PI-CONVERT

File Conversion Utility
for the PI-2005 Pattern Generator

Users Guide

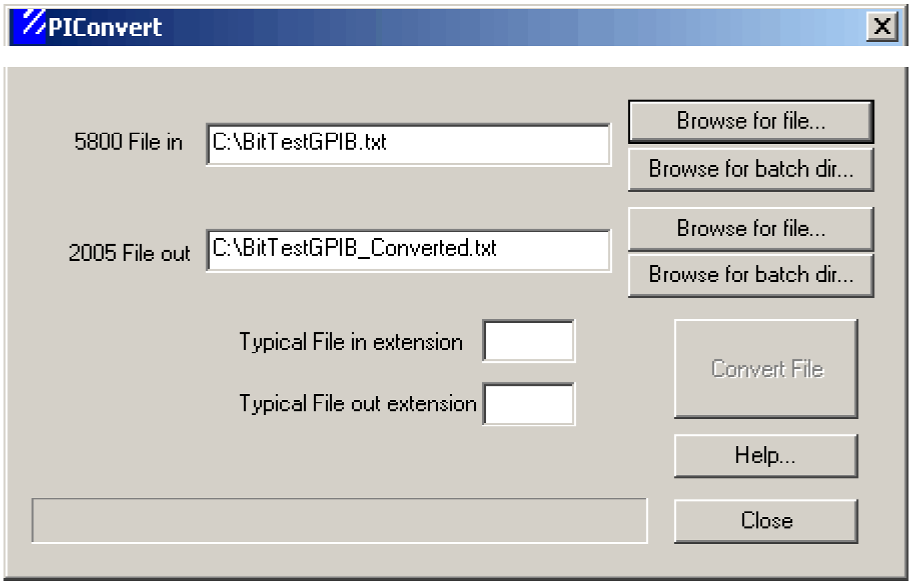
July 2001

Pulse Instruments
1234 Francisco Street
Torrance, CA 90502
(310) 515-5330
http://www.pulseinstruments.com
support@pulseinstruments.com

TABLE OF CONTENTS

LIST OF FIGURES

1. PI-CONVERT USAGE INSTRUCTIONS



5800 File in: C:\BitTestGPIB.txt

2005 File out: C:\BitTestGPIB_Converted.txt

Typical File in extension:

Typical File out extension:

Figure 1: PI-CONVERT Window

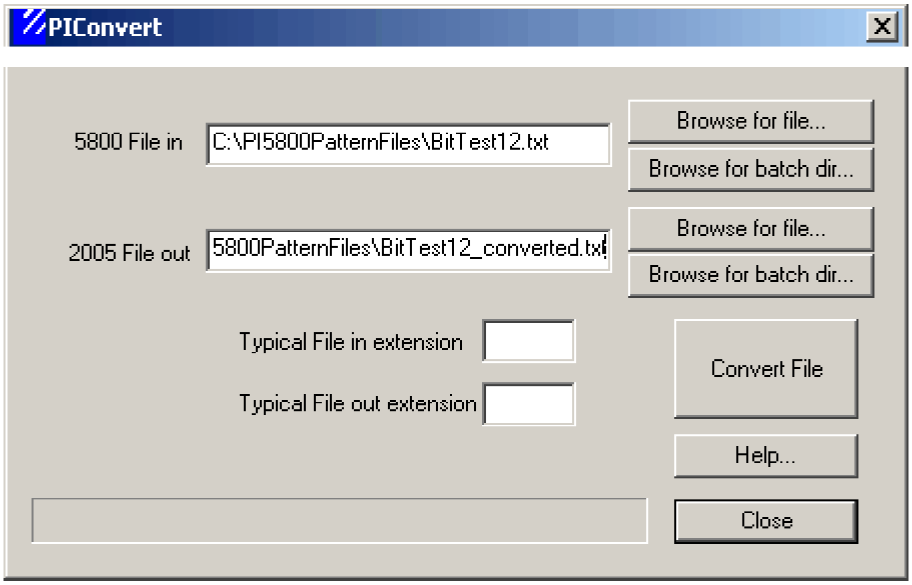
1.1. Overview

PI-CONVERT is a utility designed to convert GPIB command files written for the PI-5800 into files compatible with Pulse Instruments PI-2005 Pattern Generator. PI-CONVERT does not convert PI-PAT files (files with the extensions .w58, .w65, or .w20). To convert PI-PAT files, use the **Save As** function in PI-PAT's **File** menu.

PI-CONVERT has two modes of operation: Single file and Multiple file (Batch mode).

1.2. Single File Mode

1. Enter the file path for the file to be converted in the "**5800 File in**" edit box. (Select **Browse for file...** to select the file using the standard Windows file browser):



5800 File in: C:\PI5800PatternFiles\BitTest12.txt

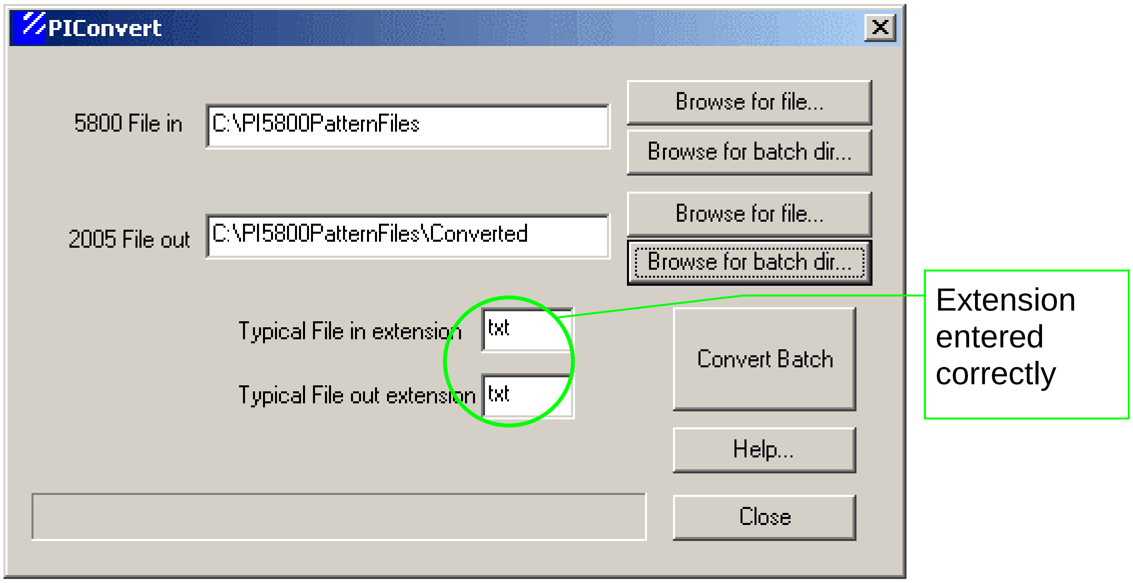
2005 File out: 5800PatternFiles\BitTest12_converted.txt

Figure 2: Single file conversion

2. Enter a file path or directory of the target location for the converted file. (Select **Browse for file...** or **Browse for batch dir . . .** to select using a browser). If a directory is entered, then the converted file will be given the same name as the source file and the extension will be that specified in the **Typical file out extension** edit box. If a filename is specified in the **2005 File Out box**, then it is not necessary to enter anything in the **Typical file out extension** box.
3. Click **Convert File** button to begin conversion. The progress bar will flash briefly while the file is being converted.

1.3. Multiple File (Batch Mode)

1. Enter a path to the directory that contains the source files to convert in the **5800 File In** edit box. (Select **Browse for batch dir...** to select the directory using a browser). The source files must all be named with the same extension as specified in the **Typical file in extension**. The file extension should not include a period.



5800 File in: C:\PI5800PatternFiles

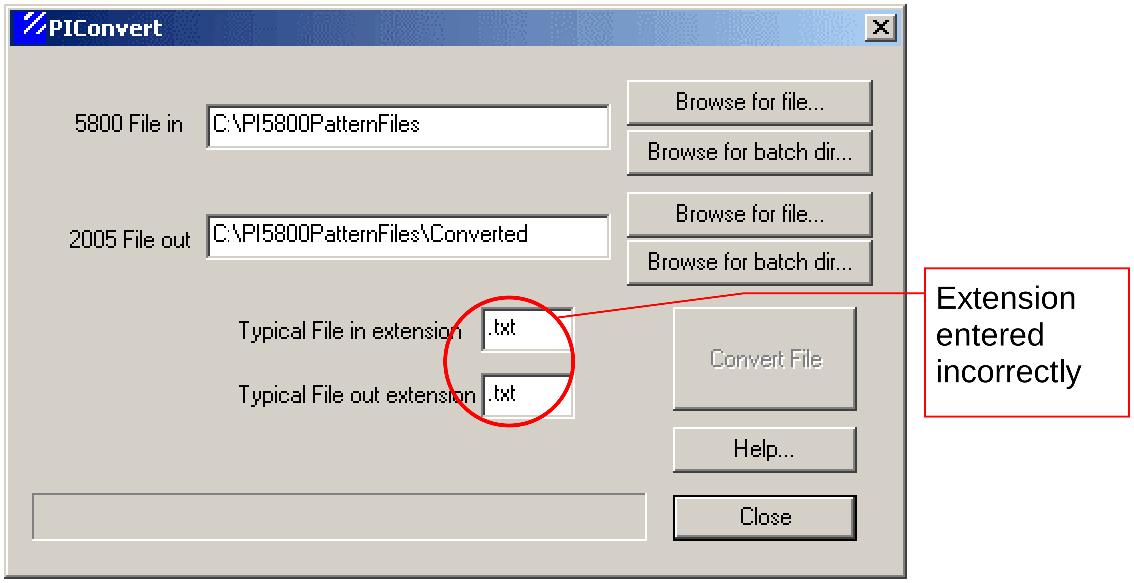
2005 File out: C:\PI5800PatternFiles\Converted

Typical File in extension: txt

Typical File out extension: txt

Extension entered correctly

Figure 3: Correct extension entry



5800 File in: C:\PI5800PatternFiles

2005 File out: C:\PI5800PatternFiles\Converted

Typical File in extension: .txt

Typical File out extension: .txt

Extension entered incorrectly

Figure 4: Incorrect extension entry

2. Enter a path to the directory of the target location for the converted files. (Select **Browse for batch dir...**to select using a browser). Each converted file will be given the same name as the source file from which it is converted, and the extension will be that specified in the **Typical file out extension** edit box.

3. Click **Convert File** button to begin the batch conversion. The progress bar will show the progress of files being converted.

1.4. Conversion Logs

PI-CONVERT produces two types of conversion logs—matching logs and batch logs. Each time a file is converted, PI-CONVERT will produce a matching log file containing a note of each change that was made and any warnings that were generated. The matching log file will have the same filename as the original file, but with the extension ".log", so if the original file had been named "**BIT_TEST12.txt**", the conversion log would be named "**BIT_TEST12.log**".

PI-CONVERT will also append notes to a running batch log. The batch log, named **PI-CONVERT.log**, will contain a batch header and a short summary for each file that was converted:

- Name of the input file
- Name of the output file or an error message that conversion was not completed
- Number of changes
- Number of warnings

If you are converting large numbers of files, it may be more convenient for you to examine the batch log to determine which files require further inspection or modification before they are used in your test plan.

By default, **PI-CONVERT.log** will be created in the same directory specified for the output files. If **PI-CONVERT.log** does not already exist, it will be created for you. If **PI-CONVERT.log** already exists, a new batch header and summaries for each converted file will be appended to the end of the existing file.

1.4.1. Changes recorded in the conversion log

Changes are any modifications to the pattern file that will result in output identical to that of the original file. Examples of changes are bit-multiplication for short subpatterns and subpattern-duplication for overlapping subpatterns:

- `SP 1 TOO SHORT. PATTERN DATA BIT-DOUBLED, CLOCK FREQUENCY DOUBLED`
- `SP 3-4 OVERLAP. SP 2 REDEFINED AT BITS 101-122`

PI-CONVERT will also make notations within the converted file itself whenever a clock period argument is changed. For example, if the converted file has an illegal 12-bit subpattern, PI-CONVERT will bit-double all pattern data and divide the clock period by 2. To facilitate any necessary modification of your application that does not occur within PI-CONVERT, PI-CONVERT will insert a comment on a new line immediately preceding the new PERIOD command. For example, if the original file had had a line reading:

```
PERIOD 40 NS
```

the converted file would have in its place the lines:

```
;PI-CONVERT:PERIOD DIV 2

PERIOD 20 NS
```

If your application sends pattern files to the 5800 and subsequently sends PERIOD commands that are not contained in the converted files, your application may need to know which pattern files have been changed from their original frequency. Your application may need to be modified to look for this comment line in each file it handles in order to determine whether or not subsequent PERIOD commands need to be modified.

1.4.2. Warnings recorded in the conversion log

Warnings will be reported for syntax errors or for modifications that may not produce identical output. Examples of warnings that may be reported are:

- `CLOCK PERIOD TOO LONG`

- CONDITIONAL BRANCH NOT SUPPORTED
- CASCADE MODE NOT SUPPORTED
- DELAY COMMENTED-OUT:DELAY ARGUMENT SHOULD BE RE-ENTERED

Command lines with unsupported commands will commented-out with the semi-colon character. Command lines with illegal arguments (e.g. a PERIOD command with an argument greater than 80 US) will be left as-is. Users should examine any pattern file that generates a warning during conversion.

2. PI-CONVERT REFERENCE

Although the PI-2005 was designed to provide maximum backward compatibility with the PI-5800A, there are a few command and parameter adjustments that may be required in order to allow the PI-2005 to run programs written for the PI-5800A.

The following sections describe modifications that are made to pattern files by PI-PAT and PI-CONVERT to ensure compatibility. For convenience in visualizing the changes, PI-PAT screenshots are used for illustration, but the concepts shown here apply to PI-PAT as well as text files containing GPIB commands.

2.1. Minimum subpattern length

The minimum allowable subpattern length in a PI-5800A program is 12 bits. The minimum subpattern length for a PI-2005 program is 20 bits. New programs should be written according to the new minimum length.

Existing programs that violate the new minimum subpattern length can be converted to run on a PI-2005 using PI-PAT or PI-CONVERT.

PI-PAT and PI-CONVERT avoid the minimum subpattern length violation by bit-multiplication and clock-multiplication. PI-CONVERT and PI-PAT replace each bit in the old program with two bits in the new program while doubling the programmed clock frequency.

2.1.1. Minimum subpattern length conversion, NRZ example

In this example, the “Unconverted” PI-5800 program runs at 50 MHz and has the following “illegal” 12-bit subpattern definition:

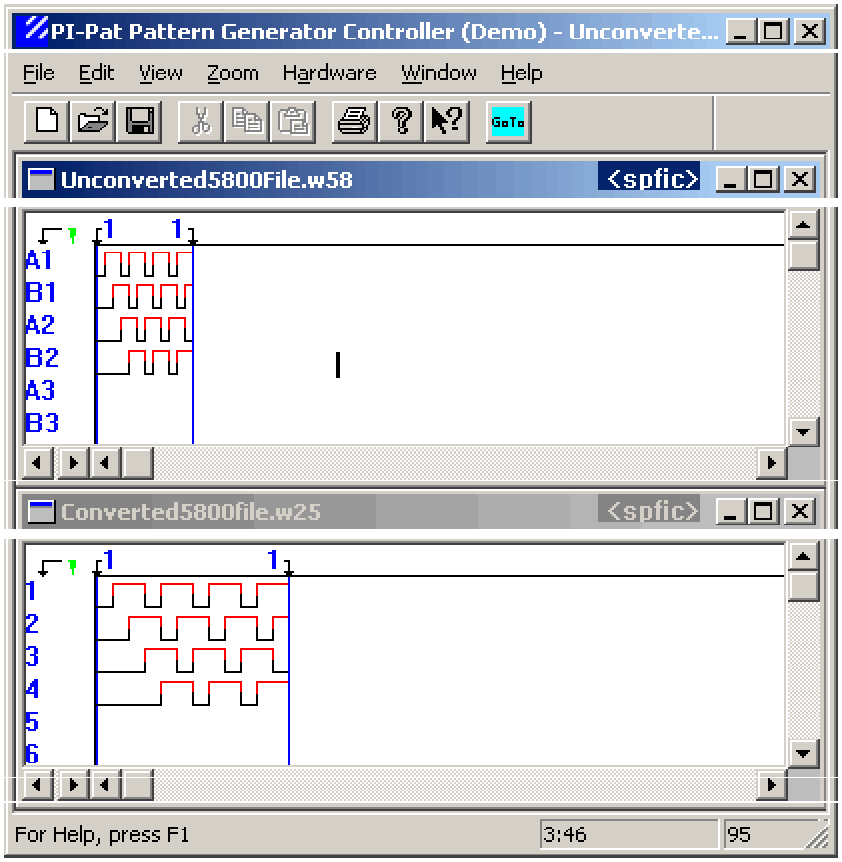


Figure 5: Converted PI-5800A file with "illegal" 12-bit subpattern definition in NRZ format

The "Converted" file has each bit replaced by two identical bits, and if run at 100 MHz, will have output identical to that of the original program run at 50 MHz on a PI-5800A. Note that this can also be verified by examining the original program at **2x** zoom using the **View** menu.

2.1.2. Minimum subpattern length conversion, RZ example

The "Unconverted" PI-5800 program shown below has channels set to RZ format:

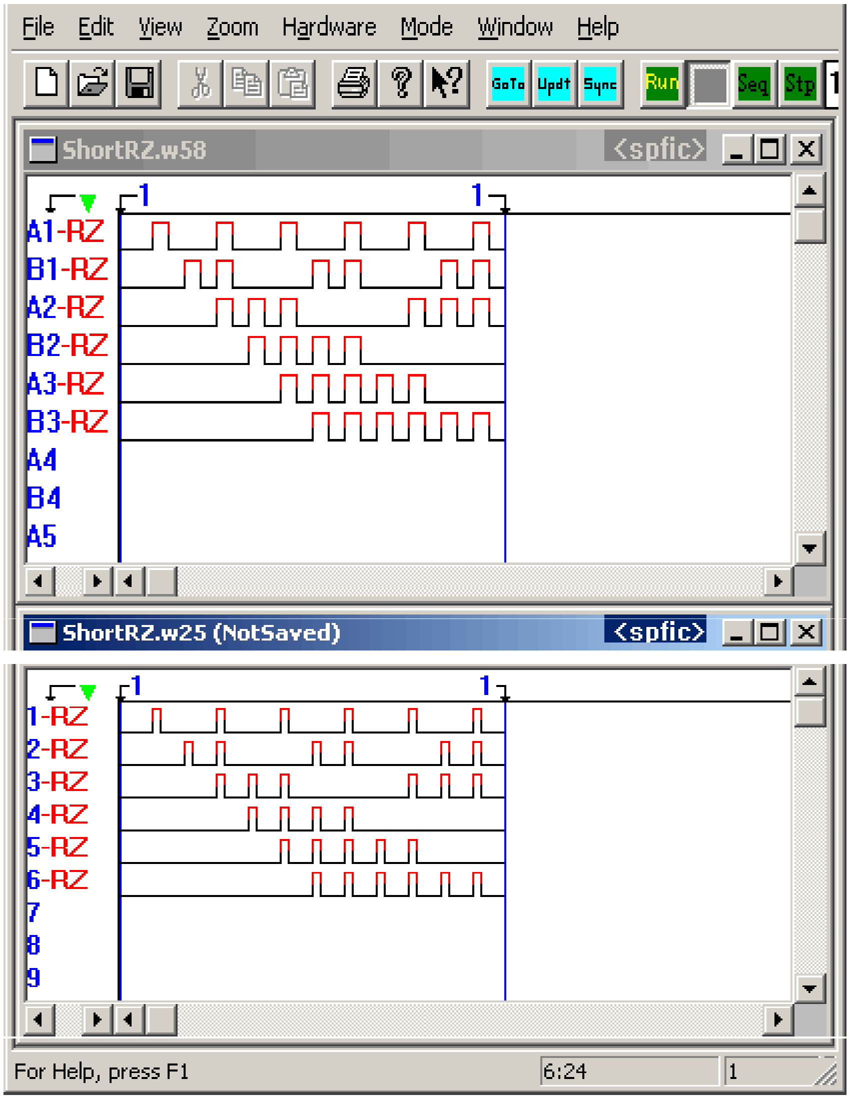


Figure 6: Converted PI-5800A file with "illegal" 12-bit subpattern definition in RZ format

Therefore the "Converted" program has a new zero bit inserted after every original bit. If this program is run at 100 MHz, it will have output identical to that of the original program run at 50 MHz on a PI-5800A. Note that this can also be verified by examining the original program at **2x** zoom using the **View** menu.

This conversion can accomplished several ways, depending on the user's preference:

- Using PI-CONVERT for batch-converting large numbers of text or GPIB files
- In PI-PAT, by loading a PI-5800 pattern file and performing a "**Save As**" command into PI-2005 format
- In PI-PAT, by manually editing the bit patterns and clock settings
- Via string-manipulation routines in custom software

2.2. Overlapping subpattern definitions

While the PI-5800A supported overlapping subpattern definitions, the PI-2005 does not. Each bit in a PI-2005 file must belong to one and only one subpattern.

Existing programs that violate the overlapping subpattern constraint can be converted to run on a PI-2005 using PI-PAT or PI-CONVERT.

PI-PAT and PI-CONVERT avoid violating the overlapping subpattern constraint by duplicating the pattern data of the second subpattern into a free area of memory, and then re-defining the second subpattern to point to the new data.

2.2.1. Overlapping subpattern conversion example

In the unconverted PI-5800A program shown below, bits 15-20 are common to SP 1 and SP 2:

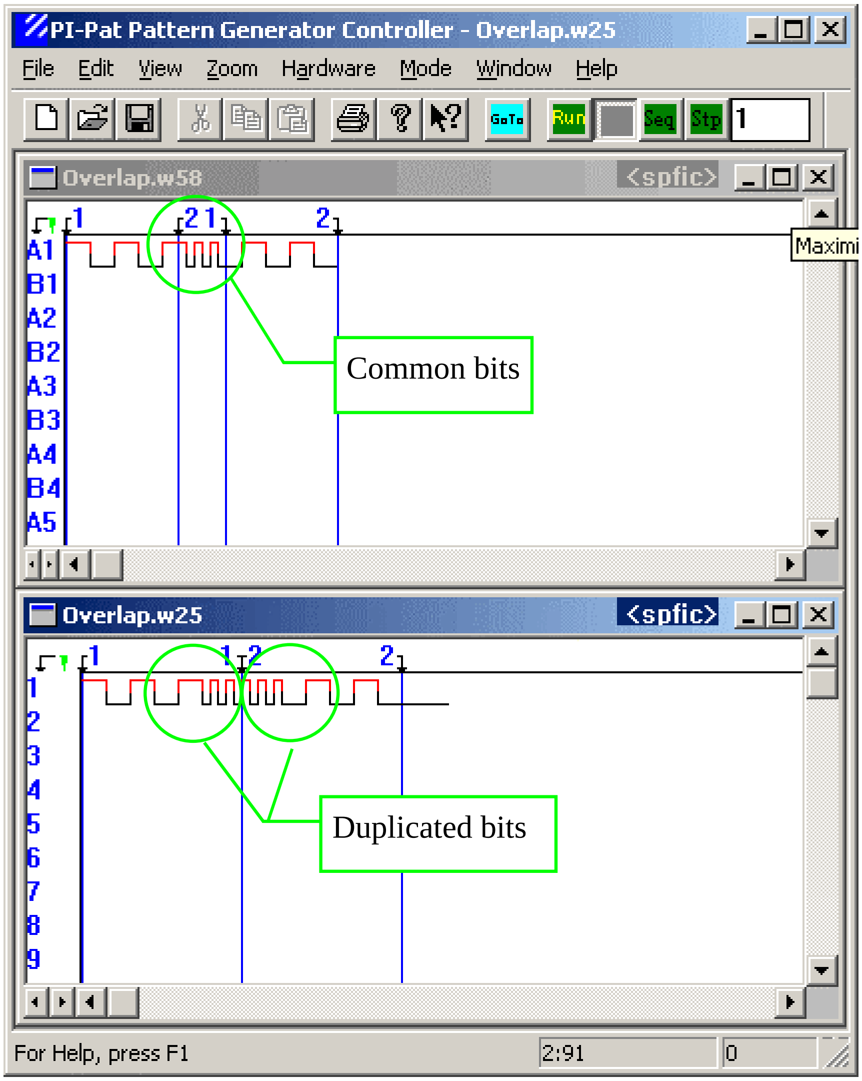


Figure 7: Converted PI-5800A file with overlapping subpatterns duplicated

After conversion, the pattern data for SP 2 has been duplicated and placed at bit positions 20-40, and the definition of SP 2 has been changed appropriately.

When run, this program will produce output identical to the original program.

This conversion can be accomplished several ways, depending on the user's preference:

- Using PI-CONVERT for batch-converting large numbers of text or GPIB files
- In PI-PAT, by loading a PI-5800 pattern file and performing a "**Save As**" command in PI-2005 format
- In PI-PAT, by manually copying the bit patterns and re-defining subpatterns
- Via string-manipulation routines in custom software

2.2.2. Overlapping subpattern editing example

When converting overlapping subpatterns, users should take note that the pattern data that was formerly common to both subpattern definitions is now in two places. Any change to pattern data that is intended for both subpatterns must be changed in both places.

For example, a user wanting to edit bits 15-20 of the original PI-5800 file would edit those bits in one place, and the changed output would be reflected in SP 1 and in SP 2. In the converted file, that change must be made in bits 15-20 and in bits 20-25.

2.3. Maximum programmable delay

The maximum programmable delay on the PI-5800A is 32 ns. The maximum programmable delay supported on the PI-2005 is 6.6 ns.

Because real-world latencies vary among systems, programmable delays should be set by each customer according to the particular requirements of the system being used. If PI-PAT converts a PI-5800 file (.w58) to a PI-2005 file (.w25), it will set all delays to 0.0 ns.

If PI-CONVERT processes a PI-PAT file that contains delays, PI-CONVERT will set all delays to 0.0 ns. If PI-CONVERT processes a GPIB command file containing DELAY commands, PI-CONVERT will comment-out those command lines and make notations in the conversion log.

2.4. GPIB Command Set Summary

The following sections describe differences in the GPIB command sets supported by the PI-5800 and PI-2005. For a complete description of GPIB control, including syntax and examples for each command, please see the **PI-2005 Operators Manual**.

2.4.1. PI-5800A Commands Supported Without Modification

The PI-2005 implements the following commands with the same syntax and arguments as the PI-5800A. Commands listed in this section may also support additional syntax and arguments, but existing programs written for the PI-5800A will work without modification, subject to the limitations listed above.

COMMAND	SHORT DESCRIPTION
:hex data	Loads hexadecimal data.
(binary data) * x	Loads binary/algorithmic data.
; remarks	Program remarks ignored by Pattern Generator.
BEGIN	Enters Begin instruction line.
CHANNEL	Sets output format of channel x to INVERT, NONINV, RZ, NRZ, HIZ or NORM.
CLEAR	Clears memory and resets data generator. Extension defines SP, INST, CHANNEL, FORMAT or all settings.
COMPILE	Selects/compiles program to run.

DEFINE SP	Defines subpattern.
DELAY	Sets optional programmable delay.
DELETE	Deletes a subpattern definition.
DUMP	Dumps data and settings from Pattern Generator back to controller.
GOTO	Points at channel or bit position to begin loading data.
ID	Returns identification/configuration information to controller.
LOAD CHANNEL	Selects channel to be loaded with data.
NUMCHAN	Returns number of channels installed in Pattern Generator.
PAUSE	Pauses Pattern Generator and preserves program counter.
PERIOD	Sets the internal clock period.
REPEAT	Enters Repeat on instruction line.
SP	Enters Subpattern instruction line.
STATUS	Returns error state and run-mode, and clears pending errors.
UPDATE	Sends data into pattern memory.

2.4.2. GPIB Command Mismatches for PI-2005

The following GPIB commands are implemented slightly differently in the PI-2005.

COMMAND	DIFFERENCE
RUN	The PI-2005 will always RUN from the beginning of the program. The PI-5800 will RUN from the last location of the program counter.
STEP	Steps through instruction sequence a specified number of subpatterns at a time. The PI-5800 would step through a specified number of clock periods.
STOP	The PI-2005 resets to the first instruction line. This would replace the STOP followed by COMPILE x commands in the PI-5800. The PI-5800 preserves the location of the program counter when STOP is received.

2.4.3. New PI-2005 GPIB Commands

The following new commands were not available in the PI-5800A. Please see the **GPIB Function Reference** section of the **PI-2005 Operators Manual** for details on using these functions.

COMMAND	FUNCTION
CLK	Selects internal clock or external clock with ECL/TTL thresholds.
GETDELAY	Returns presence/absence of optional delay feature.
POWERDOWN	Prepares the PI-2005 for powering down.
RUNMODE	Requests the run mode condition. Responses are: “Stopped”, “Running”, “WaitSeq”, “WaitStep”, “WaitStop”, “WaitRun”

SEQUENCE	Runs through the instruction set a specified number of times
SETGPIBADDR	Assigns new GPIB address to PI-2005.
STASTP	Enables/Disables Start/Stop from rear panel.

2.4.4. GPIB Command Not Supported in PI-2005

The following GPIB commands are not supported in the PI-2005. PI-CONVERT will comment-out any lines containing these commands and append notes to the conversion log. Please see the **GPIB Function Reference** section of the **PI-2005 Operators Manual** for details on implementing any workarounds suggested below.

COMMAND	FUNCTION/WORKAROUND
BRANCH	No conditional branching in PI-2005.
C FLAG	No conditional branching in PI-2005.
CASCADE	64 Kb memory depth per channel replaces need for Cascade mode. Cascade mode and CASCADE command not supported.
CYCLE	Implemented as SEQUENCE
DEFINE CHANNEL	Tri-State-On-The-Fly replaces mixed-mode in PI-2005. DEFINE CHANNEL not supported.
DYNAMIC UPDATE	Not implemented in PI-2005.