DC Integration Innovations
School of Engineering Science
Simon Fraser University, Burnaby, BC, V5A 1S6

November 5, 2001

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, BC
V5A 1S6

Re: ENSC 340 Design Specification for a Multiplexed Wiring System

Dear Dr. Rawicz:

DC Integration Innovations has the goal of developing a more efficient, expandable, and maintainable solution for DC signal wiring. We will replace complicated and expensive wiring networks with a unified bus that uses a time division multiplexing strategy to carry a large number of DC signals.

Our attached design specification outlines the methods by which we expect to meet the commitments put forth in our functional specification. We have included design details for the hardware, firmware, software and system interfaces. A system test plan is also outlined.

DC Integration Innovations is comprised of Ian Chan, Gary Lau, Erik Haberger, and Aydin Kilic. Each of these members brings their own unique skills to form a motivated and well-rounded team.

Please feel free to contact us if you have any comments or questions about our design specification. We can be reached at **dc-i2@sfu.ca**

Sincerely,

Erik Haberger, CTO
DC Integration Innovations

DC Multiplexed Wiring System

Design Specification

DC Integration Innovations

Team Members Ian Chan
Gary Lau
Erik Haberger
Aydin Kilic

Contact **Ian Chan**
dc-i2@sfu.ca

Submitted To: Dr. Andrew Rawicz
Steve Whitmore
School of Engineering Science
Simon Fraser University

Submitted: November 5, 2001

Executive Summary

With the ever-increasing complexity of electronic systems today, interconnections between various functional blocks become more and more of a challenge, and this increased complexity also brings increased costs. DC Integration Innovations proposes a solution to this growing problem with the introduction of the *Direct Current – Integrated in One* (**DC-1*i***) multiplexed wiring system. The **DC-1*i*** replaces the thousands of wires of a traditional wiring system with a single unified bus that provides an interconnection between all the nodes in the system.

The team of engineers at DC Integration Innovations has devised a design solution to meet all of the requirements put forth in the previously published functional specification. Highlights of the design include:

- Low Cost – all of the parts included are easily obtainable industry standard parts that are very inexpensive. In addition, the system is designed to facilitate ease of manufacturing and installation.
- Reliable – all system components are designed to tolerate expected environmental and operating conditions. The bus architecture allows no single point of failure, and the bus protocol has been carefully designed to eliminate any possibility of data error.
- Expandable – the design is based on an open architecture, allowing for easy expansion if future applications require enhanced capabilities or performance.
- Visible – the System Diagnostic Unit (SDU) allows technicians to see all bus activity, allowing ease of field debugging.

Implementation of the DC-1i is currently underway based on the design specifications contained in this document. Our project will be completed by December 2001.

Table of Contents

1 Introduction

The DC-1i system unifies wiring configurations involving many wires, as is common in such intricate electrical systems as those for automobiles. A unique implementation scheme makes this possible. The system includes a Central Control Unit (CCU) and several Input/Output Nodes (IONs) that are all connected by a unified bus with wrap-around redundancy. The DC-1i system is intended to be an efficient alternative to the costly and complex traditional wiring systems that exist in automobiles today. The efficiency of the DC-1i stems from the simplicity of its design, which provides for faster, cheaper, and easier implementation and diagnosis of the system.

1.1 Scope

This document describes the design that the engineers at DCI^2 have proposed to meet the functional requirements as outlined in the DC-1i Functional Specification. The scope of this document covers all aspects of the project, including the following:

- Hardware
- Firmware
- Software
- Protocols / Interfaces
- Test Plan

While all design material in this document has been carefully thought through, it is still possible and indeed likely that certain aspects will be modified. Changes will be required for one of the following two reasons: First, if the Functional Specification is modified, then the design must be also modified to ensure compliance. Second, if errors or misjudgements are recognized which imply that the designs contained in this document will not meet the functional specification, then this document will be updated with new designs that will ensure compliance.

1.2 Glossary

CCU - Central Control Unit
ION - Input/Output Node
SDU - System Diagnostic Unit
ICSP - In Circuit Serial Programming

1.3 Reference Documents

[1] DC Multiplexed Wiring System Project Proposal. DC Integration Innovations.
[2] DC1*i* Bus Protocol. DC Integration Innovations.
[3] Standard TTL Logic Levels. Twisted Pair web resource.

[4] Interfacing the Standard Parallel Port. C. Peacock.
[5] DC Multiplexed Wiring System Functional Specification, DC Integration Innovations.

1.4 Audience

- Investors and venture capitalists will use this document as a reference and a definitive source of information regarding the functionality and economic viability of the DC-1i system.

- Designated design engineers will both reference and update this document as future versions of the DC-1i system are developed.

- Other design engineers will reference this document when developing auxiliary or complementary products for the DC-1i system.

- Executive staff and management will use this document as a reference and checklist throughout product development. Having a clear and concise design document will help ensure that all of the original design goals are incorporated in the final product. Also, management may update this document if they decide to modify existing requirements or to add further requirements

- Marketing staff will reference this document as needed to prepare marketing materials. In addition, this document may be forwarded to companies (such as automobile manufacturers) who may be interested in implementing our system and would like details on what kind of performance they can expect form the DC-1i.

2 System Overview

The *Direct Current – Integrated in One* (**DC-1*i***) is a time division multiplexed wiring system that replaces the complication of a traditional wiring system with a unified bus architecture. This new multiplexed wiring approach replaces thousands of individual wires with a single data bus. A block diagram of the **DC-1*i*** system is shown below in Figure 1.

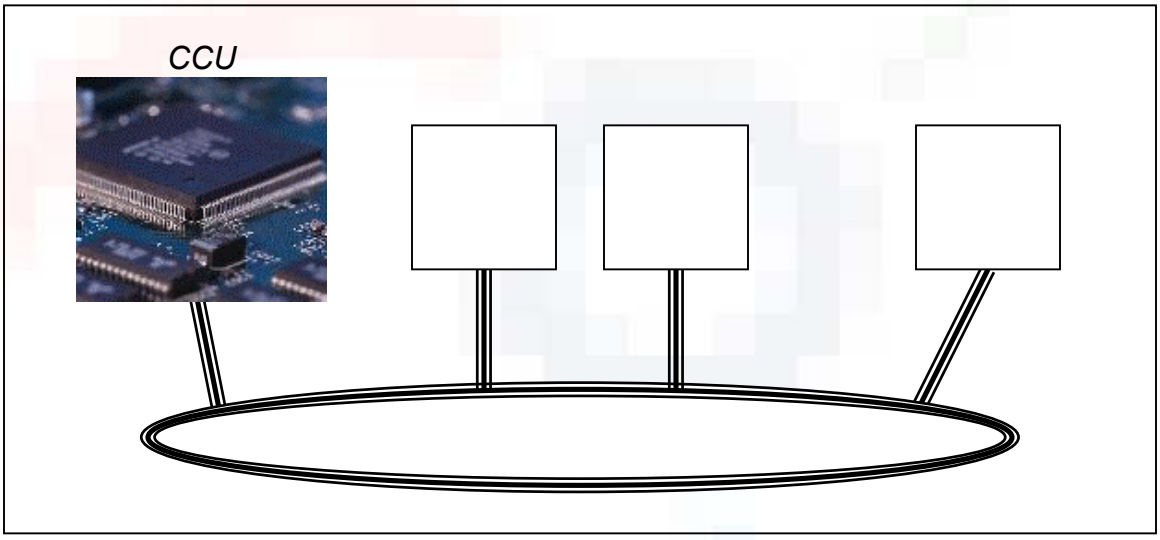


Figure 1: DC1*i* System Overview

2.1 Functional Blocks

The **DC-1*i*** system consists of a Central Control Unit (CCU) that controls the use of the unified bus. Different points on the bus provide input and output to various devices in the system, and each of these points is referred to as an Input/Output Node (ION). An additional module, called the System Diagnostic Unit (SDU), can be connected to the unified bus and used to diagnose problems in the system or in the external circuitry.

2.1.1 Central Control Unit (CCU)

The CCU is responsible for the arbitration of the bus. This unit determines the address of the transmitting and receiving channels and connects input nodes with output nodes. It alone is responsible for the timing of the bus transactions.

2.1.2 Input / Output Node (ION)

The IONs are points on the bus where the **DC-1*i*** system interfaces with the external world. At these nodes, data can get on and off the bus and transfer to or from external circuitry, such as switches, sensors, or actuators. Each node may contain multiple input or output channels. Under the supervision of the CCU, Data is transferred from a single input channel to multiple output channels via the bus.

2.1.3 System Diagnostic Unit (SDU)

The SDU is an external module that can be inserted at any point on the bus. Once attached, the SDU gathers data on every channel that is being transmitted on the bus. The SDU relays the data on all channels to a PC or laptop via a parallel port interface. This data is then interpreted and any problems with the **DC-1*i*** system are diagnosed. This unit can also, and more commonly, be connected for diagnosing problems that are not related to the **DC-1*i*** system, but instead are problems with the external circuitry. The SDU will give service personnel a detailed view of what data is being transmitted on the bus, and thereby allow them to pinpoint any malfunctions in the external circuitry.

3 System Interface

3.1 CCU / ION Interface

The CCU and IONs will communicate with each other via a strictly defined bus protocol. In this protocol, the actual data is transmitted on a peer-to-peer basis from one node to another. The CCU will act as the bus arbitrator, and will dictate the order in which the channels will be transmitted and will enforce specific timings in order to ensure that all signals get transmitted reliably and with a sufficient refresh rate.

3.1.1 Bus Wires

Here we will explain the different wires that will be used on the bus. For each wire, we will detail the purpose of the signal being carried, as well as the originator and the targets of the signal.

- Address Lines (ADDR0 – ADDR7) The address lines will carry the address of the data channel that is currently being carried on the bus. The format of the address lines is a 8-bit binary word that is transmitted in parallel across 8 conductors. The address signal is generated by the CCU, which has the right to place any address that it chooses on the address lines. Every ION in the system will monitor the address lines to see if the address present matches an address that the ION is configured to transmit or receive on.

- Address Ready (ARDY) The Address Ready line is used to prevent errors that may arise during the time when the CCU is changing the value on the address lines. Such errors may occur if an ION incorrectly recognizes an ambiguous address during transition on the bus and acts on it. The ARDY signal is generated by the CCU and serves to signal the IONs that the value on the address lines is indeed valid. The IONs will monitor ARDY and will only enable their address comparison circuitry when the ARDY is active. The CCU will ensure that there are appropriate time intervals between changing the value on the address lines and applying the ARDY signal.

- Data Ready (DRDY) The Data Ready line is used to signal that the data which is on the bus is valid. It is generated by the ION that is configured to transmit the data for the channel corresponding to the address on the address lines. The DRDY will be monitored by all IONs. If DRDY is asserted, then the ION that is configured to output the current channel will activate its sample and hold circuit to obtain the data. The CCU will also monitor the DRDY line to see if any ION is transmitting on a certain

channel. If no ION applies this signal, then either the active channel is not in use, or there is a problem with the transmitting ION.

3.1.2 Bus Timing

The timing of the bus transactions is very important because it must ensure that each channel is reliably transmitted from the source ION to the receiving ION, but at the same time ensure that the transactions occur quickly enough to allow a good refresh rate for each channel. Only the CCU controls the timing. The IONs do not have any clock source. Instead, they only contain logic components that react to the signals on the bus with a certain logic delay time. The CCU must therefore ensure that the timing of the bus signals is appropriate for the characteristic delays of the logic components.

Figure 2 shows a timing diagram for one complete bus transaction. All of the above-mentioned signals are shown, as well as the two analog switch controls. The transmitting analog switch connects the outside drive circuit to the bus data wire, and the receiving analog switch connects the bus data wire to the sample and hold circuit.

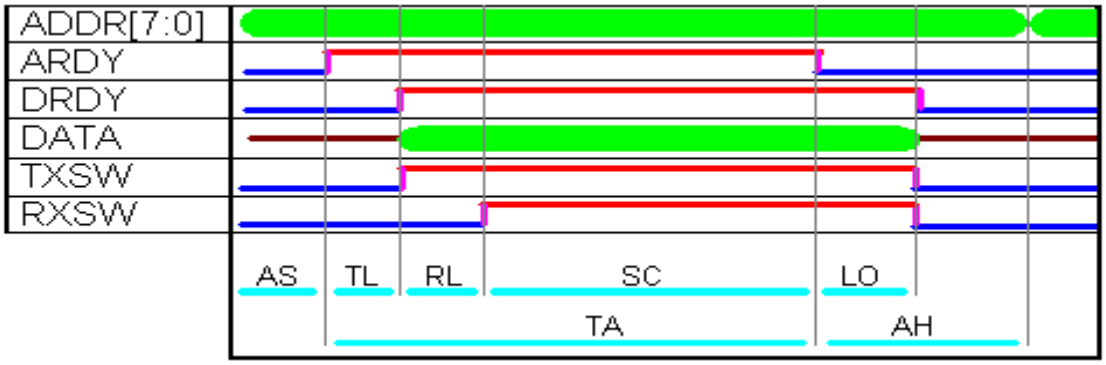


Figure 2 - Bus Timing Diagram

The above timing diagram contains many specific timings. The following list gives a description of each.

- **Address Setup Time** (t_{AS}) t_{AS} is a delay generated by the CCU to allow the address to settle on the address lines before the IONs are allowed to look at it. This is to prevent the IONs from obtaining erroneous readings during address transition.

- **Transmit Logic Time** (t_{TL}) t_{TL} is the time that it takes for the logic on the transmitting node to recognize the address, and to and engage it's analog switch to place it's signal on the bus. During this time, the transmitting node will also raise the DRDY line.

- **Receive Logic Time** (t_{RL}) t_{RL} is the time required for the logic on the receiving node to recognize the combination of the correct address with ADRY and DRDY on the bus. At the end of this time, the receiving analog switch will be engaged to complete the circuit from the source at the transmit node to the sample and hold circuit at the receiving node.

- **Transmit Allowance Time** (t_{TA}) t_{TA} is the time allowed by the CCU for the data transmission to take place. This time is fixed, and so if the transmission fails to occur during this time, it will not occur at all. The time must be fixed to avoid hanging the entire system while waiting for one transmission.

- **Sample Charge Time** (t_{SC}) t_{SC} is the length of time for which the sample and hold circuit on the receiving node is actually connected to the drive circuit on the transmitting node. During this time, the analog switches on both nodes are open. t_{SC} can be mathematically determined by subtracting t_{TL} and t_{RL} from t_{TA}. For this reason, t_{TA} must be made long enough to allow proper charging of the sample and hold circuit after the two logic delay times have elapsed.

- **Logic Off Time** (t_{LO}) t_{LO} is the time required for the analog switches and DRDY signal to be deasserted after the CCU deasserts the ARDY signal. Note that t_{LO} is the combined time for both the transmitting and receiving node to react. This time is very small and can almost be neglected, but we include it for completeness. During this time it is impossible for a false value to reach the sample and hold circuit because as soon as one analog switch is closed, the sample and hold circuit sees a high impedance which locks in its present value. It is not possible for any other node to transmit on the bus because the address lines are stable during this time.

- **Address Hold Time** (t_{AH}) t_{AH} is the delay generated by the CCU to ensure that ADRY is settled to low before the address lines go into transition. This delay ensures that no node sees an ARDY with an ambiguous address value and erroneously begins to transmit or receive.

3.2 External Input Interface

The external input interface gathers analog or digital data from different points in the system and places the data onto the bus. The external input interface is robust in the sense that it must gather data from a wide variety of input devices. For example, in an automobile environment, input devices include power door lock switches, power window switches, headlight switch, horn button, exhaust

gas temperature sensor, oxygen sensor, engine RPM sensor and speedometer sensor. All these input devices can be broken into three main categories:

- Digital inputs (ie. door lock switches)
- Analog inputs (ie. gas temperature sensor)
- Pulsed inputs (ie. engine RPM sensor)

In all of the above three categories, the data is transferred as a time-varying voltage level. The **DC-1*i*** system will have the flexibility to interface all of these types of inputs.

3.2.1 Digital Inputs

For digital inputs, the input signal to the **DC-1*i*** will be a square wave with a value of 0 or 12 volts. This voltage will be sampled and transmitted through the bus whenever the corresponding channel is active. When a transition in the input voltage occurs, the new value will not be transmitted across the bus until the next time the associated channel is active. However, since the **DC-1*i*** can guarantees very fast refresh rates, the delay will be very small. The maximum delay is the update period, which is defined as the reciprocal of the update frequency.

The channel prioritizing of the CCU allows us to specify a minimum refresh rate for each channel, and we now have a way to determine the desired refresh rate. For a power window switch, it may be desired that the maximum delay between pressing the button and the window motor activating is 0.01 sec. From this requirement, we can determine that a refresh rate of 100 Hz is required for that channel. Other, more time-sensitive, applications may require a faster rate, and would be given a higher priority by the CCU.

For many existing electrical systems, switches are implemented in such a way that they provide a specific voltage when engaged, and a high-impedance (open circuit) when disengaged. Since the **DC-1*i*** can only carry voltage signals, the high-impedance state cannot be transmitted. Therefore, a pull up or pull down resistor will be attached to the input to give an alternate voltage for the high impedance state. For example, a switch with an active ground output may have a 100k pull up resistor to 12 volts. When the switch is on, the pull-up resistor will have no effect, but when the switch is off, the voltage will be pulled up to 12V, yielding a digital signal that can be transmitted over the **DC-1*i*** system. These resistors will have negligible power losses and are very inexpensive to implement.

3.2.2 Analog Inputs

For some external circuitry systems, an analog voltage will carry a signal, and the **DC-1*i*** system must be able to carry this type of signal as well. An example of an analog input is a car's oxygen sensor, which expresses the oxygen content of exhaust gasses as an analog signal ranging from 0 to 12 volts.

If the analog voltage is time constant, then the signal will be carried perfectly across the system. If the voltage varies with time, however, then the system's periodic sampling of the input will result in a step-shaped wave that approximates the input signal. Figure 3 illustrates how the tracking process works. The blue line represents the original analog signal, and the green line represents the step-wave approximation.

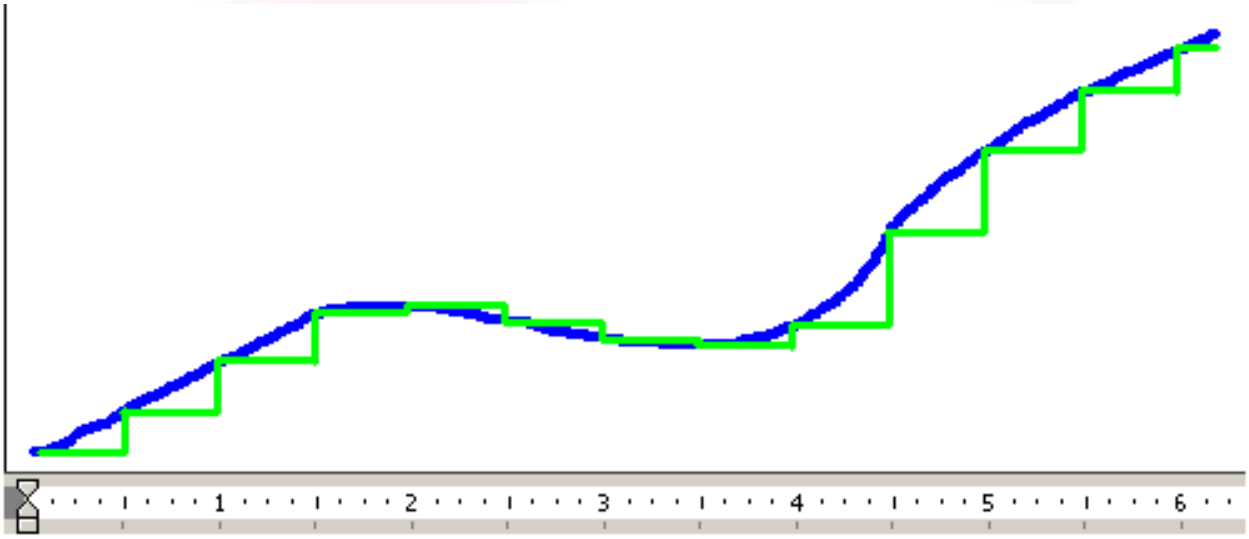


Figure 3 - Step Approximation to Analog Signal

As can be seen, when the voltage level is changing quickly with respect to time, the approximation is less accurate. However, if the sampling rate is increased, the approximation becomes better again. As long as the sampling rate is high in comparison to the frequency of the analog signal, the approximation effects are not noticeable. Therefore, the maximum frequency yields a way to determine the minimum refresh rate for an analog signal. Most analog signals in an automotive application, such as oil pressure, coolant temperature, and throttle, change relatively slowly – perhaps on the order of a few hertz. Therefore, the **DC-1*i*** can easily and accurately carry these signals.

3.2.3 Pulsed Inputs

Some signals, such as engine RPM and groundspeed are transmitted as pulses. A pulsed signal is basically a square wave that goes high for a brief period of time each time an event occurs. For example, a vehicle's speed is measured by a pulse that goes high each time the wheels make one complete revolution.

For the system to carry pulsed signals, the refresh rate of the channel must be high enough to ensure that no pulses are missed. Therefore, the sampling

period must be shorter than the pulse width, so that each pulse is sampled at least once. Usually, the sampling rate will be much higher than this bare minimum to ensure reliability.

3.3 External Output Interface

The external output interface reads analog or digital data from the bus and relays it to different points in the system. The external output interface will have the flexibility to interface to a vast variety of output devices. Returning to the automobile example, typical output devices include door lock solenoids, power window motors, horn, RPM meter and the speedometer. Once again all these output devices can be broken down into the same three categories as above, digital outputs, analog outputs and pulsed outputs. The **DC-1*i*** system will be compatible with all of the output devices discussed, and in addition the **DC-1*i*** system will respond quickly enough as to be transparent to the input and output devices. In other words, the input and output devices will operate as if they were directly connected with a traditional wiring system.

For most outputs, a relatively high current will be required in order to activate the output device, such as a motor or light bulb. In these cases, a simple relay will be used to take the signal from the output ION and provide the appropriate current gain.

3.4 SDU / PC Interface

The SDU / PC Interface will communicate with the host PC through the parallel port using the enhanced parallel port (EPP) protocol. The pins that will be used are listed in Table 1.

Table 1 - Parallel Port Pin Out

D-Type 25 Pin	Signal
1	Strobe
2	SD0
3	SD1
4	SD2
5	SD3
6	SD4
7	SD5
8	SD6
9	SD7
10	ACK
11	WAIT/BUSY
16	RESET

4 System Hardware Design

Here we will discuss the hardware implementations of the various components of the system.

4.1 TTL Logic

Logic implementation is realized with TTL components because TTL is a reliable technology that satisfies the necessary environmental conditions. In addition, TTL components are inexpensive and common, and so sourcing will not be a problem.

Also, using TTL voltage levels will assist in the propagation of signals along the bus, as the higher transmitting voltages of TTL will reduce signal degradation due to the parasitic resistance of the bus lines.

Future plans are to switch to lower power CMOS components, which will further reduce power consumption and are projected to be incorporated into the ENG3 design.

4.2 CCU Hardware

The basic hardware layout of the CCU is quite simple. A PIC16F84A microprocessor is at the heart of the system, with various peripherals complementing it. A block diagram of the CCU is shown in Figure 4.

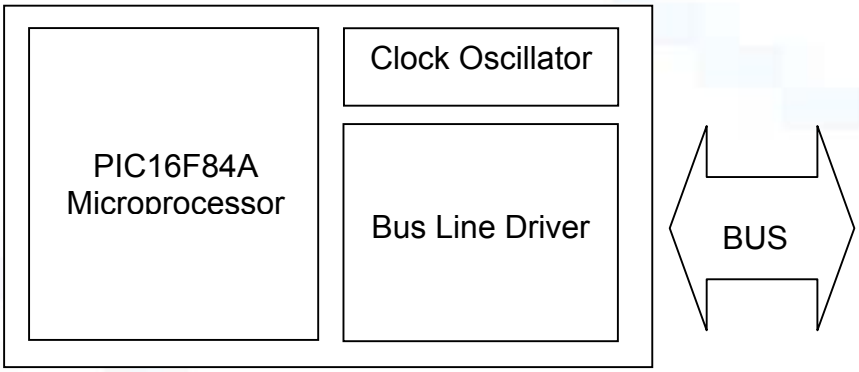


Figure 4 - CCU Block Diagram

4.2.1 Microchip PIC16F84A Microprocessor

The PIC16F84A was selected because it provides the following features:

- Relatively Inexpensive
- Large number of IO pins for parallel address and control signals
- Able to run at speed up to 20 MHz for timing resolution
- Small size (18 PDIP)
- Low power consumption
- Ease of coding and debugging
- ICSP – In Circuit Serial Programming

The PIC microprocessor runs the CCU. It is where all of the firmware for the bus arbitration control is executed. See the firmware section for more details on the responsibilities of the CCU.

4.2.2 Clock Oscillator

The CTS Reeves clock oscillator provides a clean clock signal to the microprocessor. By counting the clock pulses, the CCU can obtain accurate timing resolution necessary for generating the specific bus timings.

4.2.3 Bus Line Driver

The 74ACT245 Octal Bi-directional Transceiver is a buffer that buffers the voltage from the CCU and drives it to the bus. A line driver is used to protect the more sensitive output pin drivers on the CCU from the capacitance that can be seen on a long bus wire and the current that can be drawn by numerous comparators.

4.3 ION Hardware

An Input / Output Node consists of multiple data channels that can be configured for either input or output. A channel consists of one input device and one or more output devices, and is assigned a specific address. This address is used by the CCU to control which channel is using the bus, for example if three devices require the input from the same sensor then one input (address AA) would obtain the input from the sensor and place that information on the bus and three points of output (also address AA) would take the information from the bus and present it to their respective output devices.

The ION has two main functions: the input channel path and the output channel path. Each of these is discussed in detail below.

4.3.1 Channel Input

The input path of the ion takes data from an outside source and places it on the bus. It consists of a comparator, an 8-bit DIP switch and an analog switch. A block diagram of this configuration is shown in Figure 5.

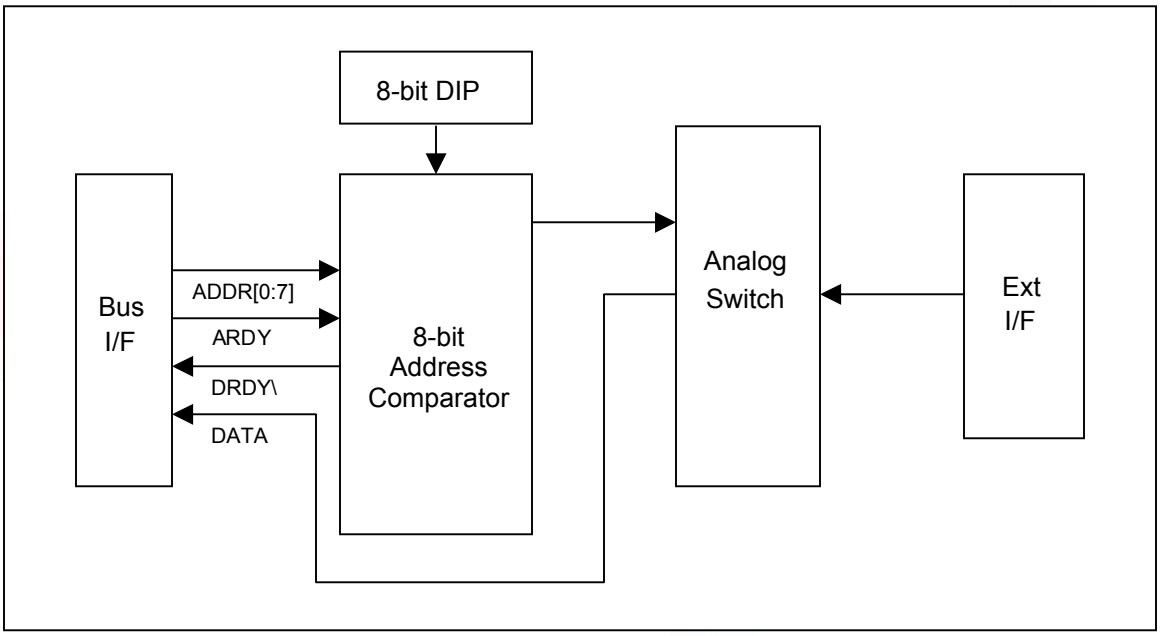


Figure 5: Channel Input Block Diagram

Note: Refer to Appendix A for a detailed schematic of the above block diagram.

4.3.1.1 8-bit DIP Switch

The 8-bit DIP switch allows the user to set the address of the input channel. The address can range from hex 00 (00000000) to hex FF (11111111). Therefore, with the 8-bit implementation, we can potentially have 256 channels.

4.3.1.2 Fairchild 74ACT521 8-bit Identity Comparator

The 8-bit identity comparator compares the address currently on the bus. The comparator enable is sensitive to the ARDY signal. When the address on the bus matches the address configured by the 8-bit DIP switch, the comparator will generate an EN signal to active the analog switch. At the same time, the comparator also generates the DRDY signal, which signals to the output devices of the same channel that valid data is on the bus.

4.3.1.3 Maxim MAX 313 SPST Analog Switch

The analog switch provides a transparent switch that connects the input device to the bus. The analog switch enable EN signal is generated by the comparator

and closes the switch when the address on the bus matches that set by the DIP switch. When the EN signal is asserted, the analog switch will connect the input device to the bus and provide valid data for all the output devices with the same address.

4.3.2 Channel Output

The implementation of a channel output is very similar to that of a channel input. It consists of a comparator, an 8-bit DIP switch, an analog switch, and a sample and hold circuit. A block diagram of this configuration is shown in Figure 6.

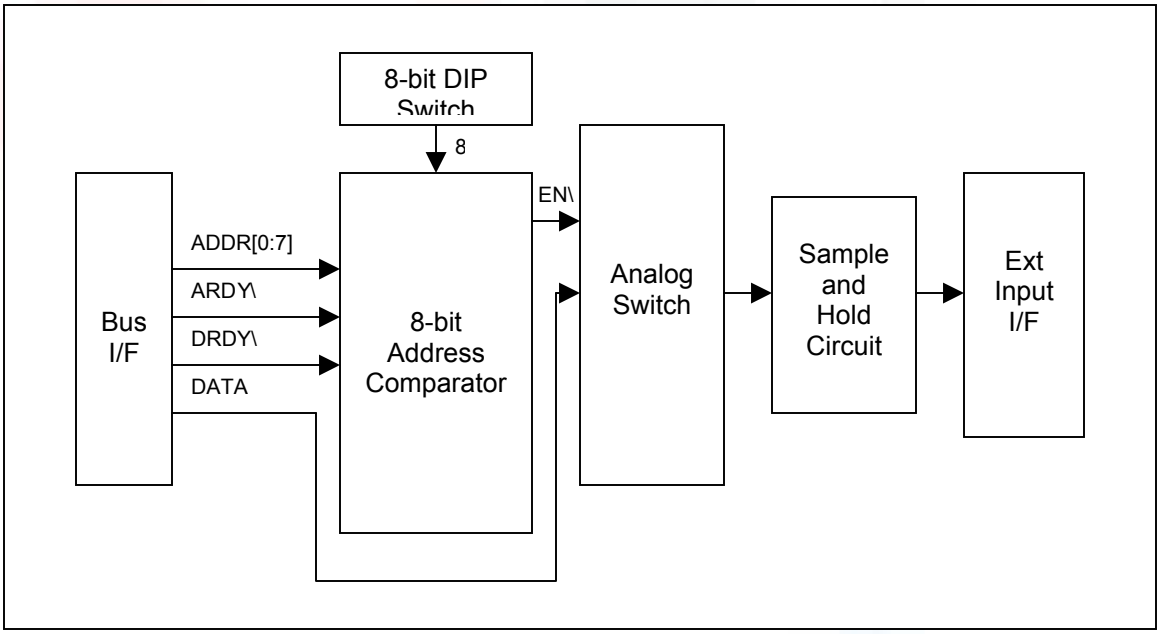


Figure 6: Channel Output Block Diagram

With the exception of the sample and hold circuit and the direction of the DRDY signal (it is now an input to the comparator), the channel output is identical to the channel input. Refer to Appendix A for a detailed schematic of the above block diagram.

4.3.2.1 8-bit DIP Switch

The 8-bit DIP switch allows the user to set the address of the output channel. The address can range from hex 00 (00000000) to hex FF (11111111). WIth the 8-bit address implementation, we can potentially have 256 channels.

4.3.2.2 Fairchild 74ACT521 8-bit Identity Comparator

The 8-bit identity comparator compares the address currently on the bus. On the channel output, the comparator enable is sensitive to the ARDY signal and the DRDY signal. When both ARDY and DRDY are low, and the address on the bus matches the address configured by the 8-bit DIP switch, the comparator will generate an EN signal to active the analog switch.

4.3.2.3 Maxim MAX313 SPST Analog Switch

The analog switch provides a transparent switch that connects the bus to the output device. The analog switch EN (enable) signal is generated by the comparator and closes the switch when the address on the bus matches that set by the DIP switch. With the EN signal asserted, the analog switch will connect the bus to the channel output and provide valid data for all the output devices with the same address.

4.4 SDU Hardware

A block diagram outlining the major components of the System Diagnostic Unit (SDU) is shown in Figure 7.

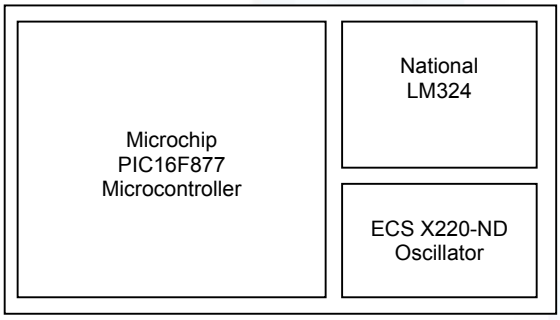


Figure 7 - Block Diagram For SDU

A detailed schematic for the SDU hardware can be found in Appendix A.

4.4.1 Microchip PIC16F877 Microcontroller

The PIC16F877 microcontroller was chosen for the implementation of the SDU because is possesses the following features:

- 8-bit operation
- 20MHz operation
- 40 I/O Pins

- 8192x14 words of FLASH program memory, 256 data memory bytes, and 368 bytes of user RAM
- 8 channel 10-bit A/D converter
- Low power consumption

These features will allow the SDU to gather data from the bus, and at the same time, transmit the data to the PC for diagnostic purposes.

4.4.2 ECS X220-ND Half Size TTL Clock Oscillator

The ECS X220-ND provides the PIC16F877 microcontroller with the necessary clock signal for operation.

4.4.3 National LM324 Operational Amplifier

The LM324 operational amplifier is used to divide the signal voltage into a voltage useable by the A/D converters on the PIC16F877. Since the signal ranges from 0 to 12 Volts, a voltage divider with op amp follower will be used to divide down the signal so that it can be measured by the ADC on the microprocessor, which has a range of 0 to 5 Volts.

4.5 Power Supply Hardware

The **DC-1*i*** system is designed to operate in a wide range of environments. In an automobile environment, the power for the operation of the logic components is derived from the electrical system of the automobile. The electrical system of an automobile operates on a 12V battery, and so the logic components will use this as a power source. However, the logic components operate on 5V power and require that the power supply rail be relatively free of noise. In addition, the power supply must be able to supply sufficient current to power all of the logic devices in the system. The power supply requirements are shown in Table 2.

Table 2 - Power Supply Requirements

Parameter	Value
Input Voltage	12.0 +/- 1.0 V
Output Voltage	5.0 V
Maximum Output Current	2 A
Maximum Noise Ripple	+/- 0.1 V

In order to meet these power requirements, a single power supply will be housed on the CCU. This power supply will convert the 12V power down to a clean 5V power rail. This power rail, called the LOGIC_PWR rail will be distributed to the various IONs of the system via the bus. The power rail will be capacitively coupled to ground at every point of distribution to further reduce the noise on the logic power rail.

4.5.1 Buck Converter

A switching mode power supply configured in the Buck topology is used to step down the 12V supply, with noise from the automobile electrical system, to a clean 5V power rail. A diagram of the buck converter power supply is shown in Figure 8.

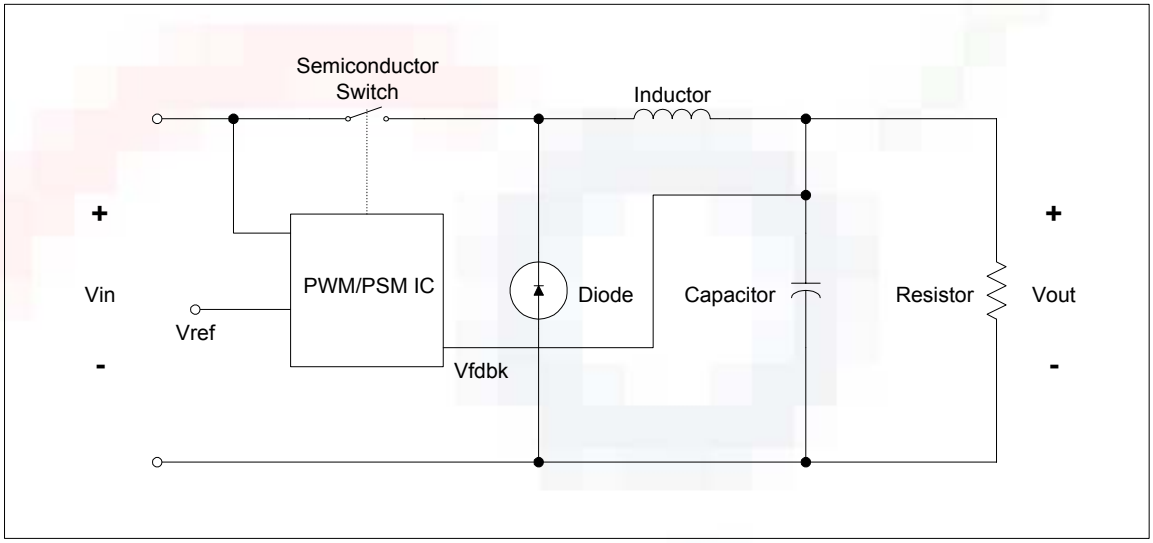


Figure 8: Buck Converter Power Supply

Refer to Appendix A for a detailed schematic of the above block diagram.

4.5.1.1 Maxim MAX727 PWM Switch-Mode DC-DC Regulator

The MAX727 is an integrated circuit that incorporates the semiconductor switch and the Pulse Width Modulation (PWM) controller in one package. PWM is a method of opening and closing the switch in which the frequency of the modulation is held constant while the duty cycle (switch close time relative to switch open time) is varied. The MAX727 will take a 12V input from the electrical system and modulate the switch to obtain a fixed output voltage of 5V.

4.5.1.2 Diode

The diode in Figure 8 provides a path for the current to flow when the switch is open.

4.5.1.3 Inductor

The inductor in Figure 8 smoothes out the ripples in current through the load when the switch is open and closed. Since an inductor opposes changes in current, it will attempt to keep a constant current flowing through the load. When the switch is closed the inductor stores energy, and when the switch is open the

inductor releases this energy. The diode provides a conductive path for the current to flow when the switch is closed. This process reduces the fluctuations in current and hence the voltage drop across the load.

4.5.1.4 Capacitor

The capacitor in Figure 8 smoothes out the ripples in voltage across the load. A capacitor opposes instantaneous changes in voltage that can result when the switch is opened. The stored energy in the charged capacitor will attempt to hold the output voltage constant thus further reducing the voltage fluctuations across the load.

5 System Firmware Design

5.1 CCU

The CCU firmware will run on the PIC16F84A on the CCU unit.

5.1.1 Design Language

The firmware for the SDU will be written in assembly according to the instruction set provided with the Microchip PIC16F84A microcontroller.

5.1.2 Basic Operation

The firmware for the CCU will cycle through the addresses sequentially and allow the bus to transmit each channel in a round-robin fashion. For each address, the appropriate control lines will be asserted at the appropriate times and with the appropriate delays as defined in the system interface section. In addition to generating these control signals, the firmware will monitor the DRDY line to see if it is being asserted by a transmit node. If it is not, then an error condition can be recorded since there is no transmission for that node.

5.1.3 Channel Priority

Since the DC-1i system may carry many different signals, there will be different levels of importance for different channels. For example, a signal relating to engine timing or a braking system would obviously be more important than an air conditioner. Also, some signals need to be updated more often than others. A pulsed signal needs to be refreshed quickly so that it does not miss a pulse, but a great degree of accuracy is not required for a door lock signal as described earlier.

For this reason, the CCU firmware will have different priorities for different channels. The channels will be grouped in ranges based on priority, and the

CCU will modify the round-robin scheme somewhat so that the higher priority channels are serviced more often than the lower priority.

5.1.4 Low Power Mode

Although the DC-1i could easily operate non-stop at full refresh rates for all channels, this is often not necessary. When the car's engine is not running, all signals related to engine management are meaningless. When the key is removed and the vehicle is locked, nearly all signals are not in use except for perhaps the door locks and a few signals relating to the vehicle security system.

If the DC-1i system were running all signals at full refresh rates during these times, power would be needlessly wasted. Also, since the vehicle's battery is not being charged, it is possible that it may be depleted to the point where the car cannot start.

To avoid this, the CCU firmware will include power saving modes. The CCU will have a direct input for whether the ignition is in the OFF, ACC or ON position and will reduce the number of channels that are serviced during the ACC and OFF modes. The channels will be organized in firmware so that the CCU knows which signals are necessary in each state. In addition, the refresh rate will be reduced and the system will momentarily power down between each refresh to further conserve power.

5.2 SDU

The SDU firmware will run on the PIC16F877 on the SDU unit.

5.2.1 Design Language

The firmware for the SDU will be written in assembly according to the instruction set provided with the Microchip PIC16F877 microcontroller.

5.2.2 Channel Data Gathering

Gathering of channel data will be doing during an interrupt sub-routine. This routine will be activated on the falling edge of the data ready line (DRDY). Figure 9 is a high-level flow chart outlining the channel data gathering process.

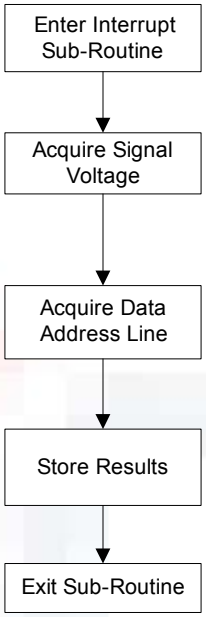


Figure 9 - SDU Channel Data Gathering Flow Chart

5.2.3 SDU / PC Interface

In regular operating mode, the SDU will wait for commands from the PC via the parallel port. When a command is received, the SDU will send the data the PC requested. Figure 10 is a high-level flow chart outlining the PC interface process.

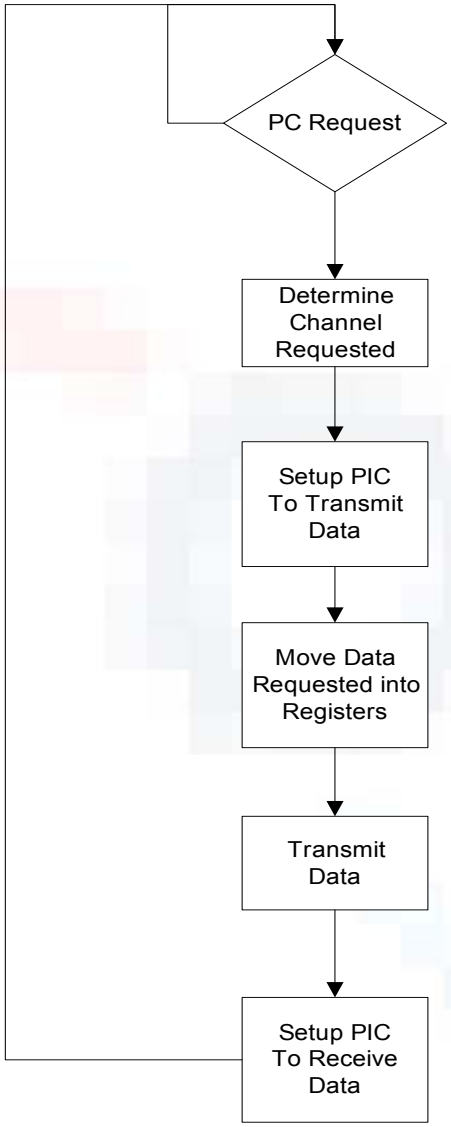


Figure 10 - SDU/PC Interface Flow Chart

6 Application Software Design

The application software is strictly for use with the SDU. It will run on a PC and connect to the SDU via the PC's parallel port. The application software will then display the data captured by the SDU on the PC's screen for a diagnostic engineer to determine any problems that may arise with the DC-1i or any external circuitry.

6.1 Platform

The SDU/PC Diagnostic Software will be written for the use on Microsoft Windows Win32 operating environment. It will be written in Visual C++ to allow hardware access and ease of GUI implementation.

6.2 Features

The SDU/PC Diagnostic Software will have the following features:

- Single channel query
- Channel scan query (scan every channel)
- Plotting of channel data (Voltage vs. Time)

6.3 High Level Implementation

The application will have multiple processes, as listed below:

1. Graphical User Interface
2. Data Gathering
3. Plotting Functions

6.3.1 Graphical User Interface

The graphical user interface will have the following menu structure,

File → **Connect**
Upon selection, PC makes connection to the SDU.

→ **Exit**
Upon selection, software will be terminated.

Display Data → **All Channels**
Start displaying in the widow voltage values of all channels in a tabular form.

→ **Select Channels**
Displays a list of tables where the user can select only the channels they want to display. Once channels are selected they will be displayed in the window in a tabular form.

Plots	→	**Select Channel** *Only one channel can be plotted at a time. Display a list of channels in which the user will choose one. Once channel is selected the data will be plotted on the screen.*
Settings	→	**Refresh Rate** *Sets the rate at which data is read from the SDU.*

6.3.2 Data Gathering

When the user selects the channels he or she wants to monitor, the data gathering process will be started. This process will make a query through the parallel port, gather the information needed, and then store this information in memory. Each channel selected will be scanned once until every channel selected is scanned. This data gathering process is continuously repeated until the user exits the program or changes the channels he or she wants to monitor.

6.3.3 Plotting Functions

When the user chooses the channel that he or she wants to plot, a new process will be launched in which it will read the data from memory and plot it onto an XY Line Plot.

7 Test Plan

The following tests apply to various aspects of the **DC-1i** system. Each has a test number with a two-letter prefix indicating the test type. Note that the test numbers are not necessarily continuous to allow the addition of additional tests as required.

7.1 Hardware Test Plan

Table 3 through Table 6 contain the test cases for the hardware. Tests will be performed on all components of the system.

7.1.1 CCU

Table 3 - CCU HW Tests

Test #	Pass/Fail	Test Name	Test Description

HW11		Connections	Verify correct components and wire connections
HW12		Power	Check LOGIC_PWR is clean 5.0V
HW13		Excessive Heat	Check for excessive heating of components
HW14		Signal Timings	Ensure that appropriate delay timings are being generated to allow proper bus transactions as described in Section 3.1.2 – Bus Timing
HW15		Refresh Rate	Ensure that each channel is receiving the proper refresh rate.
HW16		Channel Priority	Ensure that the channels are being processed in priority sequence
HW17		Low Power Mode	Ensure that minimum current levels are drawn form the battery when low power mod is entered. Also check that designated channels are still functioning at reduces refresh rates.

7.1.2 ION

Table 4 - ION HW Tests

Test #	Pass/Fail	Test Name	Test Description
HW21		Connections	Verify correct components and wire connections
HW22		Power	Check LOGIC_PWR is clean 5.0V
HW23		Excessive Heat	Check for excessive heating of components
HW24		Channel Select	Change DIP switch settings to verify channel selectivity functions correctly
HW25		Comparator	Set input address equal to DIP switch settings, verify that active low output generated when ARDY\ is low and address is valid
HW26		Analog Switch Normally Open	Verify that when active high enable presented to analog switch input the switch is closed otherwise it is normally open
HW27		Analog Switch On Resistance	Measure the on resistance of the analog switch
HW28		Sample and Hold Circuit	Verify that the sample and hold circuit will hold the output signal for a sufficient

			amount of time and ensure that the value at the channel output is identical to the value at the channel input

7.1.3 SDU

Table 5 - SDU HW Tests

Test #	Pass/Fail	Test Name	Test Description
HW31		Connections	Verify correct components and wire connections
HW32		Voltages	Compare voltages measured with DMM to voltages on screen
HW33		Graph	Input a sine wave to SDU and compare measure with oscilloscope and compare waveform to graph on screen.

7.1.4 Power Supply

Table 6 - Power Supply HW Tests

Test #	Pass/Fail	Test Name	Test Description
HW41		Connections	Verify correct components and wire connections
HW42		Voltage Level	Check power supply output is a clean 5.0V
HW43		Noise/Ripple	Check that noise/ripple are sufficiently small
HW44		Output power w/ 100Ω load	Check output voltage and output current, also check for excessive heating
HW45		Output power w/ 50Ω load	Check output voltage and output current, also check for excessive heating
HW46		Output power w/ 10Ω load	Check output voltage and output current, also check for excessive heating
HW47		Output power w/ 5Ω load	Check output voltage and output current, also check for excessive heating
HW48		Output power w/ 3Ω load	Check output voltage and output current, also check for excessive heating
HW49		Black Out	Vary the input voltage in 0.5V (0.1V around 12V) decrements from 20V down to 0V, monitor output
HW50		Brown Out	Vary the input voltage in 0.5V (0.1V around 12V) increments from 0V up to 20V, monitor output
HW51		Recovery From	Vary the input voltage from 12V to 0V

		Reset	then back up to 12V

7.2 Interface Test Plan

The interface tests ensure that the components interact properly with each other. They will focus on the reliable transmission of data between different components of the system. These tests will be performed after the components are individually tested.

7.2.1 CCU / ION Interface Testing

Table 7 - CCU-ION Interface Tests

Test #	Pass/Fail	Test Name	Test Description
IF11		Delays	Measure the logic and switching delays specified in Section 3.1.2 – Bus Timing. Ensure that t_{SC} is sufficient for the sample and hold circuit.
IF12		Propagation	Ensure that there are no problems relating to the signals propagation along the bus lines (ringing, etc.)
IF13		Failure Point	Test redundant looping bus mechanism by disconnecting cable at one point

7.2.2 External Input Interface Testing

Table 8 - External Input Interface Tests

Test #	Pass/Fail	Test Name	Test Description
IF21		Digital Inputs	12V to 0 V digital input, observe transient response
IF22		Analog Inputs	Sinusoidal input of varying frequency and amplitude
IF23		Switching Inputs	High frequency square wave input

7.2.3 External Output Interface Testing

Table 9 - External Output Interface Tests

Test #	Pass/Fail	Test Name	Test Description
IF31		Digital Outputs	12V to 0 V digital output, observe transient response
IF32		Analog Outputs	Sinusoidal output of varying frequency and amplitude

IF33		Switching Outputs	High frequency square wave output

7.2.4 SDU / PC Interface Testing

A stress test will be performed on the PC interface to make sure the system is stable. Once stable, the interface testing will continue by changing values on the hardware and ensuring they are successfully propagated to the screen.

7.3 Firmware Test Plan

The test plan for firmware will be closely tied to the system interfaces and if the specific test cases for the system interface pass, then the firmware will be considered to pass. To ensure robustness and reliability, code inspection with respect to general firmware design guidelines will be applied by members of the team other than the code's author.

7.3.1 CCU Firmware

If the CCU handles the bus arbitration properly, then the firmware will be considered to pass. In addition, there are some specific criteria for the CCU firmware, listed in Table 10.

Table 10 - CCU FW Tests

Test #	Pass/Fail	Test Name	Test Description
FW11		Reset	CCU comes out of reset and immediately begins cycling through channels
FW12		Missing Node	The CCU will continue polling all channels if a node is malfunctioning or not present.

7.3.2 SDU Firmware

The SDU firmware will be tested with the simulation tools and the in-circuit debugging tools. Each function of the firmware will be tested - data acquisition and data transmission. Table 11 lists the specific tests that will be performed.

Table 11 - SDU FW Tests

Test #	Pass/Fail	Test Name	Test Description
FW21		Reset	SDU comes out of reset and immediately begins cycling through channels
FW22		Data Acquisition	The SDU will continuously scan the address line and signal line as long it is in.
FW23		Data	The SDU will transmit data only if

		Transmission	requested so by the PC Interface.

7.4 Software Test Plan

The complexity of the SDU / PC Interface software is minimal. Testing of the software will focus on stability, error handling, and functionality. These tests are described in Table 12.

Table 12 - Software Tests

Test #	Pass/Fail	Test Name	Test Description
SW11		Stability	The software will be stress tested and subjected to extreme conditions to test for stability.
SW12		Error Handling	The software will be given errors to see how well it recovers and handles the errors.
SW13		Data Transmission	The connection between the PC and SDU will be terminated physically to see if the software handles a break in data transmission.

8 Conclusion

This document has outlined the design specification for prototyping the DC-1i. Based on these specifications, we will continue to develop our technology.

We are committed to fully implement the designs contained within this document by December 10, 2001. This design specification will provide us with a clear path to the completion our project.

Our proposal has been well planned and researched. Our goals are clearly defined, and we now have a roadmap with a clear view to the completion of our project. Our team has proven leadership, extensive technical experience as well as the motivation and funding to meet our commitments.

DC Integration Innovations is poised to revolutionize DC wiring forever.

DC Integration Innovations

9 References

- Jacques Vaisey – frequency considerations
- Patrick Leung – microprocessing requirements
- Fred Heep – component selection, component sourcing
- Coleman Cable, www.colemancable.com
- Digikey Canada, www.digikey.ca
- Maxim Integrated Products, Sunnyvale, CA

10 Technical Appendices

TTL Standard Logic Levels
http://www.twysted-pair.com/74xx.htm

Interfacing the Standard Parallel Port
http://www.beyondlogic.org/spp/parallel.pdf

11 Appendix A – Schematics

The following schematics are attached to provide additional detail for our hardware design. Please find them in the following pages.

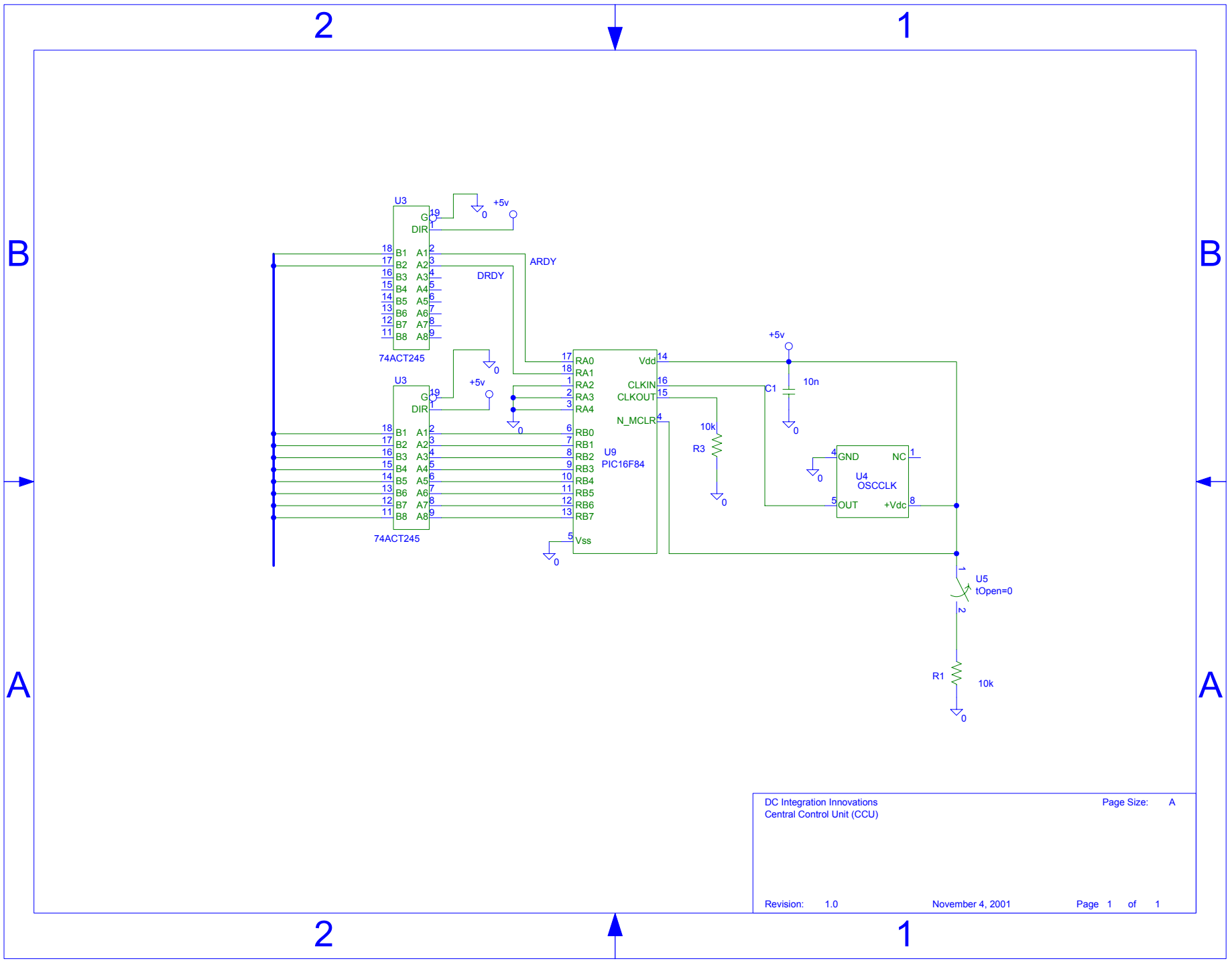
11.1 CCU Schematic

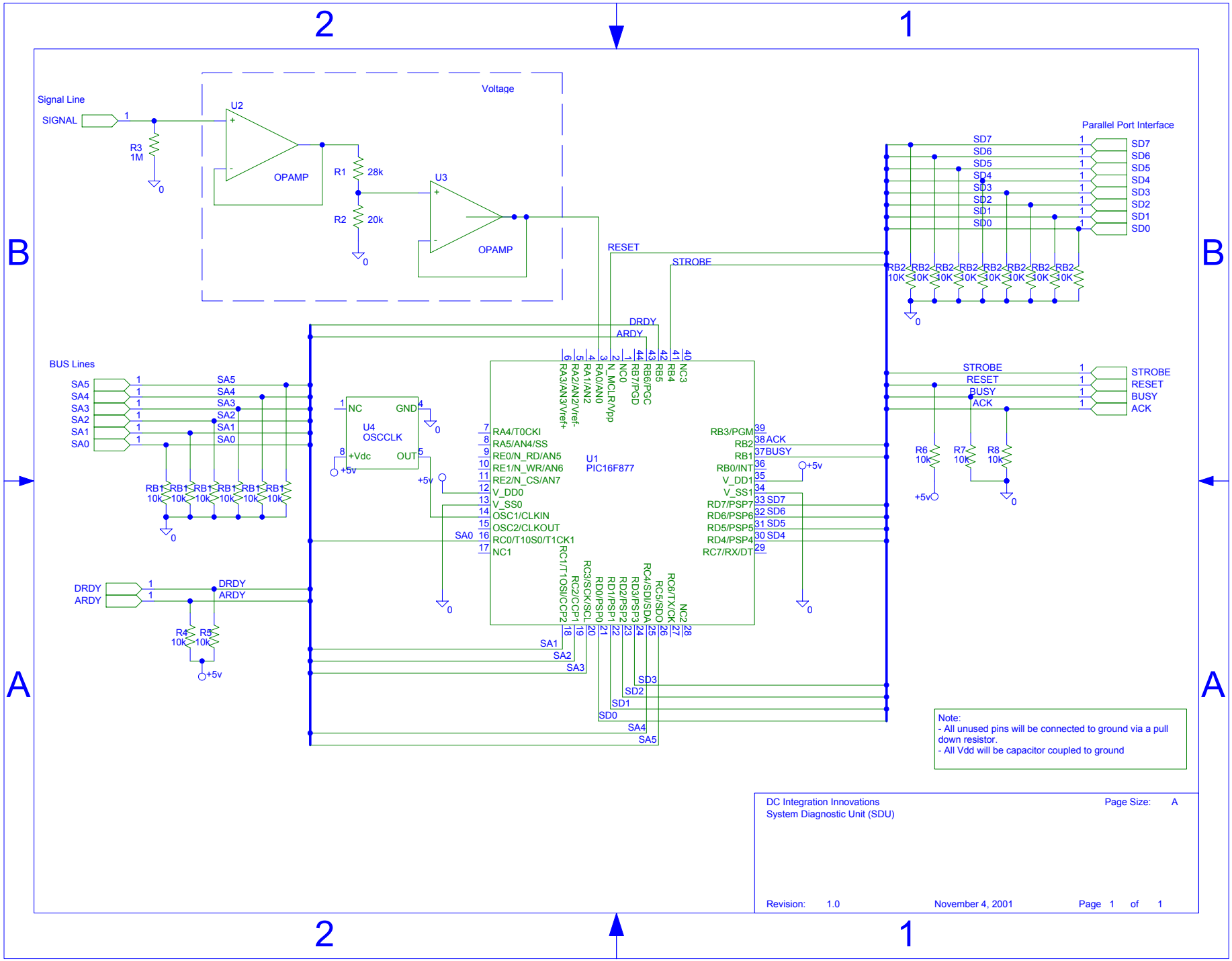
11.2 ION (Input) Schematic

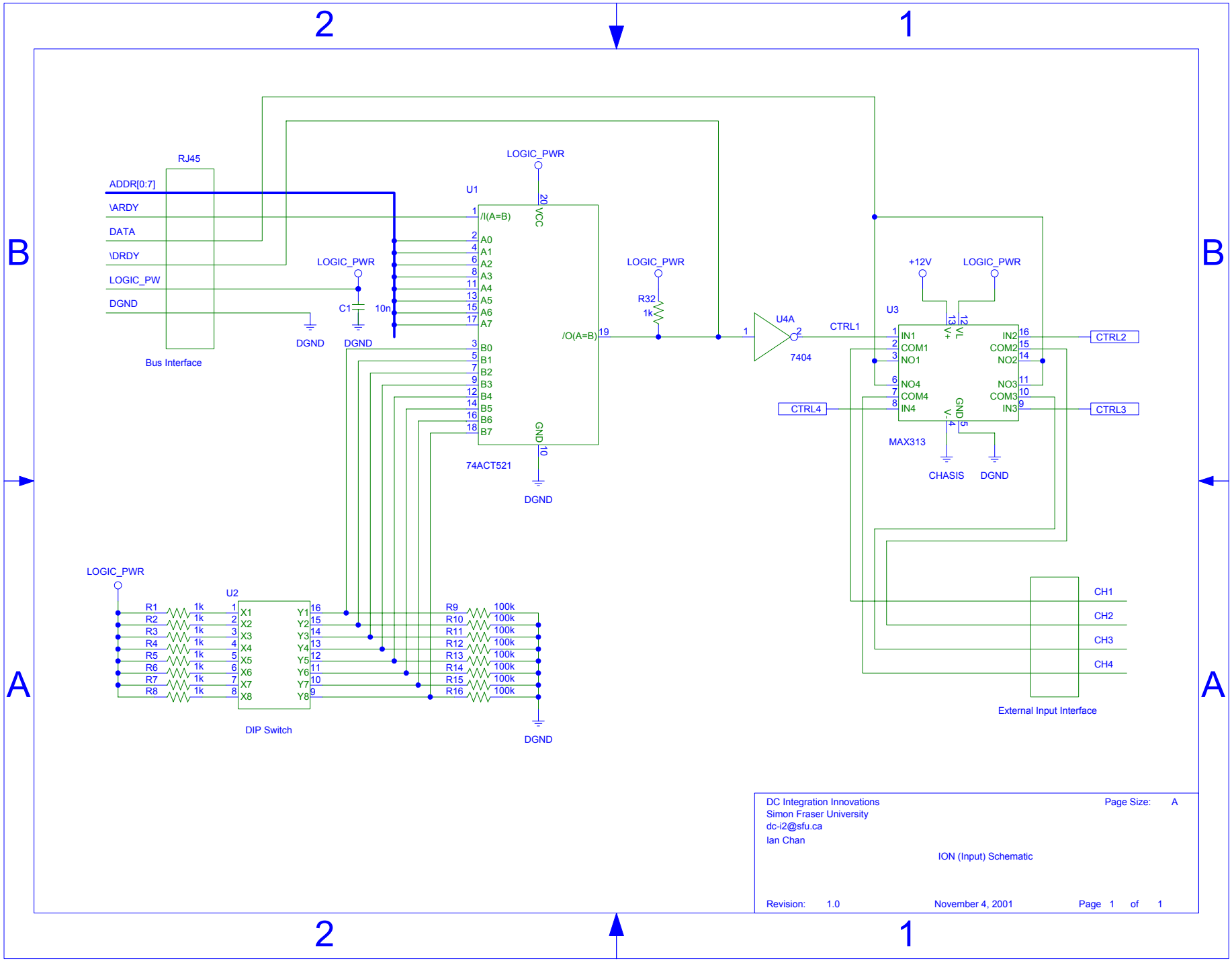
11.3 ION (Output) Schematic

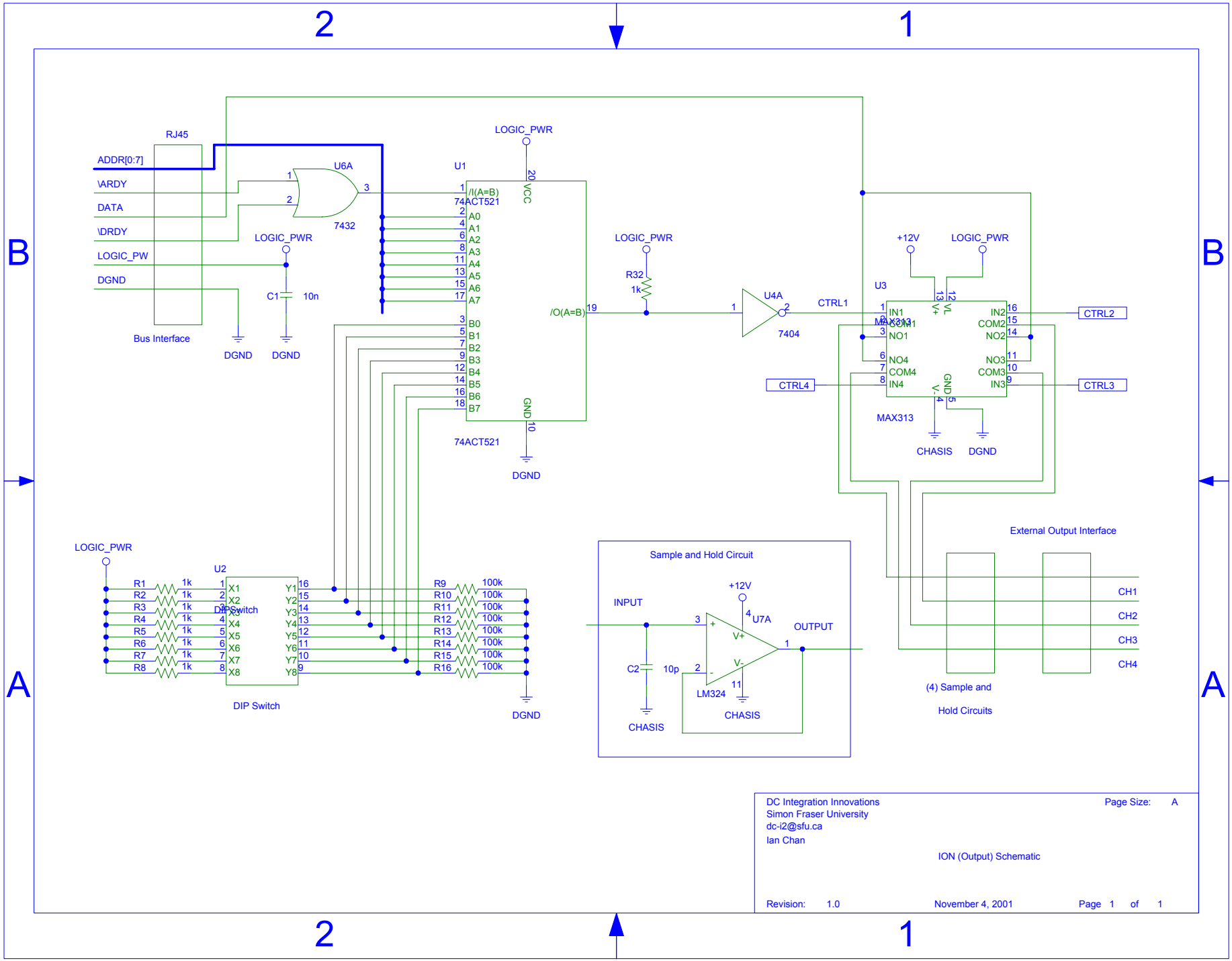
11.4 SDU Schematic

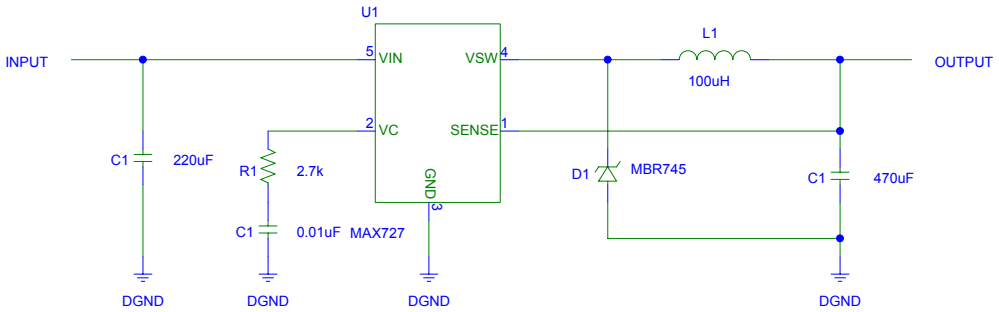
11.5 Power Supply Schematic


2
1
B
A
U3
74ACT245
U3
74ACT245
+5v
ARDY
DRDY
U9
PIC16F84
RA0
RA1
RA2
RA3
RA4
N_MCLR
RB0
RB1
RB2
RB3
RB4
RB5
RB6
RB7
Vdd
Vss
CLKIN
CLKOUT
10k
R3
C1
10n
U4
OSCCLK
GND
NC
OUT
+Vdc
U5
tOpen=0
R1
10k
DC Integration Innovations
Central Control Unit (CCU)
Page Size: A
Revision: 1.0
November 4, 2001
Page 1 of 1


Signal Line
SIGNAL
R3
1M
U2
OPAMP
Voltage
R1
28k
R2
20k
U3
OPAMP
RESET
STROBE
DRDY
ARDY
Parallel Port Interface
SD7
SD6
SD5
SD4
SD3
SD2
SD1
SD0
RB2
10K
BUS Lines
SA5
SA4
SA3
SA2
SA1
SA0
RB1
10k
U4
OSCCLK
NC
GND
+Vdc
OUT
+5v
U1
PIC16F877
RA4/T0CKI
RA5/AN4/SS
RE0/N_RD/AN5
RE1/N_WR/AN6
RE2/N_CS/AN7
V_DD0
V_SS0
OSC1/CLKIN
OSC2/CLKOUT
RC0/T10S0/T1CK1
NC1
RA3/AN3/Vref+
RA2/AN2/Vref-
RA1/AN2
RA0/AN0
N_MCLR/Vpp
NC0
RB7/PGD
RB6/PGC
RB5
RB4
NC3
RB3/PGM
RB2
RB1
RB0/INT
V_DD1
V_SS1
RD7/PSP7
RD6/PSP6
RD5/PSP5
RD4/PSP4
RC7/RX/DT
ACK
BUSY
STROBE
RESET
ACK
R6
10k
R7
10k
R8
10k
DRDY
ARDY
R4
10k
R5
10k
+5v
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
NC2
SA1
SA2
SA3
SD3
SD2
SD1
SD0
SA4
SA5
Note:
- All unused pins will be connected to ground via a pull down resistor.
- All Vdd will be capacitor coupled to ground
DC Integration Innovations
System Diagnostic Unit (SDU)
Page Size: A
Revision: 1.0
November 4, 2001
Page 1 of 1


RJ45
ADDR[0:7]
\ARDY
DATA
\DRDY
LOGIC_PW
DGND
Bus Interface
LOGIC_PWR
C1
10n
DGND
DGND
U1
/I(A=B)
VCC
20
A0
A1
A2
A3
A4
A5
A6
A7
B0
B1
B2
B3
B4
B5
B6
B7
GND
10
74ACT521
DGND
/O(A=B)
19
LOGIC_PWR
R32
1k
1
U4A
2
7404
CTRL1
+12V
LOGIC_PWR
U3
IN1
COM1
NO1
NO4
COM4
IN4
IN2
COM2
NO2
NO3
COM3
IN3
V+
VL
V-
GND
MAX313
CHASIS
DGND
CTRL2
CTRL3
CTRL4
LOGIC_PWR
U2
R1
R2
R3
R4
R5
R6
R7
R8
1k
X1
X2
X3
X4
X5
X6
X7
X8
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y8
R9
R10
R11
R12
R13
R14
R15
R16
100k
DIP Switch
DGND
CH1
CH2
CH3
CH4
External Input Interface
DC Integration Innovations
Simon Fraser University
dc-i2@sfu.ca
Ian Chan
Page Size: A
ION (Input) Schematic
Revision: 1.0
November 4, 2001
Page 1 of 1


RJ45
ADDR[0:7]
\ARDY
DATA
\DRDY
LOGIC_PW
DGND
Bus Interface
U6A
7432
LOGIC_PWR
C1
10n
DGND
U1
74ACT521
/I(A=B)
VCC
A0
A1
A2
A3
A4
A5
A6
A7
B0
B1
B2
B3
B4
B5
B6
B7
GND
/O(A=B)
R32
1k
U4A
7404
CTRL1
U3
MAX313
IN1
IN2
COM1
COM2
NO1
NO2
NO4
NO3
COM4
COM3
IN4
IN3
V+
V-
+12V
CTRL2
CTRL3
CTRL4
CHASIS
LOGIC_PWR
U2
DIP Switch
R1
R2
R3
R4
R5
R6
R7
R8
1k
X1
X2
X3
X4
X5
X6
X7
X8
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y8
R9
R10
R11
R12
R13
R14
R15
R16
100k
Sample and Hold Circuit
INPUT
OUTPUT
U7A
LM324
C2
10p
External Output Interface
CH1
CH2
CH3
CH4
(4) Sample and
Hold Circuits
DC Integration Innovations
Simon Fraser University
dc-i2@sfu.ca
Ian Chan
Page Size: A
ION (Output) Schematic
Revision: 1.0
November 4, 2001
Page 1 of 1



DC Integration Innovations
Simon Fraser University
dc-i2@sfu.ca
Ian Chan

Page Size: A

Power Supply Schematic

Revision: 1.0 November 4, 2001 Page 1 of 1

D1N750