The D-flip-flop model can be augmented to include asynchronous inputs and a complemented output as in the 74x74 discrete flip-flop, as shown in Table 8-7. This more detailed functional model shows the non-complementary behavior of the `Q` and `QN` outputs when preset and clear are asserted simultaneously. However, it does not include timing behavior such as propagation delay and setup and hold times, which are beyond the scope of the VHDL coverage in this book.

Larger registers can of course be modeled by defining the data inputs and outputs to be vectors, and additional functions can also be included. For example, Table 8-8 models a 16-bit register with three-state outputs and clock-enable, output-enable, and clear inputs. An internal signal vector `IQ` is used to hold the flip-flop outputs, and three-state outputs are defined and enabled as in Section 5.6.4.

SYNTHESIS RESTRICTIONS

In Table 8-8, the first `elsif` statement theoretically could have included all of the conditions needed to assign `D` to `IQ`. That is, it could have read "`elsif (CLK'event) and (CLK='1') and (CLKEN='1') then ...`" instead of using a nested `if` statement to check `CLKEN`. However, it was written as shown for a very pragmatic reason.

Only a subset of the VHDL language can be synthesized by the VHDL compiler that was used to prepare this chapter; this is true of any VHDL compiler today. In particular, use of the "`event`" attribute is limited to the form shown in the example, and a few others, for detecting simple edge-triggered behavior. This gets mapped into edge-triggered D flip-flops during synthesis. The nested `IF` statement that checks `CLKEN` in the example leads to the synthesis of multiplexer logic on the D inputs of these flip-flops.

8.3 Sequential PLDs

8.3.1 Bipolar Sequential PLDs

PAL16R8

The *PAL16R8*, shown in Figure 8-17, is representative of the first generation of sequential PLDs, which used bipolar (TTL) technology. This device has eight primary inputs, eight outputs, and common clock and output-enable inputs, and fits in a 20-pin package.

The PAL16R8's AND-OR array is exactly the same as the one found in the PAL16L8 combinational PLD. However, the PAL16R8 has edge-triggered D flip-flops between the AND-OR array and its eight outputs, O1–O8. All of the flip-flops are connected to a common clock input, CLK, and change state on the rising edge of the clock. Each flip-flop drives an output pin through a 3-state buffer; the buffers have a common output-enable signal, OE_L. Notice that, like

the combinational output pins of a PAL16L8, the registered output pins of the PAL16R8 contain the complement of the signal produced by the AND-OR array.

The possible inputs to the PAL16R8's AND-OR array are eight primary inputs (I1–I8) and the eight D flip-flop outputs. The connection from the D flip-flop outputs into the AND-OR array makes it easy to design shift registers, counters, and general state machines. Unlike the PAL16L8's combinational outputs, the PAL16R8's D-flip-flop outputs are available to the AND-OR array whether or not the O1–O8 three-state drivers are enabled. Thus, the internal flip-flops can go to a next state that is a function of the current state even when the O1–O8 outputs are disabled.

Many applications require combinational as well as sequential PLD outputs. The manufacturers of bipolar PLDs addressed this need by providing a few variants of the PAL16R8 that omitted the D flip-flops on some output pins, and instead provided input and output capability identical to that of the PAL16L8's bidirectional pins. For example, Figure 8-18 is the logic diagram of the *PAL16R6*, which has only six registered outputs. Two pins, IO1 and IO8, are bidirectional, serving both as inputs and as combinational outputs with separate 3-state enables, just like the PAL16L8's bidirectional pins. Thus, the possible inputs to the PAL16R6's AND-OR array are the eight primary inputs (I1–I8), the six D-flip-flop outputs, and the two bidirectional pins (IO1, IO8).

PAL16R6

PAL16R4
PAL16R8
PAL20L8
PAL20R4

Table 8-9 shows eight standard bipolar PLDs with differing numbers and types of inputs and outputs. All of the PAL16xx parts in the table use the same AND-OR array, where each output has eight AND gates, each with 16 variables and their complements as possible inputs. The PAL20xx parts use a similar

Table 8-9 Characteristics of standard bipolar PLDs.

			Inputs to AND array			
Part number	*Package pins*	*AND-gate inputs*	*Primary inputs*	*Bidirectional combinational outputs*	*Registered outputs*	*Combinational outputs*
PAL16L8	20	16	10	6	0	2
PAL16R4	20	16	8	4	4	0
PAL16R6	20	16	8	2	6	0
PAL16R8	20	16	8	0	8	0
PAL20L8	24	20	14	6	0	2
PAL20R4	24	20	12	4	4	0
PAL20R6	24	20	12	2	6	0
PAL20R8	24	20	12	0	8	0

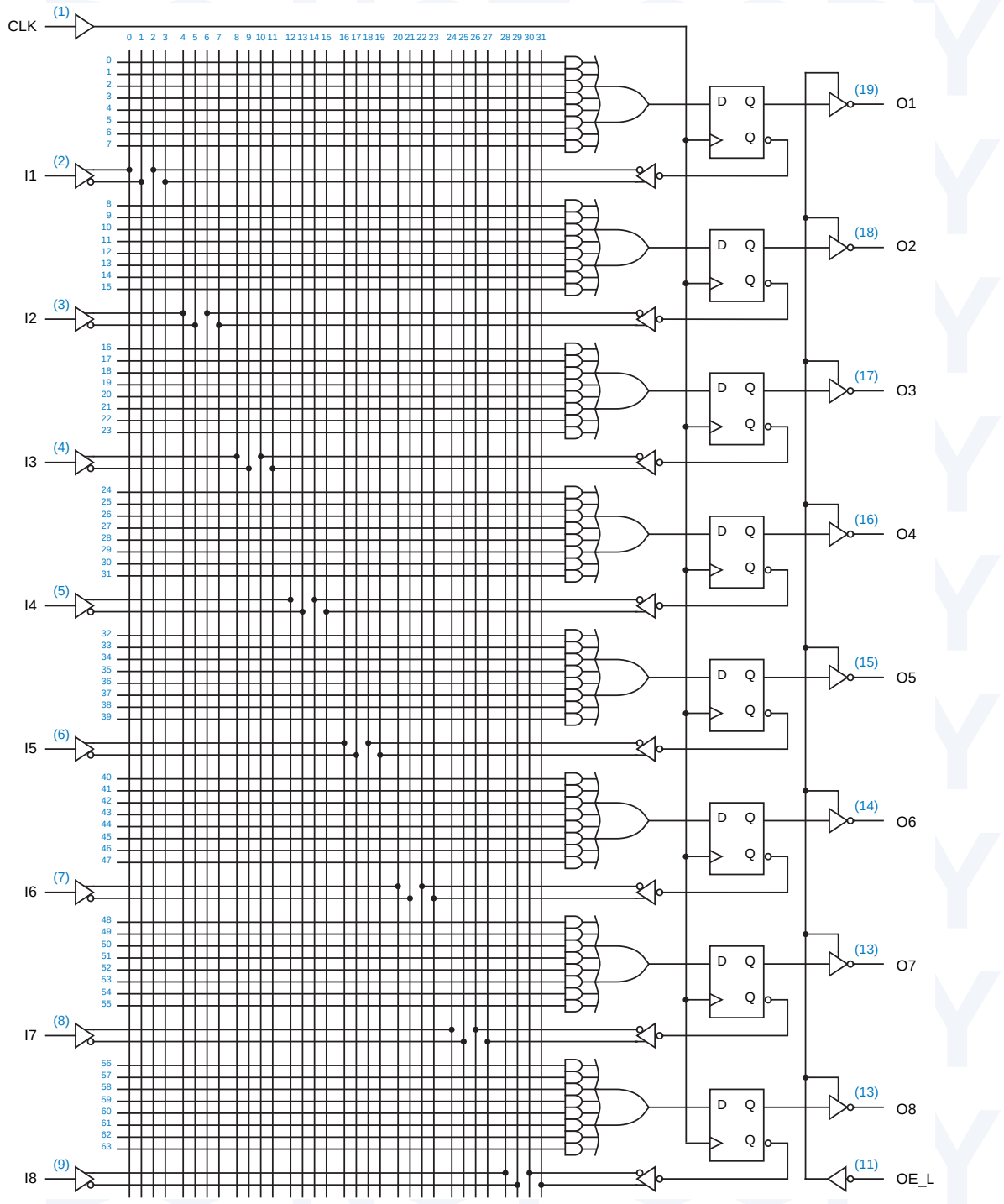


Figure 8-17 PAL16R8 logic diagram.

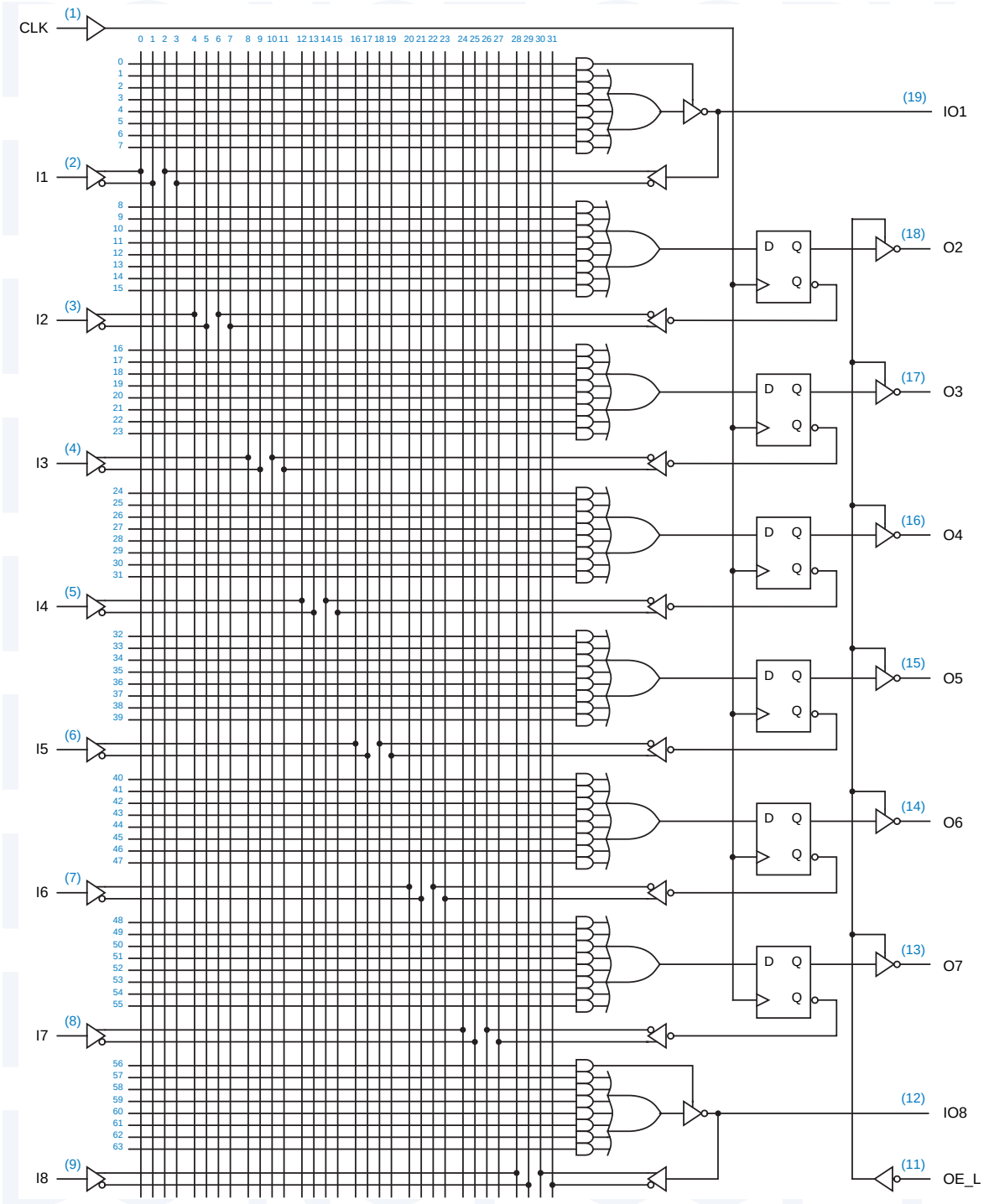


Figure 8-18 PAL16R6 logic diagram.

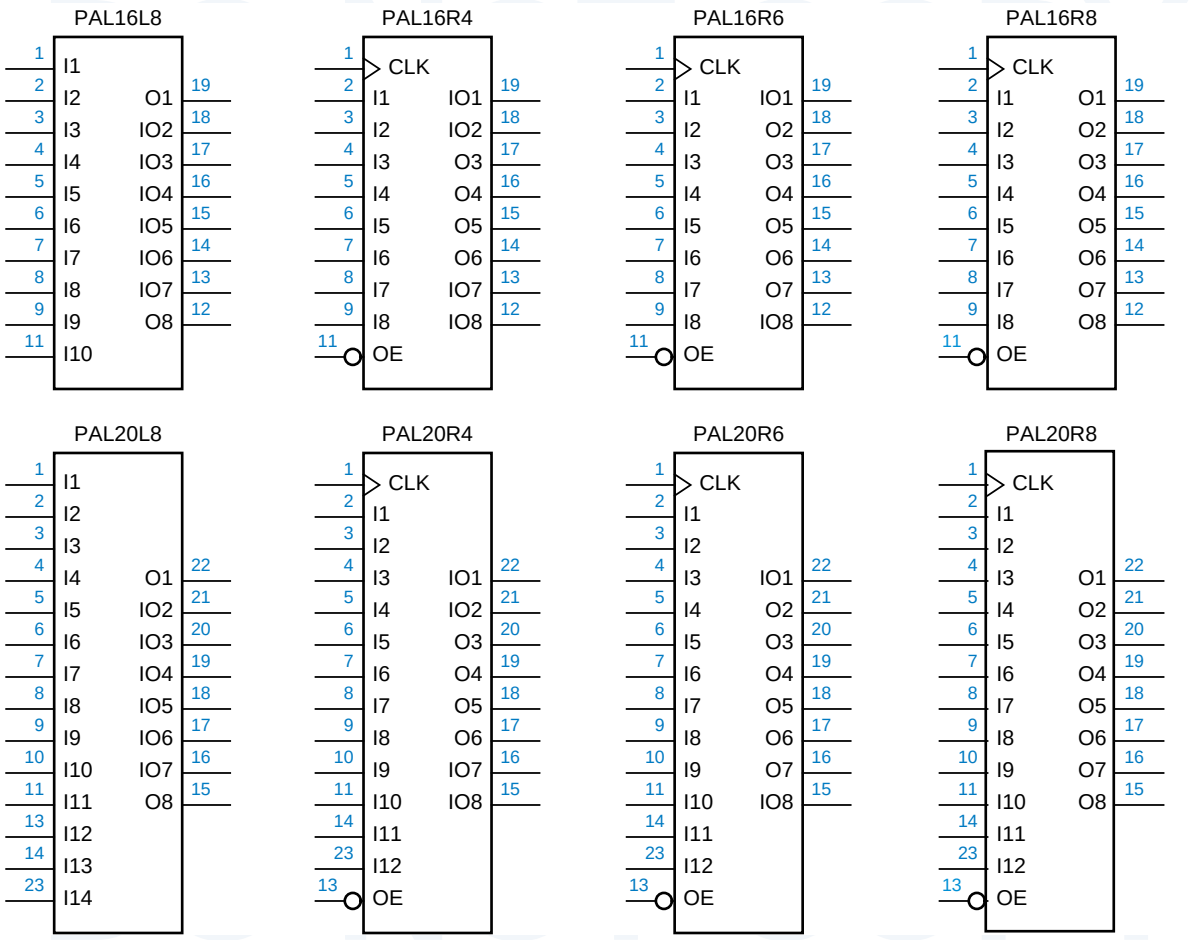


Figure 8-19 Logic symbols for bipolar combinational and sequential PLDs.

AND-OR array with 20 variables and their complements as possible inputs. Figure 8-19 shows logic symbols for all of the PLDs in the table. *PAL20R6* *PAL20R8*

8.3.2 Sequential GAL Devices

The GAL16V8 electrically erasable PLD was introduced in Section 5.3.3. Two "architecture-control" fuses are used to select among three basic configurations of this device. Section 5.3.3 described the *16V8C* ("complex") configuration, shown in Figure 5-27 on page 307, a structure similar to that of a bipolar combinational PAL device, the PAL16L8. The *16V8S* ("simple") configuration provides a slightly different combinational logic capability (see box on page 587). *16V8C* *16V8S*

The third configuration, called the *16V8R*, allows a flip-flop to be provided on any or all of the outputs. Figure 8-20 shows the structure of the device when *16V8R*

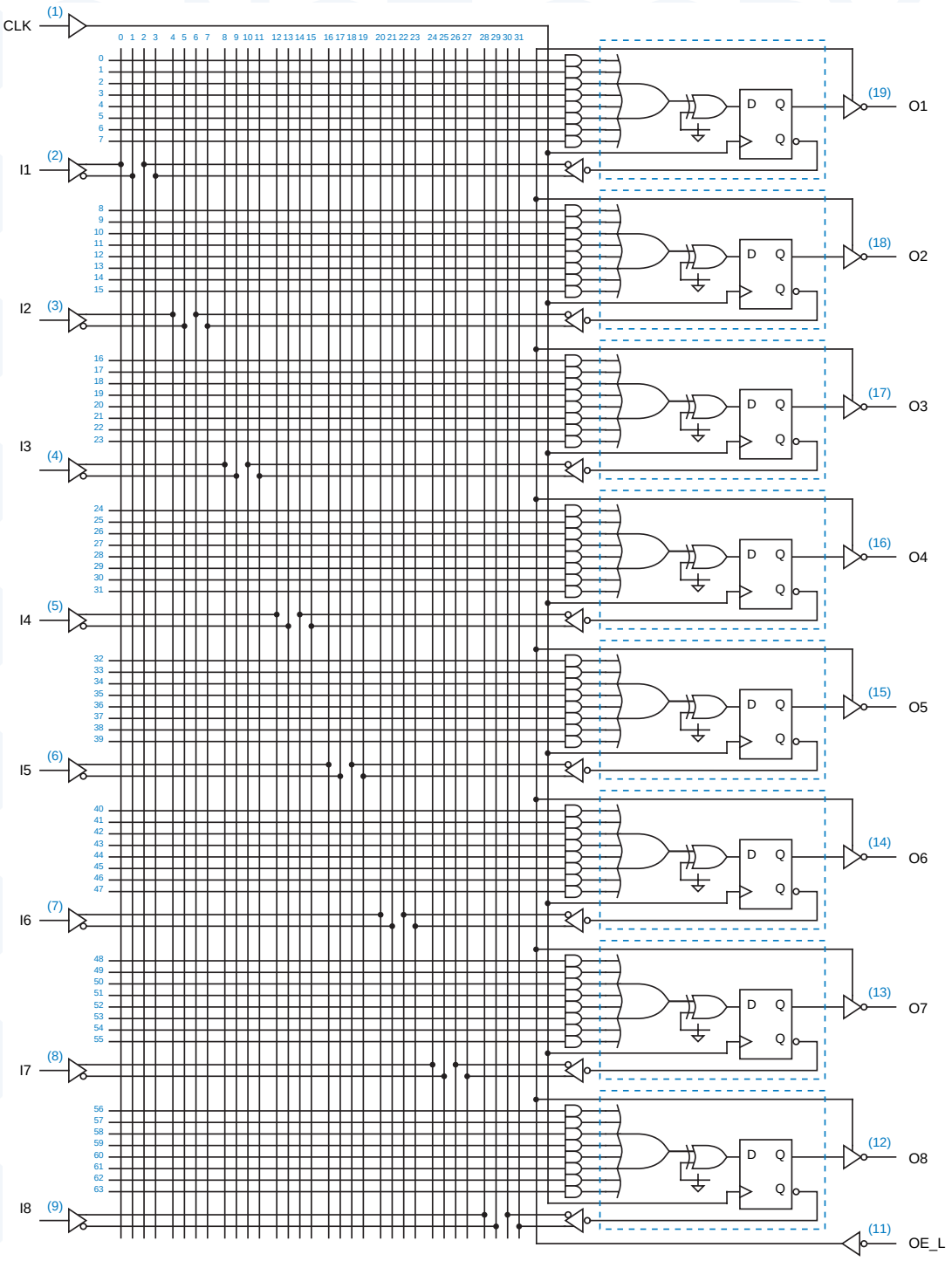


Figure 8-20 Logic diagram for the 16V8 in the “registered” configuration.

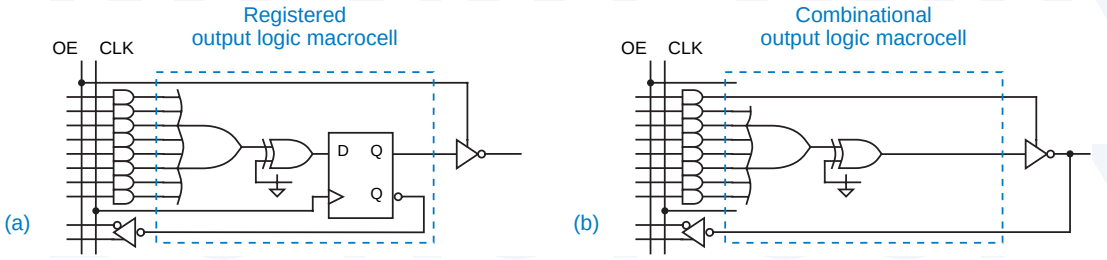


Figure 8-21 Output logic macrocells for the 16V8R: (a) registered; (b) combinational.

flip-flops are provided on all outputs. Notice that all of the flip-flops are controlled by a common clock signal on pin 1, as in the bipolar devices of the preceding subsection. Likewise, all of the output buffers are controlled by a common output-enable signal on pin 11.

The circuitry inside each dotted box in Figure 8-20 is called an *output logic macrocell*. The 16V8R is much more flexible than a PAL16R8 because each macrocell may be individually configured to bypass the flip-flop, that is, to produce a combinational output. Figure 8-21 shows the two macrocell configurations that are possible in the 16V8R; (a) is registered and (b) is combinational. Thus, it is possible to program the device to have any set of registered and combinational outputs, up to eight total.

output logic macrocell

The *20V8* is similar to the 16V8, but comes in a 24-pin package with four extra input-only pins. Each AND gate in the 20V8 has 20 inputs, hence the "20" in "20V8."

20V8

THE "SIMPLE" 16V8S

The "simple" 16V8S configuration of the GAL16V8 is not often used, because its capabilities are mostly a subset of the 16V8C's. Instead of an AND term, the 16V8S uses one fuse per output to control whether the output buffers are enabled. That is, each output pin may be programmed either to be always enabled or to be always disabled (except pins 15 and 16, which are always enabled). All of the output pins (except 15 and 16) are available as inputs to the AND array regardless of whether the output buffer is enabled.

The only advantage of a 16V8S compared to a 16V8C is that it has eight, not seven, AND terms as inputs to the OR gate on each output. The 16V8S architecture was designed mainly for emulation of certain now-obsolete bipolar PAL devices, some of which either had eight product terms per output or had inputs on pins 12 and 19, which are not inputs in the 16V8C configuration. With appropropriate programming, the 16V8S can be used as a pin-for-pin compatible replacement for these devices, which included the PAL10H8, PAL12H6, PAL14H4, PAL16H2, PAL10L8, PAL12L6, PAL14L4, and PAL16L2.

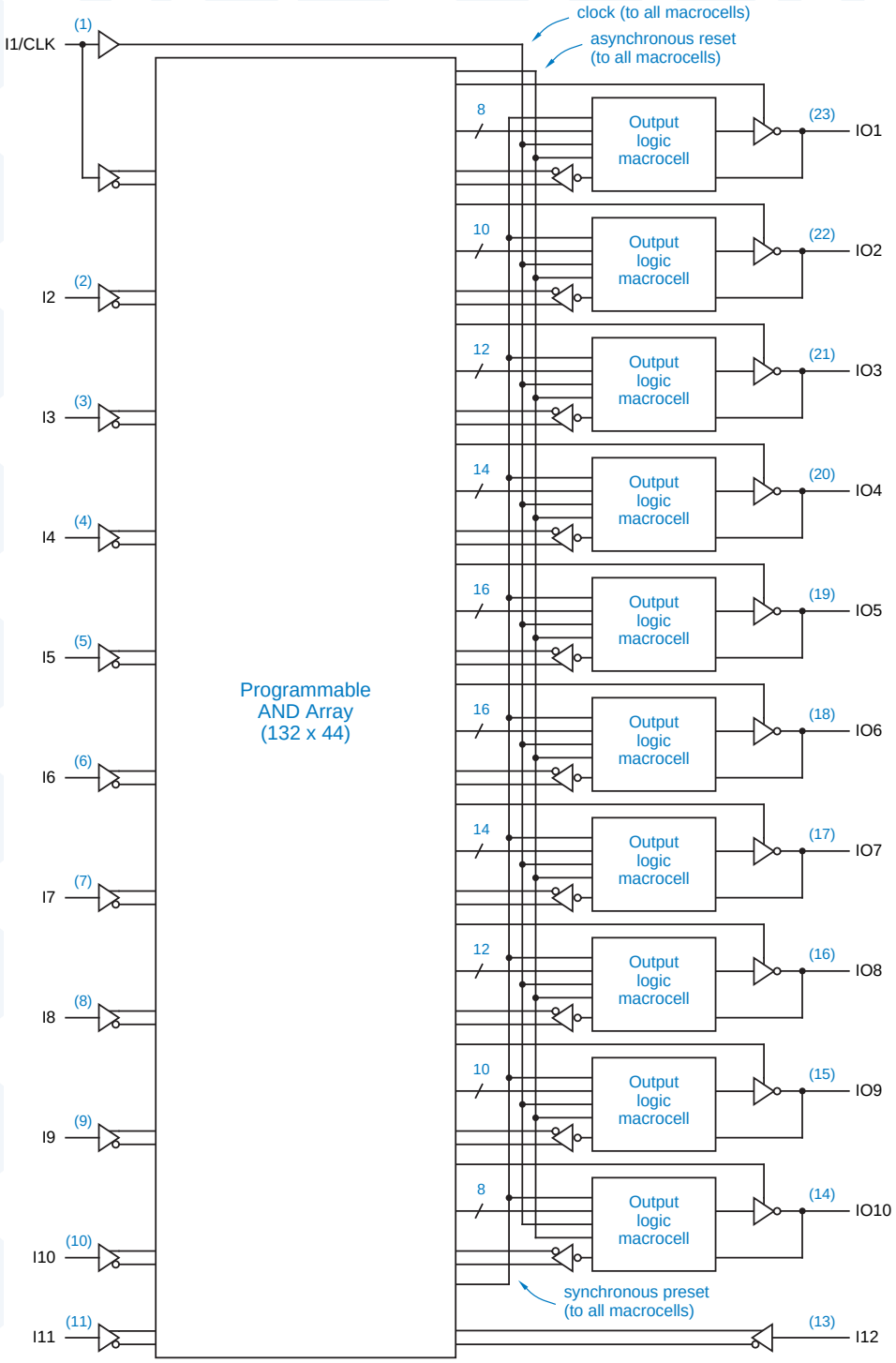


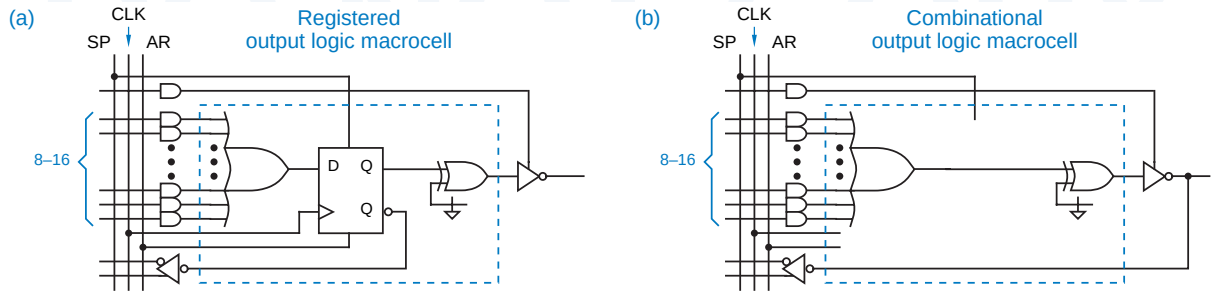
Figure 8-22
Logic diagram for the 22V10.

22V10

The *22V10*, whose basic structure is shown in Figure 8-22, also comes in a 24-pin package, but is somewhat more flexible than the 20V8. The 22V10 does not have "architecture control" bits like the 16V8's and 20V8's, but it can realize any function that is realizable with a 20V8, and more. It has more product terms, two more general-purpose inputs, and better output-enable control than the 20V8. Key differences are summarized below:

- Each output logic macrocell is configurable to have a register or not, as in the 20V8R architecture. However, the macrocells are different from the 16V8's and 20V8's, as shown in Figure 8-23.
- A single product term controls the output buffer, regardless of whether the registered or the combinational configuration is selected for a macrocell.
- Every output has at least eight product terms available, regardless of whether the registered or the combinational configuration is selected. Even more product terms are available on the inner pins, with 16 available on each of the two innermost pins. ("Innermost" is with respect to the right-hand side of the Figure 8-22, which also matches the arrangement of these pins on a 24-pin dual-inline package.)
- The clock signal on pin 1 is also available as a combinational input to any product term.
- A single product term is available to generate a global, asynchronous reset signal that resets all internal flip-flops to 0.
- A single product term is available to generate a global, synchronous preset signal that sets all internal flip-flops to 1 on the rising edge of the clock.
- Like the 16V8 and 20V8, the 22V10 has programmable output polarity. However, in the registered configuration, the polarity change is made at the output, rather than the input, of the D flip-flop. This affects the details of programming when the polarity is changed but does not affect the overall capability of the 22V10 (i.e., whether a given function can be realized). In fact, the difference in polarity-change location is transparent when you use a PLD programming language such as ABEL.

Figure 8-23 Output logic macrocells for the 22V10: (a) registered; (b) combinational.



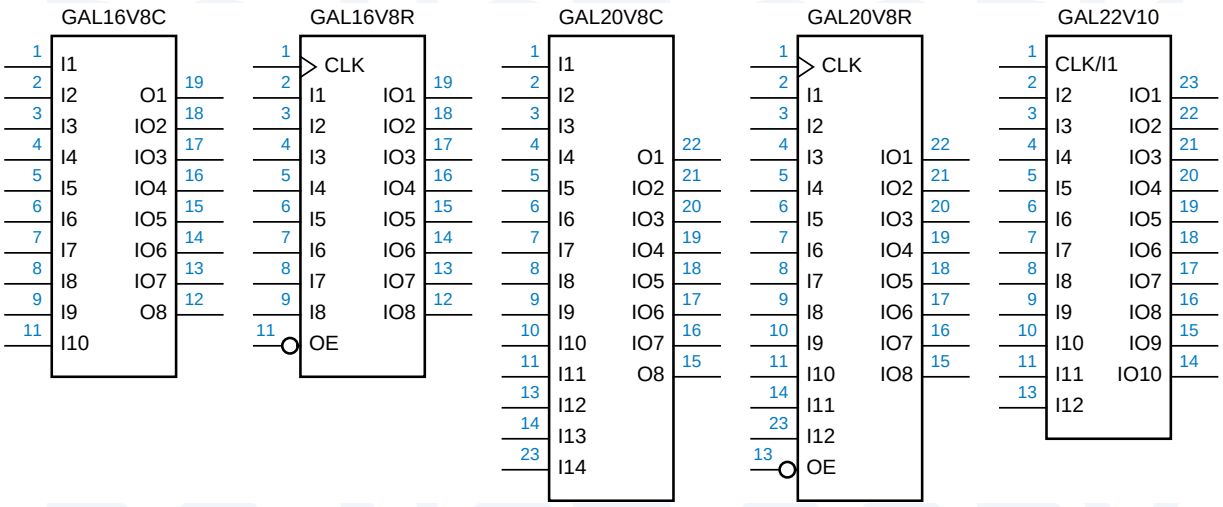


Figure 8-24 Logic symbols for popular GAL devices.

For most of the 1990s, the 16V8, 20V8, and 22V10 were the most popular and cost-effective PLDs (but see the box on page 593). Figure 8-24 shows generic logic symbols for these three devices. Most of the examples in the rest of this chapter can fit into the smallest of the three devices, the 16V8.

PALS? GALS?

Lattice Semiconductor introduced GAL devices including the GAL16V8 and GAL20V8 in the mid-1980s. Advanced Micro Devices later followed up with a pin-compatible device which they call the PALCE16V8 ("C" is for CMOS, "E" is for erasable). Several other manufacturers make differently numbered but compatible devices as well. Rather than get caught up in the details of different manufacturers' names, in this chapter we usually refer to commonly used GAL devices with their generic names, 16V8, 20V8, and 22V10.

8.3.3 PLD Timing Specifications

Several timing parameters are specified for combinational and sequential PLDs. The most important ones are illustrated in Figure 8-25 and are explained below:

t_{PD}

feedback input

t_{PD} This parameter applies to combinational outputs. It is the propagation delay from a primary input pin, bidirectional pin, or "feedback" input to the combinational output. A *feedback input* is an internal input of the AND-OR array that is driven by the registered output of an internal macrocell.

t_{CO} This parameter applies to registered outputs. It is the propagation delay from the rising edge of CLK to a primary output.

t_{CF} This parameter also applies to registered outputs. It is the propagation delay from the rising edge of CLK to a macrocell's registered output that conects back to a feedback input. If specified, t_{CF} is normally less than t_{CO}. However, some manufacturers do not specify t_{CF}, in which case you must assume that $t_{CF} = t_{CO}$.

t_{SU} This parameter applies to primary, bidirectional, and feedback inputs that proprogate to the D inputs of flip-flops. It is the setup time that the input signal must be stable before the rising edge of CLK.

t_{H} This parameter also applies to signals that proprogate to the D inputs of flip-flops. It is the hold time that the input signal must be stable after the rising edge of CLK.

f_{max} This parameter applies to clocked operation. It is the highest frequency at which the PLD can operate reliably, and is the reciprocal of the minimum clock period. Two versions of this parameter can be derived from the previous specifications, depending on whether the device is operating with external feedback or internal feedback.

External feedback refers to a circuit in which a registered PLD output is connected to the input of another registered PLD with similar timing; for proper operation, the sum of t_{CO} for the first PLD and t_{SU} for the second must not exceed the clock period.

Internal feedback refers to a circuit in which a registered PLD output is fed back to a register in the same PLD; in this case, the sum of t_{CF} and t_{SU} must not exceed the clock period.

Each of the PLDs that we described in previous sections is available in several different speed grades. The speed grade is usually indicated by a suffix on the part number, such as "16V8-10"; the suffix usually refers to the t_{PD}

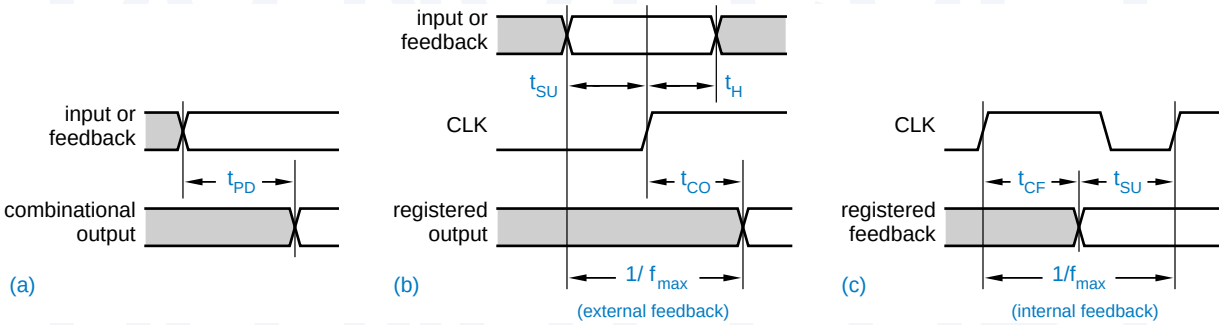


Figure 8-25 PLD timing parameters.

Table 8-10 Timing specifications, in nanoseconds, of popular bipolar and CMOS PLDs.

Part numbers	Suffix	t_{PD}	t_{CO}	t_{CF}	t_{SU}	t_H
PAL16L8, PAL16Rx, PAL20L8, PAL20Rx	-5	5	4	–	4.5	0
PAL16L8, PAL16Rx, PAL20L8, PAL20Rx	-7	7.5	6.5	–	7	0
PAL16L8, PAL16Rx, PAL20L8, PAL20Rx	-10	10	8	–	10	0
PAL16L8, PAL16Rx, PAL20L8, PAL20Rx	B	15	12	–	15	0
PAL16L8, PAL16Rx, PAL20L8, PAL20Rx	B-2	25	15	–	25	0
PAL16L8, PAL16Rx, PAL20L8, PAL20Rx	A	25	15	–	25	0
PALCE16V8, PALCE20V8	-5	5	4	–	3	0
GAL16V8, GAL20V8	-7	7.5	5	3	5	0
GAL16V8, GAL20V8	-10	10	7.5	6	7.5	0
GAL16V8, GAL20V8	-15	15	10	8	12	0
GAL16V8, GAL20V8	-25	25	12	10	15	0
PALCE22V10	-5	5	4	–	3	0
PALCE22V10	-7	7.5	4.5	–	4.5	0
GAL22V10	-10	10	7	2.5	7	0
GAL22V10	-15	15	8	2.5	10	0
GAL22V10	-25	25	15	13	15	0

specification, in nanoseconds. Table 8-10 shows the timing of several popular bipolar and CMOS PLDs. Note that only the t_{PD} column applies to the combinational outputs of a device, while the last four columns apply to registered outputs. All of the timing specifications are worst-case numbers over the commercial operating range.

When sequential PLDs are used in applications with critical timing, it's important to remember that they normally have longer setup times than discrete edge-triggered registers in the same technology, owing to the delay of the AND-OR array on each D input. Conversely, under typical conditions, a PLD actually has a negative hold-time requirement because of the delay through AND-OR array. However, you can't count on it having a negative hold time—the worst-case specification is normally zero.

HOW MUCH DOES IT COST?

Once you understand the capabilities of different PLDs, you might ask, "Why not just always use the most capable PLD available?" For example, even if a circuit fits in a 20-pin 16V8, why not specify the slightly larger, 24-pin 20V8 so that spare inputs are available in case of trouble? And, once you've specified a 20V8, why not use the somewhat more capable 22V10 which comes in the same 24-pin package?

In the real world of product design and engineering, the constraint is cost. Otherwise, the argument of the previous paragraph could be extended *ad nauseum*, using CPLDs and FPGAs with even more capability (see \chapref{CPLDsFPGAs).

Like automobiles and fine wines, digital devices such as PLDs, CPLDs, and FPGAs are not always priced proportionally to their capabilities and benefits. In particular, the closer a device's capability is to the "bleeding edge," the higher the premium you can expect to pay. Thus, when selecting a devices to realize a design, you must evaluate many trade-offs. For example, a high-density, high-cost CPLD or FPGA may allow a design to be realized in a single device whose internal functions are easily changed if need be. On the other hand, using two or more lower density PLDs, CPLDs, or FPGAs may save component cost but increase board area and power consumption, while making it harder to change the design later (since the device interconnections must be fixed when the board is fabricated).

What this goes to show is that overall cost must always be considered along with design elegance and convenience to create a successful (i.e., profitable) product. And minimizing the cost of a product usually involves a plethora of common-sense economic and engineering considerations that are far removed from the turn-the-crank, algorithmic gate minimization methods of Chapter 4.

8.4 Counters

The name *counter* is generally used for any clocked sequential circuit whose state diagram contains a single cycle, as in Figure 8-26. The *modulus* of a counter is the number of states in the cycle. A counter with *m* states is called a *modulo-m counter* or, sometimes, a *divide-by-m counter*. A counter with a non-power-of-2 modulus has extra states that are not used in normal operation.

counter
modulus
modulo-m counter
divide-by-m counter

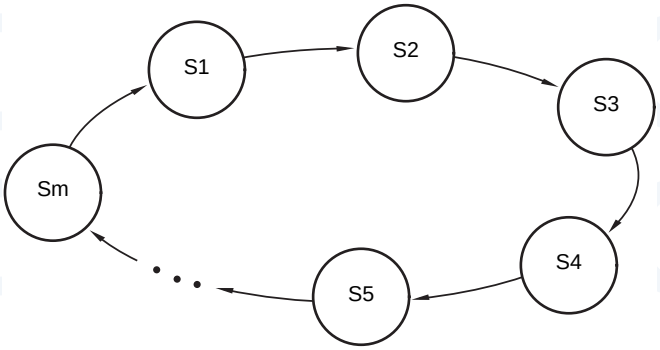


Figure 8-26 General structure of a counter's state diagram—a single cycle.