Traps

The oral history of the Z80 discusses how the designers of the chip put traps in the chip to make it harder for other companies to clone the chip. Faggin came up with the idea of using depletion load transistors that look like enhancement transistors.

As discussed earlier, NMOS gates are built from enhancement transistors that turn on and off to connect the output to ground. This is what causes the gate to work. Each gate uses a depletion transistor that acts as a resistor to pull the output high if nothing is pulling it to ground.

Looking at a chip, it's fairly easy to tell the enhancement transistors from the depletion transistors. First, their position in the gate is different: enhancement transistors form a path to ground, while depletion transistors are connected to +5. Second, depletion transistors have their source and gate connected together. Finally, depletion transistors are shaped differently - they have a lower W/L ratio . (This is the width to length ratio of the gate dimensions, where L is the distance between the source and drain, and W is the width of the gate perpendicular to the source-drain line.) The lower W/L ratio makes the pullup transistor weaker, providing less current, so any ground connection will overpower it. Since the minimum width is fixed by the manufacturing process, the effect is the length of a depletion transistor is longer.

The consequence of these factors is that it is pretty easy to tell when looking at a chip, which transistors are which type. However, there is actually no visible difference between the transistors. A depletion transistor is created from ion injection, injecting more n-type impurities into the region under the gate. (Anitimony or arsenic are example n-type dopants, with 5 valence electrons (one more than silicon). See
Introduction to VLSI Systems, Mead and Conway chapter 2).

Specifically, a separate mask is used to expose parts of the chip to the ion implantation. This step is done fairly early in the fabrication process, before the polysilicon layer is put down and the diffusion is done. The ion implantation region is slightly bigger than the depletion transistor to ensure

that the entire transistor becomes depletion. After the implantation, an annealing step is necessary. The silicon is heated to high temperature for many hours. This repairs any damage caused to the silicon by the ions slamming into its crystalline structure. Ion implantation is used because it provides more control than diffusion, allowing the threshold voltage of the transistors to be set precisely.

Prior to depletion transistors, enhancement mode transistors with the gate tied to Vdd were used as pullups in the gates. The disadvantage of enhancement pullups is the voltage is pulled up much more slowly (Mead and Conway), and remains a threshold voltage below Vdd. This makes the gates slower, with a worse noise margin. They also required a separate power supply for the load gates (Vgg).

According to Wikipedia, Mostek did much of the pioneering work on depletion load MOSFETs. Faggin introduced depletion loads to Intel (when he worked there) in 1974, redesigning the 2102 memory chip to the 2102A which was much faster. Depletion loads became very popular, used in the 6800, 6502, 8085, 8086, 8048, 6809, and other microprocessors of this time. In 1976, Electronics magazine wrote an article “Depletion mode shrinks CPU chips” saying that “Microprocessors, by adopting techniques used in calculators and memories, are also approaching microinstruction times of biolar Schottky devices.” The depletion load enhanced version of the 6800, the 6800D, was said to be twice as fast and half the size as the 6800. (New Scientist 22 July 1976)

For an enhancement transistor, the gate must be positive (relative to the source) to turn the transistor on. The voltage where it turns on is the threshold voltage. In a depletion transistor, the threshold voltage is negative, so the transistor is already on when the gate is at 0 volts. The point where the gate turns off is a negative voltage. In both cases, the transistor is off for sufficiently negative voltages and on for sufficiently positive voltages, so the transistors work in the same “direction”. (This is in contrast to PMOS transistors.) The difference is the voltage at which they switch on.

The point of this explanation is that a depletion mode transistor will always

be turned on (under standard input voltages), unlike an enhancement transistor which will switch on and off. However, the implanted ions are invisible, so both types of transistors physically look the same. The trap is to put a depletion mode transistor in a circuit where you'd expect to see an enhancement mode transistor. If you copy the die based on its physical appearance and use an enhancement mode transistor, the chip will malfunction when the transistor turns off.

This is the explanation of the traps that Faggin came up with to put in the Z80. He gave Shima the task of figuring out where to put these traps in the circuit. Shima added six of these misleading transistors to the circuit.

According to Shima, these trap transistors delayed NEC's copy of the Z80 by 6 months, so they were successful.

In our reverse-engineering of the Z80, we found traps in several ways. First, we noticed instructions that malfunctioned. By looking through the circuit, it was possible to figure out where the operation went wrong. This pointed the finger at suspicious transistors.

We confirmed the presence of traps by looking at clones of the Z80 that had eliminated the traps. If a suspicious transistor was in the original Z80 but not the clone, that was a clear indication that it was a trap transistor.

The third way to find traps is by special processing of the die before taking photographs. The chemical properties of the silicon are slightly altered by the ion implantation, causing enhancement and depletion transistors to be etched slightly differently by certain chemicals. The difference shows up in photographs, making it possible to distinguish the enhancement and depletion transistors.

A modern technique for finding enhancement and depletion transistors is to use a scanning capacitance microscope. This is an atomic force microscope with a circuit to measure the capacitance between the tip and the sample. The differential capacitance is affected by the dopant profile, allowing the dopant gradient to be measured at each point in the chip by scanning across the surface. This gives exact information on the chip doping at high resolution.

Scanning Capacitance Microscopy was invented in 1985, using the sensor from a RCA SelectaVision VideoDisk system, so it wasn't an issue when the Z80 came out. (Reference Scanning Probe Microscopy: Sergei V. Kalinin, Alexei Gruverman). For an example of a commercial lab providing SCM, see http://www.chipworks.com/en/technical-competitive-analysis/resources/blog/scm-at-chipworks/

In his oral history (http://archive.computerhistory.org/resources/text/Oral_History/Faggin_Federico/Faggin_Federico_1_2_3.oral_history.2004.102658025.pdf), Faggin discusses the traps and how they were made progressively more difficult, ending with a very subtle trap that was originally a logic bug in the conditional jump circuit that they had fixed - but they made the fix a trap.

Faggin discusses talking with the lead engineer on National Semiconductor's CMOS copy of the Z80, who had been caught by the traps years earlier. The engineer said that it took him nine months to track down the traps. Faggan replied, “Wow. We thought that a good guy could take six months to do it.”

This dopant-trap idea has gotten some attention lately in a 2013 paper, “Stealthy Dopant-Level Hardware Trojans” by Becker et al. In this paper they discuss how a malicious manufacturer could build Trojans into a chip by manipulating the dopant levels. The paper describes how a circuit such as Intel's cryptographically strong random number generator could be sabotaged by manipulating transistor dopant levels. (The paper doesn't mention the Z80, so they may not have been aware of the Z80's use of these techniques decades earlier.)

To see one of the traps in action, when we (i.e. Pavel) first simulated the Z80, some instructions took too many T cycles to complete. The cause turned out to be the LAST_T logic (described in a separate chapter). One of the gates takes M5 AND T5 AND some pla decoding to feed into the LAST_T generation. Most of the timing / decoding logic follows this pattern, combining an M signal, a T signal, and a PLA signal. In this case, it will cause those instructions to end with M5T5.

However, I noticed that some instructions went to T6 that shouldn't. For example, RET NZ took 6 cycles, but should have taken 5. PUSH BC took 6/3/3 cycles (in 3 M cycles), when it was supposed to take 5/3/3. RST and RET Z had the same behavior.

The cause is the transistor gated by M5 in the AND gate was a depletion trap. This transistor was always on, so the real gate should have been T5 AND pla, without M5. The consequence is that instructions that should end in T5 in other M cycles will continue to T6.

Another trap is in the register bus circuit. Faggin mentions this trap in the oral history:
“The first trap would inhibit all the communication from the internal bus so, basically the chip would be deaf and mute and that would be it.”

He describes how someone copying the chip would need to do another silicon spin at this point, taking a month, at which point they would hit the next trap. Since we were just using a simulator, the traps weren't quite as devastating to us.

This trap, as it was found by Pavel, turned up with the CALL instruction's return address getting pushed to the wrong address because the SP register gets corrupted on the bus.

I found that in M3T3, while the SP was getting transferred to the address latch, the data bus was also getting transferred to the register bus, corrupting the SP.

The problem was an input to a NAND gate from a pla input from the input instruction, causing the NAND gate to almost always be 1. (10/13/2013). This would mess up any instruction reading a register in M3T3.

One interesting feature of this trap is it couldn't be simply removed, since the signal passing through the transistor was used on the other side. In other words, the trap was made part of the layout and routing rather than just being

dropped into the layout at the end.

Note that a trap can be added to an existing NAND gate fairly easily, since a stuck-on transistor has no effect on the gate. A trap in a NOR gate would force the output to be 0, making the gate non-functional. Thus, a NOR trap would have to be implemented with an additional (nonfunctional) gate, rather than just an additional transistor.

Another trap was in the logic to start a new M cycle after M3T1 had an extra input connected to a PLA.

The exchange instruction flip flip circuit has some suspicious transistors that we thought were traps. After investigation, they appear to be workarounds for some timing issues, rather than traps.

Our investigation into traps also found errors that we introduced in digitizing the chip images, as well as problem with the simulator.

Another trap was at (4216,914). This trap was in the register control where a NAND gate combines T1, the write control, and the data-pin-latch-to-databus control line. The last input is the trap. The purpose of this circuit is when writing to the data pins to connect the register-side data bus to the pin-side data bus at the end of T1 / beginning of T2. The trap signal messes up the timing of this control, so the data bus isn't activated at the right time for the pass transistor. The result is the data pins end up with random data instead of the desired data.

At this point, we have found most of the traps, but not all of them. As Faggin described, the traps are placed maliciously, causing trouble in a variety of ways. In addition, the traps are designed so the control lines appear reasonable; they don't use random inputs that wouldn't make sense, but inputs that are plausible, such as using M5 in the LAST_T computation.