Information on Electronic Arrays Chipset used in MITS 1440 Calculator (and others)
R. Grieb Last updated 2/28/09

Most of the following information was collected by reading the construction/owners manual for the MITS 1440, studying US Patent #3800129, and making measurements on a partially working MITS 1440. I have tried to make it as accurate as possible, and indicate where I am not sure about some detail. The above-mentioned patent has much more detailed information on each section of the design, but applies more directly to the chip set used in the (earlier) 816 calculator, and there are a few places where the design is a little different from the 1440 chip set. If anyone has any information on these chips, please contact me. I will not be responsible for any damage caused by errors in this document.

Overview:

The MITS 1440 calculator was introduced in 1973. It uses a six-chip set, made by a company called Electronic Arrays. The functions of the six chips are: input, control, program ROM, ALU, registers, and output. These chips handle keyboard scanning, calculation, error and overflow detection and some of the output processing. Either nixie tube or LED displays could be interfaced to the same chip set. The 1440 used seven-segment LEDs to implement a 14-digit display. Electronic Arrays was bought by NEC in the late 1970’s.

Conventions:

In 1973, many MOS chips were designed with only PMOS transistors. As a result, they used “negative logic”, and ran from negative power supply voltages. In the case of the 1440, the main logic supply was -14V, and a -26V bias supply was used well. With negative logic, 0V is considered 0 or off, and -14V is considered 1 or on. Since -14V is more negative than 0V, I will use a lower case b after signals that operate this way to indicate that they are active low. The display portion of the 1440 used 7400 series TTL ICs operating at 5V, so the PMOS display signals had to be converted to those levels.

Clocking and timing:

All chips run off of the same master clock “CLK”, generated by a TTL RC oscillator circuit. In my unit, the clock frequency is about 157 KHz. The rise time at the collector of Q45 is on the order of 20 nSec, and the fall time is about 200 nSec. (The manual states that both are under 100 nSec) This calculator uses a “time slot” approach to processing. BCD Digits are four bits each. Moving a digit from one chip to another takes four clock cycles, one per bit. The ALU contains a 4-bit binary counter that defines 16 digit “time slots”. Each time slot is four clock cycles long. Sixteen time slots, numbered 0-15, make up a complete “frame”. Two decoded time slot strobes DT0b (digit time zero) and DTFb (digit time fifteen) are fed from the ALU to the other chips to keep them all in sync. The operands are stored in 64-bit registers which are constantly being shifted in a circle, so accessing any given digit position involves timing the control signals to be active during the correct time slot for that digit. Data is shifted serially between the chips. The opcode control signals from the ROM chip are presented in parallel to the control, ALU and register chips.

Reset:

A power-up reset circuit is provided, as well as a clear key. The clear key is fed to the ALU chip, where it is synced to the clock and then fed to the other chips.

Input chip (FDY 310B-7014)

This chip drives four keyboard column outputs in sequence, while scanning eight keyboard row inputs, also in sequence. A single “scan counter” is decoded to select the current column or row to be scanned. The column is selected by the two MSBits, and the row by the next three bits. If a matrix key is pressed, the scan counter will stop with a binary value that can be used to tell which key is down. Output data appears serially on the KDb pin (LSB first), with one bit per main clk cycle. The output from this chip is either a 7-bit key code, or a 4-bit BCD number value. If a number key is pressed, the output data sent immediately will be one of two codes, (the same for all numbers) depending on whether the decimal point has been entered previously. Sometime later, the TFb (transmit figure) signal will be sent from the control chip, which will cause the input chip to send the actual 4-bit BCD value of the number (0-9). If a command key is pressed, the code sent will correspond to the key itself. Some commands cause their code to be sent immediately, and some are saved in the input chip and sent when the next command key is pressed. (You can find much more about “delayed command” operation in the patent.) The 4-bit BCD values sent for number keys or the 7-bit codes sent for command keys are a direct result of that key’s position in the keyboard matrix. All command codes and both “number key pressed” codes are 7 bits, with the LSB equal to 1. These are shifted into a register in the program ROM chip to form the starting address for a “program” in the ROM (a sequence of addresses) which then generates control signals (opcode bits) for the other chips to execute the entered command.

Note: When any number key is pressed, one of two initial “number key pressed” codes is sent. At this point, the input chip is waiting for the TFb signal to tell it to send the 4-bit BCD code for that number. (Actually, it is the BUSYb signal that causes it to wait whenever BUSYb is at -14V) While it is waiting, the keyboard is not being scanned, and any future key presses while it is waiting will not cause another “number key pressed” code to be sent. (Normally the delay to TFb and BUSYb going away would be very small, but if something is wrong and BUSYb never goes inactive, the input chip will appear to be locked up.) Pressing C will reset the chips and restart scanning.

The manual states that the CE key is not in the matrix. This was the case for the Model 816, but the 1440 CE key actually is in the matrix, and has its own microprogram.

Pinout of the Input chip:

1 Ground (0V)
2 CLRb active low global reset input
3 VDD (-14V)
4 TFb, input, active low “transmit figure” strobe from control chip, four clks wide
5 test point (NC)
6 VGG (-26V)
7 BUSYb, input, active low busy signal from program ROM chip
8 CLK main clock signal
9 Y0 keyboard row input
10 Y1 keyboard row input
11 Y2 keyboard row input
12 Y3 keyboard row input
13 Y4 keyboard row input
14 Y5 keyboard row input

15 Y6 keyboard row input
16 Y7 keyboard row input
17 X0 keyboard column output
18 X1 keyboard column output
19 X2 keyboard column output
20 X3 keyboard column output
21 KDb active low keyboard data output
22 Kb input, constant K switch signal
23 D input, D key signal, for entering number of digits after decimal pt
24 Ground (0V)

Seven-bit key codes triggered by MITS 1440 keyboard:
(Note that the LSB is always 1, and is transmitted first)

		Normal	K switch closed
Key		Key Code	Key Code
Any number key before dec pt entered		21H	Same
Decimal pt		05H	Same
Any number key after dec pt entered		61H	Same
EXC		0DH	Same
X Squared		0FH	Same
Square Root		07H	Same
CM		39H	Same
M-		09H	Same
M+		0BH	Same
MR		03H	Same
SIGN		37H	Same
CE		3BH	Same
-	(Delayed command)	1DH	5DH
+	(Delayed command)	1FH	5FH
/ (Divide key)	(Delayed command)	19H	59H
X	(Delayed command)	1BH	5BH
=	(Delayed command)	17H	57H

4-bit BCD codes are also sent LSB-first, and correspond to the number key pressed, so pressing the 4 key will send a 4, etc.

Control chip (FDY 120B-5013)

This chip contains several counters and registers used to store the position of the decimal point, and to count BCD result digits generated during multiply and divide operations. It gets a number of control inputs from the program ROM chip. These can be called the “opcode”, and cause the control chip, ALU, and register chip to perform the sequence of operations necessary to execute a specific command entered on the keyboard, such as multiply or subtract. The control chip gets the DT0b (digit time 0) and DTFb timing strobes from the ALU chip, which keep it in sync with digit positions in the register and ALU chips. When the opcode is correct for reading a numeric digit’s value from the input chip, and the desired digit position’s time comes around relative to DT0b and DTFb, the TFb output pin will go low for one digit time (four clocks), causing the input chip to shift the BCD code for that digit out on its KDb pin. The KDb data line connects to the control, ALU, ROM and output chips, but not to the register chip. The control chip probably only uses it to load the decimal point position register. The decimal point position register is used by the control chip to time the TFb signal properly so that an entered digit will be placed in the correct position in the X register. (This is explained in great detail in the patent.)

Pinout of the Control chip:

1 Ground (0V)
2 Power-on clear input (1440 interconnect schem incorrectly shows this as Gnd)
3 “Set decimal point position” active low signal to output chip pin 13 through diode D10 (D10 not shown on interconnect schem)
4 output, test point (NC)
5 Status input from ALU, meaning depends on the opcode
6 Y register serial data input from ALU
7 KDb input, serial data from input chip
8 DTFb digit time slot strobe input
9 D “D” key signal for entering # of digits after decimal pt
10 Branch control output to program ROM (overflow, end of digit count reached, sign value?, etc)
11 DT0b digit time slot strobe input
12 Opcode input #20 from program ROM
13 Opcode input #21 from program ROM
14 Opcode input #18 from program ROM
15 Opcode input #5 from program ROM
16 Opcode input #9 from program ROM
17 Opcode input #6 from program ROM
18 output, test point (NC)
19 VDD (-14V)
20 input, or maybe not bonded internally? NC
21 VGG (-26V)
22 CLK main clock signal
23 TFb output “transmit figure” pulses low to tell input chip to send BCD digit code
24 CLRb active low global reset input

ROM opcode values necessary for TFb to go active:

#18	#20	#21	#9	#6	#5	
0	0	1	1	0	0	or
1	0	1	1	0	0	

Program ROM chip (FDY 320B-7006)

This chip contains a starting address shift register, an address register, and a 256-word by 15 bits ROM (ROM size inferred from my testing). Eight of the ROM outputs can be used as inputs to load the address register with the next address. The address register can be loaded in two ways: If the BUSYb signal is inactive, it is loaded from the starting address shift register. If BUSYb is active, it is loaded from the eight ROM "next address" outputs. The starting address shift register is clocked all of the time by CLK, and has the KDb serial output from the input chip as its input. In general, this signal sits at 0V (logic 0), so the shift register contains all 0's. When a serial command is sent (LSB first) it is right-shifted into the register. All commands have the LSB set to 1, so when that bit reaches the LSB of the starting address shift register, the register contains the entire command. A 1 in the LSB of this register causes the "BUSYb" output of the ROM chip to go active. At this time the address register is loaded from the shift register, and the ROM "next address" bits are selected as the address register input from then on. So each different serial command code from the input chip corresponds to a different starting 7-bit address in the program ROM. After a particular command's program is started, the ROM itself controls the next address until the next address value is "0", at which time the BUSYb signal goes inactive and the microprogram for that command is finished. The program ROM receives an input from the control chip, which sometimes contains information about register contents and overflows in the ALU chip. The control chip contains several counters used to count digits during multiply and divide operations, etc. Information related to those counters is muxed onto this signal as well for some opcodes. This information would be used to end a microprogram when all BCD digits of a multiply or divide operation have been processed. The ROM address register is advanced to the next value at the end of DTFb, but changes before DTFb rises, so is probably clocked by CLK, gated by DTFb.

Note: The signal that drives the minus sign on the display comes from the program ROM chip, and is latched in this chip, so it can persist after the microprogram has ended and the ROM address has returned to 0.

Pinout of the Program ROM chip:

1 Ground (0V)
2 VGG (-26V)
3 VDD (-14V)
4 CLK main clock signal
5 Program ROM opcode output #5
6 Program ROM opcode output #6
7 CLRb active low global reset input
8 Ground (0V)
9 Program ROM opcode output #9
10 test point (NC)
11 DTFb digit time slot strobe input
12 KDb input, serial data from input chip
13 Ground (0V)
14 Input from control chip, tested for conditional branching in microprograms
15 Output signal, to register chip, this signal is 0V for the opcodes that affect the register chip
16 Output, test point (NC)
17 Output, test point (NC)
18 Program ROM opcode output #18
19 Output, test point (NC)

20 Program ROM opcode output #20
21 Program ROM opcode output #21
22 BUSYb output (active during microprogram sequence, used to pause keyscan and blank display)
23 SIGNb output to display circuit, -14V turns on minus sign at left of display
24 Output, test point (NC)

ALU chip (FDY 320B-7002)

The ALU chip contains three 64-bit registers. I call the data entry register, or input register, the "X" register. In addition there is an accumulator, and what I call the Y register. The X register can be added to or subtracted from the accumulator, under control of the opcodes. In addition, X, and the accumulator can each be shifted one digit to the left or right, individually. The Y register can be rotated left or right one digit at a time. The X register can also be loaded with the value currently in reg17 of the register chip. The contents of the X register is always presented on ALU pins 25 and 26, and can be used to load reg17 in the register chip. Several outputs from the ALU feed to the control chip. One of these is the Y register contents, the LSD of which can be loaded into a counter in the control chip for use as a loop counter. Either the X register or the accumulator can be selected to feed to the display.

The ALU chip also syncs up the clear key input to CLK and sends it out to the other chips. In the ALU, CLK is divided by four, then is used to clock a binary counter. Decoded "0" (DT0b) and "F" (DTFb) outputs of this counter are used to set the digit timing so that the other chips can all stay in sync. Since the counter wraps after 15, (hexadecimal F) DT0b pulses low (active) during the four clock cycles immediately following DTFb.

Pinout of the ALU chip:

1 Ground (0V)
2 Ground (0V)
3 KDb input, serial data from input chip
4 TFb input (transmit figure) from control chip
5 D "D" key signal for entering # of digits after the decimal point - this probably inhibits digit entry in the ALU when setting the decimal point position.
6 VGG (-26V)
7 CLK main clock signal
8 Opcode input #20 from program ROM
9 Opcode input #21 from program ROM
10 Opcode input #18 from program ROM
11 Opcode input #5 from program ROM
12 Opcode input #9 from program ROM
13 Opcode input #6 from program ROM
14 VDD (-14V)
15 Ground (0V)
16 "C" key raw (unsynced) input
17 Output to the output chip (the serial data on this pin appears on the display)
18 DTFb digit time slot strobe output
19 CLRb output (C key synced to CLK)
20 Output to control chip (meaning depends on opcode, passed to ROM for cond branching)
21 Output, test point (NC)
22 Overflow output to display circuit: -14V turns on "E"
23 DT0b digit time slot strobe output

24 Y register serial data to control chip
25 Tied to pin 26 of ALU chip, and to regs chip. One of these two pins is the X register serial data.
26 Tied to pin 25 of ALU chip, and to regs chip. The other pin is probably one input to the serial adder, with the accumulator register being the other one.)
27 Ground (0V)
28 Input from regs chips (This is the serial data output of the register chip,
which can be used to load the X register in the ALU for processing)

Register chip (FDY 110B-5001)

The register chip contains three 64-bit circulating registers. I call these reg17, reg19 and regint. Reg17 contents are always available on pin 17 of the register chip.   Reg19 contents are always available on pin 19 of the register chip. Regint contents are not visible directly, but it can be loaded from reg19, and reg17 can be loaded from it. I think one of these registers is used as a scratch pad register, another is the Memory register, and probably the third is used as a multiplier/quotient register. The patent explains more details of the inner workings of this chip. Data is fed from this chip to the ALU chip, and from the ALU chip back. Opcode bits from the program ROM control the operation of this chip.

Pinout of the register chip:

1 Ground (0V)
2 ?, NC
3 VGG (-26V)
4 Opcode input #20 from program ROM
5 Opcode input #21 from program ROM
6 Opcode input #18 from program ROM
7 Opcode input #5 from program ROM
8 Control signal from program ROM pin 15, all opcodes that affect reg chip also have this bit low
9 Opcode input #6 from program ROM
10 VDD (-14V)
11 Input from ALU pins 25 and 26. This is always the X register contents.
12 Ground (0V)
13 DT0b digit time slot strobe input
14 Ground (0V)
15 Output, test point (NC)
16 Output, test point (NC)
17 Output to ALU chip pin 28. This is the serial output from the register chip. The contents of reg17 always appear on this pin, but the data from the other two registers can be rotated into this register, so as to appear on this pin for loading into the X register in the ALU.
18 Output, test point (NC)
19 Memory register serial data output (doesn’t seem to be used in the 1440)
20 VDD (-14V)
21 CLK main clock signal
22 Ground (0V)
23 CLRb active low global reset input
24 TFb transmit figure input, tells input chip when to transmit BCD digit values

Output chip (FDY 150B-5005)

This chip generates digit select lines for the display, and feeds one digit at a time out on four BCD data lines. The display is multiplexed, with only one character actually being driven at any one time. Signal DT0b is used by the display circuit to count through ten digit display times. Two cycles through these ten digit times are required to update the entire display. Eight of the fourteen digits are displayed during eight counts out of the first group of ten (to refresh the lower part of the display). The upper six digits are displayed during six counts out of the second group of ten, then the sequence repeats. The signals M6b and M7b are generated by the display circuit and are fed to the output chip to toggle it between the lower part of the display and the upper part. Since DT0b is used as the clock for the display multiplexing. Each digit is displayed for an entire frame, or 16 digit time slots of the MOS chipset. A complete 20-count display cycle corresponds to 20 DT0b-DTFb chipset frames. The output chip grabs one digit from the constantly circulating serial data at its input (pin 2) and presents it on the four BCD output lines for one digit display time, then it grabs the next digit from the input stream and holds it for the next digit display time, etc.

Note that the voltage levels on the outputs of this chip look a little odd, as they are driving TTL chips running off of a 5V supply.

Pinout of output chip:

1 Ground (0V)
2 Input from ALU chip (can be either the X or the accumulator register contents, constantly shifted onto this pin) This is the data which is displayed.
3 Ground (0V)
4 B1 BCD digit output LS bit, to display
5 B2 BCD digit output bit, to display
6 B4 BCD digit output bit, to display
7 B8 BCD digit output MS bit, to display
8 VDD (-14V)
9 M6b timing input from display circuit
10 CLRb active low global reset input
11 M7b timing input from display circuit
12 KDb input, keycode/BCD digit value from input chip
13 Input, active low, driven by both power-up clear (probably to clear the display decimal pt pos at power up) and the control chip. The control chip generates a low pulse on this pin at the same time as TFb when the D key is held to set the decimal point position. This pulse enables shifting the four-bit BCD number code on the KDb line into a register that controls the position of the decimal point on the display, so if the value on KDb is 2, there will be two digits to the right of the decimal point. (Pressing the C key does not affect the decimal pt position setting.)
14 P5 digit select output, to display
15 PD decimal pt select output, to display
16 P6 digit select output, to display
17 P3 digit select output, to display
18 P4 digit select output, to display
19 P0 digit select output, to display
20 P2 digit select output, to display
21 P7 digit select output, to display
22 P1 digit select output, to display
23 VGG (-26V)

24 CLK main clock signal

Possible sources:

As of 1/2009, it seems that the FDY 320B-7002 ALU may still be available at www.aattech.com, and maybe elsewhere. Also, the FDY 310B-7014 input chip and FDY 150B-5005 output chip are available at www.heldt-electronic.de.

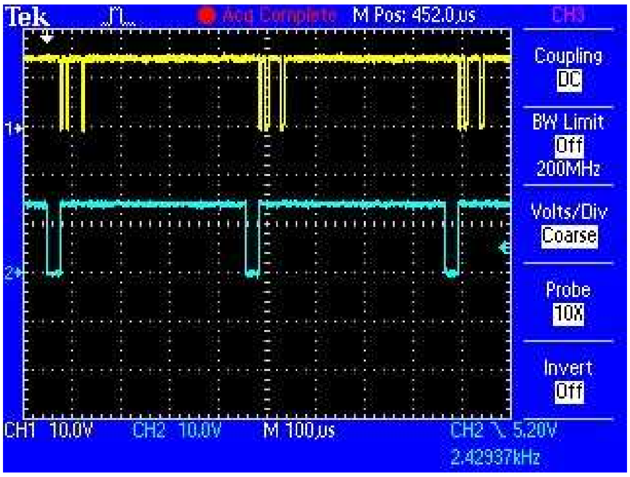
Troubleshooting info:

The 1440 that I am working on had two bad 5V zener diodes. One is used for the 7447 7-segment decoder and the other for the main -5V supply. They both had the same part number, and both had much lower zener voltages than expected (around 3.0 volts). I replaced them with 5.1V 1W zener diodes, which seem to be acceptable replacements.

The 7490 chip in my unit's display alternating circuit had a very low resistance to ground on the pin 7 input (47 ohms). That was preventing the counter from getting reset when the display was inhibited. I replaced the chip.

Some Waveforms

Photo 1:



Tek Acq Complete M Pos: 452.0µs CH3

Coupling DC

BW Limit Off 200MHz

Volts/Div Coarse

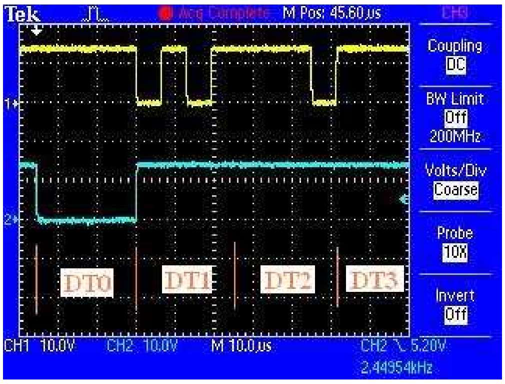
Probe 10X

Invert Off

CH1 10.0V CH2 10.0V M 100µs CH2 \ 5.20V 2.42937kHz

Yellow trace is repeating display data at pin 2 of 5005 chip. Light blue trace shows DT0b repeating timing strobe, at pin 13 of 5001 chip. Picture was taken after pressing C 8 5. So display shows 85 in lowest two positions.

Photo 2:



Tek Acq Complete M Pos: 45.60µs CH3

Coupling DC

BW Limit Off 200MHz

Volts/Div Coarse

Probe 10X

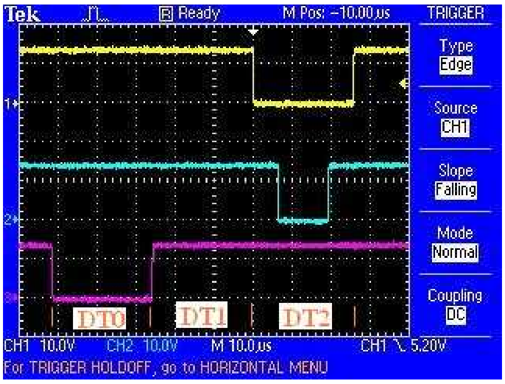
Invert Off

CH1 10.0V CH2 10.0V M 10.0µs CH2 ↘ 5.20V

2.44954kHz

Same signals and conditions as Photo 1, but zoomed in to show data more clearly. Note that display data is right-shifted one digit time from normal, so a digit appearing in digit time 1 would appear in digit 0 of the display. Each digit time contains four clk cycles, one per bit. Digit data is shifted LSB-first, and a low level is logic 1. Picture was taken after pressing C 8 5. So display shows 85 in lowest two positions.

Photo 3:



Tek Ready M Pos: –10.00µs TRIGGER

Type Edge

Source CH1

Slope Falling

Mode Normal

Coupling DC

CH1 10.0V CH2 10.0V M 10.0µs CH1 ↘ 5.20V

For TRIGGER HOLDOFF, go to HORIZONTAL MENU

Photo shows a digit being transferred from the 7014 chip into the digit 2 position of the input register in the ALU. The bottom trace is the DT0 repeating time strobe. The top trace is the TFb "transmit figure" strobe at pin 23 of the control chip, which fires in response to an opcode from the ROM chip. It is firing during digit time 2, to load the digit at that position in the register. The digit being loaded is a 6. The middle trace is the KDb key data signal from pin 21 of the input chip. Four bits of data are shifted during the TFb pulse, LSB first, and a low level is logic 1.

Differences between the 816 and the 1440:

The 816 used four chips: The 5005 for output (same as the 1440) 5004 or 7014 for input, 5002 combined ROM/control and 7010 combined ALU/registers chip. The input and output sections presumably work similarly to the 1440, although the 816 used an 8-digit display, and a key to toggle between the lower and upper 8 digits of a result.   The 816 CE key is not in the matrix, but is fed directly to the input chip, whereas it is in the matrix in the 1440 design, and has a microprogram associated with it. Presumably the ROM/control is similar to the ROM and control chips used in the 1440, although the 1440 ROM may have 256 locations as it has Sqrt (X) and X squared in addition to the four basic math functions.