pin assignments of the DUT are made after line 1040 as shown below.

```
1040  !-- TEG pin assignment ---------------------------------
1045  Drain=12          ! Drain
1050  Gate=7            ! Gate
1055  Source=1          ! Source
1060  Sub=43            ! Sub
1065  !
```

Modify the pin numbers according to the actual connections. Note that the HF port of the gate pin must *not* be the same as the HF port for the drain, source, and substrate pins. You can connect the drain, source, and substrate pins to the same HF port or to different HF ports, but make sure these pins can connect to PG12. And gate must be able to connect to PG11.

If the DUT cannot be connected properly, redesign bond wiring for package devices, or add some bypass wires on your probe card for wafer devices.

Threshold Voltage Measurement

Two Vth measurement subroutines are available: `Meas_vth` (which uses AFU and extracts Vth by analog feedback measurement) and `Meas_vth2` (which extracts Vth by linear search measurement). In `TIMES_VTH` program, Vth measurement is executed after the single write and single erase pulses are forced (see line 1455 or 1495). In `TIMES_VTH` program, `Meas_vth2` is used, but you can use `Meas_vth` by changing lines 1455 and 1495.

```
1445      Write_nor1(Level(*),Width(*),Delay(*),Ledge(*),Tedge(*)) ! Write
1450      !
1455      Meas_vth2(Vth_w(Index))    ! Measure Vth
            :
1485      Erase_nor1(Level(3),Width(3),Ledge(3),Tedge(3))  ! Erase
1490      !
1495      Meas_vth2(Vth_e(Index))     ! Measure Vth
```

The Vth measurement parameters are set in the following lines. You can modify these values according to the DUT characteristics.

```
1070  !-- Parameters for Vth measurement ----------------------
1075  Vgstart=0              ! Start voltage
1080  Vgstop=8               ! Stop voltage
1085  Vd=1                   ! Vd
1090  Id_targ=1.E-6          ! Target current
1095  Rmp=50                 ! Ramp rate                  (Meas_vth)
1100  Integ=5.E-3            ! Feedback integration time  (Meas_vth)
1105  Vgstep=.01             ! Vgstep                     (Meas_vth2)
1110  Igcomp=.1              ! Ig compliance              (Meas_vth2)
1115  Idcomp=1.1E-6          ! Id compliance              (Meas_vth2)
1120  Hold=0                 ! Hold                       (Meas_vth2)
1125  !
```

Parameter Name	Description	Notice
`Vgstart`	Start gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vgstop`	Stop gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vd`	Drain voltage	Necessary both for Meas_vth and Meas_vth2
`Id_targ`	Target drain current	Necessary both for Meas_vth and Meas_vth2
`Rmp`	Ramp rate of AFU	Necessary only for Meas_vth
`Integ`	Feedback integration time	Necessary only for Meas_vth
`Vgstep`	Gate step voltage	Necessary only for Meas_vth2
`Igcomp`	Gate current compliance	Necessary only for Meas_vth2
`Idcomp`	Drain current compliance	Necessary only for Meas_vth2
`Hold`	Hold time	Necessary only for Meas_vth2

Above parameters are passed to `Meas_vth` or `Meas_vth2` subroutine as COM variables. If `Meas_vth` or `Meas_vth2` is modified and some parameters are added to subroutine, add these parameters to `COM /M_vth/` declaration in lines 1030, 1575, and 1640.

These subroutines connect the gate, drain, source, and substrate pins to *separate* SMUs, and execute the Vth measurements. Modify the source program as required.

Pulse Parameters for Write/Erase

The pulse parameters for write/erase are set in the following lines. Modify the parameter settings for NAND type as in the following example:

```
1305    Level(1)=18       !(V)      ! Write pulse level
1310    Level(2)=18       !(V)      ! Erase pulse level
1315    !
1320    Width(1)=1.E-5    !(s)      ! Write pulse width
1325    Width(2)=1.E-5    !(s)      ! Erase pulse width
1330    !
1335    Ledge(1)=1.E-6    !(s)      ! Write pulse leading edge
1340    Ledge(2)=1.E-6    !(s)      ! Erase pulse leading edge
1345    !
1350    Tedge(1)=1.E-6    !(s)      ! Write pulse trailing edge
1355    Tedge(2)=1.E-6    !(s)      ! Erase pulse trailing edge
1360    !
```

Parameter Name	Description
`Level(1)`	Pulse level for write
`Level(2)`	Pulse level for erase
`Width(1)`	Pulse width for write
`Width(2)`	Pulse width for erase
`Ledge(1)`	Leading edge of pulse for write
`Ledge(2)`	Leading edge of pulse for erase
`Tedge(1)`	Trailing edge of pulse for write
`Tedge(2)`	Trailing edge of pulse for erase

For NAND type, you need to modify line 1210 as follows. `Delay` variable is not used by NAND type, but you do not have to delete it.

```
1210    DIM Level(2),Width(2),Delay(2),Ledge(2),Tedge(2)
```

Sequence for Times Dependency Measurement

The following part of TIMES_VTH program actually performs the times dependency measurement. The program repeatedly executes the write/erase cycle at the cell. After total number of write/erase cycles reaches 1, 10, 10^2, 10^3, 10^4, 10^5, and 10^6, Vth is measured for write, then measured for erase. Vth is plotted versus the total number of write times and erase times.

The Count variable is the *total* number of write/erase cycles that will have been performed by all loops at the end of present loop. Index variable is the loop counter and is used to calculate Count value. No_pls variable is number of write/erase cycles to repeat to reach the Count value.

The repeated write/erase cycle is performed by Repeat_nand1. The single write is performed by Write_nand1, and the single erase is performed by Erase_nand1. The following shows an example of using these subprograms in TIMES_VTH.

```
1405    FOR Index=1 TO 6
1410      Count=10^(Index-1)        ! Calculate sum # of pulses
1415      No_pls=Count-Count_old-1  ! Calculate repetitive # of pulses
1420      !
1425      IF No_pls>0 THEN          ! Skip if # of pulse = 0
1430        Repeat_nand1(No_pls,Level(*),Width(*),Delay(*),Ledge(*),Tedge(*))
1435      END IF
1440      !
1445      Write_nand1(Level(1),Width(1),Ledge(1),Tedge(1)) ! Write
1450      !
1455      Meas_vth2(Vth_w(Index))   ! Measure Vth
1460      !
1465      IF Index>2 THEN
1470         Plot_x1y(Count,Count_old,Index,Clr_wrt,Vth_w(*)) ! Plot data
1475      END IF
1480      !
1485      Erase_nand1(Level(2),Width(2),Ledge(2),Tedge(2))  ! Erase
1490      !
1495      Meas_vth2(Vthm)            ! Measure Vth
1500      !
1505      Vth_e(Index)=Vthm
1510      IF Index>2 THEN
1515         Plot_x1y(Count,Count_old,Index,Clr_wrt,Vth_e(*)) ! Plot data
1520      END IF
1525      !
1530      Count_old=Count
1532    NEXT Index
```

Graphics Axes

The graphics axes are set by the following lines:

```
1245    X1ax_min=1.0                                !
1250    X1ax_max=100000                             ! Parameters
1255    Vth_ax_min=0                                !  for Graphics
1260    Vth_ax_max=8                                !
1265    Yax_n$="Vth (V)"                            !
1270    G_name$="  Write/Erase Pulse times Dep."     !
1275    X1ax_n$="Write/Erase pulse times"           !
1280    Clr_wrt=2    ! Pen Color for Program        !
1285    Clr_ers=4    ! Pen Color for Erase          !
1290    !
1295    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0)
```

You can modify these parameters if desired. See Appendix A.

Customization of TIMES_VTH for Other Cell Types

This section describes how to modify TIMES_VTH for cell types that are not conventional NOR type or NAND type.

Creation of Write and Erase Subprograms

The PARA.F library has the following subprograms for writing and erasing flash memory cells:

Cell Type	Write	Erase	Repeated Write/Erase
Conventional NOR type	Write_nor1	Erase_nor1	Repeat_nor1
3.3V NOR type	Write_nor2, Write_nor3[1]	Erase_nor2, Erase_nor3[1]	Repeat_nor2, Repeat_nor3[1]
NAND type	Write_nand1	Erase_nand1	Repeat_nand1

1 The Write_nor3, Erase_nor3, and Repeat_nor3 subprograms use the multiplexer switch on the cable adapter.

For the 3.3V NOR type cell described in Chapter 6, you use Write_nor2, Erase_nor2, and Repeat_nor2, or Write_nor3, Erase_nor3, and Repeat_nor3.

For other cell types, you must create new subprograms for writing and erasing.

Before creating new subprograms, you must consider the pin locations of the cell. The TEG must be designed so that cell terminals that require different waveforms can connect to separate HF ports, and cell terminals that require the same pulse waveform can connect to the same HF port.

If the TEG has already been designed, and the correct connections cannot be made, you need to change the bond wiring for package devices, or use bypass wiring on the probe card for wafer devices.

The above 12 programs consist of TIS subprograms, so you can easily refer to the source program code and modify the programs as desired. You can easily create single write, single erase, and repeating write/erase programs for your desired device, which has the pin settings assigned by Set_fr subprogram.

Setup

Connect PGUs to HF ports according to the subprograms that you create.

Pin Assignments of the DUT

The pin assignments of the DUT are made after line 1040. Modify these assignments according to the DUT pin locations.

Threshold Voltage Measurement

Two Vth measurement subroutines are available: `Meas_vth` (which uses AFU and extracts Vth by analog feedback measurement) and `Meas_vth2` (which extracts Vth by linear search measurement). In `TIMES_VTH` program, Vth measurement is executed after the single write and single erase pulses are forced (see line 1455 or 1495). In `TIMES_VTH` program, `Meas_vth2` is used, but you can use `Meas_vth` by changing lines 1455 and 1495.

The Vth measurement parameters are set in the following lines. You can modify these values according to the DUT characteristics.

Parameter Name	Description	Notice
`Vgstart`	Start gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vgstop`	Stop gate voltage	Necessary both for Meas_vth and Meas_vth2
`Vd`	Drain voltage	Necessary both for Meas_vth and Meas_vth2
`Id_targ`	Target drain current	Necessary both for Meas_vth and Meas_vth2
`Rmp`	Ramp rate of AFU	Necessary only for Meas_vth
`Integ`	Feedback integration time	Necessary only for Meas_vth
`Vgstep`	Gate step voltage	Necessary only for Meas_vth2
`Igcomp`	Gate current compliance	Necessary only for Meas_vth2
`Idcomp`	Drain current compliance	Necessary only for Meas_vth2
`Hold`	Hold time	Necessary only for Meas_vth2

Above parameters are passed to `Meas_vth` or `Meas_vth2` subroutine as COM variables. If `Meas_vth` or `Meas_vth2` is modified and some parameters are added to subroutine, add these parameters to `COM /M_vth/` declaration in lines 1030, 1575, and 1640.

These subroutines connect the gate, drain, source, and substrate pins to *separate* SMUs, and execute the Vth measurements. Modify the source program as required.

Pulse Parameters for Write/Erase

The pulse parameters for write/erase are set from line 1305. Modify the parameter settings according to your DUT type.

Sequence for Times Dependency Measurement

Lines 1405 to 1530 of `TIMES_VTH` program actually perform the times dependency measurement. The program repeatedly executes the write/erase cycle at the cell. After total number of write/erase cycles reaches 1, 10, 10^2, 10^3, 10^4, 10^5, and 10^6, Vth is measured for write, then measured for erase. Vth is plotted versus the total number of write times and erase times.

The `Count` variable is the *total* number of write/erase cycles that will have been performed by all loops at the end of present loop. `Index` variable is the loop counter and is used to calculate `Count` value. `No_pls` variable is number of write/erase cycles to repeat to reach the `Count` value.

Modify the program according to the DUT type.

Graphics Axes

The graphics axes are set by the following lines:

```
1245    X1ax_min=1.0                                !
1250    X1ax_max=100000                             ! Parameters
1255    Vth_ax_min=0                                !  for Graphics
1260    Vth_ax_max=8                                !
1265    Yax_n$="Vth (V)"                            !
1270    G_name$="  Write/Erase Pulse times Dep."     !
1275    X1ax_n$="Write/Erase pulse times"           !
1280    Clr_wrt=2    ! Pen Color for Program        !
1285    Clr_ers=4    ! Pen Color for Erase          !
1290    !
1295    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0)
```

You can modify these parameters if desired. See Appendix A.

A

Graphic Subprograms

The following three graphics subprograms are used in the application programs. These are stored in the `SEMIGRAPH` library file, which is in `/usr/pcs/demo/flash/` directory. Use LOADSUB command to link these graphics commands.

```
LOADSUB ALL FROM "SEMIGRAPH"  Return
```

- Semiloggraph1

  This subprogram draws and labels the graphic axes. The X axis is logarithmic and the Y axis is linear. Two X axes, lower (X1) and upper (X2), can be set independently.

- Plot_x1y

  This subprogram draws line between two specified points. This is for the X1 axis (lower X axis).

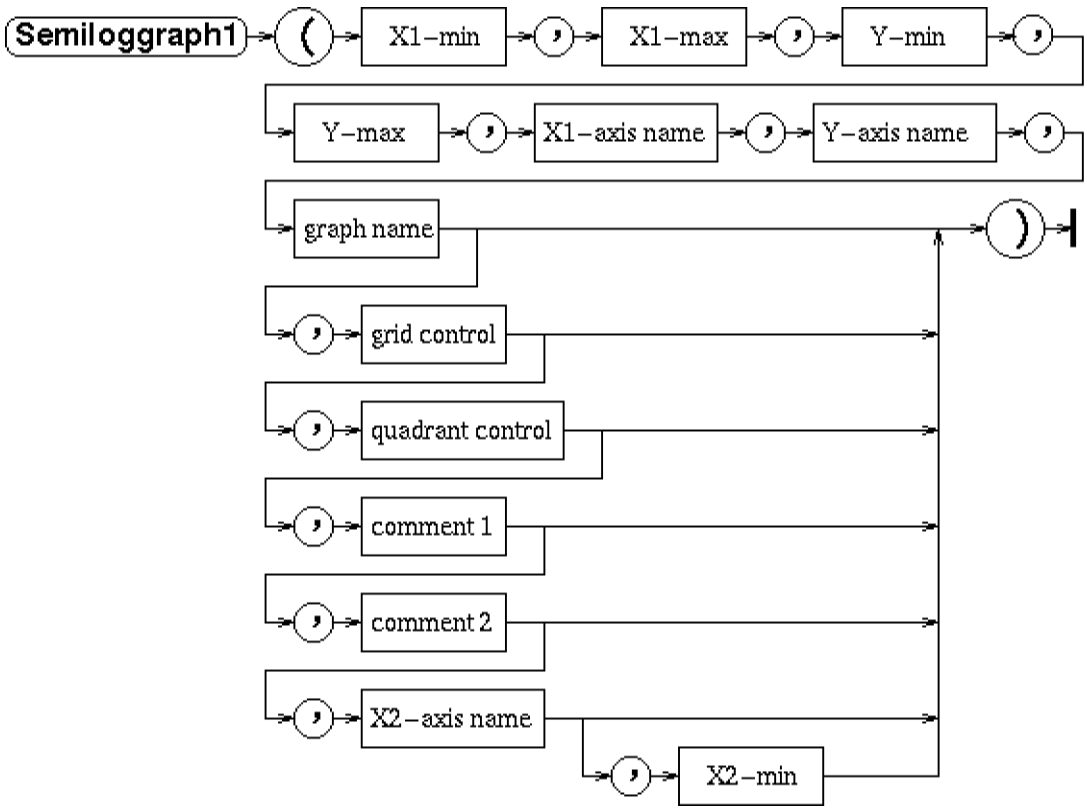
- Plot_x2y

  This subprogram draws line between two specified points. This is for the X2 axis (upper X axis).

Semiloggraph1

This subprogram draws and labels the graphic axes. The X axis is logarithmic and the Y axis is linear. Two X axes, lower (X1) and upper (X2), can be set independently.

Syntax



Item	Description/Default	Range
X1-min	numeric expression	—
X1-max	numeric expression	—
Y-min	numeric expression	—
Y-max	numeric expression	—
X1-axis name	string expression	Maximum 30 characters
Y-axis name	string expression	Maximum 30 characters
graph name	string expression	Maximum 30 characters
grid control	integer expression. default= 0	0 to 2
quadrant control	integer expression. default= 0	0 to 4
comment1	string expression	Maximum 20 characters
comment2	string expression	Maximum 20 characters
X2-axis name	string expression	Maximum 30 characters
X2-min	numeric expression	—

Example Statement

```
Semiloggraph1(1E-5,1E-1,0,10,"Pulse width (Erase)","Vth",
              "Pulse Width Dependency",1,0,"LOT1","DEVICE1",
              "Pulse width (Write)",1E-7)
```

Description

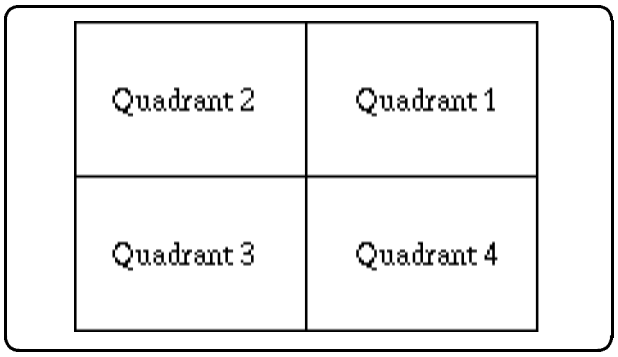
This subprogram clears CRT of all graphics, then draws and labels X-axes and Y-axis as specified.

The maximum for X2 axis is automatically determined by the number of decades from *X1-min* to *X1-max*.

grid control and *quadrant control* are specified as follows:

grid control
0: no grid line
1: broken line
2: solid line

quadrant control
0: full screen
1: upper right area
2: upper left area
3: lower left area
4: lower right area

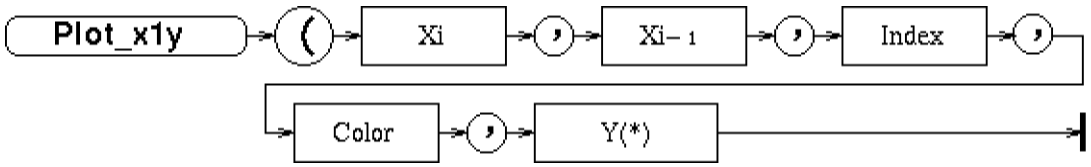


CRT Display

Plot_x1y

This subprogram draws a line between two specified points. This is for the X1 axis (lower X axis).

Syntax



Item	Description	Range
X_i	Value of X_i, numeric expression	$X_i > 0$
X_{i-1}	Value of X_{i-1}, numeric expression	$X_{i-1} > 0$
Index	Value of i, integer expression	—
Color	Pen color, integer expression	Refer to the HP BASIC manual
$Y(*)$	Array includes Y_i, Y_{i-1}	—

Example Statement

```
Plot_x1y(1E-5,1E-4,2,1,Vth_e(*))
Plot_x1y(Count,Count_old,Index,Clr,Y(*))
```

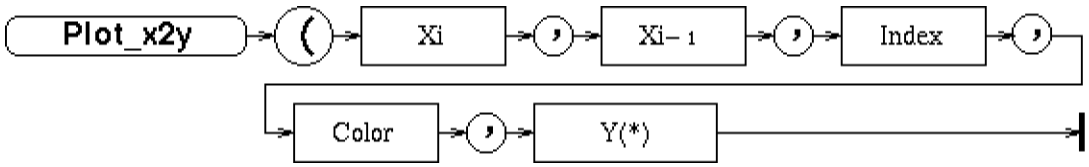
Description

This subprogram draws a line between (X_i, Y_i) and (X_{i-1}, Y_{i-1}). This is for the X1 axis (lower X axis).

Plot_x2y

This subprogram draws a line between two specified points. This is for the X2 axis (upper X axis).

Syntax



Item	Description	Range
X_i	Value of X_i, numeric expression	$X_i > 0$
X_{i-1}	Value of X_{i-1}, numeric expression	$X_{i-1} > 0$
Index	Value of i, integer expression	—
Color	Pen color, integer expression	Refer to the HP BASIC manual
$Y(*)$	Array includes Y_i, Y_{i-1}	—

Example Statement

```
Plot_x2y(1E-7,1E-6,2,1,Vth_w(*))
Plot_x2y(Count,Count_old,Index,Clr,Y(*))
```

Description

This subprogram draws a line between (X_i, Y_i) and (X_{i-1}, Y_{i-1}). This is for the X2 axis (upper X axis).

B

Program List of Application Program

List of ACC_PWD_VTH

```
1000 ! "@(#) PROGRAM ACC_PWD_VTH                    Rev. F.06.30    (), , "
1005 !                           Copyright 1983-1993 Hewlett-Packard Company.
1010  !************************************************************************
1015  OPTION BASE 1
1020  !
1025  COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1030  COM /M_vth/ Vgstart,Vgstop,Vd,Id_targ,Rmp,Integ,Igcomp,Idcomp,Hold,Vgstep
1035  !
1040  !-- TEG pin assignment ---------------------------------
1045  Drain=12         ! Drain
1050  Gate=7           ! Gate
1055  Source=1         ! Source
1060  Sub=43           ! Sub
1065  !
1070  !-- Parameters for Vth measurement ----------------------
1075  Vgstart=0              ! Start voltage
1080  Vgstop=8               ! Stop voltage
1085  Vd=.1                  ! Vd
1090  Id_targ=1.E-6          ! Target current
1095  Rmp=50                 ! Ramp rate                  (Meas_vth)
1100  Integ=5.E-3            ! Feedback integration time  (Meas_vth)
1105  Vgstep=.01             ! Vgstep                     (Meas_vth2)
1110  Igcomp=.1              ! Ig compliance              (Meas_vth2)
1115  Idcomp=1.1E-6          ! Id compliance              (Meas_vth2)
1120  Hold=0                 ! Hold                       (Meas_vth2)
1125  !
1130  Init_system                  ! Initialize 4062F
1135  Init_pg                      ! Initialize 8110As
1140  Set_fr(Source,Gate,Drain,Sub) ! Assign pins
1145  Acc_pwd_dep                  ! Measure Accumulated Pulse Width Dependency
1150  !
1155  END
1160  !
1165 Acc_pwd_dep:SUB Acc_pwd_dep
1170    OPTION BASE 1
1175    !***********************************************************************
1180    ! This is a main algorithm for accumulated write/erase  pulse
1185    ! width dependency
1190    !***********************************************************************
1195    COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1200    INTEGER Loop1,Loop2,Clr_wrt,Clr_ers,Num_of_data_w,Num_of_data_e
1205    REAL Vth_axis_min,Vth_axis_max,Vth_w(100),Vth_e(100)
1210    DIM X1ax_n$[30],Yax_n$[30],X2ax_n$[30],G_name$[35]
1215    DIM Level_w(2),Width_w(2),Delay_w(2),Ledge_w(2),Tedge_w(2)
1220    !
1225    Loop1=0   ! Loop1 is an index regarding write pulse width
1230    Loop2=0   ! Loop2 is an index regarding erase pulse width
1235    !
1240    Num_of_data_w=0 ! Number of data being measured (Write)
1245    Num_of_data_e=0 ! Number of data being measured (Erase)
1250    Total_width_w=0 ! Accumulated pulse width (Write)
1255    Total_width_e=0 ! Accumulated pulse width (Erase)
1260    !
```

```
1265    X1ax_min=1.00E-7                                !
1270    X1ax_max=1.E+0                                  !
1275    X2ax_min=1.00E-7                                ! Parameters
1280                                                    !  for Graph axis
1285    Vth_ax_min=0                                    !
1290    Vth_ax_max=8                                    !
1295    Yax_n$="Vth (V)"                                !
1300    G_name$="Accumulated Pulse Width Dep."          !
1305    X1ax_n$="Erase pulse width (s)"                 !
1310    X2ax_n$="Write pulse width (s)"                 !
1315    Clr_wrt=2    ! Pen Color for Write              !
1320    Clr_ers=4    ! Pen Color for Erase              !
1325    !
1330    !
1335    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0,"","",
X2ax_n$,X2ax_min)
1340    !
1345    Level_w(1)=14      !(V)      ! Gate pulse level (Write)
1350    Level_w(2)=7       !(V)      ! Drain pulse level (Write)
1355    Level_e=11         !(V)      ! Source pulse level (Erase)
1360    Ledge_w(1)=5.E-8   !(s)      ! Gate leading edge (Write)
1365    Ledge_w(2)=5.E-8   !(s)      ! Drain leading edge (Write)
1370    Ledge_e=5.E-8      !(s)      ! Source leading edge (Erase)
1375    Tedge_w(1)=5.E-8   !(s)      ! Gate trailing edge (Write)
1380    Tedge_w(2)=5.E-8   !(s)      ! Drain trailing edge (Write)
1385    Tedge_e=5.E-8      !(s)      ! Source trailing edge (Erase)
1390    !
1395    !--- Accumulated write pulse width vs. Vth --------------------
1400    FOR Loop1=-7 TO -1
1405      Width_w(2)=10^Loop1-Total_width_w   ! Drain pulse width
1410      Width_w(1)=Width_w(2)*1.1           ! Gate pulse width
1415      Delay_w(2)=Width_w(2)*.05           ! Drain pulse delay time
1420      Delay_w(1)=0                        ! Gate pulse delay time
1425      Total_width_w=10^Loop1              ! Accumulated pulse width
1430      GOSUB Write_meas_plot               ! Apply write pulse and meas Vth
1435      !
1440      Width_w(2)=2*10^Loop1-Total_width_w ! Drain pulse width
1445      Width_w(1)=Width_w(2)*1.1           ! Gate pulse width
1450      Delay_w(2)=Width_w(2)*.05           ! Drain pulse delay time
1455      Delay_w(1)=0                        ! Gate pulse delay time
1460      Total_width_w=2*10^Loop1            ! Accumulated pulse width
1465      GOSUB Write_meas_plot               ! Apply write pulse and meas Vth
1470      !
1475      Width_w(2)=5*10^Loop1-Total_width_w ! Drain pulse width
1480      Width_w(1)=Width_w(2)*1.1           ! Gate pulse width
1485      Delay_w(2)=Width_w(2)*.05           ! Drain pulse delay time
1490      Delay_w(1)=0                        ! Gate pulse delay time
1495      Total_width_w=5*10^Loop1            ! Accumulated pulse width
1500      GOSUB Write_meas_plot               ! Apply write pulse and meas Vth
1505    NEXT Loop1
1510    !
1515    !--- Accumulated erase pulse width vs. Vth --------------------
1520    FOR Loop2=-7 TO -1
1525      Width_e=10^Loop2-Total_width_e    ! Source pulse width
1530      Delay_e=0                         ! Source pulse delay time
1535      Total_width_e=10^Loop2            ! Accumulated pulse width
1540      GOSUB Erase_meas_plot             ! Apply erase pulse and meas Vth
1545      !
1550      Width_e=2*10^Loop2-Total_width_e  ! Source pulse width
1555      Delay_e=0                         ! Source pulse delay time
1560      Total_width_e=2*10^Loop2          ! Accumulated pulse width
1565      GOSUB Erase_meas_plot             ! Apply erase pulse and meas Vth
1570      !
1575      Width_e=5*10^Loop2-Total_width_e  ! Source pulse width
1580      Delay_e=0                         ! Source pulse delay time
1585      Total_width_e=5*10^Loop2          ! Accumulated pulse width
```

B-2 Program List of Application Program

```
1590      GOSUB Erase_meas_plot             ! Apply erase pulse and meas Vth
1595    NEXT Loop2
1600    !
1605    SUBEXIT
1610    !
1615 Write_meas_plot:!
1620    Num_of_data_w=Num_of_data_w+1
1625    Write_nor1(Level_w(*),Width_w(*),Delay_w(*),Ledge_w(*),Tedge_w(*))
1630    Meas_vth2(Vth_w(Num_of_data_w))
1635    IF Num_of_data_w>1 THEN
1640      Plot_x1y(Total_width_w,Total_width_w-Width_w(2),Num_of_data_w,Clr_wrt,Vth_w(*))
1645    END IF
1650    RETURN
1655    !
1660 Erase_meas_plot:!
1665    Num_of_data_e=Num_of_data_e+1
1670    Erase_nor1(Level_e,Width_e,Ledge_e,Tedge_e)
1675    Meas_vth2(Vth_e(Num_of_data_e))
1680    IF Num_of_data_e>1 THEN
1685      Plot_x1y(Total_width_e,Total_width_e-Width_e,Num_of_data_e,Clr_ers,Vth_e(*))
1690    END IF
1695    RETURN
1700    !
1705  SUBEND
1710  !
1715 Meas_vth:SUB Meas_vth(Vth)
1720  !*********************************************************************
1725  ! This subprogram measures Vth using analog search
1730  !*********************************************************************
1735    OPTION BASE 1
1740    COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1745    COM /M_vth/ Vgstart,Vgstop,Vd,Id_targ,Rmp,Integ,Igcomp,Idcomp,Hold,Vgstep
1750    INTEGER St
1755  !
1760    Set_tr(Source,Gate,Drain,Sub)
1765    Vth=FNVth5(Id_targ,Vd,Vgstart,Vgstop,Rmp,Integ,0)
1770  !
1775  SUBEND
1780  !
1785 Meas_vth2:SUB Meas_vth2(Vth)
1790  !*********************************************************************
1795  ! This subprogram measures Vth using linear search
1800  !*********************************************************************
1805    OPTION BASE 1
1810    COM /M_vth/ Vgstart,Vgstop,Vd,Id_targ,Rmp,Integ,Igcomp,Idcomp,Hold,Vgstep
1815    COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1820    INTEGER St
1825  !
1830    Connect(FNSmu(1),Drain)
1835    Connect(FNSmu(2),Gate)
1840    Connect(FNSmu(3),Source)
1845    Connect(FNSmu(4),Sub)
1850  !
1855    Force_v(Source,0,20)           ! Apply zero voltage to Source
1860    Force_v(Sub,0,20)              ! Apply zero voltage to Substrate
1865    Force_v(Drain,Vd,2,Idcomp)     ! Apply Vd to Drain voltage
1870  !
1875    Set_lsearch(Gate,Drain,2,Vgstart,Vgstop,Vgstep,Id_targ,Idcomp,Igcomp,Hold)
1880    Search(Vth,St,Idm)
1885  !
1890    Disable_port(Drain,Gate,Source,Sub)
1895    Connect(0,Drain,Gate,Source,Sub)
1900  SUBEND
```

List of PWD_VTH

```
1000 ! "@(#) PROGRAM PWD_VTH                    Rev. F.06.30    (), , "
1005 !                           Copyright 1983-1993 Hewlett-Packard Company.
1010  !************************************************************************
1015  OPTION BASE 1
1020  !
1025  COM /Pct_write/ Level_w_p(2),Width_w_p(2),Delay_w_p(2),Ledge_w_p(2),Tedge_w_p(2)
1030  COM /Pct_erase/ Level_e_p,Width_e_p,Ledge_e_p,Tedge_e_p
1035  COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1040  COM /M_vth/ Vgstart,Vgstop,Vd,Id_targ,Rmp,Integ,Igcomp,Idcomp,Hold,Vgstep
1045  !
1050  !-- TEG pin assignment ---------------------------------
1055  Drain=12         ! Drain
1060  Gate=7           ! Gate
1065  Source=1         ! Source
1070  Sub=43           ! Sub
1075  !
1080  !-- Parameters for Perfect Write ------------------------
1085  Level_w_p(1)=14!(V)      ! Gate pulse level
1090  Level_w_p(2)=7!(V)       ! Drain pulse level
1095  Width_w_p(1)=1.02E-1!(s) ! Gate pulse width
1100  Width_w_p(2)=1.E-1!(s)   ! Drain pulse width
1105  Delay_w_p(1)=1.E-3!(s)   ! Gate pulse delay time
1110  Delay_w_p(2)=2.E-3!(s)   ! Drain pulse delay time
1115  Ledge_w_p(1)=1.E-4!(s)   ! Gate pulse leading edge
1120  Ledge_w_p(2)=1.E-4!(s)   ! Drain pulse leading edge
1125  Tedge_w_p(1)=1.E-4!(s)   ! Gate pulse trailing edge
1130  Tedge_w_p(2)=1.E-4!(s)   ! Drain pulse trailing edge
1135  !
1140  !-- Parameters for Perfect Erase ------------------------
1145  Level_e_p=11!(V)       ! Source pulse level
1150  Width_e_p=.9  !(s)      ! Source pulse width
1155  Ledge_e_p=1.E-4!(s)    ! Source pulse leading edge
1160  Tedge_e_p=1.E-4!(s)    ! Source pulse trailing edge
1165  !
1170  !-- Parameters for Vth measurement ----------------------
1175  Vgstart=0              ! Start voltage
1180  Vgstop=8               ! Stop voltage
1185  Vd=.1                  ! Vd
1190  Id_targ=1.E-6          ! Target current
1195  Rmp=50                 ! Ramp rate                  (Meas_vth)
1200  Integ=5.E-3            ! Feedback integration time  (Meas_vth)
1205  Vgstep=.01             ! Vgstep                     (Meas_vth2)
1210  Igcomp=.1              ! Ig compliance              (Meas_vth2)
1215  Idcomp=1.1E-6          ! Id compliance              (Meas_vth2)
1220  Hold=0                 ! Hold                       (Meas_vth2)
1225  !
1230  Init_system                  ! Initialize 4062F
1235  Init_pg                      ! Initialize 8110As
1240  Set_fr(Source,Gate,Drain,Sub) ! Assign pins
1245  Pwd_dep                      ! Measure Pulse Width Dependency
1250  !
1255  END
1260  !
1265 Pwd_dep:SUB Pwd_dep
1270    OPTION BASE 1
1275    !*********************************************************************
1280    ! This is a main algorithm for program/erase pulse width dependency
1285    !*********************************************************************
1290    COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1295    COM /Pct_write/ Level_w_p(2),Width_w_p(2),Delay_w_p(2),Ledge_w_p(2),Tedge_w_p(2)
1300    COM /Pct_erase/ Level_e_p,Width_e_p,Ledge_e_p,Tedge_e_p
1305    INTEGER Loop1,Loop2,Clr_wrt,Clr_ers
1310    REAL Vth_axis_min,Vth_axis_max,Vth_w(100),Vth_e(100),Pwdw_old,Pwde_old
```

```
1315    DIM X1ax_n$[30],Yax_n$[30],X2ax_n$[30],G_name$[35]
1320    DIM Level_w(2),Width_w(2),Delay_w(2),Ledge_w(2),Tedge_w(2)
1325    !
1330    Loop1=0   ! Loop1 is an index regarding Vg or Vs value
1335    Loop2=0   ! Loop2 is an index regarding pulse width
1340    !
1345    X1ax_min=1.00E-4                              !
1350    X1ax_max=1.E-1                                !
1355    X2ax_min=1.00E-6                              ! Parameters
1360                                                  !  for Graph axis
1365    Vth_ax_min=0                                  !
1370    Vth_ax_max=8                                  !
1375    Yax_n$="Vth (V)"                              !
1380    G_name$="  Write/Erase Pulse Width Dep."      !
1385    X1ax_n$="Erase pulse width (s)"               !
1390    X2ax_n$="Write pulse width (s)"               !
1395    Clr_wrt=2    ! Pen Color for Write            !
1400    Clr_ers=4    ! Pen Color for Erase            !
1405    !
1410    Width_w_old=0
1415    Width_e_old=0
1420    !
1425    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0,"","",
X2ax_n$,X2ax_min)
1430    !
1435    FOR Loop1=1 TO 2
1440      SELECT Loop1
1445      CASE 1
1450        Level_w(1)=14  !(V)      ! Gate pulse level (Write)
1455        Level_w(2)=7   !(V)      ! Drain pulse level (Write)
1460        Level_e=11     !(V)      ! Source pulse level (Erase)
1465      CASE 2
1470        Level_w(1)=13  !(V)      ! Gate pulse level (Write)
1475        Level_w(2)=6.5 !(V)      ! Drain pulse level (Write)
1480        Level_e=10.5   !(V)      ! Source pulse level (Erase)
1485      END SELECT
1490      !
1495      FOR Loop2=1 TO 4
1500        SELECT Loop2
1505        CASE 1
1510          Width_w(1)=1.2E-6 !(s)   ! Gate pulse width        !
1515          Width_w(2)=1.0E-6 !(s)   ! Drain pulse width       ! Write
1520          Delay_w(1)=1.E-7 !(s)    ! Delay of Gate pulse     !  Parameters
1525          Delay_w(2)=2.E-7 !(s)    ! Delay of Drain pulse    !
1530          Ledge_w(1)=1.E-7 !(s)    ! Gate leading edge       !
1535          Ledge_w(2)=1.E-7 !(s)    ! Drain leading edge      !
1540          Tedge_w(1)=1.E-7 !(s)    ! Gate trailing edge      !
1545          Tedge_w(2)=1.E-7 !(s)    ! Drain trailing edge     !
1550          !
1555          Width_e=1.0E-4   !(s)    ! Source erase pulse width!
1560          Ledge_e=1.E-6    !(s)    ! Source leading edge     ! Erase
1565          Tedge_e=1.E-6    !(s)    ! Source trailing edge    !  Parameters
1570          !
1575        CASE 2
1580          Width_w(1)=1.2E-5 !(s)   ! Gate pulse width        !
1585          Width_w(2)=1.0E-5 !(s)   ! Drain pulse width       ! Write
1590          Delay_w(1)=1.E-6 !(s)    ! Delay of Gate pulse     !  Parameters
1595          Delay_w(2)=2.E-6 !(s)    ! Delay of Drain pulse    !
1600          Ledge_w(1)=1.E-6 !(s)    ! Gate leading edge       !
1605          Ledge_w(2)=1.E-6 !(s)    ! Drain leading edge      !
1610          Tedge_w(1)=1.E-6 !(s)    ! Gate trailing edge      !
1615          Tedge_w(2)=1.E-6 !(s)    ! Drain trailing edge     !
1620          !
1625          Width_e=1.0E-3   !(s)    ! Source erase pulse width!
1630          Ledge_e=1.E-6    !(s)    ! Source leading edge     ! Erase
1635          Tedge_e=1.E-6    !(s)    ! Source trailing edge    !  Parameters
```

```
1640          !
1645        CASE 3
1650          Width_w(1)=1.2E-4 !(s)   ! Gate pulse width        !
1655          Width_w(2)=1.0E-4 !(s)   ! Drain pulse width       ! Write
1660          Delay_w(1)=1.E-5 !(s)    ! Delay of Gate pulse     !  Parameters
1665          Delay_w(2)=2.E-5 !(s)    ! Delay of Drain pulse    !
1670          Ledge_w(1)=1.E-6 !(s)    ! Gate leading edge       !
1675          Ledge_w(2)=1.E-6 !(s)    ! Drain leading edge      !
1680          Tedge_w(1)=1.E-6 !(s)    ! Gate trailing edge      !
1685          Tedge_w(2)=1.E-6 !(s)    ! Drain trailing edge     !
1690          !
1695          Width_e=1.0E-2   !(s)    ! Source erase pulse width!
1700          Ledge_e=1.E-5    !(s)    ! Source leading edge     ! Erase
1705          Tedge_e=1.E-5    !(s)    ! Source trailing edge    !  Parameters
1710          !
1715        CASE 4
1720          Width_w(1)=1.2E-3 !(s)   ! Gate pulse width        !
1725          Width_w(2)=1.0E-3 !(s)   ! Drain pulse width       ! Write
1730          Delay_w(1)=1.E-4 !(s)    ! Delay of Gate pulse     !  Parameters
1735          Delay_w(2)=2.E-4 !(s)    ! Delay of Drain pulse    !
1740          Ledge_w(1)=1.E-6 !(s)    ! Gate leading edge       !
1745          Ledge_w(2)=1.E-6 !(s)    ! Drain leading edge      !
1750          Tedge_w(1)=1.E-6 !(s)    ! Gate trailing edge      !
1755          Tedge_w(2)=1.E-6 !(s)    ! Drain trailing edge     !
1760          !
1765          Width_e=1.0E-1   !(s)    ! Source erase pulse width!
1770          Ledge_e=1.E-5    !(s)    ! Source leading edge     ! Erase
1775          Tedge_e=1.E-5    !(s)    ! Source trailing edge    !  Parameters
1780          !
1785        END SELECT
1790        !
1795        !-- Write the cell completely -----------------
1800        Write_nor1(Level_w_p(*),Width_w_p(*),Delay_w_p(*),Ledge_w_p(*),Tedge_w_p(*))
1805        !-- Erase the cell ----------------------------
1810        Erase_nor1(Level_e,Width_e,Ledge_e,Tedge_e)
1815        Meas_vth2(Vth_e(Loop2))
1820        IF Loop2>1 THEN
1825          Plot_x1y(Width_e,Width_e_old,Loop2,Clr_ers,Vth_e(*))
1830        END IF
1835        !
1840        !-- Erase the cell completely -----------------
1845        Erase_nor1(Level_e_p,Width_e_p,Ledge_e_p,Tedge_e_p)
1850        !
1855        !-- Write the cell ----------------------------
1860        Write_nor1(Level_w(*),Width_w(*),Delay_w(*),Ledge_w(*),Tedge_w(*))
1865        Meas_vth2(Vth_w(Loop2))
1870        IF Loop2>1 THEN
1875           Plot_x2y(Width_w(2),Width_w_old,Loop2,Clr_wrt,Vth_w(*))
1880        END IF
1885        !
1890        Width_w_old=Width_w(2)
1895        Width_e_old=Width_e
1900      NEXT Loop2
1905    NEXT Loop1
1910  SUBEND
1915  !
1920 Meas_vth:SUB Meas_vth(Vth)
1925  !*********************************************************************
1930  ! This subprogram measures Vth using analog search
1935  !*********************************************************************
1940    OPTION BASE 1
1945    COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1950    COM /M_vth/ Vgstart,Vgstop,Vd,Id_targ,Rmp,Integ,Igcomp,Idcomp,Hold,Vgstep
1955    INTEGER St
1960  !
1965    Set_tr(Source,Gate,Drain,Sub)
```

```
1970    Vth=FNVth5(Id_targ,Vd,Vgstart,Vgstop,Rmp,Integ,0)
1975  !
1980  SUBEND
1985  !
1990 Meas_vth2:SUB Meas_vth2(Vth)
1995  !*****************************************************************
2000  ! This subprogram measures Vth using linear search
2005  !*****************************************************************
2010    OPTION BASE 1
2015    COM /M_vth/ Vgstart,Vgstop,Vd,Id_targ,Rmp,Integ,Igcomp,Idcomp,Hold,Vgstep
2020    COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
2025    INTEGER St
2030  !
2035    Connect(FNSmu(1),Drain)
2040    Connect(FNSmu(2),Gate)
2045    Connect(FNSmu(3),Source)
2050    Connect(FNSmu(4),Sub)
2055  !
2060    Force_v(Source,0,20)           ! Apply zero voltage to Source
2065    Force_v(Sub,0,20)              ! Apply zero voltage to Substrate
2070    Force_v(Drain,Vd,2,Idcomp)     ! Apply Vd to Drain voltage
2075  !
2080    Set_lsearch(Gate,Drain,2,Vgstart,Vgstop,Vgstep,Id_targ,Idcomp,Igcomp,Hold)
2085    Search(Vth,St,Idm)
2090  !
2095    Disable_port(Drain,Gate,Source,Sub)
2100    Connect(0,Drain,Gate,Source,Sub)
2105  SUBEND
```

List of TIMES_VTH

```
1000 ! "@(#) PROGRAM TIMES_VTH                     Rev. F.06.30    (), , "
1005 !                            Copyright 1983-1993 Hewlett-Packard Company.
1010  !************************************************************************
1015  OPTION BASE 1
1020  !
1025  COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1030  COM /M_vth/ Vgstart,Vgstop,Vd,Id_targ,Rmp,Integ,Igcomp,Idcomp,Hold,Vgstep
1035  !
1040  !-- TEG pin assignment ---------------------------------
1045  Drain=12         ! Drain
1050  Gate=7           ! Gate
1055  Source=1         ! Source
1060  Sub=43           ! Sub
1065  !
1070  !-- Parameters for Vth measurement ----------------------
1075  Vgstart=0              ! Start voltage
1080  Vgstop=8               ! Stop voltage
1085  Vd=1                   ! Vd
1090  Id_targ=1.E-6          ! Target current
1095  Rmp=50                 ! Ramp rate                  (Meas_vth)
1100  Integ=5.E-3            ! Feedback integration time  (Meas_vth)
1105  Vgstep=.01             ! Vgstep                     (Meas_vth2)
1110  Igcomp=.1              ! Ig compliance              (Meas_vth2)
1115  Idcomp=1.1E-6          ! Id compliance              (Meas_vth2)
1120  Hold=0                 ! Hold                       (Meas_vth2)
1125  !
1130  Init_system                  ! Initialize 4062F
1135  Init_pg                      ! Initialize 8110As
1140  Set_fr(Source,Gate,Drain,Sub) ! Assign pins
1145  Set_relay_mode(1)            ! Use semiconductor relay
1150  Times_dep                    ! Measure Times Dependency
1155  !
1160  END
1165 Times_dep:SUB Times_dep
1170    OPTION BASE 1
1175    !*********************************************************************
1180    ! This is a main algorithm for program/erase times dependency
1185    !*********************************************************************
1190    COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1195    INTEGER Clr_wrt,Clr_ers,Index
1200    REAL Vth_axis_min,Vth_axis_max,Vth_w(100),Vth_e(100),Count,Count_old
1205    DIM X1ax_n$[30],Yax_n$[30],X2ax_n$[30],G_name$[35]
1210    DIM Level(3),Width(3),Delay(3),Ledge(3),Tedge(3)
1215    !
1220    Count=0         ! Sam of # of pulses applied
1225    Count_old=0     ! Sam of # of pulses applied till last Vth measurement
1230    Index=0         !
1235    No_pls=0        ! # of pulses to be applied
1240    !
1245    X1ax_min=1.0                                !
1250    X1ax_max=100000                             ! Parameters
1255    Vth_ax_min=0                                !  for Graphics
1260    Vth_ax_max=8                                !
1265    Yax_n$="Vth (V)"                            !
1270    G_name$="  Write/Erase Pulse times Dep."    !
1275    X1ax_n$="Write/Erase pulse times"           !
1280    Clr_wrt=2    ! Pen Color for Program        !
1285    Clr_ers=4    ! Pen Color for Erase          !
1290    !
1295    Semiloggraph1(X1ax_min,X1ax_max,Vth_ax_min,Vth_ax_max,X1ax_n$,Yax_n$,G_name$,2,0)
1300    !
1305    Level(1)=14       !(V)      ! Gate pulse level (Write)
1310    Level(2)=7        !(V)      ! Drain pulse level (Write)
```

```
1315    Level(3)=11       !(V)      ! Source pulse level (Erase)
1320    !
1325    Width(1)=1.02E-2  !(s)      ! Gate pulse width (Write)
1330    Width(2)=1.E-2    !(s)      ! Drain pulse width (Write)
1335    Width(3)=2.E-2    !(s)      ! Source pulse width (Erase)
1340    !
1345    Ledge(1)=1.E-5    !(s)      ! Gate pulse leading edge (Write)
1350    Ledge(2)=1.E-5    !(s)      ! Drain pulse leading edge (Write)
1355    Ledge(3)=1.E-5    !(s)      ! Source pulse leading edge (Erase)
1360    !
1365    Tedge(1)=1.E-5    !(s)      ! Gate pulse trailing edge (Write)
1370    Tedge(2)=1.E-5    !(s)      ! Drain pulse trailing edge (Write)
1375    Tedge(3)=1.E-5    !(s)      ! Source pulse trailing edge (Erase)
1380    !
1385    Delay(1)=1.E-4    !(s)      ! Gate pulse delay time (Write)
1390    Delay(2)=2.E-4    !(s)      ! Drain pulse delay time (Write)
1395    Delay(3)=1.2E-2   !(s)      ! Source pulse delay time (Erase)
1400    !
1405    FOR Index=1 TO 6
1410      Count=10^(Index-1)        ! Calculate sum # of pulses
1415      No_pls=Count-Count_old-1  ! Calculate repetitive # of pulses
1420      !
1425      IF No_pls>0 THEN          ! Skip if # of pulse = 0
1430        Repeat_nor1(No_pls,Level(*),Width(*),Delay(*),Ledge(*),Tedge(*))
1435      END IF
1440      !
1445      Write_nor1(Level(*),Width(*),Delay(*),Ledge(*),Tedge(*)) ! Write
1450      !
1455      Meas_vth2(Vth_w(Index))    ! Measure Vth
1460      !
1465      IF Index>2 THEN
1470        Plot_x1y(Count,Count_old,Index,Clr_wrt,Vth_w(*))  ! Plot data
1475      END IF
1480      !
1485      Erase_nor1(Level(3),Width(3),Ledge(3),Tedge(3))  ! Erase
1490      !
1495      Meas_vth2(Vth_e(Index))    ! Measure Vth
1500      !
1505      IF Index>2 THEN
1510        Plot_x1y(Count,Count_old,Index,Clr_wrt,Vth_e(*))  ! Plot data
1515      END IF
1520      !
1525      Count_old=Count
1530    NEXT Index
1535  SUBEND
1540  !
1545 Meas_vth:SUB Meas_vth(Vth)
1550  !*********************************************************************
1555  ! This subprogram measures Vth using analog search
1560  !*********************************************************************
1565    OPTION BASE 1
1570    COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1575    COM /M_vth/ Vgstart,Vgstop,Vd,Id_targ,Rmp,Integ,Igcomp,Idcomp,Hold,Vgstep
1580    INTEGER St
1585  !
1590    Set_tr(Source,Gate,Drain,Sub)
1595    Vth=FNVth5(Id_targ,Vd,Vgstart,Vgstop,Rmp,Integ,0)
1600  !
1605  SUBEND
1610  !
1615 Meas_vth2:SUB Meas_vth2(Vth)
1620   !*********************************************************************
1625   ! This subprogram measures Vth using linear search
1630   !*********************************************************************
1635    OPTION BASE 1
1640    COM /M_vth/ Vgstart,Vgstop,Vd,Id_targ,Rmp,Integ,Igcomp,Idcomp,Hold,Vgstep
```

```
1645    COM /Pin_assign/ INTEGER Drain,Gate,Source,Sub
1650    INTEGER St
1655   !
1660    Connect(FNSmu(1),Drain)
1665    Connect(FNSmu(2),Gate)
1670    Connect(FNSmu(3),Source)
1675    Connect(FNSmu(4),Sub)
1680   !
1685    Force_v(Source,0,20,Idcomp)     ! Apply zero voltage to Source
1690    Force_v(Sub,0,20)               ! Apply zero voltage to Substrate
1695    Force_v(Drain,Vd,2,Idcomp)      ! Apply Vd to Drain voltage
1700   !
1705    Set_lsearch(Gate,Drain,2,Vgstart,Vgstop,Vgstep,Id_targ,Idcomp,Igcomp,Hold)
1710    Search(Vth,St,Idm)
1715   !
1720    Disable_port(Drain,Gate,Source,Sub)
1725    Connect(0,Drain,Gate,Source,Sub)
1730  SUBEND
```

C

HP 4062F System Specifications

This appendix provides the HP 4062F system specifications. These specifications are the performance standards or limits against which HP 4062F is tested. When shipped from the factory, each system meets or exceeds the specifications listed in this appendix.

For complete HP 4142B DCS specifications, refer to the HP 4142B *Operation Manual*.
For complete HP 4280A CMU specifications, refer to HP 4280A *Operation and Service Manual*.
For complete HP 4284A CMU84 specifications, refer to the HP 4284A *Operation Manual*.
For complete HP 8110A PG specifications, refer to the HP 8110A *User's Guide*

Note that reference data are not guaranteed as specifications.

General Description

HP 4062F system can output and measure dc current and dc voltage, measure capacitance, and output pulses to DUTs (which can have up to 48 pins) through the switching matrix.

Basic Measurement Functions:

- Output/measurement of dc voltage/current, capacitance measurement, high-frequency pulse output.
- For dc measurement mode: spot measurement, sweep measurement, pulsed measurement, pulsed sweep measurement, and analog search are available.

Measurement Pins: 48 (maximum)

SWM Instrument Ports: 20

- High Resolution SMU: 1 port
  Measurement range: ±20 fA to ±1 A, ±40 μV to ±200 V
- High Current SMU, Kelvin: 1 port
  Measurement range: ±2 pA to ±1 A, ±40 μV to ±200 V
- SMUs, Kelvin: 2 ports
  Measurement range: ±2 pA to ±1 A, ±40 μV to ±200 V
- GNDU, Kelvin: 1 port
  Voltage output: 0 V
- VS/VMU(AUX): 3 ports
  Output range: ±1 mV to ±40 V
  Measurement range: ±40 μV to ±40 V, differential voltage mode: ±4μV to ±2 V
- High Frequency Ports: 12 ports

Capacitance and Conductance Measurements:

CMU (HP 4280A):	Test frequency: 1 MHz Measurement range: 1 fF to 1.2 nF, 10 nS to 12 mS Dc bias voltage: ±100 V
CMU84 (HP 4284A):	Test frequency: 1 kHz, 10 kHz, 100 kHz, 1 MHz Measurement range: 1 fF to 100 nF, 0.1 nS to 7.5 mS Dc bias voltage: ±40 V

High Frequency Pulse Force: HP 8110A

- Pulse level: ±19 V (at open load)
- Pulse period: 350 ns to 999 s
- Pulse width: 50 ns to 999 ms
- Transition time: 20 ns to 200 ms

Switching Matrix Subsystem

Switching matrix subsystem for the HP 4062F consists of HP 4085F switching matrix and HP 4084B switching matrix controller.

Max. number of DUT Pins:	48 pins
Number of instrument ports:	20 ports 1 port for high resolution SMU 1 port for high current SMU, Kelvin 2 ports for SMU, Kelvin 1 port for GNDU 3 ports for VS/VMU and/or C meter 12 ports for pulse generator (HF ports)
Max. voltage between two ports:	±400 V (any two SMU ports) ±300 V (SMU port and AUX port) ±200 V (any two AUX ports)
Max. voltage at each port:	±200 V (SMU ports) ±100 V (AUX and HF ports)
Max. current, port to DUT pin:	±1.6 A (GNDU port) ±1 A (SMU ports) ±500 mA (AUX and HF ports)
Max. stray capacitance between pins:	<6 pF (between C meter ports)

High Frequency Cable Adapter

The cable adapter is available only with the HP 4085F switching matrix, and includes four switching relays, which can be directly controlled by the PGU of HP 8110A pulse generator.

Switch 1	Semiconductor relay Relay type: transfer Load voltage (peak ac) IN-OUT: 60 V Continuous load current: 0.4 A
Switch 2, 3	Semiconductor relay Relay type: make or break, selectable Load voltage (peak ac) IN-OUT: 60 V
Switch 4	Reed relay Relay type: transfer Load voltage (peak ac) IN-OUT: 200 V Continuous load current: 1 A

The relays can be controlled by an output channel (PGU) of HP 8110A.

Switching time (same for all switches): 2 ms (max.)

DC Measurement Subsystem

The dc measurement subsystem (DCS) consists of the HP 4142B mainframe and plug-in modules. The HP 4142B mainframe has eight slots for plug-in modules. The following shows the available plug-in modules and required number of slots:

HP 41421B 100 V/100 mA SMU, 1 slot
HP 41420A 200 V/1 A SMU, 2 slots
HP 41424A VS/VMU, 1 slot
HP 41425A Analog Feedback Unit (AFU), 1 slot

SMU Port 1: High Resolution Port

Both the HP 41420A and HP 41421B can be used with this port. This port is optimized for low current measurements and is non-Kelvin.

Voltage Source/Monitor Range, Resolution, and Accuracy

Full Scale V Range	Resolution Force/Meas.	Accuracy (Io= output current) ±(%reading + %range + Volt)
±2 V	100 μV/40 μV	±(0.05%+0.05%+0.5Ω×Io+100 μV)
±20 V	1 mV/400 μV	±(0.05%+0.05%+0.5Ω×Io)
±40 V	2 mV/800 μV	
±100 V	5 mV/2 mV	
±200 V[1]	10 mV/4 mV	

1 HP 41420A SMU only.

Current Source/Monitor Range, Resolution, and Accuracy

Full Scale I Range	Resolution Force/Meas.	Accuracy (Vo= output voltage) ±(%reading + %range + Amp)
±1 A[1]	50 μA/20 μA	±[0.5%+(0.1+0.002×Vo)%]
±100 mA	5 μA/2 μA	±[0.2%+(0.1+0.002×Vo)%]
±10 mA	500 nA/200 nA	
±1 mA	50 nA/20 nA	
±100 μA	5 nA/2 nA	
±10 μA	500 pA/200 pA	
±1 μA	50 pA/20 pA	±[0.5%+(0.1+0.002×Vo)%]
±100 nA	5 pA/2 pA	
±10 nA	500 fA/200 fA[2]	±[1%+(0.002×Vo)%+15 pA]
±1 nA	50 fA/20 fA[2]	±[1%+(0.002×Vo)%+8 pA]

1 HP 41420A SMU only.

2 Current measurement resolution of 200 fA/20 fA in the 10 nA/1 nA ranges are obtained when the HP 4142B integration time is set to LONG.

Residual resistance: $\leq 0.5\ \Omega$
Input resistance: $\leq 10^{12}\ \Omega$ (I source mode)
Output terminals: Force, Guard (non-Kelvin)

SMU Port 2: High Current Port

This port is optimized for high current measurements using the HP 41420A SMU.

Voltage Source/Monitor Range, Resolution, and Accuracy

Full Scale V Range	Resolution Force/Meas.	Accuracy (Io= output current) ±(%reading + %range + Volt)
±2 V	100 μV/40 μV	±(0.05%+0.05%+2.5mΩ x Io+100 μV)
±20 V	1 mV/400 μV	±(0.05%+0.05%+2.5mΩ x Io)
±40 V	2 mV/800 μV	±(0.05%+0.05%)
±100 V	1 mV/2 mV	
±200 V[1]	10 mV/4 mV	

1 HP 41420A SMU only.

Current Source/Monitor Range, Resolution, and Accuracy

Full Scale I Range[1]	Resolution Force/Meas.	Accuracy (Vo= output voltage) ±(%reading + %range + Amp)
±1 A[2]	50 μA/20 μA	±[0.5%+(0.1+0.002 x Vo)%]
±100 mA	5 μA/2 μA	±[0.2%+(0.1+0.002 x Vo)%]
±10 mA	500 nA/200 nA	
±1 mA	50 nA/20 nA	
±100 μA	5 nA/2 nA	
±10 μA	500 pA/200 pA	±[0.2%+(0.1+0.003 x Vo)%]
±1 μA	50 pA/20 pA	±[0.5%+(0.01 x Vo)%+2 nA]
±100 nA	5 pA/2 pA	±[0.5%+(0.1 x Vo)%+1 nA]

1 10 nA and 1 nA ranges are available but specs are not guaranteed.

2 HP 41420A SMU only.

Input resistance: $\leq 10^{10}$ Ω (I source mode)
Output terminals: Force, Sense, Guard

SMU Port 3 and 4: Normal Ports

Both the HP 41420A and HP 41421B can be used with these ports.

Voltage Source/Monitor Range, Resolution, and Accuracy

Full Scale V Range	Resolution Force/Meas.	Accuracy (Io= output current) ±(%reading + %range + Volt)
±2 V	100 μV/40 μV	±(0.05%+0.05%+100 μV) (0 A ≤ Io ≤ 0.1 A) Add 40 mΩ x Io[1] (0.1 A < Io ≤ 0.5 A) Add 200 mΩ x Io[1] (0.5 A < Io ≤ 1 A)
±20 V	1 mV/400 μV	±(0.05%+0.05%) (0 A ≤ Io ≤ 0.1 A) Add 40 mΩ x Io[1] (0.1 A < Io ≤ 0.5 A) Add 200 mΩ x Io[1] (0.5 A < Io ≤ 1 A)
±40 V	2 mV/800 μV	±(0.05%+0.05%)
±100 V	5 mV/2 mV	
±200 V[2]	10 mV/4 mV	

1 Reference data.

2 HP 41420A SMU only.

Current Source/Monitor Range, Resolution, and Accuracy

Full Scale I Range[1]	Resolution Force/Meas.	Accuracy (Vo= output voltage) ±(%reading + %range + Amp)
±1 A[2]	50 μA/20 μA	±[0.5%+(0.1+0.002 x Vo)%]
±100 mA	5 μA/2 μA	±[0.2%+(0.1+0.002 x Vo)%]
±10 mA	500 nA/200 nA	
±1 mA	50 nA/20 nA	
±100 μA	5 nA/2 nA	
±10 μA	500 pA/200 pA	±[0.2%+(0.1+0.003 x Vo)%]
±1 μA	50 pA/20 pA	±[0.5%+(0.01 x Vo)%+2 nA]
±100 nA	5 pA/2 pA	±[0.5%+(0.1 x Vo)%+1 nA]

1 10 nA and 1 nA ranges are available, but specs are not guaranteed.

2 HP 41420A SMU only.

Input resistance: $\leq 10^{10}$ Ω (I source mode)
Output terminals: Force, Sense, Guard

Specifications Common to All SMUs

Voltage Source/Monitor Range and Max. Current

Full Scale V Range	Max. Current	
	HP 41420A	HP 41421B
±2 V	1.0 A	100 mA
±20 V	1.0 A (≤14 V) 0.7 A (≤20 V)	100 mA
±40 V	350 mA	50 mA
±100 V	125 mA	20 mA
±200 V	50 mA	——

Current Source/Monitor Range and Max. Voltage

Full Scale I Range	Max. Voltage	
	HP 41420A	HP 41421B
±1 A	14 V (≤1 A) 20 V (≤0.7 A) 40 V (≤350 mA) 100 V (≤125 mA) 200 V (≤50 mA)	——
±100 mA	100 V (≤100 mA) 200 V (≤50 mA)	20 V (≤100 mA) 40 V (≤50 mA) 100 V (≤20 mA)
±10 mA ±1 mA ±100 nA ±10 nA ±1 nA	200 V	100 V

Maximum output power: 14 W (HP 41420A)
2 W (HP 41421B)

Current over-range: 15% of range, 0% of maximum range

Current/voltage compliance:

- Setting range: 1 pA ≤ I compliance ≤ Max. I
  V compliance ≤ Max. V
- Setting accuracy: Same as accuracy indicated previously for all SMUs.
- Opposite polarity setting accuracy: Add 2% of range (100 nA to 1 A ranges)
  Add 10% of range (1 nA to 10 nA ranges)

Accuracies are specified at 23°C ±5°C after a minimum 40 minute warm-up period, with less than 3°C temperature change after calibration. Accuracies apply at all measurement pins. Errors double at 5°C – 18°C and 28°C – 40°C.

Ref. Data Reference Data

Maximum Capacitance Load:	1000 pF
Maximum Allowable Guard Capacitance:	100 pF
Maximum Slew Rate:	0.2 V/μs

Voltage Sources (VSs)

Voltage sources (VSs) are connected to the DUT through the AUX ports on the SWM. These sources can also be used to measure current. Specifications are given at the SWM's DUT pins.

VS Output Range, Resolution, and Accuracy

Full Scale V Range	Resolution	Accuracy	Max. Current
±20 V	1 mV	±(0.1% of setting + 10 mV)	100 mA
±40 V	2 mV	±(0.1% of setting + 20 mV)	20 mA

VS Current Measurement Range, Resolution, and Accuracy

Full Scale I Range	Resolution	Accuracy	max. Voltage
±20 mA	20 μA	±(3% of reading + 200 μA)	40 V
±100 mA	100 μA	±(3% of reading + 1 mA)	20 V

Output Terminal: Force, Guard

Ref. Data Reference Data

Output Impedance:	$\leq$2.0 Ω
Maximum Capacitance Load:	10 μF
Maximum Slew Rate:	0.2 V/μs
Current Limit Accuracy:	-0%, $+20\%$

Voltage Monitors (VMs)

Voltage monitors (VMs) are connected to the DUT through the AUX ports on the SWM. VMs can also be used to provide differential voltage measurements. Specifications are given at the SWM's DUT pins.

VM Measurement Range, Resolution, and Accuracy

Full Scale V Range	Resolution	Accuracy
±2 V	40 μV	±(0.05% of reading + 1.1 mV)
±20 V	400 μV	±(0.05% of reading + 10 mV)
±40 V	800 μV	±(0.05% of reading + 20 mV)

VM Differential Measurement Range, Resolution, and Accuracy

Full Scale V Range	Resolution	Max. Common Voltage	Accuracy (Vi = VM2 input V) ±(% of reading + V + error[1])
±0.2 V	4 μV	40 V	±(0.2% + 0.5 mV + 2.5 μV×Vi)
±2 V	40 μV	40 V	±(0.1% + 2.1 mV + 25 μV×Vi)

1 Common mode error.

Input Terminal: Sense, Guard

Ref. Data Reference Data

Input Impedance: ≥50 MΩ

Pulsed Measurement

Pulsed measurements can be performed using SMUs and VS units.

Measurement functions:	Pulsed spot, pulsed linear sweep, and linear sweep with pulsed bias.
Pulse width setting range:	1 ms to 50 ms, 0.1 ms resolution.
Pulse period setting range:	10 ms to 500 ms, 0.1 ms resolution.
Hold time setting range:	0 to 655.35 s, 0.01 s resolution.
Pulsed V setting range:	2 V to 200 V.
Pulsed I force range:	10 nA to 1 A.

Ground Unit (GNDU)

Using Kelvin connections, this unit forces a 0 V ground for DUTs, so all measurements are made relative to the 0 V. This provides better circuit stability and noise immunity.

Output voltage:	0 V.
V setting accuracy:	±1 mV.
Max. current load:	±1.6 A.
Max. capacitance load:	10 μF (reference data).

Analog Feedback Unit (HP 41425A AFU)

The AFU searches for a target current or voltage on one SMU by controlling (sweeping) the output voltage of another SMU.

- Monitor (sense) SMU

Setting range and resolution:	See SMU specification.
Setting accuracy:	SMU's accuracy + 0.1% of setting + 0.1% of range.
Max. current setting:	900 mA (HP 41420A), 90 mA (HP 41421B).
Max. voltage setting:	180 V (HP 41420A), 90 V (HP 41421B).

- Search SMU

Search V range:	2 V, 20 V, 40 V, 100 V, 200 V (Note that 200 V range is only for HP 41420A SMU.)
Search V slew rate range:	0.5 mV/ms to 100 V/ms
Search start V accuracy:	0.5% of setting + 0.5% of V range
Search stop V accuracy:	3% of V range
Search V slew rate resolution:	30% of setting + 10% of slew rate range[1]
Delay time of search V:	5 μs[1]
Feedback integration time range:	2.5 μs to 500 ms
Feedback integration time accuracy:	30% of setting[1]

[1:] Reference data.

Capacitance/Conductance Measurement Subsystem

Capacitance (C)—conductance (G) measurements use an HP 4280A CMU or HP 4284A CMU84. For detailed specifications, see operation manual of each instrument.

CMU (HP 4280A 1 MHz C Meter/C-V Plotter)

Measurement range:	C: 1 fF to 1.2 nF G: 0.01 μS to 12 mS
C-t measurements:	Time interval: 60 ms to 32 s Pulse width setting range: 10 ms to 32 s
Test frequency:	1 MHz±0.01%
OCS level:	30 mVrms±10% or 10 mVrms±10%
Display resolution:	$4\frac{1}{2}$ max.

C/G Measurement Range and Accuracy

Measurement Range		Accuracy[1]
10 pF/100 μS[2]	C	± [0.8% + (0.7 + 1.5 x Gm/100μS)%]
	G	± [1.5% + (0.7 + 0.2 x Cm/10pF)%]
100 pF/1000 μS	C	± [0.5% + (0.1 + 1.5 x Gm/1000μS)%]
	G	± [1.5% + (0.2 + 0.2 x Cm/100pF)%]
1000 pF/10 mS	C	± [1.5% + (0.2 + 3.5 x Gm/10mS)%]
	G	± [3.5% + (0.4 + 0.5 x Cm/1000pF)%]

1 How to read the accuracy: ±(% of reading + % of range),
Cm=measured capacitance value, Gm=measured conductance value.
Where, test signal level is set to 30 mV and dissipation D is less than 1.
When using 10 mV test signal level, add (0.1% + 0.1%) to C or G error.
Specified C and G accuracy is between any two measurement pins on SWM at 23°C±5°C after minimum 40 minute warm-up period. Stray capacitance between two measurement pins is measured first. Then stray capacitance value is subtracted from measured data when DUT is measured.
Specified accuracy is for when guard capacitance outside SWM (that is between pins and SWM frame or between pin and its guard terminal) is 5 pF or less. If guard capacitance exceeds 5 pF, add 0.8% of reading to C or G errors for each 10 pF increment.
Errors double at 5°C – 18°C and 28°C – 40°C.

2 Reference data.

Output impedance of dc bias source: about 30 Ω
Accuracy of dc bias source (reference data): Setting range: −100 to +100 V

V Range	Range Coverage	Resolution	Accuracy ±(% of setting + offset)
2 V	±(0.000 to 1.999) V	1 mA	±(0.2% + 0.01 V)
20 V	±(2.00 to 19.99) V	10 mV	±(0.1% + 0.02 V)
100 V	±(20.0 to 100.0) V	100 mV	±(0.1% + 0.1 V)

CMU84 (HP 4284A Precision LCR Meter)

Measurement range:

C: 1 fF to 1.2 nF (1 MHz)
1 fF to 10 nF (100 kHz)
1 fF to 100 nF (10 kHz)
10 fF to 100 nF (1 kHz)

G: 10 nS to 7.5 mS (1 MHz)
1 nS to 6.3 mS (100 kHz)
0.1 nS to 6.3 mS (10 kHz)
0.1 nS to 0.63 mS (1 kHz)

Measurement frequency: 1 kHz, 10 kHz, 100 kHz, 1 MHz

Frequency accuracy: ±0.01%

OCS level: 30 mVrms

Capacitance Measurement Frequency, Range, and Accuracy

Frequency (Hz)	C (F)	Accuracy[1]
1 M	10 p[2]	±[0.8%+(1.0+0.6×Gm/63 μS)]
	100 p	±[0.8%+(0.3+0.6×Gm/630 μS)]
	1 n[3]	±[1.9%+(0.2+1.7×Gm/6.3 mS)]
100 k	10 p[2]	±[0.4%+(1.0+0.4×Gm/6.3 μS)]
	100 p	±[0.3%+(0.3+0.3×Gm/63 μS)]
	1 n	±[0.3%+(0.2+0.4×Gm/630 μS)]
	10 n	±[0.5%+(0.2+1.0×Gm/6.3 mS)]
10 k	100 p	±[0.3%+(0.2+0.3×Gm/6.3 μS)]
	1 n	±[0.3%+(0.1+0.3×Gm/63 μS)]
	10 n	±[0.3%+(0.1+0.3×Gm/630 μS)]
	100 n	±[0.3%+(0.1+1.0×Gm/6.3 mS)]
1 k	100 p[2]	±[0.4%+(0.5+0.4×Gm/0.63 μS)]
	1 n	±[0.3%+(0.1+0.3×Gm/6.3 μS)]
	10 n	±[0.3%+(0.1+0.3×Gm/63 μS)]
	100 n	±[0.3%+(0.1+0.3×Gm/630 μS)]

1 How to read the accuracy: ±(% of reading + % of range), Cm=measured capacitance value, Gm=measured conductance value. Specified C and G accuracies are at 23°C±5°C after a minimum 15 minute warm-up period with MEAS SPEED set to MEDIUM. The stray capacitance between two measurement pins is measured first. Then the stray capacitance value is subtracted from the measured data when a DUT is measured.

2 Reference data.

3 120% full scale.

Conductance Measurement Frequency, Range, and Accuracy

Frequency (Hz)	G (S)	Accuracy[1]
1 M	63 μ[2]	±[0.8%+(1.0+0.6×Cm/10 pF)]
	630 μ	±[0.8%+(0.3+0.6×Cm/100 pF)]
	6.3 m	±[1.3%+(0.2+2.2×Cm/1.0 nF)]
100 k	6.3 μ[2]	±[0.4%+(1.0+0.4×Cm/10 pF)]
	63 μ	±[0.3%+(0.3+0.3×Cm/100 pF)]
	630 μ	±[0.3%+(0.2+0.4×Cm/1 nF)]
	6.3 m	±[0.7%+(0.2+0.8×Cm/10 nF)]
10 k	6.3 μ	±[0.3%+(0.2+0.3×Cm/100 pF)]
	63 μ	±[0.3%+(0.1+0.3×Cm/1 nF)]
	630 μ	±[0.3%+(0.1+0.3×Cm/10 nF)]
	6.3 m	±[0.7%+(0.1+0.7×Cm/100 nF)]
1 k	0.63 μ[2]	±[0.4%+(0.5+0.4×Cm/100 pF)]
	6.3 μ	±[0.3%+(0.1+0.3×Cm/1 nF)]
	63 μ	±[0.3%+(0.1+0.3×Cm/10 nF)]
	630 μ	±[0.3%+(0.1+0.3×Cm/100 nF)]

1 How to read the accuracy: ±(% of reading + % of range), Cm=measured capacitance value, Gm=measured conductance value. Specified C and G accuracies are at 23°C±5°C after a minimum 15 minute warm-up period with MEAS SPEED set to MEDIUM. The stray admittance between two measurement pins is measured first. Then the stray admittance value is subtracted from the measured data when a DUT is measured.

2 Reference data.

Accuracy of dc bias source: Signal level ≤2 Vrms

V Range	Resolution	Accuracy[1]
±(0.000 to 4.000) V	1 mV	±(0.1% of setting + 1.2 mV)
±(4.002 to 8.000) V	2 mV	±(0.1% of setting + 2.2 mV)
±(8.005 to 20.000) V	5 mV	±(0.1% of setting + 5 mV)
±(20.01 to 40.000) V	10 mV	±(0.1% of setting + 10 mV)

1 The specified accuracy is when the bias current sink function is set to off. When the bias current sink function is set to on, add ±20 mV to each accuracy value (dc bias current ≤ 1 μA).

High Frequency Pulse Force

The high frequency pulse force is available from up to five HP 8110As. Refer to the HP 8110A manuals about the standalone HP 8110A specifications.

- Pulse Force Mode

  Pulse signal: 2-level and 3-level output

  Burst count: 10^9 (max.)

  Output mode: Normal mode: all pulse generators (up to 5) can force simultaneously.
  Pattern mode: all pulse generators except a master pulse generator can force simultaneously (more accurate).
  The pattern mode means the pattern mode of the HP 8110A. For relatively long pulse periods, you can set more accurate pulse width and delay time in the pattern mode.

  Load impedance: 999 kΩ (default), selectable from 2.5 Ω to 999 kΩ

- Pulse Setting Range

  Pulse period: Can be set automatically according to pulse delay and width, or can be specified. If specified:

  Normal mode: range: 350 ns to 999 s
  resolution: 4 digits, min. 10 ps

  Pattern mode: range: 120 μs to 999 s
  resolution: 4 digits, min. 100 ns

  Pulse width: Normal mode: range: 50 ns to 999 ms
  resolution: 3 digits (min. 10 ps)

  Pattern mode: range: 50 ns to 999 ms (in master unit),
  100 ns to 999 ms (in slave units)
  resolution: 3 digits, min. 10ps (in master unit),
  resolution in slave units:

  1×10^{-n} if $1000\times10^{-n} <$ pulse period $\leq 4000\times10^{-n}$
  2×10^{-n} if $4000\times10^{-n} <$ pulse period $\leq 8000\times10^{-n}$
  2.5×10^{-n} if $8000\times10^{-n} <$ pulse period $\leq 10000\times10^{-n}$

  Pulse delay: Normal mode: range: 0 to 998 ms
  resolution: 3 digits, min. 10 ps

  Pattern mode: range: 0 to 998 ms
  resolution: 3 digits, min. 10ps (in master unit)
  resolution in slave units:

  1×10^{-n} if $1000\times10^{-n} <$ pulse period $\leq 4000\times10^{-n}$
  2×10^{-n} if $4000\times10^{-n} <$ pulse period $\leq 8000\times10^{-n}$
  2.5×10^{-n} if $8000\times10^{-n} <$ pulse period $\leq 10000\times10^{-n}$

  Transition time: 20 ns to 200 ms

Pulse level:	(at 999 kΩ load impedance)	
	2-level	±19 V
	3-level	±19 V (Restriction for "3-level pulse with one PGU" output mode: sum of absolute values of both amplitudes cannot exceed 20 V.)
	Accuracy	±(2% of amplitude + 150 mV) For 2-level output with greater than 1 GΩ of load impedance.

- Pulse Accuracy (at 50 Ω load impedance)

Pulse width:	≤ ±5% ±1 ns (in normal mode)
Pulse delay:	≤ ±5% ±1 ns (in normal mode)
Transition time:	≤ (−10% to +10%) + 20 ns
Overshoot:	≤ ±5% of output level ±20 mV

- Pulse Accuracy (at 5 kΩ load impedance); Reference data

Overshoot/ringing	±5% of amplitude ±20 mV; where transition time ≥ 20 ns
Transition time:	±(10% +5 ns)
Skew between pins:	±10 ns

System Controller

Supported computer:	HP 9000 Series 300 Model 340, 345, 360, 370, 375, 380, 382
Operating system:	HP-UX 9.0 and HP BASIC/UX 6.3
Required memory:	8 megabytes or more (16 MB or more when using X11 Windows)
HP-IB interface:	Four separate HP-IB interfaces are recommended except when a diskless cluster is used. One interface for each of following:

- Peripherals
  The computer's built-in HP-IB port must be used to control any printers, plotters, and tape drives.
- HP 4062F Hardware
  HP 98624A HP-IB card used to control the HP 4062F system instruments. If an HP 4062F diskless node configuration is used with no other peripherals connected, the built-in HP-IB port can be used to control the HP 4062F.
  If it is not possible to have a dedicated HP 98624A for the wafer prober, the prober can be controlled using the same HP-IB card as the HP 4062F.
- Disk Drive
  If computer has a built-in high speed HP-IB interface (such as model 360/370), then this interface must be used to control the disk drive. If other models are purchased, then the HP 98625B (High Speed Disk Interface) and the HP 98620B (DMA Controller) cards are required. SCSI disk drives are also supported.
- Additional instruments and wafer prober
  An additional HP 98624A for non-HP 4062F hardware, such as a wafer prober, is recommended.

HP-HIL device:	HP 46084A ID module

General Specifications

Operating temp. range: 5°C to 40°C; 5% to 70% RH. Change after calibration must be less than 3°C.

Storage temp. range: −20°C to 60°C; 15% to 90% RH @65°C,

Air cleanliness: Better than class 100,000

Power requirements: 100 Vac (90 – 110 Vac), 120 Vac (108 – 127 Vac), 220/240 Vac (198 – 264 Vac)
44 – 66 Hz

Total Power Requirements for Standard Configuration, except system controller

Line Voltage	Number of HP 8110As				
	1	2	3	4	5
100 V	8.8 A	11.6 A	14.4 A	17.2 A	20.0 A
120 V	8.0 A	10.5 A	13.0 A	15.5 A	18.0 A
220/240 V	5.0 A	6.5 A	8.0 A	9.5 A	11.0 A

Where, the standard configuration consists of an HP 4084B; an HP 4142B with an HP 41420A, three HP 41421Bs, an HP 41424A and an HP 41425A; and an HP 4280A. See the table on the number of HP 8110As.

Dimensions: System cabinet: 600 mm(W)×1600 mm(H)×800 mm(D)
Switching matrix: 470 mm(W)×230 mm(H)×440 mm(D)

Weight: System cabinet: about 230 kg
Switching matrix: about 22 kg
Pin board: 180 g/board

System Software

Standard HP 4062F system software provides the following capabilities:

- Control of system subsystems (Test Instruction Sets)
- Monitor of system subsystems and measurement data (Virtual Front Panel)
- Parameter measurement utility (PARA Library)
- Probing pattern generator and wafer map display (PPG, MAP)
- BSDM data format generation library (File Creation Library)
- Wafer prober control utility subprograms include:
  - Electroglas Model 1034X/2001X/2010X/3001X
  - Tokyo Seimitsu Model APM-3000A/6000A/7000A
- Operational verification diagnostics

Typical Performance

Pin-port connection in SWM: 4.5 ms

DC I measurement[1]

Force I or V: 3.9 ms
Measure I or V: 6 ms
Analog Search[2] : 25 ms

1 Integration time SHORT, except 1 nA/10 nA ranges, excluding range changing time and settling time.

2 For MOSFET Vth measurement (Id=1 μA) using the AFU; excluding the switching time.

Typical SWM Performance

Typical performance of HP 4085F high frequency switching matrix (at 23 °C, 50% RH)

Offset voltage:	50 μV (for all ports)
Offset current:	1 pA (high resolution SMU port) 40 pA (other SMU ports)
Isolation resistance:	8×10^{13} Ω (between high resolution SMU port and another SMU port) 5×10^{11} Ω (between any two other SMU ports)
Path to ground capacitance:	30 pF (high resolution SMU port) 60 pF (other SMU ports)
−3 dB bandwidth:	35 MHz (at 50 Ω load impedance)
Crosstalk between pins:	±2% (5 kΩ load impedance)

Error Messages

If your HP 4062F detects a hardware failure or an illegal operation, an error message or error code, often both, will appear on the system controller's display.

Errors are divided into two basic categories: system controller-related and HP 4062F-related, as explained here.

- Errors on System Controller or HP BASIC/UX Programming Language System

  If a system controller error or an HP BASIC/UX language error occurs, a BASIC/UX language system error code is displayed, as in the following example:

  ```
  Error 7 in 1250  Undefined function or sub
  ```

  If a system controller hardware failure occurs, a SYSTEM ERROR message is displayed.

- Errors on HP 4062F System

  If an HP 4062F error occurs, an error message is displayed, as in the following example:

  ```
  TIS ERROR 20 Device not present.
  ERROR 449 IN 10
  ```

  Error message prefixes give the name of the program or subprogram in which the error occurred. In the example above, the error occurred in TIS. In some cases, only an error code displayed.

  BASIC/UX system error code may also be displayed. In the example above, "449" is BASIC/UX system error code.

This chapter covers HP 4062F-related error messages. For the other error messages, see "Error Message" in the HP 4062C/UX *Programming Reference* manual. For the BASIC/UX system error messages, see *BASIC/UX Language Reference* manual.

This chapter covers only HP 4062F system error messages that you may encounter when using START.F, TIS.F, VFP.AC, and DIAG.F. If you encounter error messages that are not listed here, see "Error Message" in the HP 4062C/UX manuals.

START.F Error

The following are HP 4062F error messages related to the START.F program:

START ERROR 500 Invalid FLCONFIG entry.

The FLCONFIG file may have following mistakes:

- For a non-comment line, the first field is not `PG1` to `PG5` or `HF1` to `HF12`.
- For a line that has `PG1` to `PG5` in the first field, no device selector (select code + HP-IB address) is specified.
- For a line that has `HF1` to `HF12` in the first field, the second field is not `PG11` to `PG52`.
- File type is not ASCII. Cannot handle file that was saved by STORE command, must use SAVE command.

Modify FLCONFIG file. For details on the FLCONFIG file, see "Configuration File /usr/pcs/etc/FLCONFIG" in Chapter 1.

START ERROR 501 Invalid HP-IB address.

The device selector ("select code" + "HP-IB address") of the pulse generator (PG) in the FLCONFIG file has one of following errors:

- Select code is not in range of 7 to 31.
- HP-IB address is not in range of 0 to 30
- Select codes of all PGs are not identical.

Modify FLCONFIG file. For details on the FLCONFIG file, see "Configuration File /usr/pcs/etc/FLCONFIG" in Chapter 1.

START ERROR 502 PGs with same HP-IB address are specified in FLCONFIG.

Two or more pulse generators (PGs) have the same HP-IB address in the FLCONFIG file. Modify the FLCONFIG file.

START ERROR 503 HP-IB interface card does not exist.

HP-IB interface is not recognized when the system polls the HP-IB address of the pulse generators (PGs) described in the FLCONFIG file. Check the following:

- Are the HP-IB interface cards in the system controller?
- Is the HP-IB interface card damaged?
- Is select code of HP-IB interface card identical to that described in the FLCONFIG file?

START ERROR 504 Duplicate definition.

Pulse generators (`PG`*x*) or HF ports (`HF`*xx*) are defined in duplicate in the FLCONFIG file. Modify the FLCONFIG file.

START ERROR 505 Unsupported Pulse Generator specified.

The model number of the pulse generator (PG) in the FLCONFIG file is not supported. Only HP 8110A is supported. Modify the FLCONFIG file.

START ERROR 506 Unsupported SWM used.

Connected SWM cannot be used for the HP 4062F system.

START ERROR 507 PG1 does not exist.

There is no PG1 definition in FLCONFIG file. Modify the FLCONFIG file.

START ERROR 508 Calibration data file does not exist.

Calibration data file (/usr/pcs/etc/CAL_8110) for the pulse generators (PGs) does not exist. After copying from the /usr/pcs/newconfig/CAL_8110 file to the /usr/pcs/etc directory, execute the AC Test of the PV4062.F again to obtain the calibration data.

START ERROR 509 Calibration data corrupted.

Calibration data file (/usr/pcs/etc/CAL_8110) for the pulse generators (PGs) is corrupted. After copying from the /usr/pcs/newconfig/CAL_8110 file to the /usr/pcs/etc directory, execute the AC Test of the PV4062.F again to obtain the calibration data.

TIS.F Error

The following are HP 4062F error messages related to the TIS.F program:

TIS ERROR 300 Output channel does not exist.

The program tries to force a 3-level pulse from a pulse generator (PG) that is equipped with one output unit. When forcing "3-level pulse with one PGU", use PG that has two output units.

TIS ERROR 301 Output channel busy.

Specified pulse generator output unit (PGU) is disabled by "3-level pulse with one PGU" output mode. Set the PGU to "2-level pulse" output mode using the Set_type_pg subprogram.

TIS ERROR 302 Amplitude out of range.

Specified amplitude is out of range. Reset to proper value.

TIS ERROR 303 Level out of range.

Specified amplitude and pulse base values exceed the output range of the pulse generator (PG). Reset to proper amplitude and pulse base values.

TIS ERROR 304 Transition out of range.

Specified transition times (leading-edge or trailing-edge) are not in allowable range. Reset to proper transition time.

TIS ERROR 305 Lead/trail edge times not same range.

Specified leading-edge and trailing-edge transition times are not in the same range. Reset to proper transition time.

TIS ERROR 306 Width out of range.

Specified pulse width value is out of allowable range. Reset to proper pulse width.

TIS ERROR 307 Delay out of range.

Specified delay time value is out of allowable range. Reset to proper delay time.

TIS ERROR 308 2nd delay too short.

Specified second delay time is too short. The second delay time must be greater than sum of "first delay time" + "first pulse width". Reset the second delay time.

TIS ERROR 309 Period out of range.

Specified pulse period is out of allowable range. Reset to proper pulse period.

TIS ERROR 310 Period too short.

Specified pulse period is too short. Reset to proper pulse period.

TIS ERROR 311 Impedance out of range.

Specified load impedance value is out of range. Reset to proper load impedance.

TIS ERROR 312 PG1 does not exist.

PG1 does not exist. Check the contents of the FLCONFIG file.

TIS ERROR 313 PG2 does not exist.

PG2 does not exist. Check the contents of the FLCONFIG file.

TIS ERROR 314 2nd unit ignored.

Pulse generator output unit (PGU) connected to the input B terminal of the multiplexer switch is not set to "3-level pulse using the multiplexer" output mode by the Set_type_pg subprogram. Set the PGU correctly.

TIS ERROR 315 Mode value out of range.

Specified mode value is not correct. Reset the proper mode value.

TIS ERROR 316 Number of periods is out of range.

Specified number of periods is out of range. Reset to proper number of periods.

TIS ERROR 317 Improper No. of parameters specified.

An incorrect number of parameters was specified for this subprogram. Check the number of parameters.

TIS ERROR 318 No trigger received within timeout.

Trigger signal is not correctly transferred between the pulse generators. Make sure cable between TRIGGER OUT terminal of PG2 and PLL REF IN/CLK IN terminals of the PG3 to PG5 are not broken.

VFP.AC Error

The following are HP 4062F error messages related to the VFP.F program:

VFP.AC ERROR 1 Setting file does not exist.

Specified setting data file does not exist. Check the specified file name.

VFP.AC ERROR 2 Duplicate file name.

Specified file name already exists. Select `Re-store Setup` softkey to save the setting data using the existing file name.

VFP.AC ERROR 3 More than 4 pins assigned to an HF port.

One HF port can connect to maximum four measurement pins, and HF ports can connect to specific pins only (HF1 to pin1-4, HF2 to pin5-8, and so on).

VFP.AC ERROR 4 No pair unit (2nd PGU) found.

There is no PGU connected to the input B of the multiplexer. Make sure actual connections match contents of FLCONFIG file.

VFP.AC ERROR 5 Invalid separator used.

The separators to separate multiple measurement pins are not correct. Use space or comma as the separator.

Diagnostics (DIAG.F) Error

The following are HP 4062F error messages related to the DIAG.F program.

DIAG. ERROR 50 SMU1 does not exist.

The system software cannot find the SMU1. Connect the SMU1 properly.

DIAG. ERROR 51 Self-test failed.

HP 8110A self-test failed. If the HP 8110A self-test continues to fail, contact your nearest HP Service Office.

DIAG. ERROR 52 Improper connection between SWM and PG.

The system software found abnormal connections between the switching matrix (SWM) and pulse generators (PGs). The following are possible causes:

- The actual connections do not match the connections described in the FLCONFIG file. Modify FLCONFIG file or change the connections between the HF ports of the SWM and output units of PGs.
- Cables between the HF port of the SWM and output units of PGs may be broken. Check the cables.

DIAG. ERROR 53 No trigger line.

The system software found abnormal connections between the trigger terminals of the pulse generators (PGs). The following are possible causes:

- The trigger connections between the PGs are not correct. Refer to "Trigger Connections between Pulse Generators" in Chapter 1 and check the connections.
- Cables between the trigger terminals of PGs may be broken, or the pulse splitters are damaged.

DIAG. ERROR 54 Switch not properly working

The system software found abnormal operation of the switches. Check the actual switch configuration against the switch configuration settings, then run the DIAG.F program again. If the same error is observed, contact your nearest HP Service Office.

DIAG. ERROR 55 Erroneous PG assignment.

The pulse generator output units (PGUs) are not properly assigned to the inputs of the switches or input of the switch control (CTRL). Make sure that one PG*xx* is not assigned to multiple switch inputs or switch control terminals.

Index