Final Project, Part II (revised Nov 16 and 19)

65 Points Possible

12/9/2025

Attempt 1

In Progress
NEXT UP: Submit Assignment

Add Comment

Unlimited Attempts Allowed
11/1/2025 to 12/9/2025

Details

Nov 16: The CT lab has stock of the 2764 EPROM, and not the 27C64. There are **some* differences between the two, in the PROGRAMMING* of the device.

The STMicro and Intel datasheets of the 2764 have practically the same content, with the PDF from STMicro much 'cleaner'. Use this, and keep in mind that the discussion in this original post (including the programmer power supply circuit) is based on the 27C64. ***It is left as a real-world exercise for you to modify the circuit and write the programming algorithm to suite the*** 2764. Rest assured the modifications are **pretty simple! -- go over to the "Discussions" to see some details about this -- (cyo - Nov 19)**

2764.pdf (https://dlsu.instructure.com/courses/219256/files/29146120?wrap=1) (https://dlsu.instructure.com/courses/219256/files/29146120/download?download_frd=1)

The eagle-eyed may notice that there is a basic incompatibility between Voh of the 2764 and Vih of the HC11 (if you haven't.. well....) - This can be remedied by adding a pull-up resistor for each data bit. Based on calculations (within safe current limits), ***2.7Kohm*** should work.

--- original post follows ---

Proceeding with Part II **requires Part I to be successful**.

Part II will tackle the actual programming (writing of data bytes) into the EPROM. Generally to do this:

1. The EPROM is verified to be blank (i.e. read the EPROM and ensure that the data-byte in all locations = FFh

2. Follow the recommended programming algorithm (sequence) to send data, one at a time, to each address of the EPROM.

3. Upon writing all the required data, putting the EPROM back into "read" mode, and reading all the data back and verifying it matches.

Course Chat

For this project, only 25 bytes will need to be programmed (
location 000h to 018h.

Chat is not connected.

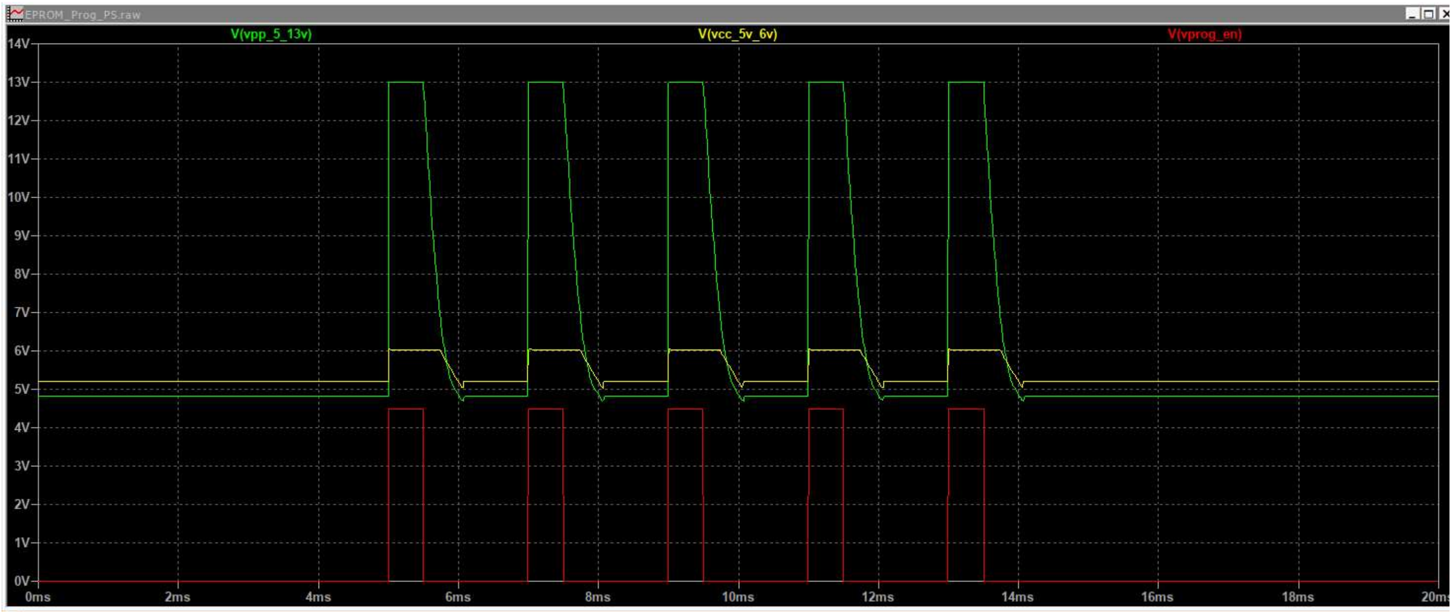
Programming the EPROM requires a **special sequence / proc**
applied to the Vcc and Vpp pins of the EPROM. The necessa
specifically in the "Programming" section. [27C64.pdf](https://dlsu.instructure.com/courses/219256/files/28969083)
(https://dlsu.instructure.com/courses/219256/files/28969083?wr
(https://dlsu.instructure.com/courses/219256/files/28969083/do

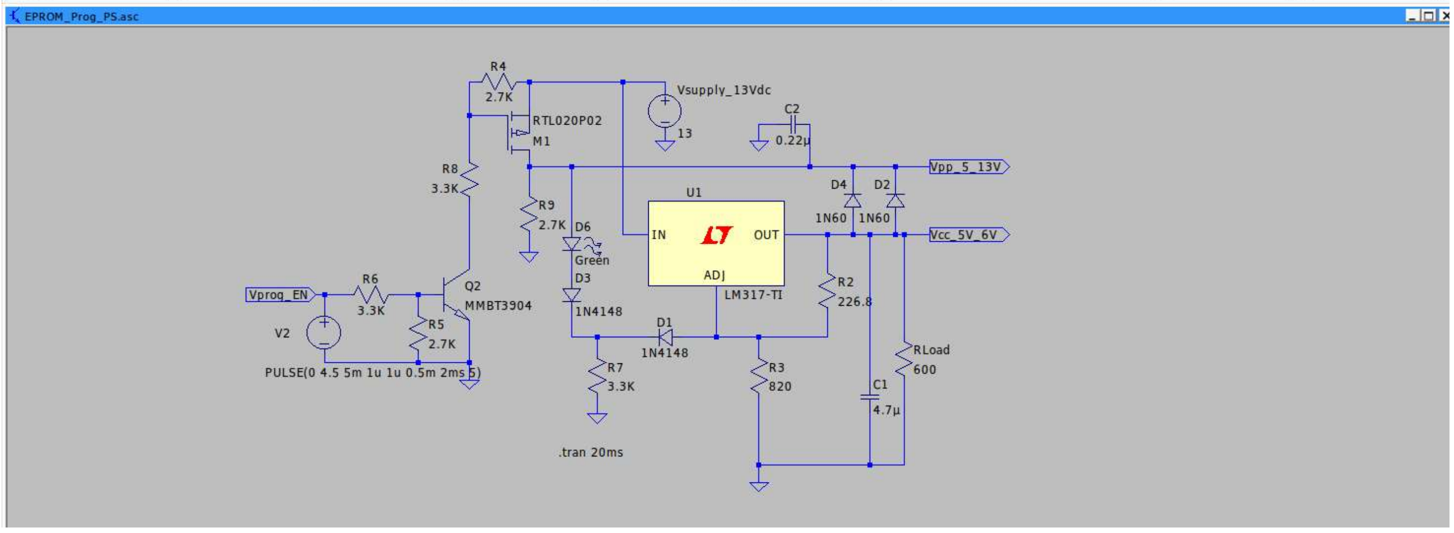
Designing the power source needed might be beyond what y
time, so I've decided to provide it. UNDERSTAND the 27C64
look into these files:

[EPROM_Programmer_PSU_Schematic_CYO.pdf](https://dlsu.instructure.com/courses/219256/files/28969095)
(https://dlsu.instructure.com/courses/219256/files/28969095?wr
(https://dlsu.instructure.com/courses/219256/files/28969095/do

Send

Here is a simulation of the PSU:





As you can see - the "Vprog_EN" signal is expected to come from the HC11 - it tells the power supply circuit to produce the required programming voltages for the 27C64 (13V for Vpp, and 6V for Vcc). A logic "1" (high) on the Vprog_EN pin (red trace) in the simulation shows a corresponding 13Vdc on the Vpp output (green trace), and 6Vdc on the Vcc output (yellow trace).

A logic "0" (Low) on the Vprog_EN pin in the simulation shows a corresponding 4.8Vdc on the Vpp output (green trace), and 5.2Vdc on the Vcc output (yellow trace).

-- you should be able to VERIFY that these voltages follow the specs in the 27C64 datasheet --

If you wish to simulate / look further into the circuit, here is the LTSpice file:

EPROM_Prog_PS.asc (https://dlsu.instructure.com/courses/219256/files/28969104?wrap=1) (https://dlsu.instructure.com/courses/219256/files/28969104/download?download_frd=1)

Building the Power Supply circuit - here's a bunch of Kicad files (version 8, which can be imported without issues by current version of Kicad (9.0+)...

Kicad_v8_EPROM_Prog_PS.zip (https://dlsu.instructure.com/courses/219256/files/28969122?wrap=1) (https://dlsu.instructure.com/courses/219256/files/28969122/download?download_frd=1)

The ZIP file contains the schematic, project file, netlist and a BLANK PCB, with the components imported into the board, but not laid-out (and no track connections). The manufacturing limitations have been set in this PCB file, so USE IT rather than make one from scratch! Complete the layout and make the PCB for yourself. The track widths are not critical, but DO NOT GO BELOW 20 mils as the CT lab cannot fabricate tracks that are too thin. It will probably be OK if make your board with 30-mil tracks for everything (feel free to use something wider, specially for the power input).

Note that the circuit design and components have been carefully selected to be the cheapest possible from e-Gizmo, or available in the CT Lab (free).

Here are the datasheets for the semiconductors used:

P-channel MOSFET, surface-mount SOT-23 package - Used to pass 13Vdc to the Vpp pin of the EPROM. **TM2301GN-VB.pdf (https://dlsu.instructure.com/courses/219256/files/28969149?wrap=1) (https://dlsu.instructure.com/courses/219256/files/28969149/download?download_frd=1)**

LM317-L, Variable Voltage Regulator - Low power version, TO-92 package. Used to provide 5V and 6V to the Vcc pin of the EPROM. **lm317l.pdf (https://dlsu.instructure.com/courses/219256/files/28969157?wrap=1) (https://dlsu.instructure.com/courses/219256/files/28969157/download?download_frd=1)**

SST3904, NPN general purpose transistor (2N3904 equivalent), surface-mount SOT-23 package - used to switch the PSU circuit to output Programming mode voltages or "ordinary - read" mode voltages.

To accomplish this - you will use the HC11 board and ADD the necessary connections / hardware to allow EPROM programming (starting point will be what you developed / got working for READING the EPROM). You will need to ADD the necessary HC11 code to follow the required EPROM programming algorithm. It is expected that you'll also slightly modify the Python PC program (so that it includes prompts like "Will run the blank check now - press a key to continue.. " etc...

Functionality / Demo expectations:

0. The EPROM will be erased (c/o the CT-Lab - we have an EPROM eraser or two, which shines UV-light into the EPROMs (several chips at a time is possible). Erasing takes about 15 minutes.

1. Do a blank-check (reject the eprom if it isn't blank) - only for the first 25 bytes

2. Program the binary pattern given to you (it will be UNIQUE for each group), 25 bytes large

3. Read back the 25-byte binary pattern and display it on screen, as a sequence of HEX digits and ASCII characters.

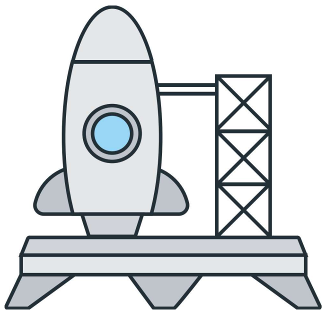
Note: Only PRINTABLE ascii characters should be displayed (generally: A-Z, a-z, 0..9, punctuation mark characters).

The usual documentation of "everything" is expected (and to be submitted).

Keep in mind, this submission will count for everyone in your Final Project Groups group.

Choose a submission type

Upload ARC More



Choose a file to upload

File permitted: PDF

or

Canvas Files

Submit Assignment