Chopper Software for ALMA Band 6 Mixer/Cartridge Measurement System

Student: Lin Qiu

Supervisor: John Effland

08/13/2004

INTRODUCTION

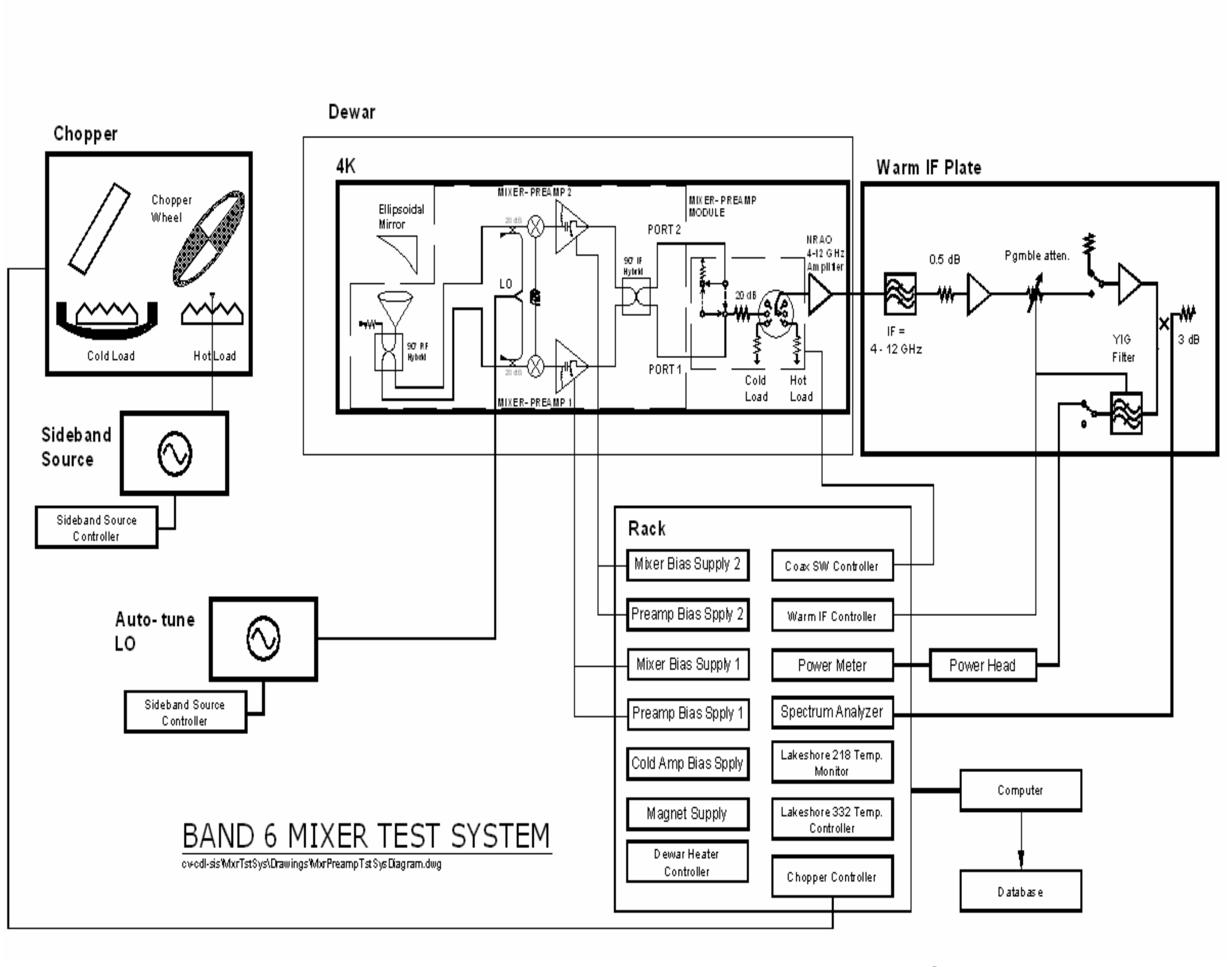
This summer I worked on the mixer and cartridge measurement systems with the ALMA Band 6 group at NRAO. The goals for the summer were to modify the chopper circuit, to write a LabView program, to integrate it with the existing BAND6DAQ measurement software, and to test different parameters to optimize the chopper speed. I have not only been given enough responsibilities for my project, but have also gained adequate support to finish it ahead of the schedule. The BAND6DAQ program synchronizes the spinning chopper wheel to a continuous record of hot load and cold load measurement using the power meter. It also calculates the y – factor by dividing the hot load noise power by cold load noise power.

SOFTWARE LOCATION

LabView code for this project is located in the directory:
\\Cvfiler\USERS\lqiu\Band6_Softw(new code)\VI

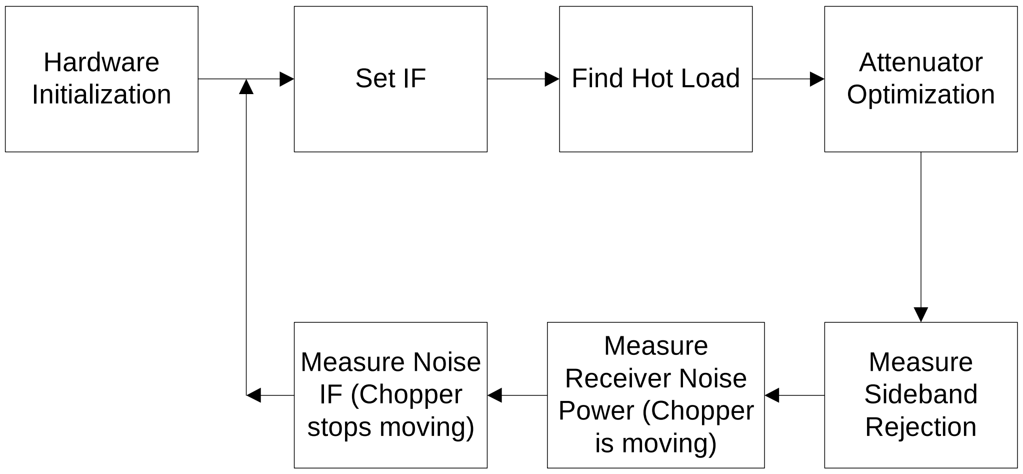
PROJECT

System Overview



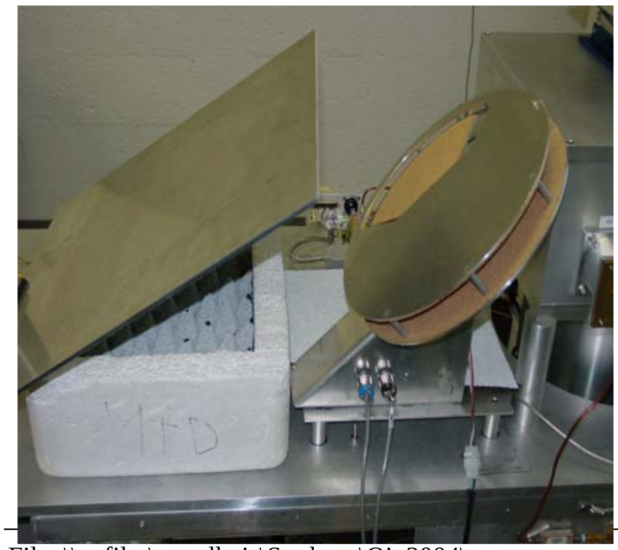
The mixer test system consists of various equipment and electronic devices, such as a refrigerator, a temperature monitor, a LO source controller, mixer bias supplies etc. Eccosorb CV – 3 immersed in liquid nitrogen produces the cold load, which has a temperature of 78K. The hot load has a normal room temperature of 300K. The receiver beam is directed to either the hot or cold load by passing through the chopper. The helium gas from the compressor is expanded in the cold head and cools the temperature inside the dewar to 3.7K. The ideal test temperature is 4K, so a heater is added to heat it back to the required temperature.

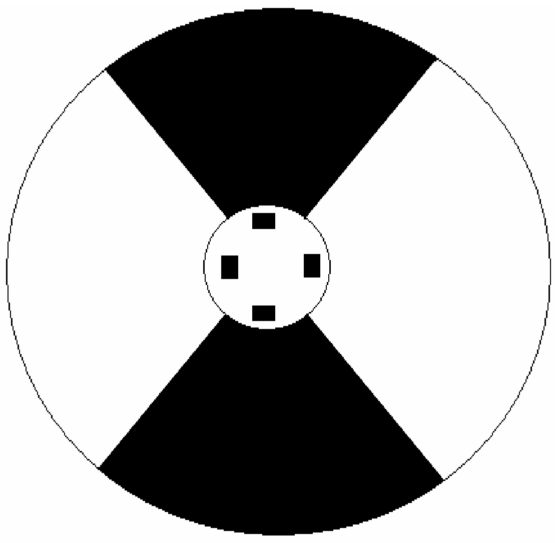
The software running on both systems is called BAND6DAQ. It includes different functions, but the most frequently used one is the “Measurement Sequence” test.


Hardware Initialization
Set IF
Find Hot Load
Attenuator Optimization
Measure Sideband Rejection
Measure Receiver Noise Power (Chopper is moving)
Measure Noise IF (Chopper stops moving)

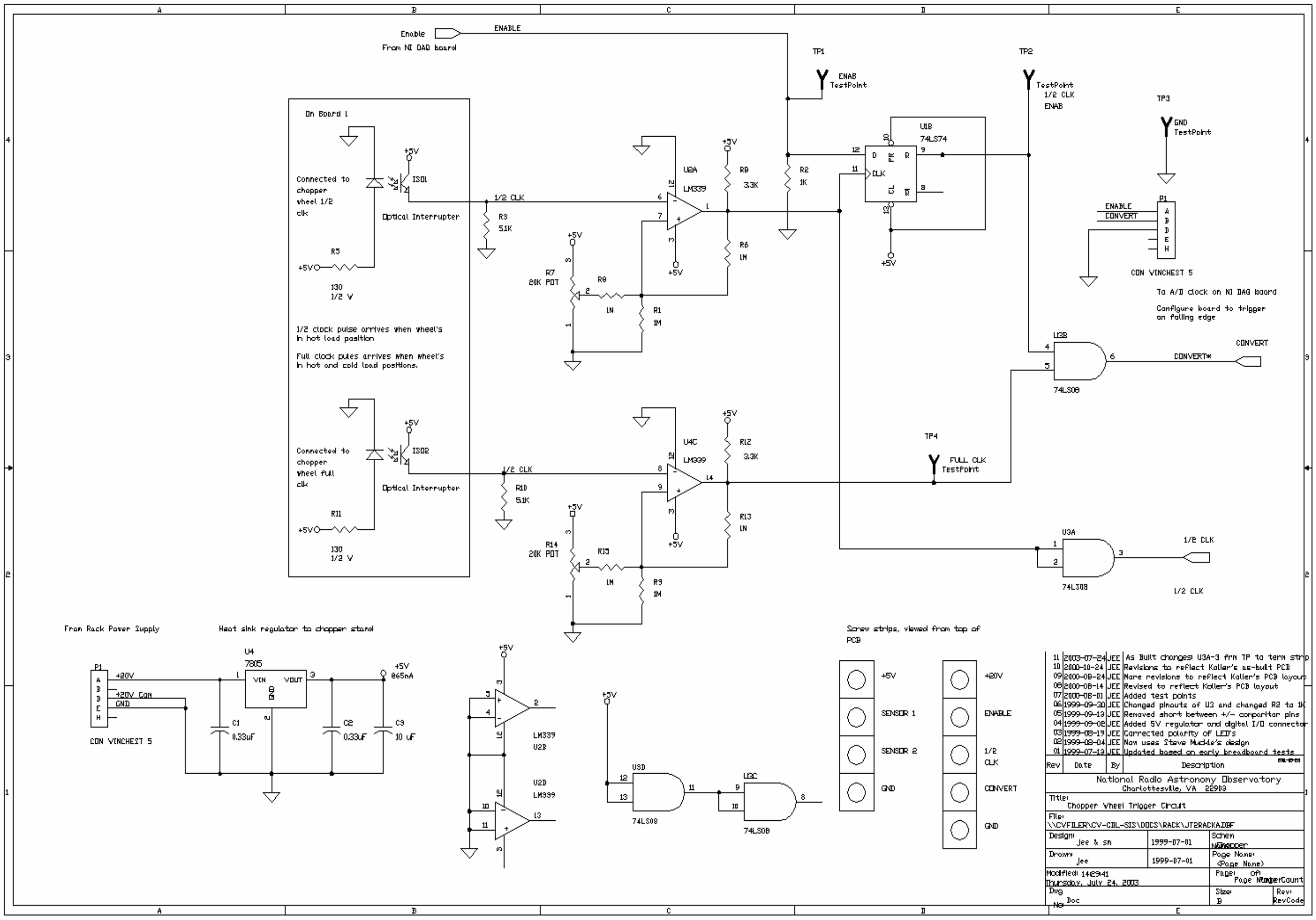
The program executes one event at a time. After the test on one LO frequency is complete, it goes to another frequency to iterate the sequence again without reinitializing the hardware. Among all those functions used during the test, “Find Hot Load” and “Measure Receiver Noise Power” are the most important ones which are directly associated with the chopper program.

Chopper Hardware



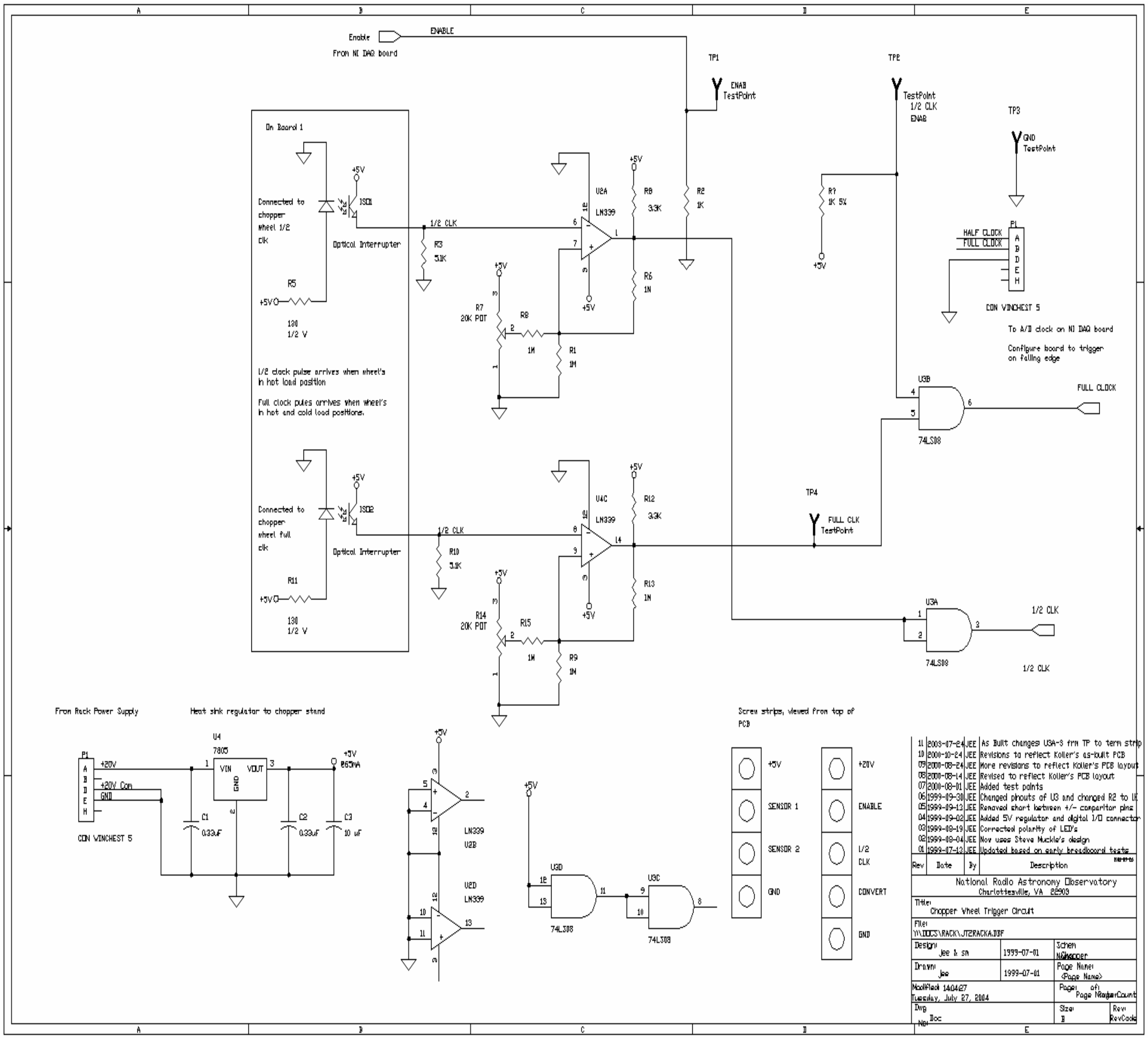


An important function of the chopper software is to locate the hot load or cold load positions of the chopper wheel. The hot load position for the Mixer Test System (MTS) is found when the chopper wheel blocks the receiver beam and reflects into the receiver room temperature noise power. The hot load position for the Cartridge Test System (CTS) is found when the noise power from the hot load in that system passes through the openings in the chopper wheel and continues into the dewar. Two optical interrupters installed on the chopper wheel generate “Half Clock” and “Full Clock” pulses. The half clock pulse occurs each time when the receiver beam passes through the open sections of the wheel. The edge of the opening in the chopper wheel is detected to capture the half clock pulse. The full clock pulse occurs when the beam passes through both open and closed sections of the wheel. Any one of the four tapes near the center is detected to output the full clock signal, which is true at either hot load or cold hot.


Enable
From NI DAQ board
ENABLE
TP1
ENAB TestPoint
TP2
TestPoint 1/2 CLK ENAB
TP3
GND TestPoint
U1B 74LS74
On Board 1
Connected to chopper wheel 1/2 clk
ISO1
Optical Interrupter
1/2 CLK
U3A LM339
R8 3.3K
R2 1K
R3 51K
R6 1M
R7 20K POT
R4 1M
R1 1M
P1
ENABLE
CONVERT
CON WINCHEST 5
To A/D clock on NI DAQ board
Configure board to trigger on falling edge
1/2 clock pulse arrives when wheel's in hot load position
Full clock pules arrives when wheel's in hot and cold load positions.
U3B 74LS08
CONVERT
CONVERT*
Connected to chopper wheel full clk
ISO2
U4C LM339
R12 3.3K
R10 51K
R13 1M
R14 20K POT
R15 1M
R9 1M
TP4
FULL CLK TestPoint
U3A 74LS08
1/2 CLK
From Rack Power Supply
Heat sink regulator to chopper stand
U4 7805
VIN VOUT GND
+5V 865mA
C1 0.33uF
C2 0.33uF
C3 10 uF
+20V
+20V Com
GND
CON WINCHEST 5
LM339 U2B
U2D LM339
U3D 74LS08
U3C 74LS08
Screw strips, viewed from top of PCB
+5V
SENSOR 1
SENSOR 2
GND
+20V
ENABLE
1/2 CLK
CONVERT
GND
11 2003-07-24 JEE As Built changes: U3A-3 frm TP to term strip
10 2000-10-24 JEE Revisions to reflect Koller's as-built PCB
09 2000-08-24 JEE More revisions to reflect Koller's PCB layout
08 2000-08-14 JEE Revised to reflect Koller's PCB layout
07 2000-08-01 JEE Added test points
06 1999-09-30 JEE Changed pinouts of U3 and changed R2 to 1K
05 1999-09-13 JEE Removed short between +/- comparitor pins
04 1999-09-08 JEE Added 5V regulator and digital I/O connector
03 1999-08-19 JEE Corrected polarity of LED's
02 1999-08-04 JEE Now uses Steve Muckle's design
01 1999-07-13 JEE Updated based on early breadboard tests
Rev Date By Description
National Radio Astronomy Observatory Charlottesville, VA 22903
Title: Chopper Wheel Trigger Circuit
File: \\CVFILER\CV-CDL-SIS\DOCS\RACK\JT2RACK.DBF
Design Jee & sn 1999-07-01
Drawn Jee 1999-07-01
Modified: 14:29:41 Thursday, July 24, 2003
Size B

On the old schematic above, the Enable line is set to be high, so the output of the D flip-flop will become true when the half clock pulse arrives. Once it is true, it is always true regardless of the changes of the half clock signal. The trigger signal coming out of the convert line is completely dependent on the full clock pulse. If it is at hot load when the first trigger is received, it will be at cold load the second time the convert pulse arrives. It forces the software to configure the first trigger to hot load, and to alternate between hot load and cold load. This approach works if the sole purpose is to find the hot load and stop the chopper. A problem occurs when recording the mixer noise power while the chopper is in motion. If the program misses one trigger, the hot load noise power can be measured at cold load or vice verse.

To solve the problem, the schematic was modified.



Enable
From NI DAQ board
ENABLE
TP1
ENAB TestPoint
TP2
TestPoint 1/2 CLK ENAB
TP3
GND TestPoint
On Board 1
Connected to chopper wheel 1/2 clk
Optical Interrupter
ISO1
ISO2
1/2 clock pulse arrives when wheel's in hot load position
Full clock pules arrives when wheel's in hot and cold load positions.
Connected to chopper wheel full clk
U2A LM339
U4C LM339
R2 1K
R3 5.1K
R5 130 1/2 V
R6 1M
R7 20K POT
R7 1K 5%
R8 3.3K
R12 3.3K
R13 1M
R14 20K POT
R10 5.1K
R11 130 1/2 V
R1 1M
R9 1M
R15 1M
P1
HALF CLOCK
FULL CLOCK
CON WINCHEST 5
To A/D clock on NI DAQ board
Configure board to trigger on falling edge
U3B 74LS08
FULL CLOCK
TP4
FULL CLK TestPoint
U3A 74LS08
1/2 CLK
From Rack Power Supply
Heat sink regulator to chopper stand
U4 7805
VIN VOUT GND
+20V
+20V Com
GND
+5V 865mA
C1 0.33uF
C2 0.33uF
C3 10 uF
LM339 U2B
U2D LM339
U3D 74LS08
U3C 74LS08
Screw strips, viewed from top of PCB
+5V
SENSOR 1
SENSOR 2
GND
+20V
ENABLE
1/2 CLK
CONVERT
GND
11 2003-07-24 JEE As Built changes: U3A-3 frm TP to term strip
10 2000-10-24 JEE Revisions to reflect Koller's as-built PCB
09 2000-08-24 JEE More revisions to reflect Koller's PCB layout
08 2000-08-14 JEE Revised to reflect Koller's PCB layout
07 2000-08-01 JEE Added test points
06 1999-09-30 JEE Changed pinouts of U3 and changed R2 to 1K
05 1999-09-13 JEE Removed short between +/- comparitor pins
04 1999-09-02 JEE Added 5V regulator and digital I/O connector
03 1999-08-19 JEE Corrected polarity of LED's
02 1999-08-04 JEE Now uses Steve Muckle's design
01 1999-07-13 JEE Updated based on early breadboard tests
Rev Date By Description
National Radio Astronomy Observatory
Charlottesville, VA 22903
Title: Chopper Wheel Trigger Circuit
File: Y:\DOCS\RACK\JT2RACKA.DBF
Design: Jee & sn 1999-07-01
Drawn: Jee 1999-07-01
Modified: 14:04:27 Tuesday, July 27, 2004
Size: B

In the new chopper circuit, the Enable line is no longer needed. The D flip-flop is taken off from the circuit board, and a 1K resistor is inserted to connect pin 13 and pin 9 to ensure one of the inputs to the AND gate is always true. In addition, both full clock and half clock signals are outputted in the new circuit.

Chopper Software

Some changes have been made since the new chopper algorithm was built and integrated with the BAND6DAQ software. The three commands “Move to Hot Load” “Move to Cold Hot” and “Find Hot Load” in the previous chopper program have been simplified to two more accurately defined commands “Find Hot Load” and “Find Cold Load”. To find the hot/cold load, we detect the rising edge of the half clock pulse and then move the chopper a few more indexes to the corresponding positions. The old command “Configure chopper to receive 1st trigger” does not exist in the new program. It is not necessary to configure the first trigger to hot load since we do not use functions to alternate between hot and cold load any more. Instead, we define the status of the load by decoding the trigger signal while the chopper is running.

MyChopper 3.vi receives an enumerated constant as input, which is wired to a case structure that contains several commands:

Open Com
Stops the chopper if it is currently moving and sets the motor controller at variable resolution mode.

Move at Fixed RPS
Causes the motor to move at a constant speed in revolutions per second (RPS). It determines the divide resolution of the controller and then calls the bMove_At_Fixed_Velocity subVI with the appropriate stepper motor pulse rate.

Set Origin
Calls the bSet_Origin.vi to set the current position as the origin, that is, the chopper controller index is set to 0.

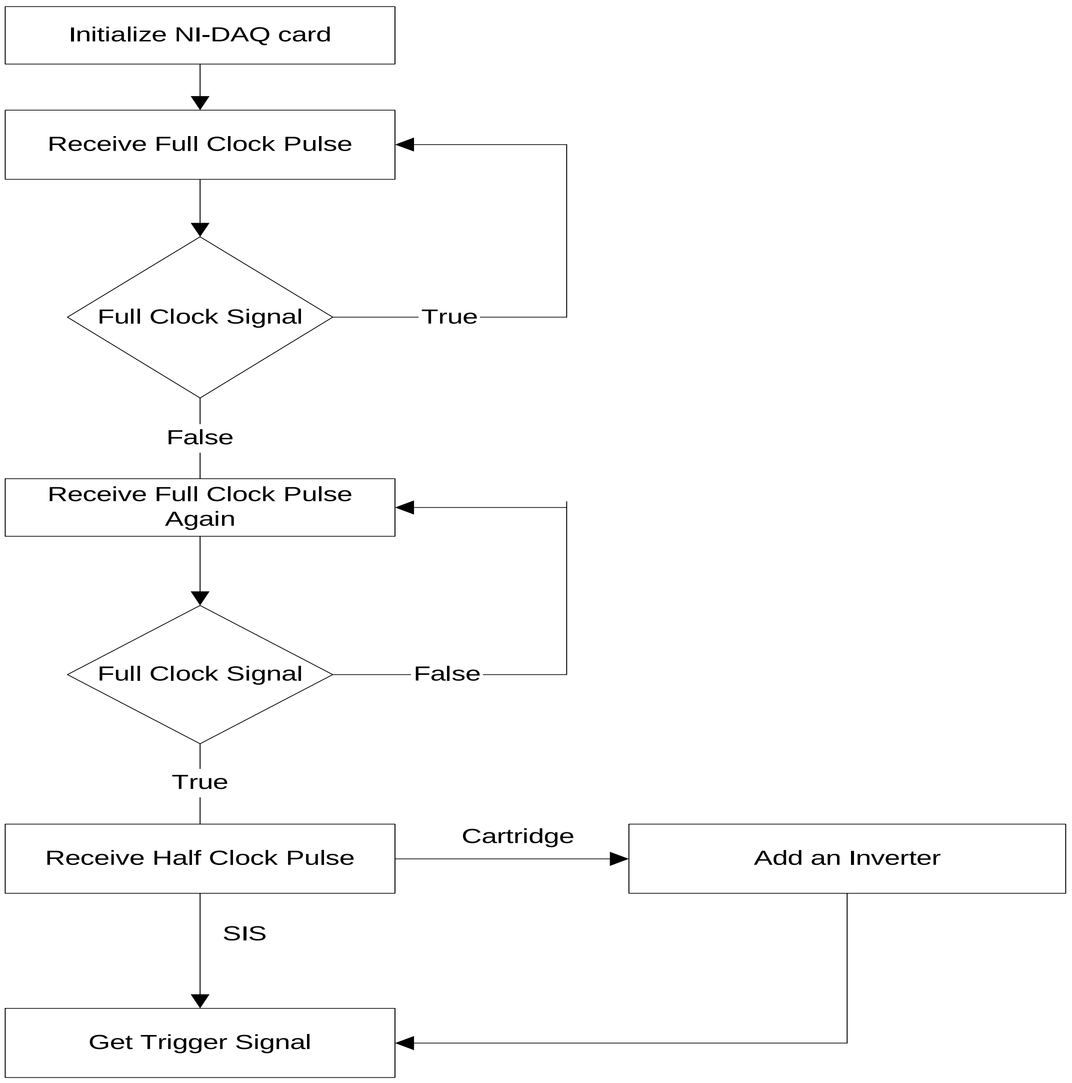
Find Cold Load
Finds the cold load position. It moves the chopper at the constant RPS until it detects the rising edge of the half clock pulse (the first while loop stops execution when the half clock signal becomes false, and the second while loop stops when it receives a true signal). It soft stops the chopper and then moves the chopper a few more indexes (around 23 for cartridge and 72 for SIS) to cold load position. The extra movement ensures that the test system receives an unrestricted view of the load.

Find Hot Load
Finds the hot load position. It moves the chopper at the constant RPS until it detects the rising edge of the half clock pulse (the first while loop stops iterating when the half clock signal becomes false, and the second while loop will not stop until it receives a true signal). It soft stops the chopper and then moves the chopper a few more indexes (around 22 for SIS and 73 for cartridge) to hot load position. The extra movement ensures that the test system receives an unrestricted view of the load.

Get Trigger
Receives the trigger signal. The boolean output is true at hot load and false at cold load.

Two “while loops” in the code detect the rising edge of the full clock pulse and provide a TRUE output. Subsequent detection of the half clock signal provides the means to identify if the full clock pulse corresponds to the hot or cold load. In this way, even though the program misses one trigger, it can still record the noise power with respect to the correct load.

Flow Chart of “Get Trigger” Command



This new routine will work only if the half clock signal leads that of the full clock. In other words, the half clock pulse must arrive slightly before the full clock does.

An inverter is also needed between receiving half clock pulse and outputting trigger signal when the cartridge test is on. This is because the cartridge test system’s hot load and cold load are set up opposite that of mixer test system’s loads.

Accomplishments

Unlike the old one, the new program is entirely coded in LabView, which frees the programmer from other languages such as Visual Basic or .Net. It is less complicated but more efficient. The algorithm is restructured to make it more understandable to the user. The architecture is simplified for easier code maintenance, so that other programmers can add more functions or make changes whenever necessary.

Using both full clock and half clock pulses to get the trigger solves the synchronization problem mentioned before. The signal itself tells which load the chopper is observing and executes the corresponding event.

By changing several parameter inputs to speed up the chopper, a single LO frequency test was shortened from 12 minutes to 9 minutes and 30 seconds. A full bandwidth test (220GHz – 270GHz) can now be completed 16 minutes earlier in the new program, which results in a 22% reduction in total test time.

Time Comparison

Freq LO (GHz)	IF (GHz)	Time Stamp (new program)	Time Stamp (current program)
270	4	2004-49-26 14:49:14	2004-14-22 15:14:25
270	5	2004-50-26 14:50:24	2004-15-22 15:15:56
270	6	2004-51-26 14:51:33	2004-17-22 15:17:26
270	7	2004-52-26 14:52:42	2004-18-22 15:18:59
270	8	2004-53-26 14:53:53	2004-20-22 15:20:25
270	9	2004-55-26 14:55:04	2004-21-22 15:21:53
270	10	2004-56-26 14:56:08	2004-23-22 15:23:18
270	11	2004-57-26 14:57:19	2004-24-22 15:24:47
270	12	2004-58-26 14:58:29	2004-26-22 15:26:16
		9min 15s	11min 51s

The whole software development process is appropriately documented to meet the design standard, which details changes being made to the hardware, old commands being discarded and new functions being created in the software, and goals being achieved on both test systems.

SUMMARY

It has been a very productive summer with NRAO. The experience of this internship is rewarding, and the skills leaned at work are valuable. Thanks for all the people with whom I have worked or may just have a conversation. Your help and cooperation have helped me achieve so much, and your constructive criticism and suggestion will certainly be taken on to benefit my future education and career.

Appendix

The following are lower level VI's that are called from myChopper 3.vi. Using sub VIs avoids having all functions written in one big VI that is difficult to debug and maintain. They further simply the communication interface between the motor controller and the higher level software.

Variable_ResMode.vi
Sets the resolution mode (fixed or variable) of the motor controller

Set_Velocities
Sets the initial and slew velocities of the chopper

Divide_Resolution.vi
Sets the step resolution of the motor controller.

Set_Currents.vi
Sets the hold and run currents used by the motor controller.

Set_Slope.vi
Sets the acceleration and deceleration slope used by the motor controller

Move_At_Fixed_Velocity
Causes the chopper to move at a constant pulse rate.

Index.vi
Causes the chopper to move the specified number of steps.

IsMoving.vi
Reads the current moving and mode status of the controller.

Set_Origin.vi
Sets the current motor position as the origin.

Soft_Stop.vi
Decelerates the motor to a stop.

Abort.vi
Stops the chopper without deceleration.