David Husband, M.Sc in IT, *Baremetal Engineer Extraordinaire*

What you see detailed on my CV are "merely" the things I have done to earn money...
In reality, my abilities, knowledge and capabilities range far beyond my CV..

I am a ***BAREMETAL ENGINEER*** and what is in this document showcases a small part of my ***baremetal engineering***...
I have an embryonic website on the subject here: http://baremetal.engineer/

The purpose of that web site[1] is to expose and document what I have spent many years doing so that, hopefully, young engineers will learn from it !!

In http://baremetal.engineer/baremetal.software.engineer.pdf I showed my proven abilities, including researching, analysing, extracting and applying freshly acquired knowledge from within the TCP/IP subject domain

In http://baremetal.engineer/baremetal.blockchain.engineer.pdf I showed my recently acquired blockchain knowledge, including my analysis of how Blockchain can be applied to embedded IoT systems.

To be brief here, I have created a number of Appendices where I go into further detail on particular topics

In this document, I present an overview of how electronic hardware has evolved over the years...

There are two aspects to modern electronic hardware:

- **VERY LITTLE HAS CHANGED REGARDING THE PRINCIPLES OF ELECTRONIC HARDWARE !!**
- **ELECTRONIC HARDWARE HAS NEVER BEEN EASIER !! (but prototyping is much more difficult !!)**

Very Little Has Changed Regarding The Principles Of Electronic Hardware

I was having a conversation recently with an agent and I mentioned that not only was I a software engineer with a recent Masters in Software Engineering from a top university, but before that I was (and I still am) a hardware engineer who designed hardware and wrote the software to run it...

He replied "How long ago was that?" I replied "25 years ago" He replied with a series of ill-informed, disdainful, disrespectful and offensive comments... "Oh, it's ALL CHANGED NOW", and other comments implying that what I knew about hardware was now "worthless"..

The reality is that ***NOTHING has essentially changed*** at all!   The underlying ***PRINCIPLES have not changed at all***!

Electronics is based upon PHYSICS and all the pioneers such as ***Michael Faraday, Thomas Edison, George Westinghouse, Nicola Tesla***, etc, were working around the 1800's; "Wireless" is essentially based upon ***James Clerk Maxwell***'s equations which were published between 1861 & 1862... Communication & Information theory by ***Nyquist***, ***Shannon***, et al, were developed between 1920 & 1940

Bluetooth, Wi-Fi, 5G are ALL based upon 160-year old (or even older) equations & theory from ***Maxwell*** and other pioneers...

The first digital codes were developed by ***Samuel Morse***[2] between 1837 & 1847 !!
Modern satellites and space exploration are ALL based upon ***Newtonian Mechanics*** from around 1687 !!
Even copper Ethernet cables are based upon "twisted-pair" wiring invented by ***Alexander Graham Bell*** in 1881 !!
All modern microprocessor arithmetic/logic[3] is based upon ***Boolean Algebra***[4] devised & developed by ***George Boole***[5] between 1847 & 1854 !!

Maxwell's equations are of fundamental importance to our modern world:
"*The equations provide a mathematical model for electric, optical, and radio technologies, such as power generation, electric motors, wireless communication, lenses, radar etc..*"[6]

What HAS changed is "merely" how things are ***IMPLEMENTED***! The microprocessor DOES NOT EXIST! It has been replaced by a much more complex "***System-on-Chip***" ("SoC"), etc, which incorporates high levels of functional integration...

---

[1] You will see from that URL that my skills & abilities do not extend to scripting web pages !!
[2] *https://en.wikipedia.org/wiki/Morse_code*
[3] *https://en.wikipedia.org/wiki/Boolean_function*
[4] *https://en.wikipedia.org/wiki/Boolean_algebra*
[5] *https://en.wikipedia.org/wiki/George_Boole*
[6] *https://en.wikipedia.org/wiki/Maxwell%27s_equations*

***"Implementation" changes all the time**, but for an engineer, it is really about developing an on-going ability to READ & DIGEST data sheets and specifications. It is __ALL RULE-BASED__, not rocket science or nuclear physics...*

Basically RTFM[7]! Then design accordingly!

Example:

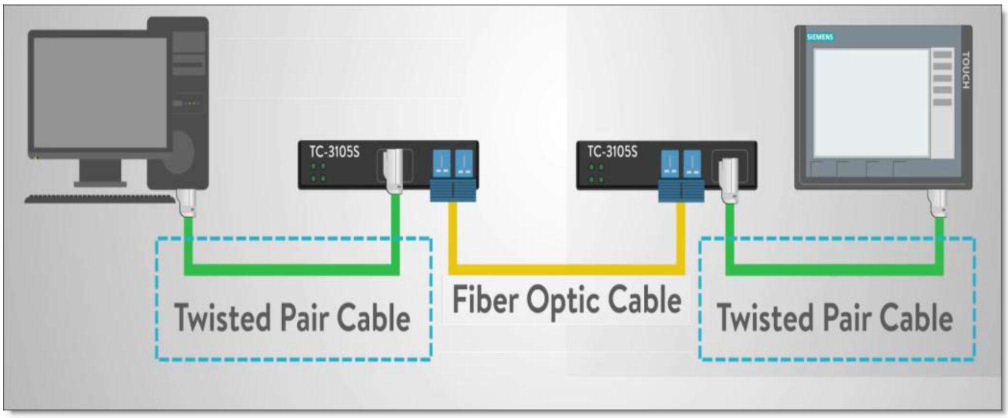

Figure 1 - A simple Ethernet System

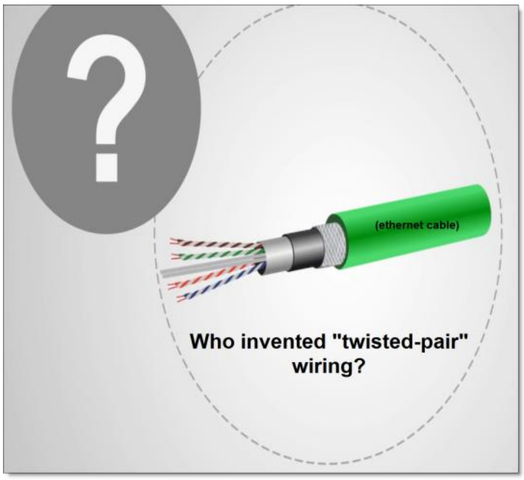

Figure 2 - The Ethernet Multiple Twisted-Pair Cable

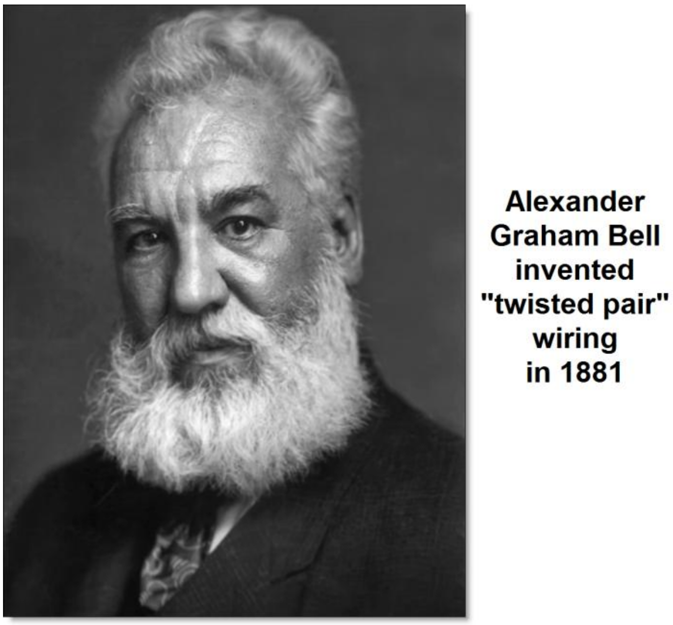

Figure 3 - Alexander Graham Bell[8] - Technology Pioneer !!

[7] Read The Flipping Manual and RTFDS -- Read The Flipping Data Sheet !!

[8] https://en.wikipedia.org/wiki/Alexander_Graham_Bell

Appendix A. In the very beginning - Before Vacuum Tubes (Thermionic Valves)

All the early pioneers had to work with the very crude technology of the day...

It was essentially based only upon electro-magnetism and current-loops and implemented electro-mechanically

Even the early radio pioneers such as Hertz[9] & Marconi[10] used electro-mechanical means to generate the crude radio-frequency ("RF") energy they used for communication

The early "spark transmitters" used a collapsing magnetic field from a de-energised coil to generate RF

"*A spark-gap transmitter[11] is an obsolete type of radio transmitter which generates radio waves by means of an electric spark. Spark-gap transmitters were the first type of radio transmitter, and were the main type used during the wireless telegraphy[12] or "spark" era, the first three decades of radio, from 1887 to the end of World War I*"

Later transmitters used high-speed alternators to generate continuous RF energy[13,14]

In due course, voice and music could be transmitted via amplitude modulation[15]

"*Early experiments in AM radio transmission, conducted by Fessenden, Valdemar Poulsen, Ernst Ruhmer, Quirino Majorana, Charles Herrold, and Lee de Forest[16], were hampered by the lack of a technology for amplification. The first practical continuous wave AM transmitters were based on either the huge, expensive Alexanderson alternator, developed 1906–1910, or versions of the Poulsen arc transmitter (arc converter), invented in 1903*"

Detecting the RF signals was an even bigger problem and severely limited the range of the transmitting equipment, because (a) RF detectors had to be devised and (b) there was no means of amplification[17]

In due course the "crystal detector"[18] was devised. "*A crystal detector is an obsolete electronic component used in some early 20th century radio receivers that consists of a piece of crystalline mineral which rectifies the alternating current radio signal and was employed as a detector[19] (demodulator) to extract the audio modulation to produce the sound in the earphones. It was the first type of semiconductor diode[20], and one of the first semiconductor electronic devices. The most common type was the so-called cat's whisker detector, which consisted of a piece of crystalline mineral, usually galena (lead sulphide), with a fine wire touching its surface*" "*Unlike modern semiconductors, such a diode required painstaking adjustment of the contact to the crystal in order for it to rectify*"

These detectors were used in the entirely passive "Crystal Radios"[21] of the time. A big step forward took place when Lee de Forest devised the vacuum tube[22] diode[23]

An even bigger step forward took place with the invention of the vacuum tube triode[24] by Lee de Forest

These developments were the dawning of the "Electronic Age" that our modern world was based on and developed from...[25]

By its very nature, thermionic tubes ("valves") were inherently analogue[26], and are in particular transconductance[27] devices

---

9 https://en.wikipedia.org/wiki/Heinrich_Hertz
10 https://en.wikipedia.org/wiki/Guglielmo_Marconi
11 https://en.wikipedia.org/wiki/Spark-gap_transmitter
12 https://en.wikipedia.org/wiki/Wireless_telegraphy
13 https://en.wikipedia.org/wiki/Alexanderson_alternator
14 https://en.wikipedia.org/wiki/Poulsen_arc
15 https://en.wikipedia.org/wiki/Amplitude_modulation
16 https://en.wikipedia.org/wiki/Lee_de_Forest
17 https://en.wikipedia.org/wiki/Amplifier
18 https://en.wikipedia.org/wiki/Crystal_detector
19 https://en.wikipedia.org/wiki/Envelope_detector
20 https://en.wikipedia.org/wiki/Diode
21 https://en.wikipedia.org/wiki/Crystal_radio
22 https://en.wikipedia.org/wiki/Vacuum_tube
23 https://en.wikipedia.org/wiki/Vacuum_tube#Diodes
24 https://en.wikipedia.org/wiki/Triode
25 https://en.wikipedia.org/wiki/Electronics - *"Electronics comprises the physics, engineering, technology and applications that deal with the emission, flow and control of electrons in a vacuum and matter"*
26 https://en.wikipedia.org/wiki/Analogue_electronics

Appendix B. VACUUM TUBES (THERMIONIC VALVES) & THE BEGINNING OF THE ELECTRONIC AGE

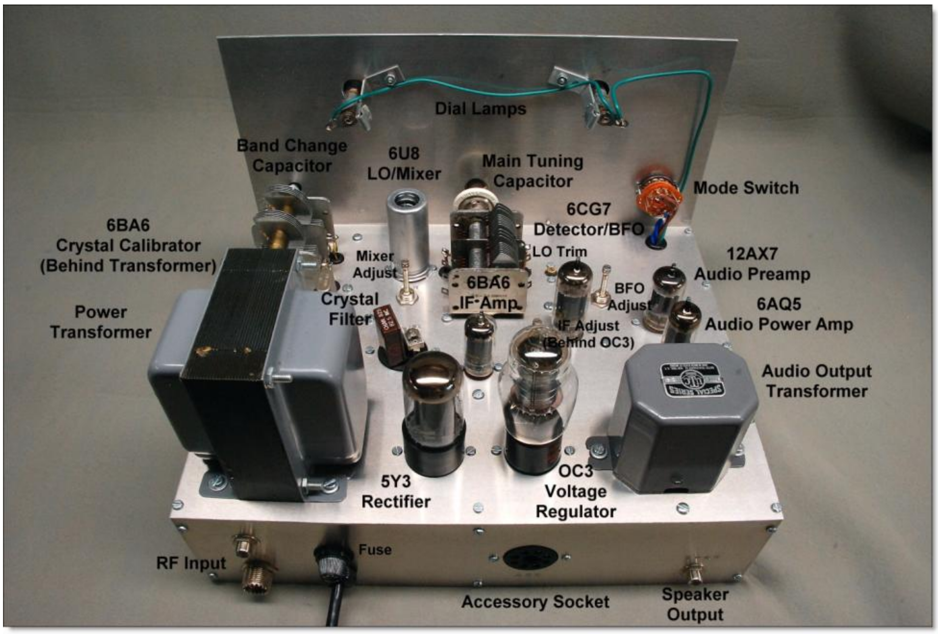

Figure 4 - An uncased view of a simple superhet[28] HF radio receiver with its functionality annotated...[29]

The invention of the Thermionic Tube ("Valve") kicked off the golden age of electronics and this eventually led to the development of everything we have in today's age...

However the technology was quite crude & "clunky" and it was hard & skilful work. See Figure 4 above and Figure 5 below.

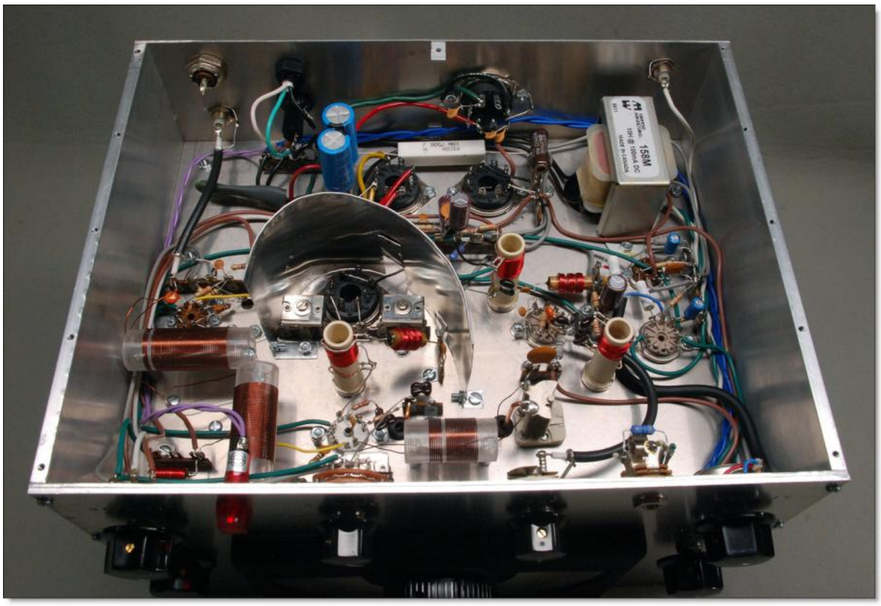

Figure 5 - The underside of the receiver

The levels of functional integration were low - but there was some in the implementation of the valves...

Valves were often "doubled-up" with effectively 2 valves in the same glass envelope

This can be seen in Figure 6 below, where the 6U8A[30] ("ECF82") is a "double valve" and uses a pentode as a mixer with a triode being used as the local oscillator for the mixer...

---

[27] ***https://en.wikipedia.org/wiki/Transconductance*** *"For vacuum tubes, transconductance is defined as the change in the plate(anode) current divided by the corresponding change in the grid/cathode voltage, with a constant plate(anode) to cathode voltage"*

[28] ***https://en.wikipedia.org/wiki/Superheterodyne_receiver#Superheterodyne***

[29] All images courtesy of ***https://www.frostburg.edu/personal/latta/ee/6x2rcvr/***

[30] ***https://mullard.org/collections/audi-valves-ecf82-6u8a-cv5065-triode-pentode***

Elsewhere, the 6CG7[31] is a double triode, along with the 12AX7A[32] with only one half in use...

See around points 1,2,3,4,5,6,7 & 17 in Figure 6 below. This is the power supply part of the radio. The rectifier valve 5Y3-GT[33] is a strange beast! It is a double-diode with a directly heated cathode of 5v with its own transformer winding ! It will have been fed by a 200-0-200 or so winding from T2 & would have produced a rectified but unsmoothed output at Point 2. Smoothing would have been provided by the components around Point 3, leading to an HT voltage of around +250V dc at points 4, 5 & 10

The +250V HT voltage was not stabilised and so would tend to vary depending upon its load, so a voltage regulator device at point 6 was employed. This is not actually a vacuum valve, but a neon tube voltage stabiliser[34] which provides a stable +108V dc which feeds the local oscillator V1b and the calibrator V8 which oscillates at 100KHz and injects that frequency and all its harmonics into the receiver's front end...

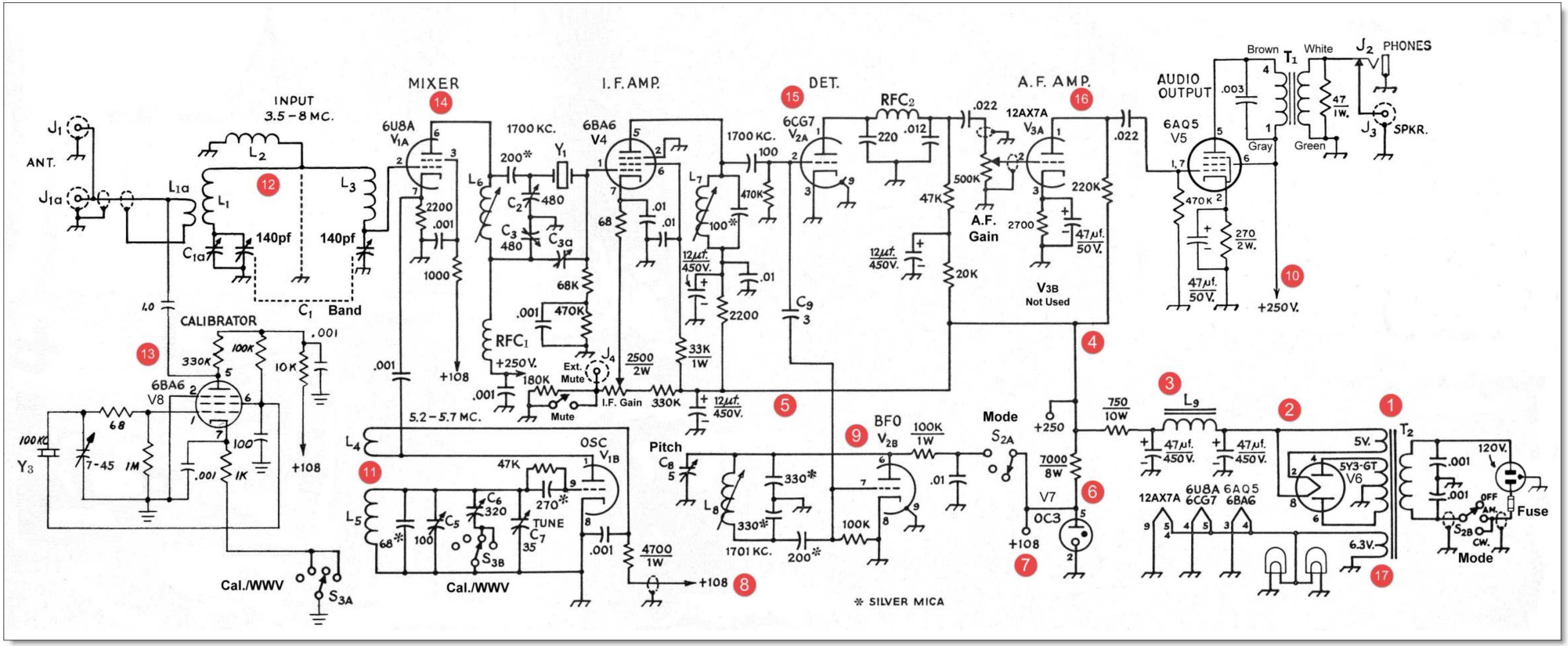

Figure 6 - A schematic diagram of the superhet HF receiver

31 https://www.radiomuseum.org/tubes/tube_6cg7.html
32 https://en.wikipedia.org/wiki/12AX7
33 https://en.wikipedia.org/wiki/5Y3
34 https://en.wikipedia.org/wiki/Voltage-regulator_tube

Appendix C. How Things Were - #1 Pencil & Paper

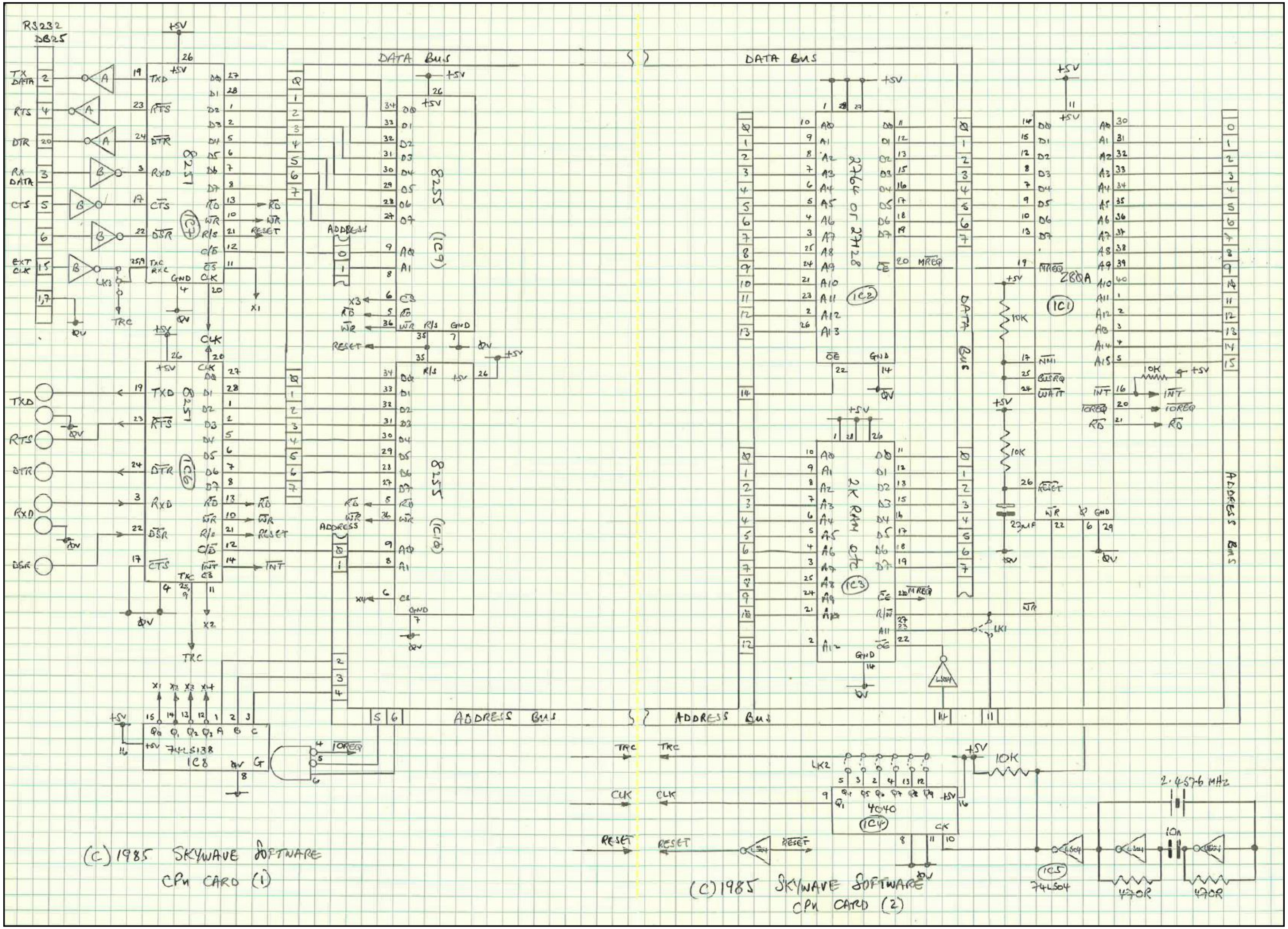

Figure 7 - An early "pencil & paper" Z80 Card Design by David Husband from 1985

In Figure 7 above is a composite scan[35] of a pencil drawing I made back in 1985 on squared paper

All drawings were "pencil & paper" efforts as there was no CAD[36] ability at this time... (Or GUI's !! Just DOS)[37]

This was not the earliest Z80 design I did. Early Z80s were fabricated using NMOS[38] technology transistors and had a limited bus-driving ability[39] and so the data and address busses to the Z80 had to incorporate address & data buffering which added an extra layer of design complexity...

Fortunately, the rapid adaptation of CMOS[40] technology put a stop to this rather tiresome limitation

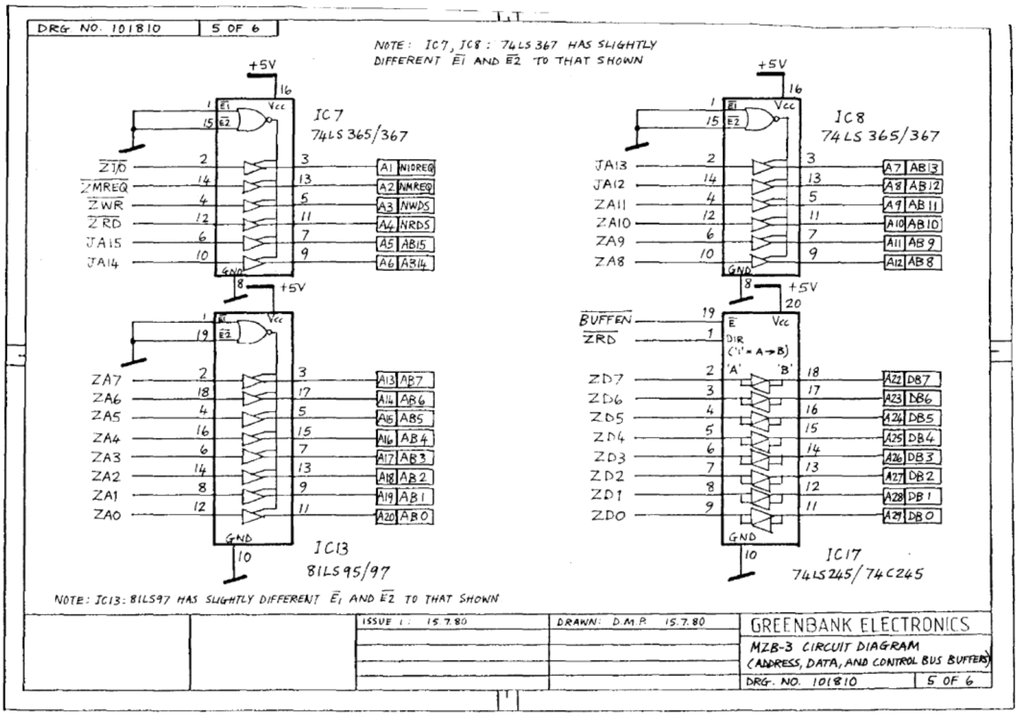

Figure 8 - An example of Address & Data Bus buffering[41]

However, working with microprocessors was hard work. Unlike today, the processor was "naked" and the designer had to furnish a CPU clock, a reset, address decoding, a baud-rate clock for any serial ports, serial ports, parallel ports, etc

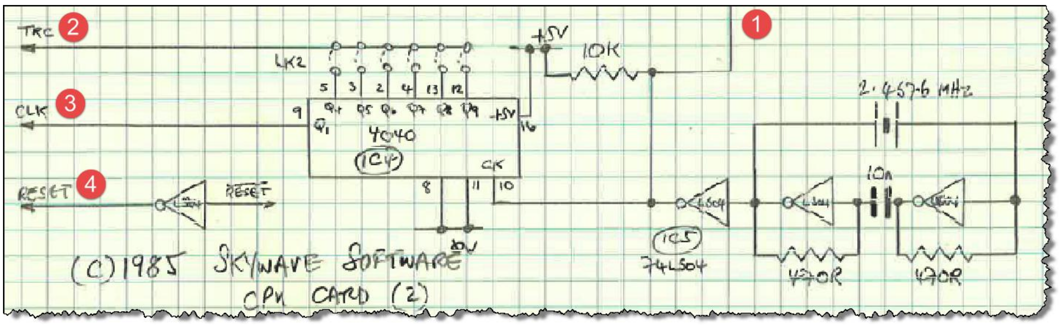

Figure 9 - Z80 clock & Baud-rate Generator..

Figure 9 above shows my favourite circuitry. The 2.4576 MHz frequency is a "magic number" because it can clock the processor and be divided down (by IC4) to provide the baud-rate clock for the serial port via selection on LK2 through to Point 2. The 8251 serial port needs a clock (not related to the baud rate) and this is provided by taking out a signal onto Point 3

Referring to Figure 10 below, the 74LS138 was my favourite address decoder and in this instance was gated with the Z80 I/O strobe signal /IOREQ to decode the two serial ports and the two parallel ports via I/O addressing

35 So this was 2 A4 pages (one on the left-hand side & one on the right-hand side with the dashed yellow line between)...
36 https://en.wikipedia.org/wiki/Computer-aided_design
37 https://en.wikipedia.org/wiki/Graphical_user_interface
38 https://en.wikipedia.org/wiki/NMOS_logic
39 https://en.wikipedia.org/wiki/Digital_electronics#Fanout
40 https://en.wikipedia.org/wiki/CMOS
41 Not for the faint-hearted designer...

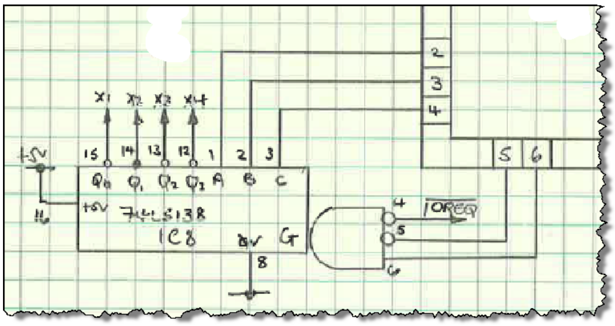

Figure 10 - Z80 (I/O) Address Decoding...

Referring to Figure 7 above, the ROM and the RAM in the system is decoded "reflectively" by using A14 and creating an /A14 signal via an inverter. Using this method means that no specific address decoding logic is needed

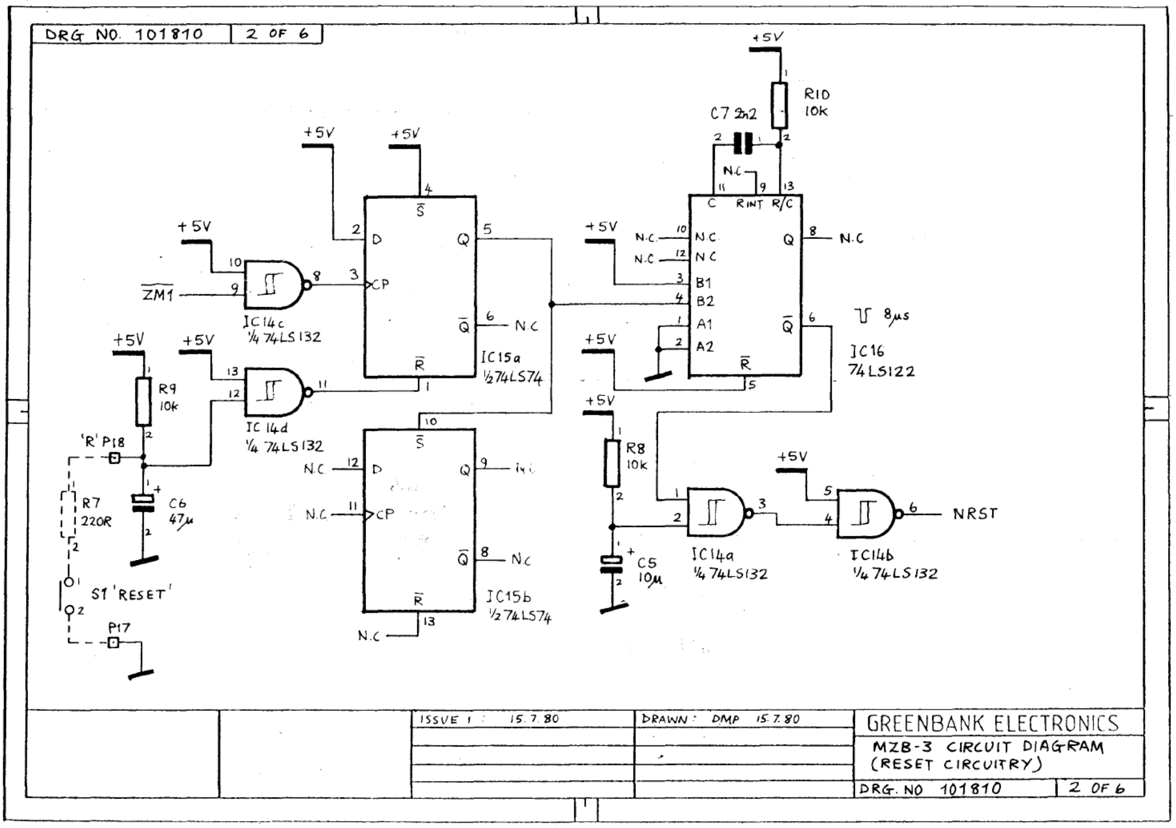

Figure 11 - An example of reset circuitry

Reset circuitry can be a game, but nowadays there are a large number of reset ICs available. Referring to Figure 11 above, this is an example of a more complicated reset arrangement for a Z80 card used in an expandable rack system. R9 & C6 create the original reset pulse upon power-up. The arrangement in dotted lines is an external reset button elsewhere in or on the rack system

IC14d is a Schmitt-trigger[42] input NAND[43] Logic Gate[44] 74LS132[45] which "cleans-up" the rising edge from C6 charging towards 5v
IC15a is a "flip-flop"[46] S/R latch 74LS74[47] which synchronises the generated reset signal with /ZM1
The signal is now fed into a 74LS122 Monostable[48] multivibrator which generates a fixed, low-going 8uS pulse when triggered via input B2

This circuitry is an example of integrated circuit[49] digital electronics[50] implementing Boolean Logic[51] functions by using the TTL[52] 7400 series[53] logic family[54]. The diagram in Figure 11 above forms a "logic block"[55]

---

42 https://en.wikipedia.org/wiki/Schmitt_trigger
43 https://en.wikipedia.org/wiki/NAND_gate
44 https://en.wikipedia.org/wiki/Logic_gate
45 https://assets.nexperia.com/documents/data-sheet/74HC_HCT132.pdf
46 https://en.wikipedia.org/wiki/Flip-flop_(electronics)
47 https://web.mit.edu/6.111/www/s2007/datasheets/74LS74.pdf
48 https://en.wikipedia.org/wiki/Monostable

VERY IMPORTANT POINT !!

Referring to Figure 9, Figure 10 & Figure 11 above which is dated "***1980***" - It is now ***2020***, ***FORTY YEARS LATER*** - Guess what?

NOTHING HAS CHANGED !!

Looking at https://digikey.co.uk at their inventory shows very large quantities of all these parts[56], both in the original 0.1" pin spacing DIL[57] packages and in SMD[58] packages....

The large quantities of DIL packages shows that:

THE SAME DEVICES ARE STILL BEING USED 40 YEARS LATER !!

The large quantities of SMD packages shows that:

THE SAME DEVICES ARE STILL BEING USED IN NEW DESIGNS 40 YEARS LATER !!

https://www.digikey.co.uk/products/en/integrated-circuits-ics/logic-signal-switches-multiplexers-decoders/743?k=74LS138	Qty: 2984 0.1" DIL	Qty: > 10,000 SMD
https://www.digikey.co.uk/products/en?keywords=74LS04	Qty: 2831 0.1" DIL	Qty: > 20,000 SMD
https://www.digikey.co.uk/products/en?keywords=CD4040	Qty: 5832 0.1" DIL	Qty: > 50,000 SMD
https://www.digikey.co.uk/products/en?keywords=74LS74	Qty: 3712 0.1" DIL	Qty: > 3,000 SMD
https://www.digikey.co.uk/products/en?keywords=74LS122	Qty: 260 0.1" DIL	Qty: > 3,000 SMD
https://www.digikey.co.uk/products/en?keywords=74LS132	Qty: 3197 0.1" DIL	Qty: > 7,500 SMD
https://www.digikey.co.uk/products/en?keywords=74LS245	Qty: 5587 0.1" DIL	Qty: > 75,000 SMD

Prototyping

Ok, prototyping a design was much easier years ago because EVERYTHING was in a 0.1" matrix Dual-in-Line ("DIL") package so circuits could be laid up as seen in Figure 12 below...

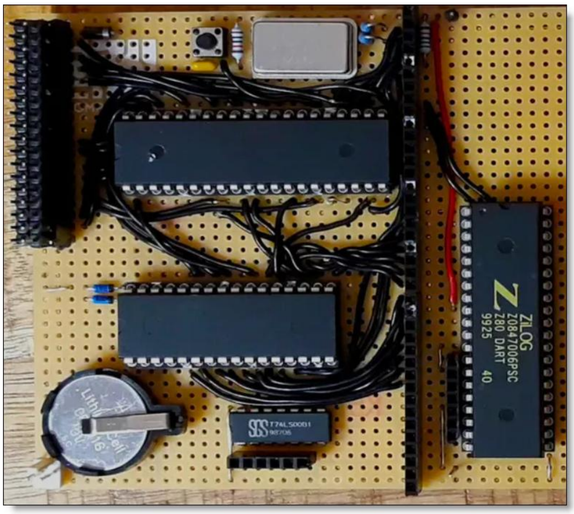

Figure 12 - An example of "patch-wiring" a prototype on a 0.1" matrix board[59]...

Shown in Figure 13 is another example of "patch-wiring" a prototype in this case on a prototyping board specifically designed to suit the 160mm card racking system. Note the multi-pin bus connector on the left-hand side...

---

49 https://en.wikipedia.org/wiki/Integrated_circuit
50 https://en.wikipedia.org/wiki/Digital_electronics
51 https://en.wikipedia.org/wiki/Boolean_function
52 https://en.wikipedia.org/wiki/Transistor%E2%80%93transistor_logic
53 https://en.wikipedia.org/wiki/7400-series_integrated_circuits
54 https://en.wikipedia.org/wiki/Logic_family
55 https://en.wikipedia.org/wiki/Logic_block
56 And this is only in the obsolescent 5v LS TTL 7400 series logic family -- there are others such as: 74ALS, 74F, 74S, 74AS, 74C, 74HC, 74HCT, 74AC, 4ACT, 74AHC, 74LVC.... See: http://www.ti.com/lit/sg/sdyu001ab/sdyu001ab.pdf
57 https://en.wikipedia.org/wiki/Dual_in-line_package
58 https://en.wikipedia.org/wiki/Surface-mount_technology
59 Image courtesy of https://www.hackster.io/michalin70/cp-m-on-a-minimal-z80-computer-cecaf7

Figure 14 shows the reverse side of the board, with the interconnections made by "wire-wrapping"[60]

Over the years, as the level of integration has increased, integrated circuits ("ICs") have sported more and more pins. This has driven a need to make the packages smaller and smaller and as mobile phones and laptop computers have become more popular and powerful. In addition sales volumes have been ever-increasing leading to automated assembly using "Surface Mounted Technology" to mount components on circuit boards rather than the traditional "through-hole" practice of the past

This has made prototyping more and more difficult and impractical. See Appendix D. below for how this is performed in the 2020's

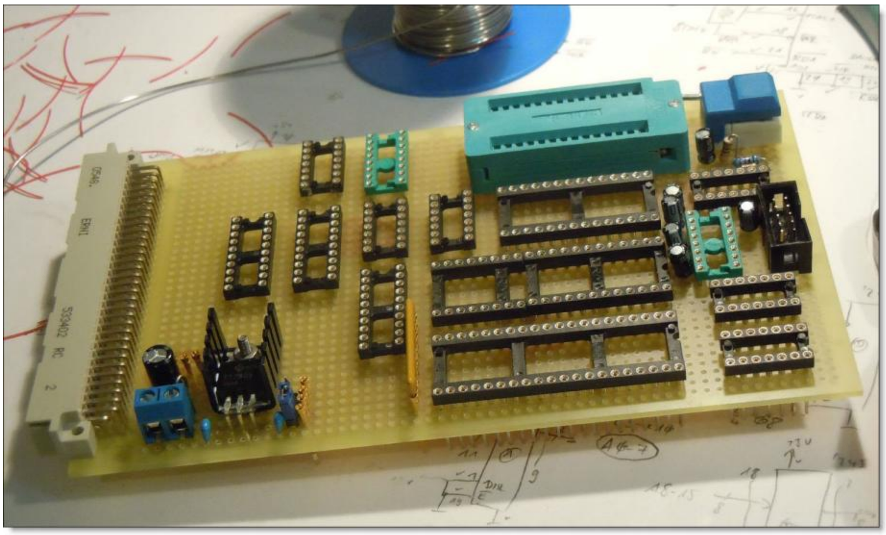

Figure 13 - Another example of a prototype[61] on a 0.1" matrix board...

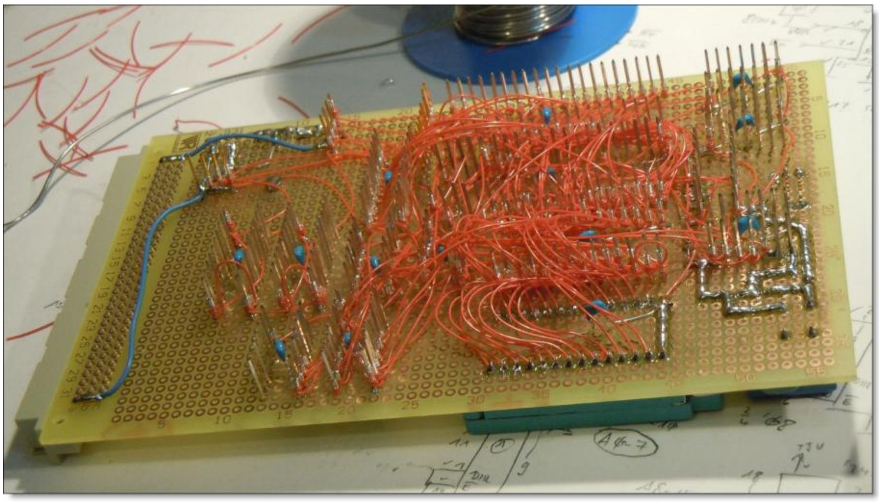

Figure 14 - The reverse of the board showing that the interconnections have been "wire-wrapped"

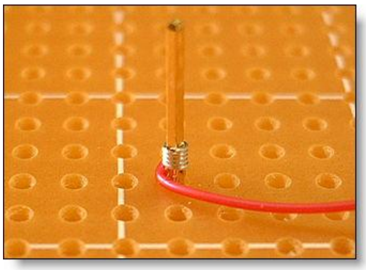

Figure 15 - Wrapping wire onto a square pin[62]

60 ***https://en.wikipedia.org/wiki/Wire_wrap***
61 All images courtesy of ***http://www.vaxman.de/projects/tiny_z80/*** And Figure 14
62 Image from: ***https://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Wire_Wrapping.jpg/330px-Wire_Wrapping.jpg***

Appendix D. More & More Integration...

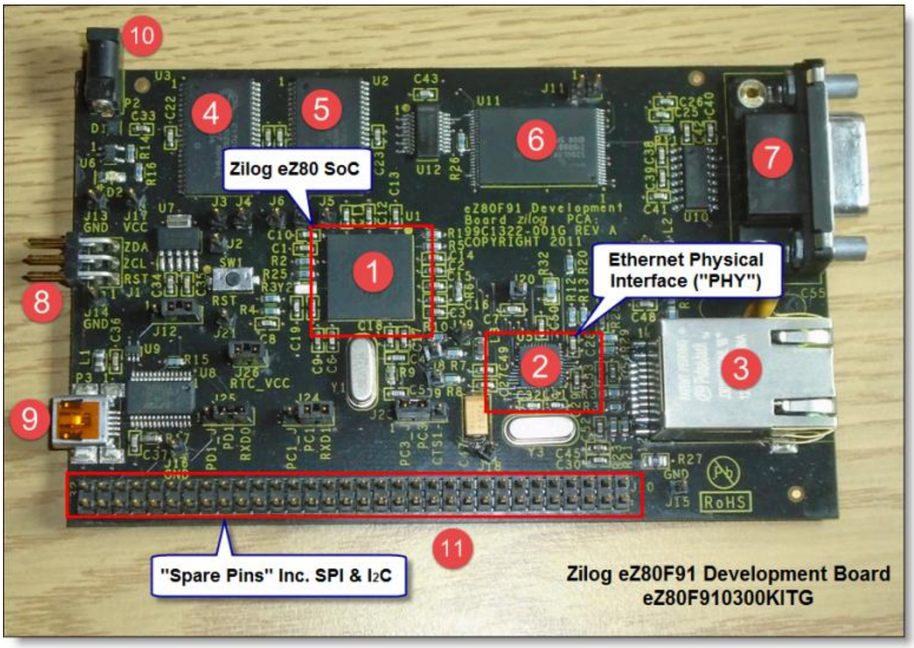

Figure 16 - The very highly integrated[63] eZ80 Development Board

The Zilog eZ80 "System-on-a-Chip"[64] ("SoC") development board shown in Figure 16 above is a good example of how the chip manufacturers encourage potential designers to adopt their offerings in new designs...

It is actually a case of manufacturers "***making a virtue out of necessity***" because in the case of Figure 16 above, the eZ80 SoC [@ 1] is such a highly integrated system (See Figure 17 below) that no potential user of the eZ80 would consider using it without the manufacturer supplying an "***evaluation platform***" along with a suite of free software !!

So this is what ALL manufacturers do !! So in that respect ***Hardware has NEVER BEEN EASIER !!***

This is the functionality contained in the eZ80 Development Board in Figure 16 above:

1) Zilog eZ80 SoC & supporting components
2) Ethernet Physical Interface[65] ("PHY") & supporting components
3) Ethernet Socket[66] to allow the connection of an Ethernet cable[67]
4) 1Mb of fast Static RAM[68] - 1 wait state
5) 1Mb of fast Static RAM - 1 wait state
6) 8Mb of slower Flash ROM[69] - 7 wait states
7) "Standard" PC-format DB-9[70] RS-232[71] serial port[72] & supporting components
8) Zilog Debug Interface ("ZDI") similar to JTAG[73] & supporting components
9) USB[74] interface & supporting components, furnishing a "virtual[75] serial port"[76] to a PC
10) 5v dc power supply socket & supporting components (optionally board can be powered via the USB interface)
11) "Spare Pins" including SPI[77] & I^{2}C[78] interfaces

---

63 https://en.wikipedia.org/wiki/System_integration *"System integration is defined in engineering as the process of bringing together the component sub-systems into one system (an aggregation of subsystems cooperating so that the system is able to deliver the overarching functionality) and ensuring that the subsystems function together as a system"*

64 https://en.wikipedia.org/wiki/System_on_a_chip

65 https://en.wikipedia.org/wiki/PHY

66 https://en.wikipedia.org/wiki/Modular_connector#8P8C

67 https://en.wikipedia.org/wiki/Ethernet_over_twisted_pair

68 https://en.wikipedia.org/wiki/Static_random-access_memory

69 https://en.wikipedia.org/wiki/Flash_memory

70 https://en.wikipedia.org/wiki/D-subminiature

71 https://en.wikipedia.org/wiki/RS-232

72 https://en.wikipedia.org/wiki/Serial_port

73 https://en.wikipedia.org/wiki/Boundary_scan

74 https://en.wikipedia.org/wiki/USB

75 https://en.wikipedia.org/wiki/Virtualization

76 https://en.wikipedia.org/wiki/USB_adapter

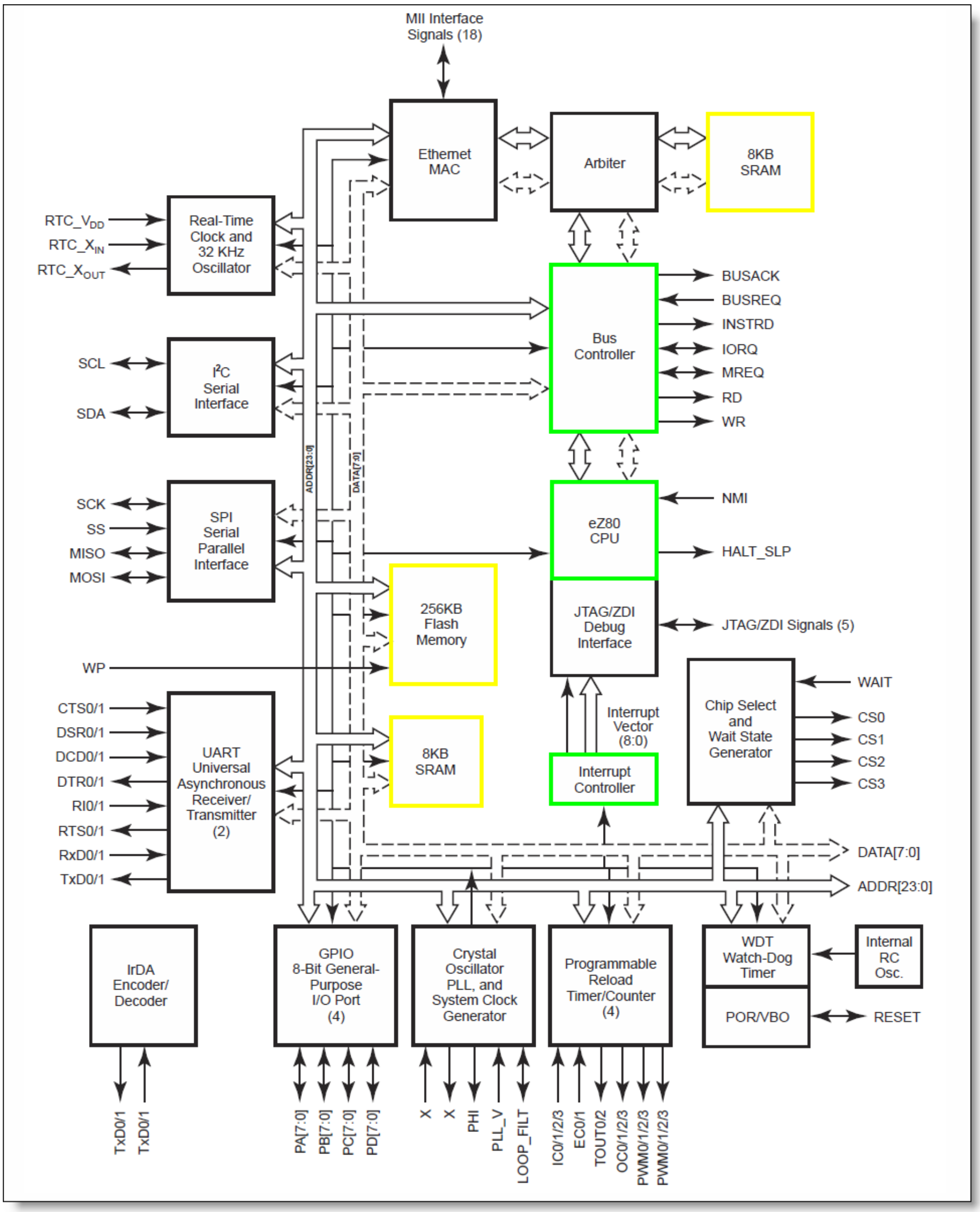

Figure 17 - The eZ80 "System-on-Chip" Block Diagram[79]

In Figure 17 above is the functionality contained in the eZ80 SoC
Outlined in green is the original Z80 functionality. Everything else is extra SoC functionality
Outlined in yellow is the extra RAM & ROM - this is internal memory and is in addition to the 10Mb of external memory on the development board detailed in Figure 16 above

Summary

So the diagrams in the early part of Appendix C. are all examples of digital logic using logic gates[80] (which implement logical conjunctions[81] using the binary number system[82] which "*was refined by Gottfried Wilhelm Leibniz*[83] *(published in 1705), influenced by the ancient I Ching's*[84] *binary system. Leibniz established that using the binary system combined the principles of arithmetic*[85] *and logic*[86]" https://en.wikipedia.org/wiki/Logic_gate

---

77 https://en.wikipedia.org/wiki/Serial_Peripheral_Interface
78 https://en.wikipedia.org/wiki/I%C2%B2C
79 From Zilog's eZ80F91 ASSP Product Specification - PS0270.pdf
80 https://en.wikipedia.org/wiki/Logic_gate
81 https://en.wikipedia.org/wiki/Logical_conjunction
82 https://en.wikipedia.org/wiki/Binary_number
83 https://en.wikipedia.org/wiki/Gottfried_Wilhelm_Leibniz
84 https://en.wikipedia.org/wiki/I_Ching
85 https://en.wikipedia.org/wiki/Arithmetic

"*In an 1886 letter, Charles Sanders Peirce*[87] *described how logical operations could be carried out by electrical switching circuits. Eventually, vacuum tubes replaced relays for logic operations. Lee De Forest's modification, in 1907, of the Fleming valve*[88] *can be used as a logic gate. Ludwig Wittgenstein*[89] *introduced a version of the 16-row truth table*[90] *as proposition 5.101 of Tractatus Logico-Philosophicus (1921). Walther Bothe*[91]*, inventor of the coincidence circuit, got part of the 1954 Nobel Prize in physics, for the first modern electronic AND*[92] *gate in 1924. Konrad Zuse*[93] *designed and built electromechanical logic gates for his computer Z1 (from 1935–38).*"

https://en.wikipedia.org/wiki/Logic_gate

"*From 1934 to 1936, NEC engineer Akira Nakashima*[94] *introduced switching circuit theory*[95] *in a series of papers showing that two-valued Boolean algebra*[96]*, which he discovered independently, can describe the operation of switching circuits. His work was later cited by Claude E. Shannon*[97]*, who elaborated on the use of Boolean algebra in the analysis and design of switching circuits in 1937. Using this property of electrical switches to implement logic is the fundamental concept that underlies all electronic digital computers. Switching circuit theory became the foundation of digital circuit design, as it became widely known in the electrical engineering community during and after World War II, with theoretical rigor superseding the ad hoc methods that had prevailed previously*"

https://en.wikipedia.org/wiki/Logic_gate

All that has changed is IMPLEMENTATION and further INTEGRATION

One of the big drawbacks of traditional digital logic design as featured in Figure 11 above is that the various logic devices are "hard wired" as part of the original circuit design and then as part of the printed circuit board design. This makes it very hard to change if it is found to contain minor faults

This has lead to the use of the field-programmable logic gate array[98] ("FPGA") devices which are "*an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence the term "field-programmable". The FPGA configuration is generally specified using a hardware description language*[99] *(HDL), similar to that used for an application-specific integrated circuit*[100] *(ASIC)*" https://en.wikipedia.org/wiki/Field-programmable_gate_array

"*FPGAs contain an array of programmable logic blocks*[101]*, and a hierarchy of "reconfigurable interconnects" that allow the blocks to be "wired together", like many logic gates that can be inter-wired in different configurations. Logic blocks can be configured to perform complex combinational functions*[102]*, or merely simple logic gates like AND and XOR*[103]*. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. Many FPGAs can be reprogrammed to implement different logic functions, allowing flexible reconfigurable computing as performed in computer software*" https://en.wikipedia.org/wiki/Field-programmable_gate_array

Virtualisation

"*In computing, virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual [...] hardware platforms ....*"[104]

Virtualisation can be simple, as shown in Figure 18 below or more complex as in a "software defined radio"[105] ("SDR") or as in a "virtual machine"[106]

---

86 https://en.wikipedia.org/wiki/Logic
87 https://en.wikipedia.org/wiki/Charles_Sanders_Peirce
88 https://en.wikipedia.org/wiki/Fleming_valve
89 https://en.wikipedia.org/wiki/Ludwig_Wittgenstein
90 https://en.wikipedia.org/wiki/Truth_table
91 https://en.wikipedia.org/wiki/Walther_Bothe
92 https://en.wikipedia.org/wiki/AND_gate
93 https://en.wikipedia.org/wiki/Konrad_Zuse
94 https://en.wikipedia.org/wiki/Akira_Nakashima
95 https://en.wikipedia.org/wiki/Switching_circuit_theory
96 https://en.wikipedia.org/wiki/Two-element_Boolean_algebra
97 https://en.wikipedia.org/wiki/Claude_Shannon
98 https://en.wikipedia.org/wiki/Field-programmable_gate_array
99 https://en.wikipedia.org/wiki/Hardware_description_language
100 https://en.wikipedia.org/wiki/Application-Specific_Integrated_Circuit
101 https://en.wikipedia.org/wiki/Logic_block
102 https://en.wikipedia.org/wiki/Combinational_logic
103 https://en.wikipedia.org/wiki/XOR_gate
104 https://en.wikipedia.org/wiki/Virtualization
105 https://en.wikipedia.org/wiki/Software-defined_radio
106 https://en.wikipedia.org/wiki/Virtual_machine

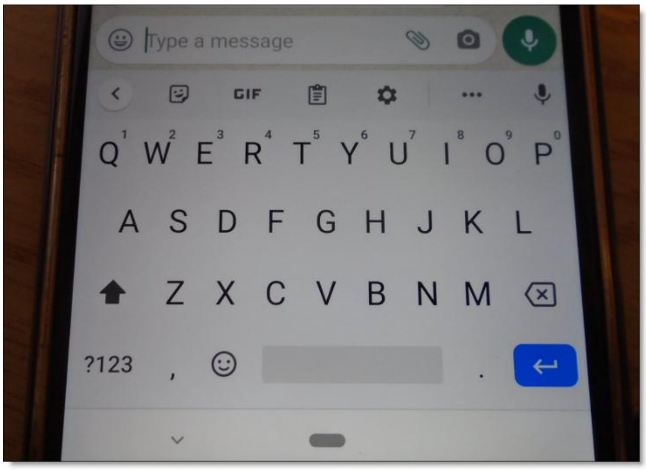

Figure 18 - A virtual keyboard on a smartphone...

Forth[107] uses a simple virtual machine although I prefer to call it a "virtual processor". I have spent many years incorporating assembly[108] language programs within a multi-tasking[109] Forth environment to drive various kinds of hardware...

See: ***http://baremetal.engineer/baremetal.software.engineer.pdf*** for much more about my software activities on hardware...

A much more complex virtualisation project is BT's "21-CN"[110] project to virtualise their 100-year-old telephone system onto their own internal TCP/IP[111] network

"The 21st Century Network (21CN) programme is the data and voice network transformation project, under way since 2004, of the UK telecommunications company BT Group plc. It is intended to move BT's telephone network from the AXE/System X Public Switched Telephone Network (PSTN) to an Internet Protocol (IP) system"

Finally...

All the technologies discussed in this document possess a very important property: ***Emergence***[112]

"*In philosophy, systems theory, science, and art, emergence occurs when an entity is observed to have properties its parts do not have on their own. These properties or behaviours emerge only when the parts interact in a wider whole. For example, smooth forward motion emerges when a bicycle and its rider interoperate, but neither part can produce the behaviour on their own*"

So the discovery of wireless and the invention of the thermionic tube, spawned all the modern communications technology we have today...

More recently, the invention of the HTTP[113] protocol which underlies the ***World Wide Web***[114], by ***Sir Tim Berners-Lee***[115] is another example of an ***emergent property*** that is so powerful it has driven a great deal of global social change and created whole new industries...

---

107 https://en.wikipedia.org/wiki/Forth_(programming_language)
108 https://en.wikipedia.org/wiki/Assembly_language
109 https://en.wikipedia.org/wiki/Computer_multitasking
110 https://en.wikipedia.org/wiki/BT_21CN
111 https://en.wikipedia.org/wiki/Internet_protocol_suite
112 https://en.wikipedia.org/wiki/Emergence#Emergent_properties_and_processes
113 https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
114 https://en.wikipedia.org/wiki/World_Wide_Web
115 https://en.wikipedia.org/wiki/Tim_Berners-Lee