Lesson 2: reading schematics: Version 1.07

View the attached schematic of the Atari Centipede (centipede-schematic-1.jpg). Find the large rectangle with the green **P** noticed on the schematics it is labeled “C2 6502A” this indicates that at the Centipede PCB board at position C2 there is a 6502A chip, which is the CPU.

Notice on the schematic on the right side of the chip the are labels on the inside and outside. The labels on the inside indicate what the function of the chip is (as generally referenced on the datasheet) the outside number is the actual pin number on the chip. Therefore A14 is pin 24 on the 6502 chip.

Notice there are 16 address line (A0-A15). When the CPU board is powered on each address line will carry either a GROUND signal which represets binary 0, or +5V which represents binary 1. Since there are 16 different address lines and each line can have two different states, there are 2^{16}= 65536 different combinations of 0s and 1s therefore there are 65536 memory locations the CPU can address.

Addresses are generally referenced in hexidecimal format such as 0x041A however the computer represents everything in terms of 0 or 1 (binary) so that it can turn it into electronic voltage levels. Luckily hex can be easily converted to binary, each single hex digit can be translated directly to 4 binary bits using the table below.

0x0 = 0000	0x4 = 0100	0x8 = 1000	0xC = 1100
0x1 = 0001	0x5 = 0101	0x9 = 1001	0xD = 1101
0x2 = 0010	0x6 = 0110	0xA = 1010	0xE = 1110
0x3 = 0011	0x7 = 0111	0xB = 1011	0xF = 1111

So for example take the example address 0x041A this can be converted easy by replacing each hex digit with its corresponding binary 4 digits.

0	4	1	A
0 0 0 0	0 1 0 0	0 0 0 1	1 0 1 0

So the binary representation of 0x041A is = 0000 0100 0001 1010

When the CPU wishes to address this memory location it will apply the proper voltages to the proper address lines. (Note Addresses bits are numbered from right to left)

Address line	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Binary	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0
Voltage	0	0	0	0	0	+5V	0	0	0	0	0	+5V	+5V	0	+5V	0

Now imagine the CPU is accessing this memory address. Go ahead and write in the appropriate values on each address line on the schematic.

Copyright 2013 Paladin Group, LLC Brian Brzezicki

Following the address lines to the right you notice they all become the input to chips labled in green **A** and **B**. These are chips B1 and C1 and are labeled LS244. This is short for 74LS244 and defines what type of chip it is. The next step is to pull up the datasheet for that chip (search google for 74LS244). It is important you read the datasheet and just get a general idea of what the chip is supposed to do. You do not need to read all the techical data of the operating conditions of the chip, generally the first page or two of description is enough. Most chips will also have a "state table" describing all the inputs and what the associated outputs will be and this is very important. In this case the 74LS244, this chip is a buffer. It's purpose in this circuit is simply to read the signal coming in and produce the same signal on the output pin. Since this is so simple some datasheets will not even show the state table. The reason a buffer is here is because while the CPU outputs it's desired "state for each address line", the CPU does not "source or provide" enough current that is required for valid inputs to most other 74LSxxx series chips. The 74LS244 in this case is somewhat like an amplifier or a repeater, and service to simply provide the necessary current on it's outputs.

Now let's take the inputs to chip C1 based on your knowledge of the inputs, what will the outputs for the associated pins be? Fill in the following table

Address bus number	Pin number	Output value
AB7	3	
AB6	5	
AB5	7	
AB4	9	
AB3	18	
AB2	16	
AB1	14	
AB0	12	

Notice also pins 1 and 19, there pins enable whether the chip is outputting a value or not. They are hardwired to ground, which makes this chips permanently ON or *enabled* and providing an output on each pin reflective of the input. Often you will see 74LS244s used in another fashion. Sometimes there will be multiple potential input chips on a shared bus (for example RAM or ROM chips). Since only 1 device can talk on the bus at one time without causing issues, 74LS244 chips are often put at the end of a storage device and it will isolate that storage from the bus. When that storage device should be talking on the bus, the buffer chip will be *enabled* and outputting values, all other times the buffer chip will have an *enable* input that tells the chip to go OFF (called *high impedance*) mode which effectively removes the chip from the bus, and allows another device to talk on the bus. You will see this in multiple places in RAM and ROM, and INPUT circuitry.

Copyright 2013 Paladin Group, LLC Brian Brzezicki

Exercise: **Exploring the address bus using a Fluke 9010A and a logic probe:**

In this experiment you will attach the Fluke 9010A to your Centipede PCB and manipulate the address bus, you will note the changes going in to the 74LS244 at position C1 as you send signals down the address bus. This assumes your Centipede PCB is fully working. It is best to setup the Centipede PCB on a jamma harness/test bench using the Centipede PCB to JAMMA adaptor available from arcadeshop.com

1. Turn off your Centipede PCB
2. Turn off your Fluke 9010A
3. Remove the 6502 CPU from the Centipede board
4. Attached your Fluke 9010A with 6502 POD into the slot left abandoned when you removed the 6502 from the centipede board
5. Attach your logic probe to the test points on the Centipede PCB board (+5V to red wires on your logic probe, GROUND to black wires on your logic probe)
6. Turn on the Fluke 9010A
7. Turn on power to the Centipede PCB board
8. Go into the Fluke Self setup, use the “more” button to move to the selection for “Active Line Force”, hit “NO” to turn this off.
9. Hit the “Bus Test” button, the Fluke should respond with “BUS TEST OK”
10. Hit the “read” button, when it asks for an address type FF
11. Hit the “Loop” button
12. Touch the tip of the logic probe to pin 17 on the 74LS244 at position C1 (address bus line 7,input). You should see the signal is high*.
13. Touch the top of the logic probe to pin 3 on the 74LS244 at position C1 (address bus line 3). You should see the signal is high*. Therefore address line 7 is “on” (which makes sense because address 0xFFFF is binary 1111 1111 1111 1111, so all lines should be logic high (+5V)
14. Hit the “read” button on the Fluke, when it asks for an address type 0000. Now the Fluke should be writing 0’s for all bits on the address line (GROUND)
15. Hit the “Loop” button.
16. Touch the tip of the logic probe to pin 17 on the 74LS244 at position C1 (address bus line 7,input). You should see the signal is traverses from high to low*. (it should NOT stay low though)
17. Touch the top of the logic probe to pin 3 on the 74LS244 at position C1 (address bus line 3). You should see the signal is traversing from high to low*. (it should NOT stay low though).

    The reason that chip inputs and outputs do not stay low is because the CPU is only taking them “low” for a fraction of a second.

* Important Note: It is very easy to misread a signal on the logic probe, if the pin is not clean you may get incorrect results (especially when pins transistion for just a fraction of a second). For example when reading the address lines that are set to 0, you might see that address pins stay "high" if you see this clean off the pin you are testing, this phenomenon has caught me many times!

Copyright 2013 Paladin Group, LLC Brian Brzezicki

18. Experiment with writing to different addresses with the Fluke (don't forget to hit "Loop"). Do the pins match what you'd expect the binary value of the address would be?

When your done don't forget to detach the fluke properly.

1. Turn off power to the Centipede PCB
2. Turn off power to the Fluke 9010A
3. Remove the POD from the CPU socket, and return it to the ZIF socket on the POD.

Special thanks to KLOV users TROXEL and BARITONOMARCHETTO for reading through this guide, doing the exercises, catching many many typos, and generally making this a better document.

Copyright 2013 Paladin Group, LLC Brian Brzezicki