5.5 Programmable Logic array types

5.5.1 The FPLA

PLA's come in many varieties. The **FPLA type** has **two Programmable areas**; the **OR array** and the **AND array**. This is the **BASIC PLA**. **Once you program it, you can't change it**. A single **FPLA** can reduce the total parts count in a design by combining many logic functions in the circuit in a single chip package.

Design Rule for an FPLA

When designing the circuit it needs to be noted that the primary limitation is the total number of product terms, therefore, when minimizing the switching functions, minimizing the number of PRODUCT TERMS is the primary goal. Not the # of literals

A search for Compromises!

Sometimes, when it is desired to **reduce the cost** of the device, or to **increase its speed**, the **fuses are omitted from either the 'AND-array' or the 'OR-array'**, leaving the array in some prearranged fixed configuration. This creates a few **sub-classes of PLA's.**

5.5.2 PROM (Programmable Read Only Memory)

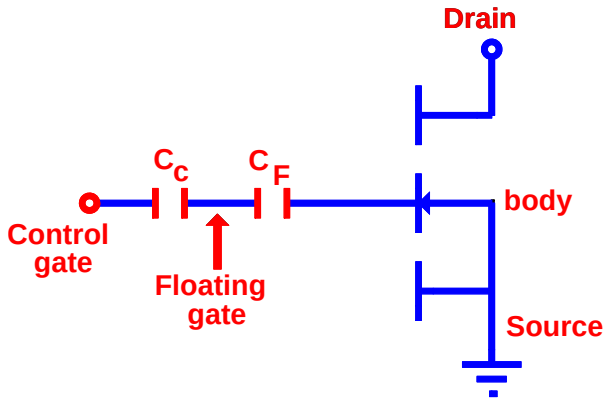
When the **AND-array is fixed**, only the provided **product terms are available** but the **OR-arrays are programmable**. These devices are known as **Programmable Read Only Memories (PROM's)**. They are the **oldest variety of programmable logic devices**. The **AND-Array** generates all 2^n possible min-term products of its (n) inputs. The **OR-array**, which is programmable, allows any combination of product terms in each sum term. '**AND**' **gate inputs with a blown fuse float to a logic "1".** They come in several different varieties. **The standard PROM is programmable with high currents.**

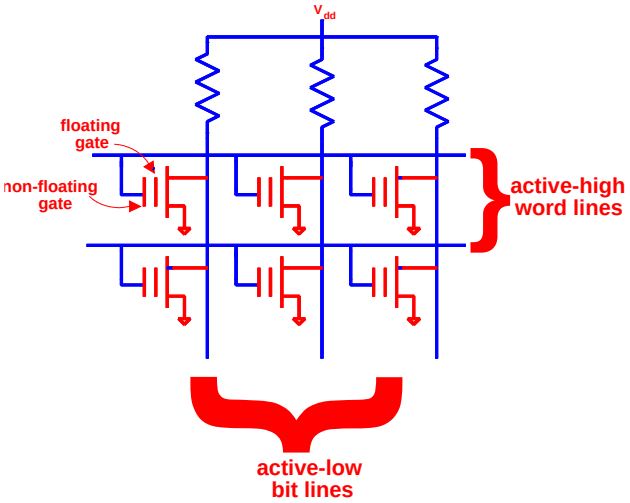
5.5.2.1 The EPROM (Erasable PROM)

A **subset of the PROM** is the **EPROM (Erasable PROM)**. This is Programmable like a PROM, but the **programming can also be erased to an all 1's state by exposing the gate structure to ultraviolet light at a wavelength below 400 nm.** **No, the light doesn't grow the fuses back!** The programming technology used in this type of device doesn't use fuses at all. It uses a technology called **"floating-gate" MOS**. The figure on the next page shows an illustration of this type of gate.

5.5.2.1.1 Floating Gate MOS Technology

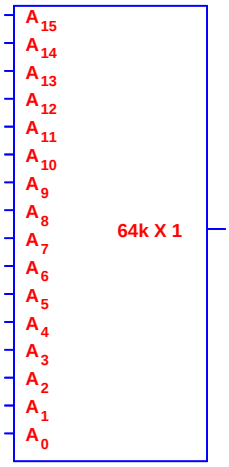
Every bit location has a **floating gate MOS transistor**. Each transistor has two gates. The **floating gate is unconnected and is surrounded by extremely high-impedance insulating material**. To program the EPROM, the programmer **applies a high voltage to the non-floating gate at each bit position where a 0 is to be stored**. This causes a breakdown in the insulating material and allows a negative charge to accumulate on the floating gate. When the HIGH voltage is removed, the negative-charge remains. During subsequent read operations, the negative-charge prevents the MOS transistor from turning on when it is selected.





5.5.2.2 EPROMS Continued

EPROMS will retain 70% of their charge for at least 10 years. However, it can be erased. The insulating material surrounding the gate becomes slightly conductive if it is exposed to ultraviolet light. Thus, **the EPROMS can be erased by exposure to this light for about 20 minutes.** This chip will have a window in it to allow the light in. This window should be covered with tape, to prevent sunlight or room light from degrading the programming.[1]



Q: **What does "Having a 64k PROM or EPROM" mean?**

A: **It doesn't mean 64000 address unless it's a 64k X 1 device like this.**

You could also have a **32k x 2** or a **8K x 8** and still have **64000 storage locations (but not addresses).**

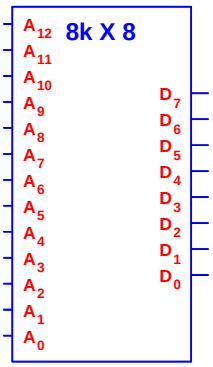
Q: In the manufacturing process, it is **easier to double the left side (address side)** of the chip to increase memory than it is to **double the right side (data side). Why?**

A: 1 address line added doubles the memory. You would have to double the # of pins available on the data side to double the memory.

[1] Wakerly, John F., Digital Design: Principles and Practices. 2nd ed. pp 734-735, Prentice Hall, 1994

Q: **What would an 8k X 8 look like?**

A: It would look like the figure to the right. Note the 8 outputs.



Q: How much memory does a chip with 11 lines actually have?

A: $2^{11} = 2048$ **But it is still known as a 2k device.**

# of addresses	# of address pins
256	**8 lines**
1k	**10 lines**
2k	**11 lines**
4k	**12 lines**
8k	**13 lines**
64k	**16 lines**

5.5.2.3 The EEPROM or E^{2}PROM

A third type of **PROM** is the **EEPROM or E^{2}PROM**. It's a lot like the **EPROM** but you can **erase and rewrite selected addresses electrically**. The floating gates in an **EEPROM** are surrounded by a much thinner insulating layer, and can be erased by applying a voltage of the opposite polarity as the charging voltage to the non-floating gate. Programming or "writing to" an **EEPROM** location takes much longer than reading it so it is no substitute for read/write memories. Also, because of the thinner insulating layer, it can be worn out by repeated reprogramming. Typically, **EEPROM's can only be reprogrammed 10,000 times per location**. For this reason,

they are used for devices that need to hold long term data which is rarely changed, like set-up or configuration files. Computers use this type to save setup information. Before they became popular, we used CMOS memory with battery powered back-up.

5.5.2.3 The EAPROM

A final type of **PROM** is the **EAPROM**. It's a sister to the **E^2PROM** except that it is **serially programmed (and modified).**

- **Very slow,**
- **Very short life span,**
- **But very inexpensive.**

The **ROM, PROM, EPROM, E^2PROM, and EAPROM** are known as **non-volatile memory**. They are designed to keep data safe, even if power is lost. Another major use of a **PROM** is as a look-up table such as trig, exponential, and log functions. In addition, many numerical calculations like addition, subtraction, multiplication and division can be easily implemented with a **PROM.**

Design Rule for the PROM Family

When preparing to program a member of the PROM family, you need to represent each function in SOP form or derive a truth table for it. There is NO ADVANTAGE to minimizing the function since its canonical form must be used to generate the PROM fuse map.

PROM EXAMPLE: Design a full adder using a PROM.

	A_i	B_i	$Carry_{i-1}$	S_i	$Carry_i$
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

$$S_i = \sum m(1, 2, 4, 7)$$
$$C_i = \sum m(3, 5, 6, 7)$$