Version 1.0

MACH64 Programmble Logic Starter Kit

Hands-On™ Guide

Andre' LaMothe

Nurve Networks LLC

MACH64 Programmable Logic Starter Kit Hands-On™ Guide Version 1.0

Author
Andre' LaMothe

Editor/Technical Reviewers
Patrick Tanner
Glenn Jones
Maksim Djackov
Daniel Quakenbush
Jodell Bumatay

A very special thanks to all the editors. Each had a different style and hopefully caught most typographical and technical errors ☺. I am still mystified how every single "it's" got turned into "its", but I think it has something to do with <Replace All> and <Find Next> ☺

Printing
0001

ISBN
Pending

Trademarks
All terms mentioned in this user manual that are known to be trademarks or service marks have been appropriately capitalized. Nurve Networks LLC cannot attest to the accuracy of this information. Use of a term in this user manual should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer
Every effort has been made to make this user manual as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an ***"as is"*** basis. The authors and the publisher shall have neither liability nor any responsibility to any person or entity with respect to any loss or damages arising from the information contained in this user manual.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Licensing, Terms & Conditions

NURVE NETWORKS LLC, . END-USER LICENSE AGREEMENT FOR HYDRA HARDWARE, SOFTWARE , EBOOKS, AND USER MANUALS

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE USING THIS PRODUCT. IT CONTAINS SOFTWARE, THE USE OF WHICH IS LICENSED BY NURVE NETWORKS LLC, INC., TO ITS CUSTOMERS FOR THEIR USE ONLY AS SET FORTH BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE OR HARDWARE. USING ANY PART OF THE SOFTWARE OR HARDWARE INDICATES THAT YOU ACCEPT THESE TERMS.

GRANT OF LICENSE: NURVE NETWORKS LLC (the "Licensor") grants to you this personal, limited, non-exclusive, non-transferable, non-assignable license solely to use in a single copy of the Licensed Works on a single computer for use by a single concurrent user only, and solely provided that you adhere to all of the terms and conditions of this Agreement. The foregoing is an express limited use license and not an assignment, sale, or other transfer of the Licensed Works or any Intellectual Property Rights of Licensor.

ASSENT: By opening the files and or packaging containing this software and or hardware, you agree that this Agreement is a legally binding and valid contract, agree to abide by the intellectual property laws and all of the terms and conditions of this Agreement, and further agree to take all necessary steps to ensure that the terms and conditions of this Agreement are not violated by any person or entity under your control or in your service.

OWNERSHIP OF SOFTWARE AND HARDWARE: The Licensor and/or its affiliates or subsidiaries own certain rights that may exist from time to time in this or any other jurisdiction, whether foreign or domestic, under patent law, copyright law, publicity rights law, moral rights law, trade secret law, trademark law, unfair competition law or other similar protections, regardless of whether or not such rights or protections are registered or perfected (the "Intellectual Property Rights"), in the computer software and hardware, together with any related documentation (including design, systems and user) and other materials for use in connection with such computer software and hardware in this package (collectively, the "Licensed Works"). ALL INTELLECTUAL PROPERTY RIGHTS IN AND TO THE LICENSED WORKS ARE AND SHALL REMAIN IN LICENSOR.

RESTRICTIONS:
(a) You are expressly prohibited from copying, modifying, merging, selling, leasing, redistributing, assigning, or transferring in any matter, Licensed Works or any portion thereof.
(b) You may make a single copy of software materials within the package or otherwise related to Licensed Works only as required for backup purposes.
(c) You are also expressly prohibited from reverse engineering, decompiling, translating, disassembling, deciphering, decrypting, or otherwise attempting to discover the source code of the Licensed Works as the Licensed Works contain proprietary material of Licensor. You may not otherwise modify, alter, adapt, port, or merge the Licensed Works.
(d) You may not remove, alter, deface, overprint or otherwise obscure Licensor patent, trademark, service mark or copyright notices.
(e) You agree that the Licensed Works will not be shipped, transferred or exported into any other country, or used in any manner prohibited by any government agency or any export laws, restrictions or regulations.
(f) You may not publish or distribute in any form of electronic or printed communication the materials within or otherwise related to Licensed Works, including but not limited to the object code, documentation, help files, examples, and benchmarks.

TERM: This Agreement is effective until terminated. You may terminate this Agreement at any time by uninstalling the Licensed Works and destroying all copies of the Licensed Works both HARDWARE and SOFTWARE. Upon any termination, you agree to uninstall the Licensed Works and return or destroy all copies of the Licensed Works, any accompanying documentation, and all other associated materials.

WARRANTIES AND DISCLAIMER: EXCEPT AS EXPRESSLY PROVIDED OTHERWISE IN A WRITTEN AGREEMENT BETWEEN LICENSOR AND YOU, THE LICENSED WORKS ARE NOW PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR THE WARRANTY OF NON-INFRINGEMENT. WITHOUT LIMITING THE FOREGOING, LICENSOR MAKES NO WARRANTY THAT (i) THE LICENSED WORKS WILL MEET YOUR REQUIREMENTS, (ii) THE USE OF THE LICENSED WORKS WILL BE UNINTERRUPTED, TIMELY, SECURE, OR ERROR-FREE, (iii) THE RESULTS THAT MAY BE OBTAINED FROM THE USE OF THE LICENSED WORKS WILL BE ACCURATE OR RELIABLE, (iv) THE QUALITY OF THE LICENSED WORKS WILL MEET YOUR EXPECTATIONS, (v) ANY ERRORS IN THE LICENSED WORKS WILL BE CORRECTED, AND/OR (vi) YOU MAY USE, PRACTICE, EXECUTE, OR ACCESS THE LICENSED WORKS WITHOUT VIOLATING THE INTELLECTUAL PROPERTY RIGHTS OF OTHERS. SOME STATES OR JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES OR LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY MAY LAST, SO THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU. IF CALIFORNIA LAW IS NOT HELD TO APPLY TO THIS AGREEMENT FOR ANY REASON, THEN IN JURISDICTIONS WHERE WARRANTIES, GUARANTEES, REPRESENTATIONS, AND/OR CONDITIONS OF ANY TYPE MAY NOT BE DISCLAIMED, ANY SUCH WARRANTY, GUARANTEE, REPRESENATION AND/OR WARRANTY IS: (1) HEREBY LIMITED TO THE PERIOD OF EITHER (A) Five (5) DAYS FROM THE DATE OF OPENING THE PACKAGE CONTAINING THE LICENSED WORKS OR (B) THE SHORTEST PERIOD ALLOWED BY LAW IN THE APPLICABLE JURISDICTION IF A FIVE (5) DAY LIMITATION WOULD BE UNENFORCEABLE; AND (2) LICENSOR'S SOLE LIABILITY FOR ANY BREACH OF ANY SUCH WARRANTY, GUARANTEE, REPRESENTATION, AND/OR CONDITION SHALL BE TO PROVIDE YOU WITH A NEW COPY OF THE LICENSED WORKS. IN NO EVENT SHALL LICENSOR OR ITS SUPPLIERS BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT LICENSOR HAD BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE LICENSED WORKS. SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

SEVERABILITY: In the event any provision of this License Agreement is found to be invalid, illegal or unenforceable, the validity, legality and enforceability of any of the remaining provisions shall not in any way be affected or impaired and a valid, legal and enforceable provision of similar intent and economic impact shall be substituted therefore.

ENTIRE AGREEMENT: This License Agreement sets forth the entire understanding and agreement between you and NURVE NETWORKS LLC, supersedes all prior agreements, whether written or oral, with respect to the Software, and may be amended only in a writing signed by both parties.

NURVE NETWORKS LLC
12724 Rush Creek Lane
Austin, TX 78732

Version & Support/Web Site

This document is valid with the following hardware, software and firmware versions:

- MACH64 Programmable Logic Starter Kit
- Lattice ispLever Classic Version 7.0.

The information herein will usually apply to newer versions but may not apply to older versions. Please contact Nurve Networks LLC for any questions you may have.

Visit **www.xgamestation.com** for downloads, support and access to the XGameStation/HYDRA user community and more!

For technical support, sales, general questions, share feedback, please contact Nurve Networks LLC at:

support@nurve.net / **nurve_help@yahoo.com**

MACH64 - Programmable Logic Starter Kit
Hands-On™ Guide Version 1.0

1.0 Overview and Goals

Welcome to the ***MACH64 – Programmable Logic Starter Kit!*** With this kit you are on your way to mastering the fascinating world of CPLDs (**C**omplex **P**rogrammable **L**ogic **D**evices) and technology. The purpose of this kit is two fold;

- First, to provide a development platform with a ***built-in*** programmer and prototyping area for the very capable Lattice Semiconductor **ispMACH 4064** CPLD that is low cost and very fast.

- Secondly, as a starter kit for users interested in learning about programmable logic either in the classroom or individually.

The manual is written in a semi-linear fashion, but each section tries to stand on its own, so you can skip to more interesting subjects. You might want to skim the entire manual before doing anything, so you know what to expect. Also, the manual is very visual especially in the software installation sections. Feel free to skim these areas as you do the install, but they are annotated so heavily since one incorrect setting in the tool chain can cause a lot of headaches.

1.1 Kit Contents

Figure 1.0 – The MACH64 Programmable Logic Starter Kit Contents (manual not shown).

NOTE	Even though the kit uses the Lattice **ispMACH 4064** CPLD chip specifically (from the **ispMACH 40xx** series CPLDs), the concepts and techniques are applicable to any CPLD from any manufacturer. CPLDs are like hard drives, you can always get one bigger and faster, but they all work in the same way and store the same kinds of data. Hence, if you write code for a **ispMACH 4064** (64 logic blocks) part, there is no problem porting it to a larger, faster part. Each chip simply differs in speed, density and amount of I/O. For example, we will make a blinking light demo that would fit on any CPLD with a single logic block, but a NTSC generator experiment might need the **ispMACH 4032** (32 logic blocks) or better to fit.

Referring to Figure 1.0, the MACH64 kit consists of the development board itself, power adapter, cables, electronic components for the experiments, as well as a CD with all the source and tools. Carefully, lay all the parts out on a table and confirm you have the following items:

MACH64 Programmable Logic Kit Parts List.

Main Components
(1) MACH64 development kit PCB itself with standoffs and 4-jumpers on J15.
(1) 9V DC 300mA wall adapter with tip + and ring -, 2.1mm barrel connector.
(1) 4-6 ft DB25 male to male parallel port programming cable.
(1) 3-6 ft RCA male to make A/V cable for NTSC and audio experiments.
(1) PC formatted CD-ROM with tools, source, and this document on it.
(1) External programming cable, 2x5 pin female to female headers, ribbon cable, 16 -18 inches.
TTL/CMOS Oscillators
(1) 3.579495 MHz NTSC color burst oscillator, half or full size.
(1) 1 MHz oscillator, half or full size.
(1) 25.175 MHz VGA oscillator, half or full size.
Resistors
(10) 100 ohm 1/8th watt, 5% tolerance, color code (brown, black, brown, gold).
(10) 220 ohm 1/8th watt, 5% tolerance, color code (red, red, brown, gold).
(10) 470 ohm 1/8th watt, 5% tolerance, color code (yellow, purple, brown, gold).
(10) 1K ohm 1/8th watt, 5% tolerance, color code (brown, black, red, gold).
(10) 2.2 ohm 1/8th watt, 5% tolerance, color code (red, red, red, gold).
(10) 10K ohm 1/8th watt, 5% tolerance, color code (brown, black, orange, gold).
Capacitors
(2) 0.1uf, monolithic, non-polarized (104 indicator written on it).
(2) 0.001 uf, monolithic, non-polarized (102 indicator written on it).
(1) 1uf, electrolytic, aluminum radial canister, polarized, + or – lead will be marked with indicator.
(1) 10uf, electrolytic, aluminum radial canister, polarized, + or – lead will be marked with indicator.
Semiconductors
(2) 2N2222 (or 2N3904) NPN bipolar transistors TO92 package.
(2) 1N4001 1A, general purpose diodes.
(2) LEDs (green or red).
Hook-up Wires and Mechanicals
(20) Long length, ~6.0", 24-26 gauge.
(20) Short length, ~3.0", 24-26 gauge.
(4) Stand-offs, screws, and washers (usuall in one of the bags with wires).

NOTE	Some of the values for the resistors and capacitors might vary slightly from batch to batch of kits.

	For example, you might have 200 ohm resistors instead of 220 ohm. The important thing is that the ratios and relationships in the circuits are the same, thus 10-20% change in components will have no effect in most cases.

Make sure you have all the parts listed in Table 1.0. Many of the components are not needed for the experiments, but come with the kit as a courtesy to save you a trip to the electronics store! Also, the MACH64 board comes with 4-screws and stand-offs that you must assemble, typically they will be in one of the bags with the hookup wires.

1.2 System Requirements

The tool chain for the Lattice CPLD development tools is called ***ispLever Classic*** and is written for a **Windows PC**, thus you will need a **Windows XP, 2000** class PC with a few hundred megs of free disk space and a decent Pentium III class processor or better. Also, the PC will need a **DB25 female parallel printer port** on it since that's how we connect the PC to the development board and use the built-in programmer. If you don't have a parallel printer port, then you can get an adapter such as the Sabrent USB to parallel port converter sold at Tiger Direct:

http://www.tigerdirect.com/applications/SearchTools/item-details.asp?EdpNo=1699044&CatId=471

Or if you want to add a new PCI card then you can get this Cables Unlimited card:

http://www.tigerdirect.com/applications/SearchTools/item-details.asp?EdpNo=1003991&CatId=512

Or while you're at it, you might as well get an extra couple old school DB9 serial ports with this cable:

http://www.worldofcables.com/store/viewitem.asp?idproduct=2065

Also, many of the more interesting experiments generate NTSC and VGA signals, so you should have a small TV available (13-17 inch is nice) with RCA A/V inputs (located on the front preferably). Additionally, if you want to see the VGA demos you will have to plug a VGA monitor into the MACH64 VGA port, so either an extra VGA monitor, or a switch box (KVM) that allows you to switch between MACH64 output and your PC is desired to get the most out of the kit.

Lattice Tool Chain Licensing Requirements

The PC that you install the software on needs a **N**etwork **I**nterface **C**ard or "NIC" (just a fancy word for Ethernet adapter) since the Lattice Semiconductor licensing engine uses the NIC's MAC (**M**edia **A**ccess **C**ontrol) to generate a key that allows the software to work only on that PC. The licensing procedure is very simple and more or less you have to log onto their internet site, fill out a form, give it your NIC's MAC address which is just a 6-byte number that identifies your NIC and more or less identifies your computer uniquely, so you don't move the software.

You can find this number by running the command **"ipconfig /all"** from the Windows command line, but we will get to that later. Once you log into the Lattice site, generate the license, they will send you an email with the license file as a **.txt** attachment, you copy into a directory the tools were installed and you don't need to worry about it for another year. The bottom line is if you are using this on a really old PC that has no NIC interface card (or a built-in unit), you are going to need one, even if you don't connect it to internet since the licensing uses this to uniquely identify your machine.

1.3 Target Audience

As mentioned this kit is for two kinds of users; experienced and non-experienced. For the experienced users that are familiar with programmable logic and CPLDs, the MACH64 PLSK is a low cost development platform for the Lattice ispMACH 4064 device with a built-in programmer. This kit costs less than any other isp4064 kit on the market, plus the built-in programmer can be used to program other Lattice devices including CPLDs and FPGAs. Thus, the kit doubles as a programmer as well as a development kit. Additionally, the I/O, RCA, VGA, and other features of the board make it ideal to experiment and develop your CPLD designs before trying them on your target boards.

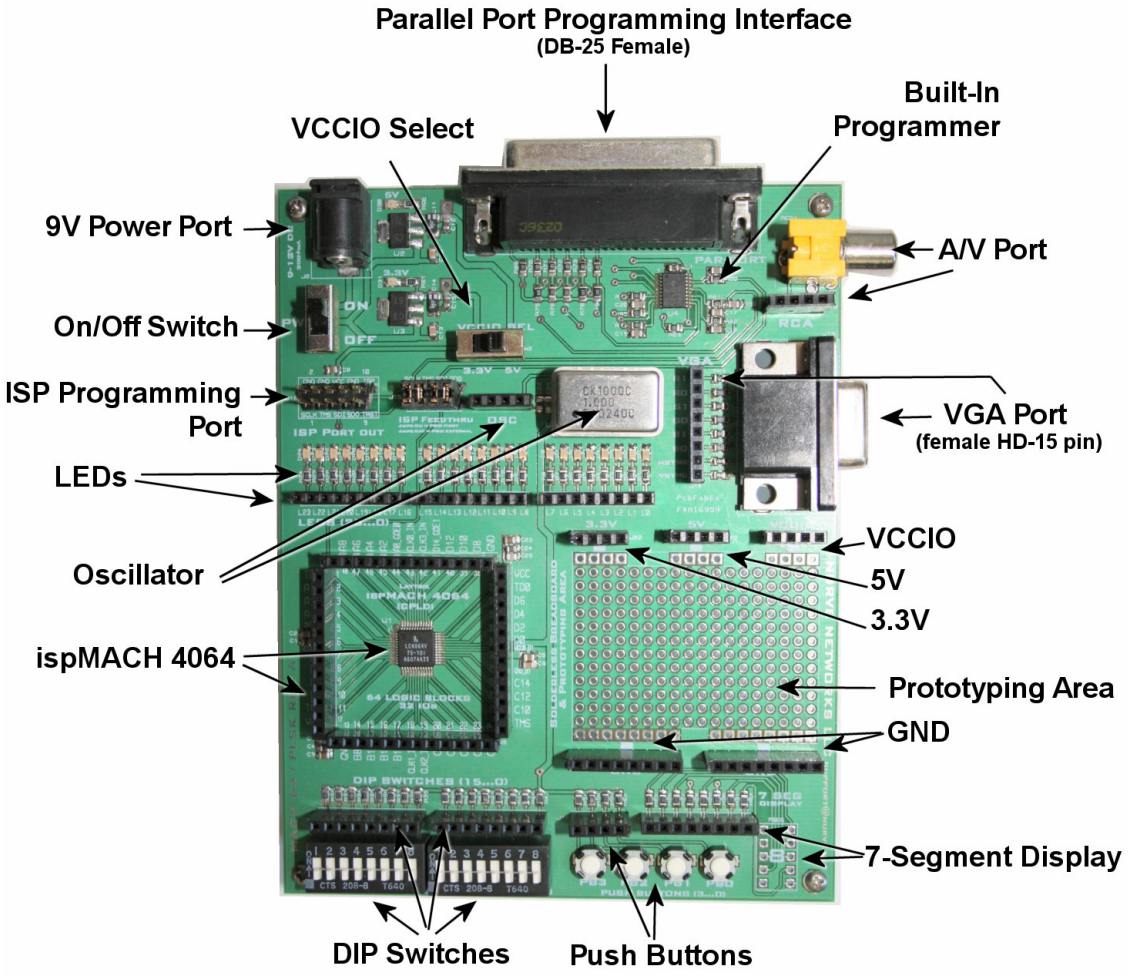
For the inexperienced users the ***MACH64 Programmable Logic Starter Kit*** is a way to get your feet wet with programmable logic. As a user of programmable logic myself, I have used just about every chip and every tool known to the human race. The single most frustrating thing is the documentation and cryptic examples deployed in all these tools and development kits. I know there are lots of people out there that are into electronics or maybe simply have an interest in electronics and want to check out what programmable logic is all about, but once they start investigating it becomes

overwhelming. Moreover, many times when you buy a development kit, a programmer must be purchased separately and lots of other gotcha's. Therefore, this kit was designed to be a one stop shop to get a new programmable logic user up and running and having fun with programmable logic as soon as possible.

That said, this is complex stuff, and there is no way around installing the tools and learning the development language(s) of programmable logic, but hopefully this kit makes it a lot easier. Additionally, this is a hands-on kit, there will be very little theory. We are going to learn by doing. Moreover, I assume you know nothing about **Verilog**, **VHDL**, **ABEL**, **CUPL**, or any other "**HDL**" (**H**ardware **D**escription **L**anguage). We are going to use **ABEL** (**A**dvanced **B**oolean **E**quation **L**anguage) for the experiments in this kit and for the most part you will learn the language on the fly. If you know C/C++, BASIC, PASCAL, ASM, or any other serious computer language you will have no problem following the examples.

Finally, the kit assumes very little knowledge of electronics as well. The idea of programmable logic is to abstract the electronics away and just deal with the behavior of the system. Of course, somewhere the rubber has to meet with the asphalt and you have to connect LEDs, switches, and wires to get signals in and out of the real-world. But, while the signals are in the programmable logic device itself, everything is virtual more or less, so you don't have to worry about these things. Thus, when we get to connecting things up even if you are inexperienced with electronics there will be step by step instructions and pictures of everything, so you shouldn't have any trouble.

Figure 2.0 - The MACH64 PCB with annotation (note the solderless breadboard hasn't been affixed).



1.4 MACH64 PCB Board Features and Overview

The **MACH64** board consists of a single ispMACH 4064 CPLD (with headers all the way around) along with a built-in parallel port programmer, a prototyping area, numerous I/O, duel 3.3V, 5.0V power supplies and a sexy colorful finish. The PCB has a built-in programmer that communicates to the PC over a parallel cable and is 100% compatible with the Lattice ispVM tool. Not only is the MACH64 the only development kit that has an on-board programmer, but the

programmer can be used to program ***external*** Lattice chips as well by removing the jumper block (at J15) and using the included 2x5 female to female programming ribbon cable.

Additionally, the PCB has a prototyping area that you can solder into as well as affix the solderless breadboard that comes with the unit. The prototyping area has power rails that output 3.3V, 5.0V as well as the VCCIO and lots of ground connections at the bottom of the region.

The MACH64 kit is full of I/O including a RCA connector for video and audio experiments, a VGA port, 16-dip switches, 4-momentary push buttons, as well as a 7-segment display. All these I/Os have labeled headers adjacent to them to get the signals to them for point to point wiring. Also, there is an oscillator dip socket that supports 8 and 14-pin oscillators with power and bypass caps, so all you do is plug in your favorite oscillator and it will start running.

Referring to Figure 2.0, let's go over the PCB itself, so you can familiarize yourself with the layout and various inputs and outputs. The paragraphs below outline the various components of the PCB (in no particular order):

Power Supply Input

At the top-left-hand corner of the board is the power input port. This is a 2.1mm female connector that accepts 9V unregulated DC (or regulated) with **TIP +**, and **RING -**. The power adapter should supply at least 300mA.

On/Off Switch and Power Supplies

The on/off switch simply gates the power to the power supply which consists of both 3.3V and 5.0V regulators. The dual power supplies are routed around the board as well as to the prototyping area. The ispMACH 4064 itself always uses 3.3V internally (the core supply), but the I/O pins can support either 3.3V or 5.0V which is referred to as **VCCIO**. Thus, the ispMACH 4064 is compatible with both 3.3V and 5.0V logic. Switching the VCCIO voltage is simple; the switch to the right of the power switch is labeled **"VCCIO SEL"**. In most cases, we are just going to leave VCCIO at 3.3V and leave it, that way, the entire system is running at 3.3V. However, all the power supplies are ***always*** available at the prototyping area.

NOTE	If you power the I/Os of the ispMACH 4064 with different voltages you will want to set the appropriate I/O type in the **"Constraint Editor"** to be safe. The Constraint Editor is a tool that is part of the Lattice tool chain that we are going to install. The Constraint Editor allows the user to program the I/O pins of the chips and fine tune various things like the voltages that the pins will output or accept. This ability helps make the chip run better. For 99% of the experiments we are going to do, you will never touch the Constraint Editor and changing the VCCIO voltage won't hurt the operation of any of the circuits developed in the labs. They are voltage tolerant in all cases.

ISP Programming Port

The ISP Programming Port is an extra feature of the MACH64 that allows the on-board programmer to program external Lattice CPLDs that are compatible (most of them). To use the external programming feature, you simply pull the jumper block from J15 and then connect the 2x5 female to female programming cable to the MACH64. The other end of the cable must go to the "target" board with the CPLD on it. And of course, you must route the signals to the proper programming pins on your target board. Appendix A. shows the design of the programmer in detail. To program the onboard CPLD (which is what we will be doing) make sure that the jumper(s) on **J15** are all on.

LEDs

There are 24 LEDs connected to header J8, J9, and J10 respectively. These LEDs referred to as LED (23..0) can be used for general outputs or whatever you're interested in. The are each current limited with a 330 ohm resistor, so they won't be damaged if you connect them directly to either the 3.3V or 5.0V supply.

Oscillator

All synchronous logic needs a clock source of some kind. You can use the ispMACH 4064 CPLD as a combinatorial device that doesn't need any clocking, but the moment you need to create a counter, state machine, or anything that needs a clock signal then some source of clock is needed. The Oscillator socket is for this purpose. Located to the **right** of the **VCCIO SEL** switch and to the left of the VGA Port, the Oscillator socket at **U5** supports both full size 14-pin oscillators as well as half size 8-pin oscillators. In either case, insert the oscillator with pin 1 to the bottom left and the

oscillator will be powered up properly. The output of the oscillator is available at header **J18** and is labeled **"OSC"**. All 4-outputs are the same signal.

ispMACH 4064 CPLD

At the lower left hand section of the PCB is the ispMACH 4064 at **U1**. It's a surface mount 48-TQFP (**T**hin **Q**uad **F**lat **P**ack) with pin 1 located at the top left of its footprint. The pins are numbered counter-clockwise and sent to the surrounding header pins which are labeled with both the pin numbers and the internal I/O pin names.

TIP	You might be thinking, ***"there is only one pin per signal on the headers around the ispMACH 4064, what if I need to send one of the signals to more than one place on the board?"*** Well, within the CPLD itself you can send signals, thus you never need to send a signal outside then wire it back inside to use it, you just use the signal name in your code and the compiler will handle its routing internally. On the other hand, say you wanted to connect one of the output pins to an LED and the same output pin to another LED? To do this, first you would connect the output pin to one of the solderless breadboard rows, then take one of the common pins from the row to each LED. It's an extra step, but needed, so a bunch of headers in parallel are not needed around the chip.

DIP Switches

There are 16-DIP switches on the MACH64 PCB in two banks of 8. The DIP switches are located on the bottom left of the board and are of the slider or rocker variety depending on your particular build. Ignore any writing on the switches, they are numbered left to right **DS15..DS0** and are **normally open** and pulled **HIGH** with a weak pull-up resistor. This detail is very important;

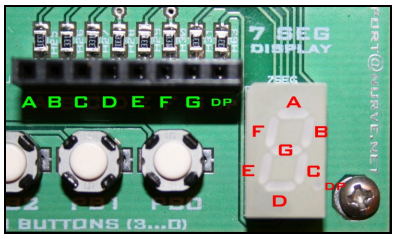
If you connect a wire to one of the output headers above the DIP switches you will get a HIGH out when the switch is OPEN (off) and a LOW out when the switch is CLOSED (on).

This may seem backwards, but is easier to implement on the PCB. Otherwise we need to use different more expensive switches and pull-down resistors which is bad design. The signals from each switch are routed to the headers **J6** and **J7** above the switches in a one-to-one fashion.

Push Buttons

There are 4-momentary push button switches located at the bottom-right of the PCB. These once again are pulled up to a weak **HIGH** when **open** and grounded **LOW** when **closed** or pressed. The 4 switches are labeled **PB3..PB0** left to right and their signals are available via the header above them to the left at **J17** (with small identifiers 3..0 below the header).

Figure 3.0 – The 7-segment LED labeling.



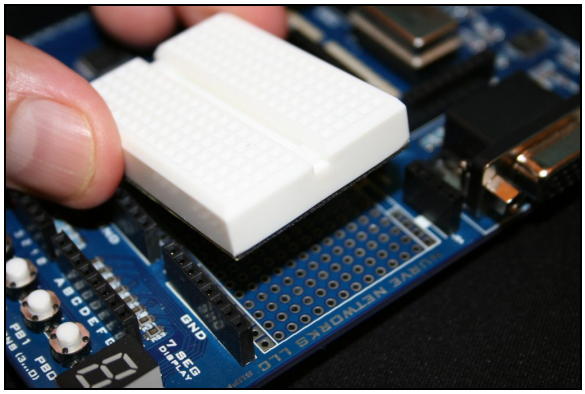
7-Segment Display

The 7-segment display is located to the bottom-right of the PCB and is a standard 0.3" 7-segment display that you might find in a clock radio. This particular 7-segment display is ***common cathode*** meaning that you apply a positive voltage to turn on each segment. The segments are labeled **A, B, C . . . F, G, DP** as shown in Figure 3.0. The **A..G** represents the segments around the digit and **"DP"** stands for **"decimal point"**. If you're lucky, you have a green display, otherwise, you have one of the usual red or yellow displays. To illuminate any of the segments simply connect 3.3V or 5.0V to the segment inputs at **J11** to the top-left of the 7-segment display itself. These inputs already have current limiting resistors, so you won't damage them by driving them directly or with the CPLD itself.

Prototyping Area and Solderless Breadboard

The prototyping area to the right of the PCB is designed, so that either soldered devices can be soldered permanently into the board or a **solderless breadboard** (a small white one comes with the kit) can be affixed to the area and used over and over. For example, you might want to solder a few chips into the PCB and turn the MACH64 dev kit into a specific application when you're done learning on it. Or on the other hand, you might want to use the included white solderless breadboard over and over, thus you simply remove the double sided tape on the back of the solderless breadboard and affix it to the prototype area (which is needed for all the labs). This is shown in Figure 2.0(b), simply remove the protective tape from the back of the solderless breadboard and affix it to the prototyping area. You can always remove the solderless breadboard with a little force and a prying carefully with a flat head screwdriver, clean the area with alcohol and it's ready for soldering again. The prototyping area also has headers both solder type and solderless for 3.3V, 5.0V, VCCIO and ground, so you have enough power to power anything you might design in the prototyping area.

Figure 2.0(b) – Adding the solderless breadboard to the MACH64 board.



VGA Port

The VGA port is located on the right-hand side of the PCB and is a standard female HD15 VGA connector at **J13** that fits a standard VGA monitor. The signals from the VGA connector for the VGA interface are the red, green, blue bits, HSYNC, VSYNC, and power. All of these signals are routed to the VGA connector properly and to the signal header as well at **J14** to the immediate left of the VGA connector. Additionally, there are 2-resistor summers on each of the red, green, and blue channels, thus a crude digital interface converts a 2-bit signal for each channel into the required 0.7V signal the VGA specification requires to drive each of the RGB signals. Additionally, there are 100 ohm current limiters on the HSYNC and VSYNC, so all the signals are LV/TTL compatible and can be driven directly from the CPLD. The signals are labeled as shown in Table 1.0.

Table 1.0 –VGA port header pin outs.

Signal Name	Description	VGA Pin
R1	High bit of Red channel	1 (R1 and R0 are summed)
R0	Low bit of Red channel	
G1	High bit of Green channel	3 (G1 and G0 are summed)
G0	Low bit of Green channel	
B1	High bit of blue channel	5 (B1 and B0 are summed)
B0	Low bit of blue channel	
HSYNC	Horizontal sync	13
VSYNC	Vertical sync	14

Note: the remainder of the VGA pins on the HD15 are connected as specified by the VGA specification, specifically all the grounds are connected to the PCB ground. All signals are TTL compatible except the RGB drives which are 0.7V p-p. A more detailed explanation of VGA will be given during the lab on the subject.

A/V Port

The RCA female A/V port is located at the top-right of the PCB. This is a passive connector that can be used to route a signal and ground off the PCB into a TV or other compatible device. The A/V RCA port is used for both NTSC video labs as well as audio labs in this manual. To send a signal to the A/V port there are 4-inputs at **J12** right under the RCA connector itself. All 4 inputs are parallel and then routed to the top of the A/V connector, while ground of the RCA connector is connected to PCB ground.

Built-In Programmer

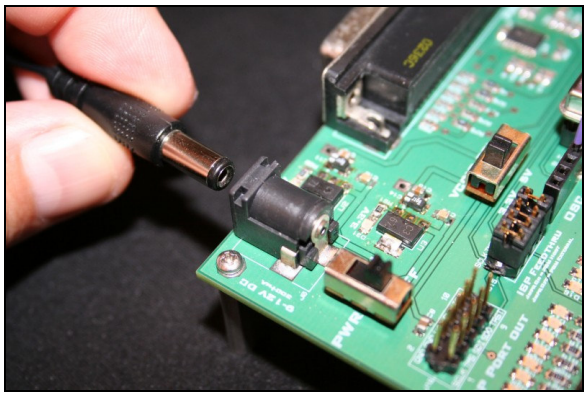
The built-in programmer is 100% compatible (for the most part) with Lattice semiconductor's parallel programming cable that they sell. By having the programming hardware built into the MACH64, you don't need to lug around a special programming cable or other hardware. Moreover, the MACH64's built-in programmer can also be used to program external Lattice CPLDs as well. You simply need to disconnect the programming interface (which consists of 4 signals) from the on-board CPLD chip, and then route the signals off board with a special cable. However, for the labs we will be programming the on-board chip and won't need this feature.

All in all the MACH64 is a very complete and compact development kit that not only allows you to develop on-board CPLD experiments with its built-in programmer, but you can use the MACH64 as a programmer and program chips on other target boards.

2.0 System Setup and Self-Test

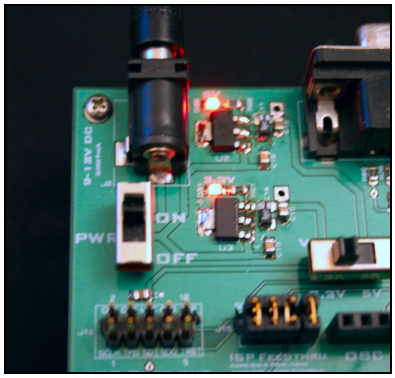
The MACH64 board comes pre-programmed with a diagnostic demo program that counts on the 7 segment display. We are going to power the MACH64 board up and confirm everything is working. Please read all the steps and review the illustrations before completing the steps, so you know what to expect.

Figure 4.0 – Inserting the power adapter into the MACH64 board.



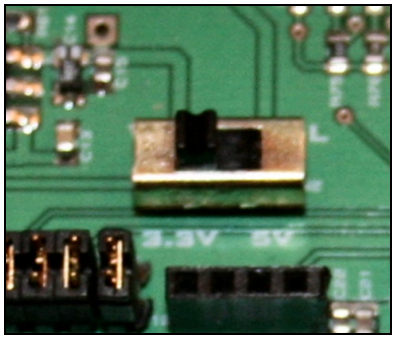
Step 1 (Plugging in the power adapter): Plug the wall power adapter into a 120V socket and then insert the male adapter into the MACH64 board as shown in Figure 4.0.

Figure 5.0 – Turning on the power.



Step 2 (Checking the power): Turn the MACH64 board **ON** with the **PWR** switch located on the **top-left** of the PCB as shown in Figure 5.0 You should see the two power indicators LEDs glow brightly indicating that power is "good" for the 5.0V and 3.3V supplies. Turn the power **OFF** after you have confirmed the board is powering up correctly.

Figure 6.0 – Confirming the I/O voltage on the PCB is set correctly.

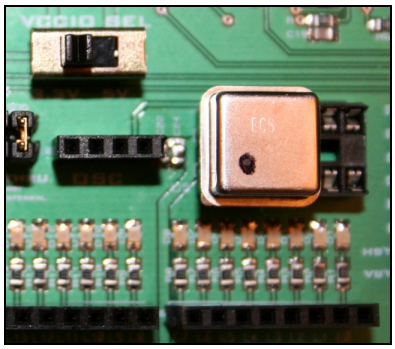


Step 3 (Verifying the voltage levels): Make sure the **VCCIO SEL** switch located at the top-middle of the board is set to **3.3V** as shown in Figure 6.0. This simply controls the voltage to the I/O pads of the chip. 5V won't hurt it, but let's set it to 3.3V to conserve power.

Figure 7.0 – Verifying all the programming jumpers are in place to route the programming signals on-board.

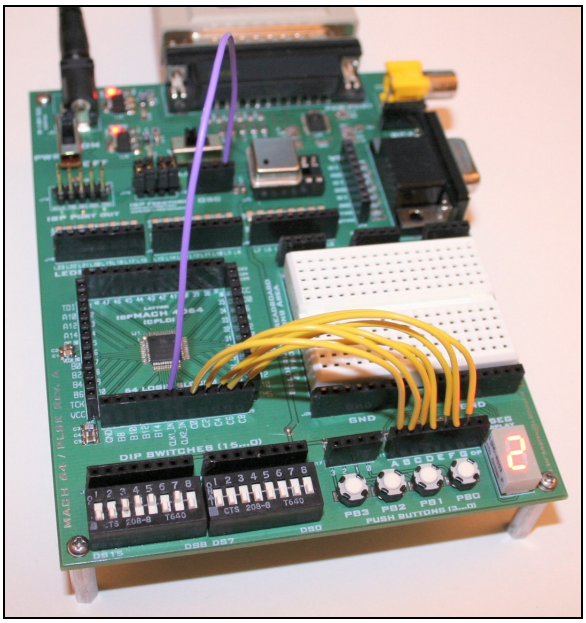
Step 4 (Verifying the programming header): Confirm that the programming header jumpers at **J15** (the **ISP Feed thru** port) are all connected as shown in Figure 7.0 These jumpers insure that the on-board chip is programmed by the tool and not an external chip connected via the external programming header (**ISP Port Out**) at **J16**.

Figure 8.0 – Insertion of the 1MHz oscillator for the demo test.



Step 5: (Insertion of the oscillator) Locate the oscillator socket at **U5**, it's a 14-pin socket that supports for full size (14-pin footprint) and half size (8-pin footprint) oscillators. Now, find the 1 MHz oscillator in the parts that came with the kit and insert it so that pin 1 of the socket (bottom left) and pin 1 of the oscillator line up as shown in Figure 8.0.

Figure 9.0 – The completed circuit for the demo program pre-programmed into the MACH64.



Step 6: (Connecting the circuit): Now, we are going to hook up the circuit. The pre-programmed demo in the ispMACH 4064 chip displays the numbers 0..9 on the 7 segment display, so the inputs to the display need to be connected to the proper output pins that were selected in the program to drive them. Take 7 of the shorter hookup wires and make the following connections shown in Table 2.0 from the bottom header under the ispMACH 4064 to the 7-segment header located to the top-left of the display itself. When complete use one of the longer wires (with a different color) to connect the clock signal to the ispMACH 4064 chip, this is shown as the last connector in the table from the **OSC** header itself (any of the 4 outputs will do) to **pin 18** (**CLK1_IN**) of the ispMACH 4064. When complete, verify your circuit is connected as shown in Figure 9.0.

Table 2.0 – Circuit Connections for the test demo program.

ispMACH 4064 (bottom header pins 20-27)	**7-Segment Header (pins A...G)**
Pin# / Signal Name	***Signal Name***
20 / C0	>A
21 / C2	>B
22 / C4	>C
23 / C6	>D
24 / C8	>E
26 / C10	>F
27 / C12	>G
	Oscillator Header (OSC)
18 / CLK1_IN	>1..4 (any of the 4 pins will work)

Step 7 (Powering up): Turn on the MACH64 and you should see the 7-segment display count from 0-9 over and over.

Troubleshooting

If the circuit doesn't work make sure of the following:

1. Verify you have the 1 MHz oscillator plugged in the correct way with pin 1 (dot on it) to the bottom left of the socket **U5**.

2. Verify that the clock signal from the oscillator via the **OSC** header is connected to **pin 18** of the ispMACH 4064 chip header, center bottom header.

3. Verify power is **ON**.

4. Check your wiring.

5. Re-check your wiring!

3.0 Quick Start Software Installation Guide

In this next section we are going to install the tool chain needed to develop applications for the ispMACH 4064 CPLD as well as program it (download the configuration bits). There is a copy of the tool chain files on the CD that works perfectly, so I suggest starting with them, so all the experiments work as expected. Then later you can download the latest version from the Lattice website.

The software we are going to use is called ***"ispLever Classic"***, it used to be called ***"ispLevel Starter"***, but recently Lattice decided to change this and split the products. Thus, the **ispLever Classic** version works with the simpler CPLDs listed in Table 3.0 below (which includes the ispMACH 4064):

Table 3.0 – Supported devices for ispLever Classic.

CPLD	**SPLD**
ispXPLD 5000MX **** ispMACH 4000B/C/V/Z*** ispMACH 5000VG ispMACH 5000B ispMACH 4A3/5 **MACH4/5** ispLSI 8000 ispLSI 5000VE ispLSI 2000VE ispLSI 1000	GAL and ispGAL **GDX** ispGDXVA ispGDX2

* - This is the device we need to support.

Note: *There is a "classic FPGA" module that can be installed that gives ispLever Classic support for a handful of FPGAs, but we don't need this for now since we will be working with CPLDs in this workbook..*

The more advanced ***ispLever Starter*** version works with the FPGAs listed below in Table 4.0:

Table 4.0 – Supported devices for ispLever Starter.

FPGA	**CPLD**
LatticeECP2: ECP2-6,12,20,35,50 (Does not include LatticeECP2S family) LatticeECP: All LatticeEC: All LatticeXP: All	MachXO: All

If you happened to want to use any of the devices listed in Table 4.0 then you would have to install **ispLever Starter**.
Verilog and VHDL Support

Additionally, both ispLever Classic and ispLever Starter have a number of additional modules that can be optionally installed such as the documentation (not really an option in my opinion) as well as **"HDL synthesis"** modules. The HDL

synthesis modules are developed by other parties and allow you to code in **Verilog** or **VHDL** and then the tool can convert your code into the suitable format for the chip. These additional modules must be installed if you're interested in Verilog or VHDL coding. By default when you download ispLever Classic's primary module it only supports **ABEL** (and a custom language EDIF) which is all we need since we are going to use ABEL for all the experiments, so we are good to go. However, when we do the installation, we are going to install the **"Precision HDL"** module since it's on the CD and nice for a rainy day. If you like you can install the **"Synplify HDL"** module as well, but you have to request a license and I am not going to cover the installation steps, but it's just a normal installer like the other HDL module.

Downloading the Tools

You do **NOT** need to download the tools, we are going to use the tools from the CD-ROM, so we are both using the exact same tool chain. However, for reference purposes and so you can request licenses, you will need to know these links:

You can download the **"ispLever Classic"** version here (as well as other modules):

http://www.latticesemi.com/products/designsoftware/isplever/ispleverclassic/index.cfm

You can download the **"ispLever Starter"** version here (as well as extra modules):

http://www.latticesemi.com/products/designsoftware/isplever/ispleverstarter/index.cfm

You are going to have to at the very least request a license for the **ispLever Classic** version, so keep that link above handy.

3.1 Installation of Software and Tools

Alright, here comes the fun part. The tool chain consists of a number of modules as noted above. You don't need all of them, but it's best to install them all since you might want them in the future. The tool chain consists of the following 3 modules:

- **Classic Primary Module** (220MB / isp7_0m1.CPLD.exe) – The main tool chain file and IDE.
- **Classic Help and User Guide** (40MB / HLP_p5.exe) – Help for the main tool chain.
- **Precision RTL Synthesis Module** (80MB / PRC_p7.exe) – This is the HDL module add on developed by Precision Corp. It supports Verilog and VHDL coding. Not needed, but nice to have if you want to try your hand at Verilog or VHDL.

Note: On the ispLever Classic download page there is another HDL/RTL synthesis tool developed by **Synplify Corp.**, you can install it as well, but it requires yet another licensing step. However, for advanced HDL design you may prefer it to the default **Precision Corp**. tool. It's up to you. For now, I suggest only installing the 3 tools above as outlined in the sections below.

Copying the Files from the CD-ROM

Before installing the tool chain you need to drag the contents of the CD-ROM to your hard drive. This will make the installations faster and will simplify the projects, so you don't have to keep accessing the CD-ROM. If you damage a file, you can always re-copy from the CD, so this is the best approach.

Insert the CD-ROM into your computer, open the device and drag and copy the entire directory MACH64\ onto your hard drive near the root of your destination drive (so path names aren't too long). From now on, we will refer to any file from its root path **MACH64**\...

3.1.1 Step 1: Installing ispLever Classic Primary Module

The installation process isn't that complex, but there are a lot of modules and a lot of places where things can go wrong. Thus, the following sections outline the installation along with detailed screen shots of a typical installation, so you can

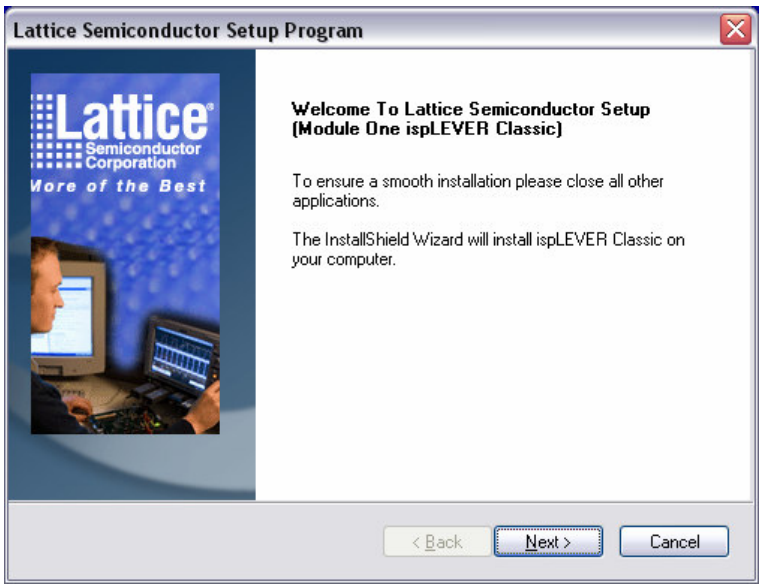
follow along exactly. This is easy stuff, but many users seem to have bad luck with installers, so these screen shots and instructions should ease the pain if something goes awry.

The first step is to install the ispLever Classic Primary Module, it's located here:

\MACH64\tools\isp7_0m1.CPLD.exe

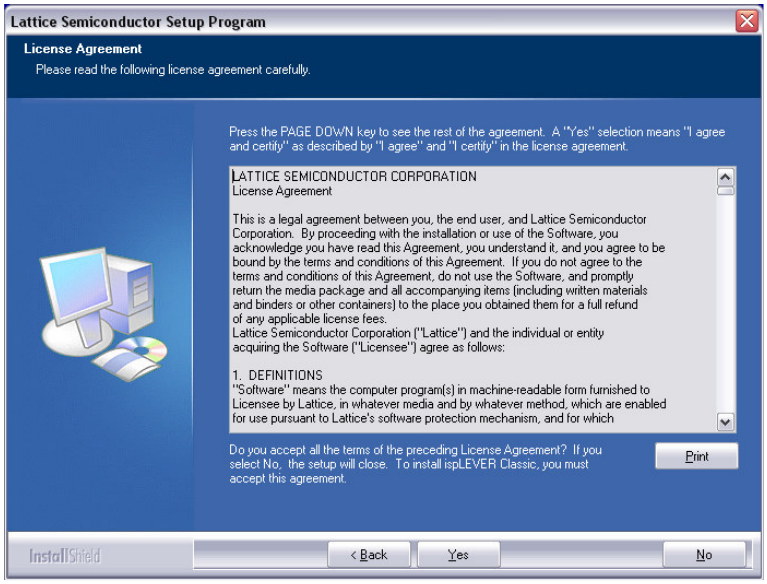
Find the file and launch the installer. Once the installer starts, you will see the initial splash screen shown in Figure 10.0,

Figure 10.0 – Initial splash screen for ispLever Classic Installer.



Click **<Next>** and the license agreement will display as shown as shown in Figure 11.0.

Figure 11.0 – ispLever Classic license agreement.



It's the typical license agreement. Click **<Yes>** to continue to the installation directory dialog shown in Figure 12.0.