1 GHZ DIGITIZER FOR SPACE BASED LASER ALTIMETER

FINAL REPORT

16 October 1991

Prepared For:

NASA GODDARD Space Flight Center
Greenbelt, Md. 20771

Contract No.: NAS5-30626
Project Manager: Jack Bufton

Submitted By:
Amerasia Technology, Inc.
2248 Townsgate Road
Westlake Village, CA 91361

Principal Investigator
Edward J. Staples
TABLE OF CONTENTS

1.0 Phase II Objective.......................................................................................... 1-1

2.0 Program Summary.......................................................................................... 2-1
   2.1 Initial Technical Approach............................................................................ 2-1
   2.2 Technical Problems Encountered and Possible Solutions............................ 2-1
   2.3 Selected Technical Approach......................................................................... 2-3

3.0 Work Accomplished......................................................................................... 3-1
   3.1 Summary...................................................................................................... 3-1
   3.2 Characterization of the AN1000H Hybrid.................................................... 3-8
      3.2.1 Design of the Test Circuit...................................................................... 3-8
      3.2.2 16-Way Active Multiplexer Design...................................................... 3-10
   3.3 System Description....................................................................................... 3-11
      3.3.1 Signal Input Stage.................................................................................. 3-17
      3.3.2 Staggered Clock Generation.................................................................... 3-17
      3.3.3 Analog Memory and Presampler............................................................. 3-18
      3.3.4 Microprocessor Interface and Communications Circuit with Host Computer ........................................................................... 3-19
      3.3.5 Power Supply Duty Cycle Control......................................................... 3-20
   3.4 Testing.......................................................................................................... 3-21
      3.4.1 Characterization of the AN1000H Hybrid.............................................. 3-21
      3.4.2 System Tests........................................................................................... 3-22
      3.4.3 Special Operations Instructions and Notes............................................. 3-24

4.0 Conclusions and Recommendations............................................................. 4-1

   I. Appendix 1
   II. Appendix 2
   III. Appendix 3
   IV. Appendix 4
   V. Appendix 5
1.0 PHASE II OBJECTIVE

The overall objective of this Phase II Program is to develop, design, fabricate, and test a feasibility model of a low power 1 GHz Waveform Digitizer. The Digitizer is intended for analyzing data collected via a space based Laser Altimeter. It has a 6-bit resolution, and is equipped with a 1 GHz surface acoustic wave (SAW) oscillator, and a random access buffer memory to interface with the 8-bit parallel bus of the altimeter system computer. Low power consumption is obtained by cutting off the power supply during the absence of data from the altimeter, thus lowering the duty cycle of power utilization.

The following technical objectives were specified for the implementation of the 1 GHz Waveform Digitizer feasibility model.

Functions:
(1) Perform analog-to-digital conversion
(2) Transfer digital data to system computer via a random access memory and an 8-bit parallel bus (DMA)
(3) Provide a 1 GHz system timing clock

Inputs:
(1) Signal bandwidth: DC to 350 MHz
(2) Signal pulsewidth: 4 to 10 Nanoseconds
(3) Repetition frequency: 40 Hz
(4) Input impedance: 50 Ohms, nominal

Outputs:
(1) Sampling rate: 1 Gigasamples/second
(2) Resolution: 6 bits
(3) Clock frequency: 1 GHz
(4) Number of samples: 128/second

Power supply:
(1) Average power consumption: 1.5 Watts
(2) Duty cycle: 2.5%
(3) Available supply voltages: 5V, -5.2V, -2V, and 12V
2.0 PROGRAM SUMMARY

2.1 Initial Technical Approach

A system diagram of the initial approach is shown in Figure 2-1. Two Tektronix flash 8-bit A/D-converters (TKAD10C) were selected to digitize the input signal. Each is capable of 500 Megasamples/second operation. By paralleling these converters an effective 1 Gigasample processor is obtained. The converters have 7-bit accuracy, as specified by its manufacturer. The digitized data can be stored in 32 RAMs, each being an 8-bit 30 MHz device. Alternatively, eight 4-bit 250 MHz RAMs were considered. Write and Read activities in the RAMs are controlled by an address counter as shown. Sampled data is transferred to the system computer via DMA type bus interface. Timing is derived from a 1 GHz SAW oscillator and distributed to the converters counters, and computer interface bus.

2.2 Technical Problems encountered and Possible Solutions

It was found later that the total set-up and hold time of the high speed memories (RAM) is equal to the access time (4 nanoseconds), hence there is no allowance for any timing errors during acquisition at full speed. The following solutions were considered to overcome this problem:

1. Replace the RAMS with high speed multistage (128) shift registers utilizing gate array technology
2. Use 16 RAMS in place of the original 8, allowing for half speed clocking.
3. Incorporate hybrid memory technology.

The first approach was investigated and found to be technically risky and not cost effective, due to the associated nonrecurring engineering cost. The second approach would increase the systems parts count, resulting in higher power consumption and reduced reliability. An existing hybrid memory and data sampling device (AN1000H) was found that is useable and met the technical requirements. This device is manufactured by Analytek and used in their Gigasample data acquisition systems. Therefore, there is minimal technical risk involved with the application of this device. In addition, the devices are readily available.
Figure 2-1, 1 Gigasample/sec. Digitizer System Diagram
2.3 Selected Technical Approach

A system diagram of the selected technical solution is shown in Figure 2-2. In this approach fast analog storage is provided by the AN1000H hybrids, which is followed by a slow readback via an A/D Converter at a moderate speed. The analog memory is made up of 4 hybrids. Each hybrid consists of four channels containing 256 randomly addressable charge storage capacitors capable of storing an analog voltage of ±2 volts with a resolution of ±1 millivolt. Each channel can operate at 62.5 Megasamples/second. By delaying the data and clock signals appropriately the system can be designed for 16 channels, equally spaced within one 16 nanosecond window, giving 1 nanosecond resolution. Fast sample/hold presamplers on each channel insure adequate aperture parameters to make the acquired signal meaningful at this rate.

Initially the input signal was delayed in increments of one nanosecond via a 16-tap delay line. This approach was not satisfactory because of differences in losses among multiple taps, although small between two adjacent taps, are excessively large thus deteriorating the acquisition accuracy of the digitizer. Moreover, stripline implementation of the tapped delay line is adversely effected by temperature due to expansion and contraction. This problem was resolved by offsetting the acquisition instants of each channel of the hybrids by one nanosecond, respectively.
Figure 2-2, Lola A/D Converter Block Diagram
3.0 WORK ACCOMPLISHED

3.1 Summary

During the Phase II development program a feasibility model of the 1 GHz Digitizer was designed, fabricated and tested. Hardware system testing was not fully completed due to limited funding. Preliminary testing of the A/D function of the hardware demonstrated satisfactory results. A software package for data processing and interface with the host computer was developed but not tested. Source code listing of the software is attached in Appendix 1.

The 1 GHz SAW clock required for system timing is included, and a timing interface circuit was designed and built as part of the hardware printed circuit board. Average power consumption is within the specified limit, while power up and power down timing were satisfactorily tested.

As mentioned earlier, initially the design was based on utilizing Tektronix's flash A/D converters (TKAD10C). For this purpose a clock distribution circuit was designed, the results of which are shown in Appendix 2. The 1GHz oscillator which was acquired for this purpose is also useable in the final design.

The hybrid devices (AN1000H) used in the final design were characterized. For this purpose a test circuit was designed and built. A schematic diagram showing the test circuit and the printed circuit board are shown in Figures 3-1 and 3-2. The results are shown in Appendix 3. During this effort one hybrid, containing 4 channels, was tested by characterizing its transfer function and input-output signal waveforms from 150 KHz to 1 GHz. Each channel of the hybrid contains 256 cells. Each of these cells have variations in gain and offset, consequently the output (sampled data) waveform becomes noisy. Corrections were made and the accompanying result shown, where the waveform is considerably cleaner.

A tapped delay line for the purpose of distributing the input signals with one nanosecond offsets to all the 16 channels of the hybrids was designed and built. This implementation is shown in Figure 3-3 and 3-4; it is basically a transmission line with 16 one nanosecond taps, constructed on epoxy fiberglass (G-10) material utilizing microstrip technique. The result of this implementation was not satisfactory due to dielectric and copper losses. A second version was built utilizing duroid printed circuit board, as shown in Figure 3-5. This implementation was also not satisfactory. Finally the approach was abandoned and this problem resolved by staggering the timing of each channel by 1 nanosecond. The test results of the delay line implementations are shown in Appendix 4.
In what follows the development of the 1 GHz Digitizer utilizing the AN1000H hybrid chips are described. It includes the characterization of these chips prior to hardware design, design of the Digitizer circuit, and software for control and interface with the system computer.
Figure 3-1, Lola Test Circuit
Figure 3-2, Test Circuit Layout
Figure 3-3, 16-Way Active Multiplexer
Figure 3-4, Microstrip Delay Line with pads for JFET mounting.
Figure 3-5, Goddard Delayline
3.2 Characterization of the AN1000H Hybrid

3.2.1 Design of the Test Circuit

The test circuit is shown in Figure 3-1 and 3-2. Since the objective is to characterize the AN1000H hybrid analog memory module, the design includes only one chip. Each hybrid has four inputs; in the 1 Gigasample Digitizer four hybrids are used and data is acquired via a 16-way multiplexer. Multiplexing is accomplished by a combination of delays in the timing and data paths such that the data latched by each of the inputs is staggered by one nanosecond referenced to time of actual occurrence. In order to achieve this, the presamplers in the hybrids must be fast enough to accurately take their samples. In addition the internal delay times must be known and included in the overall timing delay consideration.

The sampling window and timing is determined accurately in a single hybrid. The sampling waveforms should be the same as if it were one of four hybrids in the final system. The delays should be identical or at one nanosecond intervals so as to represent one or more noncontiguous repetitive samples out of the full set. Identical timing was chosen since it would provide the easiest method of determining channel to channel variations in both timing and amplitude.

In this test circuit on-board microprocessor is not included. The final circuit, however, will be controlled by a combination of microprocessor and dedicated logic, which will also unload and correct the data. Part of our objective is to determine the necessary corrections, therefore raw data is being collected. The on-board A/D converter is connected to a general purpose parallel port on an MS-DOS computer where simple programs in BASIC or C can be used to unload and save the data for further analysis. Analysis will include gain, offset and timing errors for each channel, offset and gain, offset and linearity errors for each memory cell, as well as input frequency versus amplitude errors for each memory cell, to check the sampling accuracy.

The power and control circuitry is not needed in the test board. It has been designed with adequate power and heat dissipation capacity for 100% duty cycle. Power down circuitry will be added later to determine its effect on performance and to determine minimum turn-on and turn-off times.

Since only four channels are being driven, a resistive divider is used to provide adequate impedance match. A signal function generator is used as a test source.

Referring to the circuit in Figure 3-1, several power voltages are required and a number of timing signals as shown in Figure 3-6. The hybrid needs Vtt for termination of the ECL inputs, Vsub for substrate biasing and Vofs as a fixed DC offset to the signal paths.
Voltages needed for sensitive analog portions of the circuit are isolated from noise of the
digital circuitry by ferrite beads and bypass capacitors. There are two analog grounds, one at
the input of the hybrid and the other between the hybrid and the A/D converter. Isolated
analog sections of each plane have been created by partitioning the planes.

The timing signals are summarized in the timing diagram shown in Figure 3-6. The
AN1000H collects samples during the high portion of the S12 an S34 signals. These samples
are latched in the presampler on the falling edge of S12 an S34. The data in the presampler is
transferred to a memory cell during DCLK high, and latched on the falling edge of the
DCLK. The relationship between S12, S34 and DCLK should be such as to maximize the
time from the falling edge of DCLK. DCLK also clocks the shift registers used to select the
next memory cell for writing. These registers are cleared when RSR and RST1 are low and
start counting when LSR and START-X go low, loading a 1 bit into the fast row and slow
column shift registers respectively. ORST is the same signal as RSR. All analog memory
locations are reset on the rising edge of ORST. END1-4 is used to determine the end of the
collection (WRITE) phase, so that the unload phase can be started. O2S is used to
synchronize the START-X signals. RST2 is held high to prevent read attempts during the
collection phase.

3.2.2 16-Way Active Multiplexer Design

Distribution of signals and time delay to drive the hybrids is accomplished by a
16-Way Multiplexer. System requirements and characteristics of the AN1000H hybrid dictate
the performance requirement of the multiplexer. Following are the design objectives:

Input parameters:

- $0 \pm 360 \text{ mV (720 mV P-P)}$
- DC-350 MHz
- 50 Ohms, Impedance
- VSWR 1:8, maximum

Output parameters:

- 16 Outputs
- 1 ns delay between each output
- $0 \pm 360 \text{ mV}$
- DC-360 MHz, flat frequency response
- 50-Ohm impedance load (50 Ohms shunted by 4.7 pF)
- VSWR 1.8:1 maximum

3-10
Power Supply:
5 VDC
-5.2 VDC

The design of the multiplexer is shown in Figure 3-3. The signal distribution and delay requirements are achieved by tapping a microstrip transmission line at 16 points, each 1 ns apart. The line is meandered to reduce layout length and is adequately spaced to reduce coupling between adjacent segments. Circuit loading at the taps is minimized by using JFET buffer amplifiers with high input resistance and low gate-source capacitance. Current gain is supplied by a bipolar transistor output stage.

The circuit is built on a 31-mill microstrip board utilizing G10 dielectric material. The delay line is shown in Figure 3-4. Surface mount packaged JFETs are used. Initially, test data is taken with 3.3 pF chip capacitors simulating the gate capacitance of the JFETs. The test results, showing group delay and insertion loss as a function of frequency, are shown in Appendix 4. These measurements show that the transfer characteristic of the delay line has a 3dB slope down to 350 MHz, which is too large to meet system performance objectives. A second design, shown in Figure 3-5, was tested. In this design lower loss dielectric material (Duroid) was used. A slope of 1.2 dB was obtained, which is smaller but still considered unacceptable. Therefore, this approach was abandoned, and the problem is resolved by sampling the input signal at sampling times staggered by 1 ns from channel to channel.

3.3 System Description

The 1 GHz Digitizer system block diagram is shown in Figure 2-2, and the detailed circuit diagrams are shown in Figures 3-7 through 3-10. A short duration of the input signal is stored as discrete analog samples in a set of AN1000H Hybrids. There are four hybrids, each of which has four channels. Each channel can capture a signal with picosecond precision using fast pre-samplers, but requires a 16 ns cycle to store the sample. The four hybrids provide 16 channels, each staggered by 1 ns thus acquiring a new sample every nanosecond. The pre-sampler clocks are brought out of the hybrids in pairs, allowing for eight different clocks (two nanoseconds apart). The two channels sharing the same clock are connected to different signal paths, one having an extra 1 ns of coax delay in it. Each channel is 256 cells deep, resulting in a total acquisition of 4096 samples, or over 4 microseconds of data. No attempt has been made to shorten the acquisition cycle. After the samples are collected, a readout phase begins, which transfers the successive sample of each channel to a fast 12 bit A/D converter. Only a small fraction of the 4096 samples are actually needed, but the four
channels on each hybrid are arranged such that one must be completely read before the next one can be started. The time (and power consumption) of this requirement is minimized by rapidly clocking out the samples before and after the desired segment, and slowing the clock to the speed of the A/D converter for the samples which are actually needed. This process is currently implemented using fixed constants, but could be modified to identify the region of interest, with a small increase in on time. The cells beyond the end of the region of interest in the fourth channel in each hybrid do not need to be clocked out at all, so the closer the desired data is to the beginning of the buffer, the less time will be required in the unload phase. The readout clock and A/D converter control are generated in software in the microprocessor. This is a full time task during that interval, and could not have even been considered on a slower processor. For this reason, no processing is done until the readout is completed. Upon completion, the data is scaled by gain and offset values stored from a calibration sequence and transferred to the host system. Calibration consists of starting the calibration sequence, providing a fixed voltage input, doing an acquisition cycle, providing the value of the fixed voltage, changing the voltage, and repeating the process as many times as desired (3 minimum, 16 maximum, 6 to 10 suggested). A final call to the calibration routine converts the accumulated summations into slope and intercept parameters for each cell. Separate calibration data is necessary for each cell because of variations in the sample and hold capacitors which store the samples. The host system communicates with the A/D system by means of a 16-bit bi-directional data bus with 6 handshaking wires. A command set is provided that allows not only calibration and operation, but also allows reading and writing (except EPROM) all data memory and code memory addresses, individually or by blocks, and input or output to any I/O address. Extra commands are reserved which are initially NOP's but can be patched to provide added functionality, even in flight.

In order to minimize power consumption and heat dissipation, power duty cycling has been implemented. Many of the power requirements, including all of the GaAs and ECL logic are needed only during the four microseconds of acquisition. These are known as the write loads. Several other voltages are needed only during acquisition and readout. These are known as the read loads. All switched voltages turn on within 5 microseconds, and are designed to be stable within 10 microseconds. Another 10 microseconds is allowed for the circuitry drawing the power to stabilize. Ten microseconds is allowed as a trigger window, and another 10 microseconds for the acquisition and shutdown of the write loads. The read loads remain in for several milliseconds, depending on the size and position of the data to be unloaded.
Figure 3-9, Lola A/D Converter
Figure 3-7, Lola A/D Converter
Figure 3-10. Lola A/D Converter
3.3.1 Signal Input Stage

The input amplifier is constructed in two sections, as shown in Figure 3-7, each having two stages. The sections are identical except that one section has no delay, the other a one nanosecond coaxial delay line. The first stage is a buffer amplifier which provides minimal loading of the input signal to allow bridging of the two stages, and low enough output impedance to drive the eight second stage amplifiers in parallel. The input signal is terminated in 50 ohms at the end of the 1 nanosecond length of coax. The second stage provides minimal loading to the first stage and a 50 ohm output to match the hybrids. A wideband amplifier having a frequency response which is flat from DC to 350 MHz pushes the state of the art in semiconductors and integrated circuits. It would be much simpler if DC coupling were not required, or if a DC offset could at least be tolerated. DC coupling was a requirement partly to simplify calibration, and partly because any coupling capacitors which could charge and stabilize during the 20 microsecond turn on time would not pass low enough frequencies to leave the pulses being measured undistorted. Operational amplifiers represent the most straightforward method of providing DC coupling without significant offset. The recent development of current feedback op-amps allows their use to frequencies in excess of 300 MHz. The CLC 409 was chosen for this application. With careful construction practices it can be built into a system which is fairly flat to 350 MHz.

3.3.2 Staggered Clock Generation

In order to acquire samples with 1 ns resolution, a 1 Ghz clock with fast rise and fall times must be the basis of the timing logic. A SAW oscillator was used for this purpose. The oscillator is connected to a power splitter which provides an external output for other uses, and a clock signal to a GaAs shift register. The shift register is connected to circulate a pattern of 8 zeros followed by 8 ones. By tapping this signal at various stages, the four 16 ns clocks were provided (one for each hybrid, four nanoseconds apart). The eight pre-sampler clocks come from GaAs gates which generate pulses based on the time it takes for the basic clock transition to pass from one tap of the shift register to a tap two stages later. This provides a 2 ns wide clock pulse every 16 ns. Eight of these gates provide pre-sampler clocks staggered by 2 ns intervals.
3.3.3 Analog Memory and Presampler

The heart of the system is the set of AN1000H hybrids. Each of these memories have four signal inputs, two pre-sampler clock inputs, one data transfer clock input, and various control signals, as shown in Figure 3-8. The control signals are generated by TTL logic derived from the trigger signal and the data transfer clock for the first hybrid. The control signals must be delayed by 4 ns per hybrid. This is done by re-clocking them for each of the other hybrids, using a 47AC174 quad D flip-flop clocked from that hybrid’s data clock. All pre-sampler and data clocks are routed over 50 ohm controlled impedance traces whose paths are match to within 2/1000 inches to minimize external channel to channel skew. The internals of the hybrids exhibit some skew, as was shown earlier. It has been assumed that this skew is consistent from device to device. The signal inputs to the hybrids from the front-end amplifier are connected via pieces of semi-rigid .141" coax cable. The pieces are all the same length, except for variations calculated to compensate for the internal skew. The following table shows this:

<table>
<thead>
<tr>
<th>CHANNEL</th>
<th>RELATIVE DELAY</th>
<th>CABLE LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200 ps</td>
<td>+.9&quot;</td>
</tr>
<tr>
<td>2</td>
<td>0 ps</td>
<td>+.6&quot;</td>
</tr>
<tr>
<td>3</td>
<td>135 ps</td>
<td>+.6&quot;</td>
</tr>
<tr>
<td>4</td>
<td>260 ps</td>
<td>+1.2&quot;</td>
</tr>
</tbody>
</table>

The hybrids require numerous voltages (see Figure 3-8), all of which are duty cycle switched, some write cycle only, other stay on until the end of the read cycle. The read cycle also has its own clock and control signals which are derived from I/O circuitry on the microprocessor. This approach simplifies the hardware, and adds flexibility. The analog output is shared between all sixteen channels on all four hybrids. It consists of a differential amplifier with feedback, followed by a gain stage, which drives an A/D converter. The A/D converter has 5 volt and 10 volt inputs which are tied together to give a three volt full scale sensitivity. Its input is bipolar, which fits well with the output of the hybrids. No offset was included. The A/D converter has a built in sample and hold, and is very easy to use. Each read starts a new conversion. This does mean that a priming read must be given, or conversely, the first data element is discarded, and an extra read is done at the end. The 5 volt supply to the A/D converter had to be left on all the time because the converter is connected to the computer data bus. A bus buffer chip would have solved this had it been anticipated. The -12V supply is switched. Because the original intention was to switch the
+5V supply also, the reference voltage is filtered with a much smaller capacitor than the recommended 47 uf. This may need to be changed if it adds too much noise.

3.3.4 Microprocessor Interface and Communications Circuit with Host Computer

The microprocessor is a UT1750AR RAD hard RISC processor, as shown in Figure 3-9. It uses a Harvard architecture, having separate code and data memory. Only the lower 16 bits of the code address are decoded, with the first 32k being EPROM and the last 32k SRAM. The only thing in SRAM by design is a command jump table and a few small utility code fragments. These are copied from the EPROM. The remainder of the SRAM is reserved for extensions, including in-flight modifications. The data memory consists of 64k bytes of fast CMOS SRAM. The original intention was only to populate the first half. Most of this is used up by calibration tables and the data buffer. The second half is unused at present. An RS-232 serial port is built into the processor, but is of limited use because it does not generate interrupts. It is not needed in this application, although a level translator and connector have been added to allow for its use. If it is not used, it may be desirable to cut the power jumpers to this chip to lower the power consumption. I/O decoding is done with a 74AC138. In addition two 74HC259's provide latched output discretes which are set by writing (anything) to a specific I/O address, and cleared by writing to one address lower. U75 deserved particular mention because its clear input comes from U76-Q7. This means that when U76-Q7 is high, U75 acts like a decoder, with all outputs low, except for the one being addressed at a particular moment (if any). This allows short positive pulses to be output instead of levels, which is used to generate the fast two phase read clock used to skip cells. The other outputs of U75 have been inverted as necessary so that their cleared state would be compatible with this operation. Two 74HC573's provide a 16 bit status word. The low byte contains various handshake signals, while the upper byte is a copy of the read and write end signals of all four hybrids. These latter signals are not likely to be used in operation, but are useful for testing. U72 may be optionally removed for in flight use to conserve power, although its consumption is minimal, mostly derived from the extra capacitive loading it provides to the bus. Host communication is provided by U79 - U82 (74HC574's) and U83 (status latches). The protocol for this is described elsewhere, and a listing of the software is given in Appendix 1. Notice that U83 is level sensitive, and responds to the leading edge of the pulse. This means that pulses should be less than 500 ns wide so the micro does not have time to start a new operation before the old operation is completed externally.
3.3.5 Power Supply Duty Cycle Control

The power supply consists of three sections. The first is fixed voltages. The microprocessor subsystem needs a constant source of +5V. The 1 Ghz oscillator and the RS-232 level shifter need a constant source of +12 V. The second category is switched versions of input voltages. In addition to the above voltages, 6.5V, -5.2V, and -12V, are provided to the system. These voltages and +5V are switched by power mosfets as needed for the write loads and read loads. In each case the switching signal passes through a capacitor to insure that it will time-out in hardware in case of a microprocessor upset or glitch. The third category is switched derived voltages. The write loads need, in addition to the above voltages, +5.3 V, -2 V, -3.4 V, and -4 V, as well as two offset bias voltages. These are generated by power mosfets controlled by op amps. An LM385-2.5 is used as a precision reference. The original intention was to power it on and off for the 40 microseconds of the write cycle. However, it has several internal capacitors which cannot charge that rapidly, so it is left on, lightly loaded all the time. A copy of the write power control signal is current limited in a 110 ohm resistor and clamped by this regulator. This signal provides a regulated pulse to the op amps controlling the mosfets. When this signal is off, the op-amps shut off the mosfets, and when it is on they regulate the output of the mosfets to the desired voltage. Extra resistors and capacitors were required to provide closed loop stability of the op-amp circuits, while at the same time providing the rapid turn on and settling needed by the loads. The pulse which is clamped by the LM385-2.5 is also capacitor coupled to provide a hardware time-out. This is especially important here because the mosfets which are regulating must dissipate power and are not heat sunk for continuous dissipation. Also the write loads, especially the GaAs, are not heat sunk for continuous dissipation. IT IS VERY IMPORTANT NOT TO BYPASS THIS DUTY CYCLE LIMITING.

The decision of which voltages to require as inputs and which to derive was based on power consumption considerations. All constant power sources were made inputs, since any regulator drops in these would represent full time power consumption and heat losses. It is assumed that these voltages can be generated externally with high efficiency switching power converters. Second, all voltages that would represent a major drop from the next higher available voltage are required as inputs. The +6.5V input for instance would be very inefficient to derive from +12V. The remainder of the voltages are derived from these voltages, using the mosfet regulator/switches. The voltages are all write cycle loads having a 40 microsecond duration and a .1% duty cycle, so the losses in the regulator represent a very small fraction of the total power consumption.
3.4 Testing

3.4.1 Characterization of the AN1000H Hybrid

The hybrid is characterized in the test circuit shown in Figure 3-1. It is clocked at 62.5 MHz, which is the DCLK clock. The same clock rate is used for the presampler inputs, which are 2 ns wide. The difference between this test circuit and the final implementation of the 1 GHz Digitizer is that the latter consists of four hybrids, for a total of 16 inputs, while the signal to each input are delayed by a different amount, with the differences being exactly 1 ns apart. This provides 16 equally spaced (1 ns) signals every 16 ns (at 62.5 MHz). For initial testing the test circuit was connected to a 20 MHz function generator, set to produce SINE, SQUARE, and TRIANGLE waveforms at various frequencies. The analog memories (in the hybrids) is filled from this source (all four channels in parallel), and then read out (sequentially) using a 400 Msamples/s 6-bit digital oscilloscope to observe the waveform. Plots from the digital oscilloscope are given in Appendix 3. A full set of 256 samples requires 4.096 microseconds to collect, and sequentially reading back the 1024 samples takes 9 milliseconds. Therefore the frequency of the input signal is reduced by approximately 500 to 1.

In this procedure there are several limitations. The four inputs are resistively isolated, causing significant loss of signal, hence requiring maximum output from the function generator. Consequently, little room is left for baseline offset, and may have contributed to some clipping of the waveform. In addition, the oscilloscope resolution of 6 bits tends to exaggerate the noise content of the waveform. The signal would normally be sampled by a sampling A/D converter just before each transition, when it is most stable. The oscilloscope, however, shows all the waveform settling and glitches, which otherwise would not be digitized. To isolate the digital from the analog signals two planes are used, which are segmented. The test results are given in Appendix 3. Figure A3-1 shows the input signal (lower trace) captured by the test circuit. The upper trace is a frame sync signal which defines the beginning and end of the capture interval. The input signal was a portion of the rising half of a 100 KHz triangular wave. The noise on the signal shows the typical resolution and noise of the oscilloscope.

Figure A3-2 shows the output (readback) signal resulting from the input signal. Similarly, the upper trace shows the frame sync (read frame) signal used to define the beginning and end of the readback interval. Notice that there are four copies of the input signal, corresponding to the sequential readback of the four channels. These four copies should be identical. The end purpose of this test circuit is to quantify channel to channel
variations. In Figure A3-3 the same signal as in Figure A3-2 is shown using a faster timebase so that the first channel is given an expanded view. This allows a detailed look at signal variations and noise. Figure A3-4 shows an even faster timebase, so that the individual cell readout is clearly visible. Cell changes occur at 9-microsecond intervals. Notice in the latter two cases, while there is considerable noise on the signal, the largest spike occurs immediately after a cell transition followed by one occurring near the midpoint of the cell. The last microsecond before a transition remains relatively noise free. This is the point for optimum sampling and digitization.

A 2 MHz triangular input signal is shown in Figure A3-5, and its corresponding output waveform is shown in Figure A3-6. The three discontinuities appearing at the output are not errors, but represent the transition from one channel to another. In order to show this boundary, the input signal frequency is chosen not to be a multiple of the sampling window. With a 1 MHz input sinewave, the output is shown in Figure A3-7. The channel to channel transitions are even more clearly visible. In Figure A3-8, the same output is shown with a faster timebase to show only one channel output. The input sinewave frequency is increased to 20 MHz and the resulting output is shown in Figure A3-9. Since there is no output filtering, the beat frequency between this signal and the 62.5 MHz clock is perceptible in the output waveform. The top trace belongs to the sample/hold clock used by the A/D converter which samples the output. The sample phase ends with the negative edge of this clock, which occurs immediately before the cell transition of the hybrid. The first two clock pulses are part of the initialization process, thus do not present valid data.

From the test results several conclusions are drawn. First, the basic concept is viable. The test circuit captures and reproduce the waveforms. Second, the signal-to-noise ratio is adequate for the intended application. Third, there is some channel to channel offset which needs to be dealt with during calibration. In the remaining test results, given in Appendix 3, further test and analysis are performed with input signal frequencies up to 1 GHz. Output waveforms are displayed with errors due to offset and gain variations. Their corrected versions are also given.

3.4.2 System Tests

A number of tests and considerable debugging of the system were performed, although it was not 100% functional or tested when the funding ran out. The microprocessor and support logic and memory have been fully tested and are fully functional. The intended software has been written, a listing is given in Appendix 1. The startup and monitor code have been fully tested and debugged. The one addition which might be made in this area
would be self-test routines, such as RAM test, EPROM checksum, and as much I/O testing as
the design allows for. The conversion cycle routines have been partially tested. The cycle
progresses properly without any time-out or errors. The write cycle appears to initialize the
system properly, and respond to status properly. The read cycle seems to generate the proper
clock and control signals, operate the A/D converter properly, store data in the correct place
and scale and output it correctly. Because no calibration has been done and not useful data
retrieved, this section is less than fully tested. It has been determined that output data is
dependent on input signal, but with the limited amount of testing done, it was not obvious
that the output data was a reasonable representation of the input signal. The calibration
routine has not been tested at all, and is not guaranteed crash free at this time. A copy of the
C code (known to work) from which this routine was derived will be provided with the system
as an aid to understanding and debugging this routine. The C code is based on processing one
cell at a time from several files, each containing data at a different calibration point. The
1750 code by contrast was designed to accumulate the necessary intermediate values for each
cell, one calibration point at a time, and then transform the result to slope intercept form
afterward. The power circuitry has been tested. All power supply voltages achieve nominal
value within 5 microseconds after turn on. Turn off decay depends on the current drawn by
the load, and the number and size of the bypass capacitors across it. Current from the source
stops within a microsecond or two after the shutdown command. The hardware time-out
circuitry has also been tested and works properly. The signal input circuitry has been tested,
both with static power and duty cycled. The value of the feedback resistors has been
optimized for maximally flat frequency response. The trace between the first and second
stages was found to be inadequate at 350 MHz, and had to be supplemented with a wire. The
amplifiers furthest from the source were showing more attenuation. No testing has been
done on the coax path length. This testing cannot be done until calibrated signals are
available from the system. The test procedure should consist of calibrating the system (DC)
and then placing a sine or triangle wave of 20 to 50 MHz and full scale amplitude into the
system and examining the output data graphically. Timing errors will result in certain
segments of the resulting waveform being higher or lower than they should be. A short path
will cause a rising slope to appear higher than expected, and a falling slope to appear lower.
This is in contrast with gain errors which will always appear closer to (or further from) the
midpoint than expected, or offset errors which will always appear higher (or lower) than
expected. If every other channel is off, the 1 ns delay coax will need trimming. If random
channels are off, the coax between the input amplifier and that channel must be trimmed.

The clock and control signals have been examined. It is not possible with the
available equipment to fully check out the GaAs clock signals for duty cycle and phasing in a
pulsed system. This is because the available oscilloscope is not fast enough for real time sampling, and the pulsed signal was not long enough to be captured using repetitive equivalent sampling. The signals were verified to be present and on frequency, while the control signals were verified to be present and appropriately timed. The write cycle was measured at 40 microseconds, and the read cycle at 2.8 ms (this is sample size dependent). The read clock was checked, the fast clock was 840 ns and the slow clock was 3.3 microseconds. Both had the expected number of pulses for the skip and sample values used, and were checked with different values. The START-2 signals were check and found to be inserted properly. No check was made of any end signals, although since the software does not read out the trailing skipped values from channel four of the hybrids, no end signal would have been visible unless all channels were read.

3.4.3 Special Operations Instructions and Notes

The power inputs are assumed to turn-on simultaneously. If it is necessary to sequence them for external reasons, the order should be +5 and +12 first, -5 second and then +6.5. The -12 can be turned on anytime with or after +12. There is hardware and partial software support for software triggering of a conversion cycle. This is included for testing and calibration purposes. The variable SoftTrig is set to 0 (disabled) on a cold reset. It may be changed to non-zero using the monitor write instruction. When it is non-zero, the software does not wait for a hardware arm signal, and supplies an internal hardware trigger to start an acquisition cycle. This trigger is generated approximately 30 microseconds after the acquisition command is given by the host, when SoftTrig is non-zero. All testing if the system has been done using this method of triggering. It is possible, and may prove desirable, to eliminate the hardware ARM command. Since the host system is likely to have control over the subsystem responsible for the data to be collected, it could issue the acquisition command at the time the system needs to be armed. One easy way to implement this change is to separate the bits of the SoftTrig variable such that one bit is used to skip the ARM check and another to force a software trigger. Presently both tests simply check for a non-zero value. In Appendix 5 the description of hardware interface with the system computer is given.
4.0 CONCLUSIONS AND RECOMMENDATIONS

The design of the 1 GHz Digitizer for Space Based Laser Altimeter has been completed. A feasibility model was built, and partially tested. Testing was not completed because of the limited funding available at this time. However, a unique method of digitizing wideband signals (350 MHz) with very low average power consumption was developed and proven to be feasible. The heart of the system is a state-of-the-art hybrid memory chip (AN1000H) with built in presamplers. Sufficient hardware testing has been performed to give assurance that the developed technique is feasible. Because of the much lower power consumption achieved as compared to the initially expected amount, a continuous 1 GHz timing output (from a SAW oscillator) is afforded within a limited power budget. A software package to control various functions within the digitizer and to communicate with a host system computer was also developed. Unfortunately, the allowable budget would not permit debugging and testing of the software. Because the status of the digitizer is so close to completion, it is recommended to extend funding of this project to complete all necessary testing and packaging to obtain a working model. At this point technical risk is minimal if not nonexistent. At the completion of this feasibility model, the logical action is to develop a space qualified unit for future space exploratory missions.
;This file is designed to be assembled with CROSSI6 meta-Assembler
; version 2.0

;File created 9 August 1991
; by Wilton Helm
; last modified on 10 September 1991

;Contains stand-alone UT1750AR assembly language code in native
; mode to control a 1 Gs/s A/D converter system using Analytek
; hybrid analog memory modules in a 16-way interleaved fashion.

;Code memory map is as follows (addresses are 16 bit words):
; 0000h to 7F7Fh is EPROM containing startup and core code.
; 6000h to FFFFh is SRAM containing linking tables loaded from
;   EPROM and dynamically added code modules.
; 10000h and up are not decoded and wrap into above areas.

;A separate 64k x 16 bit data address space is populated with
; SRAM, and used as described by data equates below.

;A separate I/O space is decoded as described by I/O equates below.

;Special note should be given to the hardware signals O1Y and O2Y
; which are a two-phase, non-overlapping readback clock
; generated in software. There are two modes of generating this
; signal. For skipping cells, a high speed mode is used where
; U75 is placed in non-latching decode mode (by asserting its
; CLEAR pin). Writes to O1YHi and O2YHi in this mode generate
; pulses on the respective lines. For reading cells, the CLEAR
; is removed and O1Y and O2Y become latching outputs which must
; be set by O1YHl and O2YHl respectively, and cleared by O1YLp
; and O2YLp respectively, These commands are interspersed with
; suitable timing waits, A/D reads and loop control commands.

;Note that the assembler used makes no distinction between
; code and data objects. Both share the same symbol table space.
; It is up to the programmer to use symbols appropriately. Code
; space symbols are used only with LRI and STRI instructions.

;The assembler uses one possibly non-standard mnemonic form. I
; could not see a way to differentiate between:
; sar R0, R4 ;shift right 4 places
; and
; sar R0, R4 ;shift right by amount specified
; ;in register 4
; so I adopted the following convention for the former case:
; sar R0, R+4 ;shift right 4 places
; and for consistency:
; sar R0, L+6 ;shift left 6 places

;00000
;00000

WDLN 2 ;word length (width in bytes) for the 1750
CPU "UT1750AR.TBL" ;table of mnemonics to use for 1750

;CODE SPACE SRAM EQUATES
;000008000 = ImgDst: equ 8000h ;start of RAM code space
;000008000 = CadJmp: equ 8000h ;address table for command execution
;000008010 = CadRtn: equ 8010h ;address to return to after command execution
;000008011 = WarmLink: equ 8011h ;address of warm start routine.
;000008012 = Postchk: equ 8012h ;routine to post checksum changes
;00000FFFF = CRAME: equ 0FFFFh ;end of RAM code space
;DATA MEMORY EQUATES

00000000 = CodeChk: equ 0h ;32 bit sum of code RAM
00000002 = Samples: equ 2h ;number of samples to keep (per chan)
00000003 = Skip: equ 3h ;number to skip before starting (per chan)
00000004 = SoftTrig: equ 4h ;non-zero if software trigger should be given
00000005 = CalFlag: equ 5h ;non-zero if calibration in process
00000006 = CalV_m: equ 6h ;summation of correct calibration values
00000007 = CalCnt: equ 7h ;count of number of calibration values used
00000008 = CalSkH: equ 8h ;Hold value for skip, used during calib.
00000009 = CalSaH: equ 9h ;Hold value for samples, used during calib.

00001000 = ADBuf: equ 1000h ;m from y = m * x + b
00002000 = ADGain: equ 2000h
00003000 = ADOfs: equ 3000h ;b
00004000 = CalSums: equ 4000h ;used for curve fitting, overlays gain
00005000 = CalSqr: equ 5000h ;used for curve fitting, overlays offset

;I/O ADDRESS EQUATES

0000040 = DMA: equ 40h ;DMA write and read word to host
0000050 = Status: equ 50h ;read to get system peripheral status:
0000060 = CTS: equ 60h ;b0 = 1 when host is giving command
0000070 = InEmpty: equ 70h ;b1 is RS-232 handshake (CTS)
0000080 = OutFull: equ 80h ;b2 is DMACH (0 = DATA read by host)
0000090 = ADBusy: equ 90h ;b3 is D Mare (0 = DATA ready to read)
0000100 = ConvBusy: equ 10h ;b4 = 0 if A/D conversion finished
0000110 = AcqBusy: equ 11h ;b5 = 0 at end of read cycle
0000120 = NotArm: equ 12h ;b6 = 1 when triggered and 0 at end of write cycle
0000130 = NotEnd1: equ 13h ;b7 = 0 for ARM system request
0000140 = NotEnd2: equ 14h ;b8 is *END2-1 from hybrid
0000150 = NotEnd3: equ 15h ;b9 is *END2-2 from hybrid
0000160 = NotEnd4: equ 16h ;b10 is *END2-3 from hybrid
0000170 = NotEnd1: equ 17h ;b11 is *END2-4 from hybrid
0000180 = NotEnd2: equ 18h ;b12 is *END1-1 from hybrid
0000190 = NotEnd3: equ 19h ;b13 is *END1-2 from hybrid
0000200 = NotEnd4: equ 20h ;b14 is *END1-3 from hybrid

;The following addresses are activated by writing. The value written is ignored,
;only the address matters.

0000050 = RstWrEn: equ 50h ;resets the write logic which is not needed during read
0000051 = RstWrDis: equ 51h ;releases reset
0000052 = RstWrEn: equ 52h ;resets write logic which must not be reset during read
0000053 = RstWrDis: equ 53h ;releases reset
0000054 = WrPwrDis: equ 54h ;removes power from write logic
0000055 = WrPwrUp: equ 55h ;applies power to write logic
0000056 = RdPwrDis: equ 56h ;removes power from read logic
0000057 = RdPwrUp: equ 57h ;applies power to read logic
;
; (NOTE: RdPwr is needed for Write as well)
0000058 = TrigDis: equ 58h ;prevents trigger signal from starting a conversion
0000059 = TrigEn: equ 59h ;allows trigger signal to start a conversion
000005A = StartEn: equ 5Ah ;software trigger
--- 000005B = StartDis: equ 5Bh
--- 000005C = StatOff: equ 5Ch
--- 000005D = StatOn: equ 5Dh
--- 000005E = FastRead: equ 5Eh
--- 000005F = SlowRead: equ 5Fh

; end of software trigger

; indicates next word to host is data

; indicates next word to host is status

; turns 6X writes into pulses
; this allows O1Y and O2Y to operate
; faster for skipping cells not needed in hybrids

; changes 6X writes into latched data
; this mode is needed for all
; handshaking to hybrids, and for actual
; data reading

; reading this address resets (1) status b3

; more write strobe addresses

; releases RST2 reset

; resets hybrid read logic

; sets O1Y clock line low

; sets O1Y clock line high
; during fast read, pulse this for O1Y

; sets O2Y clock line low

; sets O2Y clock line high
; during fast read, pulse this for O2Y

; trailing edge of START2-1 signal to hybrid

; leading edge of START2-1 signal

; trailing edge of START2-2 signal to hybrid

; leading edge of START2-2 signal

; trailing edge of START2-3 signal to hybrid

; leading edge of START2-3 signal

; trailing edge of START2-4 signal to hybrid

; leading edge of START2-4 signal

; reads A/D converter and starts next conversion

; write this address to generate a
; software interrupt - used to disable ints.

; REGISTER USEAGES

; R0 Short term temporary use

; R16 Subroutine return address register

; R18 Interrupt return address register

; (return registers need only be 16 bit in this system,
; since there is only 64k of code space.)

; PROGRAM CONSTANTS

; change later to allow desired ints.

; ERROR CODES

; OK status word

; Reset caused a warm start

; Reset caused a cold start

; Command completed

; Data rcvd when command expected

; Command rcvd when data expected

; Command rcvd before requested data sent

; Command rcvd during A/D cycle

; status word: no trigger after arm

; Analog write cycle failure

; A/D cycle completed, but no data because
; calibration cycles have been requested

00000 org 0
Init: otr ACC, RstWrEn ; value in ACC doesn't matter
  otr ACC, RstWrOEn ; set all resets and power down
  otr ACC, WRpwrDn
  otr ACC, TrigDis ; disable triggering
  otr ACC, StartDis ; and software trigger
  otr ACC, StatOff ; no status word ready
  otr ACC, FastRead ; this resets RstRDIs, O1YLw, O2YLw,
                   ; St2-1Dis, St2-2Dis, St2-3Dis, St2-4Dis
  otr ACC, SlowRead ; undo above resets
  otr ACC, RstREn ; Leave Read reset active
  inr ACC, TAH ; stop timers
  inr ACC, TBH

; set up any hardware needed for interrupts
  inr RO, STATUS ; get status word and check Command bit
  br NE, cold ; if set, force cold start

cold: mov RO, Image ; block move RAM constants from EPROM
  mov R1, ImgDst
  mov R0, RO
  mov R1, R2
  mov R0, R1
  str R2
  add R0, 1
  cmp R0, ImgEnd
  br LT, coldlp
  add R1, 1 ; (used if br)
coldlp: mov ACC, R0
  mov ACC, R1
  str R2
  add R0, 1
  cmp R0, ImgEnd
  br LT, coldlp
  mov RO, WarmStat
  lrt RO, call R16, RO ; execute It (will not return ; but will jump (call) warmend)
  mov RO, 0
  str R0, Skip
  str R0, SoftTrig
  mov R0, ColdStat
  jc X, startcom
  nop

warm: mov ACC, WarmLink ; get address of user warm start
  lri RO
  call R16, R0 ; execute it (will not return ; but will jump (call) warmend)
  mov RO, WarmStat

warmend: mov RO, WarmStat

startcom: ; omit for now
  otr ACC, ENBL ; turn on interrupts

  otr ACC, StatOn ; set status flag
  otr R0, DMA ; output status word regardless of
  otr R0, DMA ; handshaking
; REGISTER USAGE BY MONITOR
; R1  low portion of address being accumulated
; XR2 starting address for range
; XR4 ending address for range
; R6 non-zero if address Js in code space
; R7 data portion of command word
; ACC command / jmp address

0004F 8020  monitor:  mov  R1,0 ; set low portion to zero initially
00050 8241  mov  XR2,1 ; set starting address of 1
00051 8280  mov  XR4,0 ; and ending address of 0 (done)
00052 80C0  mov  R6,0 ; set to data space
00053 14F0050  nextword:  inr  R0,Status ; set low portion to zero initially
00055 D81D  tbr  R0,InEmpty ; see if data available
00056 FE9C  br  me,nextword ; no, wait
00057 D81F  tbr  R0,Cmd ; yes, test for command sequence
00058 14F0040  inr  R7,DMA ; get command
00059 0007  jc  eq,statout ; not command, error
0005A 7D1F0093  mov  R0,IlData ; (error msg - needed if jc)
0005C 8005  mov  R0,R7 ; copy to command area
0005D 0007  and  R7,OFFFh ; strip data section
0005E 40FF0000  and  R0,0F000h ; strip command section
00060 E803  scr  R0,L+4 ; shift command down
00062 E803  mov  ACC,R0 ; form table address
00063 03E0  or  ACC,CmdJump ; get Jump address
00064 47F80000  call  R0,R0 ; go there, does not return here

00066 84F0  CodeLow:  mov  R6,15 ; set code space flag (0)
00067 7F9FFE8  jc  x,nextword ; get rest of command
00068 0027  mov  R1,R7 ; set 12 low bits (jump taken)

00069 80C0  DataLow:  mov  R6,0 ; set data space flag (1)
0006A 7F9FFE4  jc  x,nextword ; get rest of command
0006B 0027  mov  R1,R7 ; set 12 low bits (jump taken)

0006C 80C0  StHJRng:  mov  R2,R7 ; place upper bits in starting address (2)
0006D 8060  mov  R3,0
0006E 2E5C  slr  XR2,R+4 ; shift right 4
0006F 4641  or  XR2,R1 ; and combine with low bits
00070 7F9FFDD  jc  x,nextword ; get rest of command
00071 8020  mov  R1,0 ; clear low bits for end adr (jump taken)

00072 0047  StHJrd:  mov  R2,R7 ; place upper bits in starting address (3)
00073 8060  mov  R3,0
00074 E25C  slr  XR2,R+4 ; shift right 4
00075 4641  or  XR2,R1 ; and combine with low bits
00076 7F9FF003F  jc  x,MonRd ; go read it and put on bus
00077 0292  mov  XR4,XR2 ; set end = beg for one word (jump taken)

00078 0047  StHWr:  mov  R2,R7 ; place upper bits in starting address (4)
00079 8060  mov  R3,0
0007A E25C  slr  XR2,R+4 ; shift right 4
0007B 4641  or  XR2,R1 ; and combine with low bits
0007C 7F9F0055  jc  x,MonWr ; wait for data and write it
; CMD 5
Execute: mov R2,R7 ;place upper bits in starting address (5)
            mov R3,0
            slr XR2,R+4 ;shift right 4
            or XR2,R1 ;and combine with low bits
            mov R16,monitor ;set up return address
            call R4,XR2 ;indirect jump, return is:
;               CALL R16,R16 ;which will re-enter the monitor

; CMD 6
EndHdr: mov R4,R7 ;place upper bits in ending address (6)
            mov R5,0
            slr XR4,R+4 ;shift right 4
            or XR4,R1 ;combine with low bits (jump taken)

; CMD 7
EndHwr: mov R4,R7 ;place upper bits in ending address (7)
            mov R5,0
            slr XR4,R+4 ;shift right 4
            or XR4,R1 ;combine with low bits (jump taken)

; CMD 8
IONd: jc x,MonRd ;go read it and put on bus
       or XR4,R1 ;(jump taken)

; CMD 9
IONr: inr RO,R7 ;read I/O, and put on bus (jump taken)
            tbr RO,Status ;see if data available (9)
            tbr RO,InEmpty ;no, wait
            br ne,IONr ;yes, is it data or command
            jc x,MonWr ;command, error - don't continue
            mov RO,NoData ;(error msg - needed if jc)
            inr RO,DMA ;data, read it
            jc x,monitor
            otr RO,R7 ;output data to I/O address (jump taken)

; CMD A
SetSkp: slr R7,R+4 ;truncate modulo 16
            str R7,Skip ;and divide by 16 to get
            jc x,monitor ;skip value per channel
            nop

; CMD B
ManAcq: add R7,15 ;round up modulo 16 (11)
            slr R7,R+4 ;and divide by 16 to derive
            cmp R7,0
            le,ManLim ;don't allow 0 or negative value
            cmp R7,100h
            le,ManOk ;or more than 4096 (/16)
            nop
            mov R7,7 ;default to 128 (7 * 16) if invalid
ManLim: str R7,Samples ;number of cells per channel
            jc x,Arm ;start conversion cycle
; CMD C
AutoAcq: equ monitor

MonRd: cmp XR2,XR4 ; more data to send?
        jc GT,statout ; no, return completed status
        mov R0,Cmpltd ; (used if jc)
or R6,R6 ; code or data?
        br eq,MonRDta ; data
        mov ACC,XR2 ; code, read next word to send

MonWr: inr R0,Status ; get status word
        tbr R0,InEmpty ; see if data rcvd
        br ne,MonWr ; no, wait for it
        tbr R0,Cmd ; yes, data or command?
        jc ne,statout ; command, error
        mov R0,NoData ; (error msg - needed if jc)
        or R6,R6 ; code or data memory?
        br eq,MonWRdta ; data
        mov ACC,XR2 ; code, write word
        str R0
        br x,MonWRnt

MonNxt: add XR2,1 ; ready for next pass
        cmp XR2,XR4 ; more to come?
        jc LE,MonWr ; yes, continue
        mov R0,Cmpltd ; no, return completion

statout: mov R2,0 ; timeout in case host not unloading buffer
        inr R1,Status ; get status word
        tbr R1,OutFull ; buffer empty?
        jc eq,statol ; yes, proceed
        add R2,1 ; no, increment timer (65536 is fail)
        jc ne,statout ; 100 ms. time limit, not out
        nop ; ran out, force output

statol: otr R0,StatOn ; place in status mode
        otr R0,DMA ; send data
        jc x,monitor ; status sent, return to monitor
        nop ; as a recovery

; R0 temp
; R1 temp
; R8 I/O address of O1YH1
; R9 I/O address of O2YH1 or O2Ylow as needed
; R10 Buffer address pointer
; R11 Channel number (1 - 4) within a hybrid
; R12 Hybrid number (1 - 4)
; R13 Temp cell counter for skip and read

000FE 041F0004          Arm:             lr R0,SoftTrig ; If Software triggered, don't
00100 4400 or R0,R0 ; wait for hardware arm
00101 FE8E br NE,PwrUp
00102 0000 nop
00103 141F0050 inr R0,Status ; Wait for hardware Arm status
00105 D818 tbr R0,NotArm
00106 FD09 br eq,PwrUp ; Arm request received
00107 D81D tbr R0,InEmpty ; No Arm request yet, check for abort
00108 FE95 br ne,Arm ; No data ready, loop
00109 D81F tbr R0,Cad ; Data ready, is it command
0010A 79FFFE3 jc ne,statout ; Command, give abort status
0010C 8008 mov R0,Aborted
0010D 79FFFE0 jc x,statout ; Data, abort with data error
0010F 8005 mov R0,III1Data
00110 1BF00055 PwrUp: otr ACC,WrPwrUp ; Turn on power to all of A/D system
00112 1BF00057 otr ACC,RstWroEn ; Reset all of A/D system
00114 1BF00050 otr ACC,RstWroEn
00116 1BF00052 otr ACC,RstWroEn
00118 1BF00058 otr ACC,TrigDis ; disable triggering
0011A 1BF0005B otr ACC,StartDis ; disable SW trigger initially
0011C 1BF0005E otr ACC,FastRead ; Reset read latches
0011E 101FFFD2 movc R0,-46 ; wait 20 us for power to settle
00120 FC9F ArmLp: br 1t,ArmLp ; loop time 4 cycles = 1/3 us.
00121 A001 add R0,1 ; (less 16 cycles allowed for setup below)

00122 1BF0005F Enable: otr ACC,SlowRead ; end read latch reset
00124 1BF00061 otr ACC,RstEn ; hold read logic reset
00126 1BF00053 otr ACC,RstWroDis ; end write reset condition
00128 1BF00051 otr ACC,RstWroDis
0012A 1BF00059 otr ACC,TrigEn ; allow triggering
0012C 041F0004 lr R0,SoftTrig ; see if SW trigger needed
0012E 4400 or R0,R0
0012F FD07 br EQ,Trigger ; no
00130 0000 nop
00131 101FFFE2 movc R0,-30 ; yes, wait 10 more us.
00133 FC9F EnLp: br 1t,EnLp
00134 A001 add R0,1
00135 1BF0005A otr ACC,StartEn ; and then trigger it
00137 003FFD12 Trigger: mov R1,-750 ; start 1 ms loop waiting for trigger
00139 141F0005 TrigLp: inr R0,Status ; 16 cycle loop
0013B D819 tbr R0,AcqBusy ; see if triggered
0013C 79FF0014 jc NE,Collect ; yes
0013E A021 add R1,1 ; no, count time. Time out?
0013F FC99 br LT,TrigLp ; no, loop
00140 8009 mov R0,NoTrig ; yes, exit with error code
00141 1BF0005B shutdn: otr ACC,StartDis ; end software trigger
00143 1BF00058 otr ACC,TrigDis ; disable triggering
00145 1BF00050 otr ACC,RstWroEn ; Reset and power down everything
00147 1BF00052 otr ACC,RstWroEn
00149 1BF00061 otr ACC,RstEn
0014B 1BF00054 otr ACC,WrPwrDn
0014D 1BF00056 otr ACC, RdPwrDn
0014F 79FFFE9E jc X,statout
Collect: mov Rl, -17 ; start 20 us. timing
ColLp: inr RO, Status ; 14 cycle loop
trb RO, AcqBusy ; still acquiring?
add RL.1 ; yes, time it
sub RO, R12 ; number of hybrids
mov R12, 4 ; number of hybrids
mov R10, ADBuf ; set pointer to buffer
mov R0, 01YHI ; load 01YHI address into register
NewHyb: mov R0, RdEn+2 ; set up start address for this hybrid
Unload: otr ACC, RsLnROEn ; reset write logic
otr ACC, TrigDis ; prevent further triggering
otr ACC, WrPwrDn ; turn off write power
otr ACC, RstRDls ; end Read Reset
otr ACC, R8, O1YHI ; load O1YHI address into register
otr ACC, R0 ; initiate start on channel
otr ACC, O1YHI ; do second cycle
sub R0, 1 ; convert address to disabled
otr ACC, 02YHI ; end start pulse
otr ACC, R0 ; send out start pulse
; which ends here, may be able to be simplified.
; some of the Analytec documentation indicates that
; the start pulse is positive edge triggered inside the
; analog memory chip. If this is so, the above code
; need only trigger it initially and the go directly
; into fast read mode to skip unused cells at the beginning.
; It would look like this:

otr ACC, FastRead ; this eliminates the need for
otr ACC, R0 ; RstRDls as well
otr ACC, R0 ; send out start pulse

; then skip 0 or more cells using fast read, which is already
; set up.
mov R11, 4 ; number of channels per hybrid
mov R13, skip ; number of cells to skip initially
or R13, R13 ; test for zero
SkipAgn: jc eq, Read
褚 R13, R13 ; test for zero
SkipLp: jc eq, Read
褚 R13, R13 ; test for zero
褚 ACC, R8 ; phase 1 pulse
br NE, SkipLP ; loop as needed (2 cy)
 otr ACC, R9 ; Phase 2 pulse, even if br taken (3 cy)
 ; this loop is 10 cycles long, so
 ; skips a cell every 833 ns.

Read: otr ACC, SlowRead ; return to slow mode
 otr ACC, R9 ; start a cycle to prime the A/D
 mov R9, 02YLow ; set up address needed by slow read
 otr ACC, 01YL

otr ACC, R9 ; (1 + 3 cy)

lnr R0, AD ; (4 cy)
 otr ACC, R9 ; (2 cy)

otr ACC, R9 ; (2 cy)

mov R0, 01YL

otr ACC, R0 ; Phase 1Y High (3 cy)

nop ; waste time (2 cy)

nop ; waste time (2 cy)

mov R0, 01YL ; (2 cy)

otr ACC, R0 ; Phase 1Y Lo (3 cy)

RPh0: mov R0, 02YH

mov R1, AD ; (2 cy)

RPh1: mov R0, 02YH ; (2 cy)

otr ACC, R0 ; Phase 2Y Hi (3 cy)

RPh2: inr R0, R1 ; Read A/D & restart (3 cy)

str R0, R10 ; Store value in buf (3 cy)

NOTE: full 16 bits stored, top four are not valid and must be stripped before use.

otr Acc, R9 ; Phase 2Y Low (3 cy)

jc NE, RPh0 ; jmp based on R13 (4 cy)

add R10, 1 ; update bufad (2 cy) always

RPh3: sub R11, 1 ; see if more channels in this hybrid

jc EQ, NxtHyb ; no, go to next one

nop ; yes

RPh4: mov R0, Samples ; Number of cells skipped is 256 -

sub R13, RO ; the number to be read, - 1 for bogus

RPh5: mov R13, 255 ; read used to unload A/D last time

sub R13, R0 ; part of the skipped area will be the

jc LT, RPh6 ; if all cells to be read, don't skip

RPh6: x, SkipAgn ; remainder of this channel, the rest

RPh7: NxtHyb: sub R12, 1 ; will be the first of the next chan.

jc NE, NewHyb

nop

RPh8: ConvEnd: otr ACC, RstREn ; reset read logic and

otr ACC, RstWriEn ; remaining write logic,

otr ACC, RdPwrDn ; and power it down

lr R0, CalFlag ; Is calibration in progress?

jc NE, statout ; yes, return status

(jc)

; DO NOT attempt to correct data and

; return it if calibration is in process.

; The results will be at best garbage.

; At worst it might cause an overflow

; or underflow error.

Now clean up and output the values using the following pseudo-code:
FOR cell = 0 to Samples - 1 ; cell within each channel
FOR chan = 0 to 15 ; channels multiplexed
    data = ADBuf[chan * Samples + cell] ; raw data
    gain = ADGain[chan * 256 + cell + skip] ; gain correction
    ofs = ADOfs[chan * 256 + cell + skip] ; offset correction
    res = (data * gain) / 4096 - offset ; corrected value

Notes:
1. Gain and offset values exist for all cells, data values exist only for requested cells, thus providing the tables with two different size multipliers.
2. Values were collected in Hybrid unload order for maximum unload speed, and thus minimum power consumption. The data is interleaved throughout the 16 channels on the four hybrids, and thus must be retrieved in a different order so as to come out in true time order.
3. The gain values have been multiplied by 4096 to allow scaling both upward and downward. The multiply should be done in a 32 bit register and then the result divided by 4096 (or shift right 12).
4. The offsets are subtracted off after scaling, not before. This affects the way the offsets are calculated.

        001D0 8100         mov     R8, 0 ; cell number (0 to Samples - 1)
        001D1 053F0002     lr      R9, Samples ; number of cells collected per channel
        001D3 055F0003     lr      R10, Skip
        001D5 8160         UnlLp:  mov     R11, 0 ; channel number (0 to 15)
        001D6 018B         UnlLp2: mov     R12, R11 ; calculate index to ADBuf
        001D7 6D89         muls    R12, R9 ; chan * Samples
        001D8 2188         add     R12, R8 ; + cell
        001D9 219F1000     add     R12, ADBuf ; + base address
        001DB 042C
        001DC 421F0FFF     and     XR0, OFFFh ; mask garbage and setup double word
        001DE 018B         mov     R12, R11 ; calculate index to corrections
        001DF E187         slr     R12, L-8 ; chan * 256
        001E0 2188         add     R12, R8 ; + cell
        001E1 218A         add     R12, R10 ; + skip
        001E2 219F2000     add     R12, ADGain ; + base address
        001E4 05AC         lr      R13, R12 ; get gain factor
        001E5 6E0D         muls    XR0, R13 ; scale it
        001E6 E614         sar     XR0, R+12 ; correct shift
        001E7 219F1000     add     R12, ADOfs - ADGain ; new base address, for offset
        001E9 05AC         lr      R13, R12 ; get offset correction
        001EA 302D         sub     R1, R13 ; subtract off, only 16 bit needed here
                             ; now send word to host
        001EB 141F0050     UnlSnd:  inr     R0, Status ; get status word
        001ED D81D         tbr     R0, InEmpty ; see if data rcvd
        001EE 7D1F0012     jc      eq, unlerr ; yes, should not have been
        001F0 D81C         tbr     R0, OutFull ; buffer empty?
        001F1 FE99         br      ne, UnlSnd ; no, wait
        001F2 BD6F         cmp     R11, 15 ; more channels at this cell offset?
        001F3 183F005C     otr     R1, StatOff ; place in data mode
        001F5 183F0040     otr     R1, DMA ; send data
        001F7 7C9FFFD7     jc      LT, UnlLp2 ; yes (more channels)
        001F9 A161         add     R11, 1 ; next channel (if jc)
        001FA A101         add     R8, 1 ; next cell offset
        001FB 3D09         cmp     R8, R9 ; end of cells?
        001FC 7C9FFFD7     jc      LT, UnlLp ; no, continue
        001FE 8004         mov     R0, Cmpltd ; yes, finished
Routine to generate table of gains and offsets. There are three ways to call this routine. The first call (which notices that CalFlag is not set) sets it and initializes the work areas. While CalFlag is set, this call cannot be repeated. The A/D cycle will not return data when CalFlag is set, only a completion code. This initial call also sets the skip value to zero and Samples to 256 to provide a complete calibration table. (This may not be a good idea, as timing and related temperature variations may make calibration under actual skip and sample conditions more accurate.)

Subsequent calls to calibrate should occur after each A/D cycle. They provide the value (0 to 4000) representing the voltage used in that calibration cycle. These calls cause a summation of data values to occur which will be used later by the curve fitter. The actual values given will determine the scaling of subsequent data. i.e. 0 to 1000 for 0 to 1 V would yield 1 mV per count.

The final call to calibrate is done after all calibrate cycles have been completed, and the summations have been done. The call is done with a data value of 4095 (0FFh) which signals completion of the calibration. A least squares fit is done, fitting the data to the equation \( y = m \cdot x + b \). The values of \( m \) and \( b \) are used as AdGain and AdOifs respectively. CalFlag is again cleared, allowing normal processing to resume.

Note that this routine destroys AdGain and AdOifs by using the same area of memory as workspace. It should not be aborted without completing.

; REGISTER USEAGE BY CALIBRATION ROUTINE
; RO through R15 used temporarily
Calib: lr R0, CalFlag ; is calibration in process?
    00218 041F0005
    0021A 7E9F002D
    0021C 001F0000
    0021E 008F
    0021F 089F0005
    00221 0800

Calizer: str R4, R1 ; clear array area
    00222 089F0006
    00224 089F0007
    00226 003F0000
    00228 005F0000
    0022A 007F0000
    0022C 0881

CalCont: cmp R7, OFFh ; termination code?
    0022D A021
    0022E 0882
    0022F A041
    00230 0882
    00231 A041
    00232 0883
    00233 A061
    00234 0883
    00235 B001
    00236 FE95
    00237 A061
    00238 043F0003
    0023A 083F0008
    0023C 089F0003
    0023E 041F0002
    00240 081F0009
    00242 001F0100
    00244 081F0002
    00246 7F9FFE A7
    00248 8004

CalP: lr R5, R4 ; get raw value
    00249 3CF0FFF
    0024B 003F2000
    0024D 005F3000
    0024F 7D1F002F
    00251 007F5000
    00253 049F0006
    00255 2087
    00256 089F0006
    00258 049F0007
    0025A A081
    0025B 089F0007
    0025D 009F1000
    0025F 04A4

    00260 40BF00FF
    00262 04C1
    00263 20C5
    00264 08C1
    00265 A021
    00266 0345
    00267 6F45
    00268 0502
    00269 A041
    0026A 0522
    0026B 231A
    0026C 0922
    0026D B041
str R8, R2 ; ready for next pass
add R2, 2 ; calculate raw value * correct value
mov XR10, R5
muls XR10, R7
lr R8, R3 ; add to total
add R3, 1
lr R9, R3
add XR8, XR10
str R9, R3
sub R3, 1
str R8, R3
sub R0, 1 ; loop counter
jc NE, CalLP ; repeat until done
add R3, 2 ; (if jc)
add R8, R3
jc X, statout ; then exit with completion code
mov R0, Cmpltd ; (if jc)
mov R4, AdGain ; set up pointers specific to this part
mov R5, AdOfs
mov R6, CalCnt
lr R6, CalVSum
lr R8, R1 ; get sum of data
add R1, 1 ; ready for next
mov R11, R8 ; copy it
mov R10, 0 ; (unsigned to double precision)
divs XR10, R6 ; divide by number of points.
mov R9, R11 ; R9 is an important intermediate
mov R11, R8 ; sum * R9
mov R10, 0
muls XR10, R9
lr R12, R2 ; subtract from sum of squares
add R2, 1
lr R13, R2
add R2, 1
sub XR12, XR10 ; denominator of gain term
add R3, 1
lr R10, R3 ; get constant vector
add R3, 1
mov R15, R9
mov R14, 0
muls XR14, R7 ; subtr. sum of correct values * R9
sub XR10, XR14 ; numerator of gain term
or R10, R10 ; if numerator isn't too large,
br NE, CFSD
nop
sar XR10, L+12 ; multiply it by 4096 to scale gain
br X, CFSC
nop
sar XR12, R+12 ; otherwise divide denominator by 4096
divs XR10, XR12 ; gain
str R11, R4 ; and store it
add R4, 1
muls XR10, R8 ; gain * sum of data
sar XR10, R12 ; undo scaling for this
mov XR12, R7 ; sum of correct values
sub XR12, XR10
divs XR12, R6 ; divide by number of data samples
str R13, R5 ; offset
sub R0, 1 ; loop counter

;Routine to calculate checksum of code RAM to determine if it
;has been corrupted.

;This routine should be called, followed by "str XRO, CodeChk"
;when the system is cold started, and after any modification
;to the code RAM (stri to 8000h to FFFFh). The code write
;command does NOT do this automatically. It is necessary to
;do a code execute at address 8011h after code writes.
;Chksum may be called anytime the integrity of the code RAM is
;in question. The value in XRO should agree with the value
;stored in CodeChk. Chksum is automatically called before a warm
;start and if code RAM doesn't check, a cold start is done
;instead.

;Destroys ACC and R2, returns to R16

002C0 03FF8000 Chksum: mov ACC, R16 ;where to check
002C2 8200 mov XRO, XRO ;initial sum
002C3 845F chklp: lri R2 ;value at address
002C4 2202 add XRO, R2 ;add to sum
002C5 964E inr XRO,2,ACC ;copy to 16 bit register so compare will work
002C6 3C7FFFFF cmp R3, CMPMe ;done?
002C8 FE9A br NE, chklp ;no, cont
002C9 A3E1 add ACC, R2 ;next place to check
002CA 0E31 call R16, R16 ;return with results in XRO

;RAM IMAGE DATA TO LOAD AT 8000h in code area

Image:

002CB
dwn CodeLow ;CMD = 0 (0xxxh) - set CODE flag & 12 low bits
002CC dwn DataLow ;CMD = 1 (1xxxh) - set DATA flag & 12 low bits
002CD dwn STHIrng ;CMD = 2 (2xxxh) - hi bits, form start adr
002CE dwn STHIrd ;CMD = 3 (3xxxh) - hi bits, form adr and rd 1
002CF dwn STHIwr ;CMD = 4 (4xxxh) - hi bits, form adr and wr 1
002D0 dwn Execute ;CMD = 5 (5xxxh) - hi bits, form adr and ex
002D1 dwn EndHIrd ;CMD = 6 (6xxxh) - hi bits, form end adr & rd rng
002D2 dwn EndHIwr ;CMD = 7 (7xxxh) - hi bits, form end adr & wr rng
002D3 dwn IORd ;CMD = 8 (8xxxh) - 12 bit adr and Input word
002D4 dwn IOWr ;CMD = 9 (9xxxh) - 12 bit adr and Output word
002D5 dwn SetSkp ;CMD = 10 (Axxxh) - 12 bit skip count (mod 16)
002D6 dwn ManAcq ;CMD = 11 (Bxxxh) - 12 bit cvnt cnt (mod 16) and arm
002D7 dwn AutoAcq ;CMD = 12 (Cxxxh) - reserved for auto skip and cnt
002D8 0218 dwn Calib ;CMD = 13 (Dxxxh) - calibration routine
002D9 04F dwn monitor ;CMD = 14 (Exxxh) - reserved
002DA 00F dwn monitor ;CMD = 15 (Fxxxh) - reserved
002DB 004F dwn monitor ;return address for execute, etc
002DC 0048 dwn warmend ;link address for warm start
002DD 0E3F02C0 call R16, Chksum ;callable routine to update checksum
002DF 081F0000 str R0, CodeChk ; resides at 8012h
002E1 083F0001 str R1, CodeChk+1
mov ACC, CmdRtn
lri R16
call R16, R16 ; (return to monitor)
APPENDIX #2
Clock Generator Unit

Program Review

Binneg Lao
Carl Price

Dec 07, 1989
CLOCK GENERATOR - MODULE A.

FO

<250MHz - 1GHz>
SINGLE ENDED
ECL INPUT

TKAD10C
+TRK
-TRK
ADC #1
+CLK
-CLK

TKAD10C
+TRK
-TRK
ADC #2
+CLK
-CLK

GND -12VDC
AMERASIA: CLOCK GENERATOR - MODULE A

PCB = 0.014 INCH THICKNESS
50 OHM = SIGNAL LINE VSWTH
50 OHM = GROUND LINE VSWTH
ALL SIGNAL LINES ARE 50 OHM EXCEPT AS SPECIFIED

NOTE: Decouple each IC with a 0.01µf capacitor.
Total Power Consumption
= 2.1 Watts @ -12Vdc Power Supply.

Fo = 250MHz to 1GHz
U1 = SP882231/06
U2 = MC10E131FN
U3 = MC7905ECT
U5 = MC1458U
Q1 = 2N6191 or 2N3906 (PNP)

= 400ps delay (2.775°)
= 200ps delay (0.387°)

= PHASE Tracker (SEMFLX 30136-01)
  98ps ± 16ps)
= 50 OHM SMA FINX
  (Sealec850-631-0000-31)
**OUTPUT WAVEFORMS with Fo = 250MHz.**

- Fo/2 (CLIO, ADC 01)
- Fo/2- (CLIO, ADC 02)
- Fo/4CD (TRIO, ADC 01)
- Fo/4CD (TRIO, ADC 02)

**OUTPUT WAVEFORMS with Fo = 1GHz.**

- Fo/2 (CLIO, ADC 01)
- Fo/2- (CLIO, ADC 02)
- Fo/4CD (TRIO, ADC 01)
- Fo/4CD (TRIO, ADC 02)

**SET-UP TIME:**

A ≥ 725 ps

**HOLD-TIME:**

B ≥ 1.275 ns

**PULSE-WIDTH**

C = 1 ns PW @ 1GHz

**NOTE:** The Waveforms drawn for Fo/2 & Fo/2- are the Outputs from the PHASE TRIMMERS.
SIERRA MONOLITHIC'S EXPECTED FUNCTIONAL & PERFORMANCE OPERATION FOR
THE MODULE A CLOCK GENERATOR.

Module A will take an unbalanced ECL Input signal (\(F_0\)) ranging in frequency from 250Mhz to 1Ghz, then produce as outputs, FOUR balanced ECL compatible signals, namely \(F_0/2\), \(F_0/2^-\), \(F_0/4\) (I), & \(F_0/4\) (Q).

\(F_0/4\) (I) shall be in phase with the POSITIVE edge of \(F_0/2\), & \(F_0/4\) (Q) shall be in phase with the NEGATIVE edge of \(F_0/2\). Hence, \(F_0/4\) (Q) is always 90 degrees out of phase with respect to \(F_0/4\) (I).

Over the \(F_0\) INPUT frequency range, \(F_0/4\) (I) output changes logic states at least 725ps before and at least 725ps after the negative clock edge of \(F_0/2\). Likewise, it is true for \(F_0/4\) (Q) relative to the negative clock edge of \(F_0/2^-\).

ELECTRICAL CHARACTERISTICS:

POWER CONSUMPTION:
2.1 Watts (approx).

INPUT SIGNAL PORT (\(F_0\)):
Input Sensitivity: 100mV (min) Single-Ended.
Input Impedance: 50 ohms at 1Ghz - with matching.
20 - j50 at 1GHz - w/o matching.
100 - j175 at 250MHz - w/o matching.

OUTPUT SIGNAL LEVELS: (\(F_0/2\), \(F_0/2^-\), \(F_0/4\) (I), \(F_0/4\) (Q))

<table>
<thead>
<tr>
<th></th>
<th>0C</th>
<th>25C</th>
<th>75C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output HIGH Volt</td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td>-1.02</td>
<td>-0.84</td>
<td>-0.98</td>
</tr>
<tr>
<td>Input HIGH Volt</td>
<td>-1.17</td>
<td>-0.84</td>
<td>-1.13</td>
</tr>
<tr>
<td>Output LOW Volt</td>
<td>-1.95</td>
<td>-1.63</td>
<td>-1.95</td>
</tr>
<tr>
<td>Input LOW Volt</td>
<td>-1.95</td>
<td>-1.48</td>
<td>-1.95</td>
</tr>
<tr>
<td>Output Rise/Fall time (20-80%)</td>
<td>300ps</td>
<td>800ps</td>
<td>300ps</td>
</tr>
<tr>
<td>SKEW (U3 outputs)</td>
<td>100ps</td>
<td>100ps</td>
<td>100ps</td>
</tr>
</tbody>
</table>

NOTE: Motorola states that 10E Series type IC's do not have a problem driving 100E or 100K chips.
## Parts List - Amerasia: Clock Generator - Module A.

<table>
<thead>
<tr>
<th>REF</th>
<th>PART NO.</th>
<th>DISTRIBUTOR</th>
<th>SHIP (WEEKS)</th>
<th>QTY</th>
<th>PRICE (EACH)</th>
<th>PRICE (TOTAL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>SP8822B1/DG</td>
<td>Select Electronics</td>
<td>14</td>
<td>1</td>
<td>$49.76</td>
<td>$49.76</td>
</tr>
<tr>
<td></td>
<td>(Plessey)</td>
<td>(Arlene: 714 739-8891)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sample requested (11/8/89)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U2</td>
<td>MC10E111FN</td>
<td>WYLE</td>
<td>12</td>
<td>1</td>
<td>$17.31</td>
<td>$17.31</td>
</tr>
<tr>
<td></td>
<td>(Motorolla)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3</td>
<td>MC10E111FN</td>
<td>HAMILTON AVNET</td>
<td>STOCK</td>
<td>1</td>
<td>$28.23</td>
<td>$28.23</td>
</tr>
<tr>
<td></td>
<td>(Motorolla)</td>
<td>(213) 217-6830</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U4</td>
<td>MC7905.2CT</td>
<td>Hamilton Avnet</td>
<td>12</td>
<td>1</td>
<td>$ .54</td>
<td>$ .54</td>
</tr>
<tr>
<td></td>
<td>(Motorolla)</td>
<td>(213) 217-6830 call local distr...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U5</td>
<td>MC1458U</td>
<td>Hamilton Avnet</td>
<td>STOCK</td>
<td>1</td>
<td>$ .96</td>
<td>$ .96</td>
</tr>
<tr>
<td></td>
<td>(Motorolla)</td>
<td>(213) 217-6830</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G1</td>
<td>2N619L (PNP)</td>
<td>Local Distr...</td>
<td>STOCK</td>
<td>1</td>
<td>$ 1.00</td>
<td>$ 1.00</td>
</tr>
<tr>
<td>D1</td>
<td>1N914</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>30136-01</td>
<td>Cain Technology</td>
<td>?</td>
<td>4</td>
<td>$101.00</td>
<td>$404.00</td>
</tr>
<tr>
<td></td>
<td>(Semflex)</td>
<td>(213) 326-5236 (plus $200 setup charge?)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMA</td>
<td>50-651-0000-31</td>
<td>Selectro Corp</td>
<td>?</td>
<td>9</td>
<td>$ 10.00</td>
<td>$ 90.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(914) 698-5600</td>
<td>Sealectro West</td>
<td>(213) 990-8131</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total Estimated Cost:** $542.04
AMERASIA: CLOCK GENERATOR - MODULE B.
PRELIMINARY DESIGN.

PCB = G11, 15mil THICKNESS
50 OHMS = 5mil. LINE WIDTH.
100 OHMS = 5mil. LINE WIDTH.
ALL SIGNAL LINES ARE 50 OHMS
EXCEPT AS SPECIFIED.

NOTE: Decouple each IC
with a 0.01uf capacitor.

Total Power Consumption:
2.1 Watts for the -12VDC Power Supply,
plus an additional 2 Watts for Osc (-15VDC)

Fo = 1030 MHz (+10dbm)

OSC = 30-1030-400-L 0 SGV-ANDERSON LABD
PS1 & PS2 = TRANSMISSION LINE POWER SPLITTERS.
U1 = SP9022BG/DG
U2 = MC10E127FN
U4 = MC7905.2CT
U5 = MC1458U
Q1 = 2N6191 or 2N3906 (PNP)

= 400ps delay (2.773°)
= 200ps delay (1.387°)
Φ = PHASE Tracker (SEMFLLEX 30136-01)
(1920ps ± 150ps)
= 50 OHM SMA CNS
(Celesco 38-651-0000-31)

Sierra Monolithics
**Carrier Frequency (fo)**
1030 MHz

**Frequency Deviation**
\(0.04 \times (T - T_{ref})^2 \text{ ppm max., } T_{ref} = 25°C\)

**Output Power**
\(+10 \text{dBm} \pm 2 \text{dBm (1V pk) into 50Ω}\)

**Operating Temperature Range**
\(-45°C \text{ to } +85°C\)

**Tuning Range (Δf)**
400 KHz minimum

**Tuning Voltage (Δv)**
0 - 12 V

**Spurious - Harmonic**
30dBc

**- Non-harmonic**
-60dBc

**Phase Noise**
-65dBc/Hz @ 100 Hz

**-85dBc/Hz @ 1 KHz**

**-105dBc/Hz @ 10 KHz**

**Power Supply**
+15 V DC ± 5%, 125 mA

*Note(s): 1) Tuning range supplied is sufficient to maintain the specified carrier frequency over the effects of temperature and load pulling.*
Timebase: 500 ns/div, Delay/Pos: -100.000 ns, Reference: Left, Mode: Realtime (NORMAL)

<table>
<thead>
<tr>
<th>Channel</th>
<th>Sensitivity (V/div)</th>
<th>Offset (V)</th>
<th>Probe Ratio</th>
<th>Coupling Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel 1</td>
<td>2.00</td>
<td>0.00000</td>
<td>10.00 : 1</td>
<td>dc (1M ohm)</td>
</tr>
<tr>
<td>Channel 2</td>
<td>1.00</td>
<td>-1.00000</td>
<td>10.00 : 1</td>
<td>dc (1M ohm)</td>
</tr>
</tbody>
</table>

Trigger mode: Edge
On Negative Edge Of Chani
Trigger Level: Chani = 2.00000 V (noise reject OFF)
Holdoff = 40.000 ns

FIGURE A3-1, Input Signal
FIGURE A3-2, Output Signal (Readback)

<table>
<thead>
<tr>
<th>Main</th>
<th>Timebase</th>
<th>Delay/Pos</th>
<th>Reference</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.00 ms/div</td>
<td>-100.000 us</td>
<td>Left</td>
<td>Realtime (NORMAL)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Channel 1</th>
<th>Sensitivity</th>
<th>Offset</th>
<th>Probe</th>
<th>Coupling</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00 V/div</td>
<td>0.00000 V</td>
<td>10.00 : 1</td>
<td>dc (1M ohm)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Channel 2</th>
<th>Sensitivity</th>
<th>Offset</th>
<th>Probe</th>
<th>Coupling</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00 V/div</td>
<td>3.00000 V</td>
<td>10.00 : 1</td>
<td>dc (1M ohm)</td>
<td></td>
</tr>
</tbody>
</table>

Trigger mode: Edge
On Positive Edge Of Chani
Trigger Level
Chani = 2.00000 V (noise reject OFF)
Holdoff = 40.000 ns
FIGURE A3-3, Output Signal Observed with Faster Timebase
FIGURE A3-5, 2 MHz Triangular Input Signal
FIGURE A3-7, 1 MHz Sinewave Input Signal
FIGURE A3-9, Output Signal with 20 MHz Sinewave Input
TRANSFER FUNCTION
Averaged over all channels and cells

TRANSFER FUNCTION
Averaged over all cells, per channel
250.3 MHz RF signal
uncorrected

250.3 MHz RF signal
corrected with \( Y = M \times X + B \) per cell

Channel 1  Channel 2  Channel 3  Channel 4

Channel 1  Channel 2  Channel 3  Channel 4
250.3 Mhz RF

Four channels superimposed to show phase difference

Same graph expanded horizontally

163 166 173 180 cell number
0 3 10 17 Relative cell
0 47 156 265 ps delay
1.0003 GHz RF signal
All four channels superimposed to show phase
150 Khz Sine wave (+/- .3 v pk)
Channel 1 only

150 Khz Sine wave (+/- .3v pk)
All four channels superimposed to show phase
500 Khz Sine wave
Channel 1 only

1 Mhz Sine wave
Channel 1 only
10 MHz Sine wave

20 MHz Sine wave
1 MHz RF signal (+/- .3v peak)

10 MHz RF signal
50 Mhz RF signal
(aliased)

62.8 Mhz RF signal
(aliased)
125.3 MHz RF signal
(aliened)
Figure 3. Measured end-to-end group delay of microstrip delay line.

Loaded with 16 each 3.3 pF capacitors.

DELAY, 3 ns/div
PRECEDING PAGE BLANK NO: FILM:W
Figure 4. Measured end-to-end insertion loss of microstrip delay line.
ED80 - Touchstone - 02/11/94 - 09:28:45 - DEVLINES

Line loss is each 3 dB down from continuous line.

Figure 5a. Predicted end-to-end Delay and Insertion loss of Microstrip Delay Line.
PICTURE 5

<table>
<thead>
<tr>
<th>Channel 1</th>
<th>Channel 2</th>
<th>Probe</th>
<th>Offset</th>
<th>Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>dc (1MΩ)</td>
<td>dc (2MΩ)</td>
<td>50.00 V</td>
<td>0.00 V</td>
<td>2.00 V/dIV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.000000</td>
<td>0.000000</td>
<td>2.00 V/dIV</td>
</tr>
</tbody>
</table>

HOLDoff = 40.0000 ns
Channel = 2.000000 V (note: Reference On negative edge of channel
Trigger mode: Edge
Trigger level: 2.000000 V

Timebase = 500 ns/div
Reference = 2.000000 ns
Delay/Pos = 0.000000 ns

NORMAL (Reset time)
### Figure 10

- **HOLDoff = 4.000 ms**
- **CH1/2 = 20.000 V (Note: Respect Off)**
- **Trigger Level**
  - **Trigger mode**: Edge
  - **On positive edge of CH1**

<table>
<thead>
<tr>
<th>Channel</th>
<th>DC (1M ohm)</th>
<th>Probe</th>
<th>Reference</th>
<th>Offset</th>
<th>Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH1</td>
<td>3.00 V/Div</td>
<td>Probe</td>
<td>Reference</td>
<td>Offset</td>
<td>3.00 V/Div</td>
</tr>
<tr>
<td>CH2</td>
<td>0.000 V/Div</td>
<td>Probe</td>
<td>Reference</td>
<td>Offset</td>
<td>0.000 V/Div</td>
</tr>
</tbody>
</table>

- **Main Timebase**: 10.00 ms/div
- **Reference Timebase**: 0.100,000 us
- **Delay/Pos**: -100.000 ms

- **Horizontal Scale**: 0.0000 ms
- **Vertical Scale**: 4.80000 V
- **Horizontal Scale**: 0.000000 ms
- **Vertical Scale**: 0.000000 V

---

The figure shows a waveform with oscillations, indicating a pattern of electrical signals. The waveform is displayed on a grid with time and voltage scales, allowing for precise analysis of the signal characteristics.
Figure 11

Holdoff = 40.000 ns
Channel 1 = 0.0000 V (Note: Ref. Off)
Trigger Level:
On positive edge of Channel 2
Trigger mode: Edge

Channel 2
3.00 V/div
5.00 V/div
0.00000 V
0.00000 V
Probe Offset
Sensitivity

Channel 1
3.00 V/div
5.00 V/div
0.00000 V
0.00000 V
Probe Offset
Sensitivity

Mode
Reference Delay/Pos
Realtime
Normal

Timebase
-1.00 ns/div
9.0000 ns
4.00000 ns
100.000 ns
APPENDIX 5

HARDWARE INTERFACE

1. 16 bit bi-directional parallel data

2. Four wire handshake
   a. DMAWREQ - negative pulse by host initiates transfer
   b. DMAWACK - 0 after DMAREQ, 1 when accepted by \( \mu P \).
   c. DMARAV - 0 after \( \mu P \) loads latch, 1 after read by host
   d. DMARREQ - negative pulse by host reads data and clears status

3. Write transaction (Host to \( \mu P \))
   a. data is latched on trailing ( + ) edge of DMAWREQ
   b. low level on DMAWREQ sets WR status FF (DMAWACK = 0)
   c. \( \mu P \) read clears WR status FF (DMAWACK = 1)

4. Read transaction (\( \mu P \) to Host)
   a. Generally occurs as a result of request from host
   b. Only exception is error condition status
   c. \( \mu P \) writes data to latch
   d. this sets read status FF (DMARACK = 0)
   e. DMARREQ low places data on Q bus
   f. DMARREQ low level resets read status FF (DMARACK = 1)

5. commands are 16 bit data
   a. 12 lsb is value
   b. 4 msb is command
      0 = code address low - first word of Read, Write or Execute in code memory. (with low 12 bits of address)
      1 = data address low - first word of Read or Write in data memory. (with low 12 bits of address)
      2 = start address high - second word of Read or Write range (upper 8 bits of code address or 4 bits of
code address right justified in lower 12 bits)

3 = start address high - second (final) word to read one word from memory. Next transaction is a single read transaction.

4 = start address high - second word to write one word to memory. Next transaction is a single write transaction with a 16 bit data word.

5 = execution address high - second (final) word to start execution at address (upper 8 bits of address right justified in lower 12 bits)

6 = end address high - fourth (final) word to read an inclusive range of memory. This is preceded by a second low address (0 or 1). This is followed by N + 1 read transactions.

7 = end address high - fourth word to write an inclusive range of memory. This is preceded by a second low address (0 or 1). This is followed by N + 1 write transactions.

8 = I/O read - 12 lower bits are I/O address. Next transaction is a single read transaction with data word read (no prefix command needed).

9 = I/O write - 12 lower bits are I/O address. Next transaction is a single write transaction with data word to be output (no prefix command needed).

A = skip value - 12 lower bits contain number of cells to skip at the beginning of an acquire. This should be a multiple of 16.

B = acquire value - 12 lower bits contain number of cells to read and convert. This command also does a software arm. This should be followed by a hardware arm and trigger. The hardware will respond with a sync strobe, and the μP will initiate N read cycles with data.

C = auto acquire (optional) Number of cells to skip and
read determined by the \( \mu P \) by inspecting the data. 
12 lower bits are ignored. Otherwise works like command 9.

\( D = \) calibrate. First call set mode, remaining calls give value of last acquire. Final call with value of 0FFh calculates coefficients.

\( E - F \) reserved
a. These go to jump vectors in code RAM
b. Initially they are programmed as NOP's

6. Timing
a. When host writes data to system, data is latched on trailing edge of strobe. Status is level sensitive, so responds to leading edge. In order to prevent multiple reads by system, pulse width must be less than 500 ns.
b. When host reads data, both data enable and status are level sensitive. The strobe pulse width must be less than 500 ns in order to prevent \( \mu P \) from placing second word on bus before end of pulse.
c. Read data is valid from 30 ns after DMARREQ goes low until a small time after DMARREQ goes high (20 ns typ.)
d. Write data must be valid at least 30 ns before the rising edge of DMAWREQ and remain valid at least 5 ns after the rising edge of DMAWREQ.
e. The Status output bit (Data/Status, 0/1) is valid from the time DMARAV goes low until it goes high again.
f. The Command input bit (Data/Command, 0/1) must be valid from the falling edge of DMAWREQ until the subsequent rising edge of DMAWACK. This is best accomplished by providing an non-Tristate ff whose state is changed just before data is written.
g. The command bit is interrogated during a reset (either power up reset, or external reset). If it is a 1, a cold reset is forced (all variables initialized to default,
calibration values all set to gain = 1 and offset = 0, all
code extensions eliminated and a new code memory checksum
computed. If Command = 0 a warm start is done instead,
provided that the code checksum agrees with the value stored
in the data memory.

7. Error and status codes (Status = 1)
   0. OK
   1. Warm start (after reset, unsolicited)
   2. Cold start (after reset, unsolicited)
   3. (reserved)
   4. Completed (generally at end of block reads)
   5. Illegal data - data written when not expected
   6. No data - Command written when more data was expected,
       or during operation.
   7. Write interrupt (not used)
   8. Aborted - Command received after Acquire command,
       before arm signal received.
   9. Not Triggered - A/D acquire cycle did not start,
       probably due to lack of a trigger.
  0A. No Acquire - A/D acquire cycle did not complete.
  0B. In Calibration - A/D cycle completed. No data will
      be sent because data is being used for calibration.

NOTE: attempt has been made to keep error codes in
the range of 0 - 0Fh so that they can be loaded
with a single word mov instruction.
This is the final report for the research and development of the 1 GHz Digitizer for Space Based Laser Altimeter. A Feasability model was designed, built, and tested. Only partial testing of essential functions of the Digitizer was completed, due to limited funding available at this time. Hybrid technology was incorporated which allows analog storage (memory) of the digitally sampled data. The actual sampling rate is 62.5 MHz, but executed in 16 parallel channels, to provide an effective sampling rate of 1 GHz. The average power consumption of the 1 GHz Digitizer is not more than 1.5 Watts. A 1 GHz SAW oscillator is incorporated for timing purposes. This signal is also made available externally for system timing. A software package was also developed for internal use (controls, commands, etc. and for data communications with the host computer. The Digitizer is equipped with an on-board microprocessor for this purpose.