April 2007

AMISR SSPA Automated Test Suite

Jeffrey S. Pelligrino  
*Worcester Polytechnic Institute*

John Fredrik Scimone  
*Worcester Polytechnic Institute*

Kaushal Shrestha  
*Worcester Polytechnic Institute*

Follow this and additional works at: https://digitalcommons.wpi.edu/mqp-all

Repository Citation


This Unrestricted is brought to you for free and open access by the Major Qualifying Projects at Digital WPI. It has been accepted for inclusion in Major Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact digitalwpi@wpi.edu.
AMISR SSPA
AUTOMATED TEST SUITE

Project Report

A Major Qualifying Project
submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
And performed at SRI INTERNATIONAL
in partial fulfillment of the requirements for the
Degree of Bachelor of Science

Date : March 2, 2007

WPI Advisor:
Professor John A. Orr

SRI Advisor:
Andrew Young

Submitted By:
Jeff Pelligrino
John Scimone
Kaushal Shrestha
Table of Contents

Table of Contents ........................................................................................................... ii
Table of Figures ................................................................................................................. iv
List of Tables ...................................................................................................................... v
List of Tables ...................................................................................................................... v
Abstract ............................................................................................................................. vi
Executive Summary .......................................................................................................... vii
1. Introduction ................................................................................................................... 1
2. Background ................................................................................................................... 2
   2.1. National Science Foundation .............................................................................. 3
   2.2. Advanced Modular Incoherent Scatter Radar .................................................... 3
3. Project Goals .................................................................................................................. 5
4. SSPA Design Specifications ......................................................................................... 6
5. Technical Background ................................................................................................. 7
   5.1. Phase 1: Calibration ............................................................................................ 7
   5.2. Phase 2: Pre Burn-in Test .................................................................................. 8
   5.3. Phase 3: 24-hr Burn-in Test .............................................................................. 8
   5.4. Phase 4: Post Burn-in Test .............................................................................. 9
   5.5. Phase 5: Design Verification Test ....................................................................... 9
6. Automated Test Suite ................................................................................................. 11
   6.1. System Design & Integration ............................................................................ 13
      6.1.1. Signal Generator ........................................................................................ 15
      6.1.2. Power Supply ............................................................................................. 15
      6.1.3. RF Power Meter & Power Head ................................................................. 17
      6.1.4. Switch Box ................................................................................................ 19
      6.1.5. Industrial PC .............................................................................................. 20
      6.1.6. Control Box ................................................................................................ 23
      6.1.7. Rack Design .............................................................................................. 30
   6.2. Platform ................................................................................................................ 32
      6.2.1. Hardware Architecture ............................................................................... 32
      6.2.2. Operating System ...................................................................................... 33
      6.2.3. Programming Language .......................................................................... 34
      6.2.4. Database Management System (DBMS) ................................................... 35
6.3. Software Architecture ............................................................................................. 36
   6.3.1. Object Oriented Program Design ................................................................. 37
   6.3.2. Device level Abstraction .............................................................................. 37
   6.3.3. Device Interfacing ....................................................................................... 37
   6.3.4. Class Modules ............................................................................................. 40
   6.3.5. Flow Control ............................................................................................... 46
   6.3.6. User Interface ............................................................................................... 52
   6.3.7. Exceptions and Error Handling ................................................................... 53
   6.3.8. Database Programming and Data Storage ..................................................... 54
7. Twenty-Four Hour Burn-In System ............................................................................. 58
   7.1. System Requirement ......................................................................................... 58
   7.2. System Design .................................................................................................... 58
   7.3. Power Supplies ................................................................................................. 59
   7.4. Loads .................................................................................................................. 59
   7.5. Signal Sources .................................................................................................. 59
   7.6. Signal Distribution System ............................................................................... 60
8. Results ......................................................................................................................... 62
9. Conclusion ................................................................................................................. 63
References ......................................................................................................................... 63
Appendix A ....................................................................................................................... A-1
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Artists concept of an AMISR radar with two faces</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Block Diagram of Test System</td>
<td>12</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Test Station System Block Diagram</td>
<td>14</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Agilent 5181A Signal Generator</td>
<td>15</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Agilent N6700A Power Supply</td>
<td>16</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Agilent E4417A Power Meter and E9323A Power Heads</td>
<td>18</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Agilent 87106A Load Switch</td>
<td>19</td>
</tr>
<tr>
<td>Figure 8</td>
<td>National Instruments 488.2 GPIB Card</td>
<td>21</td>
</tr>
<tr>
<td>Figure 9</td>
<td>National Instruments PCI-6220 DAQ Card</td>
<td>23</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Input Stage Couplers</td>
<td>24</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Output Stage Coupler</td>
<td>25</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Control Box System Diagram</td>
<td>25</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Control Box</td>
<td>26</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Front Panel of Control Box (Artist Concept)</td>
<td>26</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Front Panel of Control Box</td>
<td>26</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Back Panel of Control Box (Artist Concept)</td>
<td>27</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Back Panel of Control Box</td>
<td>27</td>
</tr>
<tr>
<td>Figure 18</td>
<td>7th order RF low pass filter</td>
<td>28</td>
</tr>
<tr>
<td>Figure 19</td>
<td>S-parameters vs frequency</td>
<td>29</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Rack Layout</td>
<td>31</td>
</tr>
<tr>
<td>Figure 21</td>
<td>System Wiring Diagram</td>
<td>32</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Plot of Output Power Trace (dBm) vs. Time (ms)</td>
<td>42</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Plot of Output Power vs. number of samples over 2ms pulse</td>
<td>43</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Plot of in-phase and quadrature-phase components of the phase of the RF output signal</td>
<td>45</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Plot of in-phase, quadrature-phase, pulse modulation signal and phase in radians</td>
<td>46</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Main Program Flow</td>
<td>47</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Calibration Process Flow</td>
<td>48</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Pre Burn-in Process Flow</td>
<td>49</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Post Burn-in Process Flow</td>
<td>50</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Power Meter Offset Calibration Flow</td>
<td>51</td>
</tr>
<tr>
<td>Figure 31</td>
<td>Terminal based User Interface</td>
<td>52</td>
</tr>
<tr>
<td>Figure 32</td>
<td>Screenshot of the Manual Controls GUI</td>
<td>53</td>
</tr>
</tbody>
</table>
List of Tables

Table 1: Comparison of Signal Generators ................................................................. 15
Table 2: Comparison of Power Supplies ................................................................. 16
Table 3: Power-meter specifications comparison .................................................. 17
Table 3: Power-head specifications comparison ..................................................... 18
Table 5: Load switch specification comparison ..................................................... 19
Table 5: Various industrial PC's and their respective specifications ...................... 20
Table 6: Comparison of different GPIB cards ......................................................... 21
Table 7: Comparison of different Digital I/O cards .............................................. 22
Table 8: Additional parts for the Digital I/O cards ................................................ 23
Table 9: Sample Calibration Data .......................................................................... 62
Table 10: Sample Pre Burn-in Data ........................................................................ 62
Abstract

SRI International has been developing an Advanced Modular Incoherent Scatter Radar (AMISR) for the purpose of observing upper-atmospheric phenomena with an easily deployable modular radar system. This project was mostly concerned with the power amplifier in the AMISR, specifically a Solid State Power Amplifier (SSPA). Our project was to design and build an automated test station for use during production testing and design verification testing as well as design circuitry that will be used during a twenty-four hour burn-in test.
Executive Summary

The National Science Foundation is funding the construction of a radar system to be built in Alaska called AMISR, Advanced Modular Incoherent Scatter Radar. SRI International, a non-profit research organization based in Menlo Park, California, is responsible for the design and construction of the AMISR. The radar is unusual because it is made up of a planar array of emitting elements. By controlling the phase relationships between the elements, the radar beam can be instantaneously repositioned without the use of mechanical systems. The AMISR project is slated to use 4,096 Solid State Power Amplifiers (SSPAs). SRI is currently redesigning the amplifier and production will be starting in the spring of 2007.

During production, each SSPA needs to be calibrated and tested to ensure that each meets the specifications. Transporting equipment to the radar’s location is difficult and costly because the deployment location for the AMISR is Poker Flats, Alaska. In June of 2007, a barge will be departing for Alaska and all equipment for the AMISR must be on board or else it will delay the project for another year. Since this does not leave much time for the production and testing of the SSPAs, SRI needs an automated test station that is capable of calibrating and performing a specification test on an amplifier in five minutes without unnecessary human interaction. The test station must be simple to use, self-calibrating, portable, and able to store test data in an easily accessible database.

After the initial calibration, each unit will be tested to ensure it meets the specifications set forth by SRI. These tests include power measurements at multiple frequencies, amplifier efficiency measurements, and an input power variation test. In order to conduct all required tests, an integrated system needed to be built. The system became known as the Automated Test Suite (ATS). It required the implementation of a power supply, signal generator, power meter, I/Q detector, and a computer to control the test equipment as well as provide the user interface.

A control box had to be designed to provide a signal control center for interfacing multiple devices. The control box distributes the timing signals generated by the
computer’s data acquisition card (DAQ) to various components. It also interfaces the I/Q detector with the analog inputs of the DAQ, contains the directional couplers and attenuators that are needed to make power measurements, as well as the switch that changes the load that is connected to the SSPA.

The I/Q detector consists of Analog Device’s AD8347 demodulator chip implemented using its evaluation board. The I/Q detector demodulates the in-phase and quadrature phase components of a signal by mixing a reference signal with the output RF signal. When the two signals are mixed, the output’s frequency domain has peaks at DC and 880MHz. Since the DC signal is the one that represents the phase of the SSPA, we designed an RF filter to eliminate the 880MHz signal. The I/Q Detector was also modified to disable the automatic voltage offset compensation feature.

A significant amount of time was spent developing code to intelligently calibrate and test the amplifiers without any human intervention because the station needed to be automated. Several software-related obstacles needed to be overcome during the project. The drivers for the DAQ card and the GPIB card had to be installed on Linux without manufacturer support. In addition, interfacing Python with the GPIB card and DAQ card was difficult because Python was an unfamiliar language. A Python package called PyVISA, which is a wrapper for National Instrument’s GPIB control libraries, was used to interface the GPIB card. For the DAQ card, a custom wrapper was written in C to interface the Python code with the hardware because there were no preexisting wrappers. After overcoming these difficulties, all that remained was the task of writing the software to drive the system and store measurements and results into a database.

We undertook the additional responsibility of designing and developing the burn-in system because the original project was ahead of schedule. The burn-in system is required to run ninety-six amplifiers for a twenty-four hour period. During this time, the SSPAs need to be monitored to make sure they are on for the full length of the test. The burn-in process tests the units for component failures prior to installing the amplifiers in the field. A single signal generator is used to supply the necessary RF input signal for operating all ninety-six amplifiers concurrently. The burn-in system is designed in two
stages. The first stage splits the input RF and timing signals six ways while the second stage splits each output from stage-one sixteen ways. To make operation of the burn-in system easier for the user, stage-two had to be designed with latching circuitry to indicate if an SSPA’s alarm signals tripped during the test. As an added feature, there is also an indicator to show when an amplifier has been tested for twenty-four hours. A circuit board was designed to meet the requirements of the second stage of the burn-in system. Revision one was fabricated, populated, and troubleshooting with modifications made to demonstrate the required changes for revision two. The modifications were replicated in the schematics and the revised design has been submitted for layout changes.

The Automated Test Suite project has been successfully completed. The station is physically complete and the software has been fully implemented. Since production of the SSPAs has not started, new production tests are being developed and the DVT test suite may be altered in the future. The burn-in system, which was assigned as a secondary project, is not yet complete. Stage-one has been tested and is complete, however, the stage-two circuit board has entering its second revision and will be completed by the staffs at SRI International.
1. Introduction

SRI International, a non-profit research and development organization, is in the process of developing an Advanced Modular Incoherent Scatter Radar (AMISR) for the purpose of observing upper-atmospheric phenomena with an easily deployable modular radar system. The AMISR uses phased scanning principles allowing quick redirection of the scanning lobe without any physical movements of the antenna elements. The AMISR consists of antennas, power amplifiers, low noise amplifiers, delay shifting circuitry, control circuitry, a power supply system, and a chassis to hold the equipment together. The power amplifier in the antenna element unit (AEU), specifically a Solid State Power Amplifier (SSPA), is what this project was mostly concerned with because the SSPA is being redesigned and the units will need to be tested to ensure their proper functionality. Our project was to design and build an automated test station for this purpose as well as design circuitry that will be used during a twenty-four hour burn-in test.
2. Background

Founded in 1946 by a group of west coast industrialists and Stanford University, SRI remains a research institution committed to the discovery and application of new technology in fields ranging from communications to the biomedical industry. In 1970 SRI separated from Stanford University and in 1977 formally changed its name from Stanford Research Institute to SRI International. The company holds more than 1,000 patents worldwide. There are currently over 1,400 people employed by SRI International alone. In addition, over 600 people are employed at its subsidiary Sarnoff Corporation. Their main offices are located in Menlo Park, CA however they have additional offices located in Washington, DC and Tokyo, Japan as well as several other US locations.

In an effort to move technology developed at SRI to the marketplace, SRI International has worked in conjunction with their subsidiary Sarnoff Corporation and top-tier investment and venture capital firms to form approximately two dozen new ventures. These ventures include companies such as Artificial Muscle Inc, Bridge Pharmaceuticals, Spanlink Communications, and Intuitive Surgical Inc. Sarnoff Corporation has won 10 Emmy awards and developed the HDTV standard used in the US. SRI has also won an Academy award and is recognized for developing the current automated check processing system that utilizes magnetic ink coding, the world’s first computer network which was known as ARPANET, and the first prototype of a computer mouse.

SRI International is expanding rapidly. They were recently awarded $56.9 million from the National Institute of Allergy and Infectious Diseases for research and development of drugs and antibodies for anti-infective therapeutics. SRI International is also opening a new research facility in St. Petersburg, FL for marine technology research. The facility will research and develop technologies related to ocean science, the maritime industry, and port security.
2.1. National Science Foundation

In 1950, congress created the National Science Foundation, an independent federal agency, to encourage the advancement of science, national health and prosperity, and to improve national security. The government’s reliance on innovation and scientific progress intensified during the World War II. Following the war, government involvement in universities and science was at an all-time high. At this time, several congressmen and scientists pushed for legislature to create an agency to fund scientific research through government grants, this agency became known as the National Science Foundation (NSF).

Today, the NSF is the funding source for approximately twenty percent of all federally sponsored university research. The agency also funds high-risk, high-payoff ideas that are on the bleeding edge of innovation. An example from the National Science Foundation’s website is nanotechnology. They were funding scientists who were researching ways to manipulate movement on an atomic level, years before the public had even heard of nanotechnology.

In addition to funding research, the NSF also helps finance high-cost equipment and facilities that are too expensive for one research group or researcher. This includes “giant optical and radar telescopes, Antarctic research sites, high-end computer facilities and ultra-high-speed connections, ships for ocean research, sensitive detectors of very subtle physical phenomena and gravitational wave observatories.” One of the projects that the NSF has funded for the past 10 years is the AMISR project.

2.2. Advanced Modular Incoherent Scatter Radar

The Advanced Modular Incoherent Scatter Radar, AMISR, is a mobile radar facility that will eventually be utilized by scientists to study the atmosphere and observe space weather events. The project is funded by the National Science Foundation and is being developed in a collaborative effort led by SRI International. SRI is responsible for
the lead design and construction of the radar as well as overseeing operations and the initial design verification tests. Sanmina-SCI and VECO Alaska are manufacturing the antenna element units and overseeing the structural engineering of the radar respectively. This radar will be built using a phased-array antenna system that will allow the radar to function in different configurations. The system will be controlled remotely and the radar beam will be electronically controlled allowing scientists to instantaneously position the beam to accurately measure changing weather events. The AMISR utilizes three separate radar faces, each consisting of 128 panels, which can be deployed in up to three separate locations. In Figure 1, the AMISR radar is pictured with two full faces. The first radar face is being constructed in Poker Flat, Alaska and subsequent faces will be constructed in Resolute Bay, Canada, and Nunavut, Canada. Each face of the radar is made up of 8 groups of panels. Each group has 16 panels and therefore each face has 128 panels. There can be up to 32 solid state power amplifiers (SSPA) per panel. Therefore, each face can have up to 4,096 individual transmit and receive elements.

Figure 1: Artists concept of an AMISR radar with two faces.
3. Project Goals

The goal of this project was to design and build an automated test station for the SSPA. The test station should be able to calibrate the amplifier, and then test the RF characteristics of the unit such as power gain, current draw, amplitude, phase, and VSWR at the upper and lower signal frequency limits as well as the middle operating frequency. Following the RF characteristic tests, the test station will check the fault conditions of the amplifier, such as an open output, and a mismatched load. The automated test station must also be capable of powering an SSPA during environmental testing, including periodic measurements of power, current, ambient temperature, and device temperature.

The automated test system must be capable of testing the RF characteristics of about 1000 SSPAs per month at the rate of about forty per day. Testing will be conducted twenty-three days per month and testing will take place during an eight hour shift on these days. The expected yield rate is approximately 90% based on information provided by SRI and therefore we will be designing our test system to test each SSPA in less than five minutes. For the twenty-four hour burn-in process, which will not utilize the ATS, our goal was to design a system to test multiple SSPAs during a single twenty-four hour period.
4. SSPA Design Specifications

The SSPAs are designed to meet a number of important specifications. The Automated Test Suite is designed to test these specifications to ensure that each SSPA meets the designated requirements. The requirements include:

Full Operating Temperature:
- -40 Degrees C to 35 Degrees C

RF Power:
- Input: 10dBm ± 1.5dB
- Output: 57dBm (-0/+.5dB)

RF Pulse Characteristics:
- Minimum Pulse Width: 1µs
- Maximum Pulse Width: 2ms
- Amplitude Droop: <10% at maximum pulse width

Phase Response:
- Unit-to-Unit match: ± 5 degrees

Stability/Impedance:
- Input VSWR : < 1.3:1
- Output VSWR: < 1.5:1

Protection:
- Over-drive protection if input exceeds specified range
- VSWR mismatch protection when VSWR is greater that 6:1
- Over-temperature protection

Each SSPA will also be subjected to a twenty-four hour burn-in test that will not utilize the ATS. After the burn-in procedure, each SSPA will be retested on the ATS to ensure that it is still functioning properly.
5. Technical Background

When a company produces a device in large quantities, it is important to ensure that each product meets the company’s standards. Production testing allows companies to determine if a device meets its requirements before the device is sold or delivered. SRI needs to conduct production testing on the SSPA’s to ensure that they meet their specifications. Test data from production testing provides many useful statistics to companies like SRI. Test data allows companies to determine if there are reoccurring problems or manufacturing trends that could potentially affect future products. There are five phases of testing for the AMISR SSPAs.

- Phase 1: Calibration
- Phase 2: Pre Burn-in Test
- Phase 3: 24-hr Burn-in Test
- Phase 4: Post Burn-in Test
- Phase 5: Design Verification Test

Four phases involve the ATS; they include calibration, pre burn-in testing, post burn-in testing, and the design verification testing. The third phase, the twenty-four hour burn-in process, does not utilize the ATS, however, it utilizes the burn-in distribution system which will be discussed later in this report. It is important to note that only the first four phases are included in production testing. The design verification testing (DVT) will not be performed on every SSPA; only select number of amplifiers will be subjected to DVT.

5.1. Phase 1: Calibration

Phase 1 of production testing is used to calibrate each SSPA prior to design specification testing. During calibration, the VSWR, output power, and phase of each SSPA is calibrated and set to ensure proper matching between SSPAs. A list of steps for the automated calibration phase can be found below.

- Initialize Devices
• Initialize Digital Potentiometers
• Set Driver Bias
• Set Output Bias
• Set Gain
• Set Phase
• Set VSWR

5.2.  Phase 2: Pre Burn-in Test

After the calibration procedure, pre burn-in testing is conducted to further verify the device meets the design requirements. There are five tests performed during pre burn-in testing. The output power and current are verified at three frequencies at the beginning of the test. An over pulse width test is conducted to verify that each SSPA automatically limits the pulse width to less than 2.5ms when the pulse width is outside its specifications. The fifth test verifies that the amplifier gain is consistent when the input is decreased to 8.5dBm and increased to 11.5dBm. The various tests for pre burn-in are listed below.

• Output Power & Current Verification: 440MHz
• Output Power & Current Verification: 430MHz
• Output Power & Current Verification: 450MHz
• Over Pulse Width Test
• Input Power Variation Test
• Efficiency Test
• Open Circuit Test
• VSWR 3:1 Trip Test

5.3.  Phase 3: 24-hr Burn-in Test

The next phase is the twenty-four hour burn-in test which consists of running ninety-six SSPAs at once while monitoring the two status bits—over-temperature and VSWR fault. Each amplifier that passes the twenty-four hour burn-in process is retested during post burn-in testing. The twenty-four hour burn-in test utilizes the distribution board that was designed to simplify the wiring of the system, monitor the two status bits, and to alert a user when each SSPA has run for the full twenty-four hours.
5.4. Phase 4: Post Burn-in Test

During phase 4 of production testing, the same tests that were completed during phase 2 will be completed again. This will allow engineers to determine if the twenty four hour burn-in test affected the performance characteristics of each SSPA. The various tests for post burn-in are listed below.

- Output Power & Current Verification: 440MHz
- Output Power & Current Verification: 430MHz
- Output Power & Current Verification: 450MHz
- Over Pulse Width Test
- Input Power Variation Test
- Efficiency Test
- Open Circuit Test
- VSWR 3:1 Trip Test

5.5. Phase 5: Design Verification Test

One out of every sixty-four SSPAs will be subjected to design verification testing. The design verification test package in the ATS system includes a qualification test, pre-environmental test benchmark, post-environmental test benchmark, and an operating test. The design verification test plan entails an initial qualification test and then a series of environmental tests. DVT includes many environmental tests, some environmental tests will be conducted when the amplifier is operation and others will be conducted when the amplifier is in non-operational mode. The operating test will be used to collect data during environmental tests that include an operating amplifier. The pre-environmental test will be conducted before an amplifier is subjected to a non-operational environmental test and a post-environmental test will be conducted after a non-operational environmental test. The design verification test plan is shown below.

**Qualification Test:** (Performed at 430, 440, and 450MHz for each SSPA)
- Power Consumption (Input Current and Input Voltage)
- RF Power Output
- Amplifier Gain
- RF Pulse Droop (2ms Pulse Width)
- Amplifier Insertion Phase
Environmental Tests:
- Low & high temperature stress testing (SSPA Operational).
- Random vibration stress testing (SSPA Operational).
- Temperature Cycling with & without vibration step (SSPA Operational).
- Temperature Cycling with Condensation (SSPA non-operational).
- Random Vibration (SSPA non-operational).
- Half sine shock test (SSPA non-operational).
- Low Temperature Soak Test – 22hr duration (SSPA operational).
- High Temperature Soak Test – 22hr duration (SSPA operational).
- Temperature Cycling without Condensation – 48hr duration (SSPA operational).
- Humidity Cycling without Condensation – 96hr duration (SSPA operational).
6. Automated Test Suite

The automated test suite (ATS) was designed to simplify the testing of the SSPAs. The ATS in used in the following phases of production testing:

- Calibration
- Pre Burn-in Testing
- Post Burn-in Testing
- Design Verification Testing

The ATS has been designed to meet several specifications. These specifications are listed below.

- Conduct Calibration and Pre Burn-in Test in less than 5 minutes.
- Must be capable of measuring phase, power, and current.
- ATS must be capable of being relocated easily.
- Easy to modify calibration tolerances.
- System must be self-calibrating.
- System must be network accessible.
- Data must be stored in a database.
- Data must be easily accessible.
- Simple to use.

The block diagram shown in Figure 2 summarizes the test system setup.
Figure 2: Block Diagram of Test System
6.1. **System Design & Integration**

The ATS includes a number of major components, all of which are integrated within our stand alone system. These include:

- Industrial PC
- RF power meter and power heads
- Power supply
- GPIB card
- Digital I/O card
- Signal generator
- Control Box
- Switch Box

The system diagram in Figure 3 shows how these different resources are connected and implemented. As you can see in the diagram, the industrial PC controls the test station. It communicates with various components through GPIB, Ethernet, and through the digital I/Os of the DAQ. The components that are controlled by the PC include the power supply, power meter, signal generators, and DAQ card. Each of these components connect to the SSPA, with the exception of the power meter which connects to the SSPA via the power heads. A switch box is also connected to the output of the SSPA and is used to switch between various loads.
Figure 3: Test Station System Block Diagram
6.1.1. Signal Generator

The signal generator is used to produce our test signal. The frequency that the RF amplifiers will operate, between 430 MHz and 450 MHz, is within the limits of many of the signal generators we found which are shown in Table 1. It is important that the signal generator is programmable because we need to control it using our industrial PC. The Agilent N9310A signal generator was chosen because of its short lead-time, low-cost, and it is programmable via ethernet.

<table>
<thead>
<tr>
<th>Company</th>
<th>Model #</th>
<th>Price</th>
<th>Lead Time</th>
<th>Programmable</th>
<th>Pulse Modulation Built-in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agilent</td>
<td>E4400B ESG-A</td>
<td>$8,162</td>
<td>8 weeks</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Agilent</td>
<td>N9310A</td>
<td>$6,810</td>
<td>4 weeks</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Agilent</td>
<td>N5181A</td>
<td>$6,349</td>
<td>4 weeks</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1: Comparison of Signal Generators

Figure 4: Agilent 5181A Signal Generator

6.1.2. Power Supply

The DC power supply is an important component of the test station. The power supply must be capable of covering the operating voltage and current ranges and must be able to be programmed remotely to allow automation of the testing procedure. This means that there must be a USB, Ethernet, or GPIB interface for programming. During the calibration and normal operation of the SSPA, the supply voltage needs to be 32V. Another important feature for the power supply is the current sensing resolution. The power supply must be able to supply 5A maximum current and should have a high
current sensing resolution because the current draw of the SSPA will be measured by the supply. Below is a comparison of the features of different units and their prices.

<table>
<thead>
<tr>
<th>Mfr</th>
<th>Model #</th>
<th>Voltage</th>
<th>Current</th>
<th>Power</th>
<th>Programming Interfaces</th>
<th>Current Measure Res.</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agilent</td>
<td>N5746A</td>
<td>40V</td>
<td>19A</td>
<td>760W</td>
<td>GPIB, LAN, USB 2.0</td>
<td>57mA</td>
<td>$2,442.11</td>
</tr>
<tr>
<td>Agilent</td>
<td>6653A</td>
<td>35V</td>
<td>15A</td>
<td>500W</td>
<td>GPIB</td>
<td>15mA</td>
<td>$3,434.69</td>
</tr>
<tr>
<td>Amrel</td>
<td>PD 40-25A</td>
<td>40V</td>
<td>25A</td>
<td>1kW</td>
<td>RS-232, GPIB</td>
<td>.2%</td>
<td>$3,833.00</td>
</tr>
<tr>
<td>Amrel</td>
<td>PD 40-25E</td>
<td>40V</td>
<td>25A</td>
<td>1kW</td>
<td>RS-232, GPIB, LAN</td>
<td>.2%</td>
<td>$4,528.00</td>
</tr>
<tr>
<td>Agilent</td>
<td>N6700A+ Filler Panel Kit</td>
<td>60V</td>
<td>5A</td>
<td>300W</td>
<td>GPIB, LAN, USB 2.0</td>
<td>.15%</td>
<td>$3,699.30</td>
</tr>
<tr>
<td>Agilent</td>
<td>N6700B+ Filler Panel Kit</td>
<td>35V</td>
<td>9A</td>
<td>300W</td>
<td>GPIB, LAN, USB 2.0</td>
<td>.15%</td>
<td>$3,699.30</td>
</tr>
</tbody>
</table>

Table 2: Comparison of Power Supplies

The Agilent N6700A power supply was chosen because it is capable of supplying the voltage and current necessary and it is also programmable via ethernet. In addition to these features the power supply has a current measurement resolution suitable for our application.

![Figure 5: Agilent N6700A Power Supply](image)
6.1.3. RF Power Meter & Power Head

We require RF Power-meters to be able to measure the input power into and the output power out of the power amplifier. The output power of the amplifier is expected to be approximately 500W of peak power or +57dBm. It is not easy to measure very high power directly; hence we will have to go through the process of attenuation before power measurements. Some of the specifications that we are looking in terms of power-meters and power-heads are:

- High sampling rate
- Peak power measurement capability
- High video bandwidth
- Wide power measurement range

We needed a high sampling rate because we need to perform a droop test. This involves capturing a trace of the output power and measuring the droop associated with the pulse. The peak power measurement capability is required because the RF pulse is only of 10% duty cycle which can output a maximum of 500W power or 50W average power. The high video bandwidth is required to measure sharp rising edge triggers. The wide power measurement range is an optional specification for flexibility. The tables below (Table 3 and Table 4) summarize the various power meter products and power-head products that are able to fulfill our technical requirements.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Model</th>
<th>Measurement</th>
<th>Sampling rate /Measurement rate</th>
<th>Bandwidth</th>
<th>Price US$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agilent</td>
<td>E4417A</td>
<td>Peak, Average, Peak-to-average ratio</td>
<td>20 Msample/s</td>
<td>7,077 MHz to 40 GHz</td>
<td>7,077</td>
</tr>
<tr>
<td>Agilent</td>
<td>N1912A</td>
<td>Peak, average, Peak-to-average ratio, rise time, fall time and pulse width</td>
<td>100 Msample/s</td>
<td>50 MHz to 40 GHz</td>
<td>10,214</td>
</tr>
<tr>
<td>Gigatronics</td>
<td>8540C</td>
<td>Peak, average and CW</td>
<td>500-4000 reading/s</td>
<td>100 kHz to 40 GHz</td>
<td>N/A</td>
</tr>
<tr>
<td>Gigatronics</td>
<td>8502A</td>
<td>peak power, time, fall time and pulse width plus</td>
<td>70 measurements of a point/second</td>
<td>.03 or 0.75 to 18.5, 26.5 or 40 GHz</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 3: Power-meter specifications comparison
<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Series</th>
<th>Product</th>
<th>Frequency Range</th>
<th>Power Range</th>
<th>Video BW</th>
<th>Price from (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agilent</td>
<td>E9320</td>
<td>E9321A</td>
<td>50 MHz to 6 GHz</td>
<td>-65 dBm (320 pW) to +20 dBm (100 mW)</td>
<td>300 kHz</td>
<td>1,567</td>
</tr>
<tr>
<td>Agilent</td>
<td>E9322A</td>
<td>E9322A</td>
<td>50 MHz to 6 GHz</td>
<td>-60 dBm (1 nW) to +20 dBm (100 mW)</td>
<td>1.5 MHz</td>
<td>2,089</td>
</tr>
<tr>
<td>Agilent</td>
<td>E9323A</td>
<td>E9323A</td>
<td>50 MHz to 6 GHz</td>
<td>-60 dBm (1 nW) to +20 dBm (100 mW)</td>
<td>5 MHz</td>
<td>2,867</td>
</tr>
<tr>
<td>Agilent</td>
<td>E9325A</td>
<td>E9325A</td>
<td>50 MHz to 6 GHz</td>
<td>-60 dBm (320 pW) to +20 dBm (100 mW)</td>
<td>300 kHz</td>
<td>1,880</td>
</tr>
<tr>
<td>Agilent</td>
<td>E9326A</td>
<td>E9326A</td>
<td>50 MHz to 18 GHz</td>
<td>-60 dBm (1 nW) to +20 dBm (100 mW)</td>
<td>1.5 MHz</td>
<td>2,507</td>
</tr>
<tr>
<td>Agilent</td>
<td>E9327A</td>
<td>E9327A</td>
<td>50 MHz to 18 GHz</td>
<td>-60 dBm (1 nW) to +20 dBm (100 mW)</td>
<td>5 MHz</td>
<td>3,336</td>
</tr>
<tr>
<td>Agilent</td>
<td>N192XA</td>
<td>N1921A</td>
<td>50 MHz to 18 GHz</td>
<td>-35 dBm to +20 dBm</td>
<td>N/A</td>
<td>3,763</td>
</tr>
<tr>
<td>Gigatronics</td>
<td>200mW PPS</td>
<td>80350A</td>
<td>45 MHz to 18 GHz</td>
<td>-20 to +20 dBm / +23 dBm</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Gigatronics</td>
<td>5 W PPS</td>
<td>80351A</td>
<td>45 MHz to 18 GHz</td>
<td>0 to +40 dBm / +43 dBm</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Gigatronics</td>
<td>25 W PPS</td>
<td>80352A</td>
<td>45 MHz to 18 GHz</td>
<td>+10 dBm to +50 dBm / +53 dBm</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Gigatronics</td>
<td>50 W PPS</td>
<td>80355A</td>
<td>45 MHz to 18 GHz</td>
<td>+10 to +50 dBm / +53 dBm</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 4: Power-head specifications comparison

For the power-meter, Agilent E4417A was chosen because it met all of our specifications along with our budget. For the power-head, the Agilent E9323A was chosen over other Agilent E-series primarily because if it’s high video bandwidth specification.

Figure 6: Agilent E4417A Power Meter and E9323A Power Heads
6.1.4. Switch Box

The switch box must be a single pole five throw switch cable of handling 50W average input power and 500W peak input power. A VSWR as close to 1 as possible is desirable so that the loads following the switch are properly matched to the amplifier under test. The switch must also be programmable so that the computer can control the load configuration and automate the test plan.

<table>
<thead>
<tr>
<th>Manufacturer and Part Number</th>
<th>Frequency range</th>
<th>Insertion Loss</th>
<th>Isolation</th>
<th>SWR</th>
<th>Connectors</th>
<th>Control Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agilent 87106A</td>
<td>DC to 4GHz</td>
<td>0.3dB + .015 x Freq.(GHz)</td>
<td>100dB Min</td>
<td>1.2 Max</td>
<td>SMA</td>
<td>Internal Control Logic</td>
</tr>
<tr>
<td>Agilent 87106B</td>
<td>DC to 20GHz</td>
<td>0.3dB + .015 x Freq.(GHz)</td>
<td>70dB Min</td>
<td>1.7 Max</td>
<td>SMA</td>
<td>Internal Control Logic</td>
</tr>
<tr>
<td>Agilent 87206B</td>
<td>DC to 4GHz</td>
<td>0.3dB + .015 x Freq.(GHz)</td>
<td>100dB Min</td>
<td>1.2 Max</td>
<td>SMA</td>
<td>Requires External Control Logic</td>
</tr>
</tbody>
</table>

Table 5: Load switch specification comparison

The Agilent 87106A switch was chosen because it meets the necessary requirements. The switch also provides flexibility for future adjustments to the ATS because it is a six-pole switch.

![Figure 7: Agilent 87106A Load Switch](image)
### 6.1.5. Industrial PC

The industrial PC will function as the control unit for our project. It will control all of the instruments and run the test program that we develop. A few companies that offer industrial PCs are shown in Table 6 with their respective systems. We determined that we are looking for an industrial PC with at least 512MB of RAM, an 80GB hard drive, and 6 PCI slots. We determined these specifications to be suitable for our application because our program will not be extremely large and will not require extensive resources to operate. We would prefer the PC to be a rack mountable unit to save shelf space for the other test equipment that is not available in a rack mountable package.

<table>
<thead>
<tr>
<th>Company</th>
<th>Model</th>
<th>PCI express</th>
<th>ISA Slots</th>
<th>Processor</th>
<th>Mount Type</th>
<th>Ram</th>
<th>Price US$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nortech Eng.</td>
<td>IRC Series</td>
<td>14 or 20 Slot Combination Backplane</td>
<td>P4</td>
<td>Standard</td>
<td>512MB</td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td>Allen Bradley</td>
<td>6177RR4SXXP</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>4U rack mount</td>
<td>512MB</td>
<td>TBD</td>
</tr>
<tr>
<td>Allen Bradley</td>
<td>6177RR4PXXP</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>4U rack mount</td>
<td>1GB</td>
<td>TBD</td>
</tr>
<tr>
<td>Industrial Comp.</td>
<td>4UBASICIP</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>4U rack mount</td>
<td>256MB</td>
<td>1,289</td>
</tr>
</tbody>
</table>

*Table 6: Various industrial PC’s and their respective specifications*

After careful consideration, a PC that was already in possession of SRI International was chosen for use in the ATS. The PC is not rack mountable, however the space saved by using a rack mountable PC was not deemed important or necessary. The rack that was chosen contains enough space for all the components used in the ATS including a traditional PC.

**General Purpose Interface Bus Card**

General Purpose Interface Bus (GPIB) cards are specifically designed to connect computers, peripherals and laboratory instruments for data and control transfer between them. Another name for the GPIB is IEEE-488 or HPIB, and is electrically equivalent to
IEC-625 bus. GPIB uses 16 line parallel connections which are divided into eight data lines, three handshake lines for synchronous transfer and five management lines to control the bus. To use the GPIB, we need a GPIB adaptor card in the computer and a GPIB cable. The GPIB cards generally go into the PCI slot of the computer which will be used to control the automated test unit. The GPIB will be used to communicate with the power meter and feed the measurement readings back into our program. There are a variety of GPIB cards with different specifications which are summarized in Table 7.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Part Number</th>
<th>Specifications / Comments</th>
<th>Price US$</th>
</tr>
</thead>
<tbody>
<tr>
<td>National Instruments</td>
<td>778032-01</td>
<td>NI PCI-GPIB NI-488.2 for Windows 2000/XP (Includes Type X2 Cable, 2M)</td>
<td>529.00</td>
</tr>
<tr>
<td>National Instruments</td>
<td>778032-51</td>
<td>PCI-GPIB NI-488.2 for Windows 2000/XP (Includes Type X2 Cable, 2M)</td>
<td>599.00</td>
</tr>
<tr>
<td>National Instruments</td>
<td>778686-01</td>
<td>PCI-GPIB NI-488.2 for LINUX (Includes Type X2 Cable, 2M)</td>
<td>529.00</td>
</tr>
<tr>
<td>National Instruments</td>
<td>778686-51</td>
<td>PCI-GPIB NI-488.2 for LINUX (Includes Type X2 Cable, 2M)</td>
<td>599.00</td>
</tr>
</tbody>
</table>

Table 7: Comparison of different GPIB cards

The GPIB card from National Instrument, NI PCI-GPIB NI-488.2 for Linux w/ 2 meter cable, part number 778686-51, was chosen because it fits into our budget easily and National Instruments provides extensive technical support. The fact that the GPIB card specifically says for Linux systems ensures us that there are some drivers and support for Linux.

Figure 8: National Instruments 488.2 GPIB Card
**Data Acquisition Card (DAQ)**

A data acquisition card is required to connect to the programmable digital potentiometer which will be used to set various currents and voltages for different measurements. They are also required to feed in the data regarding temperature faults and VSWR faults. The market for the DAQ cards like the GPIB cards is owned by National Instruments. However, there are a variety of DAQ cards available with different specifications and for different purposes. Table 8 below summarizes some DAQ cards that were found searching by the low cost requirement.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Part Number</th>
<th>Specifications / Comments</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>National Instruments</td>
<td>779065-01</td>
<td>16-Bit, 250 kS/s 16 Analog Inputs 24 digital I/O 32-bit counters Digital triggering Correlated DIO (8 clocked lines, 1 MHz), Includes NI-DAQmx, VI Logger Lite data-logging software, and other measurement services.</td>
<td>$ 399.00</td>
</tr>
<tr>
<td>National Instruments</td>
<td>777742-01</td>
<td>200 kS/s 12-Bit 16 Analog Input Multifunction DAQ 8 digital I/O lines Two 24-bit counters</td>
<td>$ 499.00</td>
</tr>
<tr>
<td>National Instruments</td>
<td>777743-01</td>
<td>200 kS/s, 12-Bit 16-Analog-Input Two 12-bit analog outputs 8 digital I/O lines Two 24-bit counters</td>
<td>$ 699.00</td>
</tr>
<tr>
<td>National Instruments</td>
<td>778465-01</td>
<td>200 kS/s, 16-Bit 16-Analog-Input Multifunction DAQ Two 16-bit analog outputs Eight digital I/O lines Two 24-bit counters</td>
<td>$ 999.00</td>
</tr>
</tbody>
</table>

*Table 8: Comparison of different Digital I/O cards*
Each NI PCI-6XXX series requires: 1 Cable, 1 Connector Block. The part numbers are different depending on which IO card is chosen, however the prices are the same as shown in Table 9.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Part</th>
<th>Specifications / Comments</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>National Instruments</td>
<td>Cable</td>
<td>SH68-68-EP Cable (2m)</td>
<td>$119</td>
</tr>
<tr>
<td>National Instruments</td>
<td>Connector Block</td>
<td>SCC-68 -Unshielded</td>
<td>$299</td>
</tr>
</tbody>
</table>

Table 9: Additional parts for the Digital IO cards

Digital IO card from National Instruments, NI PCI-6220, part number 779065-01, was chosen because of its low cost, and sampling rate of 250kS/s. The number of analog inputs and digital outputs (16 and 24 respectively) is suitable for our application.

6.1.6. Control Box

The control box was built in an effort to simplify the wiring and installation of the devices in the Automated Test Suite. The ATS requires a large number of connections to be made between devices and the control box was designed to make this easier for both the original designers as well as test and service technicians. The following devices are contained within or interfaced using the control box:
- Signal Generator
- Power Supply
- Power Heads
- Data Acquisition Card
- Load Switch
- Five Loads
- I/Q Detector
- Pulse Modulation Switch
- Three Couplers
- One Attenuator

The control box design allows the SSPA input, output, controller, and power to plug into the front of the control box, while the remaining connections are made in the rear of the box. The couplers and attenuator are incorporated into the control box to ensure they are never accidentally removed because that could permanently damage the power heads. The RF input amplitude is 10dBm and the signal passes through two 20dB couplers that are implemented in series as shown below. The first input coupler attenuates the signal to approximately -10dB and provides the LO signal for the I/Q detector. The second input coupler also attenuates the signal to -10dBm prior to being measured by the power meter using channel A.

![Figure 10: Input Stage Couplers](image)

The output stage of the control box utilizes a 30dB coupler and a 30dB attenuator as shown below. The output of each SSPA is approximately 57dBm and it is attenuated 60dBm using these devices.
The overall control box design is shown in the diagram below.
The front and back panels of the control box were designed for easy implementation in the rack and in an effort to keep the design simple so a technician can easily approach the rack and use it with little instruction.
I/Q Detector

The control box also contains the I/Q detector circuit which uses an AD8347 demodulator chip implemented using its evaluation board provided by analog devices. The AD8347 chip is an I/Q demodulator that directly splits an RF signal to its in-phase and quadrature phase components based on a local oscillator signal (LO) operating at the same frequency as the RF input. Using these two components, the phase of the RF signal coming from the output of the amplifier can be determined. This is important because the phase an SSPA can be adjusted by either incrementing or decrementing a digital potentiometer. Since the phase of all the amplifiers used within the AMISR needs to be within 5 degrees of each other, it is important that we accurately set the phase of each amplifier to a reference value.

The phase splitting of the RF signal is done by the mixer that mixes the LO and the RF signal. As in the case of any modulation involving a mixer, there are two resultant frequencies viz. the sum of two and the difference of two. For a 440 MHz LO and RF signal, the resultant output frequencies are centered around 880 MHz and DC value. The signal near the DC value is what we are concerned with and this requires an RF filter to filter out the higher frequency.
The evaluation board has pads to which inductors and capacitors can be mounted, however, with this restricts the filter design to a pi-network of inductors in series and capacitors in shunt. The filter was designed with the cut-off frequency of 1 MHz with a Bessel design approach to get the maximally flat phase response for the data acquisition card. The schematic of the filter is shown below in Figure 18.

![Figure 18. 7th order RF low pass filter](image)

The filter was designed on paper first and then the performance of the filter was simulated using Agilent's Advanced Design System (ADS). The values chosen for the inductors and the capacitors are standard values readily available in Digi-Key's catalog. The simulation results are show below in Figure 19.
The frequency was swept from 100 KHz to 460 MHz for simulation purpose. The $S_{11}$ parameter (or input reflection coefficient) moves outwards from the center of the Smith Chart as the frequency increases, verifying that more power is reflected than transmitted. The $S_{21}$ parameter (or the forward transmission, also known as gain) is flat at 0dB until the cutoff frequency of 1 MHz. After the cutoff frequency, the 7th order filter comes into action with sharp fall off as the filter attenuates approximately 150 dB per decade.
The evaluation board is connected to our DAQ via the breakout board located within the control box. The DAQ card utilizes differential inputs to evaluate both the in-phase and quadrature phase components of our RF signal. Using these two components and some basic mathematics skills, the resultant vector of these two components, the phase of an SSPA output, is found using the following formula.

\[ \text{phase angle} = \arctan \left( \frac{Q}{I} \right) \]

**Pulse Modulation Switch & Load Switch**

The Automated Test Suite requires that different loads be implemented within the calibration procedure and during testing. Using a load switch provided by Agilent Technologies, the ATS can switch through five different loads depending on what task is being performed. The load switch is located within the control box and is also interfaced using the DAQ card.

The pulse modulation switch was designed at SRI and is used to pulse modulate the RF input signal. The pulse modulation switch will pulse modulate an input signal based on a TTL input. The RF gate line is applied to the TTL input which will cause the pulse modulation switch to output a RF signal with a duty cycle dependent on the RF gate line.

**6.1.7. Rack Design**

After careful consideration, a rack that was already owned by SRI International was chosen for the ATS. The rack manufacturer is HP and the rack already contains casters and is large enough to hold the components used in our application. We designed the layout of the rack as shown below to provide a simple interface for a technician who tests amplifiers with our system, but still allowing for access to devices that an engineer could use to trouble shoot amplifiers that failed our tests. As shown in the rack layout, the control box in located directly above the shelf where each SSPA will be located during testing. The four connections that need to be made to the SSPA are located on the front
panel of the black box as previously explained. This will allow a technician or engineer to easily exchange SSPAs while using our test system.

The layout for the rack, as shown in Figure 20, was also chosen to simplify the installation, wiring, and replacement of components. The system wiring diagram is shown below in Figure 21.
6.2. **Platform**

Wikipedia has the following description in computing world for the word *Platform*—“In computing, a platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer's architecture, operating system, or programming languages and their runtime libraries.”

6.2.1. **Hardware Architecture**

The Dell OptiPlex GX240 PC used for the ATS comprises Intel® 845 Chipset with an Intel® Pentium® 4 2.0 GHz processor and L2 Cache of 512 KB. It belongs to the
80586 family of processors with the chipset bus speed of 400 MHz. The system memory is limited to 512MB SDRAM with the system memory speed of 133 MHz. Since the computer is an Intel® 80586 computer, with x86 (32-bit) bit architecture, it is able to support both Windows and Linux as an operating system.

### 6.2.2. Operating System

The automated test suite is driven by a PC running Fedora Core 6 distribution of Linux as an operating system. “Fedora Core is a free operating system that offers the best combination of stable and cutting-edge software that exists in the free software world.”[5] The Linux kernel version used in the system is 2.6.19-1.2895. The Fedora Project is Red Hat sponsored open-source version of Red Hat Linux, which actually sets the benchmark for Enterprise Linux. Linux operating systems supports multi-user logins as compared to windows (except for Microsoft® Windows Servers) and hence multiple users can work on the same system without having to depend on one another. Linux does not depend upon RPC\(^1\) unlike Windows which uses RPC for almost every application. Linux has been known to be more stable in comparison to Windows and the ever so important fact of Linux being an open source and free by fat beats Windows by a wide margin.

As everything has advantages and disadvantages, there are some disadvantages in using Linux as an operating system. Since Windows has more than 90% of the market, it is very difficult to find support for Linux, especially for driver installations. As expected, some problems were encountered during the driver installation of the GPIB interface card and the NI-DAQ card. Both cards came in with drivers for Linux but not for Fedora Core. The drivers provided by the National Instruments for both the GPIB and the DAQ cards were supported only for Mandrake 10.1, Mandriva 2006, SUSE 10.0 / 10.1 and Red Hat Enterprise WS 3/4 distributions of Linux.

It is possible to load the drivers for Red Hat Enterprise Distribution on Fedora Core distribution because Fedora Core resembles Red Hat Enterprise distribution. There were some modifications that were made to the install script because the Fedora

---

\(^{1}\) Remote Procedure Calls (calls made to other programs using APIs to do some other tasks)
distribution has some files located in different locations when compared to Red Hat. The second step involved patching of the NI-KAL drivers. “NI-KAL is a low-layer driver that is compiled when you install it on your machine. NI-KAL provides "glue" between your Linux kernel and other National Instruments software. A version of NI-KAL is included in every National Instruments Linux driver.” [6] Running the driver updates after the patch resulted in successful installation of both the drivers and a fully operational system.

6.2.3. Programming Language

Out of the various different high level languages available in the commercial market which include Java, C++, Microsoft Visual Studio .NET, Python, and Ruby, Python was chosen for this project solely because the advantages of using Python outweighed the drawbacks. After choosing Linux as the operating system, it was an easy decision to use an open source programming language. Java is the first one that comes to mind when someone talks about powerful open source programming languages, but Python is not far behind either.

Python and Java are very distinct but equally prevailing. Python is essentially a scripting language which can be extended as an object-oriented programming language but the reasons taken into consideration are speed, ease of use, maintenance, and support. Java being a compiler based programming language is faster because the processor only has to load the pre-parsed byte code into the memory to execute the instructions. On the other hand, Python, being an interpreted programming language, is easy to execute and can be done on the fly after changes have been made to the source code; the programmer need not go through the lengthy process of compiling the source code to the object code and then linking the object code with the libraries using a linker. Another big advantage of using Python is that it is dynamically typed (no variable declaration required) whereas Java is statically typed (all variable names along with their types must be explicitly declared). Python also allows the flexibility of assigning a variable with an object of a different type even after it had been assigned to some other type whereas Java’s explicit variable type declaration prohibits it from allowing such feature. The built-in arrays/lists, hashes/dictionaries are significant advantages for Python over Java arrays and its library
based collections. Finally, the person who will be involved in maintenance and support of the ATS is highly proficient in Python.

6.2.4. Database Management System (DBMS)

During the process of calibration and various tests performed on the SSPAs, the ATS fetches sets of data for each of them. Storing that data becomes important for the company to determine production yield, or if the designer wants to know the median for various settings. Although saving data in a plain ASCII text file is convenient, it is not convenient when there are thousands of records to handle. It is much easier to make use of DBMS so that the data can be stored in organized manner and can be easily retrieved in the format that the user wants by sunning simple SQL\(^2\) queries. There are many database management software programs available in the market, e.g. Microsoft Access, Oracle, Filemaker, Microsoft SQL Server, MySQL, and PostgreSQL.

PostgreSQL is a powerful, open source relational database system that can be installed as a package during the installation of Fedora Core (project’s platform) or can be separately installed. “It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows.” [7]

Similar to most of the commercial databases, it is fully ACID\(^3\) compliant, supports foreign keys, table joins, views, triggers, functions, and stored procedures. It also has native programming interfaces for C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, etc.

MySQL is a very popular open source database system in comparison to PostgreSQL. MySQL, however, has many limitations while PostgreSQL provides those

---

\(^2\) Structured Query Language

\(^3\) Atomicity, Consistency, Isolation, and Durability (in terms of database management systems)
functionalities and flexibilities. Vita Voom Software talks about the advantages of PostgreSQL over MySQL in their website\(^4\) which has been summarized below.

PostgreSQL is faster and more efficient than MySQL, supports unlimited row sizes, unlimited database sizes, and tables up to 16TB. It also supports inheritance, foreign keys, Unicode and is more resistant to crashes and power failures by using its logging system. PostgreSQL supports functions (which can be used as stored procedures. It supports outer joins and much more complex multi-joins than MySQL. It also supports “limit” SQL keyword that can be used to limit the number of rows returned making queries more responsive and resource economic. It supports subqueries, indexes on functions and has more flexible BLOB field. Last but not the least, it is a RDBMS\(^5\) that has grown alone, instead of MySQL which is a hack of several tools "glued" together (MSQL, Berkeley DB).

To summarize our discussion over the choice of the database management system, the key points that were taken into consideration for choosing PostgreSQL as the DBMS for the project are listed below:

- Doesn't cost money even for commercial use.
- Works at speed about the same as commercial databases.
- Supports a broader subset of SQL than MySQL like sub-selects
- Extremely responsive in high volume environments
- Supports large tables that exceed Linux' file limit.
- Fully Programmable.
- Known to be legendarily reliable and stable

### 6.3. Software Architecture

Since the test station is automated, it has to be driven by software. Software is the core of this project in terms of control and decision making. The ATS software is capable of controlling all the ATS devices and can perform each and every task the device is capable of programmatically. Software development includes program design, abstraction, device interfacing, implementation and data storage.

---

\(^4\) [http://www.vitavoom.com/postgresql.html](http://www.vitavoom.com/postgresql.html)

\(^5\) Relational Database Management System
6.3.1. Object Oriented Program Design

The purpose behind object oriented program design is simplification of complex software design and goals. Object oriented design enables programmers to think at a very high level about the available resources, their project goals, and how the goals can be achieved with optimal use of their resources. Objects are essentially real world scenarios or concepts tied together as collection of properties and attributes by object classes. The real advantage of object oriented design is easy maintenance as each object is a stand-alone entity and can be executed independent of others [1]. Another advantage of object oriented programming is the ability to reuse object (or concepts) more than once in the same sequence of instructions.

The SSPA Test Station appears straight forward, however, the logic that automates the calibration and specification procedures are complex. Moreover, the interfaces that communicate with the various drivers are even more intricate.

6.3.2. Device level Abstraction

All the devices that are used in the ATS are programmable. With object oriented design concept, it was easy to segregate each individual device as individual object. The specific properties and attributes of each device were constrained within its own object class and these objects communicated with other object solely by message passing via public methods of the object classes. Abstraction at the device level enabled us to take baby steps one at a time and focus on one particular area at one particular instance.

6.3.3. Device Interfacing

Each device used in the ATS has its own device interfacing object class and hence its own file. The protocol for the main program to communicate to the individual devices was by creating object instances of individual devices and calling the public methods to set or get values to and from the devices. Each device had different means of communication with the PC and was implemented in a different way.
With today’s instruments getting more and more complex in terms of development and capabilities, there needs to be a common standard means of communication or interface language between computers and these programmable test instruments. SCPI\(^6\) standard is one typical example of standards that companies are using as interpreter between their hardware and software that controls them. “The SCPI Standard is built on the foundation of IEEE-488.2, Standard Codes and Formats. It requires conformance to IEEE-488.2, but is pure software standard.”\(^2\) “IEEE-488.2 standard defines communication protocols that are necessary to effect application-independent and device-dependent message exchanges, and further defines common commands and characteristics useful in instrument system applications. It is intended to apply to small-scale to medium-scale instrument systems comprised mainly of measurement, stimulus, and interconnect devices outside the scope of the instrument system environment.”\(^3\)

SCPI command set comprises instructions that are simple and common “English-like” syntax which are pure ASCII texts. With these new devices supporting SCPI, the ease of use has definitely increased. The basic instruction set is common among almost all the devices including the low level or register level programming instructions. Each device can have additional high level instruction sets to accomplish some of the specific tasks for which the device is designed.

**GPIB Interface and Ethernet**

The power meter connected to the PC via IEEE 488.2 GPIB interface while the power supply and the signal generator were connected via Ethernet. Interfacing the power supply and the signal generator with the PC was straightforward because of the Ethernet connectivity. Using Berkley socket, simple TCP/IP connection was established with the devices to send SCPI commands and receive readings and data as simple ASCII texts. Controlling the GPIB interface card and establishing connection and communicating with the instrument attached to this card required the knowledge of VISA\(^7\).

---

\(^6\) Standard Commands for Programmable Instrumentation  
\(^7\) Virtual Instrument System Architecture
VISA was originally a specification developed by VXI\textit{plug\&play} Systems Alliance as a step towards industry-wide software compatibility for multi-vendor VXI\textsuperscript{8} systems. However National Instruments defines VISA as “a standard for configuring, programming, and troubleshooting instrumentation systems comprising GPIB, VXI, PXI, Serial, Ethernet, and/or USB interfaces.”[4] National Instruments has also come up with their own implementation of VISA I/O standard—NI-VISA. “NI-VISA includes software libraries, interactive utilities such as NI Spy and the VISA Interactive Control, and configuration programs through Measurement and Automation Explorer for all your development needs.”[4]

VISA libraries are C/C++ library files and cannot be used in Python unless a Python wrapper is built around the library. PyVisa package is one such wrapper built for Python over C/C++ libraries that enables the programmer to control all kinds of measurement equipment through various busses (GPIB, RS232, and USB) with Python programs. PyVISA is tailored to work with arbitrary adapters from National Instruments, Agilent, Tektronix, etc by making calls to the external library file bundled with the hardware and the software of corresponding vendors. The use of PyVISA hides a lot of low level programming required for device communication and enables the programmer to think of what-to-do instead of how-to-do.

**National Instrument’s Data Acquisition Card (DAQ)**

The NI-DAQ 6220 card was used for all the functionalities that it was capable of in addition to pure data acquisition and analog to digital conversion. The card is capable of handling digital inputs and outputs along with generating timing pulse chains or streams using its two counters. Both the counters were implemented to generate the TR gating pulse and the RF gating pulse, with the RF gating pulse being 1μs inside the TR gating pulse. Both pulse chains were of 20ms default period with the RF pulse operating at 10% default duty cycle. The TTL outputs from the counters were used to drive the pulse modulation switch for pulse modulation of the RF signal, the trigger for the power meter as well as the trigger for data acquisition from the I/Q detector, which will be

\textsuperscript{8} VME eXtensions for Instrumentation
discussed later in this section. The digital inputs were used to read in the VSWR fault status and the over temperature status lines from the SSPA. The SP6T\(^9\) load switch uses TTL signals to switch between the 6 output ports. Five of the digital signals required to switch between the five loads used in the ATS were also driven by the DAQ’s digital outputs. The digital outputs were also used to drive the control circuitry of the SSPA. Three digital lines were used to switch between different addresses during calibration, one digital output to control the up/down line and one to control the increment line.

There were basically two ways of implementing the various DAQ functionalities in Python; one was writing low level code for direct register level programming to control the DAQ’s processor and the other was using C/C++ to code the functionalities using the available library from National Instruments and writing a wrapper to export the functions as shared libraries to Python. The latter option was more desirable as Python is built on C construct and use of the library functions is more efficient and it also hides the gory details involved in low level programming of embedded systems.

6.3.4. Class Modules

The software for the ATS was inherits object oriented design concepts and hence the overall software is divided into modules, specifically known as class modules. The modules are characterized as hardware controllers and process controllers. The hardware controllers interface and control the programmable hardware, while the program flow controllers organize the overall flow of the software. The hardware controllers are discussed below. More information regarding the process controllers can be found in Section 6.3.5

(a) Power Supply (PowerSupply.py)

The PowerSupply class module controls the Agilent power supply of the ATS via the ethernet port. The module uses simple Berkley socket to establish connection and SCPI commands to set voltage and current compliances and also to read the voltage and

\(^9\) Single-Pole-6-Throw
the current draw. The module is also capable of regulating the power on and off from the power supply. The full implementation of the methods and their functionalities belonging to this class can be found in Appendix D.

(b) Signal Generator \((\text{SignalGen.py})\)

The SignalGen class module manages the Agilent signal generator using the ethernet port as well. The implementation of this module is similar to the PowerSupply module and uses similar concepts. It also uses Berkley socket to establish connection and SCPI commands to set RF power level, frequency compliance values. It is also capable of switching on and off RF output from the device. The full implementation of the methods and their functionalities belonging to this class can be found in Appendix D.

(c) Power Meter \((\text{PowerMeter.py})\)

The PowerMeter class module interfaces the Agilent power meter using the GPIB card PyVisa python module. The module uses SCPI commands to read and write device settings to and from the power meter. The power meter makes power measurements using SCPI commands as well, but since the power meter is capable of making different types of measurements related to power, specific settings need to be taken care of before making measurements. The module, whenever initialized as an object by any other process module, resets the power meter and sets it to our default settings.

The power meter settings handled by the PowerMeter class are listed below:

- Zero power-heads (both channels)
- Measurement type (peak power, average power)
- Measurement rate (single, double, fast)
- Trigger source (immediate, external)
- Trigger mode (continuous, single, free-run)
- Trigger delay (wait time after trigger before measurement)
- Wait for trigger (wait for external trigger before fetching data)
- Fetch trace data (digitized trace from each channel)
- Gain (correction factor)
The power meter has an advanced feature of getting the power level data trace. The trace however can be fetched from the power meter only if the settings required to enable this feature has been set in a specific order. The power meter has to be in single trigger mode and the trace function has to be enabled. It should also be in wait for trigger mode so that it starts getting the trace as soon as it encounters the external trigger. If these settings are not set in this order, then the power meter throws Settings Conflict error.

This trace was used to perform two of the specification tests on an SSPA, namely over-pulse test and droop test. The data fetched from the power meter was fed into MATLAB to verify the results for these specification tests. The plot shown in Figure 22 was used to verify the output pulse limiting capability of the SSPA (within 2.5ms) for any input pulse greater than the normal pulse width (2ms).

![Output Power Trace from the Power Meter](image)

**Figure 22. Plot of Output Power Trace (dBm) vs. Time (ms)**
The power meter is capable of taking in a value as desired trace length (in seconds) and auto-adjust the resolution of the trace that is returned. Fetching the trace from the power meter with trace length setting of little over 2ms increases the resolution of the output power measurement over a single pulse and hence can be used to calculate the power droop in terms of microseconds inside the pulse as shown in Figure 23.

![Figure 23. Plot of Output Power vs. number of samples over 2ms pulse](image)

(d) **Controller** *(Controller.py)*

The Controller class module is the python module that uses the shared library compiled in C to establish communication between the data acquisition card and various other devices of the ATS. This module interfaces with the control circuitry of the SSPA, the load switch, and the pulse modulator, and acquires data from the I/Q detector circuit for phase measurements.

The data acquisition module written in C *(DAQ.c)* uses the National Instruments’ NIDAQmx C library. The library implements all the digital inputs and outputs as well as the analog inputs in seven distinct phases:
- Create a task and channels and declare them as digital or analog
- Add created channels to the task
- Configure clock to be used for the task
- Define trigger source if needed
- Start the task
- Perform digital read, digital write or analog read
- Stop the task

The concept of tasks is verbose in the National Instruments’ library. Assigning channels to a specific task makes it possible to reserve those particular channels and the resources that have been assigned with the task when the task is started. Each task can be run using different clock signals present internally in the DAQ. The base clock is a 20MHz oscillator and all other clocks are derivatives of this clock. Each task can also be configured to await trigger signal, either analog or digital, on one of the Programmable Function Interface (PFI) ports. The functions that fetch the data from the I/Q detector (triggered by the RF pulse modulation signal) and the function that generate the RF pulse modulation signal (triggered by the TR gating signal) use this feature. After setting the configuration of the task it can be started and digital read/write or analog read can be performed. Finally, it is very necessary to stop the task and release the handle of the resources that were used, or else any subsequent calls would fail as it would not be able to allocate the resource. Although the DAQ retains the output values on the digital output ports, it has a limitation on the timers due to which it cannot retain the gating and the pulse modulation signals. As an exceptional case, the 10% duty-cycle timing functionality is implemented by defining task as static. The task is not stopped and cleared at the end of the function, but at the beginning of the next function call. This way the handle to the resource can be tracked as well as the timers are freely running.

The load switch, which is used to switch between five different loads during different phases of calibration and production test, is controlled digitally using the digital outputs from the data acquisition card. Five digital output lines go into the SSPA’s control circuit to enable switch between different pots as well as to increase and decrease the pot settings. Two digital lines are fed into the digital inputs of the DAQ to sense the VSWR mismatch alarm and the over-temperature alarm given out by the SSPA.
The DAQ is configured to acquire data from the I/Q detector triggered by the RF pulse modulations signal. The I/Q detector uses 1V as its reference voltage and outputs differential voltages for both in-phase component and quadrature-phase components hence the outputs are at an offset of roughly ±1V. The data received from the I/Q detector as read by the data-acquisition card has been presented as a plot below in Figure 24.

As the outputs from the I/Q detector are at an offset, there was the need to acquire data from the RF pulse modulation signal as well. This pulse modulation signal is used as the reference for calculating the bias of individual signals as the pulse modulation signal is just a TTL signal which has two states—low around 0V and high around 5V. The data after filtering out the offset is represented in Figure 25.
Figure 25. Plot of in-phase, quadrature-phase, pulse modulation signal and phase in radians

e) TempSensor (TempSensor.py)

The TempSensor class module rather than controlling the temperature sensor reads the current temperature as outputted by the built in HTTP server within itself. The temperature sensor is capable of reading 4 thermocouples at a time; for the ATS only one is used. The module reads the data as html strings and parses the html string to read the correct data. The temperatures are read in degree Fahrenheit.

6.3.5. Flow Control

The flow control or process control manages the entry point, exit point(s) and sequential flow between processes. A software package typically has a single entry point and effectively should have one exit point with proper implementation. The entry point into the software execution is generally termed as the main routine. The main routine controls the flow of the overall software and makes calls to other supporting procedures by transferring control. While the subordinate procedure does some tasks, the main control waits until it gets the control back. The idea of flow control for any software is self explanatory if presented as a flow diagram or flow chart. The next five flow diagrams
represent the core of ATS’s software architecture—the first one is the main procedure and the next four implement specific tasks that the ATS has been designed to handle.

Figure 26. Main Program Flow
Figure 27: Calibration Process Flow
Figure 28: Pre Burn-in Process Flow
Figure 29: Post Burn-in Process Flow
Device Initialization

Initialization Passed?

User connects 'Cable B' to 'Amp Input'

Did user make connection?

Run Signal Generator at 10dBm

Signal Generator offset is 10 dBm - Ch. B reading

Set Signal Generator to 10 dBm with offset

Ch. A offset is Ch. B reading – Ch. A reading

Set Signal Generator to 17 dBm store Ch. B reading

User connects 'Cable B' to Channel B and 'Amp Input' to 'Amp Output'

Did user make connection?

Set Signal Generator to 17dBm

Ch. B offset is Previous Ch. B reading – current Ch. B reading

Write offsets to offsets.ini

Display Results

Keyboard Interrupt

Catch Exceptions

Back to Main Routine

Figure 30: Power Meter Offset Calibration Flow
6.3.6. User Interface

The ATS software is accessible to the user via a terminal based user interface to perform calibration and run various tests on an SSPA. The user interface provides the user with options as main menu where the user just enters the number corresponding to the process to be executed. A screen shot of the terminal based user interface is shown in Figure 31.

![Terminal based User Interface](image)

**Figure 31: Terminal based User Interface**

The power supply, the signal generator and the power meter all had manual controls on the instruments themselves, however, to carry out manual tests having control over these three devices alone is not sufficient; having the control over the TR gating line, RF gating line and to be able to manually switch between various loads is equally important. The Graphical User Interface allows the user to control the gating lines and load switch manually.
The GUI implemented using Python and Tkinter, which is a thin object-oriented layer on top of Tcl/Tk. Tcl (Tool Command Language), is a powerful dynamic programming language suitable for a very wide range of uses, including desktop applications, networking, testing, etc. It is also an open source language that is cross platform compatible. Tk, on the other hand, is a graphical user interface toolkit that is used for developing desktop applications. Tk is the standard GUI for Tcl which can produce rich, native applications that run unchanged across different platforms like Windows, Mac OS X, Linux and many more.

![Figure 32. Screenshot of the Manual Controls GUI](image)

### 6.3.7. Exceptions and Error Handling

High-quality software ensures that the application does not crash due to the actions of the user. It is also appropriate to display useful messages in case the software is able to trap expected errors and exceptions. A program does not need to terminate in the event of errors or exceptions; instead the system can revert back to its default conditions. There are many places during the calibration and testing phases where exceptions can be expected and therefore need to be handled. Typical examples include during initialization of the devices. If the technician forgets to turn on the power supply or the ethernet connection in the back of the signal generator is disconnected, the program should
provide warnings and possible solutions to fix the error and then be able to continue from
the same spot after the error has been fixed. In order to incorporate such expected cases, a
class that extended Python’s base Exception class was written implementing a set of
custom errors and exceptions. The list of expected and handled exceptions and errors
supported by the ATS are listed below.

- `device_error_PowerSupply`: Cannot establish connection to the Power Supply
- `device_error_PowerMeter`: Cannot establish connection to the Power Meter
- `device_error_DAQCard`: Cannot establish connection to the DAQ
- `device_error_SignalGen`: Cannot establish connection to the Signal Generator
- `device_error_SSPA_Power`: Cannot establish connection to the SSPA
- `calib_initialCurrFailed`: Initial current above 0.5A during Calibration
- `calib_gatePulseFailed`: Current is above 0.5A with only gate pulse
- `calib_normalCurrFailed`: Current is below 3A or above 5A
- `calib_driverCurrFailed`: Current out of bounds while setting Driver Bias
- `calib_outputCurrFailed`: Current out of bounds while setting Output Bias
- `calib_outputPowerFailed`: Power out of bounds while setting Address 4
- `calib_phaseFailed`: Phase out of bounds while setting Address 5
- `calib_vswrFailed`: VSWR out of bounds while setting Address 6
- `calib_offsetCalibfailed`: Power Meter Offset Calibration failed
- `offsets_ini_not_found`: Offsets.init not found
- `process_abort_calib`: Abort Calibration Process
- `process_abort_offsetcalib`: Abort Offset Calibration Process
- `process_failed_offsetcalib`: Offset Calibration

### 6.3.8. Database Programming and Data Storage

The project consists of five distinct test phases; hence the simplest database
schema would be to use a table for each type of test to be performed. This separates the
recorded measurements of one test from another, avoiding complications and the need to
write complex queries. Since the database consists of simple tables, independent of one
another, the concept of foreign key was never implemented because no complex relations
between tables exist. The database consists of eight separate tables that are unique to the
different tests conducted. These tables are listed below. Their respective table names are
shown in parenthesis and the data that is stored in each table is listed below each table name.
• Calibration Data (calibration)
  - Serial Number \((sn)\)
  - Date & Time \((dt)\)
  - Username \((uname)\)
  - Initial Current \((initcurrent)\)
  - Current w/ TR only \((trcurrent)\)
  - Pot 1 Setting \((pot1\_set)\)
  - Driver Bias Current \((curr1)\)
  - Pot 2 Setting \((pot2\_set)\)
  - Output Bias Current \((curr2)\)
  - Current w/ TR & RF \((trrfcurr)\)
  - Initial Power \((initpower)\)
  - Pot 4 Setting \((pot4\_set)\)
  - Final Power \((power4)\)
  - Output Current \((curr4)\)
  - Initial Phase \((initphase)\)
  - Pot 5 Setting \((pot5\_settings)\)
  - Phase \((phase)\)
  - Phase Values \((phase\_values)\)
  - Pot 6 Setting \((pot6\_set)\)
  - Calibration Passed? \((passed)\)
  - Test Failure Code \((test\_fail\_code)\)

• Pre Burn-in \((preburnin)\)
  - Serial Number \((sn)\)
  - Date & Time \((dt)\)
  - Username \((uname)\)
  - Pulse Width \((pulsewidth)\)
  - Overpulse Passed? \((overpulse\_passed)\)
  - Initial Droop Power \((droop\_initpower)\)
  - Final Droop Power \((droop\_finalpower)\)
  - Droop Passed? \((droop\_passed)\)
  - Current at 430MHz \((current\_430)\)
  - Current at 440MHz \((current\_440)\)
  - Current at 450MHz \((current\_450)\)
  - Voltage at 430MHz \((voltage\_430)\)
  - Voltage at 440MHz \((voltage\_440)\)
  - Voltage at 450MHz \((voltage\_450)\)
  - Power at 430MHz \((power\_430)\)
  - Power at 440MHz \((power\_440)\)
  - Power at 450MHz \((power\_450)\)
  - Power w/ 8.5dbm input \((power\_8\_5)\)
  - Power w/ 11.5dbm input \((power\_11\_5)\)
  - Input Variation Passed? \((powercorrect\_passed)\)
  - Open Load Passed? \((open\_passed)\)
• Mismatch Load Passed? (mismatch_passed)
• Test Passed? (Test_passed)
• Test Failure Code (fail_reason)

• DVT: Qualification (*dvt_qualification*)
  • Same data types as Pre Burn-in

• DVT: Pre-test Benchmark (*dvt_pretest_benchmark*)
  • Same data types as Pre Burn-in

• DVT: Post-test Benchmark (*dvt_posttest_benchmark*)
  • Same data types as Pre Burn-in

• DVT: Operational Test (*operational*):
  • Serial Number (sn)
  • Date & Time (dt)
  • Username (uname)
  • User Entered Label (label)
  • Current (current)
  • Input Power (power_in)
  • Output Power (power_out)
  • Minutes Elapsed (mins)
  • Temperature (temp)
  • Temperature Flag (tempflag)
  • VSWR Flag (vswrflag)

• Offset Calibration (*offset_calibration*)
  • Serial Number (sn)
  • Date & Time (dt)
  • Username (uname)
  • Pulse Width (pulsewidth)
  • Overpulse Passed? (overpulse_passed)
  • Initial Droop Power (droop_initpower)
  • Final Droop Power (droop_finalpower)
  • Droop Passed? (droop_passed)
  • Current at 430MHz (current_430)
  • Current at 440MHz (current_440)
  • Current at 450MHz (current_450)
  • Voltage at 430MHz (voltage_430)
  • Voltage at 440MHz (voltage_440)
  • Voltage at 450MHz (voltage_450)
  • Power at 430MHz (power_430)
  • Power at 440MHz (power_440)
  • Power at 450MHz (power_450)
- Power w/ 8.5dbm input (power_8_5)
- Power w/ 11.5dbm input (power_11_5)
- Input Variation Passed? (powercorrect_passed)
- Open Load Passed? (open_passed)
- Mismatch Load Passed? (mismatch_passed)
- Test Passed? (Test_passed)
- Test Failure Code (fail_reason)

**Database Adapter and Data Access Layer**

*Psycopg2* is a PostgreSQL database adapter for the Python programming language. It is the second version of the adapter which is a complete rewrite of the original code to provide new style classes for connection and cursor objects and some other additional features. Similar to the original psycopg, psycopg2 was written with the aim of being small, fast, and stable.

Psycopg is different from other database adapters as it is designed for heavily multi-threaded applications that create and destroy lots of cursors and make a conspicuous number of concurrent INSERTs or UPDATEs. The sole reason behind using psycopg over any other adapters like PygreSQL or PyPgSQL is because the adapter is very intuitive and a lot of support is available online.
7. Twenty-Four Hour Burn-In System

Burn-in is meant to test a production unit’s capability to run under load for an extended amount of time. The concept of burn-in is that you can reduce the probability of a unit being defective in the field by running each production unit while monitoring for malfunctions and errors. An additional benefit to burn-in testing is that it gives the manufacturer more data that can used to judge the quality of production.

7.1. System Requirement

The SSPA’s burn-in system must be capable of running ninety-six amplifiers for twenty-four consecutive hours. The amplifier must run at a frequency between 430MHz and 450MHz at ten percent duty cycle. During the burn-in, the VSWR alarm and over-temperature alarm status must be monitored. If an alarm trips during the burn-in test, then the unit fails the burn-in test and the system should alert a technician.

7.2. System Design

The burn-in system needs to be able to run the amplifier under load and monitor the alarm outputs. Each unit will have its own power supply and load because each amplifier produces more than 500W of peak power and draws almost five amps of current. It was decided that there would be one RF signal source and one transmit/receive signal source for all ninety-six amplifiers. One pulse generator will be used and the signals will be split to all amplifiers to minimize costs. The pulse generator must be capable of producing a synchronized transmit/receive signal and RF modulation signal. Because each of the amplifiers has two alarm signals that need to be monitored and the transmit/receive signal needs to be split to each amplifier, it was decided that a distribution circuit would be designed. The distribution circuit needs to latch to the alarm signals from the SSPAs, alert the operator when a unit has been on for twenty-four hours, and split the transmit/receive signal to all SSPAs. We chose to have only sixteen SSPAs connect to each distribution circuit for simplicity.
7.3. Power Supplies

Each SSPA will have its own power supply. We will be using Mean Well 27V power supplies. The SSPAs require 32V, so the power supplies will be adjusted up to that voltage level.

7.4. Loads

Each SSPA will be producing five-hundred watts peak and needs to have that energy dissipated in a matched load. We chose the MFJ model 264 and ran it at sixty degrees Celsius for twelve hours to make sure that its load characteristics would not change when used in a warm environment for long periods.

7.5. Signal Sources

The two signals that are required to operate an SSPA are the transmit/receive signal and the RF signal. The RF signal is a pulse-modulated 10dBm continuous wave. The pulses of RF must be within the pulses of the transmit/receive signal. This signal enables the amplification stage and must be enabled during RF in order to properly operate the SSPA. The RF signal is going to be generated using a phase-locked oscillator (PLO) at 449 MHz. The PLO requires a 10MHz reference that can be provided by the internal trigger of the Stanford Research Systems DG535 pulse generator chosen to create the transmit/receive signal and the pulse modulation signal.

The DG535 is capable of creating the two pulses needed to properly time the burn-in system. To operate an SSPA properly, the transmit/receive pulse must exceed the RF input by at least one microsecond on both sides. The timing diagram can be found in Appendix C.
7.6. Signal Distribution System

Having only one signal source for ninety-six amplifiers means that there needs to be a distribution system that is capable of supplying the transmit/receive signal and 10dBm of RF to each amplifier. For simplicity, it was decided that the signal splitting would be done in two stages.

The first stage of distribution amplifies the RF signal to compensate for power losses due to splitting. This ensures that 10dBm will be present at the input of each SSPA during the burn-in test. The splitter at this stage is an eight-way power splitter with ten decibels of loss. Also in the first distribution stage is an eight-way buffer that is used to split the transmit/receive signal eight ways.

The second stage of the distribution system splits each of the output signals from stage one sixteen ways. The RF will be split using a two-way splitter going to two eight-way splitters. We’re using two eight-way splitters instead of a sixteen-way splitter because SRI already has six dual eight-way splitters that were designed and manufactured for the AMISR panels available. Altogether, the loss due to stage two is fourteen decibels.

This stage of the distribution system is also to be used for providing information to the technician running the burn-in tests. There are three indicators for each of the sixteen amplifiers. One indicator shows the status of the VSWR alarm, one shows the status of the over-temperature alarm, and the third shows whether the amplifier has been running in the system for twenty-four hours.

The VSWR and over-temp indicators work identically. The alarm signal goes through an inverter and clocks a flip-flop whose input is tied high. The outputs of the flip-flop drive transistors that in turn drive the bi-color LEDs. The twenty-four hour indicator uses the transmit/receive signal as a time-base to increment a counter. When the counter reaches count 72,000, it clocks another flip-flop that is tied high. This flip-flop controls the bi-color LED in the same manner as before, by driving transistors.
The alarm and time indicators need circuitry so that they can be reset when the technician inserts a new SSPA to test. Instead of requiring the technician to press a reset button for each amplifier he or she turns on, the reset circuitry will be activated by the thirty-two volt supply voltage to the amplifier. The reset circuitry uses a comparator to determine when the voltage supply exceeds twenty-eight volts. The comparator’s output is connected to two flip-flops in series that are used with an AND gate to generate a pulse on the rising edge of the comparator’s signal. The output of the AND gate is connected to the reset inputs of all the indicators’ flip-flops. The benefit of the reset circuitry is to avoid the possibility of the technician forgetting to reset the indicators when testing the units.
8. Results

This section provides a sample of the data that the ATS will collect for each amplifier. Table 9 is a sample of the calibration data that is collected for each SSPA that is calibrated using the ATS. The actual database returns more data for each amplifier than shown below. As shown below, in Tables 9 and 10, an engineer can easily analyze this data and monitor manufacturing trends. The data shown below were collected by running query on the custom views that were created within the database. The views enable the user to view the data in a more readable and customized manner.

<table>
<thead>
<tr>
<th>SN</th>
<th>Driver Bias Pot</th>
<th>Driver Current</th>
<th>Output Bias Pot</th>
<th>Output Current</th>
<th>Output Power Pot</th>
<th>Output Power</th>
<th>Phase Pot</th>
<th>Phase (rads)</th>
<th>VSWR Pot</th>
<th>Passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>74</td>
<td>0.303</td>
<td>77</td>
<td>0.505</td>
<td>69</td>
<td>57.216</td>
<td>75</td>
<td>1.670</td>
<td>24</td>
<td>YES</td>
</tr>
<tr>
<td>6</td>
<td>75</td>
<td>0.300</td>
<td>77</td>
<td>0.499</td>
<td>69</td>
<td>57.266</td>
<td>87</td>
<td>1.693</td>
<td>26</td>
<td>YES</td>
</tr>
<tr>
<td>7</td>
<td>73</td>
<td>0.295</td>
<td>68</td>
<td>0.501</td>
<td>54</td>
<td>57.209</td>
<td>57</td>
<td>1.745</td>
<td>64</td>
<td>YES</td>
</tr>
<tr>
<td>9</td>
<td>73</td>
<td>0.297</td>
<td>71</td>
<td>0.495</td>
<td>57</td>
<td>57.260</td>
<td>58</td>
<td>1.685</td>
<td>66</td>
<td>YES</td>
</tr>
<tr>
<td>10</td>
<td>35</td>
<td>0.297</td>
<td>45</td>
<td>0.496</td>
<td>&lt;100</td>
<td>56.964</td>
<td></td>
<td></td>
<td></td>
<td>NO</td>
</tr>
</tbody>
</table>

Table 10: Sample Calibration Data

<table>
<thead>
<tr>
<th>SN</th>
<th>Init Power</th>
<th>Final Power</th>
<th>Droop (%)</th>
<th>PWR @ 430</th>
<th>Eff430 (%)</th>
<th>PWR @ 440</th>
<th>Eff440 (%)</th>
<th>PWR @ 450</th>
<th>Eff450 (%)</th>
<th>PWR 8.5dBm</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>57.135</td>
<td>56.61</td>
<td>11.39</td>
<td>57.72</td>
<td>39.32</td>
<td>57.18</td>
<td>39.32</td>
<td>57.35</td>
<td>39.32</td>
<td>57.13</td>
</tr>
<tr>
<td>6</td>
<td>57.294</td>
<td>56.794</td>
<td>10.87</td>
<td>57.90</td>
<td>40.4</td>
<td>57.38</td>
<td>40.4</td>
<td>57.67</td>
<td>40.4</td>
<td>57.27</td>
</tr>
<tr>
<td>7</td>
<td>57.29</td>
<td>56.777</td>
<td>11.15</td>
<td>57.51</td>
<td>41.89</td>
<td>57.27</td>
<td>41.89</td>
<td>57.30</td>
<td>41.89</td>
<td>57.26</td>
</tr>
<tr>
<td>9</td>
<td>57.269</td>
<td>56.703</td>
<td>12.23</td>
<td>57.38</td>
<td>37.62</td>
<td>57.28</td>
<td>37.62</td>
<td>57.57</td>
<td>37.62</td>
<td>57.25</td>
</tr>
</tbody>
</table>

Table 11: Sample Pre Burn-in Data

The unit that failed calibration during one of the runs (SN 10) as highlighted in Table 9 above, failed due to the power output from the amplifier being less than 57.0dBm after the pot was set to the maximum value of 100. Similarly, the unit that passed calibration during one of the runs, but failed pre burn-in test (SN 9) is highlighted in red in Table 10. The unit failed because the efficiency at all three measurement frequencies—430MHz, 440MHz and 450MHz—was below 38%.
9. Conclusion

Our primary goal was to design a system to be used during production testing of the Solid State Amplifiers (SSPA) used in the Advanced Modular Incoherent Scatter radar (AMISR) being built by SRI International. The system needed to be capable of calibrating each SSPA and conducting multiple tests including a twenty-four hour burn-in test. To accomplish these tasks, we designed two independent systems.

The first system was the Automated Test Suite (ATS) and will be used to calibrate the amplifiers as well as conduct a variety of specification tests. The system is fully operational and is ready to begin testing amplifiers once production begins. The system can also be easily modified to allow engineers to develop new tests in the future. We met all of our goals for this system, including the capability of calibrating and conducting a pre burn-in test in less than 5 minutes. The data collected during calibration and the other test phases is collected and stored in a database that allows it to be easily exported to excel. In the future SRI International will be able to design a custom interface that can be used to monitor the testing of amplifiers from a remote location. There was also a limited supply of amplifiers while we were designing the system. SRI International will need to test more amplifiers to determine the standard for amplifiers being calibrated and tested on the Automated Test Suite.

The second system is know as the Burn-in System. It will be used to conduct a twenty-four hour operational test. The test consists of running ninety-six amplifiers at a 10% duty for a twenty-four hour period. This portion of our project consisted of design a system that could use only one signal generator and a single source for the gate and RF pulse signals to operate ninety-six amplifiers. The system was designed in two stages. Currently the first stage has been completed and tested to verify it functions properly. The second stage has not been finished; the second revision of the distribution board we designed is currently being produced. Once the boards are returned and fully populated, SRI International will need to construct and wire the rack system that will be used to test ninety-six amplifiers.
References

   <http://fmt.isti.cnr.it/~gnesi/matdid/Object_oriented_dsfdesign.pdf>
   <http://www.ivifoundation.org/Combined%20Organizations/SCPI.htm>
3. IEEE. “IEEE Std 488.2-1992 IEEE Standard Codes, Formats, Protocols, and
   <http://standards.ieee.org/reading/ieee/std_public/description/im/488.2-
   1992_desc.html>
   <http://www.ni.com/visa/>
   Questions.” <http://www.ni.com/linux/support.htm#kal>
   <http://www.postgresql.org/about/>
Appendix A
Appendix C
Appendix D
Appendix E
Appendix A

Automated Test Suite
User’s Guide

Authors:
Jeff Pelligrino
John Scimone
Kaushal Shrestha

Date: February 26, 2007
Table of Contents

1. System Startup Instructions: ................................................................. A-3
2. Using the ATS to test SSPA’s. ............................................................... A-6
  2.1. Production Test Package................................................................. A-7
  2.2. Design Verification Test Package.................................................. A-8
  2.3. Manual Controls Package ............................................................. A-9
1. System Startup Instructions:

   Step 1:   Plug grey ATS power cord into electrical outlet.

   Step 2:   Turn computer on.

   Step 3:   Turn computer monitor on.
Step 4: Turn Signal Generator on.

Step 5: Turn Power Supply on.

Step 6: Turn Power Meter on.
Step 7: Turn Control Box on. (Switch located on the rear of box)

Step 8: Verify Power Supply output is OFF.

Step 9: Verify Signal Generator output is OFF.

SYSTEM STARTUP COMPLETE.
2. Using the ATS to test SSPA’s.

There are 3 test packages available on the ATS.

- Production Test Package
- Design Verification Test Package
- Manual Controls Package
2.1. **Production Test Package**

Step 1: Select the desktop icon labeled “Production Test Package.”

Step 2: Enter the username and press the <Enter> key.

Step 3: To select an option for the main menu, press the number on the keyboard corresponding to the option you want to select and press the <Enter> key.
2.2. Design Verification Test Package

Step 1: Select the desktop icon labeled “Design Verification Test Package.”

Step 2: Enter the username and press the <Enter> key.

Step 3: To select an option for the main menu, press the number on the keyboard corresponding to the option you want to select and press the <Enter> key.

* If you select operational test, verify the SSPA is conducted and the environmental chamber is ready before starting the test.
2.3. Manual Controls Package

Step 1: Select the desktop icon labeled “Manual Controls Package.”
Step 2: Use load switch settings and timing card settings to conduct manual tests.
Appendix B

Control Box
Technical Guide

SRI International
Automated Test Suite

Authors:
Jeff Pelligrino
John Scimone
Kaushal Shrestha

Date: February 26, 2007
Table of Contents

1. Control Box Parts List ..................................................................................................... B-3
1.1. I/Q Detector ........................................................................................................ B-4
2. Control Box Connections .......................................................................................... B-7
2.1. Data Acquisition Card Connections ......................................................................... B-8
2.2. Load Switch Connections ...................................................................................... B-9
2.3. Paragrine Switch Connections ............................................................................. B-10
2.4. Control I/O Connections ................................................................................... B-11
2.5. Amp DC Output Connections ............................................................................. B-12
3. Control Box Wiring Diagram ................................................................................... B-13
4. Control Box Layout ................................................................................................... B-14
4.1. Internal Layout .................................................................................................... B-15
4.2. Input Stage Couplers ......................................................................................... B-16
4.3. Output Stage Coupler ........................................................................................ B-16
4.4. Front Panel Layout ............................................................................................ B-17
4.5. Back Panel Layout ............................................................................................. B-17
1. Control Box Parts List

- Front Panel Connectors
  - N-type to N-type Bulkhead Adaptor (Amp Output).
  - SMA to SMA Bulkhead Adaptor (Amp Input).
  - DB9 Panel Mount Receptacle (Control I/O).
  - 4-Pin Panel Mount Receptacle (Amp DC Output).
  - SMA to N-type Bulkhead Adaptor (Power Meter Ch. A).
  - SMA to N-type Bulkhead Adaptor (Power Meter Ch. B).
  - 120V AC Input with built-in switch and fuse.

- Back Panel Connectors
  - 68-pin SCSI Panel Mount Receptacle (DAQ).
  - BNC to BNC Bulkhead Adaptor (Power Meter Ext. Trigger).
  - Agilent SP6T Load Switch.
  - SMA to N-type Bulkhead Adaptor (I/Q Detector Input).
  - N-type to N-type Bulkhead Adaptor (Load Switch Input).
  - N-type to SMA Bulkhead Adaptor (Signal Generator Input).
  - 2-pin Panel Mount Connector (Power Supply Input).

- Internal Components
  - I/Q Detector*
  - Output Coupler: 30dBm
  - Input Coupler 1: 20dBm
  - Input Coupler 2: 20dBm
  - Attenuator: 30dBm
  - DAQ Breakout Box
  - Load Switch TTL Connector
  - Paragrine Switch
  - Power Supply: 5V
  - Power Supply: 24V

* See Section 1.1 of this document for detailed information regarding the I/Q Detector
1.1. **I/Q Detector**

The I/Q detector consists of the AD8347 Direct Conversion Quadrature Demodulator chip implemented using its evaluation board. The functional block diagram for the AD8347 is shown below, followed by the evaluation board schematic.

![Functional Block Diagram](image1)

![Evaluation Board Schematic](image2)
The evaluation board allows a user to implement filters to in-phase and quadrature-phase signals. In this application low pass filters were implemented on the evaluation board. These filters are shown below.

![Diagram of 7th order RF low pass filter](image)

**Figure 1: 7th order RF low pass filter**

The evaluation board has pads that allow inductors and capacitors to be implemented easily, however, pads are placed in a configuration suitable only for a pi-network of inductors in series and capacitors in shunt. The filter was designed with the cut-off frequency of 1 MHz with Bessel approach to get the maximally flat phase response for the data acquisition card. The filter was designed on paper first and then the performance of the filter was simulated using Agilent's Advanced Design System (ADS). The values chosen for the inductors and the capacitors are standard values taken off DigiKey's catalog. The simulation results are show below.
The evaluation board also allows connections to be made to the AD8347 using SMA connectors. The connections made to the I/Q detector evaluation board are shown in the table below.

<table>
<thead>
<tr>
<th>Eval. Board Pin Name</th>
<th>Connection</th>
<th>Signal Desc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>QOPP (J2)</td>
<td>DAQ Pin-33</td>
<td>I/Q Q Pos</td>
</tr>
<tr>
<td>IOPN (J6)</td>
<td>DAQ Pin-34</td>
<td>I/Q I Neg</td>
</tr>
<tr>
<td>IOPP (J5)</td>
<td>DAQ Pin-68</td>
<td>I/Q I Pos</td>
</tr>
<tr>
<td>QOPN (J1)</td>
<td>DAQ Pin-66</td>
<td>I/Q Q Neg</td>
</tr>
<tr>
<td>TP1 (+Vs)</td>
<td>Power Supply: +5V</td>
<td>Positive</td>
</tr>
<tr>
<td>TP4 (GND)</td>
<td>Power Supply: GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>
2. Control Box Connections

This section outlines connections made within the control box including:

- Data Acquisition Card Connections
- Load Switch Connections
- Paragrine Switch Connections
- Amp DC Output Connections
- Control I/O Connections
## 2.1. Data Acquisition Card Connections

<table>
<thead>
<tr>
<th>Pin</th>
<th>Pin Name</th>
<th>Signal Name</th>
<th>Connection Name</th>
<th>Pin</th>
<th>Pin Name</th>
<th>Signal Name</th>
<th>Connection Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PFI 14/P2.6</td>
<td></td>
<td></td>
<td>68</td>
<td>AI 0</td>
<td>I/Q I Pos</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PFI 12/P2.4</td>
<td>RF Signal</td>
<td>Paragrine TTL Signal</td>
<td>67</td>
<td>AI GND</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DAQ Pin-11</td>
<td>66</td>
<td>AI 9</td>
<td>I/Q Q Neg</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Power Meter Ext. Trigger</td>
<td>65</td>
<td>AI 2</td>
<td>Thermo Cpl 2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PFI 9/P2.1</td>
<td>RF Trigger</td>
<td>DAQ Pin 40</td>
<td>64</td>
<td>AI GND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>D GND</td>
<td></td>
<td></td>
<td>63</td>
<td>AI 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PFI 6/P1.6</td>
<td>Increment</td>
<td>Amp DC Output (Grey)</td>
<td>62</td>
<td>AI SENSE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>PFI 5/P1.5</td>
<td></td>
<td></td>
<td>61</td>
<td>AI 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>D GND</td>
<td></td>
<td></td>
<td>60</td>
<td>AI 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>+5 V</td>
<td>(not 5V)</td>
<td></td>
<td>59</td>
<td>AI GND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>D GND</td>
<td></td>
<td></td>
<td>58</td>
<td>AI 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>PFI 1/P1.1</td>
<td>VSWR</td>
<td>Control I/O Pin-3</td>
<td>57</td>
<td>AI 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>PFI 0/P1.0</td>
<td>IQ Trigger</td>
<td></td>
<td>56</td>
<td>AI GND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>D GND</td>
<td></td>
<td></td>
<td>55</td>
<td>NC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>D GND</td>
<td></td>
<td></td>
<td>54</td>
<td>NC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>+5 V</td>
<td>Paragrine +5</td>
<td></td>
<td>53</td>
<td>D GND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>D GND</td>
<td></td>
<td></td>
<td>52</td>
<td>P0.0</td>
<td>Address 0</td>
<td>Control I/O Pin-2</td>
</tr>
<tr>
<td>16</td>
<td>P0.6</td>
<td>Switch Path 4</td>
<td>Load Switch Logic Pin-9</td>
<td>51</td>
<td>P0.5</td>
<td>Switch Path 3</td>
<td>Load Switch Logic Pin-7</td>
</tr>
<tr>
<td>17</td>
<td>P0.1</td>
<td>Address 1</td>
<td>Control I/O Pin-4</td>
<td>50</td>
<td>D GND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>D GND</td>
<td></td>
<td></td>
<td>49</td>
<td>P0.2</td>
<td>Address 2</td>
<td>Control I/O Pin-6</td>
</tr>
<tr>
<td>19</td>
<td>P0.4</td>
<td>Switch Path 2</td>
<td>Load Switch Logic Pin-5</td>
<td>48</td>
<td>P0.7</td>
<td>Switch Path 5</td>
<td>Load Switch Logic Pin-11</td>
</tr>
<tr>
<td>20</td>
<td>NC</td>
<td></td>
<td></td>
<td>47</td>
<td>P0.3</td>
<td>Switch Path 1</td>
<td>Load Switch Logic Pin-3</td>
</tr>
<tr>
<td>21</td>
<td>NC</td>
<td></td>
<td></td>
<td>46</td>
<td>PFI 11/P2.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>NC</td>
<td></td>
<td></td>
<td>45</td>
<td>PFI 10/P2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>AI 15</td>
<td></td>
<td></td>
<td>44</td>
<td>D GND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>AI GND</td>
<td></td>
<td></td>
<td>43</td>
<td>PFI 2/P1.2</td>
<td>Over Temp</td>
<td>Control I/O Pin-1</td>
</tr>
<tr>
<td>25</td>
<td>AI 6</td>
<td></td>
<td></td>
<td>42</td>
<td>PFI 3/P1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>AI 13</td>
<td></td>
<td></td>
<td>41</td>
<td>PFI 4/P1.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>AI GND</td>
<td></td>
<td></td>
<td>40</td>
<td>PFI 13/P2.5</td>
<td>TR Signal</td>
<td>Control I/O Pin-9</td>
</tr>
<tr>
<td>28</td>
<td>AI 4</td>
<td></td>
<td></td>
<td>39</td>
<td>PFI 15/P2.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>AI GND</td>
<td></td>
<td></td>
<td>38</td>
<td>PFI 7/PP1.7</td>
<td>Up/Down</td>
<td>Amp DC Output (Brown)</td>
</tr>
<tr>
<td>30</td>
<td>AI 3</td>
<td>Thermo Cpl 1</td>
<td></td>
<td>37</td>
<td>PFI 8/P2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>AI 10</td>
<td></td>
<td></td>
<td>36</td>
<td>D GND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>AI GND</td>
<td></td>
<td></td>
<td>35</td>
<td>D GND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>AI 1</td>
<td>I/Q Q Pos</td>
<td></td>
<td>34</td>
<td>AI 8</td>
<td>I/Q I Neg.</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>AI 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## 2.2. Load Switch Connections

<table>
<thead>
<tr>
<th>Pin</th>
<th>Func.</th>
<th>Color</th>
<th>Load Switch TTL Logic Connections</th>
<th>Load Switch Path Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>drive common</td>
<td>brown</td>
<td>Load Input</td>
<td>L/S Input from Black Box</td>
</tr>
<tr>
<td>2</td>
<td>indicator common</td>
<td>red</td>
<td>Load 1</td>
<td>50 Ω</td>
</tr>
<tr>
<td>3</td>
<td>drive path 1</td>
<td>orange</td>
<td>Load 2</td>
<td>3:1 Mismatch</td>
</tr>
<tr>
<td>4</td>
<td>indicator path 1</td>
<td>yellow</td>
<td>Load 3</td>
<td>Short</td>
</tr>
<tr>
<td>5</td>
<td>drive path 2</td>
<td>green</td>
<td>Load 4</td>
<td>Open</td>
</tr>
<tr>
<td>6</td>
<td>indicator path 2</td>
<td>blue</td>
<td>Load 5</td>
<td>I/Q Detector</td>
</tr>
<tr>
<td>7</td>
<td>drive path 3</td>
<td>violet</td>
<td>Load 6</td>
<td>Unused</td>
</tr>
<tr>
<td>8</td>
<td>indicator path 3</td>
<td>grey</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>drive path 4</td>
<td>white</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>indicator path 4</td>
<td>black</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>drive path 5</td>
<td>brown</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>indicator path 5</td>
<td>red</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>drive path 6</td>
<td>orange</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>indicator path 6</td>
<td>yellow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Common Ground</td>
<td>green</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>open all paths</td>
<td>blue</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Diagram of Load Switch**

![Load Switch Diagram](image-url)
2.3. Paragrine Switch Connections

<table>
<thead>
<tr>
<th>Pin</th>
<th>Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>Power (5V)</td>
</tr>
<tr>
<td>J2</td>
<td>RF Clock</td>
</tr>
<tr>
<td>RF1</td>
<td>Not Connected</td>
</tr>
<tr>
<td>RF2</td>
<td>RF Input (Signal Generator)</td>
</tr>
<tr>
<td>RFC</td>
<td>Input Coupler 1</td>
</tr>
</tbody>
</table>
### 2.4. Control I/O Connections

<table>
<thead>
<tr>
<th>Pin</th>
<th>Color</th>
<th>Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Black</td>
<td>Temp. Status</td>
</tr>
<tr>
<td>2</td>
<td>Brown</td>
<td>Address 0</td>
</tr>
<tr>
<td>3</td>
<td>Red</td>
<td>VSWR Status</td>
</tr>
<tr>
<td>4</td>
<td>Orange</td>
<td>Address 1</td>
</tr>
<tr>
<td>5</td>
<td>Yellow</td>
<td>Gnd</td>
</tr>
<tr>
<td>6</td>
<td>Green</td>
<td>Address 2</td>
</tr>
<tr>
<td>7</td>
<td>Blue</td>
<td>Not Connected</td>
</tr>
<tr>
<td>8</td>
<td>Purple</td>
<td>Gnd</td>
</tr>
<tr>
<td>9</td>
<td>Grey</td>
<td>T/R</td>
</tr>
</tbody>
</table>
2.5. Amp DC Output Connections

<table>
<thead>
<tr>
<th>Pin</th>
<th>Color</th>
<th>Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Red</td>
<td>32V</td>
</tr>
<tr>
<td>2</td>
<td>Gray</td>
<td>Increment</td>
</tr>
<tr>
<td>3</td>
<td>Black</td>
<td>Gnd</td>
</tr>
<tr>
<td>4</td>
<td>Brown</td>
<td>Up/Down</td>
</tr>
</tbody>
</table>
3. Control Box Wiring Diagram
4. Control Box Layout

This section describes the layout of the control box including:

- Internal Layout
- Input Stage Couplers
- Output Stage Coupler
- Front Panel
- Back Panel
4.1. Internal Layout
4.2. **Input Stage Couplers**

4.3. **Output Stage Coupler**
4.4. Front Panel Layout

4.5. Back Panel Layout
Appendix C

Burn-in
Technical Guide

Authors:
Jeff Pelligrino
John Scimone
Kaushal Shrestha

Date: February 26, 2007
# Table of Contents

1. Signal Distribution: Stage 1 Diagrams ................................................................. C-3
2. Signal Distribution: Stage 2 Diagrams ............................................................... C-4
3. Signal Distribution: Stage 2 Schematics ............................................................ C-6
1. Signal Distribution: Stage 1 Diagrams

Figure 1: Burn-in Stage 1 Flow Diagram

Figure 2: Burn-in Stage 1 Block Diagram
2. Signal Distribution: Stage 2 Diagrams

Figure 3: Burn-in Stage 2 Flow Diagram
Stage 2: TR Splitter

Stage 2: RF Splitter

Stage 2: Channels 1 to 16 (Identical)

Figure 4: Burn-in Stage 2 Block Diagram
3. Signal Distribution: Stage 2 Schematics

Figure 5: TR Splitter and Power Connection Circuitry
Figure 6: Distribution Channel Circuitry
Appendix D

Programming Guide

SRI International
Automated Test Suite

Authors:
Jeff Pelligrino
John Scimone
Kaushal Shrestha

Date: February 26, 2007
# Table of Contents

<table>
<thead>
<tr>
<th>Appendix D</th>
<th>D-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Operating System</td>
<td>D-3</td>
</tr>
<tr>
<td>1.1 Installation</td>
<td>D-3</td>
</tr>
<tr>
<td>1.2 Getting the latest updates</td>
<td>D-3</td>
</tr>
<tr>
<td>1.3 Driver Installation</td>
<td>D-3</td>
</tr>
<tr>
<td>2 Software Packages</td>
<td>D-5</td>
</tr>
<tr>
<td>2.1 PyVISA</td>
<td>D-5</td>
</tr>
<tr>
<td>2.2 NI-VISA</td>
<td>D-5</td>
</tr>
<tr>
<td>2.3 Python Packages</td>
<td>D-6</td>
</tr>
<tr>
<td>2.4 Database Server</td>
<td>D-6</td>
</tr>
<tr>
<td>3 Programming Agilent Power Meter</td>
<td>D-7</td>
</tr>
<tr>
<td>4 Programming Agilent Power Supply</td>
<td>D-9</td>
</tr>
<tr>
<td>5 Programming Agilent Signal Generator</td>
<td>D-10</td>
</tr>
<tr>
<td>6 Programming NI-6220 Data Acquisition Card</td>
<td>D-11</td>
</tr>
<tr>
<td>7 Database Module</td>
<td>D-13</td>
</tr>
<tr>
<td>8 Temperature Sensor Module</td>
<td>D-14</td>
</tr>
<tr>
<td>9 Source Code</td>
<td>D-15</td>
</tr>
</tbody>
</table>

D-2
1 Operating System

Fedora Core 6 is readily available online because it is an open source operating system. The ISO image file of the installation DVD can be downloaded from
1. Fedora’s download site (http://fedora.redhat.com/Download/), or
2. Mirror sites (http://fedora.redhat.com/Download/mirrors.html), or

The complete release notes and the installation guide can be found under Fedora’s documentation section at http://fedora.redhat.com/docs/.

1.1 Installation

The installation process is very easy and intuitive. It is as simple as booting from the DVD and proceeding through the installation with default settings. Do a clean install on the hard drive by selecting to format the partition that you want to install Fedora Core 6 with minimal options.

1.2 Getting the latest updates

Getting the latest update for the working kernel is important along with other updates. Fedora Core has a built in program that searches all the repositories for all available updates that are possible for the current system and configuration. Open up the terminal and as root type the following command to get the updates.

# yum update

1.3 Driver Installation

All the hardware present in the Automated Test Suite (ATS) PC is detected by Fedora’s installer, except for the National Instruments’ GPIB card and the DAQ card.

1. Installing NIDAQ card

Step 1: Insert the CD that came with the ATS
Step 2: Create a symbolic link to asm-offsets.h by typing the following in the terminal window:
# ln -s /lib/modules/$(uname -r)/source/include/asm/asm-offsets.h /lib/modules/$(uname -r)/source/include/asm/asm_offsets.h

Step 3: Open /lib/modules/$(uname -r)/source/include/linux/utsrelease.h and copy the line #define UTS_RELEASE "2.6.xx-1.xxxx.fc6", into /lib/modules/$(uname -r)/source/include/linux/version.h, and save it.

Step 4: As root run the install script from the CD and use the default installation paths for everything.

# <CD-Mount Directory>/NI-DAQ/INSTALL

Step 5: When INSTALL finishes, do not reboot.
Instead patch the nikal using the file found in the Patch directory of the CD using the following command.

# patch -p0 < nikal.patch

Step 6: Copy the modprobe file `modpost` from the Patch directory of the CD to /lib/modules/$(uname -r)/source/scripts/mod/

Step 7: Run updateNIDrivers and reboot.

2. Installing GPIB card

Step 1: Insert the CD that came with the ATS
Step 2: Create a symbolic link to asm-offsets.h:

# ln -s /lib/modules/$(uname -r)/source/include/asm/asm-offsets.h /lib/modules/$(uname -r)/source/include/asm/asm_offsets.h

Step 3: Open /lib/modules/$(uname -r)/source/include/linux/utsrelease.h and copy the line #define UTS_RELEASE "2.6.xx-1.xxxx.fc6", into /lib/modules/$(uname -r)/source/include/linux/version.h, and save it.

Step 4: As root run the install script from the CD and use the default installation paths for everything.

# <CD-Mount Directory>/NI-488225L/INSTALL

Step 5: When INSTALL finishes, do not reboot.
Instead patch the nikal using the file found in the Patch directory of the CD using the following command.

# patch -p0 < nikal.patch
Step 6: Copy the modprobe file ‘modpost’ from the Patch directory of the CD to 
/lib/modules/$(uname -r)/source/scripts/mod/

Step 7: Run `updateNIDrivers` and reboot.

2 Software Packages

Before proceeding with installing software packages that are required to successfully 
run the ATS software, make sure that Python programming environment was installed 
during the native installation of Fedora Core 6. In case you forgot to install Python, 
running yum as root from the shell will install it.

# yum install python

2.1 PyVISA

PyVISA is needed by the ATS software to communicate with the Agilent power 
meter via the National Instrument GPIB interface using Python. PyVISA can be 
downloaded from the internet (http://pyvisa.sourceforge.net/). Alternatively, it can be 
installed by installing the RPM file from PyVISA directory of the Installation CD. The 
RPM can be installed by browsing the directory using the GUI and double-clicking the 
`PyVISA-1.1-1.noarch.rpm` RPM file, or by typing the following command as root.

# rpm -install PyVISA-1.1-1.noarch.rpm

2.2 NI-VISA

PyVISA is a Python wrapper for National Instruments’ VISA Library. The NI-VISA 
Library must be installed before PyVISA can be functional from within Python. To install 
NI-VISA library run the install script as root.

# <CD-Mount Directory>/NI-VISA/INSTALL
2.3 Python Packages

There are some additional Python packages that need to be installed to get the ATS fully operational. The packages namely are python-ctypes, pytz, and python-psyzopg2 which do not come with the native install of Fedora Core 6. They can be installed by running the yum command as root.

    # yum install python-ctypes  pytz  python-psyzopg2

2.4 Database Server

The ATS uses PostgreSQL as the database server. If the PostgreSQL server was not installed during the installation of Fedora Core, it is readily available via yum. Install the database server using ‘yum’ as root.

    # yum install postgresql-server

By default, the Postgres databases can only be accessed by root or postgres users. The user should login as root or the postgres and create a database (preferably ‘SSPA’) using the command `createdb`.

    # su – postgres
    $ createdb sspa

The database can be deleted by using the `dropdb` command.

    $ dropdb sspa

The default database schema for the ATS is provided in the Installation CD and can be imported into the database created by using `psql` (client for PostgreSQL).

    $ psql sspa <  <CD-Mount Directory>/Database/SSPA-schema.sql
3 Programming Agilent Power Meter

The Agilent Power Meter is controlled specifically by the PowerMeter class in

*PowerMeter.py* and the methods within the class. The functions available within the power meter class are listed below with their functionalities.

**Public Functions:**

a. getMeasInput()
   - Gets a peak power measurement from power head channel A.

b. getMeasOutput()
   - Gets a peak power measurement from power head channel B.

c. getTraceData()
   - Gets the trace data from Power Head Channel B.
   - Accepts length of trace (ms).
     - By default, `trace_length = 0.02`.
   - Accepts length of trigger delay (ms).
     - By default, `trigger_delay = 0`.

d. zeroPowerHeads()
   - Zeros power meter channel A and power meter channel B.

e. setFreq()
   - Sets the frequency of power meter channel A and B.
   - Accepts a frequency (MHz).
     - By default, `freq = 440`.

**Private Functions:**

a. __setCaptureRate()
   - Sets the measurement speed on power meter channel A and B
   - Accepts the measurement speed mode: “NORMAL”, “DOUBLE”, or “FAST”.
     - By default, `mode = “DOUBLE”`

b. __setMeasurementSettings()
   - Sets power measurement mode, trigger mode, and trigger source
   - Accepts measurement mode: “PEAK”, or “AVER”
     - By default, `pow = “peak”`
   - Accepts trigger mode: “CONT”, or “IMM”
     - By default, `mode = “CONT”`
   - Accepts trigger source: “BUS”, “EXT”, “HOLD”, “IMM”, or “INT”
     - By default, `trigger = “EXT”`
c. __setTriggerDelay()
   • Sets the trigger delay for power meter channel A and B.
   • Accepts a length of time (s).
     o By default, \( delay = 100e-6 \).
     o

d. __setTraceUnits()
   • Sets the trace function units on power meter channel A and B.

e. __enableTrace()
   • Enables the trace capture for power meter channel B.

f. __setPowMeasurement()
   • Set the power measurement type for power meter channel A and B.
   • Accepts a measurement mode: “PEAK” or “AVER”
   • By default, type = “PEAK”.

g. __setTrigger()
   • Sets the trigger mode for power meter channel A and B.
   • Accepts a trigger mode: “EXT” or “IMM”
   • By default, trigger = “EXT”.

h. __waitTrigger()
   • Sets power meter channel A and B in wait for trigger state.

i. __getOffsetInput()
   • Gets dB loss offset or input stage.

j. __getOffsetOutput()
   • Gets dB loss offset for output stage.
4 Programming Agilent Power Supply

The Agilent Power Supply functions are available in the PowerSupply class in the
PowerMeter.py file and the methods within the class control the functionalities provided
by the power supply. The functions available within the class are listed below with their
functionalities.

Public Functions:

a. setCurr()
   • Sets the current limit.
   • Accepts a current value (A).

b. getVoltageSetpoint()
   • Returns the power supply voltage level setting.

c. setVolt()
   • Sets the voltage limit.
   • Accepts a voltage value (V).

d. getCurr()
   • Returns the power supply current level setting.

e. getVoltageReading()
   • Measures the output voltage.
   • Returns the measured value.

f. setPowerOn()
   • Enables the power supply output and allows the capacitors to charge.

g. setPowerOff()
   • Disables the power supply output.

h. getPowerStatus()
   • Returns the power supply output status (on/off).
5 Programming Agilent Signal Generator

The Agilent Signal Generator functions are available in the SignalGen class in the SignalGen.py file. These functions / methods within the class control various functionalities of the signal generator. The functions available within the class are listed below with their functionalities.

Public Functions:
   a. setRFOff()
      • Disables the RF output of the signal generator.
   b. setRFOn()
      • Enables the RF output of the signal generator.
   c. setAmplitude()
      • Sets the RF output amplitude of the signal generator.
      • Accepts an amplitude (dBm) between -110dBm and 17dBm.
   d. setFreq()
      • Sets the RF output frequency of the signal generator.
      • Accepts a frequency (Hz) between 250kHz and 1GHz.

Private Functions:
   a. __getOffsetOutput()
      • Gets dB loss offset for the output stage.
6 Programming NI-6220 Data Acquisition Card

The controller class is used to control the data acquisition card. The functions available within the controller class are shown below. The functions available in the Controller.py file make calls to the shared object file compiled using GNU C and nidaqmx library provided by National Instruments.

**Public Functions:**

a. `getPhase()`
   - Gets the phase of an amplifier in radians.

b. `getPhaseD()`
   - Gets the phase of an amplifier in degrees.

c. `getPhaseValues()`
   - Gets the in-phase and quadrature phase values in radians.

d. `getPhaseValuesD()`
   - Gets the in-phase and quadrature phase values in degrees.

e. `getVSWR_Flag()`
   - Returns the status of the VSWR flag.

f. `getTemp_Flag()`
   - Returns the status of the temperature flag.

g. `setAddr()`
   - Sets the address.
   - Accepts a number between 0 and 7 inclusive.

h. `setLoad()`
   - Sets the load switch path.
   - Accepts a number between 1 and 5 inclusive.

i. `setIncr()`
   - Sets increment.
   - Accepts either 1 or 0.

j. `setUpDown()`
   - Sets UpDown.
   - Accepts either 1 or 0.
k. Increment()
   • Toggles setIncr() a specified amount while UpDown is high.
   • Accepts the number of times setIncr() will be toggled.
     o By default, number_of_pulses = 1.

l. Decrement()
   • Toggles setIncr() a specified amount while UpDown is low.
   • Accepts the number of times setIncr() will be toggled.
     o By default, number_of_pulses = 1.

m. setTR()
   • Sets the TR gate line pulse.
   • Accepts a value for on/off: 0 or 1.
   • Accepts the period of the pulse (s).
     o By default, period = 20e-3*.
   • Accepts the pulse width (s).
     o By default, pw = 2e-3*.

   *Since the RF pulse needs to be within the TR pulse, the python wrapper for the DAQ will increase the period and pulse width of the TR gate line by 2µs. This will envelope the RF gate line within the TR gate line when they use their default pulse widths and periods.

n. setRF()
   • Sets the RF gate line pulse.
   • Accepts a value for on/off: 0 or 1.
   • Accepts the period of the pulse (s).
     o By default, period = 20e-3.
   • Accepts the pulse width (s).
     o By default, pw = 2e-3.

o. setRFHigh()
   • Sets the RF gate line high.

p. setRFLow()
   • Sets the RF gate line low.
7 Database Module

The database module functions are available in the Database class in the `Database.py` file. These functions / methods within the class are used to communicate with the database. The functions available within the class are listed below with their functionalities.

**Public Functions:**

e. `connect()`  
   - Creates a channel between the code and the database.

e. `storeCalibrationReadings()`  
   - Stores the data collected during calibration.

g. `storeOperationalTestReadings()`  
   - Stores the data collected during operational tests.

h. `storePreBurninReadings()`  
   - Stores the data collected during pre burn-in tests.

i. `storePostBurninReadings()`  
   - Stores the data collected during post burn-in tests.

j. `storeDVTReadings()`  
   - Stores the data collected during non-operational DVT tests.

k. `storeOffsetSettings()`  
   - Stores the data collected during offset calibration.

l. `insertTuples()`  
   - Inserts data into a database.
   - Accepts table name, field names and their corresponding values.
8 Temperature Sensor Module

The Temperature Sensor functions are available in the TempSensor class in the
TempSensor.py file. These functions / methods within the class are used to receive data
from the SensaTronics temperature sensor. The functions available within the class are
listed below with their functionalities.

Public Functions:
  a. getTemperature()
     • Calls a private function to get the temperature data.

Private Functions:
  a. __getSensaTronicTemp(sensor)
     • Gets temperature reading from the designated sensor.
9 Source Code

SRI International
Automated Test Suite
#!/usr/bin/env python

import Controller
import PowerMeter
import PowerSupply
import SignalGen
import Database
import time
import sys

import math
from testexception import *
from utilities import *

import math

print "Please enter the Serial Number of the SSPA :

sn = raw_input(""
serialno = str(sn).strip().replace("","")
values['sn'] = "%s" % serialno
values['uname'] = "%s" % self.user

# First Stage of Calibration
# Startup
# ==========================================

self.controller.setRFOff() # From signal generator
self.controller.setLoad(self.controller.load_50_ohms)
self.powersupply.setPowerOff()
self.powermeter = PowerSupply.PowerSupply()
self.siggen = SignalGen.SignalGen()
self.powermeter = PowerMeter.PowerMeter()
self.controller = Controller.Controller()

self.user = uname
self.siggen.setFreq( calibration_freq ) # Calibration at this freq
self.powermeter.setFreq( calibration_freq )

# Issues with using the default __del__()
try:
    del self.controller
    del self.siggen
    del self.powersupply
except Exception, ex:
    raise TestException(TestException.device_error_SSPA_Power)

# Start of "Public Functions" Block
# ==================================================

def runCalibration (self, serialno = ""):
    values = {   "sn" : "", 
        "uname" : ", 
        "inistcurrent" : ", 
        "trcurr" : ", 
        "pot1_set" : ", 
        "cur1" : ", 
        "pot2_set" : ", 
        "cur2" : ", 
        "trrfcurr" : ", 
        "pot4_set" : ", 
        "initpower" : ", 
        "power4" : ", 
        "cur4" : ", 
        "pot3_set" : ", 
        "phase" : ", 
        "pot6_set" : ", 
        "initphase" : ", 
        "phase_values" : ", 
        } # Values dictionary

    while serialno = ": ";

    # Values dictionary

    print "[ OK ]
    sys.stdout.write ( "

    "[ FAILED ]
    sys.stdout.write ( "

    # End of First Stage
    # *****************************************************

    # Second Stage of Calibration
    # Initialize Digital Potentiometers
    # *****************************************************

    print "Initializing digital potentiometers..."
    countAddr1 = 0
    countAddr2 = 0
    countAddr4 = 0
    countAddr5 = 0
    countAddr6 = 0

    # Set Addr 1 (Driver Stage -- Bias)
    # *****************************************************

    print "Resetting pot 1--driver bias--to %s" % countAddr1
    self.controller.setAddr(1)
    self.controller.setTR(0)
self.controller.Decrement(100)

# Set Addr 2 (Output Stage -- Bias)
self.controller.setAddr(2)
sel controller.setTR(0)
sel controller.Decrement(100)

# Set Addr 4 (Power Level)
print * Resetting pot 4---power level---to %s % countAddr4
self.controller.setAddr(4)
sel controller.setTR(0)
sel controller.Decrement(100)

# Set Addr 5 (Phase)

print * Resetting pot 5---phase---to %s % countAddr5
self.controller.setAddr(5)
sel controller.setTR(0)
sel controller.Decrement(100)

# Set Addr 6 (VSWR)

print * Resetting pot 6---vswr alarm---to %s % countAddr6
self.controller.setAddr(6)
sel controller.setTR(0)
sel controller.Decrement(100)

# Set Addr 0 (no pots selected)

self.controller.setAddr(0)
sel controller.setTR(0)
sel powerSupply.setPowerOff()

# End of Second Stage
# "-------------------------

# Third Stage of Calibration
# Set the potentiometers
# Set Bias

self.powerSupply.setPowerOff()
sel powerSupply.setVolt(32)
sel powerSupply.setCurr(1)
self.powerSupply.setPowerOn()

print *Starting TR Gating line only...*
self.controller.setTR(1) # only TR running at default dutycycle
if (current > initcurrent_max)
    raise TestException(TestException.calib_driverCurrFailed)

self.controller.setAddr(1)
self.controller.setTR(0)
sel controller.setAddr(2)
sel controller.setTR(0)
sel controller.setAddr(0)

self.controller.setTR(0)
sel powerSupply.setPowerOff()

while ((current < driver_bias_min) \ 
      (current > driver_bias_max)):
    raise TestException(TestException.calib_driverCurrFailed)

    self.controller.setAddr(1) # Set to Address 1

    driver_bias = 0.30 # A Mike's new settings
    driver_bias_range = 0.010 # A

    driver_bias_min = driver_bias - driver_bias_range/2
    driver_bias_max = driver_bias + driver_bias_range/2

    print *Setting Addr1 (Driver Bias)to %.0f mA\ % (driver_bias * 1000)

    # Now the first and second stages have passed, set the pot1 to 50
    # so that we can speed up the process of setting pot 1
    countAddr1 += 50
    self.controller.Increment( countAddr1 )

    while ((current < driver_bias_min) \ 
           (countAddr1 < 100)):
        raise TestException(TestException.calib_driverCurrFailed)

        self.controller.setAddr(1)
        self.controller.setTR(0)
        output_bias = 0.500 # A Mike's new settings

        output_bias_range = 0.010 # A
        output_bias_min = output_bias - output_bias_range/2
        output_bias_max = output_bias + output_bias_range/2

        print *Setting Addr1 (Output Bias)to %.0f mA\ % (output_bias * 1000)

        driver_bias = 0.500 # A Mike's new settings
        driver_bias_range = 0.010 # A

        driver_bias_min = driver_bias - driver_bias_range/2
        driver_bias_max = driver_bias + driver_bias_range/2

        print *Setting Addr1 (Output Bias)to %.0f mA\ % (driver_bias * 1000)

        # Now the first and second stages have passed,
# set the pot2 to 50 so that
countAddr2 += 50
self.controller.Increment( countAddr2 )
while ((current < output_bias_min) ∨ (current > output_bias_max)):
    msg = "%s: %sCurrent: %.0f mA % (countAddr2, current*1000)
sys.stdout.write( "%s\n%s\nmsg = "%s Pot2 : %3d	Current : %.0f mA

sys.stdout.write( "%s" % msg )
    current = self.powersupply.getCurr()        if ((current < normcurrent_min)
        ∨ (current > normcurrent_max)):
    # we can speed up the process of setting pot 2
    trrfcurr
values['pot2_set'] = ("%s" % countAddr2)
values['curr2'] = ("%s" % current)
self.__saveCalibrationResults( 
    values, 0, TestException.calib_normalCurrFailed)
    raise TestException('default')
    setStyle('red')
    values['pot2_set'] = ("%s" % countAddr2)
    curr2
values['pot4_set'] = ("%s" % countAddr2)
values['power4'] = ("%s" % power)
    raise TestException('default')
    self.powersupply.setPowerOn()
    self.controller.setAddr(0)
    power = self.powermeter.getMeasOutput()
values['initpower'] = ("%s" % power)
print "Driver and Output Bias Calibration Complete......\n"# End of Third Stage
# =*

# Fourth Stage of Calibration
# Measure and Set Output
# =*

# Set Current to 5A
self.powersupply.setPowerOff()
self.powersupply.setVolt(32)
self.powersupply.setCurr(5)

# Setting address to normal working conditions
self.controller.setAddr(0)
self.controller.setLoad( self.controller.load_50_ohms )
# Turn Power on
self.powersupply.setPowerOn()
# Start RF signal
self.siggen.setRFOn()
print "Turning on RF Gating line..."
self.controller.setRF(1)    # RF and TR running at default values
time.sleep( warmup_period )    # Time delay before current measurement
# =*

# Measure Current and Adjust Output Power
# =*
current = self.powersupply.getCurr()
sys.stdout.write( "Current drawn with RF: %.3s A..." % current )
sys.stdout.flush() ***

We wish to save the initial, pot = 50, current draw and output power and see how this trends over the amplifiers. This should stay in control over the production of the amp.

*** values['trrfcurr'] = ("%s" % current)
if ((current < normcurrent_min) ∨ (current > normcurrent_max)):
    self.__saveCalibrationResults( 
        values, 0, TestException.calib_normalCurrFailed)
    setStyle('bold', 'red')
sys.stdout.write( "[ FAILED ]\n"
    setStyle('default')
    raise TestException('default')
    setStyle('red')
    sys.stdout.write( "[ OK ]\n"
    setStyle('default')
    sys.stdout.write( "Setting Addr 4 (Power Level)...\n"
sys.stdout.write( "Pot4 : %3d	Power : %.5f dBm\n"
    countAddr4 += 1
    self.controller.Increment( step )
    self.controller.Decrement( step )
    countAddr4 -= step

# =*
# Simple Search Algorithm
# Initial bounds for search algorithm
lowerbound = 0
upperbound = 100
while ((power < power_lowend) ∨ (power > power_highend)):
    step = int(math.floor((upperbound − countAddr4)/2))
    if step ≡ 0:
        step = 1
    # Now shift the lowerbound to the value of countAddr4
    lowerbound = countAddr4
    self.controller.setAddr(4)
    countAddr4 += step

    else:
        self.controller.setAddr(4)
        countAddr4 -= step

# Set Current to 5A
self.powersupply.setPowerOff()
self.powersupply.setVolt(32)
self.powersupply.setCurr(5)

# Setting address to normal working conditions
self.controller.setAddr(0)
self.controller.setLoad( self.controller.load_50_ohms )
# Turn Power on
self.powersupply.setPowerOn()
# Start RF signal
self.siggen.setRFOn()
print "Turning on RF Gating line..."
self.controller.setRF(1)    # RF and TR running at default values
time.sleep( warmup_period )    # Time delay before current measurement
# =*

# Measure Current and Adjust Output Power
# =*
current = self.powersupply.getCurr()
sys.stdout.write( "Current drawn with RF: %.3s A..." % current )
sys.stdout.flush() ***

We wish to save the initial, pot = 50, current draw and output power and see how this trends over the amplifiers. This should stay in control over the production of the amp.

*** values['trrfcurr'] = ("%s" % current)
if ((current < normcurrent_min) ∨ (current > normcurrent_max)):
    self.__saveCalibrationResults( 
        values, 0, TestException.calib_normalCurrFailed)
    setStyle('bold', 'red')
sys.stdout.write( "[ FAILED ]\n"
    setStyle('default')
    raise TestException('default')
    setStyle('red')
    sys.stdout.write( "[ OK ]\n"
    setStyle('default')
    sys.stdout.write( "Setting Addr 4 (Power Level)...\n"
sys.stdout.write( "Pot4 : %3d	Power : %.5f dBm\n"
    countAddr4 += 1
    self.controller.Increment( step )
    self.controller.Decrement( step )
    countAddr4 -= step

# =*
# Simple Search Algorithm
# Initial bounds for search algorithm
lowerbound = 0
upperbound = 100
while ((power < power_lowend) ∨ (power > power_highend)):
    step = int(math.floor((upperbound − countAddr4)/2))
    if step ≡ 0:
        step = 1
    # Now shift the lowerbound to the value of countAddr4
    lowerbound = countAddr4
    self.controller.setAddr(4)
    countAddr4 += step

    else:
        self.controller.setAddr(4)
        countAddr4 -= step

# Set Current to 5A
self.powersupply.setPowerOff()
self.powersupply.setVolt(32)
self.powersupply.setCurr(5)

# Setting address to normal working conditions
self.controller.setAddr(0)
self.controller.setLoad( self.controller.load_50_ohms )
# Turn Power on
self.powersupply.setPowerOn()
while ( (phase < phase_low) ∨ (phase > phase_high) ):  
    lowerbound = 0       upperbound = 100  
    sys.stdout.write( "Current after setting power: %.3f A\n" %  
        self.controller.setAddr(0)        current = self.powersupply.getCurr()        sys.stdout.write( "| %s\n" % current)  
        if current < normcurrent_min:  
            self.controller.setAddr(5)        phase = self.controller.getPhase()        sys.stdout.write( "| %s\n" % phase)        self.controller.setTR(0)        self.siggen.setRFOff()        self.powersupply.setPowerOff()  
        else:  
            self.controller.setAddr(6)        self.siggen.setRFOn()        self.powersupply.setPowerOn()  
    sys.stdout.write( "\n" )  
    phase = self.controller.getPhase()        phase_values = self.controller.getPhaseValues()        print "Phase reading at Pot setting of %s is %.5f radians.\n" %  
        (countAddr5, phase)        values['phase'] = "%s" % phase        values['phase'] = "%s" % phase        values['phase'] = "%s" % phase        values['phase'] = "%s" % phase        values['phase'] = "%s" % phase        values['phase'] = "%s" % phase        values['phase'] = "%s" % phase        values['phase'] = "%s" % phase        values['phase'] = "%s" % phase        values['phase'] = "%s" % phase        values['phase'] = "%s" % phase
self.controller.setRF(1)

print "Setting VSWR Mismatch level to 6:1"
# If we start in the middle and initially if it trips then
# it has been set to a lower tolerance and hence we need to
# increase the tolerance
# Simple Search Algorithm
# Initial bounds for search algorithm
lowerbound = 0
upperbound = 100
step = 100 # Just make sure it is not 0 or 1

while True:
    vswr_fault = self.controller.getVSWR_Flag()
    sys.stdout.write("Pot6 : %3s	6:1 mismatch reached: %s
"
    )
    sys.stdout.flush()

    if ( vswr_fault == 1):
        step = int(math.floor((upperbound - countAddr6)/2))
        countAddr6 += step
        self.controller.Increment(step)

    elif ( vswr_fault == 0):
        step = math.floor((upperbound - countAddr6)/2)
        countAddr6 -= step
        self.controller.Decrement(step)

    else:
        break

    if ( countAddr6 == 1):
        sys.stdout.write("VSWR POT
"
        )
        sys.stdout.flush()
        self.__saveCalibrationResults(values)

        print "VSWR Mismatch Calibration Complete......"

        break

# SAVE DATA TO THE DATABASE
self.__saveCalibrationResults(values)

# Seventh Stage of Calibration
self.controller.setAddr(7)
self.controller.setIncr(1)
# Shutdown
self.powersupply.setPowerOff()
self.powersupply.setVolt(32)
self.powersupply.setCurr(1)
self.controller.setTR(0)

# Cold switching
self.siggen.setRFOff()

# From signal generator
self.controller.setLoad(self.controller.load_50_ohms)
# End of Seventh Stage
import DAQ
import time
import math
from Utilities import *
def int2bin(integer, bits = 5):
    # Convert integer value to binary value
    bin = ''
    while (True):
        bin = str(integer % 2) + bin
        integer = int(integer / 2)
        if (integer == 0):
            break
    return str(('0'*bits) + bin)
class Controller:
    # Load enumerations
    load_50_ohms = 1
    load_3_1_mismatch = 2
    load_6_1_mismatch = 3
    load_open = 4
    load_IQ = 5
    def __init__(self):
        self.setTR(0)
        self.setAddr(0)
        self.setLoad(1)
        self.setIncr(1)
        self.setUpDown(0)
def __del__(self):
    self.setTR(0)
# Analog Input
# ==============================================================
def getPhase(self):
    phase = self.getPhaseValues()
    phase_avg = sum(phase) / len(phase)
    return phase_avg

def getPhaseValues(self):
    # Not sure if this is needed
    time.sleep(0.25)
    # Call the DAQ to get the I, Q, and pulse mod values
data[0] = list of I values
data[1] = list of Q values
data[2] = list of rf pulse modulation values
    # data = DAQ.acquireData(1000, 0.06)
    q_offsets = []
    i_offsets = []
    i_values = []
    q_values = []
    phase_values = []
    for i,q,rf_pulse in zip(data[0], data[1], data[2]):
        if rf_pulse < 2.5:
            # Low on the RF Gate
            # RF pulse low, 0 W, I and Q at their bias levels
            i_offsets.append(i)
            q_offsets.append(q)
        else:
            # RF pulse high, 57dBm, I and Q tell us phase
            i_values.append(i)
            q_values.append(q)
            i_offset = sum(i_offsets) / len(i_offsets)
            q_offset = sum(q_offsets) / len(q_offsets)
            for i, q in zip(i_values, q_values):
                i -= i_offset
                q -= q_offset
                temp = math.atan2(q, i)
                if temp < 0:
                    temp += 2 * math.pi
                phase_values.append(temp)
        return phase_values

def getPhaseValuesD(self):
    phase = self.getPhaseValues()
    phaseDeg = []
    for p in phase:
        phaseDeg.append(p * 180 / math.pi)
    return phaseDeg
# Digital Inputs
# ==============================================================
def getVSWR_Flag(self):
    time.sleep(0.5) # Wait before taking the sample
data = DAQ.readDigital()
    vals = []
    for i in range(0, 200):
        vals.append(val)
    return 1
else:
    return 0
def getTemp_Flag(self):
    time.sleep(0.5) # Wait before taking the sample
data = DAQ.readDigital()
def setAddr(self, addr):
    # Mask with 0000 0111 to ensure addr is from 000 − 111
    addr = 0x07 & addr
    DAQ.writeDigital(0, "2:0", int2bin(addr, 3))

def setLoad(self, switchno = 1):
    # Switch can be from 1 to 5
    self.setTR(0)  # Makes sure of cold switching
time.sleep(0.25)
    switchno = 1 << (switchno - 1)
    DAQ.writeDigital(0, "7:3", int2bin(switchno, 5))
time.sleep(0.25)

def setIncr(self, value):
    # Value takes in a 1 or a 0
    # Incrementf.setIncr(1) is Bit number 3 in the 8 bit DIO
    if (bool(value) ≡ True):
        DAQ.writeDigital(1, "6", "1")
    else:
        DAQ.writeDigital(1, "6", "0")

def setUpDown(self, value):
    # Value takes in a 1 or a 0
    # UpDown is Bit number 4 in the 8 bit DOUT
    if (bool(value) ≡ True):
        DAQ.writeDigital(1, "7", "1")
    else:
        DAQ.writeDigital(1, "7", "0")

def Increment(self, number_of_pulses=1):
    ***
    This function toggles setIncr a specified amount
    while self.setUpDown is set to one.

    number_of_pulses is the number of times you want the Incr Line to
toggle from 1 to 0
    ***
    self.setUpDown(1)
timedelay = 0.0625
time.sleep(timedelay)
    self.setIncr(1)
    for i in range(0,2*number_of_pulses):
        self.setIncr(i % 2)
time.sleep(2e-6)
    self.setIncr(1)
time.sleep(timedelay)

def Decrement(self, number_of_pulses=1):
    ***
    This function toggles setIncr a specified amount
    while self.setUpDown is set to zero.

    number_of_pulses is the number of times you want the Incr Line to
toggle from 1 to 0
    ***
#!/usr/bin/env python

import sys
import string
import datetime
import pytz
import psycopg2

# load the psycopg extras module
import psycopg2.extras

UTC = pytz.timezone( 'UTC' )

class Database:
    def __init__(self):
        self.dbcon = self.connect()
    
    def connect(self):
        dbcon = None
        try:
            dbhost = None
            # None or hostname
            dbname = "sspa"  # Database name
            dbuser = None
            # "postgres"
            if dbhost:
                # if host is given then auth mechanism is different
                connect_string = 'host=%s dbname=%s user=%s' % ( dbhost, dbname, dbuser )
            else:
                if dbuser:
                    connect_string = 'dbname=%s user=%s' % ( dbname, dbuser )
                else:
                    connect_string = 'dbname=%s' % ( dbname )
                
            dbcon = psycopg2.connect( connect_string )
        except psycopg2.OperationalError, inst:
            msg = "%s db dbcon failed: %s"% (__name__,inst)
            print msg
            raise
        return dbcon

    def storeCalibrationReadings(self, names_str, values_str):
        names_str  = names_str  + ', dt
        values_str = values_str + ',%s' % datetime.datetime.utcnow()
        self.insertTuples( "calibration", names_str, values_str )

    def storeOvenTestReadings(self, names_str, values_str):
        names_str  = names_str  + ', dt
        values_str = values_str + ',%s' % datetime.datetime.utcnow()
        self.insertTuples( "oventest", names_str, values_str )

    def storePreBurninReadings(self, names_str, values_str):
        names_str  = names_str  + ', dt
        values_str = values_str + ',%s' % datetime.datetime.utcnow()
        self.insertTuples( "preburnin", names_str, values_str )

    def storePostBurninReadings(self, names_str, values_str):
        names_str  = names_str  + ', dt
        values_str = values_str + ',%s' % datetime.datetime.utcnow()
        self.insertTuples( "postburnin", names_str, values_str )

    def storeDVTReadings(self, names_str, values_str, tablename):
        names_str  = names_str  + ', dt
        values_str = values_str + ',%s' % datetime.datetime.utcnow()
        self.insertTuples( tablename, names_str, values_str )

    def storeOffsetSettings(self, names_str, values_str):
        names_str  = names_str  + ', dt
        values_str = values_str + ',%s' % datetime.datetime.utcnow()
        self.insertTuples( "offset_calibration", names_str, values_str )

    def insertTuples( self, tablename, names_str, values_str ):
        dbcur = self.dbcon.cursor()
        strSQL = "INSERT INTO %s (%s) VALUES (%s)" % ( tablename, names_str, values_str )
        try:
            dbcur.execute(strSQL)
        except psycopg2.ProgrammingError, e:
            msg = "ProgrammingError: %s" % e
            print msg
            self.dbcon.rollback()
            raise
        try:
            self.dbcon.commit()
        except psycopg2.ProgrammingError, e:
            msg = "ProgrammingError: %s, not committing" % e
            print msg
            self.dbcon.rollback()
            raise
        dbcur.close()

if __name__ == "__main__":
    db=Database()
    for i in db.fetchData():
        print i
#!/usr/bin/env python
import DVTTests
import OperationalTest
import string
import os
import sys
import time
from TestException import *
from Utilities import *
class DVT:
    def __init__(self):
        os.system('reset')
        self.WelcomeScreen()
    def WelcomeScreen(self):
        os.system('clear')
        setStyle('bold')
        print 60 * "="
        print "Welcome to the AMISR SSPA Design Verification Test Suite"
        print 60 * "="
        setStyle('default')
        self.UserName = LoginName
    def Login(self):
        print """Please login below (avoid spaces) ... ""
        LoginName = ""
        while LoginName == "":
            try:
                LoginName = str(raw_input("Username :"))
                LoginName = LoginName.strip().replace(" ","_")
            except KeyboardInterrupt:
                print "Process Terminated By User ... \n"
                setStyle( 'default' )
                sys.exit(1)
        self.UserName = LoginName
    def run (self):
        while True:
            self.Login()
            self.WelcomeScreen()
            while (self.UserName == ""):
                print """0. Log Off"", self.UserName
                print "1. Qualification Test" # Pre-burnin
                print "2. Operational Test" # Oventest
                print "3. Pre-test Benchmark" # Pre-burnin
                print "4. Post-test Benchmark" # Post-burnin
                print "Q. Quit"\n"
                inp = ""
                while inp == "":
                    try:
                        inp = raw_input("Enter our choice :")
                        inp = string.lower( str(inp) )
                        inp = inp.strip()
                    except KeyboardInterrupt:
                        setStyle( 'bold', 'red' )
                        print "\n\nProcess Terminated By User ... \n"
                        setStyle( 'default' )
                        sys.exit(1)
                if inp == "q" :
                    return
                try:
                    try: # Try , finally block
                        if ( inp == "1" ) :
                            proc = DVTTests.DVTTests(self.UserName)
                            proc.runDVT( proc.qualification_test )
                        elif ( inp == "2" ) :
                            proc = OperationalTest.OperationalTest (self.UserName)
                            proc.runOperationalTest()
                        elif ( inp == "3" ) :
                            proc = DVTTests.DVTTests(self.UserName)
                            proc.runDVT( proc.pretest_benchmark )
                        elif ( inp == "4" ) :
                            proc = DVTTests.DVTTests(self.UserName)
                            proc.runDVT( proc.posttest_benchmark )
                    except KeyboardInterrupt:
                        setStyle( 'bold', 'red' )
                        print "Process Terminated By User ... \n"
                        setStyle( 'default' )
                        sys.exit(1)
                    except Exception, err:
                        setStyle('bold', 'red')
                        print "\n\nProcess Terminated By User ... \n"
                        setStyle( 'default' )
                        for _msg in msg:
                            print _msg + _msg
                        setStyle( 'default' )
                finally:
                    # Delete the object so that it calls its destructor
                    # to restore settings to default
                    try:
                        proc.destructor()
                        del proc
                    except:
                        pass
if __name__ == "__main__":
    DVT().run()
    print "Please wait while the system shuts down ... \n\n"
#!/usr/bin/env python

import SpecTests
import Database
import time
import string
import sys
import math

from TestException import *
from Utilities import *

class DVTTests:
    qualification_test = 1
    pretest_benchmark = 2
    posttest_benchmark = 3
    dict_tests = {
        qualification_test: "Qualification Test",
        pretest_benchmark: "Pre−test Benchmark",
        posttest_benchmark: "Post−test Benchmark"
    }

    def __init__(self, username):
        self.spectest = SpecTests.SpecTests(username)
        self.user = username
        raise

    def __del__(self):
        try:
            del self.spectest
        except:
            del self.spectest
        pass

    # Values dictionary
    sn = raw_input("Please enter the Serial Number of the SSPA : ")
    serialno = str(sn).strip().replace(" ", ")

    # Initialize devices
    self.spectest.initDevices()

    # Overpulse with 3ms pulse
    overpulse = self.conductOverPulseTest()
    values = {'pulsewidth': str(string.join(overpulse[0], ','))}
    overpulse_passed = True
    if bool(overpulse[1]):
        self.spectest.initDevices()
        return False

    # Droop
    droop = self.conductDroopTest()
    values = {'droop_initpower': str(droop[0]), 'droop_finalpower': str(droop[1])}
    droop_passed = True
    if droop_passed:
        self.spectest.initDevices()
        return True

    # Efficiency at 430, 440, 450
    for i in [430, 440, 450]:
        values = {eval('current_%s' % i): str(values[i])}
        for j in [430, 440, 450]:
            values = {eval('voltage_%s' % j): str(values[j])}
            for k in [430, 440, 450]:
                values = {eval('power_%s' % k): str(values[k])}
                if i == 440:
                    values['droop_imppower'] = "%s" % str(droop[0])
                    values['droop_finpower'] = "%s" % str(droop[1])
                    values['droop_passed'] = True
                if droop_passed:
                    self.spectest.initDevices()
                    return False

    # Efficiency Test after burnin ...	
testFailFlag = False

    # Overpulse with 3ms pulse
    overpulse = self.conductOverPulseTest()
    values = {'pulsewidth': str(string.join(overpulse[0], ','))}
    overpulse_passed = True
    if bool(overpulse[1]):
        self.spectest.initDevices()
        return False

    # Droop
    droop = self.conductDroopTest()
    values = {'droop_initpower': str(droop[0]), 'droop_finalpower': str(droop[1])}
    droop_passed = True
    if droop_passed:
        self.spectest.initDevices()
        return True

    # Efficiency at 430, 440, 450
    for i in [430, 440, 450]:
        values = {eval('current_%s' % i): str(values[i])}
        for j in [430, 440, 450]:
            values = {eval('voltage_%s' % j): str(values[j])}
            for k in [430, 440, 450]:
                values = {eval('power_%s' % k): str(values[k])}
                if i == 440:
                    values['droop_imppower'] = "%s" % str(droop[0])
                    values['droop_finpower'] = "%s" % str(droop[1])
                    values['droop_passed'] = True
                if droop_passed:
                    self.spectest.initDevices()
                    return False

    # Efficiency Test after burnin ...	
testFailFlag = False

    # Overpulse with 3ms pulse
    overpulse = self.conductOverPulseTest()
    values = {'pulsewidth': str(string.join(overpulse[0], ','))}
    overpulse_passed = True
    if bool(overpulse[1]):
        self.spectest.initDevices()
        return False

    # Droop
    droop = self.conductDroopTest()
    values = {'droop_initpower': str(droop[0]), 'droop_finalpower': str(droop[1])}
    droop_passed = True
    if droop_passed:
        self.spectest.initDevices()
        return True

    # Efficiency at 430, 440, 450
    for i in [430, 440, 450]:
        values = {eval('current_%s' % i): str(values[i])}
        for j in [430, 440, 450]:
            values = {eval('voltage_%s' % j): str(values[j])}
            for k in [430, 440, 450]:
                values = {eval('power_%s' % k): str(values[k])}
                if i == 440:
                    values['droop_imppower'] = "%s" % str(droop[0])
                    values['droop_finpower'] = "%s" % str(droop[1])
                    values['droop_passed'] = True
                if droop_passed:
                    self.spectest.initDevices()
                    return False

    # Efficiency Test after burnin ...	
testFailFlag = False

    # Overpulse with 3ms pulse
    overpulse = self.conductOverPulseTest()
    values = {'pulsewidth': str(string.join(overpulse[0], ','))}
    overpulse_passed = True
    if bool(overpulse[1]):
        self.spectest.initDevices()
        return False

    # Droop
    droop = self.conductDroopTest()
    values = {'droop_initpower': str(droop[0]), 'droop_finalpower': str(droop[1])}
    droop_passed = True
    if droop_passed:
        self.spectest.initDevices()
        return True

    # Efficiency at 430, 440, 450
    for i in [430, 440, 450]:
        values = {eval('current_%s' % i): str(values[i])}
        for j in [430, 440, 450]:
            values = {eval('voltage_%s' % j): str(values[j])}
            for k in [430, 440, 450]:
                values = {eval('power_%s' % k): str(values[k])}
                if i == 440:
                    values['droop_imppower'] = "%s" % str(droop[0])
                    values['droop_finpower'] = "%s" % str(droop[1])
                    values['droop_passed'] = True
                if droop_passed:
                    self.spectest.initDevices()
# Power Correct

powercorrect = self.conductPowerCorrectTest()
values [ 'power_8_5' ] = powercorrect[0][0]
values [ 'power_11_5' ] = powercorrect[0][1]
values [ 'power_correct_passed' ] = "%%s" % 
if bool(powercorrect[1]) == True:
    testFailFlag = True

# Open and Mismatch

variousloads_passed = self.conductVariousLoadsTest()
values [ 'open_passed' ] = "%%s" % 
values [ 'mismatch_passed' ] = "%%s" % 
if bool(variousloads_passed[0]) == False or 
    bool(variousloads_passed[1]) == False:
    testFailFlag = True

# Save Data

# -------------------------------
saveResults(values, testType)
# -------------------------------

# End of Procedure

print "%a"
if testFailFlag == True:
    setStyle('bold', 'red')
    print 35 * '-'
    print "%s Failed !!!" % self.dict_tests[ testType ]
    print 35 * '-'
else:
    setStyle('bold', 'green')
    print 35 * '-'
    print "%s Completed !!!" % self.dict_tests[ testType ]
    print 35 * '-'
setStyle('default')

# Data Saving Routines

# -------------------------------

saveResults(self, results, testType):

tablename = {self.qualification_test  : "dvt_qualification",
             self.pretest_benchmark   : "dvt_pretest_benchmark",
             self.posttest_benchmark  : "dvt_posttest_benchmark"}

names = []
values = []
for fieldname, value in results.items():
    names.append( "%%s" % fieldname )
    values.append( "%%s" % value )

names_str = string.join(names, ',
values_str = string.join(values, ',

db = Database.Database()
db.storeDVTReadings(names_str, values_str, tablename[ testType ])
from Tkinter import 
import Tkinter as Tk
import Controller

def setLoadOne(event):
    selectRFandGateOff()
    controller.setLoad(controller.load_50_ohms)
    print "load switched to one"

def setLoadTwo(event):
    selectRFandGateOff()
    controller.setLoad(controller.load_6_1_mismatch)
    print "load switched to two"

def setLoadFive(event):
    selectRFandGateOff()
    controller.setLoad(controller.load_IQ)
    print "load switched to five"

def setLoadFour(event):
    selectRFandGateOff()
    controller.setLoad(controller.load_open)
    print "load switched to four"

def setLoadThree(event):
    selectRFandGateOff()
    controller.setLoad(controller.load_3_1_mismatch)
    print "load switched to three"

def setRFandGateOff(event):
    controller.setTR(0)
    print "Both Off"

def setGateOn(event):
    controller.setTR(1)
    print "Gate Only"

def selectRFandGateOff():
    SELF.frame2_RFandGateOff.select()    SELF.frame2_RFandGateOn = Radiobutton (frame2, text="RF and Gate ON", variable=timevar, value=3, cursor="hand2")
    self.frame2_RFandGateOn.place(in_=frame2,x=140,y=10)
    self.frame2_RFandGateOn.bind("<Button−1>", func=setRFandGateOn)

    self.frame1_loadOne = Radiobutton(frame1, text="50 Ohm", variable=load, value=1, cursor="hand2")
    self.frame1_loadOne.place(in_=frame1,x=50,y=60)
    self.frame1_loadOne.bind("<Button−1>", func=setLoadOne)

    self.frame1_loadTwo = Radiobutton(frame1, text="6:1 Mismatch", variable=load, value=2, cursor="hand2")
    self.frame1_loadTwo.place(in_=frame1,x=50,y=90)
    self.frame1_loadTwo.bind("<Button−1>", func=setLoadTwo)

    self.frame1_loadThree = Radiobutton(frame1, text="3:1 Mismatch", variable=load, value=3, cursor="hand2")
    self.frame1_loadThree.place(in_=frame1,x=50,y=120)
    self.frame1_loadThree.bind("<Button−1>", func=setLoadThree)

    self.frame1_RFandGateOn = Radiobutton (frame1, text="RF and Gate ON", variable=timeVar, value=1, cursor="hand2")
    self.frame1_RFandGateOn.place(in_=frame1,x=70,y=10)
    self.frame1_RFandGateOn.bind("<Button−1>", func=setRFandGateOn)

    self.frame1_loadOne = Radiobutton(frame1, text="50 Ohm", variable=load, value=1, cursor="hand2")
    self.frame1_loadOne.place(in_=frame1,x=50,y=60)
    self.frame1_loadOne.bind("<Button−1>", func=setLoadOne)

    self.frame1_loadTwo = Radiobutton(frame1, text="6:1 Mismatch", variable=load, value=2, cursor="hand2")
    self.frame1_loadTwo.place(in_=frame1,x=50,y=90)
    self.frame1_loadTwo.bind("<Button−1>", func=setLoadTwo)

    self.frame1_loadThree = Radiobutton(frame1, text="3:1 Mismatch", variable=load, value=3, cursor="hand2")
    self.frame1_loadThree.place(in_=frame1,x=50,y=120)
    self.frame1_loadThree.bind("<Button−1>", func=setLoadThree)

    self.frame2_RFandGateOn = Radiobutton (frame2, text="RF and Gate OFF", variable=timevar, value=1, cursor="hand2")
    self.frame2_RFandGateOn.place(in_=frame2,x=70,y=10)
    self.frame2_RFandGateOn.bind("<Button−1>", func=setRFandGateOff)

    self.frame2_RFandGateOff = Radiobutton (frame2, text="RF and Gate OFF", variable=timevar, value=2, cursor="hand2")
    self.frame2_RFandGateOff.place(in_=frame2,x=110,y=10)
    self.frame2_RFandGateOff.bind("<Button−1>", func=setRFandGateOff)

    self.frame2_RGateOnly = Radiobutton (frame2, text="Gate Only", variable=timevar, value=3, cursor="hand2")
    self.frame2_RGateOnly.place(in_=frame2,x=70,y=140)
    self.frame2_RGateOnly.bind("<Button−1>", func=setGateOn)

    self.frame1_RFandGateOn = Radiobutton (frame1, text="RF and Gate ON", variable=timeVar, value=1, cursor="hand2")
    self.frame1_RFandGateOn.place(in_=frame1,x=70,y=10)
    self.frame1_RFandGateOn.bind("<Button−1>", func=setRFandGateOn)

    self.frame1_RFandGateOff = Radiobutton (frame1, text="RF and Gate OFF", variable=timeVar, value=2, cursor="hand2")
    self.frame1_RFandGateOff.place(in_=frame1,x=70,y=110)
    self.frame1_RFandGateOff.bind("<Button−1>", func=setRFandGateOff)

    self.frame1_RFandGateOff = Radiobutton (frame1, text="RF and Gate OFF", variable=timeVar, value=3, cursor="hand2")
    self.frame1_RFandGateOff.place(in_=frame1,x=70,y=120)
    self.frame1_RFandGateOff.bind("<Button−1>", func=setRFandGateOff)

    self.frame2_RFandGateOff = Radiobutton (frame2, text="RF and Gate OFF", variable=timeVar, value=1, cursor="hand2")
    self.frame2_RFandGateOff.place(in_=frame2,x=70,y=10)
    self.frame2_RFandGateOff.bind("<Button−1>", func=setRFandGateOff)

    self.frame2_RFandGateOff = Radiobutton (frame2, text="RF and Gate OFF", variable=timeVar, value=2, cursor="hand2")
    self.frame2_RFandGateOff.place(in_=frame2,x=70,y=110)
    self.frame2_RFandGateOff.bind("<Button−1>", func=setRFandGateOff)

    self.frame2_RFandGateOff = Radiobutton (frame2, text="RF and Gate OFF", variable=timeVar, value=3, cursor="hand2")
    self.frame2_RFandGateOff.place(in_=frame2,x=70,y=120)
    self.frame2_RFandGateOff.bind("<Button−1>", func=setRFandGateOff)
self.frame1_loadThree.place(in_=frame1,x=50,y=120)
self.frame1_loadThree.bind("<Button-1>", func=setLoadThree)

self.frame1_loadFour = Radiobutton(frame1, text="Open",
variable=load,
value=4, cursor="hand2")
self.frame1_loadFour.place(in_=frame1,x=50,y=150)
self.frame1_loadFour.bind("<Button-1>", func=setLoadFour)

self.frame1_loadFive = Radiobutton(frame1, text="IQ Detector",
variable=load, value=5,
cursor="hand2")
self.frame1_loadFive.place(in_=frame1,x=50,y=180)
self.frame1_loadFive.bind("<Button-1>", func=setLoadFive)

self.frame3_backButton = Button(frame3, text="Quit",
cursor="hand2")
self.frame3_backButton.place(in_=frame3, x=575, y=0)
self.frame3_backButton.bind("<Button-1>", func=exitWindow)

if __name__ == "__main__":
    root = Tk.Tk()
    root.resizable(0,0)
    root.title('Welcome to the Automated Test Suite (ATS)')

    # Center the window in the the display screen
    w , h = 675, 275
    ws, hs = root.winfo_screenwidth(), root.winfo_screenheight()
    x , y = (ws/2) − (w/2), (hs/2) − (h/2)

    root.geometry('%dx%d+%d+%d' % (w, h, x, y))
    w = manualControlsInterface (root)
    root.mainloop()
#!/usr/bin/env python

import Controller
import PowerMeter
import SignalGen
import Database

import math
import time
import string
import os

from TestException import *

offset1_ref = 2.3  # signal generator dBm
offset1_tol = 0.5  # plus/minus 0.5 dB
offset2_ref = 20  # input power dBm
offset2_tol = 1   # plus/minus 1 dB
offset3_ref = 59  # output power
offset3_tol = 1   # plus/minus 1 dB

class OffsetCalibration:
    def __init__(self):
        try:
            # Resetting offsets values in the offsets.ini file
            self.__writeOffsets (0,0,0)
            print "unInitializing Devices...
            self.powermeter = PowerMeter.PowerMeter( True )
            # True to allow offsets.ini not found condition
            self.siggen = SignalGen.SignalGen( True )
            # True to allow offsets.ini not found condition
            self.controller = Controller.Controller()
        except TestException:
            raise

    def destructor(self):
        try:
            del self.powermeter
            del self.siggen
            del self.controller
        except:
            pass

    # ******************************************************************
    # # Public Functions Block
    # # ******************************************************************

    def CalibrateOffset(self):
        # print "[n]Welcome to the Offset Calibration Wizard."
        # print "If you have an SSPA connected, please disconnect it !!!"
        inp = "n"
        while inp != "y" and inp != "q":
            print "InStep 1: Connect 'Amp Input' to 'CABLE B'
            print "InStep 2: Connect 'Amp Input' to 'CABLE B'
            print "InStep 3: Connect 'Amp Input' to 'CABLE B'
            if inp = "q":
                raise TestException (TestException.process_abort_offsetcalib)
            print "InTurning on RF. Please wait ...

        self.controller.setRFHigh()  # Sets pulse mod to high, i.e., on
        power = 999
        self.siggen.setAmplitude(10)
        offset1 = 0
        self.siggen.setRFOn()
        power = self.powermeter.getMeasOutput( offsetcalibration = True )
        print "Power reading : %s dBm (without offset compensation)" % power
        target_power_at_sspa_input = 10
        offset1 = target_power_at_sspa_input − power
        print "Calculated offset for Signal generator: %s dB" % offset1
        target_power_at_sspa_input = 10
        offset2 = target_power_at_sspa_input − chAReading
        print "Corresponding reading in Channel A is %s dBm" % chAReading
        chAReading = self.powermeter.getMeasInput( offsetcalibration = True )
        offset2_ref = 20
        offset2_tol = 1
        offset2 < (offset2_ref − offset2_tol)
        if offset2 > (offset2_ref + offset2_tol):
            TestException (TestException.process_abort_offsetcalib)
            raise (offset2_ref, offset2_tol, offset2)
        self.__saveResults( offset1, offset2)
        self.siggen.setRFOff()
        self.siggen.setAmplitude(target_power_at_sspa_input + offset1)
        power = self.powermeter.getMeasOutput( True )
        offset1_ref = 2.3
        offset1_tol = 0.5
        offset1 < (offset1_ref − offset1_tol)
        if offset1 > (offset1_ref + offset1_tol):
            TestException (TestException.process_abort_offsetcalib)
            raise (offset1_ref, offset1_tol, offset1)
        self.__saveResults( offset1)
        self.siggen.setAmplitude(target_power_at_sspa_input + offset1)
        power = self.powermeter.getMeasOutput( True )
        offset3_ref = 59
        offset3_tol = 1
        offset3 < (offset3_ref − offset3_tol)
        if offset3 > (offset3_ref + offset3_tol):
            TestException (TestException.process_abort_offsetcalib)
            raise (offset3_ref, offset3_tol, offset3)
        self.__saveResults( offset1, offset2, offset3)
        self.siggen.setRFOff()
        self.siggen.setAmplitude(17)  # dBm
        while inp != "y":
            print "Turning on RF, Please wait ...

        self.siggen.setRFOn()
        self.powermeter.zeroPowerHeads()
        self.siggen = SignalGen.SignalGen( True )
        self.controller = Controller.Controller()
chBReading = self.powermeter.getMeasOutput(1)
print "Corresponding reading in Channel B is %s dBm" % chBReading
offset3 = (fullpower) - chBReading
print "Offset for Channel B is %s dB" % offset3

# Sanity check for offset 3
if offset3 < (offset3_ref - offset3_tol) or
   offset3 > (offset3_ref + offset3_tol):
   print "Output Power offset:"
   print "\tExpected: %s, plus/minus %s dB \t Calculated: %s dB"\n   % (offset3_ref, offset3_tol, offset3)
   inp = ""
   while (inp != "y" or inp != "n"):
      inp = raw_input("Do you wish to redo this step? [y/n]")
      inp = string.lower(str(inp))
   if inp == "y":
      continue
   elif inp == "n":
      self.__saveResults( offset1, offset2, offset3 )
      raise TestException (TestException.calib_offsetCalibfailed)
break  # Break While if it passes successfully
self.__writeOffsets(offset1, offset2, offset3)
self.__saveResults(offset1, offset2, offset3);
siggen.setRFOff()
controller.setRFLow()

print "\nOffset Calibration Complete!"
return 1

# **************************************************
# Private Functions Block
# **************************************************

def __writeOffsets(self,siggen, input, output):
   # Writes the offsets to offsets.ini
   import pytz
   import datetime
   UTC = pytz.timezone('UTC')
   mytz = pytz.timezone('America/Los_Angeles')
   my_time = datetime.datetime.utcnow().replace(tzinfo = UTC)
   f = open( os.environ[ 'HOME' ] + '/Code/offsets.ini', 'w')
   f.write("This file contains the offset due to losses and attn in dB.
")
   f.write("Input Offset : %s
")
   f.write("Output Offset : %s
")
   f.write("Sig Gen Offset: %s
")
   f.write("Date and Time Stamp : %s
")
   f.close()

def __saveResults(self, offset_1, offset_2 = 'NULL', offset_3 = 'NULL'):
   offsets = [ offset_1, offset_2, offset_3 ]
   names = [ ]
   values = [ ]
   cnt = 1
   for i in offsets:
      names.append( "offset_%s" % cnt )
      values.append( "%s" % i )
      cnt += 1
   names_str = string.join(names, ',')
   values_str = string.join(values, ',')
   db = Database.Database()
   db.storeOffsetSettings (names_str, values_str)
#!/usr/bin/env python

import Controller
import PowerMeter
import PowerSupply
import SignalGen
import TempSensor
import Database
import time
import string
import sys
from TestException import *
from Utilities import *

class OperationalTest:
    def __init__(self, uName):
        try:
            print "Initializing Devices..."
            self.powersupply = PowerSupply.PowerSupply()
            self.powermeter = PowerMeter.PowerMeter()
            self.siggen = SignalGen.SignalGen()
            self.sensor = TempSensor.TempSensor()
            # Make sure this is initialized last
            self.controller = Controller.Controller()
            self.user = uName
        except Exception:
            raise

    # *********************************************************************
    # Public functions
    # *********************************************************************
    def destructor(self):
        try:
            del self.sensor
            del self.controller
            del self.powersupply
            del self.siggen
        except:
            pass

def runOperationalTest(self):
    values = {
        "sn" : "",
        "uname" : "",
        "label" : "",
        "current" : "",
        "power_in" : "",
        "power_out" : "",
        "nans" : "",
        "temp" : "",
        "tempflag" : "",
        "vswrflag" : ""  
    }  # Values dictionary

    serialno = ""
    while serialno == ":"
        sn = raw_input( "Please enter the Serial Number of the amplifier:" )
        serialno = str(sn).strip().replace(" ",")

    label = ""
    while label == "":
        print "Please enter a label for this test run:"
        label = label.strip().replace(" ",")

        values["an"] = "%s" % serialno
        values["uname"] = "%s" % self.user
        values["label"] = "%s" % label

        how_many_minutes = int (raw_input("How many minutes to run?:"))
        setStyle( "bold", "white" )
        print "Initializing Environmental Test Procedures..."
        setStyle( "default" )
        self.controller.setTR(0)
        self.siggen.setRFOff()  # From signal generator
        self.powersupply.setPowerOff()
        self.powersupply.setVolt(32)
        self.powersupply.setCurr(1)
        self.controller.setLoad(self.controller.load_50_ohms)
        self.controller.setAddr(0)
        self.powersupply.setPowerOff()
        current = self.powersupply.getCur()  
        print "Initial current of ", current, ", Amps"
        print "Starting TR only ..."
        self.controller.setTR(1)  # only TR running at default dutycycle
        current = self.powersupply.getCur()
        print "Current measurement of "+ current + "Amps with only TR running..."

        if (current > 0.5):
            print "Excess initial current drawn... Test Cannot proceed ...
            self.__saveCalibrationResults(values, 0, TestException.calib_initialCurrFailed)
            raise TestException(TestException.calib_initialCurrFailed)
        elif (current < 0):
            print "No SSPA Connected... Test Cannot proceed ...

            # Set Current to 5A
            print "Self test passed... Setting current to 5 Amps..."
            self.powersupply.setPowerOff()
            self.powersupply.setVolt(32)
            self.powersupply.setCurr(5)
            self.powersupply.setPowerOn()
            current = self.powersupply.getCur()
            print "Initial current of ", current, ", Amps"
            if (current > 0.5):
                print "Excess initial current drawn... Test Cannot proceed ...
                self.__saveCalibrationResults(values, 0, TestException.calib_initialCurrFailed)
                raise TestException(TestException.calib_initialCurrFailed)
            elif (current < 0):
                print "No SSPA Connected... Test Cannot proceed ...

                # Set Current to 5A
                print "Self test passed... Setting current to 5 Amps..."
                self.powersupply.setPowerOff()
                self.powersupply.setVolt(32)
                self.powersupply.setCurr(5)
                self.powersupply.setPowerOn()

                print "Starting TR only ..."
                self.controller.setTR(1)  # only TR running at default dutycycle
                current = self.powersupply.getCur()
                print "Current measurement of "+ current + "Amps with only TR running..."

                if (current > 0.5):
Too much current drawn when turning on TR only..." + 
"Test Cannot proceed ...
raise TestException(TestException.calib_gatePulseFailed)

Test Cannot proceed ...

return self.__saveCalibrationResults(values, 
0, TestException.calib_gatePulseFailed)

print "Waiting for RF to turn on"
time.sleep(2)
# Start RF signal
self.siggen.setRFOn()

# Start RF Input and TR gating
print "Turning RF gate on..."
self.controller.setRF(1)
# RF and TR running at default values
time.sleep(5)

while True:
    minutes = 0
    sys.stdout.write( "
Acquiring Data ... 		"
    sys.stdout.flush()
    current  = self.powersupply.getCurr()
    powerin  = self.powermeter.getMeasInput()
    powerout = self.powermeter.getMeasOutput()
    values['current'] = ("%s" % current)
    values['power_in'] = ("%s" % powerin)
    values['power_out'] = ("%s" % powerout)
    values['mins'] = minutes
    values['temp'] = self.sensor.getTemperature()
    values['tempflag'] = ("%s" % self.controller.getTemp_Flag())
    values['vswrflag'] = ("%s" % self.controller.getVSWR_Flag())
    sys.stdout.write( "
Saving Data ...
"
    sys.stdout.flush()
    print "Power in :%s, Power out :%s, Current :%s"
    (powerin, powerout, current)
    # Database save call
    self.__saveOvenTestResults(values)
    # Check for break condition
    if minutes ≥ how_many_minutes:
        break
    # Go back to the loop
    seconds = 0
    while seconds < 60:
        str_time = str(minutes) + ' minutes and ' + str(seconds) + ' seconds...'
        sys.stdout.write( "\nWaiting:%s"
        sys.stdout.flush()
        time.sleep(1)
        seconds += 1
        minutes += 1
    # End of Oven Test Procedure
print "Oven Test Completed !!!"
print 35 * '-'

# **************************************************
# Private functions
# **************************************************
def __saveOvenTestResults(self, results, teststatus=1, failcode='NULL'):
    # By default the test passes and the fail code is 'NULL'
    names = [ ]
    values = [ ]
    for fieldname, value in results.items():
        names.append("%s" % fieldname)
        values.append("%s" % value)
    values_str = string.join(values, ',
    names_str = string.join(names, ',')
    db = Database.Database()
    db.storeOvenTestReadings(names_str, values_str)

if __name__ == '__main__':
    OvenTest("kaushal").OvenTest()
#!/usr/bin/env python

# PowerMeter Class is the direct connection between the main computer and
# the Agilent Power Meter.
# This is the class that should be used to interface with the power meter
# This class implements various functions of the power meter using the SCPI
# command set.

from visa import *
from TestException import *
from Utilities import *

import sys
import time
except TestException, e:
    self.__setMeasurementSetting( "PEAK", "IMM" )
    self.__setTriggerDelay( triggerdelay )
    self.setFreq( 440e6 )
    self.__setTraceUnits()        self.__setCaptureRate( "DOUBLE" )
    self.__setTriggerDelay(100e-6)  
    timeout = 15
    while timeout > 0:
        sys.stdout.write( "Please wait for %2s seconds to fetch %s...
        %s " % ( timeout, timelength, self.device.ask("TRAC2:STAT?")) )
        sys.stdout.flush()            time.sleep(1)
        timeout -= 1
    sys.stdout.write( "%s " % timelength )
    sys.stdout.flush()

except Exception:
    self.__setTriggerDelay(100e-6)        self.device.write("SENS2:TRAC:TIME %s")
    self.__setCaptureRate( "DOUBLE" )
    self.__setTriggerDelay(100e-6)

 except Exception:
    raise TestException(TestException.invalid_offsets)

def getTraceData(self, timelength = .02, triggerdelay = 0):
    # Get the trace data on channel B
    # Pass the length of time you want to capture as time
    # By default, captures only 1 pulse (20ms)
    self.setFreq( 440e6 )
    self.__setTraceUnits()
    self.__setCaptureRate( "DOUBLE" )
    self.__setTriggerDelay(100e-6)
    timeout = 15
    while timeout > 0:
        sys.stdout.write( "Please wait for %2s seconds to fetch " + 
        "the trace...
        %s " % ( timeout, timelength) )
        sys.stdout.flush()            time.sleep(1)
        timeout -= 1
    sys.stdout.write( "%s
" % timelength )
    sys.stdout.flush()

    # Medium resolution decimates the data to 1000 points
    self.device.write("TRAC2:DATA? MRES")
result = self.device.read_values()

# Reset back
self.device.write("INIT1:CONT 1")
self.device.write("INIT2:CONT 1")
self.device.write("TRIG:SEQ1:SOUR EXT")
self.device.write("TRIG:SEQ2:SOUR EXT")
sel.\_setTriggerDelay(100e-6)

result_offset_comp = []
for i in result:
    result_offset_comp.append(i)
return result_offset_comp

def zeroPowerHeads(self):
    self.device.write("CAL1:ZERO:NORM:AUTO ONCE")
time.sleep(3)
self.device.write("CAL2:ZERO:NORM:AUTO ONCE")
time.sleep(3)

def setFreq(self, freq = 440e6):
    # set the frequency to 440 MHz BY DEFAULT
    self.device.write("SENS1:FREQ %sHz" % freq)
    self.device.write("SENS2:FREQ %sHz" % freq)

# ******************************************************
# Private functions
# ******************************************************
def _setGain(self, gainA, gainB):
    self.device.write("SENS1:CORR:GAIN2 %s" % gainA)
    self.device.write("SENS2:CORR:GAIN2 %s" % gainB)

def _setCaptureRate(self, mode = "DOUBLE"):  
    self.device.write("SENS1:MRAT %s" % mode)
    self.device.write("SENS2:MRAT %s" % mode)

def _setMeasurementSetting(self, pow="PEAK", mode="CONT", trigger ="EXT"):
    # mode is acquisition mode and takes in "CONT", "SING" or "FREE"
    # trigger is triggering source and takes in "EXT" or "IMM"
    import string
    mode = string.upper(mode)
    trigger = string.upper(trigger)
    pow = string.upper(pow)
    if mode =="CONT" ∨ mode =="FREE":  
        # continuous trigger mode with external or imm trigger
        self.device.write("INIT1:CONT 1")
        self.device.write("INIT2:CONT 1")
    elif mode =="SING":  
        # single trigger run mode with internal trigger
        self.device.write("TRIG:SEQ1 CONT 0")
        self.device.write("TRIG:SEQ2 CONT 0")
    self._setPowMeasurement(pow)
    self._setTrigger(trigger)

    def __setTriggerDelay(self, delay=100e-6):
        self.device.write("TRIG:SEQ:DEL %s" % delay)

def __setTraceUnits(self):
    # Sets the trace function units on channel A and B
    self.device.write("TRAC1:UNIT DBM")
    self.device.write("TRAC2:UNIT DBM")

def __enableTrace(self):
    # Enable trace capture for channel B (output)
    self.device.write("TRAC2:STAT 1")

    def __setPowMeasurement(self, type="PEAK"):  
        # Setting for peak power or average power measurement
        import string
        type = string.upper(type)
        if type ≠ "PEAK" ∨ type ≠ "AVER":
            print "Invalid option"
            return
        self.device.write("CALC1:FEED \"POW:%s\"")
        self.device.write("CALC2:FEED \"POW:%s\"")

    def __setTrigger(self, trigger="EXT"):  
        # Configures trigger system to respond to
        # immediate or external trigger for both channels
        self.device.write("TRIG:SEQ1:SOUR %s")
        self.device.write("TRIG:SEQ2:SOUR %s")

    def __waitTrigger(self):
        # Places channel 1 & 2 in wait for trigger state
        self.device.write("INIT1:IMM")
        self.device.write("INIT2:IMM")

    def __getOffsetInput(self):
        # Gets dB loss offset for input stage
        return float(values[1])

    def __getOffsetOutput(self):
        # Gets dB loss offset for output stage
        return float(values[1])
PowerSupply.py

# Wait until the output voltage has reached what it has been set to
# Typical value is 32V
# Time delay to wait for Power Supply to be ready

set = self.getVoltageSetpoint()
time.sleep( 0.25 )
meas = self.getVoltageReading()
diff = abs( meas - set )
lim = 0.1

if had_to_loop:
    msg = "VoltageSetpoint: %4.1f V, Reading: %4.1f V, Diff = %4.1f V > %.1f V 
  % ( set, meas, diff, limit )

    sys.stdout.write( "%s

    sys.stdout.flush()

    if had_to_loop:
        msg = "VoltageSetpoint: %4.1f V, Reading: %4.1f V, Diff = %4.1f V > %.1f V 
  % ( set, meas, diff, limit )

        sys.stdout.write( "%s

        sys.stdout.flush()

    def setPowerOff(self):
        # Disable output
        self.socket.send('OUTP OFF, (@1)@m')

    def getPowerStatus(self):
        # Get Power Status
        self.socket.send('OUTP?(@1)@m')
        return float(self.socket.recv(100))

    # Enable output and let the caps charge
    self.socket.send('OUTP ON, (@1)@m')
#!/usr/bin/env python

# SignalGen Class is the direct connection between the main computer and
# the Agilent Signal Generator.
# This is the class that should be used to interface with the signal generator
# This class implements functionality of turning RF power on and off from the
# signal generator.

import os
import sys
import time
import socket

from TestException import *
from Utilities import *

class SignalGen:
    
    """Controls the Signal Generator"
    """
    
    IP_ADDR = "192.168.0.3"
    SOCKET_NO = 5025

    def __init__(self, offsets_ini_not_found_allow = False):
        if offsets_ini_not_found_allow == False:
            # offsets_ini_not_found_allow is False by default
            # When called from the PowerMeter Offset calibration routine
            # we don’t have to check for the presence of the file.
            try:
                self.siggenoffset = self.__getOffsetOutput()
            except IOError:
                raise TestException(TestException.offsets_ini_not_found)
        else:
            self.siggenoffset = 0
            # Connect to the signal generator via ethernet on mfg. default port 5025
            self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

        self.socket.connect((self.IP_ADDR, self.SOCKET_NO))
        self.setRFOff()
        self.setAmplitude(10)
        self.setFreq(440e6)
        self.socket.send('FREQ '+str(440e6)+'
' )

    def __del__(self):
        try:
            self.setRFOff()
        del self.socket
        except Exception:
            raise

    def setRFOff(self):
        #disable RF output from signal generator
        self.socket.send('OUTP OFF
' )
        time.sleep(2)

    def setRFOn(self):
        #enable RF output from signal generator
        self.socket.send('OUTP ON
' )
        time.sleep(2)

    def setAmplitude(self, val):
        #Sets the power (amplitude) designated by val in dBm
        val += self.siggenoffset
        self.socket.send('POW '+str(val)+' dBm
' )

    def setFreq(self, val):
        #Sets the frequency designated by val in Hz
        #val takes in a real number between 250e3 to 1e9
        val += self.siggenoffset
        self.socket.send('FREQ '+str(val)+' Hz
' )

    def __getOffsetOutput(self):
        # Gets dB loss offset for output stage
        f = open( os.environ[ 'HOME' ] + '/Code/offsets.ini' )
        f.readline() # By pass first 3 lines
        f.readline()
        f.readline()
        values = f.readline().split(':
' )
        return float(values[1])
Pre Burnin Test
1. OverPulse Test
2. Droop Test
3. Power Efficiency Test
4. Power Correct Test
5. Open Load Test
6. Mismatch Load Test

```python
def runPreBurninTest(self, serialno = ""):
    values = {
        "sn" : "",
        "uname" : "",
        "pulsewidth" : "",
        "overpulse_passed" : "",
        "droop_initpower" : "",
        "droop_finalpower" : "",
        "droop_passed" : "",
        "current_450" : "",
        "voltage_450" : "",
        "power_450" : "",
        "current_440" : "",
        "voltage_440" : "",
        "power_440" : "",
        "current_430" : "",
        "voltage_430" : "",
        "power_430" : "",
        "power_8_5" : "",
        "power_11_5" : "",
        "powercorrect_passed" : "",
        "open_passed" : "",
        "mismatch_passed" : "",
        "test_passed" : "",
        "fail_reason" : ""
    }
    testFailFlag = False
    testFailCode = []
    while serialno == "":
        sn = raw_input("Please enter the Serial Number of the SSPA : ")
        serialno = str(sn).strip().replace("", "_")

    values['sn'] = "%s" % serialno
    values['uname'] = "%s" % self.user

    print "Initializing Pre Burnin Test Porcedures..."
    self.initDevices()
    overpulse = self.conductOverPulseTest()
    values["pulsewidth"] = "%s" % string.join(overpulse[0], ",")
    values["overpulse_passed"] = "%s" % str(bool(overpulse[1]))
    if bool(overpulse[1]) == True:
        testFailFlag = True  # Returns fail code as 1 if fails
        testFailCode.append (fail_overpulse)
```

# Droop

(initpower, finalpower, passed) = self.conductDroopTest()
values['droop_initpower'] = "%s" % initpower
values['droop_finalpower'] = "%s" % finalpower
values['droop_passed'] = "%s" % passed
if droop_passed == False:
    testFailFlag = True
testFailCode.append(fail_droop)

# Efficiency at 430, 440, 450

for i in [430, 440, 450]:
    # indexes 0,1,2 (430, 440, 450)
    values['current_%s' % i] = "%s" % currents[i/10 - 43]
    values['voltage_%s' % i] = "%s" % voltages[i/10 - 43]

# Efficiency Test before burnin ...

def runPostBurninTest(self, serialno = ""):  
    values = {   
        'sn'                 : "
        'uname'              : "
        'pulsewidth'         : "
        'overpulse_passed'   : "
        'droop_initpower'    : "
        'droop_finalpower'   : "
        'droop_passed'       : "
        'current_430'        : "
        'voltage_430'        : "
        'power_430'          : "
        'current_450'        : "
        'voltage_450'        : "
        'power_450'          : "
        'current_440'        : "
        'voltage_440'        : "
        'power_440'          : "
        'power_8_5'          : "
        'power_11_5'         : "
        'powercorrect_passed': "
        'open_passed'        : "
        'mismatch_passed'    : "
        'test_passed'        : "
        'fail_reason'        : "
    }
    if bool(testFailFlag) == True:
        testFailCode.append(fail_open)
        testFailFlag = False
        testFailCode.append(fail_mismatch)
    
# Open and Mismatch

variousloads_passed = self.conductVariousLoadsTest()
    
# End of PreBurnin Procedure

print "\n" if testFailFlag == True:
values['uname'] = "%s" % self.user

print "Initializing Post Burnin Test Procedures..."

# Initialize devices
self.initDevices()

# Overpulse with 3ms pulse
overpulse = self.conductOverPulseTest()
values['pulseshift'] = "%s" % string.join(overpulse[0], ",")
values['overpulse_passed'] = "%s" % str(not bool(overpulse[1]))
if bool(overpulse[1]) == True: # Returns fail code as 1 if fails
testFailFlag = True
testFailCode.append(fail_overpulse)

# Droop
(initpower, finalpower, passed) = self.conductDroopTest()
values['droop_initpower'] = "%s" % string.join(initpower[0], "\n")
values['droop_finalpower'] = "%s" % finalpower
values['droop_passed'] = "%s" % str(bool(finalpower[1]))
if bool(finalpower[1]) == False: # Fail bit
    testFailFlag = True
testFailCode.append(fail_droop)

# Efficiency at 430, 440, 450
(currencts, voltages, powers, failflag) = self.conductEfficiencyTest()
for i in [430, 440, 450]: # indexex 0,1,2 (430, 440, 450)
    values['current_%s' % i] = "%s" % currents[i/10 - 43]
    values['voltage_%s' % i] = "%s" % voltages[i/10 - 43]
    values['power_%s' % i] = "%s" % powers[i/10 - 43]

if bool(failflag) == True: # Returns fail code as 0 if passed
    sys.stdout.write( "Efficiency Test after burnin ...\n")
else:
    # Failed
    setStyle('bold', 'red')
    sys.stdout.write("[ FAILED ]")
    testFailFlag = True
    testFailCode.append(fail_mismatch)

# Power Correct
powercorrect = self.conductPowerCorrectTest()
values['power_8.5'] = powercorrect[0][0]
values['power_11.5'] = powercorrect[0][1]
values['power_corrected'] = "%s" % powercorrect_passed
    str(not bool(powercorrect_passed))
if bool(powercorrect[1]) == True: # Pass bit
    testFailFlag = True
    testFailCode.append(fail_powercorrect)

# Open and Mismatch
variousloads_passed = self.conductVariousLoadsTest()
values['openedPassed'] = "%s" % str(not bool(variousloads_passed[0]))
values['mismatchPassed'] = "%s" % str(not bool(variousloads_passed[1]))
if bool(variousloads_passed[0]) == False: # Fail bit
    testFailFlag = True
    testFailCode.append(fail_open)
if bool(variousloads_passed[1]) == False: # Fail bit
    testFailFlag = True

print "%s\nConducting Overpulse test ...								
Conducting Droop test ...<\nprint "%s\nPostBurnin Test Failed !!!"
print 35 * '-'
setStyle('default')

# End of PostBurnin Procedure
print "%s\nPostBurnin Test Completed !!!"
print 35 * '-'
setStyle('default')

# Efficiency at 430, 440, 450
(currencts, voltages, powers, failflag) = self.conductEfficiencyTest()
for i in [430, 440, 450]: # indexex 0,1,2 (430, 440, 450)
    values['current_%s' % i] = "%s" % currents[i/10 - 43]
    values['voltage_%s' % i] = "%s" % voltages[i/10 - 43]
    values['power_%s' % i] = "%s" % powers[i/10 - 43]

if bool(failflag) == True: # Returns fail code as 0 if passed
    sys.stdout.write("Efficiency Test after burnin ...	\n")
else:
    # Passed
    setStyle('bold', 'green')
    sys.stdout.write("[ PASSED ]")
    testFailFlag = bool(failflag)
    Efficiency Test after burnin ...								sys.stdout.write("\n
# Setup Power Supply with 32 V and 5 A
# We will not be turning off the power supply in the
# middle of sequence of tests
# "
self.powersupply.setPowerOff()
self.powersupply.setCurr(5)
self.powersupply.setVolt(32)
self.powersupply.setPowerOn()

# Conducts Overpulse test
if testFailFlag == True:
    setStyle('default')
    # Conducts Overpulse test
if bool(overpulse[1]) == True:
    testFailFlag = True
    testFailCode.append(fail_overpulse)

# Conducts Droop test
if bool(droop_max) == True:
    testFailFlag = True
    testFailCode.append(fail_mismatch)
initpower, finalpower = self.__DroopTest()  # dBm
power_init_mW = 10 ** initpower / 10.0
power_final_mW = 10 ** finalpower / 10.0

sys.stdout.write("Droop test ...
# −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−")
if power_final_mW > ((1 - droop_max) * power_init_mW):
    droop_passed = 'f'
sys.stdout.write("[ PASSED ]")
else:
    droop_passed = 'f'

# Note the power is converted into Watts
power_init_mW = 10 ** initpower / 10.0
power_final_mW = 10 ** finalpower / 10.0

eff_0 = (10 ** (powers[0] / 10)) / 10000 / (currents[0] * voltages[0]) * 100
eff_1 = (10 ** (powers[1] / 10)) / 10000 / (currents[1] * voltages[1]) * 100

if (eff_0 ≥ efficiency ∧ eff_1 ≥ efficiency ∧ eff_2 ≥ efficiency):
    droop_passed = 'f'
    setStyle('bold', 'green')
    sys.stdout.write("[ PASSED ]")
else:
    # Failed
    failflag = fail_power_430
    ef_0 < efficiency):
# shutoff capability of the SSPA.

```python
traceLength = 0.100  # seconds
period = 40e−3  # seconds
pw = 3e−3  # 3ms pulse
outputPulseLength = []

# Turn off Signal Gen and Gating
self.siggen.setRFOff()  self.controller.setTR(0)
# Settings
self.controller.setAddr(0)  self.controller.setLoad(self.controller.load_50_ohms)
# Turn on Signal Generator
self.siggen.setFreq(440e6)  self.siggen.setAmplitude(10)  self.siggen.setRFOn()
# Conduct Test
self.controller.setRF(1, period, pw)  # start the rf
time.sleep( warmup_period )
try:
    # get trace data traceLength in seconds
    traceData = self.powermeter.getTraceData(traceLength)
except Exception:
    sys.stdout.write( "Fetching trace				"
setStyle( 'bold', 'red' )
print( "[ FAILED ]"
setStyle( 'default' )
raise
else:
    sys.stdout.write( "Fetching trace				"
setStyle( 'bold', 'green' )
print( "[ OK ]"
setStyle( 'default' )

failstate = 0  # initial state = pass
i = 0  # pulse index
while i < len(traceData) − 5:  # checks every data point (except the last few to avoid false failure)
    count = 0  # initial measured output pulse width is 0
    while traceData[i + count] > 45:  # count how long the output pulse is
        count += 1
    if count ≠ 0 :  # convert number of samples to time
        outputPulseLength.append(str(count*(traceLength/1000)))
if float(outputPulseLength[index]) > 2.5e−3:  # fail SSPA if output pulse width too long
    failstate = 1
    print "Pulse length over 2.5 ms => %s ms" % outputPulseLength[index]
    index += 1  # check next pulse
i += count + 1

# Turn off RF Gating
self.controller.setTR(0)
# Turn off Signal Generator
self.siggen.setRFOff()
return (outputPulseLength, failstate)
```

```
def __PowerCorrectTest(self):
    ""
    Check for the correctness of the power
    ""
    # This test verifies that the SSPA can compensate for
    # variable input powers
    # The input power varies by 10% and the output power
    # should remain at or above 57dBm
    # Turn Signal Gen and Gating
    self.siggen.setRFOff()  self.controller.setTR(0)
    # Turn on Signal Generator
    self.siggen.setFreq(440e6)  self.siggen.setAmplitude(10)  self.siggen.setRFOn()
    # Conduct Test
    self.controller.setRF(1)  # start the rf
time.sleep( warmup_period )
    measured = []
    fail = 0
    for k in range(0, 3, 2):
        power_input = 8.5 + (k*1.5)
        self.siggen.setAmplitude( power_input )
        self.controller.setRF(1)
        print "Setting input power to %s dBm..." % power_input
        time.sleep( warmup_period )
        power_output = self.powermeter.getMeasOutput()
        print "Output Power read is %s dBm" % power_output
        measured.append( power_output )
        self.controller.setRF(0)
        if power_output < 56.8:
            fail = 1
            time.sleep(.3)
    self.controller.setTR(0)
    self.siggen.setRFOff()
    result = [measured, fail]
    return

def __DroopTest(self):
    ""
    Checks for the power droop of the output pulse
    ""
    # This test supplies 2ms pulses at 50Hz (40ms cycle) to the SSPA
    # to test the SSPA's Output Power Droop
    traceLength = 0.0025  # seconds
    period = 20e−3  # seconds
    pw = 2e−3  # 2ms pulse
droop = []

    # Turn off Signal Gen and Gating
    self.siggen.setRFOff()  self.controller.setTR(0)
    # Settings
    self.controller.setAddr(0)
```

```
Feb 27, 07 16:26 SpecTests.py Page 10/13

Check for the correctness of the power
***
# This test verifies that the SSPA can compensate for
# variable input powers
# The input power varies by 10% and the output power
# should remain at or above 57dBm
# Turn Signal Gen and Gating
self.siggen.setRFOff()  self.controller.setTR(0)
# Turn on Signal Generator
self.siggen.setFreq(440e6)  self.siggen.setAmplitude(10)  self.siggen.setRFOn()
# Conduct Test
self.controller.setRF(1)  #start the rf
time.sleep( warmup_period )
measured = []
fail = 0
for k in range(0, 3, 2):
    power_input = 8.5 + (k*1.5)
    self.siggen.setAmplitude( power_input )
    self.controller.setRF(1)
    print "Setting input power to %s dBm..." % power_input
    time.sleep( warmup_period )
    power_output = self.powermeter.getMeasOutput()
    print "Output Power read is %s dBm" % power_output
    measured.append( power_output )
    self.controller.setRF(0)
    if power_output < 56.8:
        fail = 1
        time.sleep(.3)
self.controller.setTR(0)
self.siggen.setRFOff()
result = [measured, fail]
return

def __DroopTest(self):
    ""
    Checks for the power droop of the output pulse
    ""
    # This test supplies 2ms pulses at 50Hz (40ms cycle) to the SSPA
    # to test the SSPA's Output Power Droop
    traceLength = 0.0025  # seconds
    period = 20e−3  # seconds
    pw = 2e−3  # 2ms pulse
droop = []

    # Turn off Signal Gen and Gating
    self.siggen.setRFOff()  self.controller.setTR(0)
    # Settings
    self.controller.setAddr(0)
```

```
Feb 27, 07 16:26 SpecTests.py Page 9/13

# shutoff capability of the SSPA.

```
def _EfficiencyTest(self, freq):
    # This test supplies RF signal at the given freq to the SSPA
    # to test the SSPA's Output Power efficiency
    # cold switching done in the setLoad() method
    self.powermeter.setFreq(freq)
    self.siggen.setFreq(freq)
    self.siggen.setAmplitude(10)
    self.powermeter.setFreq(freq)
    self.siggen.setRFOn()

    # Turn off Signal Generator
    self.siggen.setRFOff()
    self.controller.setTR(0)
    self.controller.setAddr(0)
    self.controller.setLoad(self.controller.load_open)
    self.controller.setRF(1)
    time.sleep(warmup_period)
    current = self.powersupply.getCurr()
    voltage = self.powersupply.getVoltageReading()
    power = self.powermeter.getMeasOutput()
    if current > 1:
        open_passed = False
        print("[ Failed ]")
        print("Power found to be %s dBm")
        setStyle("default")
    else:
        open_passed = True
        print("[ PASSED ]")
        setStyle("default")

    # Load :: VSWR Mismatch 3:1
    self.controller.setLoad(self.controller.load_3_1_mismatch)
    self.controller.setRF(1)
    vswr_flag = self.controller.getVSWR_Flag()
    if vswr_flag == 1:
        print("[ Passed ]")
        setStyle("default")
    else:
        print("[ Failed ]")
        setStyle("default")
vswr_passed = False
setStyle( 'bold', 'red' )
print "[Failed]"
setStyle( 'default' )
else:
vswr_passed = True
setStyle( 'bold', 'green' )
print "[PASSED]"
setStyle( 'default' )

# Turn off RF Gating
self.controller.setTR(0)

# Turn off Signal Generator
self.siggen.setRFOff()

return [open_passed, vswr_passed]

############################################################
## Data Saving Routines
############################################################
def __savePreBurninResults(self, results):
    
    """Formats the Pre Burnin data for the database adapter"
    
    names = []
    values = []
    for fieldname, value in results.items():
        names.append( "%s" % fieldname )
        values.append( "%s" % value )
    names_str = string.join(names, ',')
    values_str = string.join(values, ',')
    db = Database.Database()
    db.storePreBurninReadings(names_str, values_str)

def __savePostBurninResults(self, results):
    
    """Formats the Post Burnin data for the database adapter"
    
    names = []
    values = []
    for fieldname, value in results.items():
        names.append( "%s" % fieldname )
        values.append( "%s" % value )
    names_str = string.join(names, ',')
    values_str = string.join(values, ',')
    db = Database.Database()
    db.storePostBurninReadings(names_str, values_str)

# ********************************************************************
# End of Private Functions Block
# ********************************************************************
#!/usr/bin/env python

import Calibration
import SpecTests
import OvenTest
import OffsetCalibration

import string
import os
import sys
import time

try:
    serialno = -1
    proc = Calibration.Calibration(self.UserName)
    serialno = proc.runCalibration()
    if serialno != -1:
        print "\n" + _msg
finally:
    proc.destructor()

if __name__ == "__main__":
    SSAPI_Test().run()
#!/usr/bin/env python
import sys
import re
import time
import string
import urllib2
importdatetime
import pytz
from Utilities import *
from TestException import *

class TempSensor:
    """
    TempSensor does a HTTP GET on a URL to capture a string of temperatures
    from a probe located at Toolik Alaska, and then stores the results
    in a database through the supporting Database class
    """

    sensor = "192.168.0.4"

    def __init__( self ):
        """
        """

    def getTemperature( self ):
        sensor_reading = self.__getSensaTronicTemps( self.sensor )
        return sensor_reading

    def __getSensaTronicTemps( self, sensor ):
        """
        Return a dict[ probe_name ] = probe_value
        The probe_values are assumed to be in F.
        An unconnected sensor port will report ~99.9. It is worth
        reporting these in case a wire comes loose, however if we have
        a 16 port sensor with only 2 ports in use then it is foolish
        to report the other 14 ports all the time. So we need some way
        of determining of a port is actually in use.

        We count on the configuration of the sensor for this. The default
        probe name is 'probeN'. If we find a default name then we assume
        that the port is not in use and don't bother
        with it's ~99.9 reading. As a check to this we will log a
        warning to any actual value found on the port with a default
        name.
        """

        probe_re = re.compile( "Probe \d\d\" )

        # SensaTronic's embedded HTTP server offers the page "temps"
        # to get the current readings.
        url = "http://%s/temp" % sensor

        try:
            fid = urllib2.urlopen(url)
        except urllib2.HTTPError, e:
            msg = "HTTPError: %s" % e
            print msg
            return None
        except urllib2.URLError, e:
            msg = "URLError: %s" % e
            print msg
            return None

        try:
            data = fid.read()
        except:
            return None

        fid.close()

        # "SwitchRoom|reading0|EngineRoom|reading1|probe2|reading2"
        fields = data.split( '|' )

        # Make sure we have an even number of fields, name|reading pairs
        n_fields = len( fields )
        if n_fields % 2 != 0:
            msg = "Odd number of fields in "%s" % data
            print msg
            return None

        # Our dict to return
        sensor_readings = {}

        # Separate the probes
        n_probes = n_fields / 2
        for i in range( n_probes ):
            probe_name = fields[ i * 2 ]
            probe_reading_str = fields[ i * 2 + 1 ]
            if probe_re.match( probe_name ):
                # Default probe name so we assume not in use
                # But should also have no legitimate value.
                if probe_reading_str != "-99.9":
                    probe_reading = float( probe_reading_str )
                    probe_reading = (probe_reading - 32) * 5 / 9.0
                    return probe_reading
                msg = "Actual reading (%s) on un-named probe %s
                ( probe_reading_str, probe_name )
                print msg
            else:
                sensor_readings[ probe_name ] = probe_reading

        return -99.9

    if __name__ == "__main__":
        print TempSensor().getTemperature()
Class that handles the TestResults (The Calibration process and the actual test) along with the Handling of various exceptions.

```python
class TestException (Exception):
    device_error_PowerSupply = 100  # Cannot handshake with the Power Supply
    device_error_PowerMeter = 101   # Cannot handshake with the Power Meter
    device_error_SignalGen = 103    # Cannot handshake with the Signal Generator
    device_error_DAQCard = 102      # Cannot handshake with the DAQ
    device_error_SSPA_Power = 104   # Cannot handshake with the SSPA
    calib_initialCurrFailed = 200   # Initial current above 0.5 A during Cal
    calib_gatePulseFailed = 201     # Current is above 0.5A with only gate pulse turned on
    calib_normalCurrFailed = 202    # Current is out of range
    calib_driverCurrFailed = 203    # Current out of bounds Pot 1 (Driver Bias)
    calib_outputCurrFailed = 204    # Current out of bounds Pot 2 (Output Bias)
    calib_outputPowerFailed = 205   # Power out of bounds Pot 3 (Power Level)
    calib_phaseFailed = 206        # Phase out of bounds Pot 4 (Phase)
    calib_vswrFailed = 207         # VSWR out of bounds Pot 5 (VSWR)
    calib_offsetCalibFailed = 208  # Power Meter Offset Calibration failed
    offsets_ini_not_found = 300     # Offsets.ini file not found
    process_abort_calib = 500       # Abort Calibration Process
    process_abort_offsetcalib = 501 # Abort Offset Calibration Process
    invalid_data_type = 700         # Invalid parameter trapped
    invalid_offsets = 701          # Invalid values for offsets found in offsets.ini file

    # Key:Value pair for device errors and corresponding strings
    eStr = {device_error_PowerSupply : "Cannot establish connection to the Power Supply.
Please check the connections and the power is turned on.",
    device_error_PowerMeter : "Cannot establish connection to the Power Meter.
Please check the connections and the power turned on.",
    device_error_SignalGen : "Cannot establish connection to the Signal Generator.
Please check the connections and the power turned on.",
    device_error_DAQCard : "Driver issue with DAQ Card.
Please contact the administrator.",
    device_error_SSPA_Power : "Cannot establish connection to the SSPA.
Please check the power connections.",
    calib_initialCurrFailed : "Initial current during calibration is above 0.5A",
    calib_gatePulseFailed : "Current exceeds 0.5A with only gate pulse turned on",
    calib_normalCurrFailed : "Current is out of bounds when setting Address 1 (Driver Bias)",
    calib_driverCurrFailed : "Current out of bounds while setting Address 2 (Output Bias)",
    calib_outputCurrFailed : "Power out of bounds while setting Address 3 (Power Level)",
    calib_outputPowerFailed : "Phase out of bounds while setting Address 4 (Phase)",
    calib_phaseFailed : "Phase out of bounds while setting Address 5 (Phase)",
    calib_vswrFailed : "VSWR out of bounds while setting Address 6 (VSWR)",
    calib_offsetCalibFailed : "Offset Calibration failed.
Please check the connections, & the parameters in offsets.ini",
    offsets_ini_not_found : "Offsets.ini file not found.
Please run the Power Meter Offset Calibration Procedure.",
    process_abort_calib : "Aborting Calibration Process...",
    process_abort_offsetcalib : "Aborting Offset Calibration Process...",
    invalid_data_type : "Invalid parameter trapped",
    invalid_offsets : "Invalid values for offsets found in offsets.ini file."

    def __init__(self, value):
        self.errno = value
    def __str__(self):
        try:
            return str(self.eStr[self.errno])
        except:
            return "Unexpected and unhandled error. %s" % self.errno
```
```
import sys

__styles = { 'default' : '\033[0m',
             'bold' : '\033[1m',
             'underline' : '\033[4m',
             'blink' : '\033[5m',
             'reverse' : '\033[7m',
             'concealed' : '\033[8m'}

__colors = { 'black' : '\033[30m',
             'red' : '\033[31m',
             'green' : '\033[32m',
             'yellow' : '\033[33m',
             'blue' : '\033[34m',
             'magenta' : '\033[35m',
             'cyan' : '\033[36m',
             'white' : '\033[37m',
             'on_black' : '\033[40m',
             'on_red' : '\033[41m',
             'on_green' : '\033[42m',
             'on_yellow' : '\033[43m',
             'on_blue' : '\033[44m',
             'on_magenta' : '\033[45m',
             'on_cyan' : '\033[46m',
             'on_white' : '\033[47m'}

def setColor(color):
sys.stdout.write ( __colors[color] )

def setStyle(style, color=None):
    if style == '\default':
        sys.stdout.write ( __styles[style] )
    else:
        if color == '\default' :
            sys.stdout.write ( __colors[color] )
            sys.stdout.write ( __styles[style] )
#include <stdlib.h>
#include <stdio.h>
#include <NIDAQmx.h>
#include <math.h>
#include <string.h>

#define DAQmxErrChk(functionCall) if( DAQmxFailed(error=(functionCall)) ) \
    goto Error;

float64 * acquireData(int samples_per_channel, float pulse_width, int *length);
int loadCounter(int start, int rf, double ipp, double pw);
uint8 * readDigital(uint8 *data);
int writeDigital(int port, char lines[], char values[]);

PyObject *Convert_Big_Array(PyObject *pylist, double array[], int start, int end) {
    int i = 0, j;
    int length = end - start;
    pylist = PyList_New( length );
    for (i=0, j=start; i < length; j++, i++) {
        PyList_SetItem(pylist, i, PyFloat_FromDouble(array[j]));
        //printf("%f \n",array[i]);
    }
    return pylist;
}

PyObject *py_acquireData(PyObject *self, PyObject *args) {
    int samples_per_channel = 1000;
    float pulsewidth = 0.02*3;
    float64 *data;
    int length;
    PyObject *pylist1, *pylist2, *pylist3;
    if (!PyArg_ParseTuple(args,"d", &samples_per_channel, &pulsewidth )) {
        printf("Restoring to default\n");
        if (!PyArg_ParseTuple(args,"d");
            return Py_BuildValue("d", -1);
    }
    //printf("d if",samples_per_channel,pulsewidth);
    data = acquireData(samples_per_channel, pulsewidth, &length);
    pylist1 = Convert_Big_Array( pylist1, data, 0, length/3 );
    pylist2 = Convert_Big_Array( pylist2, data, length / 3, length * 2/3 );
    pylist3 = Convert_Big_Array( pylist3, data, length*2/3, length );
    return Py_BuildValue( "{OOO}"
    , pylist1, pylist2, pylist3);
}

PyObject *py_loadTimers (PyObject *self, PyObject *args) {
    int trrf = 0; // only TR 1 is both TR and RF
    float period = 0;
    float pw = 0; // Duty cycle in percentage
    if (!PyArg_ParseTuple(args,"df", &trrf, &period, &pw)) {
        return Py_BuildValue("f", -1);
    }
    // Check if the samplingrate exceed 250KSamples/sec, the limit of the DAQ
    if (samplingrate > 250000) {
        Py_BuildValue("d", samplingrate * pulse_width);
    }
```c
int32_t error = 0;
TaskHandle_t taskHandle = 0;

DAQmxErrChk (DAQmxCreateTask("Task_DAQ", &taskHandle));

// DAQmx Configure Code
DAQmxErrChk (DAQmxCreateAIVoltageChan(taskHandle, "Dev1/ai0", "", DAQmx_Val_Diff, -5.0, 5.0, DAQmx_Val_Volts, NULL));
for (i = 0; i < No_of_Channels; i++) {
    DAQmxErrChk (DAQmxCreateAIVoltageChan(taskHandle, "Dev1/ai1", "", DAQmx_Val_Diff, -5.0, 5.2, DAQmx_Val_Volts, NULL));
    DAQmxErrChk (DAQmxCreateAIVoltageChan(taskHandle, "Dev1/ai2", "", DAQmx_Val_RSE, -0.1, 5.2, DAQmx_Val_Volts, NULL));
}    DAQmxErrChk (DAQmxCfgSampClkTiming(taskHandle, "", DAQmx_Val_Rising, DAQmx_Val_1000000, DAQmx_Val_GroupByChannel));

// RF Pulse Modulation is pin #11: because using pin 11
DAQmxErrChk (DAQmxCreateDevIOChan(taskHandle, "Dev1/line1:2", DAQmx_Val_Rising, DAQmx_Val_GroupByChannel));

DAQmxErrChk (DAQmxCreateTask("", &taskHandle));
DAQmxErrChk (DAQmxCreateDIChan(taskHandle, "Dev1/port1/line1:2", DAQmx_Val_Rising, DAQmx_Val_GroupByChannel));

Data = (double *) calloc(BuffSize, sizeof(double));
errBuff = (char *) calloc(BuffSize, sizeof(char));

if (No_of_Channels > 1) {
    DAQmxErrChk (DAQmxEventWait(taskHandle, "", DAQmx_Val_Rising, 10000));
    DAQmxErrChk (DAQmxWriteAnalogF64(taskHandle, BuffSize / No_of_Channels, 0, DAQmx_Val_GroupByChannel, data, &read, NULL));
    for (i = 0; i < BuffSize; i++) {
        *(data+i) = (rand() % 10);    }
}    *length = BuffSize;
return data;

uInt8_t *readDigital(uInt8_t *data){
    int32_t error = 0;
    TaskHandle_t taskHandle = 0;
    //uInt8_t data[100];
    char errBuff[2048]={'\0'};
    int32_t read, bytesPerSamp;
    DAQmxErrChk (DAQmxCreateTask("", &taskHandle));
    DAQmxErrChk (DAQmxCreateDIChan(taskHandle, "Dev1/port1/line1:2", DAQmx_Val_Rising, DAQmx_Val_GroupByChannel));
    DAQmxErrChk (DAQmxWriteDigitalLines(taskHandle, 100, 10000, DAQmx_Val_GroupByChannel, data, &read, &bytesPerSamp, NULL));
    DAQmxErrChk (DAQmxStartTask(taskHandle));
    char errBuff[2048]={'\0'};
    DAQmxErrChk (DAQmxGetDigitalState(taskHandle, &read, &bytesPerSamp, NULL));
    DAQmxErrChk (DAQmxGetDigitalState(taskHandle, &read, &bytesPerSamp, NULL));
    DAQmxErrChk (DAQmxStopTask(taskHandle));
    DAQmxErrChk (DAQmxClearTask(taskHandle));
    DAQmxErrChk (DAQmxTerminateTask(taskHandle));
```

```c
/**** DAQmx Configure Code
// DAQmx Start Code
/****
```

```c
// 8 seconds timeout
DAQmxErrChk (DAQmxReadAnalogF64(taskHandle, BuffSize / No_of_Channels, 5.0, 0, data, BuffSize, &read, NULL));
```

```c
// Error:
if (DAQmxFailed(error)) {
    DAQmxErrChk (DAQmxGetExtendedErrorInfo(errBuff, 2048));
    DAQmxErrChk (DAQmxStopTask(taskHandle));
    DAQmxErrChk (DAQmxClearTask(taskHandle));
    DAQmxErrChk (DAQmxTerminateTask(taskHandle));
    printf("DAQmx Error: %s
", errBuff);
    BuffSize = 200;
    data = (double *) calloc(BuffSize, sizeof(double));
    for (i = 0; i < BuffSize; i++) {
        *(data+i) = (rand() % 10);    }
    *length = BuffSize;
    return data;
}
```

```c
uInt8_t *readDigital(uInt8_t *data){
    int32_t error = 0;
    TaskHandle_t taskHandle = 0;
    //uInt8_t data[100];
    char errBuff[2048]={'\0'};
    int32_t read, bytesPerSamp;
    DAQmxErrChk (DAQmxCreateTask("", &taskHandle));
    DAQmxErrChk (DAQmxCreateDIChan(taskHandle, "Dev1/port1/line1:2", DAQmx_Val_Rising, DAQmx_Val_GroupByChannel));
    DAQmxErrChk (DAQmxWriteDigitalLines(taskHandle, 100, 10.0, DAQmx_Val_GroupByChannel, data, 100, &read, &bytesPerSamp, NULL));
    DAQmxErrChk (DAQmxStartTask(taskHandle));
    char errBuff[2048]={'\0'};
    DAQmxErrChk (DAQmxGetDigitalState(taskHandle, &read, &bytesPerSamp, NULL));
    DAQmxErrChk (DAQmxStopTask(taskHandle));
    DAQmxErrChk (DAQmxClearTask(taskHandle));
    DAQmxErrChk (DAQmxTerminateTask(taskHandle));
    printf("DAQmx Error: %s
", errBuff);
```
# Digital Functions Remain in Force Even When the Task Is Cleared.

This is not so for timer tasks. See also loadCounter.

```c
if ( start == 0 ) {
    return 0;
}
if ( rf == 1 ) {
    DAQmxErrChk (DAQmxCreateTask("Counter_1",&taskHandle1));
    DAQmxErrChk (DAQmxCreateCOPulseChanFreq(taskHandle1,"Dev1/ctr0","",DAQmx_Val_Hz,DAQmx_Val_Low, 0.00, freq, dutycycle2));
    // Trigger Counter 0 on rising edge of 1
    DAQmxErrChk (DAQmxCreateCOPulseChanFreq(taskHandle2,"Dev1/ctr0","",DAQmx_Val_Hz,DAQmx_Val_Low, delay, freq, dutycycle1));
    // Most of the routines use local's for TaskHandle, but the timers * require static handles that are not destroyed upon function returns */
    static TaskHandle taskHandle1=0, taskHandle2 = 0;
    char errBuff[2048]="\0";
    /*********************************************/
    // DAQmx Configure Code
    /*********************************************/
    if( taskHandle1 ≠ 0 ) {
        DAQmxStopTask(taskHandle1);
        DAQmxClearTask(taskHandle1);
        taskHandle1 = 0;
    }
    if( taskHandle2 ≠ 0 ) {
        DAQmxStopTask(taskHandle2);
        DAQmxClearTask(taskHandle2);
        taskHandle2 = 0;
    }
    if ( rf == 0 )
    DAQmxGetExtendedErrorInfo(errBuff,2048);
    DAQmxStartTask(taskHandle1);
    DAQmxStartTask(taskHandle2);
    if( DAQmxFailed(error) )
        return 1;
    int loadCounter(int start, int rf, double ipp, double pw) {
        double delay = 1.0 / 1000000; // 1us delay
        double dutycycle1 = pw / ipp;
        double dutycycle2 = (pw + 2 * delay) / ipp; // duty cycle for TR
        int ipp1 = ipp;
```
if (rf == 1) {
    DAQmxErrChk (DAQmxCfgDigEdgeStartTrig(taskHandle1, "/Dev1/PFI9",
                                            DAQmx_Val_Rising));
    DAQmxErrChk (DAQmxCfgImplicitTiming(taskHandle1,DAQmx_Val_ContSamps,10));
    DAQmxErrChk (DAQmxCfgImplicitTiming(taskHandle2,DAQmx_Val_ContSamps,10));
    DAQmxErrChk (DAQmxRegisterDoneEvent(taskHandle,0,DoneCallback,NULL));
    // DAQmxErrChk (DAQmxRegisterDoneEvent(taskHandle,0,DoneCallback,NULL));
}  

/********************************************
// DAQmx Start Code
/********************************************
// Start counter 1 first (because this is what is waiting for trigger)
DAQmxErrChk (DAQmxStartTask(taskHandle2));

if (rf == 1) {
    // Start counter 0 now (this triggers the counter 1)
    DAQmxErrChk (DAQmxStartTask(taskHandle1));
}

Error:
if( DAQmxFailed(error) ) {
    DAQmxGetExtendedErrorInfo(errBuff,2048);
    DAQmxStopTask(taskHandle1);
    DAQmxClearTask(taskHandle1);
    DAQmxStopTask(taskHandle2);
    DAQmxClearTask(taskHandle2);
    printf("DAQmx Error: %s",errBuff);
    return 0;
}

// ============================================================

Appendix E

Distribution Board
Component Data Sheets
# Table of Contents

1. 2N7002: N-Channel Enhancement Mode Field Effect Transistor ........................................F-3
2. 74AC74: Dual D-Type Positive Edge Triggered Flip-Flop ........................................F-11
3. SN74AHCT14: Hex Schmitt-Trigger Inverter .................................................................F-22
4. SN74ALS08: Quadruple 2-Input Positive-AND Gates ..................................................F-38
5. SN74HCT74: Dual D-Type Positive-Edge-Triggered Flip Flops ...................................F-53
6. SN74AHCT541: Octal Buffers/Drivers with 3-State Outputs ..........................................F-67
7. SN74HC21: Dual 4-Input Positive-AND Gates ...............................................................F-82
8. 564-0700-111F: 3mm LED CBI Tri-Level Circuit Board Indicator ...............................F-95
9. HEF4521B: 24-Stage Frequency Divider and Oscillator ............................................F-97
10. 179-009-513R571: Center Dual Port D-SUB Female/Female ....................................F-109
11. LTC1441/SO: Ultra Low Power Dual Comparator with Reference .............................F-111
12. 5227161-2: BNC Jack, Right Angle .............................................................................F-128
13. PJ-102BH: DC Power Jack .........................................................................................F-130
1. 2N7002: N-Channel Enhancement Mode Field Effect Transistor

Fairchild Semiconductor
N-Channel Enhancement Mode
Field Effect Transistor
2N7002
## Electrical Characteristics

$T_a = 25^\circ C$ unless otherwise noted

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Conditions</th>
<th>Type</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>BV&lt;br/&gt;DSS</td>
<td>Drain-Source Breakdown Voltage</td>
<td>$V_{GS} = 0 , V$, $I_D = 10 \mu A$</td>
<td>All</td>
<td>60</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 48 , V$, $V_{GS} = 0 , V$</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>$\mu A$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 60 , V$, $V_{GS} = 0 , V$</td>
<td>$T_a=125^\circ C$</td>
<td>1</td>
<td></td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>I&lt;br/&gt;DSS</td>
<td>Zero Gate Voltage Drain Current</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 15 , V$, $V_{DS} = 0 , V$</td>
<td>2N7000</td>
<td>10</td>
<td></td>
<td></td>
<td>nA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 20 , V$, $V_{DS} = 0 , V$</td>
<td>2N7002 NDS7002A</td>
<td>100</td>
<td></td>
<td></td>
<td>nA</td>
</tr>
<tr>
<td>I&lt;br/&gt;GSSF</td>
<td>Gate - Body Leakage, Forward</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = -15 , V$, $V_{DS} = 0 , V$</td>
<td>2N7000</td>
<td>-10</td>
<td></td>
<td></td>
<td>nA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = -20 , V$, $V_{DS} = 0 , V$</td>
<td>2N7002 NDS7002A</td>
<td>-100</td>
<td></td>
<td></td>
<td>nA</td>
</tr>
<tr>
<td>I&lt;br/&gt;GSSR</td>
<td>Gate - Body Leakage, Reverse</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 10 , V$, $I_D = 500 , mA$</td>
<td>2N7000</td>
<td>0.6</td>
<td>2.5</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 4.5 , V$, $I_D = 75 , mA$</td>
<td>2N7002</td>
<td>0.14</td>
<td>0.4</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 10 , V$, $I_D = 500 , mA$</td>
<td>2N7002</td>
<td>0.6</td>
<td>3.75</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>R&lt;br/&gt;DS(ON)</td>
<td>Static Drain-Source On-Resistance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 10 , V$, $I_D = 500 , mA$</td>
<td>2N7000</td>
<td>1.2</td>
<td>5</td>
<td></td>
<td>$\Omega$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 4.5 , V$, $I_D = 75 , mA$</td>
<td>2N7002</td>
<td>1.7</td>
<td>3.5</td>
<td></td>
<td>$\Omega$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 10 , V$, $I_D = 500 , mA$</td>
<td>NDS7002A</td>
<td>1.2</td>
<td>2</td>
<td></td>
<td>$\Omega$</td>
</tr>
<tr>
<td>V&lt;br/&gt;DS(ON)</td>
<td>Drain-Source On-Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 10 , V$, $I_D = 500 , mA$</td>
<td>2N7000</td>
<td>0.6</td>
<td>2.5</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 4.5 , V$, $I_D = 75 , mA$</td>
<td>2N7002</td>
<td>0.14</td>
<td>0.4</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 10 , V$, $I_D = 500 , mA$</td>
<td>2N7002</td>
<td>0.6</td>
<td>3.75</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 5.0 , V$, $I_D = 50 , mA$</td>
<td>NDS7002A</td>
<td>0.09</td>
<td>1</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 10 , V$, $I_D = 500 , mA$</td>
<td>NDS7002A</td>
<td>0.6</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{GS} = 5.0 , V$, $I_D = 50 , mA$</td>
<td>NDS7002A</td>
<td>0.09</td>
<td>0.15</td>
<td></td>
<td>V</td>
</tr>
</tbody>
</table>
### Electrical Characteristics \( T_a = 25^\circ \text{C} \) unless otherwise noted

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Conditions</th>
<th>Type</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_{\text{ON}} )</td>
<td>On-State Drain Current</td>
<td>( V_{GS} = 4.5 \ \text{V}, \ V_{DS} = 10 \ \text{V} )</td>
<td>2N7000</td>
<td>75</td>
<td>600</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( V_{GS} = 10 \ \text{V}, \ V_{DS} \geq 2 \ V_{\text{DS(ol)}} )</td>
<td>2N7002</td>
<td>500</td>
<td>2700</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( V_{GS} = 10 \ \text{V}, \ V_{DS} \geq 2 \ V_{\text{DS(ol)}} )</td>
<td>NDS7002A</td>
<td>500</td>
<td>2700</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>( g_f )</td>
<td>Forward Transconductance</td>
<td>( V_{DS} = 10 \ \text{V}, \ I_D = 200 \ \text{mA} )</td>
<td>2N7000</td>
<td>100</td>
<td>320</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( V_{DS} \geq 2 \ V_{\text{DS(ol)}}, \ I_D = 200 \ \text{mA} )</td>
<td>2N7002</td>
<td>80</td>
<td>320</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( V_{DS} \geq 2 \ V_{\text{DS(ol)}}, \ I_D = 200 \ \text{mA} )</td>
<td>NDS7002A</td>
<td>80</td>
<td>320</td>
<td>mA</td>
<td></td>
</tr>
</tbody>
</table>

### Dynamic Characteristics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Conditions</th>
<th>Type</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{\text{iss}} )</td>
<td>Input Capacitance</td>
<td>( V_{DS} = 25 \ \text{V}, \ V_{GS} = 0 \ \text{V}, \ f = 1.0 \ \text{MHz} )</td>
<td>All</td>
<td>20</td>
<td>50</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>( C_{\text{oss}} )</td>
<td>Output Capacitance</td>
<td>( f = 1.0 \ \text{MHz} )</td>
<td>All</td>
<td>11</td>
<td>25</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>( C_{\text{rss}} )</td>
<td>Reverse Transfer Capacitance</td>
<td>All</td>
<td>4</td>
<td>5</td>
<td>pF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( t_{\text{on}} )</td>
<td>Turn-On Time</td>
<td>( V_{DD} = 15 \ \text{V}, \ R_L = 25 \ \Omega, \ I_D = 500 \ \text{mA}, \ V_{GS} = 10 \ \text{V}, \ R_{\text{GEN}} = 25 )</td>
<td>2N7000</td>
<td>10</td>
<td>ns</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( V_{DD} = 30 \ \text{V}, \ R_L = 150 \ \Omega, \ I_D = 200 \ \text{mA}, \ V_{GS} = 10 \ \text{V}, \ R_{\text{GEN}} = 25 )</td>
<td>2N700</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( V_{DD} = 30 \ \text{V}, \ R_L = 150 \ \Omega, \ I_D = 200 \ \text{mA}, \ V_{GS} = 10 \ \text{V}, \ R_{\text{GEN}} = 25 )</td>
<td>NDS7002A</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( t_{\text{off}} )</td>
<td>Turn-Off Time</td>
<td>( V_{DD} = 15 \ \text{V}, \ R_L = 25 \ \Omega, \ I_D = 500 \ \text{mA}, \ V_{GS} = 10 \ \text{V}, \ R_{\text{GEN}} = 25 )</td>
<td>2N7000</td>
<td>10</td>
<td>ns</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( V_{DD} = 30 \ \text{V}, \ R_L = 150 \ \Omega, \ I_D = 200 \ \text{mA}, \ V_{GS} = 10 \ \text{V}, \ R_{\text{GEN}} = 25 )</td>
<td>2N700</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( V_{DD} = 30 \ \text{V}, \ R_L = 150 \ \Omega, \ I_D = 200 \ \text{mA}, \ V_{GS} = 10 \ \text{V}, \ R_{\text{GEN}} = 25 )</td>
<td>NDS7002A</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Drain-Source Diode Characteristics and Maximum Ratings

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Conditions</th>
<th>Type</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_D )</td>
<td>Maximum Continuous Drain-Source Diode Forward Current</td>
<td>( V_{GS} = 0 \ \text{V}, \ I_D = 115 \ \text{mA} )</td>
<td>2N7002</td>
<td>115</td>
<td>mA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \text{Note 1} )</td>
<td>NDS7002A</td>
<td>280</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( I_{SM} )</td>
<td>Maximum Pulsed Drain-Source Diode Forward Current</td>
<td>( V_{GS} = 0 \ \text{V} )</td>
<td>2N7002</td>
<td>0.8</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \text{Note 1} )</td>
<td>NDS7002A</td>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{SD} )</td>
<td>Drain-Source Diode Forward Voltage</td>
<td>( V_{GS} = 0 \ \text{V} )</td>
<td>2N7002</td>
<td>0.88</td>
<td>1.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( I_D = 400 \ \text{mA} )</td>
<td>NDS7002A</td>
<td>0.88</td>
<td>1.2</td>
<td>V</td>
<td></td>
</tr>
</tbody>
</table>

Note: 1. Pulse Test: Pulse Width \( \leq 300 \mu \text{s} \), Duty Cycle \( \leq 2.0\% \).
Typical Electrical Characteristics

Figure 1. On-Region Characteristics

Figure 2. On-Resistance Variation with Gate Voltage and Drain Current

Figure 3. On-Resistance Variation with Temperature

Figure 4. On-Resistance Variation with Drain Current and Temperature

Figure 5. Transfer Characteristics

Figure 6. Gate Threshold Variation with Temperature
Figure 7. Breakdown Voltage Variation with Temperature

Figure 8. Body Diode Forward Voltage Variation with Temperature

Figure 9. Capacitance Characteristics

Figure 10. Gate Charge Characteristics

Figure 11.

Figure 12. Switching Waveforms
Figure 13. 2N7000 Maximum Safe Operating Area

Figure 14. 2N7002 Maximum Safe Operating Area

Figure 15. NDS7000A Maximum Safe Operating Area

Figure 16. TO-92, 2N7000 Transient Thermal Response Curve

Figure 17. SOT-23, 2N7002 / NDS7002A Transient Thermal Response Curve
### TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

<table>
<thead>
<tr>
<th>Trademark</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACEx™</td>
<td></td>
</tr>
<tr>
<td>Bottomless™</td>
<td></td>
</tr>
<tr>
<td>CoolFET™</td>
<td></td>
</tr>
<tr>
<td>CROSSVOLT™</td>
<td></td>
</tr>
<tr>
<td>DOME™</td>
<td></td>
</tr>
<tr>
<td>E^2CMOS™</td>
<td></td>
</tr>
<tr>
<td>EnSigna™</td>
<td></td>
</tr>
<tr>
<td>FACT™</td>
<td></td>
</tr>
<tr>
<td>FACT Quiet Series™</td>
<td></td>
</tr>
<tr>
<td>FAST®</td>
<td></td>
</tr>
<tr>
<td>GlobalOptisolator™</td>
<td></td>
</tr>
<tr>
<td>GTO™</td>
<td></td>
</tr>
<tr>
<td>HiSeC™</td>
<td></td>
</tr>
<tr>
<td>ISOPLANAR™</td>
<td></td>
</tr>
<tr>
<td>MICROWIRE™</td>
<td></td>
</tr>
<tr>
<td>OPTOLOGIC™</td>
<td></td>
</tr>
<tr>
<td>OPTOPLANAR™</td>
<td></td>
</tr>
<tr>
<td>PACMAN™</td>
<td></td>
</tr>
<tr>
<td>POP™</td>
<td></td>
</tr>
<tr>
<td>PowerTrench®</td>
<td></td>
</tr>
<tr>
<td>QFET™</td>
<td></td>
</tr>
<tr>
<td>QS™</td>
<td></td>
</tr>
<tr>
<td>QT Optoelectronics™</td>
<td></td>
</tr>
<tr>
<td>Quiet Series™</td>
<td></td>
</tr>
<tr>
<td>QS™</td>
<td></td>
</tr>
<tr>
<td>SILENT SWITCHER®</td>
<td></td>
</tr>
<tr>
<td>SMART START™</td>
<td></td>
</tr>
<tr>
<td>SuperSOT™-3</td>
<td></td>
</tr>
<tr>
<td>SuperSOT™-6</td>
<td></td>
</tr>
<tr>
<td>SuperSOT™-8</td>
<td></td>
</tr>
<tr>
<td>SyncFET™</td>
<td></td>
</tr>
<tr>
<td>TinyLogic™</td>
<td></td>
</tr>
<tr>
<td>UHC™</td>
<td></td>
</tr>
<tr>
<td>VCX™</td>
<td></td>
</tr>
</tbody>
</table>

### DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

### LIFE SUPPORT POLICY

FAIRCHILD’S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

### PRODUCT STATUS DEFINITIONS

**Definition of Terms**

<table>
<thead>
<tr>
<th>Datasheet Identification</th>
<th>Product Status</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advance Information</td>
<td>Formative or In Design</td>
<td>This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.</td>
</tr>
<tr>
<td>Preliminary</td>
<td>First Production</td>
<td>This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.</td>
</tr>
<tr>
<td>No Identification Needed</td>
<td>Full Production</td>
<td>This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.</td>
</tr>
<tr>
<td>Obsolete</td>
<td>Not In Production</td>
<td>This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.</td>
</tr>
</tbody>
</table>
2N7000 / 2N7002 / NDS7002A
N-Channel Enhancement Mode Field Effect Transistor

General Description

These N-Channel enhancement mode field effect transistors are produced using Fairchild's proprietary, high cell density, DMOS technology. These products have been designed to minimize on-state resistance while provide rugged, reliable, and fast switching performance. They can be used in most applications requiring up to 400mA DC and can deliver pulsed currents up to 2A. These products are particularly suited for low voltage, low current applications such as small servo motor control, power MOSFET gate drivers, and other switching applications.

Features

- High density cell design for low $R_{DS(ON)}$.
- Voltage controlled small signal switch.
- Rugged and reliable.
- High saturation current capability.

Absolute Maximum Ratings

$T_a = 25^\circ C$ unless otherwise noted

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>2N7000</th>
<th>2N7002</th>
<th>NDS7002A</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DSS}$</td>
<td>Drain-Source Voltage</td>
<td>60</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{DGR}$</td>
<td>Drain-Gate Voltage ($R_{GS} \leq 1 , M\Omega$)</td>
<td>60</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{GSS}$</td>
<td>Gate-Source Voltage - Continuous</td>
<td>±20</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>- Non Repetitive ($t_p &lt; 50\mu s$)</td>
<td>±40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_D$</td>
<td>Maximum Drain Current - Continuous</td>
<td>200</td>
<td>115</td>
<td>280</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td>- Pulsed</td>
<td>500</td>
<td>800</td>
<td>1500</td>
<td></td>
</tr>
<tr>
<td>$P_D$</td>
<td>Maximum Power Dissipation</td>
<td>400</td>
<td>200</td>
<td>300</td>
<td>mW</td>
</tr>
<tr>
<td></td>
<td>Derated above 25°C</td>
<td>3.2</td>
<td>1.6</td>
<td>2.4</td>
<td>mW/°C</td>
</tr>
<tr>
<td>$T_{J,STG}$</td>
<td>Operating and Storage Temperature Range</td>
<td>-55 to 150</td>
<td>-65 to 150</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>$T_L$</td>
<td>Maximum Lead Temperature for Soldering Purposes, 1/16&quot; from Case for 10 Seconds</td>
<td>300</td>
<td></td>
<td></td>
<td>°C</td>
</tr>
</tbody>
</table>

THERMAL CHARACTERISTICS

| $R_{JUA}$ | Thermal Resistance, Junction-to-Ambient | 312.5 | 625   | 417      | °C/W  |

© 1997 Fairchild Semiconductor Corporation

2N7000.SAM Rev. A1
2. 74AC74: Dual D-Type Positive Edge Triggered Flip-Flop

Fairchild Semiconductor
Dual D-Type Positive Edge Triggered Flip-Flop
74AC74
74AC74 • 74ACT74
Dual D-Type Positive Edge-Triggered Flip-Flop

General Description
The AC/ACT74 is a dual D-type flip-flop with Asynchronous Clear and Set inputs and complementary (Q, $\overline{Q}$) outputs. Information at the input is transferred to the outputs on the positive edge of the clock pulse. Clock triggering occurs at a voltage level of the clock pulse and is not directly related to the transition time of the positive-going pulse. After the Clock Pulse input threshold voltage has been passed, the Data input is locked out and information present will not be transferred to the outputs until the next rising edge of the Clock Pulse input.

Asynchronous Inputs:
- LOW input to $\overline{S}$ (Set) sets $Q$ to HIGH level
- LOW input to $\overline{C}$ (Clear) sets $Q$ to LOW level
- Clear and Set are independent of clock
- Simultaneous LOW on $\overline{C}$ and $\overline{S}$ makes both $Q$ and $\overline{Q}$ HIGH

Features
- $I_{CC}$ reduced by 50%
- Output source/sink 24 mA
- ACT74 has TTL-compatible inputs

Ordering Code:

<table>
<thead>
<tr>
<th>Order Number</th>
<th>Package Number</th>
<th>Package Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>74AC74SC</td>
<td>M14A</td>
<td>14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150” Narrow</td>
</tr>
<tr>
<td>74AC74SC_NL</td>
<td>M14A</td>
<td>Pb-Free 14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150” Narrow</td>
</tr>
<tr>
<td>74AC74SJ</td>
<td>M14D</td>
<td>Pb-Free 14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide</td>
</tr>
<tr>
<td>74AC74MTC</td>
<td>MTC14</td>
<td>14-Lead Thin Shrink Small Outline Package (TSSOP), JEDEC MO-153, 4.4mm Wide</td>
</tr>
<tr>
<td>74AC74MTCX_NL</td>
<td>MTC14</td>
<td>Pb-Free 14-Lead Thin Shrink Small Outline Package (TSSOP), JEDEC MO-153, 4.4mm Wide</td>
</tr>
<tr>
<td>74AC74PC</td>
<td>N14A</td>
<td>14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300” Wide</td>
</tr>
<tr>
<td>74ACT74SC</td>
<td>M14A</td>
<td>14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150” Narrow</td>
</tr>
<tr>
<td>74ACT74SC_NL</td>
<td>M14A</td>
<td>Pb-Free 14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150” Narrow</td>
</tr>
<tr>
<td>74ACT74SJ</td>
<td>M14D</td>
<td>Pb-Free 14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide</td>
</tr>
<tr>
<td>74ACT74SJX_NL</td>
<td>M14D</td>
<td>Pb-Free 14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide</td>
</tr>
<tr>
<td>74ACT74MTC</td>
<td>MTC14</td>
<td>14-Lead Thin Shrink Small Outline Package (TSSOP), JEDEC MO-153, 4.4mm Wide</td>
</tr>
<tr>
<td>74ACT74PC</td>
<td>N14A</td>
<td>14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300” Wide</td>
</tr>
</tbody>
</table>

Device also available in Tape and Reel. Specify by appending suffix letter “X” to the ordering code.

Pb-Free package per JESD-020B.

Note 1: " _NL" indicates lead-free product (per JEDEC J-STD-020B).

Note 2: " _NL" indicates lead-free product (per JEDEC J-STD-020B). Device is available in Tape and Reel only.

FACT™ is a trademark of Fairchild Semiconductor Corporation.
**Connection Diagram**

**Pin Descriptions**

<table>
<thead>
<tr>
<th>Pin Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D&lt;sub&gt;1&lt;/sub&gt;, D&lt;sub&gt;2&lt;/sub&gt;</td>
<td>Data Inputs</td>
</tr>
<tr>
<td>CP&lt;sub&gt;1&lt;/sub&gt;, CP&lt;sub&gt;2&lt;/sub&gt;</td>
<td>Clock Pulse Inputs</td>
</tr>
<tr>
<td>C&lt;sub&gt;D1&lt;/sub&gt;, C&lt;sub&gt;D2&lt;/sub&gt;</td>
<td>Direct Clear Inputs</td>
</tr>
<tr>
<td>S&lt;sub&gt;D1&lt;/sub&gt;, S&lt;sub&gt;D2&lt;/sub&gt;</td>
<td>Direct Set Inputs</td>
</tr>
<tr>
<td>Q&lt;sub&gt;1&lt;/sub&gt;, Q&lt;sub&gt;1&lt;/sub&gt;, Q&lt;sub&gt;2&lt;/sub&gt;, Q&lt;sub&gt;2&lt;/sub&gt;</td>
<td>Outputs</td>
</tr>
</tbody>
</table>

**Logic Symbols**

**IEEE/IEC**

**Truth Table**

(Each Half)

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td>Output</td>
</tr>
<tr>
<td>S&lt;sub&gt;D&lt;/sub&gt;</td>
<td>C&lt;sub&gt;D&lt;/sub&gt;</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>

H – HIGH Voltage Level
L – LOW Voltage Level
X – Immaterial
\(\nearrow\) – LOW-to-HIGH Clock Transition
Q<sub>0</sub> (Q<sub>0</sub>) – Previous Q (Q) before LOW-to-HIGH Transition of Clock
Logic Diagram

Please note that this diagram is provided only for the understanding of logic operations and should not be used to estimate propagation delays.
### Absolute Maximum Ratings (Note 3)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>DC Input Diode Current (I_{IP})</th>
<th>DC Input Voltage (V_i)</th>
<th>DC Output Diode Current (I_{OK})</th>
<th>DC Output Voltage (V_o)</th>
<th>DC Output Source or Sink Current (I_{O})</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC Supply Voltage (V_{CC})</td>
<td>-0.5V to +7.0V</td>
<td>-0.5V to V_{CC} + 0.5V</td>
<td>-0.5V to V_{CC} + 0.5V</td>
<td>-0.5V to V_{CC} + 0.5V</td>
<td>±50 mA</td>
</tr>
<tr>
<td>V_i = -0.5V</td>
<td>-20 mA</td>
<td>+20 mA</td>
<td>-20 mA</td>
<td>+20 mA</td>
<td></td>
</tr>
<tr>
<td>V_i = V_{CC} + 0.5V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Recommended Operating Conditions

- **Supply Voltage (V_{CC})**: 2.0V to 6.0V
- **Input Voltage (V_i)**: 0V to V_{CC}
- **Output Voltage (V_o)**: 0V to V_{CC}
- **Operating Temperature (T_{A})**: -40°C to +85°C
- **Minimum Input Edge Rate (ΔV/Δt)**: 125 mV/ns

### DC Electrical Characteristics for AC

#### Symbols and Parameters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>V_{CC} (V)</th>
<th>V_{IH} (V)</th>
<th>I_{IH} (mA)</th>
<th>V_{IL} (V)</th>
<th>I_{IL} (mA)</th>
<th>I_{OHD} (mA)</th>
<th>V_{OL} (V)</th>
<th>I_{OL} (mA)</th>
<th>I_{IN} (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Minimum HIGH Level Input Voltage</td>
<td>3.0</td>
<td>1.5</td>
<td>2.1</td>
<td>2.1</td>
<td>V</td>
<td>V</td>
<td>0.002</td>
<td>0.001</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>Level Input Voltage</td>
<td>3.0</td>
<td>1.5</td>
<td>2.1</td>
<td>2.1</td>
<td>V</td>
<td>V</td>
<td>0.002</td>
<td>0.001</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>Maximum LOW Level Input Voltage</td>
<td>4.5</td>
<td>2.25</td>
<td>3.15</td>
<td>3.15</td>
<td>V</td>
<td>V</td>
<td>0.001</td>
<td>0.001</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>Level Output Voltage</td>
<td>4.5</td>
<td>2.25</td>
<td>3.15</td>
<td>3.15</td>
<td>V</td>
<td>V</td>
<td>0.001</td>
<td>0.001</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>Voltage</td>
<td>5.5</td>
<td>2.75</td>
<td>3.85</td>
<td>3.85</td>
<td>V</td>
<td>V</td>
<td>0.001</td>
<td>0.001</td>
<td>5.5</td>
</tr>
</tbody>
</table>

### Notes

- **Note 4**: All outputs loaded; thresholds on input associated with output under test.
- **Note 5**: Maximum test duration 2.0 ms, one output loaded at a time.
- **Note 6**: I_{IN} and I_{CC} @ 3.0V are guaranteed to be less than or equal to the respective limits @ 5.5V V_{CC}.
### DC Electrical Characteristics for ACT

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>( V_{CC} ) (V)</th>
<th>( T_A = -25^\circ C )</th>
<th>( T_A = -40^\circ C ) to ( 85^\circ C )</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{IH} )</td>
<td>Minimum HIGH Level Input Voltage</td>
<td>4.5</td>
<td>1.5</td>
<td>2.0</td>
<td>V</td>
<td>( V_{OUT} = 0.1V ) or ( V_{CC} = 0.1V )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5</td>
<td>1.5</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{IL} )</td>
<td>Maximum LOW Level Output Voltage</td>
<td>4.5</td>
<td>1.5</td>
<td>0.8</td>
<td>V</td>
<td>( V_{OUT} = 0.1V ) or ( V_{CC} = 0.1V )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5</td>
<td>1.5</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{OH} )</td>
<td>Minimum HIGH Level Output Voltage</td>
<td>4.5</td>
<td>4.49</td>
<td>4.4</td>
<td>V</td>
<td>( I_{OUT} = -50\mu A )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5</td>
<td>5.49</td>
<td>5.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5</td>
<td>3.86</td>
<td>3.76</td>
<td>V</td>
<td>( V_{IN} = V_{IL} ) or ( V_{IH} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5</td>
<td>4.86</td>
<td>4.76</td>
<td></td>
<td>( I_{OHL} = 24mA ) (Note 7)</td>
</tr>
<tr>
<td>( V_{OL} )</td>
<td>Maximum LOW Level Output Voltage</td>
<td>4.5</td>
<td>0.001</td>
<td>0.1</td>
<td>V</td>
<td>( I_{OUT} = 50\mu A )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5</td>
<td>0.001</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( I_{IN} )</td>
<td>Maximum Input Leakage Current</td>
<td>5.5</td>
<td>( \leq 0.1 )</td>
<td>( \leq 1.0 )</td>
<td>( \mu A )</td>
<td>( V_{IN} = V_{CC} ) or ( GND )</td>
</tr>
<tr>
<td>( I_{CCT} )</td>
<td>Maximum ( I_{CC} ) Input Leakage Current</td>
<td>5.5</td>
<td>0.6</td>
<td>1.5</td>
<td>mA</td>
<td>( V_{IN} = V_{CC} - 2.1V )</td>
</tr>
<tr>
<td>( I_{OLD} )</td>
<td>Minimum Dynamic Output Current (Note 8)</td>
<td>5.5</td>
<td>75</td>
<td>( mA )</td>
<td></td>
<td>( V_{OLD} = 1.65V Maximum )</td>
</tr>
<tr>
<td>( I_{OHD} )</td>
<td>Output Current (Note 8)</td>
<td>5.5</td>
<td>( -75 )</td>
<td>( mA )</td>
<td></td>
<td>( V_{OHD} = 3.85V Minimum )</td>
</tr>
<tr>
<td>( I_{CC} )</td>
<td>Maximum Quiescent Supply Current</td>
<td>5.5</td>
<td>2.0</td>
<td>20.0</td>
<td>( \mu A )</td>
<td>( V_{IN} = V_{CC} ) or ( GND )</td>
</tr>
</tbody>
</table>

**Note 7:** All outputs loaded; thresholds on input associated with output under test.

**Note 8:** Maximum test duration 2.0 ms, one output loaded at a time.

### AC Electrical Characteristics for AC

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>( V_{CC} ) (V)</th>
<th>( T_A = -25^\circ C )</th>
<th>( T_A = -40^\circ C ) to ( 85^\circ C )</th>
<th>Units</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( C_L = 50 \ pF )</td>
<td>( C_L = 50 \ pF )</td>
<td></td>
<td></td>
<td>MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Min</td>
<td>Typ</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>( f_{MAX} )</td>
<td>Maximum Clock Frequency</td>
<td>3.3</td>
<td>100</td>
<td>125</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0</td>
<td>140</td>
<td>160</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>( f_{PLH} )</td>
<td>Propagation Delay</td>
<td>3.3</td>
<td>3.5</td>
<td>8.0</td>
<td>12.0</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td>( C_{DH} ) or ( S_{DH} ) to ( Q_n ) or ( Q_n )</td>
<td>5.0</td>
<td>2.5</td>
<td>6.0</td>
<td>9.0</td>
<td>2.0</td>
</tr>
<tr>
<td>( f_{PHL} )</td>
<td>Propagation Delay</td>
<td>3.3</td>
<td>4.0</td>
<td>10.5</td>
<td>12.0</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td>( C_{DH} ) or ( S_{DH} ) to ( Q_n ) or ( Q_n )</td>
<td>5.0</td>
<td>3.0</td>
<td>8.0</td>
<td>9.5</td>
<td>2.5</td>
</tr>
<tr>
<td>( f_{LH} )</td>
<td>Propagation Delay</td>
<td>3.3</td>
<td>4.5</td>
<td>8.0</td>
<td>13.5</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td>( C_{PH} ) to ( Q_n ) or ( Q_n )</td>
<td>5.0</td>
<td>3.5</td>
<td>6.0</td>
<td>10.0</td>
<td>3.0</td>
</tr>
<tr>
<td>( f_{HL} )</td>
<td>Propagation Delay</td>
<td>3.3</td>
<td>3.5</td>
<td>8.0</td>
<td>14.0</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td>( C_{PH} ) to ( Q_n ) or ( Q_n )</td>
<td>5.0</td>
<td>2.5</td>
<td>6.0</td>
<td>10.0</td>
<td>2.5</td>
</tr>
</tbody>
</table>

**Note 9:** Voltage Range 3.3 is 3.3V ± 0.3V

Voltage Range 5.0 is 5.0V ± 0.5V
### AC Operating Requirements for AC

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>( V_{CC} ) (V)</th>
<th>( T_A = -25^\circ C )</th>
<th>( T_A = -40^\circ C ) to -85(^\circ C )</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_s )</td>
<td>Set-up Time, HIGH or LOW</td>
<td>\begin{align*} V_{CC} &amp; = 3.3 \quad C_L = 50 \text{ pF} \ t_s &amp; = 1.5 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_s &amp; = 2.0 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( t_h )</td>
<td>Hold Time, HIGH or LOW</td>
<td>\begin{align*} V_{CC} &amp; = 3.3 \quad C_L = 50 \text{ pF} \ t_h &amp; = 3.0 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_h &amp; = 4.5 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( t_W )</td>
<td>Pulse Width</td>
<td>\begin{align*} V_{CC} &amp; = 3.3 \quad C_L = 50 \text{ pF} \ t_W &amp; = 2.5 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_W &amp; = 4.5 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( t_{rec} )</td>
<td>Recovery Time</td>
<td>\begin{align*} V_{CC} &amp; = 3.3 \quad C_L = 50 \text{ pF} \ t_{rec} &amp; = 2.5 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_{rec} &amp; = 0 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

**Note 10:** Voltage Range 3.3 is 3.3V ± 0.3V

Voltage Range 5.0 is 5.0V ± 0.5V

### AC Electrical Characteristics for ACT

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>( V_{CC} ) (V)</th>
<th>( T_A = -25^\circ C )</th>
<th>( T_A = -40^\circ C ) to -85(^\circ C )</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{MAX} )</td>
<td>Maximum Clock Frequency</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ f_{MAX} &amp; = 145 \text{ MHz} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ f_{MAX} &amp; = 210 \text{ MHz} \end{align*}</td>
<td></td>
<td>MHz</td>
</tr>
<tr>
<td>( f_{PLH} )</td>
<td>Propagation Delay</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ f_{PLH} &amp; = 3.0 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ f_{PLH} &amp; = 5.5 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( f_{PHL} )</td>
<td>Propagation Delay</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ f_{PHL} &amp; = 3.0 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ f_{PHL} &amp; = 6.0 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( f_{PLH} )</td>
<td>Propagation Delay</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ f_{PLH} &amp; = 5.0 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ f_{PLH} &amp; = 7.5 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( f_{PHL} )</td>
<td>Propagation Delay</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ f_{PHL} &amp; = 5.0 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ f_{PHL} &amp; = 6.0 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

**Note 11:** Voltage Range 5.0 is 5.0V ± 0.5V

### AC Operating Requirements for ACT

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>( V_{CC} ) (V)</th>
<th>( T_A = -25^\circ C )</th>
<th>( T_A = -40^\circ C ) to -85(^\circ C )</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_s )</td>
<td>Set-up Time, HIGH or LOW</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_s &amp; = 1.0 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_s &amp; = 3.0 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( t_h )</td>
<td>Hold Time, HIGH or LOW</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_h &amp; = 0.5 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_h &amp; = 1.0 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( t_W )</td>
<td>Pulse Width</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_W &amp; = 3.0 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_W &amp; = 5.0 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( t_{rec} )</td>
<td>Recovery Time</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_{rec} &amp; = 2.5 \text{ ns} \end{align*}</td>
<td>\begin{align*} V_{CC} &amp; = 5.0 \quad C_L = 50 \text{ pF} \ t_{rec} &amp; = 0 \text{ ns} \end{align*}</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

**Note 12:** Voltage Range 5.0 is 5.0V ± 0.5V

### Capacitance

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{IN} )</td>
<td>Input Capacitance</td>
<td>pF</td>
<td>( V_{CC} = \text{OPEN} )</td>
</tr>
<tr>
<td>( C_{PD} )</td>
<td>Power Dissipation Capacitance</td>
<td>pF</td>
<td>( V_{CC} = 5.0V )</td>
</tr>
</tbody>
</table>

www.fairchildsemi.com
Physical Dimensions inches (millimeters) unless otherwise noted

14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150" Narrow
Package Number M14A
Physical Dimensions inches (millimeters) unless otherwise noted (Continued)

Pb-Free 14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
Package Number M14D

NOTES:
A. CONFORMS TO EIAJ ED-720 REGISTRATION, ESTABLISHED IN DECEMBER, 1988.
B. DIMENSIONS ARE IN MILLIMETERS.
C. DIMENSIONS ARE EXCLUSIVE OF BURRS, MOLD FLASH, AND TIE BAR EXTRUSIONS.

M14DRevB1

www.fairchildsemi.com
Physical Dimensions inches (millimeters) unless otherwise noted (Continued)

14-Lead Thin Shrink Small Outline Package (TSSOP), JEDEC MO-153, 4.4mm Wide
Package Number MTC14
Physical Dimensions inches (millimeters) unless otherwise noted (Continued)

Fairchild does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and Fairchild reserves the right at any time without notice to change said circuitry and specifications.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

www.fairchildsemi.com
3. SN74AHCT14: Hex Schmitt-Trigger Inverter

Texas Instruments
Hex Schmitt-Trigger Inverter
SN74AHCT14
**Inputs Are TTL-Voltage Compatible**

**Latch-Up Performance Exceeds 250 mA Per JESD 17**

**ESD Protection Exceeds JESD 22**
- 2000-V Human-Body Model (A114-A)
- 200-V Machine Model (A115-A)
- 1000-V Charged-Device Model (C101)

---

**description/ordering information**

The 'AHCT14 devices contain six independent inverters. These devices perform the Boolean function \( Y = \overline{A} \). Each circuit functions as an independent inverter, but because of the Schmitt action, the inverters have different input threshold levels for positive-going (\( V_{T+} \)) and for negative-going (\( V_{T-} \)) signals.

---

**ORDERING INFORMATION**

<table>
<thead>
<tr>
<th>TA</th>
<th>PACKAGE†</th>
<th>ORDERABLE PART NUMBER</th>
<th>TOP-SIDE MARKING</th>
</tr>
</thead>
<tbody>
<tr>
<td>–40°C to 85°C</td>
<td>QFN – RGY</td>
<td>Tape and reel</td>
<td>SN74AHCT14RGYR</td>
</tr>
<tr>
<td></td>
<td>PDIP – N</td>
<td>Tube</td>
<td>SN74AHCT14N</td>
</tr>
<tr>
<td></td>
<td>SOIC – D</td>
<td>Tube</td>
<td>SN74AHCT14D</td>
</tr>
<tr>
<td></td>
<td>SOP – NS</td>
<td>Tape and reel</td>
<td>SN74AHCT14NSR</td>
</tr>
<tr>
<td></td>
<td>SSOP – DB</td>
<td>Tape and reel</td>
<td>SN74AHCT14DBR</td>
</tr>
<tr>
<td></td>
<td>TSSOP – PW</td>
<td>Tape and reel</td>
<td>SN74AHCT14PW</td>
</tr>
<tr>
<td></td>
<td>TVSOP – DG</td>
<td>Tape and reel</td>
<td>SN74AHCT14DGVR</td>
</tr>
<tr>
<td>–55°C to 125°C</td>
<td>CDIP – J</td>
<td>Tube</td>
<td>SNJ54AHCT14J</td>
</tr>
<tr>
<td></td>
<td>CFP – W</td>
<td>Tube</td>
<td>SNJ54AHCT14W</td>
</tr>
<tr>
<td></td>
<td>LCCC – FK</td>
<td>Tube</td>
<td>SNJ54AHCT14FK</td>
</tr>
</tbody>
</table>

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

---

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.
sn54ahct14, sn74ahct14
hex schmitt-trigger inverters

function table
(each inverter)

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
<th>(each inverter)</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>l</td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>h</td>
<td></td>
</tr>
</tbody>
</table>

logic diagram, each inverter (positive logic)

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

<table>
<thead>
<tr>
<th>parameter</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>supply voltage range, ( V_{CC} )</td>
<td>(-0.5 ) ( V ) to ( 7 ) ( V )</td>
</tr>
<tr>
<td>input voltage range, ( V_I ) (see Note 1)</td>
<td>(-0.5 ) ( V ) to ( 7 ) ( V )</td>
</tr>
<tr>
<td>output voltage range, ( V_O ) (see Note 1)</td>
<td>(-0.5 ) ( V ) to ( V_{CC} ) (+ 0.5 ) ( V )</td>
</tr>
<tr>
<td>input clamp current, ( I_{IK} ) (( V_I &lt; 0 ))</td>
<td>(-20 ) ( mA )</td>
</tr>
<tr>
<td>output clamp current, ( I_{OK} ) (( V_O &lt; 0 ) or ( V_O &gt; V_{CC} ))</td>
<td>( \pm 20 ) ( mA )</td>
</tr>
<tr>
<td>continuous output current, ( I_O ) (( V_O = 0 ) to ( V_{CC} ))</td>
<td>( \pm 25 ) ( mA )</td>
</tr>
<tr>
<td>continuous current through ( V_{CC} ) or ( GND )</td>
<td>( \pm 50 ) ( mA )</td>
</tr>
<tr>
<td>package thermal impedance, ( \theta_{JA} ) (see Note 2): D package</td>
<td>( 86^\circ C/W )</td>
</tr>
<tr>
<td></td>
<td>( 96^\circ C/W )</td>
</tr>
<tr>
<td></td>
<td>( 80^\circ C/W )</td>
</tr>
<tr>
<td></td>
<td>( 76^\circ C/W )</td>
</tr>
<tr>
<td></td>
<td>( 113^\circ C/W )</td>
</tr>
<tr>
<td></td>
<td>( 47^\circ C/W )</td>
</tr>
<tr>
<td>storage temperature range, ( T_{stg} )</td>
<td>(-65^\circ C ) to ( 150^\circ C )</td>
</tr>
</tbody>
</table>

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

notes:
1. The input and output voltage ratings may be exceeded if the input and output current ratings are observed.
2. The package thermal impedance is calculated in accordance with JESD 51-7.
3. The package thermal impedance is calculated in accordance with JESD 51-5.

recommended operating conditions (see Note 4)

<table>
<thead>
<tr>
<th>parameter</th>
<th>SN54AHCT14</th>
<th>SN74AHCT14</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{CC} )</td>
<td>( 4.5 ) ( V )</td>
<td>( 5.5 ) ( V )</td>
</tr>
<tr>
<td>( V_I )</td>
<td>( 0 ) ( V )</td>
<td>( 5.5 ) ( V )</td>
</tr>
<tr>
<td>( V_O )</td>
<td>( 0 ) ( V_{CC} )</td>
<td>( 0 ) ( V_{CC} )</td>
</tr>
<tr>
<td>( I_{OH} )</td>
<td>( \pm 8 ) ( mA )</td>
<td>( \pm 8 ) ( mA )</td>
</tr>
<tr>
<td>( I_{OL} )</td>
<td>( 8 ) ( mA )</td>
<td>( 8 ) ( mA )</td>
</tr>
<tr>
<td>( T_A )</td>
<td>( -55 ) ( ^\circ C )</td>
<td>( 125 ) ( ^\circ C )</td>
</tr>
</tbody>
</table>

note 4: All unused inputs of the device must be held at \( V_{CC} \) or \( GND \) to ensure proper device operation. Refer to the TI application report, Implications of Slow or Floating CMOS Inputs, literature number SCBA004.
electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>VCC</th>
<th>SN54AHCT14</th>
<th>SN74AHCT14</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MIN</td>
<td>TYP</td>
<td>MAX</td>
<td>MIN</td>
</tr>
<tr>
<td>$V_{T+}$ Positive-going input</td>
<td>$T_A = 25,^\circ$C</td>
<td>4.5 V</td>
<td>0.9</td>
<td>1.9</td>
<td>0.9</td>
</tr>
<tr>
<td>threshold voltage</td>
<td></td>
<td>5.5 V</td>
<td>1</td>
<td>2.1</td>
<td>1</td>
</tr>
<tr>
<td>$V_{T-}$ Negative-going input</td>
<td></td>
<td>4.5 V</td>
<td>0.5</td>
<td>1.5</td>
<td>0.5</td>
</tr>
<tr>
<td>threshold voltage</td>
<td></td>
<td>5.5 V</td>
<td>0.6</td>
<td>1.7</td>
<td>0.6</td>
</tr>
<tr>
<td>$\Delta V_T$ Hysteresis</td>
<td>$T_A = 25,^\circ$C</td>
<td>4.5 V</td>
<td>0.4</td>
<td>1.4</td>
<td>0.4</td>
</tr>
<tr>
<td>(V $T_+$ – V $T_-$)</td>
<td></td>
<td>5.5 V</td>
<td>0.4</td>
<td>1.5</td>
<td>0.4</td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>IOH = –50 $\mu$A</td>
<td>4.5 V</td>
<td>4.4</td>
<td>4.5</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td>IOH = –8 mA</td>
<td>4.5 V</td>
<td>3.94</td>
<td>3.8</td>
<td></td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>IOL = 50 $\mu$A</td>
<td>4.5 V</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>IOL = 8 mA</td>
<td>4.5 V</td>
<td>0.36</td>
<td>0.44</td>
<td></td>
</tr>
<tr>
<td>$I_I$</td>
<td>$V_I = 5.5,V$</td>
<td>0 V to 5.5 V</td>
<td>±0.1</td>
<td>±1</td>
<td>±1</td>
</tr>
<tr>
<td></td>
<td>$V_I = V_{CC}$</td>
<td>5.5 V</td>
<td>2</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>$I_O = 0$</td>
<td>5.5 V</td>
<td>1.35</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>One input at 3.4 V, Other inputs at $V_{CC}$ or GND</td>
<td>5.5 V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_{I}$</td>
<td>$V_I = V_{CC}$</td>
<td>5 V</td>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$V_I = GND$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* On products compliant to MIL-PRF-38535, this parameter is not production tested at $V_{CC} = 0\,V$.
† This is the increase in supply current for each input at one of the specified TTL voltage levels, rather than 0 V or $V_{CC}$.

switching characteristics over recommended operating free-air temperature range $V_{CC} = 5\,V \pm 0.5\,V$ (unless otherwise noted) (see Figure 1)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>FROM (INPUT)</th>
<th>TO (OUTPUT)</th>
<th>LOAD CAPACITANCE</th>
<th>$T_A = 25,^\circ$C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>CL = 15 pF</td>
<td>SN54AHCT14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MIN</td>
</tr>
<tr>
<td>$I_{P_LH}$</td>
<td>A</td>
<td>Y</td>
<td></td>
<td>4**</td>
</tr>
<tr>
<td>$I_{P_HL}$</td>
<td>A</td>
<td>Y</td>
<td></td>
<td>4**</td>
</tr>
<tr>
<td>$I_{P_LH}$</td>
<td>A</td>
<td>Y</td>
<td>CL = 50 pF</td>
<td>5.5</td>
</tr>
<tr>
<td>$I_{P_HL}$</td>
<td>A</td>
<td>Y</td>
<td>CL = 50 pF</td>
<td>5.5</td>
</tr>
</tbody>
</table>

** On products compliant to MIL-PRF-38535, this parameter is not production tested.

noise characteristics, $V_{CC} = 5\,V$, $C_L = 50\,pF$, $T_A = 25\,^\circ$C (see Note 5)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SN74AHCT14</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UNIT</td>
</tr>
<tr>
<td>$V_{OL(P)}$</td>
<td>Quiet output, maximum dynamic $V_{OL}$</td>
</tr>
<tr>
<td>$V_{OL(V)}$</td>
<td>Quiet output, minimum dynamic $V_{OL}$</td>
</tr>
<tr>
<td>$V_{OH(V)}$</td>
<td>Quiet output, minimum dynamic $V_{OH}$</td>
</tr>
<tr>
<td>$V_{IH(D)}$</td>
<td>High-level dynamic input voltage</td>
</tr>
<tr>
<td>$V_{IL(D)}$</td>
<td>Low-level dynamic input voltage</td>
</tr>
</tbody>
</table>

NOTE 5: Characteristics are for surface-mount packages only.

operating characteristics, $V_{CC} = 5\,V$, $T_A = 25\,^\circ$C

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>TYP</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{pd}$</td>
<td>No load, $f = 1,MHz$</td>
<td>12</td>
<td>pF</td>
</tr>
</tbody>
</table>
PARAMETER MEASUREMENT INFORMATION

From Output Under Test
Test Point

LOAD CIRCUIT FOR TOTEM-POLE OUTPUTS

From Output Under Test

LOAD CIRCUIT FOR 3-STATE AND OPEN-DRAIN OUTPUTS

TEST S1

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpLH/tpHL</td>
<td>Open</td>
</tr>
<tr>
<td>tpPLZ/tpPL</td>
<td>VCC</td>
</tr>
<tr>
<td>tpPHZ/tpPZH</td>
<td>GND</td>
</tr>
<tr>
<td>Open Drain</td>
<td>VCC</td>
</tr>
</tbody>
</table>

VOLTAGE WAVEFORMS

VOLTAGE WAVEFORMS
PULSE DURATION

VOLTAGE WAVEFORMS
PROPAGATION DELAY TIMES
INVERTING AND NONINVERTING OUTPUTS

NOTES:
A. C_L includes probe and jig capacitance.
B. Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control.
   Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control.
C. All input pulses are supplied by generators having the following characteristics: PRR ≤ 1 MHz, Z_O = 50 Ω, t_r ≤ 3 ns, t_f ≤ 3 ns.
D. The outputs are measured one at a time with one input transition per measurement.
E. All parameters and waveforms are not applicable to all devices.

Figure 1. Load Circuit and Voltage Waveforms
NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package is hermetically sealed with a ceramic lid using glass frit.
D. Index point is provided on cap for terminal identification only on press ceramic glass frit seal only.
E. Falls within MIL STD 1835 GDIP1-T14, GDIP1-T16, GDIP1-T18 and GDIP1-T20.
NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a ceramic lid using glass frit.
D. Index point is provided on cap for terminal identification only.
E. Falls within MIL STD 1835 GDFP1-F14 and JEDEC MO-092AB
NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a metal lid.
D. The terminals are gold plated.
E. Falls within JEDEC MS-004
**NOTES:**

A. All linear dimensions are in inches (millimeters).

B. This drawing is subject to change without notice.

C. Falls within JEDEC MS-001, except 18 and 20 pin minimum body length (Dim A).

D. The 20 pin end lead shoulder width is a vendor option, either half or full width.
NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.15 per side.
D. Falls within JEDEC: 24/48 Pins – MO-153
14/16/20/56 Pins – MO-194
NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. QFN (Quad Flatpack No-Lead) package configuration.
   ▲ The package thermal performance may be enhanced by bonding the thermal die pad to an external thermal plane. This pad is electrically and thermally connected to the backside of the die and possibly selected ground leads.
E. Package complies to JEDEC MO-241 variation BA.
PLASTIC SMALL-OUTLINE PACKAGE

8 PINS SHOWN

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.006 (0.15).
D. Falls within JEDEC MS-012

DIM PINS ** 8 14 16

A MAX 0.197 (5.00) 0.344 (8.75) 0.394 (10.00)

A MIN 0.189 (4.80) 0.337 (8.55) 0.386 (9.80)
## Mechanical Data

### NS (R-PDSO-G**)

**Plastic Small-Outline Package**

14-PINS SHOWN

![Diagram of package dimensions]

<table>
<thead>
<tr>
<th>DIM</th>
<th>14</th>
<th>16</th>
<th>20</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>A MAX</td>
<td>10,50</td>
<td>10,50</td>
<td>12,90</td>
<td>15,30</td>
</tr>
<tr>
<td>A MIN</td>
<td>9,90</td>
<td>9,90</td>
<td>12,30</td>
<td>14,70</td>
</tr>
</tbody>
</table>

**NOTES:**

A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.15.

---

**Texas Instruments**

[www.ti.com](http://www.ti.com)
NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0.15.
D. Falls within JEDEC MO-150
PW (R-PDSO-G**)

PLASTIC SMALL-OUTLINE PACKAGE

14 PINS SHOWN

NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0.15.
D. Falls within JEDEC MO-153
IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifiers</td>
<td>Audio</td>
</tr>
<tr>
<td>Data Converters</td>
<td>Automotive</td>
</tr>
<tr>
<td>DSP</td>
<td>Broadband</td>
</tr>
<tr>
<td>Interface</td>
<td>Digital Control</td>
</tr>
<tr>
<td>Logic</td>
<td>Military</td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>Optical Networking</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td>Telephony</td>
</tr>
<tr>
<td></td>
<td>Video &amp; Imaging</td>
</tr>
<tr>
<td></td>
<td>Wireless</td>
</tr>
</tbody>
</table>

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2003, Texas Instruments Incorporated
4. SN74ALS08: Quadruple 2-Input Positive-AND Gates

Texas Instruments
Quadruple 2-Input Positive-AND Gates
SN74ALS08
description

These devices contain four independent 2-input positive-AND gates. They perform the Boolean functions \( Y = A \cdot B \) or \( Y = A + B \) in positive logic.

The SN54ALS08 and SN54AS08 are characterized for operation over the full military temperature range of \(-55^\circ C\) to \(125^\circ C\). The SN74ALS08 and SN74AS08 are characterized for operation from \(0^\circ C\) to \(70^\circ C\).

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>L</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>L</td>
</tr>
</tbody>
</table>

logic symbol†

logic diagram (positive logic)

† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers shown are for the D, J, and N packages.
absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SN54ALS08</th>
<th>SN74ALS08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage, ( V_{CC} )</td>
<td>7 V</td>
<td>7 V</td>
</tr>
<tr>
<td>Input voltage, ( V_I )</td>
<td>7 V</td>
<td>7 V</td>
</tr>
<tr>
<td>Operating free-air temperature, ( T_A ): SN54ALS08</td>
<td>–55°C to 125°C</td>
<td>0°C to 70°C</td>
</tr>
<tr>
<td>SN74ALS08</td>
<td>–55°C to 70°C</td>
<td></td>
</tr>
<tr>
<td>Storage temperature</td>
<td>–65°C to 150°C</td>
<td></td>
</tr>
</tbody>
</table>

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

recommended operating conditions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SN54ALS08</th>
<th>SN74ALS08</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{CC} ) Supply voltage</td>
<td>4.5 to 5.5 V</td>
<td>4.5 to 5.5 V</td>
</tr>
<tr>
<td>( V_{IH} ) High-level input voltage</td>
<td>2 V</td>
<td>2 V</td>
</tr>
<tr>
<td>( V_{IL} ) Low-level input voltage</td>
<td>0.8 V, 0.7 V</td>
<td>0.8 V, 0.7 V</td>
</tr>
<tr>
<td>( I_{OH} ) High-level output current</td>
<td>–0.4 mA</td>
<td>–0.4 mA</td>
</tr>
<tr>
<td>( I_{OL} ) Low-level output current</td>
<td>4 mA</td>
<td>8 mA</td>
</tr>
<tr>
<td>( T_A ) Operating free-air temperature</td>
<td>–55°C to 125°C</td>
<td>0°C to 70°C</td>
</tr>
</tbody>
</table>

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Test Conditions</th>
<th>SN54ALS08</th>
<th>SN74ALS08</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{IK} )</td>
<td>( V_{CC} = 4.5 ) V, ( I_I = -18 ) mA</td>
<td>–1.5 V</td>
<td>–1.5 V</td>
</tr>
<tr>
<td>( V_{OH} )</td>
<td>( V_{CC} = 4.5 ) V to 5.5 V, ( I_{OH} = -0.4 ) mA</td>
<td>( V_{CC} - 2 ) V</td>
<td>( V_{CC} - 2 ) V</td>
</tr>
<tr>
<td>( V_{OI} )</td>
<td>( V_{CC} = 4.5 ) V, ( I_{OL} = 4 ) mA</td>
<td>0.25 V</td>
<td>0.25 V</td>
</tr>
<tr>
<td></td>
<td>( I_{OL} = 8 ) mA</td>
<td>0.35 V</td>
<td>0.35 V</td>
</tr>
<tr>
<td>( I_{II} )</td>
<td>( V_{CC} = 5.5 ) V, ( V_I = 7 ) V</td>
<td>0.1 mA</td>
<td>0.1 mA</td>
</tr>
<tr>
<td>( I_{IH} )</td>
<td>( V_{CC} = 5.5 ) V, ( V_I = 2.7 ) V</td>
<td>20 ( \mu )A</td>
<td>20 ( \mu )A</td>
</tr>
<tr>
<td>( I_{IL} )</td>
<td>( V_{CC} = 5.5 ) V, ( V_I = 0.4 ) V</td>
<td>–0.1 mA</td>
<td>–0.1 mA</td>
</tr>
<tr>
<td>( I_{O#} )</td>
<td>( V_{CC} = 5.5 ) V, ( V_O = 2.25 ) V</td>
<td>–20 mA</td>
<td>–112 mA</td>
</tr>
<tr>
<td>( I_{O#} )</td>
<td>( V_{CC} = 5.5 ) V, ( V_I = 4.5 ) V</td>
<td>1.3 mA</td>
<td>1.3 mA</td>
</tr>
<tr>
<td>( I_{CCH} )</td>
<td>( V_{CC} = 5.5 ) V, ( V_I = 0 )</td>
<td>2.2 mA</td>
<td>2.2 mA</td>
</tr>
</tbody>
</table>

† All typical values are at \( V_{CC} = 5 \) V, \( T_A = 25 \) °C.
# The output conditions have been chosen to produce a current that closely approximates one half of the true short-circuit output current, \( I_{OS} \).
### Switching Characteristics

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>FROM (INPUT)</th>
<th>TO (OUTPUT)</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>IpLH</td>
<td>A or B</td>
<td>Y</td>
<td>ns</td>
</tr>
<tr>
<td>IpHL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vcc</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Vih</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Vil</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>IoH</td>
<td>–2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Iol</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Ta</td>
<td>–55</td>
</tr>
<tr>
<td></td>
<td>125</td>
</tr>
</tbody>
</table>

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

### Absolute Maximum Ratings

- Supply voltage, VCC: 7 V
- Input voltage, VI: 7 V
- Operating free-air temperature range, TA: SN54AS08: –55°C to 125°C, SN74AS08: 0°C to 70°C
- Storage temperature range: –65°C to 150°C

‡ Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

### Recommended Operating Conditions

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SN54AS08</th>
<th>SN74AS08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vcc</td>
<td>MIN 575</td>
<td>MIN 55</td>
</tr>
<tr>
<td></td>
<td>NOM 5</td>
<td>NOM 5</td>
</tr>
<tr>
<td></td>
<td>MAX 5.5</td>
<td>MAX 5.5</td>
</tr>
<tr>
<td>ViH</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Vil</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>IoH</td>
<td>–2</td>
<td>–2</td>
</tr>
<tr>
<td>Iol</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Ta</td>
<td>–55</td>
<td>125</td>
</tr>
</tbody>
</table>

### Electrical Characteristics

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>SN54AS08</th>
<th>SN74AS08</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MIN 575</td>
<td>MIN 55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOM 5</td>
<td>NOM 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAX 5.5</td>
<td>MAX 5.5</td>
</tr>
<tr>
<td>Vih</td>
<td>Vcc – 2</td>
<td>Vcc – 2</td>
<td></td>
</tr>
<tr>
<td>Vol</td>
<td>0.35</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>Ii</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>Ih</td>
<td>20</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Iil</td>
<td>–0.5</td>
<td>–0.5</td>
<td></td>
</tr>
<tr>
<td>Io</td>
<td>–30</td>
<td>–30</td>
<td></td>
</tr>
<tr>
<td>Icch</td>
<td>5.8</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td>Iccl</td>
<td>14.9</td>
<td>14.9</td>
<td></td>
</tr>
</tbody>
</table>

§ All typical values are at Vcc = 5 V, Ta = 25°C.

¶ The output conditions have been chosen to produce a current that closely approximates one half of the true short-circuit output current, IOs.
### Switching Characteristics (see Figure 1)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>From (Input)</th>
<th>To (Output)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_{PLH}</td>
<td>A or B</td>
<td>Y</td>
<td>ns</td>
</tr>
<tr>
<td>t_{PHL}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>From (Input)</th>
<th>To (Output)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_{PLH}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t_{PHL}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

**Conditions:**
- $V_{CC} = 4.5$ V to 5.5 V,
- $C_L = 50$ pF,
- $R_L = 500$ $\Omega$,
- $T_A = \text{MIN to MAX}^\dagger$

^\dagger For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
PARAMETER MEASUREMENT INFORMATION
SERIES 54ALS/74ALS AND 54AS/74AS DEVICES

VOLTAGE WAVEFORMS
SETUP AND HOLD TIMES

VOLTAGE WAVEFORMS
PULSE DURATIONS

VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES, 3-STATE OUTPUTS

VOLTAGE WAVEFORMS
PROPAGATION DELAY TIMES

NOTES:
A. \( C_L \) includes probe and jig capacitance.
B. Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control.
   Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control.
C. When measuring propagation delay items of 3-state outputs, switch S1 is open.
D. All input pulses have the following characteristics: \( PRR \leq 1 \text{ MHz}, t_r = t_f = 2 \text{ ns}, \text{ duty cycle} = 50\% \).
E. The outputs are measured one at a time with one transition per measurement.

Figure 1. Load Circuits and Voltage Waveforms
### PACKAGING INFORMATION

<table>
<thead>
<tr>
<th>Orderable Device</th>
<th>Status (1)</th>
<th>Package Type</th>
<th>Package Drawing</th>
<th>Pins</th>
<th>Package Qty</th>
<th>Eco Plan (2)</th>
<th>Lead/Ball Finish</th>
<th>MSL Peak Temp (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5962-86842012A</td>
<td>ACTIVE</td>
<td>LCCC</td>
<td>FK</td>
<td>20</td>
<td>1</td>
<td>TBD</td>
<td>POST-PLATE</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>5962-8684201CA</td>
<td>ACTIVE</td>
<td>CDIP</td>
<td>J</td>
<td>14</td>
<td>1</td>
<td>TBD</td>
<td>A42 SNPB</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>5962-8684201DA</td>
<td>ACTIVE</td>
<td>CFP</td>
<td>W</td>
<td>14</td>
<td>1</td>
<td>TBD</td>
<td>A42</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>JM38510/37401B2A</td>
<td>ACTIVE</td>
<td>LCCC</td>
<td>FK</td>
<td>20</td>
<td>1</td>
<td>TBD</td>
<td>POST-PLATE</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>JM38510/37401BCA</td>
<td>ACTIVE</td>
<td>CDIP</td>
<td>J</td>
<td>14</td>
<td>1</td>
<td>TBD</td>
<td>A42 SNPB</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SN54ALS08J</td>
<td>ACTIVE</td>
<td>CDIP</td>
<td>J</td>
<td>14</td>
<td>1</td>
<td>TBD</td>
<td>A42 SNPB</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SN54AS08J</td>
<td>ACTIVE</td>
<td>CDIP</td>
<td>J</td>
<td>14</td>
<td>1</td>
<td>TBD</td>
<td>A42 SNPB</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SN74ALS08D</td>
<td>ACTIVE</td>
<td>SOIC</td>
<td>D</td>
<td>14</td>
<td>50</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74ALS08DE4</td>
<td>ACTIVE</td>
<td>SOIC</td>
<td>D</td>
<td>14</td>
<td>50</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74ALS08DG4</td>
<td>ACTIVE</td>
<td>SOIC</td>
<td>D</td>
<td>14</td>
<td>50</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74ALS08DR</td>
<td>ACTIVE</td>
<td>SOIC</td>
<td>D</td>
<td>14</td>
<td>2500</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74ALS08DRE4</td>
<td>ACTIVE</td>
<td>SOIC</td>
<td>D</td>
<td>14</td>
<td>2500</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74ALS08DRG4</td>
<td>ACTIVE</td>
<td>SOIC</td>
<td>D</td>
<td>14</td>
<td>2500</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74ALS08N</td>
<td>ACTIVE</td>
<td>PDIP</td>
<td>N</td>
<td>14</td>
<td>25</td>
<td>Pb-Free (RoHS)</td>
<td>CU NIPDAU</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SN74ALS08N3</td>
<td>OBSOLETE</td>
<td>PDIP</td>
<td>N</td>
<td>14</td>
<td>TBD</td>
<td>Call TI</td>
<td>Call TI</td>
<td></td>
</tr>
<tr>
<td>SN74ALS08NE4</td>
<td>ACTIVE</td>
<td>PDIP</td>
<td>N</td>
<td>14</td>
<td>25</td>
<td>Pb-Free (RoHS)</td>
<td>CU NIPDAU</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SN74ALS08NSR</td>
<td>ACTIVE</td>
<td>SO</td>
<td>NS</td>
<td>14</td>
<td>2000</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74ALS08NSRE4</td>
<td>ACTIVE</td>
<td>SO</td>
<td>NS</td>
<td>14</td>
<td>2000</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74AS08D</td>
<td>ACTIVE</td>
<td>SOIC</td>
<td>D</td>
<td>14</td>
<td>50</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74AS08DE4</td>
<td>ACTIVE</td>
<td>SOIC</td>
<td>D</td>
<td>14</td>
<td>50</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74AS08DR</td>
<td>ACTIVE</td>
<td>SOIC</td>
<td>D</td>
<td>14</td>
<td>2500</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74AS08DRE4</td>
<td>ACTIVE</td>
<td>SOIC</td>
<td>D</td>
<td>14</td>
<td>2500</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74AS08N</td>
<td>ACTIVE</td>
<td>PDIP</td>
<td>N</td>
<td>14</td>
<td>25</td>
<td>Pb-Free (RoHS)</td>
<td>CU NIPDAU</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SN74AS08N3</td>
<td>OBSOLETE</td>
<td>PDIP</td>
<td>N</td>
<td>14</td>
<td>TBD</td>
<td>Call TI</td>
<td>Call TI</td>
<td></td>
</tr>
<tr>
<td>SN74AS08NE4</td>
<td>ACTIVE</td>
<td>PDIP</td>
<td>N</td>
<td>14</td>
<td>25</td>
<td>Pb-Free (RoHS)</td>
<td>CU NIPDAU</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SN74AS08NSR</td>
<td>ACTIVE</td>
<td>SO</td>
<td>NS</td>
<td>14</td>
<td>2000</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SN74AS08NSRE4</td>
<td>ACTIVE</td>
<td>SO</td>
<td>NS</td>
<td>14</td>
<td>2000</td>
<td>Green (RoHS &amp; no Sb/Br)</td>
<td>CU NIPDAU</td>
<td>Level-1-260C-UNLIM</td>
</tr>
<tr>
<td>SNJ4ALS08FK</td>
<td>ACTIVE</td>
<td>LCCC</td>
<td>FK</td>
<td>20</td>
<td>1</td>
<td>TBD</td>
<td>POST-PLATE</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SNJ4ALS08J</td>
<td>ACTIVE</td>
<td>CDIP</td>
<td>J</td>
<td>14</td>
<td>1</td>
<td>TBD</td>
<td>A42 SNPB</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>Orderable Device</td>
<td>Status (1)</td>
<td>Package Type</td>
<td>Package Drawing</td>
<td>Pins</td>
<td>Package Qty</td>
<td>Eco Plan (2)</td>
<td>Lead/Ball Finish</td>
<td>MSL Peak Temp (3)</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------</td>
<td>--------------</td>
<td>-----------------</td>
<td>------</td>
<td>-------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>SNJ54ALS08W</td>
<td>ACTIVE</td>
<td>CFP</td>
<td>W</td>
<td>14</td>
<td>1</td>
<td>TBD</td>
<td>A42</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SNJ54AS08FK</td>
<td>ACTIVE</td>
<td>LCCC</td>
<td>FK</td>
<td>20</td>
<td>1</td>
<td>TBD</td>
<td>POST-PLATE</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SNJ54AS08J</td>
<td>ACTIVE</td>
<td>CDIP</td>
<td>J</td>
<td>14</td>
<td>1</td>
<td>TBD</td>
<td>A42 SNPB</td>
<td>N / A for Pkg Type</td>
</tr>
<tr>
<td>SNJ54AS08W</td>
<td>ACTIVE</td>
<td>CFP</td>
<td>W</td>
<td>14</td>
<td>1</td>
<td>TBD</td>
<td>A42</td>
<td>N / A for Pkg Type</td>
</tr>
</tbody>
</table>

(1) The marketing status values are defined as follows:
   **ACTIVE:** Product device recommended for new designs.
   **LIFEBUY:** TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
   **NRND:** Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
   **PREVIEW:** Device has been announced but is not in production. Samples may or may not be available.
   **OBSOLETE:** TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
   **TBD:** The Pb-Free/Green conversion plan has not been defined.
   **Pb-Free (RoHS):** TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
   **Pb-Free (RoHS Exempt):** This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
   **Green (RoHS & no Sb/Br):** TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material).

(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

**Important Information and Disclaimer:** The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.
CERAMIC DUAL IN-LINE PACKAGE

<table>
<thead>
<tr>
<th>DIM</th>
<th>PINS **</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MAX</td>
<td>0.300 (7,62)</td>
<td>0.300 (7,62)</td>
<td>0.300 (7,62)</td>
<td>0.300 (7,62)</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>0.245 (6,22)</td>
<td>0.245 (6,22)</td>
<td>0.220 (5,59)</td>
<td>0.245 (6,22)</td>
</tr>
<tr>
<td>B</td>
<td>MAX</td>
<td>0.785 (19,94)</td>
<td>0.840 (21,34)</td>
<td>0.960 (24,38)</td>
<td>1.060 (26,92)</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>——</td>
<td>——</td>
<td>——</td>
<td>——</td>
</tr>
<tr>
<td>C</td>
<td>MAX</td>
<td>0.300 (7,62)</td>
<td>0.300 (7,62)</td>
<td>0.310 (7,87)</td>
<td>0.300 (7,62)</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>0.245 (6,22)</td>
<td>0.245 (6,22)</td>
<td>0.220 (5,59)</td>
<td>0.245 (6,22)</td>
</tr>
</tbody>
</table>

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package is hermetically sealed with a ceramic lid using glass frit.
D. Index point is provided on cap for terminal identification only on press ceramic glass frit seal only.
E. Falls within MIL STD 1835 GDIP1-T14, GDIP1-T16, GDIP1-T18 and GDIP1-T20.
W (R—GDFP—F14) CERAMIC DUAL FLATPACK

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a ceramic lid using glass frit.
D. Index point is provided on cap for terminal identification only.
E. Falls within MIL STD 1835 GDFP1—F14 and JEDEC MO–092AB

4040180–2/D 07/03
NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a metal lid.
D. The terminals are gold plated.
E. Falls within JEDEC MS-004
N (R-PDIP-T**)

16 PINS SHOWN

PLASTIC DUAL-IN-LINE PACKAGE

<table>
<thead>
<tr>
<th>PINS **</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>A MAX</td>
<td>0.775 (19.69)</td>
<td>0.775 (19.69)</td>
<td>0.920 (23.37)</td>
<td>1.060 (26.92)</td>
</tr>
<tr>
<td>A MIN</td>
<td>0.745 (18.92)</td>
<td>0.745 (18.92)</td>
<td>0.850 (21.59)</td>
<td>0.940 (23.88)</td>
</tr>
<tr>
<td>MS-001</td>
<td>AA</td>
<td>BB</td>
<td>AC</td>
<td>AD</td>
</tr>
<tr>
<td>VARIATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0.045 (1.14)</th>
<th>0.030 (0.76)</th>
<th>0.005 (0.13)</th>
<th>0.015 (0.38)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN</td>
<td>0.020 (0.51)</td>
<td>0.020 (0.51)</td>
<td>0.015 (0.38)</td>
<td>0.010 (0.25)</td>
</tr>
<tr>
<td>MAX</td>
<td>0.200 (5.08)</td>
<td>0.200 (5.08)</td>
<td>Seating Plane</td>
<td>14/18 Pin Only 20 Pin vendor option</td>
</tr>
<tr>
<td>MIN</td>
<td>0.125 (3.18)</td>
<td>0.125 (3.18)</td>
<td>14/18 Pin Only 20 Pin vendor option</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>0.430 (10.92)</td>
<td>0.430 (10.92)</td>
<td>14/18 Pin Only 20 Pin vendor option</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.325 (8.26)</td>
<td>0.325 (8.26)</td>
<td>0.300 (7.62)</td>
<td>0.015 (0.38)</td>
</tr>
<tr>
<td></td>
<td>0.010 (0.25)</td>
<td>0.010 (0.25)</td>
<td>0.010 (0.25)</td>
<td>0.010 (0.25)</td>
</tr>
</tbody>
</table>

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-001, except 18 and 20 pin minimum body length (Dim A).
D. The 20 pin end lead shoulder width is a vendor option, either half or full width.

4040049/E 12/2002
NOTES:

A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.

⚠️ Body length does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed .006 (0.15) per end.

⚠️ Body width does not include interlead flash. Interlead flash shall not exceed .017 (0.43) per side.

E. Reference JEDEC MS-012 variation AB.
NOTES:

A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.15.
IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifiers</td>
<td>Audio</td>
</tr>
<tr>
<td>Data Converters</td>
<td>Automotive</td>
</tr>
<tr>
<td>DSP</td>
<td>Broadband</td>
</tr>
<tr>
<td>Interface</td>
<td>Digital Control</td>
</tr>
<tr>
<td>Logic</td>
<td>Military</td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>Optical Networking</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>Security</td>
</tr>
<tr>
<td>Low Power Wireless</td>
<td>Telephony</td>
</tr>
<tr>
<td></td>
<td>Video &amp; Imaging</td>
</tr>
<tr>
<td></td>
<td>Wireless</td>
</tr>
</tbody>
</table>

Mailing Address: Texas Instruments  
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2006, Texas Instruments Incorporated
5. SN74HCT74: Dual D-Type Positive-Edge-Triggered Flip Flops

**Texas Instruments**
Dual D-Type Positive-Edge-Triggered Flip Flops with Clear and Preset
SN74HCT74
SN54HCT74, SN74HCT74
DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS
WITH CLEAR AND PRESET
SCLS169D – DECEMBER 1982 – REVISED AUGUST 2003

- Operating Voltage Range of 4.5 V to 5.5 V
- Outputs Can Drive Up To 10 LSTTL Loads
- Low Power Consumption, 40-μA Max I_{CC}
- Typical $t_{pd} = 17$ ns
- ±4-mA Output Drive at 5 V
- Low Input Current of 1 μA Max
- Inputs Are TTL-Voltage Compatible

**Description/Ordering Information**

The 'HCT74 devices contain two independent D-type positive-edge-triggered flip-flops. A low level at the preset (PRE) or clear (CLR) inputs sets or resets the outputs, regardless of the levels of the other inputs. When PRE and CLR are inactive (high), data at the data (D) input meeting the setup time requirements are transferred to the outputs on the positive-going edge of the clock (CLK) pulse. Clock triggering occurs at a voltage level and is not directly related to the rise time of CLK. Following the hold-time interval, data at the D input may be changed without affecting the levels at the outputs.

**Ordering Information**

<table>
<thead>
<tr>
<th>$T_A$</th>
<th>Package™</th>
<th>Orderable Part Number</th>
<th>Top-Side Marking</th>
</tr>
</thead>
<tbody>
<tr>
<td>-40°C to 85°C</td>
<td>PDIP – N</td>
<td>Tube of 25</td>
<td>SN74HCT74N</td>
</tr>
<tr>
<td></td>
<td>Tube of 50</td>
<td></td>
<td>SN74HCT74D</td>
</tr>
<tr>
<td></td>
<td>Reel of 2500</td>
<td></td>
<td>SN74HCT74DR</td>
</tr>
<tr>
<td></td>
<td>Reel of 250</td>
<td></td>
<td>SN74HCT74DT</td>
</tr>
<tr>
<td></td>
<td>SOP – NS</td>
<td>Reel of 2000</td>
<td>SN74HCT74NSR</td>
</tr>
<tr>
<td></td>
<td>SSOP – DB</td>
<td>Reel of 2000</td>
<td>SN74HCT74DBR</td>
</tr>
<tr>
<td></td>
<td>TSSOP – PW</td>
<td>Tube of 90</td>
<td>SN74HCT74PW</td>
</tr>
<tr>
<td></td>
<td>Reel of 2000</td>
<td></td>
<td>SN74HCT74PWR</td>
</tr>
<tr>
<td></td>
<td>Reel of 250</td>
<td></td>
<td>SN74HCT74PWT</td>
</tr>
<tr>
<td>-55°C to 125°C</td>
<td>CDIP – J</td>
<td>Tube of 25</td>
<td>SNJ54HCT74J</td>
</tr>
<tr>
<td></td>
<td>Tube of 150</td>
<td></td>
<td>SNJ54HCT74W</td>
</tr>
<tr>
<td></td>
<td>LCCC – FK</td>
<td>Tube of 55</td>
<td>SNJ54HCT74FK</td>
</tr>
</tbody>
</table>

™ Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.
### Function Table

<table>
<thead>
<tr>
<th>PRE</th>
<th>CLR</th>
<th>CLK</th>
<th>D</th>
<th>Q</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>H</td>
<td>X</td>
<td>X</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>X</td>
<td>X</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>X</td>
<td>X</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>↑</td>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>↑</td>
<td>L</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>L</td>
<td>X</td>
<td>Q₀</td>
<td>Q₀</td>
</tr>
</tbody>
</table>

† This configuration is nonstable; that is, it does not persist when PRE or CLR returns to its inactive (high) level.

### Logic Diagram (Positive Logic)

![Logic Diagram](image)

### Absolute Maximum Ratings Over Operating Free-Air Temperature Range (Unless Otherwise Noted)‡

- Supply voltage range, \( V_{CC} \): \(-0.5 \text{ to } 7 \text{ V}\)
- Input clamp current, \( I_{IK} \) (\( V_I < 0 \) or \( V_I > V_{CC} \)) (see Note 1): \( \pm 20 \text{ mA} \)
- Output clamp current, \( I_{OK} \) (\( V_O < 0 \) or \( V_O > V_{CC} \)) (see Note 1): \( \pm 20 \text{ mA} \)
- Continuous output current, \( I_O \) (\( V_O = 0 \) to \( V_{CC} \)): \( \pm 25 \text{ mA} \)
- Continuous current through \( V_{CC} \) or GND: \( \pm 50 \text{ mA} \)
- Package thermal impedance, \( \theta_{JA} \) (see Note 2): D package: \( 86 \text{°C/W} \)
  - DB package: \( 96 \text{°C/W} \)
  - N package: \( 80 \text{°C/W} \)
  - NS package: \( 76 \text{°C/W} \)
  - PW package: \( 113 \text{°C/W} \)
- Storage temperature range, \( T_{stg} \): \(-65 \text{°C to } 150 \text{°C} \)

‡ Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES:
1. The input and output voltage ratings may be exceeded if the input and output current ratings are observed.
2. The package thermal impedance is calculated in accordance with JESD 51-7.
recommended operating conditions (see Note 3)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>V CC</th>
<th>SN54HCT74</th>
<th>SN74HCT74</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{CC} )</td>
<td>Supply voltage</td>
<td></td>
<td>MIN</td>
<td>NOM</td>
<td>MAX</td>
</tr>
<tr>
<td>( V_{IH} )</td>
<td>High-level input voltage</td>
<td>4.5 V to 5.5 V</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>( V_{IL} )</td>
<td>Low-level input voltage</td>
<td>4.5 V to 5.5 V</td>
<td>0.8</td>
<td>0.8</td>
<td>V</td>
</tr>
<tr>
<td>( V_I )</td>
<td>Input voltage</td>
<td></td>
<td>0</td>
<td>( V_{CC} )</td>
<td>0</td>
</tr>
<tr>
<td>( V_O )</td>
<td>Output voltage</td>
<td></td>
<td>0</td>
<td>( V_{CC} )</td>
<td>0</td>
</tr>
<tr>
<td>( \Delta t/\Delta v )</td>
<td>Input transition rise/fall time</td>
<td></td>
<td>500</td>
<td>500</td>
<td>ns</td>
</tr>
<tr>
<td>( T_A )</td>
<td>Operating free-air temperature</td>
<td></td>
<td>85</td>
<td>125</td>
<td>85</td>
</tr>
</tbody>
</table>

**NOTE 3:** All unused inputs of the device must be held at \( V_{CC} \) or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>V CC</th>
<th>SN54HCT74</th>
<th>SN74HCT74</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{OH} )</td>
<td>( V_I = V_{IH} ) or ( V_{IL} )</td>
<td>4.5 V</td>
<td>4.4</td>
<td>4.999</td>
<td>4.4</td>
</tr>
<tr>
<td>( I_{OH} )</td>
<td>( V_I = V_{IH} ) or ( V_{IL} )</td>
<td>4.5 V</td>
<td>3.98</td>
<td>4.3</td>
<td>3.7</td>
</tr>
<tr>
<td>( V_{OL} )</td>
<td>( V_I = V_{IH} ) or ( V_{IL} )</td>
<td>4.5 V</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>( I_{IL} )</td>
<td>( V_I = V_{CC} ) or 0</td>
<td>5.5 V</td>
<td>( \pm 0.1 )</td>
<td>( \pm 100 )</td>
<td>( \pm 1000 )</td>
</tr>
<tr>
<td>( I_{CC} )</td>
<td>( V_I = V_{CC} ) or 0, ( I_O = 0 )</td>
<td>5.5 V</td>
<td>4</td>
<td>80</td>
<td>40</td>
</tr>
<tr>
<td>( \Delta I_{CC} ) †</td>
<td>One input at 0.5 V or 2.4 V, Other inputs at 0 or ( V_{CC} )</td>
<td>5.5 V</td>
<td>1.4</td>
<td>2.4</td>
<td>3</td>
</tr>
<tr>
<td>( C_i )</td>
<td>4.5 V to 5.5 V</td>
<td>3</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

† This is the increase in supply current for each input that is at one of the specified TTL voltage levels, rather than 0 V or \( V_{CC} \).

timing requirements over recommended operating free-air temperature range (unless otherwise noted)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>V CC</th>
<th>SN54HCT74</th>
<th>SN74HCT74</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{clock} )</td>
<td>Clock frequency</td>
<td>4.5 V</td>
<td>27</td>
<td>18</td>
</tr>
<tr>
<td>( f_{W} )</td>
<td>Pulse duration</td>
<td>5.5 V</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>( f_{su} )</td>
<td>Setup time before ( CLK ) †</td>
<td>4.5 V</td>
<td>12</td>
<td>18</td>
</tr>
<tr>
<td>( f_{H} )</td>
<td>Hold time, data after ( CLK ) †</td>
<td>5.5 V</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

PRODUCT PREVIEW information concerns products in the formative or design phase of development. Characteristic data and other specifications are design goals. Texas Instruments reserves the right to change or discontinue these products without notice.

Texas Instruments
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

3
SN54HCT74, SN74HCT74
DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS
WITH CLEAR AND PRESET
SCLS169D – DECEMBER 1982 – REVISED AUGUST 2003

switching characteristics over recommended operating free-air temperature range, $C_L = 50 \text{ pF}$ (unless otherwise noted) (see Figure 1)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>FROM (INPUT)</th>
<th>TO (OUTPUT)</th>
<th>$V_{CC}$</th>
<th>$T_A = 25^\circ C$</th>
<th>SN54HCT74</th>
<th>SN74HCT74</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{max}$</td>
<td>PRE or CLR</td>
<td>$Q$ or $\bar{Q}$</td>
<td>4.5 V</td>
<td>21</td>
<td>53</td>
<td>44</td>
<td>MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 V</td>
<td>17</td>
<td>48</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>$t_{pd}$</td>
<td>CLK</td>
<td>$Q$ or $\bar{Q}$</td>
<td>4.5 V</td>
<td>20</td>
<td>42</td>
<td>35</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 V</td>
<td>18</td>
<td>38</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>$t_t$</td>
<td>$Q$ or $\bar{Q}$</td>
<td>4.5 V</td>
<td>8</td>
<td>15</td>
<td>22</td>
<td>19</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 V</td>
<td>7</td>
<td>14</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

operating characteristics, $T_A = 25^\circ C$

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>TYP</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{pd}$</td>
<td>No load</td>
<td>35</td>
<td>pF</td>
</tr>
</tbody>
</table>

PARAMETER MEASUREMENT INFORMATION

NOTES:
A. $C_L$ includes probe and test-fixture capacitance.
B. Phase relationships between waveforms were chosen arbitrarily. All input pulses are supplied by generators having the following characteristics: PRR $\leq 1 \text{ MHz}$, $Z_O = 50 \Omega$, $t_r = 6 \text{ ns}$, $t_f = 6 \text{ ns}$.
C. For clock inputs, $f_{max}$ is measured when the input duty cycle is 50%.
D. The outputs are measured one at a time with one input transition per measurement.
E. $t_{PLH}$ and $t_{PHL}$ are the same as $t_{pd}$.

Figure 1. Load Circuit and Voltage Waveforms

PRODUCT PREVIEW information concerns products in the formative or design phase of development. Characteristic data and other specifications are design goals. Texas Instruments reserves the right to change or discontinue these products without notice.
J (R-GDIP-T**)  
14 LEADS SHOWN  

CERAMIC DUAL IN-LINE PACKAGE

<table>
<thead>
<tr>
<th>Pins **</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.300 (7.62) BSC</td>
<td>0.300 (7.62) BSC</td>
<td>0.300 (7.62) BSC</td>
<td>0.300 (7.62) BSC</td>
</tr>
<tr>
<td>B MAX</td>
<td>0.785 (19.94)</td>
<td>0.840 (21.34)</td>
<td>0.960 (24.38)</td>
<td>1.060 (26.92)</td>
</tr>
<tr>
<td>B MIN</td>
<td>——</td>
<td>——</td>
<td>——</td>
<td>——</td>
</tr>
<tr>
<td>C MAX</td>
<td>0.300 (7.62)</td>
<td>0.300 (7.62)</td>
<td>0.310 (7.87)</td>
<td>0.300 (7.62)</td>
</tr>
<tr>
<td>C MIN</td>
<td>0.245 (6.22)</td>
<td>0.245 (6.22)</td>
<td>0.220 (5.59)</td>
<td>0.245 (6.22)</td>
</tr>
</tbody>
</table>

NOTES:  
A. All linear dimensions are in inches (millimeters).  
B. This drawing is subject to change without notice.  
C. This package is hermetically sealed with a ceramic lid using glass frit.  
D. Index point is provided on cap for terminal identification only on press ceramic glass frit seal only.  
E. Falls within MIL STD 1835 GDIP1-T14, GDIP1-T16, GDIP1-T18 and GDIP1-T20.

4040083/F 03/03
W (R-GDFP-F14) CERAMIC DUAL FLATPACK

NOTES:  
A. All linear dimensions are in inches (millimeters).  
B. This drawing is subject to change without notice.  
C. This package can be hermetically sealed with a ceramic lid using glass frit.  
D. Index point is provided on cap for terminal identification only.  
E. Falls within MIL STD 1835 GDFP1-F14 and JEDEC MO-092AB
FK (S-CQCC-N**)

28 TERMINAL SHOWN

<table>
<thead>
<tr>
<th>NO. OF TERMINALS **</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIN</td>
<td>MAX</td>
</tr>
<tr>
<td>20</td>
<td>0.342</td>
<td>(8,69)</td>
</tr>
<tr>
<td></td>
<td>0.307</td>
<td>(7,80)</td>
</tr>
<tr>
<td>28</td>
<td>0.442</td>
<td>(11,23)</td>
</tr>
<tr>
<td></td>
<td>0.406</td>
<td>(10,31)</td>
</tr>
<tr>
<td>44</td>
<td>0.640</td>
<td>(16,26)</td>
</tr>
<tr>
<td></td>
<td>0.495</td>
<td>(12,58)</td>
</tr>
<tr>
<td>52</td>
<td>0.739</td>
<td>(18,78)</td>
</tr>
<tr>
<td></td>
<td>0.495</td>
<td>(12,58)</td>
</tr>
<tr>
<td>68</td>
<td>0.938</td>
<td>(23,83)</td>
</tr>
<tr>
<td></td>
<td>0.850</td>
<td>(21,6)</td>
</tr>
<tr>
<td>84</td>
<td>1.141</td>
<td>(28,99)</td>
</tr>
<tr>
<td></td>
<td>1.047</td>
<td>(26,6)</td>
</tr>
</tbody>
</table>

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a metal lid.
D. The terminals are gold plated.
E. Falls within JEDEC MS-004
N (R-PDIP-T**)

16 PINS SHOWN

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-001, except 18 and 20 pin minimum body length (Dim A).
D. The 20 pin end lead shoulder width is a vendor option, either half or full width.
NOTES:

A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.006 (0.15).
D. Falls within JEDEC MS-012
NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.15.
NOTES:  
A. All linear dimensions are in millimeters. 
B. This drawing is subject to change without notice. 
C. Body dimensions do not include mold flash or protrusion not to exceed 0.15. 
D. Falls within JEDEC MO-150
PW (R-PDSO-G**)  
PLASTIC SMALL-OUTLINE PACKAGE

14 PINS SHOWN

NOTES:  
A. All linear dimensions are in millimeters.  
B. This drawing is subject to change without notice.  
C. Body dimensions do not include mold flash or protrusion not to exceed 0.15.  
D. Falls within JEDEC MO-153
## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI’s standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifiers</td>
<td>Audio</td>
</tr>
<tr>
<td>Data Converters</td>
<td>Automotive</td>
</tr>
<tr>
<td>DSP</td>
<td>Broadband</td>
</tr>
<tr>
<td>Interface</td>
<td>Digital Control</td>
</tr>
<tr>
<td>Logic</td>
<td>Military</td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>Optical Networking</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td>Telephony</td>
</tr>
<tr>
<td></td>
<td>Video &amp; Imaging</td>
</tr>
<tr>
<td></td>
<td>Wireless</td>
</tr>
</tbody>
</table>

Mailing Address: Texas Instruments  
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2003, Texas Instruments Incorporated
6. SN74AHCT541: Octal Buffers/Drivers with 3-State Outputs

Texas Instruments
Octal Buffers/Drivers with 3-State Outputs
SN74AHCT541
Inputs Are TTL-Voltage Compatible
- Latch-Up Performance Exceeds 250 mA Per JESD 17
- ESD Protection Exceeds JESD 22
  - 2000-V Human-Body Model (A114-A)
  - 200-V Machine Model (A115-A)
  - 1000-V Charged-Device Model (C101)

description/ordering information

The 'AHCT541 octal buffers/drivers are ideal for driving bus lines or buffer memory address registers. These devices feature inputs and outputs on opposite sides of the package to facilitate printed circuit board layout.

The 3-state control gate is a 2-input AND gate with active-low inputs so that if either output-enable ($OE_1$ or $OE_2$) input is high, all corresponding outputs are in the high-impedance state. The outputs provide noninverted data when they are not in the high-impedance state.

To ensure the high-impedance state during power up or power down, $OE$ should be tied to $V_{CC}$ through a pullup resistor; the minimum value of the resistor is determined by the current-sinking capability of the driver.

ORDERING INFORMATION

<table>
<thead>
<tr>
<th>$TA$</th>
<th>PACKAGE†</th>
<th>ORDERABLE PART NUMBER</th>
<th>TOP-SIDE MARKING</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PDIP – N</td>
<td>Tube</td>
<td>SN74AHCT541N</td>
</tr>
<tr>
<td></td>
<td>SOIC – DW</td>
<td>Tube</td>
<td>SN74AHCT541DW</td>
</tr>
<tr>
<td></td>
<td>SOP – NS</td>
<td>Tape and reel</td>
<td>SN74AHCT541NSR</td>
</tr>
<tr>
<td></td>
<td>SSOP – DB</td>
<td>Tape and reel</td>
<td>SN74AHCT541DBR</td>
</tr>
<tr>
<td></td>
<td>TSSOP – PW</td>
<td>Tape and reel</td>
<td>SN74AHCT541PW</td>
</tr>
<tr>
<td></td>
<td>TVSOP – DGV</td>
<td>Tape and reel</td>
<td>SN74AHCT541DGVR</td>
</tr>
<tr>
<td>$-40°C$ to $85°C$</td>
<td>CDIP – J</td>
<td>Tube</td>
<td>SNJ54AHCT541J</td>
</tr>
<tr>
<td></td>
<td>CFP – W</td>
<td>Tube</td>
<td>SNJ54AHCT541W</td>
</tr>
<tr>
<td>$-55°C$ to $125°C$</td>
<td>LCIC – FK</td>
<td>Tube</td>
<td>SNJ54AHCT541FK</td>
</tr>
</tbody>
</table>

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.
FUNCTION TABLE
(each buffer/driver)

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OE1</td>
<td>OE2</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>H</td>
</tr>
</tbody>
</table>

logic diagram (positive logic)

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, \( V_{CC} \) ................................................................. \(-0.5 \text{ V} \) to \(7 \text{ V}\)
Input voltage range, \( V_I \) (see Note 1) ................................................................. \(-0.5 \text{ V} \) to \(7 \text{ V}\)
Output voltage range, \( V_O \) (see Note 1) ................................................................. \(-0.5 \text{ V} \) to \(V_{CC} + 0.5 \text{ V}\)
Input clamp current, \( I_{IK} \) (\( V_I < 0 \)) ................................................................. \(-20 \text{ mA}\)
Output clamp current, \( I_{OK} \) (\( V_O < 0 \) or \( V_O > V_{CC} \)) ................................................................. \(\pm 20 \text{ mA}\)
Continuous output current, \( I_O \) (\( V_O = 0 \) to \( V_{CC} \)) ................................................................. \(\pm 25 \text{ mA}\)
Continuous current through \( V_{CC} \) or GND ................................................................. \(\pm 75 \text{ mA}\)
Package thermal impedance, \( \theta_{JA} \) (see Note 2): DB package ................................. \(70{\degree}\text{C/W}\)
DGV package ................................................................. \(92{\degree}\text{C/W}\)
DW package ................................................................. \(58{\degree}\text{C/W}\)
N package ................................................................. \(69{\degree}\text{C/W}\)
NS package ................................................................. \(60{\degree}\text{C/W}\)
PW package ................................................................. \(83{\degree}\text{C/W}\)
Storage temperature range, \( T_{stg} \) ................................................................. \(-65{\degree}\text{C} \) to \(150{\degree}\text{C}\)

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. The input and output voltage ratings may be exceeded if the input and output current ratings are observed.
2. The package thermal impedance is calculated in accordance with JESD 51-7.
### recommended operating conditions (see Note 3)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CC}$ Supply voltage</td>
<td>4.5</td>
<td>5.5</td>
<td>4.5</td>
<td>5.5</td>
</tr>
<tr>
<td>$V_{IH}$ High-level input voltage</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$V_{IL}$ Low-level input voltage</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>$V_I$ Input voltage</td>
<td>0</td>
<td>5.5</td>
<td>0</td>
<td>5.5</td>
</tr>
<tr>
<td>$V_O$ Output voltage</td>
<td>0</td>
<td>$V_{CC}$</td>
<td>0</td>
<td>$V_{CC}$</td>
</tr>
<tr>
<td>$I_{OH}$ High-level output current</td>
<td>–8</td>
<td>–8</td>
<td>–8</td>
<td>–8 mA</td>
</tr>
<tr>
<td>$I_{OL}$ Low-level output current</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>mA</td>
</tr>
<tr>
<td>$\Delta I / \Delta V$ Input transition rise or fall rate</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>ns/V</td>
</tr>
<tr>
<td>$T_A$ Operating free-air temperature</td>
<td>–55</td>
<td>125</td>
<td>–40</td>
<td>85 °C</td>
</tr>
</tbody>
</table>

**NOTE 3:** All unused inputs of the device must be held at $V_{CC}$ or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.

### electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>$V_{CC}$</th>
<th>$T_A = 25°C$</th>
<th>SN54AHCT541</th>
<th>SN74AHCT541</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{OH}$</td>
<td>$I_{OH} = –50 \mu A$</td>
<td>4.5 V</td>
<td>4.4</td>
<td>4.5</td>
<td>4.4</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td>$I_{OH} = –8 mA$</td>
<td></td>
<td>3.94</td>
<td>3.8</td>
<td>3.8</td>
<td></td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>$I_{OL} = 50 \mu A$</td>
<td>4.5 V</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I_{OL} = 8 mA$</td>
<td></td>
<td>0.36</td>
<td>0.44</td>
<td>0.44</td>
<td></td>
</tr>
<tr>
<td>$I_I$</td>
<td>$V_I = 5.5 V$ or GND</td>
<td>0 V to 5.5 V</td>
<td>±0.1</td>
<td>±0.1</td>
<td>±1*</td>
<td>±1</td>
</tr>
<tr>
<td>$I_{OZ}$</td>
<td>$V_O = V_{CC}$ or GND</td>
<td>5.5 V</td>
<td>±0.25</td>
<td>±2.5</td>
<td>±2.5</td>
<td>±2.5</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>$V_I = V_{CC}$ or GND, $I_O = 0$</td>
<td>5.5 V</td>
<td>4</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>$\Delta I_{CC}$†</td>
<td>One input at 3.4 V, Other inputs at $V_{CC}$ or GND</td>
<td>5.5 V</td>
<td>1.35</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>$C_I$</td>
<td>$V_I = V_{CC}$ or GND</td>
<td>5 V</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>$C_O$</td>
<td>$V_O = V_{CC}$ or GND</td>
<td>5 V</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* On products compliant to MIL-PRF-38535, this parameter is not production tested at $V_{CC} = 0$ V.
† This is the increase in supply current for each input at one of the specified TTL voltage levels, rather than 0 V or $V_{CC}$. 
switching characteristics over recommended operating free-air temperature range, $V_{CC} = 5\, \text{V} \pm 0.5\, \text{V}$ (unless otherwise noted) (see Figure 1)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>FROM (INPUT)</th>
<th>TO (OUTPUT)</th>
<th>LOAD CAPACITANCE</th>
<th>$T_A = 25^\circ\text{C}$</th>
<th>SN54AHCT541</th>
<th>SN74AHCT541</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$C_L = 15, \text{pF}$</td>
<td>MIN</td>
<td>TYP</td>
<td>MAX</td>
<td>MIN</td>
</tr>
<tr>
<td>tPLH</td>
<td>A</td>
<td>Y</td>
<td></td>
<td>4.1*</td>
<td>6*</td>
<td>1*</td>
<td>6.5*</td>
</tr>
<tr>
<td>tPHL</td>
<td></td>
<td></td>
<td></td>
<td>3.7*</td>
<td>5.5*</td>
<td>1*</td>
<td>6.5*</td>
</tr>
<tr>
<td>tPZH</td>
<td>OE</td>
<td>Y</td>
<td></td>
<td>5*</td>
<td>7*</td>
<td>1*</td>
<td>8*</td>
</tr>
<tr>
<td>tPZL</td>
<td></td>
<td></td>
<td></td>
<td>5*</td>
<td>7*</td>
<td>1*</td>
<td>8*</td>
</tr>
<tr>
<td>tPHZ</td>
<td>OE</td>
<td>Y</td>
<td></td>
<td>4.5*</td>
<td>7*</td>
<td>1*</td>
<td>8*</td>
</tr>
<tr>
<td>tPZL</td>
<td></td>
<td></td>
<td></td>
<td>4.5*</td>
<td>7*</td>
<td>1*</td>
<td>8*</td>
</tr>
<tr>
<td>tPLH</td>
<td>A</td>
<td>Y</td>
<td></td>
<td>6.2</td>
<td>8.5</td>
<td>1</td>
<td>9.5</td>
</tr>
<tr>
<td>tPHL</td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>8.5</td>
<td>1</td>
<td>9.5</td>
</tr>
<tr>
<td>tPZH</td>
<td>OE</td>
<td>Y</td>
<td></td>
<td>7.5</td>
<td>10</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>tPZL</td>
<td></td>
<td></td>
<td></td>
<td>7.5</td>
<td>10</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>tPHZ</td>
<td>OE</td>
<td>Y</td>
<td></td>
<td>7</td>
<td>10</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>tPZL</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>10</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>tsk(o)</td>
<td></td>
<td></td>
<td>$C_L = 50, \text{pF}$</td>
<td>1**</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* On products compliant to MIL-PRF-38535, this parameter is not production tested.
** On products compliant to MIL-PRF-38535, this parameter does not apply.

operating characteristics, $V_{CC} = 5\, \text{V}$, $T_A = 25^\circ\text{C}$

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>TYP</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{pd}$</td>
<td>No load, $f = 1, \text{MHz}$</td>
<td>12</td>
<td>pF</td>
</tr>
</tbody>
</table>
PARAMETER MEASUREMENT INFORMATION

From Output Under Test ● Test Point ● From Output Under Test

\( R_L = 1 \, k\Omega \)

\( S_1 \)

\( V_{CC} \)

\( \text{Open} \) \( \text{GND} \)

LOAD CIRCUIT FOR TOTEM-POLE OUTPUTS

LOAD CIRCUIT FOR 3-STATE AND OPEN-DRAIN OUTPUTS

VOLTAGE WAVEFORMS
PULSE DURATION

Input

1.5 V

3 V

1.5 V

0 V

VOLTAGE WAVEFORMS
PROPAGATION DELAY TIMES
INVERTING AND NONINVERTING OUTPUTS

In-Phase Output

Out-of-Phase Output

Input

1.5 V

3 V

1.5 V

0 V

\( t_{PLH} \)

\( t_{PLZ} \)

\( t_{PZH} \)

\( t_{PHZ} \)

\( t_{PHL} \)

\( t_{PZH} \)

\( V_{OH} \)

\( 50\% \, V_{CC} \)

\( V_{OL} \)

\( 50\% \, V_{CC} \)

\( V_{OL} \)

\( V_{OH} \)

\( 50\% \, V_{CC} \)

\( V_{OL} \)

\( 50\% \, V_{CC} \)

\( V_{OL} \)

\( \approx \)

\( V_{CC} \)

\( 0 \, V \)

\( \approx \)

\( 0 \, V \)

\( V_{CC} \)

\( V_{OL} \)

\( V_{OL} + 0.3 \, V \)

\( V_{OL} \)

\( V_{OH} - 0.3 \, V \)

\( V_{OH} \)

\( \leq \)

1 MHz

\( Z_O = 50 \, \Omega \)

\( t_r \leq 3 \, ns \)

\( t_f \leq 3 \, ns \)

NOTES:
A. \( C_L \) includes probe and jig capacitance.
B. Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control. Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control.
C. All input pulses are supplied by generators having the following characteristics: \( PRR \leq 1 \, MHz \), \( Z_O = 50 \, \Omega \), \( t_r \leq 3 \, ns \), \( t_f \leq 3 \, ns \).
D. The outputs are measured one at a time with one input transition per measurement.
E. All parameters and waveforms are not applicable to all devices.

Figure 1. Load Circuit and Voltage Waveforms
**CERAMIC DUAL IN-LINE PACKAGE**

**J (R-GDIP-T**)**

14 LEADS SHOWN

---

<table>
<thead>
<tr>
<th>PINS **</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIM A</td>
<td>0.300 (7.62) BSC</td>
<td>0.300 (7.62) BSC</td>
<td>0.300 (7.62) BSC</td>
<td>0.300 (7.62) BSC</td>
</tr>
<tr>
<td>B MAX</td>
<td>0.785 (19.94)</td>
<td>0.840 (21.34)</td>
<td>0.960 (24.38)</td>
<td>1.060 (26.92)</td>
</tr>
<tr>
<td>B MIN</td>
<td>——</td>
<td>——</td>
<td>——</td>
<td>——</td>
</tr>
<tr>
<td>C MAX</td>
<td>0.300 (7.62)</td>
<td>0.300 (7.62)</td>
<td>0.310 (7.87)</td>
<td>0.300 (7.62)</td>
</tr>
<tr>
<td>C MIN</td>
<td>0.245 (6.22)</td>
<td>0.245 (6.22)</td>
<td>0.220 (5.59)</td>
<td>0.245 (6.22)</td>
</tr>
</tbody>
</table>

---

**NOTES:**
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package is hermetically sealed with a ceramic lid using glass frit.
D. Index point is provided on cap for terminal identification only on press ceramic glass frit seal only.
E. Falls within MIL STD 1835 GDIP1-T14, GDIP1-T16, GDIP1-T18 and GDIP1-T20.

4040083/F 03/03
W (R-GDFP-F20)  CERAMIC DUAL FLATPACK

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a ceramic lid using glass frit.
D. Index point is provided on cap for terminal identification only.
E. Falls within Mil-Std 1835 GDFP2-F20
FK (S-CQCC-N**)

LEADLESS CERAMIC CHIP CARRIER

28 TERMINAL SHOWN

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a metal lid.
D. The terminals are gold plated.
E. Falls within JEDEC MS-004

<table>
<thead>
<tr>
<th>NO. OF TERMINALS **</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIN</td>
<td>MAX</td>
</tr>
<tr>
<td>20</td>
<td>0.342 (8.69)</td>
<td>0.358 (9.09)</td>
</tr>
<tr>
<td>28</td>
<td>0.442 (11.23)</td>
<td>0.458 (11.63)</td>
</tr>
<tr>
<td>44</td>
<td>0.640 (16.26)</td>
<td>0.660 (16.76)</td>
</tr>
<tr>
<td>52</td>
<td>0.739 (18.78)</td>
<td>0.761 (19.32)</td>
</tr>
<tr>
<td>68</td>
<td>0.938 (23.83)</td>
<td>0.962 (24.43)</td>
</tr>
<tr>
<td>84</td>
<td>1.141 (28.99)</td>
<td>1.165 (29.59)</td>
</tr>
</tbody>
</table>

NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a metal lid.
D. The terminals are gold plated.
E. Falls within JEDEC MS-004
N (R-PDIP-T**)

16 PINS SHOWN

NOTES:  
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-001, except 18 and 20 pin minimum body length (Dim A).
D. The 20 pin end lead shoulder width is a vendor option, either half or full width.
MECHANICAL DATA

DW (R-PDSO-G**)

PLASTIC SMALL-OUTLINE PACKAGE

16 PINS SHOWN

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0.006 (0.15).
D. Falls within JEDEC MS-013
## MECHANICAL DATA

### NS (R-PDSO-G**)

**PLASTIC SMALL-OUTLINE PACKAGE**

14-PINS SHOWN

<table>
<thead>
<tr>
<th>DIM</th>
<th>14</th>
<th>16</th>
<th>20</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>A MAX</td>
<td>10,50</td>
<td>10,50</td>
<td>12,90</td>
<td>15,30</td>
</tr>
<tr>
<td>A MIN</td>
<td>9,90</td>
<td>9,90</td>
<td>12,30</td>
<td>14,70</td>
</tr>
</tbody>
</table>

### NOTES:

A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0,15.

4040062/C 03/03
NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0.15.
D. Falls within JEDEC MO-150.
### PW (R-PDSO-G**) PLASTIC SMALL-OUTLINE PACKAGE

#### 14 PINS SHOWN

**NOTES:**

A. All linear dimensions are in millimeters.

B. This drawing is subject to change without notice.

C. Body dimensions do not include mold flash or protrusion not to exceed 0.15.

D. Falls within JEDEC MO-153

<table>
<thead>
<tr>
<th>DIM</th>
<th>PINS **</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td>A MAX</td>
<td>3.10</td>
</tr>
<tr>
<td>A MIN</td>
<td>2.90</td>
</tr>
</tbody>
</table>

4040064/F 01/97
IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifiers</td>
<td>Audio</td>
</tr>
<tr>
<td>Data Converters</td>
<td>Automotive</td>
</tr>
<tr>
<td>DSP</td>
<td>Broadband</td>
</tr>
<tr>
<td>Interface</td>
<td>Digital Control</td>
</tr>
<tr>
<td>Logic</td>
<td>Military</td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>Optical Networking</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td>Telephony</td>
</tr>
<tr>
<td></td>
<td>Video &amp; Imaging</td>
</tr>
<tr>
<td></td>
<td>Wireless</td>
</tr>
</tbody>
</table>

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2003, Texas Instruments Incorporated
7. SN74HC21: Dual 4-Input Positive-AND Gates

Texas Instruments
Dual 4-Input Positive-AND Gates
SN74HC21
Wide Operating Voltage Range of 2 V to 6 V
Outputs Can Drive Up To 10 LSTTL Loads
Low Power Consumption, 20-µA Max \( I_{CC} \)

Typical \( t_{pd} \) = 11 ns
±4-mA Output Drive at 5 V
Low Input Current of 1 µA Max

These devices contain two independent 4-input AND gates. They perform the Boolean function \( Y = A \cdot B \cdot C \cdot D \) or \( Y = \overline{A} + \overline{B} + \overline{C} + \overline{D} \) in positive logic.

### ORDERING INFORMATION

<table>
<thead>
<tr>
<th>TA</th>
<th>PACKAGE†</th>
<th>ORDERABLE PART NUMBER</th>
<th>TOP-SIDE MARKING</th>
</tr>
</thead>
<tbody>
<tr>
<td>–40°C to 85°C</td>
<td>PDIP – N</td>
<td>Tube of 25</td>
<td>SN74HC21N</td>
</tr>
<tr>
<td></td>
<td>SOIC – D</td>
<td>Tube of 50</td>
<td>SN74HC21D</td>
</tr>
<tr>
<td></td>
<td>SOP – NS</td>
<td>Reel of 2500</td>
<td>SN74HC21DR</td>
</tr>
<tr>
<td></td>
<td>TSSOP – PW</td>
<td>Tube of 90</td>
<td>SN74HC21PWL</td>
</tr>
<tr>
<td></td>
<td>CDIP – J</td>
<td>Tube of 25</td>
<td>SNJ54HC21J</td>
</tr>
<tr>
<td></td>
<td>CFP – W</td>
<td>Tube of 150</td>
<td>SNJ54HC21W</td>
</tr>
<tr>
<td></td>
<td>LCCC – FK</td>
<td>Tube of 55</td>
<td>SNJ54HC21FK</td>
</tr>
</tbody>
</table>

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.
FUNCTION TABLE
(each gate)

| INPUTS | OUTPUT 
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C D</td>
<td>Y</td>
</tr>
<tr>
<td>H H H H</td>
<td>H</td>
</tr>
<tr>
<td>L X X X</td>
<td>L</td>
</tr>
<tr>
<td>X L X X</td>
<td>L</td>
</tr>
<tr>
<td>X X L X</td>
<td>L</td>
</tr>
<tr>
<td>X X X L</td>
<td>L</td>
</tr>
</tbody>
</table>

logic diagram (positive logic)

Pin numbers shown are for the D, J, N, NS, PW, and W packages.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

- Supply voltage range, $V_{CC}$: $-0.5$ V to 7 V
- Input clamp current, $I_{IK}$ ($V_I < 0$ or $V_I > V_{CC}$) (see Note 1): $\pm20$ mA
- Output clamp current, $I_{OK}$ ($V_O < 0$ or $V_O > V_{CC}$) (see Note 1): $\pm20$ mA
- Continuous output current, $I_O$ ($V_O = 0$ to $V_{CC}$): $\pm25$ mA
- Continuous current through $V_{CC}$ or GND: $\pm50$ mA
- Package thermal impedance, $\theta_{JA}$ (see Note 2): D package: 86°C/W, N package: 80°C/W, NS package: 76°C/W, PW package: 113°C/W
- Storage temperature range, $T_{stg}$: $-65$°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES:
1. The input and output voltage ratings may be exceeded if the input and output current ratings are observed.
2. The package thermal impedance is calculated in accordance with JESD 51-7.
recommended operating conditions (see Note 3)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CC}$</td>
<td>Supply voltage</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>High-level input voltage</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Low-level input voltage</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{I}$</td>
<td>Input voltage</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{O}$</td>
<td>Output voltage</td>
<td>$V$</td>
</tr>
<tr>
<td>$\Delta t/\Delta v$</td>
<td>Input transition rise/fall time</td>
<td>ns</td>
</tr>
<tr>
<td>$T_A$</td>
<td>Operating free-air temperature</td>
<td>°C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{OH}$</td>
<td>$V_I = V_{IH}$ or $V_{IL}$</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>$V_I = V_{IH}$ or $V_{IL}$</td>
<td>$V$</td>
</tr>
<tr>
<td>$I_I$</td>
<td>$V_I = V_{CC}$ or 0</td>
<td>nA</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>$V_I = V_{CC}$ or 0, $I_O = 0$</td>
<td>µA</td>
</tr>
<tr>
<td>$C_I$</td>
<td>$V_{CC}$ to 6 $V$</td>
<td>pF</td>
</tr>
</tbody>
</table>

**SN54HC21**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CC}$</td>
<td>Supply voltage</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>High-level input voltage</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Low-level input voltage</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{I}$</td>
<td>Input voltage</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{O}$</td>
<td>Output voltage</td>
<td>$V$</td>
</tr>
<tr>
<td>$\Delta t/\Delta v$</td>
<td>Input transition rise/fall time</td>
<td>ns</td>
</tr>
<tr>
<td>$T_A$</td>
<td>Operating free-air temperature</td>
<td>°C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{OH}$</td>
<td>$V_I = V_{IH}$ or $V_{IL}$</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>$V_I = V_{IH}$ or $V_{IL}$</td>
<td>$V$</td>
</tr>
<tr>
<td>$I_I$</td>
<td>$V_I = V_{CC}$ or 0</td>
<td>nA</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>$V_I = V_{CC}$ or 0, $I_O = 0$</td>
<td>µA</td>
</tr>
<tr>
<td>$C_I$</td>
<td>$V_{CC}$ to 6 $V$</td>
<td>pF</td>
</tr>
</tbody>
</table>

**SN74HC21**

**NOTE 3:** All unused inputs of the device must be held at $V_{CC}$ or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.

**electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{OH}$</td>
<td>$V_I = V_{IH}$ or $V_{IL}$</td>
<td>$V$</td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>$V_I = V_{IH}$ or $V_{IL}$</td>
<td>$V$</td>
</tr>
<tr>
<td>$I_I$</td>
<td>$V_I = V_{CC}$ or 0</td>
<td>nA</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>$V_I = V_{CC}$ or 0, $I_O = 0$</td>
<td>µA</td>
</tr>
<tr>
<td>$C_I$</td>
<td>$V_{CC}$ to 6 $V$</td>
<td>pF</td>
</tr>
</tbody>
</table>

**NOTE 3:** All unused inputs of the device must be held at $V_{CC}$ or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.
SN54HC21, SN74HC21
DUAL 4-INPUT POSITIVE-AND GATES

switching characteristics over recommended operating free-air temperature range, \( C_L = 50 \, \text{pF} \) (unless otherwise noted) (see Figure 1)

| PARAMETER | FROM (INPUT) | TO (OUTPUT) | V<sub>CC</sub> | \( T_A = 25^\circ \text{C} \) SN54HC21 | | SN74HC21 | | UNIT |
|------------|--------------|-------------|--------------|-----------------------------------|-------------------|-------------------|-----|
| \( I_{pd} \) | A, B, C, or D | Y | 2 V | 44 | 110 | 165 | 140 | ns |
|             |              |             | 4.5 V | 14 | 22 | 33 | 28 |     |
|             |              |             | 6 V | 11 | 19 | 28 | 24 |     |
| \( I_t \)   |              | Y | 2 V | 29 | 75 | 110 | 95 | ns |
|             |              |             | 4.5 V | 10 | 15 | 22 | 19 |     |
|             |              |             | 6 V | 8 | 13 | 19 | 16 |     |

operating characteristics, \( T_A = 25^\circ \text{C} \)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>TYP</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{pd} )</td>
<td>Power dissipation capacitance per gate</td>
<td>No load</td>
<td>25</td>
</tr>
</tbody>
</table>
PARAMETER MEASUREMENT INFORMATION

NOTES:
A. $C_L$ includes probe and test-fixture capacitance.
B. Phase relationships between waveforms were chosen arbitrarily. All input pulses are supplied by generators having the following characteristics: $PRR \leq 1$ MHz, $Z_O = 50 \, \Omega$, $t_r = 6$ ns, $t_f = 6$ ns.
C. The outputs are measured one at a time with one input transition per measurement.
D. $t_{PLH}$ and $t_{PHL}$ are the same as $t_{pd}$.

Figure 1. Load Circuit and Voltage Waveforms
CERAMIC DUAL IN-LINE PACKAGE

<table>
<thead>
<tr>
<th>DIM</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.300 (7.62) BSC</td>
<td>0.300 (7.62) BSC</td>
<td>0.300 (7.62) BSC</td>
<td>0.300 (7.62) BSC</td>
</tr>
<tr>
<td>B MAX</td>
<td>0.785 (19.94)</td>
<td>0.840 (21.34)</td>
<td>0.960 (24.38)</td>
<td>1.060 (26.92)</td>
</tr>
<tr>
<td>B MIN</td>
<td>——</td>
<td>——</td>
<td>——</td>
<td>——</td>
</tr>
<tr>
<td>C MAX</td>
<td>0.300 (7.62)</td>
<td>0.300 (7.62)</td>
<td>0.310 (7.87)</td>
<td>0.300 (7.62)</td>
</tr>
<tr>
<td>C MIN</td>
<td>0.245 (6.22)</td>
<td>0.245 (6.22)</td>
<td>0.220 (5.59)</td>
<td>0.245 (6.22)</td>
</tr>
</tbody>
</table>

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package is hermetically sealed with a ceramic lid using glass frit.
D. Index point is provided on cap for terminal identification only on press ceramic glass frit seal only.
E. Falls within MIL STD 1835 GDIP1-T14, GDIP1-T16, GDIP1-T18 and GDIP1-T20.
FK (S-CQCC-N**)

LEADLESS CERAMIC CHIP CARRIER

28 TERMINAL SHOWN

<table>
<thead>
<tr>
<th>NO. OF TERMINALS **</th>
<th>A MIN</th>
<th>A MAX</th>
<th>B MIN</th>
<th>B MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.342 (8.69)</td>
<td>0.358 (9.09)</td>
<td>0.307 (7.80)</td>
<td>0.358 (9.09)</td>
</tr>
<tr>
<td>28</td>
<td>0.442 (11.23)</td>
<td>0.458 (11.63)</td>
<td>0.406 (10.31)</td>
<td>0.458 (11.63)</td>
</tr>
<tr>
<td>44</td>
<td>0.640 (16.26)</td>
<td>0.660 (16.76)</td>
<td>0.495 (12.58)</td>
<td>0.560 (14.22)</td>
</tr>
<tr>
<td>52</td>
<td>0.739 (18.78)</td>
<td>0.761 (19.32)</td>
<td>0.495 (12.58)</td>
<td>0.560 (14.22)</td>
</tr>
<tr>
<td>68</td>
<td>0.938 (23.83)</td>
<td>0.962 (24.43)</td>
<td>0.850 (21.6)</td>
<td>0.858 (21.8)</td>
</tr>
<tr>
<td>84</td>
<td>1.141 (28.99)</td>
<td>1.165 (29.59)</td>
<td>1.047 (26.6)</td>
<td>1.063 (27.0)</td>
</tr>
</tbody>
</table>

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. This package can be hermetically sealed with a metal lid.
D. The terminals are gold plated.
E. Falls within JEDEC MS-004
N (R-PDIP-T**)  
16 PINS SHOWN

**NOTES:**  
A. All linear dimensions are in inches (millimeters).  
B. This drawing is subject to change without notice.  
C. Falls within JEDEC MS-001, except 18 and 20 pin minimum body length (Dim A).  
D. The 20 pin end lead shoulder width is a vendor option, either half or full width.
D (R-PDSO-G**)  
PLASTIC SMALL-OUTLINE PACKAGE

8 PINS SHOWN

**NOTES:**
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.006 (0,15).
D. Falls within JEDEC MS-012

<table>
<thead>
<tr>
<th>DIM</th>
<th>PINS **</th>
<th>8</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MAX</td>
<td>0.197 (5.00)</td>
<td>0.344 (8.75)</td>
<td>0.394 (10.00)</td>
</tr>
<tr>
<td>A</td>
<td>MIN</td>
<td>0.189 (4.80)</td>
<td>0.337 (8.55)</td>
<td>0.386 (9.80)</td>
</tr>
</tbody>
</table>
MECHANICAL DATA

NS (R-PDSO-G**)
14-PINS SHOWN

PLASTIC SMALL-OUTLINE PACKAGE

NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.15.
NOTES:  
A. All linear dimensions are in millimeters.  
B. This drawing is subject to change without notice.  
C. Body dimensions do not include mold flash or protrusion not to exceed 0.15.  
D. Falls within JEDEC MO-153
IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifiers</td>
<td>Audio</td>
</tr>
<tr>
<td>Data Converters</td>
<td>Automotive</td>
</tr>
<tr>
<td>DSP</td>
<td>Broadband</td>
</tr>
<tr>
<td>Interface</td>
<td>Digital Control</td>
</tr>
<tr>
<td>Logic</td>
<td>Military</td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>Optical Networking</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td>Telephony</td>
</tr>
<tr>
<td></td>
<td>Video &amp; Imaging</td>
</tr>
<tr>
<td></td>
<td>Wireless</td>
</tr>
</tbody>
</table>

Mailing Address: Texas Instruments  
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2003, Texas Instruments Incorporated
8. 564-0700-111F: 3mm LED CBI Tri-Level Circuit Board Indicator

**Dialight**

3mm LED CBI Tri-Level Circuit Board Indicator

564-0700-111F
3mm
LED CBI® Circuit Board Indicator
Tri-Level

PART NO. | COLOR*
---|---
HIGH EFFICIENCY - LED TYPE 01
564-0100-111 | Red-Red-Red
564-0100-132 | Red-Yellow-Green
564-0100-222 | Green-Green-Green
564-0100-777 | Orange-Orange-Orange
564-0100-999 | Blue-Blue-Blue

LOW CURRENT - LED TYPE 02
564-0200-111 | Red-Red-Red
564-0200-132 | Red-Yellow-Green
564-0200-222 | Green-Green-Green

INTEGRAL RESISTOR, 5 Volts - LED TYPE 03
564-0300-111 | Red-Red-Red
564-0300-132 | Red-Yellow-Green
564-0300-222 | Green-Green-Green

BI-COLOR - LED TYPE 07
564-0700-111 | Red/Green-Red/Green-Red/Green
564-0700-444 | Yellow/Green-Yellow/Green-Yellow/Green

* Top-Middle-Bottom LED

Features

- Multiple CBIs form horizontal LED arrays on 4.45mm (0.175") center-lines.
- High Contrast, UL 94 V-0 rated, black housing
- Oxygen index: 29%
- Polymer content: PBT, 0.078 g
- Housing stand-offs facilitate PCB cleaning
- Solderability per MIL-STD-202F, method 208F
- LEDs are safe for direct viewing per IEC 825-1, EN-60825-1

Tolerance note: As noted, otherwise:

- LED Protrusion: ±0.04 mm [±0.016]
- CBI Housing: ±0.02mm[±0.008]

Custom Combinations

- Contact factory for information on custom color combinations

**ATTENTION**
Observe Precautions for Handling Electrostatic Sensitive Devices

Dialight Corporation • 1501 Route 34 South • Farmingdale, NJ 07727 • TEL: (732) 919-3119 • FAX: (732) 751-5778 •www.dialight.com
9. HEF4521B: 24-Stage Frequency Divider and Oscillator

Philips Semiconductor
24-Stage Frequency Divider and Oscillator
HEF4521B
DATA SHEET

For a complete data sheet, please also download:

- The IC04 LOCMOS HE4000B Logic Family Specifications HEF, HEC
- The IC04 LOCMOS HE4000B Logic Package Outlines/Information HEF, HEC

HEF4521B
MSI
24-stage frequency divider and oscillator

Product specification January 1995
File under Integrated Circuits, IC04
DESCRIPTION

The HEF4521B consists of a chain of 24 toggle flip-flops with an overriding asynchronous master reset input (MR), and an input circuit that allows three modes of operation. The single inverting stage (I₂/O₂) will function as a crystal oscillator, or in combination with I₁ as an RC oscillator, or as an input buffer for an external oscillator. Low-power operation as a crystal oscillator is enabled by connecting external resistors to pins 3 (VSS') and 5 (VDD'). Each flip-flop divides the frequency of the previous flip-flop by two, consequently the HEF4521B will count up to \(2^{24} = 16777216\). The counting advances on the HIGH to LOW transition of the clock (I₂). The outputs of the last seven stages are available for additional flexibility.

![Functional diagram](image)

Fig.1 Functional diagram.

FAMILY DATA, \(I_{DD}\) LIMITS category MSI

See Family Specifications
24-stage frequency divider and oscillator

**HEF4521B**  
**MSI**

![Pinning diagram](image)

**COUNT CAPACITY**

<table>
<thead>
<tr>
<th>OUTPUT</th>
<th>COUNT CAPACITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>O₁₈</td>
<td>2¹⁸ = 262 144</td>
</tr>
<tr>
<td>O₁₉</td>
<td>2¹⁹ = 524 288</td>
</tr>
<tr>
<td>O₂₀</td>
<td>2²⁰ = 1 048 576</td>
</tr>
<tr>
<td>O₂₁</td>
<td>2²¹ = 2 097 152</td>
</tr>
<tr>
<td>O₂₂</td>
<td>2²² = 4 194 304</td>
</tr>
<tr>
<td>O₂₃</td>
<td>2²³ = 8 388 608</td>
</tr>
<tr>
<td>O₂₄</td>
<td>2²⁴ = 16 777 216</td>
</tr>
</tbody>
</table>

HEF4521BP(N): 16-lead DIL; plastic (SOT38-1)  
HEF4521BD(F): 16-lead DIL; ceramic (cerdip) (SOT74)  
HEF4521BT(D): 16-lead SO; plastic (SOT109-1)  
( ) : Package Designator North America

**FUNCTIONAL TEST SEQUENCE**

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>CONTROL TERMINALS</th>
<th>OUTPUTS</th>
<th>REMARKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR</td>
<td>I₂</td>
<td>O₂</td>
<td>Vₛₛ’</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>L</td>
<td>V_dd</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>V_dd</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>V_ss</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>L</td>
<td>V_ss</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>L</td>
<td>V_ss</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td></td>
<td>V_ss</td>
</tr>
</tbody>
</table>

A test function has been included for the reduction of the test time required to exercise all 24 counter stages. This test function divides the counter into three 8-stage sections by connecting V₁₈’ to V_dd and V_dd’ to V_ss. Via I₂ (connected to O₂) 255 counts are loaded into each of the 8-stage sections in parallel. All flip-flops are now at a HIGH state.

The counter is now returned to the normal 24-stage in series configuration by connecting V₁₈’ to V_ss and V_dd’ to V_dd. One more pulse is entered into input I₂, which will cause the counter to ripple from an all HIGH state to an all LOW state.

January 1995
24-stage frequency divider and oscillator

Fig. 3. Logic diagram; for schematic diagram of clock circuit see Fig. 4.
24-stage frequency divider and oscillator

AC CHARACTERISTICS

$V_{SS} = 0 \text{ V; } T_{amb} = 25 \text{ °C; } C_L = 50 \text{ pF; input transition times } \leq 20 \text{ ns}$

<table>
<thead>
<tr>
<th>$V_{DD}$</th>
<th>SYMBOL</th>
<th>MIN.</th>
<th>TYP.</th>
<th>MAX.</th>
<th>TYPICAL EXTRAPOLATION FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>$t_{PHL}$</td>
<td>950</td>
<td>1900</td>
<td>ns</td>
<td>$923 \text{ ns } + (0.55 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>350</td>
<td>700</td>
<td>ns</td>
<td>$339 \text{ ns } + (0.23 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>220</td>
<td>440</td>
<td>ns</td>
<td>$212 \text{ ns } + (0.16 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>5</td>
<td>$t_{PLH}$</td>
<td>950</td>
<td>1900</td>
<td>ns</td>
<td>$923 \text{ ns } + (0.55 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>350</td>
<td>700</td>
<td>ns</td>
<td>$339 \text{ ns } + (0.23 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>220</td>
<td>440</td>
<td>ns</td>
<td>$212 \text{ ns } + (0.16 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>5</td>
<td>$t_{PHL}$</td>
<td>40</td>
<td>80</td>
<td>ns</td>
<td>$13 \text{ ns } + (0.55 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>15</td>
<td>30</td>
<td>ns</td>
<td>$4 \text{ ns } + (0.23 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>10</td>
<td>20</td>
<td>ns</td>
<td>$2 \text{ ns } + (0.16 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>5</td>
<td>$t_{PLH}$</td>
<td>40</td>
<td>80</td>
<td>ns</td>
<td>$13 \text{ ns } + (0.55 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>15</td>
<td>30</td>
<td>ns</td>
<td>$4 \text{ ns } + (0.23 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>10</td>
<td>20</td>
<td>ns</td>
<td>$2 \text{ ns } + (0.16 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>5</td>
<td>$t_{PHL}$</td>
<td>120</td>
<td>240</td>
<td>ns</td>
<td>$93 \text{ ns } + (0.55 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>55</td>
<td>110</td>
<td>ns</td>
<td>$44 \text{ ns } + (0.23 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>40</td>
<td>80</td>
<td>ns</td>
<td>$32 \text{ ns } + (0.16 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>5</td>
<td>$t_{PLH}$</td>
<td>90</td>
<td>180</td>
<td>ns</td>
<td>$63 \text{ ns } + (0.55 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>35</td>
<td>70</td>
<td>ns</td>
<td>$24 \text{ ns } + (0.23 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>25</td>
<td>50</td>
<td>ns</td>
<td>$17 \text{ ns } + (0.16 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>5</td>
<td>$t_{PHL}$</td>
<td>60</td>
<td>120</td>
<td>ns</td>
<td>$33 \text{ ns } + (0.55 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>30</td>
<td>60</td>
<td>ns</td>
<td>$19 \text{ ns } + (0.23 \text{ ns/pF}) \times C_L$</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>20</td>
<td>40</td>
<td>ns</td>
<td>$12 \text{ ns } + (0.16 \text{ ns/pF}) \times C_L$</td>
</tr>
</tbody>
</table>
# 24-stage frequency divider and oscillator

## AC CHARACTERISTICS

\( V_{SS} = 0 \text{ V; } T_{\text{amb}} = 25 \pm {\circ}\text{C; } C_{L} = 50 \text{ pF; input transition times } \leq 20 \text{ ns} \)

<table>
<thead>
<tr>
<th>( V_{DD} ) V</th>
<th>SYMBOL</th>
<th>MIN.</th>
<th>TYP.</th>
<th>MAX.</th>
<th>TYPICAL EXTRAPOLATION FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output transition times</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIGH to LOW</td>
<td>5</td>
<td>60</td>
<td>120</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>30</td>
<td>60</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>20</td>
<td>40</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( 10 \text{ ns} + (1.0 \text{ ns/pF}) C_{L} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( 9 \text{ ns} + (0.42 \text{ ns/pF}) C_{L} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( 6 \text{ ns} + (0.28 \text{ ns/pF}) C_{L} )</td>
</tr>
<tr>
<td>LOW to HIGH</td>
<td>5</td>
<td>60</td>
<td>120</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>30</td>
<td>60</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>20</td>
<td>40</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( 10 \text{ ns} + (1.0 \text{ ns/pF}) C_{L} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( 9 \text{ ns} + (0.42 \text{ ns/pF}) C_{L} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( 6 \text{ ns} + (0.28 \text{ ns/pF}) C_{L} )</td>
</tr>
</tbody>
</table>

## Dynamic power dissipation per package (P)

<table>
<thead>
<tr>
<th>( V_{DD} ) V</th>
<th>TYPICAL FORMULA FOR P (( \mu \text{W} ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic power</td>
<td></td>
</tr>
<tr>
<td>dissipation per</td>
<td></td>
</tr>
<tr>
<td>package (P)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>( 1 200 f_{i} + \Sigma (f_{o} C_{L}) \times V_{DD}^{2} )</td>
</tr>
<tr>
<td>10</td>
<td>( 5 100 f_{i} + \Sigma (f_{o} C_{L}) \times V_{DD}^{2} )</td>
</tr>
<tr>
<td>15</td>
<td>( 13 050 f_{i} + \Sigma (f_{o} C_{L}) \times V_{DD}^{2} )</td>
</tr>
</tbody>
</table>

where
- \( f_{i} \) = input freq. (MHz)
- \( f_{o} \) = output freq. (MHz)
- \( C_{L} \) = load capacitance (pF)
- \( \Sigma (f_{o} C_{L}) \) = sum of outputs
- \( V_{DD} \) = supply voltage (V)

January 1995
Fig. 5 Waveforms showing minimum pulse widths for MR and I₂, recovery time for MR.
**APPLICATION INFORMATION**

![Diagram](image)

(1) Optional for low power operation.

Fig.6 Crystal oscillator circuit.

**Typical characteristics for crystal oscillator circuit (Fig.6):**

<table>
<thead>
<tr>
<th></th>
<th>500 kHz CIRCUIT</th>
<th>50 kHz CIRCUIT</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Crystal characteristics</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>resonance frequency</td>
<td>500</td>
<td>50</td>
<td>kHz</td>
</tr>
<tr>
<td>crystal cut</td>
<td>S</td>
<td>N</td>
<td>–</td>
</tr>
<tr>
<td>equivalent resistance; $R_S$</td>
<td>1</td>
<td>6.2</td>
<td>kΩ</td>
</tr>
<tr>
<td><strong>External resistor/capacitor values</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R_o$</td>
<td>47</td>
<td>750</td>
<td>kΩ</td>
</tr>
<tr>
<td>$C_T$</td>
<td>82</td>
<td>82</td>
<td>pF</td>
</tr>
<tr>
<td>$C_S$</td>
<td>20</td>
<td>20</td>
<td>pF</td>
</tr>
</tbody>
</table>

January 1995
24-stage frequency divider and oscillator

HEF4521B
MSI

Fig. 7  RC oscillator circuit;

\[
f = \frac{1}{2 \times 3 \times R_{TC} \times C}; \quad R_S \geq 2 \times R_{TC}, \text{ in which:}
\]

\[
f \text{ in Hz, } R \text{ in } \Omega, \ C \text{ in F.}
\]

\[
R_S + R_{TC} < \frac{V_{IL \max}}{I_{LI}} \quad (\text{maximum input voltage LOW})
\]

\[
I_{LI} \quad (\text{input leakage current})
\]

Fig. 8  Oscillator frequency as a function of \(R_{TC}\) and \(C\);
\(V_{DD} = 10 \text{ V};\) test circuit is Fig. 7.
24-stage frequency divider and oscillator

**Fig. 9** Test set-up for measuring forward transconductance $g_{fs} = \frac{di_o}{dv_i}$ at $v_o$ is constant (see also graph Fig. 10).

**Fig. 10** Typical forward transconductance $g_{fs}$ as a function of the supply voltage at $T_{amb} = 25 \, ^\circ C$.

A: average,
B: average $+ 2 \, \sigma$,
C: average $- 2 \, \sigma$, in which: `$\sigma$' is the observed standard deviation.
24-stage frequency divider and oscillator

**Fig. 11** Voltage gain $V_O/V_I$ as a function of supply voltage.

**Fig. 12** Supply current as a function of supply voltage.

**Fig. 13** Test set-up for measuring graphs of Figs 11 and 12.
10. 179-009-513R571: Center Dual Port D-SUB Female/Female

NorComp
Center Dual Port D-SUB
Female/Female
179-009-513R571
SPECIFICATIONS:
MATERIALS:
SHELL: STEEL, NICKEL PLATED
INSULATOR: NYLON - UL 94V-0 RATED BLACK
CONTACTS: BRASS - GOLD FLASH IN MATING AREA TIN ON SOLDER TAILS
RoHS COMPLIANT

ELECTRICALS:
CURRENT RATING: 5 AMPS
DIELECTRIC STRENGTH: 1000V AC RMS
INSULATOR RESISTANCE: 1000 MOHMS MIN
TEMPERATURE: -55°C TO +105°C

NorComp
11. LTC1441/SO: Ultra Low Power Dual Comparator with Reference

Linear Technology
Ultra Low Power Dual Comparator with Reference
LTC1441/SO
The LTC®1440/LTC1441/LTC1442 are ultralow power single and dual comparators with built-in references. The comparators feature less than 3.7µA supply current over temperature (LTC1440), a 1.182V ±1% reference, programmable hysteresis (LTC1440/LTC1442) and TTL/CMOS outputs that sink and source current. The reference output can drive a bypass capacitor of up to 0.01µF without oscillation.

The comparators operate from a single 2V to 11V supply or a dual ±1V to ±5.5V supply (LTC1440). Comparator hysteresis is easily programmed by using two resistors and the HYST pin (LTC1440/LTC1442). Each comparator’s input operates from the negative supply to within 1.3V of the positive supply. The comparator output stage can continuously source up to 40mA. By eliminating the cross-conducting current that normally happens when the comparator changes logic states, the power supply glitches are eliminated.

The LTC1440 is available in 8-pin PDIP, SO, MSOP and DFN packages. The LTC1441/LTC1442 are available in 8-pin PDIP and SO packages.

**FEATURES**
- Ultralow Quiescent Current: 2.1µA Typ (LTC1440)
- Reference Output Drives 0.01µF Capacitor
- Adjustable Hysteresis (LTC1440/LTC1442)
- Wide Supply Range:
  - Single: 2V to 11V
  - Dual: ±1V to ±5.5V
- Input Voltage Range Includes the Negative Supply
- TTL/CMOS Compatible Outputs
- 12µs Propagation Delay with 10mV Overdrive
- No Crowbar Current
- 40mA Continuous Source Current
- Pin Compatible Upgrades for MAX921/922/923
- 3mm x 3mm x 0.75mm DFN Package (LTC1440)

**APPLICATIONS**
- Battery-Powered System Monitoring
- Threshold Detectors
- Window Comparators
- Oscillator Circuits

**DESCRIPTION**

**TYPICAL APPLICATION**

**Micropower 2.9V V\text{CC} Threshold Detector**

**LTC1440 Supply Current vs Temperature**
**ABSOLUTE MAXIMUM RATINGS**

(Note 1)

Voltage

- $V^+ \text{ to } V^-$, $V^+ \text{ to } GND$, $GND \text{ to } V^-$ ........... 12V to –0.3V
- $IN^+, \ IN^-, \ \text{HYST}$ ........... $(V^+ + 0.3V) \text{ to } (V^- – 0.3V)$
- REF ................. $(V^+ + 0.3V) \text{ to } (V^- – 0.3V)$
- OUT (LTC1440) .............. $(V^+ + 0.3V) \text{ to } (GND – 0.3V)$
- OUT (LTC1441/LTC1442) ... $(V^+ + 0.3V) \text{ to } (V^- – 0.3V)$

Current

- $IN^+, \ IN^-, \ \text{HYST}$ ................................................. 20mA
- REF ................................................................... 20mA
- OUT .................................................................. 50mA

OUT Short-Circuit Duration ($V^+ \leq 5.5V$) ........... Continuous

Power Dissipation .......................................................... 500mW

Operating Temperature Range

- LTC144XC ............................................... 0°C to 70°C
- LTC144XI ........................................... –40°C to 85°C

Storage Temperature Range ................. –65°C to 150°C

- (DD Package) ................................... –65°C to 125°C

Junction Temperature ......................... 150°C

- Junction Temperature (DD Package) ........... 125°C

Lead Temperature (Soldering, 10 sec) ........... 300°C

**PACKAGE/ORDER INFORMATION**

Consult LTC Marketing for parts specified with wider operating temperature ranges.

* The temperature grade is identified by a label on the shipping container.

Order Options

- Tape and Reel: Add #TR
- Lead Free: Add #PBF
- Lead Free Tape and Reel: Add #TRPBF

**ORDER PART NUMBER**

- LTC1440CDD
- LTC1440IDD

**DD8 PART MARKING*”

- LBTH

**ORDER PART NUMBER**

- LTC1440CN8
- LTC1440CS8
- LTC1440IN8
- LTC1440IS8

**S8 PART MARKING**

- 1440
- 1440I

**ORDER PART NUMBER**

- LTC1440CMS8
- LTC1440IMS8

**MS8 PART MARKING*”

- LTBX

**ORDER PART NUMBER**

- LTC1441CN8
- LTC1441CS8
- LTC1441IN8
- LTC1441IS8

**S8 PART MARKING**

- 1441
- 1441I

**ORDER PART NUMBER**

- LTC1442CN8
- LTC1442CS8
- LTC1442IN8
- LTC1442IS8

**S8 PART MARKING**

- 1442
- 1442I
## ELECTRICAL CHARACTERISTICS

The ● denotes the specifications which apply over the full operating temperature range, otherwise specifications are at $T_A = 25^\circ C$. $V^+ = 5V$ and $V^- = GND = 0V$ unless otherwise noted.

### SYMBOL | PARAMETER | CONDITIONS | MIN | TYP | MAX | UNITS
--- | --- | --- | --- | --- | --- | ---
**Power Supply**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V^+$</td>
<td>Supply Voltage Range</td>
<td>$V^+_i = V^- + 80mV$ [HYST = \text{REF} \ (\text{LTC1440/LTC1442})]</td>
<td>●</td>
<td>2.0</td>
<td>11.0</td>
<td>V</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>Supply Current</td>
<td>$I_{IN}^+ = I_{IN}^- + 80mV$ [HYST = \text{REF} \ (\text{LTC1440/LTC1442})]</td>
<td>●</td>
<td>2.1</td>
<td>4.0</td>
<td>$\mu A$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{LTC1441}$</td>
<td>●</td>
<td>3.5</td>
<td>5.7</td>
<td>$\mu A$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{LTC1442}$</td>
<td>●</td>
<td>3.5</td>
<td>5.7</td>
<td>$\mu A$</td>
</tr>
</tbody>
</table>

**Comparator**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Conditions</th>
<th>Min</th>
<th>TYP</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{\text{CM}}$</td>
<td>Comparator Input Common Mode Range</td>
<td>$V^+ - V^- - 1.3V$</td>
<td>●</td>
<td>V$^-$</td>
<td>V$^+ - 1.3V$</td>
<td>V</td>
</tr>
<tr>
<td>$V_{\text{IN}}$</td>
<td>Input Leakage Current ($I_{IN}^+; I_{IN}^-$)</td>
<td>$V_{IN}^+ = V_{IN}^- = 2.5V$</td>
<td>●</td>
<td>$\pm 0.01$</td>
<td>$\pm 1.0$</td>
<td>nA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{CMR}$</td>
<td>●</td>
<td>$\pm 0.02$</td>
<td>$\pm 1.0$</td>
<td>nA</td>
</tr>
</tbody>
</table>

**Noise**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Conditions</th>
<th>Min</th>
<th>TYP</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{\text{NOISE}}$</td>
<td>Voltage Noise</td>
<td>100Hz to 100kHz</td>
<td>100</td>
<td>$\mu V_{\text{RMS}}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Reference**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Conditions</th>
<th>Min</th>
<th>TYP</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{\text{REF}}$</td>
<td>Reference Output Source Current $\Delta V_{\text{REF}} \leq 1mV$ [\text{LTC1442}]</td>
<td>●</td>
<td>100</td>
<td>$\mu A$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\Delta V_{\text{REF}} \leq 2.5mV$ [\text{LTC1442}]</td>
<td>10</td>
<td>20</td>
<td>$\mu A$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\Delta V_{\text{REF}} \leq 10\mu A$ [\text{LTC1440}]</td>
<td>0.5</td>
<td>1.5</td>
<td>mV</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\Delta V_{\text{REF}} \leq 10\mu A$ [\text{LTC1440}]</td>
<td>0.5</td>
<td>1.5</td>
<td>mV</td>
<td></td>
</tr>
</tbody>
</table>
## ELECTRICAL CHARACTERISTICS

The ● denotes the specifications which apply over the full operating temperature range, otherwise specifications are at $T_A = 25^\circ C$. $V^+ = 3V$ and $V^- = GND = 0V$ unless otherwise noted.

### Power Supply

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>CONDITIONS</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V^+$</td>
<td>Supply Voltage Range</td>
<td>●</td>
<td>2</td>
<td>11</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>Supply Current</td>
<td>$IN^+ = IN^- + 80mV$</td>
<td>●</td>
<td>2</td>
<td>3.9</td>
<td>$\mu A$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$HYST = REF$ (LTC1440/LTC1442)</td>
<td></td>
<td></td>
<td>4.3</td>
<td>$\mu A$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$LTC1440$ $0^\circ C \leq T_A \leq 70^\circ C$</td>
<td>●</td>
<td>3.5</td>
<td>5.7</td>
<td>$\mu A$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-40^\circ C \leq T_A \leq 85^\circ C$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$LTC1441$</td>
<td>●</td>
<td>3.5</td>
<td>5.7</td>
<td>$\mu A$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$LTC1442$</td>
<td>●</td>
<td>3.5</td>
<td>5.7</td>
<td>$\mu A$</td>
</tr>
</tbody>
</table>

### Comparator

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>CONDITIONS</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DS}$</td>
<td>Comparator Input Offset Voltage</td>
<td>$V_{CM} = 1.5V$</td>
<td>●</td>
<td>±3</td>
<td>±10</td>
<td>mV</td>
</tr>
<tr>
<td>$I_{IN}$</td>
<td>Input Leakage Current ($IN^+, IN^-$) Input Leakage Current (HYST)</td>
<td>$V_{IN}^+ = V_{IN}^- = 1.5V$</td>
<td>●</td>
<td>±0.01</td>
<td>±1</td>
<td>nA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{CM}$</td>
<td>●</td>
<td>±0.02</td>
<td>±1</td>
<td>nA</td>
</tr>
<tr>
<td>$V_{CM}$</td>
<td>Comparator Input Common Mode Range</td>
<td>$V^-$ to $V^+ - 1.3V$</td>
<td>●</td>
<td>0.1</td>
<td>1</td>
<td>mV/V</td>
</tr>
<tr>
<td>CMRR</td>
<td>Common Mode Rejection Ratio</td>
<td>$V^+ = 2V$ to $11V$ (LTC1441)</td>
<td>●</td>
<td>0.1</td>
<td>1</td>
<td>mV/V</td>
</tr>
<tr>
<td>PSRR</td>
<td>Power Supply Rejection Ratio</td>
<td>$V^+ = 2.5V$ to $11V$ (LTC1440/LTC1442)</td>
<td>●</td>
<td>0.1</td>
<td>1</td>
<td>mV/V</td>
</tr>
<tr>
<td>NOISE</td>
<td>Voltage Noise</td>
<td>100Hz to 100kHz</td>
<td>●</td>
<td>100</td>
<td></td>
<td>$\mu V_{RMS}$</td>
</tr>
<tr>
<td>$V_{HYST}$</td>
<td>Hysteresis Input Voltage Range</td>
<td>$LTC1440/LTC1442$</td>
<td>●</td>
<td>REF – 50mV</td>
<td>REF</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_O = 8mA$</td>
<td>●</td>
<td>$V^+ \leq 0.4V$</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>Output Low Voltage</td>
<td>$I_O = 0.8mA$</td>
<td>●</td>
<td>$GND \leq 0.4V$</td>
<td>$V^- \leq 0.4V$</td>
<td>V</td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>Output High Voltage</td>
<td>$I_O = 0.8mA$</td>
<td>●</td>
<td>$V^+ \leq 0.4V$</td>
<td></td>
<td>V</td>
</tr>
</tbody>
</table>

### Reference

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>CONDITIONS</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{REF}$</td>
<td>Reference Voltage</td>
<td>No Load</td>
<td>●</td>
<td>1.170</td>
<td>1.182</td>
<td>1.194</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$LTC1440/LTC1442$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0^\circ C \leq T_A \leq 70^\circ C$</td>
<td>●</td>
<td>1.164</td>
<td>1.200</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-40^\circ C \leq T_A \leq 85^\circ C$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$LTC1440$ (MSOP, DFN)</td>
<td>●</td>
<td>1.164</td>
<td>1.200</td>
<td>V</td>
</tr>
<tr>
<td>$I_{SOURCE}$</td>
<td>Reference Output Source Current</td>
<td>$\Delta V_{REF} \leq 1mV$ (LTC1442)</td>
<td>●</td>
<td>60</td>
<td>120</td>
<td>$\mu A$</td>
</tr>
<tr>
<td>$I_{SINK}$</td>
<td>Reference Output Sink Current</td>
<td>$\Delta V_{REF} \leq 2.5mV$ (LTC1442)</td>
<td>●</td>
<td>10</td>
<td>20</td>
<td>$\mu A$</td>
</tr>
<tr>
<td>$\Delta V_{REF}$</td>
<td>Reference Source Current</td>
<td>$0 \leq I_{SOURCE} \leq 1mA$ (LTC1440)</td>
<td>●</td>
<td>0.8</td>
<td>5.5</td>
<td>mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0 \leq I_{SINK} \leq 10\mu A$ (LTC1440)</td>
<td>●</td>
<td>0.5</td>
<td>1.5</td>
<td>mV</td>
</tr>
<tr>
<td>NOISE</td>
<td>Voltage Noise</td>
<td>100Hz to 100kHz</td>
<td>●</td>
<td>100</td>
<td></td>
<td>$HV_{RMS}$</td>
</tr>
</tbody>
</table>

**Note 1:** Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. Exposure to any Absolute Maximum Rating condition for extended periods may affect device reliability and lifetime.
TYPICAL PERFORMANCE CHARACTERISTICS

Comparator Response Time vs Input Overdrive

Comparator Short-Circuit Sink Current vs Supply Voltage

Comparator Short-Circuit Source Current vs Supply Voltage
Comparator Response Time vs Load Capacitance with 100mV Input Overdrive

Comparator Response Time at Low Supply Voltage

Comparator Output Voltage High vs Load Current
TYPICAL PERFORMANCE CHARACTERISTICS

Comparator Output Voltage Low vs Load Current

LTC1440/LTC1442 Hysteresis Control

LTC1440 Supply Current vs Temperature
TYPICAL PERFORMANCE CHARACTERISTICS

Reference Output Voltage vs Output Load Current

Reference Output Voltage vs Output Load Current (Sink)

Reference Voltage vs Temperature
**LTC1440**

**GND (Pin 1):** Ground. Connect to V− for single supply operation.

**V− (Pin 2):** Negative Supply. Connect to ground for single supply operation. Potential should be more negative than GND.

**IN+ (Pin 3):** Noninverting Comparator Input. Input common mode range from V− to V+ −1.3V. Input current typically 10pA at 25°C.

**IN− (Pin 4):** Inverting Comparator Input. Input common mode range from V− to V+ −1.3V. Input current typically 10pA at 25°C.

**HYST (Pin 5):** Hysteresis Input. Connect to REF if not used. Input voltage range is from VREF to VREF − 50mV.

**REF (Pin 6):** Reference Output. 1.182V with respect to V−. Can source up to 200µA and sink 15µA at 25°C. Drive 0.01µF bypass capacitor without oscillation.

**V+ (Pin 7):** Positive Supply. 2V to 11V.

**OUT (Pin 8):** Comparator CMOS Output. Swings from GND to V+. Output can source up to 40mA and sink 5mA.

**LTC1441**

**OUT A (Pin 1):** Comparator A CMOS Output. Swings from V− to V+. Output can source up to 40mA and sink 5mA.

**V− (Pin 2):** Negative Supply.

**IN A+ (Pin 3):** Noninverting Input of Comparator A. Input common mode range from V− to V+ −1.3V. Input current typically 10pA at 25°C.

**IN A− (Pin 4):** Inverting Input of Comparator A. Input common mode range from V− to V+ −1.3V. Input current typically 10pA at 25°C.

**IN B− (Pin 5):** Inverting Input of Comparator B. Input common mode range from V− to V+ −1.3V. Input current typically 10pA at 25°C.

**IN B+ (Pin 6):** Noninverting Input of Comparator B. Input common mode range from V− to V+ −1.3V. Input current typically 10pA at 25°C.

**V+ (Pin 7):** Positive Supply. 2V to 11V.

**OUT B (Pin 8):** Comparator B CMOS Output. Swings from V− to V+. Output can source up to 40mA and sink 5mA.

**LTC1442**

**OUT A (Pin 1):** Comparator A CMOS Output. Swings from V− to V+. Output can source up to 40mA and sink 5mA.

**V− (Pin 2):** Negative Supply.

**IN A+ (Pin 3):** Noninverting Input of Comparator A. Input common mode range from V− to V+ −1.3V. Input current typically 10pA at 25°C.

**IN B− (Pin 5):** Inverting Input of Comparator B. Input common mode range from V− to V+ −1.3V. Input current typically 10pA at 25°C.

**REF (Pin 6):** Reference Output. 1.182V with respect to V−. Can source up to 200µA and sink 15µA at 25°C. Drive 0.01µF bypass capacitor without oscillation.

**V+ (Pin 7):** Positive Supply. 2V to 11V.

**OUT B (Pin 8):** Comparator B CMOS Output. Swings from V− to V+. Output can source up to 40mA and sink 5mA.
LTC1440/LTC1441/LTC1442 are a family of micropower comparators with built-in 1.182V reference. Features include programmable hysteresis (LTC1440/LTC1442), wide supply voltage range (2V to 11V) and the ability of the reference to drive up to a 0.01μF capacitor without oscillation. The comparators’ CMOS outputs can source up to 40mA and the supply current glitches, that normally occur when switching logic states, have been eliminated.

Power Supplies

The comparator family operates from a single 2V to 11V supply. The LTC1440 includes a separate ground for the comparator output stage, allowing a split supply ranging from ±1V to ±5.5V. Connecting V– to GND on the LTC1440 will allow single supply operation. If the comparator output is required to source more than 1mA, or the supply source impedance is high, V+ should be bypassed with a 0.1μF capacitor.

Comparator Inputs

The comparator inputs can swing from the negative supply V– to within 1.3V max of the positive supply V+. The inputs can be forced 300mV below V– or above V+ without damage and the typical input leakage current is only ±10pA.

Comparator Outputs

The LTC1440 comparator output swings between GND and V+ to assure TTL compatibility with a split supply. The LTC1441 and LTC1442 outputs swing between V– and V+. The outputs are capable of sourcing up to 40mA and sinking up to 5mA while still maintaining microampere quiescent currents. The output stage does not generate crowbar switching currents during transitions which helps minimize parasitic feedback through the supply pins.

Voltage Reference

The internal bandgap reference has a voltage of 1.182V referenced to V–. The reference accuracy is 1.5% from −40°C to 85°C. It can source up to 200μA and sink up to 20μA with a 5V supply. The reference can drive a bypass capacitor of up to 0.01μF without oscillation and by inserting a series resistor, capacitance values up to 100μF can be used (Figure 1).

Figure 2 shows the resistor value required for different capacitor values to achieve critical damping. Bypassing the reference can help prevent false tripping of the comparators by preventing glitches on V+ or reference load transients from disturbing the reference output voltage.

Figure 3 shows the bypassed reference output with a square wave applied to the V+ pin. Resistors R2 and R3 set 10mV of hysteresis voltage band while R1 damps the reference response. Note that the comparator output doesn’t trip.
APPLICATIONS INFORMATION

Figure 3a. Reference Transient Response Test Circuit

Figure 3b. Reference and Comparator Output Transient Response

Hysteresis

Hysteresis can be added to the LTC1440 by connecting a resistor (R1) between the REF and HYST pins and a second resistor (R2) from HYST to V– (Figure 4).

The difference between the upper and lower threshold voltages, or hysteresis voltage band (V_{HB}), is equal to twice the voltage difference between the REF and HYST pins.

When more hysteresis is added, the upper threshold increases the same amount as the low threshold decreases. The maximum voltage allowed between REF and HYST pins is 50mV, producing a maximum hysteresis voltage band of 100mV. The hysteresis band could vary by up to 15%. If hysteresis is not wanted, the HYST pin should be shorted to REF. Acceptable values for I_{REF} range from 0.1\mu A to 5\mu A. If 2.4M is chosen for R2, then the value of R1 is equal to the value of V_{HB}.

Figure 4. Programmable Hysteresis

Level Detector

The LTC1440 is ideal for use as a micropower level detector as shown in Figure 5. R1 and R2 form a voltage divider from V_{IN} to the noninverting comparator input. R3 and R4 set the hysteresis voltage, and R5 and C1 bypass the reference output. The following design procedure can be used to select the component values:

1. Choose the V_{IN} voltage trip level, in this example 4.65V.
2. Calculate the required resistive divider ratio.
   \[ \text{Ratio} = \frac{V_{\text{REF}}}{V_{\text{IN}}} \]
   \[ \text{Ratio} = \frac{1.182V}{4.65V} = 0.254 \]

3. Choose the required hysteresis voltage band at the input \( V_{\text{HBIN}} \), in this example 60mV. Calculate the hysteresis voltage band referred to the comparator input \( V_{\text{HB}} \).
   \[ V_{\text{HB}} = (V_{\text{HBIN}}) \times (\text{Ratio}) \]
   \[ V_{\text{HB}} = (60\text{mV}) \times (0.254) \]
   \[ V_{\text{HB}} = 15.24\text{mV} \]

4. Choose the values for \( R3 \) and \( R4 \) to set the hysteresis.
   \[ R4 = 2.4\text{M} \]
   \[ R3(\text{k}\Omega) = V_{\text{HB}} = 15\text{k} \]

5. Choose the values for \( R1 \) and \( R2 \) to set the trip point.
   \[ R1 = \frac{V_{\text{REF}}}{I_{\text{BIAS}}} = \frac{1.182V}{1\mu\text{A}} = 1.18\text{M} \]
   \[ R2 = R1 \left[ \frac{V_{\text{IN}}}{V_{\text{REF}} + \frac{V_{\text{HB}}}{2}} - 1 \right] \]

   \[ R2 = 1.18\text{M} \left[ \frac{4.65V}{1.182V + \frac{15\text{mV}}{2}} - 1 \right] \]
   \[ R2 = 3.40\text{M} \]

**Low Voltage Operation**

It is important to note that the voltage references internal to the LTC1440 and LTC1442 can exceed the common mode range of the comparators at low supply voltages. The input common mode range of the LTC1440/LTC1441/LTC1442 comparators is guaranteed to extend up to \((V^+ - 1.3V)\). Therefore, if one of the comparator inputs is at the 1.182V reference voltage, the minimum supply voltage is 2.5V for a valid output reading.

The guaranteed minimum operating voltage for the LTC1440/LTC1441/LTC1442 is 2V (or ±1V). However, both the reference and comparator(s) will function with a supply voltage as low as 1.5V, but performance will degrade as the voltage goes below 2V. The voltage reference temperature coefficient will degrade slightly, and the comparators will have less output drive with an increase in propagation delay. At the reduced supply voltages, the input common mode range of the comparator(s) will still typically extend from the negative supply to approximately 1.1V below the positive supply.
TYPICAL APPLICATIONS

10-Bit 30µA A/D Converter

32.768kHz “Watch Crystal” Oscillator
PACKAGE DESCRIPTION

LTC1440/LTC1441/LTC1442

DD Package
8-Lead Plastic DFN (3mm × 3mm)
(Reference LTC DWG # 05-08-1698)

NOTE:
1. DRAWING TO BE MADE A JEDEC PACKAGE OUTLINE MO-229 VARIATION OF (WEED-1)
2. DRAWING NOT TO SCALE
3. ALL DIMENSIONS ARE IN MILLIMETERS
4. DIMENSIONS OF EXPOSED PAD ON BOTTOM OF PACKAGE DO NOT INCLUDE MOLD FLASH. MOLD FLASH, IF PRESENT, SHALL NOT EXCEED 0.15mm ON ANY SIDE
5. EXPOSED PAD SHALL BE SOLDER PLATED
6. SHADED AREA IS ONLY A REFERENCE FOR PIN 1 LOCATION ON TOP AND BOTTOM OF PACKAGE

MS8 Package
8-Lead Plastic MSOP
(Reference LTC DWG # 05-08-1660)

NOTE:
1. DIMENSIONS IN MILLIMETER/(INCH)
2. DRAWING NOT TO SCALE
3. DIMENSION DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
4. DIMENSION DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSIONS.
5. LEAD COPLANARITY (BOTTOM OF LEADS AFTER FORMING) SHALL BE 0.102mm (.004”) MAX
PACKAGE DESCRIPTION

N8 Package
8-Lead PDIP (Narrow 0.300)
(LTC DWG # 05-08-1510)

S8 Package
8-Lead Plastic Small Outline (Narrow 0.150)
(LTC DWG # 05-08-1610)

NOTE:
1. DIMENSIONS ARE INCHES
2. DRAWING NOT TO SCALE
3. THESE DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.
   MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED .010 INCH (0.254mm)
10kHz V/F Converter

**RELATED PARTS**

<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>DESCRIPTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTC1443</td>
<td>1.182V Reference with Micropower Quad Comparators</td>
<td>1% Accuracy, 8.5µA Maximum Current, Ref Output Drives 0.01µF</td>
</tr>
<tr>
<td>LTC1444/LTC1445</td>
<td>1.2V Reference with Quad Comparator with Adjustable Hysteresis</td>
<td>1% Accuracy, 8.5µA Maximum Current, Ref Output Drives 0.01µF</td>
</tr>
<tr>
<td>LTC1540</td>
<td>1.182V Reference with Nanopower Comparator with Adjustable Hysteresis</td>
<td>DFN Package 0.3µA Quiescent Current (Typical), Reference Drives 0.01µF</td>
</tr>
<tr>
<td>LTC1541</td>
<td>1.2V Reference with Micropower Amplifier and Comparator</td>
<td>DFN Package 1.25% Accuracy, Rail-to-Rail Out, Low Offset Amplifier</td>
</tr>
<tr>
<td>LTC1842/LTC1843</td>
<td>1.82V Reference with Dual Comparators with Adjustable Hysteresis</td>
<td>1% Accuracy, Open-Drain Out, Reference Drives 0.01µF</td>
</tr>
<tr>
<td>LTC1998</td>
<td>1.2 Reference with Comparator with Adjustable Thresholds</td>
<td>Li-Ion Low Battery Monitor, SOT23, 1% Accuracy</td>
</tr>
<tr>
<td>LT6700-1</td>
<td>0.4 Reference with Low Voltage Dual Comparators</td>
<td>SOT23, 1.4V to 18.5V Supply Range, ±2% Over Temperature</td>
</tr>
<tr>
<td>LT6700-2/LT6700-3</td>
<td>0.4 Reference with Low Voltage Dual Comparators</td>
<td>SOT23, 1.4V to 18.5V Supply Range, ±2% Over Temperature</td>
</tr>
</tbody>
</table>
Tyco Electronics
BNC Jack, Right Angle
5227161-2
13. PJ-102BH: DC Power Jack

CUI Inc
DC Power Jack
PJ-102BH
SPECIFICATIONS:
RATING: 24V DC @ 5A
CONTACT RESISTANCE: 50m OHMS MAX
INSULATION RESISTANCE: 50M OHMS MIN; 100V DC
VOLTAGE WITHSTAND: 500V AC R.M.S. FOR 1 MINUTE
LIFE: 5,000 CYCLES

CENTER PIN DIAMETER 2.5mm Dia.

MODEL NO.  PJ-102BH
SCHEMATIC  

CENTER PIN  Copper  Nickel
TERMINAL 1  Brass  Silver
TERMINAL 2  Copper Alloy  Silver
TERMINAL 3  Brass  Silver
HOUSING  PBT