EE 552

The S.L.G. Project

Final Documentation

2003-04-01

Group Members:
Chris Lee : ccl1@ualberta.ca
Dave Yeung : dkyeung@ualberta.ca
Michael Mah : mwmah@ualberta.ca
Jaffer Kapasi : jkapasi@ualberta.ca

Declaration of Original Content

The design elements of this project and report are entirely the original work of the authors and have not been submitted for credit in any other course except as follows:

- clk_divider.vhd code taken from reference [11]
- LCDdriver.vhd was modified from reference [9]
- LCD control algorithm was taken from manufacturer's spec sheet. Reference [12]
- LCD schematic taken from manufacturer's spec sheet. Reference [13]
- ADC schematic taken from manufacturer's spec sheet.
- ADC control algorithm taken from manufacturer's spec sheet.
- ad_converter.vhd was modified from reference [14]

Dave Yeung: ________________________

Michael Mah: ________________________

Jaffer Kapasi: ________________________

Chris Lee: ________________________

Abstract

Sign Language Glove: American Sign Language Translation Device

This project is the design and implementation of a portable, lightweight, glove-mounted sign language translation device. Previous implementations of sign language devices require the use of video and ultrasonic detection techniques, which are inherently bulky and therefore impractical for hearing/speech-impaired individuals to carry around. This implementation utilizes a conventional glove on which flex sensors are arranged to collect hand motion data. A Field Programmable Gate Array (FPGA) will be used to translate electrical signals produced from the glove into text, and a small Liquid Crystal Display (LCD) monitor will display the text. An efficient hardware implementation on an FPGA will be used rather than a software-based approach. The FPGA stores the binary representation of each letter in the ASL alphabet, in memory, where it can be compared to digitized output from the glove (by analog-to-digital conversion (ADC)). As a preliminary design, the scope of the device's communication abilities will be simplified to the American Sign Language (ASL) finger alphabet, and will therefore require spelling the words. The purpose of the SLG is to break down communication barriers between speech-impeded people and the general public. Its uses could also be extended to mobile keyboards, covert communication, or an intermediate to synthesized speech conversion.

Table Of Contents

Achievements

Overall, the goals, which were set for the SLG project have been achieved. These will be quantified into some of the more important accomplishments, which our team has made.

Overall Functionality

- The glove was able to convert sign language into the correct letter with a high accuracy rate of ~90 %. Details of this characterization is in the Results of Experimentation Section.

Physical Glove

- Overcame problems with paper clip based contact sensors (breakage of sensor wiring and insufficient contact) by coiling more flexible wire around the thumb and middle fingers. These coiled wires were made perpendicular to each other, allowing better contact at crisscross points.
- Utilized fabric pockets fastened by electrical tape to hold the flex sensor securely to the glove, while allowing enough slack for the flex sensors to bend.
- Attached sixth flex sensor under the wrist, allowing the measurement of wrist rotation utilized in the letters 'I 'and 'J'.
- Soldered long ribbon cable to the glove wiring. This allows the full extension of the arm while wearing the glove.

Analog To Digital Converter

- Implemented the ADC to convert analog flex sensor voltages into digital values

LCD Component

- Implemented the LCD screen to correctly display the outputs from the SLG program

Training Program

- Created training mode for SLG device allowing the unit to be customized to the particularly signing of an individual. Learns the signing techniques for each letter and overwrites the default values in RAM.

Translation Accuracy

- Experimentation with the fully working SLG system revealed sign translation accuracy of 90%.
- The accuracy was improved from previous measurements by removing from comparison immediately letters whose flex values differed by more than 0.7 V. This was in addition to an algorithm, which picked the correct letter by calculating deviation from the currently measured flex sensor values.

Memory Component

- The memory implementation utilized was LPM_ram. Flash memory had been a favorite previously, however it was found that LPM_ram offered the same features with a much simpler implementation. LPM_ram allowed us to load default values of flex/contact sensors for each letter of the alphabet and to customize them on power up.

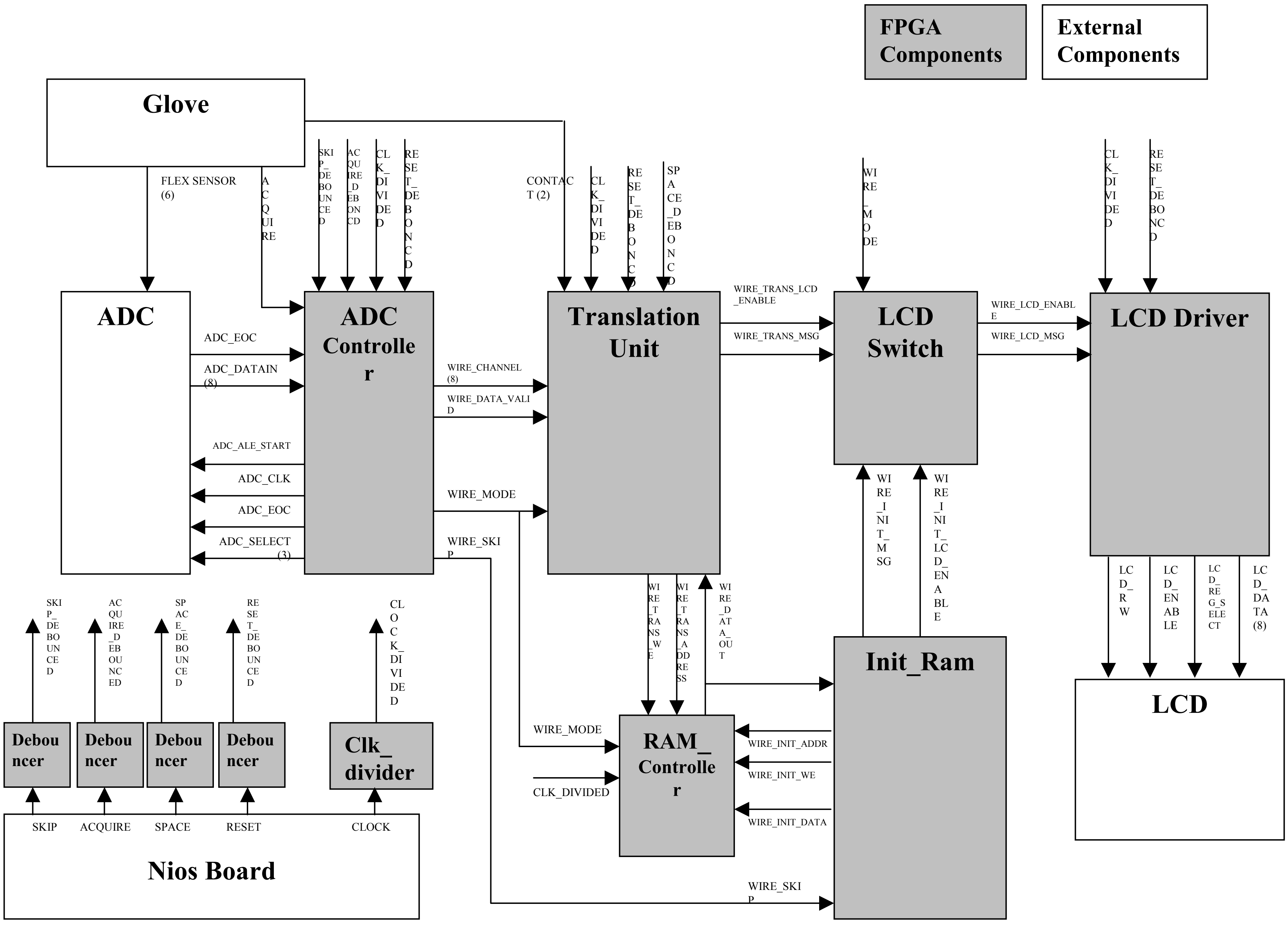
Power Supply

- Power supply for the Glove circuitry was lowered to 5V from 9V.
- Lower power better for the FPGA/ADC circuitry.
- Still allows for an optimal spacing between the 256 levels of voltage (and therefore levels of flex) of the sign language glove.

Software Achievements

The following table describes the SLG software accomplishments. Description of the software components are available in section Description of Operation.

Software Component	**Achievements**
SLG • clk_divider.vhd • debouncer.vhd	Code complete. Simulated. Hardware implemented.
ADC • ad_converter.vhd • nregister.vhd • en_rigister.vhd • multiplexer.vhd	Code complete. Simulated. Hardware implemented.
LCD • lcddriver.vhd • lcd_switch.vhd	Code complete. Simulated. Hardware implemented.
RAM • ram_controller.vhd • lpm_ram.dq	Code complete. Simulated. Hardware implemented.
Training Unit • init_ram.vhd	Code complete. Simulated. Hardware implemented.
Translation Unit • translation_unit.vhd	Code complete. Simulated. Hardware implemented.


FPGA Components
External Components
Glove
FLEX SENSOR (6)
ACQUIRE
CONTACT (2)
ADC
ADC Controller
Translation Unit
LCD Switch
LCD Driver
ADC_EOC
ADC_DATAIN (8)
ADC_ALE_START
ADC_CLK
ADC_EOC
ADC_SELECT (3)
WIRE_CHANNEL (8)
WIRE_DATA_VALID
WIRE_MODE
WIRE_SKIP
WIRE_TRANS_LCD_ENABLE
WIRE_TRANS_MSG
WIRE_LCD_ENABLE
WIRE_LCD_MSG
WIRE_MODE
WIRE_INIT_MSG
WIRE_INIT_LCD_ENABLE
WIRE_TRANS_WE
WIRE_TRANS_ADDRESS
WIRE_DATA_OUT
Init_Ram
RAM_Controller
WIRE_MODE
CLK_DIVIDED
WIRE_INIT_ADDR
WIRE_INIT_WE
WIRE_INIT_DATA
WIRE_SKIP
LCD
LCD_RW
LCD_ENABLE
LCD_REG_SELECT
LCD_DATA (8)
Debouncer
Debouncer
Debouncer
Debouncer
Clk_divider
SKIP_DEBOUNCED
ACQUIRE_DEBOUNCED
SPACE_DEBOUNCED
RESET_DEBOUNCED
CLOCK_DIVIDED
SKIP
ACQUIRE
SPACE
RESET
CLOCK
Nios Board

Description of Operation

Power Supply

A 5 V power supply is used to power all necessary components (besides the FPGA).

Clock and Reset

A 500 kHz clock drives the SLG because of constraints imposed by its components (ADC in particular). The SLG is implemented using an APEX20KE EP20K200EFC484-2X FPGA, which has on onboard clock of 33 MHz. Thus a clock divider will be used to divide the clock to 500 kHz.

The SLG has a synchronous reset, which resets the entire system.

Sports Glove

The purpose of the glove is to acquire data related to the position of the user's hand. By signing a certain letter, sensors on the glove (flex sensors and contact sensors) obtain certain values. Flex sensors are variable resistance devices that change resistance when bent. According to the manufacturer's specifications, the flex sensors have a resistance range of 10 – 40 kΩ and in order to differentiate the different hand positions, a total of 6 flex sensors are used. Contact sensors are basically switches creating a closed circuit when the two ends are in contact.

A schematic of the glove is included in Appendix A.

A table summarizing how the letters of the alphabet are differentiated is included in the Results of Experimentation and Characterization section.

Analog-to-Digital Conversion

The SLG design requires that the output of the glove be converted to a digital signal in order to be processed and translated. The ADC0809 is chosen in the design for this purpose. This ADC allows for 8 analog inputs (it has 8 channels), which is sufficient for the 6 flex sensor inputs. Refer to Appendix A for datasheet, schematics and timing diagram.

The Data processing Unit coordinates the operation of the analog to digital converter and the translation unit. The top-level component is called adconverter.vhd and is based on the code of Shauna Rae, extended to utilize 6 channels of this particular ADC. It runs in a cycle in which all six flex sensor voltages are converted to digitized values. This cycle is triggered to begin by a press of the "acquire" or "skip" push buttons and the converted values are stored in one of six multiplexer-enabled registers corresponding to the channel converted. Once conversion is completed for all six channels, a data_valid flag is pulsed high for one clock cycle in order to communicate with other related components.

The circuit works in the following manner:

1) If the reset button is pressed all values of signal flags and control signals are cleared.

2) Conversion starts if either the enable (acquire) or skip buttons are pressed. The skip button only responds if the system is still in the initialization mode. The initialization mode exists for 26 conversions after a reset press and allows user calibration of the sign language glove for all letters of the alphabet. If skip is pressed the calibration of the current letter is skipped and default values are held.

3) There are 2 parts to conversion: Setting the adc_loadAddressStart signal (ALE and Start control signals tied together) and waiting for an A/D conversion to complete.

4) The completion of an A/D conversion is ascertained when the adc_dataReady signal is pulled up by the ADC (EOC signal). In this case a process is triggered, which toggles the flag_wait flag. When the channel select signal, sel, is odd, flag_wait = 0 will allow the register corresponding to that channel to be enabled, when even, flag_wait = 1 will allow the register corresponding to that channel to be enabled. When this register is enabled, the channel select signal is incremented to begin conversion of the next flex sensor.

5) Since the conversion process begins by checking the level of the enable and/or skip buttons, and is not edge triggered, there must be a means to make sure only one conversion cycle is completed per push of a button, otherwise an effect resembling a non-debounced button will occur. This job is handled by the acq_flag which goes high when the data_valid signal is pulsed (to stop another cycle from taking place) and goes low when the enable and skip buttons are high (button is back up and a new cycle will only be triggered by another button push).

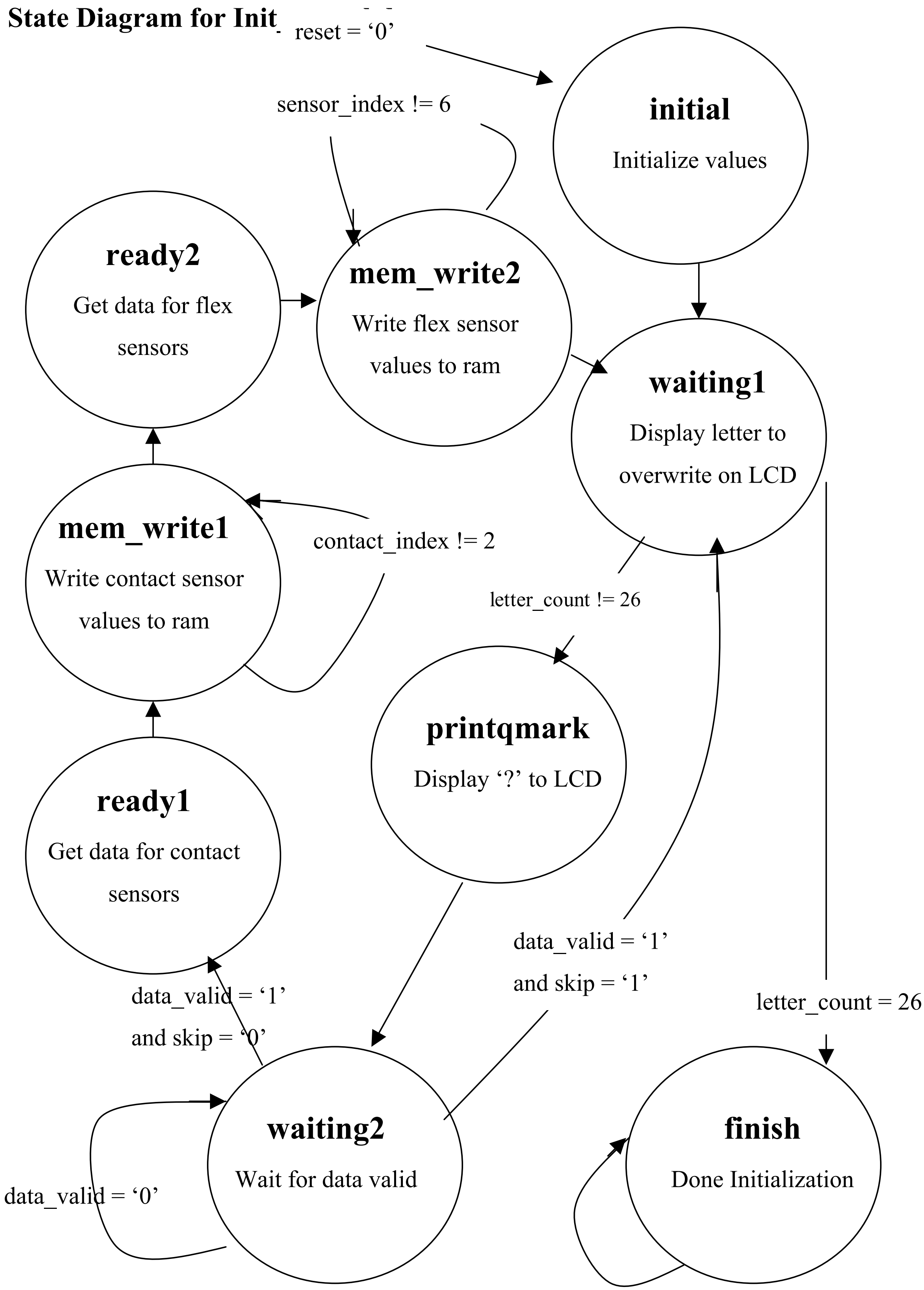
Enable Switch

In order to capture a glove position correctly, an enable switch is used. When a user has signed a letter, setting the enable button starts the actual analog-to-digital conversion. This prevents any transitions caused by changing one position to another to be translated. The enable switch is separate from the glove and is to be operated by the hand not doing the signing.

RAM Initialization

At startup, 512 bytes of RAM are initialized to contain the initial sensor values. These values are summarized in the Results of Experiments and Characterization section. Since these initial values may vary from person to person, at startup, the user may override the initial values. In order to implement this feature, the first 26 button pushes (after startup or reset) are preset to allow the user to change the sensor values. The user must manually input the values by signing the letter displayed on the LCD (‘A’ – ‘Z’) and pressing the “enable” push button. If the user does not wish to override a certain letter, a “skip” push button may be pushed. After the first 26 button pushes, the SLG resumes normal operation.

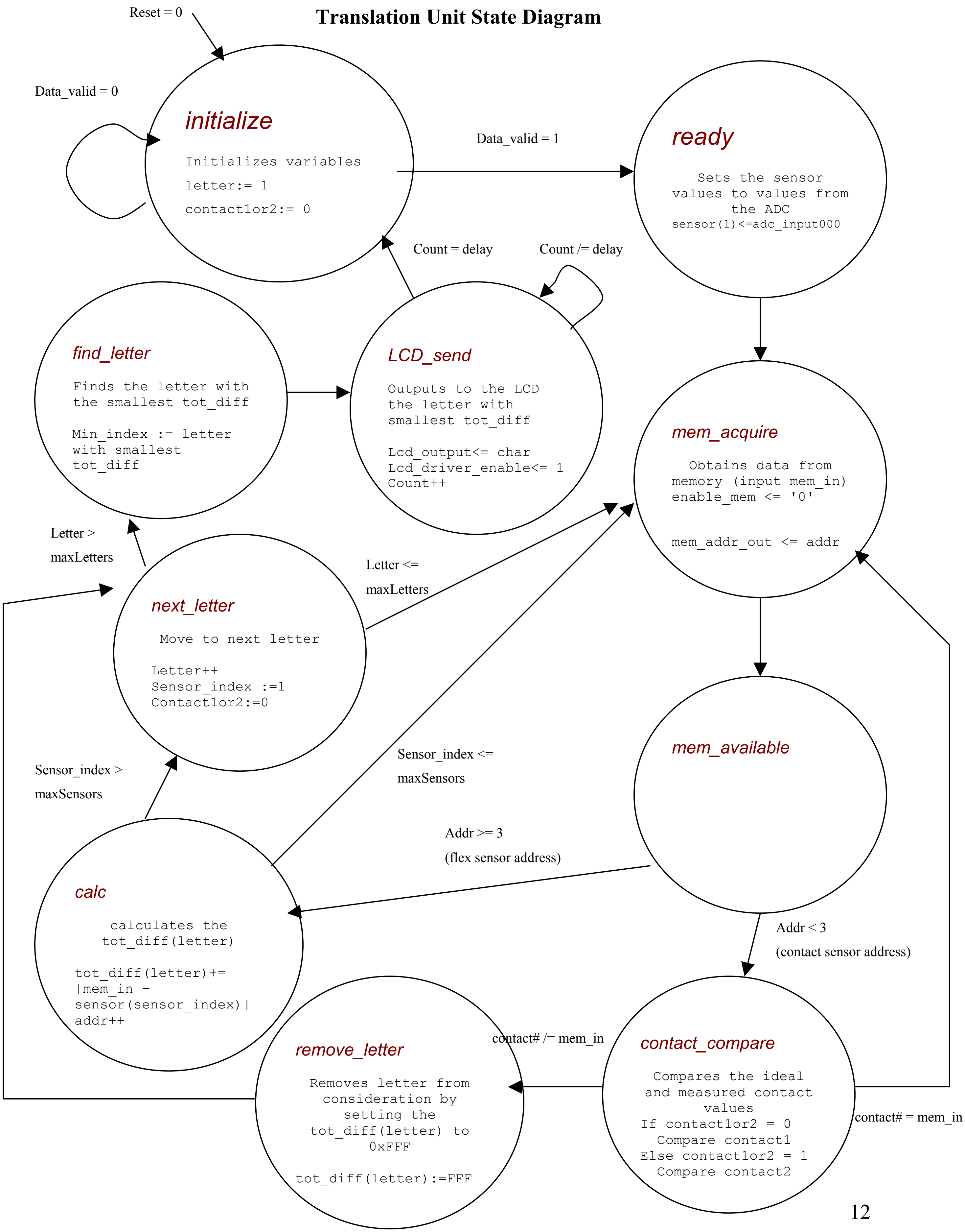
State Diagram for Init



Translator

The purpose of the translation is to interpret the signals from the glove into a letter to be displayed on the LCD. It does this by taking the 'measured' glove sensor values from the ADC and then compares it with a set of 'ideal' glove sensor values in memory. Each letter has its own set of ideal flex sensor values stored in memory. The absolute difference between the ideal values and the measured values is calculated and accumulated for each sensor and for each letter. The letter with the lowest total difference is chosen to be 'the letter'. In addition, if the measured contact sensor values (either 1 or 0) do not match with those in memory for a letter, that letter is removed from consideration. The translation unit then outputs to the LCD controller which letter to display on the LCD.

Translation Unit State Diagram



RAM Controller

Since many components in the SLG may be reading and writing to the memory, a RAM controller is required to ensure that only one component is using the memory at a time. If the SLG is in initialization mode (i.e. user is overriding sensor values), then the controller will ensure that only the initialization component of the SLG is writing to the RAM. If the SLG is in normal operation mode, then the controller will ensure that only the translation unit is reading values from the RAM.

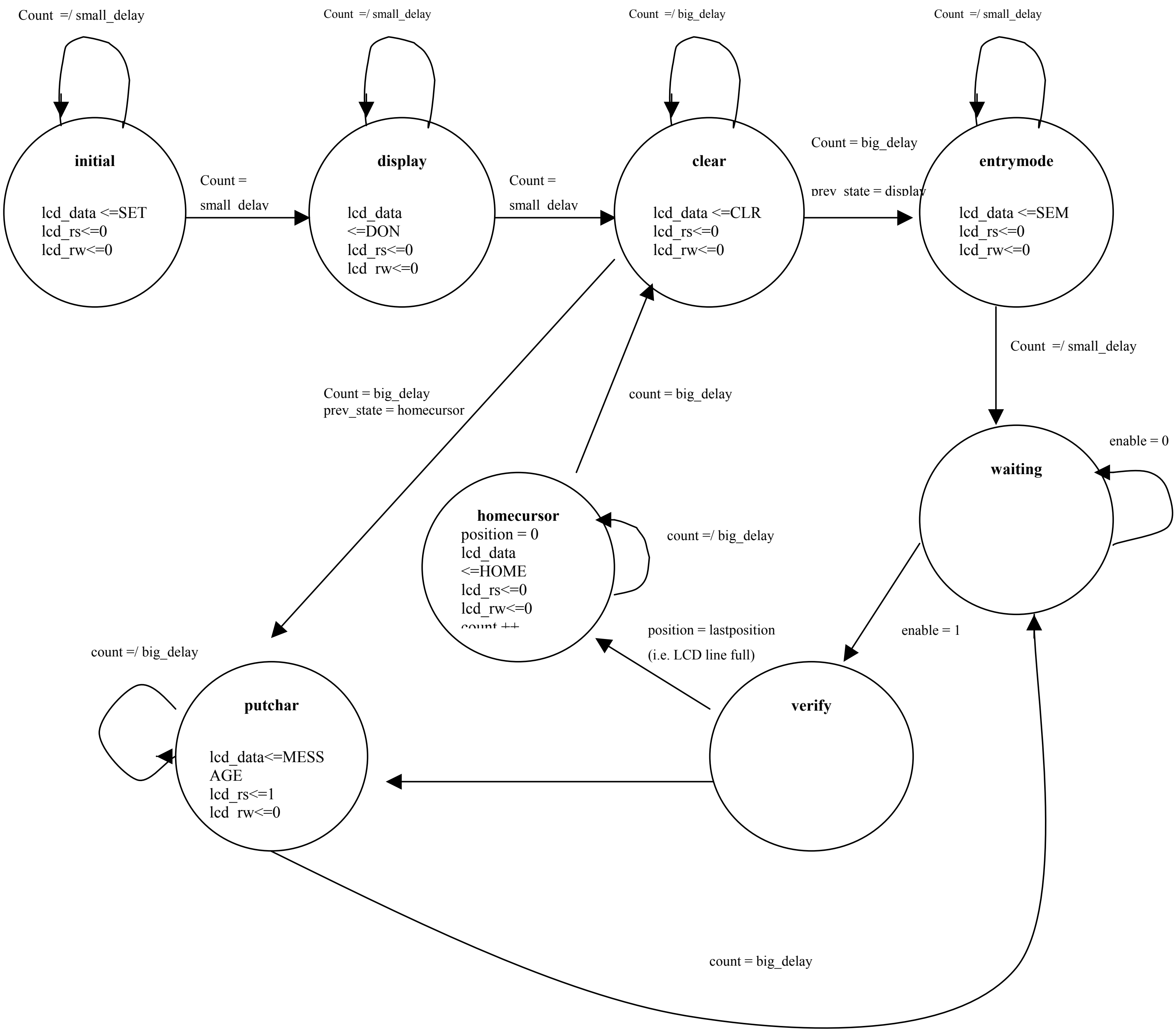
LCD Switch

This component is similar in functionality to the RAM controller. Since many components in the SLG may be using the LCD, the LCD switch ensures that only one component can write messages to the LCD at a time.

LCD Driver

The LCD that the SLG uses is the Lumex model: LCM-S01602DSR/F-Y which uses the Samsung controller IC S6A0069. The LCD attaches to the JP12 port of the Nios Board.

State Diagram for LCDdriver



Datasheet for SLG – Sign Language Glove

General Description

The SLG is a glove implemented sign language translation device. The SLG translates hand positions into letters. The glove is optimized for the American Sign Language.

The SLG utilizes six flex sensors and two contact sensors mounted onto the glove. The contact sensors determine if there is contact between the thumb and index finger and between the index and middle finger. The flex sensors are mounted onto to fingers and measure the bend. Depending on the bend, the flex sensors output a varying voltage. A National Semiconductor ADC0809 analog to digital converter is used to convert the flex sensor voltages into digital values. The output of the SLG program is displayed on a Lumex LCM-S01602DSR/F-Y LCD display.

There are two phases of the SLG. The initial phase is the training phase.

The purpose of the training phase is to train the SLG logic to recognize each individual user. When the SLG program is turned on, the training program is automatically started.

Once training is complete, the translation phase is started. The translation phase translates the users hand positions into a letter and displays this on the LCD.

Features

- Translates hand positions into letters of the English alphabet
- Comfortable, “one size fits all” glove
- Training program adapts the SLG for any user
- Internal memory allows hand position for letters to be stored
- Displays hand positions clearly onto an LCD screen

Electrical Characteristics

The SLG device (EP20K200EFC484-2X) requires a single supply voltage.

Symbol	Parameter	Min	Typ	Max	Units
Vin	Core supply voltage	1.8	1.8	1.8	V

The SLG has the following internal electrical characteristics for its connection with the LCD, ADC and the glove flex sensors.

Symbol	Parameter	Min	Typ	Max	Units
V_{FLEX}	Voltage from Flex Sensor	2.65	3.5	4.42	V
V_{REF} (+)	Reference Voltage for ADC	4.5	5.3	6	V
V_{REF} (-)	Negative Reference Voltage for ADC	-0.1	0	0.5	V
$V_{CONTACT}$	Contact Sensor Voltage	4.5	5.3	6	V
V_{LCD}	Output Voltage to the LCD for each LCD signal	3	10	13	V

Pinouts

Signal	Description	FPGA Pin
Clock		
clock	Clock from FPGA (33MHz)	L6
ADC Pinouts		
ADC_EOC	End of conversion	N19
ADC_OE	Output enable	U5
ADC_clock	Clock for the ADC	K1
ADC_Select[0]	ADC select lines	P19
ADC_Select[1]	ADC select lines	R20
ADC_Select[2]	ADC select lines	N1
ADC_ALE_Start	ADC start conversion	M1
ADC_datain[0]	Digitized output bit 0	N17
ADC_datain[1]	Digitized output bit 1	L14
ADC_datain[2]	Digitized output bit 2	R3
ADC_datain[3]	Digitized output bit 3	M3
ADC_datain[4]	Digitized output bit 4	P1
ADC_datain[5]	Digitized output bit 5	T21
ADC_datain[6]	Digitized output bit 6	T20
ADC_datain[7]	Digitized output bit 7	T22
Contact Sensors		
contact1[0]	Contact sensor 1 (thumb and index)	P18
contact2[0]	Contact sensor 2 (index and middle)	R1
Buttons		
acquire	Acquires data from glove	W9
skip	Skips data from glove (used in training mode only)	T9
reset	Resets SLG program	Y8
space	Puts a space onto the LCD (used in translation mode only)	Y9
LCD Pinouts		
LCD_data[0]	Output message to LCD bit 0	U21
LCD_data[1]	Output message to LCD bit 1	P17
LCD_data[2]	Output message to LCD bit 2	U1
LCD_data[3]	Output message to LCD bit 3	U2

LCD_data[4]	Output message to LCD bit 4	T2
LCD_data[5]	Output message to LCD bit 5	T3
LCD_data[6]	Output message to LCD bit 6	U4
LCD_data[7]	Output message to LCD bit 7	U19
LCD_reg_select	Selects instruction or data mode for LCD	W20
LCD_rw	Read/write select	R18
LCD_enable	Enables LCD	N15

Functional Description

Functional Unit	***Purpose***	***Composition***
Sign Language Glove	Glove worn by signee that converts finger movements into changing voltages.	Flex sensor located at each finger, thumb and under the wrist. Contact sensors between thumb and index, and between index and middle fingers.
ADC0809	Converts fluctuating analog voltages into 8 bit digital sequences.	Circuit made up of ADC0809, taking glove and FPGA inputs via header pins.
ADconverter	Manages various control signals for the ADC0809 and registers the 6 flex sensor voltages. Communicates whether device is in initialization or translation mode.	VHDL code.
InitRam	In initialization mode stores or leaves default values (skips) of sensors for each letter in LPM ram to allow customization of signs for a particular signee.	VHDL code.
RamController	Controls reading and writing to the LPM ram.	VHDL code.
TranslationUnit	Compares currently measured sensor values to those stored in ram in order to translate the hand signs.	VHDL code.
LCDdriver	Communicates with the Samsung driver to display messages.	VHDL code.

Major Blocks of the SLG device

Button	***Purpose***
Acquire	Initialization- Writes flex/contact sensor values for a particular letter to RAM. Translation- Tranlates the hand sign to a letter at LCD display.
Skip	Initialization- Leaves default flex sensor values for a particular value in RAM. Translation- Nothing.
Reset	Resets all parts of the SLG system.
Space	Space bar for the LCD display.

Buttons

Timing Characteristics

The SLG device is constrained in its operating speed due to the dual use of the signal ADC_ALE_Start which controls both the ALE and Start control lines on the ADC0809. This limits the speed of the ADC clock to at or below 500kHz and in order to remove the possibility of clock skew limits the clock speed of the entire device to 500KHz.

Physical Characteristics

Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**
T_{OPR}	Operating Temperature	-30	20	85	°C
T_{STG}	Storage Temperature	-55	20	125	°C

Values obtained from examination of LCD and ADC characteristics

Size/Resources and Chip Performance

Entity	# of Logic Blocks	Total Pins	Memory Bits
clk_divider	17	3	0
debouncer	18	4	0
ad_converter	116	70	0
Init_ram	486	81	0
Translation_Unit	1980	95	0
ram_controller	18	46	4096
LCD_switch	9	29	0
LCDdriver	145	22	0

Measured Number of Logic Blocks for Designed Architectures

SLG total Resource Utilization:

Total logic block Utilization- 2682
Total Register Utilization- 792
Total Memory Utilization-4096 bits
Total I/O Pins- 47

Performance

The maximum clock speed for the SLG top level is 500kHZ, which is limited by the maximum clock rate of the ADC0809 in its current configuration. Without these constraints the maximum clock frequency is limited to 4.16MHz.

Results of Experiments and Characterization

Flex Sensors

The resistivity of each flex sensor varies with bend angle. Below, the minimum and maximum resistance is recorded. Minimum resistance occurs when the fingers are straight, while maximum resistance occurs when fingers are fully bent (ie: in a fist).

[(Flex sensor number) = (Position on the hand), (minimum and maximum)]
1 = flex sensor 1 on the thumb, (11.2 Kohms, 55.6 Kohms)

2 = flex sensor 2 on the index finger, (11.3 Kohms, 55.4 Kohms)

3 = flex sensor 3 on the middle finger, (10.5 Kohms, 53.5 Kohms)

4 = flex sensor 4 on the ring finger, (10.3 Kohms, 53.3 Kohms)

5 = flex sensor 5 on the pinky, (10.9 Kohms, 54.0 Kohms)

6 = flex sensor 1 on the wrist, (10.4 Kohms, 53.5 Kohms)

SLG Alphabet Translation

This chart describes the correlation between the output of the A/D converter and hand positions for each letter of the alphabet. These values will also be used as initial values to be stored in RAM.

Refer to schematic of glove and ADC located in Appendix A

	1	**2**	**3**	**4**	**5**	**6**	**C1**	**C2**
A	62	39	35	33	2D	67	1	1
B	43	7A	84	92	80	67	1	1
C	46	52	55	53	6E	67	0	1
D	52	58	44	45	41	67	0	0
E	41	3A	34	3A	31	67	0	1
F	49	4C	84	85	74	67	1	0
G	47	5C	3C	3C	31	31	0	0
H	47	5C	3C	3F	33	31	0	1
I	52	36	3A	39	47	39	1	1
J	52	36	3A	39	47	67	1	1
K	5A	73	71	36	31	67	0	1
L	60	67	3E	36	31	67	0	0
M	47	6B	62	6A	37	67	0	1
N	51	27	59	3B	37	67	0	1
O	4F	43	3D	42	3D	67	0	1
P	42	64	52	3B	2E	32	0	0
Q	51	61	36	39	34	32	0	0
R	53	77	55	3C	35	67	0	0
S	4A	34	31	33	4E	67	0	1
T	4A	5B	32	3B	31	67	0	0
U	46	46	47	3C	57	67	0	1
V	46	7F	7A	3F	38	67	0	0
W	46	85	7F	6F	40	67	0	0
X	4A	59	32	3A	4F	67	0	0
Y	7B	40	33	3E	67	67	0	1
Z	4A	59	32	3A	4F	38	0	0

Measured Values from the ADC in hex (5 V power supply)

Experimentation

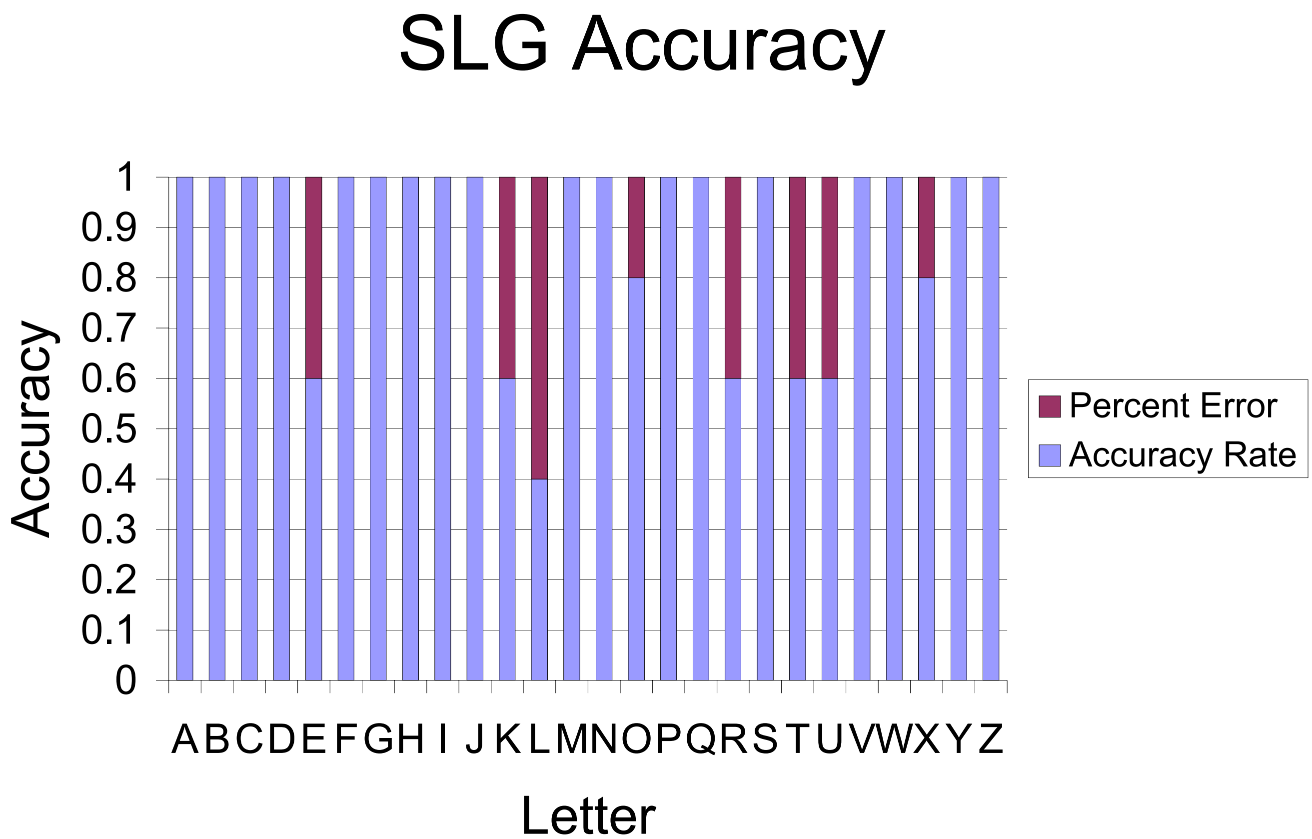
For experimentation and characterization of the glove, a project member put on the glove and calibrated it, using the training program. When all of the hand positions for each letter were recorded into memory, the user tested translation by signing and acquiring each letter of the alphabet. The hand position of the signed letter should correspond to the respective recorded hand position for the letter of the alphabet in memory. Whether the respective letter was shown or not is recorded. This test is repeated four times.

The chart below is the output of our tests and describes the outputs of the glove when signing a letter of the alphabet.

Sign Language Glove Characterization X = Correct O = Wrong

Letter	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Accuracy Rate	Percent Error
A	X	X	X	X	X	100.00%	0.00%
B	X	X	X	X	X	100.00%	0.00%
C	X	X	X	X	X	100.00%	0.00%
D	X	X	X	X	X	100.00%	0.00%
E	O	O	X	X	X	60.00%	40.00%
F	X	X	X	X	X	100.00%	0.00%
G	X	X	X	X	X	100.00%	0.00%
H	X	X	X	X	X	100.00%	0.00%
I	X	X	X	X	X	100.00%	0.00%
J	X	X	X	X	X	100.00%	0.00%
K	O	X	X	O	X	60.00%	40.00%
L	O	X	O	X	O	40.00%	60.00%
M	X	X	X	X	X	100.00%	0.00%
N	X	X	X	X	X	100.00%	0.00%
O	X	X	X	X	O	80.00%	20.00%
P	X	X	X	X	X	100.00%	0.00%
Q	X	X	X	X	X	100.00%	0.00%
R	X	O	X	X	O	60.00%	40.00%
S	X	X	X	X	X	100.00%	0.00%
T	O	X	X	O	X	60.00%	40.00%
U	X	X	X	X	X	60.00%	40.00%
V	O	X	O	X	X	100.00%	0.00%
W	X	X	X	X	X	100.00%	0.00%
X	X	O	X	X	X	80.00%	20.00%
Y	X	X	X	X	X	100.00%	0.00%
Z	X	X	X	X	X	100.00%	0.00%

Overall Accuracy	Overall Error
88.46%	11.54%



Overall, the glove is found to be approximately 88.5% accurate. However, there were several particular letters that gave a higher percent error:

Trouble Letters

Actual Letter Signed	Most Common Letters Displayed
E	S
L	O
V	R
K	U
O	X
T	J, K

The letters (E, L, V, K, O, T) above have a high relatively high percent error due to their similarity with their corresponding letter. It is observed that Trouble Letters can be easily corrected by slightly exaggerating hand motion when signing. For an individual who has excellent sign language abilities, the errors shown will be smaller, since the person will have consistent hand positions for every letter. However, in this case, the users (project members) have limited sign language experience and thus, greater errors are apparent. Overall, the experiments show that the glove is accurate and successful.

References

1. *United States Patent Application 20020152077 'Sign Language Translator'*. Patterson, R. 2002. Accessed January 12, 2003 at http://www.uspto.gov

2. *CMPE 401 – Computer Interfacing.* Department of Electrical and Computer Engineering University of Alberta. 2003. Accessed January 12, 2003 at http://www.ee.ualberta.ca/~kaya/cmpe401/

3. Maxim Integrated Products. 2002. Accessed January 12, 2003 at http://www.maxim-ic.com/

4. *How To Build An Instrumented Glove Based On The Powerglove Flex Sensors.* Axel Mulder, School of Kinesiology, Simon Fraser University. 1994. Accessed January 15, 2003 at http://www.cs.sfu.ca/~amulder/personal/vmi/PCVR.html

5. *The Alphabet American Sign Language.* Accesed January 11, 2003 at http://where.com/scott.net/asl/abc.html

6. Jameco Electronics. 2002. Accessed January 30, 2003 at http://www.jameco.com

7. *ROM_Tutorial2.* Peter Ericka. 1998. Accessed January 28, 2003 at http://www.ee.ualberta.ca/~elliott/ee552/studentAppNotes/1998f/ROM_Tutorial2

8. *How to Write an LCD With Minimal Pain and Anguish.* Kierstead, Jeremy and McMillan, Dave. 1997. Accessed January 28, 2003 at http://www.ee.ualberta.ca/~elliott/ee552/studentAppNotes/1997f/lcd/lcd.html

9. *Simple to use LCD driver (DMC 16207).* David Li, Eric Cheung and Wilson Kwan. 2000. Accessed February 23, 2003 at http://www.ee.ualberta.ca/~elliott/ee552/studentAppNotes/2000_w/interfacing/lcd_driver/lcddriver.vhd

10. *8 Bit ADC with build in sample and hold.* Felicia Cheung, Eric Cheung, David Li, and Tin Wai Kwan. 2000. Accessed February 23, 2003 at http://www.ee.ualberta.ca/~elliott/ee552/studentAppNotes/2000_w/interfacing/ADC_sample_hold/ADC.html

11. http://www.ee.ualberta.ca/~elliott/ee552/studentAppNotes/2000_w/interfacing/ADC_sample_hold/cld_divider1.vhd

12. *S6A006940 SEG / 16 COM DRIVER & CONTROLLER FOR DOT MATRIX LCD.* Samsung. 2000 June.

13. LCM-*X01602DXX/F-Y Spec Sheet.* Lumex. 1998.

14. *Getting an ADC0809 Analog to Digital Converter to Work for You.* Shauna Rae. November 1999. Accessed February 25, 2003 at http://www.ee.ualberta.ca/~elliott/ee552/studentAppNotes/1999f/ad_converter/