IBM 1620 Jr.
Project Status
Report #3 (9/30/2017)

This is the third status report on the IBM 1620 Jr. project. An overview of the project is detailed in *IBM 1620 Jr. Project Description, Version 1.2 (8/31/2017)*. The project is structured as several sub-projects, each with a set of volunteers working on it. Note that this is only a summary of the work done to date. Details are available in email & data files.

General (Team)

July was a very busy month getting ready for VCF West XII on August 5th and 6th. It was a wonderful opportunity for the IBM 1620 Jr. team to share their great work to-date. The judges awarded the project First Place in the Mainframes and Minis category. The team plans to exhibit again in 2018 with more of the system operational. A separate report, including videos, details the event.

Work on the project slowed in August and September as team members were busy with jobs, family, and other commitments. This is a project being done by part-time volunteers, many of which have full-time jobs.

A new document, *Re-creation vs. Restoration*, is being written to document the tradeoff decisions being made during the project.

The Silicon Valley Forth Interest Group has requested that the team present and demo the IBM 1620 Jr. at one of their monthly meetings at Stanford University. That would best be done once the simulator and console typewriter are functionally complete and an application program from the IBM 1620 Software Library could be run. A discussion thread has been started on the group's mailing list about the feasibility of porting eForth to the IBM 1620. The key issue is dealing with the IBM 1620's ***very*** limited character set.

Front Panel (Steve Casner, Joe Fredrick)

The front panel sub-project consists of all physical and electrical work done on the spare IBM 1620 front panel, the Raspberry Pi 3 (RPi), and the interface circuitry connecting the two.

All of the functional work on the front panel was completed in June and it was moved to Southern California to begin development of the simulator.

There are a few, non-functional items remaining to complete the front panel:

- Acquire a "new stock" incandescent light and repeat the brightness calibration work. One of the two original manufacturers of bulbs for the IBM 1620, Eldema, still makes the same light (#CF03-CCB-1869). Since incandescent lights can fade due to use and the original lights in the spare front panel being used seem dimmer than the team remembers, it is worth investigating further.

- Label the added indicator lights and switches. This will most likely be done with press-on lettering. Silk screening would require disassembly of the entire front panel – a tedious task now that the panel is completely wired.

- Apply Loctite 271 to the threads of the brass toggle switch handles.

- Repair the lower front right corner of the wooden case.
- Add an internal latch to prevent the front panel from being opened when unattended.
- Add a hinged, lockable, plexiglass back to protect the internals, but allow viewing.
- Add external connectors for power, USB, etc. to the back or side of the case.
- Add folding handles to the sides of the case, so that it can be carried safely.
- Paint the case metallic gray or add a metal veneer to better match the real IBM 1620 cabinet.

Console Device (Dave Babcock, Joe Fredrick, Dag Spicer)

The console device sub-project consists of all physical and electrical work done on the console I/O device, also known as the console typewriter.

Two IBM/Lexmark Wheelwriter 1000's were purchased for the project in June – one for hardware team and one for the simulator development. Both typewriters will be modified so that there are primary and spare units.

The hardware team's Wheelwriter has been opened, inspected, and explored for adaptor board placement. A breakout board, between the keyboard and motherboard, was built to measure and reverse-engineer the interface. The ribbon cable zero-force connectors on the typewriter motherboard were replaced and will be used on the breakout and prototype adapter boards. This is a temporary measure while a low-quantity supply of zero-force connectors can be located. The goal is that no changes are needed to the Wheelwriter beyond inserting the adaptor board. There is some corrosion and oxidation in this "as-is" Wheelwriter that will need to be cleaned. The other Wheelwriter is in better physical condition.

A possible source for the Wheelwriter 1000 maintenance manual is being pursued. It was confirmed that there wasn't a tractor feed option for this typewriter model.

Work on the *IBM 1620 Jr. Console Typewriter Functional Specification* is underway. The hardware team needs this document to design and program the custom adaptor board that will turn the Wheelwriter 1000 into an IBM 1620 console typewriter.

There is some interest from the vintage computer community in adapting what is done for the IBM 1620 Jr. console typewriter for other purposes (Teletype KSR-33, PDP-1 console typewriter, etc.). Some consideration will be given in the design of the adapter board to make other uses possible.

Card Read-Punch Device (Dave Babcock)

The card read-punch device sub-project consists of emulating the physical punched card device in the software simulator using USB memory sticks to represent individual card decks.

Development of the card read-punch device has not begun.

Simulator (Dave Babcock, Steve Casner)

The simulator sub-project consists of developing a new IBM 1620 cycle-level simulator, coded in C, that interfaces with the front panel hardware. It will simulate an IBM 1620 Model 1 Level F punched card system with 60,000 digits of memory and the automatic divide, indirect addressing, MF/TNF/TNS, and floating point processor options.

Development of the simulator began in July and enough basic functionality was operational to run the "Powers of 2" demo program at VCF West in early August.

The following instructions are fully implemented and pass their portions of IBM's CU01 diagnostic program:

- Add (A – 21), Add Immediate (AM – 11)
- Subtract (S – 22), Subtract Immediate (SM – 12)
- Compare (C – 24), Compare Immediate (CM – 14)
- Transmit Digit (TD – 25), Transmit Digit Immediate (TDM – 15)
- Transmit Field (TF – 26), Transmit Field Immediate (TFM – 16)
- Branch and Transmit (BT – 27), Branch and Transmit Immediate (BTM – 17)
- Transmit Record (TR – 31)
- Set Flag (SF – 32)
- Clear Flag (CF – 33)
- No Operation (NOP – 41)
- Branch Back (BB – 42)
- Branch on Digit (BD – 43)
- Branch No Flag (BNF – 44)
- Branch No Record Mark (BNR – 45)
- Branch Indicator (BI – 46) for all 1620 indicators
- Branch No Indicator (BNI – 47) for all 1620 indicators
- Halt (H – 48)
- Branch (B – 49)
- Branch No Group Mark (BNG – 55)

The implementation of most of the remaining instructions was stalled until the appropriate IBM Customer Engineering manuals were located. All of the needed documentation has now been found and development of the simulator can continue. The instructions still to be implemented are:

- Indirect Addressing
- Multiply (M – 23), Multiply Immediate (MM – 13)
- Load Dividend (LD – 28), Load Dividend Immediate (LDM – 18)
- Divide (D – 29), Divide Immediate (DM – 19)
- Move Flag (MF – 71)
- Transmit Numeric Strip (TNS – 72)
- Transmit Numeric Fill (TNF – 73)
- Floating Add (FADD – 01)
- Floating Subtract (FSUB – 02)
- Floating Multiple (FMUL – 03)
- Floating Shift Left (FSL – 05)
- Transmit Floating (TFL – 06)
- Branch and Transmit Floating (BTFL – 07)
- Floating Shift Right (FSR – 08)
- Floating Divide (FDIV – 09)

Temporary versions of some of the console typewriter I/O instructions (Control – K 34, Write Numerically – WN 38, and Write Alphamerically – WA 39) were written to run the VCF demo program. These instructions, and the remaining console typewriter I/O instructions (Dump Numerically – DN 35, Read

Numerically – RN 36, and Read Alphamerically – RA 37), will be properly implemented once the *IBM 1620 Jr. Console Typewriter Functional Specification* is finalized. Punched card I/O instructions will be implemented after the Card Read-Punch Device is designed.

The Windows-based development environment consists of the Eclipse IDE v4.7.1 and an GNU ARM C cross-compiler v6.3.0. The development system is networked with the IBM 1620 Jr. VNC is used to remotely log into the RPi system.

The simulator is written in C and runs on the Raspbian (Linux) operating system on the Raspberry Pi 3. The simulator program is organized as 3 threads, each running on a dedicated RPi processor core – the main thread, the simulator thread, and the front panel thread. The simulator and front panel threads run independently and asynchronously, with only two non-interlocked data structures between them. The main simulation loop is synched to the RPi's system hardware timer and takes 20 microseconds (+/- 0.25usec, with no drift) – the IBM 1620 Model 1's cycle time. The front panel's display loop runs every 10 milliseconds, synched to Raspbian's real-time clock. The intensity of each of the front panel lights is averaged and integrated over time and produces an authentic brightness and pulsing.

As part of the simulator work, a new file format (cmem) was created to store IBM 1620 **c**ore **mem**ory images. This simple [editable] text format is used by the simulator to save and reload its "core memory" between runs to simulate non-volatile core memory. Until punched card I/O is implemented, it also serves as a mechanism for loading programs, specifically the diagnostics and demo programs.

Development has started on a graphical front panel that runs on Windows and eventually Mac. Sharing the same simulator thread, this virtual IBM 1620 simulator will be needed for console typewriter development/testing and by the Software Library team to test application programs in the library. The Eclipse IDE, with the addition of a native GNU C compiler v6.4.0, generates the Windows IBM 1620 simulator as well as the RPi IBM 1620 Jr. simulator. The graphical front panel uses SDL2 (Simple DirectMedia Library, Version 2) and OpenGL v4.6.

Software Library (Lee Courtney, Dave Babcock)

The software library sub-project consists of converting all of the museum's IBM 1620 software collection to a format usable by the IBM 1620 Jr., assembling the associated documentation, and testing the programs on the IBM 1620 Jr.

The "Powers of 2" demo program and the IBM CU01 (Basic Instructions and Automatic Divide), CU02 (Error Check), CU03 (Indirect Addressing), CU04 (Memory) and CU05 (Special Instructions) diagnostic programs were all converted to the cmem format for demoing/testing the simulator.

The two engineers at VCF who volunteered to help will be needed once work on the software library begins. Priority will be given to processing the Baseball Simulation program, requested by Major League Baseball's for their Hall of Fame.

Operations Guide (Dave Babcock)

The operations guide sub-project consists of writing a brief document on the operation of the IBM 1620 Jr.

Notes for the Operations Guide are being kept as the simulator is being developed.

Website (Dave Babcock, Team)

The website sub-project consists of selecting, configuring, and populating a dedicated website on the IBM 1620 which will document the IBM 1620, the CHM IBM 1620 restoration project, and the IBM 1620 Jr. project.

To locate the detailed technical information needed to develop the simulator, a more complete list of all IBM 1620 manuals has been compiled. This list, including PDF versions of some of the documents, will be a part of the website.

Development of the website itself has not started.