

In the “lessons 1- 3” you learned the basics of the CPU signals and how to read schematics. In this lesson you will learn to use the Fluke 9010A to verify ROMs in circuit. Verifying ROMs is one of the essential steps in order to ensure correct system operation as the ROMs are the actual game code that the CPU runs.

If you are new to the arcade collecting hobby and you have a game that’s not working, and ask for advice, the most common advice you’ll get is “remove and reseat the ROMS”. If you don’t have a Fluke 9010A that’s not a bad idea, however if you have a Fluke, I’d highly encourage you NOT to do that. It’s not that is a bad idea in theory, but in practice it ends up causing more problems than it solves. There are two issues with this

1. Doing anything in a “shotgun” like fashion is dangerous with CPU boards. Unlike monitor boards or power supply boards where “shotgun replacing capacitors” is a good idea as the caps degrade and are usually the cause of most of the issues, CPU boards are generally purely digital rather than analog, and generally do not have components that degrade, they either work or they do not work. What’s more is unlike monitor and power boards that have thick heavy traces and easy to work on components like capacitors where its hard to replace things without bringing them; CPU boards are very fragile with small fragile componets and EASILY damaged.
2. While I’ve seen one or two boards where an issue was a dirty ROM or ROM socket, it’s literally been one or two out of probably now hundreds of boards. I have however seen a TON of boards where someone tried to reseat ROMS and ended up breaking a ROM pin, bending a ROM pin, inserting a ROM with the pins on the inside or outside of the socket, inserting a ROM in backwards, or inserting a ROM into the wrong socket, all cause the game not to work.

If you have a Fluke it is a much better idea to PROVE that there is a ROM error and narrow it down to the actual ROM that is the cause and then take whatever action is necessary, whether that be replacing the ROM, replacing the socket or what ever needs to be done.

How to test ROMS with help of the MAME project, “KI4SWY’s (Hamster’s) Web based ROMident” page, and the Fluke 9010A:

If you have a Fluke you can use it to internally read each ROM from the CPU’s standpoint, and perform a checksum on each ROM to make sure it is valid and reading correctly. This is very important. You could remove each ROM and perform a checksum using a ROM burner/reader on your PC, however in my experience ROMs rarely go bad. What’s more a ROM might read fine externally but not read correctly in circuit, this is not due to a bad ROM chip, but due to failure of supporting chips. There are many more components needed for the ROM to read correctly in the circuit and any of them are more likely to fail than the ROM itself. You have already seen one of these circuits, the *address decoding circuitry*. But there are many others, including the chips in the *data bus* circuitry. A good ROM will NOT read correctly if any of the supporting chips go bad, even if the ROM itself is OK.

To check the ROMs in circuit, you will need your Fluke 9010A and CPU POD. But you'll also need to grab some other information from the Internet.

1. The first thing you will need are known good copies of the ROM data from the game in question. There is no universal way of getting this information, but if you google "*game_name* **MAME ROM**" (example "**centipede MAME ROM**") you probably will find a zip file containing the game's ROM. Keep in mind any given game might have multiple ROM versions as bugs were found and vendors released new game code.
2. Once you have the ROMs unzip the ROM folder somewhere on your computer so you can access the individual ROM files.
3. Now head over to KI4SWY's RomIdent page (<http://romident.coinopflorida.com/>), For each ROM in the file, upload the ROM image and copy the, "ROM name" "SHA1 checksum" and the "FLUKE SIG"

game	maws link	desc	rom name
centiped	centiped	Centipede (revision 3)	136001-307.d1
centipdd	centipdd	Centipede Dux (hack)	136001-307.d1

Other information useful about this file:

CRC32B	5ab0d9de
CRC32	76c4d1ee
SHA1	8ea6e3304202831aabaf31dbd0f970a7b3bfe421
FLUKE SIG	BFA2
CHECKSUM	7675

4. Use a editor (notepad, wordpad, MS Word, or vi/vim if you are really cool) to copy down the ROM name, the FLUKE SIG, and the CHECKSUM of each ROM file.

- Download and install MAME (<http://mamedev.org/>).

What is MAME:

MAME is the **M**ultiple **A**rcade **M**achine **E**mulator. It is a program that emulates the actual hardware of the arcade systems, and lets you play games using the REAL game code. MAME is an excellent project and does an excellent job making old games playable to anyone. However, the project's goal is not to play games but to document and simulate the actual hardware, which is often not easily found anymore. The documentation MAME provides and the ability to see how the hardware works is a great tool to fixing the actual hardware.

- Make a spreadsheet that looks like the following, ensure you have enough lines for each ROM file.

ROM Name	SHA1	FLUKE SIG	Size	Start	End	Main cpu

- Fill in the sheet with the ROM name, and the SHA1 and FLUKE SIG from the data generated from to KI4SWY's RomIdent page.

ROM Name	SHA1	FLUKE	Size	Start	End	Main cpu
136001-307.d1	8ea6e3304202831aabaf31dbd0f970a7b3bfe421	BFA2				
136001-308.e1	af4fdbf32c23b1864819d620a874e7f205da3cdb	14EB				
136001-309.fh1	689fa560d40a384dcbcad7c8095bc12e91875580	600D				
136001-310.j1	c557db83876afc8ab52047ab1a3c3bfef34d6351	A93C				
136001-211.f7	6c862352c329776f2f9974a0df9dbe41f9dbc361	A7C3				
136001-310.j1	c557db83876afc8ab52047ab1a3c3bfef34d6351	A93C				

- Use **mame** to printout the game information for example typing

mameXX.exe gamename -listxml

Example:

```
C:\Users\student\Downloads>mame64.exe centiped -listxml
```

(if you don't know the actual game name that MAME uses, just type the first few letters of the game name and MAME will tell you the games it knows about that might be the game your looking for)

```
C:\Users\student\Downloads>mame64.exe cen -listxml
No matching games found for 'cen'

"cen" approximately matches the following
supported games (best match first):

centaur          Centaur
centipdb         Centipede (bootleg)
centipdd         Centipede Dux (hack)
centiped         Centipede (revision 3)
centiped2        Centipede (revision 2)
centtime         Centipede (1 player, timed)
cmmh162          Centipede / Millipede / Missile Command
v 1.62)
m4centpt         Centrepoint (Barcrest) (v1.3) (MPU4)
m4centpta        Centrepoint (Barcrest) (v1.5) (MPU4)
m5centc1         Centurion Club (Empire) (MPU5) (set 1)
m5centc120       Centurion Club (Empire) (MPU5) (set 2)
m5centc121       Centurion Club (Empire) (MPU5) (set 3)
m5centc121a      Centurion Club (Empire) (MPU5) (set 4)
m5centcla        Centurion Club (Empire) (MPU5) (set 5)
10yard85         10-Yard Fight '85 (US, Taito license)
area51a          Area 51 (Atari Games license)
```

- Once you've successfully run "**mame64.exe game_name -listxml**" you will get A LOT of output. Scroll back till you find the lines that start with the letters "<rom name="

```
<rom name="136001-307.d1" size="2048" crc="5ab0d9de" sha1="8ea6e
3304202831aabaf31dbd0f970a7b3bfe421" region="maincpu" offset="2000"/>
<rom name="136001-308.e1" size="2048" crc="4c07fd3e" sha1="af4fd
bf32c23b1864819d620a874e7f205da3cdb" region="maincpu" offset="2800"/>
<rom name="136001-309.fh1" size="2048" crc="ff69b424" sha1="689f
a560d40a384dcbcad7c8095bc12e91875580" region="maincpu" offset="3000"/>
<rom name="136001-310.j1" size="2048" crc="44e40fa4" sha1="c557d
b83876afc8ab52047ab1a3c3bfef34d6351" region="maincpu" offset="3800"/>
<rom name="136001-211.f7" size="2048" crc="880acfb9" sha1="6c862
352c329776f2f9974a0df9dbe41f9dbc361" region="gfx1" offset="0"/>
<rom name="136001-212.hj7" size="2048" crc="b1397029" sha1="974c
03d29aeca672fffa4dfc00a06be6a851aach" region="gfx1" offset="800"/>
<rom name="136001-213.p4" size="256" crc="6fa3093a" sha1="2b7aec
a74c1ae4156bf1878453a047330f96f0a8" region="proms" offset="0"/>
```

- You should see a few things here

- Name of the ROM like you found in your rom .zip file, the size of the ROM, the sha1 checksum, and the offset, and the region

11. Get out the spreadsheet you made a few steps earlier, For each ROM in the list, find the matching ROM line from the **-listxml** output and verify the SHA1 checksum matches (you usually only have to look at the first few numbers and last few numbers to make sure they match). If the SHA1 checksums don't match you need to make sure the ROMS you are looking at are the right version.

ROM Name	SHA1	FLUKE	Size	Start	End	Main cpu
136001-307.d1	8ea6e3304202831aabaf31dbd0f970a7b3bfe421	BFA2				
136001-308.e1	af4fdbf32c23b1864819d620a874e7f205da3cdb	14EB				
136001-309.fh1	689fa560d40a384dcbcad7c8095bc12e91875580	600D				
136001-310.j1	c557db83876afc8ab52047ab1a3c3bfef34d6351	A93C				
136001-211.f7	6c862352c329776f2f9974a0df9dbe41f9dbc361	A7C3				
136001-310.j1	c557db83876afc8ab52047ab1a3c3bfef34d6351	A93C				

12. Now for each ROM in the spreadsheet, find the "region" section from the **-listxml** output and if it says "main cpu" then put an X in the maincpu column

ROM Name	SHA1	FLUKE	Size	Start	End	Main cpu
136001-307.d1	8ea6e3304202831aabaf31dbd0f970a7b3bfe421	BFA2				X
136001-308.e1	af4fdbf32c23b1864819d620a874e7f205da3cdb	14EB				X
136001-309.fh1	689fa560d40a384dcbcad7c8095bc12e91875580	600D				X
136001-310.j1	c557db83876afc8ab52047ab1a3c3bfef34d6351	A93C				X
136001-211.f7	6c862352c329776f2f9974a0df9dbe41f9dbc361	A7C3				
136001-310.j1	c557db83876afc8ab52047ab1a3c3bfef34d6351	A93C				

13. Now for each ROM in the spreadsheet, find the "offset" line from the **-listxml** output and write that in the "start" section. You can skip ROMs that do not include the Maincpu X, in fact we'll remove them from the chart.

ROM Name	SHA1	FLUKE	Size	Start	End	Main cpu
136001-307.d1	8ea6e3304202831aabaf31dbd0f970a7b3bfe421	BFA2		0x2000		X
136001-308.e1	af4fdbf32c23b1864819d620a874e7f205da3cdb	14EB		0x2800		X
136001-309.fh1	689fa560d40a384dcbcad7c8095bc12e91875580	600D		0x3000		X
136001-310.j1	c557db83876afc8ab52047ab1a3c3bfef34d6351	A93C		0x3800		X

14. Now for each ROM find the size field of the **-listxml** output, this is in decimal (the number system most humans normally use) you will need to turn that into hexadecimal (hex). You can use your calculator on your PC to turn that into hex. Some common values are

Decimal	Hex
2048	0x0800
4096	0x1000
8192	0x2000
16384	0x4000

Write these hex sizes into the “size” column.

ROM Name	SHA1	FLUKE	Size	Start	End	Maincpu
136001-307.d1	8ea6e3304202831aabaf31dbd0f970a7b3bfe421	BFA2	0x0800	0x2000		X
136001-308.e1	af4fdbf32c23b1864819d620a874e7f205da3cdb	14EB	0x0800	0x2800		X
136001-309.fh1	689fa560d40a384dcbcad7c8095bc12e91875580	600D	0x0800	0x3000		X
136001-310.j1	c557db83876afc8ab52047ab1a3c3bfef34d6351	A93C	0x0800	0x3800		X

15. Now for each line above add the “size column” + “start column” and subtract 1. Place the result in the “end” column. Remember these are HEX numbers so they don’t add the same as normal decimal numbers. If you don’t know how to add hex, The calculator that is built into MS Windows and Linux (gnome-calculator) have a programming mode that can do this for you. Just make sure to switch to programming mode then set the type to hex

ROM Name	SHA1	FLUKE	Size	Start	End	Maincpu
136001-307.d1	8ea6e3304202831aabaf31dbd0f970a7b3bfe421	BFA2	0x0800	0x2000	0x27FF	X
136001-308.e1	af4fdbf32c23b1864819d620a874e7f205da3cdb	14EB	0x0800	0x2800	0x2FFF	X
136001-309.fh1	689fa560d40a384dcbcad7c8095bc12e91875580	600D	0x0800	0x3000	0x37FF	X
136001-310.j1	c557db83876afc8ab52047ab1a3c3bfef34d6351	A93C	0x0800	0x3800	0x3FFF	X

16. Setup the Fluke and the logic probe properly (you should know how to do this correctly by now, but remember power the Fluke on FIRST before the Centipede PCB)
17. In the Fluke hit the “Setup” button, choose “more” until “Active Line Force” is selected, choose “No”
18. Hit “Bus Test” to make sure all is working OK
19. For each line in your ROM chart test the ROM Range with the Fluke
- Hit “ROM”
 - When it asks for a start range, enter the value in the “Start” column
 - When it asks for an end range enter the value in the “End” column
 - When it asks for a SIG, enter the value in the “FLUKE” column
20. For each RANGE you test the Fluke should return an “OK”.

Troubleshooting ROMs:

- Test each ROM range, even if the first ROM test shows bad.
- If all the ROMs are bad, take one of the ROMs out and use a ROM reader to copy the contents to a file, use KI4SWY's RomIdent page to verify that the ROM is good AND the CORRECT version. It is possible your game has a different version of the ROM code than what you originally tested against. If so try to find the ROM file for the game revision that you have.
- If all the ROMs are bad, but they check out in the ROM programmer, you probably have a problem with the *address decoding circuitry* or the *data bus circuitry*. Use what you learned from "Lesson 3" to make sure the ROMs are being properly enabled.
- If one of the ROMs shows bad, first use what you learned from "Lesson 3" to make sure the ROM is getting enabled properly, and that NO other ROMs are being enabled at the same time. If the ROM enabling is working correctly, it's probably a bad ROM chip, replace it. If you cannot burn a ROM yourself consider HobbyRoms.com (<http://www.hobbyroms.com>) who can burn you ROMs for a very reasonable fee.

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.