

The HP-35 "Bluebird" tester

Fred K. Beckhusen
July 2016

The HP-35 calculator was a huge sales success in 1972, the year I graduated from high school. There is a lot of information online today about this marvelous machine, but very little about how it was manufactured, or how it was tested. I wish I had know back then what I know today, because I worked on the HP-35 for a long time, without even knowing about it. So here it is, 40 years later, and I am writing about the HP-35 chip set and architecture, and the Bluebird tester we tested them with.

This photo from the 1973 Mostek data book shows our first computerized IC tester. We called it the "Bluebird" tester. It also shows the PDP-11 computer. The Bluebird was painted primary blue. This horrible color eventually was put on every piece of gear we had.

The Bluebird was originally designed to test the HP-35 "West Coast" calculator chips. The "West Coast" was because HP was on the west coast, while we were in Carrollton, near Dallas. The first machine went live in February 1972.

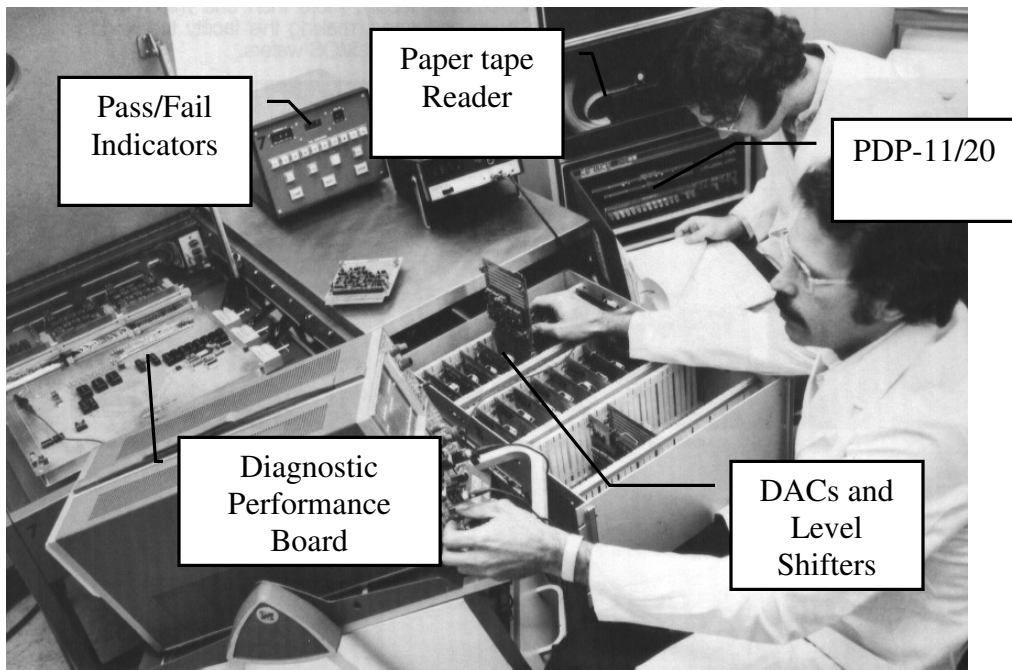


Figure 1 - Designers John Lundy and Lee Perry working on level translator card

There are many interesting things about this tester and the IC's which made up the HP-35 which I'll go over in more detail. The Bluebird was used for all of the HP family of HP-35, HP-45 and other calculator parts, and for "random logic chips", which at Mostek meant "not RAM". RAM chips required much more extensive tests. Bluebirds were made to be general-purpose and tested almost all ICs (except RAMS) at Mostek for many years. About 20 of these were built during my stay there.

This custom test system was the first computer-controlled IC test system at Mostek. All other testers before then were digital state machines we called 'dedicated testers'. which were controlled by logic gates, counters, and in the case of some calculators, by an endless loop paper tape reader. We made small DRAMS, too, and tested them with dedicated testers with rows of potentiometers to set up the critical time. None of the earlier dedicated testers had a CPU.

Bluebird Specification	Value
Clock rate	200 kHz
Test vector to DUT	24 bits
Test vector to CPU	16 bits
Pattern memory	via DMA
DC precision	12 bits

The original reason we built the Bluebird was to test a chip set we designed for Hewlett Packard, for the HP-35 calculator . I'll explain more about the HP-35 chipset next, as you need to know about the chips and what needs testing to understand this early IC tester.

The HP-35 was the first hand-held, battery powered scientific calculator. It was a huge success. HP originally thought that 50,000 units could be sold, but they all were sold in just the first two months. G.E. bought 20,000 units just for internal use!

The HP-35 sold for \$395.00 in 1972, the year I graduated from high school. Over 300,000 HP-35s were sold by 1975. I remember reading about it in a magazine around 1972, and was awestruck by how powerful it was. I had no idea as a young kid still in high school that just a few years later I would be working on them at an intimate level at the very company that designed and made the IC's.

To put this in perspective, a \$395 calculator was the equivalent of \$2,777.00 in US dollars as of 2016 - equivalent to \$840,000,000.00 in today's dollars in just three years. And that is just for one model! The chip sets and derivatives of them were used by the HP company for many years. The HP contract was the largest contract Mostek received for many years. It was not exceeded until we developed the multiplexed DRAM, which took over the world.

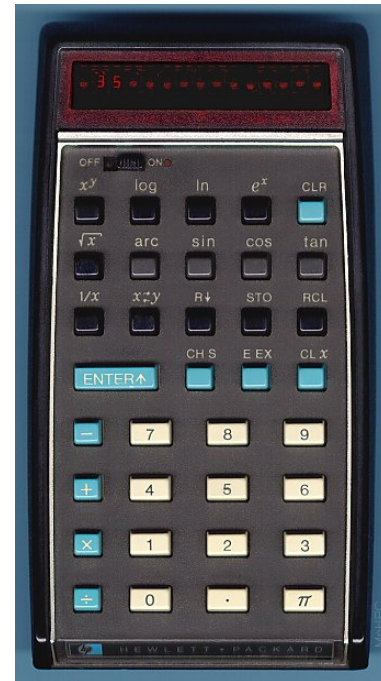


Figure 2 first production HP-35
Photo credit HP Museum¹

The Mostek contract required the custom design of three PMOS IC's, each with roughly 6,000 transistors¹. These were the MK6022, MK6023, MK6024 ROMs which differed only in the pattern loaded in them, a MK6021 Timing and Control IC, and a MK6020 Arithmetic and Register chip². Together with the 1820-0853 anode driver and system clock chip, three 1820-0854 LED drivers, three LED 5-digit LED chips, and a 1820-0855 clock generator, they made a small ROM- programmed computer. The HP-35 used 3 ROM chips, with 768 10-bit instructions each. Each chip handled a 'page' of memory, and pages were organized in banks of four, called 'quads'. So the HP-35 had 3/4 quad. Later HP calculators such as the HP-80 used as many as seven chips, and the chips were quadrupled in size to hold more data. The chip set could address 8 ROMs, but never used more than 5, because the chips were quadrupled up in size, thus leaving only a single spare memory bank of 256 instructions.

Each chip had features that were about 10 microns (0.0004 inch) on a side, enormous compared with modern versions, about 2000 of which would fit inside each one.³

HP series of calculators and memory sizes:

Model	Instructions	Quads *
HP 35	768 instructions	3/4 quad (3 pages of 256 instructions each)
HP 45	2048 instructions	2 quads
HP 80	2048 instructions	2 quads
HP 65	3072 instructions	3 quads
HP 70	2048 instructions	2 quads
HP 55	3072 instructions	3 quads
HP 19C	5120 instructions	5 quads
HP 10	1536 instructions	1 1/2 quads
HP 21	1024 instructions	1 quad
HP 25	2048 instructions	2 quads
HP 22	2048 instructions	2 quads
HP 91	3072 instructions	3 quads
HP 29C	4096 instructions	4 quads
HP 27	3072 instructions	3 quads
HP 25C	2048 instructions	2 quads
HP 67	5120 instructions	5 quads
HP 97	6144 instructions	6 quads
HP 92	6144 instructions	6 quads

* 1 quad = 1024 instructions or 4 pages of 256 instructions each.

¹ <https://web.archive.org/web/20040203113550/http://www.vcalc.net/hp-35.htm>

² <http://www.hpmuseum.org/hp-35.htm>

³ <http://pmonta.com/calculators/hp-35/>

I knew of several engineers at Mostek who unsoldered their HP-35 ROMS and replaced them with HP-45 chips as an upgrade. This was possible because the HP-35 had two Quad chips, and the only other difference was some extra keys were added. As a technician who knew how to solder, I got to take one designer's precious calculator apart, desolder the ROMS, and solder new ones in. This was quite a thrill for a 20 year old!



This test data that came from the CPU obviously needed to vary from part to part. Eventually we made 24 different parts for 13 different HP calculators. For example, the HP-35 and HP-45 differed only in the mask patterns in the ROMS and in a extra row of keys.

<i>Type</i>	<i>Year</i>	<i>Function</i>	<i>Calculator</i>	<i>Comments</i>
HP65-CPU	1973	Programmable Scientific	HP-65	Hybrid CPU-module
MK6111	1973	ROM	HP-65	ROM#1
MK6112	1973	ROM	HP-65	ROM#2
MK6113	1973	ROM	HP-65	ROM#3
MK6020	1972	Arithmetic Chip	HP-35, HP-45, HP-55, HP-70	
MK6021	1972	Control and Timing Chip	HP-35	
MK6022	1972	ROM 256*10 bits	HP-35	ROM#1

MK6023	1972	ROM 256*10 bits	HP-35	ROM#2
MK6024	1972	ROM 256*10 bits	HP-35	ROM#3
MK6036	1973	RAM 10*56 bits	HP-45, HP-55, HP-70	
MK6134	1973	ROM 1024*10 bits	HP-55	
MK6215	1974	Arithmetic Chip	HP-22, HP-25, HP-25C	
MK6216	1974	Arithmetic Chip	HP-29C, HP95C, HP-97	
MK6217	1974	Arithmetic Chip	HP-21	
MK6220	1974	RAM 16*56 bits	HP-22, HP-25	
MK6250	1975	Card Reader Control	HP-67, HP-97	
MK60038	1975	ROM	HP-67	added Anode drivers
MK60126	1975	ROM	HP-25, HP-25C	
MK60142	1975	ROM	HP-19C	
MK60171	1975	ROM/RAM	HP-67, HP-97	
MK60172	1975	ROM/RAM	HP-67, HP-97	
MK60175	1975	ROM/RAM	HP-67	
MK60176	1975	ROM/RAM	HP-67	
MK60185	1975	ROM/RAM	HP-19C, HP-29C	

Src: http://www.datamath.org/Mostek_IC.htm

It was obvious from the start that a programmable tester was needed that could be rapidly updated to test different ROMS. There were unusual requirements compared to the way microcomputers and microprocessors work today. In order to keep power dissipation low, the HP-35 ran at only 200Khz, and was a serial computer, not parallel like today's chips. The digit-serial, bit-serial organization reduced package size, cost and also increased reliability. This set a requirement that the new tester had to run at 200 KHz speeds and be programmable. As a result., one of the very first PDP-11 systems was chosen. 200 kHz was easily within the range of a DMA cycle over the PDP-11's Unibus system. Unibus could run at megahertz speeds⁴.

Because the HP family was well thought out in advance, and Mostek needed better testers, a general purpose re-programmable tester was designed - the Bluebird.

This programmability was accomplished by hand-keying in a bootstrap loader in the PDP-11's switch register, loading an "absolute loader" tape with the bootstrap, which loaded the test program tape. That PDP-11/20 system in the photo was one of the very first PDP-11's. They came with 4K of core memory. We needed more speed and more RAM, so Mostek designed our own RAM memory board. And every time we had a power failure, the other technicians and I would run around as fast as we could, re-keyed in the bootstrap loader and loaded all the tapes again. This seemed to happen far too often very early on Sunday mornings.

I keyed in that bootstrap so often, I still remember the beginning of the bootstrap, 42 years later.

```
012737      move.w #1, #164162 // Put 1 in the Bluebird Online register
000001
164162
000002      Wait for Interrupt
....
```

⁴ <http://textfiles.com/bitsavers/pdf/dec/unibus/UnibusSpec1979.pdf>

At this time the Bluebird would signal an interrupt over the Unibus cable.

The Unibus was an 120 ohm impedance-controlled flat cable that memory mapped the Bluebird into the PDP-11's 32K word address space⁵. The Unibus was much like any computer bus of today. It ran through the PDP-11, but also was extendable by a long, flat impedance controlled cable to external devices. The closest analogy I know to the Unibus architecture is the MK68000 chipset, which is not a surprise as the 68K was largely inspired by the PDP-11. The Unibus had an 18-bit address and 16 bits of data, plus two parity bits, one for each byte. More importantly, there was a Master Sync that would clock the addresses, much like the 68K Address Strobe pin, and going the other way was Slave Sync, just like DTACK on the 68K. This let Mostek designers create registers inside the Bluebird. It also had a Non Processor Request (DMA) pin for requesting and a NPG (Bus Grant) pin for granting a DMA cycle, and 4 interrupts. The Bluebird used these signals and the Unibus to DMA signals to and from the HP35 ICs as part of the test.

The Bluebird tester had two drawers of digital logic. The bottom drawer held the circuits for interfacing to the Unibus, and via it, access to the PDP-11 memory. The data for stimulating the parts came from the PDP-11 memory via DMA over the Unibus.



Figure 3 Unibus cable

I had worked mostly on dedicated testers in my first few months at Mostek. On a dedicated tester, you triggered an oscilloscope probe with the "fail lamp" driver and used a second probe to look at the IC under test pins, the level translators, comparators, and so on until you traced your way back to the test inputs to try to find the problem. It could be a pin loose in a socket, or a blown circuit in a level translator. The triggered scope would show you the exact instant the test failed.

Photo credit [Wikimedia Commons](https://commons.wikimedia.org/wiki/File:Unibus_cable.jpg), the free media repository [Creative Commons Attribution-Share Alike 3.0 Unported](https://creativecommons.org/licenses/by-sa/3.0/)

So when I started working on Bluebirds, I was in for a surprise. The first time I tried this "trigger on the fail lamp" technique on the Bluebird, I connected the scope to the latch, and traced my way backwards. I ended up in the lower digital drawer, and realized I was tracing a signal all the way back to the Unibus, chasing after the Master Sync signal that indicated the program had just executed a move opcode to write a data bit into a register in the Bluebird that toggled the fail lamp on!

Imagine looking at something like this circuit from the Dec Unibus manual, and realizing this was not the right direction to find out what was wrong with the chip in the test socket!

⁵ <http://html.alldatasheet.com/html-pdf/92680/NSC/DS8640N/46/1/DS8640N.html>

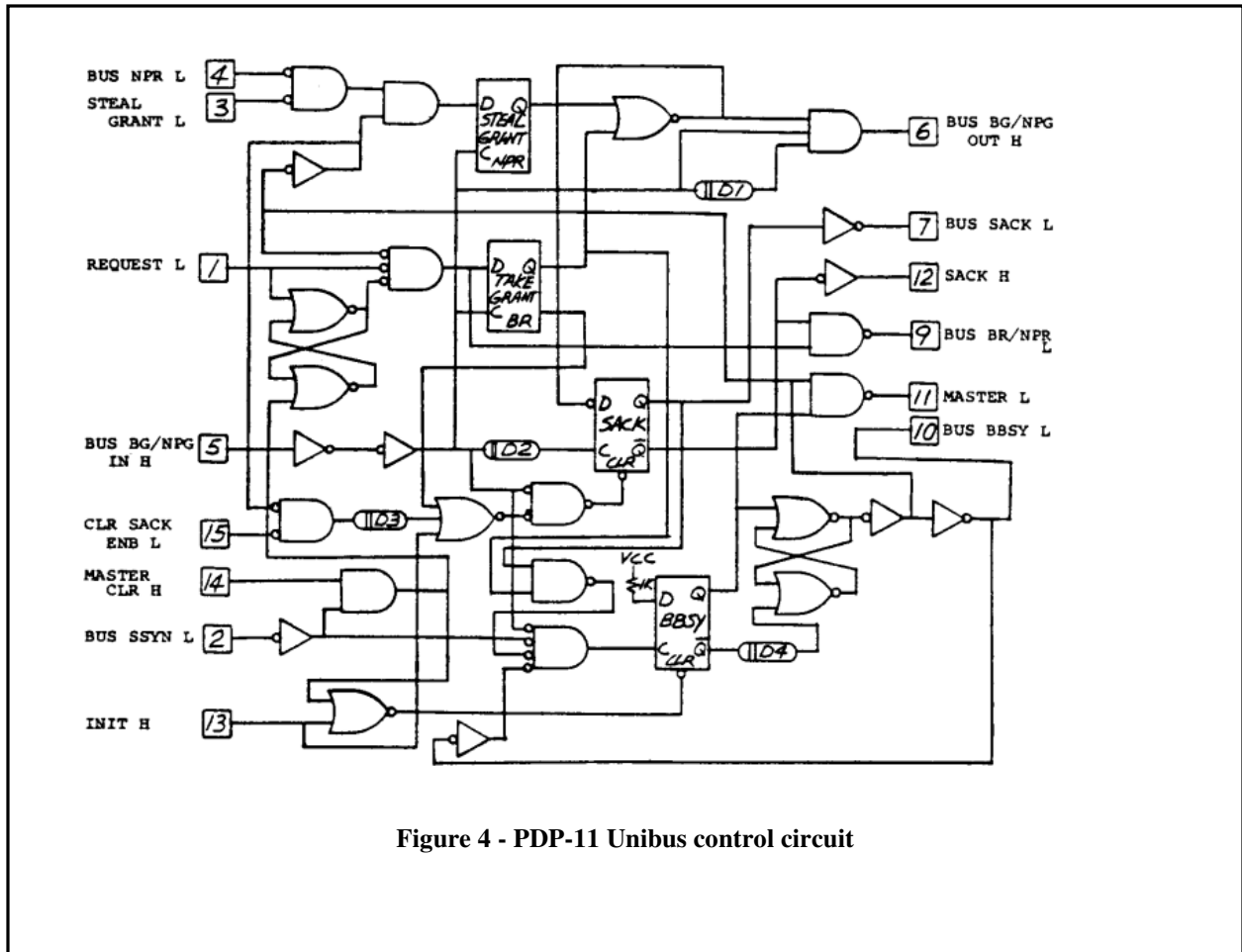


Figure 4 - PDP-11 Unibus control circuit

I realized that night that learning the programming of the CPU was needed to figure out what was wrong. This led later to my career as a designer of and programmer for Unix systems and many microprocessors. Unix originally ran on Dec PDP hardware, and also was used back then on the 68000 microcomputer that Mostek also made.

AMI was also selected as a vendor and designed a similar set of three IC's. Here is a comparison of the size and power dissipation of the chips from Mostek versus AMI. The goal was 250 mw power dissipation. As you can see the Mostek IC was considerably lower in power. This mean the batteries could easily run several times the 4 hour requirement.

The CPU instruction word was 10 bits, but to save on chip size, the parts were based on serial data. The 10 bits were clocked out, one at a time, and all calculations were done by a half-adder inside the Arithmetic and Register chip. This feature of the chip required a word select line from the Control and Timing chip to signal to the A&R chip when the appropriate digit was available. There was also a SYNC pin that signaled the beginning of an instruction word. This signal went to a special card in the Bluebird tester, that we called the "West Coast Sync" board⁶.

	CIRCUIT TYPE	AMI	Mostek
MK6022,23, & 24	Instruction ROM (256X10):	116 x 108	126 x 134
MK6020	Arithmetic and Register:	138 x 167	156 x 200
MK6021	Timing and Control:	163 x 169	150 x 212
	Power Used:	100-120 mw	73 mw
	Size:	63,121 sq mils	79,882 sq mils

Only a few people at Mostek had any idea these chips were for a HP calculator. It was a carefully kept secret for a long time. The code name was used whenever they could use it - even in the schematics for this tester. I have a vivid memory of working on a Bluebird that was failing a test, all day long. I was stuck deep inside the digital logic on the "WC Sync" board inside the bottom drawer of logic, getting nowhere. I asked several engineers what it was about, and they told me to just fix it. I finally found a sympathetic engineer who realized I had a "need to know", so he told me the chips were for the HP-35, and how the "West Coast Sync" worked. Needless to say, I was a bit shocked. The HP-35 was famous!

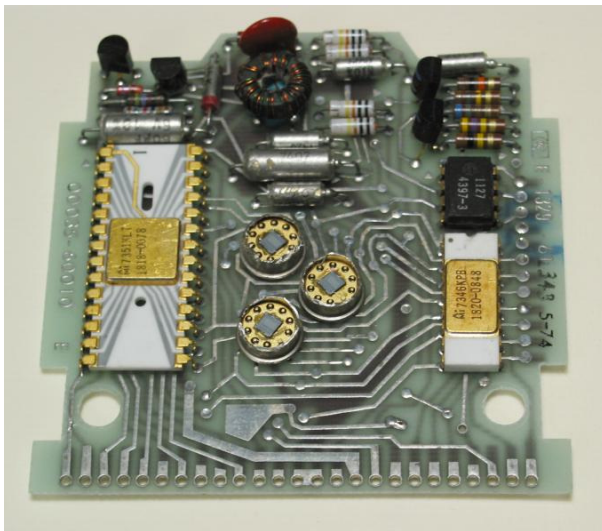


Figure 5 - HP 35 calculator with ROM lids removed

⁶ <http://www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/articles.cgi?read=385>

From the calculator's perspective, a timing circuit on each ROM is synchronized to the trailing edge of the SYNC signal issued by the Control & Timing chip when the calculator is turned on. This circuit then keeps the ROM chip running synchronously with the rest of the system. The Sync signal is found in this timing diagram from the HP-45, which was the same as the HP-35 except for extra keys and different, 4X the size ROMS.

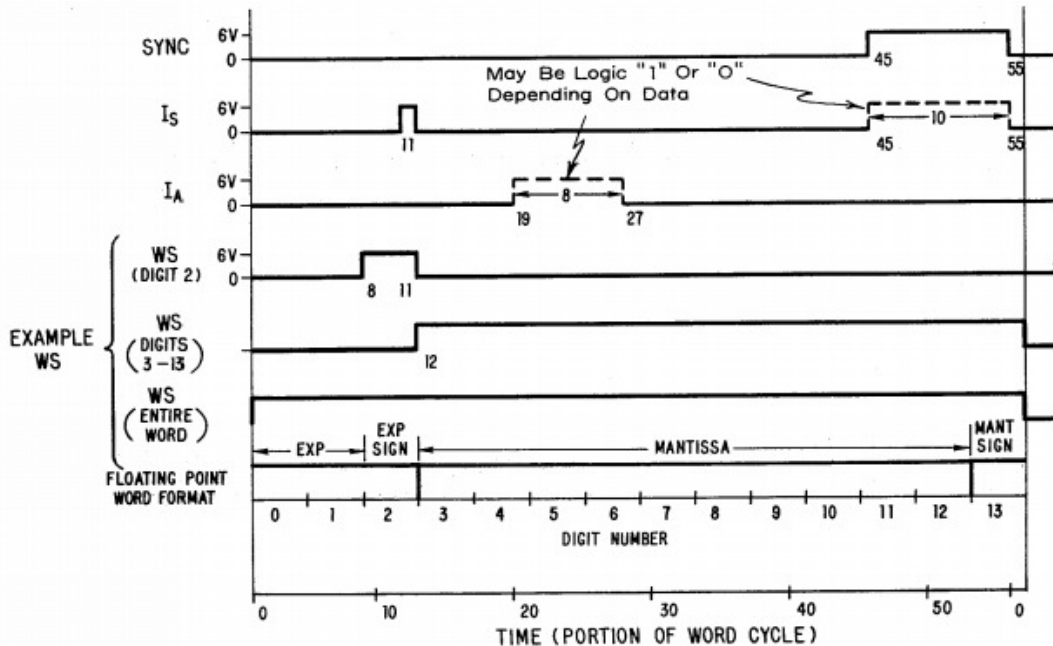


Figure 6 - Timing diagram

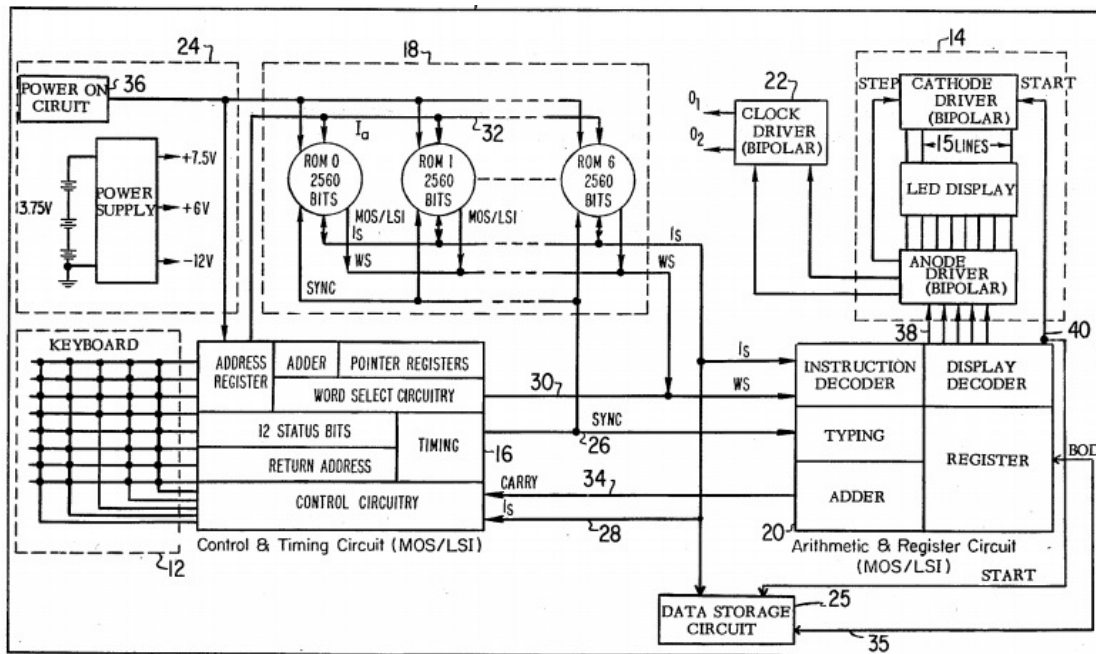
Figure 7 - block diagram

To do a good test, we needed to be able to test all 6,000 transistors and all the logic paths. Every single transistor in each part had to be tested and verified it operated to specification. In addition, the patterns programmed into the ROMS had to be checked and verified against a simulation written by Jim Duley of HP Laboratories⁷. This simulation generated the test pattern Mostek needed to test the final IC's. After a simulation was running correctly, a pattern for each input and out on each IC was recorded. This pattern was stored in the PDP-11 memory. The tester would compare that simulation, which came from paper tape, via DMA from the PDP-11 memory which stimulated the Device-Under-Test (DUT).

⁷ <http://www.hpl.hp.com/hpjournal/pdfs/IssuePDFs/1972-06.pdf>

Just like inside the calculator, the CPU and the DUT needed to be in phase with each other so that the bit patterns would match. The West Coast Sync board would stop the DMA until the Sync and Word Select pins were at a known state on the DUT. The clocks would then be in sync. Now each output can be compared to the PDP-11 registers expected output with a LM311 comparator IC.

You can also see that SYNC signal on this block diagram of the calculator⁸:



The comparator IC was also strobed by the Bluebird clock to compare the two chips at the correct time. That IC failed on one of the test boards, and always produced an 'OK' signal. No one detected the problem, and we shipped a large batch of defective parts to HP. It caused a huge ruckus, and everybody was upset.

All parts from then onward had to be hand tested in a hacked HP-35 calculator that did a simple 'reset' test and a 1+2 test. It worked something like this: First the display showed 2, then a 3, and the chip would reset. It then showed 1, 2 and then a 3. If the final 3 happened, the test passed. I found out many years later that Mostek had put in special mask 'cheats' to prevent people from copying our chip. Our chief designer, Robert Proebstein⁹, had found out a way to put a pad on top of a silicon hill, but without connecting it to the obvious trace below. A copier would not be able to see this hill that served as an insulator unless they used a side-viewing microscope, which was rarely used. Copying was always done with a top-illuminated microscope. The copier would think it was just an ordinary via, and connect the trace to the layer below. This chip had a circuit to signal the very specific 2+3 operation, and this would then be connected by the wrongly copied pad to the reset pin of the chip.

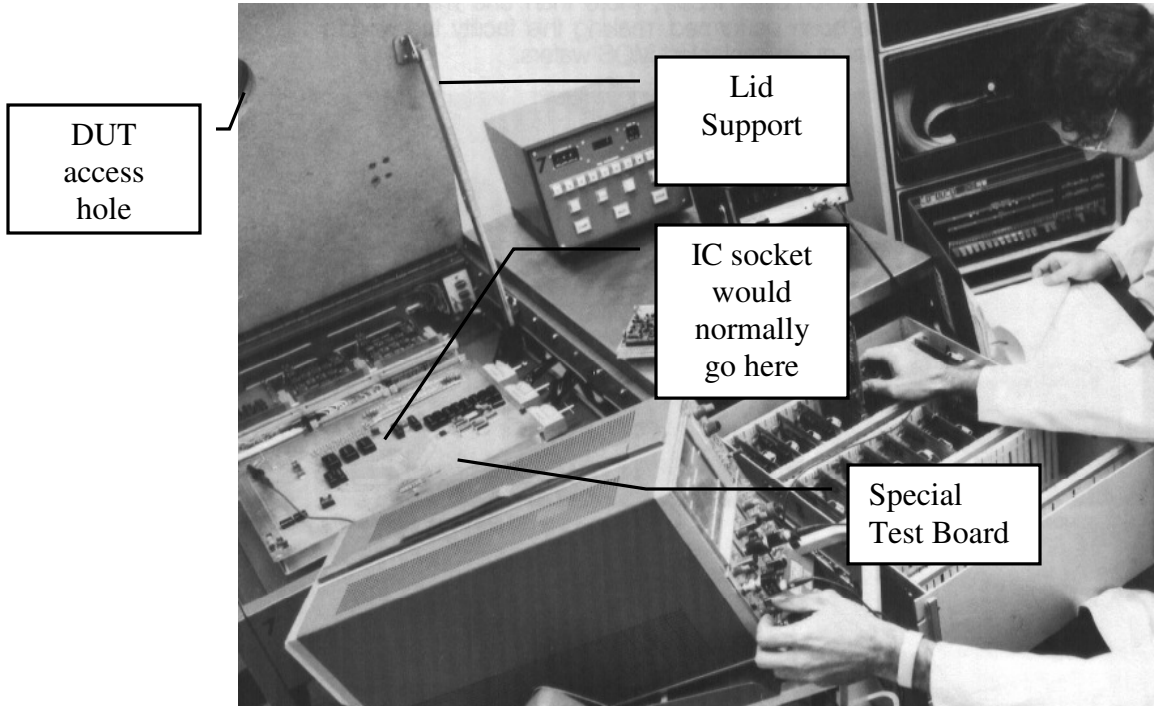
⁸ <http://pmonta.com/calculators/us-patent-4001569.pdf>

⁹ <https://www.cs.utexas.edu/~hunt/class/2016-spring/cs350c/documents/Robert-Proebsting.pdf>

The test set would just press 2, Enter (It was a reverse polish calculator) , a 3, and then +. This would reset the chip. It then pressed 1, Enter, 2, and +, displaying a 3. If the chip was from AMI, or a bad chip, a different pattern would appear.

We also hired more QC people. They tested every lot of parts before and after a test was set up. I actually met my future wife because of that incident - she was one of the new QC personnel they had to hire for the increased workload for this production area. And she was cute. It took a few years, but eventually I got her to agree to marry a geek, and we have been together for 36 years as of this writing in 2016.

All these tests and logic circuits in the Bluebird in turn had to be tested. so a special test program and circuit was created just to test the Bluebird. You can see this in the photo. The lid is raised up to access the test board which held various circuits specific to each IC. There is a hole in the lid, just to the upper left, where the Device Under Test (DUT) would be placed into a test jig on top of the test board.



There were two drawers of logic in the Bluebird. The bottom held three rows of logic to interface to the PDP-11. The top drawer held analog circuits, such as level shifters and D/A converters. The very expensive sets of 12 bit D/A converters (from 1970, when the system was first designed) were plugged into card slots in the top rack.

Bluebirds also ran our wafer-level probe tests on the raw wafers. A Electroglas wafer prober would be set on top of this lid, and a cable ran through the hole to the test point card in the Electroglas. The tests for leakage and speed would be reduced when testing bare die by the PDP-11.



Figure 8 - Electroglas wafer prober

The Bluebird had 24 output bits that could be changed on-the fly by setting memory in the PDP-11 with 1's and 0's/ This was done by writing a 15-bit start address and length into a Bluebird register, writing a test destination address into another register, and then enabling the tester. The logic circuits would automatically request a 16-bit memory address, write the data to a temporary register, read the next 8-bit byte, and write the resulting 24 bits to the DUT when the timing generator indicated it was time to do so. We referred to this as 'cycle stealing' from the PDP-11. These bits would travel through level translators to be at the correct PMOS levels. Other TTL compatible chips could be tested by reprogramming the DACs that drove the level shifter.

We didn't just drive the signals from rail to rail either. We had to drive them between the Min and Max, and at the correct rise and fall times. Every data sheet value and waveforms, such as the signals in the diagram in Figure 5 had to be tested. For a TTL part, these input levels would be typically set at 0.8 volts to +2.4 volts with V_{cc} set to Max of +5.5 V. The parts had to always see a worst-case input. We also measured the input capacitance, which is very difficult to do at the end of a long cable. Leakage current was tested, and we also back-biased the inputs to check that the anti-static protection diodes were working.

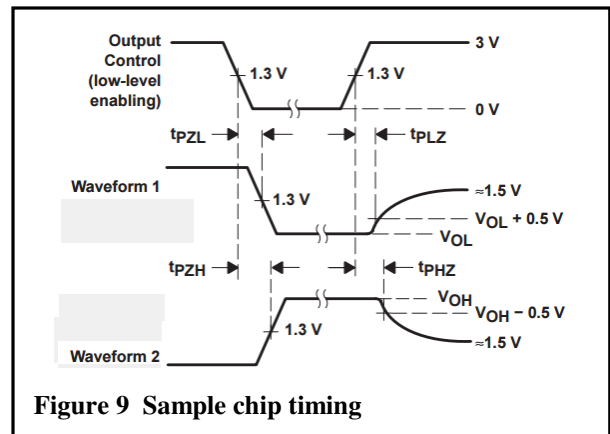
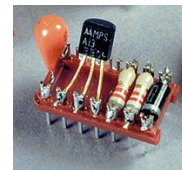


Figure 9 Sample chip timing

All the level shifters and analog circuits had to be custom designed for this purpose. We used standardized wire-wrap cards with small plastic headers to hold the discrete transistors and resistors that made up a level translator or a comparator.



Another requirement is that the HP chips had to meet worst-case output levels. If the part was supposed to drive an output to 0.4 volts at -4 ma, and 2.4 volts high to +3 ma, we had to test for exactly that. This required a bank of comparators (the Bluebird had 16) and suitable voltage sources and test loads, like the circuit at the right, to compare against. These comparators can be seen in first photo just behind the test board. The timing generators would strobe the comparator banks, and write the result data to the PDP-11 memory using a third DMA cycle.

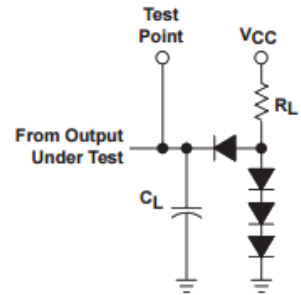


Figure 10 TTL load

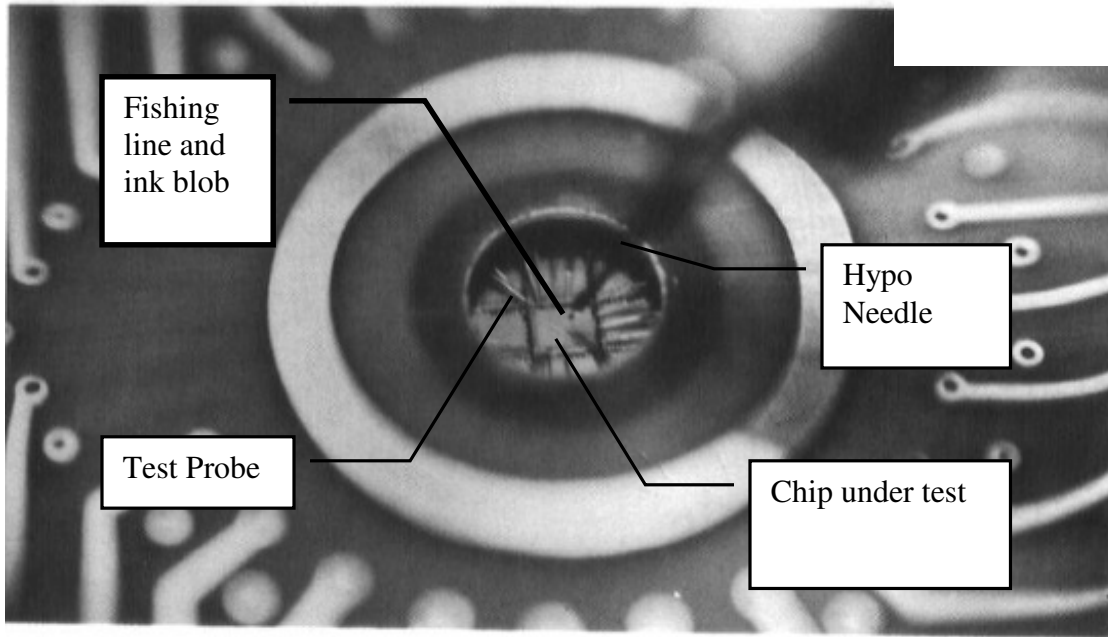


Figure 11 - Test Probe for wafers - Photo credit Mostek 1973 data book

You can see a wafer being tested by a Bluebird in the above photo. I cannot guarantee it, but it is almost certainly an HP-35 chip, just like the one to the right, but it is very likely that the only chips qualified to run at that time were the HP chips.

A 3" wafer would be loaded into an Electoglas and aligned under a microscope. The test person would press start, and the Electoglas would step across the chip and press a set of microscopic probes attached to the PCB (above, has a circular hole in it). The blurry part that looks like a pencil was an ordinary medical syringe, filled with ink. It had a plastic fishing line running through it. The fishing line can be seen in the photo just above the IC. A solenoid was attached to the fishing line and the line was cut off just below the tip of the syringe needle.

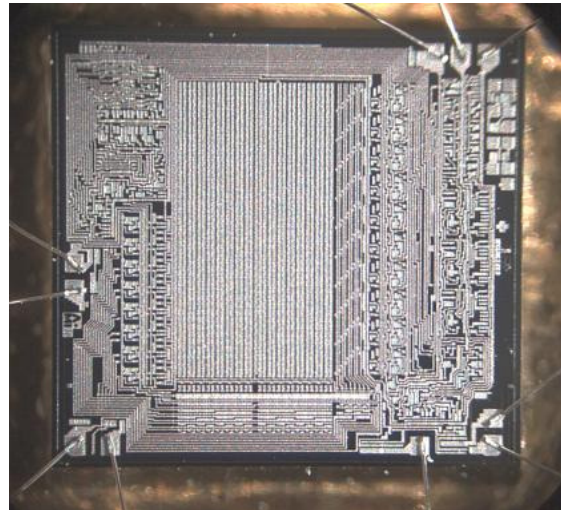


Figure 12 - HP ROM

If the part failed the test, the Bluebird would enable the solenoid, which would push the fishing line, and jot a blob of ink onto the part.

The wafer would be sent to the dicing department which scored and broke the wafer into individual parts and discarded the inked bits. The rest would go to the bonding and packaging department, and get routed back to another Bluebird line for testing in the DIP package. In some cases, the parts would go through a heat or chill cycle in a large oven attached to another row of Bluebirds.

This cheap and effective method of marking bad die sometimes led to interesting patterns on the wafer. I was once asked to check out a Bluebird that seemed to be passing almost every part on a wafer. This was usually a nightmare - it was never broken in a way that was easily fixable. You can't fix what ain't broke. In this case, it was easy to see that only the parts at the edges of the round wafers that were cut by the round edge were failing the test. This was either continuity or leakage test, and not the functional test which usually mean that there as a problem in the logic of the tester that tested the functions of the circuits.

I eventually noticed that one wafer had a line of dots running across it that was a great clue. The tester was correctly failing functionally bad parts. Looking at that wafer under a special microscope showed a microscopic grain boundary in the silicon - and transistors do not work across a grain boundary! They require a single crystal of silicon. The Product Engineer for the part later told me it was a final run for very old part, a MK1002 dual 128 bit shift register. These parts were now being built in a new class 100 clean room using modern equipment and methods, and so the yield was now close to 100%.

Oh, about that horrible blue 'Bluebird' color. We painted everything bright blue. All test equipment had to be that way. Rows and rows and rows of testers. Room after room. A technician named Mike Chambers got permission to design and build a board-level tester for our test lab based on the Z80 chip we were making at the time. And since we were in charge, he ordered a very pretty burnt-orange and beige rack for our new "Puma" - short for "Programmable Universal Maintenance Aid". We figured we could get away with it as it was a test lab, not a production department, and we had a different set of bosses.

Eventually the tester rolled out and went online in our lab. Shortly after we got word that the Vice President who was in charge of color was upset and wanted us to take it apart and paint it all horrid blue. We ignored it. A few weeks later, we got an actual memo. We ignored it. We figured no one would take a very expensive machine like that apart. We were right, too.

We came in on Monday morning, and the room was painted burnt orange and beige.

The Bluebird was a workhorse of a tester. It lasted for many years and tested almost every IC Mostek made except for RAM and the microcomputers. RAMs were tested on the Master and MiniMaster product lines (Gray cabinets) and CPU chips were tested on Fairchild Sentry 600 and 700 series machines (Beige cabinets). Bluebirds received several upgrades over the next 5 years - the wire wrap boards were retired, printed circuit boards were hand-taped, and wire wrap was used only for the two backplanes. The tape readers in the PDP-11's were replaced with serial RS-232 lines running to a VAX-11-780. We designed and laid out a diode ROM board for the PDP-11's so keying in the bootstrap was no longer required.

After five years of this type of work, I got a lateral transfer to the microcomputer department and helped the engineers design STD Bus, Euro card, and VME bus systems. After 8 years I left Mostek and started a Unix microcomputer company based on STD Bus systems and the 68000 microprocessor. I still work with Rex Brown at Micro Technology Services, (www.mitsi.com). Rex was employee #13 at Mostek, one of their best test equipment designers, and the person who hired me into the microcomputer department so many years ago. We worked on STD and VME busses together for several years, and have hired each other back-and-forth at different companies for almost 40 years now.

~ end~

For further reading:

<https://web.archive.org/web/20040203113550/http://www.vcalc.net/hp-35.htm>

<https://en.wikipedia.org/wiki/Unibus>

<http://textfiles.com/bitsavers/pdf/dec/unibus/UnibusSpec1979.pdf>

<http://www.hpmuseum.org/hp-35.htm>

http://www.datamath.org/Mostek_IC.htm

<http://www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/articles.cgi?read=385>

<http://www.hpl.hp.com/hpjournal/pdfs/IssuePDFs/1972-06.pdf>

<https://www.cs.utexas.edu/~hunt/class/2016-spring/cs350c/documents/Robert-Proebsting.pdf>

<http://pmonta.com/calculators/hp-35/>

<http://pmonta.com/calculators/us-patent-4001569.pdf>

<https://people.rit.edu/lffeee/PMOS150.pdf>

My first real job was Mostek from 1973 to 1982 as a test technician. I was just a 19 year-old kid, still going to school. Over the next five years I became a test equipment group leader working with 12 other technicians on my shift. We repaired all of Mostek's IC test equipment. They were all state-of the art for that time, and cost many millions of dollars. As the business grew from a few hundred employees to 6,000 people, we ended up with three very large rooms filled with rows and rows of Bluebirds, Masters, Mini-Masters, Teradyne's, Fairchild Sentry 600 and 700 series machines, and a lot of custom systems.