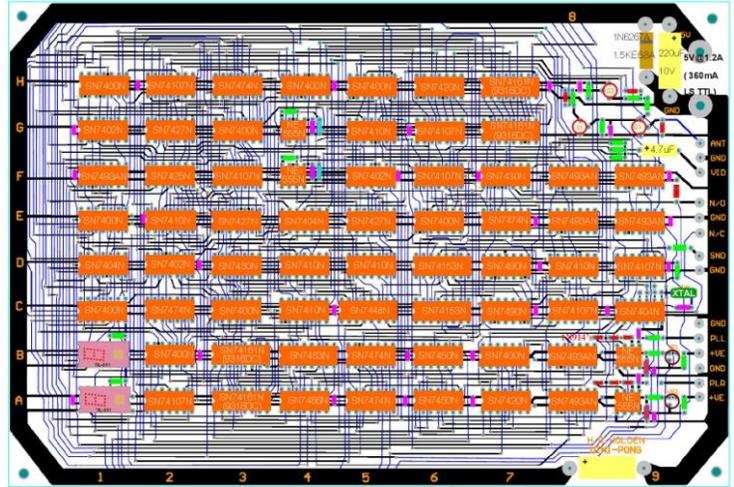


MINI – ARCADE PONG – 6 Bugs Safe.

Dr. H. Holden. Feb. 2016. (up-dated March 2016)



The purpose of this project was to design and build the best Arcade Pong PCB ever made and to make a fun home Pong game. The original boards are large and becoming rarer and for History's sake, probably should not be modified.

Also to have a new pcb with a compact format, yet still keeping the IC's on an original grid configuration. This way, when an IC is referred to say at location C5, it matches up with Atari's original schematic. Also the design must eliminate the six bugs present in the original design and provide some simple on board diagnostics with two TIL311 hexadecimal displays.

Some new computer designed pcb's to make Pong “reborn” have the IC's in a completely different configuration and they are also larger in size than the one presented here.

Firstly the presence of “bugs” in the original design did not detract at all from the brilliance and creativity of the original circuit design from 1972. And for a circuit of such complexity, needing to get to market quickly, some unresolved issues would be expected.

Why is Arcade Pong worthy of such attention and effort ?

Because **Arcade** Pong is the most sophisticated and brilliant version of Pong ever created. It completely out classes any coded or software based Pong created so far and also out classes any hardware based Pong on a single LSI chip.

See the article:

http://worldphaco.com/uploads/LAWN_TENNIS.pdf

(The article cited above contains a lot more detail on this masterpiece created by Mr. Allan Alcorn at Atari in 1972)

Pong functionality:

The Paddle architecture alone in the arcade version of Pong was more complex than any home Pong version giving 42 possible states of motion of the ball.

The ball motion “vector” (to think of ball motion in analog terms) is formed from combined horizontal and vertical motion components.

On the vertical motion side of things there are three up and three down ball motion components. There is also a state of *zero* vertical motion, leaving a horizontal motion component only in that condition.

There are three horizontal motion components too, determined by the HIT counter, which combine with the vertical motion components to produce an overall perceived motion vector for the ball that a player observes on the video screen. Although this description is an “analog” one and the ball motions are generated digitally, it is how a player perceives the motion due to the persistence of the phosphor on the CRT screen and other factors.

In total there are 3 horizontal and 3 vertical components which combine to produce a motion vector. And this occurs in 4 screen quadrants because the ball could be travelling up or down, or left or right. So this gives $4(\text{quadrants}) \times 3 \times 3 = 36$ states of motion or ball “velocity vectors”.

However, as noted, there are three additional states of motion, or motion components, which have zero vertical velocity. These are the horizontal states of motion on their own, determined by the HIT counter during game play. These are left or right motion alone. This adds another $3 \times 2 = 6$ states of possible ball motion during game play to the 36 other states. The grand total is 42 states of possible perceived motion of the ball, or 42 unique ball velocity vectors. This is more an enough to convince the player that the game is functioning in a smooth and analog fashion.

The genius of the game was that the vertical components of ball motion were determined by where on the paddle the ball made contact. When this interaction occurs, data relating to the condition is clocked into the vertical velocity encoder circuitry, which is one of the *many* very

clever sub circuits in the game. The further away from the paddle center, that the ball and paddle interact, encodes a higher vertical velocity. The upper half of the paddle encoded for increasing vertical velocity upwards, while the lower half of the paddle encoded for increasing downward motion. The paddle centre encoded for zero vertical velocity. Also the horizontal motion would speed up in a volley when there were no misses by either player. After 4 consecutive hits the horizontal component of ball velocity would increase. By 12 hits with no misses the horizontal velocity component sped up yet again.

These ball motion features combined with the sound effects and score keeping made for a very realistic version of Pong that out classes all other versions.

Again all of these sub circuits were cleverly implemented, for example using a single standard binary to 7 segment display encoder IC, then multiplexing that into the video for the on screen score displays for both players.

In addition the screen size of the player paddles and the size of the score segments in the arcade game were a well proportioned use of the video display area and appeared much better in Arcade Pong than in some of the home Pong versions where the scores and paddles (bats) appeared larger. Clearly there were some compromises made when this arcade circuit of around 66 TTL IC's was miniaturized down into a single LSI integrated circuit for home Pong use.

Computer Pong ? Every "coded" or computer version of Pong I have trialed so far is a very poor facsimile of Arcade Pong in my view. However, only recently has a full description of Arcade Pong's architecture been available (article cited above) for software coders to attempt to emulate. So it might be that better versions could become available in the future. An attempt has been made to build the entire circuit with an FPGA, however I'm not sure if the designer in Germany has completed this, or if it worked out well.

Problems with the original Arcade Pong design and the original PCB design:

Unfortunately, in the original arcade Pong Syzygy E pcb, there was a "bug" that limited these 42 ball motion possibilities. This bug, one of six, is now called "The Ghost in the Machine Bug". This is a reference to a remark in the *I Robot* movie, as is the title of this article "6 bugs safe" or "3 laws safe".

With no negative reflection on the genius of the original design implied here, the arcade PCB itself was physically very large and cumbersome in size. Of course there was plenty of space inside an arcade game cabinet, so it hardly mattered.

Of note too, I could say that out of all the circuits I have seen after lifetime of interest in Electronics, that have impressed me more than any others, Pong is up there in the **top two**. One reason for this is the combination of technical creativity and fun, making the best out of the current technology of the time, seldom seen together, all wrapped up in one design.

(The other circuit in the top two is a Television horizontal deflection circuit for electrostatic deflection which achieves the seemingly impossible; it creates two *linear* anti-phase saw-tooth voltages with a rapid fly-back, each of 450V peak to peak, at a frequency of 15,750 Hz, *running from a 250V power supply rail alone* (aside from the 6.3v tube heater), but it only has one active device, a single triode. Can you imagine how that was done?)

Mini – Pong – debugged or “six bugs safe”

This article describes the design of a variant of Arcade Pong with full bug corrections on a miniaturized pcb format of only 245 x 165mm yet still using the original 74 series DIL TTL IC's. (LS) TTL IC's can also be used to reduce the power consumption to around 360mA @ 5v rather than about 1.2A with standard TTL. However there is something quite wonderful about the original power hungry 74 series TTL's. This is the sort of robust TTL technology which comprised the computers in Apollo Spacecraft. They are very trustworthy and very robust IC's.

It has been possible to design a much smaller version than the original arcade pcb by altering the pcb track design & artwork completely and running the IC's power rails down the long axis of the IC's, unlike the original design which had them perpendicular to the IC's long axis. Also the new pcb can be powered from any common garden modern 5V switch-mode power supply so space was also saved that way too by not having the power supply components on the pcb as they were, on the original arcade pcb version.

The relative pcb sizes are shown below:



This new PCB has been designed by the author "by hand" without any form of pcb design tools/software or "track router" to place tracks & vias. It was initially drawn as just a .png image. Later this was transferred into Altium to create a file for pcb manufacture.

The original arcade pcb was also designed by hand.

I worked on this design for about 2 years on and off to place the tracks and vias. It has achieved a good result, possibly as good as or better than a computer aided design. The board has also been kept to a simple double layer format to keep the expense down.

Some "on board diagnostics" have been added by the way of two TIL311 hexadecimal displays. One display monitors the 4 bit data from the output of the vertical velocity encoder and the other the 4 bit data from the Hit counter. This is useful to see that everything is working normally, but most, perhaps not all faults, if present, are usually evident in game play.

In addition the six bugs have been removed from the design by incorporating unused gates and other circuit changes.

THE SIX CORRECTED BUGS:

- 1) The Ghost in the Machine Bug.
- 2) The Ball Trapped in Blanking Bug.
- 3) The Paddle Range Bug.
- 4) The Screen Video Horizontal Image Displacement Bug.
- 5) The Weak Net Bug.
- 6) The Ball Monitor Sync Disturbance Bug.

1) The Ghost in the Machine Bug.

This bug which "lurked in the machine" was not found for over 40 years. It came about due to a mistake in the original pcb design. Due to the fact that pin 10 and pin 1 of IC A6 had reversed labels on Atari's original schematic, the original pcb designer copied this. This meant that the least significant bit of paddle data, processed by the vertical velocity encoder, was switched between the two players. The result (described in detail in the article cited above) meant that one player's paddle could influence the other player paddle's interaction with the ball. Also, on top of that, the number of possible ball motion states became reduced. It produced a "spooky" and unpredictable effect in that sometimes the ball would bounce from an unexpected angle from the paddle, depending on where the other player's paddle was positioned on the screen.

Bug fixed by connecting the tracks correctly to IC 7450 on location A6.

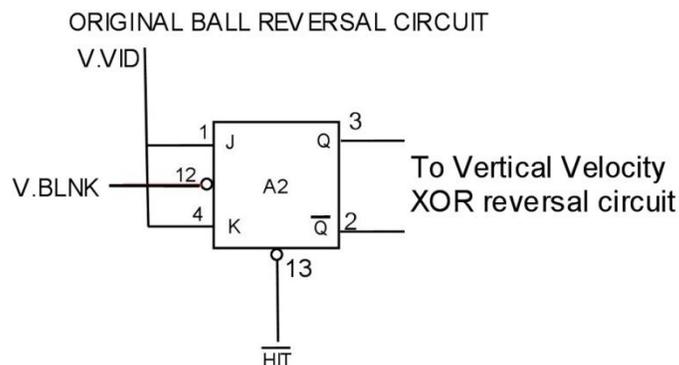
Of note, this bug is not present in Arcade Pong Doubles, only the original Pong E Syzygy boards.

2) Ball Trapped in Banking Bug:

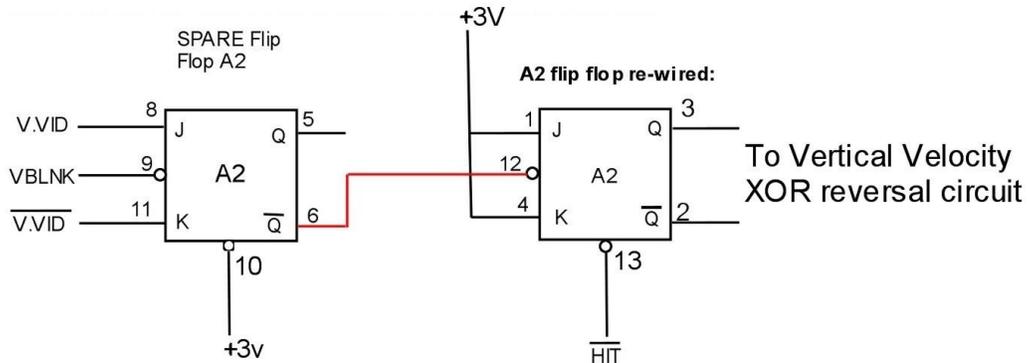
This is a very complex and an infrequently presenting bug and it is described in detail in the above cited article. In effect it represents a "logic race" that the game cannot escape from if it gets into it. The effect is that the ball can become "trapped" inside the vertical blanking interval instead of inside the active raster scan time interval. It could occur for example with the paddle range potentiometer was out of adjustment (or the paddle range increased see below) and the ball oscillates momentarily between the paddle edge and vertical blanking. The ball would then become "trapped" synchronized with vertical blanking and be seen in a vertically elongated form (if the vertical blanking area was visible on the monitor screen) and to be moving horizontally to and fro in the vertical blanking interval, unable to escape.

This can be corrected by turning the original game on and off usually. It was an infrequent bug to appear, but when it did the game was "locked up". Very rarely the bug would appear at switch on of the arcade game, disabling the game until switch off & on again giving a hard reset. The cure is to deploy an unused 74107 Flip Flop in IC package A2:

Firstly the original sub circuit:



Modified circuit to eliminate bug:



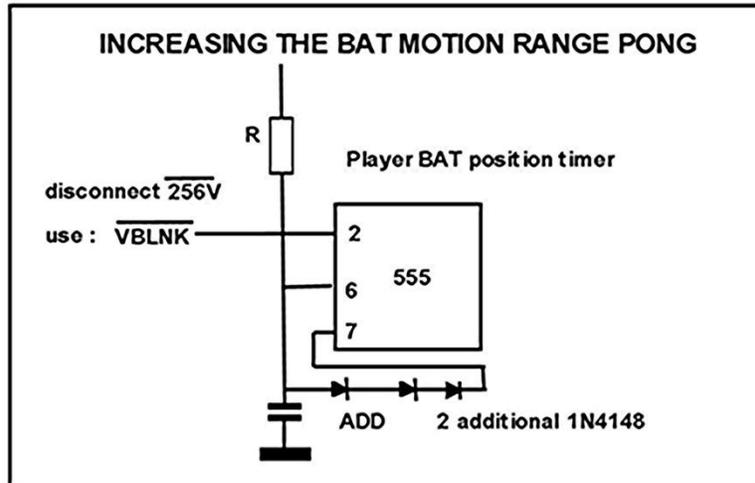
Flip flop A2 on the right above is used to reverse the ball's vertical velocity after the ball signal V.VID and vertical blanking VBLNK become coincident. The addition of the flip flop (unused half of A2 on the left) and the rewiring of A2 on the right solves the problem and allows enough time for the ball to always escape the vertical blanking interval without remaining trapped there. Also wiring the right hand A2 flip flop with both J and K tied high makes its toggling very reliable. For the details of this complex problem and why it is solved by this modification, see the article cited above.

3) Paddle Range Bug:

The original paddle range was limited. This had the effect of allowing the ball to travel above a player's paddle no matter how hard they rotate the control against its stop. This made some players "angry" as they knew they could have hit the ball if the control had allowed them. Though it was pointed out by the designer that this "feature" always meant the game would finish, as two experts could possibly play it for a long time. But even for good players, someone usually misses especially when the ball's horizontal velocity is at maximum after 12 consecutive hits and the player has a beer or a hamburger in the other hand and they are chatting to friends.

The modification here allows full range of the paddles close to the vertical blanking intervals so that the ball cannot get around (over or under) the paddle. It does not cause any problems, provided the *ball trapped in blanking bug fix*, described above, is present. If not the extended

paddle range fix can more likely push the ball into blanking where it can get trapped. Three rather than the original one 1N4148's are required,



(Note: If building TTL Pong circuit do not attempt to use any other variant of the 555 timer IC than the NE555N, preferably use an early Signetics unit. Many other 555 types, whether CMOS versions, or the NE555P or the LM555's have small differences that show up in the timing and generation of the Paddle image with the paddle appearing early or late or non-linear control, that would never show up in 1000's of other timer circuits using 555's).

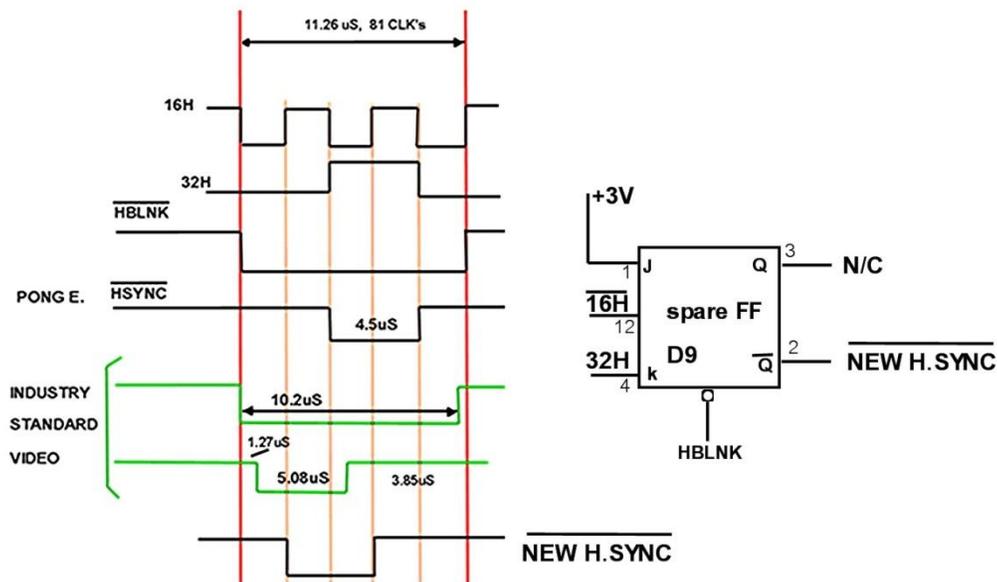
4) Screen Video Horizontal Image Displacement Bug:

The horizontal sync pulse placement, with respect to horizontal blanking, was placed in a position such that when the horizontal hold setting is properly adjusted on the monitor, the screen image, especially the line of the Net, is seen displaced to the left .

To improve this, the H. Hold control on the monitor can be turned a little. This is because most monitors have a horizontal AFC circuit with a DC control to their local H scan oscillator, so any offset in the sync timing with respect to the video(image) signal causes a horizontal phase shift or displacement of the position of the video image on the scanning raster.

However when the H. Hold control is turned to center the screen image (have the net about in the middle of the screen or just to the left of it) the monitor can sometimes lose horizontal picture lock when first turned on from cold as the AFC circuit goes out of “capture range”.

Therefore the sync pulse is better repositioned, within the blanking time, to be closer to industry video standards (or typical NTSC video). As explained in the article cited above there are a number of ways to do this with spare gates & FF's available in the game. The method below is simple as it just uses one spare flip flop:



The (not) NEWH.SYNC signal replaces the (not) HSYNC signal that feeds into pin 12 of the sync pulse mixing XOR gate on location A4. Although this arrangement doesn't give exactly standard sync pulse to blanking relations, it is very close and the picture centering is much better on the monitor. The (not)16H signal is available from pin 4 of the IC at location E4 and the 32H from pin 9 of IC at location F9 and HBLNK is available at pin 8, 12 and 4 of the IC at location H5.

(Of note, this bug was fixed in Atari Pong Doubles, but that particular circuit fix required 3 extra gates as well as the flip flop to achieve exactly the same result as the single flip flop alone shown above).

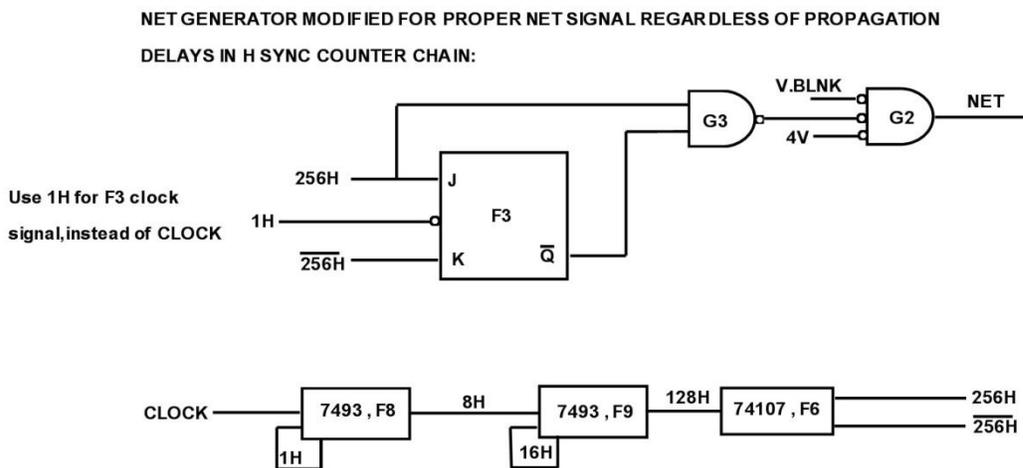
Also I found that if a modification is made to place the sync pulse almost identically to industry standard (or NTSC) video, and center the net line almost perfectly, then the images of the scores appear to be displaced a little to the right. So the better average horizontal picture

position appears with the net still just a small amount left displaced and the score image a little right displaced when the H. Hold control on the monitor is correctly set. Some monitors (very few) have an internal H. phasing control so that the image horizontal picture centering can be adjusted after the H hold control is correctly set.

5) The Weak Net Bug:

This bug occurs due to the propagation delays in the two 7493 counters in the horizontal sync generator. Cumulative delays in this "ripple counter" system can upset the timing in the generation of the net signal. When specimens of the 7493 counter IC had shorter propagation delays in each flip flop, typically the 7493AN counter chips, a timing error developed in the drive to the flip flop *pulse synchronizer circuit* F3 and G3 that generated the net pulses.

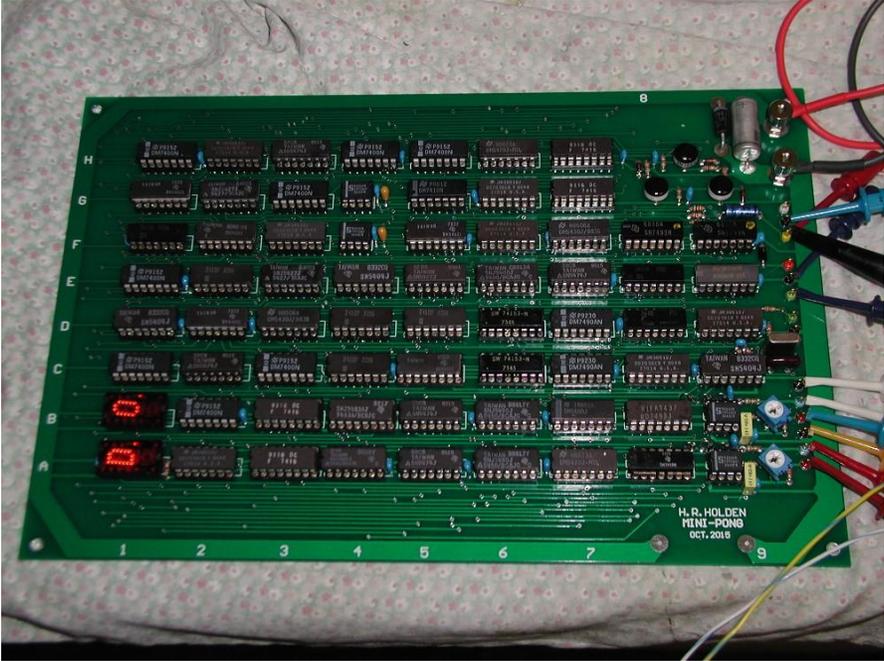
The result is that a very short net pulse or a weak appearing thin or faint looking net on the screen image. The fix is to clock the flip flop at location F3 pin 9 with the 1H signal rather than the clock signal. This way the particular timing errors or differences in 7493 IC's have no effect of note on the width of the net pulse. This modification is shown in the diagram below.



HORIZONTAL SYNC CHAIN: A variable propagation delay in the 7493 IC's (all still in spec) results in a + or -20nS advance or delay of rising edge of 256H with respect to clock falling low. In the case where the rising edge of 256H precedes the clock falling low, this results in a weak(short) net pulse.

The earlier 7493 counter IC's had about 18nS delay per internal flip flop and there being 8 FF's in two 7493's together, yielding about 144nS delay, plus about 16nS for the 74107 flip flop. The total delay being about 160nS between 256H going high and the clock pulse going low. The

The photo below shows a prototype pcb being tested prior to the final design:



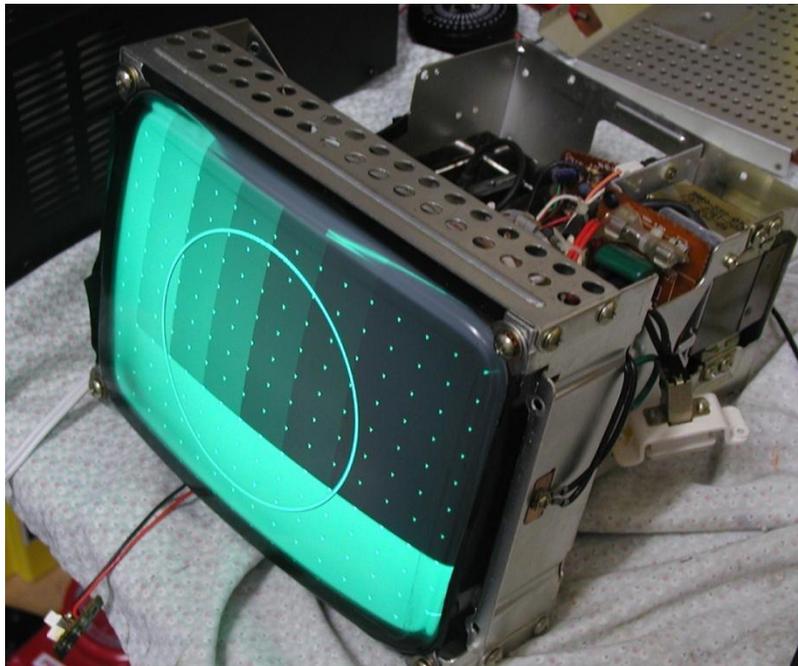
The photo below shows testing with a small CRT based VDU:



I find vintage series 74 TTL IC's very reliable. Perhaps one reason is that they are very robust to electrostatic discharge damage. Despite some remarks that back in the 1970's one in 100 or more might be faulty. In any event assembling the pcb above, from these new old stock parts which are over 35, some 43 years old, I powered the pcb and it worked perfectly first time. I think that says a lot about the quality and robust nature of the IC's.

The next step was to fit this pcb to a housing and pair it up with a suitable monochrome monitor. As it turns out a suitable monitor for this job, for a "mini-Pong" is the small monitor used with the vintage Apple IIc computer

The following image shows an Apple IIc computer monitor chassis undergoing adjustments and repairs & modification to run from a 12V DC input rather than mains power. This way the monitor can be powered from the same psu inside the Pong console. Also this is convenient because the Apple IIc normally runs from 115V AC and we have 230V in Australia:



Apple IIc monitors generally come with a green phosphor (P31) CRT, however I changed this for a white phosphor CRT, since monitors used with PONG were generally TV monitors or modified TV sets as used in the original arcade Pong and they generally had white (P4) phosphor CRT's.

The completed Pong PCB and power supply (compact switch-mode psu) and a speaker with a small associated amplifier (The Champ Amplifier module kit based on a LM386 from Silicon

Chip) were placed in a high quality Hammond painted aluminum enclosure for the final result. An illuminated push to start button was used:



The Apple IIc monitor also has a handy stand that elevates it to a good viewing level.

#####