OJO OLAOLUWA JOEL

COMPUTER ENGINEERING

15/ENG02/041

COE506 ASSIGNMENT

Microcomputers and DSPs can often be applied with the necessary sequence of instructions to produce the desired circuit function. **The first category** of standard logic devices refers to the basic functional digital components (gates, flip-flops, decoders, multiplexers, register etc.)

The microprocessor/ digital signal processing (DSP) category is a much different approach to digital system design. With microprocessor/DSP systems, devices can be controlled electronically and data can be manipulated by implementing a program of instructions that have been written for the application

The third category is called application specific integrated circuits (ASICs). This broad category the modern hardware design solution for digital systems. It is designed to implement a specific desired application.

Programmable logic devices (PLDs) sometimes referred to as field programmable logic devices can be custom configured to create any desired digital circuit, from simple logic gates to complex digital systems.

Generally, PLDs can be described as being one of three different types: **simple programmable logic devices** (SPLDs), **complex programmable logic devices** (CPLDs) or **field programmable gate arrays** (FPGAs). The first PLD type to gain the interest of circuit designers was programmed by literally blowing fuses in the programming matrix. The fuses that were left intact in these one-time programmable devices provided electrical connections for the AND/OR circuits to produce desired functions. PLDs gained wide spread acceptance with the introduction of PALs (Programmable Array Logic). The programmable fuse in a PAL are used to determine the input connections to a set of AND gates that are wired to fixed OR gates. CLPDs are devices that typically combine an array of PAL type devices on the same chip.

FUNDAMENTALS OF PLD CIRCUITRY

A simple PLD circuit is shown below. Each of the four OR gates can produce an output that is a function of the two input variables A and B. each output function is programmed with the fuses located between and gates and each of the OR gates

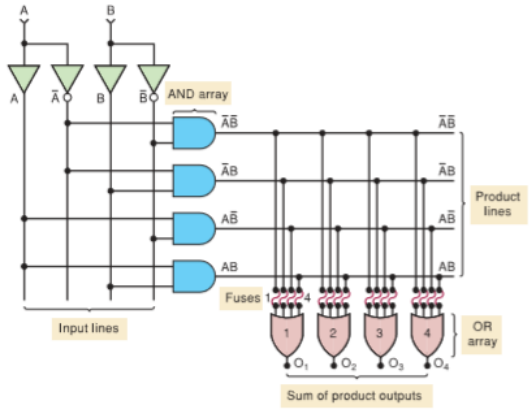

Figure 1: EXAMPLE OF A PROGRAMMABLE LOGIC DEVICE

Each of the inputs A and B feed both a non-inverting buffer and an inverting buffer to produce true and inverted forms of each variable. These are the input lines to the AND gate array. Each of the AND gates are connected to two different input lines to generate a unique product of the input variables. The AND outputs are called the product lines.

Each of the product lines is connected to one of four inputs of each OR gate through a fusible link. With all the links initially intact, each output will be a constant 1. Once all the outputs have been programmed, the device will permanently generate the selected output functions.

Figure 2 shows the same PLD circuit as figure 1 using the simplified symbols. The input numbers are represented as a single buffer with two outputs, one inverted or non-inverted. Then a single line is shown going to an AND gate to represent all four inputs. Each time the row line crosses a column represents a separate input to the AND gate. The connections from the input variable lines to the AND gate inputs are represented as dots. A dot means that this connection to the AND gate input is hard-wired (i.e. one that cannot be changed).

The inputs to each of the OR gates are also designated by a single line representing all four inputs. An X represents an intact fuse connecting a product line to one input of the OR gate. The absence of an X (or dot) at any intersection represents a blown fuse. For OR gate inputs, blown fuses are assumed to be LOW and for AND gate inputs, blown fuses are HIGH.

$O_1 = \overline{A}\,B + A\,\overline{B}$
$O_2 = AB$
$O_3 = 0$
$O_4 = 1$

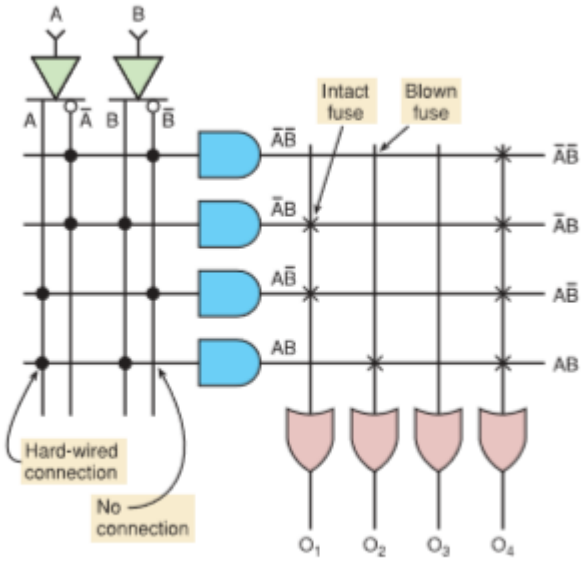

Figure 2: Simplified PLD symbology

PLD ARCHITECTURES

PROMs: the figure below shows how the PROM would be programmed to generate four specified logic functions. The first step is to construct a truth table showing the desired output level for all possible input combinations. Next write down the AND products for those cases where the output is to be 1. The

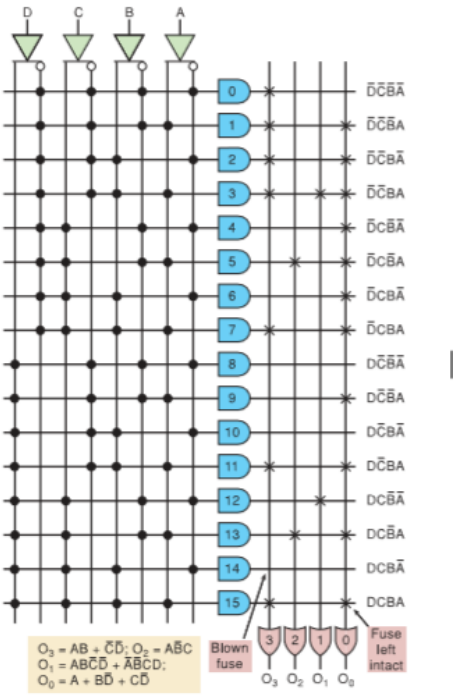

Figure 3: fuses are to be blown to program outputs for given functions.

output is to be the OR sum of these products. Thus, only the fuses that connect these product terms to the input of OR gate 3 are to be left intact. All the others are blown. The same procedure is followed to determine the status of the fuses at the other OR gate inputs.

The PROM can generate any possible logic function of the input variables because it generates every possible AND product term. Any application that requires every input combination to be available is a good candidate for PROM. However, PROMs get practical when a large number of input variables must be accommodated because the number of fuses double for each added input variable.

Programmable Array Logic (PAL)

When implementing SOP expressions, PROMs do not make very efficient use of circuitry. The architecture of PALs differs slightly form that of a PROM in that the inputs to the AND gates are programmable, whereas the inputs to the OR gates are hard-wired. This means that every AND gate can be programmed to generate any desired product of the four input variables and their compliments. Each OR gate is hard-wired to four product terms. If a function requires more than four product terms, it cannot be implemented with this PAL; one having more OR inputs will be used. If fewer than four product terms are required, the unnecessary ones can be made zero.

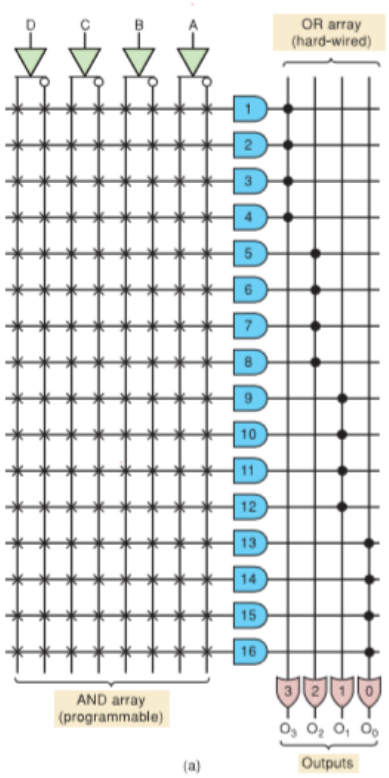

Figure 4: AND array (programmable)

FIELD PROGRAMMABLE LOGIC ARRAY (FPLA)

The field programmable logic array uses a programmable AND array as well as a programmable OR array.

THE GAL 16V8 (GENERIC ARRAY LOGIC)

The GAL chip uses an EEPROM array to control the programmable connections to the AND matrix, allowing them to be erased and reprogrammed at least 100 times. The GAL 16V8 contains optional flip-flops for register and counter applications, tristate buffers for the outputs, and control multiplexers used to select various modes of operation. The device has eight dedicated input pins, two special function input pins, and eight pins that can be used for both input and output.

The major components of the GAL devices are the input term matrix; the AND gates, which generate the product of input terms and the output logic microcells. The flexibility of the GAL lies in its programmable output logic microcell. Eight different products (outputs of AND gate) are applied as inputs to each of the eight-output logic macrocells. Within each OLMC, the products are ORed together to generate the sum of products (SOP). The TSMUX controls the tristate buffers enable input. If V_{CC} is selected, the output is always enabled, like standard combinational logic gate. If the grounded input is selected, the tri-state output of the inverter is always in its high impedance state. Another input to the MUX that may be selected comes from the OE input; this allows the output to be enabled or disabled by an external logic signal. The last possible input selection is a product term from the eighth AND gate; this allows an AND combination of terms from the input matrix to enable or disable the output.