An Interrogable Data Collection Node (DCN):
A Microprocessor Based System for Collection and Transmission of
Low Sample Rate Geophysical Data

By
Reese Cutler*
M.J.S. Johnston

Open-File Report 1981-762

This report is preliminary and has not been reviewed for conformity
with Geological Survey editorial standards and nomenclature.

*Cutler Digital Design, 2215 Sun Mor, Mountain View, California
Abstract

A microprocessor-based digital telemetry system for the acquisition of low-frequency geophysical data from remote field sites has been designed and tested. The field unit, referred to hereafter as a data collection node (DCN), is built in modular form for easy selection of operating functions, expansion of solid-state memory and re-configuration for different field requirements. A general purpose version of this DCN that responds to manual or computer generated telephone calls has collected and transmitted deformation data for several months at a field site near the San Andreas fault. Within the digitization errors these data faithfully reproduce those obtained with the standard U.S.G.S. digital telemetry system. Average power dissipation for this application was less than 500 mw depending on the number of channels monitored.
Introduction

Data collection systems of many kinds have been designed for the collection, storage, and transmission of both high and low sample rate data. Unusual problems are encountered in the field of earthquake research with the particular requirements for large-scale geophysical data collection. Field conditions can be extreme and maintenance almost non-existent. Few sites selected for geophysical reasons have power or shelter, yet, in order to monitor the slow deformations of the earth's crust accurate, stable and reliable data collection is required over periods of many years. The requirements for high reliability and stability with low power consumption and the need for immediate access to data from widely dispersed arrays of instruments has spawned the development of several low frequency digital telemetry systems (Roger et al., 1977; Cutler and Johnston, 1978, Nickerson, 1979). These systems allow collection, transmission and reception of data to a common data center.

The most recent and powerful development concerns the introduction of microprocessor based data collectors and receivers. The use of microprocessors permits these systems to replicate the operation of the various existing fixed-function telemetry systems. The important difference is that their functions are software rather than hardware controlled. This allows easy alteration of system parameters and control of both the mode of operation and data collection from a common data collection center.
This note describes such a data collection system, here termed a data collection node or DCN, whose particular characteristic for low frequency data collection are:

1) low power consumption
2) solid state memory for high reliability
3) modular hardware design for easy adaption to a new collection requirement and easy field repair (e.g., an increase in memory in increments of 2048 sixteen bit data can be accomplished by plugging extra memory modules into the communication bus.
4) input ports for up to 15 channels of analog data and one channel of 16 bit parallel digital data.
5) compatibility with telephone and radio telemetry for one or two-way communication and data flow to a base operator or computer.

Figure 1 shows a block diagram of this system. Although hardware meeting the functional requirements listed here has existed for some time, only recently have dramatic reductions in power consumption, size, complexity, and cost of microprocessors made possible the widespread use of versatile telemetry with geophysical field instrumentation.
Figure 1: Block diagram showing layout and design requirements of a data collection node (D.C.N.) for geophysical data collection.
DCN Design Considerations

The fundamental units of the D.C.N. are displayed in Figure 2. Modularity is accomplished with the use of a common communication bus. All functional parts are grouped on modules that plug into the bus and have access to all control and data information. Because of the low power requirements of available components, the power supply module contains 5V rechargable batteries with a 2 amp-hr capacity which will drive the device, in normal operation, for well over a year. The computer module presently used is a COSMAC CDP1802. Control programs are stored in non-volatile read-only-memory (R.O.M.'s) in the program module while random-access memory (R.A.M.'s) provides expandable solid state data (typically in units of 4096 8-bit data) storage in the data storage modules. Other necessary circuitry for bus termination, internal clock control, etc. is included in a separate control module.

Optional modules that can be added in order to tailor the system to a particular field application include:

1) an analog input module which contains a multiplexer to expand the number of analog input channels and an analog-to-digital converter.
2) a digital input module to allow the acceptance of digital field data.
3) a switched power supply module which provides current to attached devices only when it is required in order to reduce the total power dissipated.
Figure 2: Block diagram showing the modular units of a D.C.N.
4) a digital output control module which controls an input/output communication link and if necessary, controls the operation of the field instrumentation.

These options, or others that might be needed, can be simply added at any of the many parallel inputs to the communication bus. One example might be the addition of more on-site solid-state memory to provide insurance against the loss of the communication link for an extended period of time. Another example is the addition of specialized programs in a ROM module to do real-time data analysis or testing.

In summary, customizing a D.C.N. for a particular field application requires:

1) Definition of the functions required. For example, on receipt of a computer-controlled phone call, the D.C.N. might be required to switch on a number of instruments, take a series of measurements, check these measurements for previously identified anomalous behavior, determine a mean value and transmit this value back to the base computer.

2) Selection of the appropriate set of modules to provide these functions. Note that the program module may require the addition of a special purpose program to several of the general purpose data collection programs. In fact, each particular application will probably have a particular program module.
Application of the DCN to Tilt and Strain Data Collection

Following prototype development of this system (Roger et al., 1978) a version of the DCN configured for either manual or automatic call-up and data transfer was built and operated for several months collecting tilt data at a site near the San Andreas fault. The tiltmeter is a liquid level mercury type. It's two orthogonal components are 5 m long and are orientated in directions N 73° E and N 17° W respectively. The tiltmeter location is 37.79° N, 122.235° W. This instrument is presently monitored with standard U.S.G.S. low frequency digital telemetry (Rogers et al., 1976) so that it was possible to check the operation of the DCN by comparing data obtained with both systems after transmission to Menlo Park, California.

The block diagram of this DCN telemetry system including the particular module options chosen is shown in Figure 3. The circuitry is described in detail in Appendix A. The computer module uses a standard general purpose analog data collection program which is described in detail in Appendix B and listed in Appendix C. Besides sampling and storing data from up to 16 analog inputs, this program produces averages of the data and stores these data averages in memory. When called by telephone the program controls transmission of a fixed amount of data. In this case parameters are set in a look-up table to allow selection of:

1) the number of analog channels to be sampled.
2) the number of seconds per sample period.
3) the number of sample periods over which the averages is performed.
Figure 3: Block diagram of D.C.N. telemetry system and module options chosen for telemetry test at Presidio tiltmeter site.
Figure 4: Detailed 3-day comparison between the U.S.G.S. digital telemetry system and the D.C.N. telemetry system. Samples were taken every 10 minutes with both systems. The noise on the data is least-significant-bit-error. The traces are offset by 0.01 uradian to allow comparison.
Figure 5: Two month comparison between hour averages from the U.S.G.S. digital telemetry system those from the D.C.N. telemetry system.
4) the number of raw data samples to be transmitted.

5) the number of averages to be transmitted.

Figure 4 shows three days of individual 10-minute data samples from the tilt component PDOE as recorded by the standard U.S.G.S. low frequency telemetry (lower plot) and the DCN telemetry system (upper plot). The records are offset by 0.01 \( \mu \) radians to facilitate comparison. The periodic character of the data is of course due to the earth tides.

Figure 5 shows a longer term comparison between the two systems. Hour averages computed by the DCN from the raw data and transmitted to Menlo Park are shown in the upper plot. The lower plot shows hour averages computed from the data individually transmitted to Menlo Park with the standard U.S.G.S. telemetry.
References


Figure Captions

Figure 1: Block diagram showing layout and design requirements of a data collection node (D.C.N.) for geophysical data collection.

Figure 2: Block diagram showing the modular units of a D.C.N.

Figure 3: Block diagram of D.C.N. telemetry system and module options chosen for telemetry test at Presidio tiltmeter site.

Figure 4: Detailed 3-day comparison between the U.S.G.S. digital telemetry system and the D.C.N. telemetry system. Samples were taken every 10 minutes with both systems.

Figure 5: Two month comparison between hour averages from the U.S.G.S. digital telemetry system those from the D.C.N. telemetry system.
APPENDIX A
CIRCUIT DETAILS

The DCN Bus (refer to logic drawing 1)

Communication between all modules of the DCN is over a set of 72 electrical connections which are called, collectively, the DCN communication bus. Each of the 72 connections is made available to each module. For the purpose of this description, the 72 lines can be divided into the following categories: power and ground, normal data transfer, interrupt and external flags, bus control and direct-memory-access, service control, and special microprocessor lines.

Two lines are reserved for the nominal +5V supply, and a corresponding 2 lines are reserved for grounds. Six lines are reserved for the switched power supplies: two for +15VSW, two for -15VSW, and two for +5VSW. These six lines are only valid if there is a switched power supply module, and if it has been turned on.

At any time in the normal operation of the DCN one module is designated as the bus master. This module is responsible for setting up the correct bus control signals for data transfers. At present, the only module capable of bus mastership is the computer module.

Normal data transfers proceed as follows. Read operation: BMRDL is asserted-low, and the proper address lines are asserted (BAD15 through BAD0). Next, BTPB is asserted-high. For the duration of BTPB the addressed location should assert its contents onto the data lines BD7 through BD0. When BTPB goes low the read operation ends. Write operation: BMWRL is asserted-low, the proper address lines are asserted. At the same time that BTPB is
asserted-high, the bus master asserts the date lines BD7 through BD0 to the value that is to be written. Half way through BTPB, BMWRL is de-asserted. Normally this change of state is used to load the value on the data lines into the addressed location. The write operation ends with the de-assertion of BTPB.

To simplify the decoding for input and output devices, the top 256 bytes of the address space are reserved for I/O. There is a bus signal that indicates addressing of this block of memory (BDEVL). Input and output devices use this signal to simplify their address decoding task. BDEVL is asserted-low, asynchronously, whenever the eight high order address lines are high.

There are four external flags associated with the CDP1802 microprocessor: EF1L, EF2L, EF3L, and EF4L. They are normally high and are active-low. When more than one device is allowed to drive these lines, the devices in question should not actively hold the flag line in question is a de-asserted state (high)—there is a resistor pull-up to do this. For an example of proper interface to the external flags see the analog input module description.

There is one interrupt line (INTL which is active-low) that can be used by any module to initiate an interrupt. As with the external flags, modules should not actively hold the interrupt line in a de-asserted state; there is a resistor pull-up to do this.

To simplify interrupt circuitry on each module there are special control lines for servicing interrupts. The top byte of memory is designated as the interrupt buffer. Writing into this location sets interrupt enable flags. Reading from this location gives access to the state of interrupts, and then
clear the interrupt flags that are on. INTREAD directs interrupt circuitry to
assert the interrupt state on the appropriate bits of the data lines.
Whenever INTLDL makes the transition from being asserted to being de-asserted
(rising edge), interrupt circuitry loads the value of the appropriate data
line into an interrupt enable buffer. Finally INTCLR asynchronously clears
interrupts, after a system reset, and after interrupt status has been read.

Bus mastership is requested through the use of the daisy-chained signals
BREQIN and BREQOUT, and granted through the use of the daisy-chained signals
BGIN and BGOUT. The computer module is the default bus master, and must
reside to the right of all devices requesting bus control, with all
intermediate sockets in use. Other devices may request control, and will be
granted it with highest priority to the devices physically closest to the
computer module.

DMAI, DMAO, and direct memory access control lines specific to the
CDP1802 microprocessor and are available on the bus. For more information see
the DCP1802 documentation and refer to the computer module drawings.

"Service control" is a grab bag term to describe the lines that perform
various and special bus functions. There is a one megahertz clock (MHZ), from
which the whole system timing is derived. There is a line which enables or
disables the computer module clock: CLKENL (enables the normal clock when
active-low). There is a clock line (XCLOCKL) that can be used to replace the
normal computer module clock whenever CLKENL is not asserted. There is a
general reset request line, RSTRQL, which causes BRESET and BRESETL to be
asserted.
Finally there are seven DCP1802 control lines that are also made available. BSC1 and BSC0 reflect the state of the microprocessor. BN2, BN1 and BN0 reflect the state of an 1802 input/output instruction. BTPA is an 1802 timing pulse, and BQ is the output of an 1802 flip-flop. See the CDP1802 hardware literature, and the computer module description for more information.

BUS TERMINATION AND CLOCK MODULE (refer to logic drawing 2)

This module does many DCN communication bus housekeeping chores: resistor pull-up or pull-down of bus lines, reset circuitry, I/O device decode, interrupt servicing control, one megahertz system clock, and a one second clock with external flag, and interrupt capability.

In general bus lines are either tied to +5V through a 100 Kohm resistor or tied to ground through a 100 Kohm resistor. This is to avoid the possibility that certain lines will be floating, and cause an increase in power consumption, or an illegal logic condition.

A reset request (RSTRQL active-low) shorts the input of a schmidt trigger (loc. C-1) to ground, thereby driving BRESET and BRESETL active (high and low respectively). The RC network in front of the schmidt trigger is included to allow switch bounce on RSTRQUL, and to insure that the reset lines are active long enough. This module includes a switch that asserts RSTRQL, so that the DCN can be reset manually.

Input/output decoding is simplified with the generation of BDEVL (loc. B-15), that goes active-low whenever BAD15 through BAD8 are all active-high.

Interrupt service lines INTREAD and INTLDL are only asserted when all address lines are active-high (loc. D, E-9, 12). INTREAD, also requires that
BMRDL be active-low, while INTLDL requires that BMWRL is asserted-low. INTCLR is asserted on the falling edge of INTREAD, and cleared when the first non input/output address is put on the address lines.

There is a one megahertz clock, MHZ (loc. H, I, J-5, 6), which goes on the bus, and which drives a divide by 1,000,000 chain to produce a one second clock. The "one second" flip-flop (loc. G-13) is set once a second. If it is on, it holds EF3L to ground (by opening an analog gate, 1/4 IC5), and if the interrupt enable flip-flop is also on, it holds INTL to ground (by opening two analog gates in series, 1/2 IC5).

The bus request and bus grant daisy-chain is jumpered to allow other devices to use it.

**COMPUTER MODULE** (refer to logic drawing 3)

This module interfaces the RCA CDP1802 microprocessor to the DCN bus, and provides bus mastership control.

A BRESETL asserted-low forces the processor into the reset state. The following lines go directly from the bus into the appropriate CDP1802 inputs: INTL, DMAIL, DMAOL, EF1L, EF2L, EF3L and EF4L (loc. E,F.G-7).

If CLKENL is asserted-low and OXLOCKL (active-low) is high, then the microprocessor clock is generated by the selected output of the divider (IC12, loc. H,I-4) which is driven by the one megahertz system clock MHZ.

If the bus is controlled by this module (hence, there are no bus requests) all the following CDP1802 signals are asserted onto the DCN bus through tri-state buffers (IC1, 2, 3): BTPA, BTPB, BSC1, BSC0, BRDL, BMWRL, BN2, BN1, BN0, and BAD7 through BAD6. The high order address lines need to be latched,
and are also only asserted if no bus request has been granted (uses a tri-state register, IC7, 8). The value of the address lines BAD15 through BAD8 is present on the CDP 1802 address lines during TPA and is loaded into the tri-state register on the falling edge of TPA. An eight bit tranceiver is used (IC4, 5) to allow two way data bus transmission between the microprocessor data bus, D7 through D0, and the bus data lines BD7 through BD0.

A bus request, BREQIN, active-high causes the bus grant flip flop (loc. A-4, 5) to turn on at the next falling edge of the CDP1802 control signal TPB. The BG is daisy-chained across the bus to the requesting device, and the microprocessor buffers onto the bus are all turned off. The bus grant signal also forces the microprocessor into a wait state. As soon as the bus request input is de-asserted, the bus grant flip-flop is cleared, and the microprocessor regains control of the bus.

PROGRAM MODULE (refer to logic drawing 4)

The read only memory capacity of this module is 2048 bytes of contiguous storage. The module uses four 512 by 8 bit read only memory packages (IC12, 13, 14, 15). The address lines to these ROM arrays is generated by buffering BAD8 through BADφ. Select lines (ROMEN3L through ROMENφL) are generated with the use of four 2 to 4 decorders (IC1, 2). Jumper selection on the module allows for the placement of the read only memory in one of 32 2K address space slots. Notice that if the module is placed in the highest address slot, it does not respond if the very highest 256 bytes are addressed (BDEVL inhibits the decoder chain in this event). Notice also that this module responds only when BMRDL is asserted-low; and the date is placed from the module onto BD7
through BDØ only during BTPB.

The bus request and bus grant daisy-chain is jumpered to allow other devices to use it.

**STORAGE MODULE** (refer to logic drawings 5-7)

There are four functional sections of this 4096 by 8 bit random access memory module: address buffers, address decoder, data tranceiver, and memory array.

Since the address lines have to go to 32 different 1024 X 1 RAM packages (IC1-IC32, logic drawings 6, 7), the bus address lines BAD9 through BADØ are buffered (IC38, 39 logic drawing 5, loc. B, C., D-6, 7, 8). The high order address lines are decoded using 2 to 4 decoders (IC33, 34). The module is jumper locatable into one of sixteen 4K slots in the address space of the machine. The memories are enabled during BTPB, and they are inhibited if the 256 byte I/O buffer is addressed.

The data tranceivers (IC36, 37) allow the two way communication between the bus data lines BD7 through BDØ, and the local data lines MI7 through MIØ and MO7 through MOØ. The bus control line BMRDL determines the direction of data flow, and the local enable line RAMEN allows data flow when this module is selected.

The bus request and bus grand daisy-chain is jumpered to allow other devices to use it.

**DIGITAL INPUT MODULE** (refer to logic drawing 8)

Each digital input module allows the input of sixteen bits of digital information. The module is arranged to allow connection to the sixteen bits
with either screw down terminals, or with dual-in-line sockets. The module also allows access to the following bus signals through screw down terminals: BQ, EF1L, EF2L, EF3L, and EF4L.

There are two major functions on the module. A series of 2 to 4 decoders accomplishes the address decoding (by taking advantage of the I/O select line BDEVL - IC2, 3). The actual buffer for the digital inputs is made from a tri-state sixteen to eight multiplexer (IC4, 5).

The module has the option that the digital inputs can be tied up to +5V with 100 Kohm resistors. This allows the digital input to drive the actual inputs with an open collector transistor stage, and not worry about matching logic levels. A special case of this kind of use occurs when the optional resistors are used and dual-in-line switch arrays are inserted into the dual-in-line sockets, making the module a switch register.

Since the module allows input of 16 bits, but the data lines can only handle eight bits at a time, the full sixteen bits of data must be input in byte slices. This also means that each module uses two of the 255 I/O locations available to it. Note: the location of the module in the 255 byte I/O address space is jumper selectable.

The bus request and bus grant daisy-chain is jumpered to allow other devices to use it.

**DIGITAL OUTPUT MODULE** (refer to logic drawings 9, 10)

The digital output module is really an enlarged version of the digital input module. It allows the user to output sixteen bits with the feature that the user can read the value of the bits that have been output. It means that much of the circuitry is the same between digital input and output modules.
There is a sixteen to eight multiplexer (IC4, 5) that makes the output data available to the bus data lines. The address decoding uses a series of 2 to 4 decoders, which are enabled by BDEVL and produce write enable (IOWENL) and read enable (IORENL) control lines. IOWENL enables a write operation into the output register.

The output register is sixteen bits worth of data in four integrated circuits (ICb, 7, 8, 9). The outputs of these registers go to the input multiplexer, the dual-in-line sockets, an the screw down terminals. Since there are sixteen bits in the module and the data lines of the DCN bus can handle only eight bits, this module takes up two I/O locations.

There are two screw down terminals that signal the outside that a new eight bits has been loaded in the register. They are labelled H for the line corresponding to the high order byte, and L for the line corresponding to the low order byte. The lines H or L go active-high after the write operation is completed, and remain high until the module is deselected.

As with the digital input module BQ, EF4L, EF3L, EF2L, and EF1L are also accessible through screw down terminals.

The bus request and bus grant daisy-chain is jumpered to allow other devices to use it.

SWITCHED POWER SUPPLY MODULE (refer to logic drawings 9, 10)

The switched power supply is really just an optional part of the digital output module. It uses the least significant bit of the output port to turn on a network of transistors that provides +5VSW, a switched 5 volt supply to the DCN bus. The +5VSW is also directed into a DC to DC converter whose outputs drive the DCN bus supplies, +15VSW and -15VSW.
ANALOG INPUT MODULE (refer to logic drawing 11)

This part of the DCN can be divided into six sections: the analog multiplexer, the analog to digital converter, the buffer between the A/D output and BD7 through BD0, the address decoder, the channel select and control logic, and the interrupt logic.

The analog multiplexer (IC16) takes its sixteen inputs form sixteen screw down terminals, or from two dual-in-line sockets (not shown in the logic drawing). Since the multiplexer is only operational when the switched power supply is on, the channel select and enable control line (CH3B, CH2B, CH1B, CH0B and ENMUXB) are buffered and asserted only when the power supply is on.

The analog to digital converter is preceded with all low pass filter (IC15). The analog multiplexer output goes into the filter, and the filter output goes into the analog to digital converter. Since the analog to digital converter is also powered by the switched power supply, its control signal, TRIGB is also buffered and asserted only when power is on.

The digital outputs of the analog to digital converter are made available to the DCN through a sixteen to eight digital multiplexer (IC10, 11). The twelve lines from the converter must be tied through 100 Kohm resistors to ground in order to keep the multiplexer inputs from floating when the switched supply is off. The other four bits of the multiplexer are supplied by a channel register (IC8), that also selects which of the sixteen analog inputs is to be digitized.

The address decoder is enabled by an address in the I/O address range (where BDEVL goes active-low). An array of 2 to 4 decoders (IC3, 4) produce
the write enable (WENL) and read enable (RENL) control lines. This module's address is jumper selectable to any of the 127 two byte slots in the I/O address space. When the DCN attempts to write in either of the two I/O byte locations associated with this module, the value on the bus data lines BD3 through BD6 is loaded into the channel register. Then, if the switched power supply is on, a trigger signal (TRIG) is generated which starts the analog to digital conversion. Since this trigger is not wanted until after the channel register is loaded, an extra flip-flop is necessary (1/2 IC7).

Interrupt and flag circuitry includes three flip-flops and four analog gates. The first flip-flop (1/2 IC5) holds the status of the analog to digital conversion. Whenever this flip-flop makes the transition from not ready to ready, it sets the interrupt flip-flop (IC5, loc. G-12, 13). The RDY control line also enables an analog gate (IC1, loc. I-14) which asserts an active-low on the bus flag line EF3L. The interrupt is passed to the bus interrupt line, INTL, if the interrupt enable flip-flop (IC7 loc. H-13) is also set. This is accomplished by opening the two analog gates (IC1 loc. H-14, 15) to short INTL to ground. Finally, special bus interrupt control lines, INTREAD, INTCLR, and INTLDL allow the processor to simply read and clear the interrupt bit (in this case from BD6), and allow the processor to set the interrupt enable bit (again from BD6).

COMMUNICATION MODULE (refer to logic drawing 12)

Communication on the telephone is accomplished using a telephone data access arrangement (CBT or 1001D). It is a standard phone interface which
draws no power when not in use. This module interfaces with the DAA on one side and a digital output module on the other. Functionally there is a parallel to serial converter, a voltage controlled oscillator (VCO), a standby DAA power supply, a ring and off-hook detector, and an off-hook request driver.

When the control signal XMIT (which enters the module through a screw-down terminal) goes active-high, it turns on a trigger oscillator. This oscillator keeps requesting a transmission from a parallel to serial converter module (Larse Send). When the status output of this module indicates that the transmission has begun, the trigger oscillator turns off. Parallel input to the parallel to serial converter comes from a digital output module and connects through a set of dual-in-line sockets. The output of the converter goes into a VCO (see USGS open file report for more complete details of the oscillator, or see drawing 13). The outputs of the VCO ago directly to the DT and DR inputs of the data access arrangement. The VCO and the parallel to serial converter are powered by the DCN switched power supply.

The DAA standby power supply is provided to allow ring detection while the DCN switched power supply is no on. Three 7.8 volt NiCad batteries are used. One of the batteries has a center tap installed, so that when in series the three are configured as a +11.7V supply. A simplified battery configuration is shown in the drawing (EFGH-13, 14). This battery arrangement, through diodes D1 and D4, insures the DAA power supplies +V2 at +11V. If the switched powersupply is on, the NiCad batteries are recharged, and +V2 goes to about +13.6V.

If the DAA is receiving a telephone call, the RI output of the DAA will be switching from an open state to a -V2 state at a frequency of 40 hertz for two
seconds out of every six seconds. This signal is translated into CMOS levels with the resistor-transistor network (R8, R9, R10, Q2 loc. C-1, 3). The series of one-shots insures that the right signal is of adequate frequency and duration to be a ring and not a line transient. If so, the off-hook flip-flop (IC2 loc. D10) is set. This action causes another resistor-transistor network (ABC-13, 14, 15) to assert to -VI the two DAA control signals OH and DA, which opens the telephone communication and connects the DT and DR inputs to the telephone line. Finally, the DAA will respond with another control signal, CCT, which assures that the connection operation is complete. This signal is shifted to CMOS signal levels and gated within the output of the off-hook flip-flop to give the status signal TRDYL. When this line goes active-low, it means that the communication link has been established and transmission can begin.
Circuit Diagram of Quartz Crystal 1 Reference Digital VCO.
APPENDIX B

General Purpose Analog Data Collection Program

START

CALL SETUP SUBROUTINE

CALL SAMPLE SUBROUTINE

CALL TRANSMIT SUBROUTINE
Setup Subroutine

SETUP

TRANSFER PARAMETERS FROM
ROM STORAGE TO PARAMETER
TABLE IN RAM

ENABLE 1 SECOND INTERRUPT

PATCH9

INITIALIZE SAMPLE POINTER
AND AVERAGE POINTER

INITIALIZE TELEPHONE

SET INTERVAL FLAG

EXIT
Sample Subroutine

1. **Sample Patch**
2. **Sample Patch**
3. **Sample Patch**
4. **Sample Patch**
5. **Sample Patch**
6. **Sample Patch**
7. **Sample Patch**
8. **Sample Patch**
9. **Sample Patch**
10. **Sample Patch**

**Flowchart Directions:**
- **S1:** IS THE CURRENT INTERVAL DONE?
  - **S2:** YES
    - **S3:** SAVE REGISTERS R9, R10, R11, AND R12
    - **S4:** RELOAD SECOND COUNT REGISTER FROM PARAMETER TABLE AND CLEAR INTERVAL FLAG
    - **S5:** TURN THE POWER ON
    - **S6:** SET THE CHANNEL COUNT TO 0
      - **S7:** TURN OFF SECOND FLAG
      - **S8:** EXERCISE THE A/D
      - **S9:** WAIT FOR SECOND FLAG TO BE SET, THEN CLEAR IT
      - **S10:** PERFORM AN A/D CONVERSION ON THE CHANNEL IN CHANNEL COUNT
    - **S11:** SAVE DIGITIZED DATA USING THE SAMPLE POINTER
      - **S12:** UPDATE THE SAMPLE POINTER AND WRAP AROUND IF NEC.

**Variables:**
- **Channel Count = R9L**
- **Second Flag = R15L**
- **Sample Pointer = R13**
Sample Subroutine (cont'd.)

S6

S10
ADD DIGITIZED DATA TO THE SUM FOR THIS CHANNEL

S11
HAVE ALL CHANNELS BEEN SAMPLED?
YES

NO

S12
INCREMENT THE CHANNEL COUNT

S13
IF A TRANSMISSION IS NOT IN PROGRESS, TURN OFF THE POWER

S14
GET THE AVERAGE COUNTER (STORED IN THE PARAMETER BUFFER)

S15
DECREMENT AVERAGE COUNTER
AVERAGE COUNTER = 0?
YES

NO

S16
SAVE AVERAGE COUNTER

SEXIT
SAVE REGISTERS ROH, R9, R10, AND R11

EXIT

S17
RESET AVERAGE COUNT FROM PARAMETER TABLE
SET CHANNEL COUNT TO 0

S18
Sample Subroutine (cont'd.)

S18
TRANSLATE THE ACCUMULATED SUM FOR THIS CHANNEL INTO A FLOATING POINT FORMAT

S19
STORE THE FLOATING POINT NUMBER USING THE AVERAGE POINTER

UPDATE THE AVERAGE POINTER AND WRAP AROUND IF NEEDED.

S20
PATCH5

HAVE ALL AVERAGES BEEN STORED?

S21, NO
INCREMENT THE CHANNEL COUNT

P51
CLEAR THE AVERAGE BUFFER

EXIT
Transmit Subroutine

TRANSMIT

IS THE PHONE READY? NO, EXIT

T2

IS THE STATE EQUAL TO ZERO? NO, GO TO T9 STATE = RCL

YES

T3

SAVE SAMPLE POINTER IN THE SEND POINTER
SAVE AVERAGE POINTER IN THE AVERAGE POINTER BUFFER

T4

INCREMENT THE STATE

T5

LOAD SEND COUNT FROM SAMPLE COUNT IN PARAMETER TABLE

T9

TURN ON THE POWER
TURN ON THE POWER FLAG

IS THE LINK READY FOR ANOTHER TRANSMISSION? NO, EXIT

T10

IS THE SEND COUNT EQUAL TO 0? NO, GO TO T16

YES

PATCH6

IS THE STATE EQUAL TO 1? NO

YES

T11

T12
Transmit Subroutine (cont'd.)

T11
TURN POWER FLAG OFF
TURN POWER OFF
HANGUP TELEPHONE
SET STATE EQUAL TO 0

EXIT

T12
LOAD THE SEND COUNTER WITH THE AVERAGE SEND COUNT

T13
PATCH?
LOAD THE SEND POINTER FROM THE AVERAGE POINTER BUFFER

T14
INCREMENT THE STATE

T16
DOES THE CHANNEL COUNT EQUAL 255?

T18
NO
DECREMENT SEND POINTER AND WRAP AROUND IF NEC.
LOAD TRANSMITTER WITH DATA POINTED TO BY THE SEND POINTER AND SEND IT

T19
T21

T20
NO
INCREMENT CHANNEL COUNT

T22
EXIT

DECREMENT SEND COUNT
SET THE CHANNEL COUNT TO 255

EXIT

YES

LOAD TRANSMITTER WITH SEND POINTER AND START TRANSMISSION

T17

YES

T15

T20

T17

T15

T17

T15

T17

T15

T17

T15

T17

T15

T17

T15

T17

T15

T17
Reset Program

RESET BEGIN

| LOAD R7 WITH THE ADDRESS OF THE SWITCH REGISTER |
| PATCH1 |
| LOAD XREG WITH THE VALUE IN THE SWITCH REGISTER |
| PR1 |
| TRANSFER THE CONTENTS OF THE XREG TO PCC |
| SET UP R5 AND R6 FOR JSR AND RTS OPERATION |
| SET R2 TO 10FF16 |
| SET UP R1 FOR INTERRUPT ROUTINE |
| SET R16 TO A TEN MINUTE COUNT |
| START EXECUTING USING PCC AS THE PROGRAM COUNTER |

XREG = R8

PCC = R3

R2 = STACK POINTER FOR INT, JSR AND RTS

R16 IS USED BY THE INTERRUPT ROUTINE TO COUNT SECONDS
Interrupt Routine

INTRT
- RESTORE R7 AND D-REGISTER
- RESTORE X AND P WITH RETURN FROM INTERRUPT

INTER
- SAVE X AND P
- SAVE D-REGISTER
- SAVE R7

PATCH2
- SET R7 TO FFFF₁₆

GET INTERRUPT STATUS

PR2
- SECOND CLOCK INTERRUPT?
  - NO
  - YES
    - TURN ON SECOND FLAG

DECREMENT SECOND COUNT REGISTER

SECOND COUNT REGISTER EQUAL TO 0?
- NO
- YES
  - TURN ON INTERVAL FLAG
  - RESET THE SECOND COUNT REGISTER

SECOND FLAG = R15L
ON = 255
OFF = 0

SECOND COUNT REGISTER = R16

INTERVAL FLAG = R15H
Xadr Loop

XADR1

RETURN TO R3 AS PROGRAM COUNTER

XADR

LOAD XREG WITH VALUE FOLLOWING THE XAD INSTRUCTION

SET X EQUAL TO XREG

Jsr/Rts Loop

JSR/RTS

SEE DISCUSSION OF SUBROUTINE SOFTWARE IN RCA SOFTWARE LITERATURE
APPENDIX C
PROGRAM LISTING

C
0000     org C
0002     "set assembler constants"
0004     255+256+6 = switches
0006     0=r0
0008     1=r1
000A     2=r2
000C     3=r3
000E     4=r4
0010     5=r5
0012     6=r6
0014     7=r7
0016     R=r8=xreo
0018     9=r9
001A     10=r10
001C     11=r11
001E     12=r12
0020     13=r13
0022     14=r14
0024     15=r15
0026     r3=pcc
0028     600=scount
002A     'begin of reset program'
002C     org 0
002E     60C = s
0030     'begin of reset program'
0032     ldi (xadr hi)
0034     phi r7
0036     ldi xadr
0038     plo r7
003A     30F7
003C     br patch1
003E     0009 48  print: lda xreo
0040     000A 47  plo pcc
0042     B7     lda xreo
0044     phi pcc
0046     000C  B3  ldi (jsrst hi)
0048     phi r4
004A     000D  F8C0 ldi (jsrst hi)
004C     phi r4
004E     0010  F820 ldi (retst hi)
0050     phi r5
0052     0012  B5  ldi jsrst
0054     phi r5
0056     0013  F814 plo r6
0058     0015  A5  ldi retst
005A     0016  A5  plo r5
005C     001B  A7  ldi 16
005E     0019  F810 ldi 16
0060     001A  B2  phi r2
0062     001B  FRFF ldi 255
0064     001C  A2  plo r2
0066     'stp1' is
0068     F800 ldi (inter hi) %set up interrupt reg
006A     001F  B1  phi r1
006C     0021  F834 ldi inter
006E     0022  A1  plo r1
0070     0023  F832 ldi (scount hi)
0072     phi r1
0074     0025  B8  phi r1
0076     0027  FR58 ldi scount
0078     0028  AF  plo r1
007A     0029  D3  sep r3
007C     002A  3000 tr 0
007E     002B  C3  jmp to start of program
0080     002C  3000 tr 0
0082     'stp1' is
0084     002D  47  inttr: lda r2
0086     002E  47  phi r2
0088     002F  42  lda r2
008A     0030  42  plo r2
008C     0031  A7  plo r7
0032 42 lda r2
0033 70 ret r2
0034 22 inter: dec r2
0035 78 sav r2
0036 22 dec r2
0037 52 stn r2
0038 22 dec r2
0039 87 glo r7
003A 52 stn r2
003B 22 dec r7
003C 97 ghi r7
003D 52 stn r2
003E FF FF ldi 255
0040 30 01 hr patch?
0042 FA 0D pr?: ani 128
0044 32 2E bz intrt
0046 FF FF ldi 255
0048 AE plo re
0049 2F dec rf
004A 8F glo rf
004B 3A 2E bnz intrt
004C 9F ghi rf
004D 3A 2E bnz intrt
004E F8 02 ldi (scount hi)
0050 9F 01 phi rf
0052 01 ldi scount
0053 9F 01 phi re
0054 01 plp rf
0055 AF plo rf
0056 F8 FF ldi 255
0058 BE phi re
0059 30 2E br intrt
005C 01 org (5*16+12)
005C D5 compose: rts %get channel
005D 0F FE shl
005E 0E FE shl
005F FE shl
0060 0E FE shl
0061 73 stxd
0062 2F 0A ghi ra
0063 9A phi r9
0064 06 shr
0065 FA 07 ani 7
0066 BA phi ra
0067 08 A glo ra
0068 76 shrc
0069 AA plo ra
006A 99 ghi r9
006B 0E FE shlc
006C 33 77 bdf compos1
006F 8A glo ra
0070 FD 00 sdi 0
0072 AA plo ra
0073 9A ghi ra
0074 7D 00 sbdi 0
0076 BA phi ra
0077 9A compos1: ghi ra
0078 FA DF ani 15
0079 6F ixr
007A F1 o0r
007B F0 08 xri
007C FB DB phi ra
007E BA phi ra

%return from interrupt
%shift into ms nibble
%and save on stack
%get top byte and save
%shift and remove gart
%save
%shift into bottom byte
%and save
%evaluate sign bit
%no sign inversion
%subtract value from
%stack on channel
%complement sign bit
ORPC is a program to set up a to d address (large = 2048+2 = (large+2))

1. "tp2" is

2. org

3. D7FF02 send: xad

4. D7FF02 send: xad (large = 255+256 times+2 = (large+2))

5. 8A glo

6. 58 stn

7. 1R inc

8. 9A ohi

9. 58 stn

10. 1B inc

11. 08 ldn

12. F902 ori

13. 58 stn

14. 08 ldn

15. FAFD ani

16. 58 stn

17. 08 send1: ldn

18. FA04 ani

19. 3295 bz send1

20. D5 rts

21. 41 org 160

22. D7FF02 atod: xad

23. atd (large + 2 = atd) %set up a to d address

24. 89 glo

25. 58 stn

26. C4 nop

27. C4 nop

28. C4 nop

29. 48 lda

30. 48 lda

31. AA plc

32. 08 ldn

33. BA phi

34. 58 stn

35. D5 rts

36. D7FF02 poweron: xad

37. (large - 2) %set up power switch

38. 08 ldn

39. D9E ani

40. 58 stn

41. D5 rts

42. 46 waitit: lda

43. 88 phi

44. 46 lda

45. A8 plc

46. 88 wait1: glo

47. 32C7 bz wait11

48. 28 wait22: dec

49. 30C1 br wait1

50. 98 wait11: phi

51. 3AC4 bnz wait22

52. D5 rts

53. D3 xadr1: sep

54. 43 xadr: lda

55. B8 phi

56. 43 lda

57. A8 plc

58. 43 phi

59. 30CB br xadr1
DOO3 D x  exita:  sep  r3
DOO4 E7  jsrst:  sex  r2
DOO5 22  dec  r2
DOO6 96  ghi  r6
DOO7 73  stdx
DOO8 86  glo  r6
DOO9 73  stdx
DOOA 93  phi  r3
DOOB 86  phi  r6
DOOC 83  glo  r3
DOOED A6  plo  r6
DOOEE 46  lda  r6
DOOFG B3  phi  r3
DOOHO 46  lda  r6
DOO1 A3  plo  r3
DOOE2 30H3  br  exita
DOOE4 C4  nop
DOOE5 D3  exitr:  sep  r3
DOOE6 96  retst:  ghi  r6
DOOE7 R3  phi  r3
DOOE8 86  glo  r6
DOOE9 A3  plo  r3
DOOEAE E2  sex  r2
DOOEBA 12  inc  r2
DOOEBC 72  ldxa
DOOEBD A6  plo  r6
DOOE0 72  ldxa
DOOE0F B6  phi  r6
DOOE10 30E5  br  exitr
DOOE11 A7  patch2:  plo  r7
DOOE12 B7  phi  r7
DOOE13 07  ldn  r7
DOOE14 3042  br  pr2
DOOE15 F8FF  patch1:  ldi  (switches hi)
DOOE16 B8  phi  xreg
DOOE17 F806  ldi  switches
DOOE18 AR  plo  xreg
DOOE19 3009  br  pr1
DOOE1A  'pt3' is,
D000D  org  256
D000D  D402DA  start:  jsr  setup
D000D  D4010B  jsr  sample
D000D  D40200  jsr  transmit
D000D  3033  br  (start+3)
D000D  30C3  sample:  br  patch3
D000D  98  s2:  ghi  rb
D000D  73  stdx
D000D  88  glo  rb
D000D  73  stdx
D000D  9A  ghi  ra
D000D  73  stdx
D000D  8A  glo  ra
D000D  73  stdx
D000D  99  ghi  r9
D000D  73  stdx
D000D  89  glo  r9
D000D  73  stdx
D000D  90  ghi  r0
D000D  73  stdx
D000D  B71000  s3:  xad  pbuf (4096=pbuf)
011E 72  ldx a
011F 7F  phi  rf
0120 72  ldx a
0121 8F  plo  rf
0122 D400AD s4:  jsr  poweron
0125 F800 s5:  ldi  0
0127 3D00  br  patch4
0129 8E  s6:  glo  re
012A 3229  bz  s6
012C F800  ldi  0
012E AE  plo  re
012F D400AD s7:  jsr  atod
0132 D4005C s8:  jsr  compose
0135 9A  ghi  ra
0136 5D  stn  rd
0137 1D  inc  rd
0138 RA  glo  ra
0139 5D  stn  rd
013A 1D  inc  rd
013B 9D  glo  rd
013C 3A48  bnz  s10
013E 7100B  xad  (pbuf+11)
0141 9D  ghi  rd
0142 F5  sd
0143 3A48  bnz  s10
0145 6D  irx
0146 F0  ldx
0147 0D  phi  rd
0148 71020 s10:  xad  avgb  (pbuf+3?=avgb)%
014F 89  glo  r9
014C FE  shl
014D FE  shl
014E FC70  adi  avgt
0150 8F  plo  xreg
0151 RA  glo  ra
0152 F4  add
0153 58  stn  xreg
0154 18  inc  xreg
0155 9A  ghi  ra
0156 FAOF  ani  15
0158 74  adc
0159 58  stn  xreg
015A 1F  inc  xreg
015B F80D  ldi  0
015D 74  adc
015E 58  stn  xreg
015F D71004 s11:  xad  (pbuf+4)
0162 89  glo  r9
0163 F5  sd
0164 3269  hz  s13
0166 19  s12:  inc  r9
0167 3029  br  s6
0169 80  s13:  glo  r0
016A 3A6F  bnz  s14
016C D40055  jsr  poweroff
016F E7100E s14:  xad  (pbuf+14)
0172 72  ldx a
0173 BB  phi  rb
0174 F0  ldx
0175 AB  plo  rb
<table>
<thead>
<tr>
<th>Address</th>
<th>Code</th>
<th>Mnemonic</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>01c4</td>
<td>3A0D</td>
<td>bnz</td>
<td>s2</td>
</tr>
<tr>
<td>01c6</td>
<td>D5</td>
<td>rts</td>
<td></td>
</tr>
<tr>
<td>01d0</td>
<td>org</td>
<td></td>
<td>(256+(13*16))</td>
</tr>
<tr>
<td>01d0</td>
<td>A9</td>
<td>patch4:</td>
<td>plo</td>
</tr>
<tr>
<td>01d1</td>
<td>AE</td>
<td>plo</td>
<td>re</td>
</tr>
<tr>
<td>01d2</td>
<td>RE</td>
<td>phi</td>
<td>re</td>
</tr>
<tr>
<td>01d3</td>
<td>0400A0</td>
<td>jsr</td>
<td>atod</td>
</tr>
<tr>
<td>01e6</td>
<td>3029</td>
<td>br</td>
<td>s6</td>
</tr>
<tr>
<td>0200</td>
<td>org</td>
<td></td>
<td>512</td>
</tr>
<tr>
<td>0200</td>
<td>3503</td>
<td>transmit:</td>
<td>b2</td>
</tr>
<tr>
<td>0202</td>
<td>D5</td>
<td>rts</td>
<td></td>
</tr>
<tr>
<td>0203</td>
<td>80</td>
<td>t2:</td>
<td>glo</td>
</tr>
<tr>
<td>0204</td>
<td>3A1C</td>
<td>hnz</td>
<td>t9</td>
</tr>
<tr>
<td>0206</td>
<td>9D</td>
<td>t3:</td>
<td>ghi</td>
</tr>
<tr>
<td>0207</td>
<td>BB</td>
<td>phi</td>
<td>rb</td>
</tr>
<tr>
<td>0208</td>
<td>8D</td>
<td>glo</td>
<td>rd</td>
</tr>
<tr>
<td>0209</td>
<td>AB</td>
<td>plo</td>
<td>rh</td>
</tr>
<tr>
<td>020A</td>
<td>9C</td>
<td>phi</td>
<td>ra</td>
</tr>
<tr>
<td>020B</td>
<td>BA</td>
<td>phi</td>
<td>rc</td>
</tr>
<tr>
<td>020C</td>
<td>BC</td>
<td>glo</td>
<td>rc</td>
</tr>
<tr>
<td>020D</td>
<td>AA</td>
<td>plo</td>
<td>ra</td>
</tr>
<tr>
<td>020E</td>
<td>10</td>
<td>t4:</td>
<td>inc</td>
</tr>
<tr>
<td>020F</td>
<td>D71005</td>
<td>t5:</td>
<td>xad</td>
</tr>
<tr>
<td>0212</td>
<td>72</td>
<td>ldx a</td>
<td></td>
</tr>
<tr>
<td>0213</td>
<td>89</td>
<td>phi</td>
<td>r9</td>
</tr>
<tr>
<td>0214</td>
<td>72</td>
<td>ldx a</td>
<td></td>
</tr>
<tr>
<td>0215</td>
<td>A9</td>
<td>plo</td>
<td>r9</td>
</tr>
<tr>
<td>0216</td>
<td>F8FF</td>
<td>ldi</td>
<td>255</td>
</tr>
<tr>
<td>0218</td>
<td>80</td>
<td>phi</td>
<td>r0</td>
</tr>
<tr>
<td>0219</td>
<td>D400AD</td>
<td>jsr</td>
<td>poweron</td>
</tr>
<tr>
<td>021C</td>
<td>3425</td>
<td>t9:</td>
<td>b1</td>
</tr>
<tr>
<td>021E</td>
<td>D5</td>
<td>rts</td>
<td></td>
</tr>
<tr>
<td>021F</td>
<td>C4</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>0220</td>
<td>C4</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>0221</td>
<td>C4</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>0222</td>
<td>C4</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>0223</td>
<td>C4</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>0224</td>
<td>C4</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>0225</td>
<td>99</td>
<td>t10:</td>
<td>ghi</td>
</tr>
<tr>
<td>0226</td>
<td>3A42</td>
<td>bnz</td>
<td>t16</td>
</tr>
<tr>
<td>0228</td>
<td>89</td>
<td>glo</td>
<td>t16</td>
</tr>
<tr>
<td>0229</td>
<td>3A42</td>
<td>bnz</td>
<td>t16</td>
</tr>
<tr>
<td>0228</td>
<td>C0033E</td>
<td>lbr</td>
<td>patch6</td>
</tr>
<tr>
<td>022E</td>
<td>F800</td>
<td>t11:</td>
<td>ldi</td>
</tr>
<tr>
<td>0230</td>
<td>A0</td>
<td>plo</td>
<td>r0</td>
</tr>
<tr>
<td>0231</td>
<td>D400B5</td>
<td>jsr</td>
<td>poweroff</td>
</tr>
<tr>
<td>0234</td>
<td>7B</td>
<td>seq</td>
<td></td>
</tr>
<tr>
<td>0235</td>
<td>7A</td>
<td>reo</td>
<td></td>
</tr>
<tr>
<td>0236</td>
<td>D5</td>
<td>rts</td>
<td></td>
</tr>
<tr>
<td>0237</td>
<td>D71007</td>
<td>t12:</td>
<td>xad</td>
</tr>
<tr>
<td>023A</td>
<td>72</td>
<td>ldx a</td>
<td></td>
</tr>
<tr>
<td>023B</td>
<td>B9</td>
<td>phi</td>
<td>r9</td>
</tr>
<tr>
<td>023C</td>
<td>77</td>
<td>ldx a</td>
<td></td>
</tr>
<tr>
<td>023D</td>
<td>A9</td>
<td>plo</td>
<td>r9</td>
</tr>
<tr>
<td>023E</td>
<td>0A</td>
<td>t13:</td>
<td>ghi</td>
</tr>
<tr>
<td>023F</td>
<td>C00330</td>
<td>lbr</td>
<td>patch7</td>
</tr>
<tr>
<td>0242</td>
<td>90</td>
<td>t16:</td>
<td>ghi</td>
</tr>
<tr>
<td>0243</td>
<td>FDFF</td>
<td>sdi</td>
<td>255</td>
</tr>
<tr>
<td>0245</td>
<td>3A52</td>
<td>bnz</td>
<td>t1R</td>
</tr>
<tr>
<td>0247</td>
<td>D7FF03</td>
<td>t17:</td>
<td>xad</td>
</tr>
<tr>
<td>Address</td>
<td>Opcode</td>
<td>Mnemonic</td>
<td>Operands</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>029F</td>
<td>FF04</td>
<td>ldi</td>
<td>4</td>
</tr>
<tr>
<td>029F</td>
<td>FF01</td>
<td>f1:</td>
<td>smi</td>
</tr>
<tr>
<td>029D</td>
<td>FF01</td>
<td>f1:</td>
<td>stn</td>
</tr>
<tr>
<td>029E</td>
<td>FF01</td>
<td>f1:</td>
<td>glo</td>
</tr>
<tr>
<td>029F</td>
<td>FF01</td>
<td>f1:</td>
<td>shl</td>
</tr>
<tr>
<td>02A0</td>
<td>FF01</td>
<td>f1:</td>
<td>plo</td>
</tr>
<tr>
<td>02A1</td>
<td>FF01</td>
<td>f1:</td>
<td>ghi</td>
</tr>
<tr>
<td>02A2</td>
<td>FF01</td>
<td>f1:</td>
<td>shlc</td>
</tr>
<tr>
<td>02A3</td>
<td>FF01</td>
<td>f1:</td>
<td>phi</td>
</tr>
<tr>
<td>02A4</td>
<td>FF01</td>
<td>f1:</td>
<td>glo</td>
</tr>
<tr>
<td>02A5</td>
<td>FF01</td>
<td>f1:</td>
<td>shlc</td>
</tr>
<tr>
<td>02A6</td>
<td>FF01</td>
<td>f1:</td>
<td>plo</td>
</tr>
<tr>
<td>02A7</td>
<td>FF01</td>
<td>f1:</td>
<td>ghi</td>
</tr>
<tr>
<td>02A8</td>
<td>FF01</td>
<td>f1:</td>
<td>shlc</td>
</tr>
<tr>
<td>02A9</td>
<td>FF01</td>
<td>f1:</td>
<td>phi</td>
</tr>
<tr>
<td>02AA</td>
<td>FF01</td>
<td>f1:</td>
<td>ldn</td>
</tr>
<tr>
<td>02AB</td>
<td>FF01</td>
<td>f1:</td>
<td>bnz</td>
</tr>
<tr>
<td>02AD</td>
<td>FF01</td>
<td>f1:</td>
<td>stn</td>
</tr>
<tr>
<td>02AE</td>
<td>FF01</td>
<td>f1:</td>
<td>ghi</td>
</tr>
<tr>
<td>02AF</td>
<td>FF01</td>
<td>f1:</td>
<td>glo</td>
</tr>
<tr>
<td>02B0</td>
<td>FF01</td>
<td>f1:</td>
<td>shlc</td>
</tr>
<tr>
<td>02B4</td>
<td>FF00</td>
<td>f2:</td>
<td>ldi</td>
</tr>
<tr>
<td>02B6</td>
<td>FF00</td>
<td>f2:</td>
<td>add</td>
</tr>
<tr>
<td>02B7</td>
<td>FF00</td>
<td>f2:</td>
<td>stn</td>
</tr>
<tr>
<td>02B8</td>
<td>FF00</td>
<td>f2:</td>
<td>ghi</td>
</tr>
<tr>
<td>02B9</td>
<td>FF00</td>
<td>f2:</td>
<td>shr</td>
</tr>
<tr>
<td>02BA</td>
<td>FF00</td>
<td>f2:</td>
<td>phi</td>
</tr>
<tr>
<td>02BB</td>
<td>FF00</td>
<td>f2:</td>
<td>glo</td>
</tr>
<tr>
<td>02BC</td>
<td>FF00</td>
<td>f2:</td>
<td>shlc</td>
</tr>
<tr>
<td>02BD</td>
<td>FF00</td>
<td>f2:</td>
<td>plo</td>
</tr>
<tr>
<td>02BE</td>
<td>FF00</td>
<td>f2:</td>
<td>ghi</td>
</tr>
<tr>
<td>02BF</td>
<td>FF00</td>
<td>f2:</td>
<td>shlc</td>
</tr>
<tr>
<td>02C0</td>
<td>FF00</td>
<td>f2:</td>
<td>phi</td>
</tr>
<tr>
<td>02C1</td>
<td>FF00</td>
<td>f2:</td>
<td>glo</td>
</tr>
<tr>
<td>02C2</td>
<td>FF00</td>
<td>f2:</td>
<td>shlc</td>
</tr>
<tr>
<td>02C3</td>
<td>FF00</td>
<td>f2:</td>
<td>plo</td>
</tr>
<tr>
<td>02C4</td>
<td>FF00</td>
<td>f2:</td>
<td>br</td>
</tr>
<tr>
<td>02C6</td>
<td>FF00</td>
<td>f3:</td>
<td>ldn</td>
</tr>
<tr>
<td>02C7</td>
<td>FF00</td>
<td>f3:</td>
<td>plo</td>
</tr>
<tr>
<td>02C8</td>
<td>FF00</td>
<td>f3:</td>
<td>ldi</td>
</tr>
<tr>
<td>02CA</td>
<td>FF00</td>
<td>f5:</td>
<td>stn</td>
</tr>
<tr>
<td>02CB</td>
<td>FF00</td>
<td>f5:</td>
<td>glo</td>
</tr>
<tr>
<td>02CC</td>
<td>FF00</td>
<td>f5:</td>
<td>shr</td>
</tr>
<tr>
<td>02CD</td>
<td>FF00</td>
<td>f5:</td>
<td>plo</td>
</tr>
<tr>
<td>02CE</td>
<td>FF00</td>
<td>f5:</td>
<td>ghi</td>
</tr>
<tr>
<td>02CF</td>
<td>FF00</td>
<td>f5:</td>
<td>shlc</td>
</tr>
<tr>
<td>02D0</td>
<td>FF00</td>
<td>f5:</td>
<td>phi</td>
</tr>
<tr>
<td>02D1</td>
<td>FF00</td>
<td>f5:</td>
<td>glo</td>
</tr>
<tr>
<td>02D2</td>
<td>FF00</td>
<td>f5:</td>
<td>shlc</td>
</tr>
<tr>
<td>02D3</td>
<td>FF00</td>
<td>f5:</td>
<td>plo</td>
</tr>
<tr>
<td>02D4</td>
<td>FF00</td>
<td>f5:</td>
<td>ldn</td>
</tr>
<tr>
<td>02D5</td>
<td>FF00</td>
<td>f5:</td>
<td>smi</td>
</tr>
<tr>
<td>02D7</td>
<td>FF00</td>
<td>f5:</td>
<td>bnz</td>
</tr>
<tr>
<td>02D9</td>
<td>FF00</td>
<td>f5:</td>
<td>rts</td>
</tr>
<tr>
<td>02DA</td>
<td>FF00</td>
<td>setup:</td>
<td>xad</td>
</tr>
<tr>
<td>02DD</td>
<td>FF00</td>
<td>setup:</td>
<td>ldi</td>
</tr>
<tr>
<td>02DF</td>
<td>FF00</td>
<td>setup:</td>
<td>phi</td>
</tr>
<tr>
<td>02E0</td>
<td>FF00</td>
<td>setup:</td>
<td>ldi</td>
</tr>
<tr>
<td>02E2</td>
<td>FF00</td>
<td>setup:</td>
<td>plo</td>
</tr>
</tbody>
</table>
024A  9B     ghi  rb    %
024B  FADF  ani  15    %
024D  73     std   %
024E  8B     glo  rb    %
024F  73     std   %
0250  306C   br   t10   %
0252  8B     glo  rb    %
0253  3A65   bnz  t18a  %
0255  D710CC xad  (ptbuf+12)  %
0258  9B     ghi  rb    %
0259  F5     sd    %
025A  3262   bz   t18fin %
025C  28     dec  xreq  %
025D  28     dec  xreq  %
025E  9B     ghi  rb    %
025F  F5     sd    %
0260  3A65   bnz  t18a  %
0262  28     t18fin: dec  xreq  %
0263  FC     ldx   %
0264  RP     phi  rb    %
0265  D7FF03 t18a: xad  (larse+1)  %
0268  C0034F lbr  patch8
026B  C4     nop
026C  C4     t19:  nop
026D  C4     nop
026E  C4     nop
026F  C4     nop
0270  C4     nop
0271  C4     nop
0272  C4     nop
0273  C4     nop
0274  C4     nop
0275  C4     nop
0276  D7104  t21: xad  (pbuf+4)  %
0279  90     ghi  r0    %
027A  F5     sd    %
027B  3282   bz   t22   %
027D  90     t20:  ghi  r0    %
027E  FC01   adi  1    %
0280  B0     phi  r0    %
0281  D5     rts   %
0282  29     t22:  dec  r9    %
0283  F8FF   ldi  255    %
0285  B0     phi  r0    %
0286  D5     rts   %
0287  D71020  float: xad  avgb   %
028A  89     glo  r9    %
028B  FE     shl   %
028C  FE     shl   %
028D  FC20   adi  avob  %
028F  A8     plo  xreg  %
0290  72     ldx   %
0291  AA     plo  ra    %
0292  72     ldx   %
0293  RA     phi  ra    %
0294  72     ldx   %
0295  AB     plo  rb    %
0296  F800   ldi  0    %
0298  BB     phi  rb    %
<table>
<thead>
<tr>
<th>Address</th>
<th>Opcode</th>
<th>Label</th>
<th>Operation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>02E3</td>
<td>80 0F</td>
<td>ldi</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>02E5</td>
<td>80 A</td>
<td>plo</td>
<td>ra</td>
<td></td>
</tr>
<tr>
<td>02E6</td>
<td>49 sup1</td>
<td>lda</td>
<td>r9</td>
<td></td>
</tr>
<tr>
<td>02E7</td>
<td>5F</td>
<td>stn</td>
<td>xreg</td>
<td></td>
</tr>
<tr>
<td>02E8</td>
<td>18</td>
<td>inc</td>
<td>xreg</td>
<td></td>
</tr>
<tr>
<td>02E9</td>
<td>2A</td>
<td>dec</td>
<td>ra</td>
<td></td>
</tr>
<tr>
<td>02EA</td>
<td>8A</td>
<td>glo</td>
<td>ra</td>
<td></td>
</tr>
<tr>
<td>02EB</td>
<td>3AE6</td>
<td>bnz</td>
<td>sup1</td>
<td></td>
</tr>
<tr>
<td>02ED</td>
<td>D7 FFFF</td>
<td>xad</td>
<td>(64*1024-1)</td>
<td></td>
</tr>
<tr>
<td>02F0</td>
<td>FA 80</td>
<td>ldi</td>
<td>(9*16)</td>
<td></td>
</tr>
<tr>
<td>02F2</td>
<td>58</td>
<td>stn</td>
<td>xreg</td>
<td></td>
</tr>
<tr>
<td>02F3</td>
<td>7B</td>
<td>seq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02F4</td>
<td>80 000</td>
<td>ldi</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>02F6</td>
<td>AD</td>
<td>plo</td>
<td>r0</td>
<td></td>
</tr>
<tr>
<td>02F7</td>
<td>AC</td>
<td>plo</td>
<td>rc</td>
<td></td>
</tr>
<tr>
<td>02F8</td>
<td>AD</td>
<td>plo</td>
<td>rd</td>
<td></td>
</tr>
<tr>
<td>02F9</td>
<td>C0 0320</td>
<td>lbr</td>
<td>patch9</td>
<td></td>
</tr>
<tr>
<td>0300</td>
<td>02 58</td>
<td>supl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0302</td>
<td>00 06</td>
<td>wrd</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>0304</td>
<td>02</td>
<td>byt</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0305</td>
<td>C0 04</td>
<td>wrd</td>
<td>180</td>
<td></td>
</tr>
<tr>
<td>0307</td>
<td>C0 1E</td>
<td>wrd</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>0309</td>
<td>13</td>
<td>byt</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>030A</td>
<td>12</td>
<td>byt</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>030B</td>
<td>18</td>
<td>byt</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>030C</td>
<td>13</td>
<td>byt</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>030D</td>
<td>00 06</td>
<td>wrd</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>0320</td>
<td>D7 03D0</td>
<td>patch9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0323</td>
<td>42</td>
<td>lda</td>
<td>xreg</td>
<td></td>
</tr>
<tr>
<td>0324</td>
<td>RB</td>
<td>phi</td>
<td>rd</td>
<td></td>
</tr>
<tr>
<td>0325</td>
<td>0B</td>
<td>ldn</td>
<td>xreg</td>
<td></td>
</tr>
<tr>
<td>0326</td>
<td>BC</td>
<td>phi</td>
<td>rc</td>
<td></td>
</tr>
<tr>
<td>0327</td>
<td>FA FF</td>
<td>ldi</td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>0329</td>
<td>BE</td>
<td>nhi</td>
<td>re</td>
<td></td>
</tr>
<tr>
<td>032A</td>
<td>B5</td>
<td>rts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0330</td>
<td>0B</td>
<td>patch7</td>
<td>phi</td>
<td>rb</td>
</tr>
<tr>
<td>0331</td>
<td>RA</td>
<td>glo</td>
<td>ra</td>
<td></td>
</tr>
<tr>
<td>0332</td>
<td>AB</td>
<td>plo</td>
<td>rb</td>
<td></td>
</tr>
<tr>
<td>0333</td>
<td>10</td>
<td>inc</td>
<td>r0</td>
<td></td>
</tr>
<tr>
<td>0334</td>
<td>C0 242</td>
<td>lbr</td>
<td>t16</td>
<td></td>
</tr>
<tr>
<td>033E</td>
<td>80</td>
<td>patch6</td>
<td>glc</td>
<td>r0</td>
</tr>
<tr>
<td>033F</td>
<td>FF F1</td>
<td>smi</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0341</td>
<td>C7 0327</td>
<td>lbz</td>
<td>t12</td>
<td></td>
</tr>
<tr>
<td>0344</td>
<td>C0 022E</td>
<td>lbr</td>
<td>t11</td>
<td></td>
</tr>
<tr>
<td>034F</td>
<td>2B</td>
<td>patch8</td>
<td>dec</td>
<td>rb</td>
</tr>
<tr>
<td>0350</td>
<td>OB</td>
<td>ldn</td>
<td>rb</td>
<td></td>
</tr>
<tr>
<td>0351</td>
<td>5B</td>
<td>stn</td>
<td>xreg</td>
<td></td>
</tr>
<tr>
<td>0352</td>
<td>1R</td>
<td>inc</td>
<td>xreg</td>
<td></td>
</tr>
<tr>
<td>0353</td>
<td>2B</td>
<td>dec</td>
<td>rb</td>
<td></td>
</tr>
<tr>
<td>0354</td>
<td>OB</td>
<td>ldr</td>
<td>rb</td>
<td></td>
</tr>
<tr>
<td>0355</td>
<td>5B</td>
<td>stn</td>
<td>xreg</td>
<td></td>
</tr>
<tr>
<td>0356</td>
<td>CO 26F</td>
<td>lbr</td>
<td>t19</td>
<td></td>
</tr>
<tr>
<td>0380</td>
<td></td>
<td>org</td>
<td>(pmbf+16*8)</td>
<td></td>
</tr>
<tr>
<td>0380</td>
<td>F5</td>
<td>patch5</td>
<td>sd</td>
<td>p51</td>
</tr>
<tr>
<td>0381</td>
<td>32 86</td>
<td>bz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>Instruction</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0383 C00100</td>
<td>lbr</td>
<td>s21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0386 D7105F</td>
<td>xad</td>
<td>(avgh+63)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0389 F80D</td>
<td>ldi</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>038B 73</td>
<td>std</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>038C 8F</td>
<td>glo</td>
<td>xreg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>038D FD1F</td>
<td>std</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>038F 3A89</td>
<td>bnz</td>
<td>p52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0391 C00181</td>
<td>lbr</td>
<td>sexit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>