VIRTUAL 2315 CARTRIDGE FACILITY

Virtual 2315 Cartridge Facility

TABLE OF CONTENTS

Virtual 2315 Cartridge Facility

Overview

The Virtual 2315 Cartridge Facility (V2315CF) is an enhancement designed to add onto an IBM 1130 System that makes use of files as the contents of the disk cartridge, rather than having the disk drive read and write the physical cartridge.

It affords the sensations an operator of the system would experience if using the disk drive in its normal mode; the whir of the cartridge spinning, the grunt of the arm moving the disk heads, the vibration of the motors, even the process of sliding a cartridge into the drive slot.

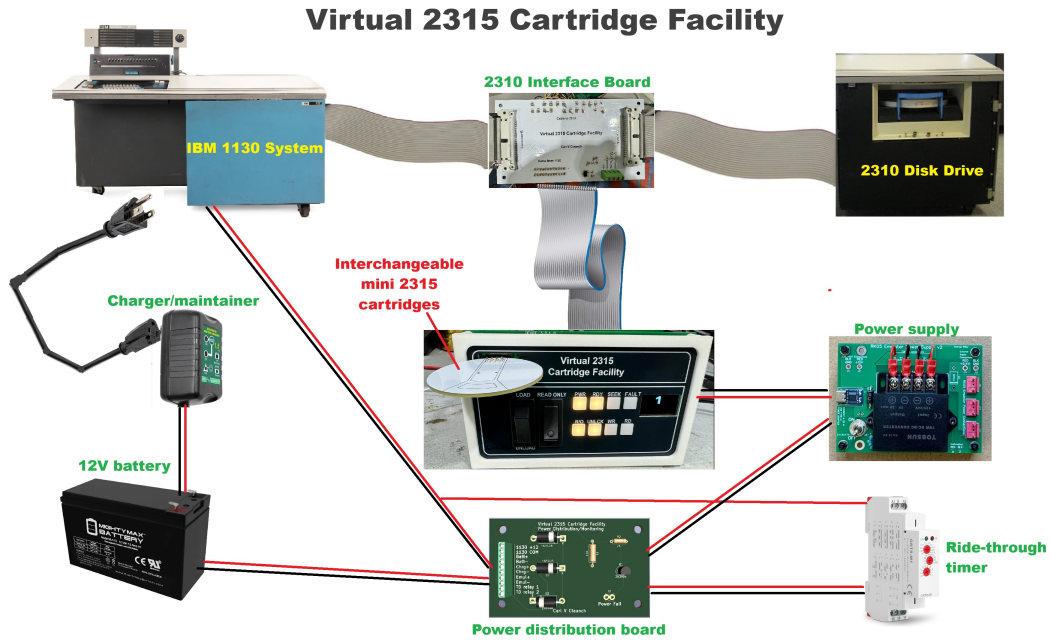
One big benefit of the V2315CF is protection of the extremely rare and sensitive disk heads in the disk drive. The heads are not loaded down onto a platter inside a 2315 cartridge, thus they are safe from head crashes and other damage.

The disk image files are similar to those used by the various IBM 1130 Simulator programs, requiring only a quick conversion utility program to switch between the V2315CF and simulator formats. Thus, many cartridge images can be created and manipulated on personal computers, plus images that were archived from physical cartridges can be used with the V2315CF.

A secondary mode of operation is available - virtual - for use when the disk drive in the 1130 is not working. The 1130 system still sees a disk drive and can read and write to the cartridge images. All that differs in virtual mode is the lack of physical sensations from a drive’s operation.

Virtual 2315 Cartridge Facility

Block Diagram of the V2315CF



The central box in the diagram is the main unit of the V2315CF. Mini 2315 cartridges are plugged into the slot on the upper left.

The 2310 Interface Board routes signals between the IBM 1130 System and its internal disk drive. It also contains the Real/Virtual mode switch and circuitry to monitor and control the Unlock lamp on the IBM 1130 console.

The power system of the V2315CF has a battery which is intelligently maintained at full charge whenever the IBM 1130 System is physically connected to building power, even if switched off.

The IBM 1130 system produces 12V power which is connected to the power supply of the V2315CF. The 1130 supply generally powers the main unit and the 2310 Interface Board.

Virtual 2315 Cartridge Facility

The battery and 1130 system power are both connected to a Power Distribution Board, which in turn passes the power to the 2310 Interface Board and V2315CF main unit.

When the IBM 1130 system power fails, it is detected and a signal sent to the V2315CF which does an unload of the disk image back to the Mini 2315 Cartridge.

A timer keeps the battery connected to the main unit for about 45 seconds, long enough for the unload, then drops power until the 1130 system power returns.

Virtual 2315 Cartridge Facility

IBM 1130 and its disk drive

The IBM 1130 System, first shipping in 1965, had a disk drive built into the machine. When the front cover of the 1130 was opened, the operator could pull on a handle to open a slot, into which a 2315 disk cartridge is inserted.

These removable cartridges each held 512K words of storage (1MB), organized in 203 cylinders. On each cylinder, four sectors of 321 words were available. As the drive had two heads, one for each side of the disk platter inside the cartridge, each cylinder held 8 sectors of data.

IBM had developed disk drives in the 1950s and early 1960s based around a 14” disk platter, coated with magnetic oxide. Their S/360 line of mainframes, also first delivered in 1965, used stacks of multiple 14” platters as disk packs. However, the smaller IBM 1130 used a single platter in a cartridge - the 2315. The disk drive that used the 2315 cartridges was internally named the 13SD. Within the IBM 1130 it is generally just referred to as the ‘internal drive’.
It was also available in a standalone box as the 2310; the box could hold 1 or 2 of the 13SD drives. The 2310 could also be attached to other systems such as the IBM 1800, a cousin of the 1130.

I will use 13SD, 2310 and internal drive interchangeably for the disk drive inside the IBM 1130 cabinet in all the documents and designs related to the V2315CF.

Virtual 2315 Cartridge Facility

Mini 2315 Cartridge is an interchangeable file holder

The disk image used by the V2315CF is held on a microSD card. That is read in and loaded into RAM inside the V2315CF when the switch on the main unit is set to Load.

All reads and writes by the IBM 1130 take place with the RAM. Thus, after a session is is likely that many sectors were written with changed or new content. These must get back onto the slower microSD card at the end of a session.

Thus, when the main unit is set to Unload, the data from RAM is fetched and the disk image on the microSD card is rewritten. Changes made by the IBM 1130 dring the session are preserved and seen when the cartridge is next loaded.

Optionally, by toggling the Read-Only switch, one can skip the rewrite of the image when unloading. This allows packs to be kept in their original unaltered form, with all updates during a session discarded at the end.

A microSD card is a small object, one that is also somewhat fragile. Thus, I created a Mini 2315 Cartridge as a 2” diameter replica of a 2315, containing a PCB inside with a microSD card plugged into its connector.

It is envisioned that users of the V2315CF will keep a number of 2315 cartridges, each one in a Mini 2315 Cartridge, interchanging them as desired. These plug into a connector on a slot in the upper left of the main unit.

Virtual 2315 Cartridge Facility

Using the V2315CF with the IBM 1130

Real mode operation

Real mode is a hybrid with the internal disk drive of the IBM 1130 operating in conjunction with the V2315CF. It is the normal method of operation.

The internal disk drive of the IBM 1130 should have a physical 2315 cartridge inserted and the blue slot handle closed. The cartridge must be an eight sector type such as used with the 1130 but the contents don't matter.

The V2315CF main unit has its Load/Unload switch set to Unload. A Mini 2315 Cartridge is plugged into the upper left connectof of the V2315CF.

Power up the IBM 1130 system, if it is not already running. Ensure that the internal disk drive of the IBM 1130 has its motor switch (below the blue slot handle) turned off.

Turn the Load/Unload switch on the V2315CF to Load. The OLED display screen should show the progress reading the virtual 2315 cartridge contents and finish with a large digit 1 on the screen.

Turn the internal disk drive motor switch on (up position). You will hear the disk cartridge begin to spin and after about 90 seconds the drive should reach its ready condition.

Virtual 2315 Cartridge Facility

You will see the RDY lamp on the V2315CF light, in addition to illuminating the green File Ready lamp on the console of the IBM 1130, to the left of the keyboard. This indicates that the system is available for software to make use of the disk drive.

Use the disk drive normally, as if it were just the internal disk drive in its unaltered state. Software can issue seek commands to move the disk arm in and out among the 203 cylinders (radial positions) on the platter. It can read or write individual sectors.

The internal disk drive will move its disk arm to match the seek commands issued by software, providing a grunting sound and some vibrations as the arm moves. The SEEK lamp on the V2315CF will flash briefly for each seek that is performed.

The V2315CF is also moving the cylinder of the virtual 2315 cartridge that was loaded, so that any read or write will occur at the current cylinder location.

When the software reads a sector into core memory on the IBM 1130, the RD lamp will flash on the V2315CF. When software writes a sector from core memory, the WR lamp flashes. The data is moving between core memory and the V2315CF main unit, not the internal disk drive.

Running the DMS2 operating software of the IBM 1130 is done the same way as if the internal disk drive was unaltered. You load a cartridge that has DMS2 on it - in this case, a Mini 2315 Cartridge which has DMS2 loaded. The V2315CF loads the image and the disk drive lights the File Ready and RDY lamps.
A boot card is placed in the primary input device, usually a 1442

or 2501 card reader, and the Prog Load button on the IBM 1130 console is pressed. The card is read into memory and that card then reads DMS2 from the disk drive and begins execution.

When it would be time to stop using the disk, or to change cartridges, turn off the internal disk drive motor switch (down). You will see the File Ready and RDY lamps go out, hear the spinning cartridge slow down, and eventually the Unlock lamp on the IBM 1130 console, near the File Ready lamp, will illuminate. In addition, the UNLK lamp on the V2315CF turns on.

At this point, by turning the Load/Unload switch on the V2315CF to Unload, the contents of the cartridge as it has been changed during your session will be written back to the Mini 2315 Cartridge.

The OLED screen on the V2315CF will display the cylinders being written back and then end with the idle screen image. The Mini 2315 Cartridge can be removed from the connector of the V2315CF.

As an alternative to updating the Mini 2315 Cartridge with the changes made to the cartridge during a session, the V2315CF could be put into read-only mode. This is done by toggling the Read Only switch to set the RO lamp on.

If RO is lit when the Load/Unload switch is turned to Unload, the V2315CF will skip writing back the changes to the Mini 2315 Cartridge. The OLED goes right to the idle screen.
Thus the contents of that cartridge are the same as when it was first loaded and any changes made to the cartridge during your

session are lost.

This is useful if one wants a training or student version of DMS to be available to multiple people in the state it was prepared.

Each student or museum visitor can use DMS2, compiling and storing programs, updating files and experimenting, seeing those changes during their session, yet the next visitor or student will have the same initial setup because the Mini 2315 Cartridge did not change.

If a new cartridge is desired during any session, after having run using one cartridge and unloaded it, plug a different Mini 2315 Cartridge into the V2315CF connector and flip the Load/Unload switch back to Load. The new cartridge will load. The internal disk drive motor switch should be turned on, then after about 90 seconds the File Ready and RDY lamps light to indicate that the new cartridge is ready for access.

Virtual 2315 Cartridge Facility

Virtual mode

Virtual mode is an alternative that does not make use of the internal disk drive. That disk drive does not have its motor switched on and the disk arm will not move in concert with the V2315CF.

The V2315CF main unit has its Load/Unload switch set to Unload. A Mini 2315 Cartridge is plugged into the upper left connector of the V2315CF.

Power up the IBM 1130 system, if it is not already running. Leave the motor switch (below the blue slot handle) of the internal disk drive off.

Turn the Load/Unload switch on the V2315CF to Load. The OLED display screen should show the progress reading the virtual 2315 cartridge contents and finish with a large digit 1 on the screen.

After 90 seconds, you will see the RDY lamp on the V2315CF light, in addition to illuminating the green File Ready lamp on the console of the IBM 1130, to the left of the keyboard. This indicates that the system is available for software to make use of the disk drive.

Use the disk drive normally, as if it were just the internal disk drive in its unaltered state and the same way you would use V2315CF in Real mode.

When ready to end a session and stop using the disk, simply flip the Load/Unload switch of the V2315CF to Unload. It will then turn off the File Ready and RDY lamps, turn on the UNLK lamp, and rewrite the changed contents back to the Mini 2315 Cartridge. The OLED will return to its idle screen.

If the RO lamp is on, then the changed contents are not written back, in the same way that read-only mode works with in Real mode.
Use of the IBM 1130 disk drive with the V2315CF in Virtual mode is the same as with Real mode, except you will not hear a cartridge spinning, hear grunts of the arm moving, nor feel the vibrations of the drive. The software uses the disk with exactly the same performance as in Real mode.

Changing between Real and Virtual mode

A small switch on the 2310 Interface Board, located inside the right side cover of the IBM 1130, is moved between Real (left) and Virtual (right) positions to control the mode. It should be moved while the 1130 is powered off.

Virtual 2315 Cartridge Facility

V2315CF is a derivative of an RK-05 Emulator

George Wiley created the RK-05 Emulator, which emulates DEC disk drives and serves up cartridge images from microSD cards. Because the RK-05 is derived from the IBM 13SD disk drive, under license, the emulator was a good starting point to develop the V2315CF.

However, very substantial differences exist between the 13SD and RK-05, the disk formats of the cartridges, and the disk controller electronics in the computing systems. The RK-05 has the operator directly handle and insert the microSD cards for each RK-05 cartridge image.

The RK-05 Emulator has a Raspberry Pi PICO processor, an FPGA, SDRAM, circuits to interface with the electronics of a DEC controller, controls, LEDs and other components.

Interface cards provide twin ribbon cables from the RK-05 emulator to the disk controller in the DEC system. A terminator board ensures that the signal lines are properly terminated to ensure good signal quality.

Fortunately, the number of input and output signal lines for the RK-05 Emulator are sufficient to meet the needs of a 13SD connected to an IBM 1130. I therefore leveraged the main box circuitry but reprogrammed both the FPGA and PICO to my needs.

I substituted a new faceplate on the main box and modified the method of connected microSD cards. The RK-05 Emulator has a

small PCB with the connector for the microSD card sitting in a slot, with a 2x5 ribbon cable connecting that PCB to the remainder of the emulator.

I relocated the ribbon cable end to the front of the slot, removing the PCB entirely. My Mini 2315 Cartridge objects have a 2x5 pin connector on the back edge which plugs into the ribbon cable connector.

I retained the power supply, interface cable board and terminator unmodified. I changed the meaning of some of the LEDs on the main unit, reflected in the labels on my replacement faceplate.

Signaling levels used with the V2315CF

The IBM 1130 Solid Logic Technology (SLT) generally uses +3V as logic high and 0V as logic low. SLT is quite tolerant of higher voltages for logic high and even negative voltages for logic low. The 1130 also uses +48V and 0V for a few signals.

The PICO processor in the V2315CF requires that its IO pins be limited to +3.3V for logic high and no less than 0 for logic low.

The FPGA inside the V2315CF is also operated with +3.3V and 0V IO pins.

Because the DEC controller logic and RK-05 disk drives operated with +5V logic high and 0V low, the V2315CF contains voltage translation circuitry such that all signals on the ribbon cables running to the 1130 controller circuity and to the 13SD disk drive operate at those levels.

The 2310 Interface Board implements some sensing and control circuits connected to the main V2315CF unit, thus these operate at +5 and 0V levels.

The Unlock signal from the 13SD drive and the Unlock lamp on the IBM 1130 console operate at +48 and 0V.

The power fail sense circuit on the Power Distribution Board senses the +12V supply but outputs with an open collector gate thus is compatible with 3.3 and 0V levels of the PICO.

Power System of the V2315CF

Normal power for the V2315CF is supplied by the 12V power rail of the IBM 1130 system, when the 1130 is powered on. This is fed through the Power Distribution Board to the V2315CF Power Supply (PS) unit. That power supply unit converts the input to 5V power which drives the V2315CF main unit. Internally, the main unit converts the 5V down to 3.3V for use with the FPGA, PICO, microSD card reader, OLED and other components.

The 12V from the Power Distribution Board (PDB) is also connected to the 2310 Interface Board where it powers a few circuits on that board.

A 12V battery in the system is also connected to the V2315CF, so that it can power the main unit when power drops from the IBM 1130. The design of the IBM 1130 causes the 12V to be rapidly cut off when the 1130 shuts down.

A circuit on the PDU detects the loss of the 12V supply from the IBM 1130 and produces a signal to the V2315CF that power was lost. The V2315CF will do an immediate unload of any cartridge that was being used, writing its changed contents back to the Mini 2315 Cartridge.

A ride-through timer is connected between the PDU and the V2315CF PS. It senses the 12V coming from the IBM 1130 power supply, but switches the output of the PDU, which is a combination of the 1130 supply and the battery.

The timer is configured to connect the PDU to the PS whenever it sees 12V from the IBM 1130, and to maintain the connection for about 40 seconds after the IBM 1130 12V supply drops out. This ensures adequate time for the V2315CF to perform the emergency unload to the Mini 2315 Cartridge.

Because of the ride-through timer, the battery is not powering the V2315CF for more then 40 seconds when power to the IBM 1130 is turned off. A battery charger/maintainer is connected to the battery and will keep it fully charged.

The charger is plugged into the IBM 1130 utility outlet that always has power as long as the main power cord of the IBM 1130 is hooked to building power supply, whether or not the IBM 1130 is switched on.

Diodes on the PDU isolate the IBM 1130 12V and the battery from each other. The voltage of the IBM 1130 12V power rail is typically several volts higher, thus it provides the majority of the power to run the V2315CF even though the battery also feeds some power through its diode.

Virtual 2315 Cartridge Facility

Utility programs (python) for handling cartridges

Disk image files used by the simh based IBM 1130 simulator programs are not in the correct format to use with the V2315CF. They need to be converted using a utility program that runs under Python on a Windows PC.

convert1130.py is a program that asks you to select the simulator format disk file, then select a location and name for the V2315CF format disk file. It will ask you to provide a cartridge identifier and a text description of what this cartridge is for.

convert2315.py is a program that will take a V2315CF format disk file and convert it to the format needed to run it on an IBM 1130 simulator. You choose the V2315CF format file and then select a location and name for the simulator format output. It lists the cartridge ID and other information from the V2315CF format file header.

listcartridges.py is a program that asks you to select a folder/directory, then lists all valid V2315CF format disk files located in that folder. It provides the file name, the cartridge identifier and the description from the file header.

showsector.py will list a sector from a V2315CF cartridge image file in hex words. You are asked to select the V2315CF format file, then enter a cylinder, head and sector number to be listed.

Theory of Operation

The heart of the V2315CF is a PCB that contains an FPGA, SDRAM and a Raspberry Pi Pico. The FPGA interfaces with the disk drive and IBM 1130 disk controller electronics. The Pico provides the user interface and manages the microSD card that holds the virtual disk image.

The FPGA and the Pico communicate over a Serial Protocol Interface (SPI), exchanging status and commands. The SDRAM is directly accessed by the FPGA.

At startup, the code in the Pico initializes the general purpose IO pins on the Pico, starts the SPI link with the FPGA, and initializes the FPGA. It also controls a small OLED screen that displays some status messages during operation.

When the operator inserts a Mini 2315 Cartridge, they are connecting a PCB holding a microSD card. SPI communications are used by the Pico to read and write to the microSD card, when the operator loads or unloads a virtual cartridge.

Loading a virtual cartridge begins when the Load/Unload switch on the V2315CF is turned to the Load position. The Pico opens the microSD card and looks for a file with the suffix .dsk which will be the disk image. The file is checked to be certain it is the correct format, then read by the Pico.

The Pico writes the data from the microSD card file through the SPI link to the FPGA. The FPGA writes the data into SDRAM

so that it is ready for disk operations by the IBM 1130.

The Pico informs the FPGA that a cartridge image was loaded into the SDRAM. The Pico then waits for the disk drive to become ready.

In real mode, the disk drive in the IBM 1130 will assert the File Ready signal line after it has been spinning for 90 seconds and believes all is ready for use. Thus the operator must switch on the disk drive in the 1130 to spin the cartridge. The V2315CF raises the File Ready signal to the 1130 which causes the File Ready lamp on the 1130 console to light up.

In virtual mode, the V2315CF begins to immediately spin up the virtual disk drive. After a wait of 90 seconds File Ready is asserted to the IBM 1130 causing the File Ready lamp to light.

The Pico periodically checks to see that the disk is still ready for access - asking the FPGA over the SPI link for a status word, one of whose bits means the disk is ready.

If the bit turns off, such as when the operator switches off the disk drive in the 1130, the Pico turns off the RDY light and gets ready to retrieve the updated contents of SDRAM so that the file on the microSD card can be kept current.

When the 1130's disk drive is not turning, the Unlock lamp on the 1130 console is illuminated, as well as the UNLCK lamp on the V2315CF. When the motor starts spinning, the lamp goes off. In virtual mode, the lamp goes off as soon as we load a cartridge image and turns back on when the cartridge is unloaded.

Virtual 2315 Cartridge Facility

While the disk is ready, the V2315CF passes commands between the 1130 and its disk drive, passes status back from the drive to the 1130, except for the data being read or written.

Thus, when a program issues a XIO Control to seek the disk arm some number of cylinders forward or backward, V2315CF passes that to the 1130 drive which physically performs the arm movement. The V2315CF is also monitoring the commands so that it knows the current cylinder of the disk drive.

The physical 2315 cartridge in the 1130's disk drive has notches cut into a ring on the bottom of the platter, which cause sector and index marker pulses to be sent from the drive to the V2315CF. In real mode, these are passed on to the 1130. In virtual mode, we simulate a spinning platter and generate the sector and index pulses to send to the 1130.

When a program executes an XIO Initiate Read, it specifies a sector number (0 to 3) and a head (top or bottom) where we will read a sector. The buffer area in core memory has a count in the first word that tells the 1130 how many words to read from the sector. A sector can contain up to 321 words.

When the 1130 sees the sector pulse that marks the start of the targetted sector, it raises a Read Gate that normally would cause the disk drive to begin streaming the data from the disk head to the 1130.

The disk drive of the 1130 will take the serial stream of pulses and separate them into clock and data pulses. Every bit on the

disk drive spans two pulse intervals. The first always has a pulse called the clock. The second interval only has a pulse if the bit value is 1, otherwise there is no pulse.

The 1130 drive separates the stream so that the clock pulses come on the Read Clock wire and the data pulses (for bit values of 1) come on the Read Data wire. Whether we are in real or virtual mode, we ignore the Read Clock and Read Data lines from the drive.

That is because we mechanically altered the drive so the heads do NOT ride down on the platter surface inside the cartridge. They remain safely above the disk surface and thus safe from disk crashes.

The V2315CF starts timing from the sector pulse when the Read Gate is turned on. We produce the stream of bits that would come from the disk head, sending them as Read Clock and Read Data to the 1130. This includes the special bits - a preamble of zero bits, a sync word, and four check bits at the end of every data word in the sector. V2315CF generates the appropriate check bits based on the word we fetched from SDRAM.

The 1130 takes this stream from the V2315CF and handles them the same as when the 1130 disk is working as it was designed. The words from the sector are assembled, error checking is done on the four extra bits, and the data word is stored in core memory in the buffer.

When a program issues an XIO Initiate Write, it specifies a head and a sector number. The buffer for the write has a word count

in the first word and then the data to be written to disk in the target sector.

The disk drive produces a clock that drives the writing of pulses onto the disk surface. The 1130 sets up a Write Clock and Data signal line to indicate whether to write a pulse or not at the next clock transition. The V2315CF passes the drive's clock to the 1130 in real mode; in virtual mode we generate the clock in the V2315CF and send it to the 1130.

The 1130 is always requesting a pulse for the clock interval but uses the data bit value to request a pulse or not in the data interval. Thus writing involves a single serial stream from the 1130 to the drive.

The 1130 produces the string of zeroes (preamble), the sync word pattern, then sends each word in turn while appending four check bits after the word. The V2315CF captures the value of the Write Clock and Data line at the time of each oscillator transistion, breaks it into clock and data intervals, the assembles the words being written.

The V2315CF checks the four error bits but discards them otherwise. We write each word as it is received into the SDRAM, overwriting the previous contents of the sector.

In real mode, when the operator switches off the disk drive in the 1130, the drive drops File Ready and the FPGA detects this. The Pico asks for status periodically, sees the disk is not ready, and thus prepares to unload the cartridge.

Virtual 2315 Cartridge Facility

Once the physical drive has come to a complete stop, the drive turns on the Unlock line and we light the Unlock lamp on the 1130 console and UNLCK light on the V2315CF.

After this point, when the operator switches the Load/Unload switch of the V2315CF to Unload, the contents of SDRAM are requested by the Pico, over SPI, to the FPGA which reads them and sends them over the link.

The Pico will rewrite each sector of data in the file on the microSD card, thus upon completion of the unload our file contains all changes made by the 1130 during use of the disk drive.

In virtual mode, the disk drive is not turning. The operator flips the Load/Unload switch to Unload to cause the V2315CF to drop File Ready. The Pico will request the contents of SDRAM from the FPGA over SPI, just as in real mode, then rewrite the virtual cartridge file on microSD card.

The FPGA will light the SEEK, RD or WR lamps on the V2315CF for about a quarter of a second each time the 1130 does a seek, sector read or sector write. This is long enough for human perception, although a sector read or write is complete in less than 10 milliseconds.

Thus these lamps gives the operator a sense of the disk activity. In real mode, the operator will also hear the grunting sound of the 1130 disk drive moving the arm back and forth.

The V2315CF has a Read Only momentary toggle switch which is used to set or clear Read Only mode. The RO lamp on the

Virtual 2315 Cartridge Facility

V2315CF lights when in Read Only mode.

Read Only will skip fetching data from SDRAM and rewriting the cartridge image file when we are unloading a cartridge. This has the effect of reverting the image on the Mini 2315 Cartridge to the same state it was in when we started, regardless of what was written to or changed during our session using the 1130.

During a session, even with RO lit, every XIO Initiate Write changes the sector being targeted and later XIO Initiate Read will pull in the altered data from the sector. Compiling programs, updating libraries and accessing data on the disk works completely normally while the disk is operating.

The cartridge image is rolled back, in effect, to the contents it had when the cartridge was loaded. This is handy to support museum visitors running work that might impact other visitors or museum demonstrations - their session is lost when the cartridge is unloaded in Read Only mode.

IBM 1130 systems provide multiple voltage rails, among them 12V and 48V. These latter two are only connected via a relay when the logic voltages such as +3, +6 and -3V are stable. When the power switch of the 1130 is turned off, the relay drops out thus rapidly shutting down the 12V supply.
The V2315CF Power Distribution Board is fed 12V from the IBM 1130, which is the primary source of power for operation. In addition, a 12V battery is connected to the Power Distribution Board so that the V2315CF has time to safely fetch SDRAM contents and rewrite a cartridge file if one was loaded at the time of power loss.

Virtual 2315 Cartridge Facility

A monitoring circuit on the Power Distribution Board notifies the Pico when the 12V from the 1130 has dropped. The Pico continues to run, being powered by the 12V battery, while the code does an emergency unload of any cartridge image that was active.

A ride-through time begins counting when the 1130's 12V supply drops. It keeps the power from the Power Distribution Board connected to the V2315CF Power Supply module for about 40 seconds, time enough for the emergency unload.

The battery is connected to an intelligent charger/maintainer that will keep it topped up. This charger is fed 120V from the utility outlet inside the 1130. As long as the main power cord of the 1130 is connected to building power, the charger works even if the 1130 is switched off.

The V2315CF Power Supply module converts the 12V input from the Power Distribution Board into 5V that is fed to the main V2315CF box.

The 12V from the Power Distribution Board that is switched by the ride-through timer is also hooked to the 2310 Interface Board, powering its circuitry. The interface board receives power while the 1130 is up and for about 40 seconds after power from the 1130 ends.

Each Mini 2315 Cartridge has a PCB with a microSD card connector and a few other components. The microSD card with the virtual disk image is plugged into the connector inside the Mini

2315 Cartridge.

Pins on the back edge of the Mini 2315 Cartridge mate with a female connector on the V2315CF main box, connecting it to the SPI link which the Pico uses to read and write the file on the card.

It is expected that an owner will have multiple Mini 2315 Cartridge units, much as they would have had multiple physical 2315 cartridges when using the disk drive as originally designed.

The physical 2315 cartridge that is inserted into the 1130's disk drive is only used to produce the sector and index marker pulses as well as provide the vibration and noise typical of the drive.

Thus a single physical cartridge remains in the drive but many different Mini 2315 Cartridges can be swapped and used with the 1130.

Disk images can be created or updated on the IBM 1130 simulators, such as the Brian Knittel's from ibm1130.org or Claudio Vincenzi's simulator.

Utility programs with the V2315CF will convert between the format of the disk files used on the simulator and the format on the microSD card that sits in a Mini 2315 Cartridge.

Another utility programs list all the V2315CF format disk image files in some selected folder, also listing the cartridge identifier and a description alongside the file name.

Virtual 2315 Cartridge Facility

The selection of real or virtual mode for the V2315CF is done by changing a tiny switch on the 2310 Interface Board. Virtual mode will operate just as real mode does, but the 1130's disk drive is not spinning or making sounds.

The SPI link exchanges 16 bit words between FPGA and Pico. The Pico is the master and always initiates each message. The first half of the word is a register address and the second half is used to transfer data.

Register 0x00 sets or clears cartridge ready status in the FPGA.

Register 0xA0 fetches that status plus other conditions from FPGA to Pico - file ready, real/virtual mode, unlocked, read-only, and fault latch.

Register 0x04 toggles the read-only state. Sent by the Pico when the Read Only toggle switch is momentarily activated.

Register 0x05 sends 1/3 of the SDRAM memory address in each message, thus three in a row are needed to select an address

Register 0x06 writes a word to SDRAM at the current address. Each message sends 1/2 of a word, so a pair of messages are needed for each word, after which the SDRAM address is auto-incremented.

Register 0x88 fetches a word from SDRAM at the then current address. Two messages are needed for each word, after the second the address is autoincremented.

Virtual 2315 Cartridge Facility

Messages 0x05, 0x08 and 0x88 are the way the Pico will load a virtual cartridge image into SDRAM and fetch back the possibly changed contents to rewrite the image file during unload.

Register 0x81 returns the current cylinder number from the FPGA to the Pico.

Register 0x82 returns the current sector number and head, the ready condition of the disk, and an ID indicating whether we are currently executing a seek, read or write.

Register 0x90 and 0x91 return the version number of the code running in the FPGA, so the Pico can verify that it is talking to a compatible version.

The Pico drives a serial link over the debug port that emits many diagnostic messages during startup and operation. It operates at 460,800 baud.

The FPGA is programmed in Verilog. George Wiley recommended doing all simulation on Xilinx Vivado while synthesis is performed on Lattice IceCube2. The bitstream is loaded to the flash rom over a JTAG connector on the board.

The Pico is programmed in C++. I set up the Raspberry Pi Pico toolchain on a Raspberry Pi 4B. The executable is loaded using the Pico as a USB drive then copying the file onto it.

Licensing

General intent for use and modification

This project was developed to be shared with others working with IBM 1130 systems. Those who wish to evolve this to support other vintage computing systems are encouraged to do so, subject only to the restrictions below.

Hardware designs and modifications by Carl V Claunch

These can be used freely and adapted for any purpose, subject only to the rights held by designers of any portions that stem from George Wiley's RK-05 Emulator hardware or components therein.

Verilog and C++ code and modifications by Carl V Claunch

Python utility programs written by Carl V Claunch

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Pico code by George Wiley

// This software and related modules included by the top level module and files used to build //

// the software are provided on an as-is basis. No warrantees or guarantees are provided or //

// implied. Users of the RK05 Emulator or RK05 Tester shall not hold the developers of this //

// software, firmware, hardware, or related documentation liable for any damages caused by //

// any type of malfunction of the product including malfunctions caused by defects in the design //

// or operation of the software, firmware, hardware or use of related documentation or any //

// combination thereof. //

FPGA code by George Wiley

// This FPGA firmware and related modules included by the top level module and files used to //
// build the software are provided on an as-is basis. No warrantees or guarantees are provided //
// or implied. Users of the RK05 Emulator or RK05 Tester shall not hold the developers of this //
// software, firmware, hardware, or related documentation liable for any damages caused by //
// any type of malfunction of the product including malfunctions caused by defects in the design //
// or operation of the software, firmware, hardware or use of related documentation or any //
// combination thereof. //

Other code included in project

LIBRARY: SSD1306.C and SSD1306.H
MIT License

Copyright (c) 2021 David Schramm

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LIBRARY: SD_FAT_SPI

/* hw_config.c
Copyright 2021 Carl John Kugler III

Licensed under the Apache License, Version 2.0 (the License); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

LIBRARY: no-OS-FatFS-SD-SPI-RPi-Pico

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.