Cal Poly Pomona

EPROM Chip input squaring, to 2 7-Segment Displays

Experiment 8

Steven Keesler, Andrew Riad
10/17/2025

0.0 PRELAB

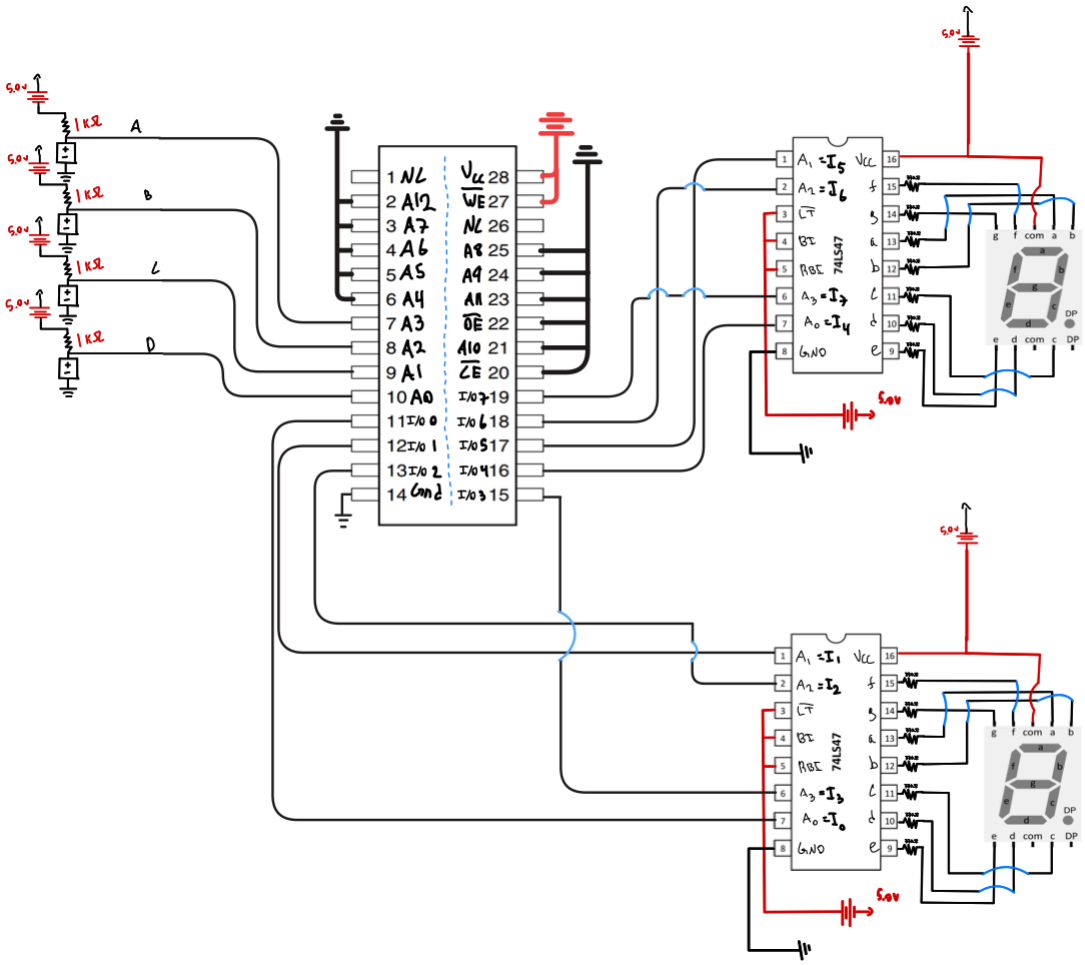
0.1 Design a 2732-based EPROM system to display the square of a BCD digit on seven segment displays. Use a minimum number of 74LS47 BCD to seven segment converters. Each BCD digit will be input to the EPROM via DIP switches.

In table 0.1 we will have a truth table for the desired EPROM inputs and outputs, and how they will be shown in 7 segment display 1, F1 (Inputs 0-3), & in our 7 segment display 2, F2 (Inputs 4-7). In figure 0.1, we will have the schematic for our desired circuit.

Table 0.1 : Truth Table

Input					Output		
D	C	B	A	Decimal	Decimal	F1	F2
0	0	0	0	0	00	0	0
0	0	0	1	1	99	9	9
0	0	1	0	2	04	0	4
0	0	1	1	3	09	0	9
0	1	0	0	4	16	1	6
0	1	0	1	5	25	2	5
0	1	1	0	6	36	3	6
0	1	1	1	7	49	4	9
1	0	0	0	8	64	6	4
1	0	0	1	9	81	8	1

Figure 0.1 : 1 EPROM -> 2 7 segment displays



NOTE: EPROM is in read mode, WE goes to power, while OE & CE go to ground, as pulled from its data sheet.

1.0 **INTRODUCTION**

Implement the above circuit from the pre-lab using a minimum number of decoder and EPROM Chip. Demonstrate the operations using switches & Seven segments display etc. as needed. Schematics will be pulled from Pre-Lab when needed.

2.0 **OBJECTIVES**

The goal is the following:

1. Generate the square of the BCD decimal equivalent from the EPROM outputs.
2. Encode these EPROM outputs to BCD and drive a 7-segment via 74LS47 .
3. Specifically show the number 99 for our group number, so 0001 will be a 99 as we are group 1.
4. Verify operation across all 10 input combinations.

3.0 **REQUIREMENT**

Implement the EPMROM and the 2 7-segment decoders so that the square of the outputs to the inputs BCDs decimal equivalent are the input squared. This is shown in the truth table below, pulled from table 0.1 in the pre-lab. NOTE: Due to use being Group one, our output for 1 will not be 1, but 99.

Table 3.1 : Truth Table

Input					Output		
D	C	B	A	Decimal	Decimal	F1	F2
0	0	0	0	0	00	0	0
0	0	0	1	1	99	9	9
0	0	1	0	2	04	0	4
0	0	1	1	3	09	0	9
0	1	0	0	4	16	1	6
0	1	0	1	5	25	2	5
0	1	1	0	6	36	3	6
0	1	1	1	7	49	4	9
1	0	0	0	8	64	6	4

1	0	0	1	9	81	8	1

4.0 **PARTS LIST**

The Parts list for this experiment is shown below in Table 4.1

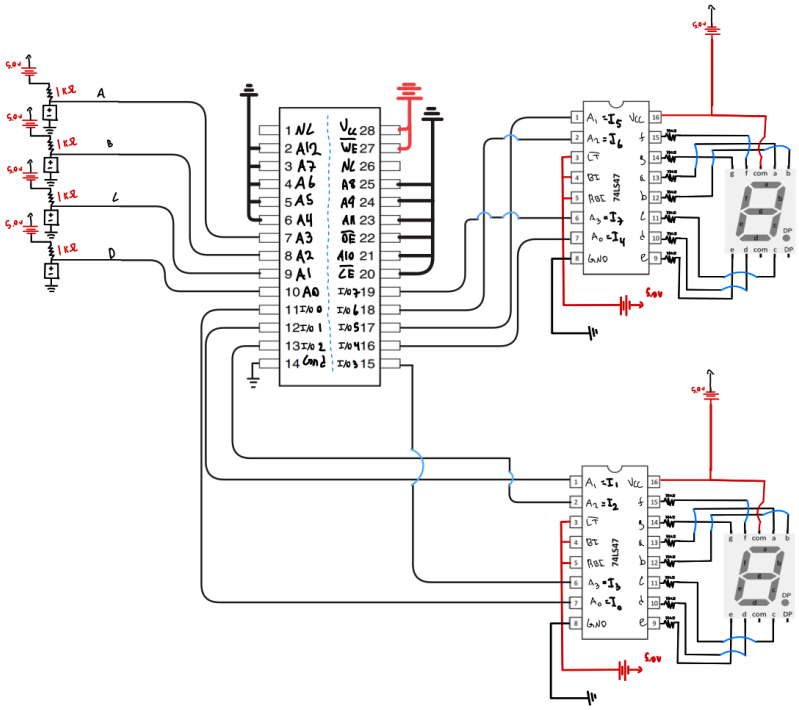
Table 4.1 - Parts List

Item No.	Part Number	Function	Quantity
3	AT28C64	EMPROM	1
2	74LS47	BCD-to-7-Seg	2
5	7-Segment	Output	2
6	DIP Switch	Input	1

5.0 **DESIGN/IMPLEMENTATION**

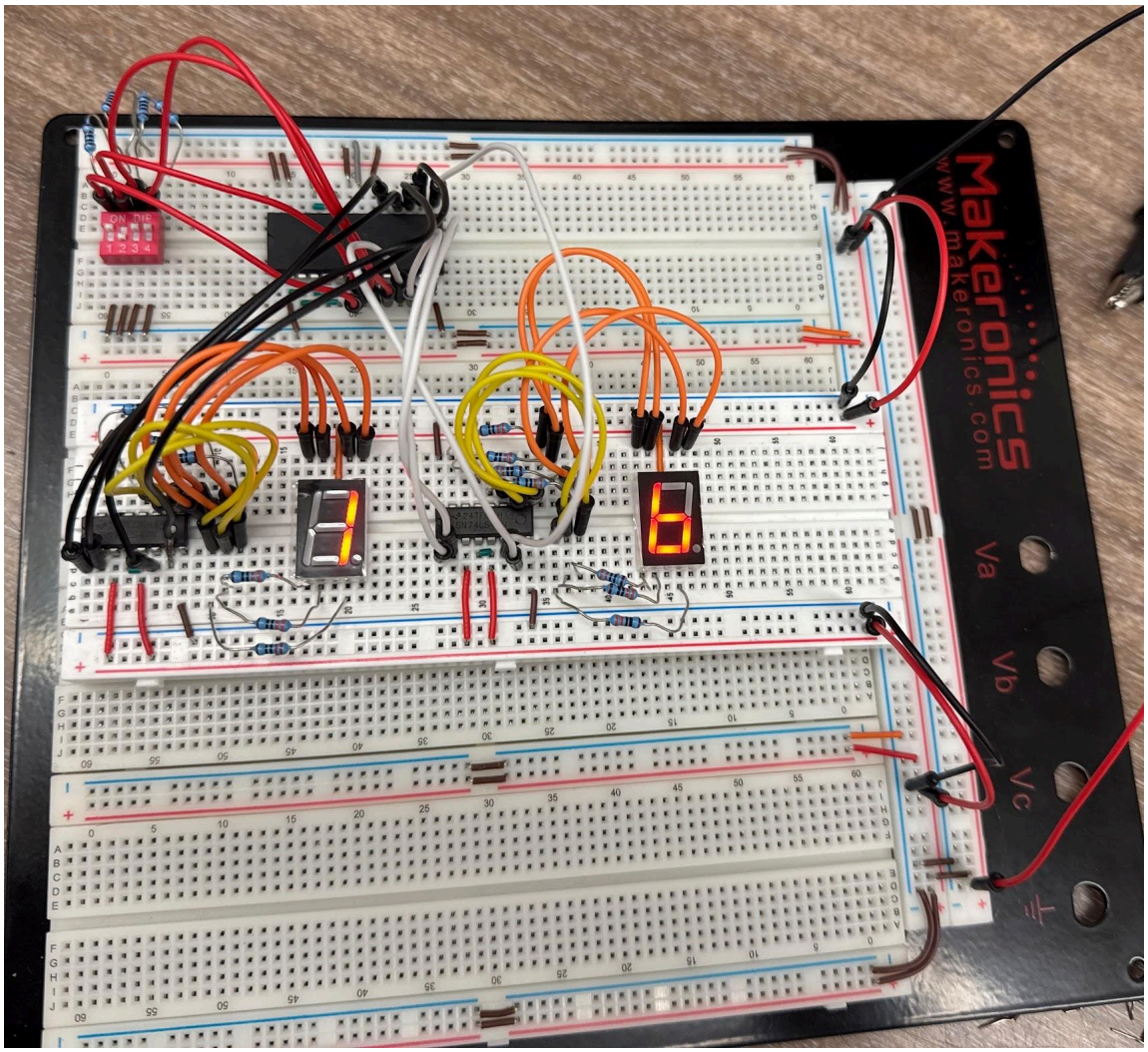
The design from figure 0.1 in the prelab will be tested, and it will be shown again below in figure 5.1. A dip switch will be used to provide the inputs to the EPROM CHIP, and the outputs will be displayed using 2 7-segment displays.

Figure 5.1



6.0 **TEST SETUP**

Below is an image of the breadboard the DIP switches provide D,C,B,A. The EPROM implements the aforementioned operation as described. We're using inputs 0001 (1) & 0100 (4) below, which show 9 9 & 1 6 respectively as our examples. The breadboard photo below shows both the decoder and the display/LED interface.



7.0 **TEST RESULTS/VERIFICATION MATRIX**

The test results are summarized below in Table 7.1:

Table 7.1 - Truth Table of Circuit

Input					Output		
D	C	B	A	Decimal	Decimal	Display 1	Display 2
0	0	0	0	0	00	0	0
0	0	0	1	1	99	9	9
0	0	1	0	2	04	0	4
0	0	1	1	3	09	0	9
0	1	0	0	4	16	1	6
0	1	0	1	5	25	2	5
0	1	1	0	6	36	3	6
0	1	1	1	7	49	4	9
1	0	0	0	8	64	6	4
1	0	0	1	9	81	8	1

8.0 Conclusion

The EPROM chip and circuit operated correctly, showing the square of each input from 0-9 on both displays. Using the EPROM chip, we generated 2 sets of BCD inputs to the 74LS47 and the 7-segment displays. The measurements across all 10 input combinations matched the expected behavior from Table 3.1 as shown above in Table 7.1.

9.0 Post Lab

1. What is the basic differences between :

(a) Volatile and nonvolatile memories.

- Volatile: Loses data when power is removed (e.g., SRAM, DRAM).

- Nonvolatile: Retains data without power (e.g., ROM, PROM, EPROM, EEPROM, Flash).

(b) EPROM and EEPROM.

- EPROM (Erasable PROM): Erased with UV light through a quartz window; erase is whole-chip; programmed electrically in a programmer; typically read-only in-circuit.

- EEPROM (Electrically Erasable PROM): Erased/programmed electrically, byte/page level, in-circuit reprogrammable; no UV window required; slower writes than reads but much more convenient.

(c) SRAM and DRAM.

- SRAM: Uses flip-flops; no refresh; fast; low density; higher power per bit; used for caches/buffers.

- DRAM: Uses capacitors; requires periodic refresh; higher density; lower cost per bit; slower; used for main memory.

(d) PLD and ROM.

- PLD (Programmable Logic Device): Implements arbitrary logic functions via programmable AND/OR (or LUT) structures; outputs are functions of inputs (combinational/sequenced), not stored data per address.

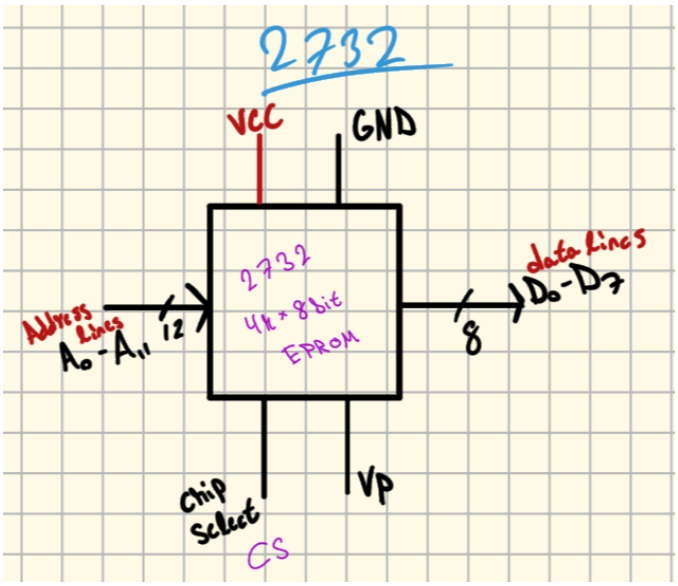
- ROM: Stores a fixed truth table (address in → data out). It can realize combinational logic but as a look-up of preprogrammed contents.

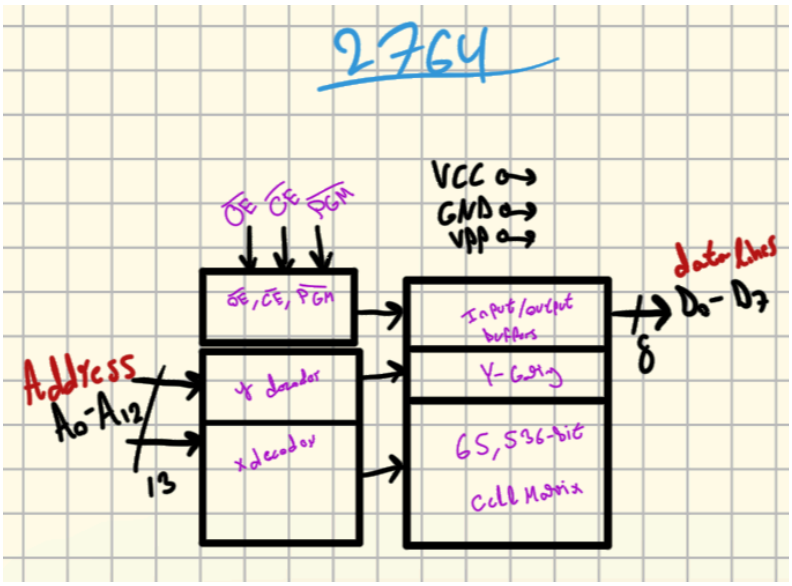
(e) CPLD and FPGA.

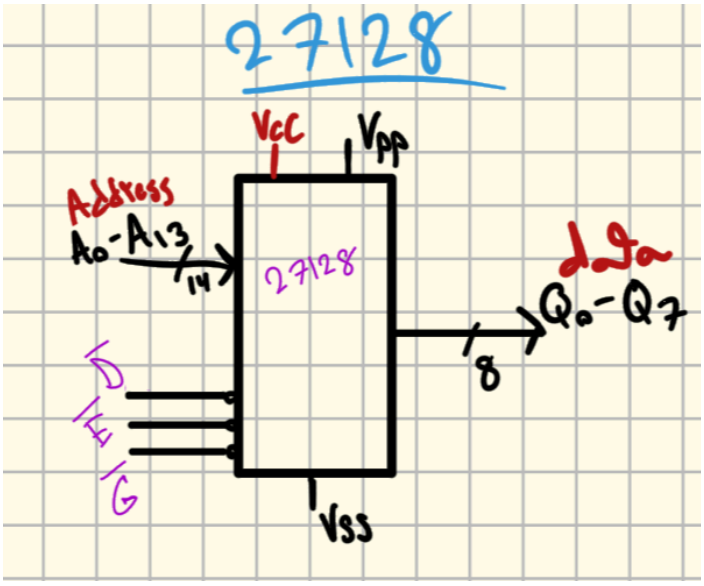
- CPLD: Coarse-grained macrocells with predictable timing; nonvolatile or instant-on; great for glue logic/decoding; lower logic capacity.

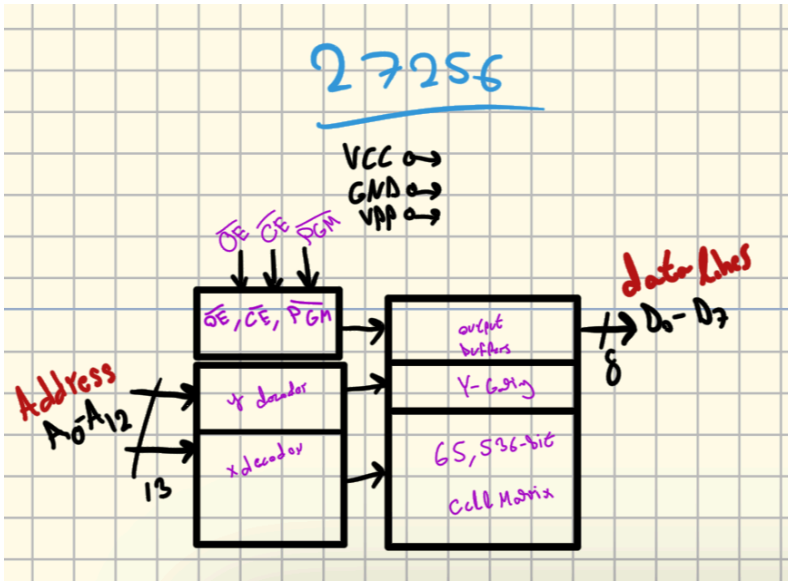
- FPGA: Fine-grained LUT fabric with vast capacity, rich DSP/BRAM/SerDes; usually SRAM-based (needs configuration at power-up); better for complex/datapath designs.

2. Draw block diagrams showing enamel, address, and data lines for each of the following: 2732, 2764, 27128, and 27256.



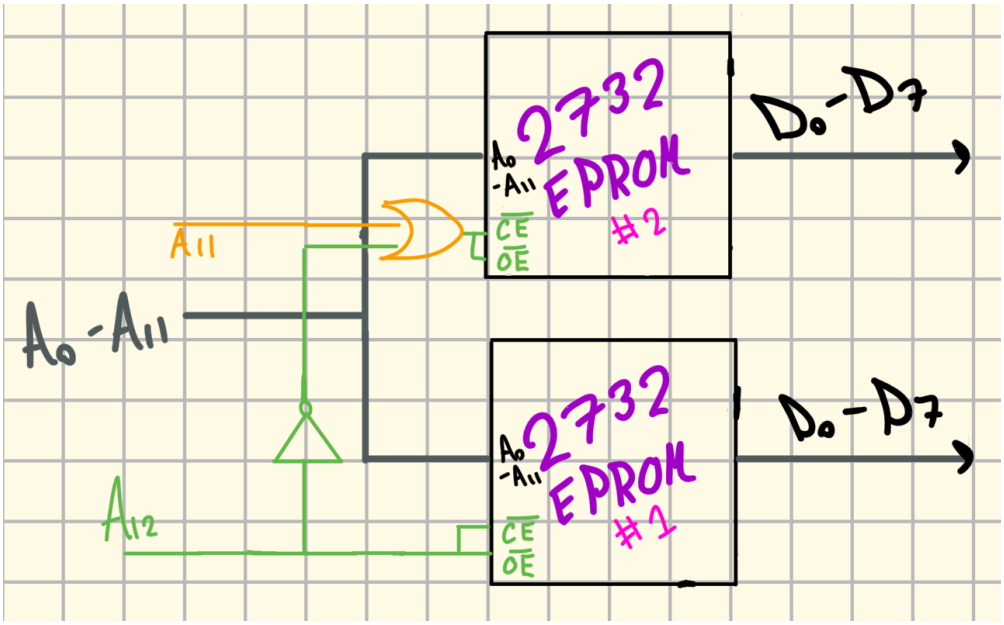






3. Draw a schematic for a 6k X 8-bit memory array using a minimum number of 2732's. What are the highest and the lowest memory addresses (in hex) in the array?

Lowest memory address: **0000h**
Highest memory address: **17FFh**



6 kB Address	First		Second	
A_{12}	0	0	1	1
A_{11}	0	1	0	0
A_{10}	0	1	0	1
A_9	0	1	0	1
A_8	0	1	0	1
A_7	0	1	0	1
A_6	0	1	0	1
A_5	0	1	0	1
A_4	0	1	0	1
A_3	0	1	0	1
A_2	0	1	0	1
A_1	0	1	0	1
A_0	0	1	0	1