## Subject: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by lynchaj on Wed, 03 Apr 2024 15:38:49 GMT

View Forum Message <> Reply to Message

Hi

Has anyone reverse engineered the SP0256 external serial speech ROMs? Specifically, the SPR016, SPR032, and/or SPR128. They are 2KB, 4KB, and 16KB in capacity respectively. To the best of my knowledge, they are only available as GI/Microchip factor mask ROMs. However, the do seem very similar to the ER1400 which is an old serial electrically alterable read only memory. Although not the same because the ER1400 is read & write while the SPR016 is read-only

https://www.tautec-electronics.de/Datenblaetter/Schaltkreise /ER1400.pdf

The post below mentions there is a code example for the PIC1650 MCU, itself the first GI/Microchip PIC microcontroller to communicate with the ER1400 EAROM. I wonder if this code example could be adapted to modern 8-bit PIC MCUs to interface to the SP0256 emulating the SPR016. However, the functionality would be reversed but still similar.

https://www.ccsinfo.com/forum/viewtopic.php?t=59929

https://developerhelp.microchip.com/xwiki/bin/view/products/ mcu-mpu/8bit-pic/

https://en.wikipedia.org/wiki/PIC_microcontrollers

https://www.eevblog.com/forum/microcontrollers/how-does-this-old-er1400-earom-work/msg4584700/#msg4584700

pages 212 to 220 entitled "Interfacing a PIC Microcomputer with the ER1400 EAROM" for PIC1650 to ER1400 code example in PIC assembler:

http://www.bitsavers.org/components/gi/PIC/1983_PIC_Series_M icrocomputer_Data_Manual.pdf

## Subject: Can the SP-0256A Speech Processor be Reverse Engineered?
Posted by jayindallas on Sat, 06 Apr 2024 16:26:51 GMT

View Forum Message <> Reply to Message

Re: Reverse Engineering the SP0256... "SP ZERO TWO FIVE SIX"

I just updated a message in the CTS-256 topic, to point out that the SP0nnn and SPRnnn are special chips developed by General Instruments, based upon their foundation of chips in music, sound-effects, video game audio, to develop "Speech Processors."

Subject: Reading G.I. SPRnnn External Serial ROM Content
Posted by jayindallas on Mon, 08 Apr 2024 17:01:42 GMT
View Forum Message <> Reply to Message

GI's target market for the SP0256 family of voice processors would generally require larger datastructures than its internal ROM, so the SP0256 was designed with a set of external, serial speech processor ROMs from the start.

The SP0256-AL2 that Radio Shack/Archer later sold as part of a simple Text-to-Voice chipset, is a standard SP0256 with a GI Allophone-based voice processor datastructure already loaded in its internal masked ROM, so it didn't require any voice processor datastructure development nor external serial ROMs, under GI's customer expected use.

SP0256 family chips function the same, only the ROM datastructure developed for each customer application, differs.

General Instrument Serial ROM for the SP0256

```
    SPR016      Signal Function
    SPR032      ======= ==================================================
    SPR128      ROM_ENA# Chip select to eliminate bus conflict at start-up
       ___________      SER_IN Serial input used to load 16-bit address
      _|   \__/   |_     SER_OUT Serial output to shift out data byte
  VSS [_| 01    16 |_] C2     CS1  Chip select; will tri-state serial out when low
      _|          |_     CS2# Chip select; will tri-state serial out when hi
   C3 [_| 02    15 |_] C1     ROM_CLK 1.56Mhz clock input from SPO256 Speech Processor
      _|          |_     VSS  Ground pin
   NC [_| 03    14 |_] SER_IN   C1  Control state pin Nxx, most significant bit
      _|          |_     C2  Control state pin xNx.
ROM_CLK [_| 04    13 |_] NC     C3  Control state pin xxN, least significant bit
      _|          |_     VDD  Positive suppy pin (+4.6V to +7.0V)
   NC [_| 05    12 |_] NC
      _|          |_
   NC [_| 06    11 |_] VDD   PART #  Rom Size and Description
      _|          |_   ======  ==================================
  CS1 [_| 07    10 |_] SER_OUT    SPR016   2K bytes, serial read only memory
      _|          |_   SPR032   4K bytes, serial read only memory
 CS2# [_| 08    09 |_] ROM_ENA#  SPR128  16K bytes, serial read only memory
      |___________|
```

A Method To Read The Contents of a SPR016, SPR032, or SPR128 External ROM
This method wraps the external ROM with a virtual interface to a SP0256, which is orchestrated by an Arduino UNO.

The ROM_CLK signal will not be free-running, because it needs to be Arduino controlled when shifting bits across the interface. When not shifting bits, the Arduino will insert a series of clock transitions, blindly, to give the SPRnnn a chance to use them, if needed to advance its state-machine to the next idle state awaiting a new control-state input. This applies to SPRnnn initialization too.

Message Portion removed for offline editing

---

## Subject: Re: Method to read a SPR-xxx External Rom | 2007 Internal Rom Dump | 2017 SPPR256 by FPGA
## Posted by lynchaj on Mon, 08 Apr 2024 17:34:37 GMT

View Forum Message <> Reply to Message

Hi
There are several versions of the SPR016 for various dedicated SP0256 applications. The SP0256A-AL2 is the one people are most familiar with and it does not use the SPR016. However, I think it could if the CTS256 firmware was updated to tell the SP0256 to use it. Here are several on UTSource

https://www.utsource.net/sch/spr016.html

I see you're using an Arduino to emulate a SP0256 to dump contents of a SPR016. I think this is about halfway to emulating a SPR016 but adjustments to be the mirror image of the commands. Instead of sending them, the SPR016 emulation has to receive them and process to pass information requested.

The main difference is in emulating the SPR016, the real SP0256 controls the ROM clock signal and so the emulation has to respond in real-time. That's where things get challenging due to the sub-microsecond response times. An Arduino might be fast enough to do this depending on which model. But that 641 ns turnaround time will be a challenge with regular microcontroller even with assembler.

---

## Subject: Method to read a SPRnnn External Rom with an Arduino
## Posted by jayindallas on Mon, 08 Apr 2024 18:03:01 GMT

View Forum Message <> Reply to Message

An Arduino would likely work because this is just used to move the bits out of the ROM and out to a serial output pin where an Arduino could easily read that logic level. Since no SP0256 chip is connected there is no sound being synthesized, so there are no performance time-constraints to maintain.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
I'm using the '82 General Instrument Data Book available on BitSavers.org (in the databooks subdir).
The SPR016 is on PDF page-offset 77. The SPR032 and SPR128 follow back-to-back with very little content.
The speech chips start on PDF page-offset 286.

While scrolling through the databook, I saw a page that appeared to be a Speech Processor development block diagram hooking up to 4 byte-wide EPROMs.

The SPR016 relies on a free-running ROM_CLOCK signal to trigger latching Control State settings into the microsequencer but for an Auduino capture of a serial ROM, that's unnecessary. Each STEP in the will now assert the C1C2C3 Control state and then cycle the ROM_CLK to LOAD any mode in.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

A useful SP-0256 URL:

https://en.wikipedia.org/wiki/General_Instrument_SP0256
Or CLICK on "A useful SP-0256 URL:" above.

2007: SP0256-AL2 ROM Image & Letter
http://spatula-city.org/~im14u2c/sp0256-al2/
Or CLICK on "2007: SP0256-AL2 ROM Image & Legal Letter" above.

2017 FPGA Implementation
An FPGA implementation of a classic eighties speech synthesizer SPO256, done for the Retro Challenge in October 2017 by Niels Moseley, is available on his GitHub account.
https://github.com/trcwm/Speech256
Or CLICK on "2017 FPGA Implementation" above.

---

## Subject: Re: Theoretical Method to read a SPR-xxx External Rom with an Arduino
## Posted by lynchaj on Tue, 16 Apr 2024 18:23:09 GMT

View Forum Message <> Reply to Message

Hi Jay

Check this out, the datasheet for SP0256B has an important clue.  Look on page 2 under test modes.  One of the test modes tell the SP0256 to ignore its on-board ROM and use the external serial speech ROM via SPR000.  All I need now is a datasheet for the SPR000 and/or SFD2000 and I can build a board to emulate a SP0256 using an external ROM.

I might also possibly have a line on an ancient SFD2000.  Maybe the owner will sell it to me for some reverse engineering.  Possibly but I wouldn't bet big money on it.

Also, with some analysis, actually replace the SPR000 with something different (eventually).  Pretty cool eh?

Thanks, Andrew Lynch

### File Attachments

1) SP0256B.pdf, downloaded 202 times

---

## Subject: RPi Pico W wrapped around a SP0250 or SP0256 *Updated 20250202

Posted by jayindallas on Wed, 17 Apr 2024 01:27:54 GMT
View Forum Message <> Reply to Message

I can understand looking at acquiring a SPR000 and SFD2000.

I'm a little more interested in wrapping a RPi Zero W around a SP0256 to virtualize the upstream and downstream interfaces. That would allow me to choose among many SP0256 implementation and ROM images. It would make voice processor database development a lot easier, and flexible. RPi WIFI-SSH connection allows remote-in-house workspace and BT5 allows remote-in-house listening space.
-------------------------------------------------
U.S. vintage GI components:

I saw a website listing General Instrument speech and sound devices that can be found in old electronic products, at garage sales, flee markets, Ebay etc. Examples, to the best of my memory:

Mattel Intellivision contains a GI CP1600 processor and an AY-3-8910 Programmable Sound Generator.
Mattel Intellivoice (optional Intellivision addition) contains one of the GI voice processor (SP0256?).
Mattel Speak-and-Spell (I think) also has one of the GI voice synthesizers.

Update Feb 02, 2025:
CONFIRMED The Mattel Intellivoice module does contain a GI CTS+SP0 set; not but the Radio Shack/Allied part numbers. I found the family Mattel Intelli*.* units and opened them up for photos and chip listings. I'll be posting that information this week.

Later I'll do an Arduino bench hack challenge to download the CTS code to see if its the exact same code as the Radio Shack chips. After that, I'll see if I can hack the data tables out of the SP0 voice synthesizer internal ROM. There is a SP0 internal diagram showing how the internal ROM is serially accessed to shift data serially through all registers until loaded all at once. I recall that there was clearly a serial bit input for external serial ROM, and possibly a serial out bit on the SP0. If the latter is true, I'll try to pass the internal ROM through the registers (serially) without loading/latching them and read them on that serial output pin. By learning how the SP0 commands access and sequence internal ROM into the registers, it should apply closely to external ROM.

I haven't yet opened up the Intellivision EMI protection module to confirm an AY-3-8910 chip inside; I have one in a wire-wrapped S-100 board so there was less need to open it. I'll open the Intellivoice again tomorrow and look for an external serial ROM chip numbers... I didn't look for that when I took the photos and don't really expect to see one. The old RPi photos were poor quality, couldn't get the chip numbers in focus. Guess I need to buy some updated toys.

Ultimately, I'd like to emulate the CTS+SP0 in a RPi Zero 2. Retro Voice would be a more interesting interface as a vocal pace/trip manager for cycling and walking trails. Retro Voice would be simpler when trying to maintain a specific walking pace; my Brother-in-Law challenged me today to match his 4mph walking pace. With a stop-watch I was about 18 seconds late... 3.96039604 mph arg! So now I have need to retro-design a 'pace & path voice manager' unit with bluetooth audio. A display on the bike is easy, but walking, it might be a nuisance. I'm exploring the town's trails and it would be nice to have that voice with some BT music. That has some possible application for the motorcycle too, though I haven't thought that through yet.

## Subject: Re: RPi Pico W wrapped around a SP0250 or SP0256
## Posted by lynchaj on Wed, 17 Apr 2024 11:10:21 GMT

View Forum Message <> Reply to Message

Hi Jay
The info in the SP0256B datasheet is useful whether you use an SPR000 or a custom replacement to interface the serial speech ROM to parallel memory. I think the ideal long term solution would be to recapture the SPR000 design as a CPLD. I think that's doable. I'd really like to implement an SPR000 as discrete components first but that's probably unlikely. The real benefit is there seems to be a clear path ahead for getting the SP0256 to use external ROM for it's info. It might not even require custom CTS256 firmware depending on how you read the datasheet. In theory, any SP0256 could be made to use external ROM and ignore whatever junk was programmed in its internal ROM. That would reintroduce many parts back on the market as usable SP0256s. The marketplace is flooded with surplus "junk" SP0256s with data other than the 59 allophones like the -AL2.

Thanks, Andrew Lynch

## Subject: Arduino lab tool wrapped-around a SP0256 to reverse engineer an unobtainium SPR000
## Posted by jayindallas on Wed, 01 May 2024 15:21:31 GMT

View Forum Message <> Reply to Message

Actually, I think the SPR000, treated as a unknown black-box boundary, can be reverse engineered relatively easy; as a parallel effort to TTL and CPLD. I've been focusing on the SP0256 and SPRnnn serial ROM documentation. While the documentation has serious gaps, I think there is enough information in the bitsavers.org links to General Instrument documents to deduce the interface, at least close enough to converge with some experimentation.

I'm composing a message to explain its basic operations as far as I can at this time... that might help someone else to see the next missing piece or help them first-design more accurately at a SPR000 design target.

As General Instrument designed both parts, they could have (and seem to have...) taken many advantages, such as reading the next ROM/EPROM byte while the previous byte is shifting out toward the SP0256. That's something that needs to be in a TTL or CPLD circuit version, to maintain the speedy transfer. For example, most documentation seems to suggest that the ROM/EPROM data read-in, would be from blocks of sequentially addressed data, that the SP0256 serializes internally to load its registers, maybe even as a parallel load to a set of registers or all registers. So an auto-incrementing address counter, could be starting the next ROM/EPROM read as soon as the previous read was latched into the DSR (Data Serial Register) to be clocked out to the SP0256. The pre-read timing should be implemented into the TTL and CPLD circuit for best realization of the SPR000; emulating serial ROMs with bytewide memory.

Loading the ASR (Address Serial Register) would only need to be done for the first address of the data block to be read and transferred to the SP0256. Thereafter as each subsequent ROM byte is available and transferred into the DSR (Data Serial Register), the ASR is incremented and the next ROM address access time begins. These 'sweet' efficiencies can be designed into GI-to-GI components... and would be enough circuitry to justify the single chip SPR000 to do the same task, at speed, to bytewide memory.
This also fits GI's market strategy at that time; designing single chip replacements for semi-standardized common circuits in TV and other electronic appliances, in an age before pick&place assembly was available. GI's single chip standard circuits improved an electronic appliance companies' manufacturing, inventory, service repairs, design time, and reduced errors characteristic to numerous components being manually insertion to circuit board. There are also advantages of less tuning needed when all components are on the same IC circuit, when compared to numerous discrete components with individual performance within wide specification ranges.

This is why I need to compose that more detailed operations message, so the TTL and CPLD parallel efforts can more closely take the same sort of transfer time advantages that GI likely took. I think after doing that, I might wrap an Arduino around a SP0256 and emulate the upstream and downstream interface to it in software. Useful for testing and converging on a reverse engineering design for the SPR000; but I'm NOT suggesting an using Arduino as a SPR000; it's just a lab-tool.

Note: The SP0256 has, 'effectively,' a single address stack. It seems to be available so a sequential block being read, can be pause and its program counter saved on the stack, to insert a read of another commonly useful addressed sequential block, or segment that block, elsewhere in ROM. This could be used to insert phrases used over and over again.

Question: Does anyone have a Link to a SP0256 internal ROM image or listing? Even some SPRnnn serial ROM images would be useful.

## Subject: Re: Arduino lab tool wrapped-around a SP0256 to reverse engineer an unobtainium SPR000
## Posted by lynchaj on Thu, 02 May 2024 00:09:39 GMT

View Forum Message <> Reply to Message

Hi
There is really good information in the SPR016 datasheet which helps explain how the SPR000 *should* work.  There is a block diagram showing the information flow.  Basically substitute a parallel EPROM (2732) for the internal mask ROM and you're there.

The SP0256B datasheet has some good information on the SP0256 serial ROM command sequences and also timing diagrams.

I think there is enough information to piece together a SPR000 workalike but not enough to reverse engineer the SPR000 itself.  Still no pinout information.

I am hoping to purchase or borrow an SFD2000 and use my VOM to ohm out the PCB and using datasheets for the other chips (74LS138, SP0256, 2732) deduce the SPR000 pinout sufficient to make a test board. Maybe that will work but it is dependent on a series of miracles happening. It's a rather dicey proposition at this point.

The timing requirements of the synchronous 1.56 MHz clock pretty much rules out the MCU approach. Maybe a Raspberry Pi Pico coded in assembler might be fast enough -- maybe -- but it also has to be 5V tolerant at least. I think the CPLD approach is probably the highest probability of success but that's no slam dunk either. The information in the SPR016 and SP0256 datasheets is pretty sparse and may be missing important details.

Best of luck! Thanks, Andrew Lynch

PS, the best info I have on the internal guts of the SP0256A-AL2 found here:
http://spatula-city.org/~im14u2c/sp0256-al2/

## File Attachments

1) SPR016.pdf, downloaded 182 times
2) SP0256B.pdf, downloaded 202 times

---

Subject: Re: Arduino lab tool wrapped-around a SP0256 to reverse engineer an unobtainium SPR000
Posted by plasmo on Fri, 03 May 2024 03:45:16 GMT
View Forum Message <> Reply to Message

Today I received the 2 SP0256AL2 I ordered from 351_windsor. I also have the CPLD protoboard checked out. It is ready to wire in the circuits for SP0256-AL2. I don't have 3.12MHz crystal, so I plan to either divide the 22Mhz Z80 clock by 7 to 3.14MHz or divide the 25MHz oscillator on the protoboard by 8 to get 3.125MHz. The audio amplifier is LM386 which I have many on hand. The immediate goal is have Z80 drive it to make some sound. I should be able to spend an hour or two a day working on this.
Bill

## File Attachments

1) CPLD_protoboard_for_SP0256AL2.jpg, downloaded 131 times
2) two SP0256_AL2_from351_windsor.jpg, downloaded 125 times

---

Subject: Re: Arduino lab tool wrapped-around a SP0256 to reverse engineer an unobtainium SPR000
Posted by lynchaj on Fri, 03 May 2024 11:17:15 GMT

View Forum Message <> Reply to Message

Hi
I got my 3.12 MHz crystal from Mouser although either of your solutions sound like they will be close enough. The ROM clock is the SP0256 clock divided by two.

Smart move confirming the SP0256 works with the Z80 CPU first. Good luck, let me know if there is anything I can do to help.

Thanks, Andrew Lynch

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by plasmo on Mon, 13 May 2024 03:46:55 GMT
View Forum Message <> Reply to Message

I also have ordered 3.12Mhz crystals from Mouser. After a bit of troubleshooting, the SP0256 from 351_windsor is working. Here is a short video of it talking. The breadboard is controlled by Z80 over RC2014 bus. The CPLD is hardly utilized in this configuration, but it will play a more active role when the internal ROM is replaced with an external ROM.
 Bill

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by plasmo on Mon, 13 May 2024 03:51:27 GMT
View Forum Message <> Reply to Message

I think the attachment is too big. Here is a smaller version
 Bill

File Attachments
1) SP0256_talking.mov, downloaded 206 times

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by lynchaj on Mon, 13 May 2024 10:12:46 GMT
View Forum Message <> Reply to Message

Hi Bill
That's great! I gather you are using the SP0256 alone and using only the Z80 with it?

Did you write or found a Z80 program to drive the SP0256 directly?

Looking forward to the next step. First is to get the SP0256 into a test mode to disable the internal ROM.

Then add the logic to the CPLD to emulate the SPR000 chip.

It's described at a high level in the SP0256B and SPR016 datasheets but far from specific implementation.

Best of luck!  Thanks, Andrew Lynch

---

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by plasmo on Mon, 13 May 2024 13:08:15 GMT
View Forum Message <> Reply to Message

The protoboard has CPLD to interface to RC2014 bus where I/O address $ee selects allophone value and $ef monitor SP0256 status.  The SP0256 is currently wired up to use internal ROM, but there is room to handle external ROM.  I'll keep the internal ROM configuration for a few days since I have 3 SP0256 coming from China which are most likely fake.  After that I'll wire it for external ROM configuration.

The SP0256 datasheet published by Archer has a section on Allophones and provided examples for 250 or so words, so the Z80 program just copies the allophone examples of days of the week.
  Bill

## File Attachments

1) SP0256_proto_comp.jpg, downloaded 550 times
2) SP0256_proto_solder.jpg, downloaded 180 times
3) say_days_of_week.asm, downloaded 186 times

---

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by lynchaj on Mon, 13 May 2024 16:56:05 GMT
View Forum Message <> Reply to Message

Hi Bill
That looks great!  Yes, the cheap SP0256s from AliExpress are nearly all fakes.  Some of those are non-AL2 SP0256A-xxx but some are just junk like OTP ROM or other 28 pin DIP chips.

The good news is if you get a non-AL2 SP0256A-xxx, we should be able to get it to work with the external EPROM and then it won't matter.  I'll bet those scammers will be surprised!

Thanks, Andrew Lynch

---

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by plasmo on Tue, 14 May 2024 03:10:23 GMT
View Forum Message <> Reply to Message

I received the 3 "SP0256-AL2" from China today.  Tried them out in known working hardware and none of them are working, no sound generated at all. The current consumption is lower and the working hardware continue to work OK afterward. I tried different allophone values associated with SP0256-012, -017, -019, -021, but no sound were generated. What else can I try? I would like to know whether it is a variant of SP0256.

Bill

File Attachments

1) DSC_76860513.jpg, downloaded 549 times
2) DSC_76870513.jpg, downloaded 524 times

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by lynchaj on Tue, 14 May 2024 10:17:15 GMT
View Forum Message <> Reply to Message

Hi Bill

One thing you can try is using the monitor and feeding allophone addresses directly to the SP0256. Also, put in an LED on the BUSY pin on the SP0256 which would light up during speech.

If you're not seeing any activity at all even on allophone 0, 1, 2, etc. or BUSY LED then you probably have a non-SP0256 fake like a relabeled OTP ROM. You could try identifying the chip in your programmer and see if that gives some clues. Try to read the contents like a 2764 EPROM. But they are mostly electronic junk

Only some of the fake SP0256s are relabeled SP0256A-xxx chips. They do that to give something that sort of works but not like you expect it would. Other fakes are like what you probably have which are just relabeled junk ICs like OTP ROMs or MCU with mask ROMs.

Here is an article on fake SP0256 chips. I think you'll recognize these parts

https://www.smbaker.com/counterfeitfakejustplainbad-sp0256a- al2-chips

Oh, how I detest the scammers!

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by plasmo on Wed, 15 May 2024 03:07:34 GMT
View Forum Message <> Reply to Message

Thanks for the link to Scott Baker. My non-functioning parts appeared in his write up, and they do not appear to be any variants of SP0256. I'll return them for refund.

I'll go ahead with external ROM implementation using the existing SP0256.

Bill

---

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by lynchaj on Wed, 15 May 2024 10:43:48 GMT
View Forum Message <> Reply to Message

Hi Bill
Yeah, the fake SP0256 chips are kind of a hit-or-miss proposition.  Sometimes you get remarked SP0256A-xxx chips and sometimes you get just non-functional junk.

I think if AliExpress wants to stay a credible source of parts, they need to crack down on these blatant fakes.  SP0256 and V9958 seem to be the worst I've seen.

I doubt it is limited to them though.

Thanks, Andrew Lynch

---

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by plasmo on Thu, 16 May 2024 14:57:52 GMT
View Forum Message <> Reply to Message

Moving forward, I grounded the 5 high order addresses of W27C512 to convert it into a 2KB ROM and wired all addresses, data, and control of the 2KB ROM to CPLD.  This way CPLD has complete control over the ROM.

The translation of serial ROM protocol of SP0256's C1, C2, C3, SerIn, SerOut, ROMCLK, and ROMDisable to parallel ROM are done inside the CPLD, specifically the ROMCntrl block.

The C1, C2, C3 signals are decoded at the rising edge of ROMCLK into ASRLD, PCLD, DSRLD, and DSRSH commands.  The other 2 commands, STACKLD and RETURN are not decoded because I don't think they are in use.  ASRLD command causes 16-bit shift register to be loaded with addresses; PCLD command causes lower 12-bit of the 16-bit shift register to be loaded into a 12-bit counter; DSRLD generates ROM chip select and load contents of ROM into a parallel-to-serial register and then post increment the counter; DSRSH command shift the parallel-to-serial register out of SerIn output pin.

I have not tested the above circuit in the prototype board, but did a test compilation to know the macrocell utilization is 60 macrocells which will fit in a small 64-macrocell CPLD in PLCC44 package.  With the current CPLD, I have resources to read the contents of 12-bit counter, parallel-serial register, and various commands for debugging purpose.

Bill

## File Attachments

1) SP0256_with_external_ROM_comp.jpg, downloaded 484 times
2) SP0256_with_external_ROM_solder.jpg, downloaded 107 times
3) ROMCntrl.pdf, downloaded 184 times

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by lynchaj on Fri, 17 May 2024 11:14:46 GMT
View Forum Message <> Reply to Message

Hi Bill
Wow! That's great news! Looks good

Regarding ASRLD and PCLD commands, the SP0256B datasheet goes into detail of its implementation. I believe it needs to be a 16-bit counter for the address bus based on the text. Below is a snippet from the datasheet

Am I correct in assuming the current design will fit in a EPM7064 but not an EPM7032 due to using 60 cells?

It's not clear to me what STACKLD or RETURN are even used for so they may not be included in the SP0256 implementation. Maybe they are leftovers for some other purpose?

Awesome progress! Thank you! Andrew Lynch

PS, maybe you're already doing that. What is component inst12? It looks like a 16-bit D-flip flop maybe serial to parallel converter? The design seems amazingly simple in the schematic layout. Many fewer parts than I anticipated.

PPS, I am guessing inst27 component is a 74LS74 D-flip flop. Is inst12 two 74LS164s chained together to make a 16-bit serial to parallel converter? What is inst13? three 74LS191s chained together? 74LS161 or 74LS163s? This is very interesting.

Can't wait to see how this does in test!

## File Attachments

1) SP0256B External and Internal Control Lines.png, downloaded 472 times

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by lynchaj on Fri, 17 May 2024 12:29:41 GMT
View Forum Message <> Reply to Message

Hi Bill

Here is a whacky idea: would it be possible to use a chip like a PIC16F13145 with CLCs (configurable logic cells) for the logic and built in Flash to store the SP0256 ROM data?

https://www.eejournal.com/article/tiny-fpga-in-microchips-la test-pic16-microcontroller-adds-real-time-speed-sells-for-le ss-than-50-cents/

Put the logic in the CLCs and store the data in the Flash. Then it would be only a single 20 pin DIP part for both logic and ROM data. Burn the PIC16F in your programmer in one operation.

Maybe it would work? Thanks, Andrew Lynch

---

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by plasmo on Fri, 17 May 2024 19:06:34 GMT
View Forum Message <> Reply to Message

inst12 is a short-hand version of 16-bit serial-to-parallel converter; I redraw it in updated ROMCntrl circuit. I have also corrected other mistakes in the updated ROMCntrl. I've powered it up and it generated chirps and beeps, but not at all close to the original sounds with internal ROM. So I need to delve deeper into the ROM contents and make sure all the bit order are correct for the address & data shift registers.

inst27 is like a 7474 D-flipflop and inst13 is a 12-bit loadable counter. inst13 is generated with Quartus's megafunction tool that can generate many common circuits like adder, counter, comparator, etc.

The circuit in current form should fit 64-macrocell CPLD like EPM7064S or ATF1504, but won't fit 32-macrocell CPLD like EPM7032S or ATF1502.

PIC16F13145 with CLCs sounds like a really useful capability except I have no experience with PIC.
Bill

## File Attachments

1) ROMCntrl.pdf, downloaded 213 times
2) shift16.pdf, downloaded 202 times

---

Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
Posted by lynchaj on Sat, 18 May 2024 01:49:13 GMT
View Forum Message <> Reply to Message

Hi Bill,
Wow! Incredible progress!

How are you getting the SP0256 into T0 test mode? There are two ways described in the SP0256B datasheet. One is to set the Test pin high and put certain address pins low. Another way is to set Test pin high and pulse the reset pin. It's T0 mode we want since that disables the internal mask ROM.

Thanks! Andrew Lynch

## Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
## Posted by plasmo on Sat, 18 May 2024 04:31:20 GMT

View Forum Message <> Reply to Message

Andrew,
I have a jumper that set TEST high or low. When low, SP0256 uses the internal ROM and when high, SP0256 uses the external ROM. I can see external ROM enabled and data shifted to SP0256, but apparently my data is not correct; still troubleshooting.

I looked at PIC16F13145 data sheet and see it does have 32 LUT which is not enough to implement the logic in current CPLD. However, it may be possible to anticipate what SP0256 is going to do, so the LUT only need to capture the incoming 12-bit addresses then proceed to look up the corresponding data and stuff it to the data shift register which will be clocked out by SP0256. Meanwhile PIC can look up next address and get ready to stuff the data shift register as soon as the 8th data bit is shifted out. So this approach will only use 20 LUT. It may be possible to use PIC16F13145.
  Bill

## Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
## Posted by lynchaj on Sat, 18 May 2024 10:27:34 GMT

View Forum Message <> Reply to Message

Hi Bill
Yes, the PIC16F13145 is pure speculation at this point but something to consider later on. Should focus on getting the CPLD proof of concept working first. That you have activity on the serial ROM port is very good news indeed

Thanks, Andrew Lynch

## Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
## Posted by lynchaj on Sat, 18 May 2024 11:29:52 GMT

View Forum Message <> Reply to Message

Hi Bill

According to the SPR016 datasheet, figure 2 shows the bit order of the address 16-bit value sent over the serial line starts with A15 and ends with A0. The data byte sequence is reversed and starts with D0 and ends with D7. Why two different bit orders in the same interface? Who knows but if there is a problem with the interface, my number one suspect is bit order.

My suggestion is to try sending the zeroth allophone to the SP0256 and see what sort of address it produces. It should a short, silent pause.

Who knows what sequence they used to capture the SP0256 internal mask ROM so it may just take some experimentation to get it right. Are you able to monitor the serial bits using a logic analyzer? That may be the way to figure this out.

The serial speech ROM interface is just weird. I think it is solvable but seems internally inconsistent.

Thanks, Andrew Lynch

---

## Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
## Posted by lynchaj on Sat, 18 May 2024 12:55:57 GMT

View Forum Message <> Reply to Message

Hi Bill
Are you using Quartus schematic capture for your design? Would you mind converting your design inst12, inst13, and inst27 to regular 74LSxxx chips and I will reimplement in Digital. Then use the simulator function to follow along with your design. Digital also has Verilog and VHDL export functions which can be re-imported into Quartus. Maybe I can use the different tool to help debug?

Thanks, Andrew Lynch

---

## Subject: Re: Has anyone reverse engineered SP0256A external serial speech ROM
## Posted by berzerkula on Tue, 16 Jul 2024 07:01:55 GMT

View Forum Message <> Reply to Message

Greetings Andrew,

Last I can find is from Designing With Speech Processing Chips book by Ricardo Jiménez. He did reference Microchip publication DS5007A-1 1984 but can't find that document anywhere.

Sincerely,

William

## File Attachments

1) image.png, downloaded 114 times